Complemented circuits

A. Bernasconi* R. K. Brayton† V. Ciriani‡ G. Trucco§ T. Villa¶

1 Decomposition with two-input Boolean operators

In the standard two level minimization, we can synthesize a function f in a SOP or a POS (i.e., the negation of a SOP) form. Some functions have a small SOP expression and others have a more compact POS form. In this work we study a generalization of SOP and POS forms that is a new three-level form, called *complemented circuit*, aiming at representing a Boolean function f as $c = \overline{f}_0 \; op \; f_1$, where op is a two input gate or LUT, and f_0 and f_1 are two SOP forms. As an example with $op(f_0, f_1) = \overline{f_0} f_1$, see the scheme depicted in Figure 1 (a). Note that if $f_0 = \emptyset$ (resp., $f_1 = \emptyset$) then c is a SOP (resp., POS) expression. Intuitively, we can distribute product terms between f_0 and f_1 according to the size of the POS and SOP, and to the don't care conditions induced by the op operator, as it will be exemplified in the motivating example. Observe that this can be seen as a special case of functional decomposition [17, 22].

The covers f_0 and f_1 are related to the off-set and on-set of f as follows. Consider the standard SOP minimization of a Boolean function f represented by the on-set f_{on}, the dc-set f_{dc}, and the off-set f_{off}. In general, a cover g of f is such that $f_{on} \subseteq g \subseteq f_{on} \cup f_{dc}$. It is also possible to find a cover $\overline{h}$ complementing the SOP h such that $f_{off} \subseteq h \subseteq f_{off} \cup f_{dc}$. So, considering our proposed logic structure, say when $op = OR$, we can implement a minterm in the on-set of f by one of three ways: by adding its cube to both the cover f_1 and the cover f_0 (the output of the latter is negated and becomes a 0 input to the OR gate), or by adding it to the cover f_1 and by not adding it to the cover f_0 (i.e., it is put in $\overline{f}_0$) so that the 0 output of f_0 is negated and becomes a 1 input to the OR gate, or by not adding it to the cover f_1 and not adding it to the cover f_0 whose negated output inputs a 1 into the OR gate. In other words, a point in the on-set can be realized by f_1 only, or by f_1 and $\overline{f}_0$, or by $\overline{f}_0$ only. In the special case where $f_0 = \emptyset$ (resp., $f_1 = \emptyset$) we have that $c = g$ (resp., $c = \overline{h}$). Intuitively, this is a generalization of the problem of choosing the best output phase assignment in the realization of a SOP (see [18]), where on top of that we exploit the functionality of the two-input gate collecting the outputs of $\overline{f}_0$ and f_1.

The previous example shows that the introduction of a two-input operator connecting the two logic blocks $\overline{f}_0$ and f_1 induces don't care conditions: e.g., when $op = OR$, if a point in the on-set is added to f_1 then we do not care if it is added also to f_0 or not, yielding the don't care condition (-1) for the two outputs of f_0 and f_1; dually, the same outputs can assume values in the don't care cube $(0-)$, since if an on-set point is not added to f_0 we do not care if it is part also of f_1 or not. However, the two output cubes $(0-)$ and (-1) cannot be expressed exactly by a single cube, but instead a Boolean relation is required to capture such degrees of indeterminacy. Therefore, we characterize the problem of optimizing the implementation of the decomposition of f in the form $c = \overline{f}_0 \; op \; f_1$, as one of defining and optimizing Boolean relations depending on op. This approach can be exploited in logic synthesis or in other contexts such as data mining [5, 13].

We had started this investigation in [3], where we gave partial results for a subset of the 10 non-trivial Boolean operators depending on two variables. In particular, we focused on ops which have one of the inputs inverted to contrast the proposed approach with the literature dealing with AND-OR-AND and AND-OR-XOR three-level forms. In this chapter we present a summary of the results for all 10 Boolean operators Moreover, here we adopted an easier flow

*Dipartimento di Informatica, Università di Pisa, Italy, anna.bernasconi@unipi.it
†Department of EECS, University of California, Berkeley, USA, brayton@berkeley.edu
‡Università degli Studi di Milano, Italy, valentina.ciriani@unimi.it
§Università degli Studi di Milano, Italy, gabriella.trucco@unimi.it
¶Università degli Studi di Verona, Italy, tiziano.villa@univr.it

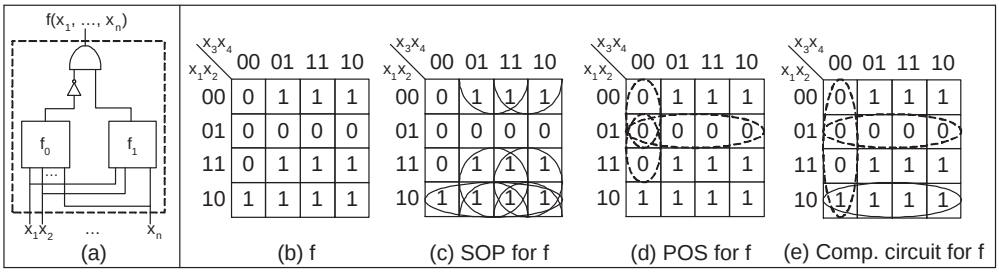

Figure 1: Circuit decomposition with AND gate (a) and example of SOP form (c) POS form (d) and complemented circuit with OR gate (e) for the Boolean function (b).

of exposition that builds upon the primitive intuition of rebalancing product terms between a representation of a function and of its complement. Finally, here we discuss how to set up a unique Boolean relation for multioutput functions.

This chapter is an extended version of the conference paper in [4] and is organized as follows: we summarize briefly previous work in Section 2, Boolean relations in Section 3, and describe in Section 4 the Boolean relations characterizing completely the flexibility in the realization of complemented circuits. Section 5 provides a discussion on the structure of the complemented circuits that we have defined. In Section 6, we report experimental results comparing our forms vs. SOPs (with different phase assignments) and specialized three-level forms like AOXMIN, after running SIS to compute areas and delays of the underlying SOPs; the experiments show average gains around $20-30\%$ in the majority of benchmarks. In this last section, we also draw conclusions and discuss possible future research.

1.1 Motivating Example

To provide more intuition for the proposed approach, we discuss a simple example of the OR complemented circuit $\overline{f}_0 + f_1$ of the function f depicted in Figure 1(b). By the De Morgan laws, a complemented SOP (Sum of Products) can be seen as a POS (Product of Sums) form with the same number of literals. For example: $\overline{x_1\overline{x}_2 + x_1x_3\overline{x}_4} = (\overline{x}_1 + x_2)(\overline{x}_1 + \overline{x}_3 + x_4)$.

Consider the function f represented by the Karnaugh map in Figure 1(b). If we compute a standard SOP cover we have the minimal SOP in Figure 1(c):

$$f^{SOP} = f_1^{SOP} = x_1\overline{x}_2 + x_1x_3 + x_1x_4 + \overline{x}_2x_3 + \overline{x}_2x_4\,,$$

that has 5 products and 10 literals. From the minimal SOP form $\overline{f} = \overline{x}_1x_2 + \overline{x}_1\overline{x}_3\overline{x}_4 + x_2\overline{x}_3\overline{x}_4$ in Figure 1(d), we get the corresponding minimal POS form:

$$f^{POS} = \overline{f}_0^{POS} = (x_1 + \overline{x}_2)(x_1 + x_3 + x_4)(\overline{x}_2 + x_3 + x_4)\,,$$

containing 3 products and 8 literals. Considering the number of products and literals, the POS representation is then more convenient vs. the SOP form. Moreover, if we enlarge the off-set to include the on-set point 1000, the minimal SOP of the enlarged off-set becomes $\overline{x}_1x_2 + \overline{x}_3\overline{x}_4$ yielding the POS $(x_1+\overline{x}_2)(x_3+x_4)$; however, to preserve the functionality, we must represent the on-set point 1000 by adding a product of the on-set that covers it, say $\overline{x}_1x_2$, paying a penalty of 1 product and 2 literals, with an overall cost of 3 products and 6 literals (better than 3 products and 8 literals). In conclusion, a minimal OR complemented circuit for f in Figure 1(e) is:

$$f^{CC} = \overline{f}_0 + f_1 = (x_1 + \overline{x}_2)(x_3 + x_4) + x_1\overline{x}_2\,,$$

(it is $f_0 = \overline{x}_1x_2 + \overline{x}_3\overline{x}_4$ with a cost of 2 products and 4 literals, and $f_1 = x_1\overline{x}_2$ with a cost of 1 product and 2 literals, for a total of 3 products and 6 literals).

Note that in Figure 1(d) the point 1000 is *both* in the off-set of $\overline{f}_0$ and in the on-set of f_1, thus it is in the on-set of the OR between $\overline{f}_0$ and f_1 $(\overline{f}_0 + f_1)$. Moreover, the points $1001, 1010$, and 1011 are *both* in the on-sets of $\overline{f}_0$ and f_1, i.e., they are covered by both $\overline{f}_0$ and f_1. This leads to investigate how to find the best cover $f^{CC} = \overline{f}_0$ *op* f_1. Note that the best solution could be $f^{CC} = f_1$ (i.e., SOP) or $f^{CC} = \overline{f}_0$ (i.e., POS or complemented SOP).

2 Previous Work

Three-level logic has been the object of intensive research for decades, a reason being that three levels are enough to produce a minimal network for most Boolean functions (see Sasao, [19]). The minimization of various forms of tree-level logic has been studied in the literature, e.g., AND-OR-AND networks consisting of two SOPs with a two-input AND gate at the output (Malik, [15] and Dubrova, [11]); OR-AND-OR networks (see Sasao, [20], and Debnath, [10]); AND-OR-EXOR networks, called EX-SOP, with a single two-input EXOR gate at the output (see Debnath, [8, 9] and Dubrova, [12]); EXOR-AND-OR networks, called SPPs (Sums of Pseudo-Products) which generalize SOP expressions by replacing products of literals with products of EXOR gates (see Luccio, [14]), further restricted to k-SPPs where EXOR factors contain at most k literals and to 2-SPPs (see Ciriani, [6, 7]) for which an efficient ESPRESSO-like minimization procedure has been designed (see Bernasconi, [2]). A way to obtain three-level forms is to apply one step of bi-decomposition, which decomposes a given logic function $F(X)$ into three blocks as $F(X) = G(X)$ *op* $H(X)$, where *op* is a two-input gate (see Sasao, [21] and Mishchenko, [16]). A strong bi-decomposition has the form $G(X_1, X_3)$ *op* $H(X_3, X_2)$ where X_1, X_2, X_3 are disjoint sets of variables; When $X_2 = \emptyset$ the bi-decomposition is weak. In our work we address the special case of weak decomposition where $X_1 = X_2 = \emptyset$.

3 Boolean Relations

The concept of Boolean relations was introduced as a more general scheme for the non-deterministic specification of logic networks, which cannot always be represented using don't cares [1].

Definition 1 *A* Boolean relation *is a one-to-many multioutput Boolean mapping* $\mathcal{R} : \{0,1\}^n \rightarrow \{0,1\}^m$*, where* $\{0,1\}^n$ *and* $\{0,1\}^m$ *are called the* input *and* output sets *of* $\mathcal{R}$*.*

A Boolean relation $\mathcal{R}$ can be considered a generalization of a Boolean function, where a point in the input set $\{0,1\}^n$ can be associated with several points in the output set $\{0,1\}^m$; indeed, because of the one-to-many nature of Boolean relations, there may be several equivalent outputs for a given input. For example, consider the mapping $\mathcal{R} : \{0,1\}^2 \rightarrow \{0,1\}^3$ such that:

x	$\mathcal{R}(x)$
00	$\{001, 100\}$
01	$\{000\}$
10	$\{101, 111\}$
11	$\{100, 010, 001\}$

Note that we cannot represent this mapping using a simple incompletely specified function, since, for example, the input 00 can have as output 001 or 100 that cannot be merged into a single cube. A relation $\mathcal{R}$ is *well-defined* if for all $x \in \{0,1\}^n$ there is $y \in \{0,1\}^m$ such that $(x, y) \in \mathcal{R}$. To any relation $\mathcal{R}$ we can associate a set $\mathcal{F}(\mathcal{R})$ of *compatible* multioutput Boolean functions, i.e. the set of all functions f such that, for all inputs $x \in \{0,1\}^n$, $f(x)$ is contained in the set $\mathcal{R}(x)$ of the outputs related to x. In this case, we write $f \subseteq \mathcal{R}$.

For example, consider the mapping $\mathcal{R}$ described in the previous example. $\mathcal{F}(\mathcal{R})$ contains the following Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}^m$:

x	$f(x)$
00	001
01	000
10	101
11	100

The problem of the optimal implementation of a Boolean relation $\mathcal{R}$ is that of selecting, among the possible functions compatible with $\mathcal{R}$, one of minimum cost according to a given metric. More precisely, the *solution* of a Boolean relation $\mathcal{R}$ is a multioutput Boolean function $f \in \mathcal{F}(\mathcal{R})$. The function f is an *optimal solution* of $\mathcal{R}$ according to a given cost function C, if for all $f' \in \mathcal{F}(\mathcal{R})$, $C(f) \leq C(f')$. In this chapter, the cost $C(f)$ will be the number of literals in a minimal SOP form for f.

4 Complemented circuits

Given a function f, defined as on-set, off-set and dc-set, we want to decompose f in f_0 and f_1, such that f is covered by $\overline{f}_0$ *op* f_1, where *op* is a given binary operation, e.g., it may be AND, OR or XOR gate. The inputs of f_0 and f_1 are the same as the inputs of f, i.e. the entire set $x_1, \ldots, x_n$. The output of f is the output of the chosen gate *op* which takes as input $\overline{y}_0$ and y_1, where $y_0 = f_0(x_1, \ldots, x_n)$ and $y_1 = f_1(x_1, \ldots, x_n)$. Figure 1(a) reports the complemented circuit obtained when *op* is represented by the AND gate. More precisely, we want to solve the problem of finding the sets f_0 and f_1 that lead to a complemented circuit of minimal cost, according to a given cost metric. In this chapter we consider SOP minimization for f_0 and f_1, thus we refer to a complemented circuit as a:

$$C(f) = \overline{SOP_0} \;\; op \; SOP_1, \tag{1}$$

where SOP_0 is an SOP form for f_0 and SOP_1 is an SOP form for f_1. This problem can be nicely formalized and efficiently solved using Boolean relations. Indeed, as shown in this section, for each binary operation *op*, we can define a relation $\mathcal{R}_{op}$ whose set of compatible functions $\mathcal{F}(\mathcal{R}_{op})$ corresponds exactly to the set of couples (f_0, f_1) occurring in all complemented circuit implementations of f, w.r.t. the chosen operation *op*, so that an optimal solution of $\mathcal{R}_{op}$ defines an optimal complemented circuit for f.

Let $\mathcal{R}_{op} : \{0,1\}^n \rightarrow \{0,1\}^2$ be a Boolean relation, whose input set is the space spanned by the variables $x_1 \ldots x_n$, while the output set describes all possible couples of functions f_0, f_1 defining a complemented circuit for f, that is, the two outputs of $\mathcal{R}_{op}$ represent the values $y_0 = f_0(x_1, \ldots, x_n)$ and $y_1 = f_1(x_1, \ldots, x_n)$, respectively. We show how to construct the relation $\mathcal{R}_{op}$ for any binary operation *op*. In our analysis, we only consider the ten (out of sixteen) binary operations depending on both input variables. Thus, we will not consider the constant zero and the constant one operations, as well as the four degenerate operations depending on only one input, i.e.,:

1. $x \; op \; y = x$, that defines the complemented circuit $C(f) = \overline{SOP_0}$, represents f iff $f_{off} \subseteq f_0 \subseteq f_{off} \cup f_{dc}$ (i.e., $C(f)$ corresponds to the complement of an SOP for $\overline{f}$);
2. $x \; op \; y = \overline{x}$, that defines the complemented circuit $C(f) = SOP_0$, represents f iff $f_{on} \subseteq f_0 \subseteq f_{on} \cup f_{dc}$ (i.e., $C(f)$ is simply an SOP for f);
3. $x \; op \; y = y$, that defines the complemented circuit $C(f) = SOP_1$, represents f iff $f_{on} \subseteq f_1 \subseteq f_{on} \cup f_{dc}$ (i.e., as before $C(f)$ is simply an SOP for f);
4. $x \; op \; y = \overline{y}$, that defines the complemented circuit $C(f) = \overline{SOP_1}$, represents f iff $f_{off} \subseteq f_1 \subseteq f_{off} \cup f_{dc}$ (i.e., $C(f)$ is the complement of an SOP for $\overline{f}$);

In summary, the complemented circuit corresponds to an SOP for f in the second and third case, and to the complement of an SOP for $\overline{f}$, that is to a POS for $\overline{f}$, in the first and fourth case. Indeed, complementing a sum of products and applying De Morgan's laws, we immediately obtain a product of sums with the same number of literals. We will therefore consider only the following 10 binary operations:

$x\,y$	AND
$\overline{x}\,y$	$\not\Leftarrow$
$x\,\overline{y}$	$\not\Rightarrow$
$\overline{x}\,\overline{y}$	NOR
$x + y$	OR
$\overline{x} + y$	$\Rightarrow$
$x + \overline{y}$	$\Leftarrow$
$\overline{x} + \overline{y}$	NAND
$x \oplus y$	XOR
$\overline{(x \oplus y)}$	XNOR

To construct the relation $\mathcal{R}_{op}$, for any of the selected operations, we must distinguish three cases depending on whether the input vector $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ belongs to the on-set, the off-set, or the dc-set of the function f. We first consider the relations corresponding to the first four operations, which are all based on the binary AND.

Boolean Relations for AND, $\not\Leftarrow$, $\not\Rightarrow$, and NOR

Let us start with the relation $\mathcal{R}_{AND}$ corresponding to the case op = AND. Recall that the complemented circuit is defined as $c = \overline{y}_0 \; op \; y_1$, where $y_0 = f_0(x_1, \ldots, x_n)$ and $y_1 = f_1(x_1, \ldots, x_n)$, and f_0 and f_1 are represented in SOP form. Therefore, for op = AND, we get $c = \overline{y}_0 y_1$, meaning that the complemented circuit c actually represents f as $f_0 \not\Leftarrow f_1$. We can define $\mathcal{R}_{AND}$ as follows

- all points $x \in f_{on}$ must be associated to the output 01, so that the output of the complemented circuit $\overline{y_0} y_1$ evaluates to 1, thus we define $\mathcal{R}_{AND}(x) = 01$;
- all points $x \in f_{off}$ must be associated to one of the three output values on which $\overline{y_0} y_1$ evaluates to 0, thus we define $\mathcal{R}_{AND}(x) = \{00, 10, 11\} = \{1-, -0\}$;
- all points $x \in f_{dc}$ can be associated to any output, thus we have $\mathcal{R}_{AND}(x) = \{--\}$.

The relations $\mathcal{R}_{\not\Leftarrow}$, $\mathcal{R}_{\not\Rightarrow}$, and $\mathcal{R}_{NOR}$, corresponding to the other three operations, can be defined in an analogous way. Observe that, because of the complement on the first input of the binary operations, the complemented circuit c represents f as the AND of f_0 and f_1 when $op = \not\Leftarrow$, as the NOR of f_0 and f_1 when $op = \not\Rightarrow$, and as $f_0 \not\Rightarrow f_1$ when op = NOR. The four Boolean relations $\mathcal{R}_{AND}$, $\mathcal{R}_{\not\Leftarrow}$, $\mathcal{R}_{NOR}$, and $\mathcal{R}_{NOR}$ are summarized in the following table

	$\mathcal{R}_{AND}$	$\mathcal{R}_{\not\Leftarrow}$	$\mathcal{R}_{\not\Rightarrow}$	$\mathcal{R}_{NOR}$
$x \in f_{on}$	$\{01\}$	$\{11\}$	$\{00\}$	$\{10\}$
$x \in f_{off}$	$\{1-, -0\}$	$\{0-, -0\}$	$\{1-, -1\}$	$\{0-, -1\}$
$x \in f_{dc}$	$\{--\}$	$\{--\}$	$\{--\}$	$\{--\}$

Boolean Relations for OR, $\Rightarrow$, $\Leftarrow$, and NAND

We now consider the relations corresponding to the four operations based on the binary OR. The relations $\mathcal{R}_{OR}$ describes the complemented circuit $c = \overline{y}_0 + y_1$ that represents f as $f_0 \Rightarrow f_1$. We can define $\mathcal{R}_{OR}$ as follows

- all points $x \in f_{on}$ must be associated to one of the three output values on which the output $\overline{y_0} + y_1$ of the complemented circuit evaluates to 1, thus $\mathcal{R}_{OR}(x) = \{00, 01, 11\} = \{0-, -1\}$;
- all points $x \in f_{off}$ must be associated to the unique output 10 on which $\overline{y}_0 + y_1$ evaluates to 0, thus we define $\mathcal{R}_{OR}(x) = 10$;
- all points $x \in f_{dc}$ can be associated to any output, and we have $\mathcal{R}_{OR}(x) = \{--\}$.

The relations $\mathcal{R}_{\Rightarrow}$, $\mathcal{R}_{\Leftarrow}$, and $\mathcal{R}_{NAND}$, corresponding to the other three operations, can be defined in an analogous way. As before, note that the complemented circuit c represents f as the OR of f_0 and f_1 when $op = \Rightarrow$, as the NAND of f_0 and f_1 when $op = \Leftarrow$, and as $f_0 \Leftarrow f_1$ when op = NAND. The four Boolean relations $\mathcal{R}_{OR}$, $\mathcal{R}_{\Rightarrow}$, $\mathcal{R}_{\Leftarrow}$, and $\mathcal{R}_{NAND}$ are summarized in the following table

	$\mathcal{R}_{OR}$	$\mathcal{R}_{\Rightarrow}$	$\mathcal{R}_{\Leftarrow}$	$\mathcal{R}_{NAND}$
$x \in f_{on}$	$\{0-, -1\}$	$\{1-, -1\}$	$\{0-, -0\}$	$\{1-, -0\}$
$x \in f_{off}$	$\{10\}$	$\{00\}$	$\{11\}$	$\{01\}$
$x \in f_{dc}$	$\{--\}$	$\{--\}$	$\{--\}$	$\{--\}$

Boolean Relations for XOR and XNOR

We finally consider the relations corresponding to the two binary operations based on the exclusive OR. The relations $\mathcal{R}_{XOR}$ describes the complemented circuit $c = \overline{y}_0 \oplus y_1 = \overline{(y_0 \oplus y_1)}$ that represents f as the XNOR of f_0 and f_1, i.e., as $f_0 \Leftrightarrow f_1$. We define $\mathcal{R}_{XOR}$ as follows

- all points $x \in f_{on}$ must be associated to one of the two outputs on which the output $\overline{y}_0 \oplus y_1$ of the complemented circuit is equal to 1, thus $\mathcal{R}_{XOR}(x) = \{00, 11\}$;
- all points $x \in f_{off}$ must be associated to one of the two outputs on which $\overline{y}_0 \oplus y_1$ evaluates to 0, thus $\mathcal{R}_{XOR}(x) = \{01, 10\}$;
- all points $x \in f_{dc}$ can be associated to any output, thus $\mathcal{R}_{XOR}(x) = \{--\}$.

When op = XNOR, the complemented circuit becomes $c = y_0 \oplus y_1$, i.e., c represents f as the XOR of f_0 and f_1. The corresponding Boolean relation $\mathcal{R}_{XNOR}$ can be immediately defined. These two Boolean relations are summarized in the following table

	$\mathcal{R}_{XOR}$	$\mathcal{R}_{XNOR}$
$x \in f_{on}$	$\{00, 11\}$	$\{01, 10\}$
$x \in f_{off}$	$\{01, 10\}$	$\{00, 11\}$
$x \in f_{dc}$	$\{--\}$	$\{--\}$

4.1 Minimization of complemented circuits

With the formalism of the Boolean relations, we can rephrase our complemented circuit minimization problem as the problem of finding an optimal implementation of $\mathcal{R}_{op}$ (for op corresponding to one of the selected binary operations), that is, of selecting among all possible two-output functions compatible with $\mathcal{R}_{op}$, the one defining the couple (f_0, f_1) leading to a circuit of minimal cost, according to a given cost metric.

Theorem 1 *The set $\mathcal{F}(\mathcal{R}_{op})$ of all two-output functions compatible with the relation $\mathcal{R}_{op}$ specifies exactly the set of all couples (f_0, f_1), occurring in all possible complemented circuit implementations, with the operator op, of f.*

Proof. First of all observe that any complemented circuit $C(f)$, with the operator op, defines a two-output function compatible with R_{op} in an obvious way. The two functions f_0, f_1 represented by the two SOPs in $C(f)$ (in Equation 1) define the two outputs, and we can easily verify that for all $x \in \{0,1\}^n$, $(f_0(x), f_1(x)) \in \mathcal{R}_{op}(x)$ for any operator op.

We now prove that any two-output function compatible with $\mathcal{R}_{op}$ defines a complemented circuit $C(f)$ for f. Observe that, for any operation op, the definition of R_{op} guarantees that the two outputs of each function $g \in \mathcal{F}(\mathcal{R}_{op})$ combined with the chosen operation op, evaluate to 1 on all points in the on-set of f and to 0 on all points in the off-set of f. Thus, if we take the first output of g as f_0 and the second output as f_1, and we represent them in SOP forms, we get a complemented circuit for f. ■

We finally prove that an optimal solution of $\mathcal{R}_{op}$ provides an optimal complemented circuit $C(f)$ for f.

Corollary 1 *An optimal solution of the Boolean relation $\mathcal{R}_{op}$, according to a given cost function μ chosen to evaluate the complemented circuits, defines an optimal complemented circuit for f w.r.t. the same cost function μ.*

Proof. The thesis immediately follows from Theorem 1, as any two-output function (f_0, f_1) compatible with $\mathcal{R}_{op}$ defines a possible complemented circuit implementation for f. Indeed, we have that $C(f) = \overline{SOP_0}\ op\ SOP_1$, and its cost, under any given cost metric μ, is determined by the cost under μ of the SOP representations of f_0, and f_1. ■

5 Structure of Complemented Circuits

Let us now examine more deeply the structure of the complemented circuits that we have defined. Let us first consider the four circuits based on the AND operation. For op = AND, the complemented circuit representation of f has the structure of a conjunction of a POS and an SOP, indeed we have

$$C(f) = \overline{SOP_0} \wedge SOP_1 = POS_0 \wedge SOP_1\,,$$

since the complement of a sum of products is a product of sums. The corresponding minimization problem is therefore that of finding the couple of function (f_0, f_1) such that the conjunction of a POS for f_0 and an SOP for f_1 gives a circuit representation of f of minimal cost, according to the given cost metric. For $op = \not\Leftarrow$, we have

$$C(f) = SOP_0 \wedge SOP_1\,,$$

i.e., the complemented circuit corresponds to an AND-OR-AND three-level network (Malik, [15] and Dubrova, [11]). When $op = \not\Rightarrow$, $C(f)$ simply consists in a POS representation of the function f, indeed

$$C(f) = \overline{SOP_0} \wedge \overline{SOP_1} = POS_0 \wedge POS_1 = POS(f)\,,$$

where the last equality follows from the fact the the last two levels composed of three AND gates can be merged into a single AND gate. Thus, the complemented circuit minimization problem in this case is equivalent to the synthesis of a function as a product of sums. Finally, when $op =$ NOR, the complemented circuit representation of f consists in the conjunction of an SOP and a POS:

$$C(f) = SOP_0 \wedge \overline{SOP_1} = SOP_0 \wedge POS_1\,.$$

This minimization problem is then equivalent to the one corresponding to $op =$ AND, indeed an optimal solution (f_0, f_1) for the first problem ($op =$ AND) immediately becomes an optimal solution for the second problem ($op =$ NOR) by simply exchanging the two functions f_0 and f_1, or, equivalently, exchanging the two outputs of an optimal implementation of $\mathcal{R}_{AND}$.

Let us now examine the four circuits based on the binary OR operation. For $op =$ OR, the complemented circuit representation of f has the structure of a disjunction of a POS and an SOP:

$$C(f) = \overline{SOP_0} \vee SOP_1 = POS_0 \vee SOP_1\,.$$

For $op = \Rightarrow$, $C(f)$ is simply an SOP representation of the function f, indeed

$$C(f) = SOP_0 \vee SOP_1 = SOP(f)\,,$$

and the minimization problem is equivalent to the synthesis of a function as a sum of products When $op = \Leftarrow$, we have

$$C(f) = \overline{SOP_0} \vee \overline{SOP_1} = POS_0 \vee POS_1\,,$$

i.e., the complemented circuit corresponds to an OR-AND-OR three-level network (Sasao, [20], and Debnath, [10]). Finally, when $op =$ NAND, the complemented circuit becomes a disjunction of an SOP and a POS

$$C(f) = SOP_0 \vee \overline{SOP_1} = SOP_0 \vee POS_1\,.$$

Af before, this minimization problem is equivalent to the one corresponding to $op =$ OR, and an optimal solution (f_0, f_1) for the first problem ($op =$ OR) immediately becomes an optimal solution for the second problem ($op =$ NAND) by simply exchanging the two functions f_0 and f_1, or, equivalently, exchanging the two outputs of an optimal implementation of $\mathcal{R}_{OR}$.

Let us finally consider the two operations based on the exclusive OR. For $op =$ XOR, we have

$$C(f) = \overline{SOP_0} \oplus SOP_1 = POS_0 \oplus SOP_1\,,$$

and the associated minimization problem is that of finding a minimal cost representation of f as the exclusive OR between a POS and a SOP, while for $op =$ XNOR, the complemented circuit becomes an AND-OR-EXOR network (Debnath, [8, 9] and Dubrova, [12]):

$$C(f) = SOP_0 \oplus SOP_1\,.$$

In summary, three (out of ten) complemented circuits describe new three-level logic implementations for Boolean functions, i.e., the three circuits described by the Boolean relations $\mathcal{R}_{AND}$, $\mathcal{R}_{OR}$, and $\mathcal{R}_{XOR}$.

5.1 Multioutput functions

We now show how to model with a single Boolean relation with input set $\{0,1\}^n$ and output set $\{0,1,-\}^{2m}$ the flexibiliy of a multioutput function $f : \{0,1\}^n \to \{0,1,-\}^m$. In the following table we provide the four Boolean relations $\mathcal{R}_{OR}$ $\mathcal{R}_{\Rightarrow}$, $\mathcal{R}_{\Leftarrow}$, and $\mathcal{R}_{NAND}$ for functions with $m = 2$ outputs.

	$\mathcal{R}_{OR}$	$\mathcal{R}_{\Rightarrow}$	$\mathcal{R}_{\Leftarrow}$	$\mathcal{R}_{NAND}$
$x \in f^1_{on}, f^2_{on}$	$\{0-,-1\} \times \{0-,-1\}$	$\{1-,-1\} \times \{1-,-1\}$	$\{0-,-0\} \times \{0-,-0\}$	$\{1-,-0\} \times \{1-,-0\}$
$x \in f^1_{on}, f^2_{off}$	$\{0-,-1\} \times \{10\}$	$\{1-,-1\} \times \{00\}$	$\{0-,-0\} \times \{11\}$	$\{1-,-0\} \times \{01\}$
$x \in f^1_{on}, f^2_{dc}$	$\{0-,-1\} \times \{--\}$	$\{1-,-1\} \times \{--\}$	$\{0-,-0\} \times \{--\}$	$\{1-,-0\} \times \{--\}$
$x \in f^1_{off}, f^2_{on}$	$\{10\} \times \{0-,-1\}$	$\{00\} \times \{1-,-1\}$	$\{11\} \times \{0-,-0\}$	$\{01\} \times \{1-,-0\}$
$x \in f^1_{off}, f^2_{off}$	$\{10\} \times \{10\}$	$\{00\} \times \{00\}$	$\{11\} \times \{11\}$	$\{01\} \times \{01\}$
$x \in f^1_{off}, f^2_{dc}$	$\{10\} \times \{--\}$	$\{00\} \times \{--\}$	$\{11\} \times \{--\}$	$\{01\} \times \{--\}$
$x \in f^1_{dc}, f^2_{on}$	$\{--\} \times \{0-,-1\}$	$\{--\} \times \{1-,-1\}$	$\{--\} \times \{0-,-0\}$	$\{--\} \times \{1-,-0\}$
$x \in f^1_{dc}, f^2_{off}$	$\{--\} \times \{10\}$	$\{--\} \times \{00\}$	$\{--\} \times \{11\}$	$\{--\} \times \{01\}$
$x \in f^1_{dc}, f^2_{dc}$	$\{--\} \times \{--\}$	$\{--\} \times \{--\}$	$\{--\} \times \{--\}$	$\{--\} \times \{--\}$

The problem with this approach is the size of the resulting Boolean relation and the time taken to construct it. Therefore in the experiments described in Section 6 we set up and minimized independently a Boolean relation for each output, trading off quality of results vs. computation time. Further investigation is needed to come up with a better compromise.

6 Experimental results

In this section we report the experimental results for the minimization of complemented circuits based on Boolean relations.

The algorithms have been implemented in C, using the CUDD library for OBDDs to represent Boolean functions, and BREL [1] for the synthesis of Boolean relations since it finds better solutions in shorter runtime than the previously known methods. The experiments have been run on a Linux Intel Core i7, 2.67 GHz CPU with 4 GB of main memory. Multioutput benchmarks have been synthesized minimizing each single output independently from the others. To show the performance in area of complemented circuits derived using Boolean relations, we compare our results, obtained running the proposed algorithms on a testing set of 131 benchmarks taken from LGSynth93 [23], vs. SOP forms synthesized using ESPRESSO. Both BREL and ESPRESSO can be run in exact and heuristic mode. To evaluate the resulting circuits, we synthesized them with the SIS system using the MCNC library for technology mapping and the SIS command `map -W -f 3 -s`.

In Table 1 we compare synthesis time (in seconds), mapped area and delay of the three complemented circuits and SOPs for a significant subset of the benchmarks. The first column reports the names of the benchmarks. The next columns report, by groups of three, the synthesis times in seconds, the areas and delays estimated by SIS. The first two groups, labeled SOP exact and SOP heuristic, refer to plain SOPs. The next group refers to complemented circuits synthesized with different *op* gates (AND, OR, XOR).

In Table 2 we compare our results to the approaches presented in [11] and in [12]. In [11] the authors present a three-level logic optimization based on a novel strategy for pairing cubes. The experiments were on a Sun Ultra 60 operating with two 360 MHz CPU and with 1024 MB RAM main storage. In [12] the authors present a three-level heuristic AND-OR-XOR minimization strategy for incompletely specified functions (ISFs). They ran their experiments on a Sun SPARC 20 operating at 50 MHz with 64 MB RAM of main storage. The first and the second column of the table report the names of the benchmarks and the number of inputs/outputs, respectively. In the following six columns, for each case (AND, OR, XOR) we report both execution time (in seconds) and number of literals of the minimized complemented circuits considering two different strategies: the first one is an exact synthesis strategy that minimizes the number of literals, while the second one is an heuristic approach. Finally, the last columns report results presented in [11] (columns 15 and 16) and in [12] (columns 17 and 18). The results show that complemented circuit synthesized with Boolean relations turned out to be more compact than the corresponding circuits proposed in [11] and in [12] in 29% of our experiments.

Table 3 summarizes the comparison between complemented circuits vs. SOP forms for both the exact and heuristic mode for all ten binary operations. The first column of the table reports the considered operator. The following columns report, by groups of three, the percentages w.r.t. time, area, and delay, by comparing the results of our approach with BREL in exact mode with the results of ESPRESSO in exact mode (groups 1 and 3) and by comparing the results of our approach with BREL in heuristic mode with the results of ESPRESSO in heuristic mode (groups 2 and 4). The values of the first two groups (GAIN) are calculated as percentages of number of cases (over the entire testing set of 131 benchmark) where we obtain better results

Table 1: Comparison of SOP vs complemented circuits

	SOP exact			SOP heuristic			AND exact			AND heuristic			OR exact			OR heuristic			XOR exact			XOR heuristic		
benchmark	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**
add6	0.77	292	25.20	0.03	292	25.20	7.60	297	25.30	0.08	290	25.40	6.97	294	22.70	0.09	287	24.10	5.32	292	26.60	0.07	327	28.10
addm4	0.04	934	42.70	0.02	959	41.90	7.31	781	30.10	0.02	918	38.50	7.51	807	36.90	0.02	899	39.60	8.10	745	31.00	0.02	1016	44.80
al2	0.86	340	15.10	0.00	340	15.10	3.76	332	14.00	0.01	335	14.20	0.58	331	13.10	0.00	443	13.80	4.87	505	16.10	0.01	425	15.10
alu2	0.01	176	16.40	0.00	176	14.50	1.71	160	12.30	0.03	203	20.50	1.68	158	14.00	0.01	195	16.50	1.81	155	17.80	0.02	187	19.90
alu3	0.01	187	16.80	0.00	157	13.30	1.66	158	12.60	0.03	199	20.80	1.69	157	13.30	0.02	199	17.00	1.75	159	17.80	0.01	179	19.10
amd	0.02	982	38.50	0.01	986	37.30	7.38	736	37.00	0.09	815	34.20	7.00	752	38.60	0.06	852	34.50	7.50	818	35.10	0.06	892	35.80
b2	0.08	3984	76.90	0.01	3957	73.00	168.31	3219	69.10	1.32	3597	69.70	170.34	3207	66.30	1.10	3492	68.70	151.47	3156	65.20	1.60	3567	76.30
b3	0.81	1126	45.00	0.03	1095	44.70	281.90	877	34.60	0.51	1250	36.20	279.14	853	35.80	0.47	1337	39.20	192.95	903	36.80	0.45	1048	34.70
b4	1.17	649	31.20	0.00	649	31.20	7.39	596	26.60	0.75	657	28.00	7.19	594	26.60	0.69	731	30.40	8.18	616	28.80	0.62	697	29.70
bcc	0.03	5167	92.10	0.03	5150	93.30	66.16	3850	66.80	1.13	5334	94.60	84.34	3799	64.30	0.72	5195	93.80	66.51	3985	67.40	0.84	5224	92.90
bench	0.01	172	15.00	0.00	144	12.10	0.97	98	12.10	0.00	120	14.00	0.62	98	11.30	0.00	110	11.70	1.03	116	15.00	0.00	141	15.10
br2	0.00	280	28.30	0.00	280	28.30	1.81	237	18.40	0.00	265	19.00	1.28	234	18.10	0.00	261	25.10	2.04	256	21.70	0.00	268	24.90
dc1	0.00	85	11.80	0.00	85	12.10	0.69	72	10.30	0.00	77	13.40	0.41	74	10.80	0.00	79	13.80	0.74	80	11.60	0.00	109	15.40
dc2	0.00	243	24.70	0.00	233	20.40	1.46	184	20.30	0.00	210	18.30	1.59	185	19.40	0.00	199	18.20	1.42	197	20.50	0.00	220	21.10
ex7	0.14	225	21.20	0.00	225	21.20	3.06	194	24.90	0.04	231	20.10	2.78	193	19.70	0.04	238	20.60	3.37	219	23.00	0.04	251	24.00
exep	0.04	1285	31.40	0.01	1275	36.10	9.18	1290	31.00	0.46	1295	33.50	1.86	1271	31.20	0.10	1300	31.40	11.08	1454	36.60	0.56	1311	36.90
exps	0.06	3579	94.50	0.02	3549	101.10	17.69	2405	62.50	0.01	2697	73.50	17.98	2387	64.30	0.01	2574	71.90	16.87	2429	65.10	0.00	2713	68.20
fout	0.05	535	30.40	0.00	472	27.10	2.82	339	23.10	0.00	398	25.00	2.80	336	21.90	0.00	418	26.20	2.40	325	23.10	0.00	402	28.00
in1	0.07	3984	76.90	0.01	3957	73.00	167.55	3219	69.10	1.28	3597	69.70	170.23	3207	66.30	1.11	3492	68.70	151.62	3156	65.20	1.56	3567	76.30
in3	0.12	1111	34.10	0.00	1111	34.10	8.71	928	32.80	0.16	985	32.80	7.79	913	32.90	0.22	970	33.60	8.95	971	36.20	0.08	974	37.80
in4	0.65	1127	44.90	0.02	1077	44.80	286.03	924	35.60	0.59	1297	36.60	280.34	900	36.80	0.51	1384	39.30	194.52	950	37.80	0.46	1114	35.60
in6	0.77	662	31.30	0.00	662	31.30	7.06	609	26.70	0.23	669	28.30	6.64	607	26.70	0.14	744	30.50	7.67	629	28.80	0.08	710	29.80
in7	0.25	319	23.60	0.00	319	23.60	4.80	305	23.30	0.13	340	26.90	4.81	304	23.10	0.09	325	28.40	4.99	338	27.10	0.09	379	33.50
luc	0.00	821	40.70	0.00	845	40.20	5.40	596	28.00	0.01	649	31.80	4.66	599	27.30	0.01	641	28.20	4.71	620	28.40	0.00	727	34.00
m181	0.24	234	20.50	0.00	174	16.40	0.92	114	11.10	0.00	141	18.10	1.08	113	12.00	0.00	127	11.50	1.34	134	14.80	0.00	169	19.30
m3	0.00	1169	47.10	0.00	1269	51.50	3.96	502	29.90	0.00	547	28.00	3.75	492	30.60	0.01	536	30.90	3.67	535	33.60	0.00	531	31.30
m4	0.04	1904	66.50	0.02	1778	54.20	5.91	799	35.20	0.00	863	36.50	5.80	807	34.80	0.02	867	36.90	5.64	812	38.50	0.01	901	38.70
mlp4	0.10	702	34.50	0.01	686	36.40	4.69	569	30.50	0.01	629	30.10	4.59	579	31.10	0.00	595	29.30	4.64	573	35.10	0.01	630	36.90
mp2d	0.08	268	20.60	0.00	251	19.80	1.72	224	17.00	0.00	269	20.30	1.36	222	19.00	0.00	233	17.20	2.17	239	17.10	0.00	238	15.20
p1	0.03	651	32.20	0.02	645	36.00	3.63	382	24.60	0.00	466	28.50	3.71	375	21.20	0.00	441	27.80	3.74	396	24.40	0.02	478	26.80
p3	0.02	447	26.70	0.00	448	26.60	2.35	245	21.50	0.00	293	23.90	2.32	248	18.70	0.01	285	19.10	2.40	258	21.50	0.00	308	23.50
shift	801.94	372	27.00	0.00	372	27.00	3.64	371	27.20	0.03	372	27.40	3.45	372	27.00	0.02	372	27.00	4.17	377	27.20	0.01	372	27.40
spla	0.80	2422	57.40	0.19	2470	68.60	14.56	1678	44.70	0.05	1859	48.40	14.38	1688	46.30	0.09	1862	47.40	16.68	1763	40.60	0.04	1932	49.00
sym10	1.23	519	29.90	0.02	592	31.70	59.06	344	35.70	0.00	379	31.50	54.97	301	32.40	0.01	350	27.50	32.66	301	30.40	0.01	379	31.50
t1	6.83	546	22.10	0.01	458	21.90	3.66	347	18.00	0.00	378	19.30	3.46	353	17.90	0.03	388	20.50	4.49	374	19.30	0.00	412	22.90
t4	0.01	102	14.10	0.00	96	11.60	0.91	96	14.90	0.00	97	14.80	0.63	93	13.00	0.00	100	9.40	0.97	114	11.90	0.00	116	19.30
test1	39.11	1392	47.50	0.02	1318	46.60	7.95	935	39.40	0.10	1351	44.90	8.37	935	35.30	0.07	1369	47.00	8.00	944	39.00	0.10	1302	46.30
tial	1.31	2376	68.70	0.20	2327	66.00	123.41	1516	51.60	0.51	1654	50.40	104.32	1489	50.80	0.31	1715	49.30	142.05	1537	53.10	0.64	2428	60.00
vg2	0.07	341	18.60	0.00	341	18.60	3.80	287	22.00	0.10	495	21.70	4.24	284	20.00	0.09	369	18.60	5.16	292	22.00	0.06	376	18.60
vtx1	0.05	324	21.30	0.00	324	21.30	3.19	238	17.90	0.07	424	28.30	3.46	259	20.50	0.10	383	25.30	4.13	257	19.70	0.05	429	27.20
x1dn	0.58	125	13.30	0.01	125	13.30	2.27	125	13.30	0.11	141	13.80	2.58	125	13.30	0.10	141	13.80	5.83	125	13.30	0.19	141	13.80
x6dn	0.04	789	34.40	0.00	762	31.20	6.91	725	36.40	0.13	772	32.20	6.91	738	32.50	0.13	770	33.50	6.75	734	32.90	0.08	777	34.80
x9dn	0.06	384	23.00	0.00	384	23.00	3.33	262	22.70	0.08	530	28.00	4.90	254	20.90	0.12	464	24.60	4.68	258	19.40	0.09	590	32.70

Table 2: Minimization results

		AND exact		AND heur.		OR exact		OR heur.		XOR exact		XOR heur.		[11]		[12]	
benchmark	**in/out**	**time**	**lit**	**time**	**lit**	**time**	**lit**	**time**	**lit**	**time**	**lit**	**time**	**lit**	**time**	**lit**	**time**	**lit**
5xp1	7/10	0.00	**17**	0.00	**17**	0.00	20	0.00	20	0.00	20	0.00	20	14.00	55	14.80	42
9sym	9/1	7.79	73	0.00	73	7.21	**71**	0.00	72	7.40	72	0.00	72			1.70	73
alu2	10/8	1.71	67	0.03	79	1.68	68	0.01	83	1.81	**48**	0.02	64	32.00	52		
alu3	10/8	1.66	67	0.03	79	1.69	70	0.02	86	1.75	49	0.01	64	24.00	**47**		
alu4	14/8	107.70	518	0.41	631	101.58	520	0.27	632	87.42	472	0.53	658			131.90	**447**
b12	15/9	0.82	42	0.01	51	0.96	40	0.00	47	1.28	42	0.00	58	30.00	**28**	9.10	31
bw	5/28	0.00	56	0.00	56	0.00	56	0.00	56	0.00	56	0.00	56			25.40	**24**
clip	9/5	3.60	149	0.01	158	3.55	150	0.00	159	3.71	116	0.03	162			10.50	**95**
con1	7/2	0.18	10	0.00	10	0.25	10	0.00	11	0.29	10	0.00	13			1.00	**9**
cordic	23/2	1909.46	345	0.15	1180	1908.45	243	0.14	2058	1024.40	158	0.15	1187			231.80	**156**
dist	8/5	5.12	153	0.01	173	5.08	153	0.01	179	4.64	146	0.01	149	170.00	**108**		
ex1010	10/10	35.51	639	0.26	1118	35.47	630	0.27	1124	34.99	**608**	0.49	1051			109.70	725
inc	7/9	1.26	58	0.00	64	1.25	57	0.00	59	1.44	55	0.00	59			6.50	**33**
misex1	8/7	1.04	43	0.00	48	0.89	43	0.00	44	0.97	40	0.00	46			11.50	**13**
misex2	25/18	1.67	76	0.00	133	0.32	104	0.00	97	2.29	116	0.00	121			6.00	**28**
misex3	14/14	106.90	585	0.44	1051	116.77	576	0.29	961	84.78	672	0.49	1270			112.00	**191**
newapla2	6/7	0.52	30	0.00	41	0.00	42	0.00	42	0.63	42	0.00	34	0.39	**5**		
newbyte	5/8	0.59	25	0.00	35	0.00	40	0.00	40	0.71	40	0.00	37	0.52	**5**		
newcpla1	9/16	1.85	81	0.00	94	1.41	81	0.01	95	2.30	99	0.00	92	26.00	**27**		
newtpla	15/5	0.99	33	0.01	50	0.74	36	0.00	54	1.04	40	0.00	38	1.20	**19**		
radd	8/5	1.54	76	0.01	86	1.43	77	0.00	75	1.18	49	0.00	52	14.00	**39**		
rd53	5/3	0.00	11	0.00	11	0.09	**6**	0.00	8	0.17	8	0.00	7	5.90	26	15.70	19
rd73	7/3	0.00	17	0.00	17	0.16	**6**	0.00	9	0.24	8	0.00	9	312.00	79	25.50	83
rd84	8/4	14.74	309	0.00	299	14.16	310	0.00	303	10.53	194	0.00	247			61.10	**192**
ryy6	16/1	0.52	8	0.01	8	0.74	**7**	0.01	**7**	0.68	**7**	0.00	113	379.00	**7**		
sao2	10/4	0.04	30	0.00	12	0.46	13	0.00	14	0.55	**8**	0.00	**8**			3.70	38
sqn	7/3	1.18	47	0.00	51	1.23	41	0.00	48	1.25	43	0.00	53	6.10	**34**		
squar5	5/8	0.69	36	0.00	37	0.61	39	0.00	39	0.85	38	0.00	43			4.40	**22**
t2	17/16	2.69	94	0.02	126	2.42	124	0.01	123	3.06	110	0.00	116	44.00	**46**		
t481	16/1	14.69	361	0.01	361	12.24	360	0.00	360	43.92	441	0.01	482			557.20	**113**
table3	14/14	62.89	751	0.58	1385	59.19	861	0.31	1094	57.80	838	0.46	1140			79.30	**176**
table5	17/15	68.56	862	0.48	1427	72.28	977	0.35	797	67.98	800	0.41	1178			100.70	**158**
x1dn	27/6	2.27	78	0.11	83	2.58	**76**	0.10	81	5.83	76	0.19	81	237.00	81		
x9dn	27/7	3.33	111	0.08	244	4.90	103	0.12	158	4.68	110	0.09	297	307.00	**81**		
xor5	5/1	0.29	16	0.00	16	0.30	16	0.00	16	0.18	**9**	0.00	**9**			10.30	10
vg2	25/8	0.61	22	0.13	42	0.63	**18**	0.15	22	0.71	**18**	0.21	21	366.00	88	43.40	102
z4	7/4	1.09	55	0.00	61	1.12	62	0.00	60	0.95	41	0.00	41	2.10	**33**		
Z5xp1	7/10	1.69	86	0.00	82	1.76	79	0.00	83	1.53	74	0.00	80	112.00	**56**		

Table 3: Gains of complemented circuits w.r.t. corresponding SOP forms

	GAIN						AVERAGE GAIN					
	EXACT			HEURISTIC			EXACT			HEURISTIC		
	time	area	delay	time	area	delay	time	area	delay	time	area	delay
AND	2%	89%	63%	2%	65%	63%	-86%	29%	19%	-1077%	16%	12%
$\not\Leftarrow$	2%	89%	63%	2%	65%	63%	-86%	29%	19%	-1077%	16%	12%
$\not\Rightarrow$	2%	88%	63%	2%	66%	64%	-168%	31%	20%	-1853%	17%	13%
NOR	2%	88%	64%	2%	67%	50%	-210%	28%	18%	-1854%	15%	-8%
OR	4%	87%	69%	4%	63%	69%	-83%	29%	20%	-853%	16%	15%
$\Rightarrow$	4%	87%	69%	4%	63%	69%	-83%	29%	20%	-853%	16%	15%
$\Leftarrow$	3%	86%	71%	3%	69%	69%	-164%	31%	21%	-1764%	19%	16%
NAND	3%	88%	70%	3%	63%	66%	-227%	29%	20%	-1833%	14%	13%
XOR	2%	78%	50%	2%	47%	43%	-63%	26%	15%	-1030%	13%	6%
XNOR	2%	78%	50%	2%	47%	43%	-63%	26%	15%	-1030%	13%	6%

Table 4: Time, area and delay: complemented circuits vs ESPRESSO with output phase assignment.

	ESPRESSO - exact			ESPRESSO - heuristic			AND exact			AND heuristic			OR exact			OR heuristic			XOR exact			XOR heuristic		
benchmark	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**	**time**	**area**	**delay**
add6	0.96	292	25.20	0.33	288	23.50	7.60	297	25.30	0.08	290	25.40	6.97	294	22.70	0.09	287	24.10	5.32	292	26.60	0.07	327	28.10
al2	1.09	340	15.10	0.05	370	15.20	3.76	332	14.00	0.01	335	14.20	0.58	331	13.10	0.00	443	13.80	4.87	505	16.10	0.01	425	15.10
alcom	0.38	222	12.30	0.00	231	12.30	2.08	210	12.20	0.02	210	11.20	0.21	210	11.20	0.01	216	11.10	3.25	345	16.50	0.01	304	15.40
alu1	0.03	53	6.80	0.00	55	6.80	0.21	53	6.80	0.00	53	6.80	0.06	53	6.80	0.00	53	6.80	0.62	59	7.00	0.00	85	12.00
b10	0.10	876	42.80	0.09	881	45.20	10.40	721	31.90	0.13	827	34.80	11.47	723	32.00	0.11	857	37.00	10.00	745	37.00	0.15	909	37.30
b3	1.12	1126	45.00	1.03	1095	44.70	281.90	877	34.60	0.51	1250	36.20	279.14	853	35.80	0.47	1337	39.20	192.95	903	36.80	0.45	1048	34.70
b4	1.62	649	31.20	1.64	649	31.20	7.39	596	26.60	0.75	657	28.00	7.19	594	26.60	0.69	731	30.40	8.18	616	28.80	0.62	697	29.70
b7	0.00	234	16.30	0.01	219	15.90	1.93	195	15.00	0.01	212	16.60	0.22	196	15.30	0.00	203	16.10	2.72	294	18.10	0.00	236	17.10
b9	0.20	225	21.20	0.04	225	21.20	3.14	194	24.90	0.04	231	20.10	2.80	193	19.70	0.03	238	20.60	3.36	219	23.00	0.05	251	24.00
bench	0.01	172	15.00	0.00	218	20.50	0.97	98	12.10	0.00	120	14.00	0.62	98	11.30	0.00	110	11.70	1.03	116	15.00	0.00	141	15.10
dist	0.01	735	36.70	0.05	790	40.40	5.12	596	33.40	0.01	646	35.20	5.08	611	32.50	0.01	625	29.90	4.64	543	33.70	0.01	600	37.90
fout	0.06	535	30.40	0.02	519	29.40	2.82	339	23.10	0.00	398	25.00	2.80	336	21.90	0.00	418	26.20	2.40	325	23.10	0.00	402	28.00
gary	0.03	1005	47.80	0.08	1030	53.90	12.39	835	36.60	0.15	979	41.40	13.37	841	38.30	0.12	988	39.40	11.99	864	38.40	0.16	1161	46.50
in0	0.03	1005	47.80	0.07	1032	54.20	12.44	835	36.60	0.17	979	41.40	13.36	841	38.30	0.13	988	39.40	12.03	864	38.40	0.14	1161	46.50
in1	0.09	3984	76.90	0.16	3957	73.00	167.55	3219	69.10	1.28	3597	69.70	170.23	3207	66.30	1.11	3492	68.70	151.62	3156	65.20	1.56	3567	76.30
in2	0.02	939	36.70	0.12	997	40.80	88.78	831	34.30	0.45	953	37.60	104.00	820	34.90	0.38	1019	37.50	86.56	847	40.50	0.38	982	37.10
in3	0.16	1111	34.10	0.28	1095	34.60	8.71	928	32.80	0.16	985	32.80	7.79	913	32.90	0.22	970	33.60	8.95	971	36.20	0.08	974	37.80
in4	0.84	1127	44.90	1.04	1128	44.00	286.03	924	35.60	0.59	1297	36.60	280.34	900	36.80	0.51	1384	39.30	194.52	950	37.80	0.46	1114	35.60
in5	0.06	865	38.50	0.28	865	38.50	8.47	686	31.90	0.15	937	34.90	8.47	696	31.40	0.18	831	31.70	11.35	720	31.60	0.21	954	40.70
in6	0.99	662	31.30	0.48	662	31.30	7.06	609	26.70	0.23	669	28.30	6.64	607	26.70	0.14	744	30.50	7.67	629	28.80	0.08	710	29.80
in7	0.32	319	23.60	0.03	321	23.90	4.80	305	23.30	0.13	340	26.90	4.81	304	23.10	0.09	325	28.40	4.99	338	27.10	0.09	379	33.50
lin	3.06	2849	89.40	0.24	3289	98.50	15.26	1755	53.90	0.02	1909	58.50	14.11	1757	52.30	0.01	1899	54.80	13.47	1645	54.30	0.01	1921	54.40
m181	0.30	234	20.50	0.02	132	12.90	0.92	114	11.10	0.00	141	18.10	1.08	113	12.00	0.00	127	11.50	1.34	134	14.80	0.00	169	19.30
m4	0.04	1904	66.50	0.09	1166	44.80	5.91	799	35.20	0.00	863	36.50	5.80	807	34.80	0.02	867	36.90	5.64	812	38.50	0.01	901	38.70
max128	0.03	2149	67.70	0.07	1588	55.80	5.08	632	30.90	0.01	701	34.50	4.89	624	31.40	0.00	698	33.00	4.71	625	30.50	0.00	683	38.00
max46	0.00	270	26.30	0.00	270	26.30	3.74	269	28.00	0.01	348	33.50	3.92	269	28.00	0.01	348	33.50	2.98	269	28.00	0.01	286	33.70
max512	0.04	919	41.00	0.11	793	38.80	6.74	664	36.00	0.02	780	44.50	6.59	669	35.20	0.02	772	43.10	6.00	659	33.20	0.01	753	43.50
mlp4	0.12	702	34.50	0.08	709	35.70	4.69	569	30.50	0.01	629	30.10	4.59	579	31.10	0.00	595	29.30	4.64	573	35.10	0.01	630	36.90
mp2d	0.10	268	20.60	0.04	227	17.40	1.72	224	17.00	0.00	269	20.30	1.36	222	19.00	0.00	233	17.20	2.17	239	17.10	0.00	238	15.20
spla	0.96	2422	57.40	0.96	2	0.00	14.56	1678	44.70	0.05	1859	48.40	14.38	1688	46.30	0.09	1862	47.40	16.68	1763	40.60	0.04	1932	49.00
sym10	1.66	519	29.90	0.06	592	31.70	59.06	344	35.70	0.00	379	31.50	54.97	301	32.40	0.01	350	27.50	32.66	301	30.40	0.01	379	31.50
t1	8.86	546	22.10	0.24	487	21.70	3.66	347	18.00	0.00	378	19.30	3.46	353	17.90	0.03	388	20.50	4.49	374	19.30	0.00	412	22.90
t2	0.00	352	24.10	0.08	351	26.90	2.69	281	21.00	0.02	300	20.40	2.42	283	20.00	0.01	300	20.70	3.06	314	19.00	0.00	322	20.40
t4	0.01	102	14.10	0.03	92	12.20	0.91	96	14.90	0.00	97	14.80	0.63	93	13.00	0.00	100	9.40	0.97	114	11.90	0.00	116	19.30
test1	50.63	1392	47.50	0.18	1474	49.70	7.95	935	39.40	0.10	1351	44.90	8.37	935	35.30	0.07	1369	47.00	8.00	944	39.00	0.10	1302	46.30
tial	1.68	2376	68.70	1.01	1928	49.60	123.41	1516	51.60	0.51	1654	50.40	104.32	1489	50.80	0.31	1715	49.30	142.05	1537	53.10	0.64	2428	60.00
vg2	0.14	46	10.70	0.01	46	10.70	3.80	287	22.00	0.10	495	21.70	4.24	284	20.00	0.09	369	18.60	5.16	292	22.00	0.06	376	18.60
vtx1	0.08	324	21.30	0.43	324	21.30	3.19	238	17.90	0.07	424	28.30	3.46	259	20.50	0.10	383	25.30	4.13	257	19.70	0.05	429	27.20
x1dn	0.78	125	13.30	0.14	125	13.30	2.27	125	13.30	0.11	141	13.80	2.58	125	13.30	0.10	141	13.80	5.83	125	13.30	0.19	141	13.80
x6dn	0.06	789	34.40	0.12	762	31.20	6.91	725	36.40	0.13	772	32.20	6.91	738	32.50	0.13	770	33.50	6.75	734	32.90	0.08	777	34.80
x9dn	0.08	384	23.00	0.64	545	37.70	3.33	262	22.70	0.08	530	28.00	4.90	254	20.90	0.12	464	24.60	4.68	258	19.40	0.09	590	32.70
Z9sym	0.30	390	25.70	0.02	366	29.40	8.18	187	30.40	0.01	120	17.90	7.75	236	29.20	0.00	120	17.90	8.32	187	30.40	0.00	120	17.90

w.r.t. ESPRESSO. The values of the last two groups (AVERAGE GAIN) report the percentages of average gains of complemented circuits with respect to the corresponding SOP forms for both the exact and heuristic mode.

Moreover, we compare simulation time, area, delay and number of literals of complemented circuits with the results of ESPRESSO running with output phase assignment, that is ESPRESSO chooses the best realization between each output and its complementation. After choosing an assignment of phase for each output, the function is minimized (in heuristic or exact mode). First of all, let us observe that in about 15% of tested benchmarks, we stopped the simulation of ESPRESSO with output phase assignment (after about 30 minutes), without obtaining minimization results. The results of this comparison are summerized in Table 4, where we compare synthesis time (in seconds), mapped area and delay of circuits synthesized with ESPRESSO after phase optimization (ESPRESSO command `-Dopo`) and complemented circuits. The first column reports the name of the benchmarks. The following ones report, by groups of three, the synthesis times in seconds, the areas and delays estimated by SIS. The first two groups refer to ESPRESSO synthesis. The next group refers to complemented circuits synthesized with different op gates (AND, OR, XOR). In the exact case, the percentages of complemented circuits with lower simulation time, area and delay w.r.t. the corresponding circuits minimized with ESPRESSO are 5%, 94%, and 89%, respectively. In the heuristic case, the percentages of complemented circuits with lower time, area and delay w.r.t. the corresponding circuits minimized with ESPRESSO are 87%, 75%, and 77%, respectively.

The conclusions are that:

1. a high percentage of complemented circuits synthesized with Boolean relations has a smaller area and delay than the corresponding SOP forms (this is true for both the exact and heuristic mode);

2. the synthesis of complemented circuits allows to obtain gains for both area and delay, at the expense of higher computation time;

3. complemented circuits synthesized with XOR and XNOR *op* gates exhibit a lower gain w.r.t. the corresponding benchmarks synthesized with other *op* gates (on average, we obtain higher gains in the exact case);

4. in the case of the operator $\Rightarrow$, it is true that our approach reduces to minimizing SOPs, as ESPRESSO does, however the results are different because the cost function is not the cardinality of SOPs, but the area of the circuit mapped with SIS (as in all the other experiments).

In conclusion we investigated a three-level form, called *complemented circuit*, which is a special type of decomposition of a Boolean function f as $\overline{f}_0$ *op* f_1, where *op* is a two-level Boolean connective. Then we characterized completely all the correct implementations of such a form, as the logic functions compatible with a Boolean relation depending on the operator *op*. Finally, for all ten non-trivial two-input Boolean operators, we reported experiments to compare such realizations vs. SOP forms, and three-level forms, in terms of area and delay evaluated by synthesing and mapping the circuits with SIS. These experiments showed high gains in a majority of benchmarks against affordable increases of run time. Future work includes: 1) studying how to choose the best operator for a particular incompletely specified function, this might be based on solving a phase assignment problem within each of the three group classifications; 2) investigating the impact of 3-input operators like multiplexers; 3) defining an iterative procedure that repeats the decomposition of the blocks and obtains a multilevel bi-decomposition, trading-off depth vs. size.

A more ambitious step would be to modify the search engine of BREL incorporating in it knowledge about the decomposition, e.g., examining at each step whether to implement a function or its complement. Moreover, even though we gave a direct formulation of a single Boolean relation to handle multioutput incompletely specified functions in order to share logic among the $2m$ outputs, the straightforward approach does not scale and more investigation is needed to trade-off quality of the final solution vs. computing time.

Finally, it would be interesting to describe a trade-off metric that measures when it is more profitable to calculate the complementation than implement original function.

References

[1] D. Bañeres, J. Cortadella, and M. Kishinevsky, "A Recursive Paradigm to Solve Boolean Relations," *IEEE Transactions on Computers*, vol. 58, no. 4, pp. 512–527, April 2009.

[2] A. Bernasconi, V. Ciriani, R. Drechsler, and T. Villa, "Logic Minimization and Testability of 2-SPP Networks," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 7, pp. 1190–1202, July 2008.

[3] A. Bernasconi, R. K. Brayton, V. Ciriani, G. Trucco, and T. Villa, "Bi-decomposition using boolean relations," in *2015 Euromicro Conference on Digital System Design, DSD 2015, Madeira, Portugal, August 26-28, 2015*, 2015, pp. 72–78.

[4] ——, "Complemented circuits," in *12th International Workshop on Boolean Problems (IWSBP2016)*, 2016.

[5] G. Borowik and T. Łuba, *Fast Algorithm of Attribute Reduction Based on the Complementation of Boolean Function.* Heidelberg: Springer International Publishing, 2014, pp. 25–41.

[6] V. Ciriani, "Synthesis of SPP Three-Level Logic Networks using Affine Spaces," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1310–1323, 2003.

[7] V. Ciriani and A. Bernasconi, "2-SPP: a Practical Trade-Off between SP and SPP Synthesis," in *5th International Workshop on Boolean Problems (IWSBP2002)*, 2002, pp. 133–140.

[8] D. Debnath and T. Sasao, "Minimization of AND-OR-EXOR Three-Level Networks with AND Gate Sharing," *IEICE Trans. Information and Systems*, vol. E80-D, no. 10, pp. 1001–1008, 1997.

[9] ——, "Multiple–Valued Minimization to Optimize PLAs with Output EXOR Gates," in *IEEE International Symposium on Multiple-Valued Logic*, 1999, pp. 99–104.

[10] D. Debnath and Z. Vranesic, "A Fast Algorithm for OR-AND-OR Synthesis," *IEEE Transactions on CAD*, vol. 22, no. 9, pp. 1166–1176, 2003.

[11] E. Dubrova and P. Ellervee, "A fast algorithm for three-level logic optimization," in *IEEE/ACM International Workshop on Logic & Synthesis (IWLS)*, 1999.

[12] E. Dubrova, D. Miller, and J. Muzio, "AOXMIN: A Three-Level Heuristic AND-OR-XOR Minimizer for Boolean Functions," in *3rd International Workshop on the Applications of the Reed-Muller Expansion in Circuit Design*, 1997, pp. 209–218.

[13] C. Jankowski, D. Reda, M. Mañkowski, and G. Borowik, "Discretization of data using Boolean transformations and information theory based evaluation criteria," *Bulletin of the Polish Academy of Sciences*, vol. 63, no. 4, pp. 923–932, 2015.

[14] F. Luccio and L. Pagli, "On a New Boolean Function with Applications," *IEEE Transactions on Computers*, vol. 48, no. 3, pp. 296–310, 1999.

[15] A. A. Malik, D. Harrison, and R. K. Brayton, "Three-level decomposition with application to plds," in *Proceedings 1991 IEEE International Conference on Computer Design: VLSI in Computer & Processors, ICCD '91, Cambridge, MA, USA, October 14-16, 1991*, 1991, pp. 628–633.

[16] A. Mishchenko, B. Steinbach, and M. Perkowski, "An algorithm for bi-decomposition of logic functions," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 103–108.

[17] M. Perkowski and S. Grygiel, "A Survey of Literature on Function Decomposition," PSU Electrical Engineering Department, Technical Report November 20, 1995, 1995.

[18] R. Rudell and A. Sangiovanni-Vincentelli, "Multiple Valued Minimization for PLA Optimization," *IEEE Transactions on CAD*, vol. 6, no. 5, pp. 727–750, 1987.

[19] T. Sasao, "On the complexity of three-level logic circuits," in *IEEE/ACM International Workshop on Logic & Synthesis (IWLS)*, 1989.

[20] ——, "OR-AND-OR Three-Level Networks," in *Representation of Discrete Functions*, T. Sasao and M. Fujita, Eds. Kluwier Academic, 1996.

[21] T. Sasao and J. Butler, "On bi-decomposition of logic functions," in *Int. Workshop on Logic Synthesis*, 1997.

[22] C. Scholl, *Functional Decomposition with Applications to FPGA Synthesis*. Springer, 2002.

[23] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," Microelectronic Center, User Guide, 1991.