Finding Minimal ESCT expressions for Boolean functions with weight of up to 7 - PRELIMINARY VERSION

Dimitrios Voudouris, Marinos Sampson and George Papakonstantinou
(dvoudour@cslab.ece.ntua.gr, sampson@cslab.ece.ntua.gr, papakon@cslab.ece.ntua.gr)
National Technical University of Athens, Greece.

January 17, 2009

Abstract

In this paper an algorithm is proposed for the synthesis and exact minimization of ESCT (Exclusive or Sum of Complex Terms) expressions for Boolean functions of up to seven complex terms, regardless of the number of input variables. This kind of logical expressions can be mapped to a special cellular architecture, called Reversible Wave Cascade Architecture. This topology is proved to be very useful, as it is reversible and therefore it may help in the design of quantum circuits. Moreover, the proposed algorithm is extended heuristically for functions with eight or more complex terms.

1 Introduction

For many years, logic synthesis was based on AND, OR and NOT gates. However, for some very important groups of applications such as arithmetic, error correcting and telecommunication applications, the use of XOR (eXclusive OR) gates can reduce the complexity of logic circuits. The most general (and most powerful) AND-XOR expression, is the "Exclusive or Sum Of Products" (ESOP) expression where a function is represented as the XOR sum of logical products (Logical ANDs of variable literals) [1]. The natural evolution to these expressions are the "Exclusive or Sum of Complex Terms" (ESCT) expressions, where every term may additionally use the logical OR and XOR operations. They were introduced by K. K. Maitra in 1962 and are suited for mapping to a Reversible Wave Cascade architecture (Fig. 1).

It has been proved [2] that an ESCT expression can be directly mapped to reversible logic gates and more specifically to Generalized Toffoli gates. A logic gate is called reversible, if it has the same number of inputs and outputs, and maps each input vector into a unique output vector and vice versa. Moreover, both fan-in and fan-out are forbidden. One of their important properties is that they consume minimal amounts of power, due to the fact that they loose no information [3]. However for current CMOS technology the power lost, because of information loss, is minimal, therefore reversible logic does not bring any real advantage.

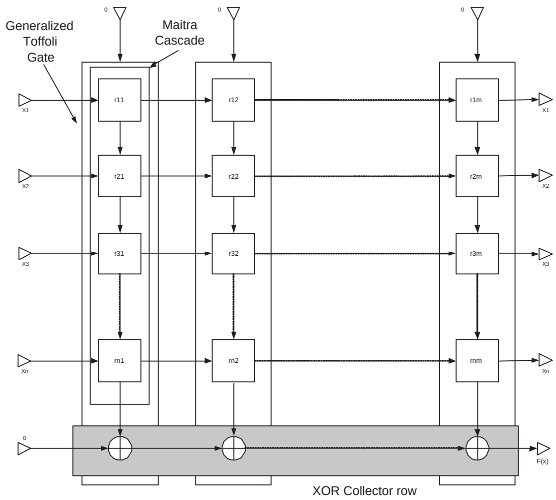

Figure 1: Reversible wave cascade architecture

In [4] it was shown that all quantum logic gates must be reversible. Due to this, the reversible wave cascades is a very attractive architecture for the implementation of reversible logic circuits and perhaps quantum logic. In Fig. 1, a Reversible Wave Cascade is shown, representing an ESCT expression. It can be observed that each column of this architecture (a complex term), along with an additional XOR function can be considered as a Generalized Toffoli gate.

There were algorithms developed in the past, for mapping switching functions to Cellular Array (CA) architectures and then minimizing the number of the produced complex terms. References [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] and [15] are some examples of the overall progress performed in this field. The state of the art in finding minimal (exact) solutions for ESCT expressions for up to 6 input variables is [15]

In this paper, we introduce an algorithm (W7XMIN) that can produce minimal expressions (the ones with the least number of complex terms) for a single-output switching function f with up to 7 complex terms. To the best of the authors knowledge, this is the first algorithm in the related literature, that can find minimal ESCT expressions for switching functions of up to 7 complex terms, regardless of the number of input variables. Hence, this paper adds, somehow, to the state of the art in exact ESCT minimization. Moreover, a heuristic extension (EW7MIN) is presented which is more efficient than W7XMIN, giving practically the same results. Lastly, EW7MIN is used as the basic simplification module in Min2 [12], resulting in heuristic algorithm MIN2-EW7MIN. Min2 can be used in conjuction with the characteristic function [14] for minimizing multi-output Boolean functions. The results of MIN2-EW7MIN for multi-output functions are comparable or better to the ones in the bibliography.

2 Theoretical background

In this section we provide some background definitions. An expression of a switching function suitable for mapping to a Maitra cascade (cell chain) is called a Maitra term. A more formal definition[2] follows:

Definition 1 *A complex Maitra term (complex term or Maitra term for simplicity) is recursively defined as follows:*

1. *Constant* 0 *or* 1 *Boolean function is a Maitra term.*
2. *A literal is a Maitra term.*
3. *If M_i is a Maitra term, x^* is a literal, and G is an arbitrary two-variable Boolean function (Maitra cell), then $M_{i+1} = G(x^*, M_i)$ is a Maitra term.*

Additionally, it is required that each variable appears in each Maitra term only once. In other words a complex term is: $P = G_n(x_n^*, G_{n-1}(x_{n-1}^*, \ldots G_1(x_1^*, 0)))$, where x_i^* are literals and x_i are the variables that P depends on. It is noted that in this previous definition x_i^* does not need to be in its negative form, since for every single output two variable Boolean function $G_n(x, y)$, there exists another single output two variable Boolean function $G_n^{'}$ such that: $G_n(x, y) = G_n^{'}(\bar{x}, y)$.

Definition 2 *An ESCT (Exclusive-or Sum of Complex Terms) expression (sometimes also called Maitra expression) for a switching function, is a XOR sum of complex terms:*

$$f = \sum_{i=1}^{m} \oplus M_i,$$

where M_i are complex terms and m (size of the ESCT expression) is their number inside the expression. The same variable ordering is used for every M_i.

It can be observed that a complex term defines a specific variable ordering for its variables. For P, x_n is the most significant variable and x_1 is the least significant ($x_n, x_{n-1}, \ldots, x_2, x_1$ is the order of variable significance). If the variable ordering changes then, it is possible that the new expression produced may not be a complex term.

Definition 3 *A minimal (or exact) expression of a switching function $f(x_1, \ldots, x_n)$ of n variables, is defined as the ESCT expression which has the least number of terms comparing to every other ESCT expression for this function.*

Definition 4 *The ESCT weight $w(f)$ (or simply weight) of a switching function $f(x_1, \ldots, x_n)$ of n variables, is defined as the number of complex terms in a minimal ESCT expression of f.*

Definition 5 *An n-wequivalent ESCT expression of a Boolean function f is an ESCT expression of f which has weight equal to $w(f) + n$.*

An ESCT expression can be directly mapped to a special cellular architecture, called Reversible Wave cascade (Fig. 1). A complex (Maitra) term is mapped to a column in the reversible wave cascade, excluding the last XOR cell. Each column is composed of cells ($r_{ij}, 1 \leq i \leq n, 1 \leq j \leq m$). The horizontal

Table 1: Cell index set

Cell index(r)	$F_r(x, y)$
1	$x + y$
2	$\overline{x} + y$
3	$\overline{x}y$
4	xy
5	$x \oplus y$
6	y

input to each cell is a variable and is propagated to the horizontal output of the cell. The vertical input is the output of the previous cell in the same column (or the constant 0 in the case of the first cell of each column). The outputs of each column are connected to the XOR collector, thus obtaining the function $F(x)$. Every such cell implements a single output two variable switching function (this function corresponds to the Maitra cell of Definition 1).

It has been proved [16] that a Maitra cell doesn't need to implement every two-variable switching function. A set of only six functions is sufficient (complete set). Of course, there are many equivalent such sets[17]. We have adopted (in the rest of the paper) one of them which can be seen in Table 1.

An ESCT expression is composed of reversible Generalized Toffoli gates [2], thus a Reversible Wave Cascade, as its name suggests, is a reversible architecture.

2.1 Representation

Next, we will present the definitions of the representation used in this paper, for an arbitrary Boolean function and an arbitrary ESCT expression.

Definition 6 *The minterm representation (MT) of a switching function f with n variables, is a bitvector of size 2^n where the i-th bit is 1 if the i-th minterm of f is 1.*

It can easily be observed that the MT representation of a Boolean function depends on its assumed variable ordering.

For the rest of this paper, the MT representation of a Boolean function will be enclosed in brackets and will be displayed using hex digits. The hexadecimal notation is used in place of the binary one, in order to improve the readability of the paper.

A complex term is characterized by its cells, since its first input is the constant 0. Therefore, we can represent it by a series of cells.

Definition 7 *The cell representation of a complex term, is a series of numbers, corresponding to the series of indices of Maitra cells (Table 1) that belong to the complex term. The leftmost cell corresponds to the cell with the constant input 0 (corresponds to the least significant variable of the complex term) and the rightmost to the one closest to the XOR collector (corresponds to the most significant variable of the complex term).*

For the rest of this paper the cell representation of a complex term will be enclosed in parenthesis (as opposed to the minterm representation, which is enclosed in brackets).

Every Boolean function can be expressed with the help of its subfunctions (they are defined below) through relations, known as Boolean decompositions (or expansions). The Boolean decompositions form the backbone of our theoretical approach for the minimization of Boolean functions.

Definition 8 *Let $f(\boldsymbol{x})$ be a switching function and $\boldsymbol{x}$ the vector of its variables. Let x_i be one of the variables in the vector $\boldsymbol{x}$. Then $f(x_1, x_2, \ldots, x_i = 0, \ldots)$, $f(x_1, x_2, \ldots, x_i = 1, \ldots)$ and $\{f(x_1, x_2, \ldots, x_i = 0, \ldots) \oplus f(x_1, x_2, \ldots, x_i = 1, \ldots)\}$ are subfunctions of f, regarding variable x_i. For simplicity, in the rest of this paper, they will be referred as f_0, f_1 and f_2 respectively and x_i will be referred as x.*

A Boolean function f can be expressed as (Shannon, Positive Davio and Negative Davio respectively):

- $f(\mathbf{x}) = \bar{x}f_0 \oplus xf_1$
- $f(\mathbf{x}) = xf_2 \oplus f_0$
- $f(\mathbf{x}) = \bar{x}f_2 \oplus f_1$

The next definition defines the generator tree, which will be used later by our minimization algorithms.

Definition 9 *Let f be an n-variable switching function. By creating the subfunctions f_1, f_0, f_2 of f and then the subfunctions of f's subfunctions and so on (recursively), a ternary tree is generated. The leftmost branch of each subtree represents the f_1 subfunction of the subtree's root, the middle one represents the f_0 subfunction and the rightmost represents the f_2 subfunction. This decomposition is applied until the constant 0 or 1 function is encountered or a leaf (a two variable Boolean function) is obtained. This tree is named the generator tree.*

It was proved in [13] that if P is a complex term, then $\overline{P}, P \oplus x$ and $P \oplus \overline{x}$ are also complex terms, where x is the most significant variable of P. Hence, the ESCT weight of an arbitrary function f is equal to the weight of $\overline{f}$, $f \oplus x$ and $f \oplus \overline{x}$.

3 ESCT Theoretical Background

The following theorems give the theoretical background for the algorithms presented in the next section.

Definition 10 *An mequivalent expression (F_2) of an ESCT expression ($F_1 = \ldots \oplus P_i \oplus \ldots \oplus P_j \oplus \ldots$) for a switching function $f(x_1, \ldots x_n)$ (x_n is its most significant variable) is an expression produced by applying the following transformations to F_1:*

- $F_2 = \ldots \oplus \overline{P_i} \oplus \ldots \oplus \overline{P_j} \oplus \ldots$.

- $F_2 = \ldots \oplus (P_i \oplus x_n^*) \oplus \ldots \oplus (P_j \oplus x_n^*) \oplus \ldots$, *where* $x_n^* = x_n, \overline{x_n}$.

The above transformations can, also, be applied to pairs of descendants of f in its generator tree.

For example if $F_1 = (1234) \oplus (2343)$, then an mequivalent expression of F_1 by complementing both terms, is: $F_2 = \{(1234) \oplus 1\} \oplus \{(2343) \oplus 1\} = (2412) \oplus (1121)$.

Lemma 1 (Two cell merging) *The relation*
$F_{r_1}(x, y_1) \oplus F_{r_2}(x, y_2) = F_r(x, y_1 \oplus y_2)$, *where* $y_1 \neq y_2$, $y_1 \neq \bar{y_2}$ *and x the most significant variable of complex terms* F_{r_1}, F_{r_2}, F_r
is true iff:
$(r_1, r_2, r) = (1,1,3), (1,3,1), (2,2,4), (2,4,2), (3,3,3),$
$(4,4,4), (5,5,6), (5,6,5), (6,6,6)$
Proof. It can, easily, be proved exhaustively.Q.E.D.

Theorem 1 *Each minimal expression of a switching function f can always be written in one of the following compact forms:*

$$f = F_p(x_n, y) \tag{1}$$

(with one subfunction constant)

OR

$$f = F_p(x_n, y) \oplus F_q(x_n, z) \tag{2}$$

OR

$$f = F_p(x_n, y) \oplus F_q(x_n, z) \oplus F_r(x_n, g) \tag{3}$$

and $f_i = y \oplus g^*, f_j = z \oplus g^*, f_k = y \oplus z^*, f_i, f_j, f_k = f_0, f_1, f_2, g^* = g, \bar{g}, z^* = z, \bar{z}$.

The valid combinations of cell indices and their corresponding inputs are presented in [12].

Proof. It has been proved in [12].Q.E.D.

For example, a minimal expression for $f(x_1, \ldots, x_5) = [A7122347]$ is: $f = (13443) \oplus (11344) \oplus (26654) \oplus (61166) \oplus (12616)$ or in its compact form: $F_3(x_5, (1344)) \oplus F_4(x_5, (1134) \oplus (2665)) \oplus F_6(x_5, (6116) \oplus (1261))$.

Algorithm Min1 was presented in [12] and it was able to find exact ESCT expressions for functions of up to five input variables. It did that by decomposing its input function, thus creating the generator tree. For each function, corresponding to a node of this tree, it produced minimal ESCT expressions by merging common complex terms between minimal expressions belonging to the functions that corresponded to its leaves. To do that it needed all minimal ESCT expressions for each node of the generator tree. But for functions with weight more than 5 it was not, always, able to produce minimal ESCT expressions or produce every possible minimal ESCT expression. Consider the following example:

***Example* 1** *Consider the following 5-input single output Boolean function (given in MT formulation):* $g = [19cd0acc]$. *Each term, inside the ESCT expressions of this example, will be expressed using both the MT formulation and the cell representation. The weight of g is equal to 5 and a minimal ESCT expression for g is:* $[00220022] \oplus [09000000] \oplus [00000a00] \oplus [000000ee] \oplus [10ef0000] = (13636) \oplus (25344) \oplus (16343) \oplus (11633) \oplus (11254)$.

But this expression cannot be produced by Min1, because this algorithm finds only some minimal ESCT expressions for 5 input Boolean functions. Although this fact does not seem important, it becomes important if we try to minimize, using Min1, the function: $f = [a75842a6be95486a]$ *(function g is a subfunction of f). A minimal ESCT expression of f is:* $[0000000000220022] \oplus [0000000009000000] \oplus [7777888877778888] \oplus [0808080808080808] \oplus [00000a0000000000] \oplus [000000ee00000000] \oplus [10ef000000000000] = (136364) \oplus (253344) \oplus (146656) \oplus (614666) \oplus (163434) \oplus (116334) \oplus (112544)$. *This expression is produced by an ESCT expression of* f_0 *and by an ESCT expression of* $f_2 = g$ *by merging two common complex terms (*$[00220022]$ *and* $[00090000]$*) to create the terms* $[0000000000220022], [0000000009000000]$ *which are shown in the above expression. Min1 is able to produce the appropriate ESCT expression* $[00220022] \oplus [09000000] \oplus [f807f807] \oplus [4fb0b04f] = (13636) \oplus (25334) \oplus (22356) \oplus (24255)$ *for* f_0.

But since Min1 cannot produce all minimal ESCT expressions for $g = f_2$ *it cannot produce the minimal ESCT expression for f.*

Cases, like the one presented in example 1, are considered in the theoretical results that follow.

3.1 Minimization Theorems

Theorem 2 *Let f a Boolean function with* $w(f) < 6$ *and* f_0, f_1, f_2 *its subfunctions. All minimal ESCT expression for f can be found from the minimal ESCT expressions of* f_0, f_1, f_2.

This Theorem denotes that all possible ESCT expressions for f can be produced.

Theorem 3 *Let f a Boolean function with* $5 < w(f) < 8$ *and* f_0, f_1, f_2 *its subfunctions. At least one minimal ESCT expression for f can be found from the minimal ESCT expressions of* f_0, f_1, f_2.

This previous theorem cannot produce all minimal solutions in every possible case.

4 The Minimizing Algorithms

Algorithm W7XMIN implements the theorems and the methodology presented in the previous section. It is an extension of algorithm Min1 [12] and it's recursive. It decomposes its input function into its subfunctions creating its generator tree. The composition method is identical to that of Min1 but in certain cases and weight combinations, as depicted by Table 2, it implements the methodology described in the previous section. At every level of the decomposition it creates ESCT expressions, using the Shannon, Positive and Negative Davio expansions, along with their mequivalents.

For example, let's suppose that W7XMIN needs to create the minimal ESCT expressions of a node (corresponding to function f) inside the generator tree of its input function. Let's suppose that the children of this node (corresponding to f's subfunctions) have weights: $w(f_0) = 4, w(f_1) = 2, w(f_2) = 4$. Then according to Table 2 this weight combination can only mean that $w(f) = 5$ or 6. Moreover it must be $w(g) = 1$. There are two pairs of functions to be considered: (f_0, f_1), (f_2, f_0) that correspond to weight combination $(4, 2)$. For the first pair W7XMIN will create and minimize functions: $f_0 \oplus g, f_1 \oplus g$ where g is one complex term from the minimal expressions of f_1, f_0 respectively. For the second pair W7XMIN will create and minimize functions: $f_2 \oplus g, f_1 \oplus g$ where g is one complex term from the minimal expressions of f_1, f_2 respectively. From the minimal ESCT expressions of functions $g, f_0 \oplus g, f_1 \oplus g, f_2 \oplus g$ W7XMIN will create expressions for f and will evaluate their size, adjusting the weight of f as necessary.

Another algorithm was constructed (EW7MIN) as a heuristic extension of W7XMIN. Algorithm EW7MIN differs from W7XMIN at one point. It only produces mequivalent expressions at those levels of the generator tree where there is at least one constant function equal to one. This is justified by the fact that, according to Theorem 1, when there are no constant subfunctions inside the generator tree of a boolean function f then there is always at least one minimal ESCT expression of f having Maitra cells only 3,4,6. This kind of Maitra cells is produced by the Shannon and Davio expansions and they are not normally produced by the mequivalent transformations. Even if there are constant subfunctions but they are equal to zero then there is at least one minimal ESCT expression having, only, Maitra cells 3,4,6. But when a subfunction is constant and equal to 1, then Maitra Cells 1,2,5 will be created (these Maitra cells are normally created by the mequivalent transformations). In these cases we create the additional mequivalent expressions.

Moreover, algorithm Min2 [12] has been modified in order to use EW7MIN as its term transformation and minimization algorithm. The new and improved Min2 algorithm is called Min2-EW7MIN and can be used for Boolean functions that are so complicated that even EW7MIN timeouts. Min2-EW7MIN is an iterative algorithm that starts from an initial ESCT expression of an arbitrary Boolean function and uses EW7MIN in groups of complex terms. Each such group is minimized and reinserted into the starting ESCT expression. Min2-EW7MIN is terminated after a predetermined number of iterations.

5 Conclusions

The theoretical background for finding minimal ESCT expressions for arbitrary Boolean function with ESCT weight at most 7 is presented in this paper. Algorithm W7XMIN will be constructed based on it as well as algorithm EW7MIN which is a heuristic extension of the first one. This algorithm , although heuristic, may produce very good results, while keeping the basic minimization methodology of W7XMIN, almost, intact. Then we can use this algorithm as a module for iterative minimizer Min2-EW7MIN.

References

[1] T. Sasao "Switching Theory For Logic Synthesis", Kluwer Academic Publishers, 1999.

[2] A. Mishchenko, M. Perkowski, "Logic Synthesis of Reversible Wave Cascades", Proc. Intrl Workshop on Logic And Synthesis 2002, New Orleans, Louisiana, June 4-7, 2002, pp. 197-202.

[3] C. Bennet "Logical Reversibility of Computation", IBM Journal of Research and Development, 17, 1973, pp. 525-532.

[4] J. Preskill "Lecture notes in quantum computing", http://www.Theory.caltech.edu/ preskill/ph229

[5] A. Sarabi, N. Song, M. Chrzanowska-Jeske, M. Perkowski"A comprehensive approach to logic synthesis and physical design for two-dimensional logic arrays", In proc. of Design Automation Conference 1994, pp. 321-326.

[6] I. Schaefer, M. Perkowski, H. Wu "Multilevel logic synthesis for cellular FPGAs based on orthogonal expansions", Proc, IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, Sept. 1993, Hamburg, Germany, pp. 42-51.

[7] G. Lee "Logic synthesis for celullar architecture FPGA using BDD", Design Automation Conference 1007, Proc. of Asian-Pacific DAC 97, Jan 1997, pp 253-258 .

[8] G. Lee, R. Drechsler "ETDD-based Synthesis of term-based FPGAs for incompletely specified Boolean functions", Design Automation Conference 1998, Proc. of Asian-Pacific DAC 1998, Feb 1998, pp 75-80.

[9] P. Lindgren, R. Drechsler, B. Becker "Look-up table FPGA synthesis from minimized multi-valued pseudo kronecker expressions", in proc. of the 28th IEEE International Symposium on Multiple-Valued Logic, 27-29 May 1998, pp. 95 - 100.

[10] N. Song, M. Perkowski "Minimization of exclusive sums of multi-valued complex terms for logic cell arrays", Proc. 28th IEEE Intl. Symposium on MV Logic, 27-29 May 1998, pp. 32 - 37.

[11] N. Song, M. Perkowski "A new approach to and/or/exor factorization for regular arrays", Proc. 1998 Euromicro, Vasteras, Sweden, August 25-27, 1998, pp. 269-276.

[12] D. Voudouris, S. Stergiou, G. Papakonstantinou "Minimization of reversible wave cascades", IEICE Trans. on Fundamentals, Vol E88-A, No. 4, 2005/04, pp. 1015-1023.

[13] D. Voudouris, G. Papakonstantinou "Maitra Cascade Minimization", in proc. of the 6th IWSBP, September 2004, Freiberg (Sachsen), Germany, pp. 209 - 220.

[14] D. Voudouris, M. Kalathas, G. Papakonstantinou "Decomposition of Multi-output Boolean Functions", HERCMA 2005, Athens, Hellas, HERMIS Journal Vol 6-2005, pp. 154-161.

[15] D. Voudouris, M. Sampson, G. Papakonstantinou "Exact ESCT Minimization for functions of up to six input variables", Elsevier Integr. VLSI J. 41, 1 (Jan. 2008), 87-105. DOI= http://dx.doi.org/10.1016/j.vlsi.2007.01.003.

[16] R. C. Minnick, ”Cutpoint cellular logic” IEEE Trans. Electron. Comput., vol. EC-13, Dec, 1964, pp. 685-698.

[17] G. Papakonstantinou ”Cascade Transformation”, IEEE Transactions on computers, Jan 1976, 25(1), pp. 93 - 95.

[18] G. Papakonstantinou, F. Gritzali ”Modulo-2 expressions of switching functions”, Electronic Letters, vol 12, issue 10, May 1976, pp. 244 - 245.

[19] G. Papakonstantinou ”Synthesis of cutpoing cellular arrays with exclusive-OR collector row”, Electronic Letters, 13(1977).

[20] S. Stergiou, D. Voudouris, G. Papakonstantinou ”Exact and Heuristic MVESOP Minimization Algorithms”, IEICE Trans. on Fundamentals, Vol E87-A No.5, pp.1226.

[21] N. Song, M. Perkowski ”Minimization of exclusive sum of products expressions for multiple-valued input incompletely specified functions”, Master thesis, EE Dept. Portland State University, Portland, OR, 1992.