NEW FAST APPROACH TO APPROXIMATE ESOP MINIMIZATION FOR INCOMPLETELY SPECIFIED MULTI-OUTPUT FUNCTIONS

Ning Song
Lattice Semiconductor Corporation
1820 McCarthy Blvd.
Milpitas, CA 95035
ning@latticesemi.com

Marek A. Perkowski
Department of Electrical Engineering
Portland State University
P.O. Box 751, Portland, OR 97207
mperkows@ee.pdx.edu

Abstract— The paper presents an improvement to Exorlink operations, and a new approach to their use for Exclusive Sum of Products (ESOP) minimization. A new approach based on look-ahead search strategies is also introduced. Our program, EXORCISM-MV-3, is the successor to EXORCISM-MV-2, and can minimize multi-valued input, binary-output, incompletely specified, multi-output functions with different and arbitrary numbers of values in every variable. This makes it also useful in many applications, such as Machine Learning, and optimization of PLAs with input decoders. We observed that the new program is extremely fast; the speed-ups of up to 141 times were observed on large functions.

I. INTRODUCTION.

Although a classic in EXOR Logic, the ESOP minimization problem is far from being solved, and numerous results [7, 24, 32] point to various possible improvements of the existing ESOP minimizers. For instance, GRM or FPRM minimizers find, for some benchmark functions, the solutions that are better than the best known solutions of ESOP minimizers. Thus, despite many important works in recent years [5, 6, 7, 21, 22, 24, 31], the ESOP minimization problem still remains an open research area.

The work presented here is an extension and improvement to the approach from [28], implemented in program EXORCISM-MV-2. Here the application of the Exorlink operation was further improved by use of look-ahead ideas. As EXORCISM-MV-2, our new program, EXORCISM-MV-3, is able to find high quality solutions to incomplete multi-output functions with arbitrary number of values for each variable (which allows to run it with input decoders having more than two binary inputs).

Its main advantage is, however, that EXORCISM-MV-3 is very fast. In many practical applications such as: the minimization of regular arrays [27], the minimization of Generalized Reed Muller Forms [32], the minimization of arithmetic functions and functions specified with initial AND/EXOR descriptions [29], the multi-method logical approaches to Machine Learning [10], the array realizations of linearly independent logic [11], the multi-level realizations of linearly independent logic [13, 14], and the lattice realizations of functions [15, 16, 17] the ESOP minimization program must be very fast, because it is used in a loop of another minimization program. EXORCISM-MV-3 satisfies this requirement.

II. BACKGROUND ON MULTPLE-VALUED EXORLINK OPERATIONS.

We review here the basic definitions of Exorlink for multiple-valued functions [28].

Definition 1. A multiple-valued input, two-valued output, incompletely specified switching function f (*multiple-valued function, for short*) is a mapping f($X_1, X_2, ..., X_n$): $P_1 \times P_2 \times ...P_n \rightarrow B$, where X_i is a *multiple-valued variable*, $P_i = \{0, 1, ..., p_i - 1\}$ is a *set of admissible values* that this variable may assume, and B = {0, 1, x} (x denotes a *don't care value*).

Definition 2. For any subset $S_i \subseteq P_i$, $X_i^{S_i}$ is a *literal* of X_i representing the function such that

$$X_i^{S_i} = \begin{cases} 1 & \text{if } X_i \in S_i \\ 0 & \text{if } X_i \notin S_i. \end{cases}$$

Definition 3. A product of literals, $X_1^{S_1} X_2^{S_2} ... X_n^{S_n}$, is referred to as a *product term* (denoted by PT, and also called *term* or *product* for short). A *minterm* is a product term that there exists only one value in each S_i for $i = 1, 2, ..., n$.

Definition 4. The *Exclusive-OR* (EXOR for short) of two products is assigned the value 1 if and only if the two products have different values. An EXOR of products is called an *Exclusive Sum of Products Expression* (*ESOP* for short). It is also called a *Multiple-Valued Input Exclusive Sum of Products Expression* (*MIESOP* for short) if one wants to emphasize that the input variables are multiple valued.

Example 1a. In binary logic, given three terms $T_1 = x\bar{y}\bar{z}$, $T_2 = \bar{x}y\bar{z}$, and $T_3 = \bar{x}\bar{y}z$. $T_1 \oplus T_2 \oplus T_3$ is an ESOP. Since the binary logic is a special case of *multiple-valued*, the above three product terms can be written in *multiple-valued* form as $T_1 = X^{\{1\}}Y^{\{0\}}Z^{\{0\}}$, $T_2 = X^{\{0\}}Y^{\{1\}}Z^{\{0\}}$, $T_3 = X^{\{0\}}Y^{\{0\}}Z^{\{1\}}$.

Example 1b. In 4-valued logic, given three terms $T_1 = X^{\{1,2\}}Y^{\{2,3\}}$, $T_2 = X^{\{2,3\}}Y^{\{1,2\}}$, and $T_3 = X^{\{0,1\}}Y^{\{1,3\}}$. $T_1 \oplus T_2 \oplus T_3$ is a MIESOP. We can also call it an ESOP.

In cube notation, a term is represented by a cube, and each literal in the term is represented by a vector:

$$c_1^0 c_1^1 ... c_1^{(p_1-1)} - c_2^0 c_2^1 ... c_2^{(p_2-1)} - ... - c_n^0 c_n^1 ... c_n^{(p_n-1)}$$

where

$$c_i^j = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{if } j \notin S_i. \end{cases}$$

For example, $X^{\{0\}}$ is denoted by 100...000, $X^{\{1\}}$ is denoted by 010...000, $X^{\{0,2\}}$ is denoted by 101...000, and $X^{\{0,1,...n\}}$ is denoted by 111...111, which represents the Boolean universe. A cube is a null cube, if one or more variables contain all 0s.

Example 2. The ESOP in *example 1a* can be written in the following cube notation:

$$[01 \text{ - } 10 \text{ - } 10] \oplus [01 \text{ - } 10 \text{ - } 10] \oplus [10 \text{ - } 10 \text{ - } 01].$$

The ESOP in *example 1b* can be written in cube notation as follows:

$$[0110 \text{ - } 0011] \oplus [0011 \text{ - } 0110] \oplus [1100 \text{ - } 0101].$$

Definition 5. The *distance of two terms* is the number of variables for which the corresponding literals have different sets of values.

Example 3a. The distance of $T_1 = abd$ and $T_2 = a\bar{b}c$ is 3, because three literals have different sets of values:

for a: $\{01\} = \{01\}$, for b: $\{01\} \neq \{10\}$,

for c: $\{11\} \neq \{01\}$, for d: $\{01\} \neq \{11\}$.

Example 3b. Given three terms $T_1 = X^{\{0\}}Y^{\{1\}}$, $T_2 = X^{\{1\}}Y^{\{0,2\}}$, and $T_3 = X^{\{1\}}Y^{\{0,1\}}$. The distance of T_1 and T_2 is 2, because two literals have different sets of values:

for X: $\{0\} \neq \{1\}$, for Y: $\{1\} \neq \{0{,}2\}$.

The distance of T_2 and T_3 is 1, because only one literal has different sets of values:

for X: $\{1\} = \{1\}$, for Y: $\{0{,}2\} \neq \{0{,}1\}$.

We write *distance(T_i, T_j)* $= d$ to indicate that the distance of two terms T_i and T_j is d.

The objective of logic minimization is to find a realization that reduces certain cost function. Our primary goal of ESOP synthesis is to *minimize the number of terms* in the ESOP expression. For the expression with the minimum number of terms our secondary goal is to minimize the total number of inputs to the AND and EXOR gates. Following the notation in [28], we use C_T to indicate the number of terms and C_L to indicate the number of literals.

Let $T_S = X_1^{S_1}...X_n^{S_n}$ and $T_R = X_1^{R_1}...X_n^{R_n}$ be two terms. The *exorlink* [28] of terms T_S and T_R is defined by the following formula:

$$T_S \otimes T_R = \bigoplus \left\{ X_1^{S_1}...X_{i-1}^{S_{i-1}} X_i^{(S_i \oplus R_i)} X_{i+1}^{R_{i+1}}...X_n^{R_n} \mid \text{for such } i = 1, ..., n, \text{that } S_i \neq R_i \right\}.$$

Here $\otimes$ denotes the exorlink operation and $\oplus$ denotes the EXOR operation.

Definition 6. Given terms T_S and T_R, if the distance of two terms is d, then $T_S \otimes T_R$ is a *distance d exorlink.* It was proved in [26] that the exorlink can be applied to any two cubes in an array, without regard to their distance. According to the above formula, we can observe that distance d exorlink generates d resultant terms.

In the remainder of this section, distance 0, distance 1, distance 2, and distance 3 exorlink operations will be discussed.

Distance 0 Exorlink.

If the distance of two terms is 0, exorlink of the terms generates no resultant terms. These two terms are then removed from the ESOP description.

Distance 1 Exorlink.

Given two terms T_S and T_R, let X^{S_i} and X^{R_i} be a pair of literals in terms T_S and T_R, respectively, such that $X^{S_i} \neq X^{R_i}$, and the other pairs of literals in the two terms are equal. Therefore, these two terms will be called *"Distance 1 Exorlinkable."* The Distance 1 Exorlink operation of the terms generates a single resultant term.

Example 4. Let $T_S = X^{\{1,2,3\}}Y^{\{2,3\}}$ and $T_R = X^{\{0,1\}}Y^{\{2,3\}}$.

$$T_S \otimes T_R = X^{\{1,2,3\} \oplus \{0,1\}}Y^{\{2,3\}} = X^{\{0,2,3\}}Y^{\{2,3\}}.$$

Distance 2 Exorlink.

Given two terms, if the distance of these two terms is 2 (assume $X^{S_i} \neq X^{R_i}$, and $Y^{S_j} \neq Y^{R_j}$), then distance 2 exorlink operation can be performed on them, and two resultant terms will be generated. Please note that when distance ≥ 2, exorlink operation is not symmetric, which means $T_S \otimes T_R$ is different from $T_R \otimes T_S$.

Example 5. Given two terms $T_S = X^{\{0,1,3\}}Y^{\{1,3\}}$ and $T_R = X^{\{2,3\}}Y^{\{0,1\}}$.

$$T_S \otimes T_R = X^{\{0,1,3\}}Y^{\{1,3\}} \otimes X^{\{2,3\}}Y^{\{0,1\}} = X^{\{0,1,2\}}Y^{\{0,1\}} \oplus X^{\{0,1,3\}}Y^{\{0,3\}}$$

$$T_R \otimes T_S = X^{\{2,3\}}Y^{\{0,1\}} \otimes X^{\{0,1,3\}}Y^{\{1,3\}} = X^{\{0,1,2\}}Y^{\{1,3\}} \oplus X^{\{2,3\}}Y^{\{0,3\}}.$$

Distance 2 operations do not directly reduce the number of terms in an ESOP. However, these operations reshape two terms to two different terms,

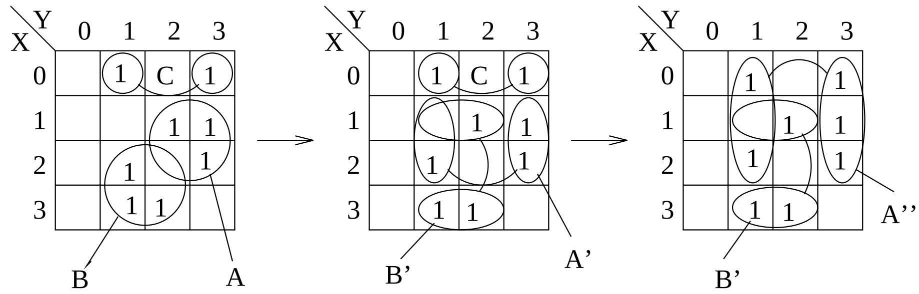

Figure 1: ESOP minimization corresponding to Example 6

thus provide opportunities for reducing the cost of ESOPs at some later stages. The non-symmetry property of the Distance 2 Exorlink gives us two ways to reshape the two terms, which increases the opportunity for searching a better result. Our method on how to apply Distance 2 and Distance 3 Exorlink in ESOP minimization is discussed in sections 3 and 4.

Example 6. Given is an ESOP with three terms: $T_1 = X^{\{1,2\}}Y^{\{2,3\}}$, $T_2 = X^{\{2,3\}}Y^{\{1,2\}}$ and $T_3 = X^{\{0\}}Y^{\{1,3\}}$. In Figure 1, the three terms T_1, T_2 and T_3 are represented by three cubes A, B and C, respectively. $A \otimes B$ generates A' and B'; $A' \otimes C$ generates A''. The ESOP with three cubes is minimized to an ESOP with two cubes, B' corresponding to term $T_4 = X^{\{1,3\}}Y^{\{1,2\}}$, and A'' corresponding to term $T_5 = X^{\{0,1,2\}}Y^{\{1,3\}}$.

Distance 3 Exorlink.

Distance 3 Exorlink generates three resultant terms from two given terms. Distance 3 Exorlink increases the number of terms in the ESOP. However, increasing the number of terms may help to reduce the number of terms at some later stage, and subsequently lead to better results.

Example 7. In binary logic, a given ESOP with 4 cubes is as follows:

$$000x,\ 0x11,\ x11x,\ 1010.$$

The distances between any pair of cubes from the above set is 3. So, there are no distance 1 or distance 2 operations that can be applied to this set of cubes. Performing Distance 3 Exorlink on the first two cubes leads to three cubes:

$$000x \otimes 0x11 = 0111 \oplus 00x1 \oplus 0000.$$

Replacing the first two cubes by these three cubes, a new ESOP with five cubes is obtained:

$$0111,\ 00x1,\ 0000,\ x11x,\ 1010.$$

Since the distance of two cubes: 0000 and 1010 is 2, a distance 2 exorlink can be performed on them:

$$0000 \otimes 1010 = x010 \oplus 00x0.$$

After this operation, the ESOP contains five cubes:

$$0111,\ 00x1,\ x010,\ x11x,\ 00x0.$$

Now, the distance of cubes $00x1$ and $00x0$ is 1, a Distance 1 Exorlink can be performed on them:

$$00x1 \otimes 00x0 = 00xx.$$

The ESOP now contains four cubes:

$$0111,\ 00xx,\ x010,\ x11x.$$

Performing Distance 2 Exorlink on cubes $x010$ and $x11x$, we obtain:

$$x010 \otimes x11x = xx10 \oplus x111.$$

The ESOP is

$$0111,\ 00xx,\ xx10,\ x111.$$

Cubes 0111 and $x111$ can be combined into one cube:

$$0111 \otimes x111 = 1111.$$

The final result is an ESOP with three cubes:

$$00xx,\ xx10,\ 1111.$$

By using Distance 3 Exorlink, the number of cubes in the ESOP is temporarily increased from 4 to 5, but this increase helps to jump out of a local minimum of the cost function, and achieve ultimately a better result of 3 cubes.

III. THE NEW EXORLINK OPERATIONS.

This section presents the description of the new exorlink operations used in EXORCISM-MV-3. Distance 0, 1 and 2 exorlink are the same as those defined in previous section and [28].

Distance 2 Exorlink. Please note that Distance 2 Exorlink operation generates 4 resultant cubes:
Given are cubes: $A = X_1\ X_2$, and $B = Y_1\ Y_2$ (for simplification of notation, we avoid here writing in cubes the parts that are the same in cubes A and B).

Applying Distance 2 Exorlink to A and B, the resultant cubes are:

$cube_1$: $X_1\ (X_2 \oplus Y_2)$, $cube_2$: $Y_1\ (X_2 \oplus Y_2)$,

$cube_3$: $(X_1 \oplus Y_1)\ Y_2$, $cube_4$: $(X_1 \oplus Y_1)\ X_2$.

Cubes $cube_1$ and $cube_3$ form one group and $cube_2$ and $cube_4$ form another group. After the Distance 2 Exorlink operation, either Group 1 or Group 2 is selected. But only the entire group can be selected, not a part of it. For instance, cubes $cube_1$ and $cube_2$ cannot be selected together as the result.

Distance 3 exorlink. Here we extend the Distance 3 Exorlink, as compared to [28]. Previously, in EXORCISM-MV-2, the Distance 3 Exorlink would generate two groups of 6 cubes.

Given the cubes: $A = X_1\ X_2\ X_3$, and $B = Y_1\ Y_2\ Y_3$, the resultant cubes are:

Group 1:

$cube_1$: $(X_1 \oplus Y_1)\ Y_2\ Y_3$, $cube_3$: $X_1\ (X_2 \oplus Y_2)\ Y_3$, $cube_5$: $X_1\ X_2\ (X_3 \oplus Y_3)$.

Group 2:

$cube_2$: $(X_1 \oplus Y_1)\ X_2\ X_3$, $cube_4$: $Y_1\ (X_2 \oplus Y_2)\ X_3$, $cube_6$: $Y_1\ Y_2\ (X_3 \oplus Y_3)$.

By performing the Distance 2 Exorlink operation on above $cube_1$ and $cube_3$, the cubes $cube_7$ and $cube_8$ are obtained:

$cube_7$: $Y_1\ (X_2 \oplus Y_2)\ Y_3$, $cube_8$: $(X_1 \oplus Y_1)\ X_2\ Y_3$.

Cubes $cube_7$, $cube_8$ and $cube_5$ form a new group of resultant cubes for Distance 3 Exorlink. By doing all the permutations, the new Distance 3 Exorlink operation generates the following 12 resultant cubes (note the new enumeration of cubes):

$cube_1$: $X_1\ X_2\ (X_3 \oplus Y_3)$, $cube_2$: $Y_1\ Y_2\ (X_3 \oplus Y_3)$, $cube_3$: $(X_1 \oplus Y_1)\ Y_2\ Y_3$,

$cube_4$: $(X_1 \oplus Y_1)\ X_2\ X_3$, $cube_5$: $X_1\ (X_2 \oplus Y_2)\ X_3$, $cube_6$: $Y_1\ (X_2 \oplus Y_2)\ Y_3$,

$cube_7$: $X_1\ (X_2 \oplus Y_2)\ Y_3$, $cube_8$: $(X_1 \oplus Y_1)\ X_2\ Y_3$, $cube_9$: $X_1\ Y_2\ (X_3 \oplus Y_3)$,

$cube_{10}$: $Y_1\ (X_2 \oplus Y_2)\ X_3$, $cube_{11}$: $(X_1 \oplus Y_1)\ Y_2\ X_3$,

$cube_{12}$: $Y_1\ X_2\ (X_3 \oplus Y_3)$.

which form 6 groups: Group 1: $cube_1$, $cube_3$, $cube_7$; Group 2: $cube_1$, $cube_6$, $cube_8$; Group 3: $cube_3$, $cube_5$, $cube_9$; Group 4: $cube_2$, $cube_4$, $cube_{10}$; Group 5: $cube_2$, $cube_5$, $cube_{11}$; and Group 6: $cube_4$, $cube_6$, $cube_{12}$. Concluding, after the Distance 3 Exorlink, there are as many as 6 groups to be selected from, instead of only 2 groups that were used in EXORCISM-MV-2.

IV. LOOK AHEAD STRATEGIES.

In this section, we define several look-ahead strategies for searching best ESOPs.

One step look ahead.

Give an array of cubes, $\{p_1, p_2, .., p_n\}$, if distance of p_1 and p_2 is 2, a Distance 2 Exorlink operation can be performed. After this operation, there are two groups, one of them to be selected. For each of the 4 resultant cubes, q_1, q_2, q_3 and q_4, the distance between q_i (i = 1,2,3,4) and p_j ($j \neq 1$ and $j \neq 2$) is checked. If it is found that there is a distance ≤ 1, the corresponding group of resultant cubes is selected, so the total number of cubes in the array can be reduced. This is called a *one-step-look-ahead* method.

Two step look ahead.

After one-step-look-ahead, if there are no Distance 0 or Distance 1 Exorlink operations possible, then if the distance of q_i and p_j is 2, the Distance 2 Exorlink can be performed again and 4 resultant cubes r_1, r_2, r_3 and r_4 are generated. Now it can be checked whether the distance of r_i $(i = 1, 2, 3, 4)$ and p_k $(k \neq 1, 2, j)$ is ≤ 1. This is called a *two-step-look-ahead* method.

Consequently, we can perform *multi-step-look-ahead.* Multi-step means 2 or more steps. Look-ahead means looking for Distance 0 and Distance 1 Exorlink operations. Look-ahead on Distance 3 Eexorlink can be also applied. The Distance 3 Exorlink increases the number of cubes by 1, the Distance 0 Exorlink will reduce the number of cubes by 2, the Distance 1 Exorlink will reduce the number of cubes by 1, and the Distance 2 Exorlink will keep the number of cubes unchanged. But Distance 2, and 3 Exorlinks will bring the ESOP expression to a different point, so they may also lead to better results of subsequent operations.

Multi-step look ahead with full backtracking.

Assume $distance(p_1,\ p_2)\ =\ 2$, and 4 resultant cubes q_1, q_2, q_3 and q_4 are generated. Assume p_{i1}, p_{i2}, ..., p_{im1} were found, there are m cubes which are of distance 2 with q_1. So for q_1, there are m_1 ways to go. Accordingly, there are m_2,m_3 and m_4 ways to go for q_2, q_3 and q_4 respectively. So, at this step, there are $m_1\ +\ m_2\ +\ m_3\ +\ m_4$ ways to go. For each of these ways, at the next step, there are $(n_1\ +\ n_2\ +\ n_3\ +\ n_4)$ ways to go. Let's assume the maximum search depth step is set to the value of K. When the K_th step is reached, the algorithm goes back one step and tries other ways. If all the possible ways at each step are tried, it is called a *full backtracking* method.

The searching space for a full backtracking is huge, when the value of K is big. (The searching space depends also on how many Distance-2 cubes can be found at each step, which in turn depends on the number of cubes currently in the array).

Multi-step look ahead with limited backtracking.

Since the full backtracking is very expensive, there are two alternatives to be investigated: the *small* K strategy, and the *limited backtracking* strategy. Small K means setting parameter K to a small value, like 2 or 3. Limited backtracking means at each step only a few ways to go are selected based on some method (random selection is used in our current implementation). Practically, we found experimentally that the method of large K with a small number of backtracks is much faster and usually gives better results than the method of small K with a large number of backtracks. Although the second one is also sometimes very helpful.

Concluding, there are two basic strategies:

1. small K and large number of backtracks.

2. large K and small number of backtracks.

Practically, we found that the first strategy leads to the very rapid reduction, and then the cost (number of PTs) remains stable for many loops (local minimum). While the second strategy leads to very slow reductions, but better reductions are found if the loop continues.

V. THE ALGORITHM OF EXORCISM-MV-3.

A. The loops.

The first method is used to obtain a local minimum quickly. Then the second method is used to get out of the local minimum. The main operations in our algorithm are distance checking and exorlink. Both operations have worst case complexity of n (number of variables), which is linear. Since these operations are performed in many loops, so the **run time** depends mainly on the number of loop repetitions, which relates to the number of cubes in the array (the more cubes, the more loop repetitions). The **stopping criteria** are the following: during the look ahead, we stop if we reach the number of loops, or complete the backtracking, or find the reduction.

Below are the loop strategies.

Distance 2 Look Ahead.

At each step, find if Distance 0 or 1 Exorlink is possible.

If yes, execute it and quit the look ahead.

else perform the Distance 2 Exorlink when possible, and move to the next step.

else if no Distance 2 Exorlink is possible, quit the look ahead.

Distance 2 Loop.

For each pair of cubes in the array,

if the distance is 0 or 1, perform Exorlink, quit the loop.

else if the distance is 2, perform Exorlink, generate q_1, q_2, q_3, q_4.

if (distance(q_i, p_j*)* $<$ *2) perform Exorlink, quit the loop.*

else pick one group of the resultant cubes, continue to loop.

The Distance 2 Loop is equivalent to N-step Look Ahead without backtracking (N is the number of Distance-2 pairs in the array).

Distance 2 Loop2. Distance 2 Loop2 is the same as the Distance 2 Loop, except that at each step, 3 groups of cubes are picked (the two resultant groups and the original group), which have the smallest literal count.

Distance 3 Look Ahead.

At each step, find if Distance 3 Exorlink is possible.

If Distance 3 Exorlink is not possible, quit the Distance 3 Look Ahead.

else, for each of the 12 resultant cubes, check if

Distance 0 or Distance 1 Exorlinks are possible.

If yes, execute the Exorlink and move to the next step.

else do not execute the Exorlink, looking for the next pair.

Distance 3 Loop1.

Perform an N step Distance-3 Look Ahead without backtracking.

Distance 3 Loop2.

Perform an N step Distance-3 Look Ahead with maximum M backtracking.

B. The algorithm of EXORCISM-MV-3.

(1) Distance 2 Loop.

(2) Distance 3 Loop1.

(3) Loop steps (1) and (2), until no improvements.

(4) Distance 3 Loop2.

(5) Loop steps (1) to (4), until no improvements.

(6) Distance 2 Loop2.

(7) If no improvements on the number of cubes, stop; else go back to step 1.

VI. EXPERIMENTAL RESULTS OF EXORCISM-MV-3.

We compared our new program, EXORCISM-MV-3 to our previous program, EXORCISM-MV-2. Both programs are run on the same machine.

We collect all the test cases from literature on ESOP minimization. By *small examples* we characterize those that satisfy the formula:

(number of input variables + number of output variables) $*$ Initial number of PTs $<$ 20,000

Table 1 presents comparison on small examples. Table 2 presents comparison on large examples. As seen in Table 2, speedup of up to 141 can be achieved on large benchmark functions. Results generated by ESPRESSO are also put into the same tables for comparison. Please note that the outputs of EXORCISM are ESOPs and the outputs of ESPRESSO are SOPs, although the definitions for their cost functions are the same. From the tables we can see that ESOP minimization is significantly slower than SOP minimization. This is partially due to the fact that ESOP minimization is a more difficult problem than its counterpart; partially due to the nature of our algorithm, which needs a large number of iterations.

VII. CONCLUSIONS.

One approach to minimize ESOPs is to apply a set of cube operations iteratively on each pair of cubes in the array [4, 8, 20, 28, 23, 24, 6]. The Exorlink operation is the most powerful operation in this approach, which can link any two cubes in an array of cubes of an arbitrary distance. The superiority of the new cube operation [28] ensures better results than the previous operations shown in the literature. Here, the application of the Exorlink operation was further improved by use of look-ahead ideas. Our program EXORCISM-MV-3 was tested on many benchmark functions, and compared to EXORCISM-MV-2. As EXORCISM-MV-2, new program EXORCISM-MV-3 is able to minimize efficiently incomplete multi-output functions with arbitrary number of values for each variable, which allows to run it with arbitrary input decoders (which is equivalent to AND/EXOR PLA decomposition to AND/EXOR PLA and several standard PLAs that realize the decoders).

By increasing the look-ahead step to infinity one can in theory obtain the exact ESOP. But increasing the step means increase of the searching space, and full backtracking means do all the permutation inside that space. The run time becomes exponential, but one can obtain exact solutions, as well as solutions as close to exact as desired by setting the values of K. In conclusion, this approach can be used to obtain exact solutions for smaller functions.

In many practical applications the ESOP minimization program must be very fast because it is used in a loop of another minimization program. The EXORCISM-MV-3 program presented here in most cases maintains or improves on the quality of EXORCISM-MV-2, but it is orders of magnitude faster on large benchmark functions.

As pointed in [10], further work is needed to minimize efficiently very weakly specified functions, such as those that have more than 95% don't cares and occur in the area of Machine Learning. It can be, however, observed, that the ideas of EXORCISM-DC for strongly unspecified functions from [10] can be also generalized to multi-output Boolean relations [12], and next combined with EXORCISM-MV-3 to create an efficient ESOP minimizer for binary and MV-input, binary-output relations.

We found also, that the look-ahead strategies can be used successfully to many other applications in logic minimization. In placement and routing algorithms, look-ahead is used for small K (usually 2 or 3) only. Our idea here is large K with small number of backtracks.

References

[1] Ph. W. Besslich and M. W. Riege, *An Efficient Program for Logic Synthesis of Mod-2 Sum Expressions*, Proc. EUROASIC, Paris, France, pp. 136-141, 1991.

[2] D. Brand and T. Sasao, *Minimization of AND-EXOR Expressions using rewrite rules*, IEEE Trans. on Comput., Vol. C42, no. 5, pp. 568-576, May 1993.

[3] L. Csanky, M. A. Perkowski and I. Schäfer, *Canonical Restricted Mixed-Polarity Exclusive Sums of Products and the Efficient Algorithm for Their Minimization*, Proc. IEEE Intern. Symp. on Circuits and Systems, San Diego, pp. 17-20, May 1992.

[4] M. Helliwell and M. A. Perkowski, *A Fast Algorithm to Minimize Multi-Output Mixed-Polarity Generalized Reed-Muller Forms*, Proc. 25th ACM/IEEE Design Automation Conference, Anaheim, CA, pp. 427-432, June 1988.

[5] T. Hirayama, and Y. Nishitani, *A Simplification Algorithm of AND-EXOR Expressions for Multiple-Output Functions,* Proc. of the Second Workshop on Applications of Reed-Muller Expansion in Circuit Design, Chiba City, Japan, pp. 88-93, August 1995.

[6] T. Kozłowski, *Heuristic Minimisation of ESOPs*, Reed-Muller Colloquium UK'95, pp. 4/1-4/5, 1995.

[7] T. Kozłowski, *Applications of Exclusive-OR logic in Technology Independent Logic Optimization,* Ph.D. Thesis, Dept. of Electr. and Electronic Engn., Univ. of Bristol, U.K., January 1996.

[8] M. A. Perkowski, M. Helliwell and P. Wu, *Minimization of Multiple-Valued Input Multi-Output Mixed-Radix Exclusive Sums of Products for Incompletely Specified Boolean Functions*, Proc. 19th ISMVL, pp. 256-263, May 1989.

[9] M. A. Perkowski and M. Chrzanowska-Jeske, *An Exact Algorithm to Minimize Mixed-Radix Exclusive Sums of Products for Incompletely Specified Boolean Functions*, Proc. of the ISCAS'90, International Symposium on Circuits and Systems, New Orleans, pp. 1652-1655, May 1990.

[10] M. A. Perkowski, T. Ross, D. Gadd, J.A. Goldman, and N. Song, *Application of ESOP Minimization in Machine Learning and Knowledge Discovery,* Proc. of the Second Workshop on Applications of Reed-Muller Expansion in Circuit Design, (RM'95), Chiba City, Japan, pp. 102-109, August 1995.

[11] M. Perkowski, A.Sarabi, and F. Beyl, *Fundamental Theorems and Families of Forms for Binary and Multiple-Valued Linearly Independent Logic*, Proc. RM'95, pp. 288-299, August 1995.

[12] M. Perkowski, M. Marek-Sadowska, L. Jóźwiak, T. Łuba, S. Grygiel, M. Nowicka, R. Malvi, Z. Wang, and J. Zhang, *Decomposition of Multi-Valued Relations,* Proc. ISMVL'97, Nova Scotia, pp. 13 - 18, May 1997.

[13] M. Perkowski, L. Jóźwiak, R. Drechsler, and B. Falkowski, *Ordered and Shared, Linearly Independent, Variable-Pair Decision Diagrams,* Proc. 1st Intern. Conf. on Information, Communication and Signal Processing, ICICS'97, Singapure, September 1997.

[14] M. Perkowski, L. Jóźwiak, and R. Drechsler, *A Canonical AND/EXOR Form that includes both the Generalized Reed-Muller Forms and Kronecker Forms,* Proc. RM'97, September 1997.

[15] M.A. Perkowski, M. Chrzanowska-Jeske, and Y. Xu, *Lattice Diagrams Using Reed-Muller Logic*, Proc. RM'97, September 1997.

[16] M.A. Perkowski, E. Pierzchała, and R. Drechsler, *Layout-Driven Synthesis for Submicron Technology: Mapping Expansions to Regular Lattices,* Proc. 7th Intern. Symp. on IC Technology, Systems, and Applications, ISIC-97, Singapure, September 1997.

[17] M.A. Perkowski, L. Jóźwiak, R. Drechsler, *New Hierarchies of Generalized Kronecker Trees, Forms, Decision Diagrams, and Regular Layouts,* Proc. RM'97, September 1997.

[18] T. Sasao, *EXMIN: A Simplification Algorithm for Exclusive-OR-Sum-of-Products Expressions for Multiple-Valued Input Two-Valued Output Functions*, 20th Int. Symp. on Multiple-Valued Logic, pp. 128-135, May 1990.

[19] T. Sasao and Ph. W. Besslich, *On the Complexity of Mod-2 Sum PLAs*, IEEE Trans. on Comput., Vol 39, no. 2, pp. 262-266, February 1990.

[20] T. Sasao, *EXMIN2: A Simplification Algorithm for Exclusive-OR-Sum-of-Products Expressions for Multiple-Valued-Input Two-Valued-*

Output Functions, IEEE Trans. on CAD, Vol. 12, no. 5, pp. 621-632, May 1993.

[21] T. Sasao, *An Exact Minimization of AND-EXOR Expressions Using BDDs*, Proc. RM'93, September 1993.

[22] T. Sasao, *An Upper Bound on the Number of Products in Minimium ESOPs*, Proc. RM'95, pp. 94-101, September 1995.

[23] J. M. Saul, *An Improved Algorithm for the Minimization of Mixed Polarity Reed-Muller Representations*, Int. Conf. on Comput. Design: VLSI in Comput. and Processors, pp. 372-375, September 1990.

[24] J. M. Saul, *An Efficient Data Structure for the Minimization of Exor Sums*, Proc. RM'95, pp. 116-122, September 1995.

[25] I. Schäfer and M. A. Perkowski, *Multiple Valued Generalized Reed-Muller Forms*, Proc IEEE 21st Int. Symp. on Multiple-Valued Logic, pp. 40-48, May 1991.

[26] N. Song, *Minimization of Exclusive Sum of Products Expressions for Multiple-Valued Input Incompletely Specified Functions*, Master thesis, EE Dept. Portland State University, Portland, OR, 1992.

[27] N. Song, M. A. Perkowski, M. Chrzanowska-Jeske, A. Sarabi, *A New Design Methodology for Two-Dimensional Logic Arrays,* VLSI Design, Vol. 3., Nos. 3-4, pp. 315-332, 1995.

[28] N. Song, M. Perkowski, *Minimization of Exclusive Sum of Products Expressions for Multi-Output Multiple-Valued Input, Incompletely Specified Functions,* IEEE Transactions on Computer Aided Design, Vol. 15, No. 4, pp. 385-395, April 1996.

[29] C. Tsai and M. Marek-Sadowska, *Multilevel Logic Synthesis for Arithmetic Functions,* Proc. DAC'96, pp. 242-247, June 1996.

[30] X. Wu, X. Chen and S. L. Hurst, *Mapping of Reed-Muller Coefficients and the Minimization of Exclusive-OR Switching Functions*, IEE Proc. Pt. E, vol. 129, no. 1, pp. 5-20, January 1982.

[31] A. Zakrevskij, *Looking for Shortest Solutions of Systems of Linear Logical Equations: Theory and Applications in Logic Design*, Proc. of 2-nd Workshop Boolesche Probleme, (B. Steinbach, Ed.), Freiberg/Sachsen, Germany, pp. 63-69, September 1996.

[32] X. Zeng, M. Perkowski, K. Dill, A. Sarabi, *Approximate Minimization of Generalized Reed-Muller Forms,* Proc. RM'95, pp. 221-230, September 1995.

Table 1: Comparison on Small Examples

			EXORCISM-MV-3			EXORCISM-MV-2			ESPRESSO		
	Inputs	Outputs	C_T	C_L	Time	C_T	C_L	Time	C_T	C_L	Time
con1	7	2	9	37	0.1	9	37	0.0	9	32	0.0
misex1	8	7	12	82	0.2	12	85	0.2	12	96	0.01
xor5	5	1	5	10	0.1	5	10	0.0	16	96	0.0
bw	5	28	22	319	1.1	22	300	0.6	22	349	0.20
squar5	5	8	19	87	0.8	19	92	0.3	25	119	0.04
misex2	25	8	27	210	1.5	27	210	2.5	28	213	0.03
rd53	5	3	14	57	0.4	14	57	0.3	31	175	0.01
inc	7	9	26	176	1.7	26	176	1.7	30	198	0.05
rot8	8	5	35	257	3.5	35	257	3.9	57	385	0.19
5xp1	7	10	32	170	2.2	33	182	1.5	65	347	0.09
rdm8	8	8	31	157	1.9	31	157	1.8	77	399	0.11
f51m	8	8	31	156	2.2	31	160	3.4	77	400	0.36
sao2	10	4	28	288	2.6	28	285	1.5	58	496	0.07
b12	15	9	28	169	1.2	28	163	1.6	43	207	0.25
adr4	8	5	31	144	1.7	31	150	1.8	69	383	0.08
rd73	7	4	35	188	4.2	38	186	3.0	127	903	0.15
nrm4	8	5	67	501	12.0	67	503	18.3	123	877	0.32
9sym	9	1	51	426	4.3	51	423	7	86	602	0.18
clip	9	5	63	472	7.4	64	497	8.1	120	793	0.52
mlp4	8	8	60	395	16.1	61	389	10.4	128	891	0.74
log8	8	8	83	615	48.7	89	675	29.7	131	990	0.79
wgt8	8	4	56	306	29.6	55	307	14.4	255	2070	0.61
rd84	8	4	59	322	20.2	57	317	19.9	255	2070	0.56
t481	16	1	13	53	50.1	13	53	41.2	481	5233	1.22
Total	212	155	837	5597	213.8	846	5671	173.1	2325	18324	6.59

Table 2: Comparison on Large Examples

			EXORCISM-MV-3			EXORCISM-MV-2			ESPRESSO		
	Inputs	Outputs	C_T	C_L	Time	C_T	C_L	Time	C_T	C_L	Time
e64	65	65	65	2272	12.4	65	2210	78.7	65	2210	0.26
duke2	22	29	78	909	11.8	79	920	72.3	86	996	0.38
ex5	8	63	72	704	10.4	72	920	61.8	74	1903	0.88
ex4	128	28	316	3140	121.4	316	3136	17122.2	279	1928	6.67
table5	17	15	156	2449	32.6	156	2453	149.4	158	2501	1.07
table3	14	14	166	2491	24.2	166	2491	189.7	175	2644	1.68
cps	24	109	135	2625	37.4	135	2462	467.0	163	2836	5.08
vg2	25	8	184	1988	29.4	184	1993	224.7	110	914	0.22
pdc	16	40	184	1646	56.1	187	1678	455.9	145	1432	22.03
sqr8	8	16	106	704	50.1	108	703	174.2	188	1419	1.64
apex3	54	50	268	4155	53.2	270	3916	2562.9	280	3292	5.12
add6	12	7	127	800	24.4	127	819	104.6	355	2551	1.41
seq	41	35	248	4822	77.0	245	4833	2996.5	336	6245	10.45
spla	16	46	262	3393	82.6	260	3420	1223.4	254	3208	2.29
apex4	9	19	439	6181	124.2	438	6292	7556.6	436	5419	10.82
apex1	45	45	286	3820	108.9	285	3796	4696.8	206	2842	2.47
misex3c	14	14	226	2134	53.4	227	2123	1397.2	197	1561	4.81
ex1010	10	10	670	7466	1141.5	695	7928	16506.3	302	2895	5.87
alu4	14	8	422	4430	506.3	447	4816	6827.9	575	5087	11.09
misex3	14	14	535	6632	669.8	545	6837	18896.5	690	7784	17.39
apex5	117	88	399	4027	457.6	400	4038	20156.8	1088	7281	59.35
Total	673	723	5344	66788	3684.7	5407	67784	101921	5063	56875	159.77