Journal of Circuits, Systems, and Computers Vol. 9, Nos. 1 & 2 (1999) 83–97

EXOR DECOMPOSITION WITH COMMON VARIABLES AND ITS APPLICATION TO MULTIPLE-OUTPUT NETWORKS

RYOJI ISHIKAWA, GORO KODA and KENSUKE SHIMIZU
Department of Computer Science, Gunma University
1-5-1 Tenjin-cho Kiryu-shi Gunma 376-8515, Japan

TAKASHI HIRAYAMA
Ashikaga Institute of Technology
268-1 Omae-cho Ashikaga-shi Tochigi 326-8558 , Japan

Received 5 January 1999
Accepted 17 May 1999

This paper presents an EXOR decomposition with common variable sets, which is an attempt to decompose an n-variable logic function f into two $(n-1)$-variable subfunctions g_0 and g_1 by using exclusive-or (EXOR) operation. We present the conditions and formulas for the decomposition. We also consider the subfunction-sharing method for multiple-output networks as an application of the EXOR decomposition. Experimental results show that the area of networks is often reduced effectively by sharing the subfunctions obtained by the decomposition.

1. Introduction

Functional decomposition[1,7,9] is a basic technique in logic synthesis. Nowadays, exclusive-OR (EXOR) decompositions[3,4,8] are being studied because the use of EXOR gates often yields more compact networks[6]. Matsunaga[3] and Sasao-Butler[8] considered the EXOR decomposition with disjoint sets of variables such that $f(X_n) = g(X_g) \oplus h(X_h)$ ($X_n = \{x_1, x_2, \cdots, x_n\}$, $X_g \cup X_h = X_n$, $X_g \cap X_h = \emptyset$) (Fig. 1). Although the subfunctions $g(X_g)$ and $h(X_h)$ will be relatively small, the decomposition is applicable to a considerably restricted class of functions only. To ease the restriction, they also presented the non-disjoint EXOR decomposition with a common variable x; $f(X_n) = g(X_g, x) \oplus h(X_h, x)$, where X_g and X_h are disjoint sets of variables and x is disjoint from X_g and X_h. However, these subfunctions $g(X_g)$, $h(X_h)$, $g(X_g, x)$, or $h(X_h, x)$ have not been formulated, since the variable sets X_g or X_h are not specified concretely.

In this paper, we propose an EXOR decomposition with common variable sets (Fig. 2), i.e., $f(X_n) = g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1})$, where $X_{n-2} = \{x_1, x_2, \cdots,$

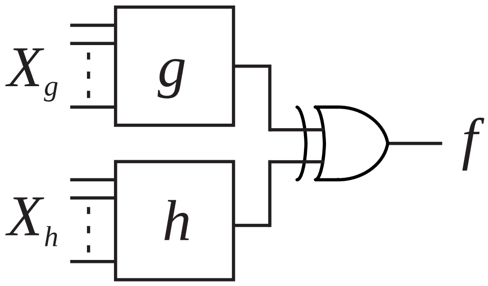

Fig. 1. Disjoint EXOR decomposition.

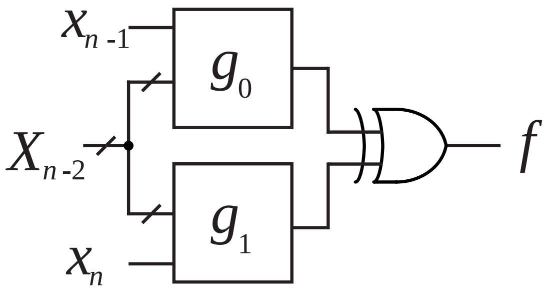

Fig. 2. EXOR decomposition with common variables.

$x_{n-2}\}$. It means that a function $f(X_n)$ is decomposed into two subfunctions with $(n-1)$ variables, $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$. Since $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ have $(n-2)$ common variables X_{n-2}, the decomposition is applicable to a larger class of functions. In this paper, the conditions to decompose functions are presented and $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ are formulated. By generalizing the decomposition, we also present the multi-decompositions such that

$$f(X_n) = \bigoplus_{0 \le i \le m-1} g_i(X_{n-m}, x_{n-i}),$$

where $X_{n-m} = \{x_1, x_2, \cdots, x_{n-m}\}$. It means that the function $f(X_n)$ is decomposed into m subfunctions with $(n-m+1)$ variables.

We also consider the subfunction-sharing method for multiple-output networks as an application of our decompositions. Experimental results over MCNC benchmarks show that our EXOR decompositions often reduces the area of networks effectively.

2. EXOR Bi-Decomposition

In this section, an EXOR bi-decomposition such that $f(X) = g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1})$ is considered.

Definition 1. *X_n represents the set of n variables $\{x_1, x_2, \cdots, x_n\}$.*

Definition 2. *For an n-variable function $f(X_n)$ and the variable x_n, cofactors of $f(X_n)$ with $x_n = 0$ and $x_n = 1$ are denoted by $f(X_{n-1}, 0)$ and $f(X_{n-1}, 1)$, respectively. Let $f(X_{n-1}, 2)$ be $f(X_{n-1}, 0) \oplus f(X_{n-1}, 1)$.*

Generally $f(X_{n-i}, 0, a_{n-i+2}, a_{n-i+3}, \cdots, a_n)$ and $f(X_{n-i}, 1, a_{n-i+2}, a_{n-i+3}, \cdots, a_n)$ are defined as the cofactors of $f(X_{n-i+1}, a_{n-i+2}, a_{n-i+3}, \cdots, a_n)$ with $x_{n-i+1} = 0$ and $x_{n-i+1} = 1$, respectively, where $a_{n-i+2}, a_{n-i+3}, \cdots, a_n$ are constants in $\{0, 1, 2\}$. $f(X_{n-i}, 2, a_{n-i+2}, a_{n-i+3}, \cdots, a_n)$ is defined as $f(X_{n-i}, 0, a_{n-i+2}, a_{n-i+3}, \cdots, a_n) \oplus f(X_{n-i}, 1, a_{n-i+2}, a_{n-i+3}, \cdots, a_n)$.

Definition 3. *Literals x, $\bar{x}$, and 1 of a variable x are represented by x^2, x^1, and x^0, respectively.*

An arbitrary function can be expanded uniquely without using negative literals as follows:

Theorem 1 (Davio expansion).

$$\begin{aligned} f(X_n) &= f(X_{n-1},0) \oplus x_n f(X_{n-1},2) \\ &= x_n^0 f(X_{n-1},0) \oplus x_n^2 f(X_{n-1},2) \end{aligned} \quad (1)$$

This is referred to as the *Davio expansion* [2,6] *with respect to* x_n.

By applying the expansion to the cofactors $f(X_{n-1},0)$ and $f(X_{n-1},2)$ with respect to the variable x_{n-1} recursively, we have

$$\begin{aligned} f(X_n) &= f(X_{n-2},0,0) \oplus x_{n-1} f(X_{n-2},2,0) \\ &\quad \oplus x_n f(X_{n-2},0,2) \oplus x_n x_{n-1} f(X_{n-2},2,2) \end{aligned} \quad (2)$$

Eq. (2) is called the *Davio expansion with respect to* $\{x_{n-1}, x_n\}$.

From the four cofactors in Eq. (2), the conditions to decompose a function $f(X_n)$ into certain subfunctions $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ are derived as the following lemma.

Lemma 1. *A function* $f(X_n)$ *can be decomposed as* $f(X_n) = g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1})$ *if and only if all of the following equations hold.*

$$\begin{cases} f(X_{n-2},0,0) = g_0(X_{n-2},0) \oplus g_1(X_{n-2},0) \\ f(X_{n-2},2,0) = g_1(X_{n-2},2) \\ f(X_{n-2},0,2) = g_0(X_{n-2},2) \\ f(X_{n-2},2,2) = 0 \end{cases}$$

Proof: By applying the Davio expansion to $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ with respect to the variables x_n and x_{n-1}, respectively, we have

$$\begin{aligned} g_0(X_{n-2}, x_n) &= g_0(X_{n-2},0) \oplus x_n g_0(X_{n-2},2) \\ g_1(X_{n-2}, x_{n-1}) &= g_1(X_{n-2},0) \oplus x_{n-1} g_1(X_{n-2},2). \end{aligned}$$

Hence

$$\begin{aligned} & g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1}) = \\ & \quad g_0(X_{n-2},0) \oplus g_1(X_{n-2},0) \oplus x_{n-1} g_1(X_{n-2},2) \oplus x_n g_0(X_{n-2},2). \end{aligned} \quad (3)$$

Since the right side of Eq. (3) has no negative literals and is represented by $(n-2)$-variable functions, it is the Davio expansion with respect to $\{x_{n-1}, x_n\}$. As mentioned before, cofactors of the Davio expansion are uniquely defined. Therefore, by comparing Eq. (2) with Eq. (3), we have the above lemma. □

From Lemma 1, a function $f(X_n)$ is decomposable if $f(X_{n-2}, 2, 2) = 0$. And $g_0(X_{n-2}, 2)$ and $g_1(X_{n-2}, 2)$ are uniquely defined by the cofactors of $f(X_n)$. $g_0(X_{n-2}, 0)$ and $g_1(X_{n-2}, 0)$ are not unique. However, if $g_0(X_{n-2}, 0)$ is specified, $g_1(X_{n-2}, 0)$ is determined as $g_1(X_{n-2}, 0) = f(X_{n-2}, 0, 0) \oplus g_0(X_{n-2}, 0)$. From the above argument, Lemma 1 can be rewritten as follows.

Theorem 2. *Let $f(X_n)$ be an n-variable function. If $f(X_{n-2}, 2, 2) = 0$, $f(X_n)$ has the EXOR decomposition such that $f(X_n) = g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1})$, where*

$$\begin{aligned} g_0(X_{n-2}, x_n) &= g_0'(X_{n-2}) \oplus x_n f(X_{n-2}, 0, 2) \\ g_1(X_{n-2}, x_{n-1}) &= f(X_{n-2}, 0, 0) \oplus g_0'(X_{n-2}) \oplus x_{n-1} f(X_{n-2}, 2, 0). \end{aligned}$$

$g_0'(X_{n-2})$ in the above equations is an arbitrary $(n-2)$-variable function.

In Theorem 2, the conditions to decompose $f(X_n)$ are presented and the subfunctions $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ are formulated. Since $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$ have $(n-2)$ common variables X_{n-2}, these subfunctions can not expect becoming compact circuits than those of the disjoint bi-decomposition[3,8]. Our decomposition, however, can be applied to larger classes of functions.

3. EXOR Multi-Decompositions

Similar to the EXOR bi-decomposition in Section , the EXOR multi-decompositions can be obtained; a function $f(X_n)$ is decomposed into m subfunctions with $(n-m+1)$ variables, where m is a constant.

To consider the EXOR multi-decompositions, we give the general form of Eqs. (1) and (2) by using $(n-m)$-variable functions.

Corollary 1. *An arbitrary function $f(X_n)$ can be expanded uniquely without using negative literals as*

$$\begin{aligned} f(X_n) = \bigoplus_{(c_{n-m+1}, c_{n-m+2}, \cdots, c_n)} & (x_{n-m+1}^{c_{n-m+1}} x_{n-m+2}^{c_{n-m+2}} \cdots x_n^{c_n} \\ & \cdot f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n)) \end{aligned} \quad (4)$$

where $c_{n-i} \in \{0, 2\}$ $(0 \leq i \leq m-1)$.

Eq. (4) is called the *Davio expansion with respect to* $\{x_{n-m+1}, x_{n-m+2}, \cdots, x_n\}$ because it can be obtained by applying Eq. (1) recursively with respect to these variables.

From the cofactors $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n)$ in Eq. (4), the conditions to decompose a function $f(X_n)$ into m subfunctions $g_0(X_{n-m}, x_n)$, $g_1(X_{n-m}, x_{n-1})$, $\cdots$, $g_{m-1}(X_{n-m}, x_{n-m+1})$ with $(n-m+1)$ variables are derived as the following lemma.

Lemma 2. *Let m be a constant such that $2 \leq m \leq n$. A function $f(X_n)$ can be decomposed as $f(X_n) = \bigoplus_{0 \leq i \leq m-1} g_i(X_{n-m}, x_{n-i})$ if and only if all of the following equations hold, where $c_{n-i} \in \{0, 2\}$ $(0 \leq i \leq m-1)$.*

$$f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n) = \begin{cases} 0 & (\sum_{0 \leq i \leq m-1} c_{n-i} \geq 4) \\ \bigoplus_{0 \leq i \leq m-1} g_i(X_{n-m}, 0) & (c_{n-m+1} = c_{n-m+2} = \cdots = c_n = 0) \\ g_0(X_{n-m}, 2) & (c_n = 2, c_{n-i} = 0 \ (i \neq 0)) \\ g_1(X_{n-m}, 2) & (c_{n-1} = 2, c_{n-i} = 0 \ (i \neq 1)) \\ \vdots & \\ g_{m-1}(X_{n-m}, 2) & (c_{n-m+1} = 2, c_{n-i} = 0 \ (i \neq m-1)) \end{cases}$$

Proof: By applying the Davio expansion to $g_0(X_{n-m}, x_n), g_1(X_{n-m}, x_{n-1}), \cdots,$ $g_{m-1}(X_{n-m}, x_{n-m+1})$ with respect to the variables $x_n, x_{n-1}, \cdots, x_{n-m+1}$, respectively, we have

$$\bigoplus_{0 \leq i \leq m-1} g_i(X_{n-m}, x_{n-i}) = \bigoplus_{0 \leq i \leq m-1} (g_i(X_{n-m}, 0) \oplus x_{n-i} g_i(X_{n-m}, 2)) \qquad (5)$$

Since the right side of Eq. (5) has no negative literals and is represented by $(n-m)$-variable functions, it is the Davio expansion with respect to $\{x_{n-m+1}, x_{n-m+2}, \cdots, x_n\}$. As mentioned before, cofactors of the Davio expansion are uniquely defined. Therefore, by comparing the cofactors in Eq. (4) with the subfunctions in Eq. (5), we have the above lemma. □

From Lemma 2, a function $f(X_n)$ is decomposable if $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n) = 0$ holds for all $c_{n-m+1}, c_{n-m+2}, \cdots, c_n$ $(c_{n-i} \in \{0, 2\})$ under the condition

$\sum_{0\leq i\leq m-1} c_{n-i} \geq 4$. And $g_i(X_{n-m}, 2)$ $(0 \leq i \leq m-1)$ are uniquely defined by the cofactors of $f(X_n)$. $g_i(X_{n-m}, 0)$ $(0 \leq i \leq m-1)$ are not unique. However, if $g_i(X_{n-m}, 0)$ $(0 \leq i \leq m-2)$ are specified, $g_{m-1}(X_{n-m}, 0)$ is determined as $g_{m-1}(X_{n-m}, 0) = f(X_{n-m}, 0, 0, \cdots, 0) \oplus \bigoplus_{0\leq i\leq m-2} g_i(X_{n-m}, 0)$. From the above argument, Lemma 2 can be rewritten as follows.

Theorem 3. *Let $f(X_n)$ be a function and m be a constant $(2 \leq m \leq n)$. If $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n) = 0$ holds for all c_{n-m+1}, c_{n-m+2}, $\cdots$, c_n $(c_{n-i} \in \{0, 2\})$ under the condition $\sum_{0\leq i\leq m-1} c_{n-i} \geq 4$, $f(X_n)$ has the EXOR decompositions such that $f(X_n) = \bigoplus_{0\leq i\leq m-1} g_i(X_{n-m}, x_{n-i})$, where*

$$\begin{aligned}
g_0(X_{n-m}, x_n) &= g'_0(X_{n-m}) \oplus x_n f(X_{n-m}, 0, 0, \cdots, 0, 0, 2) \\
g_1(X_{n-m}, x_{n-1}) &= g'_1(X_{n-m}) \oplus x_{n-1} f(X_{n-m}, 0, 0, \cdots, 0, 2, 0) \\
&\vdots \\
g_{m-2}(X_{n-m}, x_{n-m+2}) &= g'_{m-2}(X_{n-m}) \oplus x_{n-m+2} f(X_{n-m}, 0, 2, 0, \cdots, 0, 0) \\
g_{m-1}(X_{n-m}, x_{n-m+1}) &= f(X_{n-m}, 0, 0, \cdots, 0) \oplus \bigoplus_{0\leq i\leq m-2} g'_i(X_{n-m}) \\
&\quad \oplus x_{n-m+1} f(X_{n-m}, 2, 0, 0, \cdots, 0, 0).
\end{aligned}$$

$g'_0(X_{n-m})$, $g'_1(X_{n-m})$, $\cdots$, $g'_{m-2}(X_{n-m})$ in the above equations are arbitrary $(n-m)$-variable functions.

In Theorem 3, the conditions to decompose $f(X_n)$ are presented and the subfunctions $g_0(X_{n-m}, x_n)$, $g_1(X_{n-m}, x_{n-1})$, $\cdots$, $g_{m-1}(X_{n-m}, x_{n-m+1})$ are formulated. These m subfunctions have $(n-m)$ common variables $X_{n-m}(= \{x_1, x_2, \cdots, x_{n-m}\})$. Theorem 3 is a generalization of Theorem 2.

4. Number of Decomposable Functions

In this section, we discuss the number of decomposable functions. It is derived as the following theorem.

Theorem 4. *Let N be the number of decomposable functions with $(n-m)$ common variables in all n-variable functions. Then, the following equation holds.*

$$2^{(m+1)\cdot 2^{n-m}} \leq N \leq \binom{n}{m} 2^{(m+1)\cdot 2^{n-m}}$$

Proof: **(i)** From Corollary 1, a function $f(X_n)$ is specified uniquely by cofactors $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n)$, where $c_{n-i} \in \{0, 2\}$ $(0 \leq i \leq m-1)$. As mentioned in Theorem 3, the condition to decompose a function $f(X_n)$ is that $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n) = 0$ holds for all c_{n-m+1}, c_{n-m+2}, $\cdots$, c_n $(c_{n-i} \in \{0, 2\})$ such that $\sum_{0 \leq i \leq m-1} c_{n-i} \geq 4$. Therefore the rest of the cofactors of a decomposable function $f(X_n)$ have no restrictions, that is, $f(X_{n-m}, c_{n-m+1}, c_{n-m+2}, \cdots, c_n)$'s such that $\sum_{0 \leq i \leq m-1} c_{n-i} \leq 2$ are arbitrary. Since there are $(m+1)$ assignments for $c_{n-m+1}, c_{n-m+2}, \cdots, c_n$ in this case, a decomposable function is specified by $(m+1)$ cofactors. These cofactors are $(n-m)$-variable functions. Since the total number of $(n-m)$-variable functions is $2^{2^{n-m}}$, the number of decomposable functions is

$$2^{(m+1) \cdot 2^{n-m}}.$$

This is considered only in the case where $X_{n-m}(= \{x_1, x_2, \cdots, x_{n-m}\})$ is chosen as $(n-m)$ common variables to decompose a function.

(ii) Among functions that are not decomposable with X_{n-m}, there are some decomposable ones with another combination of $(n-m)$ common variables. We also consider such cases. Since there are $\binom{n}{m}$ combinations of $(n-m)$ variables, the number of decomposable functions is at most

$$\binom{n}{m} 2^{(m+1) \cdot 2^{n-m}}.$$

This is an upper bound because there exist some functions that are decomposable with more than one combinations of common variables. □

The number of all the n-variable functions is 2^{2^n}. Compared with 2^{2^n}, our decompositions are applicable to restricted classes of functions. However, in Section , we can show that many practical functions are decomposable.

5. Subfunction Sharing

In this section, subfunction sharing for multiple-output networks based on the EXOR bi-decomposition is described. A k-output network is regarded as a set of k functions. Subfunction sharing for these k functions is an effective technique for the compact logic synthesis.

Among the k functions forming a network, let us assume that there are some functions that have the EXOR decomposition of Theorem 2. Let $f_0(X_n)$ and $f_1(X_n)$ be two of such functions, and be decomposed as

$$f_0(X_n) \quad = \quad g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1}) \qquad (6)$$

$$f_1(X_n) = h_0(X_{n-2}, x_n) \oplus h_1(X_{n-2}, x_{n-1}), \tag{7}$$

respectively. From Theorem 2, the above subfunctions $g_0(X_{n-2}, x_n)$, $g_1(X_{n-2}, x_{n-1})$, $h_0(X_{n-2}, x_n)$, and $h_1(X_{n-2}, x_{n-1})$ are written as

$$g_0(X_{n-2}, x_n) = g_0'(X_{n-2}) \oplus x_n f_0(X_{n-2}, 0, 2) \tag{8}$$

$$g_1(X_{n-2}, x_{n-1}) = f_0(X_{n-2}, 0, 0) \oplus g_0'(X_{n-2}) \oplus x_{n-1} f_0(X_{n-2}, 2, 0) \tag{9}$$

$$h_0(X_{n-2}, x_n) = h_0'(X_{n-2}) \oplus x_n f_1(X_{n-2}, 0, 2) \tag{10}$$

$$h_1(X_{n-2}, x_{n-1}) = f_1(X_{n-2}, 0, 0) \oplus h_0'(X_{n-2}) \oplus x_{n-1} f_1(X_{n-2}, 2, 0). \tag{11}$$

Since both $g_0'(X_{n-2})$ and $h_0'(X_{n-2})$ in Eqs. (8)–(11) are arbitrary functions with $(n-2)$ variables, two $(n-1)$-variable subfunctions may be sharable.

Theorem 5. *Let $f_0(X_n)$ and $f_1(X_n)$ be two functions which have the EXOR bi-decompositions represented by Eqs. (6)–(11). Then $g_0(X_{n-2}, x_n)$ is sharable with $h_0(X_{n-2}, x_n)$ if Condition 1 is satisfied. Similarly $g_1(X_{n-2}, x_{n-1})$ is sharable with $h_1(X_{n-2}, x_{n-1})$ if Condition 2 is satisfied.*

Condition 1: $f_0(X_{n-2}, 0, 2) = f_1(X_{n-2}, 0, 2)$
Condition 2: $f_0(X_{n-2}, 2, 0) = f_1(X_{n-2}, 2, 0)$

Proof: If Condition 1 is satisfied, letting $g_0'(X_{n-2}) = h_0'(X_{n-2})$ leads to $g_0(X_{n-2}, x_n) = h_0(X_{n-2}, x_n)$.

If Condition 2 is satisfied, letting $g_0'(X_{n-2}) = h_0'(X_{n-2}) \oplus f_0(X_{n-2}, 0, 0) \oplus f_1(X_{n-2}, 0, 0)$ leads to $g_1(X_{n-2}, x_{n-1}) = h_1(X_{n-2}, x_{n-1})$. □

On the other hand, the disjoint bi-decompositions are not suitable for subfunction sharing because the input variables of decomposed subfunctions generally do not match each other.

If $f_0(X_n)$ and $f_1(X_n)$ are randomly selected functions, Condition 1 or 2 are rarely satisfied. However, functions forming a multiple-output network usually have correlations and often satisfy these conditions. Theorem 5 is thus applicable to multiple-output networks.

6. Experimental Results

Table 1 shows the number of decomposable functions with 3 and 4 variables, and the computing time to check their decomposability for each function on Sun Ultra 30 (CPU: Ultra SPARC-II 250MHz, 10.0 SPECint95). In the table, we count the number of functions which have at least one decomposition among all the combinations of $(n-m)$ common variables. The values in parentheses refer to the computing time (in seconds). The decomposition of $m = 2$ is applicable to more functions than the disjoint bi-decomposition[3,8].

Table 1. Number of functions with $n = 3$ and 4 which have the EXOR decompositions

# of var.	All func.	Disjoint[3,8]	$m = 2$	$m = 3$	$m = 4$
$n = 3$	256	26	112 (5.90×10^{-5} s)	16 (4.33×10^{-5} s)	
$n = 4$	65536	914	15328 (2.45×10^{-4} s)	736 (2.16×10^{-4} s)	32 (1.33×10^{-4} s)

We also apply the decompositions of $m = 2, 3$, and 4 to MCNC benchmark networks[5]. We regard a k-output network as a set of k functions and count the number of decomposable functions in the set. Among 45 networks experimented, 41 networks have decomposable functions, which are shown in Table 2, and four networks have no decomposable functions, which are 9sym, clip, cm42a, and sao2. In the table, the values in parentheses refer to the computing time (in seconds). The results indicate that our decompositions are applicable to most networks.

Next, we count the number of functions which have decompositions with a same common variable set in all the functions forming a multiple-output network. In this experiment, the same set of variables is used as the common variables throughout the decompositions. Because k functions forming a multiple-output network must have the same variable sets to use Condition 1,2 in Theorem 5. Table 3 shows the maximum number of decomposable functions with the same variable sets to all functions. Since Table 2 is not very different from Table 3, our decompositions with same common variables are applicable to most networks.

Then we made experiments in subfunction sharing of Theorem 5. By using the following algorithm, we observe C_s, C_d, F_{max}, and F_{2d} for the networks in Table 2 which have more than or equal to one decomposable functions. In the algorithm, $\mathcal{F}$ denotes the set of the functions forming a multiple-output network.

Table 2. Number of decomposable functions of benchmark networks

Name	In	Out	$m = 2$	$m = 3$	$m = 4$
5xp1	7	10	7 (0.07s)	6 (0.08s)	5 (0.08s)
alu4	14	8	7 (0.41s)	5 (0.82s)	2 (1.26s)
apex1	45	45	45 (0.41s)	45 (0.50s)	45 (0.89s)
apex3	54	50	50 (0.15s)	50 (0.17s)	50 (0.18s)
apex4	9	19	1 (0.42s)	1 (0.61s)	1 (0.73s)
apex5	117	88	88 (0.66s)	88 (1.91s)	88 (3.71s)
apex6	135	99	99 (0.07s)	99 (0.10s)	99 (0.42s)
apex7	49	37	37 (0.07s)	37 (0.07s)	37 (0.08s)
bw	5	28	4 (0.08s)	1 (0.08s)	0 (0.08s)
cm150a	21	1	1 (0.07s)	1 (0.07s)	1 (0.07s)
cm151a	12	2	2 (0.07s)	2 (0.07s)	2 (0.07s)
cm152a	11	1	1 (0.07s)	1 (0.07s)	1 (0.07s)
cm162a	14	5	5 (0.07s)	5 (0.07s)	5 (0.07s)
cm163a	16	5	5 (0.08s)	5 (0.08s)	5 (0.08s)
cm82a	5	3	2 (0.07s)	2 (0.07s)	1 (0.07s)
cm85a	11	3	1 (0.07s)	1 (0.09s)	1 (0.11s)
cmb	16	4	4 (0.08s)	4 (0.08s)	4 (0.09s)
con1	7	2	2 (0.06s)	2 (0.07s)	1 (0.07s)
count	35	16	16 (0.08s)	16 (0.08s)	16 (0.09s)
cu	14	11	11 (0.07s)	11 (0.12s)	11 (0.26)
duke2	22	29	29 (0.09s)	29 (0.10s)	29 (0.11s)
ex5p	8	63	29 (0.40s)	15 (0.42s)	10 (0.52s)
frg1	28	3	3 (0.82s)	3 (0.87s)	3 (1.17s)
misex1	8	7	7 (0.08s)	7 (0.08s)	6 (0.11s)
misex2	25	18	18 (0.08s)	18 (0.12s)	18 (0.87s)
misex3	14	14	2 (0.29s)	1 (0.79s)	0 (1.14s)
misex3c	14	14	12 (0.26s)	9 (0.43s)	8 (1.05s)
parity	16	1	1 (0.07s)	1 (0.07s)	1 (0.07s)
rd53	5	3	1 (0.07s)	1 (0.07s)	1 (0.07s)
rd73	7	3	1 (0.07s)	1 (0.07s)	1 (0.08s)
rd84	8	4	1 (0.09s)	1 (0.09s)	1 (0.09s)
seq	41	35	35 (0.29s)	35 (0.57s)	35 (5.16s)
sqrt8	8	5	3 (0.07s)	3 (0.08s)	2 (0.09s)
squar5	5	8	5 (0.07s)	4 (0.07s)	3 (0.07s)
vda	17	39	39 (0.10s)	39 (0.11s)	39 (0.15s)
vg2	25	8	8 (0.10s)	8 (0.14s)	8 (0.55s)
x1	51	35	35 (0.07s)	35 (0.14s)	35 (2.41s)
x2	10	7	7 (0.06s)	7 (0.07s)	5 (0.10s)
x3	135	99	99 (0.16s)	99 (0.19s)	99 (0.32s)
x4	94	71	71 (0.09s)	71 (0.11s)	71 (0.14s)
xor5	5	1	1 (0.07s)	1 (0.07s)	1 (0.07s)

Table 3. Number of decomposable functions with same variable set

Name	In	Out	$m = 2$	$m = 3$	$m = 4$
5xp1	7	10	7 (0.08s)	6 (0.09s)	5 (0.10s)
alu4	14	8	7 (0.57s)	3 (1.03s)	2 (2.24s)
bw	5	28	3 (0.08s)	1 (0.08s)	0 (0.08s)
cm151a	12	2	2 (0.07s)	2 (0.08s)	2 (0.10s)
cm162a	14	5	5 (0.08s)	5 (0.13s)	5 (0.38s)
cm163a	16	5	5 (0.08s)	5 (0.15s)	5 (0.62s)
cm82a	5	3	2 (0.07s)	2 (0.07s)	1 (0.08s)
cm85a	11	3	3 (0.08s)	1 (0.10s)	1 (0.13s)
cmb	16	4	4 (0.08s)	3 (0.11s)	3 (0.30s)
con1	7	2	2 (0.07s)	2 (0.07s)	1 (0.07s)
count	35	16	16 (0.23s)	16 (3.14s)	16 (6.71s)
cu	14	11	11 (0.09s)	11 (0.20s)	11 (0.83s)
ex5p	8	63	23 (0.40s)	9 (0.50s)	6 (0.58s)
duke2	22	29	29 (0.45s)	29 (2.75s)	29 (5.74s)
misex1	8	7	7 (0.08s)	7 (0.09s)	6 (0.12s)
misex2	25	18	18 (0.15s)	18 (1.50s)	18 (6.71s)
misex3c	14	14	12 (0.24s)	8 (0.58s)	8 (1.82s)
sqrt8	8	5	3 (0.07s)	3 (0.09s)	2 (0.10s)
squar5	5	8	5 (0.07s)	4 (0.07s)	3 (0.08s)
vda	17	39	39 (0.28s)	39 (1.17s)	39 (5.51s)
vg2	25	8	8 (0.57s)	7 (2.66s)	7 (3.52s)
x2	10	7	7 (0.07s)	7 (0.10s)	7 (0.15s)

(1) $C_s, C_d, F_{max}, F_{2d} \leftarrow 0$.

(2) Repeat the following for every combination of $(n-2)$ common variables.

(2-1) Let $\mathcal{F}'$ be the set of functions in $\mathcal{F}$ which have the decomposition with $m = 2$ for the selected common variables. Let $d \leftarrow |\mathcal{F}'|$.

(2-2) Decompose all of d functions in $\mathcal{F}'$ into $2d$ subfunctions. Among these subfunctions, let F be the number of subfunctions reduced by sharing of Theorem 5.

(2-3) If $d > 0$, $C_d \leftarrow C_d + 1$.

(2-4) If $F > 0$, $C_s \leftarrow C_s + 1$.

(2-5) If $F > F_{max}$, $F_{max} \leftarrow F$ and $F_{2d} \leftarrow 2d$.

(3) Return C_s, C_d, F_{max}, and F_{2d}.

C_d and C_s represent the numbers of cases where the network has decomposable functions and subfunction sharing, respectively. C_d and C_s is at most $\binom{n}{2}$, which is the number of combinations of $(n-2)$ common variables. F_{max} represents the maximum number of subfunctions reduced by sharing, and F_{2d} represents the total number of subfunctions without sharing. Table 4 shows the results of C_s, C_d, F_{max}, and F_{2d}, and the computing time for each networks.

The values of C_s are close to those of C_d for almost all of networks in Table 4. This indicates that subfunction sharing of Theorem 5 is usually applicable when the network has decomposable functions. Moreover, the large values of F_{max} in some networks demonstrate that Theorem 5 often works effectively to reduce network areas.

7. Conclusions and Comments

In this paper, we proposed an EXOR bi-decomposition with common variable sets such that $f(X_n) = g_0(X_{n-2}, x_n) \oplus g_1(X_{n-2}, x_{n-1})$. We formulated the decomposed subfunctions $g_0(X_{n-2}, x_n)$ and $g_1(X_{n-2}, x_{n-1})$, and proved that $f(X_n)$ is decomposable if the cofactor $f(X_{n-2}, 2, 2)$ is logical zero. Decomposed subfunctions are not unique. It is future work to study the method for determining these subfunctions in order to be most suitable for actual design.

Multi-decompositions were also presented, which decomposed $f(X_n)$ into m subfunctions with $(n-m+1)$ variables. These EXOR decompositions are applicable to many practical functions.

Then we proposed the subfunction-sharing method of the EXOR bi-decomposition. Experimental results over MCNC benchmarks show that this tech-

Table 4. Subfunction sharing of EXOR decomposition

Name	$C_s/\ C_d/\binom{n}{2}$	F_{max}/F_{2d}	Time
5xp1	21/ 21/ 21	9/ 14	0.01s
alu4	55/ 86/ 91	4/ 6	0.54s
bw	4/ 6/ 10	1/ 4	0.01s
cm162a	70/ 88/ 91	6/ 10	0.03s
cm163a	94/120/120	6/ 10	0.02s
cm85a	19/ 23/ 55	1/ 4	0.01s
cmb	38/ 92/120	4/ 6	0.01s
con1	0/ 15/ 21	0/ 0	0.01s
count	582/590/595	28/ 32	0.35s
cu	82/ 91/ 91	18/ 20	0.04s
duke2	223/231/231	52/ 58	0.60s
ex5p	28/ 28/ 28	23/ 46	0.33s
misex1	22/ 23/ 28	10/ 14	0.01s
misex2	291/299/300	32/ 34	0.21s
misex3c	82/ 91/ 91	7/ 10	0.21s
sqrt8	22/ 27/ 28	4/ 6	0.01s
squar5	9/ 10/ 10	5/ 10	0.01s
vda	121/126/126	51/ 78	0.31s
vg2	286/290/300	11/ 14	0.66s
x2	42/ 43/ 45	8/ 10	0.01s

nique is applicable to most multiple-output networks and often works effectively to reduce the network area.

References

1. Ashenhurst, R. L., "The decomposition of switching functions," *Proc. of International Symposium on Theory of Switching*, Apr. 1957, pp. 74–116.
2. Davio, M., Deschamps, J. P., and Thayse, A., *Discrete and Switching Functions*, McGraw-Hill International, 1978.
3. Matsunaga, Y., "An attempt to factor logic functions using exclusive-or decomposition," *Synthesis and System Integration of Mixed Technologies '96, Japan*, 1996, pp. 78–83.
4. Matsunaga, Y., "On Exclusive-Or Decomposition Algorithms of Logic Functions," *IPSJ SIG Notes*, DA-82-1, Dec. 1996, pp. 1–8.
5. McElvain, K., "IWLS'93 benchmark set: version 4.0," *Distributed as part of IWLS'93 benchmark distribution*, May. 1993.
6. Sasao, T., "Logic synthesis with EXOR gates," in (Sasao e.d.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, Jan. 1993.
7. Sasao, T., "FPGA design by generalized functional decomposition," in (Sasao e.d.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, Jan. 1993.
8. Sasao, T. and Butler, J. T., "On bi-decompositions of logic functions," *IPSJ SIG Notes*, DA-82-2, Dec. 1996, pp. 9–16.
9. Sawada, H., Yamashita, S., Nagoya, A., "Efficient Simple Disjunctive Decomposition by Detection Symmetric Variables with Application to Multi-level Logic Synthesis," *IPSJ SIG Notes*, DA-85-21, Oct. 1996, pp. 143–150.