Three Parameters to Find Functional Decompositions

Tsutomu Sasao   Ken-ichi Kurimoto
Department of Computer Science and Electronics
Kyushu Institute of Technology
Iizuka 820-8502, Japan

Abstract— Finding simple disjoint functional decompositions is a basic problem, but is generally time-consuming since there are nearly 2^n bipartitions of input variable. This paper introduces three parameters to find bipartitions of the input variables. It also defines "ideal random logic functions," and derives their properties. Experimental results using randomly generated functions and benchmark functions show the usefulness of the approach.

I. Introduction

Decompositions of logic functions have been studied for many years [1, 4]. A function f has a simple disjoint decomposition if f is represented as $f(X_1, X_2) = g(h(X_1), X_2)$. To find decompositions for look-up-table type FPGA, the number of variables in X_1 may be at most five [6, 10, 21]. So, the number of bipartitions to consider is $C(n, 5)$, where n is the number of the input variables.

However, to find general decompositions, the number of variables in X_1 is unbounded, and we have to consider nearly 2^n different bipartitions (X_1, X_2) of input variables $\{x_1, x_2, \ldots, x_n\}$. When n is large, the number of bipartitions to consider is too large, but the exhaustive search is impractical.

Roughly speaking, decomposition methods can be classified into two: Exhaustive methods [2, 8, 9, 14, 16, 18, 20, 22] and heuristic methods [3, 5, 23, 11, 12]. Exhaustive methods find all possible decompositions, but they are usually time consuming. On the other hand, heuristic methods find only a part of all possible decompositions, but they are relatively fast.

This paper, we will consider a heuristic method to find decompositions. For the heuristics, we introduce three parameters. By using these parameters, we can efficiently find decompositions.

The rest of the paper is organized as follows: Section II gives definitions and basic properties. Section III introduces two parameters, and show their application to find decompositions of cascade realizable functions. Section IV introduces "ideal random logic functions," as well as an another parameter. We will derive the properties of these parameters. Section V is the experimental results showing effectiveness of these approaches by using randomly generated functions and benchmark functions.

For the page limitation, all the proofs are omitted.

II. Definitions and Basic Properties

Definition 2.1

$$\frac{df}{dx_i} = f(x_1, x_2, \ldots, \overset{i}{0}, x_{i+1}, \ldots, x_n) \oplus f(x_1, x_2, \ldots, \overset{i}{1}, x_{i+1}, \ldots, x_n)$$

is a **Boolean difference** *of f with respect to x_i.*

Lemma 2.1 *Let f and g be functions of $x, y, z, \ldots$. Then, we have the following:*

1. $\frac{d\bar{f}}{dx} = \frac{df}{dx}$,
2. $\frac{df}{dx} = \frac{df}{d\bar{x}}$,
3. $\frac{d(f \cdot g)}{dx} = f \cdot \frac{dg}{dx} \oplus g \cdot \frac{df}{dx} \oplus \frac{df}{dx} \cdot \frac{dg}{dx}$,
4. $\frac{d(f \oplus g)}{dx} = \frac{dg}{dx} \oplus \frac{df}{dx}$,
5. $\frac{d(f \vee g)}{dx} = \bar{f} \cdot \frac{dg}{dx} \oplus \bar{g} \cdot \frac{df}{dx} \oplus \frac{df}{dx} \cdot \frac{dg}{dx}$, *and*
6. $\frac{d}{dx}(\frac{df}{dy}) = \frac{d}{dy}(\frac{df}{dx})$.

Lemma 2.2 *When f does not depend on x, and g depends on x, we have the followings:*

1. $\frac{df}{dx} = 0$,
2. $\frac{d(f \cdot g)}{dx} = f \cdot \frac{dg}{dx}$, *and*
3. $\frac{d(f \vee g)}{dx} = \bar{f} \cdot \frac{dg}{dx}$.

Definition 2.2 *Let (X_1, X_2) be a bipartition of $X = (x_1, x_2, \ldots, x_n)$. If f is represented as $f(X_1, X_2) = g(h(X_1), X_2)$, then f has a* **simple disjoint decomposition**. *Variables in X_1 are* **bound variables**, *and variables in X_2 are* **free variables**.

Lemma 2.3 *Let f be decomposed as $f(X_1, X_2) = g(h(X_1), X_2)$. If $x_1 \in X_1$, then*

$$\frac{df}{dx_1} = \frac{dg}{dh}\frac{dh}{dx_1}.$$

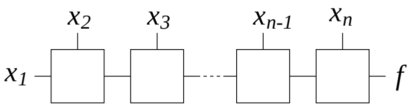

Fig. 3.1. Cascade network.

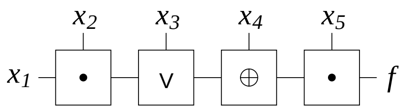

Fig. 3.2. Cascade network of five variables.

III. Cascade Realization

In this section, we will introduce two parameters that are useful to find variable orderings for cascade realizable functions.

Definition 3.1 *Let f be an n-variable logic function. The number of true minterms in f is denoted by $w(f)$. $d = w(f)/2^n$ $(0 \leq d \leq 1)$ is the* **density** *of f, and denoted by $den(f)$.*

Lemma 3.1 *Let $h_1(X_1)$ and $h_2(X_2)$ be functions with densities d_1 and d_2, respectively, where $\{X_1\} \cap \{X_2\} = \phi$. Then, $den(h_1(X_1)h_2(X_2)) = d_1 d_2$.*

Lemma 3.2 *Let x be a variable of h. Then*

$$\frac{1}{2} den(\frac{dh}{dx}) \leq \min\{den(h), den(\bar{h})\}.$$

Lemma 3.3 *Let $h(X)$ be an arbitrary logic function. Let x_k be a variable in $\{X\}$, and let x_{k+1} be a variable not in $\{X\}$.*

When $f = x_{k+1} \vee h(X)$.

$$\frac{df}{dx_{k+1}} = \bar{h}(X) \text{ and } \frac{df}{dx_k} = \bar{x}_{k+1}(\frac{dh}{dx_k}).$$

When $f = x_{k+1}h(X)$.

$$\frac{df}{dx_{k+1}} = h(X) \text{ and } \frac{df}{dx_k} = x_{k+1}\frac{dh}{dx_k}.$$

When $f = x_{k+1} \oplus h(X)$.

$$\frac{df}{dx_{k+1}} = 1 \text{ and } \frac{df}{dx_k} = \frac{dh}{dx_k}.$$

Theorem 3.1 *Let $f(x_1, x_2, \ldots, x_n)$ be realized by a cascade network of two-input gates as shown in Fig. 3.1. Then, $den(\frac{df}{dx_i}) \leq den(\frac{df}{dx_j})$, for $i < j$.*

Example 3.1 *Consider the cascade network shown in Fig. 3.2. Note that $f = [(x_1x_2 \vee x_3) \oplus x_4]x_5$.*

$$\frac{df}{dx_1} = x_2\bar{x}_3x_5,$$
$$\frac{df}{dx_2} = x_1\bar{x}_3x_5,$$
$$\frac{df}{dx_3} = (\bar{x}_1 \vee x_1\bar{x}_2)x_5,$$
$$\frac{df}{dx_4} = x_5, \text{ and}$$
$$\frac{df}{dx_5} = (x_1x_2 \vee x_3) \oplus x_4.$$

Thus, we have

$$den(\frac{df}{dx_1}) = \frac{2}{16},$$
$$den(\frac{df}{dx_2}) = \frac{2}{16},$$
$$den(\frac{df}{dx_3}) = \frac{6}{16},$$
$$den(\frac{df}{dx_4}) = \frac{8}{16}, \text{ and}$$
$$den(\frac{df}{dx_5}) = \frac{8}{16}.$$

Note that $den(\frac{df}{dx_i}) \leq den(\frac{df}{dx_j})$ $(1 \leq i < j \leq 5)$.
(End of Example)

As shown in Theorem 3.1, $den(\frac{df}{dx_i})$ is a parameter that shows the ordering of the variable in the cascade realizations. Next, we will introduce an another parameter that also shows the ordering of the variables.

Definition 3.2 *Let f be an n-variable function such that $f = \bar{x}_i f_0 \vee x_i f_1$.*

$$\rho(f : x_i) = |w(f_0) - w(f_1)| = |w(f) - 2w(f_1)|$$

and $i = 1, 2, \ldots, n$.

Lemma 3.4
1) $\rho(f : x_i) = \rho(\bar{f} : x_i)$.
2) $\rho(f : x_i) = \rho(f : \bar{x}_i)$.

Lemma 3.5 *Let x be a variable, and g be an $(n-1)$-variable function that does not depend on x. Then,*

$$w(x \oplus g) = 2^{n-1}.$$

Theorem 3.2 *Let $f(x_1, x_2, \ldots, x_n)$ be realized by a cascade network of two-input gates as shown in Fig. 3.1. Then, $\rho(f : x_i) \leq \rho(f : x_j)$, for $1 \leq i < j \leq n$.*

Example 3.2 *Consider the function $f = [(x_1x_2 \vee x_3) \oplus x_4]x_5$, which appeared in Example 3.1.*

$$\rho(f : x_1) = w((x_3 \oplus x_4)x_5) - w([(x_2 \vee x_3) \oplus x_4]x_5) = 4 - 4 = 0,$$
$$\rho(f : x_2) = w((x_3 \oplus x_4)x_5) - w([(x_1 \vee x_3) \oplus x_4]x_5) = 4 - 4 = 0,$$
$$\rho(f : x_3) = w((x_1x_2 \oplus x_4)x_5) - w(\bar{x}_4x_5) = 4 - 4 = 0,$$
$$\rho(f : x_4) = w((x_1x_2 \vee x_3)x_5) - w(\overline{(x_1x_2 \vee x_3)}x_5) = 5 - 3 = 2,$$
$$\rho(f : x_5) = |w(0) - w((x_1x_2 \vee x_3) \oplus x_4)| = 8.$$

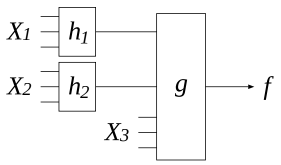

Fig. 4.1. Decomposition $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), X_3)$.

Thus, $\rho(f : x_i) \leq \rho(f : x_j)$, $(1 \leq i < j \leq 5)$. *(End of Example)*

As shown in this section, $den(\frac{df}{dx_i})$ and $\rho(f : x_i)$ show the ordering of variables in cascade realizable functions.

IV. Ideal Random Logic Functions

In the previous section, we showed that $den(\frac{df}{dx_i})$ is useful to find bipartitions for cascade realizable functions. However, only a fraction of the functions are cascade realizable. In this section, we will introduce ideal random logic functions, and show that $den(\frac{df}{fx_i})$ and $den(\frac{d^2f}{dx_i dx_j})$ are useful to find decompositions for a class of functions.

Definition 4.1 Ideal random logic functions *have the following properties:*

1) *Let* $f(X) = \bar{x}_i f_0 \vee x_i f_1$ *be an ideal random logic function, and let the density of* f *be* d*. Then,* f_0 *and* f_1 *are also ideal random logic functions with the densities* d *for all* $i \in \{1, 2, \ldots, n\}$.
2) *Let* $f_1(X)$ *and* $f_2(X)$ *be two different ideal random logic functions with densities* d_1 *and* d_2*, respectively. Then,* $den(f_1(X)f_2(X)) = d_1 d_2$.

Lemma 4.1 *Let* f *be an ideal random logic function with density* d*. Then,* $den(df/dx) = 2d(1-d)$.

Lemma 4.2 *Let* g *and* h *be ideal random logic functions with densities* d_g *and* d_h*, respectively. Let* f *be decomposed as* $f(X_1, X_2) = g(h(X_1), X_2)$*. When* x_i *is in* $\{X_1\}$, $A = den(\frac{df}{dx_i}) = 4d_g d_h(1-d_g)(1-d_h)$*. When* x_i *is in* $\{X_2\}$, $B = den(\frac{df}{dx_i}) = 2d_g(1-d_g)$.

Since $A/B = 2d_h(1-d_h) \leq 1/2$ in Lemma 4.2, we have the following:

Theorem 4.1 *Let* f *be decomposable as* $f(X_1, X_2) = g(h(X_1), X_2)$*, where* g *and* h *are ideal random logic functions. If* $x_i \in \{X_1\}$ *and* $x_j \in \{X_2\}$*, then* $w(\frac{df}{dx_i}) < w(\frac{df}{dx_j})$.

Lemma 4.3 *Let* (X_1, X_2, X_3) *be a partition of* X*, and* f *be decomposed as* $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), X_3)$*, where* g*,* h_1*, and* h_2 *are ideal random logic functions with densities* d_g*,* d_{h_1}*, and* d_{h_2}*, respectively (See Fig. 4.1). Also, assume that the Boolean differences of* g*,* h_1*, and* h_2 *are also ideal random logic functions.*

1) *When* $x_1, x_2 \in \{X_1\}$.

$$A = den(\frac{d^2 f}{dx_1 dx_2}) = 8d_g(1-d_g)d_{h_1}(1-d_{h_1})(1-2d_{h_1}+2d_{h_1}^2).$$

2) *When* $x_1 \in \{X_1\}$ *and* $x_2 \in \{X_2\}$.

$$B = den(\frac{d^2 f}{dx_1 dx_2}) = 16d_g(1-d_g)(1-2d_g+2d_g^2) d_{h_1}(1-d_{h_1})d_{h_2}(1-d_{h_2}).$$

3) *When* $x_1 \in \{X_1\}$ *and* $x_2 \in \{X_3\}$.

$$C = den(\frac{d^2 f}{dx_1 dx_2}) = 8d_g(1-d_g)(1-2d_g+2d_g^2)d_{h_1}(1-d_{h_1}).$$

4) *When* $x_1, x_2 \in \{X_3\}$.

$$D = den(\frac{d^2 f}{dx_1 dx_2}) = 4d_g(1-d_g)(1-2d_g+2d_g^2).$$

Theorem 4.2 *Let* (X_1, X_2, X_3) *be a partition of* X*, and* f *be decomposed as* $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), X_3)$*, where* g*,* h_1*, and* h_2 *are ideal random logic functions with densities* d_g*,* d_{h_1}*, and* d_{h_2}*, respectively (See Fig. 4.1). Also, assume that the Boolean differences of* g*,* h_1*, and* h_2 *are also ideal random logic functions. If* $x_1 \in \{X_1\}$*,* $x_2 \in \{X_2\}$*, and* $x_{3A}, x_{3B} \in \{X_3\}$*, then*

$$den(\frac{d^2 f}{dx_1 dx_2}) < den(\frac{d^2 f}{dx_1 dx_{3A}}) < den(\frac{d^2 f}{dx_{3A} dx_{3B}}).$$

As shown in this section, $den(\frac{df}{dx_i})$ and $den(\frac{d^2f}{dx_i dx_j})$ are useful to find the decompositions of functions that are composed of ideal random logic functions.

V. Experimental Results

In many cases, the given functions are not cascade realizable nor composed of ideal random logic functions. However, parameters introduced in Section III are still useful to find bipartitions for disjoint decompositions. To see the usefulness of the parameters, we constructed multi-level networks by using randomly generated functions.

A. Decomposition of $f = g(g(X_1), X_2)$, where $|X_1| = 5$.

We randomly generated a five-variable function g, where $w(g) = 12$, and constructed the network shown in Fig. 5.1.

Note that $f = g(g(X_1), X_2)$, where $X_1 = (x_1, x_2, x_3, x_4, x_5)$ and $X_2 = (x_6, x_7, x_8, x_9)$. Table 5.1 shows the values of $\rho(f : x_i)$ and $w(\frac{df}{dx_i})$. The values of $\rho(f : x_i)$ tend to be greater for free variables. However, $\rho(f : x_8) = 0$ even if x_8 is a free variable. So, $\rho(f : x_i)$ is not so a reliable parameter. On the other hand, the values of $w(\frac{df}{dx_i})$ for free variables are always greater than ones for bound variables.

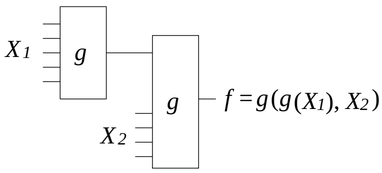

Fig. 5.1. Generation of a function $f = g(g(X_1), X_2)$.

TABLE 5.1
Parameters for a 9-variable function $f = g(g(X_1), X_2)$.

	X_1					X_2			
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
$\rho(f : x_i)$	0	0	0	0	0	24	40	0	24
$w(\frac{df}{dx_i})$	100	120	100	120	100	152	192	152	192

B. Decomposition of $f = g(g(X_1), g(X_2), X_3)$, where $|X_1| = 5$.

To see the effectiveness of the parameters $den(\frac{d^2 f}{dx_i dx_j})$, we did the following: We used the same random function g to construct the network shown in Fig. 5.2, where $X_1 = (x_1, x_2, \ldots, x_5)$, $X_2 = (x_6, x_7, \ldots, x_{10})$, and $X_3 = (x_{11}, x_{12}, x_{13})$.

Table 5.2 shows the values of $\rho(f : x_i)$ and $w(\frac{df}{dx_i})$. Also in this case, we can observe that the parameters for the free variables are larger than ones for the bound variables.

Table 5.3 shows the values of $w(\frac{d^2 f}{dx_i dx_j})$. The results are consistent with Theorem 4.2:

1) When $x_i \in X_1$ and $x_j \in X_2$, $200 \leq w(\frac{d^2}{dx_i dx_j}) \leq 288$.
2) When $x_i \in X_1$ and $x_j \in X_3$, $640 \leq w(\frac{d^2}{dx_i dx_j}) \leq 768$.
3) When $x_i, x_j \in X_3$, $640 \leq w(\frac{d^2}{dx_i dx_j}) \leq 928$.

However, in this case, these parameters are not sufficient to find the bipartition of the variables. The maximum value 1152 occurs when $(x_i, x_j \in X_1)$ or $(x_i \in X_2$ and $x_j \in X_3)$. Also, the value 640 occur for many cases.

One reason for this is that g and $\frac{df}{dx_i}$ do not satisfy the conditions of ideal random logic functions. We consider that to be ideal random logic functions, the number of variables in g should be larger.

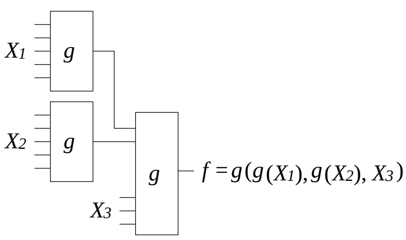

Fig. 5.2. Generation of a function $f = g(g(X_1), g(X_2), X_3)$.

C. Decomposition of $f = g(g(X_1), g(X_2), X_3)$, where $|X_1| = 7$.

We generated a random function g of 7 variables, where $w(g) = 64$. And, constructed the network shown in Fig. 5.2, where $X_1 = (x_1, x_2, \ldots, x_7)$, $X_2 = (x_8, x_9, \ldots, x_{14})$, and $X_3 = (x_{15}, x_{16}, \ldots, x_{19})$.

Table 5.4 shows the values of $w(\frac{d^2 f}{dx_i dx_j})$. In this case, the results better support Theorem 4.2:

1) When $x_i \in X_1$ and $x_j \in X_2$, $8064 \leq w(\frac{d^2}{dx_i dx_j}) \leq 20216$.
2) When $x_i \in X_1$ and $x_j \in X_3$, $18432 \leq w(\frac{d^2}{dx_i dx_j}) \leq 38912$.
3) When $x_i, x_j \in X_3$, $49152 \leq w(\frac{d^2}{dx_i dx_j}) \leq 90112$.

D. Other Benchmark Functions

In most cases, logic functions used in industries are non-random. However, we can use these parameters to find decompositions by using the following:

Algorithm 5.1

1. *Partition the multiple-output function into single-output functions, and decompose each function separately. Generate the BDD for each function.*
2. *If there is a level with width two, then decompose the function into two, and apply this step recursively.*
3. *Use $w(\frac{df}{dx_i})$ to order the input variables, and reorder the variables in the BDD to find decompositions. If tie, use $\rho(f : x_i)$ to order the variables. If tie, check the symmetry of the variables.*

Table 5.5 compares the numbers of decompositions found by Algorithm 5.1 and ones found by DECOMPOS [16]. In Table 5.5, PAR denotes the number of blocks after decompositions using Algorithm 5.1; JAC denotes the number of blocks after decompositions using DECOMPOS [16], and OUT denotes the number of outputs. Note that DECOMPOS finds all the disjoint decompositions.

$$Ratio = \frac{(PAR - OUT)}{(JAC - OUT)} \times 100$$

denotes the percentage of the decompositions found by Algorithm 5.1. Note that C1355 has no decomposition. Except for a few cases (i.e., C1908, b3 and x6dn), Algorithm 5.1 found considerable part of decompositions.

VI. Conclusions

In this paper, we introduced three parameters to find disjoint decompositions. $\rho(f : x_i)$ and $w(\frac{df}{dx_i})$ show the influence of the variables in the networks: The greater the values, the more influential the variables. Thus, the less influential variables are candidates for bound variables.

Since these parameters are relatively easily calculated, any functional decomposition systems can incorporate this method.

TABLE 5.2
Parameters for a 13-variable function $f = g(g(X_1), g(X_2), X_3)$.

	X_1					X_2					X_3		
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
$\rho(f : x_i)$	48	48	0	48	0	16	16	0	16	0	448	960	160
$w(\frac{df}{dx_i})$	1920	2304	1920	2304	1920	1600	1920	1600	1920	1600	2368	3168	2368

TABLE 5.3
Parameters for a 13-variable function $f = g(g(X_1), g(X_2), X_3)$.

		X_1					X_2					X_3		
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
	x_1	0	768	768	768	768	200	240	200	240	200	640	640	640
	x_2	768	0	384	768	768	240	288	240	288	240	768	768	768
X_1	x_3	768	384	0	768	1152	200	240	200	240	200	640	640	640
	x_4	768	768	768	0	384	240	288	240	288	240	768	768	768
	x_5	768	768	1152	384	0	200	240	200	240	200	640	640	640
	x_6	200	240	200	240	200	0	640	640	640	640	720	560	960
	x_7	240	288	240	288	240	640	0	320	640	640	864	672	1152
X_2	x_8	200	240	200	240	200	640	320	0	640	960	720	560	960
	x_9	240	288	240	288	240	640	640	640	0	320	864	672	1152
	x_{10}	200	240	200	240	200	640	640	960	320	0	720	560	960
	x_{11}	640	768	640	768	640	720	864	720	864	720	0	928	928
X_3	x_{12}	640	768	640	768	640	560	672	560	672	560	928	0	640
	x_{13}	640	768	640	768	640	960	1152	960	1152	960	928	640	0

Acknowledgments

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Culture and Sports of Japan, as well as the Takayanagi Foundation for Electronics Science and Technology. Mr. M. Matsuura formatted the LaTeX files.

References

[1] R. L. Ashenhurst, "The decomposition of switching functions," *In Proceedings of an International Symposium on the Theory of Switching*, pp. 74-116, April 1957.

[2] V. Bertacco and M. Damiani, "The disjunctive decomposition of logic functions," *Proc. ICCAD*, pp. 78-82, 1997.

[3] S. C. Crist, "Synthesis of combinational logic using decomposition and probability," *IEEE Trans. Comput.*, Vol. C-29, No. 11, pp. 1013-16, Nov. 1980.

[4] H. A. Curtis, *A New Approach to the Design of Switching Circuits*, D. Van Nostrand Co., Princeton, NJ, 1962.

[5] E. V. Dubrova, D. M. Miller, J. C. Muzio, "On the relation between disjunctive decomposition and ROBDD variable ordering," *1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PACRIM. 10 Years Networking the Pacific Rim*, 1987-1997, pp. 688-691 Vol. 2, 2 Vol. xxiii+1021, 1997.

[6] Ting-Ting Hwang, R. M. Owens, M. J. Irwin, and Kuo Hua Wang, "Logic synthesis for field-programmable gate arrays," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 13, No. 10, pp. 1280-1287, Oct. 1994

[7] Y-T. Lai, M. Pedram, and S. B. K. Vrudhula, "EVBDD-based algorithm for integer linear programming, spectral transformation, and functional decomposition," *IEEE Trans. CAD*, Vol. 13, No. 8, pp. 959-975, Aug. 1994.

[8] Y. Matsunaga, "An exact and efficient algorithm for disjunctive decomposition," *SASIMI'98*, pp. 44-50, Oct. 1998.

[9] S. Minato and G. De Micheli, "Finding all simple disjunctive decompositions using irredundant sum-of-products forms," *ICCAD-98*, pp. 111-117, Nov. 1998.

[10] R. Murgai, R. K. Brayton, and A. Sangiovanni-Vincentelli, *Logic Synthesis for Field-Programmable Gate Arrays*, Kluwer, 1995.

[11] M. Rawski, L. Jozwiak, and T. Luba, "Efficient input support selection for sub-functions in functional decomposition based on information relationship measures," *Euromicro 1999*, Milan, Sept. 1999.

[12] J. T. Proudfoot and S. M. Ngwira, "Non-exhaustive method for identification of optimal variable orderings in the decomposition of complex logic functions," *IEE Proc., Comput. Digit. Tech.*, Vol. 142, No. 5, pp. 373-5, Sept. 1995.

[13] J. P. Roth and R. M. Karp, "Minimization over Boolean graphs," *IBM Journal of Research and Development*, pp. 227-238, April 1962.

[14] T. Sasao, "FPGA design by generalized functional decomposition," in (Sasao ed.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.

TABLE 5.4

Parameters for 19-variable function $f = g(g(X_1), g(X_2), X_3)$.

		X_1							X_2							X_3				
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
	x_1	0	27648	24576	18432	27648	24576	30720	14336	14336	11648	17024	16128	10752	13440	32768	24576	24576	28672	24576
	x_2	27648	0	27648	30720	21504	18432	21504	14336	14336	11648	17024	16128	10752	13440	32768	24576	24576	28672	24576
	x_3	24576	27648	0	21504	24576	18432	24576	11648	11648	9464	13832	13104	8736	10920	26624	19968	19968	23296	19968
X_1	x_4	18432	30720	21504	0	33792	21504	30720	17024	17024	13832	20216	19152	12768	15960	38912	29184	29184	34048	29184
	x_5	27648	21504	24576	33792	0	18432	24576	16128	16128	13104	19152	18144	12096	15120	36864	27648	27648	32256	27648
	x_6	24576	18432	18432	21504	18432	0	21504	10752	10752	8736	12768	12096	8064	10080	24576	18432	18432	21504	18432
	x_7	30720	21504	24576	30720	24576	21504	0	13440	13440	10920	15960	15120	10080	12600	30720	23040	23040	26880	23040
	x_8	14336	14336	11648	17024	16128	10752	13440	0	34560	30720	23040	34560	30720	38400	40960	28672	32768	40960	32768
	x_9	14336	14336	11648	17024	16128	10752	13440	34560	0	34560	38400	26880	23040	26880	40960	28672	32768	40960	32768
X_2	x_{10}	11648	11648	9464	13832	13104	8736	10920	30720	34560	0	26880	30720	23040	30720	33280	23296	26624	33280	26624
	x_{11}	17024	17024	13832	20216	19152	12768	15960	23040	38400	26880	0	42240	26880	38400	48640	34048	38912	48640	38912
	x_{12}	16128	16128	13104	19152	18144	12096	15120	34560	26880	30720	42240	0	23040	30720	46080	32256	36864	46080	36864
	x_{13}	10752	10752	8736	12768	12096	8064	10080	30720	23040	23040	26880	23040	0	26880	30720	21504	24576	30720	24576
	x_{14}	13440	13440	10920	15960	15120	10080	12600	38400	26880	30720	38400	30720	26880	0	38400	26880	30720	38400	30720
	x_{15}	32768	32768	26624	38912	36864	24576	30720	40960	40960	33280	48640	46080	30720	38400	0	73728	65536	49152	73728
	x_{16}	24576	24576	19968	29184	27648	18432	23040	28672	28672	23296	34048	32256	21504	26880	73728	0	73728	81920	57344
	x_{17}	24576	24576	19968	29184	27648	18432	23040	32768	32768	26624	38912	36864	24576	30720	65536	73728	0	57344	65536
X_3	x_{18}	28672	28672	23296	34048	32256	21504	26880	40960	40960	33280	48640	46080	30720	38400	49152	81920	57344	0	90112
	x_{19}	24576	24576	19968	29184	27648	18432	23040	32768	32768	26624	38912	36864	24576	30720	73728	57344	65536	90112	0

TABLE 5.5

Functional decompositions using Algorithm 5.1.

Data name	IN	OUT	# Blocks PAR	# Blocks JAC	Ratio (%)
C1355	41	32	32	32	—
C1908	33	25	26	94	1.4
accpla	50	69	564	703	78.1
alu4	14	8	15	15	100.0
apex1	45	45	257	266	96.0
apex2	39	3	34	37	91.2
apex5	117	88	512	870	54.2
b3	32	20	122	183	62.6
comp	32	3	63	63	100.0
des	256	245	1080	1374	74.0
frg2	143	139	1183	1227	96.0
i3	132	6	126	126	100.0
i4	192	6	186	186	100.0
rckl	32	7	216	216	100.0
signet	39	8	27	32	79.2
t481	16	1	15	15	100.0
x2dn	82	56	104	105	98.0
x6dn	39	5	14	36	29.0
xparc	41	73	689	752	90.7

IN : Number of the input variables.
OUT: Number of the output variables.
PAR: Number of blocks after decompositions using Algorithm 5.1.
JAC: Number of blocks after decompositions using DECOMPOS [16].
$Ratio = \frac{(PAR-OUT)}{(JAC-OUT)} \times 100.$

[15] T. Sasao and J. T. Butler, "On bi-decompositions of logic functions," *ACM/IEEE International Workshop on Logic Synthesis*, Tahoe City, California, May 1997.

[16] T. Sasao and M. Matsuura, "DECOMPOS: An integrated system for functional decomposition," *1998 International Workshop on Logic Synthesis*, Lake Tahoe, June 1998.

[17] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers (1999-02).

[18] T. Sasao, "Totally undecomposable functions: Applications to efficient multiple-valued decompositions," *IEEE International Symposium on Multiple-Valued Logic*, pp. 59-65, Freiburg, Germany, May 20-23, 1999.

[19] T. Sasao and S. Kajihara, "Functional decompositions using an automatic test pattern generators and a logic simulator," *ACM/IEEE International Workshop on Logic Synthesis*, Tahoe City, California, June 28-30, 1999.

[20] T. Sasao, "Arithmetic ternary decision diagrams and their applications," *4th International Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, Victoria, Canada, August, 1999.

[21] H. Sawada, T. Suyama, and A. Nagoya, "Logic synthesis for look-up table based FPGAs using functional decomposition and support minimization," *ICCAD*, pp. 353-359, Nov. 1995.

[22] V. Y-S, Shen, A. C. Mckellar, and P. Weiner, "A fast algorithm for the disjunctive decomposition of switching functions," *IEEE Trans. Comput.*, Vol. C-20, No. 3, pp. 304-309, March 1971.

[23] W-Z Shen, J-D Huang, and S-M Chao, "Lamda set selection in Roth-Karp decomposition for LUT-based FPGA technology mapping," *32nd Design Automation Conference*, pp. 65-69, June 1995.

[24] S. Yang, "Logic synthesis and optimization benchmark user guide," Version 3.0, MCNC, Jan. 1991.