On the Number of LUTs to Realize Sparse Logic Functions

Tsutomu Sasao
Kyushu Institute of Technology,
Iizuka 820-8502, Japan

Abstract

The column multiplicity of a decomposition table for a given function is the number of distinct column patterns. This paper derives upper bounds on column multiplicities for logic functions whose number of true minterms is specified. These bounds are significant improvement over bounds in which the number of true minterms is unspecified. From these bounds, we can estimate the number of LUTs to implement logic functions in FPGA logic synthesis.

1. Introduction

A K-LUT (look-up table) is a module that realizes an arbitrary K-variable function. At one time, FPGAs (field programmable gate arrays) with 4 or 5 input LUTs were believed to be the most efficient [7, 8]. However, in current technology, FPGAs with $K = 6$ are standard [1, 2, 15]. In the future, LUTs with more inputs should be available. Thus, we have the following:

Problem 1.1 *Given an n-variable logic function f, find the minimum number of K-LUTs needed to implement f, where $K = 6, 7$ and 8.*

Unfortunately, this is a very hard problem. So, we try to obtain an upper bound by using measures of the function. The measures should be quickly calculated. Such measures include:

1. The number of variables in the function.
2. The number of literals and products in a sum-of-products expression for the function.
3. The number of true minterms in the function.

An upper bound on the number of LUTs can be obtained from the first measure [6, 11], or the second measure [6]. In this paper, we show that if the third measure is given in addition to the first measure, then we have a much better bound than the first measure only. Also, it is much better than one obtained from the second measure when the function is random. From this result, we can quickly estimate the number of LUTs to implement the given function.

This paper is organized as follows: Section 2 derives upper bounds on the number of 6-LUTs for general functions. It also introduces a C-measure of a logic function, which is useful to estimate the number of LUTs to implement the function. Section 3 shows a method to derive the C-measure for a function with small weights. Section 4 considers average C-measures for uniformly distributed functions. Section 5 shows experimental results for benchmark and randomly generated functions.

2 Functional Decomposition and LUT Cascade

In logic synthesis, estimates of the circuit size are quite useful. Before considering the general case, we start with special cases.

Lemma 2.1 *An arbitrary function of $n = K + 1$ variables can be implemented with at most three K-LUTs.*

(Proof) Let $X_1 = (x_1, x_2, \ldots, x_k)$ and $X_2 = (x_{k+1})$. Then, the function can be represented as

$$f(X_1, X_2) = \bar{x}_{k+1} f(X_1, 0) \vee x_{k+1} f(X_1, 1).$$

Thus, f can be implemented by three K-LUTs as shown in Fig. 2.1. Note that the left cell realizes $f(X_1, 0)$ and $f(X_1, 1)$, while the right cell works as a selector to realize f. The integer in a cell denotes the number of LUTs to implement the cell. (Q.E.D.)

Lemma 2.2 *An arbitrary function of $n = K + 2$ variables can be implemented with at most five K-LUTs.*

(Proof) Let $X_1 = (x_1, x_2, \ldots, x_k)$ and $X_2 = (x_{k+1}, x_{k+2})$. Then, the function can be represented as

$$f(X_1, X_2) = \bar{x}_{k+1}\bar{x}_{k+2} f(X_1, 0, 0) \vee \bar{x}_{k+1} x_{k+2} f(X_1, 0, 1) \vee x_{k+1}\bar{x}_{k+2} f(X_1, 0, 1) \vee x_{k+1} x_{k+2} f(X_1, 1, 1).$$

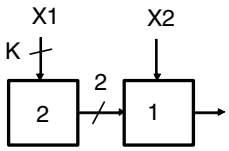

Figure 2.1. Realization of a k+1 variable function.

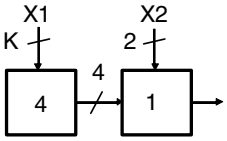

Figure 2.2. Realization of a k+2 variable function.

Thus, f can be implemented by five K-LUTs as shown in Fig. 2.2. Note that the left cell realizes $f(X_1, 0, 0)$, $f(X_1, 0, 1)$, $f(X_1, 1, 0)$, and $f(X_1, 1, 1)$, while the right cell works as a selector to realize f. (Q.E.D.)

In designing an LUT cascade, decomposition theory is essential.

Definition 2.1 *[3] Let $f(X)$ be a logic function, and (X_1, X_2) be a partition of the input variables, where $X_1 = (x_1, x_2, \ldots, x_k)$ and $X_2 = (x_{k+1}, x_{k+2}, \ldots, x_n)$. The* **decomposition table** *for f is a two-dimensional matrix with 2^k columns and 2^{n-k} rows, where each column and row is labeled by a unique binary code representing an assignment of values to the variables, and each element corresponds to the truth value of f. The function represented by a column is a* **column function***. Variables in X_1 are* **bound variables***, while variables in X_2 are* **free variables***. In the decomposition table, the* **column multiplicity** *denoted by μ_k is the number of different column patterns.*

Example 2.1 *Fig. 2.3 shows a decomposition table of a 4-variable function. Since all the column patterns are different, the column multiplicity is $\mu_2 = 4$. (End of Example)*

From Definition 2.1, we have the following:

Theorem 2.1 *[3] Let $\mu_k(n)$ be the column multiplicity of an n-variable logic function with k bound variables. Then,*

$$\mu_k(n) \leq \min\{2^k, 2^{2^{n-k}}\}.$$

When circuits are designed by LUTs, functions with smaller column multiplicities tend to have smaller realizations. The following theorem on **functional decomposition** is well known.

$X_2 = (x_3, x_4)$ \ $X_1 = (x_1, x_2)$	00	01	10	11
00	0	0	0	1
01	1	1	0	0
10	0	1	0	0
11	0	0	0	0

Figure 2.3. Decomposition table of a logic function.

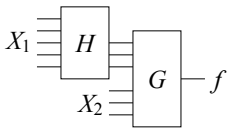

Figure 2.4. Realization logic function by decomposition.

Theorem 2.2 *[4] Let $X_1 = (x_1, x_2, \ldots, x_k)$ and $X_2 = (x_{k+1}, x_{k+2}, \ldots, x_n)$ be bound variables and free variables, respectively. Let μ_k be the column multiplicity of the decomposition table for f. Consider Fig. 2.4 which realizes the function f. Then, the necessary and sufficient number of lines between two blocks H and G is $\lceil \log_2 \mu_k \rceil$.*

Definition 2.2 *Let $f(x_1, x_2, \ldots, x_n)$ be a logic function. The* **profile** *of the function f is the vector $(\mu_1, \mu_2, \ldots, \mu_n)$, where μ_k is the column multiplicity of the decomposition table for $f(X_1, X_2)$, $X_1 = (x_1, x_2, \ldots, x_k)$ and $X_2 = (x_{k+1}, \ldots, x_n)$. The* **C-Measure** *of the function f is $\max(\mu_1, \mu_2, \ldots, \mu_n)$, and is denoted by $\mu(f)$.*

When we consider a C-measure of a function $f(x_1, x_2, \ldots, x_n)$, the order of the input variables are assumed to be $(x_1, x_2, \ldots, x_n)$. Thus, two functions $f_1 = x_1x_2 \vee x_3x_4 \vee x_5x_6$ and $f_2 = x_1x_3 \vee x_5x_2 \vee x_4x_6$ have different C-measures: $\mu(f_1) = 3$ and $\mu(f_2) = 8$. In a logic synthesis program, we try to obtain the permutation of the input variables that reduces the C-measure. However, to obtain an upper bound on C-measure, for simplicity, we only consider one variable ordering of the input variables.

Lemma 2.3 *Let f be an arbitrary n-variable function. Then,*

$$\mu(f) \leq \max_{k=1}^{n} \min\{2^k, 2^{2^{n-k}}\}.$$

By repeatedly applying functional decompositions to a given function, we have an **LUT cascade** [12] shown in Fig. 2.5. An LUT cascade consists of **cells**. The signal lines connecting adjacent cell are **rails**. A logic function with a small C-measure can be realized by a compact LUT cascade.

Lemma 2.4 *[12] An arbitrary logic function f can be implemented by an LUT cascade, whose cells have at most $\lceil \log_2 \mu(f) \rceil + 1$ inputs, and $\lceil \log_2 \mu(f) \rceil$ outputs.*

Lemma 2.5 *[13] In an LUT cascade that realizes a function f, let n be the number of input variables; s be the number of cells; $w = \lceil \log_2 \mu(f) \rceil$ be the maximum number of rails ; K be the number of inputs for a cell; $n \geq K+1$; and $K \geq \lceil \log_2 \mu(f) \rceil + 1$. Then, an LUT cascade satisfying the following condition exists:*

$$s \leq \left\lceil \frac{n-w}{K-w} \right\rceil .$$

Lemma 2.6 *Consider the cascade consisting of K-LUTs.*

1. *When the number of the external input variables to the output LUT is one, the number of the rail inputs to the LUT is at most two.*
2. *When the number of the external input variables to the output LUT is two, the number of the rail inputs to the LUT is at most four.*

(Proof) Here, we show the second case. The proof for the first case is similar. Let x_{n-1} and x_n be external input variables for the output LUT. Consider the decomposition table, where the rail inputs $X_1 = (x_1, x_2, \ldots, x_{n-2})$ denotes bound variables and the external inputs $X_2 = (x_{n-1}, x_n)$ denote the free variables. In this case, the column multiplicity is at most 16, since there exist at most $2^{2^2} = 16$ different column functions by Theorem 2.1. Also by Theorem 2.2, the number of the rail inputs to the output LUT is at most four. (Q.E.D.)

As for realizations by 6-LUTs, we have the following:

Theorem 2.3 *The number of 6-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 32$ is $5n - 35$ or less, where $n \geq 8$.*

(Proof) From Lemma 2.4, an arbitrary function with $\mu(f) \leq 32$ can be implemented by an LUT cascade, whose cells have at most $\lceil \log_2 \mu \rceil + 1 = \log_2(32) + 1 = 6$ inputs, and $w = \log_2(32) = 5$ outputs. Let $K = 6$. Then, from Lemma 2.5, the number of cells is at most $\lceil \frac{n-w}{K-w} \rceil = \frac{n-5}{6-5} = n - 5$. Note that each cell except for the rightmost cell has at most 5 outputs. From Lemma 2.6, the second cell from the right has at most four outputs as shown in Fig. 2.6. So, the total number of cells is at most $n - 6$. Note that the second cell from the right has 4 outputs, while the rightmost cell has just one output. Therefore, the total number of LUTs is at most $5(n-8) + 4 + 1 = 5n - 35$. (Q.E.D.)

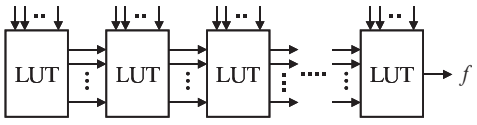

Figure 2.5. LUT cascade.

Theorem 2.4 *The number of 6-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 16$ is $2n - 11$ or less, where $n \geq 8$.*

(Proof) Let $w = \log_2(16) = 4$ and $K = 6$. From Lemma 2.5, we have $s \leq \lceil \frac{n-w}{K-w} \rceil = \frac{n-4}{6-4} = \lceil \frac{n-4}{2} \rceil$.

When $n = 2r$, each cell except for the rightmost cell has at most 4 outputs. So, the total number of LUTs is at most $4 \times (\lceil \frac{n-4}{2} \rceil - 1) + 1 = 2(n-4) - 4 + 1 = 2n - 11$.

When $n = 2r + 1$, by Lemma 2.6, the rightmost cell has one external input and at most two rail inputs, and the second cell from the right has at most two outputs. So, the total number of LUTs is at most $4 \times (\lceil \frac{n-5}{2} \rceil - 1) + 2 + 1 = 2n - 11$. (Q.E.D.)

Theorem 2.5 *The number of 6-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 8$ is $n-5$ or less when $n = 3r$, and $n-4$ or less when $n \neq 3r$, where $n \geq 8$.*

(Proof) Let $w = \log_2 8 = 3$ and $K = 6$. From Lemma 2.5, we have $s \leq \lceil \frac{n-w}{K-w} \rceil = \lceil \frac{n-3}{3} \rceil$.

When $n = 3r$, each cell except for the rightmost cell has at most 3 outputs. So, the total number of LUTs is at most $3 \times (\lceil \frac{n-3}{3} \rceil - 1) + 1 = (n-3) - 3 + 1 = n - 5$.

When $n = 3r + 1$, the rightmost cell has one external input and at most two rail inputs, and the second cell from the right one has at most 2 outputs. So, the total number of LUTs is at most $3 \times (\lceil \frac{n-4}{3} \rceil - 1) + 2 + 1 = (n-4) - 3 + 3 = n - 4$.

When $n = 3r + 2$, the rightmost cell has two external inputs. So, the total number of LUTs is at most $3 \times (\lceil \frac{n-5}{3} \rceil - 1) + 3 + 1 = (n-5) - 3 + 4 = n - 4$. (Q.E.D.)

Theorem 2.6

- *The number of 6-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 40$ is $12n - 93$ or less, where $n \geq 9$.*
- *The number of 7-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 80$ is $14n - 124$ or less, where $n \geq 10$.*

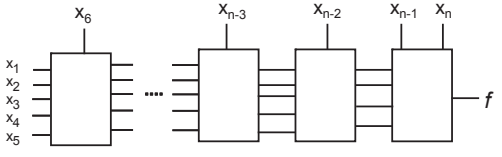

Figure 2.6. Realization with 6-LUTs.

Table 3.1. Profiles of 10-variable functions with weight 512.

NF	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
f_0	2	4	8	16	32	64	98	16	4	2
f_1	2	4	8	16	32	64	106	16	4	2
f_2	2	4	8	16	32	64	99	16	4	2
f_3	2	4	8	16	32	64	100	16	4	2
f_4	2	4	8	16	32	64	97	16	4	2
f_5	2	4	8	16	32	64	102	16	4	2
f_6	2	4	8	16	32	64	97	16	4	2
f_7	2	4	8	16	32	64	95	16	4	2
f_8	2	4	8	16	32	64	98	16	4	2
f_9	2	4	8	16	32	64	104	16	4	2
AVG	2.0	4.0	8.0	16.0	32.0	64.0	99.6	16.0	4.0	2.0

Table 3.2. Profiles of 10-variable functions with weight 64.

NF	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
f_0	2	4	8	16	26	31	21	12	4	2
f_1	2	4	8	16	28	27	22	10	4	2
f_2	2	4	8	16	26	30	20	10	4	2
f_3	2	4	8	16	27	30	19	11	4	2
f_4	2	4	8	16	27	28	21	9	4	2
f_5	2	4	8	16	27	30	19	8	4	2
f_6	2	4	8	16	29	30	19	8	3	2
f_7	2	4	8	16	29	28	19	10	4	2
f_8	2	4	8	16	28	29	20	9	4	2
f_9	2	4	8	16	28	31	18	9	4	2
AVG	2.0	4.0	8.0	16.0	27.5	29.4	19.8	9.6	3.9	2.0

- *The number of 6-LUTs needed to realize an arbitrary n-variable function with $\mu(f) \leq 160$ is $16n - 159$ or less, where $n \geq 11$.*

The proof is omitted due to the space limitation.

3 Logic Functions with Specified Weights

This part derives an upper bound on $\mu(f)$, the C-measure of a logic function f, whose weight u is specified. For most n-variable functions, C-measures increase exponentially with n. However, for the functions with a fixed weight u, $(u << 2^n)$, C-measure increase with $O(n)$. Thus, functions with small weights have small C-measures.

Definition 3.1 *The* **weight** *of a function f, denoted by u, is the number of the binary vectors $\vec{a}$ such that $f(\vec{a}) = 1$.*

Example 3.1 *Table 3.1 shows the profiles of 10 randomly generated functions with $n = 10$ and $u = 512$. The last row (AVG) denotes the average of the column multiplicities. The profile obtained by Theorem 2.1 is*

$$(2, 4, 8, 16, 32, 64, 128, 16, 4, 2).$$

Except for μ_7, the values of profiles in Table 3.1 are equal to the upper bounds given by Theorem 2.1. Note that $\mu(f_1) = 106$, while $\mu(f_7) = 95$. (End of Example)

Example 3.2 *Table 3.2 shows the profiles of 10 randomly generated functions with $n = 10$ and $u = 64$. In this case, the bounds given by Theorems 2.1 are not tight for μ_5, μ_6, μ_7 and μ_8. (End of Example)*

In general, when the weight u of a function is much less than 2^{n-1}, the bounds on the column multiplicity given by Theorem 2.1 are not tight and are not so useful. However, if we know the weight u of the function, then we can derive tighter bounds. From here, we are going to derive tighter bound using a combinatorial argument.

Lemma 3.1 *Consider boxes arranged as a rectangle with t rows and many columns. Assume that we distribute u balls to these boxes so that each box has at most one ball. Let $\lambda(t, u)$ be the maximum number of distinct column patterns. Then,*

$$\lambda(t, u) = \sum_{i=0}^{\sigma} \binom{t}{i} + r,$$

where σ is the integer satisfying the relation:

$$\sum_{i=1}^{\sigma} i\binom{t}{i} \leq u < \sum_{i=1}^{\sigma+1} i\binom{t}{i},$$

and

$$r = \left\lfloor \frac{u - \sum_{i=1}^{\sigma} i\binom{t}{i}}{\sigma + 1} \right\rfloor.$$

Note that $\lambda(t, u)$ is monotone increasing for $0 \leq u \leq t \cdot 2^{t-1}$, and takes the constant value 2^t when $u \geq t \cdot 2^{t-1}$.

Example 3.3 *Consider boxes arranged as a rectangle with $t = 4$ rows and many columns. Assume that we distribute $u = 10$ balls so that each box has at most one ball. When the balls are distributed as shown in Fig. 3.1, the maximum number of patterns occur. In this case, the first column has no ball; in the second to fifth columns, each column has just one ball; and in the last three columns, each column has two balls. The number of different column patterns can be enumerated as follows. Since,*

$$\sum_{i=1}^{1} i\binom{4}{i} = 4 \leq 10 < \sum_{i=1}^{2} i\binom{4}{i} = 16,$$

we have $\sigma = 1$, which shows that all the column patterns with weight 0 and weight 1 occur. Also,

$$r = \left\lfloor \frac{10 - \sum_{i=1}^{1} i\binom{4}{i}}{2} \right\rfloor = 3,$$

1	2	3	4	5	6	7	8
0	1	0	0	0	1	0	0
0	0	1	0	0	1	1	0
0	0	0	1	0	0	1	1
0	0	0	0	1	0	0	1

Figure 3.1. Maximum Number of column patterns for the function with weight $u = 10$

Table 3.3. Values for $\lambda(t, u)$

	t	u									
		23	47	55	95	111	147	191	223	239	375
μ_{n-2}	4	13	16	16	16	16	16	16	16	16	16
μ_{n-3}	8	**16**	28	**32**	47	52	**64**	79	90	94	**128**
μ_{n-4}	16	20	**32**	36	56	**64**	82	104	120	**128**	176
μ_{n-5}	32	24	40	44	**64**	72	90	112	**128**	136	204
μ_{n-6}	64	24	48	56	80	88	106	**128**	144	152	220

implies that there are three patterns with weight $\sigma + 1 = 2$. Thus, we have

$$\lambda(4, 10) = \sum_{i=0}^{1} \binom{4}{i} + r = 1 + 4 + 3 = 8,$$

which shows the number of different column patterns. *(End of Example)*

Example 3.4 *Table 3.3 shows the values of $\lambda(t, u)$ for $t = 4, 8, 16, 32$ and various values of u.* *(End of Example)*

Theorem 3.1 *Let $\mu_k(n, u)$ be the column multiplicity of a decomposition table for an n-variable function with weight u and k bound variables. Then,*

$$\mu_k(n, u) \leq \lambda(2^{n-k}, u).$$

Note that when $u \leq 2^{n-k}$, $\lambda(t, u) = u + 1$, while when $u > 2^{n-k}$, $\lambda(2^{n-k}, u) < u + 1$. Thus, we have

Corollary 3.1 *For any logic function f with weight u, $\mu(f) \leq u + 1$.*

For $\mu_k(n, u)$, Theorem 3.1 gives better bounds than $2^{2^{n-k}}$ and Corollary 3.1 when

$$2^{n-k} < u < 2^{2^{n-k}+(n-k)-1}.$$

From Theorem 3.1, we have the following:

Theorem 3.2 *The number of 6-LUTs needed to realize an n-variable function is:*

- *10 or less, when n=9 and $u \leq 55$; and*
- *15 or less, when n=10 and $u \leq 47$.*

(Proof) The profile of a 9-variable function given by Theorem 2.1 is

$$(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9)$$
$$= (2, 4, 8, 16, 32, 64, 16, 4, 2).$$

Let $n = 9$ and $u = 55$. By Theorem 3.1, we have $\mu_{n-3} = \mu_6 \leq \lambda(2^{n-6}, 55) = 32$. Thus, $\mu(f) \leq 32$ and by Theorem 2.3, f can be realized with at most 10 LUTs.

The profile of a 10-variable function given by Theorem 2.1 is

$$(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9, \mu_{10})$$
$$= (2, 4, 8, 16, 32, 64, 128, 16, 4, 2).$$

Let $n = 10$ and $u = 47$. By Theorem 3.1, we have $\mu_{n-3} = \mu_7 \leq \lambda(2^{n-7}, 47) = 28$, and $\mu_{n-4} = \mu_6 \leq \lambda(2^{n-6}, 47) = 32$. Thus, $\mu(f) \leq 32$, and by Theorem 2.3, f can be realized with at most 15 LUTs. (Q.E.D.)

Theorem 3.3 *The number of 7-LUTs needed to realize an n-variable function is:*

- *23 or less, when $n = 12$ and $u \leq 95$;*
- *17 or less, when $n = 11$ and $u \leq 111$; and*
- *11 or less, when $n = 10$ and $u \leq 147$.*

(Proof) From Theorems 2.1 and 3.1, and Table 3.3, we have the following profiles for 7-LUTs:
When $u \leq 95$, $(2, 4, 8, 16, 32, 64, 64, 56, 47, 16, 4, 2)$.
When $u \leq 111$, $(2, 4, 8, 16, 32, 64, 64, 52, 16, 4, 2)$.
When $u \leq 147$, $(2, 4, 8, 16, 32, 64, 64, 16, 4, 2)$.
In a similar way to the proof of Theorem 3.2, we have the numbers of LUTs. (Q.E.D.)

Theorem 3.4 *The number of 8-LUTs needed to realize an n-variable function is:*

- *33 or less, when $n = 14$ and $u \leq 191$;*
- *26 or less, when $n = 13$ and $u \leq 223$;*
- *19 or less, when $n = 12$ and $u \leq 239$; and*
- *12 or less, when $n = 11$ and $u \leq 375$.*

(Proof) For 8-LUTs, we have the following profiles:
When $u \leq 119$,
$(2, 4, 8, 16, 32, 64, 128, 128, 112, 104, 79, 16, 4, 2)$.
When $u \leq 223$, $(2, 4, 8, 16, 32, 64, 128, 128, 120, 90, 16, 4, 2)$.
When $u \leq 239$, $(2, 4, 8, 16, 32, 64, 128, 128, 94, 16, 4, 2)$.
When $u \leq 375$, $(2, 4, 8, 16, 32, 64, 128, 128, 16, 4, 2)$.
In a similar way to the proof of Theorem 3.2, we have the numbers of LUTs. (Q.E.D.)

It is interesting to compare the quality of bounds derived in this section with previous ones.

Theorem 3.5 *[6, 11] The number of K-LUTs to implement an arbitrary n-variable function is at most*

$$2^{n-K+1} - 1.$$

When n is even, and $K = 6$, we have a better bound as follows:

Theorem 3.6 *[11] Let n be even. The number of 6-LUTs to implement an arbitrary n-variable function is at most*

$$(2^{n-4} - 1)/3.$$

Theorem 3.7 *[6] Let f be represented by a sum-of-products expression with m literals and p products. Then, the number of K-LUTs to implement f is at most*

$$\left\lfloor \frac{m + p(K-3)}{K-1} \right\rfloor + \left\lceil \frac{p-1}{K-1} \right\rceil.$$

Example 3.5 *Consider the case of $K = 6$, $n = 10$ and $u = 47$. The bound given by Theorem 3.6 is*

$$(2^{n-4} - 1)/3 = 63/3 = 21.$$

On the other hand, the bound given by Theorem 3.2 is 15.

When the function is random, we can assume that $p = u = 47$, since most minterms are prime implicants [9]. Thus, we have $m = pn = 47 \times 10$. In this case, the bound given by Theorem 3.7 is

$$\left\lfloor \frac{m + p(K-3)}{K-1} \right\rfloor + \left\lceil \frac{p-1}{K-1} \right\rceil = 122 + 10 = 132.$$

This example shows that the bound given by Theorem 3.7 is useless for random functions. (End of Example)

4 Uniformly Distributed Functions

In the previous section, we considered upper bounds on column multiplicities. In this section, however, we consider upper bounds on **the average column multiplicities**. These bounds are only valid when n and u are sufficiently large. It is assumed that 1's in the truth table occur randomly.

Definition 4.1 *A set of functions is* **uniformly distributed**, *if the probability of occurrence of any function is the same as any other function.*

For example, there are $\binom{16}{4} = 1820$ different 4-variable functions with 4 true minterms. If the functions are uniformly distributed, the probability of the occurrence of any one of them is $\frac{1}{1820}$.

Theorem 4.1 *Consider a decomposition table with k bound variables that realizes a set of uniformly distributed functions of n-variables with weight u. The average number of different column functions with weight i in the decomposition table is at most*

$$\min\{1, N\alpha^i\beta^{M-i}\} \times \binom{M}{i},$$

where $N = 2^k$, $M = 2^{n-k}$, $\alpha = u/2^n$, and $\beta = 1 - \alpha$.

(Proof) Consider a column of the decomposition table. Suppose that the upper i elements take value 1, while the lower $M - i$ elements take value 0. The probability of such a column is given by

$$\alpha^i\beta^{M-i}.$$

Since there exist $\binom{M}{i}$ different ways to choose the rows with 1, the probability that a column function with weight i occur is

$$\binom{M}{i}\alpha^i\beta^{M-i}.$$

Also, the number of different column functions with weight i is at most $\binom{M}{i}$. Thus, we have the theorem. (Q.E.D.)

Note that Theorem 4.1 gives upper bounds on the **average** column multiplicity μ_k. Thus, there may exist functions whose column multiplicities are greater than the bounds given by Theorem 4.1. However, we conjecture that the fraction of such functions is small.

5 Experimental Results

5.1 Benchmark Functions

An interesting question is whether the bounds obtained in Section 3 are applicable to benchmark functions. The answer is yes when the weights of benchmark functions are in a some range. Although fractions of such functions are not so large, we could found some. For selected benchmark functions, we counted the number of variables n, and the number of true minterms u. For multiple output functions, each output are examined separately. f_i denotes, the i-th output, where the index starts from 0. The results are as follows:

Theorem 3.2 is applicable to the following functions: apex4, f_1 ($n = 9, u = 55$); pdc, f_{23} ($n = 9, u = 43$); spla, f_{43} ($n = 10, u = 28$); spla, f_{44} ($n = 10, u = 44$); amd, f_{14} ($n = 10, u = 44$).

Theorem 3.3 is applicable to the following functions: pdc, f_9 ($n = 10, u = 95$); signet, f_3 ($n = 10, u = 95$); pdf, f_{31}, ($n = 10, u = 120$).

Theorem 3.4 is applicable to the following functions: pdc, f_8 ($n = 11, u = 333$); pdc, f_{33} ($n = 11, u = 340$); signet, f_4 ($n = 11, u = 132$); in2, f_3 ($n = 12, u = 236$); ti, f_5 ($n = 13, u = 160$).

Table 5.1. Average and maximum profiles of 10-variable functions with weight 512.

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
AVG	2	4	8	16.00	32.00	63.96	100.86	16	4	2
MAX	2	4	8	16	32	64	111	16	4	2
$TH41$	2	4	8	16	32	64	128	16	4	2
$TH21$	2	4	8	16	32	64	128	16	4	2

Table 5.2. Average and maximum profiles of 10-variable functions with weight 64.

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
AVG	2	4	8	15.97	27.97	29.89	18.36	8.76	3.88	2
MAX	2	4	8	16	32	37	25	13	4	2
$TH41$	2	4	8	16	27.44	33.75	19.88	10.51	4	2
$TH21$	2	4	8	16	32	64	128	16	4	2

5.2 Randomly Generated Functions

We developed a program to derive the bounds on the column multiplicities given by Theorems 2.1 and 4.1. Also, we have obtained statistical data for functions with $n = 10$ and $n = 16$.

5.2.1 10-variable Functions

First, we randomly generated 100 functions with $u = 512$ and $n = 10$. Table 5.1 shows the average column multiplicities (AVG), the maximum column multiplicities (MAX), the upper bound derived by Theorem 4.1 (TH41), and the upper bound derived by Theorem 2.1 (TH21) for each μ_k. In this case, except for μ_6 and μ_7, Theorem 2.1 gives tight bounds. Also, Theorems 2.1 and 4.1 give identical bounds.

Second, we randomly generated 10000 functions with $u = 64$ and $n = 10$. Table 5.2 shows the average column multiplicities (AVG), the maximum column multiplicities (MAX), the upper bounds derived by Theorem 4.1(TH41), and the upper bounds derived by Theorem 2.1(TH21) for each μ_k. In this case, for μ_k, where $k = 5, 6, 7, 8$, Theorem 4.1 gives better bounds than Theorem 2.1.

Table 5.2 also shows that the average C-measure of 10-variable functions with weight 64, is less than 30. The maximum column multiplicity occurs when $k = 6$ and $\mu_6 = 37$. In these experimental results, the orderings of the input variables are fixed. If we optimize the orderings of the variables, then we can reduce the column multiplicity. We also optimized the ordering of the variables and confirmed that, in all 10000 cases, the column multiplicities are at most 32. Thus, we have the following:

Conjecture 5.1 *For most 10-variable functions with weight $u \leq 64$, the C-measure is 32 or less, if we optimize the ordering of the input variables.*

From Theorem 2.3, we have

Conjecture 5.2 *The number of 6-LUTs needed to realize most 10-variable functions with weight $u \leq 64$ is 15 or less.*

As shown in Theorem 3.2, the number of 6-LUTs needed to realize an arbitrary 10-variable function f with weight $u \leq 47$ is 15 or less. Thus, Conjecture 5.2 is true for $u \leq 47$.

5.2.2 16-variable Functions

In the case of $n = 16$, for each weight $u = 2^i$, where $i = 4$ to 15, we generated 100 functions and obtained the average of profiles. Table 5.3 compares the profiles of random functions and the average profiles derived by Theorem 4.1. For example, the row for $A16$ denotes the average profile of randomly generated functions with weight 16, while the row for C16 denotes the calculated profile using Theorem 4.1. To save the space, some fractional numbers are denoted by integers: That is, 2.00 is denoted by 2.

Table 5.3, shows that when u is small (say $u = 16$), Theorem 4.1 gives better bounds than Theorem 2.1, while when u is large (say $u = 32768$), Theorem 2.1 gives better bounds. In Table 5.3, calculated bounds denoted by integers were obtained by Theorem 2.1, while calculated bounds denoted by fractional numbers were obtained by Theorem 4.1.

6 Concluding Remarks

In this paper, we derived upper bounds on the column multiplicity of a decomposition table for logic functions with weight u. By using a combinatorial argument, we derived the following results:
The number of 6-LUTs needed to realize functions is

1. 10 or less, when $n = 9$ and $u \leq 55$.
2. 15 or less, when $n = 10$ and $u \leq 47$.

We also derived upper bounds on average column multiplicities, and obtained the following:

3. The number of 6-LUTs needed to realize most functions is 15 or less, when $n = 10$ and $u \leq 64$.

In this paper, for simplicity, we assumed that $u < 2^{n-1}$. However, when 0 and 1 are interchanged, the theory also holds. Thus, these bounds are also useful for the functions where the fraction of 0's in the truth table is much smaller than the fraction of 1's.

Table 5.3. Profiles of 16-variable functions

u	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{16}
A16	2	4	7.73	11.30	13.73	15.24	15.95	16.20	16.05	15.42	13.90	11.23	8.11	4.97	3	2
C16	2	4	7.50	11.06	13.56	15.16	16.04	16.51	16.75	16.88	16.94	16.97	9.01	5.01	3	2
A32	2	4	8	14.79	21.52	26.37	28.94	29.64	28.83	26.28	21.46	15.06	8.93	5.02	3.00	2
C32	2	4	8	13.99	21.11	26.18	29.32	31.09	32.03	32.51	32.76	17.12	9.05	5.02	3.01	2
A64	2	4	8	15.97	28.69	41.48	50.10	53.10	50.52	42.05	29.29	17.19	9.23	5.07	3.00	2
C64	2	4	8	16	26.99	41.23	51.36	57.65	61.19	63.07	33.95	17.48	9.22	5.09	3.03	2
A128	2	4	8	16	31.84	56.42	80.18	91.11	83.70	60.99	35.79	18.69	9.57	5.23	3.05	2
C128	2	4	8	16	32	53.00	81.49	101.76	114.35	72.27	36.73	18.84	9.87	5.37	3.13	2
A256	2	4	8	16	32	65.53	110.64	147.29	136.88	88.38	46.95	23.92	12.27	6.32	3.44	2
C256	2	4	8	16	32	63.84	105.05	162.09	174.95	91.84	47.34	24.23	12.45	6.49	3.50	2
A512	2	4	8	16	32	64	126.57	214.49	235.14	155.10	83.05	41.66	19.59	8.69	3.85	2
C512	2	4	8	16	32	64	126.69	209.33	262.51	156.60	86.08	44.89	22.57	10.94	4	2
A1024	2	4	8	16	32	64	128	251.86	391.99	327.91	192.81	89.33	33.93	11.07	4	2
C1024	2	4	8	16	32	64	128	252.46	407.99	332.19	214.99	120.75	38.65	11.25	4	2
A2048	2	4	8	16	32	64	128	256	499.80	658.92	455.91	176.87	48.40	12.72	4	2
C2048	2	4	8	16	32	64	128	256	504.20	626.67	568.03	188.19	49.43	12.95	4	2
A4096	2	4	8	16	32	64	128	256	512	975.44	1019.90	393.73	86.27	15.10	4	2
C4096	2	4	8	16	32	64	128	256	512	1008.54	1095.05	431.43	99.76	15.25	4	2
A8192	2	4	8	16	32	64	128	256	512	1023.92	1834.20	1055.60	150.52	15.98	4	2
C8192	2	4	8	16	32	64	128	256	512	1024	2020.45	1063.58	163.00	16	4	2
A16384	2	4	8	16	32	64	128	256	512	1024	2047.48	2738.27	238.61	16	4	2
C16384	2	4	8	16	32	64	128	256	512	1024	2048	4055.95	256	16	4	2
A32768	2	4	8	16	32	64	128	256	512	1024	2048	3971.75	256	16	4	2
C32768	2	4	8	16	32	64	128	256	512	1024	2048	4096	256	16	4	2

Acknowledgments

This work was supported in part by a Grant in Aid for Scientific Research of the JSPS, and Knowledge Cluster Project of MEXT. The author thanks Prof. Jon T. Butler for discussion and Mr. M. Matsuura for experiments.

References

[1] Altera, "Stratix IV FPGA Core Fabric Architecture," http://www.altera.com/

[2] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 3, March 2004, pp.288-298.

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

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

[5] V. Kravets and K. Sakallah, "Constructive library-aware synthesis using symmetries," *Proc. DATE'00*, pp. 208-213.

[6] R. Murgai, R. Brayton, and A. Sangiovanni Vincentelli, *Logic Synthesis for Field-Programmable Gate Arrays*, Springer, July 1995.

[7] J. Rose, R. J. Francis, D. Lewis, and P. Chow, "Architecture of field programmable gate arrays: The effect of logic block functionality on area efficiency,"*IEEE J. Solid State Circ. 25,5*, pp. 1217-1225, Oct. 1990.

[8] J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, "Architecture of field-programmable gate arrays,"*Proc. IEEE*, Vol. 81, No. 7, pp.1013-1029, July 1993.

[9] T. Sasao, "Bounds on the average number of products in the minimum sum-of-products expressions for multiple-valued input two-valued output functions," *IEEE Trans. on Comput.* Vol. 40, No. 5, pp. 645-651, May 1991.

[10] T. Sasao, "FPGA design by generalized functional decomposition," In *Logic Synthesis and Optimization*, Kluwer Academic Publisher, pages 233–258, 1993.

[11] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.

[12] T. Sasao, M. Matsuura, and Y. Iguchi, "A cascade realization of multiple-output function for reconfigurable hardware," *International Workshop on Logic and Synthesis(IWLS01),* Lake Tahoe, CA, June 12-15, 2001, pp.225-230.

[13] T. Sasao,"Analysis and synthesis of weighted-sum functions," *IEEE TCAD*, Vol. 25, No. 5, May 2006, pp. 789 - 796.

[14] I. Wegener, *Branching Programs and Binary Decision Diagrams: Theory and Applications*, Society for Industrial Mathematics, Jan. 1987.

[15] Xilinx, "Advantages of the Virtex-5 FPGA, 6-Input LUT Architecture," WP284 (v1.0), Dec. 19, 2007, http://www.xilinx.com/