IEICE TRANS. FUNDAMENTALS, VOL. E78–A, NO. 4 APRIL 1995

PAPER Special Section on Discrete Mathematics and Its Applications

Complexity of Boolean Functions Satisfying the Propagation Criterion

Shouichi HIROSE† and Katsuo IKEDA†, Members

SUMMARY Complexity of Boolean functions satisfying the propagation criterion (PC), an extended notion of the perfect nonlinearity, is discussed on several computation models. The following topics are investigated: (i) relationships between the unateness and the degree of the PC, (ii) the inversion complexity of perfectly nonlinear Boolean functions, (iii) the formula size of Boolean functions that satisfy the PC of degree 1, (iv) the area-time-square complexity of VLSI circuits computing perfectly nonlinear Boolean functions, (v) the OBDD size of perfectly nonlinear Boolean functions.

key words: *Boolean function, propagation criterion, unateness, inversion complexity, formula size, VLSI complexity, OBDD size*

1. Introduction

Nonlinearity is an important concept for the design of conventional cryptosystems. The propagation criterion (PC) [9], which is a generalized notion of the perfect nonlinearity [7], is a nonlinearity criteria of Boolean functions. The PC is a measure of randomness of the differences of output pairs to those of input pairs. It is one of the most important nonlinearity criteria because the differential cryptanalysis [1], which is one of the successful attacks to conventional cryptosystems, utilizes the bias of the distribution of the differences of output pairs and those of input pairs.

The aim of this paper is to characterize the PC in terms of the complexity of Boolean functions satisfying the criterion.

First, some relationships are presented between the unateness and the degree of the PC. It is shown that every Boolean function with four or more variables satisfying the PC of degree 1 is unate in at most two of its variables and that there exist Boolean functions with four or more variables that satisfy the PC of degree 1 and that are unate in two of their variables. It is also shown that every Boolean function with four or more variables satisfying the PC of degree 2 is not unate in any one of its variables.

Second, the inversion complexity [6] of perfectly nonlinear Boolean functions is discussed. The optimal lower bound $\lfloor \log n \rfloor - 1$ is obtained for the perfectly nonlinear Boolean functions with n variables constructed by the method of Maiorana [10].

Manuscript received September 10, 1994.
Manuscript revised November 21, 1994.

†The authors are with the Faculty of Engineering, Kyoto University, Kyoto-shi, 606–01 Japan.

Third, it is mentioned that the formula size of every Boolean function with n variables satisfying the PC of degree 1 is proved to be at least $n^2/4 - 1$ with the use of the method of Krapchenko [5]. This lower bound is nearly optimal for a perfectly nonlinear Boolean function.

Fourth, the area-time-square VLSI complexity [11] of perfectly nonlinear Boolean functions with multiple outputs is discussed. The main result of this topic is that, for every perfectly nonlinear Boolean function with n inputs and $n/2$ outputs, each of whose output functions is constructed by the method of Maiorana, the area-time-square complexity of any VLSI implementation requires $\Omega(n^2)$.

Finally, the size of ordered binary decision diagrams (OBDDs) [2] is considered. A relationship is presented between a combinatorial problem and the OBDD size of perfectly nonlinear Boolean functions in a subset of those each of whose output functions is constructed by the method of Maiorana. It is also mentioned that, for any variable ordering and for every perfectly nonlinear Boolean function with n inputs and $n/2$ outputs constructed by the method of Nyberg [8], there exist some output function of the perfectly nonlinear Boolean function such that the OBDD size of the output function is exponential in the number of its inputs.
puts.

Section 2 gives basic concepts and discusses Boolean functions satisfying the PC. The unateness and the inversion complexity are discussed in Sect. 3. Section 4 mentions the formula size. The area-time-square complexity of VLSI circuits and the OBDD size are considered in Sect. 5 and Sect. 6, respectively.

2. Preliminaries

2.1 Walsh Transform and Boolean Functions

Let $\mathbf{R}$ and $\mathbf{N}$ denote the set of reals and the set of integers, respectively.

Definition 1: The Walsh transform of a real-valued function $f : \{0,1\}^n \to \mathbf{R}$ is

$$(\mathcal{W}(f))(\omega) = \sum_{x \in \{0,1\}^n} f(x)(-1)^{\omega \cdot x},$$

where $x = (x_1, \ldots, x_n)$, $\omega = (\omega_1, \ldots, \omega_n) \in \{0,1\}^n$ and $\omega \cdot x = \omega_1 x_1 \oplus \cdots \oplus \omega_n x_n$. □

For simplicity, $(\mathcal{W}(f))(\omega)$ is often denoted by $F(\omega)$. The inverse Walsh transform is

$$f(x) = (\mathcal{W}^{-1}(F))(x) = \frac{1}{2^n} \sum_{\omega \in \{0,1\}^n} F(\omega)(-1)^{\omega \cdot x}.$$

The Walsh transform can be represented in a matrix form [10]. For $f : \{0,1\}^n \to \mathbf{R}$, let $f(i)$ denote $f(x_1, \ldots, x_n)$ when $x_1 + x_2 2 + \cdots + x_n 2^{n-1} = i$. Let $[f] = [f(0), f(1), \ldots, f(2^n - 1)]$ and $[F] = [F(0), F(1), \ldots, F(2^n - 1)]$. The Walsh transform is represented as

$$[F] = [f] H_n,$$

where H_n denotes the Hadamard matrix of order n. H_n is defined recursively by

$$H_0 = [1],$$

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}.$$

H_n is a $2^n \times 2^n$ symmetric non-singular matrix, and its inverse is $2^{-n} H_n$. The inverse Walsh transform is represented as

$$[f] = 2^{-n} [F] H_n.$$

A Boolean function is a function of the form $f : \{0,1\}^n \to \{0,1\}^m$. Let $\mathrm{B}_{n,m} = \{f \mid f : \{0,1\}^n \to \{0,1\}^m\}$. For simplicity, we denote $\mathrm{B}_{n,1}$ as B_n.

The Walsh transform can be applied to Boolean functions in B_n when they are considered to be real-valued functions. For the analysis of Boolean functions, it is often convenient to work with $\hat{f} : \{0,1\}^n \to \{-1, 1\}$, where $\hat{f}(x) \stackrel{\text{def}}{=} (-1)^{f(x)}$. The Walsh transform of $\hat{f}$ is

$$\hat{F}(\omega) = \sum_{x \in \{0,1\}^n} \hat{f}(x)(-1)^{\omega \cdot x} = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus \omega \cdot x}.$$

Proposition 1: For every $f \in \mathrm{B}_n$, $\sum_{\omega \in \{0,1\}^n} \hat{F}^2(\omega) = 2^{2n}$. □

Definition 2: The autocorrelation function of a Boolean function $f \in \mathrm{B}_n$ is $C_f : \{0,1\}^n \to \mathbf{N}$ such that

$$C_f(z) = \sum_{x \in \{0,1\}^n} \hat{f}(x) \hat{f}(x \oplus z),$$

where $x \oplus z$ denotes $(x_1 \oplus z_1, \ldots, x_n \oplus z_n)$. □

Proposition 2 shows a relationship between the autocorrelation function of f and the Walsh transform of $\hat{f}$.

Proposition 2: For every $f \in \mathrm{B}_n$, $C_f = \mathcal{W}^{-1}(\hat{F}^2)$. □

2.2 The Propagation Criterion

For a set S, let $|\mathrm{S}|$ denote the number of elements in S. For $f \in \mathrm{B}_{n,m}$ and $b \in \{0,1\}^m$, let $f^{-1}(b) = \{x \mid f(x) = b\}$.

Definition 3: A Boolean function $f \in \mathrm{B}_{n,m}$ is balanced if and only if $|f^{-1}(b)| = 2^{n-m}$ for every $b \in \{0,1\}^m$. □

For every $\{0,1\}$-vector a, let $W(a)$ be the Hamming weight of a, that is, the number of 1's in a.

Definition 4: $f \in \mathrm{B}_{n,m}$ is said to satisfy the propagation criterion (PC) of degree k if and only if $f(x) \oplus f(x \oplus a)$ is balanced for every $a \in \{0,1\}^n$ such that $1 \leqq W(a) \leqq k$. f is perfectly nonlinear if and only if f satisfies the PC of degree n. □

Let $\mathrm{PC}_{n,m}(k)$ denote the set of Boolean functions in $\mathrm{B}_{n,m}$ satisfying the PC of degree k. For simplicity, $\mathrm{PC}_{n,1}(k) = \mathrm{PC}_n(k)$.

The condition of the above definition states that

$$\Pr(f(x) \oplus f(x \oplus a) = b) = 1/2^m$$

for every $a \in \{0,1\}^n$ such that $1 \leqq W(a) \leqq k$ and every $b \in \{0,1\}^m$. This means that, for $f \in \mathrm{PC}_{n,m}(k)$, each of the outputs of f changes with probability $1/2$ if at most any k of inputs change and that the changes of the outputs are independent of each other. This sensitivity of the outputs to the inputs is desirable for cryptographic transformations.

The following proposition directly follows from the definition of the autocorrelation function and the PC.

Proposition 3: Let $f \in \mathrm{B}_n$. $f \in \mathrm{PC}_n(k)$ if and only if $C_f(a) = 0$ for every $a \in \{0,1\}^n$ such that $1 \leqq W(a) \leqq k$. □

For perfectly nonlinear Boolean functions, the following proposition was proved [7].

Proposition 4: Let $f \in \mathrm{B}_n$. f is perfectly nonlinear if and only if $\left|\hat{F}(\omega)\right| = 2^{n/2}$ for every $\omega \in \{0,1\}^n$. □

Nyberg [8] presented a necessary and sufficient condition for the perfect nonlinearity. Let $\mathrm{V}_n = \{0,1\}^n - \{(0, \ldots, 0)\}$.

Proposition 5 [8]: $f = (f_1, \ldots, f_m) \in \mathrm{B}_{n,m}$ is perfectly nonlinear if and only if, for every $c = (c_1, \ldots, c_m) \in \mathrm{V}_m$,

$$c \cdot f = c_1 f_1 \oplus \cdots \oplus c_m f_m$$

is perfectly nonlinear. □

Proposition 6 [8]: If $f = (f_1, \ldots, f_m) \in \mathrm{B}_{n,m}$ is perfectly nonlinear, then n is even and $m \leqq n/2$. □

A method of construction of perfectly nonlinear Boolean functions was presented by Maiorana [10]. Let $\pi \in \mathrm{B}_{k,k}$ be any permutation and $g \in \mathrm{B}_k$ be any Boolean function. Let $f \in \mathrm{B}_{2k}$ be represented as

$$f(x, y) = \pi(x) \cdot y \oplus g(x),$$

where $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$. Then, f is perfectly nonlinear.

Proposition 7: Let $n = 2k$. Let $f = (f_1, \ldots, f_m) \in \mathrm{B}_{n,m}$ and, for every i such that $1 \leqq i \leqq m$, f_i is represented as $f_i(x, y) = \pi_i(x) \cdot y \oplus g_i(x)$, where $\pi_i \in \mathrm{B}_{k,k}$ is

a permutation and $g_i \in \mathrm{B}_k$. Then, f is perfectly nonlinear if and only if, for every $c = (c_1, \ldots, c_m) \in \mathrm{V}_m$, $c_1\pi_1 \oplus \cdots \oplus c_m\pi_m$ is a permutation.

Proof: Suppose that $c_1\pi_1 \oplus \cdots \oplus c_m\pi_m \in \mathrm{B}_{k,k}$ is a permutation for every $c = (c_1, \ldots, c_m) \in \mathrm{V}_m$.

$$\begin{aligned} c \cdot f(x,y) &= c_1 f_1(x,y) \oplus \cdots \oplus c_m f_m(x,y) \\ &= (c_1\pi_1(x) \oplus \cdots \oplus c_m\pi_m(x)) \cdot y \oplus \\ &\quad (c_1 g_1(x) \oplus \cdots \oplus c_m g_m(x)). \end{aligned}$$

Thus, $c \cdot f$ is perfectly nonlinear.

From Proposition 4, f is perfectly nonlinear if and only if $c \cdot f$ is perfectly nonlinear for every $c = (c_1, \ldots, c_m) \in \mathrm{V}_m$. If $c \cdot f$ is perfectly nonlinear, then

$$\begin{aligned} &c \cdot f(x,y) \oplus c \cdot f(x, y \oplus b) \\ &\quad = (c_1\pi_1(x) \oplus \cdots \oplus c_m\pi_m(x)) \cdot b \end{aligned}$$

is balanced for every $b \in \mathrm{V}_k$. Thus, $c_1\pi_1 \oplus \cdots \oplus c_m\pi_m$ is a permutation for every $c = (c_1, \ldots, c_m) \in \mathrm{V}_m$. This completes the proof. □

Let $n = 2k$ and $m \leqq k$. Let

$$\mathrm{P}_{n,m} = \left\{ f \;\middle|\; \begin{array}{l} f = (f_1, \ldots, f_m) \in \mathrm{PC}_{n,m}(n), \text{ and,} \\ \text{for each } i \text{ such that } 1 \leqq i \leqq m, \\ f_i(x,y) = \pi_i(x) \cdot y \oplus g_i(x), \\ \text{where } \pi_i \in \mathrm{B}_{k,k} \text{ is a permutation} \\ \text{and } g_i \in \mathrm{B}_k. \end{array} \right\}.$$

For simplicity, $\mathrm{P}_{n,1} = \mathrm{P}_n$.

3. Unateness and Inversion Complexity

3.1 Unateness

In this subsection, some relationships between the unateness and the degree of the PC are presented. We begin by defining the unate functions [4].

Definition 5: A Boolean function $f(x_1, \ldots, x_n) \in \mathrm{B}_n$ is said to be positive (negative) in variables $x_{i_1}, \ldots, x_{i_k}$ if there exists a disjunctive or conjunctive expression of f in which $x_{i_1}, \ldots, x_{i_k}$ appears only in uncomplemented (complemented) form. If f is positive (negative) in all of its variables, then f is simply said to be positive (negative). □

Definition 6: A Boolean function $f(x_1, \ldots, x_n) \in \mathrm{B}_n$ is said to be unate in variables $x_{i_1}, \ldots, x_{i_k}$ if f is positive or negative in each one of $x_{i_1}, \ldots, x_{i_k}$. If f is unate in all of its variables, then f is simply said to be unate. □

For example, $f(x_1, x_2, x_3) = x_1 x_2 \vee \overline{x_2}\,\overline{x_3}$ is unate in x_1 and x_3 and not unate in x_2.

Figures 1 and 2 give all Boolean functions in $\mathrm{PC}_2(2)$ and $\mathrm{PC}_3(2)$, respectively. From these figures, it is observed that both $\mathrm{PC}_2(2)$ and $\mathrm{PC}_3(2)$ contain positive, negative and unate functions.

$x_1 x_2$	$x_1 \vee x_2$
$\overline{x_1} x_2$	$\overline{x_1} \vee x_2$
$x_1 \overline{x_2}$	$x_1 \vee \overline{x_2}$
$\overline{x_1}\,\overline{x_2}$	$\overline{x_1} \vee \overline{x_2}$

Fig. 1 Boolean functions in $\mathrm{PC}_2(2)$.

$x_1 x_2 \vee x_1 x_3 \vee x_2 x_3$	$\overline{x_1}\,\overline{x_2} \vee \overline{x_1}\,\overline{x_3} \vee \overline{x_2}\,\overline{x_3}$
$x_1 x_2 \vee x_1 \overline{x_3} \vee x_2 \overline{x_3}$	$\overline{x_1}\,\overline{x_2} \vee \overline{x_1} x_3 \vee \overline{x_2} x_3$
$x_1 \overline{x_2} \vee x_1 x_3 \vee \overline{x_2} x_3$	$\overline{x_1} x_2 \vee \overline{x_1}\,\overline{x_3} \vee x_2 \overline{x_3}$
$\overline{x_1} x_2 \vee \overline{x_1} x_3 \vee x_2 x_3$	$x_1 \overline{x_2} \vee x_1 \overline{x_3} \vee \overline{x_2}\,\overline{x_3}$
$x_1 \overline{x_2}\,\overline{x_3} \vee \overline{x_1} x_2 x_3$	$\overline{x_1}\,\overline{x_2} \vee x_1 x_3 \vee x_2 \overline{x_3}$
$x_1 \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3}$	$\overline{x_1}\,\overline{x_2} \vee \overline{x_1} x_3 \vee x_1 x_2$
$x_1 x_2 \overline{x_3} \vee \overline{x_1}\,\overline{x_2} x_3$	$\overline{x_1} x_2 \vee x_1 x_3 \vee \overline{x_2}\,\overline{x_3}$
$\overline{x_1}\,\overline{x_2}\,\overline{x_3} \vee x_1 x_2 x_3$	$x_1 \overline{x_2} \vee \overline{x_1} x_3 \vee x_2 \overline{x_3}$

Fig. 2 Boolean functions in $\mathrm{PC}_3(2)$.

Suppose that $f(x_1, \ldots, x_n) \in \mathrm{B}_n$ is unate in x_n. Then, for every $(a_1, \ldots, a_{n-1}) \in \{0,1\}^{n-1}$, $f(a_1, \ldots, a_{n-1}, 0) = 0$ if $f(a_1, \ldots, a_{n-1}, 1) = 0$ or $f(a_1, \ldots, a_{n-1}, 1) = 0$ if $f(a_1, \ldots, a_{n-1}, 0) = 0$. This regularity does not seem compatible with the PC. In the following, this conjecture is shown to be correct for $f \in \mathrm{B}_n$ when $n \geqq 4$. Before showing the results, several lemmas are presented.

Lemma 1 [3]: Let w, x, y, z and m be integers such that $w \geqq x \geqq y \geqq z \geqq 0$ and $m \geqq 0$. Let $w^2 + x^2 + y^2 + z^2 = 2^m$. Then,

- for even m, $w = x = y = z = 2^{(m-2)/2}$, or $w = 2^{m/2}$ and $x = y = z = 0$,
- for odd m, $w = x = 2^{(m-1)/2}$ and $y = z = 0$. □

For $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ and i such that $1 \leqq i \leqq n$, let $\langle x \rangle_i = (x_1, \ldots, x_i)$.

Lemma 2: Let $n \geqq 2$ and $f \in \mathrm{PC}_n(1)$. Then, for every i such that $1 \leqq i \leqq n$, $f(x_1, \ldots, x_n)$ is positive in x_i if and only if $\hat{F}(0, \ldots, 0, \overset{i}{\check{1}}, 0, \ldots, 0) = 2^{n-1}$.

Proof: Suppose that $i = n$. Since $f \in \mathrm{PC}_n(1)$, $f(x_1, \ldots, x_{n-1}, 0) \oplus f(x_1, \ldots, x_{n-1}, 1)$ is balanced,that is,

$$|\{\langle x \rangle_{n-1} \mid f|_{x_n=0} \neq f|_{x_n=1}\}| = 2^{n-2}.$$

Thus,

$$\begin{aligned} &\hat{F}(0, \ldots, 0, 1) \\ &\quad = \sum_{x \in \{0,1\}^n} \hat{f}(x)(-1)^{x_n} \\ &\quad = \sum_{\langle x \rangle_{n-1} \in \{0,1\}^{n-1}} (\hat{f}|_{x_n=0} - \hat{f}|_{x_n=1}) \\ &\quad = 2|\{\langle x \rangle_{n-1} \mid f|_{x_n=0} = 0, f|_{x_n=1} = 1\}| \\ &\qquad -2|\{\langle x \rangle_{n-1} \mid f|_{x_n=0} = 1, f|_{x_n=1} = 0\}| \\ &\quad = 2^{n-1} - 4|\{\langle x \rangle_{n-1} \mid f|_{x_n=0} = 1, f|_{x_n=1} = 0\}|. \end{aligned}$$

Hence, $\hat{F}(0,\ldots,0,1) = 2^{n-1}$ if and only if $f(x)$ is positive in x_n. The same argument can be applied to the case where $1 \leqq i \leqq n-1$. □

The following lemma can be proved in the same way as Lemma 2.

Lemma 3: Let $n \geqq 2$ and $f \in \mathrm{PC}_n(1)$. Then, for every i such that $1 \leqq i \leqq n$, $f(x_1,\ldots,x_n)$ is negative in x_i if and only if $\hat{F}(0,\ldots,0,\overset{i}{1},0,\ldots,0) = -2^{n-1}$. □

Lemma 4: Let $f \in \mathrm{B}_n$. Then, $f \in \mathrm{PC}_n(k)$ if and only if

$$\sum_{a\cdot\omega=0} \hat{F}^2(\omega) = \sum_{a\cdot\omega=1} \hat{F}^2(\omega) = 2^{2n-1}$$

for every $a \in \{0,1\}^n$ such that $1 \leqq W(a) \leqq k$.

Proof: This lemma is easily derived from Proposition 1, 2 and 3. □

Theorem 1: Let $n \geqq 4$. If $f \in \mathrm{PC}_n(1)$, then f is unate in at most two of its variables.

Proof: This theorem is derived from Lemma 2, 3 and 4. □

The optimality of the above result can also be proved.

Theorem 2: Let $n \geqq 4$. There exist Boolean functions in $\mathrm{PC}_n(1)$ that are unate in two of their variables.

Proof: Suppose that $f \in \mathrm{PC}_n(1)$. Then, f is unate in x_1 and x_2 if and only if

$$\hat{F}^2(1,0,0,\ldots,0) = \hat{F}^2(0,1,0,\ldots,0) = 2^{2(n-1)}.$$

Since $\sum_{\omega_i=0} \hat{F}^2(\omega) = 2^{2n-1}$ for every i such that $3 \leqq i \leqq n$,

$$\omega \notin \left\{ \begin{array}{ll} (1,0,0,\ldots,0), & (0,1,0,\ldots,0), \\ (1,0,1,\ldots,1), & (0,1,1,\ldots,1), \\ (0,0,1,\ldots,1), & (1,1,1,\ldots,1) \end{array} \right\}$$

$$\Rightarrow \hat{F}(\omega) = 0.$$

Let

$$\begin{array}{ll} \hat{F}(1,0,0,\ldots,0) = \hat{F}_1, & \hat{F}(0,1,0,\ldots,0) = \hat{F}_0, \\ \hat{F}(1,0,1,\ldots,1) = \hat{F}_{10}, & \hat{F}(0,1,1,\ldots,1) = \hat{F}_{01}, \\ \hat{F}(0,0,1,\ldots,1) = \hat{F}_{00}, & \hat{F}(1,1,1,\ldots,1) = \hat{F}_{11}. \end{array}$$

Since $\sum_{\omega_i=0} \hat{F}^2(\omega) = \sum_{\omega_i=1} \hat{F}^2(\omega) = 2^{2n-1}$ for $i = 1, 2$,

$$\hat{F}_{00}^2 + \hat{F}_{01}^2 = \hat{F}_{10}^2 + \hat{F}_{11}^2 = \hat{F}_{00}^2 + \hat{F}_{10}^2 = \hat{F}_{01}^2 + \hat{F}_{11}^2 = 2^{2(n-1)}.$$

From Lemma 1, there are following two cases:

C-1. $\left|\hat{F}_{10}\right| = \left|\hat{F}_{01}\right| = 2^{n-1}$, $\left|\hat{F}_{00}\right| = \left|\hat{F}_{11}\right| = 0$,

C-2. $\left|\hat{F}_{00}\right| = \left|\hat{F}_{11}\right| = 2^{n-1}$, $\left|\hat{F}_{10}\right| = \left|\hat{F}_{01}\right| = 0$.

For C-1, let

$$b^1 = (1,0,0,\ldots,0), \quad b^2 = (0,1,0,\ldots,0)$$
$$b^3 = (1,0,1,\ldots,1), \quad b^4 = (0,1,1,\ldots,1).$$

Then, $\left[\hat{f}\right]$ can be represented as

$$\left[\hat{f}\right] = \frac{1}{2^n}\left[\hat{F}\right] H_n$$
$$= \frac{1}{2^n}\left(\hat{F}_1\left[\hat{l}_{b^1}\right] + \hat{F}_0\left[\hat{l}_{b^2}\right] + \hat{F}_{10}\left[\hat{l}_{b^3}\right] + \hat{F}_{01}\left[\hat{l}_{b^4}\right]\right).$$

Since $b^1 \oplus b^2 \oplus b^3 \oplus b^4 = (0,\ldots,0)$, for every $x \in \{0,1\}^n$, an even number of $\hat{l}_{b^1}(x)$, $\hat{l}_{b^2}(x)$, $\hat{l}_{b^3}(x)$ and $\hat{l}_{b^4}(x)$ are equal to 1, and the others are equal to -1. Thus, an odd number of $\hat{F}_0$, $\hat{F}_1$, $\hat{F}_{10}$ and $\hat{F}_{01}$ are equal to 2^{n-1} and the others are equal to -2^{n-1} since $f \in \mathrm{B}_n$. Conversely, if an odd number of $\hat{F}_0$, $\hat{F}_1$, $\hat{F}_{10}$ and $\hat{F}_{01}$ are equal to 2^{n-1} and the others are equal to -2^{n-1}, then $f \in \mathrm{PC}_n(1)$ and $f(x)$ is unate in x_1 and x_2.

The same argument as the above one can be applicable to C-2. □

The following example gives a method of the construction for Boolean functions that satisfy the PC of degree 1 and that are unate in two of their variables.

Example 1: We construct $f \in \mathrm{PC}_4(1)$ which is negative in x_1 and positive in x_2. Let

$$\left[\hat{F}\right] = [0,-8,8,0,0,0,0,0,0,0,0,0,8,0,0,8].$$

Then,

$$\left[\hat{f}\right] \frac{1}{2^4}\left[\hat{F}\right] H_4$$
$$= [1,1,-1,1,-1,1,-1,-1,-1,1,-1,-1,1,1,-1,1].$$

Thus,

$$f(x_1,x_2,x_3,x_4)$$
$$= \overline{x_1}x_2 \vee \overline{x_1}\,\overline{x_3}x_4 \vee \overline{x_1}x_3\overline{x_4} \vee x_2\overline{x_3}x_4 \vee x_2x_3\overline{x_4}.$$

The method can be applicable to every $n \geqq 4$. □

The following theorem shows the non-unateness of the Boolean functions satisfying the PC of degree 2.

Theorem 3: Let $n \geqq 4$. If $f \in \mathrm{PC}_n(2)$, then f is not unate in any one of its variables.

Proof: Suppose that $f \in \mathrm{PC}_n(2)$ is unate in x_1. Then,

$$\hat{F}^2(1,0,\ldots,0) = 2^{2(n-1)}.$$

If $n = 4$, then, for every $f \in \mathrm{PC}_4(2)$, f is known to be perfectly nonlinear [9]. Thus, from Proposition 4, $\hat{F}^2(\omega) = 16$ for every $\omega \in \{0,1\}^4$, which is a contradiction.

Let $n \geqq 5$. Let i, j be any integers such that $2 \leqq i < j \leqq n$. Since $f \in \mathrm{PC}_n(2)$, $\sum_{\omega_i=0} \hat{F}^2(\omega) = 2^{2n-1}$, $\sum_{\omega_j=0} \hat{F}^2(\omega) = 2^{2n-1}$, and $\sum_{\omega_i\oplus\omega_j=1} \hat{F}^2(\omega) = 2^{2n-1}$. Thus,

$$\sum_{\omega_i=0} \hat{F}^2(\omega) + \sum_{\omega_j=0} \hat{F}^2(\omega) - 2\hat{F}^2(1,0,\ldots,0)$$

$$-\sum_{\omega_i \oplus \omega_j = 1} \hat{F}^2(\omega) = 0.$$

From this equation, if $\hat{F}(\omega) \neq 0$, then at most one of $\omega_2, \ldots, \omega_n$ is 0 or $\omega = (1, 0, \ldots, 0)$. Thus,

$$\begin{aligned}
&\sum_{\omega_i \oplus \omega_j = 1} \hat{F}^2(\omega) \\
&= \hat{F}^2(0, 1, \ldots, 1, \overset{i}{\check{0}}, 1, \ldots, 1) \\
&\quad + \hat{F}^2(0, 1, \ldots, 1, \overset{j}{\check{0}}, 1, \ldots, 1) \\
&\quad + \hat{F}^2(1, 1, \ldots, 1, \overset{i}{\check{0}}, 1, \ldots, 1) \\
&\quad + \hat{F}^2(1, 1, \ldots, 1, \overset{j}{\check{0}}, 1, \ldots, 1) \\
&= 2^{2n-1}.
\end{aligned}$$

From Lemma 1, for every $\omega \in \{0,1\}^n$ such that only one of $\omega_2, \ldots, \omega_n$ is 0 and $\hat{F}(\omega) \neq 0$, $\hat{F}^2(\omega) = 2^{2n-2}$. Since

$$\sum_{\omega_1 = 0} \hat{F}^2(\omega) = \sum_{\omega_1 = 1} \hat{F}^2(\omega) = 2^{2n-1}$$

and

$$\hat{F}^2(1, 0, \ldots, 0) = 2^{2(n-1)},$$

at most three of $\hat{F}^2(\omega)$'s are 2^{2n-2} for $\omega \in \{0,1\}^n$ such that only one of $\omega_2, \ldots, \omega_n$ is 0. On the other hand, since

$$\sum_{\omega_2 \oplus \omega_3 = 1} \hat{F}^2(\omega) = \sum_{\omega_4 \oplus \omega_5 = 1} \hat{F}^2(\omega) = 2^{2n-1},$$

at least four of $\hat{F}^2(\omega)$'s are 2^{2n-2} for $\omega \in \{0,1\}^n$ such that only one of $\omega_2, \ldots, \omega_n$ is 0. This causes a contradiction. Thus, the theorem has been proved. □

3.2 Inversion Complexity

Even if a Boolean function is not unate in many of its variables, combinational circuits computing the function do not necessarily require many ¬-gates. For example, $x_1 \cdots x_n \vee \overline{x_1} \cdots \overline{x_n}$ can be computed with one ¬-gate, while it is not unate in any one of its variables. This subsection shows that, for every perfectly nonlinear Boolean function in P_n, many ¬-gates are required by every combinational circuit consisting of ∧-gates, ∨-gates and ¬-gates that computes the function.

A combinational circuit is a logic circuit without any loop. Let B be a set of Boolean functions. A B-circuit is a combinational circuit consisting of the gates, each of which computes a function in B.

Definition 7: The inversion complexity [6] of a Boolean function f, $I(f)$, is the smallest number of ¬-gates necessary to compute f by $\{\wedge, \vee, \neg\}$-circuits. □

For $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ and $b = (b_1, \ldots, b_n) \in \{0,1\}^n$, $a \leqq b$ if and only if $a_i \leqq b_i$ for every i such that $1 \leqq i \leqq n$. $a < b$ if and only if $a \leqq b$ and $a_j < b_j$ for some j such that $1 \leqq j \leqq n$.

Definition 8 [6]: Let $C = (\alpha^1, \ldots, \alpha^k)$ be a sequence such that $\alpha^i \in \{0,1\}^n$ for $1 \leqq i \leqq k$ and $\alpha^j < \alpha^{j+1}$ for $1 \leqq j \leqq k-1$. C is called a sign-variable chain of length k of $f \in \mathrm{B}_n$ if and only if

$$f(\alpha^i) = \begin{cases} 1 & \text{if } i \text{ is odd,} \\ 0 & \text{if } i \text{ is even.} \end{cases}$$

for every i such that $1 \leqq i \leqq k$. □

Definition 9: $alt(f)$ is the length of the longest sign-variable chain of f. □

The inversion complexity $I(f)$ is completely characterized by $alt(f)$.

Lemma 5 [6]: For every $f \in \mathrm{B}_n$,

$$I(f) = \begin{cases} 0 & \text{if } alt(f) = 0 \\ \lfloor \log_2 alt(f) \rfloor & \text{otherwise.} \end{cases}$$

□

The following proposition is trivial from the definition of alt.

Proposition 8: For every $f \in \mathrm{B}_n$, $I(f) \leqq \lfloor \log_2(n+1) \rfloor$. □

Lemma 6: For every $f \in \mathrm{P}_n$, $I(f) \geqq \lfloor \log_2 n \rfloor - 1$.

Proof: Let $n = 2k$. $f(x, y) = \pi(x) \cdot y \oplus g(x)$, where $\pi \in \mathrm{B}_{k,k}$ be a permutation and $g \in \mathrm{B}_k$.

Since π is a permutation, for some $v \in \{0,1\}^k$, $\pi(v) = (1, \ldots, 1)$ and

$$f(v, y) = y_1 \oplus \cdots \oplus y_k \oplus g(v).$$

Let $C = (\alpha^1, \ldots, \alpha^k)$ such that $\alpha^i = (v, \underbrace{1, \ldots, 1}_{i}, 0, \ldots, 0) \in \{0,1\}^{2k}$ for every i such that $1 \leqq i \leqq k$. If $g(v) = 0$, then

$$f(\alpha^i) = \begin{cases} 1 & \text{if } i \text{ is odd,} \\ 0 & \text{if } i \text{ is even.} \end{cases}$$

and C is a sign-variable chain of f. It also can be shown that there exists a sign-variable chain of length $k+1$ of f if $g(v) = 1$. This completes the proof. □

Theorem 4: For every $f \in \mathrm{P}_n$, $\lfloor \log_2 n \rfloor - 1 \leqq I(f) \leqq \lfloor \log_2(n+1) \rfloor$. □

The following example shows that the bounds of Theorem 4 are optimal.

Example 2: Let $n = 2k$. Let

$$\begin{aligned}
f(x, y) &= x_1 y_1 \oplus \cdots \oplus x_k y_k, \\
g(x, y) &= x_1 y_1 \oplus \cdots \oplus x_k y_k \oplus x_1 \oplus \cdots \oplus x_k \oplus 1.
\end{aligned}$$

Then, $f, g \in \mathrm{P}_n$. $I(f) = \lfloor \log_2 n \rfloor - 1$ and $I(g) = \lfloor \log_2(n+1) \rfloor$. □

4. Formula Size

In this section, a lower bound on the formula size of any Boolean function in $\mathrm{PC}_n(1)$ is obtained with the use of the method of Krapchenko [5].

Let B be a set of Boolean functions. A B-formula is a B-circuit with fan-out 1.

Definition 10: The formula size of f on B, $L_{\mathrm{B}}(f)$, is the smallest number of gates of B-formulas computing f. □

Some notations are defined in the following definition.

Definition 11 [12]: Let $\mathrm{Q}, \mathrm{S} \subseteq \{0,1\}^n$ and $f \in \mathrm{B}_n$. Let $H(\mathrm{Q}, \mathrm{S})$ be the set of neighbors in (Q, S), i.e.,

$$H(\mathrm{Q}, \mathrm{S}) = \left\{ (q,s) \;\middle|\; \begin{array}{l} (q,s) \in (\mathrm{Q}, \mathrm{S}) \text{ and there exists} \\ \text{some } i \text{ such that } 1 \leqq i \leqq n, \\ q_i \neq s_i \text{ and } q_j = s_j \text{ for every } j \\ \text{such that } 1 \leqq j \leqq n \text{ and } j \neq i \end{array} \right\}.$$

Let

$$K_{\mathrm{Q,S}} = \frac{|H(\mathrm{Q}, \mathrm{S})|^2}{|\mathrm{Q}||\mathrm{S}|},$$

$$K(f) = \max\{K_{\mathrm{Q,S}} \mid \mathrm{Q} \subseteq f^{-1}(1), \mathrm{S} \subseteq f^{-1}(0)\}. \quad \square$$

Lemma 7 [12]: For every Boolean function f, $L_{\mathrm{U}}(f) \geqq K(f) - 1$, where $\mathrm{U} = \mathrm{B}_2 - \{\oplus, \equiv\}$. □

Theorem 5: For every $f \in \mathrm{PC}_n(1)$, $L_{\mathrm{U}}(f) \geqq n^2/4 - 1$.

Proof: Suppose that $f \in \mathrm{PC}_n(1)$. Then,

$$K(f) \geqq K_{f^{-1}(1), f^{-1}(0)} = \frac{(n2^{n-2})^2}{|f^{-1}(1)|(2^n - |f^{-1}(1)|)} \geqq n^2/4.$$

This completes the proof. □

The lower bound in Theorem 5 is almost optimal even for perfectly nonlinear Boolean functions. For the perfectly nonlinear Boolean function f in Example 2, $L_{\mathrm{U}}(f) \leqq n^2/2 - 1$ when n is a power of 2.

For the same f, $L_{\mathrm{B}_2}(f) = n - 1$. On the formula model, $\oplus$ and $\equiv$ is essential for the efficient computation of the Boolean functions satisfying the PC.

5. VLSI Complexity

This section gives some results on the area-time-square complexity of VLSI circuits computing $f \in \mathrm{P}_{n,m}$.

We adopt the grid model for VLSI circuits [11]. In the grid model, a rectangular grid is assumed. The spacing of the grid lines is some fixed constant. Wires run along grid lines. There are one or more layers and the number of them is some fixed constant. Each layer has at most one wire on every grid line. Circuit elements, such as input/output pads, contacts, logic elements and so on are located on grid points. All the circuits are assumed to be convex and the area of a circuit is defined to be that of the smallest rectangle whose sides are on grid lines and which covers the circuit. The unit of time is also assumed in this model. Each input/output appears on some input/output pad during a unit of time and signals propagate on wires in a unit of time. The computation time of a circuit is defined to be the number of units of time between the first input and the last output.

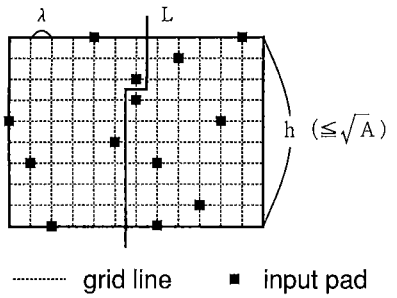


Fig. 3 A VLSI chip.

Lemma 8 [11]: Let I be any set of input variables and C be a VLSI chip. Let A be the area of C and λ be the spacing of grid lines of C. If no more than one third of the inputs in I are fed into an input pad of C, then a line of length at most $\sqrt{A} + \lambda$ can divide C into two parts each of which includes between one third and two thirds of the inputs in I. □

The proofs of the results make use of the information flow argument [11]. Let C be a VLSI chip of area A and time T computing some function, and let L be a line on C satisfying the condition of the above lemma (Fig. 3). If some of the outputs on a side of L depend on some of the inputs on the other side, then some amount of information must be transferred across L during the computation. If the amount of the information is proved to be at least J, then $(\sqrt{A} + \lambda)T \geqq J$. Thus, $AT^2 = \Omega(J^2)$.

Theorem 6: Let $n = 2k$ and $f \in \mathrm{P}_{n,k}$. For every VLSI circuit that computes f, $AT^2 = \Omega(n^2)$.

Proof: Let $f = (f_1, \ldots, f_k) \in \mathrm{P}_{n,k}$ such that, for $1 \leqq i \leqq k$,

$$f_i(x,y) = \pi_i(x) \cdot y \oplus g_i(x),$$

where $\pi_i = (\pi_{i,1}, \ldots, \pi_{i,k}) \in \mathrm{B}_{k,k}$ is a permutation and $g_i \in \mathrm{B}_k$.

If some input pad accepts $k/3 = n/6$ or more inputs in $y = (y_1, \ldots, y_k)$, then $T \geqq n/6$ and $AT^2 = \Omega(n^2)$.

Suppose that each input pad accept less than $k/3$ inputs in y. Then, Lemma 8 implies that the chip can be split in two parts by a line of length at most $\sqrt{A} + \lambda$ so that each of the sides has between $1/3$ and $2/3$ of the inputs in $y = (y_1, \ldots, y_k)$, where λ is the spacing of grid lines. Without loss of generality, it can be assumed that the left side of the chip contains at least half of the outputs. Let $r = \lceil k/3 \rceil$. Choose r of the inputs y on the right side of the chip and r of the outputs on the left side of the chip. Without loss of generality, we can assume that they are $y_1, \ldots, y_r$ and $f_1, \ldots, f_r$.

For $c \in \mathrm{V}_r$, let

$$\pi_0^c = (\pi_{0,1}^c, \ldots, \pi_{0,k}^c) = c_1\pi_1 \oplus \cdots \oplus c_r\pi_r.$$

Since, for every $c \in \mathrm{V}_r$, π_0^c is a permutation,

$$\left|\{x \mid \pi^c_{0,1}(x) = \cdots = \pi^c_{0,r}(x) = 0\}\right| = 2^{k-r}.$$

Thus,

$$\left|\{x \mid \exists c(\pi^c_{0,1}(x) = \cdots = \pi^c_{0,r}(x) = 0)\}\right| \leqq 2^{k-r}(2^r - 1) = 2^k - 2^{k-r}.$$

There exists $a \in \{0,1\}^k$ such that, for every $c \in \mathrm{V}_r$, there exists some j such that $1 \leqq j \leqq r$ and $\pi^c_{0,j}(a) \neq 0$. Hence,

$$(\pi_{1,1}(a), \ldots, \pi_{1,r}(a)), \ldots, (\pi_{r,1}(a), \ldots, \pi_{r,r}(a))$$

are linearly independent.

Thus, for every pair of $(y_1, \ldots, y_r), (y'_1, \ldots, y'_r) \in \{0,1\}^r$ such that $(y_1, \ldots, y_r) \neq (y'_1, \ldots, y'_r)$,

$$(f_1(a, y_1, \ldots, y_r, b), \ldots, f_r(a, y_1, \ldots, y_r, b)) \neq (f_1(a, y'_1, \ldots, y'_r, b), \ldots, f_r(a, y'_1, \ldots, y'_r, b)),$$

where $b \in \{0,1\}^{k-r}$.

From the above discussions, during the computation, at least r bits of information must be transferred across the line splitting the chip. Hence, $(\sqrt{A}+\lambda)T \geqq r$ and $AT^2 = \Omega(n^2)$. □

The proof of Theorem 6 can be extended to prove the following corollary.

Corollary 1: Let n be even and ϵ be a constant such that $0 < \epsilon \leqq 1/2$. Let $f \in \mathrm{P}_{n,\lceil \epsilon n \rceil}$. For every VLSI circuit that computes f, $AT^2 = \Omega((\epsilon n)^2)$. □

It is considered to be more realistic that input/output pads are located on the border of a VLSI chip. Under the restriction, the following theorem can also be proved.

Theorem 7: Any VLSI circuit computing $f \in \mathrm{P}_{n,m}$ that has all of its input/output pads on its border requires $AT^2 = \Omega(nm)$. □

The above result has some implication for the VLSI implementation of cryptographic transformations. For nonlinear elements in secure cryptographic transformations, their area-time-square VLSI complexity is expected to grow in proportion to the number of inputs and that of the outputs.

6. OBDD Size

In this section, we consider the OBDD size of perfectly nonlinear Boolean functions in a subset of $\mathrm{P}_{n,m}$.

A binary decision diagram (BDD) [2] is an acyclic directed graph that has one source node and two sink nodes. Each of the nodes except two sink nodes is labeled by an input variable and has two outgoing edges which are labeled by 0 and 1, respectively. Two sink nodes are labeled by 0 and 1, respectively. Each input variable appears at most once on each path from a source node to a sink node.

A BDD represents a Boolean function $f(x_1, \ldots, x_n) \in \mathrm{B}_n$ if, for every $(b_1, \ldots, b_n) \in \{0,1\}^n$, the path from the source node along each edge outgoing from a node labeled by x_i and labeled by b_i leads to the sink node labeled by $f(b_1, \ldots, b_n)$. The BDD size of a Boolean function is the smallest number of nodes of BDD's representing the function.

An ordered BDD (OBDD) [2] is a BDD in which the order of the occurrence of variables are determined by a total ordering of the variables. If a variable x_i precedes a variable x_j in the total ordering, then x_i appears before x_j on every path that contains both x_i and x_j.

For every perfectly nonlinear Boolean function, its outputs change independently of each other if any change of inputs occurs. This independence induces a conjecture that, for every perfectly nonlinear Boolean function, there exist no variable ordering such that all of the output functions can be represented by OBDD's of small size. We make some approaches to this conjecture.

We consider $f = (f_1, \ldots, f_k) \in \mathrm{P}_{2k,k}$ such that, for every i such that $1 \leqq i \leqq k$,

$$f_i(x, y) = (P_i x^{\mathrm{T}}) \cdot y \oplus g_i(x),$$

where P_i is a $k \times k$ $\{0,1\}$-matrix and $g_i \in \mathrm{B}_k$. For every $(c_1, \ldots, c_k) \in \mathrm{V}_k$, $c_1 P_1 \oplus \cdots \oplus c_k P_k$ is non-singular. Let $\mathrm{P}^M_{2k,k}$ be the set of such functions.

Let A be a matrix. Let $rank(A)$ denote the rank of A and let $A[i_1, \ldots, i_a][j_1, \ldots, j_b]$ denote an $a \times b$ matrix whose (u,v)-element is (i_u, j_v)-element of A, where $i_p \neq i_q$ and $j_s \neq j_t$ for every p, q and s, t such that $1 \leqq p < q \leqq a$ and $1 \leqq s < t \leqq b$, respectively.

The following theorem gives a relationship between the OBDD size of a perfectly nonlinear Boolean function in $\mathrm{P}^M_{2k,k}$ and a combinatorial problem.

Theorem 8: Let $f = (f_1, \ldots, f_k) \in \mathrm{P}^M_{2k,k}$ and, for each i such that $1 \leqq i \leqq k$,

$$f_i(x, y) = (P_i x^{\mathrm{T}}) \cdot y \oplus g_i(x),$$

where P_i is a $k \times k$ $\{0,1\}$-matrix and $g_i \in \mathrm{B}_k$. Let

$$r(f) \stackrel{\mathrm{def}}{=} \min_{\substack{1 \leqq i_1, \ldots, i_{\lceil k/2 \rceil} \leqq k \\ 1 \leqq j_1, \ldots, j_{\lceil k/2 \rceil} \leqq k}} \max_{1 \leqq l \leqq k} rank(P_l[i_1, \ldots, i_{\lceil k/2 \rceil}][j_1, \ldots, j_{\lceil k/2 \rceil}]).$$

Then, for any variable ordering, there exists some i such that $1 \leqq i \leqq k$ and the OBDD size of f_i is $\Omega(2^{r(f)})$.

Proof: For each i such that $1 \leqq i \leqq k$, $f_i(x, y) = y P_i x^{\mathrm{T}} \oplus g_i(x)$.

In a variable ordering of (x, y), suppose that x_{u_p} precedes x_{u_q} and y_{v_p} precedes y_{v_q} if $1 \leqq p < q \leqq k$.

Suppose that $x_{u_{\lceil k/2 \rceil}}$ precedes $y_{v_{\lceil k/2 \rceil}}$. Then, for some d such that $1 \leqq d \leqq k$, $rank(P'_d) \geqq r(f)$, where $P'_d = P_d[v_{\lfloor k/2 \rfloor + 1}, \ldots, v_k][u_1, \ldots, u_{\lceil k/2 \rceil}]$. Thus,

$$\left|\{a \mid a = P'_d b^{\mathrm{T}}, b \in \{0,1\}^{\lceil k/2 \rceil}\}\right| \geqq 2^{r(f)}.$$

Let $x' = (x_{u_1}, \ldots, x_{u_{\lceil k/2 \rceil}})$ and $y' = (y_{v_{\lfloor k/2 \rfloor + 1}}, \ldots, y_{v_k})$. Let $g' = g|_{x_{u_{\lceil k/2 \rceil + 1}} = \cdots = x_{u_k} = 0}$. Then,

$$f'_d = y' P'_d(x')^{\mathrm{T}} \oplus g'(x')$$

is a sub-function of f_d constructed by substituting 0's for $x_{u_{\lceil k/2 \rceil + 1}}, \ldots, x_{u_k}$ and $y_{v_1}, \ldots, y_{v_{\lfloor k/2 \rfloor}}$. For the variable ordering $(x_{u_1}, \ldots, x_{u_{\lceil k/2 \rceil}}, y_{v_{\lceil k/2 \rceil + 1}}, \ldots, y_{v_k})$, the OBDD size of f'_d is $\Omega(2^{r(f)})$. Thus, the OBDD size of f_d is $\Omega(2^{r(f)})$ if $x_{u_{\lceil k/2 \rceil}}$ precedes $y_{v_{\lceil k/2 \rceil}}$.

For the case where $y_{v_{\lceil k/2 \rceil}}$ precedes $x_{u_{\lceil k/2 \rceil}}$, the theorem can be proved in the same way. □

We conjecture that $r(f) \geq \lfloor k/4 \rfloor$ for every $f \in \mathrm{P}^M_{2k,k}$.

A method of construction for perfectly nonlinear Boolean functions in $\mathrm{P}^M_{2k,k}$ was proposed by Nyberg [8]. Let R be a $k \times k$ $\{0,1\}$-matrix which expresses a state transition function of a linear feedback shift register of length k with a primitive feedback polynomial. Let $f = (f_1, \ldots, f_k) \in \mathrm{B}_{2k,k}$ such that, for each i such that $1 \leqq i \leqq k$,

$$f_i(x, y) = (R^{i-1} x^{\mathrm{T}}) \cdot y \oplus g_i(x),$$

where $g_i \in \mathrm{B}_k$. Then, $f \in \mathrm{P}^M_{2k,k}$.

The following theorem can be immediately derived from the proof of the OBDD size of integer multiplication in Ref. [2].

Theorem 9: Suppose that $f = (f_1, \ldots, f_k) \in \mathrm{P}^M_{2k,k}$ is constructed with the method of Nyberg. Then, for any variable ordering, there exists some i such that $1 \leqq i \leqq k$ and the OBDD size of f_i is $\Omega(2^{k/16})$. □

7. Conclusion

We have discussed the complexity of Boolean functions satisfying the PC on several computation models. We have considered the unateness, the inversion complexity, the formula size, the area-time-square VLSI complexity and the OBDD size. The proofs of the theorems on relationships between the unateness and the PC show that the Walsh transform is a powerful technique to analyze properties of Boolean functions.

An open question is to obtain a good lower bound on $r(f)$ in Theorem 8. It is also left as a future work to study the inversion complexity, the area-time-square VLSI complexity, the OBDD size of Boolean functions in $\mathrm{PC}_{n,m}(k)$. It is also interesting to investigate the complexity of Boolean functions satisfying the other nonlinearity criteria.

Acknowledgement

The authors would like to thank referees for their valuable comments.

References

[1] Biham, E. and Shamir, A., *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[2] Bryant, R., "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol.C-35, no.8, pp.677–691, 1986.

[3] Hirose, S. and Ikeda, K., "Propagation characteristics of Boolean functions and their balancedness," to appear in *IEICE Trans. Fundamentals*, vol.E78-A, no.1, pp.11–18, 1995.

[4] Kohavi, Z., *Switching and finite automata theory*, 2nd ed., Tata McGraw-Hill, 1978.

[5] Krapchenko, V., "A method of determining lower bounds for the complexity of π schemes," *Math. Notes Acad. Sci. USSR*, vol.11, pp.474–479, 1971.

[6] Markov, A.A., "On the inversion complexity of a system of functions," *J. ACM*, vol.5, no.4, pp.331–334, 1958.

[7] Meier, W. and Staffelbach, O., "Nonlinearity criteria for cryptographic functions," *Proc. EUROCRYPT'89, LNCS*, no.434, pp.549–562, 1990.

[8] Nyberg, K., "Perfect nonlinear S-boxes," *Proc. EUROCRYPT'91, LNCS*, no.547, pp.378–386, 1991.

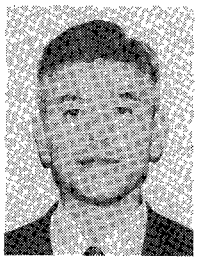
[9] Preneel, B., Leekwijk, W.V., Linden, L.V., Govaerts, R. and Vandewalle, J., "Propagation characteristics of Boolean functions," *Proc. EUROCRYPT'90, LNCS*, no.473, pp.161–173, 1991.

[10] Rueppel, R.A., "Stream ciphers," in *Contemporary cryptology: The science of information integrity*, ed. G. Simmons, pp.65–134, IEEE Press, 1991.

[11] Ullman, J.D., *Computational aspects of VLSI*, Computer Science Press, 1984.

[12] Wegener, I., *The complexity of Boolean functions*, John Wiley & Sons, 1987.

Shouichi Hirose was born in Kyoto, Japan, on December 30, 1965. He received the B.E. and M.E. degrees in information science from Kyoto University, Kyoto, Japan, in 1988 and 1990, respectively. He is an Instructor at the Department of Information Science, Faculty of Engineering, Kyoto University. His current interests include cryptography, computer security, computational complexity and Boolean functions. He is a member of IPSJ.



Katsuo Ikeda was born in 1937 in Shiga, Japan. He received the B.E. and M.E. degrees in electronic engineering and the D.E. degree in information science from Kyoto University in 1960, 1962 and 1978, respectively. He is currently a Professor of the Department of Information Science, Kyoto University. From 1965 to 1971 he was a Research Associate and from 1971 to 1978 an Associate Professor of the Faculty of Engineering, Kyoto University. From 1978 to 1988 he was a Professor of the Science Information Center and the Department of Information Science, University of Tsukuba. He was a guest researcher at the University of Utah and the Massachusetts Institute of Technology in the academic year 1971–1972, and at the Swiss Federal Institute of Technology (ETH) for two months in 1983. His primary research interests include construction of intelligent information media environment. He is the author of Structure of a Computer Utility (in Japanese; Shokodo, 1974) and Data communication (in Japanese; Shokodo, 1993), and the translator of System Programming (J.J. Donovan, 1974) and Operating System (J.J. Donovan, 1976). He is a member of IPSJ, IECEJ, IEEE, ACM and the editorial board of Information Processing Letters, Elsevier. He is also working as a leader of standardization activities related to the Japanese national body of the SC18 of ISO/IEC JTC1 (Document processing and related communication).