<table>
<thead>
<tr>
<th>Title</th>
<th>On Patterns of Threshold Circuits computing the PARITY function (Foundations of Theoretical Computer Science: For New Computational View)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Uchizawa, Kei; Takimoto, Eiji</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (2008), 1599: 91-96</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2008-05</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/81790">http://hdl.handle.net/2433/81790</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

Kyoto University
On Patterns of Threshold Circuits computing the PARITY function
（しきい値回路のパターン数について）

東北大学大学院・情報科学研究科 内沢 啓 (Kei Uchizawa)
東北大学大学院・情報科学研究科 瀬本 英二 (Eiji Takimoto)
Graduate School of Information Sciences
Tohoku University

Abstract
In the paper, we prove that any threshold circuit computing the PARITY function of $n$ variables has at least $n+1$ patterns, where a pattern is defined to be the sequence of gate states that arise during computation for an assignment.

1 Introduction
Circuits consisting of threshold gates are called threshold circuits, and have been extensively studied for a few decades [1, 2, 3, 4, 5, 6]. Recently, we have introduced a new notion of patterns into threshold circuits [7]. For an input assignment $x$, the pattern for $x$ is defined to be the sequence of gate states that arise during computation for the assignment. In Ref. [7], we show that the number of patterns that arise in a threshold circuit is closely related to the size of the circuit, where the size of a circuit is the number of gates contained in the circuit. In particular, in Ref. [7], by estimating the number of patterns that arise in threshold circuits, we prove that threshold circuits with some restrictions need an exponential number of gates to compute a particular Boolean function, i.e., the Inner-Product function. However, our argument fails to derive non-trivial lower bounds for many simpler functions, including the PARITY function.

In the paper, we consider threshold circuits computing the PARITY function, and derive a lower bound on the number of patterns of threshold circuits. More precisely, we prove that threshold circuits computing the PARITY function of $n$ variables must have at least $n+1$ patterns.

Moreover, from the lower bound we derive a tight lower bound on the size of threshold circuits computing the PARITY function. Note that one can find the same lower bound derived by a different proof method in [6]. However, we derive the lower bound, since it is a good example to give insight into the relation between patterns and the size of circuits.

2 Definitions
In the section, we first give definitions and several terms needed to describe our results.
For every input $x = (z_1, z_2, \ldots, z_m) \in \{0, 1\}^m$, a threshold gate $g$ (with weights $w_1, w_2, \ldots, w_m$ and a threshold $t$) computes a linear threshold function given by

$$g(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} w_i z_i \geq t; \\ 0 & \text{otherwise.} \end{cases}$$
A threshold circuit $C$ with $n$ input variables is represented by a directed acyclic graph; the graph has exactly $n$ nodes of in-degree 0, each associated with an input variable and called an input node; each of the other nodes represents a threshold gate. For an assignment $x \in \{0, 1\}^n$ to the $n$ input variables, the output of all gates in $C$ are computed in topological order of the nodes in the directed acyclic graph. For a gate $g$ in $C$, we denote by $g[x]$ the output of $g$ for an input $x$ to circuit $C$ (although the actual input to gate $g$ will in general consist of some variables from $x$ and, in addition, or even exclusively, the outputs of some other gates in $C$).

The size of a threshold circuit $C$ is the number of gates in $C$. Since we consider only a threshold circuit that computes a Boolean function, one may assume without loss of generality that the circuit has exactly one gate of out-degree 0, called the top gate. We say that a threshold circuit $C$ computes a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ if the output of the top gate for $x$ equals to $f(x)$ for every input $x \in \{0, 1\}^n$.

Assume that a threshold circuit $C$ consists of $m$ threshold gates, $g_1, g_2, \ldots, g_m$ for some $m \geq 1$. For an input assignment $x$ for $C$, we call the $m$-tuple of the gate outputs,

$$(g_1[x], g_2[x], \ldots, g_m[x]),$$

the pattern of $C$ for $x$, and we say that the pattern arises for $x$ in $C$. Let $\text{PAT}(C)$ be the set of all patterns of $C$. That is,

$$\text{PAT}(C) = \{(g_1[x], g_2[x], \ldots, g_m[x]) \mid x \in \{0, 1\}^n\}.$$ 

For every input assignment

$$x = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n,$$

the PARITY function of $n$ variables is defined to be

$$\text{PARITY}(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i \text{ is odd;} \\ 0 & \text{if } \sum_{i=1}^{n} x_i \text{ is even.} \end{cases}$$

### 3 Main Result

In the section, we prove the theorem below.

**Theorem 1.** Any threshold circuit $C$ computing the PARITY function of $n$ variables has at least $n + 1$ patterns. That is, 

$$|\text{PAT}(C)| \geq n + 1.$$ 

To give a proof of the theorem, we need the following two lemmas, Lemma 1 and Lemma 2. We prove Lemma 1 in the section, while we prove Lemma 2 in the next section.

**Lemma 1.** Any threshold circuit $C$ computing the PARITY function of two variables has at least three patterns. That is, 

$$|\text{PAT}(C)| \geq 3.$$
Proof. Let $C$ be a threshold circuit computing the PARITY function of two variables, $x_1$ and $x_2$. Assume that the circuit $C$ contains threshold gates $g_1, g_2, \ldots, g_m$ indexed in the topological order, where $m$ is the size of $C$. That is, for every index $i$, the gate $g_i$ receives inputs only from $g_1, g_2, \ldots, g_{i-1}$ as well as from input variables $x_1$ and $x_2$. That is, for each index $i$, $1 \leq i \leq m$, we have

$$g_i(x_1, x_2) = g_i(x_1, x_2, g_1[x_1, x_2], \ldots, g_{i-1}[x_1, x_2])$$

We prove the lemma by contradiction. Assume that the circuit has just two patterns. Clearly, one of the two patterns must be for the case where $x_1 + x_2$ is even, and the other must be for the case where $x_1 + x_2$ is odd. Therefore, one of the two patterns arises for inputs $(x_1, x_2) = (0, 0), (1, 1)$, and the other does for $(x_1, x_2) = (0, 1), (1, 0)$. Then, let $i$ be the least index such that $g_i[0, 0] \neq g_i[0, 1]$. Note that $g_i[1, 1] = g_i[0, 0]$ and $g_i[0, 1] = g_i[1, 0]$. Since the outputs of the gates $g_j$ for $j < i$ are considered to be a constant, we can consider that the gate $g_i$ computes a threshold function of $x_1$ and $x_2$. More precisely, the function $f$ defined as

$$f(x_1, x_2) = g_i(x_1, x_2) = g_i(x_1, x_2, g_1[x_1, x_2], \ldots, g_{i-1}[x_1, x_2])$$

is a threshold function. Since $g_i[0, 0] = g_i[1, 1] \neq g_i[0, 1] = g_i[1, 0]$, we have

$$f(0, 0) = f(1, 1) \neq f(0, 1) = f(1, 0),$$

which implies that $f$ computes the PARITY function of two variables. This contradicts the fact that the PARITY function is not a threshold function. \hfill \Box

Lemma 2. Let $C$ be any threshold circuit computing the PARITY function of $n$ variables. Let $C_0$ be the threshold circuit obtained by replacing the input node $x_n$ of $C$ with constant input $0$. Then

$$|\text{PAT}(C_0)| \leq |\text{PAT}(C)| - 1.$$

Using the two lemmas, we prove the theorem in the following.

Proof (of the theorem) We will prove by induction on $n$ that

$$|\text{PAT}(C)| \geq n + 1$$

for any threshold circuit $C$ computing the PARITY function of $n$ variables.

Obviously, Lemma 1 confirms the basis, i.e., the case where $n = 2$, of the induction.

Below we show the induction step. Let $C$ be any threshold circuit computing the PARITY function of $n$ variables. Let $C_0$ be the circuit obtained by replacing the input node $x_n$ of $C$ with constant input 0. Since $C_0$ computes the PARITY function of $n - 1$ variables, the induction hypothesis implies that

$$|\text{PAT}(C_0)| \geq n.$$ 

(2)

By Lemma 2 and Eq. (2), we have

$$|\text{PAT}(C)| \geq |\text{PAT}(C_0)| + 1 \geq n + 1,$$
which confirms Eq. (1).

By Theorem 1, we can easily derive as below that any threshold circuit computing the PARITY function of $n$ variables needs $\log(n+1)$ gates. Althogh one can find the same lower bound derived by a different proof method in [6], we put it as corolary to give a insight into the relation between patterns and the size of circuits.

**Corollary 1.** Every threshold circuit computing the PARITY function of $n$ variables has at least $\log(n+1)$ gates.

**Proof.** By Theorem 1, any threshold circuit computing the PARITY function of $n$ variables needs $n+1$ patterns. To realize $n+1$ patterns, the circuit needs $\log(n+1)$ gates. $\square$

This lower bound is tight, since the PARITY function is computable by a threshold circuit of size $O(\log n)$ [6].

## 4 Proof of Lemma 2

In the rest of the paper, we prove Lemma 2.

Let $C$ be a threshold circuit computing the PARITY function of $n$ variables, and let $m$ be the size of $C$. Assume that the circuit $C$ contains threshold gates $g_1, g_2, \ldots, g_m$ indexed in topological order. Let $C_0$ be a circuit obtained by replacing the input node $x_n$ of $C$ with constant input 0. We prove that

$$|\text{PAT}(C_0)| \leq |\text{PAT}(C)| - 1. \quad (3)$$

We give a proof by contradiction. Assume that

$$|\text{PAT}(C_0)| = |\text{PAT}(C)|, \quad (4)$$

that is,

$$\text{PAT}(C_0) = \text{PAT}(C). \quad (5)$$

Similarly to the definition of $C_0$, let $C_1$ be a circuit obtained by replacing the input node $x_n$ of $C$ with constant input 1. By Eq. (5), we have

$$\text{PAT}(C_1) \subseteq \text{PAT}(C_0) = \text{PAT}(C). \quad (6)$$

Let

$$X_0 = \{(x_1, x_2, \ldots, x_n) \in \{0, 1\}^n \mid x_n = 0\}.$$ 

Now we consider the following sequence of patterns, $p_1, p_2, \ldots, p_s$ of length $s = |\text{PAT}(C)| + 1$. The sequence starts with any pattern $p_1 \in \text{PAT}(C_1)$. Let $x_1 \in X_0$ be an input for which the pattern $p_1$ arises in $C$. Equation (6) guarantees that there must exists such input $x_1$. For any positive integer $j$, $1 \leq j \leq s$, the $j+1$-th pattern $p_{j+1} \in \text{PAT}(C)$ of the sequence is the one that arises for the input $x_j'$ in which all bits but the $n$-th bit are the same as those in $x_j$. Let $x_{j+1} \in X_0$ be an input for which the pattern $p_{j+1}$ arises in $C$. Equation. (6) also guarantees that there must exist such input $x_{j+1}$. 


Since the length $s$ of the sequence is larger than $|PAT(C)|$, there must exist a pattern that appears twice in the sequence. Assume without loss of generality that the pattern $p_1$ appears twice. That is,

$$p_1 = p_k$$

for some $k \geq 2$.

Using the sequence, we next define a sequence of gates. For each integer $j$, $1 \leq j \leq k$, we choose the $j$-th gate of the sequence as follows: the $j$-th gate has the least index among the gates $g$ such that $g[x_j] \neq g[x_{j+1}]$. Let $I_j$ be the index of the $j$-th gate of the sequence. Furthermore, let

$$t = \arg \min_{1 \leq t \leq k} I_t.$$  

Now we look at the outputs of the gate $g_t$ for inputs $x_1, x_2, \ldots, x_k$. Note that $g_t[x_j]$ is the $I_t$-th bit of the $t$-th pattern $p_t$. Assume without loss of generality that

$$g_t[x_1] = 0.$$  

By the definition of $g_t$, we have

$$g_t[x_j] = 0$$

for every index $j$, $1 \leq j \leq t$, and

$$g_t'[x_t'] = g_t'[x_{t+1}] = 1$$

This implies that the gate $g_t$ has a positive weight for the input variable $x_t$. Equation (8) together with the fact implies that

$$g_t[x_j] = 1$$

for every index $j \geq t + 1$. Therefore, we have

$$g_t[x_k] = 1.$$  

Equations (9) and (10) contradict Eq. (7).

References


