

Size and Energy of Threshold Circuits Computing Mod Functions

Kei Uchizawa* Eiji Takimoto† Takao Nishizeki‡

*‡{uchizawa, nishi}@ecei.tohoku.ac.jp †eiji@i.kyushu-u.ac.jp

Abstract

Let C be a threshold logic circuit computing a Boolean function $\text{MOD}_m : \{0, 1\}^n \rightarrow \{0, 1\}$, where $n \geq 1$ and $m \geq 2$. Then C outputs “0” if the number of “1”s in an input $\mathbf{x} \in \{0, 1\}^n$ to C is a multiple of m and, otherwise, C outputs “1.” The function MOD_2 is the so-called PARITY function, and MOD_{n+1} is the OR function. Let s be the size of the circuit C , that is, C consists of s threshold gates, and let e be the energy complexity of C , that is, at most e gates in C output “1” for any input $\mathbf{x} \in \{0, 1\}^n$. In the paper, we prove that a very simple inequality $n/(m-1) \leq s^e$ holds for every circuit C computing MOD_m . The inequality implies that there is a tradeoff between the size s and energy complexity e of a threshold circuit C computing MOD_m , and yields a lower bound $e = \Omega((\log n - \log m)/\log \log n)$ on e if $s = O(\text{polylog}(n))$. We actually obtain a general result on the so-called generalized mod function, from which the result on the ordinary mod function MOD_m immediately follows.

1 Introduction

A circuit of threshold gates is a theoretical model of a neural circuit in the brain, and is well studied through decades [9, 10, 12, 13]. An input-output characteristic of a biological neuron is roughly represented by a threshold gate, but the mechanism of energy consumption of a neuron is quite different from an electrical circuit: a neural “firing” consumes substantially more energy than a “non-firing” [7, 8], while a gate in an electrical circuit consumes almost the same amount of energy in either case of outputting “1” and outputting “0” [1, 6]. A biological study reports that, due to the asymmetry of the energy consumption, the fraction of neurons firing concurrently is possibly fewer than 1% [7]. Based on the biological fact above, the energy complexity e of a threshold circuit C is defined as the maximum number of threshold gates outputting “1” over all inputs to C [15]. We then confront the following natural question from the point of computational complexity: what Boolean functions can or cannot be computed by reasonably small threshold circuits with small energy complexity? It has been shown that the energy complexity strongly influences the computational power of threshold circuits [15, 16]. In

*‡ Graduate School of Information Sciences, Tohoku University, Sendai, 980-8579, Japan.

† Faculty/Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, 819-0396, Japan.

particular, if a Boolean function f has high communication complexity, there exists a tradeoff among the following three complexities: size (that is, the number of gates) s , depth d , and energy complexity e of threshold circuits computing f [16]. However, the mod function $\text{MOD}_m : \{0, 1\}^n \rightarrow \{0, 1\}$ has low communication complexity, and hence the result in [16] does not yield any interesting tradeoff for MOD_m , where n and m are positive integers, and $\text{MOD}_m(\mathbf{x})$ is 0 if the number of “1”s in an input $\mathbf{x} \in \{0, 1\}^n$ is a multiple of m and, otherwise, $\text{MOD}_m(\mathbf{x})$ is 1. MOD_m is the PARITY function if $m = 2$, and is the OR function if $m = n + 1$.

In the paper, we deal with a fairly large class of Boolean functions, called the generalized mod function [2, 4], and show that there exists a tradeoff between the size s and energy complexity e of threshold circuits C computing the generalized mod function. The result immediately yields a very simple tradeoff for the ordinary mod function MOD_m . More precisely, we prove that $n/(m-1) \leq s^e$, that is, $\log(n/(m-1)) \leq e \log s$, for every circuit C computing MOD_m . Both n and m , and hence $n/(m-1)$, do not depend on the design of C , while s^e is monotonically increasing with respect to s and e . Therefore, s and e cannot be simultaneously small. That is, if s is small, then e must be large, and if e is small, then s must be large. The tradeoff $n/(m-1) \leq s^e$ immediately implies a lower bound on the size s expressed by n, m and e : $(n/(m-1))^{1/e} \leq s$. If $s = O(\text{polylog}(n))$, then the tradeoff also implies a lower bound on e : $e = \Omega((\log n - \log m)/\log \log n)$. The lower bound on e is tight up to a constant factor.

It is well known that there exists a tradeoff between the size s and depth d of a threshold circuit computing the PARITY function. Siu *et al.* proved that $n \leq (s/d)^{d+\epsilon}$ for any fixed $\epsilon > 0$ if the weights of the threshold gates are integers and their absolute values are sufficiently small [14]. Impagliazzo *et al.* proved that $n/2 \leq s^{2(d-1)}$ even if the absolute values of weights are arbitrarily large [5]. Our tradeoff between s and e holds even if the absolute values of weights are arbitrarily large. It should be noted that the inequality $d \leq e$ does not necessarily hold.

In Section 2, we define some terms on the generalized mod functions and threshold circuits. In Section 3, we prove our main theorem for the generalized mod functions, from which the tradeoff $n/(m-1) \leq s^e$ for MOD_m immediately follows. In Section 4, we conclude with some remarks.

2 Preliminaries

In this section, we first define some terms on mod functions and threshold circuits, and then present some examples of circuits for the PARITY function.

For $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$, we denote by $[\mathbf{x}]_m$ the hamming weight of \mathbf{x} modulo m . Thus

$$[\mathbf{x}]_m = \sum_{i=1}^n x_i \pmod{m}.$$

Let $M = \{0, 1, \dots, m-1\}$, then $m = |M|$. For a set $A \subseteq M$, the *generalized mod function* $\text{MOD}_m^A : \{0, 1\}^n \rightarrow \{0, 1\}$ is defined as follows [2, 4]:

$$\text{MOD}_m^A(\mathbf{x}) = \begin{cases} 0 & \text{if } [\mathbf{x}]_m \in A; \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

Parameters Designs	s	e	d	Notes
Small d	$n + 1$	$n + 1$	2	Fig. 2(a)
Small e	$n + 1$	2	$n + 1$	Fig. 2(b)
Moderate s, e and d	$\log n$	$\log n$	$\log n$	Ref. [13]
Moderate s, e and fairly small d	$\text{polylog}(n)$	$\text{polylog}(n)$	$\log n / \log \log n$	Ref. [11]
Moderate s, d and fairly small e	$\text{polylog}(n)$	$\log n / \log \log n$	$\text{polylog}(n)$	Sect. 3.1
Small e and d	$2^{n-1} + 1$	1	2	Truth table

Table 1: Various designs of circuits computing the PARITY function of n variables.

The *size* s of a threshold circuit C is the number of threshold gates in C . Figure 1(a) depicts a threshold circuit with $n = 3$ and $s = 5$, while Fig. 1(b) depicts a circuit with $n = 2$ and $s = 5$. (Impagliazzo *et al.* define the “size” of C to be the number of wires in C , and obtained a tradeoff between the “size” and the depth [5].)

Let C be a threshold circuit of size s , let g_1, g_2, \dots, g_s be the gates in C , and let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ be an input to C . Then the input \mathbf{z}_i to a gate g_i , $1 \leq i \leq s$, either consists of the inputs x_1, x_2, \dots, x_n to C and the outputs of the gates other than g_i or consists of some of them. However, we denote the output $g_i(\mathbf{z}_i)$ of g_i for \mathbf{z}_i by $g_i[\mathbf{x}]$, because \mathbf{x} decides $g_i(\mathbf{z}_i)$. Thus $g_i[\mathbf{x}] = g_i(\mathbf{z}_i)$. Let g_s be one of the gates of out-degree 0, and we regard the output $g_s[\mathbf{x}]$ of g_s as the *output* $C(\mathbf{x})$ of C . Thus, $C(\mathbf{x}) = g_s[\mathbf{x}]$ for every input $\mathbf{x} \in \{0, 1\}^n$. The gate g_s is called the *output gate* of C .

A threshold circuit C *computes* a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if $C(\mathbf{x}) = f(\mathbf{x})$ for every input $\mathbf{x} \in \{0, 1\}^n$.

The *depth* d of a circuit C is the number of gates in the longest path from an input node to the output gate g_s .

We define the energy complexity e of a threshold circuit C as

$$e = \max_{\mathbf{x} \in \{0, 1\}^n} \sum_{i=1}^s g_i[\mathbf{x}].$$

Thus, the energy complexity e is the maximum number of gates outputting “1” over all inputs $\mathbf{x} \in \{0, 1\}^n$. Clearly $0 \leq e \leq s$. We may assume without loss of generality that $e \geq 1$.

As summarized in the Table 1, there are various designs of threshold circuits computing the PARITY function MOD_2 . Figure 2 illustrates two of them, for which $n = 4$ and $s = n + 1 = 5$. For the circuit in Fig. 2(a) $d = 2$ and $e = n = 4$. On the other hand, for the circuit in Fig 2(b) $d = n + 1 = 5$ and $e = 2$; if the number i of “1”s in an input is odd, then only the two gates g_i and g_s output “1”; and otherwise only g_i outputs “1.”

Throughout the paper, we denote by n the number of input variables to a threshold circuit C , by s the size of C , and by e the energy complexity of C . We may assume without loss of generality that $n, s, e \geq 1$.

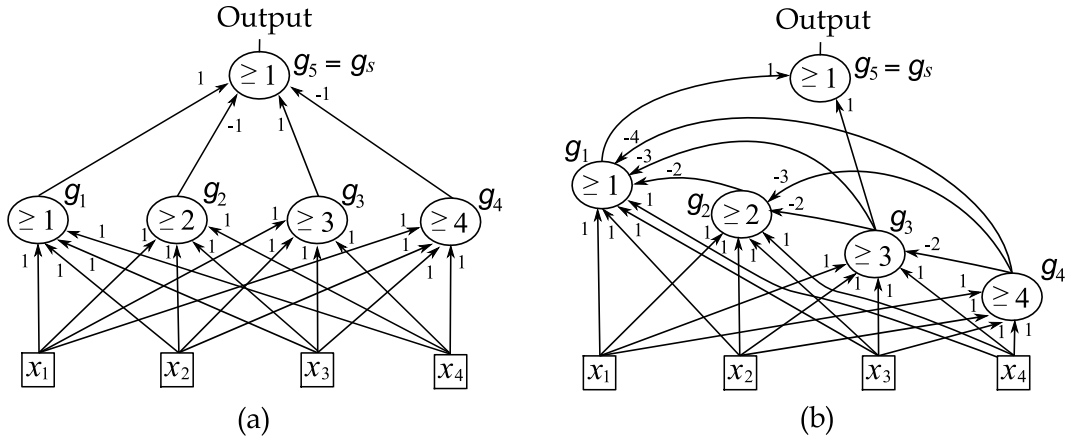


Figure 2: Threshold circuits computing the PARITY function of 4 variables;
(a) $s = 5, d = 2, e = 5$; and (b) $s = 5, d = 5, e = 2$.

3 Size-Energy Tradeoff

In Section 3.1, we present, as Theorem 1, our main result on the size-energy tradeoff for circuits computing the generalized mod function MOD_m^A . The theorem immediately yields a tradeoff for circuits computing the ordinary mod function MOD_m . In Section 3.2, we present four lemmas, and using them we prove Theorem 1. In Section 3.3, we present a tradeoff better than that in Theorem 1 if $e \geq 5$.

3.1 Main Theorem and Corollaries

Our main result is the following theorem:

Theorem 1. *Let C be a threshold circuit computing the generalized mod function MOD_m^A of n variables, and let $a = \min\{|A|, |M - A|\}$. Then the size s and energy complexity e of C satisfy*

$$\frac{n + 1 - a}{m - a} \leq s^e. \quad (5)$$

The ordinary mod function MOD_m is MOD_m^A for the case where $A = \{0\}$ and hence $a = 1$. The PARITY function is MOD_m for the case $m = 2$. We thus have the following corollary.

Corollary 1.

(a) *If a threshold circuit C computes the ordinary mod function MOD_m , then*

$$\frac{n}{m - 1} \leq s^e. \quad (6)$$

(b) *If a threshold circuit C computes the PARITY function, then*

$$n \leq s^e \quad (7)$$

and hence

$$\log n \leq e \log s. \quad (8)$$

If n , m and a are fixed, then the left side $(n + 1 - a)/(m - a)$ of Eq. (5) is a constant and does not depend on the design of C . On the other hand, s and e depend on the design of C , and the right side s^e is monotonically increasing with regards to s and e . Thus Eq. (5) implies that there exists a tradeoff between e and s . That is, e and s cannot be simultaneously small, e must be large if s is small, and e must be large if s is small.

One can know that the lower bound $(n + 1 - a)/(m - a)$ on s^e in Eq. (5) cannot be improved much, as follows. For the case where $m = n + 1$ and $A = \{0\}$, MOD_m^A is the OR function, and can be computed by a circuit C with $s = e = 1$, and hence Eq. (5) holds in equality for the circuit C . Thus, for any $\epsilon > 0$, the equation

$$(1 + \epsilon) \left(\frac{n + 1 - a}{m - a} \right) \leq s^e$$

does not hold. For the case where $m = 2$ and $A = \{0\}$, MOD_m^A is the PARITY function MOD_2 , which can be computed by a circuit C such that $s = n + 1$ and $e = 2$ as illustrated in Fig. 2(b). In this case, the right side s^e of Eq. (5) is $(n + 1)^2$ for the circuit C , while the left side is n . Therefore, for any $\epsilon > 0$, the equation

$$\left(\frac{n + 1 - a}{m - a} \right)^{2+\epsilon} \leq s^e$$

does not hold if n is sufficiently large.

Equation (5) immediately implies

$$\left(\frac{n + 1 - a}{m - a} \right)^{1/e} \leq s,$$

which is a lower bound on s expressed in terms of n, m, a and e . One can easily know from the bound that $s = \Omega(\sqrt{n})$ if $e \leq 2$ and $m = O(1)$.

From Theorem 1, one can immediately obtain a lower bound on e expressed in terms of n and m as follows.

Corollary 2. *Let C be a threshold circuit computing MOD_m . If $s = O(\text{polylog}(n))$, then*

$$e = \Omega \left(\frac{\log n - \log m}{\log \log n} \right).$$

Corollary 2 implies that if $m = o(n)$ then MOD_m cannot be computed by any threshold circuit C such that $s = O(\text{polylog}(n))$ and $e = o(\log n / \log \log n)$. Similarly to the corollary above, Sung and Nishino [11] prove that $d = \Theta(\log n / \log \log n)$ if a threshold circuit C with depth d computes the PARITY function and $s = O(\text{polylog}(n))$. Slightly modifying a circuit given in [11], one can construct a threshold circuit of size $s = O(\text{polylog}(n))$ and energy $e = O(\log n / \log \log n)$ that computes the PARITY function of n variables. Thus, the lower bound on e in Corollary 2 is best possible within a constant factor for the case where $m = 2$.

3.2 Proof of Theorem 1

In the section, we first present four lemmas and then, using them, we prove Theorem 1.

Let a threshold circuit C consist of gates g_1, g_2, \dots, g_s , and let g_s be the output gate of C : $g_s[\mathbf{x}] = C(\mathbf{x})$ for every $\mathbf{x} \in \{0, 1\}^n$. For an input $\mathbf{x} \in \{0, 1\}^n$, we define a *pattern* $\mathbf{p}_C(\mathbf{x}) \in \{0, 1\}^s$ of C for \mathbf{x} as

$$\mathbf{p}_C(\mathbf{x}) = (g_1[\mathbf{x}], g_2[\mathbf{x}], \dots, g_s[\mathbf{x}]).$$

We often denote $\mathbf{p}_C(\mathbf{x})$ simply by $\mathbf{p}(\mathbf{x})$. We denote by $P(C)$ the set of all patterns that arise in C :

$$P(C) = \{\mathbf{p}_C(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n\}.$$

The number $|P(C)|$ of patterns is closely related to the size s and the energy complexity e . One can easily prove the following lemma, which gives an upper bound on $|P(C)|$ in terms of s and e .

Lemma 1. *For an arbitrary threshold circuit C ,*

$$|P(C)| \leq s^e + 1. \quad (9)$$

Proof. If $s = 1$, then $|P(C)| \leq 2$, $s^e + 1 = 2$ and hence Eq. (9) holds. We may thus assume that $s \geq 2$. Since the energy complexity of C is e , at most e of the s gates output “1” for any input \mathbf{x} . Therefore, we have

$$|P(C)| \leq \sum_{i=0}^e \binom{s}{i} \quad (10)$$

$$\begin{aligned} &\leq 1 + s + \frac{1}{2}(s^2 + s^3 + \dots + s^e) \\ &\leq 1 + s + \frac{s^2(s^{e-1} - 1)}{2(s - 1)}. \end{aligned} \quad (11)$$

From Eq. (11) and $s \leq 2(s - 1)$, we obtain

$$\begin{aligned} |P(C)| &\leq 1 + s + s(s^{e-1} - 1) \\ &\leq 1 + s^e. \end{aligned}$$

□

For every input $\mathbf{x} \in \{0, 1\}^n$, we define an *extended pattern* $\mathbf{q}_C(\mathbf{x}) \in \{0, 1\}^s \times M$ of a threshold circuit C for \mathbf{x} as follows:

$$\mathbf{q}_C(\mathbf{x}) = (\mathbf{p}_C(\mathbf{x}), [\mathbf{x}]_m),$$

where $M = \{0, 1, \dots, m - 1\}$. We often denote $\mathbf{q}_C(\mathbf{x})$ simply by $\mathbf{q}(\mathbf{x})$. We denote by $Q(C)$ the set of all extended patterns that arise in C :

$$Q(C) = \{\mathbf{q}_C(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n\}. \quad (12)$$

Since $|M| = m$, we have

$$|Q(C)| \leq |P(C)| \cdot m. \quad (13)$$

For a circuit C computing MOD_m^A , we can obtain an upper bound on $|Q(C)|$, which is expressed in terms of $|P(C)|$, m and a , and is better than Eq. (13).

Lemma 2. *Let C be a threshold circuit computing MOD_m^A , and let $a = |A|$. Then*

$$|Q(C)| \leq (|P(C)| - 1)(m - a) + a. \quad (14)$$

Proof. We give a proof only for the case where $|A| \leq |M - A|$ and hence $a = |A|$, because the proof for the other case where $|A| > |M - A|$ is similar.

The set $P(C)$ can be partitioned into the following two subsets $P_1(C)$ and $P_0(C)$:

$$P_1(C) = \{\mathbf{p}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 1\}$$

and

$$P_0(C) = \{\mathbf{p}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 0\}.$$

Since g_s is the output gate of C , we have $g_s[\mathbf{x}] = 1$ if $C(\mathbf{x}) = 1$, and $g_s[\mathbf{x}] = 0$ if $C(\mathbf{x}) = 0$. Thus $P_1(C) \cap P_0(C) = \emptyset$. Similarly, the set $Q(C)$ can be partitioned into the following two subsets $Q_1(C)$ and $Q_0(C)$:

$$Q_1(C) = \{\mathbf{q}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 1\}$$

and

$$Q_0(C) = \{\mathbf{q}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 0\}.$$

Clearly

$$|P(C)| = |P_1(C)| + |P_0(C)| \quad (15)$$

and

$$|Q(C)| = |Q_1(C)| + |Q_0(C)|. \quad (16)$$

If $C(\mathbf{x}) = \text{MOD}_m^A(\mathbf{x}) = 1$, then $[\mathbf{x}]_m \in M - A$ by Eq. (1). We thus have

$$|Q_1(C)| \leq |P_1(C)| \cdot (m - a). \quad (17)$$

On the other hand, if $C(\mathbf{x}) = 0$ then $[\mathbf{x}]_m \in A$. We thus have

$$|Q_0(C)| \leq |P_0(C)| \cdot a \quad (18)$$

Substituting Eqs. (17) and (18) to Eq. (16), we have

$$|Q(C)| \leq |P_1(C)| \cdot (m - a) + |P_0(C)| \cdot a. \quad (19)$$

Equations (15) and (19) imply that

$$\begin{aligned} |Q(C)| &\leq (|P(C)| - |P_0(C)|) \cdot (m - a) + |P_0(C)| \cdot a \\ &= |P(C)| \cdot (m - a) - |P_0(C)| \cdot (m - 2a). \end{aligned} \quad (20)$$

By Eq. (2) we have $m - 2a \geq 0$. Therefore, the right side of Eq. (20) is non-increasing with respect to $|P_0(C)|$. Since $a \geq 1$ by Eq. (2), we have $A \neq \emptyset$. Since $A \subseteq M = \{0, 1, \dots, m - 1\}$ and $m - 1 \leq n$ by Eq. (3), there is an input $\mathbf{x} \in \{0, 1\}^n$ such that $[\mathbf{x}]_m \in A$ and hence $C(\mathbf{x}) = \text{MOD}_m^A(\mathbf{x}) = 0$. Thus $\mathbf{p}(\mathbf{x}) \in P_0(C)$ and hence $|P_0(C)| \geq 1$. Thus, the right side of Eq. (20) takes the maximum value when $|P_0(C)| = 1$. Thus Eq. (14) holds. \square

For a threshold circuit C with $n(\geq 2)$ inputs, we denote by C_0 a circuit obtained from C by fixing the n -th variable x_n of input $\mathbf{x} = (x_1, x_2, \dots, x_n)$ to the constant 0. As illustrated in Fig. 1, one can obtain C_0 from C by deleting the n -th input node for x_n and all the wires linked from the node. We call C_0 the *0-fixed* circuit of C . The 0-fixed circuit C_0 has $n - 1$ inputs, but the size of C_0 is the same as that of C .

Let $X_0 \subseteq \{0, 1\}^n$ be a set such that

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

For every input $\mathbf{x}' = (x_1, x_2, \dots, x_{n-1}) \in \{0, 1\}^{n-1}$ to C_0 , let $\mathbf{x} = (x_1, x_2, \dots, x_{n-1}, 0) \in X_0$, then clearly $[\mathbf{x}']_m = [\mathbf{x}]_m$ and $\mathbf{p}_{C_0}(\mathbf{x}') = \mathbf{p}_C(\mathbf{x})$. We thus have

$$\begin{aligned} P(C_0) &= \{\mathbf{p}_{C_0}(\mathbf{x}') \mid \mathbf{x}' \in \{0, 1\}^{n-1}\} \\ &= \{\mathbf{p}_C(\mathbf{x}) \mid \mathbf{x} \in X_0\} \subseteq P(C) \end{aligned} \quad (21)$$

and

$$\begin{aligned} Q(C_0) &= \{(\mathbf{p}_{C_0}(\mathbf{x}'), [\mathbf{x}']_m) \mid \mathbf{x}' \in \{0, 1\}^{n-1}\} \\ &= \{(\mathbf{p}_C(\mathbf{x}), [\mathbf{x}]_m) \mid \mathbf{x} \in X_0\} \subseteq Q(C). \end{aligned} \quad (22)$$

If a threshold circuit C computes the function MOD_m^A of n variables, then clearly the 0-fixed circuit C_0 computes the function MOD_m^A of $n - 1$ variables. We now have the following key lemma on $Q(C)$ and $Q(C_0)$.

Lemma 3. *If a threshold circuit C computes the function MOD_m^A of $n(\geq 1)$ variables, then*

$$|Q(C_0)| + 1 \leq |Q(C)| \quad (23)$$

where $|Q(C_0)|$ is assumed to be 1 if $n = 1$.

Equation (23) is very simple, but the proof is not simple and fairly sophisticated. The proof is omitted in the extended abstract due to the page limitation, but is described in the appendix.

From Lemma 3 one can easily prove the following lower bound on $|Q(C)|$.

Lemma 4. *If a threshold circuit C computes the function MOD_m^A of $n(\geq 1)$ variables, then*

$$n + 1 \leq |Q(C)|. \quad (24)$$

Proof. By Eq. (3) we have $n \geq m - 1$, and hence we prove by induction on n that Eq. (24) holds for every integer n such that $n \geq m - 1$.

For the inductive base, we assume that $n = m - 1$. Clearly, for every integer $i \in M$, there exists an input $\mathbf{x} \in \{0, 1\}^n$ such that $[\mathbf{x}]_m = i$. Thus $|Q(C)| \geq |M| = m = n + 1$, and hence Eq. (24) holds.

For the inductive hypothesis, we assume that $n \geq m(\geq 2)$ and that Eq. (24) holds for every threshold circuit computing the function MOD_m^A of $(n - 1)$ variables. Let C be a threshold circuit computing MOD_m^A of n variables. Since the 0-fixed circuit C_0 of C computes the function MOD_m^A of $n - 1$ variables, the induction hypothesis implies that

$$|Q(C_0)| \geq (n - 1) + 1 = n. \quad (25)$$

Equations (23) and (25) yields

$$|Q(C)| \geq |Q(C_0)| + 1 \geq n + 1.$$

□

There exists a threshold circuit C computing the function MOD_m^A of n variables such that $Q(C) = n + 1$, as illustrated in Fig. 2(a) for $m = 2$ and $A = \{0\}$. Therefore, the lower bound on $|Q(C)|$ in Eq. (24) is best possible.

Using Lemmas 1, 2 and 4, one can easily prove Theorem 1, as follows.

Proof of Theorem 1. Equations (14) and (24) imply that

$$n + 1 \leq (|P(C)| - 1)(m - a) + a. \quad (26)$$

Equations (9) and (26) imply that

$$\frac{n + 1 - a}{m - a} \leq |P(C)| - 1 \leq s^e. \quad (27)$$

□

3.3 Theorem 2

In the section, we present a tradeoff which is better than that in Theorem 1 if $e \geq 5$.

Applying a counting argument ([3, p.102, p.122]) and the Stirling's formula to Eq (10), one can easily prove the following upper bound on $|P(C)|$, which is better than the bound in Eq. (9) if $e \geq 5$:

$$|P(C)| \leq \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e} \right)^e \quad (28)$$

where $c_{npr} \cong 2.718$ is the Napier's (or mathematical) constant. Similarly to the proof of Theorem 1, we can prove the following theorem from Eqs. (26) and (28):

Theorem 2. Let C be a threshold circuit computing the function MOD_m^A of n variables. Then the size s and energy complexity e of C satisfy

$$\frac{n+1-a}{m-a} + 1 \leq \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e} \right)^e. \quad (29)$$

4 Conclusions

The class of generalized mod functions MOD_m^A is fairly large; it includes the ordinary mod function MOD_m , the PARITY function, the OR function, the MAJORITY function, *etc.* In the paper, we show that there exists a very simple tradeoff

$$\frac{n+1-a}{m-a} \leq s^e$$

between the size s and the energy complexity e of a threshold circuit computing MOD_m^A , where n is the number of input variables, $2 \leq m \leq n+1$, and $a = \min\{|A|, |M-A|\}$. Thus, $n/(m-1) \leq s^e$ for the ordinary mod function MOD_m , for which $a = 1$. The inequality immediately implies lower bounds on s and e : $(n/(m-1))^e \leq s$; and $e = \Omega((\log n - \log m)/\log \log n)$ if $s = O(\text{polylog}(n))$. The lower bound on e is tight up to a constant factor. The main idea of the proof of our result is to show that the number of patterns of a circuit is at most $s^e + 1$ and the number of extended patterns is at least $n + 1$. The key Lemma 3 is very simple, but the proof is sophisticated and is interesting in its own right.

In the paper we have so far considered circuits of threshold logic gates, but our result can be extended to a more general class of circuits, as follows. A function $g : \{0, 1\}^n \rightarrow \{0, 1\}$ is called a *unate* (or also called a *generalized monotone*) function if

$$g(z_1, \dots, z_{i-1}, 0, z_{i+1}, \dots, z_n) \leq g(z_1, \dots, z_{i-1}, 1, z_{i+1}, \dots, z_n)$$

or

$$g(z_1, \dots, z_{i-1}, 1, z_{i+1}, \dots, z_n) \leq g(z_1, \dots, z_{i-1}, 0, z_{i+1}, \dots, z_n)$$

holds for each i -th input variable z_i and all the other variables $z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n \in \{0, 1\}$. Clearly the functions for a threshold gate, OR gate, AND gate, *etc.* are unate functions. (See Eq. (4).) Consider a circuit C consisting of logic gates computing unate functions, let the size s of C be the number of gates in C , and let the energy complexity e of C be the maximum number of gates outputting “1” over all inputs. Then one can easily observe that our proof scheme for threshold circuits can be applied to the class of circuits consisting of logic gates computing unate functions and yields the same tradeoffs as in Theorem 1 and Theorem 2.

References

- [1] A. Aggarwal, A. Chandra, and P. Raghavan. Energy consumption in VLSI circuits. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 205–216, 1988.

- [2] A. Chattopadhyay, N. Goyal, P. Pudlak, and D. Therien. Lower bounds for circuits with MOD_m gates. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 709–718, 2006.
- [3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1989.
- [4] V. Grolmusz and G. Tardos. Lower bounds for $(\text{MOD}_p - \text{MOD}_m)$ circuits. *SIAM Journal on Computing*, 29(4):1209–1222, 2000.
- [5] R. Impagliazzo, R. Paturi, and M. E. Saks. Size-depth trade-offs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.
- [6] G. Kissin. Upper and lower bounds on switching energy in VLSI. *Journal of the Association for Computing Machinery*, 38:222–254, 1991.
- [7] P. Lennie. The cost of cortical computation. *Current Biology*, 13:493–497, 2003.
- [8] T. W. Margrie, M. Brecht, and B. Sakmann. In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain. *Pflugers Arch.*, 444(4):491–498, 2002.
- [9] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1988.
- [10] I. Parberry. *Circuit Complexity and Neural Networks*. MIT Press, Cambridge, MA, 1994.
- [11] S. Shao-Chin and T. Nishino. The complexity of threshold circuits for parity functions. *IEICE Transactions on Information and Systems*, 80(1):91–93, 1997.
- [12] J. Sima and P. Orponen. General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation*, 15:2727–2778, 2003.
- [13] K. Y. Siu, V. Roychowdhury, and T. Kailath. *Discrete Neural Computation; A Theoretical Foundation*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1995.
- [14] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):455–466, 1994.
- [15] K. Uchizawa, R. Douglas, and W. Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2006.
- [16] K. Uchizawa and E. Takimoto. Exponential lower bounds on the size of threshold circuits with small energy complexity. *Theoretical Computer Science*, 407(1-3):474–487, 2008.

Appendix: Proof of Lemma 3

In the appendix, we prove Lemma 3, that is, we verify Eq. (23). Suppose that $n = 1$ and $\mathbf{x} = (x_1) \in \{0, 1\}$. Since $n = 1$, $|Q(C_0)|$ is assumed to be 1, and $m = 2$ by Eq. (3). Therefore $[\mathbf{x}]_m = 0$ if $x_1 = 0$, and $[\mathbf{x}]_m = 1$ if $x_1 = 1$. Thus $2 \leq |Q(C)|$, and hence Eq. (23) holds. We may thus assume that $n \geq 2$.

Suppose that a threshold circuit C computes the function MOD_m^A of $n(\geq 2)$ variables, and that C consists of s threshold gates g_1, g_2, \dots, g_s . One may assume that g_1, g_2, \dots, g_s are topologically ordered with respect to the underlying directed acyclic graph of C [3], and that g_s is the output gate of C . Thus, for each i , $1 \leq i \leq s$, the input \mathbf{z}_i of a gate g_i either consists of the inputs x_1, x_2, \dots, x_n to C and the outputs of g_1, g_2, \dots, g_{i-1} or consists of some of them.

Assume for a contradiction that Eq. (23) does not hold, that is,

$$|Q(C_0)| \geq |Q(C)|. \quad (30)$$

By Eq. (22) $Q(C_0) \subseteq Q(C)$. Therefore, Eq. (30) implies that

$$Q(C_0) = Q(C). \quad (31)$$

Let X_1 be a subset of $\{0, 1\}^n$ such that

$$X_1 = \{(x_1, x_2, \dots, x_n) \in \{0, 1\}^n \mid x_n = 1\}.$$

Let Q_1 be a subset of $Q(C)$ such that

$$Q_1 = \{(\mathbf{p}(\mathbf{x}), [\mathbf{x}]_m) \mid \mathbf{x} \in X_1\}. \quad (32)$$

Since $Q_1 \subseteq Q(C)$, we have from Eq. (31)

$$Q_1 \subseteq Q(C_0). \quad (33)$$

In the rest of the section, we derive a contradiction from Eq. (33).

Let $h = |P(C)|$. To derive a contradiction, we construct the following sequence of $2h + 1$ inputs to C :

$$\mathbf{x}_0 \rightarrow \mathbf{y}_0 \rightarrow \mathbf{x}_1 \rightarrow \mathbf{y}_1 \rightarrow \dots \rightarrow \mathbf{x}_{h-1} \rightarrow \mathbf{y}_{h-1} \rightarrow \mathbf{x}_h, \quad (34)$$

where, for every index j ,

$$\mathbf{x}_j \in X_1 \text{ and } \mathbf{y}_j \in X_0. \quad (35)$$

We arbitrarily choose \mathbf{x}_0 from the set X_1 , and choose $\mathbf{y}_0, \mathbf{x}_1, \dots, \mathbf{x}_h$ by the following Procedures 1 and 2.

Procedure 1: $\mathbf{x}_j \rightarrow \mathbf{y}_j$

For each j , $0 \leq j \leq h-1$, we obtain \mathbf{y}_j from \mathbf{x}_j as follows. Equation (35) implies that $\mathbf{x}_j \in X_1$, and hence Eqs. (32) and (33) imply $(\mathbf{p}(\mathbf{x}_j), [\mathbf{x}_j]_m) \in Q_1 \subseteq Q(C_0)$. Therefore, by Eq. (22), there exist one or more inputs $\mathbf{y} \in X_0$ such that

$$(\mathbf{p}(\mathbf{x}_j), [\mathbf{x}_j]_m) = (\mathbf{p}(\mathbf{y}), [\mathbf{y}]_m) \in Q(C_0). \quad (36)$$

We choose \mathbf{y}_j as in the following claim.

Claim 1. *There exists $\mathbf{y}_j \in X_0$ such that*

$$\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y}_j), \quad (37)$$

$$(\mathbf{p}(\mathbf{y}_j), [\mathbf{y}_j]_m) \in Q(C_0), \quad (38)$$

and

$$(\mathbf{p}(\mathbf{y}_j), \mathbf{y}'_j) \notin Q(C_0) \quad (39)$$

where

$$\mathbf{y}'_j = [\mathbf{y}_j]_m + 1 \pmod{m}. \quad (40)$$

Proof. Let $B (\subseteq M)$ be the set of all integers $i \in M$ such that there exists $\mathbf{y} \in X_0$ satisfying $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y})$ and $[\mathbf{y}]_m = i$, and hence $(\mathbf{p}(\mathbf{y}), [\mathbf{y}]_m) \in Q(C_0)$. Then by Eq. (36) we have $[\mathbf{x}_j]_m \in B$ and hence $B \neq \emptyset$.

We now prove that

$$\text{if } [\mathbf{x}_j]_m \in A \text{ then } B \subseteq A. \quad (41)$$

Suppose that $[\mathbf{x}_j]_m \in A$, and let i be an arbitrary integer in B . Since $[\mathbf{x}_j]_m \in A$, by Eq. (1) $\text{MOD}_m^A(\mathbf{x}_j) = g_s(\mathbf{x}_j) = 0$. Since $i \in B$, there exists $\mathbf{y} \in X_0$ satisfying $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y})$ and $[\mathbf{y}]_m = i$. Since $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y})$, $g_s(\mathbf{x}_j) = g_s(\mathbf{y})$. Therefore, $\text{MOD}_m^A(\mathbf{y}) = g_s(\mathbf{y}) = g_s(\mathbf{x}_j) = \text{MOD}_m^A(\mathbf{x}_j) = 0$, and hence by Eq. (1) $i = [\mathbf{y}]_m \in A$. Thus $B \subseteq A$.

Similarly as above, one can prove that

$$\text{if } [\mathbf{x}_j]_m \notin A \text{ then } B \subseteq M - A. \quad (42)$$

We then prove that $B \neq M$. Suppose for a contradiction that $B = M$. If $\text{MOD}_m^A(\mathbf{x}_j) = 0$, then $[\mathbf{x}_j]_m \in A$ and hence by Eq. (41) $A = B = M$. If $\text{MOD}_m^A(\mathbf{x}_j) = 1$, then by Eq. (1) $[\mathbf{x}_j]_m \notin A$ and hence by Eq. (42) $A = \emptyset$. Either case contradicts Eq. (2).

We are now ready to complete the proof of Claim 1. We give a proof only for the case $\text{MOD}_m^A(\mathbf{x}_j) = 0$, because the proof for the other case where $\text{MOD}_m^A(\mathbf{x}_j) = 1$ is similar. Since $\text{MOD}_m^A(\mathbf{x}_j) = 0$, we have $[\mathbf{x}_j]_m \in A$ and hence $B \subseteq A$ by Eq. (41). Since $B \neq \emptyset$ and $B \neq M$, there is an integer $i \in B$

such that $(i + 1 \bmod m) \notin B$. Since $i \in B$, there exists $\mathbf{y}_j \in X_0$ satisfying $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y}_j)$ and $[\mathbf{y}_j]_m = i$. Thus $(\mathbf{p}(\mathbf{y}_j), [\mathbf{y}_j]_m) \in Q(C_0)$. Since $(i + 1 \bmod m) \notin B$, we have

$$\begin{aligned} y'_j &= [\mathbf{y}_j]_m + 1 \pmod{m} \\ &= i + 1 \pmod{m} \\ &\notin B, \end{aligned}$$

and hence, $(\mathbf{p}(\mathbf{y}_j), y'_j) \notin Q(C_0)$. □

Procedure 2: $\mathbf{y}_j \rightarrow \mathbf{x}_{j+1}$

For each j , $0 \leq j \leq h - 1$, we obtain $\mathbf{x}_{j+1} \in X_1$ from $\mathbf{y}_j \in X_0$ simply by flipping the n -th input of $\mathbf{y}_j \in X_0$. Thus, if

$$\mathbf{y}_j = (y_1, y_2, \dots, y_{n-1}, 0), \quad (43)$$

then

$$\mathbf{x}_{j+1} = (y_1, y_2, \dots, y_{n-1}, 1). \quad (44)$$

Repeating the two procedures above h times, we construct the sequence (34) from \mathbf{x}_0 . For every index j , Eq. (35) clearly holds. In addition, Eqs. (43) and (44) imply that

$$[\mathbf{x}_{j+1}]_m = [\mathbf{y}_j]_m + 1 \pmod{m} = y'_j. \quad (45)$$

The sequence (34) of $2h + 1$ inputs corresponds to the following sequence of patterns:

$$\mathbf{p}(\mathbf{x}_0) \rightarrow \mathbf{p}(\mathbf{y}_0) \rightarrow \mathbf{p}(\mathbf{x}_1) \rightarrow \mathbf{p}(\mathbf{y}_1) \rightarrow \dots \rightarrow \mathbf{p}(\mathbf{x}_{h-1}) \rightarrow \mathbf{p}(\mathbf{y}_{h-1}) \rightarrow \mathbf{p}(\mathbf{x}_h). \quad (46)$$

We now prove the following Claim 2 on the sequence (46).

Claim 2. For every index j , $0 \leq j \leq h - 1$,

$$\mathbf{p}(\mathbf{y}_j) \neq \mathbf{p}(\mathbf{x}_{j+1}). \quad (47)$$

Proof. Equations (33) and (39) imply

$$(\mathbf{p}(\mathbf{y}_j), y'_j) \notin Q_1. \quad (48)$$

On the other hand, Eqs. (32) and (45) imply

$$(\mathbf{p}(\mathbf{x}_{j+1}), [\mathbf{x}_{j+1}]_m) = (\mathbf{p}(\mathbf{x}_{j+1}), y'_j) \in Q_1. \quad (49)$$

By Eqs. (48) and (49), we have $\mathbf{p}(\mathbf{y}_j) \neq \mathbf{p}(\mathbf{x}_{j+1})$. □

By Eqs. (37) and (47), $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y}_j)$ and $\mathbf{p}(\mathbf{y}_j) \neq \mathbf{p}(\mathbf{x}_{j+1})$. We rewrite the sequence (46) as follows:

$$\mathbf{p}(\mathbf{x}_0) \Rightarrow \mathbf{p}(\mathbf{y}_0) \rightarrow \mathbf{p}(\mathbf{x}_1) \Rightarrow \mathbf{p}(\mathbf{y}_1) \rightarrow \cdots \rightarrow \mathbf{p}(\mathbf{x}_{h-1}) \Rightarrow \mathbf{p}(\mathbf{y}_{h-1}) \rightarrow \mathbf{p}(\mathbf{x}_h). \quad (50)$$

Then the patterns on the two sides of each arrow “ \Rightarrow ” are same, and the patterns on the two sides of each arrow “ \rightarrow ” are different.

The sequence (50) contains $h + 1$ patterns $\mathbf{p}(\mathbf{x}_0), \mathbf{p}(\mathbf{x}_1), \dots, \mathbf{p}(\mathbf{x}_h)$, but $h = |P(C)|$. Therefore, there is a pair of indices l and r , $0 \leq l < r \leq h$, such that

$$\mathbf{p}(\mathbf{x}_l) = \mathbf{p}(\mathbf{x}_r). \quad (51)$$

We now consider the following subsequence of the sequence (50)

$$\mathbf{p}(\mathbf{x}_l) \Rightarrow \mathbf{p}(\mathbf{y}_l) \rightarrow \mathbf{p}(\mathbf{x}_{l+1}) \Rightarrow \mathbf{p}(\mathbf{y}_{l+1}) \rightarrow \cdots \rightarrow \mathbf{p}(\mathbf{x}_{r-1}) \Rightarrow \mathbf{p}(\mathbf{y}_{r-1}) \rightarrow \mathbf{p}(\mathbf{x}_r), \quad (52)$$

and find a sequence of gates

$$g_l, g_{l+1}, \dots, g_{r-1}, \quad (53)$$

as follows. Equation (47) implies that, for each j , $l \leq j \leq r - 1$, there are one or more gates that output $b \in \{0, 1\}$ for \mathbf{y}_j and output the complement \bar{b} of b for \mathbf{x}_{j+1} . Let g_{i_j} be the gate with the smallest index among all these gates. Thus, for every index j , $l \leq j \leq r - 1$,

$$g_{i_j}[\mathbf{y}_j] \neq g_{i_j}[\mathbf{x}_{j+1}] \quad (54)$$

and for every index k , $1 \leq k \leq i_j - 1$,

$$g_k[\mathbf{y}_j] = g_k[\mathbf{x}_{j+1}]. \quad (55)$$

Let i_t , $l \leq t \leq r - 1$, be the smallest index among $i_l, i_{l+1}, \dots, i_{r-1}$. Then for all indices j , $l \leq j \leq r - 1$, and k , $1 \leq k \leq i_t - 1$, we have

$$g_k[\mathbf{x}_j] = g_k[\mathbf{y}_j] = g_k[\mathbf{x}_{j+1}]. \quad (56)$$

Thus the outputs of the gates $g_1, g_2, \dots, g_{i_t-1}$ do not change in the sequence (52).

If

$$g_{i_t}[\mathbf{x}_l] \neq g_{i_t}[\mathbf{x}_r], \quad (57)$$

then Eq. (57) contradicts Eq. (51). Thus it suffices to prove Eq. (57).

We now prove the following claim.

Claim 3. *The n -th input node x_n of C is directly connected to the gate g_{i_t} , and the weight w_n is not zero.*

Proof. Since $l \leq t \leq r - 1$, by Eq. (54) we have

$$g_{i_t}[\mathbf{y}_t] \neq g_{i_t}[\mathbf{x}_{t+1}]. \quad (58)$$

Since the gates g_1, g_2, \dots, g_s are topologically ordered, the output of g_{i_t} depends only on the outputs of $g_1, g_2, \dots, g_{i_t-1}$ and the inputs x_1, x_2, \dots, x_n to C . By Eq. (56), the outputs of the gates $g_1, g_2, \dots, g_{i_t-1}$ for the input \mathbf{y}_t are same as those for the input \mathbf{x}_{t+1} . Furthermore, Eqs. (43) and (44) imply that \mathbf{x}_{t+1} is different from \mathbf{y}_t only at the n -th input x_n . Thus, Eq. (58) implies that the input node x_n is directly connected to the gate g_{i_t} , and the weight w_n is not zero. (See Eq. (4).) \square

Claim 3 implies that the weight w_n is either $w_n > 0$ or $w_n < 0$. We consider the following two cases in order to verify Eq. (57).

Case 1: $w_n > 0$.

The input to the gate g_{i_t} either consists of the outputs of $g_1, g_2, \dots, g_{i_t-1}$ and the inputs x_1, x_2, \dots, x_n or consists of some of them. Equation (56) implies that the outputs of $g_1, g_2, \dots, g_{i_t-1}$ do not change in the sequence (52). Furthermore, Eqs. (43) and (44) imply that, for every j , $l \leq j \leq r - 1$, \mathbf{x}_{j+1} is different from \mathbf{y}_j only at the n -th input. Consequently, the sum of products of inputs and weights of the gate g_{i_t} increases by exactly $w_n > 0$ at each arrow “ \rightarrow ” in the sequence (52). (See Eq. (4).) Thus, for every j , $l \leq j \leq r - 1$, we have

$$g_{i_t}[\mathbf{y}_j] \leq g_{i_t}[\mathbf{x}_{j+1}]. \quad (59)$$

Equations $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y}_j)$, (58), and (59) imply that

$$0 = g_{i_t}[\mathbf{x}_t] = g_{i_t}[\mathbf{y}_t] \neq g_{i_t}[\mathbf{x}_{t+1}] = 1, \quad (60)$$

and

$$g_{i_t}[\mathbf{x}_j] = 0 \quad (61)$$

for every j , $l \leq j \leq t - 1$, and

$$g_{i_t}[\mathbf{x}_j] = 1 \quad (62)$$

for every j , $t + 2 \leq j \leq r$.

Equations (60)–(62) imply that

$$0 = g_{i_t}[\mathbf{x}_l] \neq g_{i_t}[\mathbf{x}_r] = 1,$$

and hence Eq. (57) holds.

Case 2: $w_n < 0$.

Similarly to Case 1, for every index j , $l \leq j \leq r - 1$, we have

$$g_{i_t}[\mathbf{y}_j] \geq g_{i_t}[\mathbf{x}_{j+1}] \quad (63)$$

since $w_n < 0$. Equations $\mathbf{p}(\mathbf{x}_j) = \mathbf{p}(\mathbf{y}_j)$, (58) and (63) imply that

$$1 = g_{i_t}[\mathbf{x}_t] = g_{i_t}[\mathbf{y}_t] \neq g_{i_t}[\mathbf{x}_{t+1}] = 0, \quad (64)$$

and

$$g_{i_t}[\mathbf{x}_j] = 1 \quad (65)$$

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

$$g_{i_t}[\mathbf{x}_j] = 0 \quad (66)$$

for every index j , $t+2 \leq j \leq r$. Equations (64)–(66) imply that

$$1 = g_{i_t}[\mathbf{x}_l] \neq g_{i_t}[\mathbf{x}_r] = 0,$$

and hence Eq. (57) holds.