Size and Energy of Threshold Circuits Computing Mod Functions

Kei Uchizawa[1,⋆], Takao Nishizeki[1], and Eiji Takimoto[2,⋆⋆]

[1] Graduate School of Information Sciences, Tohoku University,
Aramaki Aoba-aza 6-6-05, Aoba-ku, Sendai, 980-8579, Japan
[2] Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
{uchizawa,nishi}@ecei.tohoku.ac.jp, eiji@i.kyushu-u.ac.jp

Abstract. Let C be a threshold logic circuit computing a Boolean function $\mathrm{MOD}_m : \{0,1\}^n \to \{0,1\}$, where $n \geq 1$ and $m \geq 2$. Then C outputs "0" if the number of "1"s in an input $\boldsymbol{x} \in \{0,1\}^n$ 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 $\boldsymbol{x} \in \{0,1\}^n$. 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 threshold circuits computing MOD_m, and yields a lower bound $e = \Omega((\log n - \log m)/\log\log n)$ on e if $s = O(\mathrm{polylog}(n))$. 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. Our results on threshold circuits can be extended to a more general class of circuits, called unate circuits.

1 Introduction

A circuit of threshold gates is a theoretical model of a neural circuit in the brain, and is well studied through decades [10,11,13,14]. 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" [8,9], while a gate in an electrical circuit consumes almost the same amount of energy in either case of outputting "1" and outputting "0" [1,7]. A biological study reports that, due to the asymmetricity of the energy consumption, the fraction of neurons firing concurrently is possibly fewer than 1% [8]. 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 [16]. We then confront the following natural question from the

⋆ Supported by MEXT Grant-in-Aid for Young Scientists (B) No.21700003.
⋆⋆ Supported by MEXT Grant-in-Aid for Scientific Research (C) No.20500001.

R. Královič and D. Niwiński (Eds.): MFCS 2009, LNCS 5734, pp. 724–735, 2009.

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 [16,18]. In 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 [18]. However, the mod function $\mathrm{MOD}_m : \{0,1\}^n \to \{0,1\}$ has low communication complexity, and hence the result in [18] does not yield any interesting tradeoff for MOD_m, where n and m are positive integers, and $\mathrm{MOD}_m(\boldsymbol{x})$ is 0 if the number of "1"s in an input $\boldsymbol{x} \in \{0,1\}^n$ is a multiple of m and, otherwise, $\mathrm{MOD}_m(\boldsymbol{x})$ 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 [2,3,5], 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 $n/(m-1) \le s^e$, that is, $\log(n/(m-1)) \le e \log s$, 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 $n/(m-1) \le s^e$ immediately implies a lower bound on the size s expressed by n, m and e: $(n/(m-1))^{1/e} \le s$. If $s = O(\mathrm{polylog}(n))$, 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. Our results on threshold circuits can be extended to a more general class of circuits, called "unate circuits," as stated in Section 4.

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 $n \le (s/d)^{d+\epsilon}$ for any fixed $\epsilon > 0$ if the weights of the threshold gates are integers and their absolute values are sufficiently small [15]. Impagliazzo *et al.* proved that $n/2 \le s^{2(d-1)}$ even if the absolute values of weights are arbitrarily large [6]. Our tradeoff between s and e holds even if the absolute values of weights are arbitrarily large. It should be noted that the inequality $d \le e$ does not necessarily holds, and that if a Boolean function f can be computed by a polynomial-size threshold circuit C of energy complexity e then the function f can be computed by a polynomial-size threshold circuit C' of depth $d' = 2e + 1$ [17].

2 Preliminaries

For $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \{0,1\}^n$, we denote by $[\boldsymbol{x}]_m$ the Hamming weight of $\boldsymbol{x}$ modulo m and hence $[\boldsymbol{x}]_m = \sum_{i=1}^n x_i \pmod{m}$. Let $M = \{0, 1, \cdots, m-1\}$, then $m = |M|$. For a set $A \subseteq M$, the *generalized mod function* $\mathrm{MOD}_m^A : \{0,1\}^n \to \{0,1\}$ is defined as follows [2,3,5]:

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

Let $a = \min(|A|, |M - A|)$. We may assume that the generalized mod function MOD_m^A is not trivial, and hence

$$1 \leq a \leq \left\lfloor \frac{m}{2} \right\rfloor \tag{2}$$

and

$$2 \leq m \leq n + 1. \tag{3}$$

If $A = \{0\}$, then MOD_m^A is the ordinary mod function MOD_m, and

$$\mathrm{MOD}_m(\boldsymbol{x}) = \begin{cases} 0 \text{ if } [\boldsymbol{x}]_m = 0; \\ 1 \text{ otherwise.} \end{cases}$$

If $m = 2$ and $A = \{0\}$, then MOD_m^A is the so-called PARITY function. If $m = n + 1$ and $A = \{0\}$, then MOD_m^A is the OR function. If $m = n + 1$ and $A = \{0, 1, \cdots, \lfloor n/2 \rfloor\}$, then MOD_m^A is the MAJORITY function. Thus, the class of generalized mod functions MOD_m^A is fairly large.

In the paper, a *threshold gate* is the so-called linear threshold logic gate, and can have an arbitrary number k of inputs. For every input $\boldsymbol{z} = (z_1, z_2, \cdots, z_k) \in \{0, 1\}^k$ to a threshold gate g with weights $w_1, w_2, \cdots, w_k$ and a threshold t, the output $g(\boldsymbol{z})$ of the gate g for $\boldsymbol{z}$ is defined as follows:

$$g(\boldsymbol{z}) = \begin{cases} 1 \ \text{ if } \sum_{i=1}^{k} w_i z_i \geq t; \\ 0 \ \text{ otherwise,} \end{cases} \tag{4}$$

where $w_1, w_2, \cdots, w_k$ and t are arbitrary real numbers.

A *threshold (logic) circuit* C is a combinatorial circuit of threshold gates, and is expressed by a directed acyclic graph as illustrated in Fig. 1. Let n be the

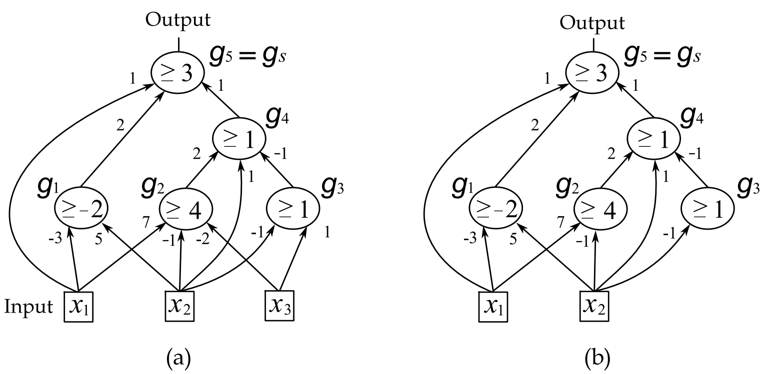

Fig. 1. (a) A threshold circuit C with $n = 3$ and $s = 5$; and (b) the 0-fixed circuit C_0 of C

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

Designs \ Complexities	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. [14]
Moderate s, e and fairly small d	polylog(n)	polylog(n)	$\log n/\log\log n$	Ref. [12]
Moderate s, d and fairly small e	polylog(n)	$\log n/\log\log n$	polylog(n)	Sect. 3.1
Small e and d	$2^{n-1}+1$	1	2	Truth table

number of input variables to C, then C has n input nodes of in-degree 0, each of which corresponds to one of the n input variables $x_1, x_2, \cdots, x_n$.

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 [6].)

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

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

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, and corresponds to the parallel computation time.

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

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

Thus, the energy complexity e is the maximum number of gates outputting "1" over all inputs $\boldsymbol{x} \in \{0,1\}^n$. Clearly $0 \le e \le s$. We may assume without loss of generality that $e \ge 1$.

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."

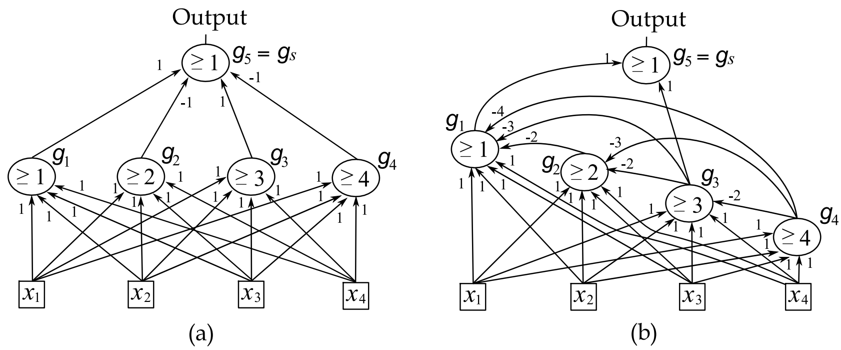

Fig. 2. Threshold circuits computing the PARITY function of $n = 4$ variables; (a) $s = 5$, $d = 2$, $e = 4$; and (b) $s = 5$, $d = 5$, $e = 2$

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 $n/(m - 1) \leq s^e$.
(b) If a threshold circuit C computes the PARITY function, then $n \leq s^e$ and hence $\log n \leq e \log s$.

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.

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 inequality

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

does not hold in general. 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 inequality

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

does not hold if n is sufficiently large.

Equation (5) immediately implies

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

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 $e \le 2$ 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 $s = O(\mathrm{polylog}(n))$, 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 $s = O(\mathrm{polylog}(n))$ and $e = o(\log n/\log\log n)$. Similarly to the corollary above, Sung and Nishino [12] prove that $d = \Theta(\log n/\log\log n)$ if a threshold circuit C with depth d computes the PARITY function and $s = O(\mathrm{polylog}(n))$. Slightly modifying a circuit given in [12], one can construct a threshold circuit of size $s = O(\mathrm{polylog}(n))$ and energy $e = O(\log n/\log\log n)$ that computes the PARITY function of n variables. (See Table 1.) 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

Let a threshold circuit C consist of gates $g_1, g_2, \cdots, g_s$, and let g_s be the output gate of C: $g_s[\boldsymbol{x}] = C(\boldsymbol{x})$ for every $\boldsymbol{x} \in \{0,1\}^n$. For an input $\boldsymbol{x} \in \{0,1\}^n$, we define a *pattern* $\boldsymbol{p}_C(\boldsymbol{x}) \in \{0,1\}^s$ of C for $\boldsymbol{x}$ as $\boldsymbol{p}_C(\boldsymbol{x}) = (g_1[\boldsymbol{x}], g_2[\boldsymbol{x}], \cdots, g_s[\boldsymbol{x}])$. We often denote $\boldsymbol{p}_C(\boldsymbol{x})$ simply by $\boldsymbol{p}(\boldsymbol{x})$. We denote by $P(C)$ the set of all patterns that arise in C: $P(C) = \{\boldsymbol{p}_C(\boldsymbol{x}) \mid \boldsymbol{x} \in \{0,1\}^n\}$.

One can easily prove the following lemma.

Lemma 1. *For an arbitrary threshold circuit C, $|P(C)| \leq s^e + 1$.*

Proof. If $s = 1$, then $|P(C)| \leq 2$, $s^e + 1 = 2$ and hence Lemma 1 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 $\boldsymbol{x}$. Therefore, we have

$$|P(C)| \leq \sum_{i=0}^{e} \binom{s}{i} \tag{6}$$

$$\leq 1 + s + \frac{1}{2}\left(s^2 + s^3 + \cdots + s^e\right)$$

$$\leq 1 + s + \frac{s^2(s^{e-1} - 1)}{2(s-1)}. \tag{7}$$

From Eq. (7) and $s \leq 2(s-1)$, we obtain

$$|P(C)| \leq 1 + s + s(s^{e-1} - 1) = 1 + s^e.$$ □

For every input $\boldsymbol{x} \in \{0,1\}^n$, we define an *extended pattern* $\boldsymbol{q}_C(\boldsymbol{x}) \in \{0,1\}^s \times M$ of a threshold circuit C for $\boldsymbol{x}$ as follows: $\boldsymbol{q}_C(\boldsymbol{x}) = (\boldsymbol{p}_C(\boldsymbol{x}), [\boldsymbol{x}]_m)$, where $M = \{0, 1, \cdots, m-1\}$. We often denote $\boldsymbol{q}_C(\boldsymbol{x})$ simply by $\boldsymbol{q}(\boldsymbol{x})$. We denote by $Q(C)$ the set of all extended patterns that arise in C: $Q(C) = \{\boldsymbol{q}_C(\boldsymbol{x}) \mid \boldsymbol{x} \in \{0,1\}^n\}$. Since $|M| = m$, we have $|Q(C)| \leq |P(C)| \cdot m$. A better upper bound on $|Q(C)|$ can be obtained for a circuit C computing MOD_m^A, as follows.

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

$$|Q(C)| \leq (|P(C)| - 1)(m - a) + a. \tag{8}$$

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 $P_0(C)$:

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

and

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

Since g_s is the output gate of C, we have $g_s[\boldsymbol{x}] = 1$ if $C(\boldsymbol{x}) = 1$, and $g_s[\boldsymbol{x}] = 0$ if $C(\boldsymbol{x}) = 0$. 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 $Q_0(C)$:

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

and

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

Clearly

$$|P(C)| = |P_1(C)| + |P_0(C)| \tag{9}$$

and

$$|Q(C)| = |Q_1(C)| + |Q_0(C)|. \tag{10}$$

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

$$|Q_1(C)| \le |P_1(C)| \cdot (m-a). \tag{11}$$

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

$$|Q_0(C)| \le |P_0(C)| \cdot a \tag{12}$$

Substituting Eqs. (11) and (12) to Eq. (10), we have

$$|Q(C)| \le |P_1(C)| \cdot (m-a) + |P_0(C)| \cdot a. \tag{13}$$

Equations (9) and (13) imply that

$$\begin{aligned} |Q(C)| &\le (|P(C)| - |P_0(C)|) \cdot (m-a) + |P_0(C)| \cdot a \\ &= |P(C)| \cdot (m-a) - |P_0(C)| \cdot (m-2a). \end{aligned} \tag{14}$$

By Eq. (2) we have $m - 2a \ge 0$. Therefore, the right side of Eq. (14) is non-increasing with respect to $|P_0(C)|$. Since $a \ge 1$ by Eq. (2), we have $A \ne \emptyset$. Since $A \subseteq M = \{0, 1, \cdots, m-1\}$ and $m - 1 \le n$ by Eq. (3), there is an input $\boldsymbol{x} \in \{0,1\}^n$ such that $[\boldsymbol{x}]_m \in A$ and hence $C(\boldsymbol{x}) = \mathrm{MOD}_m^A(\boldsymbol{x}) = 0$. Therefore, $\boldsymbol{p}(\boldsymbol{x}) \in P_0(C)$ and hence $|P_0(C)| \ge 1$. Thus, the right side of Eq. (14) takes the maximum value when $|P_0(C)| = 1$, and hence Eq. (8) holds. □

For a threshold circuit C with $n(\ge 2)$ inputs, we denote by C_0 a circuit obtained from C by fixing the n-th variable x_n of input $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ 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.

Define $X_0 \subseteq \{0,1\}^n$ as follows: $X_0 = \{(x_1, x_2, \cdots, x_n) \in \{0,1\}^n \mid x_n = 0\}$. For each input $\boldsymbol{x}' = (x_1, x_2, \cdots, x_{n-1}) \in \{0,1\}^{n-1}$ to the 0-fixed circuit C_0 of C, let $\boldsymbol{x} \in X_0$ be the input to C such that $\boldsymbol{x} = (x_1, x_2, \cdots, x_{n-1}, 0)$. Then clearly $[\boldsymbol{x}']_m = [\boldsymbol{x}]_m$ and $\boldsymbol{p}_{C_0}(\boldsymbol{x}') = \boldsymbol{p}_C(\boldsymbol{x})$. We thus have

$$\begin{aligned} P(C_0) &= \{\boldsymbol{p}_{C_0}(\boldsymbol{x}') \mid \boldsymbol{x}' \in \{0,1\}^{n-1}\} \\ &= \{\boldsymbol{p}_C(\boldsymbol{x}) \mid \boldsymbol{x} \in X_0\} \subseteq P(C) \end{aligned} \tag{15}$$

and

$$\begin{aligned} Q(C_0) &= \{(\boldsymbol{p}_{C_0}(\boldsymbol{x}'), [\boldsymbol{x}']_m) \mid \boldsymbol{x}' \in \{0,1\}^{n-1}\} \\ &= \{(\boldsymbol{p}_C(\boldsymbol{x}), [\boldsymbol{x}]_m) \mid \boldsymbol{x} \in X_0\} \subseteq Q(C). \end{aligned} \quad (16)$$

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 $|Q(C_0)| + 1 \leq |Q(C)|$ where $|Q(C_0)|$ is assumed to be 1 if $n = 1$.*

Sketchy proof. Clearly Lemma 3 holds for the case where $n = 1$, and hence one may 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 $g_1, g_2, \cdots, g_s$. One may assume that $g_1, g_2, \cdots, g_s$ are topologically ordered with respect to the underlying directed acyclic graph of C, and that g_s is the output gate of C. Assume for a contradiction that Lemma 3 does not hold. Then, $Q(C_0) = Q(C)$ because $Q(C_0) \subseteq Q(C)$ by Eq. (16). Let X_1 be the subset of $\{0,1\}^n$ such that $X_1 = \{(x_1, x_2, \cdots, x_n) \in \{0,1\}^n \mid x_n = 1\}$. Let Q_1 be the subset of $Q(C)$ such that $Q_1 = \{(\boldsymbol{p}(\boldsymbol{x}), [\boldsymbol{x}]_m) \mid \boldsymbol{x} \in X_1\}$. Then we have $Q_1 \subseteq Q(C) = Q(C_0)$.

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

$$\boldsymbol{x}_0 \to \boldsymbol{y}_0 \to \boldsymbol{x}_1 \to \boldsymbol{y}_1 \to \cdots \to \boldsymbol{x}_{h-1} \to \boldsymbol{y}_{h-1} \to \boldsymbol{x}_h, \quad (17)$$

where $\boldsymbol{x}_j \in X_1$ and $\boldsymbol{y}_j \in X_0$ for every index j. We arbitrarily choose $\boldsymbol{x}_0$ from the set $X_1(\neq \emptyset)$, and choose $\boldsymbol{y}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_h$ as in the following (a) and (b):

(a) From $\boldsymbol{x}_j$, $0 \leq j \leq h-1$, we obtain $\boldsymbol{y}_j \in X_0$ such that $\boldsymbol{p}(\boldsymbol{y}_j) = \boldsymbol{p}(\boldsymbol{x}_j)$ and $(\boldsymbol{p}(\boldsymbol{y}_j), y'_j) \notin Q(C_0)$ for $y'_j = [\boldsymbol{y}_j]_m + 1 \pmod{m}$.

(b) From $\boldsymbol{y}_j \in X_0$, $0 \leq j \leq h-1$, we obtain $\boldsymbol{x}_{j+1} \in X_1$ simply by flipping the n-th input of $\boldsymbol{y}_j \in X_0$.

The sequence (17) corresponds to the following sequence of patterns:

$$\boldsymbol{p}(\boldsymbol{x}_0) \to \boldsymbol{p}(\boldsymbol{y}_0) \to \boldsymbol{p}(\boldsymbol{x}_1) \to \boldsymbol{p}(\boldsymbol{y}_1) \to \cdots \to \boldsymbol{p}(\boldsymbol{x}_{h-1}) \to \boldsymbol{p}(\boldsymbol{y}_{h-1}) \to \boldsymbol{p}(\boldsymbol{x}_h). \quad (18)$$

One can prove that $\boldsymbol{p}(\boldsymbol{x}_j) = \boldsymbol{p}(\boldsymbol{y}_j) \neq \boldsymbol{p}(\boldsymbol{x}_{j+1})$ for each j, $0 \leq j \leq h-1$. Therefore, the sequence (18) contains $h+1$ patterns $\boldsymbol{p}(\boldsymbol{x}_0), \boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_h)$, but $h = |P(C)|$. Thus, there is a pair of indices l and r, $0 \leq l < r \leq h$, such that $\boldsymbol{p}(\boldsymbol{x}_l) = \boldsymbol{p}(\boldsymbol{x}_r)$. We now consider the following subsequence of (18)

$$\boldsymbol{p}(\boldsymbol{x}_l) \to \boldsymbol{p}(\boldsymbol{y}_l) \to \boldsymbol{p}(\boldsymbol{x}_{l+1}) \to \boldsymbol{p}(\boldsymbol{y}_{l+1}) \to \cdots \to \boldsymbol{p}(\boldsymbol{x}_{r-1}) \to \boldsymbol{p}(\boldsymbol{y}_{r-1}) \to \boldsymbol{p}(\boldsymbol{x}_r),$$

and find a sequence of gates $g_{i_l}, g_{i_{l+1}}, \cdots, g_{i_{r-1}}$, as follows. Since $\boldsymbol{p}(\boldsymbol{y}_j) \neq \boldsymbol{p}(\boldsymbol{x}_{j+1})$ for each j, $l \leq j \leq r-1$, there are one or more gates that output $b \in \{0,1\}$ for $\boldsymbol{y}_j$ and output the complement $\bar{b}$ of b for $\boldsymbol{x}_{j+1}$. Let g_{i_j} be the gate with the smallest index among all these gates. Let i_t, $l \le t \le r-1$, be the smallest index among $i_l, i_{l+1}, \cdots, i_{r-1}$. Then one can prove that 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. From the fact one can derive $g_{i_t}[\boldsymbol{x}_l] \ne g_{i_t}[\boldsymbol{x}_r]$, which contradicts to $\boldsymbol{p}(\boldsymbol{x}_l) = \boldsymbol{p}(\boldsymbol{x}_r)$. The details are omitted, due to the page limitation. □

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 $n(\ge 1)$ variables, then*

$$n + 1 \le |Q(C)|. \tag{19}$$

Proof. By Eq. (3) we have $n \ge m - 1$, and hence we prove by induction on n that Eq. (19) holds for every integer n such that $n \ge m - 1$.

For the inductive basis, we assume that $n = m - 1$. Clearly, for every integer $i \in M$, there exists an input $\boldsymbol{x} \in \{0, 1\}^n$ such that $[\boldsymbol{x}]_m = i$. Thus $|Q(C)| \ge |M| = m = n + 1$, and hence Eq. (19) holds.

For the inductive hypothesis, we assume that $n \ge m(\ge 2)$ and that Eq. (19) 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 $|Q(C_0)| \ge (n-1)+1 = n$. Therefore, by Lemma 3 we have $|Q(C)| \ge |Q(C_0)| + 1 \ge n + 1$. □

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 Lemma 4 is best possible.

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

Proof of Theorem 1. By Lemma 2 and Lemma 4 we have

$$n + 1 \le (|P(C)| - 1)(m - a) + a. \tag{20}$$

Slightly modifying Eq. (20) and using Lemma 1, we have

$$\frac{n + 1 - a}{m - a} \le |P(C)| - 1 \le s^e. \qquad \square$$

3.3 Theorem 2

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

Applying a counting argument ([4, p.102, p.122]) and the Stirling's formula to Eq. (6), one can easily prove the following upper bound on $|P(C)|$, which is better than the bound in Lemma 1 if $e \ge 5$:

$$|P(C)| \le \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e}\right)^e \tag{21}$$

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. (20) and (21).

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*

$$\frac{n+1-a}{m-a} + 1 \le \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e} \right)^e . \tag{22}$$

4 Conclusions

In the paper, we show that there exists a very simple tradeoff $(n+1-a)/(m-a) \le s^e$ 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, $2 \le m \le n+1$, and $a = \min\{|A|, |M-A|\}$. 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 $n+1$.

We have so far considered circuits of threshold logic gates, but our result can be extended to a more general class of circuits, called "unate circuits." A function $g(z_1, z_2, \cdots, z_k) : \{0,1\}^k \to \{0,1\}$ is said to be *unate in variable* z_i if

$$g(z_1, \cdots, z_{i-1}, 0, z_{i+1}, \cdots, z_k) \le g(z_1, \cdots, z_{i-1}, 1, z_{i+1}, \cdots, z_k)$$

or

$$g(z_1, \cdots, z_{i-1}, 1, z_{i+1}, \cdots, z_k) \le g(z_1, \cdots, z_{i-1}, 0, z_{i+1}, \cdots, z_k)$$

holds for all $z_1, \cdots, z_{i-1}, z_{i+1}, \cdots, z_k \in \{0,1\}$. A function g is said to be *unate* if g is unate in every variable z_i, $1 \le i \le k$. A *unate gate* is a logical gate computing a unate function. Clearly, a threshold gate, OR gate, AND gate, *etc.* are unate gates, while there is a unate function which cannot be computed by any single threshold gate. A *unate circuit* is a combinatorial circuit C consisting of unate gates, 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 observe that our proof scheme for threshold circuits can be applied to unate circuits and yields the same tradeoffs as in Theorem 1 and Theorem 2.

References

1. Aggarwal, A., Chandra, A., Raghavan, P.: Energy consumption in VLSI circuits. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, pp. 205–216 (1988)
2. Beigel, R., Maciel, A.: Upper and lower bounds for some depth-3 circuit classes. Computational Complexity 6(3), 235–255 (1997)
3. Chattopadhyay, A., Goyal, N., Pudlak, P., Therien, D.: Lower bounds for circuits with MOD_m gates. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 709–718 (2006)

4. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, Cambridge (1989)
5. Grolmusz, V., Tardos, G.: Lower bounds for (MOD_p - MOD_m) circuits. SIAM Journal on Computing 29(4), 1209–1222 (2000)
6. Impagliazzo, R., Paturi, R., Saks, M.E.: Size-depth trade-offs for threshold circuits. SIAM Journal on Computing 26(3), 693–707 (1997)
7. Kissin, G.: Upper and lower bounds on switching energy in VLSI. Journal of the Association for Computing Machinery 38, 222–254 (1991)
8. Lennie, P.: The cost of cortical computation. Current Biology 13, 493–497 (2003)
9. Margrie, T.W., Brecht, M., Sakmann, B.: In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain. Pflugers Arch. 444(4), 491–498 (2002)
10. Minsky, M., Papert, S.: Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge (1988)
11. Parberry, I.: Circuit Complexity and Neural Networks. MIT Press, Cambridge (1994)
12. Shao-Chin, S., Nishino, T.: The complexity of threshold circuits for parity functions. IEICE Transactions on Information and Systems 80(1), 91–93 (1997)
13. Sima, J., Orponen, P.: General-purpose computation with neural networks: A survey of complexity theoretic results. Neural Computation 15, 2727–2778 (2003)
14. Siu, K.Y., Roychowdhury, V., Kailath, T.: Discrete Neural Computation; A Theoretical Foundation. Prentice-Hall, Inc., Upper Saddle River (1995)
15. Siu, K.Y., Roychowdhury, V.P., Kailath, T.: Rational approximation techniques for analysis of neural networks. IEEE Transactions on Information Theory 40(2), 455–466 (1994)
16. Uchizawa, K., Douglas, R., Maass, W.: On the computational power of threshold circuits with sparse activity. Neural Computation 18(12), 2994–3008 (2006)
17. Uchizawa, K., Nishizeki, T., Takimoto, E.: Energy complexity and depth of threshold circuits. In: Proceedings of the 17th International Symposium on Fundamentals of Computation Theory. Springer, Heidelberg (to appear)
18. Uchizawa, K., Takimoto, E.: Exponential lower bounds on the size of threshold circuits with small energy complexity. Theoretical Computer Science 407(1-3), 474–487 (2008)