1991

G. Kissin, J.T. Tromp

The energy complexity of threshold and other functions
(Preliminary version)

Computer Science/Department of Algorithmics and Architecture Report CS-R9101 January

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

The Energy Complexity of Threshold and Other Functions
(Preliminary Version)

Gloria Kissin
John Tromp

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
and
Department of Mathematics and Computer Science
University of Amsterdam, Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands

ABSTRACT

A novel construction is described that yields fast, minimum energy VLSI circuits that compute threshold and counting functions. The results are obtained in the Uniswitch Model of switching energy.

1985 Mathematics Subject Classification: 68Q05, 68Q35
CR Categories: B.7.1, F.1.1
Key Words and Phrases: circuits, counting, embedding, energy efficient, switching energy, threshold, uniswitch, USM, VLSI.

Report CS-R9101
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The Energy Complexity of Threshold and Other Functions

1. INTRODUCTION

This paper obtains upper bounds on the switching energy used to compute threshold, majority and counting functions with VLSI circuits. The bounds on K-threshold and count-to-K functions, for K a constant, are optimal, ie. linear in the length of the input.

Switching energy is theoretically interesting because it is believed [MC80] to be intrinsic to computation and a fundamental complexity measure of VLSI computations. Energy is practically motivated in VLSI design because energy consumed by a circuit is transformed into heat. How well a circuit can dissipate heat determines its operational limitations. Thus, the less heat produced the better. Further, energy considerations determine a significant portion of the overall costs of a computer [Me86].

Common to all physical devices is the switching energy [MC80] consumed when a wire or gate changes state from 1 to 0 or vice versa. The amount of switching energy consumed is proportional to the area switched.

The results in this paper are obtained in the *Uniswitch Model* (*USM*) of energy consumption, described in the next section. *USM* was first defined by Kissin [Ki82] and has become the primary model for the asymptotic analysis of switching energy ([Le84], [ST86], [Ty87], [ACR88]). Kissin [Ki85] also described the first energy saving design technique, by obtaining energy-efficient circuits for *OR*, *AND*, *Compare* and *Addition* functions. Lengauer and Mehlhorn [LM81] showed that n−input functions realizeable in $AT^2 = O(n^2)$ require $\Omega(AT)$ switching energy, where A is *area* and T is *time* in the Thompson model [Th80]. Aggarwal et al [ACR88] improved the result of Lengauer and Mehlhorn to obtain an $\Omega(n^2)$ energy bound for the class of *transitive* functions [Vu83]. Leo [Le84] showed that, for a specialized circuit basis, the parity function requires $\Omega(A)$ average switching energy, where A is the area of the parity circuit. Tyagi [Ti87] studied the average energy consumption of logic level structures such as PLAs.

USM measures the differences between two stable states of a circuit. Race conditions (aka hazards) are neglected; they are the domain of the *Multiswitch Models*, which are defined and discussed in [Ki87] and [Ki90]. *USM* provides a lower bound on the total energy consumed by a circuit.

The rest of this paper is organized as follows. Section 2 describes and motivates the *Uniswitch Model* of energy consumption, aka *uniswitch energy*. (The term *energy* refers to switching energy for the duration of this paper). The definition of a circuit is extended to include wires with bandwidth > 1. In section 3, upper bounds are obtained in *USM*. In particular, a fast VLSI circuit is described for *K−threshold* functions, which is optimal in consuming $O(n)$ worst case uniswitch energy. The threshold construction also yields bounds on majority and counting functions. Conclusions follow in section 4.

2. THE SETTING

The *Uniswitch Model* of energy consumption defines an energy cost measure for VLSI circuits. *USM* measures the differences between pairs of states of a circuit. The following discussion sets the stage for a precise definition of *USM*.

A *VLSI circuit* is a combinational circuit [Bo77] embedded in a plane as in [BK81]. Salient assumptions of the VLSI circuit model that are important to *USM* are as follows. A circuit is acyclic. A wire (edge) in a VLSI circuit has constant width > 0 and constant bandwidth ≥ 1. At most a constant number of wires, $v \geq 2$, can overlap or intersect at any point in a VLSI circuit. A gate (aka node) has constant area $\geq \lambda^2$. Input nodes have fanin 0 and fanout 1. Output nodes have fanout 0. A non-input node has constant fanin ≥ 1. A non-input, non-output node has constant fanout ≥ 1 and $\leq r$. A non-input node

computes a binary function of $\leq 2r$ inputs in constant time (see discussion below). Gates are separated by distances $\geq \lambda$.

Definition:

Let w be a wire in a VLSI circuit such that w has bandwidth b. If $b > 1$, then w is a *cable* consisting of b parallel wires, each of minimum width λ, bandwidth 1, and separated by distance λ. w has constant width $2\lambda b - 1$.

The threshold circuit described in this paper uses a circuit basis that includes addition, subtraction, minimum, and comparison functions of 2 k-bit numbers, for k a constant. These functions can be decomposed into standard boolean subcircuits that use only a constant amount of resources, as long a k remains constant. In the analysis that follows, node complexity is neglected when k is constant.

Definitions:

A *legal state*, (hereafter also called *state* or *stable state*) s, is a function that attributes values to the nodes and wires of a circuit C. ie. $C = (V, W)$ where V is the node set with maximum degree d, and W is the set of wires. $s: V \cup W \rightarrow \{0,1\}^r$, for $r \leq d$. Input node x has some value x_0 where $x_0 \in \{0,1\}$. Edge w emanating from input node x has value $s(w) = x_0$. Non-input nodes and edges have values consistent with the inputs and the labeling of the nodes (eg. $s(+(1,1)) = 10$, $s(\min(101,100)) = 100$). s_X denotes the state of C for input X. $X \rightarrow s_X$ is a bijection between an input vector and a state of circuit C. Since a state and its associated input vector are closely allied, they are used interchangeably in the following discussion. C is in state s_i at time t_i. s_0 is the *initial state* of C.

The switching energy of a circuit C is defined on a pair of states. In particular, we are interested in what happens when one input vector to C is replaced by another input vector. In the definitions that follow, the pair of states in question is often denoted as (s_0, X), where s_0 is the initial state and X is an input vector that induces a second (ie. final) state.

Definitions:

Suppose VLSI circuit C changes state from s_0 to s_f, denoted $C: s_0 \rightarrow s_f$. Further assume that wire w has initial value $s_0(w) = w_0$ and final value $s_f(w) = w_f$ where $w_0, w_f \in \{0,1\}^b$, for b the bandwidth of w. This change in the value of w is denoted $w: w_0 \rightarrow w_f$. Then w *is switched* (*switches*) iff $w_0 \neq w_f$. Let h_w be the Hamming distance between w_0 and w_f. A wire w of length L that switches accounts for $(h_w \times L)/p$ *switching energy*, where $p > 0$ is the area of wire that accounts for 1 unit of switching energy. Let v be the node at the tail of wire w; then v *switches* iff w *switches*. A node of area A that switches accounts for A/p *switching energy* If $W = \{w\}$ is the set of wires in circuit C, $V = \{v\}$ is the set of nodes in C, and X is the input set such that $C: s_0 \rightarrow X$, then the *wire energy*, E_w, consumed by C is $E_w(C, s_0, X) \triangleq \frac{1}{p} \sum_{w \in W} h_w \times \|w\|$, where $\|w\|$ is the area of wire w; and the *node energy*, E_v, consumed by C is $E_v(C, s_0, X) \triangleq \frac{1}{p} \sum_{\substack{v \in V \\ s_0(w) \neq s_X(w)}} \|v\|$, where $\|v\|$ is the area of node v. Let $E_{w+v}(C, s_0, X) = E_w(C, s_0, X) + E_v(C, s_0, X)$. $E_{w+v}(C, s_0, X) \leq \frac{\text{area}(C)}{k}$ where area(C) is the total area of C.

Definitions:

If C_n is a VLSI circuit computing $f_n: \{0,1\}^n \rightarrow \{0,1\}^m$ such that C_n is in state s_0 at time t_0, and $E_{w+v}(C_n, s_0, X)$ is the energy consumed by f_n when $X = (x_1, \ldots, x_n)$ is the input to C_n at time $t > t_0$, then $E_{worst}(C_n)$, the *worst case uniswitch energy*, is given by

$$E_{worst}(C_n) \triangleq \max_{(s_0,X)} E_{w+v}(C_n, s_0, X)$$

and $E_a(C_n)$, the *average case uniswitch energy* is given by

$$E_a(C_n) \triangleq 2^{-2n} \sum_{(s_0,X)} E_{w+v}(C_n, s_0, X)$$

where 2^{2n} is the number of (s_0, X) pairs. This definition of $E_a(C_n)$ assumes that the input vector is uniformly distributed over $\{0,1\}^n$.

Definitions:

A function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is *energy efficient* iff $\exists$ a family $C = (C_n)_{(n \in \mathbf{N})}$ of circuits with C_n realizing $f \lceil \{0,1\}^n$, and $E_{worst}(C_n) = \Theta(n)$. Circuit family $C = (C_n)_{(n \in \mathbf{N})}$ is *energy efficient* iff $\forall$ families $\hat{C} = (\hat{C}_n)_{(n \in \mathbf{N})}$ of circuits with $\Phi(\hat{C}) = \Phi(C)$: $E_{worst}(\hat{C}_n) = \Omega(E_{worst}(C_n))$.

Throughout this paper, $\log n$ means $\log_2 n$.

2.1 Model Motivation

The intent of this section is to motivate the *Uniswitch Model* in light of physical considerations. *USM* is a good model for obtaining lower bounds because it conservatively estimates a circuit's switching behaviour. Thus, a lower bound in *USM* is an equally valid lower bound on *multiswitch* energy.

USM takes no notice of how a circuit arrives at a particular state. This is the domain of the *Multiswitch Models*, which are discussed in [Ki87] and [Ki90]. However, in order to discuss the relevance of using *USM* to obtain upper bounds, the following multiswitch notions are introduced.

The switching behaviour of physical circuits is influenced by various delay functions, such as gate delay δ, wire delay Δ and input delay I. δ determines the switching speed of a gate. Δ determines the time to transmit a bit along a wire. I determines when an input value arrives at an input port.

Definitions:

Let (C_n, δ, Δ, I) denote a *circuit scheme*, where C_n is a VLSI circuit with gate delay δ, wire delay Δ, and input delay I. A circuit scheme (C_n, δ, Δ, I) exhibits the *uniswitch property* if each node or wire of C_n switches at most once when C_n changes from one input setting to another, according to δ, Δ and I. Otherwise, (C_n, δ, Δ, I) exhibits the *multiswitch property*.

Using *USM* to obtain upper bounds is justified for circuit schemes that exhibit the uniswitch property. For example, if each node of a circuit receives its inputs at the same time, race conditions cannot arise. The uniswitch property is thus ensured. Some real circuits have this timing property. Where race conditions derive solely from a circuit's asynchrony (ie. the paths to a node vary in length), a circuit scheme can acquire the uniswitch property if the circuit can be made synchronous. A "bad" input schedule can be offset by varying gate delays. These approaches to designing circuit schemes that achieve the uniswitch property are discussed in [Ki87] and [Ki90]. Further, according to C. Mead [Me86], many CMOS designs are synchronized to ensure that the corresponding circuit schemes have the uniswitch property.

USM is the first step in the systematic asymptotic analysis of switching energy consumption in VLSI circuits. As such, *USM* is justified as an upper bound model. In addition, *USM* is motivated by designers' practical efforts to prevent hazards and thus ensure the uniswitch property. *USM* is used for upper bound analysis in [Ty87] and [ACR88].

USM is defined for acyclic circuits. The study of combinational circuits (without loops) has a long and distinguished history. Krohn and Rhodes' [KR65ab] seminal work in this area showed that each

sequential machine (with loops) can be decomposed into structures consisting of only combinational circuits and flip-flops.

A recommended architecture for sequential machines is the finite state machine in which the combinational logic is isolated from the looping structure [MC80]. See Figure 2.0. This architecture lends itself to analysis of the combinational logic distinct from the looping buffers.

3. WORST CASE UPPER BOUNDS

3.1 Energy-Efficient K-Threshold Circuit

In this section, a novel energy-efficient threshold circuit is described, which uses the techniques of the energy-efficient *OR* and *AND* circuits described in [Ki85] ,[Ki91], and some additional novelty. The energy-efficient *OR* circuit used the observation that it is sufficient to turn on one *OR* input to turn on the output. Thus, when many inputs to *SOR* (*S*mart *OR* circuit) are "1", all but one of these "1" signals are "killed". In a completely analagous manner, it is sufficient to turn off one *AND* input in order to turn off the output. Thus, when many inputs are turned off, only one "0" signal must propagate all the way to the output. The principle idea is that the input may provide more information than the function requires, and suppressing unneeded input bits results in energy savings. This idea was also used in [Ki85] to design energy-efficient comparator circuits.

A threshold function, T_K, is defined on a boolean vector $X = \{x_1, \cdots, x_n\}$ as follows: $T_K(X) = 1$ iff $x_1 + x_2 + \cdots + x_n \geq K$. Hence, at most K input bits that are "1" must reach the output; the rest can be "killed". The energy-efficient threshold circuit, $K\text{–}Thr$, described below, effectively uses only necessary information, killing off the rest, resulting in only linear worst case switching energy consumption.

In a novel way, $K\text{–}Thr$ uses $(K+1)$-ary logic to count to K. It contains 2 types of nodes: $+$-nodes that sum the inputs, and min-nodes that "kill" inputs that exceed K. A third node that compares the final sum with K occurs once.

Like the layout of the *SOR*/*SAND* circuit, the $K\text{–}Thr$ circuit is laid out so that the area used to "kill" signals is at most linear in the input size, and the area of both the "successful" paths to the output plus the "killed" paths is at most linear in the input size.

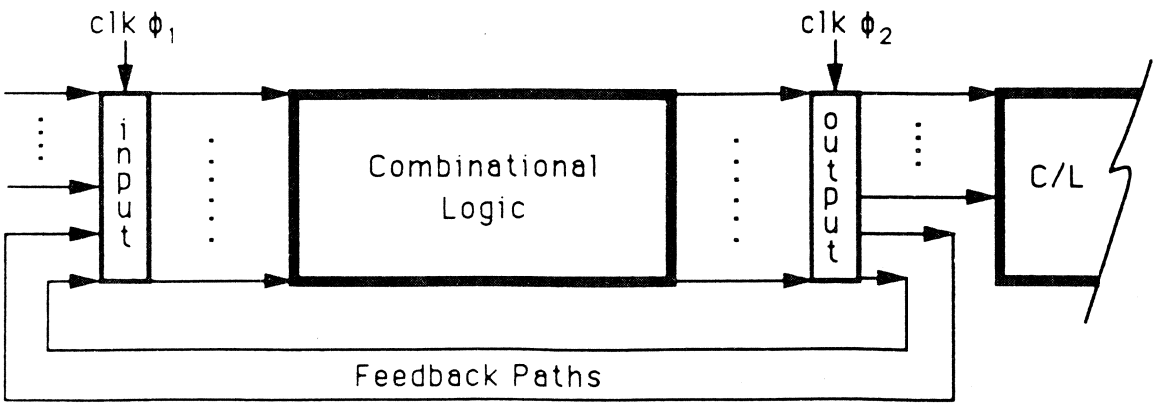


Figure 2.0. Preferred Sequential Circuit: a Finite State Machine

The Energy Complexity of Threshold and Other Functions

The following recurrences describe the boolean K–*threshold* functions $T_K : \{0,1\}^n \rightarrow \{0,1\}$ in a novel way. n is assumed to be a power of 2; the generalization to other values of n is straightforward. "min" is an abbreviation for "minimum". The reader can verify that $T_K(x_1, \ldots, x_n) = 1$ iff $x_1 + x_2 + \cdots + x_n \geq K$. The *USM* realization of T_K is the energy-efficient K–*Thr* circuit.

Recurrences:

$$T_K(x_1, \ldots, x_n) = (LT(x_1, \ldots, x_{n/2}) + RT(x_{(n/2)+1}, \ldots, x_n)) \geq K$$

$$LT(x_i, x_j) = RT(x_i, x_j) = x_i + x_j$$

$$LT(x_1, \ldots, x_n) = LT(x_1, \ldots, x_{n/2}) + \min\,(RT(x_{(n/2)+1}, \ldots, x_n),\ K - LT(x_1, \ldots, x_{n/2}))$$

$$RT(x_1, \ldots, x_n) = \min\,(LT(x_1, \ldots, x_{n/2}),\ K - RT(x_{(n/2)+1}, \ldots, x_n)) + RT(x_{(n/2)+1}, \ldots, x_n)$$

$T_K(x_1, \ldots, x_n)$ is abbreviated by $T_K(n)$. $LT(x_1, \ldots, x_n)$ is abbreviated by $LT(n)$. RT is similarly abbreviated. $(x_l, \ldots, x_n)$ is also written as $(x_l ; x_n)$.

The discussion that follows is a formal description of the construction used to obtain energy-efficient threshold circuits. To clarify the formalism, the reader is advised to refer to Figures 3.0, 3.1 and 3.2, which illustrate a VLSI circuit called LF. LF is an embedding in the plane of circuit K–*Thr*, which computes threshold function T_K. Figure 3.0 illustrates the bottom level of the construction; Figure 3.1 recursively depicts LF on n inputs without the top level; and Figure 3.2 illustrates the top level. Circuit K–*Thr* is formally defined below.

Definition:

K–$Thr(n) = (V_{SD}, W_{SD})$ is a circuit, illustrated in Figures 3.0, 3.1 and 3.2, such that

$V_{SD} = I \cup L$ where

I are the input nodes $\{x_1, x_2, \ldots, x_n\}$.

$L = L_1 \cup L_2 \cup L_3$, the set of interior nodes, is defined as follows. Each node $v \in L$ is labelled by a pair consisting of the function computed at v and a unique identifier.

$L_1 = \{(+, v_1^{1,k}), (+, v_2^{1,k}) \mid 1 \leq k \leq \frac{n}{2}\}$,

$L_2 = \{(+, v_1^{i,k}), (+, v_2^{i,k}), (\min, v_3^{i,k}), (\min, v_4^{i,k}) \mid 2 \leq i \leq \log_2 n - 1,\ 1 \leq k \leq \frac{n}{2^i}\}$, and

$L_3 = \{(+, v^{\log n, 1}), (\geq K, v^{\log n + 1, 1})\}$.

$v^{\log n + 1, 1}$ is the output node. For consistency, x_k is also denoted as $v^{0,k}$.

$W_{SD} = W_1 \cup W_2 \cup W_3$, the set of edges, is defined as follows. Each edge in W_{SD} is a pair consisting of a unique identifier and the integer valued bandwidth of the edge.

$W_1 = \{(1, e_j^{1,k}) \mid 1 \leq j \leq 4,\ 1 \leq k \leq n/2$ and

$$e_1^{1,k} = (v^{0,2k-1}, v_1^{1,k}),\ e_2^{1,k} = (v^{0,2k}, v_2^{1,k}),$$

$$e_3^{1,k} = (v^{0,2k-1}, v_2^{1,k}),\ e_4^{1,k} = (v^{0,2k}, v_1^{1,k})\}.$$

$W_2 = \{(\lceil \log(K+1) \rceil, e_j^{1,k}) \mid 1 \leq j \leq 8,\ 2 \leq i \leq \log_2 n - 1,\ 1 \leq k \leq \frac{n}{2^i}$ and

$$e^{i}_{1},^{k} = (v^{i}_{1}{}^{-1,2k-1}, v^{i}_{1},^{k}),\ e^{i}_{2},^{k} = (v^{i}_{2}{}^{-1,2k}, v^{i}_{2},^{k}),$$

$$e^{i}_{3},^{k} = (v^{i}_{2}{}^{-1,2k-1}, v^{i}_{4},^{k}),\ e^{i}_{4},^{k} = (v^{i}_{1}{}^{-1,2k}, v^{i}_{3},^{k}),$$

$$e^{i}_{5},^{k} = (v^{i}_{2}{}^{-1,2k-1}, v^{i}_{3},^{k}),\ e^{i}_{6},^{k} = (v^{i}_{1}{}^{-1,2k}, v^{i}_{4},^{k}),$$

$$e^{i}_{7},^{k} = (v^{i}_{3},^{k}, v^{i}_{1},^{k}), \qquad e^{i}_{8},^{k} = (v^{i}_{4},^{k}, v^{i}_{2},^{k})\}.$$

$W_3 = \{ (\lceil \log(K+1) \rceil, e^{\lfloor \log n, 1}), (\lceil \log(K+1) \rceil, e^{\lfloor \log n, 1}_{2}), (\lceil \log(K+1) \rceil + 1, e^{\lfloor \log n + 1, 1}) \text{ where}$

$$e^{\lfloor \log n, 1} = (v^{\lfloor \log n - 1, 1}_{2}, v^{\lfloor \log n, 1}),\ e^{\lfloor \log n, 1}_{2} = (v^{\lfloor \log n - 1, 2}, v^{\lfloor \log n, 1}),$$

$$e^{\lfloor \log n + 1, 1} = (v^{\lfloor \log n, 1}, v^{\lfloor \log n + 1, 1}) \}.$$

The indices i, j and k are used to label the nodes and edges of $K\text{–}Thr$ uniquely. The subscript j distinguishes between types of nodes and edges, and superscripts i and k distinguish within a type. In particular, i indexes $K\text{–}Thr$ along a vertical axis, increasing from 0 at the inputs along the bottom to $\log n + 1$ (ie. depth of $K\text{–}Thr$) at the top. i is thus called a *vertical index*. k indexes $K\text{–}Thr$ along a horizontal axis, increasing from left to right, and is called a *horizontal index*.

Let F_n be the function realized by circuit $K\text{–}Thr(n)$. Let $X \in \{0,1\}^n$.

$F_n : \{0,1\}^n \rightarrow \{0,1\}$ such that $F_n(X) = v^{\lfloor \log n + 1, 1}$

The reader can verify that $v^{\lfloor \log n + 1, 1} = T_K(x_1, \ldots, x_n)$

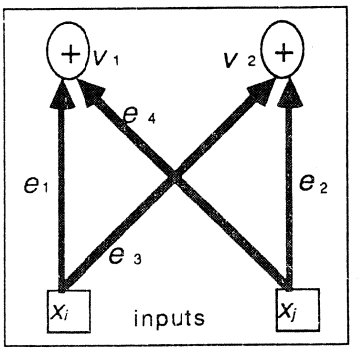
$F_n(x_1, \ldots, x_n)$ is abbreviated as $F_n(X)$ or $F(n)$.

Let $LF(x_1, \ldots, x_n)$ be an embedding of $K\text{–}Thr(n)$, as illustrated in Figures 3.0, 3.1 and 3.2. Recall that an embedding is a mapping of a circuit into the plane. Typically, circuits are embedded into regular structures like the grid shown in Figure 3.3b. To account for the cables in $K\text{–}Thr(n)$, the circuit is embedded into a *fat* grid like that shown in Figure 3.3a. To simplify the presentation of LF, the bandwidth of wires is not shown in Figures 3.0, 3.1 and 3.2. In particular, each wire or cable is illustrated as a single edge. In the complexity analysis of LF below, constant bandwidths are shown to increase resource use by only a constant factor; hence they can be neglected.

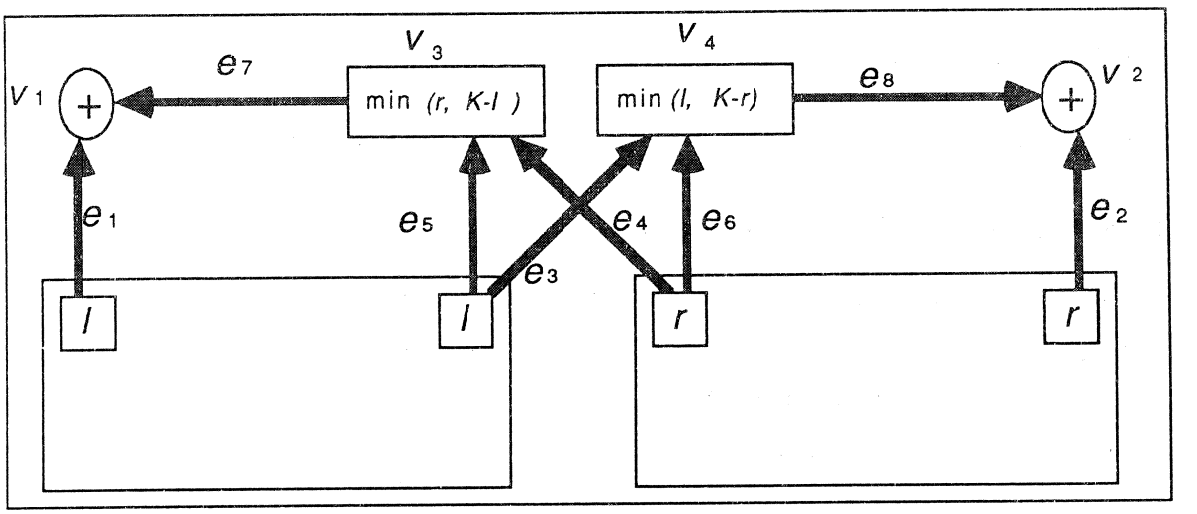
Note that in Figure 3.1, each min-node is shown to incorporate two operations - a subtraction followed by the min function. It is easy to see that this does not affect the asymptotic complexity of the circuit. Hence for simplicity of presentation, the two operations are combined into one node. This was also possible in Figure 3.2 for the final addition and comparison. Hence they too could be abbreviated into one node, but are also easily illustrated as two nodes.

Also note from Figures 3.1 and 3.2 that the value of K, the threshold bound, is used at many nodes of the circuit. We assume that K is "hardwired" into the circuit, using a small amount (eg. $\lceil \log(K+1) \rceil$) of memory at each node where it is used. It is easy to see that this strategy uses $O(n)$ area and energy since a constant amount of memory is needed at $O(n)$ nodes in the circuit.

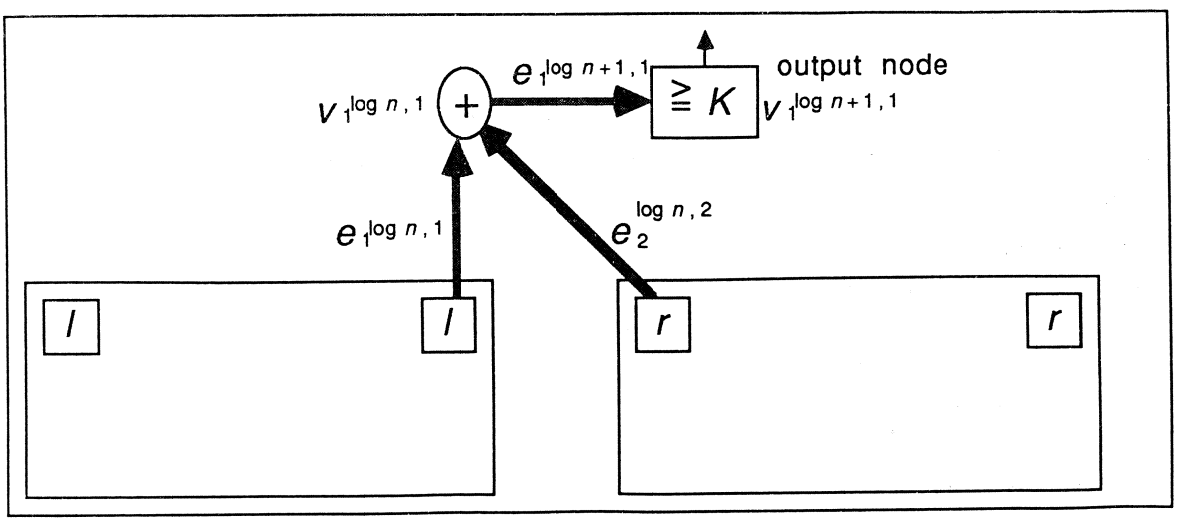
The following specifies layout parameters of $LF(x_1, x_2, \ldots, x_n)$. Input nodes are separated by distance $2\lambda\lceil \log(K+1) \rceil$ on a line. The relative location of interior nodes of $K\text{–}Thr(n)$ in the layout is evident from the recursive description, illustrated in Figures 3.0, 3.1 and 3.2. Wire (cable) lengths are as follows.



Bottom Layer of an Embedding of Circuit $K-Thr(n)$
Figure 3.0.



Middle Layers of an Embedding of Circuit $K-Thr(n)$
Figure 3.1.



Top Layer of an Embedding of Circuit $K-Thr(n)$
Figure 3.2.

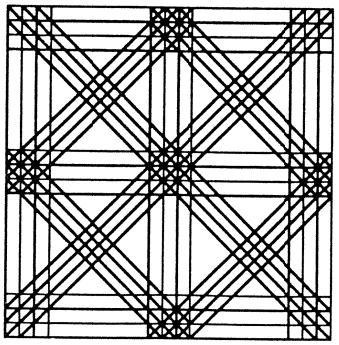


Figure 3.3a. $GD^{4}_{2,2}$

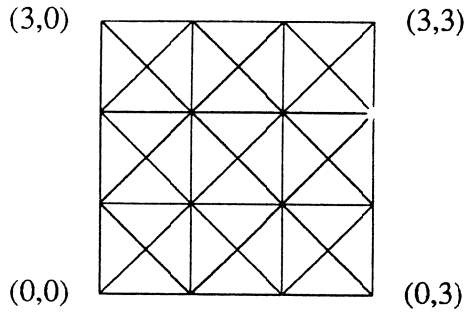


Figure 3.3b. $GD_{3,3}$

$\| e^{1,k}_{1} \| = \| e^{1,k}_{2} \| = 2, \| e^{1,k}_{3} \| = \| e^{1,k}_{4} \| = 2\sqrt{2}$ for $1 \le k \le n/2$;

$\| e^{i,k}_{1} \| = \| e^{i,k}_{2} \| = \| e^{i,k}_{5} \| = \| e^{i,k}_{6} \| = 1, \| e^{i,k}_{3} \| = \| e^{i,k}_{4} \| = \sqrt{2}$, and

$\| e^{i,k}_{7} \| = \| e^{i,k}_{8} \| = (2^{i} - 2)\lceil \log(K+1) \rceil$ for $2 \le i \le \log n - 1$, $1 \le k \le \frac{n}{2^{i}}$;

$\| e^{\log n,1}_{1} \| = \| e^{\log n+1,1}_{1} \| = 2, \| e^{\log n,2}_{2} \| = 2\sqrt{2}$.

$LF(x_1, \ldots, x_n)$ is abbreviated by $LF(n)$ or LF. Some facts about $LF(n)$ and $K\text{–}Thr(n)$:

1) height($LF(n)$) $\le c\lceil \log(K+1) \rceil$ + height($LF(\frac{n}{2})$) for some constant $c > 0$
   $= O(\log_2 n \ (\log_2(K+1)))$

2) area($LF(n)$) = height($LF(n)$) × width($LF(n)$)
   $\le c \ (\log_2 n \times \log_2(K+1)) \times (2n-1)\lceil \log(K+1) \rceil$ for some constant $c > 0$
   $= O(n \log_2 n \ (\log_2(K+1)))$

3) Let $D(n)$ be the depth of the $K\text{–}Thr(n)$ circuit.
   $D(n) \le 2d + D(\frac{n}{2})$ where $d > 0$ is the maximum depth of a node in $K\text{–}Thr(n)$.
   $= O(\log_2 n)$

Theorem 3.1:

For all pairs of legal states, the worst case uniswitch energy consumed by embedding $LF(n)$ of threshold circuit $K\text{–}Thr(n)$ is $E_{worst}(LF(n)) = O(n \times K \log^2(K+1))$, where $K\text{–}Thr(n)$ has depth $O(\log n)$.

Proof Sketch:

The proof proceeds as follows:

(i) Show that the "short" wires use only linear area and hence linear uniswitch energy. In the $K\text{–}Thr$ circuit, nodes are not always minimum area, but can also be shown to occupy linear area and linear energy, when K is constant.

(ii) Show that only constant area of "long" wires switch for any pair of inputs, using at most $O(n)$ uniswitch energy, for K a constant.

(i) The *short wires* in $K\text{–}Thr$ are $\{ e^{i,k}_{j} \mid 1 \le j \le 6, 1 \le i \le \log n, 1 \le k \le n/2^{i} \}$, which are each $\le 2\sqrt{2}\lambda$ long. Most of these short wires are cables containing up to $\lceil \log(K+1) \rceil$ individual wires. Hence, the width of a cable is $\le 2\lambda \lceil \log(K+1) \rceil - 1$. Clearly the size (ie. number of nodes) of

K–*Thr* (n) is $O(n)$ and the total number of wires is $O(n)$. Since each short wire uses constant area, the total area contributed by the short wires is $O(n)$. Hence, the total uniswitch energy consumed by the short wires is $O(n)$.

Consider the node energy consumed by $LF(n)$. Nodes $\{v_1^{i,k}, v_2^{i,k} \mid 1 \le i \le \log_2 n, 1 \le k \le n/2^i\}$ each compute addition of two binary numbers. The inputs to a +-node may be two cables, each of width $\le \lceil \log(K+1) \rceil$. The area of a +-node must be at least large enough to accomodate its inputs. Further, fast addition of 2 m-bit numbers is known to use $\Theta(m \log m)$ worst case uniswitch energy, when the inputs are laid out as in $LF(n)$. Hence, the energy used by a +-node is $O(\log K \operatorname{loglog} K)$. Since there are $O(n)$ +-nodes in K–*Thr* (n), the total energy contributed by these nodes is $O(n \log K \operatorname{loglog} K)$, which is linear when K is constant. Similarly, the min-nodes (eg. v_3 and v_4) and the compare node (eg. $v_{\lceil \log n \rceil + 1}$) contribute at most linear energy when K is constant.

(ii) The *long wires* of $LF(n)$ are $\{e_7^{i,k}, e_8^{i,k} \mid 1 \le i \le \log_2 n, 1 \le k \le n/2^i\}$ Each long wire is a cable that is $\le 2^i \lambda \lceil \log(K+1) \rceil$ long and $\le 2\lambda \lceil \log(K+1) \rceil$ wide. Since there are $O(n)$ long wires, the total area contributed by these wires is $O(n \log n \log^2(K+1))$. However, the uniswitch energy used by these long wires is asymptotically less, by the following argument.

In the embedded *SOR/SAND* circuit [Ki91], when a long wire L switches, none of the long wires "underneath" L switch. In the embedded K–*Thr* circuit, at most K long wires "underneath each other" can switch. To see this, consider the $\{e_7\}$ long wires. (The $\{e_8\}$ long wires follow by the same argument.) At the head of each e_7 wire is +-node v_1. It is not difficult to see that each v_1 node sums up to K. As soon as node $v_1^{i,k}$ sums to K, no additional e_7 wires "above" node $v_1^{i,k}$ can become active, ie. take on non-zero values, because min-nodes at the tails of these long wires "kill" any incoming non-zero values. However, to arrive at a sum of up to K, up to K long wires "underneath each other" can be used to propagate the K inputs. These long wires are cables of width $O(\log(K+1))$. Hence, the area of long wires that may switch on is $O(n \times K \log^2(K+1))$. In particular, for constant values of K, the worst case uniswitch energy needed to compute a K-threshold function quickly is $\Theta(n)$. []

Theorem 3.1 yields the following bounds for *counting* and *majority* functions.

Corollary 3.1:
For K a constant, to *count to* K in $O(\log n)$ depth, the worst case uniswitch energy is $O(n)$.

Proof:
Immediate from the construction above of $LF(n)$. Clearly, the v_1 and v_2 nodes at level $\log n - 1$ in $LF(n)$ (Figure 3.1) each count to K.

Corollary 3.2:
The *majority* function can be computed using worst case uniswitch energy $O(n \log^2 n)$.

4. CONCLUSIONS

This paper describes a construction that yields fast, minimum energy VLSI circuits to compute threshold functions and to count up to a constant. This same construction yields VLSI circuits for majority and general counting functions, but it is an open question whether the energy bounds for these functions are optimal.

ACKNOWLEDGEMENTS:

We thank Evangelos Kranakis, Danny Krizanc and Paul Vitanyi for many helpful discussions.

REFERENCES:

[ACR88]

Aggarwal, A., A. Chandra, P. Raghavan, "Energy Consumption in VLSI Circuits", *Proceedings of 20th ACM STOC,* May 1988, pp. 205-216.

[Bo77]

Borodin, A., "On Relating Time and Space to Size and Depth", *SIAM Journal of Computing,* Vol. 6, No. 4, December 1977, pp. 733-744.

[BK82]

Brent, R.P., H.T. Kung, "A Regular Layout for Parallel Adders", *IEEE Transactions on Computers,* Vol. C-31, No. 3, March 1982, pp. 260-264.

[BK81]

Brent, R.P., H.T. Kung, "The Area-Time Complexity of Binary Multiplication", *JACM,* Vol. 28, No. 3, July 1981, pp. 521-534.

[Ki91]

Kissin, G., "Upper and Lower Bounds on Switching Energy in VLSI", *JACM,* Vol. 38, No. 1, January 1991.

[Ki90]

Kissin, G., "Models of Multiswitch Energy", *CWI Quarterly,* Vol. 3, No. 1, March 1990, pp. 45-66.

[Ki87]

Kissin, G., "Modeling Energy Consumption in VLSI Circuits", *PhD Thesis,* Department of Computer Science, University of Toronto, 1987.

[Ki85]

Kissin, G., "Functional Bounds on Switching Energy", *Proceedings of 1985 Chapel Hill Conference on Very Large Scale Integration,* May 1985, pp. 181-196.

[Ki82]

Kissin, G., "Measuring Energy Consumption in VLSI Circuits: a Foundation", *Proceedings of 14th ACM STOC,* May 1982, pp. 99-104.

[KR65a]

Krohn, K., J. Rhodes, "Algebraic Theory of Machines. I. Prime Decomposition Theorem for Finite Semigroups and Machines", *Transactions of American Mathematical Society,* Vol. 116, 1965, pp. 450-464.

[KR65b]

Krohn, K., J. Rhodes, "Results on Finite Semigroups", *Proceedings of the National Academy of Science,* USA, Vol. 53, 1965, pp. 499-501.

[Le84]

Leo, J., "Energy Complexity in VLSI", *M.S. Thesis,* University of Nymequen, The Netherlands, February 1984.

[LM81]

Lengauer, T., K. Mehlhorn, "On the Complexity of VLSI Computations", *Proceedings of CMU Conference on VLSI,* Computer Science Press, October 1981, pp. 89-99.

[MC80]

Mead, C., L. Conway, *Introduction to VLSI Systems,* Addison-Wesley, 1980.

[Me86]

Mead, C., private communication

[ST86]

Snyder, L., A. Tyagi, "The Energy Complexity of Transitive Functions", *Proceedings of 24th Allerton Conference on Communication, Control and Computing,* October 1986, pp. 562-572.

[Th80]

Thompson, C., "A Complexity Theory for VLSI", *PhD Thesis,* Dept. of Computer Science, Carnegie-Mellon University, 1980.

[Ty87]

Tyagi, A., "Energy Complexity of Logic Level Structures," *Proceedings of Stanford Conference on Advanced Research in VLSI,"* MIT Press, 1987.

[Vu83]

Vuillemin, J., "A Combinatorial Limit to the Computing Power of VLSI Circuits", *IEEE Transactions on Computers,* Vol. C-30, No. 2, 1983, pp. 135-140.