

Models of Multiswitch Energy

Gloria Kissin

*Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands*

*Department of Mathematics and Computer Science
University of Amsterdam, Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands*

A technology independent framework is established for measuring the *switching energy* consumed by very large scale integrated (VLSI) circuits. Techniques are developed for analyzing functional energy consumption, and for designing energy-efficient VLSI circuits. A wire (or gate) in a circuit uses switching energy when it changes state from 1 to 0 or vice versa. This paper[†] develops the *Multiswitch Models (MSM)* of energy consumption, which trace a circuit through the time it changes from one stable state to another. In particular, *MSMs* account for *race conditions*, whereby a node can change states several times before settling down to a final value. Several *MSMs* are defined to account for various physical factors that can cause delays in a computation. Gate delays, wire delays, input delays, and circuit synchrony are considered. An *MSM* is viewed as *optimistic* if race conditions occur only when a circuit is *asynchronous*. Otherwise, *MSM* is considered to be *pessimistic*. This paper demonstrates that race conditions can be very expensive in terms of switching energy. In particular, the *multiswitch energy* of a VLSI circuit can be asymptotically larger than the circuit's *area*. However, the multiswitch energy of a circuit is shown to be bounded. The *parity* function is shown to use $\Theta(n^2)$ energy in a pessimistic *MSM*. Novel circuits and layouts are obtained for n -bit *OR* and compare functions that have shallow depth and use only linear energy, in an optimistic *MSM*.

1. INTRODUCTION

1.1. Motivation

The advancement of technology to very large scale integration (VLSI) has brought with it a number of issues not accounted for by such traditional models of computation as Turing Machines [9], Random Access Machines (RAM) [9], and combinational circuits [3]. This paper addresses the issue of *energy* consumption in VLSI circuits. A circuit in any practical technology requires energy for its operation. Thus, energy consumption is believed to be intrinsic to computation [19], and the complexity of a finite VLSI computation appears to be closely related to the energy it consumes.

Further, the energy consumed by a circuit is transformed into heat. How

[†] Preparation of this paper was partially completed at CSRI, University of Toronto.

well a circuit can dissipate heat determines its operational limitations. In VLSI where thousands of transistors are packed onto a chip of 1 cm^2 , heat can be produced faster than it can be dissipated. This can result in chip failure. The amount of heat dissipated by a circuit is proportional to the energy consumed. Hence, as packing densities increase, it becomes increasingly important to reduce the amount of heat produced, by designing energy-efficient circuits.

An additional concern is the indirect systems cost to drive circuits. Although a single chip at current densities consumes less than one watt of power, assembling large systems of chips can be expensive, since much of the support structure in a computer serves to power and cool the circuitry [20].

1.2. Switching energy

Common to all physical devices is the *switching energy* consumed when a wire or gate changes state from 1 to 0 or vice versa [19]. The amount of switching energy consumed in a finite computation is proportional to the area switched. The intent of this paper is to lay the groundwork for measuring the switching energy consumed in VLSI circuits.

Many technologies consume more than switching energy. All MOS (*Metal Oxide Silicon*) devices except CMOS (*Complementary MOS*) also dissipate steady state DC power, as do bipolar silicon devices and Gallium Arsenide logic [30]. Devices that consume DC power contain ‘pull-up’ transistors that use energy even when the device is in a steady ‘0’ state. In these technologies, the DC power dominates the energy consumption. CMOS and some Josephson Junction (JJ) logics (e.g. JJ-Current Steering logic) consume only switching energy [19]. Hence the results in this paper are a precise model for CMOS and JJ-CS and a lower bound on the total energy used by other technologies.

1.3. Related work

The *Multiswitch Models (MSM)*, defined in the next section, first appeared in [14]. Much of this paper also comprises part of [12]. [14] also introduced the *Uniswitch Model (USM)* of energy consumption, which is simpler than *MSM* and better for lower bounds. But *USM* may be optimistic for studying upper bounds, as the model neglects race conditions. Much of the literature in switching energy uses the *Uniswitch Model* [2,11,12,15,16,17,28]. [2] also obtained bounds in *MSM*, which will be cited in the body of this paper.

The rest of this paper is organized as follows. Section 2 defines the *Multiswitch Models* of energy consumption. Section 3 demonstrates that race conditions can be very expensive in terms of energy. To illustrate this expense, an n -input VLSI circuit that uses $O(n \log n)$ area is shown to consume $\Theta(n^2)$ multiswitch energy. However, tree circuits with all \wedge -gates or all \vee -gates are shown to use a quantity of multiswitch energy that is proportional to the tree’s area, in the worst case. The *parity* function is shown to use $\Theta(n^2)$ multiswitch energy. In Section 4, a VLSI circuit that realizes the *OR* function is shown to be energy-efficient in an *MSM* that is not too pessimistic.

Throughout the paper we will make use of the following standard notation. $f(n) = O(g(n))$ (respectively, $f(n) = \Omega(g(n))$) if and only if there is a positive constant c such that $f(n) \leq c \cdot g(n)$ (respectively, $f(n) \geq c \cdot g(n)$) and $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

2. THE SETTING

2.1. Introduction

The *Multiswitch Models* of energy consumption define a set of energy cost measures for VLSI circuits, which account for *race conditions* (also known as *hazards*). *MSM* traces a circuit's switching behaviour as it changes from one stable state to another. This contrasts with the *Uniswitch Model* [11], which measures the differences between two stable states. *MSMs* account for the intermediate switching that can occur during a state transition. Henceforth, the term *Multiswitch Model of energy consumption* is usually abbreviated by the acronym *MSM* or by the term *multiswitch energy*.

Consider the following informal description of how a circuit operates. Let C be a circuit with initial state s_0 and final state s_f . Assume that C is in state s_0 at time t_0 , and in state s_f at time t_f , for $t_f > t_0$. From time to time between t_0 and t_f , input ports may change their values. Assume that an input port switches at most once between the initial and final states of C . Once an input changes, a value α will appear at node v of C at certain times until t_f . When α appears as an input to node v , v computes an output, possibly with some delay. The output is then transmitted along v 's out edges, possibly with some delay.

It is apparent from the above description that a 2-input node of a circuit can receive its inputs at different times. This timing difference can cause *race conditions*, whereby a node can change state several times before settling down to a final value. *MSM* traces a circuit through the time it changes from one *stable* state to another. This dynamic process can involve *metastable* states, whereby a node's value is transiently inconsistent with the functional labeling of the node and its inputs. Hence, a node (and wire) might switch several times before reaching a new stable state. *MSM* accounts for this intermediate switching.

The following informally defines the timing issues that are addressed by *MSM*.

DEFINITIONS. Let C be a VLSI circuit. Let v be a node in C . Let p be a path in C . Let w be a wire in C . The *depth* of p is the number of edges in p . The *area* of w is the length of w in the plane. C is *synchronous* if all input paths to v are equal in depth; otherwise, C is *asynchronous*.

DEFINITIONS. Let C be a VLSI circuit. C has *wire delay* Δ if, for any wire w in C of area k , $\Delta(k)$ is the time to transmit a bit from the tail of w to its head. Wire delay is related to signal propagation delay in real circuits. C has *gate delay* δ if $\delta(v)$ is the time taken by non-input node v to compute an output.

Gate delay allows for variable node switching speeds. C has *input delay* I if input node x acquires a value at time $I(x)$. Input delay is motivated by the physical assumption that input arrival times may vary in real circuits.

Circuit synchrony, wire delay, gate delay, and input delay are the timing issues accounted for by the *Multiswitch Models* defined in the next section. As well as accounting for timing delays, *MSMs* can also model the multiswitch behaviour of sequential circuits (i.e. with feedback) and pipelined circuits. This paper, however, is primarily concerned with the multiswitch characteristics that result from race conditions.

2.2. Preliminaries

The following establishes a framework for the definitions of the *Multiswitch Models*.

DEFINITION [3,5]. A *VLSI circuit* is a combinational circuit [3] embedded in a plane as in [5]. Salient assumptions of the Borodin/Brent/Kung models that are important to *MSM* are as follows. A circuit is acyclic. A wire (edge) in a VLSI circuit has constant minimum width $\lambda > 0$. A non-input node (gate) in a circuit computes a logical function of 1 or 2 inputs (e.g. AND (\wedge), OR (\vee), NOT (\neg)). Gates are separated by distances $\geq \lambda$. A gate has area $\geq \lambda^2$, and a non-output gate has fanout 1 or 2. Input nodes have fanin 0. Output nodes have fanout 0. At most a constant number of wires, $v \geq 2$, can overlap or intersect at any point in a VLSI circuit.

Some of the energy lower bounds described in this paper are functions of the wire area of a VLSI circuit. These results require that the circuit in question be connected, that the circuit inputs preclude constant values, and that interior nodes are functionally dependent on the inputs (i.e. each interior node has two states).

DEFINITION. A *CID VLSI circuit* is a VLSI circuit that satisfies the following two properties.

- E1: Each input has fanout at most p , for $p \geq 1$. This limits the number of *free* duplicates of any input bit. Presumably, a real circuit receives one or a few instances of an input bit, and if more are needed the input must be replicated by the circuit. This costs area and energy that cannot realistically be neglected. In Section 2.3, an example is given that illustrates the asymptotic affect of replicating inputs.
- E2: All instances of any input appear at input nodes that are within a constant distance of each other. This again is an input fanout constraint. In this paper, input and output nodes are sometimes assumed to be on a convex boundary of the circuit layout.

Properties E1 and E2 are called *constant input duplicates (CID) assumptions*. This paper is primarily concerned with CID VLSI circuits and their physical

analog. Hence, the term *circuit* generally refers to a CID VLSI circuit. The terms *real circuit* or *physical circuit* refer to the physical analog of a CID VLSI circuit. (In [12], CID circuits are called CIF circuits.)

DEFINITION. 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 , i.e. $C = (V, W)$ where V is the node set and W is the set of wires. $s: V \cup W \rightarrow \{0,1\}$. 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 input and the labeling of the nodes (e.g. $s(\text{AND}(0,1))=0$, $s(\text{NOT}(0))=1$). 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 .

DEFINITIONS. Assume that at time t_0 , wire w has initial value w_0 , and at time $t_1 > t_0$, w has value w_1 , where $w_0, w_1 \in \{0,1\}$. Then w is *switched* (*switches*) iff $w_0 \neq w_1$. A wire of area L that switches accounts for L/k *switching energy*, where $k > 0$ is the area of wire that accounts for one unit of switching energy.

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 CID 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*.

MSMs are related to the *multiswitch property*, whereby an interior gate in a circuit may switch more than once when the circuit changes from one input setting to another.

2.3. Motivating CID circuits

This section discusses the following assumptions about the inputs to a VLSI circuit.

- 1) Each input has at most constant fanout.
- 2) All instances of an input appear within constant distance of each other.

These two constraints are called *constant-input-duplicates (CID)* assumptions. It is reasonable to expect that a real VLSI circuit will receive a single copy of any input, or at most a small number of copies. It is then up to the circuit to replicate the input as required and to transmit copies to distant locations on

the chip as required. The following gives an example of the energy costs of replicating and transmitting input. The example shows that since these costs can have an asymptotic effect on the total energy, they cannot be neglected in practice. The constant-input-duplicates assumptions thus force the circuit designer to account for the cost of replicating and transmitting input.

Let $A = (a_0, \dots, a_{n-1})$ and $B = (b_0, \dots, b_{n-1})$ be sets of boolean variables. Consider the following DNF expression:

$$\begin{aligned} H(A, B) = & (a_{n-1} \wedge b_{n-1}) \vee \\ & [(a_{n-1} \oplus b_{n-1}) \wedge (a_{n-2} \wedge b_{n-2})] \vee \\ & \vdots \vee \\ & [(a_{n-1} \oplus b_{n-1}) \wedge \dots \wedge (a_1 \oplus b_1) \wedge (a_0 \wedge b_0)] \end{aligned}$$

where $a_i, b_i \in \{0, 1\}$, $0 \leq i < n$.

Let $\hat{A} = \sum_{i=0}^{n-1} a_i 2^i$ and $\hat{B} = \sum_{i=0}^{n-1} b_i 2^i$. $H(A, B)$ computes the $n+1$ st bit of $\hat{A} + \hat{B}$.

Consider a circuit C_n for H in which each occurrence of a_i and b_i in $H(A, B)$ is provided at a distinct input node, and interior nodes have fanout at most one. Then C_n is a tree with $O(n^2)$ leaves. Brent and Kung [6] and Yao [32] showed that such a tree requires area $\Omega(n^2 \log n)$ when it has $O(\log n)$ depth and the leaves are on a convex boundary of the layout. Thus, an optimal embedding of C_n in the *Uniswitch Model* will use no more than $O(n^2 \log n)$ energy, which is consumed when H switches from a state where $A = B = 0^n$ to a state where $A = 1^n$ and $B = 0^{n-1} 1$.

The energy cost of H can be improved to $O(n^2)$ by using a nontree-like circuit D_n in which the conjunctive clauses of H are energy-efficient (i.e. use at most $O(n)$ energy per clause). The technique for realizing such a circuit is discussed in Section 5. As in C_n , each occurrence of a_i and b_i in $H(A, B)$ is provided at a distinct input node in D_n .

The analysis above charges only unit energy per input instance, although many inputs have $O(n)$ instances and instances can be up to $O(n^2)$ distance apart when the input nodes are laid out as in Figure 2.1. Consider an analysis of H that realistically accounts for these factors.

Figure 2.1 illustrates a circuit \hat{D}_n for H that includes the input fanout trees. The example in the figure uses $n=4$. Recall that F denotes a fanout node that computes the identity function of its input. Consider the area of \hat{D}_n .

More than half the $2n$ inputs each have at least $n/2$ instances in H . Hence the area of each of these input fanout trees in \hat{D}_n is $\Omega(n \log n)$, even when the large separation between instances is ignored. When an input switches the entire fanout tree switches. Hence the total energy for the input fanout trees is $\Omega(n^2 \log n)$, which exceeds the $O(n^2)$ energy cost of the non-input portion of \hat{D}_n (assuming \hat{D}_n uses energy-efficient conjunctive clauses).

If we realistically assume that an input will arrive at a single input port, then the large separation between input instances in D_n will be manifested by long fanout wires in \hat{D}_n . In fact, $\Omega(n)$ inputs in \hat{D}_n have instances that are

$\Omega(n^2)$ apart. This drives the energy cost to at least $\Omega(n^3)$.

In [11], H is shown to be computable in $O(n)$ energy in USM .

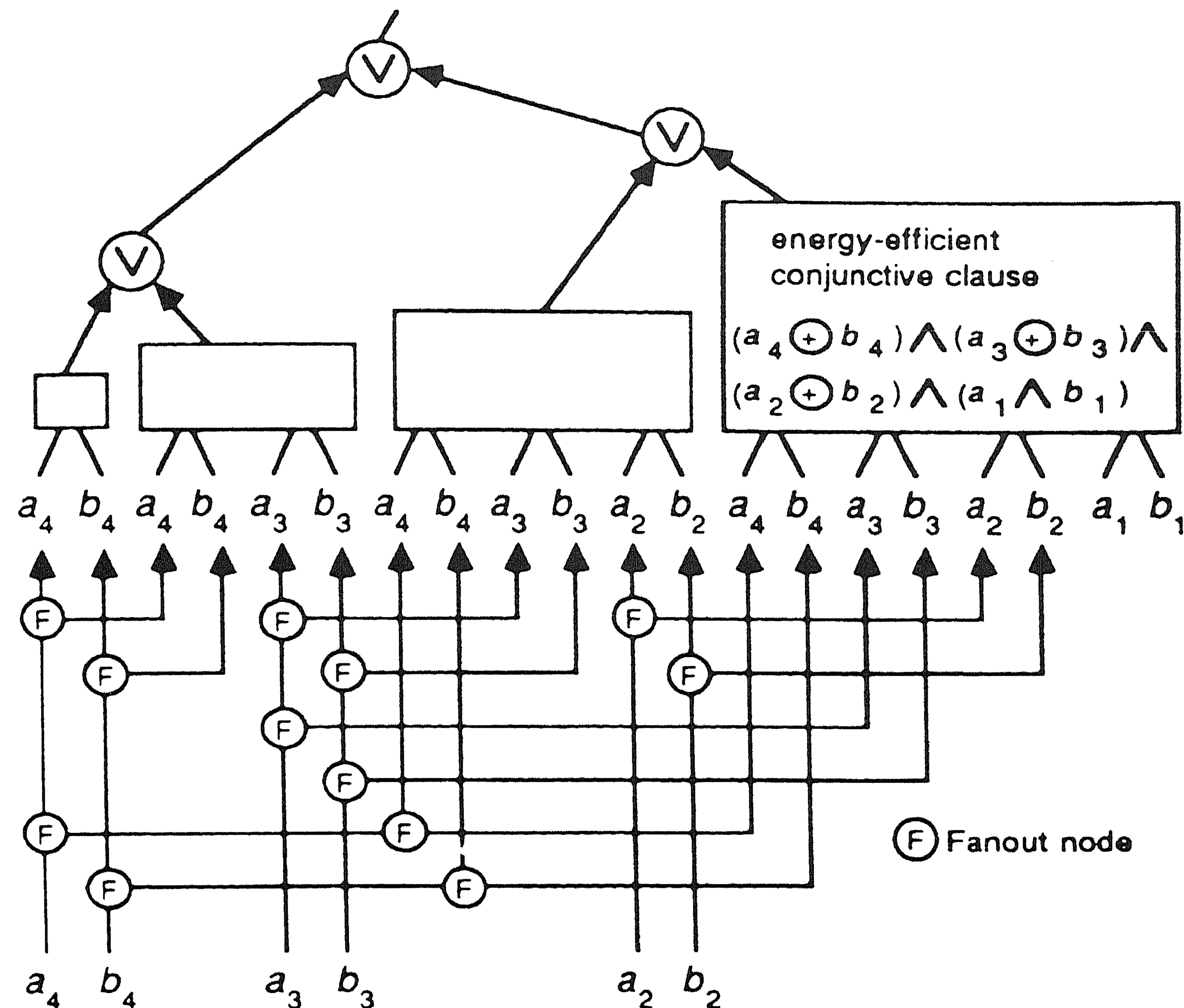


FIGURE 2.1. \hat{D}_4 , a VLSI circuit with large input fanout

3. THE MODEL DEFINITIONS

In this section, several MSM s are defined that are related to physical assumptions about the timing of a real circuit. The timing issues discussed are circuit synchrony, gate delay, wire delay and input delay, which were informally defined in Section 2.1. In the following discussion, the physical timing assumptions that are the basis of each MSM defined are first given. Each MSM is then defined in terms of particular delay functions and the relevant circuit notions.

Physical assumptions for M_1

- 1) The time for a bit to travel along a wire is independent of the length of the wire. A real circuit with this property has *constant wire delay*.
- 2) All inputs arrive at the input ports at the same time. A real circuit with this property has *constant input delay*.
- 3) Each non-input node computes an output at the instant that its inputs arrive. A real circuit with this property has *zero gate delay*.

For a real circuit that has zero gate delay, constant wire delay and constant input delay, race conditions can arise when the circuit is asynchronous; that is, some paths to some node of the circuit have different depth.

The constant wire delay assumption originated in the thesis of C. Thompson [29] and the papers of Brent and Kung [4,5], and was subsequently adopted by many VLSI theory researchers—[1,18,23,25,31] to name a few. However, the validity of the constant wire delay model has been subject to debate within the research community [7,22,24], with models emerging that define time as variously a logarithmic [21], linear [8,10], or a quadratic [27] function of wire length. The only consensus seems to be that the time for a bit to travel along a wire is a monotonic nondecreasing function of the wire's length. The constant wire delay model may be applicable to some specific circuits.

Physical assumptions for M_2

- 1) The time for a bit to travel along a wire is a monotonic increasing function of the wire's length.
- 2) Constant input delay.
- 3) Zero gate delay.

Race conditions in M_2 can arise if a circuit is asynchronous, or if a node of a circuit has input wires of different lengths.

Physical assumptions for M_3

- 1) The time for a bit to travel along a wire is a monotonic increasing function of the wire's length.
- 2) Inputs may arrive at the input ports at different times.
- 3) Zero gate delay.

Like M_1 and M_2 , race conditions in M_3 can arise if a circuit is asynchronous. In addition, varying wire delays and varying arrival times of the primary inputs can cause race conditions if an interior node receives its inputs at different times.

Physical assumptions for M_4

- 1) The time for a bit to travel along a wire is a monotonic increasing function of the wire's length.
- 2) Inputs may arrive at the input ports at different times.
- 3) Non-input gates have varying switching speeds.

Moving from M_1 to M_2 relaxes the physical requirement that signal propagation times be constant, and moving from M_2 to M_3 relaxes the requirement that primary inputs be synchronized. Moving from M_3 to M_4 relaxes the requirement that gates have constant (zero) delay.

Recall from Section 2.1 that circuit $C = (V, W)$ has *wire delay* Δ if, for w a wire in C , $\Delta(\text{area}(w))$ is the time to transmit a bit from the tail of w to the head of w . If $\forall w \in W$, $\Delta(\text{area}(w)) = 1$ (or some other positive constant), then C has *constant wire delay*, which is denoted as Δ_1 . C has *gate delay* δ if $\delta(v)$ is the time taken by non-input node v to compute an output. If $\forall v \in V$, $\delta(v) = 0$, then C has *zero gate delay*, which is denoted as δ_0 .

DEFINITION. Let C be a circuit and let $Z = \{z_1, \dots, z_n\}$ be the input nodes of C . Let $\alpha = \{1, \dots, n\}$ be the index set for the input to C . An *input schedule* I is a partial function $I: Z \times \alpha \rightarrow \mathbb{R}$ such that for $1 \leq i \leq n$, the i -th input arrives at input node z_i at time $I(z_i, i)$. Let $C: s \rightarrow_I X$ denote that circuit C switches from state s to state s_X according to input schedule I . I_0 is an input schedule such that for $1 \leq i \leq n$, the i -th input arrives at input node z_i at time 0 (or some other non-negative constant). If $C: s_0 \rightarrow_{I_0} X$, then C has *constant input delay*. Note that the definition of an input schedule assumes that each input bit arrives at a distinct input port.

DEFINITION. Suppose $W = \{w_i\}$ is the set of wires in CID VLSI circuit C , X is the input vector, Δ is the wire delay function, δ is the gate delay function, and I is the input schedule. If wire w_i switches z_i times when $C: s_0 \rightarrow_I X$, then the *wire energy*, E_w , consumed by C is $E_w(C, s_0, X, \Delta, \delta, I) \triangleq (1/k) \sum_{i=1}^{|W|} z_i \star (\|w_i\|)$, where $|W|$ is the cardinality of set W and $\|w_i\|$ is the area of wire w_i .

DEFINITIONS. If C_n is a CID 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 $E_w(C_n, s_0, X, \Delta, \delta, I)$ is the wire energy consumed by f_n when $X = (\alpha_1, \dots, \alpha_n)$ is the input to C_n after time t_0 according to input schedule I , then $E_{worst}^{M_1}(C_n)$, the M_1 *worst case multiswitch energy*, is given by

$$E_{worst}^{M_1}(C_n) \stackrel{\Delta}{=} \max_{(s_0, X)} E_w(C_n, s_0, X, \Delta, \delta, I_0).$$

$E_{worst}^{M_2}(C_n)$, the M_2 *worst case multiswitch energy*, is given by

$$E_{worst}^{M_2}(C_n) \stackrel{\Delta}{=} \max_{(s_0, X, \Delta)} E_w(C_n, s_0, X, \Delta, \delta, I_0).$$

$E_{worst}^{M_3}(C_n)$, the M_3 *worst case multiswitch energy*, is given by

$$E_{worst}^{M_3}(C_n) \stackrel{\Delta}{=} \max_{(s_0, X, \Delta, I)} E_w(C_n, s_0, X, \Delta, \delta, I).$$

$E_{worst}^{M_4}(C_n)$, the M_4 *worst case multiswitch energy*, is given by

$$E_{worst}^{M_4}(C_n) \stackrel{\Delta}{=} \max_{(s_0, X, \Delta, \delta, I)} E_w(C_n, s_0, X, \Delta, \delta, I).$$

Note that $E_{worst}^{M_1}(C_n)$ and $E_{worst}^{M_2}(C_n)$ are special cases of $E_{worst}^{M_3}(C_n)$, which is a special case of $E_{worst}^{M_4}(C_n)$. $E_{worst}^{M_i}(C_n)$ is abbreviated as $E_{worst}^{M_i}(n)$ or $E_{worst}^{M_i}$, for $1 \leq i \leq 4$. In the following discussion, when the MSM under consideration is clear from the context, the superscript M_i is omitted.

To see that there exists a maximum value for the multiswitch energy of a circuit, the following discussion demonstrates that the total amount of switching incurred by a circuit when it changes state is bounded. That the energy is bounded thus follows since circuits use finite area.

DEFINITION. Let W be the set of wires of VLSI circuit C_n . Let I be the input schedule of C_n . Suppose $\forall w_i \in W$, w_i switches r_i times when $C: s_0 \rightarrow_I X$. Let $Sw(C_n)$ denote the worst case switching of C_n .

$$Sw(C_n) \stackrel{\Delta}{=} \max_{(s_0, X, \Delta, \delta, I)} \sum_{i=1}^{|W|} r_i.$$

THEOREM 3.0. Let C_n be a VLSI circuit of depth $D(n)$ and size $S(n)$. $Sw(C_n) = O(2^{D(n)} * S(n))$.

PROOF. Let z be a node of C_n . Suppose the longest path from any input to z has depth $\leq d$. Let $B_z(d)$ denote the number of times z switches when $C_n: s_0 \rightarrow_I X$.

LEMMA 3.0. $B_z(d) = O(2^d)$.

PROOF. By induction on d .

Basis:

$d=0$. $\therefore C_n$ is the input node z . $B_z(d) \leq 1$ by definition of an input node.

Induction step:

Let u, v be the input nodes to z . The longest path from any input to node u or to node v has depth $\leq d-1$.

$$B_z(d) \leq B_u(d-1) + B_v(d-1).$$

If z has only one input, let $B_v(d-1) = 0$.

$$B_z(d) \leq 2 * \max(B_u(d-1), B_v(d-1)).$$

Without loss of generality, assume that $\max(B_u(d-1), B_v(d-1)) = B_u(d-1)$.

$$\therefore B_z(d) \leq 2 B_u(d-1).$$

By the induction hypothesis, $B_u(d-1) \leq c 2^{d-1}$.

$$\therefore B_z(d) \leq 2(c 2^{d-1}).$$

$$\leq c 2^d. \quad \square \text{ Lemma 3.0}$$

Since $d \leq D(n)$, any node of C_n switches at most $O(2^{D(n)})$ times. C_n has $S(n)$ interior nodes. $\therefore Sw(C_n) = O(2^{D(n)} * S(n))$. \square

The following definition of energy efficiency is analogous to the *USM* definition in [11]. Let $f \upharpoonright \{0,1\}^n$ denote a restriction of the family f of functions, such that the domain of the n -th function, f_n , is the set of boolean strings of length n . Let $\Phi(C)$ denote the family of functions computed by circuit family C .

DEFINITIONS. Let $1 \leq i \leq 4$. A function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is *energy efficient in M_i* iff \exists a family $C = (C_n)_{(n \in \mathbb{N})}$ of circuits with C_n realizing $f \upharpoonright \{0,1\}^n$, $E_{\text{worst}}^{M_i}(C_n) = \Theta(n)$. Circuit family $C = (C_n)_{(n \in \mathbb{N})}$ is *energy efficient in M_i* iff \forall families $\hat{C} = (\hat{C}_n)_{(n \in \mathbb{N})}$ of circuits with $\Phi(\hat{C}) = \Phi(C)$, $E_{\text{worst}}^{M_i}(\hat{C}_n) = \Omega(E_{\text{worst}}^{M_i}(C_n))$.

In this paper, n is a power of 2 and $\log n$ means $\log_2 n$.

4. WORST CASE UPPER BOUNDS

In this section, circuits are analyzed under the most pessimistic Multiswitch Models, M_3 and M_4 . The worst case energy consumption is obtained for complete binary tree circuits. In particular, circuits with all \wedge -gates or all \vee -gates are shown to use an amount of multiswitch energy that is proportional to the circuit's total area.

We describe a technique that induces a circuit to use asymptotically more multiswitch energy than the area of the circuit. This large energy consumption is obtained by using a 'bad' *input schedule*, which is defined below. This input schedule is then used to obtain an $\Omega(n^2)$ lower bound on the multiswitch energy required by the *parity* function on n inputs. (Parity is shown to use $\Theta(n^2)$ multiswitch energy.) Similarly, [2] used the bad input schedule to show that *addition* of two n -bit numbers requires $\Omega(n^2)$ multiswitch energy.

DEFINITIONS. Let $w: a \rightarrow b$ denote that wire or node w is switched (switches) from state a to state b , where $a, b \in \{0,1\}$. a is called the *initial state* of w , i.e. $s_0(w) = a$ or $s(w) = a$ at time t_0 . b is the *final state* of w , i.e. $s(w) = b$ at time t_f . w *settles down* at time t_f . Let $t(w: a \rightarrow b)$ denote that w switches from a to b at time t , where $t > t_0$, and $\forall t' < t, \neg t'(w: a \rightarrow b)$. Note that $t(w: a \rightarrow a)$ denotes that w does not switch at time t . $w: a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k$ denotes a *switching sequence* of w from initial state a_1 to final state a_k , where $a_i \in \{0,1\}$ for $1 \leq i \leq k$.

The following analysis considers tree circuits that consist of all \vee -gates or all \wedge -gates. These specialized circuits are shown to inhibit race conditions. Consequently, even in Multiswitch Model M_3 and M_4 , these tree circuits use energy proportional only to the area of the circuit, in the worst case.

Let a complete binary tree with n leaves (input nodes) and an \wedge -gate (\vee -gate) at each non-input node be called an *n -AND tree* (*n -OR tree*).

SWITCHING LEMMA 4.1. For an n -AND tree T , if each input of T switches at most once, each node (or edge) of T switches at most twice.

PROOF. Let z be a node of T . Let f_z denote the function computed at z . Let $s_X(z)$ denote the value of z for input vector X , and let $s_t(z)$ denote the value of z at time t .

LEMMA 4.1.1. $\forall X \in \{0,1\}^n$, the switching sequence $z: 1 \rightarrow 0 \rightarrow 1$ cannot be achieved in T .

PROOF. Induction on the depth d of T .

Basis:

$d=0$. z is an input node. z switches at most once by hypothesis.

Induction step:

Let v_L and v_R be the input nodes to z .

$\therefore f_z = f_{v_L} \wedge f_{v_R}$ by definition of T .

$s_0(z) = 1 \Rightarrow s_0(v_L) = 1$ and $s_0(v_R) = 1$.

$\exists t_k > t_0$ such that $t_k(z: 1 \rightarrow 0)$

$\Rightarrow \exists t_j$ such that $t_0 < t_j < t_k : t_j(v_L: 1 \rightarrow 0)$ and/or $t_j(v_R: 1 \rightarrow 0)$.

\Rightarrow by the induction hypothesis, $\forall t_l > t_j, s_{t_l}(v_L) = 0$ and/or $s_{t_l}(v_R) = 0$.

Since $f_z = f_{v_L} \wedge f_{v_R}$, z will not turn on again until v_L and/or v_R turns on.

End of proof of Lemma 4.1.1.

The result follows since the only state changes that any node z can make are $z: 0 \rightarrow 1 \rightarrow 0$, $z: 1 \rightarrow 0$, or $z: 0 \rightarrow 1$. \square

SWITCHING LEMMA 4.2. If each input node of an n -OR tree \hat{T} switches at most once, each node (or edge) of \hat{T} switches at most twice.

PROOF. Assume \hat{T} is in initial state \bar{s}_0 where s_0 is the initial state of n -AND tree T . The result follows from Switching Lemma 4.1 by the duality of AND and OR. The unachievable sequence for any edge in \hat{T} is '010'. \square

THEOREM 4.1. An n -AND (n -OR) tree T of area A consumes worst case energy $E_{\text{worst}}^{M_4}(T) = \Theta(A)$.

PROOF. By Switching Lemma 4.1 (4.2), no edge in T switches more than twice. Therefore, $E_{\text{worst}}(T) \leq 2 \star A$. Since T is a monotone circuit, $E_{\text{worst}}(T) \geq \frac{1}{2} \star A$ by a Theorem of [11]. \square

To see that $E_{\text{worst}}^{M_3} \gg \text{area}$ for some acyclic circuits, consider an n -OR/AND tree T defined as follows.

DEFINITIONS. Let $n = 2^k$ for $k \in \mathbb{Z}^+$. An n -OR/AND tree is an $n/2$ -AND tree with a 2-input \vee -gate at each of the $n/2$ leaves. An n -AND/OR tree is an $n/2$ -OR tree with a 2-input \wedge -gate at each of the $n/2$ leaves. An 8-OR/AND tree is illustrated in Figure 4.1.

Recall that M_3 accounts for circuits where inputs may arrive at the input nodes at different times. The following defines an input schedule whereby each input arrives 'much later' than the previous input. This input schedule is

then shown to cause wires of an n -OR/AND tree to switch many times before settling down to a final value.

DEFINITION. At time t_0 , let n -OR/AND tree T be in initial state s_0 such that input nodes (a_1, \dots, a_n) have initial value $(10)^{n/2}$. Let w_l denote a longest wire in T . *Input Schedule I_1* : Input vector $(01)^{n/2}$ arrives at the input nodes of T in the following manner. Input i arrives at leaf a_i at time t_i such that $t_1 > t_0$, and $t_{i+1} \geq t_i + (\text{depth}(T) * \Delta(\|w_l\|))$, for $1 < i < n$ (e.g. input 2 arrives at a_2 after input 1 arrives at a_1 and after T settles down). See Figure 4.1.

Let $Sw(k)$ denote the total number of switching events that occurs when a k -OR/AND tree T with k leaves changes state.

SWITCHING LEMMA 4.3. The total number of switching events exhibited by an n -OR/AND tree T with input schedule I_1 is $Sw(n) = \Theta(n \log n)$.

PROOF. The total switching of T is given by the following recurrence.

$$Sw(2) = 4$$

$$Sw(n) = 2Sw\left(\frac{n}{2}\right) + n = \Theta(n \log n). \quad \square$$

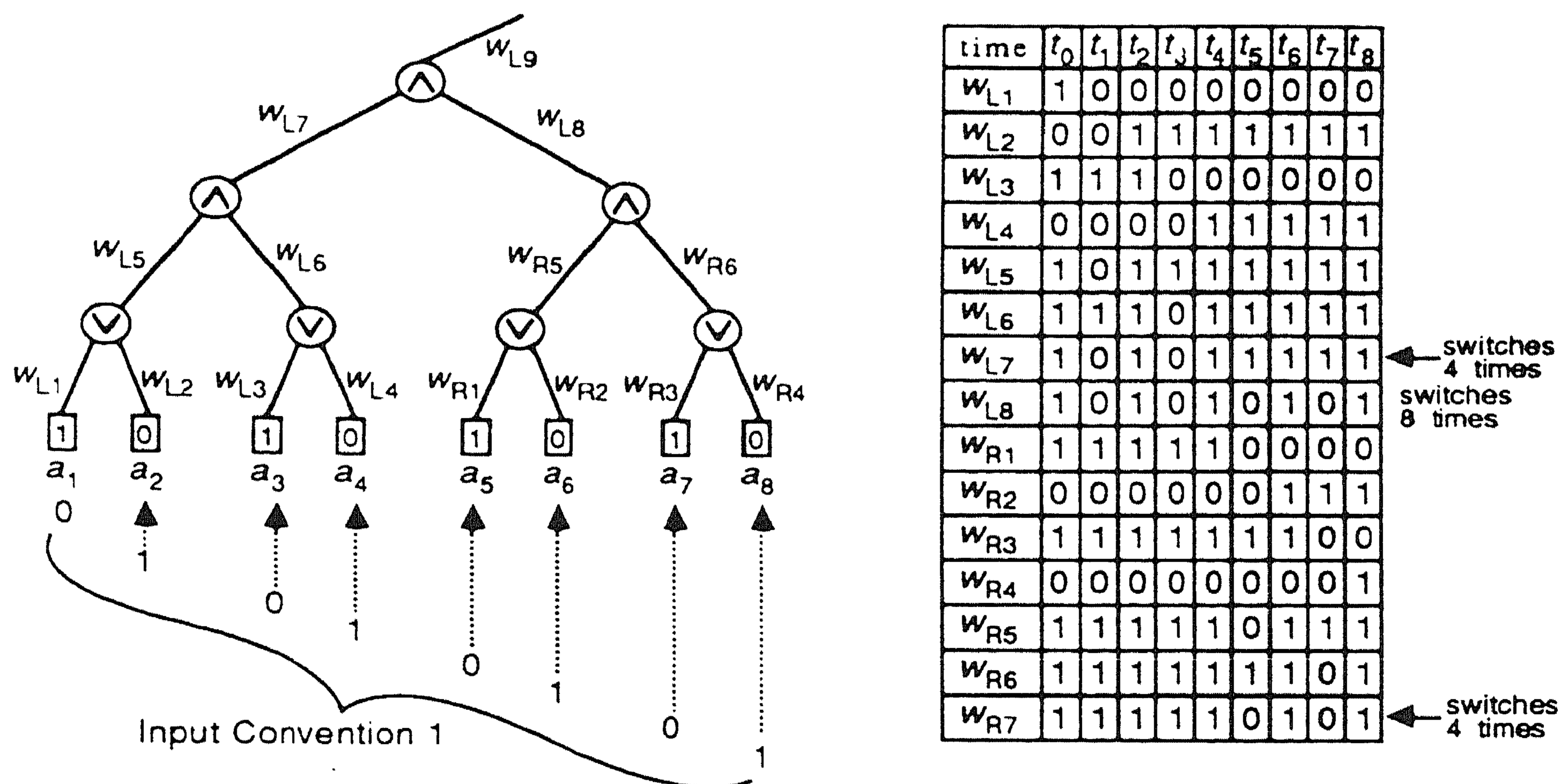


FIGURE 4.1. 8-OR/AND tree

The large amount of switching incurred by circuit T with input schedule I_1 results in an energy expenditure that is asymptotically greater than the area of T . To see this, consider two popular tree embeddings, defined below.

DEFINITIONS

Embedding 1: Let T_1 be an embedding of n -OR/AND tree T , illustrated in Figure 4.2a, such that $A_1(n)$, the area of T_1 on n leaves, is given by the following recurrence: ($A_1(n)$ counts only the horizontal wire area of T_1 , because the vertical wires contribute only $O(n)$ area.)

$$A_1(2) = 1$$

$$A_1(n) = 2A_1\left(\frac{n}{2}\right) + \frac{n}{2} = O(n \log n).$$

Embedding 2: Let T_2 be the well-known H-tree embedding of n -OR/AND tree T as shown in Figure 4.2b, such that the area $A_2(n)$ of T_2 on n leaves is given by the following recurrence [16]:

$$A_2(1) = 1$$

$$A_2(n) = 4A_2\left(\frac{n}{4}\right) + 4\sqrt{A_2\left(\frac{n}{4}\right)} = O(n).$$

The following two results show that the worst case multiswitch energy expended by an n -OR/AND tree can be asymptotically worse than the area of the tree.

THEOREM 4.2. \forall wire delay functions Δ , the worst case multiswitch energy consumed by an n -OR/AND tree T_1 under embedding 1 (Figure 4.2a) is $\Theta(n^2)$.

PROOF. Assume that $T_1: (10)^{n/2} \rightarrow_{I_1} (01)^{n/2}$. Input schedule I_1 causes any node of T_1 to switch every time its inputs switch. See Figure 4.1. Since the top level wires are $\Theta(n)$ long in embedding 1, and since they compute functions of $\Theta(n)$ inputs, these wires alone use $\Theta(n^2)$ energy in the worst case. Thus, the total worst case energy of T_1 is given by the following recurrence. Let $E_{\text{worst}}(k)$ denote the M_3 worst case energy used by T_1 with k inputs.

$$E_{\text{worst}}(2) = 1$$

$$E_{\text{worst}}(n) = 2E_{\text{worst}}\left(\frac{n}{2}\right) + \frac{n^2}{4} = \Theta(n^2). \quad \square$$

THEOREM 4.3. $\forall \Delta$, the worst case multiswitch energy consumed by an n -OR/AND tree T_2 under embedding 2 (Figure 4.2b) is $\Theta(n^{3/2})$.

PROOF. Let $E_{\text{worst}}(k)$ be the M_3 worst case energy used by VLSI circuit T_2 with k inputs. By an argument similar to that of Theorem 4.2, the following recurrence for $E_{\text{worst}}(k)$ is obtained.

$$E_{\text{worst}}(1) = 1$$

$$E_{\text{worst}}(n) = 4E_{\text{worst}}\left(\frac{n}{4}\right) + \left[\sqrt{A_2\left(\frac{n}{4}\right)} * 3 \right]$$

$$= \Theta(n^{3/2}). \quad \square$$

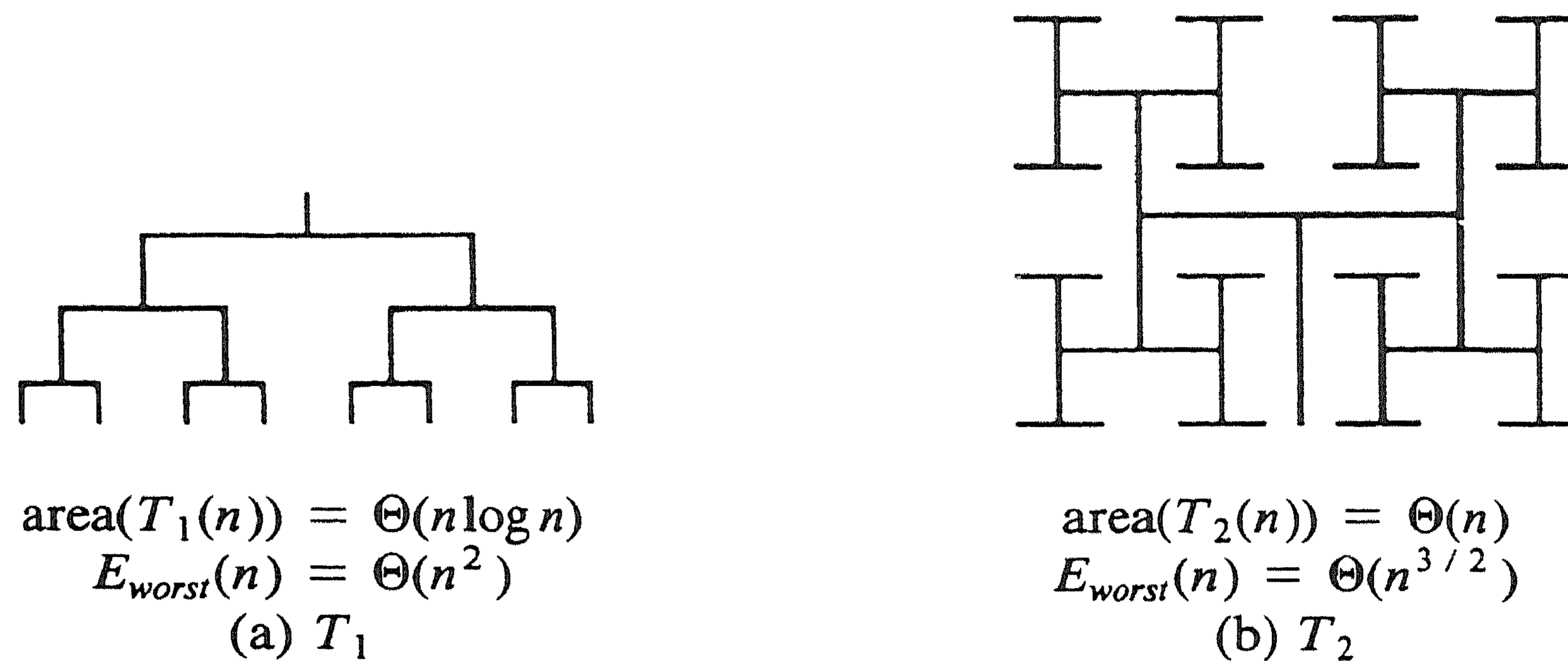


FIGURE 4.2.

Note that in the n -OR/AND tree, the row of \vee -gates enables the ‘101’ sequence, which Switching Lemma 4.1 showed was not possible in an n -AND tree. In the same way, an n -AND/OR tree enables the ‘010’ sequence, which by Lemma 4.2 cannot be achieved in an n -OR tree. Thus, the bounds on Sw and E_{worst} for an n -OR/AND tree also apply to an n -AND/OR tree with input schedule I_1 .

Consider an arbitrary parity circuit, but restrict the leaves to be on the boundary of the layout. The following derives an energy bound for parity in $MSM M_3$, which results from the bad input schedule I_1 .

THEOREM 4.4. *Computing the parity function by a CID VLSI circuit P_n consumes $E_{\text{worst}}^{M_3}(P_n) = \Theta(n^2)$, when the input nodes are on a convex boundary of P_n .*

PROOF. Consider the lower bound. Let z be the output node of P_n . Let N be a subset of $n/2$ inputs that are farthest away from z . Thus, z is $O(n)$ distance away from any input in N . Apply input schedule I_1 , i.e. switch each input one at a time. Each time an input $x_i \in N$ switches, z switches. Hence a path from x_i to z must switch. Such a path is $O(n)$ long and $n/2$ such long paths switch, using $\Omega(n^2)$ multiswitch energy.

The upper bound of $O(n^2)$ is exhibited by a circuit in which each node is a \oplus -gate. Thus, for an embedding C in which the input ports are on a convex boundary, $E_{\text{worst}}^{M_3}(C) = \Theta(n^2)$. \square

The results above show that race conditions can be very expensive of energy. The circuit designer concerned with energy conservation is thus advised to design circuits that preclude race conditions. The following section illustrates an approach for eliminating some of the multiswitch characteristics of a circuit.

5. AN ENERGY-EFFICIENT OR CIRCUIT

In [13], a novel circuit called *SOR/SAND* and layout were described for *OR* ing and *AND* ing n boolean bits. This new VLSI circuit was shown to have depth $O(\log n)$ and use worst case energy $O(n)$ in the *Uniswitch Model*. This section describes minor modifications to *SOR/SAND* that extend the circuit's energy-efficiency to the Multiswitch Model M_1 . In particular, gate delays are introduced to *SOR/SAND* via strategically placed fanout (F) nodes. In this manner, *SOR/SAND* is made synchronous.

The following recurrences describe the boolean functions $OR : \{0,1\}^n \rightarrow \{0,1\}$ and $AND : \{0,1\}^n \rightarrow \{0,1\}$ in a novel way. The reader can verify that $OR(x_1, \dots, x_n) \equiv x_1 \vee x_2 \vee \dots \vee x_n$ and $AND(x_1, \dots, x_n) \equiv x_1 \wedge x_2 \wedge \dots \wedge x_n$. The M_1 circuit realization of *OR* and *AND* is $D(\vee, \wedge)$, a modified version of the energy-efficient *SOR/SAND* circuit.

RECURRENCES

$$1) \quad OR(x_i, x_j) = x_i \vee (\bar{x}_i \wedge x_j)$$

$$OR(x_1, \dots, x_n) = OR(x_1, \dots, x_{n/2}) \vee [AND(\bar{x}_1, \dots, \bar{x}_{n/2}) \wedge OR(x_{(n/2)+1}, \dots, x_n)].$$

$$2) \quad AND(x_i, x_j) = (x_i \vee \bar{x}_j) \wedge x_j$$

$$AND(x_1, \dots, x_n) = [AND(x_1, \dots, x_{n/2}) \vee OR(\bar{x}_{(n/2)+1}, \dots, \bar{x}_n)] \wedge AND(x_{(n/2)+1}, \dots, x_n).$$

$OR(x_1, \dots, x_n)$ is abbreviated by $OR(n)$. $OR(\bar{x}_1, \dots, \bar{x}_n)$ is abbreviated by $OR(n)$. AND is similarly abbreviated. (x_1, \dots, x_n) is also written as $(x_i; x_n)$.

The reader may recall that in *MSM* M_1 , race conditions can arise when the paths to a node in a circuit have different depth. However, the appropriate placement of gate delays in *SOR/SAND* causes all paths to a given node of the circuit to have the same number of edges. The modified *SOR/SAND* circuit is denoted as $D(\vee, \wedge)$. $(D(\vee, \wedge), \Delta_1, \delta_0, I_0)$ has the uniswitch property in Multiswitch Model M_1 , when $D(\vee, \wedge)$ has constant wire delay Δ_1 , zero gate delay δ_0 , and constant input delay I_0 .

The following discussion formally describes the $D(\vee, \wedge)$ circuit and corresponding layout, $L\hat{F}$, which are illustrated recursively in Figures 5.3 and 5.4. The formal description of the revised *SOR/SAND* circuit is very similar to the original definition, described in detail in [11]. Therefore, the reader familiar with [11] may easily assess the changes from Figure 5.4.

DEFINITION. $D(\vee, \wedge) = (V_D, W_D)$ is a circuit illustrated in Figures 5.3 and 5.4, such that

$$V_D = J_1 \cup J_2 \cup G \text{ where}$$

J_1 are the input nodes $\{x_1, x_2, \dots, x_n\}$ and

J_2 are the input nodes $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$.

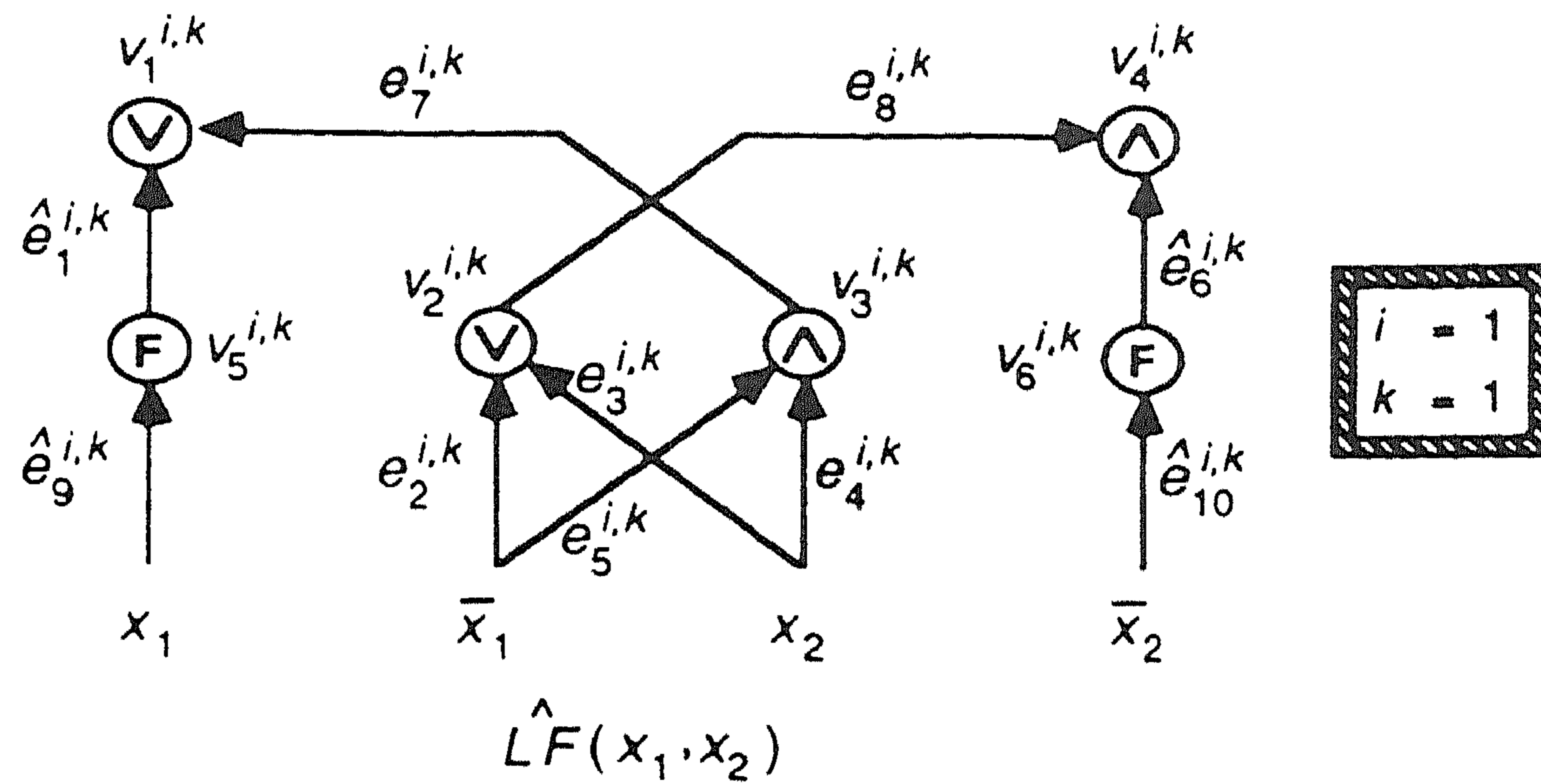


FIGURE 5.3. Base Case

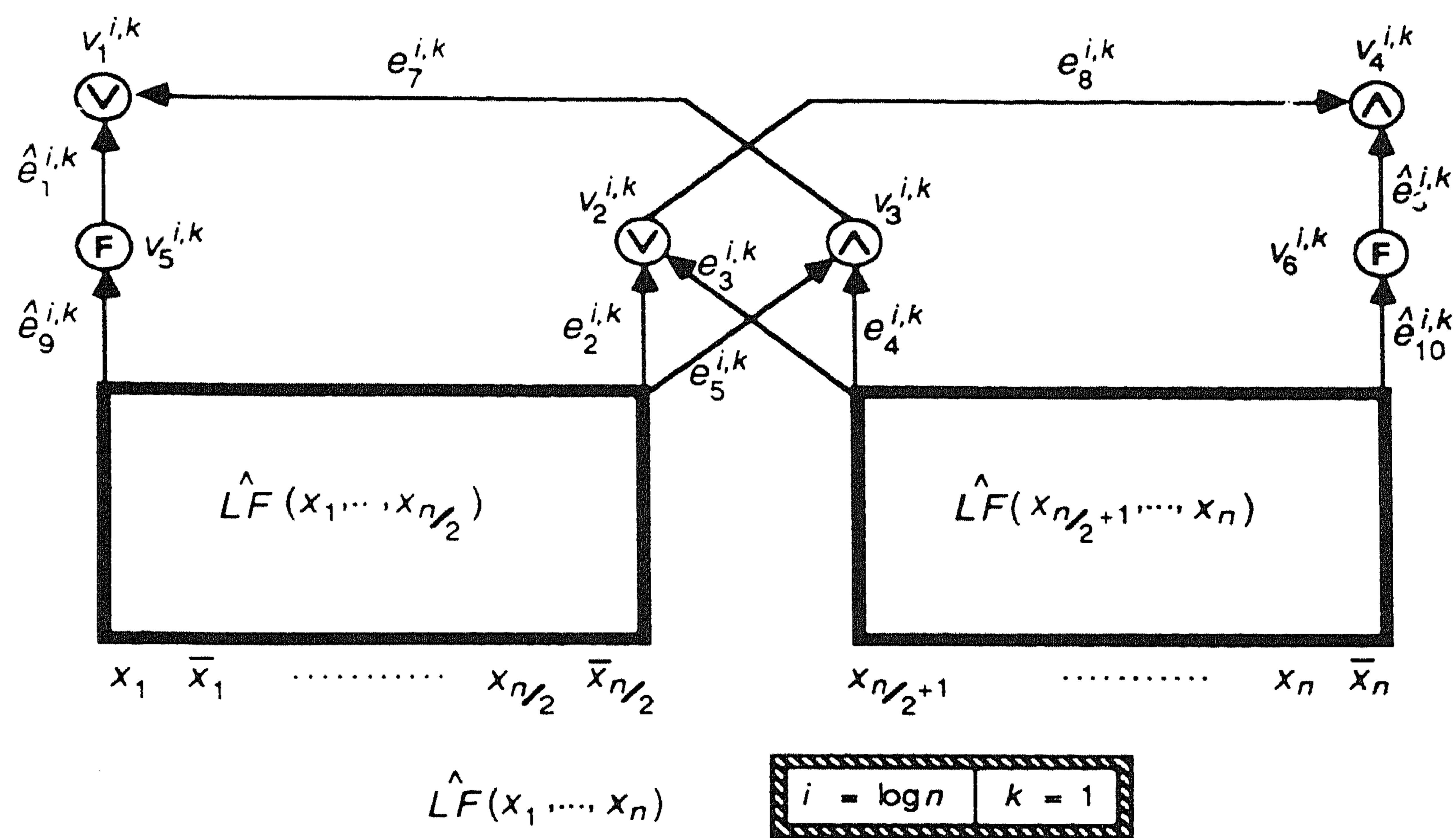


FIGURE 5.4.

G , the set of interior nodes, is as follows.

$$G = \{(v_1^{i,k}, \nabla), (v_2^{i,k}, \nabla), (v_3^{i,k}, \wedge), (v_4^{i,k}, \wedge), (v_5^{i,k}, F), (v_6^{i,k}, F)\}$$

where $1 \leq i \leq \log_2 n$, $1 \leq k \leq \frac{n}{2^i}$.

For consistency, x_k is also denoted $v_1^{0,k}$ and \bar{x}_k is also denoted as $v_4^{0,k}$.

W_D , the set of edges, is as follows.

$$W_D = \{e_j^{i,k}, \hat{e}_l^{i,k} \mid j \in \{2, 3, 4, 5, 7, 8\}, l \in \{1, 6, 9, 10\}, 1 \leq i \leq \log_2 n, 1 \leq k \leq \frac{n}{2^i} \text{ and}$$

$$\begin{aligned}
\hat{e}_1^{i,k} &= (v_5^{i,k}, v_1^{i,k}), & e_2^{i,k} &= (v_4^{i-1, 2k-1}, v_2^{i,k}), \\
e_3^{i,k} &= (v_4^{i-1, 2k-1}, v_3^{i,k}), & e_4^{i,k} &= (v_1^{i-1, 2k}, v_3^{i,k}), \\
e_5^{i,k} &= (v_1^{i-1, 2k}, v_2^{i,k}), & \hat{e}_6^{i,k} &= (v_6^{i,k}, v_4^{i,k}), \\
e_7^{i,k} &= (v_3^{i,k}, v_1^{i,k}), & e_8^{i,k} &= (v_2^{i,k}, v_4^{i,k}), \\
\hat{e}_9^{i,k} &= (v_1^{i-1, 2k-1}, v_5^{i,k}), & \hat{e}_{10}^{i,k} &= (v_4^{i-1, 2k}, v_6^{i,k}).
\end{aligned}$$

The indices i, j, k and l are used to label the nodes and edges of $D(\vee, \wedge)$ uniquely. $D(\vee, \wedge)$ differs from *SOR/SAND* in that the new fanout nodes in $D(\vee, \wedge)$ cause some edges to be split into two. In particular, fanout nodes $\{v_5^{i,k}\}$ and their incident edges $\{\hat{e}_1^{i,k}\}$ and $\{\hat{e}_9^{i,k}\}$ in $D(\vee, \wedge)$ replace edges $\{e_1^{i,k}\}$ in *SOR/SAND*; and fanout nodes $\{v_6^{i,k}\}$ and their incident edges $\{\hat{e}_6^{i,k}\}$ and $\{\hat{e}_{10}^{i,k}\}$ in $D(\vee, \wedge)$ replace edges $\{e_6^{i,k}\}$ in *SOR/SAND*.

As the reader might expect, the fanout nodes of $D(\vee, \wedge)$ do not affect the circuit's functionality. Hence, $D(\vee, \wedge)$ is functionally equivalent to *SOR/SAND*. Thus, $D(\vee, \wedge)$ computes $(OR(x_1, \dots, x_n), AND(\bar{x}_1, \dots, \bar{x}_n))$, which were recursively defined above.

The following defines an embedding of a circuit in the plane.

DEFINITION. An (I, J) -grid-with-diagonals $GD_{IJ} = (\hat{V}, \hat{E})$ is a graph where $\hat{V} = \{(k, m) \mid 0 \leq k < I, 0 \leq m < J\}$ (i.e. set of cartesian coordinates), and edges of \hat{E} join vertex pairs that are either unit distance apart or distance $\sqrt{2}$ apart. GD_{44} is illustrated in Figure 5.5.

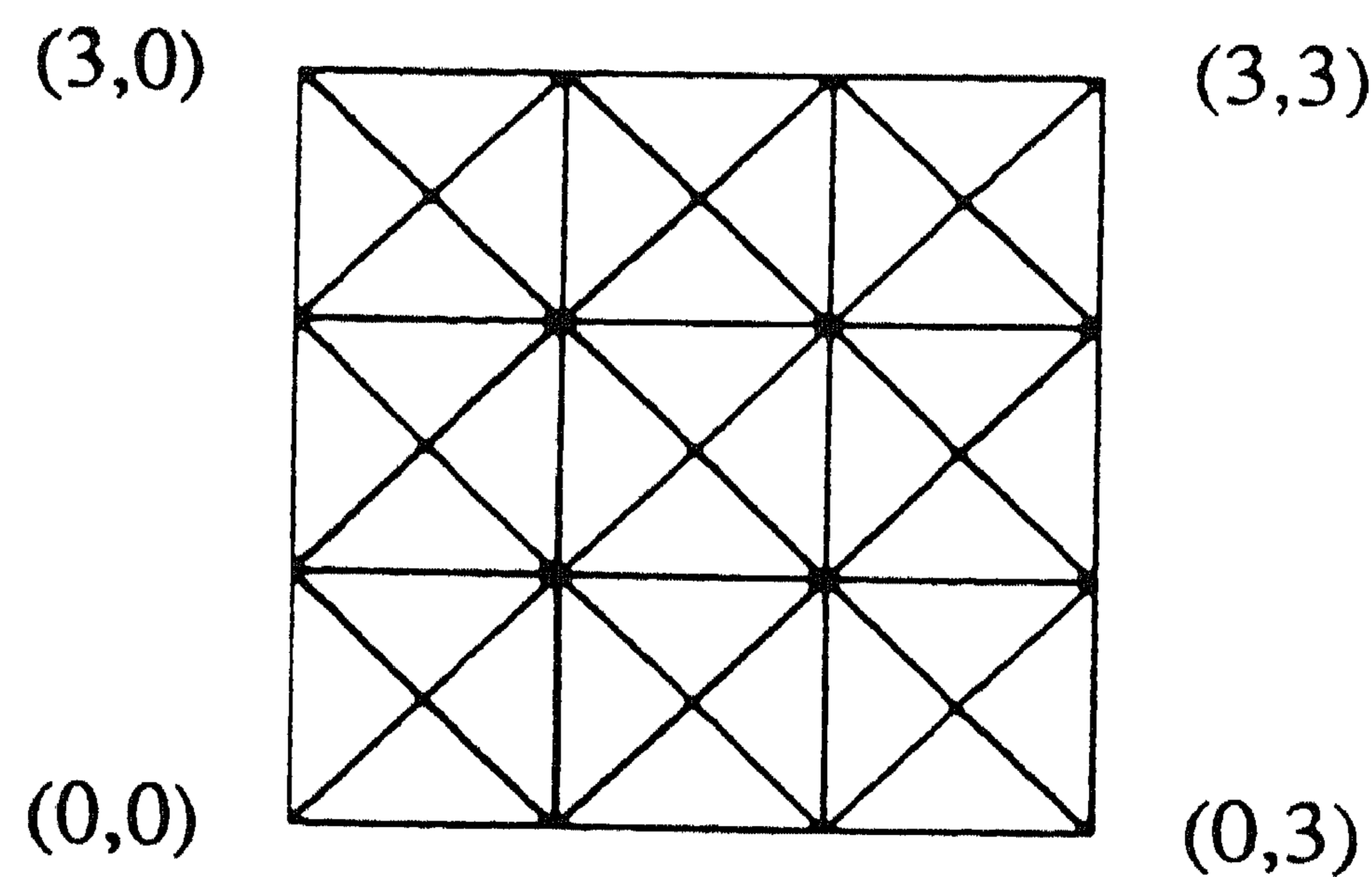


FIGURE 5.5. GD_{44}

DEFINITION. A layout (embedding, placement), Ψ , of graph $G = (V, E)$ into GD_{IJ} is a 1-to-1 mapping of V into \bar{V} and E into paths (wires) of \bar{E} such that $\forall (x, y) \in E$, $\Psi(x, y)$ is a path from $\Psi(x)$ to $\Psi(y)$, and every pair of paths in \bar{E} is edge-disjoint.

DEFINITIONS.

$$\text{height}(GD_{IJ}) \stackrel{\Delta}{=} I - 1$$

$$\text{width}(GD_{IJ}) \stackrel{\Delta}{=} J - 1$$

$$\text{area}(GD_{IJ}) \stackrel{\Delta}{=} \text{height}(GD_{IJ}) * \text{width}(GD_{IJ}).$$

Let $\hat{LF}(x_1, x_2, \dots, x_n) = \Psi(D(\vee, \wedge))$ in the following manner. (In [11], $LF(n) = \Psi(SOR/SAND(n))$ in a similar manner.) The input nodes are unit spaced on a line, with \bar{x}_j placed to the immediate right of x_j for $1 \leq j \leq n$, and the wire lengths are as follows.

$$\| \hat{e}_1^{i,k} \| = \| \hat{e}_6^{i,k} \| = 1, \| e_2^{i,k} \| = \| e_4^{i,k} \| = 1, \| \hat{e}_9^{i,k} \| = \| \hat{e}_{10}^{i,k} \| = 1, \text{ and}$$

$$\| e_3^{i,k} \| = \| e_5^{i,k} \| = \sqrt{2}, \text{ and } \| e_7^{i,k} \| = \| e_8^{i,k} \| = 2^i + \sqrt{2} - 1 \text{ for}$$

$$1 \leq i \leq \log n, 1 \leq k \leq \frac{n}{2^i}.$$

The relative location of the nodes of $D(\vee, \wedge)$ in the layout is evident from the recursive description of \hat{LF} , illustrated in Figures 5.3 and 5.4. The delay nodes are embedded as follows:

$v_5^{i,k}$ is embedded midway between $v_1^{i-1, 2^{k-1}}$ and $v_1^{i,k}$, and $v_6^{i,k}$ is embedded midway between $v_4^{i-1, 2^k}$ and $v_4^{i,k}$.

$\hat{LF}(x_1, \dots, x_n)$ is abbreviated by $\hat{LF}(n)$ or LF . Some facts about $\hat{LF}(n)$ and $D(\vee, \wedge)$:

$$1) \text{ height}(\hat{LF}(n)) = 2 + \text{height}\left(\hat{LF}\left(\frac{n}{2}\right)\right)$$

$$= 2 \log_2 n.$$

$$2) \text{ width}(\hat{LF}(n)) = 2n - 1.$$

$$3) \text{ area}(\hat{LF}(n)) = \text{height}(\hat{LF}(n)) * \text{width}(\hat{LF}(n))$$

$$= 2 \log_2 n * (2n - 1)$$

$$\approx 4n \log_2 n.$$

Note that if $T(n)$ is a complete binary tree with n leaves unit spaced on a line, $\text{height}(T(n)) = \log n$ and $\text{width}(T(n)) = n - 1$.

$\therefore \text{area}(LF(n)) \approx 4 * \text{area}(T(n)).$

4) Let $D(n)$ be the depth of $D(\vee, \wedge)$ on n inputs.

$$\begin{aligned} D(n) &= 2 + \text{depth}(D(\frac{n}{2})) \\ &= 2\log_2 n. \end{aligned}$$

The fanout nodes of \hat{LF} are used as gate delays to make the circuit synchronous. That is, all paths to a given node of LF consist of the same number of edges. Hence, when \hat{LF} changes state, the inputs to a node of \hat{LF} switch at the same time. This effectively precludes race conditions in M_1 . Hence the following theorem.

THEOREM 5.4. *The worst case energy used by $\hat{LF}(n)$ in Multiswitch Model M_1 is $E_{\text{worst}}(\hat{LF}(n)) = O(n)$.*

PROOF. Because $\hat{LF}(n)$ is synchronous, $(\hat{LF}(n), \Delta_1, \delta_0, I_0)$ has the uniswitch property in M_1 . Hence the switching behaviour of $\hat{LF}(n)$ is the same as $LF(n)$, the embedding of the *SOR/SAND* circuit. In addition, since $\text{area}(\hat{LF}(n)) = \text{area}(LF(n))$, $E_{\text{worst}}(\hat{LF}(n)) = E_{\text{worst}}(LF(n))$. $E_{\text{worst}}(LF(n)) = O(n)$ by a Theorem of [11]. \square

[2] showed that in *MSM* M_4 , $\Omega(n \log n)$ multiswitch energy is required to compute *OR* on n inputs. The problem is open for *MSM* M_2 and M_3 . In particular, assuming constant input delay, (i.e. all inputs arrive at the input ports at the same time), but non-constant wire delays, can $\hat{LF}(n)$ be made energy-efficient?

In a manner similar to the construction above, the comparator circuits of [11] can be made synchronous, and hence energy-efficient in *MSM* M_1 .

REFERENCES

1. H. ABELSON, P. ANDREAE (1980). Information transfer and area-time tradeoffs for VLSI multiplication. *CACM*, January, 20-23.
2. A. AGGARWAL, A. CHANDRA, P. RAGHAVAN (1988). Energy consumption in VLSI circuits. *Proceedings of 20th ACM STOC*, 205-216.
3. A. BORODIN (1977). On relating time and space to size and depth. *SIAM Journal of Computing* 6(4), 733-744.
4. R.P. BRENT, H.T. KUNG (1982). A regular layout for parallel adders. *IEEE Transactions on Computers* C-31, 260-264.
5. R.P. BRENT, H.T. KUNG (1981). The area-time complexity of binary multiplication. *JACM*, Vol. 28, No. 3, 521-534.
6. R.P. BRENT, H.T. KUNG (1980). On the area of binary tree layouts. *Information Processing Letters* 11(1), 46-48.

7. G. BILARDI, M. PRACCHI, F.P. PREPARATA (1981). A critique and appraisal of VLSI models of computation. *Proceedings of CMU Conference on VLSI Systems and Computations*, 81-88.
8. B. CHAZELLE, L. MONIER (1981). A model of computation for VLSI with related complexity results. *Proceedings of 13th ACM STOC*, 318-325.
9. J. HOPCROFT, J. ULLMAN (1979). *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley.
10. Z.M. KEDEM (1982). Optimal allocation of computation resources in VLSI. *Proceedings of 23rd IEEE FOCS*, 379-385.
11. G. KISSIN (1989). Upper and lower bounds on switching energy in VLSI. *JACM*, to appear.
12. G. KISSIN (1987). *Modeling Energy Consumption in VLSI Circuits*, PhD Thesis, Department of Computer Science, University of Toronto.
13. G. KISSIN (1985). Functional bounds on switching energy. *Proceedings of 1985 Chapel Hill Conference on Very Large Scale Integration*, 181-196.
14. G. KISSIN (1982). Measuring energy consumption in VLSI circuits: a foundation. *Proceedings of 14th ACM STOC*, 99-104.
15. G. KISSIN, E. KRANAKIS, J. TROMP, P. VITÁNYI (1989). *The Energy Complexity of Threshold and Other Functions*, CWI Technical Report, to appear.
16. J. LEO (1984). *Energy Complexity in VLSI*, M.S. Thesis, University of Nijmegen, The Netherlands.
17. T. LENGAUER, K. MEHLHORN (1981). On the complexity of VLSI computations. *Proceedings of CMU Conference on VLSI*, Computer Science Press, 89-99.
18. R.J. LIPTON, R. SEDGEWICK (1981). Lower bounds for VLSI. *Proceedings of 13th ACM STOC*, 300-307.
19. C. MEAD, L. CONWAY (1981). *Introduction to VLSI Systems*, Addison-Wesley.
20. C. MEAD (1986). Private communication.
21. C. MEAD, M. REM (1979). Cost and performance of VLSI computing structures. *IEEE Journal of Solid-State Circuits SC-14(2)*, 455-462.
22. M.S. PATERSON, W.L. RUZZO, L. SNYDER (1981). Bounds on minimax edge length for complete binary trees. *Proceedings of 13th ACM STOC*, 293-299.
23. F.P. PREPARATA, J.E. VUILLEMIN (1980). Area-time optimal VLSI networks for multiplying matrices. *Information Processing Letters 11(2)*, 77-80.
24. V. RAMACHANDRAN (1986). On driving many long lines in a VLSI layout. *JACM 33*, 687-701.
25. J.E. SAVAGE (1979). Area-time tradeoffs for matrix multiplication and related problems in VLSI models. *Proceedings of 17th Allerton Conference on Communications, Control and Computing*, 670-676.
26. J.E. SAVAGE (1976). *The Complexity of Computing*, John Wiley & Sons.
27. C.L. SEITZ (1980). System Timing. C. MEAD, L. CONWAY (eds.). *Introduction to VLSI Systems*, Chapter 7, Addison-Wesley.

28. L. SNYDER, A. TYAGI (1986). The energy complexity of transitive functions. *Proceedings of 24th Allerton Conference on Communication, Control and Computing*, 562-572.
29. C. THOMPSON (1980). *A Complexity Theory for VLSI*, PhD Thesis, Dept. of Computer Science, Carnegie-Mellon University.
30. C. THOMPSON, P. RAGHAVAN (1984). On estimating the performance of VLSI circuits. *Proceedings of 1984 Conference on Advanced Research in VLSI*, MIT.
31. J. VUILLEMIN (1983). A combinatorial limit to the computing power of VLSI circuits. *IEEE Transactions on Computers C-30*, 135-140.
32. A. YAO (1981). The entropic limitations on VLSI computations. *Proceedings of 13th ACM STOC*, 308-311.