Journal of
CIRCUITS, SYSTEMS, AND COMPUTERS

Volume 14 • Number 1 • February 2005

Power-Speed Trade-Off in Parallel Prefix Circuits

S. Vanichayobon, S. K. Dhall, S. Lakshmivarahan
and J. K. Antonio

World Scientific
NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Journal of Circuits, Systems, and Computers
Vol. 14, No. 1 (2005) 65–98

World Scientific
www.worldscientific.com

POWER-SPEED TRADE-OFF IN PARALLEL PREFIX CIRCUITS

S. VANICHAYOBON, S. K. DHALL*, S. LAKSHMIVARAHAN and J. K. ANTONIO
School of Computer Science, University of Oklahoma,
Norman, OK 73019-6151, USA
**sdhall@ou.edu*

Received 15 March 2003
Revised 1 February 2004

Optimizing area and speed in parallel prefix circuits has been considered important for a long time. The issue of power consumption in these circuits, however, has not been addressed. This paper presents a comparative study of different parallel prefix circuits from the point of view of power–speed trade-off. An effective circuit capacitance model that is verified through PSpice simulations is used to investigate the power consumption in parallel prefix circuits. The model results in an analytical function for power consumption for each prefix circuit considered. The degrees of freedom studied include different parallel prefix circuits and voltage scaling. The results of the study were applied to evaluate power–speed trade-offs when different prefix circuits are used in Brent's parallel adder.[5]

Keywords: Power–speed trade-off; prefix circuits; parallel adder.

1. Introduction

The three most widely accepted metrics for measuring the quality of a circuit are its area, speed, and power consumption. Optimizing area and speed has been considered important for a long time, but minimizing power consumption has been recently gaining prominence.[1–3] One important reason for minimizing power consumption of a circuit is the proliferation of portable electronic systems, such as laptops, mobile phones, and wireless devices, where maximizing battery life is important. Since it is desirable to increase the time between battery recharges in such devices, finding methods of reducing power consumption has assumed considerable importance recently.

In this paper, we study power–speed trade-off for prefix circuits. Prefix circuits play an important role in many applications. They appear in a number of areas such as the carry-look-ahead adder, ranking, packing, radix sort, etc.[4] Many new approaches for prefix circuits with the goal of optimizing depth (i.e., speed) and size (i.e., area) have been proposed.[4–8] As a result, predicting performance in terms

*Corresponding author.

of speed and area has improved. The issue of power consumption in these circuits, however, has not been addressed in existing literature. Therefore, our goal is to make a comparative study of different prefix circuits from the point of view of power–speed trade-off in order to facilitate the design choices, specifications, and resource limitations. In this study, we use the power–delay product as a quality measure for the prefix circuits. The power–delay product is the product of the circuit's power consumption and propagation delay, which represents the energy consumed by the circuit per computation.

Two major issues are addressed in this paper. The first deals with the proposed power analysis of prefix circuits. This analysis, which is verified through PSpice simulations,[9] is used to investigate the power consumption in the prefix circuits considered. The simulations were carried out considering voltage scaling. It is found that amongst all the prefix circuits considered in this paper, the divide-and-conquer prefix circuit consumes the most power. Also, the power–delay product of the LYD prefix circuit is the lowest amongst the circuits considered while the power–delay product of the divide-and-conquer is the highest.

The second issue deals with an investigation of the prefix circuit application to binary addition. The adders are implemented with selected prefix circuit designs. A parameter in the implementation is the choice of block size for computing carries in parallel. The PSpice simulations were carried out on 8-, 16-, 32-, and 64-bit binary adders. In regard to power–delay product, we have found that an optimum block size falls somewhere around the middle among the various possible block sizes for a given adder.

The rest of this paper is divided into five sections. Section 2 presents a brief survey of various prefix circuits. Section 3 reviews the sources of power consumption in a CMOS circuit and presents strategies to estimate power consumption of the circuit. Section 4 focuses on modeling the power consumption of the prefix circuits studied. Section 5 describes the analysis of power–speed trade-off of prefix circuits considered. Section 6 shows the formulation of carry calculation of binary addition as a prefix problem, followed by simulation results. Finally, Sec. 7 concludes with a discussion of the results of the paper.

2. Prefix Circuits — An Overview

A *prefix computation* is the process of taking N input values $x_1, x_2, \ldots, x_{N-1}, x_N$ and producing N output values $y_1, y_2, \ldots, y_{N-1}, y_N$ such that $y_1 = x_1$ and

$$y_i = y_{i-1} \bullet x_i = x_1 \bullet x_2 \bullet \cdots \bullet x_{i-1} \bullet x_i\,, \quad \text{for } 2 \leq i \leq N\,,$$

where $\bullet$ is an associative binary operation. A prefix circuit with N inputs can also be viewed as a layered directed acyclic graph with N input nodes, N output nodes, and at least $N - 1$ operation nodes. An operation node is neither an input nor an output node. Figure 1 illustrates the layout and the components of a prefix circuit. The numbers along the left-hand side of the layout give the depth (level) of the operation nodes on the right.

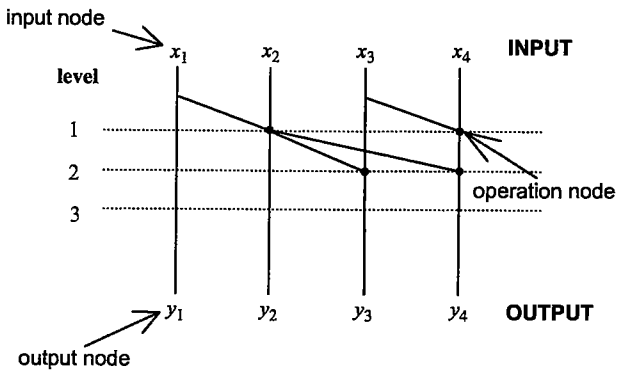

Fig. 1. An illustration of the prefix circuit's layout.

The traditional metrics for measuring the performance of a prefix circuit include its size, depth, fan-in, and fan-out. The *size* of a prefix circuit, $size(N)$, is the total number of operation nodes in the circuit. The size of a prefix circuit is related to its area. The *depth* of a prefix circuit, $depth(N)$, is the length of the longest path measured in terms of the number of operations along the path in the circuit from its input nodes to its output nodes. The circuit depth is related to a circuit's speed. A circuit with smaller depth is generally faster than one with greater depth when the fan-out of most nodes in the two circuits is similar.[10] A prefix circuit is *depth-optimal* if the circuit has the smallest depth among all possible circuits. An N-input prefix circuit is (*size, depth*)-*optimal* if $size + depth = 2N - 2$.[8] Prefix circuits have size–depth trade-off property[6] — a reduction of the circuit depth is generally achieved at the cost of an increase in circuit size. The *fan-in* of a prefix circuit is the maximum fan-in of all nodes in the circuit. The *fan-out* of a prefix circuit is the maximum fan-out of all nodes in the circuit. In this study, we are interested in prefix circuits with a fan-in of two (a fan-in of two corresponds to a binary operation) and we assume that the fan-out of the prefix circuit may be a function of N. In the rest of this section, we give a brief review of the architecture of some known prefix circuits. For full description of these circuits, refer to Refs. 4 and 11.

2.1. *The serial prefix circuit*

The layout of the serial circuit for N inputs, denoted $S(N)$, is illustrated in Fig. 2. Clearly, both size and depth of this circuit are $N - 1$. The serial prefix circuit has the smallest size amongst all prefix circuits. Moreover, the circuit is (size, depth)-optimal because the sum of its size and depth is $2N - 2$.

2.2. *Parallel prefix circuits*

Figures 3–9 give illustrations of the divide-and-conquer prefix circuit, Ladner–Fischer prefix circuit (LF_0), Ladner–Fischer circuit (LF_k), Brent–Kung circuit,

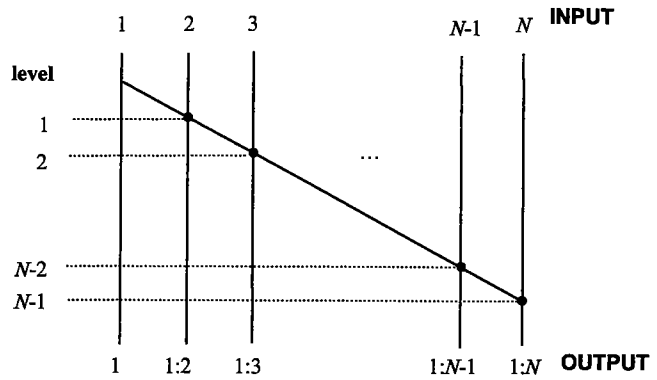

Fig. 2. An illustration of the serial prefix circuit, $S(N)$.

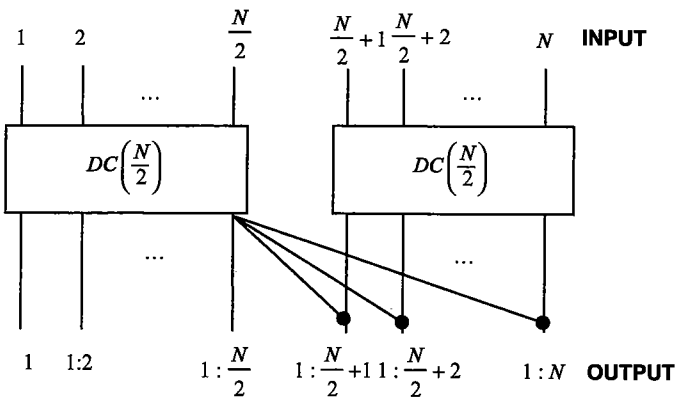

Fig. 3. An illustration of the divide-and-conquer prefix circuit, $DC(N)$.

Snir circuit, Shih–Lin prefix circuit, and LYD circuits, respectively. Information about their size, depth, and fan-out is given in Table 1. For complete details, refer to Ref. 4. All these circuits have a depth $O(\lg N)$. Snir circuits are a family of circuits whose depth lies in the range $[\max(\lg N, 2\lg N - 2), N - 1]$. The divide-and-conquer circuit and LF_0 both have $O(N)$ fan-out, whereas all the other circuits have a fan-out of $O(\lg N)$. Ladner and Fischer[6] were the first to discuss the size–depth trade-off in prefix circuits. They introduced a family of circuits, $LF_k(N)$, where $k(0 \leq k \leq \lceil \lg N \rceil)$ refers to the extra depth (above $\lceil \lg N \rceil$) used to bring about the reduction in size. The circuit size and depth depend on the value of k. Snir[8] showed that the sum of depth and size of any prefix circuit with N inputs is bounded below by $2N - 2$. He also introduced an algorithm to construct the (size, depth)-optimal prefix circuit for any N with the depth in the range $\max(\lceil \lg N \rceil, 2\lceil \lg N \rceil - 2) \leq \mathit{depth}(N) \leq N - 1$. It was conjectured by Snir[8] that (size, depth)-optimal parallel prefix circuits with depth in the range

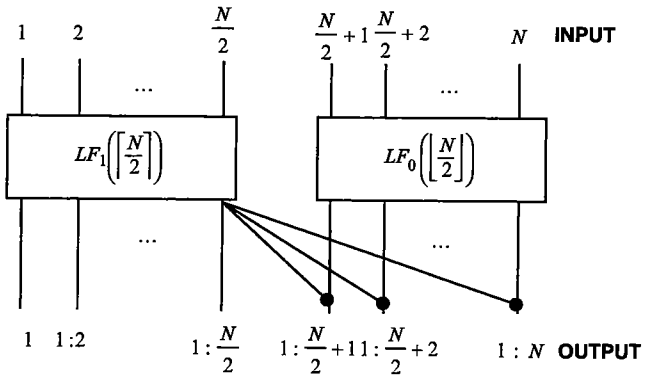

Fig. 4. An illustration of the Ladner–Fischer parallel prefix circuit when $k = 0, LF_0(N)$, derived from Ref. 6.

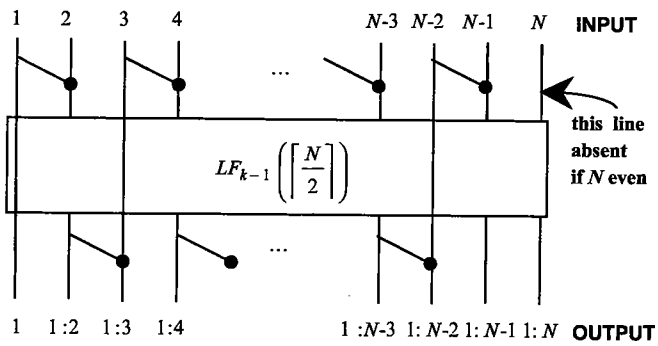

Fig. 5. An illustration of the Ladner–Fischer parallel prefix circuit when $k \neq 0, LF_k(N)$, derived from Ref. 6.

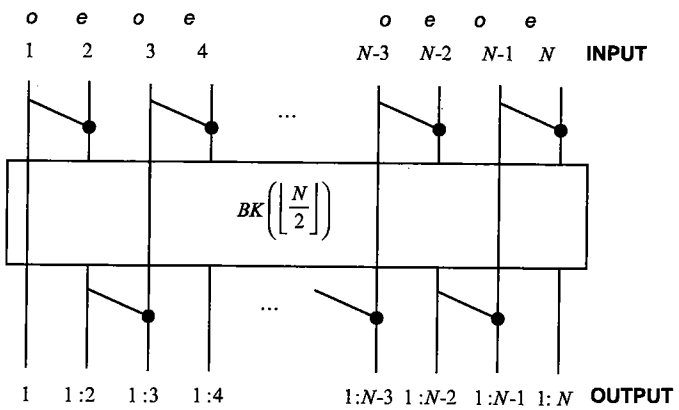

Fig. 6. A Brent–Kung parallel prefix circuit, $BK(N)$, based on divide-and-conquer strategy (o = odd, e = even).

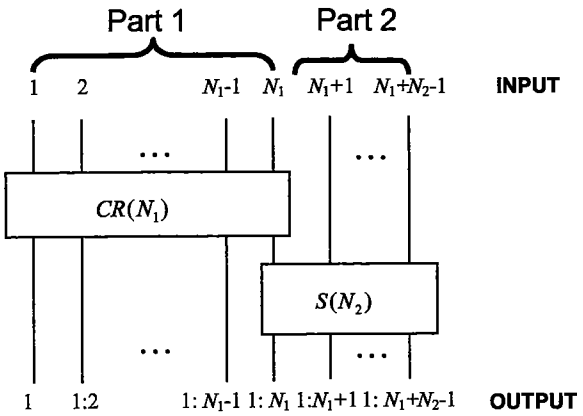

Fig. 7. An illustration of the Snir prefix circuit, $SN(N)$.

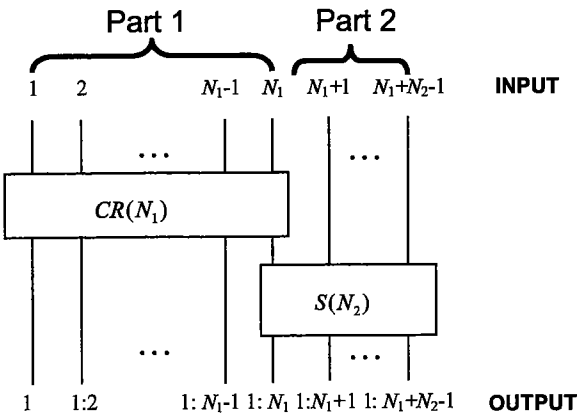

Fig. 8. An illustration of the Shih–Lin prefix circuit, $SL(N)$.

$\lceil \lg N \rceil \leq depth(N) \leq \max(\lceil \lg N \rceil, 2\lceil \lg N \rceil - 3)$ may not exist. Lakshmivarahan *et al.*[12] were the first to introduce an algorithm for a (size, depth)-optimal parallel prefix circuit with the depth in the above range. Their design provides (size, depth)-optimal circuits with a smaller depth than hitherto known in the literature. Furthermore, for $N = 9$–12, $N = 17$–20, and $N = 33$, the LYD circuits are not only (size, depth)-optimal, but also depth-optimal.

2.3. *Comparison of size and depth*

Table 1 provides a comparison of the prefix circuits illustrated in the previous subsection. While the parallel prefix circuits have desirable depths, which are all $O(\lg N)$, they differ widely in the number of nodes (i.e., size). Only four prefix

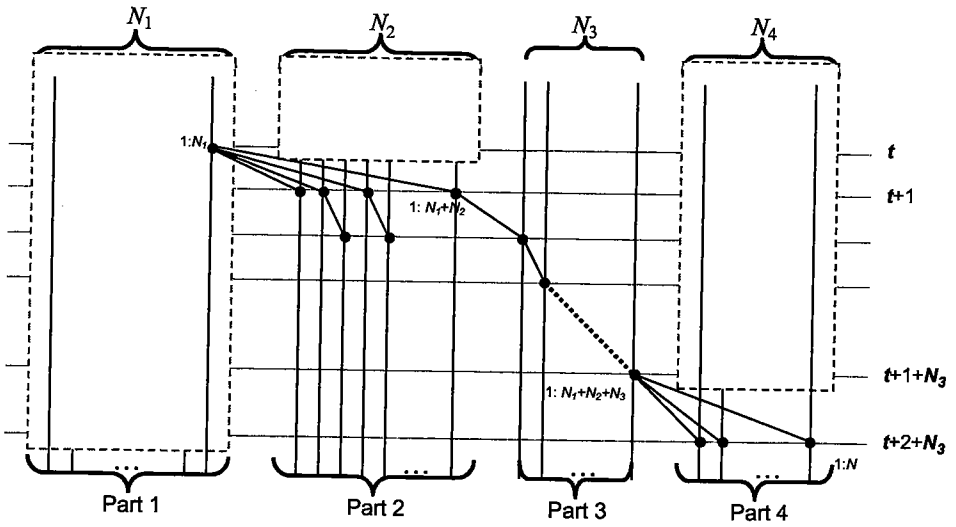

Fig. 9. The structure of $LYD(N)$, derived from Ref. 4.

circuits (i.e., serial, Snir, Shih–Lin, and LYD) are (size, depth)-optimal. The divide-and-conquer circuit and the LF_0 prefix circuit have the shortest depth and the serial circuit has the smallest size.

The size–depth trade-off does apply to every prefix circuit. For example, the serial prefix circuit has the smallest size (compared to others), but also has the longest depth, while the divide-and-conquer prefix circuit has the largest size, and the smallest depth. Although the Shih–Lin prefix circuit and the Snir prefix circuit have similar circuit layouts, Shih–Lin's circuit has a smaller depth than Snir's circuit. All circuits have unbounded fan-out except the serial circuit that has a constant fan-out of two. The divide-and-conquer prefix circuit and the LF_0 prefix circuit have the largest fan-out $((N/2)+1)$. Brent–Kung circuit, Shih–Lin circuit, and Snir circuit have the same fan-out $(\lceil \lg N \rceil + 1)$, which is smaller than that of the LYD circuit, which is $(2\lceil \lg N \rceil - 2)$.

3. Power Consumption in Circuits

In the previous section, we examined size and depth trade-offs of different prefix circuit designs. We now examine the power consumption characteristics of these circuits. In this section, after a brief summary of the sources of power consumption in complementary-symmetry metal oxide semiconductor (CMOS) circuits, strategies to estimate the power consumption of the prefix circuits are presented.

In order to minimize power consumption of a circuit, designers have to consider optimizations that reduce power dissipation at all design levels,[2,13] i.e., technology, circuit, architecture (structure), algorithm (behavior), and system levels, as shown

Table 1. A comparison of the six prefix circuits illustrated in Sec. 2, when $N = 2^n$.

Prefix circuit	Size	Depth	Fan-out	(Size, depth)-optimal
Serial	$N-1$	$N-1$	2	Yes
Divide-and-conquer	$(N/2)\lg N$	$\lg N$	$(N/2)+1$	No **depth-optimal**
LF_0	$4N - F(5+\lg N)+1$	$\lg N + k$	$(N/2^{k+1})+k$	No
LF_k, when $0 < k < \lg N - 2$	$2N(1+(1/2^k))$ $-F(5+\lg N-k)-k+1$			
LF_k, when $k \geq \lg N - 2$	$2N - \lg N - 2$	$2\lg N - 2$	$\lg N + 1$	
Brent–Kung	$2N - \lg N - 2$	$2\lg N - 2$	$\lg N + 1$	No
Snir	$2N - 2 - depth$	$\max(\lg N, 2\lg N - 2)$ $\leq depth \leq N-1$	$\lg N + 1$	Yes
LYD	$2N - 2 - depth$	$2\lg N - 6 \leq depth \leq 2\lg N - 3$	$2\lg N - 2$	Yes
Shih–Lin	$2N - 2 - depth$	$2\lg N - 5 \leq depth \leq 2\lg N - 3$	$\lg N + 1$	Yes

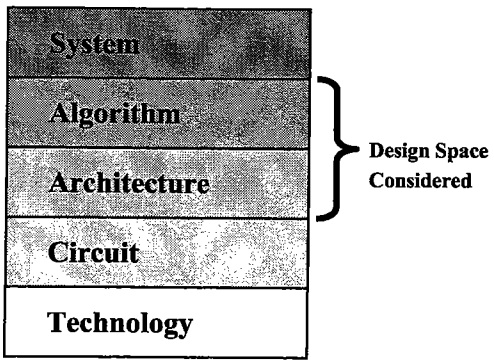

Fig. 10. Hierarchical design of electronic systems, derived from Ref. 13.

in Fig. 10. Other work has concentrated on technology (e.g., transistor sizing, optimal gate oxide thickness, etc.), circuit (e.g., pre-charge logic, domino logic, etc.), and system (i.e., cooling, packing, etc.) level power minimization.[13] However, the goal of this work concentrates on an approach for low-power design encompassing the architecture and algorithm levels. The architecture and algorithm level optimizations can achieve power savings by reducing the supply voltage (i.e., through the use of parallel structures) and reducing the effective capacitance (i.e., by decreasing the number of operation nodes). The result of this work can be used for optimizing power at the early stages of the design flow.

Note that in this work we focus on static CMOS that is locally connected, for example, systolic arrays or asynchronous systems.

3.1. *Sources of power consumptions*

Presently, CMOS technology is the most popular technology used by the digital IC (integrated circuit) industry because of its low power consumption, good scalability, and speed.[2,3,10] In CMOS circuits, power consumption is due to the following three types of current flow[10]:

(1) Static power consumption due to leakage currents.
(2) Dynamic power consumption due to short-circuit currents.
(3) Dynamic power consumption due to switching currents from repetitively charging and discharging the parasitic capacitances at the transistors' gates.

In properly designed CMOS circuits, the major portion of the power consumption is from dynamic switching.[2,3,10] As a result, in this study, we focus on the dynamic component due to the repetitive charging and discharging of the capacitive loads. Although it is possible to choose technology parameters so that the static power consumption due to leakage current is a dominant factor, such an assumption is generally an oversimplification of the complexity associated with modeling power consumption in practical circuits. If technology parameters are such that

static power consumption is dominant, then designs having a smaller size become more important. However, in situations where dynamic power consumption is dominant, selection of the most appropriate architecture becomes much more challenging because power consumption depends on the switching activities of all the circuit's signals that depend strongly on the architecture selected.

The average dynamic power consumption in a CMOS gate or module due to switching can be written as[3,10]:

$$P_{\text{switching}} = C_{\text{eff}} V_{\text{DD}}^2 f\,, \tag{1}$$

where C_{eff} is the effective capacitance switched, V_{DD} is the supply voltage, and f is the clock frequency. C_{eff} is the product of two components, the switching activity (signal transition activity), p_{f}, and the load capacitance, C_{L}. Thus, for a given circuit running at a given speed (i.e., C_{L} and f constant), power consumption is a function of the supply voltage and switching activity. Therefore, power reduction can be achieved by either operating the circuit at a lower voltage or by choosing an architecture that reduces the switching activity and/or load capacitance of the circuit's signals. In the discussion above, C_{L} and p_{f} are associated with a particular signal. Thus, power minimization (through architectural selection) must attempt to minimize C_{L} and/or p_{f} for all of the circuit's signals.

3.2. *Impact of voltage scaling*

Due to the quadratic relationship between the supply voltage and the power consumption, refer to Eq. (1), lowering supply voltage can be an effective way to achieve dramatic power savings. However, as supply voltage is decreased, circuit delay generally increases relatively independent of the logic function and style (Fig. 11). Thus, reducing supply voltage unfortunately reduces the system throughput. This loss in throughput can be recovered in some cases by applying architectural techniques to compensate for the additional delay (e.g., utilization of parallelism and pipelining). Reference 2 shows that by changing a circuit's architecture (i.e., using parallelism and pipelining) it is possible to gain significant speed improvements with only a slight increase in power requirement, hence enabling some voltage downscaling while maintaining the throughput and power consumption.

3.3. *Impact of switching activity*

Power in CMOS circuits is dissipated when logic signals in the circuit switch (i.e., change values). As a result, the level of switching activity impacts the power consumption. The manner in which the nodes in a circuit are interconnected, i.e., the circuit's architecture, can have a strong influence on the overall switching activity.[2] Some architectures may induce extra transition activity at the operation nodes called glitching transitions or dynamic hazards, which consume extra power. Glitching can be a major problem that increases the effective switching activity, causing a circuit node to undergo several rapid transitions in a single clock cycle.[2,3]

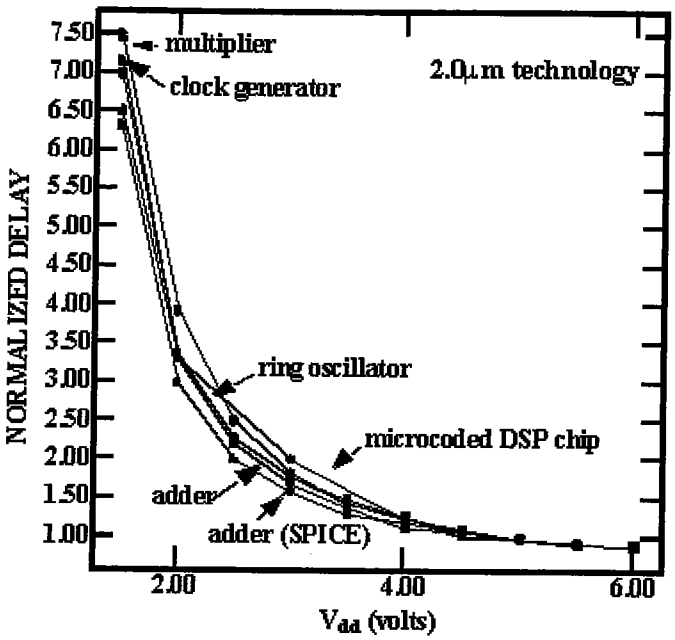

Fig. 11. Plots of normalized delay versus supply voltage (V_{dd}) for a variety of different logic circuits, derived from Ref. 2.

Figure 12 illustrates an example of the glitching behavior for a chain of eight NAND gates[3] by using a PSpice® simulation.[9] In the circuit, all of the upper inputs of the NAND gate are connected to the previous output and all lower inputs synchronously transition from logic "zero" to "one". For an ideal circuit without propagation delays, the resultant outputs VOUT2, 4, 6, and 8 would stay logic "one" all the time. However, due to the presence of delays, these outputs switch to low temporarily. This glitching causes extra power to be dissipated (due to discharging of the load capacitance). Outputs VOUT1, 3, 5, and 7 do not glitch; they just experience some propagation delay. It is noted that the degree of glitching depends on the switching pattern of the input signals;[3] the example given is just one of many possible situations where glitching occurs.

To reduce glitching activity, the architecture of the circuit, including the depth of the signal paths in the circuit, can often be modified. The following is an illustration of two different circuit architectures of a four-input adder. In Fig. 13(a), assume that all primary inputs (A, B, C, and D) arrive at the time t_0 and the implementation is nonpipelined. While the first adder makes one transition by computing A + B, the second adder also makes one transition based on C and the previous (i.e., initial) value of A + B. After the correct value of A + B has propagated through the first adder at time say $t_0 + t_p$, the second adder re-evaluates (A + B) + C, which is complete at time $t_0 + 2t_p$. Thus, there is a second transition at the second adder. Similarly, there will be three transitions at the third adder. With the "path-balanced" architecture tree of Fig. 13(b), while the first and second adders make one

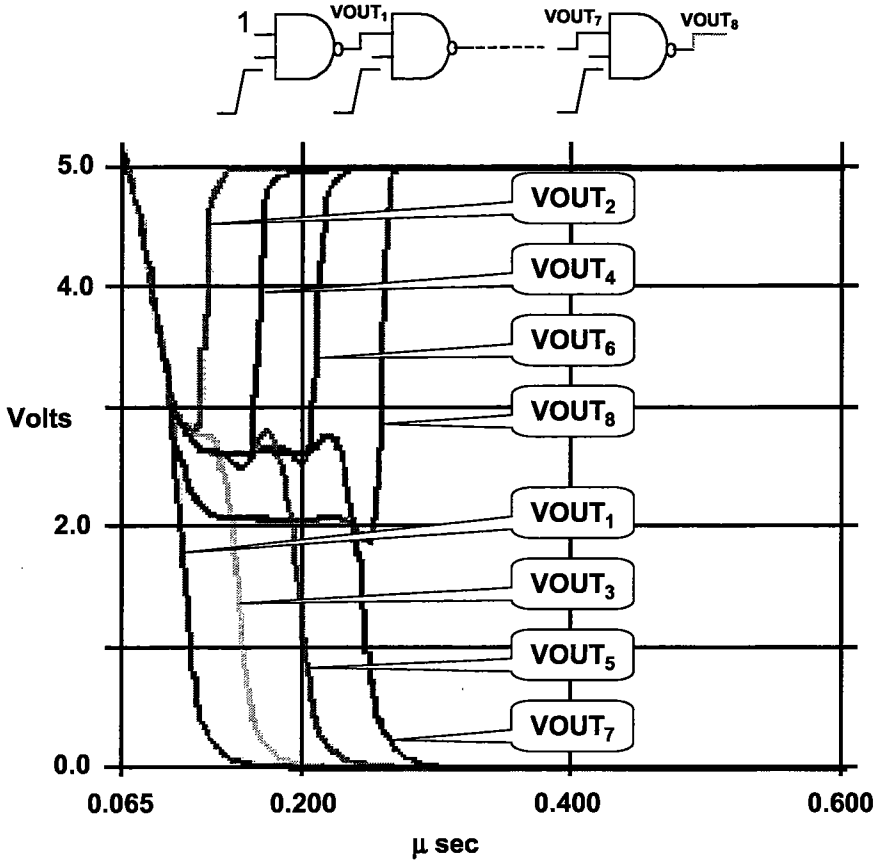

Fig. 12. An illustration of the glitching behavior of a chain of eight NAND gates, derived from Ref. 3.

transition the third adder will make only two transitions to ultimately produce the same output as in Fig. 13(a). In Ref. 2, the "total switched capacitance" of the circuit layout in Figs. 13(a) and 13(b) has been simulated by using a switch-level simulator over random input patterns. The results show that the switched capacitance of the circuit layout in Fig. 13(a) is larger than that of the circuit layout in Fig. 13(b) by a factor of 1.5 for a four-input addition, and 2.5 for an eight-input addition. Hence, increasing circuit depth generally increases the total switched capacitance due to glitching and thus increases power consumption.[2] As a consequence, the amount of transition activity (switching activity) for a layered and nonpipelined circuit can be a function of depth d and the number of nodes at each level i, w_i, as[2]:

$$\sum_{i=1}^{d} iw_i \,. \tag{2}$$

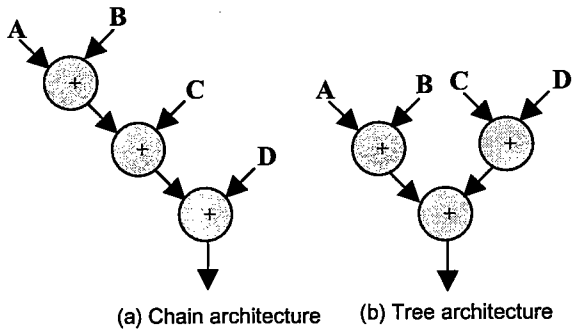

Fig. 13. An illustration of extra transition activity, derived from Ref. 2.

From this, it follows that in the worst-case estimate for the switching activity of such a circuit can grow according to $O(d^2)$, assuming a constant number of nodes at each level.

From the previous discussion and the example of Fig. 13, we observe that different circuit architectures for performing the same function can consume different amounts of power. Therefore, the implementation of the various prefix circuits in an application will have different power consumption characteristics as well. Although depth is an important parameter to consider, we cannot say with certainty that a prefix circuit with longer depth will consume more power than one with shorter depth. The reason is that depth, the number of operation nodes, and fan-out (discussed in the following subsection) among the candidate prefix circuits may differ. In prefix circuits, when the depth decreases, the number of operation nodes generally increases and vice versa. This is known as the size–depth trade-off.[4,6] As a result, the overall switching activity in a prefix circuit not only depends on its depth but also on the number of operation nodes. This will be considered in detail in the power model developed in Sec. 4.

3.4. *Impact of fan-out*

Besides the switching activity at an operation node, the node's fan-out also has an effect on power consumption in a circuit design:[10,14] the larger the fan-out, the more power the gate/circuit consumes because there are more signals. For example, using the PSpice over random input patterns, the power consumed by a two-input XOR gate is dependent upon the fan-out and the relationship is affine (Fig. 14). Therefore, to take into account the effect of fan-out on the output load capacitance of an operation node, the load capacitance of a fan-out k is modeled by $C_0 + C'(k-1)$, where C_0 is the load capacitance of a node with fan-out 1, and C' is the load capacitance for each additional fan-out (see Fig. 15).

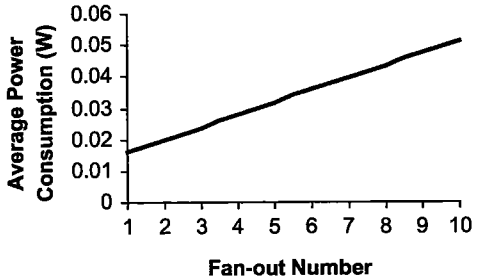

Fig. 14. Effect of fan-out on power consumption of a two-input XOR gate.

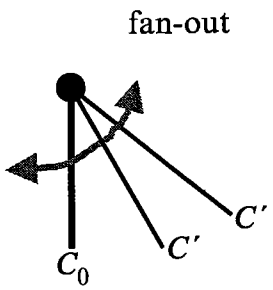

Fig. 15. Example load capacitance of a node with fan-out $k = 3$.

4. Power Modeling of Prefix Circuits

In this section, we analyze switching activity and power consumption to estimate and investigate the power–speed trade-off between various types of prefix circuits.

Having reviewed the various sources of power consumption in the previous section, we now focus on an analytical model that considers voltage scaling, switching activity (including glitching), and fan-out effect for various prefix circuits. The key to our model is in defining the overall effective capacitance of a prefix circuit.

The *effective circuit capacitance* of an N-input prefix circuit, $cap_{\text{eff}}(N)$, is the effective load capacitance of all nodes in the circuit. The effective circuit capacitance depends on signal activity and the effects of signal glitching. Thus, if a node output experiences two transitions due to glitching, its effective capacitance is twice that of the physical capacitance. Because the degree of glitching depends on input signal patterns, we consider derivations of a worst-case scenario in which glitching at the nodes are assumed to be the maximum possible. Though studying an average case behavior might be useful, we have chosen here to model the worst-case behavior. For, in studying expected behavior, it would require the assumptions about the underlying probability distributions of the input signal patterns. Secondly, the

worst-case behavior does bound the expected behavior. By scaling the effective circuit capacitance by the circuit clock frequency and V_{DD}^2, we arrive at our power estimate:

$$P = cap_{\text{eff}}(N) V_{DD}^2 f \,. \tag{3}$$

The effective circuit capacitance evaluation for various circuits according to our method is made in two steps. As the first step, in Sec. 4.1, we assume that the load capacitance for each operation node is independent of the fan-out and has value C_0. In the second step, we first obtain the residual circuit by reducing the fan-out of each operation node by one, and then compute its load capacitance assuming that the load capacitance of each node is C'. This step is repeated $k-1$ times where k is the fan-out of the given circuit. This step is described in detail in Sec. 4.2. The effective circuit capacitance is the sum of the values obtained in steps 1 and 2.

Due to lack of space, in the following, we compute the effective circuit capacitance for the divide-and-conquer prefix circuit. The effective circuit capacitance for the other prefix circuits introduced in Sec. 2 can be computed similarly (for details refer to Ref. 11).

4.1. *Step 1 — The constant output capacitance*

In this step, we assume that the physical output capacitance of each operation node is constant, C_0. Let $Kcap_{\text{eff}}(N)$ be the effective circuit capacitance under the constant output capacitance assumption, $depth(N)$ be the depth of the circuit, w_i be the number of operation nodes in the circuit at level i, and C_0 as the assumed constant load capacitance of one node. Then, from Eq. (2),

$$Kcap_{\text{eff}}(N) = \left(\sum_{i=1}^{depth(N)} i w_i \right) C_0 \,. \tag{4}$$

4.1.1. *Applying step 1 to the divide-and-conquer prefix circuit*

Let $N = 2^n$. From the layout of the divide-and-conquer prefix circuit, $DC(N)$, in Fig. 3, $DC(N)$ is built from two $DC(N/2)$ circuits and by connecting output $1{:}N/2$ from the first $DC(N/2)$ to each of the outputs of the second $DC(N/2)$ at level $depth(N/2)+1 = \lg(N/2)+1 = \lg N$. Thus,

$$Kcap_{\text{eff}}(N) = 2Kcap_{\text{eff}}(N/2) + ((N/2)\lg N)\, C_0 \,, \quad \text{with } Kcap_{\text{eff}}(2) = 1 \cdot C_0 \,.$$

The first part of $Kcap_{\text{eff}}(N)$ is the constant output capacitance from the two circuits with $(N/2)$ inputs while the second part is the capacitance from the last level of $DC(N)$. Solving this recurrence, we get

$$Kcap_{\text{eff}}(N) = (N/4)\left((\lg N)^2 + \lg N\right) C_0 \,.$$

$Kcap_{\text{eff}}(N)$ for the other prefix circuits can be computed similarly, although they are generally more challenging because w_i is not a simple function of the input size and level (for details refer to Ref. 11).

4.2. *Step 2 — Capacitance of residual circuit*

We have assumed that a node with fan-out $k \geq 1$ has an output capacitance modeled by $C_0 + (k-1)C'$. However, the capacitance computed in Sec. 4.1 is based on the assumption that the capacitance of each node is C_0 irrespective of the fan-out of the node. We next account for the residual component $(k-1)C'$ for a node with fan-out $k, k > 1$, by introducing the concept of the *residual circuit.* The residual circuit of a prefix circuit is the circuit obtained by eliminating one of the fan-out lines from each operation node of the given prefix circuit. For example, Fig. 16 shows the residual circuit of the divide-and-conquer prefix circuit. This residual circuit is the result of removing one of the fan-outs (i.e., the vertical fan-out) from each operation node of the circuit in Fig. 3. We can compute the capacitance of this residual circuit, $Rcap_{\text{eff}}(N)$, by assuming constant output capacitance of C' for all operation nodes. We then construct the residual circuit of the current residual circuit by removing one fan-out from each operation node and compute its residual output capacitance. We continue accumulating the capacitances after every reduction until there are no more fan-outs to remove. Thus, the overall effective circuit capacitance of the prefix circuit is then calculated by

$$cap_{\text{eff}}(N) = Kcap_{\text{eff}}(N) + Rcap_{\text{eff}}(N)\,.$$

4.2.1. *Applying step 2 to the divide-and-conquer prefix circuit*

From the layout of the divide-and-conquer prefix circuit in Fig. 3, an operation node at level $depth(N/2)$ has the maximum fan-out, which is $(N/2) + 1$. After removing the vertical fan-outs, the residual circuit is shown in Fig. 16. The operation

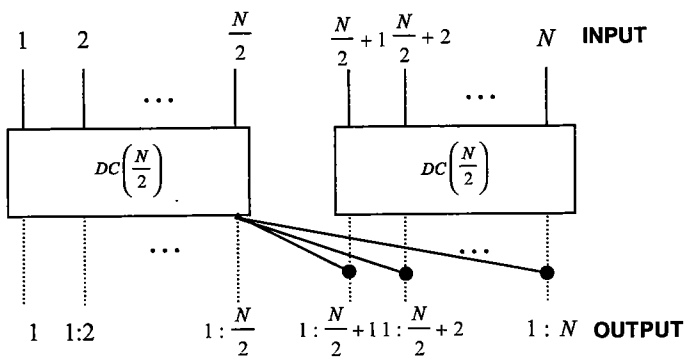

Fig. 16. The residual circuit of the divide-and-conquer prefix circuit, $DC(N)$, shown in solid lines.

node of the residual circuit at level $depth(N/2)$ has the maximum fan-out, which is $(N/2)$.

Let $N = 2^n$. The capacitance of the residual circuit is as follows:

$$Rcap_{\text{eff}}(N) = 2Rcap_{\text{eff}}(N/2) + ((N/2)\lg(N/2))\,C', \quad \text{with } Rcap_{\text{eff}}(2) = 0\,.$$

The first part of $Rcap_{\text{eff}}(N)$ is the residual output capacitance of the two circuits with $(N/2)$ inputs while the second part is the residual output capacitance of the last node in the first residual circuit. Solving the recurrence, we get

$$Rcap_{\text{eff}}(N) = 2Rcap_{\text{eff}}(N/2) + ((N/2)\lg(N/2))\,C' = (N/4)\left((\lg N)^2 - \lg N\right)C'\,.$$

Thus, the effective circuit capacitance for the divide-and-conquer prefix circuit is as follows:

$$cap_{\text{eff}}(N) = \left\{(N/4)\left((\lg N)^2 + \lg N\right)\right\}C_0 + \left\{(N/4)\left((\lg N)^2 - \lg N\right)\right\}C'\,.$$

To summarize, the divide-and-conquer prefix circuit has $O(N \lg N)$ size, $O(\lg N)$ depth, and $O(N(\lg N)^2)$ effective circuit capacitance.

Table 2 provides a comparison of the effective circuit capacitance of the prefix circuits described in Sec. 2. The serial prefix circuit has the largest effective circuit capacitance, $O(N^2)$. All parallel prefix circuits have $O(N \lg N)$ effective circuit capacitance, except the divide-and-conquer prefix circuit and the LF_0 prefix circuit whose values are $O(N(\lg N)^2)$.

5. Simulation Studies

This section uses PSpice to simulate the prefix circuits' power consumption so that it can be compared with the power model based on effective circuit capacitance of Sec. 4. The degrees of freedom studied include different prefix circuit designs and voltage scaling. First, results from the effective capacitance model are presented, followed by the corresponding simulation studies.

5.1. *Results from effective capacitance model*

Figures 18, 20 and 22 give analytical estimates of delay, power consumption, and power–delay product. Figure 18 is obtained by assuming the circuits' delay to be proportional to the circuits' depth and applying the normalized delay from Fig. 17 in order to account for the effect of the supply voltage on the delay. The power consumption curves of Fig. 20 are based on our effective capacitance model of Sec. 4; for this study we assumed that $C_0 = 3C'$.[15] The power consumption values are "normalized" by the factor $C'f$. For example, at a supply voltage of 2.8 V, the normalized power consumed by the divide-and-conquer circuit for $N = 64$ is calculated as

$$P(normalized) = cap_{\text{eff}}(N)V_{\text{DD}}^2 f/(C'f) = (2496C')\,(2.8)^2\,f/(C'f) \approx 19\,569\,.$$

Table 2. Comparison of effective circuit capacitance of prefix circuits.

Prefix circuit	$cap_{\text{eff}}(N)$
Serial	$\left\{\frac{N(N-1)}{2}\right\}C_0 + \left\{\frac{(N-1)(N-2)}{2}\right\}C'$
Divide-and-conquer	$\left\{\frac{N}{4}\left((\lg N)^2 + \lg N\right)\right\}C_0 + \left\{\frac{N}{4}\left((\lg N)^2 - \lg N\right)\right\}C'$
Brent–Kung	$\left\{1 + \frac{3}{2}N\lg N - \frac{1}{2}\left[2N + (\lg N)^2 + \lg N\right]\right\}C_0 + \left\{3\left(1 + \frac{N}{2}\lg\frac{N}{2}\right) - \frac{1}{2}\left(3N + \left(\lg\frac{N}{2}\right)^2 + \lg\frac{N}{2}\right)\right\}C'$
LF_k	$\left\{\frac{N}{4}\left((\lg N)^2 + \lg N\right)\right\}C_0 + \left\{\frac{N}{4}\left((\lg N)^2 - \lg N\right)\right\}C' \le LF_k \le \left\{1 + \frac{3}{2}N\lg N - \frac{1}{2}\left[2N + (\lg N)^2 + \lg N\right]\right\}C_0 + \left\{3\left(1 + \frac{N}{2}\lg\frac{N}{2}\right) - \frac{1}{2}\left(3N + (\lg\frac{N}{2})^2 + \lg\frac{N}{2}\right)\right\}C'$
Snir	$\left\{\left[1 + \frac{3}{2}N_1(\lg N_1)\right] - \left[\frac{1}{2}\left[2N_1 + (\lg N_1)^2 + (\lg N_1)\right]\right] + \left[N_2\lceil(\lg N_1)\rceil - \lceil(\lg N_1)\rceil + \left(\frac{N_2^2 - N_2}{2}\right)\right]\right\}C_0 + \left\{\left[3\left(1 + \frac{N_1}{2}\lg\frac{N_1}{2}\right) - \frac{1}{2}\left(3N_1 + \left(\lg\frac{N_1}{2}\right)^2 + \lg\frac{N_1}{2}\right)\right] + \left[\lceil\lg N_1\rceil(N_2 - 1) + \frac{1}{2}\left(N_2^2 - 3N_2 + 2\right)\right]\right\}C'$

Table 2. (*Continued*)

Prefix circuit	$cap_{\text{eff}}(N)$
Shih–Lin	$\{[1+\frac{3}{2}N_1(\lg N_1)] - [\frac{1}{2}[2N_1+(\lg N_1)^2+(\lg N_1)]] + \left[N_2\lceil(\lg N_1)\rceil - \lceil(\lg N_1)\rceil + \left(\frac{N_2^2-N_2}{2}\right)\right]\}C_0$ $+\left\{\left[3\left(1+\frac{N_1}{2}\lg\frac{N_1}{2}\right)-\frac{1}{2}\left(3N_1+\left(\lg\frac{N_1}{2}\right)^2+\lg\frac{N_1}{2}\right)\right]+[\lceil\lg N_1\rceil(N_2-1)+\frac{1}{2}(N_2^2-3N_2+2)]\right\}C'$
LYD	$\left\{[1+\frac{3}{2}N_1\lg N_1]-[\frac{1}{2}[2N_1+(\lg N_1)^2+\lg N_1]]+\left[\frac{2\lceil\lg N_1\rceil^3}{3}+2\lceil\lg N_1\rceil^2+\frac{4\lceil\lg N_1\rceil}{3}+1\right]\right.$ $\left.+[(N_3+N_4)(\lceil\lg N_1\rceil+\frac{3}{2})+\frac{1}{2}(N_3^2+N_4^2)+(N_3N_4)]\right\}C_0$ $+\left\{\left[3\left(1+\frac{N_1}{2}\lg\frac{N_1}{2}\right)-\frac{1}{2}\left(3N_1+\left(\lg\frac{N_1}{2}\right)^2+\lg\frac{N_1}{2}\right)\right]+\left[\frac{2}{3}\lceil\lg N_1\rceil+\lceil\lg N_1\rceil^2+\frac{\lceil\lg N_1\rceil}{3}\right]\right.$ $\left.+\left[\frac{N_3}{2}(2\lceil\lg N_1\rceil+N_3+1)+\frac{N_4^2}{2}+N_4\lceil\lg N_1\rceil+N_3N_4+1-\frac{N_4}{2}\right]\right\}C'$

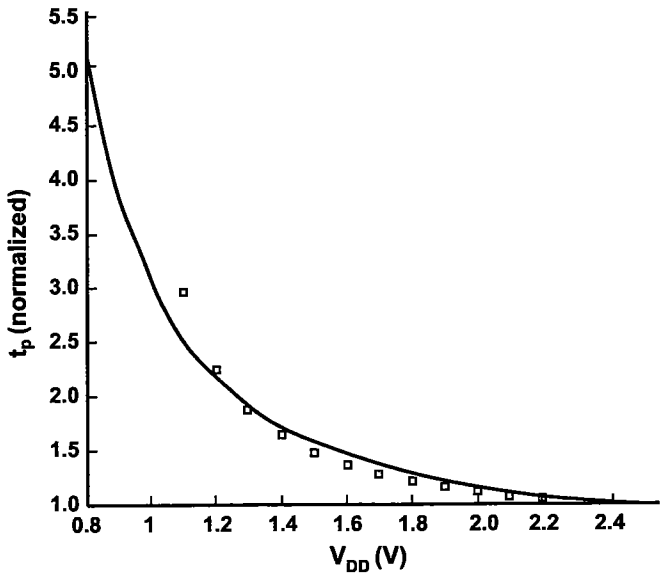

Fig. 17. Plot of supply voltage versus normalized delay, derived from Ref. 2.

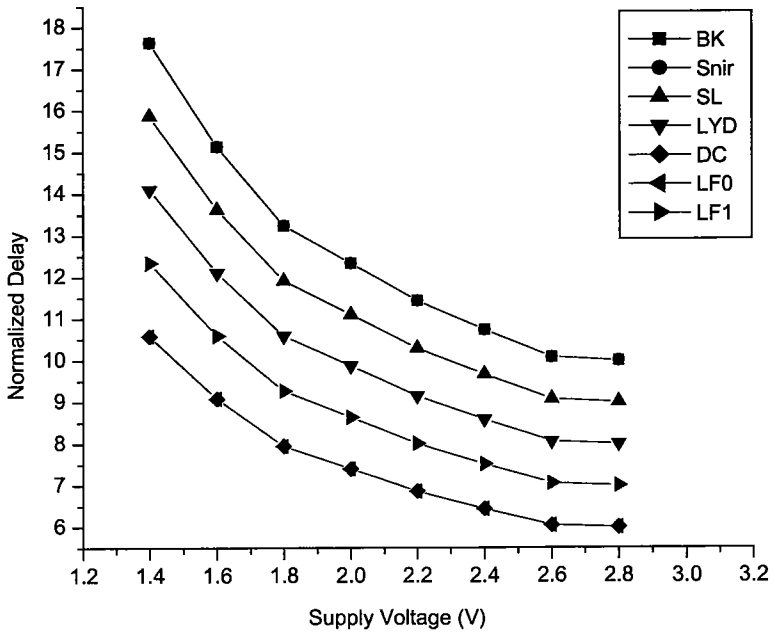

Fig. 18. Estimated delay of parallel prefix circuits when $N = 64$.

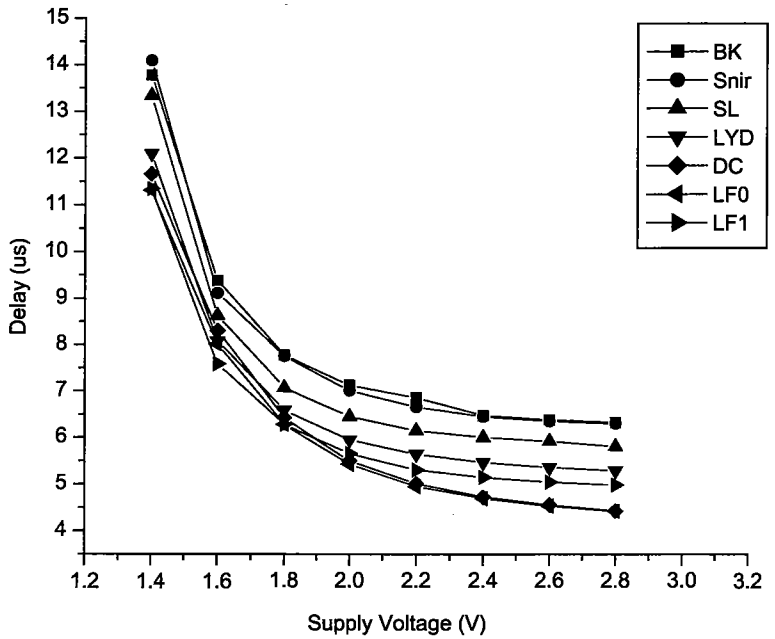

Fig. 19. Delay of the 64-bit XOR parallel prefix circuits, obtained through PSpice simulation.

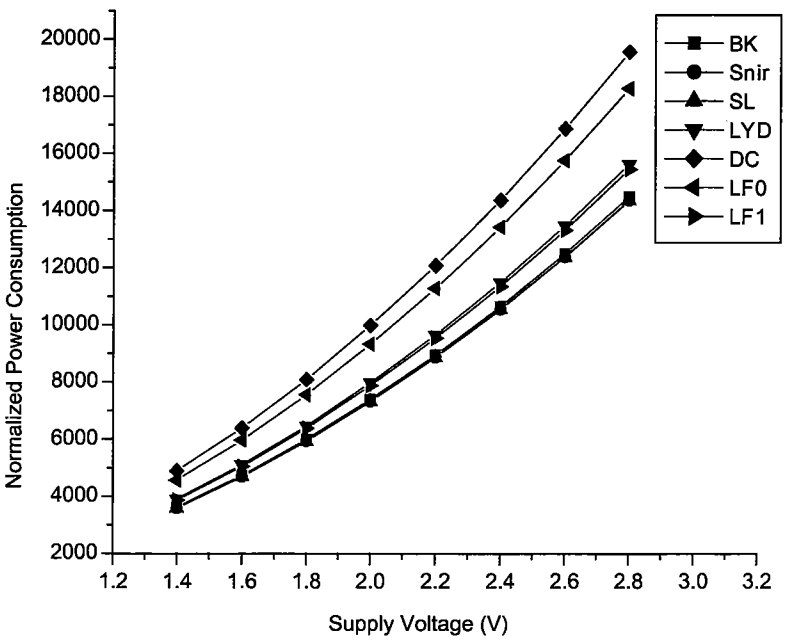

Fig. 20. Estimated power consumption of parallel prefix circuits when $N = 64$.

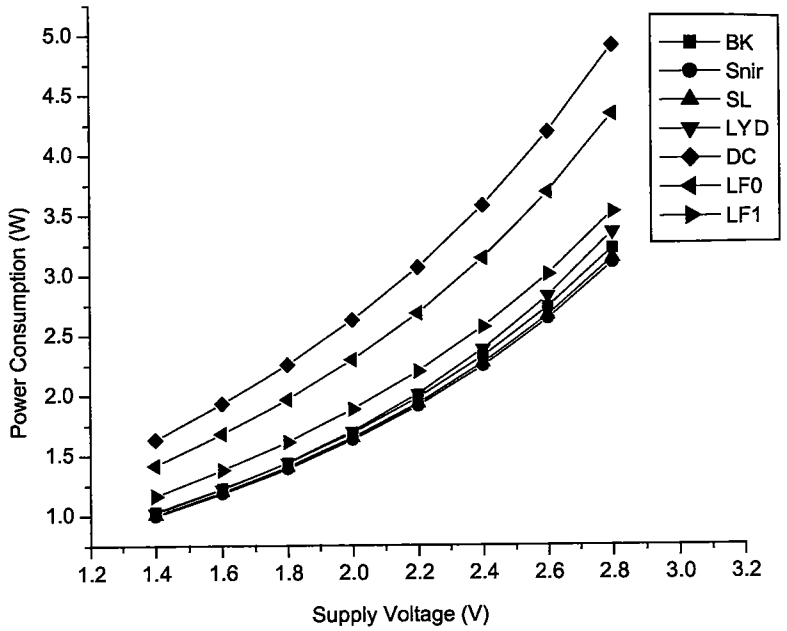

Fig. 21. Power consumption of the 64-bit XOR parallel circuits, obtained through PSpice simulation.

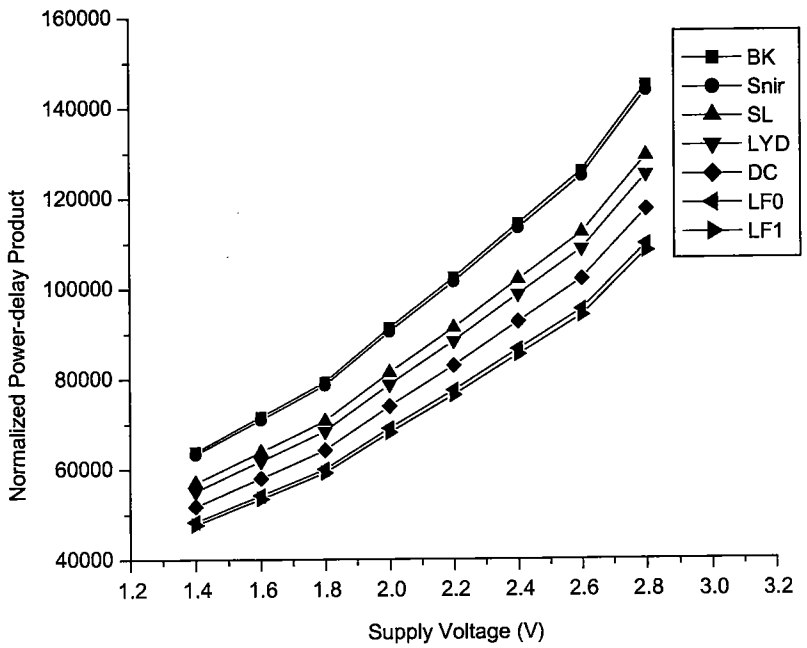

Fig. 22. Estimated power–delay product of parallel prefix circuits when $N = 64$.

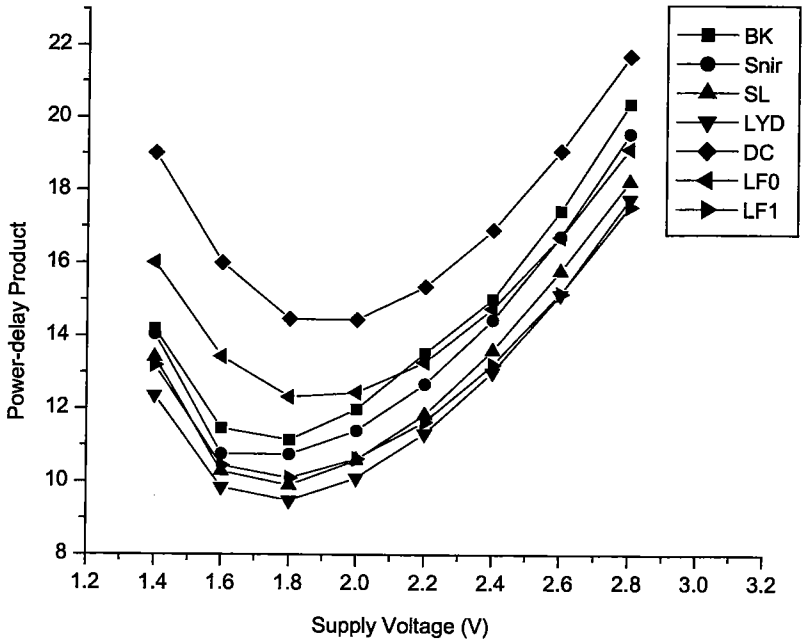

Fig. 23. Power–delay product of the 64-bit XOR parallel prefix circuits, obtained through PSpice simulation.

From Fig. 20, the divide-and-conquer prefix circuit consumes the most power for all values of supply voltage considered. Figure 22 illustrates the power–delay product. The Brent–Kung prefix circuit has the highest power–delay product while the divide-and-conquer and the LF_0 prefix circuits have a power–delay product lower than that of the Brent–Kung prefix circuit, the Snir prefix circuit, the Shih–Lin prefix circuit, and the LYD prefix circuit.

Table 3 shows the estimated power consumption of the different prefix circuits at fixed and reduced supply voltage for $N = 64$. When the supply voltage is fixed at 2.8 V, the divide-and-conquer prefix circuit consumes the most power.

Consider the objective of lowering power consumption by reducing the supply voltage, assuming a fixed acceptable delay. Delay is assumed to be proportional to depth, and that a delay proportional to a depth of 10 with $V_{DD} = 2.8\,V$ is acceptable. Thus, from Table 3, the voltage of the Brent–Kung and Snir circuits cannot be lowered, and the delay of the serial circuits is not acceptable. The voltages of five prefix circuits (i.e., the divide-and-conquer prefix circuit, the LF_0 prefix circuit, the LF_1 prefix circuit, the Shih–Lin prefix circuit, and the LYD prefix circuit, the circuits with depth less than 10) can be dropped from 2.8 V and still achieve the acceptable delay. For example, because the delay for the divide-and-conquer prefix circuit is proportional to 6 at 2.8 V, the voltage can be dropped from 2.8 to 1.48 V. This value is obtained from Fig. 18, which gives a plot of the

Table 3. Estimated power consumption based on Eq. (3) for various prefix circuits for $N = 64, C_0 = 3C'$.

Prefix circuit	Depth	$cap_{\text{eff}}(64)$	Power (normalized), $V_{\text{dd}} = 2.8\,\text{V}$	New power (normalized), after reducing V_{dd}
Serial	63	$2016C_0 + 1953C'$	62 728	—
Divide-and-conquer	6	$672C_0 + 480C'$	19 569	3280 $V_{\text{dd}} = 1.48\,V$
Brent–Kung	10	$492C_0 + 372C'$	14 488	14 488 $V_{\text{dd}} = 2.8\,V$
LF_0	6	$625C_0 + 457C'$	18 283	3065 $V_{\text{dd}} = 1.48\,V$
LF_1	7	$527C_0 + 390C'$	15 453	3987 $V_{\text{dd}} = 1.7\,V$
Snir	10	$487C_0 + 371C'$	14 363	14 363 $V_{\text{dd}} = 2.8\,V$
Shih–Lin	9	$487C_0 + 370C'$	14 355	9491 $V_{\text{dd}} = 2.4\,V$
LYD	8	$528C_0 + 410C'$	15 633	6381 $V_{\text{dd}} = 2\,V$

effect of supply voltage on the delay. The operating frequency can be decreased by a factor of 0.6. Thus, the normalized power consumed by the divide-and-conquer prefix circuit (for $N = 64$) is

$$P(\textit{normalized}) = cap_{\text{eff}}(N)V_{\text{DD}}^2 f/(C'f) = (2496C')\,(1.48)^2\,(0.6)f/f(C'f) \approx 3280\,.$$

After scaling the supply voltage, there is a power reduction in the circuits having depth shorter than 10. Among these circuits, the LF_0 prefix circuit has a major reduction in power due to its shortest depth.

5.2. *Simulation results*

PSpice simulation was carried out on different parallel prefix circuits with 64 inputs using the XOR gate as an associative binary operation. Figures 19, 21 and 23 give delay, power consumption, and power–delay product obtained through the simulation over random inputs. As was the case for the model of Sec. 4, the simulation results show that the divide-and-conquer prefix circuit consumes the most power. As the supply voltage is reduced, power consumption is also reduced, as also predicted based on circuit depth. Also, though the delay of the divide-and-conquer prefix circuit is the least for some values of the voltage supply, it is not so for very low voltages. This may be due to its very high fan-out compared to others ($\text{O}(N)$ versus $\text{O}(\lg N)$). From the point of view of the power–delay product metric, the LYD prefix circuit is found to be the best across the entire voltage scaling. This means that the circuit provides a balanced trade-off between power and delay. Another result of the simulation studies shows that the power–delay product of the divide-and-conquer circuit is the highest, followed by that of the LF_0 circuit. This is at variance with our model prediction and may be due to the fact that these

circuits have a very high fan-out (see Table 1 for fan-out). Although our effective capacitance model considers fan-out in predicting power, the effect of fan-out is not accounted for in our model for delay (i.e., proportional to circuit depth).

Also according to the simulation, with the voltage-scaling technique, the LYD prefix circuit has the least power consumption compared to other circuits. For example, let us assume the maximum acceptable delay is 6.4 μs. From Figs. 19 and 21, to achieve this time delay, the supply voltage of the divide-and-conquer, LF_0, LF_1, Shih–Lin, and LYD prefix circuits can be scaled to 1.8, 1.78, 1.78, 2, and 1.8 V, respectively. Therefore, the power consumptions of divide-and-conquer, LF_0, LF_1, Shih–Lin, and LYD prefix circuits are 2.25, 1.94, 1.59, 1.64, and 1.44 W, respectively. This shows that power reduction of about 1.6 times can be obtained without speed loss by using the LYD prefix circuit with appropriately chosen power supply compared to the divide-and-conquer prefix circuit.

5.3. *Summary*

In this section, the power consumption and the power–delay product of seven parallel prefix circuits were compared based on analytical models and simulation studies. We have shown that the use of our effective circuit capacitance model provides results that are accurate when compared to PSpice simulations. We have also shown that exploiting parallelism to a certain level, coupled with the use of low supply voltage, can be used to reduce the power consumption without throughput loss. The main discrepancy between the analytical model and the simulation is the power–delay product metric of the divide-and-conquer and the LF_0 prefix circuits. This may be due to the fact that the fan-out of these circuits is very high as compared to other circuits, and the effect of fan-out on delay is not accounted for analytically. In this analysis, we have assumed that the delay is uniquely determined by the depth of the circuit. The results of the simulation of the divide-and-conquer circuit in particular indicate that large fan-out, in addition to contributing to more power, may also indirectly affect the delay.

6. Binary Addition as a Prefix Problem

In the previous section, the performance in terms of time delay, power consumption, and power–delay product of parallel prefix circuits considered was investigated. In this section, we extend the investigation to their application in binary addition.

An addition of two binary numbers is of great interest to digital designers since it is the most commonly used operation in many other operations (e.g., counting, multiplication, division, etc.). Many researchers have investigated the various implementations of adder circuits. Examples are Refs. 16–20. There are a number of ways of formulating the process of binary addition. Each way provides different insights and thus suggests different implementations. Although each implementation is available to serve different requirements, the focus of various implementations is on the quick calculation of all carry bits, since the key to fast addition is

the fast calculation of all the carries. In one of these implementations, the addition of two binary numbers is expressed as a prefix problem by transforming the computation of all carry bits to prefix computation. The adder using this technique is called a prefix adder. In our study, we concentrate on investigating and comparing power consumption of a prefix adder based on Brent's algorithm.[21] Brent's algorithm transforms the carry computations to a prefix problem and hence is an ideal candidate for studying prefix circuits.

In the following, the method of implementing a fast parallel prefix addition based on Brent's algorithm[21,22] is described and details of how the computation of all carry bits is transformed into the prefix computation are given.

Let $a = a_N a_{N-1} \ldots a_2 a_1$ and $b = b_N b_{N-1} \ldots b_2 b_1$ be two integers to be added. Let $s = (a + b) \bmod 2^N$, where $s = s_N \ldots s_2 s_1$. Define $p_i = a_i \vee b_i$ and $g_i = a_i \wedge b_i$. Then, for $1 \leq i \leq N$,

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$

with

$$c_i = p_i \wedge (g_i \vee c_{i-1}) , \tag{5}$$

and $c_0 = 0$. The carry bit c_i is the carry from the ith bit position, p_i is a *carry propagate* condition, and g_i is a *carry generate* condition. According to Eq. (5), we need the carry bit c_{i-1} $(1 \leq i \leq N)$ for computing the sum bit s_i. By distributing the propagate bit p_i and the generate bit g_i to $c_i = p_i \wedge (g_i \vee c_{i-1})$ in Eq. (5), we obtain

$$c_i = p_i \wedge (g_i \vee (p_{i-1} \wedge (g_{i-1} \vee \ldots \vee (p_1 \wedge g_1) \ldots))) , \tag{6}$$

where $c_1 = p_1 \vee g_1$. Therefore, the implementation of the fast addition can be carried out in three stages: the preprocessing stage, the carry computation stage, and the postprocessing stage (see Fig. 24). The preprocessing stage computes the carry propagate bit p_i and the carry generate bit g_i in parallel in just one unit step (that is $p_i = a_i \vee b_i$ and $g_i = a_i \wedge b_i$; for $1 \leq i \leq N$). In the carry computation stage, the calculation of all carry bits is converted into the prefix circuit problem, which is discussed later in this section. The inputs of the prefix circuit for carry calculation are the carry propagate bits and the carry generate bits from the preprocessing stage. Once all the carry bits are known, the postprocessing stage produces the sum bits in two steps (that is, $s_i = a_i \oplus b_i \oplus c_{i-1}$; for $1 \leq i \leq N$). The time in preprocessing and postprocessing stages is negligible compared to the computation time of the carry. As a result, computing all carry bits quickly is the key to high-speed addition.

Brent[21] has presented an algorithm to transform the computation of the carry for the parallel addition of two N-bit integers to a prefix problem. The following discussions are derived from Refs. 21 and 22. Let $T_{\mathrm{A}}(N)$ be the time required to add two N-bit binary numbers. Then from the above discussion

$$T_{\mathrm{A}}(N) = CarryComputationTime + 3 .$$