Area-Time-Power Tradeoffs in Parallel Adders

Chetana Nagendra, *Member, IEEE,* Mary Jane Irwin, *Fellow, IEEE,* and Robert Michael Owens, *Member, IEEE*

Abstract—**In this paper, several classes of parallel, synchronous adders are surveyed based on their power, delay and area characteristics. The adders studied include the linear time ripple carry and manchester carry chain adders, the square-root time carry skip and carry select adders, the logarithmic time carry lookahead adder and its variations, and the constant time signed-digit and carry-save adders. Most of the research in the last few decades has concentrated on reducing the delay of addition. With the rising popularity of portable computers, however, the emphasis is on both high speed and low power operation. In this paper we adopt a uniform static CMOS layout methodology whereby short circuit power minimization is used as the optimization criterion. The relative merits of the different adders are evaluated by performing a detailed transistor-level simulation of the adders using HSPICE. Among the two's complement adders, a variation of the carry lookahead adder, called ELM, was found to have the best power-delay product. Based on the results of our experiments, a large adder design space is formulated from which an architect can choose an adder with the desired characteristics.**

I. Introduction

WHETHER it is a general-purpose system or an application specific processor, addition is by far the most frequently used operation. For example, Chen *et al.* found that addition was the most frequently used operation among a set of real-time digital signal processing benchmarks [10]. It is not surprising, therefore, that adders have received a lot of attention from researchers [5], [8], [12], [15], [25] and consequently computer architects find a myriad of adder designs at their disposal. They would, of course, like to use the best adder; but what identifies the "best" adder? Is it the fastest or the smallest or the least power hungry or the one that is easiest to integrate into the system or the one that is most fault-tolerant? It has not yet been possible to integrate the various performance criteria into a single cost function as is evident in the following words of Mead and Conway: "*Perhaps the greatest challenge that VLSI presents to computer science is that of developing a theory of computation that accommodates a more general model of costs involved in computing*" [17]. With this in mind, we do not attempt to rank adders according to some fixed criterion. Instead, the goal of the study presented in this paper is to evaluate the characteristics of different classes of parallel, synchronous adders. Apart from aiding a designer in selecting an adder with favorable characteristics, this paper is also aimed at providing insight into design tradeoffs that can save power and enhance performance.

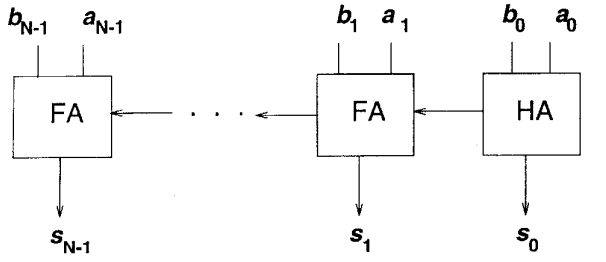


Fig. 1. Ripple carry adder.

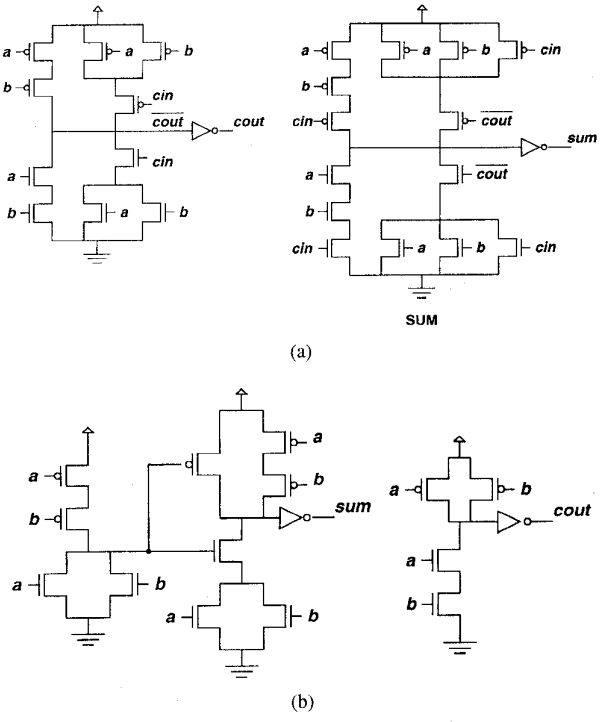


Fig. 2. Transistor diagrams of (a) full adder. (b) Half adder.

In this paper, we study parallel adders where all the inputs are available before the start of the computation. The parallel adders we have chosen for our experiments vary widely in their speed, area and power requirements. Since asynchronous systems are not very common, we deal with synchronous parallel adders only. The adders studied include linear time ripple carry and manchester carry chain adders, square-root time carry skip and carry select adders, logarithmic time carry lookahead adder and its variations, and constant time signed-digit and carry-save adders. In order to perform a fair and

Manuscript received January 27, 1995; revised August 22, 1995. This paper was recommended by Associate Editor W. Burleson.

C. Nagendra is with the California Microprocessor Division, Advanced Micro Devices, Sunnyvale, CA 94088 USA.

M. J. Irwin and R. M. Owens are with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 USA.

Publisher Item Identifier S 1057-7130(96)06488-9.

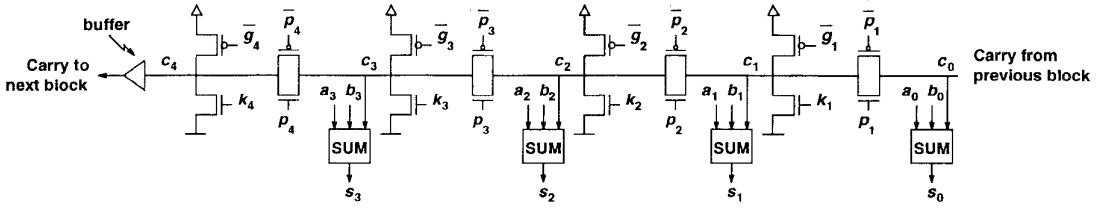


Fig. 3. A 4-b block of a Manchester Carry Chain Adder.

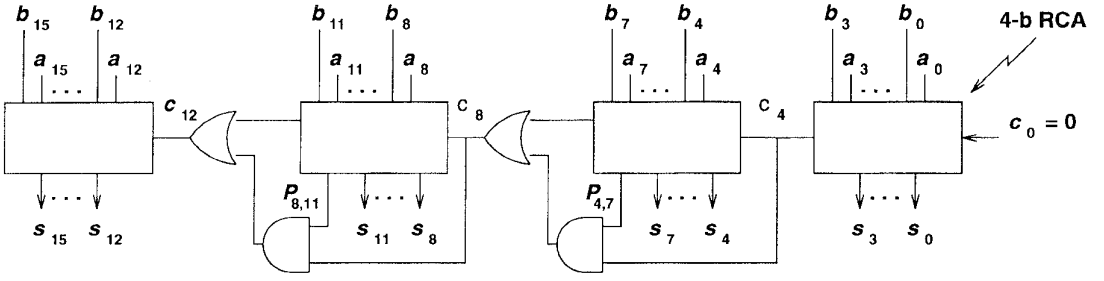


Fig. 4. A 16-b Carry Skip Adder.

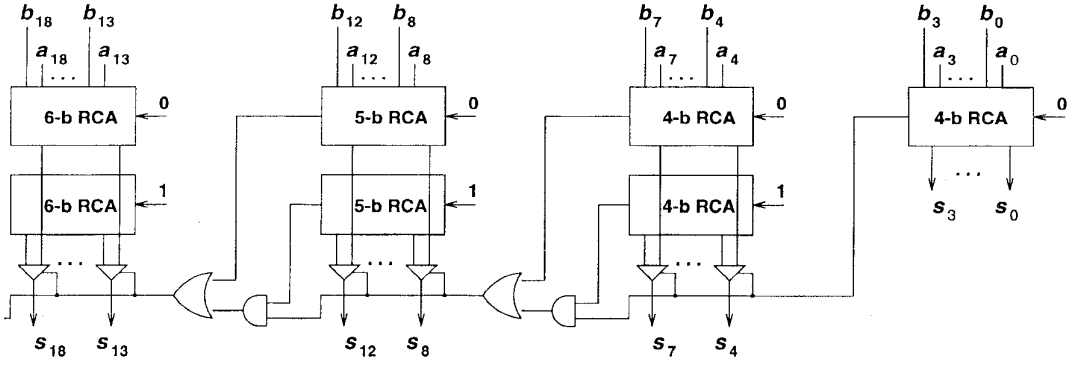


Fig. 5. A 19-b Carry Select Adder.

realistic evaluation of the different adders, we generate layouts of each adder for a wide range of precisions and simulate them using HSPICE. We use static CMOS circuit design style, since it is popular for its high packing density, low power dissipation and high yield [26]. Since speeding up addition has been widely investigated, our layout methodology concentrates on reducing power consumption. The layout techniques we use to reduce short circuit power consumption have an added advantage of improving the speed of the circuit [4].

The results presented in this paper build on those in [20]. Callaway and Swartzlander [7] have studied some types of parallel adders, but this paper consists of a more comprehensive and in-depth analysis. Our earlier results [21], [19] were based on circuits which were sized from a speed perspective. In this paper, we use a different approach whereby power minimization is used as the optimization criterion.

The rest of the paper is organized as follows. We define the problem and give an overview of the number systems used in this paper in Section II. In Section III, we give a brief description of the different adders, following which the sources of power dissipation in CMOS are identified in Section IV. Our circuit generation methodology is described in Section V. Next, Section VI details the experimental results and the results are analyzed in Sections VII–XI, before ending with conclusions in Section XII.

II. Preliminaries

We define the addition problem as: given an n-bit augend A and an n-bit addend B, produce their n-bit sum S, where A, B, and S are all represented in the same format. For the sake of simplicity, the carry-in to the least-significant bit is assumed to be zero and overflow/underflow is ignored. While most of the adders use a two's complement format, the signed-digit adders use a signed-digit number system, and the carry save adders use a carry save representation. Following is a brief overview of the three number representations. A more detailed description can be found in [14] and [25].

2.1 Two's Complement Number System

Suppose $A = a_{n-1} \cdots a_1 a_0$ is an n-bit two's complement (TC) number represented in binary, then a_{n-1} denotes the sign

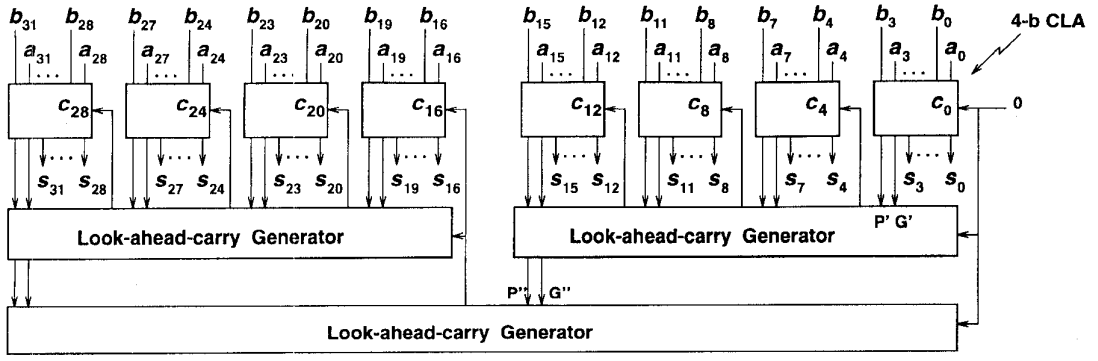


Fig. 6. A 32-b Carry Lookahead Adder.

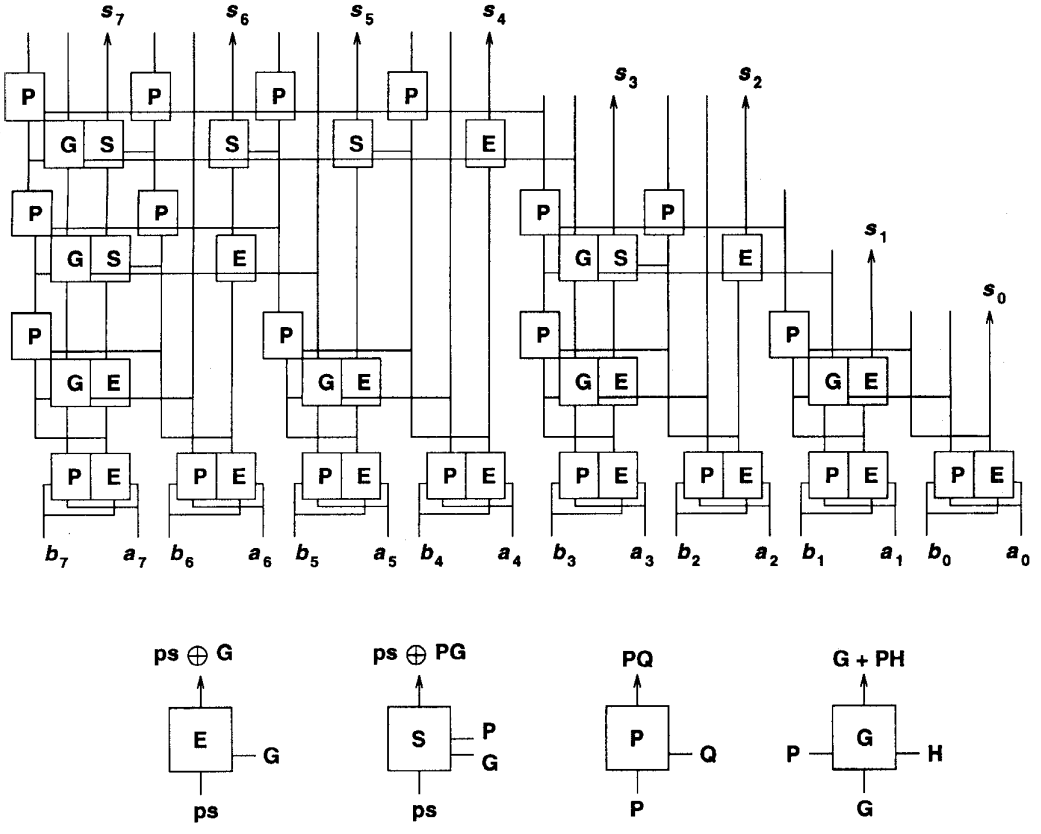


Fig. 7. An 8-b ELM adder.

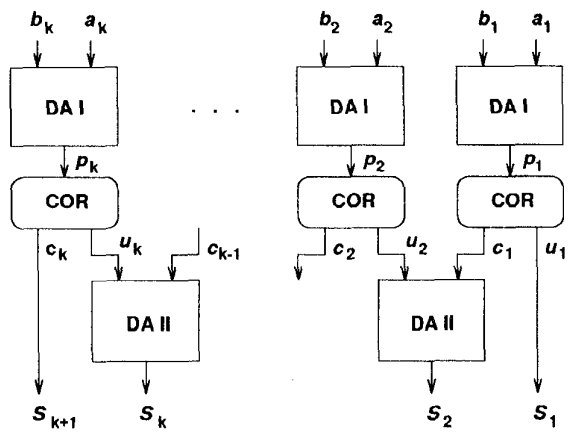
of the number. The magnitude of X is given by $-a_{n-1}.2^{n-1}+\Sigma_{i=0}^{n-1}\ a_i.2^i$. The advantage of using TC representation over sign-magnitude representation is that it does not require any special sign computation.

2.2 *Signed-Digit Number Representation*

In [1], Atkins gives a good overview of the signed digit (SD) number system, where he describes them as positional number representations with a constant base $r \geq 3$ in which the individual SD's $\in \{-\alpha, \cdots, -1, 0, 1, \cdots, \alpha\}$, where $r/2 < \alpha < r$. In this paper we consider only those bases which are powers of 2 (i.e., 4, 8, 16, 32, and so on.) Each signed digit is encoded using TC encoding, where each signed digit occupies $b = (\lceil \log r \rceil + 1)$ bits. To represent an n-bit number, $k = \lceil n/\lceil \log r \rceil \rceil$ signed digits are required. The value of a k-digit

Signed Digit Addition (A, B, S)
For $i = 0$ to m do
1. Compute a temporary sum digit $p_i = a_i + b_i$.
2. Compute an interim sum digit u_i and a carry digit c_{i+1}: $r \cdot c_{i+1} + u_i = p_i$ such that $|u_i| \leq r - 2$ and $|c_{i+1}| \leq 1$.
3. Compute the final sum: $s_i = u_i + c_{i+1}$.

Fig. 8. The radix-r OSD addition algorithm.



(a)

Base r	# digits k	# bits/digit b
128	5	8
64	6	7
32	7	6
16	8	5
8	11	4
4	16	3

n = 32 bits

(b)

Fig. 9. Signed digit adder. (a) SDA. (b) Values of k and b for different r.

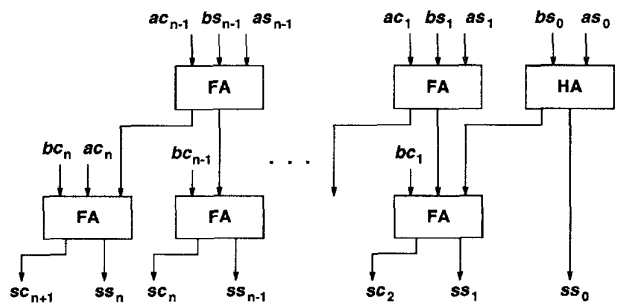


Fig. 10. Carry save adder.

TABLE I
ASYMPTOTIC TIME AND AREA REQUIREMENTS OF n-bit ADDERS

Adder Type	Abbreviation	Time	Area
Ripple Carry Adder	RCA	$O(n)$	$O(n)$
Manchester Carry Chain adder	MCC	$O(n)$	$O(n)$
Constant width carry SKip adder	CSK	$O(\sqrt{n})$	$O(n)$
Variable width carry SKip adder	VSK	$O(\sqrt{n})$	$O(n)$
Carry SeLect adder	CSL	$O(\sqrt{n})$	$O(n)$
Carry Lookahead Adder	CLA	$O(\log n)$	$O(n \log n)$
Brent and Kung adder	B&K	$O(\log n)$	$O(n \log n)$
ELM adder	ELM	$O(\log n)$	$O(n \log n)$
Signed Digit adder (base-r)	SD-r	$O(b)^\dagger$	$O(n)$
Carry Save Adder	CSA	$O(1)$	$O(n)$

† b is the number of bits per digit.

radix-r SD number, $A = a_{k-1}a_{k-2}\cdots a_1a_0$ is given by

$$A = \sum_{i=0}^{k-1} a_i \cdot r^i.$$

2.3 Carry-Save Number Representation

The carry-save (CS) number system is an encoded number representation, which is most commonly used when three or more operands are to be added as in the accumulation of the partial products in multiplication. An n-digit CS number A is made up of n CS digits, each of which consists of a carry and sum pair (ac_{i+1}, as_i), where $ac_{i+1}, as_i \epsilon \{0, 1\}$. The magnitude of $A = ac_nac_{n-1}\cdots ac_1 + as_{n-1}as_{n-2}\cdots as_0$ is given by:

$$X = \sum_{i=0}^{n} (ac_i + as_i) \cdot r^i$$

where $as_n = 0$, and the least significant carry ac_0 is assumed to be zero.

III. AN OVERVIEW OF THE ADDERS

Most of the adders we have simulated are standard ones whose descriptions can be found in literature and in computer arithmetic books [14], [25]. However, for the sake of completeness, we present a brief description of the different adder classes along with appropriate references. All adders were optimized by removing inverters, wherever possible, by switching signal polarities between successive levels.

3.1 Ripple Carry Adders

The ripple carry adder (RCA) and the manchester carry chain adder (MCC) are both $O(n)$ time, $O(n)$ area adders, where n is the width of the operands. In the worst case, a carry can propagate from the least significant bit position to the most significant bit position. A block diagram of the RCA is shown in Fig. 1. The full adder and half adder designs used to build the RCA are shown in Fig. 2. The full adder (**FA**), also referred to as a (3,2) counter, computes a sum bit, $\Sigma = a \cdot b \cdot cin + (a + b + cin) \cdot \overline{cout}$ and an outgoing carry bit, $cout = a \cdot b + (a + b) \cdot cin$. That is, it adds two operand bits with the incoming carry bit to produce a sum bit and an outgoing carry bit. The half adder (**HA**), also referred to as a (2,2) counter, adds two operands to compute: $sum = a \oplus b$ and $cout = a \cdot b$.

The MCC, shown in Fig. 3 is a VLSI adder that makes use of the well-known carry generate, propagate and kill functions [25].

$$\text{Propagate:} \quad p_i = a_i \oplus b_i$$
$$\text{Generate:} \quad g_i = a_i \cdot b_i$$
$$\text{Kill:} \quad k_i = \overline{a_i} \cdot \overline{b_i}.$$

Once a carry is generated, it is quickly propagated along a chain of pass transistors until it is killed or reaches the most-

```
add( a, b, s, cin, cout) {
  <((a & b) | cin) | (((a | b) | cin) & t1)> t2 = 0;
  <((!a & !b) & !cin) | (((!a | !b) | !cin) & !t1)> t2 = 1;
  <(a & b) | (cin & (b | a))> t1 = 0
  <(!a & !b) | (!cin & (!b | !a))> t1 = 1;
  <t1> cout = 0;
  <!t1> cout = 1;
  <t2> s = 0;
  <!t2> s = 1;
}
```

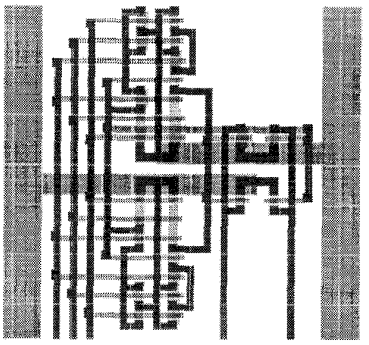


Fig. 11. Transistor Level Description to MAGIC Layout using *Perflex*.

significant bit position. Buffers are inserted after blocks of four bits each to regenerate the signal. This serves a dual purpose of minimizing delay and strengthening the carry signal, thereby reducing short-circuit power. The module **SUM** is that part of an **FA** which computes the sum [see Fig. 2(a)].

3.2 Carry Skip Adder

In the carry skip adder [15], carries start rippling simultaneously through groups. If the groups are k-bits each, then each group computes a propagate function given by

$$P_{i,i+k} = p_i \cdot p_{i+1} \cdots p_{i+k}, \quad \text{where } p_j = a_j + b_j, \quad \text{for } j = i, \cdots, i+k.$$

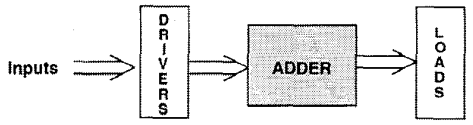
If any group generates a carry, it feeds not only the next group, but skips to the succeeding group as well if the propagate signal is true. It is an $O(\sqrt{n})$ time and $O(n)$ area adder. We consider two kinds of carry skip adders, namely CSK, which uses skip paths over equal-sized groups, and VSK, which uses nonequal groups. A block diagram of a 16-b CSK using 4-b blocks is shown in Fig. 4.

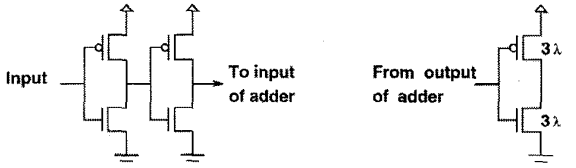
3.3 Carry Select Adder

In the carry select adder (CSL), two additions are performed in parallel—one assuming the carry-in is zero and the other assuming the carry-in is one [11]. When the carry is finally known, the correct sum is simply selected. A design using variable-sized blocks and ripple carry addition within each block is shown in Fig. 5.

3.4 Carry Lookahead Adders

Carry lookahead adders (CLA) have become popular due to their speed and modularity. They are $O(\log n)$ time and $O(n \log n)$ area adders. We use the standard scheme given in [25], which uses internal carry lookahead within 4-b slices, and full carry lookahead across blocks. For convenience, we give the block diagram of a 32-b CLA in Fig. 6. The Brent and Kung adder (B&K) [5] and the ELM adder [12] are variations of the basic CLA, designed from the point of view of design regularity. The novelty of the ELM adder is that it directly computes the sum bits in parallel, thereby reducing the number of interconnects. Fig. 7 shows the structure of an 8-b ELM adder as given by Kelliher *et al.* in [12]. Although we





(The sizes of the FETs depend on the load.)

Fig. 12. Adder with driver and load.

eliminated inverters wherever possible, the "white" processors (which are nothing but inverters) were retained in B&K in keeping with the spirit of their design [5].

3.5 Signed Digit Adder

The use of SD numbers allows addition to be performed in constant time by restricting carry propagation to at most one digit position [2]. The SD addition algorithm is given in Fig. 8. Because of the restriction on the range of the interim sums and carries in step 2, step 3 can be performed carry free.

Fig. 9(a) shows the block structure of a signed digit adder (SDA), which performs the algorithm in Fig. 8. We refer to an SDA of base r as SD-r, where the base r is either 4, 8, 16, 32, 64, or 128. Fig. 9(b) lists the number of digits (k) and the number of bits per digit (b) for 32-b adders of different bases. A TC encoding is used for the digits and small RCA's of precision b are used to perform the digit addition in steps 1 and 3 of the SD addition algorithm (see Fig. 8). These additions are handled by the blocks labeled **DA I** and **DA II**, respectively. The block **COR** is used to perform step 2, i.e., correct the temporary sum p_i and generate u_i and c_i.

3.6 Carry Save Adder

Fig. 10 illustrates a carry save adder (CSA) for adding two CS numbers $A = ac_n ac_{n-1} \cdots ac_1 + as_{n-1} as_{n-2} \cdots as_0$ and $B = bc_n bc_{n-1} \cdots bc_1 + bs_{n-1} bs_{n-2} \cdots bs_0$, which calls for

TABLE II
ADDER CIRCUIT DESCRIPTION

Adder Type	*Area ($\times 10^6$ λ^2)*			*No. of transistors*			*Max transistor size (n/p)*		
	16-bit	*32-bit*	*64-bit*	*16-bit*	*32-bit*	*64-bit*	*16-bit*	*32-bit*	*64-bit*
RCA	0.40	0.80	1.60	596	1204	2420	3/3	3/3	3/3
MCC	0.48	0.96	1.90	642	1298	2610	4/4	4/4	4/4
CSK	0.82	1.62	3.22	682	1410	2866	4/4	4/4	4/4
VSK	0.81	1.76	3.44	706	1440	2900	3/3	3/3	3/3
CSL	0.76	1.45	2.75	914	1982	4128	5/7	6/10	8/13
CLA	1.14	2.27	4.55	1038	2132	4348	4/4	4/4	4/4
B&K	1.25	3.00	6.76	1072	2442	5444	3/3	3/3	3/3
ELM	1.08	2.36	5.38	892	2078	4752	3/5	5/7	8/12
CSA	1.05	2.03	3.90	1176	2360	4728	3/3	3/3	3/3
SD-4	1.36	2.71	5.41	1550	3166	6398	3/5	3/5	3/5
SD-8	1.11	2.42	4.61	1228	2812	5452	3/5	3/5	3/5
SD-16	1.09	2.17	4.32	1186	2490	5098	3/5	3/5	3/5
SD-32	0.97	2.25	4.16	1020	2572	4900	3/5	3/5	3/5
SD-64	1.12	2.23	4.07	1180	2530	4780	3/5	3/5	3/5
SD-128	1.27	2.11	3.78	1340	2364	4412	3/5	3/5	3/5

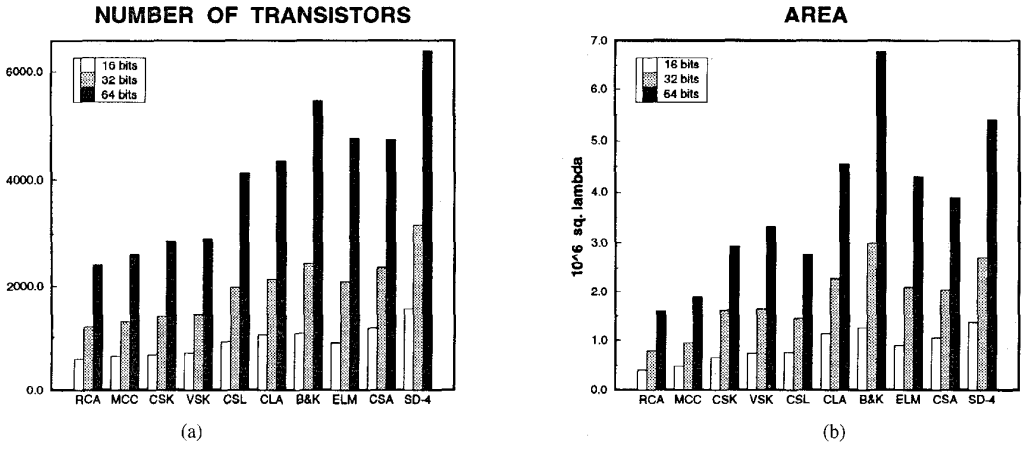


Fig. 13. (a) Number of transistors. (b) Area.

two levels of full adders. The designs of **FA** and **HA** are the same as in Fig. 2.

Finally, Table I summarizes the delay and area characteristics as well as the abbreviations used for the various adders.

IV. SOURCES OF POWER DISSIPATION IN CMOS

In our experiments, we used static CMOS (Complementary Metal-Oxide-Semiconductor), which is a popular logic style because of its high packing density, low power dissipation and high yield [26]. The three major sources of power dissipation in CMOS circuits are

$$\begin{aligned} P_{\text{total}} &= \text{switching power} + \text{short circuit power} + \text{leakage power} \\ &= p_f \cdot C_L \cdot V_{dd}^2 \cdot f + I_{sc} \cdot V_{dd} + I_{\text{leakage}} \cdot V_{dd}. \end{aligned} \tag{1}$$

The first term in (1) represents the switching component of power which is due to the charging and discharging of load capacitances, where p_f is the activity factor of the circuit, C_L is the load capacitance, V_{dd} is the supply voltage and f is the clock frequency. The switching power is the dominant term in a well designed circuit and it can be lowered by reducing any one or more of p_f, C_L, V_{dd} and f, while retaining the required speed and functionality. In our experiments, we vary the load capacitance by transistor sizing (circuit design style) and vary the activity factor of the circuit by using different types of adders. In this paper, we do not study the effects of varying the supply voltage and keep it fixed at 5 V. Since a quadratic improvement in power consumption may be obtained by lowering V_{dd}, many researchers have investigated the effects of lowering the supply and threshold voltages. Unfortunately, reducing the supply voltage reduces power, but the delays increase with the effect being more drastic at voltages close to the threshold voltage [9]. Liu and Svensson [16] have demonstrated that power reductions of about 40 times can be obtained without speed loss by using supply voltages down to about 0.48 V. However the threshold voltage also needs to be scaled and this requires highly advanced device technology and manufacturing strategies.

The second term in (1) is due to the short circuit current I_{sc}, which arises when both the n- and the p-transistors are on simultaneously for a short period of time during $1 \rightarrow 0$ or $0 \rightarrow 1$ transitions [26]. The longer the input transition time, the longer this period and hence more power is consumed. Therefore the short circuit power dissipation can be reduced by properly sizing and reordering transistors such that the output

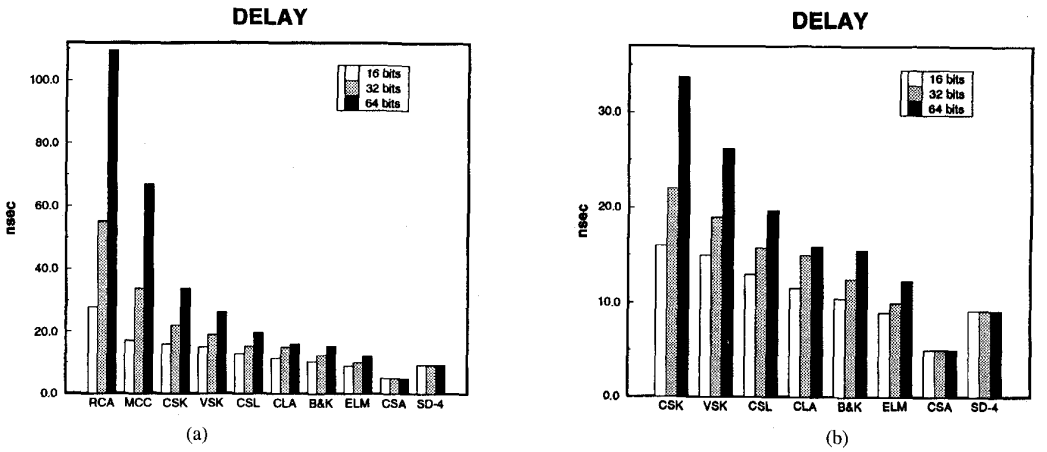


Fig. 14. (a) Delay. (b) Detail of the faster adders.

transition time is minimized [3]. In Section V, we describe our methodology for minimizing the short circuit power of a circuit.

When the inputs are steady, either the p- or the n-transistors (but not both) are on. However, there is some small reverse bias leakage current, I_{leakage}, between the diffusion regions and the substrate, represented by the third term in (1). This static power dissipation is negligible in static CMOS technology (1–2 nW in an inverter operating at 5 V) and is normally ignored, as is the case in this paper also.

V. Layout Generation

To generate layouts of all the adders, we decided to use an automatic layout generation tool since it would enable us to uniformly apply the same methodology to all the circuits. A level 'playing field' is essential for a fair comparison, and a hand layout would be too tedious. The next question was which automatic layout generator to use. Although it is obvious that circuits should be optimally sized, tools such as Timberwolf [24], which use standard cells tend to use oversized transistors in order to improve the delay. Thus they increase the load capacitance more than necessary, leading to excessive power dissipation. *Perflex* is an in-house performance driven, two dimensional, gate matrix CMOS layout generator [13]. Given a transistor level description of a circuit, *Perflex* can generate a layout, as shown in Fig. 11. *Perflex* uses cluster based simulated annealing to do the placement and routing, and has been shown to perform better than Timberwolf [13]. A typical 32-b or 64-b adder consists of several hundred gates and *Perflex* produces good (comparable to hand design) layouts for circuits with fewer than 100 gates. We therefore split up each adder into a number of modules with at most 50 gates each. *Perflex* was used to generate layouts of each module, which were then connected together using hand layout.

Perflex is a performance driven layout generator. It is capable of generating circuits with transistors sized just enough to satisfy some user-specified delay constraint. However, it is impossible to maintain a uniform delay constraint across different modules, which means that transistors in some modules may be sized larger than others affecting power consumption as well. Thus, it is not possible to maintain fairness and consistency by sizing for speed. On the other hand, it is possible to generate modules such that the power consumption of each module is minimized, by sizing the driver transistors sufficiently depending on the load. From Section IV, we know that this reduces the short circuit power. More detail regarding the optimal size of the driver transistors for different loads can be found in [3] and [4]. In addition, *Perflex* reorders transistors such that series transistors are arranged in the order of their input arrival times so that the latest-arriving input is assigned to the transistor closest to the output node.

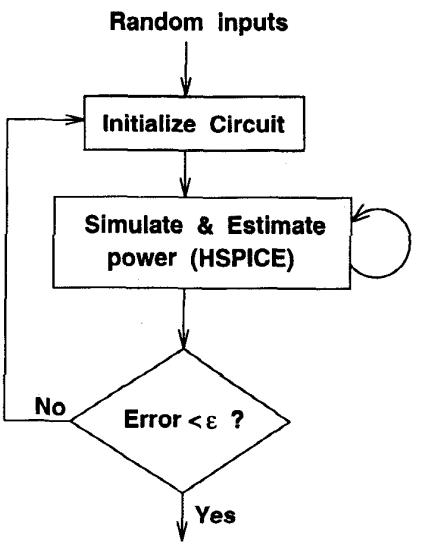


Fig. 15. Measuring average power dissipation.

The nature of the sizing entails that transistors be sized starting at the module outputs, working backward toward the inputs. The primary outputs of the module, therefore, are always minimum sized. When several modules are connected,

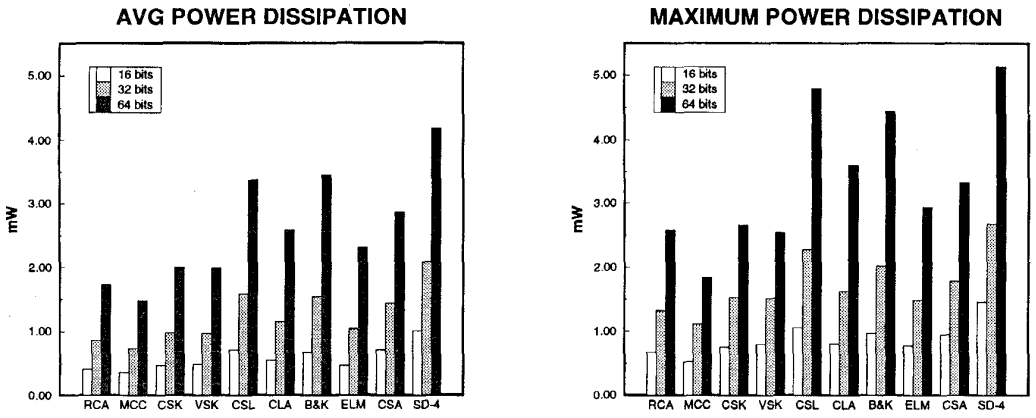


Fig. 16. (a) Average power dissipation per addition. (b) Maximum power dissipation per addition.

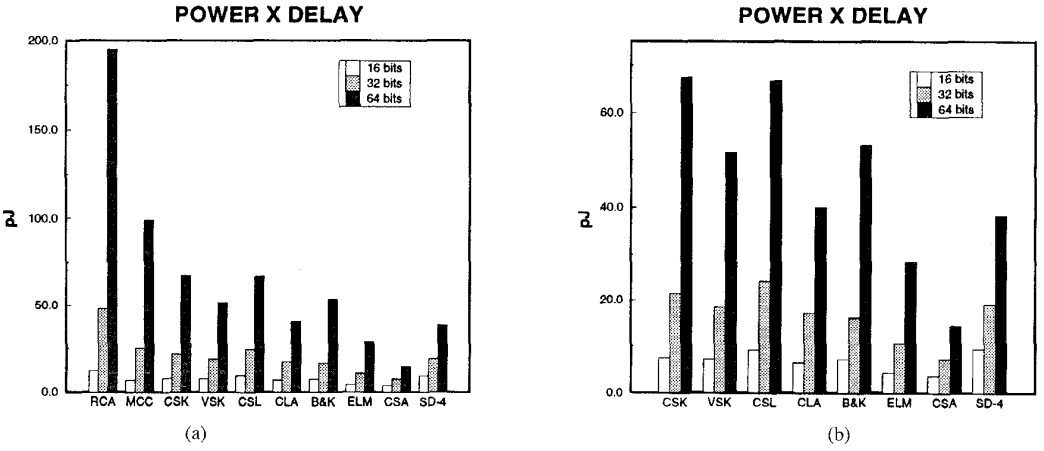


Fig. 17. (a) Power × delay. (b) A closer look at some adders.

these outputs may drive other gates. We found that as long as the fan-out is less than four, no further sizing is necessary and this is the case with all the linear time adders. However, in the case of the carry select and carry lookahead adders, additional sizing had to be done by hand in some cases.

Since our low-level sizing optimization criterion favors power rather than speed, the resulting circuit has smaller transistors, smaller layouts, shorter wires, and consequently lower capacitance. A 16-b ELM adder generated by *Perflex* operating in the power sizing mode described above was found to consume about 4% less power when compared to the same circuit with all unit-sized transistors (3λ). Less than 5% of the gates were sized and the maximum width of the p transistor was 5λ and that of the n transistor was 3λ.

For the sake of realism, all measurements in Section VI were taken with each input supplied through a driver consisting of two inverters in series and each output node driving a unit sized inverter load, as shown in Fig. 12. The drivers and the loads were included in the transistor description given as input to *Perflex* so that they featured in the power dissipation based transistor sizing.

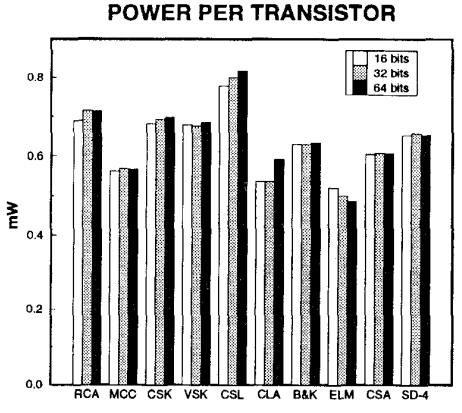


Fig. 18. Power consumed per transistor.

VI. Experiments

The experimental results described in this section were obtained using the extraction style parameters from MOSIS for hp1.2 micron scalable CMOS technology. A detailed transistor

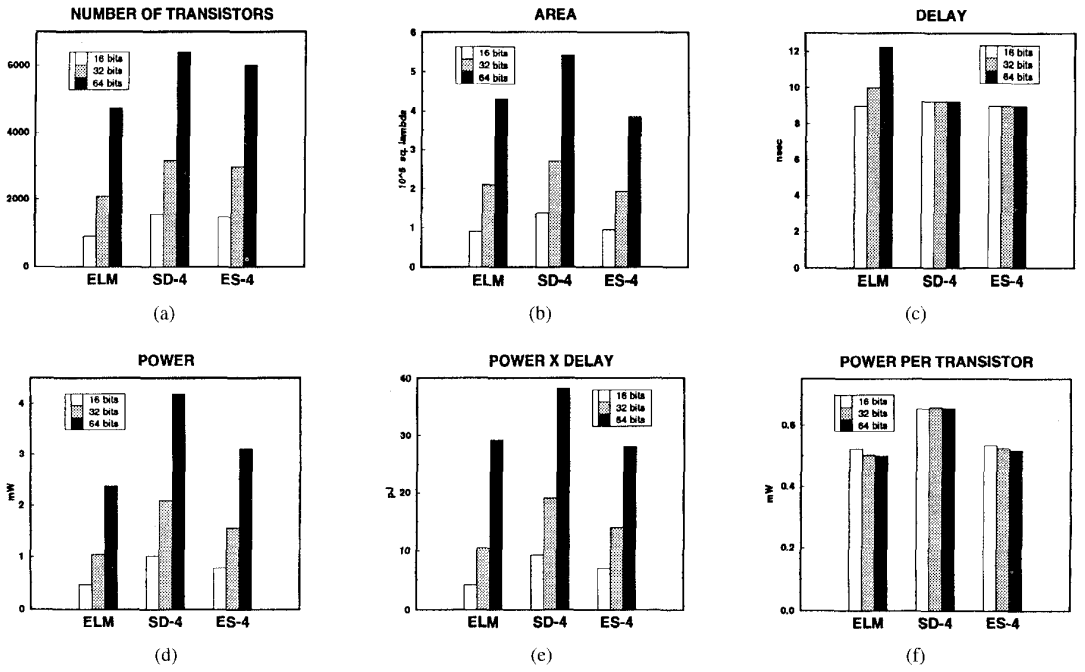


Fig. 19. Comparing the performance of ES-4 with ELM and SD-4.

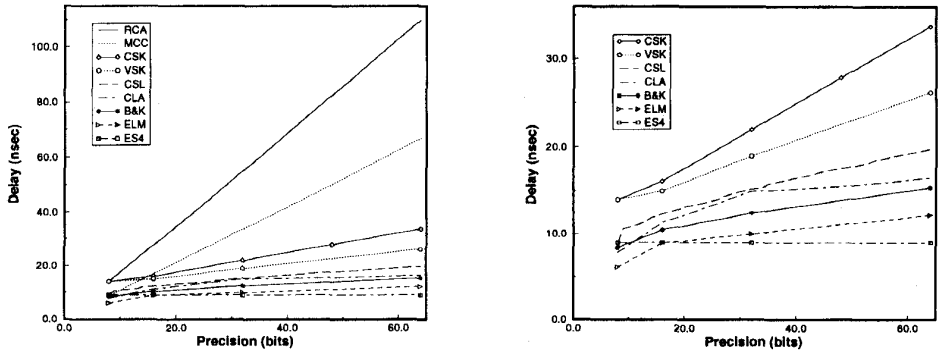


Fig. 20. (a) Variation of delay with precision. (b) Detail of the faster adders.

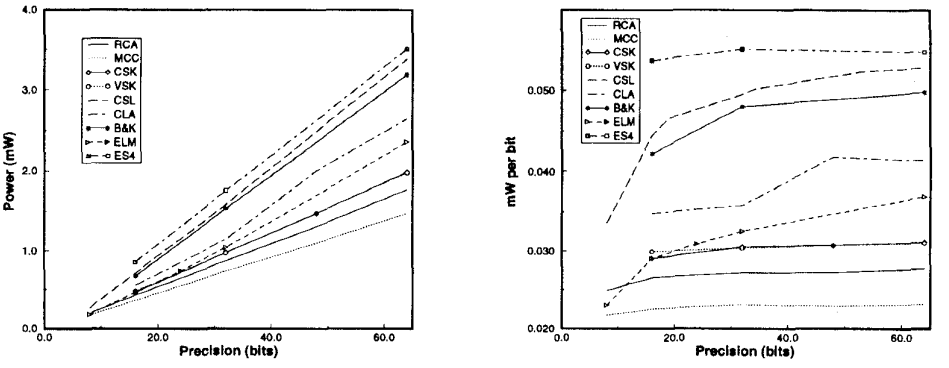
level simulation was done using HSPICE (version 93a) [18] to obtain the power and delay measures. Simulations were carried out at 27°C and a supply voltage of 5 V was used for all adders. The actual values for speed and power are inevitably dependent upon the technology and the style of implementation. Readers should be aware that the numbers presented in this paper should be looked at for their *relative*, not absolute, values. The controlled set of experiments provide insight into the behavior of different types of adders.

6.1 Adder Circuit Description

Table II summarizes the area, transistor count, as well as the maximum transistor size for each adder. The transistor count includes those contained in the drivers and loads as well. Fig. 13(a) and (b) portrays the number of transistors and the area of each adder in a graphical form. Although no serious attempts were made to absolutely minimize area, the numbers presented are a good indication of the relative areas of the adders which account not only for the transistors, but for the interconnections as well. For example, even though B&K has fewer transistors than SD-4, it has longer interconnects, which is reflected by its larger area.

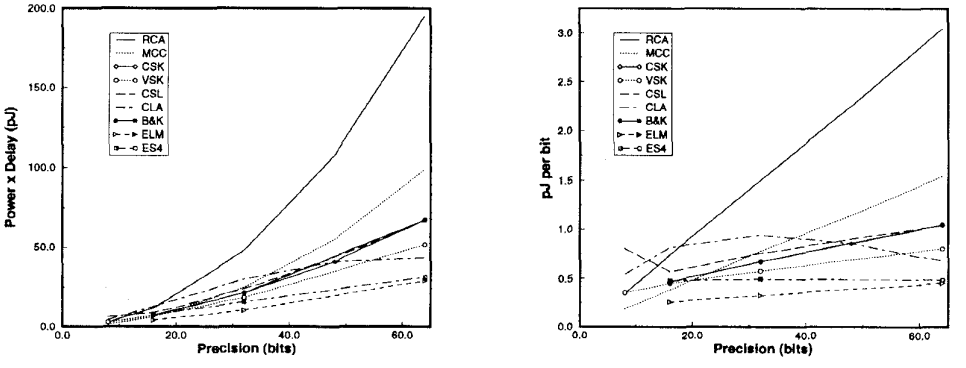
6.2 Delay

The delay of each adder was measured directly from the output waveforms generated by simulating the adder with HSPICE for the worst case inputs, that is, inputs which cause the carry to ripple from the least significant bit position to the most significant bit position. The RCA and the MCC are the simplest adders, but as Fig. 14(a) shows, they provide reasonable speed for small precisions only. The MCC is an attractive choice up to 16-b, since it occupies very little area and is fast. For faster addition at higher precisions, various schemes described in Section III must be adopted. The CSA



(a) (b)

Fig. 21. (a) Variation of power consumption with precision. (b) Power consumption per bit.



(a) (b)

Fig. 22. (a) Variation of power × delay. (b) Power × delay per bit with precision.

is the fastest adder, followed by the SD4. However, it must be remembered that they do not use the conventional two's complement number system, but are operating in a system using CS and SD base-4 representations, respectively. Among the adders using TC representation, the ELM proved to be the fastest.

6.3 Power Dissipation

The flow-chart in Fig. 15 illustrates our methodology for obtaining the average power dissipation of the adders. Each adder is presented with independent, pseudorandom inputs and the power consumed is monitored using HSPICE. Since the inputs are independent, power can be approximated to be *normally distributed* [6]. Hence, the mean power dissipation of the circuit is given by $(\overline{x} \pm t_c(\sigma/\sqrt{S}))$, where $\overline{x}$ is the sample mean, σ is the sample standard deviation, S is the number of samples and t_c is obtained from the t-distribution for $c\%$ confidence interval. The adders were simulated until the result was obtained within a 95% confidence interval and 3% error for the 16-b and 32-b adders and 5% error for the 64-b adders.

All adders were clocked at 10 MHz in order to accommodate the slowest adder. The power dissipation measures in Fig. 16 include the power consumed by the drivers and the loads. The average power consumed by a single driver driving a single load is about 57 μW. Fig. 16(b) gives the maximum value of the power dissipation, which is the maximum height of the power curve over all the inputs simulated.

VII. Some Performance Measurements

Below, we present two performance criteria, namely, power-delay product and power per transistor.

7.1 Power-Delay Product

Until a decade or two ago, area and speed were the primary concerns of the VLSI community. As silicon area became cheaper, the emphasis turned to low power consumption and real-time operation. The power-delay product, shown in Fig. 17, is a useful performance metric in present day signal processing and mobile computing units, which demand both low power consumption and high speed operation. This, however, does *not* mean that area is completely ignored. In a well designed circuit, larger area usually means more complicated circuits, which means more transistors or interconnects or both. This implies a higher load capacitance, hence more power dissipation. Thus area indirectly factors into the performance figures.

Being very simple, it is not surprising that the RCA consumes the least power; but it suffers a greater delay due to its long carry chain. The MCC is significantly better and at precisions less than 16-b, it is comparable to the CLA. However, it is evident from Fig. 17 that the linear

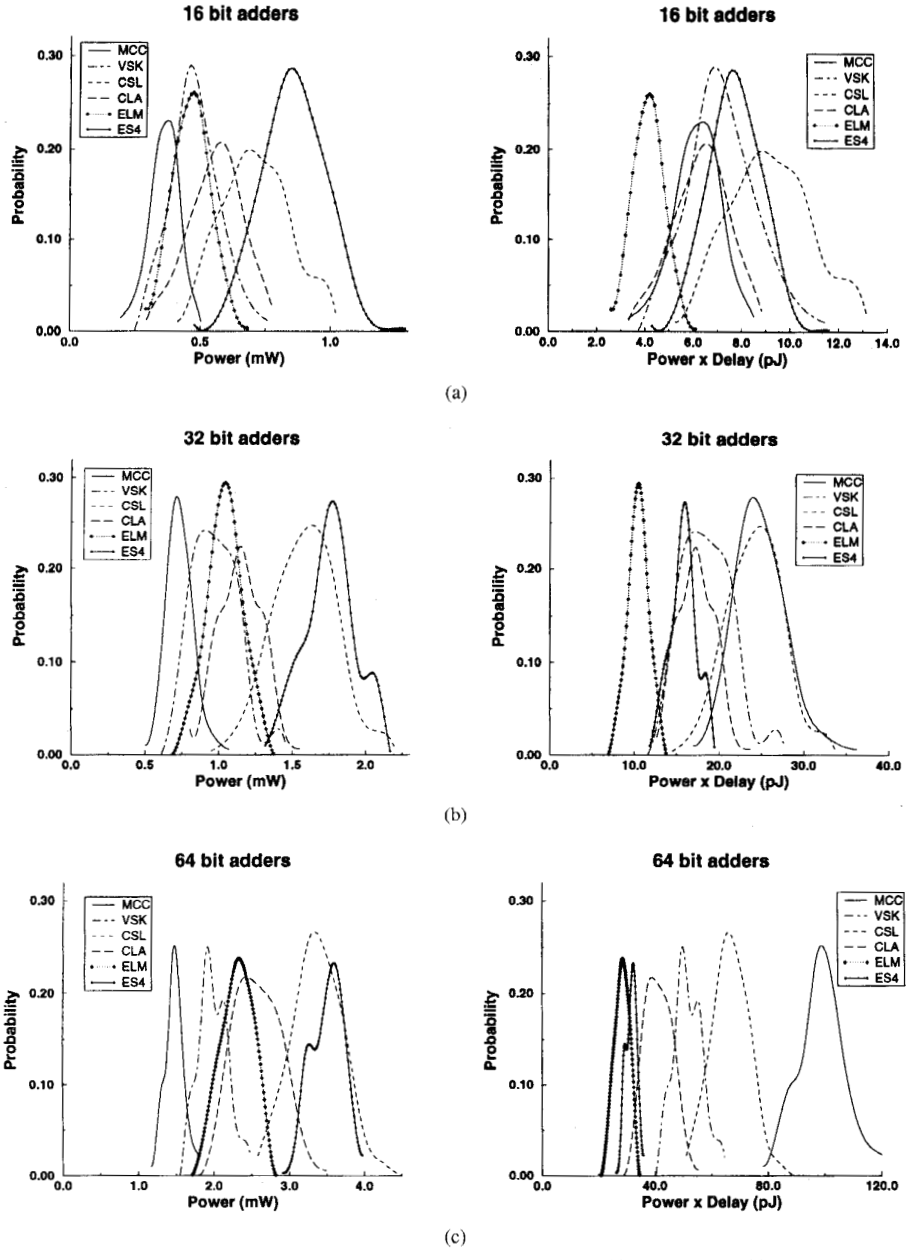


Fig. 23. Probability distribution of power and power × delay for (a) 16-b adders. (b) 32-b adders. (c) 64-b adders.

adders have very poor power-delay products, especially at higher precisions. Our paper shows that using variable sized groups of carry skip improves speed, but reduces power consumption only by a marginal amount. The VSK, by virtue of being faster, is superior to the CSK in terms of the power-delay product, the advantage being greater at higher precisions. The CSL is fast, but consumes a lot of power on account of the usage of two sets of adders (for carry-ins of zero and one) and multiplexers. Among the carry lookahead adders, ELM is the fastest and consumes the least power when compared to B&K. Both ELM and B&K have been designed specially from a VLSI layout point of view and an attempt is made to minimize the number of interconnections between modules. However the ELM uses fewer interconnects, only $(2n \log n + n)$ as opposed to $(3n \log n)$ in the B&K. Thus our experimental result reinforces the need for reducing interconnects in VLSI.

7.2 *Power per Transistor*

Fig. 18 was obtained by dividing the average power consumption in Fig. 16(a) by the number of transistors in Fig. 13(a) to get the average power per transistor. Notice that power per transistor for a particular adder varies very little with precision. The RCA is the most compact adder where the carries ripple through the adder, and its power per transistor is the highest. Although the MCC is also based on rippling of carries, the transistors used to compute

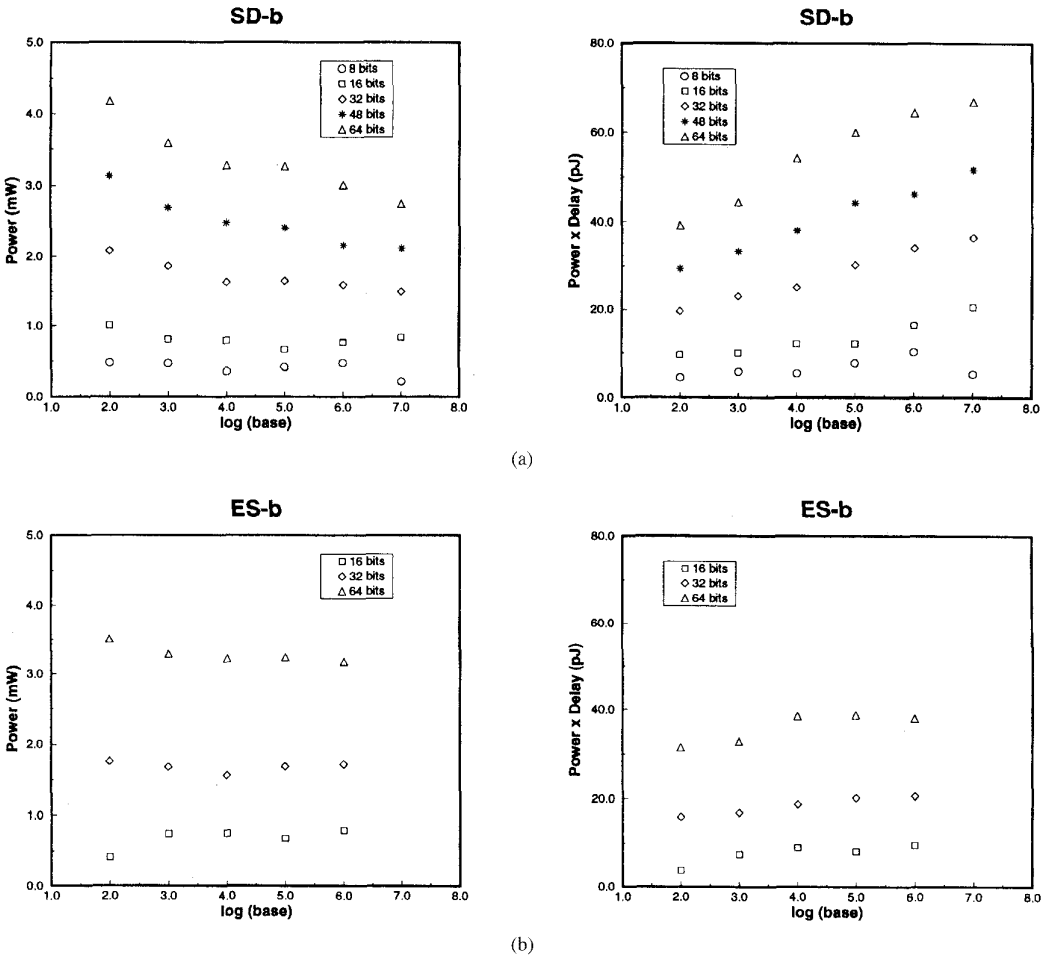


Fig. 24. Varying the base of the (a) SD-b. (b) ES-b.

p_i's, g_i's, and k_i's operate in parallel and switch from $0 \rightarrow 1$ or vice versa at most once. Very few transistors are in the ripple path. Hence, it has a significantly lower power per transistor compared to the RCA. The computation of the CLA is based on a tree structure with a high degree of parallelism and a correspondingly low power per transistor. The blocks in the carry skip adders are RCA's and that explains why their value is close to that of an RCA.

To summarize, in the adder designs with more parallelism, computation in the less significant part causes fewer gates or nodes to switch in the more significant part of the adder. This implies that the following two substitutions should give better power per transistor values.

1) Using ELM's instead of RCA's to implement **DA I** and **DA II** in SD-b and
2) using MCC's instead of RCA's in CSK and VSK.

The first substitution was done and the new SD-b is called ES-b, where b is the base. Fig. 19 compares the performance of ES-4 with ELM and SD-4. It can be seen that the substitution resulted in a superior adder compared to both the ELM and the SD-4. The modules **DA I** and **DA II** in ES-4 and SD-4 call for small 4-b and 3-b adders, respectively. We found that a 4-b (3-b) ELM has fewer transistors than a 4-b (3-b) RCA. (This is not true at higher precisions.) Hence ES-4 has fewer transistors and lower area compared to SD-4. Its delay improves only marginally, and as expected, most of the improvement is in the power consumption. Overall, it leads to a lower power-delay product than the SD-4 and is comparable to the ELM adder.

VIII. How Precision Affects Performance

Figs. 20, 21, and 22 show the how delay, power and power-delay product vary with precision.

It can be observed in Fig. 20 that at higher precisions, the difference in speed among the adders is significant. Both SD-4 and CSA are high speed constant time adders and only the ES-4 is shown for the sake of clarity. B&K and ELM exhibit similar behavior with the ELM adder being better for the entire range of precisions studied.

IX. Probability Distribution of Power

It was mentioned in Section VI-C that the power dissipation is obtained based on random inputs. The power consumed for

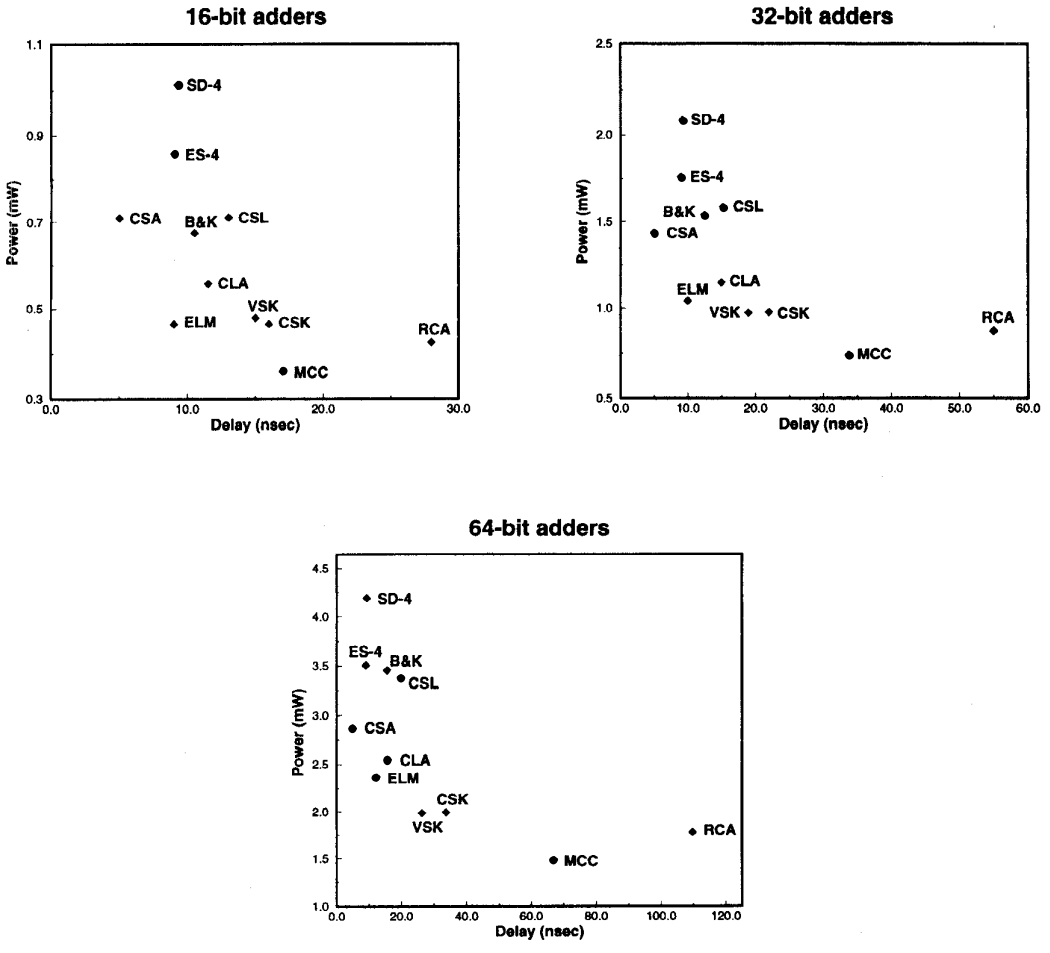


Fig. 25. Power versus delay.

each input was monitored and the probability distribution of the power consumption and power-delay product was obtained. For the sake of clarity the distributions for only a few selected adders are shown in Fig. 23.

X. Revisiting Signed Digit Adders

The signed-digit adders are of special interest in signal processing hardware because of their speed and suitability for digit-level pipelining [23]. But unfortunately their speed advantage does not come for free. They are logically more complicated than the other adders as evinced by the large number of transistors in Table II and hence consume more power. The larger the base of an SDA, the lesser the overhead. This is due to the fact that fewer digits are needed to represent an operand and hence fewer correction circuits are required. Accordingly, we found that the average power consumption of the SD-b decreased with increasing base, as shown in Fig. 24(a). But since the digit adders of SD-b are RCA's, the delay increases as the ripple carry chains become longer with increasing base. The result is an increase in the power-delay product. On the other hand, in the case of the ES-b's, where ELM adders are used, the power-delay product remains fairly constant [Fig. 24(b)].

XI. Adder Design Space

The power and delay values for 16, 32, and 64-b adders are summarized in Fig. 25 in the form of a large design space, where the faster adders lie to the left of the figure and the less power hungry adders lie toward the bottom of the figure.

XII. Conclusion

In this paper, we have compared several types of parallel adders based on various performance criteria such as, area, delay, power and power-delay product. With the rising popularity of mobile computing, the emphasis on low power designs has increased. In such cases, "use a design that is fast enough and consumes the least power" is a better rule of thumb than "use the fastest design." A uniform static CMOS layout strategy is used for all adders, whereby the short circuit power consumption is minimized. All the adders have been simulated using HSPICE and the extraction style parameters for hp1.2 μ scalable CMOS technology. Using the ES-4 as an example, we have shown how a design can be improved based on the power per transistor ratio. The ELM adder was found to be the best among the two's complement adders for the entire range of precision studied.

REFERENCES

[1] D. E. Atkins, "Introduction to the role of redundancy in computer arithmetic," *Computer*, pp. 74–77, June 1975.

[2] A. Avizienis, "Signed-digit number representation for fast parallel arithmetic," *IRE Trans. Electron. Comput.*, p. 389, 1961.

[3] M. Borah, M. J. Irwin, and R. M. Owens, "Minimizing power consumption of static CMOS circuits by transistor sizing and input reordering," in *VLSI Design '95*, Jan. 1995, pp. 294–298.

[4] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint," in *1995 Int. Symp. Low Power Design*, Apr 1995, pp. 167–172.

[5] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, pp. 260–264, Mar. 1982.

[6] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 63–71, Mar. 1993.

[7] T. K. Callaway and E. E. Swartzlander, Jr., "Estimating the power consumption of CMOS adders," in *Proc. 11th Symp. Comp. Arithmetic*, June 1993, pp. 210–219.

[8] P. K. Chan and M. D. F. Schlag, "Analysis and design of CMOS manchester adders with variable carry-skip," *IEEE Trans. Comput.*, vol. C-39, pp. 983–992, Aug 1990.

[9] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low power CMOS digital design," *IEEE J. Solid-State Circuits*, Apr 1992, pp. 685–691.

[10] D. C. Chen, L. M. Guerra, E. H. Ng, M. Potkonjak, D. P. Schultz, and J. M. Rabaey, "An integrated system for rapid prototyping of high performance algorithm specific data paths," in *Proc. Application Specific Array Processors*, Aug 1992, pp. 134–148.

[11] J. Hennesey and D. A. Patterson, *Computer Architecture—A Quantitative Approach*. San Mateo, CA: Morgan Kaufmann, 1990. Appendix A.

[12] T. P. Kelliher, R. M. Owens, M. J. Irwin, and T.-T. Hwang, "ELM—A fast addition algorithm discovered by a program," *IEEE Trans. Comput.*, vol. 41, Sept. 992.

[13] S. Kim, "CMOS VLSI layout synthesis for circuit performance," Ph.D. dissertation, The Penn State Univ., University Park, 1992.

[14] I. Koren. *Computer Arithmetic Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993, ch. 5.

[15] M. Lehman and N. Burla, "Skip techniques for high-speed carry-propagation in binary arithmetic circuits," *IRE Trans. Electron. Comput.*, pp. 691–698, Dec. 1961.

[16] D. Liu and C. Svensson, "Trading speed for low power by choice of supply and threshold voltages," *IEEE J. Solid-State Circuits*, vol. 28, pp. 10–17, Jan. 1993.

[17] C. Mead and L. A. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.

[18] Meta-Software. *HSPICE User's Manual version H92*. Campbell, CA, 1992. Meta-Software, Inc., 1300 White Oaks Rd., Campbell, CA 95008.

[19] C. Nagendra, U. K. Mehta, R. M. Owens, and M. J. Irwin, "A comparison of the power-delay characteristics of CMOS adders," in *Int. Workshop Low Power Design*, Apr. 1994, pp. 231–236.

[20] C. Nagendra, R. M. Owens, and M. J. Irwin, "Low power tradeoffs in signal processing hardware primitives," in *IEEE Workshop VLSI Signal Processing*, Oct. 1994, pp. 276–285.

[21] ——, "Power-delay characteristics of CMOS adders," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 377–381, Sept. 1994.

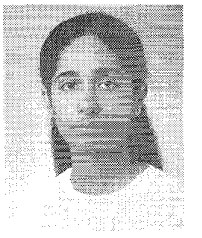
[22] ——, "Unifying carry-sum and signed-digit number representations for low power," in *1995 Int. Symp. Low Power Design*, Apr. 1995, pp. 15–20.

[23] ——, "Digit systolic algorithms for fine-grain architectures," in *Proc. Application Specific Array Processors*, Oct 1993, pp. 466–477.

[24] C. Sechen and A. L. Sangiovanni-Vincentelli, "The Timberwolf placement and routing package," in *Proc. Custom Integrated Circuit Conf.*, May 1984, pp. 522–532.

[25] S. Waser and M. J. Flynn. *Introduction to Arithmetic for Digital Systems Designers*. New York: Holt, Rinehart and Winston, 1982.

[26] N. H. E. Weste and K. Eshraghian. *Principles of CMOS Logic Design, A Systems Perspective*. Reading, MA: Addison-Wesley, 1988.



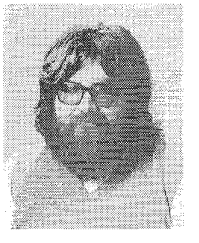
Chetana Nagendra (M'95) received the B.E. degree in computer science and engineering from the BMS College of Engineering, Bangalore University, India, in 1990 and the M.S. and Ph.D. degrees in computer science from The Pennsylvania State University, University Park, in 1992 and 1996, respectively.

She is currently with the California Microprocessor Division of Advanced Micro Devices. Her research interests include computer architecture, low power design, and computer arithmetic.

Mary Jane Irwin (F'94) received the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana-Champaign, in 1975 and 1977, respectively.

She is a Professor of computer science and engineering at The Pennsylvania State University, University Park. She has authored more than 125 scholarly works. Her primary research interests include computer architecture, the design of application specific VLSI processors, high speed computer arithmetic, and VLSI CAD tools.

Dr. Irwin is on the executive committees of the Design Automation Conference and the Supercomputing Conference, as well as on the editorial board of *The Journal of VLSI Signal Processing*, and the IEEE TRANSACTIONS ON COMPUTERS, and is an elected member of the Computing Research Board, the IEEE Computer Society Board of Governors, and the ACM Council. She is the principal investigator of a Small Scale Institutional Infrastructures grant from NSF.



Robert Michael Owens (M'95) received the M.S. degree in computer science from the Virginia Polytechnic Institute and State University, Blacksburg, in 1977 and the Ph.D. degree in computer science from The Pennsylvania State University, University Park, in 1980.

He is presently a Professor of computer science and engineering at Penn State. Before joining Penn State, he was with IBM and the Naval Surface Weapons Center. He has authored more than 100 scholarly works. His research interests include computer architecture, massively parallel computing, VLSI architectures and the CAD tools associated with their implementations. He is the author of UREP, a computer communication package which has been distributed to hundreds of locations worldwide.

Dr. Owens is on the IEEE Signal Processing Technical Committee on VLSI and served as program Co-Chair for the ASAP Conference in 1994.