Power-Delay Characteristics of CMOS Adders

Chetana Nagendra, Robert Michael Owens, and Mary Jane Irwin

Abstract—An approach to designing CMOS adders for both high speed and low power is presented by analyzing the performance of three types of adders—linear time adders, log\textsuperscript{N} time adders and constant time adders. The representative adders used are a ripple carry adder, a blocked carry look-ahead adder and several signed-digit adders, respectively. Some of the tradeoffs that are possible during the logic design of an adder to improve its power-delay product are identified. An effective way of improving the speed of a circuit is by transistor sizing which unfortunately increases power dissipation to a large extent. It is shown that by sizing transistors judiciously it is possible to gain significant speed improvements at the cost of only a slight increase in power and hence a better power-delay product. Perplex, an in-house performance driven layout generator, is used to systematically generate sized layouts.

Index Terms—Power-delay product, static CMOS adders, transistor sizing.

I. INTRODUCTION

The three most widely accepted metrics for measuring the quality of a circuit are area, delay and power. Minimizing area and delay has always been considered important, but reducing power consumption has been gaining prominence recently [8], [7], [6], [5]. This can be attributed to the increasing popularity of mobile communication systems. Since dramatic improvements in battery technology are not foreseen, low-power designs are crucial not only to lighten the overall weight, but also to reduce the time between charges. On the other hand, with advances in CMOS technology and reduction of the feature size, area is no longer as scarce a resource as it once used to be. Portability imposes a strict limitation on power dissipation while still demanding high computational speeds as required by real-time tasks.

Manuscript received December 7, 1993; revised February 24, 1994 and May 10, 1994. This work was supported in part by NSF Grant MIP-9102500.

The authors are with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 USA.

IEEE Log Number 9403170.

Hence we use the power-delay product as the metric of performance in this paper and downplay the importance of the area occupied by the circuit.

Chandrasekaran et. al. describe the four degrees of freedom available in the design of low-power circuits and systems, namely technology, circuit design styles, architecture and algorithms [8]. One popular technology family is static CMOS since it has large noise margins and consumes the least power in its class [9]. The dynamic power consumption of a CMOS gate is given by \( P = p_f C_f V_s^2 f \), where \( p_f \) is the activity factor, \( C_f \) is the load capacitance, \( V_s \) is the supply voltage and \( f \) is the clock frequency. In this paper, we assume that the supply voltage is fixed at 5 V and clock all adders at the same frequency. We attempt to lower the load capacitance and the activity factor of the circuit by playing with the circuit design styles (transistor sizing) and addition algorithms, respectively. We compare the power consumed and the delay of adders using three different addition algorithms, namely, ripple carry addition, blocked carry lookahead addition and signed-digit addition. The choice of the addition algorithm significantly affects \( p_f \), the activity factor of a circuit [8].

Reference [6] compares the power consumption of some CMOS adders. But they do not size transistors whereas practical circuits usually employ transistor sizing to improve speed. Further, estimating the power consumption of signed-digit adders has not been done before.

The remainder of this paper is organized as follows. Section II presents a brief description of the three adder topologies. An overview of the layout generation of the adder circuits using Perplex, a performance driven module generator, is given in Section III. In Section IV, the experiments conducted to estimate the energy and the delay of the adders are described along with the performance numbers and we conclude in Section V.

II. ADDER TOPOLOGIES

We have chosen adders with a wide spectrum of timing and complexity which makes it interesting to compare their performance in terms of power and power-delay product. The adders range from the simple but slow (linear time) ripple carry adder to the fairly complex but extremely fast (constant time) signed-digit adders. The third type of adder is the \( O(\log N) \) time blocked carry lookahead adder. The \( N \)-bit operands in the ripple carry and blocked carry lookahead adders are represented in \( 2^w \)'s complement format and the output is generated in the same format. The operands in the signed-digit adder are, of course, signed-digit.

A. Ripple Carry Adder

The basic unit of a ripple carry adder (RCA) is a Full Adder (FA) which computes a sum bit and a carry bit: \( s_i = x_i \oplus y_i \oplus c_i \) and \( c_{i+1} = x_i y_i + x_i c_i + y_i c_i \). That is, it adds the two operand bits with the incoming carry bit to produce a sum bit and an outgoing carry bit. Since, in the worst case, the carry can propagate from the least significant bit position to the most significant bit position, the addition time of an \( N \)-bit RCA is \( O(N) \).

B. Blocked Carry Lookahead Adder

We use the structure of the "ELM" blocked carry lookahead adder (BCLA) [14]. Compared to the Brent and Kung adder [4], the ELM adder occupies the same area \( O(\sqrt{N} \log N) \), but uses fewer
interconnects $(2N \log N + N$ as opposed to $3N \log N$). The ELM adder makes use of a binary tree of simple processors to perform addition in $O(\log N)$ time. The leaves of the tree compute partial sums and pass them on to the next level. The nodes at the higher levels of the tree receive partial sums as well as the generate and propagate information necessary to update the partial sums at that level of the tree.

C. Signed Digit Adder

Signed-digit (SD) representations [1] are positional number representations with a constant radix $r \geq 3$ in which the individual SD's $\in \{-a, \ldots, -1, 0, 1, \ldots, a\}$ where:

$$\frac{1}{2}(r_0 + 1) \leq a \leq r_0 - 1,$$

for odd indices $r_0 \geq 3$

$$\frac{1}{2}r_0 + 1 \leq a \leq r_0 - 1,$$

for even indices $r_0 \geq 4$

In this paper we consider only those bases which are powers of 2. The use of SD numbers allows addition to be performed in constant time by restricting carry propagation to at most one digit position [2]. Each SD is encoded using a 2's complement encoding, where each SD occupies $n = \lceil \log r \rceil + 1$ bits. To represent an $N$-bit number, $k = \lceil N/(n - 1) \rceil$ SD's are required. The SD addition algorithm is given in Table 1 [2].

Fig. 1 shows the general structure of a base-$2^{n-1}$ SD adder (denoted as SDA-$2^{n-1}$) and its modules. Step 1 of the SD addition algorithm involves the sign extended addition of two $n$-bit SD's. The advantage of using a 2's complement encoding is that the digit addition can be done using a simple $n$-bit RCA, namely module A1. The output at this stage lies between $-2r + 2$ and $2r - 2$ and module COR is used to correct the output and generate the interim sum digit and the 2-b carry such that the conditions in step 1 are satisfied. Since the carry can only be $+1, 0$ or $-1$, it is encoded using two bits $inc$ and $dec$ (increment and decrement respectively). Thus step 2 of the SD addition algorithm calls for just an increment/decrement circuit, namely module A2. The longest path in the SDA-$2^{n-1}$ is marked DA (Digit Adder) in Fig. 1(a).

III. LAYOUT STYLES AND ADDER LAYOUT GENERATION

A few alternatives were considered to produce the adder layouts. Hand design usually results in smaller layouts which can be optimized for performance but can be very tedious and time consuming. On the other hand, automatic layout generation tools result in a faster turn-around time but seldom match the hand layouts in area or performance. But a comparison based on a level 'playing field' was a prime consideration for a controlled set of experiments and hence we decided to use the latter option. Among the automatic layout generation tools, we had to choose between standard cell place and route tools (for example, Timberwolf-SC4 [18]) and Perflex, an in-house performance driven module generator [15, 16]. Experimental results showed that Perflex can significantly improve the timing of a circuit while the layout area is kept comparable to that obtained when the layout is optimized for area. A 286-transistor 8-b adder

description based on gates from the MinII standard-cell library [3] was given as input to both Timberwolf-SC4 and Perflex. HSPICE [17] simulations showed that the layouts generated by Perflex were better in all respects of maximum signal delay, layout area and power consumption. Hence we decided to use Perflex.

Perflex uses a new flexible CMOS layout style that supports sizing of individual transistors to generate fast static combinational CMOS circuit modules. The techniques used for optimizing timing are transistor sizing, transistor reordering and reduction of wiring capacitances along the critical paths. All the three are performed in close interaction with a simulated annealing layout process and transistors are sized only when necessary for improving the speed and just enough to satisfy the slack constraints. This increases the load capacitance to a minimum extent and therefore the increase in power consumption is much less than in the case of standard cells.

A. Transistor Reordering

Transistor reordering is a simple, yet effective timing optimization technique which is achieved by arranging transistors connected in
Table II: A comparison of two 8-b SDA-4s (unit sized inverter load).

<table>
<thead>
<tr>
<th></th>
<th>Sized Full DA</th>
<th>Sized parts DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max transistor size (m²/ft²)</td>
<td>35/21 λ</td>
<td>5/5 λ</td>
</tr>
<tr>
<td>Routing area</td>
<td>279 x 23832 λ²</td>
<td>268 x 16783 λ²</td>
</tr>
<tr>
<td>Average power dissipation</td>
<td>109.994 mW</td>
<td>70.066 mW</td>
</tr>
<tr>
<td>Max instantaneous dissipation</td>
<td>33.729 mW</td>
<td>29.418 mW</td>
</tr>
<tr>
<td>Worst case delay</td>
<td>12.655 ns</td>
<td>12.625 ns</td>
</tr>
<tr>
<td>Power x Delay</td>
<td>1.393142 x 10⁻⁹</td>
<td>0.894578 x 10⁻⁹</td>
</tr>
</tbody>
</table>

series in the order of their input arrival times so that the latest arriving input is assigned to the transistor closest to the output node. Our experiments showed that reordering can not only improve the speed of a gate, but can decrease power consumption as well. Reordering in Perfex is carried out once prior to the layout and whenever transistors are sized. The 3-input NOR gate example in Fig. 2 illustrates the effects of reordering.

B. Transistor Sizing

The transistor sizing algorithm used in Perfex is the posynomial programming approach used in AT&T's TILOS [12]. But in the TILOS heuristic, transistors are sized only as a preprocessing step to the layout process whereas in Perfex sizing is done repeatedly as the simulated annealing layout optimizer slowly solidifies the gate placement to the final one. This allows transistors to be sized more optimally.

Although increasing the transistor size improves the speed of the circuit, it also increases power dissipation since the load capacitance increases. Using Perfex, we generated two types of sized layouts of an 8-b SDA-4 and observed their performance. In the first type of adder, a logical description of a single DA of SDA-4 was given as input to Perfex. Since the critical path of a SDA lies entirely within a single DA, optimizing the performance of the DA is sufficient to optimize the performance of the entire adder. This "sized full DA" was then used to construct an 8-b SDA-4.

In the second type of adder, each basic module comprising the DA, namely, HA.FA.HAS and COR, was sized individually using Perfex. These modules were then interconnected with hand layout to generate "sized parts DA". The results in Table II show that the 8-b SDA-4 built using the latter DA exhibited almost no loss in speed and consumed far less power giving a superior power-delay product. This is because the maximum size of a transistor in the "sized full DA" is much larger than that in the "sized parts DA". However this extra sizing produces negligible speed improvement (0.003%) which does not justify the cost of increasing power dissipation by 56%.

Hence our layout strategy for optimizing the power-delay product of the adder is to optimize the individual modules separately using Perfex which sizes the transistors only as much as necessary. Then the modules are interconnected by hand for a compact layout. Since the modules are automatically generated, most of the tedium of hand layout is avoided. Fig. 3 shows an example of a layout generated in this manner.

IV. EXPERIMENTAL RESULTS

Layouts of 32-b versions of the adders presented in Section II were generated in 1.2μ static CMOS technology as described in Section III. Table III lists their salient features. These layouts were then used to extract circuit equivalents for use in a detailed circuit simulation using HSPICE v. 9.3a [17] to obtain the power and delay measures. The transistor parameters used were from a recent 1.2μ MOSIS fabrication run and the simulations were carried out at 27°C with an input frequency of 10 MHz. All measurements 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 (see Fig. 4).

The delay of each adder was measured directly from the output waveforms by presenting the adder with inputs that caused the maximum carry ripple. It can be seen from Table III that the RCA suffers a greater delay due to its long carry chain. Among the SDA's, SDA-4 has the least delay and the delay of the SDA increases with the base since the digit addition carry chains become progressively longer.

The estimation of power dissipation of a circuit is a difficult problem and has received a lot of attention [5], [10], [11], [13]. HSPICE can measure the power consumed by a circuit given a set of inputs. But the power dissipation is a strong function of the inputs.
### TABLE III

<table>
<thead>
<tr>
<th>Adder Type</th>
<th>No. of transistors</th>
<th>Area ($\mu m^2$)</th>
<th>Worst case delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCA</td>
<td>884</td>
<td>$183 \times 3145$</td>
<td>63.4</td>
</tr>
<tr>
<td>BCLA</td>
<td>2228</td>
<td>$792 \times 1200$</td>
<td>32.7</td>
</tr>
<tr>
<td>SDA-64</td>
<td>2184</td>
<td>$288 \times 4956$</td>
<td>26.4</td>
</tr>
<tr>
<td>SDA-32</td>
<td>2236</td>
<td>$288 \times 4655$</td>
<td>23.1</td>
</tr>
<tr>
<td>SDA-16</td>
<td>2184</td>
<td>$288 \times 4853$</td>
<td>19.7</td>
</tr>
<tr>
<td>SDA-8</td>
<td>2500</td>
<td>$288 \times 5568$</td>
<td>16.2</td>
</tr>
<tr>
<td>SDA-4</td>
<td>2904</td>
<td>$288 \times 6489$</td>
<td>12.6</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Adder</th>
<th>Mean power dissipation per addition (mW)</th>
<th>Power x Delay ($\times 10^{-8}$)</th>
<th>Max instantaneous power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCA</td>
<td>$149.96 \pm 2.148$</td>
<td>$9.56 \pm 0.1362$</td>
<td>$72.727$</td>
</tr>
<tr>
<td>BCLA</td>
<td>$300.162 \pm 3.293$</td>
<td>$9.830 \pm 0.1078$</td>
<td>$74.739$</td>
</tr>
<tr>
<td>SDA-64</td>
<td>$298.145 \pm 3.924$</td>
<td>$7.67 \pm 0.1009$</td>
<td>$88.665$</td>
</tr>
<tr>
<td>SDA-32</td>
<td>$310.312 \pm 3.791$</td>
<td>$7.168 \pm 0.0875$</td>
<td>$87.425$</td>
</tr>
<tr>
<td>SDA-16</td>
<td>$306.772 \pm 3.756$</td>
<td>$6.643 \pm 0.0740$</td>
<td>$82.170$</td>
</tr>
<tr>
<td>SDA-8</td>
<td>$355.903 \pm 3.780$</td>
<td>$5.751 \pm 0.0604$</td>
<td>$97.830$</td>
</tr>
<tr>
<td>SDA-4</td>
<td>$409.055 \pm 3.677$</td>
<td>$5.162 \pm 0.0693$</td>
<td>$103.21$</td>
</tr>
</tbody>
</table>

The node voltages of each adder were initialized by supplying the adder with zero inputs. Then each adder was presented with 500 independent, pseudorandom inputs and the power consumed was monitored. The power dissipated during the initialization phase was not considered. Since the propagation delay of the slowest adder is approximately 63 ns, each addition was allowed a time of 100 ns for the voltages to stabilize before the next addition. The power dissipation measures include the power consumed by the drivers and the loads. The average power consumed by a single driver driving a single load is about 0.8165 mW. For example, a 32-b SDA-4 has 48 pairs of inputs and 48 outputs and therefore requires 96 drivers and 48 loads.

The height of the power curve was recorded as the maximum instantaneous power dissipation of the circuit. Since the inputs are independent, power can be approximated to be normally distributed [5]. Hence the mean power dissipation of the circuit is given by $\bar{x} \pm t_c \frac{\sigma}{\sqrt{N}}$, where $\bar{x}$ is the sample mean, $\sigma$ is the standard deviation, $N$ is the number of samples and $t_c$ is obtained from the $t$-distribution for $95\%$ confidence interval [19]. (For $95\%$ confidence and $N = 500$, $t_c = 1.9647$.) The power-delay product, i.e. (power consumed per addition x worst case delay), is computed in a similar way. The power measures for the seven adders are shown in Table IV. Figs. 5 and 6 show the probability distribution of the power dissipation and the power-delay product, respectively.

The RCA has approximately one third the number of transistors contained in either a BCLA or a SDA. Because of its simplicity, it is not surprising that it exhibits the lowest average and maximum instantaneous power dissipations. The SDA's may be very fast; but unfortunately their speed advantage does not come for free. They are logically more complicated than the RCA's and consume a lot more power. But they are clearly superior in the power-delay product since their speed advantage over the RCA is sufficient to overcome even their higher power consumption. However, the BCLA fails to achieve the same performance level.

Among the SDA's, the larger the base, the fewer the number of digit adders. Thus with larger bases there is a saving in the logic since fewer sign bits and corrections have to be handled. Therefore we observe a reduction in the power consumption with increasing base. It can be observed from Table IV as well as the power-delay histogram of Fig. 6 that the reduction in power is not enough to overcome the increase in delay. Finally, Fig. 7 shows a plot of the sample mean power dissipation versus the delay of the adders.

### V. Conclusion

In this paper, we have compared the power-delay product of three types of adders, namely, RCA, BCLA and SDA. We have also shown a design methodology wherein, by sizing the transistors only to the extent necessary, we can derive the speed benefits of sizing without increasing power dissipation to a large extent. The SDA's exhibit very good power-delay products compared to the RCA and BCLA. Although higher bases reduce power consumption, the delay of the
adder is more due to longer ripple carry chains in the digit adders. The reduction in power dissipation is not sufficient to balance the increase in delay when the digit addition is done using RCA’s. Hence we observe the phenomenon of increasing power-delay product with the base of the SDA. We are investigating the design of faster DA’s by using simple BCLA’s instead of RCA’s to perform signed-digit addition.

REFERENCES