A Hybrid Radix-4/Radix-8 Low Power, High Speed Multiplier Architecture for Wide Bit Widths

Brian S. Cherkauer[1] and Eby G. Friedman[2]

[1]Intel Corporation
2200 Mission College Blvd.
Santa Clara, California 95052

[2]Department of Electrical Engineering
University of Rochester
Rochester, New York 14627

Abstract – A hybrid radix-4/radix-8 architecture targeted for high bit multipliers is presented as a compromise between the high speed of a radix-4 multiplier architecture and the low power dissipation of a radix-8 multiplier architecture. In this hybrid radix-4/radix-8 multiplier architecture, the performance bottleneck of a radix-8 multiplier, the generation of three times the multiplicand for use in generating the radix-8 partial product, is performed in parallel with the reduction of the radix-4 partial products rather than serially, as in a radix-8 multiplier. This hybrid radix-4/radix-8 multiplier architecture requires 13% less power for a 64 x 64 bit multiplier, and results in only a 9% increase in delay, as compared with a radix-4 implementation. When supply voltage is scaled such that all multipliers exhibit the same delay, the 64 x 64 bit hybrid radix-4/radix-8 multiplier dissipates less power than either the radix-4 or radix-8 multipliers. The hybrid radix-4/radix-8 architecture is therefore appropriate for those applications that must dissipate minimal power and operate at high speeds.

I. Introduction

High speed multipliers are fundamental elements in signal processing and arithmetic based systems. The higher bit widths required of modern multipliers provide the opportunity to explore new architectures which would be impractical for smaller bit width multiplication. Architectures for circuit elements historically were designed to operate at maximum speed, notwithstanding the resulting power dissipation. Recently, greater emphasis has been placed on reducing the power dissipation of important circuit functions while maintaining these high speeds. Therefore, power dissipation as well as circuit speed should be considered at the architectural level.

A leveling off of the power factor, the power dissipated per bit^2•Hz, and hence the power efficiency, has recently been observed [1]. This leveling of the power factor is illustrated in Figure 1. This trend leads to the conclusion that to further improve the power efficiency of multipliers, power dissipation must be addressed at the architectural level as well as at the circuit level.

The data in Figure 1 present the power factors for a number of recent multiplier implementations. Sharma *et al.* utilized Booth radix-4 encoding along with a reduction array of carry save adders (CSAs) generated by a recursive algorithm to produce the 16 x 16-bit multiplier in [2]. In [3], Yano *et al.* introduced the complementary pass-transistor logic family (CPL) and implemented a 16 x 16-bit multiplier in CPL which used no encoding but did use a Wallace tree for partial product reduction. Nagamatsu *et al.* presented a 32 x 32-bit multiplier in which Booth radix-4 was used to generate the partial products and a tree of 4:2 counters was used to reduce these partial products [4]. Mori *et al.* designed a 54 x 54-bit multiplier similar in structure to that of [4], also utilizing Booth radix-4 and 4:2 counters [5]. In [6], Goto *et al.* presented a 54 x 54-bit multiplier with Booth radix-4 partial product generation, but used a regularly structured tree for partial product reduction, thereby simplifying the physical layout. Lu and Samueli were most concerned with throughput in the design of the multiplier-accumulator described in [7], and thus they presented a 13-stage, deeply pipelined 12 x 12-bit multiplier-accumulator which used no encoding and was implemented with a quasi-domino dynamic logic family. The data point representing this work is a 64 x 64-bit multiplier using both Booth radix-4 and radix-8 encoding with a Dadda reduction tree.

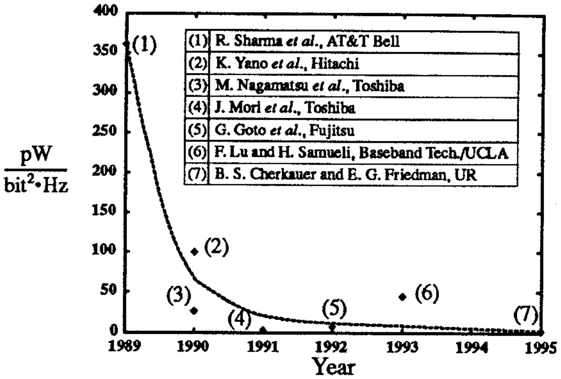

Figure 1. Multiplier power factor

In this paper a hybrid Booth radix-4/radix-8 multiplier architecture is presented as a method to trade-off speed and power dissipation in two's complement signed multipliers. The improved speed and power dissipation characteristics of this new multiplier architecture are compared with that of standard radix-4 and radix-8 based multipliers. The hybrid radix-4/radix-8 architecture presented in this paper is described in Section II. The speed and power dissipation characteristics of the three multiplier architectures are compared in Section III. Finally, some conclusions are drawn in Section IV.

II. Hybrid Radix Architecture (Radix-4/Radix-8)

The proposed hybrid radix-4/radix-8 multiplier architecture uses a combination of modified Booth radix-4 and radix-8 encoding [8–10]. The hybrid radix-4/radix-8 architecture mitigates the delay penalty associated with the generation of $3B$ (see Figure 2) for radix-8 encoding by using the additional parallelism of the radix-4 encoding/reduction. In this manner the hybrid radix-4/radix-8 multiplier combines the speed advantage of the radix-4 multiplier with the reduced power dissipation of the radix-8 multiplier.

In a radix-8 architecture, the multiplication process is serially dependent upon the time required to generate $3B$: while $3B$ is being generated by a high speed adder, no other processing can take place within the multiplier. This requirement to generate $3B$ leads to a significant delay penalty, on the order of 10-20%, as compared with a radix-4 architecture (where the partial products may be generated by simple shifting and/or complementing) [11].

In the hybrid radix-4/radix-8 architecture, a subset of the partial products are generated using radix-4 modified

This research was supported in part by the National Science Foundation under Grant No. MIP-9208165 and Grant No. MIP-9423886, the Army Research Office under Grant No. DAAH04–93–G-0323, and by a grant from the Xerox Corporation.

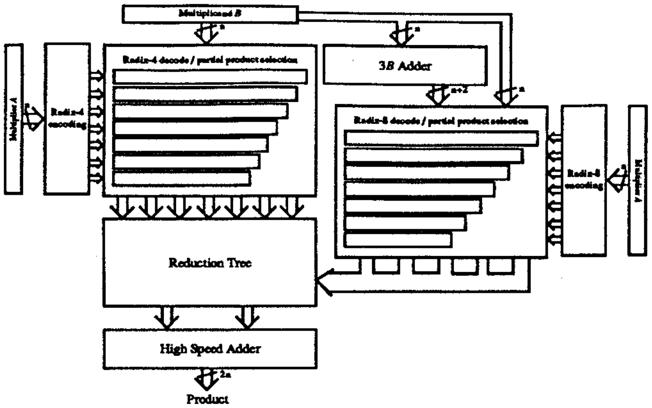

Figure 2. Hybrid radix-4/radix-8 multiplier architecture

Booth encoding. Reduction begins on these radix-4 partial products while $3B$ is simultaneously being generated by a high speed adder. Upon generating $3B$, the remaining partial products are generated using radix-8 encoding, and these partial products are subsequently included within the reduction tree. A Wallace/Dadda structure is assumed for the reduction tree [12,13]. In this manner, some reduction of the partial products takes place while the high speed adder is generating $3B$; therefore, less of a delay penalty is incurred. Utilizing radix-8 encoding for many of the partial products reduces the total number of partial products, thereby reducing the power required to sum the partial products. As described in Section III, three reduction steps take place during the generation of $3B$ for both the 32 x 32 bit multiplier and the 64 x 64 bit multiplier. A diagram of the hybrid radix-4/radix-8 architecture is shown in Figure 2.

It is important to note that the delay penalty associated with the generation of $3B$ can not be entirely mitigated using this hybrid approach. An additional delay penalty is incurred since all of the partial products are not immediately available when the reduction process is initiated. As Wallace/Dadda reduction trees utilize parallel adder cells to perform the partial product reduction, the more parallel data available to the tree, the more time efficient the reduction steps become. Thus, the availability of only a subset of the partial products at the initiation of the reduction process reduces the efficiency of the early reduction steps.

By delaying the generation of the radix-8 partial products until three reduction steps have been completed, fewer bits in parallel are initially available. Thus, the reduction process is not as time efficient, requiring additional reduction steps as compared with an architecture in which all the partial products are available simultaneously when the reduction process begins. In essence, the parallelism of the reduction tree is reduced in exchange for operating the reduction tree in parallel with the $3B$ adder.

By selecting the number of partial products generated by radix-4 and radix-8 encoding, it is possible to limit the number of reduction steps to just one more step than is required by a radix-4 multiplier (assuming 32 x 32 bit and 64 x 64 bit multipliers). For a 64 x 64 bit hybrid radix-4/radix-8 multiplier, ten partial products are generated by the radix-4 encoding and 15 by the radix-8 encoding. As the radix-8 partial products are not immediately available, it is convenient to use radix-8 on the lower order partial products, as the low order bits are not used in the early reduction steps. A 32 x 32 bit hybrid radix-4/radix-8 multiplier implementation has eight partial products generated by the radix-4 encoding and six partial products generated by the radix-8 encoding.

For this 64 x 64 bit hybrid radix-4/radix-8 implementation, the required nine reduction steps are as follows: $11 \rightarrow 9 \rightarrow 6 \rightarrow 4 + 15 \rightarrow 13 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$. For comparison, a 64 x 64 bit radix-4 multiplier requires eight reduction steps: $34 \rightarrow 28 \rightarrow 19 \rightarrow 13 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$, while a 64 x 64 bit radix-8 multiplier requires only seven reduction steps: $23 \rightarrow 19 \rightarrow 13 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$ [11]. Note that by using the one's complement plus the carry-in to form the two's complement, the number of bits at the start of the reduction process is one bit greater than the number of partial products. This additional bit is the carry-in of the highest order partial product. Thus, the hybrid reduction begins at eleven bits, although there are only ten partial products. However, when the radix-8 partial products become available after the third reduction step, the carry-in from the highest order radix-8 partial product does not align with any of the resultant bits from the first three reduction steps. Hence, the fourth reduction step begins with the four resultant bits plus the 15 radix-8 partial products, rather than four bits plus 16 partial products.

With a 32 x 32 bit multiplier, seven steps are required for partial product reduction in a hybrid radix-4/radix-8 implementation, as compared with six for a radix-4 implementation and five for a radix-8 implementation. The reduction steps for the 32 x 32 bit hybrid radix-4/radix-8 implementation are: $9 \rightarrow 6 \rightarrow 4 \rightarrow 3 + 6 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$.

III. Performance

The propagation delay, transistor count, and power dissipation characteristics of the 32 x 32 bit and the 64 x 64 bit multipliers are presented in this section. In subsection A, the delay of the new hybrid radix-4/radix-8 multiplier architecture is compared with the delay of the radix-4 and radix-8 multiplier architectures. In subsection B, the number of transistors required to implement each of the multipliers is presented. The power dissipation characteristics of the three architectures are compared in subsection C. The power dissipation characteristics of the three architectures with scaled power supply voltages assuming constant delay are compared in subsection D.

A. Delay Analysis

The 32 x 32 bit and 64 x 64 bit multipliers have been simulated in SPICE based on a 5 volt, 1.2 μm CMOS process technology. The delay of the worst case path of each multiplier architecture is shown in Table I. The radix-4 multiplier exhibits the least delay, and the radix-8 multiplier exhibits the most delay. The hybrid radix-4/radix-8 delay falls between those of the radix-4 and radix-8 multipliers. Note that the delays shown in Table I do not include the effects of interconnect impedances.

Table I. Technology dependent delay of multiplier architectures (1.2 μm, 5 volt CMOS)

		Radix-4	Hybrid Radix-4/8	Radix-8
64 x 64 bit	Partial Product Generation	3.3 ns	3.3 ns	9.2 ns
	Reduction	13.9 ns	16.3 ns	12.2 ns
	Final High Speed Addition	9.0 ns	9.0 ns	9.0 ns
	Total	26.2 ns	28.6 ns	30.4 ns
32 x 32 bit	Partial Product Generation	3.3 ns	3.3 ns	7.4 ns
	Reduction	10.4 ns	12.2 ns	8.7 ns
	Final High Speed Addition	7.9 ns	7.9 ns	7.9 ns
	Total	21.6 ns	23.4 ns	24.0 ns

B. Transistor Count

The number of transistors required to implement a multiplier architecture can provide a metric by which to judge the relative area requirements and power dissipation of the different architectures, assuming that switching probabilities for the transistors are relatively constant across architectures, as is the case in these multipliers. The transistor count for the 32 x 32 bit and 64 x 64 bit implementations of each of the three architectures are compared in Table II. The radix-8 implementations require the fewest transistors, while the radix-4 implementations require the most transistors. The number of transistors required to implement the hybrid radix-4/radix-8 multipliers falls between those of the radix-4 and radix-8 multipliers.

Table II. Transistor count for each multiplier implementation

Bit Width	Radix-4	Hybrid Radix-4/8	Radix-8
32 x 32	28,522	25,678	23,542
64 x 64	108,038	90,210	83,412

C. Power Dissipation

The power dissipation of the multipliers has also been analyzed based on a 5 volt, 1.2 μm CMOS process technology. The average power dissipation of each circuit operating at 10 MHz is determined from SPICE using the Kang power meter [14]. The power dissipation of each component is averaged over 100 random input vectors. The input control signals that drive the decoder/selector circuitry are weighted such that the probabilities of the control signals within the test set conform to the signal assertion probabilities generated by the encoder, e.g., *C1* is twice as likely to be asserted as either *C0* or *C2* in a radix-4 decoder/selector. The results of these simulations are presented in Table III.

Note that the encoder cells in the 64 x 64 bit and 32 x 32 bit multipliers are identical; however, the loading differs by a factor of approximately two. This distinction accounts for the differences in encoder power dissipation between the two multiplier configurations. In an n x n bit multiplier, a radix-4 encoder drives n+1 decoders, while a radix-8 encoder drives n+2 decoders. Tapered buffers [15] have been included between the encoders and decoders to drive this large fanout, and the power dissipation of these buffers has been included in the total power dissipation of the encoder listed in Table III. As with the delay values presented in Table I, these power dissipation figures do not account for interconnect impedances.

Also note that although the sign generation circuitry is identical for both the radix-4 and radix-8 implementations, the power dissipation of this circuit is not identical for both applications. This disparity between the radix-4 and the radix-8 power dissipation exists because the control signal input *C0* does not toggle as frequently in a radix-8 implementation as it does in a radix-4 implementation. This disparity in toggling frequency leads to lower dynamic power dissipation in the sign bit generation circuit. As the number of sign extension bits varies for each partial product, the power dissipation of the sign generation circuitry (as shown in Table III) accounts for only the loading of the first stage of the tapered buffers which drive the sign extension bits. Since the tapered buffer is customized for the specific loading of each partial product, the power dissipation of these buffers is included in the architecture-specific power dissipation totals presented in Tables IV and V. Finally, note that the power dissipation of the high speed adders are estimations based upon the simulated power dissipation of the adders assuming the 1.2 μm, 5 volt CMOS technology used in these example circuits and upon extrapolations from data presented in [16].

Table III. Power dissipation of multiplier components. 5 V, 1.2 μm CMOS, 10 MHz.

Multiplier Components	Power Dissipation (μW)
Full Adder Cell	18.5
Radix-4 Encoder	
64 x 64 bit	236
32 x 32 bit	154
Radix-4 Decoder	15.0
Radix-8 Encoder	
64 x 64 bit	380
32 x 32 bit	258
Radix-8 Decoder	19.9
Radix-4 Sign Generation	31.5
Radix-8 Sign Generation	29.9
3*B* Adder	
65 bit	2239
33 bit	1130
Final High Speed Adder	
128 bit	4403
64 bit	2202

Table IV. Total hardware and power dissipation for a 64 x 64 bit multiplier. 5 V, 1.2 μm CMOS, 10 MHz.

	Radix-4		Hybrid Radix-4/8		Radix-8	
Component	Number of Instances	Power Dissipation (mW)	Number of Instances	Power Dissipation (mW)	Number of Instances	Power Dissipation (mW)
1-Bit Adder	2914	53.91	2179	40.31	1953	36.13
Rad-4 Encoder	32	7.55	10	2.36	–	–
Rad-4 Decoder	2080	31.20	650	9.75	–	–
Rad-8 Encoder	–	–	15	5.70	22	8.36
Rad-8 Decoder	–	–	990	19.70	1452	28.89
Rad-4 Sign Gen.	32	1.01	10	0.32	–	–
Rad-8 Sign Gen.	–	–	15	0.45	22	0.66
Sign bits	961	1.00	657	0.70	630	0.66
3*B* Adder	–	–	1	2.24	1	2.24
Final Adder	1	4.40	1	4.40	1	4.40
Total Power Dissipation	–	99.1	–	85.9	–	81.3

The total power dissipated by each multiplier architecture is shown for a 64 x 64 bit multiplier in Table IV and for a 32 x 32 bit multiplier in Table V. For simplicity, half adder cells have been considered as full adder cells and are shown as 1-bit adders in Tables IV and V.

Table V. Total hardware and power dissipation for a 32 x 32 bit multiplier. 5 V, 1.2 μm CMOS, 10 MHz.

	Radix-4		Hybrid Radix-4/8		Radix-8	
Component	Number of Instances	Power Dissipation (mW)	Number of Instances	Power Dissipation (mW)	Number of Instances	Power Dissipation (mW)
1-Bit Adder	690	12.77	531	9.82	455	8.42
Rad-4 Encoder	16	2.46	7	1.08	–	–
Rad-4 Decoder	528	7.92	231	3.47	–	–
Rad-8 Encoder	–	–	6	1.55	11	2.84
Rad-8 Decoder	–	–	204	4.06	374	7.44
Rad-4 Sign Gen.	16	0.50	7	0.22	–	–
Rad-8 Sign Gen.	–	–	6	0.18	11	0.33
Sign bits	225	0.27	159	0.20	145	0.18
3*B* Adder	–	–	1	1.13	1	1.13
Final Adder	1	2.20	1	2.20	1	2.20
Total Power Dissipation	–	26.1	–	23.9	–	22.5

As described previously and shown in Tables IV and

V, a radix-8 multiplier dissipates less power than a radix-4 multiplier. The hybrid radix-4/radix-8 architecture dissipates power at a level between that of the radix-4 and radix-8 multipliers. Thus, the hybrid radix-4/radix-8 multiplier architecture is a useful architecture for those applications which require low power while operating at speeds greater than that of a full radix-8 multiplier. Radix-8 multiplication is appropriate for those ultra-low power systems in which added delay can be tolerated.

D. The Effects of Voltage Scaling on Performance

Voltage scaling, reducing the power supply voltage, may be applied to higher speed multipliers to reduce the power dissipation of these circuits, while simultaneously increasing delay. The delay of the multipliers is proportional to the power supply, V_{DD}, as shown in (1), where V_T represents the transistor threshold voltage, and the power dissipation is proportional to the square of the power supply voltage as shown in (2) [17].

$$Delay \propto \frac{V_{DD}}{(V_{DD} - V_T)^2} \qquad (1)$$

$$Power \propto (V_{DD})^2 \qquad (2)$$

The power dissipation of the radix-4, hybrid radix-4/radix-8, and radix-8 multipliers after voltage scaling is compared in Table VI. Note that the scaled voltage levels are referenced to the radix-8 multiplier operating at 5 volts. For shorter bit widths such as exemplified by a 32 x 32 bit multiplier, the delay and power dissipation overhead due to the additional $3B$ adder and more complex encoding is not outweighed by the reduction in delay and power dissipation associated with the partial product summation. In this case, the simpler radix-4 encoded multiplier provides the lowest power dissipation at a given delay.

However at higher bit widths, as exemplified by the 64 x 64 bit multipliers, the radix-4 and radix-8 multipliers dissipate approximately equivalent power at a given delay, both of which are greater than the hybrid radix-4/radix-8 multiplier.

Table VI. Comparison of performance of voltage scaled multiplier architectures

		Radix-4	Hybrid Radix-4/8	Radix-8
64 x 64 bit	V_{DD} for 30.4 ns delay	4.53 V	4.79 V	5.00 V
	Power dissipation	81.3 mW	78.8 mW	81.3 mW
32 x 32 bit	V_{DD} for 24.0 ns delay	4.65 V	4.91 V	5.00 V
	Power dissipation	22.5 mW	23.1 mW	24.0 mW

IV. Conclusions

As higher bit widths and lower power become important design issues in multipliers, the opportunity to develop new architectures to meet these requirements arises. A new hybrid radix-4/radix-8 multiplier architecture is presented in this paper that is both low power and high speed; this architecture provides a trade-off between the high speed of a radix-4 multiplier architecture and the low power dissipation of a radix-8 multiplier architecture. In this hybrid radix-4/radix-8 multiplier architecture, the performance bottleneck of a radix-8 multiplier (the generation of $3B$ for the radix-8 partial product generation) is performed in parallel with the reduction of the radix-4 partial products rather than serially, as in a radix-8 multiplier. Thus, the hybrid radix-4/radix-8 multiplier accomplishes a portion of the partial product reduction while a high speed adder is generating $3B$. This strategy minimizes a portion of the delay penalty incurred by the radix-8 multiplier in generating $3B$. The hybrid radix-4/radix-8 multiplier architecture dissipates 13% less power in a 64 x 64 bit multiplier with only a 9% increase in delay, as compared to a radix-4 implementation. When the supply voltage of the 64 x 64 bit multipliers is scaled such that the radix-4, radix-8, and hybrid radix-4/radix-8 multipliers exhibit the same delay, the hybrid radix-4/radix-8 multiplier dissipates the least power. The hybrid radix-4/radix-8 architecture therefore provides a trade-off between high speed and low power for application to those systems which require both high speed and low power signed multiplication.

References

[1] S. R. Powell and P. M. Chau, "Estimating Power Dissipation of VLSI Signal Processing Chips: the PFA Technique," *VLSI Signal Processing IV*, ch. 24, New York: IEEE Press, 1990.

[2] R. Sharma, A. D. Lopez, J. A. Michejda, S. J. Hillenius, J. M. Andrews, and A. J. Studwell, "A 6.75ns 16x16-bit Multiplier in Single-Level-Metal," *IEEE Journal of Solid-State Circuits*, Vol. SC-24, No. 4, pp. 922–927, August 1989.

[3] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, "A 3.8-ns CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic," *IEEE Journal of Solid-State Circuits*, Vol. SC-25, No. 2, pp. 388–395, April 1990.

[4] M. Nagamatsu, S. Tanaka, J. Mori, K. Hirano, T. Noguchi, and K. Hatanaka, "A 15ns 32x32b CMOS Multiplier with an Improved Parallel Structure," *IEEE Journal of Solid-State Curcuits*, Vol. SC-25, No. 2, pp. 494–497, April 1990.

[5] J. Mori, M. Nagamatsu, M. Hirano, S. Tanaka, M. Noda, Y. Yoyoshima, K. Hashimoto, H. Hayashida, and K. Maeguchi, "A 10ns 54x54b Parallel Structured Full Array Multiplier with 0.5μm CMOS Technology," *IEEE Journal of Solid-State Circuits*, Vol. SC-26, No. 4, pp. 600–606, April 1990.

[6] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54x54-b Regularly Structured Tree Multiplier," *IEEE Journal of Solid-State Circuits*, Vol. SC-27, No. 9, pp. 1229–1236, September 1992.

[7] F. Lu and H. Samueli, "A 200-MHz CMOS Pipelined Multiplier-Accumulator using a Quasi-Domino Full-Adder Cell Design," *IEEE Journal of Solid-State Circuits*, Vol. SC-28, No. 2, pp. 123–132, February 1993.

[8] A. D. Booth, "A Signed Binary Multiplication Technique," *Quarterly Journal of Mechanics and Applied Mathematics*, Vol. 4, No. 2, pp. 236–2450, June 1951.

[9] O. L. MacSorley, "High-Speed Arithmetic in Binary Computers," *Proceedings of the IRE*, Vol. 49, pp. 67–91, January 1961.

[10] H. Sam and A. Gupta, "A Generalized Multibit Recoding of Two's Complement Binary Numbers and Its Proof with Application in Multiplier Implementations," *IEEE Transactions on Computers*, Vol. C-39, No. 8, pp. 1006–1015, August 1990.

[11] B. Millar, P. E. Madrid, and E. E. Swartzlander, Jr., "A Fast Hybrid Multiplier Combining Booth and Wallace/Dadda Algorithms," *Proceedings of the 35th IEEE Midwest Symposium on Circuits and Systems*, pp. 158–165, August 1992.

[12] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, Vol. EC-13, pp. 14–17, February 1964.

[13] L. Dadda, "Some Schemes for Parallel Multipliers," *Alta Frequenza*, Vol. 34, No. 5, pp. 349–356, May 1965.

[14] S. M. Kang, "Accurate Simulation of Power Dissipation in VLSI Circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, No. 5, pp. 889–891, October 1986.

[15] B. S. Cherkauer and E. G. Friedman, "A Unified Design Methodology for CMOS Tapered Buffers," *IEEE Transactions on VLSI Systems*, Vol. VLSI-3, No. 1, pp. 99–111, March 1995.

[16] T. K. Callaway and E. E. Swartzlander, Jr., "Estimating the Power Consumption of CMOS Adders," *Proceedings of the 11th IEEE Symposium on Computer Arithmetic*, pp. 210–216, June/July 1993.

[17] A. P. Chandrakasan, S. Sheng, and R. W. Broderson, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol. SC–27, No. 4, pp. 473–483, April 1992.