# Brief Contributions

## A Reduced Complexity Wallace Multiplier Reduction

Ron S. Waters, *Member*, *IEEE*, and
Earl E. Swartzlander, Jr., *Fellow*, *IEEE*

**Abstract**—Wallace high-speed multipliers use full adders and half adders in their reduction phase. Half adders do not reduce the number of partial product bits. Therefore, minimizing the number of half adders used in a multiplier reduction will reduce the complexity. A modification to the Wallace reduction is presented that ensures that the delay is the same as for the conventional Wallace reduction. The modified reduction method greatly reduces the number of half adders; producing implementations with 80 percent fewer half adders than standard Wallace multipliers, with a very slight increase in the number of full adders.

**Index Terms**—High-speed multiplier, Wallace multiplier, Dadda multiplier.

## 1 Introduction

THE well-known Wallace high-speed multiplier uses carry save adders to reduce an N-row bit product matrix to an equivalent two row matrix that is then summed with a carry propagating adder to give the product [1]. It is a fully parallel version of the multiplier used in the IBM Stretch computer [2]. The carry save adders are conventional full adders whose carries are not connected, so that three words are taken in and two words are output. As described more clearly in [3], the Wallace multiplier also uses half adders in the reduction phase. This paper presents a modified design that greatly reduces the number of half adders in Wallace multipliers.

This section describes the notation and terminology used in this paper. Throughout this paper, both input operands are assumed to be N-bit unsigned words. Section 2 reviews the two main previous multiplier reduction approaches (Wallace and Dadda). Section 3 presents the modified Wallace approach to multiplier reduction. Section 4 presents the results for multipliers of sizes ranging from 8 to 64 bits. Finally, Section 5 summarizes the conclusions.

The multipliers involve three distinct phases. First, the matrix of partial products is formed. These are simply the AND of each bit of the first operand with each bit of the second operand. This forms an N-row skewed matrix of partial product terms where each row contains N single bit terms. In the second phase, the partial product matrix is reduced with carry save adders to a height of two terms. Finally, in the third phase, the two terms are added with a carry propagating adder to generate the product. It is usually assumed that a carry lookahead adder is used for the addition.

This paper is concerned with the second phase where the N rows of partial product bits are reduced to two rows. In this phase, the Wallace approach uses several stages of full and half adders as carry save adders that are arranged to maximize the reduction at each stage. Full adders take in three bits and output two bits for a net reduction of one bit per full adder. Half adders take in two bits and output two bits. They simply shift the position of one of the bits. For an N-bit by N-bit Wallace multiplier, the number of half adders is roughly proportional to $N^{1.5}$ which results in the second phase of Wallace multipliers being more complex and larger than that of Dadda multipliers which use $\mathrm{N}-1$ half adders.

The Dadda dot notation for the second stage of multipliers (array reduction) [3] is used here. The multipliers, in this paper, are for unsigned operands, but the same process is used for multipliers for two's complement operands by inverting the bit products along the bottom and top edge of the initial bit product array from the center column to the next to the leftmost column and adding a one to the first column to the left of the center column [4]. Adding the one can be done using a special half adder that computes the sum of the two data plus one. It is the same complexity as a conventional half adder. Thus, the complexity of the reduction for two's complement operands is about the same as that for unsigned operands.

- *R.S. Waters is with TSMC North America, Austin, Texas 78759 and the University of Texas at Austin, Texas. E-mail: ron.waters@ieee.org, ron@2talkback.com.*
- *E.E. Swartzlander, Jr. is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Mailcode C0803, Austin, Texas 78712. E-mail: eswartzla@aol.com, e.swartzlander@ieee.org.*

*Manuscript received 24 Jan. 2009; revised 11 July 2009; accepted 11 Oct. 2009; published online 28 Apr. 2010.*
*Recommended for acceptance by P. Montuschi.*
*For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2009-01-0047.*
*Digital Object Identifier no. 10.1109/TC.2010.103.*

## 2 Previous Approaches

### 2.1 Wallace Reduction Approach

For the conventional Wallace reduction method, once the partial product array (of $\mathrm{N}^2$ bits) is formed, adjacent rows are collected into nonoverlapping groups of three. Each group of three rows is reduced by 1) applying a full adder to each column that contains three bits, 2) applying a half adder to each column that contains two bits, and (3) passing any single bit columns to the next stage without processing. This reduction method is applied to each successive stage until only two rows remain. The final two rows are summed with a carry propagating adder. This process is illustrated by the conventional 9-bit by 9-bit Wallace multiplier shown in Fig. 1. Light lines show the three row groupings. The reduction is performed in four stages (each with the delay of one full adder) with a total of 50 full adders and 21 half adders. The third phase will require a 13-bit wide adder.

### 2.2 Dadda Reduction Approach

In contrast to the Wallace reduction, the Dadda method [3] does the least reduction necessary at each stage. To determine how much reduction is required, the maximum height of each stage is calculated by working back from the final stage. It has a height of two rows. Each preceding stage height can be no larger than $\lfloor 3 \bullet \mathit{successor\ height}/2 \rfloor$ where $\lfloor \mathrm{x} \rfloor$ denotes the integer portion of x. This gives 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, etc., as the maximum heights for the various stages. The Dadda reduction then uses just enough full and half adders to achieve the limits. A 9-bit by 9-bit Dadda multiplier is shown in Fig. 2. The reduction is performed in four stages (the same as the Wallace reduction) with a total of 48 full adders and 8 half adders. The third phase will require a 16-bit wide adder.

The Dadda multiplier uses two fewer full adders and 13 fewer half adders in the second phase reduction than the Wallace multiplier and requires a slightly larger carry propagating adder in the third phase as a result.

## 3 Modified Wallace Reduction

### 3.1 Modified Wallace Reduction Approach

This section presents the modified Wallace method for reducing the partial product array (the second phase). As shown in Fig. 3, the initial partial product array of the first phase is changed by shifting bits in the left half of the array upward to form an inverted pyramid array (the partial product generation method is the same, no data are changed; only their vertical position is shifted).

In the second phase, the modified Wallace approach is similar to the conventional Wallace approach in that it uses as many full adders as possible, but different in that it only uses half adders when necessary to ensure that the number of reduction stages is the same as for a conventional Wallace multiplier.

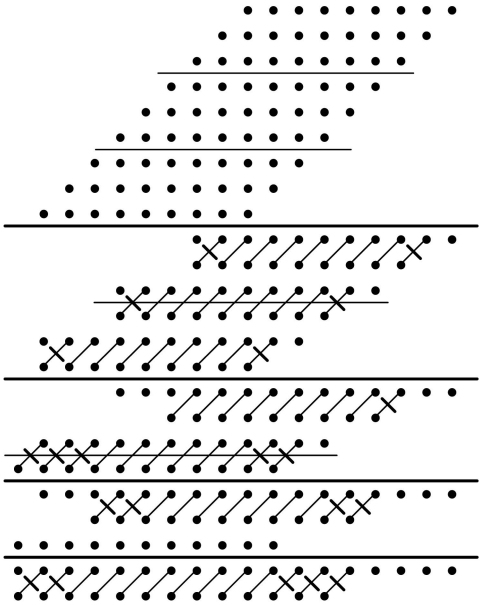

Fig. 1. Conventional Wallace 9-bit by 9-bit reduction.

The modified Wallace reduction method divides the matrix into three row groups and uses full adders for each group of three bits in a column like the conventional Wallace reduction. A group of two bits in a column is not processed, that is, it is passed on to the next stage (in contrast to the conventional Wallace reduction). Single bits are passed on to the next stage as in the conventional Wallace reduction.

The only time half adders are used is to ensure that the number of stages of the modified Wallace multiplier does not exceed that of a conventional Wallace multiplier. For some cases, half adders are only used in the final stage of reduction.

In the 9-bit by 9-bit example, a half adder is used in the first and the second stages to handle partial product terms that, if not addressed, would increase the number of reduction stages and increase the multiplier delay. Also two half adders are used in the final stage. The reduction is performed in four stages (the same as the Wallace reduction) with a total of 52 full adders and four half adders. The third phase will require a 16-bit wide adder. The modified Wallace reduction uses two more full adders and 17 fewer half adders than the conventional Wallace reduction.

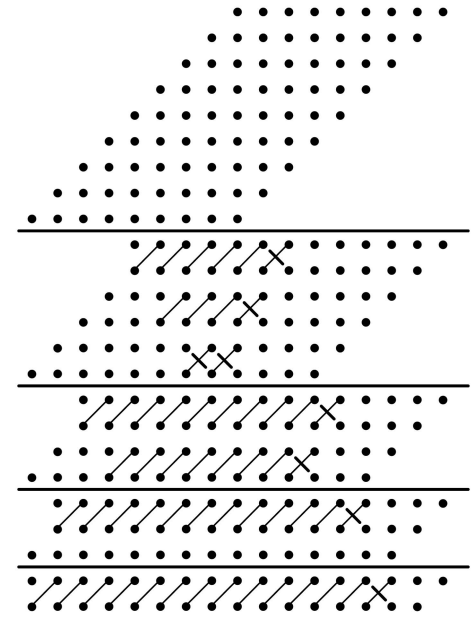

Fig. 2. Dadda 9-bit by 9-bit reduction.

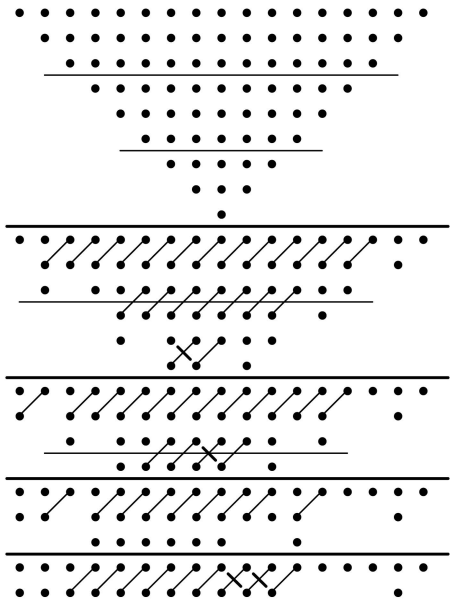

Fig. 3. Modified Wallace 9-bit by 9-bit reduction.

### 3.2 Multiplier Configuration Details

Prior to working through the reduction procedure, the maximum number of rows, $r_i$, in ith stage of the reduction can be derived. For an N-bit multiplier, the number of rows in the initial bit product array, $r_0$, is N. The number of rows in subsequent stages of a conventional Wallace multiplier is

$$r_{i+1} = 2\lfloor r_i/3 \rfloor + r_i \bmod 3. \quad (1)$$

$$\text{If } r_i \bmod 3 = 0, \text{then } r_{i+1} = 2r_i/3, \quad (2)$$

where, y mod z denotes the smallest nonnegative remainder of y/z. For a 9-bit by 9-bit multiplier, the stage heights given by (1) are: $r_0 = N = 9$, $r_1 = 6$, $r_2 = 4$, $r_3 = 3$, and $r_4 = 2$. Therefore, four stages are required for the reduction. As seen in the reduction of the 9-bit by 9-bit modified Wallace multiplier shown in Fig. 3, without the half adder in the first stage, the second matrix would have seven rows instead of the six that are required for the conventional Wallace reduction.

Analysis of the reduction stages indicates that if the jth reduction stage has a number of rows that is exactly divisible by three, (i.e., if $\mathrm{r_j} \bmod 3 = 0$), then if no half adders are used, the $\mathrm{j}+1$th stage will contain an "extra" row. In the case of a 9-bit by 9-bit multiplier, the initial stage has nine rows and a half adder is needed to reduce the next stage to six rows. Similarly, the second stage (with six rows) needs a half adder to reduce the next stage to four rows. Finally, the penultimate stage has three rows and needs half adders to reduce it to two rows.

To show the validity of this, three cases can be considered: $\mathrm{r_j} \bmod 3 = 0$, $\mathrm{r_j} \bmod 3 = 1$, and $\mathrm{r_j} \bmod 3 = 2$.

To simplify the analysis, let $\mathrm{r_0} \bmod 3 = 0$, as shown, for example, in Fig. 3. In the first stage, column N (the center column) contains nine rows (an integer multiple of three rows). From (1), stage 2 should have no more than six rows. Column 8, in stage 2, has $(\mathrm{N}/3) - 1$ sum bits, two unprocessed bits, and $(\mathrm{N}/3) - 1$ carry bits from the adjacent column to its right for a total of 2 N/3 bits. Column 9, in stage 2, has N/3 sum bits and $(\mathrm{N}/3) - 1$ carry bits from the adjacent column to its right for a total of $(2\mathrm{N}/3) - 1$ bits. Without the use of half adders, column 10 of stage 2 has $(\mathrm{N}/3) - 1$ sum bits, 2 unprocessed bits, and (N/3) carry bits from column 9 for a total of $(2\mathrm{N}/3) + 1$ bits which

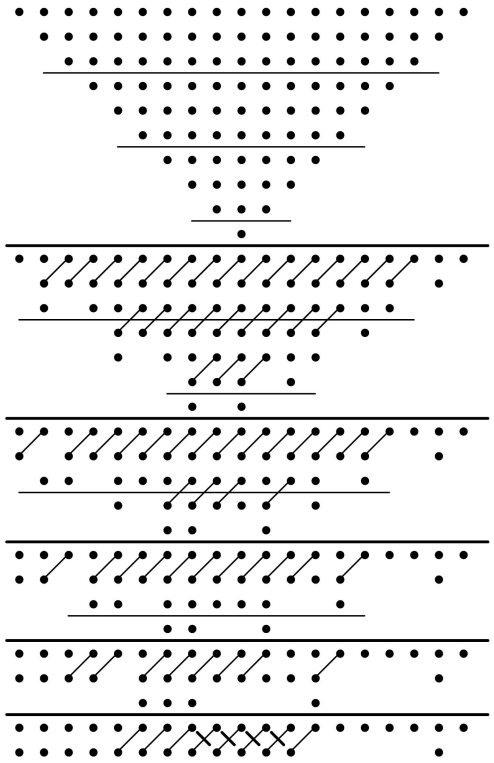

Fig. 4. Modified Wallace 10-bit by 10-bit reduction.

violates (1). With a half adder, column 10 has two sum bits, one sum bit from the half adder, and 3 carry bits from column 9 for a total of 6 bits. Thus, a half adder is required to satisfy (1).

For this example, $r_1 = 6$, so $r_2 = 4$ from (1). Columns 8 and 9 of the second stage have 6 and 5 bits, respectively. Without the use of half adders, column 9 of stage three has one sum bit, two unprocessed bits, and two carry bits from column 8 for a total of 5 bits. With the use of a half adder for the two unprocessed bits, column 9 has one sum bit from the full adder, one sum bit from the half adder, and two carry bits from column 8 for a total of 4 bits.

Also, for this example, $r_3 = 3$, so $r_4 = 2$ from (1). Columns 6, 7, and 8 of the fourth stage have 3, 2, and 2 bits, respectively. Without the use of half adders, column 7 of stage 5 has two unprocessed bits and one carry bit from column 6 for a total of 3 bits. With the use of a half adder for the two unprocessed bits, column 7 has one sum bit from the half adder and one carry bit from column 6 for a total of 2 bits. A similar situation occurs with column 8.

Now, let $\mathrm{r}_0 \bmod 3 = 1$, as shown by the example for $\mathrm{N} = 10$ in Fig. 4. Column $\mathrm{N}-1$ in stage 2 has $\lfloor N/3 \rfloor$ sum bits from column $\mathrm{N}-1$, and $\lfloor N/3 \rfloor - 1$ carry bits from column $\mathrm{N}-2$ for a total of $(2\lfloor N/3 \rfloor) - 1$ bits. Column N of stage 2 has $\lfloor N/3 \rfloor$ sum bits and one unprocessed bit from column N, and $\lfloor N/3 \rfloor$ carry bits from column $\mathrm{N}-1$ for a total of $(2\lfloor N/3 \rfloor) + 1$ bits. Column $\mathrm{N}+1$ has $\lfloor N/3 \rfloor$ sum bits from column $\mathrm{N}+1$ and $\lfloor N/3 \rfloor$ carry bits from column N for a total of $2\lfloor N/3 \rfloor$ bits. Thus, no half adders are needed to satisfy (1).

Since $r_i \bmod 3 \neq 0$ for $\mathrm{i} = 0, 1, 2, 3$, no half adders are needed for stages 1-4, but $r_4 \bmod 3 = 0$ so half adders are needed in stage 5. Column 7 of stage 5 has 3 bits while columns 8-11 have 2 bits each. Without the use of half adders, column 8 of stage 6 has two unprocessed bits and one carry bit from column 7 for a total of 3 bits. With the use of a half adder for the two unprocessed bits, column 8 has one sum bit from the half adder and one carry bit from column 7 for a total of 2 bits. A similar situation occurs with columns 9-11.

Now, let $r_0 \bmod 3 = 2$, as shown by the example for $\mathrm{N} = 8$ in Fig. 5. Column 7 in stage 2 has $\lfloor N/3 \rfloor$ sum bits and one unprocessed bit from column 7, and $\lfloor N/3 \rfloor$ carry bits from column 6 for a total of $(2\lfloor N/3 \rfloor) + 1$ bits. Column 8 of stage 2 has

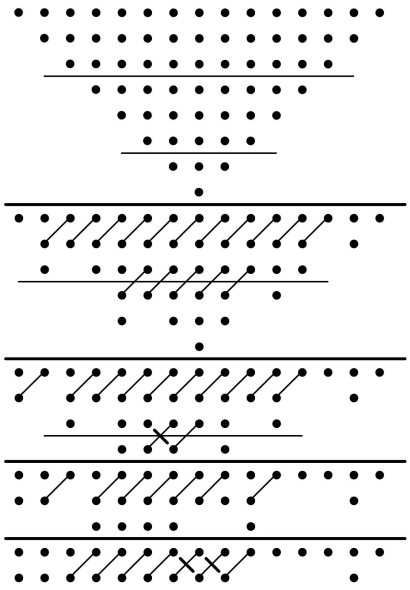

Fig. 5. Modified Wallace 8-bit by 8-bit reduction.

$\lfloor N/3 \rfloor$ sum bits and two unprocessed bits from column 8, and $\lfloor N/3 \rfloor$ carry bits from column 7 for a total of $(2\lfloor N/3 \rfloor) + 1$ bits. Column 9 has $\lfloor N/3 \rfloor$ sum bits and one unprocessed bit from column 9 and $\lfloor N/3 \rfloor$ carry bits from column 8 for a total of $(2\lfloor N/3 \rfloor) + 1$ bits. Thus, no half adders are needed to satisfy (1) in the first stage. Stage 2 has six rows so a half adder is needed to meet the constraint that stage 3 has four rows. Columns 8 and 9 of stage 2 have six and five rows, respectively and the situation is similar to stage 2 of the 9-bit reduction shown in Fig. 3.

### 3.3 Carry Propagating Adder

Although a complete analysis of the carry propagating adder (used in the third phase) is beyond the scope of this paper, some comments are in order. Figs. 3, 4, and 5 illustrate the requirements for the final adder for the modified Wallace multiplier. A direct implementation requires a $2\,\mathrm{N}-2$ bit carry propagating adder (the same as for the Dadda multiplier).

One possible carry propagating adder for the modified Wallace multiplier is a hybrid adder consisting of $\mathrm{S}+1$ ripple carry half adders at the least significant end combined with a $2\,\mathrm{N}-(\mathrm{S}+3)$ bit carry lookahead adder at the most significant end, where S is the number of stages in the reduction. If the hybrid final adder is used, the resulting modified Wallace multiplier is similar to the reduced area multiplier [5].

If the hybrid approach described above is used, the final adder for a Dadda multiplier would require a larger carry lookahead adder than either the conventional Wallace or the modified Wallace multiplier. If a carry lookahead adder is used, the increased length may require an additional level of lookahead that would cause the Dadda multiplier to be slightly slower than either the Wallace or the modified Wallace multiplier. For example, if $10 \leq \mathrm{N} \leq 12$, the Dadda carry lookahead adder size is 18 to 22 bits (14 gate delays) while the modified Wallace carry lookahead adder size is 12 to 16 bits (10 gate delays). These delays assume that the carry lookahead adder is implemented with gates with a fan in of up to 4 bits, each with one unit of delay. The additional delay might be significant in highly pipelined multipliers (where the delay of the final adder might determine the maximum pipeline clock rate). In nonpipelined multipliers, the second phase delay of the Dadda multipliers is usually faster than that of conventional Wallace multipliers [6], so the small increase in final adder delay would be negligible. The modified Wallace multiplier delay should be about the same as the conventional Wallace multiplier delay.

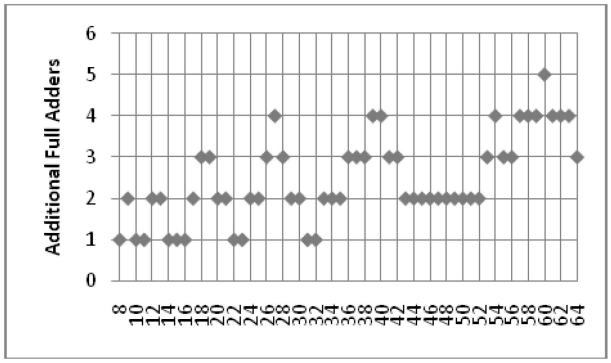


Fig. 6. Number of extra full adders for the modified wallace reduction compared to the Wallace reduction.

TABLE 1
Complexity of the Reduction

| INPUT SIZE (N) | 8 | 16 | 24 | 32 | 64 |
|---|---|---|---|---|---|
| STAGES (S) | 4 | 6 | 7 | 8 | 10 |
| ***WALLACE*** | | | | | |
| FULL ADDERS | 38 | 200 | 488 | 906 | 3,850 |
| HALF ADDERS | 15 | 52 | 100 | 156 | 430 |
| TOTAL GATES | 402 | 2,008 | 4,801 | 8,778 | 36,388 |
| ***MODIFIED WALLACE*** | | | | | |
| FULL ADDERS | 39 | 201 | 490 | 907 | 3,853 |
| HALF ADDERS | 3 | 9 | 16 | 23 | 53 |
| TOTAL GATES | 363 | 1,845 | 4,474 | 8,263 | 34,889 |
| ***DADDA*** | | | | | |
| FULL ADDERS | 35 | 195 | 483 | 899 | 3,843 |
| HALF ADDERS | 7 | 15 | 23 | 31 | 63 |
| TOTAL GATES | 343 | 1,815 | 4,439 | 8,215 | 34,839 |

## 4 Results

This section analyzes the number of half adders and full adders for the second phase of conventional Wallace, modified Wallace, and Dadda multipliers for a wide range of sizes.

### 4.1 The Number of Half Adders

The second phase reduction of multipliers for sizes from 8 to 64 bits were analyzed using the conventional and modified Wallace approaches. Designs were made for each size with generators programmed in Matlab. The generators are available on the web [7]. For the conventional Wallace multiplier, the number of half adders is roughly proportional to $N^{1.5}$. For the modified Wallace multiplier, the number of half adders used in the reduction was found empirically to be $\mathrm{N} - \mathrm{S} - 1$. Thus, for $\mathrm{N} \geq 8$, the conventional Wallace reduction requires at least five times as many half adders as the modified Wallace reduction.

### 4.2 The Number of Full Adders

For all multiplier sizes analyzed, the number of full adders required for the modified Wallace reduction is slightly higher than that of the conventional Wallace reduction, but the increase is less than 0.2 percent. Fig. 6 illustrates that the increase in the full adder count for the modified Wallace multiplier is between one and five full adders for $8 \leq \mathrm{N} \leq 64$.

### 4.3 Final Adder Complexity

As indicated in Section 3.3, the third phase adder for the modified Wallace multiplier requires a $2\,\mathrm{N} - 2$-bit carry propagating adder. One implementation of this is with a hybrid adder with a $2\,\mathrm{N} - (\mathrm{S} + 3)$-bit fast adder and $\mathrm{S} + 1$ half adders. The conventional Wallace and Dadda multipliers require $2\,\mathrm{N} - (\mathrm{S} + 1)$-bit and $2\,\mathrm{N} - 2$-bit fast adders, respectively. Thus, the fast adder is about the same size for the modified Wallace and the conventional Wallace multipliers and both of these are slightly smaller than the fast adder for the Dadda multiplier.

### 4.4 Reduction Complexity Analysis

Table 1 shows the complexity of the conventional and modified Wallace reductions as produced with the Matlab generator programs. The complexity of the Dadda reductions is shown for completeness.

The total gate count is based on the use of gate level designs (with full adders implemented in nine gates and half adders implemented in four gates). In practice, optimized transistor level designs may be used resulting in a different total complexity, but the relative complexity is expected to be generally similar. The table only includes the reductions, neither the $N^2$ AND gates that form the bit products nor the carry propagating adder are included.

As shown clearly in Table 1, for all word sizes, the modified Wallace reduction uses only a very few full adders more than the conventional Wallace reduction. On the other hand, the number of half adders is reduced by a factor of five (for the 8-bit reduction) to a factor of eight (for the 64-bit reduction). The total gate count is always less for the modified Wallace reduction than for the conventional Wallace reduction by 10 (for the 8-bit reduction) to four percent (for the 64-bit reduction).

For all word sizes, the modified Wallace reduction uses a few more full adders than the Dadda reduction. The number of half adders is reduced significantly for the modified Wallace reduction. The Dadda reduction requires a slightly smaller total gate count than the modified Wallace reduction, but a complete Dadda multiplier will require more gates due to its larger carry propagating adder.

An interesting point is that the sum of the number of full adders and the number of half adders is the same for the modified Wallace reduction and the Dadda reduction. It is not evident why this occurs.

## 5 Conclusion

This paper presents a modification to the second phase reduction (that reduces N rows of bit products to two rows) used in Wallace multipliers. The modified Wallace reduction reduces the number of half adders required by at least 80 percent compared to the conventional Wallace reduction with only a very slight increase in the number of full adders. Both the conventional Wallace and modified Wallace reductions have the same number of stages and consequently the delay is expected to be the same. It is significant that both the conventional and modified Wallace second phase reductions use more gates than the Dadda reduction, although the penalty is less for the modified Wallace reduction.

## Acknowledgments

The authors would like to thank the anonymous referees for their constructive comments.

## References

[1] C.S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Trans. Electronic Computers,* vol. 13, no. 1, pp. 14-17, Feb. 1964.
[2] E. Bloch, "The Engineering Design of the Stretch Computer," *Proc. Eastern Joint IRE-AIEE-ACM Computer Conf.,* no. 16, pp. 48-58, 1959.
[3] L. Dadda, "Some Schemes for Parallel Multipliers," *Alta Frequenza,* vol. 34, pp. 349-356, 1965.
[4] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs.* Oxford Univ. Press, p. 179, 2000.
[5] K.C. Bickerstaff, M.J. Schulte, and E.E. Swartzlander Jr., "Parallel Reduced Area Multipliers," *J. VLSI Signal Processing Systems,* vol. 9, no. 3, pp. 181-191, Apr. 1995.
[6] W.J. Townsend, E.E. Swartzlander Jr., and J.A. Abraham, "A Comparison of Dadda and Wallace Multiplier Delays," *Proc. SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations XIII,* pp. 552-560, 2003.
[7] R.S. Waters, *MATLAB based Wallace and Modified Wallace Multiplier Generator Programs,* http:\\ronwaters.com\, 2007.