A comparison of Dadda and Wallace multiplier delays

Whitney J. Townsend,[*] Earl E. Swartzlander, Jr.,[**] and Jacob A. Abraham[*]

[*] Computer Engineering Research Center, The University of Texas at Austin
[**] Department of Electrical and Computer Engineering, The University of Texas at Austin
Austin, TX 78712

ABSTRACT

The two well-known fast multipliers are those presented by Wallace and Dadda. Both consist of three stages. In the first stage, the partial product matrix is formed. In the second stage, this partial product matrix is reduced to a height of two. In the final stage, these two rows are combined using a carry propagating adder. In the Wallace method, the partial products are reduced as soon as possible. In contrast, Dadda's method does the minimum reduction necessary at each level to perform the reduction in the same number of levels as required by a Wallace multiplier. It is generally assumed that, for a given size, the Wallace multiplier and the Dadda multiplier exhibit similar delay. This is because each uses the same number of pseudo adder levels to perform the partial product reduction. Although the Wallace multiplier uses a slightly smaller carry propagating adder, usually this provides no significant speed advantage. A closer examination of the delays within these two multipliers reveals this assumption to be incorrect. This paper presents a detailed analysis for several sizes of Wallace and Dadda multipliers. These results indicate that despite the presence of the larger carry propagating adder, Dadda's design yields a slightly faster multiplier.

Keywords: Dadda multipliers, Wallace multipliers, multiplier delay

1. INTRODUCTION

The two well-known fast multipliers are the column compression multipliers presented by Wallace [1] and Dadda [2]. Both of these multipliers consist of three stages. In the first stage, the partial product matrix is formed. In the second stage, this partial product matrix is reduced to a height of two. In the final stage, these two rows are combined using a carry propagating adder. In the Wallace method, the partial products are reduced as soon as possible. In contrast, Dadda's method does the minimum reduction necessary at each level to perform the reduction in the same number of levels as required by a Wallace multiplier.

Because each method uses the same number of pseudo adder levels to perform the partial product reduction it is generally assumed that, for a given size, the Wallace multiplier and the Dadda multiplier exhibit similar delay. However, as the Wallace multiplier requires a smaller carry propagating adder, it is sometimes assumed to be the faster of the two methods [3]. A closer examination of the delays within these two multipliers reveals this assumption to be incorrect. This paper presents a detailed analysis for several sizes of Wallace and Dadda multipliers. These results indicate that despite the presence of the larger carry propagating adder, Dadda's design yields a slightly faster multiplier.

This paper considers unsigned multipliers with multiplicands and multipliers of equal size. Baugh and Wooley [4] have presented the modifications required to use signed operands with column compression multipliers. The remainder of this paper is organized as follows. Sections 2 and 3 review Dadda's multiplier design methodology and Wallace's multiplier design methodology respectively. Section 4 describes the gate level examination used to compare these two methodologies for delay and area. Section 5 presents the results of calculations for Dadda and Wallace multipliers of varying operand sizes. Section 6 provides conclusions.

Further author information:
W.J.T.: E-mail: whitney@cerc.utexas.edu, Address: Computer Engineering Research Center, The University of Texas at Austin, 1 University Station C8800, Austin, TX 78712

2. DADDA MULTIPLIERS

Dadda multipliers are a refinement of the parallel multipliers first presented by Wallace in 1964 [1]. Thus both multiplier methodologies consist of three stages. The partial product matrix is formed in the first stage by N^2 AND gates. In the dot diagram notation developed by Dadda [2] each partial product is represented by a dot. The dot diagram for an 8 by 8 Dadda multiplier is shown in Figure 1. The eight rows of eight dots each at the top of the figure represent the partial product matrix formed by the AND gates.

In the second stage the partial product matrix is reduced to a height of two. Dadda replaced Wallace's pseudo adders in this stage with parallel (n,m) counters. A parallel (n,m) counter is a circuit which has n inputs and produce m outputs which provide a binary count of the number of ONEs present at the inputs. A full adder is an implementation of a (3,2) counter which takes 3 inputs and produces 2 outputs. Similarly a half adder is an implementation of a (2,2) counter which takes 2 inputs and produces 2 outputs. Although other sizes of counters are possible as discussed by Dadda [5], this paper considers Dadda and Wallace multipliers with compression trees consisting only of (3,2) and (2,2) counters.

Dadda multipliers use a minimal number of (3,2) and (2,2) counters at each level during the compression to achieve the required reduction. The reduction procedure for Dadda compression trees is given by the following recursive algorithm [6].

1. Let $d_1 = 2$ and $d_{j+1} = \lfloor 1.5 \cdot d_j \rfloor$. D_j is the height of the matrix for the j^{th} stage. Repeat until the largest j^{th} stage is reached in which the original N height matrix contains at least one column which has more than d_j dots.
2. In the j^{th} stage from the end, place (3,2) and (2,2) counters as required to achieve a reduced matrix. Only columns with more than d_j dots or which will have more than d_j dots as they receive carries from less significant (3,2) and (2,2) counters are reduced.
3. Let $j = j - 1$ and repeat step 2 until a matrix with a height of two is generated. This should occur when $j = 1$.

The dot diagram shown in Figure 1 shows this algorithm implemented for an 8 by 8 multiplier. Four reduction levels are required with matrix heights of 6, 4, 3, and 2. In the figure, two dots joined by a diagonal line indicate that these two dots are the outputs from a (3,2) counter. Similarly two dots joined by a crossed diagonal line indicate that these two dots are the outputs from a (2,2) counter. 64 AND gates, 35 (3,2) counters, 7 (2,2) counters, and a 14-bit carry propagating adder are required to form the 16-bit product.

The number of (3,2) and (2,2) counters required for a Dadda multiplier depends on N, the number of bits of the operands, and is determined as follows [7].

$$\text{(3,2) counters} = N^2 - 4 \cdot N + 3$$
$$\text{(2,2) counters} = N - 1$$

Dadda's scheme for placing these counters was determined to be optimal by Habibi and Wintz [8]. Dadda multipliers require fewer (3,2) and (2,2) counters during the compression stage than do the corresponding Wallace multipliers.

Once the matrix has been reduced to a height of two, the final stage consists of using a carry propagating adder to produce the final product. The size of the final carry propagating adder is determined as follows [7].

$$\text{CPA length} = 2 \cdot N - 2$$

Within this paper Ripple Carry Adders (RCA) and Carry Lookahead Adders (CLA) will be considered as final carry propagating adders.

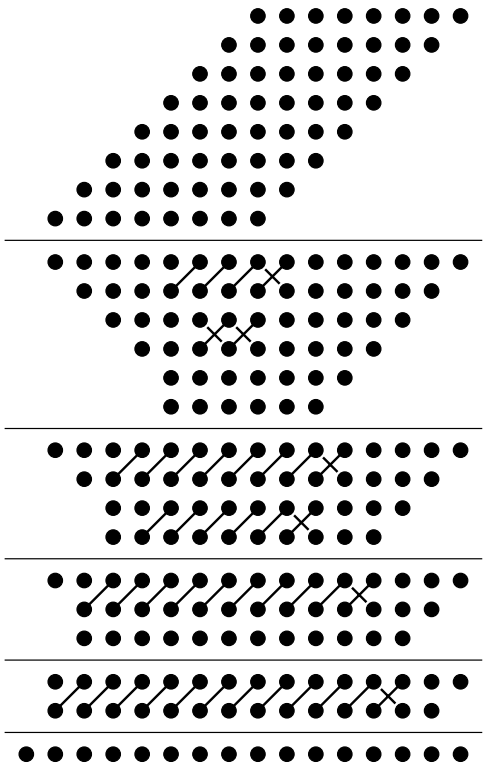

Figure 1. Dot Diagram for an 8 by 8 Dadda Multiplier

3. WALLACE MULTIPLIERS

For Wallace multipliers the partial products are formed by N^2 AND gates in the same manner as for Dadda multipliers. Next the N rows of partial products are grouped together in sets of three rows each. Any additional rows that are not a member of a group of three are transferred to the next level without modification. Within each group of three rows, (3,2) counters are applied to the columns containing three bits and (2,2) counters are applied to the columns containing two bits. Columns containing only a single bit are transferred to the next level unchanged. The height of the matrix in the jth reduction stage, w_j is given by the following recursive equations [9].

$$w_0 = N$$
$$w_{j+1} = 2 \cdot \lfloor \frac{w_j}{3} \rfloor + w_j \; mod \; 3$$

As for the Dadda multipliers, when the matrix has been reduced to a level with a height of two, a carry propagating adder is used to perform the final addition whose sum is the product of the multiplication. Wallace and Dadda multipliers each require the same number of levels to perform the reduction to a level with a height of two, however, the heights of the different levels can vary between the two methodologies. Although Wallace and Dadda multipliers contain nearly identical numbers of full adders, more of the Wallace full adders are applied during the reduction of the matrix. This and the additional half adders used in a Wallace reduction result in the shorter final carry propagating adder.

A dot diagram for an 8 by 8 Wallace multiplier is shown in Figure 2. Four reduction stages are required with matrix heights of 6, 4, 3, and 2. 64 AND gates, 1 OR gate, 38 (3,2) counters, 15 (2,2) counters, and a 10-bit carry propagating adder are required to form the 16-bit product.

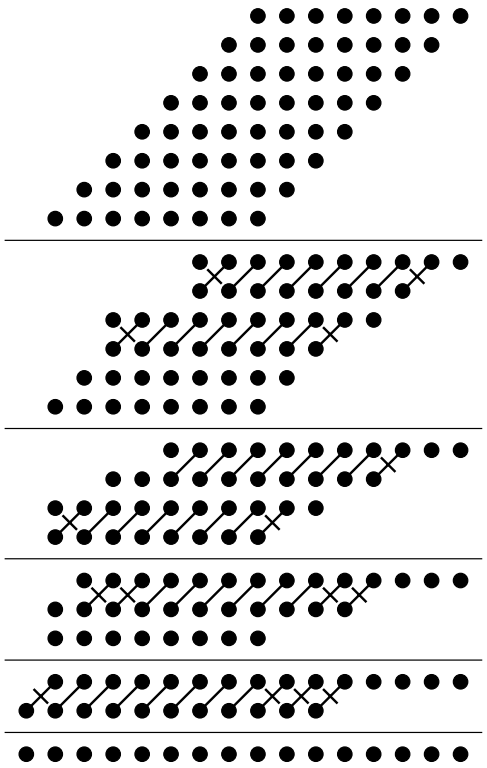

Figure 2. Dot Diagram for an 8 by 8 Wallace Multiplier

The number of (3,2) counters and the size of the final carry propagating adder required for a Wallace multiplier depends on N, the number of bits of the operands, and S, the number of stages in the reduction, and can be determined as follows [7].

$$3 \leq N \leq 5$$

$$\text{(3,2) counters} = N^2 - 4 \cdot N + 3 + S$$
$$\text{CPA length} = 2 \cdot N - 2 - S$$

$$5 < N$$

$$\text{(3,2) counters} = N^2 - 4 \cdot N + 2 + S$$

or

$$\text{(3,2) counters} = N^2 - 4 \cdot N + 1 + S$$
$$\text{CPA length} = 2 \cdot N - 1 - S$$

The number of (2,2) counters required by a Wallace multiplier is either equal to or greater than N. While always at least N, this number is often much greater than N and results in Wallace multipliers typically requiring more gates, thus more area, than the corresponding Dadda multipliers despite the smaller final carry propagating adder.

4. DELAY METHODOLOGY

Typically when the delays of Dadda and Wallace multipliers are compared, the number of reduction levels is compared as well as the size of the final carry propagating adder. Each reduction level is considered to have a delay equivalent to the delay of a (3,2) counter and the final carry propagating adder the appropriate delay for its type. Using this methodology the two designs are assumed either to exhibit similar delay or the Wallace multiplier is assumed to be faster due to the slightly smaller carry propagating adder. With closer examination these assumptions are revealed as incorrect.

The multiplier designs considered here are composed of (3,2) counters which are composed of nine gate full adders [10]. These gates are limited to 2-input AND and 2-input OR gates as well as inverters. The (2,2) counters are implemented using four 2-input gates. Due to the limitation on the number of inputs to the gates, each gate will be considered to be roughly equivalent in speed and area and gate counts will be used to determine relative speed and area for the two designs.

On the left side of Figure 3 is the dot diagram for a 4 by 4 Dadda multiplier. The top matrix representing the formation of the partial products contains sixteen dots. On the right side of Figure 3 is the corresponding delay diagram for this multiplier. As each partial product has been formed by a two-input AND gate each dot is represented by a 1 which gives the delay through each AND gate in forming the partial product matrix.

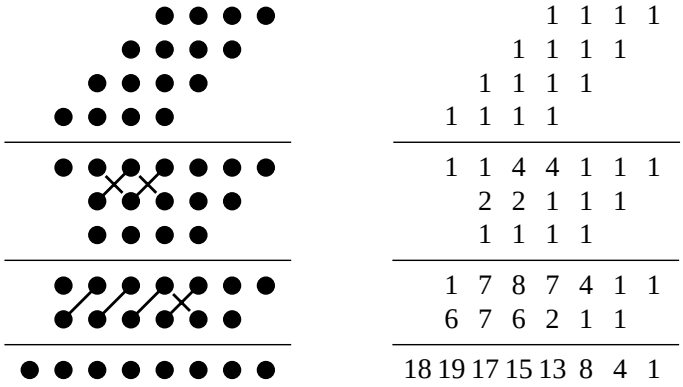

Figure 3. Dot Diagram for a 4 by 4 Dadda Multiplier and the Corresponding Delay Diagram

The first reduction level in a 4 by 4 Dadda multiplier is to a height of three. This requires two (2,2) counters to be applied, one to the column which contains four dots, and the other to the column on its left which has only three dots but which will receive an additional dot from the first (2,2) counter. These two (2,2) counters are represented in the dot diagram on the left. The delay through a (2,2) counter is 3 gate delays for the least significant bit which corresponds to the sum for a half adder and 1 gate delay for the most significant bit which corresponds to the carry for a half adder, (assuming that all inputs arrive simultaneously). Thus in the delay diagram on the right of Figure 3 the two 4s in the first row represent the least significant bits from the two (2,2) counters and the two 2s in the second row represent the most significant bits from the two (2,2) counters. All of the other partial products have been passed to the second level unchanged as reflected by the ONEs in the matrix.

The second reduction level is to a height of two. This requires one (2,2) counter and three (3,2) counters as shown in the dot diagram. The delay through a (3,2) counter is 6 gate delays for the least significant bit which corresponds to the sum for a full adder and 5 gate delays for the most significant bit which corresponds to the carry for a full adder, (assuming that all inputs arrive simultaneously). The 2 and the 4 in the third matrix represent the outputs of the (2,2) counter. The pair, 7 and 6, immediately to the left of the 4 and 2 represent the output of the (3,2) counter on the right which has inputs with delays of 1, 1, and 4. The pair, 7 and 8, represent the output of the (3,2) counter in the middle which has inputs with delays of 1, 2, and 4. The output of the left (3,2) counter which has inputs with delays of 1, 2, and 1 is the pair, 7 and 6, on the left side of the matrix. Five of the partial products are brought to this level unchanged represented by the ONEs.

A 6-bit RCA completes the multiplication. The inputs to the half adder arrive at 1 and 1 producing a carry at 2 and the sum bit at 4. The carry at 2 and the unmodified partial product bit are used by the next full adder to produce a carry at 7 and a sum bit at 8. The carry continues to ripple through the final adder and the corresponding sum bit arrival times are shown for each bit position. The overall delay of the 4 by 4 Dadda multiplier is found to be 19 gate delays using this methodology.

On the left side of Figure 4 is the dot diagram for a 4 by 4 Wallace multiplier. The top matrix representing the formation of the partial products contains sixteen dots. On the right side of Figure 4 is the corresponding delay diagram for this multiplier. As before for the Dadda multiplier each dot is represented by a 1 which gives the delay through the AND gates which form the partial products.

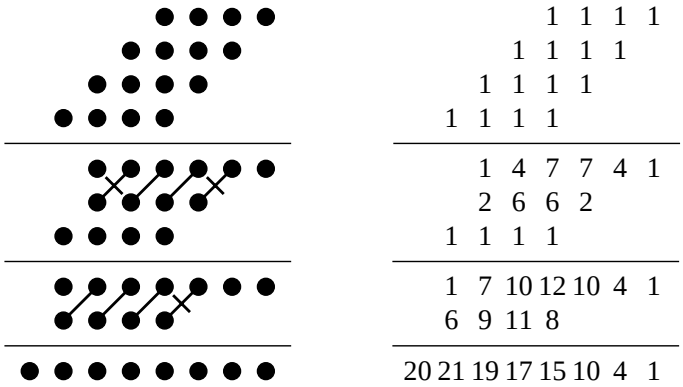

Figure 4. Dot Diagram for a 4 by 4 Wallace Multiplier and the Corresponding Delay Diagram

The first reduction level in a 4 by 4 Wallace multiplier is to a height of three. Grouping the top three rows of the partial product matrix results in two columns containing three dots, two columns containing two dots, and two columns containing a single dot. This grouping requires two (3,2) counters to be applied to those columns which contain three dots and two (2,2) counters to be applied to those columns which contain two dots. These counters are represented in the dot diagram on the left side of Figure 4. In the corresponding delay diagram on the right side of Figure 4 the two pairs of 4 and 2 are the outputs of the two (2,2) counters and the two pairs of 6 and 7 are the outputs of the two (3,2) counters. The two ONEs in the top row correspond to the columns in the grouping which had single dots and were unchanged. At the bottom of the matrix is the fourth row from the partial product matrix which has been brought to this level unmodified.

The second reduction level is to a height of two. The three rows form one grouping which requires one (2,2) counter and three (3,2) counters. The pair, 8 and 10, correspond to the outputs of the (2,2) counter which has inputs of 2 and 7. The pair, 11 and 12, correspond to the outputs of the (3,2) counter on the right which has inputs of 1, 6, and 7. The pair, 9 and 10, correspond to the outputs of the (3,2) counter in the middle which has inputs of 1, 6, and 4. The pair, 6 and 7, correspond to the outputs of the (3,2) counter on the left which has inputs of 1, 2, and 1. Two partial products from the original partial product matrix remain unchanged represented by the two ONEs.

A 4-bit RCA completes the multiplication. The first three low order bits of the product are already computed. The inputs to the half adder are 8 and 12 which produce a sum at 15 and a carry at 12. The carry ripples through the adder producing the corresponding sum bit arrival times as shown for each bit position. The overall delay of the 4 by 4 Wallace multiplier is found to be 21 gate delays using this methodology.

The delay diagram for the 8 by 8 Dadda multiplier whose dot diagram was shown in Figure 1 is shown in Figure 5. The top matrix representing the formation of the partial products contains sixty-four ONEs, comprised of eight rows of eight ONEs each. The first reduction level in the 8 by 8 Dadda multiplier is to a height of six as shown. The next reduction level is to a height of three and illustrates the use of the following heuristics for the interconnections between the levels.

							1	1	1	1	1	1	1	1	
						1	1	1	1	1	1	1	1		
					1	1	1	1	1	1	1	1			
				1	1	1	1	1	1	1	1				
			1	1	1	1	1	1	1	1					
		1	1	1	1	1	1	1	1						
	1	1	1	1	1	1	1	1							
1	1	1	1	1	1	1	1								
1	1	1	1	1	7	7	7	4	1	1	1	1	1	1	
	1	1	1	6	6	6	2	1	1	1	1	1	1		
		1	1	1	1	4	4	1	1	1	1	1			
			1	1	2	2	1	1	1	1	1				
				1	1	1	1	1	1	1					
				1	1	1	1	1	1						
1	1	1	7	9	12	12	10	7	7	4	1	1	1	1	
	1	6	8	11	11	9	6	6	2	1	1	1	1		
		1	1	7	7	7	7	7	4	1	1	1			
		1	6	6	6	6	6	2	1	1	1				
1	1	7	12	13	14	13	12	12	8	7	4	1	1	1	
	6	11	12	13	12	11	11	7	6	2	1	1	1		
	1	6	8	11	12	12	10	7	7	4	1	1			
1	9	14	18	19	18	18	17	15	13	10	7	4	1	1	
8	13	17	18	17	17	16	14	12	9	6	2	1	1		
36	37	35	33	31	29	27	25	23	21	19	16	13	8	4	1

Figure 5. Delay Diagram for an 8 by 8 Dadda Multiplier with a Ripple Carry Final Adder

1. The delay numbers in any given column are considered from lowest to highest regardless of their row position within the column.
2. Once the number of (3,2) and (2,2) counters has been determined from the dot diagram, inputs are assigned from lowest to highest delay values, first to (3,2) and then to (2,2) counters in each column.
3. If any column has remaining dots left to be transferred to the next level unchanged, those delay values with the highest numerical values are the ones that should be transferred.

One example application of these heuristics can be seen in the second reduction level in Figure 1. In the sixth column from the left, two (3,2) counters are placed. The first (3,2) counter receives inputs at gate delays of 1, 1, and 1 and produces outputs at gate delays of 6 and 7. The second (3,2) counter receives inputs at gate delays of 2, 6, and 7 and produces outputs at gate delays of 11 and 12. An application of the last heuristic can be seen in the third reduction level. In the fifth column from the right, a (3,2) counter is placed. It receives the inputs at gate delays of 1, 1, and 1 producing outputs at gate delays of 6 and 7 while the remaining data in the column with a gate delay value of 4 is transferred to the next level unchanged.

The delay diagram for the 8 by 8 Wallace multiplier whose dot diagram was shown in Figure 2 is shown in Figure 6.

5. RESULTS

Dadda and Wallace multipliers with operand sizes of 4, 8, 16, and 32 bits are compared within this section. The multipliers containing ripple carry adders for the final carry propogating adder are composed entirely of two-input AND gates, two-input OR gates, and inverters. Table 1 presents the delay comparisons for all four sizes of multipliers containing ripple carry final adders. Despite the presence of longer carry propagating adders

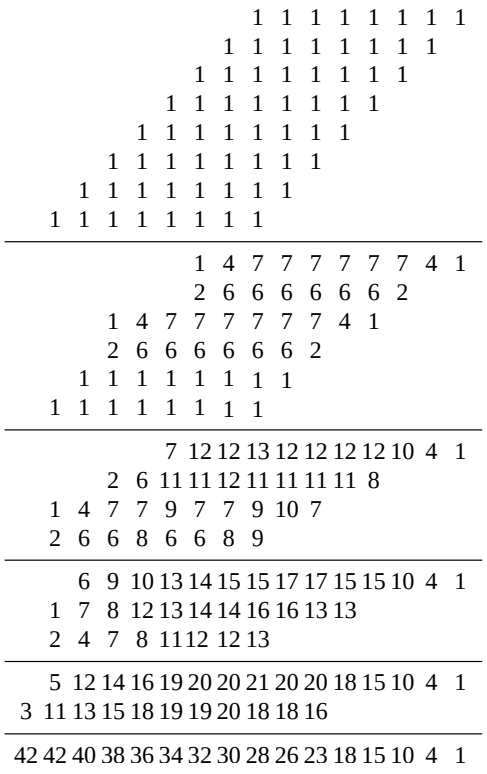

Figure 6. Delay Diagram for an 8 by 8 Wallace Multiplier with a Ripple Carry Final Adder

within the Dadda multipliers, each is faster than the corresponding Wallace multiplier by 9% - 14%. Table 1 also provides the corresponding complexity comparison for these multipliers. For the smallest pair of multipliers, the results are the same. However, as the size increases there becomes a 5% increase in the number of gates required to implement the Wallace multiplier over the Dadda multiplier. Thus Wallace multipliers containing ripple carry final adders are found to be both larger and slower than the corresponding Dadda multipliers with respect to gate delay calculations and the number of gates required for implementation.

Table 1. Delay and Complexity Comparisons for Multipliers with RCAs

Multiplier Size	Dadda Delay	Wallace Delay	Dadda Complexity	Wallace Complexity
4 by 4	19 (100%)	21 (111%)	104 (100%)	104 (100%)
8 by 8	37 (100%)	42 (114%)	528 (100%)	552 (105%)
16 by 16	69 (100%)	77 (112%)	2336 (100%)	2476 (106%)
32 by 32	133 (100%)	145 (109%)	9792 (100%)	10283 (105%)

Multipliers containing carry lookahead adders for the final carry propogating adder are also considered. These multipliers are composed only of two-input AND gates, two-input OR gates, and inverters for those portions of the circuit which form the partial products and for the compression, but these multipliers also contain three-input AND gates, three-input OR gates, four-input AND gates, and four-input OR gates within the final adder portion of the circuit. Modified full adders containing only two-input gates as described in [10] are used within the carry-lookahead blocks to generate the initial generates and propagates. Thus four bit carry lookahead blocks are used although they add a slightly optimistic bias to the delay and complexity results if these multipliers are compared with the ripple carry versions.

Table 2 presents the delay comparisons for all four sizes of multipliers containing carry lookahead adders. The

Dadda multipliers are slightly faster than the corresponding Wallace multipliers for each size considered despite the larger carry lookahead adders required. For the smallest pair of multipliers, the Dadda multiplier requires two levels of carry lookahead logic, while the Wallace multiplier requires only one. Despite this, all of the Dadda multipliers retain a slight advantage. Table 2 also provides the corresponding complexity comparison for these multipliers. Due to the extra level of carry lookahead logic for the smallest multiplier, the Dadda multiplier requires more gates than the corresponding Wallace multiplier. For all of the larger multipliers however, the Wallace multipliers again require approximately 5% more gates than the corresponding Dadda multipliers. Thus Dadda multipliers are found to be faster than the corresponding Wallace multipliers for all sizes considered and to require fewer gates for every size except the smallest.

Table 2. Delay and Complexity Comparisons for Multipliers with CLAs

Multiplier Size	Dadda Delay	Wallace Delay	Dadda Complexity	Wallace Complexity
4 by 4	15 (100%)	18 (120%)	120 (100%)	112 (93%)
8 by 8	29 (100%)	31 (107%)	573 (100%)	582 (102%)
16 by 16	43 (100%)	45 (105%)	2440 (100%)	2557 (105%)
32 by 32	54 (100%)	56 (104%)	10013 (100%)	10475 (105%)

6. CONCLUSIONS

This paper has presented a gate level comparison of Dadda and Wallace multiplier areas and delays. Although it has generally been assumed that a Wallace multiplier yields a slightly faster design due to its smaller final adder, a closer examination of the delays within these two multipliers considered at the gate level rather than at the full adder level has proven this assumption to be incorrect. Results have been presented for multipliers of varying operand sizes with both ripple carry and carry lookahead final adders which confirm that Dadda multipliers are both faster and smaller than the corresponding Wallace multipliers.

ACKNOWLEDGMENTS

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

REFERENCES

1. C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. on Computers,* vol. 13, pp. 14–17, 1964.
2. L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza,* vol. 34, pp. 349–356, 1965.
3. B. Parhami, *Computer Arithmetic Algorithms and Hardware Designs*, New York: Oxford University Press, 2000.
4. C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. on Computers,* vol. 22, pp. 1045–1047, 1973.
5. L. Dadda, "On parallel digital multipliers," *Alta Frequenza,* vol. 45, pp. 574–580, 1976.
6. E. E. Swartzlander Jr., "Merged arithmetic," vol. 29, pp. 946–950, 1980.
7. K. A. C. Bickerstaff, M. Schulte, and E. E. Swartzlander Jr., "Reduced area multipliers," *Intl. Conf. on Application-Specific Array Processors*, pp. 478–489, 1993.
8. A. Habibi and P. A. Wintz, "Fast multipliers," *IEEE Trans. on Computers,* vol. 19, pp. 153–157, 1970.
9. K. A. C. Bickerstaff, E. E. Swartzlander Jr., and M. J. Schulte, "Analysis of column compression multipliers," *15th IEEE Symp. on Computer Arithmetic*, pp. 33–39, 2001.
10. E. E. Swartzlander, Jr. and G. Goto, "Computer arithmetic," *The Computer Engineering Handbook*, V. G. Oklobdzija, ed., Boca Raton, FL: CRC Press, 2002.