

# The Proposed Full-Dadda Multipliers

Vikas Kaushik\* and Himanshi Saini

Department of Electrical and Computer Engineering, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, India

## Review Article

Received date: 16/01/2018

Accepted date: 23/04/2018

Published date: 30/04/2018

### \*For Correspondence

Vikas Kaushik, Department of Electrical and Computer Engineering, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, India.

**E-mail:** willingvikas@gmail.com

**Keywords:** Tree multiplier, Interconnects area, Comparative analysis, Full Dadda, Digital multiplier, Partial product reduction, Compressor, Wallace-tree

### ABSTRACT

Dadda multipliers require less area and are slightly faster than Wallace tree multipliers. Among tree multipliers, Dadda multiplier is the most popular multiplier. This paper presents a new tree multiplier named Full-Dadda which has a new reduction scheme. This novel reduction scheme results in less number of bit-interconnects and less interconnect area than those in Dadda multiplier. This proposed reduction scheme is also a superior one in terms of regularity, simplicity and alternative use. This regularity and simplicity do not require any extra hardware. The comparative performance analysis of the Dadda and the Full-Dadda multipliers with different operand sizes is presented in this paper. Figures and tables are given to illustrate above advantages of the proposed multiplier.

## INTRODUCTION

Multiplier is one of the most important circuits for designing computers and computing devices. The Dadda technique for partial product reduction is based on the idea of 'avoid use of full adder.' But the use of full adder is more regular in other Wallace tree multipliers. This paper presents a modified Dadda technique based on the idea of 'prefer the use of full adder over the use of half adder.' Only the last stage that is 'three to two reductions' is the exception. This idea used in the modified technique makes it more regular and simple. Hence, based on the idea, it is named as 'Full-Dadda'. The fact behind saying this technique an alternative approach is that this technique results in same number of full adders, half adders, same size of final carry propagation adder and same number of compressors (adders) at each stage as required in the Dadda technique. Therefore the proposed multiplier can be used in place of Dadda multiplier in all its applications. This paper presents a comparative performance analysis of the proposed multiplier with the Dadda multiplier. For this comparison each multiplier with different operand sizes is taken. The main disadvantages of the Dadda multiplier are: (i) it is less regular, (ii) more complex and (iii) it reduces less number of bits at early stages of reductions. The proposed 'Full-Dadda' multiplier is more regular, simple and reduces more number of bits at early stages of summand reduction.

In section 2, there is brief overview of literature and in section 3; there are general rules and equations for reduction scheme and number of hardware components respectively. The comparison between multipliers is arranged in five sub-sections under section 4.

## LITERATURE REVIEW

The parallel multipliers are faster and more fancied<sup>[1]</sup>. There are many ways and schemes available for multiplication<sup>[1]</sup>, such

as Array multiplication scheme, Booth multiplication and Vedic multiplication etc. Among these schemes, the tree multiplication scheme is one of the most popular schemes [1-9]. The most popular multiplier among tree multipliers is the Dadda multiplier [2]. A Reduced Area multiplier is also the best optimized multiplier in terms of Area if interconnects are properly managed [3]. The basic steps involved in a tree multiplication scheme, first used by Wallace [6] in 1963 are:

(1) Generation of the partial products [3] (or summands [5,6] or matrix of partial products [1]).

(2) Partial products reduction: reduction in column height by using pseudo adders (there are many parallel addition schemes [8] and partial product reduction schemes [9] implemented by using pseudo additions such as conditional sum addition [7] etc.).

(3) Use of final CPA (carry propagation adder): After the last stage of column-height reduction (remaining bits in a column are only two), there is a need of final addition of two rows to obtain final product of multiplied operands.

The size and type of final CPA used plays an important role in determining overall performance such as delay, area and power consumption [2,3,5].

### DADDA MULTIPLIERS

Dadda method for partial product reduction uses only necessary reduction determined by the Wallace table shown in **Table 1** [2-4]. It uses an approach to increase the number of half adders and to decrease the number of full adders required for overall reduction of partial products. The detail of Dadda technique is given in the references [2,5,9].

The Dadda method keeps as these are all right columns having height less than or equal to the necessary bits required at a stage after reduction. This is repeated at every stage.

**Table 1.** Number of reduction stages according to number of bits in the column of maximum height in partial products matrix.

Bits in a multiplier (N)	Number of stages needed
3	1
4	2
5 to 6	3
7 to 9	4
10 to 13	5
14 to 19	6
20 to 28	7
29 to 42	8
43 to 63	9
64 to 94	10

### THE PROPOSED MULTIPLIER

This multiplier uses a modified Dadda technique for partial product reduction. The main difference between the two lies in the partial product reduction method. Dadda’s method prefers use of half adders whereas the proposed Full-Dadda multiplier uses full adder with preference except at last stage.

### GENERAL RULES

The rules for partial products reduction used in the proposed multiplier are explained in a simple way as follows:

(1) Keep to the next stage as these are all right columns having height less than or equal to the necessary bits required at a stage after reduction. These necessary bits are determined according to **Table 1**. This is similar step as in Dadda method. It is done at every stage.

(2) For the reduction of partial products, use full adders that is (3, 2) compressors preferably over the use of (2, 2) compressor (half adder). Use half adders only when full adders can’t be used that is only when two bits are available for further reduction.

(3) At last stage of reduction where 3 bits to 2 bits reduction is to be done; use half adder preferably. This last stage rules are also the same as of Dadda method. The height of a column at this stage is 3. Therefore, there is no need of long counting of the column bits and hence it is easy to give preference to half adder.

### GENERAL EQUATIONS

For a multiplier of size N by N, the total number of full adders, half adders and size of CPAs used in the proposed multiplier is given by the following equations:

- Number of Half Adders = N-1 and

- Number of Full Adders =  $(N-1)*(N-3)$  (this is for 3 by 3 and above sizes).
- Size of CPAs (in bits) =  $2N-2$

### COMPARISON BETWEEN PROPOSED REDUCTION AND DADDA REDUCTION SCHEMES

Figure 1 show the notations used in this paper for summand bit, half adder and full adder. The reduction schemes for Dadda and Proposed multiplier of 4 by 4, 6 by 6 and 9 by 9 operand sizes are shown in Figures 2-7 respectively. The following three facts are clear from these Figures: (1) the final CPA size is equal for both multipliers of same operand sizes; (2) the number of half adders and the number of full adders are same in both multipliers of same operand sizes, (3) at each stage, both multipliers have equal number of compressors for summands reduction and (4) in the proposed technique, there is requirement of less number of bits to be transferred to the next stage.



Figure 1. Notations for summand bit, half adder ((2, 2) compressor) and full adder ((3, 2) compressor).

It is clear from Figures 2 and 3 that 4 by 4 Dadda-reduction scheme requires 2 half adders at first stage and 3 full adders and one half adder at second stage whereas in 4 by 4 proposed-reduction scheme requires 2 full adders at first stage and 1 full adder with 3 half adders at second stage. In 4 by 4 Dadda, the number of bits required transferring from first stage to second and second to CPA are 16 and 13 respectively whereas in 4 by 4 proposed, these are 14 and 13 respectively. The reduced number of bits in proposed scheme is due to use of full adders at initial stages. The number of used-bits and unused-bits at first stage in 4 by 4 Dadda, are 4 and 12 respectively whereas in 4 by 4 proposed, are 6 and 10 respectively. Similarly we can find these numbers for other stages and for other operand-sizes multipliers. The figures for 6 by 6 and 9 by 9 operand-size multipliers are given in Figures 4-7 respectively.

The tables for comparison are arranged under different subsections. These are categorized on the basis of: (i) hardware component required (ii) number of bits transferred and (iii) area of interconnects. The five advantages of the proposed multiplier are explained in following five sub-sections.

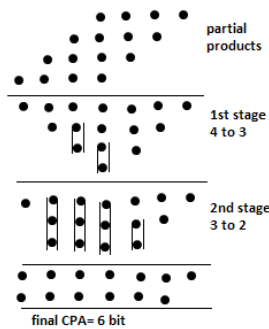


Figure 2. 4 by 4 Dadda multiplier reduction scheme.

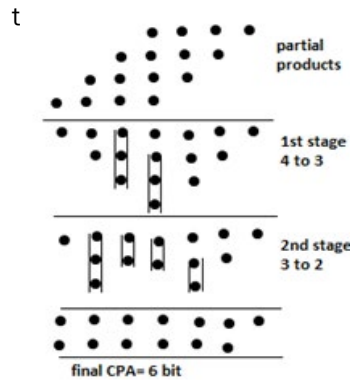


Figure 3. 4 by 4 proposed multiplier reduction scheme.

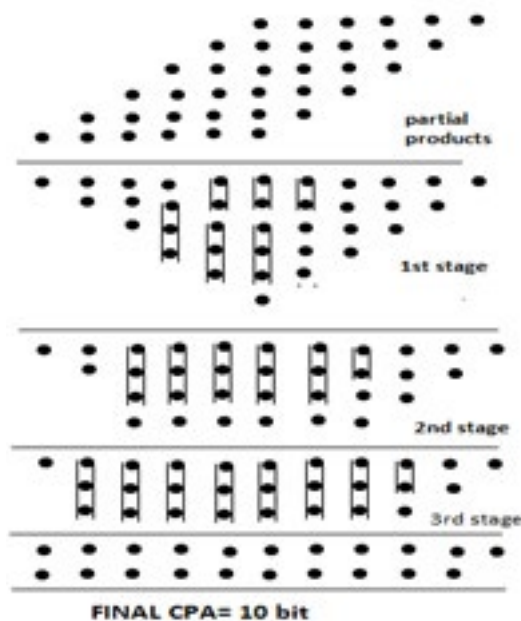


Figure 4. 6 by 6 Dadda multiplier reduction scheme.

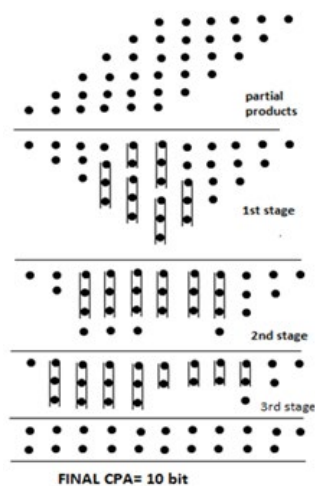


Figure 5. 6 by 6 proposed multiplier reduction scheme.

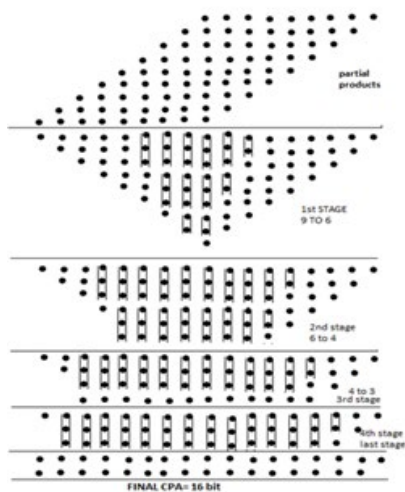


Figure 6. 9 by 9 Dadda multiplier reduction scheme.

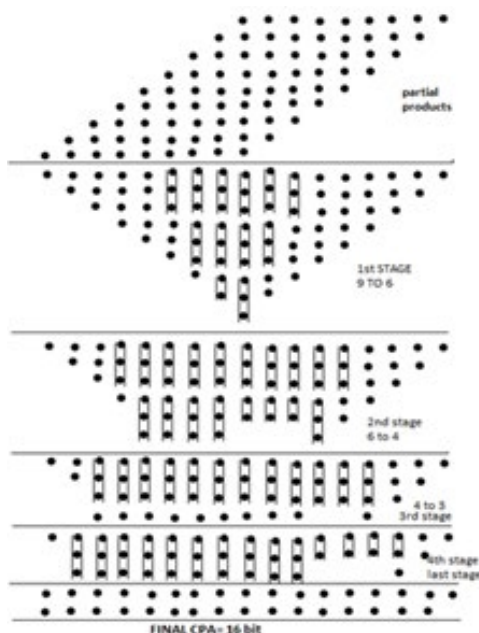


Figure 7. 9 by 9 proposed multiplier reduction scheme.

### REGULARITY

The regularity lies in the fact that the proposed approach is more similar to other approaches used in designing of most of the tree multipliers, e.g.; Wallace tree multipliers. The proposed technique for summands reduction uses full adder ((3, 2) compressor) preferably as the other tree multipliers do. But Dadda technique uses (2, 2) compressors preferably which makes it less regular.

### SIMPLICITY

The simplicity lies in the fact that if there is a need of further reduction in a column height (according to Wallace reduction table), and if there are only two bits remaining in a column for further reduction, only then use half adders ((2,2) compressor) otherwise use full adder. But the Dadda technique uses half adder even if there are three or more than three bits are available in a column for further reduction. This makes it necessary to count precisely every time the height of column with previous carries and to decide about the use of half adder even if there is availability of three or more bits in further grouping for reduction. This is somewhat difficult because our practice in designing other tree multipliers is not similar to this.

### ALTERNATIVE APPROACH

This proposed approach is an alternative to Dadda’s approach because the same number of hardware units such as half adders, full adders and same size of CPA and also the same number of compressors at each stage are used in both Dadda and Proposed techniques. No extra hardware is required which makes the proposed multiplier capable of replacing the Dadda multiplier in all its applications.

Table 2 shows all types of hardware components are equal in number in both the multipliers. Table 3 shows that at each respective stage of the two multipliers, there are equal number of (3, 2) and (2, 2) compressors.

Table 2. Total number of half adders (HA), full adders (FA) and size of CPA required in Dadda and proposed multipliers.

	DADDA Multiplier			PROPOSED Multiplier		
	HA	FA	CPA-Size (bit)	HA	FA	CPA-Size (bit)
<b>4 by 4</b>	3	3	6	3	3	6
<b>6 by 6</b>	5	15	10	5	15	10
<b>8 by 8</b>	7	35	14	7	35	14
<b>9 by 9</b>	8	48	16	8	48	16
<b>12 by 12</b>	11	99	22	11	99	22
<b>16 by 16</b>	15	195	30	15	195	30

<b>24 by 24</b>	23	483	46	23	483	46
<b>32 by 32</b>	31	870	62	31	870	62
<b>64 by 64</b>	63	3843	126	63	3843	126

**Table 3.** Stage-wise comparison of number of compressors (HA + FA).

Stage No. →	DADDA Multiplier					PROPOSED Multiplier				
	1	2	3	4	5	1	2	3	4	5
<b>4 by 4</b>	2	4	-	-	-	2	4	-	-	-
<b>6 by 6</b>	6	6	8	-	-	6	6	8	-	-
<b>8 by 8</b>	6	14	10	12	-	6	14	10	12	-
<b>9 by 9</b>	12	18	12	14	-	12	18	12	14	-
<b>12 by 12</b>	12	30	30	18	20	12	30	30	18	20

### LESS NUMBER OF BITS TO BE TRANSFERRED

This is the disadvantage of Dadda multiplier that it reduces less number of summands-bits at early stages and as a result of this, large number of summands-bits are required to send from one stage to next stage. Interconnection between stages also plays an important role in deciding overall area, delay and power consumption of a tree-multiplier [3]. **Table 4** shows total number of bits passing through successive stages. From this table, it is clear that the proposed multiplier transfers less number of total bits to the next stage. From a stage, three types of bits go to next stage. These are (1) Sum bit: obtained at pseudo adder sum-output, (2) Carry bit: obtained at pseudo adder carry-output and (3) Unused summands-bits: not fed as an input to any compressor; only transferred as it is to the next stage. It is better if minimum number of unused bits are there to send to next stage because more bits means more number of interconnects. The Proposed multiplier uses less number of interconnects than that required in Dadda multiplier. This is an advantage of the proposed multiplier over the Dadda multiplier. This advantage shows that the proposed multiplier has not only an alternative approach but also has an improved architecture.

### LESS AREA OF INTERCONNECTS

The estimation of interconnects-area for Dadda and proposed multipliers of different operand sizes is shown in **Table 5**. This area in **Table 5** is the relative area obtained on the basis of the LSI logic standard cell data book used in reference [3]. According to this book, at a stage, if area of used summand bit-interconnect is equal to 1 then area of unused summand bit-interconnect is equal to 2. The hardware difference between the Proposed and the Dadda multipliers is only in terms of interconnects because all other hardware components are equal in number. For example, AND array for summands generation, final CPA size and number of half adders and full adders etc. are exactly matching in both types of multipliers of same operand sizes. It is clear from **Table 5** that the proposed multipliers have less interconnect area than that of the respective Dadda multipliers. This decreased area in the proposed Full-Dadda multiplier would enhance its performance.

**Table 4.** Number of bits passing through successive stages S= sum bits, C= carry bits, U=unused-bits at previous stage, T= total number of bits.

Stage Transfer	DADDA Multiplier				PROPOSED Multiplier				
	S	C	U	T	S	C	U	T	
<b>4 by 4</b>	1 <sup>st</sup> to 2 <sup>nd</sup>	2	2	12	16	2	2	10	14
	2 <sup>nd</sup> to CPA	4	4	5	13	4	4	5	13
<b>6 by 6</b>	1 <sup>st</sup> to 2 <sup>nd</sup>	6	6	21	33	6	6	19	31
	2 <sup>nd</sup> to 3 <sup>rd</sup>	6	6	16	28	6	6	13	25
	3 <sup>rd</sup> to CPA	8	8	5	21	8	8	5	21

<b>8 by 8</b>	1 <sup>st</sup> to 2 <sup>nd</sup>	6	6	49	61	6	6	46	58
	2 <sup>nd</sup> to 3 <sup>rd</sup>	14	14	21	49	14	14	19	47
	3 <sup>rd</sup> to 4 <sup>th</sup>	10	10	20	40	10	10	17	37
	4 <sup>th</sup> to CPA	12	12	5	29	12	12	5	29
<b>9 by 9</b>	1 <sup>st</sup> to 2 <sup>nd</sup>	12	12	49	73	12	12	46	70
	2 <sup>nd</sup> to 3 <sup>rd</sup>	18	18	21	57	18	18	19	55
	3 <sup>rd</sup> to 4 <sup>th</sup>	12	12	22	46	12	12	19	43
	4 <sup>th</sup> to CPA	14	14	5	33	14	14	5	33
<b>12 by 12</b>	1 <sup>st</sup> to 2 <sup>nd</sup>	12	12	112	136	12	12	108	132
	2 <sup>nd</sup> to 3 <sup>rd</sup>	30	30	49	109	30	30	46	106
	3 <sup>rd</sup> to 4 <sup>th</sup>	30	30	21	81	30	30	19	79
	4 <sup>th</sup> to 5 <sup>th</sup>	18	18	28	64	18	18	25	61
	5 <sup>th</sup> to CPA	20	20	5	45	20	20	5	45

**Table 5.** Interconnects area (area estimation).

For 4 by 4 multipliers	Stage Number	Number of Used-Bits	Number of Unused-Bits	Stage Area
DADDA	1	4	12	$4+12*2 = 28$
	2	11	5	$11+5*2 = 21$
<b>TOTAL AREA</b>				49
PROPOSED	1	6	10	$6+10*2=26$
	2	9	5	$9+5*2=19$
<b>TOTAL AREA</b>				45
Percentage Difference (45=100%)				8.89%
For 6 by 6 multipliers	Stage Number	Number of Used-Bits	Number of Unused-Bits	Stage Area
DADDA	1	15	21	$15+21*2= 57$
	2	17	16	$17+16*2= 49$
	3	23	5	$23+5*2=33$
<b>TOTAL AREA</b>				139
PROPOSED	1	17	19	$17+19*2=55$
	2	18	13	$18+13*2=44$
	3	20	5	$20+5*2=30$
<b>TOTAL AREA</b>				129
Percentage Difference (129=100%)				7.75%
For 8 by 8 multipliers	Stage Number	Number of Used-Bits	Number of Unused-Bits	Stage Area
DADDA	1	15	49	113
	2	40	21	82
	3	29	20	69
	4	35	5	45
<b>TOTAL AREA</b>				309
PROPOSED	1	18	46	110
	2	39	19	77
	3	30	17	64
	4	32	5	42
<b>TOTAL AREA</b>				293
Percentage Difference (293=100%)				5.46%

For 9 by 9 multipliers	Stage Number	Number of Used-Bits	Number of Unused-Bits	Stage Area
DADDA	1	32	49	130
	2	52	21	94
	3	35	22	79
	4	41	5	51
<b>TOTAL AREA</b>				<b>354</b>
PROPOSED	1	35	46	127
	2	51	19	89
	3	36	19	74
	4	38	5	48
<b>TOTAL AREA</b>				<b>338</b>
Percentage Difference (338=100%)				4.73%
For 12 by 12 multipliers	Stage Number	Number of Used-Bits	Number of Unused-Bits	Stage Area
DADDA	1	32	112	256
	2	87	49	185
	3	88	21	130
	4	53	28	109
	5	59	5	69
<b>TOTAL AREA</b>				<b>749</b>
PROPOSED	1	36	108	252
	2	86	46	178
	3	87	19	125
	4	54	25	104
	5	56	5	66
<b>TOTAL AREA</b>				<b>725</b>
Percentage Difference (725=100%)				3.31%

### CONCLUSION

In comparison to the Dadda scheme, the proposed Full-Dadda multiplier has not only an alternative, simple and more regular scheme for summands reduction but also has an improved architecture. The improvement lies in the fact that in the proposed multiplier, less number of bits are needed to pass through successive stages and the total area required by the bit-interconnects is less than that required in the Dadda multiplier. This decrement in area of the proposed multiplier is due to less number of unused bits at initial stages of the proposed multiplier. The future scope of this multiplier lies in the fact that it is a basic unit of many other big devices used in computing, signal processing, electronic and data communication etc. fields. The bigger devices can be designed by using this proposed multiplier. This paper also gives an insight of analyzing methods for tree multipliers which can be used in designing new multiplier. A suggestion for the layout designer and HDL programmer is that there are three types of bits in summands which are not equally fast. Give proper preference to bits during reduction. Otherwise the obtained result will not be optimized one. For example, sum-bits are faster than the carry-bits for simple full adders.

### REFERENCES

1. Stenzel WJ, et al. A Compact High-Speed Parallel Multiplication Scheme. IEEE Transactions on Computers. 1977;26:948-957.
2. Waters RS, et al. A reduced complexity Wallace multiplier reduction. IEEE Transactions on Computers. 2010;59:1134-1137.
3. KAC Bickerstaff, et al. Reduced area multipliers. International Conference on Application Specific Amy Processors. 1993;478-489.
4. Bickerstaff KC, et al. Analysis of column compression multipliers. IEEE Symposium on Computer Arithmetic. 2001:33-39.
5. Habibhi A, et al. Fast multipliers. IEEE Transactions on Computers. 1969;19:153-157.
6. Wallace CS. A suggestion for a fast multiplier. IEEE Transactions on Electronic Computers. 1964;13:14-17.



7. Sklansky J. Conditional-sum addition logic. IRE Transactions on Electronic Computers. 1960;9:226-231.
8. Brent RP, et al. A regular layout for parallel adders. IEEE Transactions on Computers. 1982;31:260-264.
9. C Wesley, et al. Implementation of a high speed multiplier using carry lookaheads. Signals, Systems and Computers. 2013.