ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Design and Implementation of Signed, Rounded and Truncated Multipliers using Modified Booth Algorithm for Dsp Systems

K. Ram Prakash[1]
Professor
Department of Electronics and Communication
Kumarauru College of Technology
Coimbatore - India

A. V. Sanju[2]
PG scholar
Department of Electronics and Communication
Kumarauru College of Technology
Coimbatore - India

Abstract: ***Multipliers have a significant impact in the performance of the entire Dsp system. Many high-performance algorithms and architectures have been proposed to accelerate multiplication without increasing the hardware. In previous papers, the truncation error is reduced by adding error compensation circuits. In this paper, truncation error is not more than 1 ULP (unit of least position). So there is no need of error compensation circuits. Truncated multipliers offer significant improvements in area, delay, and power. The modified booth algorithm helps to reduce the number of partial products to be generated. It is known to be the fastest multiplication algorithm. Partial products bits were compressed by using operations such as deletion, reduction, truncation, rounding and final addition. Thus the resultant multipliers shows a good performance which were used in various high-speed applications. The proposed method reduces the number of full adders and half adders during the tree reduction. The proposed method experimentally shows a reduction of area and power.***

Keywords: Truncated Multiplier; Truncated Error; Booth Algorithm; High Speed Application.

I. INTRODUCTION

Multiplication of two numbers generates a product with twice that of the original bit width. To reduce area cost, leading to the design of truncated multipliers it is desirable to truncate the product bits to the required [1][2][3][4] or fixed-width multipliers[5][6][7][8][9][10]. Fixed-width multipliers are commonly applied to digital signal processor operations such as filtering, convolution, and fast Fourier or discrete cosine transform. There are mainly two truncated multiplier design methods specifically constant and variable corrections. It depends on significant partial product (PP) bits i.e., how to recompense the error occurs due to the elimination of the least significant partial product (PP) bits. Constant correction designs systematically compute the average truncation error of the truncated PP bits and estimate the error by adding a row of constant into the matrix of the PP bits[1][3]. Variable correction designs will reduce the total truncation error in view of the least significant part of the PP bits [2][10]. Most of the fixed-width multiplier is designed to focus on different compensation approaches to reduce various error metrics such as maximum absolute, average, and mean-square errors. There are new truncated multiplier designs that can accomplish realistic rounding results. In this paper, it is jointly considers the tree reduction, truncation, and rounding of the PP bits during the design of fast parallel truncated multipliers so that the final truncated product satisfies the precision requirement. The multipliers have a significant impact on the performance of the entire system. Many high-performance algorithms and architectures have been used to accelerate multiplication. Various multiplication algorithms such as Booth modified Booth, Braun, Baugh-Wooley taken into consideration. The modified booth algorithm helps to reduce the number of partial products to

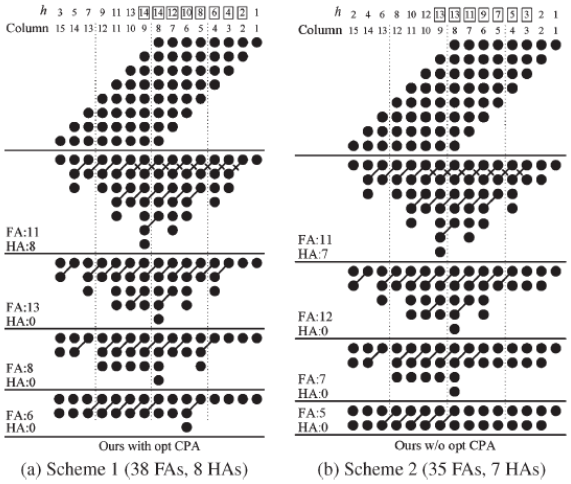
be generated. It is known as the fastest multiplication algorithm. Thus the resultant multipliers shows a good performance which is used in the very high-speed applications.

II. TREE REDUCTION OF PARALLEL MULTIPLIES

A parallel tree multiplier design typically consists of three major steps, i.e., partial products generation, partial products reduction, and final carry propagate addition. Partial products generation will produce partial products bits from the multiplicand and the multiplier. The main goal of partial product reduction is to compact the number of partial products to two, and summed up by the final addition. The two most well-known reduction methods are Wallace tree [13] and Dadda tree [14] reductions. Wallace tree reduction manages to constrict the partial products as early as possible, whereas Dadda reduction only performs compression if it is necessary without increasing the number of carry-save addition (CSA) levels. There are two reduction schemes that intend to minimize the use of half adders (HAs) in each column because the full adder (FA) cell has a higher compression rate compared with that of Half Adder cell. In tree reduction of parallel multipliers, hybrid Scheme-1 and Scheme-2 reductions were adopted for the truncated multiplier design in order to minimize the area cost. Fig. 1(a) and (b) shows the reduction procedures by Scheme 1 and Scheme 2 to each column of partial product bits, starting from least significant Column.

TABLE 1:
Number of Full Adders and Half Adders for the reduction of h bits in one column.

# of bits (height) before reduction	# of bits after reduction = 1 (Scheme 1)		# of bits after reduction = 2 (Scheme 2)	
	# of carry bits = $n_{FA}+n_{HA}$	n_{HA}	# of carry bits = $n_{FA}+n_{HA}$	n_{HA}
2*	1	1	-	
3	1	0	1	1
4	2	1	1	0
5	2	0	2	1
6	3	1	2	0
7	3	0	3	1
8	4	1	3	0
9	4	0	4	1
⋮	⋮		⋮	
h	$\lfloor h/2 \rfloor$	$(h-1) \bmod 2$	$\lfloor (h-1)/2 \rfloor$	$h \bmod 2$



(a) Scheme 1 (38 FAs, 8 HAs) (b) Scheme 2 (35 FAs, 7 HAs)

Fig. 1. Tree reduction of 8 × 8 multiplication

Scheme 1 in Table I is used to study whether a half adder is needed. It is not compulsory that the number of bits after the reduction is always one.

Table II. Cost comparison of various reduction methods.

	cost	Dadda	Wallace	RA [15]	Scheme 2	Scheme 1
8 × 8 Multiplier	levels	4	4	4	4	4
	FA	35	38	39	35	38
	HA	7	15	7	7	8
	CPA (bits)	14	11	10	14	10
16 × 16 Multiplier	levels	6	6	6	6	6
	FA	195	200	201	195	200
	HA	15	52	15	15	16
	CPA (bits)	30	25	24	30	24
7-operands (3-bit each)	levels	4	4	4	3	3
	FA	11	12	13	11	12
	HA	5	4	2	3	3
	CPA (bits)	5	4	3	5	3
20-operand (6-bit each)	levels	7	7	7	6	6
	FA	101	101	104	101	103
	HA	10	19	4	3	4
	CPA (bits)	10	9	7	10	7

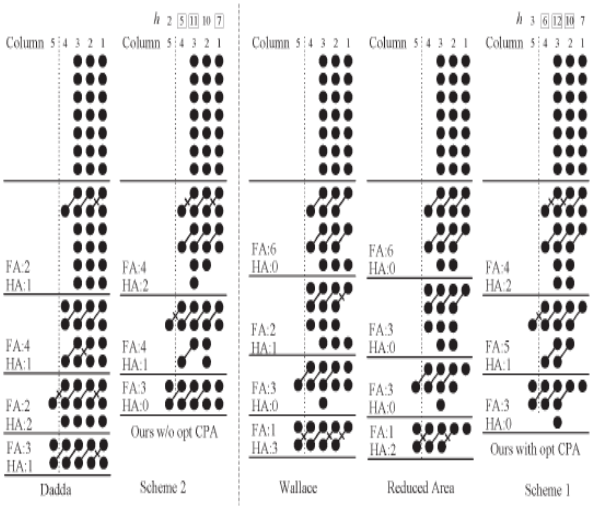


Fig. 2. Reduction of seven-operand addition with 3-bit erands.

Table II compares different reduction methods. For multiplication, the Scheme 2 leads precisely to the same results as that of the Dadda method. Scheme 1 has better reduction efficiency with a minimum CPA bit width when compared with the Wallace method and has almost the same results compared with the RA method [15]. For multi operand additions, reduction methods could achieve better results. It is easy to combine Scheme-1 and Scheme-2 reductions for different columns for the truncated multiplier design.

III. TRUNCARED MULTIPLIER DESIGN

Let as assume $n \times n$ unsigned multiplication of two fractional numbers design is to estimate the

$$Z = \sum_{k=1}^{2n} z_k = X * Y = \sum_{i=1}^{n} x_i 2^{-i} * \sum_{j=1}^{n} y_i 2^{-i} \quad \textbf{(1)}$$

P MSBs of the product with a maximum truncation error of no more than and only P MSBs of the product are kept, as shown in Fig. 3. The objective of the truncated multiplier 1 ULP, where 1 ULP = 2^{-p}.

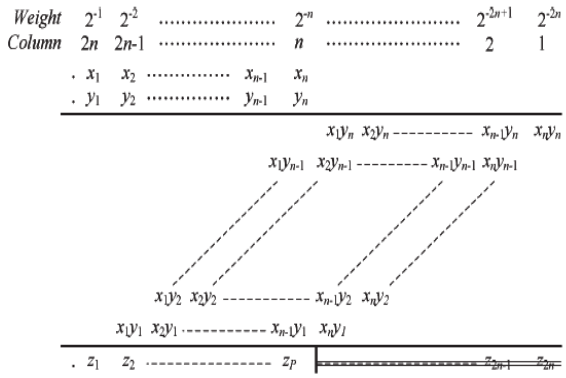


Fig. 3. Unsigned multiplication of two fractional numbers

The truncated multiplier consists of several operations including deletion, reduction, truncation, rounding, and final addition.

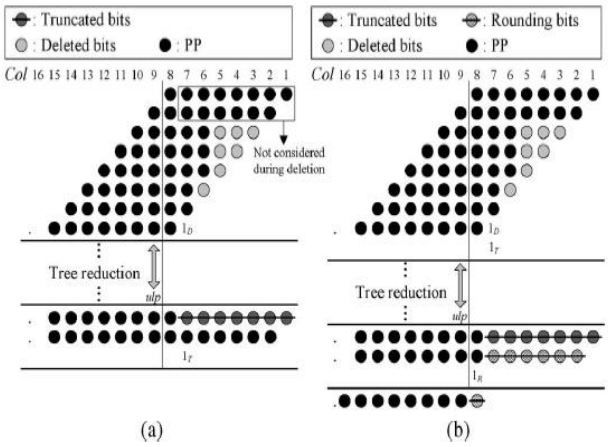


Fig 4. 8 × 8 truncated multiplication with eight product bits truncated. (a) Deletion, reduction, and truncation of PP bits. (b) Deletion, reduction, truncation, and rounding plus final addition.

A. Deletion, Reduction, Truncation

In the first step, deletion which removes all the redundant partial product bits that do not need to be generated, as shown by the gray dots in Fig. 4(a) the deletion error after the bias adjustment is

$$-1/4\text{ulp} \leq ED \leq 1/4\text{ulp}. \qquad (2)$$

The reduction generates two rows of partial product bits in per-column reduction of Scheme 2, as mentioned in Section II. The truncation further removes the first row of n − 1 bits from column 1 to column n – 1. The adjusted truncation error is bounded by

$$-1/4\text{ulp} < \text{ET} \leq 1/4\text{ulp} \qquad (3)$$

B. Rounding and Final Addition

The PP bits are added using a CPA to generate the final product of *P* bits, as shown in Fig. 4(b).

In the final CPA, Adding a bias constant of 1/2 ulp, in order to attain the round-to nearest rounding with the rounding error i.e.

$$-1/2\text{ulp} < ER \leq 1/2\text{ulp} \qquad (4)$$

The total error for the design of the realistically rounded truncated multiplier is bounded by

$$-\text{ulp} < E = (ED + ET + ER) \leq \text{ulp} \qquad (5)$$

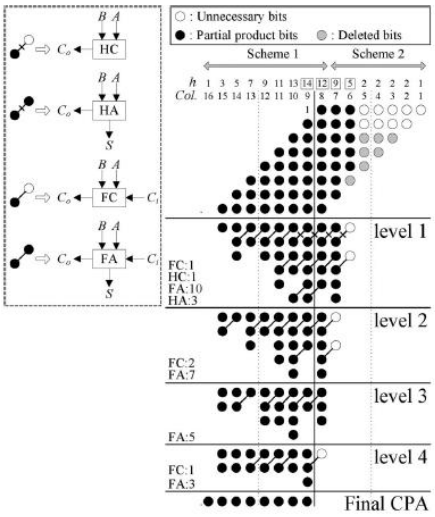


Fig. 5. Example of designing 8 × 8 unsigned fractional multiplication faithfully truncated to eight fractional bits ($T = 8$, $P = 8$).

In the stage 1 partial product will undergone for deletion process there by pp will added correspondingly using gates. First two rows are unchanged during the process. These bits will participate in subsequent reduction, truncation and rounding process as shown in fig.5.

C. Modified booth Multiplier

The modified Booth algorithm reduces the number of partial products to be generated and is known as the fastest multiplication algorithm. Multiplication consists of three steps: the first step to generate the partial products; the second step to add the generated partial products until the last two rows are remained; the third step to compute the final multiplication results by adding the last two rows. In this paper the modified Booth encoding (MBE) used as a proposed scheme [12]. It is identified as the most efficient Booth encoding and decoding scheme. Table I shows the rules to generate the encoded signals by MBE scheme and Fig. 5 (a) shows the corresponding logic diagram. Using the encoded signals the Booth decoder can generates the partial products as shown in fig (b).

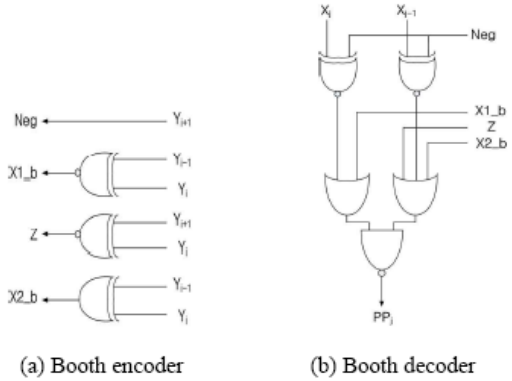


Fig. 6 Encoder and decoder for MBE scheme

Fig. 7 shows the generated partial products and sign extension scheme [16] of the 8-bit modified Booth multiplier. The partial products generated by the modified Booth algorithm are added in parallel using the Wallace tree.

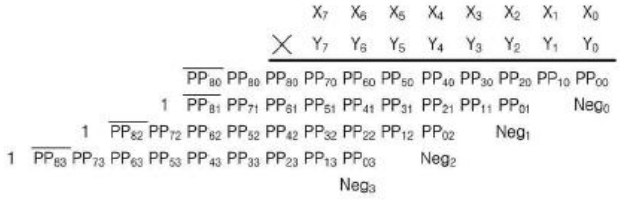


Fig. 7 Generated partial products and sign extension scheme

The architecture of the modified Booth multiplier as shown in fig 8. The inputs of the multiplier are multiplicand X and multiplier Y. The last two rows are added to generate the final multiplication results using the carry look-ahead adder (CLA).

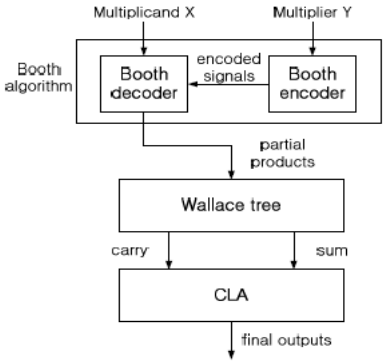


Fig. 8 Architecture of the modified Booth multiplier

IV. RESULTS

The booth algorithm is implemented to the 8 X 8 multiplier and the result proved to be fruitful. The overall result approximately reduced by 50% from the previous algorithms. This also reduced the overall hardware components significantly over the previous one. The comparison of the improvements is tabulated in table 1. The simulation results are shown in Fig1 and Fig 2.

Table III
Result comparison of the existing one and the previous one.

EXISTING ALGORITHM	MODIFIED BOOTH ALGORITHM
Capable of doing 8 X 8 unsigned integer	Capable to do 8 X 8 signed integer
Results in 8 Partial Products	Results in 4 partial products
Total cache count: 828	Total cache count: 631
Memory usage: 178 MB	Memory usage: 176 MB
Power consumption : 47mW	Power consumption: 44 m W

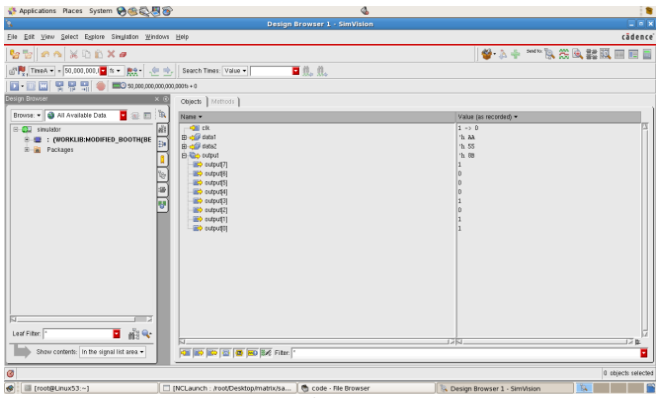


Fig 8: Design browser for 8 X 8 signed multiplier.

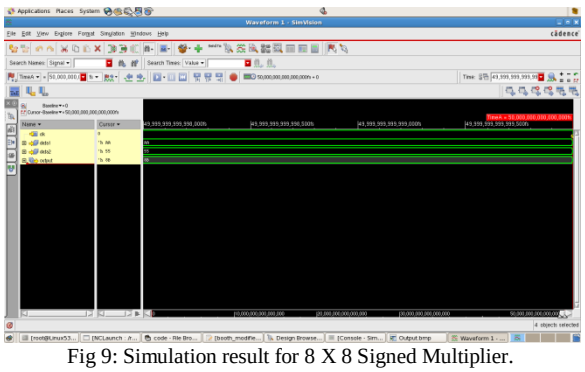


Fig 9: Simulation result for 8 X 8 Signed Multiplier.

V. CONCLUSION

The Fast Truncated multipliers with modified both algorithm to carry out the signed multiplication in along with deletion, truncation, rounding reduction techniques shows significant improvement in the area consumption and power dissipation. The proposed multiplier will be applied to an fir filter to show the optimum usage in various dsp applications. The Truncated multiplier can be designed with optimum gates using cadence backend tool to show further improment in area and power dissipation. The cadence layout optimization can be taken as future extension of current proposed multiplier design.

References

1. M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in VLSI Signal Processing VI. Piscataway, NJ: IEEE Press, 1993, pp. 388–396.
2. E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," in Proc. 31st Asilomar Conf. Signals, Syst. Comput.,1997, pp. 1178–1182.
3. M. J. Schulte, J. G. Hansen, and J. E. Stine, "Reduced power dissipation through truncated multiplication," in Proc. IEEE Alessandro Volta Memorial Int. Workshop Low Power Des.,1999, pp. 61–69.
4. J. E. Stine and O. M. Duverne, "Variations on truncated multiplication," in Proc. Euromicro Symp. Digit. Syst. Des., 2003, pp. 112–119.
5. J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixed-width multipliers for DSP applications," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 46, no. 6, pp. 836–842, Jun. 1999.
6. L.-D. Van and C.-C. Yang, "Generalized low-error area-efficient fixedwidth multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.
7. T.-B. Juang and S.-F. Hsiao, "Low-error carry-free fixed-width multipliers with low-cost compensation circuits," IEEE Trans. Circuits Syst. I, Reg.Papers, vol. 52, no. 6, pp. 299–303, Jun. 2005.
8. A. G.M. Strollo, N. Petra, and D. De Caro, "Dual-tree error compensation for high-performance fixed-width multipliers," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 52, no. 8, pp. 501–507, Aug. 2005.
9. N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G.M. Strollo, "Truncated binary multipliers with variable correction and minimum mean square error," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.
10. J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified booth multipliers for lossy applications," in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Jan. 2011, vol. 19, no. 1, pp. 52–60.
11. J.-A. Pineiro, S. F. Oberman, J. M. Muller, and J. D. Bruguera, "Highspeed function approximation using a minimax quadratic interpolator," IEEE Trans. Comput., vol. 54, no. 3, pp. 304–318, Mar. 2005.
12. E. G. Walters and M. J. Schulte, "Efficient function approximation using truncated multipliers and squarers," in Proc. 17th IEEE Symp. ARITH,2005, pp. 232–239.
13. C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
14. L. Dadda, "Some schemes for parallel multipliers," Alta Frequenza, vol. 34, pp. 349–356, 1965.
15. K. C. Bickerstaff, M. Schulte, and E. E. Swartzlander, Jr., "Parallel reduced area multipliers," J. VLSI Signal Process., vol. 9, no. 3, pp. 181–191, 1995.
16. M. D. Ercegovac and T. Lang, Digital Arithmetic, Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 2003
17. C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. On Computers, vol. BC13, pp. 14-17, Feb. 1964.
18. www.wikipedia.com

