Area Efficient Fixed Width Modified Booth Multiplier

E. L. K. Priya[1], Prof. P. Lakshmi Sarojini[2], G. Rajesh Kumar[3]
Swarnandhra College of Engineering & Technology[1,2], Vishnu Institute of Technology[3]

Abstract

This paper proposes the design of area efficient fixed-width modified booth multipliers for lossy applications such as multimedia applications. Area efficiency can be obtained by omitting hardware circuit required to generate the lower half part of the partial product matrix of Booth multiplier which leads to truncation error. Accuracy can be improved by way of modifying the partial product matrix of Booth multiplication and subsequently deriving an effective error compensation function which is composed of basic gates.In addition a simple compensation circuit composed of basic gates, derived from sorting network is also proposed for further improvement of accuracy. Compared to the Post Truncated Multiplier (PTM), the proposed Fixed width Multiplier requires very less hardware.The generated partial products addition is implemented using Dadda tree for minimization ofthe delay and area. To reduce the area, the adder and Exclusive OR are implemented using 11 transistors and 6 transistors respectively.

Keywords: *Error Composition circuit, fixed-width multiplier, modified booth multiplier, Partial product matrix, Error compensation function, Post Truncated Multiplier*

1. Introduction

Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. In parallel multipliers, high performance can be achieved by using modified Booth encoding ([1]-[3]), which decreases the number of partial products by a factor of 2 through multiplier recording. Fixed word size in lossy systems that allow small accuracy loss to output data, can be maintained by using n×n fixed width multipliers which result in only *n* most significant product bits. Complexity reduction in hardware and significant area and power savings can be achieved by removing the adder cells of standard multiplier for the computation of the *n* less significant bits of *2n*-bit output product. However, the fallout being high truncation error will be introduced into the system to such kind of direct-truncated fixed-width multiplier.

Variety of error compensation methods that add estimated compensation value to inputs of the reserved adder cells, which effectively arrests the truncation error. As it is universally known that error compensation value can be derived by the Adaptive/Constant scheme. Regardless of current input data value influence, the Constant Scheme ([4],[5]) arrives to constant error compensation value and same is used to carry inputs of the retained adder cells while performing multiplication operations.Less hardware results in a relatively large truncation error to the Constant Scheme.Adaptive scheme ([6]-[10]) has been designed such that high accuracy is achieved by adjusting the compensation value according to the input data at the cost of higher hardware complexity. However, most of the adaptive error compensations are developed only for Fixed-width array multipliers & has less influence on reduction of truncation error for a fixed width modified booth multipliers directly.

Various error compensation approaches ([11], [12]), have been proposed toeffectively scale down the truncation errors of Fixed-width modified Booth Multiplier at the cost of hardware complexity. Simple error compensation circuit results in better error performance and area with booth encoded outputs as inputs to generate the error compensation value. A systematic design methodology for area efficient fixed width modified booth multiplier through exploring the influence of various indices in a binary threshold was developed to reduce the product error can be seen in [13].

Many DSP applications and Multimedia, the output data has direct bearing to the accumulation of a series of products over a single multiplication operation. In

those cases, truncation errors results in large output error, which can be countered by performing additional compensation, which isagain dependent on different applications needing different compensation values.
Thus to mitigate such kind of customizations, to decrease the accumulated output error, fixed-width multipliers with very minute error with simple errorcompensation circuit are extensively preferred to obtain more accurate output data.

Through this Paper simple error compensation circuit for fixed-width modified Booth multiplier is proposed which reduces most of the hardware. In order to achieve desired outcome, slight modification of the partial product matrix of Booth multiplication to reduce the partial product bits in the truncated portion of DTFM is required. Thereby, we ascertain the correlation between the Booth encoded outputs and the truncated product error of DTFM is analysed to derive a simple yet effective compensation function, which can result in an approximation to the carry value generated by truncated portion of DTFM, to reduce the errors result in truncation and make the error distribution as symmetric and centralized as possible. Subsequently, developing of simple modified sorting network along with some adder cells as per theproposed error compensation function. Results derived from implementation and simulation depict that the proposed fixed-width modified Booth multiplier occupies less area and achieves approximately same accuracy as PTMwhich is most accurate existing fixed width modified Booth multipliers in terms of the mean error and the mean-square error, all of this still by maintaining the approximate Hardware overhead .

2. Booth Multiplier Architecture

In Booth multiplier Multiplication operation when considered between multiplicand and the multiplier namely A & B respectively, representation of which is depicted as below:

$$A = -a_{n-1}2^{2n-1} + \sum_{j=0}^{n-2}(a_j 2^j) \quad (1)$$

$$B = -b_{n-1}2^{2n-1} + \sum_{j=0}^{n-2}(b_j 2^j) \quad (2)$$

Modified Booth encoding groups the multiplier bits into triplets, and are encodedas given in table I, then B can be expressed as:

$$B = \sum_{j=0}^{\frac{n}{2}-1}(-2b_{2j} + b_{2j} + 2b_{2j})2^{2j} = \sum_{j=0}^{\frac{n}{2}-1} K_j 2^{2j} \quad (3)$$

Where $b_{-1}=0$ and k_j belongs to the set of values {-2, -1,0,1,2}.

Table 1. Modified Booth encoding table

Booth encoder i/p	Operation	Booth encoder o/p
$b_{2j+1}b_{2j}b_{2j-1}$		$n_j t_j o_j z_j c_j$
0 0 0	+0	0 0 0 1 0
0 0 1	+A	0 0 1 0 0
0 1 0	+A	0 0 1 0 0
0 1 1	+2A	0 1 0 0 0
1 0 0	-2A	1 1 0 0 1
1 0 1	-A	1 0 1 0 1
1 1 0	-A	1 0 1 0 1
1 1 1	-0	1 0 0 1 0

According to the Modified Booth encoding table the existing booth encoder and partial product generation circuit are shown in fig. 1.a and fig.1.b respectively.

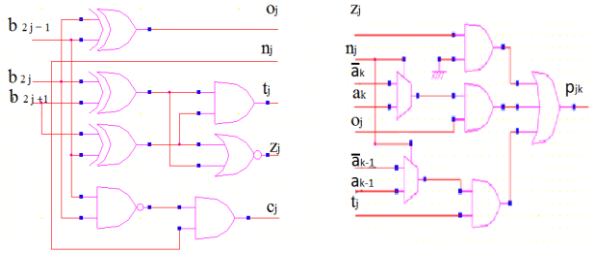

Fig 1.a) Booth encoder circuit b)PP generation circuit

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PP0						$\bar{s}_0$	s_0	s_0	$p_{0,7}$	$p_{0,6}$	$p_{0,5}$	$p_{0,4}$	$p_{0,3}$	$p_{0,2}$	$p_{0,1}$	$p_{0,0}$
PP1					1	$\bar{s}_1$	$p_{1,7}$	$p_{1,6}$	$p_{1,5}$	$p_{1,4}$	$p_{1,3}$	$p_{1,2}$	$p_{1,1}$	$p_{1,0}$		c_0
PP2				1	$\bar{s}_2$	$p_{2,7}$	$p_{2,6}$	$p_{2,5}$	$p_{2,4}$	$p_{2,3}$	$p_{2,2}$	$p_{2,1}$	$p_{2,0}$		c_1	
PP3		1	$\bar{s}_3$	$p_{3,7}$	$p_{3,6}$	$p_{3,5}$	$p_{3,4}$	$p_{3,3}$	$p_{3,2}$	$p_{3,1}$	$p_{3,0}$		c_2			
PP4										c_3						
	P_{15}	P_{14}	P_{13}	P_{12}	P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Fig 2. Partial product matrix of Booth multiplier

With relation to negation operation, each bit of multiplicand *A* is complemented and an extra binary value '1' is added to least significant bit pertaining to next partial product row. To implement correction bit c_j addition of '1' is being used and thereby indicating the partial product row PP_j as positive (c_j=0) or negative (c_j=1). As each partial product row is represented in two's complementation, the sign bit for each PP_j must be properly extended upto the $(2n-1)^{th}$ bit position. Many ways [14] are proposed to give a solution to eliminate the problem of sign extension bits as they affect the performance and power consumption of the parallel multiplier with large values of *n*. The partial product matrix of an 8×8 modified booth multiplier with sign extension elimination technique is illustrated in Fig2, where in $p_{j,k}$ and s_j denote the k^{th} product bit and the sign bit of partial product row PP_j, respectively.

3. Area efficient Fixed Width Modified Booth multiplier

The partial product matrix is bifurcated into two segments namely MSP(Most Significant Part) & LSP(Least Significant Part), where LSP is further bifurcated into LSP_{minor} and LSP_{major} as shown in fig. 3 (except '1').A truncated multiplier is an *n*× *n* multiplier with *n* bits output. As the partial products divided into two subsets LSP includes the n less significant columns of the partial product matrix, while the MSP includes the remaining columns the full-width multiplier output, P is given by(4)where S(MSP) and S(LSP) represent the weighted sum of the elements of MSP and LSP respectively.

$$P = S(MSP) + S(LSP) \quad (4)$$

In the direct truncated multiplier the partial products of the LSP are discarded assuming that their contribution to the *n* most significant bits of the output is negligible. This solution is very advantageous in terms of hardware performances. The cells for the LSP matrix are not present and the final circuit halves the number of cells compared the full-width one. However, a straightforward analysis shows that the direct elimination of the partial products of the LSP causes a very big error bounded by $n(2-1)$LSB.

In the post-truncated modified Booth multiplier (PTM) the with the addition of an extra 'l'(carry value by LSP_{minor}) at the $(n-1)^{th}$ bit position of partial product matrix as depicted in Fig 3 (including'1')then outputs the highly significant *n* product bits. With this the most accurate error compensation value $\hat{e}$ is generated by PTM. Then final multiplication result *P* of *A* and *B* can be expressed as:

$$P = S(MSP) + S(LSP_{major}) + S(LSP_{minor}) + 1 \quad (5)$$

(in$(n-1)^{th}$ position)

The main disadvantage of PTM is approximate hardware complexity and power consumption to the standard modified Booth Multiplier.

As Partial product bits are generated from the booth encoder outputs, exploration of the relation between outputs of the booth encoders and the carry propagated from LSP_{minor} to LSP_{major} is to be made. Then, to reduce the truncation error, an effective and simple error compensation function, whose inputs are the outputs of booth encoders, accordingly generates the approximate carry value. Finally, to reduce the area of fixed width modified booth multiplier, a simple & fast compensation circuit is derived, ensuring that truncation error is minimal.

3.1.Modified partial product matrix to improve the accuracy

Accuracy can be improved by decreasing the partial product bits in LSP_{minor} which modify the partial product matrix of PTM. To obtain the new matrix first the LSB $p_{n/2-1,0}$ of $PP_{n/2-1}$ and $c_{n/2-1}$are added in advance which produce $\varepsilon_{n/2-1}$ as sum an λ as carry at $(n-2)^{th}$ and $(n-1)^{th}$ positions respectively. As the weight of extra '1' in PTM and λ are same, both can be added which outputs $\bar{\lambda}$ as sum &λ as carry propagated to the n^{th} bit position. Then this carry is incorporated with sign extension bits of PP_0 which modify $\bar{s_0}s_0s_0$to new partial product bits $\omega_2\omega_1\omega_0$.

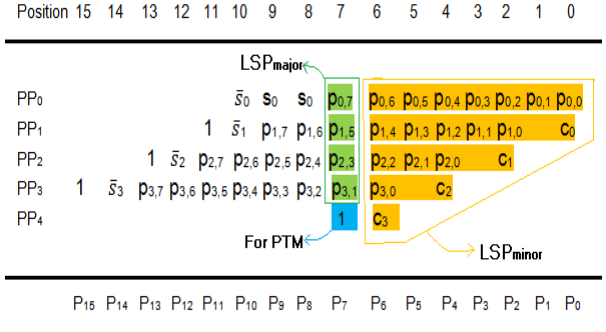

Fig 3.Partial product matrix of PTM.

The truth tables to generate $\varepsilon_{n/2-1}$,λ and $\omega_2\omega_1\omega_0$ are shown in table 2, 3 respectively and corresponding logic equations are given below.

$$\varepsilon_{n/2-1,0} = a_0 . o_{n/2-1,0} \quad (6)$$

$$\lambda = \overline{\varepsilon_{n/2-1,0}} . \overline{z_{n/2-1,0}} . b_{n-1} \quad (7)$$

$$\omega_2 = \overline{(s_0 . \bar{\lambda})} \quad (8)$$

$$\omega_1 = \overline{\omega_2} \quad (9)$$

$$\omega_0 = \overline{(s_0 + \bar{\lambda}) . \omega_2} \quad (10)$$

Table 2.Truth table for $\varepsilon_{n/1-1}$,λ

$t_{n/2-1,0}$	$o_{n/2-1,0}$	$z_{n/2-1,0}$	$c_{n/2-1,0}$	$p_{n/2-1,0}$	$\varepsilon_{n/2-1,0}$	λ
0	0	1	0	0	0	0
0	1	0	0	a_0	a_0	0
0	1	0	1	$\bar{a}_0$	a_0	$\bar{a}_0$
1	0	0	0	0	0	0
1	0	0	1	1	0	1

Table 3.Truth table for ω2ω1ω0

S_0	λ	ω_0	C_{out0}	S_0	ω_1	C_{out1}	$\bar{S}_0$	ω_2
1	0	1	0	1	1	0	0	0
1	1	0	1	1	0	1	0	1
0	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1

According to logic equations (6 -10) the circuit to generate $\omega_2\omega_1\omega_0$&$\bar{\lambda}$ is shown in fig.4.

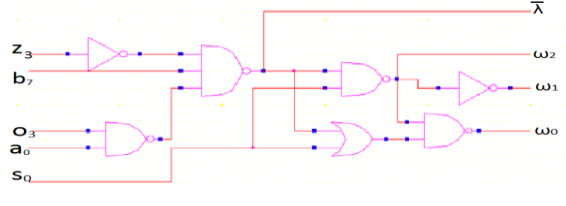

Fig 4.Error compensation function generation circuit.

The modified partial product matrix of 8×8 Booth multiplier is shown in fig.5.

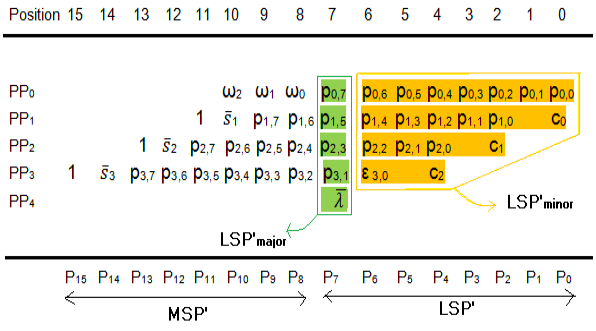

Fig 5.Proposed modified partial product matrix

3.2.Proposed low error Area efficient errorcompensation circuit

In the modified partial product matrix the new LSP, MSP, LSP_{minor}&LSP_{major} are denoted asLSP`, MSP`,$LSP`_{minor}$&$LSP`_{major}$ respectively as shown in fig. 5. The partial product bits in $LSP`_{minor}$ are omitted in the fixed with multipliers results accuracy loss ,which can be compensated by a simple modified sorting network so that hardware required to generate partial products in $LSP`_{minor}$is removed instead of which a simple compensation circuit composed of basic gates is used. Hence, developing a simple error compensation function whose output value approximates the most accurate error compensation value $\hat{e}$ as follows:

Now the error compensation function $\hat{e}$ can be expressed as:

$$\hat{e}=[0.5(\ominus_{major}+\bar{\lambda})+CP(LSP`_{minor})] \quad (11)$$

Where $CP(LSP'_{minor})$ represents approximate carry value propagated from $LSP`_{minor}$ to $LSP`_{major}$. To explore the correlation between $CP(LSP'_{minor})$ and the outputs of the booth encoder, consider the encoder output z_j. If it is one, the partial product bit of PP_j in $LSP`_{minor}$ must be equal to zero. The different combination of z_j for $0 \le j \le n/2-1$, can be represented by a generalized index ϕ is defined as (12).

$$\Phi = \bar{z}_{n/2-1} \times 2^{n/2-1} + \bar{z}_{n/2-2} \times 2^{n/2-2} + \ldots + \bar{z}_0 \times 2^0 \quad (12)$$

According to (12), the range of ϕ is from 0 to $2^{n/2}-1$. For a specific ϕ, all combinations that produce the

same ϕ can be utilized to calculate the average value of S(LSP`$_{minor}$)denoted as $S(\phi)$. The normalized value of $S(\phi)$ is $S(\phi)/2^{n-1}$ can be substituted for CP(LSP`$_{minor}$). Thus the compensation error function can be written as

$$\hat{e}=[0.5(\theta_{major}+\bar{\lambda})+S(\phi)/2^{n-1}] \quad (13)$$

The correlation between ϕ and $S(\phi)/2^{n-1}$ for 8 bit fixed width modified booth multiplier is shown in Table 4.

Table 4.The correlation between ϕ and $S(\phi)/n-1$ with $n=8$

Φ	S(ϕ)/2^{n-1}	I(ϕ)	Φ	S(ϕ)/2^{n-1}	I(ϕ)
0	0	0	8	0.1665	0
1	0.5025	0	9	0.66930	
2	0.5105	0	10	0.67710	
3	0.99890		111.1655	1	
4	0.54150		12	0.70810	
50.98850			13	1.15531	
60.9963	0		14	1.16311	
71.50351			15	1.67031	

In (13)θ_{major},$\bar{\lambda}$ are always integers. Moreover $\hat{e}$ is also an integer. Then the contribution of $S(\phi)/2^{n-1}$ to $\hat{e}$ can be approximated to an integer $I(\phi)$ as follows:

$$I(\phi) = S(\phi)/2^{n-1} \quad (14)$$

$I(\phi)$ can be related to summation of$\bar{z}_j$for $0 \le j \le n/2-1$

Let$\hat{k} = \sum_{j=0}^{\frac{n}{2}-1} \bar{z}_j$(15)

The relation between $\hat{k}$&$I(\phi)$ with $n=8$ is shown in Table 5.From Table.5 the relation between $I(\phi)$&$\hat{k}$ can be expressed as:

$$I(\phi) = (\hat{k}-1)/2 \quad (16)$$

The final error compensation function can be obtained by, substituting (14) into (13), then (16) into that equation.

$$\hat{e}=[0.5(\theta_{major}+\bar{\lambda})+(\hat{k}-1)/2] \quad (17)$$

According to (17), by using compression tree structure $(\theta_{major}+\bar{\lambda})$and ($\hat{k}$-1)/2can be compressed with the partial product bits in MSP to generate the final fixed width product.θ_{major}, $\bar{\lambda}$ can be generatedfrom the partial product bits in LSP`$_{major}$ and MSP.

Table 5.The correlation between$\hat{k}$ and I(ϕ) with n=8

$\hat{k}$	I(ϕ)	$\hat{k}$	I(ϕ)	$\hat{k}$	I(ϕ)	$\hat{k}$	I(ϕ)
0	0	4	1	8	1	12	2
1	1	5	2	9	2	13	3
2	1	6	2	10	2	14	3
3	2	7	3	11	3	15	4

For quick production of $(\hat{k}-1)/2$ a simple and efficient circuit denoted as SC-generator is to be designed.By (15),the inputs of SC-generator are $\bar{z}_j$ for $0 \le j \le n/2-1$ which generates 'm'outputs as $\alpha_1, \alpha_2, \ldots \alpha_m$ where $m=(n/2-1)/2$, which is depicted in Fig.6.

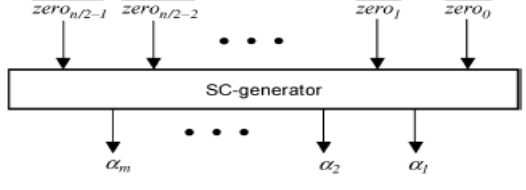

Fig.6.SC-generator block diagram.

Summation of outputs of SC-generator is equal to $I(\phi)$. The proposed SC-generator is composed of a modified sorting network instead of adder cells based on following observation.Sorting $\bar{z}_j$results in largest bits gathering at least significant positions, then $\alpha_k = \beta_{2k}$ for $1 \le k \le m$, where β_k is the output of sorting network for $0 \le k \le n/2-1$.To design the SC-generator, consider the different sorting networks such as bitonic and odd-even merge sorting networks, which suits for hardware implementation. Out of these, odd-even merge sorting network requires less hardware thus a simplified odd-even merge sorting network is adopted in implementing SC-generator. The odd-even merge sorting networks for n=8 & 16are shown in Fig.7.

The odd even merge sorting network is modified to derive the desired SC-generator which is composed of basic gates. The proposed SC-generator is shown in fig.8.

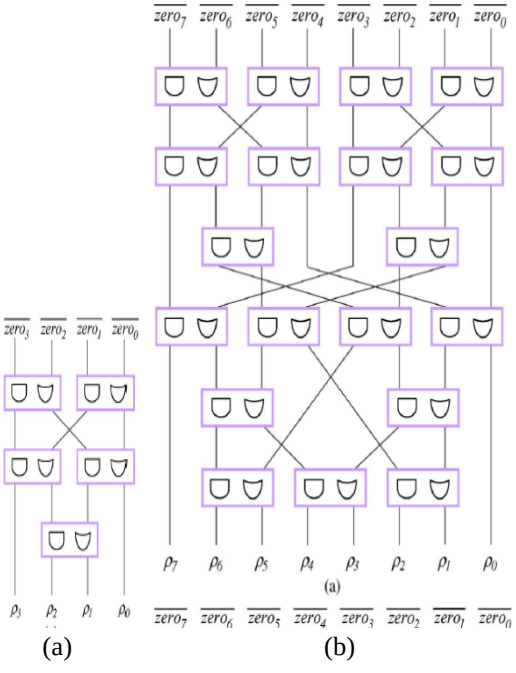

Fig.7.Odd even merge sorting networks for n=8 &n=16 respectively.

The SC-generator for different n can be constructed in the same manner. The outputs of SC-generator are fed into LSP`$_{major}$ to produce final fixed width product.

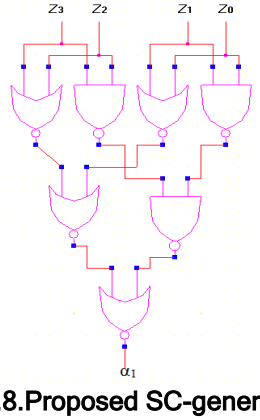

Fig.8.Proposed SC-generator.

The partial product matrix of the proposed fixed width booth multiplier for *n=8* is depicted in Fig.9.In the final partial product matrix, the partial product bits in LSP$_{minor}$ and carries thus generated are substituted by SC-generator.

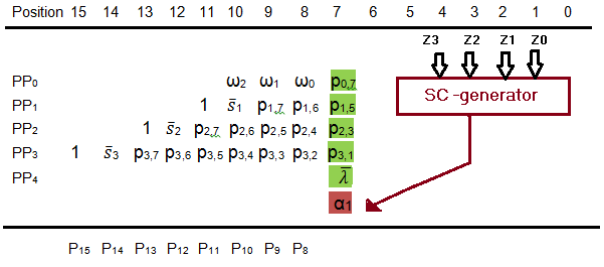

Fig.9.Final proposed partial product matrix

3.3.Final circuit &implementation

This paper mainly deals with drawing the layout of the proposed fixed width modified Booth multiplier for *n*=8. Fig.10 shows the block diagram of the final circuit.

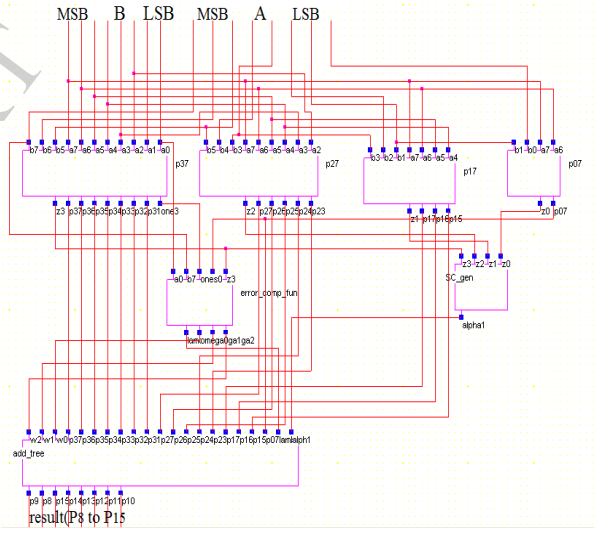

Fig10.Block diagram of the proposed multiplier.

The above block diagram is implemented using Digital Schematic tools, and layout is implemented in Bottom-up approach using IC station tools.The proposed multiplier is suffering from little accuracy loss compared to full widthmultiplier which is negligible formost of the lossyapplications such image processing etc.

The partial product generator is slightly modified to reduce the Mos transistors. In the partial product generator the multiplexer which require 14 Mos

transistors,is replaced with exclusive or gate which require 6 Mos transistors. The implementation of EXOR gate with transmission gate is shown in fig.11.

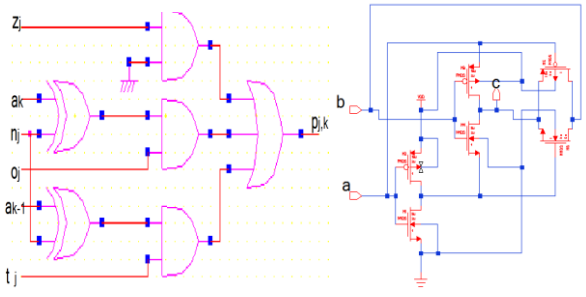

Fig.11.Exor gate implementation using Transmission gates

The internal circuit of p07 is shown in fig.12.Similarly P17, P27, P3can be designed.

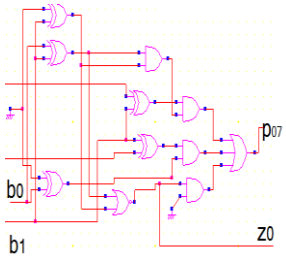

Fig.12 Internal circuit of P07

Parallel multipliers are often implemented as either array multipliers or as multiplier trees. Hardware saving and less delay can be achieved by Dadda tree ([15],[16]) implementation. Here the adder is implemented using 11 MOS transistors which is shown in fig. 13 and its layout is shown in fig. 14.

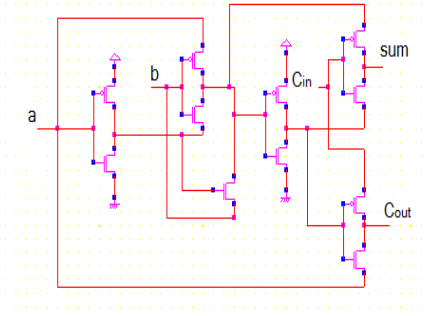

Fig.13 GDI based 11 transistor full adder

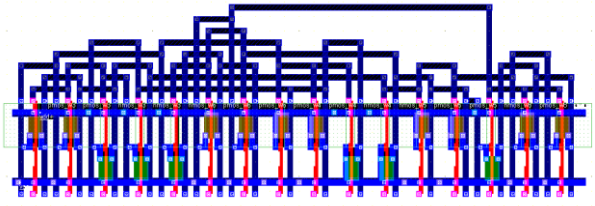

Fig.14 Layout of GDI based 11 transistor full adder

4. EXPERIMENTAL RESULTS

The comparison of hardware required for proposed multiplier with PTM is given in Table 8.

Table 8.Thecomparison of components requirement forPTM and proposed multiplier

Component	PTM	Proposed multiplier
NAND gate(3i/p)		1
NAND gate(2i/p)	4	6
NOR gate(2i/p)	4	8
NOT gate	3	5
OR gate(3i/p)	32	16
AND gate(2i/p)	76	41
OR gate(2i/p)		1
EXOR gate(2i/p)	12	32
Half Adder	14	3
Full Adder	31	17
Multiplexer	36	

The multiplier descriptions are mapped on a 0.5 µmCMOS standardcell library using synthesis tool from Mentor Graphics.From the experimental analysis,area is significantly reducedcompared to PTM. Results of the simulation clearly show that the proposed multiplier architecture requires less hardware and performs approximately same as existing PTM.

5. Conclusion

Fixed width Modified Booth multiplier is designed and is verified funcionally. In this new approach lower half of the partial product matrix is omitted. This leads to

truncation error but the chip area reduces considerably. Here 18% reduction in area is observed for 8×8 multiplication.In high speed applications and where accuracy is not much important this type of structure is well used. Because the lower half of the partial products are omitted hardware required to construct those partial products also not constructed. This leads to less area, Low power consumption and High speed.

6. References

[1]W.C. Yeh and C.W. Jen, "High-speed Booth encoded parallel multiplier design," *IEEE Trans. Computers*, vol. 49, no. 7, pp. 692–701, July 2000.

[2] G. O. Young, A. Inoue, R. Ohe, S. Kashiwakura, S. Mitarai, T. Tsuru,and T. Izawa, "A 4.1-ns compact 54 x 54 multiplier utilizing signselectBooth encoders," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1676–1682, Nov. 1997.

[3] K. Choi and M. Song, "Design of a high performance 32 x32-bit multiplier with a novel sign select Booth encoder," in *Proc. IEEE Int.Symp. Circuits and Systems*, 2001, vol. 2, pp. 701–704.

[4]S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. CircuitsSyst. II, Exp. Briefs*, vol. 43, no. 2, pp. 90–94, Feb. 1996.

[5] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in *Proc. VLSI Signal Processing, VI*, New York, 1993, pp. 388–396.

[6] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixedwidthmultipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.

[7] J.-S.Wang, C.-N. Kuo, and T.-H. Yang, "Low-power fixed width array multipliers," in *Proc. Int. Symp. Low Power Electron.Des.*, 2004, pp. 307–312.

[8] L. D. Van, S. S.Wang, and W. S. Feng, "Design of the low error fixed width multiplier and its application," *IEEE Trans. Circuits Syst. II, Exp.Briefs*, vol. 47, no. 10, pp. 1112–1118, Oct. 2000.

[9] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixedwidthmultipliers for DSP applications," *IEEE Trans. Circuits Syst.I,Exp. Briefs*, vol. 46, no. 6, pp. 836–842, June 1999.

[10] Y.-C. Liao, H.-C.Chang, and C.-W.Liu, "Carry estimation for two's complement fixed-width multipliers," in *Proc. IEEE Workshop SignalProcessing Systems*, 2006, pp. 345–350.

[11] S. J. Jou, M.-H.Tsai, and Y.-L.Tsao, "Low-error reduced-width Booth multipliers for DSP applications," *IEEE Trans. Circuits Syst.I, Fudam.Theory Appl.*, vol. 50, no. 11, pp. 1470–1474, Nov. 2003.

[12] K.-J. Cho, K.-C.Lee, J.-G.Chung, and K. K. Parhi, "Design of lowerrorfixed-width modified Booth multiplier," *IEEE Trans. Very LargeScaleIntegr. (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.

[13] M.-A. Song, L.-D.Van, and S.-Y.Kuo, "Adaptive low-error fixedwidthBooth multipliers," *IEICE Trans. Fundamentals*, vol. E90-A, no. 6, pp. 1180–1187, Jun. 2007.

[14] E. de Angel and E. E. Swartzlander, Jr., "Low power parallel multipliers," in *Workshop VLSI Signal Process., IX*, 1996, pp. 199–208.

[15] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*,vol. 34, pp. 349–359, 1965.

[16] K. C. Bickerstaff, E. E. Swartzlander, Jr., and M. J. Schulte, "Analysis of column compression multipliers," in *Proc. 15th IEEE Symp. ComputerArithmetic*, 2001, pp. 33–39.

E.L.K.Priya:persuingM.Tech (VLSI System Design) in Swarnandhra college of Engineering & Technology, India. Has four years of teaching experience. Major working areas are Digital electronics, Embedded Systems and VLSI.

P.Lakshmi Sarojini received M.Tech degree in DSCE from JNTU,Hyderabad. She worked in DRDO as a Scientist and presently working asProfessor & Head, Dept of ECE, Swarnandhra College of Engineeringand Technology Narsapur. Her area of research includes IC design,Mixed signal VLSI and programming digital systems.

G. Rajesh Kumar: Asst. Professor in Vishnu institute of technology, India. Has five years of teaching experience. Major working areas are Digital electronics, Embedded Systems and VLSI. Presented research papers in five international conferences and three national conferences.Published three research papers in International Journals.Research Scholar in JNTUK.