International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 2, Issue 12 (December 2013), PP.82-88

Customized Booth Multiplier for MM Applications

K.VANI[1], G.VENKATA RAMANA[2]
[1](*ECE, CMR Engineering College/JNTUH, INDIA)*
[2]*(ECE, CMR Engineering College/JNTUH, INDIA)*

Abstract:- This paper proposes the design of a Customized area efficient fixed- width (CAEFW) modified booth multiplier for Multimedia (MM) applications are commonly known as lossy applications. By substituting software code in place of hardware, Area efficiency can be obtained in the generation of the lower half part of the partial product matrix of Booth multiplier which leads to truncation error. Accuracy can be enhanced by the CAEFW and subsequently deriving an effective error compensation function which is composed of basic gates. Further accuracy can be achieved by compensation circuits of basic gates. This CAEFW requires very less hardware. 11 transistors are used in ADDER as well as 6 transistors are used in Exclusive-OR implementation to reduce the area.

Keywords:- CAEFW, Error Composition circuit, fixed-width multiplier (FWM), Partial product matrix, Error compensation function, Post Truncated Multiplier (PTM). Lossy systems

I. INTRODUCTION

The today's world requires fast response systems with well performance. The key component of high performance systems such as FIR filters, microprocessors, digital signal processors, etc are depends on their multipliers. In real time applications, we require computations of signed numbers. An elegant approach to multiplying signed numbers is the booth multiplier 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 nxn 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.

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 to effectively 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. those cases, truncation errors results in large output error, which can be countered by performing additional compensation, which is again 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 error compensation 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 analyzed 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 the proposed 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 PTM which 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 .

II. CUSTOMIZED AREA EFFICIENT FIXED- WIDTH (CAEFW)

CAEFW is divided into two parts, they are MSB and LSB. Again LSB is subdivided into two parts, namely LSB_{minor} and LSB_{major}.The nxn multiplier with n bits as output is used in truncated multiplier. the partial products divided into two subsets LSB includes the n less significant columns of the partial product matrix, while the MSB includes the remaining columns the full-width multiplier output, P is given by(4)where SUM(MSB) and SUM(LSB) represent the weighted sum of the elements of MSB and LSB respectively.

P =SUM (MSB) + SUM (LSB) (4)

LSB bits are negligible in direct truncated multiplier in partial product assuming that output is negligible for the contribution of n most significant bits. Hardware performance is advantage for this CAEFW.This solution is very advantageous in terms of hardware performances. The cells for the LSB 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 LSB causes a very big error bounded by n (2 − 1) LSB.

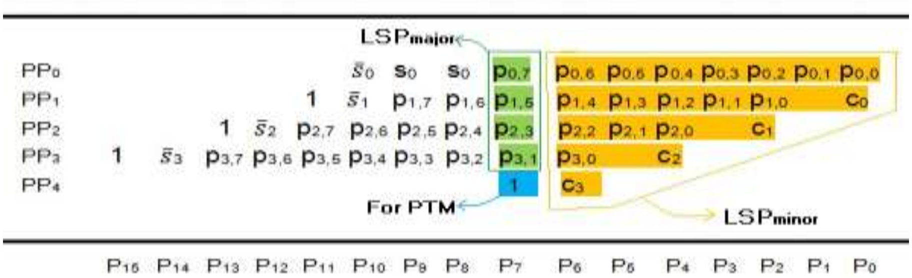
In the post-truncated modified Booth multiplier (PTM) the with the addition of an extra „1‟(carry value by LSPminor) at the (n-1)th bit position of partial product matrix as depicted in Fig 3 (including‟1‟)then outputs the highly significant n product bits. With this the most accurate error compensation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

$P=S(MSB) +S(LSB_{major}) +S (LSB_{minor}) +1$

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 LSB_{minor} to LSB_{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. In CAEFW, simple & fast compensation circuit is derived, truncation error is minimal.

III. MODIFIED PARTIAL PRODUCT MATRIX TO IMPROVE THE ACCURACY

Decreasing the bits in LSBminor Accuracy can be improved and PTM matrix can be modified. To obtain the new matrix first the LSB *p*n/2-1,0 of *PPn/2-1* and *c*n/2-1are added in advance which produce εn/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 PP0 which modify $\bar{s}_0 s_0 s_0$to new partial product bits ω2ω1ω0.


Position 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
LSPmajor
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 1 c_3
For PTM
LSPminor
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

The truth tables to generate *εn/2-1*,λ and ω2ω1ω0 are shown in table 1, 2 respectively and corresponding logic equations are given below.

$$\varepsilon_{n/2-1,0} = a_0 \cdot o_{n/2-1,0} \qquad (1\ )$$

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

$$\omega_2 = \overline{(s_0 \,.\bar{\lambda}\,)} \qquad (3)$$

$$\omega_1 = \overline{\omega_2} \qquad (4)$$

$$\omega_0 = \overline{(s_0 + \bar{\lambda}\,).\,\omega_2} \qquad (5)$$

t n/2-1,0	o n/2-1,0	z n/2-1,0	cn/2-1,0	pn/2-1,0	ɛ n/2-1,0	λ
0	0	1	0	0	0	0
0	1	0	0	a0	a0	0
0	1	0	1	ā0	a0	ā0
1	0	0	0	0	0	0
1	0	0	1	1	0	1

S0	λ	ω0	Cout0	S0	ω1	Cout1	0	ω2
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

Table 2. Truth table for ω2ω1ω0

According to logic equations (1-4) the circuit to generate ω2ω1ω0& $\bar{S}$is shown in fig.2.

3.1. Area efficient reduced error compensation circuit

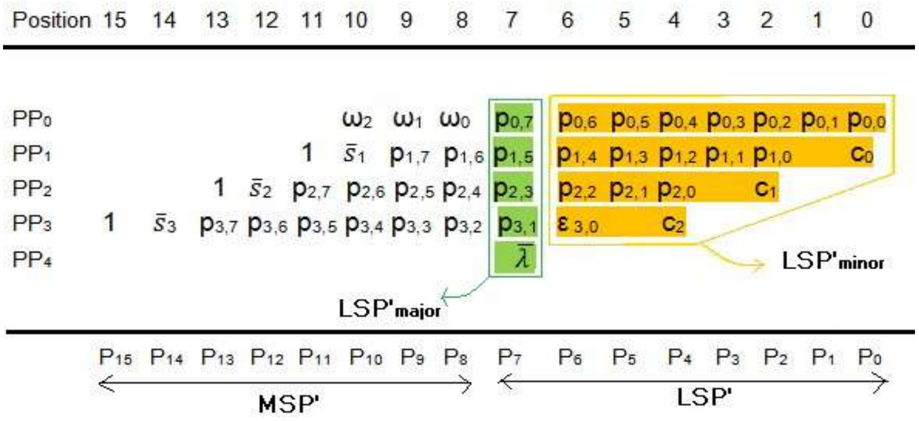
In the CAEFW new LSB, MSB, LSB minor & LSB major are denoted as LSB`, MSB`, LSB` minor & LSB `major respectively as shown in fig.2. The partial product bits in LSP `minor are omitted in the fixed with multipliers results accuracy loss ,which can be compensated by a simple modified sorting network so that hardware required to generate partial products in LSB `minoris 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(ɵ*major*+$\bar{\lambda}$)+CP(LSB`minor)] (5)

Where CP(LSB‟ minor) represents approximate carry value propagated from LSB` minor to LSB `major. To explore the correlation between CP(LSB ‟minor) and the outputs of the booth encoder, consider the encoder output *zj*. If it is one, the partial product bit of PPI in LSB `minor must be equal to zero. The different combination of *zj* for *0 ≤ j ≤ n/2-1*, can be represented by a generalized index ϕ is defined as

Φ= $\bar{z}$*n/2-1*× *2n/2-1 +* $\bar{z}$*n/2-2*× *2 n/2-2 +.0*×*20* *(6)*

The modified partial product matrix of 8 8 Booth Multiplier is shown in fig.


Position 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PP0
PP1
PP2
PP3
PP4
ω2 ω1 ω0 p0,7 p0,6 p0,5 p0,4 p0,3 p0,2 p0,1 p0,0
1 s̄1 p1,7 p1,6 p1,5 p1,4 p1,3 p1,2 p1,1 p1,0 C0
1 s̄2 p2,7 p2,6 p2,5 p2,4 p2,3 p2,2 p2,1 p2,0 C1
1 s̄3 p3,7 p3,6 p3,5 p3,4 p3,3 p3,2 p3,1 ε3,0 C2
λ̄
LSP'major
LSP'minor
P15 P14 P13 P12 P11 P10 P9 P8 P7 P6 P5 P4 P3 P2 P1 P0
MSP'
LSP'

According to (6), the range of ϕ is from 0 to *2n/2-1*.For a specific ϕ, all combinations that produce. Same ϕ can be utilized to calculate the average value of S(LSB`minor)denoted as S(ϕ). The normalized value of *S*(ϕ*)* is *S*(ϕ *)/2n-1* can be substituted for CP(LSB`minor).Thus the compensation error function can be written as

$\hat{e}=[0.5(\theta major+\bar{\lambda})+S(\phi)/2n-1]$ (7)

The correlation between ϕ and *S*(ϕ*)/2n-1* for 8 bit Fixed width modified booth multiplier.

Table 3. The correlation between ϕ and S(ϕ)/n-1 with n=8.

Φ	$S(\Phi)/2^{N-1}$	I(ϕ)	Φ	$S(\Phi)/2^{N-1}$	I(ϕ)
0	0	0	8	0.1666	0
1	0.5030	0	9	0.6692	0
2	0.4995	0	10	0.6667	0
3	0.9980	0	11	1.1675	1
4	0.5415	0	12	0.7081	0
5	0.9984	0	13	1.4552	1
6	0.9964	0	14	1.1642	1
7	1.5023	1	15	1.6801	1

In (7) $\theta major, \bar{\lambda}$ 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(ϕ) as follows:

$I(\phi) = S(\phi)/2_{n-1}$ ***(8)***

I(ϕ) can be related to summation of $\bar{z}$jfor 0 ≤ j ≤ n/2-***1***

Let $k = \sum_{j=0}^{\frac{n}{2}-1} \bar{z}_j$ (9)

Table 2.From Table.5 the relation between I(ϕ)*&*ƙ *can be expressed as:*

$I(\phi) = (k-1)/2$ (10)

The final error compensation function can be obtained by, substituting (8) into (7), then (10) into that equation.

$\hat{e}=[0.5(\theta major+\bar{\lambda})+(k-1)/2]$ (11)

According to (11), compression tree structure $(\theta_{major}+\bar{\lambda})$; and (ƙ-1)/2can be compressed with the partial product bits in MSB to generate the final fixed width product. ɵ*major*, $\bar{\lambda}$ can be generated from the partial product bits in LSB`major and MSB.

K	I (ϕ)	K	I (ϕ)	K	I (ϕ)	K	I (ϕ)
0	0	3	1	8	1	12	2
1	1	5	2	9	2	13	2
2	1	6	2	10	2	14	3
3	2	7	3	11	3	15	4

Table 4. The correlation between ƙ and **I(**ϕ) with n=8

For quick production of *(*ƙ*-1)/2* a simple and efficient circuit denoted as SC-generator is to be designed.

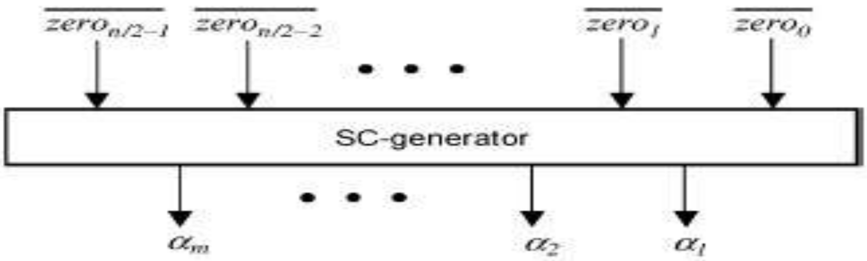


Fig: 3 SC –Generator Block diagram

Partial product bits in MSB to generate the final fixed Summation of outputs of SC-generator is equal to *I(ϕ)*. 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 & 16. The odd even merge sorting network is modified to derive the desired SC-generator which is composed of basic gates.

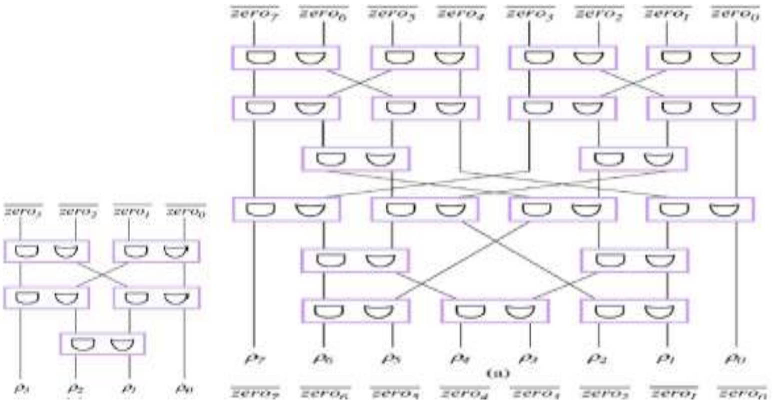


Fig.3. Odd even merge sorting networks for n=8 &n=16 respectively.

Different values of n can be constructed in the same manner in SC-generator.

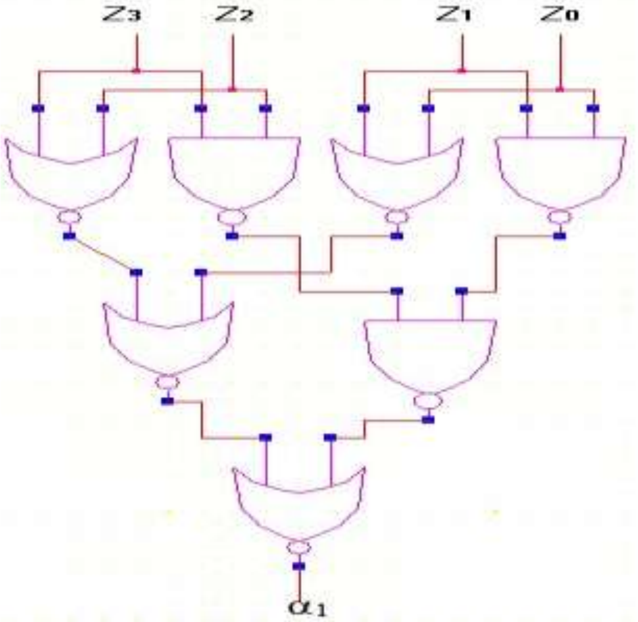


Fig.4.SC-generator CAEFW method

The partial product matrix of the CAEFW for *n=8* is shown in Fig.5.In the final partial product matrix, the partial product bits in LSBminor and carries thus generated are substituted by SC-generator.

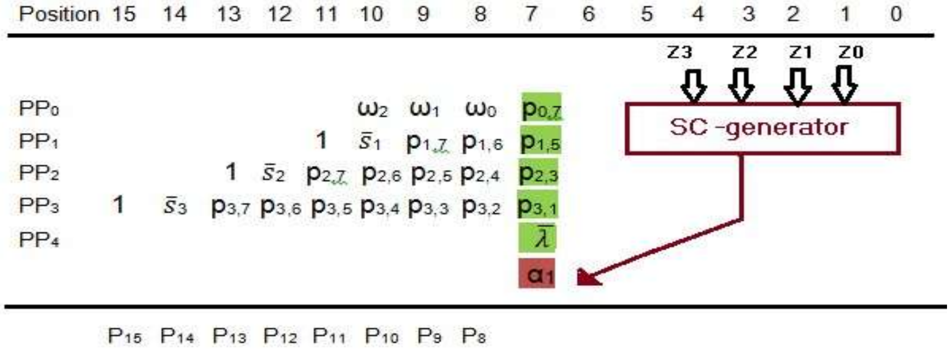

Position 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Z3 Z2 Z1 Z0
SC -generator
PP0 ω_2 ω_1 ω_0 $p_{0,7}$
PP1 1 $\bar{s}_1$ $p_{1,7}$ $p_{1,6}$ $p_{1,5}$
PP2 1 $\bar{s}_2$ $p_{2,7}$ $p_{2,6}$ $p_{2,5}$ $p_{2,4}$ $p_{2,3}$
PP3 1 $\bar{s}_3$ $p_{3,7}$ $p_{3,6}$ $p_{3,5}$ $p_{3,4}$ $p_{3,3}$ $p_{3,2}$ $p_{3,1}$
PP4 $\bar{\lambda}$
α_1
P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8

Fig.5.Final proposed partial product matrix.

3.2. CAEFW & implementation

Implementation of CAEFW mainly deals with the layout with n=8.

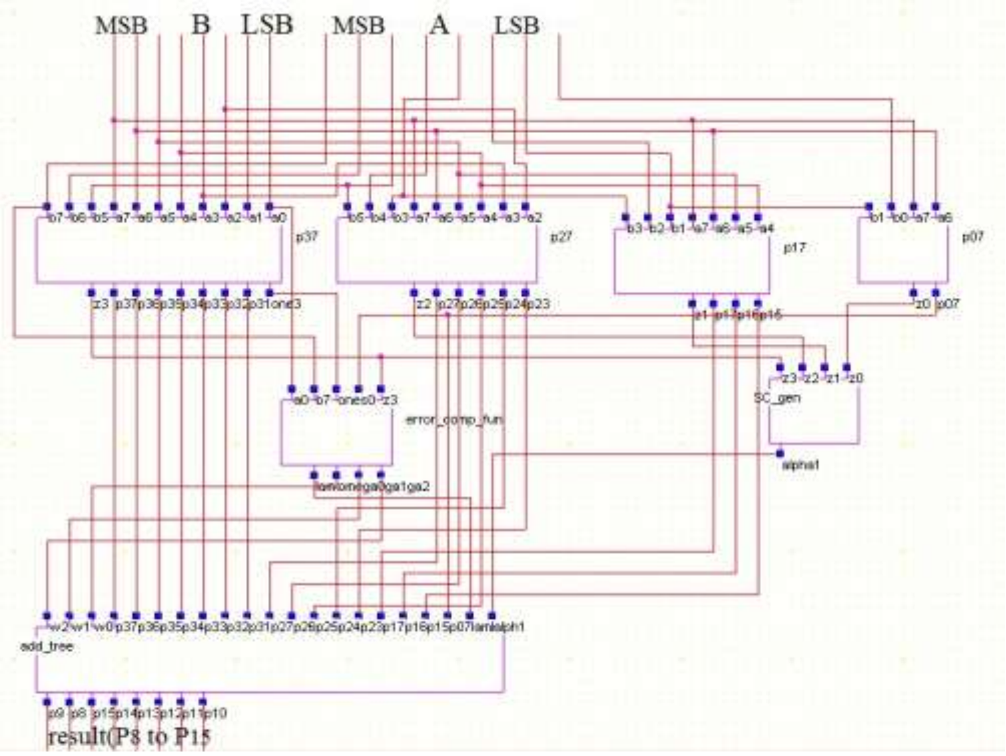


Fig 6.Block diagram of the CAEFW

Digital Schematic tools are mainly used for implementation of CAEFW and layout is implemented in Bottom- up approach using IC station tools. CAEFW is suffering from little accuracy loss compared to full width multiplier which is negligible for most of the lossy applications such image processing etc

IV. Results and Discussions

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(2i/p)		1
NAND	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.

V. Conclusions

CAEFW designed and is verified functionally. 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 8x8 multiplications. 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.

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.

[14]. L. Dadda, “Some schemes for parallel multipliers,” AltaFrequenza,vol. 34, pp. 349–359, 1965.

[15]. 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