International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 1; Issue: 7; November–2013

HIGH SPEED FIXED-WIDTH MODIFIED BOOTH MULTIPLIERS

Jeena James[1],
PG student,
Saintgits College of Engineering,
MG University, Kottayam,India.
jeenajms68@gmail.com

Prof.Binu K Mathew[2],
Associate Professor,
Saintgits College of Engineering,
MG University, Kottayam,India.
binu.k@saintgits.org

***Abstract*- Multiplication is a fundamental arithmetic operation used in multimedia and DSP applications like convolution,filtering and compression. Since multipliers have a significant impact on the performance of the entire system, many high performance algorithms and architecture have been proposed to accelerate multiplication. Fixed-width booth multipliers provide high performance by reducing the number of partial products and reduces hardware complexity by removing adder cells from the least significant parts. There by huge truncation errors occurs. Thus area and speed are two conflicting constraints. So improving speed results always in larger areas. So here we try to find out the best trade off solution among the both of them.**

In this project,a high speed fixed-width modified booth multiplier is proposed. Here we have first tried to design multipliers with different adders at the final stage and compare their speed and complexity of circuit i.e. the area occupied. While comparing the adders for designing a high speed fixed-width booth multiplier we found out that Carry Select Adders posses high speed which can be used as the final stage. Here we uses an adaptive compensation method which saves the establishment time of compensation circuit, provides varying column information and achieves high accuracy performance. The results are compared with conventional booth multipliers with different bits. Also a simple compensation circuit composed of simple logic gates is developed according to the proposed error compensation function.

Index Terms- Adaptive conditional-probability estimator (ACPE), Booth multiplier, Discrete cosine transform (DCT), fixed-width.

I.INTRODUCTION

Fixed-width booth multipliers are usually used in many Digital Signal Processing (DSP) applications,Moving Picture Expert Group (MPEG) coding and multimedia application for operation elements in internal multiplications, which has demonstrated the importance of fixed-width multiplier. However, unfortunately, the fixed-width multiplier must truncate half of the output width of the multiplier, which certainly will produce truncation errors. In fixed-width multiplier, the multiplication of an L-bit multiplier by an L-bit multiplicand will produce an L-bit result. To achieve high performance, the modified Booth encoding which reduces the number of partial products through performing the multiplier recoding has been widely adopted in multipliers. The $n \times n$ fixed-width multipliers that generate only the n most significant product bits are frequently utilized to maintain a fixed word size such that significant hardware complexity reduction and power saving can be achieved. This is made possible by directly removing the adder cells of the n least significant bits of standard multiplier of $2n$-bit output product. However, a huge truncation error will be introduced to this kind of direct-truncated fixed-width multiplier (DTFM).

To effectively reduce the truncation error, various error compensation methods, which add estimated compensation value to the carry inputs of the reserved adder cells, have been proposed. The error compensation value can be produced by the constant scheme or the adaptive scheme. The constant scheme pre-computes the constant error compensation value and then feeds them to the carry inputs of the retained adder cells when performing multiplication operations regardless of the influence of the current input data value. With the advantage of simplification, the truncation error of the constant scheme is relatively large. On the contrary, the adaptive scheme was developed to achieve higher accuracy than the constant scheme through adaptively adjusting the compensation value according to the input data at the expense of a little higher hardware complexity. However, most of the adaptive error compensation approaches are developed only for fixed-width array multipliers and cannot be applied to significantly reduce the truncation error of fixed-width modified Booth multipliers directly. To overcome this problem, several error compensation approaches have been proposed to effectively reduce the truncation error of fixed-width modified Booth multipliers. In certain approaches, the compensation value was generated by using statistical analysis and exhaustive simulation analysis. Recently, many works use more information from Booth encoder and partial products to achieve higher accuracy performance. The area cost is increased due to extra information of compensation circuits, i.e., there is a trade-off between accuracy and area cost.

This project aims at high-speed fixed-width modified Booth multiplier. In this project adaptive conditional probability estimator (ACPE) is derived from the conditional-probability theory and a simple compensation circuit composed of simple logic gates is developed according to the proposed error-compensation

function. Simulation and implementation results show that the proposed fixed-width modified Booth multiplier actually achieves much higher performance than existing fixed-width modified Booth multipliers.

III. FIXED -WIDTH MODIFIED BOOTH MULTIPLIER

Modified Booth encoding is popular to reduce the number of partial products .Two L-bit inputs X and Y and a $2L$-bit standard product SP (without truncationerror) can be expressed in two's complement representation as follows:

$$X = -x_{L-1} \cdot 2^{L-1} + \sum_{i=0}^{L-2} xi.\ 2^{i}$$

$$Y = -y_{L-1} \cdot 2^{L-1} + \sum_{i=0}^{L-2} yi.\ 2^{i}$$

$$SP = X \times Y. \qquad (1)$$

The modified Booth encoder maps three concatenated inputs y_{2i+1}, y_{2i}, and y_{2i-1} into y'_i. After encoding, there are $Q = L/2$ rows in the partial product array with an even width L. The corresponding partial products represented in input x_i are tabulated in Table I, where the last column n_i stands for the sign of each partial product.

TABLE I: partial product array for 10 bit booth encoder

y_i'	$P_{10,i}$	$P_{9,i}$	$P_{8,i}$	$P_{7,i}$	$P_{6,i}$	$P_{5,i}$	$P_{4,i}$	$P_{3,i}$	$P_{2,i}$	$P_{1,i}$	$P_{0,i}$	n_i
0	0	0	0	0	0	0	0	0	0	0	0	0
1	X_9	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	0
-1	$\overline{X_9}$	$\overline{X_9}$	$\overline{X_8}$	$\overline{X_7}$	$\overline{X_6}$	$\overline{X_5}$	$\overline{X_4}$	$\overline{X_3}$	$\overline{X_2}$	$\overline{X_1}$	$\overline{X_0}$	1
2	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	0	0
-2	$\overline{X_9}$	$\overline{X_8}$	$\overline{X_7}$	$\overline{X_6}$	$\overline{X_5}$	$\overline{X_4}$	$\overline{X_3}$	$\overline{X_2}$	$\overline{X_1}$	$\overline{X_0}$	1	1

The partial product array can be divided into two parts: the main part (MP),which includes ten most significant columns (MSCs), and the truncation part (TP), which includes ten LS columns (LSCs) in the case of 10 × 10 multiplier. The TP can also called LP (lower part) .The SP can be rewritten as follows:

$$SP = MP + TP.$$

In the fixed-width multiplication, TP can be estimated and the quantized product QP can be defined as

$$QP = MP + \sigma \cdot 2^{L}$$

where σ representing the estimation bias (EB) from TP can be further decomposed into TP_{Major} (MSC of TP) and TP_{minor} (LSCs of TP) parts as

$$\sigma = \text{Round}(TP_{Major} + TP_{minor}) \qquad (2)$$

where Round(k) is rounding k to the nearest integer. Because TP_{Major} affects more than TPminor while contributing toward the EB σ, the σ value can be obtained by calculating TPMajor and estimating TPminor in order to reduce truncation errors.

IV. PROPOSED ARCHITECTURE

Fixed width multipliers are used in many applications related with signal processing systems and multimedia. The n × n multipliers that generate only the most significant n terms are widely used in applications related with lossy systems. In fixed width modified Booth multipliers ,the adder cells need for the computation of the n least significant output bits of the multiplier are removed hence making significant hardware reduction and power saving. But the problem of truncation error introduced into the output need to be compensated.

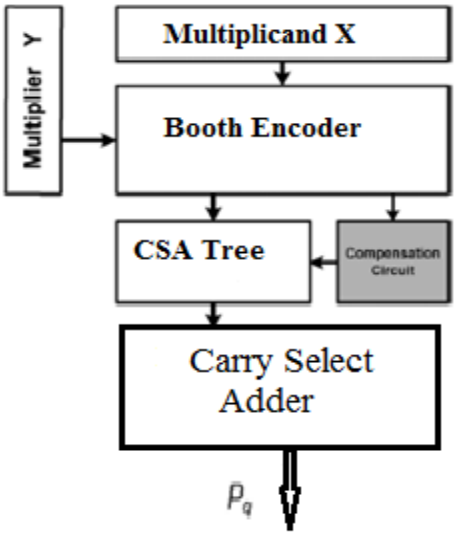


Figure 1: Proposed architecture of fixed-width modified booth multiplier

The multiplication consists of two factors, one is named the multiplicand, the other one multiplier. In general multiplication takes place by adding the multiplicands after shifting depending on the position of the positive correlated bit of the multiplier. This leads to a defined number of partial products which equals the number of bits of the multiplier. As a result logic has to be designed for as many partial products as bits of the multiplier. Using the modified Booth recoding [4] technique leads to the advantage that the number of partial products to be added is reduced to one half of the original wordlength. As a consequence the delay and area occupation shrinks substantially. A reduced number of partial products minimizes the number of additions to consolidate the result. The realisation of the multiplier can be divided into three sections. The first section calculates the partial products to be added. These partial products are reduced in a second stage to two final bit vectors and in a third step the final addition of those two bit vectors is realized. Therefore the fixed-width booth multiplier consists of booth encoder which generate the partial products, error compensation circuit to compute the compensation value and adders to sum up the partial products.

In this project work, an error compensation circuit for the fixed-width modified Booth multiplier with high speed and simpler hardware structure using simple logic gate is proposed. Also implemented the fixed width booth multiplier with ACPE using different adders like ripple carry, carry look-ahead and carry select adders and its performance are compared. The proposed architecture is shown in figure 1.

V .PROPOSED BOOTH ENCODER

For the calculation of the partial products the multiplier is partitioned into three-bit-groups that overlap by one bit and are recoded in the range of {-2, -1, 0, 1, 2}. The first group consists of the two LSBs of the multiplier and ´0´. The following groups are constructed by the next two consecutive bits of the multiplier plus the MSB of the previous group. The order of significance of the bits in the groups remains the same as in the multiplier. The groups are recoded as in Table II.

As a result we receive the fixed number of partial products as mentioned. In the case of a 32 x 32 bit multiplication 16 partial products are produced. Depending on the result of the recoding, the multiplicand affects addition process positive, negative, once, twice or will not be added. The modified booth encoder can be directly implemented using behaviour modelling in VHDL. The figure 2 shows booth encoder of 10×10 multiplier for partial product generation.

Table II: Modified booth encoder

y_{2i+1}	y_{2i}	y_{2i-1}	y_i'	nz_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	2	1
1	0	0	-2	1
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	0

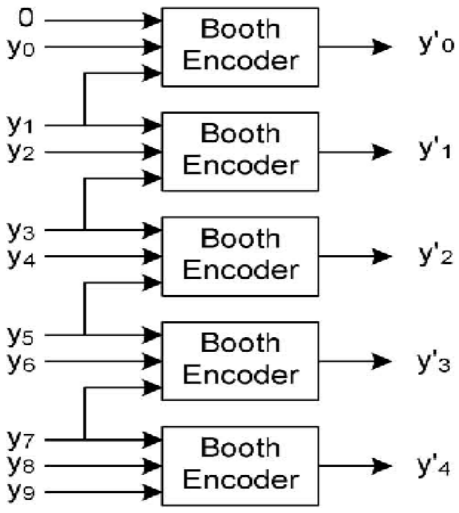


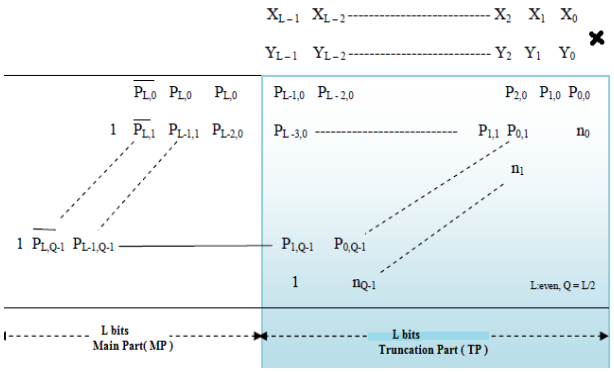
Figure 2: Example of 10 × 10 Booth Multiplier (Booth encoder)

VI . PARTIAL PRODUCT ARRAY

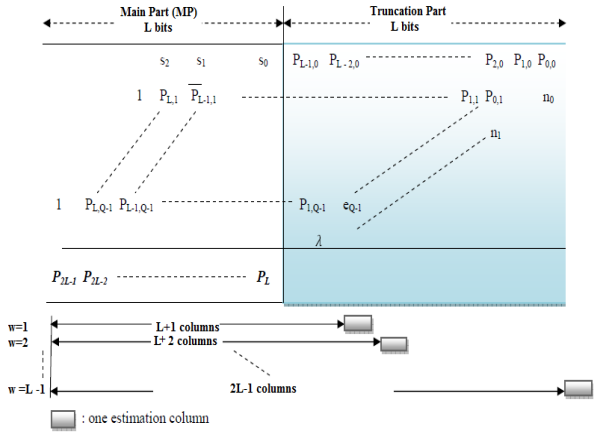
Sign Extention In Modified Booth Algorithm

In the case of conventional post truncated booth multiplier, some of the products are truncated by using a rounding operator to hold the data length fixed in L-bit. Therefore, an extra one binary bit 1 added into the most significant column of truncation part in figure 3(a), which indicates the rounding off operation of the P-T Booth multipliers.In the figure 3(b) in compared with figure 3 (a):

$$S0 = p_{L,0}$$
$$S1 = p_{L,0}$$
$$S2 = \overline{p_{L,0}}$$
$$\lambda = 1$$
$$e_{Q-1}=p_{0,Q-1}+n_{Q-1} \quad (3)$$



(a)



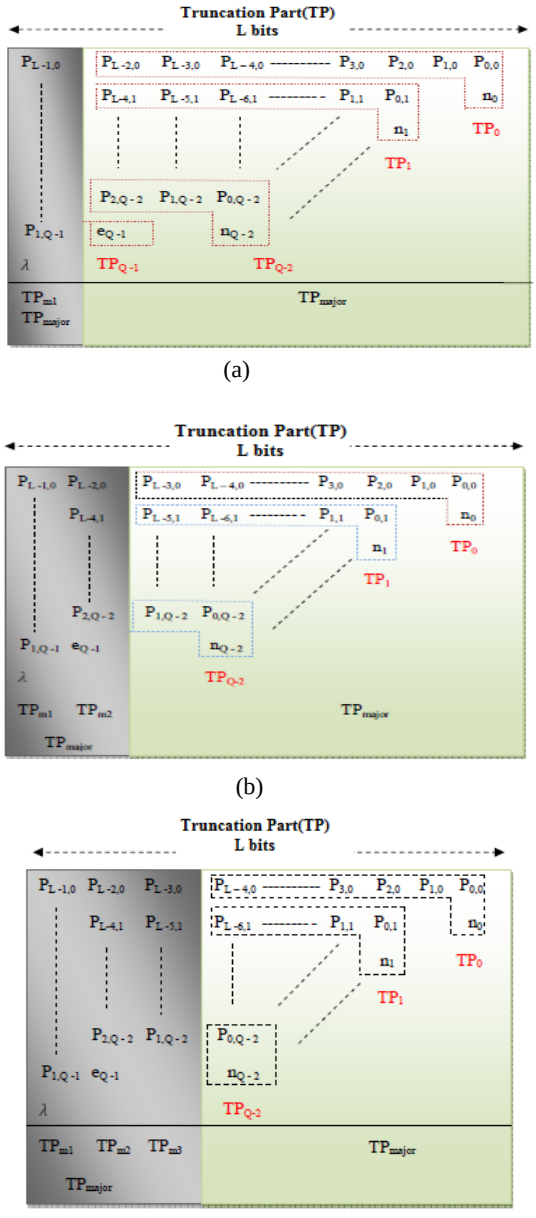
(b)

Figure 3: Algorithms of L× L fixed width booth multiplier (a) Conventional post-truncated booth algorithm of L× L booth multiplier (b) modified algorithm of L× L booth multiplier

Truncation part

The partial product array in Figure 4 also can be divided into two parts: the main part (MP) including

the most significant columns, and the truncation part (TP) including the least significant columns. Besides, the column information *w* is included to adjust accuracy with respect to system requirements, and *w* means that L + *w* most significant columns (MSCs) are calculated and the $(L + w + 1)^{th}$ MSC is chosen to estimate the compensation values.Therefore, L + *w* +1 MSCs are used to produce the results.



(a)

(b)

(c)

Figure 4 : Partition of truncation part with (a) *w* =1,(b) *w* =2, (c) *w* =3

Taking *w* =2 as an example, the results are calculated by the partial products of most significant L + 3(=L + *w* +1) columns. It can be seen in figure 4.

VII .FINAL STAGE ADDER

As adders are one of the most widely used components in integrated circuits, designing efficient adders has been the goal of much research in VLSI design. Ripple carry adders (RCAs) have the most compact design among all types of adders, they are the slowest types of adders, the other hand, Carry Look-ahead adders (CLAs) are the fastest adders, but they are the worst from the area point of view. Carry select adders have been considered as a compromise between RCAs and CLAs because they offer a good trade-off between the compact area of RCAs and the short delay of CLAs.

In this project work, the fixed-width modified booth multiplier with ACPE (FWBMACPE) is implemented using these different adders and its performance are compared. The number of gates and delay of 10×10, 16×16, 32×32 bit multipliers with different final stage adders are synthesised and compared with corresponding Conventional Multipliers.

VIII. PROPOSED ERROR COMPENSATION CIRCUIT

In the adaptive conditional probability formula, the σ depends on nz_j, the non-zero conditional code in the TP_{minor} part. So the various possibilities of nz_j are taken from 0 to Q-1-$\left\lfloor \frac{w}{2} \right\rfloor$. The design of the proposed compensation circuit can be got from the truth table given by table III.

Table III: Truth table for compensation circuit

nz(0)	nz(1)	nz(2)	nz(3)	ecomp
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Karnaugh map reductions are made for the truth table. The equation using Karnaugh map reduction can be given as in equation (4).

$$ecomp = ((nz(0)\ .\ nz(1)) + (nz(2).nz(3))) + (nz(0)+nz(1))\ .\ (nz(2)+\ nz(3)) \quad (4)$$

The circuit implementation for the equation (4) is shown in figure 5.

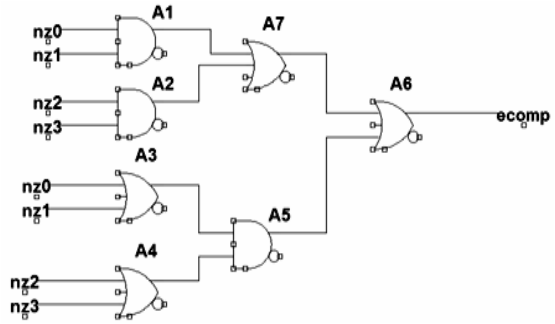
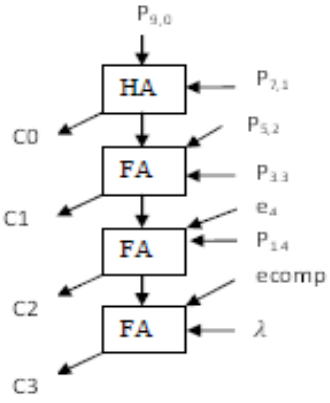
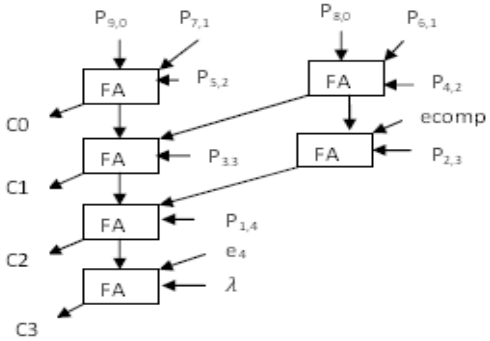


Figure 5: Proposed compensation circuit using simple gates.

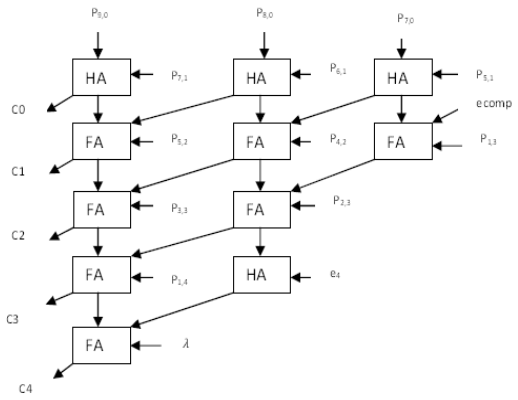
The ecomp from the compensation circuit is given to the TP_{major} of the partial product array and the implementation of truncation part of L=10 bit for w =1,w = 2 and w = 3 are shown below in figure 6.The proposed compensation circuit was compared with previous circuits like compensation circuit of ACPE multiplier. The comparison results are shown in table IV.



(a)



(b)



(c)

Figure 6: Proposed Compensation circuits (10×10) for (a) *w* =1,(b) *w* =2, (c) *w* =3

The comparison results show that the critical path delay of the proposed compensation circuit is lesser when compared with the ACPE compensation circuit. Also, for the proposed design, there is reduction in the number of gates which makes the proposed design better than ACPE circuit.

Table IV: Comparison result for compensation circuit

L = 10 bits	Delay	Number of gates
Proposed Compensation Circuit	6.142ns	6
ACPE	6.425ns	21

IX. SIMULATION RESULTS

The proposed modified Booth multiplier is synthesised in XILINX ISE 8.1 using VHDL code and simulated using Modelsim. The simulation and synthesis results are shown below.

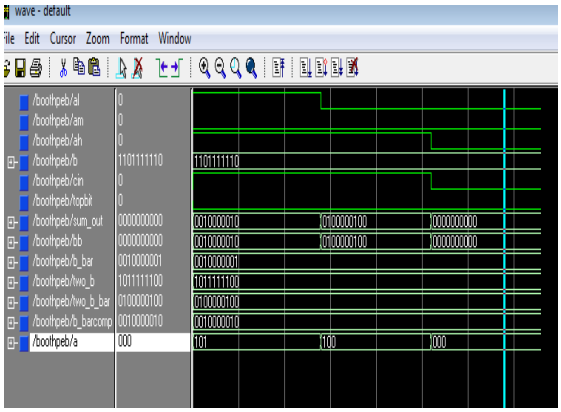


Figure 7: Simulation result of booth encoder -10×10 FWMBM

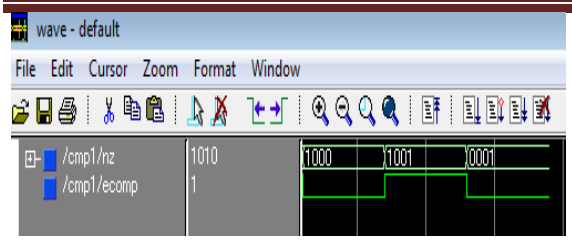
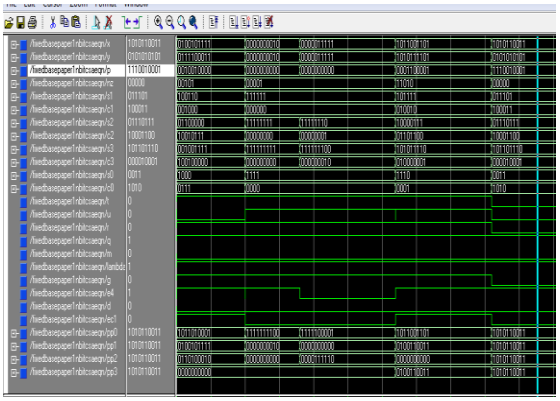
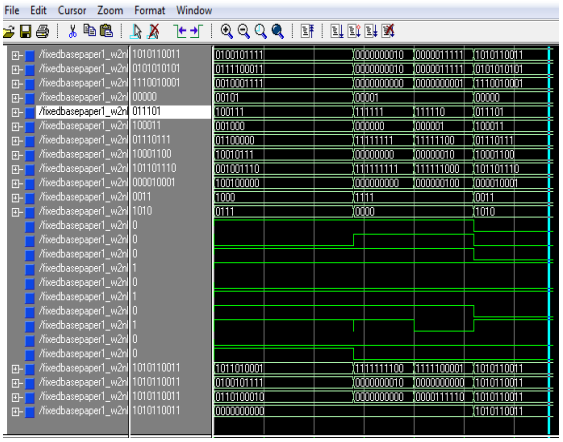


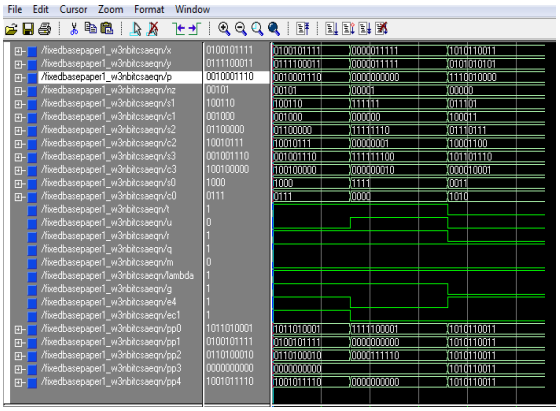
Figure 8: Simulation Waveform of proposed compensation circuit



(a)



(b)



(c)

Figure 9 : Simulation waveform of proposed FWMBM with (a) w=1(b)w=2 (c)w=3

Table V: Synthesis of conventional booth multipliers

	Delay	Number of gates
Conventional L= 10 booth multiplier	36.696ns	1584
Conventional L=16 booth multiplier	54.149ns	5034
Conventional L=32 booth multiplier	99.531ns	20631

Table VI: Synthesis comparison for L=10bits FWMBM with ACPE

Logic utilisation L=10 bits	FWMBM with ACPE using CPA			FWMBM with ACPE using CLA			FWMBM with ACPE using CSLA		
	w =1	w=2	w=3	w =1	w=2	w=3	w =1	w=2	w=3
GATE COUNT	1161	1260	1392	1167	1266	1320	1185	1284	1338
DELAY(ns)	30.98	32.47	30.02	30.94	32.42	32.49	29.09	30.59	30.53

Table VII: Synthesis comparison for L=16bits FWMBM with ACPE

Logic utilisation L=16 bits	FWMBM with ACPE using CPA			FWMBM with ACPE using CLA			FWMBM with ACPE using CSLA		
	w =1	w=2	w=3	w =1	w=2	w=3	w =1	w=2	w=3
GATE COUNT	3162	3348	3459	3192	3378	3489	3228	3414	3525
DELAY(ns)	38.93	40.29	42.83	38.58	38.67	40.80	32.48	32.69	34.64

Table VIII: Simulation comparison for L= 32bits FWMBM with ACPE

Logic utilisation L=32 bits	FWMBM with ACPE using CPA			FWMBM with ACPE using CLA			FWMBM with ACPE using CSLA		
	w =1	w=2	w=3	w =1	w=2	w=3	w =1	w=2	w=3
GATE COUNT	12564	12984	13332	12870	13293	13638	12720	13155	13500
DELAY(ns)	73.98	75.47	75.60	66.49	67.89	68.03	54.98	54.99	55.08

Table IX: Synthesis result of proposed FWMBM

Logic utilisation L=10 bits	Proposed FWMBM		
	w =1	w=2	w=3
GATE COUNT	1179	1278	1326
DELAY(ns)	27.99	29.48	29.54

The proposed fixed width modified Booth multiplier can be considered as a better multiplier than the other multipliers since the critical path delay are reduced and there is comparable reduction in numbers of gates with respect to conventional booth multipliers.

VI. CONCLUSION

In this project, a high speed fixed-width modified Booth multiplier has been proposed. In the proposed multiplier, the n most significant bits of the partial product are taken and remove the adder cells from the n least significant bits of 2n-bit output product. A simplified compensation circuit using simple logic gates has been designed to realize the compensation function. Implementation results showed that the proposed fixed-width modified Booth multiplier can achieve reduction in the area as well as critical path delay when compared with the previous circuits. The proposed architecture can be applied in DCT applications. This architecture can be easily applied to large length Booth multipliers for achieving higher speed performance.

REFERENCES

[1] Yuan-Ho Chen and Tsin-Yuan Chang**,** "A High-Accuracy Adaptive Conditional Probability Estimator for Fixed-Width Booth Multipliers," *IEEE Trans.* circuits and systems—I: regularpapers, vol. 59, no. 3, March 2012.

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

[3] C. Y. Li, Y. H. Chen, T. Y. Chang, and J. N. Chen, "A probabilistic estimation bias circuit for fixed-width Booth multiplier and its DCT applications," *IEEETrans.Circuits Syst. II, Exp. Briefs*, vol. 58, no. 4, pp. 215–219, Apr. 2011.

[4] J. P. Wang, S. R. Kuang, and S. C. Liang, "High-accuracy fixed-width modified Booth multipliers for lossy applications," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol 19, no. 1, pp. 52–60, Jan. 2011

[5] S. R. Kuang, J. P. Wang, and C. Y. Guo, "Modified Booth multipliers with a regular partial product array," *IEEE Trans. Circuits Syst. II, Exp.Briefs, vol. 56, no. 5, pp. 404–408,May 2009.*

[6] H. A. Huang, Y. C. Liao, and H. C. Chang, "A self-compensation fixedwidth Booth multiplier and its 128-point FFT applications," in *Proc. IEEE Int. Symp. Circuits Syst.,2006, pp. 3538–3541.*

[7] Kiat-Seng Yeo and Kaushik Roy," Low-voltage,Low-power VLSI subsystems" ,McGrawHill Edition 2009.

[8] 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.

[9] M. J. Schulte and E.E. Swartzlander, Jr. "Truncated multiplication with correction constant", *VLSI Signal Processing, VI* New York: IEEE Press, 1993, pp. 338-396.