International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

A Comparison of Different Fixed Width Multipliers Based On MLCP

Rima N[1], Nisha R[2]

[1]M. Tech Student (VLSI & ES), Department of ECE, FISAT, MG University Kerala, India

[2]Assistant professor, Department of ECE, MG University, Kerala, India

Abstract: ***The fixed-width multiplier is attractive to many multimedia and digital signalprocessing systems which are desirable to maintain a fixed format and allow a little accuracy loss to output data. This brief includes a comparative study of different fixed-width Booth multipliers- PT(most accurate fixed width multiplier) and DT(least area requirement) fixed width multipliers with the Proposed accuracy-adjustment fixed-width Booth multipliers that compensates the truncation error using a multilevel conditional probability (MLCP) estimator.To consider the trade-off between accuracy and area cost, the MLCPprovides varying column information to adjust the accuracy with respect to system requirements. Unlike previous conditional-probability methods, the proposed MLCP uses entire nonzero code, namely MLCP, to estimate the truncation error and achieve higher accuracy levels. And a comparative study of different fixed width multipliers based on MLCP using differentfast adders such as carry look ahead adder, Kogge-stone adder etc are also included. The design was modeled using Verilog, simulated and synthesized using Xilinx ISE 14.7.***

Keywords: Direct truncated(DT),multi -level conditional probability(MLCP), Post truncated (PT)

1. Introduction

Multiplier is a widely used component for digital signal processing applications such as discrete cosinetransform(DCT), fast fourier transform(FFT), finite impulse response(FIR) filters etc. In some DSP applications such as digital filters,wavelet transformers etc, it is necessary that the width of arithmetic data should remain fixed throughout the entire computation. The fixed-width multiplier is attractive to many multimedia and digital signal processing systems which are desirable to maintain a fixed format and allow a little accuracy loss to output data. In order to achieve this goal a fixed width multiplier capable of receiving an N bit multiplicand and an N bit multiplier and producing N bit result is necessary. And generally in-order to produce an output that is having the same width as input the fixed with multipliers truncate half least significant bits. And therefore truncation error can occur in fixed width multiplier designs.

Truncation can be performed in two ways viz; Post truncation(PT)and direct truncation(DT).post truncated fixed width multipliers are considered as the fixed width multipliers with highest accuracy.PT fixed width multipliers truncates half of LSBs results after calculating all the products and gives high accuracy but it takes large circuit area to calculate the truncation part products. By contrast a direct truncated fixed width multipliers will truncate half of the least significant partial products directly to conserve the circuit area but it produces truncation error.

To effectively reduce the truncation error, various error compensation methods, have been proposed. The error compensation value can be produced by the constant scheme or the adaptive scheme. The constant schemepre-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. By contrast the adaptive scheme was developed to achieve higher accuracy than the constant scheme through adaptively adjusting the compensation value accordingto the input data at the expense of a little higher hardware complexity. To achieve a balanced design between accuracy (P-T) and area cost(D-T), several researchers have presented various error-compensated circuits to alleviate the truncation errors. Majority of these works are done on modified booth multipliers due to the high-speed computation and also few partial products are truncated after Booth encoding, therefore these multipliers have a smaller truncation error. Therefore, the truncation error of the fixed-width Booth multiplier is reduced due to the decreasing of the truncated partial products.

For this reason, several error-compensation works are presented for fixed-width Booth multipliers design. To reduce the hardware complexity, Jouet al.present statistical and linear regression analysis to reduce the hardware complexity [2]. However, the truncation error cannot be depressed because the input information is limited in estimating the carry propagation from the truncated part. A self-compensation approach [3] using conditional mean method is presented to reduce the hardware complexity. In [4] and [5], by taking more information provided by Booth encoder, the compensation bias can reduce the truncation error with the huge area penalty. Besides, Song et al. present binary threshold and more partial products for error compensation bias to reduce truncation error [6], and the hardware cost is increased, too. In addition, for the high-accuracy applications, more information of partial products to estimate the error compensation bias can achieve higher goal of accuracy [6]. The generalized forms of more than columns' information to estimate error-compensation biases are derived in [6]. Nevertheless, the resulting carry estimation circuit must be designed in a heuristic way, and that the high-accuracy fixed-width multipliers would result in large circuit area is a constant truth. Therefore, building an area-efficient estimation circuit with high accuracy is a challenging task.In earlier adaptive conditional probability estimator was used to improve accuracy, it uses single

Licensed Under Creative Commons Attribution CC BY

nonzero code to calculate the truncation errors whereas the MLCP method employing all nonzero code to estimate truncation error. The compensated circuit will respond quickly, produces a closed form with various bit-widths L and column information w. Thus accuracy can be adjusted by changing column information*w*.

This paper is organized as follows. In Section II, the back-ground of the fixed-width modified Booth multiplier is given. The proposed MLCP method is discussed in Section III, which includes the generalized form's derivation,the systematic procedure and the proposed MLCP circuit .Section IV includes the results and comparative study..The comparisons of accuracy area, are made in this Section . In addition, SectionI V presents the performance of MLCP with the proposed compensation circuit. Finally, conclusions of this paper are drawn in Section V.

2. Fixed Width Modified Booth Multiplier

Multiplication can be divided into three steps-:

a) Generating partial products
b) Summing up all partial products until only two rows remain
c) Adding the remaining two rows of partial products by using a carry propagation adder.

In the first step, two methods are commonly used to generate partial products. The first method generates partial product directly by using a 2-input AND gate. The second one uses Baugh Wooley, radix-2, radix-4 modified Booth's encoding (MBE),and radix-8 to generate partial products. Radix-4 MBE has been widely used in parallel multipliers to reduce the number of partial products by a factor of two. Baugh Wooley generally not used because they are not suitable for large size operands. Techniques like Wallace tree, Compressor tree etcare used in the second step to reduce the number of rows of the partial product. During third step, advanced adding concepts like carry-look-ahead, carry select adderetc are used.

Modified Booth encoding is commonly used in multiplier designs to reduce the number of partial products. It is known as the most efficient Booth encoding and decoding scheme. To multiply X by Y using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {-2, -1, 0, 1, 2}.Table1 shows the rules to generate the encoded signals by MBE scheme and Fig.1 shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Fig. 2.

The 2L-bit product P can be expressed in two's complement representation as follows:

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

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

$$P = X * Y \qquad (1)$$

TableII lists three concatenated inputs y_{2i+1} , y_{2i} ,and y_{2i-1}mapped into y_i' using a Booth encoder, in which the nonzero code z_i is an one-bit digit of which the value is determined according to whether y_i' equals zero. If it is zero ,z_i will be zero and in all other cases it will not be a zero.. After encoding, the partial product array with an even width L contains Q=L/2 rows.

Table 1: Truth Table for Booth Encoder

	y_i	y_{i-1}	Operation	X	2X	Neg
0	0	0	+0 *X	0	0	0
0	0	1	+1 * X	1	0	0
0	1	0	+1 * X	1	0	0
0	1	1	+2 * X	0	1	0
1	0	0	-2 * X	0	1	1
1	0	1	-1 * X	1	0	1
1	1	0	-1 * X	1	0	1
1	1	1	-0 * X	0	0	1

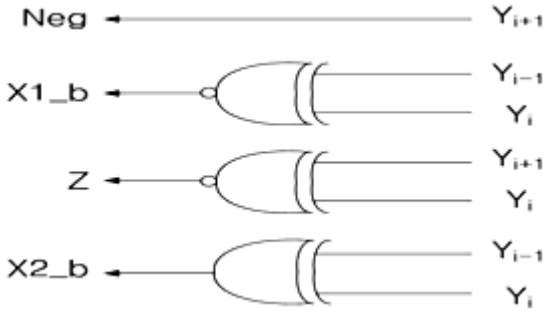

Figure 1: Encoder for modified booth encoding scheme

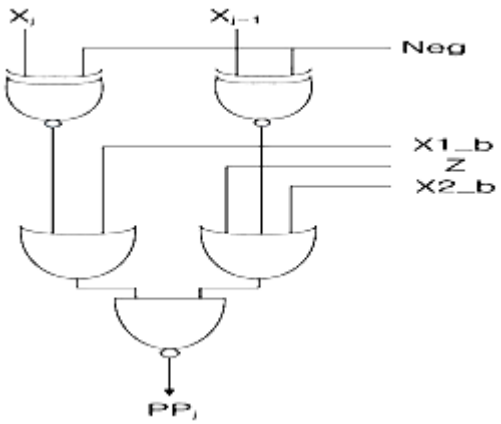

Figure 2: Decoder for MBE scheme

Table 2: Mapped Table of A Modified Booth Encoder

y_{2i+1}	y_{2i}	y_{2i-1}	y_i'	z_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**

A . A typical implementation

Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then

Licensed Under Creative Commons Attribution CC BY

performing a rightward arithmetic shift on P. Let m and r be the multiplicand and multiplier, respectively; and let x and y represent the number of bits in m and r.

1.Determine the values of A and S, and the initial value of P. All of these numbers should have a length equal to (x + y + 1).

A: Fill the most significant (leftmost) bits with the value of m. Fill the remaining (y + 1) bits with zeros.

S: Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining (y + 1) bits with zeros.

P: Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant (rightmost) bit with a zero.

2. Determine the two least significant (rightmost) bits of P.
- If they are 01, find the value of P + A. Ignore any overflow.
- If they are 10, find the value of P + S. Ignore any overflow.
- If they are 00, do nothing. Use P directly in the next step.
- If they are 11, do nothing. Use P directly in the next step.

3. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.

4. Repeat steps 2 and 3 until they have been done y times.

5. Drop the least significant (rightmost) bit from P. This is the product of m and r.

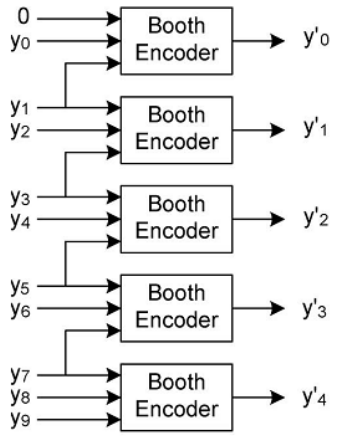

Figure 3: Booth encoder architecture

3. Design of Fixed Width Booth Multiplier

The 2n bit product of the n by n 2-complement multiplication can be divided in to two sections

$$P = XY = MP + TP \quad (2)$$

It is known that the most accurate truncated product is given by

$$P = MP + \sigma \ * 2^n \quad (3)$$

Where σ is the compensated bias presented by the multilevel conditional probability (MLCP) estimator. This can be further decomposed into TP_{Major} and TP_{Minor} parts as

$$\sigma = \lfloor TP_{Major} + TP_{minor} \rfloor \ (4)$$

Where $\lfloor . \rfloor$ represents rounding operation.The TP_{Major} and TP_{Minor} are the major and the minor compensation part in TP,respectively. TP_{Major} has greater weight than TP_{Minor} with regard to its effect on MP.The TP_{Major} provides true information to estimate MLCP and is same as MP , and the TP_{Minor} contributes compensation bias to MP based on conditional probability estimation or the expected value. Therefore, the compensation bias can be calculated by obtaining TP_{Major} and estimating TP_{Minor} . Column information w is introduced to adjust TP_{Major} and is adapted to adjust the accuracy for different applications.

A. Derived MLCP Formula

The TP can be partitioned into encoding group set (G)and column set (T) as shown in fig(3). By inspecting the figure it can be seen that G contain encoding groups and T contain column groups.The encoding groups in G are defined as follows:

$$G_0 = 2^{-L}(P_{0,0} + n_0) + \cdots + 2^{-1-w} P_{L-1-W,0}$$
$$G_1 = 2^{-(L-2)}(P_{0,1} + n_1) + \cdots + 2^{-1-w} P_{L-3-W,0}$$
$$\vdots$$
$$G_{Q-1} = 2^{-2}(P_{0,Q-1} + n_{Q-1}) \ (5)$$

And the column groups in Tset are defined as follows:

$$T_1 = 2^{-1}(P_{L-1,0} + P_{L-3,1} + \cdots + P_{1,Q-1})$$
$$T_2 = 2^{-2}(P_{L-2,0} + P_{L-4,1} + \cdots + n_{Q-1})$$
$$\vdots$$
$$T_L = 2^{-L}(P_{0,0} + n_0) \ (6)$$

Licensed Under Creative Commons Attribution CC BY

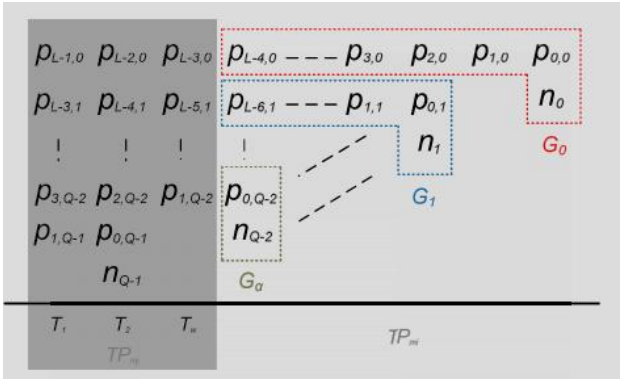

Figure 3: G set and T Set of the proposed booth multiplier with w=3

The value of TP_{Major} and TP_{Minor} can be obtained from the encoding groups in set G and column groups from set T as shown in the following equations:

$$TP_{mj} = T_1 + T_2 + \cdots + T_W = \sum_{i=0}^{w} TP_i \quad (7)$$

$$TP_{mi} = G_0 + G_1 + \cdots + G_\alpha = \sum_{i=0}^{\alpha} G_i \quad (8)$$

where $\alpha = Q - 1 - \left\lfloor \frac{w}{2} \right\rfloor$ where, $\lfloor . \rfloor$ represents the flooring operation. Here we can see that the TP_{mi} depends upon the term α which in turn depends on the column information w . Therefore the compensation bias also depends on column information. ie, the accuracy can be adjusted by varying the parameter column information.
The expected value of TP_{Minor} can be calculated as:

$$E[TP_{Minor}] = \mathrm{E}[(TP_0 + TP_1 + \cdots \ldots \ldots \ldots TP_\alpha)|Z]$$

$$= E[TP_0|Z_0] + E[TP_1|\{Z_0, Z_1\}] + \cdots .. E[TP_\alpha|Z] = E_0 + E_1 + \cdots \ldots \ldots \ldots E_\alpha = \sum_{i=0}^{\alpha} E_i \quad (9)$$

This conditional probability values depends greatly on the non -zero code z which is obtained from booth encoding which in turn depends on the inputs that are fead to the multiplier. Therefore the compensated circuit will improve the accuracy of the fixed width multiplier.

Equation (8)can be approximated as:

$$TP_{mi} \cong sone * 2^{-w} \quad (10)$$

$$\text{Where } sone = \begin{cases} \left\lfloor \frac{\beta - 1}{2} \right\rfloor \; when \; z \neq 00..0 \\ 0 \; when \; z = 00..0 \end{cases} \quad (11)$$

Where $sone$ is the sum of non-zero code z with corresponding w. Therefore

$$\sigma = \left\lfloor TP_{mj} + sone * 2^{-w} \right\rfloor \quad (12)$$

B. Architecture of the proposed booth multiplier

The modified Booth multiplier can be implemented in accordance with (12). Fig. 4 illustrates the entire architecture of the proposed Booth multiplier, which includes a modified booth encoder for generating reduced number of partial products, tree-based carry-save reduction in-order to further reduce the partial product array to the addition of only two operands followed by parallel-prefix adder for adding the remaining two rows.. The steps involved can be summarized as follows:

1) Select the specifications: Word length L and column information w are selected based on the accuracy requirement of the application.
2) The value of α and compensation bias σ can be calculated from the specifications selected.
3) Compensation circuit design: Compensation bias is obtained by summing TP_{mj} and TP_{mi} by using a carry save adder(CSA) tree. The tree is comprised of full adders an half adders.
4) Design of MP circuit: All of the partial products in MP and carry value from compensation circuit to MP are summed using the CSA tree and parallel-prefix adder. The CSA is comprised of 4-2 adders, FAs, or Has. The high speed 4-2 adder comprises two Fas. The priority of 4-2 adder is higher than FA and HA.

To lower the latency of partial product accumulation stage 4-2 compressors are widely employed nowadays for high speed multipliers.fig 6 shows the conventional implementation of 4-2 compressor.It is composed of two serially connected full adders and comprises of 5 inputs x1,x2,x3,x4 and receives an input c_{in} from preceedingmodule.And produces3 outputs a sum, a carry and c_{out} which propogates to the next module. c_{out} is independent of c_{in} .therefore speed performance of multiplier increases.

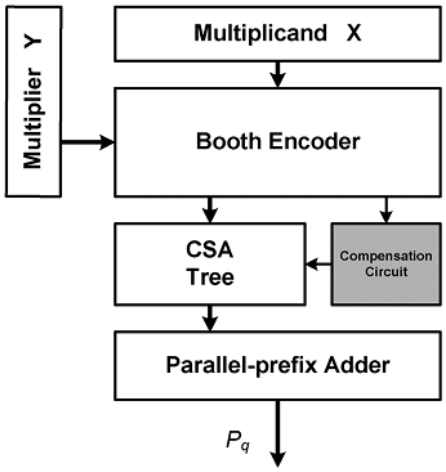

Figure 4: The whole architecture of the proposed booth multiplier

Licensed Under Creative Commons Attribution CC BY

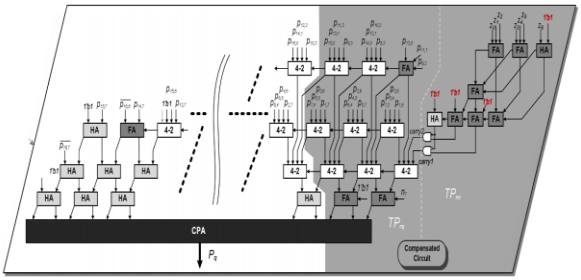

Figure 5: Architecture of the proposed MLCP Booth multiplier for L=16 and w=3.

Fig. 5represents the detailed architecture of the propose booth multiplier with L=16 and column information w=3.Carry save adder architecture and function of subtracting one is designed by adding all one values for twos complementation representation. Theproposed MLCP circuits depend on α, thus,various word lengths L and column information w can use the same MLCP circuit.

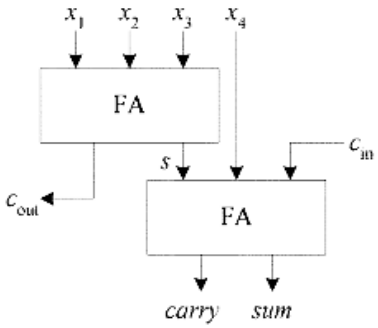

Figure 7: 4-2 Compressor

4. Performance Comparison and Results

4.1 Performance comparisons

4.1.1 comparison of MLCP multiplier with DT and PT fixed width multipliers

This sub-section addresses the accuracy,and area of direct truncated (DT) and post truncate (PT) fixed-width Booth multipliers with that of the proposed MLCP multiplier.The proposed multiplier is expected to achieve a balance between area and accuracy.for comparison, we also implemented and synthesized the conventional post truncated multiplier with the partial product array similar to the one shown in Fig7.Consider the example shown. Let the inputs given are:

din1=0110100000010001(26641)
din2=0000100010100001(2209)

Table 3: Comparison of DT, PT AND MLCP fixed width booth multipliers

The output obtained by DT	0000011000110111	1591
The output obtained by PT	0000011010001001	13384
The output obtained by proposed MLCP method	0000001111001001 00	3876

By inspecting the results it can be seen that the result obtained from MLCP is more accurate than DT but it is not as accurate as the result obtained from PT fixed width multiplier which is the fixed width multiplier with highest accuracy. And since it does not calculate all partial products in contrast to PT, it requires less area than that of PT fixed width multipliers.Ie the proposed fixed width multiplier based on MLCP achieves a balance between accuracy and area.

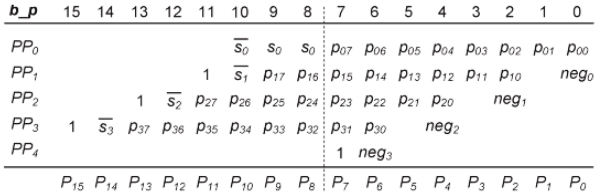

Figure 7: MBE partial product arrays for 8×8 post truncated multiplication

4.1.2 Comparison of various MLCP multipliers using different fast adders

For comparison, we have implemented several MLCP multipliers whose final parallel –prefix adder is replaced by using different fast adders.The adder block in the proposed MLCP architecture is replaced with different fast adders and the area and delay are compared. Adder blocks used are ripple carry adder, carry look ahead adder, kogge-stone adder ,parallel prefix adder and Brent -kung adder.These multipliers were modeled in Verilog HDL and synthesized by using xilinx ISE 14.7inorder to compare area and delay. Among all these architectures the fastest architecture for the proposed method is the implementation using Kogge- stone adder. The slowest architecture is the one using ripple carry adder.

4.1.2.1 Ripple carry adder:

The ripple carry adder is composed of a chain of full adders with length n, where n is the length of the input operands.The following Boolean expressions describe the full adder.

$$p = a \oplus b \; and \; g = a.b \qquad (13)$$

Here a and b are the input operands and p and g are the propagate and generate signals respectively. Carry is propagated if p is high or is generated if g is high.Thus, the sum S and carryout C_o signals can be expressed as:

$$s = a \oplus b \oplus c_i = p \oplus c_i \; and$$
$$c_o = g + p.c_i \qquad (14)$$

4.1.2.2 Carry Look-ahead Adder

Weinberger and Smith proposed this scheme in 1958 . It uses look-ahead technique rather than carry-rippling technique to speed-up the carry propagation. By using additional logics, group generate and propagate signals can be generated. Equation15 Equation16 and Equation 17 show the logical expression of prefix-4 group generate, propagate and carry-out signals respectively. Thus, multiple levels of carry-lookahead logics can be used to propagate carry-in from the least significant bit (LSB) to the most significant bit (MSB).

$$P^* = P_0 P_1 P_2 P_3 \qquad (15)$$
$$G^* = g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0 \qquad (16)$$
$$C_{out} = G^* + P^*.C_{in} \qquad (17)$$

Licensed Under Creative Commons Attribution CC BY

4.1.2.3 Kogge-Stone Adder

Kogge and Stone proposed a general recurrence scheme for parallel computation in 1973 . The tree structure of this generalization is shown in Figure 3-3. Adders implemented using this technique is in favor due to the following:

1) Regular layout
2) Controlled fan-out

However, they are nothing but prefixed carry-lookhead adders. The intermediate carries is generated by replicating the prefix tree shown in Figure 8 at every bit position. Figure 9 shows the resultant prefix graph of a 16-bit prefix-2 Kogge Stone adder.

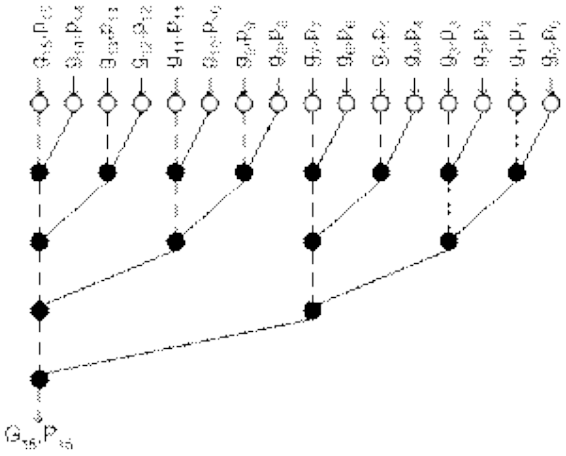

Figure 8: 16-bit Prefix-2 Tree Structure

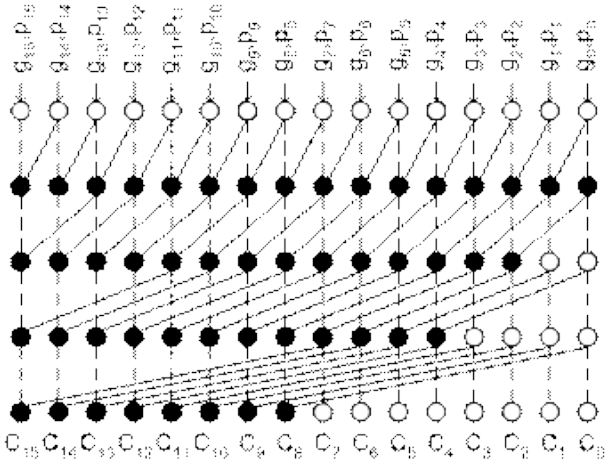

Figure 9: Prefix graph of 16-bit prefix-2 Kogge Stone adder

4.1.2.4 Brent-Kung Adder

The replicated Kogge Stone structure to generate intermediate carries shown in Figure 9 is very attractive to high-performance applications. However, it comes at the cost of area and power. A simpler tree structure could be formed if only the carry at every power of two positions is computed as proposed by Brent and Kung . Figure 10 shows a 16-bit prefix tree of their idea. An inverse carry tree is added to compute intermediate carries. Its wire complexity is much less than that of Kogge Stone adder.

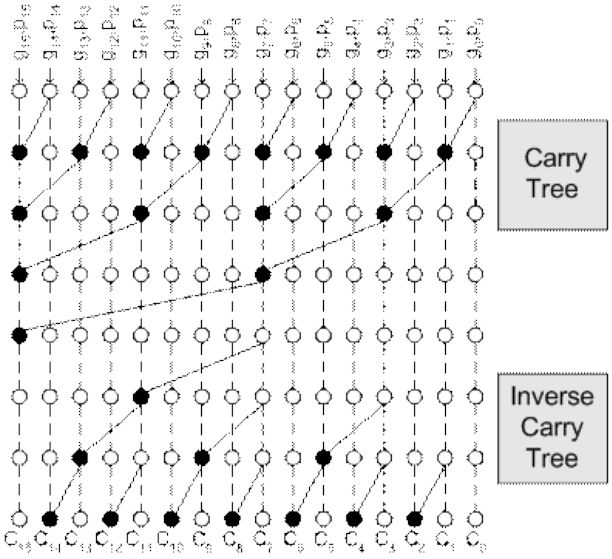

Figure 10: 16-bit Brent-Kung Scheme

Table 4: Comparison of Different MLCP Multipliers Using Various Fast Adders

Name of adder	*Area (1 unit/gate)*	*Delay (ns)*	*Number of LUTs*
Brent- kung	1484	18.301	224
CLA adder	1599	18.983	263
Kogge- stone	1416	18.261	214
Ripple carry adder	2568	20.006	414

4.2 Simulation Result

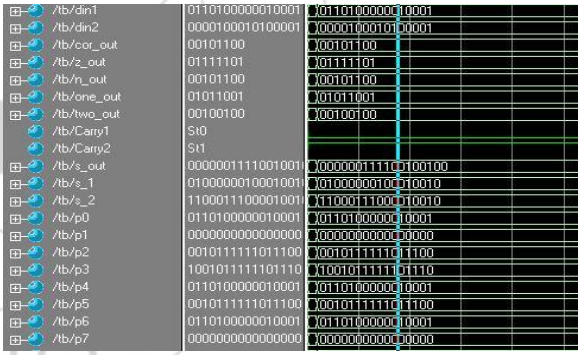

Figure 11: Simulation result

4.3 Synthesis Result

Table 5: Tabulation of Synthesis Results

No of slices	225/4656
Gate delay	13.359.100ns
Net delay	6.6474ns
Area(1 unit = 1 nand gate area)	2568

The total delay of the whole circuit is the total sum of delay associated with each single gate and interconnection between them. Our proposed MLCP multiplier is having a total delay of 20.006 ns, which includes a gate delay of 13.359ns and net delay of 6.647ns.The size of the circuit can be estimated on total number of gates used. The actual size of chip depends upon routing of these gates.

Licensed Under Creative Commons Attribution CC BY

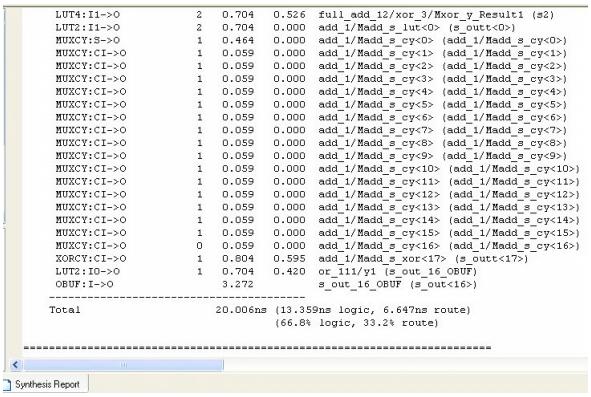

Figure 12: Synthesis result illustrating the delay

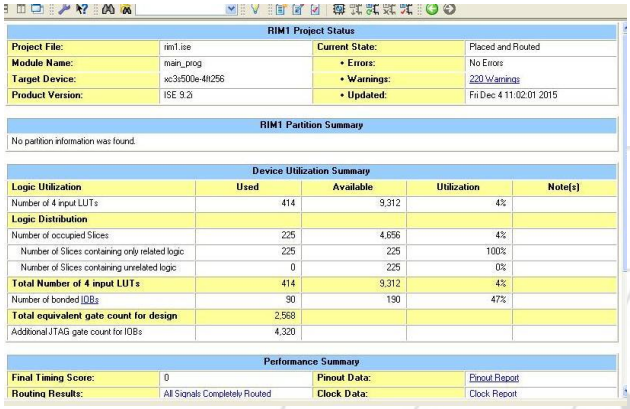

Figure 13: Synthesis result illustrating the delay

Figure 14 shows the RTL schematic of the proposed system. The proposed fixed width booth multiplier is having two 16 bit inputs din1, din2. The output which is an 18 bit number , the partial produts p0,p1,p2,p3,p4,p5,p6,p7 and the encoded results of a modified booth encoder cor_out, n_out, one_out, two_out, z_out can be obtaind as the outputs.

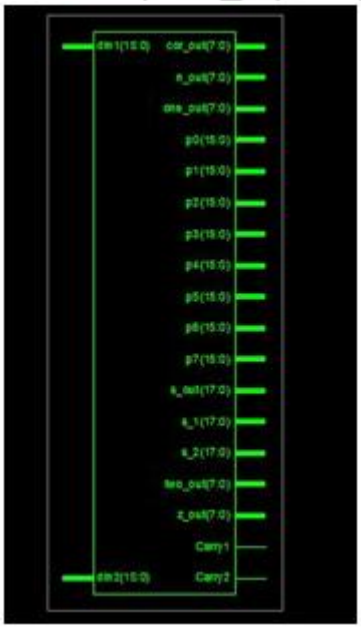

Figure 14: RTL schematic of the proposed MLCP multiplier

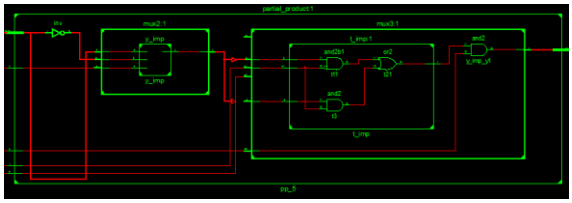

Figure 15: Partial product generation circuit

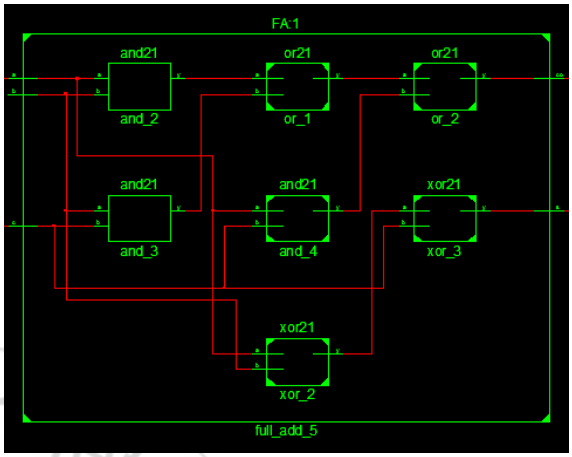

Figure 16: Full adder

5. Conclusions

This paper proposes a fixed-width Booth multiplier based on multi-level conditional probability.The MLCP Booth multiplier outperforms almost all previous solutions for accuracy loss in fixed width booth multipliers with regard to accuracy or circuit performance. Accuracy increased since it use more information from booth encoder and partial products.Introduction of column information *w* provide more choices between accuracy and area cost.the compensation function is also established to adjust the accuracy with respect to system requirements based on varying *w* .And also speed performance is higher since conditional probability is used in compensation circuit and also since fast CSA tree is used.

References

[1] Y. H. Chen, "An accuracy-adjustment fixed-width Booth multiplier based on multilevel conditional probability,"*IEEE Trans. Very Large Scale Integr.* (VLSI) Syst., vol. 23, no. 1, pp. 203–207, Jan. 2015.
[2] S. J. Jou, M. H. Tsai, and Y. L. Tsao, "Low-error reduced-width Booth multipliers for DSP applications,"*IEEE Trans. Circuits Syst. I*, vol. 50,no. 11, pp. 1470–1474, Nov. 2003.
[3] H. A. Huang, Y. C. Liao, and H. C. Chang, "A self-compensation fixed-width Booth multiplier and its 128-point FFT applications," *inProc.IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3538–3541.
[4] 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 (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.
[5] 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 (VLSI) Syst.*, vol. 19, no. 1, pp. 52–60, Jan. 2011.

Licensed Under Creative Commons Attribution CC BY

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

Author Profile

Rima N received the B.Tech degree in Electronics and Communication Engineering in 2014, from Calicut University, Calicut University Institute Of Engineering And Technology, Calicut and currently pursuing M.Tech in VLSI & Embedded system from M.G.University, Kerala, FISAT, Mookkannoor, AngamalyErnakulam, and Kerala, India.

Nisha R received the Bachelor's degree in Electronics and Communication Engineering and MTech. She has been working as Assistant professor at FISAT, Angamaly,Mookkannoor,Ernakulam, and Kerala, India.

www.ijsr.net
IJSR
ISSN (Online): 2319-7064