

# Implementation of Modified Lifting and Flipping Plans in D.W.T Architecture for Better Performance

G. Ashok Jeevan<sup>1</sup>, Y. V. A. Prasad<sup>2</sup>, Dr. B. Subrahmanyeswara Rao<sup>3</sup>

<sup>1</sup>Student, M.Tech , VLSISD, Dept. of ECE, Swarnandhra College of Engg. & Tec, *twentycraze@gmail.com*

<sup>2</sup> Associate Professor, Dept. of ECE Swarnandhra College of Engg. & Tec, *yaralagaddavap@gmail.com*

<sup>3</sup> Professor, Dept. of ECE, Swarnandhra College of Engg. & Tec, *boyina-a@yahoo.com*

**ABSTRACT:** Data compression is one of the major fields of research in the present world. In this regard image compression is also having its own significance, so many algorithms for DFT, DCT, DWT, etc. have been developed and among them DWT is most likely used there are two plans are existing for generating 2-DWT and are lifting and flipping plans. The above two plans architecture are having its own fixed scaling constants, multipliers and adders. In my project work I am proposing a lifting plan and flipping plan such that the modification is done in its internal architecture of S.M.B.Multiplier and radix-4 booth multiplier with replacing adder in it with spanning tree parallel prefix adder. This modification has improved Power Delay Product (PDP) by 7% in lifting plan and 5% in flipping plan.

**Keywords:** *Lifting and Flipping , Discrete Wavelet Transform, Spanning Tree Adder*

\*\*\*\*\*

## I. INTRODUCTION

Discrete wavelet transform (DWT) is famously utilized as a part of speech and picture compression [1]. There are two principle computational plans for the DWT calculation, to be specific the lifting plan [1], and the convolution-based plans. The lifting plan is prevalently utilized since it includes altogether less number of augmentations than convolution-based plans Flipping plan is an altered variant of lifting plans which is all the more prevalently utilized because of its lower basic way [2]. The match of scaling constants of lifting plans are ideal converse of each other while those of flipping plans are not ideal opposite of each other. In spite of this component, lifting plans isn't prominent like the flipping plans because of its more drawn out basic way. In this paper, we introduce basic way examination of lifting and flipping plans and propose a plan to diminish the basic way of lifting structure. Likewise we propose productive pipeline models for the usage of the DWT in view of lifting and additionally flipping plans. Since, the lifting plans offers sparing of a few multipliers when scaling operations are coordinated and performed in one stage, and it has possibly higher CPD than the flipping structure, lifting plan gives a superior region defer effective structure than flipping structure.

## II. EFFICIENT DATA-PATH FOR DWT COMPUTATION BY LIFTING AND FLIPPING PLAN

### A. Lifting and Flipping plans for 1-D DWT

The coefficients of 9/7 bi orthogonal wavelet filter can be computed by the lifting scheme as

$$s1(n) = x(2n - 1) + \alpha(x(2n) + x(2n - 2)) \quad (1a)$$

$$s2(n) = x(2n) + \beta(s1(n) + s1(n - 1)) \quad (1b)$$

$$uh'(n) = s1(n) + \gamma(s2(n) + s2(n - 1)) \quad (1c)$$

$$ul'(n) = s2(n) + \delta uh'(n) + uh'(n - 1) \quad (1d)$$

$$uh(n) = 1/k uh'(n) \quad (1e)$$

$$u\lambda(n) = K.u\lambda(n) \quad (1f)$$

Where  $x(n)$  is the input sequence,  $u1(n)$  and  $uh(n)$ , are respectively the low-pass and the high-pass components of 1-level DWT.  $\alpha, \beta, \gamma, \delta$  are the lifting constants for 9/7 bi-orthogonal wavelet filter.  $K$  and  $K^{-1}$  are respectively, the scale normalization constants of the low-pass and the high-pass lifting DWT coefficients. Interestingly, scaling constants  $K$  and  $K^{-1}$  are perfect inverse of each other. This is an important feature of lifting scheme which could be exploited to save several multipliers in a 2-D DWT structure.

Based on the flipping plan coefficients of 9/7 wavelet filter can be computed as

$$r1(n) = 1/\alpha x(2n-1) + x(2n) + x(2n-2) \quad (2a)$$

$$r2(n) = 1/\alpha\beta x(2n) + r1(n) + r1(n-1) \quad (2b)$$

$$uh'(n) = 1/\beta\gamma r1(n) + r2(n) + r2(n-1) \quad (2c)$$

$$ul'(n) = 1/\gamma\delta r2(n) + uh'(n) + uh'(n-1) \quad (2d)$$

$$uh(n) = \alpha/\beta\gamma K uh'(n) \quad (2e)$$

$$u\lambda(n) = K\alpha\beta\gamma\delta ul'(n) \quad (2f)$$

where,  $1/\alpha, 1/(\alpha\beta), 1/(\alpha\beta\gamma),$  and  $1/(\alpha\beta\gamma\delta)$  are the flipping constants of 9/7 bi-orthogonal wavelet filter.  $(k\alpha\beta\gamma)$  and  $((\alpha\beta\gamma)/k)$  are respectively, the scale normalization constants of the low-pass and the high-pass flipping DWT. Note that the scaling constants of flipping DWT are not perfect inverse of each other {i.e., determined a productive pipeline structure utilizing (1) for the calculation of 1-level lifting DWT. The structure is appeared in  $Fi(k\alpha\beta\gamma\delta) \times (\alpha\beta\gamma/k) \neq 1$  } not at all like those of the lifting DWT. We have g.1(a). The

internal structure of each lifting cell (LC) is appeared in Fig.1(b) and the lifting scaling unit (LSU) is appeared in Fig.1(c).The proposed lifting DWT structure includes 6 multipliers, 8 adders and 12 registers, and it computes two DWT parts in each cycle. The length of cycle period is  $T = T_M + 2T_A$  where  $T_M$  and  $T_A$  are the time required for multiplication and an addition

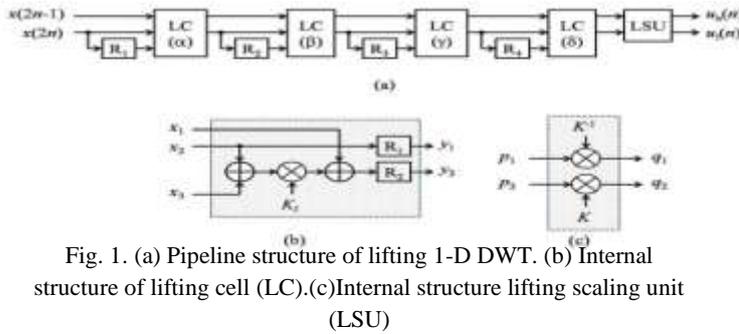


Fig. 1. (a) Pipeline structure of lifting 1-D DWT. (b) Internal structure of lifting cell (LC). (c) Internal structure lifting scaling unit (LSU)

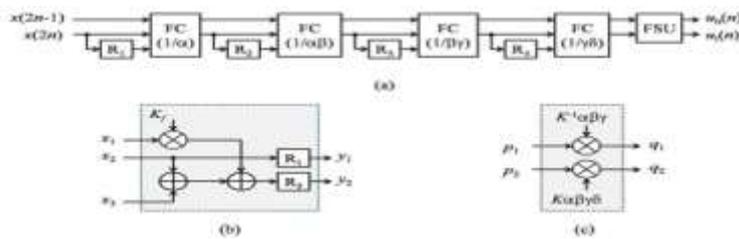


Fig. 2. (a) Pipeline structure of flipping 1-D DWT. (b) Internal structure of flipping cell (FC).

The proposed pipeline structure for the calculation of 1-level flipping DWT is appeared in Fig.2(a). The inner structure of each flipping cell (FC) is appeared in Fig.2(b) and the flipping scaling unit (FSU) is appeared in Fig.2(b). The proposed flipping DWT structure includes 6 multipliers, 8 adders and 12 registers. It ascertains two DWT segments in each clock cycle. The span of cycle period  $T = T_M + T_A$ . The lifting and flipping based structures of 1-D DWT have a similar equipment unpredictability, however the flipping-based structure has less CPD than the lifting-based structure

### B. Integrated Scaling for Lifting and Flipping 2-D DWT

Utilizing distinct approach, row-wise and column-wise computation of 1-D DWT are performed on the input data matrix to acquire four coefficients 1 matrices of 2-D DWT in particular low-low (A), low-high (B), high-low (C), and high-high (D). The scale normalization of row-wise and column-wise lifting 2-D DWT can be integrated and performed toward the finish of the stage-2 utilizing the accompanying condition

$$\begin{aligned} a(m, n) &= K^2 a' (m, n) & (3a) \\ b(m, n) &= b' (m, n) & (3b) \\ c(m, n) &= c' (m, n) & (3c) \\ d(m, n) &= K^2 d' (m, n) & (3d) \end{aligned}$$

Similarly, the scale normalization of row-wise and column-wise computation of flipping 2-D DWT are integrated and performed at the end of column-wise DWT computation using the following equation

$$\begin{aligned} a(m, n) &= K^2 \alpha^2 \beta^2 \gamma^2 \delta^2 a' (m, n) & (4a) \\ b(m, n) &= \alpha^2 \beta^2 \gamma^2 \delta^2 b' (m, n) & (4b) \\ c(m, n) &= \alpha^2 \beta^2 \gamma^2 \delta^2 c' (m, n) & (4c) \\ d(m, n) &= K^2 \alpha^2 \beta^2 \gamma^2 d' (m, n) & (4d) \end{aligned}$$

As appeared in (1)- (2) the lifting and flipping computations of 2-D DWT are with the identical except of the integrated scaling operation given in (3) and (4). Out of four sub-band parts, two sub-band segments [low-low {  $a(m, n)$  } and high-high {  $b(m, n)$  }] just need scaling operation in lifting 2-D DWT whereas in case of flipping 2-D DWT all the four sub-band segments should be scaled. Consequently, the scaling unit of flipping 2-D DWT includes two more multipliers over the lifting 2-D DWT structure with a specific end goal to process all the four sub-band parts in parallel. To show this we have considered the square based lifting 2-D DWT structure of [6] for piece estimate  $p = 4$ . A flipping-based 2-D DWT structure is acquired from the lifting 2-D DWT structure by supplanting all the LCs with FC and the scaling unit by FSU (appeared in Fig2(c)). As appeared in Fig.3(a) and Fig.3(b), the SU of lifting 2-D DWT structure includes 2 multipliers for square size 4 while the SU of flipping 2-D DWT structure includes 4 multipliers. By and large, a piece based lifting 2-D DWT structure includes  $p/2$  less multipliers than those of flipping-based structure for block-size  $p$ . This offers a significant saving of area when the structure is implemented for large block size.

### III. CRITICAL-PATH ESTIMATE OF LIFTING AND FLIPPING CELLS FOR THE COMPUTATION OF 2-D DWT

We have considered radix-4 Booth algorithm for the multiplications. The Booth-multiplier has three units, (i) Booth encoder (BE), (ii) partial product generator (PPG) and (iii) adder unit (AU). The AU uses carry-save data-path to compress the partial products. The delay of AU is  $T_{AU} = T_A + (q - 2)T_{FA}$ , where  $q = w/2$  and  $T_{FA}$  is the FA delay and  $T_A$  is the delay of  $2w$ -bit final ripple carry adder (RCA). The data-path of flipping cell (using radix-4 Booth multiplier is shown in Fig.4(a)). The critical-path delay of FC is  $T_{FC} = T_{BE} + T_P + T_A + (q - 1)T_{FA}$ , where  $T_{BE}$ ,  $T_P$  are, respectively the delay of BE and PPG units. The data-path of lifting cell using radix-4 Booth multiplier is shown in Fig.4(b). In order to perform an addition operation prior to the multiplication operation, we have followed the

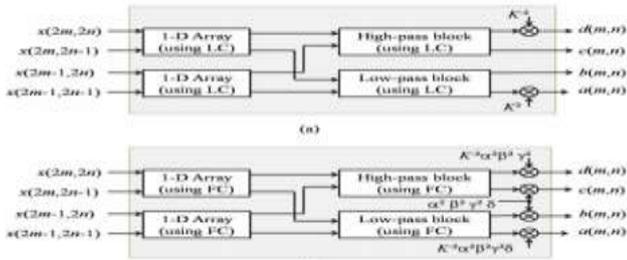


Fig. 3. (a) Parallel structure of lifting 2-D DWT for block-size 4. (b) Parallel structure of flipping 2-D DWT for block-size TABLE I

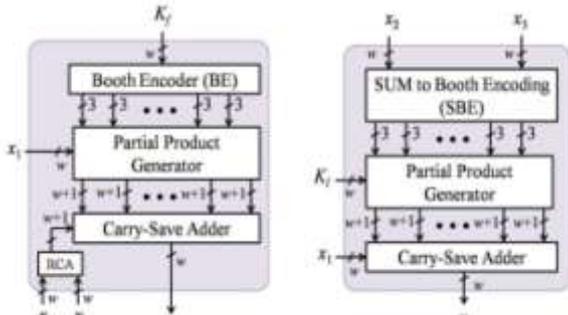


Fig. 4. (a) Existing Data-path of flipping-cell using radix-4 Booth multiplier. (b) Existing Data-path of lifting-cell using radix-4 Booth multiplier well as the existing lifting and flipping 1-D DWT structures for comparison since they have the same hardware and the same time complexity.

**IV. PROPOSED ACHITECTURE FLIPPING PLAN:**

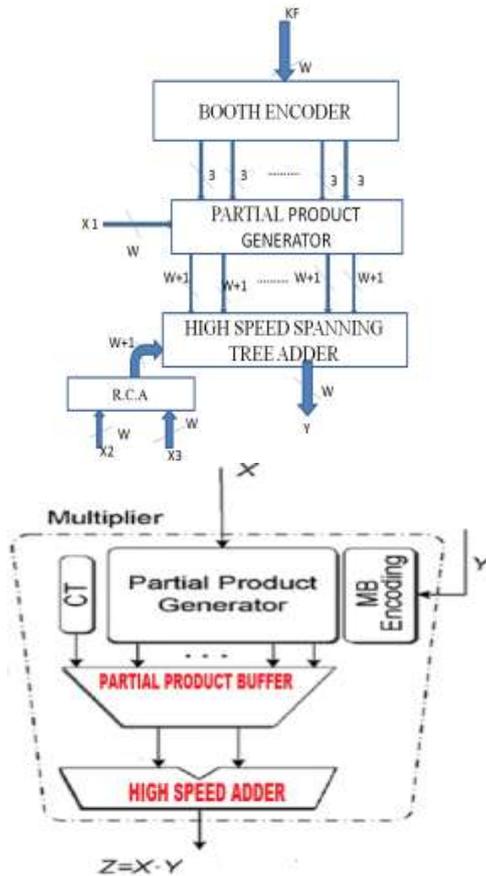


Fig. 5: Internal block diagram of Modified B.E

**V. PROPOSED ARCHITECTURE LIFTING PLAN:**

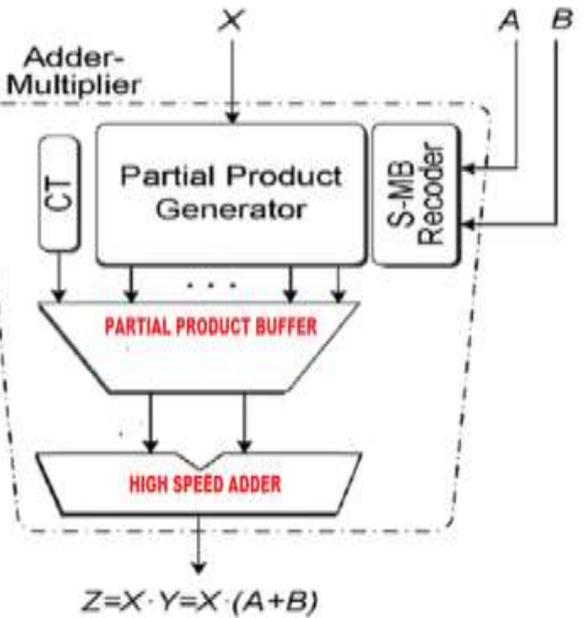
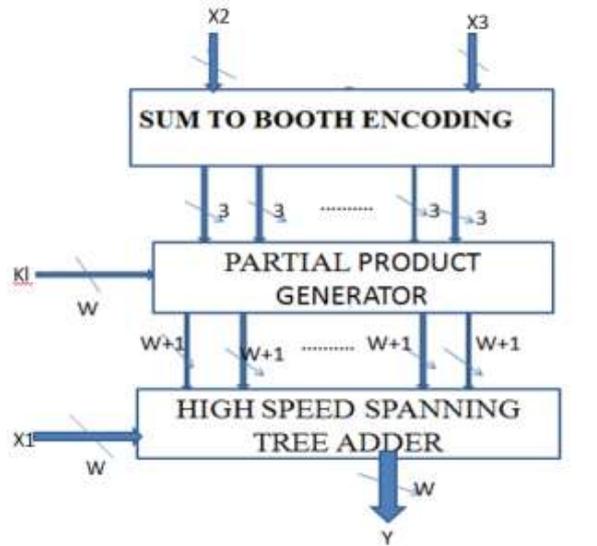


Fig 6 : Internal block diagram of Modified S.M.B.E

Proposed flipping-based and lifting-based structures have the same adder and memory complexity for a given block-size; however they have different multiplier and register complexity. The proposed flipping-based 2-D DWT structure includes 2 less CPD than the proposed lifting-based structure, yet it includes/2 a more number of multipliers and registers than the lifting-based structure for a block size. Since, 2 is small compared with , and the multiplier and register saving in lifting-based structure is significant, ADP of lifting structure is expected to be less than the flipping-based structure for higher block-sizes .The structure of [5] is the best amongst the existing flipping based structures while the structure of [6] is the best amongst.

**High Speed Spanning Adder:-**

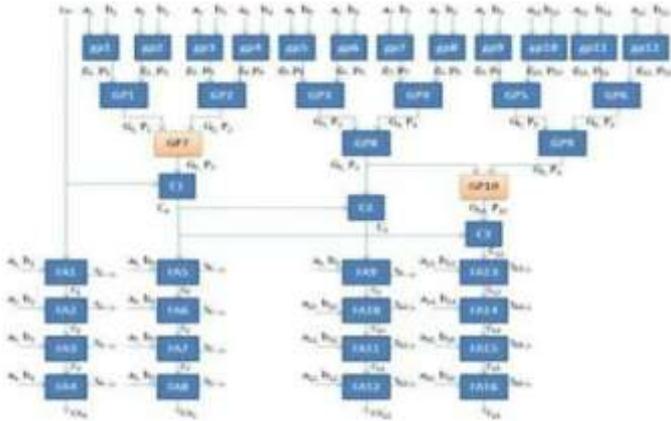


Fig: 7: High n Speed Spanning tree

These high performance adders are essential since the speed of the digital processor depends heavily on the speed of the adders used in the system. Also, it serves as a building block for synthesis of all other arithmetic operations. Adders are most commonly used in various electronic applications e.g. Digital signal processing in which adders are used to perform various algorithms like FIR, IIR etc. In past, the major challenge for VLSI designer is to reduce area of chip by using efficient optimization techniques. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today’s microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors. The redundancy associated with signed-digit numbers offers the possibility of carry free addition. The redundancy provided in signed-digit representation allows for fast addition and subtraction because the sum or difference digit is a function of only the digits in two adjacent digit positions of the operands for a radix greater than 2, and 3 adjacent digit positions for a radix of 2. Thus, the add time for two redundant signed-digit numbers is a constant independent of the word length of the operands, which is the key to high speed computation. Quaternary half adders are designed using binary logic gates and radix converters. This design is appropriate to be applied for construction of a high performance adder like spanning tree which consists of many full adder components and this can be reduced by using quaternary full adders in place of normal full adders.

**VII. RESULTS**

S.No		B.E	P.BE	SMB	P SMB
1.	Gate count	1113	1185	<b>1236</b>	1314
2.	Total no. of LUTs	180	192	202	215
3.	Critical Delay(ns)	31.524	30.142	32.877	31.495
4.	Estimated power	595	575	1145	1128

This project presents a flipping plan using booth encoder which consists of high speed spanning tree adder in the adder blocks. It is simulated by using Xilinx ISE tool and results are verified by using Modelsim SE 6.3f . Experimental results showed that P.D.P. is reduced in the proposed method compare to existing method

Fig-8: Comparison of BE AND MODIFIED BE LUTS,CRITICALDELAY,GATECOUNT,POWER

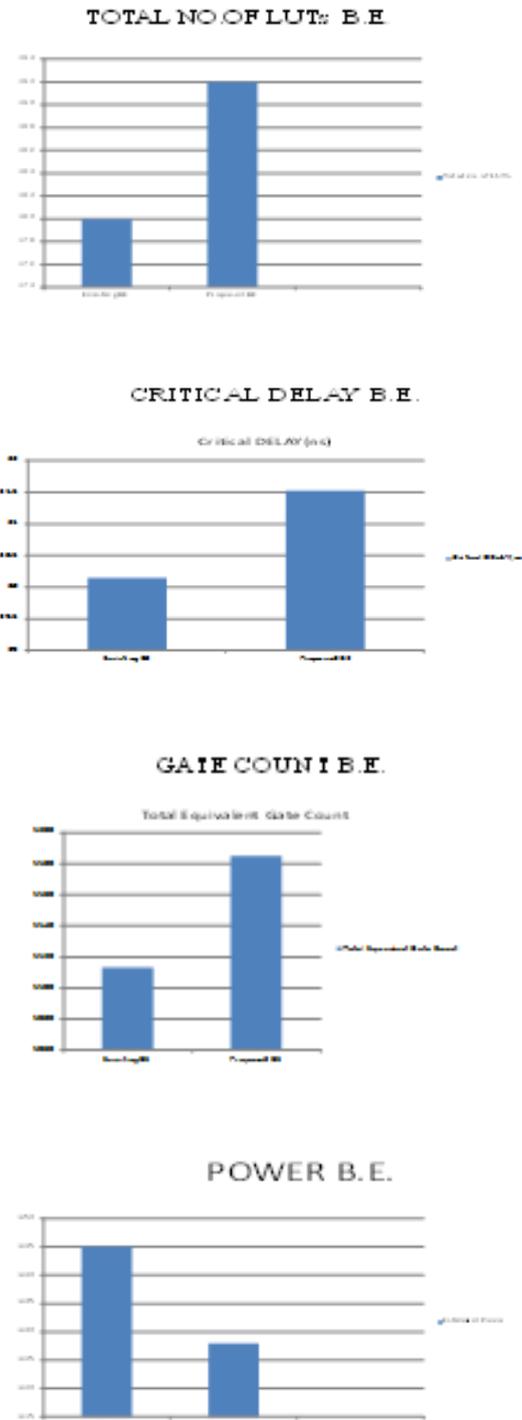
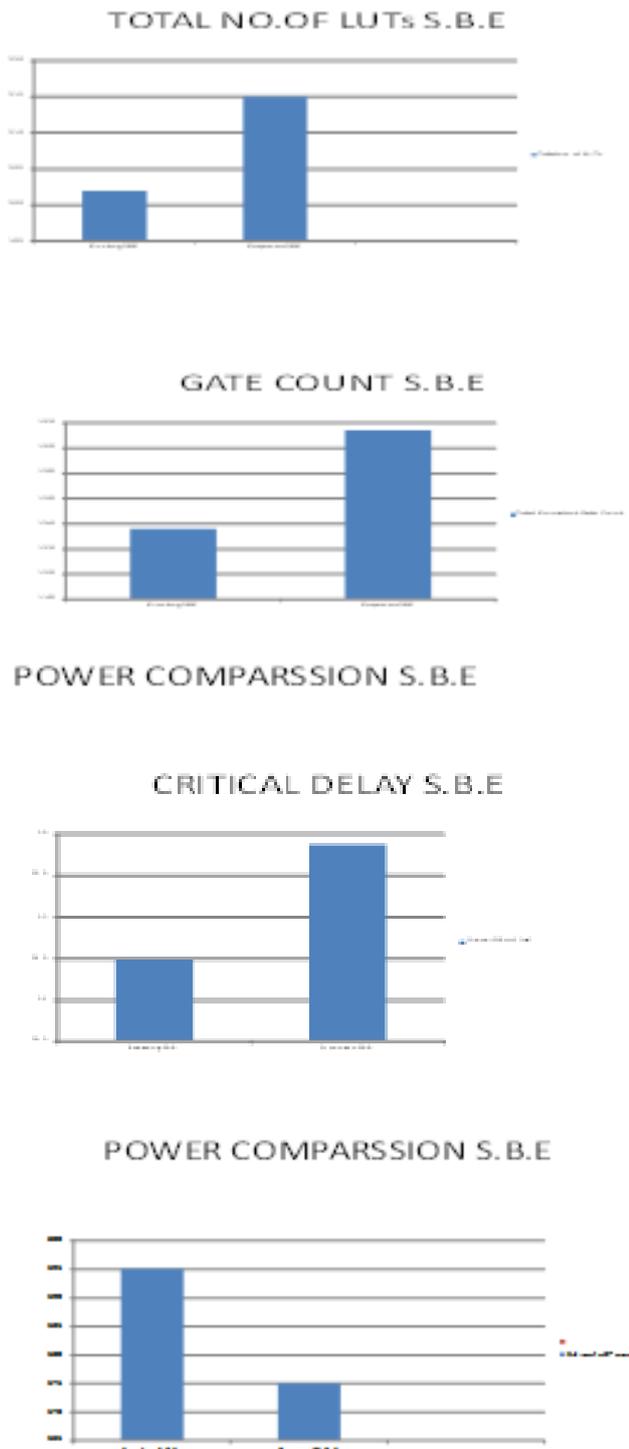


Fig-9: Comparison of SMBE AND MODIFIED SMBE LUTS,CRITICALDELAY,GATECOUNT,POWER



This project presents a Lifting plan using sum to booth encoder which consists of high speed spanning tree adder in the adder blocks. It is simulated by using Xilinx ISE tool and results are verified by using Modelsim SE 6.3f . Experimental results showed that P.D.P. is reduced in the proposed method compare to existing method.

### VII. Conclusion

This project presents a flipping plan using booth encoder which consists of high speed spanning tree adder in the adder blocks. It is simulated by using Xilinx ISE tool and results are verified by using Modelsim SE 6.3f . Experimental results showed that P.D.P is 5% reduced in the proposed method compare to existing method.

This project presents a Lifting plan using sum to booth encoder which consists of high speed spanning tree adder in the adder blocks. It is simulated by using Xilinx ISE tool and results are verified by using Modelsim SE 6.3f . Experimental results showed that P.D.P is 7%reduced in the proposed method compare to existing method.

### REFERENCES

- [1] Daubechies and W. Sweldens, Factoring wavelet change into lifting steps, *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 245– 267, Mar. 1998.
- [2] C.- T. Huang, P.- C. Tseng, and L.- G. Chen, Flipping structure: An effective VLSI engineering for lifting-based discrete wavelet change, *IEEE Trans. Flag Process.*, vol. 52, no. 4, pp. 10801089, Apr. 2004.
- [3] G. Shi, W. Liu, and L. Zhang, An effective collapsed design for lifting based discrete wavelet change, *IEEE Transaction on Circuit and System-II, Express Brief*, vol. 56, no. 4, pp. 290– 294, Apr. 2009.
- [4] Y.- K. Lai, L.- F.Chen and Y.- C.Shih, "An elite and memory-effective VLSI engineering with parallel examining technique for 2-D lifting-based discrete wavelet change," *IEEE Trans. on Consumer Electronics*, vol. 55, no. 2, pp. 400-407, May. 2009.
- [5] W. Zhang, Z. Jiang, Z. Gao, and Y. Liu, "An effective VLSI design for lifting-based discrete wavelet change", *IEEE Transaction on Circuit and System-II, Express Brief*, vol.59, no.3, pp.158– 162, Mar. 2012.
- [6] B. K. Mohanty, A. Mohajan, and P. K. Meher, "Region control effective high-throughput usage of lifting 2-D DWT", *IEEE Transaction on Circuit and System-II, Express Brief*, vol.59, no.7, pp.434– 438, July 2012.
- [7] A. Darji, S. Agrawal, A. Oza, V. Sinha, A. Verma, S. N. Shipper, and A.N. Chandorkar, "Dua-examine parallle flipping engineering for a lifting-based 2-D discrete wavelet change", *IEEE Transaction on Circuit and System-II, Express Brief*, vol.61, no.6, pp.433– 437, June. 2014.
- [8] K. Tsoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos, and K.Pekmestzi, "An advanced modified Booth recorder for proficient outline of the include increase administrator", *IEEE Transaction on Circuit and System-I: Regular Papers*, vol.61, no.4, pp.1133– 1143, Apr. 2014.
- [9] Critical-Path Optimization for Efficient Hardware Realization of Lifting and Flipping DWTs 978-1-4799-8391-9/15/\$31.00©2015 IEEE