Jestr

Journal of Engineering Science and Technology Review 10 (2) (2017) 108- 113

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Design of Efficient Signed Multiplier Using Compressors for FFT Architecture

Marimuthu R* and P.S Mallick

School of Electrical Engineering, VIT University, Vellore, India

Received 16 December 2016; Accepted 12 May 2017

Abstract

This paper presents design of signed multiplier using various compressors. We have designed 4-3 and 5-3 compressors with two and one signed bit for signed multipliers. This signed multiplier provides low power dissipation and high-speed than a conventional signed multiplier. In addition, we have designed Radix -2 four-point FFT structure using the proposed signed multipliers. We have used 5-3 multicolumn compressor to combine adders and subtractors in the four-point FFT structure. It gives better performance in terms of speed and power. Additionally, pipeline concept has been incorporated in the four-point butterfly structure, which further decrease delay and power. This design was implemented using Cadence RTL compiler with TSMC 90nm technology.

Keywords: Signed Multiplier, FFT, Compressors, Butterfly, Pipeline

1. Introduction

Digital signal processing is one of the frequently used techniques for audio and video applications. Many techniques are available in the digital signal processing domain to analyze the video or audio signals. Discrete Fourier Transform (DFT) is widely used algorithm in digital signal processing applications such as linear filtering, convolution, spectrum analysis and correlation [1]. DFT is used to specify the relationship between a time-domain signal and its frequency-domain representation. Direct computation of DFT is inefficient because it does not make use of symmetry and periodicity properties of a twiddle factor. Cooley-Tukey proposed a new algorithm [2, 3] called Fast Fourier Transform (FFT), which is faster than DFT. An FFT algorithm can be applied when the number of samples in the signal is a power of two. The FFT computation takes $(N/2) \times \log2\ N$ multiplications and $N \times \log2\ N$ additions.

When comparing to DFT, FFT takes fewer numbers of computations. Two different forms are available in FFT. They are, Decimation In Time (DIT) and Decimation In Frequency (DIF). Both DIT and DIF use the butterfly structure to compute FFT [4]. The main goal of this paper is to design a high speed and power efficient four-point butterfly structure from the Radix-2 FFT with Decimation in Time (DIT).

Butterfly structure has three major components. They are 1. Signed multipliers 2. Adders 3.Subtractors. Optimizing multiplier performance is challenging task. Several techniques have been used to obtain low power like bypassing techniques and various types of new compressors are introduced in the partial product reduction stage [5,6,7,8,9,10,11,12,13,14,15,16,17,18]. In addition, we have used 5-3 compressors to combine adder /subtractors in the FFT. Different types of compressors like 3-2, 4-2, 4-3, 5-2, 5-3, 7-2, 8-4 and 9-4 have been used in multiplier and FFT s structure. It offers better performance than non-existing compressor structure. Further, we have incorporated pipelining into the modified FFT architecture in order to optimize the performance. The main advantage of the pipeline is that it increases the speed of the operation by utilizing the processor's maximum hardware [19, 20]. Xilinx ISE software is used to verify the functionality of the architecture, and the Cadence RTL compiler is used for logical synthesis of butterfly structure.

This paper is organized as follows. Methodology is described in Section 2. In section 3, we discussed about the results of multiplier and four point butterfly. Conclusion is presented in section 4.

2. Methodology

Various popular signed multipliers are available to multiply both signed and unsigned numbers. They are booth, Pezaris and Baugh – wooley multipliers. We have chosen Pezaris array multiplier because of its simplicity and regular format array.

2.1 Compressor Design

Existing compressors are not suitable for the signed multiplier because it cannot handle signed bit. We have designed various new compressors like 3-2, 4-3 and 5-3 which can handle both signed and unsigned bits.

2.1.1 3-2 Compressor with one or two signed bits

This proposed 3-2 compressor is similar to a full adder, wherein it takes three bits and gives out two bits, but the only difference being it deals with negative bits. In this compressor, there can either be one or two negative bits, and accordingly we get the result. Depending on the number of the negative bits the inputs are arranged, and since the compressor deals with negative numbers one of the output

*E-mail address: rmarimuthu@vit.ac.in

bits is also a negative bit. If there are two negative bits, input 'a' and 'b' are the negative bits and 'C_0' is the negative output bit while if there is only one negative bit then 'c' takes the negative input and 's' gives the negative output. When 'a' and 'b' are positive bits and are high bits, the resulting output is '10' where 'C_0' is positive and's' is negative. Similarly, when 'a' and 'b' bits are high but both of them are negative numbers, the resulting output is 10 but here 'C_0' is negative and's' is positive. Logic expressions of this compressor are given in Eq. 1 and Eq. 2.

$$S = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c \quad (1)$$

$$C_0 = \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c \quad (2)$$

2.1.2 4-3 Compressor with two signed bits

Fig. 1 shows the proposed 4-3 compressor with two signed bits. This compressor has four inputs a, b, c and d and three output S, C_0 and C_1, out of which 'a' and 'b' are signed inputs and 'C_0' is signed output. Boolean equations of this compressor are given in Eq. 3-5. This function is implemented with the help of MUX. Multiplexer based design offers high speed and low power than existing design. The proposed design has a critical path of one XOR gate and one 4:1 MUX.

$$S = (c \oplus d) \cdot \overline{(a \oplus b)} + \overline{(c \oplus d)} \cdot (a \oplus b) \quad (3)$$

$$C_0 = \bar{a} \cdot \bar{b} \cdot (c \cdot d) + a \cdot \bar{b} \cdot \overline{(c + d)} + b \cdot \bar{a} \cdot \overline{(c + d)} + a \cdot b \cdot \overline{(c \bullet d)} \quad (4)$$

$$C_1 = c \cdot d \cdot \overline{(a + b)} \quad (5)$$

The maximum possible outputs in this design is, when all the unsigned inputs are high and the signed bits are low, the output produced is 110, which is basically +2 (as C_0 bit is signed). The minimum output is produced when both the signed bits are high and unsigned bits are low, output is 010, this represents -2.

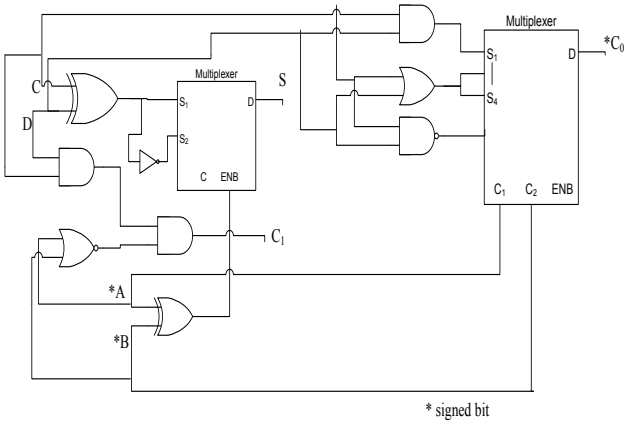

Fig. 1. Implementation of 4-3 compressor with two sign bits

2.1.3 4-3 Compressor with one signed bit

This compressor has four inputs a, b, c, and d with bit 'a' as signed input and three outputs S, C_0 and C_1 with 'S' as signed output bit. During the compression stage of our multiplier architecture, we require different kinds of compressors depending on the number of inputs and number of signed bits. Even though we have the same number of input bits, the change of compressor properties depends on the number of signed bit. As shown in the Fig. 1, both the 4-3 compressors are different, and we need to choose the correct compressor. The output is +3 when all unsigned bits are high and signed bit are low and the output is -1 when the signed bit is high and unsigned bits are low. Boolean expression of this compressor is given in Eq. 6 – 8. Fig. 2 shows the implementation of 4-3 compressor with one sign bit.

$$S = (c \oplus d) \cdot \overline{(a \oplus b)} + \overline{(c \oplus d)} \cdot (a \oplus b) \quad (6)$$

$$C_0 = \bar{a} \cdot \bar{b} \cdot (c + d) + a \cdot \bar{b} \cdot \overline{(c \bullet d)} + b \cdot \bar{a} \cdot (c \cdot d) + a \cdot b \cdot \overline{(c + d)} \quad (7)$$

$$C_1 = c \cdot d \cdot b \cdot \bar{a} \quad (8)$$

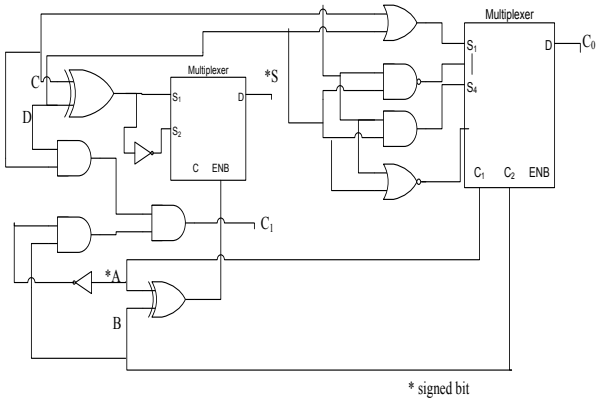

Fig. 2. Implementation of 4-3 compressor with one sign bit

2.1.4 5-3 Compressor with two signed bits

The design contains five inputs a, b, c, d and e and three output s, C_0 and C_1. The inputs 'a', 'b' are signed inputs bits and 'C_0' is the signed output bit. The critical path for this design is two XOR and one 2:1 MUX. Logical equations of this compressor are given in Eq. 9-11. Fig. 3 shows the implementation of 5-3 compressor with two signed bits.

$$S = \overline{(a \oplus b \oplus c)} \cdot (d \oplus e) + (a \oplus b \oplus c) \bullet \overline{(d \oplus e)} \quad (9)$$

$$C_1 = \bar{a} \cdot \bar{b} ((c \cdot d) + (c \cdot e) + (d \cdot e)) + ((\bar{a} \cdot b) + (a \cdot \bar{b})) (c \cdot d \cdot e) \quad (10)$$

$$C_0 = \bar{a} \cdot \bar{b} ((c \cdot d) + (c \cdot e) + (d \cdot e)) + ((\bar{a} \cdot b) + (a \cdot \bar{b})) ((c \cdot d \cdot e) + \overline{(c + d + e)}) + a \cdot b \overline{((c \bullet d) + (c \bullet e) + (d \bullet e))} \quad (11)$$

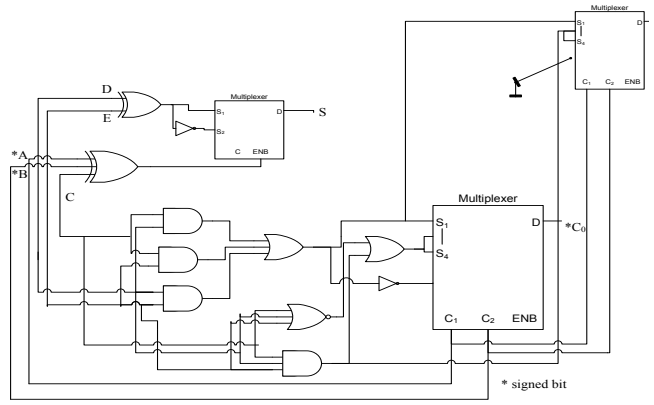

Fig. 3. Implementation of 5-3 compressor with two sign bits

2.1.5 5-3 Compressor with one signed bit

The design is based on the previous compressor design. The main differences are the number of signed bits, and change in output signed bits. This compressor consists of five input bits a, b, c, d and e where 'a' is the signed bit, and three output bits are S, C_0, C_1 where 'S' is the signed output bit. Boolean expression of this compressor is given in Eq. 12-14.

$$S = \overline{(a \oplus b \oplus c)} \cdot (d \oplus e) + (a \oplus b \oplus c) \cdot \overline{(d \oplus e)} \quad (12)$$

$$C_1 = (\bar{a} \cdot b) \cdot ((c \cdot d) + (c \cdot e) + (d \cdot e)) + (\overline{(\bar{a} \cdot \bar{b})} + (a \cdot \bar{b})) (c \cdot d \cdot e) \quad (13)$$

$$C_0 = (\bar{a} \cdot b) \cdot ((c \cdot d) + (c \cdot e) + (d \cdot e)) + (\overline{(\bar{a} \cdot \bar{b})} + (a \cdot b)) ((c \cdot d \cdot e) + \overline{(c + d + e)}) + (a \cdot \bar{b}) \cdot \overline{((c \cdot d) + (c \cdot e) + (d \cdot e))} \quad (14)$$

2.2 Multiplier Design Using Compressors

In FFT structure, all multipliers are designed by using various signed and unsigned compressors. Apart from the above signed compressor, we have also used some conventional unsigned compressors in the first "n" columns of partial product reduction. Two signed bits are available from $n+1^{th}$ column onwards because the most significant bit of multiplicand produces one signed bit and most significant bit of multiplier produces another signed bit. It is necessary to include signed compressor from $n+1^{th}$ column to add signed bit. In every column of the multiplier, we have chosen correct combination of compressors to reduce the partial product. Signed compressors are used in the multiplier whenever is required. Fig. 4 shows the dot implementation of Pezaris array multiplier. Results are shown in Tab. 1. Compressor based multipliers provide better performance than a conventional signed multiplier in terms of speed, and energy delay product (EDP).

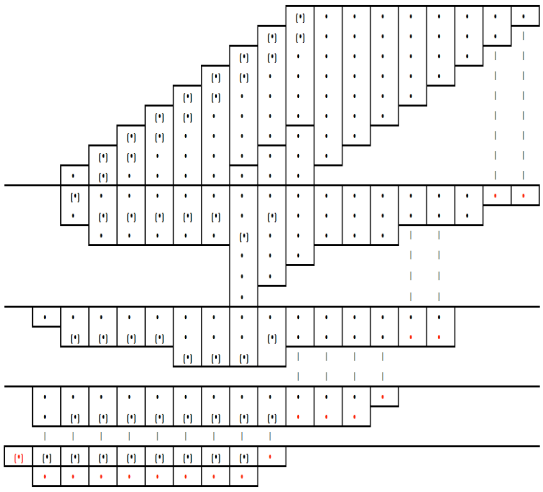

Fig. 4. Dot implementation of Pezaris array multiplier

2.3 Modified Four-Point Butterfly (Radix -2) Structure

Conventional Radix-2 four-point butterfly structure requires 46 multipliers to compute FFT. In order to reduce the number of multipliers, conventional structure is modified, and it has only 32 multipliers, which results high speed. By looking at the conventional equation, we can understand that outputs E and G, F and H have the common terms with few sign changes. Since input can be signed and unsigned magnitude integers, it is required to use a signed multiplier. Pezaris signed multiplier is used in FFT [21]. Advantages of using this multiplier are, (i) correction factor is not required. (ii) Relatively simpler than booth multiplier (iii) Recoding is not required in the partial product reduction stage. This multiplier is further modified by arranging adders into three sections [22]. This multiplier is called tri section Pezaris array multiplier. This modified four-point butterfly structure is not significantly faster, but consumes less power than the conventional structure. So in order to increase the speed, further modification of a four-point butterfly is required. To get optimized structure, we have used proposed signed multipliers, and replaced some of the adders and subtractors with 5-3 multicolumn compressors. Fig. 5(a) & (b) show the modified structure of a four-point butterfly.

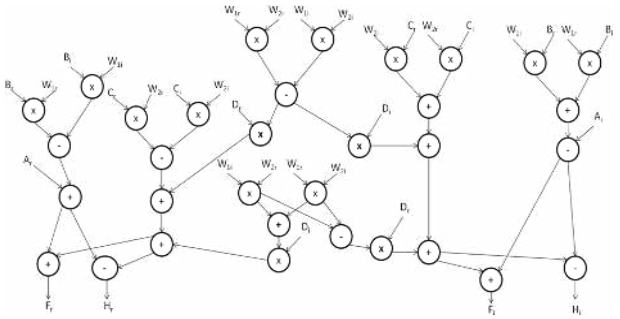

Fig. 5. (a) Modified four-point butterfly structure (F and H)

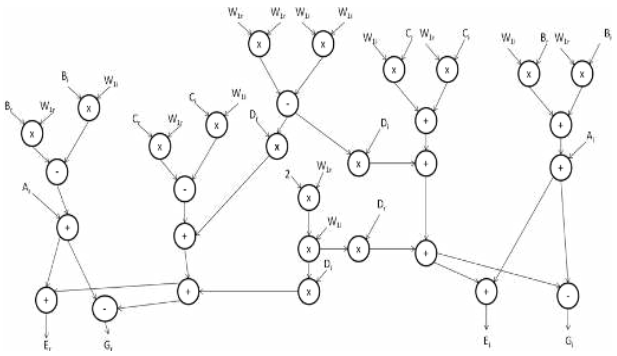

Fig. 5. (b) Modified four-point butterfly structure (E and G)

2.4 Butterfly Structure Using Compressors with Pipeline

Adders and subtractors used in four-point butterfly structure are replaced by 5-3 multicolumn compressors. A detail discussion about 5-3 compressor is given by Ohsang K won et. al [23]. We have used 5-3 multicolumn compressor in the butterfly structure. For subtraction, 2's complement is done and result is directly added with the help of 5-3 compressors. Ten adders and six subtractors are replaced by eight 5-3 compressors, which results less power consumption and high speed. This four-point butterfly structure is 13.64% faster and consumes 32.39% less power than the four – point butterfly structure by using conventional multiplier and adder. To get further improvement in terms of delay and power, three -stage pipeline are used in butterfly structure [24]. Fig. 6 (a) & (b) show the architectures of output's E, G and F, H using 5-3 multicolumn compressors with pipelining.

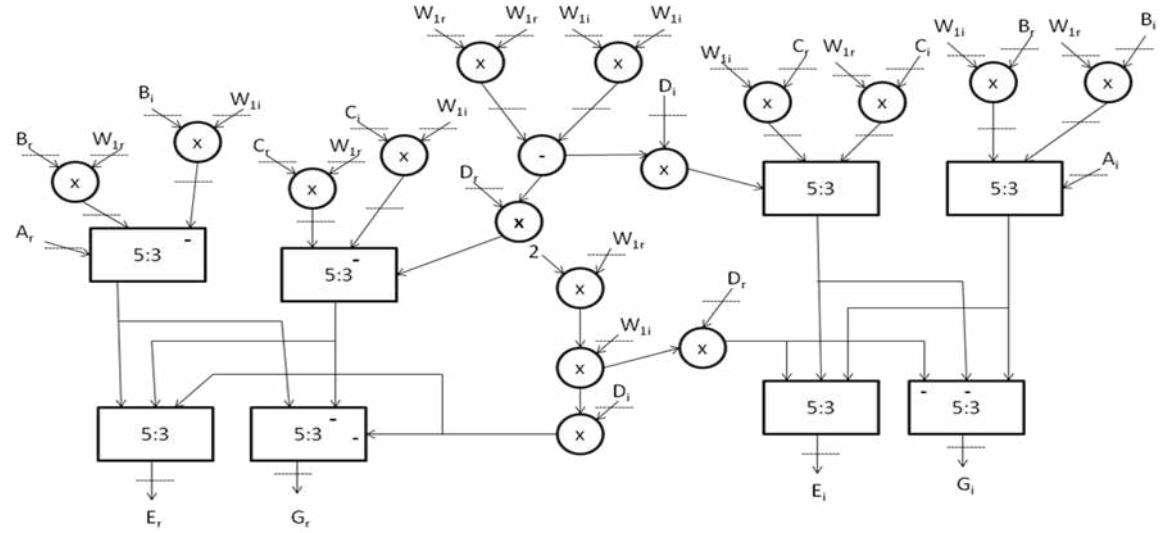

Fig. 6. (a) Butterfly structure using 5 – 3 multicolumn compressor with pipeline (E and G)

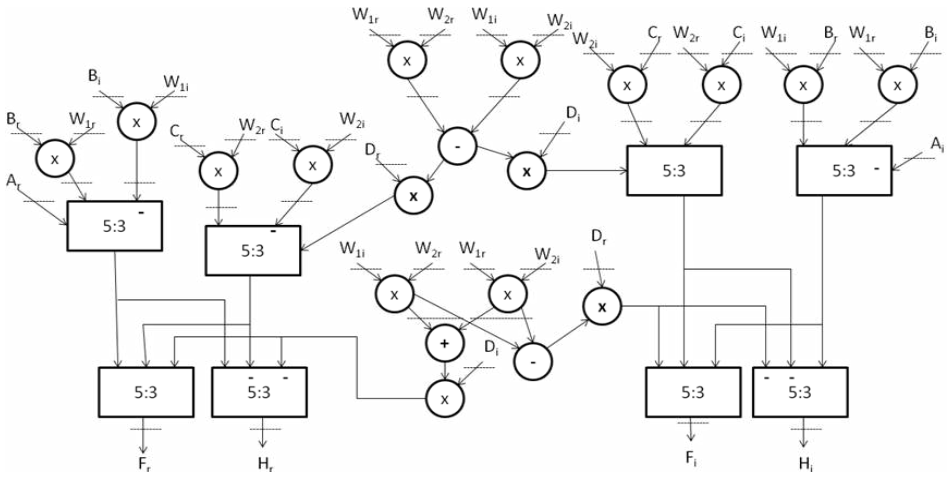

Fig. 6. (b) Butterfly structure using 5 – 3 multicolumn compressor with pipeline (F and H)

3. Results and Discussion

All butterfly structures are described in Verilog HDL and functionality of those structures are verified using Xilinx ISE 12.4. Architectures were synthesized by using the Cadence RTL compiler, and the results are obtained with the help of 90nm technology library. Random test vectors are used to compute the power consumption. Here we have included total power (Static + Dynamic) of the architecture.

3.1 Multiplier

Tab. 1 shows the performance of Pezaris multiplier. In 4 × 4, multiplier using proposed compressor provides 17.33 % power reduction than conventional multiplier. But conventional multiplier is faster than proposed multiplier by 1.32 %. 8 × 8

multiplier using proposed compressor provides 22.6 % reduction in power and 2.27 % faster than conventional multiplier. Multiplier designed using compressor has area overhead. We have involved multiplexers in the proposed compressors. This leads to occupy more are than conventional multiplier. Total number of adders and compressors are used in multipliers are listed in Tab. 1.

Table 1 Performance of Signed Multiplier

Multiplier	Type	Power (mW)	Delay (ns)	EDP (Ws^2 x 10^{-24})	Number of Full Adders (FA's)	Number of different types of Compressors
4 × 4	Conventional [22]	18.00	1.49	40.14	12	0
	Proposed	14.88	1.51	33.93	4	10
8 × 8	Conventional [22]	87.49	2.64	610.76	56	0
	Proposed	67.70	2.58	453.46	26	32

3.2 Four-Point Butterfly Structure

Tab. 2 shows the synthesis results of butterfly structures which are designed by using various methods. As expected, butterfly structure designed using conventional multiplier and adder – subtractors (design 1) provides highest delay and high power consumption than other techniques. In second design, conventional multiplier is used in butterfly structure adder – subtractors are replaced by 5-3 multicolumn compressor. This structure provides 2.3 % less delay and 6.4 % power gain than design 1. In third design, multipliers are designed using proposed compressors and conventional adder – subtractors are used in butterfly structure. This structure provides 3.9 % less latency and 8.5 % power improvement than design 1.

In fourth design, compressor based multipliers and 5-3 compressors are used in butterfly structure. Delay of this butterfly structure is 13.64 % better than the design 1 and 11 % better than second design. Similarly, power and energy delay product is also reduced by 32.39 % and 49.5 % than design 1. This design provides overall better results than first three techniques. In fifth design, pipeline technique has been applied to second approach of butterfly structure. Pipeline structure provides 16.3 % reduction in delay and 14.4 % less power than non-pipeline approach. Similarly, energy delay product is also reduced by 36.49 % than design 2. In design 6, pipeline has been applied to fourth design of butterfly structure. This structure provides the best result than all existing approach. This approach provides 17.43 % less latency and 20.96 % than fourth approach. Pipeline approach has limitation of area overhead. Area got increased by 9.2 % than non-pipeline approach.

Table 2 Synthesis Results of four point butterfly structure

Butterfly Design	Delay (ps)	Power (µw)	Area (μm^2)	EDP ($\times 10^{-22}$)
1 [5]	6383	284.46	22,756.30	115.89
2 [23]	6235	266.20	24283.43	103.48
3	6132	260.26	24656.72	97.86
4	5512	192.31	33456.73	58.42
5 [24]	5337	169.06	32514.06	48.15
6	4599	152.00	41442.71	32.14

4. Conclusions

This paper proposes design of signed 4-3 and 5-3 compressors with one and two sign bit. Then signed multiplier is designed with the help of proposed compressor. The increasing demand of signed multiplier needs a better design. Our compressor based design of signed multiplier will be able to play a role towards this increasing demand because it has reduced delay, power consumption and EDP, with an overhead of area. Future research may focus on this. Also, in future one can design high point butterfly structure using proposed compressors.

Acknowledgement

The authors would like to acknowledge the faculty colleague of our university Mr. S Balamurugan for the support.

References

[1] A. S˘al˘agean, A. Alecu, "An Approximation Algorithm for computing the k-error Linear Complexity of Sequences Using the Discrete Fourier Transform," IEEE International Symposium on Information theory, Canada, 6-11 July 2008, pp. 2414-2418, (ISIT 2008).

[2] W.T. Cochran, J.W. Cooley, D.L. Favin, H.D. Helms, R.A. Kaenel, W.W.Lang, G.C Maling, Jr., D.E. Nelson, C.M. Rader, P.D. Welch, "What is the fast Fourier transform?," IEEE Transactions on Audio and Electroacoustics, **15**, pp. 45-55, (1967).

[3] J. Cooley, J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier series", Mathematics of Computations. **19**, pp. 297 – 301, (1965).

[4] S.A. White, "A Simple FFT Butterfly Arithmetic Unit", IEEE Transactions on Circuits and Systems. **28** (4),pp.352-355, (1981).

[5] M.B. Fonseca, A. Eduardo, C. Costa, B. Joa˜o, S. Martins, "Design of power efficient butterflies from Radix-2 DIT FFT using adder compressors with a new XOR gate topology", Analog Integrated Circuit Signal Processing. **73**(3), pp. 945–954, (2012).

[6] S. Balamurugan, S. Ghosh, Atul, R. Balakumaran, R. Marimuthu, P.S Mallick, "Design of low power fixed-width multiplier with row bypassing", IEICE Electronics Express **9**(20), pp.1568-1575, (2012).

[7] R. Marimuthu, E. Y. Rezinold, P.S Mallick, " Design and Analysis of Multiplier Using Approximate 15-4 Compressor", IEEE Access, **5,** pp. 1027-1036, (2017). doi: 10.1109/ACCESS.2016.2636128.

[8] R. Menon, D. Radhakrishnan, "High performance 5: 2 compressor architectures", IEE Proceedings - Circuits, Devices and Systems, **153**(5), pp. 447-452, (2006).

[9] A. Pishvaie, G. Jaberipur, A. Jahanian, "Improved CMOS (4; 2) compressor designs for parallel multipliers, " Computers and Electrical Engineering. **38**(6), pp. 1703–1716, (2012).

[10] R. Marimuthu, S. Balamurugan, P.S Mallick, "Design of 5-3 Multicolumn Compressor for High Performance Multiplier", International journal for computer Aided Engineering and Technology. (2016). Article in Press.

[11] A. Pishvaie, G. Jaberipur, A. Jahanian, "High Performance CMOS (4:2) compressors", International journal of electronics. **101**(11), pp. 1511-1525, (2014).

[12] R. Shenda, P. Zode, P. Zode, "Efficient design 2k−1 binary to residue converter", International Conference on Devices, Circuits and Systems, Coimbatore, India, pp. 482-485, (ICDCS 2012).

[13] A. Pishvaie, G. Jaberipur, A. Jahanian, " Redesigned CMOS (4; 2) compressor for fast binary multipliers", Canadian Journal of Electrical and Computer Engineering. **36**(3), pp.111-115, (2013).

[14] R. Marimuthu, S. Balamurugan, B.K. Tirumala, P.S Mallick, "FPGA implementation of High Speed Multiplier Using Higher Order Compressors", International conference on Radar, Communication and Computing, Tiruvannamalai, India, pp. 210 – 212, (ICRCC 2012).

[15] R. Marimuthu, D. Bansal, S. Balamurugan, P.S Mallick, P.S. "Design of 8 – 4 and 9 – 4 Compressors for High Speed Multiplication", American Journal of Applied Sciences. **10**(8), pp. 893-900, (2013).

[16] M. Ma, S. Li, "A New High Compression Compressor for Large Multiplier", International conference on Solid-State and Integrated-Circuit Technology, Beijing, China, pp. 1877 – 1880, ICSICT 2008.

[17] M Nagamatsu, S. Tanaka, J. Mori, K. Hirano, T.Noguchi, K. Hatanaka, "A 15-ns 32 × 32-b CMOS Multiplier with an Improved Parallel Structure", IEEE Journal of Solid-State Circuits. **25**(2), pp. 494-497, (1990).

[18] K. Prasad, K.K. Parhi K.K, "Low Power 4-2 and 5-2 compressors", Asilomar Conference on Signals, Systems and Computers, Western USA. Pacific Grove, California, pp.129-133, (2001).

[19] M. Potkonjak, J. Rabaey, "Pipelining: Just another Transformation. Proceedings of the international conference on Application Specific Array Processors", Western USA. Berkeley, California, pp. 163 – 175, (1992).

[20] M. Hatamian, L.C Glenn, "High Speed Signal Processing, Pipelining, and VLSI, Acoustics, Speech, and Signal Processing", IEEE International Conference on Acoustics, Speech and Signal Processing, Tokyo, Japan, pp. 1173 – 1176, (ICASSP 1986),

[21] S.D. Pezaris, "40- ns 17- bit by 17- bit array multiplier", IEEE Transaction on computers. **20**(4), pp.442-447, (1971).

[22] J. Stohmann, E. Barke, "A universal Pezaris array multiplier generator for SRAM-based FPGAs", IEEE International Conference on Computer Design, Central USA, Austin, Texas, USA, pp. 489-495, (ICCD 1997).

[23] O. Kwon, K. Nowka, E. Swartzlander, Jr., "A 16-Bit by 16-Bit MAC Design Using Fast 5:3 Compressor Cells", Journal of VLSI Signal Processing. **31**(2), pp.77–89, (2002).

[24] X. Liu, F. Yu, Z.K. Wang, "A pipelined architecture for normal I/O order FFT", Journal of Zhejiang University: Science C. **12**(1), pp.76-82, (2011).