International Journal of Engineering & Technology, 7 (3.4) (2018) 213-217

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



High Speed RADIX-2 Butterfly Structure Using Novel Wallace Multiplier

Garima Thakur[1], Harsh Sohal[2], Shruti Jain[3*]

[1]*Dept of Electronics & Comm. Engg., Jaypee University of Information Technology, Distt. Solan, Himachal Pradesh, India*
[2]*Dept of Electronics & Comm. Engg., Jaypee University of Information Technology, Distt. Solan, Himachal Pradesh, India*
[3]*Dept of Electronics & Comm. Engg., Jaypee University of Information Technology, Distt. Solan, Himachal Pradesh, India*
**Corresponding author E-mail: jain.shruti15@gmail.com*

Abstract

In Signal Processing applications the arithmetic units mainly consists of adders and multipliers. These arithmetic units are used in to enhance the performance of Fast Fourier Transform (FFT) Butterfly structure implementation. This paper discusses the addition and multiplication algorithms for parameters like speed, area and power. The best suited among all adders are Kogge Stone Adder (KSA) while among multipliers are Wallace multiplier(WM) which is used for the implementation of the FFT structure. Verilog coding is used for implementation of circuit and the tool used is Xilinx ISE 14.1 Design suite.

Keywords: *Fast Fourier Transform; FPGA; Kogge Stone Adder; Wallace multiplier; Xilinx ISE 14.1.*

1. Introduction

In Signal Processing applications the most important algorithm used is FFT, which is basically used for frequency analysis of the signal. The FFT core uses the Radix-2 decomposition for computing the DFT. N/2 butterflies are present in each stage of FFT if there is N- point FFT using Radix-2 decomposition. It is important to implement butterfly structure because by improving the structure the performance of the FFT algorithm can be improved which impacts the overall performance of the circuit. Butterfly structure consists of adders and multipliers. Firstly it is important to implement complex addition and multiplication block. There are various types of adders and multipliers. This paper shows the structure using Wallace multiplier (WM) and Kogge Stone Adder (KSA) because WM and KSA is best in terms of speed for computation. Bias K. et al 2016 [1] explained the comparison of various adders in terms of speed and area utilization for different bit streams and in the end concluded that parallel prefix KSA is best in terms of speed. Anjana R. et al 2014 [2] and Rajaram S. et al 2011 [3] used parallel prefix adder for implementation of efficient Wallace multiplier. By using KSA in final stage of multiplier block there is improvement in the speed of the circuit. Increasing the speed of the circuit overall performance of the circuit can be improved. Murugugeswari S. et al 2014 [4] implemented efficient multiplier but by using multiplexer based full adder. Singh A. K. et al 2017 [5] designed an efficient algorithm for FFT. Butterfly unit is implemented in this paper and transform performance is improved because efficient adder and Vedic multiplier is used.

By increasing one parameter of the circuit whole performance of the circuit is affected. Our aim is to design an efficient circuit. Firstly, we designed design an efficient 16 bit and 32 bit KSA and then used this KSA for implementation of 16 bit and 32 bit Wallace multiplier. Using these adders and multipliers we implemented FFT structure using Xilinx software tools.

The paper is organized as follows: In Section 2 and Section 3 efficient adders and multipliers are implemented for 16 bit and 32 bit respectively. Lastly, in Section 4, FFT circuit based on butterfly structure has been synthesized using best adders and multiplier which gives better result than existing implementation in the literature reviewed.

2. Adders

Analysis of the various adder circuits lead to the conclusion that Ripple Carry Adder (RCA) takes more time to propagate the carry and gave the maximum delay as compared to other adder circuits. KSA came out to be the best as it overcame the problems of RCA and other circuits by generating carry using prefix network in parallel [1]. KSA consists of three stages i.e. pre-computation stage, prefix network stage and post-computation stages as shown in Figure 1.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

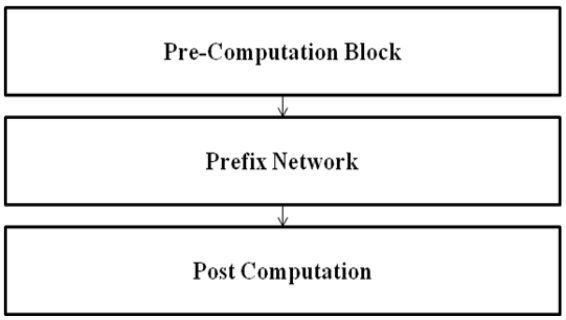


Fig. 1: Block diagram of KSA

In pre-computation stage, there is computation of 'Generate' and 'Propagate' term as expressed in Eq. (1) and Eq. (2):

$$\text{Generate}(G_i) = A_i \text{ AND } B_i \quad (1)$$

$$\text{Propagate}(P_i) = A_i \text{ XOR } B_i \quad (2)$$

In prefix network, the 'Generate' and 'Propagate' term is expressed as Eq. (3) and Eq. (4):

$$\text{Propagate}(P) = P_i \text{ AND } P_{iprev} \quad (3)$$

$$\text{Generate}(G) = (P_i \text{ AND } G_{iprev}) \text{ OR } G_i \quad (4)$$

Finally, in post-computation stage there is computation of final 'Sum' and 'Carry' which is expressed in Eq. (5) and Eq. (6):

$$\text{Sum, } S_i = P_i \text{ XOR } C_{i-1} \quad (5)$$

$$\text{Carry, } C_i = G_i \quad (6)$$

KSA have some drawback that its complexity increases with the number of bits. In this paper 16 bit and 32 bit KSA has been implemented. In higher processing applications we need larger number of bits to solve complex calculation. In Figure 2 and Figure 3 shows the RTL schematic of 16 bit and 32 bit.

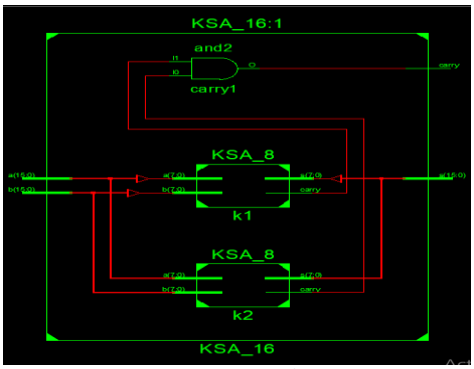


Fig. 2: RTL Schematic of 16 bit KSA

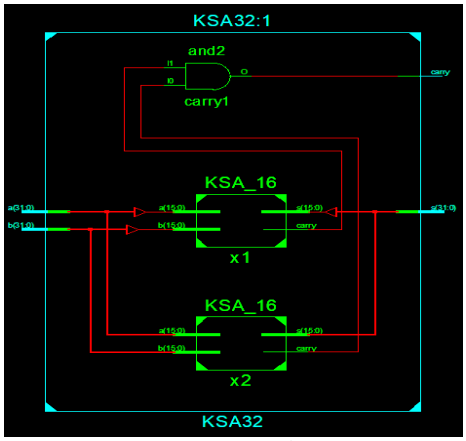


Fig. 3: RTL Schematic of 32 bit KSA

Table 1: Results of KSA

Circuit Designed		KSA 16 bit	KSA 32 bit
No. of bonded IOB	I –Buf	32	64
	O- Buf	17	33
No. of Slice LUT		35	71
Power Total (W)	Leakage	0.081	0.081
	IOs	0.017	0.017
Delay (ns)	Logic	4.406	4.402
	Router	2.853	3.345
Power Delay Product (WnS)		0.7113	0.7592

3. Multiplier

Multiplication plays a crucial role in Digital Signal Processing and for faster FFT we require a high speed multiplication block [6-7]. High speed multiplication block is implemented by using three main steps: first step: generation of partial products, second step: addition of partial product which is done with the help of adder and speed of the circuit is derived from this stage and in the last stage: final result is generated by adding two-row output. Section 2 already described that KSA is best in terms of speed that's why KSA is used for designing of multiplication circuit. WM gives the best result compared to other two (array multiplier and booth multiplier). For implementation of Array multiplier (AM) more number of circuits are required. As increase in the number of components in AM there is an increase in the delay. WM using KSA is implemented to increase the speed of the circuit. In this section 16 bit and 32 bit WM is implemented using KSA.

3.1. 16 Bit Wallace Multiplier Using KSA

Figure 4 shows the architecture for 16 bit WM. In this architecture, a 16 bit input is applied by using four 8×8 WM blocks. The output of 8×8 WM blocks is applied to KSA blocks. Three KSA blocks are used to add output that is coming from different 8×8 WM blocks. Finally 32 bit sum and 1 bit carry is generated. This is an efficient architecture in terms of delay and we can use this architecture for fast calculation in any digital signal processing application.

3.2. 32 Bit Wallace Multiplier Using KSA

Figure 5 shows the architecture for 32 bit WM. In this architecture a 32 bit input is applied by using four 16×16 WM blocks. The output of 16×16 WM blocks is applied to KSA blocks. Three KSA blocks are used to add output that is coming from different blocks of 16×16 WM blocks. Finally the 64 bit sum and 1 bit carry is generated. This is an efficient architecture in terms of delay and we can use this architecture for fast calculation in signal processing applications requiring bit length of 32. Table 2 shows the results for a 16 bit and a 32 bit Wallace multipliers that have been implemented using Xilinx software tools and different evaluation parameters are calculated from the synthesis report in terms of area utilization and delay. Power is also calculated using the Power Analyzer tool.

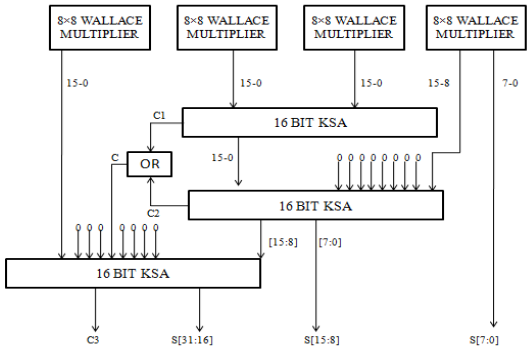


Fig. 4: Wallace multiplier 16 bit

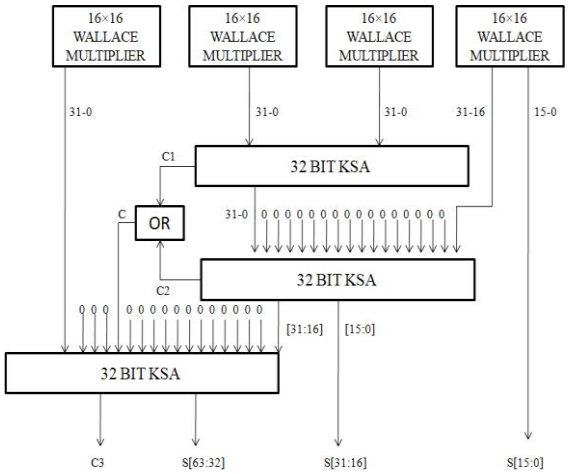


Fig 5: Wallace multiplier 32 bit

4. Transforms

Transforms are helpful for conversion of difficult calculation into easiest way as possible [8-9]. The analysis of the signal is possible in any form either in frequency or time domain. The most important algorithm is used in signal processing including biomedical signal processing and analysis is FFT. The basic component of FFT algorithm is Butterfly structure which plays a crucial role in transforms. In Butterfly, for solving the complex calculation addition and multiplication blocks are used. This paper presents an efficient Butterfly structure implementation with best adder (KSA) and multiplier (WM) blocks discussed in the previous sections.

Table 2: Results of Wallace Multiplier

Circuit Designed		**16 bit WM**	**32 bit WM**
No. of bonded IOB	I –Buf	32	64
	O- Buf	32	64
No. of Slice LUT		610	2109
Power Total (W)	Leakage	0.081	0.113
	IOs	0.017	0.017
Delay (ns)	Logic	6.641	7.455
	Router	14.369	18.134
Power Delay Product (WnS)		2.0589	3.326

Butterfly real and imaginary parts [10] are calculated using DFT equation which is expressed by Eq. (7).

$$X_p = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}np} \tag{7}$$

Eq. (8) is expressed with the help of Eq. (7) by splitting it into even and odd parts:

$$X_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)p} \tag{8}$$

$$= A_p + W^p B_p \tag{9}$$

When frequency is p+N⁄2 then Eq. (7) is expressed by Eq. (10).

$$X_{p+N/2} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}(n)p} - e^{-j\frac{2\pi}{N}p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}np} \tag{10}$$

$$= A_p - W^p B_p \tag{11}$$

Where, A_p = even-numbered sample, B_p = odd-numbered samples and

$$W^p = e^{-j\frac{2\pi}{N}} \tag{12}$$

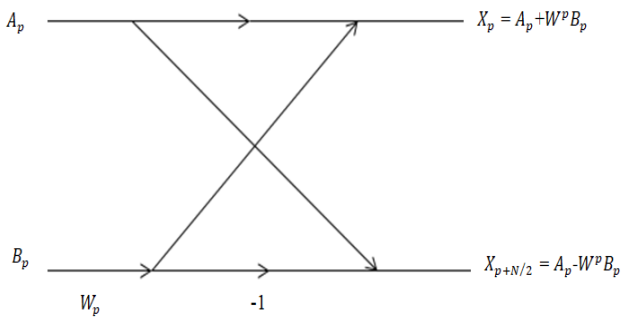


Fig. 6: Butterfly Structure

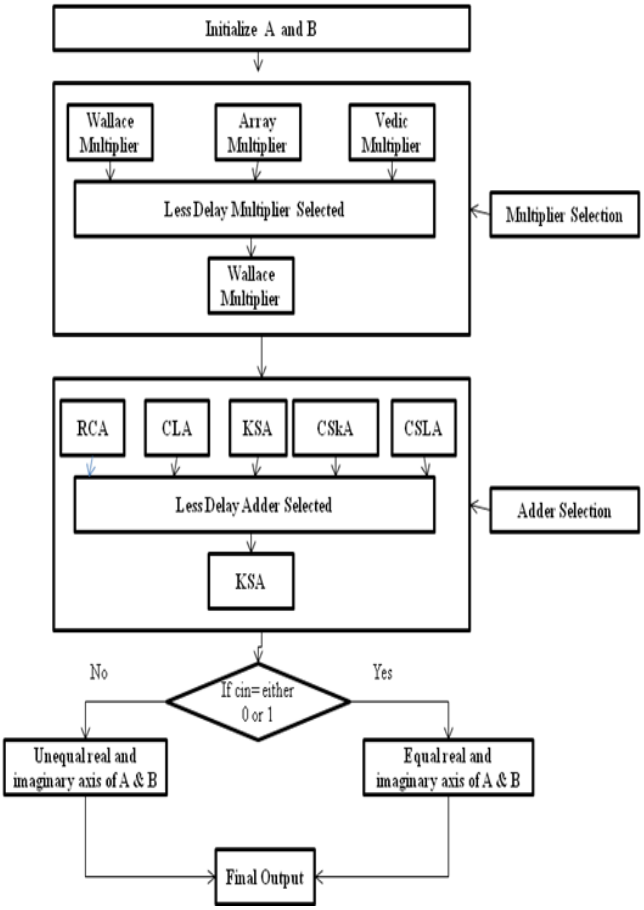


Fig. 7: Flow Diagram of Butterfly

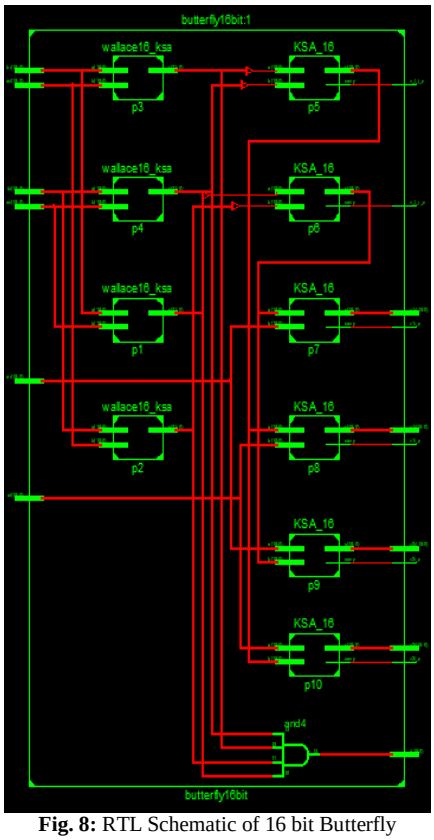


Fig. 8: RTL Schematic of 16 bit Butterfly

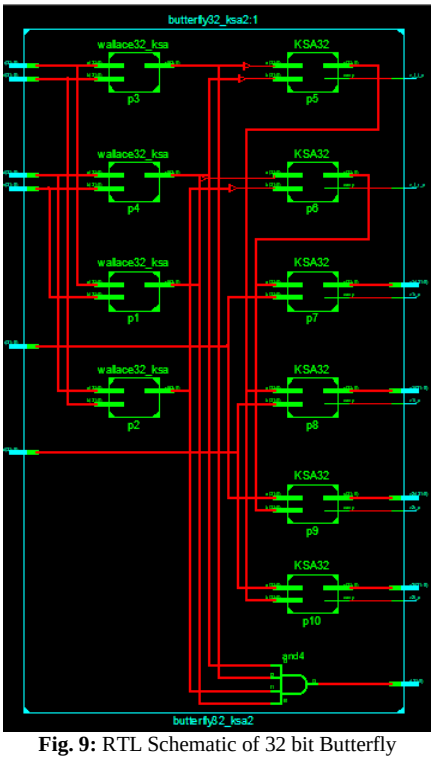


Fig. 9: RTL Schematic of 32 bit Butterfly

Modified Butterfly structure is implemented with the help of efficient adder and multiplier which is already explained in Section 2 and Section 3. Flow diagram of butterfly structure is shown in Figure 7. In the flow diagram, initialize the inputs A and B and then applied the input to multiplier selection block where it selects the efficient multiplier in terms of delay. Multiplier is used for multiplication of two numbers and the output of the multiplication block is applied to adder selection block where it selects the efficient adder and adds the output getting from the multiplier block. Lastly, depending on the value of Cin the real and imaginary axis is calculated. If Cin is either 0 or 1 the equal real and imaginary axis is calculated otherwise not.

The performance of any circuit in VLSI design limits by the constituent factors like power, delay and area. Fig 8 and Fig 9 shows the RTL schematic of 16 bit and 32 bit Butterfly structure respectively. Table 3 shows the results for 16 bit and 32 bit Butterfly structure that has been calculated from the synthesis report in terms of area utilization and delay. With the help of power analyzer, power is calculated. The design is tested and verified by Verilog HDL coding and simulation is carried out by Xilinx ISE 14.1 design suite.

Table 3: Performance Evaluation of Radix 2 Butterfly Structure

Circuit Designed		16 bit WM	32 bit WM
No. of bonded IOB	I –Buf	96	192
	O- Buf	86	166
No. of Slice LUT		2638	10861
Power	Leakage	0.082	0.115
Total (W)	IOs	0.050	0.050
	Logic	7.049	8.062
Delay (ns)	Router	16.522	21.426
Power Delay Product (WnS)		3.1113	4.8360

5. Conclusion

The performance of any circuit in VLSI design is limited by the constituent factors like power, delay and area. In processing application there is a requirement of an efficient algorithm. Butterfly structure in FFT algorithm consumed most of the power. So, implementation of an efficient Butterfly is the most important part of the FFT algorithm. Adders and multiplier have been designed, implemented and simulated using Xilinx ISE 14.1 design suite while targeting Spartan 3E FPGA for synthesis. In this paper KSA adder is implemented because as compared to other adders KSA is best in terms of delay. Using this efficient adder high speed multiplier circuit was implemented leading to the development of an efficient multiplier block. These efficient adders and multipliers blocks are used for designing Butterfly structure so to design an efficient FFT algorithm implementation for various applications. Implementation is done using Verilog coding and tool used is Xilinx ISE 14.1 design suite. Power is calculated with the help of Xpower analyzer. Future work is dedicated for reducing the power consumption.

References

[1] Bais K. & Ali Z., "Comparison of various adder designs in terms of delay and area", International journal of science and research (IJSR), Volume 5, Issue 5, May 2016.

[2] Anjana.R, Abishna.B, Harshitha.M.S., Abhishek.E, Ravichandra V., Suma M. S., "Implementation of Vedic Multiplier using Kogge-Stone Adder", International Conference on Embedded Systems - (ICES), 2014.

[3] S. Rajaram, K. Vanithamani, "Improvement of Wallace multipliers using Parallel prefix adders", International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011.

[4] S.Murugeswari, S.Kaja Mohideen, "Design of Area Efficient and Low Power Multipliers using Multiplexer based Full Adder", 2nd International Conference on Current Trends in Engineering and Technology, ICCTET, 2014.

[5] A. K. Singh & A. Nandi, "Design of Radix 2 Butterfly Structure using Vedic multiplier and CLA on Xilinx", Proc. IEEE Conference on Emerging Devices and Smart Systems, 3-4 March 2017.

[6] A. Thomas, A. Jacob, S. Shibu & S. Sudhakaran, " Comparison of Vedic Multiplier with Conventional Array and Wallace Tree Multiplier", International Journal of VLSI System Design and Communication Systems, Vol. 04, Issue. 04, April 2016.

[7] A. Maiti , K. Chakraborty, R. Sultana & S. Maity, "Design and implementation of 4-bit Vedic Multiplier", International Journal of

Emerging Trends in Science and Technology, Vol. 03, Issue 05, Pages 3865-3868, May 2016.

[8] A. Mankar, FPGA Implementation of Fast Fourier Transform Core using NEDA, Mtech Thesis , National Institute of Technology, Rourkela, 2011-2013.

[9] M. Patrikar & V. Tehre, "Design and Power Measurement of 2 and 8 Point FFT using Radix-2 Algorithm for FPGA Implementation", IOSR Journal of VLSI and Signal Processing, Vol. 7, Issue 1, Jan-Feb 2017, PP 44-48.

[10] J. G. Proakis & D. G. Manolakis, "Digital Signal Processing Principle, Algorithms, and Applications", 4th Edition, 2007, Prentice Hall India Publications.

[11] M. J. Rashmi, G. S. Biradar & M. Patil, "Efficient VLSI Architecture using DIT-FFT Radix-2 and Split Radix FFT Algorithm", International Journal for Technological Research in Engineering, Vol. 1, Issue 10,June 2014.