ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 6, June 2017

AN EFFICIENT VLSI IMPLEMENTATION OF MULTIPLIER LESS FFT DESIGN

Mrs.N.Madhubala and Dr.A.Kavitha,Associate Professor

***Abstract*— The proposed system is to implement a novel approach to implement multiplier less unity-gain fast Fourier transforms(FFT) and inverse fast fourier transform (IFFT). Previous methods achieve unity-gain FFTs by using either complex multipliers with additional scaling compensation. Conversely, this brief proposes unity-gain FFTs without compensation circuits, even when using non-unity-gain rotators. This is achieved by a joint design of rotators, hence entire FFT is scaled by a power of two, which is then shifted to unity. This reduces the amount of hardware resources of the FFT architecture. The proposed approach can be Implemented using Verilog HDL and Simulated by Model sim 6.4 c Finally its Synthesized by Xilinx tool.. The 16 point DIT-FFT and DIF-IFFT architecture is implemented as a Multiplier less architecture. The Sixteen point DIT-FFT architecture consists of two points, four points, and Eight Point and Sixteen point as Stages. The design of the FFT & IFFT architecture is based on Decimation-in-Time and Decimation-in –Frequency respectively.**

Index terms-FFT,IFFT and Shift-and-add implementation.

I. INTRODUCTION

In Digital signal processing (DSP) world, there is often a need to convert signals between time and frequency domains. For this reason, the fast Fourier transform (FFT) has become one of the most important algorithms in the field. In order to calculate the FFT efficiently, various hardware architectures have been proposed. The implementation of FFT processor for OFDMA system shows the MCM (Multi Carrier Modulation) techniques by integrating an FFT processor with the existing OFDMA system. The OFDM technique (Orthogonal Frequency Division Multiplexing) has been used ealier in applications that require a huge transmission rate like ADSL modem, wireless network (Wi-Fi 802.11), DVB (Digital Video Broadcasting) and DAB (Digital Audio Broadcasting).

***N.Madhubala**, Electronics and Communication Engineering, Jayaram College of Engineering and Technology, (Trichy, India, 9787673509*

***Dr.A.Kavitha**, Electronics and Communication Engineering, Jayaram College of Engineering and Technology, Trichy, India,9750962271*

In all those cases, there is a need to implement an integrated circuit (IC) that performs the necessary chip functions. For prototyping circuits FPGA is better choice than using ASIC. The FPGA implementations is allow these modern technologies of data transmission to use and it is necessary to develop the efficient techniques of digital modulation. Recently OFDM is utilized principally more, because it uses the efficient FFT algorithm to do the modulation. When high performance is required, feedback and feed forward hardware FFT architectures are attractive options. They offer high throughput capabilities. Single-delay feedback (SDF) FFT architectures consist of a series of stages that process one sample per clock cycle. Each stage contains a butterfly and a rotator. The butterfly calculates additions, and the rotator carries out rotations in the complex plane by given rotation angles, called twiddle factors. Compared with the additions of the butterfly, rotations are more costly operations. For this reason, different approaches to implement rotators have been proposed in the past. The most straightforward approach is to use a complex multiplier, which consists of four real multipliers and two adders. In addition, it requires a memory to store the twiddle factors.

In this brief, we present the novel multiplier less unity-gain FFTs. The discrete Fourier transform (DFT) matrix factorization based on the Kronecker product. It express the family of radix single-path delay feedback / single-path delay commutator(SDF/SDC) pipeline fast Fourier transform (FFT) architectures.[2] This paper presents a new approach to design the multiplier less constant rotators. This approach is based on a combined coefficient selection and shift-and-add implementation (CCSSI) for the design of the rotators. [11] This paper presents a power and area minimization of flexible FFT processors methodology. It is based on the power-area tradeoff space obtained by adjusting algorithm, architecture, and circuit variables. [13] This brief is organized as follows. In Section II, we explain an introduction to the SDF architecture and the FFT twiddle factors. In Section III, we explain the proposed approach and provide optimized architecture FFT. In Section IV, we compare the proposed architectures to previous Normal Multiplier based FFT Implementation. In Section V, we present the experimental results, and in Section VI, we summarize the main conclusions of this brief.

II. OBJECTIVE & OVERVIEW OF FFT & IFFT

A. Objective

The main objective of the work is to implement the novel multiplier less unity-gain FFTs. They are obtained by designing the rotators in all FFT stages simultaneously, so that the output of the FFT has unity gain. Thus, the proposed approach requires neither the use of costly unity-gain rotators, nor circuits to compensate the scaling. This reduces the complexity of the FFT rotators and guarantees unity gain for the FFT. In this brief, we study different FFT sizes and propose suitable solutions for each size. The 16 point DIT-FFT architecture is implemented as a Multiplier less architecture. The sixteen point DIT-FFT architecture consist of two point, four point, eight point and sixteen point as Stages. The design of the FFT architecture is based on Decimation in Time.

B. Overviwew of FFT & IFFT

Let consider the computation of the N = 2v point DFT by the divide-conquer approach. Here N-point data sequence is split into two N/2-point data sequences f1(n) and f2(n), corresponding to the even-numbered and odd-numbered samples of x(n), respectively that is

$f_1(n)=x(2n)$

$f_2(n)=x(2n+1)$ $\qquad n=0,1,\ldots.(N/2-1)$

Above f1(n) and f2(n) are obtained by decimating x(n) by a factor of 2. Thus the resulting FFT algorithm is called a decimation-in-time algorithm.

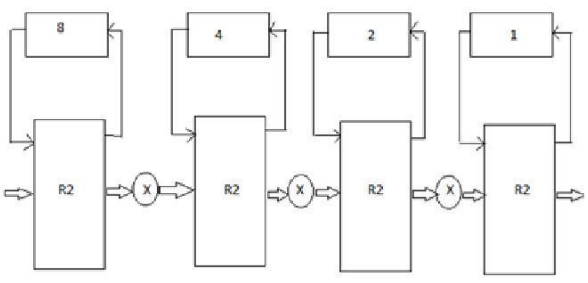

Fig 1 General block diagram

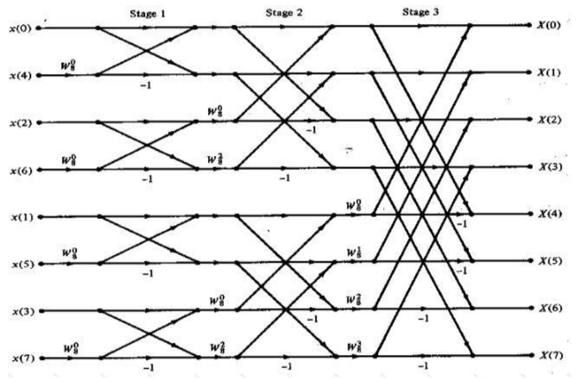

Fig 2 Eight-point decimation-in-time FFT algorithm

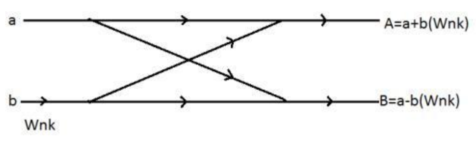

Fig 3 Basic butterfly computation in the DIT FFT Algorithm

Another important radix-2 FFT algorithm, called as the decimation-in-frequency algorithm. It is obtained by using the divide-and-conquer approach. To derive the algorithm, we begin the DFT formula by splitting into two summations, one of which involves the sum over the first N/2 data points and the second sum involves the last N/2 data points. Thus we obtain

[illegible]

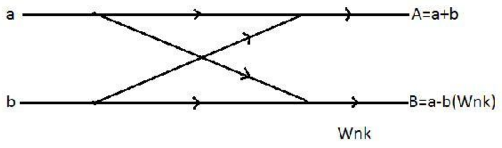

Fig 4 Basic butterfly computation in the decimation-in-frequency

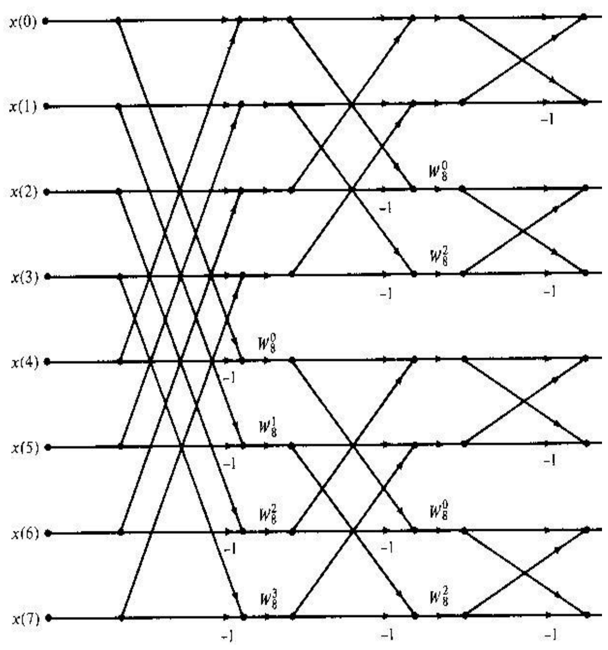

Fig 5 N = 8-piont decimation-in-frequency IFFT algorithm

The input data x(n) results in natural order, but the output DFT occurs in bit-reversed order. We also note that the computations are performed in place. However, it is possible to reconstruct the decimation-in-frequency algorithm. Hence the input sequence occurs in bit-reversed order while the output DFT occurs in normal order. Furthermore, it is also possible to have both the input data and the output DFT in normal order, if we abandon the requirement that the computations be done in place.

C. Proposed Multiplierless FFT

The proposed system is to implement the novel multiplier less unity-gain FFT & IFFT. They are obtained by designing the rotators in all FFT & IFFT stages simultaneously, so that the output of the FFT has unity gain. Hence the proposed approach neither requires the use of costly unity-gain rotators, nor circuits to compensate the scaling. This reduces the complexity of the FFT& IFFT rotators and guarantees unity gain for the FFT & IFFT. In this brief, suitable solutions for each different FFT & IFFT sizes are proposed. The 16 point DIT-FFT architecture is implemented as a Multiplier less architecture. The Sixteen point DIT-FFT architecture consists of two points, four points, and eight Point and sixteen point as Stages. The Design of the FFT architecture is based on Decimation in Time. Instead of Complex multiplier we made Shift and add Multiplier.

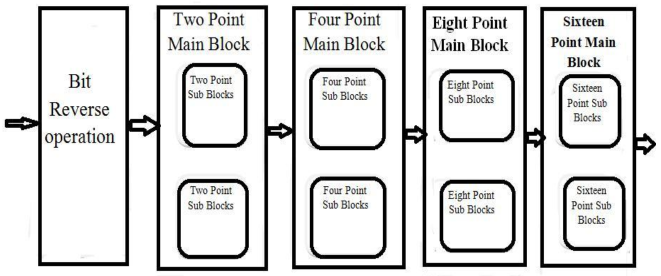

Fig 6 Block diagram of the proposed system

Some of the important applications of FFT includes polynomial multiplication and Fast large integer, Efficient matrix-vector multiplication for Toeplitz, circulant and other structured matrices Filtering algorithms, Fast algorithms for discrete cosine or sine transforms.

III. Output Results & Comparison

Simulation & Synthesis Result

Fig 7 Simulation results of Proposed FFT& IIFFT Design

Fig 8 Main RTL Schematic View of Proposed Design

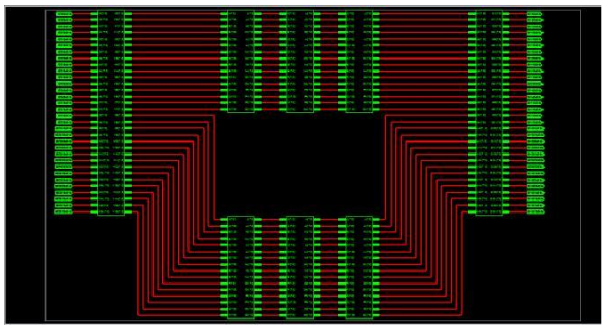

Fig 9 Internal View of Proposed Design

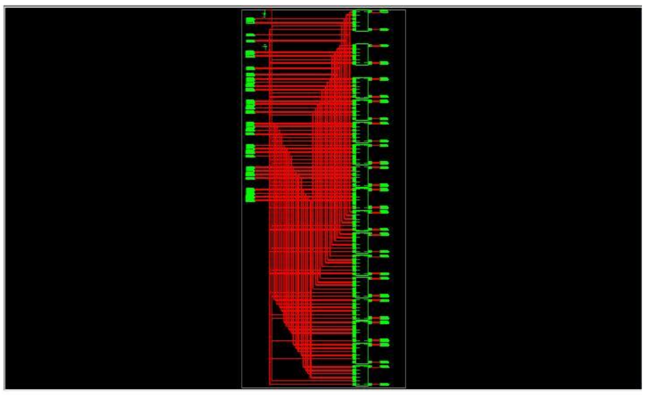

Fig 10 Sixteen point FFT RTL schematic

A. Comparison Results

Here we compare the Proposed Multiplier Less FFT Design and Existing Normal FFT Implementation. Here we compare area and delay in terms LUT, Slices, Gate count and Gate delay, Path Delay. The bellow table provides comparison between the normal FFT multiplier and the proposed FFT multiplier.

TABLE I

METHOD NAME	AREA				DELAY		
	LUT	SLICES	GATE	IOB	Over all Delay	Gate Delay	Path Delay
Normal Multiplier Based FFT	2890	1474	30659	512	35.254ns	17.334ns	17.920ns
Proposed Multiplier Less FFT	1698	990	18478	512	26.801ns	13.561ns	13.240ns

TABLE II

METHOD NAME	AREA				DELAY		
	LUT	SLICES	GATE	IOB	Over all Delay	Gate Delay	Path Delay
Normal Multiplier Based IFFT	2821	1427	30138	512	34.643ns	17.875ns	16.768ns
Proposed Multiplier Less IFFT	1988	1160	22314	512	29.876ns	16.140ns	13.736ns

IV. CONCLUSION

This brief shows how to design multiplier less unity-gain 16 Point FFT and IFFT architectures. The proposed architectures are not

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 6, June 2017

only multiplier less and achieve Less area and Delay, but also require the smallest number of adders among current SDF FFTs. The proposed architectures achieve good figures of merit in terms of clock frequency, area. The 16 Point FFT & IFFT implements by Verilog HDL& System Verilog and Simulated in Modelsim 6.4 c and synthesized by Xilinx tool.

REFERENCES

1. Abdullah S.S., Nam H., Mc Dermot M. and Abraham J. A. (2009), 'A high throughput FFT processor with no multipliers', in Proc. IEEE Int. Conf. Comput. Design, pp. 485–490.

2.Adiono T., Irsyadi M.S., Hidayat Y.S. and Irawan A. (2009), '64-point fast efficient FFT architecture using radix-23 single path delay feedback', in Proc. Int. Conf. Elect. Eng. Inform. (ICEEI), vol. 2, pp. 654–658.

3. Cho T. and Lee H. (2013), 'A high-speed low-complexity modified radix-25 FFT processor for high rate WPAN applications', IEEE Trans. VLSI Syst., vol. 21, no. 1, pp. 187– 191.

4. Cortes A., Velez I., and Sevillano J.F. (2009), 'Radix rk FFTs: Matricial representation and SDC/SDF pipeline implementation', IEEE Trans. Signal Process., vol. 57, no. 7, pp. 2824–2839.

5. Cuong N.H., Lam N.T., and Minh N.D. (2012), 'Multiplier-less based architecture for variable-length FFT hardware implementation', in Proc. IEEE 4th Int. Conf. Commun. Electron., pp. 489–494.

6. Garrido M., Grajal J., Sánchez M.A., and Gustafsson O. (2013), 'Pipelined radix-2k feedforward FFT architectures', IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 21, no. 1, pp. 23–32.

7. Garrido M., Gustafsson O., and Grajal J. (2011), 'Accurate rotations based on coefficient scaling', IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 10, pp. 662–666.

8. Garrido M., Qureshi F., and Gustafsson O. (2014), 'Low-complexity multiplier less constant rotators based on combined coefficient selection and shift and- add implementation (CCSSI)', IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 7, pp.

9. Meher P.K., Valls J., Juang T.B., K.Sridharan, and Maharatna K. (2009), '50 years of CORDIC: Algorithms, architectures, and applications', IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907.

10. Yang C.H., Yu T.H., and Markovic D. (2012), ;Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example', IEEE J. Solid- State Circuits, vol. 47, no. 3, pp. 757–768.

11. Yang L., Zhang K., Liu H., Huang J., and Huang S. (2006), 'An efficient locally pipelined FFT processor', IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 7, pp. 585–589.

12. Zhou B., Peng Y., and Hwang D. (2009), 'Pipeline FFT architectures optimized for FPGAs', Int. J. Reconfigurable Comp., vol., pp. 1–9, Jun. 2009, Art. no. 219140.

13. Zhong G., Zheng H., Jin Z., Chen D., and Pang Z. (2011), '1024-point pipeline FFT processor with pointer FIFOs based on FPGA,' in Proc. IEEE/IFIP Int. Conf. VLSI-SoC,pp. 122–125.