International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 3 (March 2014), PP.06-10

Design of an area efficient FFT/IFFT processor for WPAN applications

S.Leena jebamoney[1], A.Nandhini[2], S.Krishnapriya[3]
[1]Department of ECE, M.Tech VLSI Design, Sathyabama University, Tamilnadu, India.
[2]Department of ECE, Assistant professor, Sathyabama University, Tamilnadu, India.
[3]Department of ECE, M.Tech VLSI Design, sathyabama university, Tamilnadu, India.

Abstract:- In a high speed WPAN applications the Orthogonal Frequency Division Multiplexing (OFDM) modulation has been adopted and FFT Processor is a key component. This paper presents a new technique to reduce the complex multiplication in FFT processor with less area occupied. In this paper a FFT/IFFT Processor using Radix-2^5 algorithm and a Wallace tree multiplier for efficient FFT processor is designed. This method reduces the size of twiddle factor memory and number of complex multiplications. The Results shows that the proposed design is area efficient design which achieves less Hardware complexity

Keywords:- Fast Fourier Transform (FFT), Orthogonal Frequency Division Multiplexing (OFDM), Wallace tree multiplier modified radix-2^5, Wireless Personal Area Network (WPAN).

I. INTRODUCTION

High rate WPAN systems will provide various Multimedia applications such as home network and real time video streaming. WPAN systems make use of Orthogonal Frequency Division Multiplexing modulation has been adopted. In that FFT processor is a key component in OFDM modulation of high rate WPAN system the high hardware complexity of FFT/IFFT processor exists, IEEE 802.11ad consists of 512 subcarriers hence the FFT/IFFT processor computes the FFT for 512-point arithmetic and should provide high throughput rate. The FFT/IFFT processors must be pipelined to provide high throughput. Many FFT architectures hav been designed to utilize the OFDM transmission effectively such as Single path Delay Feedback(SDF) and Multi-path Delay Feedback (MDF).among that MDF is mostly preferred in order to provide a high throughput .so to reduce the area and power consumption several FFT algorithms have been proposed

The radix of the algorithm greatly influences the architecture of the FFT/IFFT processor. Small radix results in simple butterfly and higher radix will reduce the twiddle factor multiplication. hence r^k algorithm is much preferred so that simple butterflies and reduced twiddle factor multiplication is achieved. The Radix-2 algorithm is a well known simple algorithm for FFT/IFFT processor. the Radix-4algorithm is suitable to achieve a high Data rate[8].The 512-point FFT/IFFT processor is designed with the modified Radix-2^5 algorithm[1] is proposed which can attain a high throughput rate. The speed of the processor is increased in the design of Proposed FFT/IFFT processor with the use of high speed Wallace tree multiplier instead of complex booth multiplier.

The important units in FFT processor design are input unit, control unit, output unit and computational unit. the necessary computations are performed in computational unit, the computational complexity of tis unit need to be reduced in order to achieve high speed processor design hence Wallace tree multiplier is designed and used in the proposed FFT/IFFT processor design. The Wallace tree multiplier has less interconnection delay hence high speed can be achieved .the wallce tree is an efficient hardware implementation of digital circuit that multiplies two operands and produce results.so the area is occupied is occupied efficiently. It takes in the operands and produces a results. The implementation of Wallace tree multiplier using 5:2 compressor decreases the latency.

The simple block diagram of 512-point FFT/IFFT processor that makes use of complex booth multiplier is shown in fig.1 which occupies more area in designing a FFT/IFFT processor for WPAN applications .the booth multiplier results in increased look up table (LUT) size hence the twiddle factor memory size is also increased.

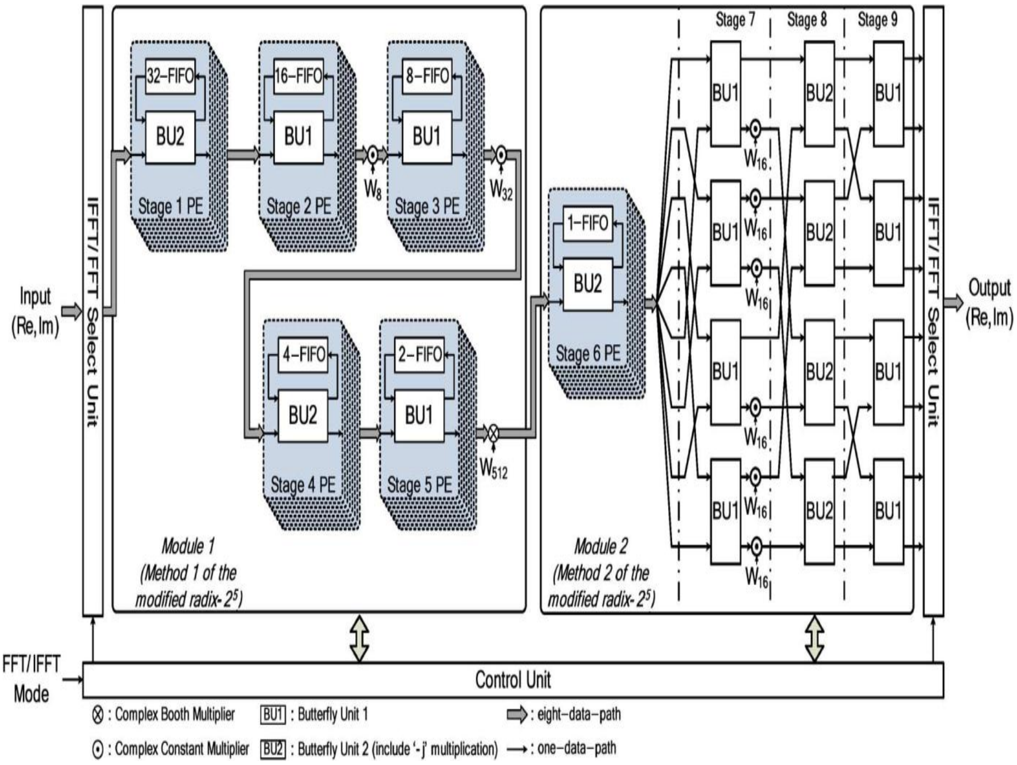


Fig.1: Block diagram of 512-point FFT/IFFT processor with booth multiplier

II. PROPOSED FFT PROCESSOR

In this paper an area efficient FFT processor is designed.The proposed architecture consists of two modules with butterfly units, wallace multipliers, constant multipliers, first-in first-out (FIFO), and a control unit. The butterfly units perform complex additions and subtractions of two input data.The main objective of proposed processor is to reduce the area occupied. firstly we will describe about the proposed block diagram.

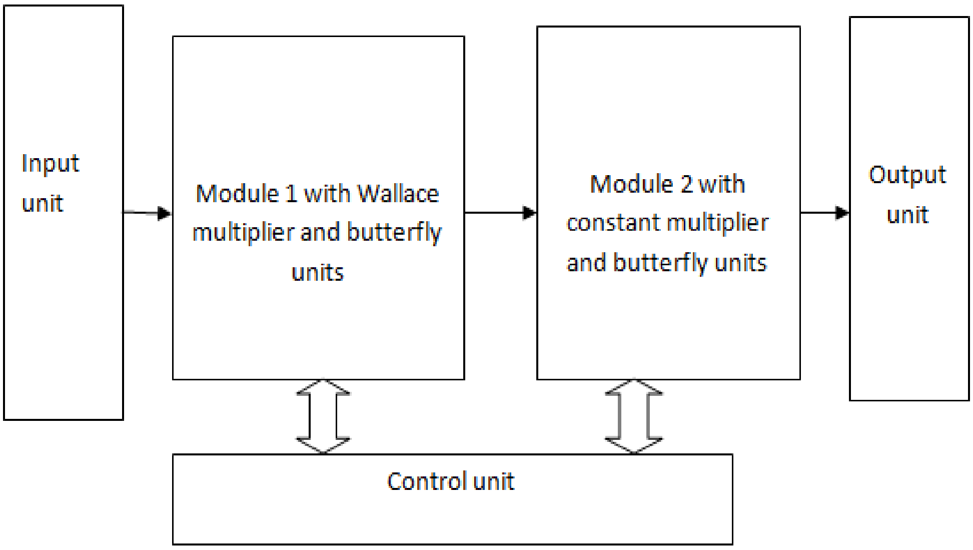

Input unit
Module 1 with Wallace multiplier and butterfly units
Module 2 with constant multiplier and butterfly units
Output unit
Control unit

Fig.2 Block diagram of proposed FFT/IFFT processor

A. Butterfly units

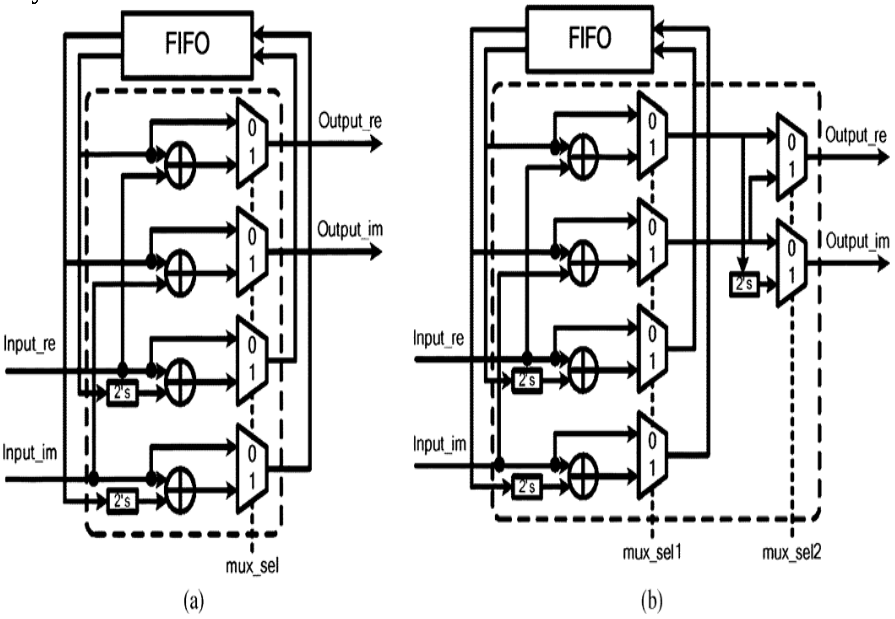


Fig.3: Butterfly units

The butterfly units perform complex additions and subtractions of two input data X[n] and X [n+N/2]. The behaviour of the butterfly units is as follows. All input values are saved into the FIFO until the N/2th input is entered. Then, the butterfly units conduct calculations between the input values and FIFO outputs, after entering (N/2)+1st the input. During the last N/2 clock cycles, all butterfly calculations are performed at each stage. Among the butterfly outputs, the complex addition outputs are fed to the next stage. And, the complex subtraction outputs are saved in the FIFO, and then during the next N/2 clock cycles, the FIFO outputs are fed to the next stage. Butterfly unit 1 (BU1) conducts complex additions and subtractions only. However, butterfly unit 2 (BU2) includes twiddle factor W_4 multiplication utilizing the multiplexers and control signals.

B. Constant multiplier

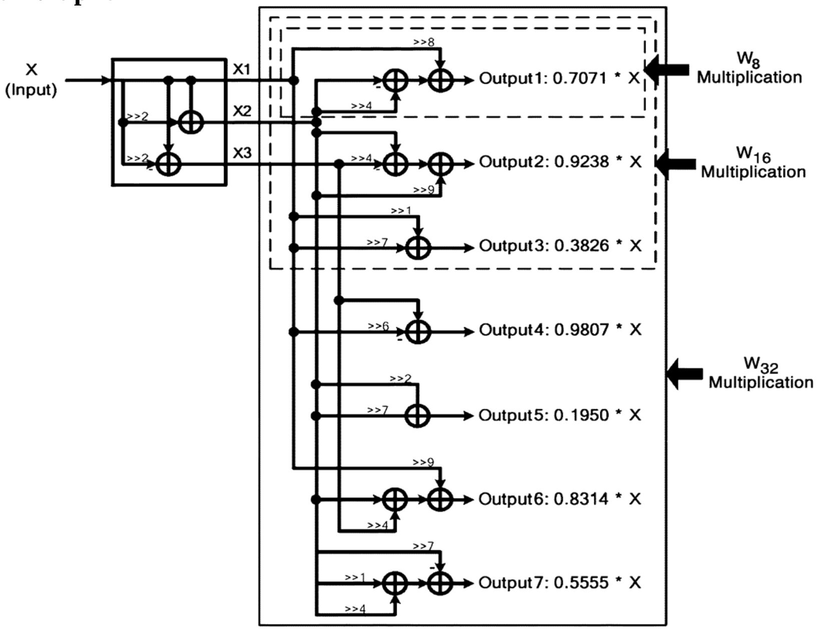


Fig.4 Constant multiplier

A complex multiplier can compute (a + j · b) (c + j · d) = (ac − bd) + j · (ad + bc). Here a + j · b is the multiplicand and c + j · d is the multiplier. These have both real and imaginary parts. This operation can be done by four real multipliers, one adder and one subtractor. The subtractor can be implemented by an adder with a carry 1. The twiddle factor multiplication is conducted using fixed width complex multipliers.

C. Wallace tree multiplier

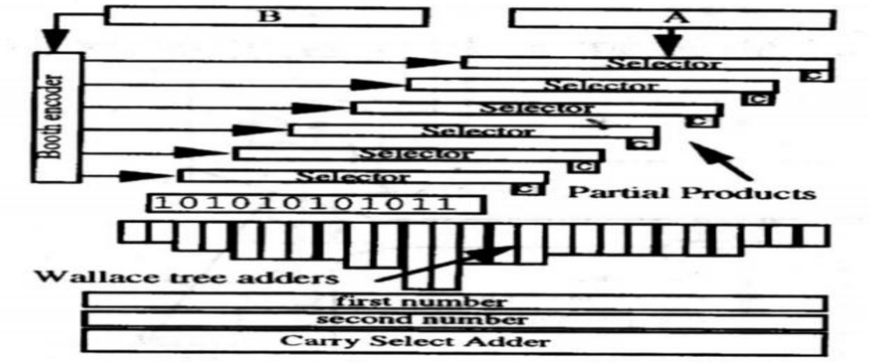


Fig.5 Wallace tree multiplier

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. Since Non-regularities in the construction of traditional multipliers result in a large amount of wasted area. The multiplier takes in operands: the multiplier (MR) and the multiplicand (MD), and produces the 8 multiplication result of the two at its output .The Desired architecture of the multiplier primarily consists of four major modules. Booth encoders, Wallace Tree adders and the Carry select Adders

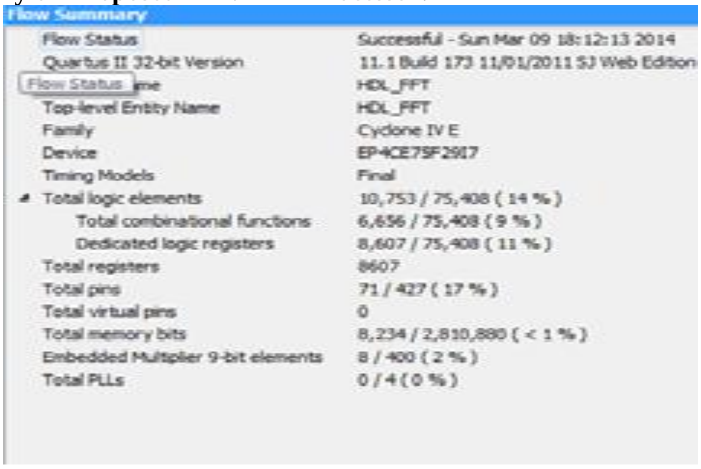
III. RADIX-2^5 ALGORITHM

The Radix-2^5 algorithm incorporates a series sum split of four every stage, as compared to two in Radix-2. This way, the initial N-point DFT is divided to give four *(N/4)*-point FFT computations. Since the Radix-2^5 algorithm is the mutation of the Radix-2, all predominant features that apply to Radix-2 hold true here as well namely the highly regular structure, the 'in-place' data-addressing between stages, scope of pipelining, etc. The mathematical formulation of the algorithm. Since, every butterfly works on four inputs - $x(4n)$, $x(4n+1)$, $x(4n+2)$, $x(4n+3)$, where $n = 0, 1, \ldots, (N/4)-1$ – feeding the butterfly with non-consecutive data ensemble values makes vectorial processing a bit tricky for generic type DSP processor cores The same applies to the addressing and handling of twiddle factors, since each twiddle factor needs to be referenced and fetched separately. Thus, while the Radix-2^5 algorithm is 'faster' from an algorithmic perspective – as it completes the N point FFT in $log_4(N)$ stages, as compared to $log_2(N)$ stages in Radix-2. There is an added detail that the algorithm is better suited for a FFT of size N that is a proper power of 5 [9].

IV. PERFORMANCE ANALYSIS

This section mainly discuss about flow summary of the proposed design and the simulation results from which we can infer the area occupied The proposed FFT/IFT processor is designed in verilog HDL and simulated to verify its functionality. both the simulation and synthesis is carried out using Quartus II and model sim tool .The proposed design is implemented in cyclone iv family to analyse the area occupied. The results show that the proposed design occupies less area, which reduces the hardware complexity.

A. Flow Summary of Proposed FFT/IFFT Processor:



Flow Summary

Flow Status	Successful - Sun Mar 09 18:12:13 2014
Quartus II 32-bit Version	11.1 Build 173 11/01/2011 SJ Web Edition
Flow Status me	HDL_FFT
Top-level Entity Name	HDL_FFT
Family	Cyclone IV E
Device	EP4CE75F29I7
Timing Models	Final
Total logic elements	10,753 / 75,408 (14 %)
Total combinational functions	6,656 / 75,408 (9 %)
Dedicated logic registers	8,607 / 75,408 (11 %)
Total registers	8607
Total pins	71 / 427 (17 %)
Total virtual pins	0
Total memory bits	8,234 / 2,810,880 (< 1 %)
Embedded Multiplier 9-bit elements	8 / 400 (2 %)
Total PLLs	0 / 4 (0 %)

Fig .6: Flow summary

This flow summary shows the area, total logic elements are 14% area using cyclone IV family. This figure discuss about area of the FFT processor.

B. Comparison Table:

The table 1 shows the comparison of area occupied by the design.

Table I: Comparison Table

PROPOSED DESIGN	OCCUPIED	AVAILABLE
TOTAL LOGIC ELEMENTS OCCUPIED	10,753 (14%)	75,408
TOTAL MEMORY BITS	8,234 (<1%)	2,810,880
TOTAL COMBINATIONAL FUNCTIONS	6,656 (9%)	75,408
TOTAL PINS	71 (17%)	427

V. CONCLUSIONS

We proposed an area efficient FFT/IFFT Processor with Wallace tree multiplier The number of complex multipliers and twiddle factor LUTs are reduced using the Radix-2^5 algorithm. The proposed FFT/IFFT design is the most area-efficient architecture for FFT processors.

In future this can use in any WPAN application depending upon the need and also any multiplier that reduces the complex twiddle factor multiplication can be used in the design to improve its performance.

REFERENCES

[1]. M.Ayinala.., .M.Brown., and Parhi.K.K,(Jun.2012) “A Pipelined parallel FFT architectures via folding transformation,” IEEE Trans. Very Large Scale Intergr. (VLSI) Syst., vol. 20, no. 6, pp 1068–1081.

[2]. H.F.Chi and Z.H.Lai ,(May 2005) “A cost-effective memory-based real-valued FFT and Hermitian symmetricIFFT processor for DMT-based wire-line transmission systems,” in Proc. IEEE Int. Symp. Circuits Syst., vol. 6, pp. 6006–6009.

[3]. K.Cho., J.Chung.,K Lee., and Parhi.K.K ,(May 2004) “Design of low-error fixed width modified booth multiplier,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531.

[4]. Y.Chen., C.Lee. C.Lin., Y.Tsao., and Y. Wei.,(Feb.2008) “An indexed-scaling pipelined FFT processor for OFDM-based WPAN applications,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 2, pp. 146–150.

[5]. A.Cortes., J.F.Sevillano, and I.Velez.,(Jul.2009) “Radix –r^k FFTs :Matrical representation SDC/SDF pipeline implementation,” IEEE Trans. Signal Process., vol. 57, no. 7, pp. 2824–2839

[6]. M.Garrido., J.Grajal., and Parhi.K.K,(Dec.2009) “A pipelined FFT architecture for real-valued signals,” IEEE Trans.Circuits Syst. I, Reg. Papers, vol. 56, no. 12, pp. 2634–2643

[7]. R.I. Hartley.,(Oct.1996).“Subexpression sharing in filters using canonic signed digit multipliers,” IEE Trans.circuits Syst. II, Exp. Briefs, vol. 43, no. 10, pp. 677–688.

[8]. C.Y.Lin. and C.Y. Lee, (Apr.2007).“Design of an FFT/IFFT processor for MIMO OFDM systems,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 4, pp. 807–815.

[9]. Y.Lin,C. Lee.,and H.Liu., (Aug.2005) “A 1-GS/s FFT/IFFT processor for UWB applications,” IEEE J. Solid-State Circuits, vol. 40, no. 8, pp. 1726–1735.

[10]. J.Lee and H.Lee (Apr.2008) “A high-speed two-parallel Radix-2^4 FFT/IFFT processor for MB-OFDMUWB systems,” IEICE Trans. Fundam., vol. E91-A, no. 4, pp. 1206–1211.

[11]. S.Tang, T.Chang, and J.Tsai, (Jun.2010). “A 2.4-GS/s FFT processor for OFDM based WPAN applications,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 6, pp. 451–455.

[12]. Y.Voronenko. and M.Püschel., (Jan.2009) “Algebraic signal processing theory: Cooley-Tukey type algorithm for real DFTs,” IEEE Trans. Signal Process., vol. 57, no. 1, pp. 205–222.