# **Implementation of Low-Power Split-Radix FFT Processor**

B.Ramamohan Associate professor, Department of ECE, Lendi Institute of Engg & Tech.Vizianagaram, A.P,INDIA. ramamohanmailid@gmail.com

V.Nancharaiah Associate professor, Department of ECE, Lendi Institute of Engg & Tech, Vizianagaram, A.P, INDIA. nanch84@gmail.com

## V.Y.S.S.Sudheer Patnayakuni

Assistant professor, Department of ECE, Lendi Institute of Engg&Tech.Vizianagaram, A.P, INDIA. Sudheer.its7@gmail.com

**Abstract:** Fast Fourier Transform is an important algorithm in signal processing applications like MIMO-OFDM, 4G, VoLTE and 5G. FFT computation involves with complex arithmetic operations addition/subtraction and multiplication. The power dissipation of FFT depends on the number of arithmetic computations. Split Radix FFT (SRFFT) has the lowest number of arithmetic operations among all the FFT. SRFFT is an ideal solution for low power FFT design. In this paper, we proposed the implementation of SRFFT for 4/8/16-point FFT on Xilinx Zed board. System generator tool is used for FFT design. SRFFT design is done using Xilinx blockset and code generation is done using Xilinx Vivado. The generated code is implemented on Xilinx Zed board. Comparisons are made in terms of LUTs occupied and power dissipation between 4-point, 8-point and 16-point FFT implementation.

Keywords—low power, Radix-2, Split-Radix Fast Fourier Transform (SRFFT), twiddle factors, system generation tool, Vivado suit, Xilinx blockset.

## **I.INTRODUCTION**

Fast Fourier transform (FFT) is the most important and commonly used algorithm in Multiple Input and Multiple Output-Orthogonal Frequency Division Multiplexing (MIMO-OFDM), High definition multimedia applications, wireless applications and signal processing applications. Many variants of the FFT algorithm have been developed, such as radix-2, radix-4 and radix-8 FFT. A new variant of FFT algorithm was proposed by Duhamel and Hollmann [1] called splitradix FFT (SRFFT). FFT algorithm involves with arithmetic operations like Addition/Subtraction and Multiplication of complex numbers. The overall system power consumption depends on number of arithmetic computations in FFT. SRFFT algorithm involves with least number of arithmetic computations among all the known FFT algorithms. SRFFT is a good candidate for the implementation of a low-power FFT processor.

FFTs are classified into two groups: shared-memory and pipelined processors. Pipelined processor architectures [2-3] provide high throughputs at the cost of more hardware resource requirement. Shared-memory processor architectures [4-11] require less hardware resources with slower throughput. In this paper, we propose the implementation of SRFFT algorithm of 4-point, 8-point and 16-point using system generator tool. The design of FFT is carried out using Xilinx blockset. The code generation is performed and synthesized using Xilinx Vivado tool.

The rest of this paper is organized as follows. Section II provides a theoretical comparison of the number of complex multiplications between the radix-2 FFT and the SRFFT. Section III discusses the implementation of the proposed design. Section IV provides the results and Section V concludes this paper.

### II. COMPARISON OF SRFFT AND RADIX-2 FFT

The N-point discrete Fourier transform is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$
(1)

Where k=0,1,2.....N-1and  $W_N^{nk} = e^{-j2\pi nk/N}$ . If we split X(k) into even and odd terms, radix-2 FFT can be derived as

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x(n+N/2)] W_{N/2}^{nk}$$
(2)  
$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) - x(n+N/2)] W_{N/2}^{nk} W_N^n$$
(3)

The basic idea behind the SRFFT is the application of a radix-2 index map to the even-index terms and a radix-4 map to the Odd-index terms. For the even-index terms, it can be decomposed as (2).

$$X(4k+3) = \sum_{n=0}^{\frac{N}{4}-1} [x(n) - x\left(n + \frac{N}{2}\right) + j\left(x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right)\right)]W_N^n W_{N/4}^{nk}(4)$$
  
$$X(4k+3) = \sum_{n=0}^{\frac{N}{4}-1} [x(n) - x\left(n + \frac{N}{2}\right) + j\left(x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right)\right)]W_N^n W_{N/4}^{nk}$$
(5)

Where k = 0, 1, ..., N/4. The formulas above result in the L-shaped split-radix butterfly structure, which can be found in [2] and the scheduling of the L-shaped butterfly is irregular. Table 1 shows the comparison of arithmetic computations required in Radix-2 and Split radix of 16-point FFT. Figs. 1 and 2 show the signal flow graph of Radix-2 and Split Radix 16-point FFT.

 TABLE I

 Comparison of arithmetic computations between Radix-2 and Split radix FFT of 16-point

Туре	Additions/Subtractions	Multiplications
Radix-2	64	24
Split Radix	46	21

We present a split radix FFT algorithm consisting of radix-4 butterflies. The major advantages of the proposed algorithm include mixed radix butterflies, whose structure is more regular than the conventional split radix algorithm. The split radix FFT is obtained by dissolving the radix of 4 point into 2 and 4 divisional part subsequently so these divisions help in reducing the power and many other related computations.



Fig.1: Signal flow graph of radix-2 FFT

## International Journal of Management, Technology And Engineering



The above diagrams represent the difference in the structural representation of the both FFT and split radix FFT.

## **III.IMPLEMENTATION OF PROPOSED DESIGN**

The proposed design of SRFFT of 4-point, 8-point and 16-point is implemented on Zed board using Xilinx Vivado tool. The proposed design is carried out using Xilinx System generator tool. The design is done using Xilinx Blockset tool of System generator tool.



Fig.3: Block diagram of butterfly element.

Fig.3 shows the block diagram of butterfly element used in FFT design. It performs the addition in the upper section and subtraction in the lower section. Twiddle factor multiplication is to be performed in the lower section of butterfly element. The multiplication of complex twiddle factor is performed as shown in Fig.4.



Fig. 4: Block diagram of twiddle factor multiplication

Using the proposed butterfly element and twiddle factor multiplication units, the design of FFT for 4-point, 8-point and 16-point is carried out. Fig. 5 shows the design of 4-point FFT.



Fig. 5:Design of 4-point FFT

After design of FFT using Xilinx blockset, code generation is performed. The generated Verilog code is synthesized and implemented on Zed board using Xilinx Vivado tool. Fig.6 shows the RTL schematic of proposed design.



Fig.6: RTL schematic of 4-point split radix FFT

Similarly, the design for 8-point and 16-point FFT is carried out and synthesized using Vivado tool. Fig.7 shows the design of 8-point split radix FFT and Fig.8 shows the RTL schematic of 8-point split radix FFT. Fig.9 shows the RTL schematic of 16-point split radix FFT.



Fig.7: Design of 8-point split radix FFT.



Fig.8: RTL schematic of 8-point split radix FFT



Fig.9: RTL schematic of 16-point split radix FFT

## **IV RESULTS**

The proposed Split radix 4-point, 8-point and 16-point FFT is implemented on Zed board. Table II shows the comparison of different split radix FFT algorithms in terms of LUTs occupied and DSP power dissipation.

Table II

Comparison of synthesis results of 4-point, 8-point and 16-point split radix FFT

Design	LUT	DSP	IOB	DSP Power
4-POINT	16576	88	512	79%
8-POINT	51616	308	1024	76%
16-POINT	142304	924	2048	74%

### **V CONCLUSION**

In this paper, we proposed the implementation of split radix FFT of 4-point, 8-point and 16-point. The proposed designs are implemented on Zed board. The comparison results show that the split radix FFT can minimize the power dissipation due the reduction of arithmetic computations in FFT. The proposed designs can be extended to higher point FFT.

### **V REFERENCES**

[1] P. Duhamel and H. Hollmann, "Split radix' FFT algorithm," Electron. Lett., vol. 20, no. 1, pp. 14–16, Jan. 1984.

[2] M. A. Richards, "On hardware implementation of the split-radix FFT," IEEE Trans. Acoust., Speech Signal Process., vol. 36, no. 10, pp. 1575–1581, Oct. 1988.

[3] J. Chen, J. Hu, S. Lee, and G. E. Sobelman, "Hardware efficient mixed radix-25/16/9 FFT for LTE systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 2, pp. 221–229, Feb. 2015.

[4] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," IEEE Trans. Circuits Syst. II,.

[5] D. Cohen, "Simplified control of FFT hardware," IEEE Trans. Acoust., Speech, Signal Process., vol. 24, no. 6, pp. 577–579, Dec. 1976.

[6] X. Xiao, E. Oruklu, and J. Saniie, "An efficient FFT engine with reduced addressing logic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 11, pp. 1149–1153, Nov. 2008.

[7] Z. Qian, N. Nasiri, O. Segal, and M. Margala, "FPGA implementation of low-power split-radix FFT processors," in Proc. 24th Int. Conf. Field Program. Logic Appl., Munich, Germany, Sep. 2014, pp. 1–2.

[8] A. N. Skodras and A. G. Constantinides, "Efficient computation of the split-radix FFT," IEE Proc. F-Radar Signal Process., vol. 139, no. 1, pp. 56–60, Feb. 1992.

[9] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," IEEE Trans. Acoust., Speech Signal Process., vol. 34, no. 1, pp. 152–156, Feb. 1986.

[10] J. Kwong and M. Goel, "A high performance split-radix FFT with constant geometry architecture," in Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE), Dresden, Germany, Mar. 2012, pp. 1537–1542.

[11] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Signal Process., vol. 51, no. 3, pp. 864–874, Mar. 2003.