RESEARCH ARTICLE OPEN ACCESS

A Comprehensive Review of High Speed Low Complexity FFT Processor

Harpreet Kaur
Sri Guru Granth Sahib World University
Jyoti Saroop, Fatehgarh Sahib
Punjab – India

ABSTRACT

The aim of this work is to show the development of a high-speed and low-power-dissipation fast Fourier transforms (FFT) processor The author has been working on the development of the core processing circuit, a butterfly processing circuit. In our previous study, the author designed an integer type butterfly processing circuit. All Simulation of FFT Processor is done on Xilinx Software with VHDL Software. The VHDL simulation serves as a basis for testing complex designs and validating the design prior to fabrication. As a result, the redesign will reduce, the design cycle is shortened, and the product is brought to market sooner. A VHDL program can be considered as a description of a digital system; the associated simulator will use this description to produce behavior that will mimic that of the physical system.

Keywords:- Butterfly processing circuit, Fast Fourier transform (FFT), orthogonal frequency-division multiplexing (OFDM), very large scale integration (VLSI).

I. INTRODUCTION

The FFT processor is a very important component in Orthogonal Frequency Division Multiplexing (OFDM) communication system. In this paper, we studied variable-length radix-8/4/2 FFT architecture for OFDM systems. FFT processor is based on radix-8 FFT algorithm and also supports radix-4 or radix-2 FFT computation. Orthogonal frequency -division multiplexing (OFDM), essentially identical to coded OFDM (COFDM) and discrete multi-tone modulation (DMT), is a frequency - division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub - carrier. Each sub - carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase - shift keying) at a low symbol rate, maintaining total data rates similar to conventional single - carrier modulation schemes in the same bandwidth. OFDM (Orthogonal frequency – division multiplexing) has developed into a popular scheme for wideband digital communication , whether wireless or over copper wires, used in applications such as digital television and audio broadcasting, wireless networking and broadband internet access.

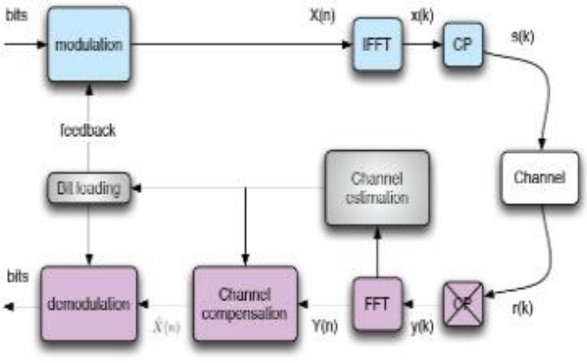


Figure1. OFDMA system structure

FFT

Fast Fourier transform is the key computational blocks in orthogonal frequency division multiplexing system. In order to increase transmission bandwidth or transmission efficiency, an ultra-long-size FFT processor is needed in OFDM system. The system utilizes advanced techniques that increase the flexibility of the broadcast system

II. FFT PROCESSOR ARCHITECTURES

Butterfly Processor

In the context of fast Fourier transform algorithms, a butterfly is a portion of the computation that combines the results of smaller discrete Fourier transforms into a larger DFT, or vice versa. The name "butterfly" comes from the shape of the data-flow diagram in the radix-2 case, as described below. The same structure can also be found in the viterbi algorithm, used for finding the most likely sequence of hidden states. Most commonly, the term "butterfly" appears in the context of the Cooley–Tukey FFT algorithm, which recursively breaks down a DFT of composite size $n = rm$ into r smaller transforms of size m where r is the "radix" of the transform.

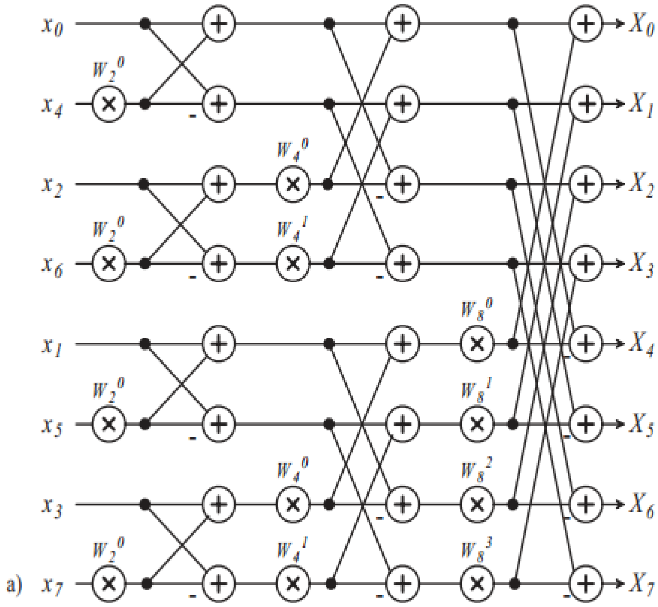


Figure2. Signal flow graph of radix-2 for 8-point FFT Algorithm

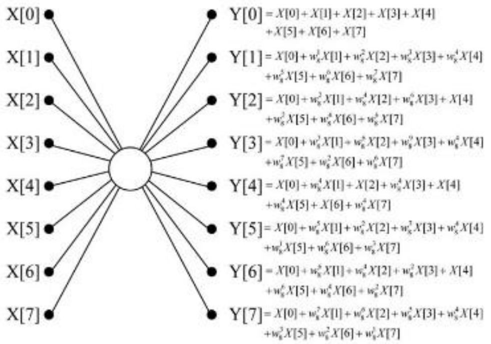
Radix-2

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and conquer approach. We split the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is,

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1), \qquad n = 0,1,\ldots,{}^{N}\!/_{2} - 1$$

Thus $f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2, and hence the resulting FFT algorithm is called a decimation-in-time algorithm.

This Block diagram Proposes and concentrate on the design of Radix 8 FFT and its performance analysis. By using VHDL as a design entity the synthesis and stimulation is done on Xilinx ISE Design Suite 13.2. The Fig [3] shows butterfly unit of Radix – 8 Multiplier and it's The decomposed radix-8 butter y unit in Fig, [4]


X[0]
X[1]
X[2]
X[3]
X[4]
X[5]
X[6]
X[7]
$Y[0] = X[0]+X[1]+X[2]+X[3]+X[4]+X[5]+X[6]+X[7]$
$Y[1] = X[0]+w_8^1X[1]+w_8^2X[2]+w_8^3X[3]+w_8^4X[4]+w_8^5X[5]+w_8^6X[6]+w_8^7X[7]$
$Y[2] = X[0]+w_8^2X[1]+w_8^4X[2]+w_8^6X[3]+X[4]+w_8^2X[5]+w_8^4X[6]+w_8^6X[7]$
$Y[3] = X[0]+w_8^3X[1]+w_8^6X[2]+w_8^1X[3]+w_8^4X[4]+w_8^7X[5]+w_8^2X[6]+w_8^5X[7]$
$Y[4] = X[0]+w_8^4X[1]+X[2]+w_8^4X[3]+X[4]+w_8^4X[5]+X[6]+w_8^4X[7]$
$Y[5] = X[0]+w_8^5X[1]+w_8^2X[2]+w_8^7X[3]+w_8^4X[4]+w_8^1X[5]+w_8^6X[6]+w_8^3X[7]$
$Y[6] = X[0]+w_8^6X[1]+w_8^4X[2]+w_8^2X[3]+X[4]+w_8^6X[5]+w_8^4X[6]+w_8^2X[7]$
$Y[7] = X[0]+w_8^7X[1]+w_8^6X[2]+w_8^5X[3]+w_8^4X[4]+w_8^3X[5]+w_8^2X[6]+w_8^1X[7]$

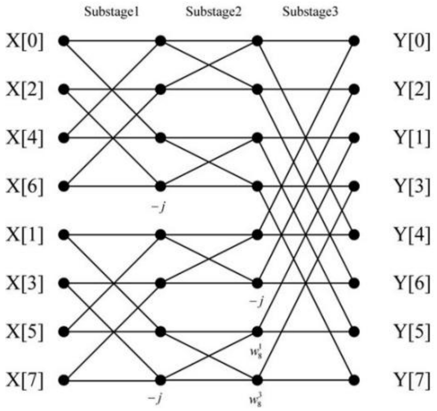


Figure3. Radix-8 butterfly unit

Figure4. Decomposed radix-8 butterfly unit

For most signal processing applications, your far better off designing your system with Radix 2 FFT's in mind (For example, chose your sampling frequency so that a Radix 2 FFT will give you the resolution you want from a spectrum analysis).

There are circumstances where the use of this method can bring worthwhile performance benefits. Sometimes you don't have the flexibility to design your system for radix 2 FFT's. In the worst case, forcing the use of a radix 2 algorithm will double the FFT size required. This loss of performance is compounded in multi-dimensional FFT's. Two dimensional image processing is a good example. Images tend to come in a variety of sizes. The input to each N/4 - Point DFT is a linear combination of four signal sample s scaled by a twiddle factor. This procedure is repeated v times, where v = log4N. The complete butterfly operation for Radix-4 DIF is shown in figure5 (a) and in a more compact form in figure5 (b).

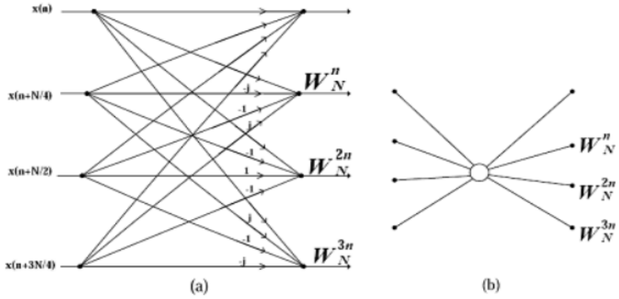


Figure5. Basic Butterfly for radix - 4 DIFFFT

Radix - 8 is another FFT algorithm which increases speed of functioning and this can be achieved by changing the base to 8. It operates on the DFT equation and divides it into eight N/8 point DFTs. The following equations illustrate radix - 8 decimation in frequency (DIF)

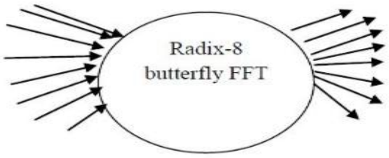


Figure6. Radix-8 butterfly FFT

Pipeline FFT

Pipelining is an important technique used in several applications such as digital signal processing (DSP) systems, microprocessors, etc. It originates from the idea of a water pipe with continuous water sent in without waiting for the water in the pipe to come out. Accordingly, it results in speed enhancement for the critical path in most DSP systems. For example, it can either increase the clock speed or reduce the power consumption at the same speed in a DSP system.

III. SINGLE-PATH DELAY

The single path delay architecture in figure7. It utilizes the delay element more efficiently by sharing the same storage between the outputs and inputs. A single data stream goes through te multiplier in every stage.

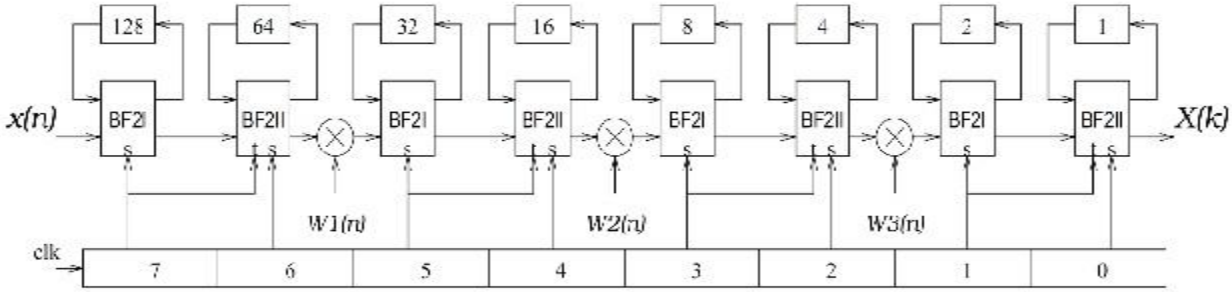


Fig.7. N-point SDF pipelined FFT Processor

IV. MULTI-PATH DELAY

The Multi- path delay architecture is shown in figure 8. Multi-path delay commutated structures are utilized to improve the throughput rate of radix-2 and radix-4 FFT computation by a factor of 2 to 4. Latency can also be reduced by a factor of 2 to 3. Compared with previous radix-2 and radix-4 FFT structures, the proposed high-throughput FFT with doubled throughput rate requires similar or even less hardware cost. Although split radix FFT design is more hardware efficient, the regular structure of proposed FFT structures are attractive for high throughput FFT design.

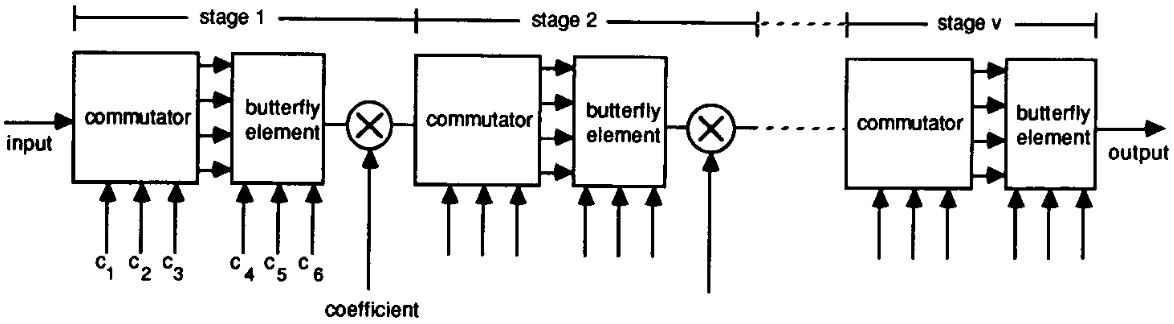


Fig.7. N-point radix-4 pipelined FFT Processor

V. SYSTEM DESIGN

The prototype FFT processor is constructed using coefficient multiplier and adder circuits to synthesize the FFT SFG, The first FFT prototype design was limited to in order to minimize the chance of implementation errors and to validate the proposed approach. Higher order DIT-FFTs can be implemented based on this eight point core in a manner similar to the all-digital FFT approach used in.

VI. CONCLUSION

In this article, two butterfly algorithms -- parallel and dual butterfly algorithms are proposed. The main idea of these two algorithms is to make the operation without multiplications (mainly contains addition operations) and the one with multiplications run in parallel. Because the area that addition units occupy is very small, the FFT processor based on the two butterfly algorithms requires very small areas and has high processing speed. Performance evaluation and practical implementation proved that the FFT processor with these two novel algorithms is suitable for wireless LAN applications. Moreover, it can also be used in other OFDM applications like digital video broadcasting (DVB) and wireless MAN (802.16).

REFRENCES

[1] G. Purna Chandra Rao, B. Ashok and B. Saritha, "Design and Implantation of High Speed FFT Processor for OFDMA System Using FPGA" International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7, July 2013.

[2] Debalina Ghosh and Depanwita Debnath, "FPGA Based Implementation of FFT Processor Using Different Architectures" in IJAITI 2012.

[3] Jarmo Takala and Konsta Punkka, "Jarmo Takala and Konsta Punkka" in Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, FINLAND, jarmo.takala@tut.fi, konsta.punkka@tut.fi.

[4] Miss. Jaishri Katekhaye, Mr. Amit Lamba and Mr. Vipin Bhure , "REVIEW ON FFT PROCESSOR FOR OFDM SYSTEM" in IJAICT Volume 1, Issue 7, November 2014 Doi:01.0401/ijaict.2014.07.14 Published on 05 (1 2) 2014 and ISSN 2348 – 9928.

[5] Jungmin Park, "Design of a radix-8/4/2 FFT processor for OFDM systems" in CPRE/COMS.

[6] C. Yang, J. Vasquez, and J. Chaffee, “Fast direct P(Y)-code acquisition using XFAST,” in Proc. ION GPS-99

[7] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” Math. Comput. , vol. 19, no. 90, pp. 297–297, May 1965.

[8] H. Park, Y. Yamanashi, K. Taketomi, N. Yoshikawa, A. Fujimaki, and N. Takagi, “Novel Serial-Parallel Converter Using SFQ Logic Circuits,” Phys. C Supercond. , vol. 468, no. 15–20, pp. 1977–1982, Sep. 2008.

[9] H. Hara, K. Obata, Y. Yamanashi, K. Taketomi, N. Yoshikawa, M. Tanaka, A. Fujimaki, N. Takagi, K. Takagi, and S. Nagasawa, “Design, Implementation and On-Chip High-Speed Test of SFQ Half-Precision Floating-Point Multiplier,” IEEE Trans. Appl. Supercond. , vol. 19, no. 3, pp. 657–660, Jun. 2009.

[10] T. Kato, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and S. Nagasawa, “60-GHz Demonstration of an SFQ Half-precision Bitserial Floating-point Adder Using 10 kA/cm2 Nb Process,” 2013 IEEE 14th Int. Supercond. Electron. Conf., pp. 1 –3, Jul. 2013.

[11] Z. J. Deng, N. Yoshikawa, S. R. Whiteley, and T. Van Duzer, "DataDriven Self-Timed RSFQ High Speed Test System,” IEEE Trans. Appl. Superconductivity, vol. 7, pp. 3830–3833, Dec. 1997.