High Performance Pipelined Design for FFT Processor based on FPGA

A.A. Raut[1], **S. M. Kate**[2]

[1]Sinhgad Institute of Technology, Lonavala, Pune University, India

[2]Sinhgad Institute of Technology, Lonavala, Pune University, India

Abstract: ***It is important to develop a high-performance FFT processor to meet the requirements of real time and low cost in many different systems. So a radix-2 pipelined FFT processor based on Field Programmable Gate Array (FPGA) for Wireless Local Area Networks (WLAN) is proposed. Unlike being stored in the traditional ROM, the twiddle factors in our pipelined FFT processor can be accessed directly. This paper concentrates on the development of the Fast Fourier Transform (FFT), based on Decimation-In-Time (DIT) domain, Radix-2 algorithm, this paper uses VHDL as a design entity, and their Synthesis by Xilinx Synthesis Tool on SPARTAN kit has been done. The input of Fast Fourier transform has been given by a PS2 KEYBOARD using a test bench and output has been displayed using the waveforms on the Xilinx Design Suite 12.1.The synthesis results show that the computation for calculating the 32-point Fast Fourier transform is efficient in terms of speed. The implementation was made on a Field Programmable Gate Array (FPGA) because it can achieve higher computing speed than digital signal processors, and also can achieve cost effectively ASIC-like performance with lower development time, and risks. This results show that the processor achieves higher throughput and lower area and latency.***

Keywords: FFT; radix-2; FPGA; Butterfly; VHDL.

1. Introduction

This paper proposes the design of 32-points FFT processing block. The work of the project is focused on the design and implementation of FFT [13] for a FPGA kit. This design computes 32-points FFT and all the numbers follow fixed point format of the type Q8.23, signed type input format is used. The direct mathematical derivation method is used for this design.

In this project the coding is done in VHDL [3][8] & the FPGA synthesis and logic simulation is done using Xilinx ISE Design Suite 12.1. The Discrete Fourier Transform (DFT) [5][7] plays an Important role in the analyses, design and implementation of the discrete-time signal-processing algorithms and systems it is used to convert the samples in time domain to frequency domain. The Fast Fourier Transform (FFT) is simply a fast (computationally efficient) way to calculate the Discrete Fourier Transform (DFT). The wide usage of DFT's in Digital Signal Processing applications is the motivation to Implement FFT's. Almost every branch of engineering and science uses Fourier methods. The words "frequency," "period," "phase," and "spectrum" are important parts of an engineer's vocabulary. The Discrete Fourier transform is used to produce frequency analysis of discrete non periodic signals. The FFT is another method of achieving the same result, but with less overhead involved in the calculations [1].

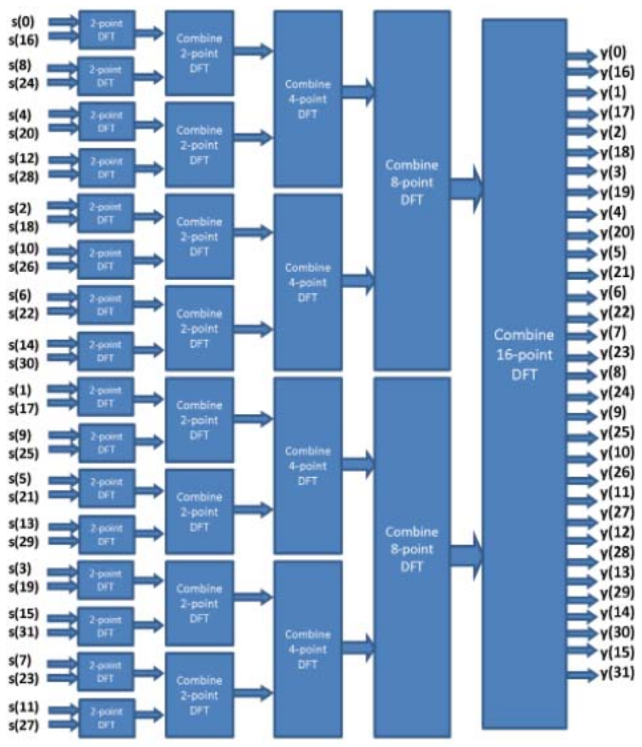


Figure 1: Radix-2 Decimation in Time Domain FFT Algorithm for Length of 32 Signals

Transforms basically convert a function from one domain to another with no loss of information. Fourier Transform converts a function from the time (or spatial) domain to the frequency domain. The mathematical formula used for the Fourier transform is as follows

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

DFT is identical to samples of the Fourier transform at equally spaced frequencies. Consequently, computation of the N-point DFT corresponds to the computation of N samples of the Fourier transform at N equally spaced frequencies ┐ k = 2└ k/N. Considering input x[n] to be complex, N complex multiplications and (N-1) complex additions are required to compute each value of the DFT, if computed directly from the formula given as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad 0 \le k \le N-1$$

$$W_N = e^{-j2\pi/N}$$

To compute all N values therefore requires a total N2 complex multiplications and N (N-1) complex additions. Each complex multiplication requires four real multiplications and two real additions and each complex addition requires two real additions. Therefore a total of 4N2 real multiplications and N (4N-2) real additions are required. Besides these multiplications and additions there should be provision for storing N complex input sequences and also to store N output values. Contrary to this by using Decimation in Time [11] FFT radix-2 algorithm the number of complex multiplications and additions will be reduced to (N/2) log2N and Nlog2N to compute the DFT of a given complex x[n]. Hence in this project the Decimation in Time FFT radix-2 algorithm is implemented to compute the DFT of a sequence.

2. Radix-2 DIT FFT Algorithm

The radix-2 algorithms are the simplest FFT algorithms. The decimation-in-time (DIT) radix-2 FFT recursively partitions a DFT into two half-length DFTs [13] of the even-indexed and odd-indexed time samples. The outputs of these shorter FFTs are reused to compute many outputs, thus greatly reducing the total computational cost. The radix-2 decimation-in-time and decimation-in-frequency fast Fourier transforms (FFTs) are the simplest FFT algorithms. Like all FFTs, they gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs.

The radix-2 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: a sum over the even-numbered discrete-time indices n=[0,2,4,…,N┘ 2] and a sum over the odd-numbered indices n=[1,3,5,…,N┘ 1] .

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) e^{-\left(i\frac{2\pi nk}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi\times(2n)k}{N}\right)} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} + e^{-\left(i\frac{2\pi k}{N}\right)} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} \\
&= \mathrm{DFT}_{\frac{N}{2}}[[x(0),x(2),\ldots,x(N-2)]] + W_N^k \mathrm{DFT}_{\frac{N}{2}}[[x(1),x(3),\ldots,x(N-1)]]
\end{aligned}$$

This is called decimation in time because the time samples are rearranged in alternating groups and a radix-2[5] algorithm because there are two groups. A basic butterfly [13] operation is shown in Figure 2, which requires only N2 twiddle-factor multiplies per stage.

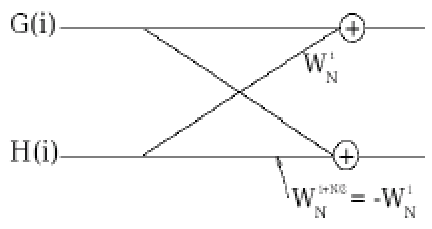


Figure 2: Basic butterfly computation in the decimation-in-time FFT algorithm.

The same radix-2 decimation in time can be applied recursively to the two length N2 DFTs to save computation. When successively applied until the shorter and shorter DFTs reach length-2, the result is the radix-2 DIT FFT algorithm.

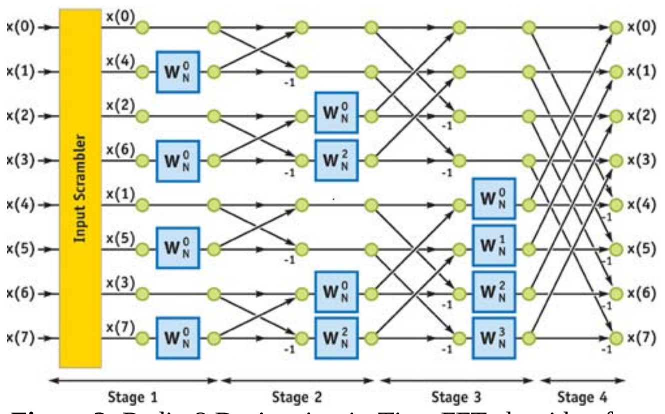


Figure 3: Radix-2 Decimation-in-Time FFT algorithm for a length- 8 signal

3. FPGA

The introduction of field programmable gate arrays (FPGAs), has made it feasible to provide hardware for application specific computation design. The changes in designs in FPGA's can be accomplished within a few hours, and thus result in significant savings in cost and design cycle. FPGAs offer speed comparable to dedicated and fixed hardware systems for parallel algorithm. the 32-point FFT proposed in this paper is been simulated and synthesized using the Xilinx design suite 12.1 with the device family as Spartan 3 (low power). The summary of the device description of the vertex FPGA used is explained in the table below

Table 1. Summary of FPGA Features

Device Family	*Spartan 3*
Device	XC3S200
Package	vq100
Speed Grade	-5

This is called decimation in time because the time samples are rearranged in alternating groups and a radix-2 the features of the Spartan 3 FPGA [2] used in this proposed work with Xilinx Design Suite 12.1 are listed in the table below.

Table 2: Summary Of Spartan 3 Features

Features	*Spartan 3*
System gates	200K
Equivalent logic cells	4320
CLB array	480
Distributed RAM bits	30K
Blocked RAM Bits	216K
Dedicated Multipliers	12
DCMs	4
Maximum User I/O	173
Maximum Differential I/O Pairs	76

4. Software Simulation and Results

The proposed FFT block of signal length 32 is been simulated and synthesized using the Xilinx Design Suite 12.1. The RTL block thus obtained for the decimation in time domain radix -2 Fast Fourier transform algorithm is shown

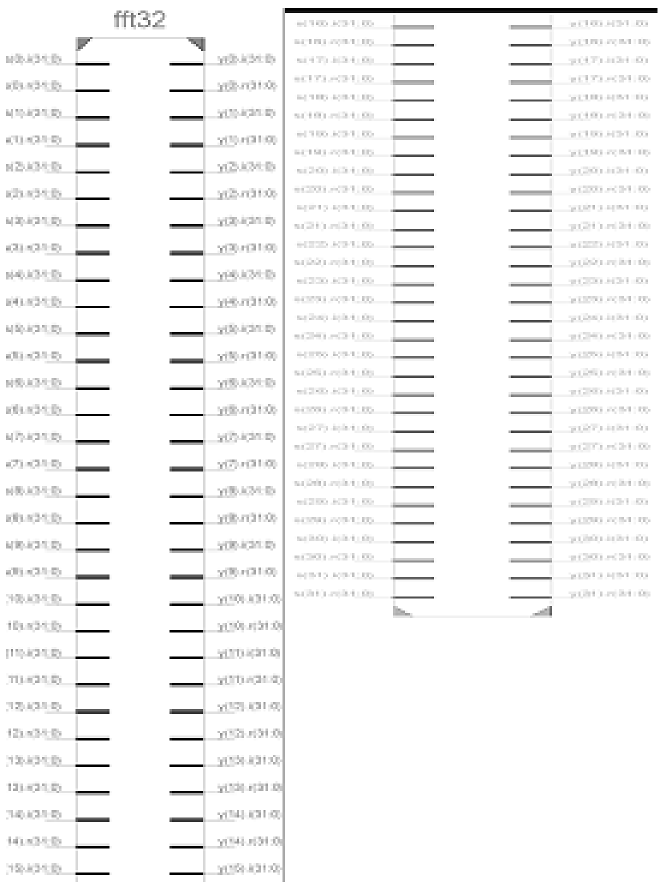


Figure 4: RTL view of a 32-point FFT

The RTL view of the butterfly structure obtained after the simulation of the 32-point FFT block, Decimation in time domain[11] is shown next and also the internal architecture of the butterfly block is shown.

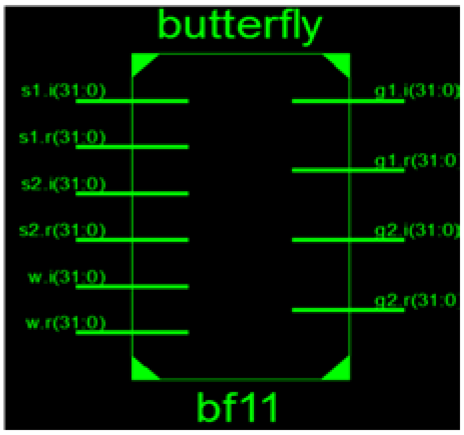


Figure 5: RTL View of a Butterfly Component Used In 32-Point FFT

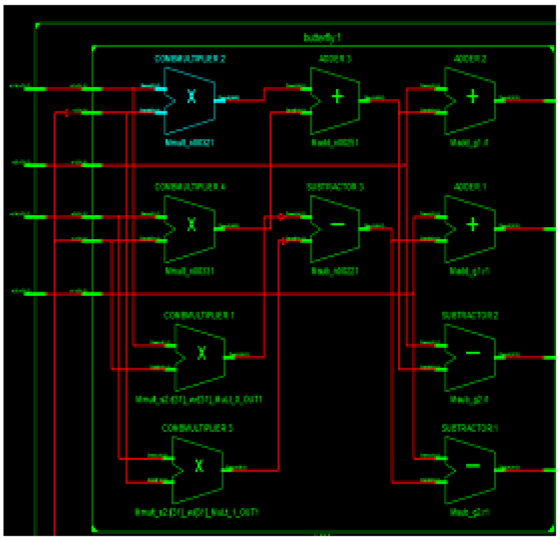


Figure 6: Internal Architecture of The Butterfly Component

The next are shown the simulation results of the 32-point FFT block. The s is the given simulation result is the applied input, its 32 complex numbers, y is the output in binary format while y1 is same as y but its in "real" format, so we can easily see the outputs in waveform.

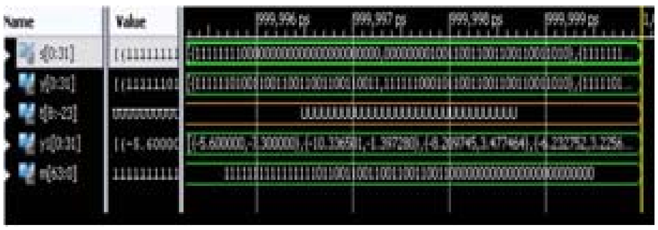
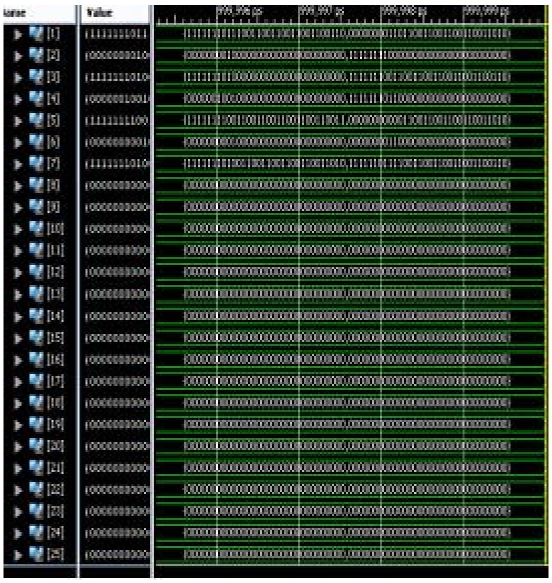
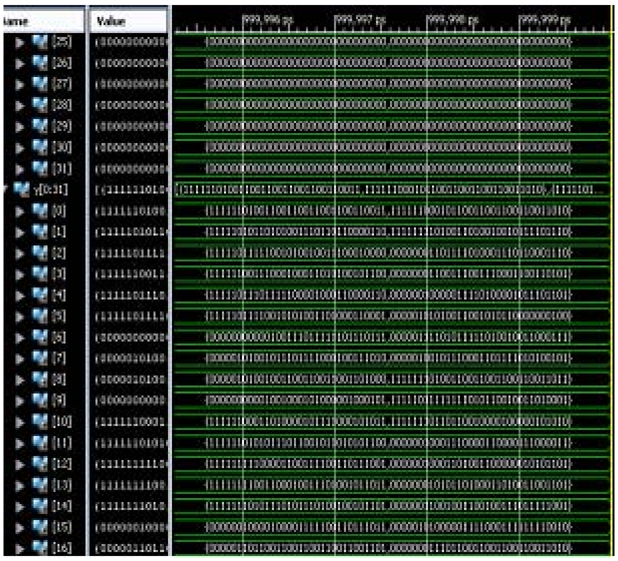
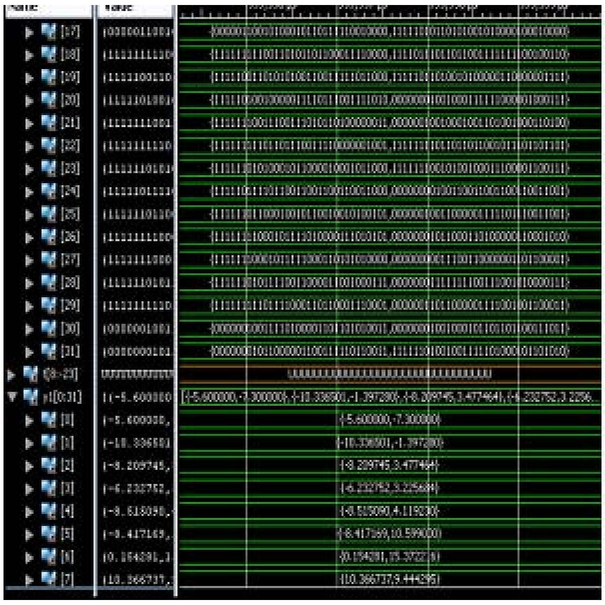
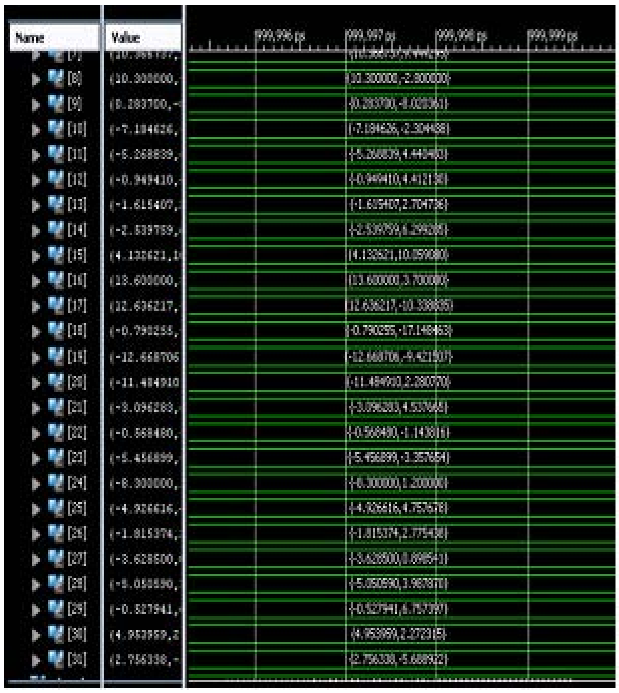


Figure 7: Simulation results of the 32-point FFT









The final synthesis report is shown using the table below

Table 3: Summary Oo Spartan 3 Features Used In The 32-Point Fast Fourier Transform

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slices	1646	1920	85%	
Number of 4 input LUTs	2884	3840	75%	
Number of bonded IOBs	892	63	1415%	Resource overuse
Number of MULT18X18s	12	12	100%	

Table4: Shows The Time Delay Of 32-Point Fft(Radix-2 Decimation In Time Domain)

Min. delay	42.58ns (22.515ns logic, 20.003ns route) (53.0% logic, 47.0% route)
Total REAL time to Xst completion: 20.00 secs Total CPU time to Xst completion: 19.77 secs Total memory usage is 219856 kilobytes	

References

[1] Sneha N.kherde, Meghana Hasamnis, “Efficient Design and Implementation of FFT”, International Journal of Engineering Science and Technology (IJEST), ISSN : 0975- 5462 NCICT Special Issue Feb 2011

[2] Ahmed Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladawy,“Efficient FPGA implementation of FFT/IFFT Processor”,INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING, Issue 3, Volume

[3] 3, 2009

[4] Hardware Description Language. URL: http://en.wikipedia.org/wiki/Hardware_description_language

[5] Very High Speed Integrated Circuit Hardware Description Language.URL: http://electrosofts.com/vhdl/

[6] Alan V. Oppenheim, Ronald W. Schafer with John R. Buck, Discrete Time Signal Processing, Second Edition

[7] B. Parhami, Computer Arithmetic, Algorithms and Hardware Designs, 1999
[8] James W. Cooley and John W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series
[9] Peter J. Ashenden, The Designer's Guide to VHDL, Second Edition.
[10]N. Weste, M. Bickerstaff, T. Arivoli, P.J. Ryan, J. W. Dalton, D.J. Skellern", and T.M. Percivalt A 50 Mhz 16Point-FFT processor for WLAN applications.
[11]Saad Bouguezel, M. Omair Ahmad,"IMPROVED RADIX- 4 AND RADIX-8 FFT ALGORITHMS"IEEE. Department of Electrical and Computer Engineering Concordia University 1455 de Maisonneuve Blvd. West Montreal, P.Q., Canada.
[12]Ali Saidi ," DECIMATION-IN-TIME-FREQUENCY FFT ALGORITHM" Motorola Applied Research, Paging and Wireless Data Group Boynton Beach.
[13]Rizalafande Che Ismail and Razaidi Hussin "High Performance Complex Number Multiplier Using Booth-Wallace Algorithm" School of Microelectronic Engineering Kolej University Kejuruteraan Utara Malaysia.
[14]Bergland, G. D. "A Guided Tour of the Fast Fourier Transform." IEEE Spectrum 6, 41-52, July 1969.

Author Profile

Atul Raut received the B.E. degrees in Electronics and Telecommunications Engineering from Sinhgad Institute of Technology in 2009. Currently he is doing M.E. in same institute.

Prof. Sandeep M. Kate have completed ME Electronics and working as Assistant Professor in Sinhgad Institute of Technology, Lonavala, Pune University, India. He guided several ME students for project.