IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 1 Ver. III (Jan. 2014), PP 42-50
www.iosrjournals.org

Designing and Simulation of 32 Point Fft Using Radix-2 Algorithm for Fpga

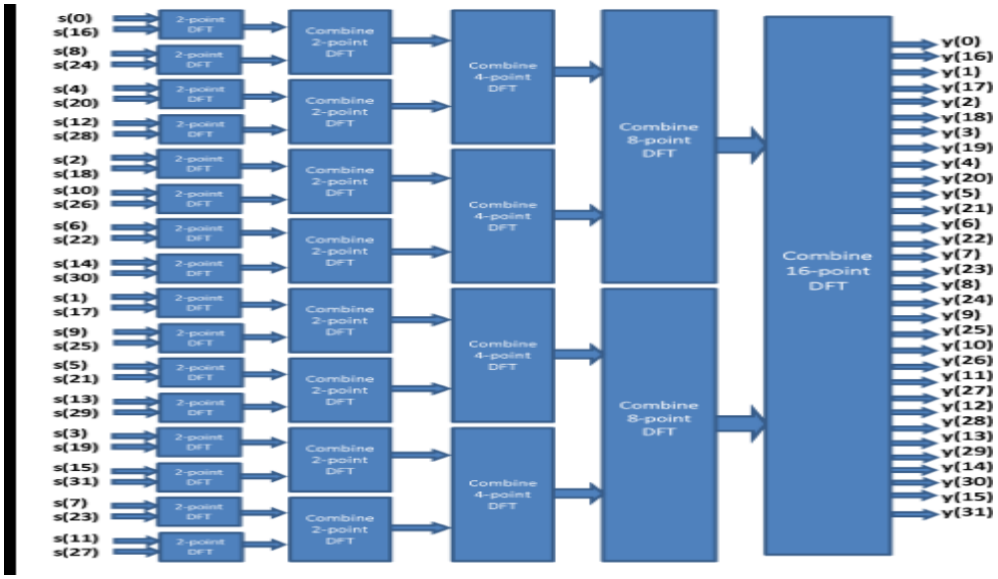
Afreen Fatima
Department : Electronics Affiliated To : JNTU (HYD)

Abstract: *The Fast Fourier Transform (FFT) is one of the rudimentary operations in field of digital signal and image processing. Some of the applications of the fast Fourier transform include Signal analysis, Sound filtering, Data compression, Partial differential equations, Multiplication of large integers, Image filtering etc. Fast Fourier transform (FFT) is an efficient implementation of the discrete Fourier transform (DFT).*
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 VERILOG as a design entity. The input of Fast Fourier transform has been given by a keyboard using a test bench and output has been displayed using the waveforms on the Xilinx Design Suite 13.1 and synthesis results in Xilinx show that the computation for calculating the 32-point Fast Fourier transform is efficient in terms of speed.

I. Introduction

This proposes the design of 32-points FFT processing block. The work of the project is focused on the design and implementation of FFT 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 [8] & the FPG synthesis and logic simulation is done using Xilinx ISE Design Suite 13.1. The Discrete Fourier Transform (DFT) 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. Fig1:



Radix-2 Decimation in Time Domain FFT Algorithm For LENGTH OF 32 POINT SIGNAL

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 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.

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 evenindexed 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] .

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

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

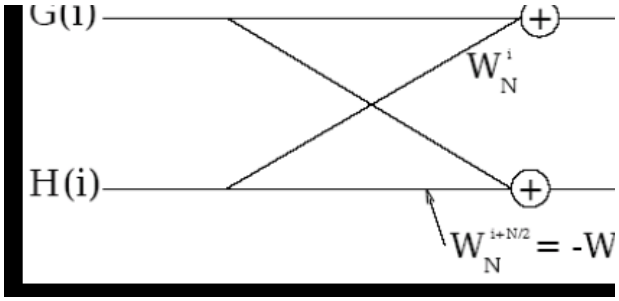
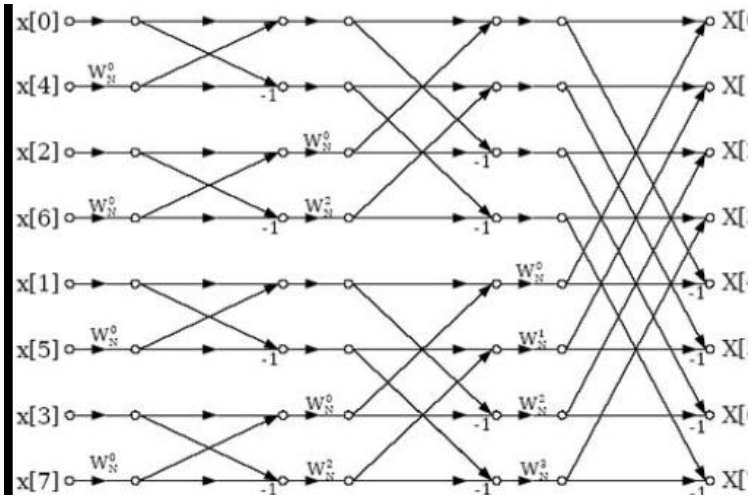


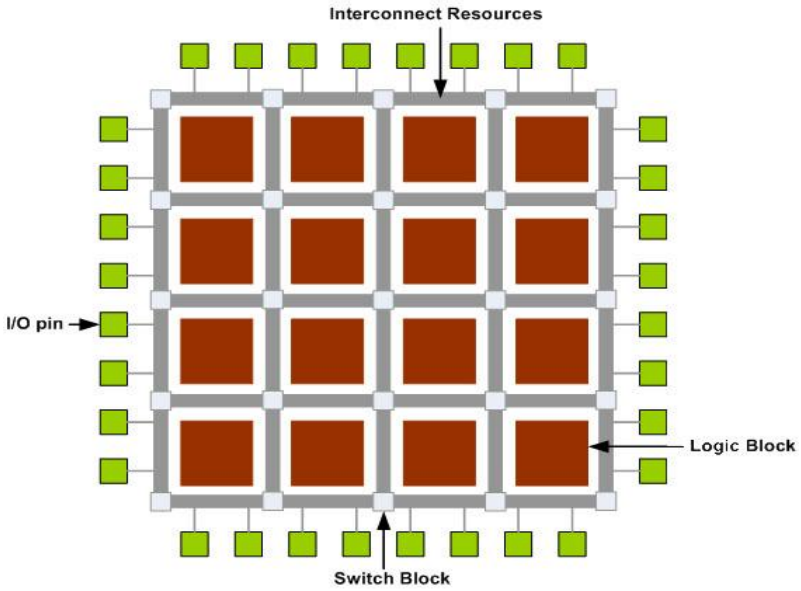
Fig 2: Basic butterfly computation in the decimation-in-time FFT algorithm.

The same radix-2 decimation in time can be applied recursively to the two length *N*2 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.



Radix-2 Decimation-in-Time FFT algorithm for a length-8 signalFPGA

FPGA contains a two dimensional arrays of logic blocks and interconnections between logic blocks. Both the logic blocks and interconnects are programmable. Logic blocks are programmed to implement a desired function and the interconnects are programmed using the switch boxes to connect the logic blocks. To be more clear, if we want to implement a complex design (CPU for instance), then the design is divided into small sub functions and each sub function is implemented using one logic block. Now, to get our desired design (CPU), all the sub functions implemented in logic blocks must be connected and this is done by programming the interconnects.



The introduction of field programmable gate arrays (FPGAs), has made it feasible to provide hardware forapplication 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 synthesised using the xilinx design suite 12.1 with the device family as Vertex 6 (low power).the summary of the device description of the vertex FPGA used is explained in the table below

II. Summary Of Fpga Features

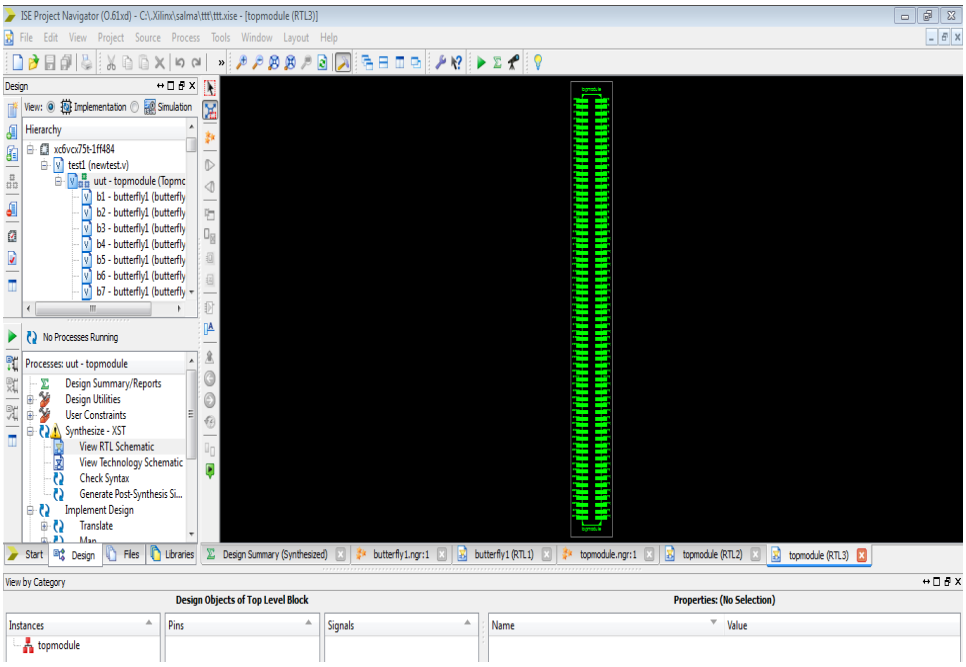
Device Family	**Vertex 6**
Device	XC6VLX75TL
package	FF484
Speed grade	-1L

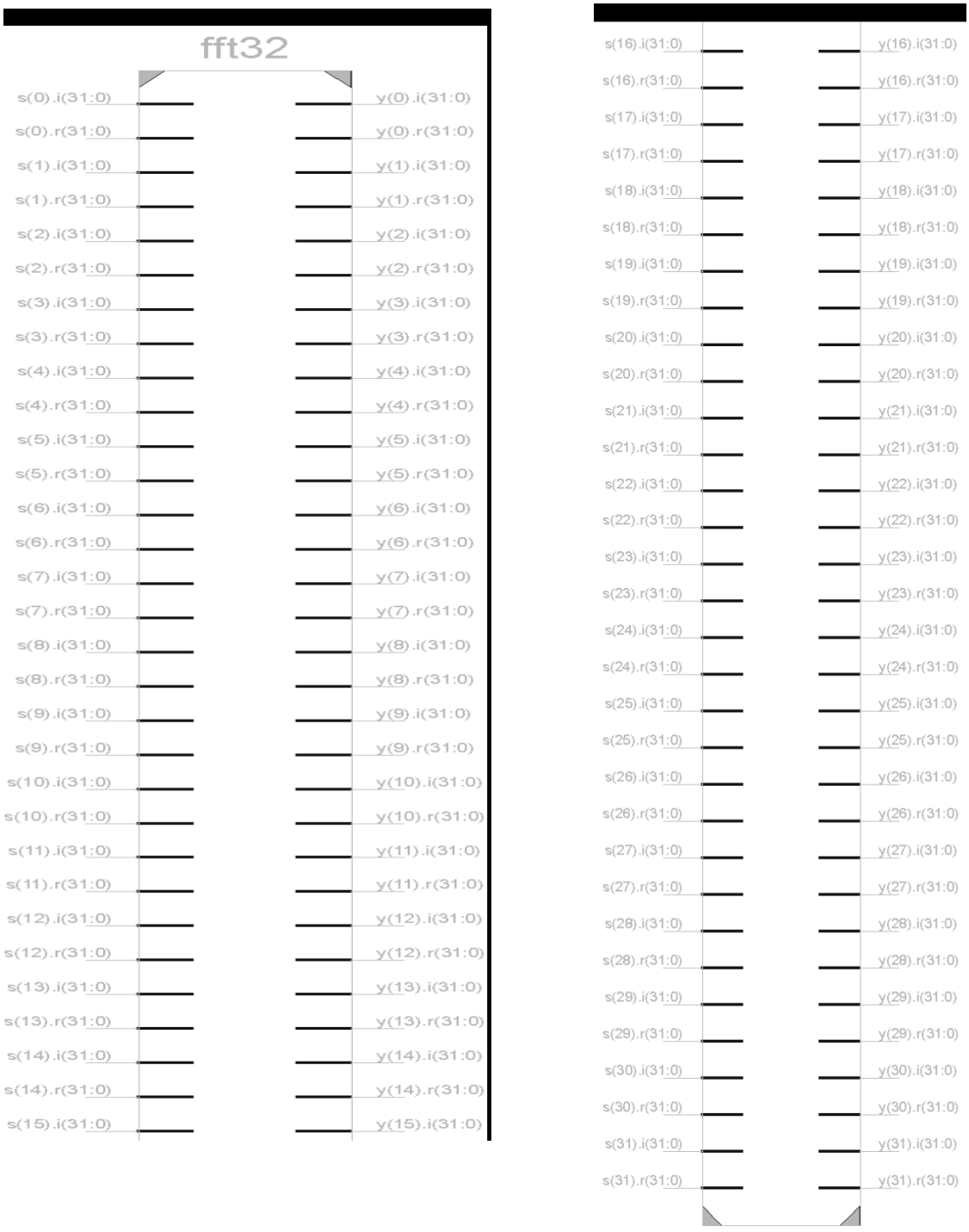
TABLE II. SUMMARY OF VERTEX 6 FEATURES

Features	**Virtex-6**
Logic cells	760,000
BlockRAM	38Mb
DSP slices	2,016
DSP performance (symmetric FIR)	2.419GMACS
Transceiver count	72
Transceiver speed	11.18Gb/s
Total tranceiver	536Gb/s
Agile mixed signal (AMS)/XADC	1,066Mb/s
Configuration AES	Gen2x8
	Yes
	Yes
i/o pins	1,200
i/o voltage	1.2v,1.5v,1.8v,2.5v
EasyPath Cost Reduction Solution	yes

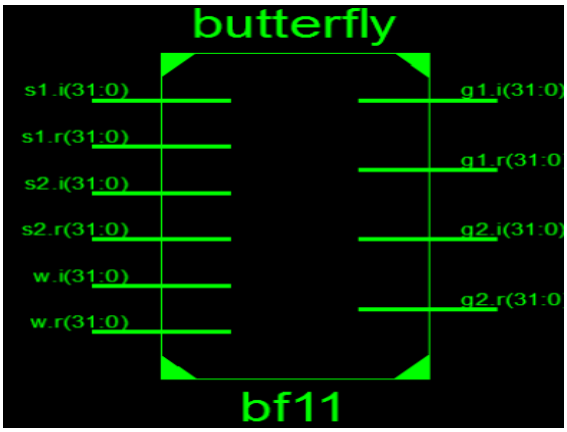
III. Software Simulation And Results

The proposed FFT block of signal length 32 is been simulated and synthesised using the Xilinx Design Suite 13.1. The RTL block thus obtained for the decimation in time domain radix -2 Fast Fourier transform algorithm is shown

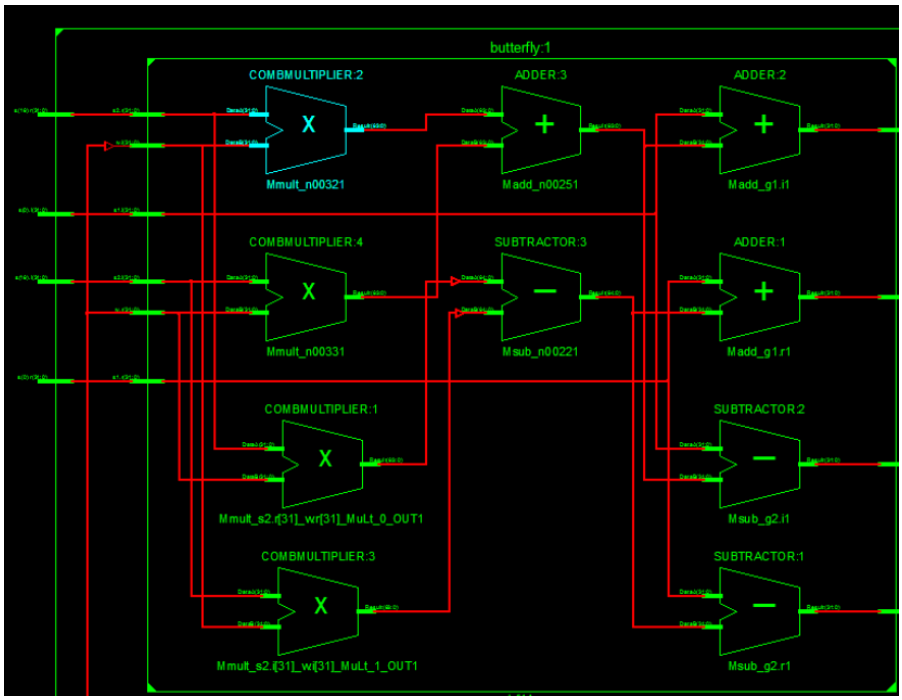




The RTL view of the butterfly structure obtained after the simulation of the 32-point FFT block, Decimation in time domain is shown next and also the internal architechture of the butterfly block is shown. Fig

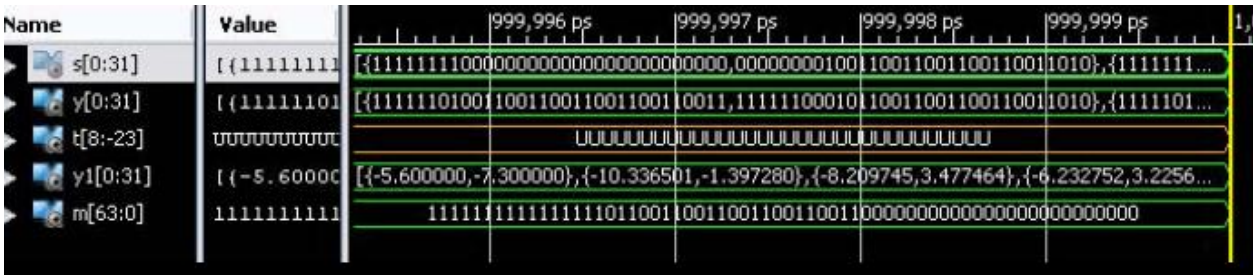


RTL View Of A Butterfly Component Used In 32-Point FFT

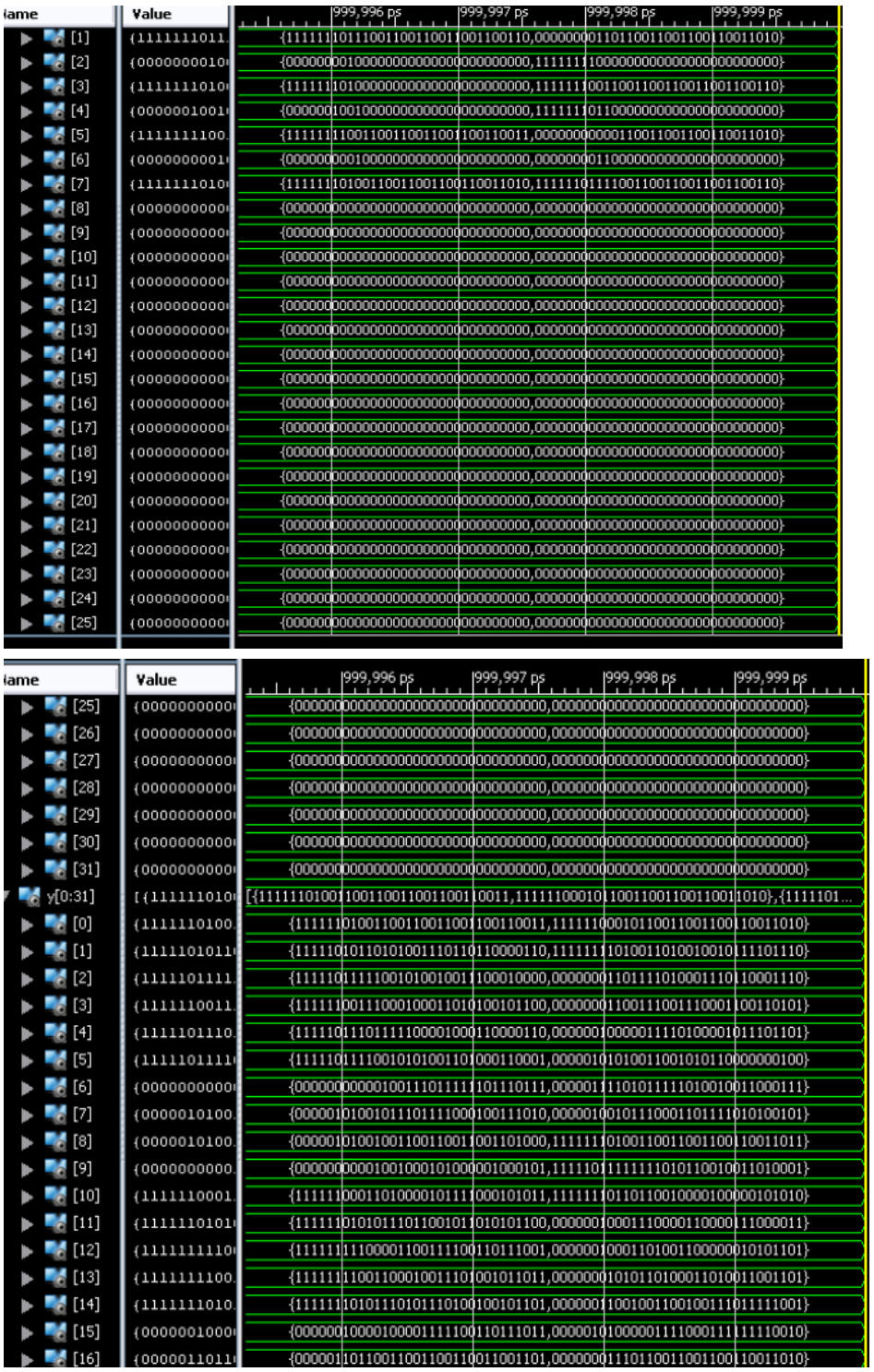


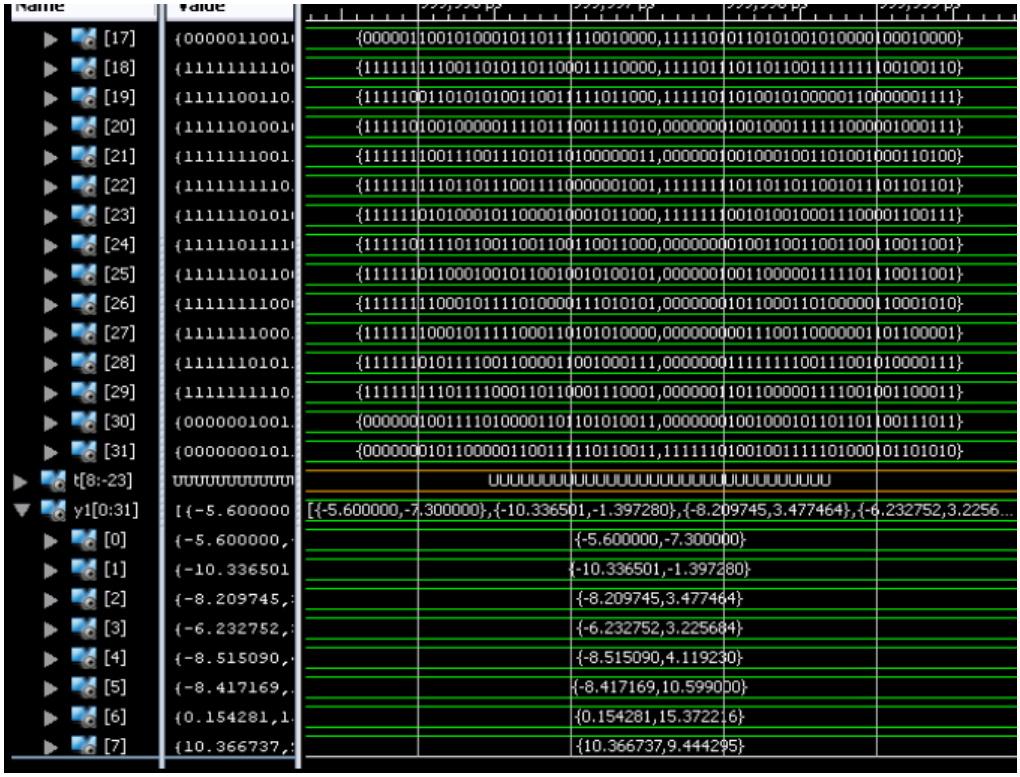
Internal Architecture of The Butterfly Component

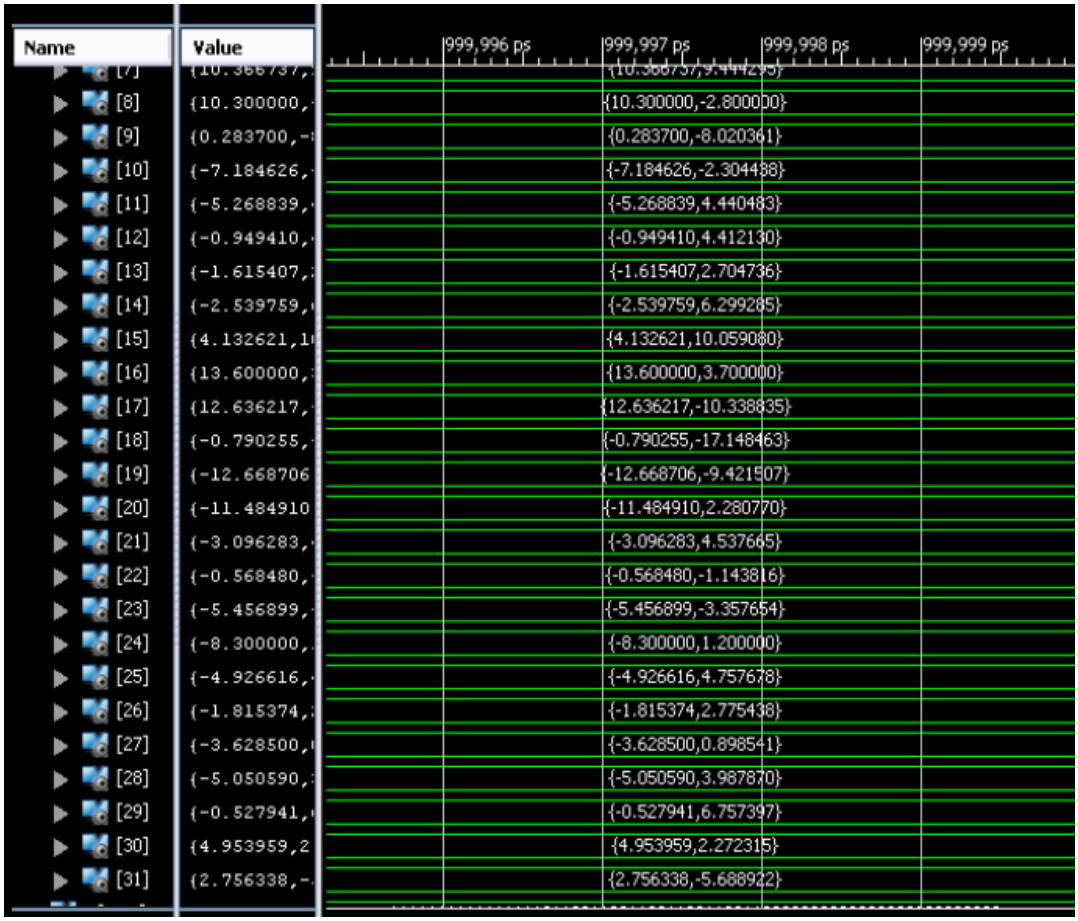
The next are shown the simulation results of the 32-point FFT block.The s is the given simulation resul is the applied input, its 32 complex numbers.y is the output in binary format while y1 is same as y but its in "real" format, so we can easily see the outputs in waveform.



Simulation results of the 32-point FFT


Value
999,996 ps
999,997 ps
999,998 ps
999,999 ps
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]
[10]
[11]
[12]
[13]
[14]
[15]
[16]
[17]
[18]
[19]
[20]
[21]
[22]
[23]
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
y[0:31]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]
[10]
[11]
[12]
[13]
[14]
[15]
[16]





The final synthesis report is shown using the table bel

APPLICATIONS OF FAST FOURIER TRANSFORM

- Spectrum analysis – used for analysing and detecting signals
- Coding – audio and speech signals are often coded in the frequency domain using FFT variants (MP3, …)
- Another recent application is in a modulation scheme called OFDM, which is used for digital TV broadcasting (DVB) and digital radio (audio) broadcasting (DAB).
- Background noise reduction for mobile telephony, speech and audio signals is often implemented in the frequency domain using FFTs

IV. Conclusion

The result demonstrated that radix 2 FFT method uses least no. of slices and requires less computations.Modeling and Hardware description of 32 bit FFT using radix 2 FFT algorithm by verilog hardware description language and realization of this on Xilinx FPGA chip was proposed

References

[1] Asmita Haveliya , ‘’Design and simulation of 32 point FFT using Radix 2 Algorithm For FPGA Implementation’’ .
[2] 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
[3] 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, 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] 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.
[7] Ali Saidi ,” DECIMATION-IN-TIME-FREQUENCY FFT ALGORITHM” Motorola Applied Research, Paging and Wireless Data Group Boynton Beach.
[8] Comparitive Analysis of FFT Algorithm By, Aravind Ganapathiraju, Jonathan Hamaker, Joseph Picone, Anthony Skjellum.
[9] Implementing FFT Algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.11, November 2011 .
[10] Text Book Of Digital Signal Processing Fourth Edition– P.Ramesh Babu.
[11] Digital Signal Processing First Edition – J.S.Chitode