IMPLEMENTATION OF FAST FOURIER TRANSFORMATION ON FPGA

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

By

Nishant Singh (141070)

Gaurav Verma (141076)

Vishavjit Sharma (141081)

UNDER THE GUIDANCE OF
Dr. Harsh Sohal

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
Month-May, Year-2018

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

(Established by H.P. State Legislative vide Act No. 14 of 2002)
P.O. Waknaghat, Teh. Kandaghat, Distt. Solan - 173234 (H.P.) INDIA
Website: www.juit.ac.in
Phone No. (91) 01792-257999
Fax: +91-01792-245362

CERTIFICATE

This is to certify that the work reported in the B.Tech project report entitled **"IMPLEMENTAION OF FFT ON FPGA"** which is being submitted by Nishant Singh (141070), Gaurav Verma(141076), Vishavjit Sharma(141081) in fulfillment for the award of Bachelor of Technology in Electronics and Communication Engineering by the Jaypee University of Information Technology, is the record of candidate's own work carried out by him/her under my supervision. This work is original and has not been submitted partially or fully anywhere else for any other degree or diploma.

Dr. Harsh Sohal
Assistant Professor (Senior Grade)
Department of Electronics & Communication Engineering
Jaypee University of Information Technology, Waknaghat

TABLE OF CONTENTS

DECLARATION BY THE SCHOLARS

We hereby declare that the work reported in the B-Tech thesis entitled **"IMPLEMENTAION OF FFT ON FPGA"** submitted at **Jaypee University of Information Technology, Waknaghat India,** is an authentic record of our work carried out under the supervision of **Dr**. Harsh Sohal. We have not submitted this work elsewhere for any other degree or diploma.

Nishant Singh

Gaurav Verma

Vishavjit Sharma
Department of Electronics and Communication Engineering
Jaypee University of Information Technology, Waknaghat, India

ACKNOWLEGDEMENT

I owe my profound gratitude to my project supervisor **Dr. Harsh Sohal**, who took keen interest and guided me all along in my project work titled **—”IMPLEMENTATION OF FAST FOURIER TRANSFORMATION ON FPGA**” until the completion of my project by providing all the necessary information for developing the project. The project development helped me in research and I have to know many new things in my domain. I really thank him for the constant unstinted support and invaluable guidance throughout the completion of my project.

LIST OF ABBREVIATIONS

FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
FPGA	Field Programmable Gate Array
CPLD	complex programmable basis device

LIST OF FIGURES

LIST OF TABELS

ABSTRACT

The Fast Fourier Transform (FFT) algorithm now play in important role in the design of digital signal processing system for communications, measurement and control applications. Although the algorithm was developed for use on general- purpose computers, or special purpose microprocessor (DSP μPs). Semiconductor technology now make it economically feasible to implement FFT using field programming grids array (FPGA) technique. In this paper 8 point FFT algorithm was implemented using FPGA technique type (Xilinx) with the aid of schematic software package.

The equipment portrayal and demonstrating of advanced flag handling (DSP) calculations and applications for actualizing on Field Programmable Gate Array (FPGA) chips are testing issues. In this paper, some functional Fast Fourier Transform (FFT) calculations including Radix-2 are demonstrated by Verilog equipment portrayal dialect and their execution are thought about regarding chip zone use and most extreme recurrence activity.

CHAPTER 1
INTRODUCTION

The Fourier change changes over data from the time area into the recurrence space. It is an essential investigative instrument in such differing fields as acoustics, optics, seismology, broadcast communications, and discourse.

Flag preparing, and picture handling. The FFT is the nonexclusive name for a class of computationally effective calculations and are broadly utilized as a part of the field of advanced flag preparing. FFT can be executed by utilizing distinctive procedures. One of the advanced and new strategies is FPGA approach. FPGA as it is more broadly called is a sort of programmable gadget. Programmable gadget are a class of general – reason chips that can be arranged for a wide assortment of uses.

There are two primary fundamental equipment outline approach at present accessible for FPGA procedure: Schematic and dialect based plans. With the end goal of this paper, the first was picked because of this plan gives better streamlining usage as far as both zone and speed. In this paper, to begin with, fundamental idea of FPGA method will be portrayed in subtle elements, which will cover the vital subjects that required in the application outline, the plan strategies for FPGA procedure will be displayed then an utilization of usage of 8-focuses FFT calculation will be given its outcomes.

CHAPTER 2

SOFTWARE

2.1 Software Requirement and specification

2.1.1 Minimum Hardware Requirement specification:

1. Intel Pentium processor
2. 256 MB RAM
3. 20GB HDD
4. LAN Card for LAN connection

2.1.2 Minimum Software Requirement specification:

1. Xilinx ISE simulator
2. Software Verilog HDL

2.2 Hardware Description Language (HDL)

In devices, a hardware delineation vernacular t from a class of codings, detail lingo, or showing tongues for formal depiction and plan of electronic circuits, and most-ordinarily, mechanized reason. It can depict the circuit's undertaking, its framework and affiliation, and tests to check its movement at any level. HDL plays out the customized translation of plan depiction into a course of action of method of reasoning condition. It is used to depict the outline and lead of discrete electronic system.

2.2.1 Benefits of HDL

1) We can check outline usefulness ahead of schedule in the plan composed as a HDL.
2) Reduced non-repeating designing expenses.
3) Design reused is empower.
4) Increase adaptability to configuration changes.
5) Better and simpler outline check. Hardware/software co-design.

2.2.2 Types of HDL

There are two types of hardware description language (HDL):

1) VHDL – VHSIC Hardware description language
2) VHSIC –Very High Speed Integrated Circuit

Verilog is other main HDL
Primarily targeted for design of ASICs
Developed by Cadence Data systems and later transferred to a consortium called
Open Verilog International (OVI)
ASIC – Application Specific Integrated Circuit.

2.3 Verilog

Verilog means Verifying Logic. It was developed by Phil Moor as a simulation language. In 1989, Cadence Design Systems bought it after that it is also used as a synthesis language. It was an exclusive dialect, being the property of Cadence Design Systems. In the late 1980's it appeared to be obvious that originators would have been moving far from restrictive dialects like n speck, Hilo and Verilog towards the US Department of protection standard VHDL. In this way, Cadence Design Systems opened Verilog to open in 1990.It turns into an IEEE standard named as IEEE-1364 out of 1995 and after that reexamined in 2000 and till now it is an IEEE standard.

The Verilog Hardware Description Language (HDL) is utilized to depict an equipment outline and a test system that considers testing the rationale of an equipment plan. Therefore, the originator can precisely reenact the activity of a wide assortment of advanced systems, for example, combinatorial rationale, consecutive systems and no concurrent systems.

A portrayal of an equipment configuration utilizing the Verilog HDL is known as a Verilog show and is the fundamental building obstruct for recreation. This model can depict both the capacity of a plan and the parts and associations of segments. Along these lines Verilog HDL is both a social and an auxiliary dialect .Verilog models can be produced at a few levels of deliberation. At the most abnormal amount, the originator can create algorithmic that execute a plan as a calculation in an abnormal state dialect like the C programming dialect. The following lower level is build up a model that depicts the stream of information amongst registers and how a plan forms that information. This kind of demonstrating utilizes the dialect which is called Register Transfer Language (RTL). The fashioner can grow more solid models at the entryway level that depict the individual rationale doors and the associations between rationale entryways in an outline. At long last, the creator can grow exceptionally solid models at the switch-level that portray the individual transistors and capacity hubs in a gadget and the associations between them.

2.4 Difference between Verilog and VHDL:

We can use any of hardware description language (HDL) to design a circuit, but there is some difference between Verilog and VHDL.

S.NO.	Verilog	VHDL
1	Depends on C	Depends on ADA
2	Case-sensitive	Case-insensitive
3	Easy to learn	Difficult to learn
4	More constructs are synthesizable	Less constructs are synthesizable
5	Data types are only built in	Data types are only built in
6	No packages are required	Required for functions, constants, types etc.
7	Automatic conversion and declaration	Conversion functions and all signals must be declared.

Table 2.1 – Difference between Verilog and VHDL

2.5 Xilinx Integrated Software Environment (ISE)

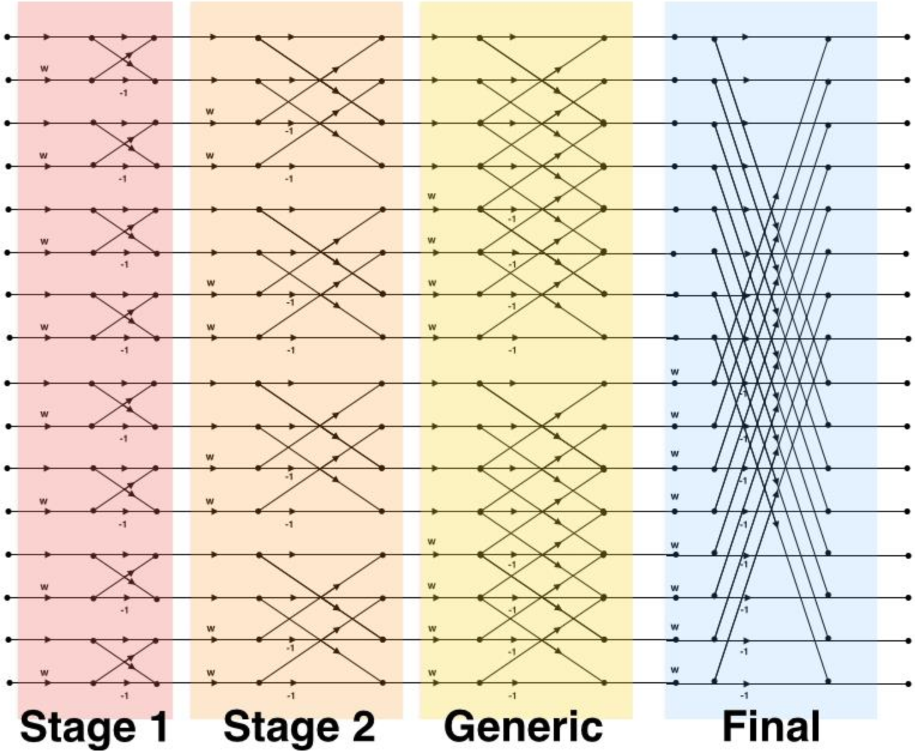
The Xilinx ISE system is a fused framework condition that includes a game plan of ventures to make (get), recreate and execute propelled designs in a field programmable entryway display (FPGA) or complex programmable basis device (CPLD) target contraption.

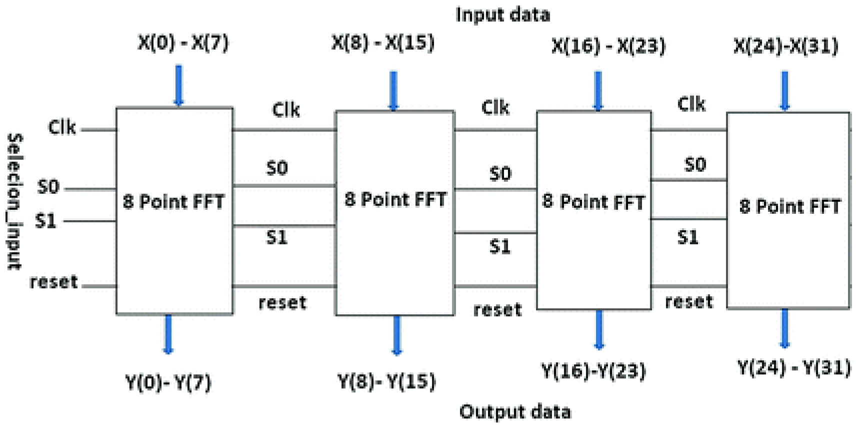
1) Create Verilog configuration input file(s) utilizing bunch driven distribution manager.
2) Compile and finish the Verilog outline file(s).
3) Create the test-vectors and reenact the design (profitable stimulation) without utilizing a PLD (FPGA or CPLD).
4) Assign data/yield pins to finish the game plan on an objective gadget.
5) Test course of action on FPGA/CPLD gadget.

Introductions: data and yield ports, registers and wires.

Rationale Descriptions: conditions, state machines and rationale work

End: endmodule




Input data
X(0) - X(7)
X(8) - X(15)
X(16) - X(23)
X(24)-X(31)
Selecion_input
Clk
S0
S1
reset
8 Point FFT
Clk
S0
S1
reset
8 Point FFT
Clk
S0
S1
reset
8 Point FFT
Clk
S0
S1
reset
8 Point FFT
Y(0)- Y(7)
Y(8)- Y(15)
Y(16)-Y(23)
Y(24) - Y(31)
Output data

2.6 Field Programmable Gate Array (FPGA)

FPGA are a moderate class of coordinated circuits. Firstly, it is presented by the Xilinx organization in 1985. Various contending plans are produced by organizations like, Advanced Micro Devices, and Texas Instruments. A nonexclusive FPGA comprises of a variety of rationale components together with a between interface arrange which can be designed by the client at the purpose of use. Client programming indicates both the rationale capacity of each square and the associations between the pieces. Programming can take two structures; one-time programmable chips, and reprogrammable chips. The Programming advances which enable these gadgets to be "field programmable" are EPROM and EEPROM transistors, against breakers, and SRAM cells. The choice of a programming innovation is critical in light of the fact that it influences chip thickness, flag and rationale engendering postponements, and chip control utilization. FPGAs can be modified by changing the qualities of an exchanging component utilizing EPROM and EEPROM programming innovations. The elective strategy to plan FPGA gadget is to utilize hostile to meld component. The hostile to intertwine protection which is regularly high $(> 100M\Omega)$is for all time changed to a low protection $(200 - 500\omega)$ by the use of proper programming voltage.

Programmable associations in these FPGAs are made utilizing mux, transmission doors, and put the data away in SRAM cells. Since, SRAM is unpredictable in nature, these FPGAs must be modified to set the circuit design each time that power is connected to the chip.

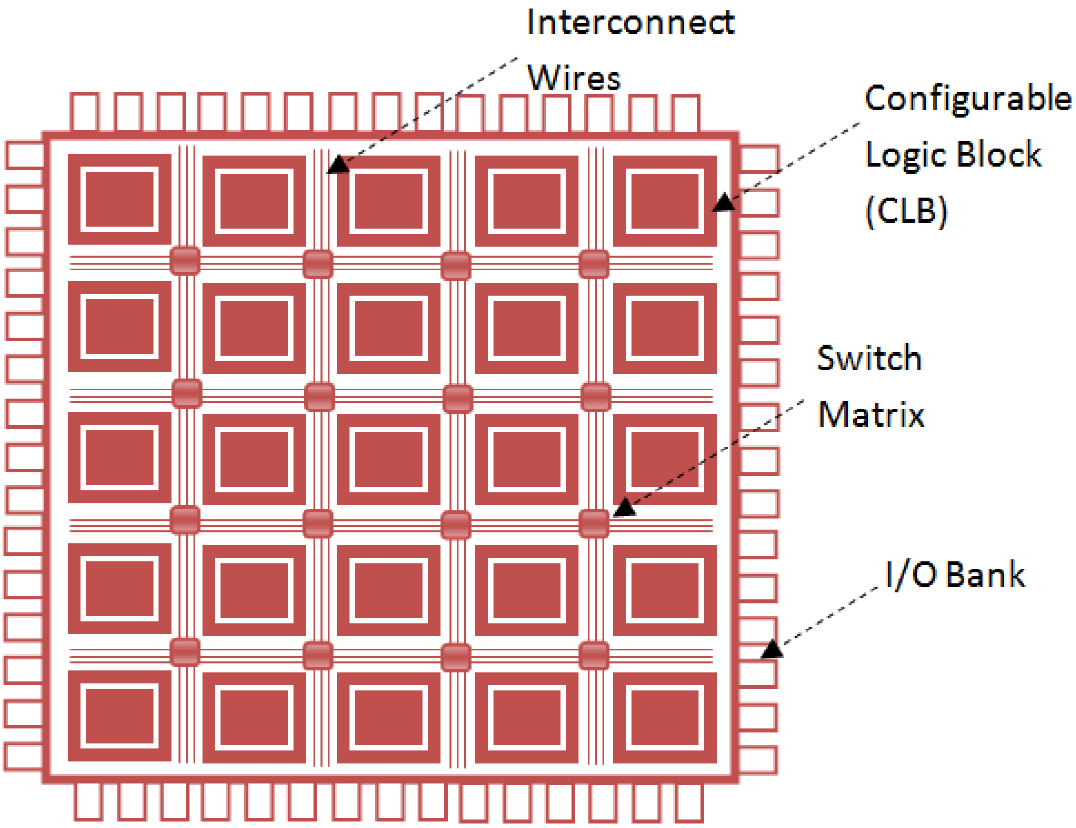


Fig.2.6 architecture

2.6.1 Why FPGA?

A field-programmable entryway group (FPGA) is an organized circuit, which is planned to be orchestrated by the customer or maker resulting to amassing. FPGA can be used to execute any method of reasoning work. This wander is engaged for FPGA for the going with reasons:

1. FPGA speak to another bearing for DSP, correspondence and continuous frameworks and there is much unique work to be done as far as upgraded calculations for this kind of frameworks.

2. They fill a need in the plan space of advanced frameworks, correlative to the part of small scale processors. Since, microchips can be utilized as a part of assortment of, however they depend on the product condition to actualize capacities. Henceforth, small scale processors are by and large slower and more power hungry than the custom chips.

3. There is no hold up from finishing the outline to get the working chip that is configuration can be modified into FPGA and tried quickly.

4. They are brilliant prototyping vehicles. At the point when utilized as a part of definite outline, the bounce from model to item is significantly littler and simpler to arrange.

5. In a few distinct outlines, they are utilized to diminish the stock cost.

2.6.2 Characteristics of FPGA

The qualities of FPGA are given underneath:

1. They are standard parts: i.e. not intended for a specific capacity but rather are modified by the client for a specific reason.

2. They execute multilevel rationale: Means rationale hinders inside FPGA can be associated in a system of subjective profundity. Since FPGA executes multilevel rationale, they for the most part require both programmable rationale pieces and programmable interconnects. PLD's utilization settled interconnects and just change the rationale work appended to the wires. FPGA, interestingly requires programming rationale pieces and interfacing them together so as to execute work.

3. One of significant normal for FPGA is that it can be customized, which is altogether different than in chip. Since FPGA's program are straightforwardly entwined into rationale structure of FPGA.

2.7 SPARTAN-6 FPGA

2.7.1 GENERAL DESCRIPTION

They were often used as a piece of model since they could be altered and installed into load up, in the blink of an eye/minutes. Notwithstanding, they didn't make it into the last thing. Moreover, now a days they are used as a piece of all sort of cutting edge structures: as video enlivening operator in home, singular video recorder (PVR's). They gives programmable method of reasoning segments and programmable interconnects The Spartan-6 family outfits driving structure blend limits with the most negligible total cost for high-volume applications. The thirteen-section family passes on stretched out densities going from 3,840 to 147,443 reason cells, with a vast part of the power usage of past Spartan families, and snappier, more thorough accessibility. These join 18 Kb (2 x 9 Kb) square RAMs, second period DSP48A1 cuts, SDRAM memory controllers, updated mixed mode clock organization pieces, Select I/0 development, control streamlined quick serial handset squares, PCI Express immaculate Endpoint squares, impelled system level power organization modes, auto-recognize outline options, and enhanced IP security with AES and Device DNA affirmation. These features give a negligible exertion programmable other alternative to custom ASIC things without scarcely lifting a finger of use. Stark 6 FPGAs offer the best response for high-volume method of reasoning designs, customer arranged DSP frameworks, and cost-delicate embedded applications. And FPGA are the hardware sections that engage originators to base on headway when their change cycle starts.to create the multilevel basis work.

2.7.2 SPARTAN-6 CONFIGUATION

Straightforward 6 FPGAs store the modified setup data in SRAM-type inside locks. The amount of bits lies between 3 Mb and 33 Mb depending upon contraption size and customer design execution decisions. The setup amassing is erratic and must be reloaded at whatever point the FPGA is controlled up. This amassing can in like manner be reloaded at whatever point by pulling the PROGRAM stick Low. A couple of procedures and data positions for stacking course of action are open.

The bit stream setup data is created by the ISE programming utilizing a program called BitGen. The setup procedure commonly executes the accompanying grouping:

- Detects control up or PROGRAM when Low.
- Clears the entire arrangement memory.
- Samples the mode pins to decide the arrangement mode.
- Loads the design information beginning with the transport width recognition design took after by a synchronization word, checks for the best possible gadget code, and closures with a cyclic excess check (CRC) of the entire bit stream.

2.7.3 CLOCKMANAGEMENT

Every S-6 FPGA has up to six CMTs, every single comprising of two DCMs and single PLL, which can be utilized separately. The DCM can recognize and track normal spread-run clock inputs, in the event that they agree to the data check particulars recorded in the Spartan-6 FPGA Data Sheet. Direct 6 FPGAs can create a spread-extend clock source from a standard settled repeat oscillator.

2.7.4 INPUT/OUTPUT

Quantity of I/O pins differs to 102 - 576, contingent upon gadget & bundle estimate. Every I/O stick is configurable & can consent to countless, utilizing something like 3.3V. Except for supply pins and a couple of devoted design sticks, all other bundle pins have a similar I/O abilities, compelled just by certain managing an account rules. All client I/O is bidirectiona

This segment portray the accessible rationale assets associated with the I/O interfaces. All sources of info and yields can be designed as either combinatorial or enlisted. Twofold information rate (DDR) is upheld by all sources of info and yields. Any information or yield can be independently deferred by up to 256 augmentations (aside from in the - 1L speed review). This is actualized as IODELAY2. The indistinguishable defer esteem is accessible either for information info or yield. For a bidirectional information line, the exchange from contribution to yield defer is programmed. The quantity of defer steps can be set by arrangement and can likewise be augmented or decremented while being used.

Since these tap delays change with supply voltage, process, and temperature, a discretionary adjustment system is incorporated with each IODELAY2:

•For source synchronous outlines where more exactness is required, the alignment system can (alternatively) decide powerfully what number of taps are expected to defer information by one full I/O clock cycle, and after that projects the IODELAY2 with half of that esteem, along these lines focusing the I/O check amidst the information eye.

2.7.5 SPARTAN-6 FPGA BOARD

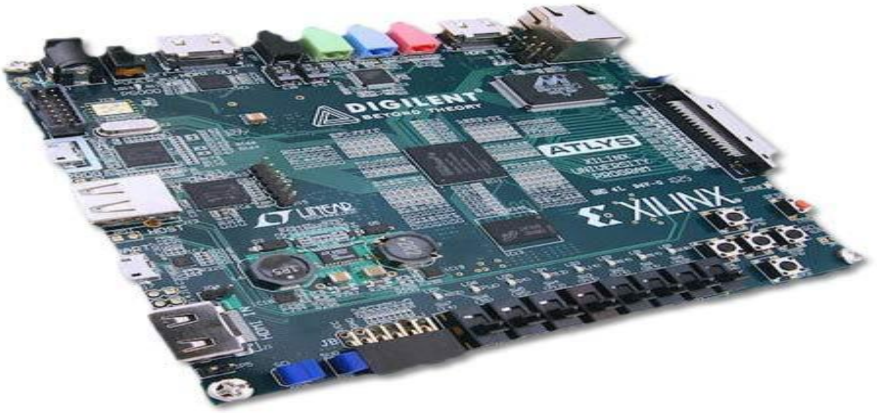


Figure 2.1 SPARTAN-6 FPGA BOARD

CHAPER 3

FFT MECHANISM

3.1 FAST FOURIER TRANSFORM

A FFT is a capable figuring to enlist the DFT and its inverse. There are various specific FFT figurings including a broad assortment of science, from essential math to store up speculation and number theory. The brisk Fourier Transform is an exceptionally profitable philosophy for figuring the DFT of a restricted game plan and requires less number of counts than that of direct evaluation of DFT. It diminishes the figurings by misusing the way that the estimation of the coefficients of the DFT should be possible iteratively. Along these lines, FFT count system is used as a piece of cutting edge unearthly examination, channel amusement, autocorrelation and case affirmation. The FFT relies upon deterioration and breaking the change into smaller changes and merging them to get the total change. It improves the execution by a factor of no less than 100 over direct appraisal of the DFT. This undertaking is important in various fields however enrolling it particularly from the definition is consistently too move back to be in any capacity rational.

A FFT is a way to deal with process a comparative result more quickly enlisting a DFT of N centers in the prominent way, using the definition, takes O(N2) arithmetical errands, while a FFT can figure a comparative result in just O(N log N) exercises. The refinement in speed can be critical, especially for long instructive lists where N may be in the thousands or millions—for all intents and purposes, the estimation time can be lessened by 1 a couple of solicitations of degree in such cases, and the change is for the most part comparing to N/log (N). This huge change made various DFT-based counts utilitarian. FFT' offer of marvelous hugeness to a wide collection of usages, from automated banner getting ready and settling inadequate differential conditions to estimations for quick increment of far reaching entire numbers. The most without a doubt comprehended FFT counts depend on the factorization of N , yet here are FFT with O(N log N) multifaceted nature for all N , despite for prime N . Various FFT counts simply depend upon the way that is a nth unrefined establishment of solidarity, and thusly can be associated with nearly looking like changes over any restricted field, for instance, number the progressions.

The Fast Fourier Transform estimation manhandle the two central properties of the twiddle factor - the symmetry property and periodicity property which decreases the amount of complex increases required to perform DFT. FFT counts rely upon the basic standard of separating the estimation of DFT of a gathering of length N into continuously more diminutive discrete Fourier changes. There are two classes of FFT estimations.

1) Decimation In Time (DIT) estimation
2) Decimation In Frequency (DIF) estimation.

In destruction in-time, the game plan for which we require the DFT is logically parceled into smaller groupings and the DFTs of these subsequences are managed in a case to get the required DFT of the entire progression. In the destruction - in-repeat approach, the repeat trial of the DFT are disintegrated into more diminutive and more diminutive subsequences similarly. The amount of complex increment and development errands required by the clear structures both the DFT and IDFT is of demand N/2 as there are N data centers to process, each one of which require N complex number juggling exercises. The discrete Fourier transform (DFT) is defined by the formula:

$$X(k)=\sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} \qquad k=0,1,2,3,\ldots,N-1$$

3.2 The Radix-2 FFT algorithms

Radix-2 calculations are the least demanding FFT calculations. The devastation in-time(DIT) radix-2 FFT algorithmic segment a DFT into two half-length DFTs of the levels listed and odd ordered time test. The yields of the shorter FFT are reprocessed to assess numerous yields, consequently profoundly lessening the aggregate computation cost. The radix-2 obliteration in time and pulverization in recurrence quick Fourier change (FFTs are the most straightforward FFT calculations. Like each FFTs, this get its speed through reprocessing the eventual outcomes of more diminutive once, widely appealing tally to survey diverse DFT repeat yields.The Radix-2 DIT calculation modifies the DFT of the capacity xn into two sections:

A whole finished the even-numbered lists n = 2m and an aggregate over the odd-numbered files :

n = 2m+1:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}.$$

Given a sequence $x(n) = [1\,,2\,,3\,,4\,,4\,,3\,,2\,,1]$. determine $X(k)$ using DIT FFT algorithm.

Solution:

$$W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$$

Given that $N = 8$

Therefore

$$W_8^0 = e^{-j\left(\frac{2\pi}{8}\right)0} = 1$$

$$W_8^1 = e^{-j\left(\frac{2\pi}{8}\right)} = \cos {}^{\pi}/_{4} - j \sin {}^{\pi}/_{4} = 0.707 - j\,0.707$$

$$W_8^2 = e^{-j\left(\frac{2\pi}{8}\right)2} = \cos {}^{\pi}/_{2} - j \sin {}^{\pi}/_{2} = -j$$

$$W_8^3 = e^{-j\left(\frac{2\pi}{8}\right)3} = \cos {}^{3\pi}/_{4} - j \sin {}^{3\pi}/_{4} = -\,0.707 - j\,0.707$$

Index	Binary representation	Bit reversed binary	Bit reversed index
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Table 3.1 –
Bit Reversal

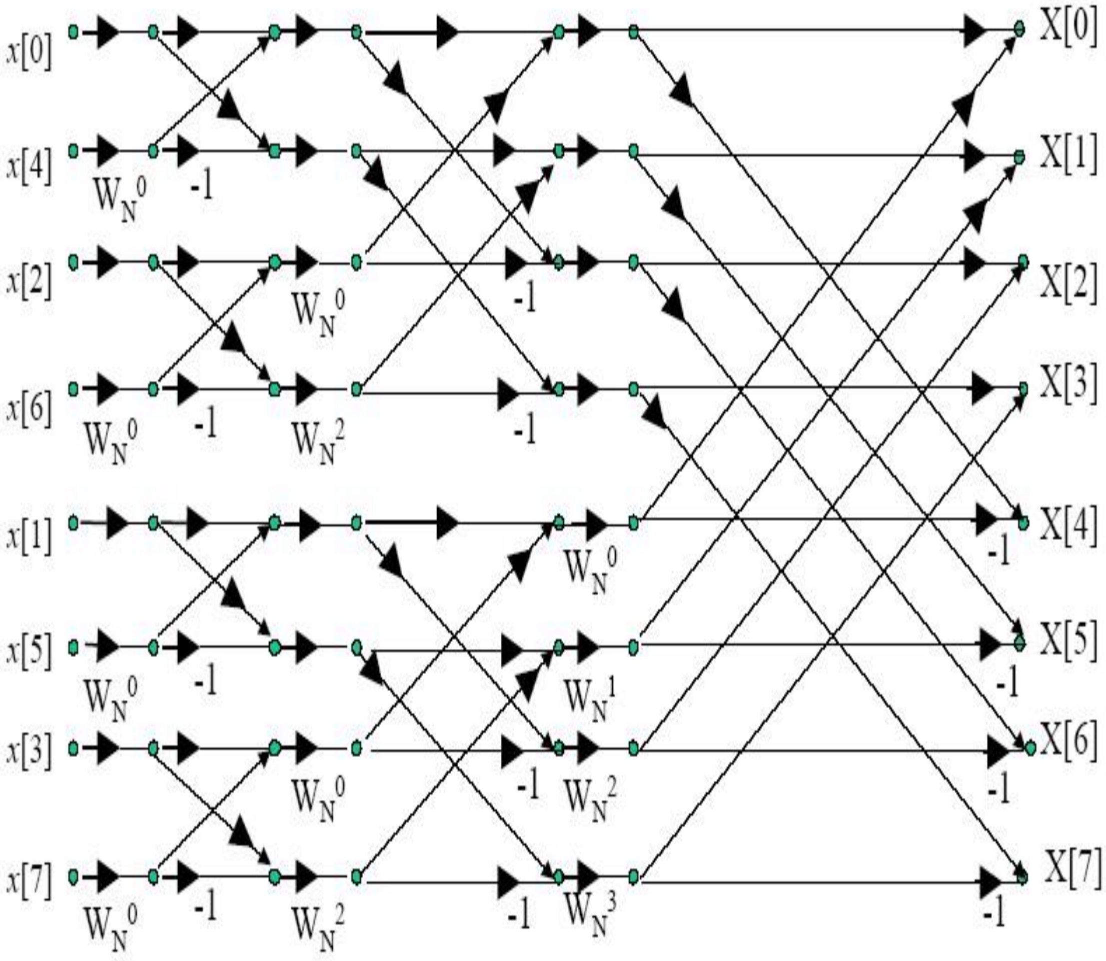


Fig. 3.1 – 8-point DFT

3.3 Decimation-In-Time (DIT) FFT Butterfly

In the radix-2 DIT FFT calculation, the time destruction process goes through an aggregate of M stages where N=2M with N/2-point FFT or butterflies per organize, giving a sum of (N/2)log2N butterflies for every N-point FFT. For the instance of 8-point FFT executed utilizing radix-2 DIT FFT calculation, each information tests are handled through three phases. Four butterflies are required per organize, giving twelve butterflies in the radix-2 usage. Each butterfly is 2-point DFT of the frame portrayed in figure (6). P and Q are the contributions to the radix-2 DIT FFT butterfly. As appeared in this figure, the yields P' and Q' of the radix-2 butterfly are given

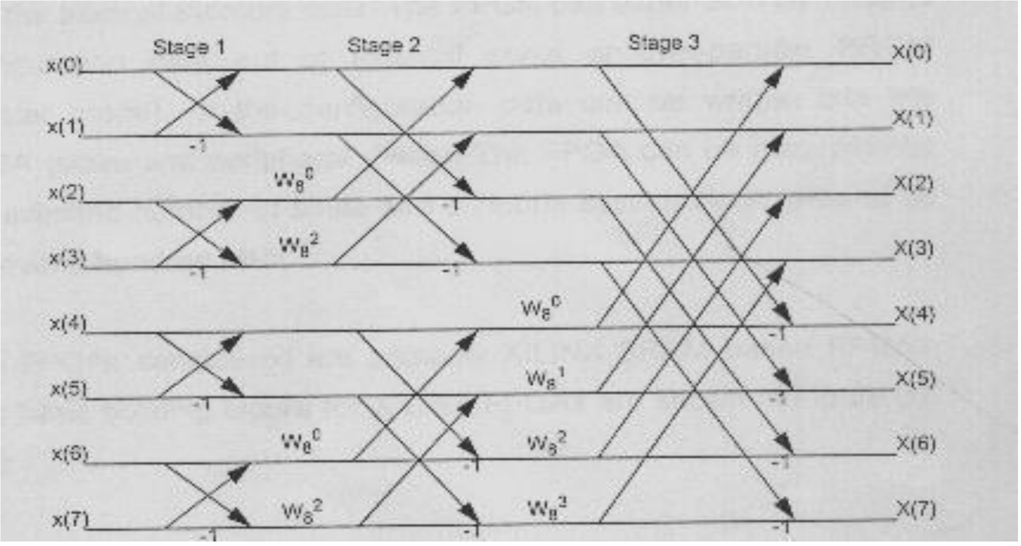


Fig.3.2 - Radix – 2 DFT structure for 8- point FFT

3.4 Butterfly diagram

With respect to fast Fourier change estimations, a butterfly is a portion of the count that solidifies the eventual outcomes of humbler DFT into a greater DFT, or the different way. "butterfly" begins from the condition of the data stream plot in the radix-2 case, as delineated underneath. The timeliest occasion in print of the term is accepted to be in a 1969 MIT particular report. A comparative structure can in like manner be found in the Viterbi estimation, used for finding the no uncertainty gathering of covered states.

Most by and large, the articulation "butterfly" appears with respect to the Cooley– Tukey FFT figuring, which recursively isolates a DFT of composite size n = rm into r smaller changes of size m where r is the "radix" of the change.

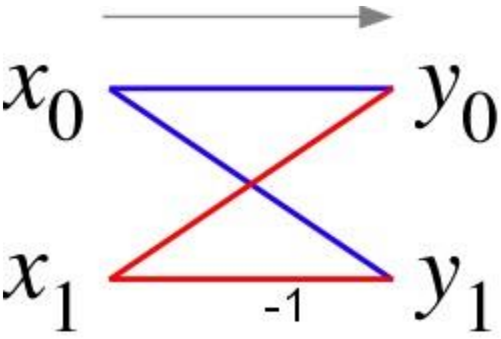


Fig 3.3 – Butterfly Diagram

3.5 ARCHITECTURE IMPLEMENTED

This engineering takes contribution as x0, x1, x2, x3, x4, x5, x6, x7 and these sources of info are given to the primary phase of the FFT design. Second stage inputs are taken from the primary stage and we are getting second stage yields X0, X 1, X2, X3, X4, X5, X6, X7. The framework design is appeared in Fig:

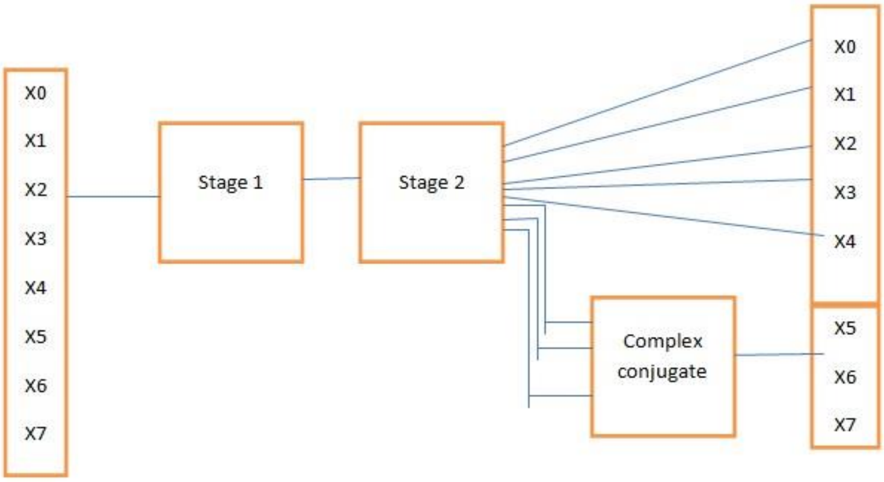
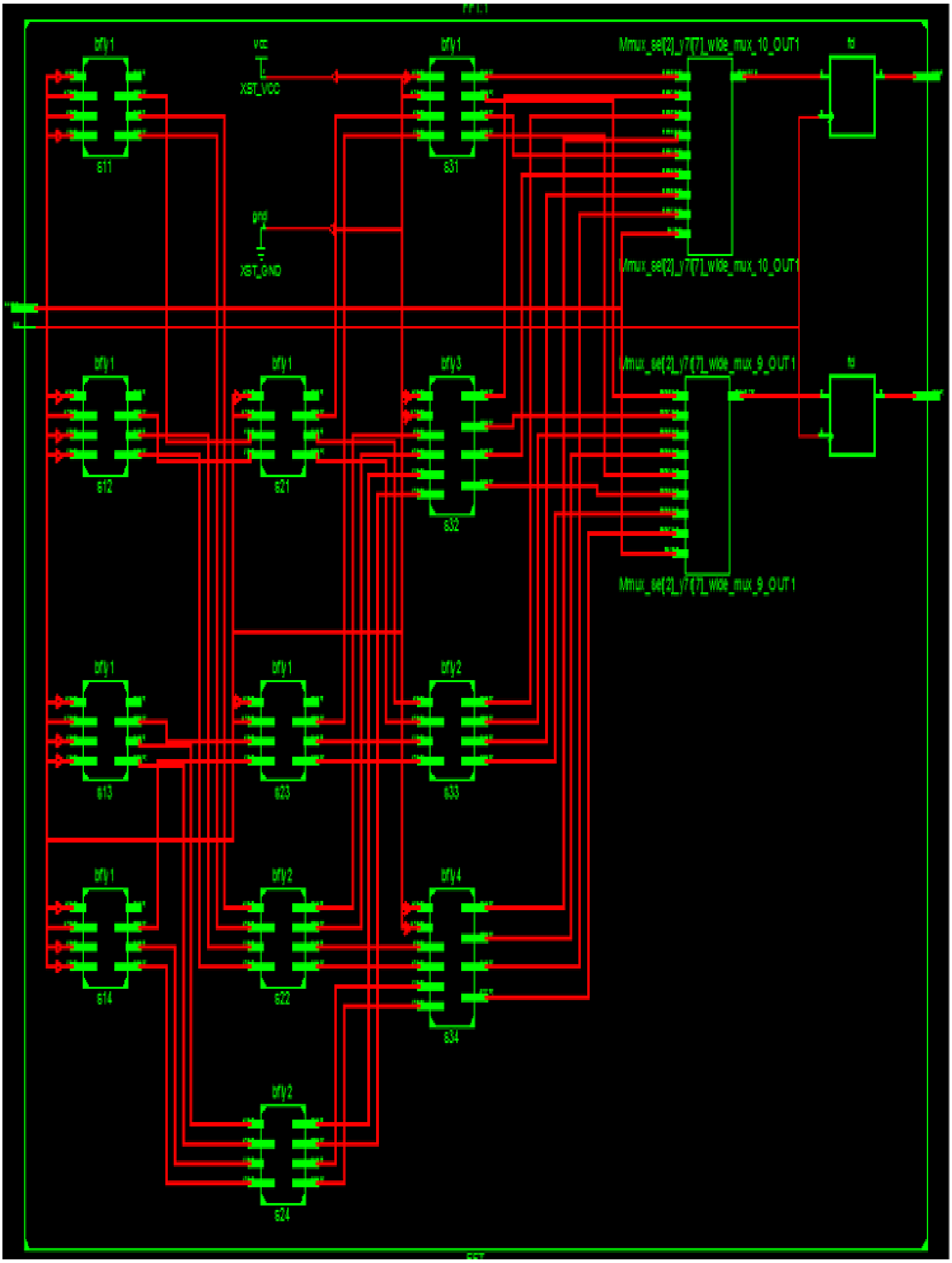


Figure 3.4 Architect

3.6 OUTPUT AND RESULTS

Output is RTL schematic form:



bfly1
s11
vcc
XST_VCC
gnd
XST_GND
bfly1
s31
Mmux_sel[2]_y7[7]_wide_mux_10_OUT1
fd
Mmux_sel[2]_y7[7]_wide_mux_10_OUT1
bfly1
s12
bfly1
s21
bfly3
s32
Mmux_sel[2]_y7[7]_wide_mux_9_OUT1
fd
Mmux_sel[2]_y7[7]_wide_mux_9_OUT1
bfly1
s13
bfly1
s23
bfly2
s33
bfly1
s14
bfly2
s22
bfly4
s34
bfly2
s24

Figure 3.5 Output

CHAPTER 4

CONCLUSION

The plan and execution of 8-point radix-2 FFT has been finished. A butterfly is a section of the count that joins the outcomes of more diminutive discrete Fourier changes (DFTs) into a greater DFT, or the different way (greater DFT into sub changes).FFT calculation has been illuminated utilizing destruction in time technique. In the radix-2 DIT FFT calculation, the time pulverization process goes through a sum of M stages where N=2M with N/2-point FFT or butterflies per arrange, giving a sum of (N/2) log2N butterflies for each N-point FFT. Utilizing twiddle factor figuring's and butterfly graph FFT has been ascertained.

REFERENCE

1. https://www.researchgate.net/publication/212094163

2. http://paper.ijcsns.org/07_book/201111/20111123.pdf

3. https://en.wikipedia.org/wiki/Butterfly_diagram

4. https://www.researchgate.net/publication/228798352_Application_of_the_Short_Time_Fourier_Transform_in_Evaluation_of_the_Acoustic_Emission_Signals_Generated_by_Partial_Discharges?ev=auth_pub