ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

Design And Simulation Of Pipelined Radix-2^k Feed-Forward FFT Architectures

T.S. Ghouse basha[1], Peerla Sabeena sulthana[2]

Associate Professor and Head of Department, KORM Engineering College, Kadapa, Andhra Pradesh, India[1]

Student, M. Tech (VLSI), Department of ECE, KORM Engineering College, Kadapa, Andhra pradesh, India[2]

Abstract: It is vital to develop a superior FFT processor to satisfy the necessities of real time and low price in several different systems. This paper discusses about the design of FFT processor using VHDL. Here we simulated the 64-point FFT processor with radix-4 in VHDL code using ModelSIM 6.5e and the synthesis was performed using Xilinx ISE 8.1i. The architectures of 32 point FFT with radix-2 and 64-point FFT with radix-4are shown in this paper. Finally the simulation graphs of pipelined 64-point FFT processor are generated.

Keywords: FFT, Radix-2, Radix-4, Pipelined architecture.

I. INTRODUCTION

Fast Fourier rework (FFT) processor is wide used in different applications, like local area network, image method, spectrum measurements, measuring device and transmission communication services [1]. However, the FFT formula is a hard task and it should be exactly designed to urge an efficient implementation. If the FFT processor is created flexible and quick enough, a transportable device equipped with wireless transmission is possible. Therefore, an efficient FFT processor is needed for period of time operations [2] and planning a quick FFT processor may be a matter of nice significance.
A Fast Fourier transform is used to compute the Discrete Fourier transform called as DFT and its inverse IDFT. A Fourier transform is one which converts time domain (or space/frequency) to frequency domain and vice versa. FFT performs such transformations very quickly. As a result, these(fast Fourier transform) are most widely implemented in several applications. Some of them like in engineering, medical science, and also in mathematics etc., The Fast Fourier transforms have been described as "the most important numerical algorithm of our lifetime."

There are many different FFT algorithms involving a wide range of basic mathematics like beginning from simple, complex-number arithmetic operations to a group and number theory. This article presents a brief view about the number of available techniques and also about their properties.

The Discrete Fourier Transform can be generated by decomposing a sequence of N values into group of components having different frequencies. This process is useful in many applications as stated in the above fields (see the properties and applications of discrete Fourier transform). But the computation of DFT directly from its definition is practically often very slow. In such cases an FFT is an algorithm used to compute the same operation quickly. computing the DFT of N points in the naive way, using the definition, takes O(N^2) arithmetical operations, where as the FFT can perform the same computation of DFT in just O(N log N) operations. The difference of both techniques can be variated in terms of speed and it can be enormous particularly when the value of N may be in thousands or millions. practically, the time taken for the computation can be minimized in several orders of magnitudes. In such cases, roughly the improvement is proportional to N / log(N). This large improvement made the calculation of the DFT practically possible. FFTs are having great importance to a wide variety of applications like digital signal processing, solving the partial differential equations in to algorithms for fast multiplication of large integers.

A well-known FFT algorithm depends upon the factorization of N, but there are some FFTs which are having complexity of O(N log N) for all N, even for prime N also. All the FFT algorithms depend on the factor $e^{-\frac{2\pi i}{N}}$ which is an N_{th} primitive root of unity, and thus it can be applied to analogous transforms. And can be applied over any finite field, such as number-theoretic transforms. Since the IDFT(Inverse DFT) is the even as the DFT, but with the opposite sign in the exponential and a 1/N factor, any FFT algorithm can easily adapted for it.

An FFT performs the computation on the DFT and produces exactly the same result as evaluating the definition of DFT directly. But the only difference is that an FFT is much faster. Even the presence of round-off error may also make many FFT algorithms to be more accurate than evaluating the DFT definition directly.

Let x_0, ..., x_{N-1} be a sequence of complex numbers. Then the DFT is of the complex sequence is defined by the formula given below

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \dots, N-1.$$

Evaluating the definition directly will require N^2 operations. If there are N number of outputs X_k, and if

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

each output requires a sum of N terms then in such case an FFT is a method to compute the same results in (N log N) operations. More precisely, all the FFT algorithms require O(N log N) computations.

Where, O only denotes an upper bound.

To state the FFT savings, consider the count of complex multiplications and additions. Evaluating the DFT's sums directly involves N^2 complex multiplications and N(N−1) complex additions (of which O(N) operations are protected by eliminating trivial operations such as multiplications by 1].

The standard strategy to speed up the FFT algorithm is to follow the divide and conquer rule. But we need to find some alternate way to group all the terms in the equation

$$V[k] = \Sigma_{n=0..N-1} W_N^{kn} v[n]$$

Let's see what happens when we separate odd ns from even ns (from now on, let's assume that N is even):

$$\begin{aligned} V[k] &= \Sigma_{n\,even} W_N^{kn} v[n] + \Sigma_{n\,odd} W_N^{kn} v[n] \\ &= \Sigma_{r=0..N/2-1} W_N^{k(2r)} v[2r] + \Sigma_{r=0..N/2-1} W_N^{k(2r+1)} v[2r+1] \\ &= \Sigma_{r=0..N/2-1} W_N^{k(2r)} v[2r] + \Sigma_{r=0..N/2-1} W_N^{k(2r)} W_N^{k} v[2r+1] \\ &= \Sigma_{r=0..N/2-1} W_N^{k(2r)} v[2r] + W_N^{k} \Sigma_{r=0..N/2-1} W_N^{k(2r)} v[2r+1] \\ &= (\Sigma_{r=0..N/2-1} W_{N/2}^{kr} v[2r]) + W_N^{k} (\Sigma_{r=0..N/2-1} W_{N/2}^{kr} v[2r+1]) \end{aligned}$$

where we have used one crucial identity:

$$\begin{aligned} W_N^{k(2r)} &= e^{-2\pi i*2kr/N} \\ &= e^{-2\pi i*kr/(N/2)} = W_{N/2}^{kr} \end{aligned}$$

Notice an interesting thing that is the two sums are nothing but the individual N/2-point Fourier transforms of N sequence. One set of N/2 represents the even subset and another represents the odd subset of samples. Terms with k greater or equal N/2 can be reduced using another identity:
$W_{N/2}^{m+N/2} = W_{N/2}^{m} W_{N/2}^{N/2} = W_{N/2}^{m}$
which is true because $W_m^m = e^{-2\pi i} = \cos(-2\pi) + i \sin(-2\pi) = 1$.
If we start with N that is a power of 2, we can apply this decomposing into its sub divisions recursively until we get down to a final 2-point transforms.
Similarly,We can also go backwards initiating with the 2-point transform:
$V[k] = W_2^{0*k} v[0] + W_2^{1*k} v[1], \quad k=0,1$

The two components are:

$$V[0]=W_2^0 v[0]+W_2^0 v[1]=v[0]+W_2^0 v[1]$$
$$V[1] = W_2^0 v[0] + W_2^1 v[1] = v[0] + W_2^1 v[1]$$

the above two equations for the components of the 2-point transform can be graphically represented as shown below which is popularly known as 'butterfly'.

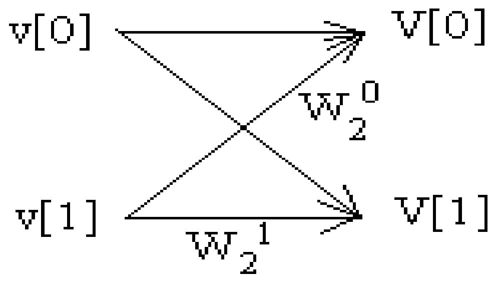


Fig: 1.1Butterfly computation

Further, by using the divide and conquer principle, a 4-point transform can be decomposed to two 2-point transforms. Out of which one for even elements and another for odd elements. All the odd ones will be multiplied by the factor W_4^k. Diagrammatically, this can be represented as two levels of butterflies. Notice that the factor $W_{N/2}^n = W_N^{2n}$. we can often express all the multiplication factors as powers of the same W_N (in this case we choose N=4.)

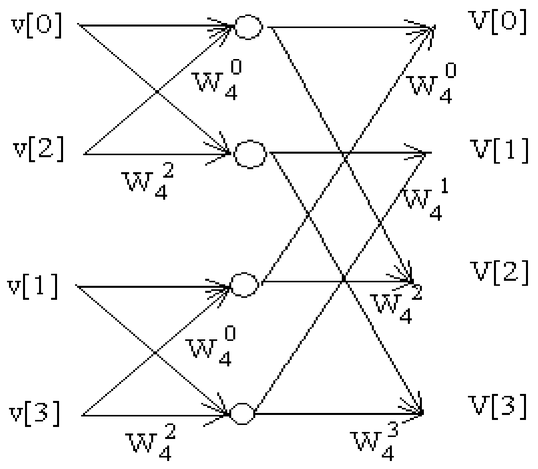


Fig 1.2: Diagrammatical view of 4-point Fourier transform computation

Similarly, the analogous diagrammatical representation for the sequence of N=8 is as shown below. What will become obvious is that all the butterflies have similar form.

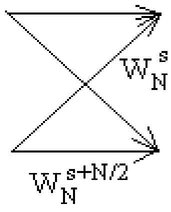


Fig 1.3: Generic butterfly graph.

II. RADIX-4 FFT

It can be realized in two domains that is either in time domain(DIT) or in frequency domain(DFT). The decimation-in-time (DIT) radix-4 FFT algorithm will recursively make partitions of the DFT into four quarter-length DFTs. The output results of these small FFTs are again used to compute number of

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

outputs. Thus it reduces the total computational cost to be required. The radix-4 decimation-in-frequency(DFT) FFT combines every fourth output sample into shorter-length DFTs in order to save the computations. The implementation of radix-4 FFTs require only 75% as many complex multiplications as the radix-2 FFTs.
The radix-4 decimation-in-time algorithm rearranges the equation of Discrete Fourier Transform (DFT) into four parts. The sums over all groups of every fourth discrete-time index are
n=[0,4,8,…,N−4],
n=[1,5,9,…,N−3],
n=[2,6,10,…,N−2] and
n=[3,7,11,…,N−1]
where three of them are multiplied by so-called twiddle factors
WkN=e−(i2πkN), W2kN, andW3kN.

.

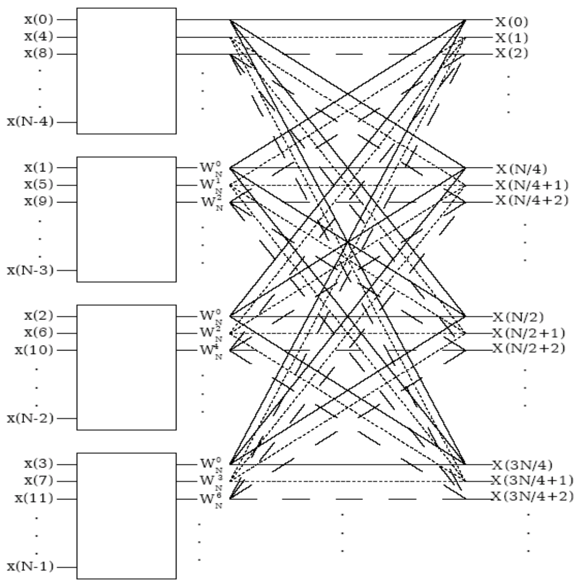


Fig. 2.1: Structure of Radix-4 FFT

III. THE PIPELINED FFT ARCHITECTURE

A Computer can handle millions of instructions for each second. If one instruction is processed, the next one will also be in line and is processed in parallel. A pipeline allows multiple instructions to be processed at the same time.

While one stage of an instruction is being processed, other instructions may be undergoing processing at a different stage. Without a pipeline, each instruction would have to wait for the previous one to finish before it could even be accessed.

- Fetch
- Decode
- Execution

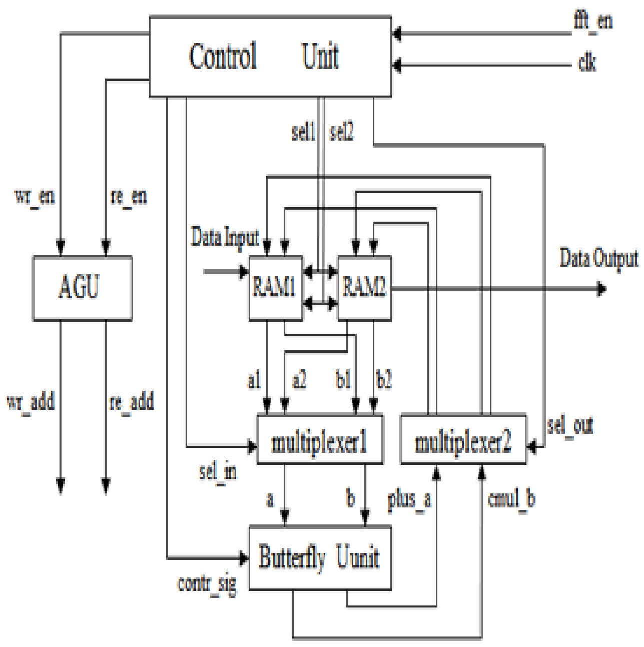


Fig.3.1: The block diagram of pipelined FFT architecture

Control unit is one which generates all the control signals for whole system and is also responsible for control operations of the processor. A 48-bit signal w_con controls the whole FFT processor, and this signal w_con generates two parameters, write_en and read_en, to control the Address Generating Unit (AGU). AGU will create 8 read and 8 write addresses, which determine the data access to outer memories. It also generates sel1 and sel2 signals to select data from two RAMs, each of which is made up of 8 32-bit registers. The RAM1 and RAM2 are made up of eight 32-bit registers respectively. And data is always written to the outside memories from RAM2, and it is always read to RAM1 from the outside memories. The output signals collected from the RAM units are feed as the inputs to the multiplexers and the outputs generated by the MUX is given to the butterfly unit where the computation follows. The BU and the remaining parts are controlled by w_con as well. This control unit harmonizes all steps of the FFT processor based on a 7-bit counter.

IV. IMPLEMENTATION

FPGA:
The Field Programmable Gate Array is mostly implemented for the generation of ASIC IC's for the purpose of computations. FPGA offers high speed in execution process. Hence, for the generation of ASIC IC's FPGA's play a major role and thus they are widely used. Here, the 64-Point FFT algorithm with radix 4 is simulated and synthesized as well as implemented on the FPGA of below configuration.

Table.1: Configuration of FPGA

Property Name	Value
Family	Spartan 3AN
Device	XC3S50AN
Package	TQG144
Speed Grade	-4

Simulation results:

The RTL view of the 64-Point FFT and the butterfly structure of the same are obtained after the simulation of the 64-point FFT block. And later its internal architecture is shown. And next to that the simulation and synthesis reports are also generated.

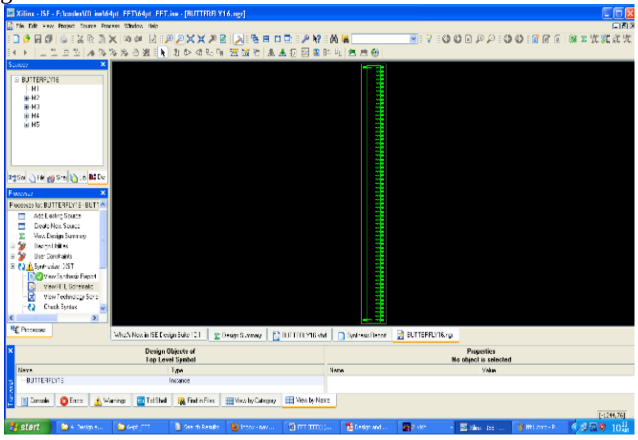


Fig 4.1: RTL View Of the Butterfly Component in 64-Point FFT

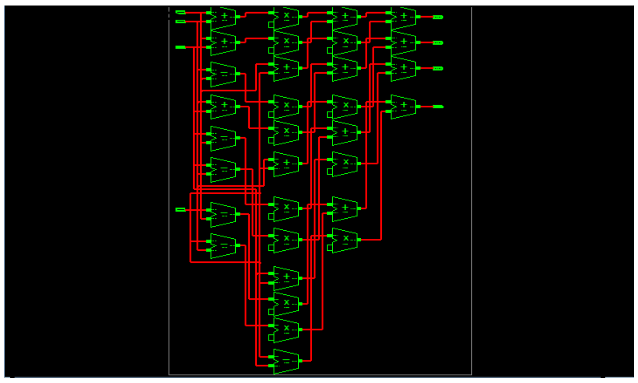


Fig 4.2: Internal Architecture of the Butterfly component

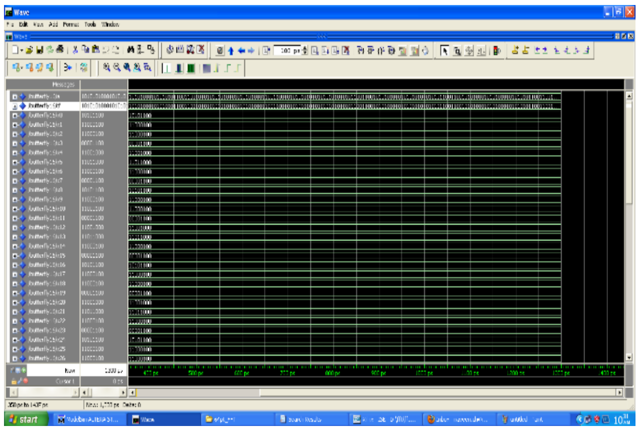


Fig 4.3: Simulation result of 64 FFT

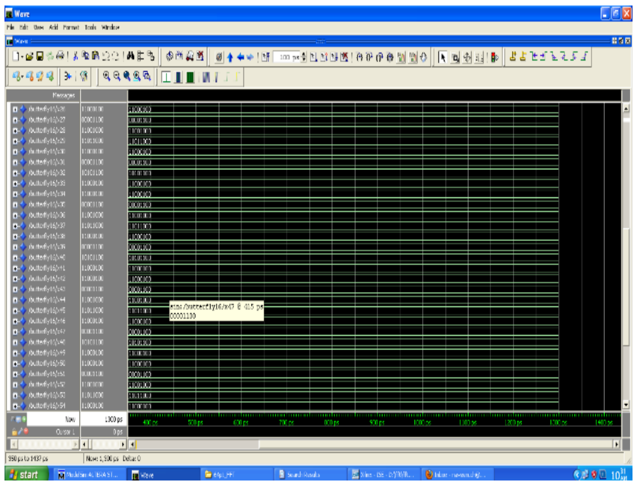
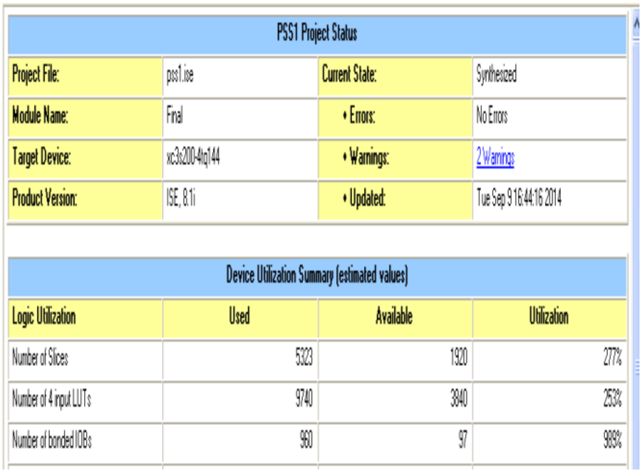


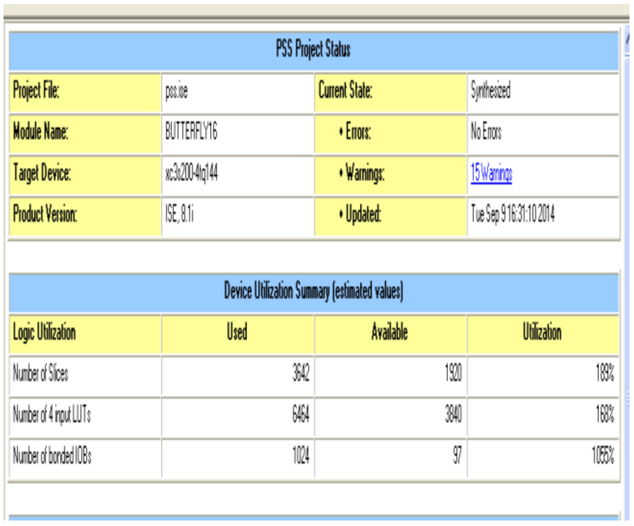
Fig 4.4: Simulation result of 64 FFT



PSS1 Project Status			
Project File:	pss1.ise	Current State:	Synthesized
Module Name:	Final	• Errors:	No Errors
Target Device:	xc3s200-4tq144	• Warnings:	2 Warnings
Product Version:	ISE, 8.1i	• Updated:	Tue Sep 9 16:44:16 2014

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	5323	1920	277%
Number of 4 input LUTs	9740	3840	253%
Number of bonded IOBs	960	97	989%

Fig 4.5: Design summary of 32 point FFT



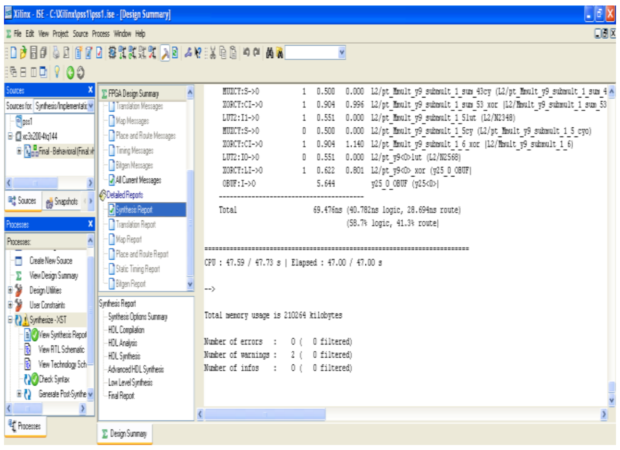
PSS Project Status			
Project File:	pss.ise	Current State:	Synthesized
Module Name:	BUTTERFLY16	• Errors:	No Errors
Target Device:	xc3s200-4tq144	• Warnings:	15 Warnings
Product Version:	ISE, 8.1i	• Updated:	Tue Sep 9 16:31:10 2014

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	3642	1920	189%
Number of 4 input LUTs	6464	3840	168%
Number of bonded IOBs	1024	97	1055%

Fig 4.6: Design summary of 64 point FFT

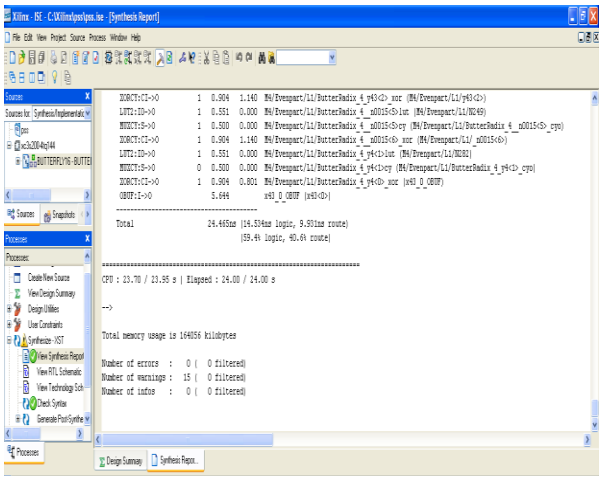
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014



```
   MUXCY:S->O          1   0.500   0.000  L2/pt_Mmult_y9_submult_1_sum_43cy (L2/pt_Mmult_y9_submult_1_sum_4
   XORCY:CI->O         1   0.904   0.996  L2/pt_Mmult_y9_submult_1_sum_53_xor (L2/Mmult_y9_submult_1_sum_53
   LUT2:I1->O          1   0.551   0.000  L2/pt_Mmult_y9_submult_1_5lut (L2/N2348)
   MUXCY:S->O          0   0.500   0.000  L2/pt_Mmult_y9_submult_1_5cy (L2/pt_Mmult_y9_submult_1_5_cyo)
   XORCY:CI->O         1   0.904   1.140  L2/pt_Mmult_y9_submult_1_6_xor (L2/Mmult_y9_submult_1_6)
   LUT2:I0->O          0   0.551   0.000  L2/pt_y9<0>lut (L2/N2568)
   XORCY:LI->O         1   0.622   0.801  L2/pt_y9<0>_xor (y25_0_OBUF)
   OBUF:I->O               5.644          y25_0_OBUF (y25<0>)
  ----------------------------------------
  Total                   69.476ns (40.782ns logic, 28.694ns route)
                                   (58.7% logic, 41.3% route)

=========================================================================
CPU : 47.59 / 47.73 s | Elapsed : 47.00 / 47.00 s

-->

Total memory usage is 210264 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    2 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```

Fig 4.7: Synthesis report of 32FFT



```
   XORCY:CI->O         1   0.904   1.140  M4/Evenpart/L1/ButterRadix_4_y43<2>_xor (M4/Evenpart/L1/y43<2>)
   LUT2:I0->O          1   0.551   0.000  M4/Evenpart/L1/ButterRadix_4__n0015<5>lut (M4/Evenpart/L1/N249)
   MUXCY:S->O          1   0.500   0.000  M4/Evenpart/L1/ButterRadix_4__n0015<5>cy (M4/Evenpart/L1/ButterRadix_4__n0015<5>_cyo)
   XORCY:CI->O         1   0.904   1.140  M4/Evenpart/L1/ButterRadix_4__n0015<6>_xor (M4/Evenpart/L1/_n0015<6>)
   LUT2:I0->O          1   0.551   0.000  M4/Evenpart/L1/ButterRadix_4_y4<1>lut (M4/Evenpart/L1/N282)
   MUXCY:S->O          0   0.500   0.000  M4/Evenpart/L1/ButterRadix_4_y4<1>cy (M4/Evenpart/L1/ButterRadix_4_y4<1>_cyo)
   XORCY:CI->O         1   0.904   0.801  M4/Evenpart/L1/ButterRadix_4_y4<0>_xor (x43_0_OBUF)
   OBUF:I->O               5.644          x43_0_OBUF (x43<0>)
  ----------------------------------------
  Total                   24.465ns (14.534ns logic, 9.931ns route)
                                   (59.4% logic, 40.6% route)

=========================================================================
CPU : 23.70 / 23.95 s | Elapsed : 24.00 / 24.00 s

-->

Total memory usage is 164056 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :   15 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```

Fig 4.8: Timing Report Of 64FFT

V. CONCLUSION

In this paper, a 32 point FFT with radix-2 and 64point with radix-4 processor was designed using FPGA System successfully. The processor uses VHDL language to describe the circuit. Xilinx ISE8.1i software is used to build the model, and ModelSim SE 6.5e software for simulation.

REFERENCES

[1] R. W. Chang, "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission," Bell System Tech. J., vol. 45, pp. 1775–1796, Dec. 1966.
[2] ETS 300401, ETSI, "Digital Audio Broadcasting (DAB);DAB to mobile, portable and fixed receivers," 1995.
[3] J. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come," IEEE Communications Magazine, vol. 8, pp. 5–14, May 1990.
[4] S.B. Weinstein and P. M. Ebert, "Data transmission by frequency division multiplexing using the discrete Fourier transform," IEEE Transactions on Communications, vol. 19, pp. 628–634, Oct. 1971.
[5] R. Morrison, L. J. Cimini, and S. K. Wilson, "On the Use of a Cyclic Extension in OFDM," in Proc. of Vehicular Technology Conference, VTC 2001 Fall, vol. 2, Atlantic City, NJ, USA, Oct. 7-11 2001, pp. 664–668.
[6] A.Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," in Int. Conf. Acoustic, Speech, Signal Processing, Denver, CO, 1980, pp. 964–967.
[7] L.J. Cimini, "Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing ," IEEE Transactions on Communications, vol. 33, pp. 665–675, July 1985.

BIOGRAPHIES

T.S. Ghouse Basha is presently working as an Associate Professor and HOD in the Department of Electronics and Communication Engineering in KORM College of Engineering, Kadapa. He carried out his M.Tech project work in Defence Research and Development Laboratory, Hyderabad and working in teaching field since eleven years in different cadres. He received his B.Tech and M.Tech from the Department of Electronics and Communication Engineering from JNTU University and Nagarjuna University respectively.He has submitted his Ph D thesis in microwave antennas to JNTUA. His areas of interest include microwave antennas, digital signal processing and mobile communications.

Peerla Sabeena Sulthana, Student, is currently Pursuing her M.Tech VLSI., in ECE department from KORM Engineering College, kadapa. She has completed B.Tech in Electronics and Communication Engineering in Kandula Lakshumma Memorial College Of Engineering For Women. Her interest areas are VLSI systems, Micro processors, Electronic devices & circuits, & Digital signal processing.