

A FPGA implementation for high speed OFDM trans-recv system using high performance FFT algorithm

Prathibha T, Ashwini K B, Anand S

Abstract— increasing the performance is the best criterion for developing any communication system. The Fast Fourier Transform (FFT) and its inverse (IFFT) are important algorithms in signal processing and the most promising modulation technique i.e. Orthogonal Frequency Division Multiplexing (OFDM). So when zero valued inputs/outputs outnumber nonzero inputs/outputs, then general IFFT/FFT algorithm for OFDM is no longer efficient. This paper presents a high performance FFT/IFFT for OFDM transceiver which performs its operation in a single clock cycle by input zero traced FFT pruning (IZTFFTP) algorithm and pipelined architecture. The design has been coded in VHDL and targeted into Xilinx Spartan3 FPGAs.

Index Terms— FFT, IFFT, OFDM, Orthogonality, NC-OFDM, pruning techniques, pipelined architecture.

1 INTRODUCTION

The heart of the OFDM is the FFT/IFFT design. There are several algorithms are available to design the FFT/IFFT among them radix-2 is the simplest algorithm. To match with the order or requirement of a system, the common method is to extend the input data sequence $x(n)$ by padding number of zeros at the end of it and which is responsible for a increased value of computational time. But calculation on undesired frequency is unnecessary. As the OFDM based cognitive radio [1] has the capability to nullify individual sub carriers to avoid interference with the licensed user. So that there could be a large number of zero valued inputs/outputs compare to non-zero terms. This is the most important thing in the orthogonal frequency division multiplexing (OFDM), which is used as baseband transmission in spectrum pooling technique. Though large bandwidth supports high data rates but practically it is impossible to find contiguous empty bandwidth. So much efficient data rates are achieved by using non-contiguous vacant subcarriers of a targeted spectrum pool. This type of OFDM is known as non-contiguous OFDM or NC-OFDM [2-3], which helps to avoid the harmful interference by deactivating those subcarriers, which are acquired by different licensed users. That means the input values of the IFFT's of those particular subcarriers is zero. As NC-OFDM consists of large number of de-activated or null subcarrier i.e. numbers of zero valued inputs/outputs outnumber non-zero inputs/outputs. Several researchers have proposed different

In this paper we have proposed an input zero traced FFT pruning (IZTFFTP) algorithm with a pipelined architecture which is suitable for NC-OFDM based transceiver. It is based on normal Cooley-Tukey radix-2 DIF algorithm using matrix factorizing process. Result shows it is more efficient than ordinary FFT.

2 LITERATURE SURVEY

FFT have some major application in the field of signal processing, many researches are still going on to make it more efficient and flexible in terms of system requirement. In order to decrease the computation time of general FFT algorithm, several pruning algorithm have been proposed. In 1971 J.D Markel proposes first FFT pruning algorithm [4]. It was based on DIF FFT model. Following the similar way Skinner [5] has proposed an algorithm based on DIT FFT model using the inputs points only. Then, comprising both the input & output pruning Rao & Srinivas have proposed another algorithm [6]. In 2000 R.G. Alves gave an idea for general FFT pruning algorithm [7]. A different view of FFT algorithm, Transform decomposition method have been proposed by Sorensen & Burrus [8]. TD method generally a modified version of CTFFT algorithm, there DFT is decomposed into two smaller DFTs. Though it is complex but in terms of hardware implementation it is more efficient and flexible rather than pruning.

The previous papers were based on the formation of a matrix, consists of N rows & columns. But this generation of an assistant matrix itself is a complicated and time consuming process and hard to understand also. To get rid-off from these Inefficiencies we have proposed IZTFFTP algorithm, which is the modified version of general Cooley-Tukey algorithm (using matrix factorization) [9], by simple changing in the programming approach. Designing this algorithm in pipeline architecture provides higher efficiency in terms of speed and utilization of clock cycles. The radix-2 algorithm for N points con-

- Prathibha T, Assistant Professor, electronics and communication engineering in channabasaveshwara Institute of Technology, Gubbi, India, PH-08131-223818. E-mail: prathibha.t@cittumkur.org
- Ashwini K B, Assistant Professor, electronics and communication engineering in channabasaveshwara Institute of Technology, Gubbi, India, PH-08131-223818. E-mail: ashwini.kb@cittumkur.org.
- Anand S, Design Engineer, VEDLABS, Bangalore, India, PH-08042040494. E-mail: anandabhavan87@gmail.com

way to make FFT faster by "pruning" the conventional one [4].

sists of $\log_2 N$ stages and if it is implemented using pipeline architecture then it requires totally $N/2$ clock cycles to complete its operation by executing $N/4$ points for one clock cycles [10].

Even this utilization of clock cycles can be reduced by executing of all N points in all stages for a single clock cycle. So that OFDM modem performs high speed of operation.

3 OFDM STRUCTURE

OFDM is a special form of Multi Carrier Modulation (MCM) and the OFDM time domain waveforms are chosen such that mutual orthogonality is ensured even though sub-carrier spectra may over-lap. With respect to OFDM, it can be stated that orthogonality is an implication of a definite and fixed relationship between all carriers in the collection. The sinc function, illustrated in Fig.1 exhibits this property and it is used as a carrier in an OFDM system.

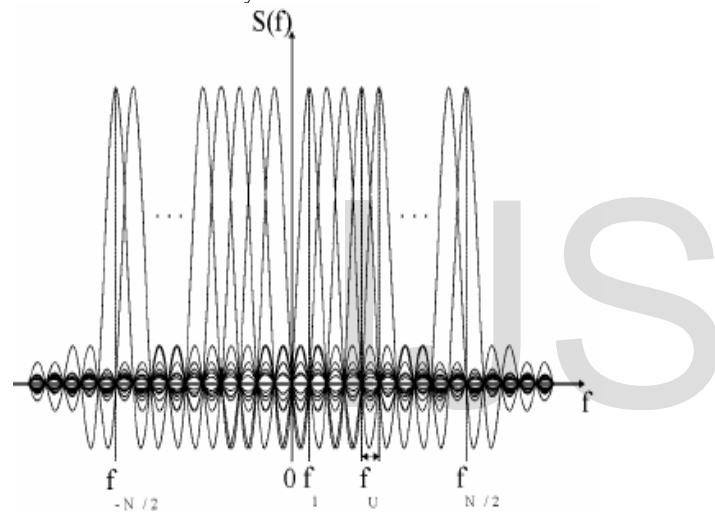


FIG. 1: OFDM SUB CARRIERS IN THE FREQUENCY DOMAIN

3.1 Generation of OFDM Signals

To implement the OFDM transmission scheme, the message signal must first be digitally modulated. The carrier is then split into lower-frequency sub-carriers that are orthogonal to one another. This is achieved by making use of a series of digital signal processing operations. The message signal is first modulated using a scheme such as BPSK, QPSK or some form of QAM (16QAM or 64QAM for example). The sub-carriers can now be generated using Fast Fourier Transforms (FFT). The FFT is used to calculate the spectral content of the signal. It moves a signal from the time domain where it is expressed as a series of time events to the frequency domain where it is expressed as the amplitude and phase of a particular frequency. The inverse FFT (IFFT) performs the reciprocal operation. The underlying principle here is that the FFT can keep tones orthogonal to one another if the tones have an integer number

of cycles in a symbol period. In the example Fig.2, the signals with 1,2 and 4 cycles respectively that form an orthogonal set.

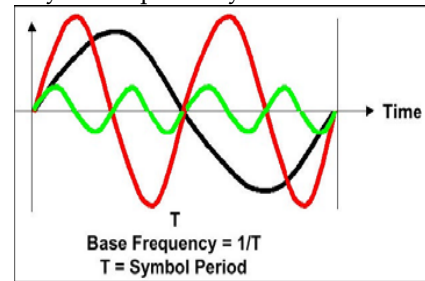


Fig.2: A set of orthogonal signals

To convert the sub-carriers to a set of orthogonal signals, the data is first combined into frames of a suitable size for an FFT or IFFT. A FFT should be always in the length of $2N$ (where N is an integer). Next, an N -point IFFT is performed and the data stream is the output of the transmitter. Thus when the signals of the IFFT output are transmitted sequentially, each of the N channel bits appears at a different sub-carrier frequency. By using an IFFT process, the spacing of the sub carriers is chosen in such a way that at the frequency where the received signal is evaluated, all other signals is zero. In order for this orthogonality, the receiver and the transmitter must be perfectly synchronized. This means they both must assume exactly the same modulation frequency and the same time-scale for transmission. At the receiver, the exact inversions are performed to recover the data. Since the FFT is performed in this stage, the data is back in the frequency domain and it is then demodulated [11].

3.2 Maintenance of Orthogonality

When transmitting an MCM (Multi-Carrier Modulation) symbol immediately followed by another, orthogonality will not be kept longer. When this happens a sub-channel starts to interfere in another sub-channel, leading to ICI (Inter-Channel Interference).

Another problem that occurs is that there would be distortions in the linear channel, such as delays due to multipath and reflections, which generates the symbolic ISI (Inter-Symbol Interference). Both ICI and ISI cause loss of orthogonality of the system. To solve this problem, simply enter a period of custody v between one symbol and another, so that the end of a symbol does not interfere with the next [1].

Basically there are three ways to add this withdrawal period:

1. Add the last v samples at the beginning of the signal, thus a cyclic-prefix gets added.
2. Add the first v samples at the end of the signal, thus a cyclic-suffix gets added.
3. Do not transmit anything on the guard interval in the receiver and add the latest v samples to the beginning.

Among these the most widely used techniques of the cyclic prefix, shown in Fig.3.

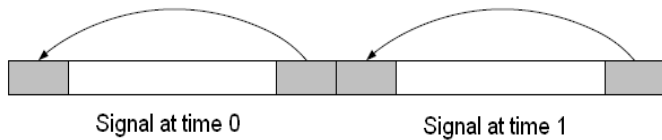


Fig.3: Cyclic Prefix

The efficiency of the channel that gets degraded is being introduced as a v , the period in which samples do not transmit any information. It can be transfer channel impulse response and thus decrease the size of the withdrawal period that is generated by the inclusion of equalizers. Therefore it should be a compromise between efficiency and the complexity of equalizer's time to specify the size of the withdrawal period. The improvement that the period of custody introduced is very rewarding, being practically minded, it is impossible to use the MCM system without cyclic prefix.

3.3 The OFDM Transmitter

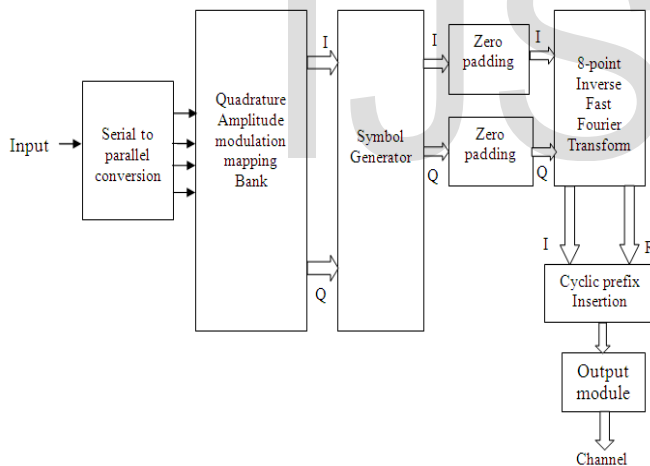


fig.4: OFDM transmitter

Fig.4 illustrate the function of the OFDM transmitter is to receive a block of M bits and generate the OFDM signal. The first step in OFDM transmitter is to convert the serial binary data into parallel data streams to realize the concept of parallel data transmission. The serial to parallel converter receive the M serial bits to be transmitted, and those bits are divided into N sub blocks. Each of these N sub blocks is then mapped to individual data sub-carrier and modulated using some sorts of PSK (Phase Shift Keying) or QAM (Quadrature Amplitude Modulation) i.e. BPSK, QPSK, 16-QAM, 64-QAM. By dividing the data stream into N sub blocks, the symbol period is made N times longer by a symbol generator, which will reduces the

delay spread or chromatic dispersion relative to the symbol time. Zero padding is used to suppress the inter block interference (IBI) by appending zeros in beginning and at the end of the data input. Here two zero padding is used to perform operations on in-phase and quadrature phase data individually.

The Inverse Fast Fourier Transform (IFFT) transforms the signals from the frequency domain to the time domain. The number of subcarriers determines the number of sub-bands the available spectrum is split into. In this paper 8 point radix - 2 butterfly DIF algorithm is used in a pipelined architecture manner. Each input and output point is of 16 bits in length. The complete operation is performed in one cycle. IFFT operations for in-phase and quadrature phase data will be performed separately and produce 8 point of real and imaginary outputs where each point is of 16 bits in length and then finally combined to provide 128 bits of real and imaginary outputs. The 128 bits of real and imaginary outputs of

3.4 The OFDM receiver

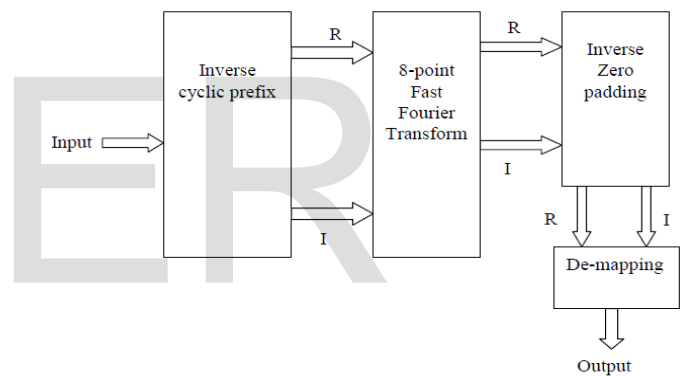


Fig.5: OFDM Demodulator

The OFDM receiver is illustrated in fig.5, containing inverse cyclic prefix, 8 point FFT, inverse zero padding, de-mapping. Inverse cyclic prefix receives 16 bits of input from the output module of the transmitter. In order to receive complete OFDM transmitter data, inverse cyclic prefix will take the input for 19 clock pulse and then it removes the first 48 bits and then next 128 bits is considered as real and the remaining is considered as imaginary data which is given to the FFT module.

The Fast Fourier Transform (FFT) transforms the signals from the time domain to the frequency domain. In this project pipelined architecture 8 point radix - 2 butterflies DIT algorithm is used to design FFT. Each input and output point is of 16 bits in length. The complete operation is performed in one clock cycle. FFT operations for Real and imaginary data will be performed separately and then finally this will provide 128 bits of real and 128 bits of imaginary outputs. The inverse zero padding will remove the 32 bits at the beginning and at the end of the real and imaginary outputs of the FFT module and then

sends data to the de-mapping which makes the decision about the binary sequence that represents the received signal. NC-OFDM transceiver contains many deactivated subcarriers that are used by the licensed users. Those signals are also transmitted to the receiver but have no significant contribution in the IFFT/FFT computation.

4 FFT ALGORITHM

The FFT/IFFT is the most critical part of any signal processing system. The OFDM based transceivers are the most computational intensive blocks of the total system. So an inefficient IFFT/FFT may decrease the overall system response. This paper is based on radix-2 DIF FFT algorithm, which have a divide and conquer approach, where N-points DFT is decomposed into successively small DFTs(with odd and even part separately).

The basic formula for DFT is-

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \quad (1)$$

Where, K is an integer ranging from 0 to N - 1.

The Radix-2 DIT algorithm rearranges the DFT of the function x_n into two parts as shown in equation (2): a sum over the even-numbered indices $n = 2m$ and a sum over the odd-numbered indices $n = 2m+1$.

$$X(K) = \sum_{n=0}^{N/2-1} x_{(2m)} e^{-j2\pi(2m)k/N} + \sum_{n=0}^{N/2-1} x_{(2m+1)} e^{-j2\pi(2m+1)k/N} \quad (2)$$

One can factor a common multiplier out of the second sum in the equation. It is the two sums are the DFT of the even-indexed part and the DFT of odd-indexed part of the function. Denote the DFT of the Even-indexed inputs by and the DFT of the Odd-indexed inputs by and it is shown in equation (3):

$$X(K) = \underbrace{\sum_{n=0}^{N/2-1} x_{(2m)} e^{-j2\pi nk/(N/2)} + e^{-j2\pi k/N}}_{\text{DFT of even indexed part of } X_{2m}} + \underbrace{\sum_{n=0}^{N/2-1} x_{(2m+1)} e^{-j2\pi nk/(N/2)}}_{\text{DFT of Odd indexed part of } X_{2m}} \quad (3)$$

There are basically two classes of FFT algorithms:

- Decimation In Time (DIT) algorithm.
- Decimation In Frequency (DIF) algorithm.

In the decimation-in-frequency approach, the frequency samples of the DFT are decomposed into smaller subsequences. In terms of computational work load, there would appear to be very little to choose between the DIF and DIT algorithms. Both perform exactly the same number of butterflies. Each butterfly requires exactly one complex multiply and two complex additions. Both algorithms use exactly the same number of butterflies. The most significant difference between simple DIF and DIT algorithms is that DIF starts with normal order input and generates bit reversed order output. In contrast, DIT starts

with bit reversed order input and generate normal order output so use DIF for the forward transform and DIT for the inverse transform. For the inverse transform it is necessary to conjugate the twiddle factors.

4.1 Pipeline Architecture

There are various ways to implement these three stages whereas in this project pipeline architecture is used for FFT/IFFT design. Throughput of any system consisting of a series of operations is limited by the single slowest operation in the complete series. So if this single slow operation is further broken down into a number of stages, the intermediate results will be available faster for the next stage. Each stage result(s) are stored in registers and the next stage can start operating on these results of the previous stage & simultaneously the previous stage can start operating on the next succeeding set of data. Thus simultaneously no stage of the pipeline is free. DSPs can take advantage of pipelining as the data inputted in a DSP is always continuous. In process which has many level deep pipelines, the startups and terminations are slow while the pipeline fills or empties but when the pipeline is filled all results will appear continuously [10].

Here three stages working simultaneously and provide complete output in a one or two clock cycles. The block diagram of process block is shown in Fig.6.

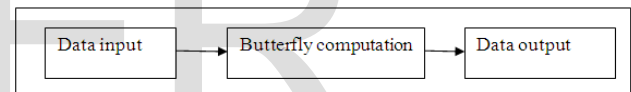


Fig.6: Data Process Block

This block consist of data input, butterfly computation and data output. The data is read in every rising edge of clock and stored in the memory register. Butterfly computation block compute the stored data before going to data output process. The data is kept in the register before it is read out. The FFT radix-2 processor architecture consist of a butterfly architecture, memory register, control circuit, serial to parallel and parallel to serial converter. Twiddle factor are stored as signed fixed point word. The block diagram representation of FFT architecture design is shown in Fig.7.

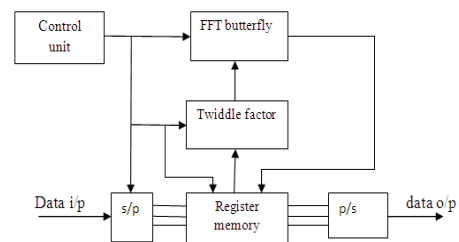


Fig.7: FFT Processor Architecture

The most important element in FFT processor is a butterfly structure. It takes two signed fixed-point data from memory register and computes the FFT algorithm. The output results

are written back in same memory location as the previous input stored. This method is called in-placement memory storage whereby it can reduce the hardware utilization. The butterfly architecture is shown in Fig.7. Multiplication by the twiddle factor is done at the beginning or at the end of the butterfly based on pattern used (DIF or DIT). The multiplier forms the partial product of the complex multiplication and produces two times bigger the n input bit. Shift register would shift the bits to avoid overflow issue. Output of this butterfly would be kept in the register for the subsequent stage.

The FFT processor event is determined by the control circuit depending on the feedback it receives from the surrounding unit. Moore machine approach is adapted whereby the output signal dependent to the value of next state. This design functions as asynchronous design which controlled by clk signal. The input signal rst is used to reset the FFT processor including the input buffer which holds data for next stage.

As processed to higher-point FFTs, the structure for computing the FFT becomes more complex and the need for an efficient complex multiplier to be incorporated within the butterfly structure arises. Hence this paper proposes an algorithm for an efficient complex multiplier that overcomes the complication of using complex numbers throughout the process. A radix-2 FFT can be efficiently implemented using a butterfly processor which includes, besides the butterfly itself, an additional complex multiplier for the twiddle factors. A radix-2 butterfly processor consists of a complex adder, a complex subtractor, and a complex multiplier for the twiddle factors. The complex multiplication with the twiddle factor is often implemented with four real multiplications and two add/subtract operations.

4.1 Pruning technique

To increase the efficiency of the FFT technique several pruning and different other techniques have been proposed by many researchers. In this paper, a new pruning technique i.e. IZ-TFFTP by simple modification and some changes in the conventional flowchart of FFT [9] and it also includes pipeline techniques to reduce the total execution time.

Zero tracing- as in wide band communication system a large portion of frequency channel may be unoccupied by the licensed user, so no. of zero valued inputs are much greater than the non-zero valued inputs in a FFT/IFFT operation at the transceiver. Then this algorithm will give best response in terms of reduced execution time by skip the operation as they are dual node to each other reducing the number of complex computation required for twiddle factor calculation. IZTFFTP have a strong searching condition, which have a 2-D array for storing the input & output values after every iteration of butterfly calculation. In a input searching result whenever it found "zero" at any input, simply omit that calculation by considering following two useful condition:

- ⊙ = '0' operation [full pruning]
- = single operation [partial pruning]
- = full butterfly operation [no pruning]

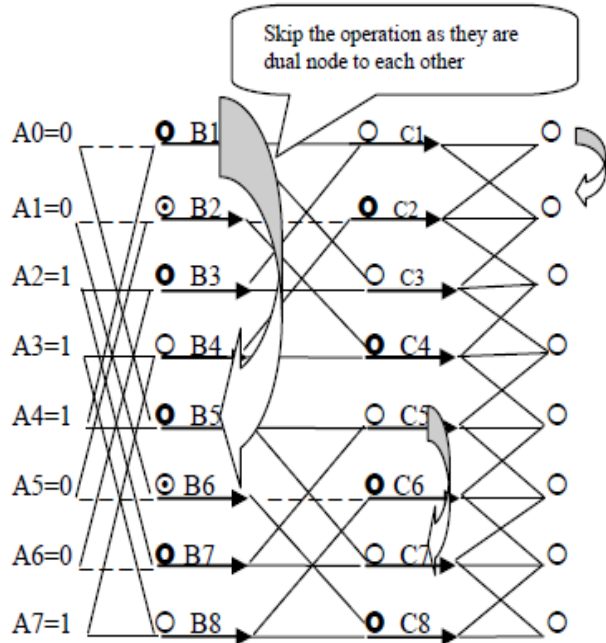


Fig.8: 8-point DIF-FFT flow graph

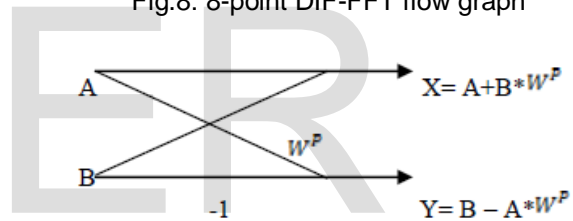


Fig.9: small butterfly unit of a dual node pair

Half Butterfly computation or partial pruning-The basic computation part of FFT is the butterfly calculation. From fig.8 shows a single part of a standard butterfly unit. In the fig.9, A & B two complex inputs are dual node to each other. A full butterfly calculation requires 4 complex multiplications and 6 complex additions/subtractions [4].

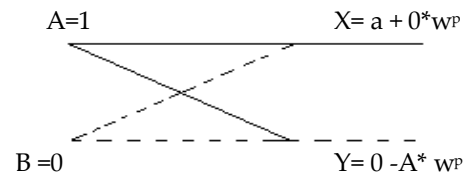


Fig.10: partial pruning structure

But in fig.10 showed that when any of the input value is zero of a dual node pair, then the output of that particular node is the simple copied version of the input. So for a partial pruning calculation 4 complex multiplication and 2 addition is re-

quired which is the basic requirement for a single twiddle factor calculation also.

“0” operation or complete pruning- Fig.11 shows the part where both the input of the dual node pair is zero then outputs obtained from the mathematical calculation of equation 6-7 is also zero.

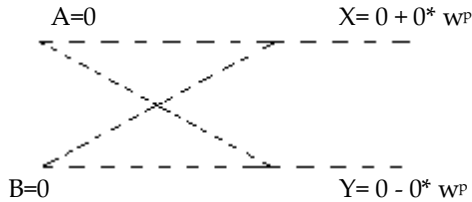


Fig.11: complete pruning structure

In such cases where number of zero is remarkably very large IZTFFTP works most effectively and program will automatically goes to the next node for required computation omitting these unnecessary complex calculations. In this way the total no of complex multiplications and additions are reduced remarkably and the execution time.

5 RESULTS

The coding for OFDM module was done using VHDL as the hardware description language. The code was simulated using ModelSim software. The Synthesis, place and route was performed using Xilinx ISE. The device utilization summaries are as shown below.

Table 1: Device utilization of a transmitter module

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2392	3584	66%
Number of Slice Flip Flops	775	7168	10%
Number of 4 input LUTs	4120	7168	57%
Number of bonded IOBs	19	141	13%
Number of MULT18X18s	12	16	75%
Number of GCLKs	1	8	12%

Table 2: Device utilization of a receiver module

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1817	3584	50%
Number of Slice Flip Flops	1027	7168	14%
Number of 4 input LUTs	2761	7168	38%
Number of bonded IOBs	22	141	15%
Number of MULT18X18s	6	16	37%
Number of GCLKs	4	8	50%

6 CONCLUSION

This paper presents an effective implementation of Fast Fourier Transform on FPGA. The pipeline architecture is used

to implement 8 point radix-2 FFT/IFFT and provides a better speed with an execution time of one clock cycles which in turn provides high speed operation for OFDM modem. Results shows IZTFFTP is much efficient than ordinary FFT algorithm as it takes very less time to compute where number of Zero valued inputs/outputs are greater than the total number of non Zero terms, with maintaining a good trade-off between time and space complexity, and it is also independent to any input data sets.

7 REFERENCES

- [1] CHANG, R. W. High-speed multi-channel data transmission with band limited orthogonal signals. Bell System Technology Journal, v. 45, p. 1775-1796, December 1966.
- [2] J. D. Poston and W. D. Horne, "Discontinuous OFDM considerations for dynamic spectrum access in idle TV channels," in Proc. IEEE Int. Symp. New Frontiers Dynamic Spectra. Access Networks, vol. 1, (Baltimore, MD, USA), pp. 607-610, Nov.
- [3] R. Rajbanshi, A. M. Wyglinski, and G. J. Minden, Cognitive Radio Communication Networks, ch. 5. Springer-Verlag, 2007.
- [4] J. D. Markel, "FFT Pruning," IEEE Trans. Audio Electroacoust., vol. 19, pp. 305 - 311, Dec. 1971.
- [5] D. P. Skinner, "Pruning the Decimation in time FFT algorithm," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 24, Apr. 1976, pp.193 - 194
- [6] T. V. Sreenivas and P. Rao, "FFT algorithm for both input and output Pruning," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process, vol. 27, June 1979, pp. 291 - 292.
- [7] R. G. Alves, P. L. Osorio, and M. N. S. Swamy, "General FFT Pruning Algorithm," in Proc. 43rd IEEE Midwest Symp. Circuits and Systems, vol. 3, Aug. 2000, pp. 1192 - 1195.
- [8] H. V. Sorensen and C. S. Burrus, "Efficient computation of the DFT with only a subset of input or output points," IEEE Trans. Signal Processing, vol. 41, pp. 1184 - 1200, Mar. 1993.
- [9] E. Oran. Brigham, "The Fast Fourier Transform and Its Applications" Prentice Hall Publication, 1988, ISBN: 0133075052.
- [10] Vinay Savla, —DSP Architecture for System Design, IIT-Mumbai, November 2002.
- [11] Ramjee Prasad, —OFDM for Wireless Communications Systems—, Artech House Universal Personal Communications series, 2004.