Implementation of efficient FFT Algorithm for WiMAX applications

K. Sandhya Rani¹, Kishore Sonti²

M.Tech VLSI, Dept. of ECE, Satyabama University, Chennai, India¹
Professor, Dept. of ECE, Satyabama University, Chennai, India²

Abstract: The advancements in VLSI technologies for Digital signal processing have opened many windows in past few years for designing real time applications, using efficient algorithms with custom chips. Multicarrier modulation techniques (OFDM) are rapidly moving in modern communication systems. Mobile WiMAX (Worldwide Interoperability for Microwave Access) or 802.16e standards use this OFDM Technology. The FFT and IFFT pairs are most important blocks used to modulate and demodulate the data constellation on the subcarriers in the OFDM (Orthogonal Frequency-Division Multiple Access) system. In this paper we presented high level implementation of a FFT/IFFT to increase the speed and performance of OFDM Modulator and Demodulator. This paper presents the 256-point Radix-4 FFT algorithm. The design has been coded in VHDL and simulated in MODELSIM, targeted into Xilinx Spartan3 FPGAs.

Keywords: WiMAX, OFDM, FFT, IFFT, FPGA

I. INTRODUCTION

WiMAX—which stands for Worldwide Interoperability for Microwave Access—is bringing the wireless and Internet revolutions to portable devices across the globe. The WiMAX modules utilize the OFDMA scheme in their physical layer of communication. OFDM exploits the frequency diversity of the multipath mobile broadband channel by coding and interleaving the information across the subcarriers prior to transmission. The OFDM modulation is cost effectively realized by the Inverse Fast Fourier Transform (IFFT) that enables the use of a large number of subcarriers up to 1024. According to the Mobile WiMAX standard, OFDM is the OFDMA symbol is extended by its cyclic prefix followed by digital-to-analog (D/A) conversion at the Transmitter. At the receiver end, after analog-to-digital (A/D) conversion, the cyclic prefix is discarded and OFDM demodulation is applied through the Fast Fourier Transform (FFT). The OFDM structure with IFFT/FFT is shown in Fig.1.

![Fig.1 Structure of simple OFDM system](image)

One of the key components of OFDM system is the Fast Fourier Transform (FFT). There are more and more communication systems require higher points FFT and higher symbol rates. The requirement establishes challenges for low power and high speed FFT design with large points. The FFT algorithm eliminates the redundant calculation which is needed in computing Discrete Fourier Transform (DFT) and is thus very suitable for efficient hardware implementation. By considering all these factors we proposed the work with high efficiency Radix-4 FFT with 256 point.

II. FFT ALGORITHM

A. Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT) is one of the algorithm used for the complex mathematical computations in digital signal processing, for the sequence of data \{x(n)\} of length N.

The Discrete Fourier Transform (DFT) X(k) of N points is given by

\[ X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad \{0 \leq k < N-1\} \]

where \( W_N^{nk} = \exp\left(\frac{-2\pi ik}{N}\right) \) is the twiddle factor.

X(k) and x(n) are frequency-domain sequences and time-domain sequence.

N is the no. of points.

It is observed that to compute total N values, \( N^2 \) complex multiplications and \( N^2-N \) complex additions are required. Also DFT is inefficient for exploiting symmetry and periodicity properties of phase factor.

B. Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is more efficient algorithm for computing complex multiplications. The basic idea of the algorithm is to divide the N-point DFT into M, N/M point DFTs, hence if M=2 then it is divided into two N/2 DFTs. These are called the radix-2. Similarly we have radix-4, 8, 16...etc. The efficiency of algorithm increases with Radix.

For example, the number of complex multiplications required using direct computation

Copyright to IJARCCE

DOI 10.17148/IJARCCE.2016.53250
for $N = 64$, is

$$N^2 = 64^2 = 4096$$

The number of complex multiplications can be reduced by using FFT,

$$(N/2) \log_2 N = (64/2) \log_2 64 = 192$$

with speed improvement factor is 21.3%

Instead of the direct implementation of the equation, the FFT algorithm factorizes a large DFT recursively into many small point DFT in order to reduce the overall operations. There are two well-known types of decompositions in FFT called Decimation In Time (DIT) and Decimation In Frequency (DIF). The only difference between these two algorithms is that, DIT starts with bit reverse order input and generates normal order output. Nevertheless DIF starts with normal order input and generates bit reverse order output.

### III. RADIX-4 FFT ARCHITECTURE

Radix-4 FFT algorithm is that when the number of data points $N$ in the DFT is a power of 4 (i.e. $N = 4$). In Radix-4 algorithm the $N$-point input sequence split or decimate into four sub sequences,

$$x(4k), x(4k+1), x(4k+2), x(4k+3), n = 0,1,\ldots,N/4$$

$$X(4k) = \sum_{n=0}^{N/4-1} x(n) + jx(n + \frac{N}{4}) + jx(n + \frac{N}{2}) + jx(n + \frac{3N}{4}) W_N^{nk}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} x(n) - jx(n + \frac{N}{4}) + jx(n + \frac{N}{2}) + jx(n + \frac{3N}{4}) W_N^{nk}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} x(n) + jx(n + \frac{N}{4}) - jx(n + \frac{N}{2}) + jx(n + \frac{3N}{4}) W_N^{nk}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} x(n) - jx(n + \frac{N}{4}) - jx(n + \frac{N}{2}) + jx(n + \frac{3N}{4}) W_N^{nk}$$

Thus the four $N/4$-point DFTs obtained from the above equation are combined to yield the $N$-point DFT. The expression for combining the $N/4$-point DFTs defines a radix-4 decimation-in-time butterfly, which can be expressed in matrix form.

$$\begin{bmatrix}
X(0,q) \\
X(1,q) \\
X(2,q) \\
X(3,q)
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 & 1 \n1 & -j & -1 & j \n1 & -1 & 1 & -1 \n1 & j & -1 & -j
\end{bmatrix}
\begin{bmatrix}
W_N^{0 \cdot q} F(0,q) \\
W_N^{-1 \cdot q} F(1,q) \\
W_N^{2 \cdot q} F(2,q) \\
W_N^{3 \cdot q} F(3,q)
\end{bmatrix}$$

Basic butterfly diagram for Radix-4 is as shown below.

![Butterfly diagram for Radix-4 FFT](image)

Signal flow diagram for Radix-4 FFT (Decimation in Frequency) is shown below.

![Signal flow diagram for Radix-4 FFT (DIF)](image)

Radix-4 decimation in frequency with $N$ points is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} , 0 \leq k \leq N - 1$$

$$= \sum_{n=0}^{N/4-1} x(n) W_N^{nk} + \sum_{n=N/4}^{N/2-1} x(n) W_N^{nk}$$

$$+ \sum_{n=N/2}^{3N/4-1} x(n) W_N^{nk} + \sum_{n=3N/4}^{N-1} x(n) W_N^{nk}$$

$$= \sum_{n=0}^{N/4-1} x(n) W_N^{nk}$$

$$+ W_N^{Nk/4} \sum_{n=0}^{N/4-1} x(n + \frac{N}{4}) W_N^{nk}$$

$$+ W_N^{Nk/2} \sum_{n=0}^{N/4-1} x(n + \frac{N}{2}) W_N^{nk}$$

$$+ W_N^{3Nk/4} \sum_{n=0}^{N/4-1} x(n + \frac{3N}{4}) W_N^{nk}$$

The Twiddle factors are,

$$W_N^{Nk/4} = -(j)^k , W_N^{Nk/2} = -(1)^k \text{ and } W_N^{3Nk/4} = (j)^k$$

Thus,

$$X(k) = \sum_{n=0}^{N/4-1} (x(n) + (-1)^k x(n + \frac{N}{4})$$

$$+ (-1)^k x(n + \frac{N}{2})$$

$$+ (j)^k x(n + \frac{3N}{4})) W_N^{nk}$$

### IV. IMPLEMENTATION OF FFT/IFFT IN OFDM SYSTEM

The fundamental principle of the OFDM system is to decompose the high rate data stream (bandwidth=$W$) into $N$ lower rate data streams and then to transmit them simultaneously over a large number of sub carriers.
The IFFT and the FFT are used in modulating and demodulating the data constellations respectively on the orthogonal sub-carriers. In an OFDM system, the Transmitter section contains IFFT block and Receiver contains FFT⁻¹ (IFFT) block. The FFT processor must finish the transform within 312.5 ns to serve the purpose in the OFDM system. Our FFT architecture effectively fits into the system since it has a minimum required time period of 10.827 ns.

In practice, OFDM systems are implemented using a combination of fast Fourier Transform (FFT) and inverse fast Fourier Transform (IFFT) blocks that are mathematically equivalent versions of the DFT and IDFT, respectively, but more efficient to implement. An OFDM system treats the source symbols (e.g., the QPSK or QAM symbols that would be present in a single carrier system) at the transmitter as though they are in the frequency-domain. These symbols are used as the inputs to an IFFT block that brings the signal into the time domain. The IFFT takes in N symbols at a time where N is the number of subcarriers in the system. Each of these N input symbols has a symbol period of T seconds. The basic functions for an IFFT are N orthogonal sinusoids.

These sinusoids each have a different frequency and the lowest frequency is DC. Each input symbol acts like a complex weight for the corresponding sinusoidal basis function. Since the input symbols are complex, the value of the symbol determines both the amplitude and phase of the sinusoid for that subcarrier. The IFFT output is the summation of all N sinusoids. Thus, the IFFT block provides a simple way to modulate data onto N orthogonal subcarriers. The block of N output samples from the IFFT make up a single OFDM symbol. The length of the OFDM symbol is NT where T is the IFFT input symbol period mentioned above.

After some additional processing, the time-domain signal that results from the IFFT is transmitted across the channel. At the receiver, an FFT block is used to process the received signal and bring it into the frequency domain. Ideally, the FFT output will be the original symbols that were sent to the IFFT at the transmitter. When plotted in the complex plane, the FFT output samples will form a constellation, such as 16-QAM. However, there is no notion of a constellation for the time-domain signal. When plotted on the complex plane, the time-domain signal forms a scatter plot with no regular shape. Thus, any receiver processing that uses the concept of a constellation (such as symbol slicing) must occur in the frequency-domain.

V. SIMULATED RESULTS

VHDL Simulation for the Radix-4 FFT 256 point is shown below. The presented OFDM system is shown in the above is designed using VHDL and synthesized using Xilinx Project Navigator Xilinx ISE tools. Simulation results are verified by using ISE Simulator.

In practice, OFDM systems are implemented using a combination of fast Fourier Transform (FFT) and inverse fast Fourier Transform (IFFT) blocks that are mathematically equivalent versions of the DFT and IDFT, respectively, but more efficient to implement. An OFDM system treats the source symbols (e.g., the QPSK or QAM symbols that would be present in a single carrier system) at the transmitter as though they are in the frequency-domain. These symbols are used as the inputs to an IFFT block that brings the signal into the time domain. The IFFT takes in N symbols at a time where N is the number of subcarriers in the system. Each of these N input symbols has a symbol period of T seconds. The basic functions for an IFFT are N orthogonal sinusoids.

These sinusoids each have a different frequency and the lowest frequency is DC. Each input symbol acts like a complex weight for the corresponding sinusoidal basis function. Since the input symbols are complex, the value of the symbol determines both the amplitude and phase of the sinusoid for that subcarrier. The IFFT output is the summation of all N sinusoids. Thus, the IFFT block provides a simple way to modulate data onto N orthogonal subcarriers. The block of N output samples from the IFFT make up a single OFDM symbol. The length of the OFDM symbol is NT where T is the IFFT input symbol period mentioned above.

After some additional processing, the time-domain signal that results from the IFFT is transmitted across the channel. At the receiver, an FFT block is used to process the received signal and bring it into the frequency domain. Ideally, the FFT output will be the original symbols that were sent to the IFFT at the transmitter. When plotted in the complex plane, the FFT output samples will form a constellation, such as 16-QAM. However, there is no notion of a constellation for the time-domain signal. When plotted on the complex plane, the time-domain signal forms a scatter plot with no regular shape. Thus, any receiver processing that uses the concept of a constellation (such as symbol slicing) must occur in the frequency-domain.

V. SIMULATED RESULTS

VHDL Simulation for the Radix-4 FFT 256 point is shown below. The presented OFDM system is shown in the above is designed using VHDL and synthesized using Xilinx Project Navigator Xilinx ISE tools. Simulation results are verified by using ISE Simulator.
TABLE I  DEVICE UTILIZATION SUMMERY

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Slices</td>
<td>466</td>
<td>4656</td>
<td>10%</td>
</tr>
<tr>
<td>No. of Slice Flip-flops</td>
<td>306</td>
<td>9312</td>
<td>3%</td>
</tr>
<tr>
<td>No. of 4 input LUT’s</td>
<td>883</td>
<td>9312</td>
<td>9%</td>
</tr>
<tr>
<td>No. of bonded IOB’s</td>
<td>69</td>
<td>92</td>
<td>75%</td>
</tr>
<tr>
<td>No. Of GCLK’s</td>
<td>1</td>
<td>24</td>
<td>4%</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

Since there is a high demand for the efficient algorithms in Digital Signal Processing the proposed FFT will work efficiently works in OFDM system. The designed FFT is coded in VHDL and simulated for FFT Radix-4 for 256 points. The simulations of the FFT samples were verified using simulated tool MODELSIM.

The parallel processing of FFT has been proposed to be used in multiprocessor algorithms. When an FFT is implemented in special purpose hardware, errors and constraints due to finite word lengths are unavoidable. While deciding the word length needed for an FFT, these quantization effects must be considered.

REFERENCES