IMPLEMENTATION OF FFT ALGORITHM USING FLOATING POINT NUMBERS IN WIMAX COMMUNICATION SYSTEM

P S RAJA KUMAR¹, G. SURYA NARAYANA REDDY²
¹DEPT OF ECE, SVCET, CHITTOOR, A.P. INDIA
²rk04353@gmail.com, ²principal.iitcell@gmail.com

ABSTRACT
Mobile WiMAX (Worldwide Interoperability for Microwave Access) or 802.16e standard was ratified by the IEEE in late 2005 as a potential to emerge as a real viable competitor to existing 3G technologies. Mobile WiMAX uses an OFDMA™ technology called 1K-FFT. Orthogonal Frequency-Division Multiple Access (OFDMA) is a multi-user version of the popular Orthogonal frequency-division multiplexing (OFDM) digital modulation scheme. In the widely used OFDM systems, the FFT and IFFT pairs are used to modulate and demodulate the data constellation on the subcarriers. This paper presents a high level implementation of a high performance FFT for OFDM Modulator and Demodulator. The design has been coded in Verilog and targeted into Xilinx VERTEX4 FPGAs. Radix-2^2 Algorithm is proposed and used for the OFDM communication system. This algorithm has the same multiplicative complexity as the radix-2^2 algorithm, but retains the butterfly structure of radix-2 algorithm. Increase in number of butterflies makes the system to work in efficiency mode and also computation of twiddle factors is very easy and it is implemented by using floating point multiplier. It reduces time complexity and increases computation speed. Here we implemented for 32 bit data length using number of desired clock cycles N/2 and required number of butterflies structures.

Keywords—WIMAX, FFT, IFFT, FPGA, OFDMA, RADIX 2^2FFT, VERILOG HDL.

1. INTRODUCTION
The true Mobile WiMAX standard of 802.16e is divergent from Fixed WiMAX. Here we using radix 2^2 algorithm and its utilization in OFDM based communication system[1]. While clearly based on the same OFDM base technology adopted in 802.1, the 802.16e version is designed to deliver service across many more sub-channels than the OFDM 256-FFT. It is important to note that both standards support single carrier, OFDM 256-FFT and at least OFDMA 1K-FFT. OFDM technology is used for many communication systems such as Asymmetric digital subscriber line (ADSL), Wireless Local Area Network (WLAN) or Multimedia Communication Services. the wimax module utilize the OFDMA scheme in their physical layer of communication[2]. One of the key components in OFDM system is the Fast Fourier Transform (FFT)[3][4]. 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[6][9]. In addition to computing efficient DFT, the FFT also finds applications in linear filtering, digital spectral analysis and correlation analysis, Ultra Wide Band (UWB) applications, etc. A hardware oriented radix-2^2 algorithm is developed by integrating a twiddle factor decomposition technique in divide and conquer approach to form a spatially regular Signal Flow Graph (SFG). Mapping the algorithm to the cascading delay feedback structure leads to the proposed architecture[6]. The next section describes architecture & design methodology, followed by its implementation in VERILOG Hardware Description Language (VERILOG) code and utilization, performance and implementation in OFDM systems. Finally we conclude with a comparison of hardware requirement of R2^2SDF and several other popular pipeline architectures.
A high level implementation of a high performance FFT for OFDM modulator and demodulator is presented in this work[7][8]. The design has been coded in Verilog and targeted into Xilinx VERTEX4 field programmable gate arrays. Radix-2^2 algorithm[10] is proposed and used for the OFDM communication system. The designed FFT is implemented and applied to Fixed WiMAX (IEEE 802.16d) and Mobile WiMAX (IEEE 802.16e) communication standards[11]. The results are tabulated and the hardware parameters are compared.

2. ARCHITECTURE AND DESIGN METHODOLOGY OF THE PROPOSED SYSTEM
2.1. Radix-2^2 Decimation in Frequency FFT Algorithm
A useful state-of-the-art review of hardware architectures for FFTs was given by He et al[12], and different approaches were put into functional blocks with unified terminology. From the Definition of DFT of size N [13]
where \( WN \) denotes the primitive \( N \)th root of unity, with its exponent evaluated modulo \( N \), \( x(n) \) is the input sequence and \( X(k) \) is the DFT. He applied a 3-dimensional linear index map,

\[
\begin{align*}
n &= \left( \frac{N}{2} n_1 + \frac{N}{4} n_2 + n_3 \right) \mod N \\
k &= \left( \frac{N}{3} k_1 + 2k_2 + 4k_3 \right) \mod N
\end{align*}
\]

and Common factor algorithm (CFA) to derive a set of 4 DFTs of length \( N/4 \) as,

\[
X(k_1+2k_2+4k_3) = \sum_{n_0} H(k_1,k_2,n_0) WN^{(k_1+2k_2+n_0)} \mod N
\]

where \( n_1,n_2,n_3 \) are the index terms of the input sample \( n \) and \( k_1,k_2,k_3 \) are the index terms of the output sample \( k \) and where \( H(k_1,k_2,k_3) \) is expressed in below equation

\[
H(k_1,k_2,n_0) = \left[ x(n_0) + (-1)^k x \left( n_0 + \frac{N}{2} \right) \right] + \\
\left[ x \left( n_0 + \frac{N}{4} \right) + (-1)^k x \left( n_0 + \frac{3N}{4} \right) \right]
\]

The equation represents the first two stages of butterflies with only trivial multiplications. Full multipliers are required after the three butterflies in order to compute the product of the decomposed twiddle factor \( n_3 (k_1 2 k_2) N W \) in eqn. Note the order of the twiddle factors is different from that of radix-4 algorithm. Applying this CFA procedure recursively to the remaining DFTs of length \( N/4 \) in eqn, the complete radix-2^2 Decimation-in-frequency (DIF FFT) algorithm is obtained. The corresponding FFT flow graph for \( N=32 \) which involves only real-imaginary swapping and sign inversion [12].

2.2. Radix2^2 FFT Architecture

Mapping radix-2^2 DIF FFT algorithm derived to the radix-2 SDF architecture, a new architecture of R2^2SDF approach is obtained[10]. Outlines an implementation of the R2^2SDF architecture for \( N=32 \), note the similarity of the data-path to R2SDF and the reduced number of multipliers. The implementation uses three types of butterflies; one identical to that in R2SDF, the other contains also the logic to implement the trivial twiddle factor multiplication. Due to the spatial regularity of Radix-2^2 algorithm, the synchronization control of the processor is very simple. A (log2\( N \))-bit binary counter serves two purposes: synchronization controller and address counter for twiddle factor reading in each stage. With the help of the butterfly structures shown in Fig 3, the scheduled operation of the R2^2SDF processor in Fig 2 is as follows. On first \( N/2 \) cycles, the 2-to-1 multiplexers in the first butterfly module switch to position “0”, and the butterfly is idle. The input data from left is directed to the shift registers until they are filled. On next \( N/2 \) cycles, the multiplexers turn to position “1”, the butterfly computes a 2-point DFT with incoming data and the data stored in the shift registers. The butterfly outputs \( Z1(n) \) and \( Z1(n+N/2) \) are computed according to the equation. \( Z1(n) \) is sent to apply the twiddle factors, and \( Z1(n+N/2) \) is sent back to the shift registers to be “multiplied” in still next \( N/2 \) cycles when the first half of the next frame of time sequence is loaded in. The operation of the second and third butterfly is similar to that of the first one, except the “distance” of butterfly input sequence are just \( N/4 \) and the trivial twiddle factor multiplication has been implemented by real-imaginary swapping with a commutator and controlled add/subtract operations, which requires two bit control signal from the synchronizing counter.
data then goes through a full complex multiplier, working at 75% utility, accomplishes the result of first level of radix-2^2 DFT word by word. Further processing repeats this pattern with the distance of the input data decreases by half at each consecutive butterfly stages. After N-1 clock cycles, the result of the complete DFT transform streams out to the right, in bi transversed order. The next frame of transform can be computed without pausing due to the pipelined processing.

2: Pipeline architecture for radix 2^2 with a period difference of N/2 cycles

Fig3. Implementation of BFS in pipelined architecture.

2.3. Floating Point Multiplication Algorithm

Floating point multiplication algorithm[5] which is used for complex multiplication for developing the twiddle factor in FFT as follows:-

As stated in the introduction, normalized floating point numbers have the form of Z= (-1S) * 2 (E Bias) * (1.M).

To multiply two floating point numbers the following is done:

1. Multiplying the significand; i.e. (1.M1*1.M2)
2. Placing the decimal point in the result
3. Adding the exponents; i.e. (E1 + E2 – Bias)
4. Obtaining the sign; i.e. s1 xor s2
5. Normalizing the result; i.e. obtaining 1 at the MSB of the results’ significand
6. Rounding the result to fit in the available bits
7. Checking for underflow/overflow occurrence

3. IMPLEMENTATION IN VERILOG

VERILOG Hardware Description Language (VERILOG) was introduced by Gateway Design Automation in 1984 as a proprietary hardware description and simulation language[14][15]. VERILOG synthesis tools can create logic-circuit structures directly from VERILOG behavioral descriptions, and target them to a selected technology for realization. Using VERILOG, we can design, simulate, and synthesize anything from a simple combinational circuit to a complete microprocessor system on a chip. VERILOG started out as documentation and modeling language, allowing the behavior of digital-system designs to be precisely specified and simulated[14][15] and language specification allows multiple modules to be stored in a single text file. All these features of VERILOG will help better in simulation and synthesis of our proposed architecture. The R2^2SDF presented above has been fully coded in VERILOG Hardware Description Language (VERILOG). Once the design is coded in VERILOG, the Modelsim 6.2g[15] compiler and the Xilinx Foundation ISE Environment 14.3[16] generate a net-list for FPGA configuration. The net list can then be downloaded into the FPGA using the same Xilinx tools and Texas Instruments prototyping board. From the architecture of R2^2SDF, the butterfly blocks BF2I and BF2II are described as building blocks in VERILOG code. Here we use floating point multiplication[5] in which it uses floating multiplication algorithm for signed binary numbers is used for complex multipliers which is explained in above. Thus, the overall latency of the real implementation varies as the processing word length changes [10]. Look-up-table (LUT) based Random Access Memories (RAMs) and Flip-Flops are used to implement feedback memory of the very last stages where are the RAM blocks in the FPGA are used for the rest of the stages. Similarly, LUT-based Read Only Memories (ROMs) are used to implement twiddle ROMs of the very last stages whereas Block RAMs are used for the rest of stages. The FFT
is heavily pipelined\cite{17}\cite{18} to achieve as highest clock frequency as possible. Twiddle factors are generated by an external program and embedded to the verilog code.

4. IMPLEMENTATION OF THE PROPOSED FFT FOR N=32 IN OFDM COMMUNICATION 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 subcarriers\cite{7}. The IFFT and the FFT are used for, respectively, modulating and demodulating the data constellations on the orthogonal subcarriers. In an OFDM system, the transmitter and receiver blocks contain the FFT modules. 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.

![Fig4. OFDM module a)OFDM transmitter b)OFDM receiver](image)

Orthogonal subcarriers, with baseband data on each subcarrier being independently modulated commonly using some type of quadrature amplitude modulation (QAM) or quadrature phase shift keying (QPSK) \cite{8}. This composite baseband signal is typically used to modulate a main RF carrier. $s[n]$ is a serial stream of binary digits. By inverse multiplexing, these are first de multiplexed into $N$ parallel streams, and each one mapped to a (possibly complex) symbol stream using some modulation constellation QPSK. Note that the constellations may be different, so some streams may carry a higher bit-rate than others. The receiver picks up the signal $r(t)$, which is then quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. This also creates signals centered on $2f_c$, so low-pass filters are used to reject these. The baseband signals are then sampled and digitized using analogue-to-digital converters (ADCs), and a forward FFT is used to convert back to the frequency domain. Orthogonal Frequency-Division Multiple Access (OFDMA) is a multi-user version of the popular Orthogonal frequency division multiplexing (OFDM) digital modulation scheme. Multiple access is achieved in OFDMA by assigning subsets of subcarriers to individual. This allows simultaneous low data rate transmission from several users \cite{7}.

![Fig5. OFDM Subcarrier Pattern](image)

OFDMA can be seen as an alternative to combining OFDM with time division multiple access (TDMA) or time-domain statistical multiplexing, i.e. packet mode communication. Low-data-rate users can send continuously with low transmission power instead of using a “pulsed” high-power carrier. OFDMA is considered as highly suitable for road band wireless networks, due to advantages including scalability and MIMO-friendliness, and ability to take advantage of channel frequency selectivity\cite{7}\cite{8}.

<table>
<thead>
<tr>
<th>Wireless standards</th>
<th>Wi-Fi 802.11</th>
<th>WI-MAX 802.16</th>
<th>MOBILE-FI 802.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Speed</td>
<td>54 Mbps</td>
<td>10 to 100 Mbps</td>
<td>16 Mbps</td>
</tr>
<tr>
<td>Coverage</td>
<td>300 feet</td>
<td>50 Miles</td>
<td>Several Miles</td>
</tr>
<tr>
<td>pros</td>
<td>Speed, Price</td>
<td>Speed, Range, Versatile</td>
<td>Speed, Mobility</td>
</tr>
</tbody>
</table>
5. RESULTS

CONCLUSION

We have implemented the FFT, designed for efficient OFDM communication system, by 32 bits word length and synthesized in Xilinx ISE 14.3 design compiler. The maximum operation frequency is 166 MHz and it consists of 50120 gates. When the constraint of clock changes to 125MHz, gate count reduces to 31050 gates. The gate count report does not include the memory. Our proposed design needs $N \log_4 N +11$ clock cycles to finish an $N$ points FFT. The 802.16d requires 92.4 s for 2K points FFT. The proposed FFT work more than the required to satisfy real-time processing.

Further, we have implemented the FFT, designed for efficient mobile WiMAX, by 32 bits word length and synthesized in Xilinx ISE 14.3 design compiler. The OFDMA symbol period for mobile WiMAX 802.16e is 102.9μsec (including a guard time of 11.4μsec). The results clearly indicate that the processing time of the proposed FFT is below the OFDM symbol period of 102.9μsec, which ensures an appropriate operation for the communication standard 802.16e.
We have proposed a memory based recursive FFT design which has much less gate counts, lower power consumption and higher speed. The proposed architecture has three main advantages: fewer butterfly iteration to reduce power consumption, pipeline of radix-2^2 butterfly to speed up clock frequency, even distribution of memory access to make utilization efficiency in SRAM ports. In summary, the speed performance of our design easily satisfies most application requirements of Fixed 802.16d and Mobile 802.16e WiMAX, which uses OFDMA modulated wireless communication system. Our design also occupies lesser area, hence lower cost and power consumption.

ACKNOWLEDGMENT
I express my heart full thanks towards to my project guide, Mr. G. SURYA NARAYANA REDDY, Associate professor, DEPT OF ECE, SVCET, CHITTOOR, for his bright guidance and useful suggestions, which helped me for completion of project work effectively and successfully.

REFERENCES