

Indian Journal of  
**Engineering**

## Implementation of FFT for WIMAX Applications

**Chandrashekhar M Tavade**

Professor, Electronics and Communication Engineering Department, BKIT, Bhalki-585328, India; Email id: [cmttc1@gmail.com](mailto:cmttc1@gmail.com)

### Publication History

Received: 17 February 2014

Accepted: 11 April 2014

Published: 16 April 2014

### Citation

Chandrashekhar M Tavade. Implementation of FFT for WIMAX Applications. *Indian Journal of Engineering*, 2014, 10(22), 57-62

### ABSTRACT

The IEEE 802.16d communication standard uses orthogonal frequency division multiplexing (OFDM). In the widely used OFDM systems, the fast Fourier transform (FFT) and inverse fast Fourier transform pairs are used to modulate and demodulate the data on the sub-carriers. In this paper, a high level implementation of a high performance FFT for OFDM modulator and demodulator is presented. The design has been coded in Verilog and targeted into Xilinx Spartan3 field programmable gate arrays. Radix-2<sup>2</sup> algorithm is proposed and used for the OFDM communication system. The design of the FFT is implemented and applied to fixed - IEEE 802.16d communication standard. The results are tabulated and the hardware parameters are compared. The proposed architecture is least in number of multipliers used and the memory size, and second to the least in number of adders used.

**Keywords** - Fast Fourier Transform, Orthogonal Frequency Division Multiplexing, Radix Conversion, Verilog.

### 1. INTRODUCTION

OFDM technology is used for many communication systems such as Asymmetric Digital Subscriber Line (ADSL), Wireless Local Area Network (WLAN) or multimedia communication services (T. Lenart et al. 2004). One of the key components in OFDM system is the Fast Fourier Transform (FFT). There are more and more communication systems requiring higher points FFT and higher symbol rates. The requirement establishes challenges for low power and high speed FFT design with large points. In the target application the IEEE 802.16d (fixed WiMAX) standard requires the OFDM symbol rates from 1.75 MHz to 20 MHz and the FFT up to 2048 points (T. Lenart et al. 2004).

There are in general two approaches in implementing the FFT for OFDM processing: the pipeline processing and the memory-based recursive processing. In the search for high performance FFT, this paper presents a pipelined architecture called R<sup>2</sup>SDF, of which the implementation on Field Programmable Gate Arrays (FPGAs) uses Verilog

Hardware Description Language (HDL). The next section describes architecture and design methodology, followed by its implementation in Verilog code and utilization, performance and implementation in OFDM systems. Finally, this project concludes with a comparison of hardware requirement of R<sup>2</sup>SDF and several other popular pipeline architectures.

The proposed work is based on a memory based recursive FFT design which has much less gate counts, lower power consumption, and higher speed. The speed performance of the design easily satisfies most application requirements of IEEE 802.16d communication standard, which uses OFDMA modulated wireless communication system. The design uses fewer gates and hence lower cost and power consumption. WiMAX profiles based on 802.16-2004 are better suited to fixed applications that use directional antennae because OFDM is inherently less complex than SOFDMA. As a result, 802.16-2004 networks may be deployed faster and at a lower cost. In addition, 802.16-2004 WiMAX Forum CERTIFIED products will be available earlier and will be adopted by service providers that plan to deploy a network in the near future. OFDMA gives 802.16e profiles more flexibility when managing different user devices with a variety of antenna types and form factors. It brings a reduction in interference for user devices with omni directional antennas and improved NLOS capabilities that are essential when supporting mobile subscribers (H. Lin et al. 2008). Sub channelization defines sub channels that can be allocated to different subscribers depending on the channel conditions and their data requirements. These gives the operator more flexibility in managing the bandwidth and transmit power, and leads to a more efficient use of resources.

## 2. ARCHITECTURE AND DESIGN METHODOLOGY

### 2.1. Radix-2<sup>2</sup> Decimation in Frequency FFT Algorithm

A useful state-of-the-art review of hardware architectures for FFTs was given in F. Wakerly et al. (2006) and different approaches were put into functional blocks with unified terminology. From the definition of DFT of size N (Y.Zhang et al. 2007),

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

Where  $W_N$  denotes the primitive Nth root of unity, with its exponent evaluated modulo N,  $x(n)$  is the input sequence and  $X(k)$  is the DFT. He F. Wakerly et al. (2006) applied a 3-dimensional linear index map,

$$n = \left( \frac{N}{2} n_1 + \frac{N}{4} n_2 + n_3 \right) \quad (2)$$

$$k = (k_1 + 2k_2 + 4k_3)$$

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

$$X(k_1 + 2k_2 + 4k_3) = \sum_{n=0}^{N-1} \left[ H(k_1, k_2, n_3) W_N^{n_2(k_1 + 2k_2)} \right] W_{N/4}^{n_3 k_3} \quad (3)$$

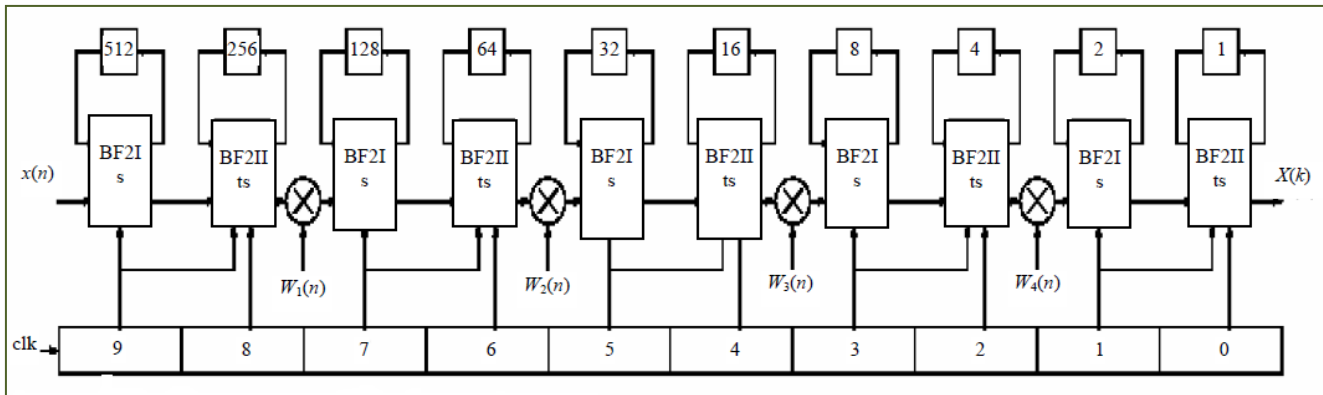
Where  $n_1, n_2, n_3$  are the index terms of the input sample  $n$  and  $k_1, k_2, k_3$  are the index terms of the output sample  $k$  and  $H(k_1, k_2, n_3)$  is expressed in (4).

$$H(k_1, k_2, n_3) = \left[ x(n_3) + (-1)^{k_1} x\left(n_3 + \frac{N}{2}\right) \right] + (-j)^{(k_1 + 2k_2)} X \left[ x\left(n_3 + \frac{N}{4}\right) + (-1)^{k_1} x\left(n_3 + \frac{3N}{4}\right) \right] \quad (4)$$

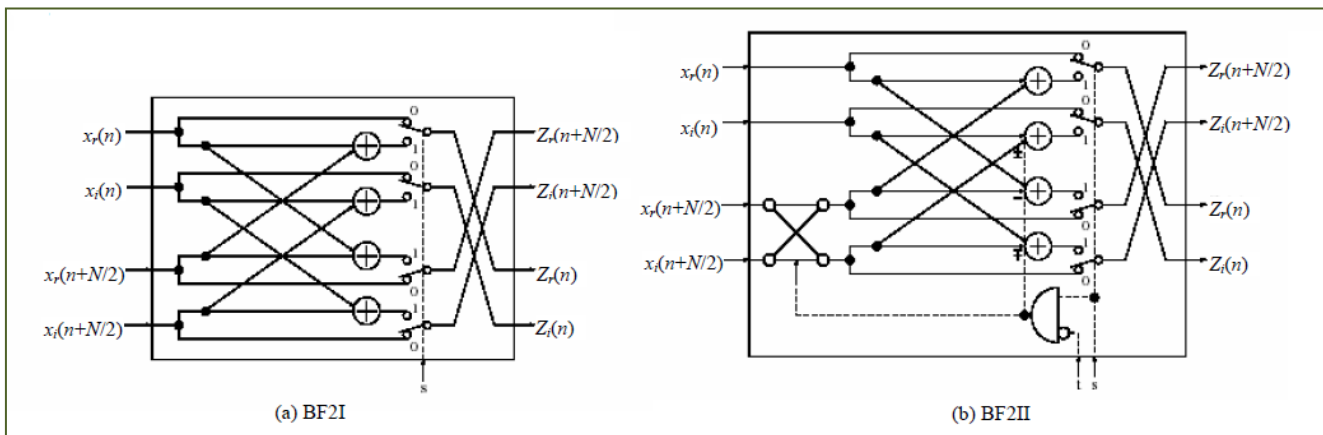
Equation (4) represents the first two stages of butterflies with only trivial multiplications in the signal flow graph (SFG), as Butterfly I (BF2I) and Butterfly II (BF2H). Full multipliers are required after the two butterflies in order to compute the product of the decomposed twiddle factor  $W_N^{n_2(k_1 + 2k_2)}$  in (3). 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 (3), the complete radix-2<sup>2</sup> decimation-in-frequency (DIF) FFT algorithm is obtained. The corresponding FFT flow graph for N-16 is shown in Figure 1.

### 2.2. Radix-2<sup>2</sup> FFT Architecture

Mapping radix-2<sup>2</sup> DIF FFT algorithm derived to the radix-2 SDF architecture, a new architecture of R2SDF approach is obtained (Sankhsawas et al. 2006). Figure 2 outlines an implementation of the R2SDF architecture for N-1024, note the similarity of the data-path to R2SDF and the reduced number of multipliers (Harikrishna et al. 2009). The implementation uses two types of butterflies: one identical to that in R2SDF, the other contains also the logic to implement the trivial twiddle factor multiplication, as shown in Figure 2 (a) and (b) respectively (Sankhsawas et al. 2006). Due to the spatial regularity



**Figure 1**  
R<sup>2</sup> SDF Pipeline FFT Architecture for N=1024



**Figure 2**  
Butterfly Structure for R<sup>2</sup> SDF FFT

of radix-2<sup>2</sup> algorithm, the synchronization control of the processor is very simple. A (log<sub>2</sub>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 Figure 2, the scheduled operation of the R22SDF processor in Figure 1 is as shown. On the first N/2 cycles, the 2-to-1 multiplexers in the first butterfly module switch to position "O", 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 "I", the butterfly computes a 2-point DFT with incoming data and the data stored in the shift registers.

$$\begin{aligned}
 Z_1(n) &= x(n) + x(n + N/2) \\
 Z_1\left(n + \frac{N}{2}\right) &= x(n) - x\left(n + \frac{N}{2}\right) \quad 0 \leq n \leq \frac{N}{2}
 \end{aligned}$$

The butterfly outputs Z<sub>1</sub>(n) and Z<sub>1</sub>(n+N/2) are computed according to the equations given in (5). Z<sub>1</sub>(n) is sent to apply the twiddle factors, and Z<sub>1</sub>(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 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, as shown in Figure 2 (a) and (b), which requires two bit control signal from the synchronizing counter. The data then goes through a full complex multiplier working at 75% utility, and accomplishes the result of first level of radix-4 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.

Table 1: Implementation result

Logic utilization	Used	Available	Utilization (%)
No. of slices	3155	3584	88
No. of slice flip flops	1514	7168	21
No. of 4 input LUTs	5916	7168	82
No. of bonded IOBs	32	97	32
No. of 8x8 Multiplexers	16	16	100
No. of GCLKs	1	8	12

Table 2: Timing summary

Minimum period	10.827 (ns) (maximum frequency: 92.366MHz)
Minimum input arrival time before clock	5.406
Maximum output required time after clock	6.216

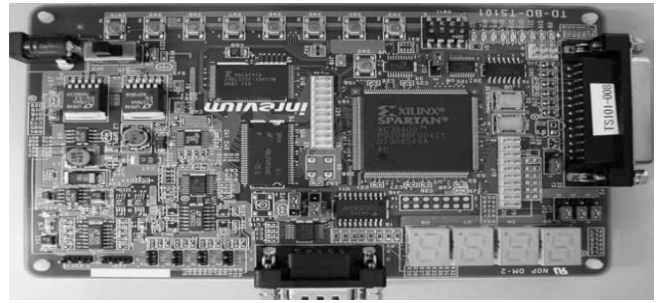


Figure 3  
Xilinx spartan 3 FPGA kit

Table 3  
Hardware requirement comparison

	Multiplier #	Adder #	Memory size
R2MDC <sup>[14]</sup>	$2(\log_4 N - 1)$	$4\log_4 N$	$3N/2 - 2$
R2SDF <sup>[15]</sup>	$2(\log_4 N - 1)$	$4\log_4 N$	$N - 1$
R4SDF <sup>[16]</sup>	$\log_4 N - 1$	$8\log_4 N$	$N - 1$
R4MDC <sup>[14]</sup>	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$
R4SDC <sup>[17]</sup>	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$
Proposed	$\log_4 N - 1$	$4\log_4 N$	$N - 1$

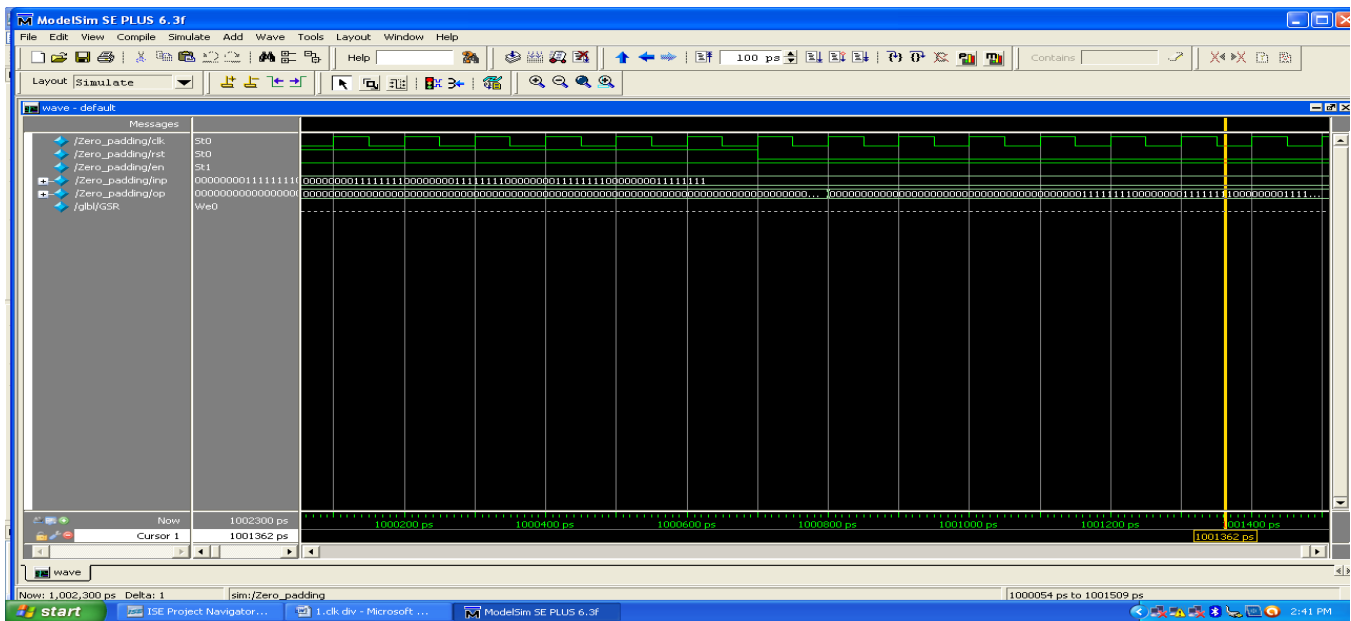


Figure 4  
8- point FFT simulation result

to the right, in bit-reversed order. The next frame of transform can be computed without pausing due to the pipelined processing of each stage.

### 3. IMPLEMENTATION USING VERILOG

The R2<sup>2</sup>SDF presented above has been fully coded in Verilog HDL. Once the design is coded in Verilog, the Modelsim

XEHI 6.2c compiler and the Xilinx Foundation ISA Environment 9.1i 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<sup>2</sup>SDF in Figure 2, the butterfly blocks BF2I and BF2II are described as building blocks in Verilog code. Booth multiplication algorithm for signed binary numbers is used for complex multipliers. Thus, the overall latency of the real implementation varies as the processing word length changes. Look-up-table (LUT) based random access memories (RAMs) and flip-flops are used to implement feedback memory of the very last stages where 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 (F. Wakerly et al. 2006). The FFT is heavily pipelined to achieve as highest clock frequency as possible. Twiddle factors are generated by an external program and embedded to the VHDL code. The implementation results after implementing in Xilinx Spartan3 FPGA (Figure 3) are listed in Table 1 and Table 2. Table 1 shows the implementation results whereas Table 2 shows the timing summary. The resulting figures show that our implementation outperforms the other implementations of that kind its speed nearly matches that of the Xilinx core but its throughput is more than 3 times higher due to its pipeline nature.

## 4. PERFORMANCE AND IMPLEMENTATION

### 4.1. Hardware Requirement

The radix-4 butterfly needs 3 complex adders and 1 complex multiplier, while the proposed butterfly structure needs only 4 complex adders and 1 complex multiplier. This is because of design implements the constant multiplier by 4 reused complex adders. Figure 3 also shows the radix-8 butterfly and radix-4 butterfly. All of the above-mentioned use separated single static random access memory (SRAM) into 2 smaller SHAMS. This design can double SRAM throughput with inter-leaving access. In Table 3, the hardware requirement of the proposed design is compared with various pipelined designs.

The radix-4 butterfly needs 3 complex adders and 1 complex multiplier, while the proposed butterfly structure needs only 4 complex adders and 1 complex multiplier. This is because our design implements the constant multiplier by 4 reused complex adders. Figure 3 also shows the radix-8 butterfly and radix-4 butterfly. All of the above-mentioned use separated single static random access memory (SRAM) into 2 smaller SHAMS. This design can double SRAM throughput with inter-leaving access. In Table 3, the hardware requirement of the proposed design is compared with various pipelined designs.

### 4.2. Power Consumption

The power consumption is measured by the number of times of data transition. The data transition times are proportional to the SRAM access times. Here we assume that the adders and multipliers are active at each clock cycle because of the pipelining architecture. The more the SRAM access times are, the higher the power consumption is the SRAM access times versus N points FFT. The SRAM access times are linear to the number of the recursive iterations in FFT as described in (6). The SRAM accessed twice each clock cycle, so (6) is multiplied by 2. This shows that the proposed design has less access than the radix-4 FFT by 20% to 40%. Therefore, the proposed architecture consumes much lower power. SRAM access times  $\times$  N (iteration times)  $\times$  2 (6).

### 4.3. Implementation

The Fujitsu WiMAX SoC, MB87M3550, fully complies with the IEEE 802.16d standard using an OFDM. The system-on-chip (SoC) can operate in all the available channel bandwidths from 1.75MHz up to 20MHz bandwidths (C. H. Su et al. 2009). This SoC is designed to support frequencies from 2 GHz to 11 GHz in both licensed and license-exempt bands. A programmable frequency selection generates the sample clock for any desired bandwidth. Uplink sub channelization is supported as defined in the standard. The SoC's integrated ARM-926 RISC engine implements 802.16 upper layer MAC, scheduler, drivers, protocol stacks, and user application software. A multi-channel DMA controller handles high-speed transactions among various agents on a high performance bus. To offload processing from the upper layer MAC and enhance performance, the Fujitsu WiMAX SoC includes a separate ARC RISC/DSP engine to execute 802.16 lower layer MAC functions. The chip's multiple hardware-based encryption/decryption engines are tightly coupled with this lower layer MAC processor.

## 5. CONCLUSION

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: 1) fewer butterfly iterations to reduce power consumption, 2) pipeline of radix-2<sup>2</sup> butterfly to speed up clock frequency, 3) even distribution of memory access to make utilization efficiency in SRAM ports. Simulation result of 8 bit FFT code is shown in figure 4.

## REFERENCES

1. T. Lenart and et al., "Architectures for dynamic data scaling in 2/4/8K pipeline FFT cores", *IEEE Trans., on Very Large Scale Integration Systems*, Vol. 14, no. 1, pp. 1286-1290, 2004.
2. M.S. Minallah and G. Raja, "Real Time FFT Processor Implementation", in *Proc. Of the 2<sup>nd</sup> International Conference on Emerging Technologies*, Peshvar, Pakistan, pp. 192-195, 2006.
3. Sankhsawas and K.Benkrid, "A High Level Implementation of a High Performance Pipeline FFT on Virtex-E FPGA", in *Proc. of IEEE Comp. Society Annual Symp. on VLSI Emerging Trends in Systems Design*, Lafayette, LA, USA, pp. 229-232, 2006.
4. Harikrishna, T. R. Rao and V. A. Labay, "A Radix-2<sup>2</sup> Pipeline FFT for Ultra Wide Band Technology", in *Proc. of International Conference on Computer and Network Technology*, Chennai, India, pp. 171-175, 2009.
5. F. Wakerly, *Digital Design Principles and Practices*, 4<sup>th</sup> edition., Upper Saddle River, NJ: Pearson Education., Inc 2006.
6. Y.Zhang and H.H.Chen, "Mobile WIMAX", Boca Raton, FL: Auerbach Publications, Taylor and Francis Group, 2007.
7. G. Andrews, A. Gosh and R. Mohammed, "Fundamentals of WIMAX", Upper Saddle River, NJ: Prentice Hall Publication, 2007.
8. S. Jha and R.Prasad, "OFDM towards Fixed and Mobile Broadband Wireless Access", London, Artech House, 2007.
9. H. Lin and G. Li, "OFDM Based Broadband Wireless Networks", Hoboken, New Jersey, John Wiley and Sons, 2008.
10. C. H. Su and J. M. Wu, "Reconfigurable FFT design for Low Power OFDM Communication Systems", in *Proc. of 2009*.