



FFT/IFFT Processor Design for 5G MIMO OFDM Systems

Gnanishivaram K¹, Dr. S. Neeraja²

¹GITAM Institute of Technology, India, gnanishivaram@gmail.com

²GITAM Institute of Technology, India, neeraja413@gmail.com

ABSTRACT

In this paper, a Fast Fourier Transform (FFT) or inverse FFT processor for Fifth-Generation (5G) Multiple Input Multiple Output (MIMO) Orthogonal Frequency Division Multiplexing (OFDM) system's baseband processor is implemented. The proposed 128-point FFT/IFFT processor employs mixed-radix (radix-2 and radix-2³) algorithm to reduce the number of complex multiplications. The pipelined FFT architecture with Multipath Delay Feedback (MDF) is chosen for FFT/IFFT processor implementation to have very high throughput rate and minimum power consumption. The resulting Mixed-Radix MDF (MRMDF) architecture provides a very high throughput rates for 1-8 simultaneous data sequences to meet new emerging standards of the MIMO-OFDM based systems. The hardware description is developed using Verilog and synthesized using Xilinx Virtex 5 FPGA family aiming to optimize the design in terms of area and speed at low frequency.

Key words : Fifth-Generation (5G), Fast Fourier Transform (FFT), Mixed-Radix Multipath Delay Feedback (MRMDF), Multiple-Input Multiple-Output (MIMO), Orthogonal Frequency Division Multiplexing (OFDM).

1. INTRODUCTION

Fifth-Generation wireless and mobile systems are currently the focus of research and development. In recent years, Multiple-Input Multiple-Output (MIMO) wireless technologies gained a very fast pace into research, given the capacity increase achievable with such schemes [1], [2]. MIMO exploits usage of multiple antennas at both the base station transmitter and the user terminal receiver on downlink. The processing at the terminal by spatial antenna will be able to eliminate the effects of complex multipath scattering, providing fundamentally, access to parallel independent propagation paths between the base station and the user. Thus, instead of having access through a single data path, as with conventional wireless system design, a wireless system using MIMO technology enhances the system data rate and capacity by the presence of multiple parallel lines. MIMO has now evolved into certain path, and is being investigated in the Fifth Generation Projects for the evolution of the Fourth-Generation (4G LTE) systems.

Another technology that has been considered by the industry for 4G systems evolution is Orthogonal Frequency-Division Multiplexing (OFDM). The Wireless World Research Forum (WWRF) considers OFDM the most important technology for a future public cellular radio access technology [3]. OFDM is a widely adopted modulation technique for high-speed wireless communications. The OFDM-based physical layer has been chosen for several standards, such as the IEEE 802.11n Wireless Local Area Network (WLAN) in both 2.4-5GHz band, the IEEE 802.11g WLAN in the 2.4GHz band, and the European Digital Video Broadcasting system (DVB-T). It is also under consideration as the high rate alternate physical layer to the draft IEEE P802.15.3 Wireless Personal Area Network (WPAN), the IEEE 802.20 Mobile Broadband Wireless Access (MBWA), and the IEEE 802.16 [4] Wireless Metropolitan Area Networks (WMAN). OFDM is very robust to typical multipath fading (i.e., frequency-selective) channels as data is multiplexed on many narrowband subcarriers. The subcarriers can easily be generated at the transmitter and recovered at the receiver, using highly efficient digital signal processing based on fast Fourier transform (FFT) as it equalizes with the process of DFT.

Since MIMO is also considered an important building block of future wireless systems, it is important to note that OFDM is particularly well suited to MIMO technologies. Because the narrowband subcarriers in the OFDM signal experience flat fading, MIMO reception does not require complex channel equalization schemes. Enhancement of the data rates for the next generation wireless communication systems operating in frequency-selective fading environments can only be possible by combination of MIMO signal processing with OFDM communication system. As MIMO systems and OFDM, provide the infrastructure for high spectral efficiency and consequently high data rate in the physical layer they are best chosen for wireless networks.

Achieving the standards is a challenge to realize the physical layer of the advanced MIMO OFDM system with minimal hardware complexity and power consumption—especially the computational complexity—in VLSI implementation. The highest computational complexity modules in the physical layer of the MIMO-OFDM systems are FFT/IFFT processor. Hence, employing the traditional approach to solve the

simultaneous multiple data sequences, several FFT/IFFT processors are needed in the physical layer of a MIMO OFDM system. Thus, the hardware complexity of the physical layer in advanced MIMO OFDM system will be very high. This paper proposes an FFT/IFFT processor with a unique multipath pipelined architecture to deal with the issue of the multiple data sequences for advanced MIMO OFDM applications. The 128-point FFT/IFFT with 1–8 simultaneous data sequences can be supported in this proposed processor with minimal hardware complexity. One to eight sequences are considered as part of up gradation concerned with present trending systems and standards by IEEE. Higher radix FFT algorithm implementation saves power consumption compared with lower radix.

The paper is organized as follows. Section 2 describes the problems of the implementation of the FFT/IFFT processor in a MIMO OFDM system. The 128-point mixed-radix FFT algorithm including radix-2 FFT algorithm and three-step radix-8 FFT algorithm and the IFFT algorithm is described in Section 3. Section 4 focuses on describing the proposed FFT/IFFT architecture for advanced MIMO-OFDM applications. In Section 5, the simulation results of the proposed FFT/IFFT processor are presented. Section 6 details conclusions at the end of paper.

2. DESIGN ISSUE OFFFT/IFFT PROCESSOR FOR ADVANCEDMIMO OFDM SYSYTEM

As a test case, the OFDM-based wirelesslocal area network (WLAN) standard IEEE 802.11ac is considered, but the results are applicable more generally. A block diagram of the receiver of IEEE 802.11ac standard is shown in Figure 1[5]. It contains eight RFs, eight Guard Interval removers (GIs), eight FFTs, a MIMO decoder, eight de-mapper and de-interleaver, a spatial-stream de-parser, an encoder de-parser, a descrambler, a synchronization block, and a channel estimation block. Depending on the desired data rate in compliance with 802.11T Gac, the modulation scheme can be

Binary Phase Shift Keying (BPSK), Quaternary Phase Shift Keying (QPSK), or Quadrature Amplitude Modulation (QAM) with 1–6 bits. The FFT/IFFT processor has to calculate simultaneous 1–8 data sequences depending on the number of spatial sequence, the guard interval and the FFT size.

Till date, many FFT architectures are introduced, majority of them are single-memory architecture, dual-memory architecture [6], pipelined architecture [7], array architecture [8], and cached-memory architecture [9], have been proposed. Traditionally, multiple FFT processors are included to deal with multiple data sequences in a MIMO OFDM system, as shown in Figure 1 to deal with multiple data sequences. This causes a large increase in the hardware complexity with power consumption, which is not desirable in low power mobile device OFDM application. Single FFT processor deployment is major concern instead of multiple FFT processors to cater the needs of high speed mobile devices. A good FFT processor shouldnot only provide a high throughput rate, but also deal with multiple data sequences effectively for MIMO OFDM applications. In this paper, the pipelined architecture is analyzed as the best choice for the very high throughput rate applications since it can provide high throughput rate with acceptable hardware cost. The innovative approach for present day MIMO OFDM standards with very high throughput rate, least power consumption, less hardware is Multipath Delay Feedback (MDF) architecture. MDF is the combination of MDC and SDF features [10]. The operation of the complex multiplication consumes highest of power in the FFT processor. In order to save power dissipation, higher radix FFT algorithm can be used to reduce the number of complex multiplications [11]. Three-step radix-8 FFT algorithm is chosen in this design to save complex multiplications. Because 128-point FFT is not a power of 8, the mixed-radix FFT algorithm combining two different FFT algorithms is needed. In this proposed processor, mixed-radix multipath delay feedback (MRMDF) FFT architecture [13] can not only deal with 1–8 simultaneous data sequences for MIMO OFDM applications but also save lots of hardware complexity, compared with the traditional approach as shown in Figure1.

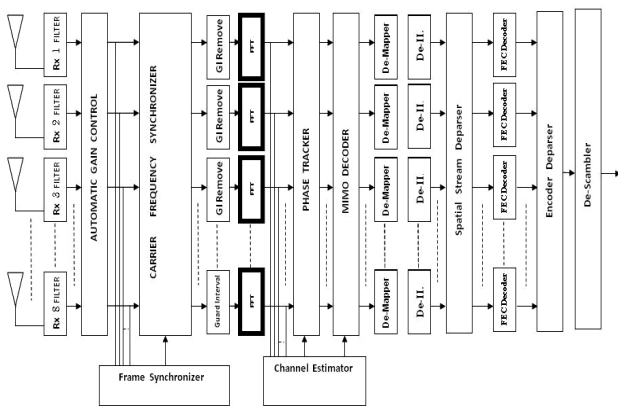


Figure. 1:Block diagram of the receiver of IEEE 802.11ac standard.

3. ALGORITHM

Given a sequence, $x(n)$ an N -point discrete Fourier transform(DFT) is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, \dots, 127. \quad (1.1)$$

Where $x(n)$ and $X(k)$ are complex numbers. The twiddle factoris

$$W_N^{kn} = e^{-j\left(\frac{2\pi kn}{N}\right)} = \cos\left(\frac{2\pi kn}{N}\right) - j\sin\left(\frac{2\pi kn}{N}\right) \quad (1.2)$$

In (1.1), the computational complexity is $O(N^2)$ through directly performing the required computation. By using the FFT algorithm, the computational complexity can be reduced to $O(N \log_r N)$, where r means the radix- r FFT. The radix- r FFT algorithm can be easily derived from DFT by decomposing the N -point DFT into a set of recursively related r -point FFT transform, if N is a power of r . Higher radix FFT algorithm has less number of the nontrivial complex multiplications, compared with the radix-2 FFT algorithm which is the simplest form in all FFT algorithms [8]. In an example for 128-point FFT, the number of nontrivial complex multiplications of radix-8 FFT algorithm is 152, which is only 58.9% of that of radix-2 FFT algorithm [8]. Thus, in order to save power dissipation of the complex multiplier operation, radix-8 FFT algorithm chosen to reduce the number of nontrivial complex multiplications. The mixed-radix FFT algorithm, including radix-2 and radix-8 FFT algorithm, is needed to effectively implement 128-point FFT. The algorithm has been described briefly here.

Consider

$$\left. \begin{aligned} N=128, n_1=0,1 \\ n=64n_1+n_2, n_2=0,1, \dots, 63 \\ k_1=0,1 \\ k=k_1+2k_2, \end{aligned} \right\} \left. \begin{aligned} k_2=0,1, \dots, 63 \end{aligned} \right\} \quad (1.3)$$

Using (1.3), (1.1) can be rewritten as

$$\begin{aligned} X(2k_2+k_1) &= \sum_{n_2=0}^{63} \sum_{n_1=0}^1 \chi(64n_1+n_2) W_{128}^{(64n_1+n_2)(2k_2+k_1)} \\ &= \sum_{n_2=0}^{63} \left\{ \sum_{n_1=0}^1 \chi(64n_1+n_2) W_2^{n_1 k_1} W_{128}^{n_2 k_1} \right\} W_{64}^{n_2 k_2} \\ &= \sum_{n_2=0}^{63} BU_2(k_1, n_2) W_{64}^{n_2 k_2} \end{aligned} \quad (1.4)$$

Equation (1.4) can be considered as a two-dimensional DFT. One is 64-point DFT and the other is 2-point DFT. Then, by decomposing the 64-point DFT into the 8-point DFT recursively 2 times, the 128-point mixed-radix FFT algorithm can be completed. In order to implement a radix-8 FFT algorithm more efficiently, using the radix-2³

FFT algorithm [7], it is further decomposed to the butterfly of radix-8 FFT algorithm into three steps and apply the radix-2 index map to the radix-8 butterfly. The three-step radix-8 FFT explained and derived in detail in [10]. Figure 2 shows the signal flow graph (SFG) of the 128-point mixed-radix FFT algorithm and two 64-point three-step radix-8 FFT algorithms. In Figure 2, there are three stages in 128-point FFT algorithm. The radix-2 FFT algorithm is used in the first stage, and the radix-8 algorithm is applied in the second and third stages. And the butterfly of radix-8 in stage 2 and stage 3 can be further decomposed into three steps by using the radix-2³ FFT algorithm. The twiddle factors, $W_8^1 = e^{j(2\pi \times 1)/8}$ and $W_8^3 = e^{j(2\pi \times 3)/8}$, at the third step in each radix-8 butterfly are trivial complex multiplications because they can be written as $\sqrt{2}/2(1-j)$ and $-\sqrt{2}/2(1+j)$, respectively. Thus, a complex multiplication with one of the two coefficients can be computed using additions and a real multiplication, whose hardware can be realized by six shifters and four adders. The black point shown in Figure 2 means that the specific twiddle factor will be multiplied at that point. Stage 2 and stage 3 can be considered as two 64-point radix-8 FFT algorithms.

The IFFT of an N -point sequence $X(k), k=0, \dots, N-1$ is defined as

$$x(n) = \left(\frac{1}{N}\right) \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad k = 0, 1, \dots, 127. \quad (1.5)$$

In order to implement the IFFT algorithm more efficiently, (1.5) can be written as [10],

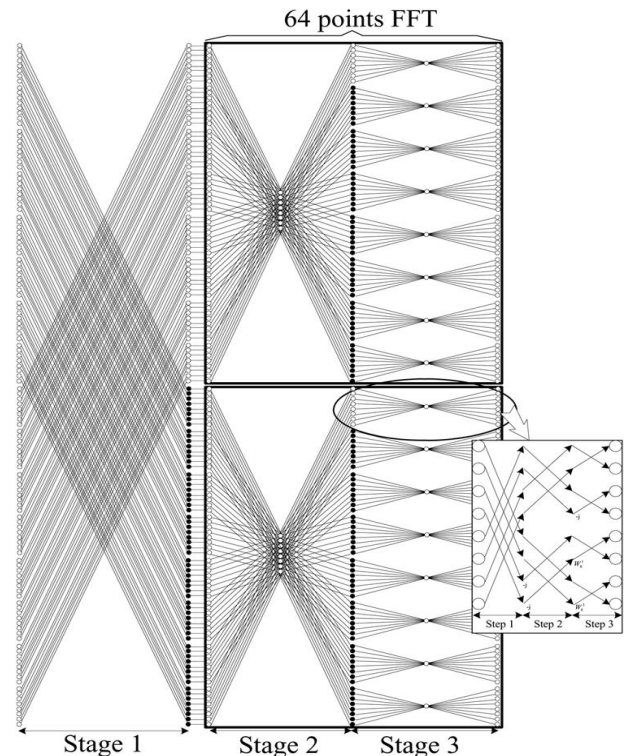


Figure 2: SFG of 128-point mixed-radix FFT

$$x(n) = \left(\frac{1}{N}\right) \left\{ \sum_{k=0}^{N-1} X^*(k) W_N^{kn} \right\}^* \quad (1.6)$$

According to (1.6), the IFFT can be performed by taking the complex conjugate of the incoming data first and then the outgoing data without changing any coefficient in the original FFT algorithm so that the hardware implementation can be more efficient.

4. ARCHITECTURE

Proposed FFT Architecture for Advanced MIMO OFDM System

According to (1.4), (1.6), and SFG (Figure 2), the proposed novel 128-point FFT/IFFT processor is proposed to support 1–8 simultaneous data sequences for a MIMO OFDM system, as shown in Fig. 3.

The proposed FFT architecture consists of Module 1, Module 2, Module 3, conjugate blocks, a division block, and multiplexers. The features of the proposed FFT architecture are as follows. First, 128-point FFT with 1–8 simultaneous data sequences can be operated in this design. Second, the proposed FFT architecture can provide very high throughput rate to achieve the requirements advanced WLAN standard. Third, the minimal memory is needed by using the delay feedback scheme to reorder the input data and the intermediate results of each module; the hardware complexity of the complex multipliers can be reduced by using the scheduling approach and the specified constant multipliers; so the proposed FFT processor has less hardware complexity, compared with the approach using multi-FFT processors. And at last, higher radix FFT algorithm can be implemented to save power dissipation.

In the MRMDF architecture, the input sequence and the output sequence are in the specified order. The order of the output sequence is the bit reversal of the order of the input sequence, as seen in Figure 3. The operation of the FFT or IFFT is controlled by the control signal, FFT/IFFT, as shown in Figure 3. When an IFFT is performed in this processor, the sign of the imaginary part of the input sequences will be changed to conjugate input data and then they will be performed by the process in treating FFT. The sign of the imaginary part of output data from FFT will be changed again and then will be divided by 128 as in Figure 3. As 128 is power of two, the operation of the division is implemented by shifting the decimal point location to ease hardware implementation. The function of Module 1 is to implement a radix-2 FFT algorithm, corresponding to the first stage of the SFG, as shown in Figure 2. Modules 2 and 3 are for the radix-8 FFT algorithm, corresponding to the second and third stages of the SFG, as displayed in Figure 2.

To ensure the correction of the FFT output data and for the minimization of memory requirement, two different structures are built in Modules 2 and 3 to implement the radix-8 FFT algorithm. In addition, the complex multiplier’s hardware complexity will be also considered in the proposed architecture. In the next few subsections, each Module will be described in more detail.

4.1 Module 1:

Module-1 comprises of a 64-complex data storage-register file, one butterfly unit (BU), multiplexers, two ROMs, and four complex multipliers, as shown in Figure 4. The function of module-1 is analogous to stage-1 of SFG in Figure 2. BU operation does not start until both the input sequences $x(n)$ and $x(64+n)$ are available as it employs radix-2 FFT for 128

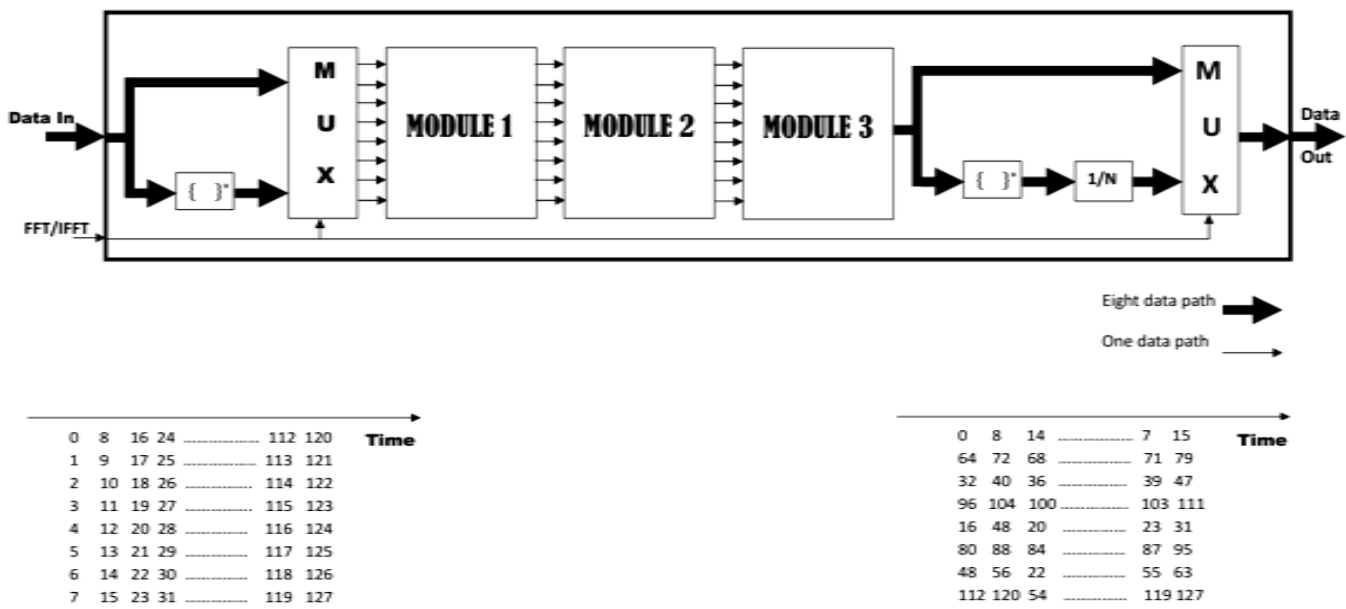


Figure. 3: Block diagram of the proposed 128-point FFT/IFFT processor.

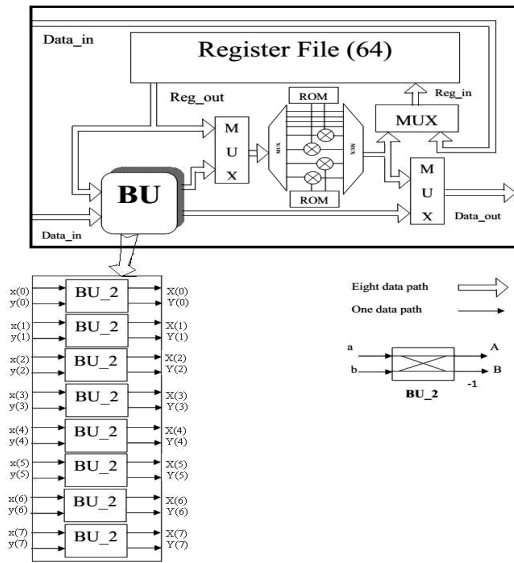


Figure. 4: Block diagram of the Module 1

point. The input is 8 parallel multiple sequences hence, the order implemented is $in(8m)$, $in(8m+1)$, $in(8m+3)$, $in(8m+4)$, $in(8m+5)$, $in(8m+6)$, $in(8m+7)$ respectively, where m is from 0, 1, ..., 15. The BU consists of eight BU_2s, which operate the complex addition and complex subtraction from two input data as in Figure 4. The two sets of 8-input data are available to BU after 8-cycles from the input initiated into module-1. At first 8-cycles, the first 64-data are stored in the register file. In the next 8-cycles, the input data $x(i)$ and $y(i)$ of the BU are received from the register file and input, respectively as seen in Figure 4. The BU generates the outputs data $X(i)$ and $Y(i)$. The $Y(i)$ need to be multiplied by the twiddle factors to complete DIF radix-2 FFT operation. Meanwhile, eight output data, $X(i)$, generated by the BU, are fed to Module 2 directly, and the other eight output data $Y(i)$ are stored into the register file. Instead of using 8-complex multipliers for multiplying twiddle factors and $Y(i)$ a modified approach using 4-complex multipliers is used. In order to reduce complex multipliers, the $Y(i)$ s generated by the BU $Y(0)$, $Y(1)$, $Y(2)$, and $Y(3)$ are multiplied by twiddle factors before the $Y(i)$ s are stored in the register file. After 16 clock cycles other 4-output data $Y(i)$ s, $Y(4)$, $Y(5)$, $Y(6)$, and $Y(7)$ are multiplied before $Y(i)$ s are fed to module-2. By this time-rescheduling of complex multiplications, only 4-complex multipliers are needed instead of 8-complex multipliers. Hence utilization will be 100% for complex multipliers.

The twiddle factors are stored ROM. Only 1/8 period of cosine and sine waveforms are stored in ROM and rest of them can be reconstructed by the stored values which will be detailed in next subsection.

4.2 Module 2: Module 2 consists of eight BU_8 structures and one modified complex multiplier, as shown in Figure 5.

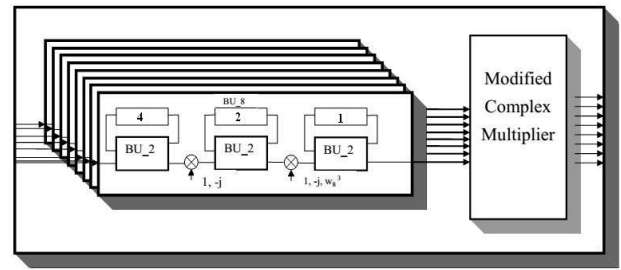


Figure. 5: Block diagram of the Module 2.

These eight BU_8s operate in the same way. The module-2 architecture employs three-step radix-8 FFT algorithm, whose SFG is shown in Figure 2. Each of the BU_8 unit maps to SDF architecture with delay elements four, two and one, respectively, as shown in Figure 5.

When the 8-data from module-1 initially input to module-2, each of them drive into separate BU_8. BU_2 units inside BU_8 unit compute radix-8 FFT by three-step radix algorithm. The delay elements store the incoming data till next remaining data is input to employ BU_2 operation as in module-1. Meanwhile, the twiddle factors $1, -j, W_8^1$ or W_8^3 are multiplied in intermediate stages as shown in Figure 5. The multiplications in all the modules as in here are efficiently carried out using shifters and adders [12] for area and logical unit reduction instead of multiplier units.

The modified complex multiplier in Figure 5 operates multiplication analogous to final stage of multiplication process in second stage of SFG in Figure 2. Multiplying all 8-output data from BU_8 units with twiddle factors simultaneously in one-clock cycle would enhance data rate. Hence a modified approach is followed as in [12], simplifying the complex multiplier. Considering the symmetric property of twiddle factors, only nine sets of constant values (X_p, Y_p) , out of 32 twiddle factors are sufficient for modified complex multiplier. The rest of them can be generated using the mapping Table 1. 8-sets out of nine sets of constant values suffice the requirement as the first set $(1, 0)$ needs trivial multiplication. Table 2 shows the scheduling of the twiddle factor in each data path. In some of them the requirement of constant sets is more than once.

TABLE 1: Mapping table used to determine the values of the twiddle factors

Twiddle factor	Real	Imaginary
$W_{64}^0 - W_{64}^8$	X_p	Y_p
$W_{64}^9 - W_{64}^{16}$	$-Y_p$	$-X_p$
$W_{64}^{17} - W_{64}^{24}$	Y_p	$-X_p$
$W_{64}^{25} - W_{64}^{32}$	$-X_p$	Y_p
$W_{64}^{33} - W_{64}^{40}$	$-X_p$	$-Y_p$
$W_{64}^{41} - W_{64}^{48}$	Y_p	X_p
$W_{64}^{49} - W_{64}^{56}$	$-Y_p$	X_p
$W_{64}^{57} - W_{64}^{64}$	X_p	$-Y_p$

TABLE 2: Scheduling of the twiddle factor, W_{64}^p , where, p is from 1 to 8

		→							
Time slot		0	1	2	3	4	5	6	7
Data path									
1		0	0	0	0	0	0	0	0
2		0	4	8	4	0	4	8	4
3		0	2	4	6	8	6	4	2
4		0	6	4	2	8	2	4	6
5		0	1	2	3	4	5	6	7
6		0	5	6	1	4	7	2	3
7		0	3	6	7	4	1	2	5
8		0	7	2	5	4	3	6	1

This is overcome by having additional constant sets to reduce latency. The block diagram of modified complex multiplier is shown in Figure 7.

4.3 Module 3:

The radix-8 FFT algorithm is employed in module-3 as in third stage of SFG. Though radix-8 FFT is implemented in both module-2 and module-3 the architecture remain different because the two available data for the BU_2 in second and third stages of SFG are in different data paths. The block diagram of module 3 is as shown in Figure 8.

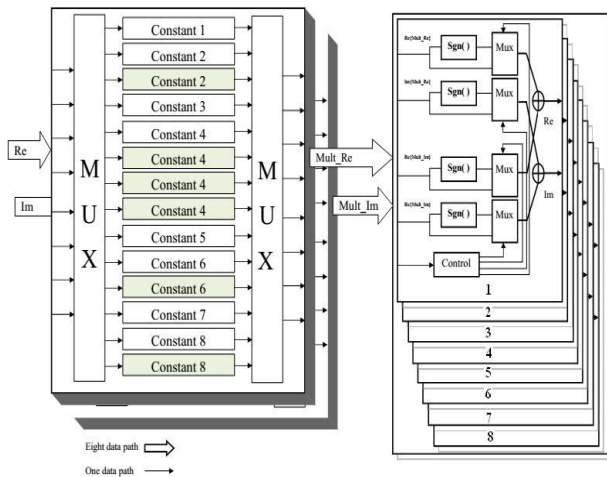


Figure. 7: Block diagram of the modified complex multiplier.

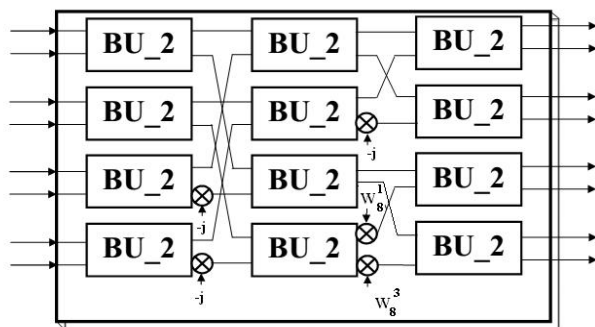


Figure. 8: Block diagram of the Module 3

This architecture is well suited with correction of the FFT output data. The twiddle factor multiplications are performed in intermediate stages as in Figure 8. At this final stage the output is out if it had been operated in FFT mode otherwise the imaginary portion multiplied by -1 furthermore divided by 128 to output in IFFT mode.

5. DESIGN IMPLEMENTATION

The proposed design has been coded in Verilog HDL. It is synthesized and simulated on Virtex 5 XC5VLX110 based device using Xilinx Foundation ISA Environment 10.1 and ModelSim-XE. The XST synthesis tool has been used to synthesize the proposed model whose design summary containing allocated resources is shown in Table 3. Figure 9 shows simulated waveforms for the FFT implementation. The IFFT mode simulation is much similar to that of FFT. The outputs of the implementation are verified in comparison with MATLAB simulated for the same input sequences.

TABLE 3:Design utilisation summary on XC5VLX110FF1153-3

Logic Utilization	Used	Available	Utilization
Number of slice Registers	4721	207360	2%
Number of Slice LUTs	4533	207360	2%
Number of fully used LUT-FF pairs	3660	5594	65%
Number of bounded IOBs	260	1200	21%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48Es	52	192	27%

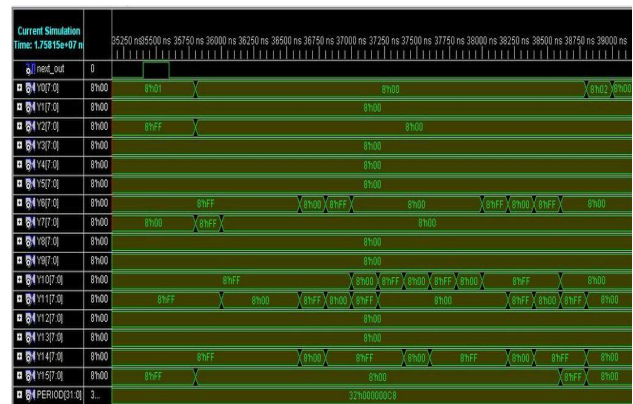


Figure. 9: Simulated waveforms for the FFT mode.

6. CONCLUSION

This paper presented a very efficient FFT architecture for advanced MIMO OFDM WLAN standards. An area and speed efficient technique using FPGA has been analyzed

and implemented to cope with raising needs of new standards. The proposed architecture has following advantages:

- Small number of butterfly iterations to reduce power consumption.
- Pipelined architecture of mixed radix- $2/2^3$ FFT to speed up operations performed per clock frequency.
- Exploitation of symmetry of twiddle factors for reduced memory usage.

In summary, the speed performance of this design easily satisfies most application requirements of MIMO OFDM. One of them is 802.11 based advanced MIMO OFDM WLANs.

The proposed designs can be implemented scalable to any arbitrary FFT size N (especially 64/256/512) which may be compliant with 802.11T Gac. The proposed method can still be improved with low power design methodologies [9], like algorithm development for low power multiplier components, butterfly structures, ROMs in accordance with different OFDM application.

ACKNOWLEDGEMENT

The authors would like to thank Department of Electronics and Communications, GITAM Institute of Technology, Visakhapatnam for the support.

REFERENCES

1. E. Teletar, **Capacity of Multi-Antenna Gaussian Channels**, *Bell Labs tech. rep.*, June 1995.J.
2. G. J. Foschini, **Layered Space-Time Architecture for Wireless Communications in a Fading Environment when Using Multiple Antennas**, *Bell Labs Tech. J.*, vol. 1, no. 2, Fall 1996.
3. WWRF, *The Book of Visions 2001, Visions of the Wireless World — An Invitation to Participate in the Making of the Future of Wireless Communications*, 2001.
4. Local and Metropolitan Area Networks—Part 16, **Air Interface for Fixed Broadband Wireless Access Systems**, IEEE Standard IEEE 802.16a.
5. Daisuke Nojima, Leonardo Lanante Jr., Yuhei Nagao, Masayuki Kurosaki and Hiroshi Ochi, **Performance Evaluation for Multi-User MIMO IEEE 802.11ac Wireless LAN System**, Feb.19~22, 2012 ICACT 2012.
6. S. Magar, S. Shen, G. Luikuo, M. Fleming, and R. Aguilar, **An application specific DSP chip set for 100-MHz data rates**, in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, Apr. 1988, vol. 4, pp. 1989–1992.
7. H. Shousheng and M. Torkelson, **Designing pipeline FFT processor for OFDM (de)modulation**, in *Proc. URSI Int. Symp. on Signals, Syst, Electron.*, Oct. 1998, vol. 29, pp. 257–262.
8. J. O'Brien, J. Mather, and B. Holland, **A 200 MIPS single-chip 1k FFT processor**, in *Proc. IEEE Int. Solid-State Circuits Conf.*, 1989, vol. 36, pp. 166–167., 327.
9. B. M. Bass, **A low-power, high-performance, 1024-point FFT processor**, *IEEE J. Solid-State Circuits*, vol. 34, no. 3, pp. 380–387, Mar. 1999.
10. Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, **A 1 GS/s FFT/IFFT processor for UWB applications**, *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1726–1735, Aug. 2005.
11. W.-C. Yeh and C.-W. Jen, **High-speed and low-power split-radix FFT**, *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 51, no. 3, pp. 864–874, Mar. 2003.
12. K. Maharatna, E. Grass, and U. Jagdhold, **A 64-point fourier transform chip for high-speed wireless lan application using OFDM**, *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 484–493, Mar. 2004.
13. Yu-weblin and Chen-Y; Lee, **Design of an FFT/IFFT processor for MIMO-OFDM system**, *IEEE Transactions on circuits and systems*, vol.54, no.4, April 2007.