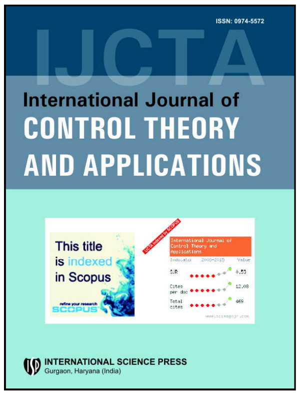


International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 6 • 2017

FPGA Implementation of Efficient FFT Architecture for OFDM Applications

S. Manoj Prabhakar[1], K. Kalyani[2] and S. Rajaram[3]

[1] PG Student, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India, Email: manojprabha24@gmail.com
[2] Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India, Email: k_kalyani@tce.edu
[3] Department of Electronics and Communication Engineering Thiagarajar College of Engineering, Madurai, Tamil Nadu, India, Email: rajaram_siva@tce.edu

Abstract: Fast Fourier Transform plays a vital role in a modern digital signal processing and telecommunications, especially in orthogonal frequency division multiplexing systems. Orthogonal frequency division multiplexing (OFDM) is a multi-carrier modulation technique used in MIMO. In general, FFT and IFFT are used as a digital transmitter and receiver in the OFDM systems. The computational complexity in FFT unit gets increased as the number of sample point increases. When these sample points are processed as a single stream in FFT, further it increases the area and reduces the speed of the FFT process. In this paper the efficiency of FFT unit is improved by redesigning the architecture in such a way that the given data stream is split into two data streams and processed simultaneously to reduce the area. This makes the architecture suitable for high-speed real time applications. The proposed architecture is simulated and synthesized using Verilog HDL in Xilinx ISE 14.1 and it is implemented in Virtex -5 XC5VLX110T-1 FF1136 FPGA board.

Keywords: Fast Fourier Transform, OFDM, MIMO, FPGA

1. INTRODUCTION

Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing (MIMO-OFDM) is the prevailing air interface for 4G and 5G broadband wireless communications [1]. This technology is widely adopted in most of the communication standards, such as IEEE 802.11n WLAN, IEEE 802.6a WIMAX, LTE and LTE advanced. It combines multiple input, multiple output (MIMO) technology, which raises the capacity by transmitting different signals over multiple antennas, and orthogonal frequency division multiplexing (OFDM), which divides a radio channel into a large number of closely spaced sub channels to provide more reliable communications at high speed. Orthogonal frequency division multiplexing [2] [3] is a multi-carrier digital transmission scheme that combines modulation and multiplexing. Multiple sub-carriers at different frequencies are used to carry data. The sub-carrier frequencies are chosen in such a way that they are orthogonal to each other. Figure 1 show the MIMO-OFDM transmission technique. Efficient implementation of MIMO-OFDM system is based on the Fast Fourier Transform [4], which plays a vital role in the MIMO-OFDM transceiver. Fast Fourier Transform (FFT) is the algorithm used to compute Discrete Fourier Transform (DFT) effectively.

There are several FFT architectures proposed in [5]-[7] for OFDM applications .In [5], reconfigurable FFT architecture was proposed for OFDM standards. It reduces the power consumption but hardware is underutilized in this architecture. The FFT processor proposed in [6] also employed to save the power consumption. It operates under different configurations to support the power-aware feature. In [7] FFT processor proposed for MIMO-OFDM system to reduce the hardware resources. But, execution speed of the architecture is reduced. The hardware complexity of FFT unit usually increases due to the bit reversal circuit used in the architecture to bit reverse the output samples. Several bit reversal architectures was proposed to reduce the hardware resources. In [8] optimal bit reversal circuit is proposed by using minimum shift registers. Similarly bit reversal circuit for pipelined architecture is proposed in [9]. The FFT architecture proposed in [10] eliminates the usage of bit reversal circuit by using a novel method.

The proposed architecture computes the N point Fast Fourier Transform as two N/2 point Fast Fourier Transform. It reduces the hardware resources utilized in this architecture and increases the computational time. Thus makes the architecture efficient for high speed real time applications.

2. PROPOSED ALGORITHM

The proposed architecture computes the N point FFT by computing two N/2 FFT, where N is the number of input samples. The methodology of the proposed architecture is given below

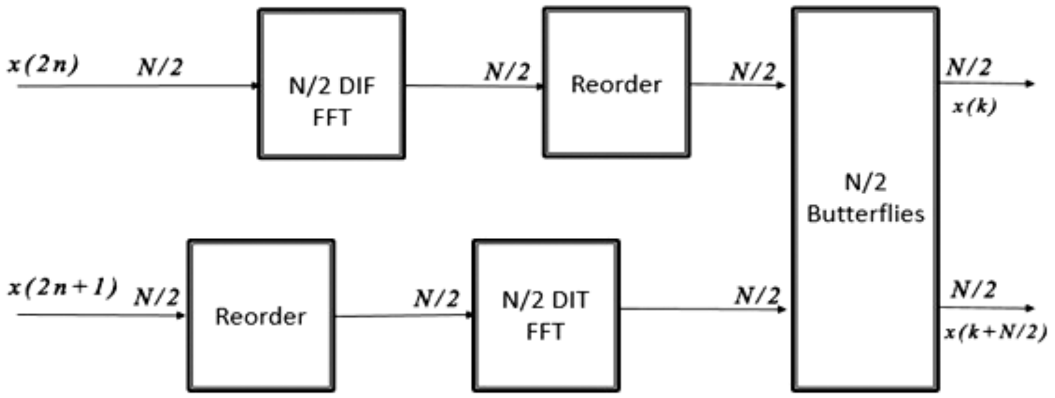


Figure 1: Proposed Methodology

2.1. Proposed FFT architecture

The proposed architecture separates the N input samples into two N/2 input samples which consists of N/2 odd samples and N/2 even samples. DIF (Decimation in Frequency) FFT operation is performed on the N/2 odd samples of the input and DIT (Decimation in Time) FFT operation is performed on the N/2.Finally the output of the two stages are combined together by N butterfly operations to get the N point FFT output. The figure 2 shows the block schematic of proposed FFT architecture. It has two eight-point FFT architecture to process two data streams. The outputs of these two eight-point FFT are processed by 8 butterfly units.

2.2. DIT FFT block

DIT FFT block does the DIT FFT operation on the even samples {x(0),x(2),x(4)……..x(14)} of the sixteen-point FFT inputs. The proposed 16 point FFT architecture has to compute the 8 point DIT-FFT, therefore the 8

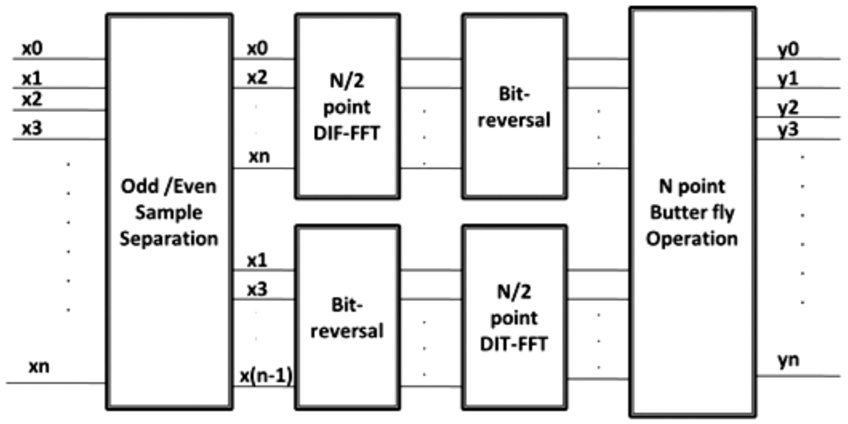


Figure 2: N-point FFT Block Diagram

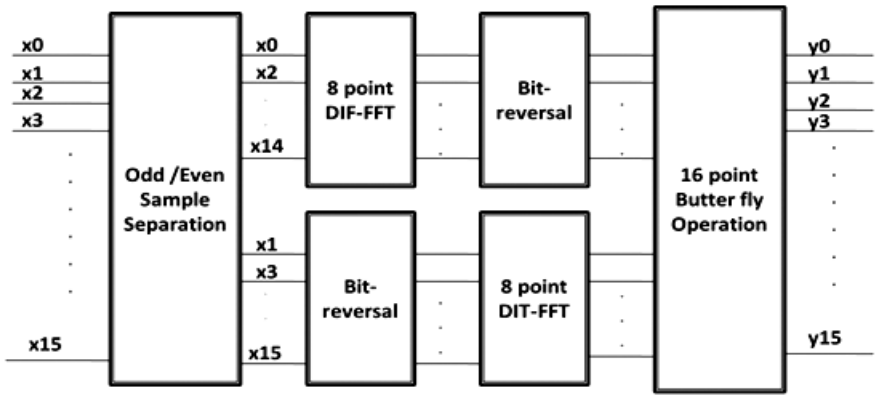


Figure 3: 16-point FFT Block Diagram

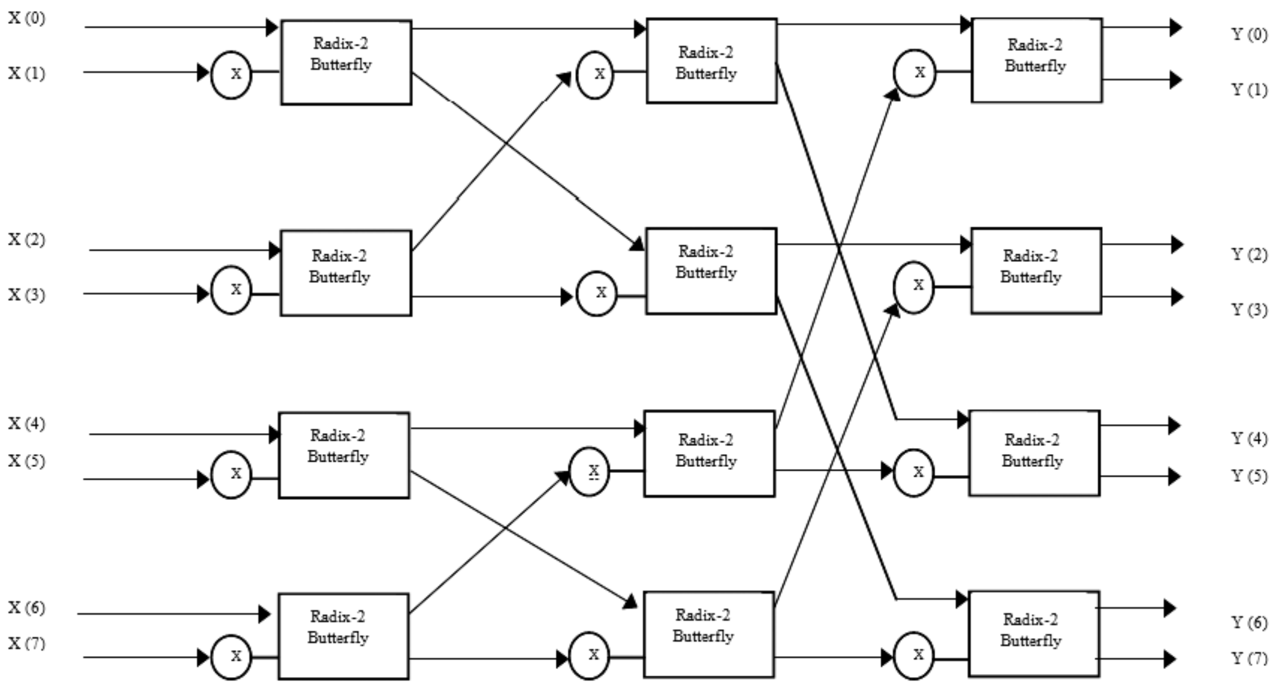


Figure 4: 8-point DIT FFT Block

point DIT-FFT block has 12 radix-2 butterfly units. The odd samples from the input is given to the DIT-FFT block, the samples are given as a pair to the Radix-2 butterfly unit. Basic butterfly operation is performed here. Since this is a DIT-FFT the twiddle factors are multiplied prior to the butterfly operation. Since this is an 8 point DIT-FFT block, there are totally 3 stages required to perform this operation (for N-point FFT, number of stages needed is m i.e., $N=2^m$). The obtained output samples at each stage will be passed to the very next stage and butterfly operation is performed in all these stages. Each stage will have 4 butterfly units. Finally the outputs of these 8 samples are obtained at stage 3.

$\otimes$ - Corresponds to twiddle factor multiplication

The input values are x(0) =x(0); x(1)=x(2); x(2)=x(4); x(3)=x(6); x(4)=x(8); x(5)=x(10); x(6)=x(12); x(7)=x(14) and the twiddle factor values are $w^8_0 = 1$ (stage 1) ; $w^8_0 = 1$, $w^8_2 = -1$ (stage 2); $w^8_0 = 1$,$w^8_1 = 0.707 – 0.707j$,$w^8_2 = -1$,$w^8_3 = -0.707 – 0.707j$ (stage3)

Radix 2 Butterfly unit

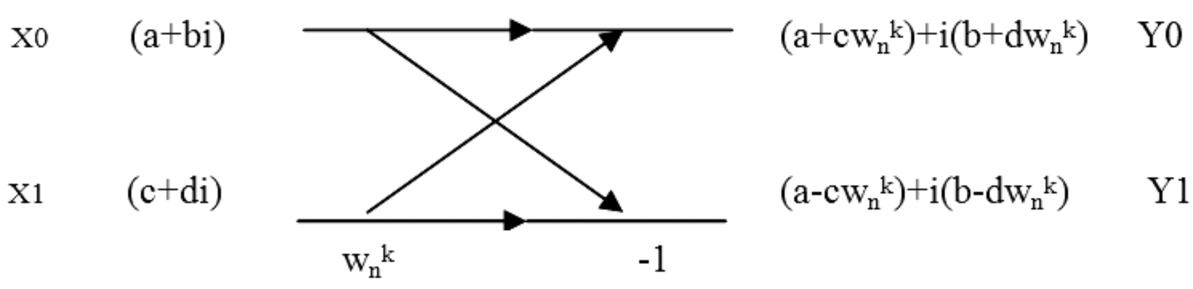


Figure 5: DIT Butterfly Unit

Figure 4 shows the basic butterfly unit for DIT block.a+bi and c+di are the two inputs available at the input of Radix-2 Butterfly unit, where a,b are the real part of the input sample and b,d are the imaginary part of the input sample. The input c+di is multiplied with the twiddle factor w_n^K.The butterfly unit will have two outputs namely O1 and O2.

$$O1=(a+cw_n^k)+i(b+dw_n^k)$$

$$O2=(a-cw_n^k)+i(b-dw_n^k)$$

Where, $a+cw_n^k$ is the real part of O1 and $b+dw_n^k$ is the imaginary part of the O1

$a-cw_n^k$ is the real part of O2 and $b-dw_n^k$ is the imaginary part of O2

2.3. DIF FFT block

DIF FFT operation is performed on the odd samples { x(1),x(3),x(5)……..x(15)} of the sixteen-point FFT inputs . The proposed 16 point FFT architecture has to compute the 8 point DIF-FFT, therefore the 8 point DIF-FFT block has 12 radix-2 butterfly units. The odd samples from the input is given to the DIF-FFT block, the samples are given as a pair to the Radix-2 butterfly unit. Basic butterfly operation is performed here. Since this is a DIF-FFT the twiddle factors are multiplied after the butterfly operation. In this is an 8 point DIF-FFT block, there are totally 3 stages required to perform this operation (for N-point FFT, number of stages needed is m i.e., $N=2^m$). The obtained output samples at each stage will be passed to the very next stage and butterfly operation is performed in all these stages. Each stage will have 4 butterfly units. Finally the output of these 8 samples is obtained at stage 3.

The input values are x(0) =x(1); x(1)=x(3); x(2)=x(5); x(3)=x(7); x(4)=x(9); x(5)=x(11); x(6)=x(13); x(7)=x(15) and the twiddle factor values are $w^8_0 = 1$,$w^8_1 = 0.707 – 0.707j$,$w^8_2 = -1$,$w^8_3 = -0.707 – 0.707j$ (stage1); $w^8_0 = 1$, $w^8_2 = -1$ (stage 2);$w^8_0 = 1$ (stage 3)

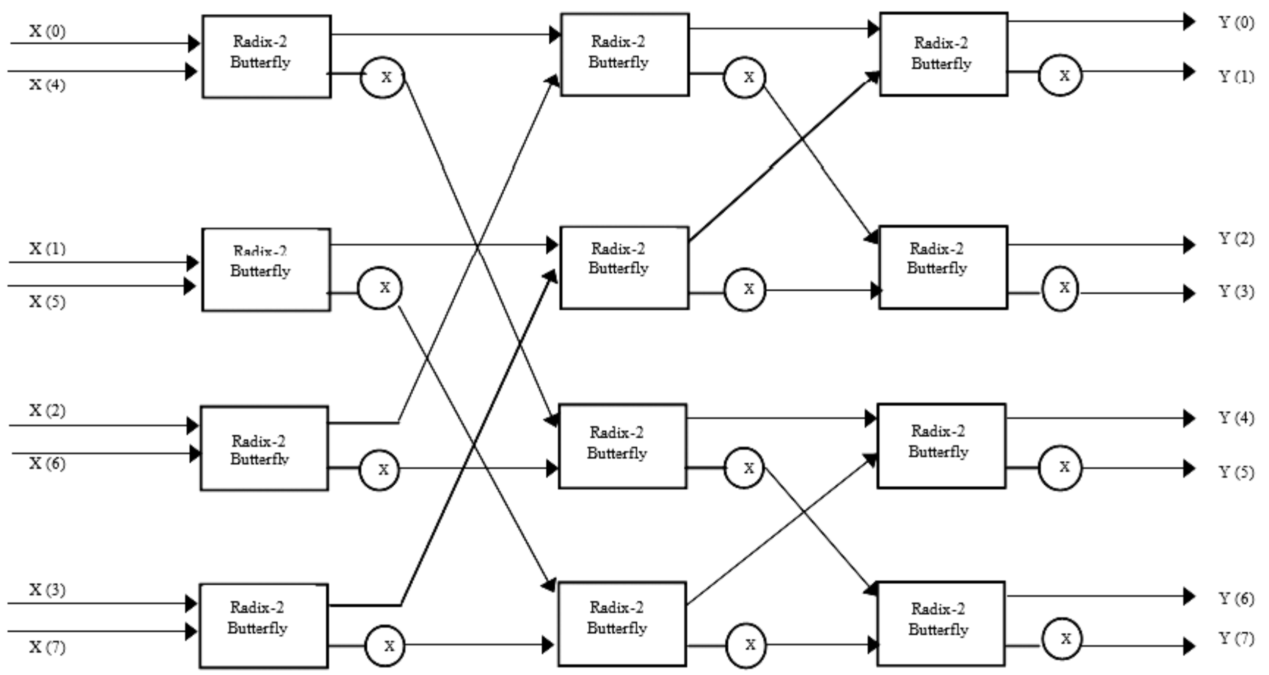


Figure 6: 8-point DIF FFT Block

Radix 2 Butterfly unit

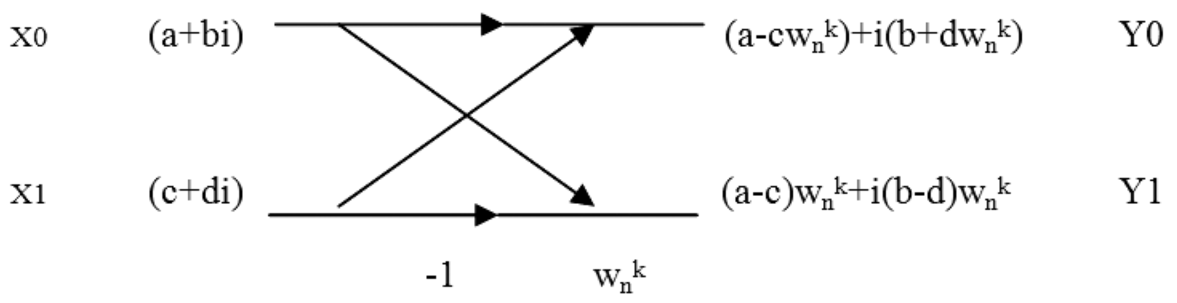


Figure 7: DIF Butterfly Unit

Figure 6 shows the basic butterfly unit for DIT block a+bi and c+di are the two inputs available at the input of Radix-2 Butterfly unit, where a,b are the real part of the input sample and b,d are the imaginary part of the input sample. The input c+di is multiplied with the twiddle factor w_n^K.The butterfly unit will have two outputs namely O1 and O2.

$$O1=(a-c)+i(b-d)\ w_n^k$$

$$O2=(a-c)\ w_n^k+i(b-d)\ w_n^k$$

Where (a-c) is the real part of O1 and (b-d) w_n^k is the imaginary part of the O1

(a-c) w_n^k is the real part of O2 and (b-d) w_n^k is the imaginary part of O2

2.4. N/2 Butrterfly Operations

The outputs obtained from DIT-FFT block and DIF FFT block are further processed by N/2 Butterfly where radix-2 butterfly operations are performed on the outputs obtained from the eight-point DIT FFT and DIF-FFT which is the output of sixteen-point FFT.

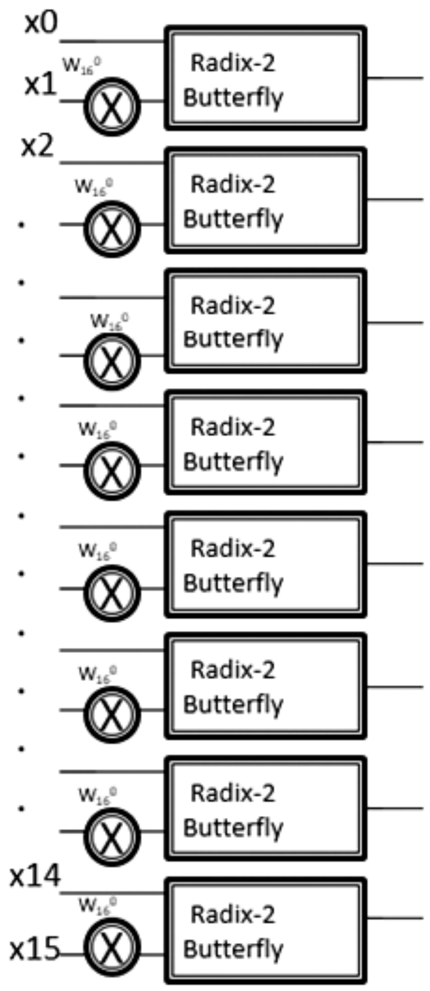


Figure 8: 8-Butterfly Operations

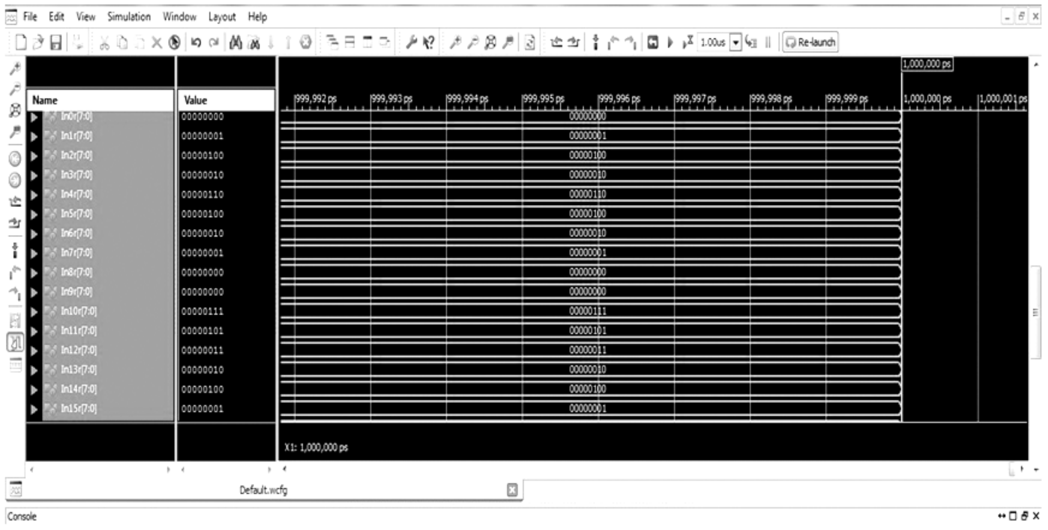
3. EXPERIMENT AND RESULT

The proposed FFT architecture was designed in Verilog and synthesized by Xilinx ISE 14.1 design suite and implemented in Xilinx Virtex 5 board with XC5VLX110T-1 FF1136 as a target device.

3.1. Simulation Result

The sixteen point input samples of [0,1,4,2,6,4,2,1,0,0,7,5,3,2,4,1] is separated into two eight point samples (eight odd samples and eight even samples).The input samples are shown in figure 8, where Inr is the real part of the input sample and Ini is the imaginary part of the input sample. Eight even samples of the input [0,4,6,2,0,7,3,4] will be the input to the eight point DIT-FFT block. The output of this eight point sample [26,-8,-9-5i,-9+5i,-1+2i,1-8i,1+8i,-1-2i] is given in figure 9.Similarly eight odd samples of the input [1,2,4,1,0,5,2,1] will be the input to the eight point DIF-FFT block. The output of this eight point sample [16,-7+i,-2-2i,5+7i,0,5-7i,-2+2i,-7-i] is given in figure 10.These two output sequence are processed by final eight butterfly units to produce the output of sixteen point FFT [42 ,-2 + 1i ,-16 - 5i, 6 + 5i, -8 + 2i , -4 - 8i, -2 + 5i, 1+i, 10, 1 - i, -2 - 5i ,-4 + 8i, -8 - 2i ,6 - 5i, -16 + 5i ,-2 - 1i] which is given in figure 11.

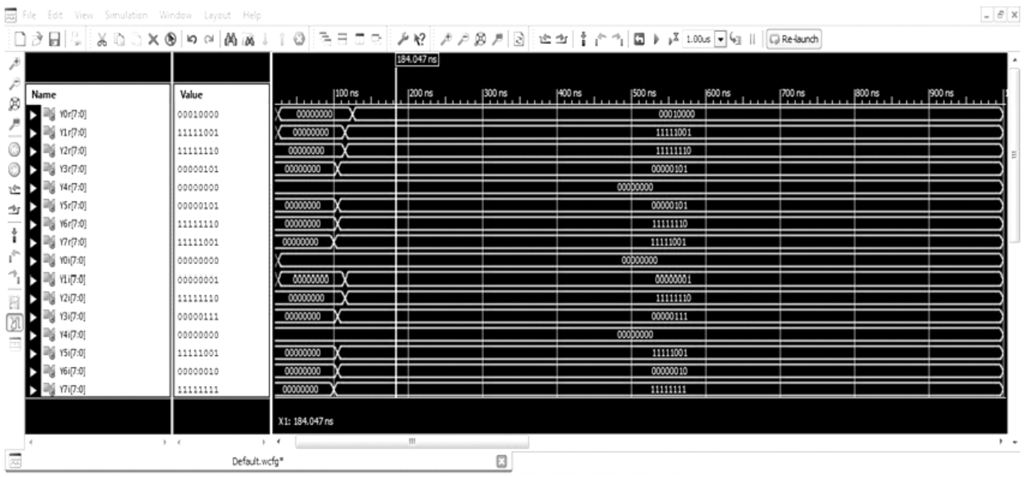
Or is the real part of the output and Oi is the imaginary part of the output. Input and output samples are represented in 8-bits 2's complement form.



```
Name          Value
In0r[7:0]     00000000
In1r[7:0]     00000001
In2r[7:0]     00000100
In3r[7:0]     00000010
In4r[7:0]     00000110
In5r[7:0]     00000100
In6r[7:0]     00000010
In7r[7:0]     00000001
In8r[7:0]     00000000
In9r[7:0]     00000000
In10r[7:0]    00000111
In11r[7:0]    00000101
In12r[7:0]    00000011
In13r[7:0]    00000010
In14r[7:0]    00000100
In15r[7:0]    00000001

X1: 1,000,000 ps
```

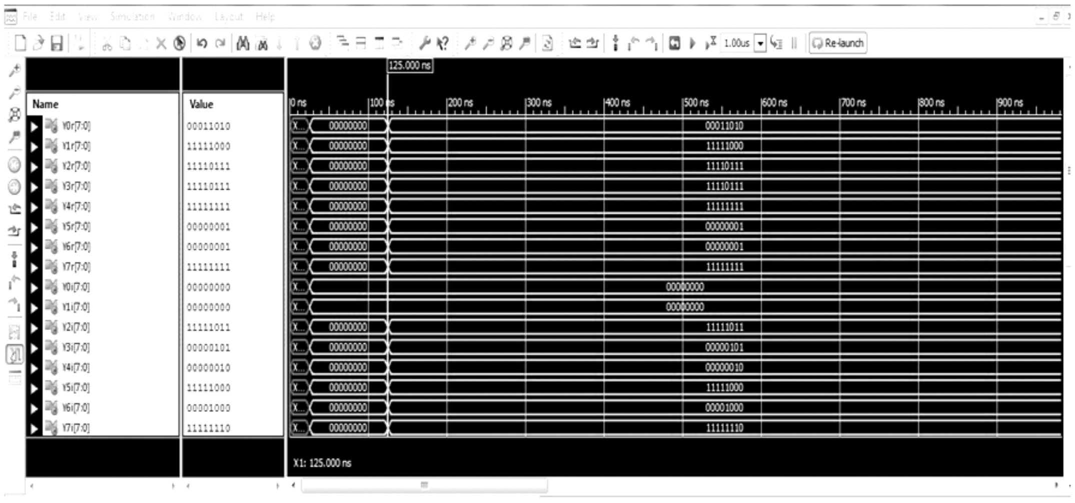
Figure 9: 16-point FFT Input samples



```
Name        Value
Y0r[7:0]    00010000
Y1r[7:0]    11111001
Y2r[7:0]    11111110
Y3r[7:0]    00000101
Y4r[7:0]    00000000
Y5r[7:0]    00000101
Y6r[7:0]    11111110
Y7r[7:0]    11111001
Y0i[7:0]    00000000
Y1i[7:0]    00000001
Y2i[7:0]    11111110
Y3i[7:0]    00000111
Y4i[7:0]    00000000
Y5i[7:0]    11111001
Y6i[7:0]    00000010
Y7i[7:0]    11111111

X1: 184.047 ns
```

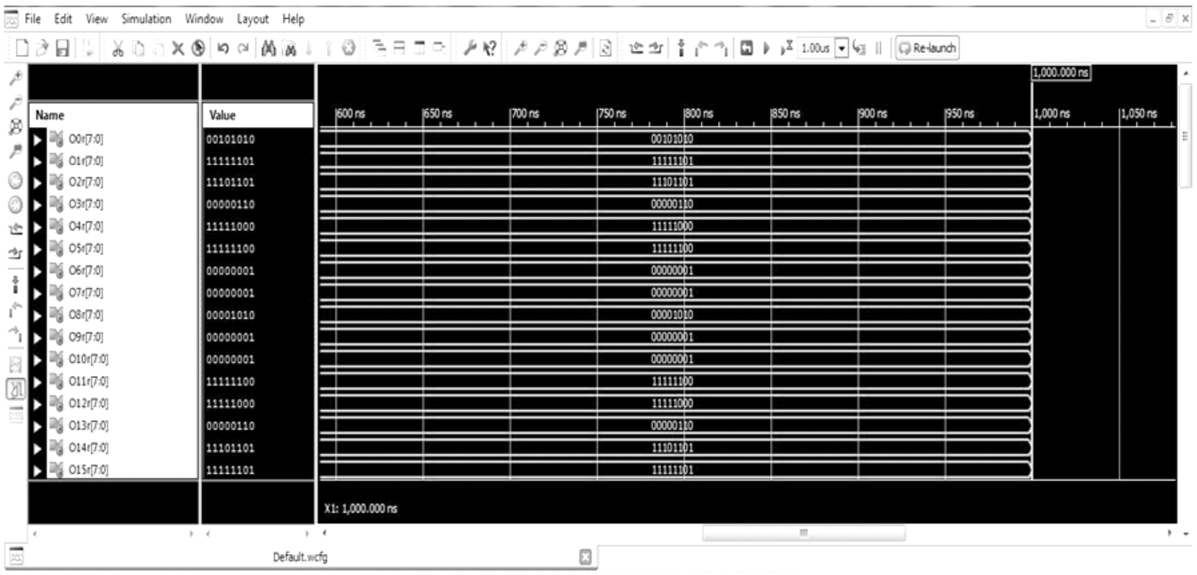
Figure 10: 8-point Output of DIT-FFT block



```
Name        Value
Y0r[7:0]    00011010
Y1r[7:0]    11111000
Y2r[7:0]    11110111
Y3r[7:0]    11110111
Y4r[7:0]    11111111
Y5r[7:0]    00000001
Y6r[7:0]    00000001
Y7r[7:0]    11111111
Y0i[7:0]    00000000
Y1i[7:0]    00000000
Y2i[7:0]    11111011
Y3i[7:0]    00000101
Y4i[7:0]    00000010
Y5i[7:0]    11111000
Y6i[7:0]    00001000
Y7i[7:0]    11111110

X1: 125.000 ns
```

Figure 11: 8-point Output of DIF-FFT Block



Name	Value
O0r[7:0]	00101010
O1r[7:0]	11111101
O2r[7:0]	11101101
O3r[7:0]	00000110
O4r[7:0]	11111000
O5r[7:0]	11111100
O6r[7:0]	00000001
O7r[7:0]	00000001
O8r[7:0]	00001010
O9r[7:0]	00000001
O10r[7:0]	00000001
O11r[7:0]	11111100
O12r[7:0]	11111000
O13r[7:0]	00000110
O14r[7:0]	11101101
O15r[7:0]	11111101

Figure 12: 16-Point FFT Outuput samples

3.2. Synthesis Result

The conventional method of computing FFT and the proposed method are synthesized in Xilinx ISE 14.1 and implemented in Xilinx Virtex 5 board with XC5VLX110T-1 FF1136 as a target device to compute the area complexity and computational time.

The synthesis report obtained in Xilinx is used to compare the efficiency of proposed method with the conventional method. Table 1 shows the comparison of logic utilization of the proposed method with the conventional one. It is evident that the proposed method reduces the hardware resources.

Table 1
Comparison of Hardware Utilization

S. No	*Logic Utilization*	*Existing Method*	*Proposed Method*	*Percentage of reduction*
1	Number of Slice Registers	816	616	24%
2	Number of Slice LUTs	2254	2236	1%
3	Number of fully used FF-LUT pairs	547	378	31%

The conventional method utilizes 816 slice registers out of 69120 slice registers, 2254 Slice LUT's out of 69120 and 547 fully used FF-LUT pairs out of 2474.The proposed method utilizes 616 out of 69120 slice registers, 2236 Slice LUT's out of 69120 and 378 fully used FF-LUT pairs out of 2474.

Table 2 shows the comparison of computational time of the proposed method with the conventional method. This show that computational time is reduced by 28% in the proposed method thus makes it suitable for high speed real time applications.

Table 2
Timing Report

	Existing method	*Proposed method*	*Percentage reduction*
Computational time(ns)	175	125	28%

4. CONCLUSION

The MIMO-OFDM is a communication system that need to process more than one data at a same time in high speed with less amount of hardware. FFT is a main module in OFDM. The proposed architecture increases the speed by splitting inputs into two independent data streams and performs computation using two N/2 point FFTs simultaneously. This proposed architecture is simulated and synthesized using Xilinx ISE 14.1 tool. From the area utilization table as given in Table 1, the area required to implement the proposed method is small when compared to the existing method. The computational time of this FFT unit is 125 ns which is reduced by 28% from existing method. This makes the proposed architecture is suitable for high speed real time MIMO-OFDM applications. This method can be implemented in the MIMO-OFDM applications which require small area and in future this method can be extended to pipelined architecture to increase the throughput.

REFERENCES

[1] IEEE standard for local metropolitan area networks – part 16: Air interface of broadband wireless access systems.

[2] Ramjee Prasad Richard Vaqn Nee. OFDM for Wireless Multimedia Communications.

[3] R. W. Stewart Y.Awad, L.H. Crockett. OFDM Transceiver for IEEE802.20Standards. Dept. of Electrical Engineering, University of Strathclyde. Glasgow, UK.

[4] Fast Fourier Transform - Algorithms and Applications - Rao, K. R., Kim, Do Nyeon, Hwang, Jae Jeong

[5] P. P. Boopal, M. Garrido, and O. Gustafsson, "A reconfigurable FFT architecture for variable-length and multi-streaming OFDM standards," in Proc. IEEE ISCAS, May 2013, pp. 2066–2070.

[6] S.-G. Chen, S.-J. Huang, M. Garrido, and S.-J. Jou, "Continuous-flowparallel bit-reversal circuit for MDF and MDC FFT architectures,"IEEETrans. Circuits Syst. I, Reg. Papers, vol. 61, no. 10, pp. 2869–2877,Oct. 2014.

[7] K.-J. Yang, S.-H. Tsai, and G. C. H. Chuang, "MDC FFT/IFFTprocessor with variable length for MIMO-OFDM systems,"IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 4, pp. 720–731,Apr. 2013.

[8] M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bitreversal," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 10,pp. 657–661, Oct. 2011.

[9] Chen, S.-G. , Huang, S.-J. , Garrido, M. and Jou, S.-J. (2014) 'Continuous-flow parallel bit-reversal circuit for MDF and MDC FFT architectures', IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 10, pp. 2869–2877.

[10] Glittas, A.X. , Sellathurai, M and Lakshminarayanan, G. (2016) "A Normal I/O Order Radix-2 FFT Architecture to Process Twin Data Streams for MIMO", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6.