

# Design of 2x2 MIMO OFDM Architecture for Fixed Wimax

Ganeshkumar.S

K.S.Rangasamy College of Technology,  
Tiruchengode

Venkatesh.S

K.S.Rangasamy College of Technology,  
Tiruchengode

## ABSTRACT

Multiple Input Multiple Output Orthogonal Frequency Division Multiplexing technology is an advanced transmission technique for wireless communication systems. In this paper, the 64 point pipeline FFT/IFFT processor is introduced for efficient implementation of OFDM architecture. The IFFT processor is used to modulate the subcarrier in transmitter section and FFT processor demodulate the subcarrier in receiver section in the architecture. Our design adopts a single-path delay feedback style requiring less memory space and reconfigurable complex constant multiplier and bit parallel multiplier used in pipeline FFT/IFFT processor, instead of using ROM's to store twiddle factors that consuming lower power. The design of ROM-less FFT/IFFT processor is applied to OFDM architecture in the IEEE 802.16d communication standard. The result shows overall architecture design using the FFT/IFFT processor that gives efficient power, area and timing specifications considerably.

## Index Terms

MIMO, OFDM, FFT, IFFT.

## 1. INTRODUCTION

The rapid growth of digital communication in recent years, which need for high-speed data transmission, has been increased. The mobile telecommunications industry faces the problem of providing the technology that be able to support a variety of services ranging from voice communication with a bit rate of a few kbps to wireless multimedia in which bit rate up to 2 Mbps. Many systems have been proposed and OFDM system has gained much attention for different reasons. Although OFDM was first developed in the 1960s, only in recent years, it has been recognized as an outstanding method for high-speed cellular data communication where its implementation relies on very high-speed digital signal processing. This method has only recently become available with reasonable prices versus performance of hardware implementation.

Generally, the pipeline FFT processors have two popular design types. One uses single-path delay feedback (SDF) pipeline architecture and the other uses multiple-path delay commutator (MDC) pipeline architecture. The single-path delay feedback (SDF) pipeline FFT is good in its requiring less memory space (about  $N-1$  delay elements) and its multiplication computation utilization being less than 50%, as well as its control unit being easy to design. Such implementations are advantageous to low-power design, especially for applications in portable DSP devices. Based on

these reasons, the SDF pipeline FFT is adopted in our work. However, the FFT computation often needs to multiply input signals with different twiddle factors for an outcome, which results in higher hardware cost because a large size of ROM is needed to store the wanted twiddle factors. Therefore, to throw off these ROM's for area-efficient consideration, Mao-Hsu Yen, have proposed an efficient ROM-less FFT/IFFT processor. The complex multipliers used in the processor are realized with shift-and-add operations. Hence, the processor uses only a two-input digital multiplier and does not need any ROM for internal storage of coefficients. However, low speed and higher hardware cost caused by the proposed complex multiplier are the pay-off. Lin et al have employs a smart structure for ROM-size reduction to produce twiddle factors as well as to compact the chip area.

In MIMO-OFDM systems, the information bits are carried by the transmitter and receiver section. Here transmitter contains scrambler, encoder, interleaver, modulation, IFFT processor, pilot insertion, adding cyclic prefix and preamble then convert into the RF frequency to the antenna. The reverse operation is performed in the receiver section. In order to reduce the area and power consumption in the FFT processor, the overall architecture chip area is minimized. That will give the efficient OFDM architecture for reliable data transmission.

This paper is organized as follows. In section 2, the system block diagram of the MIMO-OFDM is shown, which presents each block design explanation in the OFDM systems including the proposed pipeline FFT/IFFT processor for application in wireless communication systems. In section 3, Simulation results, area and power report analysis. Finally, Conclusions are in section 4.

## 2. OFDM ARCHITECTURE

The architecture of the transmitter and receiver is illustrated in Figure 1 and Figure 2 respectively [2], [3]. The bit stream which has been scrambled and interleaved is separated into spatial streams by stream parser. Then the spatial streams are mapped into constellation. The points on the constellation are through the STBC encoder to transform the spatial streams to space-time streams. After that spatial mapper maps space-time streams into transmit chains. And the transmit chains are inserted pilot IFFT modulated, added CP (Cyclic Prefix), then transmitted through the RF modules. The transmitted signals are received through RF modules and remove CP. The received chains are FFT modulated, pilot extraction, channel estimation and passed through the STBC decoder to transform to spatial streams from space-time stream. The spatial streams are demapped, interleaved, and descrambled to get the original bit stream.

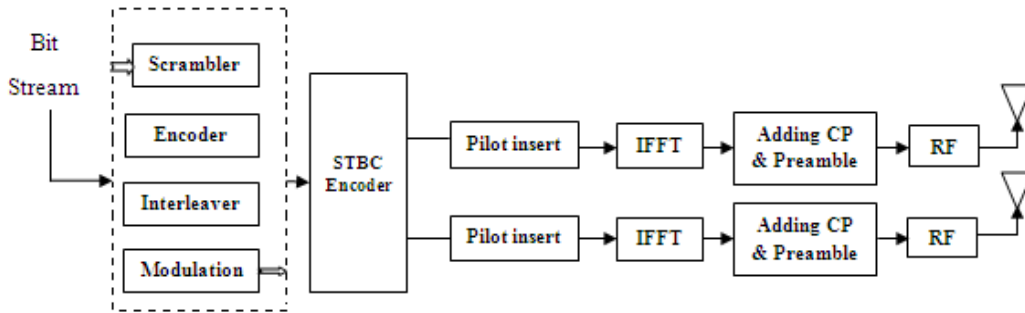


Fig.1 OFDM Architecture of the Transmitter

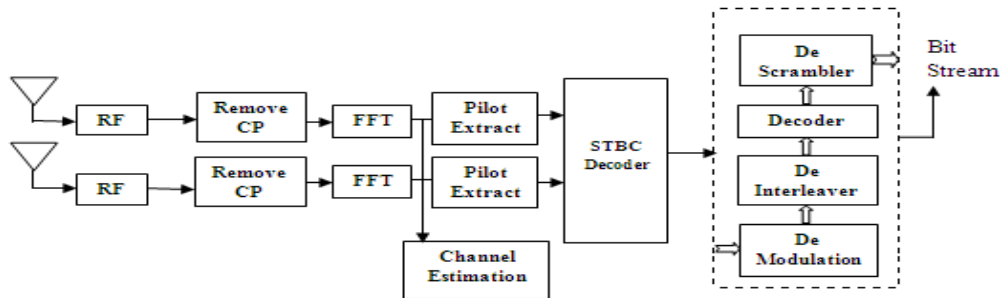


Fig.2 OFDM Architecture of the Receiver

**2.1 Scrambler and Descrambler**

A scrambler is a device that describes randomizes the data stream to remove repeated patterns. In the transmitter, a pseudorandom cipher sequence is added to the data sequence to produce a scrambled data sequence. [4] The pseudorandom cipher sequence is described by the generating polynomial  $G(x) = X^{11} + X^9 + 1$ . In the receiver, the same pseudorandom cipher sequence is subtracted from the scrambled data sequence to recover the transmitted data sequence. The information bits are randomized before the transmission. The randomizer, which is the first block in the transmitter, performs randomization of input data on each burst on each allocation to prevent a long sequence of 1's and 0's. This is implemented by using a Pseudo Random Binary Sequence (PRBS) generator. By using descrambler section the data sequence is retrieved from the scrambled data by reversing the operation.

**2.2 Encoder and Decoder**

When data is transmitted from one location to another there is always the possibility that an error may occur. There are a number of reliable codes that can be used to encode data so that the error can be detected and corrected. By using decoding techniques the redundant information to detect and correct errors occurred during transmission. A Hamming Code can be used to detect and correct one bit change in an encoded code word. The block diagram is shown in Figure 3. This approach can be useful as a change in a single bit is more probable than a change in two bits or more bits.

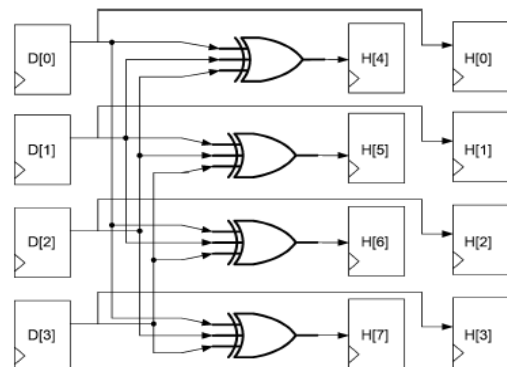


Fig.3 Block Diagram for Hamming Code

**2.3 Modulation and Demodulation**

The Modulator passes interleaved data through a serial to parallel converter, mapping groups of bits to separate carriers, and encoding each bit group by frequency, amplitude, and phase. The QPSK is a phase modulation scheme, used in constellation mapping. [6] The constellation map of QPSK modulator is shown in Figure 4. Here the input bits stream is converted into complex stream using equation (1) and where I and Q both are in phase with I-out and Q-out respectively are shown in table I. QPSK modulator accepts the binary bits as inputs consider as a symbol and converts them into complex value. QPSK takes only 4 symbols and generate its complex value in this fashion because the bit rate is 1/2.

$$D = (I + jQ) * KMOD \text{ where } KMOD = 1/1.414 \dots\dots (1)$$

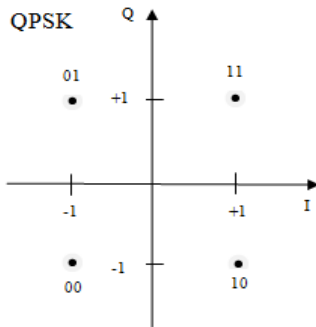


Fig.4 QPSK Constellation Mapping

TABLE I - Inputs and Outputs of QPSK Modulator

Input Bits	I out	Q out
00	-1	-1
01	-1	+1
10	+1	-1
11	+1	+1

### 2.4 STBC Encoder

Space time block coding is a technique used in wireless communications to transmit multiple copies of a data stream across a number of antennas and to exploit the various received versions of the data to improve the reliability of data-transfer. Alamouti's transmit diversity scheme with two transmit antennas and two receive antennas are used. [10] Alamouti's scheme is a space-time block code and suitable when two transmit antennas and an arbitrary number of receive antennas are used.

### 2.5 Pilot Insertion

Pilot insertion adds the values for pilot and guard subcarriers to OFDM symbols. A transmitting entity transmits a "base" pilot in each protocol data unit (PDU). A receiving entity is able to derive a sufficiently accurate channel response estimate of a MIMO channel with the base pilot under nominal (or most) channel conditions. The transmitting entity selectively transmits an additional pilot if and as needed, e.g. (Figure 5), based on channel conditions and/or other factors. The additional pilot may be adaptively inserted in almost any symbol period in the PDU.

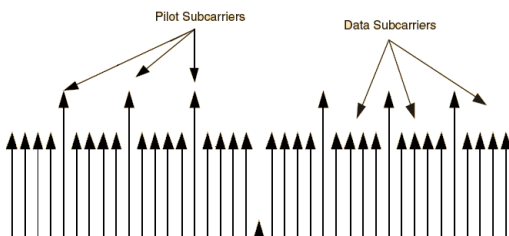


Fig.5 Pilot insertion in Subcarriers

The receiving entity is able to derive an improved channel response estimate with the additional pilot. The transmitting entity sends signaling to indicate that additional pilot is being sent. [5] This signaling may be embedded within pilot symbols

sent on a set of pilot sub bands used for a carrier pilot that is transmitted across most of the PDU. The signaling indicates whether additional pilot is being sent and possibly other pertinent information.

### 2.6 Cyclic Prefix and Preamble

The cyclic prefix refers to the prefixing of data with a repetition of the end. As a guard interval, it eliminates the inter symbol interference from the previous data. It allows the linear convolution of a frequency-selective multipath channel to be modeled as circular convolution, which in turn may be transformed to the frequency domain using FFT. This approach allows for simple frequency-domain processing, such as channel estimation and equalization. [10] Preamble is used for synchronization and channel estimation at the receiver.

### 2.7 Proposed FFT/IFFT Processor

#### 2.7.1 Rom-Less FFT/IFFT Processor

Traditional hardware implementation of FFT/IFFT processors usually employs a ROM to look up the wanted twiddle factors, and then word length complex multipliers to perform FFT computing. However, this introduces more hardware cost, thus a bit-parallel complex constant multiplication scheme is used to improve the foregoing issue. Besides, since the twiddle factors have a symmetric property. The complex multiplications used in FFT computation is one of the following three operation types [1]:

$$\text{Type1: } W_N^k \cdot (a + jb) = W_N^{k-(N/4)} \cdot (b - ja), \quad N/4 < k < N/2, \dots (1)$$

$$\text{Type2: } W_N^k \cdot (a + jb) = -W_N^{k-(N/2)} \cdot (b - ja), \quad N/2 < k < 3N/4, \dots (2)$$

$$\text{Type3: } W_N^k \cdot (a + jb) = -W_N^{k-(3N/4)} \cdot (b - ja), \quad 3N/4 < k < N, \dots (3)$$

Any twiddle factor can be obtained by a combination of these twiddle-factor primary elements. In other words, arbitrary twiddle factor used in FFT can utilize these operation types to derive the wanted value, thus can significantly shorten the size of ROM used to store the twiddle factors. Moreover, for hardware implementation consideration, we add two extra operation types to further decrease the size of ROM. Our method can also run away the critical path in the designed hardware such that the system clock becomes faster compared to [7]-[9]. The two additional operation types are given by:

$$\text{Type 4: } W_N^k \cdot (a + jb) = [W_N^{(N/4)-k} \cdot (b + ja)], \quad 1 \leq k < N/4, \dots (5)$$

$$\text{Type 5: } W_N^k \cdot (a + jb) = -j \cdot [W_N^{(N/2)-k} \cdot (b + ja)], \quad N/4 < K < N/2, \dots (6)$$

A radix-2 64-point pipeline FFT/IFFT processor with low power consumption, as shown in Figure 6. [1], [3] The proposed architecture is composed of three different types of processing elements (PEs), a complex constant multiplier, delay-line (DL) buffers, and some extra processing units for computing IFFT. Here, the conjugate for extra processing units is easy to implement, which only takes the 2's complement of the imaginary part of a complex value.

### 2.7.2 Processing Elements

Based on the radix-2 FFT algorithm, the three types of processing elements (PE3, PE2, and PE1) used in our design are illustrated in Figure 7, Figure 8, and Figure 9, respectively. The functions of these three PE types correspond to each of the butterfly stages as shown in Figure 6. First, the PE3 stage is used to implement a simple radix-2 butterfly structure only, and serves as the sub modules of the PE2 and PE1 stages.

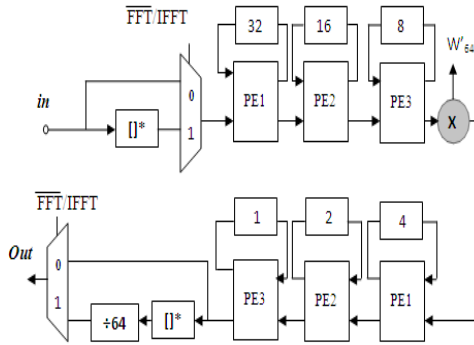


Fig.6 Radix-2 64 Point Pipeline FFT/IFFT

In the Figure 7,  $I_{in}$  and  $I_{out}$  are the real parts of the input and output data, respectively.  $Q_{in}$  and  $Q_{out}$  denote the image parts of the input and output data, respectively. Similarly,  $DL_{I_{in}}$  and  $DL_{I_{out}}$  stand for the real parts of input and output of the DL buffers, and  $DL_{Q_{in}}$  and  $DL_{Q_{out}}$  are for the image parts, respectively.

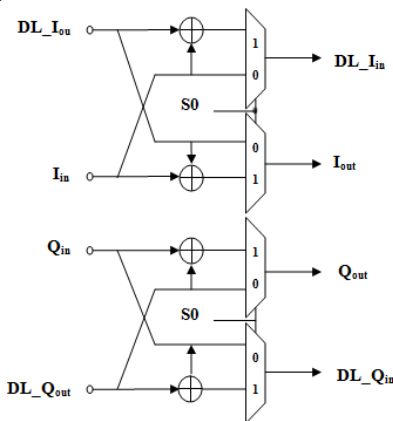


Fig.7 Circuit Diagram of PE3 Stage

As for the PE2 stage, it is required to compute the multiplication by  $-j$  or  $1$ . Note that the multiplication by  $-1$  in figure 6 is practically to take the 2's complement of its input value. In the PE1 stage, the calculation is more complex than the PE2 stage, which is responsible for computing the multiplications by  $-j$ ,  $W_N^{N/8}$ , and  $W_N^{3N/8}$ , respectively.

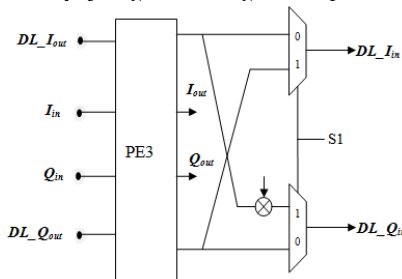


Fig.8 Circuit Diagram of PE2 Stage

Since  $W_N^{3N/8} = -j W_N^{N/8}$ , it can be given by either the multiplication by  $W_N^{N/8}$  first and then the multiplication by  $-j$  or the reverse of the previous calculation. Hence, the designed hardware utilizes this kind of cascaded calculation and multiplexers to realize all the necessary calculations of the PE1 stage. This manner can also save a bit-parallel multiplier for computing  $W_N^{3N/8}$ , which further forms a low-cost hardware.

### 2.7.3 Bit-Parallel Multipliers

In this section the multiplication by  $1/2$  can employ a bit parallel multiplier to replace the word length multiplier and square root evaluation for chip area reduction.

$$\text{Output} = in \times \frac{\sqrt{2}}{2} = in \times (2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-14}) \dots (7)$$

If a straightforward implementation for the above equation is adopted, it will introduce a poor precision due to the truncation error, and will spend more hardware cost. Therefore, to improve the precision and hardware cost, Eq. (7) can be rewritten as:

$$\text{Output} = in \times \frac{\sqrt{2}}{2} = in \times [1 + (1 + 2^{-2})(2^{-6} + 2^{-2})] \dots (8)$$

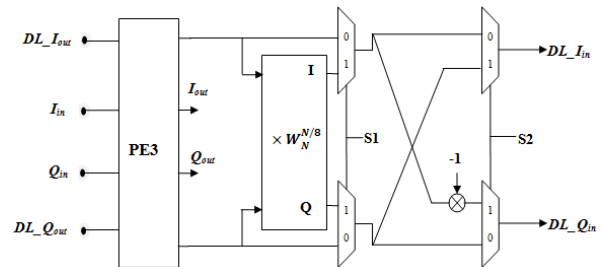


Fig.9 Circuit diagram of our proposed PE1 stage

According to (8), the circuit diagram of the bit-parallel multiplier is illustrated in fig.10. The resulting circuit uses three additions and three barrel shift operations. The realization of complex multiplication by  $W_N^{N/8}$  using a radix-2 butterfly structure with its both outputs commonly multiplied by  $1/\sqrt{2}$  is shown in Figure 11. This circuit has just been used in the PE1 stage.

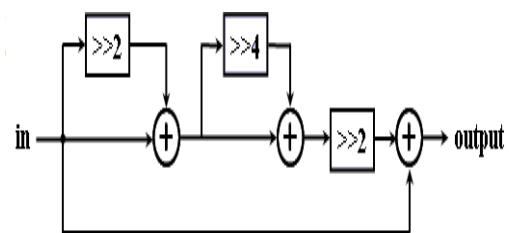


Fig.10 Circuit diagram of the Bit-Parallel Multiplication by  $1/\sqrt{2}$

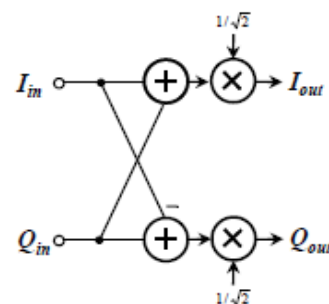


Fig.11 Circuit diagram of the multiplication by  $W_N^{N/8}$

### 2.7.4 Reconfigurable Complex Constant Multipliers

A reconfigurable complex constant multiplier for computing  $W^{i64}$  is proposed, as shown in Figure 10 and Figure 11. This structure of this complex multiplier also adopts a cascaded scheme to achieve low-cost hardware. Here, the meaning of two input signals ( $I_{in}$  and  $Q_{in}$ ) and two output signals ( $Q_{in}$  and  $Q_{out}$ ) are the same as the signals in the PE1 stage. In Figure 12, this circuit is responsible for the computation of multiplication by a twiddle factor  $W^{i64}$ , which is also an important circuit of our FFT/IFFT processor. The word length multiplier used in Figure 13 adopts a low-error fixed width modified Booth multiplier for hardware cost reduction.

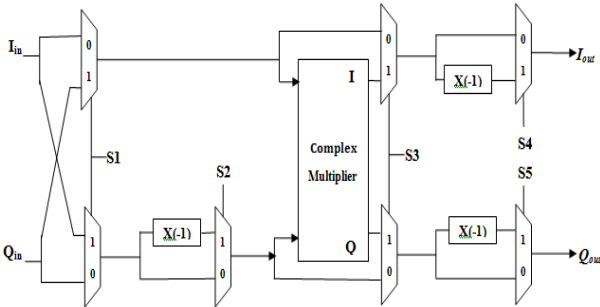


Fig.12 Proposed Reconfigurable Complex Constant Multiplier.

The coefficient values  $i1-i8$  and  $q1-q8$  are listed in Table I, which can be used to synthesize the entire twiddle factors required in our proposed 64-point FFT processor. Besides, we need not to use bit-parallel multipliers to replace the word length one for two reasons. One is on the operation rate.

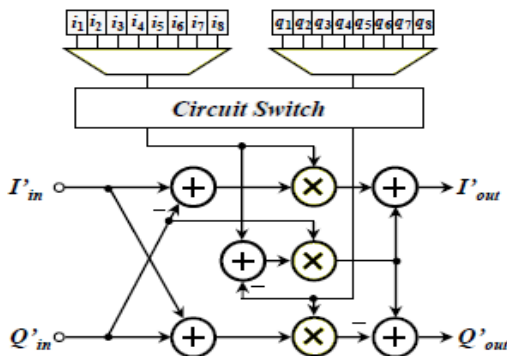


Fig.13 Complex multiplier used in Fig.16

If bit-parallel multipliers are used, the clock rate is decreased due to the many cascaded adders. The other reason is the introduction of high wiring complexity because many bit-parallel multipliers are required to be switched for performing multiplication operations with different twiddle factors. Besides, we need not to use bit-parallel multipliers to replace the word length one for two reasons. One is on the operation rate. If bit-parallel multipliers are used, the clock rate is decreased due to the many cascaded adders.

### 3. Simulation and Synthesis Results

The simulation of OFDM architecture was described in VHDL and the simulation was done in ModelSim and the code was functionally verified to be correct. The simulation results of various blocks in the OFDM architecture are shown in the Figure 14, 15, 16, and 17 respectively.

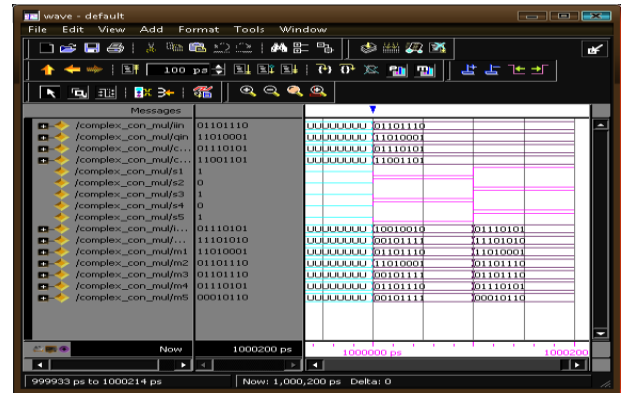


Fig.14 Simulation result of Complex Constant Multiplier

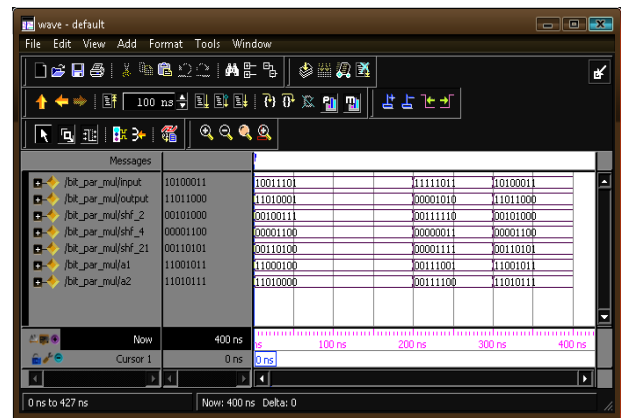


Fig.15 Simulation result of Bit Parallel Multiplier

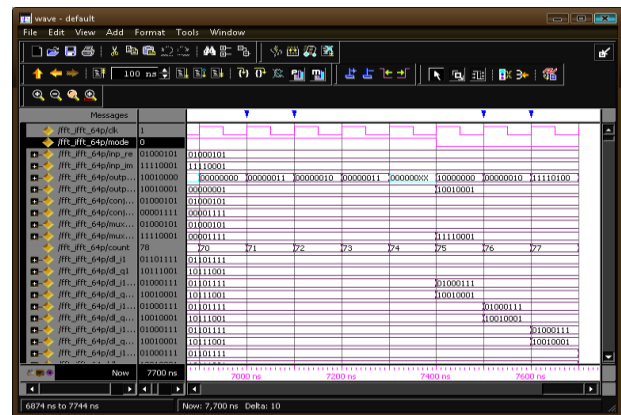


Fig.16 Simulation result of Radix-2 64 Point Pipeline FFT/IFFT processor

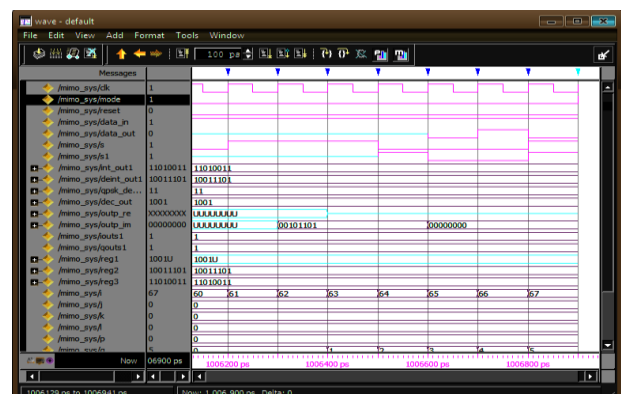


Fig.17 Simulation result of MIMO OFDM architecture



The power and area analysis were done in Xilinx ISE 8.1i tool and was found that for OFDM architecture, the total dynamic power and area specifications were met. The below shown result describes the power and area analysis and the result was found that for an OFDM architecture using Xilinx Synthesis/implementation tool.

### 3.1 Power Report

XPower and Datasheet may have some Quiescent Current differences. This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design scenarios.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		61
Vccint 1.80V:	30	55
Vcco33 3.30V:	2	7
Clocks:	15	27
Inputs:	0	0
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

### 3.2 Area Report

The Table II describes the detailed area utilization of all devices in the architecture. Table III gives comparison analysis of area, power and time specifications between FFT/IFFT processor and OFDM architecture.

TABLE II - Area Utilization Report

Description	Area	Utilization %
Number of Slice Flip Flops	1332 out of 13824	9%
Number of 4 input LUTs	1486 out of 13824	10%
Number of occupied Slices	1270 out of 6912	18%
Total equivalent gate count for design: 20,793		

TABLE III – Comparison Analysis

Description	FFT/IFFT processor	OFDM Architecture
Gate count	10992	20,793
Power	44mW	61mW
Time Delay	11.6ns	47.581ns

## 4. CONCLUSION

The efficient MIMO-OFDM system architecture is described in this work. Considering all the blocks in the architecture were designed especially reconfigurable complex constant multiplier and bit parallel multiplier was designed for pipeline FFT/IFFT block in order to reduce the size of twiddle factor ROM. The result of OFDM architecture shows that the design meets the efficient power, area and timing specifications. Using Xilinx 8.1i tool the architecture consumes 61mW power, 20793 number of gate counts and 47.6ns delay for timing specifications. The future work can extended to analyze the different encoding techniques in the design process and to obtain low power OFDM architecture.

## 5. REFERENCES

- [1] Chu Yu, Mao-Hsu Yen, Pao-Ann Hsiung, Sao-Jie Chen, 2011. A Low-Power 64-point Pipeline FFT/IFFT Processor for OFDM Applications" IEEE Transactions on Consumer Electronics, Vol. 57, No. 1.
- [2] Zoha Pajouhi, Sied Mehdi Fakhraie, Sied Hamidreza Jamali, IEEE 2008. Hardware Implementation of a 802.11n MIMO OFDM Transceiver" International Symposium on Telecommunication.
- [3] R.Premalatha and M.Shanthi 2012 VLSI Implementation of a 2x2 MIMO-OFDM System on FPGA. Bonfring International Journal of Power Systems and Integrated Circuits, Vol. 2, Special Issue 1.
- [4] Alistair Coles, 1997. Scrambler and Descrambler Functions (Informative).
- [5] Ian Griffiths and Brett Ninness 2005 FPGA Implementation of MIMO Wireless Communications system.
- [6] Sudhakar Reddy Penubolu and Ramachandra Reddy Gudheti. 2009. VLSI Implementation of Least Square Channel Estimation and QPSK Modulation Technique for 2x2 MIMO System.
- [7] Chin-Teng Lin, Yuan-Chu Yu, and Lan-Da Van, 2006. A low-power 64-point FFT-IFFT design for IEEE 802.11a WLAN application," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), pp. 4523-4526.
- [8] Yuan Chen, Yu-Wei Lin, and Chen-Yi Lee, 2006. A Block Scaling FFT/IFFT Processor for WiMAX Applications, in Proc. IEEE Asian Solid-state Circuits Conf., pp. 203-206.
- [9] Bruno Fernandes. Helena Sarmento. 2012. FPGA implementation and testing of a 128 FFT for a MB-OFDM receiver. Analog Integr Circ Sig Process.
- [10] Samar Jyoti Saikia, Kandarpa Kumar Sarma, 2012. ANN based STBC-MIMO set-up for Wireless Communication" International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) ISSN No. 2248-9738 Volume-1, Issue-3.
- [11] Nikolaos Bartzoudis, Oriol Font-Bach, Antonio Pascual-Iserte and David López Bueno. 2011. Real-Time FPGA-based mobile WiMAX transceiver supporting multi-antenna configurations. Argentine School of Micro-Nanoelectronics, Technology and Applications.