

# Design and Implementation of Low Power and High Speed Autocorrelator and CORDIC Architectures for OFDM

Iqbal Hussain and Cyril Prasanna Raj

**Abstract**—OFDM systems are currently being widely used in 4G and UWB communication systems. OFDM system consisting of IFFT and modulator increases the complexity of hardware implementation. In this paper, design and implementation of Autocorrelator and CORDIC algorithm for OFDM is discussed for optimized power and delay. Autocorrelator is used for frame detection and carrier frequency offset estimation. The CORDIC is used to estimate the frequency offset and to calculate the division in the channel estimation algorithm. HDL and test bench is developed to simulate and verify the functionality of both the modules. The design is implemented on Xilinx FPGA and semicustom ASIC targeting 130nm technology. The autocorrelator design is optimized to consume 5684 cells (13969.5 $\mu\text{m}^2$  of area) with power reduced to 78.7 $\mu\text{W}$ . The CORDIC architecture occupies 2288  $\mu\text{m}^2$  total cell area. The total area comprising of both the modules is 2558  $\mu\text{m}^2$  consuming 12.05 $\mu\text{W}$ .

**Index Terms**—Autocorrelator, CORDIC, OFDM, FPGA, ASIC.

## I. INTRODUCTION

Today's communication standards such as Hiperlan/2, IEEE 802.11a/g, IEEE 802.16 and digital video broadcasting (DVB) already use orthogonal frequency division multiplexing (OFDM) for digital modulation. OFDM is one such modulation technique that takes into account the issues concerning the reliability and speed of network. With these vital advantages, OFDM has been adopted by many wireless standards such as DAB, DVB, Wireless LAN, and WMAN. MIMO, on the other hand, employs multiple antennas at the transmitter and receiver sides to open up additional sub channels in spatial domain. Since parallel channels are established over the same time and frequency, high data rates are achieved without the need of extra bandwidth. Due to this bandwidth efficiency, MIMO is included in the standards of future BWA. Overall, these benefits have made the combination of MIMO-OFDM an attractive technique for future high data rate systems [1], [2].

## II. ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING

A combination of modulation and multiplexing constitutes to orthogonal frequency division multiplexing, in other words OFDM. Independent signals that are a sub-set of a main signal are multiplexed in OFDM and also the signal itself is first split into independent channels, modulated by

data and then re-multiplexed to create the OFDM carrier. Orthogonality of the sub-carriers is the main concept in OFDM. The property of Orthogonality allows simultaneous transmission on a lot of sub-carriers in a tight frequency space without interference from each other. This acts as an undue advantage in OFDM [3][4][5]. Therefore, OFDM is becoming the chosen modulation technique for wireless communication. With the help of OFDM, sufficient robustness can be achieved to provide large data rates to radio channel impairments. In an OFDM scheme, a large number of orthogonal, overlapping narrow band sub-channels or sub-carriers transmitted in parallel divide the available transmission bandwidth. Compact spectral utilization with utmost efficiency is achieved with the help of minimally separated sub-carriers. Main attraction of OFDM lies with how well the system handles the multi-path interference at the receiver end. Fig 1 shows the block diagram of transmitter and receiver section of an OFDM system [1]. The blocks circled in Fig.1 are the area of interest of this work and details of which are provided in the next section.

## III. AUTOCORRELATION

Autocorrelation is the cross-correlation of a signal with itself. It is a measure of how well a signal matches a time shifted version of itself, as a function of the amount of time shift. Autocorrelation is useful for finding repeating patterns in a signal, such as determining the presence of a periodic signal which has been buried under noise or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. Autocorrelation involves only one signal and provides information about the structure of the signal or its behavior in the time domain. In the receiver, the radio frequency (RF) data is converted to digital data and is down converted to intermediate frequency (IF) band. The down converted signal needs to be synchronized with the carrier signal. Any mismatch between the transmitter carrier and receiver carrier is rectified in the synchronizer unit. The autocorrelation equation is given by:

$$r[j] = \frac{1}{N} \sum_{n=0}^{N-1-j} x[n] \cdot x[n+j] \quad (1)$$

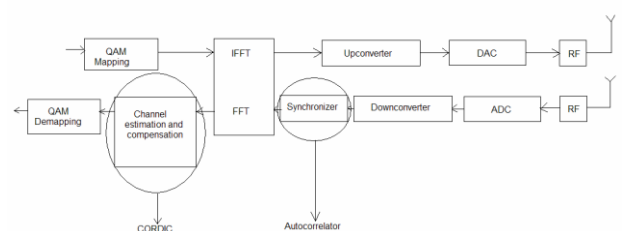


Fig.1 Hiperlan /2Transceiver

Manuscript received February 22, 2011; revised May 12, 2011.

Iqbal Hussain, Senior Design Engineer, Ph: +91 9560259968, E-mail: iqbal.husseini@gmail.com.

Cyril Prasanna Raj, VLSI System design, MSRSAS, Ph: +919448686332, E-mail: cyrilyahoo@gmail.com.

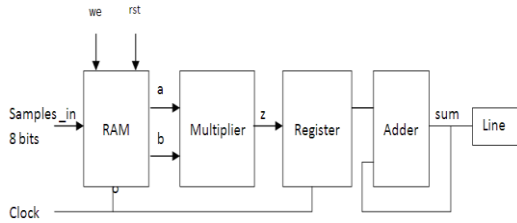


Fig. 2 Block Diagram of Autocorrelator

The synchronizer unit consists of an Autocorrelator. In this design, the samples of the incoming signal are auto correlated and the peak amplitude point is detected and the time delay between the transmitter and receiver carrier is computed. This information of time difference synchronizes the receiver with the transmitter module. The synchronized data is fed into the FFT module for data demodulation. The incoming data samples of 128 samples, each represented by 8 bit signed representation is delayed and is correlated with the incoming signal. Thus, the two signals (incoming signal and time shifted version of the incoming signal) are stored in a memory and are processed. Fig. 2 shows the block diagram of the Autocorrelator design. The design is carried out in Verilog HDL and simulated using ModelSim.

The Autocorrelator is designed in such a way that ‘N’ number of inputs can be fed in. The output port named as ‘line’ displays the auto correlated version of it. In this design a RAM is modeled to accommodate 128 samples each of 8 bits wide. The address of the RAM is generated with the help of a counter and the outputs of the RAM are ‘a’ and ‘b’. For example if the address of ‘a’ is [addr], then address of ‘b’ is minus one position i.e. [addr-1]. Therefore, when the first sample goes into the RAM, it is stored in ‘a’ register and ‘b’ register is 0. When the second sample goes into the RAM, the previous value of the register ‘a’ is stored in ‘b’ and ‘a’ gets the new value. This operation, in other words can be called as the shift operation. Here, each sample is getting shifted by one position i.e. 8 bits. The writing operation is done only when the write enable pin (we) is high and the ‘we’ becomes low when the RAM is filled. Reset pin is turned high for 10ns after which it is made low. The duty cycle of the clock is 100ns. The writing operation occurs in the negative or falling edge of the clock to save the writing time. Therefore, for 128 samples, this design performs 127 forward shifts and produces 127 outputs serially from a single output port namely ‘line’. The accumulator accumulates the result and is fed from an intermediate register only in the event of a clock are shown in fig 3. Multiplier and adder are the two major blocks that consume power. Various multipliers are analyzed for its area and power performances.

A. FPGA Implementation of Multipliers and Adders

The implementation process was done using XILINX-ISE 10.1 version and then implemented on SPARTAN-II board. The synthesis reports taken for the multipliers are presented in Table I. From the FPGA implementation results, Wallace tree multiplier was found to consume less power with minimum delay but at the cost of increased area and therefore was found to be suitable for the design of Autocorrelator; and since Ripple Carry Adders consume less power and area when compared to other adders, they were employed in the design.

TABLE I: THE SYNTHESIS REPORT

	Array Multiplier	Wallace tree multiplier	Baugh-Wooley multiplier	Modified booths multiplier
No. of Slices	159 out of 2352 6%	169 out of 2352 7%	158 out of 2352 6%	274 out of 2352 11%
No. of 4 input LUTs	277 out of 4704 5%	295 out of 4704 6%	275 out of 4704 5%	521 out of 4704 11%
No. of IOs	48	48	48	48
Delay (ns)	48.590	48.239	46.943	66.180
Power dissipation (mW)	1.3403	0.0958113	1.12	1.5224

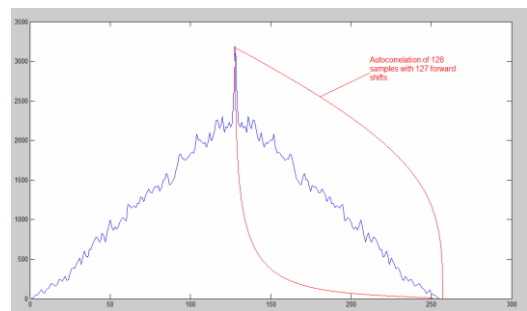


Fig. 3 Auto correlated output of 128 input samples

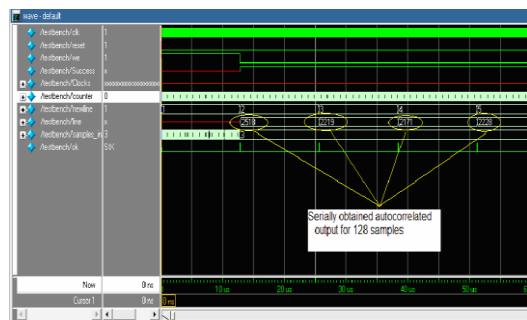


Fig.4 Behavioral simulation of Autocorrelator using test bench

B. Simulation Results of Autocorrelator

The number of slices utilized by the Autocorrelator design on Spartan-3 are 68 out of 768 which is considerably very economical and area saving. The number of bonded IOB’s is 35 out of 124 available resources. The design works at a frequency of 134.252MHz. Therefore this design takes less area and performs the autocorrelation of 128 samples with 127 shifts producing exact outputs. The results are compared with MATLAB simulation and to no surprise they were exact. Fig. 3 and Fig. 4 present the Matlab and HDL simulation results.

IV. COORDINATE ROTATION DIGITAL COMPUTER (CORDIC)

CORDIC is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions. All of the trigonometric functions can be computed or derived from functions using vector rotations. Vector rotation can also be used for polar to rectangular and rectangular to polar conversions, for vector magnitude, and as a building block in certain transforms such as the DFT and DCT [7],[8]. The

CORDIC algorithm provides an iterative method of performing vector rotations by arbitrary angles using only shifts and additions. The algorithm, credited to Volder [2] is derived from the general (Givens) rotation transform as shown in equations (2) and (3).

The basic equations required to implement CORDIC are:

$$X_{i+1} = X_i \cos\phi - Y_i \sin\phi \tag{2}$$

$$Y_{i+1} = Y_i \cos\phi + X_i \sin\phi \tag{3}$$

The equations (2) and (3) modified as follows

$$X_{i+1} = \cos\phi(X_i - Y_i \tan\phi) \tag{4}$$

$$Y_{i+1} = \cos\phi(Y_i + X_i \tan\phi) \tag{5}$$

$$X_{i+1} = K_i[X_i - Y_i d_i 2e^{-i}] \tag{6}$$

$$Y_{i+1} = K_i[Y_i + X_i d_i 2e^{-i}] \tag{7}$$

where  $K_i = \cos(\tan^{-1} 2e^{-i}) = 0.60722$  and  $d_i = \pm 1$

$$Z_{i+1} = Z_i - d_i \phi \tag{8}$$

If the rotation angles are restricted so that  $\tan\phi = \pm 2e^{-i}$ , the multiplication by the tangent term is reduced to simple shift operation. According to [4] and [5], the CORDIC processor can be configured to work as a circular rotation and vectoring mode, and also as linear vectoring mode. The circular vectoring mode is used to calculate coarse Carrier Frequency Offset (CFO) estimation and fine CFO estimation. The circular rotation mode is used to correct coarse CFO of the broadcast preamble, and to correct received section C and received OFDM symbols. The linear vectoring mode is used to calculate the division in the channel estimation stage.

The Fig.5 shows a fully pipelined fast iterative CORDIC architecture [8] that is adopted in this design. The iterative CORDIC architecture can be obtained simply by duplicating each of the above difference equations shown in equations (2) to (5). The decision function decides the sign of the ‘Y’ or ‘Z’ register depending on whether it is operated in rotation or vectoring mode. In operation, the initial values are loaded into the ‘X’ and ‘Y’ registers. Then on each of the next ‘n’ clock cycles, the values from the registers are passed through the shifters and adder-subtractors and the result is placed back in the registers.

### A. Pipelined CORDIC Architecture Design

The simulation results for CORDIC block are discussed. The total number of slices utilized by this design is 112 out of 960 in Spartan-3, which is about 11% utilization factor. The operating frequency for the pipelined architecture of fast CORDIC is 219.443 M Hz. Fig.5 shows the simulation result of the pipelined CORDIC architecture run with the help of an exhaustive test bench.

### B. Simulation Results of CORDIC Design

Fig. 6 shows the simulation result when  $X_0 = 215 = -41 \text{ mod } 256$ , and  $Y_0 = 55$ . The radius ‘r’ =  $1.618 \sqrt{(X_0^2 + Y_0^2)} = 111$ . The accumulated angle in degrees  $\arctan(Y_0/X_0) = 123$  degrees. The operating frequency of the design is 219.44 M Hz.

## V. ASIC IMPLEMENTATION RESULTS

The design of both Autocorrelator and CORDIC was modeled using Verilog HDL and synthesized to implement

on a Spartan-3 Xilinx FPGA. The architectures of both Autocorrelator and CORDIC was further processed through ASIC Design flow to check for the timing of both setup as well as hold time for 130nm technology. Design Compiler tool which helped in generating the timing analysis for the same providing details of area and power for both the architectures. The Autocorrelator design raked up 5684 cells taking up  $13969.5 \mu\text{m}^2$  of area for the cells and total area of  $16609.6 \mu\text{m}^2$  consuming  $78.7 \mu\text{W}$ . The CORDIC design raked up 2288  $\mu\text{m}^2$  total cell area for 552 numbers of cells.

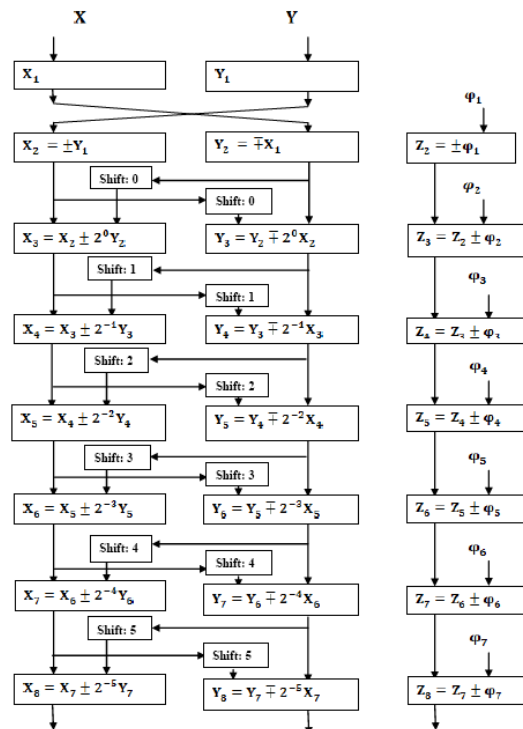


Fig. 5 Fast CORDIC Pipe line [9]

The total area comprising  $2558 \mu\text{m}^2$  consuming  $12.05 \mu\text{W}$ . The SDC files created during DC flow were taken as inputs to perform Physical Design. Prior to Physical Design, Prime time tool was used in performing timing analysis to check for setup and hold time.

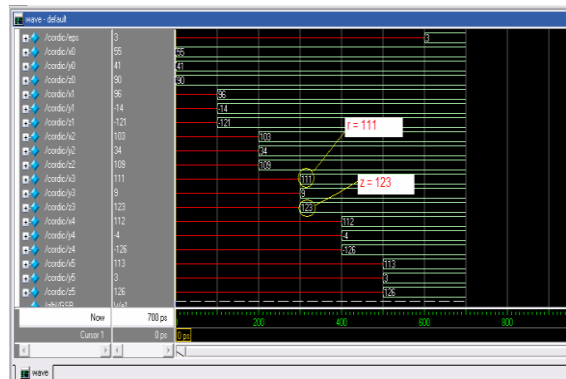


Fig. 6 Simulation Result of Pipelined Fast CORDIC

## VI. CONCLUSION

An Autocorrelator was designed to perform the autocorrelator of 128 samples each of 8 bits wide. The building blocks for the scheme were a 128x8 RAM, a multiplier, an accumulator and a counter. Matlab simulations were performed prior to the Verilog HDL coding to check if

the functionality was achieved.

A fully pipelined CORDIC processor was designed with the help of Verilog HDL and synthesized. An exhaustive test bench was also written to simulate the functionality of the processor. The CORDIC processor worked as per the expectations giving us exact values for high inputs. For future advancements a bit serial iterative CORDIC architecture can be adopted. In bit serial iterative CORDIC architecture the interconnections can be made minimal and the logic between resistors would be simple making it even more efficient.

#### REFERENCES

- [1] Jose.M Canet, Felip Vicedo, Vicenc Almenar, Javier alls, "FPGA Implementation of an IF transceiver for OFDM-based WLAN", IEEE Workshop on Signal Processing Systems, Vol.2, pp. 227-232, October, 2004.
- [2] Almenar V, Abedi S and Rahim Tafazolli, "Synchronization Techniques for HIPERLAN/2", 54th IEEE Vehicular Technology Conference, Vol. 2, pp. 762-766, 2001(Fall).
- [3] David Gesbert, Mansoor Shafi, Dashan Shiv, Peter J Smith, and Aymon Naguib, "From Theory to Practice: An Overview of MIMO Space-Time coded Wireless Systems", IEEE Journal on selected areas in Communications, Vol.21, No.3, April 2003.
- [4] Siavash M.Alamouti, "A Simple Transmit diversity Technique for Wireless Communications", IEEE Journal on selected areas in Communications, Vol.16, No.8, October 1998.
- [5] Kaiser, Wilzeck, Berentsen and Rupp, "Prototyping for MIMO Systems: An Overview", International Conference EUSIPCO, Austria, pp.681-688, Sept 6-10, 2004.
- [6] John G. Proakis and Dimitis G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications*, Prentice-Hall India Publications, Third Edition, ISBN-81-203-1129-9.
- [7] Jack Volder, "The CORDIC Trigonometric Computing Technique", IRE Transactions on Electronic Computing, Vol EC-8, pp.330-334, September 1959.
- [8] Ray Andraka, "A survey of CORDIC algorithms for FPGA based computers", ACM 0-89791-978-5/98/01, 1998.
- [9] U. Meyer-Baese, "Digital Signal Processing with Field Programmable Gate Arrays", Springer- Verlag, ISBN 81-8128-045-8, 2003.