European Journal of Scientific Research
ISSN 1450-216X Vol.25 No.2 (2009), pp.200-213

http://www.eurojournals.com/ejsr.htm

Design and Implementation of Autocorrelation and CORDIC Algorithm for OFDM Based WLAN

Sudhakar Reddy.P
Associate Professor, Dept. of ECE, SKIT, Srikalahasthi, Andhra Pradesh, India
E-mail: psr_vlsi_dsp@rediffmail.com
Tel: +91-9705325002

Ramachandra Reddy.G
Sr.Member IEEE Professor, Dept. of ECE, Sri Venkateswara University
Tirupathi, Andhra Pradesh, India
Tel: +91-9705325002

Abstract

This paper deals with design and implementation of Autocorrelator and CORDIC algorithm for OFDM based WLAN on FPGA. The 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. A fast pipelined CORDIC architecture and autocorrelator is designed and implemented on FPGA. HDL and test bench is developed to simulate and verify the functionality of both the modules. The design is implemented on Spartan-3 100K Xilinx FPGA; the design consumes 180 slices out of 960 slices. The design is also taken through ASIC design flow using 130nm technology, the design is synthesized to meet the setup and hold time requirements. The design is optimized for area and power requirements. The autocorrelator design raked up 5684 cells taking up 13969.5μm² of area for the cells and total area of 16609.6 μm² consuming 78.7μW. The CORDIC design raked up 2288 μm² total cell area for 552 numbers of cells. The total area comprising of both the modules is 2558 μm² consuming 12.05μW.

Keywords: Autocorrelation, CORDIC, ASIC, OFDM, WLAN, FPGA

1. Introduction

As we all know that wireless LAN or WLAN is a wireless local area network where two or more systems are linked to each other without using wires. To enable communication between the systems or devices, WLAN takes help of spread spectrum technique or OFDM modulation technique based on radio waves. The advantage one acquires with WLAN is the mobility to move around a certain coverage area and still be connected to the network.

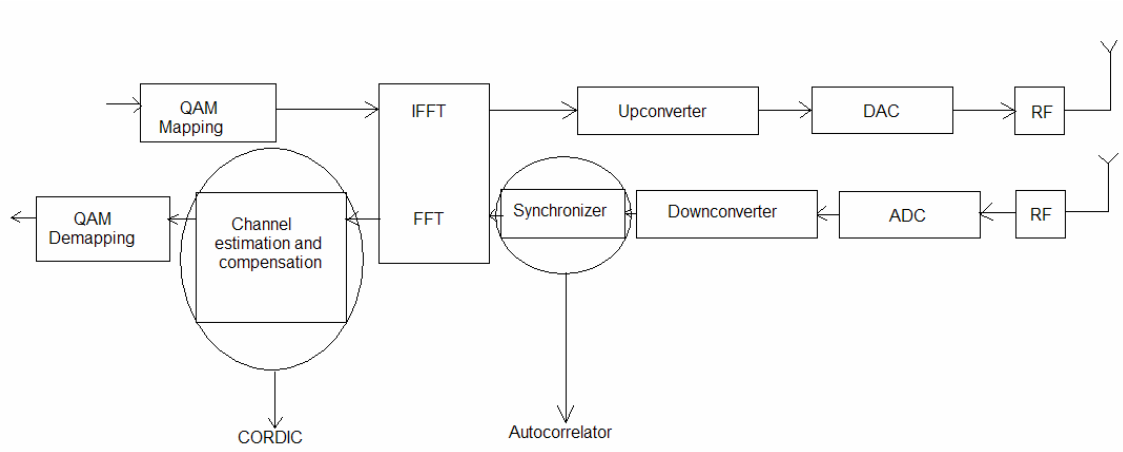
With the advent of WLAN in the late 70's, advancements have been made progressively with time by various research organizations keeping in mind by the protocol standards. The WLAN hardware originally was expensive from the time of its advent that its primary use was that of an alternative technology to wired LAN places where cabling was difficult or a mere impossibility. By the

end of 1990's, the WLAN protocols were replaced by standards namely IEEE 802.11 (Wi-Fi) and hi-performance LAN i.e.HIPERLAN [1].

In today's world, the popularity of WLAN has reached a pinnacle in day-to-day lives of people due to the cost efficient hardware of WLAN and the ease of integration with other networks and network components. The other benefits of WLAN include convenience as one can access the network resources from any convenient location within the coverage area; mobility as users can access the internet from almost any location outside such as coffee shops; deployment, as the initial set up for WLAN includes only an access point and no cables which acquire huge amount of area and space; and last but no the least is the cost of WLAN.

Though the advantages of WLAN may take up a higher number when compared to the disadvantages, one cannot rule out these disadvantages. The major disadvantages one experiences with WLAN include security, range of network, reliability and speed of network. OFDM is one such modulation technique that takes into account with issues concerning the reliability and speed of network.

Figure 1: Hiperlan/2 Transceiver [1]



2. 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. Keeping in mind the integral area under one period of a sine or a cosine wave is zero and that the carriers are all sine/cosine wave, we can conclude that when we multiply a sinusoid of frequency m by a sinusoid of frequency n, the area under the product constitutes to zero. Therefore, in general for all integers' n and m, sinmx, sinnx, cosmx and cosnx are all orthogonal to each other. These frequencies are called harmonics. This property is the key to understanding OFDM [2]. 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 the system handles the multipath interference at

the receiver end. Fig 1 shows the block diagram of transmitter and receiver section of OFDM system. The blocks circled in Fig.1 are the area of interest of this work, details of which is provided in the next section.

3. Autocorrelation

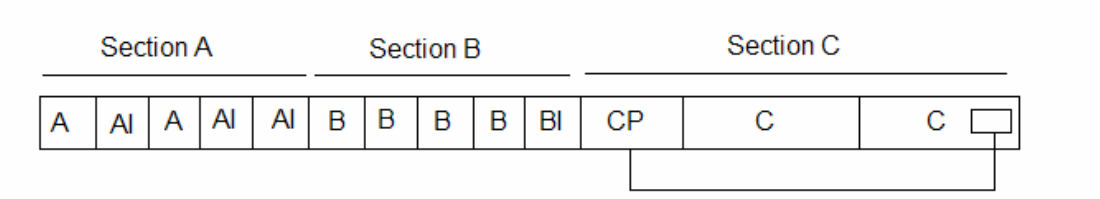
According to John G. Proakis and Dimitris G. Manolakis [6], autocorrelation is a measure of how well a signal matches a time shifted version of itself, as a function of the amount of time shift. More precisely autocorrelation is the cross-correlation of a signal with itself. According to [6], autocorrelation 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. The autocorrelation equation is given by

$$r(J) = 1/N \sum_{n=0}^{N-1} x(n) \cdot x(n+j)$$

In WLAN systems synchronization and channel compensation depend on using a preamble as shown in Fig.2. So, in Hiperlan/2, the access point transmits a broadcast preamble that is used by mobile terminals (MT) to perform frame synchronization, time synchronization, frequency synchronization and channel estimation. Therefore the synchronization phase in an MT can be divided into three parts: detection of the broadcast preamble, time synchronization, which consists of estimating the sample when the OFDM symbol starts and carrier frequency offset estimation.

The structure of the Hiperlan/2 preamble is shown in Fig. 2. It is evident from the preamble that there exists three sections namely A, B and C. Section A is used for frame detection, Section B is intended for time synchronization and coarse carrier frequency offset (CFO) estimation, and Section C is used for fine CFO estimation and channel estimation.

Figure 2: Hiperlan/2 Preamble Structure [3].



Therefore, the autocorrelator or autocorrelation scheme helps us in frame synchronization, time synchronization, frequency synchronization and carrier frequency offset estimation.

Frame detection is the first phase of the whole synchronization process, and the most important, since only after a correct frame is detected, an MT can become a member of the wireless network. Here autocorrelation of the received signal comes in handy and the top level block diagram is shown later as the report progresses. The received signal is correlated with a delayed version of itself to find a peak in sections A, B and C of the preamble.

Therefore we can conclude that the autocorrelator takes care of frame synchronization, time synchronization, frequency synchronization and carrier frequency offset estimation by cross-correlating the received signal with a delayed version of itself. The implementation of the autocorrelator in this design requires one eight bit multiplier, an adder and a 128-by-8 bit wide RAM.

4. Coordinate Rotation Digital Computer (CORDIC)

CORDIC stands for COordinate Rotation DIgital Computer. 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 adds. The algorithm, credited to Volder [2] is derived from the general (Givens) rotation transform as shown in equations (i) and (ii).

The basic equations required to implement CORDIC are:

$$X(i+1) = X(i)\text{Cos}\varphi - Y(i)\text{Sin}\varphi. \quad \text{(i)}$$

$$Y(i+1) = Y(i)\text{Cos}\varphi + X(i)\text{Sin}\varphi \quad \text{(ii)}$$

$$X(i+1) = \text{Cos}\varphi\,(X(i) - Y(i)\text{Tan}\varphi) \quad \text{(iii)}$$

$$Y(i+1) = \text{Cos}\varphi\,(Y(i) + X(i)\text{Tan}\varphi) \quad \text{(iv)}$$

If the rotation angles are restricted so that $\text{Tan}\varphi = \pm 2\exp(-i)$, the multiplication by the tangent term is reduced to simple shift operation. Equations (iii) and (iv) can now be expressed for each iteration as:

$$X(i+1) = K_i\,[\,X(i) - Y(i)d_i.\,2\exp(-i)] \quad \text{(v)}$$

$$Y(i+1) = K_i\,[\,Y(i) - X(i)d_i.\,2\exp(-i)] \quad \text{(vi)}$$

Where $K_i = \text{Cos}\,(\,\text{Tan}^{-1}\,2.\exp(-i)) = 0.60725$ and $d_i = \pm 1$

The third parameter $Z(i+1) = Z(i) - d_i\varphi$

So to reach an expected angle, a series of iterations are required to be performed and in this design the number of iterations are i = 5 and in every iteration the new values of x, y and z depend upon the previous values of the same. According to [4][5], the CORDIC processor can be configured to work as circular rotation and vectoring mode, and 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. Therefore we can conclude that the application of CORDIC algorithm in OFDM is vast, important and a vital one in the channel estimation stage, frequency compensation and in estimating frequency offset.

5. Design

In this section, block diagram and designs of top level OFDM, Autocorrelator and CORDIC are explained.

5.1. OFDM Top Level Design

Fig.3 shows the block diagram of the top level of OFDM [5]. We can see from the figure that the main components of OFDM are transmitter and the receiver.

The main building blocks of the transmitter section are the quadrature phase shift keying (QPSK) modulator, a serial to parallel converter block, an inverse fast Fourier transform (IFFT) generator, a cyclic prefix block where 4 bits are added to the signal to avoid inter symbol interference (ISI) and inter carrier interference (ICI) and a parallel to serial converter block. The receiver section also has the same blocks but in reverse order and in place of a QPSK modulator and IFFT we have a QPSK demodulator and an FFT generator.

The number of data bits transmitted was 64. The serial to parallel converter sends 4 blocks of data each of 16 bits wide to the IFFT generator. 4 bit data as cyclic prefix is appended to the output of the IFFT generator. Then the data is parallel to serial converted and then transmitted in the channel. At the receiver section, the received data is serial to parallel converted before getting into the cyclic prefix

block. Here the 4 bit cyclic prefix data is removed from the signal before sending to the FFT generator block. The signal is processed in the FFT generator block and the output of this block which is parallel is sent to the parallel to serial converter block. Before the data is sent to the QPSK demodulator block, the signal is first parallel to serial converted. And finally, one can see the output at the receiver section after the signal is QPSK demodulated. The architectures of cordic and autocorrelator are implemented on FPGA [9][10] for further improvements.

5.2. Autocorrelator Design

An autocorrelator which can take in input of length 128 samples each of 8 bits wide is designed. Fig. 4 shows the block diagram of the autocorrelator design. The design is carried out in Verilog HDL and simulated using Model Sim.

Figure 3: Top Level Block Diagram of OFDM

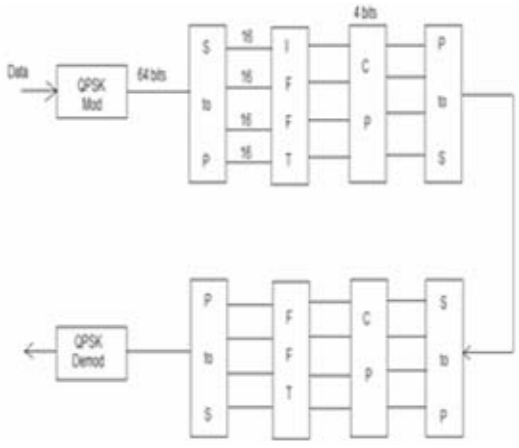


Figure 4: Block Diagram of Autocorrelator.

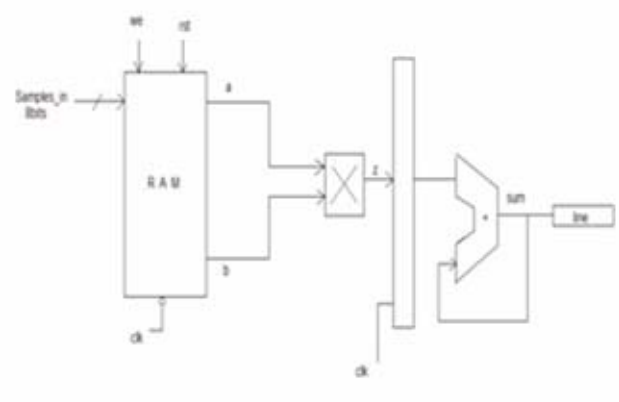
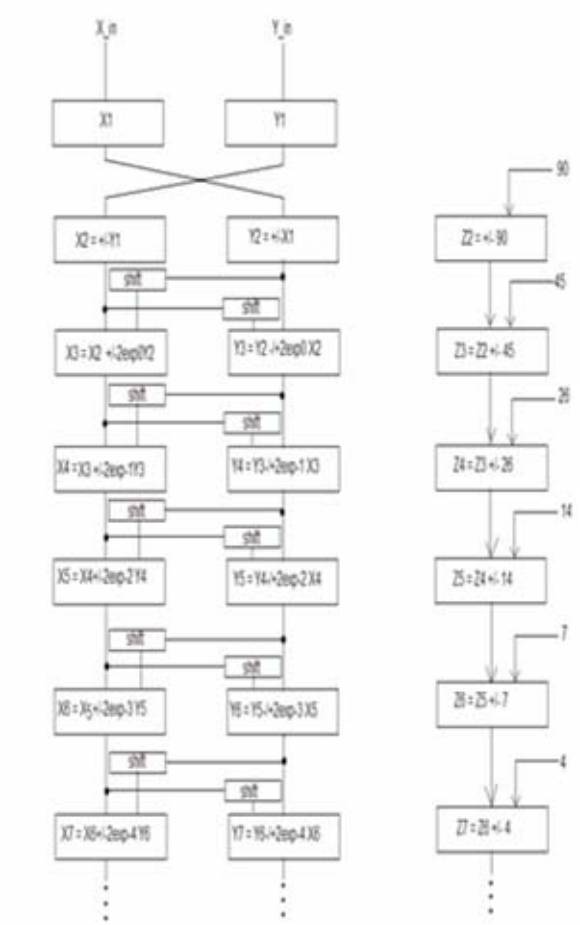


Figure 5: Fast CORDIC Pipe line [8]



The autocorrelator is designed in such a way that ‘N’ number of inputs can be fed in and the output port named as ‘line’ displays the auto correlated version of it. In this design a RAM was 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’ 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 terms can be called as the shift operation. Here the 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. To save the number of IOB’s in the Xilinx FPGA, the design has been modified to read the data serially, and this reduces the number of IOB’s but affects the speed.

5.3. Cordic Design

The Fig.5. Shows a fully pipelined fast CORDIC architecture adopted in this design.

Fig.5 is that of a pipelined CORDIC processor which employs iterative CORDIC architecture. The iterative CORDIC architecture can be obtained simply by duplicating each of the three difference equations shown in equations (i-vi). The decision function, d_i, is driven by 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 placed back in the registers. The shifters are modified on each iteration to cause the desired shift for the iteration.

The algorithm checks for the value of 'y' register. Initially if x_in>0, then x0 will be loaded with x_in, y0 will be loaded with y_in and no rotations will take place.

Now the algorithm checks for the condition y_in>0, if so, x0 will be loaded with y_in, y0 will be loaded with the negated (shifted) value of x_in (y_in = -x_in) and z0 will be rotated with 90 degrees. If the condition y_in>0 fails i.e. y_in<0, then x0 = -y_in, y0 = x_in, and z0 = -90 degrees.

Now the algorithm checks for the condition y0 > 0 and if it finds so, then x1=x0+y0, y1 = y0 – x0, and z1 = z0+45degrees else x1 = x0-y0, y1 = y0+x0 and z0=z0-45degrees. Further, the algorithm checks for y1>0, if found true then x2=x1+y1/2 and y2=y1-x1/2 and z2=z1+26. The divide operation is performed with the help of right shift operation. For divide by 2 operations, the MSB is retained and the remaining 7 bits are right shifted. For the next condition when a divide by 4 operations is required, the 2 MSB's are retained and the remaining 6 bits are right shifted.

As the iterations increase, we get an exact value. Verilog HDL is written to realize a fast pipelined CORDIC processor and an exhaustive test bench is also written to simulate the functionality with the help of Model Sim.

6. Results and Discussions

In this section, discussion of results of OFDM top level, Autocorrelator and CORDIC Algorithm is presented.

6.1. Simulation Results of OFDM Spectra

A MATLAB simulation is performed to obtain the spectrum of OFDM. Fig.6. shows the simulation result of the OFDM spectrum obtained through MATLAB simulation.

For the simulation, the number of data bits transmitted is 1024 and data symbol per frame to IFFT is 64. The Guard Interval (GI) length is 2. From the simulation results it is concluded that by carefully selecting the carrier spacing, the OFDM signal spectrum is made flat and the orthogonality among the sub-channels is guaranteed. The guard interval is added to maintain the orthogonality of the sub-carriers and to avoid inter symbol interference (ISI) and inter carrier interference (ICI).

Since the spectra of the OFDM signal is not strictly band limited, linear distortion such as multipath cause each sub-channel to spread energy into the adjacent channels and consequently cause ISI. A simple solution is to increase the symbol duration or the number of carriers so that distortion becomes insignificant.

6.2. Simulation Results of OFDM Top Level

Random data is taken and transmitted; from the screenshots we can see that the transmitted data (original data) matches the received data.

Fig.13 shows the data coming out of the QPSK demodulator in the receiver section. We see that the Fig.7 named as original data matches with Fig.13 clearly demonstrating that the transmitted data after being appended with cyclic prefix is received at the receiver side after removing the cyclic prefix.

Fig.7 shows the screenshot of the simulation results of original data to the OFDM top level block. This data is fed as input to the QPSK Modulator block as shown in Fig.8

After the signal is QPSK modulated, it is fed as input to the IFFT block where the IFFT operation is performed. The simulation result for this block is shown in Fig.9. Fig. 10 and Fig.11 show

the signal characteristics at the transmitter and the receiver respectively. From the figures we can clearly see the presence of noise in the channel. Due to this noise, the transmitted signal does not match the received signal. The received signal is corrupted with the noise present in the channel as clearly seen in Fig.11

Fig.12 shows the simulation result of the FFT block. Fast Fourier Transform is performed on all the four sub-carriers and the signal characteristics are shown in Fig.12.

Fig13 is the simulation result of the QPSK demodulator which is also the output signal. This signal clearly matches the original signal demonstrating that the transmitted data is clearly received by the receiver, thus proving the problem statement.

Figure 6: OFDM Spectrum

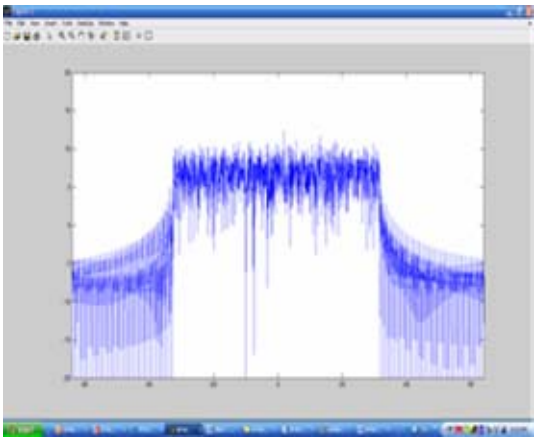


Figure 7: Original Data

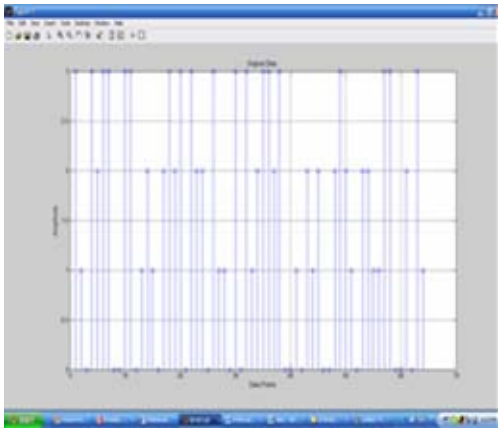


Figure 8: QPSK Modulation output

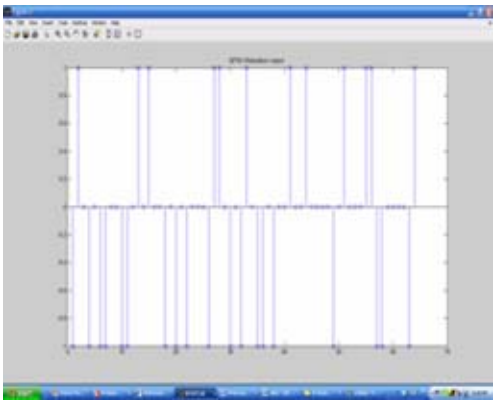


Figure 9: IFFT on all the sub carriers

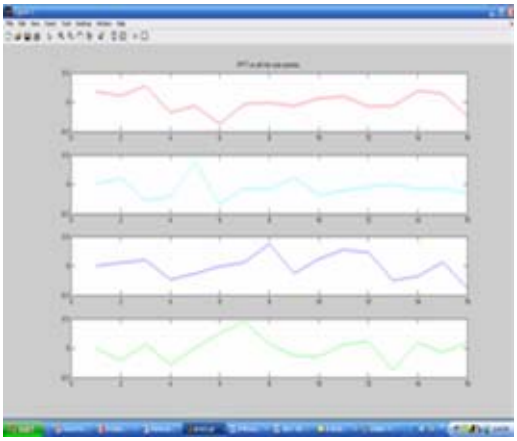


Figure 10: OFDM Signal at the Transmitter

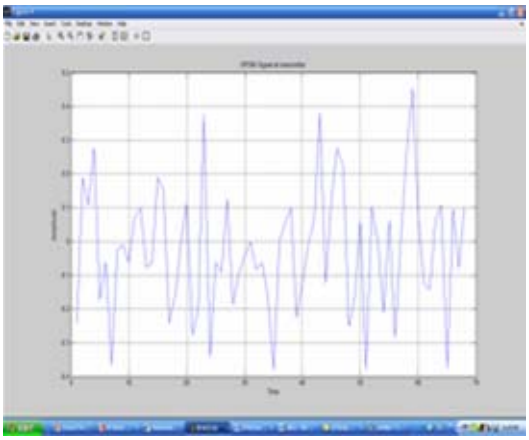


Figure 11: OFDM Signal at the Receiver

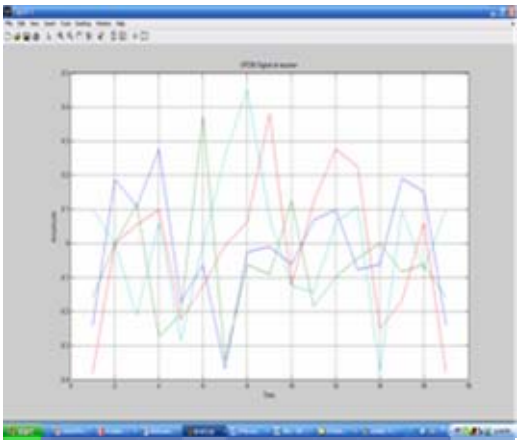


Figure 12: FFT on all the 4 sub-carriers

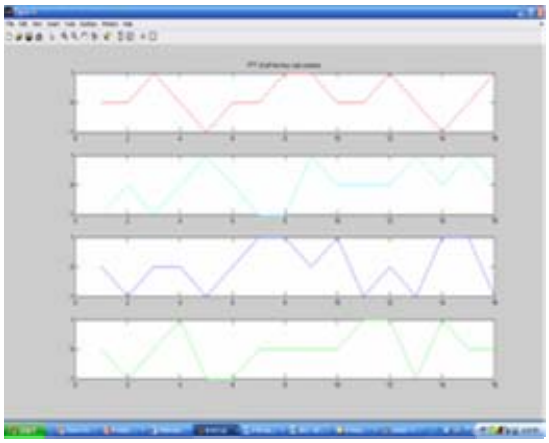


Figure 13: QPSK Demodulation output

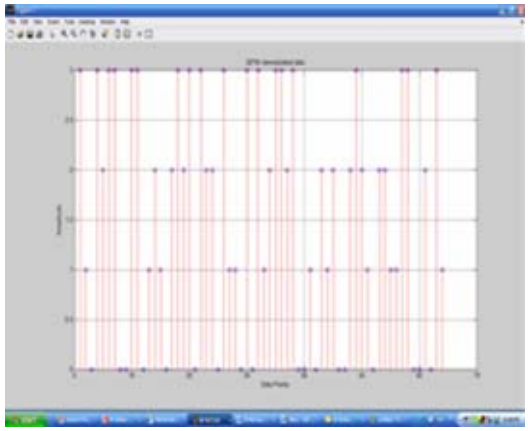


Figure 14: MATLAB Simulation of the 128 input samples

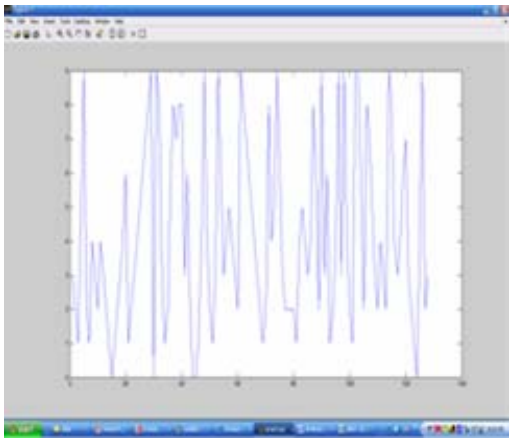
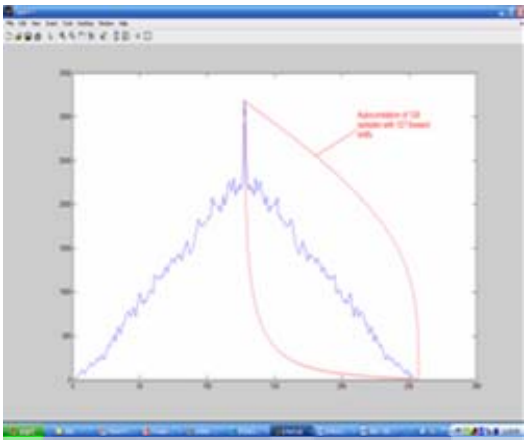


Figure 15: Auto correlated output of 128 input samples

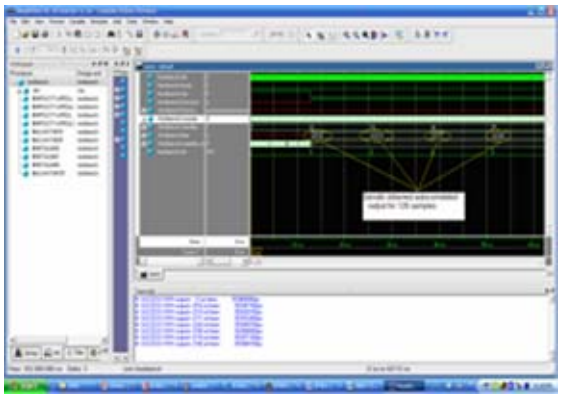


6.3. Simulation Results of Autocorrelator

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 which also is very convenient. The design works at a frequency of 46.252MHz. If a pipelined architecture was adopted, then the operating frequency would have been close to 72 MHz. This can be recommended for future advancements. 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. 14 and 15 shows the MATLAB simulation of the inputs and the outputs.

Figure 16: Behavioral simulation of Autocorrelator using test bench



From Fig.16, we can notice the outputs are obtained serially from a single output port named as 'line'. These values were compared with the MATLAB Simulated one and both matched. An exhaustive test bench was written to simulate the design. In the Fig.16 we can also see the reset pin which becomes low after 10ns and the write enable pin (we) remains high until the all the inputs are fed into the RAM.

6.4. Simulation Results of CORDIC

The simulation results for CORDIC block are displayed in this section. 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 MHz. Fig.17 shows the simulation result of the pipelined CORDIC architecture run with the help of an exhaustive test bench.

Figure 17: Simulation Result of Pipelined Fast CORDIC

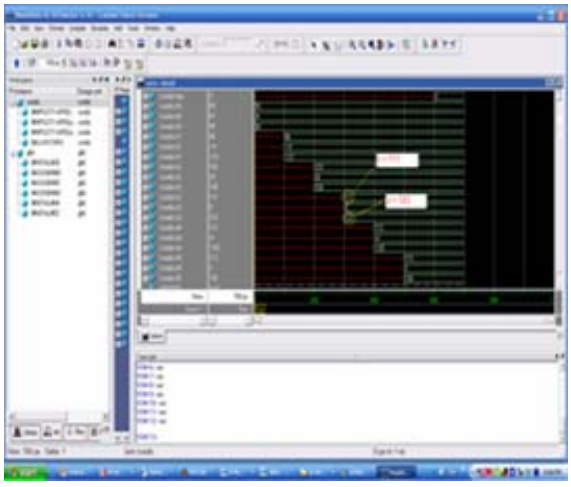


Table 1: Hardware resources

	Slices	Mults	Bonded IO's	Frequency
AutoCorrelator	68	1	34	46.252MHZ
CORDIC	112	0	41	219.443MHZ

Fig. 17 shows the simulation result when X0 = 215 = -41 mod 256, andY0 = 55. The radius 'r' = 1.618 sqrt($X0^2 + Y0^2$) = 111. The accumulated angle in degrees arctan (Y0/X0) = 123 degrees. The operating frequency of the design is 219.44 MHz.

7. Fpga/Asic Implementation

The design of both Autocorrelator and CORDIC was modeled using Verilog HDL and synthesized to implement on a Spartan -3 Xilinx FPGA. The Table 1 summarizes the resources that are required by each block.

The architectures of both Autocorrelator and CORDIC is 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 13969.5μm^2 of area for the cells and total area of 16609.6 μm^2 consuming 78.7μW. The CORDIC design raked up 2288 μm^2 total cell area for 552 numbers of cells. The total area comprising 2558 μm^2 consuming 12.05μW. The SDC files created during DC flow are taken as inputs to perform Physical Design flow. Prior to Physical Design, Prime time tool is used in performing timing analysis to check for setup and hold time.

8. Conclusions

An OFDM Top level block is simulated using MATLAB and the transmitted signal is received at the receiver. The signal at the transmitter is QPSK modulated, and then serial to parallel converted before feeding the signal to an IFFT generator. A 4 bit cyclic prefix is appended and the signal is again parallel to serial converted. At the receiver end the signal is converted from serial to parallel and cyclic prefix is removed before feeding the signal to the FFT generator. The output is observed only when the signal is processed by the QPSK demodulator. Though the signal at the receiver section is added with noise, the original transmitted signal is obtained at the receiver. An autocorrelator is designed to perform the autocorrelator of 128 samples each of 8 bits wide. The building blocks for the scheme are a 128x8 RAM, a multiplier, an accumulator and a counter. The Matlab simulations are performed prior to the Verilog HDL coding to check if the functionality is achieved. A fully pipelined CORDIC processor is designed with the help of Verilog HDL and synthesized. An exhaustive test bench is also written to simulate the functionality of the processor. The CORDIC processor worked as per the expectations giving us exact values for high inputs. Recommendation work for the future would be that of to design a pipelined architecture that can support both autocorrelator and cordic. 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 Valls, "*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,Aprial 2003.

[4] Siavash M.Alamouti, "*A Simple Transmit diversity Technique for Wire less 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 Dimities 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] Meyer-Baese.U, "*Digital Signal Processing with Field Programmable Gate Arrays"*, Springer-Verlag, ISBN 81-8128-045-8, 2003.

[10] Iqbal Hussain, Selva Kumar and Cyril Prasanna Raj P, "*Design and FPGA implementation of auto correlator and CORDIC algorithm*", SASTECH Journal, MSRSAS, Vol 6, March, 2008.