

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/346676065>

VLSI Implementation of OFDM Transmitter chain for 4G Communication

Article in *International Journal of Innovative Research and Growth* · May 2018

CITATIONS

0

READS

29

2 authors, including:



[Jawaaz Ahmad](#)

Islamic University of Science and Technology

23 PUBLICATIONS 5 CITATIONS

SEE PROFILE

VLSI Implementation of OFDM Transmitter chain for 4G Communication

Jawaaz Ahmad¹, Lakshmisha S K²

¹Post Graduate Student, Department of ECE, Alpha College of Engineering, Bangalore, Karnataka

²Professor, Department of ECE, Alpha College of Engineering, Bangalore, Karnataka

Abstract- Demand for broadband access is increasing at a quick rate, and at the same time, is not limited to areas that already have an existing high quality infrastructure. For instance, developing countries and rural areas may not have the existing telecom infrastructure or the existing connections, typically over copper, to meet the requirements of Digital Subscriber Line (DSL) technology. Furthermore, it is expected that users will require more bandwidth on the move. While current technologies can meet this bandwidth demand, the useful range is limited. This limitation opens up opportunities for technologies such as Orthogonal Frequency Division Multiplexing.

Index Terms- LTE, 4G, VLSI, OFDM, VLSI, Transmitter Chain.

I. INTRODUCTION

OFDM is a digital modulation technique; therefore an introduction to digital communication systems is being provided. A digital communication system involves the transmission of information in digital form from one point to another point as shown in Figure 1.1.

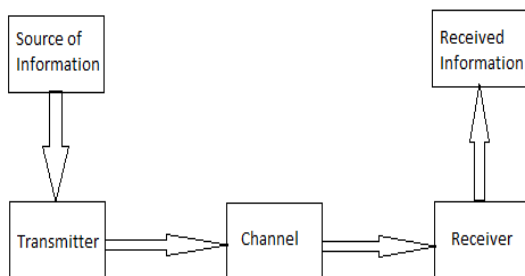


Figure 1.1: A typical Digital Communication Systems

The three basic elements in a communication system are transmitter, channel and receiver. The source of information is the messages that are to be transmitted to the other end in the receiver. A transmitter can

consist of source encoder, channel coder and modulation. Source encoder provides an efficient representation of the information through which the resources are conserved. A channel coder may include error detection and correction code. A modulation process then converts the base band signal into band pass signal before transmission.

During transmission, the signal experiences impairment which attenuates the signals amplitude and distort signals phase. Also, the signals transmitting through a channel also impaired by noise, which is assumed to be Gaussian distributed component.

At the receiving end, the reversed order of the steps taken in the transmitter is performed. Ideally, the same information must be decoded at the receiving end.

Orthogonal Frequency Division Multiplexing:

Orthogonal Frequency Division Multiplexing (OFDM) is a digital multi-carrier modulation scheme that extends the concept of single subcarrier modulation by using multiple subcarriers within the same single channel. Rather than transmit a high-rate stream of data with a single subcarrier, OFDM makes use of a large number of closely spaced orthogonal subcarriers that are transmitted in parallel. Each subcarrier is modulated with a conventional digital modulation scheme (such as QPSK, QAM etc.) at low symbol rate. However, the combination of many subcarriers enables data rates similar to conventional single-carrier modulation schemes within equivalent bandwidths.

OFDM is a combination of modulation and multiplexing. Multiplexing generally refers to independent signals, those produced by different sources. So it is a question of how to share the spectrum with these users. In OFDM the question of multiplexing is applied to independent signals but

these independent signals are a sub-set of the one main signal.

In OFDM the signal itself is first split into independent channels, modulated by data and then re-multiplexed to create the OFDM carrier. OFDM is based on the well-known technique of Frequency Division Multiplexing (FDM). In FDM different streams of information are mapped onto separate parallel frequency channels. Each FDM channel is separated from the others by a frequency guard band to reduce interference between adjacent channels.

OFDM is a special case of Frequency Division Multiplex (FDM). As an analogy, a FDM channel is like water flow out of a faucet, in contrast the OFDM signal is like a shower. In a faucet all water comes in one big stream and cannot be sub-divided. OFDM shower is made up of a lot of little streams.



Figure 1.2: Comparing FDM and OFDM

Think about what the advantage might be of one over the other? One obvious one is that if we put our thumb over the faucet hole, we can stop the water flow but we cannot do the same for the shower. So although both do the same thing, they respond differently to interference. Another way to see this intuitively is to use the analogy of making a shipment via a truck.

We have two options, one hire a big truck or a bunch of smaller ones. Both methods carry the exact same amount of data. But in case of an accident, only 1/4 of data on the OFDM trucking will suffer as shown in the Fig 1.2



Figure 1.3: FDM and OFDM Analogy

These four smaller trucks when seen as signals are called the sub-carriers in an OFDM system and they must be orthogonal for this idea to work. The independent sub-channels can be multiplexed by

frequency division multiplexing (FDM), called multi-carrier transmission.

The OFDM scheme differs from traditional FDM in the following interrelated ways:

1. Multiple carriers (called subcarriers) carry the information stream,
2. The subcarriers are orthogonal to each other, and
3. A guard interval is added to each symbol to minimize the channel delay spread and inter symbol interference.

The figure 1.3 illustrates the main concepts of an OFDM signal and the inter-relationship between the frequency and time domains. In the frequency domain, multiple adjacent tones or subcarriers are each independently modulated with complex data. An Inverse FFT transform is performed on the frequency-domain subcarriers to produce the OFDM symbol in the time domain. Then in the time domain, guard intervals are inserted between each of the symbols to prevent inter-symbol interference at the receiver caused by multi-path delay spread in the radio channel. Multiple symbols can be concatenated to create the final OFDM burst signal. At the receiver an FFT is performed on the OFDM symbols to recover the original data bits.

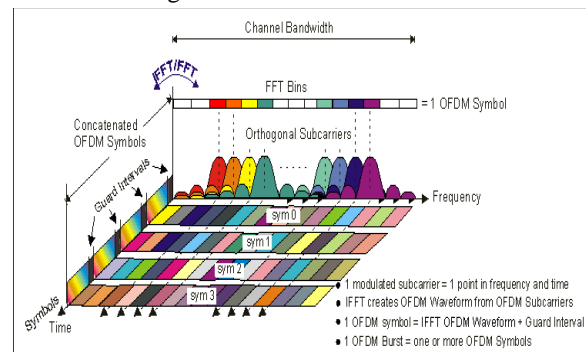


Figure 1.4: Time representation of OFDM

II. UNDERSTANDING ORTHOGONALITY

The OFDM signal can be described as a set of closely spaced FDM subcarriers. In the frequency domain, each transmitted subcarrier results in a sinc function spectrum with side lobes that produce overlapping spectra between subcarriers, see "OFDM Signal Frequency Spectra" Figure 1.4. This results in subcarrier interference except at orthogonally spaced frequencies. At orthogonal frequencies, the individual peaks of subcarriers all line up with the nulls of the other subcarriers. This overlap of spectral energy

does not interfere with the system’s ability to recover the original signal. The receiver multiplies (i.e., correlates) the incoming signal by the known set of sinusoids to recover the original set of bits sent.

The use of orthogonal subcarriers allows more subcarriers per bandwidth resulting in an increase in spectral efficiency. In a perfect OFDM signal, orthogonality prevents interference between overlapping carriers. In FDM systems, any overlap in the spectrum of adjacent signals will result in interference. In OFDM systems, the subcarriers will interfere with each other only if there is a loss of orthogonality. For example, frequency error will cause the subcarrier frequencies to shift so that the spectral nulls will no longer be aligned resulting in inter-subcarrier-interference.

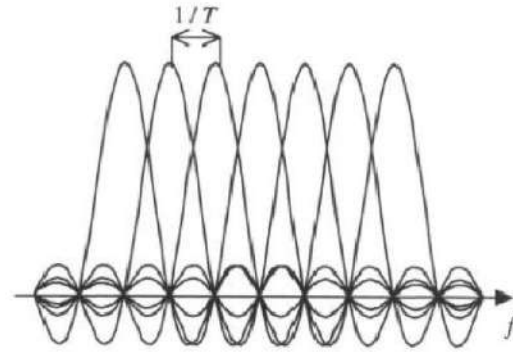


Figure 1.7: OFDM signal representation in frequency domain.

III. A TYPICAL OFDM SYSTEM

Figure 1.3 shows a detailed OFDM communications system. Each block is briefly defined below:

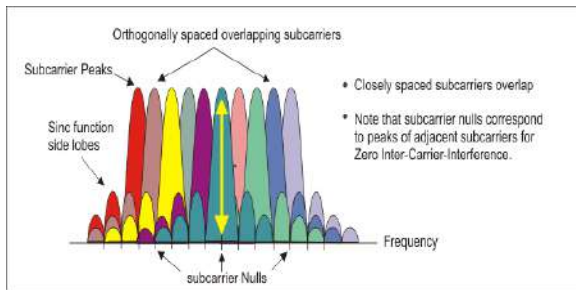


Figure 1.5: OFDM signal frequency spectrum.

OFDM is a special case of FDM:

Let’s first look at what a Frequency Division Multiplexing FDM is? If we have a bandwidth that goes from frequency a to b, we can subdivide this into a frequency space of five equal spaces (or five carriers say a, b, c, d, e), there is no relationship between the subcarriers a, b, c, d and e can anything within the given bandwidth. In frequency space the modulated carriers would look as shown in the Fig 1.5.

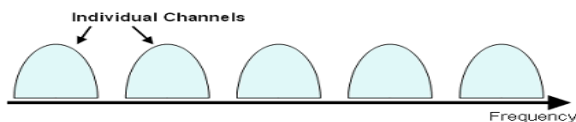


Figure 1.6: FDM signal representation in frequency domain

If the carriers are harmonics, say $(b=2a, c=3a, d=4a, e=5a)$, integral multiple of fundamental component a) then they become orthogonal. This is a special case of FDM, which is called OFDM as implied by the word – ‘orthogonal’ in OFDM.

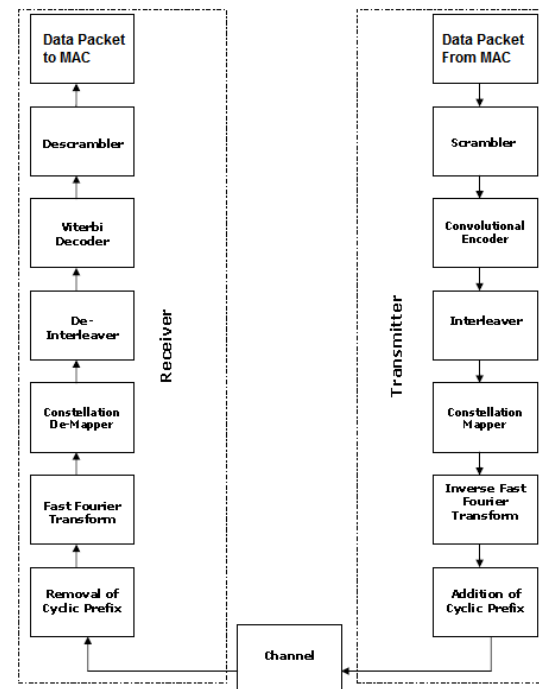


Figure 2.1: Complete OFDM System

2.1 Scrambler / De-Scrambler:

Data bits are given to the transmitter as inputs. These bits pass through a scrambler that randomizes the bit sequence. This is done in order to make the input sequence more disperse so that the dependence of input signal’s power spectrum on the actual transmitted data can be eliminated. At the receiver end descrambling is the last step. De-scrambler simply recovers original data bits from the scrambled bits.

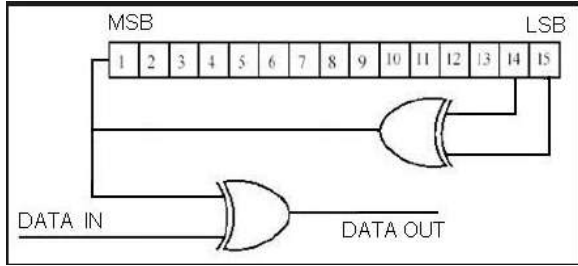


Figure 2.2: Scrambler

2.2 Interleaver / De-Interleaver:

Interleaving is done to protect the data from burst errors during transmission. Conceptually, the incoming bit stream is re-arranged so that adjacent bits are no more adjacent to each other. The data is broken into blocks and the bits within a block are re-arranged [5]. Talking in terms of OFDM, the bits within an OFDM symbol are re-arranged in such a fashion so that adjacent bits are placed on non-adjacent sub-carriers. As far as De-Interleaving is concerned, it again rearranges the bits into original form during reception.

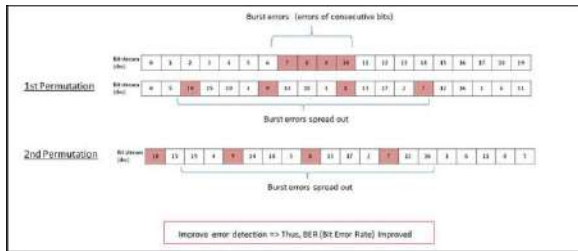


Figure 2.3: Interleaver/ De-Interleaver

2.3 Convolution code:

Convolution encoding is a method of adding redundancy to a data stream in a controlled manner to give the destination the ability to correct bit errors without asking the source to retransmit. A convolution code is used to correct errors in the receiver hence they are called as Forward Error Codes(FEC) A binary convolutional code is denoted by a three-tuple (n, k, m) . 'n' output bits are generated whenever k input bits are received. The current n outputs are linear combinations of the present 'k' input bits and the previous $m \times k$ input bits. 'm' designates the number of previous k-bit input blocks that must be memorized in the encoder. 'm' is called the memory order of the convolutional code.

2.3.1 Convolution encoder parameters:

Convolution Encoder adds redundant bits to the input data stream to correct or remove the random noise.

A binary convolutional encoder is conveniently structured as a mechanism of shift registers and modulo-2 adders, where the output bits are modular-2 additions of selective shift register contents and present input bits. 'n' in the three-tuple notation is exactly the number of output sequences in the encoder. 'k' is the number of input sequences (and hence, the encoder consists of k shift registers). 'm' is the maximum length of the 'k' shift registers. The definition of constraint length of a convolutional code is defined in several ways. The most popular one is $m + 1$

A convolutional code is a type of Forward Error Correction (FEC) code that is specified by $CC(m, n, k)$, in which each in-bit information symbol to be encoded is transformed into an n-bit symbol, where m/n is the code rate ($n > m$) and the transformation is a function of the last k information symbols, where k is the constraint length of the code .

2.2.2 Encoder for binary convolutional code:

The convolution encoder maps a continuous information bit stream into a continuous bit stream of encoder output. The convolution encoder is a finite state machine, which is a machine having memory of past inputs and also having a finite number of different states. The encoder used in this design is shown in the Figure 3.2. It is a 1/2 rate encoder i.e., two bit output is generated for one bit input.

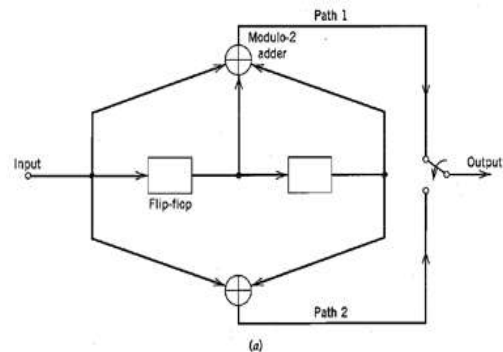


Figure 2.4: Convolution encoder

Because of the memory associated with a convolutional encoder, we must have a convenient means of determining the specific output bit sequence

generated for a given input sequence. The present state, next state and output table for the convolutional encoder used is given in the table 2.1

PRESENT STATE	INPUT	NEXT STATE	OUTPUT
00(0)	0	00	00(0)
00(0)	1	10	11(3)
01(1)	0	00	11(3)
01(1)	1	10	00(0)
10(2)	0	01	10(2)
10(2)	1	11	01(1)
11(3)	0	01	01(1)
11(3)	1	11	10(2)

Table 2.1: Present state, next state and output table

2.4 Modulation and Demodulation:

Fundamental to all wireless communications is modulation, the process of impressing the data to be transmitted on the carrier. Most wireless transmissions today are digital, and with the limited spectrum available, the type of modulation is more critical than it has ever been. The main goal of modulation today is to squeeze as much data into the least amount of spectrum possible. That objective, known as spectral efficiency, measures how quickly data can be transmitted in an assigned bandwidth. The unit of measurement is bits per second per Hz (b/s/Hz). Multiple techniques have emerged to achieve and improve spectral efficiency. In digital modulation techniques a set of basis functions are chosen for a particular modulation scheme. Generally the basis functions are orthogonal to each other. Basis functions can be derived using Gram Schmidt orthogonalization procedure.

In Quadrature Phase Shift Keying (QPSK) two sinusoids (sin and cos) are taken as basis functions for modulation. Modulation is achieved by varying the phase of the basis functions depending on the message symbols. In QPSK, modulation is symbol based, where one symbol contains 2 bits. The constellation diagram shown in the Figure 2.5 outlines QPSK modulation technique.

If the symbol is present in the first quadrant then it is symbol '0', if present in second quadrant then symbol '1', if present in third quadrant then symbol '3', else it is symbol '2'. They are represented using gray code.

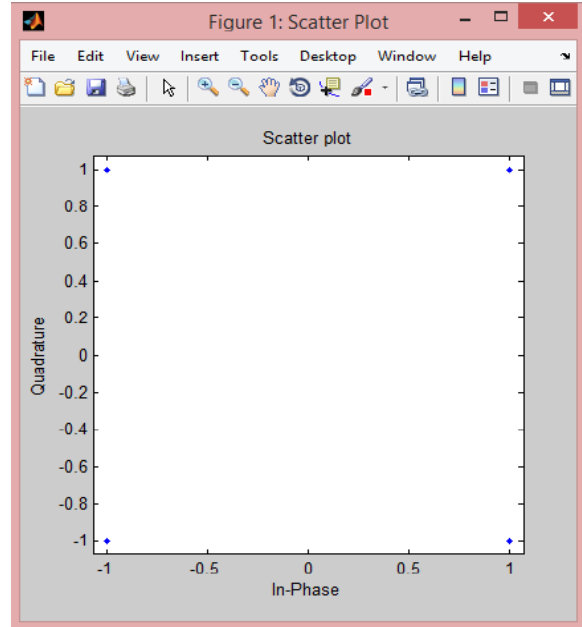


Figure 2.5: Constellation diagram of QPSK

Each adjacent symbol only differs by one bit, sometimes known as quaternary or quadriphase PSK or 4-PSK, or 4-QAM. QPSK uses four points on the constellation diagram, equispaced. With four phases, QPSK can encode two bits per symbol shown in the diagram to minimize the BER- twice the rate of BPSK. Analysis shows that this may be used either to double the data rate compared to a BPSK system while maintaining the bandwidth of the signal or to maintain the data rate of BPSK but half the bandwidth needed. The implementation of QPSK is more general than that of BPSK and also indicates the implementation of higher order PSK. Writing the symbols in the constellation diagram in terms of sine and cosine waves used to transmit them. This yields the four phases $\pi/4$, $3\pi/4$, $5\pi/4$ and $7\pi/4$ as needed. This results in a two dimensional signal space with unit basis functions. The first basis function is used as the in-phase component of the signal and the second as the quadrature component of the signal. Hence the signal constellation consists of the signal-space 4 points.

The demodulation of the received symbols is done based on the area i.e., presence of symbols on the plot of constellation diagram. The Figure 2.6 shows the scatterplot of the received data which is passed through an Additive White Gaussian Noise (AWGN) channel.

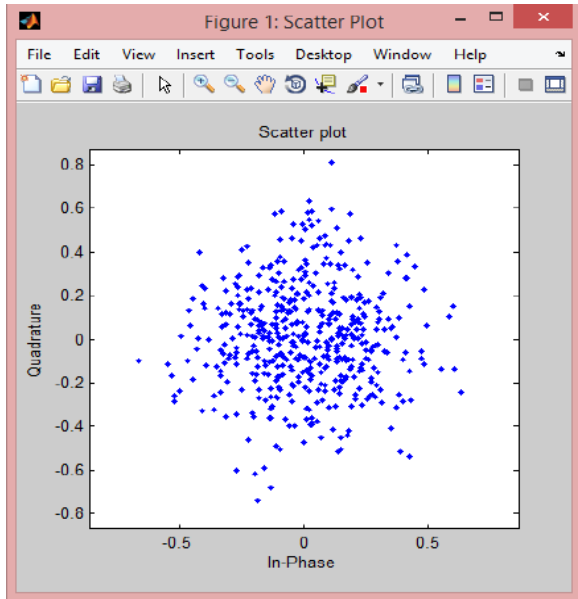


Figure 2.6: Received symbols

If the symbols are present in the first quadrant then they are decoded as symbol 0, if the symbols are present in the second quadrant then it is decoded as symbol 1, if the symbols are present in the third quadrant then it is decoded as symbol 3, else the symbol is decoded as symbol 2.

The demodulated symbols are represented on the constellation diagram as shown in the Figure 2.7.

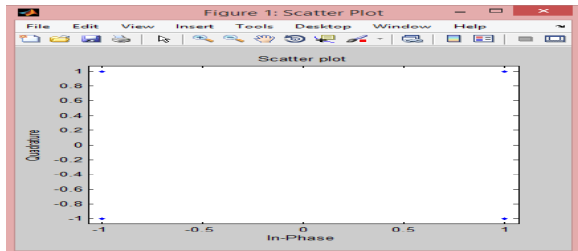


Figure 2.7: Demodulated symbols.

There might be errors in the demodulated data because the symbol positions may be altered when they are passed through the noisy channel. These errors are corrected by the Viterbi decoder. In this case the channel used is an Additive White Gaussian Noise (AWGN) channel.

2.5 Additive White Gaussian Noise (AWGN) channel:

In communication theory it is often assumed that the transmitted signals are distorted by some noise. The most common noise to assume is additive Gaussian noise, i.e. the so called Additive White Gaussian

Noise channel, AWGN. Even though the noise in reality is more complex, this model is very efficient when simulating. In a communication system data is often represented in a binary form. However, binary digits are elements in a discrete world, and, in all cases, we need to represent it in a continuous form. A signal in a digital communication system can be represented as by a continuous random variable. This value can be decomposed in two parts added together where,

$$Y = X + Z$$

is the information carrier component and noise component

The average power allocated by the variable is defined as the second moment,

$$P = E[X^2]$$

Now we can define an AWGN channel as: A Gaussian channel is a time-discrete channel with input and output = + where models the noise and is Normal distributed,

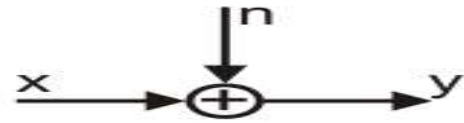


Figure 2.8: AWGN channel noise model.

2.6 Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT) in OFDM:

The concepts used in the simple analog OFDM implementation can be extended to the digital domain by using a combination of Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) digital signal processing. These transforms are important from the OFDM perspective because they can be viewed as mapping digitally modulated input data (data symbols) onto orthogonal subcarriers. In principle, the IFFT takes frequency-domain input data (complex numbers representing the modulated subcarriers) and converts it to the time-domain output data (analog OFDM symbol waveform).

In a digitally implemented OFDM system, the input bits are grouped and mapped to source data symbols that are a complex number representing the modulation constellation point (e.g., the BPSK or QAM symbols that would be present in a single subcarrier system). These complex source symbols are treated by the transmitter as though they are in the frequency-domain and are the inputs to an IFFT

block that transforms the data into the time domain. The IFFT takes in N source symbols at a time where N is the number of subcarriers in the system. Each of these N input symbols has a symbol period of T seconds. Recall that the output of the IFFT is N orthogonal sinusoids. These orthogonal sinusoids each have a different frequency and the lowest frequency is DC.

The input symbols are complex values representing the mapped constellation point and therefore specify both the amplitude and phase of the sinusoid for that subcarrier. The IFFT output is the summation of all N sinusoids. Thus, the IFFT block provides a simple way to modulate data onto N orthogonal subcarriers. The block of N output samples from the IFFT make up a single OFDM symbol. After some additional processing, the time-domain signal that results from the IFFT is transmitted across the radio channel. At the receiver, an FFT block is used to process the received signal and bring it into the frequency domain which is used to recover the original data bits.

The block diagram of Figure 3.7 shows a general representation of the system employing Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) to generate OFDM signal.

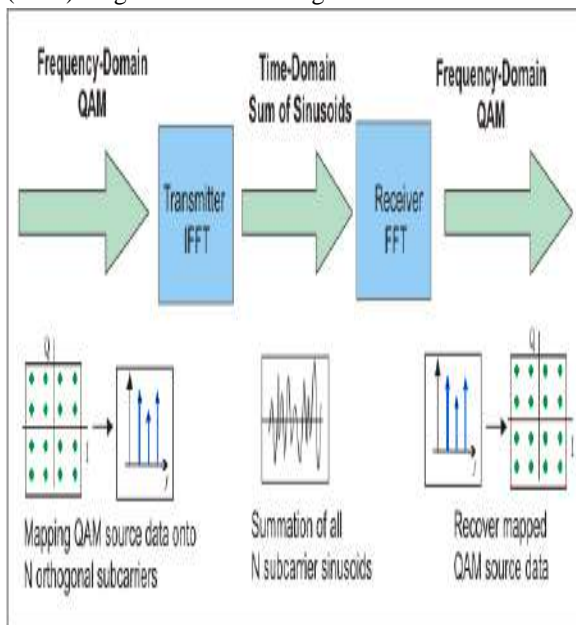


Figure 2.9: Simplified OFDM system block diagram.
2.6.1 Working of IFFT:

Here the bit stream is taken, distributed the bits, one bit at a time to the four sub-carriers as shown in Figure 2.10

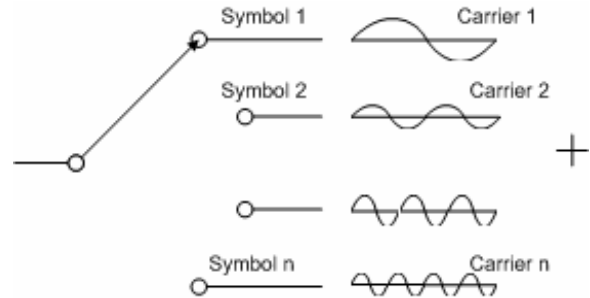


Figure 2.10 Selection of independent carriers.

The time and frequency domain representation of the individual modulated carrier is as shown in the Figure 2.11

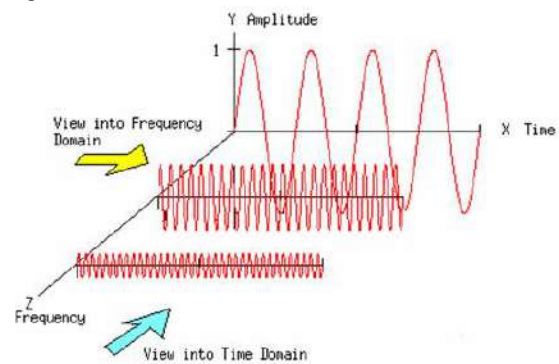


Figure 2.11: Time and Frequency domain representation of modulated carriers

Now all four modulated carriers are added to create the OFDM signal, often produced by a block called the IFFT as shown in the Figure 2.12.

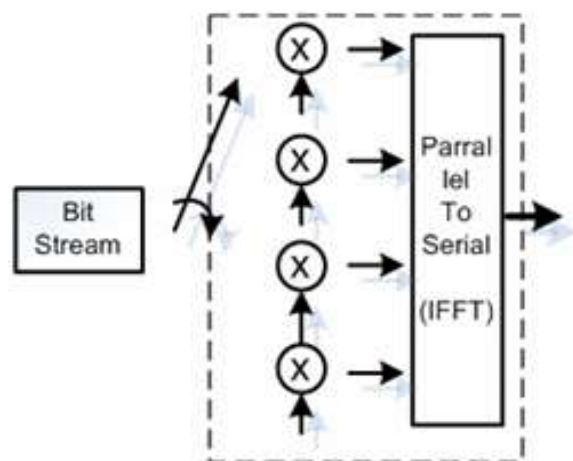


Figure 2.12: Generation of OFDM signal using IFFT.

The time domain output of IFFT block appears as shown in the Figure 2.13

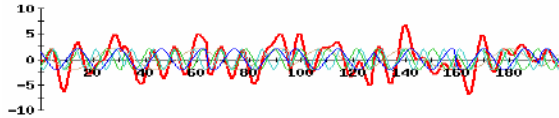


Figure 2.13: OFDM time domain signal

This wave is transmitted through the communication channel such as an AWGN channel as described in section 2.5. The receiving antenna extracts the signal and performs FFT.

2.6.2 Working of FFT:

FFT takes a time domain signal as input and generates a frequency spectrum corresponding to it. The working of FFT is shown in the Figure 2.14

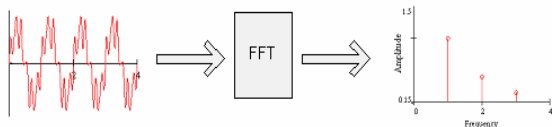


Figure 2.14: Fast Fourier Transform.

For an OFDM the spectrum can be represented as shown in the Figure 2.15

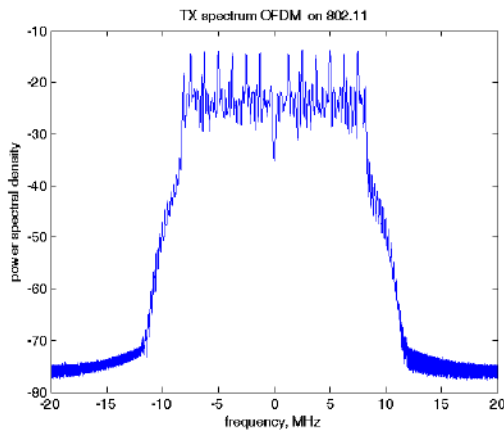


Figure 2.15: OFDM spectrum

2.7 Addition / Removal of Cyclic Prefix:

In order to preserve the sub-carrier orthogonality and the independence of subsequent OFDM symbols, a cyclic guard interval is introduced. The guard period is specified in terms of the fraction of the number of samples that make up an OFDM symbol.

The cyclic prefix contains a copy of the end of the forthcoming symbol. Addition of cyclic prefix results in circular convolution between the transmitted signal and the channel impulse response. Frequency domain

equivalent of circular convolution is simply the multiplication of transmitted signal's frequency response and channel frequency response, therefore received signal is only a scaled version of transmitted signal (in frequency domain), hence distortions due to severe channel conditions are eliminated [6].

Removal of cyclic prefix is then done at the receiver end and the cyclic prefix-free signal is passed through the various blocks of the receiver

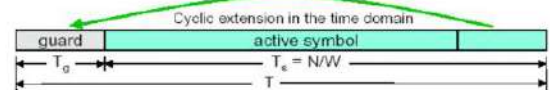


Figure 2.16: Cyclic Prefix.

IV. DESIGN FLOW

The design procedure consists of following steps:

- Creating a top level design of the complete system.
- Determining the basic operation of each block and creating the appropriate logic.
- I/O integration of the various logic blocks.
- Description of design functionality using Verilog hardware description language.
- Modelsim is used to simulate the design functionality and to report errors in desired behaviour of the design.
- Synthesis of the defined hardware is done which includes slack optimization, power optimizations followed by placement and routing.
- FPGA bit stream file is fed to the hardware.

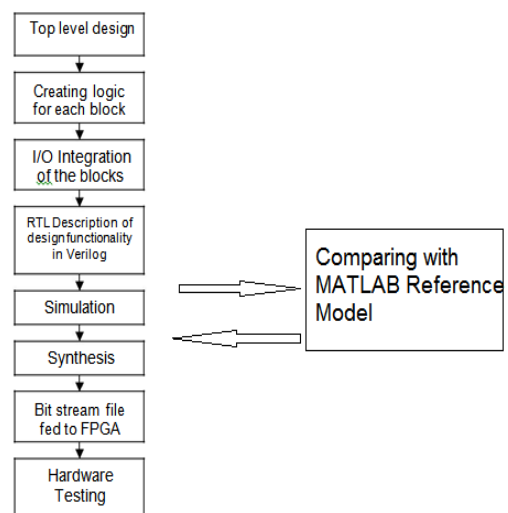


Figure 3.1 Project Design Flow

V. FIELD PROGRAMMABLE GATE ARRAY

By modern standards, a logic circuit with 20000 gates is common. In order to implement large circuits, it is convenient to use a type of chip that has a large logic capacity. A field-programmable gate arrays (FPGA) is a programmable logic device that support implementation of relatively large logic circuits [6].

FPGA is different from other logic technologies like CPLD and SPLD because FPGA does not contain AND or OR planes. Instead, FPGA consists of logic blocks for implementing required functions.

An FPGA contains 3 main types of resources: logic blocks, I/O blocks for connecting to the pins of the package, and interconnection wires and switches. The logic blocks are arranged in a two-dimensional array, and the interconnection wires are organized as horizontal and vertical routing channels between rows and columns of logic blocks [7].

The routing channels contain wires and programmable switches that allow the logic blocks to be interconnected in many ways. FPGA can be used to implement logic

configurable logic blocks, configurable I/O blocks, and programmable interconnect. Additional logic resources such as ALUs, memory, and decoders may also be available. The three basic types of programmable elements for an FPGA are static RAM, anti-fuses, and flash EPROM. Segments of metal interconnect can be linked in an arbitrary manner by programmable switches to form the desired signal nets between the cells. FPGAs can be used in virtually any digital logic system and provide the benefits of high integration levels without the risks of expenses of semicustom and custom IC development. The FPGA architecture can be seen in figure 5.

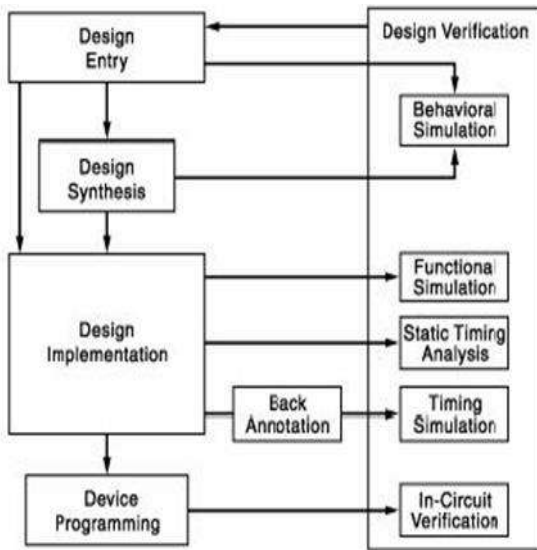


Fig4.1 FPGA Design Flow

This section concentrates on Field-programmable gate arrays (FPGAs) and their underlying architecture. FPGAs were invented by Xilinx in 1984. They are structured like the gate arrays form of Application Specific Integrated Circuits (ASIC). However, the architecture of the FPGAs architecture consists of a gate-array-like architecture, with

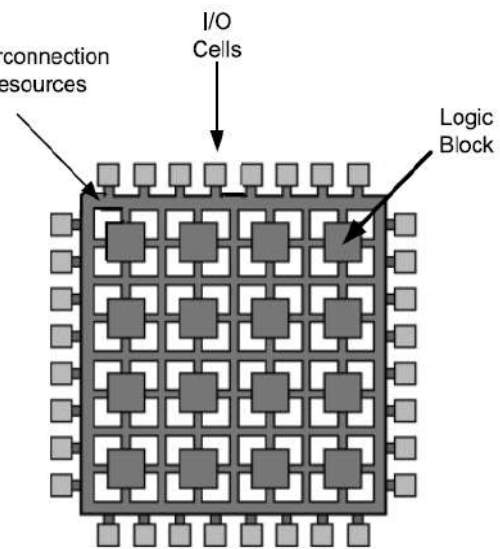


Figure 4.2 Interconnections in FPGA

FPGAs give us the advantage of custom functionality like the Application Specific Integrated Circuits while avoiding the high development costs and the inability to make design modifications after production. The FPGAs also add design flexibility and adaptability with optimal device utilization while conserving both board space and system power. The gate arrays offer greater device speed and greater device density.

Taking these into consideration, FPGAs are especially suited for rapid prototyping of circuits, and for production purposes where the total volume is low.

FPGA design flow which includes the necessary steps required to design an FPGA using Verilog HDL shown in Figure 4.3.

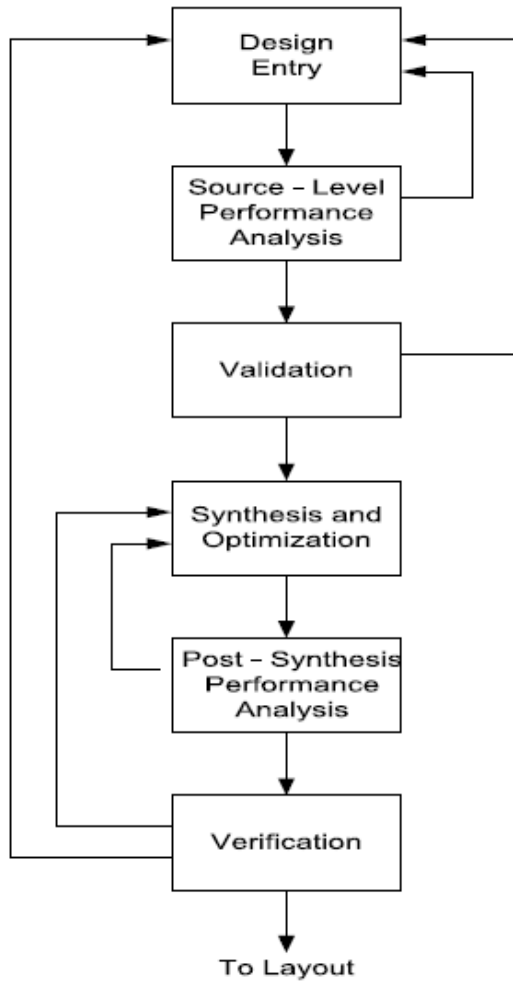


Figure 4.3 The necessary steps required to design an FPGA using Verilog HDL

Spartan Architecture Virtex-4 devices are user-programmable gate arrays. They consist of various configurable elements and embedded cores which are optimized for high-density and high-performance system designs. Functionality of Virtex-4 devices are I/O blocks, configurable logic blocks (CLBs), block ram modules, embedded Xtreme DSP slices and digital clock manager (DCM) blocks.

I/O blocks provide the interface between internal configurable logic and package pins. They are enhanced for source-synchronous applications [1]. The IOB registers are either edge-triggered D-type flip-flops or level-sensitive latches.

(F&G), two storage elements, arithmetic logic gates, large multiplexers and carry look-ahead chain. The function generators are configurable as 4-input look-up tables (LUTs). Storage elements are either edge-

triggered D-type flip-flops or level sensitive latches. Block ram modules provide 18Kbit true dual-port ram blocks which are programmable from $16K \times 1$ to 512×36 . And these modules are cascadable to form larger memory blocks. Block ram modules contain optional programmable FIFO logic for increased utilization. Each port in the block ram is totally synchronized and independent and offers three “read-during-write” modes. Embedded Xtreme DSP slices are cascadable and contain 18×18 -bit dedicated 2’s complement signed multiplier, adder logic and 48-bit accumulator. Each multiplier or accumulator can be used independently. Device uses DCM and global-clock multiplexer buffers for global clocking. DCM block provides self-calibrating, fully digital solutions for clock distribution delay compensation, coarse-/fine-grained clock phase shifting and clock multiplication/division. There are up to twenty DCM blocks available.

The general routing matrix (GRM) provides an array of routing switches fast between components. Programmable elements are tied to a switch matrix which allows multiple connections to the general routing matrix. All components in devices use the same interconnect scheme and the same access to the global routing matrix.

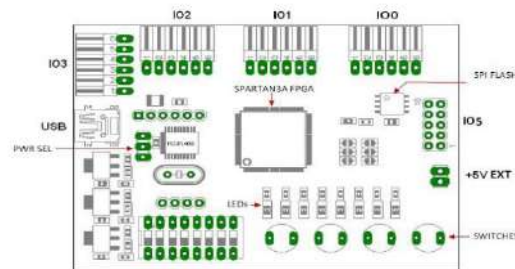


Figure 4.4 XC4VLX200 FPGA device schematic used in simulation

VI. VERILOG HARDWARE DESCRIPTION LANGUAGE

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integrated circuit (IC) designers. The other one is VHDL.

HDL allows the design to be simulated earlier in the design cycle in order to correct errors or experiment with different architectures. Designs described in HDL are technology-independent, easy to design and

debug, and are usually more readable than schematics, particularly for large circuits.

Verilog can be used to describe designs at four levels of abstraction [20]:

1. Algorithmic level (much like c code with if, case and loop statements).
2. Register transfer level (RTL uses registers connected by Boolean equations).
3. Gate level (interconnected AND, NOR etc.).
4. Switch level (the switches are MOS transistors inside gates).

The language also defines constructs that can be used to control the input and output of simulation.

More recently Verilog is used as an input for synthesis programs which will generate a gate-level description (a netlist) for the circuit. Some Verilog constructs are not synthesizable. Also the way the code is written will greatly affect the size and speed of the synthesized circuit.

VII. SYNTHESIS PROCESS IN VERILOG HDL

Synthesis is to construct a gate-level net list from a model of a circuit described in Verilog. The synthesis process is described in diagram below.

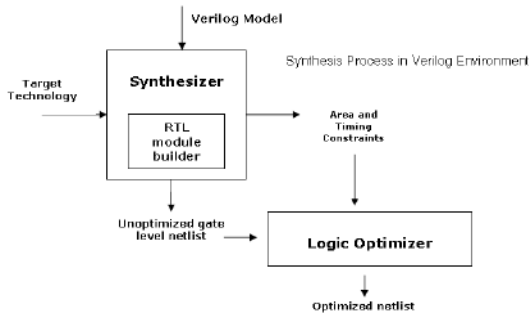


Figure 6.1 Synthesis Process in Verilog

A synthesis program may generate an RTL net list, which consists of register-transfer level blocks such as flip-flops, arithmetic-logic-units and multiplexers interconnected by wires. All these are performed by RTL module builder.

This builder is to build or acquire from a library predefined components, each of the required RTL blocks in the user-specified target technology.

The above synthesis process may produce an unoptimized gate level net list. A logic optimizer can use the produced net list and the constraint specified

to produce an optimized gate level net list. This net list can be programmed directly into a FPGA chip.

REFERENCES

- [1.] BER Analysis with OFDM using PSK/QAM Techniques for Wireless Communication, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 7, September - 2012
- [2.] Performance Enhancement of MIMOOFDMA: A Review Paper, International Journal of Computer Science and Communication Engineering, Volume 2 Issue 2 (May 2013 Issue)
- [3.] Mujtabaet al., TGn Sync Proposal Tech.Specification for IEEE 802.11 Task Group 2005, IEEE 802.11-04/0889r3.
- [4.] On the Capacity of OFDM-Based Spatial Multiplexing Systems, IEEE Transaction on Communication, Vol. 50, No.2, February 2002.
- [5.] K. Pagiantzis and P.G. Gulak, "Empirical performance prediction for IFFT/FFT cores for OFDM systems-on-a-chip," The 2002 45th Midwest Symposium on Circuits and Systems, vol. 1, 4-7 Aug. 2002, pp. 583-586.
- [6.] J. Park, Y. Kim, Y. Hur K. Lim and K.H. Kim, "Analysis of 60 GHz band indoor wireless channels with channel configurations," The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, Sep. 1998, pp. 617 – 620.
- [7.] R. Prasad, OFDM for Wireless Communications Systems, Artech House, 2004.
- [8.] L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall, 1975.