Summer 2011

Field programmable gate array based multiple input multiple output transmitter

Richa Shekhar

Department:

FIELD PROGRAMMABLE GATE ARRAY BASED

MUTIPLE INPUT MULTIPLE OUTPUT TRANSMITTER

by

RICHA SHEKHAR

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2011

Approved by

Dr. Kurt Kosbar, Advisor
Dr. Randy Moss
Dr. Steven Grant

ABSTRACT

MIMO is an advanced antenna technology compared to Single Input Single output (SISO), Multiple Input Single Output (MISO), and Single Input Multiple Output (SIMO) and is used to obtain high data rate in the system. Multiple-Input Multiple-Output (MIMO) systems have at least two transmitting antennas, each generating unique signals. However some applications may require three, four, or more transmitting devices to achieve the desired system performance. This thesis describes a comparison between different approaches like the microcontroller, ASICs and the FPGA available in the market for baseband signal generation. It also describes the design of a scalable MIMO transmitter, based on field programmable gate array (FPGA) technology that was selected among the processors due to its capability to provide reconfigurable hardware and software. Each module of the MIMO transmitter contains a FPGA, and associated digital-to-analog converters, I/Q modulators, and RF amplifiers needed to power one of the MIMO transmitters. The system is designed to handle up to a 10 Mbps data rate, and transmit signals in the unlicensed 2.4 GHz ISM band.

ACKNOWLEDGMENTS

I would like to express deep gratitude to my advisor Dr. Kurt Kosbar for giving me an opportunity not only to work in the Telemetry lab at Missouri S&T, but also for providing immense help and guidance during the entire course of my graduate studies. I would also like to thank Dr. Steven Grant and Dr. Randy Moss for their supervision throughout.

I would like thank my fiancé Pratik for unconditional support during my graduate studies. I would like to thank my parents, my brother, and my sister for their constant support and encouragement throughout my education.

I would also like to thank the staff and faculty of the ECE department for their constant guidance and for bearing patiently with my doubts and questions.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Multiple-Input Multiple-Outputs (MIMO) communication systems use multiple antennas at both the transmitter and receiver to increase data rates, or to provide more reliable communication than Single-Input-Single-Output (SISO) systems [1]. The greatest improvement in performance occurs when the systems are used in rich multipath and fading environments. It is a significant challenge to model these environments accurately, using analytically tractable mathematical models. Software simulations provide more flexibility, but as with many problems, the true performance of the system is not known until a physical device is constructed and tested in the target environment.

There is a need for a MIMO platform that is reasonably inexpensive, and can be easily reconfigured in the field to test a variety of space-time block codes [2]. It would also be helpful if the number of transmit antennas could be varied. This thesis describes the architectural design and testing plan, for a modular MIMO transmitter. The system can be configured to use an arbitrary number of transmit antennas. There were various types of processors studied for generation of baseband signals. The advantages and disadvantages of a microcontroller, ASICs, FPGA and DSP are studied and evaluated. The Field Programmable Gate Array (FPGA) is found to be best suited to baseband signal processing as the desirable data rate can be provided by it in a cost efficient way. As its name implies, this device can be reconfigured to produce arbitrary baseband in-phase (I) and quadrature-phase (Q) signals.

The use of FPGAs allows the end user to quickly alter the modulation and coding format used for the MIMO system which is not possible in the case of the microcontroller in which the entire processor may have to be changed to alter the modulation scheme. In addition, the transmitter can be configured to implement a SISO transmitter, a series of independent SISO transmitters, or it can implement beam forming algorithms and other modulation and coding schemes. The system is intended to work with data rates up to 10Mbps, and is currently configured to use the 2.4GHz unlicensed ISM band.

The thesis is divided into six sections; the first section provides the motivation and the basic outline of the thesis work. The second section describes the background for

the thesis work. The working of a typical communication system and various different antenna technologies used in the modern day communication system is described in detail. The section also describes the limitations of the Single-Input Single-Output antenna technology and the need of Multiple-Input Multiple-Output for the given transmitter design. The third section describes the Multiple-Input Multiple-Output System and also focuses on the design of a single module of the MIMO transmitter. The third section also describes that the multiple module can work independently to form multiple SISO transmitter or they can be daisy-chained. The fourth section describes a detailed literature survey and analysis of various types of processors available for use as the central processor in the transmitter. The microcontroller, DSP, ASICs and FPGA characteristics are studied in detail and evaluated. The fifth section describes a detailed overview of the design of various blocks for a MIMO transmitter. The schematic for a single transmitter module is also illustrated.

2. BACKGROUND

2.1. TYPICAL COMMUNICATION SYSTEM

Communication generally refers to physical transfer of information from one point to another. In the early twentieth century Claude Shannon also known as the ‘father of information theory’ gave a mathematical formula that allowed error-free encoding and digital transmission of the digital signals. This section describes a model of a typical communication system. The general digital communication model is as shown in Figure 2.1. It comprises of the source that originates the message that is transmitted by a transmitter in form of a signal. The signal travels through a channel. The channel and the source at which the message originates cannot be controlled by the user. External disturbances and noise are introduced in the signal during its propagation through the channel. The signal is then received by the receiver and passed to the destination.

The first block is the source of the message. The source can generate either an analog or a digital signal message also known as the baseband signal. One characteristic of the baseband signal is low frequency which makes it unsuitable for transmission as it may require very large antenna size for transmission. Also the signals that need to be transmitted fall in a finite bandwidth, so if they are transmitted simultaneously then there can be a lot of interchannel interference. Also due to low power content, the baseband signals cannot overcome the path losses, scattering and diffraction. To overcome the shortcomings of the baseband signal, it is converted into a passband signal by modulating it with a high frequency signal prior to transmission over a channel. The message from the source is passed through a source encoder that usually encodes the data into symbols formed from finite alphabet which are then transmitted. The main job of the source encoder is to remove the redundancy from the source message to decrease the bandwidth of the source message. The characteristics of the source coder depend on the type of message; e.g., the voice signals have an analog to digital converter, the keyboard contains the ASCII map coder, etc. The baseband signal after source encoding is not suitable for transmission as it may get corrupted by the noise in the channel. The process of adding extra bits to the baseband signal in a controlled way for error correction and detection purpose is known as channel encoding. The channel encoding improves the immunity of

the signal against corruption from noise. The channel encoder processes the information received by the source encoder and maps the input symbols to particular output symbols. The redundancy added in the channel encoding depends upon the type of coding scheme used in that particular digital communication channel. The channel encoder encodes the data such that there is reliable transmission of data over the channel.

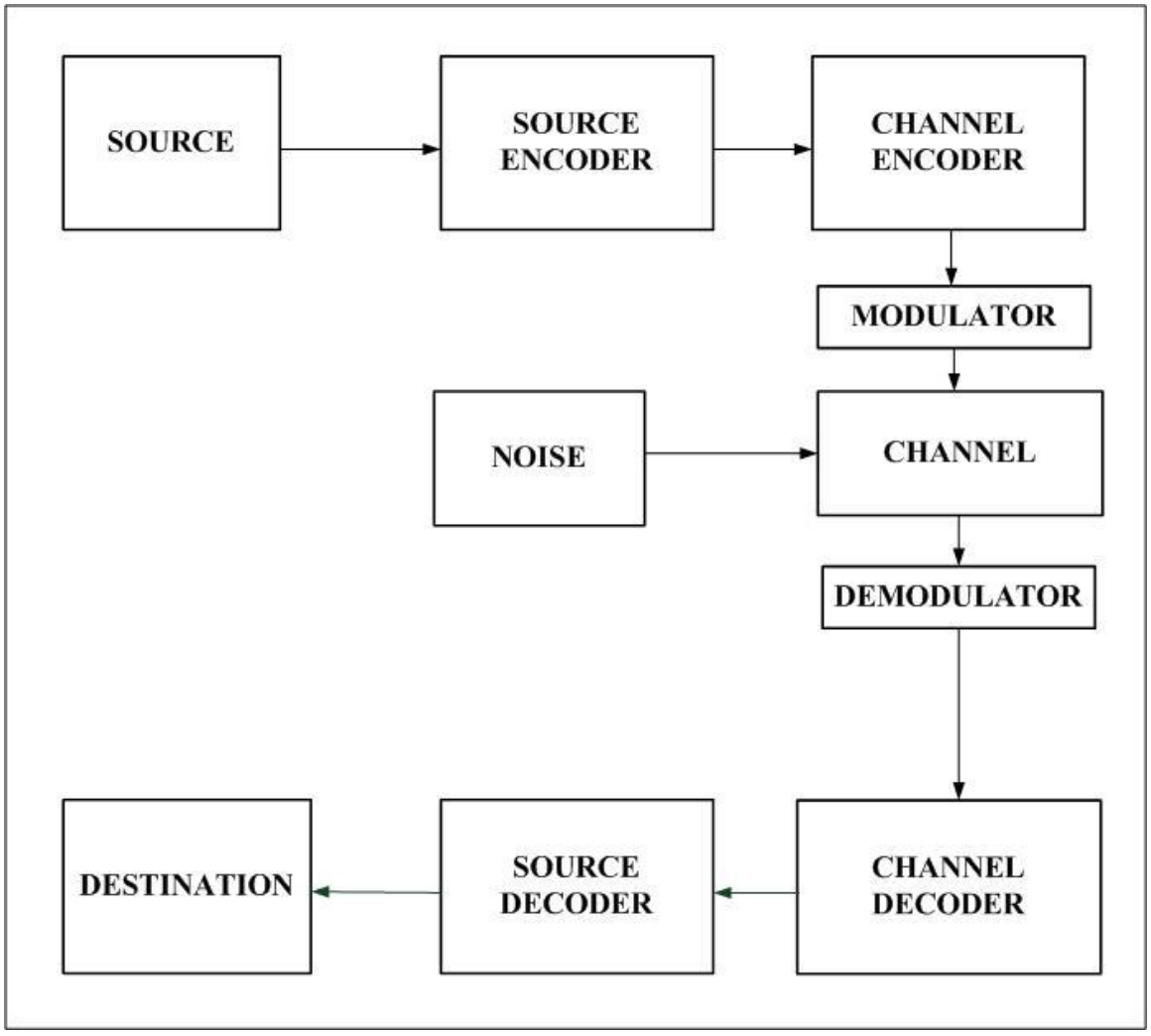


Figure 2.1. Digital Communication System

The digital modulation techniques convert the digital baseband signal into a time-varying analog signal by either changing the phase, amplitude or frequency of the carrier according the baseband digital signal. The modulated signal is passed into the channel; the channel model depends upon the media on which the transmission takes place. The received signal at the receiver is demodulated to obtain the baseband signal. Next, the channel decoder processes the signal to detect and possibly correct errors. The source decoder converts the signal back to the original message

2.1.1. **Advantages of Digital Communication System.** The digital communication scheme has significant advantages over the analog communication scheme [3]. The output of the digital communication system belongs to a finite set of waveforms unlike the analog communication system in which any value is mapped on the carrier signal. The output value of the digital communication is from a finite range whereas the values in the analog communication system are from an infinite range. Thus advanced signal processing techniques can be applied to digital communication system at both the transmitter end and receiver end, and that makes the system less susceptible to noise. This reduces the overall noise in the system that is otherwise amplified in an analog system in a long distance communication. The less noise in the system gives a better signal quality thus increasing the signal to noise ratio (SNR). The multiplexing of several channels in a digital communication system is easier. It is easier to implement digital systems using the digital logic gates and circuits. Also the cost of digital integrated circuits is rapidly decreasing and making the digital communication system cheaper.

2.1.2. **Disadvantages of Digital Communication System.** In order to increase the SNR of the system there is redundancy added in the transmitted signals that leads to an increase in the bandwidth of digital signals. Also the conversion of signals from analog to digital at high speed introduces distortion due to non-linearity of the analog to digital converters (ADCs). Quantization noise is introduced by the digital to analog converters (DACs) which reduces the SNR of the signal. To recover the signal at the receiver, synchronization is required between the transmitter and receiver.

2.2. SINGLE-INPUT SINGLE-OUTPUT SYSTEM

Different ways in which the antenna are configured for transmission of data from transmitter to receiver are single input single output (SISO), single input multiple output (SIMO), multiple input single output (MISO), and multiple input multiple output (MIMO). The diagram of a basic single input single output System is shown in Figure 2.2.

The single input single output system utilizes a single antenna at the transmitter and a single antenna at the receiver. As the data is transmitted by a single antenna at the source, the data rate of transmission is comparatively smaller than that of MIMO. The SISO systems are also less complex than the MIMO systems [4].

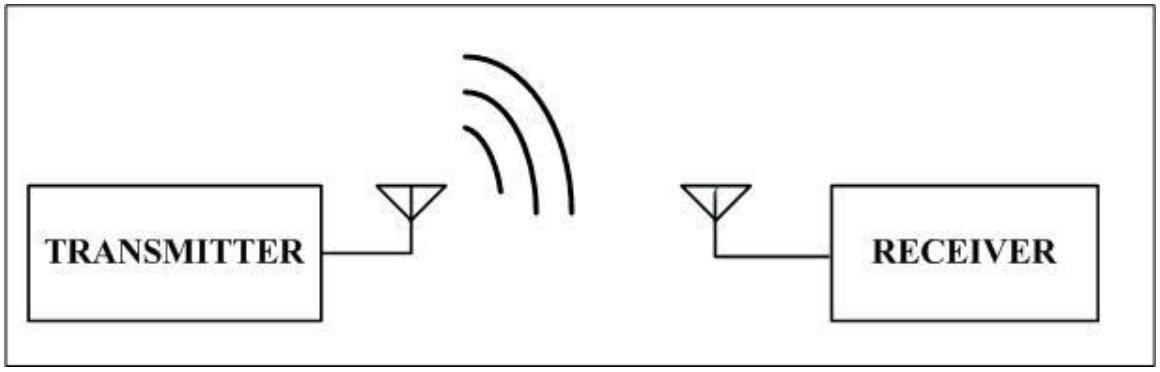


Figure 2.2. Single-Input Single-Output System

2.2.1. **Problems in SISO.** In a SISO system there is a high probability that the transmitter is obscured and the transmitted signal doesn't reach the receiver. This may be due to obstruction of the signal by natural structures such as hills or a cliff and also due to manmade structures such as buildings. This obstruction leads to the signal being reflected and diffracted; thus leading to attenuation, delay and phase shifts in the signals. The reflections of the signals lead to various constructive or destructive interference in the signals. The obstruction also tends to scatter the signals. As a result of scattering, the receiver detects varied signal power which seems to be traveling in multiple paths to

reach the receiver. This phenomenon is also known as multiple path fading. The multiple paths taken by the signal from transmitter to receiver is illustrated in Figure 2.3. Here T is the transmitter and R is the respective receiver. The signal to noise ratio is very poor in a SISO system as the single transmitted signal is more prone to noise. Thus there is a need to improve the SNR and efficiency of data transmission.

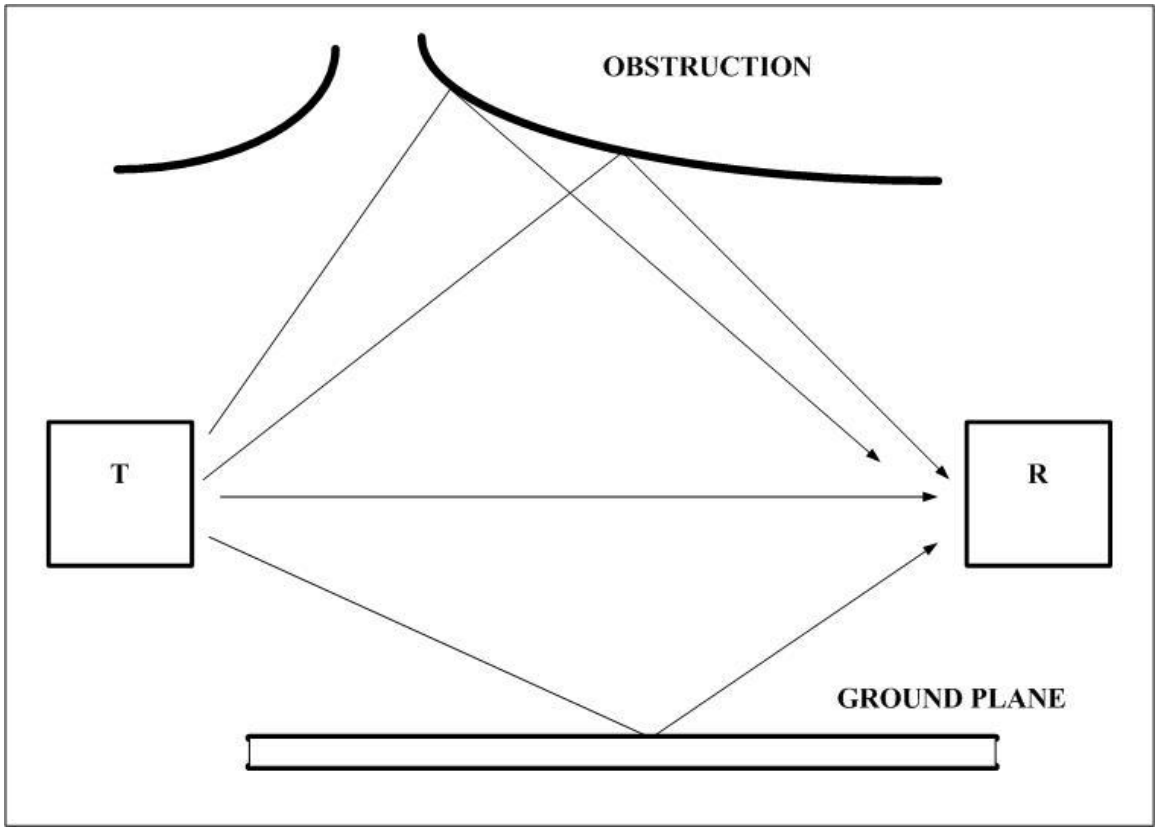


Figure 2.3. Multipath Fading

2.2.2. **Need for MIMO System.** According to the proposed Edholm's law of rates, the data rates are expected to increase exponentially [5]. The present data rate of 1Mbps to 2 Mbps for mobile cellular technology and 10 to 50Mbps for indoor systems are expected to increase to 20 Mbps for mobile systems and 500 Mbps for fixed systems

in the next five years. Higher order modulation schemes are already in use in most of the cellular standards but that is not enough for achieving the proposed data rates, Also exploiting the modulation schemes further will put a lot of complexity on baseband processing of the signals [6]. So in order to increase the data rate, a shorter inter-symbol duration can be employed. However this means that the frequency spectrum has to widen to accommodate the short inter-symbol duration [7]. This will possibly increase the inter-symbol interference and will lead to decreased SNR. The problem of inter-symbol interference can be solved by the use of multiple carrier technology which would again require high bandwidth. Thus there is a need for advance methods that provide high data rate while maintaining low SNR and minimum inter symbol interference. The work of Foschini and Gans [8] and Telatar [9] suggested MIMO antenna technology to achieve high data rates in digital communication. The work suggested use of more than one antenna at the transmitter and receiver.

3. MULTIPLE-INPUT MULTIPLE-OUTPUT SYSTEM

3.1. MIMO SYSTEMS

MIMO has been widely implemented to improve the data rates. It has already been implemented in the latest wireless standard of 802.11, LTE 3 GPP, HSPA and 4G. The wireless standard 802.16e recommends the use of two antennas at the base station and at the mobile station to increase interoperability of Wimax [10].

In MIMO the system does not focus on increasing the strength of one particular signal to increase the data rate, but rather it focuses on various different signals which are emitted by the multiple transmitter antennas. The block diagram of a basic MIMO system is as shown in Figure 3.1. According to Telatar [9], if there are M antennas present at the transmitter and N antennas present at the receiver the system achieves the capacity of min (M, N) separate channels. In MIMO systems, the gain achieved in terms of data throughput is much higher as compared to conventional SISO systems.

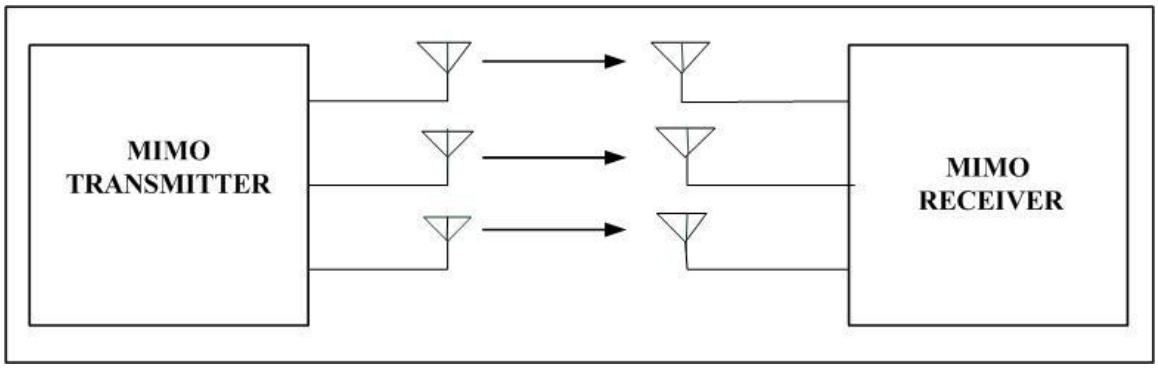


Figure 3.1. MIMO Communication System

There are multiple transmission schemes adopted for MIMO systems. These schemes are broadly classified as spatial multiplexing schemes and space-time coding schemes [8]. Spatial multiplexing schemes send independent data streams from the

different transmitter antennas which are separated by the antenna of the receiver using the interference cancellation type of algorithm [7]. Spatial multiplexing has a condition that the number of antennas on the receiver and transmitter should be the same. The data throughput of the system is increased by spatial multiplexing and high data rates are achieved. Space time coding uses the spatial diversity, i.e. if there are multiple signals from the transmitter to receiver, then the probability that all the signals will fade will be less compared to fading of a single signal. All the signals generated by the transmitter are likely to take different paths to reach the receiver so there a much smaller chance of more than one signal meeting the same type of obstruction. Thus the bit error rate is reduced by the space-time encoding. Space-time trellis codes are example of space time coding. Space-time coding does not require the same number of antennas on the transmitter and the receiver; thus it is advantageous in the cellular systems. Thus space-time codes are very beneficial in systems where the receiver has to be smaller compared to the transmitter.

Theoretically there should be a double increase in the capacity if the number of antennas is doubled. But practically the capacity is not doubled due to various losses at the channel and non-linearity of the transmitter and the receiver. There are several implementation challenges faced during design of MIMO system such as maintaining transmitter linearity, the receiver range imbalance and leakage of the signals from the modulators. Also there are problems due to the cross-talk between the two channels of the MIMO system.

3.2. SYSTEM ARCHITECTURE OF ONE MODULE

The system is designed to use a set of identical modules. Each module will drive one, and only one, of the transmit antennas. A high level block diagram of the module is shown in Figure 3.2. Data is fed into an FPGA, which performs the necessary coding and waveform generation operations. The FPGA may perform digital filtering, and arbitrary waveform synthesis. Two digital waveforms are generated by the FPGA, and sent to the digital-to-analog converter (DAC). The DAC output waveforms are then fed to a

modulator, and the RF waveform sent through a power amplifier before being transmitted.

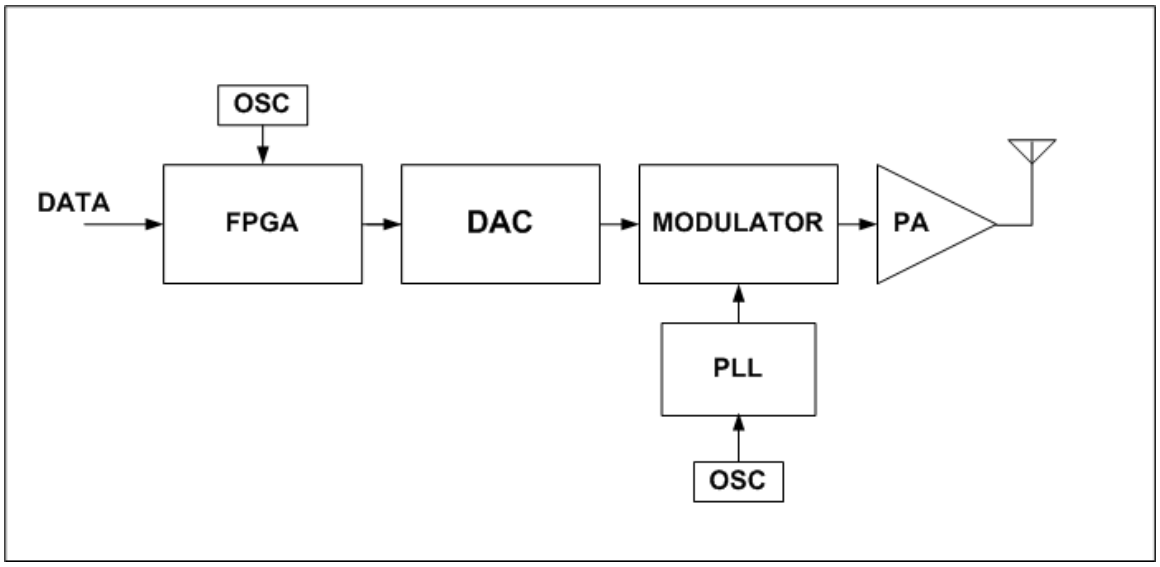


Figure 3.2: Architecture of One Module

The modulator uses an RF reference produced by a phase-locked loop device, which is in turn controlled by a crystal time base. This particular test bed is based on an Alter Cyclone II FPGA [11], which includes an embedded processor. The processor will be used to not only filter, and code, the data, but also to generate test patterns for testing the system and pilot symbols for measuring the channel state information during transmission.

A more detailed block diagram is shown in Figure 3.3. Beginning from the left hand side, we see the FPGA is clocked by oscillator 2. The goal is to perform pulse shaping using digital signal processing algorithms in the FPGA. The target data rate is 10 Mbps, and we would like to have at least 10 samples per symbol, so the FPGA needs to be clocked at a rate of at least 100 MHz

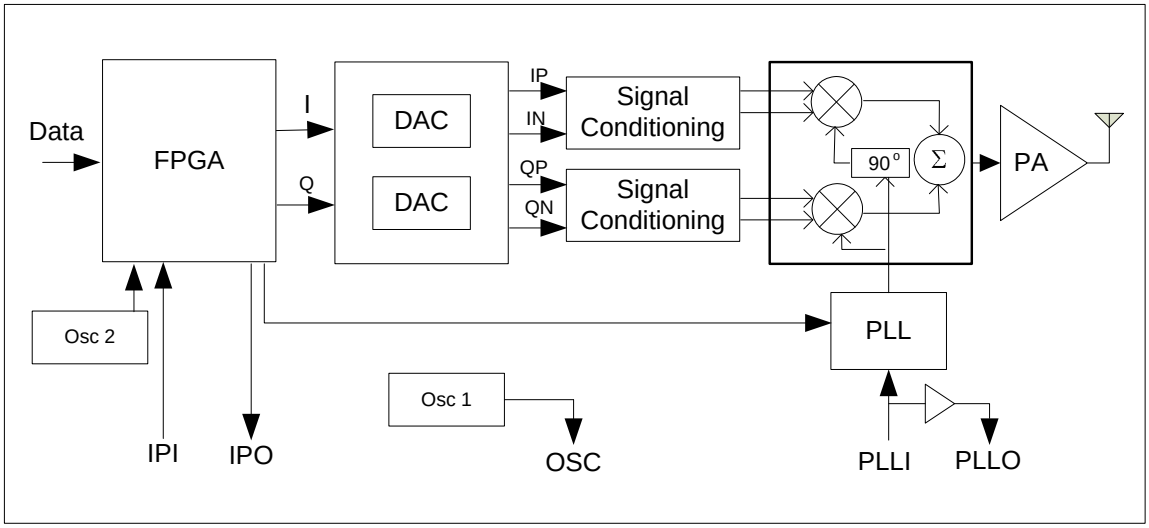


Figure 3.3. Detailed Module Architecture

The FPGA must generate two digital signals, and pass these to a pair of DACs to generate the I and Q baseband waveforms. These differential analog signals are then fed to signal conditioning electronics before being passed to the quadrature modulator. The signal conditioning circuit converts the current output from the DACs into voltage output.

The I and Q signals modulate an RF carrier produced by a phase locked loop (PLL). The PLL is stabilized by a crystal oscillator, labeled "Osc 1" in Figure 3.3. The ratio between the PLL output frequency and the oscillator is controlled through internal registers which are set by the FPGA as part of its power-up sequence.

When multiple modules are used, it may be necessary to synchronize the PLL. To accommodate this, the crystal oscillators are not connected directly to the PLL. The two must be connected via an external jumper, and may be daisy chained if necessary.

3.3. CONNECTION TO MULTIPLE MODULES

The modular design of this system allows it to be configured in a variety of ways. In the most basic configuration, the boards can be used simply as independent transmitters, each operating on a different data pattern, as shown in Figure 3.4. Each

module will need to have a jumper connecting the oscillator output to the PLL input. In this configuration, there will be no synchronization between the modules. This would be appropriate if the modules were used with frequency division multiple access systems, or code division multiple access.

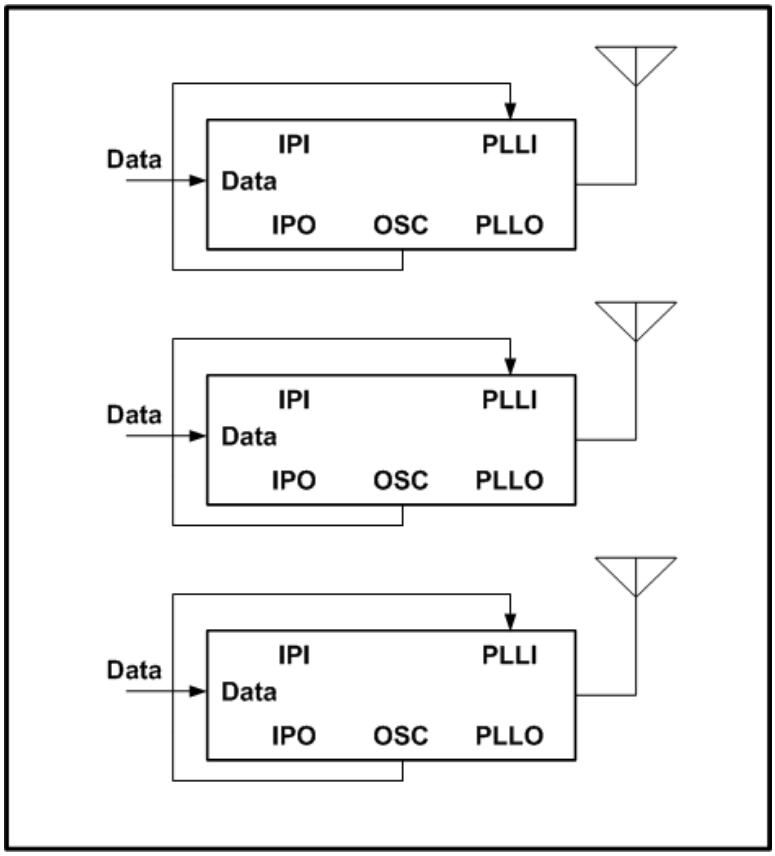


Figure 3.4. Multiple SISO Transmitter Configurations

In the multiple SISO operation, there is no centralized control of the transmit frequency of the various modules. Each module references its carrier frequency to an on-board crystal oscillator. These oscillators drift with time and temperature, so the corresponding carrier frequencies will also not remain synchronized. There are applications where it may be desirable to have all transmitters phase locked to the same

reference oscillator. This can be accomplished using the "PLLO" outputs from the modules.

As shown in Figure 3.5, the crystal oscillator from the top module may be used to drive the phase locked loops of an unlimited number of other modules. Each module contains a buffer amplifier which will reproduce a copy of the PLLI signal at the PLLO port. By daisy chaining the modules together, all of them can be frequency locked to the same crystal oscillator. There will be a phase shift introduced by the interconnecting cables, in addition to the buffer amplifiers. Perhaps even more importantly, when it comes to phase synchronization, there will be an unresolved phase ambiguity in the PLL of each module.

Because of the issues listed above, it will not be possible to develop phase synchronous RF carriers on each of the modules. However the common crystal time base should maintain the modules in frequency lock. In the configuration shown in Figure 3.5, the modules are still using independent data paths. These could be connected together at the user's discretion, to create a system which had frequency diversity.

The final configuration, and the one for which the system was originally planned, is shown in Figure 3.6. This is the MIMO configuration. As with the previous arrangement, all of the modules are frequency locked to a common low frequency crystal time base. It is not critically important that the top module's time base be used for this; it should work equally well to use the time base from any of the modules.

The new feature in Figure 3.6 is that only one of the modules is fed the data to be transmitted. This module examines the data, to determine the waveform which it must transmit. The module then sends the same data over the interprocessor (IP) data bus to the second module. This module will use a slightly different algorithm for determining its transmitted waveform. The details of the algorithms used will depend on the particular space time block code used.

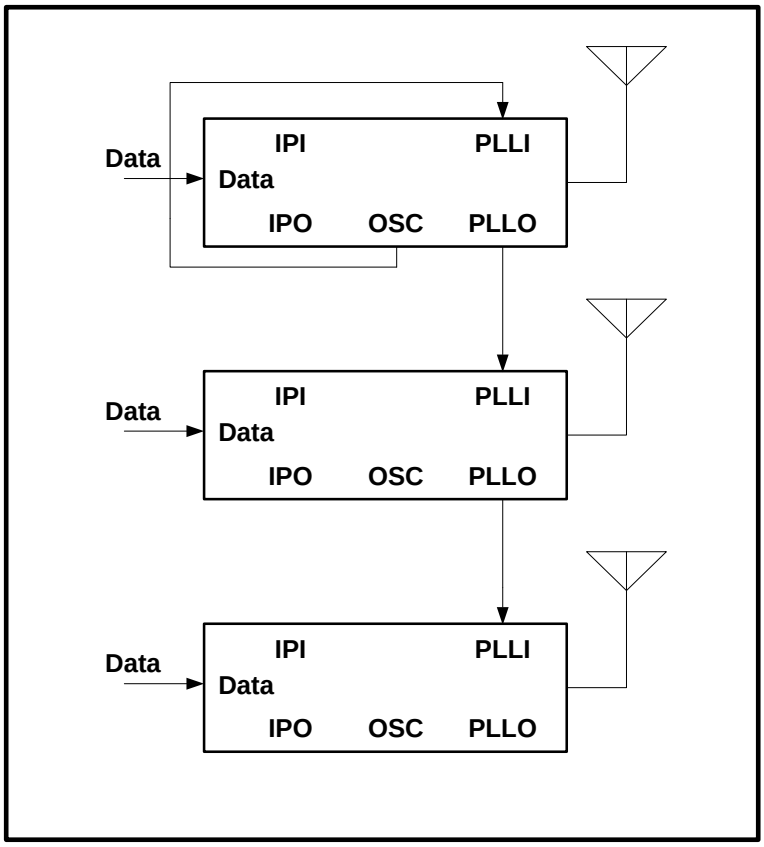


Figure 3.5. Multiple SISO Transmitters with Synchronized Carriers

If there are more than two transmit antennas in the MIMO system, the second module will relay the data to the third, and subsequent, modules through a daisy chained interprocessor data bus, as shown in Figure 3.6. Each module will need to know which portion of the space time block code it is assigned to generate.

As an alternative, the interprocessor data bus could be used to send configuration information in addition to the data. A framing protocol could be used, where in addition to the data bit to be transmitted; each processor would insert a digital word to indicate which part of the space time block code it is implementing. The subsequent modules could then read this data, and determine which portions of the code they should implement.

Another portion of the interprocessor communication must be a synchronization signal to indicate when each processor should begin transmitting a new symbol. It is

important that all modules begin, and end, the symbols at the same time. The PLL bus only serves to synchronize the carrier frequencies, and would not normally be used for symbol synchronization.

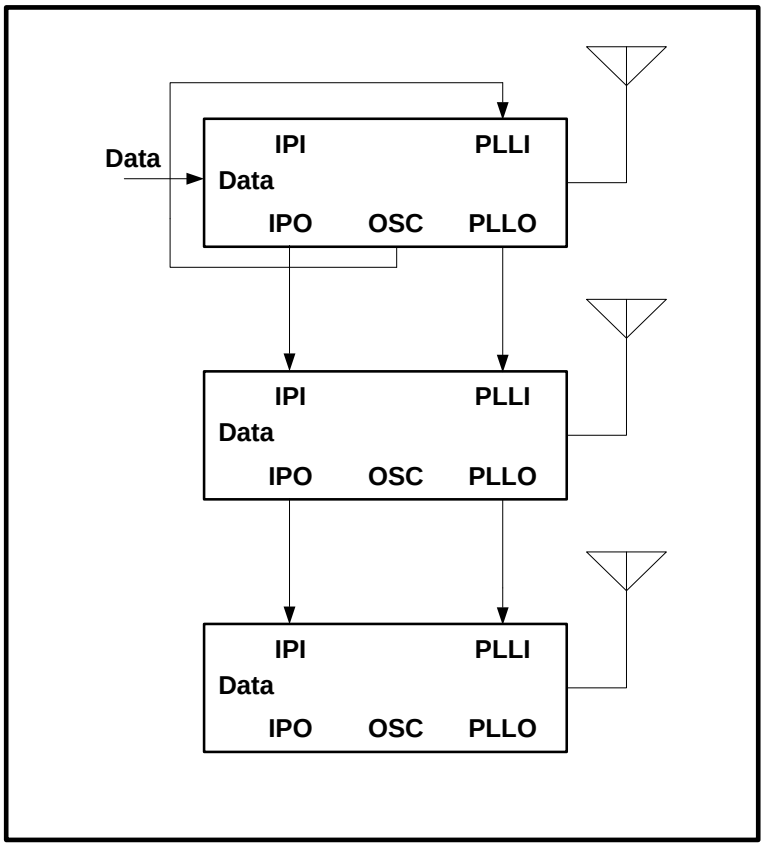


Figure 3.6. MIMO Configuration

4. BASEBAND SIGNAL GENERATION APPROACH

4.1. MICROCONTROLLER-BASED APPROACH

A microcontroller is a general purpose device and is often called a mini computer. It is an advancement of the microprocessor and that includes typical features of a microprocessor, like the central processing unit (CPU), arithmetic logic unit (ALU), and program counter. Along with the basic features of the microcontroller it also consists of serial and parallel input/output ports, hardware counters, on-chip ROM, RAM, etc. A microcontroller is basically a self-contained chip that is utilized to control small devices like the microwave, washing machine, etc. The main function of the microcontroller is to take up all the data using the various input ports, process the data, make logical decisions to operate the system and often output the result on the output ports [12]. The input and output ports provide the external interface to the microcontroller with other systems on the board.

The block diagram of the microcontroller is as shown in Figure 4.1. The arithmetic logic unit (ALU) of the microcontroller ranges from a 4-bit to a 64-bit processing unit. The size of the ALU determines the size of the data bus. The read-only memory (ROM) stores the boot-up program. The boot-up program is the first program that initializes the microcontroller every time it is switched on. The random access memory (RAM) is used to store the various temporary data during the run time of the microcontroller. The physical memory locations on-chip, called registers, are used to control and temporarily store operands and results. The program counter stores the address of the next instruction to be executed. On reset, the clock counter normally resets to the first location in the ROM. Also the stack pointer points to the top of the stack which is used to store the register information and the program counter during a system call or interrupt routine. The clock circuit in the microcontroller provides the clock cycle and timing reference required for execution and communication with the help of an external crystal oscillator. The microcontroller also consists of serial interfaces like the universal asynchronous receiver/transmitter (UART) and the universal peripheral bus (USB) bus interface to communicate with external devices [13].

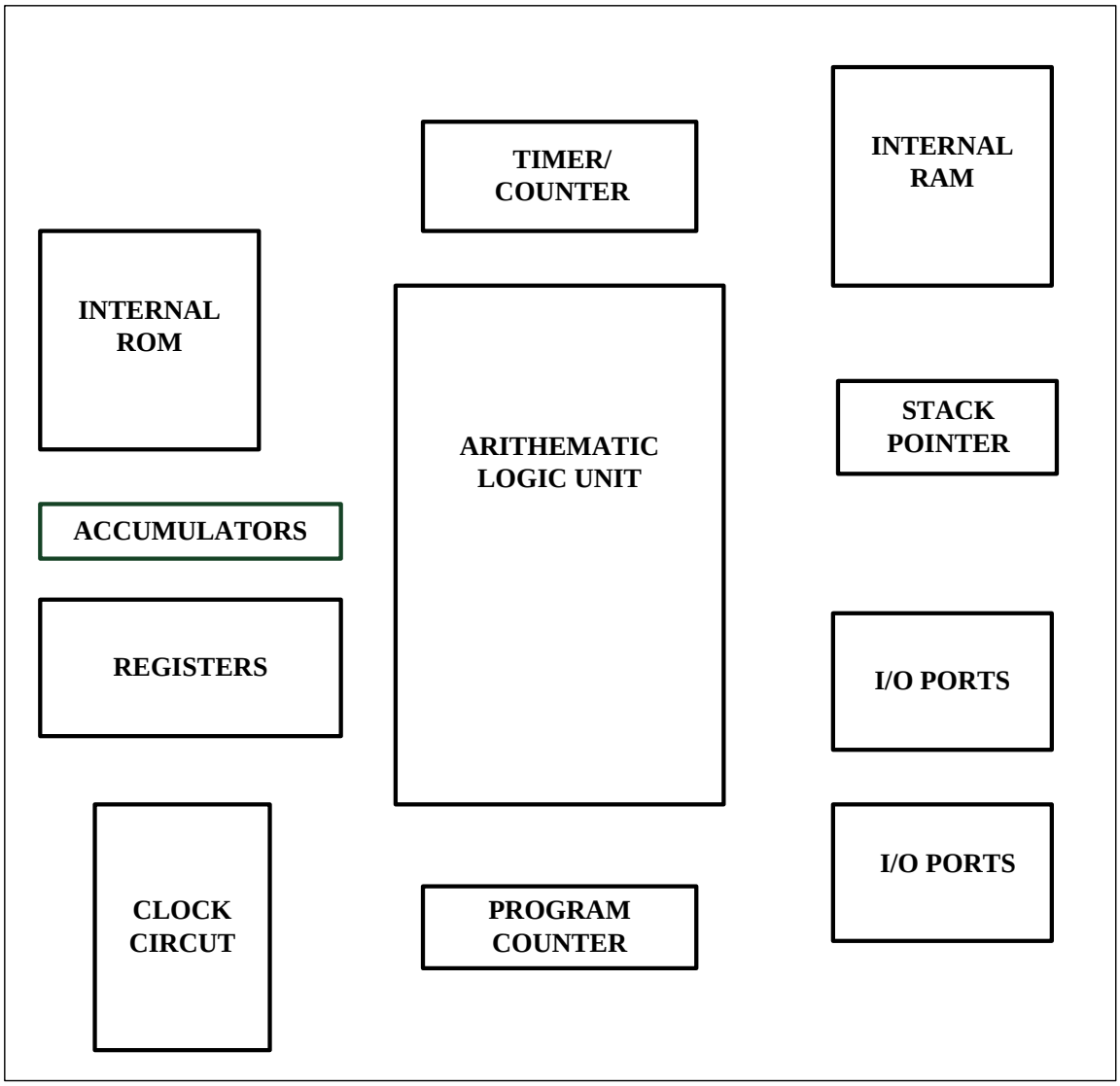


Figure 4.1. Microcontroller Block Diagram

4.1.1. **Microcontroller Application and Design Cycle.** A microcontroller can be a suitable choice as a central processing unit in systems that require less computational power. The low power consumption and low cost of the microcontroller make it an ideal choice for building systems having a large deployment in the consumer market like the microwave, washing machine, etc. The microcontroller has various low power modes that can be switched on depending upon the application in consideration. The microcontroller uses a very low static power but the disadvantage is that, if the

microcontroller becomes fully functional, it requires more time to attain full functionality while it is in its lower power consumption mode.

Figure 4.2 below shows the design cycle of the systems based on a microcontroller as their central processor. The requirement of the system is analyzed first and a suitable microcontroller is chosen depending on various parameters like input/output ports required, processing speed requirements, peripherals required, etc. The selected microcontroller has fixed hardware structure like the memory capacity and the number of interfaces. Software code is written in a high-level language such as the 'C language' or a low-level language such as 'assembly language'. The assembly and the high level code can be compiled by using assemblers and compilers respectively to obtain the hex file that can be directly written on the ROM of the processor. Many off –the-shelf tools can be used to compile the high level code.

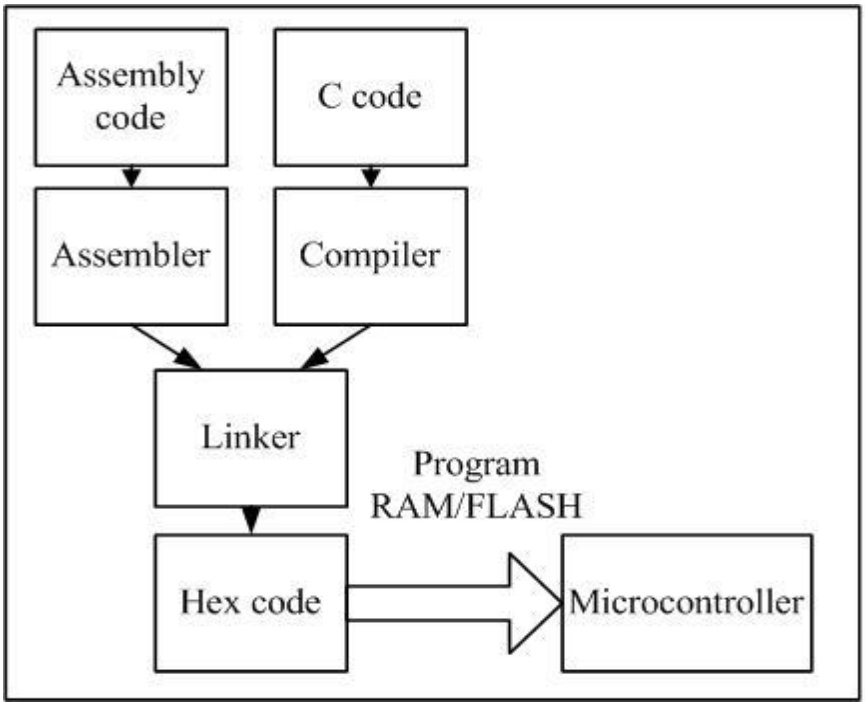


Figure 4.2. Design Cycle of Microcontroller

There are several integrated development environments available like the KIEL on which the hardware and software code can be simulated. If the simulation works perfectly the program is loaded on to the microcontroller and the microcontroller performs as desired by the user. Other debugging techniques include use of JTAG emulators and in-circuit debuggers that can be used to modify/read microcontroller registers, step through the program and insert breakpoints.

4.1.2. **Suitability for MIMO Transmitters.** The microcontroller is difficult to incorporate as the main processor for implementing the baseband processing of the MIMO transmitter due to its small computation power. The microcontroller compromises the speed of the system for low power consumption. The microcontroller is beneficial in systems that are resource constrained, applications like the mobile-based application, and portable products. In these types of applications the power consumption and size of the processor are important constraint [14]. There are a lot of parameters that are taken into consideration before the choice of processor is made. The non-recurring engineering cost, time to market, and cost of change were a few major parameters that led to the microcontroller not being selected as the central processor of the MIMO transmitter.

This transmitter is designed to handle a variety of space-time block codes and modulation schemes suiting the needs of a variety of communication channels so the program scheme may have to be upgraded from time to time. The microcontroller has a fixed hardware structure and has no scope of addition of any more desired hardware structure that may be needed later for working of a particular program scheme. Also, the fixed hardware structure makes it difficult to upgrade the system as per the demands of any new communication channel in the future. Due to the above concerns, the non-recurring cost is high. The microcontroller deployed to one particular modulation scheme has to be programmed again for a different modulation scheme because multiple modulation schemes cannot be implemented simultaneously on a microcontroller due to the small memory capacity of a microcontroller.

For a MIMO transmitter implementation, there is a need of a processor which could support the design for a long time without making a lot of changes. If the adopted microcontroller becomes out of date, a new processor cannot be implemented without

writing a new software code. By changing the hardware the software needs to be updated according to the new microcontroller which can be quite a troublesome task using a lot of resources in terms of labor, cost and time. This cost of change is one of the deterrents for using it for MIMO transmitters.

The cost of the microcontroller can be an attractive feature to adopt it as the central processor in the MIMO transmitter but this project is a developmental project that once tested can be developed on more cost-effective ASICs for large volumes. The most attractive feature of the microcontroller is the power consumption, comparatively lesser than any DSP, FPGA or ASICs, but in this proposed system the power consumption is not the important consideration as the transmitters are not mobile handheld devices operating on batteries.

Due to above discussed disadvantages such as cost of change and non-recurring cost, the microcontroller was not regarded as the choice for baseband processing for the proposed transmitter design.

4.2. DIGITAL SIGNAL PROCESSOR-BASED APPROACH

Digital Signal Processors or DSPs are basically derived from the microcontrollers and microprocessor to perform computation on real world signals and data. The microcontrollers are good for performing logic decisions and the functions that require more of a control-based approach, but don't have a high speed requirement. The high speed processing is necessary to perform quick mathematical operations on real-time or time-critical data. The DSP generally consists of a central processing unit like the microprocessor but does not necessarily have all the features of a microcontroller like the high number of general purpose I/O ports. The most important feature of a DSP includes the MAC unit, or the multiply and accumulate unit, and special division units to perform fast number crunching. This provides execution of the multiply and accumulate instructions in a single instruction cycle [15]. To incorporate the multiply and add feature the DSP has integrated accumulator and multiplier functional units in its arithmetic processing unit. The DSP also has extra accumulator registers to avoid overflow of the

accumulated result. This kind of feature is very helpful in signal processing applications like digital filters and Fourier transforms [15].

The DSP has another specialized feature in that it can complete multiple accesses to the memory in a single instruction cycle. So while performing the task of fetching an instruction from the memory it can also store the value of the previous operand in the memory. There are multiple tasks like these that can be performed simultaneously without interruption and compromise on speed. Most of the application of signal processing is data intensive requiring the DSP to process and store a lot of data so a lot of bandwidth is required for transferring the data between the processor and the memory that is facilitated by the memory structure of the DSP processor. To enhance the simultaneous use of memory the DSP chips have multiple independent memories and multiple on chip buses [15]. The DSPs also have enhanced features like direct memory access to handle the input and output commands of the DSP efficiently. The peripheral design has also been modified to access the other chips on the board efficiently.

Broadly, the DSPs are divided into floating point DSP and fixed Point DSP. The floating point DSP has an advantage that it can perform operations on floating point numbers or fractions directly. The fixed point DSP cannot perform operations on a floating point number directly, but it may require much larger execution time than that on fixed integers.

Figure 4.3 below shows the design cycle of a project in the digital signal processor. The design cycle of the system based on DSP is similar to that of the microcontroller as shown in Figure 4.3. The signal processing application is first analyzed and identified. The market has a large variety of the digital signal processing chips available so the suitable DSP for a particular application is selected. The software code for DSP chips is written in a high level language such as the ‘C language’. The DSP companies offer a large variety of integrated development platforms like the Texas Instrument’s Code Composer Studio for compiling and debugging code. The high level code is compiled and converted into a hex file to be loaded on the DSP chip. The DSP chip then performs the functions as desired by the user.

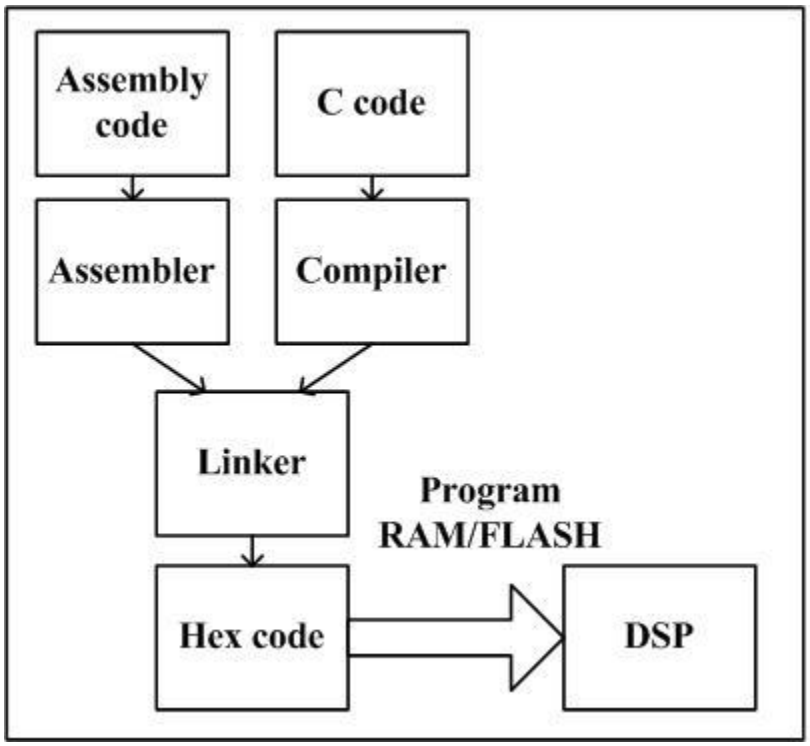


Figure 4.3: Design Cycle of a DSP Processor

4.2.1. **Differences Between a Microcontroller and DSP.** The authors in state that the microcontrollers and microprocessors are designed on Von Neumann architecture and they are not designed to handle real-time signals [15]. The Figure 4.4 shows the Von Neumann architecture [16]. In the Von Neumann architecture the instructions are loaded sequentially from the program stored in the memory. The program and data memory are unified leading to contention for the memory. The digital signal processors are a derivative of microcontrollers and generally use the modified Harvard architecture having a separate data and program memory. The Figure 4.5 [16] shows the Harvard architecture of digital signal processors.

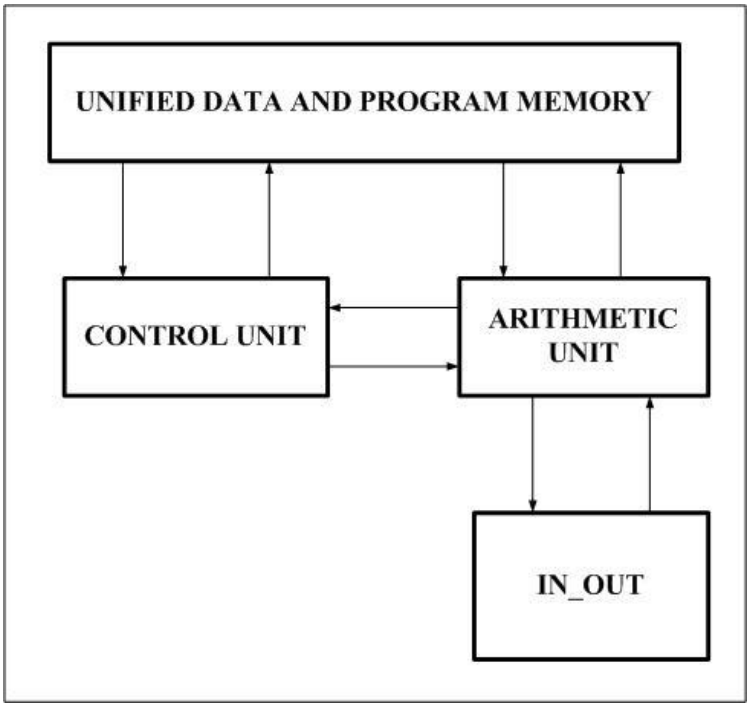


Figure 4.4: Von Neumann Architecture

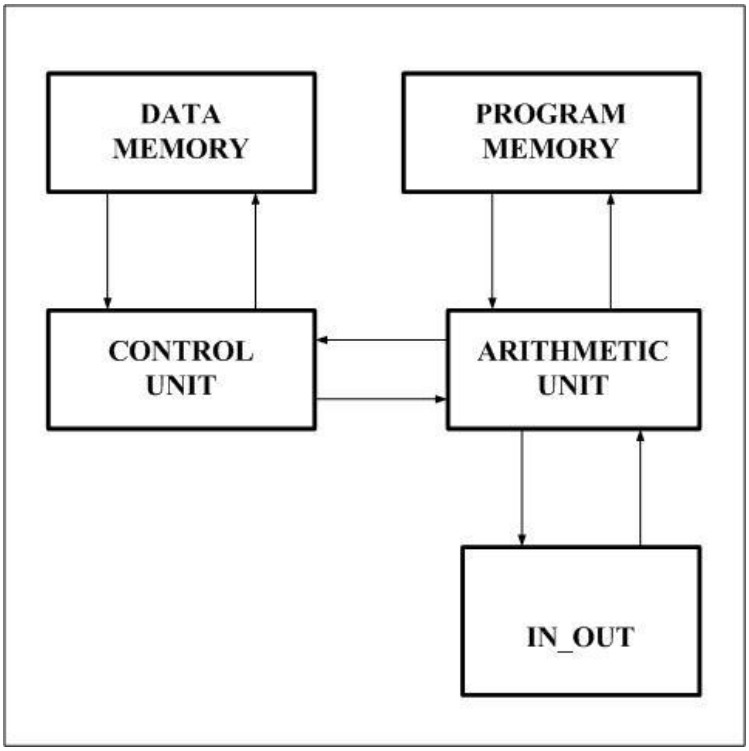


Figure 4.5: Harvard Architecture

The DSP processors offer increased parallelism and pipelining than the microcontroller due to separate program and data memory. Most of the instructions in the instruction set of DSP are modified for increase in parallelism. The main advantage of DSP Processors is in the field of speech processing like speech compression, speech modification and communication. They are able to do quick signal processing due to the new incorporated instruction set, and their ability to perform convolution that forms the basis of many of the operations in signal processing. They can perform multiple instructions in one clock cycle [15]. Also the MUL instruction in a microcontroller takes about four clock cycles, whereas the MPY instruction in a DSP takes just one clock cycle. All the digital signal processors have few features in common that allow increased parallelism and faster processing in the field of signal processing.

4.2.2. **Suitability for MIMO Transmitter.** The DSP may have multiple MAC units but the signal processing application may require additional units that can be operated independently in parallel. Any number of parallel multipliers can be implemented in FPGAs and ASICs using hardware description languages. It implies that if there are N terms to be multiplied and summed, then by using an implementation of M multipliers the execution can take place in N divided by M cycles [17]. Another disadvantage of the DSP processors is that they have very long pipelines due to which the latency of the DSP is increased. The change in the program flow in the DSP processor causes the entire pipelined data to crash thus decreasing the performance of DSP processor.

The DSP and microcontrollers have an advantage that there is lot of support available from the industry and their features are well documented which is not the case with FPGAs and ASICs as they require more engineering and more research from the developer side to manufacture a particular application [17]. The project under consideration requires the feature of re-configurability of hardware and software for changing the modulation schemes for different communication channels and environments that cannot be provided using the DSP processor. Also the speed at which the data has to be generated requires advanced features like multiple parallel multipliers on the hardware that can be configured using an FPGA or ASIC.

4.3. ASIC-BASED APPROACH

The application specific integrated circuits or ASICs architecture is generally made up of logic cells and metal interconnects. The cells are logical blocks which are actually realized by the use of transistors. These several cells are connected by the long metal interconnects to form an ASIC. The ASICs are generally divided into full custom ASICs and semi-custom ASICs.

4.3.1. **Full Custom ASICs.** In a fully customized ASIC all the circuit elements and designs are for a particular application. The full custom ASIC takes the longest time to design as each component has to be exclusively designed. The functions that are difficult to be implemented using a semi-custom ASICs and standard ICs are implemented using full custom ASIC. The full custom ASIC improves the performance of the system and minimizes the chip size significantly, and by minimizing the chip size, there is less silicon used to manufacture the chip. The reduced chip size can increase fabrication yield and decrease the cost per chip if the manufacturing is done in high volume. The fabrication yield formula is given by [18],

$$Y = \left[\frac{1 - e^{-AD}}{AD} \right]^2 \tag{1}$$

Here A is the area of the chip and D is the average number of defects per square centimeter. The formula for number of die per wafer is given as,

$$N = \frac{\left[\pi \left(R - \sqrt{A} \right)^2 \right]}{A} \tag{2}$$

Here R is the area radius and A is the area of the die again.

There are additional costs in testing the good die, packaging them and then doing the final testing. There is lot of expertise required to design a full custom ASICs and also advanced computer hardware and software programs are needed to manufacture full

custom ASICs. The time to market for the full custom ASIC is very high. A full custom ASICs is designed such that there can be no changes made after it is manufactured.

4.3.2. **Semi-Custom ASICs.** A semi-custom ASICs is different from a full custom ASICs as the basic circuit blocks are already designed and saved in the CAD software libraries. The user has to select the components and libraries, place them on the circuit and interconnect them. The designing of the semi custom ASIC is done by using gate arrays, functional blocks, standard cells and analog arrays. The analog array is used for analog functions whereas the functional block is used for both analog and digital functions. A semi-customized ASIC can be modified for different applications after it is manufactured.

4.3.3. **Development Cycle of ASICs.** The initial stage in designing the ASICs involves an inspection of the set of functions needed to be performed by the system. It is then decided by the system engineer and the IC designer to segregate the functions to be performed by the regular standard component and the functions to be performed by the custom IC to be developed. After the functions are determined, it is then decided whether the function will be executed in analog or digital domain and ways to implement the function are identified. Depending upon the functionally selected, it is determined to either design a full custom ASIC or a semi custom ASIC. The Figure 4.6 shows the development cycle of an ASIC.

Initially, only logic cells and interconnects are present on the ASIC. The hardware circuit decided by the system engineer is configured on the ASIC. This hardware structure on an ASIC is generally irreversible which means the hardware of the ASIC is not reconfigurable. Once the hardware is loaded, the software code is finalized, depending upon the requirement of the system.

4.3.4. **ASIC Compared to FPGA.** The ASICs provide a lot of benefits of speed and accuracy for a particular system. The ASIC has lower power consumption as illustrated in Power/Gate of the ASICs versus the FPGA in [19]. A FPGA look-up table has 12- 14 times the delay compared to an ASIC [20]. Also the density of the ASIC is 45 times greater than that of an FPGA [20]. But in contrast to the advantages of an ASIC, it provides certain disadvantages. The ASICs have a higher non-recurring engineering cost and the high turn-around time. All the design manufactured on the ASIC

is done with the masks during the fabrication. These masks help to connect the logic cells to the metal interconnection to develop a particular gate to perform a specific function. The cost of mask is increasing due to complex sub-wavelength lithography technique, as given in [19]; the cost of mask is almost doubling at each technology node. There are applications that are company specific and require fewer of ASICs, thus the cost of

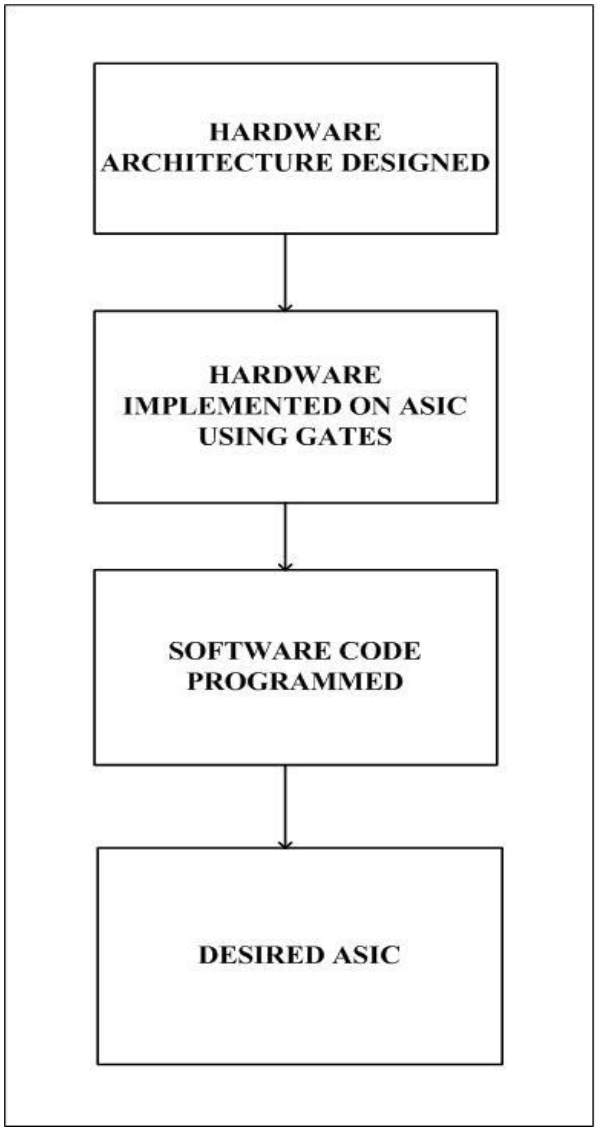


Figure 4.6: Design Cycle of an ASIC

the mask is divided by the number of ASICs required by the company. This makes the cost of a single ASIC comparatively higher. The ASICs have higher turn-around time as a single fabrication of an ASIC needs a time of around 2-5 months for preparing the mask and making the design; thus the ASICs take a lot of time to hit the market as opposed to the FPGA which is ready to be deployed in 1-4 weeks. The ASICs are used in very critical applications that have to process functions and data that cannot be processed using standard components available in the market [18]. It is a lengthy process to debug the ASIC during its fabrication time as the mask will have to be changed and the design will have to be re-fabricated. Thus the FPGA provides a significant advantage over the ASIC due to reduced non-recurring engineering cost as the hardware is reconfigurable unlike that of an ASIC. Also FPGAs are better suited for developmental projects like the MIMO transmitter. The FPGA also has a shorter turnaround time [19] for the varied modulation schemes to be implemented on the MIMO transmitter. The flexibility of the FPGA allows changing and modifying the modulation schemes without having to change a lot of parameters. The use of the FPGA in this system also balances the cost of the transmitter.

4.4. FPGA-BASED APPROACH

The FPGA consists of input/output blocks and programmable logic blocks. The input and output blocks of the FPGA are located on the periphery of the FPGA chip to establish connections with other chips on the board. The core of the FPGA consists of programmable logic cells and a programmable routing structure. Any hardware processor on the FPGA can be specified by programming in a hardware description language like VHDL or Verilog. The FPGA is created into a particular type of processor by programming the routing path that consists of switches and wires to connect the logic cells. Any design can be implemented on the FPGA by programming the logic cells and the routing path. The designing of the hardware is reconfigurable and is done on field and not during the time of fabrication hence it is known as a Field Programmable Gate Array [21]. Figure 4.7 shows the architecture of the FPGA. It shows the programmable logic

cells that are surrounded by metal interconnect. The input and output blocks are at the periphery.

The look-up table and D type flip-flop generally form the small configurable logic block of the FPGA. The look-up table can be conveniently used to form any number of combinational inputs, and the input of the look-up table is directly fed or stored in the delay flip-flop. An N-input look-up table allows N variables to be processed. The small look-up table increases the logic execution due to which the delay in the circuit increases. Additionally the programmable logic cells of the FPGA may also contain the macro cells that support the logic cells. Generally the macro cells are fabricated at the transistor level and are used to form the memory, combinational multipliers, and I/O interface.

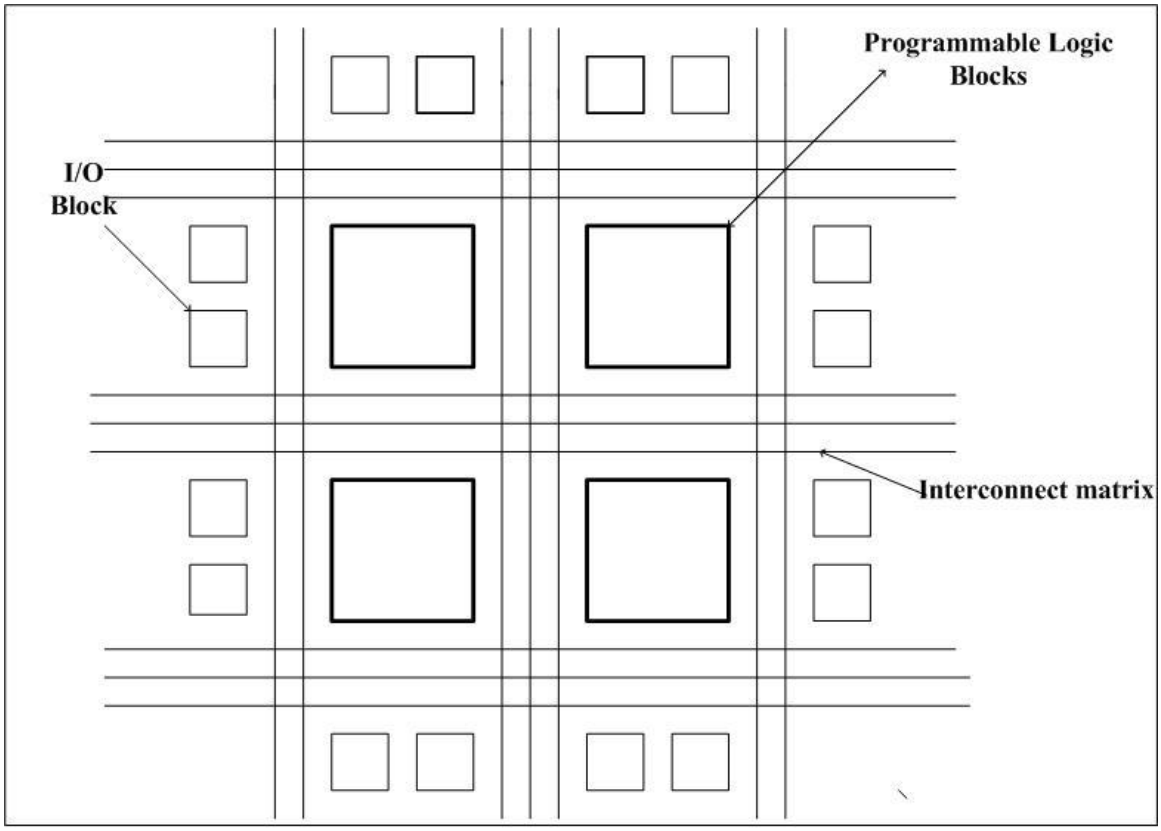


Figure 4.7. Structure of FPGA

4.4.1. **Technology in Programming the FPGA.** The FPGA is configured by programming the switches in the metal interconnect. These switches connect the wires in the metal interconnect and also different logic element blocks to form a particular logic. There are three technologies used in programming the switches in the FPGA; they are fuse technology, SRAM technology and FLASH technology. In the fuse technology, the fuse is a metallic link that can be programmed. If the FPGA is in the un-programmed state the fuse has very high resistance such that it can be considered as an open circuit. The programming of the fuse can be done by applying very high voltage and current to the input of the FPGA. The high voltage and current cause the fuse to lose its resistance and become connected. Once the fuse is wired the process cannot be reversed so this is a one-time process and debugging is not possible. The fuse technology is considered to be faster than SRAM and FLASH technology [22].

The Static RAM technology used in FPGAs works on a principle that once a valued is stored in a SRAM it cannot be modified or changed until it is re-programmed. In the SRAM technology the logic cells are made up of SRAM multi-transistor storage elements which drive the control transistors. The control transistors give an output of one or zero to the switch depending on their input from the SRAM transistors. The switch is on at one and is off at zero. The SRAM technology is the most widely used technology in the FPGA in the world [23] but it has its own disadvantages; it needs to be re-programmed once the power is turned off. Also five to six transistors form a single storage element. The ROM can be used to store the boot-up program of the SRAM technology. The third technology is the FLASH technology; this technology is replacement of EEPROM technology.

4.4.2. **Design Cycle of FPGA.** The design cycle of the FPGA Cyclone II is explained in great detail in the next chapter. The basic design cycle of the FPGA is demonstrated in Figure 4.8. It has a similar design cycle to that of an ASIC but the hardware circuitry is never permanent. The hardware can be reconfigured any number of times.

4.4.3. **Suitability for MIMO Transmitter.** FPGAs are used extensively for DSP-based applications [23]. The DSP chips are generally used for low rate application whereas for better performance and high rate applications ASICs are more suitable. The

FPGA provides a parallel between the DSP and ASICs. It also provides design flexibility and modification of design after development, generally not possible in the ASIC. The FPGA design also conserves the board space and system power unlike in a DSP processor. The FPGA is a favorable processor when the design of a particular system is modified frequently, and also the adaptability of a new design on the system is an issue. The FPGA offers a parallel multiplier structure, which provides the high speed necessary for this application compared to the processes in DSP that take place sequentially on each clock cycle. The FPGA offers a wider bandwidth than the DSP processor and also a high speed computation of the data.

The FPGA is reconfigurable, thus any amount of modification can be implemented easily without changing the actual hardware on the board. Also it requires less knowledge to build a complex DSP application on the FPGA than on an ASIC thus a group of less experienced people can also operate on the FPGA comfortably [23]. The newer technology in FPGA is making the FPGA a very suitable processor for signal processing applications. The new FPGAs offer low power consumption, high input and output bandwidth, internal high performance clock management for high speed of data [22] and are thus suited for the current project of developing a MIMO transmitter.

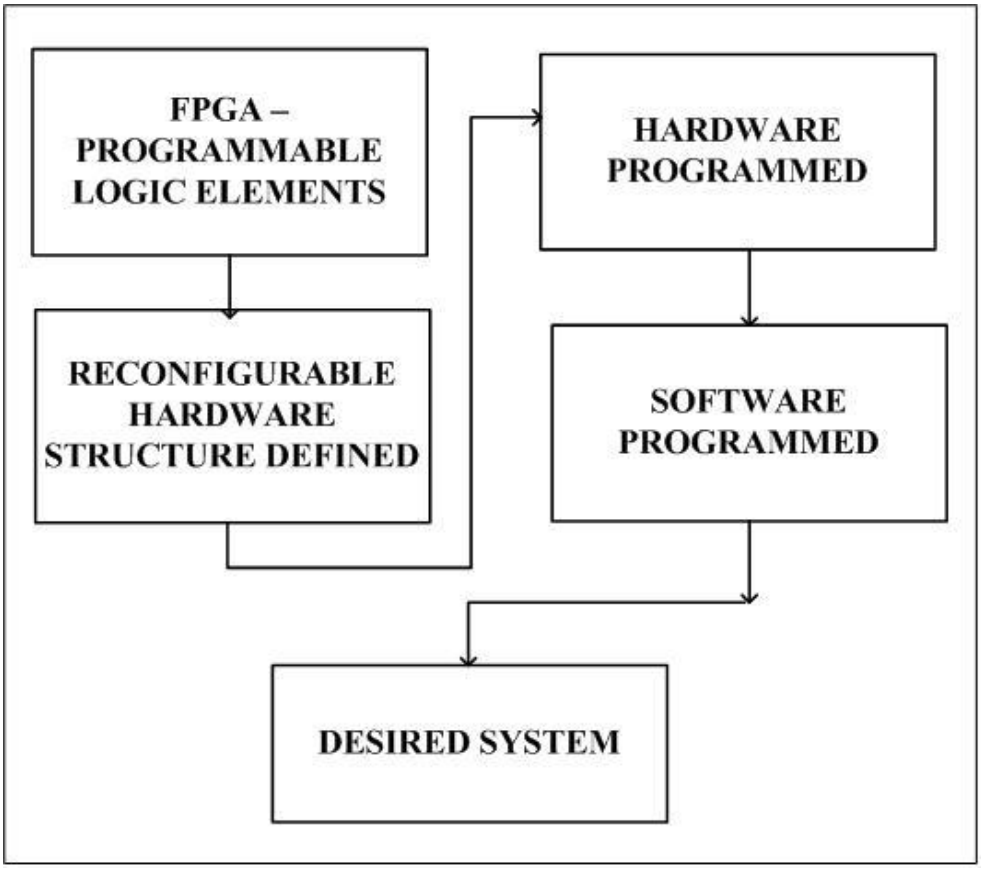


Figure 4.8 Design Cycle of the FPGA

5. DETAILED MIMO TRANSMITTER DESIGN

5.1. FIELD PROGRAMMABLE GATE ARRAY

The first block of the designed transmitter is the processing unit of the transmitter. The field programmable gate array is chosen to be the central processing unit of the transmitter due to its capability to offer high data rate throughput and reconfigurable hardware and software. There are two leading companies in the market that offer FPGA solutions, Altera and Xilinx. The Altera Company offers a large variety of FPGAs ranging from the high-end FPGA of the Stratix family to the mid-range FPGA of the Arria Family. A very high end processor is not needed to build this developmental project so the Cyclone family of FPGA is selected. The Cyclone II FPGA is a low-cost, low-power FPGA that is suited for applications that need to maintain benefits of low cost and low power consumption. The power consumption of Cyclone II is half compared to other 90 nm FPGAs available in the market. The graph in [24] illustrates the comparison of Cyclone II FPGA in terms of power consumption with other FPGAs available in the market.

The Cyclone II FPGA is based on 90 nm CMOS process technology. Cyclone II FPGA is used for manufacturing products in high volume due to its low cost. There are inbuilt 18 X 18 multipliers in this FPGA for parallel processing. There are different types of Cyclone II available of which the EP2C8T144 version is selected. The EP2C8T144 has 85 input and output pins that are suitable for the requirement of this project. There are a total of 28 pins required to transmit the 14-bit I signal and 14-bit Q signal. The additional of four input and output pins is required to program the micro-wire interface in the phase locked loop to provide data, clock, and latch enable. The WRT and CLK commands of the digital to analog converter are also programmed by FPGA. Table 5.1 shows the detailed features of the EP2C8 Cyclone II FPGA [24].

The Cyclone II FPGA requires two power sources in the circuit: one is 1.2 V for VCCINT and other of 3.3V or 2.5 V (user configurable) for VCCIO. The logic element of the EP2C8T144 consists of four look-up tables along with registers. The Cyclone II FPGA can support input and output data rates of 622 Mbps. The MIMO transmitter

Table 5.1. EP2C8T144 Basic Characteristics

Logic Element	8256
RAM Blocks	36
Total RAM bits	165888
Embedded multipliers	18
PLL	2
Maximum I/ O pins	85

requires a data rate of 10 Mbps which is achieved by using Cyclone II. The FPGA outputs I and Q baseband signals to the DAC. There is a need of at least 10 samples per symbol from the FPGA so the FPGA should be clocked at 100 MHz by an external crystal oscillator.

There is extensive intellectual property support available to the Altera FPGA. It is very time consuming to build the entire design of the FPGA from scratch. The time to market is an important consideration and one of the biggest advantages of the FPGA. The FPGA vendors provide intellectual property blocks (IP); these are a few functional blocks that are repetitive to almost all the hardware circuit design. These designs can be directly used in the FPGA, thus there can be more time devoted to the more critical parts of the design. The Cyclone II processor has a large variety of IP available like the NIOS II processor, PCI compiler, Viterbi compiler, etc.

The NIOS II soft embedded processor is one such intellectual property block supported by Cyclone II. More than one NIOS II processor can be configured on the FPGA. The NIOS II processor is completely user configurable. The term user configurable here means that the NIOS processor has many features that can be altered, removed or added based on the system requirement of price and performance [25]. There are three versions of NIOS II supported on the Cyclone II FPGA and they all use the same set of instructions, thus making them compatible with each other. The Quartus II software compiler is used to burn the NIOS II processor on the FPGA. The software

programming of the FPGA can be done using the NIOS II embedded processor software tool. The way to assemble a FPGA to make it into the processor of our system can be explained by this block diagram in Figure 5.1 [24].

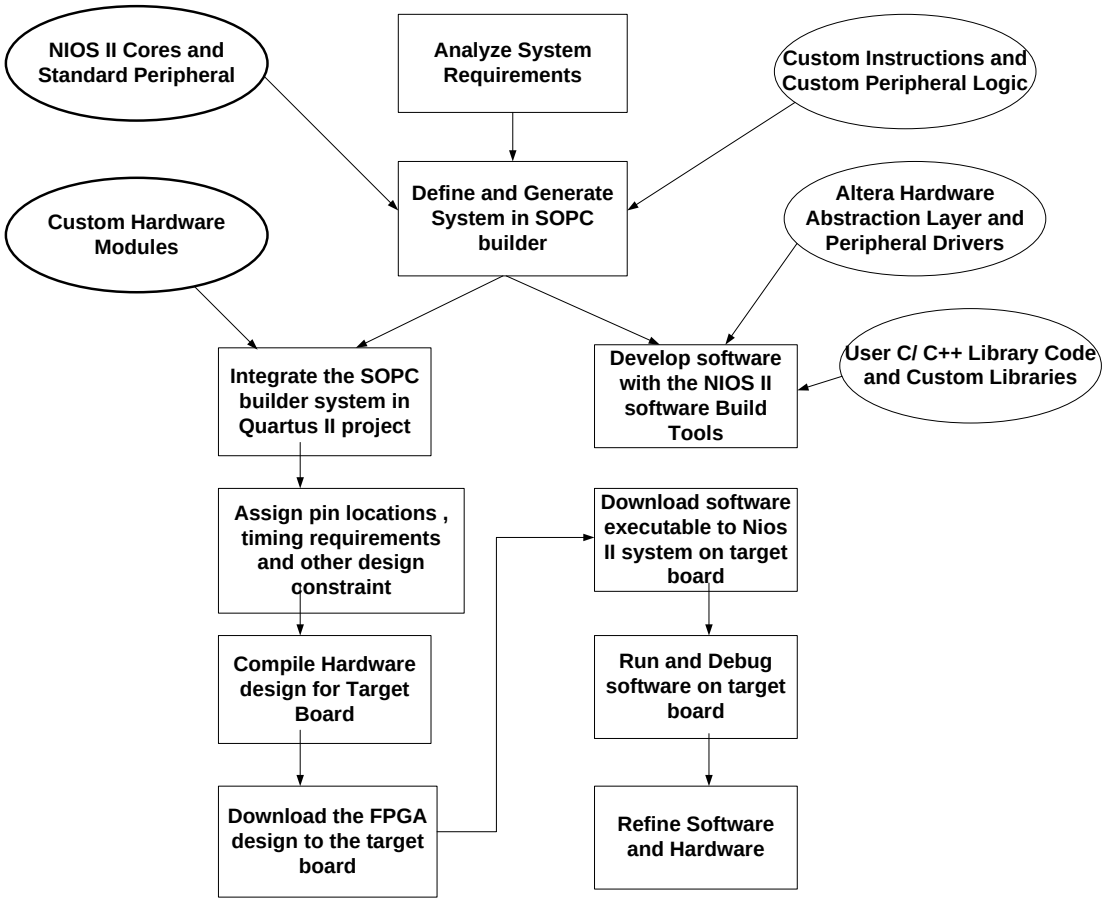


Figure 5.1. Development of NIOS II Embedded Processor

The first step is to analyze the system requirements like the interfaces, components, throughput, the real-time operating system, etc. Once the system requirements are analyzed the SOPC builder is used to specify the NIOS II processor, the memory and other components used by system. The NIOS II processor is selected from the three typical standard NIOS II processors available in the Cyclone II. The NIOS II

processor can also be complimented with additional feature like a different processor used in past, etc., to improve the system performance. The output of the SOPC builder is the hardware description language file that defines the hardware structure of the FPGA. It also outputs a human readable SOPC information file and hardware content of SOPC builder system. After a NIOS II system file is generated by the SOPC builder it is integrated with the Quartus II. The Quartus II is used to assign the pin location and the clock requirement of the circuit. The program on Quartus II is complied to form the .sof file, the .sof file is downloaded on the FPGA in the target system using the AS interface or the JTAG interface. The programming done by the JTAG interface is volatile and the FPGA has to be re-programmed again if the power is turned off, whereas the AS interface programming stores the program in 4 Mb Flash Memory. After the program is downloaded on the chip the FPGA behaves as the processor that is compiled by the Quartus II. The NIOS II software tool has peripheral drivers and a hardware abstraction layer so the programs can be written in high-level language; it also provides a debugging facility for debugging the software errors. The software for this processor can be written in the C/ C++ language using the NIOS II software build tool. Thus by using the above method the desired baseband signals can be generated by the FPGA. The output of the FPGA is fed to the digital to analog converter.

5.2. DIGITAL TO ANALOG CONVERTER

The digital to analog converter is the second block in the circuit. The DAC receives the digital signals from the FPGA and converts them into analog signals. The digital to analog converter is basically used as a bridge between the digital processor and the analog circuit components. The DACs takes the binary inputs and outputs an analog value. There are various DACs available in the market but all the DACs work on the concept of weighted current or voltage. There are switches to input the binary value. The DAC can be best understood by the basic example of R to 2R model of a DAC [26]. The Figure in [27] shows the basic R-2R model of a DAC. The operational amplifier in an inverting mode and has a gain of one. There are electronic switches like the transistors;

these transistors input the N-bit parallel data in the DAC circuit. There is a reference voltage source required to convert the data from digital to analog. This reference voltage can be settled to various different values like five volts, three volts or ten volts, depending upon the requirement of the circuit. The accuracy of the DAC circuit is dependent upon the stability and precision of the reference voltage.

The reference voltage if not properly stabilized can result in wrong values from the DAC. The binary values are whether one or zero, in other words, either high or low. If the values are low then the output of the DAC is equal to zero as the switch is not connected, however there may be some output due to the internal offset voltage of the operational amplifier or the noise. The output voltage of the DAC is given by the product of current and the resistor, thus if all the binary inputs are set to zero there is no current in the circuit. Generally an output amplifier is connected to the DAC to have low impedance at the output, However if there is no output amplifier connected, the output generally has a high impedance. The output voltage of the DAC is given by equation 3.

$$v_{output} = v_{reference} * \frac{A}{2^N} \tag{3}$$

Here, $v_{reference}$ = the reference voltage provided

A = decimal value of the binary input

N= Number of bits in applied binary input.

The MIMO transmitter system has 14 bit digital I and Q signals from the FPGA that is fed in the DAC. There is a need of a DAC capable of working at 100 Mega samples per second as per the requirement of the system to provide a high data rate to the system. The DAC used in this application should be able to process I and Q signals separately, thus a dual DAC could fit in as the solution. The DAC 2904 is selected as the digital to analog converter for this system. There are two converter channels in this DAC; each channel has its own WRT and CLK line controlled by the FPGA. The WRT line controls the input latches in the DAC and the CLK line controls the DAC latches. The rising edge of the WRT line loads the data in the input latch and the falling edge of the WRT line presents the data to the DAC latch. On the immediate next rising edge of the

CLK line the new input data is presented to the DAC and is converted into analog data. It is very important to control the WRT line and CLK line using the FPGA accurately.

The output full-scale range of DAC 2904 is 2mA to 20mA and the output impedance is 200K ohms. The reference voltage of this DAC can be set at any voltage from 3.3 volts to 5 volts. The power dissipation is comparatively lesser than its counterparts, it is around 310mW.

DAC 2904 uses current steering logic to achieve a high update rate [28]. In current steering technology the current is the basic analog unit, hence the output of this DAC 2904 is differential current output.

5.3. QUADRATURE MODULATOR

Modulation is a process of modifying the signals to transmit them without interference in a particular medium. The baseband signals generated are of a very low frequency and cannot be directly transmitted over a channel. The low frequency signals increase the size of antenna used for transmission [29]. The signals are modulated with higher frequency signals that are known as the carrier signal to increase the frequency range.

There are three ways to modulate the digital signals; they are amplitude shift keying, frequency shift keying, and phase shift keying. In amplitude shift keying the amplitude of the carrier signal is varied according to the digital signal. In frequency shift keying the frequency of the carrier signal is varied according to the transmitted digital signal. In phase shift keying the phase of the carrier signal is varied according to the transmitted digital signal. The amplitude shift keying is the simplest of all the modulation schemes. Also it is the most inefficient way of modulating any signals and so is rarely used.

Quadrature amplitude modulation is a variation of amplitude modulation and phase modulation. It is a widely used modulation scheme because it allows additional signal to be transmitted without increasing the bandwidth of the system. In the QAM modulation scheme, two signals are used: the I signal and the Q signal. The I and Q

signals are 90 degrees out of phase with each other. A better description on I and Q signals is shown in Figure 5.2 [30]. If there is a signal S, s11 is I component of the signal and s12 is the Q component of the signal [30].

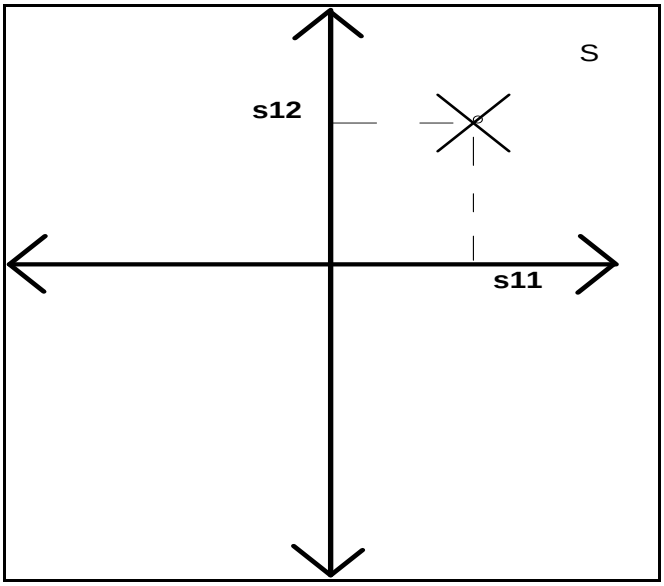


Figure 5.2. I and Q Signals

The I projection of the signal is on the X axis and the Q projection of the signal is on the Y axis. The QAM modulation is considered to be better than other modulation schemes as it provides modulation without increasing the bandwidth. The QAM has found application extensively in digital radio. The quadrature modulator consists of two mixers and a summation circuit. The baseband signal I and Q are fed in the quadrature modulator from the digital to analog converter. The I signal is multiplied by the sine signal whereas the Q signal is multiplied by the cosine wave. The high frequency used to modulate the baseband signals is provided by the phase locked loop. The quadrature modulator internally phase shifts the local oscillator frequency (the high frequency) by 90 degrees to provide the quadrature component. The final output of the quadrature

modulator is the summation of I and Q components of the signal. The final summation of the quadrature modulator is given by [30] as,

$$sum = I * sin\ \omega t\ + Q * cos\ \ \omega t \tag{4}$$

The block diagram of the quadrature modulator is shown in Figure 5.3 [31].

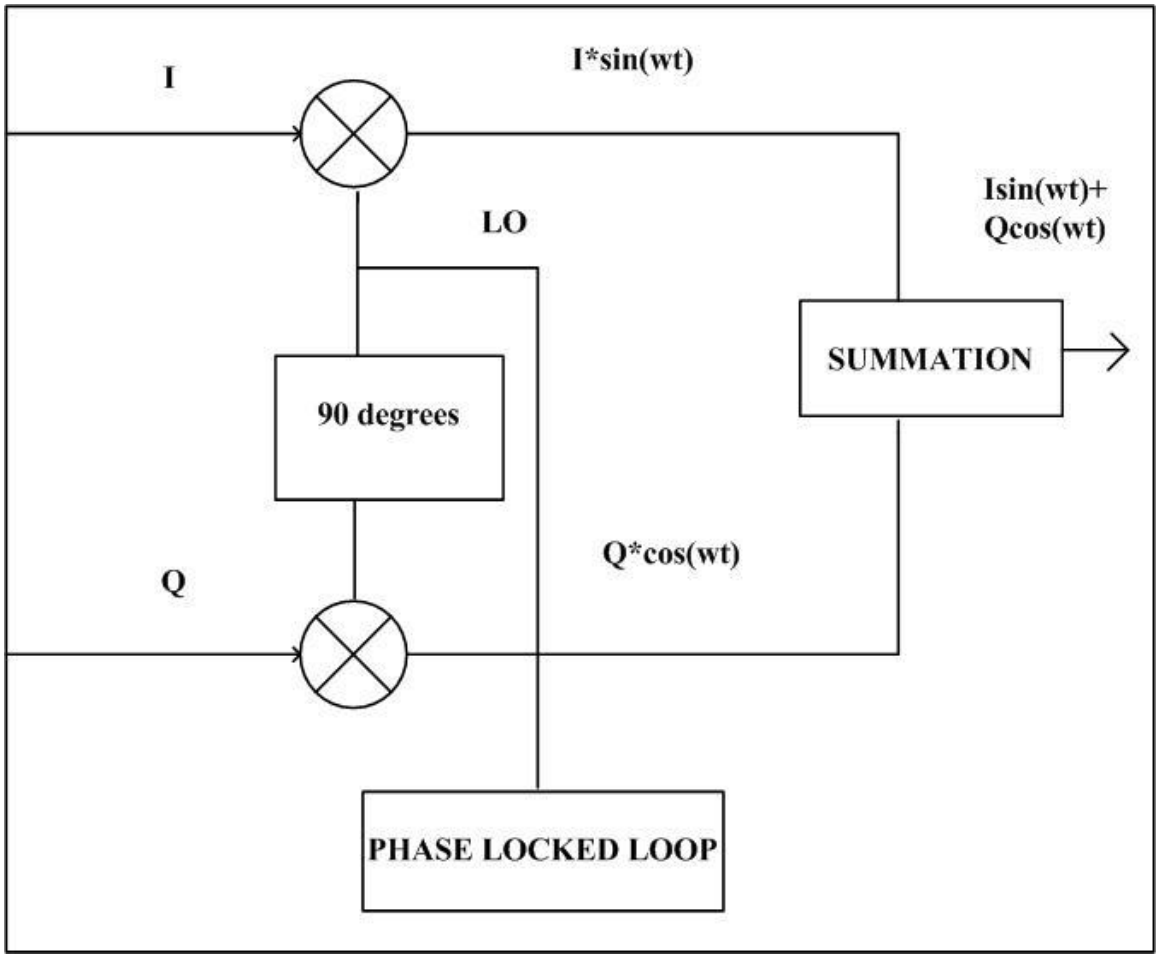


Figure 5.3. Basic Quadrature Modulation

The quadrature modulator selected for this application is Linear Technology 5572. It has an output impedance of 50 ohms. The output impedance of 50 ohms is necessary for load matching with the RF amplifier. The internal impedance for a LO frequency of 2.4 GHz is 47.3-j11.3 [32]. This quadrature modulator can operate on frequencies from 1.5 GHz to 2.5 GHz. It receives the local oscillator frequency of 2.4 GHz from the phase locked loop.

The input to this quadrature modulator is the baseband differential signal and the output is a single RF signal. The output signal from the quadrature modulator is fed in the RF amplifier for amplification and then terminated in the antenna for transmission.

5.4. PHASE LOCKED LOOP

The phase locked loop (PLL) is used for modulating and demodulating the signals. The PLL is used to provide a stable frequency from a reference frequency. Figure 5.4 [33] illustrates the block diagram of the PLL.

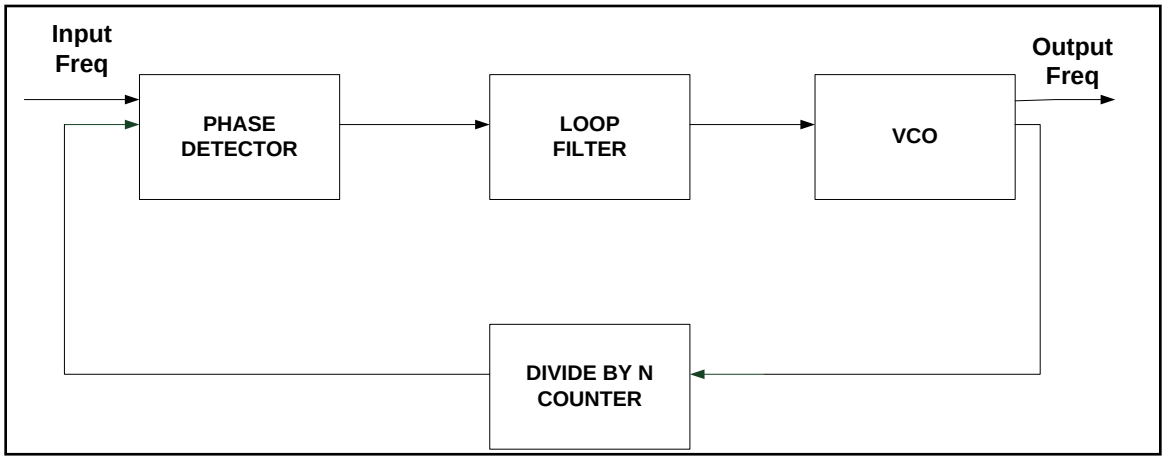


Figure 5.4. Block Diagram of a PLL

The PLL works on one basic principle given in [33]. There is a feedback loop in the PLL that maintains the stable voltage controlled oscillator frequency. Figure 5.4 is

block diagram of a basic PLL multiplier circuit. The phase detector compares the phase difference between the input frequency provided by the external source and the feedback frequency from the VCO and outputs the voltage corresponding to the phase difference. The frequency is nothing but the derivative of the phase so if there is a lot of difference in the phase of the two inputs that means that the PLL is not working on the correct frequency. If there is a mismatch the VCO is adjusted according to the phase difference obtained from the phase detector. The output of the phase detector is passed to the loop filter that determines the characteristics of the PLL in terms of stability. The loop filter is used to reject the high ripple content in the output of the phase detector. It is also used to attenuate the signals received from the phase detector and band rejects the high frequency content. The output of the VCO is actually N times the reference frequency and hence the output is divided by N to receive a comparison input for the phase detector.

The LMX2531 PLL is used in this system due to its capability to provide with frequency of 2.4 GHz [34]. The PLL is programmed by the FPGA; that gives the data, latch enable, chip enable and clock to the PLL. The PLL consists of an ‘Osc’ input that connects to an external crystal oscillator providing the input of 10MHz. The input of 10MHz is passed through an 'R' divider block to obtain the phase detector frequency. The R divider can be anything between 1, 2, 4, 8, 16, and 32. If 'R' is selected as 1, the highest phase detector frequency is obtained that is beneficial in reducing the noise in the PLL, for every twice the highest frequency the noise immunity increases by 3dB. Also the noise from the sigma delta modulator can be reduced with high phase detector frequency, but the phase detector frequency should be maintained at 2.5 MHz for maintaining an optimum spur performance. The R is selected as 4 to make the phase detector frequency equal to 2.5MHz. The VCO multiplies the phase detector frequency by N. Now the frequency required is equal to 2.4 GHz and the phase detector frequency is 2.5 MHz so the N is set to be 960.

$$F_{PD} = \frac{F_{OSC}}{R} \tag{5}$$

$$F_{VCO} = \frac{F_{PD} * N}{R} \quad (6)$$

There are two loop filters available on the PLL (one is external) and there is also a partially integrated internal loop filter present on the PLL. The value of the internal loop filter in LMX 2531 can be fixed using the micro wire interface using the FPGA. The attenuation of the loop filter can be minimized or maximized by using the different values for resistor and capacitor. The maximum value of attenuation can be achieved by setting the values of R at 40K ohms and C at 100pF and the minimum value is obtained by setting them to zero. The internal loop filter is present on-chip and is helpful to reduce the spurs more effectively than the external loop filter; hence it is recommended to keep the values of internal loop filter such that maximum attenuation is achieved. The registers of the PLL is programmed by the FPGA using the micro wire interface. The LMX 2531 consists of temporary shift registers that store the data and address of the internal registers. The latch enable is low at first so the data is fed in the shift registers during the rising edge of the clock. Then the latch enable goes high and the data is transferred to the internal registers. Thus the phase locked loop is able to provide with the fixed frequency of 2.4GHz needed to modulate I and Q signals in the quadrature modulator.

5.5. RF AMPLIFIER

The RF amplifier used in this circuit is the ERA 33SM+ which uses the Darlington pair of transistors to provide the required amplification. Figure 5.5 shows the basic Darlington configuration. The first transistor is in the common collector configuration and the second transistor loads the first transistor. The collector of both the transistors T1 and T2 are connected to each other.

The emitter of T1 is connected to the base of the transistor T2 [35]. The base of T1 becomes the base of the pair and the emitter of T2 becomes the emitter of the pair. The emitter current of T1 is the base current of T2. Firstly the emitter current of T1 is amplified by $\beta 1$ times the base current of T1 to achieve the amplification. Then this

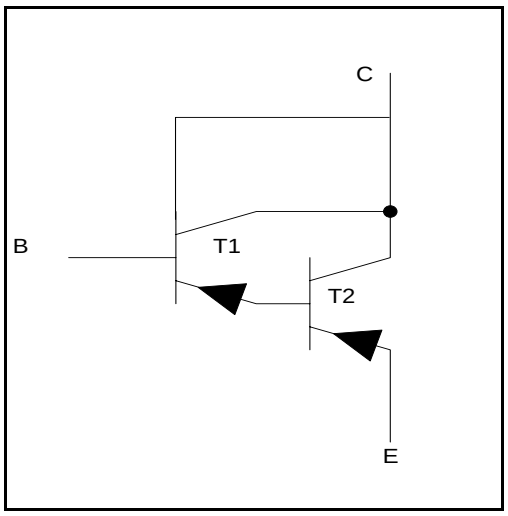


Figure 5.5. Darlington Pair of Transistors

emitter current of T1 is amplified by β2 again to obtain the emitter current of T2. Thus the total amplification is equal to the product of amplification of T1 and T2.

$$\beta_{total} = \beta 1 * \beta 2 \qquad (7)$$

The RF amplifier can amplify the signals between DC to 3 GHz and also has a very high reliability. The input in the RF amplifier is received by the quadrature modulator and the output is fed into the antenna.

5.6. TESTING OF THE BOARD

To verify the design, and also to help debug the coding and modulation algorithms to be analyzed, a number of test features will be built into the base modules. The FPGA portion of the design should be reasonably low risk. Some of the unused general purpose I/O pins will drive LEDs, which can be toggled by the user to indicate

when particular portions of the code are executed. There will be a number of low impedance outputs which drive the DAC, which will be brought to headers to allow easy connection of a logic analyzer. In addition, the LSB and MSB of these outputs will drive LEDs, to allow the user to quickly observe when there is activity on the output data busses.

The FPGA is to be connected to the DAC through jumpers, which could be removed so the user could inject signals directly to the DAC. However given the number of digital lines involved and the lower reliability of jumper connections in the final design, it was decided not to use that approach. Once the design has been validated, the LEDs can be removed from future implementations to help reduce the power consumption.

The DAC can potentially inject a wide variety of imperfections that could impact system performance, including non-linearity and timing jitter or delays. To measure these effects, the DAC outputs will be brought to test points which will allow the connection of oscilloscopes, spectrum analyzers, and other test equipment. During initial testing, the FPGA will be programmed to generate sinusoids at the DAC outputs. Imperfections such as nonlinearities can then be measured through harmonic analysis of the output signals. The impedances will be selected so that it will be possible to also inject signals at this point. Should the FPGA be unable to generate the signals required to test the remainder of the module, the DAC will be disabled and signals injected at the DAC outputs.

In previous student projects, the phase locked loop has proved to be a challenging device to work with. The circuit design for the PLL chip is reasonably straight forward. However during power up the device must be programmed, so that the output frequency and power level will fall in the correct range for the quadrature modulator. To help observe, diagnose, and potentially overcome these problems, the serial communication link that connects the FPGA to the PLL will pass through headers/jumpers, and to test points. This will allow a logic analyzer to be connected to the FPGA output, to observe the sequence of programming bits sent to the PLL. Should the FPGA fail to generate the proper signals, the jumpers will be removed, and an external logic source will be used to program, or reprogram, the PLL. The output of the PLL will be passed through a power splitter, with one of the outputs going to a test point, and the other to the modulator. This

test point can be used to confirm the power level, and frequency, of the PLL output – along with measuring phase jitter, and other imperfections in the carrier signal.

The quadrature modulator device cannot be probed; however its output will be split between the power amplifier, and a test point. During testing, constant levels will initially be applied to the I and Q, signals to verify that both sides of the quadrature modulator are functional. This will be followed by sinusoidal signals on both I and Q, to essentially perform single-side-band modulation. Phase and gain imbalances can then be measured by observing the unwanted sideband at the modulator output. Finally, BPSK and QPSK signals will be generated, and their time domain and spectral domain signatures analyzed at the modulator output.

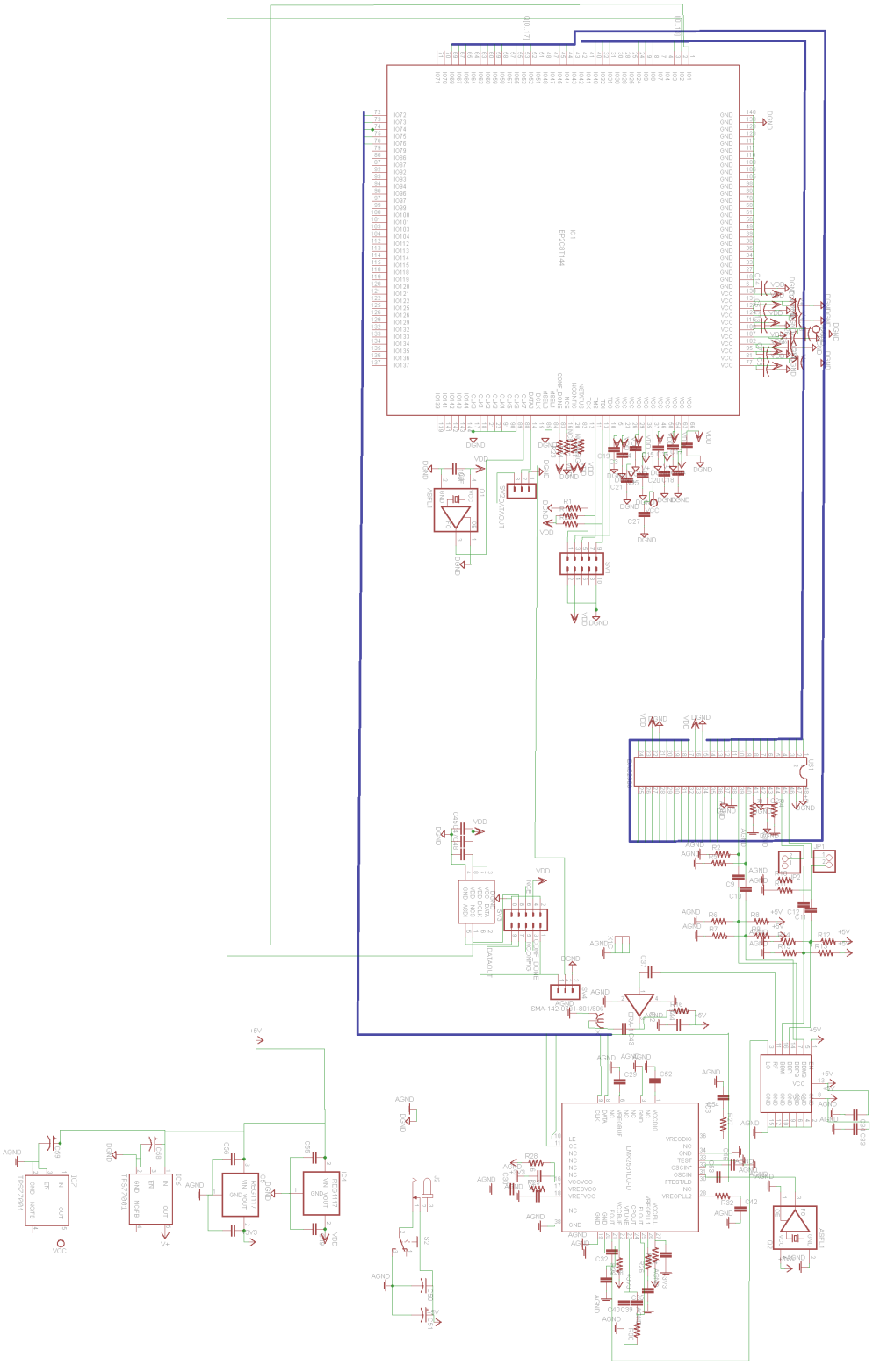
Once the modulator output has been validated, the final test will be to observe the output of the RF power amplifier. Initially sinusoidal signals of varying frequencies and amplitudes will be generated by the modulator, to verify the power level of the output. In addition, spectral analysis will be used to measure the nonlinear characteristics of the power amplifier.

6. CONCLUSION

The Multiple-Input Multiple-Output transmitters are designed to improve the system performance and data rate. The MIMO antenna technology is one of the most upcoming technologies that will replace all the other antenna technologies in future. The use of FPGA in the given project is a smart decision considering the speed challenges and other performance parameters of this system. This thesis presented an architectural description of a modular MIMO test bed. Each module consists of a FPGA-based baseband processor, frequency reference, PLL, quadrature modulator and RF amplifier. The configuration of the modules will allow a system to be constructed as a series of SISO transmitters, a series of frequency-synchronized SISO transmitters, or a MIMO transmitter. The system is scalable, so that an arbitrary number of transmit antennas can be used. The future work would be to develop the hardware design proposed in this thesis and implement various different modulation schemes and space time code to test the system performance in different kinds of multipath environments. Also, the schematic design of a single module can be extended to include multiple transmitters and different MIMO configurations can be tested.

APPENDIX

SCHEMATIC DIAGRAM OF MIMO TRANSMITTER



BIBLIOGRAPHY

[1] D. Bates, S. Henriksen, B. Ninness, and S.R. Weller, "A 4x4 FPGA-based wireless testbed for LTE applications," *Proc. of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communication,* pp.1-5, 2008.

[2] Veena M.B, Raj C.P and Swamy. M.N.S, "FPGA-based reconfigurable 200 MHz Transmitter and Receiver front end for MIMO OFDM," *Proc. of Emerging Trends in Engineering and Technology,* pp.819-824, 2010.

[3] Krzysztof Wesolowski. *Introduction to Digital Communication Systems.* John Wiley & Sons, 2009.

[4] S. Moghe and R. Upadhay, "Comparison of SISO and MIMO Techniques in 802.11n Wireless Local Area Network," *Proc of Emerging Trends in Electronic and Photonic Device & Systems*, pp.245-246, 2009.

[5] S. Cherry, "Edholm's law of bandwidth," *IEEE Spectrum*., vol. 41, pp. 50, 2004.

[6] Thomas Kaiser, "When Will Smart Antennas Be Ready for the Market? Part I," IEEE *Signal Processing magazine*, vol. 22, pp. 87-92, 2005.

[7] A. Mietzer and P.A. Hoeher," Boosting the Performance of Wireless Communication Systems: Theory and Practice of Multiple-Antenna Techniques", *Communications Magazine*, vol. 42, pp.40, 2004.

[8] G.J. Foschini and M.J. Gans, "On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas," *Wireless Personal Comm*., vol. 6, pp. 311-335, 1998.

[9] E. Telatar, "Capacity of Multi-antenna Gaussian channels," *Euro. Trans. Comm*., vol. 10, no. 6, pp. 585–95, 1999.

[10] S.A. Bassam, M. Helaoui, F.M. Ghannouchi, "BER Performance Assessment of Linearized MIMO Transmitters in Presence of RF Crosstalk", *Proc. of Radio and Wireless Symposium (RWS)*, pp. 33-36, 2010.

[11] Y. Palaskas, A. Ravi and S. Pellerano, "MIMO techniques for high data rate radio communications," *Proc. of IEEE Custom Integrated Circuits Conference*, pp.141-148, 2008.

[12] Craig Steiner. *8051/8052 Microcontroller: Architecture, Assembly Language, and Hardware*. Universal Publishers, 2005.

[13] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of Wireless Sensor Networks.* Wiley, 2010.

[14] S.A. Ito and L. Carro, "A Comparison of Microcontrollers Targeted to FPGA-Based Embedded Applications," *Proc. of the 13th Symposium on Integrated circuits and systems design*, pp.397, 2000.

[15] Dobrivoje P. and Ljubo V. *Mechatronics in engineering design and product development.* CRC Press, 1999.

[16] Dake Liu. *Embedded DSP Processor Design Application specific.* Morgan Kaufmann, 2008.

[17] R.C. Restle, "Choosing between DSP, µPs, FPGA and ASICs to implement digital signal processing," *Conference Proc. of ICSPAT: DSP world*, Autumn 2000.

[18] Jerry Whittaker. *The Electronics Hand book.* Taylor and Francis Group LLC, 2005.

[19] P.S. Zuchowski, et. al., "A Hybrid ASIC and FPGA structure," *Proc. of the 2002 IEEE/ACM International Conference on computer-aided design*, pp.187-194, 2002.

[20] I. Kuon and J. Rose, "Measuring the Gap between FPGA and ASIC," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp.203-215, 2007.

[21] Pong Chu. *FPGA Prototyping by Verilog Examples: Xilinx Spartan*. John and Wiley, 2008.

[22] Clive Max Field. *FPGAs: Instant Access.* Newnes, 2008.

[23] G.R. Goslin, "A guide to use Field Programmable Gate Array (FPGAs) for Application-Specific Digital Signal Processing Performance," *Xilinx Corporation*, 1995.

[24] http://www.altera.com/products/devices/cyclone2/features/power/cy2-power.html.

[25] N. Zainalabedin. *Embedded Core Design with FPGA. McGraw Hill*, 2006.

[26] Joseph J Carr. *Linear IC Application, A Designers Handbook.* Newnes, 1996.

[27] U.A, Bakshi and A.P. Godse. *Analog and Digital Electronics*. Technical Publications, 2008.

[28] “DAC 2904 Data sheet,” http://focus.ti.com/docs/prod/folders/print/dac2904.html.

[29] E.F. Marvin. *Digital Signal Processing in Communication Systems*. Springer, 1994.

[30] “NI developer Zone - Tutorial,” http://zone.ni.com/devzone/cda/tut/p/id/4805.

[31] Avigdor Brillant. *Digital and Analog Fibre Optic Communication for CATV and FTTx Application.*SPIE Publications, 2008.

[32] “LT5572 Datasheet,” http://cds.linear.com/docs/Datasheet/5572f.pdf.

[33] Poole L., “Phase Locked Loop Basics,” *Radio Communications*, vol. 74, 1998.

[34] “LMX2531 Datasheet,” www.national.com.

[35] James C. *Fundamentals of Linear Electronics: Integrated and Discrete.* Cengage, 2001.

VITA

Richa Shekhar was born in India on January 4, 1987. She completed her high school from Mumbai and enrolled in Thakur College of Engineering and Technology, Mumbai University to complete her undergraduate studies. She received the Degree of Bachelor of Engineering in Electronics and Telecommunication Engineering in 2008. She joined the Electrical and Computer Engineering Department of Missouri University of Science and Technology to pursue her Master of Science in Electrical Engineering. She finished the degree requirements of course work in 2011 and will receive her Master's in Electrical Engineering in August 2011.