2010

Design and Implementation of an RF Front-End for Software Defined Radios

Mahadevan Balakrishnan

Design and Implementation of an RF Front-End for Software Defined Radios

(Spine title: Design and Implementation of an RF Front-End for SDRs)

(Thesis format: Monograph)

by

Mahadevan Balakrishnan

Graduate Program
in
Electrical Engineering
Electrical and Computer Engineering Department

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Engineering Science Degree

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Certificate of Examination

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES
CERTIFICATE OF EXAMINATION

Chief Advisor:

Dr. Abdallah Shami

Advisory Committee:

Examining Board:

Dr. Luiz Fernando Capretz

Dr. Abdelkader Ouda

Dr. Mike Katchabaw

The thesis by
Mahadevan Balakrishnan
entitled:
Design and Implementation of an RF Front-End for Software Defined Radios
is accepted in partial fulfillment of the
requirements for the degree of
Masters of Engineering Science Degree

Date: _____

Chair of Examining Board
Dr. Roger Khayat

Abstract

Software Defined Radios have brought a major reformation in the design standards for radios, in which a large portion of the functionality is implemented through programmable signal processing devices, giving the radio the ability to change its operating parameters to accommodate new features and capabilities. A software radio approach reduces the content of radio frequency and other analog components of the traditional radios and emphasizes digital signal processing to enhance overall receiver flexibility. Field Programmable Gate Arrays (FPGA) are a suitable technology for the hardware platform as they offer the potential of hardware-like performance coupled with software-like programmability.

Software defined radio is a very broad field, encompassing the design of various technologies all the way from the antenna to RF, IF, and baseband digital design. The RF section primarily consists of analog hardware modules. The IF and baseband sections are primarily digital. It is the general process of the radio to convert the incoming signal from RF to IF and then IF to baseband for better signal processing system.

In this thesis, some of major building blocks of a Software defined radio are designed and implemented using FPGAs. The design of a Digital front end, which provides the bridge between the baseband and analog RF portions of a wireless receiver, is synthesized. The Digital front end receiver consists of a digital down converter(DDC) which in turn comprises of a direct digital frequency synthesizer (DDFS), a phase accumulator and a low pass filter. The signal processing block of the DDFS is executed using Co-ordinate Rotation Digital Computer (CORDIC)

algorithm. Cascaded-Integrator-Comb filters (CIC) are implemented for changing the sample rate of the incoming data. Application of a DDC includes software radios, multicarrier, multimode digital receivers, micro and pico cell systems,broadband data applications, instrumentation and test equipment and in-building wireless telephony. Also, in this thesis, interfaces for connecting Texas Instruments high speed and high resolution Analog-to-Digital converters (ADC) and Digital-to-Analog converters (DAC) with Xilinx Virtex-5 FPGAs are also implemented and demonstrated.

Acknowledgements

My deepest gratitude goes to my supervisor, Dr. Abdallah Shami, for having accepted me as a graduate student and providing continuous support through my graduate studies. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank OCE and Jeff Musson from Broadband Wizard for providing the platform for the research work I have done.

I am indebted to many of my colleagues for providing a stimulating and fun environment. In particular, Chris for all his help and suggestions in this thesis, Dan for the technical discussions, motivation and initiative to help, Tomasz for his valuable time and support, Ayman for being a confidant and Oscar for all the fun moments.

I am fortunate to have Ajit, Indranil, Aditi and Rachita as my friends who have always been by my side and made my stay in Canada comfortable.

My heartfelt thanks to my roommate Arun who is no less than a brother to me.

I wish to offer my gratitude to my close friend Sneha for her valuable inputs, both academically and morally that has helped me in my thesis.

I am extremely thankful to my extended family for all their moral support. In particular, my aunt and my cousins who were very supportive.

Above all, I wish to thank my parents, my aunt, my sister and my brother-in-law from whom I have learnt a lot and owe my gratitude for making it all possible. I dedicate this thesis to them.

Table of Contents

List of Tables

List of Figures

Acronyms

ADC	*Analog to Digital Converter*
AGC	*Automatic Gain Control*
AI-SR	*Adaptive Intelligent Software Radio*
ASIC	*Application-specific integrated circuit*
BPF	*BandPass Filter*
CLB	*Configurable Logic Block*
CORDIC	*Coordinate Rotation Digital Computer*
CIC	*Cascaded Integrator-Comb filter*
DAC	*Digital to Analog Converter*
DCM	*Digital Clock Manager*
DDC	*Digital Down Converter*
DDFS	*Direct Digital Frequency Synthesis*
DDR	*Double Data Rate*
DSP	*Digital Signal Processor*
DUC	*Digital Up Converter*
ENOB	*Effective Number Of Bits*
FFT	*Fast Fourier Transform*
FIFO	*First In First Out*
FIR	*Digital Up Converter*
FPGA	*Field Programmable Gate Arrays*
FSM	*Finite State Machine*
GSM	*Global System for Mobile Communications*
HPA	*High Power Amplifier*
IF	*Intermediate Frequency*
LNA	*Low Noise Amplifier*
LO	*Local Oscillator*
LUT	*Look Up Table*
LVDS	*Low Voltage Differential Signaling*

MSB	*Most Significant Bit*
MSPS	*Mega Samples Per Second*
NCO	*Numerically Controlled Oscillator*
PA	*Power Amplifier*
PLL	*Phase Lock Loop*
RAM	*Random Access Memory*
ROM	*Read Only Memory*
RF	*Radio Frequency*
RX	*Receiver*
SDR	*Software Defined Radio*
SFDR	*Spurious Free Dynamic Range*
SINAD	*Signal-to-Noise Distortion Ratio*
SNR	*Signal-to-Noise Ratio*
SPI	*Serial Peripheral Interface*
SR	*Software Radio*
TDMA	*Time Division Multiple Access*
TX	*Transmitter*
VCO	*Voltage Controlled Oscillator*
VGA	*Voltage Gain Amplifier*

Chapter 1
Introduction

1.1 Motivation

Modern digital devices play a major role in today's world and it is important for them to deliver optimum performance. With this growing need for top digital devices, there is a huge demand for energy efficient wireless communication. More functions of a radio system are being performed via software, leading towards the concept of a software defined radio(SDR). An SDR can be defined as a transceiver in which most of the operations are performed using versatile reconfigurable hardware which are in-turn configured via software. The primary goal is to shift from employing hardware-focused, application-specific approach to radio implementation to using software application to perform the radio tasks on computing platform. The goal for this thesis is to implement a model that functions with a high degree of reliability, is reconfigurable, is efficient in terms of hardware cost, and is suitable for application in Software radio terminals.

Current advances in field-programmable gate array (FPGA) technology have enabled high-speed processing in a compact footprint, while retaining the flexibility and programmability of software radio technology. FPGAs are popular for high-speed, compute-intensive, reconfigurable applications (fast Fourier transform (FFT), finite impulse response (FIR) and other multiply-accumulate operations). Reconfigurable cores are available from FPGA and board vendors (Xilinx and Altera) and enable

implementation of modulator, demodulator and CODEC functionality in the FPGA. System designers are increasingly looking for front-end acquisition/converter products with integrated FPGA to offload the baseband processing and reduce data transfer rates.

1.2 Implementation - Big picture

The initial implementation in this thesis is a Digital Front-end (DFE) design which basically reduces or upgrades the Intermediate frequency of the incoming signal to a lower or upper frequency for processing depending on the application. A general idea about the DFE is shown in Figure 1.1 As shown in the figure, the DFE is

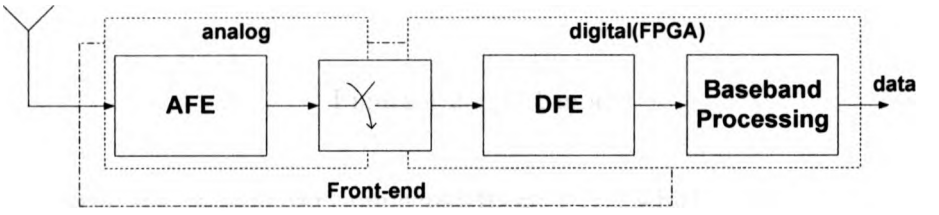


Figure 1.1: Concept of Digital Frond End

implemented within the FPGA to perform digital modulation of the signal coming from the Analog front-end (AFE) which primarily comprises of the RF transceiver. The DFE is employed here in the thesis to provide the user with an additional feature of reconfigurability in terms of frequency translation and sample rate conversion as per the requirements of the application in hand. The FPGA used for the DFE design is Altera Stratix II.

In this thesis, the SDR architecture is implemented for a multi-purpose reconfigurable platform for wireless communication technologies using the 802.11 b/g

wireless LAN protocol. The experimental architecture layout of the SDR adopted in this thesis is shown in Figure 1.2 As shown in the figure, the receiver architecture

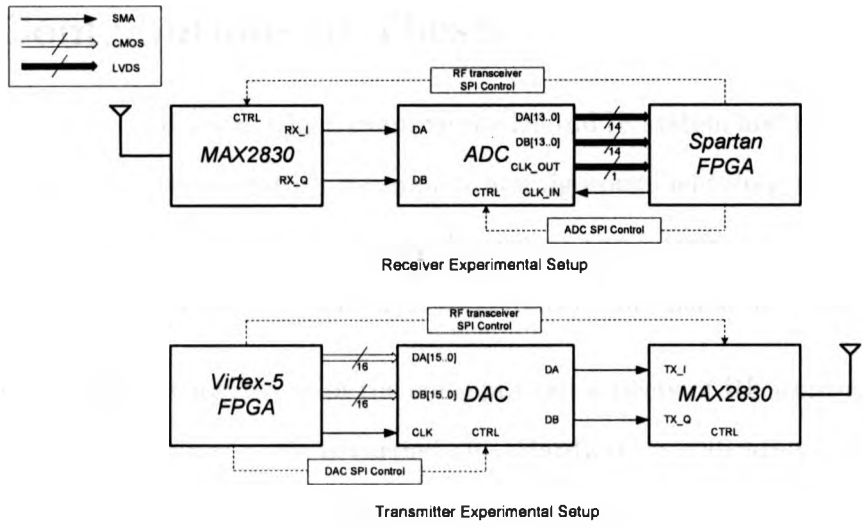


Figure 1.2: SDR model

consists of the RF transceiver connected to an ADC which is then connected to an FPGA while the transmitter has the FPGA connected to a DAC that goes to the RF transceiver. To realize the concept of an SDR, the FPGA controls the functionalities of the ADC, DAC and RF transceiver via software control. An interface has to be built to connect the high speed ADC and DAC with the FPGA in order to have a proper data flow. The interface used for the ADC is Double Date Rate Low Voltage Differential Signaling (DDR-LVDS) and CMOS is used for the DAC. The RF transceiver chosen for this design is the MAX2830 from Maxim-IC which adopts the Wi-Fi 802.11 g/b protocol. The ADC and DAC used here are ADS62P43 and DAC5687 evaulation boards from Texas Instruments respectively. The ADC board is interfaced with a Spartan 3A FPGA since it has the appropriate LVDS headers. The

DAC board is interfaced with a Virtex-5 FPGA because the CMOS headers in the two boards were compatible.

1.3 Contributions of Thesis

While there are several works which examine the Digital IF system and the integration of Data converters in the literature, a complete and rigorous Field Programmable Gate Array (FPGA) implementation of a Digital IF system and Interfacing with ADCs and DACs has not yet been documented. The contributions are listed as follows:

- A detailed literature review of the concepts and aspects of RF front-end design of a Software defined radio is carried out. Hardware specifications of the components involved in an SDR are discussed and a survey of the available signal processing hardware is done. The basic considerations of an SDR receiver and transmitter are pointed out. The different architectures that have been adopted to implement the SDR and the problems involved in them have been reviewed.

- Detailed design and explanation of the Digital Front End (DFE) comprising of the Digital Up-and Down-Conversion and the FPGA implementation of these systems is illustrated. It demonstrates the implementation aspects and performance of a Digital Down converter and the Digital Up converter separately using a Direct Digital frequency synthesizer(DDFS). A modified hardware efficient multi-rate filter chain is employed for sample rate conversion.

- The LVDS interface, shown in Figure 1.2 is implemented in the Spartan FPGA to connect it with the high speed ADC. Likewise, the CMOS interface is implemented in the Virtex-5 FPGA to connect it to the DAC. The reconfigurable software controls shown in Figure 1.2 are also implemented in the FPGAs. These

control entities enable the necessary signal interactions between the FPGAs and the data converters via Serial Peripheral Interface (SPI) communication.

- SPI control entities for the RF transceiver(Figure 1.2) is also implemented and verified in the Spartan and Virtex FPGAs.

1.4 Software used

Although there has been a huge improvement of the application development tools, FPGA design should be considered as hardware development, and it requires a different skill set than software development. The general FPGA design flow for both Xilinx ISE design software and Altera Quartus II design software is depicted in Figure 1.3

The first step in the design process involves translating the design's specification to something that the tools can understand. This is done by HDL coding (AHDL, VHDL or Verilog), schematic design entry or by the design and integration of IP blocks or embedded processors or Digital Signal Processing (DSP) modules.

Once the design entry has been completed, the synthesis engine compiles the design to transform HDL sources into an architecture-specific design netlist.

The simulator is used to verify the functionality of a design (functional simulation), the behavior and the timing (timing simulation) of your circuit. Timing simulation is run after implementing the circuit in the FPGA since it needs to know the actual placement and routing to find out the exact speed and timing of the circuit.

After generating the netlist file (synthesis step), the implementation will convert the logic design into a physical programming file that can be downloaded on the target device (e.g. Virtex FPGA). This is carried out in the place and route module. The

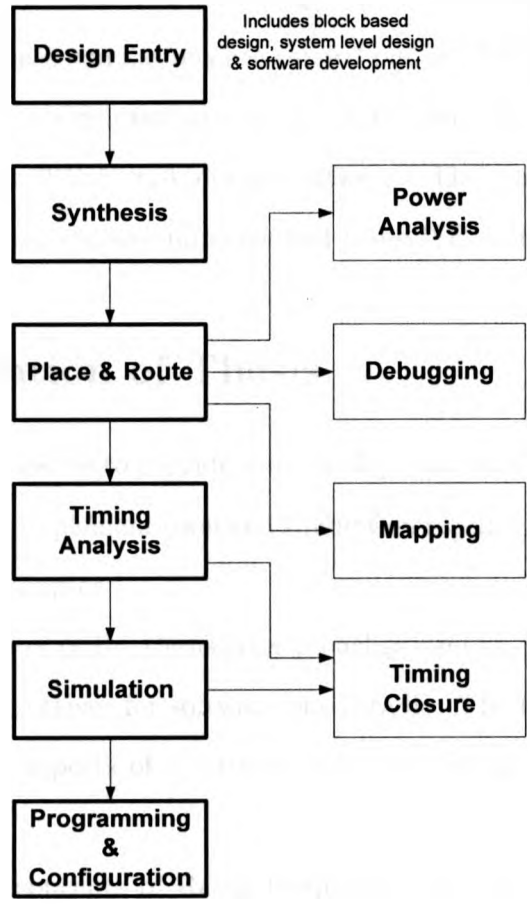


Figure 1.3: FPGA design flow

design constraints to specify timing, pin placement and other design requirements are entered here. The place and route also provides tools to map the resources, debug and target the target device's power saving architecture during the compilation process. Design performance optimization is dealt with by the timing analysis feature that uses the ASIC industry standard.

Finally, the target FPGA is programmed and configured by downloading the programming file from the host computer to the Xilinx/Altera FPGA.

The design entry used in this thesis is VHDL. The complete architecture of

the DFE is devised and verified in Altera Quartus II FPGA design software. The multi-stage filter response simulations are carried out in MATLAB.

The LVDS and CMOS interfaces for the ADC and DAC are implemented and simulated using Xilinx ISE FPGA design software. The controllers for the ADC, DAC and RF transceiver are also implemented using ISE design software.

1.5 Organization of Thesis

The structure of this thesis is to provide some background on Software radio concepts, followed by the design, implementation and simulation results. The focus of this thesis has been outlined in Chapter 1.

The second chapter of the thesis gives a background on software radio concepts and architectural perspectives for software based radio. The hardware specifications and the digital design aspects of a software defined radio are also discussed in this chapter.

The underlying concepts of Radio Frequency (RF) front end design and the basic considerations behind the design of a receiver/transmitter and RF to baseband conversion architectures are illustrated in Chapter 3.

The fourth chapter presents the design and implementation of the digital front end system. Some background on the design of a frequency synthesizer is provided and its implementation with a hardware efficient rotation algorithm is demonstrated. The design and implementation of decimation and interpolation filters, after the frequency conversion using a CORDIC DDFS, are also shown.

The fifth chapter deals with interfacing ADCs and DACs with FPGAs. This chapter illustrates both serial and parallel interface for connecting a Texas Instru-

ments ADC with a Virtex-5 FPGAs. Design of controllers for these interfaces are also described in this chapter.

The RF transceiver and its controller design is illustrated in Chapter 6. This chapter also depicts a prototype model of the RF front end design with simulation results.

At last, Chapter 7 gives the conclusion and recommendations for future work.

Chapter 2

Software Radio Concepts and Specifications

Primitive attempts to implement radio communication systems were entirely carried out using hardware elements. In hardware based systems, the focus for the implementations were based on the idea of supporting only one type of protocol. The physical layer of the hardware system was engrafted in specific hardware solutions and the Radio Frequency (RF) front-end was also designed for the same standard. The important blocks in such a system were the transmitter, receiver, power amplifier and some sort of encryption module for security purposes[1].

As communication technology progressed, more functions of contemporary radio systems were implemented in software, leading toward the Software Defined Radio(SDR). The SDR contains many processing elements that can be programmed to deliver the required functionality. The fundamental intent is to shift from employing a traditional hardware-focused, application-specific approach to radio implementation to using software application to perform the radio tasks on computing platform. Further it was realized that the incoming signal had to be tailored to satisfy the requirements of individual platforms. As a result, SDRs that can perform multiple tasks on multiple platforms came into picture. This chapter reviews the architectures of SDR and highlights the hardware specifications for designing a SDR system.

2.1 Software Radio Architectures

In an ideal world, a Software Defined Radio (SDR) would be able to transmit and receive signals of any frequency, power level, bandwidth , and modulation technique. An ideal SDR system is shown in Figure 2.1.

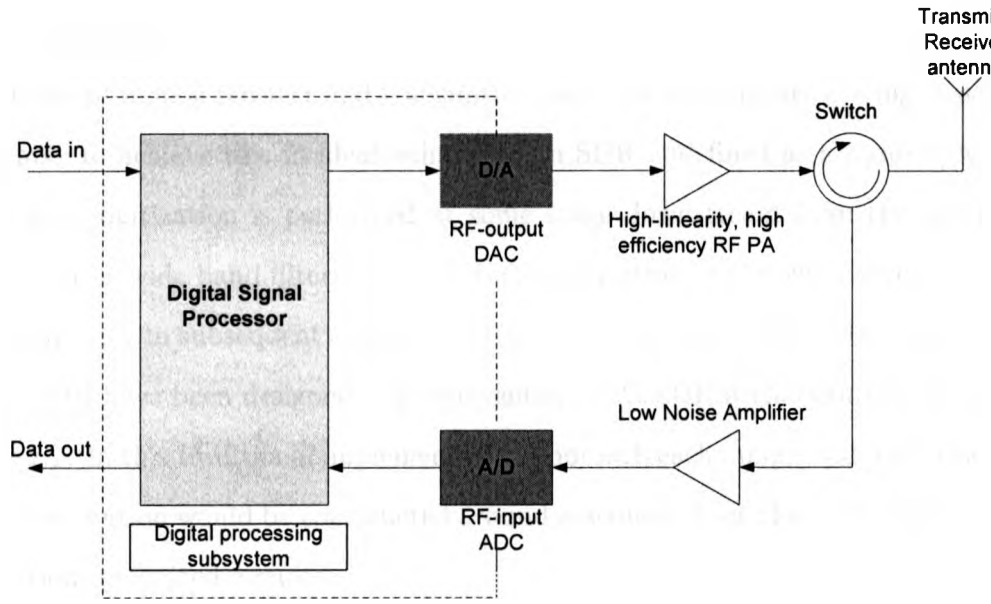


Figure 2.1: Ideal software defined radio architecture

[2]

The figure depicts the radio with minimum of analog components. The modulation schemes, channelization and protocols for transmit and receive are all determined via software within the digital processing subsystem. The switch depicted in the figure is a circulator that separates the transmit and receive path signals without any frequency restrictions. This circuit provides an ideal solution by matching itself with the antenna requirements and the impedances from the power amplifier. Moreover, the circulator has to be highly broadband which most of the current radios are not.

An ideal transfer of the RF modulation from the DAC to a high-power signal suitable for transmission is ensured by the linear power amplifier. The signal from the receiver antenna is passed to the low noise amplifier (LNA) where it is amplified to the desired signal power. The losses in the receiver chain are reduced by the gain in the LNA. Analog-to-digital converter(ADC) of the appropriate resolution samples the data to generate digital signals.

Current analog receiver and transmitter hardware sections are getting closer to being able to achieve this in ideal behavior. An SDR is defined as a radio in which the receive digitization is performed at some stage downstream from the antenna, typically after wide band filtering, low noise amplification, and down conversion to a lower frequency in subsequent stages [3]. Over the years, many different architectures for the SDR have been designed. The conventional 2G SDR architecture is shown in Figure 2.2. In this traditional implementation approach each unique radio interface or band combination would be constructed around a dedicated set of specific application or function integrated circuits.

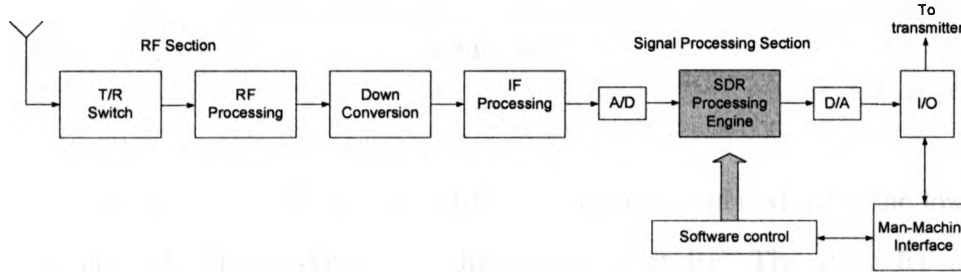


Figure 2.2: Software Defined Radio - 2G model: single band and single mode [4]

It basically consisted of a RF section, where the RF processing took place, a

down conversion circuit that reduces the operating frequency to an intermediate stage, and a signal processing block where the IF to baseband conversion and digital signal processing takes place. The ADC is placed after the intermediate frequency (IF) processing. As shown in the figure,the radio baseband processing is under software control and a man-machine interface permits some manual input from the user. This architecture is considered to be SDR since some, but not all, of the signal processing is accomplished in software. If the ADC is to be moved closer to the antenna, the radio moves closer to the ideal, namely the SR. This is portrayed in Figure 2.3 which illustrates the software radio concept in which digitization is near the antenna.

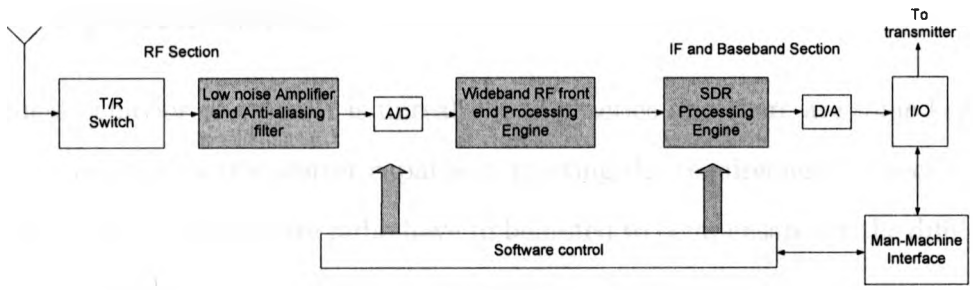


Figure 2.3: SDR evolution - 4G : highly reconfigurable, multi band and multi mode architecture

[4]

As indicated in the figure, the ADC is immediately placed after the low noise amplifier and anti-aliasing filter, i.e., digitization is at RF. The other RF and IF processing is performed by the wideband RF front end processing engine. The intermediate case, which is not shown, would be a direct conversion to baseband from RF performed in the RF processing engine thereby eliminating analog IF processing. It is evident from the inspection of these two figures that there is a key transition from SDR to SR.

Figure 2.3 also illustrates the concept of Adaptive Intelligent SR (AI-SR) in which the radio has the capability of adapting to the operational environment [5]. The RF front processing engine is under the control of a powerful software control processing engine. This engine implements the artificial intelligence and the processing algorithms that give the SR a truly adaptive capability. While this capability is the desired goal, the implementation of this concept is in the embryo stage. The next section talks about the requirements in hardware specifications of an SDR.

2.2 Software Defined Radio Hardware Specifications

The ideal behavior of an SDR is unrealistic in practice and there are some key difficulties in realizing a transceiver capable of meeting the requirements. Specifications for each block in the software radio have to be noted to compensate for the difficulties as much as possible. Some of the specifications are summarized as follows:

- *Antenna*: For ideal situations, a frequency range of 5 octaves is required and the gain/loss figure under such conditions should be around 0 dBi [6]. The challenges are more when the required size of the antenna is small and a near-omnidirectional coverage pattern is required. A more realistic specification for wireless applications will be 1-2 octave with a gain/loss of 2-3 dBi [2].

- *Circulator/duplexer/RX/TX switch:* The main requirement with the circulator is a high degree of isolation to prevent the transmit signals from overloading the receiver front-end. This is usually very challenging to achieve with the frequency range limitations in these components and also with small size

requirements. Typical isolation values range from 10dB to 60dB. Primary application of protecting the transmitter from wide range signals can be achieved with these isolation values. But usually the transmitter is disabled during the receive portions of the communication. Another requirement with these circuits is a broadband coverage range which again is not a trivial requirement.

- *A/D converter:* Radio architectures may incorporate different sampling techniques. Direct sampling techniques are based on Nyquist sampling requiring that the sampling rate is at least two times the highest frequency component of the analog signal. In quadrature sampling the analog input signals are split into in-phase and quadrature components, each occupying only half of the bandwidth of the original signal. Intermediate frequency(IF) sampling(or subsampling) of the bandpass signal requires the sampling frequency to be at least two times the bandwidth of the signals. In all these cases, the ADC is critical for system operation. One of the main factor is the dynamic range of the converter which is affected by statistical properties of the input signal, peak to average ratio, level of the interference etc. [7]. ADCs available for wireless applications are 14 bit resolution devices operating in excess of 100MHz. But recently for better approximation, 20-bit resolution ADCs operating in excess of 250MHz are being used by latest technologies. These ADCs maintain a 120 dB spurious free dynamic range(SFDR) over a typical signal-to-noise ratio of 65 dB [8]. In the case of undersampling, the sampling rate could fall to 20 MSPS. This would make the sampling requirement more stringent. If a synthesizer and downconversion circuits are employed , then low cost, moderate resolution ADCs can be employed.

- *D/A converter:* High performance DACs are used in the transmit path to recon-

struct one or more of the carrier signals. These converters are easily realizable, although they consume a lot of power, with adequate power control in the power amplifier or prior to it. In many cases, it is the performance of the DAC that determines whether a particular modulation scheme or a system architecture can meet the specification. Selecting a DAC for a given wireless system requires an understanding of how to interpret various specifications and an appreciation of their effects on system performance. State-of the-art DAC devices are 14-20 bit devices with SNR higher than 80dB and sampling rate of 400 MSPS and above [8].

- *Anti-alias Filtering:* Based on the Nyquist sampling that was discussed above, an attenuation of atleast 60dB is required around 20 MHz from the channel edge [9]. This would be difficult to realize in a bandpass filter capable of tuning from 100MHz to 2.2GHz. Current technologies employ downconversion circuits that incorporate anti-aliasing along with sample rate conversion.

- *Signal Processors and equivalent technologies:* The primary issue at present is that of power consumption in these processors. Reconfigurable hardware like FPGAs are likely to yield the best performance in terms of power consumption. These technologies are discussed in the next section.

- *RF Power Amplifier:* Many techniques have been employed for the linearization of power amplifiers like RF predistortion, digital predistortion and feedforward techniques achieving upto -70dB intermodulation product levels [9]. At present, the digital predistortion method is more popularly employed.

- *RF Low Noise Amplifier:* The most important drawback here is the gain in the low noise amplifier. The gain in the low noise amplifier should only be

enough to overcome the receiver's input-referred noise. State-of-the-art low noise amplifier devices for wireless applications are able to achieve 120 dB with a corner frequency of 4Hz.

2.3 Digital Aspects of a Software Defined Radio

The digital processing area is as challenging as the analog processing in a software defined radio. The two biggest issues at present are the power consumption and cost of various options. Costs in processing the device, costs involved in interfacing devices, costs related to fabrication and mask-set(associated with Application specific integrated circuits), costs involved with cooling of the devices, additional cost involved in the power supply due to over power consumption, associated costs in tools and training of software, costs in terms of development and time to market are some of the factors that influence the cost of digital elements of an SDR.

2.3.1 Digital Signal Processors (DSPs)

DSPs have the advantage of complete flexibility, wide applicability and a wide availability of skilled practitioners in their software. They are high volume devices and they have the benefits of economies of scale. This makes up for their lack of optimization. It is well suited to extremely complex maths-intensive tasks, with conditional processing rather than very high-speed front-end applications. They are often used for online processing of data that has been acquired from another device after some preprocessing. It is limited in performance by the clock rate, and the number of useful operations it can do per clock. DSPs are optimized for use of external memory and re-use of processing units.

2.3.2 Field Programmable Gate Arrays (FPGAs)

Field Programmable Gate Arrays (FPGAs) have been suggested as an enabling technology for the hardware platform as they offer the potential of hardware-like performance coupled with software-like programmability [10]. In the past, the use of digital signal processors was nearly ubiquitous, but with the needs of many applications outstripping the processing capabilities of digital signal processors (measured in millions of instructions per second (MIPS)), the use of FPGAs is growing rapidly. FPGAs also have the inherent advantage in product reliability and maintainability, fast time-to-market,simple design cycle and field reprogrammability. The high speed functionality of FPGAs makes them ideally suited for the real-time processing of Fast Fourier Transforms (FFTs), Finite Impulse Responses (FIRs),Digital Down Converters (DDCs) and Digital Up Converters (DUCs). It is also possible to add IP processor cores into an FPGA. This makes possible a single-chip solution in some applications and this may be important for size and reliability reasons. FPGAs have undergone many upgrades over the years that they are no longer used as prototypes but can now be used throughout volume productions.

A typical FPGA device consists of an array of configurable logic blocks(CLBs) surrounded by configurable routing. Each logic block consists of resources which can be configured to define discrete logic, registers, mathematical functions, and even random access memory. Configurable pads provide connections to other electronic devices. Figure 2.4 illustrates the classic FPGA architecture.

The function of all of these configurable resources can be defined at any time during the operation of the device to form a large logic circuit. Parallel and pipelined data flows are possible, providing an excellent resource for execution of the signal processing algorithm. Partial reconfiguration is another important enhancement in

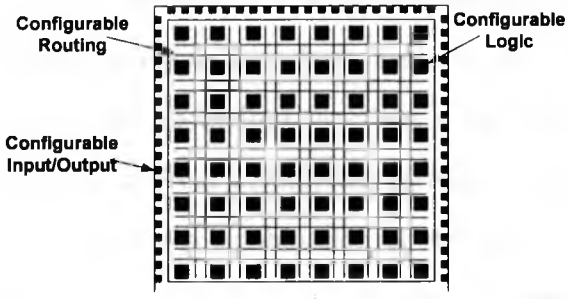


Figure 2.4: Classical FPGA architecture
[10]

FPGAs; sections of the FPGA logic can be reconfigured without interrupting any processing being simultaneously carried out in the other parts of the same device. The important issues while using FPGAs as a processing unit are the CLB architecture, the RAM architecture, the I/O signaling and the clock. There are two different types of CLBs based on their architecture : fine-grained and coarse-gained. Course-grained CLBs are used for particular features like dedicated RAM or arithmetic logic, thereby providing higher processing speeds due to the specific optimized silicon circuitry and minimal routing requirements between them [10]. Fine grained CLBs are relatively fast and perform small simple logic functionality, but pay the performance price arising from the extra routing required to interconnect them [10].

2.3.3 Application-specific integrated circuit (ASIC)

Application-specific integrated circuit are integrated circuits that are programmed for a specific application rather than intended for general-purpose use. While the ASIC is very fast and consumes little power, it generally can be tested as a finished IC. ASICs provide the full custom capability for design since device is manufactured to design specs. The main issue with utilizing ASICs is their lack of flexibility i.e.,implementing designs in an ASIC preludes changes later in the design cycle and any errors in the

finished design cause significant delay for designing, testing and processing of new silicon. But recently many attempts have been made to introduce flexibility within an ASIC like provision of multiple toolbox functions with flexible input parameters, provision of hardware for all current modulation formats with the ability to switch between different paths and also a combination of the previous two methods with some DSP functionality [11, 12, 13, 14, 15]. Another problem with ASICs are as more modes and bands are supported by a single handset, the number of ASICs required increases linearly. This results in large silicon area, associated cost, and power consumption problems.

2.4 Conclusion

In this chapter, the ideal software radio has been described and its architecture has been explained. The required hardware specfications to design an SDR has been discussed to demonstrate why the ideal SDR is not feasible. Some of the key issues and aspects in the digital processing section have been pointed out. The three main digital hardware being used today are DSP, ASIC and FPGA. These devices have been individually explained and their advantages and disadvantages have been discussed. It has been observed that in the current technological era, FPGA's have an edge over ASICs and DSPs due to their flexibility, fast time-to-market capability and the modern techniques used to conserve power.

Chapter 3

RF Front End Technology and Architectures

This chapter is focused on explaining and presenting some design techniques for synthesis of SDR RF translation architectures. The architectural issues behind receiver design are considered and required specifications for radio linearity are defined. Then the requirements for a superheterodyne SDR receiver are covered. The design of an SDR transmitter is considered and the design and linearity issues are highlighted. Transmitter architectures are examined and similar conclusions are drawn as were drawn in the receiver design. The zero IF stage is examined and the specific issues that arise in this architecture are described. The Low IF design is also considered as a possible compromise between the superheterodyne and zero IF architectures.

3.1 SDR Receiver Specifications and Design Considerations

The most important design parameters when dealing with SDR receiver are the input sensitivity, the maximum expected input signal and the blocker specification [4].

3.1.1 Basic Receiver Considerations

The basic receiver function is to take a real,low power, RF signal and down-convert it to a complex (in-phase and quadrature, I/Q) baseband signal. During this process, the signal power level is increased. There are several characteristics of the input signal that define the output signal from the receiver. One of them is the signal type which has to be real. Second is the low power which has to be below -107 dBm [4]. Third is the dynamic range of the signal, which has to be as high as -15 dBm[7]. Finally, the spectrum has to be bandpass, with center frequencies varying from 876 MHz to 5725 MHz[7]. The output signal type will be complex in nature with the phase and quadrature components(I/Q). The spectrum of the output signal will be baseband, with bandwidth up to 20 MHz. The dynamic range of the output signal is reduced in the IF stage to meet the requirements of the ADC. These input and output characteristics of the receiver must keep the signal power sufficiently greater than the noise power. This will ensure that the output signal-to-noise ratio (SNR) is sufficiently high to allow appropriate BER performance. It should be ensured that high power input signals do not overload components of the receiver and do not affect the detection of a wanted signal. These factors can be tackled by selection of an appropriate architecture and application if appropriate technological methods such as image reject mixing, linearization, and variable preselect filters.

3.2 Receiver Architectures

The primary distinction between receivers is the number of stages taken to down-convert a signal to baseband. Direct conversion takes one down-conversion, super-heterodyne receivers employ two or more. Complexity increases with the number of down-conversions. The simplicity of direct conversion brings with it several technical

problems which would appear to make direct conversion architecture inappropriate for an SDR receiver.

3.2.1 Direct Conversion Architecture

A basic direct conversion receiver architecture is shown in Figure 3.1. The receiver consists of a low noise amplifier (LNA) which provides modest RF gain at a low noise figure. The output signal from the LNA is filtered in a preselect filter, and down-converted in a complex (I,Q) mixer. The majority of the gain and automatic gain control(AGC) is provided in a high gain baseband amplifier.

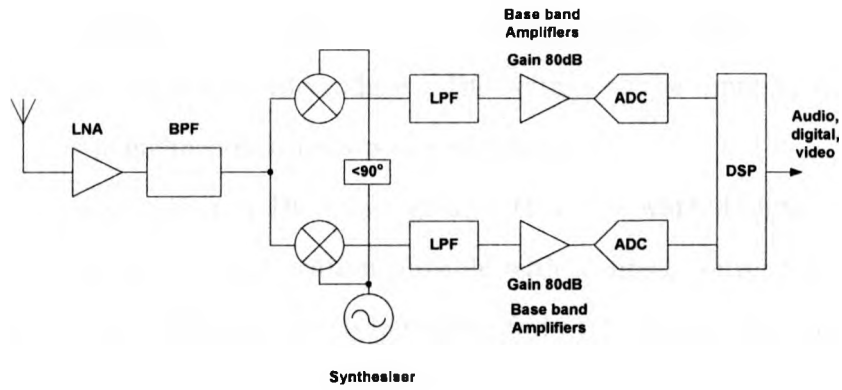


Figure 3.1: Direct conversion Receiver Architecture
[16]

The advantages of having a direct conversion architecture is that it has low complex architecture and does not require an IF stage. Image signals in a direct conversion architecture turns out to be a frequency-inverted image of the wanted signal itself and, although some image rejection advantages can be achieved, the image cannot be ignored. If the I and Q components of the local oscillator are in precise phase quadrature and precise amplitude balance, then the image signal will

be eliminated. But if this is not achieved, then a small sideband will be superimposed on the wanted sideband, resulting in magnitude and phase errors[17]. Since the image signal is derived from the wanted signal itself, the requirements for image rejection are not quite as great as would be the case in superheterodyne architecture. Another advantage is that the filter requirements are simple and straight forward. It is suitable for integrated circuit realization[17]. Inspite of these advantages, the direct conversion architecture has several technical problems. A significant problem with direct conversion architecture is the introduction of a DC offset. This DC offset can arise from a number of sources, one typical source is shown in Figure 3.2. It can be seen that the leakage of the local oscillator signal to the input of the LNA, occurs through an imbalance in the mixer. Once these signals are in the system, they are then mixed with themselves to produce a DC voltage at the input to the baseband amplifier. This phenomenon is known as self-mixing [18].

In such a situation, a DC offset greater than the wanted signal will place a limit on the amount of amplification possible with a direct conversion receiver. It is possible that the reflection of the LO signal from the antenna will also vary with time. This results in time-varying DC offset. This DC offset can be removed by AC coupling the signal or by adaptively canceling the offset (e.g., TDMA systems). The local oscillator signal not only causes problems by being reflected off the antenna and self-mixed down to baseband, but it can also cause problems by being transmitted from the antenna[19]. This may cause interference to adjacent receivers.

Another problem is due to the large amount of gain placed at baseband frequencies. The overall noise figure of a cascade of amplifiers is given by

$$F = F_1 + \frac{F_2 - 1}{G_1} \tag{3.1}$$

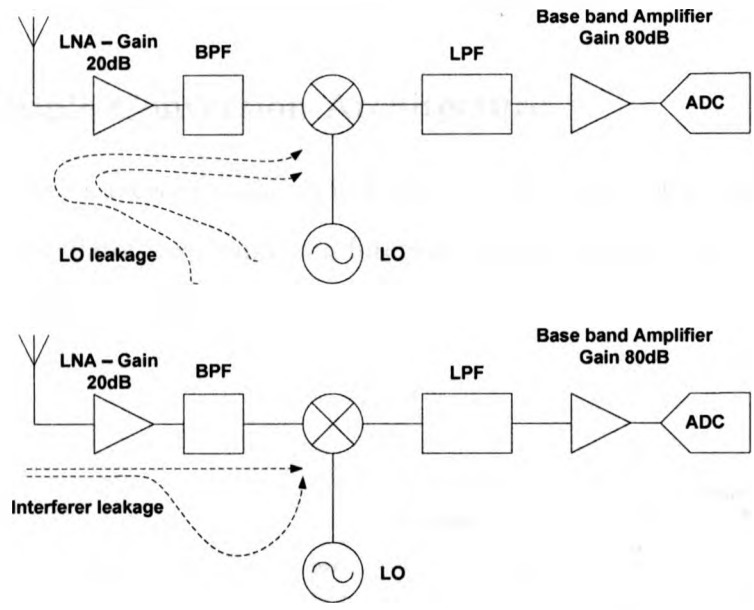


Figure 3.2: Sources of DC offset
[19, 20]

where F is the overall noise figure of the receiver, F_1 is the noise figure of the LNA, F_2 is the noise figure of the baseband amplifier and G_1 is the gain of the LNA. It can be seen that the noise contribution of the second stage is reduced by gain of the first stage. However, the second stage will have an excess noise figure due to its operation at baseband, which incorporates a $1/f$ (flicker noise) component [21]. This makes the noise performance of the second stage just as critical as the first.

Harmonics generated by third-order distortion is an important parameter in the design of a conventional heterodyne receiver. In a direct conversion receiver, the components $(f_1 - f_2)$ and $(f_2 - f_1)$ are still out-of-band in the RF part of the receiver but fall within the passband of the baseband amplifier [21]. These components may reach baseband through direct leakage through the mixer or second harmonics of the RF signal mixing with the second harmonic of the local oscillator signal. These may

result in a spurious term being present in the receiver output.

3.2.2 Multiple Conversion Architecture

A multiple conversion receiver is shown in Figure 3.3. The receiver consists of a LNA, a variable local oscillator, two band pass filters and an IF amplifier that amplifies the intermediate frequency signal.

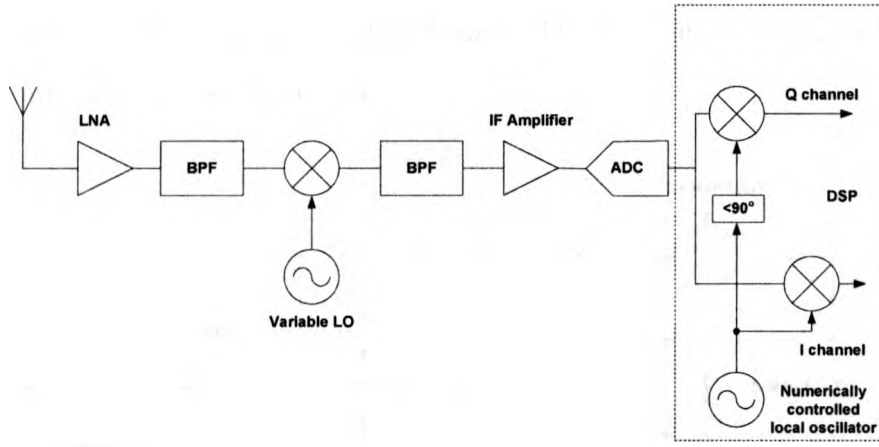


Figure 3.3: Multiple conversion Receiver Architecture
[16]

Although the multiple conversion stage only shows two explicit down-conversions (one in the RF hardware and one in digital signal processing(DSP), further conversions can be done via the processes of decimation and/or subsampling. In this architecture, the first conversion may be done in RF hardware, and all of the others are done in FPGA/DSP.

3.2.3 Low IF receivers

Low IF architectures represents a pragmatic attempt to combine the advantages of a superheterodyne structure with the advantages of a direct conversion architecture. Having a low IF means that the image rejection requirements are not as onerous as with the superheterodyne structure, and the fact that the LO signal is not the same frequency as the wanted signal minimizes the DC offset problems. The architecture basically translates all the RF signals to low IF frequency signals which is then down-converted to Baseband signal in digital domain. The block diagram of Low-IF receiver architecture is shown in Figure 3.4

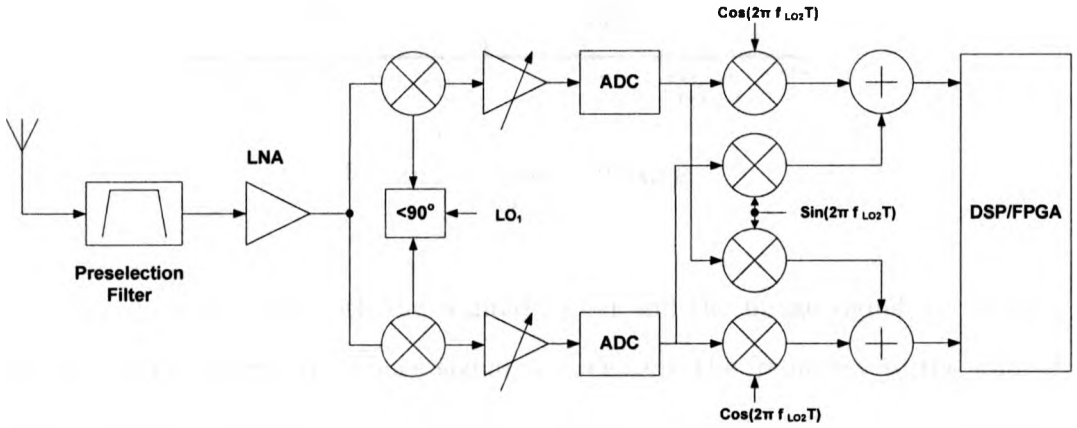


Figure 3.4: Low-IF Receiver Architecture
[16, 22]

After preselection filtering and amplification, all the RF channels are quadrature mixed and downconverted to low IF containing both wanted and unwanted signals. The IF frequency is just one or two channels bandwidth away from DC, which is just enough to overcome DC offset problems. It is then amplified and filtered before sampled by ADC. Since the ADC samples both wanted and unwanted signals, there will be higher demand on ADC dynamic range requirements. The ac-coupled signal

path to ADC eliminates the need of DC offset compensation circuitry. The sampled digital data is fed to image reject mixer which is implemented in digital domain. The signal spectra with a typical low IF down-conversion is shown in Figure 3.5.

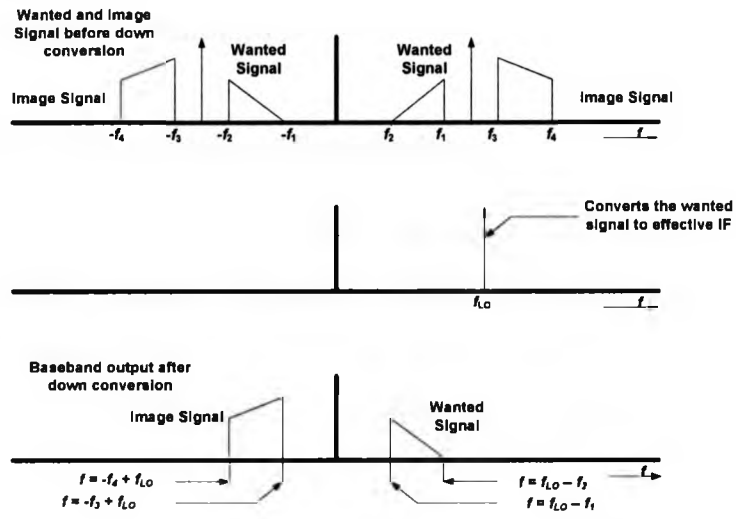


Figure 3.5: Low-IF Mixing
[20]

It can be seen that both the wanted signal and the image signal are present in the baseband, placing the image signal very close in the frequency to the wanted signal, making the image difficult to remove using conventional filtering. To remove the image requires image reject mixing to be performed. Image reject requirements for a low IF stage are not as stringent as those for a conventional superheterodyne stage, nor are they as low as the zero IF stage. An alternative to image reject mixing is use of a complex filter. A conventional filter, operating on real signals, is only capable of realizing complex poles in pairs. If the signal is complex, however, then a filter that contains a single complex pole may be realized [23].

3.3 Transmitter Architectures

Basically the same choice applies to transmitter architectures as applies to receiver architectures. The advantages and disadvantages associated with receiver architectures more or less translate to transmitters.

3.3.1 Direct Conversion Transmitter

The block diagram of a direct conversion transceiver is shown in Figure 3.6. The transmitter consists of Digital-to-Analog converters and a IQ modulator that converts the baseband frequency directly to RF. The output signal is then filtered in a band pass filter and then amplified using a High Power Amplifier(HPA).

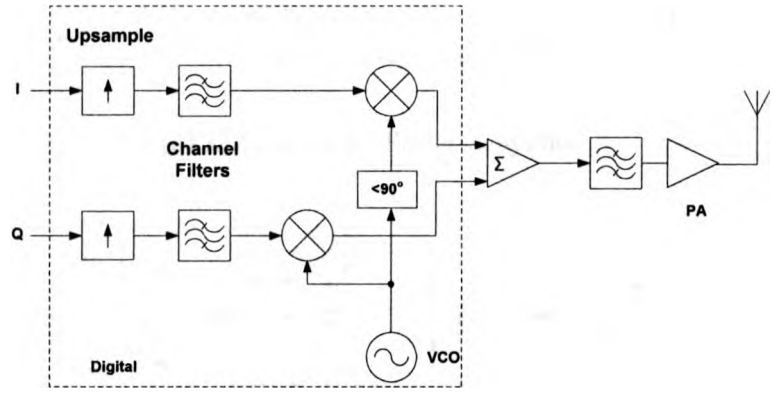
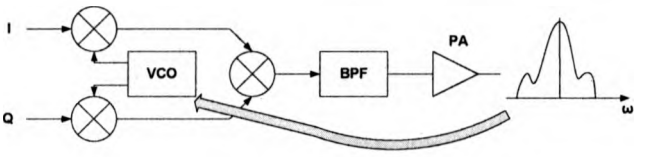


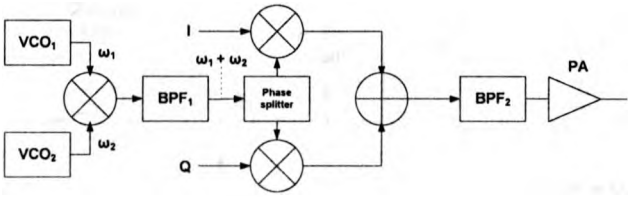
Figure 3.6: Direct up-conversion transmitter

Low complexity is one of the main advantage of a direct conversion transmitter. It also has simple filter requirements and it's suitable for integrated circuit realization. Due to lower number of parts used in this architecture, it helps to reduce the current consumption. Moreover, unwanted signals or image signals and sideband problems

are dealt with easily. The direct-conversion architecture nonetheless suffers from an important drawback: disturbance of the local oscillator by the power amplifier output [20]. This issue arises because the PA output is a modulated waveform having a high power and a spectrum centered around the LO frequency (Figure 3.7(a)). Despite various shielding techniques employed to isolate the VCO, the noisy output of the PA still corrupts the oscillator spectrum. This corruption occurs through injection pulling or injection locking [24, 17], whereby the frequency of an oscillator tends to shift towards the frequency of an external stimulus. The local oscillator leakage through the mixer will be radiated through the antenna. The power amplifier linearization and the final mixers have to operate over a wide frequency band.



(a) LO pulling by Power Amplifier



(b) Direct conversion transmitter with offset LO

Figure 3.7: (a) LO pulling by Power Amplifier and (b) Direct conversion transmitter with offset LO
[25]

The phenomenon of LO pulling is alleviated if the PA output spectrum is sufficiently far from the oscillator frequency. For quadrature upconversion, this can

be accomplished by offsetting the LO frequency, that is, by adding or subtracting the output frequency of another oscillator. The selectivity of the first bandpass filter, BPF_1, in Figure 3.7(b) impacts the quality of the transmitted signal. Owing to nonlinearities in the offset mixer, many spurs appear at the input of BPF_1. If not adequately suppressed by the filter, such components degrade the quadrature generation of the carrier phases as well as create spurs in the upconverted signal.

3.3.2 Multiple Conversion Transmitter

Another approach to circumventing the problem of LO pulling in transmitters is to upconvert the baseband signal in multiple steps so that the PA output spectrum is far from the frequency of the VCOs. A multiple conversion architecture is shown in Figure 3.8.

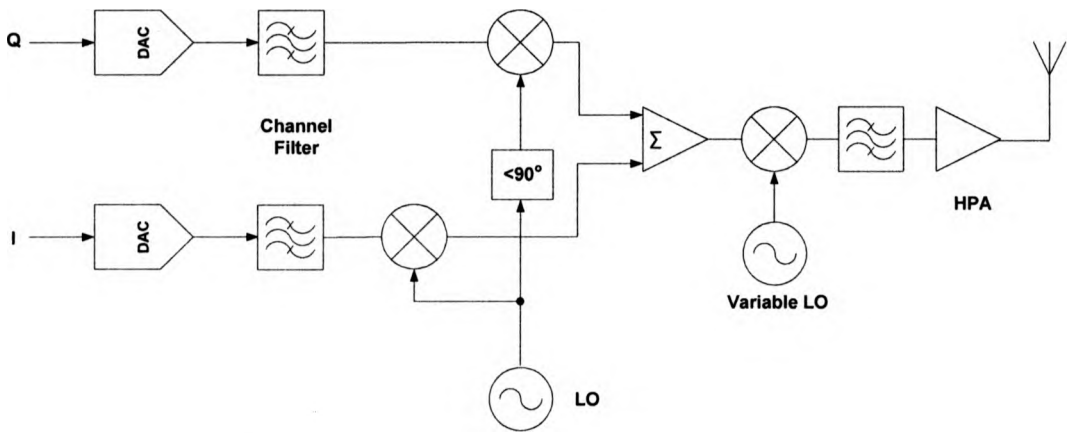


Figure 3.8: Multiple conversion transmitter
[22]

Here the baseband I and Q channels undergo quadrature modulation at a lower

intermediate frequency, and the resulting output is upconverted by mixing and band-pass filtering.

3.4 Conclusion

In this chapter, an attempt is made to review the issues associated with the design of radio frequency hardware. Different receiver and transmitter architectures are studied and issues in some of the candidate architectures are illustrated. The two significant differences between SDR RF hardware and traditional requirements are that its operating frequency and channel bandwidth are not predetermined. The standard zero IF architecture has advantages of modest image rejection requirements and circuit simplicity. This has led to immense development in these transceivers. A low IF architecture provides a good compromise between the conventional model and the zero IF approaches.

Chapter 4

Digital Front-End Design

4.1 Introduction

In many digital communication applications, the input signal may not be in the correct frequency spectrum for processing. The signal band may be in kHz width and the incoming signal band may be in MHz range. For example in a GSM module, the input rate is around 80 MHz and the required out rate for digital processing is 270 kHz [26]. If the signal is sampled, according to Nyquist criteria, the data rate of the signal will be very high [27]. The amount of hardware required for processing the data at this high data rate is very high and the implementation is complex. In a software radio, Digital intermediate frequency (IF) section extends the scope of digital signal processing beyond the baseband domain out to the antenna – to the RF domain. Digital IF modulation performs the critical tasks of required frequency translation, channelization and data rate adaptation. This increases the flexibility of the system while reducing manufacturing costs. Moreover, digital frequency conversion provides greater flexibility and higher performance (in terms of attenuation and selectivity) than traditional analog techniques [28]. Thus, digital IF design plays an important role in digital communications.

The basic functionalities of a digital front-end are channelization (down/up conversion and filtering) and sample-rate conversion. These are pictorially represented in Figure 4.1.

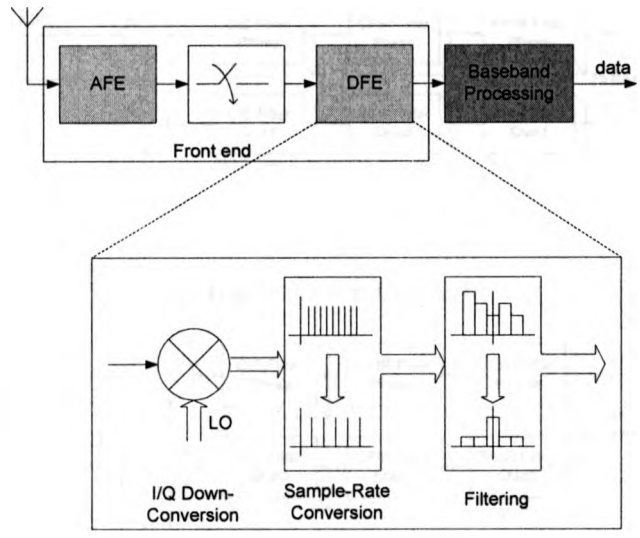
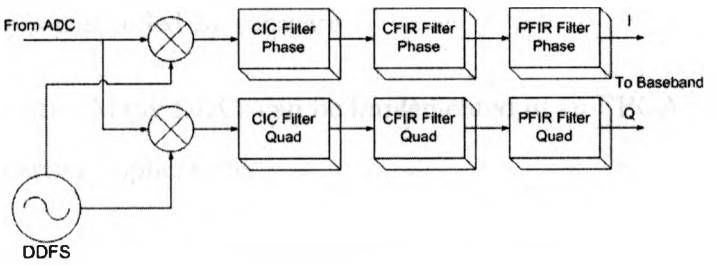


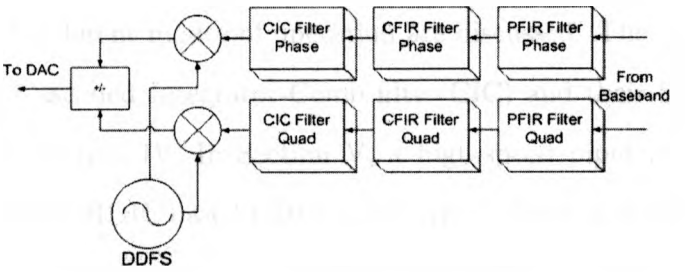
Figure 4.1: Digital front-end of a digital receiver

The digital IF front-end mainly consists of a DDC in the receiver and a DUC in the transmitter. The block diagrams of a conventional DDC and a DUC are shown in Figure 4.2.

The digital signal from the analog-to-digital converter (ADC) is multiplied with the cosine and sine components, generated by the Direct Digital Frequency Synthesizer, for the phase and quadrature channels respectively. This operation guarantees a downward shift in the frequency spectrum of the signal. As a result of this quadrature sampling, the hardware required is doubled, but the data rate of the signal is halved. Identical components must be used for the phase and quadrature channels. The advantage with quadrature sampling is that the negative frequencies are canceled [28]. The I and Q signals are then processed by a concatenation of filters. By attenuating the unwanted frequencies the signal can be resampled at a lower rate. The reduced sample rate relaxes the processing after the DDC. Depending on the selected frequency range, sample rate, and desired quality, different filter combinations can be



(a) Digital Down Converter



(b) Digital Up Converter

Figure 4.2: Conventional Models

used to perform the DDC. The phase and quadrature outputs of the DDC are connected to the baseband section where further processing like Fast Fourier Transform, Orthogonal Frequency Division Multiplexing etc. are carried out to analyze the signal spectrum [29, 28]. The reverse operation of the DDC is the DUC in the transceiver, where the input signal is up sampled by a cascade of filters and modulated to a higher frequency by a direct digital frequency synthesizer(DDFS).

One may ask , why use digital conversion over analog techniques. DDC offers the following advantages over analog methods:

- Stability : Problems regarding temperature variations, component tolerance or manufacturing processes are eliminated [28]

- Flexibility : Frequency hops in the incoming signal can be controlled easily since

everything is controlled by software

- Multiple units : Many DDCs can be implemented in an FPGA which is suitable for multi-carrier applications.

The rest of this chapter is organized as follows: In Section II, the conventional DDFS model is discussed. In Section III, the basics of the CORDIC algorithm, it's general architecture and different modes of operation are discussed. The theory behind low-pass filters like Cascaded Integrator-Comb filter(CIC) and their frequency response are discussed in Section IV. In Section V, a high-speed pipelined architecture for implementing the CORDIC based DDFS is presented. Error analysis on the CORDIC algorithm is also carried out here. A modified CIC filter chain design for hardware efficient sample rate conversion is presented in Section VI. The complete DDC and DUC design and architecture along with simulation results are depicted in Section VII. Section VIII summarizes the chapter.

4.2 Direct Digital Frequency Synthesizer

With regards to system performance, the critical component in digital down conversion is the Direct Digital Frequency Synthesizer. This component is a digital signal generator which generates a sampled digital sinusoid, which when mixed with the incoming signal, a frequency translation or shift of the signals spectrum occurs [?]. In the conventional model (Figure 4.3), the DDFS consists of a phase accumulator and a phase angle-to-wave shape converter (conventionally a sine/cosine ROM)[30].

It is implemented by multiplying the digitized input signal with amplitude values of sine and cosine functions. The sine and cosine functions, stored in the ROM

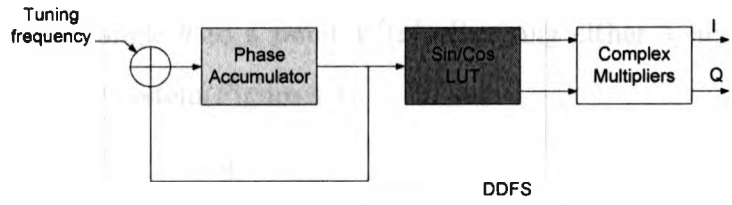


Figure 4.3: Conventional ROM table based DDFS

table, are addressed directly by the phase accumulator. The phase accumulator consists of a j-bit frequency register which stores a digital phase increment word followed by a j-bit full adder and a phase register [31]. The digital input phase increment word is entered in the frequency register. At each clock pulse this data is added to the data previously held in the phase register. The phase increment word represents a phase angle step that is added to the previous value at each $1/F_{clk}$ seconds to produce a linearly increasing digital value [31]. The tuning frequency for the phase accumulator is given by

$$F_t = \frac{F_{clk}P}{2^j} \tag{4.1}$$

where, F_clk is the accumulator clock, P is the tune word, j is the number of bits in the accumulator and F_t is the tuned frequency. For applications demanding high resolution, this technique requires a large look-up table, resulting in large chip area, high power consumption, lower speed and increased costs. A more hardware-efficient solution for implementing the DDFS is Volder's CORDIC algorithm [32].

4.3 Basic CORDIC equation

The Coordinate Rotation Digital Computer is an iterative arithmetic algorithm that uses only shifts and adds to convert between polar and rectangular coordinates. The

basic task performed in the CORDIC algorithm is to rotate a two-dimensional vector $V(x, y)$, through an angle θ to a point $V'(x', y')$, using either a linear, circular or hyperbolic coordinate system(Figure 4.4).

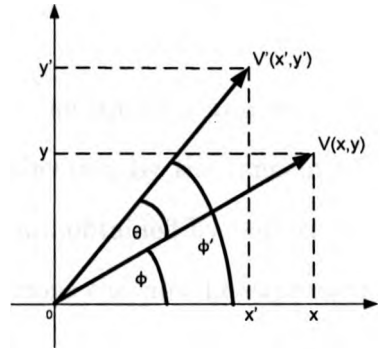


Figure 4.4: Rotation of a vector V by the angle θ

This is accomplished in the algorithm through a series of incremental angles whose algebraic sum approximates the desired rotation angle. These elementary angles are stored in a small look-up table. This removes the need for the large Rom table used in the earlier model. Since the iterations are performed using shifts and adds, the CORDIC structure can be easily realizable in hardware using FPGAs. Another advantage of the CORDIC algorithm is that it uses discrete logic rather than multipliers as required for operation of the conventional DDFS.

The algorithm is derived from the basic rotation transform:

$$x' = xcos\theta - ysin\theta \tag{4.2}$$

$$y' = ysin\theta + xcos\theta \tag{4.3}$$

These equations rotate the Cartesian plane by the angle θ. Simplifying these equa-

tions, we get

$$\begin{aligned} x' &= cos\theta(x - ytan\theta) \\ y' &= cos\theta(y + xtan\theta) \end{aligned}$$

The assumption here is that the rotation angles are constrained such that $tan\theta = 2^{-i}$. In this way, the multiplication by the tangent term is reduced to simple shift operation. Arbitrary angles are obtained by performing a series of small elementary rotations. The iterative rotations can now be expressed as

$$\begin{aligned} x^{i+1} &= K_i[x_i - y_i.d_i.2^{-i}] \\ y^{i+1} &= K_i[y_i + x_i.d_i.2^{-i}] \end{aligned}$$

where K_i is the scale factor and is given as

$$\begin{aligned} K_i &= cos(atan2^{-i}) = \frac{1}{\sqrt{1 + 2^{-2i}}} \\ d_i &= \pm 1 \end{aligned} \tag{4.4}$$

Removing the scale coefficient from the formula it turns out that the algorithm now contains only addition and shift operations. K_i coefficient may be taken in account on any step of the operations performance, including either before or after carrying out all iterations. Then it is marked as A_n

$$A_n = \prod_n \sqrt{1 + 2^{-2i}} \tag{4.5}$$

This factor approaches 1.647 as the number of iterations goes to infinity. Compensation for the scale factor can be done elsewhere or treated as a part of the processing

gain. The sequence of directions of the elementary rotations is determined by an additional adder-subtracter that accumulates the rotation angles at each iteration. Those angular values are supplied by a small lookup table or are hardwired. Therefore, the third CORDIC equation is

$$z_{i+1} = z_i - d_i.tan^{-1}(2^{-i})$$

The CORDIC algorithm operates in two modes: rotation mode and vectoring mode. In the rotation mode, the input vector is rotated by a specified angle. The decision at each iteration is based on the sign of the residual angle after each step. The CORDIC equations for rotation mode are

$$x_{i+1} = x_i - y_i.d_i.2^{-i} \tag{4.6}$$

$$y_{i+1} = y_i + x_i.d_i.2^{-i} \tag{4.7}$$

$$z_{i+1} = z_i - d_i.tan^{-1}(2^{-i}) \tag{4.8}$$

where

$$d_i = -1\ if\ z_i < 0, +1\ otherwise$$

The final CORDIC equations in rotation mode are:

$$\begin{aligned} x_n &= A_n[x_0 cosz_0 - y_0 sinz_0] \\ y_n &= A_n[y_0 cosz_0 + x_0 sinz_0] \\ z_n &= 0 \end{aligned} \tag{4.9}$$

where A_n is the cumulative gain obtained after all the iterations. In vectoring mode, the y component of the input vector is minimized at each rotation. The result is

a rotation angle and a scaled magnitude of the original vector. The sign of the y component is used to determine the direction of rotation at each step.

$$d_i = \begin{cases} 1 & \text{if } y_i < 0 \\ -1 & \text{otherwise} \end{cases}$$

If the angle value accumulator is initialized with zero, it will contain the resultant angle at the end of all iterations. For this mode the CORDIC equations are

$$\begin{aligned} x_n &= A_n\sqrt{x_0^2 + y_0^2} \\ y_n &= 0 \\ z_n &= z_0 + tan^{-1}\left(\frac{y_0}{x_0}\right) \end{aligned} \tag{4.10}$$

In order to convert from polar to rectangular coordinates rotation mode is used and if the conversion is from rectangular to polar, vectoring mode is employed. The CORDIC rotator can be used to compute several trigonometric functions like sine, cosine, magnitude and phase (arctangent) to any desired precision [33]. There are a number of ways to implement a CORDIC processor. The ideal architecture depends on the speed versus area trade offs in the intended application. The fundamental architecture is the direct translation of the basic CORDIC equations discussed above. This is shown in Figure 4.5.

The structure consists of three adder-sub tractor units, two shifter circuits to supply the terms $2^{-1}x$ and $2^{-1}y$ to the adder- subs tractor units, and three registers for buffering the output. The function $tan^{-1}2^{-i}$ is precomputed and stored in a look-up table for different values of i. The number of iterations should not exceed the bit width (n). First, the initial values are fed into the registers or hardwired. The

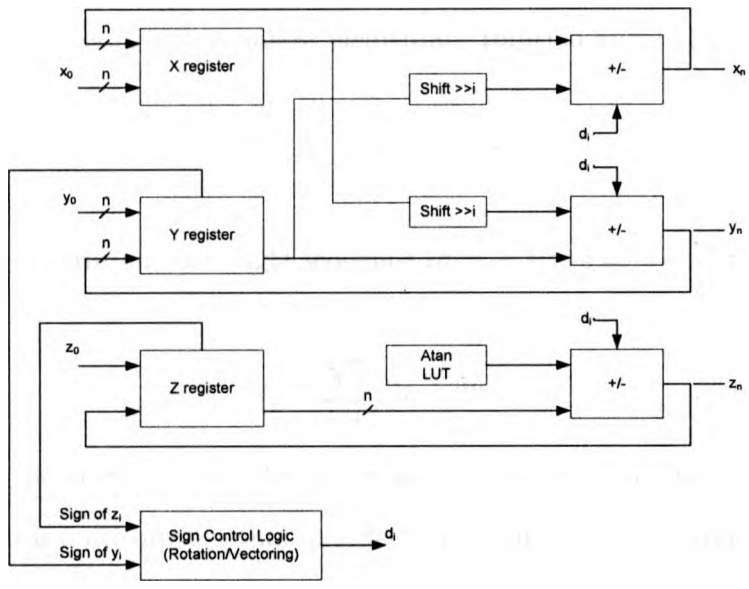


Figure 4.5: CORDIC Rotator architecture(Bit Iterative Parallel)

sign of the angle accumulator, in the case of rotation mode, and the sign of the y component, in the case of vectoring mode, determines the mode of the adder - sub tractor. Signals in the x and y branch pass the shift units and are then added to or subtracted from the non- shifted signal in the opposite path. The z register's value is combined with the values taken from the look up table whose address is changed accordingly to the number of iteration. The inputs to the structure and the outputs are represented in 2's compliment format. A simple sign control logic performs the tasks of generating control signals that organize the shift distance and the addressing of the lookup table.

The CORDIC algorithm introduces errors due to quantization and rounding of the final result. The quantization error is created due to approximation of the rotation angles and rounding of the oscillator's output[34]. For the circular mode of operation,

the approximation error is the component z_n after n iterations in the forward rotation mode and is bounded by the smallest elementary rotation angle $a(n-1)$ [34].

$$|z_n| \leq a(n-1) \tag{4.11}$$

To satisfy this condition, the angle sequence $(a_i; i = 0\ to\ i = n-1)$ must be chosen so that

$$a_i - \sum_{j=i+1}^{n-1} a_j \leq a_{n-1} \tag{4.12}$$

where, a_i are the elementary angles and n is the number of iterations. Based on the above result, it is quite obvious that in order to minimize the approximation error, the smallest elementary rotation angle a_{n-1} must be made small. This can be achieved by increasing the number of the CORDIC iterations. In the reverse rotation mode, the error is derived as

$$|z_n - z_0| = tan^{-1}\left|\frac{y_n}{x_n}\right| \tag{4.13}$$

The rounding error occurs due to the finite word length used to represent the data signals. The rounding error is derived using a error propagation formula [34] which is given by

$$\begin{aligned} f(n) &= Q\left[\hat{v}(i)\right] - v(n) \\ &= e(n) + \sum_{n-1}^{j=0} \{B(j)e(j)\} \\ f(n) &= \sqrt{2}\epsilon\left(1 + \sum_{n-1}^{j=0} \|B(j)\|\right) \end{aligned} \tag{4.14}$$

where $B(j) \equiv \prod_{i=j}^{n-1} p_i$, $\epsilon = 2^{-bb-1}$ and

$$p_i = \begin{bmatrix} 1 & d_i 2^{-i} \\ -d_i 2^{-i} & 1 \end{bmatrix}$$
$$= \sqrt{1+2^{-2i}} \begin{bmatrix} cosa_i & d_i sina_i \\ -d_i sina_i & cosa_i \end{bmatrix} \tag{4.15}$$

. The variance of the rounding error is given as

$$\sigma_I^2 = \sigma_Q^2 = \frac{2^{-2bb}}{12} \tag{4.16}$$

The variance of the rounding error of I_i and Q_i is

$$\sigma_{IQ}^2 = \sigma_I^2 + \sigma_Q^2 = \frac{2^{-2bb}}{6} \tag{4.17}$$

In each CORDIC iteration, the rounding error consists of two components: the rounding error propagated from the previous iterations and the rounding error introduced in the present iteration. Therefore the variance due to the rounding error of I_n and Q_n at the CORDIC rotator output is

$$\sigma_{tot2}^2 = \sigma_{IQ}^2 \left\{ 1 + \sum_{j=0}^{n-1} \left\| \prod_{i=j}^{n-1} K_i^2 \right\| \right\} \tag{4.18}$$

where K_i^2 is obtained from equation(4).

4.4 Cascaded Integrator-Comb Filter

Filter design predominantly depends on the application requirements in hand. For high speed signal processing, a series of low pass digital filters are employed out of which CIC and FIR filters are typical [35]. The CIC filter is basically used for decimation and to remove anti-aliasing components in the signal spectrum. The CIC filters, introduced by Hogenauer, are a group of linear phase FIR filters, which are primarily used in high decimation or interpolation systems [36]. These filters are very hardware efficient because they require no multipliers and they use limited storage. The filter employs adders, subtracters and registers to perform the operation. The CIC filter structure is divided into two sections; an integrator section functioning at high sampling rate and a comb section (differentiator) operating at low sampling rate [36]. The structure of a N stage CIC decimation filter is shown in Figure 4.6.

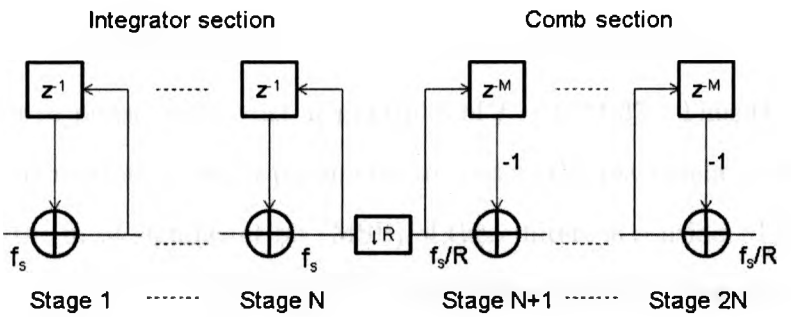


Figure 4.6: CIC Decimation Filter

The decimation process is carried out by cascading N Integrators (each with unity feedback coefficient), operating at high sampling rate, with N differentiators or combs operating at a low sampling rate. A rate changing switch bridges the integrator and comb sections. The switch subsamples the output from the last integrator, reducing the sampling rate from fs to $\frac{fs}{R}$ [36]. The transfer function of the CIC filter

on the z plane is given as

$$H(z) = H_I N(z) H_C N(z) = \left(\frac{1 - z^{-RM}}{1 - z^{-1}} \right)^N \tag{4.19}$$

where, M is the differential delay that controls the filter's frequency response and R is the rate change factor that determines the order of the CIC filter. The frequency response of a CIC filter is given by equation 1.22 evaluated at

$$z = e^{j(2\pi f/R)} \tag{4.20}$$

where f is the frequency relative to the low sampling rate f_s/R. The Power response of the CIC filter is a sinc function given by

$$P(f) = \left[\frac{sin\pi Mf}{sin\frac{\pi f}{R}} \right]^{2N} \tag{4.21}$$

For this power response, nulls exist at multiples of $f = 1/M$[37]. Thus the differential delay M can be used as a design parameter to control the placement of nulls. In any filter design, the most significant bit (MSB) of these filters is considered a function of the overall register growth [38]. For CIC decimations, the CIC datapath undergoes internal register growth that is a function of all the design parameters: R,M,N in addition to the input precision B_{in} [38]. So if B_{in} is the number of input bits, the most significant bit B_{max} at the filter output is defined by

$$B_{max} = [Nlog_2(RM) + B_{in} - 1] \tag{4.22}$$

In this design, B_{max} bits are internally used for each of the integrator and comb stages, to produce full-precision result at the filter output port.

4.5 Implementation of CORDIC DDFS

Using the above iteration structure, the computation time is determined by the cycle time and the number of iteration steps. In order to accelerate the data rate and improve the accuracy of the CORDIC system, a more efficient architecture is adopted here. The architecture discussed in this section presents an unrolled parallel pipelined version of the CORDIC algorithm. Instead of reusing the same hardware for all iteration stages, the parallel architecture has a separate hardware processor for every CORDIC iteration. Each of the n processors performs a specific iteration, and a particular processor always performs the same iteration. This leads to a simplification of the hardware. All the shifters perform the fixed shift, which means these can be implemented in the FPGA wiring. Every processor utilizes a particular arctan value that can also be hardwired to the input of every angle accumulator. Yet another simplification is an absence of a state machine. The pipelined structure accepts new input data and puts out the results at every clock cycle. The only concern with this structure is that it introduces a latency of n clock cycles.

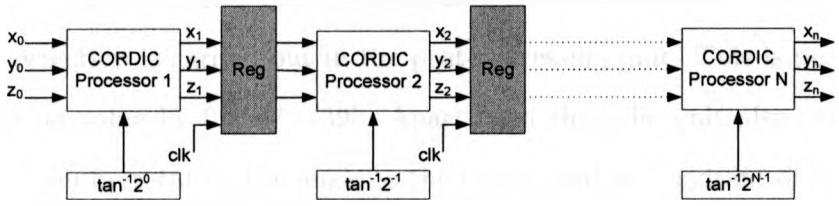


Figure 4.7: Pipelined Architecture of CORDIC

The CORDIC processor is designed with three fundamental blocks: the preprocessing block, CORDIC main core block and a post processing block(Figure 4.8).

The arctan table used in the CORDIC algorithm only converges in the range of -1(radians) to +1(radians). To use the CORDIC algorithm over the entire 2π range

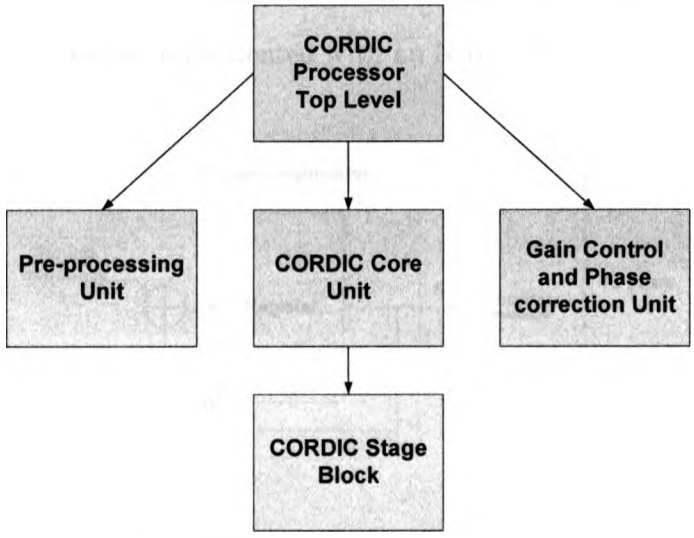


Figure 4.8: CORDIC Processor

the inputs need to be manipulated to fit in the -1 to +1 rad. range. This is handled by the pre-processor. It detects the right quadrant where the given vector is located, and fits it to the range from $0 to 45^o$. All iterations are carried out in parallel, using a pipelined structure. Due to this structure the highest possible throughput is achieved. As mentioned earlier, a scale gain is accumulated after completion of the algorithm. The gain correction is carried out in the post-processing unit. This is performed by multiplying its value by 0.85879 [39]. Apart from this, the unit also performs the angle correction by rotating the angle to the corresponding quadrant of the plane.

The tuning frequency for the CORDIC based DDFS is determined by the phase angle which is generated from the phase accumulator. The CORDIC z_0 input needs a phase input that takes values in the interval $[-\pi, \pi]$, so a multiplication by π is required to extend the interval of the normalized term θ to the interval required by CORDIC [40]. A difference between the conventional method and the CORDIC method is that the phase accumulator generates an integer value that addresses an

LUT in the LUT-based method, while it generates an angle in CORDIC-based DDFS. This accumulator is easily implemented with an N-bit adder.

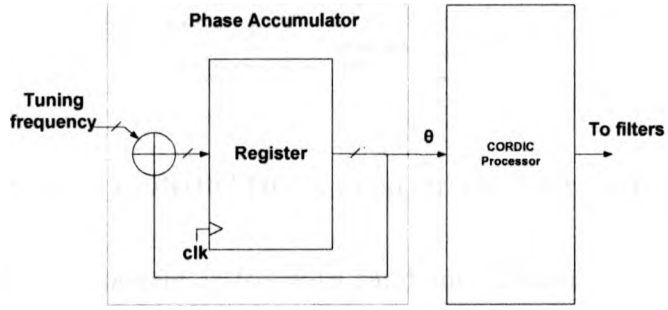


Figure 4.9: CORDIC Direct Frequency Synthesizer

A two's complement fractional numeric format (only one integer bit) is considered; hence a signal in the interval [-1,1] is generated, and a multiplier by π is introduced to achieve the desired range.

The pipelined CORDIC processor is implemented in Altera Stratix II device using Quartus II software. In real software radio receiver applications, the phase and quadrature components can be obtained by loading the x input with the current sample of an IF-input signal, the y input with zero and the z input with the current sample of phase angle at each clock interval. In the transmitter side, the z component is set to zero and the x and y components are given with the phase and quadrature values of the signal respectively. The x, y and z inputs are represented in unsigned 14 bit 2's compliment format. In order to verify the validity of the VHDL design , the pipelined nature of the structure and the throughput, a loop is created between the CORDIC DDFS of the transmitter and the CORDIC DDFS of the receiver. This means that values are fed to the transmitter and it's outputs are given as inputs to the receiver to obtain the initial values.

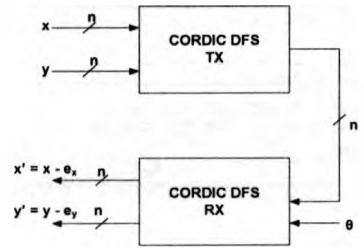


Figure 4.10: CORDIC DDFS Loop structure for verification

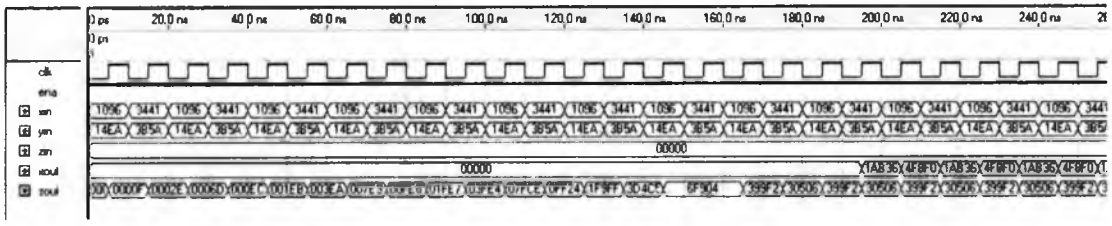
The x and y component is fed with randomly chosen fractional values. The z component is set to a zero for verification purpose. As mentioned earlier a small look-up table is used to store the atan values. The calculation for the arc table upto 7 iterations is shown in Table 4.1.

Iteration	$k = 2^i$	**1+kj**	**Phase)** $tan^{-1}(k)$	**Magnitude**	**CORDIC gain**
0	1.0	1+1.0j	45.00000	1.41421356	1.41421356
1	0.5	1+0.5j	26.56505	1.11803399	1.58113883
2	0.25	1+0.25j	14.03624	1.03077641	1.62980060
3	0.125	1+0.125j	7.12502	1.00778222	1.64244841
4	0.0625	1+0.0625j	3.57633	1.00195122	1.64568892
5	0.03125	1+0.03125j	1.78991	1.00048816	1.64649228
6	0.015625	1+0.015625j	0.89517	1.00012206	1.64669325
7	0.007813	1+0.007813j	0.44761	1.00003052	1.64674351

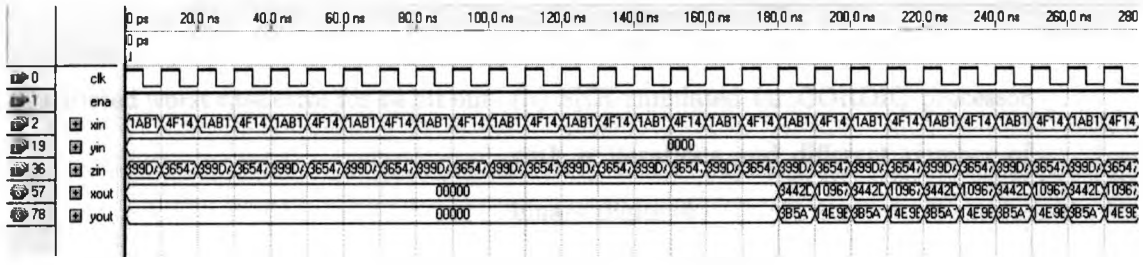
Table 4.1: Look-up table calculation

As it is shown, the CORDIC gain approaches to a value of 1.647 as the number of iterations increase. The phase angle input z, as mentioned earlier, is determined by the phase accumulator. Simulations for a CORDIC based DDFS in the receiver side and transmitter side are shown in Figure 4.11

The simulations depict that the signal given to the DFS of the transmitter is attained back through the DFS of the receiver. It can also be seen that the output



(a) CORDIC DDFS Transmitter



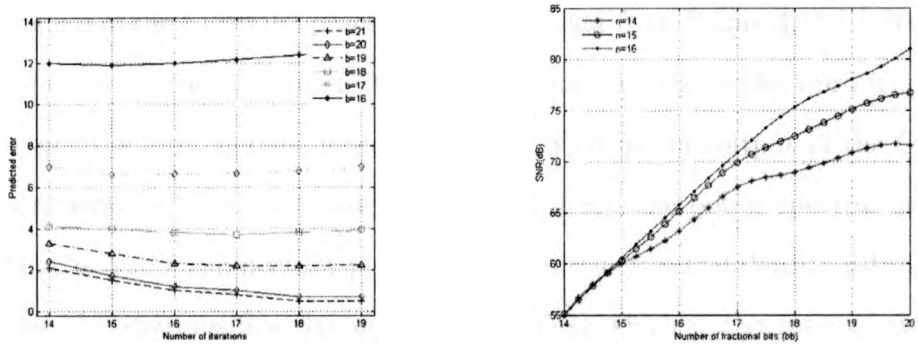
(b) CORDIC DDFS Receiver

Figure 4.11: Simulation Results for CORDIC Processor

is generated at every clock cycle, hence confirming a higher throughput than the conventional model. The generated output has a precision upto the first 14 bits.

4.5.1 CORDIC Error Analysis

One of the prospective applications with the error analysis of the CORDIC algorithm is to facilitate the choice of the length of the shift sequence n and the internal word length b so that the overall quantization error is minimized. The combined overall error is computed for a 14 bit output for different number of iterations. The input signal is taken as a constant one with no input quantization error. Quantization error is modeled as a additive white noise source from which the SNR is estimated. The predicted approximation error bounds are plotted in Figure 4.12(a).



(a) Predicted worst case error for 14 bit output

(b) SNR simulated for CORDIC processor with n iterations and different number of binary digits bb

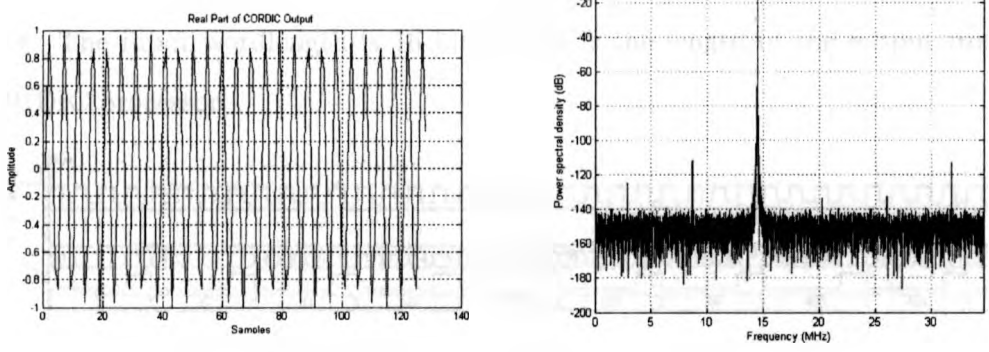
Figure 4.12: Error Analysis of CORDIC

For different values of the word length b the approximation error is plotted by the increasing the number of iterations. From the graph it can be said that as the number of iterations are increased the predicted error is reduced with the increase in the value of b. In the next section, multi-rate filters for sample rate conversion are discussed. The overall quantization noise can be modeled as a white noise source from which the SNR can be calculated [41, 42]. The SNR at the CORDIC output is measured as

$$\frac{S}{N} = \frac{2\sigma^2}{\sigma_{tot}^2} \tag{4.23}$$

where σ^2 is the variance of I_0 and Q_0 data(the mean of I_0 and Q_0 is assumed to be zero) and σ_{tot}^2 is the variance of the overall quantization error [43]. The CORDIC algorithm was implemented in MATLAB and the real component of the output has been depicted in Figure 4.13(a). The SNR equation is also evaluated in MATLAB for different values of n and bb from the above computed worst case error estimates. This

is plotted in Figure 4.12(b). The SNR plots help in making an educated choice in the dimensioning of the CORDIC processor. It is relevant from Figure 4.12(b) that higher the number of iterations in the CORDIC processor and higher the word length used to represent the data signals, more will be the SNR at the output of the CORDIC processor. Besides SNR and least mean square error, the power spectral density of the output is also an important property. The power spectral density estimates the bound for the spurious-free dynamic range(SFDR) which is the ratio of the power of the desired signal and the power of the strongest spur. The CORDIC processor design is tested in MATLAB with a 5 MHz signal and a sampling frequency of 70 MSps. Figure 4.13(b) shows the power spectrum of the CORDIC processor with 14 iterations.



(a) Real part of CORDIC output (b) Power spectral density of CORDIC processor

Figure 4.13: CORDIC output

The spectrum plot shows a peak at 14.5 MHz, which is the CORDIC processor's tuning frequency. The SFDR obtained is about 107 dB. The spurs in the spectrum are due to the effects of finite word lengths used in the fixed point arithmetic. The spurs

generated by the CORDIC processor are mixed with the input signal. If the input signal has more spurs than the CORDIC processor, then the SFDR is determined by the power of the spurs from the input signal.

4.6 Implementation of CIC filter chain

The differential delay (M), rate change factor (R) and the number of stages (N) are the main parameters that determine the response of the CIC filter. The differential delay is either 1 or 2, but the value of rate change factor depends on the application. Due to the unity feedback of the integrators, an overflow is created at the output. This can be resolved by implementing the filter in 2's compliment arithmetic. The CIC filter is implemented and simulations are carried out using the Quartus II 9.0 software in Altera. Figure 4.14 shows the simulation result for a CIC decimation filter. The input wordlength is 16 bits which is the length of the output from the CORDIC processor.

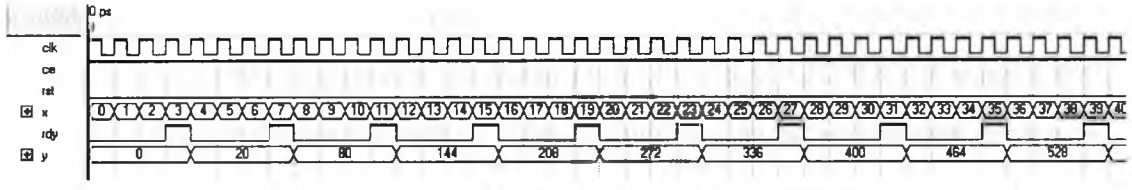


Figure 4.14: Simulation of CIC filter hardware design

The simulation shows that the filter output is generated every 5th cycle. This is indicated with the rdy output signal. CIC filter implementation was carried out in MATLAB to analyze its frequency response. As mentioned earlier, the CIC filter has a droop in the passband which is dependant on the decimation ratio of the filter. The droop is shown in Figure 4.15

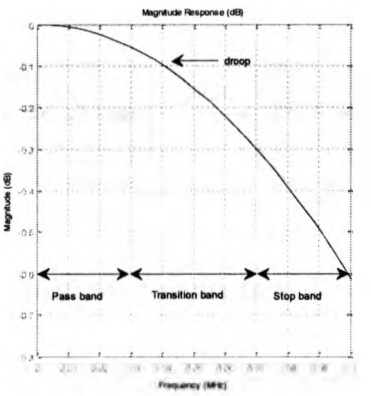


Figure 4.15: Droop in passband of CIC decimation filter

In order to compensate for the droop, CIC filters are usually followed by low pass FIR filters to clean-up or shape the frequency response of the signal [44]. But in this design, instead of immediately following the CIC filter with an FIR filter, two additional CIC filters with a lower order and decimation rates are cascaded with the first CIC filter. The second CIC filter is a two stage filter that operates at a lower speed ,performs the task of compensating for the droop in the passband of the first filter and also further decimates the signal by 2. The third CIC filter is again a two stage filter that performs the task of pulse shaping the signal and further decimates by 2(Figure 4.16). The wordlength throughout the filter chain is maintained at 16 bits.

The advantage of using a CIC compensation filter over an FIR filter at the early stage of sample rate conversion is that no multipliers are required to deal with the high sampling rate. Moreover the filter can be easily implemented since it involves adders and registers alone. The second filter is a low order filter running at a lower sampling rate of f_s/R_1. The third filter operates at a even lower sampling rate of $\frac{f_s}{R_1 R_2}$. The frequency and magnitude response of the single stage and multistage CIC

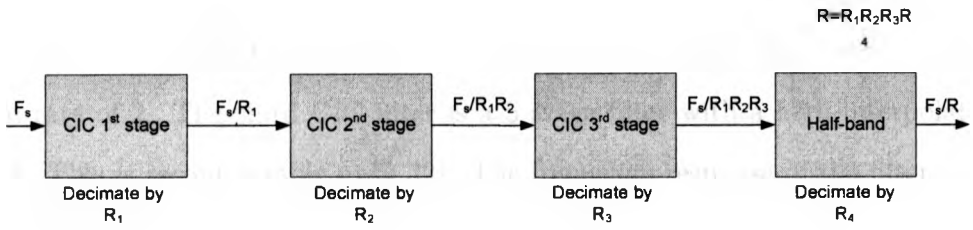


Figure 4.16: Block diagram of multistage decimation filter design

filter design are shown in Figure 4.17. It can be clearly understood from the graph that the high order CIC filter compensates for the droop in the first CIC filter to a certain extent ,thereby giving a good passband characteristics.

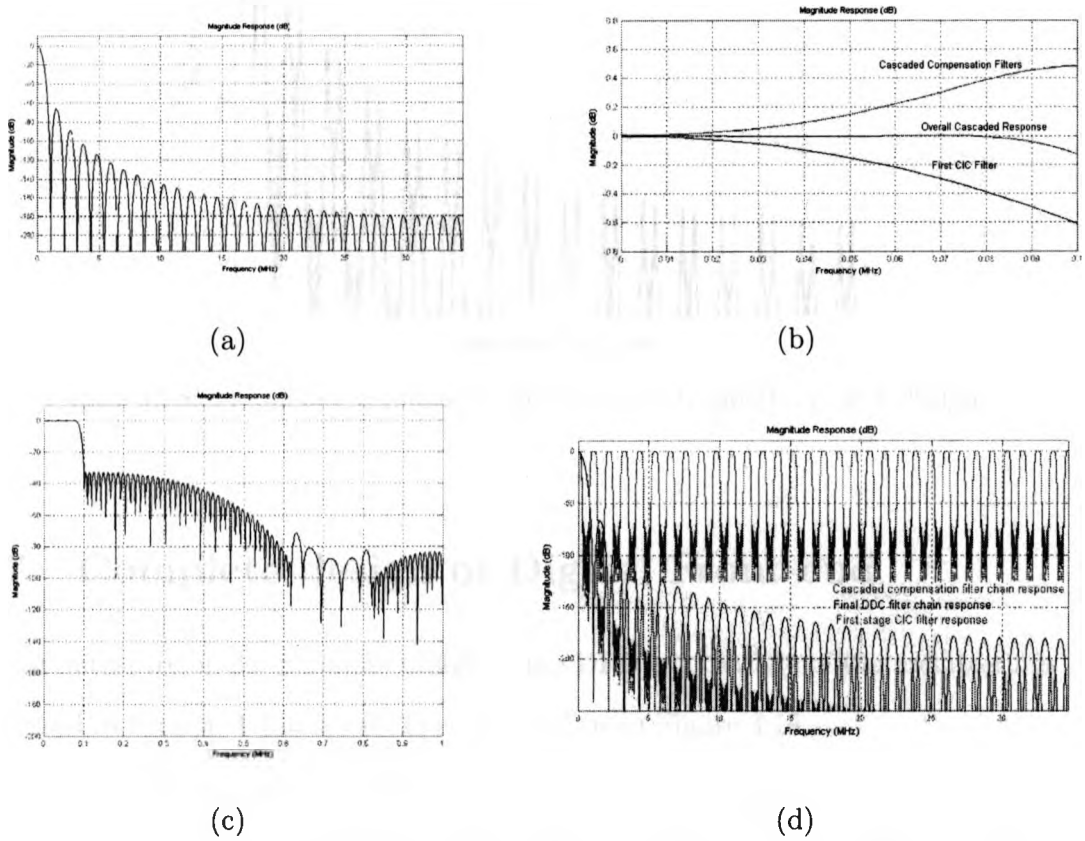


Figure 4.17: Overall filter chain frequency and magnitude results

Similar CIC filters is implemented to perform multi-rate interpolation in the transmitter side. The first and second CIC filters are two stage filters with a interpolation rate of 2. The third CIC filter is a 5 stage filter with a high interpolation rate of 8 (This is reconfigurable upto 32). The frequency response of the filter chain is depicted in Figure 4.18

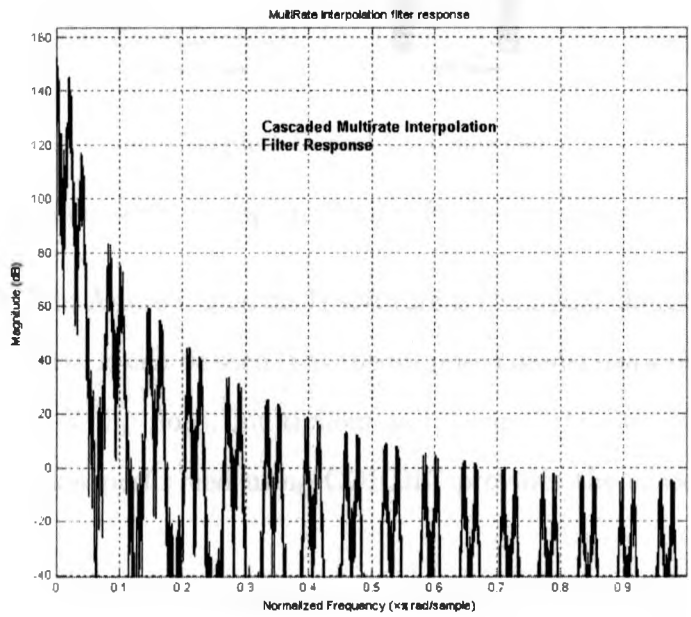
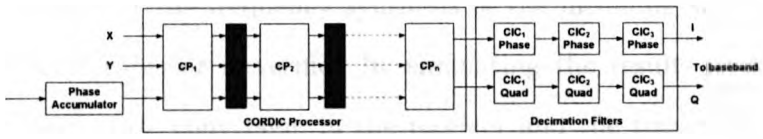


Figure 4.18: Frequency response of multistage interpolation filter design

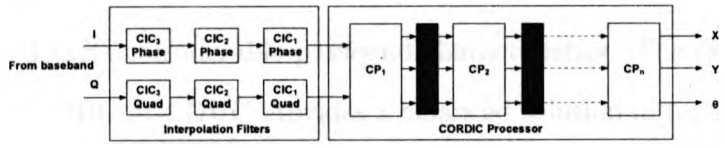
4.7 Complete design of Digital Front-end

The architecture of the complete design of the DFE is a cascade of the architectures discussed in Section 4.4 and 4.6. This is shown as in Figure 4.19

The proposed DDC and DUC models are implemented in the EP2S130F780C5 model of Altera's Stratix II family. The synthesis result of a single channel version of the proposed DDC architecture is shown in Table 4.2. A conventional DDC model



(a) Complete Digital Down Converter



(b) Complete Digital Up Converter

Figure 4.19: Complete Digital Front-End design

using cores available in Altera's Quartus II software is compared along with the design created using the cores. It can be seen that the proposed model has a reduced resource utilization in terms of flip flops, multipliers and memory blocks, at the same the quadrature modulation performed using CORDIC provides the speed in processing the data.

Table 4.2: Synthesis Report for CORDIC DDFS 4.2

Resources	**Proposed Model**	**Model using cores**
Slice LUTs	1272	1142
Flip Flop pairs	1078	1257
Multipliers	10	28
IOs	94	72
GCLKs	1	1
DSP/Memory blocks	0	34

4.8 Conclusion

In this chapter, an area efficient method for implementing a reconfigurable DDC and DUC has been shown. A unrolled pipelined structure of the CORDIC processor has

been implemented for the frequency synthesis of the incoming signal. The working of the CORDIC processor is verified by simulating the results of a feedback loop structure. Sample rate conversion in the receiver and the transmitter is performed using a series of modified CIC filter design followed by a half band filter. The filter chain design reduces the requirement of multipliers, the need for using ROMs to store coefficients and depicts considerable passband characteristics. The overall design of the reconfigurable DDC and DUC provides a balanced solution in terms of speed and area and it proves to be suitable for high speed applications as required in software radio terminals.

Chapter 5
Interfacing High Speed Data Converters to FPGAs

5.1 Introduction

Analog to Digital converters (ADC) and Digital to Analog converters (DAC) are critical components as they are the interface between the analog and digital domains. The continuous demand for higher bandwidths and resolutions in communication,video and other digital systems has propelled the development of high-performance mixed-signal data converters in recent years. This poses some challenges to preserve the signal-to-noise ratio specifications of these devices in the signal processing chain. Modern FPGAs offered by Xilinx like the Virtex-5 and Spartan 3E and the ones from Altera like Stratix-IV and Cyclone-IV provide resources for high-performance mixed-signal systems, supported by efficient development tools spanning all phases of design, from system-level exploration to final implementation [10, 45].

There are two main key trends that impact designers in the design of interface to ADCs and DACs. High sample rates to enable direct up/down conversion is one of them. In order to handle faster data, the sample rates are continuingly being increased by manufacturers. The other trend is that, designers make use of multiple ADCs/DACs per device which allows them to manipulate with the fast incoming data. These trends have led to increased demand in the ADC/DAC interface.

5.1.1 Interface Challenges

Designers commonly face four challenges in the design of FPGA to ADC/DAC interface. Firstly, the conversion from Double Data rate (DDR) to a Single Data Rate(SDR) in LVDS (Low Voltage Differential Signaling) interface design poses a big challenge. An interface operating with DDR transfers data on both the rising and falling edge of the data clock signal [46]. By using both the edges of the clock, the data signals operate with the same limiting frequency but at double the data transmission rate. The second challenge is economic use of LVDS buffers to handle the differential data coming out from the ADC [47]. The requirement for high data rates in FPGAs is the third challenge that designers have to come across. Finally, the tight I/O timing margins and clock domain transfers cause further complications [47]. These challenges have typically channeled designers to choose high performance FPGAs.

In this chapter, the interface of a Texas instruments ADC with serial LVDS outputs to a Spartan 3A FPGA is described in detail utilizing the dedicated deserializer functions of the Virtex-5 FPGA family. Similarly the interface of a Texas Instruments DAC with parallel CMOS outputs is also illustrated. The characteristic features of the Texas instruments ADC ADS62P43 and DAC DAC5687 are described and the possible applications of these converters are mentioned. The digital output design for both Parallel CMOS and serial LVDS interface are depicted. A basic LVDS interface for connecting the ADC converter with high-speed serial interface is illustrated. Similarly a CMOS interface for interfacing the DAC with the FPGA is also shown. Data controller design and implementation is carried out for both the ADC and DAC. The serial peripheral interface controller for both the ADC and DAC is implemented.

5.2 Analog to Digital Converter

The high speed ADC chosen in this thesis is the ADS62P43 from Texas Instruments. It is a dual channel 14-bit A/D converter with maximum sample rates up to 125 MSPS [48]. The ADC uses an internal sample and hold and low jitter clock buffer, the ADC supports high SNR and high SFDR at high input frequencies. The ADS62P43 includes a digital processing block that consists several useful and commonly used digital functions such as ADC offset correction, fine gain correction, decimation by 2,4,8 and in-built and custom programmable filters. By default, the digital processing block is bypassed, and its functions are disabled. The features of the ADS62P43 are summarized below.

5.2.1 Features

- 14 - bit resolution
- No Missing Codes
- Maximum sample rate of 125 MSPS
- Two output interface options - Parallel CMOS and DDR LVDS
- Programmable Course and Fine Gain Options - 3.5 dB Coarse Gain and Fine Gain upto 6dB for better SFDR performance at lower full-scale input ranges
- Digital Processing
- Fine Gain Correction , in steps of 0.05 dB
- Decimation by 2/4/8
- Custom Programmable Low-/High-/Band-Pass filters

- Offset Correction
- Power dissipation of 594 mW
- 71 dB signal-to-noise (SNR) for a 20 MHz interface
- Internal and External references
- 3.3V digital/analog supply

The analog-to-digital conversion is usually the part that limits the performance of the receiver. The most important parameters in choosing this component are resolution, sampling frequency, SNR and SFDR.

5.2.2 Signal-to-noise ratio

ADS62P43 includes gain settings that can be used to get improved SFDR performance (over 0dB gain mode). The coarse gain is a fixed setting of 3.5 dB and is designed to improve SFDR with little degradation in SNR. The fine gain is programmable in 0.5 dB steps from 0 to 6 dB; however the SFDR improvement is achieved at the expense of SNR. So, the programmable fine gain makes it possible to trade-off between SFDR and SNR. The coarse gain makes it possible to get best SFDR but without losing SNR significantly. The ADC resolution is defined as the number of bits at its output, i.e. the size of the binary word which represents the sampled analog signal. There are different error sources in an ADC degrading its performance. When all sources are included, the resolution is usually lower than the specified number of bits of the converter. That is why the effective number of bits (ENOB) of an ADC is such an important parameter and represents the noise-free bits. Table 5.1 shows the SNR , signal-to-noise distortion ratio (SINAD), SFDR and EOB characteristics for different input frequencies [48].

Table 5.1: SNR at different input frequencies

Frequency	SNR	EOB	SINAD	SFDR	UNIT
10 MHz	74.8	-	74.6	93	dBFS
50 MHz	74.4	-	74.2	87	dBFS
70 MHz	73.9	11.3	73.5	89	Bits
190 MHz	71.5	12	71.1	83	dBc

It can be observed that the required values of SNR lowers with increasing input frequency, which is a good reason to choose a high input frequency. Besides,choosing the sampling frequency high would help relaxing the requirements on the analog filters and improve the overall SNR.

5.2.3 Clock input

The clock inputs can be driven differentially (SINE, LVPECL or LVDS) or single-ended (LVCMOS), with little or no difference in performance between them. The common-mode voltage of the clock inputs is set to VCM using internal $5-k\Omega$ resistors. This allows using transformer-coupled drive circuits for sine wave clock or ac-coupling for LVPECL, LVDS clock sources [49]. Single ended CMOS clock can be ac-coupled to CLKP input, with CLKM connected to ground with a $0.1\mu F$ capacitor(Figure 5.1).

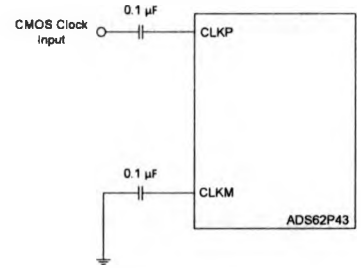


Figure 5.1: Single Ended clock driving circuit

[48]

For best performance, the clock inputs should be driven differentially, reducing susceptibility to common-mode noise. For high input frequency sampling, the clock source is used with very low jitter. Bandpass filtering of the clock source can help reduce the effect of jitter.

5.2.4 Gain and Offset Correction

ADCs generally have small DC offsets in their output. ADS62P43 has an internal offset correction algorithm that estimates and corrects dc offset up to 10 mV [48]. Once enabled, the algorithm estimates the channel offset and applies the correction every clock cycle. The ADS62P43 also has the ability to make corrections to the ADC channel gain in steps of 0.05 dB.

5.2.5 Decimation Filters

ADS62P43 includes option to decimate the ADC output data with in-built low pass, high pass or band pass filters. Decimation rates of 2, 4, or 8 are available and either low pass, high pass or band pass filters can be programmed. The decimation filter is implemented as 24-tap FIR with symmetrical coefficients (each coefficient is 14-bit signed). The in-built filter types (low pass, high pass, and band pass) use pre-defined coefficients. The coefficients can also be set manually.

5.2.6 Applications

The ADS62P43 is intended to be used in several applications

- Software Defined Radio
- Wireless Communications Infrastructure

- Power Amplifier Linearization
- 802.16d/e
- Medical Imaging
- Radar Systems
- Test and Measurement Instrumentation

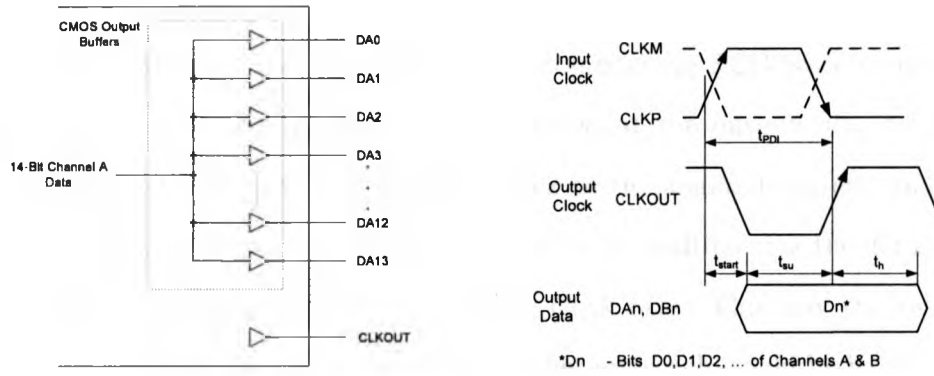
5.3 ADC Digital Output Information

The ADS62P43 provides 14-bit data per channel and a common output clock synchronized with the data. The output interface can be either parallel CMOS or DDR LVDS voltage levels. In the CMOS mode, the output buffer supply can be operated over a wide range from 1.8 V to 3.3 V [48]. Each data bit is output on separate pin as CMOS voltage level, every clock cycle.

5.3.1 Parallel CMOS Interface

In the CMOS mode, the output buffer supply can be operated over a wide range from 1.8 V to 3.3 V. Each data bit is output on separate pin as CMOS voltage level, every clock cycle. (Figure 5.2(a))

The ADS62P43 offers two clock modes of operation using the CMOS interface. In the internal clock mode, the CMOS output clock (CLKOUT) is used to latch data in the receiving chip. The rising edge of CLKOUT can be used to latch data in the receiver, even at the highest sampling speed. It is important to make sure that the output data and the clock traces to the receiver match with each other in order to minimize the skew between them. In the external clock mode, a separate delayed



(a) Parallel CMOS Outputs (b) Parallel CMOS timing diagram

Figure 5.2: Parallel CMOS interface specifications [48]

input clock is used to get desired setup/hold times. The timing diagram for a parallel CMOS interface is shown in Figure 5.2(b). It can be seen that, each data bit is output at every clock cycle,on separate pin as CMOS voltage level. Switching noise (caused by CMOS output data transitions) can couple into the analog inputs during the instant of sampling and degrade the SNR. The coupling and SNR degradation increases as the output buffer drive is made stronger. To minimize this, ADS62P43 CMOS output buffers are designed with controlled drive strength to get best SNR. Inspite of this, the CMOS interface being single ended will have a some amount of noise generated and a lot of capacitance is required to be driven single ended. It is also difficult to achieve the complete operating frequency and a reliable eye pattern. Moreover, additional efforts have to taken to align the phase clock with the CMOS data.

5.3.2 DDR LVDS Interface

ADCs need to drive the receiving logic and accompanying PCB trace capacitance. Usually current switching transients are developed at the outputs due to driving the load. These currents can switch reflect back to the analog front end and cause performance issues. One way to reduce this effect is by multiplexing the data output ports , thereby providing the data at half the clock rate. This way the switching time between the transients are increased. Another approach would be having LVDS outputs. Although the CMOS interface seems straight forward, there will be problems in terms of the performance of the ADC. Even after precautionary measures, CMOS interface generates a lot of noise and a large capacitance has to be driven single ended [50]. On the other hand, the DDR LVDS interface enables to achieve high data rates, low power, a low cost and does not depend on a specific power supply.

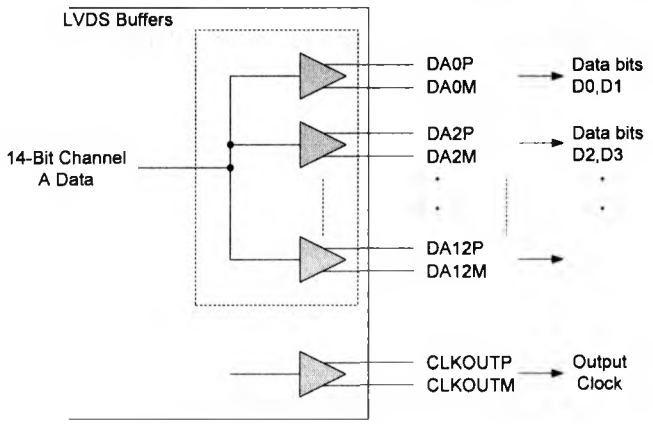


Figure 5.3: DDR LVDS Outputs
[47]

The switching times are reduced with lower voltage signal swings and also EMI concerns are minimized[50]. With LVDS signaling, the common mode voltages are

also rejected. Unlike CMOS outputs, LVDS outputs require a resistor termination at the receiver (100Ω). As a result, a fixed dc current is supplied on the outputs, thereby shunning current spikes. The chosen ADS62P43 provides 14-bit data per channel and a common output clock synchronized with the data. Figure 5.3 shows the DDR LVDS outputs from the ADC. A bank of six LVDS output buffers are employed for a single channel, with each buffer streaming out two data bits in the same clock. As shown in Figure 5.3, the first LVDS buffer gives D0 and D1 bit pair, the second buffer gives D2 and D3 and so on. A global clock buffer is used to produce the differential clock signal CLKP and CLKM.

5.4 Digital to Analog Converter

Texas Instruments DAC5687 is used for converting high speed digital signals to analog. The DAC5687 is a dual channel 16-bit high speed digital-to-analog converter with a sample rate of 500MSPS. Some of the features are listed below [51]

5.4.1 Features

- 16 - bit resolution
- Maximum sample rate of 500 MSPS
- Flexible input options (Even/Odd Multiplexed, Single port demultiplexed)
- Complex Mixer with 32 bit NCO
- Digital Processing
- Fine Gain Correction , in steps of 0.05 dB

- Full IQ Compensation
- Selectable 2x-8x Interpolation of Data
- Custom Phase correction
- Offset Correction
- 71 dB signal-to-noise (SNR) for a 20 MHz interface
- Internal and External references
- 1.8V or 3.3V digital/analog supply

5.4.2 Input FIFO

DAC5687 offers an optional input FIFO that allows the latching of data on both the channels based on the user provided clock CLK1 or the input data rate clock provided to the PLLLOCK pin. The input interface FIFO incorporates a four-sample register file, an input pointer, and an output pointer. Initialization of the FIFO pointers can be programmed to one of seven different sources.

5.4.3 DAC Clock Modes

The DAC5687 has mainly three clock modes for generating the internal clocks for the logic, FIR filters and DACs. These are the external clock mode, PLL clock mode and the dual clock mode. In the external clock mode, the clock signal is provided by the user at the DAC output sample rate through the external clock input CLK2. The input data-rate clock and the interpolation rate in the filters are selected through register values and the output is through the PLLLOCK pin. The PLLLOCK clock can be used to drive the input data source that sends the data to the DAC. The

input data is latched on either the rising or falling edge of PLLLOCK, which is sensed internally at the output pin. In the PLL clock mode, the DAC is driven at the input sample rate through the PLL clock input CLK1. In this case, there is no phase ambiguity on the clock. The DAC generates the higher-speed DAC sample-rate clock using an internal PLL/VCO. In PLL clock mode, the user provides a differential external reference clock on CLK1. The reference clock is compared with a feedback clock(selected by one register and is used for synchronization to the input clock) using a phase-frequency detector which drives the PLL to maintain synchronization between the two clocks. The feedback clock is generated by dividing the VCO output by 1, 2, 4, or 8. The feedback clock is also used for the data input rate, so the ratio of DAC output clock to feedback clock sets the interpolation rate of the DAC5687. The PLLLOCK pin is an output indicating when the PLL has achieved lock. In the dual clock mode, the DAC is driven at the DAC sample rate through CLK2 and the input data rate through CLK1. The dual clock mode offers the option of enabling or disabling the input FIFO. If the FIFO is not used, the CLK1 input is used to set the phase of the internal clock divider. If FIFO is used, CLK1 is used as an input latch and CLK2 is used to generate the internal divided clock.

5.4.4 Modes of operation

The DAC5687 has six digital signal processing blocks:

- FIR1 and FIR2 - Digital FIR filters are enabled to interpolate-by-two
- FMIX - Fine frequency mixer which uses a NCO to provide sin and cos for mixing

- QMC - Quadrature modulation phase correction that provides means for changing the phase balance of the complex signal to compensate for I and Q imbalance present in an analog quadrature modulator
- FIR3 - Third interpolate-by-two digital filter
- CMIX - Coarse frequency mixer that provides mixing capability at the DAC output rate with fixed frequencies. The output of the CMIX block is complex

5.4.5 Digital Data and Clock Inputs

The DAC5687 has straight forward CMOS inputs for data with internal pulldown resistors. The clock circuit has various input configurations for driving different clock input.

5.5 Implementation of a single channel LVDS Interface

For a single channel LVDS design, the ADC is made to run at 20 MHz with 14-bit serial LVDS interface and in Dual data rate mode. The ADS62P43 transmits edge-aligned data and sync signals with a 90-degree phase shift clock [48]. A digital clock manager(DCM) is used in fixed phase shift mode to register the received data into the FPGA with the receive clock. If there is any nonalignment with the data and the clock, then the DCM can switch to dynamic phase shift mode.

5.5.1 Single Channel Interface

The timing diagram and characteristics for a single channel interface are shown in Figure 5.4.

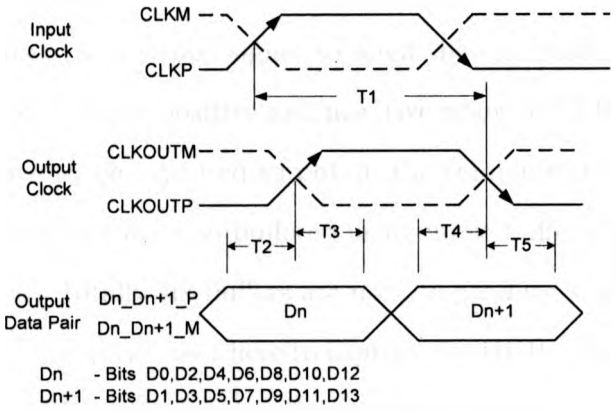


Figure 5.4: DDR LVDS Timing
[48]

Table 5.2: LVDS interface timing characteristics

PARAMETER	MAX	TYP	MIN	UNIT
T1 - Clock Propagation Delay	5.8	7.3	8.8	ns
T2,T4 - Data Setup Time	4.3	5.8		ns
T3,T5 - Data Hold Time	4.2	5.7		ns

The ADS62P43 has two LVDS outputs, each one providing a serial 14-bit data bitstream with MSB first. A differential pair LVDS high speed clock (CLKM and CLKP) are also provided. The LVDS interface works only with 3.3-V supply.A digital clock manager(DCM) is used in fixed phase shift mode to register the received data into the FPGA with the receive clock. If there is any nonalignment with the data and the clock, then the DCM can switch to dynamic phase shift mode.

The diagram of the single channel LVDS interface is shown in Figure 5.2. The differential clock pair CLKP and CLKM from the ADC are buffered using a IBUFDS design element (available in Xilinx ISE library), to generate a single 20 MHz clock. This 20 MHz clock is used as the interface clock and is fed into the DCM. The DCM generates two phase aligned clocks, CLK0 and CLK180. The data aligned version of the LVDS clock is used as a strobe signal to align the captured 14 bit words. The strobe signal is sampled at the positive and negative edges of CLK0. Both the rising and falling edges have to be captured to obtain the complete set of data bits. The differential pair of data bits are also buffered using input buffer element IBUFDS to generate DDR signals. Totally six buffers are used to produce six DDR signals.

The primitive component used here to capture the DDR signals is IDDR2 available in the Spartan3A FPGA. It consists of a pair of storage elements (IFF1 and IFF2) which allows an I/O to receive a DDR signal[52]. The incoming DDR clock signal activates one register and the inverted clock signal activates the other register. The IOB registers the incoming data on the rising edge of CLK0(D1) and the rising edge of CLK180(D2), which is basically the same as the falling edge of CLK0. Normally, odd data bits D1,D3,D5,D7,D9 are clocked on the rising edge of CLK180 and even data bits D0,D2,D4,D6,D8,D10 are clocked on the rising edge of CLK0. This data is then transferred to the FPGA fabric. At some point, both signals must be brought to the same clock domain, typically CLK0.

The IDDR2 component provides the option of cascading it's storage elements with those in other IOB having a differential pair. Here, the cascade feature is employed with DDR aligned to CLK0. All the data is then re-registered to CLK0 and then only fed to the FPGA fabric.

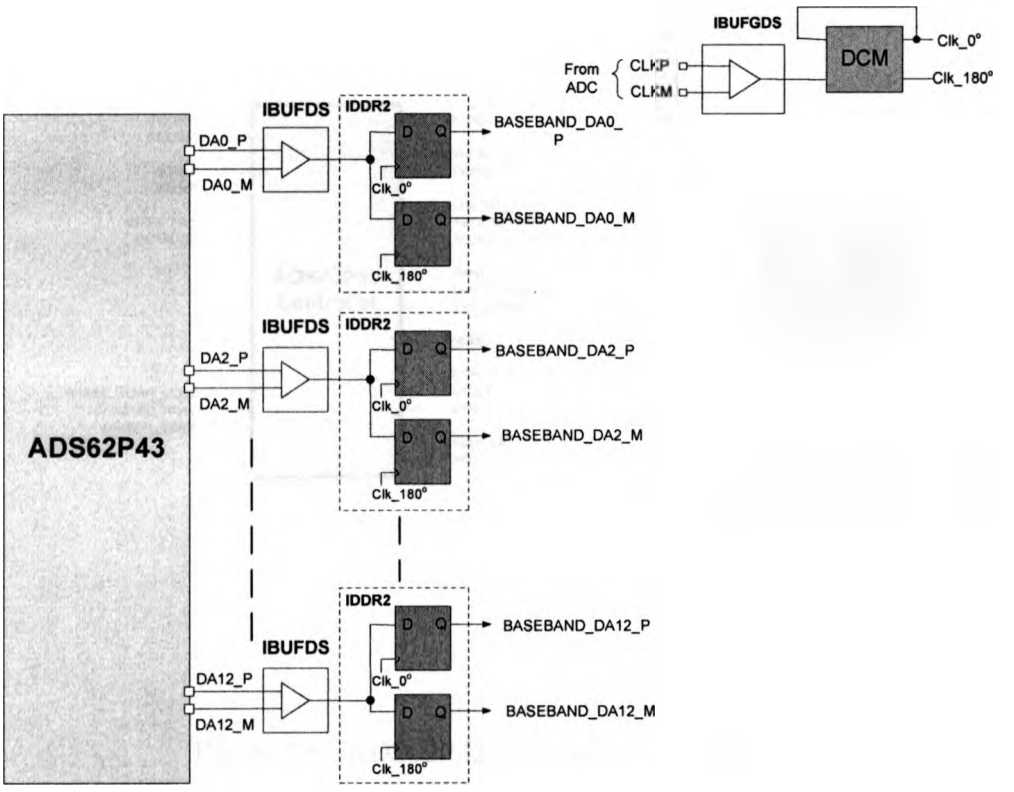


Figure 5.5: 14 bit single channel receiver

5.6 ADC Data Controller Design

In this section , the implementation of the interface controller for the ADS62P43 is shown. The ADC provides several modes of operation which needs to be controlled to ensure the normal functioning of the ADC. Two finite state machines(FSM) are implemented to traverse through the different modes of operation. One FSM carries out all the control of all operation modes and data flow in the ADS62P43. The second FSM is an internal SPI FSM that implements the SPI communication between the ADC and FPGA. Figure 5.6 shows the block diagram of the top level implementation

of ADC controller.

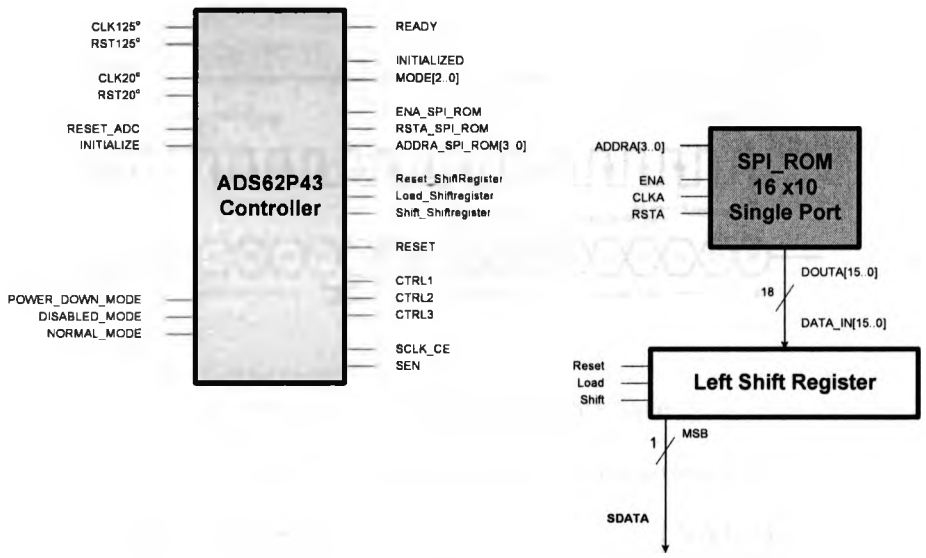


Figure 5.6: ADS62P43 Data Interface Controller

The controller generates the necessary control signals for SPI communication and mode switching in the ADC. The ADC's register values are stored in a 16x10 ROM. The serial data from the ROM are read via a left shift register. The ADS62P43 allows it to be programmed through Serial Peripheral Interface(SPI). This is done by accessing the pins SCLK(Serial Interface Clock), SEN(Serial Interface Enable) and SDATA(Serial Interface Data). The SPI timing diagram is shown in Figure 5.7.

The SEN signal enables the shifting of serial bits into the device when it is low. When this happens,serial data SDATA is latched at every falling edge of SCLK. SDATA is loaded into the register at every 16th SCLK falling edge when SEN is low(Figure 5.7). In case the word length exceeds a multiple of 16 bits, the excess bits are ignored. Data can be loaded in multiple of 16-bit words within a single active

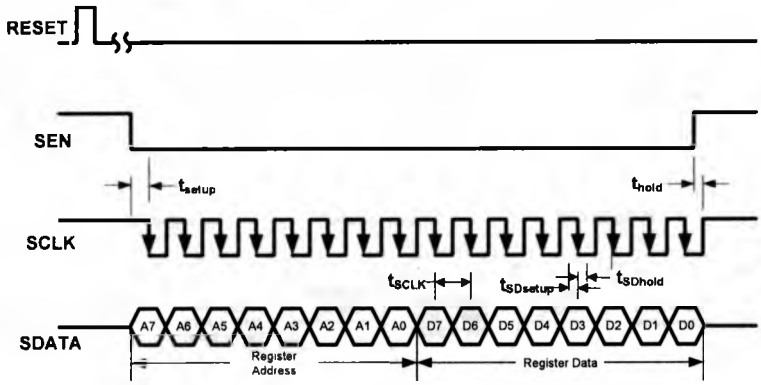


Figure 5.7: Serial timing interface

Table 5.3: Serial interface timing characteristics

PARAMETER		VALUE
f_{SCLK}	SCLK frequency	20(MHz)
t_{setup}	SEN to SCLK setup time	25(ns)
t_{hold}	SCLK to SEN hold time	25(ns)
$t_{SDsetup}$	SDATA setup time	25(ns)
t_{SDhold}	SDATA hold time	25(ns)

SEN pulse. The first 8 bits form the register address and the remaining 8 bits the register data. The serial interface timing characteristics are summarized in Table 5.3.

The interface can work with SCLK frequency from 20 MHz down to low speeds (few Hertz).

Figure 5.8 shows the state diagram of the top level FSM. Before starting any data transaction, the ADC is reset. Therefore , the first mode of operation in the FSM is reset. No mode of operation is assigned during this stage. Along with the ADC, the 125 MHz clock source from the FPGA, the counters and the control registers are reset. There are two delays to be considered while resetting the ADS62P43. The first delay is a 5ms delay from power-up of the AVDD to Reset active signal. This

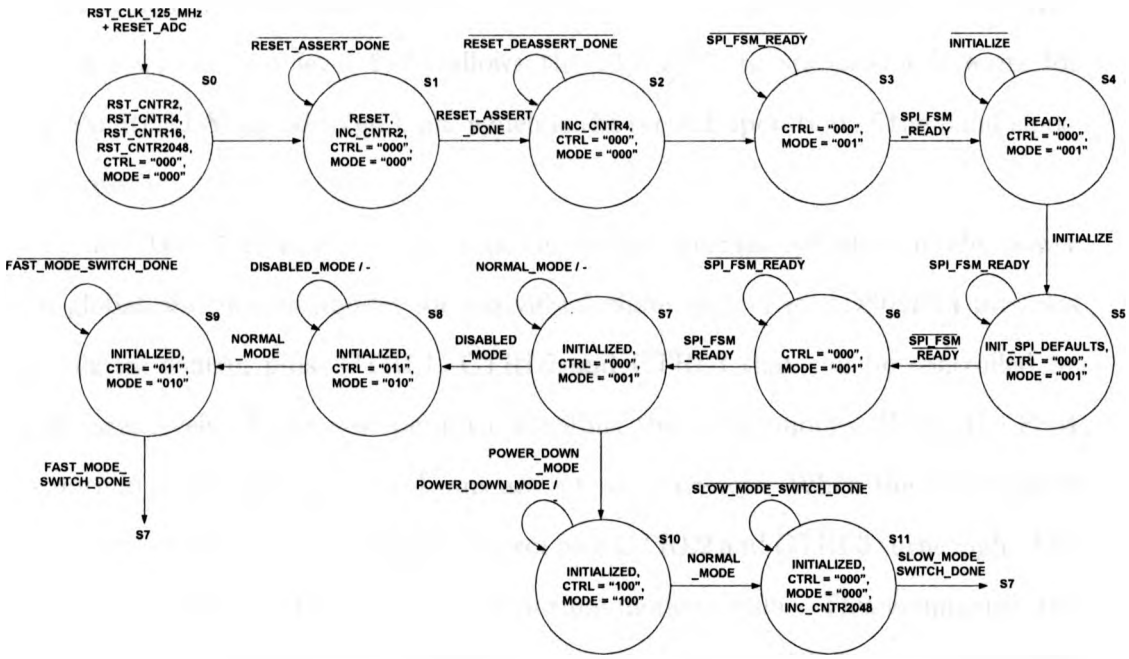


Figure 5.8: ADS62P43 Interface Controller State Machine

delay is implemented using a mod 2 counter. The FSM waits on the mod 2 counter for the RESET_ASSERT signal to be done. Once this happens, another counter is activated for the second delay which occurs from the time the reset signal is disabled and the SEN signal is made active. This is a 25 ns long delay and is implemented in the FSM using a Mod 4 counter. The FSM then waits for the second SPI FSM to complete one SPI transaction. Once this is done, the FSM waits to be initialized and the mode of operation is changed to normal mode. Upon initialization, initial SPI default values are assigned to the registers. Again the top level FSM waits on the SPI FSM to complete the cycle of loading the registers with initial values via SPI.

A hand shaking process is executed whenever the SPI FSM is prompted to perform a transaction. The process has two operations. First the top level FSM waits for the SPI_FSM_READY signal to go low during which the SPI FSM is executed

and initial default values are loaded into the registers. Once the SPI_FSM_READY signal is low, the top level FSM allows the SPI FSM to stabilize and waits for SPI_FSM_READY signal to go high, which is the second operation of the hand shaking process.

The ADS62P43 has three main power modes- normal operation mode, power down global and disable mode(for individual channels). The ADS62P43 provides three digital control pins- CTRL1, CTRL2 and CTRL3 that can be controlled by digital logic levels. These pins permit controlling the power modes. When the FSM is in normal mode, all the control pins are set to active low. When the stage moves from normal mode to disabled mode, control pins CTRL2 and CTRL3 turns high. The wake up time from the disabled mode to normal mode is 100ns. This is implemented with a mod 16 counter. The FSM waits on the signal FAST_MODE_SWITCH_DONE for the mode switch to take place. Once this is done , the state moves back to normal mode. When a global power down is activated on the ADC, control pin CTRL1 is driven high. The wake up time from power down mode to data becoming valid in normal mode is slow with 15μS. Here, the FSM waits on a SLOW_MODE_SWITCH_DONE signal to move from power down mode back to normal mode. A mod 2048 counter is implemented to execute the slow wake up time.

5.6.1 Serial Peripheral Interface of ADC

As mentioned earlier, the SPI FSM is an internal process which is implemented to execute all SPI transactions(Figure 5.9).

In the first stage of the SPI FSM, reset is applied to the 20 MHz clock source, to all the counters , the SPI ROM and the shift register. Once the reset is deactivated, the SEN signal is driven high and initial default values are loaded. The ROM input

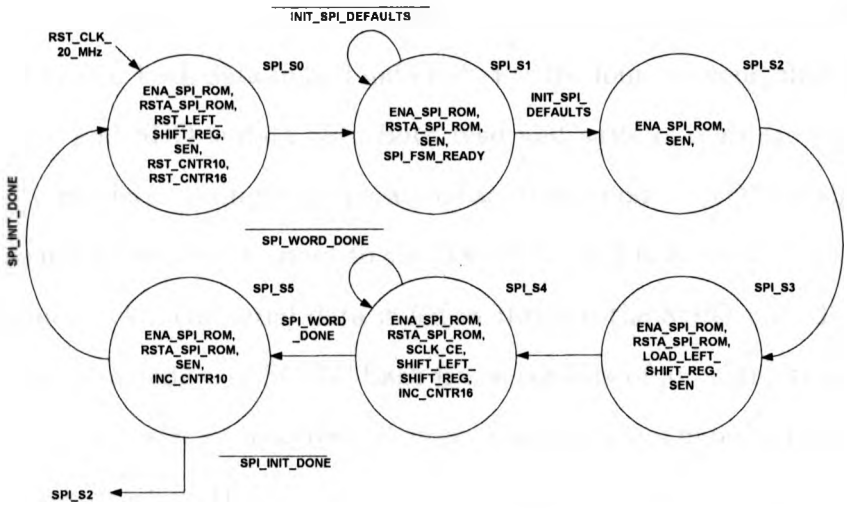


Figure 5.9: ADS62P43 SPI Controller State Machine

signals are also generated at the same time. After this, the necessary SPI transaction is carried out by reading the data serially from the ROM by loading the left shift register. The SPI FSM waits on the signal SPI_WORD_DONE for SCLK enable to go high and for the data to be shifted. The serial data is 16 bits long , so a mod 16 counter is implemented to drive SPI_WORD_DONE low. There are a total of 10 registers each with 16 bit values. So the FSM is made to wait on a mod 10 counter for all the values to be loaded. After this, SPI_INIT_DONE signal is driven high to indicate that all the data from the ROM is serially read out and the FSM is ready for the next transaction.

5.7 DAC Data Controller

The DAC controller is also implemented using two FSMs - a top level FSM and a SPI FSM. The SPI communication in the DAC5687 can be configured as a three-pin

or a four-pin interface. In both the configuration, SCLK is the serial interface clock, SDENB is the serial interface enable. For the three pin configuration, SDIO is the bidirectional pin for both data in and data out. For the four-pin configuration, SDIO is data in only and SDO is data out. Both read and write options are provided by the DAC SPI interface. Both these operations are framed on the SDENB signal. The SDENB signal is an active-low input to the DAC5687. SCLK is provided as the serial interface input clock. The serial data is fed in through the SDIO pin. The data is clocked on the rising edges of SCLK. Each frame consists of an instruction cycle describing the data transfer cycle as read or write , the number of bytes to be transferred and the address to which the data has to be transferred. For a three-pin configuration , the SDIO is data out and SDO pin is in high impedance state. For a four-pin configuration, the SDO pin is data out pin. The DAC5687 provides three clocking modes - EXTERNAL CLOCK MODE, PLL CLOCK MODE and a DUAL CLOCK MODE. In this implementation , the PLL CLOCK MODE is adopted. Here the DAC is driven at the input sample rate through one clock channel(CLK1/CLK1C), and the other clock input channel is not used. In this case, there is no phase ambiguity on the clock. The DAC generates the higher-speed DAC sample-rate clock using an internal PLL/VCO [51].

Figure 5.10 shows the top level FSM for the DAC controller. In the initial state, the DAC and the counters used for implementing the delays are reset. The top level FSM loops around till the RESET_ASSERT_DONE is not turned high. The reset in the DAC5687 is active-low signal and it's hold time is for 50 ns. This is implemented using a mod 7 counter. Once the RESET_ASSERT_DONE signal is high, the top level FSM waits on the SPI FSM , as mentioned previously for the ADC controller, to carry out the reset transaction and waits till the system is initialized.

Upon initialization, the initial default values are loaded into the registers. A

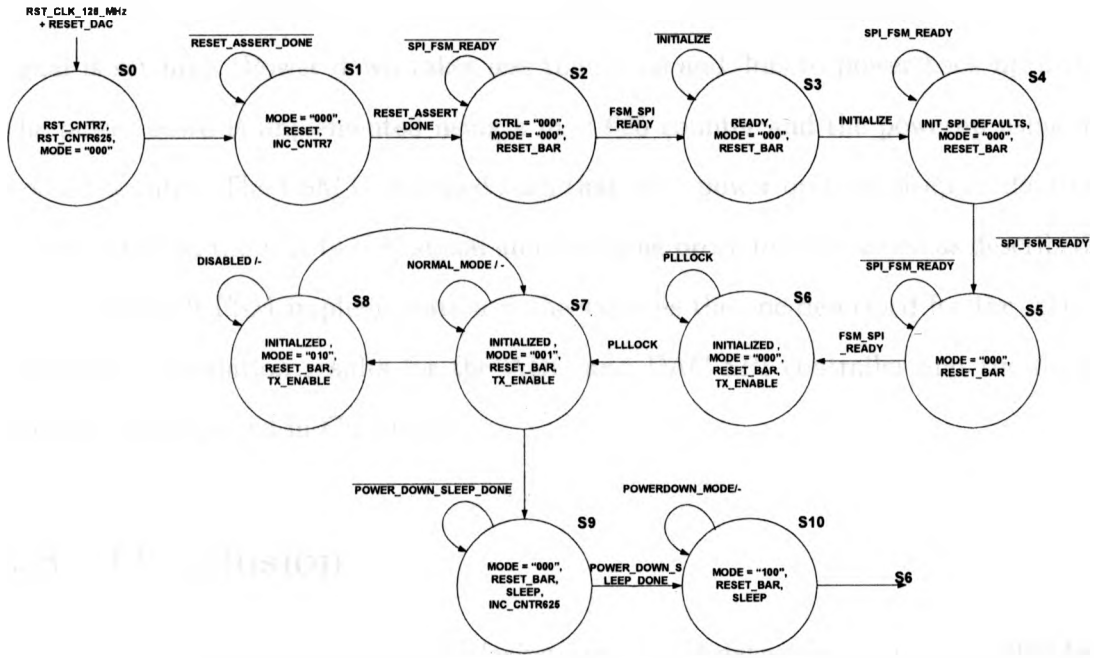


Figure 5.10: DAC5687 Controller State Machine

similar hand shaking process as implemented in the ADC controller is executed here to synchronize the flow of information between the top level FSM and the SPI FSM. Once the FSM is done with the hand-shaking, it waits for the PLL to maintain synchronization between the reference clock provided and the feedback clock generated within the PLL. When the PLLLOCK signal is set high, the FSM moves to the normal mode of operation.

The DAC5687 also has the same three operating modes - normal mode, disabled mode and power down mode. When in normal mode, the system is initialized and TXENABLE signal is set high. TXENABLE must be high for the DATA to the DAC to be enabled. When TXENABLE is low, the digital logic section is forced to all 0, and any input data presented to the both data channels are ignored. To switch from normal mode to disabled mode, the TXENABLE signal is set to low. No delay

occurs during this time, so no counter is used. To enter the sleep mode, the reset signal is set high. Power down takes less than 5 μs and 3ms to power back up [51]. The power down is implemented using a mod 625 counter and the powerup using a mod 32 counter. The FSM is designed such that after power up from sleep mode, the system waits on the PLLLOCK signal and the same procedure is carried as described above. The SPI FSM implementation is the same as the one described for the ADC controller. Simulation results for the ADC and DAC SPI controller and the data interface are depicted in Chapter 5.

5.8 Conclusion

This chapter mainly focuses on interfacing high speed data converters with FPGAs in the receiver and the transmitter sections. The digital specifications of the chosen ADC ADS62P43 and DAC DAC5687 have been discussed. A detail explanation of an LVDS interface between ADS62P43 and a Spartan 3A FPGA has been described. An SPI control entity is implemented in the Spartan 3A FPGA to control the registers of the ADC. For the DAC, a CMOS interface with Virtex-5 FPGA has been designed. SPI control entity for the DAC5687 is also implemented.

Chapter 6

Prototype Model of RF Front-End and Results

So far we discussed about data converters and the intermediate section of a software radio. This chapter now discusses the direct conversion RF transceiver that was used in the experimental setup. The transceiver, like the data converters is also controlled via SPI. A controller for the transceiver evaluation board is also designed and implemented in this chapter. Finally the complete RF front-end experimental setup is depicted and simulation results are presented.

6.1 MAX2830 RF transceiver

The MAX2830 is a direct conversion RF transceiver. This means no IF stage is required in the architecture chain. It is specifically designed to operate using the 802.11 g/b WLAN protocol for the frequency range of 2.4GHz to 2.5GHz. The biggest advantage with the MAX2830 is that it integrates all the necessary circuitry required to implement the RF transceiver. It includes a RF power amplifier next to the antenna port in the transmitter side and a low noise amplifier in the receiver side. It includes an Rx/Tx and antenna diversity switch to change between receive and transmit modes. It also provides a voltage controlled oscillator, frequency synthesizer, crystal oscillator, RF-to-baseband for receive path and baseband-to-RF transmit path and a baseband

control interface [53]. The MAX2830 also provides better performance by providing I/Q calibration,DC-offset cancellation and error and current leakage circuits. It also implements on-chip filters in both receiver and transmitter. These devices are suitable for the full range of 802.11g OFDM data rates (6Mbps to 54Mbps). The MAX2830 is widely used in Wi-Fi and PDA mobile handsets, wireless speakers and headphones and other general purpose radios.

A Rx/Tx switch is integrated with an antenna diversity switch in both the receiver before the LNA and the transmitter after the PA in the MAX2830. Two antenna ports are provided and a set of two pins(Rx/Tx pin and antenna pin) decide which ports are utilized in the radio transmission. An option for disabling a antenna port and using a single port is also provided for resource management. The switches are followed by the LNA and VGA that digitally program the gain control range. There are three gain modes offered in the MAX2830: max gain, max gain -16dB and max gain -33dB [53]. These modes are programmed using SPI interface. These are connected to analog down converters that directly frequency translate from RF to baseband. Sample rate conversion is then carried out on the down converted signal using I/Q baseband low-pass filters. These filters provide an upper -3dB corner frequency of 8.5Mhz with 50dB of attenuation at 20MHz [53]. Further the MAX2830 allows coarse fine tuning of the -3dB corner frequency to suit different standards. An analog received signal strength indication(RSSI) is also provided which measures the power present in the carrier signal. This provides an effective means of calibrating the system model. An analog voltage proportional to the log of the sum of the squares of the I and Q channels, measured after the receive baseband filters and before the VGA. The MAX2830 receiver provides both AC and DC coupling of the I/Q baseband signals. LO leakage and other DC offsets are removed by the AC coupling. Attenuation of the received signal is avoided using the DC coupling. The baseband

outputs are internally biased to a common-mode voltage of 1.2V and are intended to be DC-coupled to the in-phase and quadrature ADC inputs of the accompanying baseband IC [53].

The transmitter side of the MAX2830 is populated with low pass baseband filters, direct up-conversion mixers, a VGA, a PA driver and a linear RF PA with power detector. The differential inputs to the transmitter require a common-voltage of 0.9V to 1.3V [53]. This is intended to be provided by the DC coupled DAC outputs of the accompanying baseband IC. The up-converted signal is filtered with low-pass filters that can be tuned to -3dB corner frequency for different standards. The VGA of the transmitter provides 31dB of gain control range which is programmable via SPI interface. The RF signals are then amplified using the PA which provides +17.1 dBm of output power the PA bias currents and enable delay can be set via SPI according to the requirements. The PA output is integrated with a power detector to provide an analog voltage proportional to the PA output. The phase noise in the system is compensated using a 20-bit sigma-delta frequency synthesizer. The synthesizer provides options to divide the signal frequency by 1 or 2 with programmable divisor and fractional range. The integer and fractional divider ratios can be calculated as

$$LOfrequencydivider = \frac{f_{RF}}{f_{COMP}} \tag{6.1}$$

where f_RF is the RF frequency(2437MHz) and f_COMP is the comparison frequency(20MHz for nominal 802.11 g/b).

The 40 MHz crystal oscillator included can be made to function with a external reference frequency source. In such a case, the crystal oscillator acts as a buffer and the reference signal has to be AC coupled. The reference oscillator has a divider and a buffered output for routing the reference clock to the accompanying baseband IC.

The PLL in the circuit is constructed by connecting the PLL charge-pump output to a lowpass RC loop filter. The RC filter output is connected to the voltage tuning input of the VCO. A bandwidth of 150kHz is provided by the loop filter which ensures to achieve the desired Rx/Tx turn-around settling time, good stability and phase noise. The PLL also features a lock detect option which indicates when the PLL is locked.

6.2 MAX2830 Controller

As mentioned earlier, the MAX2830 allows Master-Slave SPI communication for it to be controlled by external circuits. Here a data controller similar to the controllers designed for the ADC and DAC is implemented to read and write instructions from and to the MAX2830. Figure 6.1 depicts the top level design of the data controller.

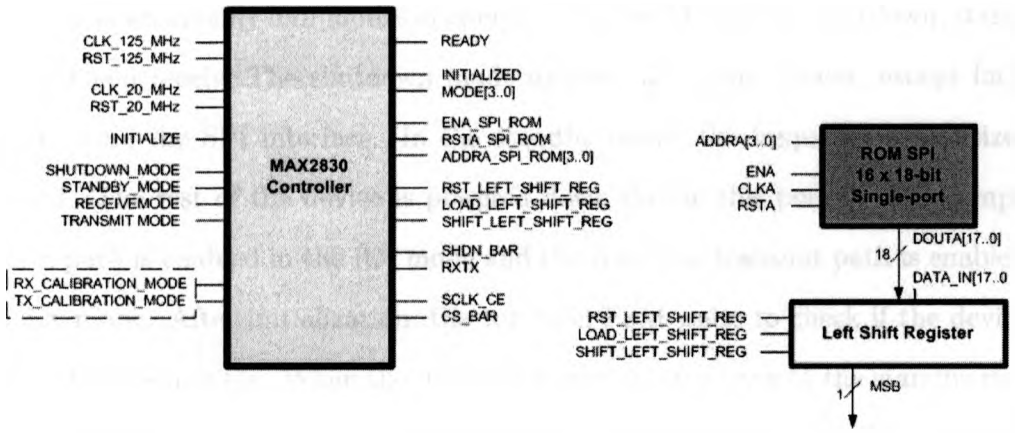


Figure 6.1: MAX2830 transceiver top level controller

The controller for the MAX2830 is implemented using two FSMs similar to the ones used for the ADC and DAC. Like the procedure for any other system, the data controller for the MAX2830 is reset first by resetting the user clock. For a clock period of 8ns, the reset signal is held high for 8μs. This is implemented in the controller using

a mod1024 counter. At this point, the FSM waits on the SPI_READY signal from the second SPI FSM to indicate that it's ready for initialization. Now the system waits till it is loaded with the SPI defaults values. These register values are read from a single port ROM.Once the counter is done counting and brought to zero, the system is initialized. A hand shaking process, similar to the process involved in the ADC's controller design, is executed whenever the SPI FSM is prompted to perform a transaction. First the top level FSM waits for the SPI_READY signal to go low during which the SPI FSM is executed and initial default values are loaded into the registers. Once the SPI_READY signal is low, the top level FSM allows the SPI FSM to stabilize and waits for SPI_READY signal to go high, which is the second operation. Now the system is initialized and is in the normal mode of operation. See Figure 6.2 for the top level FSM design.

There are mainly four modes of operation for the MAX2830: shutdown, standby, transmit and receive.The shutdown mode disables all circuit blocks, except for the registers and the SPI interface. In the standby mode, the frequency synthesizer is enabled. The rest of the device is powered down during this period. The complete receive path is enabled in the RX mode and the complete transmit path is enabled in the TX mode. After initialization, the top level FSM waits to check if the device is still in shutdown mode. When the device is turned on, it moves to the standby mode. All the components that permit transmit and receive of data like the PLL , VCO and LO are left to be turned on so that TX and RX modes can be easily enabled from here. If the system has to move to the receive mode, a turn-on time of $1.9\mu s$ is implemented using mod 1024 counter. The system waits on the MODE_SWITCH_DONE signal to go high. Once the mode switching is done, the system is ready to receive data. If the system has to move back to standby mode, a turn-off time of $0.1\mu s$ is implemented using a mod 1024 counter. For transmit mode , the turn-on time from standby mode

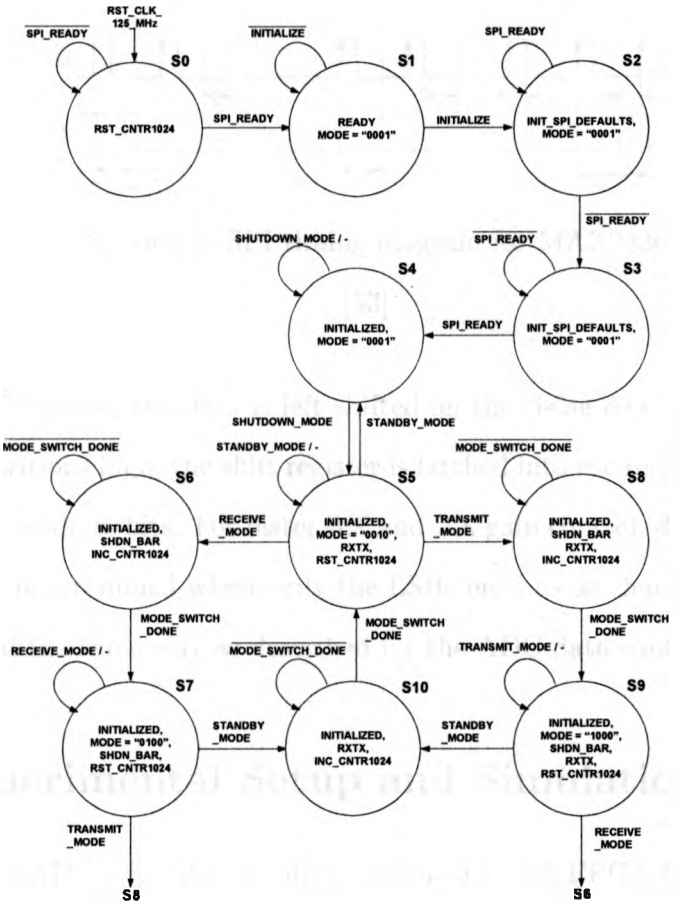


Figure 6.2: Finite State Machine for MAX2830 data controller

is 1.5μs and turn-off time is 1μs.

The MAX2830 includes 16 registers with 18 bit word-length. The 14 MSBs are used to represent the data and the 4 LSBs contain the register address. The data is captured on the active low signal $\overline{CS}$ which acts like the serial enable signal.

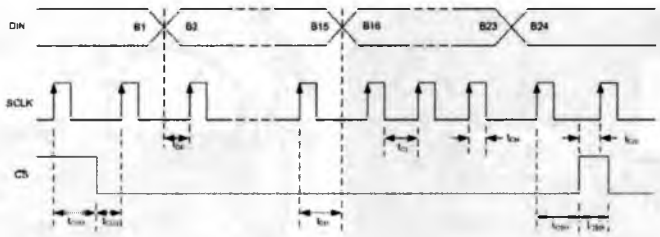


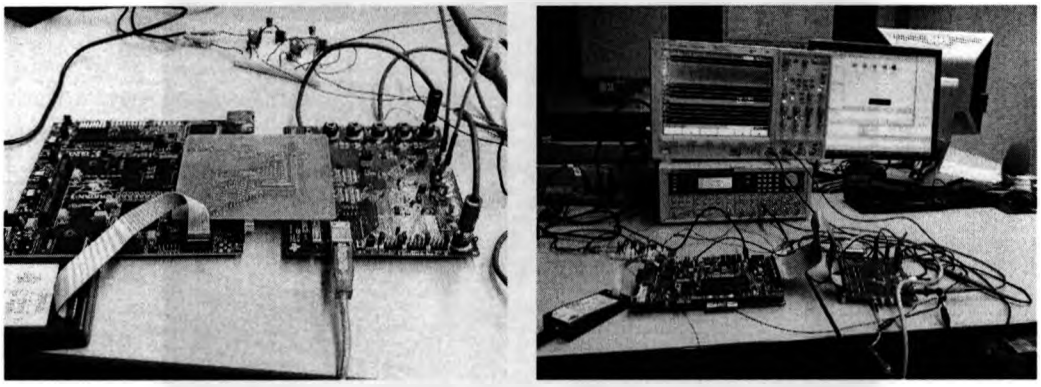
Figure 6.3: SPI timing diagram for MAX2830

[53]

When $\overline{CS}$ is low, the data is left shifted on the rising edge of the active clock. When CS transitions high, the shift register is latched into the register selected by the contents of the address bits. For faster RX and TX gain control, data words less than 14 bits can be programmed where only the LSBs need to be changed. The SPI FSM is implemented the same way as described for the ADC data controller in chapter 4.

6.3 Experimental Setup and Simulation Results

The ADS62P43 ADC is interfaced with a Spartan 3A 1800 FPGA board via SAMTEC LVDS headers. A PCB was designed and fabricated to physically connect the ADS62P43 evaluation board and the FPGA board. A similar setup for interfacing the DAC5687 with Virtex 5 was carried out. The physical connection was made using the CMOS pins in the boards via ribbon cables. The experimental set-up of the ADC-FPGA interface and DAC-FPGA interface are shown in Figure 6.4(a)

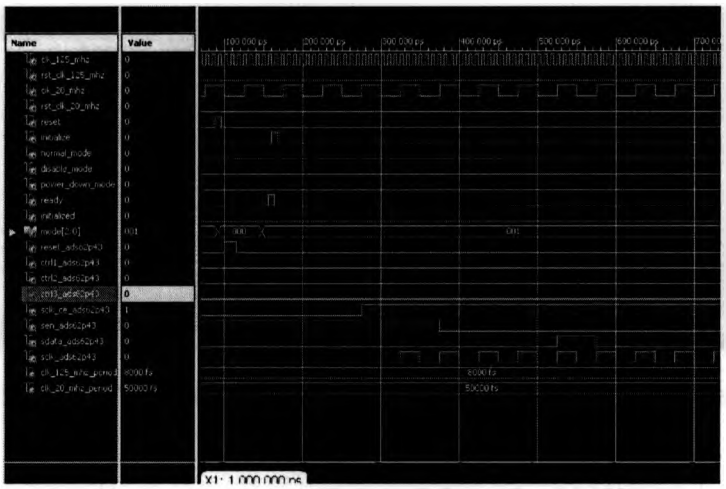


(a) Experimental set-up for interfacing ADS62P43 with Spartan 3A FPGA

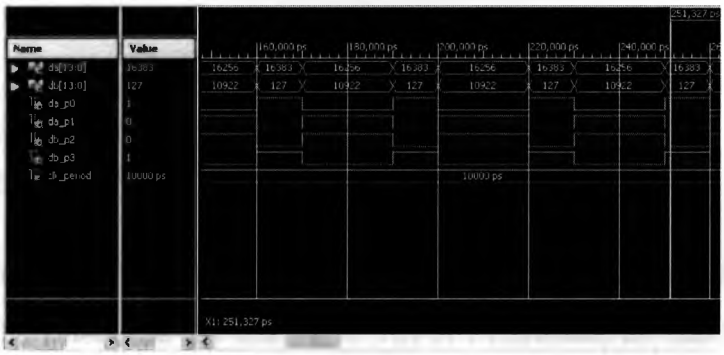
(b) Experimental set-up for interfacing DAC5687 with Virtex-5 FPGA

Figure 6.4: Experimental Setup

The ROM instantiated in the ADC's SPI controller is loaded with the register values such that DDR LVDS data outputs with bit-wise operation is selected. The data patterns fed to the ADC are such that it outputs all ones. These patterns are decoded by the interface design decribed earlier. The output data patterns were detected on the Spartan 3A FPGA using a bank of LED lights. The simulation result for the ADS62P43 controller and data patterns are shown in Figure 6.5



(a) Simulation Result for ADS62P43 SPI controller



(b) ADS62P43 Data Patterns

Figure 6.5: Simulation results for ADS62P43

The simulation was carried out by initializing all the registers to default values except for regiter 14 which is set to 0X0A(hex). This parameter will enable the ADS62P43 to give LVDS outputs. This is depicted in the simulation above(Figure 6.5(a)) where the register input of 0x0A is being sent out serially from the controller.

The MAX2830 controller was also tested and simulated. The simulation is depicted in Figure6.6

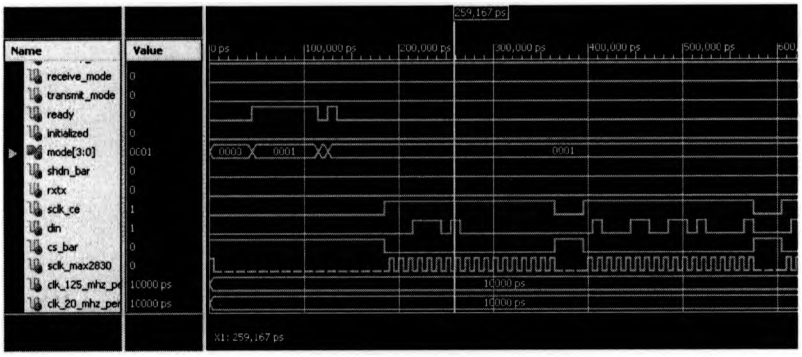


Figure 6.6: Simulation Result for MAX2830 SPI controller

The simulation shows the inputs being fed to the MAX2830 registers via the output *din*. The simulation verifies the SPI communication between the FPGA and MAX2830.

6.4 Conclusion

In this chapter, the chosen RF transceiver MAX2830 has been introduced and it's features and characteristics are discussed. An SPI controller for the MAX2830 has been designed and implemented. The experimental setup for interfacing the data converters with FPGA has been depicted. Simulation results for the ADC's, DAC's and the RF transceiver's SPI controllers and data interfaces have been presented. These simulations confirm the proper functioning of the interafce and control entities for the data converters, thereby offering the ability to reconfigure these devices on the fly.

Chapter 7

Conclusion and Future Work

Software Defined Radio concepts and architectures have been reviewed and discussed. The basics behind RF front- end receivers and transmitters are also shown. The Digital front-end model for Software radio terminals was successfully created, implemented and simulated using Quartus- Vector waveform and using MATLAB simulators. This model has shown the effect of each building block of the digital front-end on the output. The DDFS architecture is studied and it implemented using the CORDIC algorithm. An exclusive CORDIC architecture is adopted to perform the quadrature modulation. The errors in the CORDIC algorithm have been analysed and simulated. These simulations have provided a better understanding of the response of the CORDIC output and enables to improve the architecture.

Different digital filters have been briefly examined to choose the appropriate area-efficient digital filter. The theory behind multi-rate CIC filters and FIR filters have been studied to get a better understanding of their characteristics. This has enabled the choice of simple area efficient channelization model. Sample rate conversion has been carried out using of chain of CIC and FIR filters. FPGA implementation of the filter model is done and simulations are carried using Quartus vector waveform and MATLAB simulators. These simulations help in understanding the design parameters of the filters and allow the user to manipulate according to the specified application.

A class of high-speed data converters from Texas Instruments have been chosen and their digital input/output information have been illustrated. The chosen ADC was ADS62P43 and the DAC was DAC5687. Key aspects of the ADC and DAC like SNR, Gain control , clock input and filter specfications are discussed. A single channel LVDS data interface for the ADS62P43 is designed and implemented on Spartan 3A FPGA. Similarly a CMOS interface is implemented for the DAC on Virtex-5 FPGA. Data interface controllers for both the DAC and ADC have been designed, implemented and simulated using ModelSim. SPI controllers for the ADC and DAC have been realized and tested.

MAX2830 RF transceiver from Maxim has been studied and employed in this thesis. Data controller for the MAX2830 has also been designed and implemented on Virtex-5 FPGA. An ADC-FPGA interface and DAC-FPGA interface experimental setups is illustrated. Simulation results for the controllers and interface are carried out. These simulations verify the functioning of the controllers and the interface betwween data converters and the FPGA concerned.

As the future step, the MAX2830 evaluation boards will be interfaced with the ADS62P43 and DAC5687 evaluations boards to construct a uni-directional wireless set up. After the uni-directional set up is achieved, a bi-directional model will be designed, tested and verified

References

[1] A. Tribble, "The software defined radio: Fact and Fiction," *IEEE Radio and Wireless Symposium*, pp. 5–8, Jan 2008.

[2] P. Kenington, Ed., *RF and Baseband Techniques for Software Defined Radio.* Mobile Communication Series, 2005.

[3] W. Tuttlebee, "Software-defined radio: facets of a developing technology," *IEEE Personal Communications*, vol. 6, no. 2, pp. 38–44, Apr. 1999.

[4] M. Dillinger, K. Madani, and N. Alonistioti, Eds., *Software Defined Radio: Architectures, Systems and Functions.* John Wiley and Sons, Ltd, Chichester, UK., 2003.

[5] H. Arslan and A. Gorcin, "Cognitive Radio and software defined radio: signal processing perspectives," *IEEE 16th Signal Processing, Communication and Applications Conference, 2008. SIU 2008.*, pp. 1–5, Apr 2008.

[6] T. Hentschel, *Sample Rate Conversion in Software Configurable Radios.* Mobile Communication Series, 2002.

[7] W. Tuttlebee, Ed., *Software Defined Radio: Enabling Technologies.* John Wiley and Sons, Ltd, Chichester, UK., 2002.

[8] "http://focus.ti.com/analog/docs/dataconvertershome," Texas Instruments Ltd.

[9] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Jan 2009.

[10] "http://www.xilinx.com," Xilinx Inc.

[11] F. Khalid, L. Jones, M. Prydderch, Q. Morrissey, J. Lipp, and R. Stephenson, "A programmable analogue front-end ASIC for gas micro-strip detectors having a wide range of input capacitance," *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, pp. 1936 –1940, Nov 2009.

[12] C. Toal, K. McLaughlin, S. Sezer, and X. Yang, "Design and Implementation of a Field Programmable CRC Circuit Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1142 –1147, Aug 2009.

[13] Z. Razak, A. Erdogan, and T. Arslan, "ASIC Design of an Adaptive Control Unit for Reconfigurable Analog-to-Digital Converters," *2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 179 –184, Jul 2010.

[14] G. Liu and Q. Feng, "ASIC design of low-power reconfigurable FFT processor," *7th International Conference on ASIC, 2007. ASICON '07.*, pp. 44 –47, Oct 2007.

[15] P. Zipf, C. Stotzler, and M. Glesner, "A configurable pipelined state machine as a hybrid ASIC and configurable architecture," *IEEE Computer society Annual Symposium on VLSI, 2004. Proceedings.*, pp. 266 – 267, Feb 2004.

[16] H. Yoshida, T. Kato, T. Tomizawa, S. Otaka, and H. Tsurumi, "Multimode software defined radio receiver using direct conversion and low-IF principle: Implementation and evaluation," *Electronics and Communications in Japan (Part I: Communications)*, vol. 86, no. 10, p. 55 65, Oct 2003.

[17] C.-H. Hsiao, C.-J. Li, F.-K. Wang, T.-S. Horng, and K.-C. Peng, "Study of direct-conversion transmitter pulling effects in constant envelope modulation systems," *2010 IEEE MTT-S International Microwave Symposium Digest (MTT)*, pp. 1174–1177, May 2010.

[18] S.-B. Park and M. Ismail, "DC offsets in direct conversion multistandard wireless receivers: Modeling and cancellation," *Analog Integrated Circuits and Signal Processing*, vol. 49, no. 2, pp. 123–130, Oct 2006.

[19] H. Okuni, R. Ito, H. Yoshida, and T. Itakura, "A Direct Conversion Receiver with Fast-Settling DC Offset Canceller," *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007.*, pp. 1–5, Sep 2007.

[20] J. Crols and M. Steyaert, "Low-IF topologies for high-performance analog front ends of fully integrated receivers," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 269 –282, Mar 1998.

[21] J. Park, C.-H. Lee, B.-S. Kim, and J. Laskar, "Design and Analysis of Low Flicker-Noise CMOS Mixers for Direct-Conversion Receivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 12, pp. 4372 –4380, Dec 2006.

[22] S. G. Glisic, Ed., *Advanced wireless communications : 4G technologies.* Wiley Publications, Jun 2004.

[23] L. Huac and S. Hao-shan, "A digital image-rejection sub-system for low-IF receivers," *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.*, vol. 2, pp. 762 – 765, Apr 2003.

[24] M. Valkama, K. Salminen, and M. Renfors, "Digital I/Q imbalance compensation in low-IF receivers: principles and practice," *14th International Conference on Digital Signal Processing, 2002. DSP 2002.*, vol. 2, pp. 1179 – 1182, 2002.

[25] P. Eloranta, P. Seppinen, and A. Parssinen, "Direct-digital RF-modulator: a multi-function architecture for a system-independent radio transmitter," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 144 –151, Apr 2008.

[26] T. Bijlsma, P. Wolkotte, and G. Smit, "An Optimal Architecture for a DDC," *20th International Parallel and Distributed Processing Symposium, 2006. IPDPS 2006.*, pp. 8 pp.–, Apr 2006.

[27] A. Corporation, "Accerlating DUC and DDC Systems Design for WiMax," *IEEE Communications Magazine*, vol. 1, no. 421, May 2007.

[28] H. Engineering, "Theory of Digital Down Conversion."

[29] T.-H. Yu, C.-L. Yu, K.-Y. Jheng, and A.-Y. Wu, "On-line MSR-CORDIC VLSI architecture with applications to cost-efficient rotation-based adaptive filtering systems," *IEEE Workshop on Signal Processing Systems Design and Implementation, 2006. SIPS '06.*, pp. 422–427, Oct 2006.

[30] S. Kadam, D. Sasidaran, A. Awawdeh, L. Johnson, and M. Soderstrand, "Comparison of various numerically controlled oscillators," *IEEE Canadian Conference on Electrical and Computer Engineering, CCECE'08, Niagara Falls, Ontario, Canada*, vol. 3, pp. 200–202, Aug 2002.

[31] R. Romero-Troncoso and G. Espinosa-Flores-Verdad, "Algorithm for phase accumulator synthesis for applications in DDS," *Third International Workshop on Design of Mixed-Mode Integrated Circuits and Applications, 1999*, pp. 210–213, Jul 1999.

[32] J. E. Volder, "The CORDIC Trigonometric Computing Technique," *IEEE Transactions on Electronic Computers*, vol. 8, no. 3, pp. 330–334, Sep 1959.

[33] T.-Y. Sung and H.-C. Hsin, "Design and simulation of reusable IP CORDIC core for special-purpose processors," *Computers and Digital Techniques, IET*, vol. 1, no. 5, pp. 581–589, Sep 2007.

[34] Y. Hu, "The Quantization Effects of the CORDIC Algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 834–844, Apr 1992.

[35] K. Yeung and S. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 12, pp. 2444–2459, Dec 2004.

[36] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 155–162, Apr 1981.

[37] B. White and M. Elmasry, "Low-power design of decimation filters for a digital IF receiver," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 339–345, Jun 2000.

[38] U. Meyer-BaeseDornhege, Ed., *Digital Signal Processing with Field Programmable Gate Arrays*. Springer, 2007.

[39] J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar, "The use of CORDIC in software defined radios: a tutorial," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 46–50, Sep 2006.

[40] W. Han, Z. Yousi, and L. Xiaokang, "A Parallel Double-Step CORDIC Algorithm for Digital Down Converter," *Seventh Annual Communication Networks and Services Research Conference, 2009. CNSR '09.*, pp. 257–261, May 2009.

[41] J. Granado, A. Torralba, J. Chavez, and V. Baena-Lecuyer, "Design of an efficient CORDIC-based architecture for synchronization in OFDM," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 774–782, Aug 2006.

[42] M. Bekooij, J. Huisken, and K. Nowak, "Numerical Accuracy of Fast Fourier Transforms with CORDIC Arithmetic," *The Journal of VLSI Signal Processing*, vol. 25, no. 2, pp. 187–193, Jun 2000.

[43] I. Janiszewski, B. Hoppe, and H. Meuth, "Numerically controlled oscillators with hybrid function generators," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 49, no. 7, pp. 995 –1004, Jul 2002.

[44] S. Chu and C. Burrus, "Multirate filter designs using comb filters," *IEEE Transactions on Circuits and Systems*, vol. 31, no. 11, pp. 913–924, Nov 1984.

[45] "http://www.altera.com," Altera Corporation.

[46] "Interfacing Analog to Digital Converters to FPGAs," White paper, Lattice Semiconductor Corporation, 2007.

[47] M. Defossez, "Connecting Xilinx FPGAs to Texas Instruments ADS527x series ADCs," Application Note: Virtex-II, Virtex-II Pro, and Spartan-3 Families, 2006.

[48] *ADS62P43- DUAL CHANNEL, 14-BITS, 125/105/80/65 MSPS ADC WITH DDR LVDS/CMOS OUTPUTS*, Texas Instruments, May 2009.

[49] M. Defossez, "An Interface for Texas Instruments Analog-to-Digital Converters with Serial LVDS Outputs," Application Note: Virtex-4 and Virtex-5 FPGAs, 2008.

[50] F. Zhang, Z. Yang, W. Feng, H. Cui, L. Huang, and W. Hu, "A High Speed CMOS Transmitter and Rail-to-Rail Receiver," *4th IEEE International Symposium on Electronic Design, Test and Applications, 2008. DELTA 2008.*, pp. 67 –70, Jan 2008.

[51] *DAC5687- 16-BIT, 500 MSPS 28 INTERPOLATING DUAL-CHANNEL DIGITAL-TO-ANALOG CONVERTER (DAC)*, Texas Instruments, Sep 2006.

[52] *UG331 - Spartan-3 Generation FPGA User Guide; Extended Spartan-3A, Spartan-3E, and Spartan-3 FPGA Families*, Xilinx Inc., Jan 2009.

[53] *MAX2830- 2.4GHz to 2.5GHz 802.11 g/b RF Transceiver, PA and Rx/Tx/Antenna Diversity Switch*, Maxim Instruments, Jul 2009.