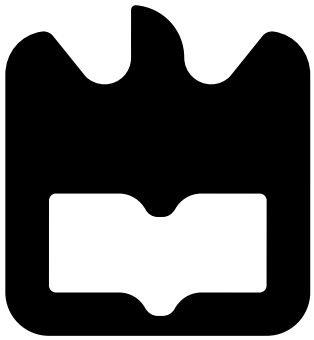
Universidade de Aveiro
Departamento de
Electrónica, Telecomunicações e Informática
2012

Luís Carlos
de Oliveira Matos

Plataforma para Projeto de Sistemas de Rádio Definidos por Software

Design Platform for Software Defined Radio Systems



Luís Carlos de Oliveira Matos

Plataforma para Projeto de Sistemas de Rádio Definidos por Software

Design Platform for Software Defined Radio Systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Arnaldo Silva Rodrigues de Oliveira, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor Joao Nuno Pimentel da Silva Matos
Professor Associado, Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Professor Doutor Pedro Renato Tavares de Pinho
Professor Adjunto, Area Departamental de Engenharia de Eletronica e Telecomunicacoes e de Computadores do Instituto Superior de Engenharia de Lisboa (Arguente Principal)

Professor Doutor Arnaldo Silva Rodrigues oliveira
Professor Auxiliar, Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

The conclusion of this dissertation would not have been possible without the help of many people.

At the professional area, first of all, I would like to thank my coordinator for his help, guidance and wisdom passed along the project's duration. To M.Sc. Nelson Cardoso for his help regarding the PCB board design as well as some hardware choices. From the Telecommunications Institute staff, i would like to thank Mr. Paulo Gonçalves for the precious help assembling some hardware parts on the PCB board.

At the personal level, I would like to tank all my friends made since I started my studies at Aveiro's University, ones closer than others, but all of them have some importance in its own way. From my hometown, to my friends since childhood that every weekend were there for me, sharing my joys and sorrows.

To my family, that always was there for me: for listened, encouraged, advised and above all loved me. Also to a special friend that in the past months has been a tremendous help and support.

Muito obrigado a todos!!

Resumo

Este trabalho tem como objetivos o projeto e a realização de uma plataforma para desenvolvimento de sistemas baseados em tecnologia Software Defined Radio (SDR). Num sistema SDR todas as tarefas de um rádio (ou pelo menos banda base e eventualmente Frequencia Intermédia), anteriormente efetuadas por hardware específico num contexto analógico, são efetuadas no domínio digital por software ou hardware reconfigurável. Esta característica confere a este tipo de rádio uma maior simplicidade, em termos de hardware bem como maior flexibilidade, pois o mesmo dispositivo pode executar diferentes funções apenas alterando o seu firmware/software. Existem diferentes abordagens relativas ao uso desta tecnologia, quer ao nível da arquitetura usada (varia consoante a frequência onde ocorre a digitalização do sinal), quer relativas à topologia de utilização (controlada por hardware reconfigurável, rotinas de sotfware ou ambos). A motivação deste trabalho resulta na necessidade de conceção de uma plataforma para fins académicos baseada num hardware reprogramável, Field Programmable Gate Array (FPGA), de baixo custo, flexível, com interfaces de comunicação digitais e analógicas e que faculte a possibilidade de ser usada em diferentes topologias de utilização. Efetuada a especificação e o estudo necessário ao projeto bem como a escolha apropriada de componentes, conseguiu-se uma plataforma baseada num módulo FPGA contendo um dispositivo Xilinx, da família Spartan-6, bem como outro hardware relevante. A comunicação com outros dispositivos é assegurada por interfaces USB e gigabit Ethernet. A plataforma concebida está também equipada com interfaces analógicas (conversores AD/DA) bem como algumas interfaces de interação com o utilizador consistindo em switches e LEDs. Em suma foi projetada e desenhada uma plataforma aberta e flexível, que pode ser usada com todas as ferramentas de desenvolvimento, programação e depuração, com fácil acesso a todos os sinais relevantes potenciando a sua utilização para efeitos de ensino e investigação em SDR.

Abstract

The main objective of this dissertation relies on projecting and designing a platform suitable for Software Defined Radio (SDR) system development. On an SDR system all, or at least base band and maybe Intermediate Frequency (IF) radio functions, before handled by analog specific hardware, are now performed on the digital domain by software or an reconfigurable hardware device. This feature provides to this type of radios a major simplicity regarding hardware as well as another flexibility level since, through a firmware/software upgrade, the same equipment can perform different functions. There are some approaches related to the used of this technology, either regarding architecture implementation (they differ in which frequency the digitalization occurs) or utilization topologies (an SDR device can be controlled by a reconfigurable hardware, software routines or both). This project's motivation results from the need of designing a flexible, low-cost platform, to be used on academic purposes, in which the central component would be a reconfigurable hardware, a Field Programmable Gate Array (FPGA). It must provide both analog and digital interfaces so that can be used on various utilization scenarios. Accomplished all the necessary study, design and hardware selection the result is a platform based on an FPGA module, containing an *Xilinx* device from the *Spartan-6* family as well as other relevant hardware. The interaction with other devices is ensured by both gigabit Ethernet and 2.0 Universal Serial Bus (USB) connections. The platform also features analog interfaces (AD/DA converters) as well as some digital end-user interfaces performed by switches and Light Emiter Diodes (LED)s. Concluding, it was built an open and flexible platform in which can be use with all provided development, programming and debugging tools and all the relevant signals have easy access enhancing its use for teaching and researching on SDR technology.

Contents

Contents i

List of Figures iii

List of Tables v

Acronyms vii

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 3
- 1.4 Dissertation Outline . . . 3

2 Software Defined Radio - Definition and Applications 5
- 2.1 Introduction . . . 5
- 2.2 Definition . . . 6
- 2.3 Architectures . . . 7
  - 2.3.1 Baseband Digitalization . . . 7
  - 2.3.2 IF Digitalization . . . 8
  - 2.3.3 RF Digitalization . . . 9
- 2.4 Cognitive Radio . . . 9
- 2.5 Implementation Technologies . . . 10

3 SDR Platforms 13
- 3.1 Introduction . . . 13
- 3.2 USRP - Universal Software Radio Peripheral . . . 13
  - 3.2.1 Bus series . . . 14
  - 3.2.2 Embedded Series . . . 17
  - 3.2.3 Network series . . . 18
- 3.3 Sundance . . . 19
- 3.4 Rational to build a new SDR Platform . . . 23

4 Project Specification and Design 25
- 4.1 Features specification . . . 25
  - 4.1.1 Reconfigurable Hardware . . . 25
  - 4.1.2 Data Conversion . . . 25

4.1.3 Communications Interfaces . . . 26
4.1.4 Programming/Debugging Interfaces . . . 26
4.1.5 User Interaction . . . 26
4.1.6 Power Supply . . . 26
4.2 Block Diagram . . . 26
4.3 Design and Hardware Selection . . . 26
4.3.1 FPGA Module . . . 26
4.3.2 Data Conversion . . . 28
4.3.3 Communication Interfaces . . . 28
4.3.4 Programming/Debugging Interfaces . . . 29
4.3.5 User Interfaces . . . 30
4.3.6 Power Supply . . . 30
4.3.7 Bill of Materials . . . 30
4.3.8 Final Platform . . . 30

5 Platform Tests and Results **35**
5.1 Digital IO Test . . . 35
5.2 DAC Test . . . 36
5.2.1 Limit Range Test . . . 36
5.2.2 Test with an Output Sine wave . . . 37
5.3 ADC test . . . 38
5.3.1 Range Test . . . 38
5.3.2 ADC to DAC Loopback . . . 38
5.3.3 Other Possible Tests . . . 41

6 Conclusion and Future Work **45**
6.1 Conclusion . . . 45
6.2 Future Work . . . 46

Bibliography **47**

A Schematic Diagrams **49**

B PBC Layout **60**

List of Figures

2.1 SDR - Ideal Architecture . . . 6
2.2 Super Heterodyne receiver, adapted from [7] . . . 6
2.3 Baseband Digitalization Receiver . . . 8
2.4 IF Digitalization Receiver . . . 8
2.5 RF Digitalization Receiver . . . 9
2.6 Radio Technology Evolution, from [9] . . . 10

3.1 USRP1 Block Diagram, from [13] . . . 15
3.2 USRP B100 Block Diagram, from [14] . . . 16
3.3 USRP E100/E110 Block Diagram, from [15] . . . 17
3.4 USRP N200/N210 Block Diagram, from [16] . . . 18
3.5 SDR3.0 Block Diagram, from [17] . . . 20
3.6 SDR4.0 Block Diagram, from [18] . . . 21
3.7 RadioGiga 2.0 Block Diagram, from [19] . . . 22

4.1 Platform Block Diagram . . . 27
4.2 *Trenz TE0630*, Top and Bottom View and Block Diagram, from [20] . . . 28
4.3 *Trenz GigaBee*, Top and Bottom View and Block Diagram, from [21] . . . 29
4.4 *Enclustra Mars-MX2*, Top and Bottom View and Block Diagram, from [22] . . . 29
4.5 FT2232H Mini Module, from [23] . . . 30
4.6 Final Block Diagram . . . 32
4.7 Final Platform . . . 33

5.1 Setup Test . . . 36
5.2 Digital IO Test - Diagram . . . 37
5.3 Digital IO Test - Platform View . . . 37
5.4 DAC Tests - Diagram . . . 38
5.5 DAC Time Diagram, from [24] . . . 39
5.6 DAC Limit Range Test 25 MHz - a) Channel 1: Clock, channel 2: Dac Output b) Channel 1: Clock, channel 2: Dac "write" signal . . . 39
5.7 DAC Limit Range Test 50 MHz - a) Channel 1: Clock, channel 2: Dac Output b) Channel 1: Clock, channel 2: Dac "write" signal . . . 40
5.8 Test With an Output Sin wave - a) DAC output with a 25 MHz clock b) DAC output with a 50 MHz clock. . . . 40
5.9 ADC to DAC Loopback - Diagram . . . 41
5.10 ADC Range test - Platform View . . . 42

5.11 ADC to DAC loopback - Channel 1: Wave from the signal generator, channel 2: DAC Output . . . 42
5.12 ADC to DAC loopback - Channel 1: Wave from the signal generator, channel 2: DAC Output . . . 43

A.1 ADC Schematic . . . 49
A.2 ADC Input Opamp Schematic . . . 50
A.3 DAC Schematic . . . 51
A.4 USB Module Schematic . . . 52
A.5 Mars MX2 Connector Schematic . . . 53
A.6 Power Network Schematic . . . 54
A.7 USB and JTAG Programming Schematic . . . 55
A.8 Digital User Interfaces Schematic . . . 56
A.9 Command and Status Interfaces Schematic . . . 57
A.10 Ethernet Schematic . . . 58
A.11 High Speed Connector Schematic . . . 59

B.1 Top Layer . . . 61
B.2 Bottom Layer . . . 62
B.3 Ground Layer . . . 63
B.4 Power Layer . . . 64
B.5 Drill Chart . . . 65

List of Tables

3.1 URSP 1 Overview 14
3.2 USRP B100 Overview 15
3.3 USRP E100/E110 Overview 17
3.4 USRP N200/N210 Overview 18
3.5 SDR 3.0 Overview 19
3.6 SDR 4.0 Overview 20
3.7 Radio Giga 2.0 Overview 21

4.1 Bill of Materials 31

Acronyms

AC Alternating Current

ADC Analog to Digital Converter

CR Cognitive Radio

DAC Digital to Analog Converter

DDC Digital Down Converter

DSP Digital Signal Processor

DUC Digital UP Converter

FFT Fast Fourier transform

FIFO First In First Out

FPGA Field Programmable Gate Array

GPP General Purpose Processor

GPS Global Positioning System

HDMI High-Definition Multimedia Interface

IEEE Institute of Electrical and Electronic Engineers

I2C Inter-Integrated Circuit

IF Intermediate Frequency

JTAG Joint Test Action Group

LED Light-Emitting Diode

LUT Lookup Table

MAC Multiply And Accumulate

MIMO Multiple-Input and Multiple-Output

MPSSE Multi-Protocol Synchronous Serial Engine

MSPS Mega Samples Per Second

OFDM Orthogonal Frequency-Division Multiplexing

PCB Print Circuit Board

RF Radio Frequency

SDR Software Defined Radio

SPI Serial Peripheral Interface

SR Software Radio

UART Universal Asynchronous Receiver/Transmitter

USB Universal Serial Bus

USRP Universal Software Radio Peripheral

VCP Virtual COM Port

VHDL Very high speed Integrated circuits Hardware Description Language

Chapter 1

Introduction

1.1 Overview

Software Defined Radio (SDR), as the name suggests, is a radio system whose all or some parts are digitally controlled by software or a reconfigurable hardware device. Despite of being used since 1970's for military purposes, this concept was first published in 1991 by *Joseph Mitola*, who defined it as "a radio whose channel modulation wave-forms are defined in software" [1]. Also known as Software Radio (SR), the wireless Innovation Forum™, former SDR Forum, working along with the Institute of Electrical and Electronic Engineers (IEEE), set a definition for Software-defined Radio as a "Radio in which some or all of the physical layers functions are software defined" [2].

This technology defends that an Analog to Digital Converter (ADC) can receive a signal from a wideband antenna, and all the aspects refer to signal identification, down-conversion, demodulation, decryption, among other aspects would be handled by software/reconfigurable hardware. Due to the state-of-the-art of both the converters and antennas, this architecture is still not feasible. Some hardware is still needed to handle the signal properly before the digital conversion. The previous statements only refer to the receiver process, the transmitter side follows the same idea but in reverse order.

SDR has been a tremendous booster on telecommunication system's development. Systems with this technology are more simple, economic, versatile and robust. They require less hardware parts which mean that are much simpler to analyze and build. The same device can perform a completely different task, all it needs is a new configuration. Systems became more efficient in terms of performance and energy consumption. Also the migration to a new standard, frequency or modulation is much easier than before. [3]

Due to its reconfigurability, increasing performance and logic capacity, Field Programmable Gate Array (FPGA) usage on complex systems, such as in the telecommunication domain, has grown considerably in the last decade. A single state-of-the-art FPGA can integrate multi-core processors handling operating system and applications software, system's hardware modules along with the entire logic network needed to connect them reducing the number of the system's physical components. On the other hand, FPGA's improvements in the past years, is one of the main reasons for the expansion of the software defined radio systems due the parallel processing capability as well as dedicated signal processing blocks regarding filters, Fast Fourier transform (FFT) among others.

The research on this field is far from finished and new technologies, derived from SDR,

like adaptive radio or cognitive radio are also emerging. A system based on adaptive radio technology has the ability of monitoring its performance in order to modify its parameters improving quality service. Cognitive radio can be defined as an adaptive radio that can also be aware of their internal state and environment, such as location and utilization of Radio Frequency (RF) frequency spectrum in that location. They can make decisions about their radio operation behaviour by mapping it against predefined objectives [2].

1.2 Motivation

Regarding the development and research of SDR systems a large variety of items can be found also with plenty varied features. The variety of platforms differ in the reconfigurable hardware device, FPGA, Digital Signal Processor (DSP), General Purpose Processor (GPP)s or yet an interaction of more than one, the performance regarding data conversion, analog to digital and vice-versa, as well as communication protocols and standard interfaces. Other external peripherals featuring the platform are also important, like external memory and standard I/O interfaces. All of these platforms are available for different prices and performance levels. The following item list intends to demonstrate how performance and price are related.

- **From 600$ to 700$**, items can be found equipped with a decent FPGA which is pre-programmed with a Digital UP Converter (DUC) and Digital Down Converter (DDC). Also featuring both an ADC and Digital to Analog Converter (DAC) with a reasonable performance. The access to the FPGA is not easily done, in a way that these platform must be used on a topology where, besides the up and down conversion, all the remaining digital processing should be done in software thought a Universal Serial Bus (USB) connection.

- **From 1300$ to 1700$**, much similar with the previous item. Apart from an improved FPGA , specially designed for signal processing purposes as well as improved data converters and addition of communication interfaces, like High-Definition Multimedia Interface (HDMI) or Ethernet connections. Again, not suitable for standalone applications.

- **From 13000€to 18000€**, high-performance level platforms combining more than one high yield FPGAs and DSPs. Data converters with high sampling rate and resolution bits, a large variety of communication standards, high-speed data transfer protocols and also both shared and dedicated memory blocks. these platforms are capable of being used in a standalone topology or in communication with other device, the signal processing could be preformed by reconfigurable hardware, by software or both giving the project designer a greater variety of choice, fact that benefits the overall system performance.

This project's main motivation goal relies on the project and validation of a low cost and flexible platform, suitable for SDR systems development. Should feature an FPGA as a reconfigurable hardware device as well as competent performance ADCs and DACs. In order to be used in different SDR approaches, must also include standard communication interfaces. Summing, a platform with attractive characteristics for academic, either pedagogic or researching purposes.

1.3 Objectives

As mentioned before, the main goal of this dissertation concerns on the specification, production and validation of a design platform for SDR systems suitable for both pedagogic and researching purposes. The objective's timeline is presented next.

- Find an approach regarding a reconfigurable hardware suitable for different SDR topologies;
- Platform specification. Decide which features or characteristics the platform should or shouldn't present based on a research regarding other academic development platforms;
- Select the remaining components required, ADCs, DACs, voltage regulators along with the other support hardware. Perform a market research regarding component's price and performance.
- Development and validation of the project's electric schematic ;
- Construction and test of the prototype.

1.4 Dissertation Outline

The presented document is organized into six chapters.

- **Introduction** brief project approach where are explained the technology where its inserted as well as the main objectives and motivation.
- **Software Defined Radio - Definition and Applications** explains the technology concept problems and future challenges.
- **FPGA Platforms** present a brief market research regarding other SDR platforms.
- **Project Specification and Design** where are presented the desirable features for the final project.
- **Platform Tests and Results** contain the information related to the test preformed, explanation and how was done, as well as as brief reflection regarding the consequent results.
- **Conclusion and Future Work** final considerations regarding the overall work as well as describes what still can be done.
- **Schematic Diagrams** Appendix containing all the schematic diagram images.
- **PCB Layout** Appendix containing all the platform's layouts.

Chapter 2

Software Defined Radio - Definition and Applications

2.1 Introduction

Since the beginning of times men had the need to communicate. Through sound, fire or smoke, ancient techniques continued to be improved over the ages. With the advances on the study of electromagnetic theory a wireless communications system wasn't a utopia anymore, and in 1896, Guglielmo Marconi invented the wireless telegraph, it was the event that boosted the development on wireless technology. The "radio" concept began to gain form and in 1914 occurred the first voice over the radio transmission and by 1940, radio transmission were used already by the majority of the police stations. Six years later with the birth of mobile telephony technology start to evolve at a faster rhythm. Just as telephony evolved, also did the internet. Its progression was made side by side along with the computers. New communication protocols, not only between computers but also with other devices, were created and the hardware should handle them properly. At this point it wasn't clever anymore to create communications devices with just one function.

Alongside technological developments, the improvement in digital processing techniques was also important to communication systems innovation. Digital systems are already known for their performance efficiency and the appearance of the digital processor was the key for the development of reconfigurable communication systems [4].

Merging all the aspects together we've come to the technology age. In the last two decades we've assisted the "boom" of telecommunications improvement. Commercial wireless network standards have been in constant evolution. Each new generation or technology has some differences from the last one, normally is faster, better and, in most cases, cheaper. In some cases just a minor software adjustment is enough, but others come when the hardware must be changed. The idea of a one job performing device is getting less and less followers. So engineers began to anticipate these sudden changes and start to realize that the SDR's reconfigurable skills would handle those changes much easily, they could be made just by a software adjustment instead of the design, construction and application of a new hardware platform. Also for network operators, upgrading for new generation technologies would be less expensive since wouldn't involve so many high cost hardware changes.

With all of this SDR has generated tremendous interest in the wireless communications environment for the wide-ranging economic and deployment benefits it offers. [3]

2.2 Definition

Despite not being called Software Defined Radio yet, its general concept started to be developed in 1970's in a military project called Speakeasy [5]. The project was developed by the U.S. military, its main objective was to gather several radios into one using digital techniques. They managed to assure communications between 10 different radios operating between 2 MHz and 2 GHz [4].

Joseph Mitola was the first person to come up with a definition of Software Defined Radio. He was the first one to have published an article about SDR. According to him a SDR system is:

> *" ... a radio whose channel modulation waveforms are defined in software. That is, waveforms are generated as sampled digital signals, converted from digital to analog via wideband DAC and then possibly upconverted from Intermediate Frequency (IF) to RF. The receiver, similarly, employs a wideband ADC that captures all of the channels of the software radio node. The receiver then extracts, downconverts and demodulates the channel waveform using software on a general purpose processor. [6] "*

It means that data conversion, digital to analog in the transmitter and analog to digital in the receiver, should occur immediately after the signal capture, i.e., as closer as possible to the antenna and with that having, as much as possible, a wider digital bandwidth.

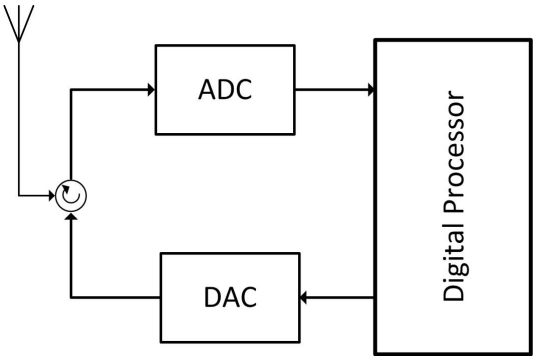


Figure 2.1: SDR - Ideal Architecture

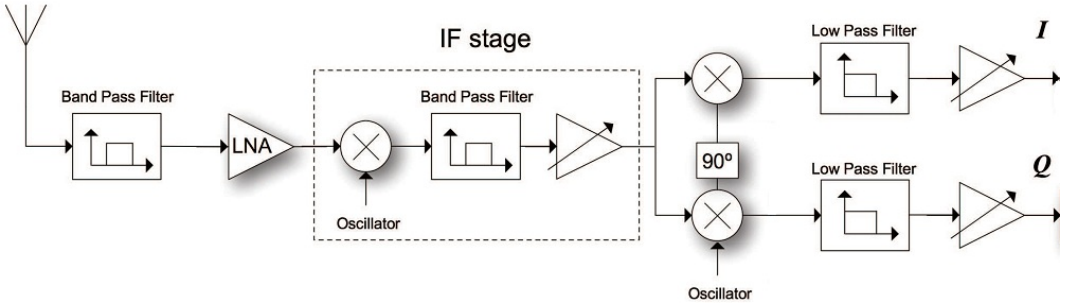


Figure 2.2: Super Heterodyne receiver, adapted from [7]

Analysing the block diagram of the ideal Software Defined Radio system, represented on the figure 2.1, and comparing to the traditional one (the figure 2.2 exemplifies one possible receiver architecture), is visible that all the support hardware is missing. In a SDR system all the work performed by filters, mixers, modulators and other related hardware is handled by a reconfigurable hardware device. That points out one of the most important feature of a SDR system, just by making a software adjustment, the system can perform a completely different task.

The idea of an SDR system comes with different application technologies. When is mentioned that a software adjustment is enough to completely chance a device behaviour, depending on the application used, that can means an actually software update, a recalibration of the reconfigurable hardware or both.

When an SDR system is used on a standalone application, that means that all the signal processing functions are performed in reconfigurable hardware. In other topologies, for example, an SDR system in communication with a computer, some function could be preformed in hardware as well as on software routines on the computer. Regarding the topology's should be considered both project's specifications and the amount of available resources.

In past years has been known the fast emerge of new communication standards. There are a large variety of different protocols and it is on the best interest that a single device can handle a good part of them. SDR technology enables an efficient and not so expensive solution, allowing multi-tasked communication devices which can be improved using a software/firmware upgrade.

Since the definition of SDR systems wasn't that consistent, a group called SDR Forum emerged. In association with the IEEE, they worked on a definition for SDR so that this technology becomes more solid, clear to understand its features and benefits. The SDR Forum is now called Wireless Innovation Forum.

2.3 Architectures

As mentioned before, the ideal SDR architecture is still not feasible. This fact is due to some utopian features required. Infinite bandwidth antenna and sampling frequency data converters.

The following sections describe the considered architectures in which the main difference between them is the frequency where the digitalization occurs. Regarding cleaner reading, only the receiver architecture is mentioned, it's assumed a dual architecture for the transmitter.

2.3.1 Baseband Digitalization

Analysing the figure 2.3, is visible that the support hardware performs two downconversions, from RF to an IF signal, and from that to baseband. Once this process is finished the ADC samples the signal and the remaining radio function are handle by software/reprogrammable hardware.

The advantage of this architecture is that it uses low-price components which are abundant on the market. In contrast, those components have a very limited bandwidth and it is needed a considerably number of hardware increasing the architecture complexity. Due to it's reduced digitalized bandwidth, is difficult for this devices to perform other function unless it requires the same frequency range.

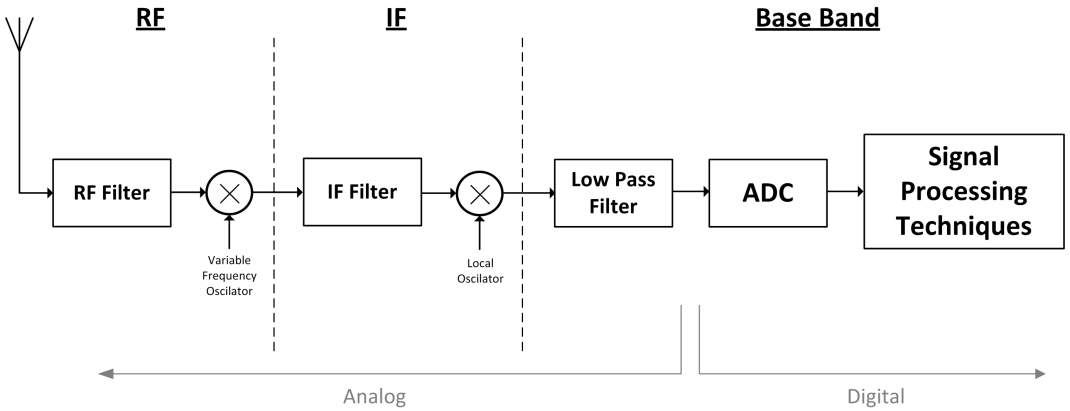


Figure 2.3: Baseband Digitalization Receiver

It goes against two of the SDR principles, wideband purpose and low hardware complexity.

2.3.2 IF Digitalization

The figure 2.4 represents the architecture of a receiver based on IF topology. In this case the digitalization occurs closer to the antenna. The support hardware performs only one downconversion from RF to IF level.

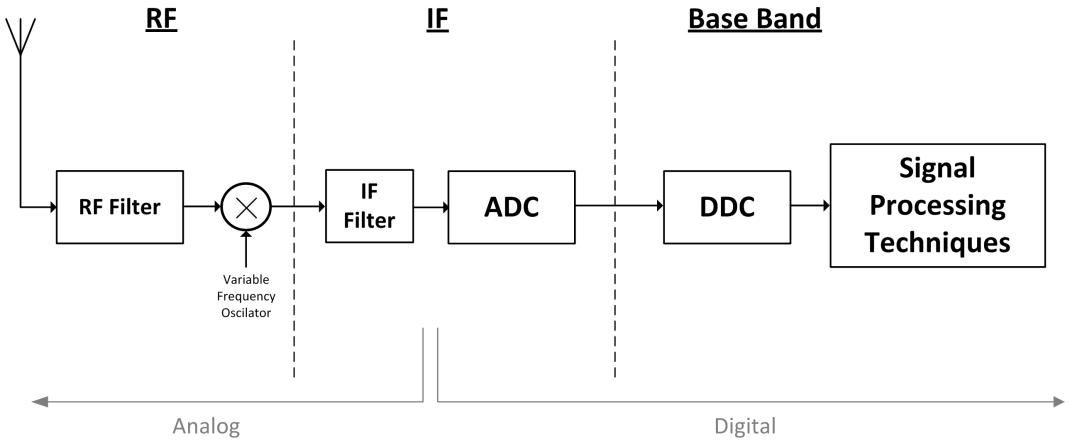


Figure 2.4: IF Digitalization Receiver

Following the chain, the signal is sampled and the remaining functions are handled properly. Comparatively to the previous section, this architecture requires hardware components from a higher price level. This is only possible due to the advances on semiconductor industry, which nowadays brings to the market data converters with attractive cost/performance ratio.

Since the data conversion takes place on the IF stage, the frequency window available for data processing is much wider. With this improvement in digital bandwidth covered the number of communication protocols that can be covered is much more significant.

Despite still not be the ideal model, it still needs some support hardware, is in fact the most common SDR architecture currently in use. The advances in hardware simplicity and

covered bandwidth bring it closer to *Mitola's* system.

2.3.3 RF Digitalization

The RF digitalization architecture, represented in the figure 2.5, is simple and only the ideal architecture of an SDR system. As mentioned before, in this architecture the signal is digitalized right after the wideband antenna with all the other processes preformed on the digital domain. Although highly desirable, this architecture is still not feasible, or at least, not at an affordable price.

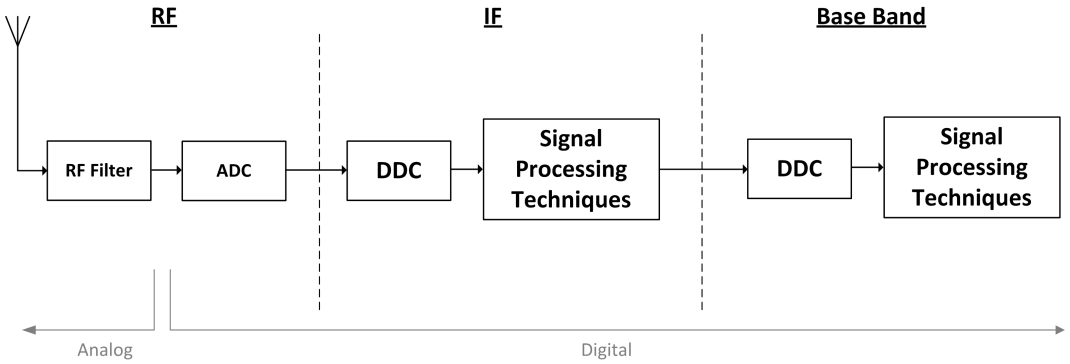


Figure 2.5: RF Digitalization Receiver

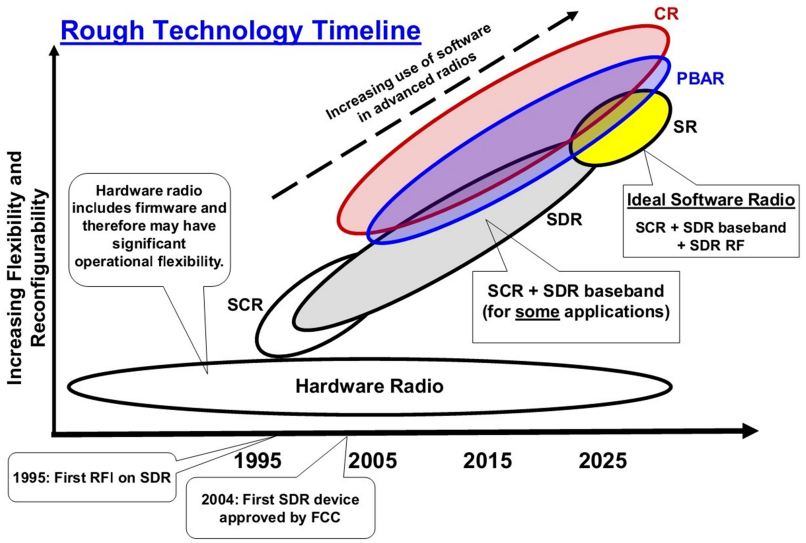
With this configuration, the system should be able to "listen" the entire electromagnetic spectrum, i.e., to be equipped with an ADC with infinite sampling speed and considerable number of resolution bits. Is still a utopia, but efforts have been made to find more configurations that bring us closer to it.

2.4 Cognitive Radio

With the idea of Software Defined Radios, a new concept came up, the Cognitive Radio (CR). Also brought up by *Joseph Mitola* [8], a cognitive radio is an upgrade to SDR. Is an Intelligent SDR that can sense its environment and adapt to it, it means that should be capable of be aware of the entire spectrum and reconfigure itself regarding better performance. For example, if a communication channel it too busy, a CR should be able to detect another frequency, change its modulation parameters and ensure communication thought other channel.

Similar to the ideal architecture of an SDR, it should have a wide band front-end in order to be capable of sensing all the spectrum and communicate with large bandwidth waveforms. Once this technology becomes more feasible, some issues probably will come up, mostly ethical ones. Spectrum allocation by the service provider companies, prices to be charged to the end-user, are some example problems that we might have to deal with in the future.

The figure 2.6 shows the evolution of radio technologies over the years. Is visible that digital radios are the future, and nowadays some of the SDR systems are already implemented but more research should be done in order to reach the perfect cognitive radio, the most flexible one.



Note: SDR currently is practical for some applications such as commercial wireless basestations , but not some wireless hand-sets. Cost, power, size and weight are critical design requirements that must be considered when considering the use of advanced radio technologies.

CR – Cognitive Radio
RFI – Request for Information
SDR – Software Defined Radio
PBAR – Policy-Base Adaptive Radio
SCR – Software Controlled Radio
SR – Software Radio

Figure 2.6: Radio Technology Evolution, from [9]

2.5 Implementation Technologies

Nowadays digital signal processing is almost everywhere, music, speech, video, navigation, telecommunications [4]. Especially modern wireless communications, they rely on an effective implementation of digital processing function as filtering, transforms or modulations. [10] Depending on the desired application, different devices can handle these functions. DSP, GPP and FPGA are three possible choices.

Comparing GPPs to DSP, can GPPs replace DSPs? Depending on the signal processing performance desired/needed for the application. If it requires high signal processing performance a DSP should be used, once it is especially designed to improve this kind of performance. On the other hand, in more modest performance requirements, a GPP can be used.

Both DSP and FPGA provide good performance handling complex signal processing functions. During the past decade, these two devices have increased their performance, in particular regarding digital signal processing aspects, but DSP processors performance isn't still good enough for a system requiring high performance, like SDR. One of the aspects that most contributes for this performance deficit is their limited parallelism problems, most of the DSP instructions are executed sequentially [11].

On the other side, FPGAs have also increased their overall performance in past years, and also started to come equipped with, for example, Multiply And Accumulate (MAC) units, in order to better implement digital signal processing functions. Once they have dedicated blocks dedicated to these functions, there's no need to use Lookup Table (LUT)s and therefore the configuration latency is reduced. One other advantage relies on the parallelism offered by the FPGA. It is capable of preforming many signal processing functions at the same time. Using a Orthogonal Frequency-Division Multiplexing (OFDM) communication system as an

example, it requires many FFT operations to be computed in each of many communication channels at once. Unless using a large array of DSPs, A single DSP is unable to handle this systems. Besides being an expensive manner to address this issue, the synchronization process as well as sharing memory and other resources, is not a trivial task. A single state of the art FPGA can handle more than one OFDM system. [12]

Each device serves its purposes. Depending on the system requirements, GPP, DSP or FPGA can be chosen. The GPP offer a better cost/efficiency regarding less complex systems, with no complex digital processing functions. From another point of view, DSP are used for system demanding complex digital signal processing functions without many channels involved or high data rates. FPGA are the most suitable for more complex systems demanding parallel processing as well as high performance signal processing functions.

Chapter 3

SDR Platforms

3.1 Introduction

As mentioned before, various FPGA platforms can be found on the market. The difficult task is to find, among them, one with a good cost/performance ratio.

Regarding this type of devices, their performance is assessed considering its characteristics set together with its operation mode regarding SDR purposes.

The characteristics set is encompassed by the type of reconfigurable hardware device, DSP, GPP or FPGA, the type of communication allowed, digital, analog or both and also the data converters performance. Other peripheral hardware is also important like memory, clock generators and I/O interfaces.

The operation mode is related to how the platform interacts with its surroundings, working alone or communicating with other device. As previously stated, on a standalone operation mode, all the digital signal processing is performed within the platform. Concerning non standalone topologies, it can be fully dependent or hybrid. The fully dependent refers to a device where the majority of the processing part has to be implemented in software routines, the hybrid ones are endowed with fully accessible reconfigurable hardware and also communication interfaces which allows them use software routines to process information.

The following sections present an overall overview of some SDR suitable platforms.

3.2 USRP - Universal Software Radio Peripheral

From Ettus Research^{TM}, company specialized on SDR systems development, Universal Software Radio Peripheral (USRP) devices provide worldwide users different solutions to use in the most varied application areas. It's designed for RF applications, with a large frequency operating range, providing Global Positioning System (GPS) disciplined synchronization, Multiple-Input and Multiple-Output (MIMO) configurations and embedded systems.

Regarding development tools, it is integrated with GNU Radio, Labview and Simulink. Being Labview and Simuling general purpose engineering development systems, Gnu Radio is a free and open-source software development toolkit providing pre-compiled signal processing blocks to implement software radios. Also allows to end-users to program their own signal processing blocks using *Python* and *C++* programming languages.

The USRP family combine three different categories, the Bus, Embedded and Network series, each one of them with characteristics suitable for different purposes/applications.

The following sections each device along with the features within.

3.2.1 Bus series

The Bus series is constituted by the USRP1 and the USRP B100. The USRP 1, is the original Universal Software Radio Peripheral hardware. Designed with a daughterboard topology, it allows the USRP to operate from DC to 6 GHz, using different daughterboards. It can operate with two complete daughterboards at the same time.

The FPGA it's pre-programmed with both a downconverter and updonverter and it's access for reprogramming purposes is not easy guaranteed, at least for beginner user. This features compels this devices to be uses on a non standalone topology. It must be used communicating, via USB connection, with software routines, GNU Radio, Labview or Simulink, which perform all the signal processing functions besides up and downconversions.

The following figure 3.1 and table 3.1 summarises the USRP 1 key features.

Price	**700 $**
FPGA	Altera Cyclone
Memory	
Ethernet	
USB	USB 2.0
Communications	Connectivity for two complete Tx/Rx chains; up to 16 MHz USB streaming; SMA Connector IN/OUT
User IOs	
Programming	Use With GNU Radio, LabView and Simulink
A/D Conversion	2 x Dual 64 Mega Samples Per Second (MSPS) / 12 bits
D/A Conversion	2 x Dual 128 MSPS / 14 bits

Table 3.1: URSP 1 Overview

The USRP B100, is the new member of the USRP family. Following the line of the USRP 1, it completes the BUS series family offering the flexibility of the USRP product line.

Contrary to the USRP 1, it is equipped with a different FPGA family, a *Spartan 3A - DSP*, from *Xilinx*. An FPGA specially designed for digital signal processing purposes.

The FPGA is also pre-programmed with and up and downconversion stage,leaving the remaining digital processing to software routines. Thus to be used on a non standalone topology.

The overall features of this platform are described on the figure 3.2 and table 3.2 next presented.

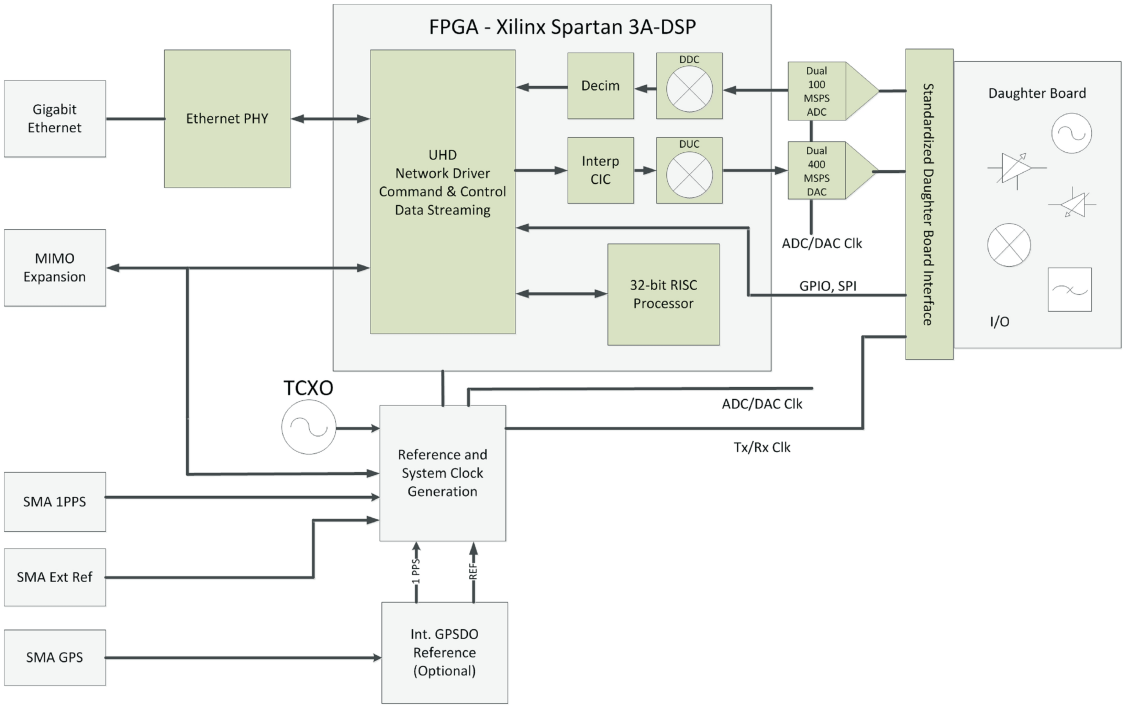


Figure 3.1: USRP1 Block Diagram, from [13]

Price	**650 $**
FPGA / DSP	Xilinx Spartan 3A
Memory	
Ethernet	
USB	USB 2.0
Communications	USB 2.0 SMA Connector IN/OUT
User IOs	
Programming	
A/D Conversion	Dual 64 MSPS / 12 bits ADC
D/A Conversion	Dual 128 MSPS / 14 bits DAC

Table 3.2: USRP B100 Overview

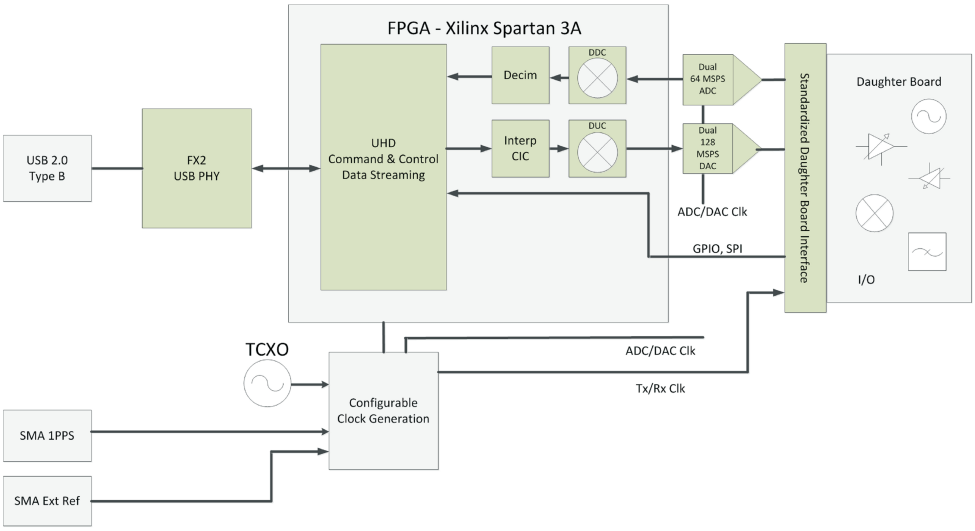


Figure 3.2: USRP B100 Block Diagram, from [14]

3.2.2 Embedded Series

The USRP E100 and E110 form the USRP embedded series. The difference between them relies on a larger FPGA equipped on the E110 model. They differ from the ones mentioned on the previous section on the topology level as well as peripheral hardware available.

The table 3.3 resumes its main characteristics.

Price	**1300 $ / 1500 $**
FPGA / DSP	Xilinx Spartan 3A - DSP
Memory	512 MB RAM / 4 GB SD Card
Ethernet	10/100 Ethernet
USB	USB Console, OTE and Host
Communications	Strereo Line In/Out HDMI Out USB Ethernet SMA Connector IN/OUT
User IOs	
Programming	
A/D Conversion	Dual 64 MSPS / 12 bits ADC
D/A Conversion	Dual 128 MSPS / 14 bits DAC

Table 3.3: USRP E100/E110 Overview

This series offer more communication interfaces and external memory available. With an embedded processor, a TI OMAP 3, gives it another level of flexibility regarding operation topology. Seeking better understanding of its features, the USRP E100/E110 block diagram is represented on the figure 3.3

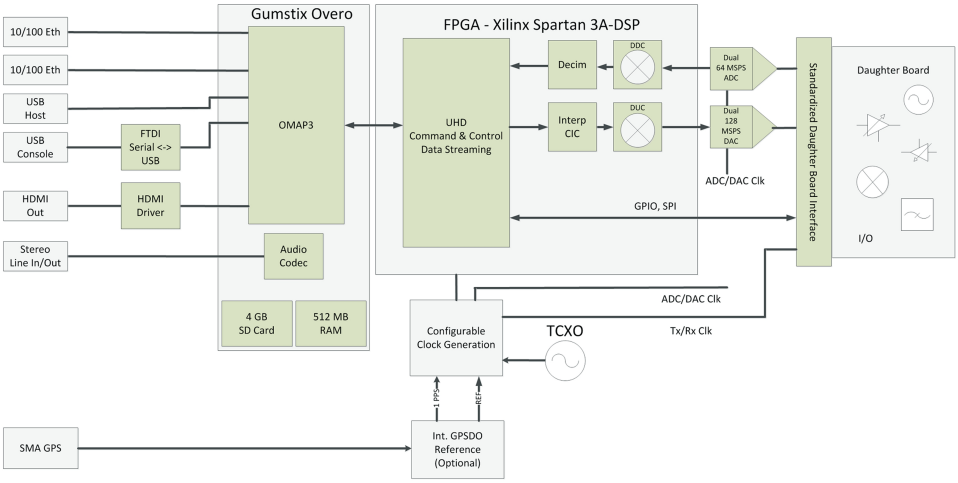


Figure 3.3: USRP E100/E110 Block Diagram, from [15]

3.2.3 Network series

Equipped with the same FPGA of the embedded series also with the N210 featuring a larger device. regarding signal conversion, it provides better performance in both cases, analog to digital and digital to analog. This characteristic allows it to provide both a higher bandwidth and dynamic range when compared with the previously mentioned devices. Multiple N200/210 devies can be used in a MIMO configuration. This feature is possible due to the existent expansion port that allows then to be properly synchronized.

The figure 3.4 and table 3.4 resume this devices main features.

Price	**1500 $ / 1700 $**
FPGA / DSP	Xilinx Spartan 3A - DSP
Memory	
Ethernet	Gigabit Ethernet
USB	
Communications	up to 50 MHz GigE streaming Gigabit Ethernet interface to Host SMA Connector IN / OUT
User IOs	
Programming	
A/D Conversion	Dual 100 MSPS / 14 bits ADC
D/A Conversion	Dual 400 MSPS / 16 bits DAC

Table 3.4: USRP N200/N210 Overview

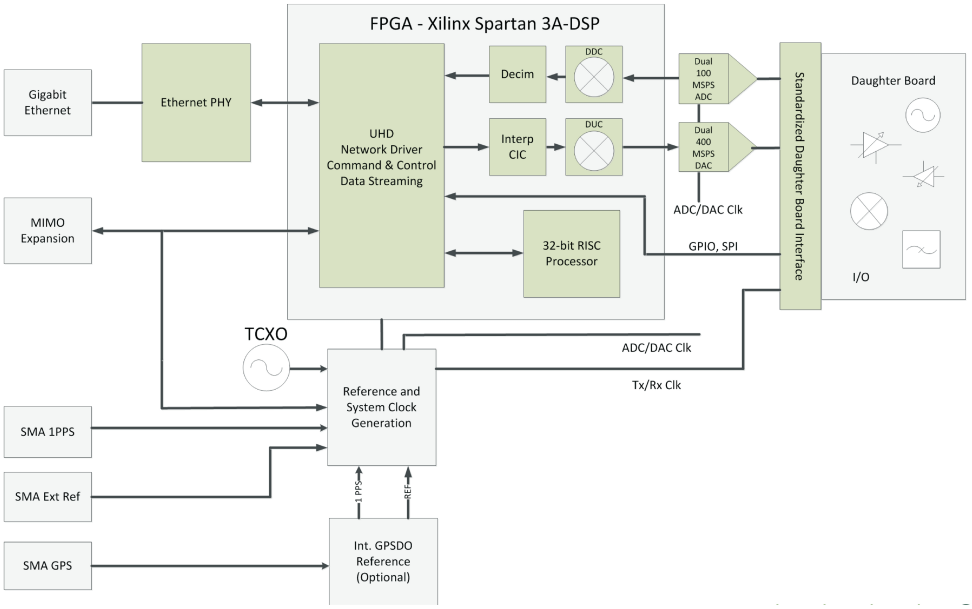


Figure 3.4: USRP N200/N210 Block Diagram, from [16]

3.3 Sundance

The three platforms presented next, SDR 3.0, SDR 4.0 and RadioGiga 2.0 differ from one another in terms of complexity, memory and processing capabilities and it's for the project designed to choose one that suits the project. When compared with all the USRP items, they have a much more complex architecture and a even much higher performance capabilities.

All of these three platforms are equipped with a scalable microprocessor based on high performance FPGAs and DSPs as well as an embedded processor, a Power PC440.

This feature aggregate grants them a high flexibility on the most varied aspects. They are capable of being used in all before mentioned implementation topologies, more hardware or software oriented. The high capable data converters offer them a large processable bandwidth allowing multi communication protocols to be in consideration.

Apart from the Radio Giga 2.0, that only is equipped with analog to digital converters, all the remaining are equipped with both type converters.

The following figure 3.5 and table 3.5, figure 3.6 and table 3.6 along with figure 3.7 and table 3.7, resumes respectively the main features of the SDR 3.0, SDR 4.0 and Radio giga 2.0.

Price	**13,048.99 €**
FPGA / DSP	Virtex -5 and Virtex - 4 FPGAs and two 1GHz C6455 DPSs
Memory	2 x 256MB DDR2 SDRAM, 2 GB DDR2 SDRAM, 9MB ZBT SRAM
Ethernet	Gigabit Ethernet (for remote network reconfiguration)
USB	Usb 2.0
Communications	RJ 45 RS 232 USB SATA FireWire RS-485
User IOs	
Programming	
A/D conversion	Dual 14-bit 250 MSPS
D/A conversion	Dual 16-bit 800 MSPS

Table 3.5: SDR 3.0 Overview

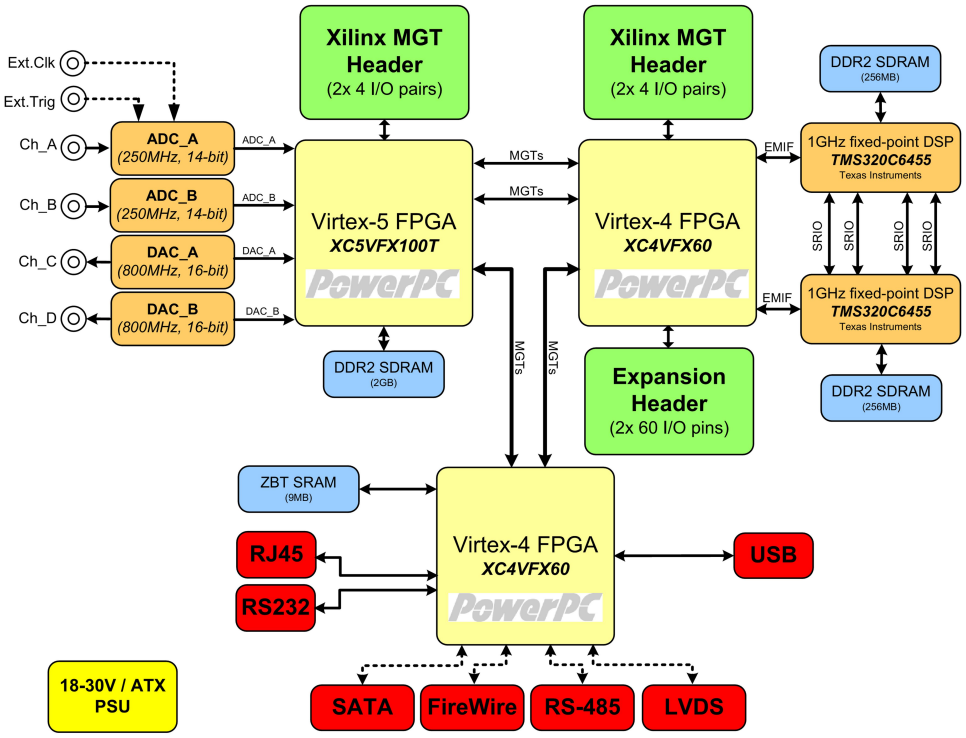


Figure 3.5: SDR3.0 Block Diagram, from [17]

Price	**15,352.43 €**
FPGA / DSP	Virtex -5 and Virtex - 4 FPGAs and two 1GHz C6455 DPSs
Memory	2x2GB DDR2 SDRAM, 2x256MB DDR2 SDRAM, 9MB ZBT SRAM
Ethernet	Gigabit Ethernet (for remote network reconfiguration)
USB	USB 2.0
Communications	RJ 45 RS 232 USB SATA FireWire RS-485
User IOs	
Programming	
A/D Conversion	Dual 8-bit, 1000MSPS
D/A Conversion	Dual 14-bit, 1000MSPS

Table 3.6: SDR 4.0 Overview

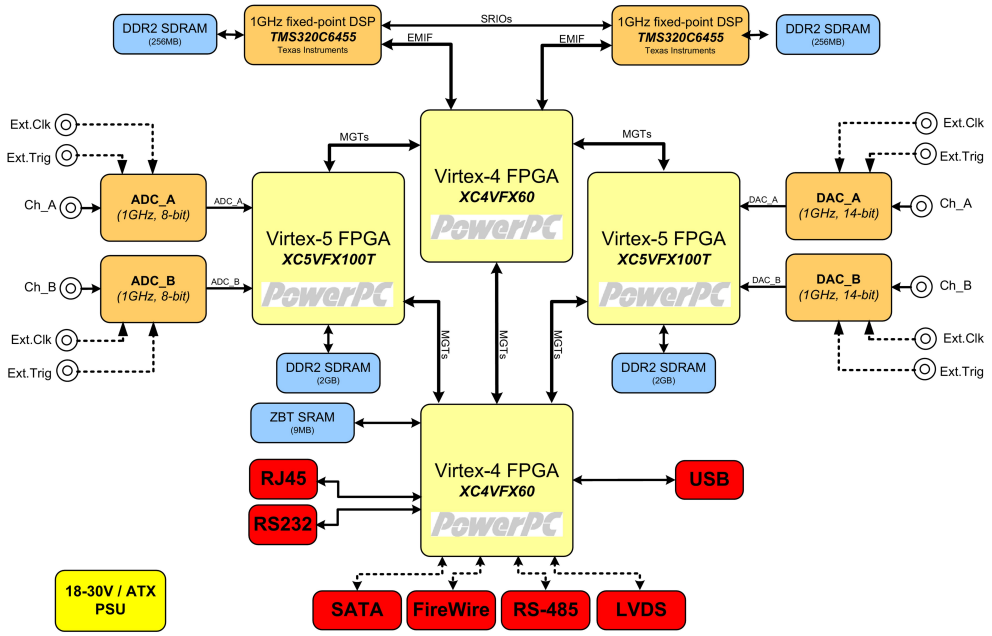


Figure 3.6: SDR4.0 Block Diagram, from [18]

Price	**17,655.87 €**
FPGA / DSP	Virtex -5 and Virtex - 4 FPGAs and two 1GHz C6455 DPSs
Memory	3x2GB DDR2 SDRAM, 2x256MB DDR2 SDRAM, 9MB ZBT SRAM
Ethernet	Gigabit Ethernet (for remote network reconfiguration)
USB	USB 2.0
Communications	RJ 45 RS 232 USB SATA FireWire RS-485
User IOs	
Programming	
A/D Conversion	6 port, 8-bit, 1000MSPS
D/A Conversion	None

Table 3.7: Radio Giga 2.0 Overview

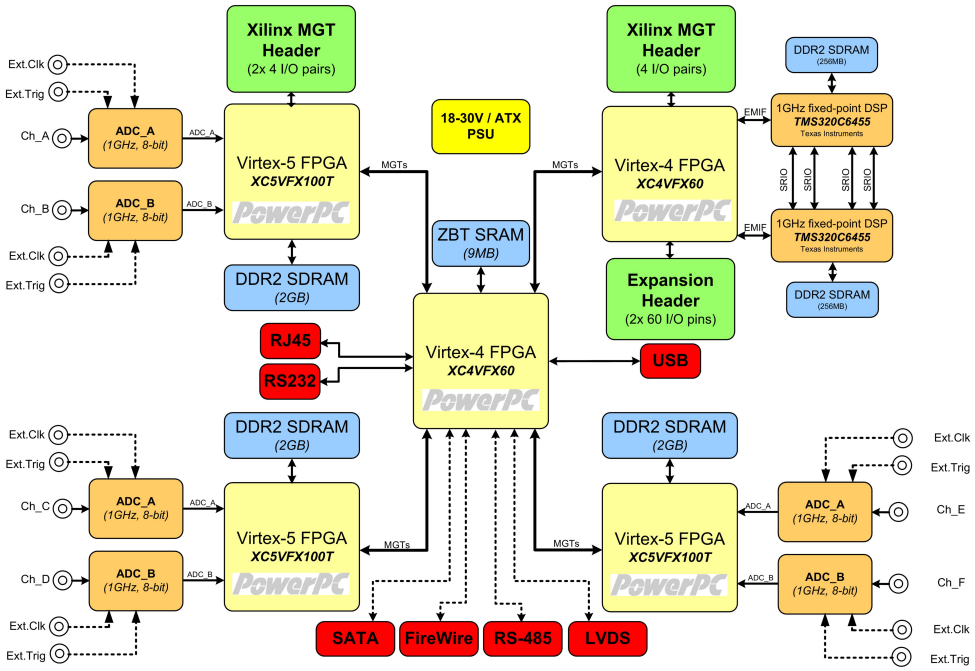


Figure 3.7: RadioGiga 2.0 Block Diagram, from [19]

3.4 Rational to build a new SDR Platform

The analysis of the previous section easy implies that a large number of platforms suitable for SDR purposes is available on the market.

Different characteristics , prices or implementation topology are aspects to be considered when academic purposes is in perspective.

It would be interesting to have a platform able to be used in all topologies, hardware, software or both, in order to test and implement a large variety of SDR systems. A platform centered on competent reconfigurable hardware device either designed or capable of performing specific signal processing functions. Regarding digital signal bandwidth, a set of data converters that can should be able to ensure enough bandwidth in order to cover some communication protocols. Integrating digital and analog interfaces is also a relevant feature, it would allow an easy integration with other systems as well as communication with other devices.

Summarizing, and open and flexible platform with easy access to important signals, easy integration with good development/debugging tools and enhances the available the SDR key features in exchange for an attractive price.

Chapter 4

Project Specification and Design

This project's main motivation relies on the conception of a hardware platform suitable for SDR purposes. It should enable the use of various usage scenarios, with preference for hardware, software or both. In order to that shall include both digital and analog communication interfaces.

All the support hardware needed for capture and sample an analog signal as well as the inverse process, converting the digital signal and provide it to the analog output.

The following sections will describe in further detail the platform's features as well as the selected hardware to perform them.

4.1 Features specification

After analysing different perspectives was reached an agreement regarding the final architecture. The following section present a more detailed overview of the expected features of this project's platform.

4.1.1 Reconfigurable Hardware

The platform should be centered on an FPGA device. The interaction between the FPGA and the platform should be hard wired to it or through an FPGA module containing not only the reconfigurable device but also some other hardware relevant to the project. The mentioned FPGA must have compatible features regarding SDR systems, considerably logic capability and enough IO pins available to ensure the communication with all the needed hardware.

4.1.2 Data Conversion

It will be considered that the analog signal is delivered to the analog inputs already with a frequency to be proper digitalized (IF Digitalization Architecture).

Seeking a better signal quality and reasonable available bandwidth, the ADCs and DACs should have a number of resolution bits between 12 and 14 and a sampling frequency between 100 MSPS and 135 MSPS. Once it might be possible to receive a signal in quadrature modulation, it would be convenient one individual component for both I and Q interfaces or a dual-port device.

4.1.3 Communications Interfaces

Being a platform with academic purposes one of the main present objectives, it is almost mandatory to have one or more communication interfaces. This would allow different feasible implementation topologies alongside with ensuring interconnection with other devices. Demanding a certain level of performance the possible interfaces could be USB 2.0 and Fast/Gigabit Ethernet.

4.1.4 Programming/Debugging Interfaces

Regarding programming and debugging tools two options should be considered. Relating debugging issues there is no other way except a feature cable. It is a more expensive option since the cable, normally using Joint Test Action Group (JTAG) interface, is a little expensive, but developing prototype system is vital to have available good debugging techniques. Using this cable would also allow programming skills as well as interaction with development software tools provided by the reconfigurable hardware manufacture

The other option consists on developing the programming structure. This option would not allow debugging techniques but provides a much inexpensive programming solution. In order to achieve a more holistic platform, the previous mentioned communication interfaces should be considered to perform this task.

4.1.5 User Interaction

Regarding prototype systems testing along other required functions it would be interesting to integrate some basic interfaces. Light-Emitting Diode (LED)s, switches or push buttons will provide both end-user and system's developer a easy way to interact with the platform.

4.1.6 Power Supply

As a matter of simplicity should be chosen, as far as possible, hardware parts with the same supply voltage. this measure would prevent the overuse of power regulators.

4.2 Block Diagram

For a better perception of the described characteristics the figure 4.1 represents a global block diagram and tries to illustrate the platform's desired architecture. Shows how the it is centered on the reconfigurable hardware and all the other devices are connected to it.

4.3 Design and Hardware Selection

The following sections explain why was chosen certain hardware over others.

4.3.1 FPGA Module

Regarding the choice of the FPGA Module two manufactures were considered, *Enclustra* and *Trenz*. The FPGA device family in consideration is the *Spartan-6* from *Xilinx*. It was told to be the recommended for new projects and, for example, in terms of memory resources is much similar to the *Spartan-3A* used on the USRP devices.

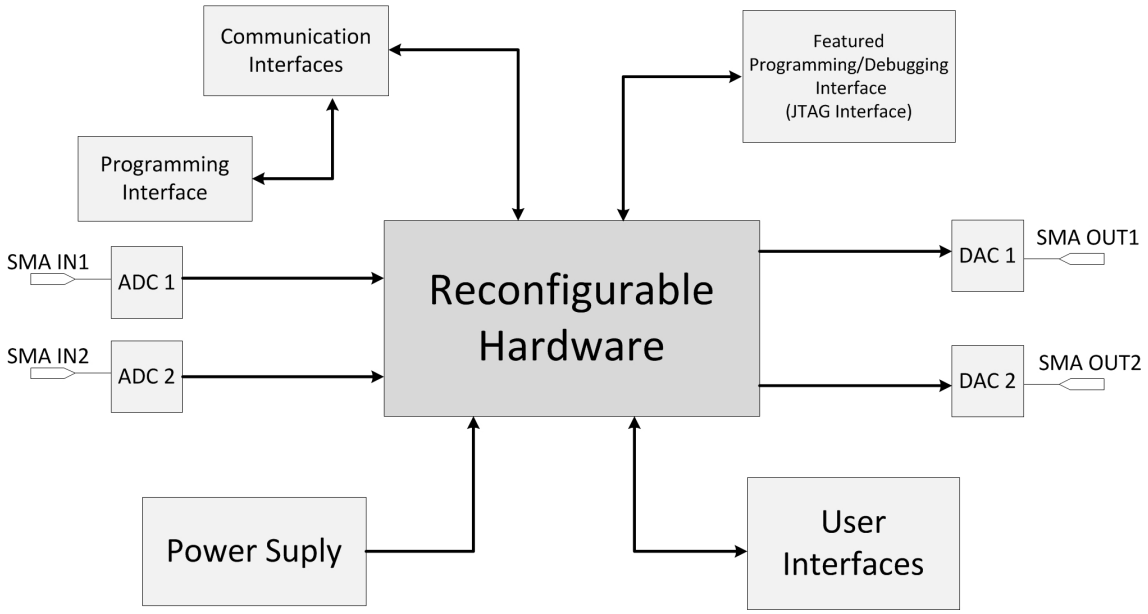


Figure 4.1: Platform Block Diagram

From *Enclustra* the analyzed module was the *Mars MX2* and from *Trenz* two modules, the *TE0630* and the *GigaBee*. In general, all the three modules, are very similar with just a few differences.

The *TE0630* module, illustrated on the figure 4.2, is equipped with a built in USB connection controlled by a *Cypress* chip, one SPI Flash with 64 MB, 128 MB DDR3 SDRAM. It have two clock signals connected to the FPGA, one with 24 MHz and another with 100 MHz, is equipped with 4 LEDs and 2 DIP switches. The connection to other boards is assured by to B2B connectors being the power supply delivered by one of these connectors or by the USB port. it has available 55 user I/O pins and it costs 329 €.

The *GigaBee* module, represented on the figure 4.3, instead a USB connection is equipped with an Gigabit Ethernet PHY. It has 109 available user I/O pins and in terms of memory it has two 128 MB DDR3 SDRAM and one SPI Flash with 8MB. The cost is 229 €.

Provided by *Enclustra*, the module considered is the *Mars MX2*, shown on the figure A.5. With a different type of connection, a DDR2 SO-DIMM module, it features also a Gigabit Ethernet PHY, 256 MB DDR2 SDRAM and 16 MB SPI Flash. Has 108 user I/O pins available, one 50 MHz oscillator, possibility of a real time clock. Contains already 4 user LEDs assembled and just needs to be delivered 3.3v, the remaining power ramifications are already made on board. It costs 200 €.

As mentioned before, all the three modules are very similar and the final choice was the *Mars MX2*. Being all much similar this one the cheaper one, the SO-DIMM connector is easy to find on the market and it provides compatibility with future *enclustra* modules. In conclusion, it was the one thought to have more suitable characteristics for the project. It is equipped with a *Spartan-6* XC6SLX45T.

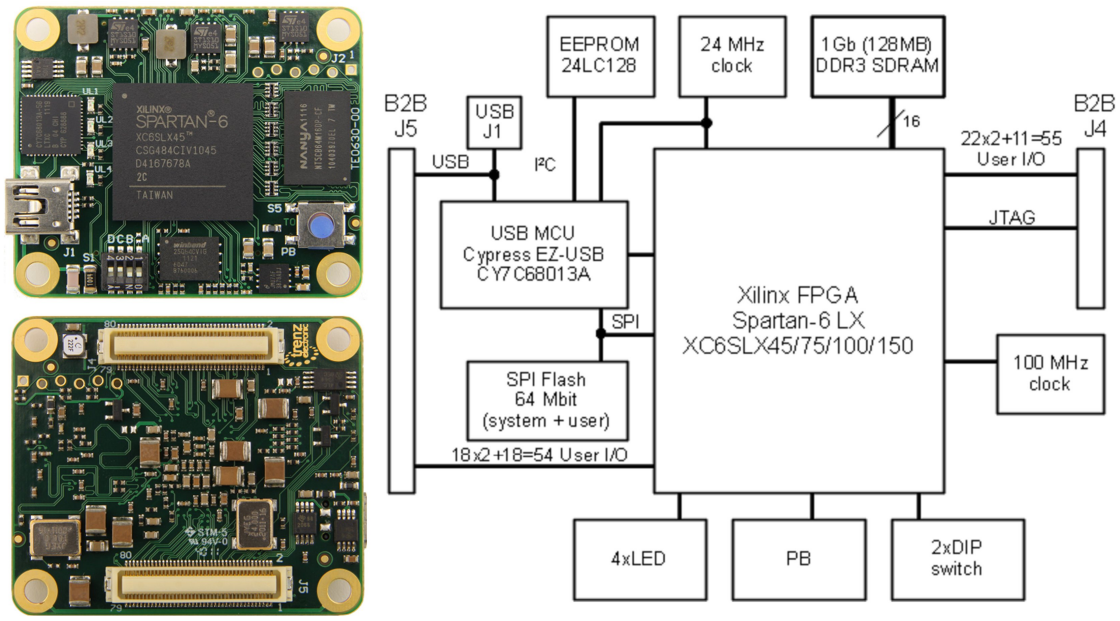


Figure 4.2: *Trenz TE0630*, Top and Bottom View and Block Diagram, from [20]

4.3.2 Data Conversion

To preform analog to digital conversion, was chosen the AD9432 fom *Analog Devices*. It has 12 resolution bits ans it optimized for high speed conversion rates, it can operates up to 105 MSPS. Since it has only one port, two devices were used.

The AD9767, also from *Analog Devices*, was the choice to preform the digital to analog conversion. Using CMOS technology, it features two high quality *TxDAC+*, on chip, with 14 bis each. It has been designed and optimized for processing I/Q data in communication interfaces. It can operate featuring two separate data ports, with dedicated control signals or single interleaved high-speed data port.

The respective signal conditioning needed was chosen according with the manufacturer's advice. Is used as operational amplifier, an AD8138 from *Analog Devices*, it is responsible for converting the single ended input signal to a differential one in order to be driven by the ADC devices. The dual process occurs with the DAC output signal but on this case using an RF 1:1 transformer. The output differential signal is converted into single ended.

4.3.3 Communication Interfaces

The platform will be endowed with both Gigabit Ethernet and 2.0 USB connections. Once the Ethernet PHY as weel as all needed support hardware is already connected on the FPGA module, to ensure Ethernet communication is only necessary to assemble an Ethernet socket and assure the needed connections between it and the FPGA module.

To handle the USB connection will be used and FTDI controller, the FT2232H. Being the FTDI's 5th generation of USB devices, the FT2232H chip makes a communication bridge from a USB 2.0 High Speed (480 Mb/s) to Universal Asynchronous Receiver/Transmitter (UART) or Multi-Protocol Synchronous Serial Engine (MPSSE). It has the capability of being configured in a variety of industry standard serial or parallel interfaces. Has two independent

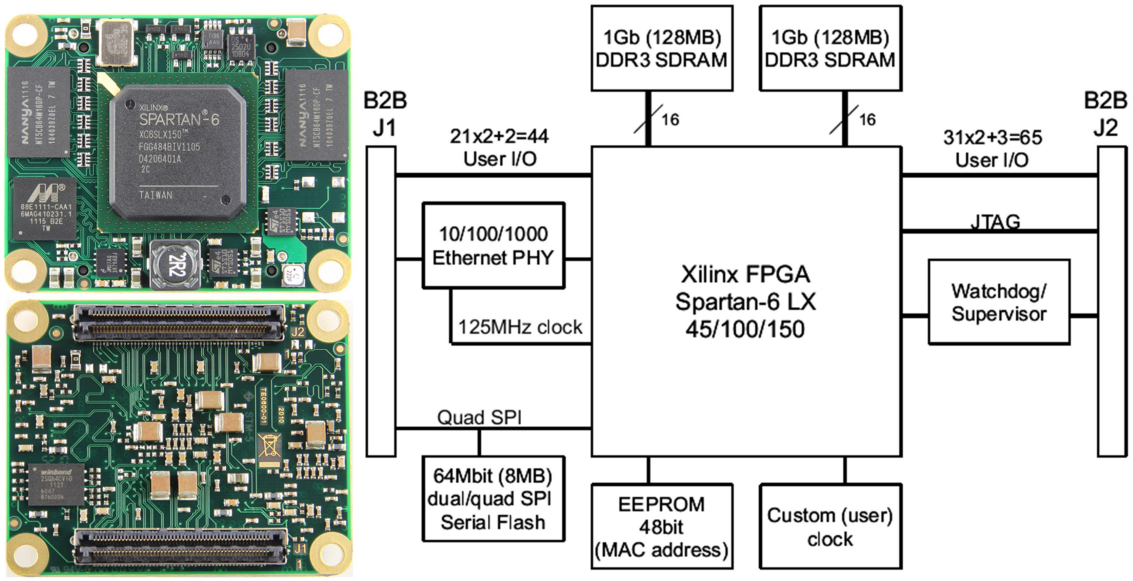


Figure 4.3: *Trenz GigaBee*, Top and Bottom View and Block Diagram, from [21]

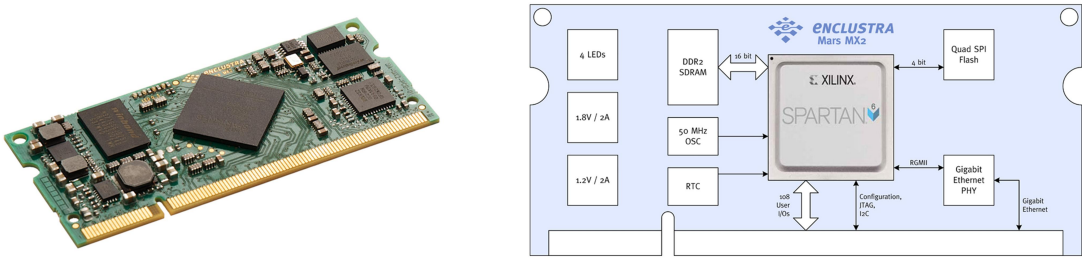


Figure 4.4: *Enclustra Mars-MX2*, Top and Bottom View and Block Diagram, from [22]

interfaces which can be configured as UART, First In First Out (FIFO), JTAG, Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) or bit-bang mode. With no firmware requirement, the FT2232H effectively integrates UBS protocol on a chip. [FT2232H Manual].

It has two communications channels and each one can be configured to perform one of the previously mentioned protocols. On the platform the connections were made in order to have available SPI protocol on one channel and a synchronous FIFO on the other.

4.3.4 Programming/Debugging Interfaces

The FPGA module provides all the JTAG interface signals. In order to use a featured programming/debugging cable is only needed to connect those signals to a proper connector and this function can be done using the appropriate cable in association with the correspondent development software.

One of this project's objectives is to implement one way of programming the FPGA without featured accessories. This was another reason to chose the FTDI USB controller.

The selected FPGA module, Mars-MX2 from *Enclustra* suggests a way of programming

the platform using USB connection. For that purpose they recommend the use of an FTDI chip, the FT2232H or similar.

In order to simplify the assembly process, instead of using the the FT2232H chip,will be used a FT2232H Mini Module, represented on the figure 4.5. The FT2232H chip, as well as all the support hardware needed, is previously assembled on a Print Circuit Board (PCB) board and the connection is ensured through two male connectors with 26 pin each. Besides simplify the assembly process, it also makes it easier to replace in case of malfunction.

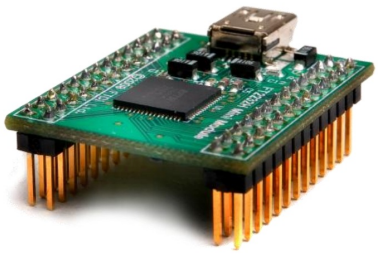


Figure 4.5: FT2232H Mini Module, from [23]

4.3.5 User Interfaces

The end-user/developer interaction is ensured by eight switched and eight LEDs. They are connected to the FPGAs available I/Os pins being the switches used as inputs and the LEDs as outputs. Some push buttons are also assembled but not for generic I/O purposes. They have their function and it can not be reprogrammed.

4.3.6 Power Supply

The diversity of hardware chosen deliberated two required voltage levels, 3,3V and 5V, digital and analog. The power network consists on a switched regulator providing voltage signal to other secondary linear regulators.

Was used a competent switched regulator, the LM2267, that receives the Alternating Current (AC) voltage a deliver an output with 5V/5A. This signal is used to further power up two LM1084, for both analog and digital 3.3V voltage level, and a *Texas Instruments* UCC383TDKTTT-3 that provides a digital 3.3V level only for the FPGA module.

Once the ADCs require a analog 5V level, was used another linear regulator, the MC78M05CDTX, providing the analog voltage to both devices. To avoid undesirable noise distortion, this regulator receiver the signal also directly from the AC voltage.

4.3.7 Bill of Materials

The table 4.1 summarizes the overall price spent on hardware components as well as the assemble process.

4.3.8 Final Platform

The final block diagram is described on the figure 4.6, in which all the important components are mentioned, as well as the respective connections.

Component	Reference	Quantity	Price
Resistors and capacitors	SMD (0805)	–	aprox. 25€
Dual DAC	AD9767ASTZ	1	19,08 €
ADC	AD9432BSTZ-105	2	43,03 €
SMA Conector	CONSMA002	4	2,84 €
Dual Multiplexer	FSUSB30	2	1,38 €
Socket SODIMM	1565917-4	1	3,59 €
Header FTDI 2x13	NPTC122KFMS-RC	2	2,39 €
Header JTAG 2x7		1	3,81 €
Header JTAG 1x6	68000-406HLF	1	0,30 €
Socket Ethernet	6605814-6	1	10,15 €
Dual Transistor Chip	EM6K6T2R	1	0,39 €
LED	SML-LXT0805GW-TR	15	0,34 €
Batery Socket	BU1632SM-JJ-GTR	1	0,50 €
Switched Regulator	LM 22677	1	9,64 €
Secundary regulator	UCC383TDKTTT-3	1	2,54 €
Secundary Regulator	LM 1084	2	3,62 €
Dual Flip Flop	SN74AUP2G79DCUR	1	0,50 €
Diode	MBR745	1	0,68 €
Diode	P6SMB30A	1	0,44 €
MF R300	MF R300	1	0,53 €
Power jack	PJ-002BH-SMT	1	1,54 €
Jumper 1	87224-2	2	0,68 €
Jumper 2	87227-2	3	1,66 €
Opamp	AD8138	2	4,72 €
Regulator ADC	MC78M05CDTX	1	0,53 €
Switches	EG1218	8	0,58 €
Mars MX2 Module		1	200,00 €
USB Minimodule		1	24,00 €
PCB Board		1	148,74 €
Total (Price x Quantity)			589,68 €

Table 4.1: Bill of Materials

The final platform is presented on the figure 4.7. The PCB board was ordered in a external company, the *Euro Circuits*. Its consists in a four layered PCB in which the top and bottom layers were used to signal routing and the inner layers to route power networks.

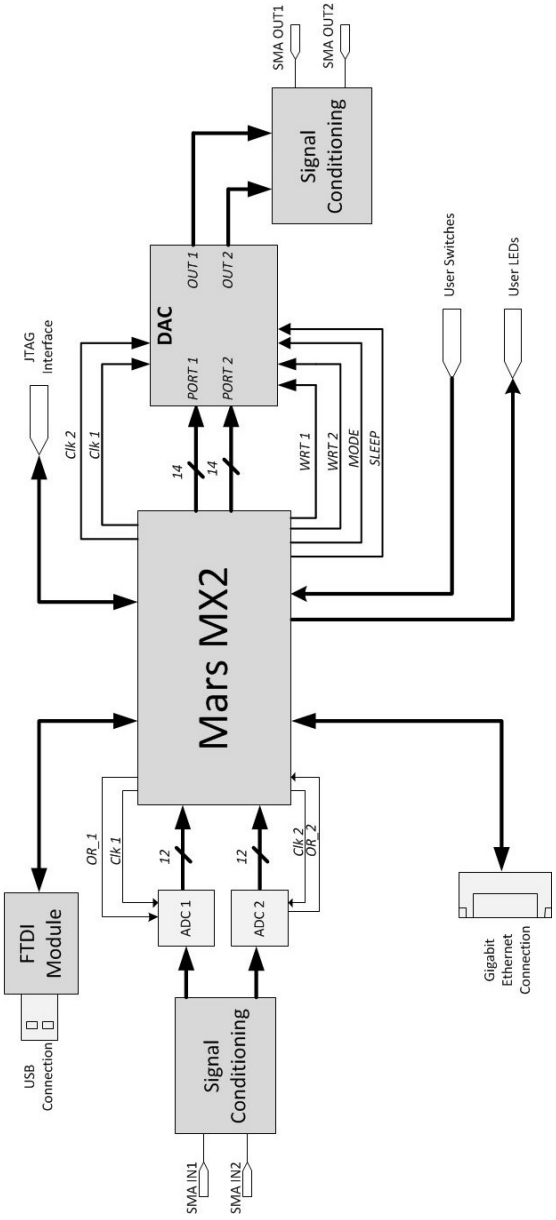


Figure 4.6: Final Block Diagram

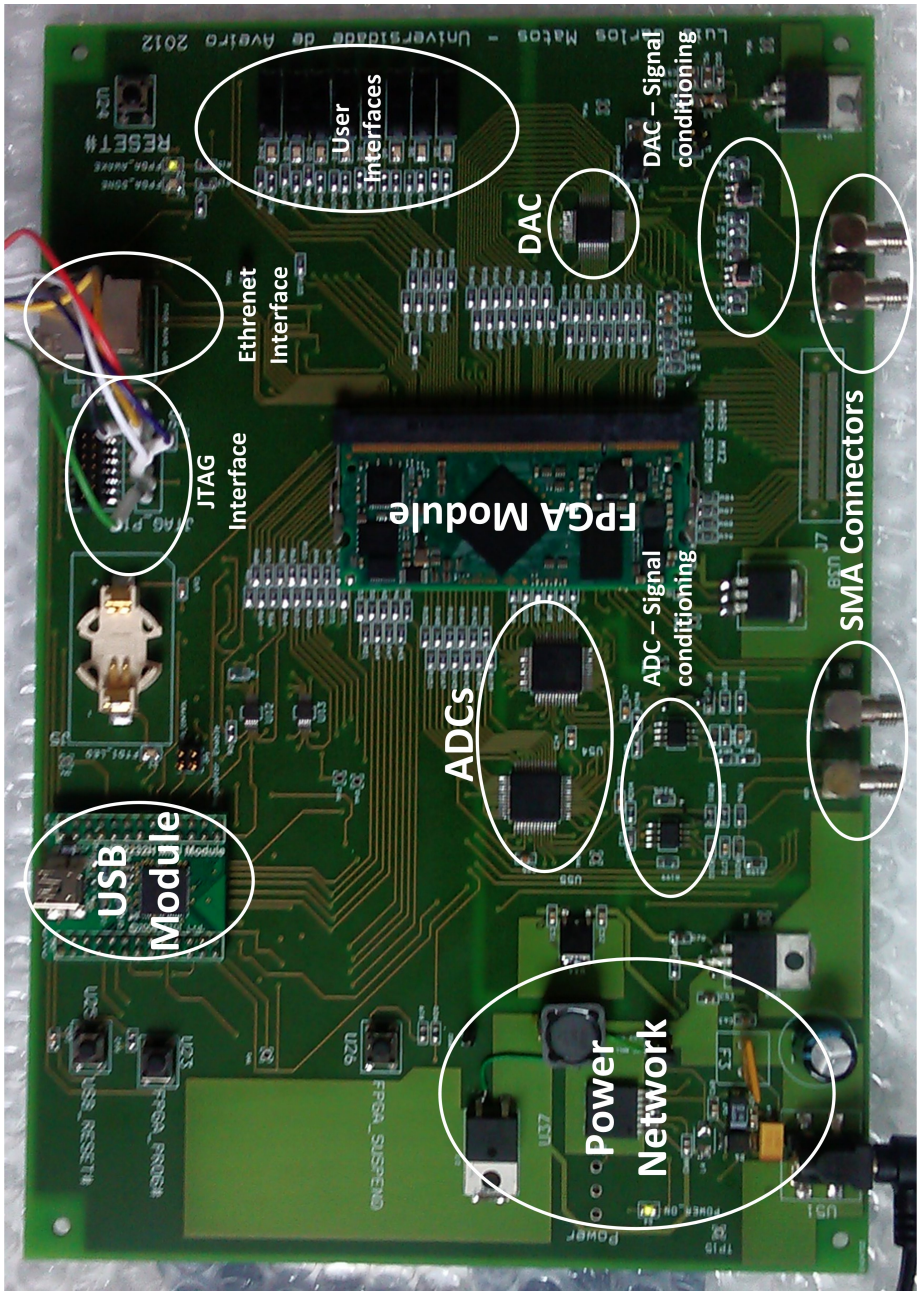


Figure 4.7: Final Platform

Chapter 5

Platform Tests and Results

Now that the board is fully build and assembled, some tests are needed to ensure that everything is working properly, according to what was planned. Some preliminary tests were made already, before the assemblage of the relevant chips, ADCs, DACs and the FPGA module. All the voltage regulators were checked to guarantee that the right voltage was being delivered to each component.

The conducted platform tests were as follows:

- **IO Testing** Performed to check the good functioning of user interfaces.
- **DAC Tests** performed to evaluate the DACs behaviour.
- **ADC Tests** performed to evaluate both ADCs functioning.
- **ADC + DAC Tests** performed to evaluate the overall data conversion operation

In order to be able to perform the referred tests, was need to assemble a setup test environment. It was composed by a power source, a signal generator, a digital oscilloscope, a JFAG programming cable and a computer endowed with a development software tool. The software used was the *Xilinx ISE design suite 13.1*. The figure 5.1 describes the setup test assembled.

5.1 Digital IO Test

This test was made to check is the model's connector was well assembled. If verifies if the pins on the connector were well welded to the PCB's copper tracks. The figure 5.2 enhances the platform's used hardware on this test.

Using the software development tool available, Very high speed Integrated circuits Hardware Description Language (VHDL) programming language was used to program wire connections between different I/O pin within the FPGA, the one connected to the switches and LEDs.

The figure 5.3 is a picture taken from the platform while the program was running. It shows the LEDs blinking when the switches were activated. The current consumption was 550 mA.

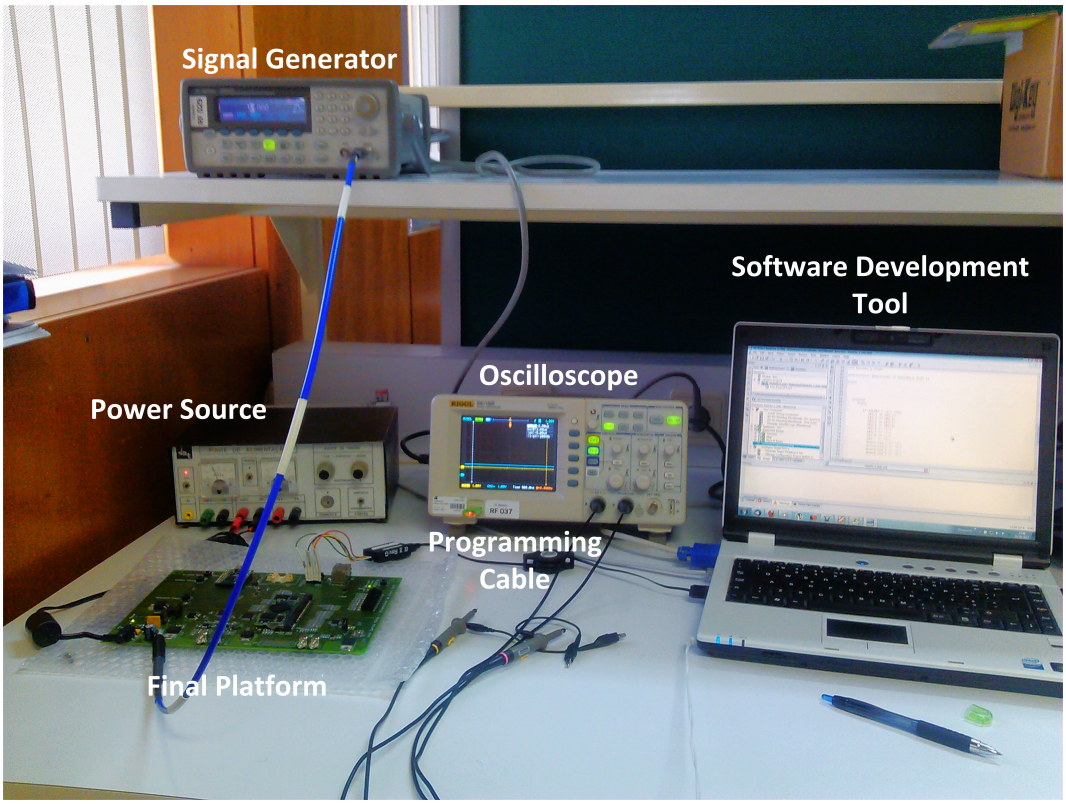


Figure 5.1: Setup Test

5.2 DAC Test

The following section describes the corresponding DACs behaviour tests. The figure 5.4 shows what platform parts where used to perform the next described tests.

5.2.1 Limit Range Test

This test has the objective of checking of the DAC's output signal corresponds with the expected values. To test this feature the development program consists on changing all the DAC's input signals from "0" to "1" repeatedly according to a certain clock frequency.

The figure 5.5 was taken from the DAC's manual. It explains how timing related is the output signal. Once the rising edge of the "write" signal must occur before of at same time as the clock one, it was to be the clock's complementary signal.

In order to test their proper work the "SLEEP" and "MODE" control pins were connected to two different switches.

The figures 5.6 and 5.7 represent the achieved results for two different clock frequencies, 25 and 50 MHz respectively. After the FPGA is programmed, the platform was consuming 600 mA using the 25 MHz clock and 625 mA with 50 MHz.

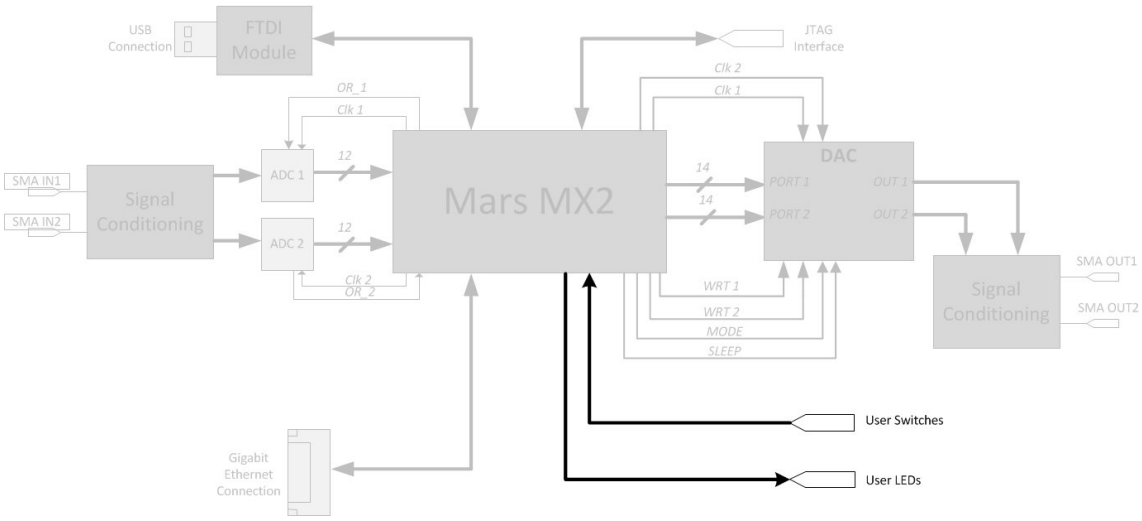


Figure 5.2: Digital IO Test - Diagram

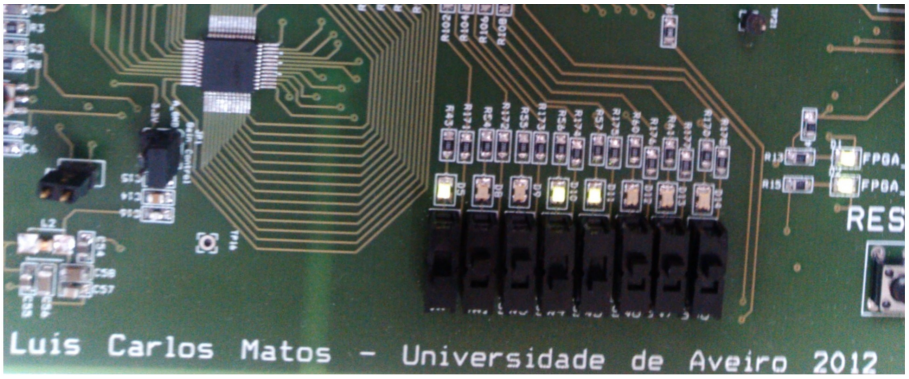


Figure 5.3: Digital IO Test - Platform View

5.2.2 Test with an Output Sine wave

This test tries to put at the analog outputs a sin wave. Using the Xilinx IPCORE synthesizer, a ROM memory was programmed. The 256 positions of 14 bits each were fulfilled with precalculated sin values. A binary counter with 8 bits length addresses the Memory, and its output is connected to both DAC ports.

Once again, some trials were performed changing the clock frequency.

The figure 5.8 show the output wave forms collected. Since the memory block used contained 128 values, the output wave frequency is calculated by the following expression:

$$f_{output} = \frac{1}{\frac{1}{f_{clk}} \times 128}$$

Using 25 MHz we have 195,3 kHz, when changing the clock frequency to 50 MHz we have 390,6 kHz. The current consumption varies from 590 mA when using 25 MHz clock to 600 mA using 50 MHz.

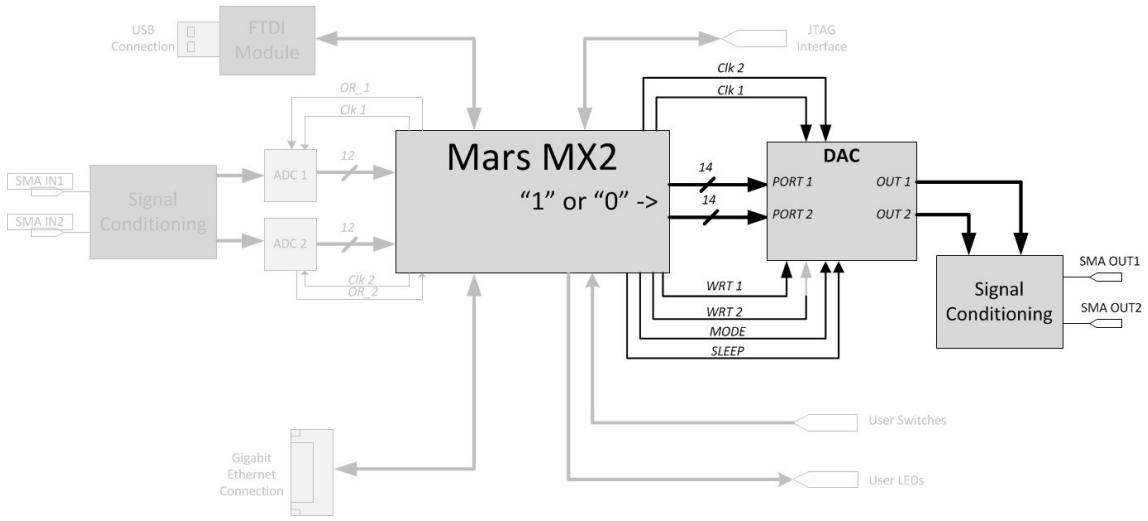


Figure 5.4: DAC Tests - Diagram

5.3 ADC test

Regarding the ADC devices, two more tests were performed, testing the range limit and loopback with the DAC. The figure 5.9 shows the test related used resources on the platform.

5.3.1 Range Test

Similar to the Test 1, this one testes the ADC's operation. The eight most significant bits of each ADC port are connected to a MUX 2:1, that using a selection bit routes the the desirable signal to the eight user LEDs on the board. the "OR" pins (out of range) are connected to two of the module LEDs, active low, that points out when the signal entering the ADC has an amplitude beyond the conversion range.

Other alternatives of this test were tried changing clock frequencies as well as connecting the less significant bits to the platform LEDs.

In this test the current consumption were 650 mA. the figure 5.10 tries to illustrate how the signal was delivered to the platform as well as the blinking LEDs. The platform only has 8 LED thus was tried with the most significant bits and then with the less ones. The results were as expected.

5.3.2 ADC to DAC Loopback

On this test all the twelve bits from each ADC port are connected to the twelve most significant bits of the corresponding DAC port. The main idea is to drive an analog signal into the ADCs and check if can be found a similar signal, same shape and frequency, once reconverted by the DAC. Once this test is preformed and approved, can be said that the analog to digital and digital to analog conversion chains are fully functional.

With this test can also be checked the sampling theorem, which states:

> *"If the highest frequency contained in an analog signal $x_a(t)$ is $F_{max} = B$ and the signal is sampled at a rate $F_s > 2 \times F_{max} \equiv 2 \times B$, then $x_a(t)$ can be exactly recovered from its sample values"* [25]

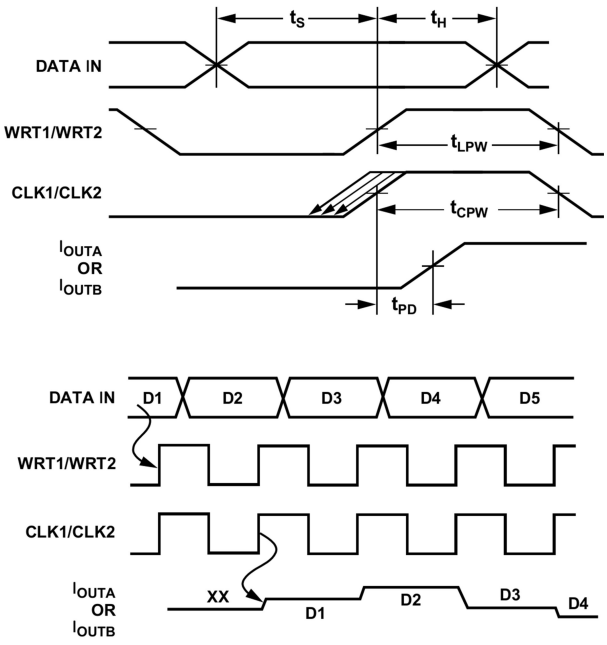
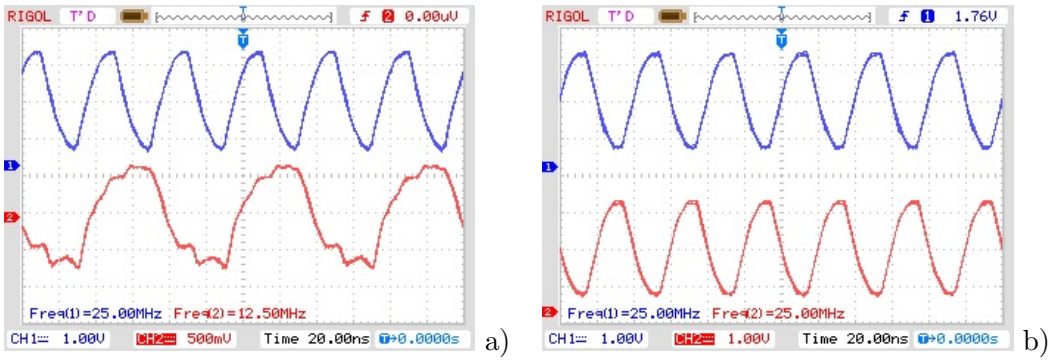


Figure 5.5: DAC Time Diagram, from [24]

It means that if the frequency of the input signal exceeds half of the clock frequency, the signal on the DAC's output would be distorted.

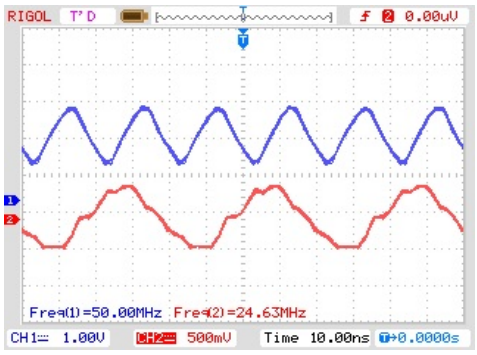
The wave forms resisted on the figure 5.11 were acquired using a a 50 MHz clock. The result was as expected a wave with 10 MHz, as was used a sin wave with the same frequency by the signal generator.

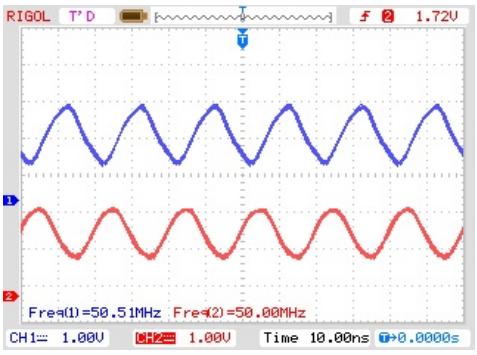
On the figure 5.12 was used a clock with 25 MHz. in the a) the input signal frequency was 12.5 MHz, the signal was recovered. On b), the input was about 14MHz and as expected once the sample frequency is less than two times the input signal frequency, that signal can not be recovered. On this test the platform consumed 625 mA.



a) b)

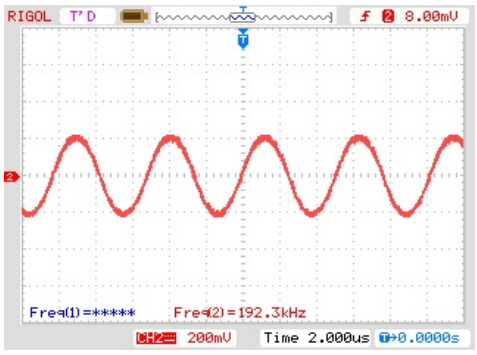
Figure 5.6: DAC Limit Range Test 25 MHz - a) Channel 1: Clock, channel 2: Dac Output b) Channel 1: Clock, channel 2: Dac "write" signal

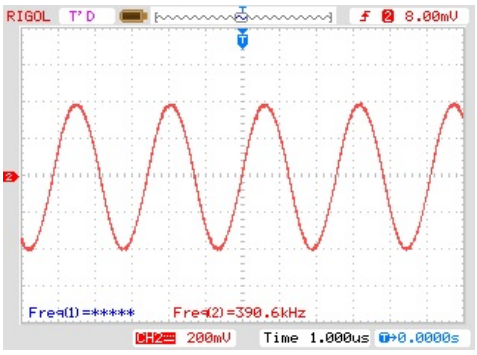


a)


b)

Figure 5.7: DAC Limit Range Test 50 MHz - a) Channel 1: Clock, channel 2: Dac Output b) Channel 1: Clock, channel 2: Dac "write" signal



a)


b)

Figure 5.8: Test With an Output Sin wave - a) DAC output with a 25 MHz clock b) DAC output with a 50 MHz clock.

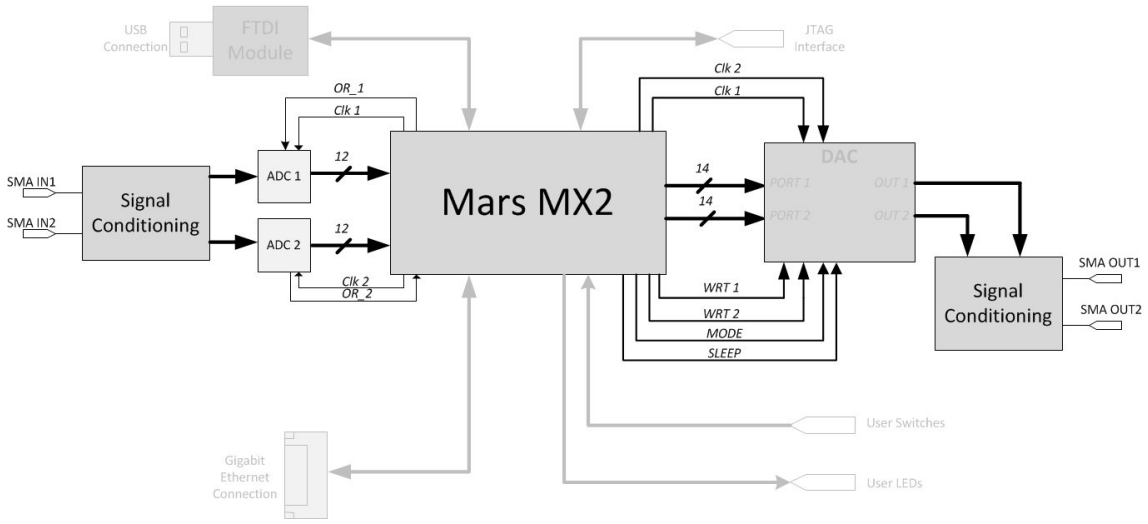


Figure 5.9: ADC to DAC Loopback - Diagram

5.3.3 Other Possible Tests

In order to fully test the final platform, other tests should be performed. The USB as well as the Ethernet connections should be verified. Also related to data conversion,should be performed tests in order to evaluate the maximum input signal frequency acceptable on the platform, in other words how fast can the data converters operate.

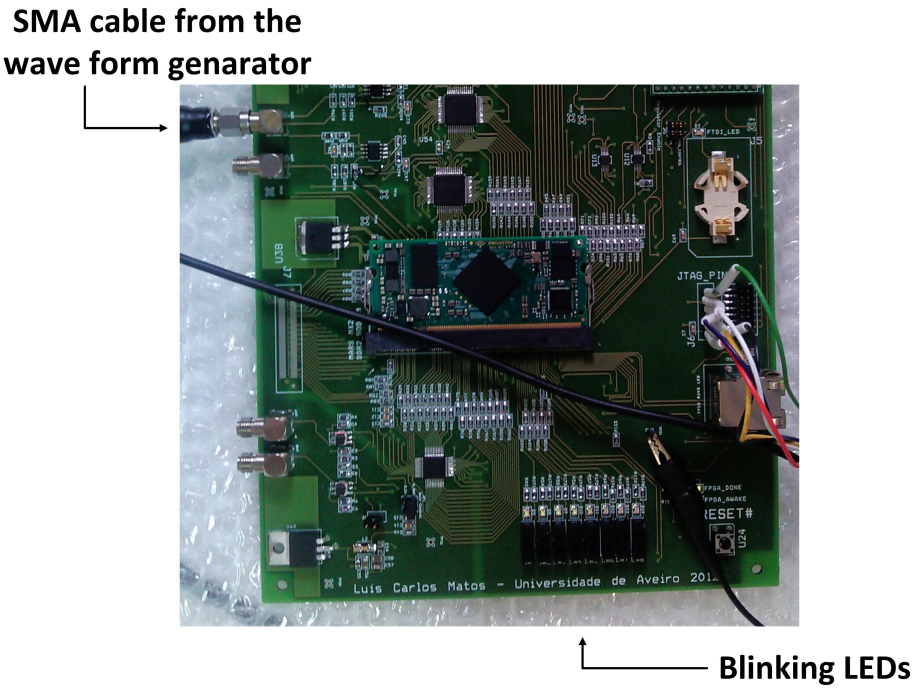


Figure 5.10: ADC Range test - Platform View

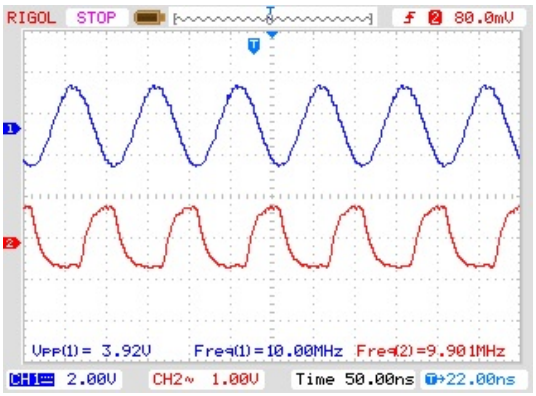
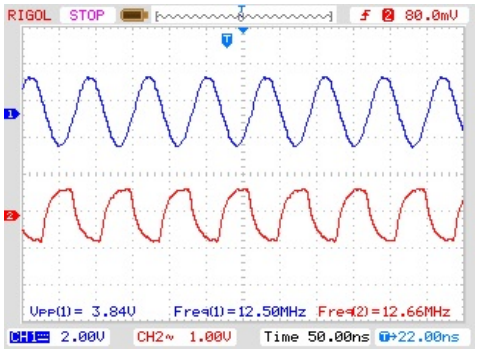
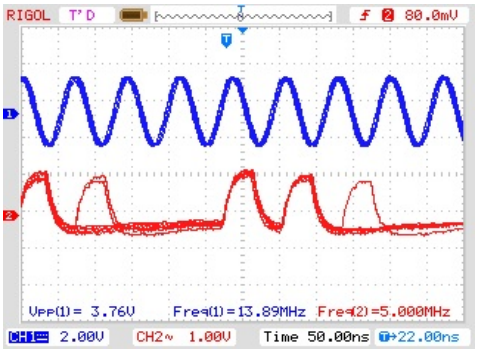


Figure 5.11: ADC to DAC loopback - Channel 1: Wave from the signal generator, channel 2: DAC Output



a)



b)

Figure 5.12: ADC to DAC loopback - Channel 1: Wave from the signal generator, channel 2: DAC Output

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The main objective of this dissertation was the design, assemble and test of a platform suitable for SDR projects assuring a good balance between features, performance and overall price.

After some bibliography analysis was decided to use an FPGA of the the *Xilinx's Spartan-6* family, the low-cost, low-power devices advised to use on the design of new projects. A modular system has been preferred, it was the best thought solution in order to ensure, among other characteristics, an easier upgrade, i.e., was ensured compatibility with future *enclustra* modules so that a different FPGA, memory or other characteristics contained in future models can also be applied to the platform. On behalf of academic purposes, was build a platform able to be used in all SDR implementation topologies, featuring good communication interfaces, data converters and easy interaction with development tools. With easy access to all relevant signals as well as digital interfaces ensuring end-user interface.

Using acquired knowledge as well as manufacturer's suggestions, the key components as well as the relative support hardware was chosen and platform schematic was designed. The PCB board was designed using a four layers topology, being the top and bottom layer used for signal routing and the inner layer for power networks, one for the different power supplies and other analog and digital ground planes. Different from the PCB, that was ordered to a outside campus company, the assembling process occurred entirely at the *Telecommunications Institute*.

In Order to verify the platform proper operation, a test set was thought and programmed as described on the previous chapter, where major features were tested, FPGA programming, data converters and I/O interfaces. Through those test results can be concluded that in general the platform is ready to hold an SDR system that fits the platform characteristics, i.e., with an IF architecture and a frequency suitable for the DACs and ADCs. The FPGA has been programmed without any problem using a featured programming cable from a *Digilent* device (a *Xilinx* cable would suit as well), the user I/Os (switches and LEDs) are working perfectly as well as the DAC and the ADC. In summary the project was well succeeded and its outcome was a robust platform thought to be suitable for academic purposes.

6.2 Future Work

Besides the power supply hardware, that was the first one to be tested, the key components regarding SDR system, DAC, ADC and FPGA, were tested. It follows that, as future work, the remaining components test should be done, the USB module and the Ethernet connection.

The tests performed allow concluding that the key components regarding SDR system, DAC, ADC and FPGA, are properly working. However, as before mentioned, further tests are required to fully evaluate the overall platform performance. The maximum sampling rate acceptable for the data converters good functioning and testing the remaining communication interfaces, USB and Ethernet.

Regarding the USB Module, on the *FTDI* website a complete library of already programmed functions and device drivers can be downloaded in order to use the module as a Virtual COM Port (VCP), in order to perform a real-time way of communication with a computer. Despite not being completely done, the objective of creating a application to program the FPGA might yet be done. All the necessary hardware connection were performed and the remaining tasks to complete that can be done also using, the already mentioned, functions on the *FTDI* website.

The Ethernet connection, once the physical layer is already assembled on the FPGA module, it would be enough to just load an Ethernet controller software in order to test it.

Although no being a primary objective, it would be interesting and advantageous to design and build an RF front-end in order to connect it to the platform and preform a complet SDR system.

Bibliography

[1] J. Mitola III. *"Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio"*. PhD thesis, Department of Teleinformatics, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.

[2] SDR Forum. http://www.sdrforum.org/, March 2009.

[3] Wipro Technologies. White paper: Software-defined radio, a technology overview. 2002.

[4] A. Luiz Garcia Reis, A.F. Barros, K. Gusso Lenzi, L.G. Pedroso Meloni, and S.E. Barbin. Introduction to the software-defined radio approach. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 10(1):1156 –1161, jan. 2012.

[5] R.I. Lackey and D.W. Upmal. Speakeasy: the military software radio. *Communications Magazine, IEEE*, 33(5):56 –61, may 1995.

[6] J. Mitola. "The Software Radio Architecture". *IEEE Communications Magazine*, 33(5):26 – 38, 1995.

[7] L. Antunes. "Software Defined Radio em FPGA". Master's thesis, Universidade de Aveiro, 2009.

[8] J. Mitola III and G. Q. Maguire. "Cognitive Radio: Making Software Radios More Personal". *IEEE Personal Communications*, 6(4):13–18, August 1999.

[9] J. Guenin. "Jim Hoffmeyer on Overview of SCC41". http://www.scc41.org/, September 2007.

[10] Song Wenmiao. Implementation of digital if receiver based on sdr using dsp builder. In *Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications (MAPE), 2011 IEEE 4th International Symposium on*, pages 266 –269, nov. 2011.

[11] A. Saha and A. Sinha. An fpga based architecture of a novel reconfigurable radio processor for software defined radio. In *Education Technology and Computer, 2009. ICETC '09. International Conference on*, pages 45 –49, april 2009.

[12] Hongjiang He and Hui Guo. The realization of fft algorithm based on fpga co-processor. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 3, pages 239 –243, dec. 2008.

[13] Etthus Research. USRP 1 Manual. `https://www.ettus.com/content/files/Ettus_USRP1_DS_FINAL_1.27.12.pdf`. [Online; accessed 5-July-2012].

[14] Etthus Research. USRP B100 Manual. https://www.ettus.com/content/files/Ettus_B100_DS_FINAL_1.27.12_4.pdf. [Online; accessed 5-July-2012].

[15] Etthus Research. USRP E100-110 Manual. https://www.ettus.com/content/files/Ettus_E100-110_DS_FINAL_1.27.12.pdf. [Online; accessed 5-July-2012].

[16] Etthus Research. USRP N200-210 Manual.

[17] Sundance Multiprocessor Technology. SDR 3.0 Block Diagram. http://www.rapiddevsys.biz/pdf/block_diagrams/SDR%203.0fbd.pdf. [Online; accessed 5-July-2012].

[18] Sundance Multiprocessor Technology. SDR 3.0 Block Diagram. http://www.rapiddevsys.biz/pdf/block_diagrams/SDR%203.0fbd.pdf. [Online; accessed 5-July-2012].

[19] Sundance Multiprocessor Technology. RadioGiga 2.0 Block Diagram. http://www.rapiddevsys.biz/pdf/block_diagrams/Radio%20Giga%202.0fbd.pdf. [Online; accessed 5-July-2012].

[20] Trenz Electronics. TE0630 Block Diagram. http://www.trenz-electronic.de/fileadmin/docs/Trenz_Electronic/TE0630_series/TE0630/documents/UM-TE0630.pdf. [Online; accessed 5-July-2012].

[21] Trenz Electronics. GigaBee Block Diagram. http://www.trenz-electronic.de/fileadmin/docs/Trenz_Electronic/TE0600-GigaBee_series/TE0600/documents/UM-TE0600.pdf. [Online; accessed 5-July-2012].

[22] FPGA Solution Enclustra. Mars MX2 User Manual. http://www.enclustra.com/assets/files/Products/FpgaModules/MarsMx2/Mars_MX2_UserManual.pdf. [Online; accessed 5-July-2012].

[23] Future Technology Devices International Ltd. FT2232H Mini Module Datasheet. http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_FT2232H_Mini_Module.pdf. [Online; accessed 5-July-2012].

[24] Analog Devices. AD9767 Datasheet. http://www.analog.com/static/imported-files/data_sheets/AD9763_9765_9767.pdf. [Online; accessed 5-July-2012].

[25] J. Proakis and D. Manolakis. *"Digital Signal Processing - Principles, Algorithms and Applications"*. Prentice Hall, 4 edition, April 2006.

Appendix A

Schematic Diagrams

The electric schematics were designed using the OrCAD Capture software from *Cadence systems*. The following images illustrate in detail the resulted schematic sheets.

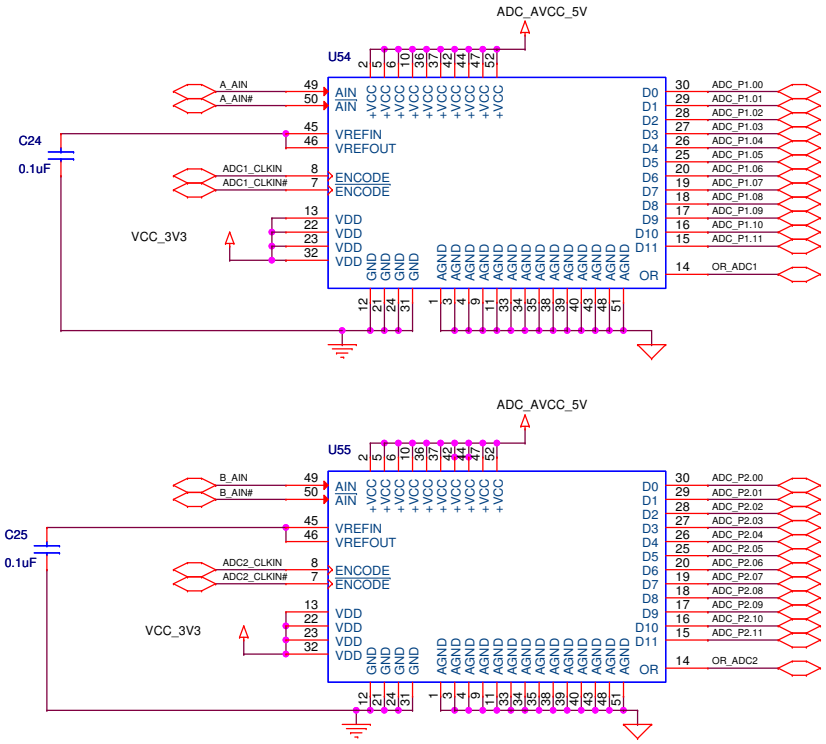


Figure A.1: ADC Schematic

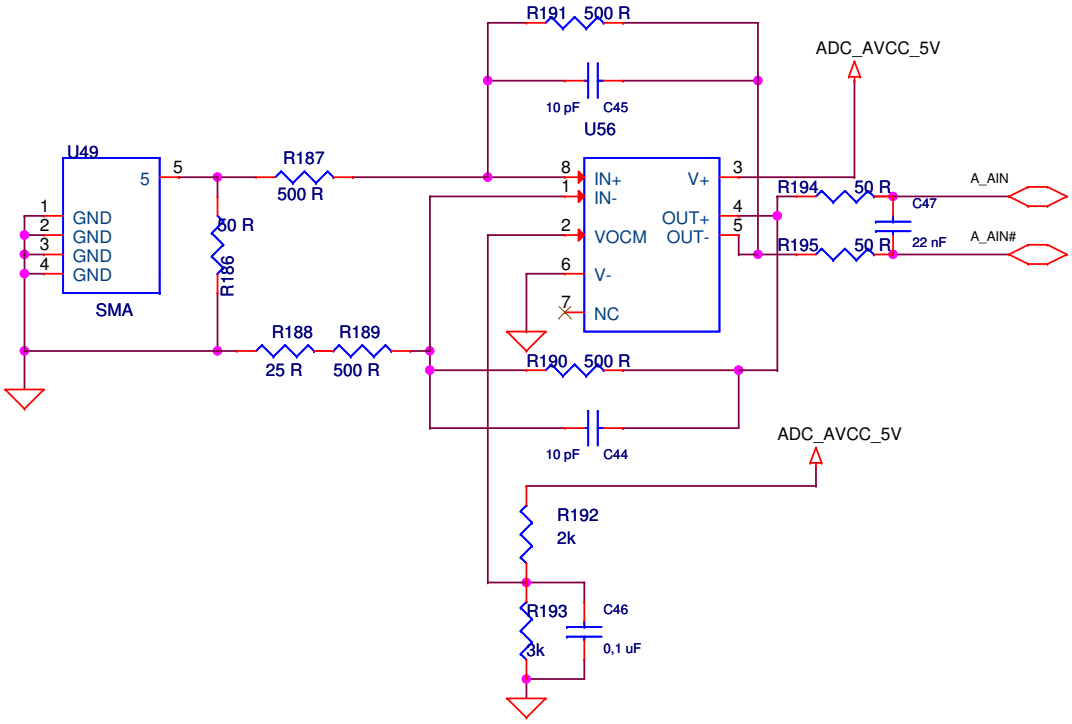


Figure A.2: ADC Input Opamp Schematic

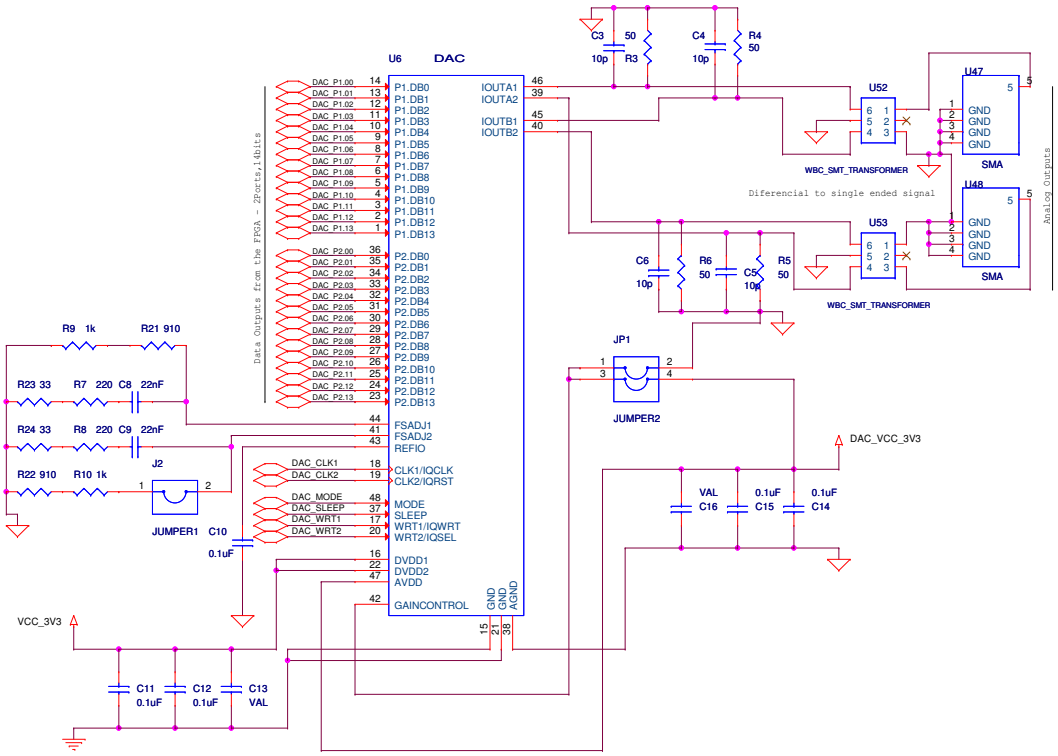


Figure A.3: DAC Schematic

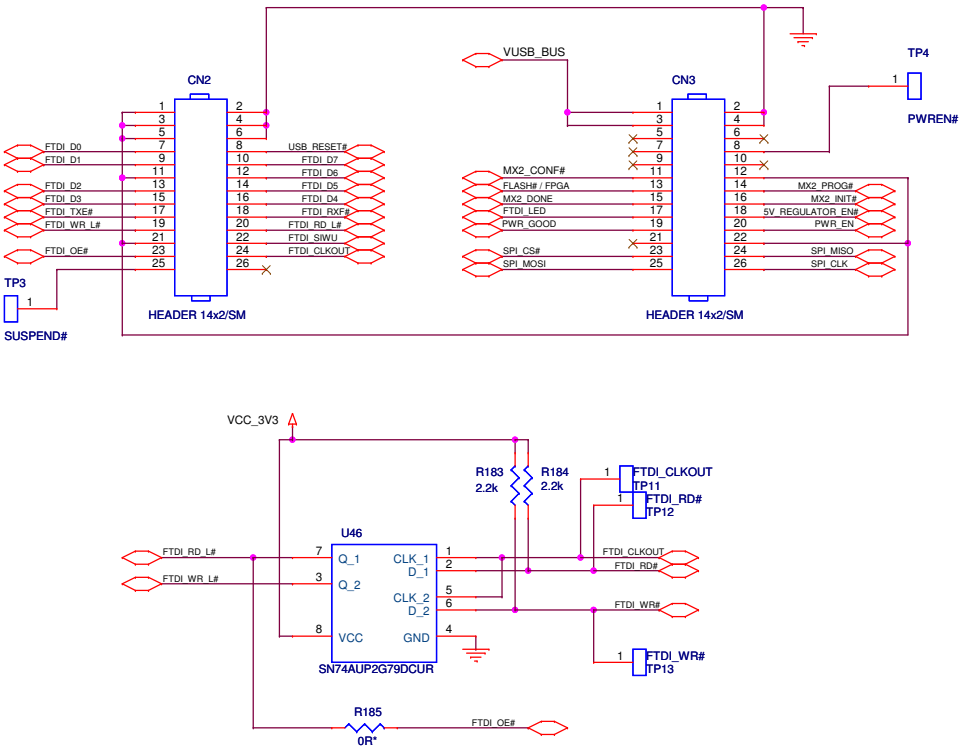


Figure A.4: USB Module Schematic

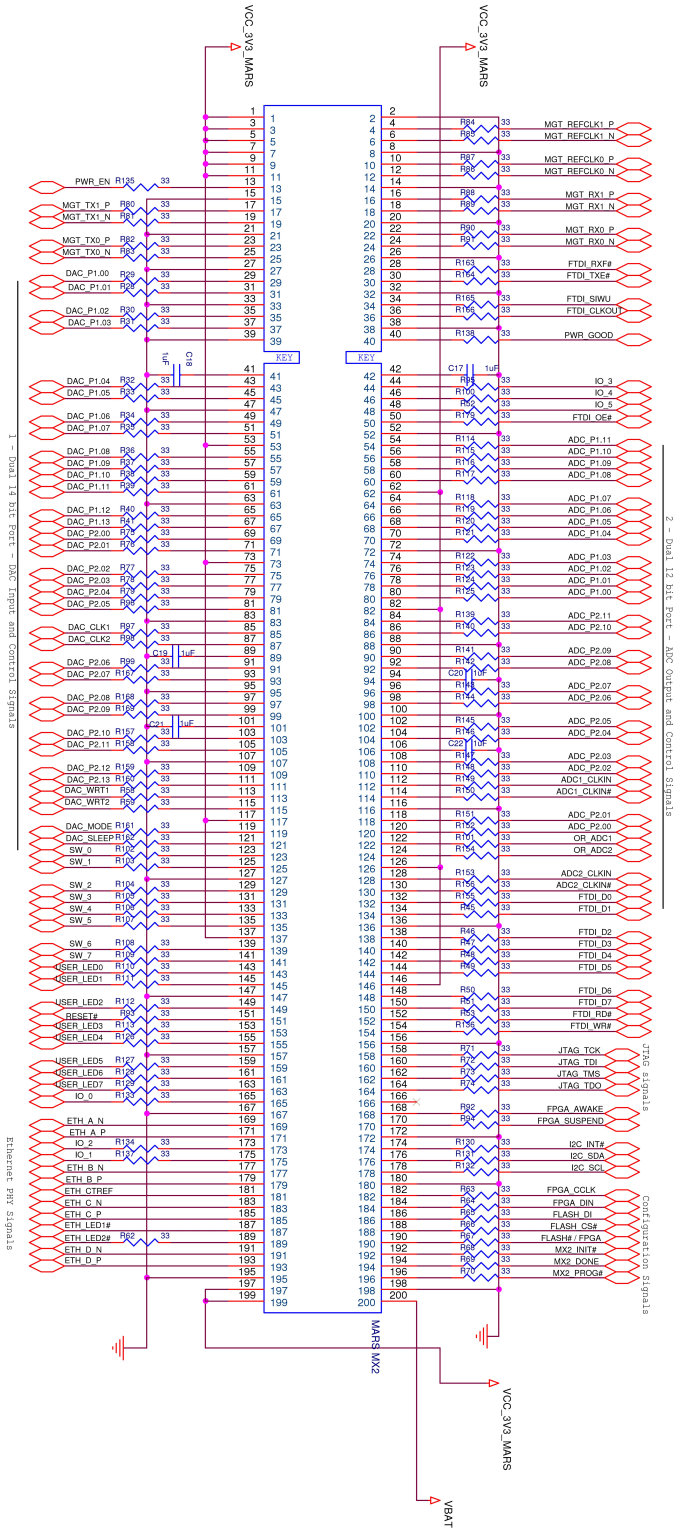


Figure A.5: Mars MX2 Connector Schematic

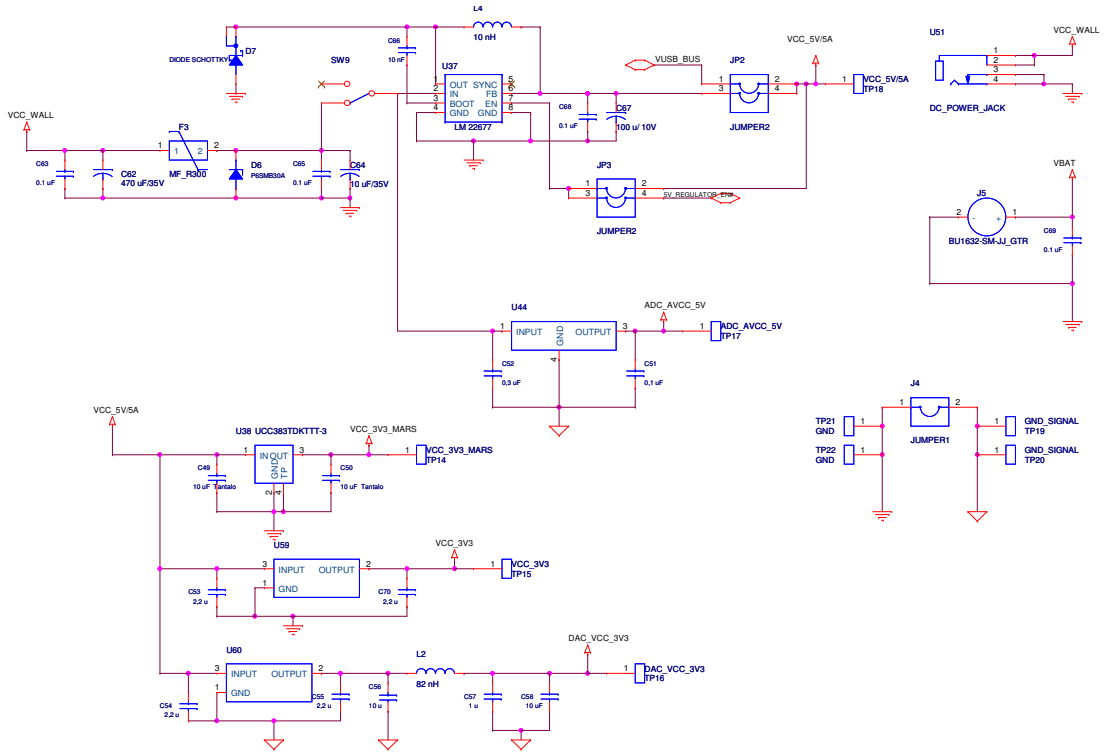


Figure A.6: Power Network Schematic

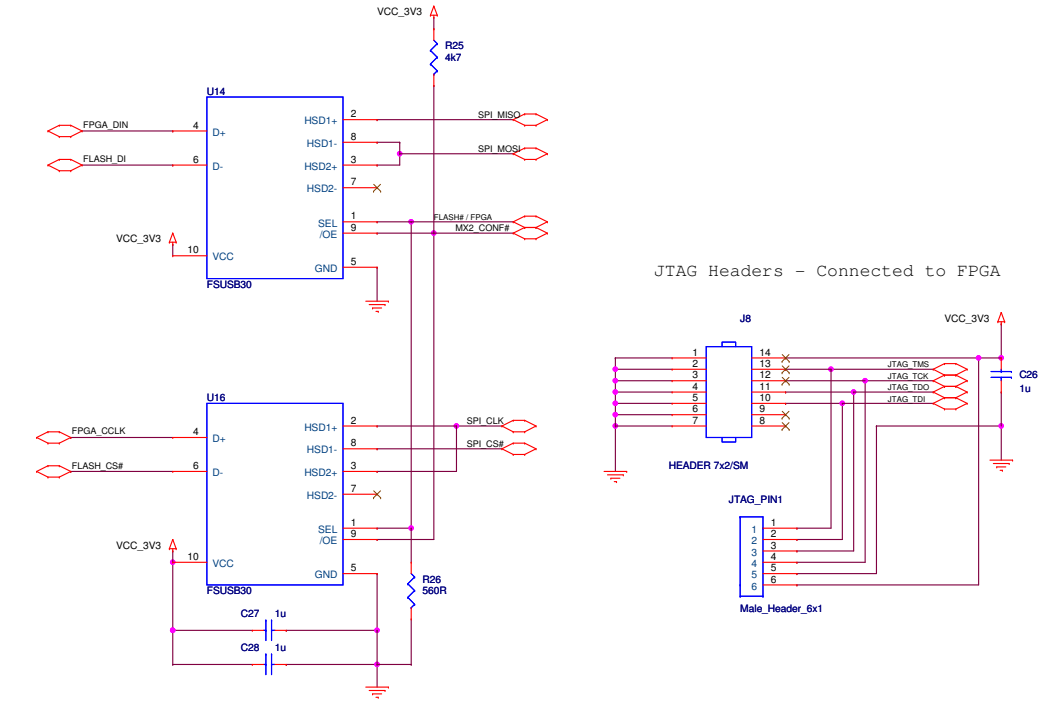


Figure A.7: USB and JTAG Programming Schematic

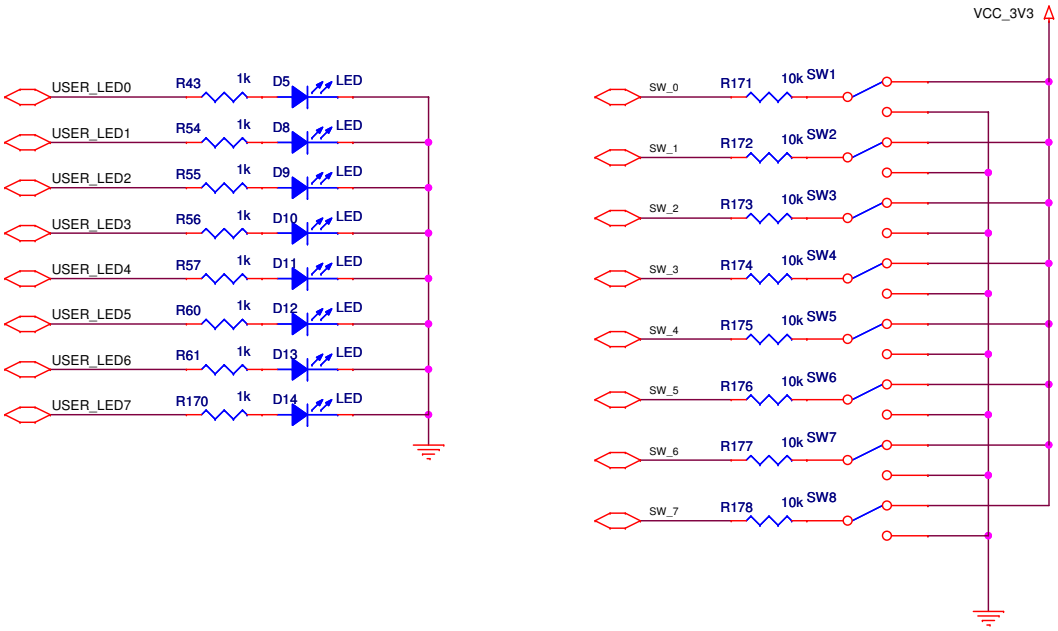


Figure A.8: Digital User Interfaces Schematic

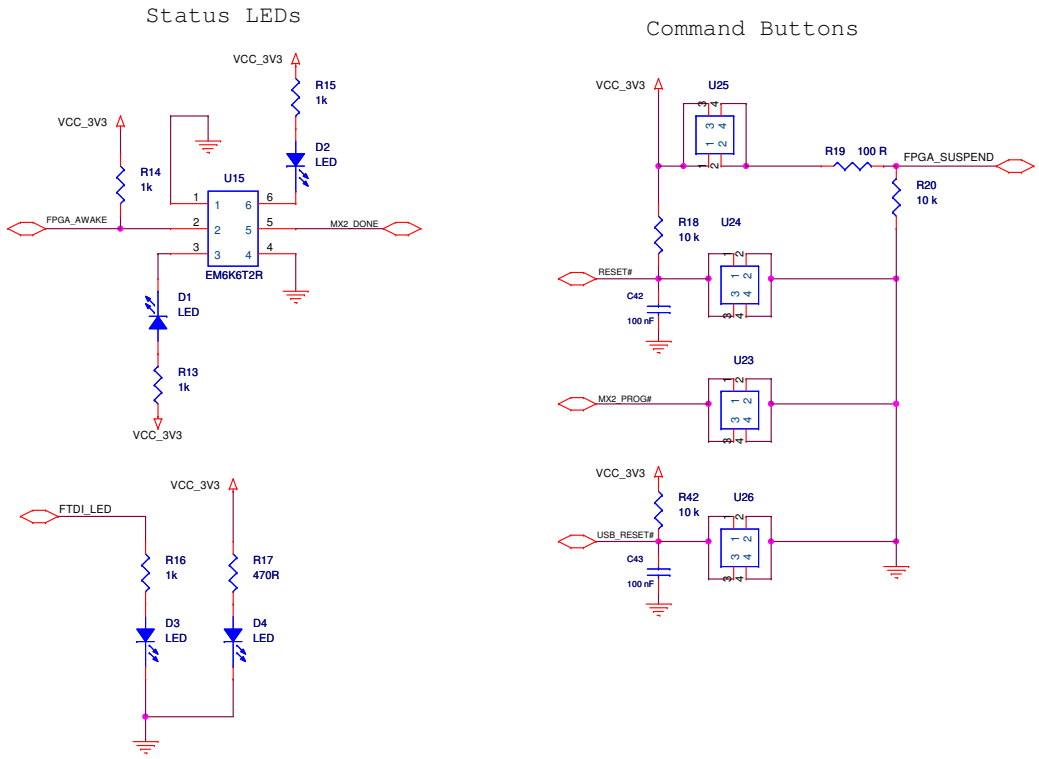


Figure A.9: Command and Status Interfaces Schematic

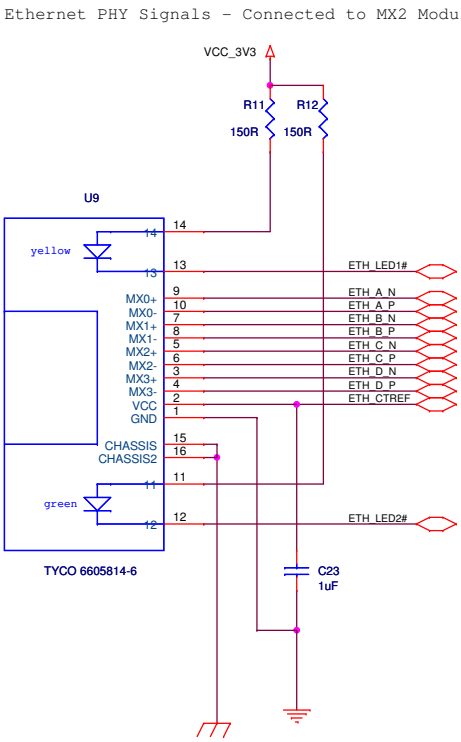


Figure A.10: Ethernet Schematic

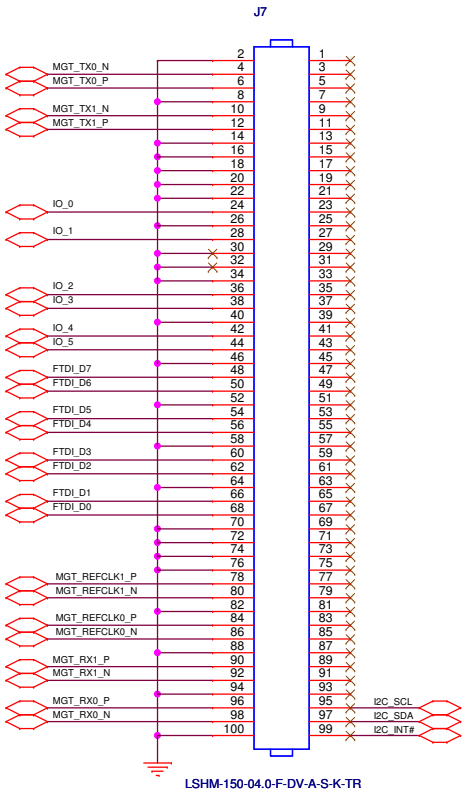


Figure A.11: High Speed Connector Schematic

Appendix B

PBC Layout

The PCB layout was designed using the OrCAD Layout software also from *Cadence systems*. The following images illustrate in detail the designed four layers.

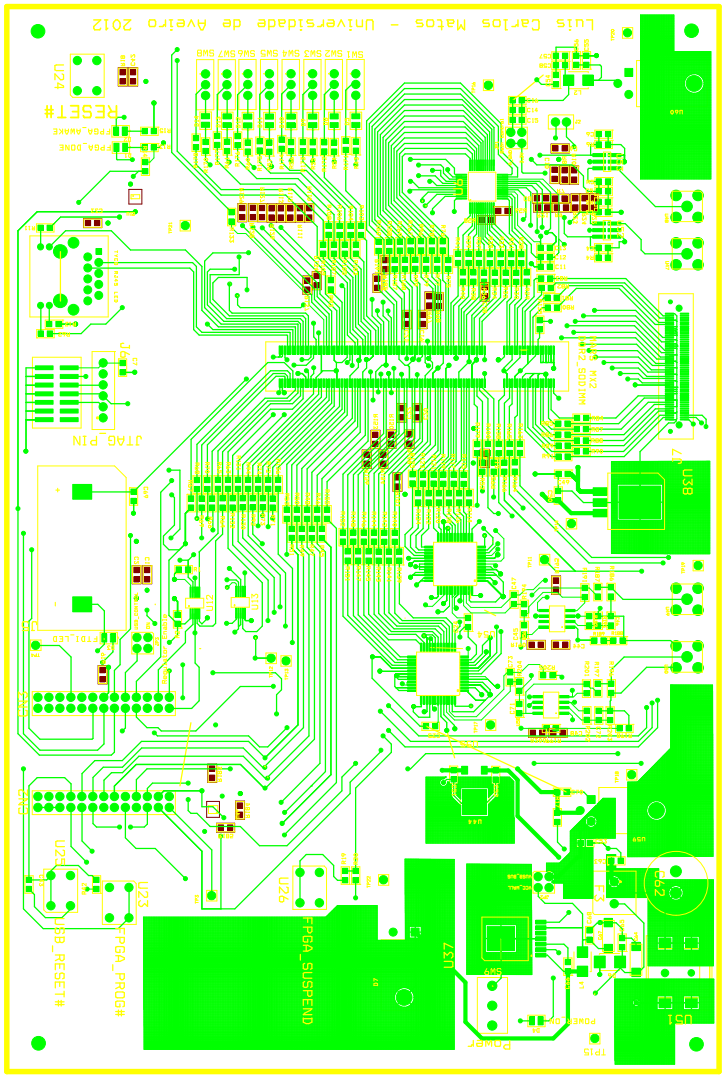


Figure B.1: Top Layer

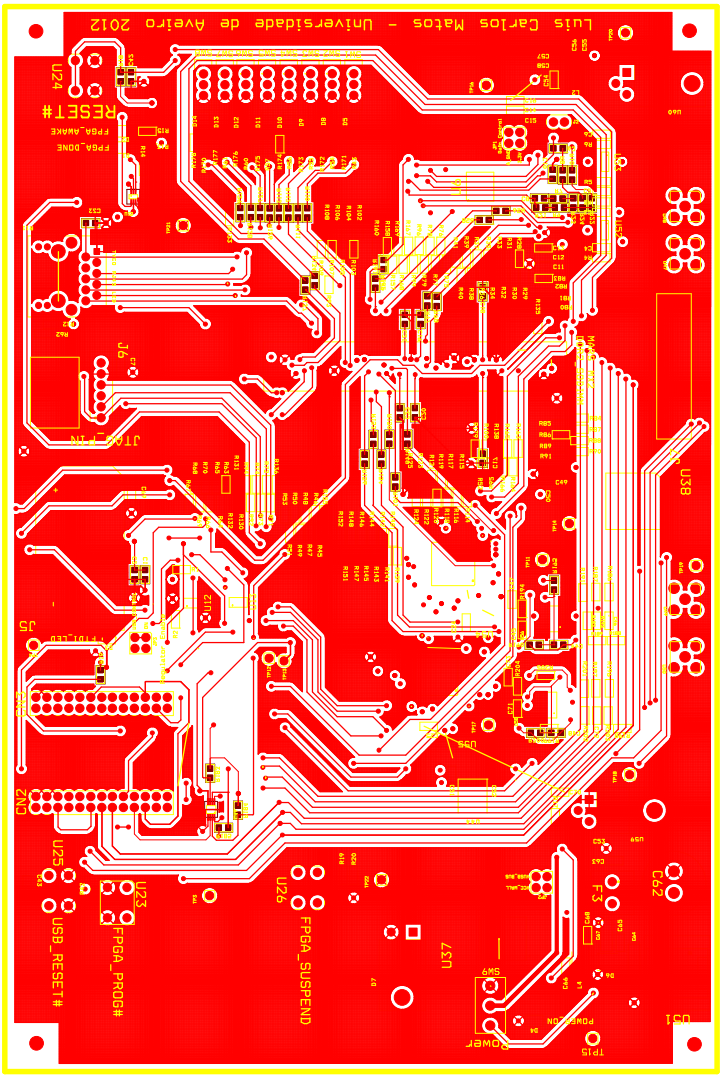


Figure B.2: Bottom Layer

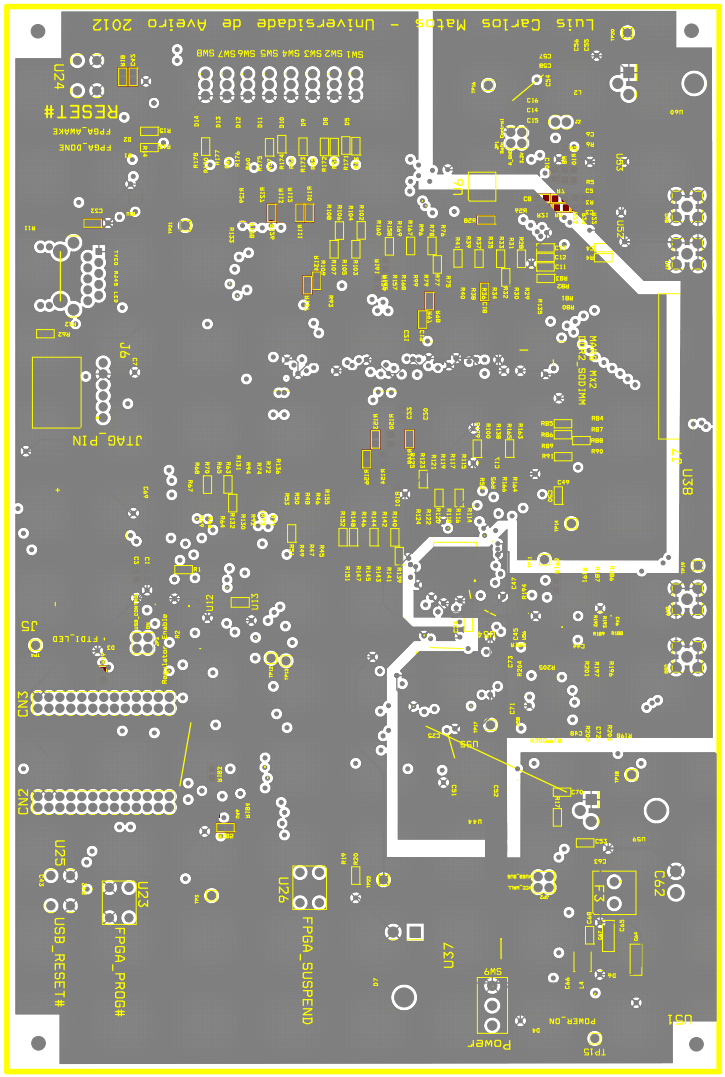


Figure B.3: Ground Layer

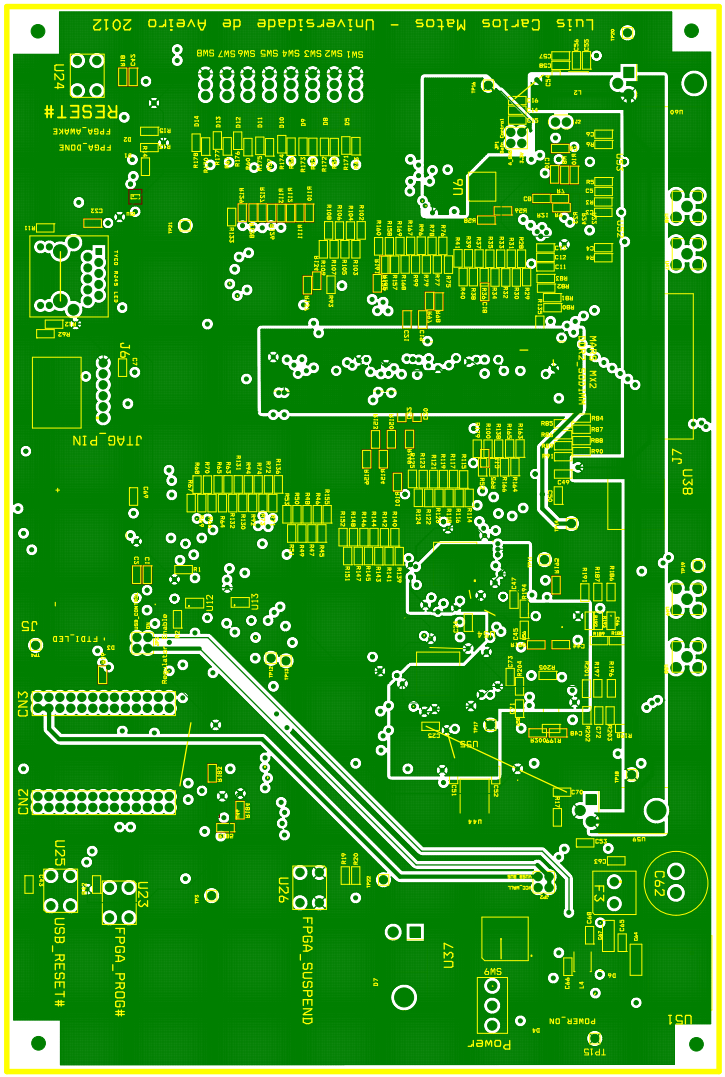


Figure B.4: Power Layer

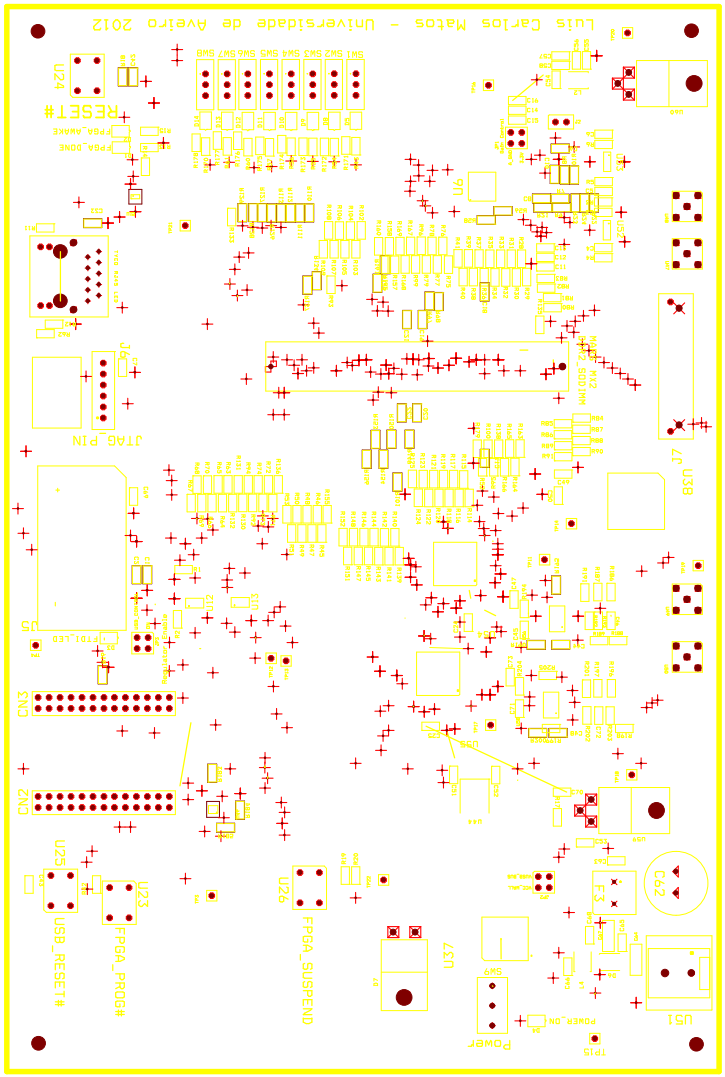


Figure B.5: Drill Chart