Development of an FPGA-based Controller PCB used in the Control System of a Modular Multiport Power Converter

By

Mariam Saeed Hazkial Gerges

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Energy Conversion and Power Systems
at the
UNIVERSIDAD DE OVIEDO
July 2015

Author……...

Certified by……..…….

Fernando Briz del Blanco
Full Professor
Thesis Supervisor

Development of an FPGA-based Controller PCB used in the Control System of a Modular Multiport Power Converter

by

Mariam Saeed Hazkial Gerges

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems
On July 17, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Energy Conversion and Power Systems

Abstract

The aim of this thesis is to develop a fully functioning Field Programmable Gate Array (FPGA-based) digital controller printed circuit board to be integrated in a Dual Active Bridge (DAB) module. It will serve as the slave control unit in a partially distributed control structure of a Solid State Transformer (SST). The SST topology consists of stacked Input-Series-Output-Parallel connected modules; each module is composed of a half-bridge cell accompanied by an isolation stage formed by a DAB module.

The work is outlined by four ideas, first understanding the addressed SST topology to get an idea about the slave controller requirements and optimize its design, second, developing the FPGA-based controller PCB design and commissioning the manufactured prototype, third developing the analogue to digital converter needed to interface this digital controller with the controlled system and finally general understanding of the implemented communication link between this developed slave unit and the central control.

Thesis Supervisor: Fernando Briz del Blanco
Title: Full Professor

Acknowledgments

I am very grateful to my Supervisor Prof. Fernando Briz. I am, in every sense, honored to have him as my Master's Thesis supervisor. Never sparing his valuable time from my help, makes me in debt of his dedication. His utmost trust in my decisions gave me unbelievable confidence and interest in my work. Especially, I would like to thank him for giving me this project experience, for opening many interesting opportunities in front of me and helping improve my skills daily.

Most importantly, I would like to thank my Mother and my Father. I believe that without them, none of this would have been possible. Their unbelievable love and support made miracles during these two years. Their faith in me gave me strength each day.

I attribute most of my success to my twin sister Sarah, for she is always beside me, encouraging, inspiring and sharing the best and the worst moments with me.

I would like to thank my fiancé, Guirguis for his care, for giving meaning to moments in my life and for the happiness he brings about.

I would like to express my gratitude to all the Professors of the Master who saved no effort in passing me their knowledge and helping me in many aspects. I would like to thank Prof. Jorge Garcia for being my mentor throughout the two years of the Master, his respect and appreciation has been inspiring.

I am also thankful to my colleagues especially Dr. Alberto Rodríguez for providing valuable advice.

Contents

List of Figures

List of Tables

Glossary

SST	Solid-State-Transformer
DAB	Dual-Active-Bridge
PCB	Printed Circuit Board
SiC	Silicon Carbide
SPS	Single-Phase-Shift
FPGA	Field-Programmable-Gate-Array
DSP	Digital Signal Processor
CPLD	Complex Programmable Logic Device
ASIC	Application-Specific Integrated Circuit
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
PROM	Programmable Read-Only Memory
JTAG	Joint Test Action Group
TCK	Test Clock
TMS	Test Mode Select
TDI	Test Data Input
TDO	Test Data Output
I/O	Input / Output
CCLK	Configuration Clock
DIMM	Dual In-line Memory Module
LED	Light Emitting Diode
ISOP	Input-Series-Output-Parallel

DC	Direct Current
AC	Alternating Current
TQFP	Thin-Quad-Flat Package

Chapter One

1. Introduction

This chapter is intended to give a general view over the project surroundings and the theoretical context to which it belongs. The aim is to deliver a clear understanding of the motive of the project and to provide a detailed description of the project itself as well as its role in the overall system.

1.1. Project Background

Recently, Power Electronic Converters have become the heart of power systems, due to their powerful advantages especially in converting and storing electrical energy, which are crucial requirement for nowadays rising needs related to the adoption of the distributed generation and the Microgrids.

Power electronic converters utilize semiconductor devices to transform or convert the electrical energy to the required format. They realize the conversion at any of the stages of generating, transmitting or distributing the electrical energy [1]. Their applications are getting wider constantly and thus their research field is widening. Improvements in power electronic converters are in two major fields: topologies, which addresses the main components of the power converter and the way they are arranged to provide the desired function or behavior. This field also addresses the field of control of the power converters finding more adequate, more accurate and faster control strategies. While the other field is power devices, which addresses improvements in their characteristics resulting in better behavior of the converter, e.g: using Silicon Carbide (SiC) devices.

This project belongs to the field of control enhancement of power electronic converters. The importance of studying the control strategies and their adequate implementation are discussed in the next subsection.

1.2. Motivation

The motive to this project is the importance of the converter control scheme and its implementation technology, which is one of the needs of the big project.

The importance is centered around the following advantages; applying an adequate control scheme can increase the functionality of the system and enhance the performance in a significant way. Also, precisely choosing the controller implementation technology and utilizing the suitable platform can result in very accurate system behavior with the desired response to control actions.

If the adequate scheme along with the suitable controller implementation were achievable, then full controllability over almost all the converter parameters is achievable as well.

The aim of this project is to theoretically study the control needs of a specific power electronic converter topology and develop an adequate controller platform for its implementation. More specifically, the work done resulted in developing a Field Programmable Gate Array (FPGA-based) high frequency printed circuit board (PCB) used to control a single-phase-shift-controlled (SPS) dual active bridge converter (DAB). The DAB is an isolated bidirectional power flow converter where the power flow is basically controlled by controlling the phase shift between the primary and the secondary Bridges.

The FPGA PCB is linked with a central control unit through a communication ring using TosNet protocol.

The FPGA PCB is the building unit of a larger control scheme composed of several units of the developed FPGA PCB and one central control unit forming a ring-like structure. It is also for that reason that the FPGA PCB will be often called Slave unit or Slave FPGA. This is later explained in more detail.

This FPGA-based PCB is intended to be function-tailored; it has no general purpose components (i.e LEDS, switches, push-buttons,...etc) like its functionally similar commercial PCBs, but only includes the programmable and the protection components. However, it can be regarded as a general controller because enough I/Os are exited from the FPGA to the PCB output interface, which makes the PCB flexible to be used in many potential applications. Also the design of the PCB is made as simple as possible, so that any modification can be easily implemented as well as tested.

Based on the importance of applying appropriate control implementation technology; time should be invested in the selection of the control platform/device, e.g: FPGA, DSP, CPLD, ASIC,..etc. Thus, the selection of the FPGA for controlling the DAB is addressed with details in Chapter Two.

1.3. Project description

This Master thesis project was part of a European project sponsored by the European Commission in the 7th Framework Programme; the title of the project is ‘’**Silicon Carbide Power Electronics Technology for Energy Efficient Devices**’’ or the **SPEED** Project, more information is available on the SPEED website [2].

The project objectives are to employ high voltage SiC devices with higher performances compared to existing Silicon technology in high voltage power electronic converter topologies.

Among the consortium partners of the project are; University of Oviedo, in collaboration with another university (Hannover), one large company (Ingeteam) and one SME (INAEL). Main responsibilities of University of Oviedo are designing and fabrication of Power Cells according to the peculiarities of SiC devices, and development of new SST topologies.

Based on these requirements, the project was defined to be divided into two stages;

First stage is building a prototype which consists of a low voltage MMC-based multiport power converter integrated with a DAB with serialized input-parallelized output (ISOP)topology providing functionalities of a Solid-State-Transformer (SST).To achieve this, 24 stacked DC/DC converter modules are going to be built; one module is composed of a half bridge connected to a DAB through a DC-link, the parallel output of this ISOP DC/DC converter is connected to a DC/AC rectifier (voltage source rectifier, VSR). This allows connection of two AC systems, one high-voltage and one low-voltage, including galvanic isolation. Further information can be found in[3] and [4].

Second stage is building a cell for the final high voltage prototype, which will consist of a FB-DAB module. The core of the DAB will be a high frequency transformer providing high galvanic isolation (around 24KV) between the primary and the secondary bridges of the DAB. Thus many isolation issues should be taken into consideration and using optical fibers. Power modules are intended to be made of SiC rather than Silicon modules using an enhanced sandwich structure (developed by Nottingham university), thus avoiding the many problems caused by the wire-bonds in the regular power module.

The work done in this Thesis is focused on the first stage of the project (the 24 stacked modules), and specifically in the DAB converter control hardware implementation and the implemented partially distributed control scheme.

The DAB converter preliminary version was already built and its control strategy was implemented as part of another work[1] and was explained in [5] and [6].However, the implementation of its control is done using a commercial field programmable gate array (FPGA) board; the Basys2 spartan-3E from Digilent Company, which is a third party FPGA Board.

[1] All the work related to the Dual Active bridge (DAB) topology implementation and control discussed in this thesis is attributed to Alberto Rodríguez Alonso.

Taking into consideration the modularity requirements and later the selected control technique; each previously mentioned module of the 24 should include a separate FPGA PCB responsible for the control, this arises the thesis objectives;

First, developing an FPGA function-tailored board to decrease the size and the cost of the control system, as well as to avoid dependence on third parties.

Second, implementing communication between converter modules, through making a communication link; linking all the FPGA PCBs with the central control unit, in order to accurately achieve the desired functionalities (i.e power transfer commanded to each DAB module).

The schematic is shown in Figure 1. This simplified model shows one module (i.e HB-DAB) and specifies the location of the FPGA PCB which will be integrated into the DAB converter module. The communication ring linking the slave FPGA with the central control unit is also shown.

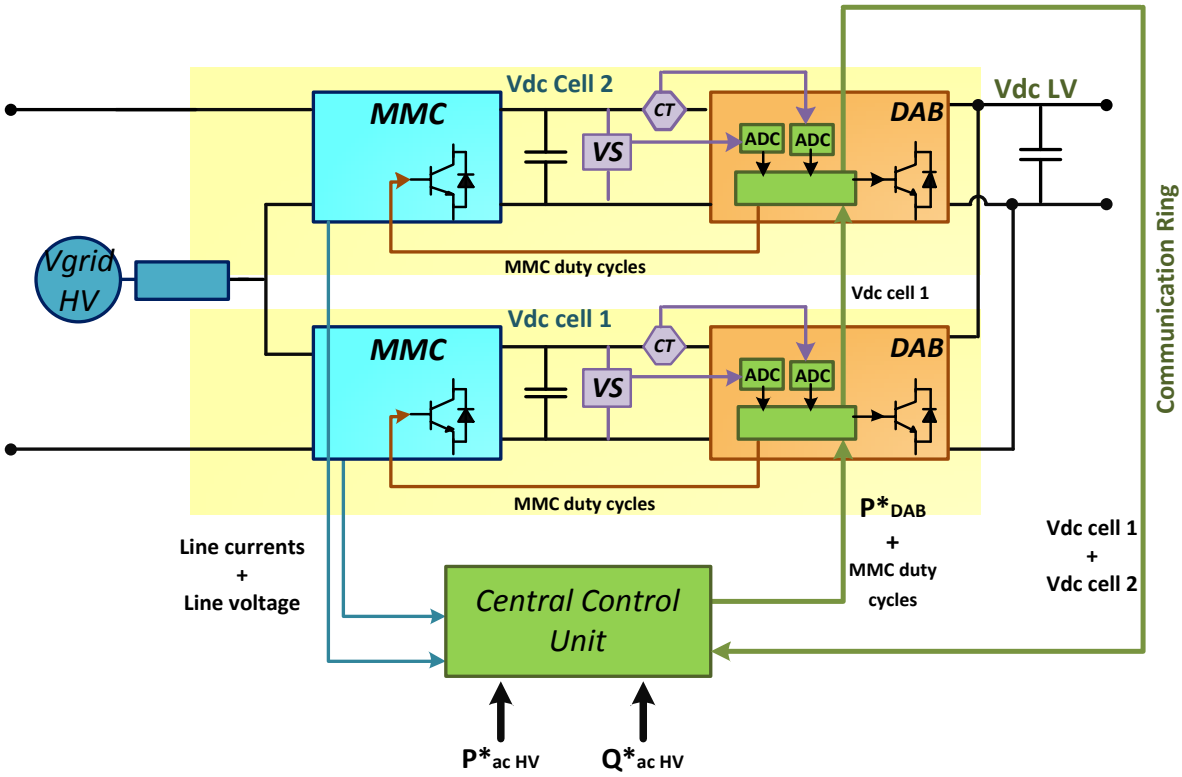


Figure 1: An overall schematic of the addressed modular multiport converter topology.

1.4. Skeleton of the document

This document is divided as follows;

Chapter One, is an introduction to give a warm up about the subject and the background behind the work done.

Chapter Two, is a theoretical discussion mainly indicating the aim of the project and understanding some basics needed for the flow of the technical material.

Chapter Three, deals in detail with the developed FPGA PCB starting from a simple understanding of FPGA technology and discussing its detailed role in the control scheme indicating the selected components to start developing the Design.

Chapter Four, deals with the design procedure for developing the slave FPGA controller followed by developing a theoretical debugging procedure to configure and test the developed PCB, then three integrating subjects are discussed; the analogue to digital converters as well as the MMC adaptation circuit and finally the communication link between the slave FPGAs and the central control, all these together serving to fit the slave units into the control system. In the end, the commissioning process of the manufactured board is discussed in details.

Chapter Five, summarizes the project outcomes and conclusions and discusses the future work to be done in integration with the developed project.

Chapter Two

2. Theoretical background

This chapter is intended to give a basic knowledge about the background of the Field programmable gate arrays (FPGAs). First, performing a general review on the types of controller implementations used in a power electronic converter and summarizing the main differences between them. Then focusing on the Digital controller option and stating the most used digital technologies for computational devices nowadays. Finally, with a little more detail the DSP solution is compared with the FPGA solution, stating clearly the main differences on which a selection choice can be made. Also, indicating the bases, on which the selection of the controller for this specific project is made.

2.1. Control of Power electronic converters

As discussed before in the introduction Chapter; the control of a power electronic converter is very important for achieving the required functionalities with acceptable performance, as well as protection of the converter itself and the subsequent system. All these targets are bound to the adequate selection of the control scheme, as well as the adequate selection of the controller implementation technology or in other words the device that will be responsible for performing the computations needed by the control strategy to result in a certain control action fed to the controlled system.

Power electronic converters are advancing enabling new functionalities to appear. However the usability of these new functionalities and new performance standards strongly depends on the effectiveness of the control implementation. Appropriately choosing the electronic control platform and the computing devices, results directly in improvement of the converter performance and a possible reduction in software development time. It also has a direct impact on reducing the hardware components or reducing the computational time [7].

Therefore, it is important to dedicate time to understand the literature background about the possible controller implementations.

2.2. Controller implementation

For electronic power-converter applications, the choice of the most suitable technology to implement the controller is not always evident, thus, this section gives a general overview on the possible options for implementing the controller of a power electronic converter. It provides some possible guidelines for the selection of the implementation technology of the controller.

2.2.1. Analog Vs Digital

There are two major types of control implementation; analog and digital. Analog implementation mainly consists of designed circuits of analog components such as operational amplifiers. On the other side, digital implementation consists of digital processors computing the data in binary form.

Each type has its own applications where it would be mostly feasible, however in order to be able to choose between them, it is needed to have a close look at the main differences between them. These lie in cost, hardware complexity, computational time, accuracy, reliability, capability and consumed power [8].

To simplify the points of interest; a comparison between both types of control is summarized in Table 1.

Table 1: Comparison between analog and digital Control implementations.

Points of comparison	Analog	Digital
Accuracy	Less accurate control results due to possible variation of control parameters because of noise imposed on most of the analog signals.	More accurate control implementation due to less noise effect because of binary representation of all the data [9].
Computational time	Takes time to compute events due to possible delay in the signals.	Digital processing and parallelism increase the speed of instructions computing.
Capability	Cannot perform complex control computations and more over cannot easily modify the control strategy or add more instructions without modifying the hardware.	Programming facilitates any level of control complexity and reprogramming facilitates any change in the control without touching the hardware.
Hardware Complexity	Several hardware components are needed to perform the control.	Hardware components are replaced by code [10].
Reliability	Less reliable due to the more analog components included in the system [11].	More reliable due to the less number of components needed and the possibility of decentralizing the control/ using several processors.
Cost	Components are cheap but as the system gets bigger the overall system is more expensive.	For complex control schemes the overall system is cheaper.

Consumed power	Higher voltage is needed to power the analog devices.	Voltage needed to operate digital processors is typically low.
Resolution	Continuous, thus resolution is maximum.	Discrete thus introduces inferior operating properties due to delays in sampling [9].

Table 1, could give a simple clue on the choice of the controller implementation in a controlled power electronic converter.

In the case of a cheap power converter where accuracy is not a crucial control requirement and the control scheme is very simple, analog implementation makes an adequate choice. However analog control loses its feasibility as the system control gets more complicated and accuracy and speed are essential requirements for achieving the needed function from the converter. At this point using analog control implementation makes no sense while digital processor can provide all the needed functionality with almost the same price or even cheaper for the overall control system.

2.2.2. Digital processor options

If the choice was to use a digitalized control implementation for the sake of accuracy and speed, then a tricky question would arise; which is the most adequate processor for the task? The answer is simply, it depends.

The choice depends on two main things; first the available digital processor technologies, and second, the task to be done, specifically, the complexity of the desired control scheme, the processing speed needed and the number of inputs and outputs needed for applying the control to the system.

The most used technologies of digital computation devices nowadays are as follows;

1- Microprocessor / microcontroller (MCU): when easy and fast implementation is needed.
2- Digital Signal Processor (DSP)
3- Field-Programmable Gate Array (FPGA)
4- Complex Programmable Logic Device (CPLD): when "instant-on", fast and wide decoding, low idle power consumption, and design security are important [50].
5- System-on-Chip (SoC): includes memory, ADC, DAC, microprocessor, voltage regulator, oscillators...etc.
6- Application-Specific Integrated Circuit (ASIC): fixed-function ASICs typically take months to fabricate and cost hundreds of thousands to millions of dollars to

obtain the first device, this is a tradeoff for optimizing area, performance speed and power consumption.

Generally, if the converter control is very complicated in terms of control parameters relations and required computational instructions then a DSP is very convenient. An example is the control of a three-phase DC/AC power converters, that require an efficient control for the switching devices that, in many occasions, can be very complex due to system structure. In the case of a moderate complexity control scheme with the need of many inputs and outputs, then an FPGA is a good choice. If there is the need of multiple inputs and outputs but the control scheme is rather complicated, then probably a SoC (System-on-Chip) is an adequate option to benefit from the advantage of the DSP and the FPGA. If the converter control requires general-purpose functionality and programming simplicity, a microcontroller is the more appropriate choice.

The most commonly used processors in the control implementation of power electronic converters are the DSP and the FPGA. This is due to their extremely appealing performance, flexibility and ease of us [12]. Thus, a brief comparison between the DSP and the FPGA is discussed in the next subsection.

2.2.3. DSP Vs FPGA control implementation choice

The DSP and the FPGA are the most popular digital processors in the implementation of the control algorithms of power electronic converters because of their efficient design, relatively low cost, stand-alone and real-time characteristics. However, the DSP has been the most commonly one for a long time. However, this was before higher processing capabilities (MIPS) are demanded by new applications [13].Thus it is important to compare between the FPGA and the DSP because in specific applications the FPGA may be a better choice.

The desired result after reading this subsection is having a fair idea about the main differences between the FPGA and the DSP, in order to have a clear selection regarding different converter control implementations.

The main differences between both are; the required I/Os, the complexity of the control structure, the required sampling rate/computational time/processing capabilities, implementation time and the price;

A. Complexity of control structure

If the control structure is complicated from the point of view of the required computations, then the DSP would be ahead of the FPGA. This is because it is software-based, processor programmed in a C environment. Thus, enabling any control complexity to be implemented via coding. While the FPGA is digital hardware-based device that

consists of arrays of logic blocks such as registers, gates, multipliers, memory blocks, and even CPUs. The programming in the FPGA is focused on configuring the logic blocks. All computations are carried out by interconnecting logical blocks together to perform the needed function. So the achievable complexity depends on the available logic blocks and the possible interconnections [14].

B. Computational time

The FPGA is clock based, so every clock cycle has the potential ability to perform a mathematical operation on the incoming data stream [15], and thus having a sequential and concurrent capabilities [16]. So, the FPGA has a high degree of parallelism on the device.

On the other hand, DSPs are instruction based, not clock based. May be, three to four instructions are required for a mathematical operation on a single sample. The data must first be captured at the input, then forwarded to the processing core, cycled through that core for each operation and then released through the output. Thus, although the frequency of a DSP can be much higher than that of the FPGA but it only has sequential capability (i.e computational tasks should be done in series) [15].

C. Cost

The price of the FPGA is more expensive than the DSP, this is a tradeoff for the capabilities it provides more than the DSP. In fact, in some occasions, it becomes unfair to compare between both due to the vast capabilities and speed provided by a FPGA.

D. Number of inputs and outputs of the control scheme

Due to the array-structure of the FPGA which provide it with the parallel characteristics, its computational speed is not affected when adding more inputs and outputs to the control scheme that needs more computations to be done. As long as there are unused logic blocks, the FPGA can process more computations with no effect on speed, while the DSP can't, so instructions will be prioritized. Consequently, implementing multiple I/O control schemes using FPGA enhances the overall system response and behavior significantly.

E. Implementation difficulty

Regarding programming, FPGAs are usually programmed with a bitstream, which configures the array of logic cells to perform the desired function and connect multiple logic cells as determined by the control architecture. FPGA programming technologies range from one-time programmable elements to electrically erasable devices. However,

the languages of FPGAs; e.g: Verilog and VHDL (Very High Speed Integrated Circuit Hardware Description Language)**,** are somehow complex. HDL is a high-level language, writing an HDL design description is a relatively complicated process, since it describes circuit functionality at the register transfer level (RTL) [17]. Thus, it takes significantly more time to program an FPGA, while the DSP can be programmed easily using very common C-languages.

Based on the previous facts about both, it is concluded that, if the desired control system has a low sampling rate of around few kilohertz and the needed inputs and outputs are very few (i.e single-channel implementation), then a DSP can be an obviously good choice. However, if the sampling rate is more than 1 MHz and the control structure requires accessing many inputs and outputs, then the FPGA starts to attract the attention because of its parallel structure; at high data rates the DSP struggles to capture, process and output the data without any loss. This is because of the many shared resources, buses and even the core within the processor. On the other hand, the FPGA can dedicate resources to each of these functions. However, once the control structure is becoming quite complex, then a DSP is the suitable choice.

2.3. Project-specific controller selection

Having an overview on the different types of control implementation and their advantages and disadvantages, it is time to decide the adequate controller platform for the implementation of the specific control structure requirements previously discussed.

In order to make a selection, the previously stated key requirements affecting the choice should be accurately identified;

1- Number of inputs and outputs processed by the control structure;

The control task of this controller is to control the DAB. However, meanwhile it should receive information about the duties of the MMC from the central control unit and then generate and send the relevant PWM signals to the MMC.

The inputs are;

- One current measurement from the DAB.
- One communication ring receiver data line.
- Two error signals from the DAB primary and secondary bridges.
- One measurement of the MMC DC cell voltage.

Thus the task needs five inputs.

The outputs should be;

- One PWM signal sent to the HB switches.

- One enable signal to the HB switches.
- Two control signals sent to the DAB primary side bridge drivers, and another two to secondary side bridge drivers.
- One clock signal and one chip-select signal sent to the ADCs
- One communication ring transmitter line.

Thus, nine outputs should result from the control structure.

Adding the number of inputs and outputs; a total of 14 I/Os are needed per controller which favors the selection of the FPGA.

2- Complexity of the control desired computations;

The DAB is intended to be controlled using Simple Phase Shift control (SPS).In SPS 50% duty cycle is used and the only control variable is the phase shift between the primary and secondary bridges of the DAB. However nonlinear the control relations may be, the computations needed will not require special processor skills. Also, the generation of the PWM signals of the MMC is rather simple because the control of the MMC is processed in the central control unit then only the duty cycles are sent to the slave controller.

Therefore, the simplicity of the control computations favors the selection of the FPGA over the DSP as well.

3- Previously done work;

Another consideration is that the control structure of the DAB was already previously done using FPGA. Thus the description was already written in VHDL and implemented on a commercial PCB and tested on the DAB. This saves the time needed to program all the desired control structure in C to use a DSP. Moreover, the communication protocol was implemented using VHDL.

In this case, optimization of performance and reducing computational time oblige to favor designing an FPGA-based controller. This is to reduce cost and processing time and introduce more flexibility, because the DSP will perform the processing in a sequential method which will make the processing significantly slower than the FPGA.

Chapter Three

3. Slave FPGA

This chapter discusses in details the procedure of developing the slave FPGA PCB. First briefly understanding the FPGA technology and providing some PCB design rules related with the FPGA design. Next the function of this FPGA-based controller PCB, as a part of the whole controlled system, is explained. Also, clarifying the reason for developing the PCB rather than using the commercial one. Then, the configuration (i.e programming) of the FPGA is explained to have an idea of the final procedure and take into account any possible design consideration. This includes clearly knowing the chain of connected programmable components and their specific connection, which implies certain configurations. Later the important design specifications are outlined, which could help systemize the design to some extent, as well as opening the way to seeing the challenges facing these specifications and trying to find solutions to them. Finally, a theoretical test procedure consisting of some logic steps is designed in an attempt to have a systematic testing scheme for debugging the problems in the PCB once manufactured. (i.e Hardware development/manufacturing).

3.1. Brief understanding of FPGA technology and PCB design

3.1.1. FPGA technology

FPGA is similar to a matrix of gate arrays (logic blocks), including Flip-flops, multiplexers, look-up tables (LUT) that can be programmed in the field (after manufacturing). This is shown in Figure 2.

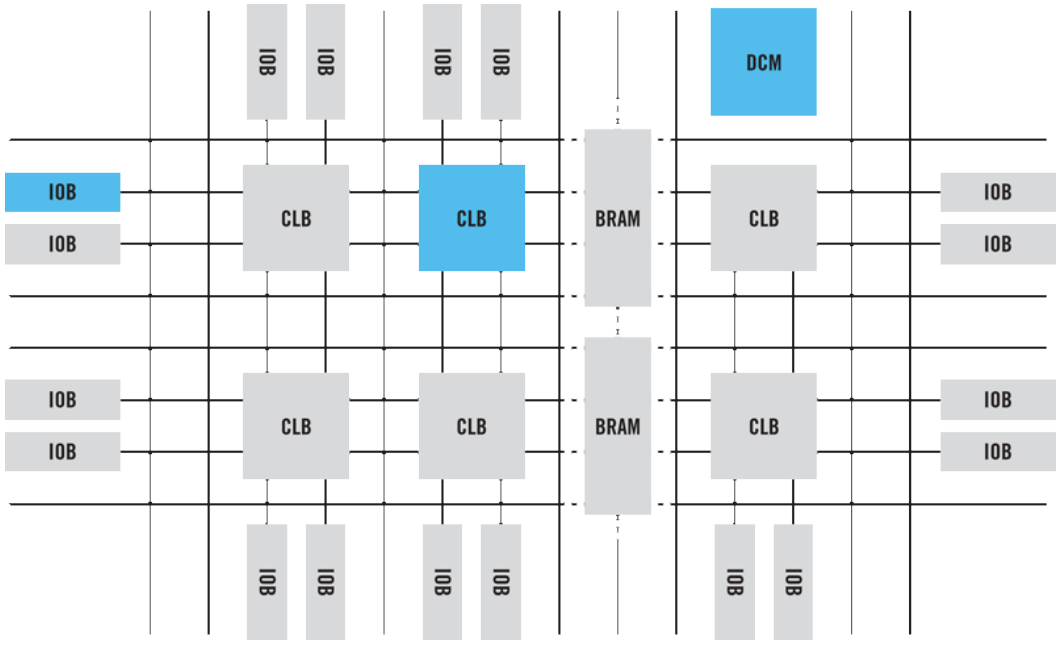


Figure 2: The FPGA internal structure [18].

CLB is Configurable Logic Block which consists of a switch matrix, DCM is Digital Clock Managers, BRAM is IP Processor Block RAM [49] and IOB is Input Output Block; I/O in FPGAs is grouped in banks [18].

These logic gates or blocks can be wired together in any way, thus constructing the hardware circuit performing the desired control task. Also as long as there are still unused gates in the device, other control tasks/computations can be configured without affecting the speed of the running control tasks.

In order to program an FPGA to do a certain control task, there are two options; either schematic entry or HDLs (Hardware Description languages). Schematics or graphical designs are very easy for small designs, but are unmanageable for a large/complex design, thus HDLs are becoming more popular. The most used HDLs are Verilog and VHDL.HDLs are high-level programming languages, similar to C-language. However, the challenge lies in the fact that the design is done describing a hardware implementation of the needed circuit that achieves the required control scheme. This is a very complex process and very different than a software design in the case of a DSP and also, they are textual languages being easy to read but hard to see the whole system as well as having strict descriptions [16].

The Language used here is the VHDL, Very high speed Hardware Description Language. It is a programming language for describing digital circuits. It gives all the potentials of designing the specifically needed processor structure/circuit which, when

applied to the FPGA, will perform the task with much higher speed than a regular microcontroller or a DSP.

The FPGA manufacturers that have most of the market share are Xilinx and Altera. Together they own around 90% of the FPGA market share in 2015 [19].

3.1.2. State-of-the-art in FPGA

Speaking about the state-of-the-art of the FPGA is in other words stating the latest technology applied to the FPGA manufacturing. However, the important to know is; what kind of "technology" is interesting here.

In the world of FPGAs, the most important characterizing features is the number of logic gates an FPGA contains. Since the silicon transistor is the building unit of the logic gate (block), thus the most important technology is related with the number of transistors that could be placed on a die. So excels in the FPGA field is related with increasing the number of transistors per area thus having a cheaper price per gate. Each milestone in the technology improvement is called a "*node*", defined as, allowing more semiconductor dies to be manufactured on the same piece of silicon wafer, resulting in less cost per product sold. This was governed by Moore's law, which states that; this reduction in price and increase in the chip density is at a rate of a factor of two yearly [20].

However, it is believed that the 28nm node technology is the last node in Moore's law; because it is a very mature technology, price increases when trying to increase the chip density more due to the complicated die technology which would cost even more. Thus wafer price increase washes away the scaling benefit [21].After the 28nm node, we can continue to make transistors smaller, but not cheaper [22].

Regarding the latest products available in market, for example, Xilinx is now offering a multi-node series called the *UltraScale*™ which includes, Virtex™ and Kintex™ FPGAs in 20nm and 16nm nodes, and also SoC and 3D IC devices. Xilinx claims that *UltraScale*™ provides 2-5X greater system level performance/watt over 28nm devices as well as higher level of security [23].

This Thesis work is more focused on High frequency FPGA-based digital controller PCB design and commissioning rather than implementing the desired control structure in VHDL; the configuration of the FPGA and the memory was done using configuration files generated from a previously prepared VHDL description code[2].

[2] All the VHDL files prepared for the control of the DAB as well as for the testing of the FPGA was developed as part of the DAB work.

3.1.3. FPGA-based PCB design

An FPGA-based high frequency PCB design is tricky because of many reasons, first is the FPGA package used; even the most hand-work friendly TQFP (Thin-Quad-Flat Package) packages has huge number of tiny pins and needs care during designing the layout. Second, an important consideration is the number of layers stack-up that are possible to contain the design. Third, attention is needed when routing high frequency signals and power signals in this PCBs due to possible cross-talks and electrical noise imposed on critical signals. Fourth, attention should be paid to the current limiting resistors and other pull-up resistors that can cause many problems if wrongly selected.

Some useful design guidelines are as follows;

1- Having a reference commercial PCB design that has available schematic documents. This helps to refer to the critical connections, the regulator circuit…etc. In this case the reference Board is the Basys2 Spartan-3E board from Digilent shown in Figure 3.
2- Avoiding difficult to route BGA packages whenever possible and using instead TQFP packages.
3- Accurately selecting the number of gates needed because some gate densities are not provided in all packages.
4- Determining if the design needs additional memory, basically a Platform Flash PROM is always needed to configure the FPGA on power up because FPGAs has a volatile configuration memory.
5- Defining the configuration method of the FPGA.
6- Following a guide for High frequency boards when routing the PCB, similar to [24].
7- Decreasing the number of layers when possible, even if in the first prototype, to be able to simplify the debugging procedure.

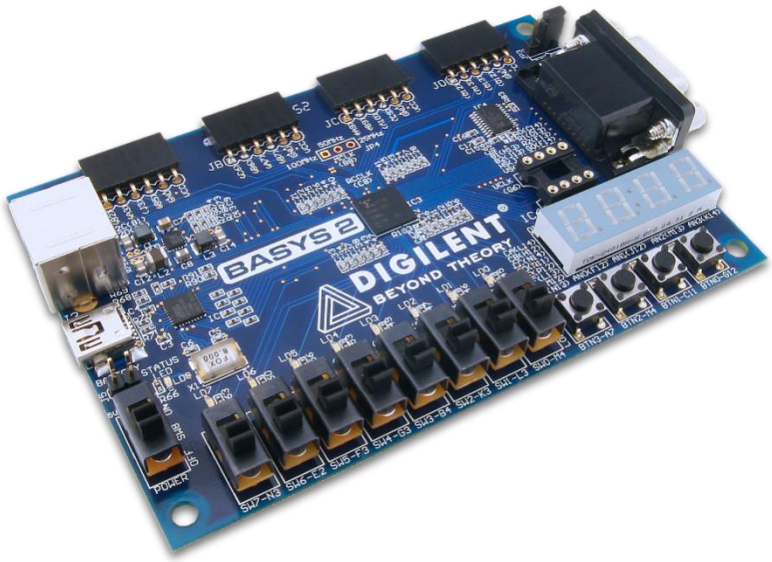


Figure 3: The reference design, Basys2 PCB.

3.2. Detailed function

The specific function of the slave FPGAs is related to the choice of implementing a partially distributed control scheme. In the partially distributed control, the control is shared between a central control unit in charge of the MMC control algorithms and slave FPGA units, one per each DAB module. The slave processing units' function is to process the DAB control and also to receive information about the PWM signals of the MMC from the central control via the communication ring and generate them and send them to the MMC.

The choice of implementing distributed control was due to the many advantages that it provides. In the partially distributed control choice, as shown in Figure 4-a.The central control unit is responsible for the MMC control algorithms while each of the slave FPGA processing units is in charge of one DAB and its corresponding MMC cell capacitor voltage.

On the other hand, in centralized control structure as shown in Figure 4-b, only one Master unit is responsible for processing the control algorithms of both the MMC cells and the DABs. This means that the central controller should be able to handle the data received from every voltage and current sensor in the system, using ADCs and then generating all the gate signals sent to the drivers in the MMC and in the DAB in order to track the commanded references by the control scheme. Consequently, this control structure requires a high processing capability in the master unit. Also, significant large number of input and output ports is needed to handle all the control input parameters and output signals, most of them should be connected via optical fiber links, to maintain electrical isolation [3].

The two control schemes are shown in Figure 4.

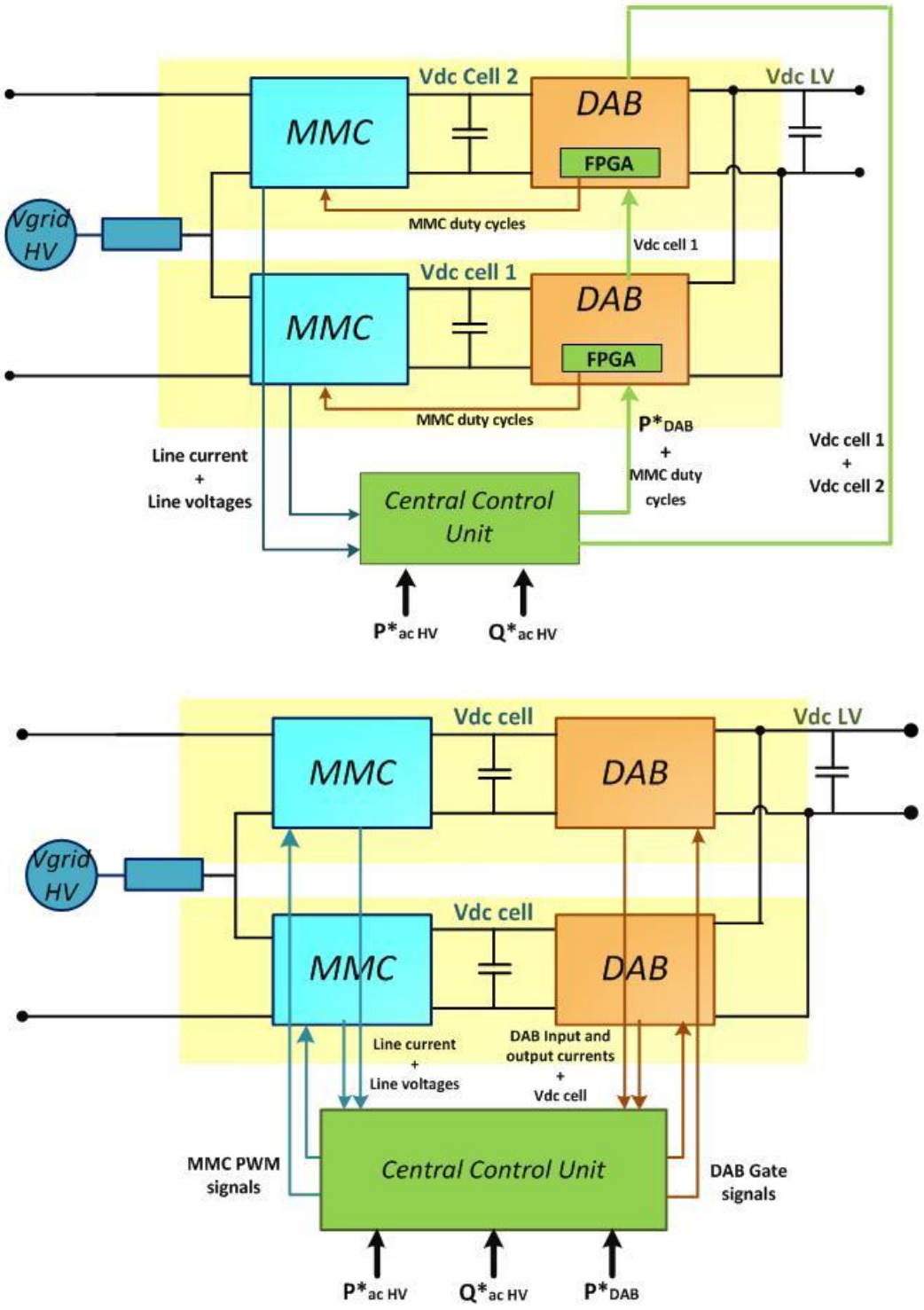


Figure 4: (a) upper, partially distributed control structure. (b) Lower, centralized control structure.

The advantages of a partially distributed control scheme can be summarized as follows [3];

1) Decrease the information sent to the central control needing to be processed, thus the computational specifications of the central unit is significantly reduced.
2) Decreasing the electrical isolation requirements related with the signals sent to the drivers and the signals received from the sensors in each DAB module by reducing the number of optical fibers needed and reducing the length of the ones needed since the FPGA is mounted on the DAB module using a DIMM socket.
3) Speeding up the control process by using the FPGA in the first place and also sharing the load between them and the SoC central unit.
4) Employing to a significant extent the idea of modularity, in which each module has its controller.

The implementation of a partially distributed control structure imposes some new requirements to be present in the system hardware structure. These are mainly two; the slave controllers and a communication link to integrate the control system by linking the central unit with the slaves units.

Thus, the slave FPGA will be mounted on the DAB module to perform two main functions;

First, its primary function is to take the sensed measurements (i.e input current sensor measurement) after converting them into digital values using A/D modules and process this data. Then generates the gate signal commands and sends them to the isolated gate drivers in order to control the switches in the two bridges if the DAB. This control is aimed to track a certain power command required from each DAB module. This power command is sent by the central control unit, through the communication link, to each slave FPGA in order to result in transferring a specific amount of total power from all the DAB modules.

Second, the Slave FPGA has also another function; which is sending the data of the measured cell capacitor voltages at the output of the MMC cell or the input of the DAB cell (Vdc_{cell}) to the central control unit through the same communication link. This data is needed by the central control, in order to balance the MMC cell capacitors voltages, the result of the processed control is generating the adequate duty cycles needed for the MMC switches. Due to the distributed control idea, these duties are sent to the FPGA via the communication ring, which already exists, then each FPGA generates the corresponding PWM signals and sends them to the corresponding MMC cell. Since each FPGA is responsible for one MMC cell and all FPGAs are connected via the same communication ring, thus this simplifies the control scheme.

In conclusion, the role of the FPGA PCB in the addressed modular multiport converter control system can be summarized briefly as follows,

1- Processing the DAB control and generating the signals driving its switches.
2- Sending the MMC cell voltage to the central control through the communication ring.
3- Sending the PWM signal to the MMC.

A detailed schematic of one MMC cell and its DAB module with the addressed topology is shown in Figure 5 [3].

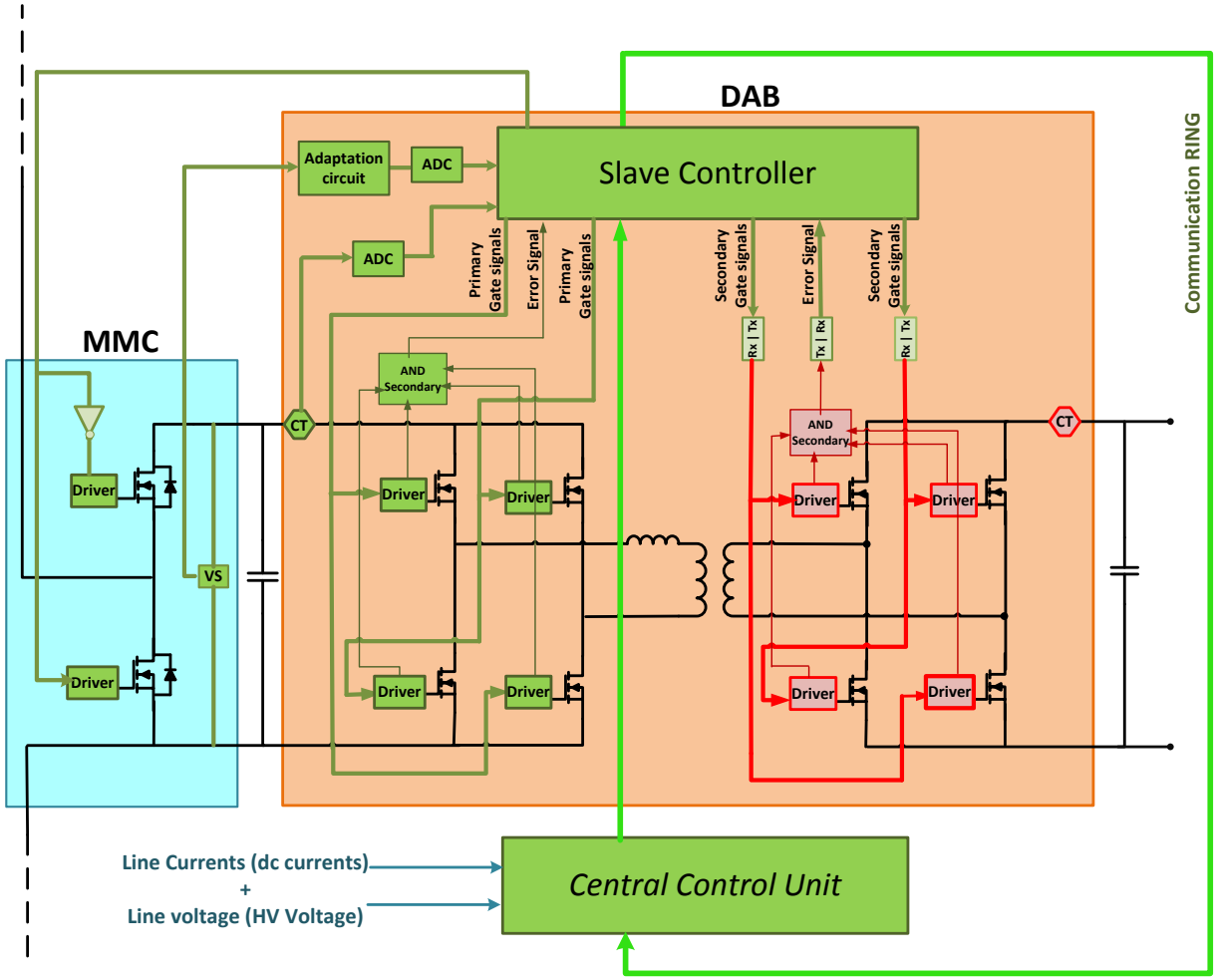


Figure 5: A detailed scheme of one MMC-DAB module.

It is clear that for the partial distributed control a communication ring is needed to communicate the data of all the modules with the central control. Through this link the central unit will send to the slave devices, the commands for the power that each DABs has to transfer, as well as the corresponding duties for the MMC cells. And the slave units

will send to the central unit through this link the measured MMC cell capacitor voltages. This communication link is discussed in Chapter Four with more details.

3.3. The reason for developing the slave FPGAs

As stated in the previous subsection; the slave FPGA is important for implementing the distributed control strategy, however the important question now is whether it is worthy to design and implement a PCB that is similar to a commercial one to use it or is it better to buy and use the commercial (i.eBasys2 Kit).

The answer to this depends on the specific needs of the SPEED project. Taking into consideration the needs of the primary prototype; building 24 stacked MMC-DAB modules, then the most significant requirements should be the necessity of modularity, low overall control system cost and the implementation of a distributed control. This drives the idea of developing a PCB just tailored to fit the task without any general purpose components, just including the programmable processor with its supporting ROM memory. Thus, the PCB will certainly have a customized and thus optimized size (low consumable area), along with an adequate cost relative to the application.

Moreover, it is preferable in long term projects to avoid using commercial and third party products due to the existing possibility of obsoletion or stopping of the production of this product by the manufacturer. This implies reaching an obligatory end-of-life for the whole project or having to invest more resources in finding alternatives later, which makes the proposed topology less modular and less robust.

Consequently, the best decision is to spend some time and resources in order to design, implement and commission a PCB that does the same function as the commercial one due to the importance of modularity, robustness and eventual lower cost.

The main function of the slave PCB; which is the control of the DAB module, was already tested using the commercial PCB. Since it is working, the commercial PCB will serve as the reference design for an FPGA slave digital controller board that will be design and developed.

3.4. Configuration of the FPGA

The function of a Spartan-3E FPGA is defined by loading application-specific configuration data into the FPGA's internal reprogrammable CMOS configuration latches (CCLs), similar to the way a microprocessor's function is defined by its application program [25].

So, configuration is the process for loading configuration data into the FPGA, it should take place on power up or when demanded [26].

However, the FPGA configuration memory is volatile thus Configuration data is stored in a PROM or other external data source. Consequently, an important issue to think about is; how will the FPGA be configured or programmed. This is a very critical question from the point of view of the PCB design and the selected components.

According to the Datasheet of the chosen Spartan-3E FPGA family, theSpartan-3E FPGAs offer several configuration options in order to decrease the impact of the configuration process on the overall system design. Basically, there are two main configuration structures; serial configuration where one data line is needed and Parallel configuration where 8-bit or 16-bit data line are used, but there are three configuration modes [25];

First, the FPGA generates a clock and loads itself from an external memory source, either serially or via a byte-wide data path. These are called "Master modes" [27].

Second, an external host such as a microprocessor downloads the FPGA's configuration data using a simple synchronous serial interface (SPI) or via a byte-wide peripheral-style interface (BPI). These are Called "Slave modes".

Third, designs having multiple FPGAs share a single configuration memory source, creating a structure called a daisy chain.

There are three MODE pins in the FPGA; M2, M1, and M0.These pins select the desired configuration mode. After the FPGA completes configuration, the mode pins are available as user-defined I/Os.

The commercial Basys2 PCB, originally used to test the DAB operation, uses the second configuration mode. Thus, requiring a microprocessor to download the configuration data in the FPGA. Then using the mode pins and a simple jumper, it loads the configuration data from a memory after repowering the PCB. The Basys2 uses a serial configuration structure called the *Master serial* configuration.

The configuration of the FPGA in the developed controller in this thesis will be done using the same Master serial configuration structure but using the first configuration mode implementing it using a JTAG configuration, unlike the Bays2 which uses a microprocessor, the reason for that is discussed in the Design challenges/issues section. The significant configuration pins of an FPGA are summarized in the Table 2.

Table 2: the FPGA configuration pins [25].

Configuration pins	name	description
MODE pins	Mode Select	Selects the FPGA configuration mode.
PROGRAM_B	Program FPGA	Input that initiates configuration. Active low and must be High during configuration to allow configuration to start.
CCLK	Configuration Clock	Generated by FPGA internal oscillator.
INIT_B	Initialization Indicator	Open-drain bi-directional pin. Active Low and Error and power stabilization flag.
DONE	FPGA Configuration Done	Open-drain bi-directional pin. Indicates completion of configuration process. Low during configuration. Goes High when FPGA successfully completes configuration.
DIN	Serial data input	Serial input for configuration data.
DOUT	Serial data output	Output to the next device in a daisy chain.

3.4.1. JTAG configuration mode

The JTAG configuration is used for two main aims; first is to configure the FPGA in a fast way and be able to debug the design and second, is to complete the chain of the Master serial configuration and be able to configure the platform Flash PROM which will always load the configuration files on power up of the FPGA.

There are dedicated JTAG pins in the FPGA JTAG port; TCK, TMS, TDO, and TDI. The TCK is the clock and it is driven by the dedicated download cable. The bitstream is generated on the computer and is downloaded via the ISE™ software iMPACT™ utility and a Xilinx programming cable [26].

3.4.2. Master Serial configuration structure

The aim of using this configuration mode is to configure a platform flash PROM to be able to load the configuration data into the FPGA on each power up. This is because the FPGA memory is volatile and is erased on each power off event.

The FPGA will be programmed using a JTAG, thus externally configuring the FPGA via a PC dedicated software (i.e iMPACT™).Then a platform flash memory is configured via the same JTAG and loaded with the FPGA configuration files. On start-up or power-on, the FPGA loads its configuration files from this memory. This is done using the MODE pins of the FPGA, setting the mode pins to HIGH allows configuring the FPGA via a PC (JTAG), setting the mode pins to LOW allows the memory to load the configuration files into the FPGA after the FPGA sends to it a configuration clock. This technique is called "*The Master Serial configuration mode*" [25].

In order to implement this configuration mode, a certain connection or a chain should be achieved between the FPGA module, the PROM module and the JTAG.

Figure 6 shows the diagram of connection related with the intended configuration procedure.

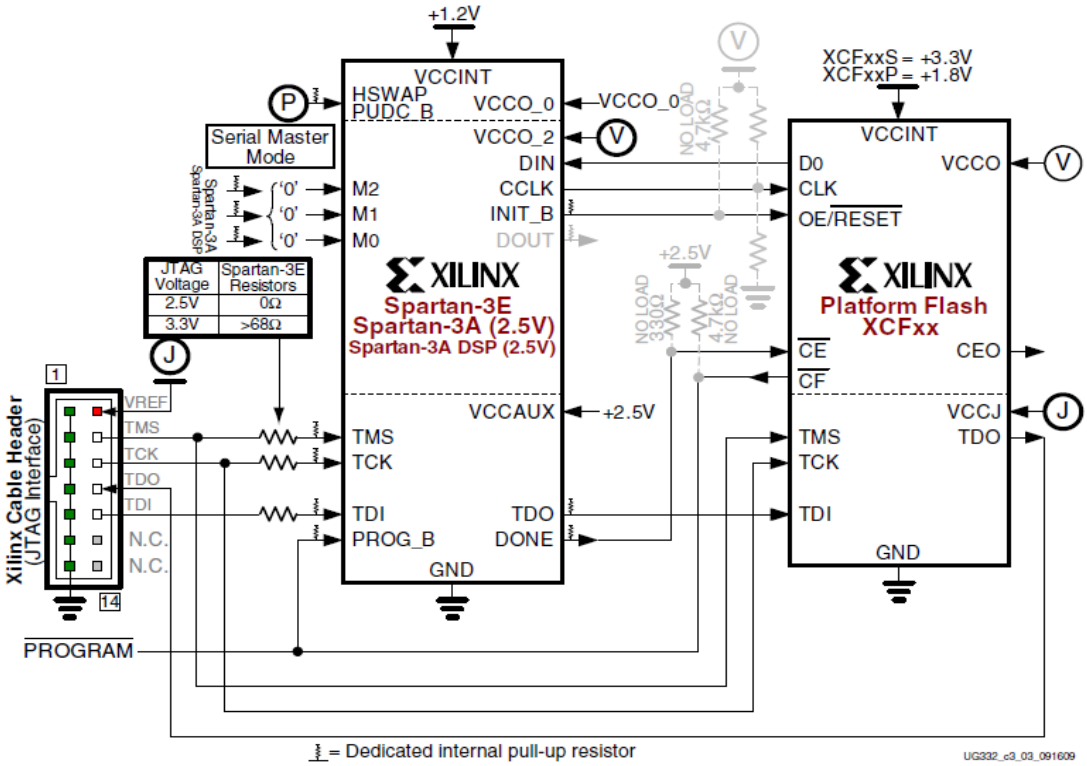


Figure 6: the Master Serial configuration structure [27].

The Spartan-3E FPGA configures itself from the attached Xilinx Platform Flash PROM, as illustrated in Figure 6. The mode pins are driven low and the FPGA supplies

the CCLK output clock from its internal oscillator to the attached Platform Flash PROM. In response, the Platform Flash PROM supplies bit-serial data to the FPGA's DIN input, and the FPGA accepts this data on each rising CCLK edge [25], it is a slow but an easy to debug configuration mode [26].

More information about this configuration structure is explained in user guide ug332, P.79 [27].

3.4.3. Configuration Process

The configuration is done using a dedicated programmer **"Platform Cable USB"** provided by Xilinx as well. This type of programmers can configure any type of Xilinx devices, the most relevant here are; Spartan® FPGA families and Platform Flash XCF00S/XCF00P/XL PROMs.

Platform Cable USB is a high-performance download cable attaching to user hardware, through a 14-conductor ribbon cable designed for high-bandwidth data transfers, for the purpose of programming or configuring FPGAs and PROMs. Device configuration and programming operations using Platform Cable USB are supported by iMPACT™ download software using Boundary-Scan (IEEE 1149.1 / IEEE 1532), slave-serial mode, or serial peripheral interface (SPI) with target clock speeds selectable from 750 kHz to 24 MHz [28]. A picture of the Platform USB Cable II (DLC9G) is shown in Figure 7.

Figure 7: The Platform USB Cable II [28].

Thus, the configuration process is as follows;

First, generating a (.bit) configuration file from the needed application program file, this file is used to configure the FPGA. For doing that, ISE™ software from Xilinx is used.

Second, generating a (.mcs) configuration file needed to program the memory; this file is generated from the previously generated (.bit) file using iMPACT™ software.

Third, connecting the USB cable to the PC and the other side to the Platform Cable USB, then waiting for the STATUS LED to turn to Amber color indicating that the cable is

correctly installed and recognized by the computer, then it should be configured inside the iMPACT using Cable Setup option.

Fourth, the ribbon cable should be connected to the FPGA PCB through the JTAG connector according to the scheme shown in Figure 8. If it senses a voltage between 1.5V and 5V on the Vref pin of the JTAG connector, the STATUS LED will turn to Green color.

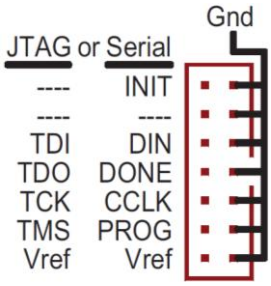


Figure 8: The JTAG connector pin labels [28].

Fifth, if the LED is green thus a target device is detected, then initialize chain process should begin resulting in defining the target system connection or configuration mode, consequently, if this process was successful, the .bit file could be downloaded into the FPGA and the ***.mcs*** file could be downloaded in the memory.

Sixth, the ribbon cable should be disconnected and the PCB power should be turned OFF, then the mode selecting jumper should be adjusted to the other position (configuration through memory), finally powering on the PCB and verifying that the program is loaded correctly from the memory to the FPGA, by measuring the output signals using an oscilloscope.

3.5. Components included in the slave FPGA PCB:

The slave FPGA has certain functionality components that are critical for its proper operation, this section lists with some details the significant components needed to successfully develop a working FPGA PCB with the required functions needed for this specific project, but it can also be used as a general digital FPGA-based controller to be used in many applications.

The main board components are as follows;

3.5.1. Digital computational device/ Processor / FPGA:

This section addresses the selection of the FPGA family specifying with details the significant advantages driving this decision and giving an idea about the different packages provided by this family indicating the main differences between them and how

to choose the suitable package for a design. Also this section specifies the procedure done for selecting the number of suitable logic gates for a certain design, thus resulting in choosing the minimum logic gates density for the intended FPGA PCB functionality to achievable with no problems.

3.5.1.1. FPGA family selection

This is the brain of the slave PCB. This is chosen to be a Field-Programmable Gate Array (FPGA) because of the necessity of having multiple input/output signals which is always an advantage in the FPGA over the DSP as explained previously.

The FPGA family used is the Spartan-3E family from Xilinx, the reason for that is that this family of FPGAs from Xilinx is characterized by relatively low cost, high-performance and wide range of densities of system logic gates. This family is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications [25]. Also, another important advantage is that this family is still on production and its end-of-life or its expected obsoletion is not intended in the coming years (at least till 2020) [29], due to its good performance and maintained success in sales.

Not only, as indicated above that this family provides wide range of logic gates; 100K, 250K, 500K, 1200K and 1600K gates, but also it provides most of them in the same package and having the same Pin-out, which is a very important advantage in this application, because when extending the function of the FPGA in the controlled system, thus increasing the input and output signals and consequently increasing the data processed by the FPGA, there will be the need of more logic gates, which would impose using a higher density FPGA module, and if this new module has the same package and the same pin-out, replacing it can be done directly in the PCB without the need to change the design and manufacture the board again.

Also, the Spartan-3E family of FPGAs are available in both quad-flat pack (QFP) and ball grid array (BGA) packaging options; specifically it has eight low-cost production packages, thus having variety of options during the PCB design; the commercial Basys2 PCB design is done in 4 layers using a Chip-Scale Package (CSP) BGA package, the CP132 [30]. However the intended design will be done using the Thin Quad Flat Pack (TQFP) package, TQ144 which reduces the complexity of the design and the number of layers needed for the PCB.

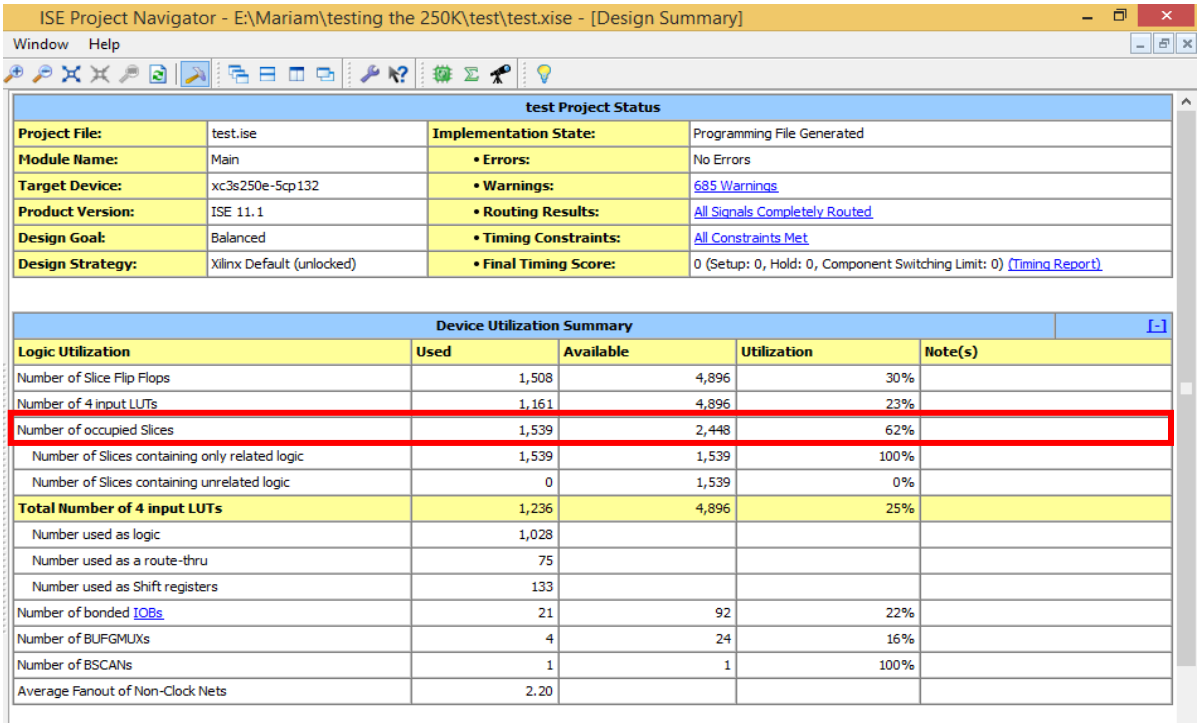
However, caution should be taken because not all densities are available in all packages, thus before designing the PCB, special attention should be given to accurately choose the most adequate package and the most adequate density of logic gates for the intended application [25].

3.5.1.2. Selection of Density of Logic Gates

Regarding the selection of the adequate density of logic gates, this is a very important step and should be done before starting the development of the PCB. This was done by linking the communication protocol files implemented in VHDL with the control of the DAB files and adding to that the PWM signals needed to be sent to the MMC; the idea is to have the expected total number of input and output signals required for the FPGA PCB to correctly perform its role in the system in order to be able to know the percent of occupied flip flop slices which would determine the number of gates needed.

A 100K FPGA means it has a 100,000 logic gates. The maximum available logic gates density in a single FPGA in the market are provided from Xilinx in the Virtex-Ultrascale™, XCVU440, which has 20 Billion transistor count [47],[31]. However, the available logic gates density in the chosen Spartan-3E family are 100K, 250K, 500K, 1200K and 1600K.

At first, in order to test, a 250K is tested because it has good price per gate, thus, a project is created on the ISE™ software choosing for that a **250K** Spartan-3E FPGA in order to test, and then after linking the files, ***Generate programming file*** option is used to test if it was possible to use the 250K or not, the file was successfully generated and the report generated indicated that the percentage used of the device is around **62%**, as shown in Figure 9, thus the selected FPGA can be the 250K gates Spartan-3E FPGA.


ISE Project Navigator - E:\Mariam\testing the 250K\test\test.xise - [Design Summary]

Window Help

test Project Status			
Project File:	test.ise	**Implementation State:**	Programming File Generated
Module Name:	Main	**• Errors:**	No Errors
Target Device:	xc3s250e-5cp132	**• Warnings:**	685 Warnings
Product Version:	ISE 11.1	**• Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**• Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	0 (Setup: 0, Hold: 0, Component Switching Limit: 0) (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,508	4,896	30%	
Number of 4 input LUTs	1,161	4,896	23%	
Number of occupied Slices	1,539	2,448	62%	
Number of Slices containing only related logic	1,539	1,539	100%	
Number of Slices containing unrelated logic	0	1,539	0%	
Total Number of 4 input LUTs	1,236	4,896	25%	
Number used as logic	1,028			
Number used as a route-thru	75			
Number used as Shift registers	133			
Number of bonded IOBs	21	92	22%	
Number of BUFGMUXs	4	24	16%	
Number of BSCANs	1	1	100%	
Average Fanout of Non-Clock Nets	2.20			

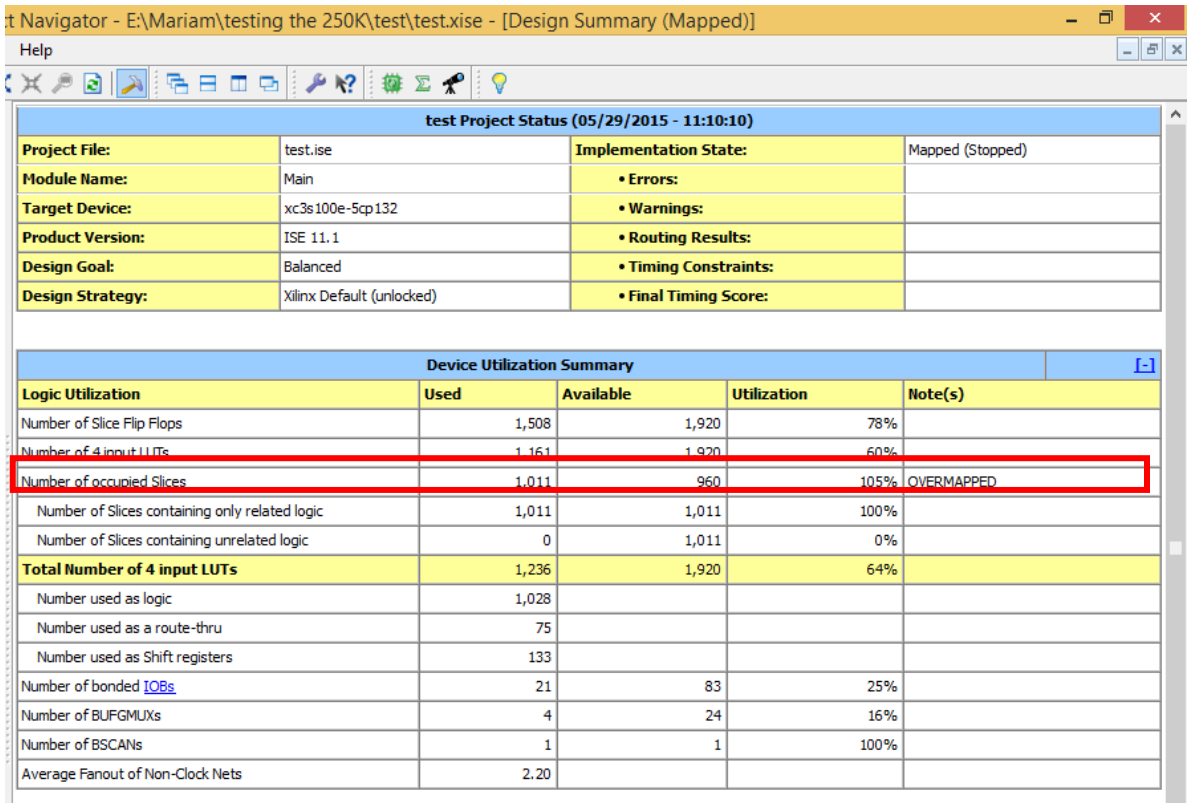
Figure 9: Design summary in the ISE™ software showing an adequate utilization percent when using the 250K gates FPGA, which meets the design requirements.

Checking for a lower density FPGA, the 100K FPGA is tested carrying out the same procedure, however, errors were generated during the Generate Programming file instruction execution as follows;

***ERROR**: Pack: 2310 - Too many comps of type "SLICEL" found to fit this device.*

***ERROR**: Pack: 18 - The design is too large for the given device and package. Please check the Design Summary section to see which resource requirement for your design exceeds the resources available in the device.*

A snapshot of the design summary is shown in Figure 10 indicating for the "*Number of occupied slices*" a utilization percent of **105%** and an **OVERMAPPED** note.



test Project Status (05/29/2015 - 11:10:10)			
Project File:	test.ise	**Implementation State:**	Mapped (Stopped)
Module Name:	Main	**• Errors:**	
Target Device:	xc3s100e-5cp132	**• Warnings:**	
Product Version:	ISE 11.1	**• Routing Results:**	
Design Goal:	Balanced	**• Timing Constraints:**	
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	

Device Utilization Summary [-]				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,508	1,920	78%	
Number of 4 input LUTs	1,161	1,920	60%	
Number of occupied Slices	1,011	960	105%	OVERMAPPED
Number of Slices containing only related logic	1,011	1,011	100%	
Number of Slices containing unrelated logic	0	1,011	0%	
Total Number of 4 input LUTs	1,236	1,920	64%	
Number used as logic	1,028			
Number used as a route-thru	75			
Number used as Shift registers	133			
Number of bonded IOBs	21	83	25%	
Number of BUFGMUXs	4	24	16%	
Number of BSCANs	1	1	100%	
Average Fanout of Non-Clock Nets	2.20			

Figure 10: Design summary in the ISE™ software showing an Overmapped FPGA utilization due to the insufficient number of logic gates (when using the 100K gates) with respect to the design needs.

More information about Spartan-3E FPGAs family of Xilinx FPGAs can be found in [25] and [27].

3.5.2. Memory

As discussed in the previous section, a memory chip should be included on the designed system in order to store the needed configuration files to enable the FPGA to load these configuration files each time the FPGA is powered-up and avoid the need to

configure the FPGA through a PC and a download cable each time the FPGA is powered-on. Normally, **a programmable read-only memory** (**PROM**) is used, this difference between the ROM and the PROM is that the PROM is manufactured as an empty memory while the ROM is programmed during manufacturing.

The memory should be carefully selected according to the utilized Spartan-3E FPGA, each FPGA module has a certain number of gates thus needs a certain number of configuration bits and consequently a certain smallest storage space Platform flash (PROM) to be used that is capable of storing these bits successfully, this is summarized in Table 3. There are 1, 2, 4 and 8Mbit PROMs available from Xilinx and compatible with the Spartan-3E family.

Table 3: Number of Bits to program a Spartan-3E FPGA and the smallest Platform Flash PROM [25].

Spartan-3E FPGA	Number of Configuration Bits	Smallest Available Platform Flash
XC3S100E	581,344	XCF01S
XC3S250E	1,353,728	XCF02S
XC3S500E	2,270,208	XCF04S
XC3S1200E	3,841,184	XCF04S
XC3S1600E	5,969,696	XCF08P or 2 x XCF04S

Therefore, for using the Spartan-3E with 250K gates; XCS250E, the smallest memory that could be used is the XCF02S with 2 Mbit storage space from Xilinx.

This memory chip is a non-volatile ROM called “Platform Flash” (PROM). Once programmed, the Platform Flash can automatically transfer a stored bit file to the FPGA at a subsequent power-on or reset event [30]. This process is controlled using a dedicated jumper (JP3) connected to the FPGA MODE pins; putting the jumper in one position should allow configuring the FPGA only from the PC and putting it in the other position allows the FPGA to load the configuration files from the PROM.

The XCF02S Flash ROM will retain a bit file until it is reprogrammed, regardless of power-cycle events [32], in other words this memory chip needs to be programmed/configured just once to be able to perform its role, even if the PCB was repowered (i.e power is switched Off and On again) several times.

3.5.3. The crystal oscillator / the clock:

The commercial FPGA Basys2 PCB has a silicon oscillator clock, however when testing that with the DAB control, the clock signal was very noisy which prevented proper operation of the DAB module. This primary silicon oscillator is flexible and

inexpensive, but it lacks the frequency stability of a crystal oscillator [30]. Thus, it is better to include a crystal oscillator clock to provide the clock signal needed for the data synchronization and later for the communication protocol. This was first tested using the IC6 socket on the Basys2 PCB, specially dedicated for integrating a crystal oscillator, and was perfectly functioning. Thus, a crystal clock of 100MHz frequency is selected from Fox Electronics FXO-HC33 Series.

3.5.4. Connector to the PC

In order to configure the FPGA and the memory for the first time; there are two case as explained before; first via a JTAG because the Spartan-3E FPGA family has four dedicated JTAG pins that are easily accessible, in this case a download cable (i.e Platform Cable USB) should be used to download the configuration files into the FPGA and the memory using the iMPACT software, thus a 14-pin JTAG connector should be included in the design to perform this function. Second case is via a microprocessor placed on the PCB itself, in this case also a USB connection is needed to upload the configuration files into the microprocessor from the iMPACT, thus a USB port should be included in the design.

The commercial PCB includes a mini USB controlled by an AVR® 8-Bit Microcontroller from Atmel (AT90USB162) used to configure the FPGA and the memory with the .bit file via the iMPACT software or any relevant software.

Therefore, either a mini-USB or a 14-pin JTAG should be included in the PCB design, the choice will be based on the design restrictions and issues discussed later resulting in choosing a JTAG connector.

3.5.5. Voltage regulator

A voltage regulator is needed for providing the different power requirements to all the FPGA Banks, as well as to the other circuits included in the design; the clock, the memory, etc... Since the PCB is intended to be supplied from a 5V supply and the FPGA needs 3.3V, 2.5V and 1.2V, thus a step-down voltage regulator circuit should be included in the design. The selected regulator chip is the same as the Basys2, LTC3545 from Linear Technology in a Plastic QFN package. This regulator chip is supplied using voltage ranging from 2.25V to 5.5V and has three individual buck regulators that are capable of providing the three needed voltages for supplying the four FPGA Banks and the other ICs in the system.

The schematic connection proposed by the datasheet of the LTC3545 [33] is shown in Figure 11.

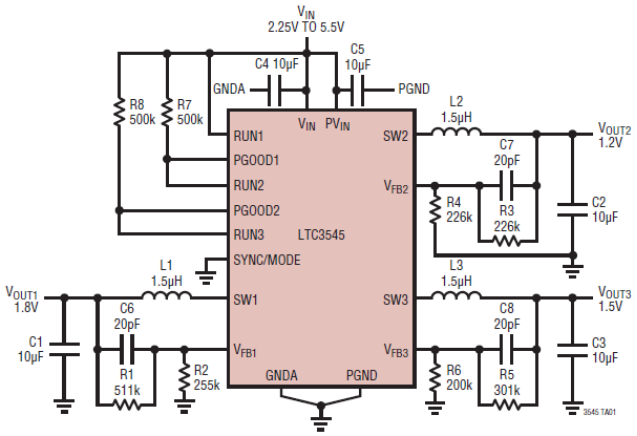


Figure 11: Voltage regulator LTC3545 circuit recommended by its datasheet [33].

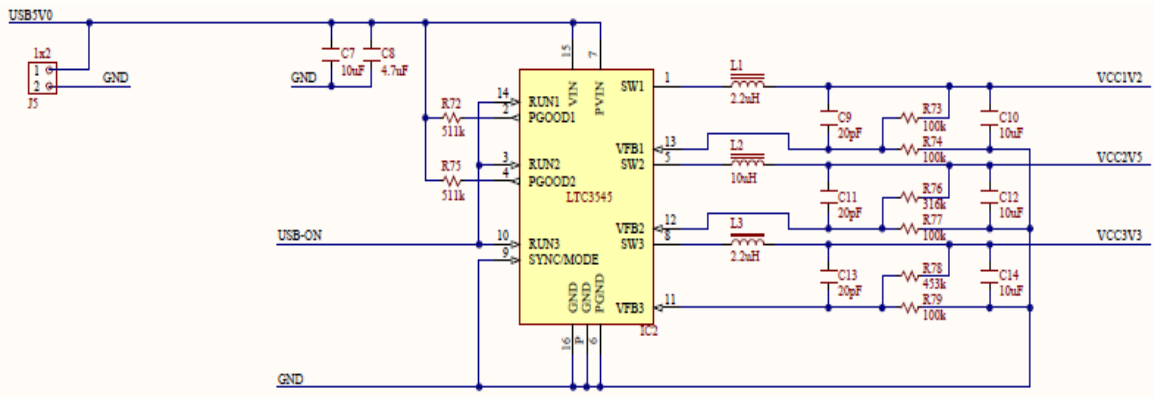


Figure 12: voltage regulator circuit implemented in the reference design [34].

Studying the voltage regulator schematic proposed by its datasheet and the one designed in the schematic of the commercial Basys2, it is easy to design a voltage regulator circuit able to provide the required voltages. However, attention should be given to whether a JTAG or a USB is used because this affects the voltage regulator circuit; as seen in Figure 11of the voltage regulator schematic of the Basys2 which uses a USB and a microcontroller to configure the FPGA, the RUN3 pin (pin10) is connected to the USB, while in the datasheet it is proposed to be directly connected to the Vin as shown in Figure 12.

3.5.6. Interface connector

The FPGA general input and output pins should be connected to an output connector to be able to use them. Instead of using the *Pmod* (peripheral module) interface connector used in the commercial PCB which is not a reliable solution to connect the FPGA to any system, an adequate DIMM socket (similar to the connector used for integrating the DSP into its general board) is chosen to avoid the use of wires making the Slave FPGA PCB integration in the DAB module more robust and reliable.

The number of PCB inputs and outputs are intended to be “63”; 60 pins for use defined general inputs (IP) and inputs/outputs (I/Os) as well as 3 power pins; one has 3.3V, one has the Ground references point of the FPGA PCB and the last is connected to the 5V net in order to be able to later supply the FPGA PCB from the DAB module, instead of an external supply, thus increasing the module reliability.

Therefore a 100-pin connector is chosen. The connector is chosen to be vertical to reduce the space needed on the DAB modules for the integration of the Slave FPGA PCB. It is shown in Figure 13.

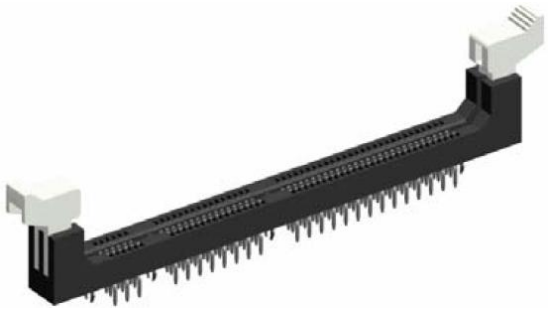


Figure 13: The 100-pin DIMM socket.

3.5.7. Passive elements

Some ceramic decoupling/bypass capacitors are needed to be placed near power pins/VCC of ICs especially the FPGA power pins in order to provide a clean and low-noise power supply. There are decoupling networks presented in the Spartan family FPGA guides and it is recommended that the designed network of decoupling capacitors should meet or exceed the performance of the simple decoupling networks presented in the guide [35].

Also, a zener-diode is placed at the JTAG connector connected to the four important JTAG signals connected to the FPGA JTAG port in order to protect the FPGA from Electro Static Discharge (ESD). This diode is selected similar to the zener-diode used at the JTAG connector of the ZED Board that was used as the central control unit; this is a TE connectivity 4- channel Flow-Through miniature Array diode (SESD0802Q4UG-0020-060). Its function is to protect the FPGA JTAG pins against damage from Electrostatic discharge (ESD) voltage strikes up to 20kV contact & air discharge; if the signal voltage goes below -0.8V or above +9.0V the diode turns on and shunts to GND [36].

Chapter Four

4. Developing the PCB

This chapter deals with all the peculiar specification needed to develop the desired FPGA PCB design, along with the significant challenges faced during the design process. Then the implementation process is further divided into two main procedures, first is the preparation of the PCB schematic and second is the development of the layout design of the board itself and applying all the previously addressed specifications and considerations, obviously in the end a testing procedure is developed, although being a theoretical procedure however it acts as a first step for a successful commissioning process.

This chapter also introduces the analogue to digital converter (ADC) designed and implemented to integrate the functionalities of the digital controller PCB with the ADC function needed to interface the FPGA with the sensing devices to be able to perform its function in the control system. Finally, the communication link established between the slave controller and the central control unit is discussed briefly being the final step needed for completing the full function of the developed FPGA digital controller PCB.

4.1. Design specifications

The chosen commercial PCB to test the functionality of the slave FPGA concept was a Basys2 Spartan-3E FPGA board. Thus it is considered as the reference design PCB, especially that the schematic of this board is available from Digilent (its supplier). The board was previously shown in Figure 3.

Some considerations where taken into account before starting the PCB design in order to simplify the design and the debugging procedure;

First, following the reference design regarding the placement of the components, the decoupling capacitors and the suppliers for most of the key components (i.e using the same voltage regulator IC, PROM chip, values and placement of coupling capacitors etc..).

Second, designing the PCB in two layers to be able to have access to all tracks which enables seeing most of the signals using an oscilloscope during the testing stage. Also, reducing the number of PCB layers improves the electrical quality of the signals [37].

Third, using an edge connector, to mount it in a reliable and robust way on the related DAB PCB with the least space consumption using a 100-pin vertical DIMM socket. Thus

avoiding the use of wires and connectors in the input signals and the output control actions and even the main supply of the FPGA, as shown later.

Fourth, the PCB dimensions are adapted with the dimensions of the MMC and the DAB integrated modules; this means that the width of the PCB shouldn't exceed the distance between the vertically stacked MMC/DAB modules, which is approximately 5cm. However, obviously a margin is taken into consideration, not only for the dimension consumed by the vertical socket which is relatively small, but also to avoid any possible scratches in the stacked modules and also for ease of assembly. Thus the dimensions of the PCB are approximately 9cm x 4cm, as clarified before, the length is restricted by the size of the vertical DIMM socket and the width is restricted by the stacked modules.

4.2. Design challenges/issues

4.2.1. FPGA Package related issue

The **first** obstacle in the design of the slave FPGA PCB was finding an appropriate fanout pattern for routing the FPGA package chosen, the meaning of fanout is related with the BGA FPGA packages and it is to route out all the FPGA pins with tracks to the outside of the package, it can be more clear when looking at Figure 14.

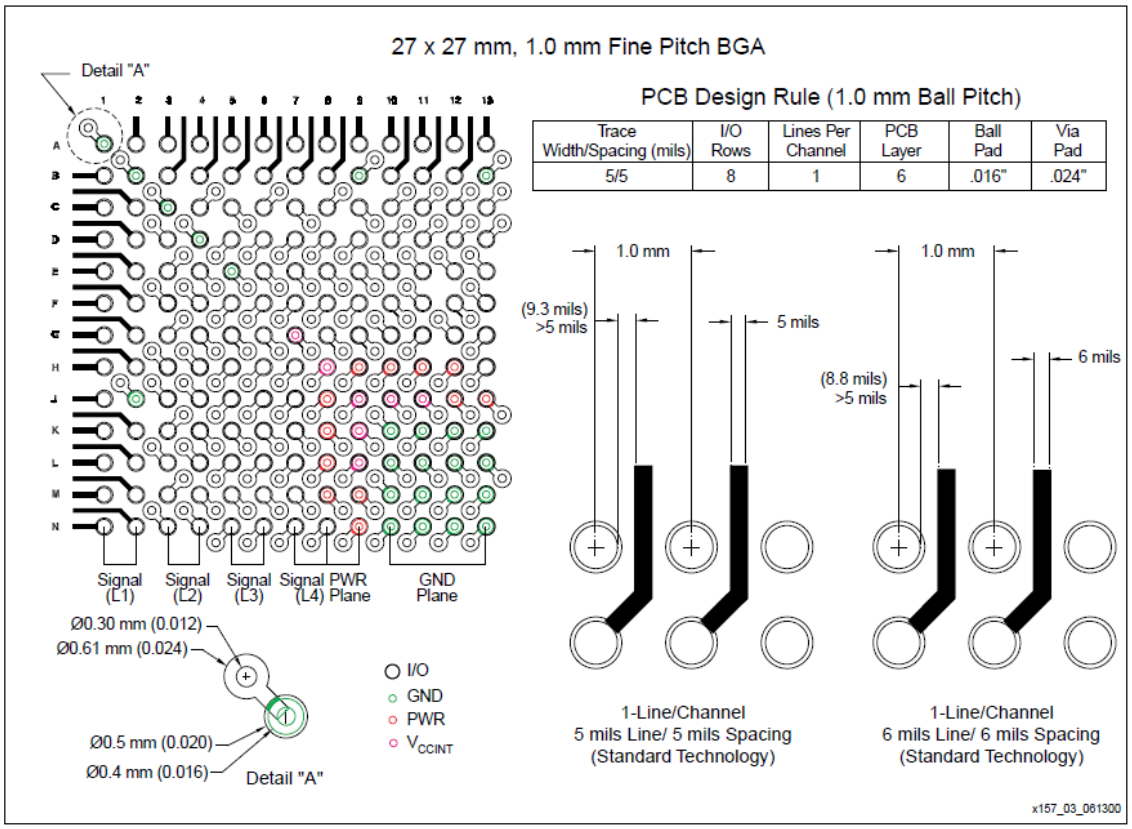


Figure 14. Shows the FG676 PC Board Layout/Land Pattern [38].

The commercial reference design uses a 132-BALL CHIP SCALE BGA with 0.5mm Pitch, CP132package which is a BGA package as shown in Figure 15.

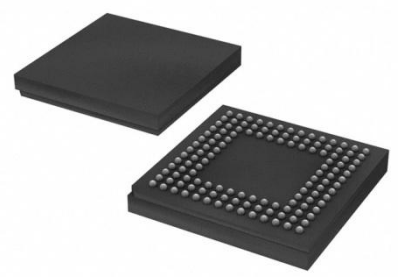
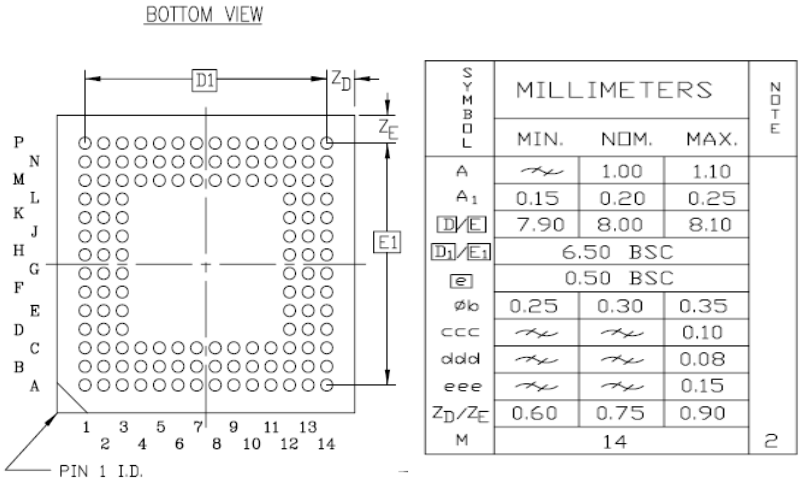


Figure 15. The CP132 BGA package [39].

When trying to use the same package of this FPGA family of Xilinx, it was impossible to make fanout of the inner pin layers while at the same time respecting the rules of the PCB manufacturer related with the track width, clearance and via dimensions, in other words it was not possible to use except the outer pins of the package, because the pitch of the BGA package is 0.5mm and the recommended line width is 0.13mm (as given by Table 4 also referring to Figure 17) which is not in the capabilities of common PCB manufacturers as well as the impossibility of designing the PCB in two layers. Figure 16 shows the pin layout of the CP132 package which, along with Figure 17, clears this idea.



SYMBOL	MILLIMETERS			NOTE
	MIN.	NOM.	MAX.	
A	---	1.00	1.10	
A_1	0.15	0.20	0.25	
D/E	7.90	8.00	8.10	
D_1/E_1	6.50 BSC			
e	0.50 BSC			
øb	0.25	0.30	0.35	
ccc	---	---	0.10	
ddd	---	---	0.08	
eee	---	---	0.15	
Z_D/Z_E	0.60	0.75	0.90	
M	14			2

Figure 16: shows the pad layout dimensions of the CP132 [40].

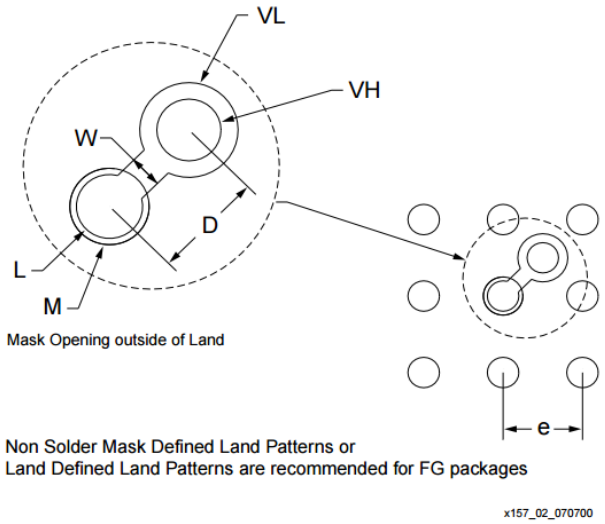


Figure 17. Suggested board layout of soldered pads for BGA, CSP and CCGA packages. [41]

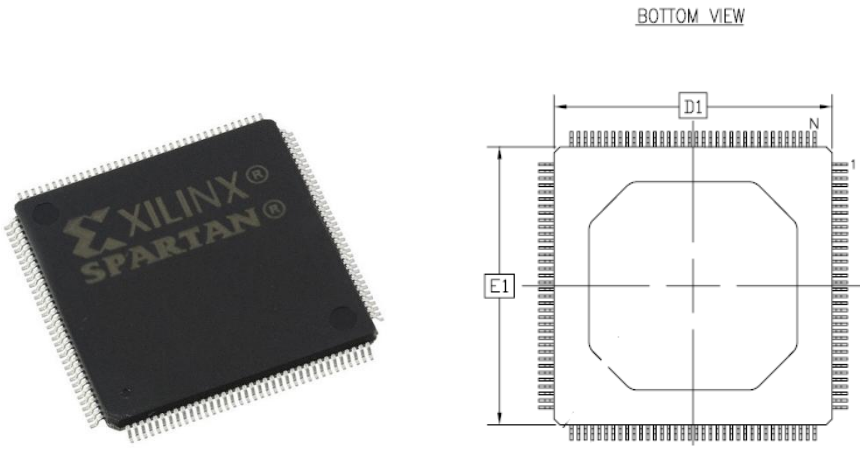
Table 4: Board routability guidelines with Xilinx fine-pitch BGA Packages. Recommended PCB design rules (mm) [41].

	BG225 BGG225	BG256 BGG256	BG352 BGG352	BG432 BGG432	BG560 BGG560	BG575 BGG575	BG728 BGG728	EF957 BF957 BFG957	CS48 CSG48 CS144 CSG144 CS280 CSG280	FS48 FSG48 CS484 CSG484	CP56 CPG56	CP132 CPG132
Solder (ball) land pitch (e)	1.50	1.27	1.27	1.27	1.27	1.27	1.27	1.27	0.80	0.80	0.50	0.50
Line width between via and land (w)	0.300	0.203	0.203	0.203	0.203	0.203	0.203	0.203	0.13	0.13	0.13	0.13
Distance between via and land (D)	1.06	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.56	0.56	0.35	0.35
Via land (VL) diameter	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.51	0.51	0.51	0.27
Through hole (VH) diameter	0.356	0.356	0.356	0.356	0.356	0.356	0.356	0.356	0.250	0.25	0.250	0.15

Note:

1. Component land pad diameter refers to the pad opening on the component side (solder mask defined).

Thus a different package from the same Spartan 3E family was used, the TQ144 package which is a 144-pin Thin Quad Flat Pack package, has much larger dimensions and has a 0.5mm pitch as well but the pins are all accessible without need for complicated fanout techniques and special manufacturing process because of the structure of the TQFP package.



SYMBOL	TQ/TQG/HT/HTG100			TQ/TQG/HT/HTG144			TQ/TQG/HT/HTG176		
	MILLIMETERS			MILLIMETERS			MILLIMETERS		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A			1.60			1.60			1.60
A_1	0.05		0.15	0.05	0.10	0.15	0.05	0.10	0.15
A_2	1.35	1.40	1.45	1.35	1.40	1.45	1.35	1.40	1.45
D/E	16.00 BSC			22.00 BSC			26.00 BSC		
D_1/E_1	14.00 BSC			20.00 BSC			24.00 BSC		
L	0.45	0.60	0.75	0.45	0.60	0.75	0.45	0.60	0.75
e	0.50 BSC			0.50 BSC			0.50 BSC		
b	0.17	0.22	0.27	0.17	0.22	0.27	0.17	0.22	0.27
c	0.09		0.20	0.09		0.20	0.09		0.20
ccc			0.08			0.08			0.08
ddd			0.08			0.08			0.08
N	100			144			176		
REF.	JEDEC MS-026-BED			JEDEC MS-026-BFB			JEDEC MS-026-BGA		

Figure 18: The TQFP (TQ144/TQG144) Spartan FPGA Package [42].

4.2.2. FPGA configuration issue

Another issue detected during the final stage of the design was regarding the use of an Atmel microcontroller to control the USB which is connected to the computer and used to program the FPGA; the problem is that this type of microcontroller has certain configurations that should be programmed before being able to control the USB, these configurations are not available by Atmel nor Xilinx. Thus, to be able to program the Xilinx Spartan-3E FPGA, a JTAG connector appears to be the obvious option. It is common to program FPGAs using a programmer from Digilent or a Platform Cable USB from Xilinx, thus the USB and the microcontroller has been replaced by a 14-pin JTAG connector.

4.2.3. PROM configuration issue

The previously mentioned problem of the microcontroller arises the worries about another possible issue; this is regarding the PROM memory used to store the program to

be able to rerun it while disconnected from the computer using the dedicated jumper connected to the FPGA Mode pins; this memory IC needs some configurations before being able to perform its role, exactly similar to the microcontroller. However, the datasheet of the XCF02S PROM stated clearly that these configuration bits are available by the manufacturer, Xilinx and are automatically generated by their dedicated software iMPACT™ according to the family of the PROM used and these files are loaded into the PROM while it is being configured.

4.3. Developing the Design

This section explains the procedure of designing the FPGA-based controller PCB, specifically it is divided into two major steps; first developing the whole schematic of the needed PCB circuits and components and making sure of all the connections and second, developing the design layout by firstly considering the most suitable organization of the components on this High frequency PCB following in this the design chosen as a reference, then routing the most important signals manually respecting the design rules for high frequency boards, and finally the rest of the signals can be auto-routed as a first iteration.

The design is developed using Altium™ software.

4.3.1. Schematic of the FPGA PCB

The first step to be done in developing a PCB is preparing the schematic; this should be done in an intelligent way following a systematic procedure in order to eliminate major mistakes. The following steps are followed to develop the schematic;

4.3.1.1. Studying the reference design

In this case the reference design is the Basys2 [30], [34], thus the schematic of the basys2 is analyzed carefully, the expected results of this analysis are three important decisions each based on the previous one;

First, being able to clearly identify the differences between this design and the design intended to be developed based on the design specifications and the design challenges as well as capabilities previously discussed. In this case, these major differences are;

a- This design is configured using a USB and a microcontroller while the configuration in the intended design is done using JTAG.
b- The FPGA package intended to be used is different as discussed before and thus the number of pins of the FPGA are different; the TQ144 has 144 pins while the CP132 has 132 pins, however theSpartan-3E FPGA family datasheet [25] provides a Pinout table with the pin indicator in a certain package and its global

name in all the family, this makes it easier to benefit from the CP132 design, by comparing both pinout tables of the two packages.

c- The design will use a crystal oscillator as a clock not the silicon clock used in the Basys2.

d- The output and input signals are connected to a 100-pin edge connector instead of the four PMOD headers used.

Second, based on the previous step, being able to decide which components should be eliminated. These components are two categories; the component **no longer needed** for the board to perform its functionality and the **general** usage components. Thus, the components to be eliminated are; the microcontroller, the mini-USB, the general LEDs, the general switches, the seven segment display, all the output connector buffers which will be later included on the DAB PCB for the used signals only, the pushbuttons, the VGA header, the IC6 socket used to integrate an additional oscillator in the Basys2, the PS2 connector, the four PMOD headers and their zener diodes which will be later added, on the DAB board, to the utilized signals only.

Third, deciding which components to be added; also depending on the previous two steps, the added components are, the 100MHz crystal oscillator replacing IC6 socket dedicated for the same reason, the 14-pin JTAG connector and an ESD protection diode connected to the JTAG connector.

In the end, a list is made with all the required components indicating the function of each of them in the design.

4.3.1.2. Placing the components

It is preferred to place each major component in different sheets of the Altium schematic sheets of the created project to avoid complicated designs, thus the design was divided into five small partitions and thus six schematic sheets to simplify it;

a- The JTAG and the ESD diode.

b- The voltage regulator circuit components; including the voltage regulator IC, the input and output capacitors, the regulator resistors and the required inductors.

c- The PROM and the relevant partition of the FPGA (i.e the FPGA component on the Altium is preferably divided into banks or partitions to be able to connect them with less complexity and less error percent).

d- The FPGA main four banks along with the crystal oscillator clock and the silicon clock chip.

e- The FPGA power bank, supplying all the FPGA with its needed power requirements and including all the decoupling capacitors, which are quite a big number to provide as clean as possible power to the FPGA.

f- The 100-pin connector, which should include all the names of the existed signals from the developed design to the real outer world where they will be utilized.

It is preferable also at this stage to create all the footprints needed for all the utilized components and include them in one pcb.lib library file, this should be done very carefully to avoid very common mistakes of mirrored footprints especially in miniature packages that are very hard to correct after the PCB is developed. It is preferable, for important and hard to solder components, such as the PROM and the FPGA, to use the footprints provided by Altium designer downloadable footprint libraries which are freely available for many component footprints. This is in order to eliminate major design problems arising due to footprints.

4.3.1.3. Connecting the components

This is the step where the design schematic is finalized by linking all the components and the relevant circuits to each other to produce a complete design that makes sense.

This is the most critical step because most of the critical, hard to fix, mistakes take place during performing the connections. Also this is the final step before starting the layout design in which it is very hard to discover mistakes in the schematic. Thus, dedicating enough time for that can avoid tiny mistakes, that result in severe problems, such as; duplicate naming, missing connections, wrongly assigning FPGA pins to their functions, wrongly written net names, wrongly assigned values for passive components, ..etc.

The schematic connections of the main five circuits are shown below;

A. JTAG

The JTAG connector is the interface between the programmable devices on the PCB and the Computer to be able to configure or program these devices (i.e PROM and FPGA). Normally this type of connection needs a protection diode for ESD problems, thus an ESD diode is attached to all the signals lines (i.e TMS, TCK, TDO and TDI) as shown in Fig.

These connections are critical as explained before, the TCK signal is connected directly to the FPGA and the PROM pins, to provide an input clock rate from the programmer for the configuration process. Also, the TDO signal is connected to the TDO of the Master-serial chain which is the TDO of the PROM in this case, also the TDI is connected to the TDI of the chain which is the TDI of the FPGA. The TMS signal is an input to both the FPGA and the PROM. The names of the signals should be assigned in a way to give as clear as possible idea of the connections as shown in Figure 19.

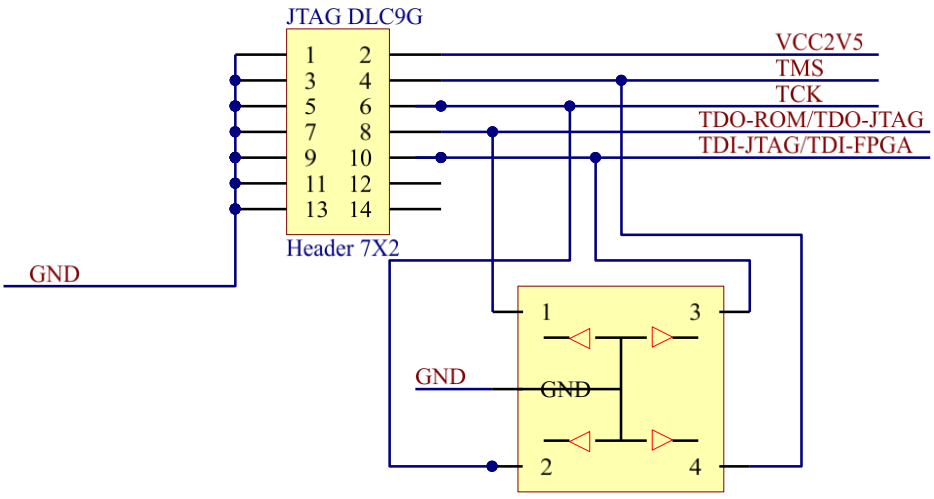


Figure 19: The connection of the JTAG along with the ESD protection diode.

B. Voltage regulator

The regulator circuit is the most important circuit in the design, it provides the power to all the components including the FPGA which has certain power requirements, thus attention should be paid while designing this circuit and the values of the passive components should be checked, in this case the reference design; the Basys2 PCB requires the same power requirements as the designed PCB, thus similar values are used for the inductors and the feedback resistor divider used in the LTC3545 circuit provided by its datasheet as shown in Figure 11.

The LTC3545 has three voltage regulator.The three RUN pins are enable pins for the three regulators. In the case of microcontroller-controlled USB connection these pins are connected to the microcontroller, however in the case of simply using a JTAG, these pins can be connected directly to the input voltage (i.e 5V), thus the regulator is always enabled if input voltage exists.

Also, as shown in Figure 20, a switch is used to connect the input voltage easily and thus this switch is considered the PCB power switch, also including two decoupling capacitors to provide an adequately clean input DC voltage to the circuit. At the output of the regulator three 10uF decoupling capacitors are used to decouple the effect of switching inside the regulator circuit from the rest of the supplied circuit which may include sensitive data lines.

As a visual indication of the functionality of the voltage regulator circuit, a LED is used at the 3.3V output.

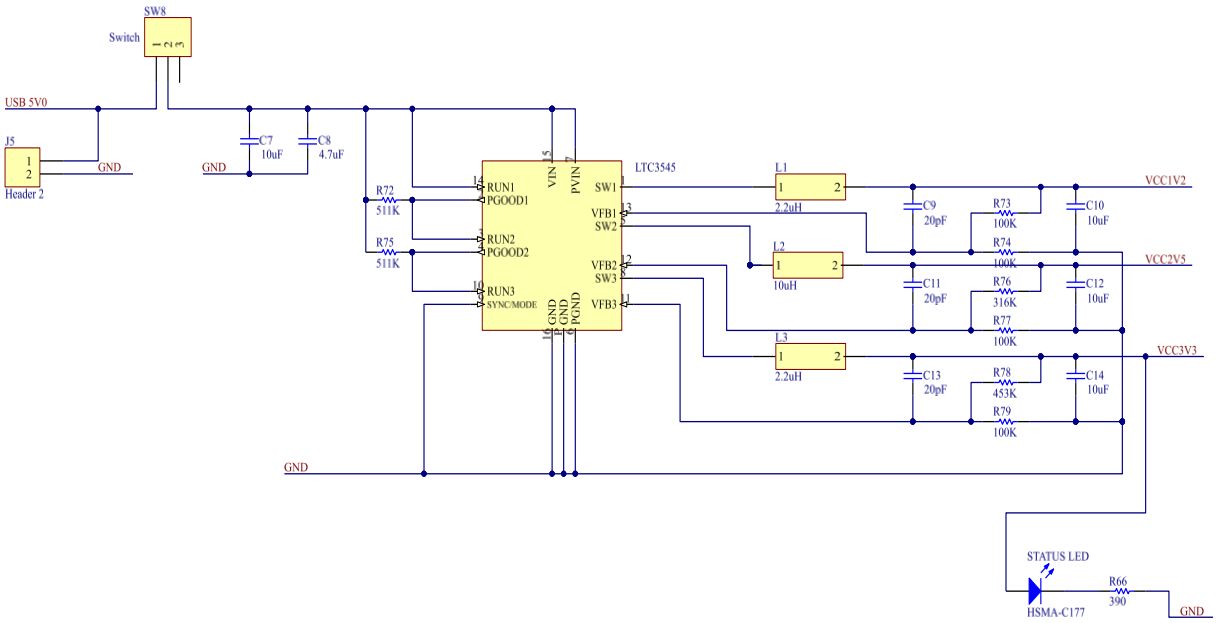


Figure 20: The Voltage regulator schematic in the developed FPGA design.

C. Platform Flash PROM

The connection between the FPGA and the PROM are done based on the Master-serial connection diagram in

Figure 6and also the reference design schematic which is a little different in this case due to the use of the JTAG instead of the USB.

The Platform Flash is connected mainly to the FPGA JTAG dedicated port in a Master-serial connection as explained before. Also it has some connections to Bank 2 of the FPGA; the serial data input (DIN) of the FPGA is connected to the serial data output (DOUT) of the PROM, also the CCLK pin of the FPGA is connected to the input CLK pin of the PROM to supply the PROM with the configuration clock needed to allow the PROM to start configuring the FPGA.

Another important consideration is the MODE pins, if these pins are High (i.e connected to VCC) then the FPGA can be configured via the JTAG and if these pins are Low (i.e connected to GND) then the FPGA configuration will start automatically via the PROM when the FPGA sends the configuration clock correctly to the PROM. This connection is clear in Figure 21.

The input current should be limited as well. However, caution should be taken when using current limiting resistors.

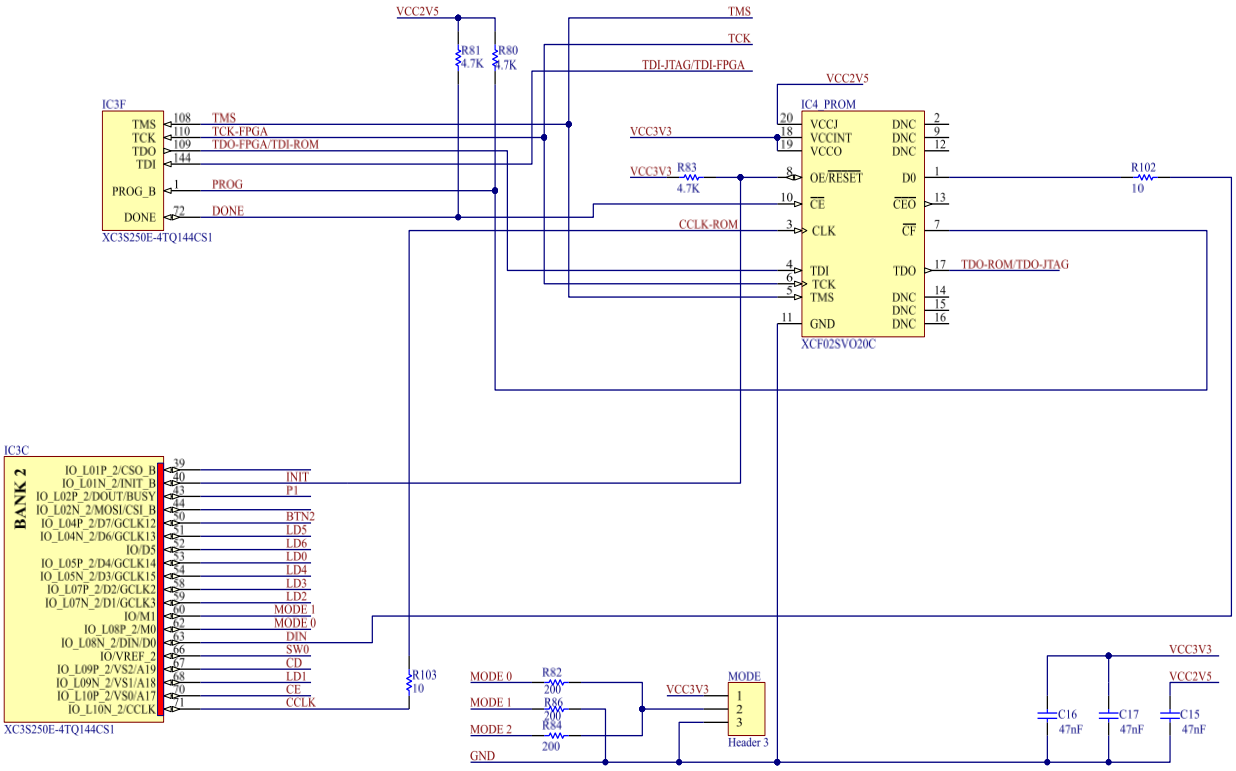


Figure 21: The Schematic of the PROM showing the connections to the FPGA JTAG port and the BANK 2 which contains the serial data input and the MODE pins.

D. FPGA

The integration of the FPGA into the design is done based on two main principles, the type and the location; this is defined by the types of pins on **Spartan-3E FPGAs** and the Pinout description Table of the chosen package TQ144, these are clearly indicated in the Spartan-3E FPGA family datasheet [25].

Most pins on a Spartan-3E FPGA are general-purpose, user-defined I/O pins, however, there are up to 11 different functional types of pins on Spartan-3E packages [25], these are briefly summarized in Table 5;

Table 5: Types of Pins on Spartan-3E FPGAs

Pin type	Description
I/O	Unrestricted, general-purpose user-I/O pin
INPUT	Unrestricted, general-purpose input-only pin
DUAL	Dual-purpose pin used in some configuration modes during the configuration process and then usually available as a user I/O after configuration.
VREF	Dual-purpose pin that is either a user-I/O pin or Input-only pin, or, along with all other VREF pins in the same bank, provides a reference voltage input for certain I/O standards

CLK	Either a user-I/O pin or Input-only pin, or an input to a specific clock buffer driver.
CONFIG	Dedicated configuration pin, two pins (DONE and PROG_B)
JTAG	Dedicated JTAG pin, four pins (TDI, TMS, TCK, TDO)
GND	Dedicated ground pin, all GND pins must be connected
VCCAUX	Dedicated auxiliary power supply pin, a; connected to 2.5V
VCCINT	Dedicated internal core logic power supply pin, all connected to 1.2V.
VCCO	Supplies power to the output buffers within the I/O bank and sets the input threshold voltage for some I/O standards.
N.C.	Not connected.

The number of pins available of each type depends on the package. Also the location of each pin of a certain type depends on the package pinout table. Figure 22, shows the different type pin locations defined with the color code.

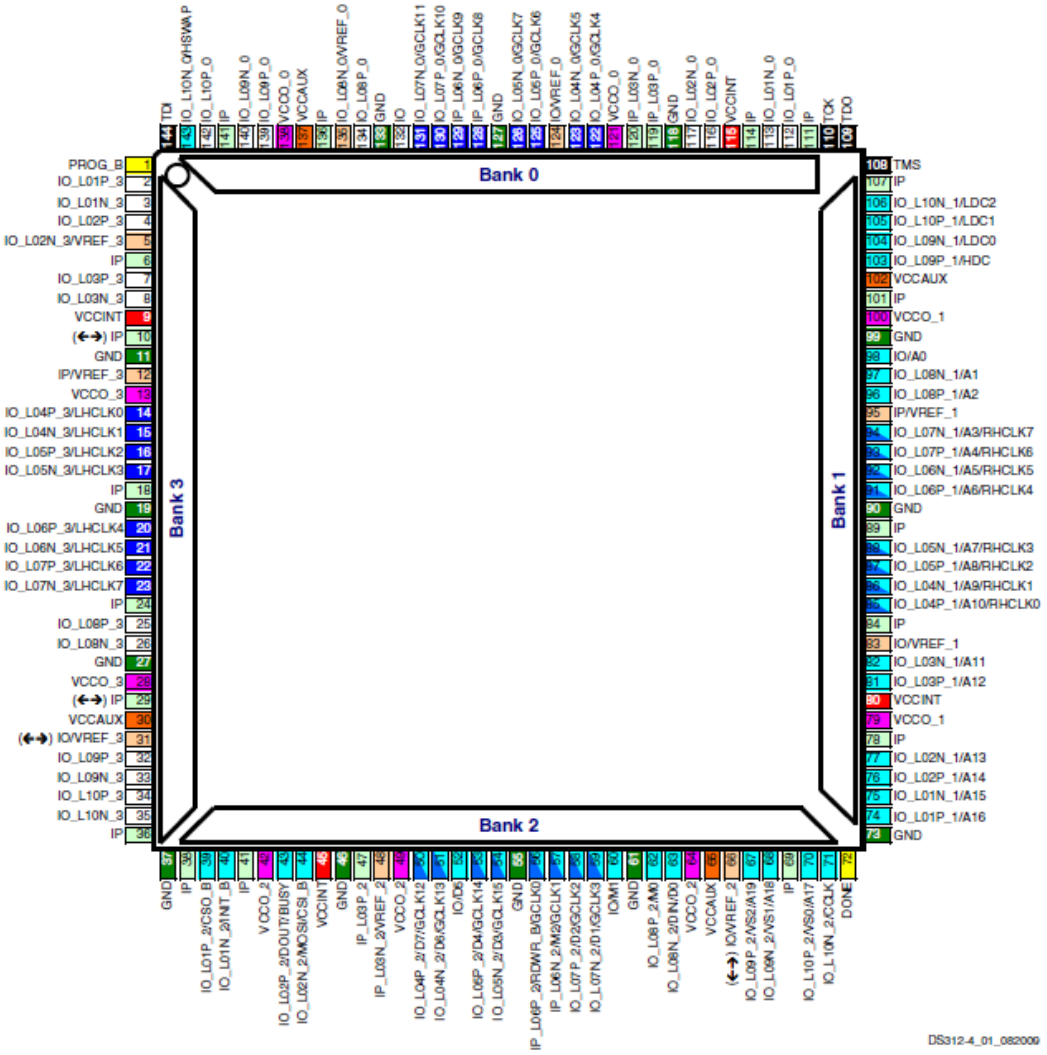


Figure 22: TQ144 Package Footprint (top view) with pin types [25].

Consequently, as a first step for embedding the FPGA into the design, it is necessary to define, for each signal to be connected to the FPGA package, its type with respect to Table 5, the bank to which it will be connected (this can vary later depending on the welfare of the layout design, i.e the location of the connected component with respect to the chosen bank) and the pin to which this signal is connected in the reference design (if exists). Following this, it simplifies the process of connecting all the signals to the FPGA in an efficient way and with minimum error percent.

When choosing the number of general I/Os to exit from the FPGA, it is important to take into consideration two important aspects, first, trying to exit as many as possible general user-defined pins to the output interface connector is not a good idea for a first prototype. This is because as the number of exited user-defined signals increases, the number of layers in which the routing of the design is possible, increases as well, which is an obvious conclusion. Second, for a function-tailored design it is better to exit a suitable number of pins to the output connector with a little margin for future added functionalities or enhancements in the system overall implemented control rather than to exit all available user-defined pins to the output connector.

Regarding the clocks, the design has a silicon clock of adjustable clock frequency (25, 50 and 100MHz) that is a low performance clock as well as a crystal clock of 100MHz.

Figure 23, shows the FPGA connection performed taking into consideration that the majority of the connected pins are general user-defined pins and also shows the clocks connections.

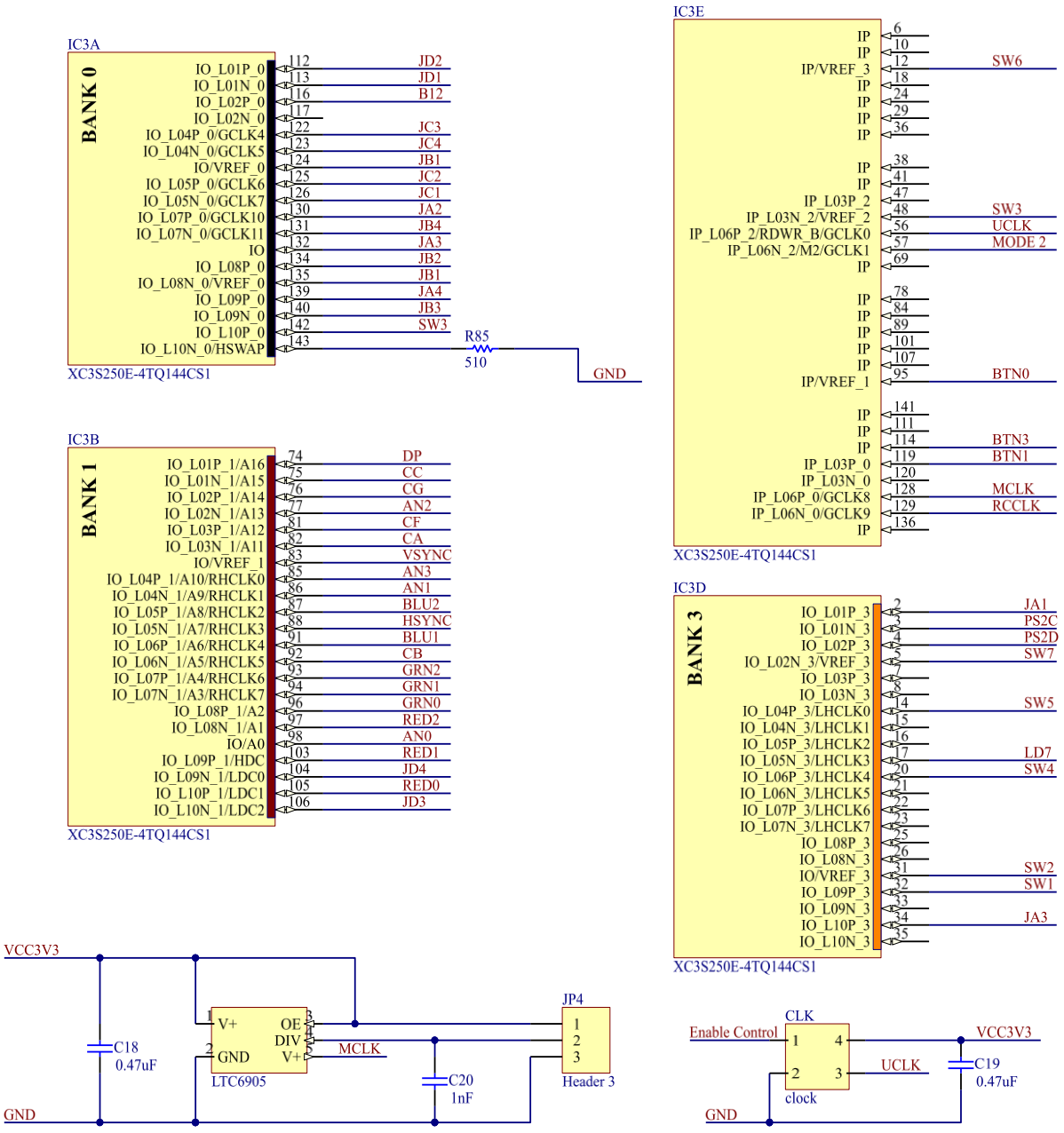


Figure 23: The schematic of FPGA BANKs and the Clocks.

E. FPGA Power

It is very important to pay attention to the FPGA power connections because doing a small mistake in the power connection will result with a 100% probability in either damaging several blocks of the FPGA or unpredictable operation and malfunction. These connections should be done strictly following Table 5, and following the available reference design which uses the same FPGA family as developed design, thus it should have exactly the same power connections as indicated by the family datasheet [25].

Power connection is clear from the Altium™ schematic snapshot shown in Figure 24, as observed the number of decoupling capacitors is very high in order to have a clean power supply input to the FPGA power pins.

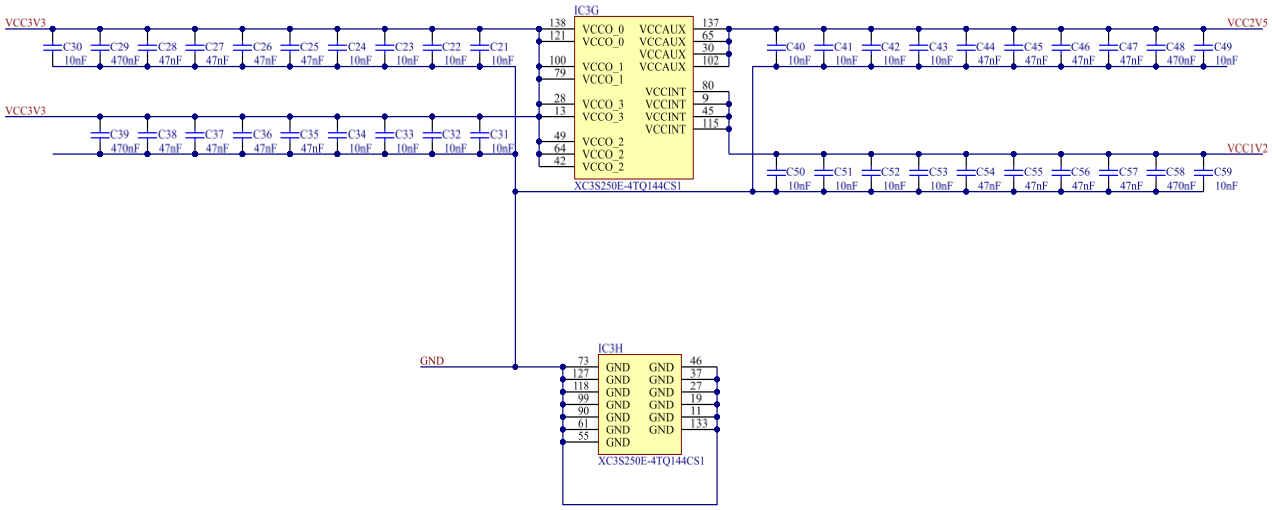


Figure 24: The Power and Ground pins along with their relevant decoupling capacitors.

F. Interface connector

The interface connector is the output connector interfacing the slave design with the intended controlled system, which is the DAB board through the DIMM 100-pin socket.

The connection of the signals to the output connector is based on the type of connector, in this case it is a DIMM 100-pin socket, thus outputs should be distributed on two layers, 50 pin on each. This also puts a restriction to the number of exited signals from the design to 100 signals, however, the design complexity and restrictions usually puts the tighter rule for this. In this design the number of exited general user-defined pins are 60, also three power pins are exited; the ground, the 5V and the 3.3V in order later to be able to supply the FPGA from the DAB as explained before.

A good question now is, how to choose the output connector pin adequate for each signal?, this is simply and solely done based on the Design layout of the PCB; in other words depends on the location of this signal with respect to the four available FPGA banks as well as the orientation of the FPGA IC and the placement of the other surrounding components, thus this schematic should be always changing in the beginning stage of the layout design.

Spending time in choosing the most adequate output pin for each exited signal can significantly decrease the complexity of the design and the needed number of layers.

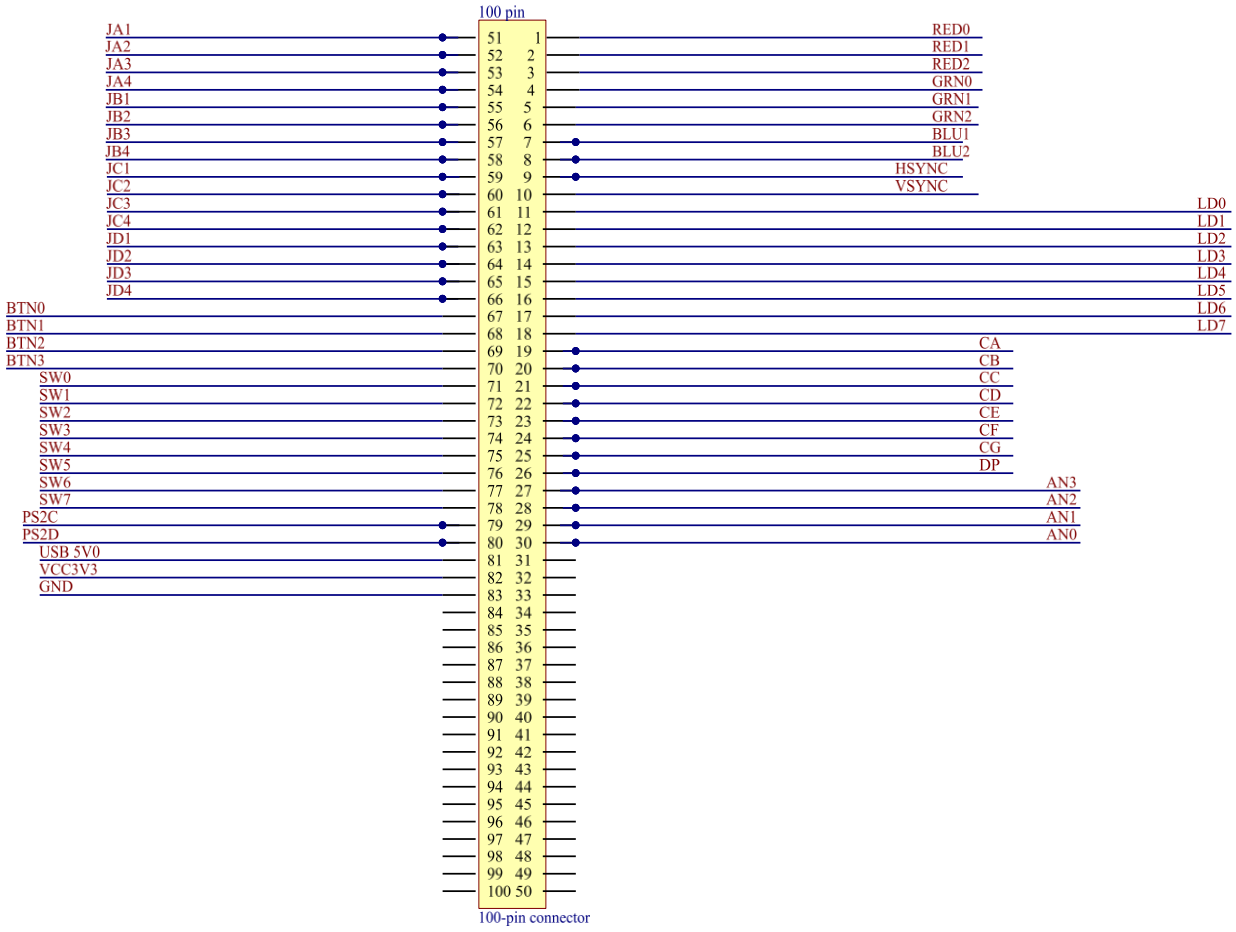


Figure 25: The 100-pin interface connector.

4.3.2. Layout design of the board

The second stage of the PCB development is the layout design or the final design of the board including the placement of all the components and the routing of the connections between them. This stage is a critical stage in the functionality of a printed circuit board; a bad layout design can cause the PCB not to function correctly or to cause problems in the overall control system in which it is embedded.

The addressed layout design is complicated due to the fact that it includes an FPGA as well as many other accompanying factors as explained later in this section, thus, after the schematic is final (with the exception of the interface connector schematic), the PCB design is initialized ruled by specific design consideration and the high frequency considerations.

4.3.2.1. Specific design considerations

These considerations are the practical implementation of the Design specifications stated in the beginning of this chapter, which could be summarized in;

1- Routing all components in two layers only.
2- Following the components placement of the reference design, including the very close placement of the decoupling capacitors to the power pins.
3- Using edge connector consequently signals are routed according to their location on the layout to the nearest output connector pin.
4- Dimensions respect the available vertical space between stacked DAB modules and the available horizontal space of the DIMM socket.

4.3.2.2. High frequency considerations

High frequency signals criticality comes from the fact that these signals can impose noise or cross-talk behavior on the surrounding signals. Also, High frequency signals includes high-speed switching signals as well (i.e I/Os of the FPGA switching at very fast rates), which produces significant transient changes in power supply voltage because a signal switching at higher frequency consumes a relatively higher amount of power than a signal switching at a lower frequency [43].

Taking into consideration the existence of 100MHz clock in the design, as well as the high-frequency switching transistors inside the FPGA, thus special consideration of high frequency signals routing should be respected.

These considerations include;

- Avoid running High frequency signals closely parallel to power traces for long distances.
- Using ground planes under signals to provide a low impedance path for the return signal.
- High-speed switching signals of the FPGA should be routed in short traces (e.g: the configuration signals).

4.3.2.3. Place-and-Route stage

Place-and-route stage in a printed circuit board design refers to two important steps related to the layout design; first step is placing the components and second step is routing the connections between the components. This layout design is relatively a complex one due to some reason introduced in the next subsection.

A. The reasons for the Layout complexity

Following the previous discussed rules and considerations, developing the design was relatively organized. However the complexity of the design still exists due to the existence of the FPGA fine pitch component. It has, within this specific design, 63 signals that should be connected to the output interface connector (60 general user-defined I/O pins, 3.3V.5V and GND) as well as the 6 signals that are connected between the FPGA and the memory for the configuration purposes (PROG_B, DONE, TDO, internal FPGA Clock, serial data DIN and INIT_B). Also the 3 JTAG signals connected to the JTAG port of the FPGA to close the Master-serial configuration mode (TMS, TCK and TDI) and the 2 input clock signals to the FPGA. Also, the HSWAP signal connected to ground to enable pull-up resistors on all I/Os and input only until configuration. Finally, the 3 MODE pins connected to the jumper to identify the configuration mode. Thus, making a total of 78 signals apart from power and ground. Counting the power and ground, then the output signals from the FPGA package needing to be routed are 108 signals. This is a relatively large number of traces for the available routing area and a two-layer-routed design with an edge connector.

Also, the clock signal is 100MHz signal which needs to be routed following certain consideration.

Another reason is that auto-routing is not a fully possible option here because it results in many mistakes due to the complexity of the connections as well as the challenge of detecting mistakes and wrong connections taking into consideration the number of signals to be routed; using the DRC (design rule check)is not straightforward because it can only catch errors that it has been programmed to point-out, despite all this, auto-routing could be in some sort useful to organize the design, as will be explained later in this sub-section.

A final reason is respecting that the manufacturer capabilities have limits for many constrains such as minimum tracks' width as well as achievable minimum clearance, these limitations should be understood when the PCB layout design procedure is started.

B. Developed design

The development stage is divided into four consecutive steps; these are explained with details below, also Figure 26 shows a simple scheme of the developed design to simplify understanding the followed procedure;

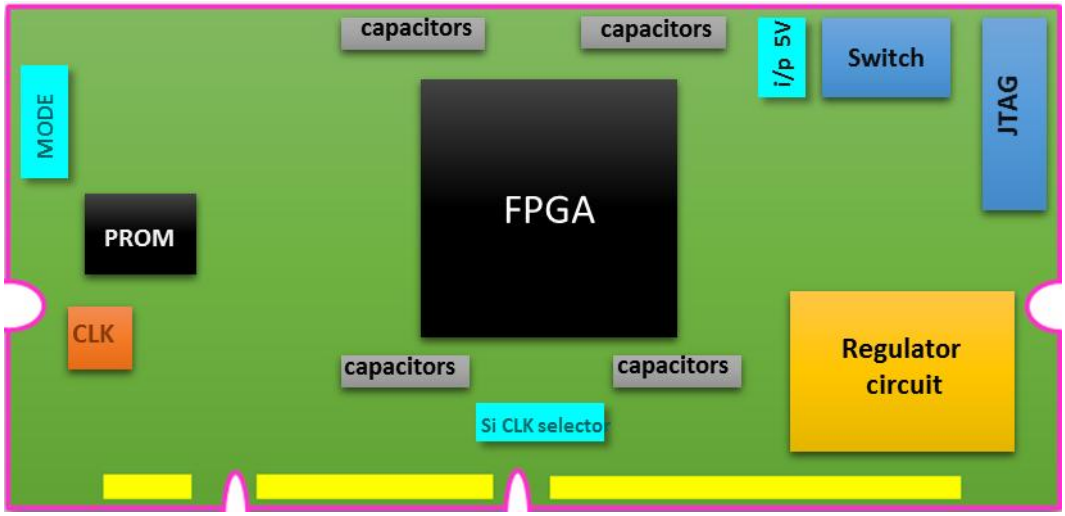


Figure 26: The simple structure of the developed Layout Design.

1- Shaping

First, specifying the PCB shape and size with a rectangle in the keepout layer specifically the horizontal width to respect the DIMM socket size which is critical. Then the areas were no components should be exist, are marked in the keepout to allow insertion and extraction of the PCB into the socket. This is clear from Figure 26.

2- Placing components

Placing the components is a challenging task, good components placement can lead to better thermal management and better functioning, as well as minimizing electrical noise imposed on signals. Also, planned component placement in the sense of putting related components close to each other can allow decreasing the number of routing layers (i.e enables achieving a two-layer design).

Placing the components has been outlined by the reference design layout through observing the critical component placement on the Basys2, however, the procedure can't be given by a reference, in this design the FPGA has been placed first in the heart of the PCB because it is connected to all the components on the PCB, only the orientation of the IC can still represents a degree of freedom chosen according to the placement of the adjacent relative components and the connections routed to the interface connector.

Next, the decoupling capacitor related with the FPGA power pins (clear in Figure 24) are placed as close as possible to all power pins to effectively play their role without increasing the tracks' inductance.

Afterwards, the memory is placed on the side where the most of the configuration and programming dedicated pins of the FPGA exist. Then, the clock is place, it is preferable to place the clock near the edges of the PCB to be able to route the Clock output high

frequency signal along the edge of the PCB to decrease as possible the number of parallel signals that may be affected by crosstalk effect (explained later).

The voltage regulator circuit or the power circuit is placed on the side of the power switch and the JTAG in order to decrease the length of the 5V trace feeding the regulator, this is a critical circuit, its components should be placed close to each other specially the decoupling capacitors related with the input voltage pins and the output regulated voltages.

3- Routing

The objective of this stage is to route the connections between all the components using traces respecting the rules of the design, for example, trying to route all the connections in two layers only. The routing should also respect some basic design rules; trace corners should be more than 90^{o} to avoid antenna effect, traces can't cross, decreasing the number of used vias when possible, decreasing the length of traces, high frequency signals shouldn't run in parallel to power lines to avoid imposing electrical noise and polluting power fed to critical components, power tracks should be as thick as possible to reduce the impedance of the track thus decreasing the losses as well as withstanding the increased current bared by these tracks.

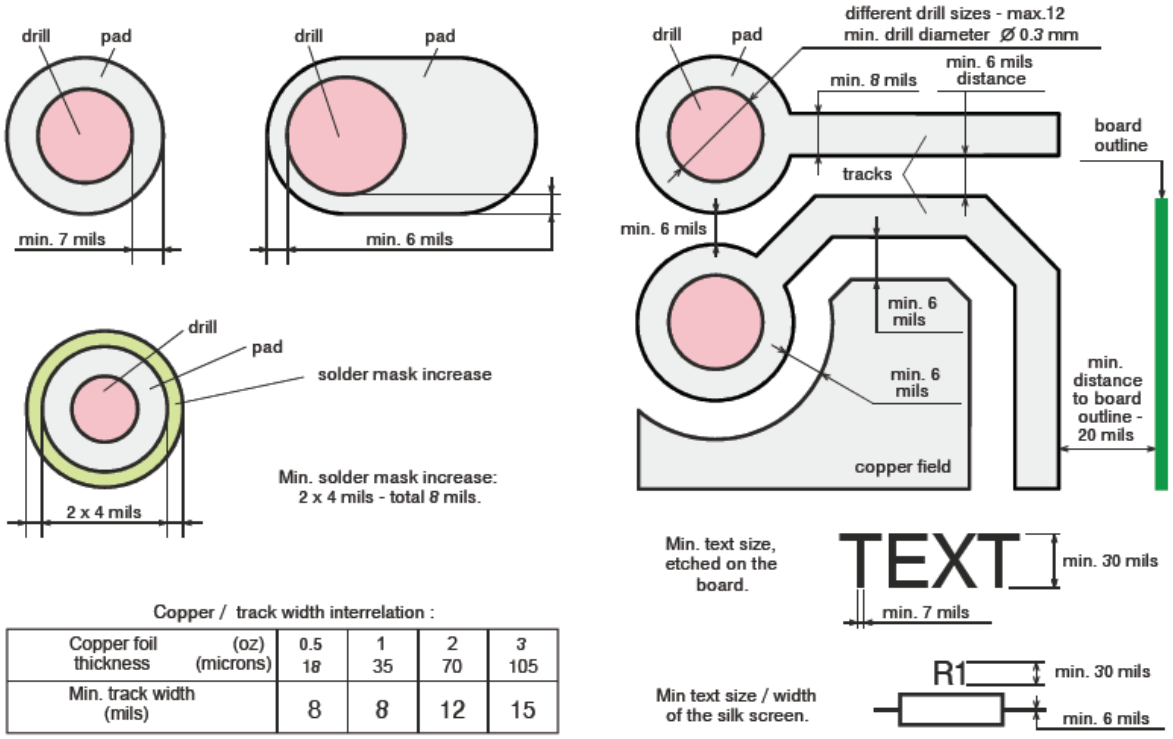
The degrees of freedom available at this stage are; the components orientation and the connections to the interface connector, these can be easily used by common sense to decrease the complexity of the routing process and achieve the previous requirements for a good design.

Auto-routing is not a feasible option due to the complexity of the design, however, Auto-routing option can be useful by choosing to automatically route the connection while selecting the option edge connector and two-layer-routing, the result is the routing of many connection (not all due to un-optimized connections to the edge connector as well as some unsuitable component orientations), these auto-routed connections are considered as a guideline to route the PCB in two layers following a simple procedure to make the routing process easier; first, each auto-routed signal should be checked to respect design consideration and then rerouted neatly, then power traces are made thicker according to their voltage, and finally unrouted connections are attempted taking into consideration the modification of the connection pins of the interface connector to the FPGA pins as suits the design.

To finalize this stage, the orientation chosen for each component should be clearly indicated by putting a point in the Overlay layer at the first pin of the component.

In the end, a DRC is run after specifying clearly the limitations set by the manufacturing company as shown in Figure 27, as well as short circuit, unconnected pins,

unrouted connections, less than 90^{o} traces and top and bottom overlay clearance with respect to components (if the PCB is to be produced with a silkscreen). ..etc.



Copper / track width interrelation :

Copper foil thickness (oz) (microns)	0.5 18	1 35	2 70	3 105
Min. track width (mils)	8	8	12	15

Figure 27: The PCB manufacturer (MICRON-20) capabilities for developing the PCB.

4- Ground planes

It is important to consider using ground planes in a merely digital design. When two traces run parallel to each other, electromagnetic induction by electromagnetic field lines of one trace signal may link the other trace causing coupling between the electrical signals from on into the other, this is called crosstalk [44]. Aground plane provides a low impedance return path close/underneath to traces (i.e Transmission line effect).This forces the opposite polarity return signal to follow this path instead of passing through any near trace and be imposed over the signal carried by this trace. Thus, ground planes reduce the crosstalk between signals because no track will have lower impedance than the ground plane especially in a dense design where tracks are very thin. This bounds all the electromagnetic field associated with the electrical signal to the area near the track reducing crosstalk which pollutes victim signals.

A snapshot of the developed design is shown in Figure 28 and the PCB information report is shown in Figure 29, the PCB has 87 components in total.

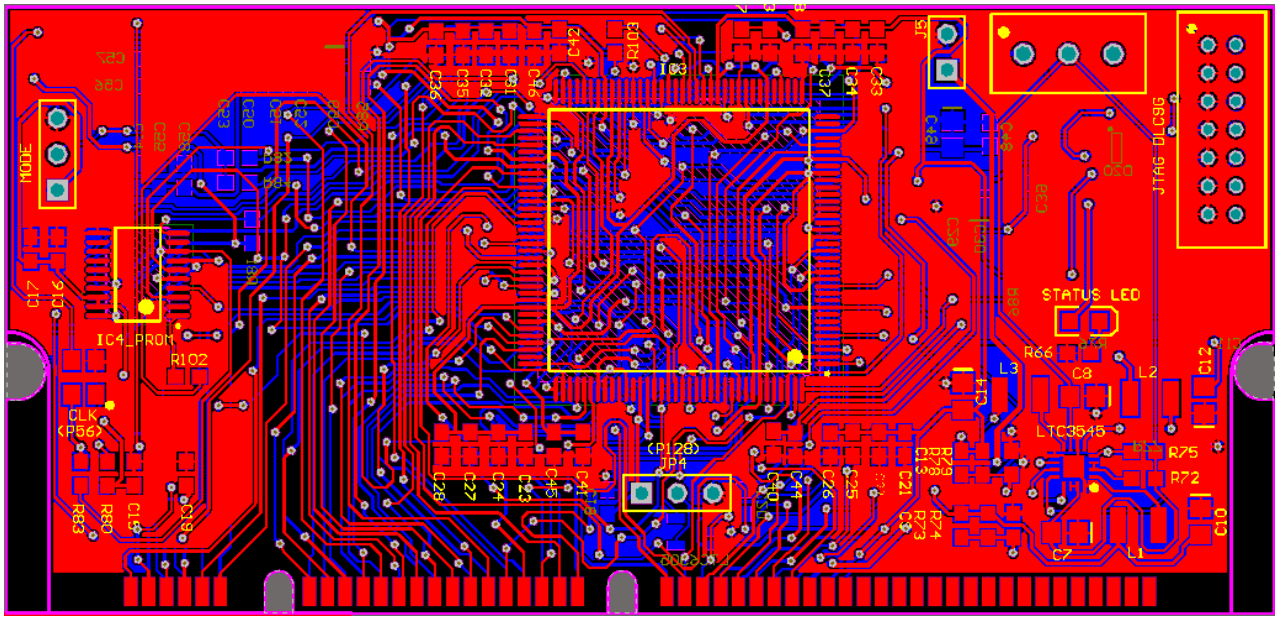


Figure 28: The developed Layout design shown from Altium.

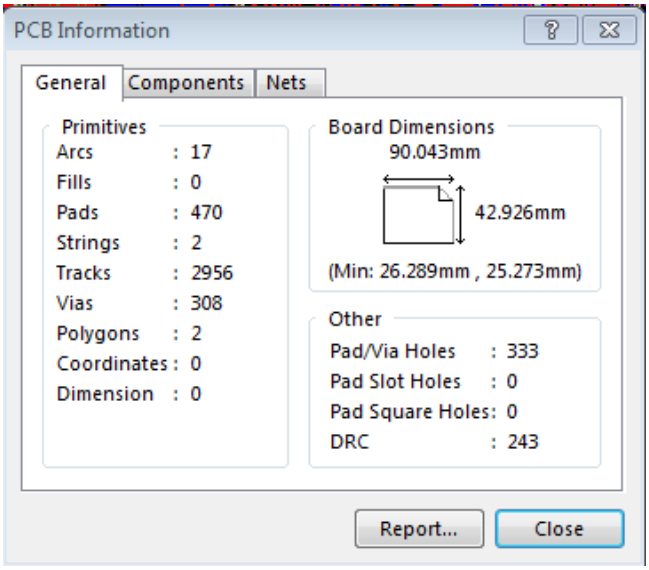


Figure 29: The developed design PCB information report generated from Altium.

After the Design is finally checked and revised several times, the manufacturing process starts, the Gerber files and the NC drill files are generated to be sent to a special manufacturing company. According to the manufacturing company, fine-pitch boards should include a solder mask to allow for soldering the fine-pitch components without mistakes, thus the board is produced with a solder mask. Also, the first prototype had a silkscreen which enables to identify the common components easily for making commissioning possible in the beginning.

4.4. Testing procedure

This stage follows sending the PCB design to the manufacturing company; it is composed of developing a simple theoretical testing procedure aiming at a preliminary or a general test of the PCB, regardless of specific issues that may appear later. This is regarded as theoretical debugging scheme for the designed PCB. This is important to provide all the material (devices) needed to apply the board commissioning later.

The developed test was divided into five logical steps:

4.4.1. Testing the power points;

This means validating the generation of the power needed by all the circuits in the PCB; as explained before, the circuits in this FPGA PCB needs three values of voltage; 1.2V, 2.5V and 3.3V, these are being generated by the voltage regulator circuit. Thus, these three output voltage are accurately identified in the regulator circuit in order to make sure of their accurate generation to avoid any risk on the critical ICs; mainly the FPGA and the PROM.

Then a simple short circuit test is to be done between these power points and the relevant pin in each IC. I order to perform this; the power pins in each IC should be identified especially in the FPGA, this is summarized in Table 6

Table 6: The detailed PROM and Spartan-3E TQ144 package power pins.

	FPGA		PROM
Voltage level	**Bank**	**FPGA Pin number**	**PROM Pin number**
1.2V	0	P115	
	1	P80	
	2	P45	
	3	P9	
2.5V	0	P137	20
	1	P102	
	2	P65	
	3	P30	
3.3V	0	P128, P138	18, 19
	1	P79, P100	
	2	P42, P49, P64	
	3	P13, P28	

4.4.2. Testing the CLOCK signal;

The external clock oscillator used in the PCB provides a 100MHz clock signal as explained before, this clock signal is important for synchronization of data and control actions sent by the FPGA. Thus, the output of the crystal oscillator should be tested using

an oscilloscope (CRO) to verify that the clock signal is generated correctly with 100 MHz frequency.

4.4.3. Configuration signals;

The JTAG signals responsible for the Master-serial configuration process should be carefully analyzed on the oscilloscope and compared to the configuration signals in the reference design (i.eBasys2 commercial PCB). These signals are easily accessible on the memory pins and the JTAG pins in the developed design, these signals are specifically; TMS, TCK, TDI, TDO, PROG_B, INIT, DONE.

4.4.4. Connecting the PCB to the computer;

After checking that the board is powered correctly and that the important signals are generated in a logic way, the board JTAG connector should be connected to the ribbon-cable which is connected to the PC through the download cable/programmer interface (i.e Platform Cable USB), the result of this step is to see the STATUS LED of the programmer turning Green color and the relevant software (i.e iMPACT^{TM}) detects the presence of an FPGA and a memory because those are the configurable devices available in a chain as previously explained.

4.4.5. Configuring the FPGA and the PROM

As a final step, the FPGA and the memory should be configured, this is done following the configuration process steps discussed in the FPGA Configuration of the FPGAsection in the previous chapter; basically, after initializing the Chain successfully via the iMPACT software, the FPGA and the memory can now be configured, this is done first for the FPGA; the simple test .bit file generated by the ISE^{TM} software is downloaded into the FPGA by assigning a new configuration file, then using Program option, the FPGA is programmed, the iMPACT will indicate whether the process was done successfully as shown later, then repeating the same process for the platform flash PROM, but this time by downloading the ***.mcs*** file generated from the previously used .bit file into the memory. Finally, the PCB is switched OFF, the mode jumper is changed to the memory configuration position, and then the PCB is repowered and tested to give the expected signals on the output connector.

4.5. Analogue to Digital Module

This section explains the work done on the Analogue to Digital converter (ADC) developed to integrate the functions of the designed FPGA-based digital controller.

It is clear that any digital controller needs at some point in the system an analog to digital converter to convert the measured analog signal, fed into the controller and acting

as the feedback of the closed-loop control, into a digital signal able to be processed by the digital controller to produce the required control action.

There are two analogue to digital converters (ADC) needed in the control system; one is intended to convert the measured voltages and currents needed by the MMC control loops, this is composed of two eight-channel parallel-bus 12-bit ADCs, its output data is managed by the central control unit, **Z-7020** from Xilinx ´s Zynq-7000SoC [3], which is responsible for processing the MMC control, this was developed as part of another project. The other ADC is intended to convert the measured current needed by the DAB control scheme and also the voltage of the MMC cell capacitor needed to generate the PWM signals sent to the related MMC cell, this is composed of one-channel serial (SPI) 10-bit ADC (thus two circuits are needed), the output data of this ADC is processed by the Slave FPGA to apply the required control scheme of the DAB and send the required PWM signals to the MMC cell.

This thesis focuses on the design and implementation of the ADCs related with the DAB modules and the MMC PWM signals and sending information to the slave FPGAs.

The location of the ADC is preferred to be on the DAB board for two reasons;

First, in order to keep the FPGA PCB a fully digital circuit without complicating the design with mixed signal requirements and limitations as well as new design requirements, including having different ground planes and paying lots of attention to noise imposed over analogue signals to main clean the critical signals needed for accurate analogue to digital conversion process, also, the design gets more crowded with components due to the need for two ADC circuits which decreases the possibility of two-layer design.

Second, in order to allow for developing the ADC circuit separately and test it, and then place it on the DAB board with confidence that it has no problems in itself.

Two one-channel ADCs are needed, one for the DAB input current and one for the MMC cell voltage, the role of each is discussed separately below.

4.5.1. Role of the DAB ADC in the system

The role of the DAB related Analogue to digital converter is to convert the analogue data measurement of the current sensor at the input of the high voltage side of the DAB into digital signals in order to be sent to the FPGA for processing and control signals generation.

In other word, the main role of this ADC module is to copy the image of the DAB to the FPGA, so that the FPGA can take the control decisions and generate the switching signals needed to achieve its commanded power transfer from this DAB module.

4.5.2. Role of the MMC ADC

The function of the MMC-related ADC is to convert the analogue measured cell voltage of the MMC into a digital signal able to be sent to the FPGA to be processed resulting in generating the adequate PWM signals sent to the two switches of the relevant MMC cell (MMC cells are formed of half bridges) to achieve the required cell voltage. An ADC is needed for each MMC cell, this ADC is also placed on the DAB module connected to each MMC cell.

4.5.3. Design and implementation

In the beginning, a commercial AD module from Digilent (Pmod AD1) was used to achieve the role of the DAB-related ADC, this module is shown in Figure 30, this module consists of two independent circuits; each circuit consists of a 12-bit 1-channel AD converter chip along with anti-aliasing filters and an output buffer. The AD1 uses the SPI/MICROWIRE^{TM} serial bus standard to send the converted data [51].

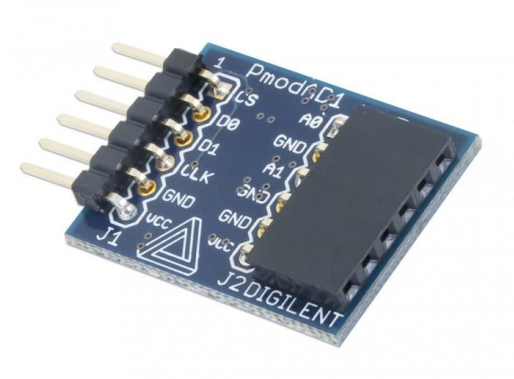


Figure 30: The commercial AD module PmodAD1 [51].

However, this ADC should be designed and implemented to be included on the DAB module rather than using the commercial PCB, this is for several reasons;

First, the commercial *Pmod AD1* has a special input and output connectors adapted to allow it to be connected to the commercial FPGA PCBs, however, this is not the case now because the FPGA PCB will be designed to be integrated on the DAB using a DIMM socket with the least components included on it, thus it receives all the needed signals through the edge connector.

Second, putting this type of connector on the DAB in order to connect this AD module is not a feasible option in terms of reliability and space, space because as shown in Figure

30, the male connector J1 is the output connector, thus a female connector similar to J2 should be placed on the DAB, which consumes a relatively significant space and is a bad choice in terms of reliability because this type of connector is loose and doesn't have any means to prevent it from being disconnected, so for the previously mentioned target SST prototype it is not acceptable.

Third, the cost of the components included in this AD module is cheaper than the cost of the commercial PCB, thus in terms of cost, it is much better to place this cheap circuit of the DAB module instead of buying the commercial one (24 AD1 modules are needed for the whole 24 DAB boards).

In order to be able to design the ADC needed for the previously explained function, the specifications of this ADC should be clear with respect to the DAB requirements. It is obvious that there is a need for isolation between the primary side and the secondary side of the DAB, however, no signal is needed to be measured from the secondary side of the DAB yet to maintain the intended control scheme, so, both ADC circuits can be referred to the ground of the primary side of the DAB without any need for isolation or use of optic fibers or digital isolators.

Consequently, the design of this ADC circuit is simple and could be directly implemented taking the commercial PCB AD1 schematic as a reference. In the beginning a separate test board is designed for only one converted signal, thus the components required in the PCB are;

1- One ADC 1-channel chip, the available ADC used is a 10-bit ADS7867 Texas instruments chip.
2- One anti-aliasing filter.
3- A buffer IC
4- An input connector with two pins, one for the sensed analogue signal and one as a ground.
5- Output connector with six pins similar to the reference design in order to enable connecting it easily to the available commercial FPGA PCB for testing its functionality, however, only 5 pins are actually used, chip select (CS), converted signal, CLK , ground and VCC.
6- A protection diode; to protect the ADC from input analogue signals of voltage higher than the maximum voltage that could be handled by the ADC chip which is 3.3V, its function is to clamp the input signal to 3.3V if it exceeds that.

4.5.3.1. Schematic

The schematic of the test ADC board is shown in Figure 31, it is clearly composed of power connections associated with some decoupling capacitors feeding the ADC chip, the buffer chip and the protection diode. The schematic is a chain like from the input

connector to the output connector (i.e the connector taking out the converter signal), first comes the protection diode connected to the input signal line, then the anti-aliasing filter is formed by two operational amplifiers along with their adapting resistors and capacitors, the resulting signal goes to the input of the ADC chip which is fed by the clock and the CS signal resulting in a converted digital signal.

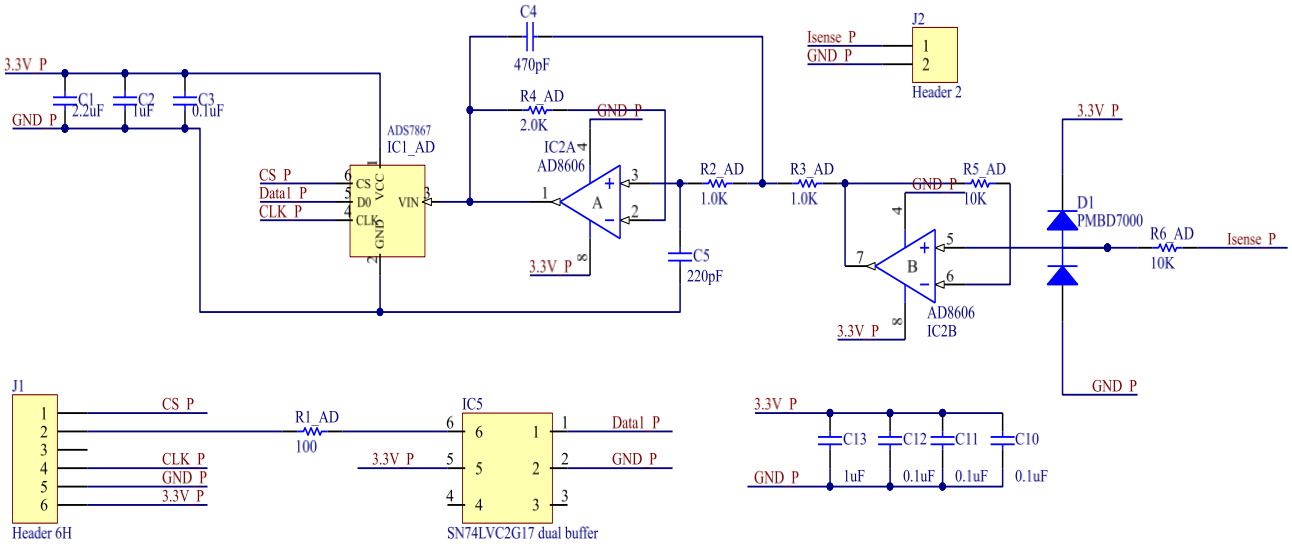


Figure 31: The Altium Schematic of the developed ADC.

4.5.3.2. Layout design

The layout design is developed following the components placement of the reference design, the resulting layout is shown in Figure 32, it is clear that the design is very simple with few number of components, thus enabling easily testing the test PCB.

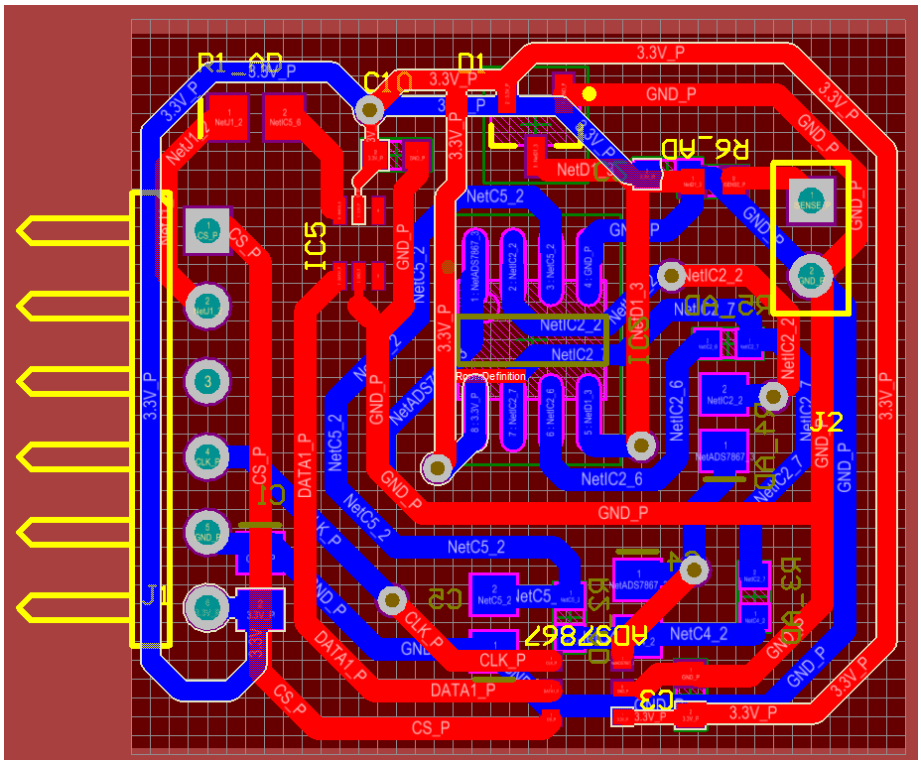


Figure 32: The developed Altium Layout Design of the ADC.

4.5.4. Testing

The developed PCB is shown in Figure 33.

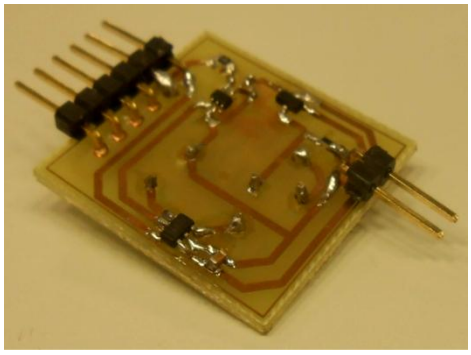


Figure 33: The developed ADC PCB.

Testing of the simple developed ADC is done using a program that displays the number of bits corresponding to the analogue signal converted by the ADC. In order to test the ADC for its full range; the input of the ADC is connected to a potentiometer giving a range of voltages from 0 to 3.3V.

Since the ADC IC used has a resolution of 10 bits, thus the displayed number of bits for the previously indicated voltage range should be varying between 0 and 2^{10} bits, this could be simply given by Eq. 1.

$$\frac{\text{Resolution of ADC}}{\text{system voltage (VCC)}} = \frac{\text{ADC reading}}{\text{Analogue voltage measured}} \quad (1)$$

For example, if a 2V is on the input, then the ADC should display an output of about 620 bits. The precision of this ADC depends on the ADC converter chip dynamic characteristics and sampling dynamics stated clearly in its datasheet [45].

The ADC was fully functioning giving the range of expected bits varying with the varying input potentiometer voltage. Thus, the schematic of the developed PCB can be easily copied to the Dab schematic with confident that it will perform its functionality.

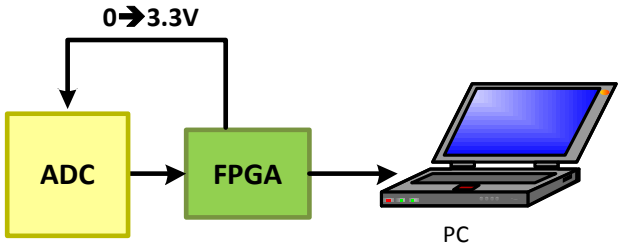


Figure 34: Testing the ADC.

4.6. MMC cell voltage adaptation circuit

The MMC cell voltage is measured using a voltage sensor (lv 25-p) this sensor outputs the measured voltage as a current signal, thus to be able to convert this analogue signal to a digital signal using the previously discussed ADC. Therefore an adaptation circuit is needed to change the output current value to the corresponding voltage value.

This is a very basic circuit usually done using a resistor of calculated value based on Ohm's law, it is important to include also an operational amplifier. Thus the connection of the circuit is as shown in Figure 35.

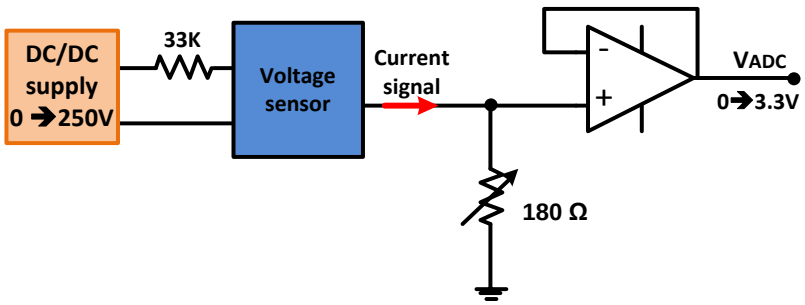


Figure 35: The adaptation circuit of the MMC cell voltage.

However, observing the structure of the ADC circuit, it already has an operational amplifier in the input filter, also observing the structure of the ADC IC from its datasheet [45], it has an Operational amplifier in the input sensed signal, thus the circuit can be simply a resistor and the voltage on this resistor is the measured MMC cell capacitor.

The resistor value should be very accurately calculated in order to protect the ADC and the DAB module from any possible problems. In order to calculate the value of this resistor, the maximum attainable MMC cell voltage measured on the output cell capacitor should be clearly identified; in this case it is 250V.

Thus, the resistor is calculated as follows;

$$V_P = I_P * R_P \tag{2}$$

$$I_P = \frac{250V}{33K\Omega} = 7.57mA \tag{3}$$

$$I_S = I_P * 2.5 = 18.93mA \tag{4}$$

$$V_S = I_S * R_S \tag{5}$$

$$R_S = \frac{3.3V}{18.93mA} = 174.24\Omega \approx 180\Omega \tag{6}$$

However, the controlled nominal MMC cell capacitor voltage is 200V, thus the range of input voltage to the ADC shall be always between 0V and 2.73V, which is a good accuracy and safety margin for protecting the ADC as well as the FPGA.

Then finally, in order to realistically test the obtained value, the voltage sensor is fed by an input signal coming from a variable DC supply and the output signal is connected to a 180 Ohms resistor, which is also connected to the input of the ADC circuit, the output of the ADC is connected to a FPGA board and using the same program used for testing the ADC, it is easy to verify the functionality of this circuit as well by varying the input voltage of the DC from 0 to 250V and observing the corresponding number of obtained bits. This test is simplified in Figure 36.

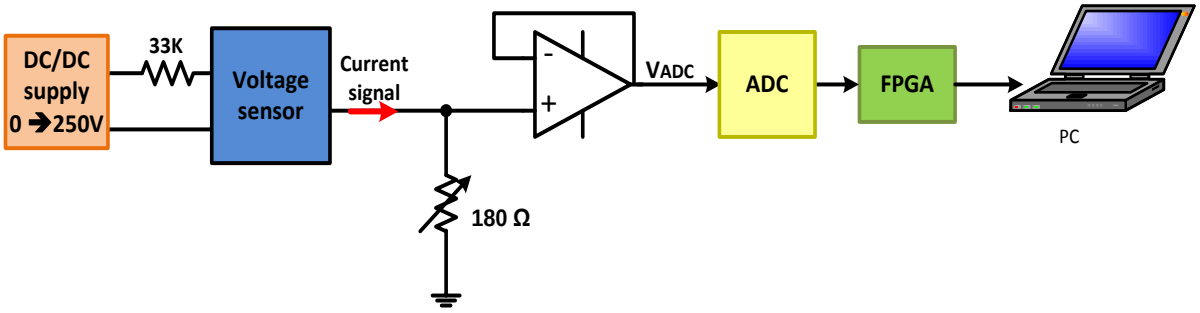


Figure 36: Testing the adaptation circuit with the ADC.

4.7. Communication with the central control unit[3]

4.7.1. Function of the communication link in the control scheme

The communication link plays an important role in the overall topology control scheme; it links the Central control unit with the slave FPGA units to achieve the functionalities of the distributed control.

Through this link, the Central control unit will transmit to the slave FPGA units the commands for the power that each DAB has to transfer, as well as the values of the duty cycle corresponding to the MMC cells to be able then to generate the gate signals for the MMC switches, on the other direction, each FPGA slave unit sends to the central control unit the measured MMC Cell capacitor voltage to enable the central unit to perform the computations needed by the MMC control algorithm to realize the balancing of the cell capacitor voltages [3]. This can be clearly understood from Figure 37.

[3] All the work done on the communication link; the selection of the protocol, the practical implementation inside the Central control unit and inside the slave FPGA units and the testing of the selected protocol, is attributed to Enrique Blanco Rodriguez. This thesis work adds no contribution to this subject but introduces it to have a comprehensive overview of the control scheme involving the FPGA units.

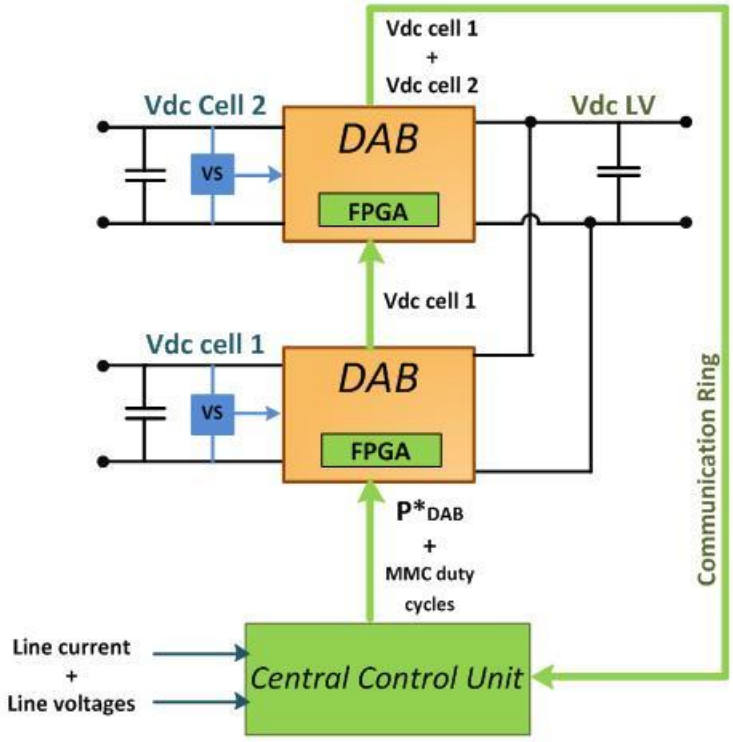


Figure 37: the communication ring linking the slave units with the central unit.

4.7.2. Selected protocol

The selected protocol was a modified version of TosNet protocol [46], this selection is based on several requirements and limitations;

Communication between the central control unit and the slave FPGAs will be bidirectional, thus two links are needed between each slave unit and the central unit, this complicates the hardware. These links are mandatory to be optical fibers for isolation requirements which increases the cost significantly, thus ring configurations are more suitable specially because of the stacked architecture of the DABs which makes the distance among DABs significantly smaller than the distance between the DABs and the central control unit thus decreases the length of the needed optical fiber links by including one transmitter and one receiver on each DAB module and thus connecting all DABs as well as the central control unit in a ring configuration using a serial communication protocol.

The communication should be able to provide the bandwidth demanded by the controller, the switching frequency of the MMC is selected to be 5 kHz [3], consequently, the information acquired by all the slave control units has to be transmitted in less than 200 μs, however, it does not need to transmit gate signals with precise timing thus a modest bandwidth is needed [3].

4.7.3. Theoretical implementation

Each ring consists of eight slave nodes, each corresponds to one FPGA slave unit in each DAB module (four MMC cells per arm, i.e eight cells per leg), and one master implemented in the central control unit [3].

Every node can tap into all communication, thereby making it possible for slaves to communicate directly with each other, while also preventing them from interfering with communication that do not regard them. Only those segments that are enabled are transmitted over the network, thereby increasing performance [46].

The setup process includes determining which node to use as master, finding out how many nodes are connected, and finally what registers are enabled in what nodes. This is done through a series of configuration packets [46].

4.8. Commissioning Process

In this chapter, the detailed procedure performed during commissioning the developed FPGA digital controller board (i.e putting it work) is analyzed and indicated step-by-step.

4.8.1. Debugging or trouble shooting

The debugging of the FPGA module PCB is divided into two levels, software debugging and hardware debugging; in other words, first studying carefully each signal trace in the Altium design and double-checking the footprints of all the components and second, powering up the board and proceeding with the theoretical test procedure trying to find out the problems in the designed PCB by testing all the important signals and comparing them to the commercial reference design. The developed PCB is shown in Figure 38.

Figure 38: the developed slave FPGA-based digital controller PCB.

4.8.1.1. Software debugging

The first version of the designed FPGA had some problems varying in their severity, these problems were detected by accurately observing the Altium design of the PCB and checking every circuit, every component footprint and every connected track, this step incurs a sufficient amount of time spent just studying the design accurately, the importance of this step is to try to avoid severe problems present in the design that can lead to destroying the design simply on the first power-up. In complex layout designs

spending time on the developed Altium design always results in yielding the existing layout and schematic errors. The addressed types of problems here are generally related with mirrored footprints, missing connections, possible misconnection of power and ground pins, misconnection of diodes, swapped components, possible misidentification of the first pin (orientation) of a certain IC, etc… The objective in good timelines is to have this stage finished before receiving the manufactured board because then hardware commissioning shall start by powering-up the PCB.

This debugging process resulted in the most severe problems detected in the PCB design, these problems are as follows;

1- Voltage regulator IC

Fortunately, one issue discovered before powering the PCB was that the voltage regulator IC was mirrored, this was the first severe problem detected in the PCB design. This was classified as a severe problem because this voltage regulator package has three ground pins, clearly all are connected together to the ground plane of the PCB, thus it is easy to imagine the possible risks if, due to the mirroring of the IC, the 5V supply was applied on one of these pins, obviously it would apply a 5V on the ground plane resulting in destroying most of the circuits especially the FPGA and the memory.

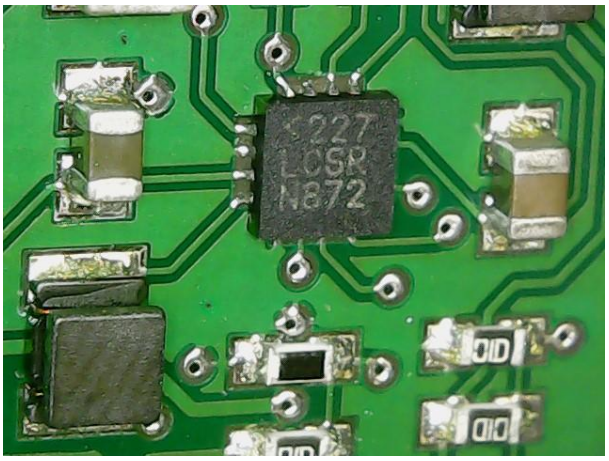


Figure 39: the voltage regulator LTC3545 pictured from the Slave FPGA PCB.

In order to fix this problem, clearly this IC should be removed or disconnected from all the system lines, then an alternative voltage supply should be used. However, it was risky to remove the IC because it is a 16-LEAD (3mm x 3mm) Plastic QFN package, (it is shown in Figure 39) and it has an under pad soldered to the PCB itself, thus hot air machine is required to be able to remove it, this will surely has many implications on the nearby circuits and components especially the FPGA, also adding a new debugging level in which every malfunctioning of the PCB will be attributed to the applied hot air, which is very complicated to analyze whether it is real or not. So, in order to avoid this risk,

disconnecting the IC from the rest of the system seems to be the less risk and more logical solution. Observing the schematic of the voltage regulator circuit, it is clear that if the three inductors, delivering the regulated voltage to the system, where removed, the voltage regulator IC will be safely disconnected from the rest of the system. This is clear from Figure 20.

Then, the problem of supplying the needed voltages had two solutions, first to connect three variable supplies to the output capacitor of the voltage regulator circuit and second to design an external voltage regulator circuit and connect its three outputs to the three capacitors of the previous circuit. The second solution seems less risky due to the inaccuracy of the variable DC supplies and the possible mistakes resulting in applying a high voltage directly to the FPGA pins.

2- JTAG reference voltage

Before powering the PCB and after deeply analyzing the datasheet of the chosen FPGA and the datasheet of the Platform Cable USB, it was concluded that, if the JTAG interface is a 3.3V interface which means that the Vref pin connected to the JTAG is supplied by 3.3V, then the High level of all the significant JTAG signals (i.e TMS, TCK, TDO and TDI) will be 3.3V.

The problem in that relies in the fact that the Spartan-3E FPGA JTAG port interface is powered by the 2.5V VCCAUX supply having an absolute maximum of 3V, thus it was impossible to apply 3.3V to these pins directly. A solution proposed by the datasheet to interface the FPGA JTAG pins with 3.3V system was to add series current-limiting resistors in all the input signals to the FPGA JTAG port which are TMS, TCK and TDI, however the TDO can directly drive a 3.3V system but with low noise immunity [25].

Looking at the schematic of the JTAG connection to the FPGA JTAG port, all the three previously mentioned input signals had an adequate value resistor except for the TMS signal.

The first solution to this problem was to add a current limiting resistor in the TMS signal line, however this was not easy to implement and also it had a risk of not totally guaranteeing the result due to not being able to know the most adequate resistor value to be added. When reading again the datasheet of the download cable, it was recognized that the Vref pin of the target device (FPGA PCB) can have any value between 1.4V to 3.3V, thus a 2.5V is possible to make the PCB recognizable by the Cable. Thus, the second solution actually lies in the fact that the platform Cable adapts all the signals to the provided reference voltage from the target system, which means that the HIGH level of all the input signals sent through the JTAG will have a maximum of 2.5V, therefore, eliminating the need for current-limiting series resistors in the input signals.

4.8.1.2. Hardware debugging

This is the second stage for the commissioning or the debugging of the FPGA PCB, this stage is intended to start by powering-up the PCB.

However, this stage starts after implementing all the solutions for the previously detected problems and making sure each solution is successfully implemented. This can be done following the designed theoretical testing procedure steps, previously explained.

The board was powered from a 5V DC supply observing the demanded current at the instant of powering the PCB to recognize if the current demanded is significantly more than expected or if the current demanded from the supply suddenly increased which in both cases would mean that there is a wrong connection or possibility that a component is damaged thus resulting in a short circuit. The demanded current was normal (i.e around 0.08) and the 3.3V LED indicated fine generation of the 3.3V signal. Then connecting the board to the computer through the Platform Cable USB the STATUS LED of the platform turned Green which indicates an adequate voltage on the Vref pin of the JTAG and thus indicates the presence of a target device. However, when trying to "initialize chain" using the iMPACT™ software *Boundary Scan* option, it didn't detect the FPGA nor the PROM.

At this point the hardware debugging procedure obviously starts. This process is a chain-like procedure; problems appear serially, thus each problem should be addressed then when solved the next step should be taken and possible errors could appear or not.

A- Initializing Chain

The first issue to address and analyze was why the PC didn't recognize the existing chain consisting of the FPGA and the PROM? This can be systematically analyzed following these steps;

First, verifying that the Platform Cable USB was configured correctly inside the iMPACT software to be sure it is doing its function, for doing that all the available configurations, in "Cable Setup" window, should be studied and, going back to the datasheet of the Cable, it is possible to know the adequate selection for all the configurations. As an example to this, the datasheet specifies a maximum configuration speed for the Spartan-3E FPGA family, this speed is 10MHz, therefore in the cable configuration the rate should not exceed 10MHz for a successful configuration of the FPGA. In this case the available choice was 6MHz.

Second, trouble shooting the configuration signals between the PC and the board. Basically there are two types of signals; signals sent from the computer to the input of the

chain and signals sent from the output of the chain to the computer, thus closing the configuration chain.

Since the iMPACT couldn´t detect the presence of a chain, thus it is more likely that one or more of these signals were not sent correctly. Consequently, all the signals transmitted through the JTAG connector in either direction are carefully analyzed.

In this case there is only one signal sent from the PCB to the computer via the JTAG; this is the signal coming from the output serial data port of the PROM (TDO_PROM) (output of the chain as shown in
Figure 6). Examining the sequence of the configuration chain, this signal is coming from the PROM TDO to the JTAG TDI labeled pin. This signal trace is accurately analyzed, and it was found that it is directly connected from the PROM to the JTAG and has the same signal as the commercial PCB, thus apparently nothing was wrong with it.

Thus following the other direction; the signal sent from the computer to the input of the chain is analyzed; this is sent via TDO labeled pin of the JTAG to the serial data input of the FPGA (TDI_FPGA). Examining this signal using an oscilloscope, no signal was found, looking at the trace in the PCB itself; it appropriately connects the JTAG TDO to the FPGA TDI through a current limiting resistor of 390Ω. In some cases current limiting resistor with inadequately high value causes degraded operation. The resistor is removed but the problem still exists. So, till this point the problem was not clear.

At this point a reference design plays a significant role; an idea was comparing the configuration chain to another commercial reference design PCB that uses the same *XCS250E* FPGA and is configured using a JTAG (i.e not using a microcontroller and USB as the commercial *Basys2* PCB).This is the *Core3S250E*from *Waveshare*^{TM}, shown in Figure 40.

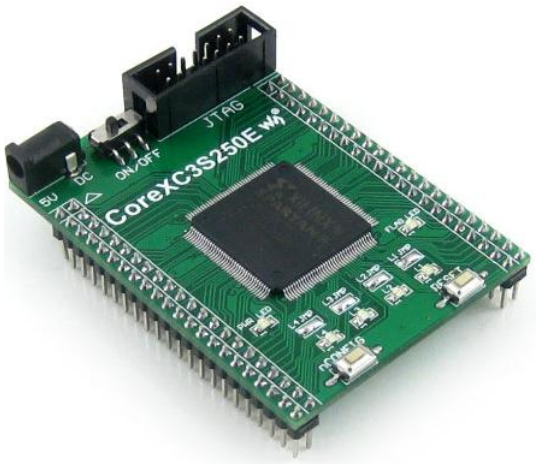


Figure 40: The *Core3S250E* from *Waveshare*^{TM} reference design with JTAG.

Deeply analyzing the schematic of this design and comparing it to the Basys2 design, several differences were detected, however, the most relevant difference was the

connection of the JTAG pins; in the Core3S250E the TDO of the JTAG was connected to the TDO of the PROM and the TDI of the JTAG was connected to the TDI of the FPGA as shown in the simplified scheme in Figure 41.This was oppositely connected in the developed design, the reason for that was that the naming of the JTAG pins is different from the rest of the chain components; TDI pin in the JTAG indicates the pin that should be connected to the input of the chain (TDI_FPGA) and TDO is the pin that should be connected to the output of the chain (TDO_PROM), which is a different naming with respect to the FPGA and the PROM; This is very clear in Figure 41 the TDO_FPGA is connected to the TDI_PROM because the TDO_FPGA indicates the serial data out of the FPGA so should be connected to the TDI_PROM which is the serial data input of the PROM, however, the TDO_PROM should be connected to the TDO_JTAG and the TDI_JTAG should be connected to the TDI_FPGA in order to close the chain.

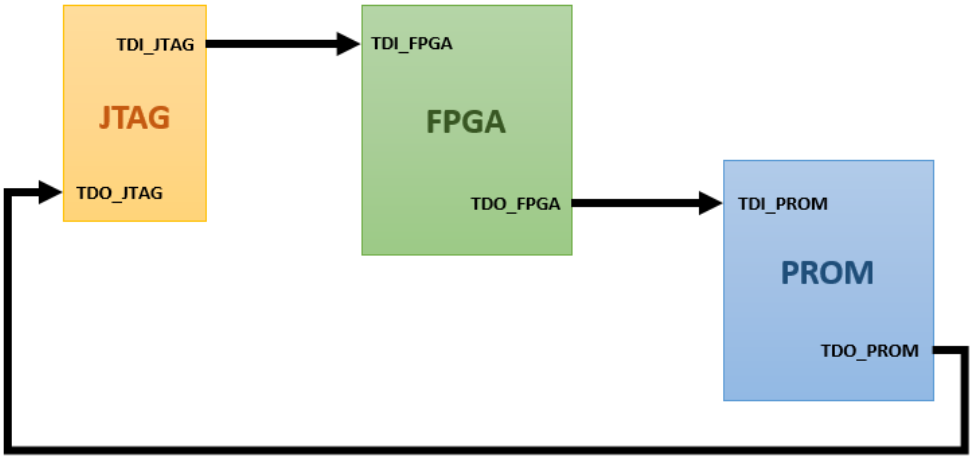


Figure 41: the simple configuration chain connection.

When swapping the two pin signals (TDO with TDI of the JTAG), the problem was solved and the iMPACT software detected the chain as shown in Figure 42.

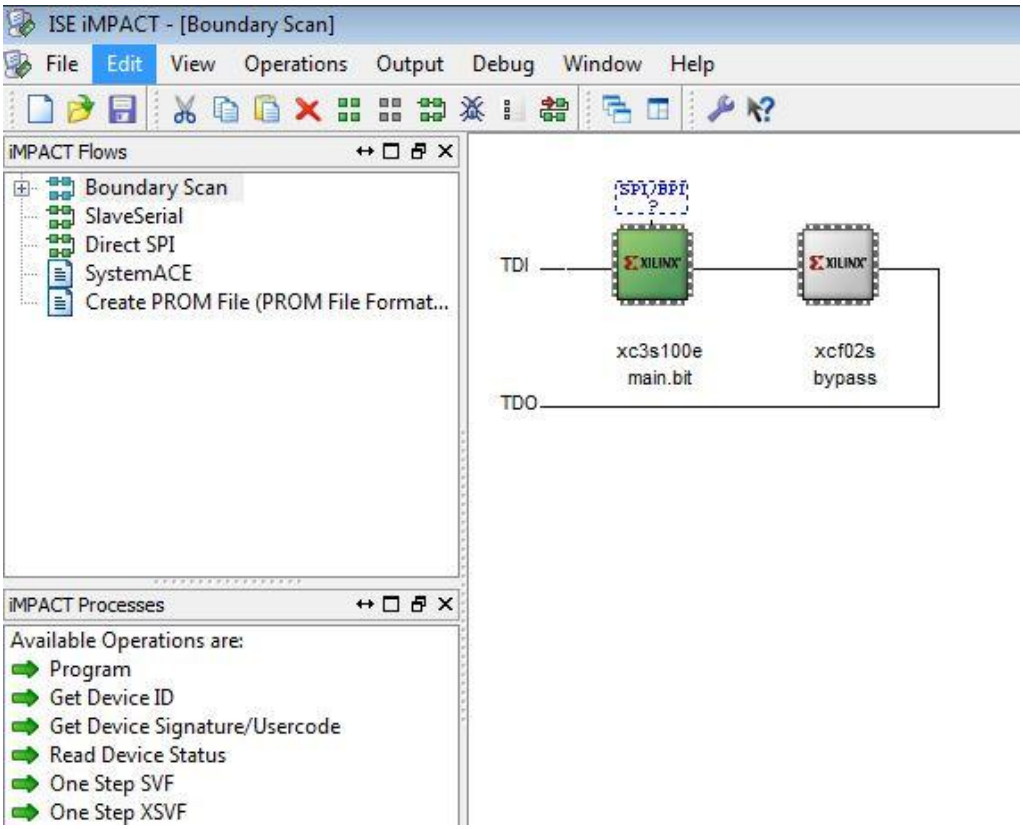


Figure 42: the chain is initialized detecting the presence of the FPGA and the PROM.

B- Configuring the FPGA

The procedure carried out for programming the FPGA was as follows;

1- A simple program was developed on the ISE™ software tool, this program is a simplified version of the DAB control program; it generates two PWM signals on two general I/O pins of the FPGA, this has no other use except testing the FPGA programming procedure. These two pins are identified on the edge connector of the PCB in order to see the PWM signals if the FPGA was successfully programmed.

2- The **.bit** file was generated from the previously developed simple test program, using ISE software tool, *Generate file* option. As shown in Figure 43.

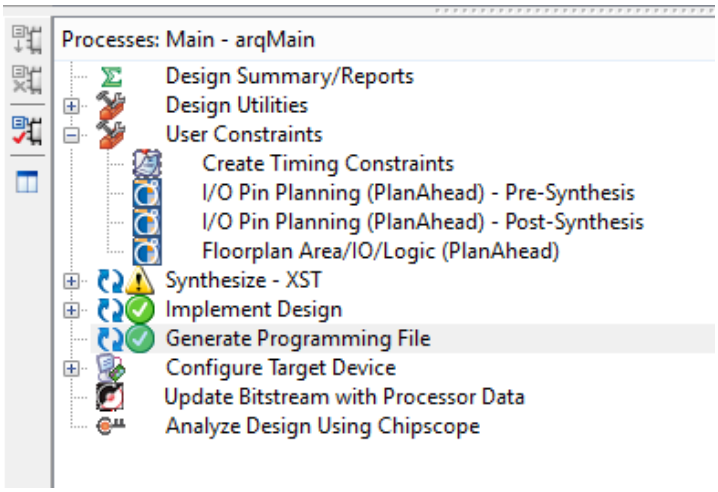


Figure 43: the Generate Programming file option, generating the .bit file.

3- The .bit file is loaded into the FPGA (xc3s250e), through iMPACT software, *Assign new configuration file* option.
4- The FPGA is programmed using *Program* option in iMPACT.

This procedure was successfully done without having any problems resulting in successful programming of the FPGA indicated by the iMPACT software, as shown in Figure 44.

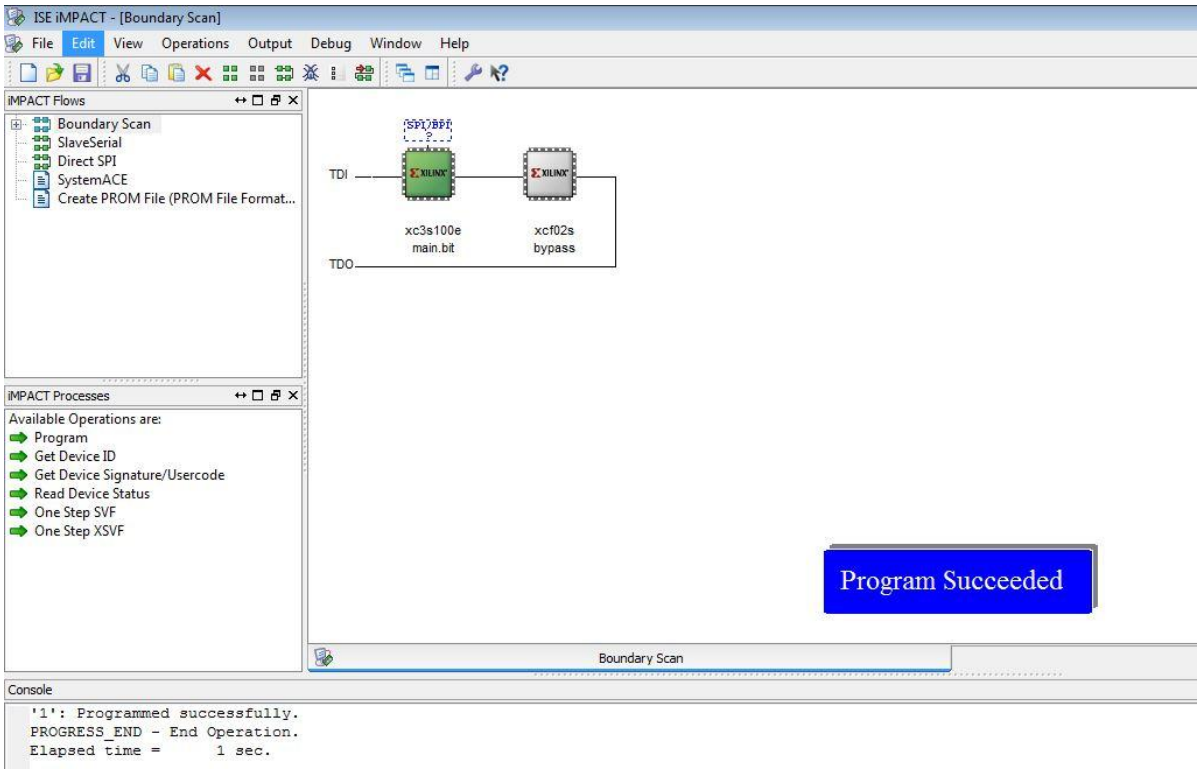


Figure 44: Successful programming of the FPGA indicated by the iMPACT software.

Also, it was verified by testing the two generated PWM signals on the relevant pins of the edge connector using an oscilloscope. A snapshot of one PWM signal is shown in Figure 45.

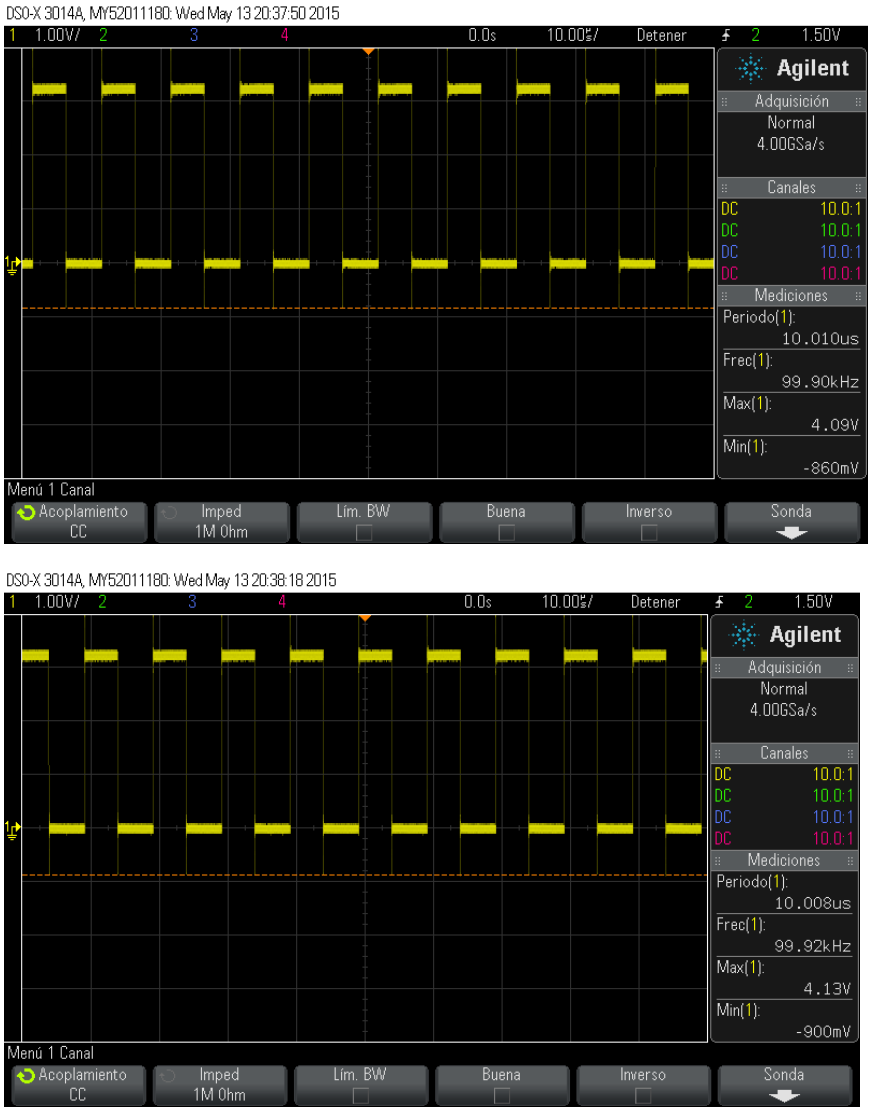


Figure 45, the PWM signals generated on the edge connector to test the configuration of the FPGA PCB.

C- Configuring the PROM

As previously discussed, the memory should be configured in order to retain a .bit file to avoid having to configure the FPGA on each power-on event.

The practical procedure carried out is as follows;

1- The .mcs file is generated from the previously generated .bit file using iMPACT software, Create PROM file option. As shown in Figure 46 , the three stages

needed for generating the .mcs file should be completed; first choosing PROM file generation, then specifying the memory used; xcf02s, finally naming the file and choosing the directory path of the .bit file.

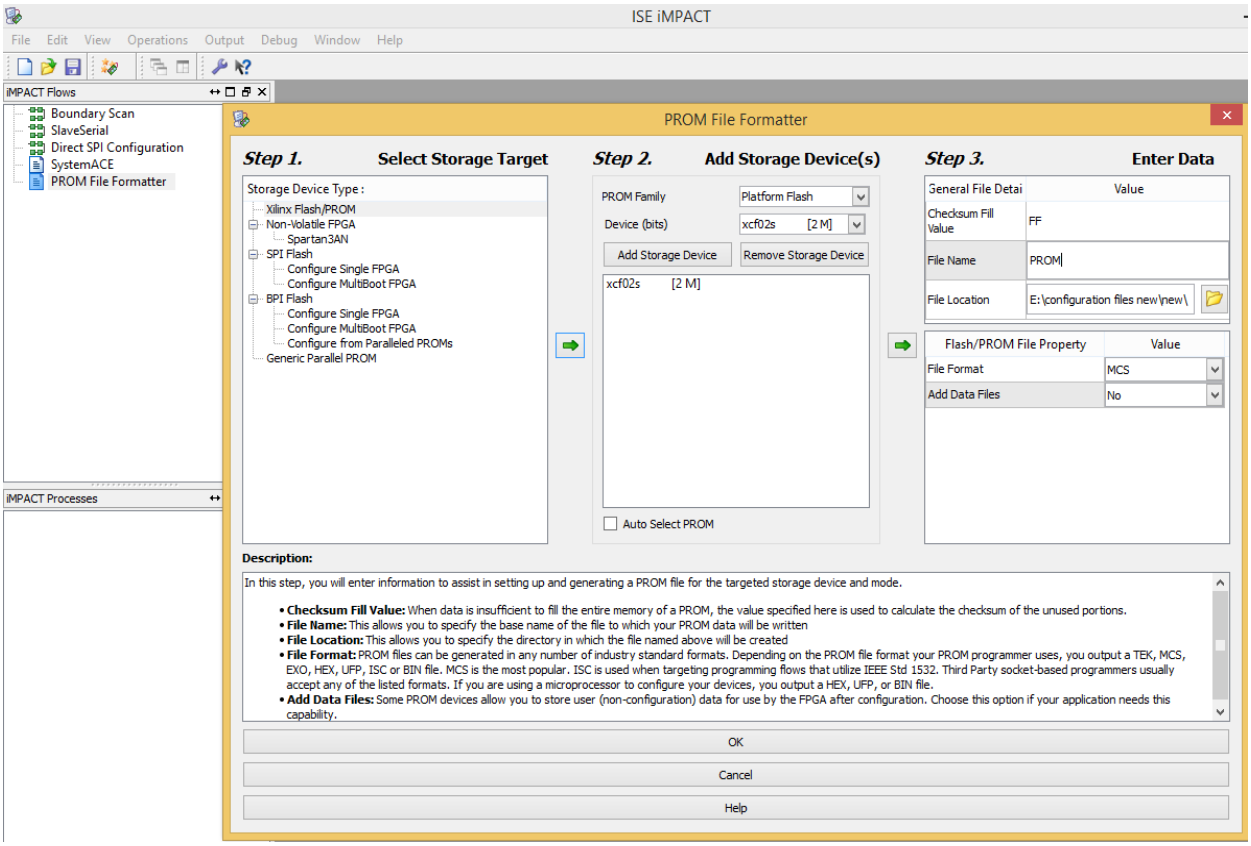


Figure 46: Generating the .mcs file to program the PROM from the iMPACT software.

2- The .mcs file is loaded into the memory detected in the chain using "*Assigning new configuration file*" option similar to the FPGA.
3- The memory is configured using "*Program*" option in the iMPACT.

This procedure results in failure of configuring the memory indicated by "*programming failed*" message generated by the ISE 11 software.

In order to fix this problem, the input signals to the PROM from the computer should be identified and accurately analyzed one by one. These signals are the TMS signal and the TCK signal.

Analyzing the TMS signal, its trace is directly connected to the PROM and the FPGA, Also, testing it using an oscilloscope and comparing it to the same signal in the Basys2 board, they are having exactly similar values, and thus there is no logical reason for this signal to be causing the problem.

Moving to TCK signal, which is the clock signal sent by the computer through the Platform download Cable. All the data sent from the computer software to the PROM and also the data sent from the output of the chain to the computer should be synchronized with the TCK. Thus the TCK signal is very important. Analyzing this signal, it was found that it is sent to a clock pin of the FPGA as well as to the clock input of the PROM, however this is done through a current limiting series resistor of value 390Ω, but after the JTAG reference voltage was changed to 2.5V, no current limiting is needed, this resistor has absolutely no other function in the system, on the other side it has a disadvantage of degrading the operation [48] by making a voltage drop in the sent signal which may cause it to be below the minimum voltage needed by the receiving device to sense the signal. Removing this resistor and repeating the same procedure again, the memory was successfully programmed indicated by the software as shown in Figure 47.

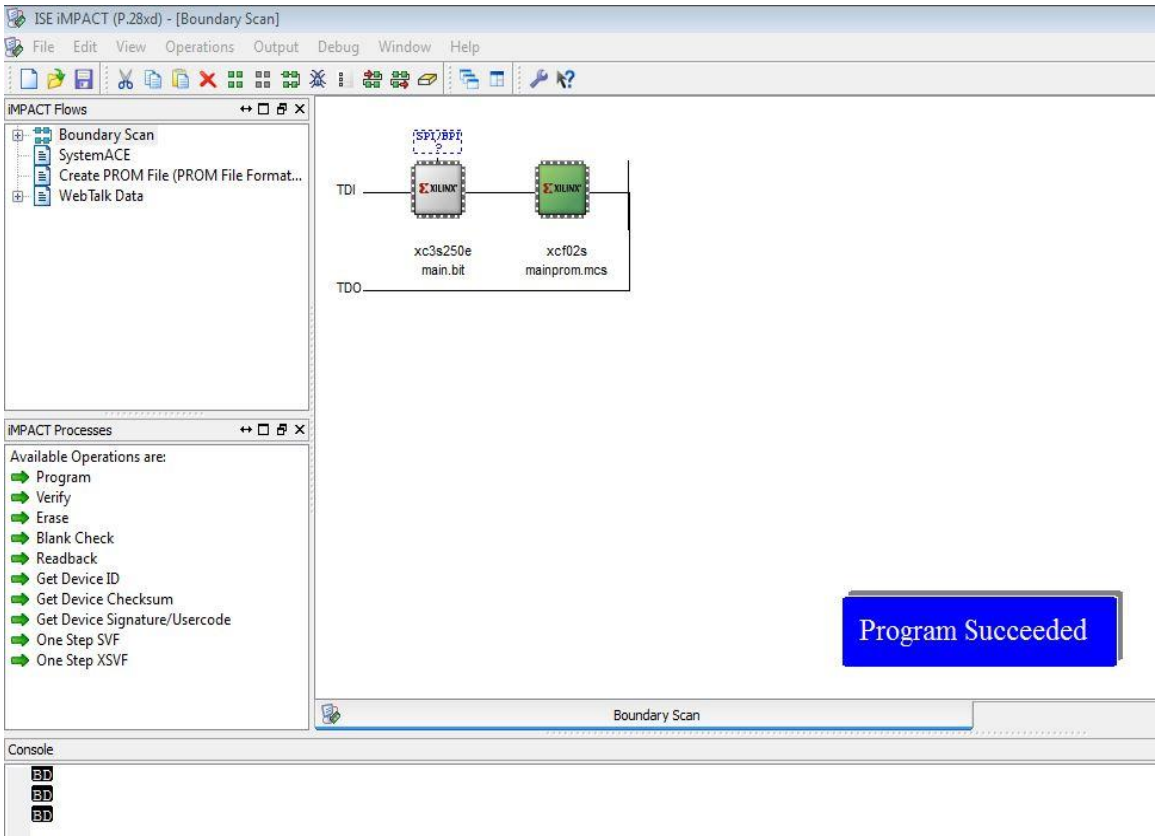


Figure 47: The PROM is successfully programmed.

D- Initiating FPGA Configuration via the PROM

The options for initiating FPGA configuration via the Platform Flash PROM include:

1- **Automatic** configuration on power up.
2- Applying an external pulse to the FPGA ***PROGRAM_B*** pin.
3- Applying the **JTAG CONFIG instruction** to the PROM.

In this application, it is interesting to automatically configure the FPGA on power up of the PCB due to the high number of stacked modules in the prototype, also there are no specific requirements for the configuration (e.g after a certain event), and therefore an automatic configuration is the perfect choice in this case.

After having configured the PROM successfully, there are no more steps needed regarding using the automatic configuration. The automatic FPGA configuration via the PROM can be practically tested by powering-off and removing the JTAG ribbon cable, then changing the jumper position to configuring the FPGA from the PROM, then powering-up again and examining the two edge connector pins where the PWM signals should be generated, however both were high (3.3V) signals which is the default chosen for any unused pin, which means that the PROM failed to be configure or load the program into the FPGA.

Many attempts were randomly implemented to fix this problem but all had the same result and the memory was not able to configure the FPGA; for example, a new .bit file was generated choosing different configuration option in the ISE tool and then the memory was reprogrammed using the corresponding ***.mcs*** file but nothing changed, finally a more systematic procedure was followed, looking at all the configuration lines between the FPGA and the PROM and identifying the most significant signals to debug.

The configuration lines between the PROM and the FPGA are inputs to the PROM from the FPGA and outputs from the PROM to the FPGA, each of these signals has a function and significance, and these are summarized in Table 7 and Table 8.

Table 7: The connection of the configuration lines going from the FPGA to the PROM.

From FPGA	To PROM	function
CCLK	CLK	**Configuration Clock**. Generated by FPGA internal oscillator. Frequency controlled by ***ConfigRate*** bitstream generator option.
DONE	CE	**FPGA Configuration Done**. Low during configuration. Goes High when FPGA successfully completes configuration.
TDO	TDI	**JTAG Serial Data** Output of the FPGA connected to JTAG Serial Data Input of the PROM. This pin is the serial Input/output for all JTAG instruction and data registers.
INIT_B	OE/RESET	Upon the completion of initialization and the beginning of configuration, INIT_B goes High, sampling the M0, M1, and M2 inputs to determine the configuration mode. Configuration data is then loaded into the FPGA.

Table 8: The connection of the configuration lines going from the PROM to the FPGA.

From PROM	To FPGA	function
CF	PROG_B	**Program FPGA**. Active Low. When asserted Low for 500 ns or longer, forces the FPGA to restart its configuration process by clearing configuration memory and resetting the DONE and INIT_B pins once PROG_B returns High.
DO	DIN	FPGA receives serial data from PROM's serial data output.

It is important to look at every signal using an oscilloscope. When evaluating each signal on the oscilloscope, all the signals indicated that the PROM failed to configure the FPGA; the *DONE* signal was always low and never turned to High which indicated that there was no *successful* configuration achieved, also the *PROG_B* signal is always High and never turned Low which means that no configuration process has taken place.

The most significant signals to analyze are the *CLK* signal and the *DO* signal, because of their function and also because all other signals are directly routed between the FPGA and the PROM. Thus these two signals are closely examined;

First, DO signal has a series resistor of value 10Ωand a capacitor connecting it to ground, shorting this resistor then trying again, it couldn't also load the program, thus the next step was removing the capacitor connected between this signal and the ground because its main function is cleaning the signal, but this didn't solve the problem and the PROM was not able to send the configuration data correctly to the FPGA.

Second, CLK signal has also a series resistor and a decoupling capacitor connecting it to ground, when shorting the resistor, the problem still existed, however, when removing the capacitor the clock was correctly sent from the FPGA to the PROM, thus the PROM was able to synchronize with the FPGA clock and successfully configured the FPGA and the two PWM signals appeared on the relevant pins of the edge connector, the logical reason is that the capacitor was delaying the clock signal sent from the FPGA to the PROM, so the PROM sent the data synchronized with a delayed clock, it is also indicated by the datasheet that poor CCLK signal integrity caused by ringing or reflections might cause double-clocking, causing the configuration process to fail [25].

At this point, the PROM is able to retain the program file configured into it and on power-up event it automatically loads the configuration data into the FPGA resulting in the correct configuration of the FPGA, consequently the PCB is tested to be fully functioning as expected.

4.8.2. Final debugging stage /finalizing the design

Finally, after fully debugging the FPGA PCB and making sure that it is fully operating as expected, an important step is needed which is to have an overview or a summary of all the modifications made which resulted in solving all the faced problems during debugging. The conclusion of this overview is to decide whether these hardware modifications are significantly severe to an extent that obligates the development of a new PCB including all these modifications and testing it before developing the 24 FPGA PCBs or not.

In this case, the modifications made were varying in severity, however, the most severe change was disconnecting the regulator circuit and providing an external voltage regulator connected to the PCB; this is considered a significant modification that pointed to the necessity of developing a test board before developing the 24 PCBs, this can be attributed to various reasons;

First, the voltage regulator circuit is a very important circuit in the design. Having a problem in the generation of the required voltages can lead to unexpected behavior or damaging or malfunction of both the FPGA and the PROM.

Second, any mistake or problem with this circuit is not easy to solve due to the necessity of providing these voltage (i.e 1.2V, 2.5V, 3.3V) externally which is not feasible for 24 PCBs.

Third, this circuit couldn't be developed separately with the available resources (without sending it to a PCB manufacturer) due to the used regulator package; which is a 16-LEAD (3mm x 3mm) Plastic QFN package, having the tiny pads under the package and including an under chip ground pad that is soldered to the PCB and that is not accessible by any means. Searching for the same regulator with a different package has been an option but no bigger package was found. Using three simple bigger package regulators was also regarded but it was not a space-wise option taking into consideration the available space for this circuit on the developed edge connected PCB.

Consequently, the output of this stage was the decision to develop another test PCB which should be the final result of all the debugging procedure, its main advantage is to eliminate any uncertainty of the existence of any other severe problem in the 24 module, which would take much time and effort to solve in the 24 module.

Thus, a new test PCB including all the modifications was ordered to be manufactured. After receiving the new prototype board exactly the same configuration procedure was repeated and FPGA was successfully configured, also the memory was successfully programmed and was able to configure the FPGA automatically on powering-up the PCB.

The developed board was integrated into the new developed DAB PCB, prepared with a socket for the slave FPGA controller integration[4], it was performing its function in the DAB in a proper way resulting in the expected operation of the DAB module. Also, it was linked to the central control unit through the implemented TosNet communication ring via transmitting and receiving connectors on the DAB module, it was also functioning as expected, sending to the central control the required data and receiving again the processed data from the Central unit.

Thus, the 24 FPGA slave units are confirmed to be developed by the manufacturing company to be integrated into the DAB modules.

Also, in order to make the FPGA controller board easy to be used and integrated in any design, specifically in the new DAB module, that has the special DIMM socket, the interface connector pins are well defined in a document. The is done by mapping all the FPGA pins on the interface connector pins indicating the unconnected pins of the interface connector and most importantly indicating clearly the input only pins and the dedicated configuration pins, because these pins can't be used as general user-defined I/O pin. Table 9 shows part of the document defining the 100 pins of the interface connector.

Table 9: Partition of the interface connector Pins Mapping document.

Top layer pins

connector	FPGA	description
1	P105	DUAL: Configuration pin, then possible user I/O
2	P103	DUAL: Configuration pin, then possible user I/O
3	P97	DUAL: Configuration pin, then possible user I/O
4	P96	DUAL: Configuration pin, then possible user I/O
5	P94	User I/O, or global buffer input (clk)
6	P93	User I/O, or global buffer input (clk)
7	P91	User I/O, or global buffer input (clk)
8	P87	User I/O, or global buffer input (clk)
9	P88	User I/O, or global buffer input (clk)
10	P83	User I/O or input voltage reference for bank
11	P53	User I/O, or global buffer input (clk)
12	P68	DUAL: Configuration pin, then possible user I/O

[4] The Dual Active Bridge module as well as testing the developed slave FPGA-based PCB was fully developed by Alberto Rodríguez Alonso.

Chapter Five

5. Conclusions and future work

This chapter is intended to give a conclusion of the work carried out in this thesis, also it presents the future work intended to be done complementing the developed project.

5.1. Conclusions

Analyzing the importance of the controller implementation in all power electronic converters, it was clear that many advantages are bound with the choice of the adequate control structure for each converter topology. Available controller implementations were discussed, especially the digital solutions. The conclusions from this discussion lead to the choice of the FPGA technology for the implementation of the slave unit controller of the addressed SST topology.

The problem has been defined, which is the need for developing an FPGA-based controller board with the least cost and the most area optimization fulfilling the needs of the SST topology for modularity, having one slave FPGA controller PCB per each module of the SST topology.

Certain specifications were regarded in order for the digital controller board to perform its role properly in the selected partial control scheme, such as, automatic configuration of the FPGA on power-up of the board. The main role of the slave FPGA was controlling the DAB converter, thus it is integrated into the DAB module.

Also, integrating the functionalities of the FPGA-based controller, an analogue to digital converter is developed and tested, however, it was placed in the DAB module (rather than on the FPGA PCB) due to the necessity for the FPGA PCB to be a digital only function-tailored board avoiding increasing complexity and wasting time in the design to follow the complicated mixed signal considerations.

The FPGA board was rather complicated to design due to the number of inputs and outputs involved in the design as well as the configuration chain restrictions and the miniature components included, however, several conclusions were clear after the design was developed and commissioned to be fully working:

- The FPGA package selection is critical for the layout design, it is closely related to the manufacturing company capabilities, the most adequate FPGA package for

ordinary design is the TQFN packages due to the fact that the pins on the edges of the package only.

- The density of the FPGA logic gates selection is critical for the application, the addressed topology control scheme should be analyzed carefully and the control task of the slave FPGA should be accurately defined and attempted to be loaded in the selected FPGA, it is good to leave a margin of unused logic units for future possible requirements or enhancements.
- The Master-serial configuration chain using a JTAG is a simple serial configuration structure that gives easy and fast FPGA and PROM configuration via any download cable. However, attention is needed regarding the JTAG, specifically, when connecting the serial input and output of the chain. Also, when voltage is chosen for the Vref pin of the JTAG.
- The FPGA-based PCB design can be optimized to have low cost and good behavior quite easy, due to the low number of components; only three components are the most expensive ones, FPGA, PROM and the voltage regulator.
- It is important to have more than one design reference using the same FPGA package and the same configuration structure.

5.2. Future Work

Future work can be divided into two perspectives;

5.2.1. Enhancing the developed design:

During developing the design and debugging the PCB, certain choices may appear to be very good choices while others may turn to be not adequate, thus resulting in possible enhancements in the design. Some proposed design enhancements are;

- The voltage regulator circuit; the idea of having the voltage regulator circuit directly connected to the FPGA and the PROM is a little scaring. Therefore, it was concluded that for the sake of the debugging stage, it may have been more adequate to have a technique to disconnect the voltage regulator from the rest of the board whenever needed, e.g mini switches at the three outputs of the regulator.

- Distribution of the general user-defined exited FPGA pins to the interface connector can be optimized for being integrated into the DAB application which has a symmetrical structure due to the two bridges. Symmetrically distributing the I/Os used to receive and send data to both bridges, can potentially enhance the PCB design of the DAB module by placing the FPGA PCB at an equal distance from both bridges thus all traces sending the signals to the gate drivers shall have the same length and thus will deliver the pulses efficiently without significant differences between them.

5.2.2. Integrating the Slave FPGA units into the SST prototype

This perspective is related with the future research work complementing this thesis work; after having the slave controller developed and tested to be functioning as expected, then the 24 developed FPGA PCBs will be integrated into the stacked MMC-DAB modules, composed of 8 modules per phase and thus 24 modules for the three-phase SST converter. The slave FPGA should be tested to be performing its role in the MMC cell voltage balancing as well.

Later, when building the full-scale SST topology the control structure will have more requirements, also the isolation requirements will significantly increase. The Slave FPGA is intended to be referred to the primary side of the DAB (consequently to the local MMC cell reference point), thus any signals sent to the secondary side of the DAB or to the communication ring shall be transmitted using Optic fibers.

References

[1] "Single-Stage Power Electronic Converters with Combined Voltage Step-Up/Step-Down Capability", PhD by Navid Golbon, the School of Graduate and Postdoctoral Studies, University of Western Ontario London, Ontario, Canada, 2013.

[2] SPEED Official Website, http://www.speed-fp7.org/

[3] M. López, A. Rodríguez, E. Blanco, M. Saeed, Á. Martínez, F. Briz, "Design and Implementation of the Control of a MMC Based Solid State Transformer," IEEE International Conference on Industrial Informatics INDIN'15, 22-24 July 2015, Cambridge, UK.

[4] Briz, F.; Lopez, M.; Zapico, A.; Rodriguez, A.; Diaz-Reigosa, D., "Operation and control of MMCs using cells with power transfer capability," *Applied Power Electronics Conference and Exposition (APEC), 2015 IEEE* , vol., no., pp.980,987, 15-19 March 2015

[5] Rodríguez, A.; Sebastian, J.; Lamar, D.G.; Hernando, M.M.; Vazquez, A., "An overall study of a Dual Active Bridge for bidirectional DC/DC conversion". IEEE Energy Conversion Congress and Exposition (ECCE), p. 1129-1135. September 2010.

[6] Rodriguez, A.; Vazquez, A.; Lamar, D.G.; Hernando, M.M.; Sebastian, J., "Different Purpose Design Strategies and Techniques to Improve the Performance of a Dual Active Bridge With Phase-Shift Control," Power Electronics, IEEE Transactions on , vol.30, no.2, pp.790,804, Feb. 2015.

[7] Habetler, T.G.; Harley, R.G., "Power electronic converter and system control," Proceedings of the IEEE, vol.89, no.6, pp.913,925, Jun 2001.

[8] Sepulveda, C.A.; Munoz, J.A.; Espinoza, J.R.; Figueroa, M.E.; Baier, C.R., "FPGA v/s DSP Performance Comparison for a VSC-Based STATCOM Control Application," *Industrial Informatics, IEEE Transactions on* , vol.9, no.3, pp.1351,1360, Aug. 2013

[9] T. W. Martin, S. S. Ang, "Digital Control for Switching Converters", Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE, 1995, pp. 480 -484 vol.2

[10] Prodic, A.; Maksimovic, D., "Design of a digital PID regulator based on look-up tables for control of high-frequency DC-DC converters," *Computers in Power Electronics, 2002. Proceedings. 2002 IEEE Workshop on* , vol., no., pp.18,22, 3-4 June 2002

[11] Y. Itoh, S. Kawauchi, "Easy Digital Control of Three- Phase PWM Convertor", 13th International Telecommunications Energy Conference, INTELEC, 1991, pp. 727 -734.

[12] Monmasson, E.; Idkhajine, L.; Cirstea, M.N.; Bahri, I.; Tisan, A.; Naouar, M.W., "FPGAs in Industrial Control Applications," *Industrial Informatics, IEEE Transactions on* , vol.7, no.2, pp.224,243, May 2011.

[13] "FPGA versus DSP design reliability and maintenance", article in July 25, 2007, http://signal-processing.mil-embedded.com/article-id/?2207=

[14] W. H. Wolf, FPGA-Based System Design. Englewood Cliffs, NJ: Prentice-Hall, 2004

[15] Electronics Weekly website; http://www.electronicsweekly.com/news/components/programmable-logic-and-asic/dsp-versus-fpga-2012-05/#sthash.o2YpySox.dpuf

[16] "Introduction to VHDL", Lesson 1, SEA Group, Universidad de Oviedo.

[17] "DSPs, Microprocessors and FPGAs in Control," article from RTC magazine: http://rtcmagazine.com/articles/view/100495.

[18] FPGA Block Structure, Xilinx Website, http://www.xilinx.com/fpga/index.htm.

[19] EEJournal Website, http://www.eejournal.com/archives/articles/20140225-rivalry/

[20] http://www.lbl.gov/Science-Articles/Archive/ALS-EUVL-sidebar.html

[21] NVidia chart from Semicon Japan, December 2013.

[22] EETimes, Website: http://www.eetimes.com/

[23] http://www.xilinx.com/products/silicon-devices/fpga.html

[24] "High-Speed Layout Guidelines," Application report, Texas Instruments, SCAA082– November 2006.

[25] Spartan-3E Family FPGA Datasheet, DS312, July 19, 2013: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf.

[26] "Basic FPGA Configuration," Xilinx's part 1 training guide.

[27] "Spartan-3 Generation Configuration User Guide," Extended Spartan-3A, Spartan-3E, and Spartan-3 FPGA Families, UG332 (v1.7) January 27, 2015. Available at: http://www.xilinx.com/support/documentation/user_guides/ug332.pdf

[28] Platform Cable USB, Product Specification Guide, DS300 (v3.3) June 25, 2014.

[29] Opal Kelly Product, https://www.opalkelly.com/products/lifecycle/

[30] Digilent Basys2 Board Reference Manual, Revision: November 11, 2010. http://www.digilentinc.com/Data/Products/BASYS2/Basys2_rm.pdf.

[31] Virtex 7family from Xilinx, http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html.

[32] Platform Flash In-System Programmable Configuration PROMs, Product Specification Guide, DS123 (v2.18), May 19, 2010.

[33] Linear Technology LTC3545 triple voltage regulator http://cds.linear.com/docs/en/datasheet/35451fb.pdf.

[34] Digilent Basys2 Board Schematic, http://www.digilentinc.com/Data/Products/BASYS2/Basys2_sch.pdf

[35] Spartan-6 FPGA PCB Design and Pin Planning Guide, Xilinx, UG393 (v1.3) October 17, 2012. http://www.xilinx.com/support/documentation/user_guides/ug393.pdf

[36] Multi-channel Silicon ESD Flow-Through Arrays Product Overview, SESD0802Q4UG-0020-060, www.te.com.

[37] "Avoiding PCB Design Mistakes in FPGA-Based Systems", **Taray, Inc.**

[38] Board Routability Guidelines with Xilinx Fine-Pitch BGA Packages, Application Note: Virtex Series, XAPP157 (v1.0) July 26, 2000.

[39] www.digikey.com

[40] Chip Scale BGA (CP132/CPG132) Package, PK373 (v1.3) June 25, 2008. http://www.xilinx.com/support/documentation/package_specs/cp132.pdf

[41] Xilinx Device Package User Guide, UG112 (v3.0) May 18, 2007.

[42] TQFP (TQ144/TQG144) Package, PK009 (v1.2) June 18, 2004. http://www.xilinx.com/support/documentation/package_specs/tq144.pdf

[43] Guidelines for Designing High-Speed FPGA PCBs, Application Note, Altera, February 2004, ver. 1.1.

[44] "Ground Planes," from Wikipedia, https://en.wikipedia.org/wiki/Ground_plane.

[45] ADS7867 analogue to digital converter with serial interface, Burr-Brown Products from Texas Instruments. http://www.ti.com/lit/ds/symlink/ads7866.pdf.

[46] Falsig, S.; Soerensen, A.S., "TosNet: An easy-to-use, real-time communications protocol for modular, distributed robot controllers," Robot Communication and Coordination, 2009. ROBOCOMM '09. Second International Conference on , vol., no., pp.1,6, March 31 2009-April 2 2009

[47] https://en.wikipedia.org/wiki/Transistor_count

[48] https://www.digilentinc.com/Data/Products/JTAG-HS3/JTAG-HS3_RM.pdf

[49] IP Processor Block RAM (BRAM) Block (v1.00a), Product Specification by Xilinx, DS444 March 1, 2011, available online in:http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf

[50] http://www.xilinx.com/support/answers/7598.html

[51] http://www.digilentinc.com/Products/Detail.cfm?Prod=PMOD-AD1