Master Thesis

Development of a wide band front end echo sounder receiver circuit

Jatin Sharma

Master of Science in Physics
Programme: Electronics

Department of Physics
University of Oslo
August 2015

Preface

With this Thesis, I am finishing my Master's Degree and during these years of my studies, I have learned a lot both theoretically and practically. I am quite satisfied with the knowledge and experience I gained in the field of Electronics, while working here at the Physics Institute, University of Oslo.

First and foremost i would like to thank my two supervisors, Dr. Helge Balk and Dr. Ketil Røed who helped me in selecting my Master Thesis in such an interesting topic of Hydro acoustics and guided me and gave me useful tips during my Masters to achieve my goal. It was very amazing that the courses which I have taken during my Master here at the Physics Institute were put into practical use during my thesis work and this helped me a lot to enhance my knowledge and learn new things.

A special thanks to the members of the E-Lab especially to Stein S. Nielsen and David M. Bang for providing the necessary electronic components, helping me with the hardware development and ordering the PCB board.

Since, my field of study Hydro-acoustics is a very broad subject. The special course 'Hydro-acoustics' offered by Helge Balk was very helpful in understanding the concepts behind propagation of sound in the sea and various phenomenons which take place during its propagation that need to be compensated in the receiver electronics. I tried to develop my echo-sounder considering the earlier efforts that other students had made and making the echo-sounder work as I desire. I learned a lot from the course FYS3220(Linear circuit Electronics) about the analog electronics section in my circuit and it helped me a lot to understand the various aspects of my design. Course FYS4220(Real time and embedded data systems) was helpful in understanding the digital part of my project.

I would also like to thank the faculty of Physics department for providing me the necessary lab equipments and my fellow students who were sitting besides me in the room 323, for helping me from time to time, when i needed them.

I am very grateful to my parents and the family members, my Uncle and Aunt living here who supported me with every thing during these years. Without their help, it would not have been possible. It was also a challenge for me to come to Norway and start again as a student.

Jatin Sharma

July, 2015

Abstract

The work which is documented in this thesis deals with the development of front end Echo sounder receiver circuit. The goal was to develop a system which can be used for wide band applications. The scope of this project could have been made limited to the development of front end receiver circuit only, but it would also be interesting to see other methods for transmission of pulses. Receiver electronics together with the transmitter have been designed on a single board to check the performance of the complete system.

The project begins with the introductory part describing about the background of Hydro acoustics and then description of various methods used in echo sounding and how they are implemented. Then, it illustrates the various options and tools available to achieve the desired goal. Design and development part provides information about how these specific methods are actually realized in practice. And at last, the developed system is tested to check the results and the concluding remarks are written.

While designing the Electronics, I tried to make the system quite flexible such that it can be operated on various frequencies.

A short description about the Sonar equations is given. An illustration showing how these equations relate to the loss of signal strength when a pulse propagates through sea water is provided and the corresponding challenges encountered by the receiver electronics in lifting this low amplitude signal.

New receiver circuit has been constructed which will help in classifying the various targets in the sea by processing the received echoes. Instead of using a front end resonance filter, a switch has been implemented. To handle the large dynamic range in the received echo, Variable Gain Amplifier has been used. It is used for providing the TVG with the help of DAC. Echo signals are converted into digital values using ADC. FPGA system has been used to provide the digital support which is further connected to the PC.

Apart from this, a different technique for generation of electrical pulses like use of MOSFET technology, is implemented in the transmitter electronics . The methods for amplifying the pulses are also reconsidered and SMPS technique is used. I introduced the possibility of easily varying the transmitted power from the transducer by doing simple modifications.

The designed system has worked quite well meeting the expectations. All the results which are shown in this project are obtained in the laboratory and further work is to measure the results in the sea.

Contents

Chapter 1

Introduction

- **1.1 Background**

  1.1.1 Hydro Acoustics

- **1.2 Echo Sounder**

  1.2.1 Earlier Developed Echo sounders
  - Pros and Cons

- **1.3 Digital Signal Processing**

  1.3.1 Programmable Logic- Background
  1.3.2 What are FPGAs?
  1.3.3 Hardware Description Language

- **1.4 Challenge and Motivation**

1.1 Background

The discovery of traveling of sound in water goes all the way back to the 15th century, when Leonardo da Vinci noticed that he can hear the sound from distant ships by inserting a long pipe in the ocean. In 1827, Daniel Colladon and Charles Sturm collaborated to measure the velocity of sound in water. They conducted this experiment in Lake Geneva, Switzerland. By timing the interval between a flash of light and the striking of a bell underwater, they determined the velocity of sound to a surprising degree of accuracy[6, Page 2].

In the Nineteenth century, phenomenon of "Transduction" was discovered which is nowadays extensively used for sending and receiving sound in water. With the beginning of the twentieth century, several attempts were made to develop such systems which could communicate or detect the targets in water and a number of devices were proposed which used the method of echo ranging. With time, the research expanded in this field of Hydro-acoustics and was used in the study of various targets like fishes, submarines, etc. Today, it is widely used in the biomass estimation, detection of fish school and differentiating between various species of fishes.

1.1.1 Hydro Acoustics

Hydro acoustics is the term associated with the study of sound in water and its applications. Sound travels as longitudinal waves in the form of compression and expansion of the molecules of the medium.This alternate compression and expansion phenomenon causes the sound to travel in the medium.

Speed of sound in water is approx. 1500 m/s and it depends upon depth, temperature and salt-content. Higher frequencies get attenuated, while the lower frequencies can travel long distances in water.

Active hydro acoustic sensing involves making a sound and listening for the echo. The intensity of received echo depends on the intensity of the transmitted wave(Source Level), the loss in intensity as the sound wave spreads in the water and absorbed by water(Transmission loss), the reflectivity of the target(TS), the position of the target in the beam and various losses in the instrument itself. All these processes are combined by a set of equations as:

$$EL = SL - 2TL + TS \qquad [6, Pg\ 23] \tag{1.1}$$

$$RL = SL - 2TL + sV + V \qquad [6, Pg\ 23] \tag{1.2}$$

These two Eq. 1.1 and 1.2 are the basic Sonar equations in the field of Hydro acoustics and one can calculate echo level(EL) provided the other parameters are given. These equations are the working relationships that combine the effects of the medium, the target and the equipment. These are very useful in echo sounder designing and also for predicting the performance of Hydro acoustic equipments.

Echoes from most underwater objects differ from the incident pulse in a number of ways other than intensity, as described by the parameter Target Strength(TS). The reflecting object imparts its own characteristics to the echo; it interacts with the incident sound wave to produce an echo that is, in general, different in wave shape and other characteristics from the incident pulse. These differences are useful to sonar engineers in two ways: they may be employed as an aid in detection, as in filtering with narrow-band filters to enhance an echo buried in reverberation; they may also be used to help in target classification to distinguish one type of target from another, as in distinguishing a submarine from a school of fish[6, Pg 322].

1.2 Echo Sounder

Echo sounder is based on the principle that water is a very good medium for the transmission of sound waves and that a sound wave will bounce off a reflecting layer, returning to its source as an echo. An Echo Sounder comprises of transmitter(or pulse generator), transducer, analog front end receiver and the digital processing unit. Earlier echo sounders had an indicator or graph recorder instead of Digital Electronics.

Echo sounders are basically SONARs(SOund Navigation And Ranging) i.e. if we take a simple single beam echo sounder and point it in various directions, we get a SONAR. Echo Sounders are more generally used as fish finders or depth sounders.

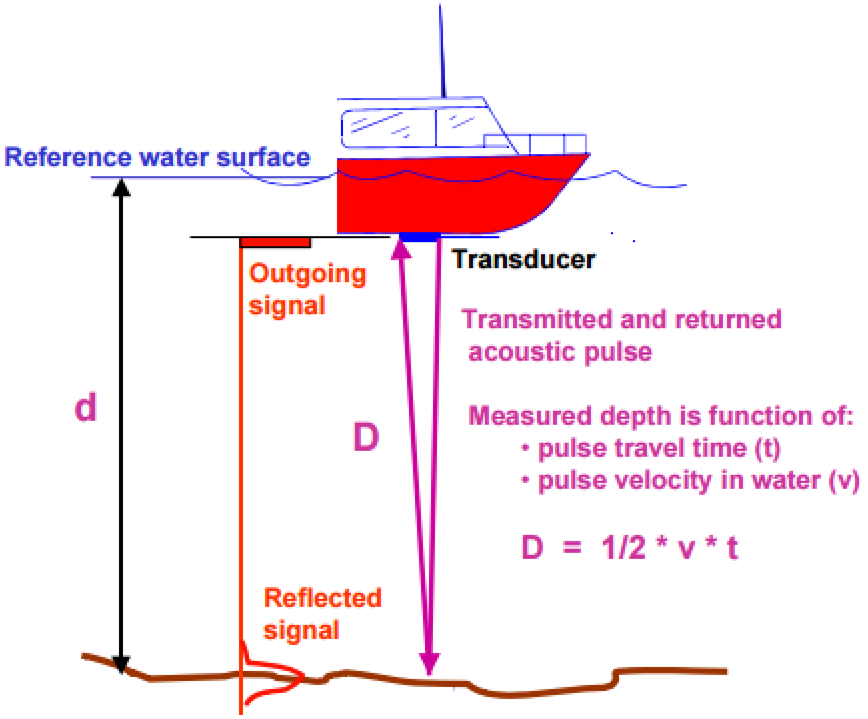


Figure 1.1: Echo sounder Principle[29]

From the expression D = $\frac{v * t}{2}$, one can calculate the depth of the sea bed or any other acoustically reflective bottom.

Transmitter is used for sending short pulses of duration τ(ms) of a particular frequency. The pulses repeat after certain time interval. Repetition Time is independent of the transducer frequency.

The duration of pulses should be large enough such that transducer can convert the electrical pulses into sound waves. The relation between pulse duration and bandwidth is given by :-

$$BW = 2/\tau \qquad [36, Pg. 22] \tag{1.3}$$

This relation is important while designing the echo sounder because the pulse width should be short but there is a limit. The bandwidth of the transmitted pulse must match with the receiver's bandwidth. There is a trade-off while selecting the echo sounder's bandwidth. It is desired to have a bandwidth large enough which can preserve the information in the received echo, and at the same time it is also necessary that the bandwidth should be as small as possible to limit the ambient noise entering from the sea.

High amplitude pulses from the pulser or transmitter part actuate the transducer. The transducer part is used for converting these electrical signals into sound waves during transmitting and vice versa while receiving.

In the receiver part, the echoes received are of very low amplitude as compared to transmitted pulses. Therefore, they need to be amplified. An echo returning back from 10m has a different intensity than one returning from 80 or 100m. As a result, Dynamic gain range must be set according to the depth of operation.

The traditional analog receiver part comprised of filter, amplifier, rectifier, envelope detector and then digital processing as shown below:

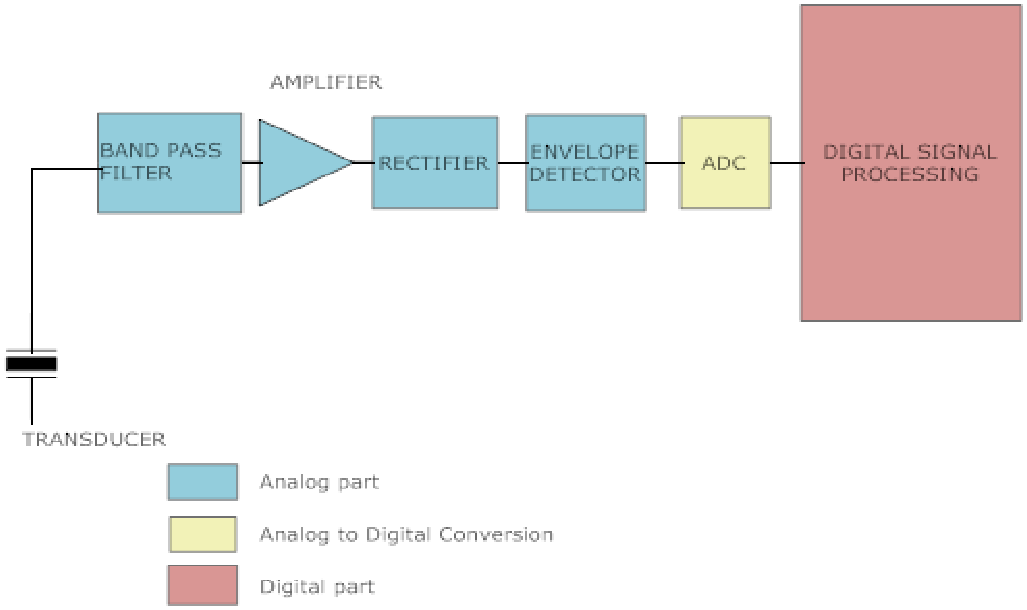


Figure 1.2: Earlier Receiver Part

There are certain losses associated with processes like rectification and envelope detection when they are performed in analog domain using op-amps and passive components like capacitors and diodes. A simple schematics of a full-wave rectifier using 2 op-amps, resistors and diodes is shown as:

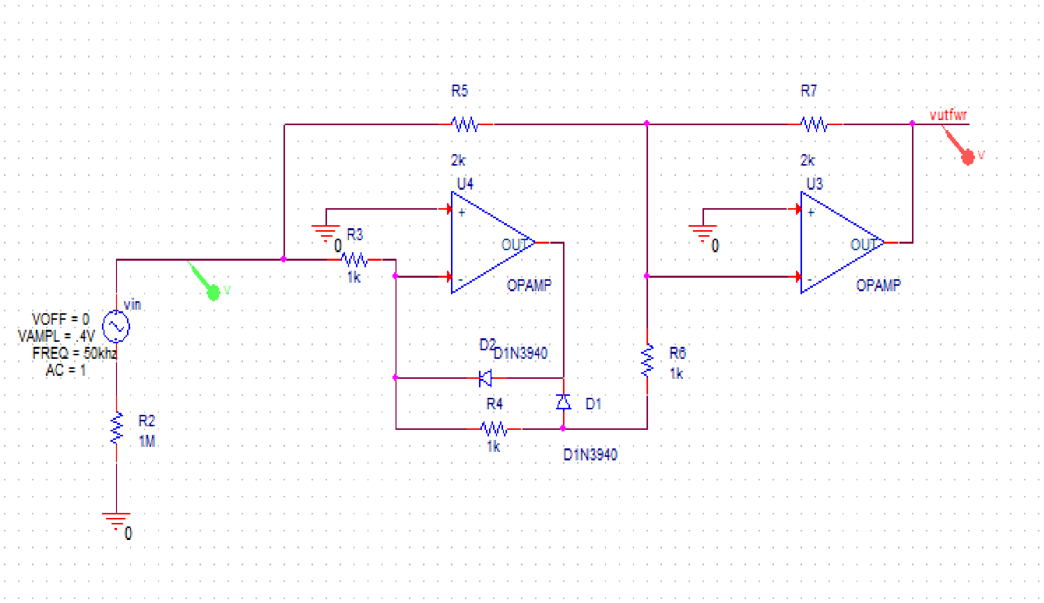


Figure 1.3: Full-wave rectifier

The gain of this rectifier can simply be changed by changing the value of resistor R7[53]. The corresponding simulation outputs of this circuit for 50 and 200kHz using PSpice(Attachment V) are shown here:

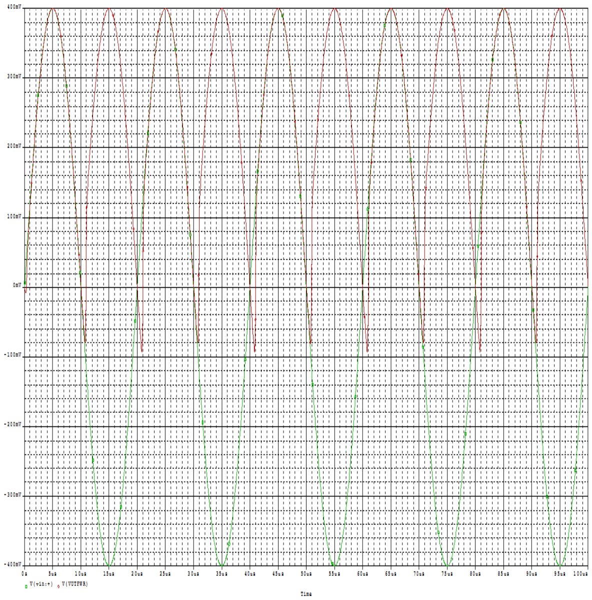


Figure 1.4: Simulation-50kHz

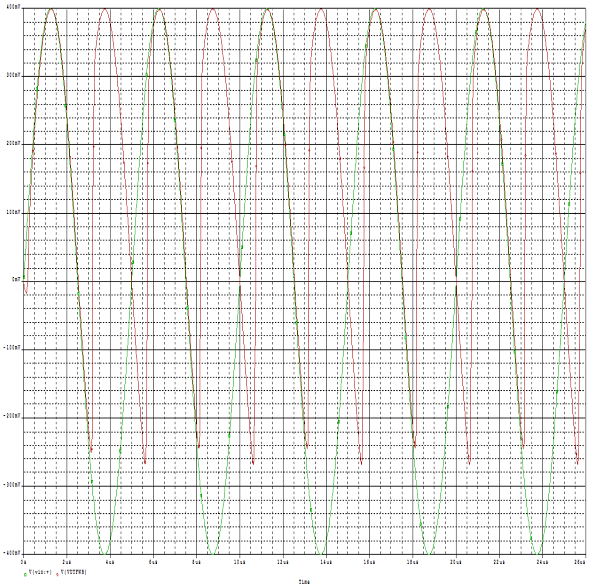


Figure 1.5: Simulation-200kHz

As seen from Fig. 1.4 and 1.5, distortion is present in the rectified signal and therefore, this operation is rather done in the Digital Electronics. Same is the case with envelope detection process which implements passive components like super diodes and capacitors.

1.2.1 Earlier Developed Echo sounders

The Hydro-acoustics group at Physics Institute, UiO has been working with the development of Echo sounder and SONAR technology from quite a long time. In earlier Echo sounders, Analog Electronics comprised a major portion. With time, the Analog Electronics began to be replaced with Digital Electronics.

Before starting the project, it is necessary that one should have a thorough knowledge about what has been done earlier in this field and to learn from others mistakes. This will help in improving the results, coming up with new and efficient solutions and saving time.

The analog front end receiver proposed by Morten Hellum[4] comprised of a transformer, 2nd order resonance circuit and a pair of back to back Zener diodes as the receiver network. This receiver unit had a high Q(Quality) factor and a Bandwidth of 4kHz. This was a very good solution and was used in the Analog front end by many students.

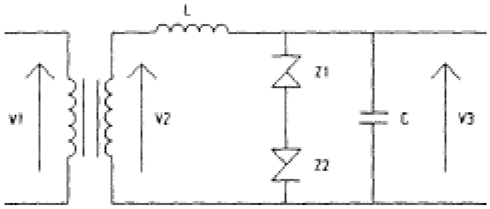


Figure 1.6: Hellum's Front End Receiver[4, Pg. 10]

This front end focuses on improving the Signal to Noise Ratio(SNR) of the received echo. The impedance caused by the Inductor at 50kHz is very large(2.5kOhm) and is much greater than the transducer impedance such that the load during transmission is virtually only constituted by the transducer[4, Pg. 18]. The Zener diodes together with the inductor protect the receiver amplifier during pulse transmission. The LC circuit also acted as a Bandpass filter and helped in removing the noise signals from the signal. On the transmitter side, Hellum describes about the use of Push-Pull Transistors for the generation of pulses.

The receiver hardware developed by Dr. Balk[36] gives outputs at 3 different channels. Each channel had a Bandpass filter with 4kHz bandwidth. All the channels had different amplifications for improving the dynamic range of the system with Channel 0 having 0dB gain for high amplitude signals, while Channel 1 and 2 had 50dB and 100dB Gain for comparatively low amplitude signals in the range of mV and μV respectively. Thus making the total dynamic range of the receiver to be

150dB. In this design, Hellum's front end[36, Pg. 64] was also implemented.
These echo sounders were designed to operate at a single frequency.

Pros and Cons

There were certain advantages and limitations in the previously developed systems and these pros and cons were kept in mind while designing the system.
The transmitter Electronics earlier developed was designed in combination with the transformer. Transformer connects the transmitter, transducer and receiver together, but a smaller and more efficient solution needs to be implemented which is only used for isolating the receiver side while transmission and get rid of bulky and space consuming transformer.

The earlier front end receivers implemented LC resonance circuit and were very efficient and simple in implementation. It was useful for improving the signal strength and at the same time provided protection against high voltages. But, these solutions were useful for narrow band applications i.e. they allowed only a specific frequency to pass through them. This project deals with the development of a Wide band receiver circuit. Therefore, the circuit must be designed for a wide frequency band operation which can work with different frequencies.

1.3 Digital Signal Processing

With the advancement in technology, the analog front end is merging within the digital electronics section and more and more processes now take place in the Digital domain.

1.3.1 Programmable Logic-Background

After the Analog Front End, there has to be a digital part which can process the signal and use it for displaying, saving or editing afterwards. Earlier in digital Electronics, the trend was to use a set of specialized discrete logic circuits, to obtain specific outputs. In order to create a slightly complex design, one needed to add few tens of such chips on a single board. This resulted in complex board layout and reduced performance. To improve performance, *programmable logic* method was introduced. The advantage of this method was its high speed of implementation[5, page 10].

Programmable Logic Devices(PLDs) started arriving in 1970s in the form of Programmable read-only memory(PROMs). PROMs can be visualized as devices consisting of a fixed array of AND functions driving a programmable array of OR functions. These devices were initially intended for use as memories to store computer programs and constant data values. However, they were also found to be useful for implementing logic functions such as lookup tables and state machines[1, Pg. 16]. A 3-input, 3-output PROM in its unprogrammed state is shown here.

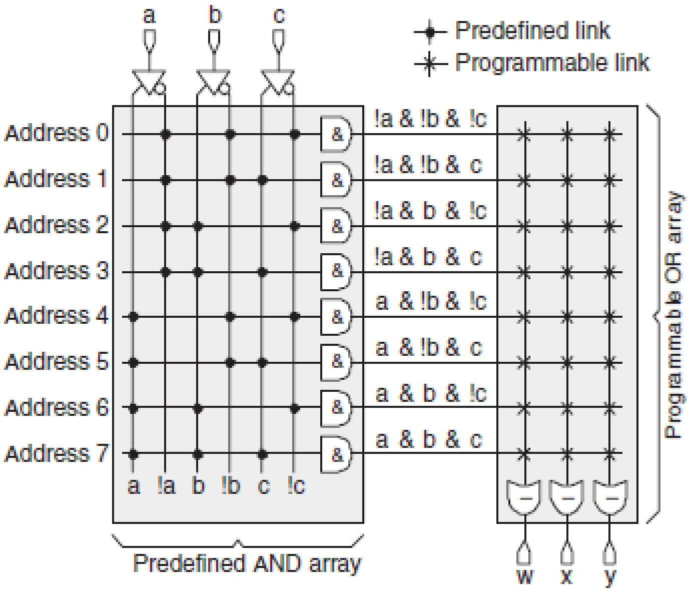


Figure 1.7: Unprogrammed PROM[1, Pg 30]

In order to step up the PLD ladder and address the limitations imposed by PROM architecture, Programmable Logic Arrays(PLAs) were introduced around 1975. These were the most user configurable of the simple PLDs as both the AND and OR arrays were programmable here. Unlike

PROMs, the number of AND and OR functions in their respective arrays are independent of the number of inputs to the device[1, page 33].

There were some speed limitations associated with PLAs. So a new class of device called Programmable Array Logic(PAL) was introduced in the late 1970s. Conceptually, PALs are almost the exact opposite of PROMs as they have a programmable AND plane and a fixed OR plane and usually include registers[5, page 12].

With time, there was a need to build bigger logic circuits and PLAs and PALs could not be implemented in large circuits because large number of fan-ins and fan-outs involved. Therefore, Complex Programmable Logic Devices(CPLDs) replaced them. CPLDs consist effectively of a number of PAL-like macro cells that can communicate through programmable interconnect[5, Pg. 12].

Another technology for implementing digital logic was introduced in 1980s which was based on "Gate Arrays" and was called Field Programmable Gate Arrays(FPGAs).

1.3.2 What are FPGAs ?

FPGAs are digital integrated circuits(ICs) that contain configurable blocks of logic along with configurable interconnects between these blocks. Depending on the way in which they are implemented, some FPGAs may only be programmed a single time, while others may be programmed over and over again. The term "Field Programmable" implies that the device is programmed by the customer i.e. in the "field" and not by the manufacturer. The advantage of using FPGAs over CPLDs is that the former can be implemented for complex designs and based on Look-up tables(LUTs) while the later ones are for simpler designs[1, Pg. 17].

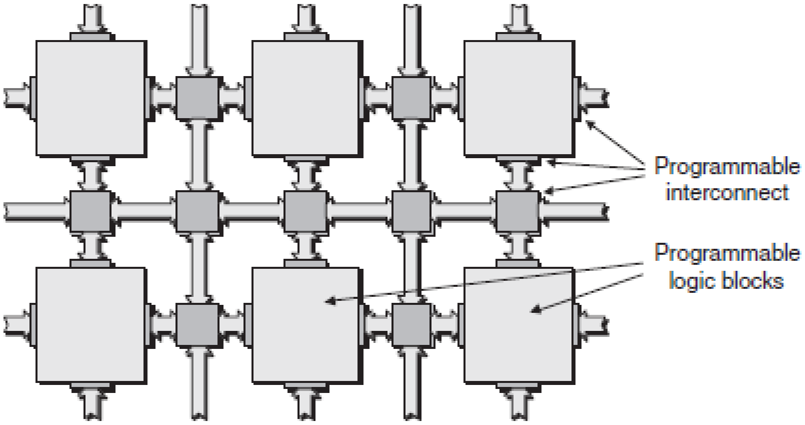


Figure 1.8: Top-down view of simple FPGA architecture[1, Pg 43]

LAB(Logic Array Block) is the Altera name for a programmable logic block containing a number of logic elements(LEs), where LE is the core building in a modern FPGA Architecture from Altera. Among other things, LE comprises of a 4-input Lookup table(LUT), a multiplexer and a register[1, Pg 503].

FPGAs are quite flexible, can contain millions of Logic gates and implementing design changes in them is much easier as compared to other logic devices. This results in widespread use of FPGAs in the market. Among the various areas of their use, Digital Signal Processing(DSP) is the area where they are in a huge demand. In this project, FPGAs will provide the DSP support.

1.3.3 Hardware Description Language

It is a fact that capturing a large design at the gate level of abstraction is prone to error and a time-consuming process. Thus, some EDA vendors started to develop design tools and flows using the Hardware Description Languages(HDLs) [1, p. 153].

Different levels of abstraction like Behavioral, Functional, Structural are followed by the HDLs for data flow. To make these HDLs easily accessible by the customers, an industry-standard HDL was needed.

To fulfill this requirement, Verilog HDL was introduced in the mid-1980s. Verilog was reasonably strong at the structural level of abstraction, very strong at functional level and it supported some behavioral constructs. One very good feature of this language was that Verilog programming level interface allowed the external software programs to pass data into an application and access data from that application[1, p. 164].

In 1980, U.S. Department of Defense(DoD) launched the very high speed integrated circuit(VHSIC) program with purpose of advancing in digital IC technology. In 1985, the first official release of VHSIC HDL(VHDL) occured[1, p. 167]. VHDL is very strong at functional and behavioral levels of abstraction and it supports some system level design constructs. In order to encourage acceptance by the industry , the developers subsequently donated all rights to the VHDL language definition to the IEEE in 1986.

VHDL was used in this project for providing the digital logic.

1.4 Challenge and Motivation

From a long time, Echo sounders have been extensively used in Hydro-acoustics research applications like seabed mapping, fish stock measurements, water quality monitoring, etc. In case of fish detection, Echo sounders can easily see targets and measure its size and strength. But a major challenge involved here is the recognition of species. It is difficult for echo sounders to get information about the same sized fishes of different species.

With the help of multi-frequency technologies, targets can up to some extent be classified into three categories:- plankton, fishes with and without swim bladders, but it is almost impossible to see the differences between species within these classes.

A solution to this problem could be the implementation of echo signature method where the shape of the returned echoes is studied. Echo's shape is very helpful to get the specific details of the target species. The echo sounders developed here at the Physics Institute used a band pass filter in the Analog front end receiver. The purpose of using filters was to suppress noise and therefore, improve Signal to Noise Ratio(SNR). But these filters also contributed in smoothing the echo shape. Thus hindering any further study of the echoes which were containing information about species recognition.

So, there is a need to develop a new Echo sounder which can give this information. For this purpose, the design of front end receiver should be reconsidered.

The receiver will be developed for wide band applications and can be used for different frequencies rather than a particular frequency. It will maintain the SNR and also provide the necessary gain for different sized targets at different depths. This dynamic gain can be achieved with the help of Logarithmic compressors or by using multiple amplifiers one after another. Voltage Amplifier chain will be the preferred method for this purpose because of huge literature resources and technical solutions available for its application. Moreover, dynamic span can be changed easily and set according to echo sounder's operation.

To build and test receiver's operation, it is desired to have both the transmitter and receiver on the same board. Earlier echo sounders used BJTs in the transmitter electronics which were connected to the transformer. This project will also look into other methods of pulse generation and amplification than the already developed solutions.

Chapter 2

Material and Methods

2.1 Softwares used in Project

2.1.1 OrCAD

OrCAD is a software tool suite for Electronic Design Automation owned by Cadence Design Systems. It is used for creating Electronic schematics and prints for generating PCBs.

2.1.1.1 OrCAD Capture

OrCAD Capture is a Schematic tool and has an in-built library for discrete components. New components can be added when necessary in the form of .olb files. The required library files are obtained from the vendor's website for the desired component and is added into OrCAD Capture with the help of the tool PSpice Model Editor.

2.1.1.2 OrCAD PSpice

After finishing with the Schematics part, OrCAD Capture exports the net-list to an external simulator known as OrCAD EE PSpice. OrCAD PSpice is a simulation tool for checking the analog verification of the circuit. The results of the simulations can be graphically displayed in a separate window.
This tool is very useful to get a knowledge of working of the circuit and is also used in the course, FYS3220.

A licensed version of OrCAD 16.6 available at E-Lab was used in this project.

2.1.2 CADSTAR

For developing the hardware part, a development tool from Zuken called CADSTAR is used. This is a licensed tool and was chosen because it was already available at E-Lab with a very large library of components with symbols and footprints. The hardware part starts from designing of the basic schematics design and then transferring the schematics(.scm) to the actual .pcb format. In .pcb file, components are placed by following some specific rules of production. Thereafter, routing between the various components is done in PR Editor XR. Tracks with different widths are used for routing depending upon the type of signal.

CADSTAR Version 15.0 was used for the hardware development.

2.1.3 QUARTUS II

Quartus II is a programming software tool by Altera. It is a free software for implementing the VHDL and Verilog languages for describing the hardware. It can be used to program FPGAs and CPLDs. Altera Board used in this project is DE1-SoC and the language used for writing the code is VHDL. The code is first compiled and then, implemented in the hardware part using the JTAG mode.

ModelSim is also a tool form Altera used for simulation and debugging of the digital logic. It can be used to check the timing requirements.

Quartus II Web Edition 14.0 was downloaded from Altera's website to be used in this project.

2.1.4 Ultra Librarian

Ultra Librarian is a free tool from Accelerated Systems used for importing and building libraries for CAD tools. It is useful for creating symbols and footprints of components and can support multiple CAD tools. These symbols and footprints can then be added in the libraries for PCB designing purposes.

2.2 System Description

The whole Echo sounder system is developed by combining the various units in subsystems like Transmitter, Receiver and Transducer.

This section will give a brief description about the various processes taking place in the Echo sounder system for solving the problems as discussed in Section 1.4. The motive is to develop a Hardware system containing all the necessary units on a single PCB board and then checking its operation.

The various tasks carried out by the system are described as follows:

- Generate pulses of desired frequency and duration and send them to transducer for conversion
- Generate High Voltages from an external board and connect it to the Echo sounder system for raising the amplitude of pulses
- Implement a protection circuit between the Transmitter and Receiver for wide band application
- Receive the low amplitude echo signals from the transducer and compensate for the Transmission Loss with the help of TVG
- Use digital method for Envelope Detection of the received pulses

The sketch in Fig. 2.1 shows how the various components are connected.

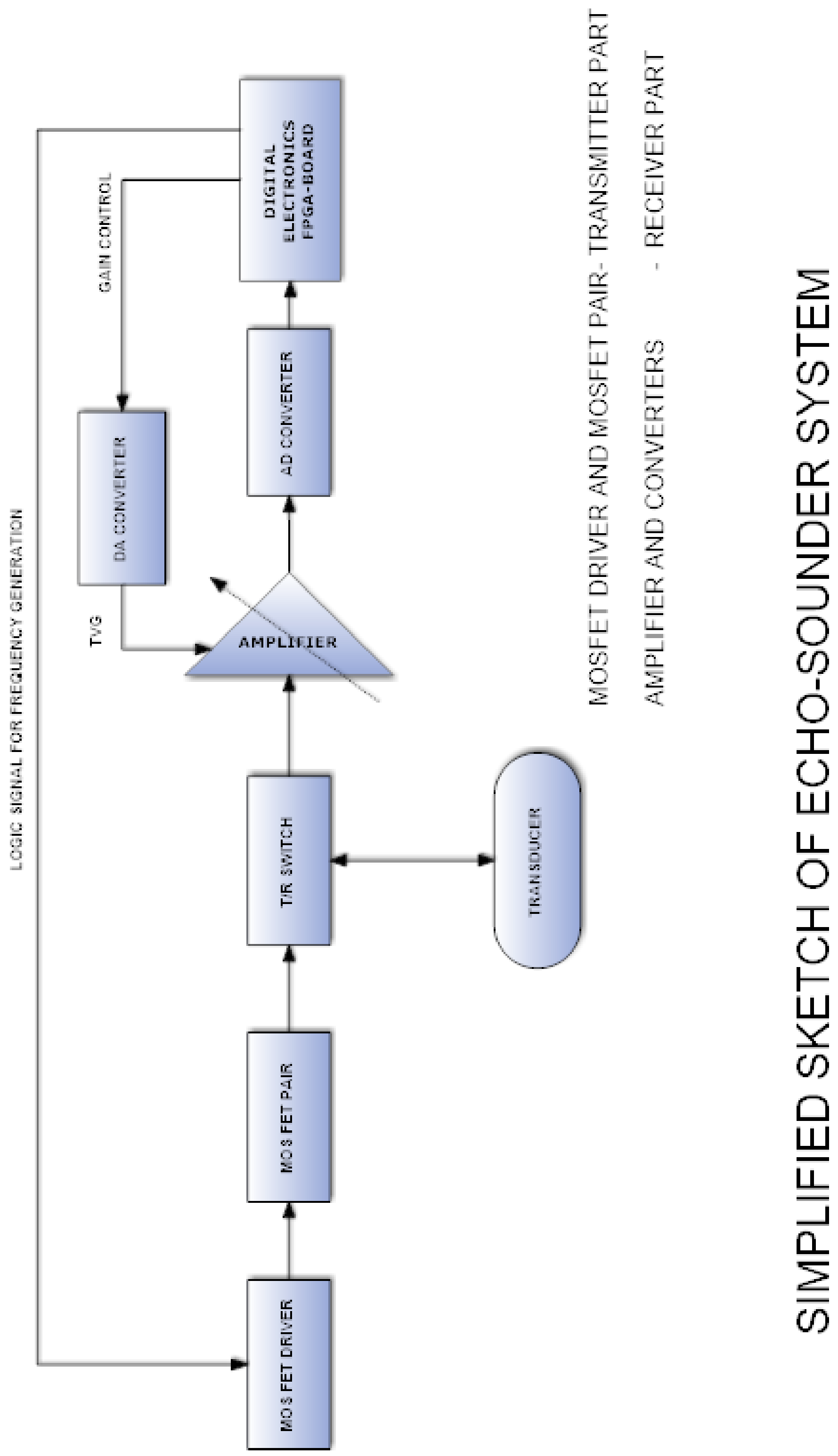

LOGIC SIGNAL FOR FREQUENCY GENERATION
GAIN CONTROL
DIGITAL ELECTRONICS FPGA-BOARD
DA CONVERTER
AD CONVERTER
TVG
AMPLIFIER
T/R SWITCH
TRANSDUCER
MOS FET PAIR
MOS FET DRIVER
MOSFET DRIVER AND MOSFET PAIR- TRANSMITTER PART
AMPLIFIER AND CONVERTERS - RECEIVER PART
SIMPLIFIED SKETCH OF ECHO-SOUNDER SYSTEM

Figure 2.1: Overview of complete Echo Sounder System

2.2.1 Pulse Generation

Electrical pulses are generated using digital logic before converting them into acoustic pulses with the help of a transducer.

For generating the electrical pulses, there should be a trigger pulse present which decides the rate at which pulses will be fired from the echo sounder. It is also called Pulse Repetition Frequency(PRF) and is important because it must always occur at a precisely defined interval of time. PRF depends on the depth of water to be surveyed i.e. time interval long enough between pulses so that all the echoes resulting from first transmission have returned, before the next transmission. Otherwise, it will affect the echo sounder's proper operation.

The time interval between the successive bursts is decided by the sound's velocity in water, c and the total distance traveled by the pulse i.e. with increasing distance, the time between bursts will increase.

The expression is given by:

$$Time = 2 * depth/c \qquad [36, Pg. 24] \tag{2.1}$$

Let depth = 75m
Therefore, Total distance traveled = 150m

So,

$$Time = 150/1500 = 0.1sec \tag{2.2}$$

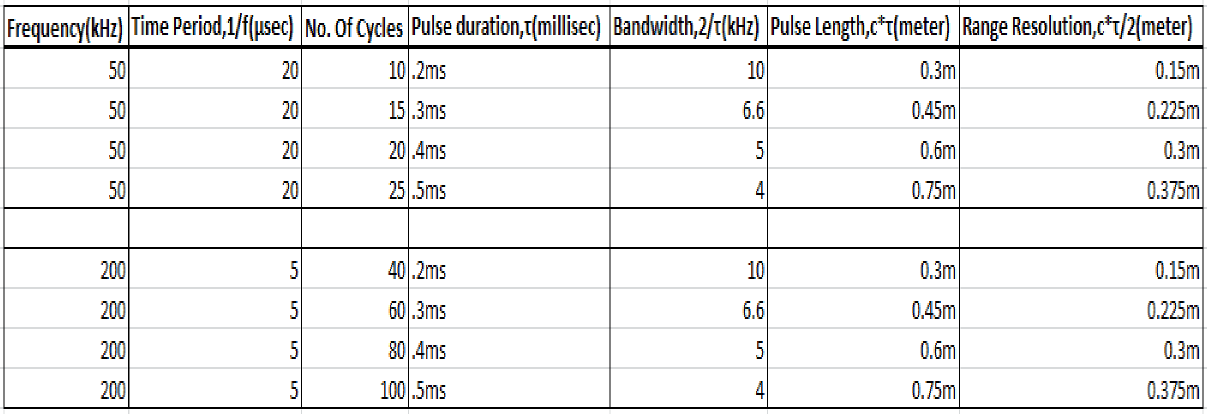
It means that time interval between successive bursts should be around 100ms.

$$PRF = 1/Time\_Interval \tag{2.3}$$

Therefore, Maximum PRF should be 10Hz or 10 bursts per second.

The trigger starts the process of generation of pulses of desired frequency and duration. The pulses are generated through Pulse Width Modulation(PWM) technique. These Square wave pulses have a 50% duty cycle i.e. each pulse, for half of the time it is in positive cycle and in the next half it is in the negative cycle. These electrical pulses usually have a defined frequency like 50, 70 or 200kHz and their duration is decided by the number of cycles. This pulse duration is called tau(τ) and is usually in the range .1-.5ms for better resolution. So, for a 50kHz pulse the number can be between 5-25 cycles.

When this τ is multiplied by $\frac{c}{2}$, range resolution is obtained. Range resolution is important to determine the minimum distance between targets i.e. between two fishes, so that their reflected echoes do not overlap each other and can be prevented from being shown as a single signal at the receiver end. Therefore, with smaller τ, better resolution can be achieved.



Frequency(kHz)	Time Period,1/f(µsec)	No. Of Cycles	Pulse duration,τ(millisec)	Bandwidth,2/τ(kHz)	Pulse Length,c*τ(meter)	Range Resolution,c*τ/2(meter)
50	20	10	.2ms	10	0.3m	0.15m
50	20	15	.3ms	6.6	0.45m	0.225m
50	20	20	.4ms	5	0.6m	0.3m
50	20	25	.5ms	4	0.75m	0.375m
200	5	40	.2ms	10	0.3m	0.15m
200	5	60	.3ms	6.6	0.45m	0.225m
200	5	80	.4ms	5	0.6m	0.3m
200	5	100	.5ms	4	0.75m	0.375m

Figure 2.2: Table showing relation between pulse duration, cycles and bandwidth

As seen from the above table, the smaller is the pulse duration, the larger will be the bandwidth. This means that such a transducer should be chosen that allows these small duration pulses to pass through it without affecting them. But the noise entering the receiver system puts a limit on transducer's bandwidth.

After generating pulses of desired time duration, it is then necessary to amplify them up to a certain level so that the output power is raised to around 50-100 watts. This output power varies with the transducer resistance. Conversion of this electrical energy into sound energy is also dependent on transducer's efficiency. As no transducer is 100% efficient, so there is a definite power loss while conversion. Impedance matching between the transmitter and transducer should also be kept in mind for effective energy conversion.

In this project, MOSFETs are used instead of BJTs in the Transmitter Electronics. Also, the transmitter is connected directly to the transducer.

2.2.2 Alternate option for voltage amplification

Using a transformer in the transmitter part is the most common approach used in the echo sounder designing because of simplicity in design but they have certain losses like core loss, copper loss, etc. associated with them. Copper losses change with load and for wide band systems, where the transducer offers different loads at different frequencies, efficiency of transformer is affected.

The transmitter circuit which is in use today can generate a sound pulse with power in the range 40-60 Watt when it is driven at its full strength[15, Pg. 16]. Therefore, if it is necessary to increase the transmitted power, say up to 100 W, this solution must be replaced.

An alternative could be to generate high voltages using Switched Mode Power Supply(SMPS) method. Here, a boost converter (step-up converter) technique is used to raise the voltages from low to high. The operation is based on the principle that the voltages which are obtained, are from the rapidly collapsing magnetic field in an inductor. This is done a number of times in a second and the obtained output pulses are then smoothed. The advantage of this design is that it is small, cheap and very efficient. Therefore, this method has been used in this project. The basic configuration of boost converter is shown here.

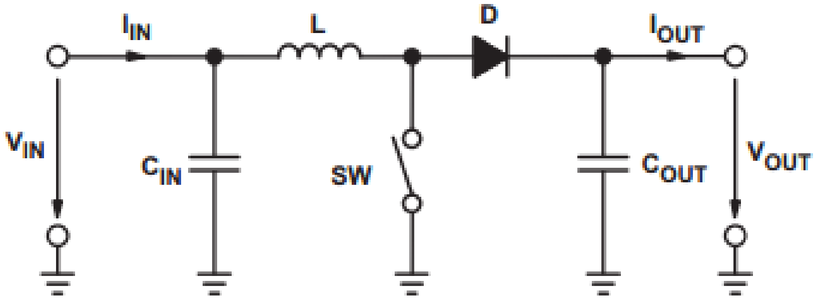


Figure 2.3: Basic schematic of Boost Converter[50, Pg. 1]

An IC is used in combination with the switch to complete the circuit. The working of boost converter is based on the switch's Open and Closed state. In the Closed state, Inductor stores energy by generating a magnetic field and there is an increase in Inductor Current and in the Open state, the magnetic field created will be destroyed to maintain the current flow towards the load. The capacitor at output together with the inductor forms a filter which filters the train of pulses to produce a DC output voltage[51].The output voltages can be adjusted by using a potentiometer.

Here, a N-channel MOSFET will act as a Switch. The board will get its input power from an external 12V supply.

These output voltages are then given to the transistors on the Echo sounder board to amplify the incoming pulses. These transistors should have very fast switching speeds and capable of handling high voltages.

2.2.3 Transducer

2.2.3.1 Choice of Transducer

The transducers convert one form of energy into another and the transducers used in hydro acoustics convert voltage(electrical) signals into sound signals and vice-versa. These also acts as filters at the resonant frequencies while receiving the signal and thus remove the surplus noise and increase the signal to noise ratio. The impedance of the transducer elements decrease at the resonant frequency.

The piezoelectric transducer element used in this project is FURUNO 520-5PSD which has two resonance frequencies centered around 50 and 200 kHz. This transducer acts as both transmitter and receiver of signals. This transducer can be considered as the first band pass filter in the receiver electronics. The various characteristics of this transducer are:-

- Maximum power: 600 W

- Beam angles : 46/10 degrees

- Plastic Thru-Hull

- 8-meter Cable with 10-pin connector

Every transducer has a certain bandwidth. For this transducer, it is around 5kHz[28, Pg 71].

2.2.3.2 Measurement of impedance of transducer

The impedance vs frequency curve of a transducer can give a lot of information about its various characteristics such as, it can provide information for impedance matching of transmitting and receiving systems with transducer. It is used for computation of transducer efficiency and driving current or voltage from transmit responses. And this response can be used to find the values of the discrete components of the equivalent electrical model. Therefore, there is a need to measure transducer's complex impedance.

The transducer's electrical impedance affects the transducer's noise performance,driving response,bandwidth and sensitivity. In the frequency function, the impedance tends to change values rapidly, in particular where resonances are involved. This transducer will show an impedance drop at its two resonant frequencies. At resonance, the impedance is purely resistive and the inductive and capacitive impedances cancel each other as voltage and current come in phase. This impedance is small and therefore, the current increases. To determine the response, an AC voltage source is added to the circuit. Typically, the circuit used to measure impedance is basically a constant current circuit[39], i.e. here the current does not depend on the impedance measured value but only on the external resistance R.

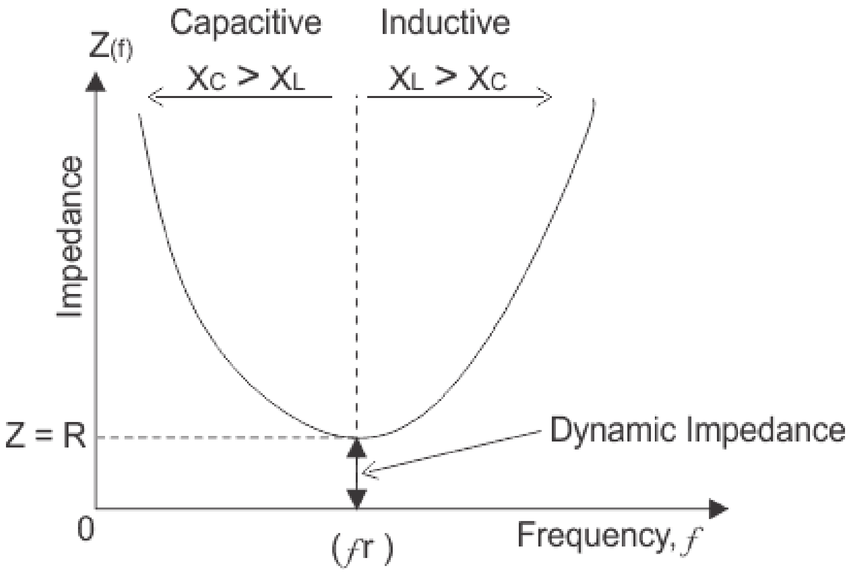


Figure 2.4: Impedance vs Frequency response[26]

In this setup, HP 33120A function generator is used to provide an AC sweep voltage of 5V between two particular frequencies to the transducer through a 1K Ohm resistor(R). The response is observed with the help of Tektronix DPO 2004B oscilloscope which is connected to the resistor(R) and the transducer. At resonance frequencies of 50 and 200kHz, the impedance of transducer changes drastically.

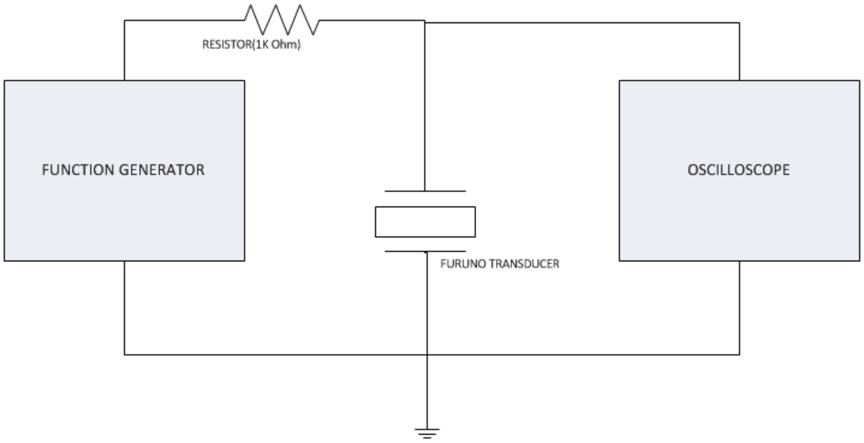


Figure 2.5: setup

The resistor(R) used here together with the transducer acts as a voltage divider circuit. A sinusoidal voltage sweep between 10kHz and 250kHz was given as the input. The provided setup is used to calculate the resistances at the resonant frequencies 50 and 200 kHz.

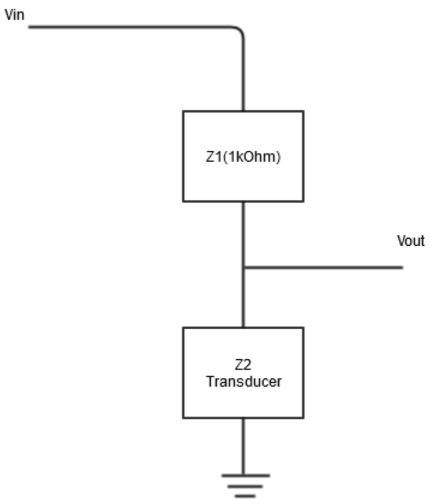


Figure 2.6: Voltage Divider Circuit

The impedance calculated at the resonant frequency of 50 kHz is given by:-

$$V_{out} = \frac{Z_2}{Z_1 + Z_2} * V_{in} \qquad [23, Pg. 187] \tag{2.4}$$

Here, V_{in} = 5V, Z_1 = 1K Ohm and V_{out} is calculated to be 0.8 V at 50 kHz.

Using the equation 2.4, Z_2 is found to be 190.48 Ohm. This is the impedance(or resistance) of the transducer at 50kHz.

Similarly, for 200 kHz V_{out} is found out to be 4.1 V and the impedance is calculated to be 4494.51 Ohm.

2.2.4 Receiver Protection Circuit

A protection circuit needs to be used to protect the receiver electronics against the high voltages coming from the transmitter.

- Filtering
- Using Switch as a receiver protector

2.2.4.1 Filtering

The signal received by the transducer will not comprise of a pure echo signal but it will be surrounded by ambient noise from sea and the Electronics equipment will also contribute to this noise. The noise can contain frequency components which can be both higher and lower than the pulse's center frequency. Therefore, it becomes necessary to filter out the unwanted noise from the received echo.

Earlier developed receiver front ends had a filtering circuit for filtering out the noise like the one developed by Randgaard used Op-amps for making Bandpass filters with center frequencies of 10 and 70kHz[3, Pg. 16]. Hellum also developed a passive filter which comprised of a LC circuit[4, Pg. 9]. These filters help in filtering out the desired frequency signal from the unwanted frequency signals and at the same time provide a certain amplification to the received signal, i.e. improve SNR depending upon the configuration of the circuit. The Inductor used in this LC circuit was specially constructed at E-Lab and built for operation at 50kHz only. This puts a limitation on this filter to be used as a front end for other frequencies.

These circuits are applicable only in narrow band receiver circuits. The other disadvantage associated with these filters is that the system gets locked for a particular frequency like 50 or 70kHz. So, for wide band receiver development these circuits are not optimal. The echo sounder system should be developed in a way that it can operate with different frequencies in the area 50-200kHz without changing anything in the hardware circuit. This way, the center frequency can be changed easily in the digital part when desired.

Option considered

As far as filtering of the received echo is concerned, the transducer itself assists in filtering and helps in removing the unwanted signals in advance.

I decided not to use any filter before the amplification stage for making the system flexible for operation at various frequencies. The other major advantage is that by doing so the echo signal will now remain unaffected and useful information about species recognition can be obtained from the later envelope detection part.

Hellum's front end, apart from filtering also helped in protecting the receiver from the high voltages. So, there is a need to isolate the receiver from the transmitter during transmission by inserting a protection circuit between the two. Among the available solutions, a choice was made to use a simple circuit which does not require any external control signals for its operation.

2.2.4.2 Using Switch as a receiver protector

A switch was introduced between the transmitter and receiver. The switch considered, should be capable of protecting the receiver side and remains normally closed i.e. in receive mode because the pulses are sent at a very low frequency of around 10Hz. It means that the total time the Switch has to remain open during a time interval of 1sec will be less than 10ms. This should allow the low amplitude signals from the transducer pass the switch unattenuated. But as soon as the voltage rises and it passes a certain level of amplitude, the switch protects the receiver by going in Open/Disconnected state. Thus, providing isolation between transmitter(T_x) and receiver(R_x) Electronics.

Moreover, it is preferred that it should be a user friendly component having small size and very less number of pins such that it can be easily tested in the lab before it is used in the design.

2.2.5 Compensation for Transmission Loss and Amplification

The acoustic pulse while traveling through the medium experiences Transmission Loss(TL). This loss is mainly due to spreading and attenuation. The spreading loss is a geometrical effect representing the regular weakening of a sound signal as it spreads outward from the source and this loss increases with the range[6, Pg. 100].This affects the signal's strength and as a result, the echo which is received has a very low amplitude. Therefore, this loss needs to be compensated.

2.2.5.1 Understanding Source Level, Transmission Loss and Echo Level

There is a relation between the projector's Source Level(SL) and the input power and is given by:-

$$SL = 171.5 + 10Log(P_e) + 10Log\eta + DI_T \qquad [6, Pg\ 75] \tag{2.5}$$

; where P_e is Electric Power,
η is efficiency of transducer and assumed to be 50%

For FURUNO 520-5psd Transducer, the Directivity Index(DI) is 13dB[28, Pg 88]. Using 100W as transmitting power, we have:-

$$SL = 171.5 + 10Log(100) - 3.01 + 13 \tag{2.6}$$

It implies that:

$$SL = 171.5 + 20 - 3.01 + 13 = 201.4dBre1uPa \tag{2.7}$$

The transmission loss responsible for the weakening of the pulse strength is given by the equation :-

$$TL = 20Log(R) + aR \qquad [6, Pg\ 111] \tag{2.8}$$

, where α is the absorption coefficient and 20Log(R) is the spreading loss.

The signal looses its strength both ways, while going towards target and coming back from it, so losses are considered equally in both directions. Value of α is dependent on various factors such as viscosity, frequency, ionic relaxation of salts like Magnesium Sulphate. Therefore, the losses caused by α are dependent on the medium and cannot be decreased.

For long distance(R = 100m), Transmission Loss is calculated out to be :-

$$TL = 20Log(100) = 40dB \tag{2.9}$$

Therefore, TL > 40 dB because losses caused by α also need to be included. As α depends on frequency, it will have different values at different frequencies, E.g. α for 50 and 200kHz is .009dB/m and .05dB/m respectively.

The next step is to find the Target Strength(TS) from a small fish. Assuming that the length of the fish is 13cm, then TS of this fish for two different frequencies, 50kHz and 200kHz can be found out by equation:-

$$TS = 19.1\log_{10}(L) - 0.9\log_{10}(f) - 62.0 \qquad [6, Pg. 316] \tag{2.10}$$

For 50kHz :-

$$TS = 21.2 - 4.22 - 62.0 = -45.02dB \tag{2.11}$$

Similarly for 200kHz :-

$$TS = 21.2 - 4.77 - 62.0 = -45.57dB \tag{2.12}$$

As seen TS mainly depends on the fish size. So, bigger size fishes will produce greater TS.

Now to find the EL, there is a need to use the Sonar equation Eq.1.1:-

$$EL = SL - 2(20logR + aR) + TS \tag{2.13}$$

So for 50kHz frequency,

$$EL = 204.5 - 2(40 + .9) - 45.02 \tag{2.14}$$

Therefore, EL = 77.68dBre1μPa

And finally it is desired to find the voltage produced across the transducer terminals, which is given by:-

$$V_{Trans}(dB) = EL + SRT \qquad [4, Pg. 25] \tag{2.15}$$

;where SRT = Receiving Response of transducer and is equal to -184dBre1V/μPa[28, Pg 94]

It implies that:-

$$V_{Trans} = 77.68 - 184 = -106.32dBV \tag{2.16}$$

which is equivalent to 4.8μV.

So, a small fish produces this much voltage across transducer terminals and receiver must be efficient enough to process echo with this level of amplitude.

For compensating this loss of amplitude due to these various phenomenons like transmission, scattering, absorption, etc. an algorithm needs to be followed which can raise this received signal's amplitude to a level which can be easily processed by the following receiver Electronics.

2.2.5.2 Gain Amplification

To provide the necessary gain to the echo signal, an amplifier is needed. This can be achieved by providing fixed amplification as implemented by Dr. Balk[36, Pg. 69] using Op-amps and discrete components in its various channels. But it is desired to have a digital control over the gain range. Therefore, it is advantageous if an amplifier is used whose gain can be changed digitally.

A single amplifier can provide the necessary gain but the Bandwidth will decrease accordingly with gain due to the Gain-Bandwidth(GBW) product. So, it is practical to use more than one amplifier to raise the low amplitude echoes. Connecting amplifiers in series, where each amplifier sends its output to the input of the next amplifier is the technique used for amplification. This also improves the attenuation of out-of-band interferes, since cascaded stages have a sharper high-frequency roll-off for the same effective gain-bandwidth product, compared to a single-stage amplifier[16, Pg. 16].

As stated above, the gain required should be large enough so that output voltages are in the operating range of ADCs. To achieve this goal, the number of cascaded amplifiers needs to be chosen according to the gain required. A large number of amplifiers are available from different vendors. But only the amplifiers with the variable gain characteristic should be used for this purpose. This is because echoes from targets i.e. fishes at different depths will require different gain due to different ranges covered by the signal. As the intensity of the signal falls directly with the distance traveled and thus, with time. Therefore, Variable Gain should be dependent upon time i.e. the purpose of applying Time Variable Gain(TVG) here is to reduce the dynamic range of echo signal and adapt it to the dynamic range of the ADC[38].

TVG is accomplished by generating a voltage waveform of a defined wave shape and applying it to the receiver amplifier. The TVG can be controlled either digitally or by analog voltage. For the simplification of instrumentation and for ease of adjustment, TVG action can be obtained by charging a capacitor with a negative voltage during the transmission cycle, and after this cycle, discharging the capacitor through a shunt resistor. The over-bias voltage then decays exponentially to the normal bias level. Thus, the amplifier is designed to have a gain in dB that is approximately linear with the bias voltage[40, Pg. 314]. But it is more accurate to provide the TVG digitally with the help of a Digital to Analog Converter(DAC) or Digital Potentiometer. A precise TVG can then be achieved and changed accordingly with range and incoming signals. The output analog voltage from DAC can then be given to the variable amplifier as gain control voltage.
This TVG should compensate for the following Transmission Loss(TL) vs. Range curve.

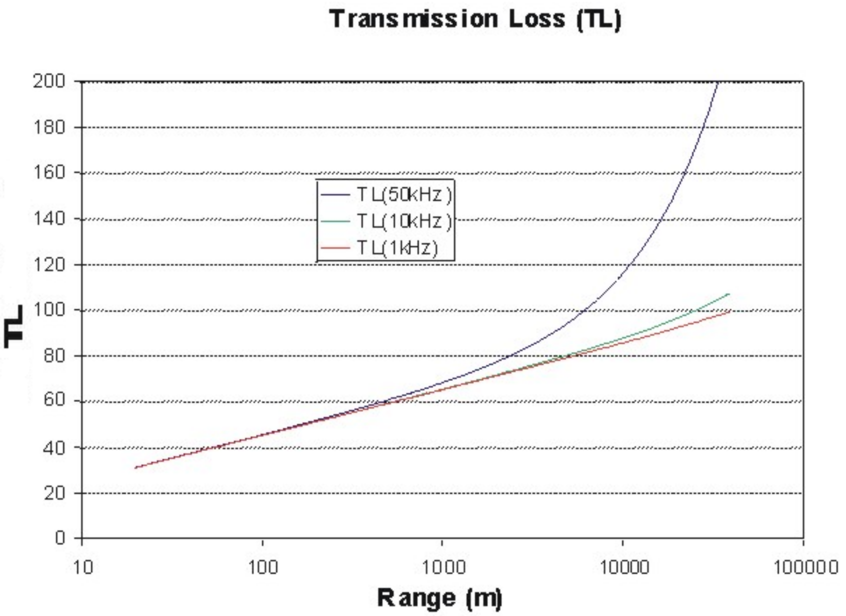


Figure 2.7: Transmission Loss vs. Range[41]

For the ultrasound systems, all the amplifiers require a common analog voltage for gain control and it becomes easier to achieve gain with a single control voltage routed to each VGA.

2.2.6 Digitalization of received echoes

After amplification, it is required to convert the analog voltage into digital values. An Analog to Digital Converter(ADC) is required for this conversion. These digital values are then sent for further signal processing. These values could be provided serially or in parallel form depending upon the choice of the ADC. Generally, serial operation requires more clock cycles than the parallel operation to send a digital value.

The ADC is the important component of this design and therefore, it needs to be protected against the time varying input signals. These signals may sometimes exceed the input range that is specified in the manual of that ADC. So, a protection circuit must be employed at the inputs of the ADC so as to avoid damage. There are ADCs available in the market which can operate at higher input voltage than their supply voltage. Most of the ADCs today have internal protection diodes,which conduct when the input voltage goes beyond the supply voltage. But these diodes are not efficient for large currents for a longer amount of time.

ADC Input Protection

A good solution for protecting ADC is to use a pair of Schottky diodes at its inputs. This method is feasible if the input signal is directly connected to the ADC input or the signal conditioning amplifiers are operating at voltages greater than the ADC analog supply voltage. In this design, this method is employed to protect the ADC for the later case when the incoming amplified signal is very large due to reflections from targets in water which are very close to the transducer.

For protection purposes, the diodes are placed at the input after anti-aliasing RC filter. It is little bit advantageous to place them after the filter because resistance acts as a current limiting device if the input goes beyond the input range of ADC. Also, if the input current goes too high, the resistance will itself get damaged but will protect the clamping diodes and the ADC[10, Page 2]. The anode of upper diode and cathode of the other are connected to the input signal and the other end of upper and lower diode are connected to positive and negative supplies respectively.

This method is applicable for single supply ADCs also. If the ADC is working only on positive supply say 3.3V, then the anode of the lower diode is connected to the ground.

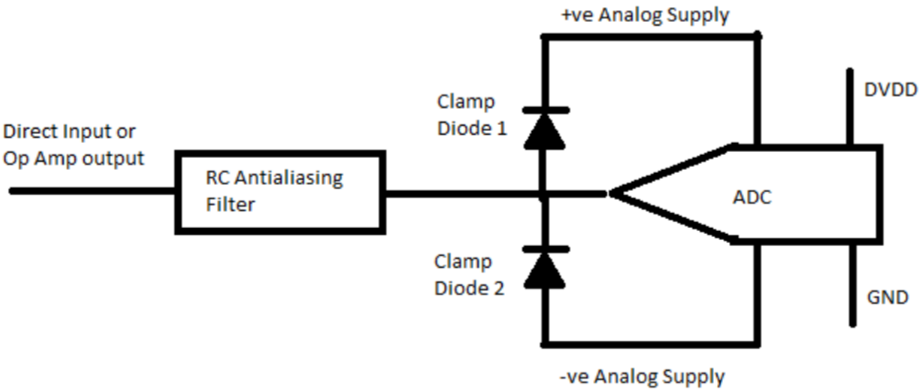


Figure 2.8: ADC Input protection[10, page 2]

Simulation and testing

The ADC chosen operates only on single positive supply. So, only cathode of upper diode needs to be connected to the voltage supply. It will be advantageous if the response of this protection circuit is checked beforehand. Therefore, the circuit in Fig.2.8 is drawn using OrCAD to perform simulation.

Schottky diodes 10BQ015 were used in the design but for performing simulations, MBRS120T3 diodes were used and added in PSpice library. This was done because their values resembled with the actually used Diodes 10BQ015 and their library(.olb) files required for simulation were easily available.

The setup is shown here showing both anti-aliasing filter and ADC input protection circuit. The response is checked by giving high and low input voltages from an AC source. The Schottky diodes clamp the positive cycle of input voltage if it goes above the supply voltage, V_{DD}= 3.3V, whereas no change takes place in the positive cycle if amplitude is less than V_{DD}.

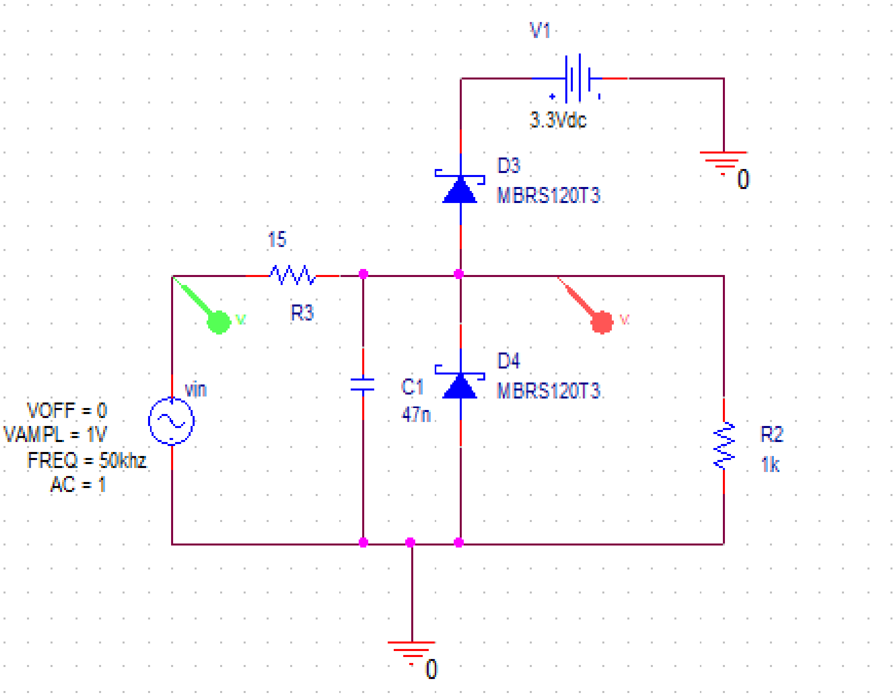


Figure 2.9: Simulation setup

Simulation is performed and it can be seen that the diodes completely remove the negative half of input as the other diode is connected to ground and only the positive half of the sinusoidal signal is passed. The output of this circuit is fed as input to the ADC. Generally, some ADCs have Common Mode voltage which helps to raise the input to a specific DC level.

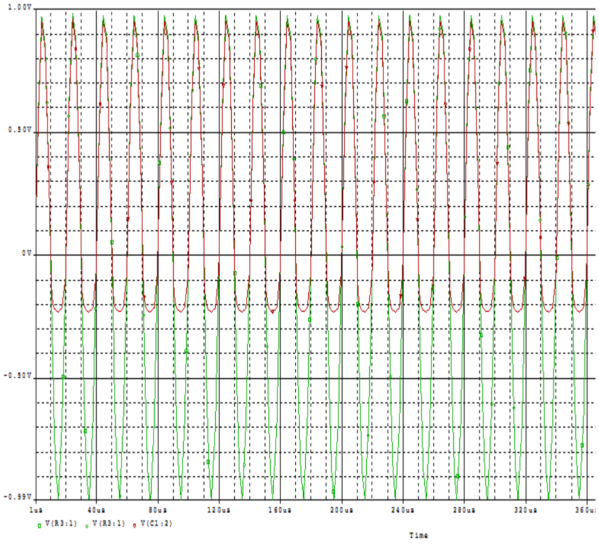


Figure 2.10: Input 1V

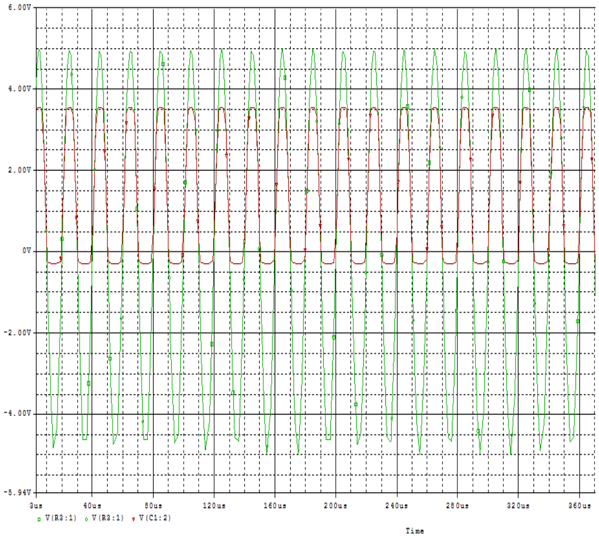


Figure 2.11: Input 5V

2.2.7 Envelope Detection

The received echo will be a pulse of Amplitude Modulated signals i.e. every cycle can have different amplitude. It is the maximum amplitude of each cycle of echo that contains information about the target and we are interested in the envelope which covers all these maximum amplitude points. The earlier methods used a diode and a capacitor[21, Pg. 29-32] in the analog electronics, where the principal was to charge the capacitor through the diode and then this voltage stored at the capacitor was used for converting into digital values, but there was certain amount of inaccuracy and distortion involved. Moreover, these methods produced bad results at higher frequencies like 200kHz. So, now the trend is to do it digitally.

This can be achieved by first passing the analog signal through ADC and then using a Micro-controller or FPGA to extract the peak values of each pulse cycle. This method is more efficient as micro-controllers and FPGAs are more accurate and much faster.

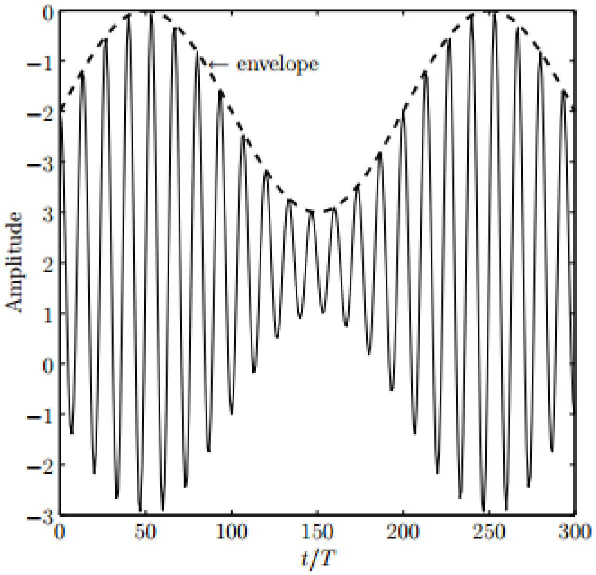


Figure 2.12: Envelope Detection of a pulse[37, Pg. 6]

FPGAs are chosen to provide signal processing support in this project. This method has been earlier demonstrated for Xilinx[3] and Altera[15] FPGAs. So, an Altera FPGA available at the Institute can be used for the purpose of envelope detection. The echo-sounder frequencies will be in the range of 50-200kHz. So, a sampling frequency of 10 times the maximum frequency will be sufficient for obtaining the input signal digitally. This sampling frequency is provided to the ADC for conversion. ADC with 14-bits resolution or more is desired to get the top values of the every pulse cycle.

Chapter 3

Design and Development

- **3.3 Software Development**

3.3.1 Simulation
3.3.1.1 Simulation of AD605
3.3.1.2 Simulation using ModelSim

3.3.2 VHDL Code Development
3.3.2.1 Clock Divider
3.3.2.1 MD1213
3.3.2.3 DAC AD5641
3.3.2.4 ADC LTC2245

3.1 Choice of components

3.1.1 Transmitter

Pulses can be transmitted using Bipolar Junction Transistors(BJTs) and Metal Oxide Semiconductor Field-Effect Transistors(MOSFETs). Both are very reliable in amplification and switching applications.

MOSFETs are voltage controlled devices and are more efficient in handling power and suitable for high-power applications. MOSFETs when properly biased have a very low drain-source ON resistance $R_{DS}(on)$, which means that they behave quite much like an actual switch when turned on. Also because of the advancement in technology, MOSFETs are the most common transistors used in the design circuitry.

3.1.1.1 MOSFET Driver

For transmission of pulses, a push-pull MOSFET pair is required and to drive these transistor pairs, a dual MOSFET driver is used. MD1213 solves this purpose[11]. This MOSFET driver operates at supply voltages between 4.5 and 13V and therefore, compatible with the FPGA board used in this project. The input stage can operate from 1.8 to 5.0V logic interface and the voltage level provided with the logic signals from the FPGA board is 3.3V. The output stage of the MD1213 has separate power connections enabling the output signal L and H levels to be chosen independently form the supply voltages. The 3 inputs OE, INA and INB will provide the Logic inputs. The outputs need to be connected to the external PMOS and NMOS Transistor pair through a series AC Coupling capacitors of 10nF each.

The OE pin serves a dual purpose. First, its logic H level is used to compute the threshold voltage level for the channel input level translators. Second, when OE is low, the outputs are disabled, with OUTA being high and OUTB low. This assists in properly pre-charging the AC coupling capacitors that will be used in series in the gate driver circuit of the N- and P-channel MOSFETs.

This MOSFET Driver is widely used in driving piezoelectric transducers both in hydro acoustics and in medical ultrasound systems. MD1213 comes in a small 12-pin QFN package[11, Page 7]. A couple of MD1213 ICs were purchased from the available vendors. Also, the symbols and footprints of MD1213 were designed and added into CADSTAR Library(and in DVD also) as they were required for Schematics and PCB development.

The other similar options were MD1210[44] and MD1211[46]. But MD1213 was the preferred choice because Bipolar output voltage of $\pm 5V$ can also be achieved with this chip if desired.

3.1.1.2 MOSFET Pair

The outputs from the MD1213 MOSFET Driver need to be fed in to a transistor pair so as to produce High Voltage pulses for transmission. An essential feature of these transistors should be that they should have a low output impedance so that the signals can be successfully injected into the transducer circuits without adversely affecting them. There are various MOSFET arrays components available and can be chosen depending upon the various parameters like $R_{DS}(on)$, cost and the breakdown voltage depending upon the application. MOSFET array which fits the requirements was TC6320 and therefore, it was used.

TC6320 is a high voltage, low threshold N and P channel Enhancement-Mode MOSFET Pair. It can operate at high speeds and high voltages. TC6320 has integrated Gate to Source resistor and Gate to Source Zener Diode on both channels. During power up and power down conditions, it is possible for transient voltages greater that 20V to appear across the gate-to-source on the output transistors. These built-in 15-18V Zener diodes help to protect against such transient voltages. These diodes will not have any Zener current during normal operation and also serve as the gate DC voltage restoring functions[33, Pg. 2].

These ICs are best suitable for amplifying and switching applications such as in driving an echo sounder transducer. TC6320 is highly robust, efficient and free from thermal runaway unlike other MOS Structures.

TC6320 comes in 8-lead SOIC and DFN packages[12, Page 2].SOIC package was chosen for this project. The footprints and symbols were already available in CADSTAR Library and were edited to be used for this component.

The outputs of this MOSFET pair needs to be connected to two external BAS16 diodes on P and N channel and then single output is taken thereafter to drive the piezoelectric transducer, FURUNO 520-5PSD.

The alternate option available was TC2320[45] but TC6320 was chosen because of its low Drain-Source Resistance(R_{DS}).

Alternate option

The other option available was to have a circuit which could solely provide the above processes within a single chip. After searching, it was found out that a high voltage, high speed pulse generator called HV7361 can be used as the complete Transmitter Unit[14]. HV7361 consists of a controller logic interface circuit, level translators, AC coupled MOSFET gate drivers and P-channel and N-channel MOSFETs, all built-in.

In addition, a Transmit/Receive Protection Switch was also included in this chip. But this option was discarded because of its 22-Lead LFGA package and the combined costs of all three described ICs were less than half of this single HV7361 chip. Moreover, the advantage of using individual ICs is that there is always an option available to change the complimentary MOSFET pair with another one depending upon the amplitude of the pulse outputs desired.

However, symbol and footprint for HV7361 have been made and added in CADSTAR library and DVD.

3.1.1.3 High Voltage Inputs

The source voltage pins(S1 and S2) of TC6320 MOSFET pair need to be connected to high negative and positive voltages, like $\pm 50V$. Such high voltages need to be produced without adversely affecting the operation of the rest of the circuit as these may lead to noise and other effects. Therefore, it is desired to make such high voltages away from the echo sounder board. So, the pins 1 and 3 of TC6320 which represent Source terminals of N-channel and P-channel respectively can be used to connect to the high voltage supplies.

To fulfill this purpose, a separate PCB board was designed which can give the necessary voltages to the echo sounder board.

3.1.1.4 Step -Up DC-DC Controller

For boosting the voltage from 12V to a higher level(say,100V), a step-up converter is required.

MAX1771 is a step-up switching controller and can be used in this project for stepping-up DC-DC Voltage from flexible input range between 2V and 16.5V to the desired value. A unique current-limited pulse-frequency-modulation(PFM) control scheme gives this device the benefits of pulse-width-modulation(PWM) converters. MAX1771 is highly efficient IC which can provide load current over the range 30mA to 2A and uses miniature external components. This IC drives an external N-channel MOSFET Switch allowing it to power loads up to 24W. [20, Page1].

MAX1771 comes in 8-pin DIP and SO packages. As this IC was exclusively used in this project and not available in E-Lab, it was purchased in SO packages and added in the library.

3.1.2 HV Protection Transmitter(T_x)/Receiver(R_x) Switch

The transmitter and receiver part are separated from each other during transmission using a switch.

The high voltage protection T/R Switch, MD0100 is a very good solution[9]. This switch is bidirectional and designed for its operation in ultrasonic applications. It can handle up to $\pm 100V$ input voltages. It is primarily designed to protect a low noise receiver circuit from high voltage transmit pulses. It has a very wide bandwidth of 100MHz. It is a normally closed(NC) Switch with a typical switching resistance of 15 Ω that allows small signals to pass through it. Once the voltage drop across the two terminals exceeds a nominal value of $\pm 2.0V$, the device will start to turn off. In the off state, the MD0100 can withstand up to $\pm 100V$ across its terminals while a small amount of current of 200μA is allowed to flow through.

It is almost a complete protection circuit which comes in user friendly SOT-89 package and needs only a pair of back-to-back diodes from the receiver side connected to switch and ground. The purpose of diodes is to allow a current path. Otherwise, if there are no diodes present, then there will be no current path and the voltage drop across terminal A and B will be less than ±2V and the switch will remain in the closed position. Also, the diodes help in clamping the voltage spikes to ±0.7V during transmit cycle[9].

Because this switch was not available in E-Lab, it was purchased and necessary symbol and footprints were added in Cadstar Library.
Switch MD0101[47] is also a good option as it has integrated clamp diodes and therefore no need of external diodes but it is useful when there are more number of channels in the system. Similarly, if voltages greater than $\pm 100V$ are used for transmission, then the switch MD0105[48] can be used. This switch can be used for protection against voltages up to $\pm 130V$.

3.1.3 Receiver

Signal Conditioning Components

Signal conditioning is a term which is frequently used in Electronics and refers to the manipulation of the analog signals such that they meet the requirements of the later stages, especially ADCs. In signal conditioning stage, processes such as amplification, filtering, range matching, isolation can be done depending upon the input signal and the requirements of the later processing part. The various components used for signal conditioning in this project are:-

3.1.3.1 Variable Gain Amplifier

Variable gain amplifiers are easily available but there are some requirements which help in filtering out the unnecessary amplifiers. It is desired that amplifiers can provide a gain range from 0 to around 90 decibels(dB). The bandwidth should also be large for different gains. Among available choices amplifier like AD8338[42] from Analog Devices was a good choice. It can provide gain up to 80dB with 18MHz Bandwidth. It was not used because it offered a slightly higher Input noise.

Amplifier AD605 is used in this project. AD605 is a low noise, differential-input, dual-channel, linear-in-dB Variable Gain Amplifier. It fulfills the desired characteristics of gain, noise, supply voltage etc. It comprises of two variable gain amplifiers which can be connected in series to get higher gain ranges, up to 96dB and it uses a common gain voltage(V_{GN}) for both the amplifiers to control the variable gain.

The gain scaling for this amplifier can be set between 20dB/V and 40dB/V with the help of VREF pin for better gain scaling. Providing VREF 2.5V will set the scale factor to be 20 dB/V while 1.25V will set it to 40 dB/V. Accordingly, the gain scales linearly in dB with control voltages(V_{GN}) of 0.1V to 2.9V for the 20 dB/V scale and 0.2V to1.2V for the 40dB/V scale. The amplifier can be used with differential/single-ended inputs and the output is single-ended. The negative inputs of both the channels are connected to the ground through 0.1μF ceramic capacitors if it is used as a single-ended amplifier.

AD605 comes in 16-lead PDIP and SOIC packages. SOIC package was used as rest of the components are also surface mounted(SM). This component was also new in Cadstar Library and its symbol and footprints were added in the library using the Ultra Librarian(UL) Software. The .bxl files were used as inputs in this software for this component.

For simulation purposes, AD605 was added as a new component(AD605AN.olb) in the library of OrCAD tool and this file is also attached in the DVD. Each unit corresponds to a single channel.

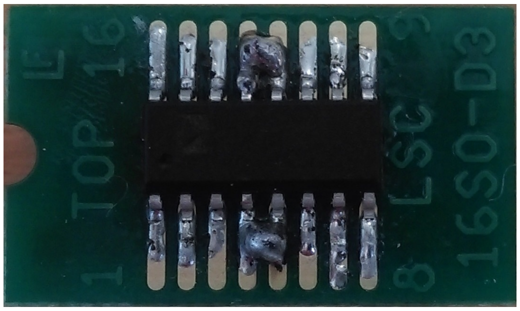


Figure 3.1: AD605 with SOIC to DIP adapter

Voltage Regulator

VREF voltage ranging between 2.5V-1.25V can be provided with the help of LM317 Voltage regulator. LM317 is a monolithic IC which is used as a positive adjustable voltage regulator. It can output a fixed output voltage from 1.2 to 37 Volts DC. The nominal output voltage is selected by means of a resistive divider, making the device exceptionally easy to use and eliminating the stocking of many fixed regulators[19]. A resistor and a potentiometer or simply two resistors are used for obtaining the required output voltage and follow the relation:

$$V_{out} = 1.25 * (\frac{R1 + R2}{R1}) + I_{ADJ}R2 \qquad [52, Pg. 4] \tag{3.1}$$

LM317 comes in various packages but TO-220 package is used.

3.1.3.2 Anti-aliasing Filter

Before giving the input into ADC, the signal is passed through an Anti-aliasing filter. An Anti-aliasing filter is a low pass filter of a particular order that has a cut-off frequency which removes the unwanted signals from the ADC input or at least attenuate them to a point that they do not degrade the circuit performance.

An ideal anti-aliasing filter passes all the appropriate input frequencies(below cut-off frequency, f_c) and cuts of all the undesired frequencies(above f_c). However,such a filter is practically not possible. The choice of the filter depends on the frequencies of interest and allowed attenuation in the passband. There are various types of filters like Chebyshev, Butter worth, Elliptic and Bessel filters. Each one has its own advantage such as Butter-worth offers flattest passband region among all the filters.

The incoming signal to the ADC is passed through a first-order low pass filter comprising of a resistor(R) and a capacitor(C). A first-order anti-alias filter is chosen for the simplicity of the design and because of the oversampling of the input signal i.e. using 5MHz sampling rate. Data-sheet[13, Pg 12] of chosen ADC recommends values of 25 Ohm(Ω) and 12pF for resistor and capacitor respectively but this gives cut-off frequency in hundreds of MHz. 15Ω resistor and 47nF capacitor were used to form the anti-aliasing filter.

$$\omega = 1/RC \tag{3.2}$$

where $\omega = 2\pi$f;

This filter simply provides a cut-off frequency around 225.7kHz and attenuates the undesired signals. Simulation uses the same setup shown in Fig.2.9 and the circuit files are present in Attachment V.
A simple plot of the filter's operation is shown below:-

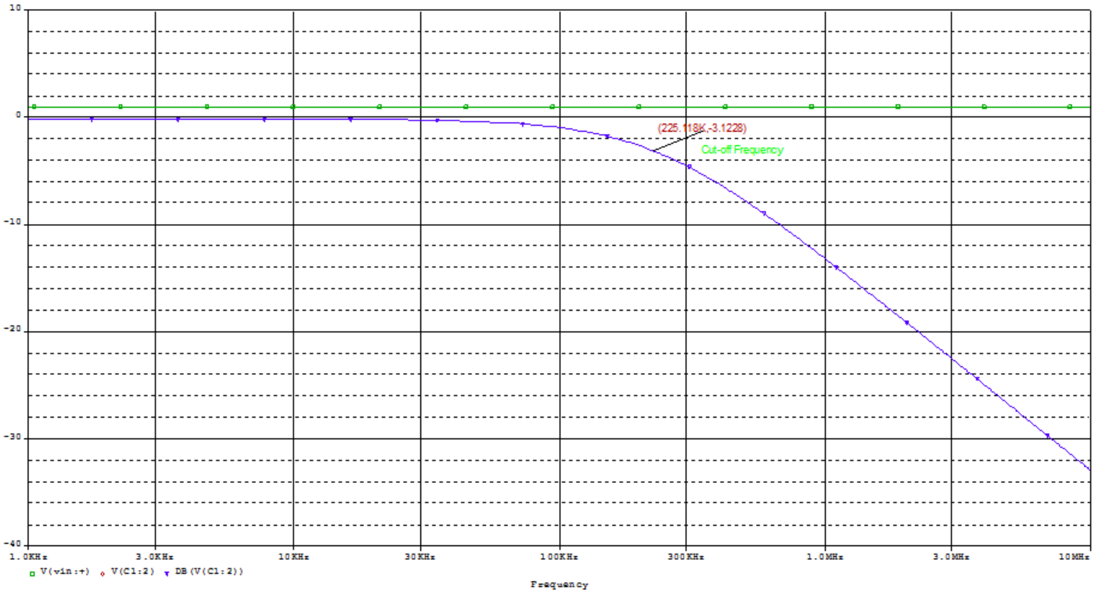


Figure 3.2: Anti-aliasing filter

A more advanced solution would be using an Active RC filter such as LTC1563-2/LTC 1563-3. These are 4th order Low-pass Filters and can easily be used for Anti-aliasing applications. These filters require only a single resistor value to set the cut-off frequency[30, Pg 1]. The cut-off frequency can be set anywhere between 256Hz and 256kHz using the following relation:-

$$R = 10k * (256kHz/f_c) \tag{3.3}$$

; f_c = desired Cut-off frequency

The other advantage of this Filter is that with different resistor values, different gains can be achieved. It can operate from a single 3V to $\pm 5V$ Supply.

3.1.3.3 Analog to Digital Converter

There could be more additional steps in the analog domain for signal conditioning, but it will be advantageous to perform them in the digital domain to get accurate results. So, using ADC directly after amplifier is a better choice.

Analog to Digital Converters(ADCs) are widely available in the market. For this project, the input frequencies used are between 50kHz and 200kHz. So, according to Nyquist Theorem an ADC which can sample the input signals with two or more times the maximum input frequency has to be used. To be on the safe side, oversampling is done which is significantly higher than the Nyquist rate and is usually 10 times the input signal frequency. Therefore, an ADC which can sample in the area between .5 Msps and 2 Msps has to be used.

Because ADC transmits data very fast and it can be placed close to the processing unit, a parallel architecture can be implemented so that a new digital value is transmitted on every edge of the sampling clock. As more number of bits result in more precise representation of the analog signal, it is desired that ADC with higher number of bits is used.

With all these above demands, it becomes easier to choose the right ADC and LTC2245 was decided to be used in the project as it was already available in the E-Lab and was also previously used in a Master project[15, page 31].

LTC2245 is a 14-bit, 10Msps ADC. It needs a single 3V Supply making it compatible with the rest of the components. It provides a SNR of 74.4dB and 90dB SFDR for signals well beyond the Nyquist frequency. Moreover, it has a flexible input range between ±.5V and ±1V, which can be selected with the help of SENSE, V_{CM} and V_{DD} pins. A separate output supply allows the outputs to drive 0.5V to 3.6V logic[13, page 8].

LTC2245 comes in 32-Lead Plastic QFN Package with 5mm x 5mm body and a pitch of .5mm. The symbol and footprint of this component were added in the CADSTAR Library using the Library Editor.

A 32-pin QFN to DIP adapter was first exclusively purchased for the time interval the schematics and PCB design of this project were under construction and this chip was then soldered on this adapter so as to test the working of the ADC.

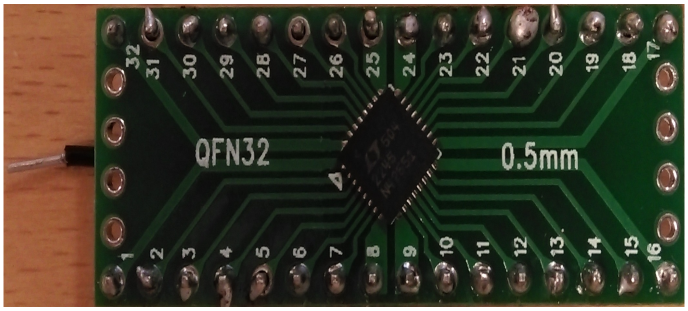


Figure 3.3: LTC2245 with QFN32 to DIP adapter

Alternate option

Besides the chosen ADC, an alternative could be to use AD7928 from Analog Devices already present on the Altera board.

AD7928 is 8-channel 12-bit A/D Converter with a conversion throughput rate up to 1Msps. This 20-pin ADC receives input as eight single-ended analog channels and produces output as a serial data stream. The analog input range for all input channels can be 0V to 2.5V or 0V to 5V, depending on the RANGE bit in the control register[27]. These eight input channels are connected to a 2x5 header with VCC5 and GND as the other two pins. This ADC is connected to FPGA and it sends and receives signals with pins $\overline{CS}$, DOUT, DIN and SCLK[2, Pg. 42].

This option was discarded because of the low conversion throughput rate and a number of unnecessary input channels as only one channel was required for receiving the input.

3.1.3.4 Digital to Analog Converter

Digital to Analog Converters(DACs) are also useful for signal processing. The basic use of DAC here is to produce a TVG voltage from the digital logic and for testing of the system to see the results on the oscilloscope.

Like ADC LTC2245, it is important that DAC can sample at higher clock frequencies, should also be 14-bit and operates at the same voltage supply of 3.3V. It is also desired that the communication interface of DAC should be same as that of ADC i.e. parallel, but this option is demanding as DAC needs to be handy and must have lesser number of pins for interfacing because it has to be used a number of times in different processes. Therefore, a DAC which communicates serially over a few wires is chosen to be used.

Among the available choices, DAC AD5641 from Analog Devices was used in this design as it uses a versatile 3-wire serial interface and is compatible with SPI Bus, MICROWIRE standards, etc. AD5641 is a 14-bit nanoDAC and operates from a single 2.7 to 5.5 V supply and with a maximum current consumption of 100μA. Its high speed serial interface operates with clock rates up to 30 MHz.

AD5641 communicates using three pins $\overline{SYNC}$, SCLK and SDIN. And produces analog output at V_{out} pin. $\overline{SYNC}$ signal is active low and when it goes low, it enables the input shift register and data is transferred[8, Pg 6].

AD5641 comes in two: 6-Lead SC70 and LFCSP packages. SC70 was used and its symbol and footprints were imported in CADSTAR library using the Ultra Librarian Software.

Other options available are AD5040 or Digital potentiometers. AD5040[49] has almost the same characteristics as AD5641 but with a different 8-Lead Package and higher price. Digital potentiometers can also do the same job of providing the variable gain to the amplifier. They are just like regular potentiometers but they use digital protocols like I2C or SPI and are controlled by a digital device.

A digital potentiometer has a certain number of steps in the resistor string and the number of resistors determine its step size or resolution. The resistor string represents the end-to-end potentiometer resistance. The digital input comes from a Micro-controller or FPGA[34].

Digital potentiometer is a good choice but the end choice was made on AD5641 because of the higher number of bits available using the DAC whereas digital potentiometers available from vendors like Analog Devices, Maxim Integrated at the time of designing had only 1024 as maximum number of steps which corresponds to 10 bits[35]. Such a low resolution may affect the results while amplifying the signal.

In this case, the DAC unit was soldered on a SC70 to DIP adapter for testing.

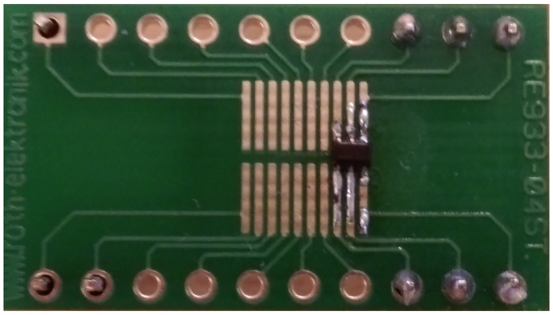


Figure 3.4: AD5641 with SC70 to DIP adapter

3.1.4 Choice of FPGA

Each FPGA vendor tends to have its own architecture. A number of companies dealing with FPGA technology are available today.

Xilinx and Altera are the two biggest companies which are dominating the market in the FPGA technology. Altera DE1-SOC board was already available at the Physics Institute and thus used in the project. The DE1-SoC Development Kit presents a hardware design platform which is built around the Altera System-on-Chip(SoC) FPGA, combining the latest dual-core Cortex-A9 embedded cores with industry-leading programmable logic for ultimate design flexibility. Also, similar boards from Altera were used in the course FYS4220.

Altera DE1-SoC development kit comprises of Cyclone V SoC 5CSEMA5F31C6 device and has 85K Programmable Logic Elements which are more than sufficient for this project. This kit requires 12V DC input and has a USB-Blaster II port which is connected with a USB cable to the PC. The FPGA device can be configured through JTAG interface on DE1-SoC board, but the JTAG chain must form a closed loop, which allows the programming software to detect the FPGA device[2, Pg. 15].

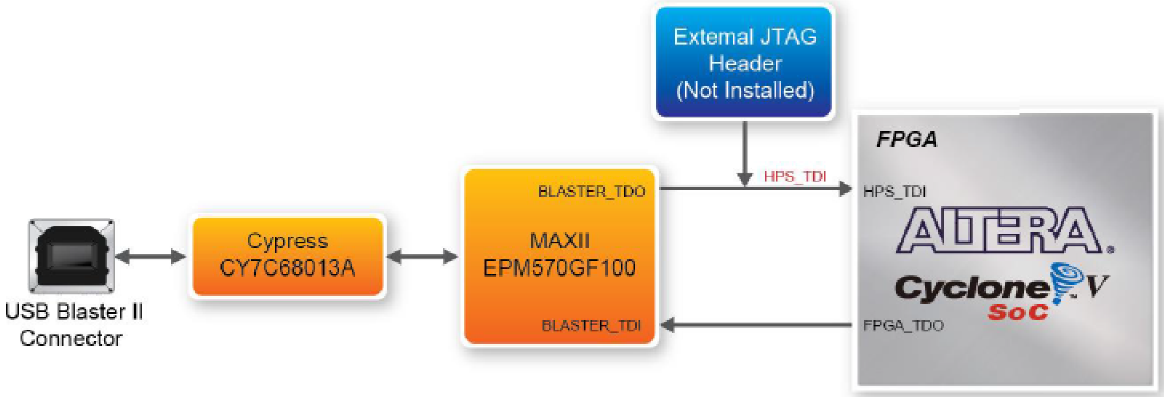


Figure 3.5: Path of the JTAG chain[2, Pg 15]

It has many on board hardware devices like Push buttons, LEDs, 7-segment displays, etc. But the ones used for interfacing with the developed front end circuit are the two 40-pin Expansion Headers. These headers are capable of providing 5V, 3.3V and ground signals.

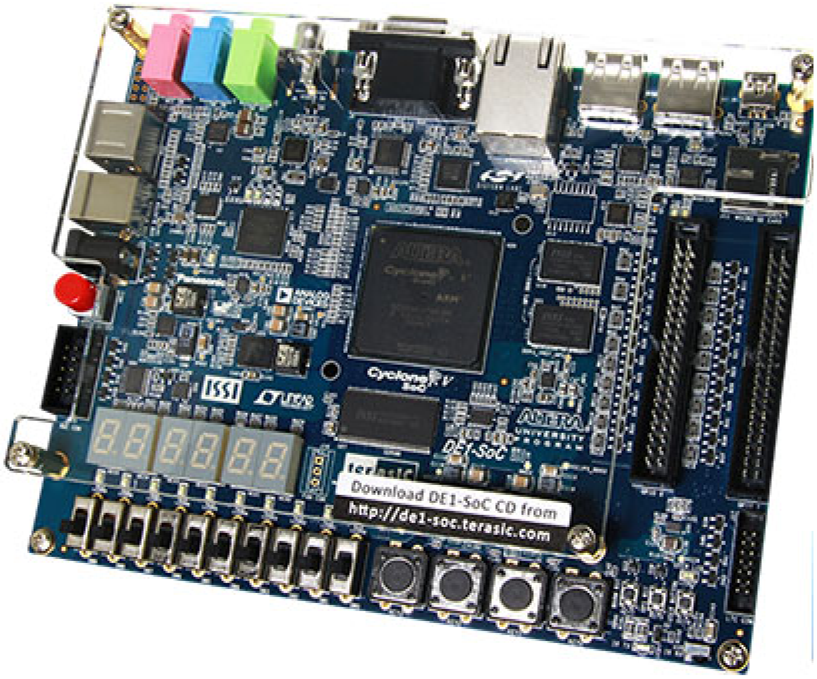


Figure 3.6: DE1-SoC Altera Development Board

3.2 Hardware Development

To begin with the Hardware Development, first of all, the designed symbols and footprints of various new components were added in the private library ’PRIVAT’ and then Schematics of various circuits were designed using CADSTAR.

The circuits of transmitter, receiver, etc. were drawn on different sheets and connected on top sheet with each other. The schematics of these parts can be found in Attachments(I and III) section and in DVD.

3.2.1 Schematics of Transmitter Part

Schematics of various components comprising the Transmitter part are described hereunder:

3.2.1.1 MD1213

The power connections VDD1 and VDD2 of MD1213 are connected to +5V and should have a ceramic bypass capacitor of 0.47μF to the ground plane. A common capacitor and voltage source is desired for these two pins.

The supplied voltages of VH and VL determine the output logic levels. These two pins are capable of drawing fast transient currents of up to 2.0A, so it is necessary that they are provided with an appropriate1.0μF bypass ceramic capacitor. VH pin should be connected to 5V while VL pins needs to be connected to ground.

Similarly, VSS1 and VSS2 pins should also be connected directly to the ground plane. The trace lengths are kept minimum so as to have low inductance[11, Pg 6].

The inputs INA, INB and enable signal(OE) are connected to the logic source FPGA. These signals have two states: GND and high logic ’1’ i.e. 3.3V[11, Pg 6]. As the VH and VL pins are at +5V and GND respectively, therefore the output levels of OUTA and OUTB should swing between VH and VL.

The outputs are finally connected to AC coupling capacitors each of 10nF before connecting them to the Gate drive circuit of FETs.

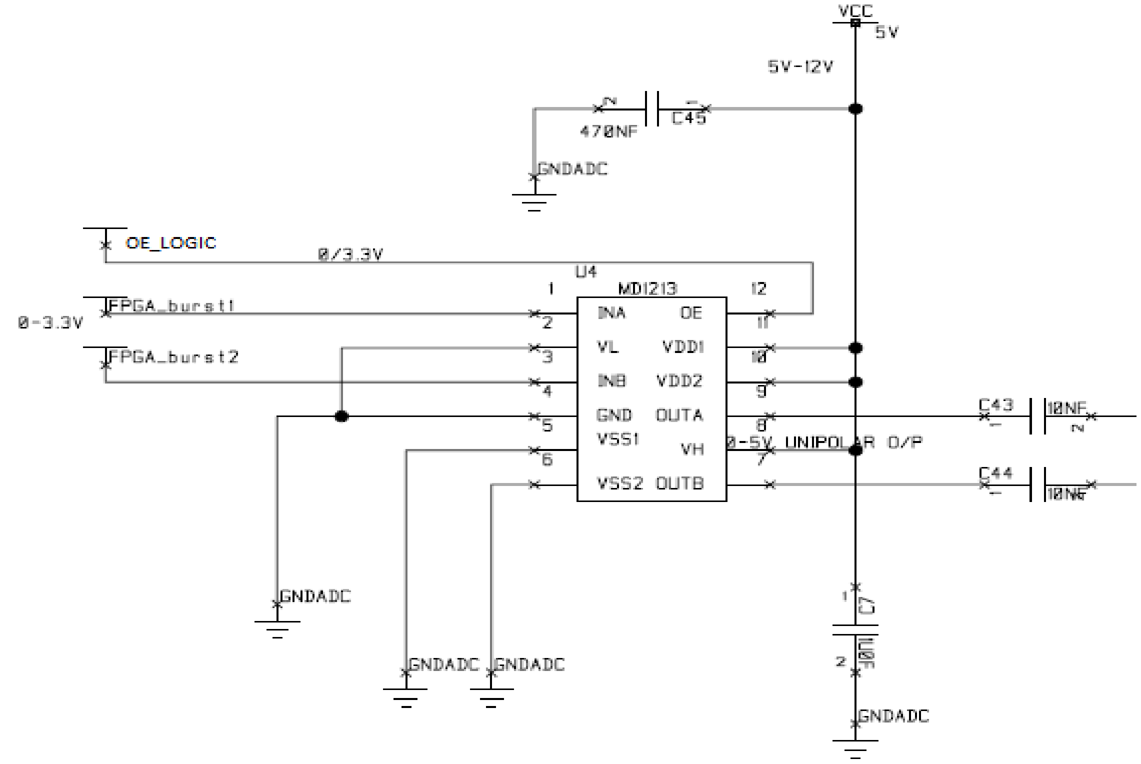


Figure 3.7: MD1213 Schematics

3.2.1.2 TC6320

The Gate pins(G1 and G2) of MOSFET pair receive the input signals from the driver. Source pins(S1 and S2) of TC6320 are connected to a 3-pin Connector CN2 which will receive high positive and negative voltages from the designed external power supply. These external high voltages are first passed through a 1.0μF bypass capacitor and a 100V Zener diode before they are connected to S1 and S2. This chip has also in-built protection against transient voltages.

The outputs of MOSFETs i.e Drain pins(D1 and D2) are first passed through BAS16 diodes and then connected with each other to form a single high amplitude pulse train which is further sent to the transducer.

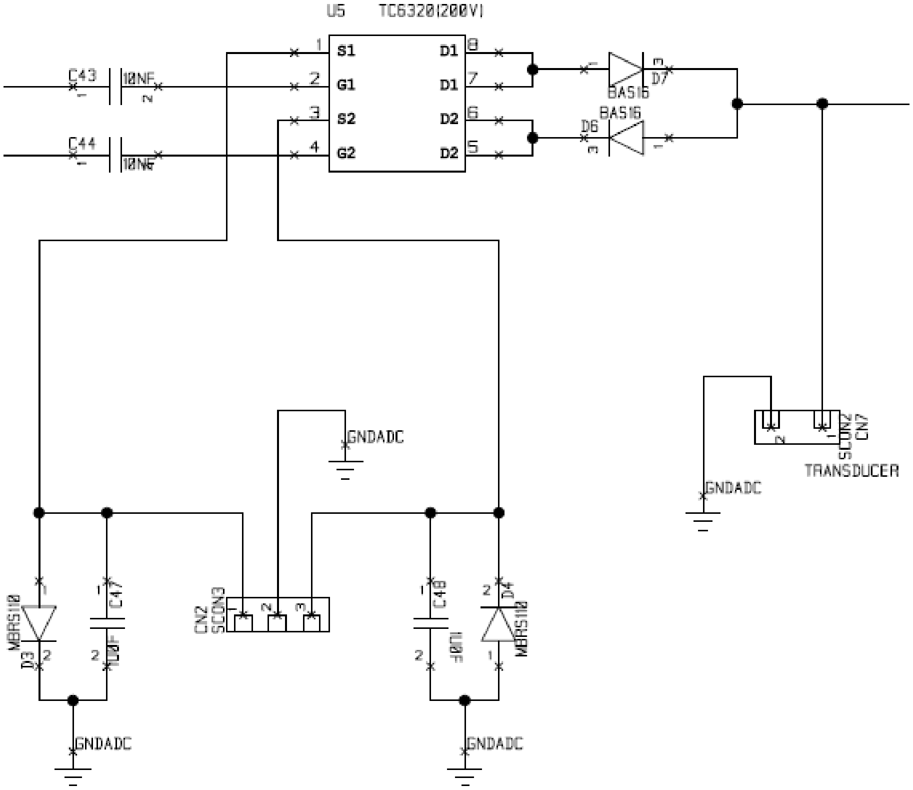


Figure 3.8: TC6320

3.2.2 Schematics of Receiver part

The receiver schematics consists of components: MD0100, AD605, LTC2245 and AD5641.

3.2.2.1 MD0100

T/R Switch, MD0100 has a very simple circuit with only two connections on Input and Output terminals. The back-to-back diodes on the output side of the switch were used to complete the circuit. The COM terminal of the switch is intentionally left floating.

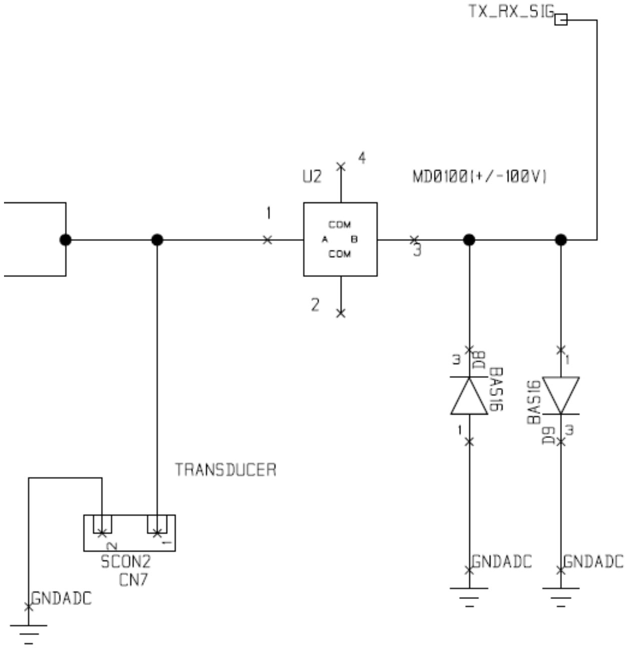


Figure 3.9: MD0100

3.2.2.2 AD605

AD605 is operated from a single 5V supply and needs a couple of decoupling capacitors to ground. AD605 consists of two amplifiers which are connected in series to double the gain range.

It is used as a single-ended amplifier with input coming from switch at +IN while the other input -IN is AC grounded. The input signal needs to be AC-coupled to accommodate ground-based signals. The capacitor C27 level-shifts the input signal from ground to the DC value, 2.5V which is set by VOCM. The capacitors at inputs together with the 175Ω looking into each of +IN and -IN inputs[7, Pg 14], act as high-pass filters with cut-off frequencies dependent on the value of capacitors. The value was chosen to be 1.0μF which gives 9kHz as the cut-off frequency.

Gain Pins, VGN1 and VGN2 are shorted and connected to a common Gain Control coming from DAC. Alternatively, these pins can also be connected to a potentiometer with the help of a 0Ω resistor to change the Gain manually. The output of potentiometer is connected to a 1nF ceramic bypass capacitor . The gain control interface is set with a gain scaling factor of 20 dB/V by providing an input voltage of 2.5V to VREF pin. The Gain increases linearly with increase in VGN.

The gain range doubles by connecting the two channels in series and providing VGN equal to 0.1 to 2.9V with the condition that FBK-to-OUT connection is left open. The VOCM pin is connected to a decoupling capacitor to ground. The output from the first channel is fed back to the input of the second channel. And the final output is taken at pin OUT2 which is again AC coupled using a capacitor for ground reference.

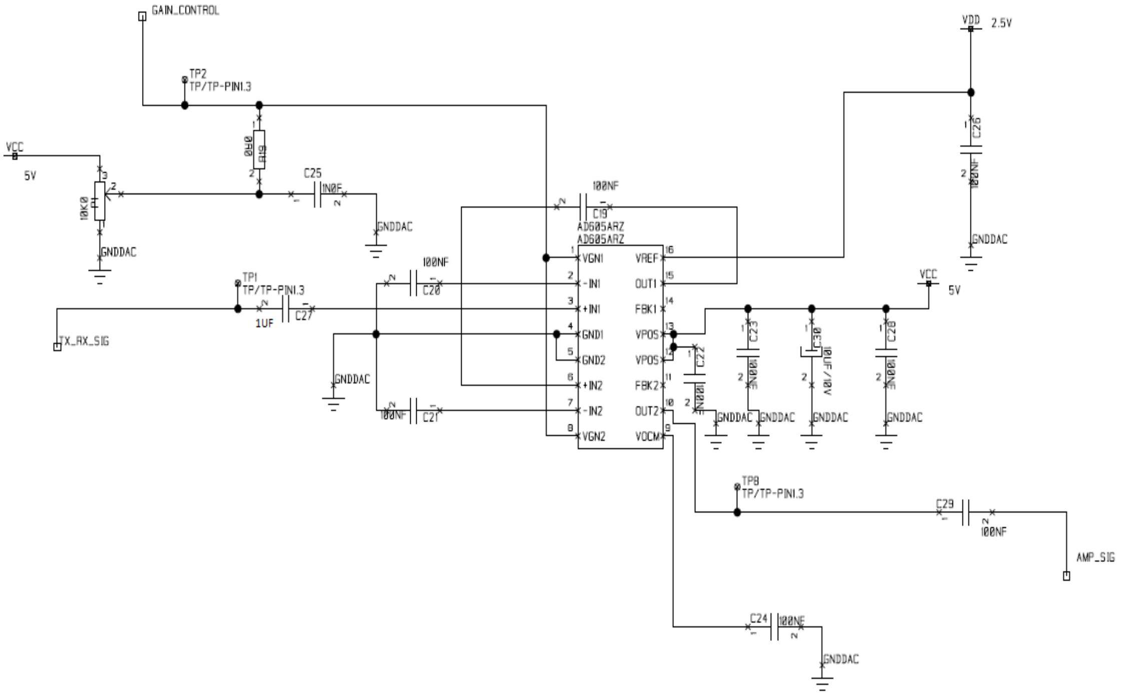


Figure 3.10: AD605

Voltage Supply for VREF

The 2.5V supply to VREF pin is obtained by using a LM317T voltage regulator. The regulator uses the +5V as input and needs two 220Ω resistors to reduce the input voltage to half, giving a 2.5V output.

The current flowing out from ADJ pin(1) is very small and constant when compared with the current through R1, it represents a small error and can usually be ignored in the equation 3.1
An LED is connected at the output of LM317T to indicate that the voltage is going to the amplifier.

For obtaining gain scale factors other than 20 dB/V, VREF voltage needs to be changed and this voltage can easily be changed by changing the two resistors, R1 and R4.

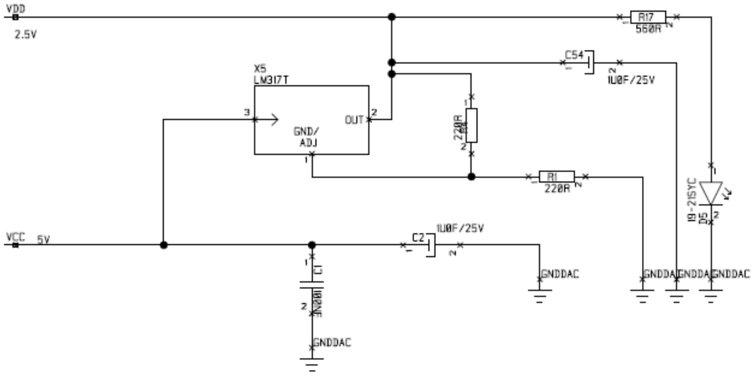


Figure 3.11: LM317T Voltage Supply Circuit

3.2.2.3 LTC2245

The output from amplifier is single ended, so ADC is also driven with single ended input, A_{IN+} and A_{IN-} is connected to V_{CM}. The input coming at A_{IN+} is passed through an anti-aliasing filter and ADC input protection circuit as discussed earlier. REFH(High Reference) pins are shorted together and bypassed to REFL(Low Reference, also shorted) pins with a 0.1μF ceramic chip capacitor and an additional 2.2μF ceramic capacitor and to ground with a 1μF ceramic capacitor. Similarly, REFL pins are also bypassed to ground with a 1μF ceramic chip capacitor.

Supply voltage, V_{DD} and OV_{DD}(pins 7,32,20) are connected to 3.3V supply from board and bypassed to Ground with 0.1μF ceramic chip capacitors. GND and OGND(pins 8,20,33) are ADC power ground.

ADC receives Clock input of 5MHz at CLK(pin 9). SHDN and $\overline{OE}$(pins 10,11) need to be connected to ground during ADC's normal operation enabling the outputs. So, they are controlled digitally. Digital outputs are received at DO-D13(pins 12-19,22-27) with D13 as MSB. OF(pin 28) is used for Over/Under Flow Output and becomes High when this condition has occurred.

MODE(pin 29) is used for setting the output format. Offset binary output format was selected by connecting MODE to GND. SENSE(pin 30) was connected to V_{DD}. This will select the internal reference and a ±1V input range. V_{CM} is 1.5V Output and is Input Common Mode Bias and bypassed to ground with 2.2μF ceramic chip capacitor. All the ceramic capacitors are placed as close as possible to the pins.

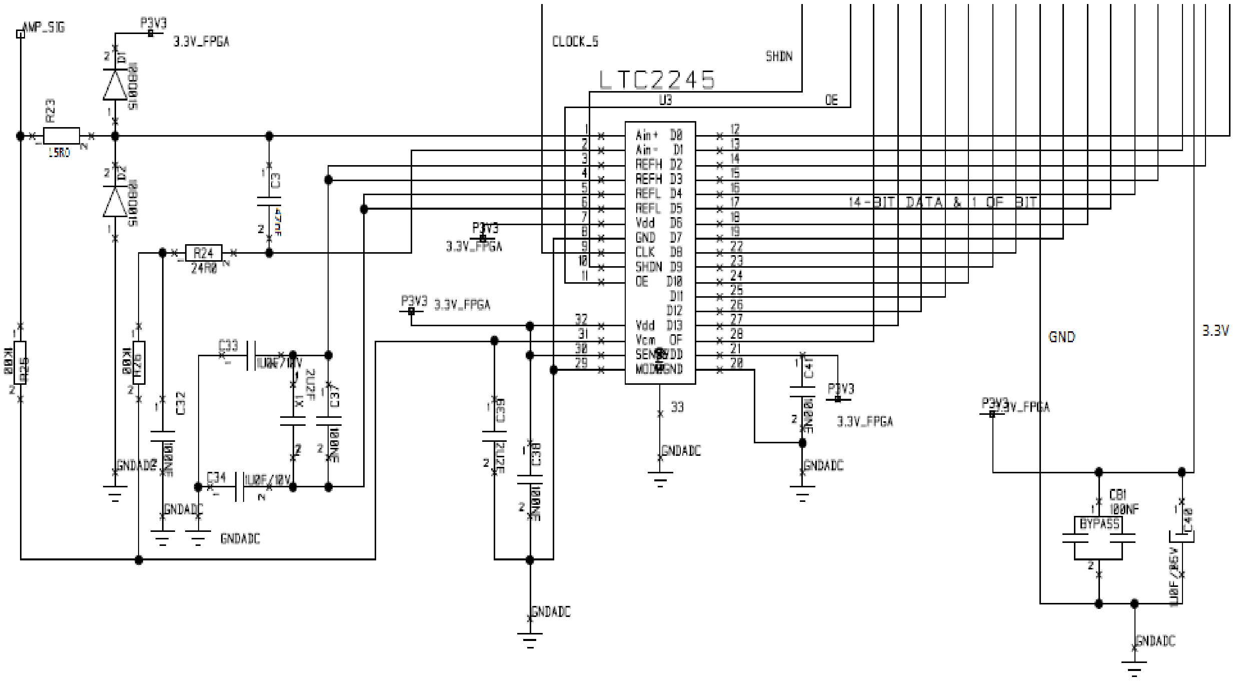


Figure 3.12: LTC2245

3.2.2.4 AD5641

AD5641 can be powered by a power supply between 2.7V and 5.5V. As ADC is also operating at 3.3V, V_{DD} pin of AD5641 was also given a 3.3V power supply. The voltage supply is passed through a couple of ceramic bypass capacitors before it is applied to the DAC so as to remove any voltage fluctuation.

The inputs SYNC, SCLK and DIN are connected to the FPGA board for serial communication with SCLK providing 5MHz clock frequency. The V_{OUT} pin provides the analog output voltage.

AD5641 can provide an output in the range 0 to 3.3V and the amplifier just needs Gain voltage between 0 and 2.9V.

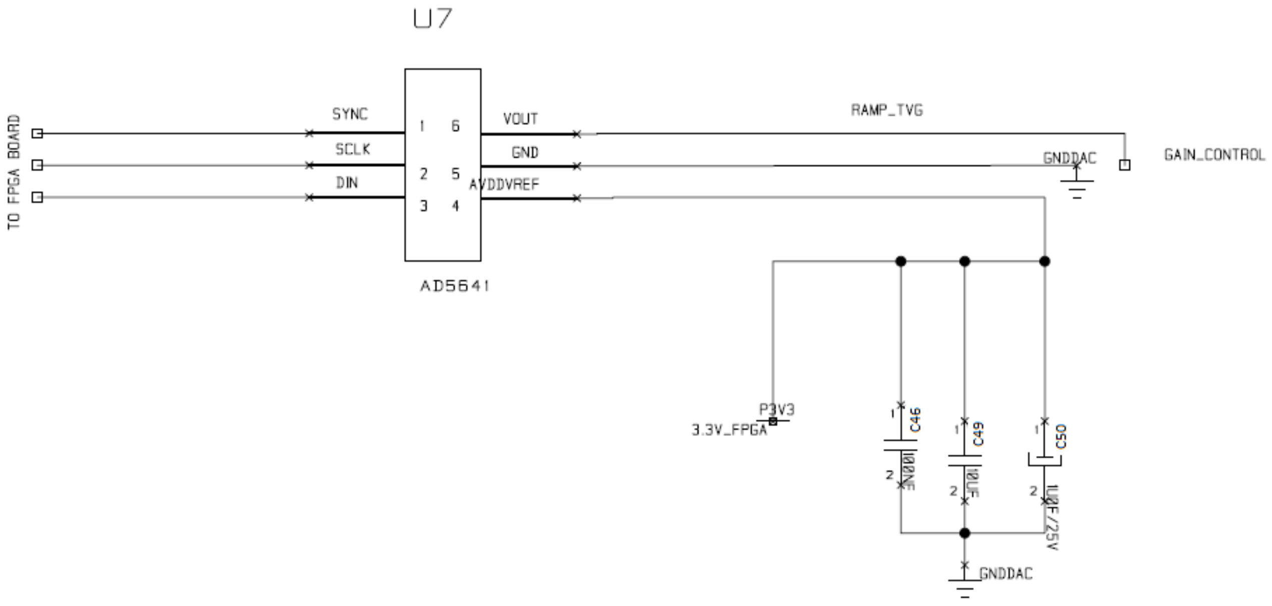


Figure 3.13: AD5641

3.2.3 High Voltage MAX1771 Board

A separate board for delivering high positive and negative voltages was made. This board uses SMPS technique for voltage conversion as discussed in Section 2.2.2

Schematics

The design uses IC MAX1771, MOSFET, Inductor, diode and a couple of resistors and capacitors for its full operation. The necessary symbols and footprints for the used components were made and imported in the CADSTAR library. This board was designed during the ending phase of this project.

IRF644PBF is a N-channel MOSFET and has a low $R_{DS}(on)$. This is crucial for its operation because the speed at which the FET switches directly effects efficiency. Components present at the output side like T1, C4, R2 must be rated at > 220V ratings, otherwise they will break down at higher voltages. D1 should be an ultra-fast(<50ns) recovery diode. Components C1, U3 and T1 should be low ESR type. Potentiometer VR1 should be multi-turn for accurately adjusting V_{out}. Capacitors U3 and C2 should be placed as close as possible to pin 2 of MAX1771[22].

This board will operate from the same Voltage supply as transformer.

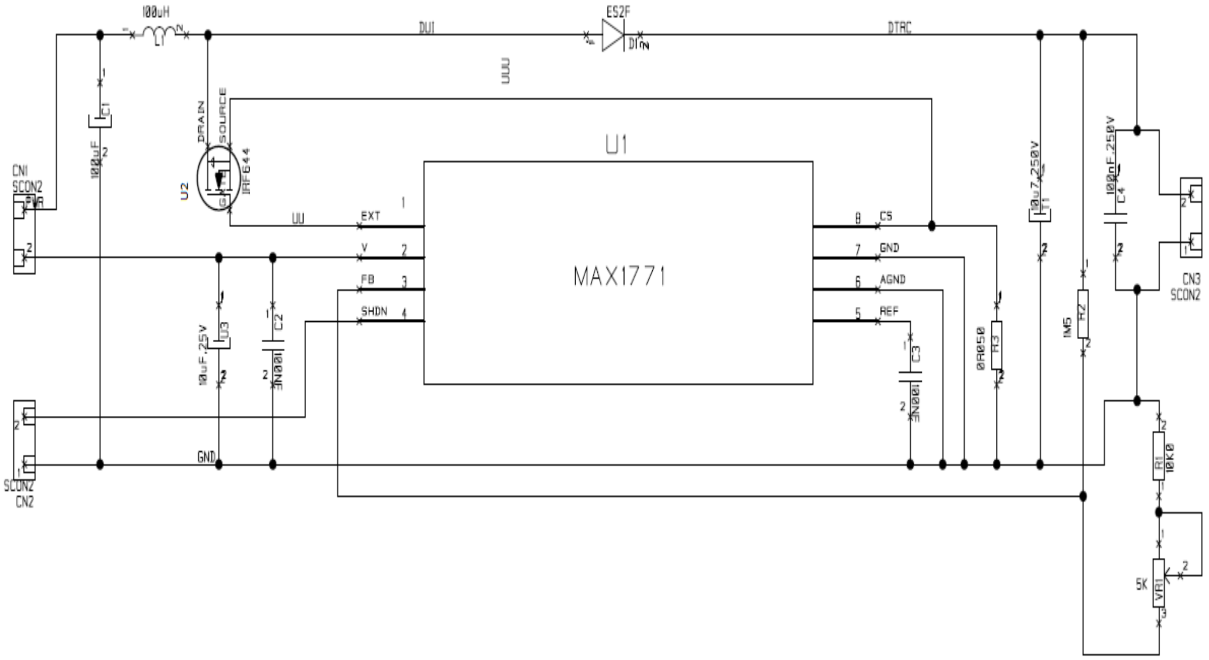


Figure 3.14: HV MAX1771 Board

3.2.4 Designed High Voltage MAX1771 Board

The layout and choice of components is very important in the design and were followed as described in the MAX1771 data sheet[20] and Nixie HV Switching PSU[22] to get the maximum efficiency. As recommended, extra care was given to the ground plane at the bottom layer and signal tracks at the top electric layer. Bottom ground plane is very important in this layout and it has been made uninterrupted without any signal tracks from above layer. Similarly, high power traces have been made wide so as to minimize the inductance losses.

After the schematics, PCB and Gerber files were made. The size of the board was made small i.e. 5cm x 5cm, so as to prevent losses. The PCB card is designed on a 2-layer board as there are only a few components and signals. Most of the components chosen were of Surface Mounted Devices(SMD) type except FET and connectors. The trace between MAX1771 and FET has to be short and FAT so as to have low resistance and inductance. Therefore, the gate charge is given to the FET in the shortest possible time. The board was eventually sent for production with the help of E-Lab as it was the fastest solution among all the options.

After the PCB arrived, the soldering of components was done by applying solder paste on pads and SMD components were then mounted on the board and placed inside the Soldering Machine. After that, the through-hole components like MOSFET, Connectors were mounted.

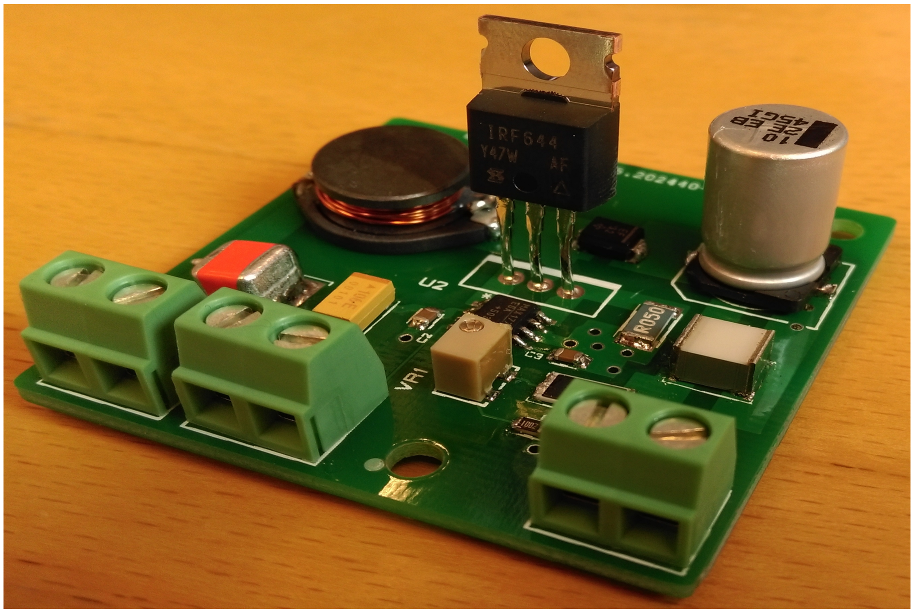


Figure 3.15: HV MAX1771 Board

For testing this board, it just needs to be connected to a voltage supply and no other external control is required for its operation. Thus, making this board a simple, efficient and reliable option for boost conversion. The corresponding layout is present in Attachment IV and DVD.

3.2.5 Designed Echo sounder Board

The Echo sounder system contains more components as compared to the High Voltage Board. So, the size of the PCB board was self decided with all the components present only on the upper side of the board.

The Echo sounder board was also decided to be made on a 2 layer board. The placement of components was done in a way that there is a sufficient distance between the adjacent components and the ground signals are well distributed all over the board. In this design, the power and control signals are present on both the top and bottom layers. Necessary test points were added in the layout. E.g. at Input and output of amplifier.

The connectors are placed on opposite sides of the board. The 40-pin connector which is used to connect to the FPGA board for sending and receiving signals is kept on one side and the connectors used for connecting to External high voltage supply and for transmitting/receiving pulses are kept on the other side.

In this case also, the PCB board was sent for production with E-Lab's help. Upon arrival, the same procedure was followed for soldering the components as discussed earlier. The solder paste was applied on the component pads by placing a stencil over the PCB board and then, pressing the paste into the gaps in the stencil.

After mounting of SMD components, the board was first checked for possible short circuits. Some pins of ADC i.e. QFN package were observed to be shorted. After few attempts, this error was successfully removed and then, the through-hole components were placed on the board.

Finally, the board was ready to use and could be used for testing purposes. The PCB design files can be found in Attachment II and DVD.

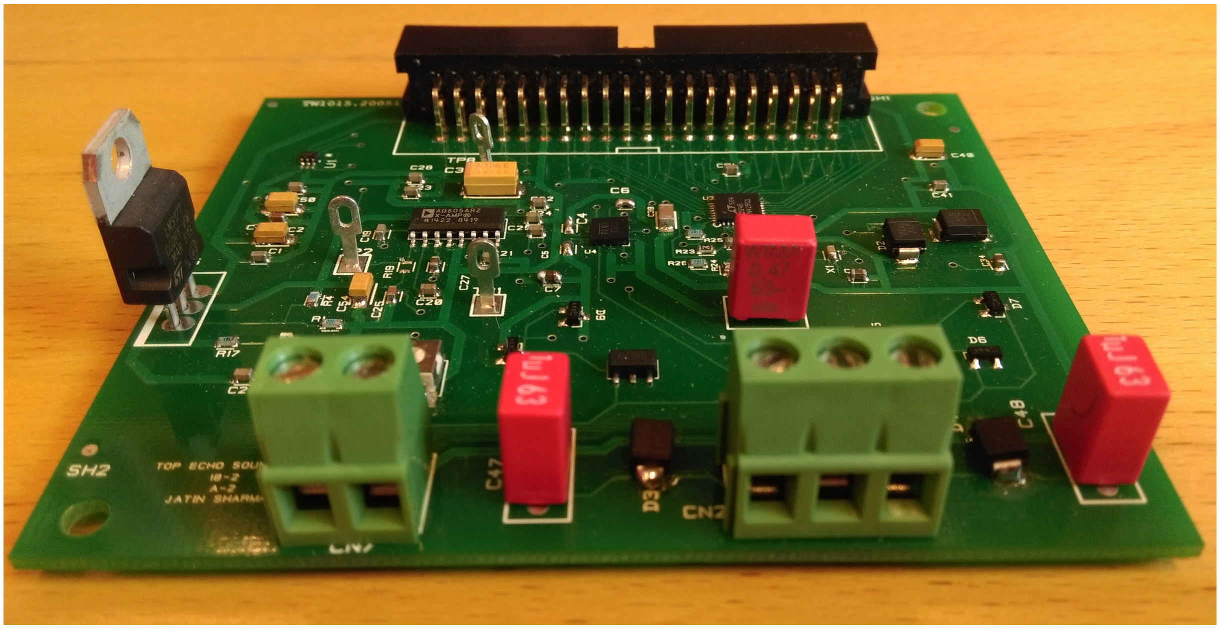


Figure 3.16: Echo sounder Board

3.3 Software Development

As discussed earlier, VHDL will be used to develop the software part. But before going to the actual testing of components, simulations have to be done.

3.3.1 Simulation

Simulation is a very good method for understanding the operation of various components and getting the better results faster without spending so much time. By performing simulation, it becomes easy to locate the error in the code as the outputs can be seen visibly. For this purpose, Pspice was used for testing the amplifier and Modelsim for the digital code.

3.3.1.1 Simulation of AD605

It was necessary to test the bandwidth and gain range of the amplifier at different gain voltages.

For this purpose, a simulation approach was followed and was performed in the PSpice software. The amplifier AD605 was included as a library component in Pspice for simulation purposes as it was not there in the already available libraries.

First of all, single channel simulation was done so as to get a gain in the range 0 to 48 dB. And then both channels were cascaded in series to get higher gains. The feedback pins on the amplifier were kept unconnected so as to get a gain range between 0 and 98.6dB. The output from the first amplifier needs to be ac coupled by a capacitor before it goes into the positive input of the second amplifier so as to avoid undesired offset voltages at different gain ranges. The output from the second Unit U2 can be directly applied to the input of the ADC. However, if the output needs to be referenced around 2.5V dc then the ac coupling capacitor can be removed.

In practice, this gain voltage(VGN) needs to be controlled by analog voltage which will increase with time such that gain increases linearly with VGN.

The set up in Fig. 3.17 shows two channels of AD605 connected in series as individual components.

VREF is taken as 2.5 V so as to obtain a better gain scaling of 20 dB/V.

The amplifier is tested with a gain voltage of 2V and is common for both amplifier channels.

The simulation files are present in Attachments(V) section.

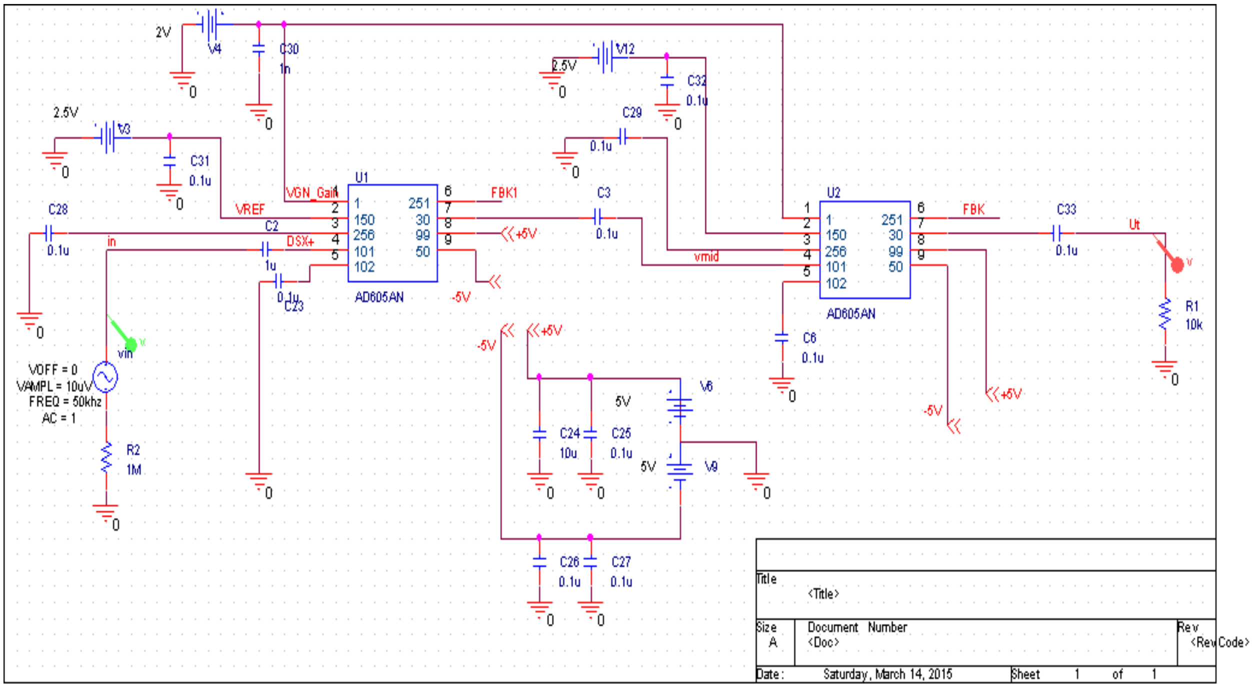


Figure 3.17: Simulation setup of AD605 Amplifier

As observed from the plot in Fig.3.18, the Gain obtained with a VGN of 2V is around 66dB. The bandwidth of the amplifier(-3dB) is observed to be between 9kHz and 5MHz. The operating frequencies for the echo sounder lie in between this range. If a lower cut-off frequency is desired, it can simply be changed by changing the ac coupling capacitors present at the Input pins.

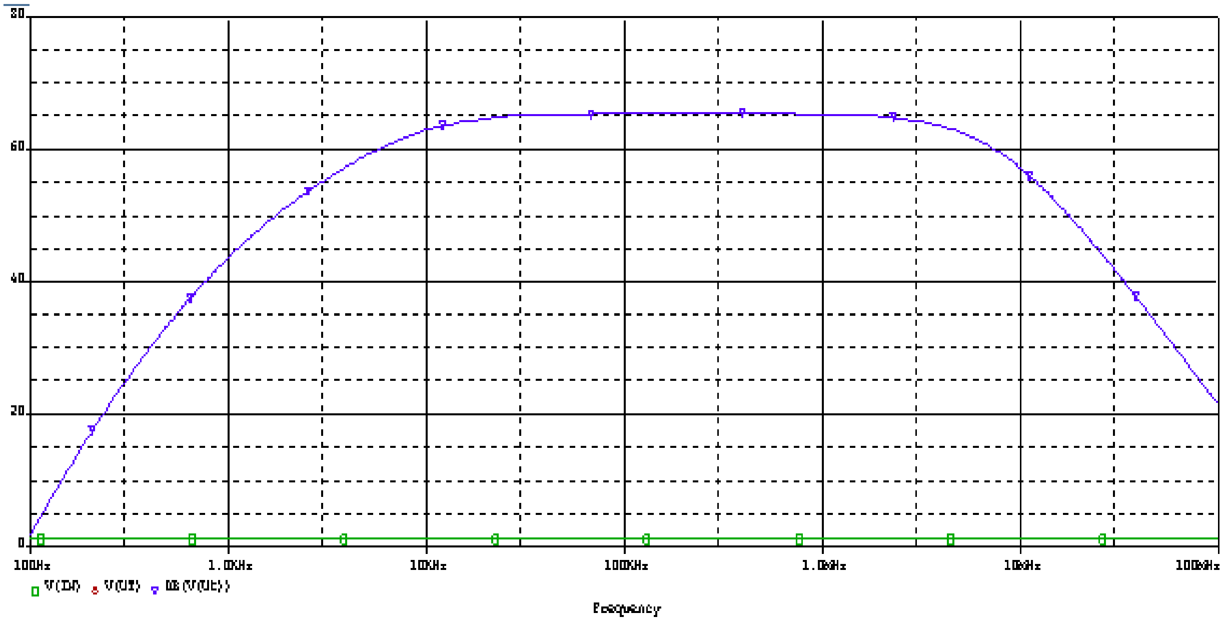
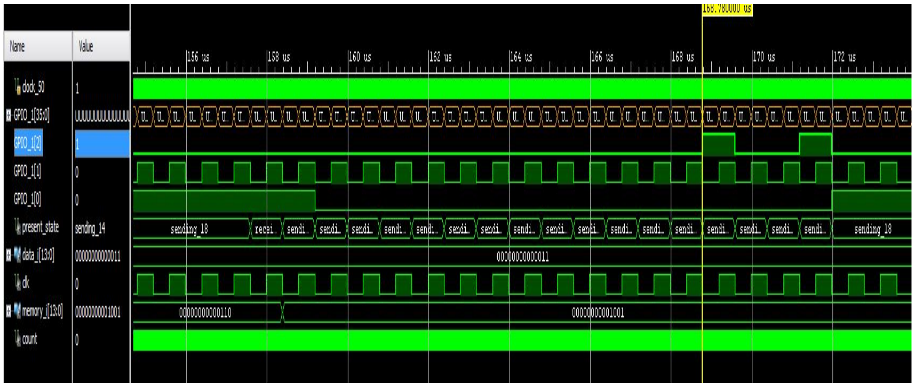


Figure 3.18: Graph showing relation between gain(dB) and Bandwidth of AD605

3.3.1.2 Simulation using ModelSim

Simulation of selected components was done using the software, ModelSim from Quartus. It was used to visibly check and understand various operations and therefore, helped in solving the problem whenever the intended operation was not performed.

A simple Write operation to DAC is shown here:



Name	Value
clock_50	1
GPIO_1[35:0]	UUUUUUUUUUUUUU
GPIO_1[2]	1
GPIO_1[1]	0
GPIO_1[0]	0
present_state	sending_14
data_i[13:0]	00000000000011
clk	0
memory_i[13:0]	00000000001001
count	0

Figure 3.19: Write operation to DAC using ModelSim

Fig.3.19 shows that the data changes on the falling edge of the clock, but it needs to be constant at that instant according to the data sheet[8]. This bug was removed and then tested on the hardware and the write operation was successful with DAC.

3.3.2 VHDL Code Development

After the board was built and simulation was done, it was time to check the operation of various components practically. For this purpose, it was necessary to communicate with the hardware and this was done using the VHDL language in Quartus. To make the software development easily accessible and editable for future use, the code development part was done in certain steps depending upon the components on the board to be driven.

Eventually, all the modules of the system were connected with each other using a Hierarchical Design method. The top level module is called Echo_sounder.vhd and is present in the Attachments(VI) Section and DVD.

3.3.2.1 Clock Divider

DE1-SoC board has four 50MHz clock signals connected to the FPGA which can be used as clock sources for user logic. Such a high frequency clock is not required in this project, so a clock divider logic was developed which divides the 50MHz clock down to 5MHz. The various components present on Echo sounder board need clock source for their operation.

ADC deals with inputs having frequencies in kHz range like 50, 200kHz. By providing a clock of 5MHz, these frequencies are sampled with a sampling rate which is much higher than the Nyquist frequency and thus gives a better resolution. Similarly, DAC also operates on 5MHz clock but it needs more number of clock cycles to transfer a digital value as compared to ADC. Still, 5MHz clock is sufficient for DAC's operations like TVG and during Envelope Detection. MOSFET Driver is also given the same clock. Thus, making the whole Echo sounder system work at a common clock frequency.

3.3.2.1 MD1213

The pulses are formed by sending the logic inputs to MD1213 chip using pins OE, INA and INB. The operation of the MD1213 is straight forward and can be understood from the following simplified block diagram.

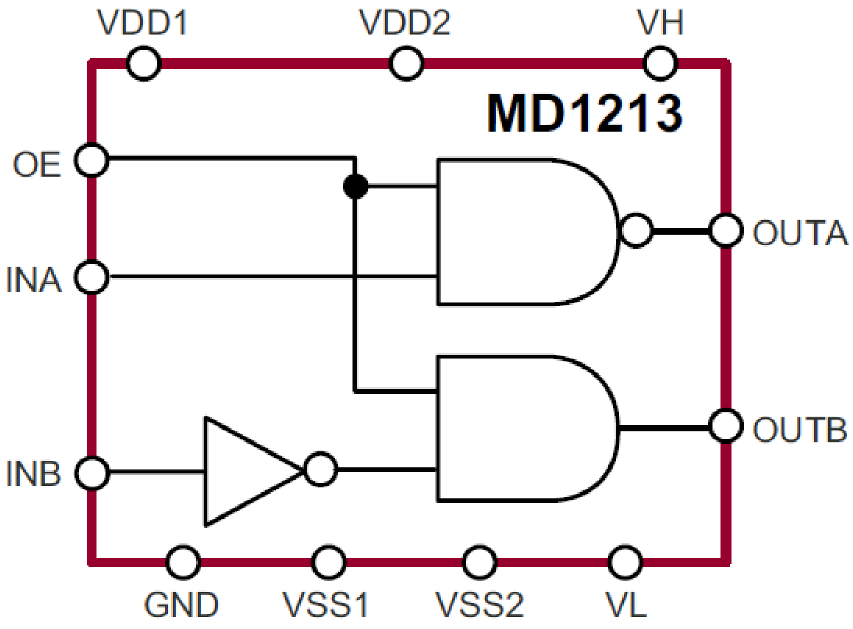


Figure 3.20: Simplified Block Diagram of MD1213[11, Pg. 5]

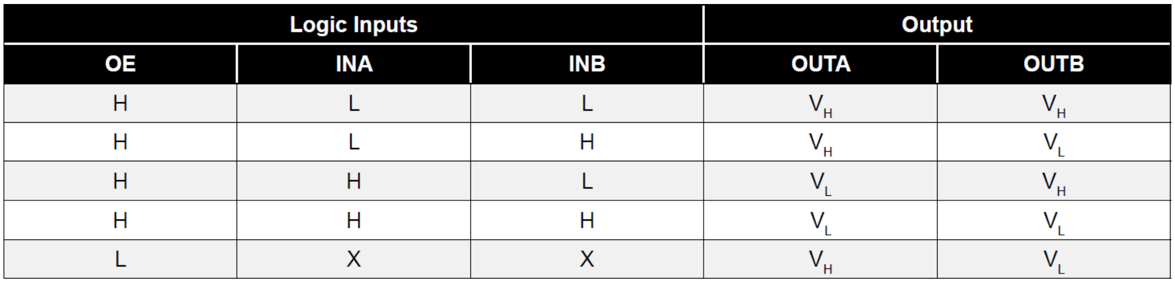
A VHDL code(Attachments VI), pulse_tx.vhd was developed to get the required pulses of frequencies 50 and 200kHz with certain number of cycles.

The duration of the pulses is set according to the bandwidth of the transducer.

Next is the state machine describing the VHDL code for pulse generation. The inputs INA and INB are initially set to be '0' and '1' respectively. The pulse generation process starts with OE going High('1') from Low('0'). The position of the Switch SW0 determines the frequency of the pulses. Here, high value of SW0 gives out 50kHz pulses and low value gives 200kHz pulses. The amplitude of these pulses toggle between 0 and 3.3V as available from the GPIO Header. When a certain number of cycles are produced, the inputs return back to their Initial values and OE also goes low thus disabling the outputs. After each pulse train is produced, the system goes to the Idle_state. Then, the same process is followed again to produce the next burst.

The pulses are generated at a PRF of 10Hz. The OUTA and OUTB pins can have unipolar or bipolar output voltage depending upon the connections at VH and VL pins. The whole process is synchronized using a 5MHz clock.

The corresponding truth table for the block diagram(Fig. 3.20) is shown as:-



Logic Inputs			Output	
OE	INA	INB	OUTA	OUTB
H	L	L	V_H	V_H
H	L	H	V_H	V_L
H	H	L	V_L	V_H
H	H	H	V_L	V_L
L	X	X	V_H	V_L

Figure 3.21: Truth Table of MD1213[11, Pg. 3]

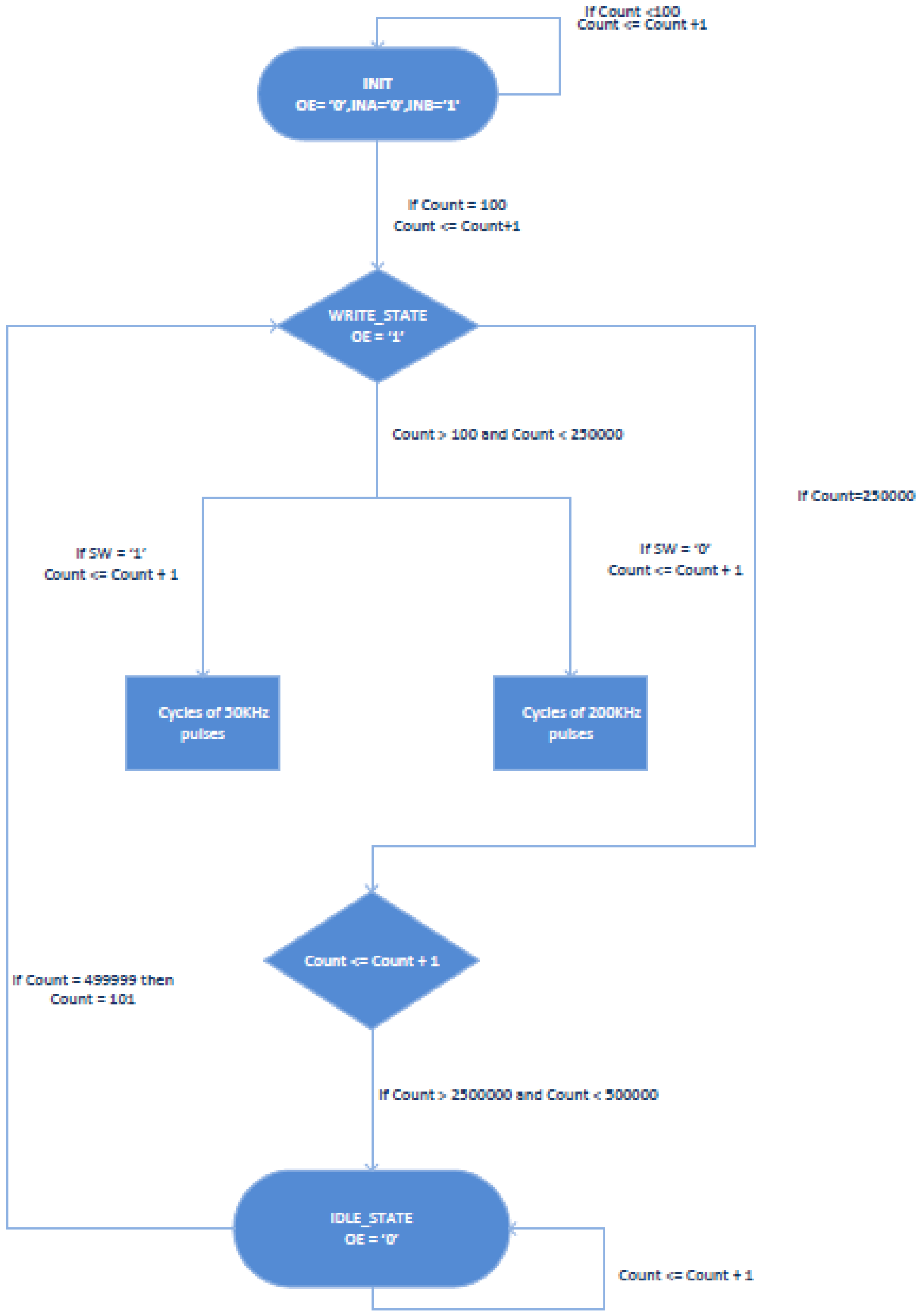


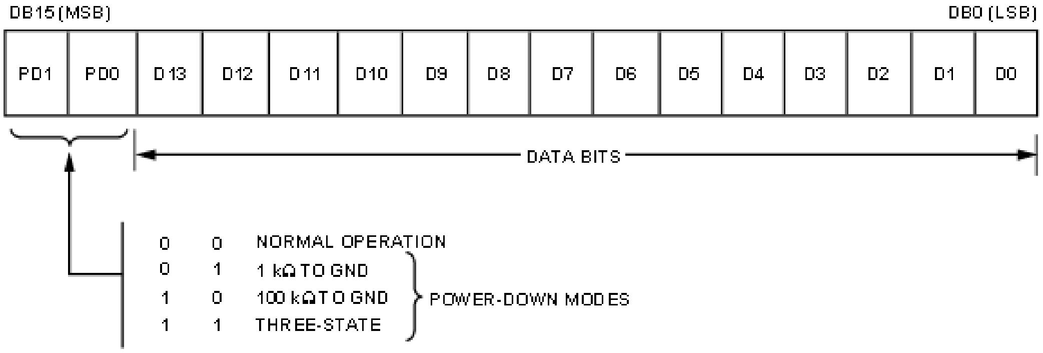
Figure 3.22: State Machine for generating 50 and 200kHz pulses

3.3.2.3 DAC AD5641

AD5641 is a nanoDAC and communicates using only 3 pins. Because of its very small size, it was soldered upon the adapter for testing purposes to be handy. Supply voltage of 3.3V was given to DAC which is available from FPGA board. DAC receives serial data and control signals from the FPGA board.

Write Operation

The input shift register is 16 bits wide. AD5641 needs 16 clock cycles for a successful write operation. The first two bits PD1 and PD0 are control bits and need to be ’0’ for normal operation. The following 14 clock cycles are used for sending the 14-bit data.



PD1	PD0	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

0	0	NORMAL OPERATION
0	1	1 kΩ TO GND
1	0	100 kΩ TO GND
1	1	THREE-STATE

Figure 3.23: Input Shift Register[8, Pg. 14]

Write operation starts with $\overline{SYNC}$ going low. Data from the SDIN line is clocked into the 16-bit shift register (Fig. 3.23) on the falling edge of SCLK. On the 16th falling clock edge, the last bit is clocked in and the programmed function is executed. $\overline{SYNC}$ line can be kept low or brought high at this point. In either case, it must be brought high for a minimum of 20ns before the next write sequence, so that a falling edge of $\overline{SYNC}$ can initiate the next write sequence[8, Page 13].

If the $\overline{SYNC}$ goes high before the 16th falling edge, this will act as an interrupt to the write sequence. The shift register is then reset and the write sequence is seen as invalid. Neither an update of the DAC register contents nor a change in the operating mode of the DAC occurs.

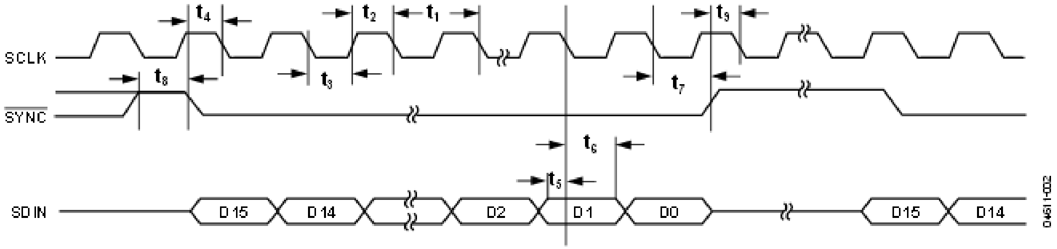


Figure 3.24: Timing Diagram[8, P. 4]

Time Variable Gain

The DAC is used to generate TVG in a way that first, it increases from 0 to its maximum value(2.9V) and then it becomes constant at this value before falling to zero and then repeating again. This happens within a time interval of 100ms in order to maintain the ping rate.

The analog output voltage range of DAC is from 0 to 3.3V which corresponds to the 14-bits ranging from 0(00000000000000) to 16384(11111111111111) values. So, one digital value corresponds to a value of 3.3/16384 = 0.201 mV. As the gain range needs to be set between 0 and 2.9 V, so it is desired to calculate the corresponding value until which the gain will increase. This is done as: (16384/3.3)*2.9 = 14398. Therefore, the digital values have to increase from 0 to 14398 (11100000111110) to produce the necessary 2.9V and thereafter, become constant and then fall down to 0 and repeat.

A new value will be sent after every 16 clock cycles. DAC communicates serially with FPGA using SPI standards. Data is clocked into the input shift register on the falling edge of the serial clock input. A clock frequency of 5MHz is provided to DAC using FPGA's on-board oscillator.

A simple state machine shows the various states of DAC's Operation for generating TVG. Every time, the shift register is increased by a value of '4' and then, the $\overline{SYNC}$ is pulled low for starting the Write operation. This value is added 3600 times to reach the maximum digital value of 14398 and when the Sample_Count crosses 3600, it goes into Hold_State and keeps that value until the time is 100ms.

The VHDL code for write operation to AD5641 lies in DAC.VHD attached in the Attachments(VI) section.

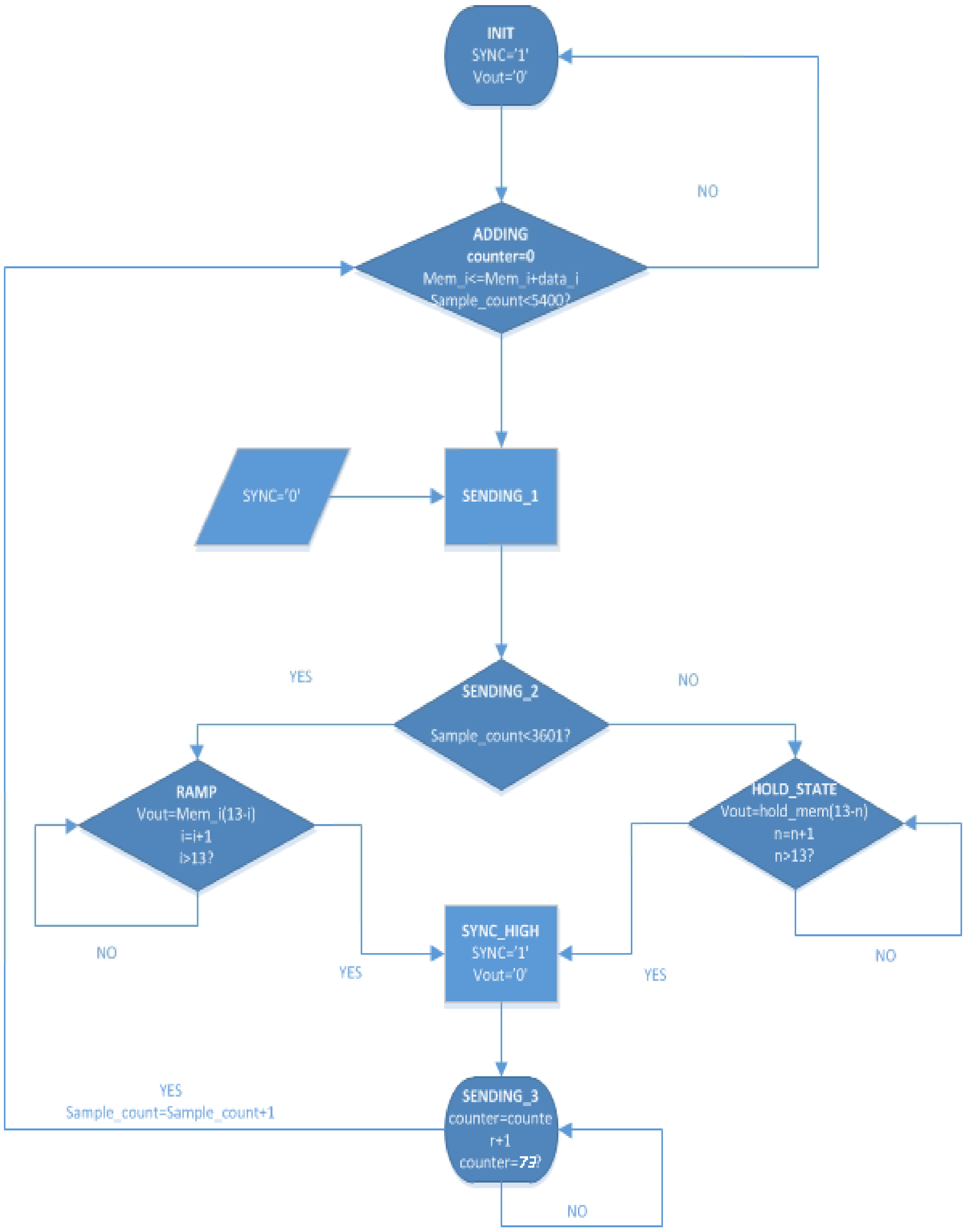


Figure 3.25: State Machine for TVG

3.3.2.4 ADC LTC2245

LTC2245 operates at 3.3V supply from the board and sends 14-bits data in parallel. Apart from these 14 wires, power, ground and control signals like $\overline{OE}$, SHDN and CLK also need to be connected with the FPGA board. Therefore, it was set on a 32-pin QFN adapter and used with a breadboard.

Converter operation

A sampled analog input will result in a digitized value five cycles later. MODE pin is in Offset Binary Format[13, Page 14]. The SHDN pin is always kept at logic '0' for its normal operation. The OF pin is set using a concurrent signal assignment statement and gets High whenever Input is beyond operating range.

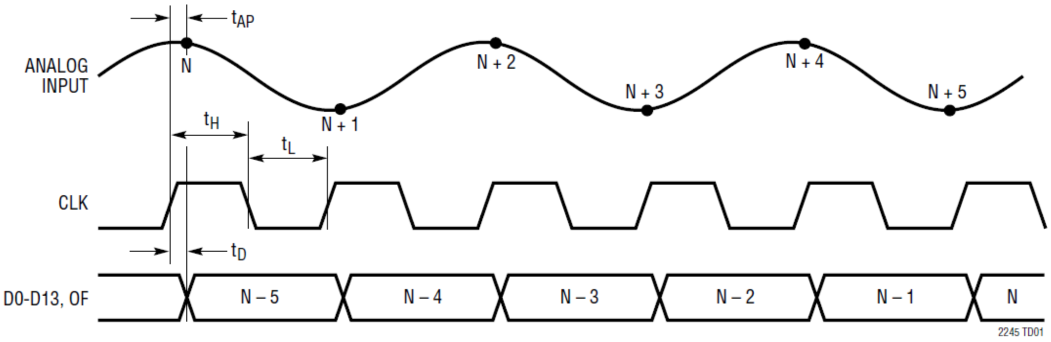


Figure 3.26: Timing Diagram LTC2245[13, page 9]

Envelope Detection

For checking the envelope detection of incoming pulses, ADC is used together with DAC to observe the outputs on the oscilloscope. The process starts with both ADC and DAC in the Idle State by keeping $\overline{SYNC}$ and $\overline{OE}$ pins at High value('1'). For receiving the incoming values, $\overline{OE}$ pin is set to '0'. Then, the position of SW1 is checked. If it is high, then the Read operation is set for 50kHz pulses otherwise, 200kHz pulses will be read.

It will be sufficient to read only the highest value during the positive cycle. Therefore, only the positive half cycle of each pulse is read. The values obtained can either be stored in a buffer or shown continuously using an oscilloscope. It was decided to check peak detection in real time, so the values were continuously sent to DAC. Therefore, the negative half cycle of pulse is used for writing to DAC.

The data from ADC is received via 14 parallel wires using GPIO bus and is stored directly in a 14-bit vector and then, the 13 bits of this vector are compared with a temporary value stored in a 13-bit vector to check the received digital values. If the received value is greater than the already present value in that 13-bit vector, then this new value is stored. Otherwise, the same value will remain in that vector.

After the positive half cycle, the $\overline{OE}$ pin is again set high and the $\overline{SYNC}$ is set low for using the DAC.
The values in the 13 bit vector are required to be transferred to DAC. Therefore, the values are simultaneously being sent to the AD5641 to check the peak detection. AD5641 sends data serially and uses 16 clock cycles for transfer of each value. The first 3 bits sent are '0'. This is because first 2 bits must be '0' for its normal operation and the 3rd bit is intentionally set to '0' as only 13 bits data has to be transferred.

When the data is transferred, the $\overline{SYNC}$ is again set to high value('1') and the related registers are emptied. The process then returns back to Idle state to read the next pulse.

The whole process is synchronized using a counter. For 50kHz, the counter counts up to 100 while for 200kHz, it counts up to 25 when a sampling clock of 5MHz is used. Both ADC and DAC operate on 5MHz.

This state machine is implemented in the file ADC_PARALLEL.vhd available in the Attachments(VI).

Alternate option

There are also other methods available for peak detection in software and the classical method for this purpose is The Hilbert Transform.

This envelope detection method involves creating the analytic signal of the input using the Hilbert transform. An analytic signal is a complex signal, where the real part is the original signal and the imaginary part is the Hilbert Transform of the original signal[25]. Mathematically, the envelope e(t) of a signal x(t) is represented by the magnitude of the analytic signal:-

$$e(t) = \sqrt{x(t)^2 + \hat{x}(t)^2} \qquad [25] \qquad (3.4)$$

where,
$\hat{x}(t)$ is the Hilbert transform of x(t)

This method is not used due to its complex design, cost and overhead involved in the processing time[24].

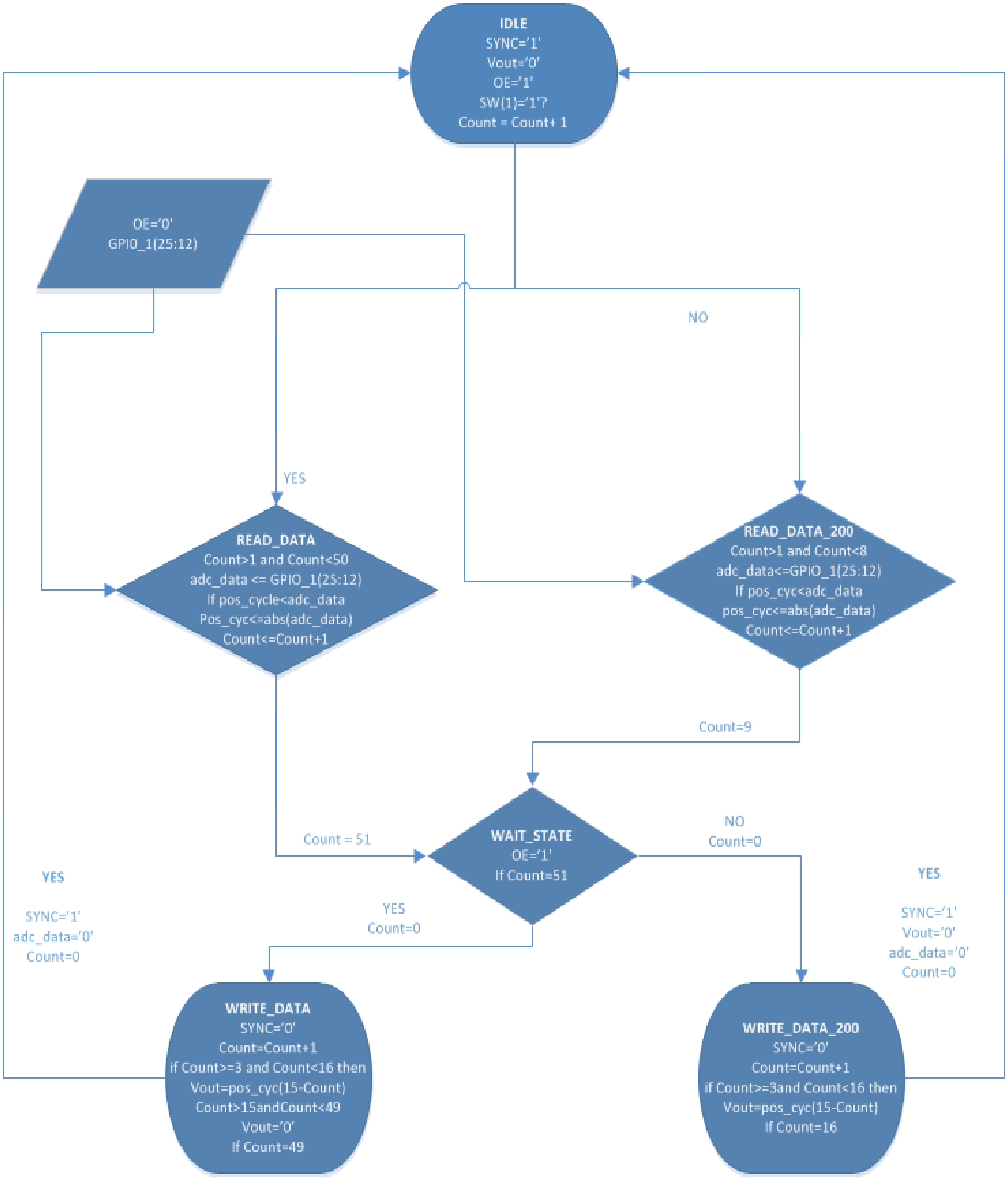


Figure 3.27: State Machine for Envelope Detection

Chapter 4

Testing and Observation

After the Hardware and Software development part, it is really necessary to check the design by testing its various components.

The components are tested individually in the lab before they are actually put in a complete echo sounder system. The various parts include analog switch, amplifier, ADC, DAC and the High Voltage Board.

4.1 Testing of Transmitter Unit

Transmitter part consists of components MD1213 and TC6320 and are tested by putting only these 2 chips on a separate Echo sounder board.

Firstly, it was decided to test the MOSFET driver alone and after that, its outputs were directly connected to the MOSFET pair which is independent of any control signals.

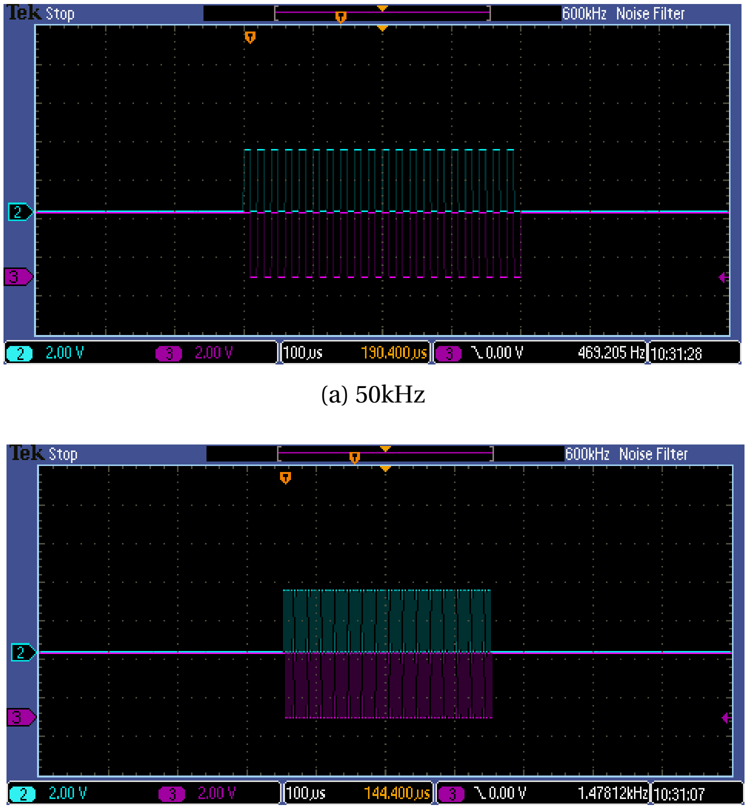
4.1.1 Testing of MD1213

It is necessary to pass the pulses through MD1213 because this MOSFET driver helps in raising the TTL or CMOS logic signals to a level with higher voltage and current ratings. One of the main purpose of doing this is to switch the gate of the dual MOSFETs completely and maximizing the switching speed by providing it more current.

As a result, power MOSFETs spends very less time in the transition state and thus saves energy and does not get hot i.e. With higher switching speeds of the MOSFETs, lesser power will be dissipated.

MD1213 operates from three logic inputs(INA, INB and OE) and has two outputs(OUTA and OUTB). Pins VDD1, VDD2 and VH are given +5V and pins VSS1, VSS2 and VL are connected to ground. Earlier, instead of connecting these later pins directly to the ground plane, three ceramic capacitors were wrongly placed in between each pin and the ground. As a result, the driver was unable to give any outputs. These capacitors were then removed and the pads were shorted using a 0 Ohm resistor.

The pulses from the FPGA going to MD1213 Inputs repeatedly toggle between High and Low states and are shown here:-



(a) 50kHz

(b) 200kHz

Figure 4.1: Inputs-INA and INB to MD1213

In Fig. 4.1, Channel 2 and 3 are respectively the inputs, INA and INB for 50 kHz and 200 kHz respectively. The OE signal is kept high during this time interval for enabling the outputs.

Output level of the pulses change between 0V(Low State) and 5V(High State). The corresponding outputs OUTA and OUTB for 50kHz are obtained at channel 2 and 3 and are shown here:-

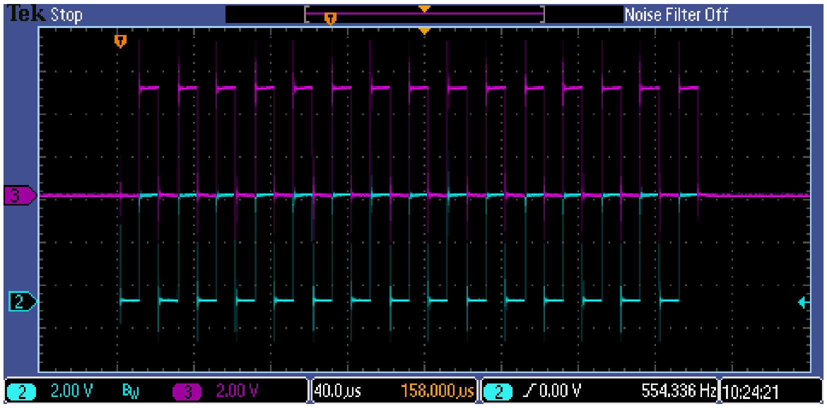


Figure 4.2: Outputs- OUTA and OUTB from MD1213

As clearly seen from the waveforms that inputs and outputs follow the Truth Table(described in Fig.3.21) i.e. Outputs are high when the inputs are low and vice-versa.

The duration of pulses is set according to transducer's bandwidth and a fixed number of cycles are chosen so as to make a particular pulse duration. E.g. 20 cycles of 50kHz will have a pulse duration of .4ms which will make a pulse length($c\tau$) of 0.6m in the water. Similarly for 200kHz frequency, the number of cycles has been changed to 80 so as to have the same pulse duration and length.

Both the operational frequencies i.e. 50 and 200 kHz can be achieved by just toggling the Switch(SW0) present on the DE1-SoC board, with SW = '1' for 50kHz and SW= '0' for 200 kHz as described in the code.

4.1.2 Testing of MD1213 + TC6320

The output burst signals from MD1213 are used as inputs at the gate terminals of the complementary MOSFET pair. The source pins of TC6320 are connected with high positive and negative voltages. The two Drain pins of each MOSFET may or may not be connected to each other. Because it is the highest voltage that is desired as the output and this circuit is not a part of multi-level pulser, these pins can be connected together and this shorting even helps in speed advantages[33, Pg 2].

4.1.2.1 Setup for Transmission of pulses

FPGA provides logic signals for generation of pulses, MOSFET driver translates the voltage and current levels to a level more compatible with dual MOSFETs inputs. MOSFET pair then changes the incoming voltage level to a much higher level depending upon the negative and positive voltages at Pin 1 and 3(or Source pins of N-channel and P-channel respectively).

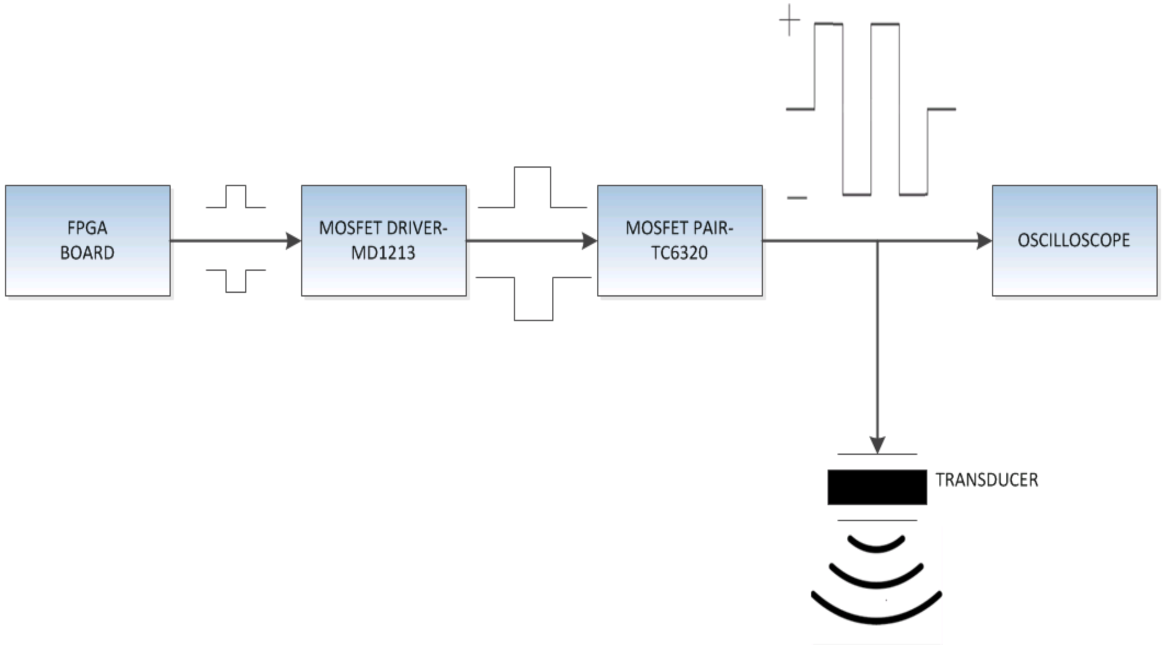


Figure 4.3: Transmitter setup for pulses

For testing purpose, a relatively high voltage of $\pm 20V$ was provided from the power supply to the source pins of N and P channel of TC6320. The output coming from Drain pins was checked by passing it through an Oscilloscope or alternatively transducer could also be connected.

4.1.2.2 Observation

When the transducer is connected, the power transmitted by it varies at different pulse frequencies because of the change in load due to change in impedance with frequencies.

Simply, the amplitude of the transmitted pulse is checked by using oscilloscope. The first attempt was unsuccessful because of the reverse placement of the protection diodes. These diodes were present at the point of incoming high positive and negative voltages from the external power supply. This error was removed and desired results were obtained with output pulses having $\pm 20V$ amplitude.

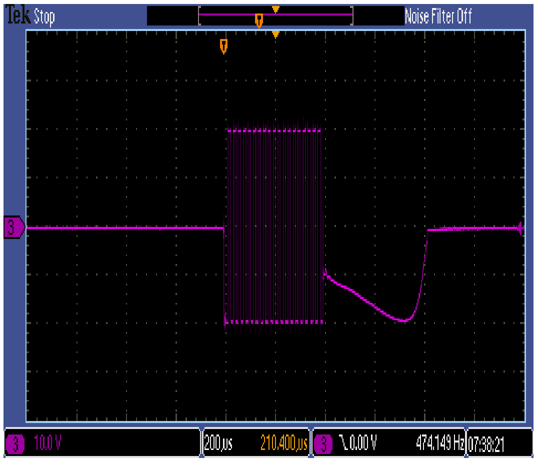


Figure 4.4: 50kHz Pulse

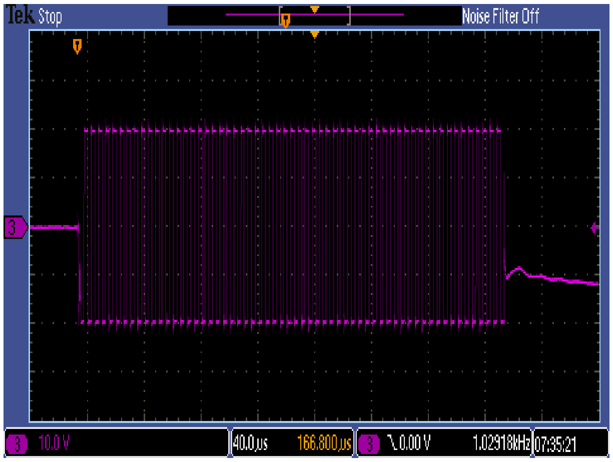


Figure 4.5: 200kHz Pulse

The pulses are eventually passed through the transducer to know the Source Level which could simply be calculated from Eq. 2.5 or by using the following equation, if the TVR of the transducer is known.

$$SL = TVR + 20Log(V_{rms}) \qquad [43, Pg. 2] \tag{4.1}$$

; where TVR is Transmitting Voltage Response of the transducer and is different at different frequencies.

The bump observed at the end of the pulse train is a characteristic of the component TC6320 because both the transistors are shut down at that moment and therefore, voltage takes time to settle down to zero voltage.

4.2 Test using a Transmitter and a hydrophone

Now, the transmitter part was used to send the pulses and a separate hydrophone was installed for receiving echoes. These 2 transducers were placed in a water chamber. The output of the hydrophone was directly connected to the oscilloscope to observe the change in the signal.

For transmission of pulses, same above setup is used. The whole setup with water chamber is shown here:

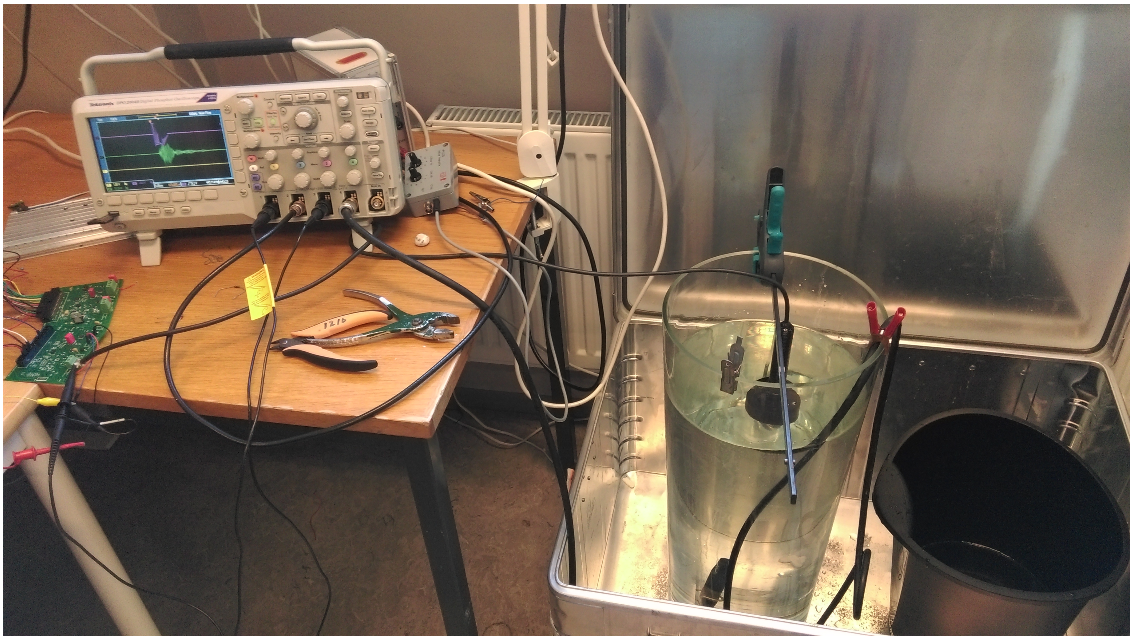


Figure 4.6: Setup for transmission and reception

The transmitting transducer and the hydrophone were placed a little more than a foot apart, and were well beyond the Fresnel Distance i.e. free from the the near field interference. Because the distance was not that much between these two, the received signal was observed exactly after the transmitted pulse somewhat overlapping each other with a slight delay. The received signal was a pulse extended in time due to interactions between the direct path and multi-path refections.

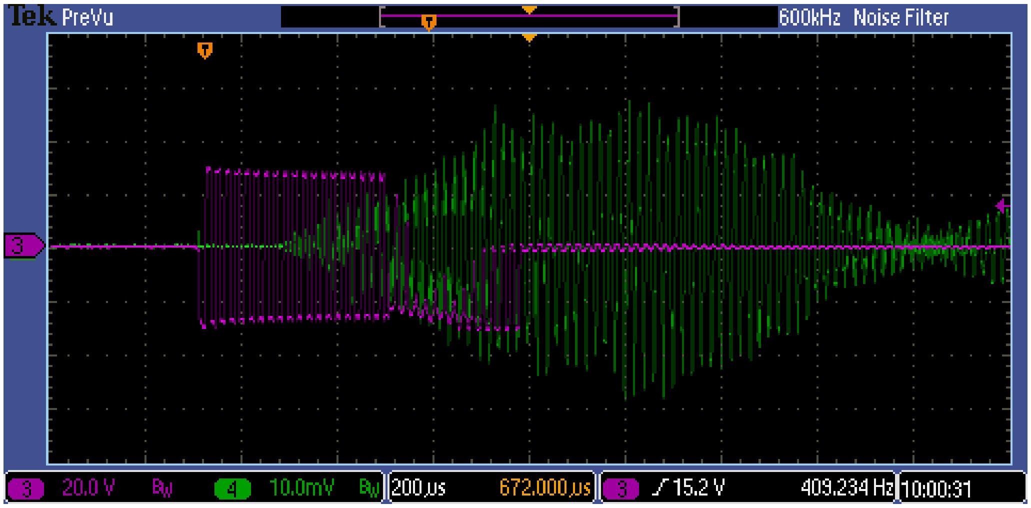


Figure 4.7: Transmitted(CH3) and received(CH4) pulses

Reduction in the amplitude of the received echo can be clearly seen at CH4 and is in mV range.

4.3 Testing of Ultrasound Switch

MD0100 switch comes in SOT-89 package and can easily be installed and tested in a lab by providing input at one terminal and observing output at the other. This switch is bi-directional, so any of the two terminals can be input.

The setup shown in fig.4.8 is for testing the operation of this T/R switch as recommended in the data sheet[9, page 6].

At terminal A(V_A) of MD0100, first a voltage of $20V_{pp}$ is applied as input. The input is basically a sinusoidal burst of 2 cycles of a particular frequency taken from the function generator. Terminal B(V_B) is the output and is connected to a pair of back to back, high speed switching diodes 1N4148 and a 1K Ohm resistor which are connected to the ground and thereafter, the output is measured with the help of an oscilloscope.

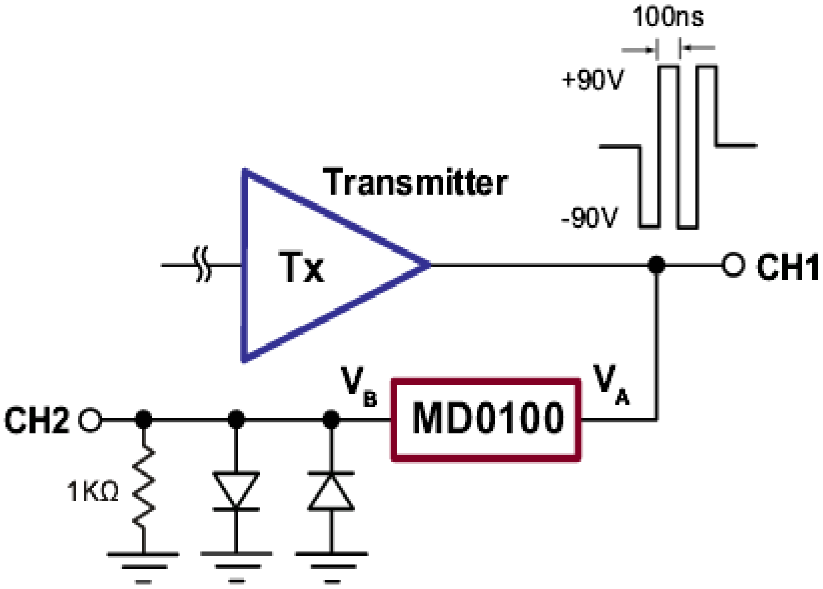


Figure 4.8: Setup for testing of Switch-MD0100[9, page 6]

The setup was tested and the results were obtained for 50 kHz frequency as shown in Fig.4.9.

Here, Channel 1(V_A) is input and Channel 2(V_B) is output. As seen from the waveforms, as soon as the input increases above $\pm 2V$, the switch starts or changes to open state and therefore, does not allow high voltage pulses to pass through it. As a result, the output voltage at terminal V_B is dropped down and calculated out to be equal to $\pm .7V$ due to the presence of back-to-back diodes.

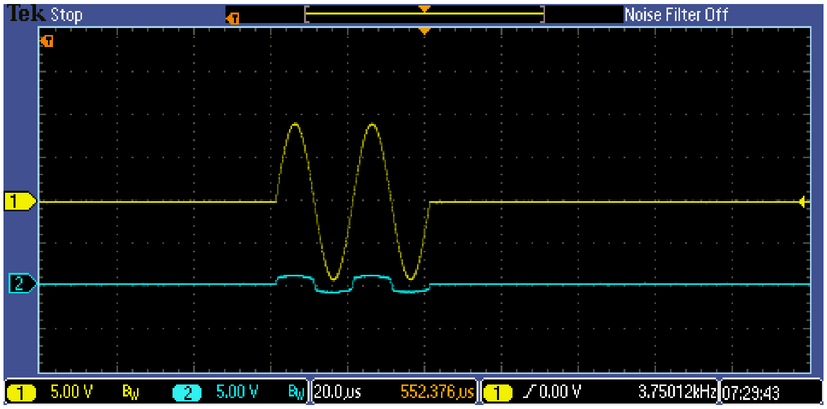


Figure 4.9: Waveforms of V_A and V_B from setup fig.4.8

4.3.1 Transmitted Voltage seen from Receiver Side

The high voltage pulses coming from the Transmitter side are harmful for the receiver circuit. The voltages entering the receiver side were observed when the pulses were being sent. These high voltages turn the switch from its Closed state to Open state for the same time duration, τ as that of the pulses.

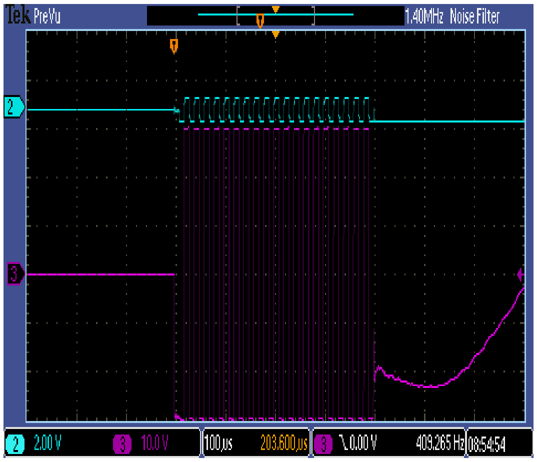


Figure 4.10: 50kHz Pulse

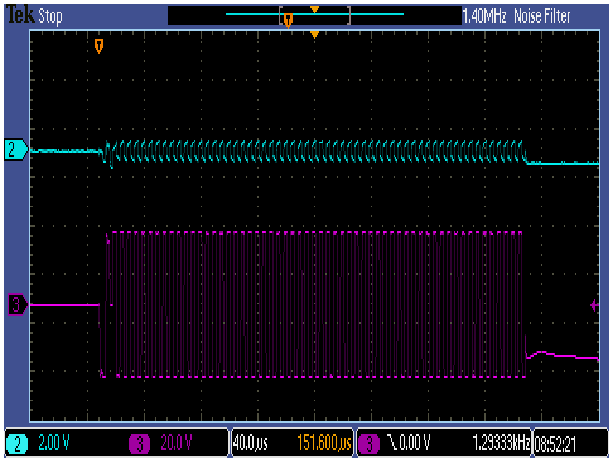


Figure 4.11: 200kHz Pulse

Figures 4.10 and 4.11 show the transmitted pulses of voltage ±30V at Channel 3 and the same pulses obtained after passing the switch at Channel 2.

Thus, the voltage entering the receiver side is of very low amplitude and cannot destroy the installed Electronics. Also, the voltages which will be received from the transducer will be weak and will pass this switch without being attenuated.

4.4 TVG using DAC

DAC is used in this project to provide the variable gain to the amplifier, but at some points it is also used to provide the analog outputs in collaboration with the FPGA board and ADC.

Gain Voltage(VGN) to the amplifier can be varied over the range between 0.1 and 2.9V, where increase in voltage increases the gain and it can be provided commonly to both the gain pins(VGN1 and VGN2) of the cascaded channels. The write operation to DAC is based on the state machine described in the section 3.3.2.3
VGN increases as a ramp voltage until 2.9V and thereafter, stays constant at this voltage before falling to 0 and repeating the cycle again.

The gain voltage's time interval is set according to the pulse repetition frequency(PRF) of the pulses in the transmitter and needs to be 100ms as the PRF is 10Hz.

The corresponding waveform for VGN is shown in Fig. 4.12

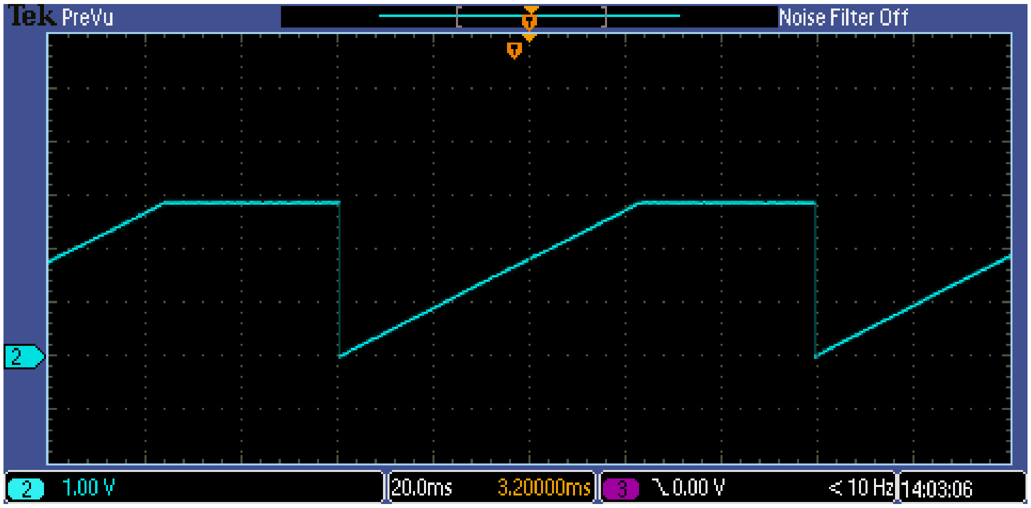


Figure 4.12: TVG from DAC

4.5 Testing of Amplifier using Variable Gain

The variable gain amplifier, AD605 takes analog voltage from DAC as VGN voltage for providing TVG to the received pulses.

This gain depends on the connection of FBK pins to the output pin and even negative gains can be achieved by shorting these two pins.

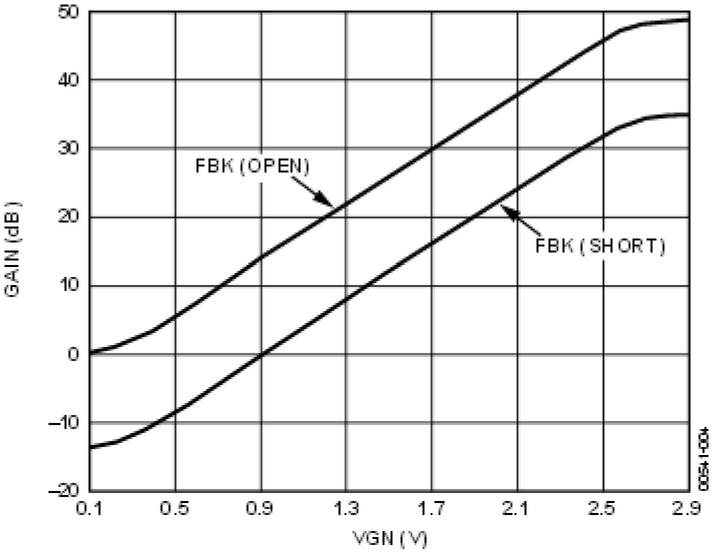


Figure 4.13: Gain range for various V_{gn} voltage with FBK pin open and short[7, Pg. 8]

The above diagram shows Gain(in dB) of a single channel and when two channels of this amplifier are cascaded, a total gain range of 0 to 96dB can be achieved by keeping the FBK pin unconnected.

4.5.1 Setup for testing of Amplifier

For testing the amplifier, it should be given a low voltage input. So, it is provided with a single-ended sinusoidal input from the function generator, TG550 with an amplitude of $\pm 2.5 mV$. The amplifier is connected to an external power supply which gives a supply of 5V to VPOS pins and 2.5V to VREF pin.

The gain(TVG) is provided. The amplifier is tested with both the channels connected in series(with FBK pins open). The amplified output is also single-ended and goes to the oscilloscope. Function Generator is also connected directly to the oscilloscope so as to compare both the input and output.

Fig. 4.14 shows the setup.

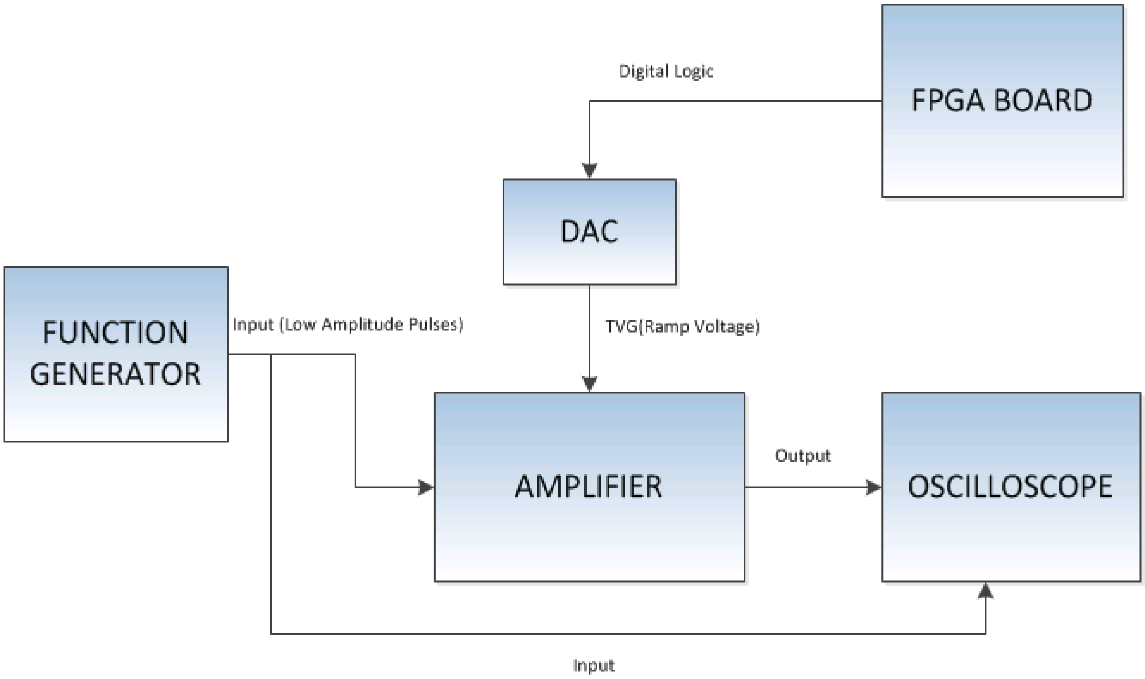


Figure 4.14: Setup for testing of Amplifier using TVG

With the given input, the full gain range(0 to 2.9V) will yield a very high output which will be very much out of the desired range. So, TVG was reset to provide a gain voltage which rises only up to half of the original voltage range i.e. around 1.5V and this will give a gain of approx. 52dB. The output, thus obtained is significantly reduced and now, it is in the working range of the later connected ADC.

4.5.2 Observation

The amplifier was tested and it worked quite well with TVG. The input is a continuous wave. Therefore, the amplitude of output wave is increasing with time.

Amplifier AD605 has a limit on the output signal range. If the output signal goes beyond this range, then the amplifier goes into the saturation state.

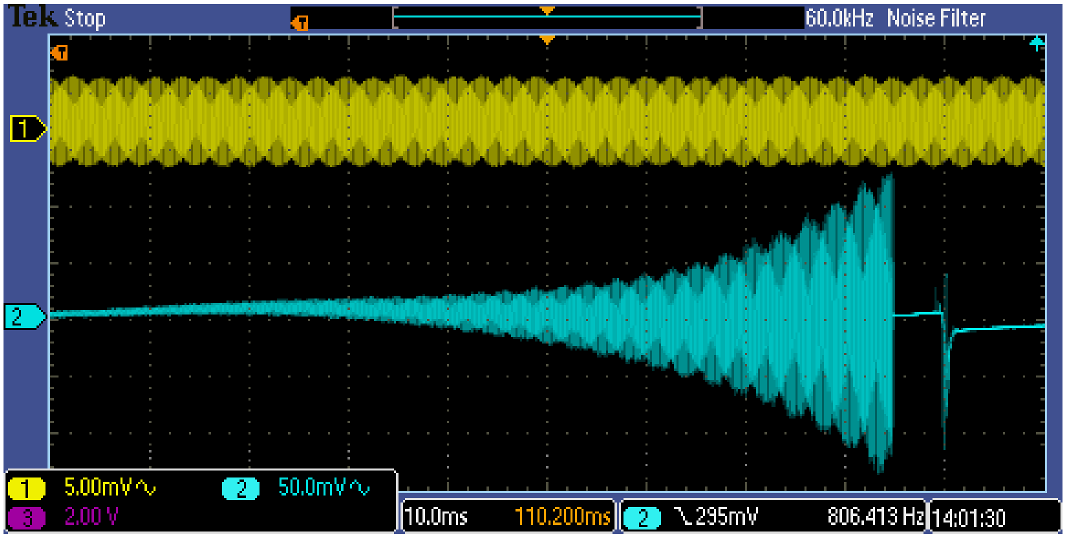


Figure 4.15: Input and TVG amplified Output

Here, Channel 1 is the input of 5mV and Channel 2 is the output showing the voltage increasing with time(or Gain).

The VHDL Code for this section lies in DAC_TVG.VHD in the DVD and Attachments(VI) Section.

4.6 Envelope Detection using ADC and DAC

Envelope Detection is tested using the state machine described in Section 3.3.2.4

4.6.1 Setup

ADC receives its inputs from Signal Generator and sends the outputs to FPGA board. DAC was also connected to the FPGA Board so as to check the logic implemented. The main purpose here is to check the peak detection of the incoming sinusoidal signals.

The reading and writing cycles are different for 50 and 200kHz frequencies. The complete setup is shown here:

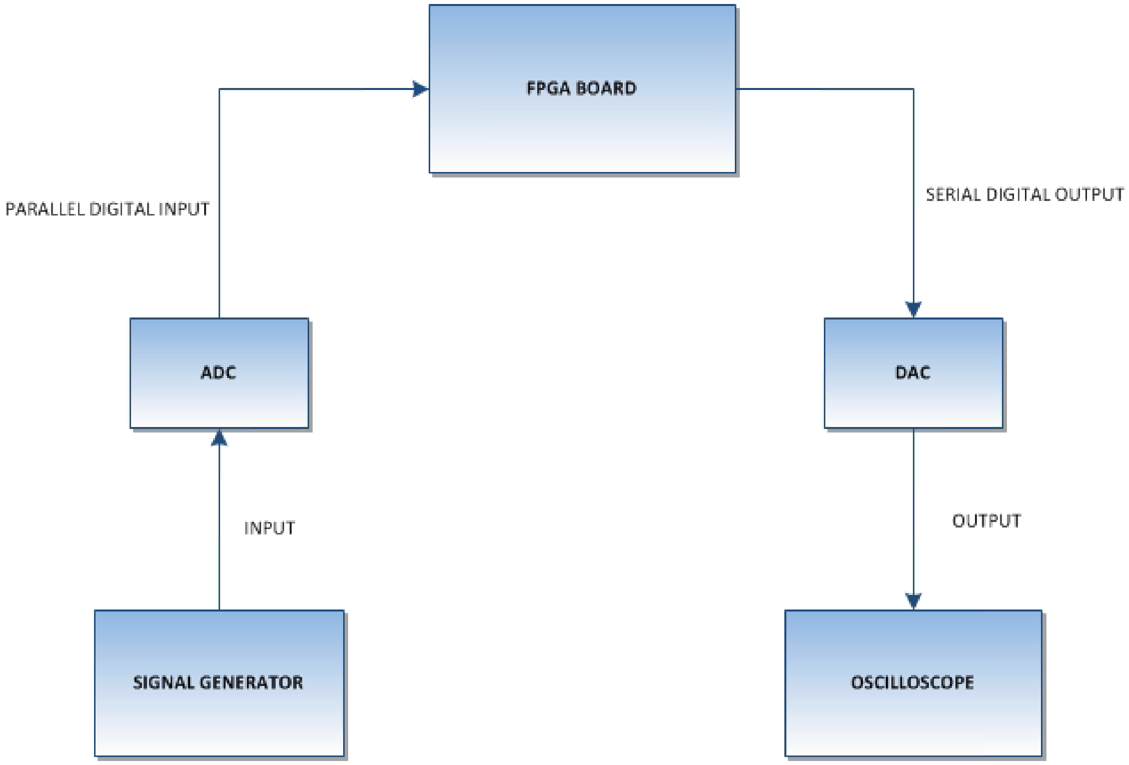


Figure 4.16: Setup for testing of ADC

A single-ended input of 0.1-0.2V was provided as input to the ADC. Both ADC and DAC are operated on 3.3V supply from FPGA.

4.6.2 Observation

The setup was tested for 50 kHz frequency. But 200 kHz pulse can also be tested. This was done in the same way as in Transmitter part where Switch(SW1) states '1' and '0' change the operations for obtaining envelopes of 50 and 200 kHz frequencies respectively.

First of all, a continuous Sine wave of 50kHz from generator was sent as input to this system and the output was observed to be touching the tops of the every cycle of the input wave. Thus, covering all the peak points of the wave.

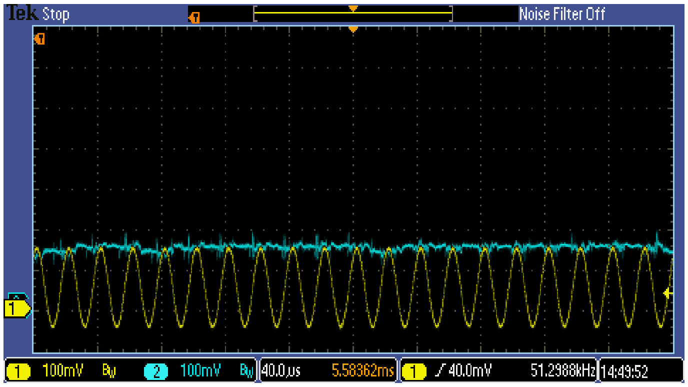
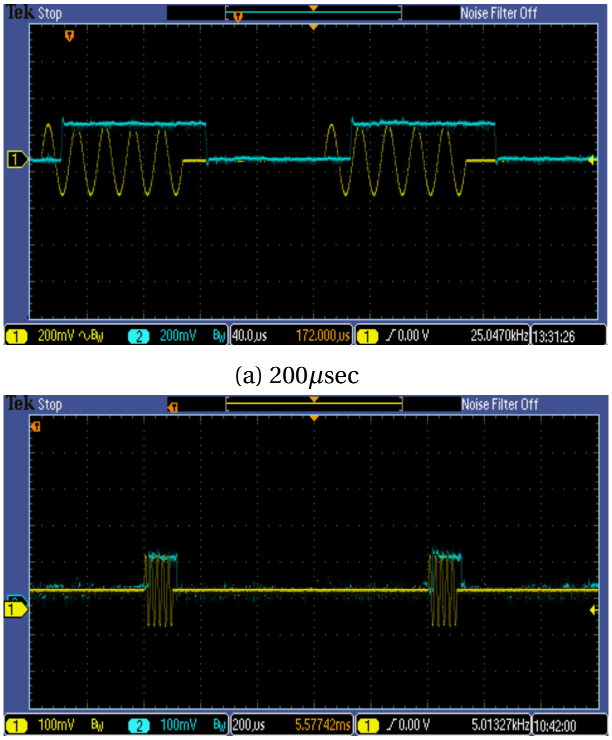


Figure 4.17: Input(1) and Output(2)- Peak Detection

And then, the inputs were switched to Simple bursts with 5 cycles and defined rates. As expected, the output was received but slightly delayed in time. The output was simply an envelope of the input covering the peaks of the 5 pulses.



(a) 200μsec

(b) 1ms

Figure 4.18: Envelope Detection of bursts with different ping rates

The delay in the output waveform at Channel 2 is due to the reason that Writing operation to the DAC starts in the negative half cycle after reading the inputs during the positive half.

As discussed earlier, the echo reflected from the sea targets is quite different from the transmitted pulses and is known to be amplitude modulated(AM) and contains useful information about the target. Therefore, this time the pulses coming from the generator were amplitude modulated and the response of the system was then observed for such signals. These amplitude modulated pulses used a carrier frequency in the range of 1-10 kilohertz for modulation.

In the first attempt, the envelope of the AM pulses were obtained, but the output was missing in every fourth pulse. This was due to an error in the code. This was removed and the outputs were obtained.

As the peaks of the AM pulses are now at different amplitudes, correspondingly the peak detection is observed to be occurring in steps, first increasing until the maximum amplitude and then decreasing.

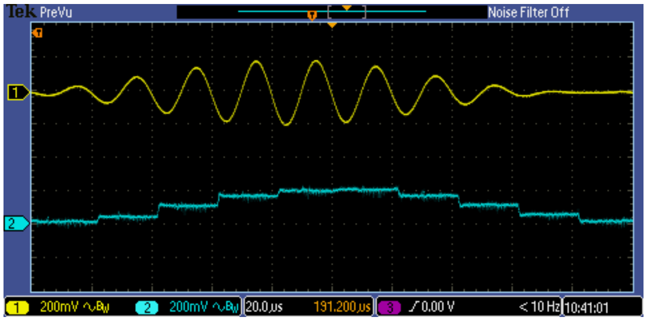


Figure 4.19: Envelope Detection of single AM input

Below are the AM inputs with a repetition rate of 200μsec.

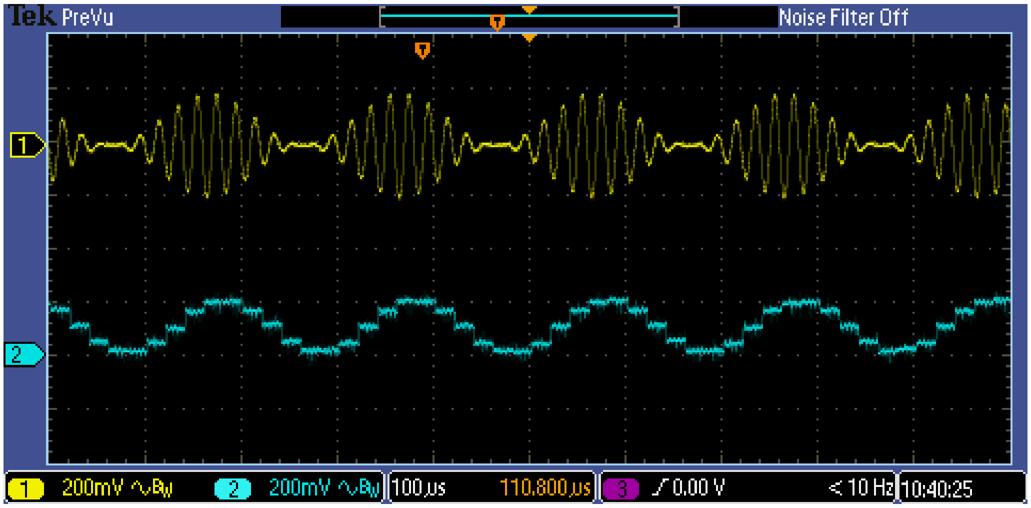


Figure 4.20: Continuous Bursts

In the finishing phase of this project, the whole Echo sounder + FPGA board connected to the PC was used to store the acquired data. This system was already developed by Dr. Ketil and tested for obtaining the envelope of the AM pulses which were fed in to the input of ADC. Here, instead of sending data back to DAC, it is stored in a file which can be plotted afterwards.

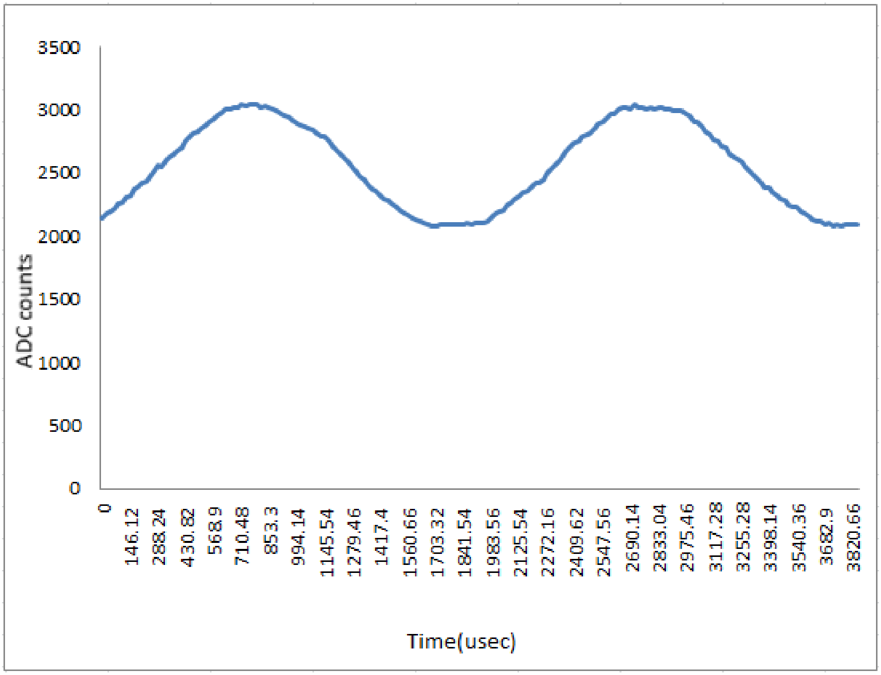


Figure 4.21: Envelope of AM bursts

Although, this is a complex solution for storing data, other solutions like storing data to a USB or communicating with UART to the computer are also possible.

4.7 HV MAX1771 Board

The produced High Voltage Board was ready for testing. So, a Power supply delivering 12V was used as source for both the pins of the connector CN1, while the pins of CN2 were connected to the ground. The output was obtained at CN3 and it was observed to vary between 181.1V and 194.3V by changing the turns of the potentiometer, VR1. Thus, it was a successful attempt for making a high voltage power supply from a source of 12V.

This board can now be directly connected to the transmitter electronics to power the MOSFET pair. The N and P-channel MOSFETs used, have +200V and -200V as their Drain-to-Source Breakdown Voltage(BV_{DSS}) respectively.

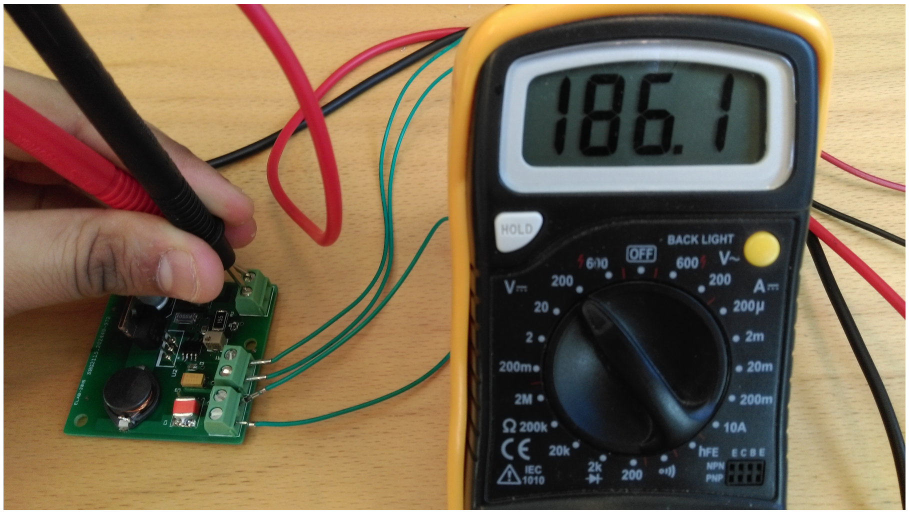


Figure 4.22: Output Voltage from MAX1771 Board

Chapter 5

Conclusion

This project demonstrated 'the development of a wide band front end echo sounder receiver circuit' using a FPGA based system. It was important to move various processes like rectification and peak detection which were previously performed in the Analog Electronics into the FPGA systems so as to minimize losses and improve the overall efficiency.

In Transmitter electronics, MOSFETs were used for sending pulses. This solution worked quite well with different frequencies giving perfect pulse outputs. Thus, designing an efficient transmitter circuit with lesser number of transistors than the previous designs.

The transformer method earlier used in the transmitter electronics was replaced by SMPS technique. It is a stand-alone unit on a separate PCB board, highly efficient and free from losses unlike transformer. In this method, the noise levels were minimized by keeping the board size small and providing proper ground plane at the bottom layer.

The various components used in this system are fully compatible with the TTL or CMOS logic levels as provided by the FPGA board.

The impedance response of the transducer FURUNO 520-5PSD was checked for measuring the resistance values at its two resonant frequencies, 50kHz and 200kHz.

Analog Front End(AFE) in the receiver part was developed to be used for wide band purposes. So, using a simple switch was an appropriate solution for allowing all the neighboring frequencies around the resonant frequency to easily pass the front end and use of any resonance filter was thus avoided.
It was demonstrated that in the front end receiver, a variable gain amplifier with its channels connected in series can be used for providing the TVG to the incoming pulses with the help of a DAC. Thus, improving the Signal to Noise Ratio(SNR) and dynamic range. This amplifier, itself offered a very low noise at various gains. Therefore, preventing any additional electronics noise being added to the signal. The variable gain is implemented in the form of ramp wave and is easily controllable, requiring only few modifications when operating at different sea depths.

The diode protection and filter circuits at the ADC input showed good results and were verified both by simulations and experiments. ADC together with the FPGA board have been successfully tested for the Envelope Detection process. This operation implemented a simple algorithm for extracting the top values in each cycle of the pulse. Tests showed that a very stable and accurate peak detection of pulses with different shapes and amplitudes was possible.

Pulses of two different frequencies can be generated and processed by the receiver by just toggling the position of Switches on the FPGA board. Thus, the user has a large control over the signal processing which is basically important for the echo sounder systems developed for research purposes.

Hence the developed Echo sounder can be used as a wide band system which can provide the outputs without influencing echo's shape.

Chapter 6

Further Development

The development procedure of the echo sounder system has followed the desired plan to quite a large extent. The simulations and testing of various circuits were performed individually before making a final system. This project can be used as a reference for further development in the coming future.

I have tried to design, develop and cover every aspect of the system within the given time-frame, but there are still some challenges which need to be overcome:

- Replace the ordinary potentiometer in the High Voltage Board with a digital potentiometer with more number of turns so that the voltage supplied to the MOSFETs can be varied over a large range. This will help in increasing the dynamic span of the power being transmitted through a pulse.
- Edit the VHDL code for TVG and make it flexible such that it can provide different Gains for different water depths.
- Develop a reliable and fast connection between the FPGA board and the PC for storing of data.
- Connect the whole system and test it in the sea.

For improvement of the system, a preamplifier providing a fixed gain can be used in front of the existing amplifier. This preamplifier unit should have a high input impedance.

The passive RC network used as anti-aliasing filters can be replaced by a programmable Low-pass filter like LTC6602 or the earlier mentioned active filter LTC1563-2/3 for getting sharp cut-off frequencies.

I personally recommend to use components packages which are easy to solder and in which short circuits can be easily noticed and removed and avoid using QFN packages.

Another important point for improving the performance is to synchronize the TVG at the receiver side with the Trigger pulse(OE) at the Transmitter side so that every time a pulse is sent out to the transducer, a signal is sent to the receiver to start the TVG.

Hopefully, new students will also show interest in joining the Hydro acoustics group and these tests performed in the laboratory, can then be helpful for this system's advanced development.

List of Abbreviations

JTAG	Joint Test Action Group
MOSFET	Metal-oxide Semiconductor Field-Effect Transistor
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
TVG	Time Variable Gain
SMPS	Switch Mode Power Supply
UART	Universal Asynchronous Receiver/Transmitter
CAD	Computer-Aided Design
DI	Directivity Index
I2C	Inter Integrated Circuit
VHDL	VHSIC Hardware Description Language
CMOS	Complementary Metal-oxide semiconductor
EDA	Electronic Design Automation
LFGA	Lead Frame Grid Array
TVR	Transmitting Voltage Response
SRT	Receiving Response of Transducer
PDIP	Plastic Dual In-Line Package
bxl	Binary Xlator
USB	Universal Serial Bus
TTL	Transistor-transistor Logic
SOIC	Small Outline Integrated Circuit
QFN	Quad Flat No-leads
SPI	Serial Peripheral Interface
PCB	Printed Circuit Board
GPIO	General Purpose Input/Output

Bibliography

[1] Clive "Max" Maxfield, The Design Warrior's Guide to FPGAs:Devices, Tools and Flows, Newnes

[2] Altera DE1-SoC User Manual, http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=4, Terasic

[3] Mats Randgaard, Ekkoloddsystem realisert med System-on-Chip på FPGA, Fysisk Institutt, UiO

[4] Morten Hellum, Optimization of Front End in Echosounder, Fysisk Institutt, UiO

[5] Mark Zwolinski, Digital System Design with VHDL, Second Edition, Prentice Hall

[6] Robert J. Urick, Principles of Underwater Sound, Third Edition, McGraw Hill Inc., 1983

[7] AD605- Low Noise,Variable Gain Amplifier, http://www.analog.com/media/en/technical-documentation/data-sheets/AD605.pdf, Analog Devices

[8] AD5641-14-Bit nanoDAC, http://www.analog.com/media/en/technical-documentation/data-sheets/AD5641.pdf, Analog Devices

[9] MD0100- High Voltage Protection T/R Switch, http://ww1.microchip.com/downloads/en/DeviceDoc/MD0100.pdf, Supertex Inc.

[10] ADC Input Protection, http://www.ti.com/lit/an/slaa593/slaa593.pdf, Texas Instruments

[11] MD1213- High speed Dual MOSFET Driver, http://ww1.microchip.com/downloads/en/DeviceDoc/md1213.pdf, Supertex Inc.

[12] TC6320- N and P channel E-mode MOSFET Pair, http://ww1.microchip.com/downloads/en/DeviceDoc/tc6320.pdf, Supertex Inc.

[13] LTC2245-Analog to Digital Converter, http://cds.linear.com/docs/en/data-sheet/2245fa.pdf,Linear Technology

[14] HV7361- Ultrasound RTZ Pulser with T/R Switch, http://ww1.microchip.com/downloads/en/DeviceDoc/HV7361.pdf, Supertex Inc.

[15] Alexander Sveinall, Embedded Sonar med GPRS, Fysisk Institutt, UiO

[16] Gharpurey, R., and Kinget, P. (2008). Ultra wideband: Circuits, transceivers and systems. New York: Springer Science+Business Media

[17] K.A. Johannsen and R.B. Mitson, Fisheries Acoustics- A practical manual for Aquatic Biomass Estimation, FAO1983

[18] Sergio Rui Silva, Advances in Sonar Technology, I-Tech, Vienna

[19] LM317-1.2 to 37V adjustable voltage regulators, http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000455.pdf, STMicroelectronics

[20] MAX1771- Step-Up DC-DC Controller, http://pdfserv.maxim-integrated.com/en/ds/MAX1771.pdf, Maxim Integrated

[21] Knut Skumvoll, Utvikling av Sonarsystem, Fysisk Institutt, UiO

[22] Nixie HV Switching PSU, http://desmith.net/NMdS/Electronics/NixiePSU.html

[23] Helge Balk, FYS3220-Lineær Kretselektronikk

[24] Signal Processing Overview of Ultrasound Systems for Medical Imaging, http://www.ti.com/lit/wp/sprab12/sprab12.pdf

[25] Mathworks-Envelope Detection, http://se.mathworks.com/help/dsp/examples/envelope-detection.html

[26] Resonance in series RLC circuit, http://www.electrical4u.com/resonance-in-series-rlc-circuit/

[27] AD7928- Analog to Digital Converter, http://www.analog.com/media/en/technical-documentation/data-sheets/AD7908_7918_7928.pdf, Analog Devices

[28] Johannes du Preez, An investigation into OFDM as a suitable modulation technique for an acoustuc underwater Modem, Dept. of Elect. and Electronic Engg., Stellenbosch University

[29] Single Beam Acoustic Depth Measurement Techniques, http://www.syqwestinc.com/media/support/depthmeasuringtechniques.pdf, SyQwest

[30] LTC1563-2/LTC1563-3- Active RC, 4th Order Low-pass Filter Family http://cds.linear.com/docs/en/datasheet/156323fa.pdf, Linear Technology

[31] Love, R.H., Dorsal Aspect Target Strength of Individual Fish, J. Acoust. Soc. Am., 49:816(1971)

[32] Love, R.H., Maximum Side-Aspect Target Strength of Individual Fish, J. Acoust. Soc. Am., 46:746(1969)

[33] AN-H53- High Voltage Pulser Circuits, http://ww1.microchip.com/downloads/en/AppNotes/AN-H53.pdf, Supertex Inc.

[34] MT-091 : Digital Potentiometers, http://www.analog.com/media/en/training-seminars/tutorials/MT-091.pdf, Analog Devices

[35] DACs vs. Digital Potentiometers: Which is right for my application?, http://pdfserv.maximintegrated.com/en/an/AN4025.pdf, Maxim Integrated

[36] Helge Balk, Digitalisering av Sonar, 1996, Fysisk Institutt, UiO

[37] Florian Xaver, Envelope Detector based on Hilbert Transform, der Technischen Univ. Wien

[38] J. Marszal, Z. Ostrowski, J. Shmidt, L. Kilian, A. Jedel, A. Schmidt, Time Variable Gain for long range SONAR with chirp sounding signal, Gdansk University of Technology, Gdansk

[39] Waldemar Lis, Jan Schmidt, Zbigniew Wojan, Determining the impedance of Ultrasonic transducers by the Added Elements Method, Gdansk University of Technology, Gdansk

[40] Alan A. Winder, Sonar System Technology, IEEE

[41] Transmission Loss, http://www.sal2000.com/ds/ds3/Acoustics/TL.htm

[42] AD8338- Low power, 18MHz Variable Gain Amplifier, http://www.analog.com/media/en/technical-documentation/data-sheets/AD8338.pdf, Analog Devices

[43] Source Level from Electric Power by Balk 2014,

[44] MD1210-High Speed Dual MOSFET Driver, http://ww1.microchip.com/downloads/en/DeviceDoc/md1210.pdf, Supertex Inc.

[45] TC2320-N- and P-Channel Enhancement-Mode Dual MOSFET, http://ww1.microchip.com/downloads/en/DeviceDoc/TC2320.pdf, Supertex Inc.

[46] MD1211-High Speed Dual MOSFET Driver, http://ww1.microchip.com/downloads/en/DeviceDoc/md1211.pdf, Supertex Inc.

[47] MD0101- High Voltage Protection T/R Switch with Clamp Diodes, http://ww1.microchip.com/downloads/en/DeviceDoc/MD0101.pdf, Supertex Inc.

[48] MD0105- Four-Channel High Voltage Protection T/R Switch, http://ww1.microchip.com/downloads/en/DeviceDoc/MD0105.pdf, Supertex Inc.

[49] AD5040-14-Bit nanoDAC, http://www.analog.com/media/en/technical-documentation/data-sheets/AD5040_5060.pdf, Analog Devices

[50] Basic Calculation of a Boost Converter's Power Stage, http://www.ti.com/lit/an/slva372c/slva372c.pdf, Texas Instruments

[51] Understanding Boost Power Stages in Switchmode Power Supplies, http://www.ti.com/lit/an/slva061/slva061.pdf, Texas Instruments

[52] Positive Adjustable Regulator, http://cds.linear.com/docs/en/datasheet/lt0117.pdf, Linear Technology

[53] Mini Tutorial, http://www.analog.com/media/en/training-seminars/tutorials/MT-211.pdf, Analog Devices

Attachments

I Circuit Schematics Echo Sounder

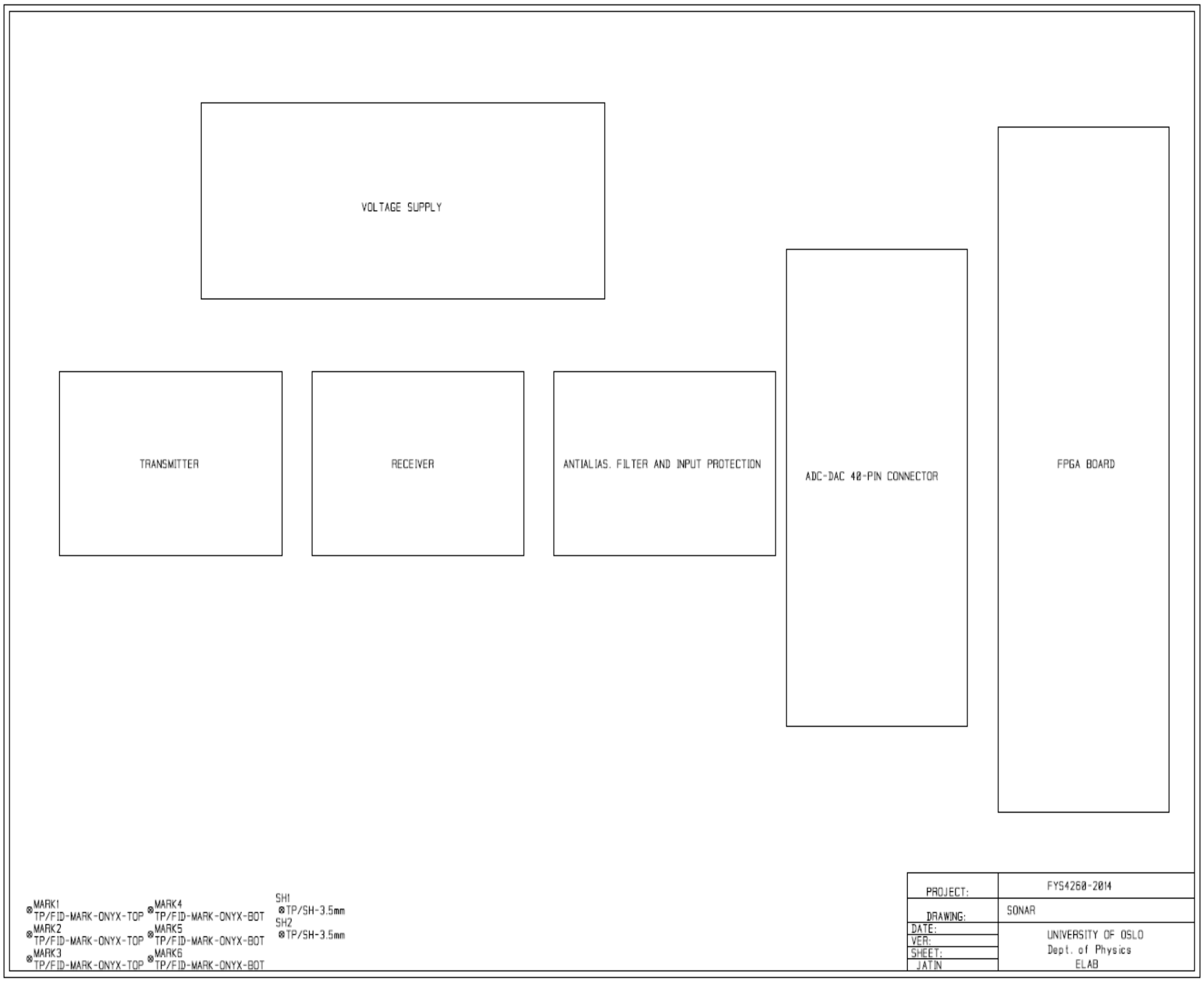


Figure 6.1: Top Schematics- Echo sounder

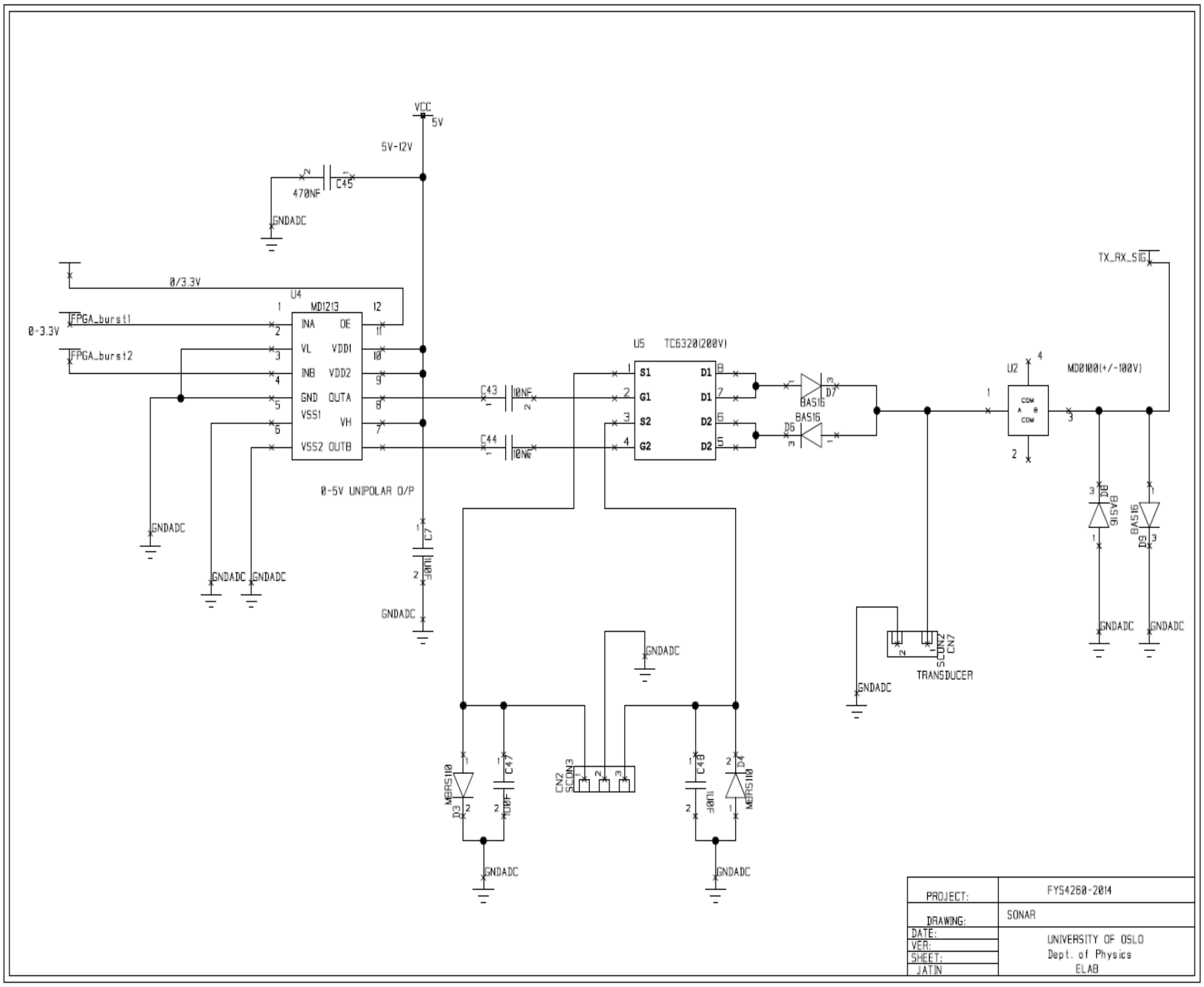


Figure 6.2: Transmitter and Switch Schematics

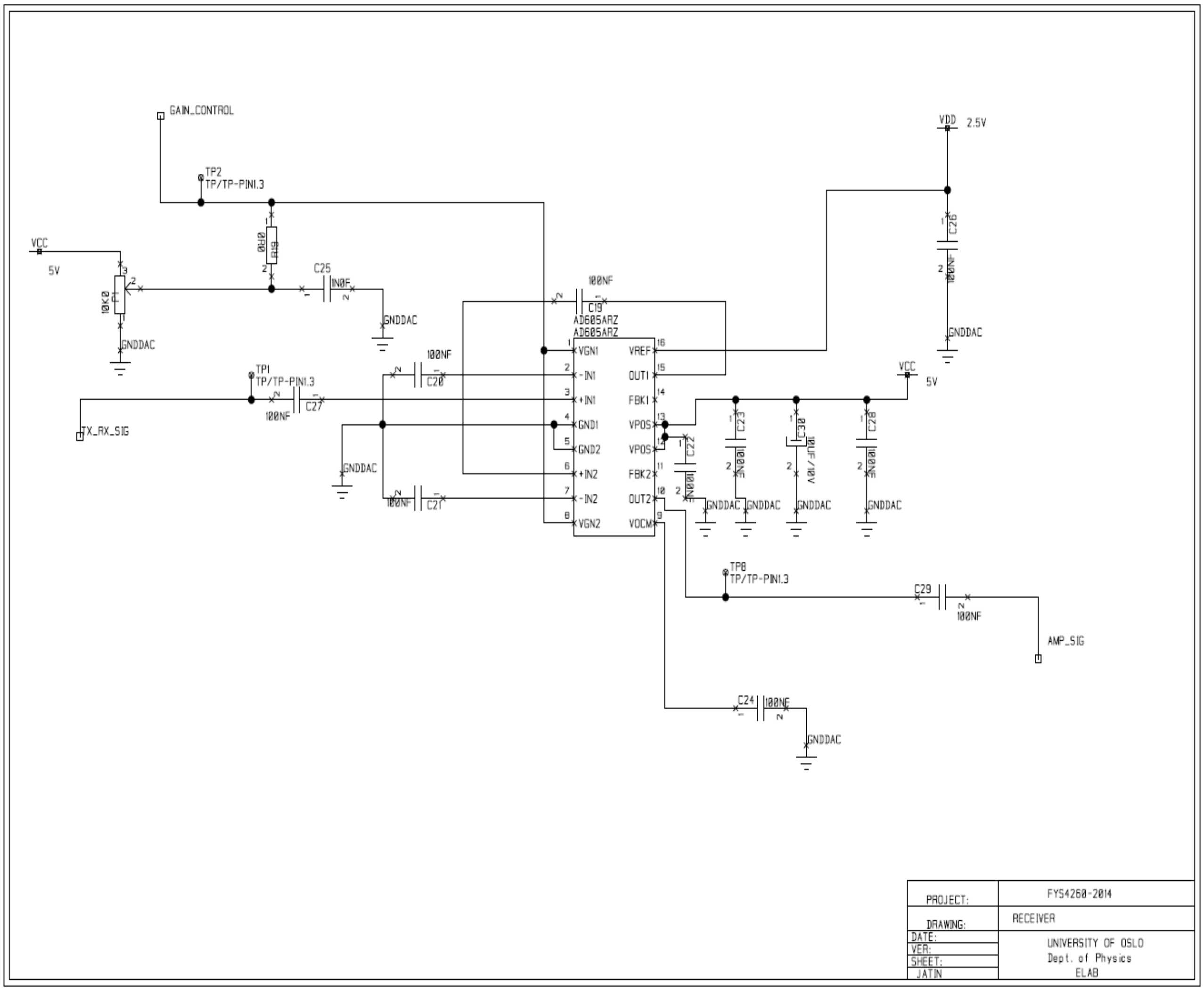


Figure 6.3: Receiver Schematics

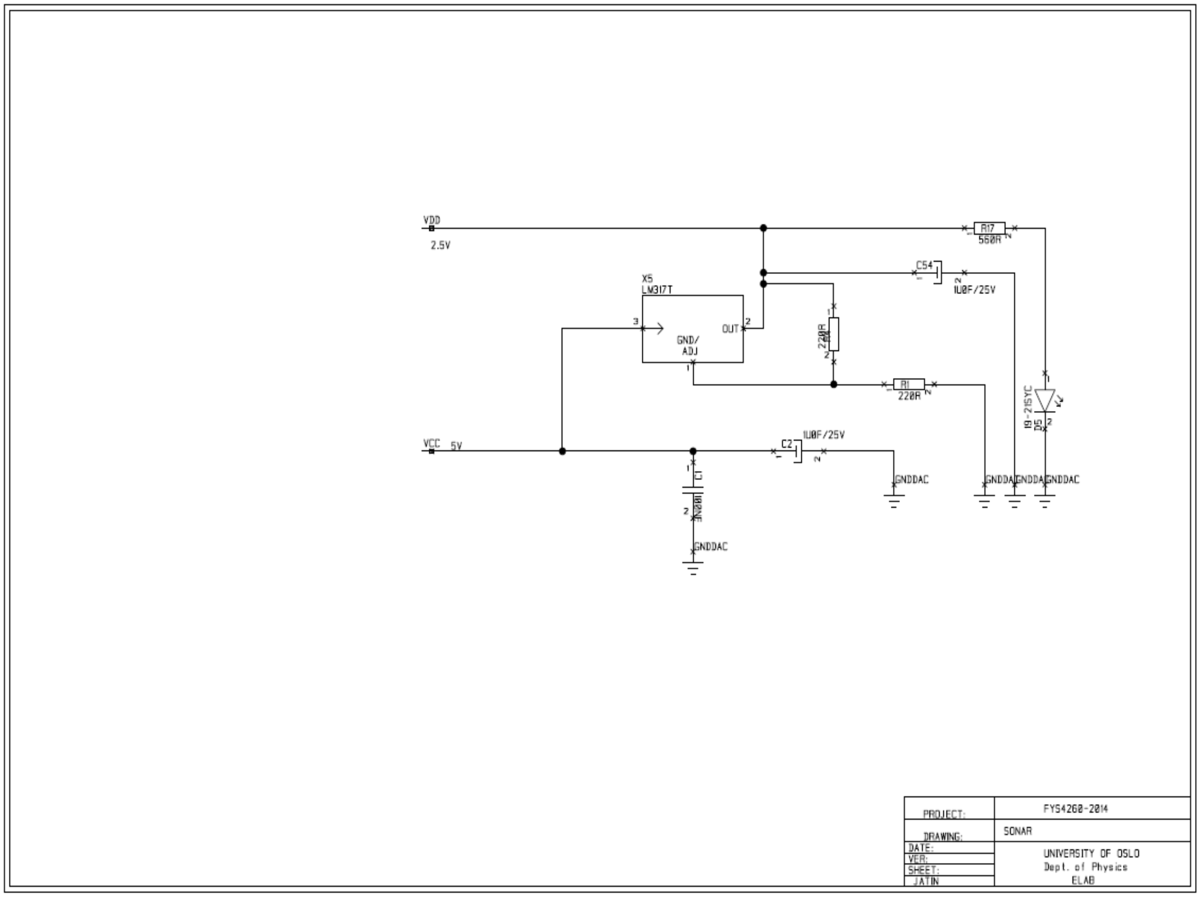


Figure 6.4: VREF Supply Schematics

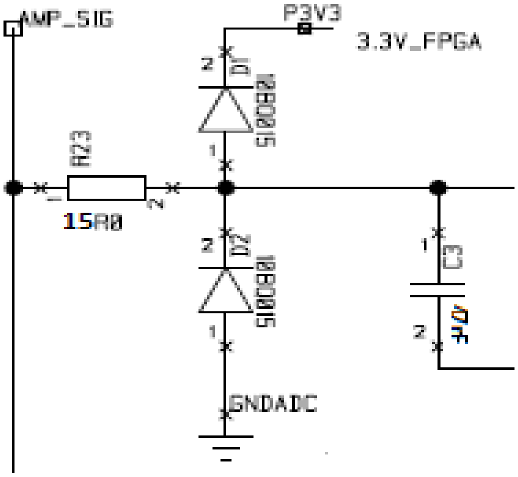


Figure 6.5: Anti-alias and ADC Input Protection Schematics

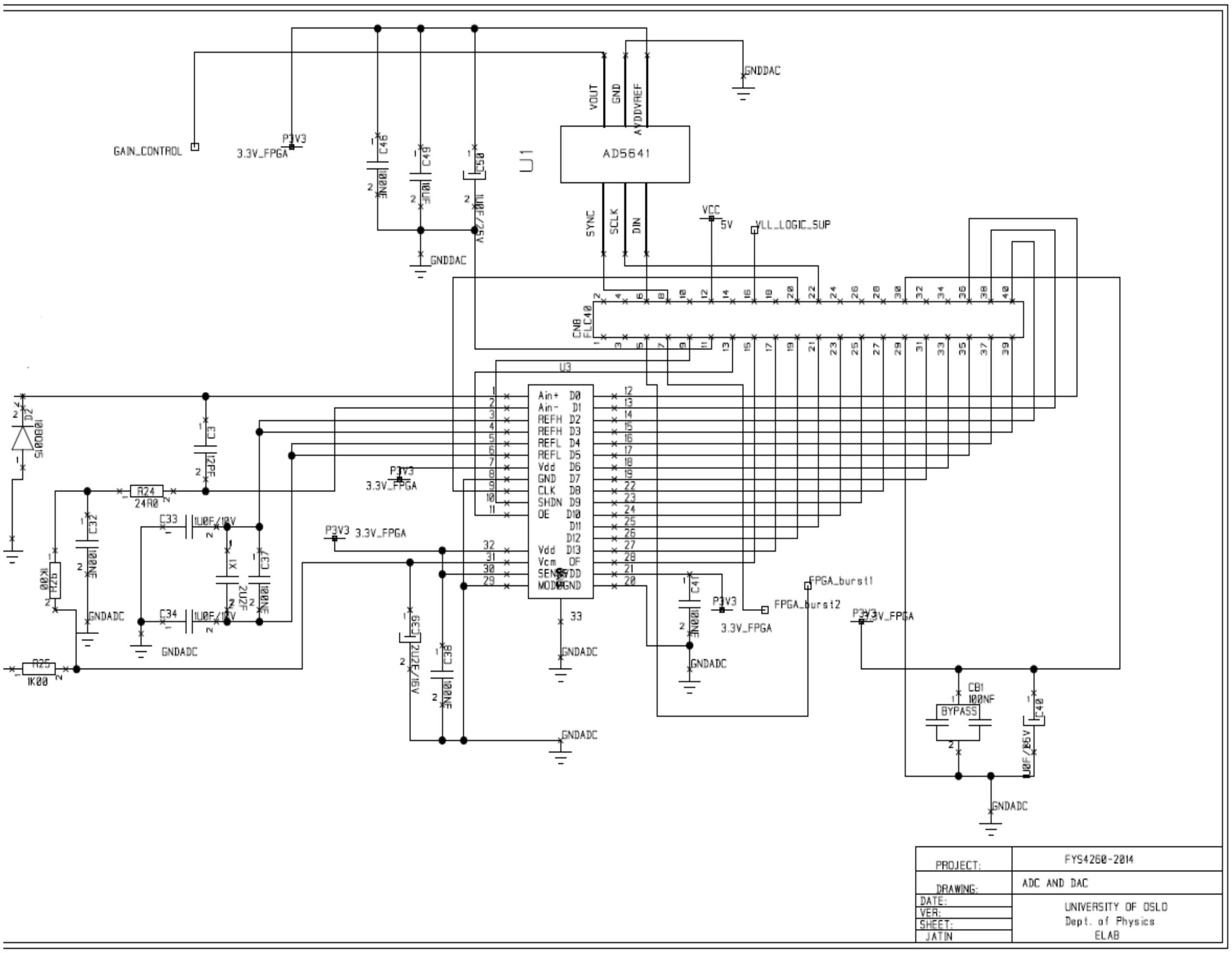


Figure 6.6: ADC-DAC and 40 Pin Connector Schematics

II PCB Layout Echo Sounder

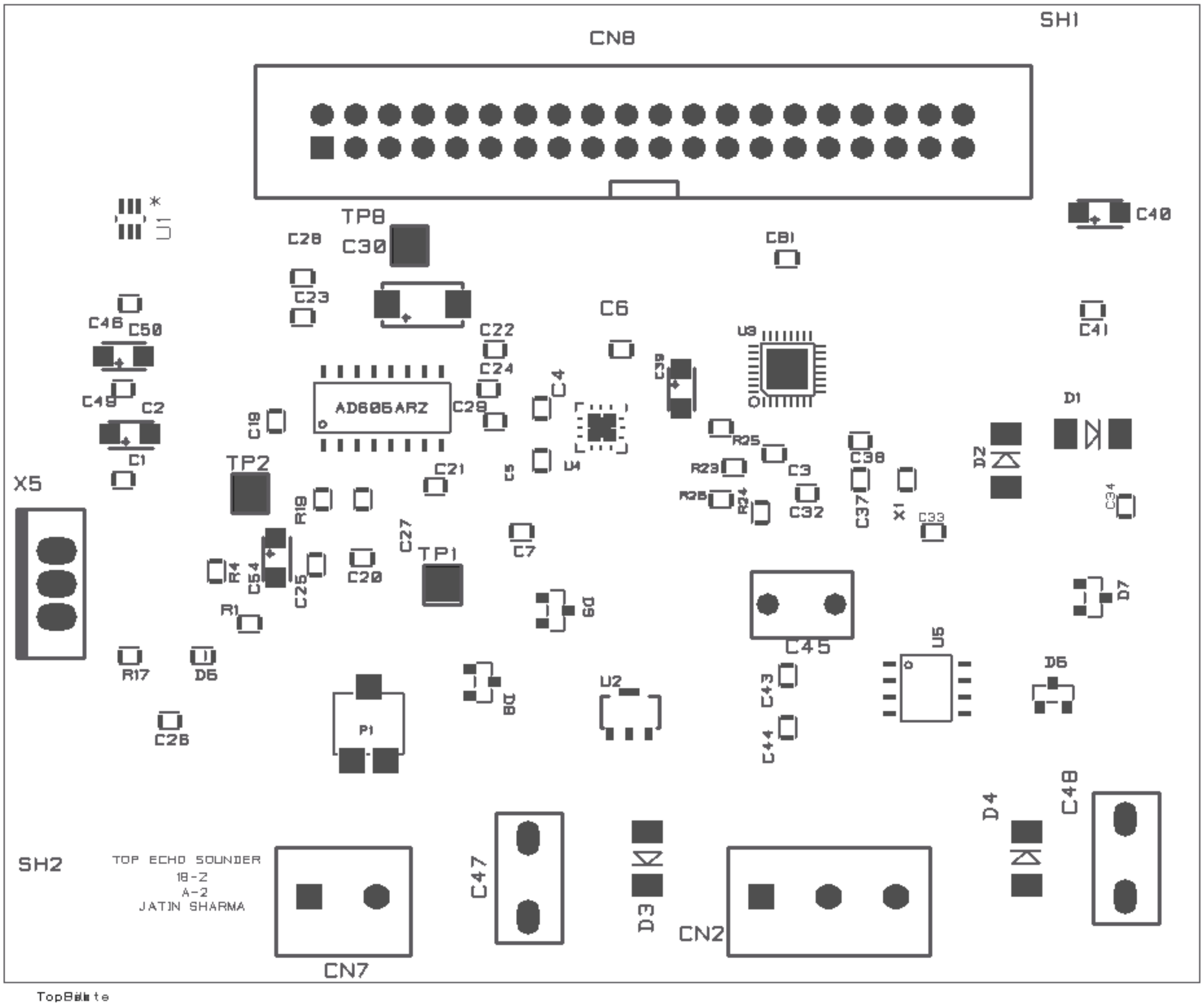


Figure 6.7: Components Placement

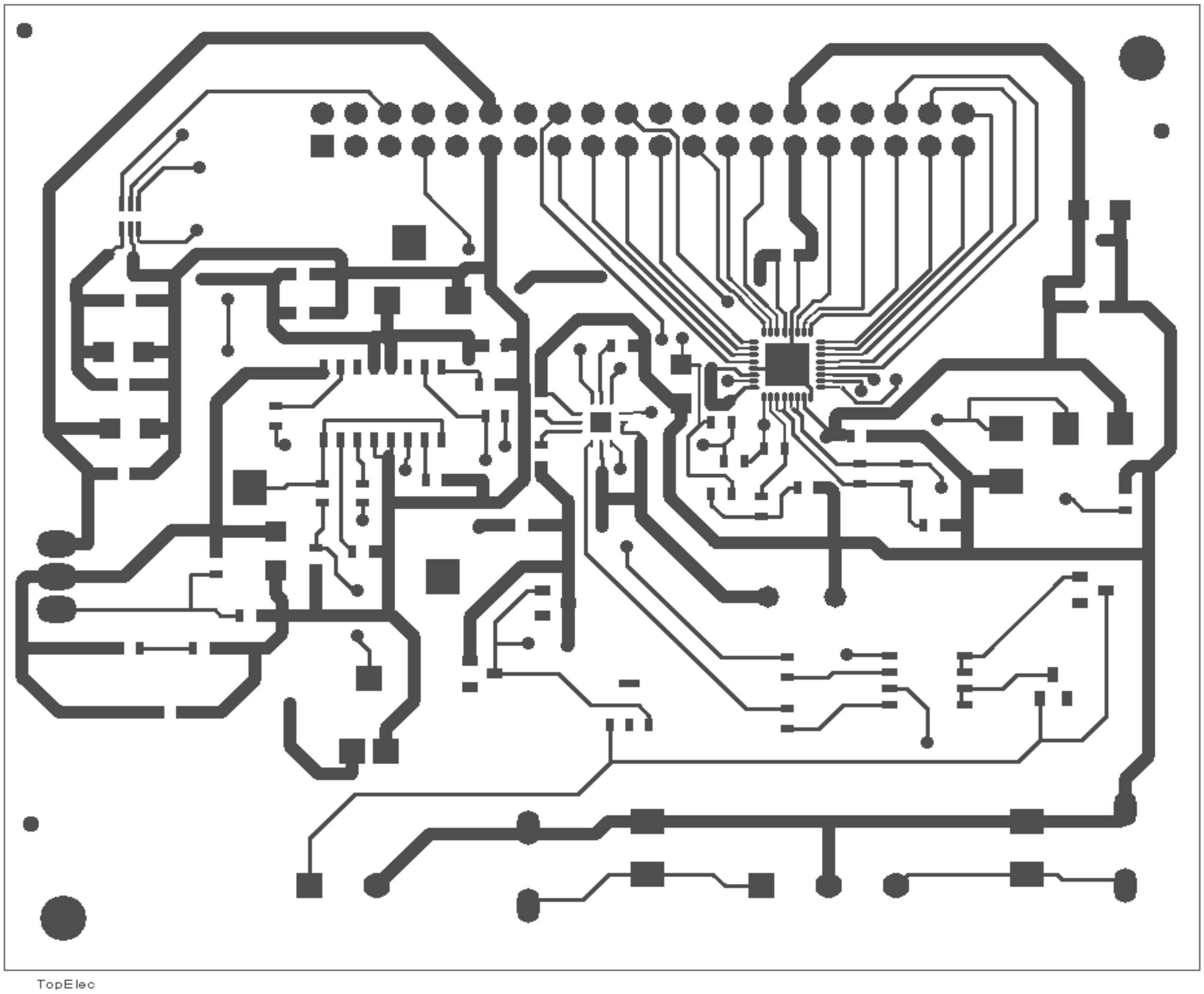


Figure 6.8: Routing on Top Electric Layer

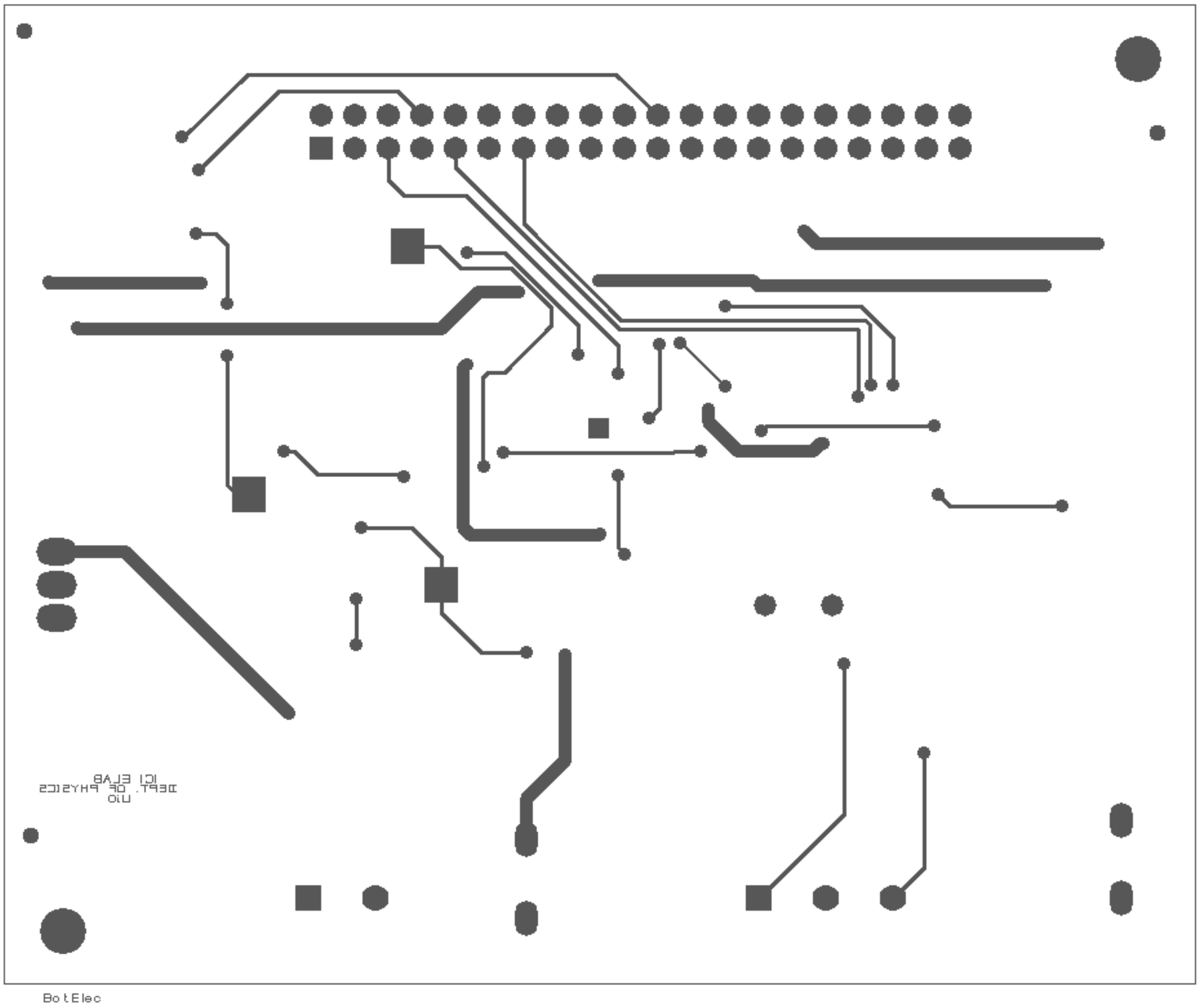


Figure 6.9: Routing on Bottom Electric Layer

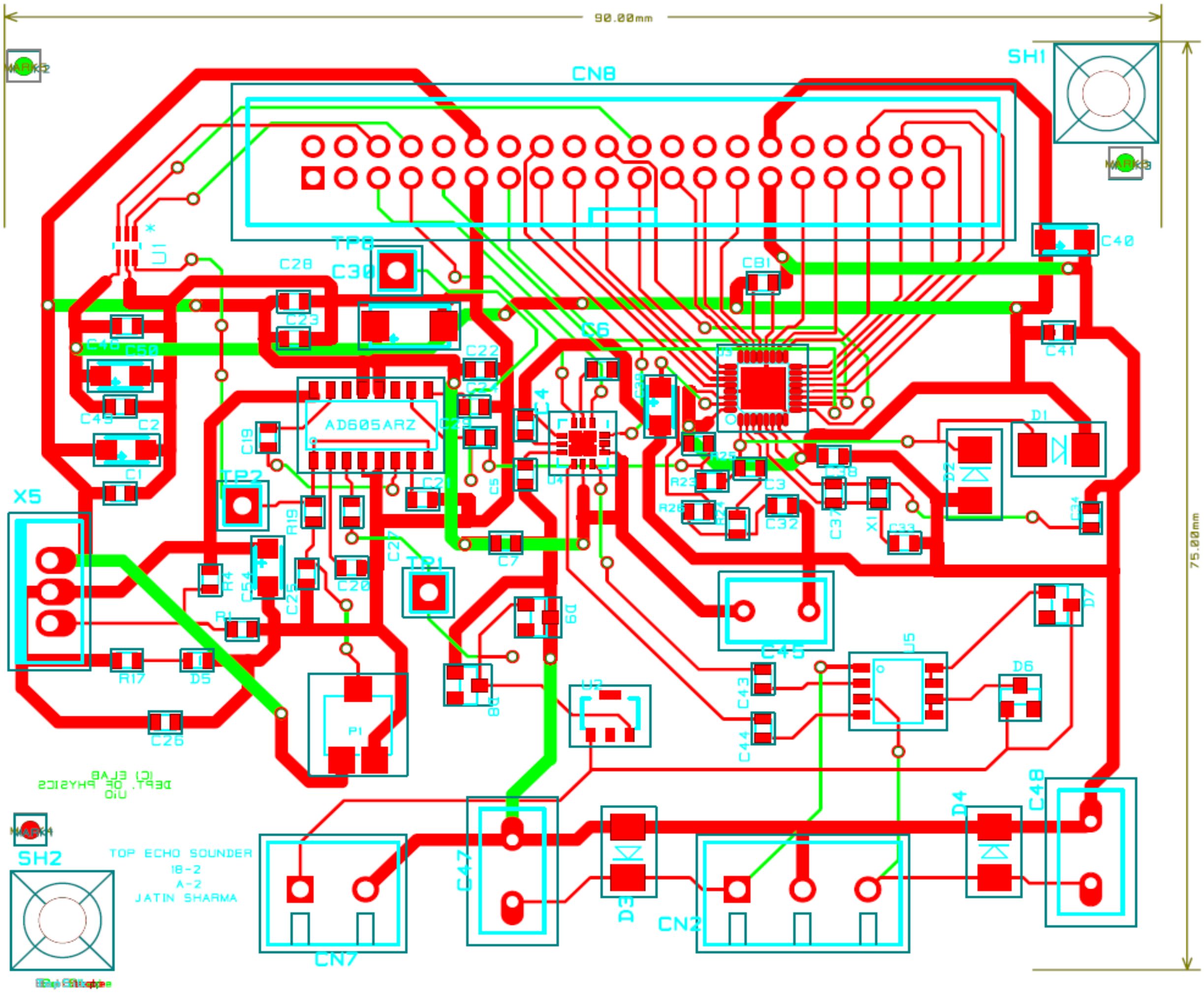


Figure 6.10: PCB Layout

III Circuit Schematics of High Voltage Board

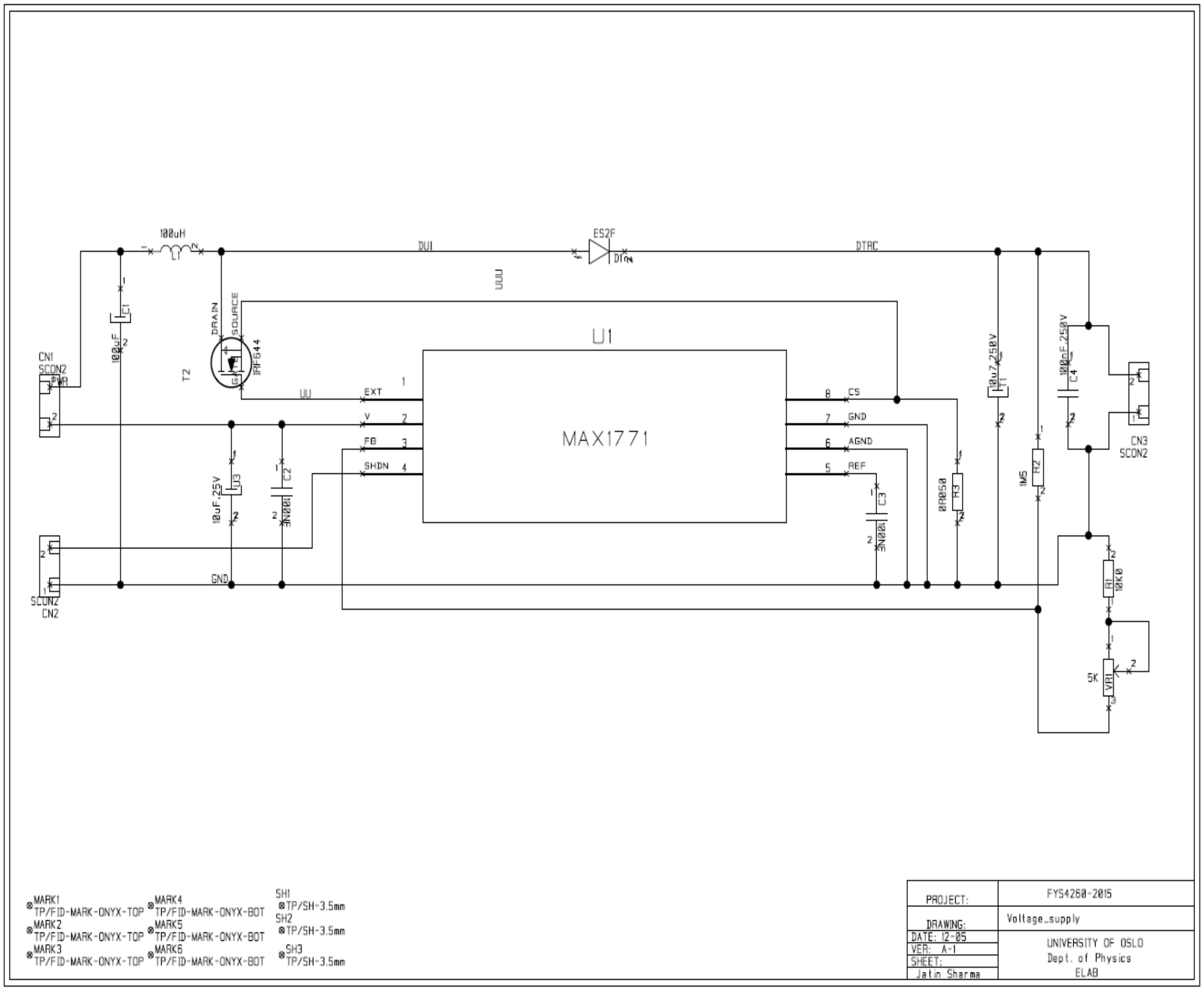


Figure 6.11: HV MAX1771 Board Schematics

IV PCB Layout of High Voltage Board

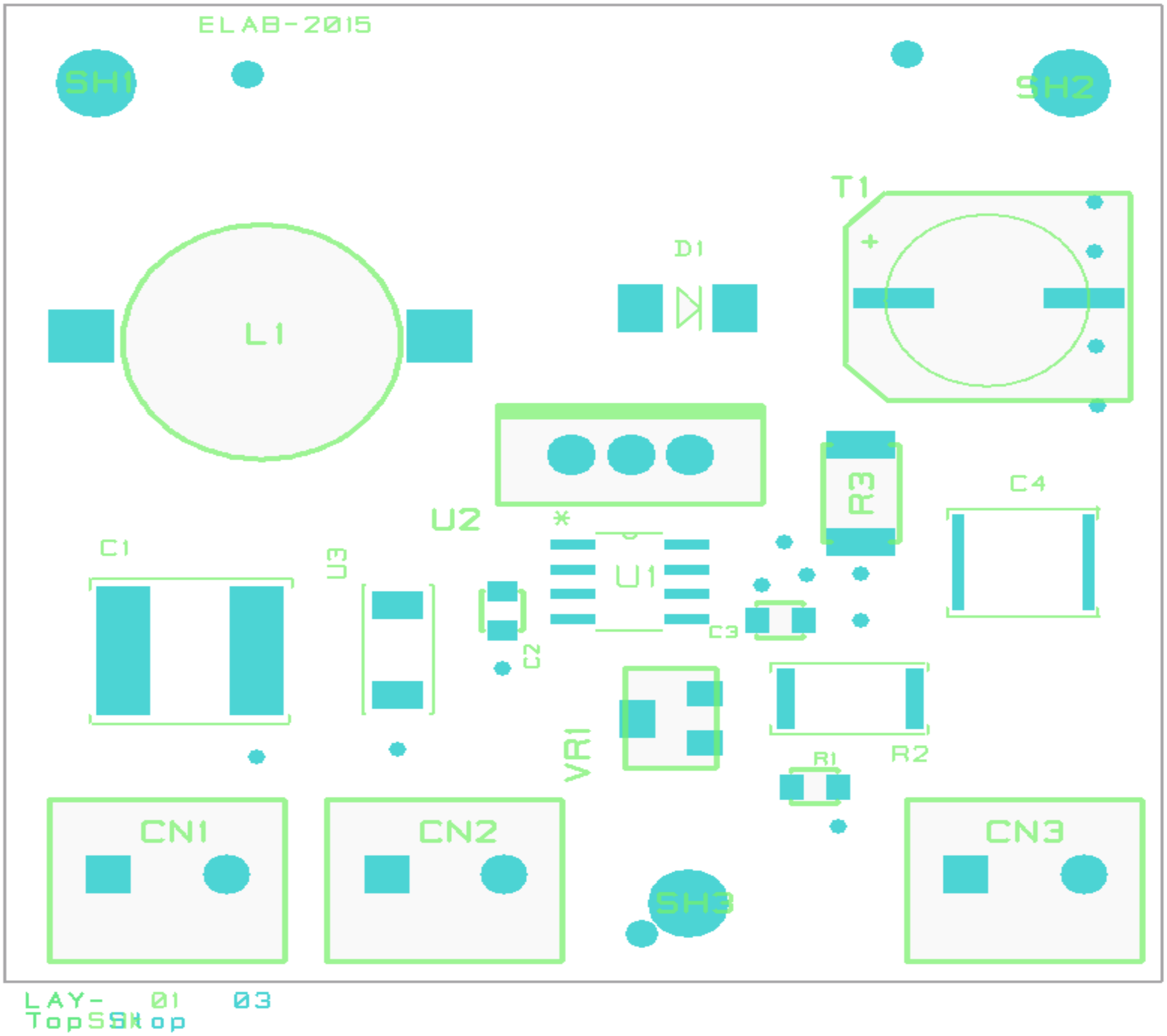


Figure 6.12: Components Placement

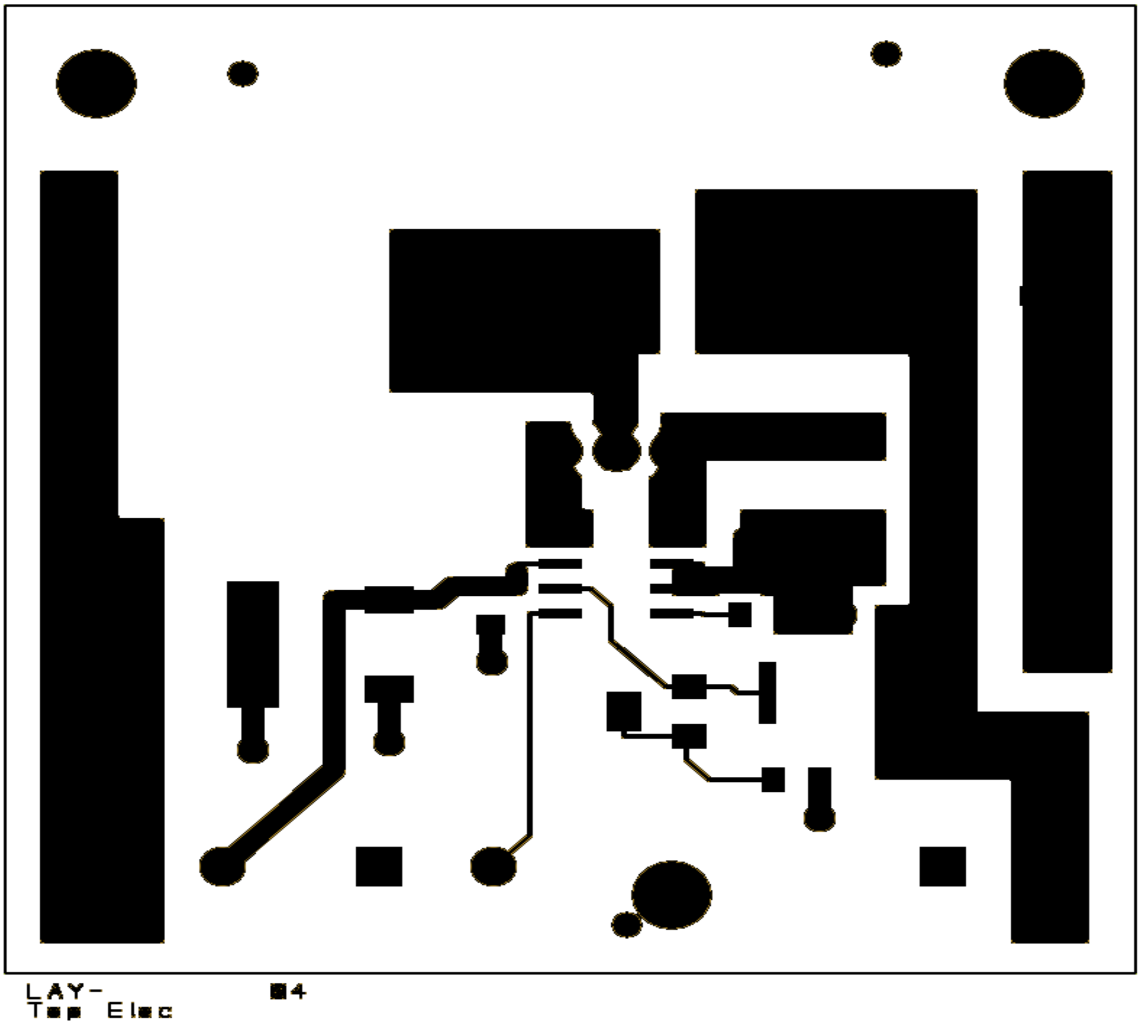


Figure 6.13: Routing on Top Electric Layer

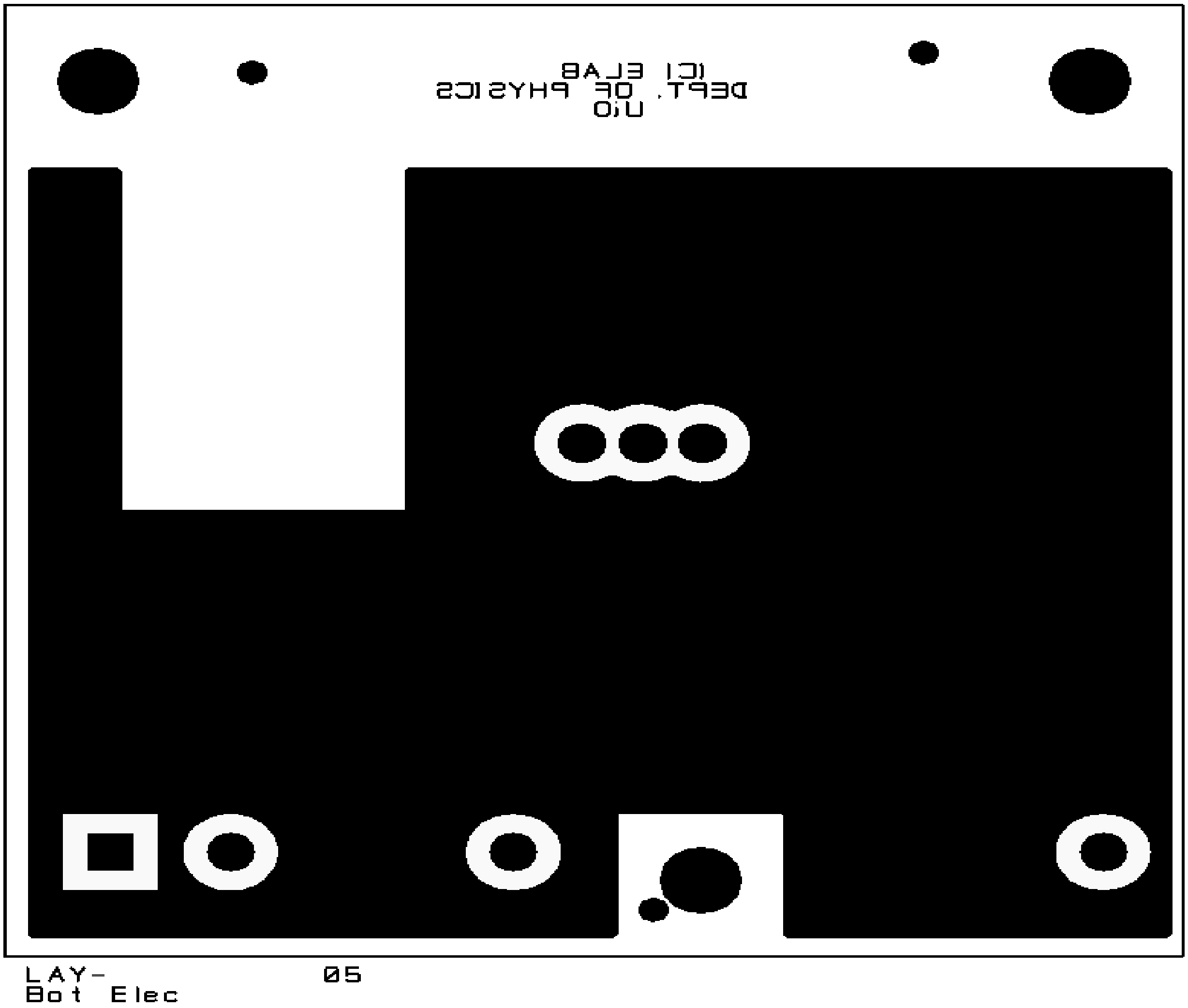


Figure 6.14: Routing on Bottom Electric Layer

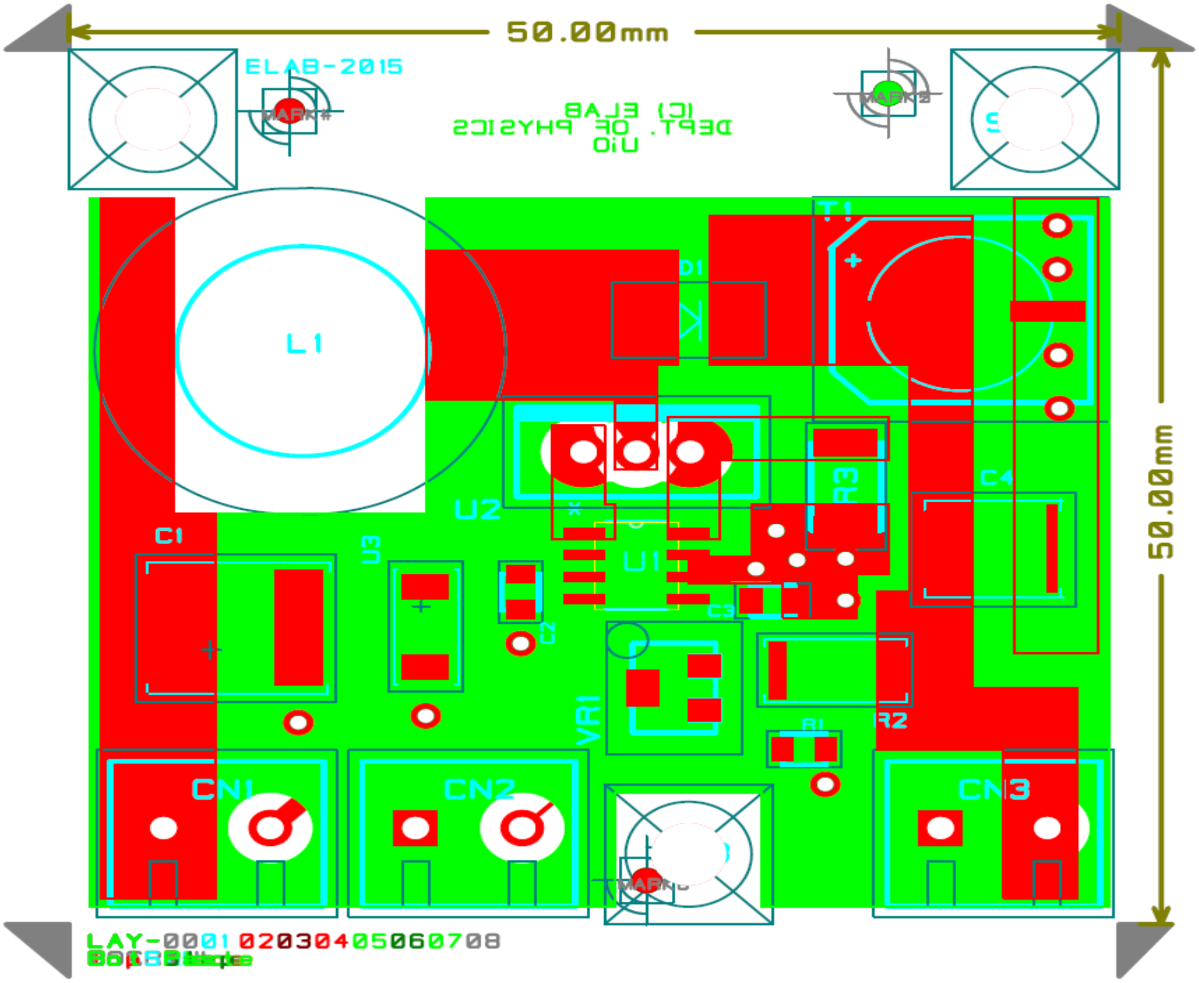


Figure 6.15: PCB Layout

V PSpice Simulation

AD605 Simulation

Circuit Description File(.cir)

```
 ***** ID# 0 ********

 ** Profile: "SCHEMATIC1-cascadedamp"  [ M:\cadence\simulations
\simulationcascaded\cascadedamp-pspicefiles\schematic1
\cascadedamp.sim


 ****      CIRCUIT DESCRIPTION


**************************************************


** Creating circuit file "cascadedamp.cir"


*Libraries:
* Profile Libraries :
* Local Libraries :
.LIB "C:/Cadence/SPB_16.6/tools/pspice/library/ad605an.lib"
* From [PSPICE NETLIST] section of C:\Users\jatinks\AppData\Roaming
\SPB_16.6\cdssetup\OrCAD_PSpice/16.6.0/PSpice.ini file:
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\ad605an.lib"
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\nom.lib"
.lib "nomd.lib"

*Analysis directives:
.AC DEC 100 100 100000K
.OPTIONS ADVCONV
.PROBE64 V(alias(*)) I(alias(*)) W(alias(*)) D(alias(*)) NOISE(alias(*))
.INC "..\SCHEMATIC1.net"
```

Netlist File(.net)

```
**** INCLUDING SCHEMATIC1.net ****
* source CASCADEDAMP
V_vin          IN N09072  AC 1
+SIN 0 .00001V 50khz 0 0 0
V_V3          VREF 0 2.5V
V_V4          VGN_GAIN 0 2V
V_V6          +5V 0 5V
C_C3          N20288 VMID  0.1u
C_C2          IN DSX+  1u
X_U1          VGN_GAIN VREF N20667 DSX+ N11903 FBK1 N20288 +5V -5V
                  AD605AN
V_V9          0 -5V 5V
X_U2          VGN_GAIN N10353 N17937 VMID N09645 FBK N26991 +5V -5V
                  AD605AN
C_C6          0 N09645  0.1u  TC=0,0
R_R1          0 UT  10k TC=0,0
C_C23          0 N11903  0.1u
R_R2          0 N09072  1M TC=0,0
C_C24          0 +5V  10u  TC=0,0
C_C25          0 +5V  0.1u  TC=0,0
C_C26          0 -5V  0.1u  TC=0,0
C_C27          0 -5V  0.1u  TC=0,0
C_C28          0 N20667  0.1u
C_C29          0 N17937  0.1u
C_C30          0 VGN_GAIN  1n  TC=0,0
C_C31          0 VREF  0.1u
C_C33          N26991 UT  0.1u
V_V12          N10353 0 2.5V
C_C32          0 N10353  0.1u

**** RESUMING cascadedamp.cir ****
.END
```

Rectifier Simulation

Circuit Description File(.cir)

```
***** ID# 0 ********

 ** Profile: "SCHEMATIC1-likeretter"  [ M:\CADENCE\likeretter
-PSpiceFiles\SCHEMATIC1\likeretter.sim ]


 ****     CIRCUIT DESCRIPTION


******************************************************************************




** Creating circuit file "likeretter.cir"

*Libraries:
* Profile Libraries :
* Local Libraries :
* From [PSPICE NETLIST] section of C:\Users\jatinks\AppData\Roaming\SPB_16.6\
cdssetup\OrCAD_PSpice/16.6.0/PSpice.ini file:
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\ad605an.lib"
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\nom.lib"
.lib "nomd.lib"

*Analysis directives:
.TRAN  0 26us 0 20ns
.OPTIONS ADVCONV
.PROBE64 V(alias(*)) I(alias(*)) W(alias(*)) D(alias(*)) NOISE(alias(*))
.INC "..\SCHEMATIC1.net"



**** INCLUDING SCHEMATIC1.net ****
* source LIKERETTER
E_U3         VUTFWR 0 VALUE {LIMIT(V(0,N40314)*1E6,-15V,+15V)}
E_U4         N40116 0 VALUE {LIMIT(V(0,N40036)*1E6,-15V,+15V)}
```

```
R_R5            N40314  N40106   2k  TC=0,0
R_R4            N40036  N40144   1k  TC=0,0
D_D2            N40116  N40036  D1N3940
D_D1            N40144  N40116  D1N3940
R_R7            VUTFWR  N40314   2k  TC=0,0
R_R3            N40106  N40036   1k  TC=0,0
R_R6            N40144  N40314   1k  TC=0,0
V_vin            N40106  N40571   AC  1
+SIN  0  .4V  200khz  0  0  0
R_R2            0  N40571    1M TC=0,0

**** RESUMING likeretter.cir ****
.END
```

ADC Protection Simulation

Circuit Description File(.cir)

```
***** ID# 0 ********

 ** Profile: "SCHEMATIC1-antialiasing"  [ M:\cadence\
antialiasing filter-pspicefiles\schematic1\antialiasing.sim ]


 ****     CIRCUIT DESCRIPTION


****************************************




** Creating circuit file "antialiasing.cir"

*Libraries:
* Profile Libraries :
* Local Libraries :
.LIB "C:/Cadence/SPB_16.6/tools/capture/library/10bq015.lib"
* From [PSPICE NETLIST] section of C:\Users\jatinks\AppData
\Roaming\SPB_16.6\cdssetup\OrCAD_PSpice/16.6.0/PSpice.ini file:
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\ad605an.lib"
.lib "C:\Cadence\SPB_16.6\tools\pspice\library\nom.lib"
.lib "nomd.lib"

*Analysis directives:
.TRAN  0 .1ms 0 10ns
.OPTIONS ADVCONV
.PROBE64 V(alias(*)) I(alias(*)) W(alias(*)) D(alias(*)) NOISE(alias(*))
.INC "..\SCHEMATIC1.net"



**** INCLUDING SCHEMATIC1.net ****
* source ANTIALIASING FILTER
R_R2          0 N10978  1k TC=0,0
```

```
V_vin            N10665 0  AC 1
+SIN 0 1V 50khz 0 0 0
R_R3            N10665 N10978  15 TC=0,0
C_C1            0 N10978  47n  TC=0,0
V_V1            N16479 0 3.3Vdc
D_D3            N10978 N16479 MBRS120T3
D_D4            0 N10978 MBRS120T3

**** RESUMING antialiasing.cir ****
.END
```

VI VHDL Code

Comments start with ’–’

Echo_sounder.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;


entity echo_sounder is
port(


CLOCK_50          : in std_logic;
SW                : in std_logic_vector(1 downto 0);
GPIO_0            : inout std_logic_vector(35 downto 0);
GPIO_1            : inout std_logic_vector(35 downto 0);
LEDR              : out std_logic_vector(9 downto 0)

);
end echo_sounder;

architecture arch of echo_sounder is


component pulse_tx                    -- Transmitter module for pulse generation

                port(

                clock_50     : in std_logic;
                SW           : in std_logic_vector(1 downto 0);
                GPIO_0       : inout std_logic_vector(35 downto 0)
                );
end component pulse_tx;


component DAC                        -- for Time Variable Gain

                port(
                clock_50     : in std_logic;
                GPIO_1       : inout std_logic_vector(35 downto 0)
                );
end component dac;


component ADC_PARALLEL         --for Envelope Detection

                port(

                clock_50     :  in std_logic;
                SW           :  in std_logic_vector(1 downto 0);
                GPIO_1       :  inout std_logic_vector(35 downto 0)
                );
end component ADC_PARALLEL;



begin

Transmitter     : pulse_tx port map(
                clock_50   => CLOCK_50,
                SW         => SW,
                GPIO_0     => GPIO_0
                );


TVG             : DAC  port map(
                clock_50   => CLOCK_50,
                GPIO_1     => GPIO_1
                );


Receiver        : ADC_PARALLEL port map(
                clock_50   => CLOCK_50,
                SW         => SW,
                GPIO_1     => GPIO_1

                );
                end architecture arch;
```

pulse_tx.vhd

```
-- FOR GENERATION OF 50 AND 200KHZ PULSES


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;


entity pulse_tx is

port(
clock_50     : in std_logic;
SW           : in std_logic_vector(1 downto 0);
GPIO_0       : inout std_logic_vector(35 downto 0)

);
end pulse_tx;


architecture arch of pulse_tx is


type state_type is (init, write_state, idle_state);  --state types

--signal declarations
signal state   : state_type := init;
signal clk_en  : std_logic := ’0’;
signal counter : integer range 0 to 9 := 0;
signal count   : integer range 0 to 500000 := 0;


begin

-- GPIO_0(0) <= OE PIN OF MD1213;
-- GPIO_0(1) <= INA PIN OF MD1213;
-- GPIO_0(3) <= INB PIN OF MD1213;

GPIO_0(35 downto 4) <= (others => ’0’);    -- REST 32 PINS OF GPIO ARE SET TO ’0’
GPIO_0(2) <=  clk_en;                      -- CLK EN FOR 5MHZ FREQUENCY


process(clock_50)            -- CLOCK DIVISION PROCESS
```

```
begin
if(rising_edge(clock_50)) then

if (counter = 9) then
clk_en <= '1';
counter <= 0;

else
counter <= counter + 1;
clk_en <= '0';

end if;
end if;

end process;


process(clock_50)    -- MAIN PROCESS

begin

if(rising_edge(clock_50)) then
if (clk_en = '1') then


case state is

when init =>         -- OE LOW, NO CHANGE IN INPUTS

GPIO_0(0) <= '0';
GPIO_0(1) <= '0';    -- INA BEING LOW AND
GPIO_0(3) <= '1';    -- INB BEING HIGH FOR 100 CYCLES

if (count = 100) then
count <= count + 1;
state <= write_state;

else
count <= count + 1;
state <= init;
end if;


when write_state => -- WRITE PROCESS FOR GENERATION OF PULSES
```

```
if (count > 100 and count < 250000) then -- IN THIS INTERVAL PULSES ARE MADE

if (SW(0) = ’1’) then -- CHECK FOR SW(0) ON FPGA BOARD
                      -- IF IT’S HIGH THEN 50 KHZ PULSES ELSE 200KHZ PULSES


count <= count + 1;

if  (count = 190000) then   -- OE GOES HIGH

GPIO_0(0) <= ’1’;
state <= write_state;

elsif(count = 200000) then   -- NOW INA  AND INB ARE TOGGLED TO PRODUCE PULSES

GPIO_0(0) <= ’1’;
GPIO_0(1) <= ’1’;
state <= write_state;

elsif(count = 200050 or count = 200150 or count = 200250 or count = 200350  ) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200100  or count = 200200 or count = 200300 or count = 200400) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200450 or count = 200550  or count = 200650 or count = 200750 ) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200500 or count = 200600 or count = 200700 or count = 200800) then
```

```
GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200850 or count = 200950 or count = 201050 or count = 201150) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif (count = 200900 or count = 201000 or count = 201100 or count = 201200) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 201250 or count = 201350 or count = 201450 or count = 201550) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 201300 or count = 201400 or count = 201500 or count = 201600) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 201650 or count = 201750 or count = 201850 or count = 201950) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
```

```
 elsif(count = 201700 or count = 201800 or count = 201900) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;


elsif(count = 202000) then

GPIO_0(0) <= ’1’;
GPIO_0(3) <= ’1’;

state <= write_state;


else

state <= write_state;
end if;



else            -- if SW(0) = ’0’ then 200KHZ PULSES
                -- 5us TIME PERIOD pulses

count <= count + 1;

if (count = 190000) then

GPIO_0(0) <= ’1’;
state <= write_state;

elsif(count = 200000) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= ’1’;
state <= write_state;

elsif(count = 200013 or count = 200038 or count = 200063 or count = 200088) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
```

```
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200025 or count = 200050 or count = 200075 or count = 200100) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200113 or count = 200138 or count = 200163 or count = 200188) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200125 or count = 200150 or count = 200175 or count = 200200) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200213 or count = 200238 or count = 200263 or count = 200288) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200225 or count = 200250 or count = 200275 or count = 200300) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200313 or count = 200338 or count = 200363 or count = 200388) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
```

```
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200325 or count = 200350 or count = 200375 or count = 200400) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200413 or count = 200438 or count = 200463 or count = 200488) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200425 or count = 200450 or count = 200475 or count = 200500) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200513 or count = 200538 or count = 200563 or count = 200588) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 200525 or count = 200550 or count = 200575 or count = 200600) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 200613 or count = 200638 or count = 200663 or count = 200688) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
```

```
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200625 or count = 200650 or count = 200675 or count = 200700) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 200713 or count = 200738 or count = 200763 or count = 200788) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200725 or count = 200750 or count = 200775 or count = 200800) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 200713 or count = 200738 or count = 200763 or count = 200788) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200725 or count = 200750 or count = 200775 or count = 200800) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 200713 or count = 200738 or count = 200763 or count = 200788) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
```

```
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200725 or count = 200750 or count = 200775 or count = 200800) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 200813 or count = 200838 or count = 200863 or count = 200888) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200825 or count = 200850 or count = 200875 or count = 200900) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 200913 or count = 200938 or count = 200963 or count = 200988) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;

elsif(count = 200925 or count = 200950 or count = 200975 or count = 201000) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
GPIO_0(3) <= GPIO_0(3) XOR ’1’;

state <= write_state;
elsif(count = 201013 or count = 201038 or count = 201063 or count = 201088) then

GPIO_0(0) <= ’1’;
GPIO_0(1) <= GPIO_0(1) XOR ’1’;
```

```
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 201025 or count = 201050 or count = 201075 or count = 201100) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 201113 or count = 201138 or count = 201163 or count = 201188) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 201125 or count = 201150 or count = 201175 or count = 201200) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 201213 or count = 201238 or count = 201263 or count = 201288) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 201225 or count = 201250 or count = 201275 or count = 201300) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 201313 or count = 201338 or count = 201363 or count = 201388) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
```

```
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 201325 or count = 201350 or count = 201375 or count = 201400) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;
elsif(count = 201413 or count = 201438 or count = 201463 or count = 201488) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;

elsif(count = 201425 or count = 201450 or count = 201475 ) then

GPIO_0(0) <= '1';
GPIO_0(1) <= GPIO_0(1) XOR '1';
GPIO_0(3) <= GPIO_0(3) XOR '1';

state <= write_state;


elsif(count = 201500) then

GPIO_0(0) <= '1';
GPIO_0(3) <= '1';

state <= write_state;

else

state <= write_state;

end if;


end if;
```

```
elsif(count = 250000) then

count <= count + 1;
state <= idle_state;

end if;


when idle_state =>     --  STOP GENERATION OF PULSES

if(count > 250000 and count < 500000) then
GPIO_0(0)  <= '0';     -- OE GOES LOW

if (count = 499999) then  -- BACK TO WRITE STATE FOR 10Hz PRF
count <= 101;
state <= write_state;

else

count <= count + 1;
state <= idle_state;

end if;

end if;


end case;

end if;
end if;

end process;


end arch;
```

DAC.vhd

```
-- TVG FOR FULL GAIN RANGE i.e. up to 2.9V
-- EVERY TIME RAMP IS INCREASED BY ADDING '4' TO THE PREVIOUSLY STORED VALUE



LIBRARY IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;


entity DAC_TVG is
port(
clock_50 : in std_logic;
GPIO_0   : inout std_logic_vector(35 downto 0)

);
end DAC_TVG;

architecture arch of DAC_TVG is

--SIGNAL DECLARATIONS
type state_type is (init, adding, sending_1, sending_2, ramp,
                        sync_high,sending_3,hold_state);

signal present_state : state_type ;
signal data_i        : std_logic_vector(13 downto 0) := "00000000000100";
                                         --INCREASE BY '4' TO GET 2.9V RAMP
signal hold_memory     : std_logic_vector(13 downto 0) := "11100000111110";
                                         --DIG. VALUE OF 2.9V
signal clk_en          : std_logic := '0';
signal memory_i        : std_logic_vector(13 downto 0) := "00000000000000";
signal i               : integer range 0 to 14 := 0;
signal count : integer range 0 to 4 := 0;
signal n : integer range 0 to 14 := 0;
signal counter       : integer range 0 to 73 := 0;
signal sample_count  : integer range 0 to 5400 := 0;


begin

GPIO_0(1) <= clk_en;
--GPIO_1(0) = SYNC PIN OF DAC;
--GPIO_1(2) = SDIN PIN OF DAC,
GPIO_0(35 downto 4) <= (others => '0'); --REST 32 PINS OF GPIO_1 ARE SET TO '0'

process(clock_50)

begin

if (rising_edge(clock_50)) then
 if (count = 4) then           -- divide TO GET A 5Mhz clock

 clk_en   <= clk_en xor '1';
 count <= 0;

 else
 count <= count + 1;

 end if;
end if;

end process;



process(clk_en)     --MAIN PROCESS


begin

if (rising_edge(clk_en)) then

case present_state is


when init=>

sample_count <= 0;
GPIO_0(0)          <= '1';
GPIO_0(2)          <= '0';
memory_i       <= (others => '0');
present_state <= adding;


when adding =>

GPIO_0(0)    <=   '1';
counter       <=    0;
```

```
GPIO_0(2)          <= '0';

memory_i <= memory_i + data_i;


if(sample_count< 5400) then
present_state <= sending_1;
else
present_state <= init;
end if;


when sending_1 =>
GPIO_0(0)              <= '0';
GPIO_0(2)              <= '0';
present_state    <= sending_2;

when sending_2 =>
GPIO_0(0)              <= '0';
GPIO_0(2)              <= '0';
if(sample_count<3601) then          -- RAMP UNTIL 3600
present_state    <= ramp;

else
present_state <= hold_state;        --THEN HOLD THIS VALUE OF 2.9V
end if;

when hold_state =>
GPIO_0(0) <= '0';
GPIO_0(2) <= hold_memory(13- n);

n <= n +1;
if (n>13) then
n <= 0;
present_state <= sync_high;

else
present_state <= hold_state;
end if;


when ramp =>
GPIO_0(0)              <= '0';
GPIO_0(2)              <= memory_i(13-i);
```

```
i <= i +1;
if (i >13) then
i <= 0;
present_state   <= sync_high;

else
present_state  <= ramp;
end if;


when sync_high =>
GPIO_0(0)             <=  '1';
GPIO_0(2)             <=   '0';
present_state   <= sending_3;


when sending_3 =>
counter  <= counter + 1;
GPIO_0(0)      <=  '1';
GPIO_0(2)      <=  '0';

if (counter = 73) then
sample_count  <= sample_count + 1;
present_state <= adding;

else
present_state <= sending_3;
end if;


when others =>
GPIO_0(0)             <=   '1';
present_state   <=  sending_3;



end case;

end if;


end process;


end architecture;
```

DAC_TVG.VHD

```
-- TVG FOR HALF GAIN RANGE, i.e. UP TO 1.5V



library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;


entity dac is

port(
clock_50 : in std_logic;
GPIO_1   : inout std_logic_vector(35 downto 0)
);

end dac;


architecture arch of dac is


--SIGNAL DECLARATIONS
type state_type is (init, adding, sending_1, sending_2, ramp,
                    sync_high,sending_3,hold_state); -- STATES FOR RAMP GENERATION

signal present_state : state_type;


signal data_i   : std_logic_vector(13 downto 0) := "00000000000001";  -- FOR
                                               -- INCREASING THE RAMP EVERY TIME


signal hold_memory : std_logic_vector(13 downto 0) := "01110100011000";--DIG. VALUE
                                                                        -- OF 1.5V
signal clk_en              : std_logic := '0';
signal memory_i            : std_logic_vector(13 downto 0);
signal i                   : integer range 0 to 14 := 0;
signal count               : integer range 0 to 5 := 0;
signal clk_ena             : std_logic  := '1';
signal n                   : integer range 0 to 14 := 0;
```

```
begin


GPIO_1(1) <= clk_en;      --CLK FOR DAC



process(clock_50)      --CLOCK DIVISION PROCESS

begin

if (rising_edge(clock_50)) then

if (count = 4) then        -- DIVIDE SYSTEM CLOCK TO GET CLK EN
count <= 0;
clk_en    <= clk_en xor '1';

else

count <= count + 1;
end if;

end if;

end process;



process(clk_en)        -- MAIN PROCESS

variable counter        : integer range 0 to 55 := 0; -- WAITS FOR 55 CYCLES EVERY
                                                      -- TIME A NEW VALUE IS ADDED TO INCREASE
                                                      --  THE VALUE OF SAMPLE_CNT BY 1


variable sample_count   : integer range 0 to 7450 := 0; -- INCREASES BY 1 EVERY
                                                        -- TIME A NEW VALUE IS SENT


begin


if (clk_en'event and clk_en = '1') then
```

```
case present_state is


when init=>     --SYNC HIGH, NO WRITE, NO DATA


sample_count := 0;

GPIO_1(0)          <= '1';
GPIO_1(2)          <= '0';

memory_i      <= (others => '0');
present_state <= adding;



when adding =>  -- INCREASE VALUE OF MEMORY_I REG BY '1'

GPIO_1(0)          <=  '1';
counter       :=   0;
GPIO_1(2)          <= '0';


memory_i <= memory_i + data_i;     -- ADD VALUE OF '1' HERE


if(sample_count< 7450) then    -- CHECK IF SAMPLE_CNT <7450 GENERATE RAMP
present_state <= sending_1;

else
present_state <= init;
end if;


when sending_1 =>  -- FIRST BIT '0' OF DAC FOR NORMAL OPERATION

GPIO_1(0)             <= '0';
GPIO_1(2)             <= '0';

present_state    <= sending_2;


when sending_2 =>  -- SECOND BIT ALSO '0' FOR NORMAL OPERATION
```

```
GPIO_1(0)          <= '0';
GPIO_1(2)          <= '0';


 if(sample_count<7449) then -- RAMP UNTIL 7448
 present_state    <= ramp;

 else
 present_state <= hold_state;

end if;


when hold_state =>    -- send 14 bits serially

GPIO_1(0) <= '0';
GPIO_1(2) <= hold_memory(13- n);

n <= n +1;

 if (n>13) then
 present_state <= sync_high;

 else
 present_state <= hold_state;

end if;


when ramp =>           -- send 14 bits serially

GPIO_1(0)          <= '0';
GPIO_1(2)          <= memory_i(13-i);

i <= i +1;

 if (i >13) then
 present_state    <= sync_high;

 else
 present_state   <= ramp;

end if;
```

```
when sync_high =>      -- STOP WRITING TO DAC, NO DATA

GPIO_1(0)              <= '1';
GPIO_1(2)              <=  '0';

present_state    <= sending_3;


when sending_3 =>   -- WAIT FOR 55 CYCLES TO INCREASE THE VALUE
                    -- OF SAMPLE_CNT AND GO BACK TO ADDING
                    -- STATE FOR NEW VALUE

counter   := counter + 1;

GPIO_1(0)      <= '1';
GPIO_1(2)      <= '0';

if (counter = 55) then

sample_count   := sample_count + 1;
present_state <= adding;

else
present_state <= sending_3;
end if;


when others =>

GPIO_1(0)              <=  '1';
present_state    <=  sending_3;


end case;

end if;

end process;


end architecture;
```

ADC_PARALLEL.vhd

```
-- ENVELOPE DETECTION USING ADC AND DAC


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;



entity ADC_PARALLEL is

port(
clock_50   : in std_logic;
SW         : in std_logic_vector(1 downto 0);
GPIO_1     : inout std_logic_vector(35 downto 0)

);

end ADC_PARALLEL;



architecture arch of ADC_PARALLEL is


type state_type is (IDLE,READ_DATA,READ_DATA_200,WAIT_STATE,
                                  WRITE_DATA,WRITE_DATA_200); -- states
signal state  : state_type;


signal clk_div  : integer range 0 to 5 := 0;
signal clock_5  : std_logic := '0';
signal adc_data : signed(13 downto 0);
signal positiv_cycle : signed(12 downto 0);-- positive cycle data from adc

signal count   : integer range 0 to 60 := 0;



begin

--GPIO_1(6) = SHDN PIN OF ADC
--GPIO_1(7) = OE PIN OF ADC
```

```
--GPIO_1(31) = OF PIN OF ADC

GPIO_1(6)  <= '0'; -- SHDN TO GROUND,
                   --SHOULD BE GROUND FOR NORMAL OPERATION OF ADC

GPIO_1(1)  <= clock_5;   -- 5MHZ OUT
GPIO_1(30) <= clock_5;   -- 5MHZ OUT

GPIO_1(31) <= '1' when (adc_data = "11111111111111") else '0'; -- CONCURRENT
                                    -- STATEMENT FOR CHECKING INPUT RANGE



process(clock_50)  -- CLOCK DIVISION PROCESS

begin

if(rising_edge(clock_50)) then

if (clk_div = 4) then
clock_5 <= clock_5 xor '1';
clk_div <= 0;

else
clk_div <= clk_div + 1;
end if;

end if;

end process;



process(clock_5)   --MAIN PROCESS

begin

if (clock_5'event and clock_5= '1') then -- DATA CHANGED ON RISING EDGE OF CLOCK


case state is


when IDLE =>
```

```
GPIO_1(0) <= '1';              -- DAC SYNC HIGH, DAC NOT IN WRITE MODE
GPIO_1(2) <= '0';              -- DAC SERIAL DATA OUT
GPIO_1(7) <= '1';              -- OE high, adc disabled


if (count = 0) then

if (SW(1) = '1') then

count <= count + 1;
state <= READ_DATA;

else
count <= count + 1;
state <= READ_DATA_200;


end if;
end if;



when READ_DATA =>   --  GET INPUTS

GPIO_1(0) <= '1';            --  dac sync high, no write
GPIO_1(7) <= '0';            -- OE down to 0,adc enabled, NORMAL OPERATION


if(count = 1) then

count <= count+ 1;
positiv_cycle <= "0000000000000";
state <= READ_DATA;


elsif(count > 1 and count < 50) then   -- POSITIVE HALF


adc_data(13 downto 0) <= signed(GPIO_1(25 downto 12)); -- INPUT FROM GPIO BUS
                                                        -- 14 BITS COMING


if(positiv_cycle < adc_data(12 downto 0)) then
```

```
positiv_cycle <= abs(adc_data(12 downto 0));

count <= count + 1;
state <= READ_DATA;

else

count <= count + 1;
positiv_cycle <= positiv_cycle;
state <= READ_DATA;

end if;


elsif(count = 50) then

count <= count + 1;
state <= WAIT_STATE;


end if;



when READ_DATA_200 =>

GPIO_1(0) <= '1';   --  dac sync high, no write
GPIO_1(7) <= '0';   -- OE down to 0,adc enabled, NORMAL OPERATION

if(count = 1) then

count <= count+ 1;
positiv_cycle <= "0000000000000";

state <= READ_DATA_200;


elsif(count > 1 and count < 8) then

adc_data(13 downto 0) <= signed(GPIO_1(25 downto 12)); -- INPUT FROM GPIO BUS
                                                       -- 14 BITS COMING


if(positiv_cycle < adc_data(12 downto 0)) then
```

```
positiv_cycle <= abs(adc_data(12 downto 0));


count <= count + 1;
state <= READ_DATA_200;

else

count <= count + 1;
positiv_cycle <= positiv_cycle;
state <= READ_DATA_200;

end if;

elsif(count = 8) then

count <= count + 1;
state <= WAIT_STATE;


end if;


when WAIT_STATE =>

GPIO_1(0) <= ’1’;       -- SYNC HIGH
GPIO_1(7) <= ’1’;       -- OE BACK HIGH OF ADC LTC2245

if (count = 51) then

count <= 0;
state <= WRITE_DATA;

elsif(count = 9) then
count <= 0;
state <= WRITE_DATA_200;


end if;


when WRITE_DATA =>

GPIO_1(0) <= ’0’;    --SYNC LOW, DAC ENABLED
GPIO_1(7) <= ’1’;
```

```
if(count = 0 or count = 1) then -- NORMAL WRITE OPERATION OF DAC
                                -- FIRST TWO BITS SHOULD BE 0

GPIO_1(2) <= '0';
count <= count+1;

state <= WRITE_DATA;

elsif(count = 2) then -- HAS TO BE 0 BECAUSE
                      -- ONLY 13 BITS HAVE TO BE TRANSFERRED


GPIO_1(2) <= '0';
count <= count + 1;
state <= WRITE_DATA;

elsif(count > 2 and count < 16) then -- 13 BITS TRANSFERRED SERIALLY


GPIO_1(2) <= std_logic(positiv_cycle(15 - count));
count <= count + 1;
state <= WRITE_DATA;

elsif (count > 15 and count < 49) then  -- NO DATA SENT

GPIO_1(2) <= '0';
count <= count + 1;
state <= WRITE_DATA;

elsif (count = 49) then

GPIO_1(0) <= '1';
count <= 0;

adc_data <= "00000000000000";
state <= IDLE;


end if;



when WRITE_DATA_200 =>
```

```
GPIO_1(0) <= '0';
GPIO_1(7) <= '1';

 if(count = 0 or count = 1) then  -- NORMAL WRITE OPERATION OF DAC
                                  -- FIRST TWO BITS SHOULD BE 0


GPIO_1(2) <= '0';
count <= count+1;
state <= WRITE_DATA_200;

 elsif(count = 2) then -- HAS TO BE 0 BECAUSE
                       -- ONLY 13 BITS HAVE TO BE TRANSFERRED


GPIO_1(2) <= '0';
count <= count + 1;
state <= WRITE_DATA_200;

 elsif(count > 2 and count < 16) then -- 13 BITS TRANSFERRED SERIALLY

GPIO_1(2) <= std_logic(positiv_cycle(15 - count));

count <= count + 1;
state <= WRITE_DATA_200;


 elsif (count = 16) then

GPIO_1(0) <= '1';
GPIO_1(2) <= '0';
count <= 0;


adc_data <= "00000000000000";
state <= IDLE;


end if;



end case;
end if;

end process;


end architecture;
```