DDS WaveForm Generator

by

Brandon Wicks and Adam Elliott

A Senior Project Report Submitted to the Faculty of
Electrical, Computer, and Software Engineering
Penn State Erie, The Behrend College

Faculty Supervisor(s): Dr. Chris Coulston, Dr. Wen-Li Wang
Industrial Supervisor: Alex Margulis
Industry Sponsor: ARTEMIS Inc, Hauppauge, NY

April 2007

TABLE OF CONTENTS

FIGURES

TABLES

Abstract

The purpose of the project is to create a continuous wave (CW) analog signal generator for Advanced Radar Techniques Equipment Microwave Integrated Systems (ARTEMIS) Inc. It will be used to simulate a direct digital synthesizer (DDS) used in Synthetic Aperture Radar's (SAR). This device will replace the expensive, long lead time and custom designed DDS's.

The analog signal generator will synthesize a sinusoidal wave. Its frequency will be adjustable and be provided to the user via SubMiniature version A (SMA) connection.

1. Problem Statement

1.1. Need:

Advanced Radar Techniques Equipment Microwave Integrated Systems (ARTEMIS) Inc. needs a low cost, low frequency analog signal generator. This device will help solve their signal generator shortage and save the company money. Generators on the market today are very expensive with many features that are not needed. ARTEMIS would like a simple signal generator that will meet their needs. This generator will be used in their upconverter test setup. This test setup requires a 0 MHz to 300 MHz analog input signal used in the upconverter's translation and generation of transmit (TX) and local oscillator (LO) signals. The test setup consists of a few high frequency, one low frequency generators and a spectrum analyzer.

1.2. Objective:

Our objective is to design a portable low cost analog signal generator using the Direct Digital Synthesis technique specified by the company. This device will have a user interface that will allow the user to interactively set the output frequency. The generator's output must be available via SMA female connectors for easy access by the user. The device will also have multiple profiles. Each profile will allow the user to set and recall a particular frequency.

1.3. Background and Technology Survey:

Most signal generators on the market today (in the range of 0-300MHz) cost anywhere from $800 to $2,000 dollars and above. [1] These generators are designed to meet the needs of many different customers and therefore support a wide range of functions and special features. Many of these extra features are not needed and will not be implemented in our design. Due to limited resources, the signal generators at ARTEMIS are shared among multiple test setups. This is an inconvenience and can cause delays in system testing. The company cannot justify buying separate generators for each test setup when only some of the features will be utilized. Our task is to design a low cost signal generator for ARTEMIS that doesn't include unnecessary (and costly) features. The signal generator will be designed to output a variable sinusoidal frequency. This signal will be used as a reference for the (Direct Digital Synthesis) DDS signal. The upconverter is one of ARTEMIS's products used for translation and generation of TX and LO signals. The TX and LO signals are generated by up converting its input signals to a high frequency radio signals.

1.3.1. Radar System:

Figure 1 shows a basic radar system block diagram. The overall function of the DDS is to generate a chirp. A chirp is a quick frequency sweep. The chirp is sent through the upconverter where it is up converted to a microwave frequency. The microwave frequencies are then amplified by the SSPA (Solid State Power Amplifier) and transmitted out of the antenna. The system is then switched to receive mode where the reflected signals are captured by the antenna and sent into the receiver. The receiver then down converts the signals and sends it to a digitizer. The digitizer uses an ADC (analog to digital converter) to convert the signal into a digital representation of the reflected signal. The data is then sent to the digital signal processor to extrapolate the image.

Our waveform generator will be replacing the DDS subsystem during ARTEMIS's testing phase. It will be used to supply an analog signal to the upconverter. Each fundament frequency of the upconverter is tested and tuned to the desired level.

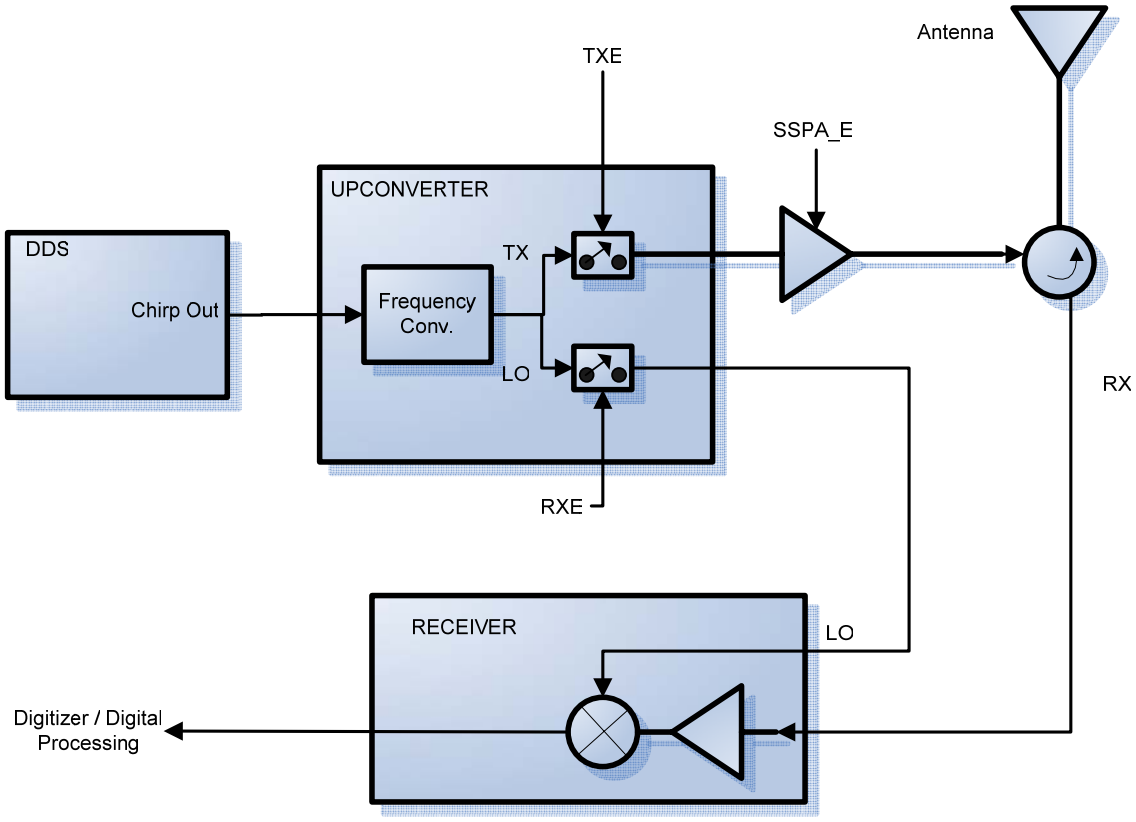

Figure 1: Basic Radar System Block Diagram

1.3.2. Basic DDS:

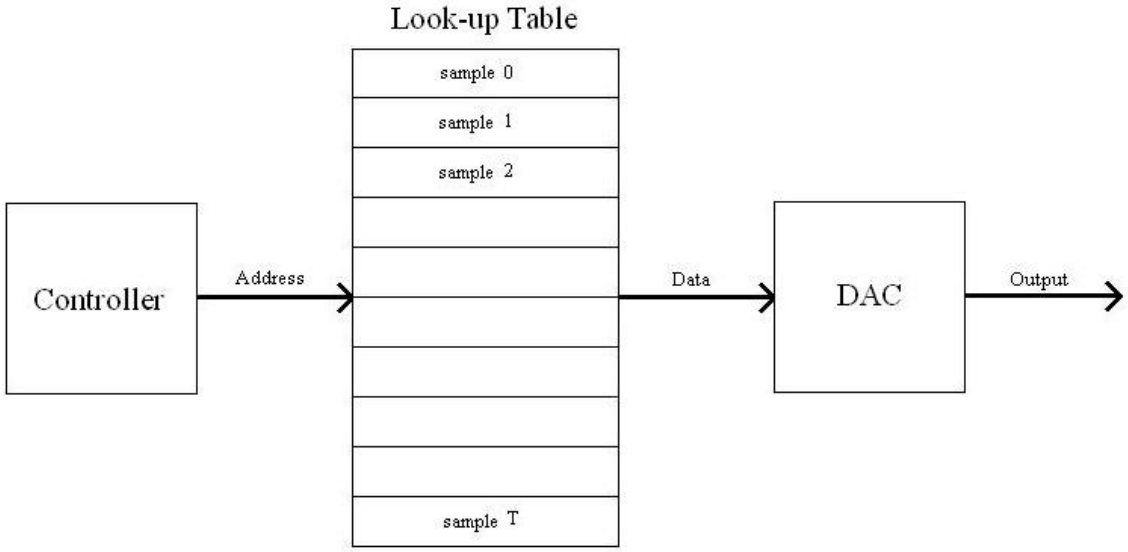

Figure 2: A basic DDS system

The basic function of a Direct Digital Synthesizer is straightforward. A lookup table holds a digital representation of the desired waveform over one period, with each row of the table containing a sample of the waveform's amplitude. A system controller steps through this lookup table at a specific rate, feeding its value of the current row to the input of a DA (digital to analog) converter. When the controller reaches the end of the lookup table it cycles back to the first row and repeats the process. This process results in the desired analog signal appearing on the output of the DAC. Since no time consuming calculations are needed during signal generation this is a notably fast technique. The output frequency can be changed by altering the values in the lookup table or reading from a separate lookup table

1.3.3. Possible Implementations of DDS:

There are multiple ways to implement a DDS system. One possibility is to directly build the DDS shown above using standard logic components. Alternately, a DDS system could be developed using a FPGA (Field Programmable Gate Array). Finally, a dedicated DDS SOC (system on a chip) could be used. Each of these implementations has advantages and disadvantages as discussed below.

Designing the system using standard logic components such as RAM memory, a micro-controller, DA converter, counter, etc would allow for a great deal of design flexibility. We would be able to choose precisely what features would be included in the design and not have to pay for other features that wouldn't be necessary. For this reason, this implementation is likely to be the cheapest of the three. One disadvantage to this solution, however, is that the design would be very complex. Extensive consideration would be required to ensure that the device could achieve the frequency range and frequency resolution specified by ARTEMIS. Also, we

could easily be “reinventing the wheel” by using this method since there are dedicated DDS chips on the market that might have the same functionality.

Using a FPGA to realize a DDS system would provide a moderate amount of design flexibility, although not as much as a basic component-based system. This implementation would reduce design complexity, since FPGAs can easily be reprogrammed and thus different design solutions could easily be tested and compared to one another. Some components may be required in addition to the FPGA, but this solution will still be much simpler than the previous one. As with the basic component-based system, this design work would be unnecessary if there is a pre-built DDS chip on the market that exhibits the functionality desired.

A dedicated DDS chip can have all of the system components integrated onto a single chip. Although designing a dedicated chip ourselves is beyond the scope of this course there are many DDS chips currently on the market that might be appropriate for this project. This method provides minimum design flexibility since we would be limited to the features already designed into the chip. However, if we find a chip that satisfies our project requirements, this will be a very viable option.

Another possible choice is to use an FPGA with a soft core. With this method, the FPGA is configured to operate as a reduced instruction set (RISC) processor. Software can then be written for this processor in assembly or even the C programming language. We would then write a program to implement a DDS system and load it onto the FPGA/RISC processor.

1.3.4. Profile Definition:

One of the marketing requirements below is that the device must be able to store and restore waveform settings. Throughout this document, we refer to the collection of all waveform settings that describe a particular output, as a profile. The user will be able to manipulate waveform settings until the desired output is achieved and then store these settings in a user profile. This profile may then be selected at a later time to restore the output waveform that was stored earlier.

1.3.5. Anti-Alias Filter:

At the output of the DSS signal an anti-aliasing filter is needed to reconstruct the analog signal. During the Direct Digital Synthesis, aliases are created and need to be removed from the output waveform. These aliases are created since the direct digital synthesis system is a sampled data system. The aliases occur when frequency components are greater than half the sample rate. There are four different types of filters (Elliptic, Chebyshev, Butterworth, and Bessel) that can be used to remove these aliases. Below in figure 3 is a diagram of the four different types of filters and their cut off response.

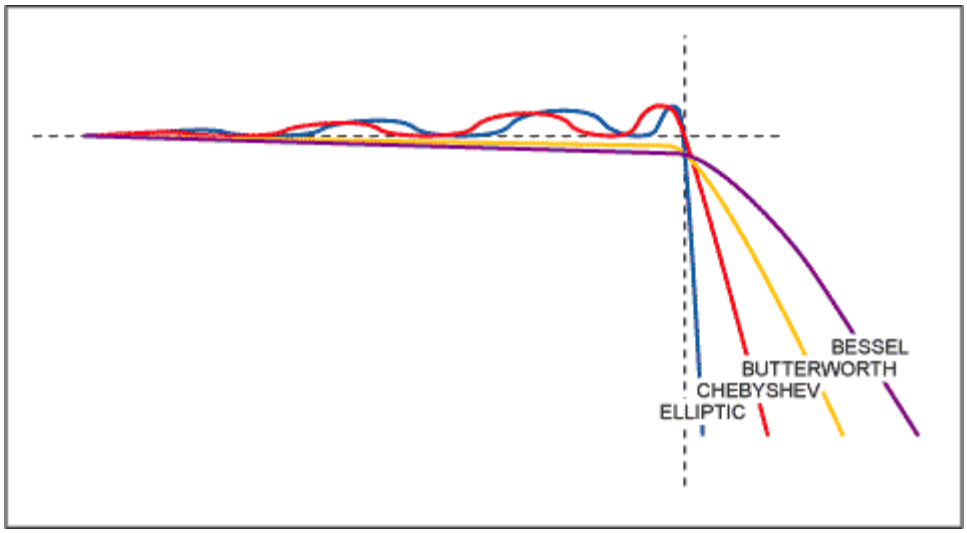

Figure 3: Filter cut off response [2]

If you look to the left of the vertical dotted line this is the pass band. This is the area in which the signals are allowed to pass. To the right of the vertical dotted lines is the stop band this is the area in which signals are rejected. If you look at the individual filter curves your see a slight ripple on a few of the responses. This is the attenuation (decreased power level) of the signal. We will be choosing a filter that will have a sharp enough transition to allow want signals through and reject unwanted signals. We also have to take into consideration a filter with the least amount of attenuation. With the help of ARTEMIS and an application called Microwave Office we will be designing a filter to meet our needs. The designed filter will likely be either an Elliptic or Chebyshev filter with multiple segments to handle a wide range of frequencies.

1.3.6. JTAG:

JTAG is an acronym for Joint Test Action Group. Its standard entitled Standard Test Access Port and Boundary-Scan Architecture is used for testing circuit boards via boundary scan. It can also be used as a programming protocol. This five-pin interface is designed so that all onboard chips have the JTAG lines daisy chained together as shown in figure 4. This allows for access to all the onboard chips. The five-pins (TMS, TCK, TRST, TDI, and TDO) are defined as follows.

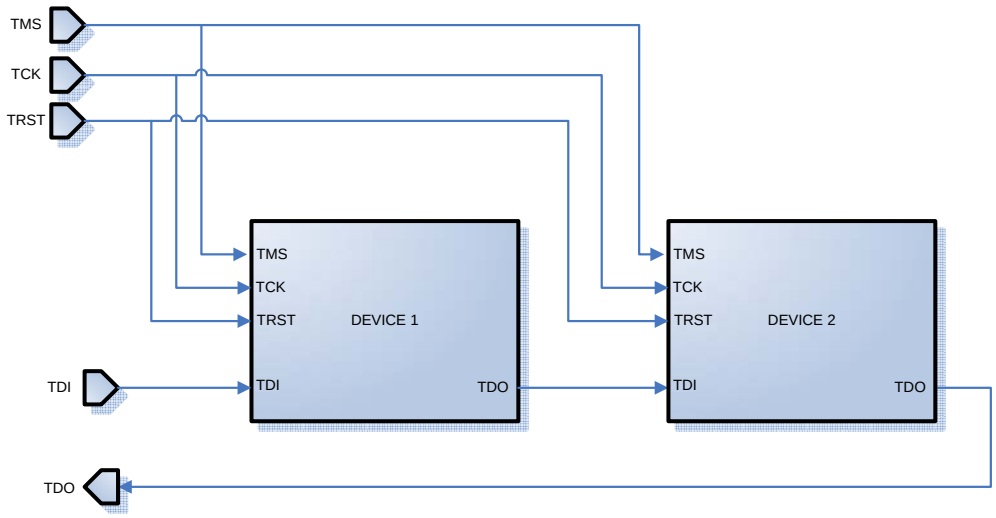

Figure 4: JTAG daisy chained circuitry

TMS stands for Test Mode Select this is used to determine the next state. TCK is Test Clock which synchronizes all of the internal finite state machine operations. TRST stands Test Reset this resets the internal finite state machine. TDI and TDO are the Test Data In and Out. If you could imagine a giant shift register this is how the TDI and TDO work. The Data is pushed into the TDI and shifts everything to the right and the last bit is pushed out the TDO. This protocol has a lot of advantages and can be used for in-circuit debugging, flashing on board chips, and chip identification.

1.3.7. Needs Identification

1.3.7.1. Marketing Requirements

1.3.7.1.1. List of Marketing Requirements:

The system should:

- Have fine frequency resolution.
- Have outputs available on SMA female connectors.
- Be able to generate a wide range of output frequencies.
- Be able to store and restore waveform settings.
- Have an easy to use interface.
- Be small, portable and lightweight.

1.3.7.1.2. Objective Tree:

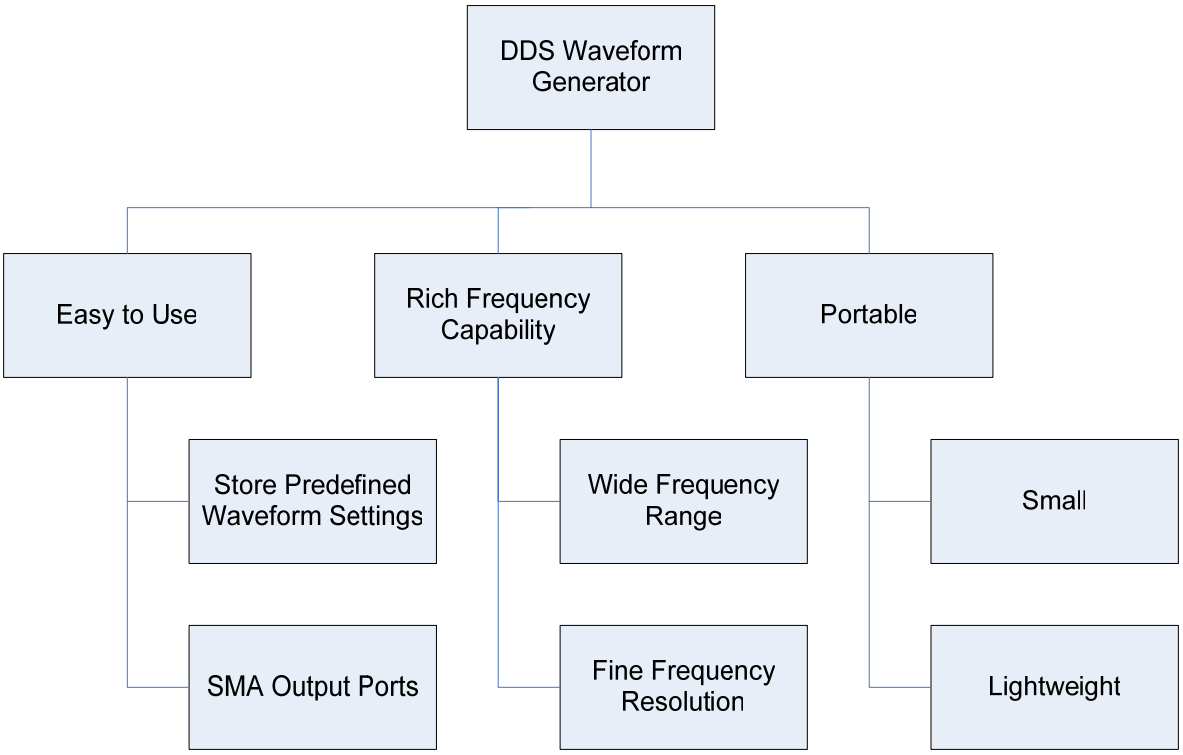

Figure 5: Objective trees for the DDS waveform generator

1.3.7.1.3. Ranked Needs:

	Easy to Use	Rich Frequency Capability	Portable	Overall Score
Easy to Use	-	0	1	1
Rich Frequency Capability	1	-	1	2
Portable	0	0	-	0

Table 1: Pairwise Comparisons Matrix – General

Rationale: The rich frequency capabilities has priority since it was requested by the company sponsors

	Store Predefined Waveform Settings	SMA Output Ports	Overall Score
Store Predefined Waveform Settings	-	0	0
SMA Output Ports	1	-	1

Table 2: Pairwise Comparison – Ease of Use

Rationale: The upconverter used in ARTEMIS's test setup all use SMA connections. The SMA ports connection is given priority so that conversion adapters are not needed.

	Wide Range	Fine Resolution	Overall Score
Wide Range	-	0	0
Fine Resolution	1	-	1

Table 3: Pairwise Comparison – Frequency Capability

Rationale: Fine Resolution is give priority since small frequency resolution is need for their test setup.

	Small	Lightweight	Overall Score
Small	-	0	0
Lightweight	1	-	1

Table 4: Pairwise Comparison - Portability

Rationale: The main object of this chart is to show that a lightweight device would be more desirable. Since this device will be moved around ARTEMIS, they requested the device be lightweight.

2. Requirements Specifications

2.1. The Requirements

Marketing Requirements	Engineering Requirements	Rationale
Have an easy to use interface	All output signals must be available on standard SMA female connectors	This allows the user to easily connect to the device, set and recall frequently used profiles
	Four profiles will be available to set and select frequency configurations	
	The user must be able to interactively select a Frequency value and store this value into one of the user profiles	
	The device will provide a quick and simple means to select the frequency profile	
Be small, portable and lightweight	The device will operate off a standard 120 volt AC wall outlet	Make the device portable and easier to move around
	The generator should fit into a 4 inch x 12 inch x 12 inch volume	
Have fine frequency resolution. Be able to generate a wide range of output frequencies.	The output frequency must be able to reach all multiples of 100 Hz between 0 and 300 MHz.	This will allow ARTEMIS to test the upconverters more throe
	The device must employ the DDS method of waveform synthesis	

Table 5: Marketing Requirements to Engineering Requirements Mapping

2.1.1. Engineering Requirements:

2.1.1.1. Engineering Requirement #: 1

Engineering Requirement #: **1**

The Requirement (list requirement below)

The device must employ the DDS method of waveform synthesis

Requirement Type: ☐ ideal ☒ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): **5**

Impacted Marketing requirements (list numbers): **2, 5**

Category (check all that apply):

☐ adaptability	☒ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☒ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☐ usability
☐ ethical	☒ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This requirement was specified by our company sponsor.

Testability/Verification. Describe the process your team will use to verify the requirement.

No testing is required. Either we will have employed the DDS method or not.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.2. Engineering Requirement #: 2

Engineering Requirement #: 2

The Requirement (list requirement below)

All output signals must be available on standard SMA female connectors.

Requirement Type: ☐ ideal ☒ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 5

Impacted Marketing requirements (list numbers): 3

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☒ solution
☐ energy	☐ manufacturability	☒ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

The usage of SMA connectors was specified by the company sponsor.

Testability/Verification. Describe the process your team will use to verify the requirement.

Testing will be straightforward. It should be clear by inspection, whether SMA connectors are used or not. However, the design will be connected to the corresponding connectors used by Artemis to ensure that the appropriate connectors were chosen.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.3. Engineering Requirement #: 3

Engineering Requirement #: 3

The Requirement (list requirement below)

Four profiles will be available to set and select frequency configurations.

Requirement Type: ☐ ideal ☒ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): **5**

Impacted Marketing requirements (list numbers): **6,4**

Category (check all that apply):

☒ adaptability	☒ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

The four profiles will allow the user to easily switch between defined frequencies.

Testability/Verification. Describe the process your team will use to verify the requirement.

Each profile will be tested by setting frequencies to each profile and check its frequency using a spectrum analyzer.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.4. Engineering Requirement #: 4

Engineering Requirement #: 4

The Requirement (list requirement below)

The frequency range should be from 0 MHz to 300 MHz

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 4

Impacted Marketing requirements (list numbers): 2

Category (check all that apply):

☐ adaptability	☒ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☒ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☐ usability
☐ ethical	☒ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

ARTEMIS needs the defined frequency range requirement for their upconverter test setup.

Testability/Verification. Describe the process your team will use to verify the requirement.

The systems frequency output will be hooked up to an oscilloscope or a spectrum analyzer. The generator will then be stepped through each frequency to verify its frequency range.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirement established on 10-23-2006. Requirement changed on 11-16-2006 the maximum frequency was mistakenly typed incorrectly. Its previous max frequency was 400 MHz. The actual max frequency is 300 MHz.

2.1.1.5. Engineering Requirement #: 5

Engineering Requirement #: 5

The Requirement (list requirement below)

The user must be able to interactively select a frequency value and store this value into one of the user profiles.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): **4**

Impacted Marketing requirements (list numbers): **6, 4**

Category (check all that apply):

☐ adaptability	☒ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☒ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This feature will allow commonly used frequencies to be stored and quickly accessed again later. It will allow ARTEMIS's upconverter tests to be executed quickly and efficiently.

Testability/Verification. Describe the process your team will use to verify the requirement.

This requirement can be tested by selecting and storing test frequencies into different user profiles. Each profile will have a value stored during the test and the test frequencies used will include boundary values (0 Hz and 300 MHz) as well as the frequencies most commonly used by ARTEMIS.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.6. Engineering Requirement #: 6

Engineering Requirement #: 6

The Requirement (list requirement below)

The output spectrum should have a spurious free dynamic range (SFDR) below 40 decibels relative to one milliwatt (dBm).

Requirement Type: ☐ ideal ☒ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 4

Impacted Marketing requirements (list numbers): 2

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☐ usability
☐ ethical	☒ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

The output signal should be relatively clean. A noisy signal could cause spurs in the upconverter causing bad test results.

Testability/Verification. Describe the process your team will use to verify the requirement.

The SMA (SubMiniature version A) output connector will be hooked up to a spectrum analyzer using a 50 ohm SMA cable. The Spectrum analyzer will be setup to view the entire frequency span (0 MHz - 400 MHz). The highest spur must be 40 dBm down from the highest peak.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.7. Engineering Requirement #: 7

Engineering Requirement #: 7

The Requirement (list requirement below)

The device will operate off a standard 120 volt AC wall outlet.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 4

Impacted Marketing requirements (list numbers): 6

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☒ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☒ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This device will be able to run anywhere there is a wall outlet.

Testability/Verification. Describe the process your team will use to verify the requirement.

All of the system operation tests will be performed using this type of power source.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006. The requirment was changed on 11-16-2006. The previous requirement powed the device with a 5-12 Volt power supply. Dr. Coulston suggested using a standare wall outlet.

2.1.1.8. Engineering Requirement #: 8

Engineering Requirement #: 8

The Requirement (list requirement below)

The output frequency must be able to reach all multiples of 100 Hz between 0 and 300 MHz.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 3

Impacted Marketing requirements (list numbers): 5

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☒ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☐ usability
☐ ethical	☒ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This frequency resolution is needed so that each frequency level in the upconverter can be inspected. If the digital - to - analog converter's reference clock is fast enough and the sin-wave table has enough resolution then this output resolution is feasible.

Testability/Verification. Describe the process your team will use to verify the requirement.

The systems frequency output will be hooked up to an oscilloscope or an spectum analyser. The generator will then be stepped through each frequency to verify it's frequency resolution.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006. Changed on 11-16-2006 the max frequency was also incorrect. The previous values was 400 MHz.

2.1.1.9. Engineering Requirement #: 9

Engineering Requirement #: 9

The Requirement (list requirement below)

The device will provide a quick and simple means (such as one push button for each profile) to select a frequency profile.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): **3**

Impacted Marketing requirements (list numbers): **4, 6**

Category (check all that apply):

☐ adaptability	☒ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☒ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

The act of selecting a output profile must be less time consuming and less complex than applying the waveform settings directly. If it is not, then there is not much point to storing profile settings in the first place.

Testability/Verification. Describe the process your team will use to verify the requirement.

Selecting a stored profile should take less than 1/3 the time it takes to apply the same settings manually.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.10. Engineering Requirement #: 10

Engineering Requirement #: 10

The Requirement (list requirement below)

The device should display the frequency value stored within each profile.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 3

Impacted Marketing requirements (list numbers): 6

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☒ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This feature allows the user to know the important information about each profile without first having to select that profile. It also allows the user to easily locate which profile is desired to be selected or overwritten.

Testability/Verification. Describe the process your team will use to verify the requirement.

This requirement can be tested by observation.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.11. Engineering Requirement #: 11

Engineering Requirement #: 11

The Requirement (list requirement below)

The device's output power level must be between +10 dBm and -10 dBm. This is the voltage terminated into a 50 ohm load.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): **3**

Impacted Marketing requirements (list numbers): **6,4**

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☐ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☒ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☐ usability
☐ ethical	☒ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

The upconverter's input can only handle so much input power.

Testability/Verification. Describe the process your team will use to verify the requirement.

The frequency output can be hooked up to a power meter or an spectrum analyzer. If using the spectrum analyzer the highest peak measurement will display the current power level.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.1.1.12. Engineering Requirement #: 12

Engineering Requirement #: 12

The Requirement (list requirement below)

The generator should fit into a 4 inch x 12 inch x 12 inch volume.

Requirement Type: ☒ ideal ☐ constraint ☐ non-verifiable

Importance (1 = lowest, 5=highest): 2

Impacted Marketing requirements (list numbers): 1

Category (check all that apply):

☐ adaptability	☐ functionality	☐ political
☐ availability	☐ health	☐ reliability
☐ budget (development)	☐ legal	☐ safety
☐ documentation	☒ look and feel	☐ social
☐ economic (sales cost)	☐ maintainability	☐ solution
☐ energy	☐ manufacturability	☐ standards
☐ environmental (impact on)	☐ operational (constraint of physical environment)	☐ sustainability/re-usability
☐ ergonomics	☐ other _____	☒ usability
☐ ethical	☐ performance	

Justification. Describe the rationale for this requirement. Why is this requirement needed? Why is it reasonable to expect it can be achieved?

This size was chosen because it will easily fit onto test carts with the other equipment.

Testability/Verification. Describe the process your team will use to verify the requirement.

The outer dimensions of the complete design will be measured to verify this requirement.

History. If you change a requirement, indicate date of change, reason, and list the previous version of the requirement.

Requirment established on 10-23-2006.

2.2. Constraints:

2.2.1. Economic:

The price of out device should not exceed more then $500 dollars.

2.2.2. Environmental:

The unit should be able durable and be able to operate under heavy work conditions. The power consumption should also be kept minimal.

2.2.3. Manufacturability:

The unit must meet all ARTEMIS's specification and be easily manufactured.

2.2.4. Health and Safety:

The device should not be able to injure users or anyone around it.

2.2.5. Social

The device should be able to be controlled by a five year old.

2.3. Standards

2.3.1. Testing

The device final testing must meet all requirements propose by the project sponsor.

2.3.2. Communications

A JTAG interface should be implemented on the device for easy programming and boundary scans. The device should also have an easy to use interface for the users

2.3.3. Documentation

All the design phases should be documented and provided to the company sponsor.

3. Design

3.1. Design Alternatives:

3.1.1. Control Unit:

Method	Strengths	Weakness
FPGA	• Fast • Readily available • No lead time • Lots of I/O's • Number of gates • RAM • Cheap	• Multiple voltages • Complicated circuitry • Dynamic programming
Microcontroller	• Easy to program • Simple circuitry • Static programming • Cheap	• Slow • Limited I/O's • No RAM
FPGA w/ soft core	• Lots of I/O's • Number of gates • Easy to program	• Multiple voltages • Complicated circuitry • Availability • Long Lead • New market • Expensive
CPLD	• Fast • Lots of I/O's • Static programming • Cheap	• Limited gates • Complicated circuitry • No RAM

Table 6: Control Unit Strengths and Weakness

Rationale: We chose to use a FPGA mainly because of the speed, availability and price. The speed is very important in our design because The DDS chip of choice (AD9858) will be running at very fast clock rate. In order to synchronize all the data, we need a controller that can run at speeds up to 120 MHz (sysclk @ 960 MHz/8). This chip is also readily available because ARTEMIS currently has Spartan 3 FPGA's available in stock. ARTEMIS also offered to help with the circuitry design if this approach was chosen. We did not choose a microcontroller because it is very slow and has limited number of I/O pins. Currently, our design needs approximately 40 I/O which complicates the design when using a microcontroller. A soft core FPGA was also considered but seemed very complex and expensive. The final controller choice was the CPLD this path was not chosen due to the circuitry design complexity.

3.1.2. LCD:

Method	Strengths	Weaknesses
Graphical LCD	• Data displayed very clearly (more pixels/resolution) • Able to display plenty of data	• More complicated to control • More expensive
Character LCD	• Easy to control • Inexpensive	• Limited display capabilities • Data is less clearly displayed (less pixels/resolution)

Table 7: LCD strengths and weakness

Rationale: We chose the character LCD option due to its lower price. Although the graphical LCD would look better, we decided that the character LCD would be sufficient to display the necessary data and that cost was more important than aesthetics.

3.1.3. Direct Digital Synthesizer:

Method	Strengths	Weaknesses
AD9858 DDS Chip	• High frequency range: up to 400+ MHz • Sharp frequency resolution • Built in frequency store/recall capabilities • Readily available	• Somewhat expensive • Requires a controller that can operate at ~120 MHz • Difficult to prototype (requires PCB with many solder mount connections)
FPGA	• Our team is familiar with FPGA design • Could feasibly be used for control as well as DDS function	• Difficult to prototype (requires PCB with many solder mount connections) • Will be difficult to achieve needed frequency range and resolution
FPGA with SoftCore	• Some softcore architectures are Open Source (free to use)	• Our team is unfamiliar with designing for a softcore system

	• Higher level system design in C	• Same weaknesses as FPGA above
Microcontroller	• Our team is familiar with microcontroller design • Easy to prototype	• Pretty much impossible to achieve necessary frequency range and resolution

Table 8: DDS strengths and weakness

Rationale: AD9858 was chosen because it will enable us to meet our frequency range and resolution requirements easily. The frequency requirements are critical for this project and it will be reassuring to use an off-the-shelf component that already meets the requirements. This DDS chip has been tested to a degree that we could never hope to match with a self-designed DDS unit, so we will be much more confident in the performance of our overall system than we would be otherwise. This design option will also allow us to put more time into designing the other aspects of the device such as the anti-aliasing filter, control unit, and user interface.

3.2. Level 0:

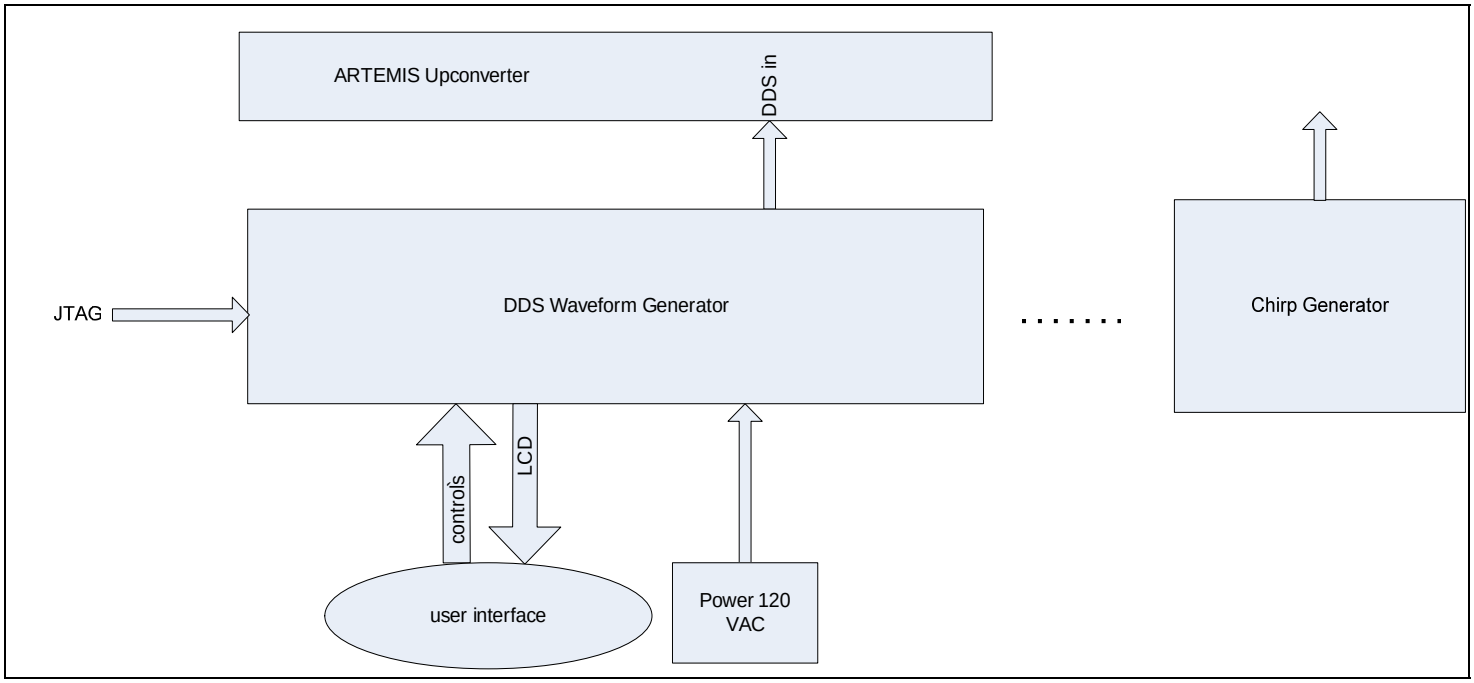

Figure 6: Level 0 Diagram

Module	DDS Waveform Generator
Inputs	-Power: 120 volts AC rms, 60 Hz -Control Buttons: 9 digital signals -JTAG Connection: Includes VCC, GND, and 4 digital signals named TMS, TDI, TDO, and TCK. See the background section for more information on the JTAG interface.
Output	-DDS sinusoid wave at the selected frequency. Range of frequencies is 0 Hz – 400 MHz. Nominal output power is 1 mW with a range of $\pm$ 10 dbm (10mW – 0.1 mW) -LCD output to the user.
Functionality	Generates a sinusoid waveform with a user-selectable frequency between 0 Hz and 300 MHz.

Table 9: Level 0 DDS Signal Generator

3.3. Level 1:

Module	Analog Devices AD9858 DDS chip
Inputs	-V_{dd}: 3.3 Volts DC -REFCLK: Reference Clock Input 50% duty cycle @ 1 GHz max +5 dBm -ADDR5- ADDR0: 6-bit Address Select for accessing onboard chip Registers (TTL signal) -D7-D0: Parallel Port Data (TTL Signal) -FUD: Frequency Update. (TTL Signal) -WR: Active low write pulse. (TTL Signal) -PS0-PS1: Used to select one of 4 profiles. (TTL Signal)
Output	-synclk: clock divider (ref clock/8) supplies the FPGA clock -iout: Digital to Analog converter output. 0 dBm
Functionality	Direct Digital Synthesizer capable of generating an analog output sine wave up to 400+ MHz. Digitally programmable with a 32-bit frequency tuning word allowing fine tuning resolution.

Table 10: Level 1 AD9858 DDS chip

Module	Power Supply
Inputs	-120 Volts AC rms, 60 Hz
Output	-Power: +3.3V DC, DDS analog power (600 mA) -Power: +3.3V DC, DDS digital power and FPGA I/O bank power -Power: +2.5V DC @ 7 mA, FPGA JTAG programming -Power +1.2V DC @ 5 mA, FPGA core logic power
Functionality	Convert AC wall outlet voltage to a positive DC output voltages and to provide enough current to drive all the chips

Table 11: Level 1 Power Supply

Module	Anti-Aliasing Filter
Inputs	-0 dBm analog output sine wave with aliases
Output	-DDS out: -3 dBm analog output sine wave with-out aliases
Functionality	Removes all the aliases created during synthesis.

Table 12: Level 1 Anti-Aliasing Filter

Module	Control Buttons
Inputs	-GND
Output	-9-bit digital signal. The output of an individual bit will equal GND when the respective button is pressed, and will be in a high-impedance state when the button is not pressed. Pull-up resistor circuitry within the FPGA will interpret GND as logic 0 and high-impedance as logic 1.
Functionality	Provides user input to select the frequency of the output.

Table 13: Level 1 Control Buttons

Module	LCD
Inputs	-3.3V Power -RS: Register Select -R/W: Read or Write Data -E: Enable communication with LCD -D7-D0: Data bits
Output	-LCD Screen: displays frequency of the output wave, which digit is selected, which profile is selected and the profile frequency. -4 rows x 16 columns of characters
Functionality	Provides output to the user displaying frequency information for current output and selected profile.

Table 14: Level 1 LCD

Module	FPGA
Inputs	-Power: 3.3V DC and 1.2V DC -Control Buttons signal: 9 bits -Program: 5-bit connection to the Flash EEPROM that is used to load our program into the FPGA on power-up. -synclk: clock signal generated by the AD9858 DDS chip.
Output	-Data and Control signals to the LCD -RS: Register Select -R/W: Read or Write Data -E: Enable communication with LCD -D7-D0: Data bits -Outputs to AD9858 DDS chip -ADDR5- ADDR0: 6-bit Address Select for accessing onboard chip Registers (TTL signal) -D7-D0: Parallel Port Data (TTL Signal) -FUD: Frequency Update. (TTL Signal) -WR: Active low write pulse. (TTL Signal) -PS0-PS1: Used to select one of 4 profiles. (TTL Signal)
Functionality	Controls inputs and outputs for the user interface and governs the frequency setting of the AD9858 DDS chip.

Table 15: Level 1 FPGA [5]

3.4. Level 2:

3.4.1. System Flow Chart :

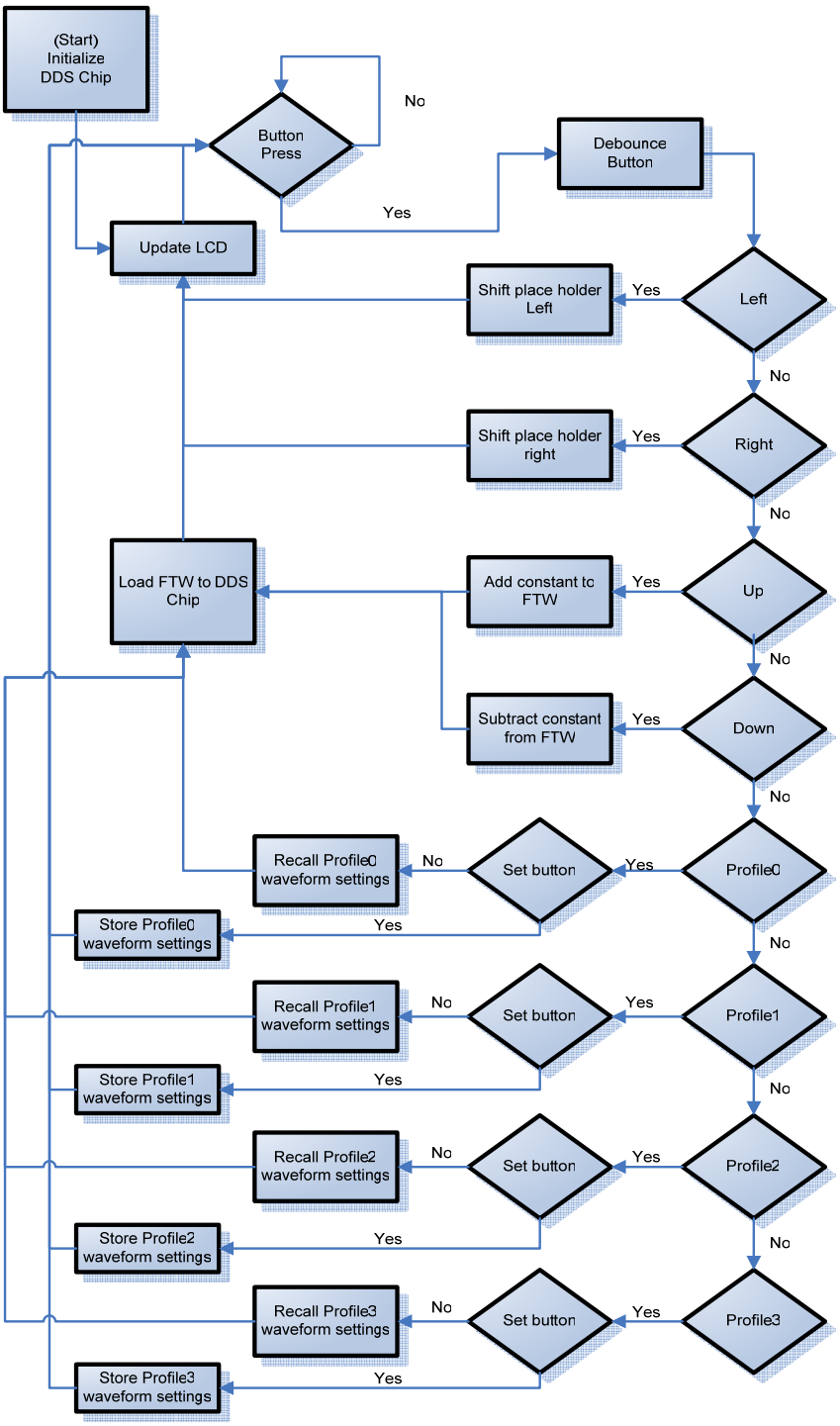

Figure 7: Level 2 System Flow Chart

Flow Chart Description

Initialize DDS Chip: Loads the AD9858 chip internal register bank with initial values to put the chip into single tone mode with the system reference clock divide by 2 function disabled.

Update LCD: This initializes the LCD and updates the LCD each time the frequency is changed.

Button Press: During each interrupt this process checks for a new button press.

Debounce: Each button is debounced to remove any button noise. Code snippets are shown in Appendix B.

Left: Checks if the left button was pressed.

Shift place holder left: Moves the LCD cursor to the left and select a new frequency constant.

Right: Checks if the right button was pressed.

Shift place holder right: Moves the LCD cursor to the right and select a new frequency constant.

Up: Checks if the up button was pressed.

Add constant to FTW: The add/sub adds a frequency constant to its current value.

Down: Checks if the down button was pressed.

Subtract constant to FTW: The add/sub subtracts a frequency constant from its current value.

Profile 0: Checks if the P1 button was pressed.

Set: Stores the current frequency into its profile register.

Recall Profile0 Waveform Setting: Loads profile 1 settings into its corresponding registers.

Store Profile0 Waveform Setting: Stores profile 1 settings into its corresponding registers.

Profile 1: Checks if the P2 button was pressed.

Recall Profile1 Waveform Setting: Loads profile 2 settings into its corresponding registers.

Store Profile1 Waveform Setting: Stores profile 2 settings into its corresponding registers.

Profile 2: Checks if the P3 button was pressed.

Recall Profile2 Waveform Setting: Loads profile 3 settings into its corresponding registers.

Store Profile2 Waveform Setting: Stores profile 3 settings into its corresponding registers.

Profile 3: Checks if the P4 button was pressed.

Recall Profile3 Waveform Setting: Loads profile 4 settings into its corresponding registers.

Store Profile3 Waveform Setting: Stores profile 4 settings into its corresponding registers.

Load FTW to DDS Chip: Writes the FTW values from the add/sub into the AD9858 internal register bank and triggers and the chip to load in the new frequency.

3.4.2. Finite State Machines:

3.4.2.1. User Interface:

Three Finite State Machines were used to control multiple interfaces. Figure 8 shows the FMS used to control the user button interface. On power up the FSM first transitions in to the reset state and then unconditionally transitions into the idle state where it waits for a user input. As the user interacts with the external button interface the FSM transition into its different states controlling internal operations.

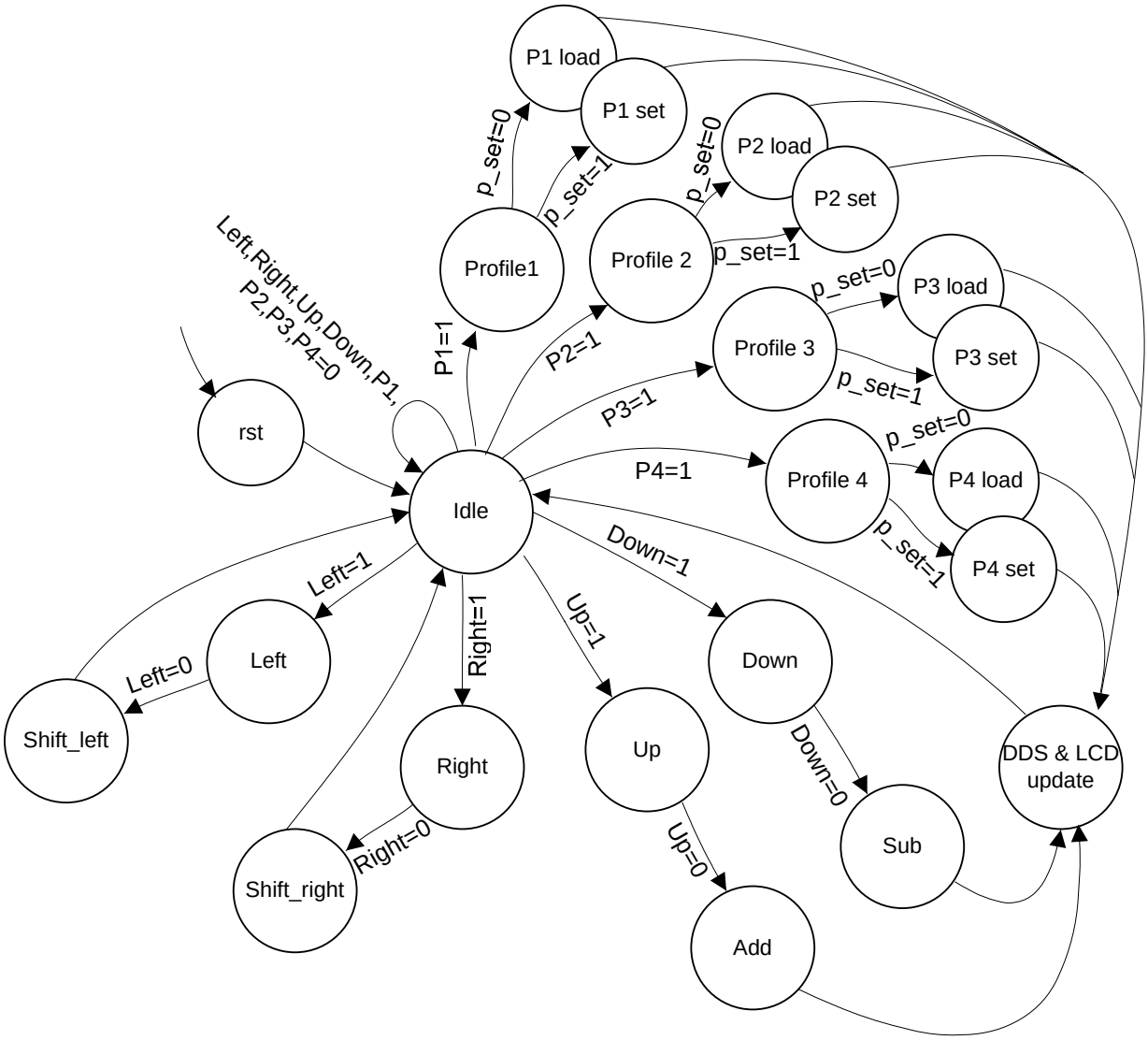

Figure 8: Level 2 Button FSM

3.4.2.2. Liquid Crystal Display:

Figure 9 displays the LCD FSM. This initializes the LCD on power up and transitions between two different states after initialization. The first control loop, lcd_write updates one digit on the LCD as the user increment and decrements its current place holder. The second loop p_lcd_write re-writes the entire LCD contents during a profile load.

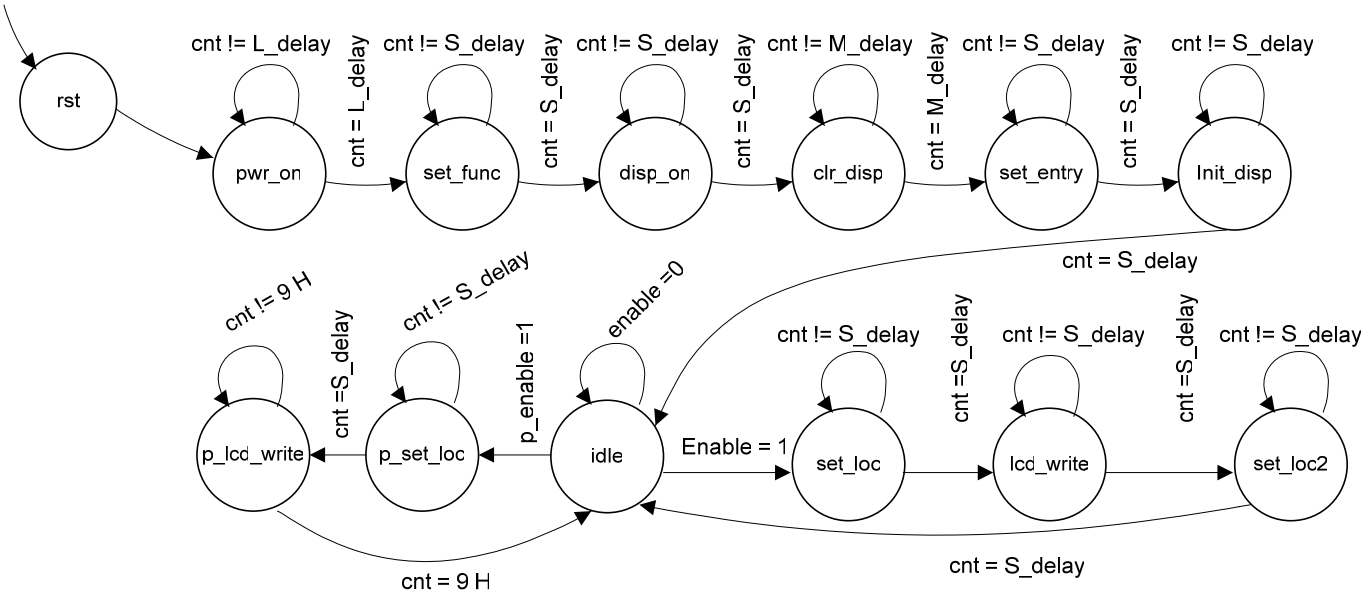

Figure 9: Level 2 LCD FSM

3.4.2.3. Direct Digital Synthesizer

The final FSM shown in figure 10 is the AD9858 control. It controls the initialization and the frequency control of the DDS chip. Each time the frequency is changed; the FSM loads a new Frequency Tuning Word into the AD9858 internal register bank and triggers the chip to load the new frequency. Code snippets of this FSM are shown in appendix B. The output frequency is calculated using the following formula.

For a desired output frequency (Fo) and sampling rate (REFCLK=960 MHz), the frequency tuning word (FTW) is calculated according to the following equation. N=32 bit

$$\mathbf{FTW} = \frac{\mathbf{Fo} \times 2^{\mathbf{N}}}{\mathbf{REFCLK}}$$

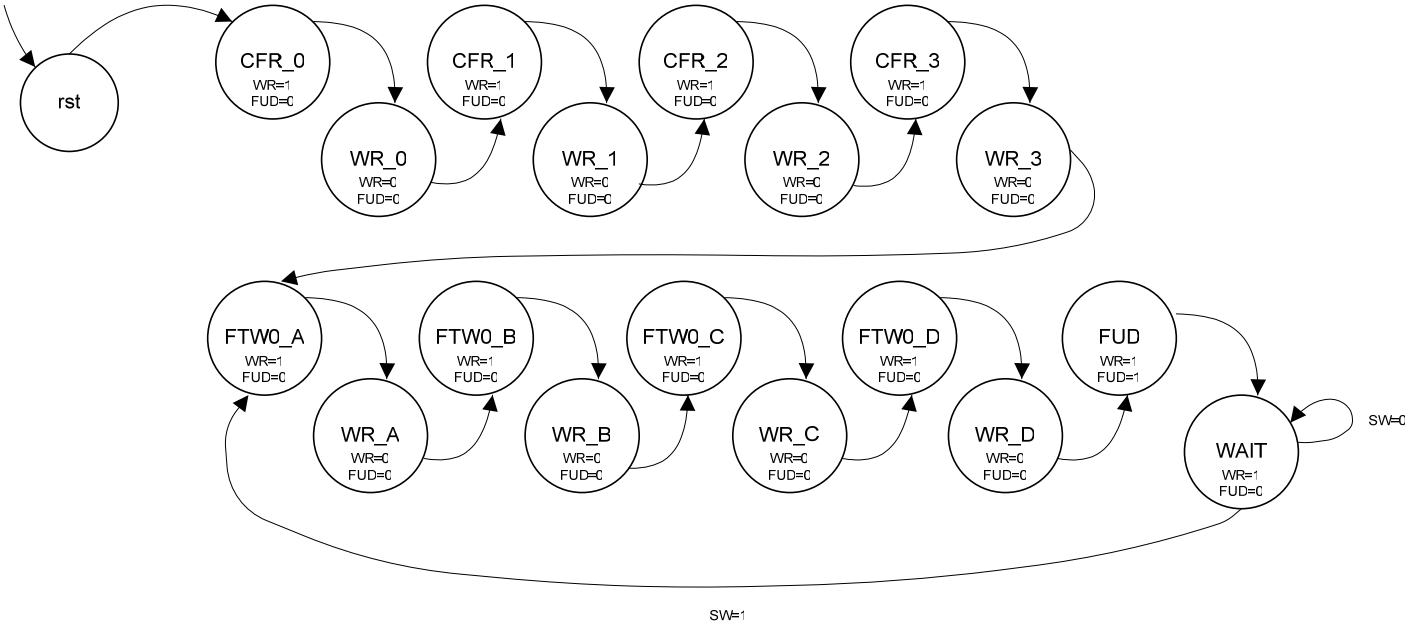

Figure 10: Level 2 AD9858 FSM

3.4.3. Low Pass Filter:

A Low pass filter is needed to reconstruct the output waveform and to remove any aliases created during the digital synthesis. A Seven Section ChebyShev filter was designed and simulated in Microwave office. The seven sections are derived from total number of passive element (4 - capacitors + 3 - inductors) shown in the circuit diagram Figure 11. The passband of the filter ranges from DC to 400 MHz and with a reject band from 400 MHz and up. Simulated frequency response of the filter is shown in Figure 12.

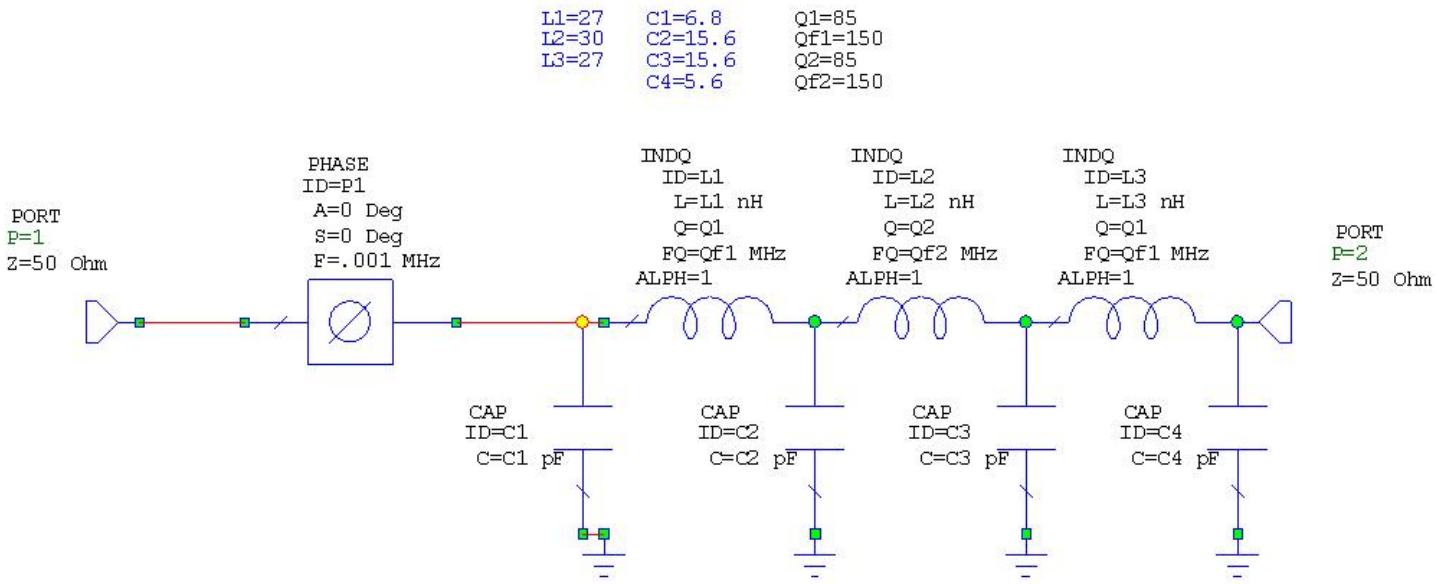

Figure 11: Level 2 LPF Circuitry

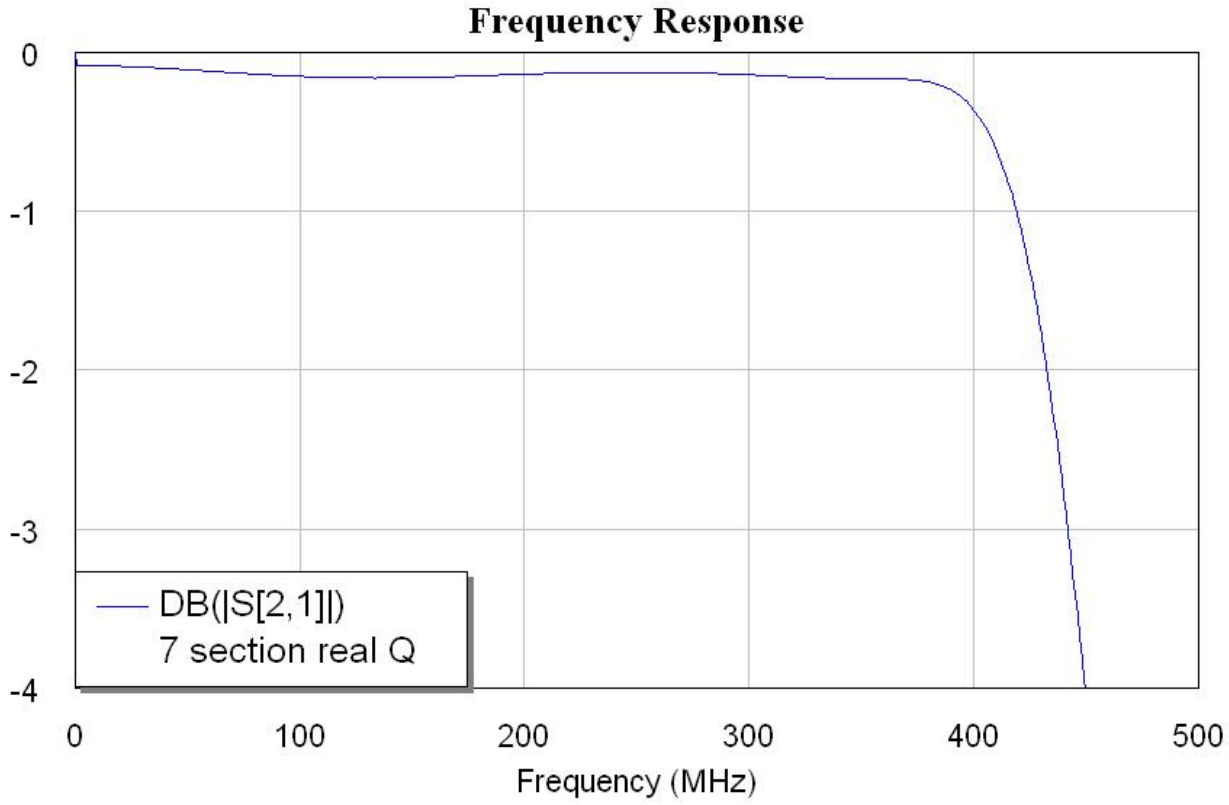

Figure 12: Level 2 LPF Simulations

3.4.4. Enclosure:

The housing was designed using AutoCAD 2004 and its layout was designed after a desk drawer. The enclosure has four piece of aluminum with slots down each side (Figure 14). The circuit board slide into each slot so that no fasteners are needed to hold the PCB into place. This reduces the number of fasteners and simplifies the fabrication.

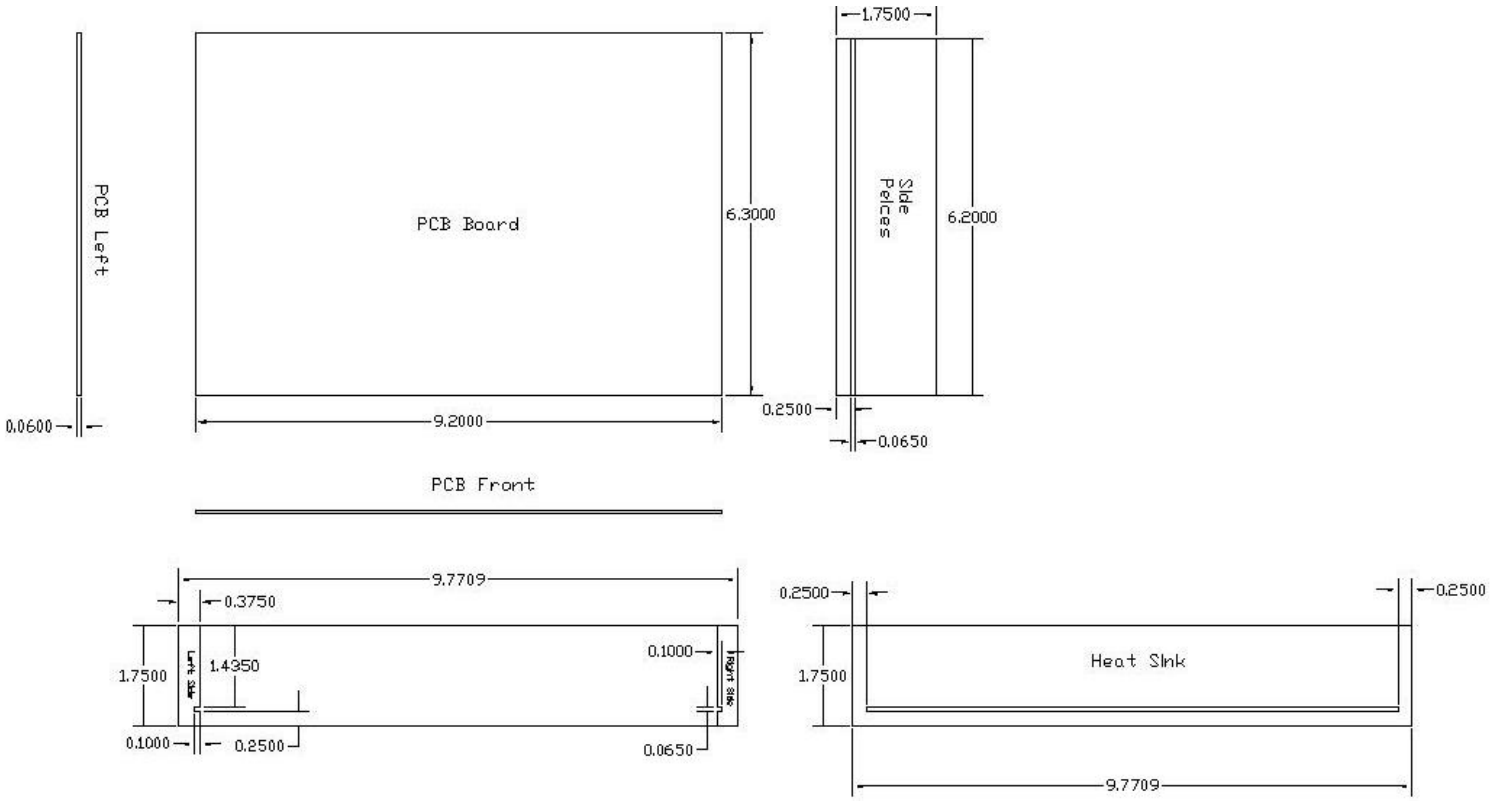

Figure 13: Level 2 Enclosure

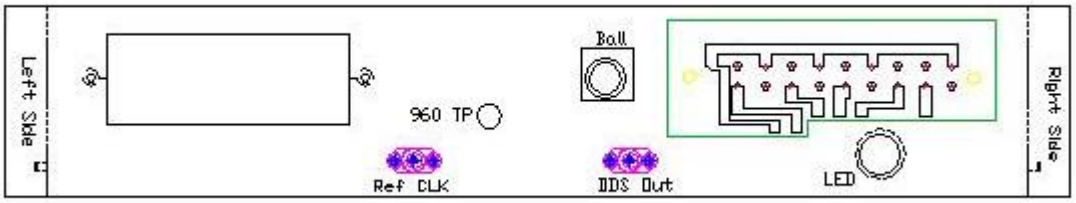

Figure 14: Level 2 Front Panel

3.4.5. Circuitry:

The following two figures show the FPGA and DDS circuitry designed in OrCAD.

3.4.5.1. AD9858:

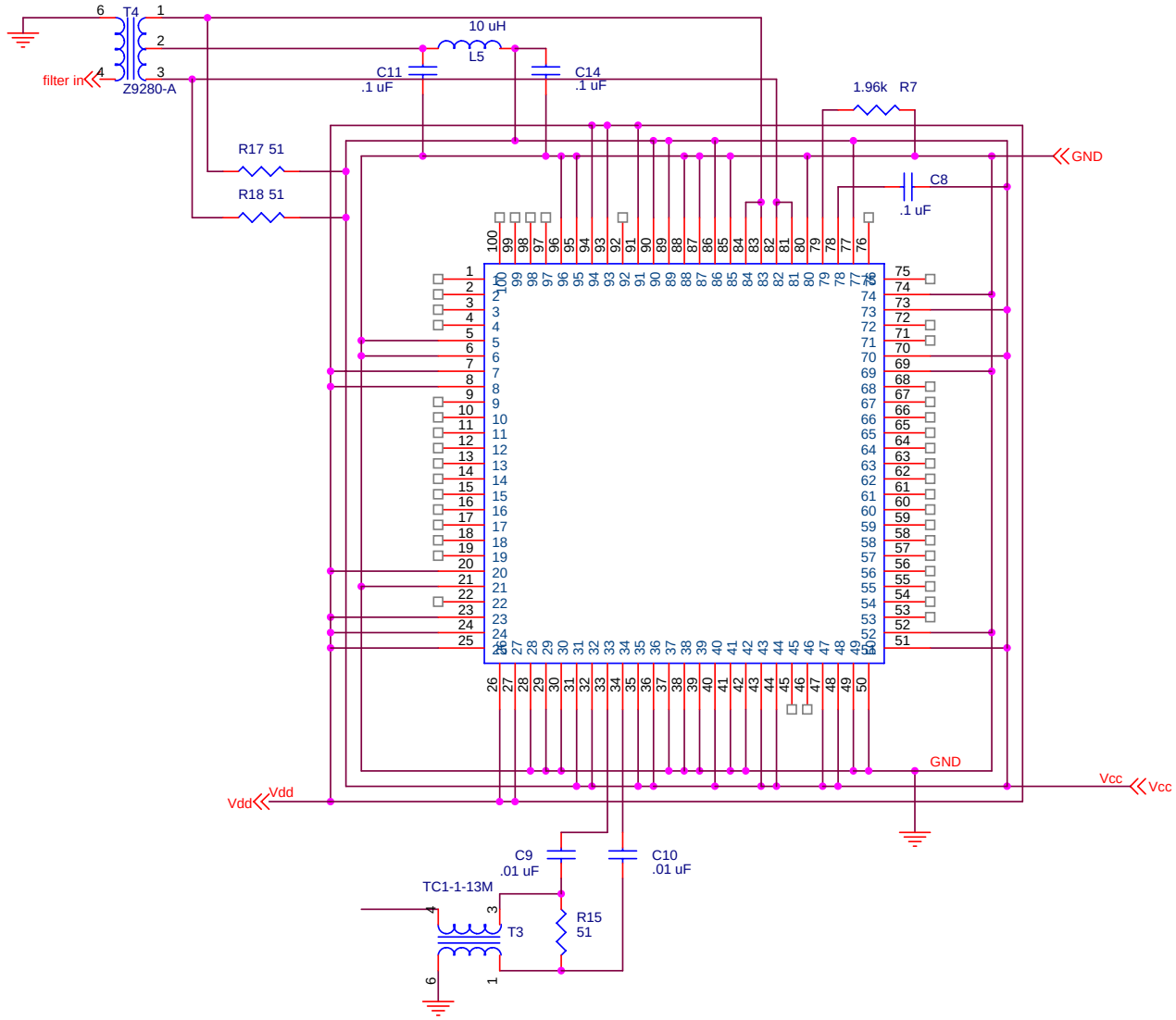

Figure 15: Level 2 AD9858 Circuitry

3.4.5.2. Spartan 3 FPGA:

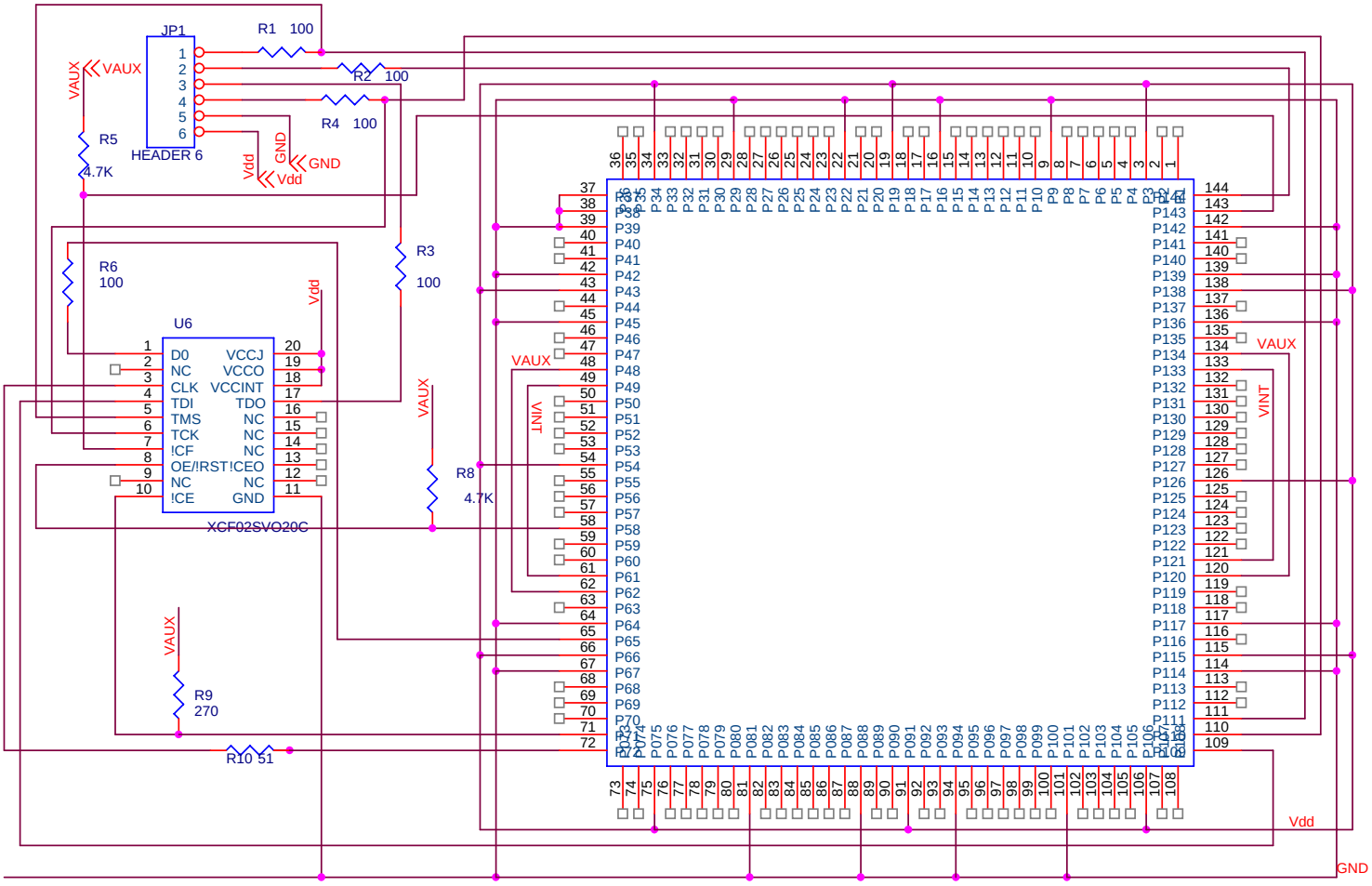

Figure 16: Level 2 Spartan 3 Circuitry

3.4.5.3. Power Supply:

A 12 Volt (1 amp) AC to DC transformer was used as the main voltage source. The 12 volts was then regulated down using voltage regulators to supply the 1.2, 2.5, 2 - 3.3 voltages (Figure 18).

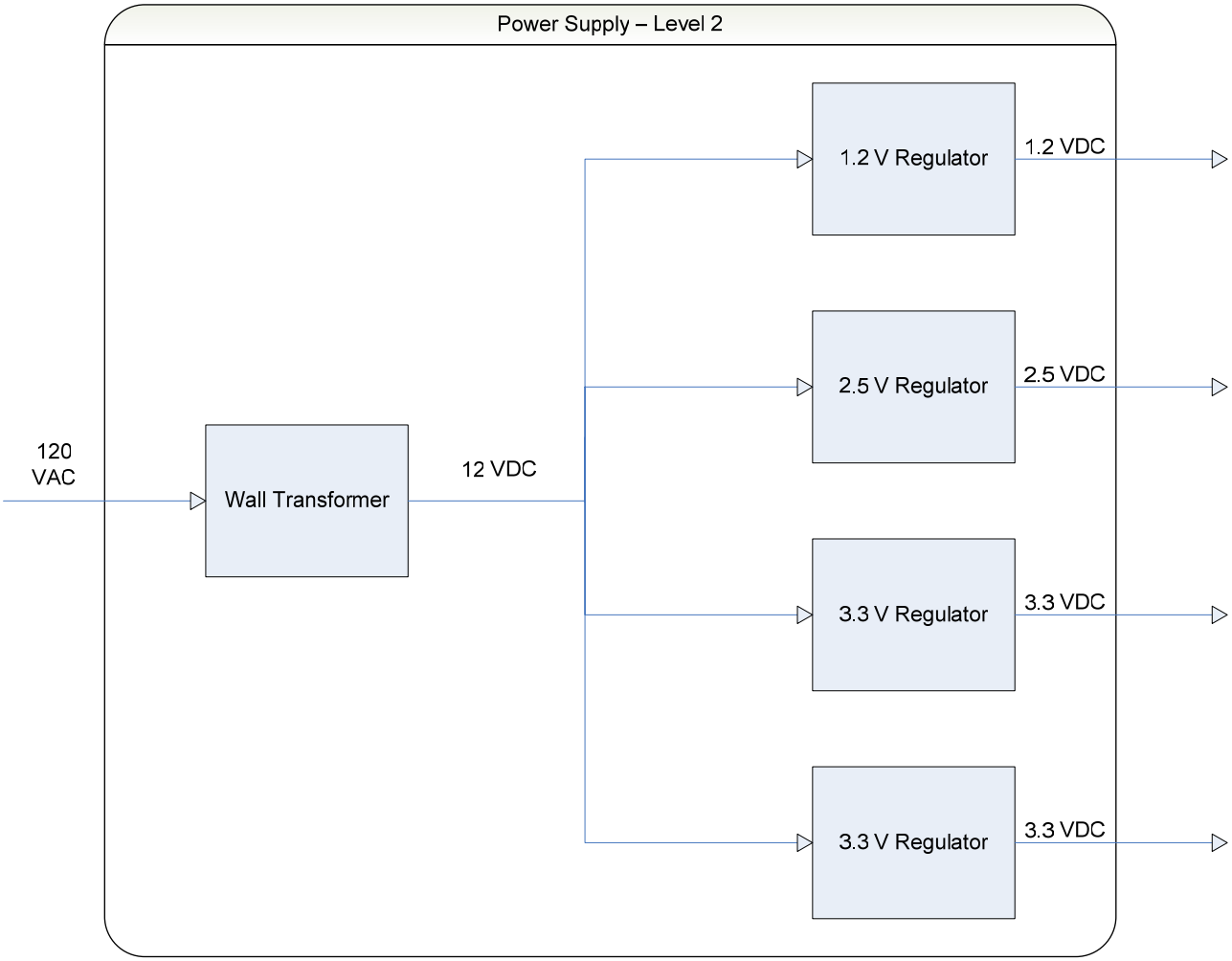

Figure 17: Level 2 Power Supply Design

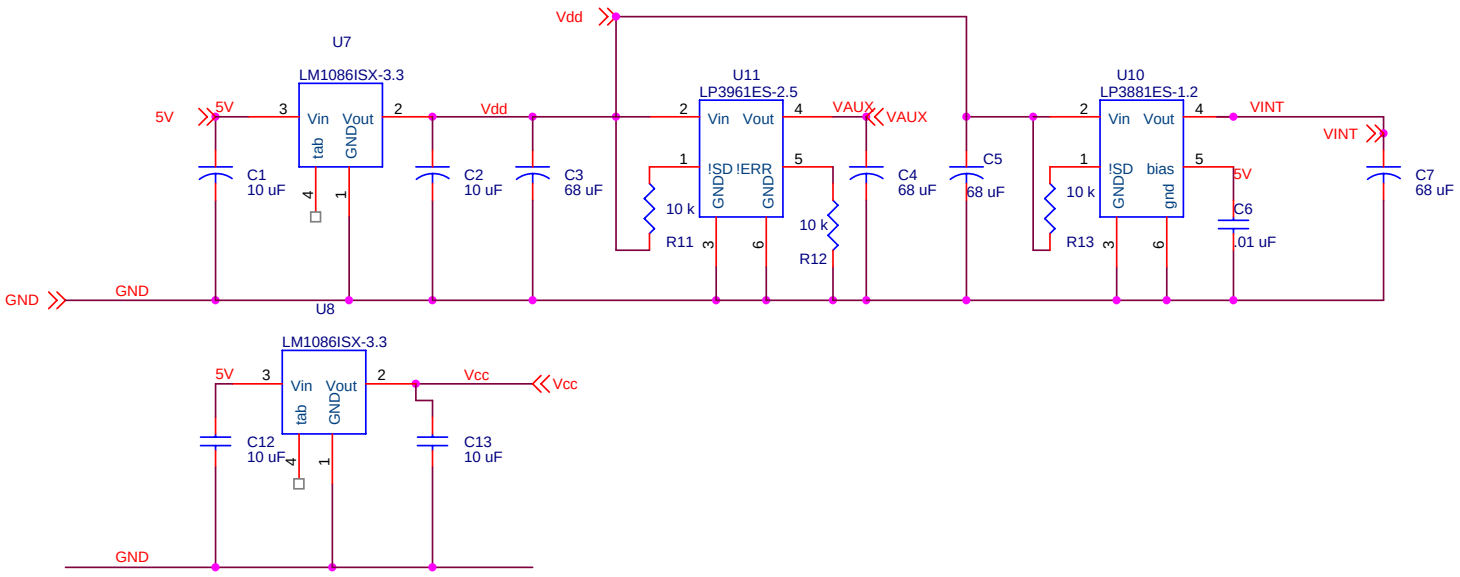

Figure 18: Level 2 Vregs Circuitry

3.4.5.4. User Interface Circuitry:

The user controls will be implemented using one track ball. The track ball gives the user four different directions for control. The +X direction was used for incrementing the current place holder while the –X direction was used for decrementing the current place holder. The incrementing and decrementing was implemented in the FPGA using a carry look ahead adder (CLA). The –Y direction allowed the user to advance the current place holder to the next higher position. The +Y direction shifted the place holder down. This behavior was implemented in the FPGA using a shift register. The track ball circuitry is made up of a series of switches that latches up when scrolled. Each switch was connected to a series of pin on the FPGA (Figure 19).

P1-P4 is the Profile Select. The track ball will but used to scroll the curser and increment and decrement the frequency. The trackball's center select will be used to set profiles.

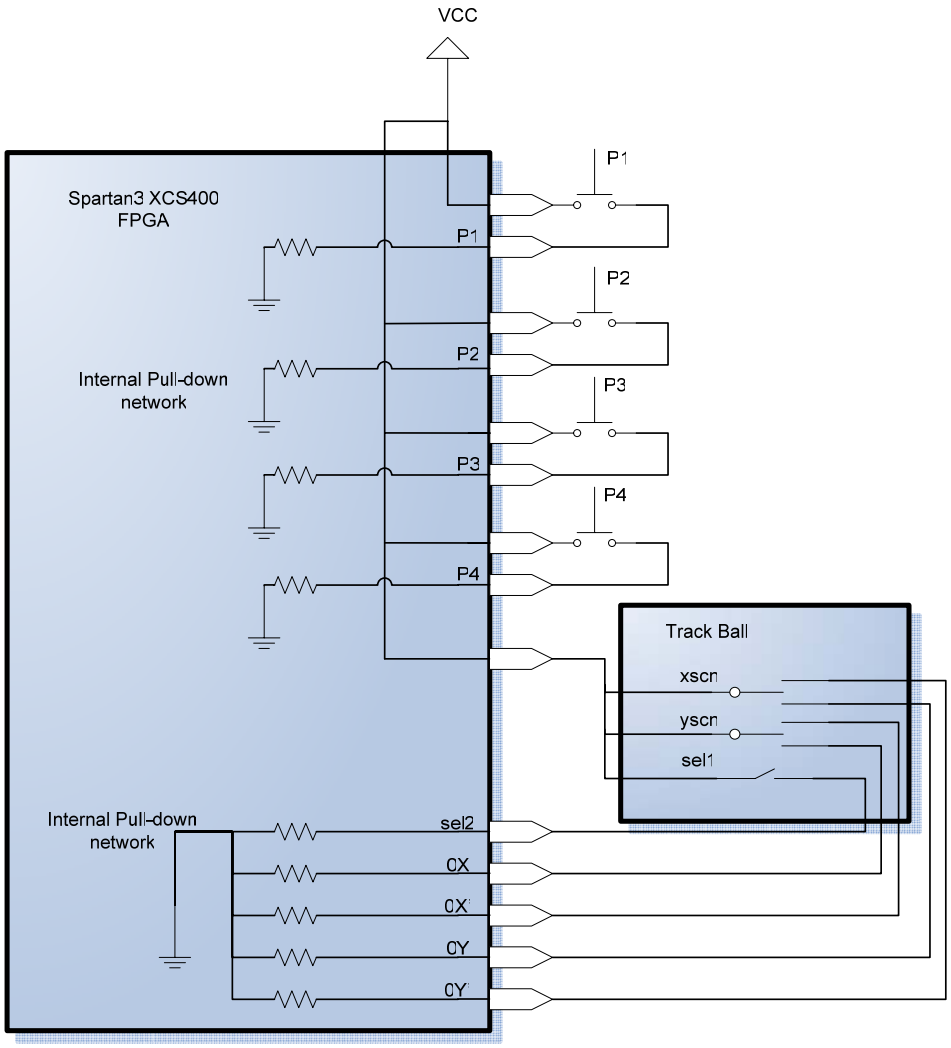

Figure 19: User interface

3.4.6. Phase Lock Oscillator:

A Phase Lock Oscillator (PLO) was used for the 960 MHz reference clock. A resonator, Peregrine Phase Lock Loop (PPL) chip, 8 MHz clock and some circuitry synthesizes the 960 MHz clock source (Figure 20). The 8 MHz Wenzel was used as a reference to lock the Oscillator. ARTEMIS provided us with the PLO.

.

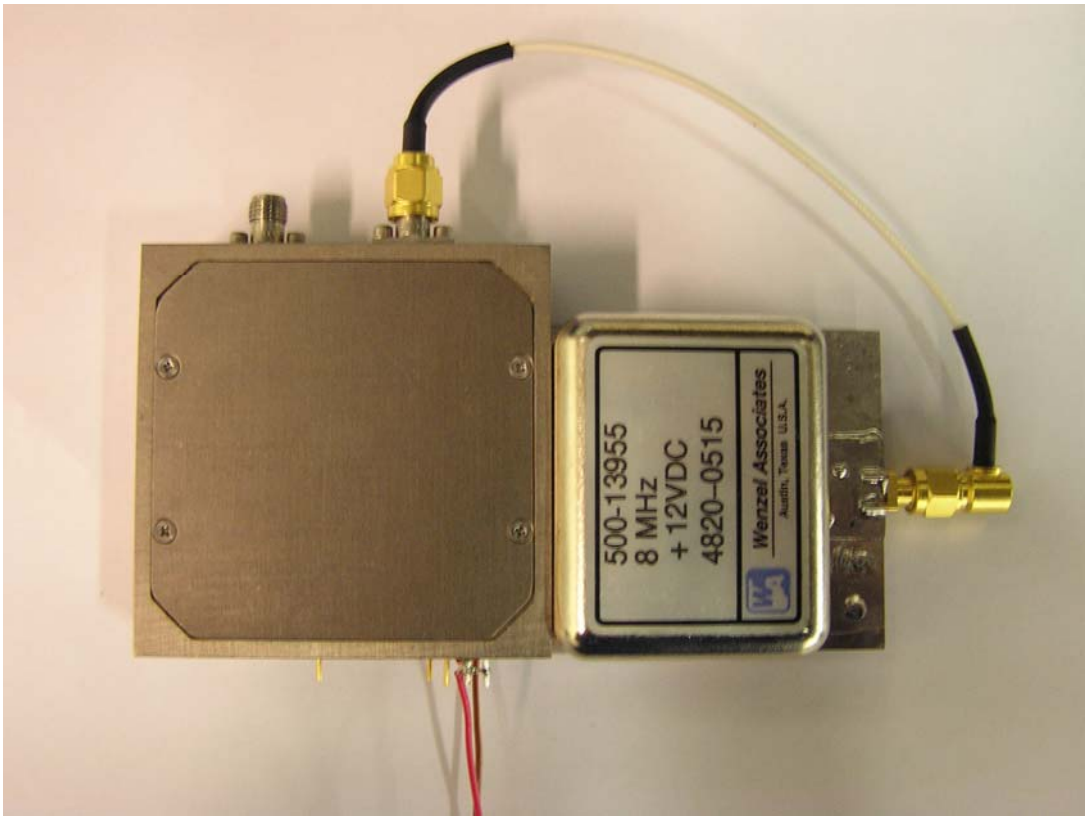

Figure 20: ARTEMIS Phase Lock Oscillator

4. Design Verification

4.1. Test Results:

4.1.1. Low Pass filter:

The filter was built inside a custom made test fixture (machined by ARTEMIS) with 2 SMA connectors inter-facing the filter's circuitry. The circuitry was designed using AUTOCAD 2004 and was cut using a T-Tek milling machine. The circuit board was then soldered into its fixture. Each capacitor and inductor was hand soldered under a microscope using tweeters. This filter's frequency response and insertion loss was then measured (Figure 22) using a Rohde&Schwarz Network Analyzer. The results showed a pass-band from 0 to 400 MHz and a stop-band beyond 400 MHz. The insertion loss (attenuation / ripple) was less than -2.3 dB. Figure 21 shows a picture of the housing and its circuitry.

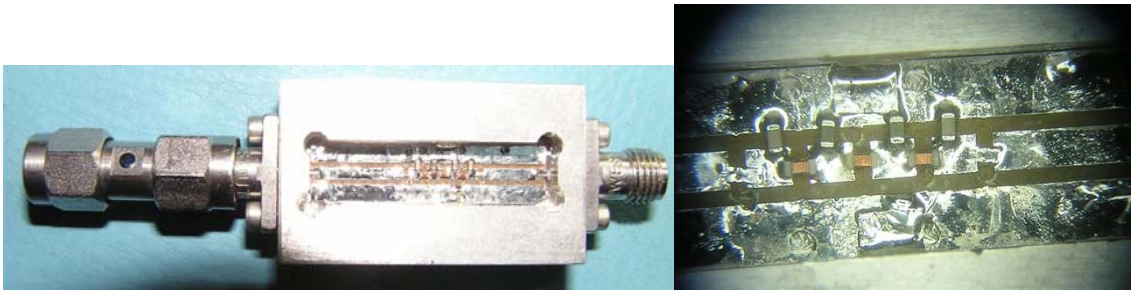

Figure 21: Seven Section Chebyshev Low Pass Filter

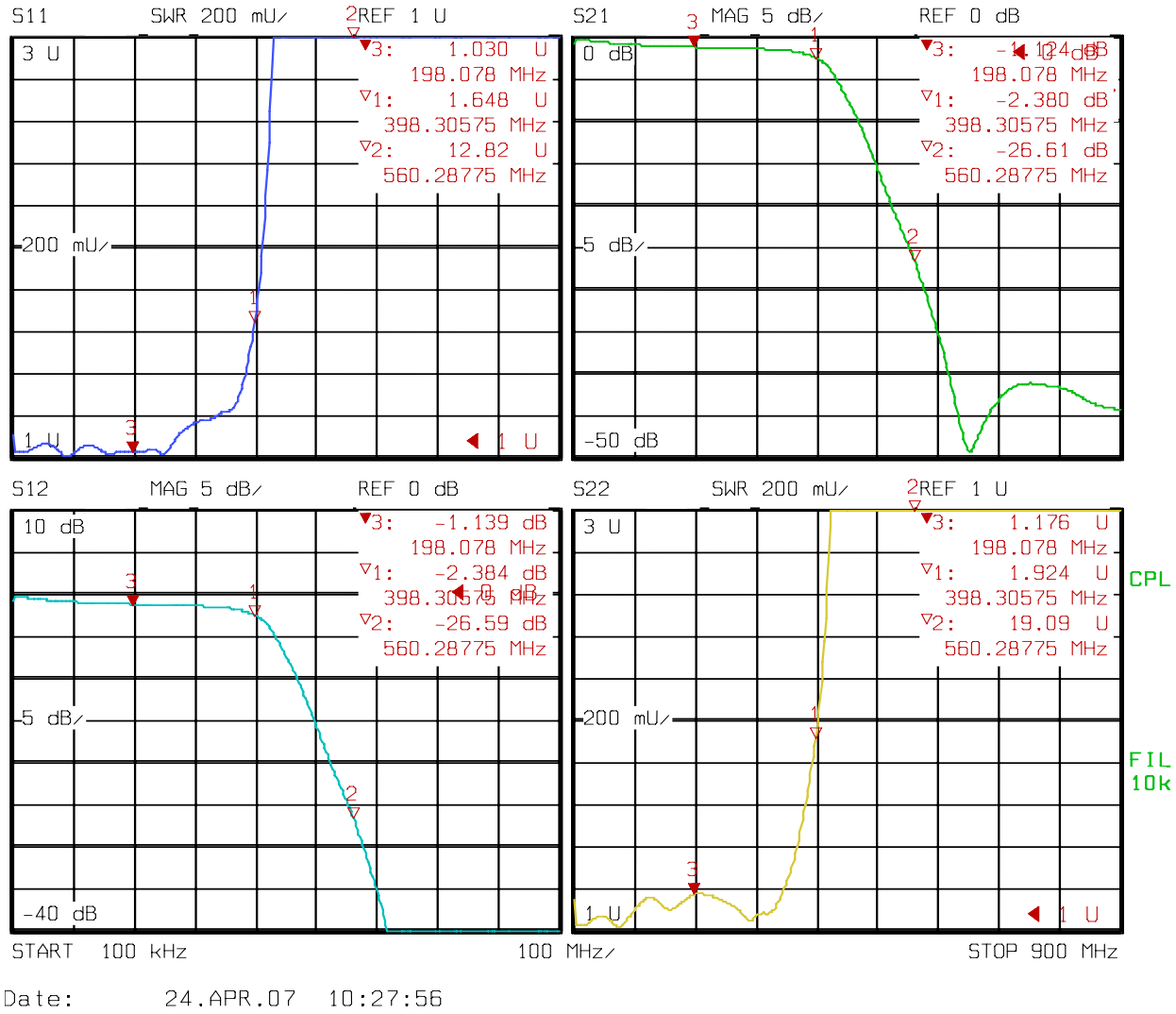

Figure 22: Network Analyzer Verification

4.1.2. Evaluation Boards:

Two evaluation boards were used for initial testing. The analog devices AD9858 (Figure 23) initially interfaced with a Xilinx Spartan 3 evaluation board (Figure 24). The two boards were connected via custom made interface circuitry (two connectors wired specially to mate the boards Figure 25). The analog devices evaluation board was feed a reference clock of 1 GHz at 0 dBm from a signal generator. The Sync clock from the AD9858 board was then feed out and used as a system clock for the Spartan 3 board. A very basic Finite State Machine (FSM) was coded and tested the communications between the two chips. The initial FSM test, configured the AD9858 in single tone mode, and a frequency was written to AD9858's frequency tuning word (FTW) registers [3]. The output of the AD9858 evaluation board was connected to a Rohde&Schwarz spectrum analyzer and its frequency was verified (Figure 26).

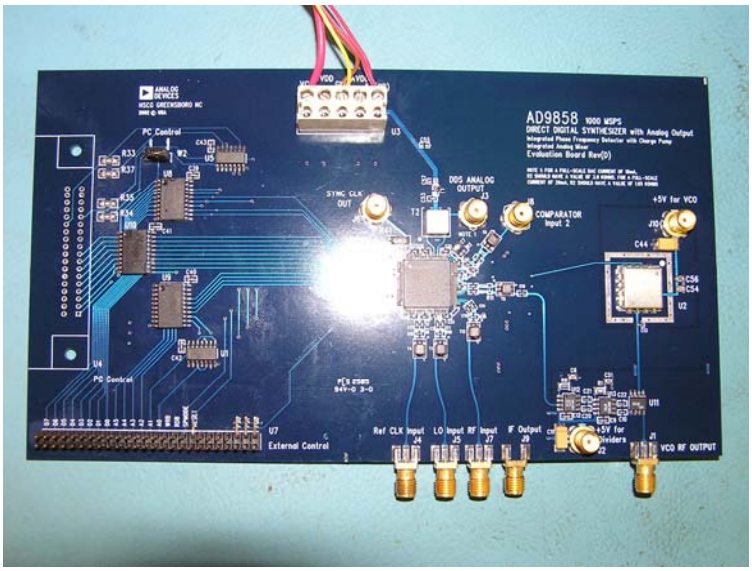

Figure 23: Analog Device's AD9858 Evaluation Board

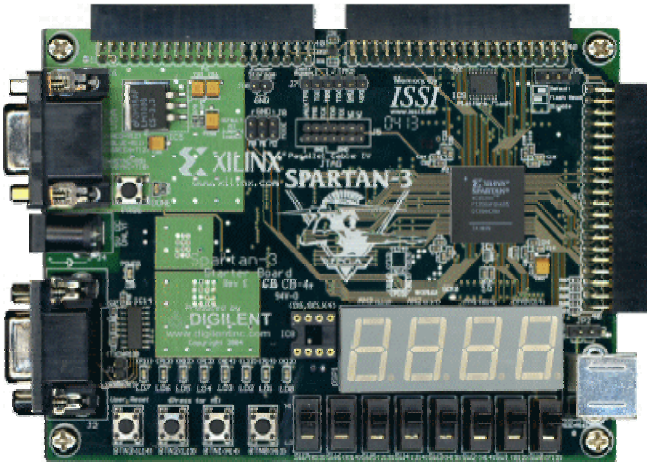

Figure 24: Digilent's Spartan 3 FPGA evaluation board [4]

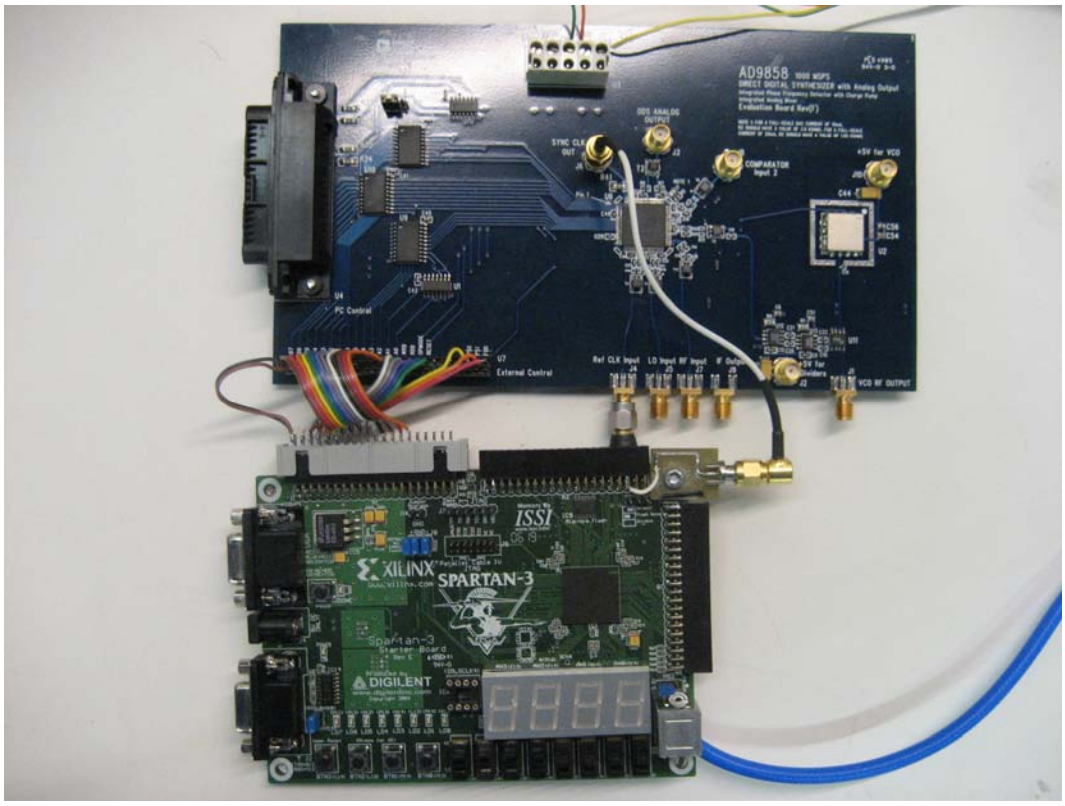

Figure 25: AD9858 interface with Spartan 3

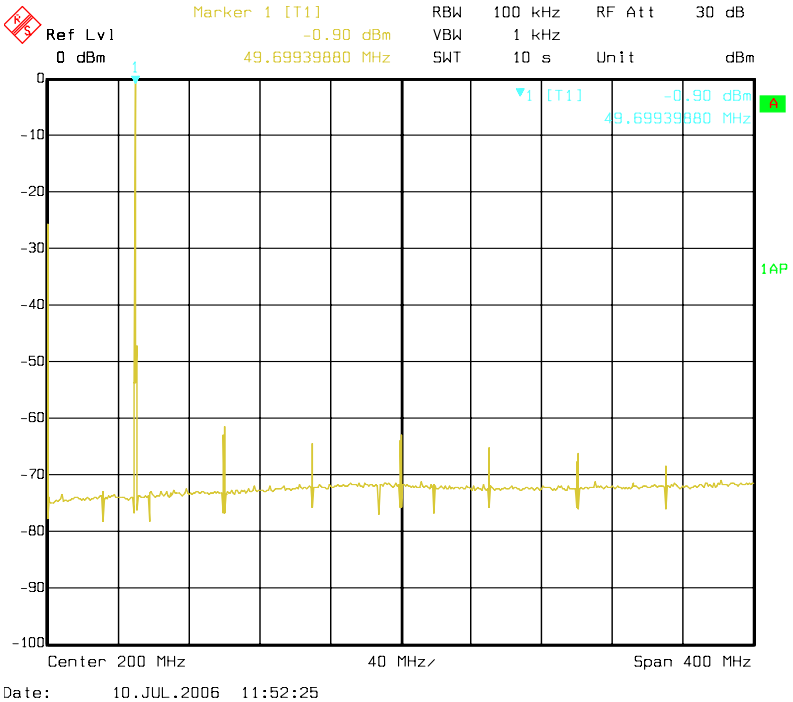

Figure 26: Spectrum Analyzer Verification

4.1.3. Printed Circuit Board:

Since evaluation boards were used for initial testing, each evaluation board provided us with their schematics. These two schematics were taken and laid out on a 5 layer printed circuit board (PCB) Figure 27. The PCB board layout was sent out to PCB123 for fabrication. When the PCB boards came in, each IC (integrated circuit) was hand soldered to the PCB under a microscope and checked over for any shorts. A simple VHDL program was written and loaded to the board to verify the FPGA circuitry.

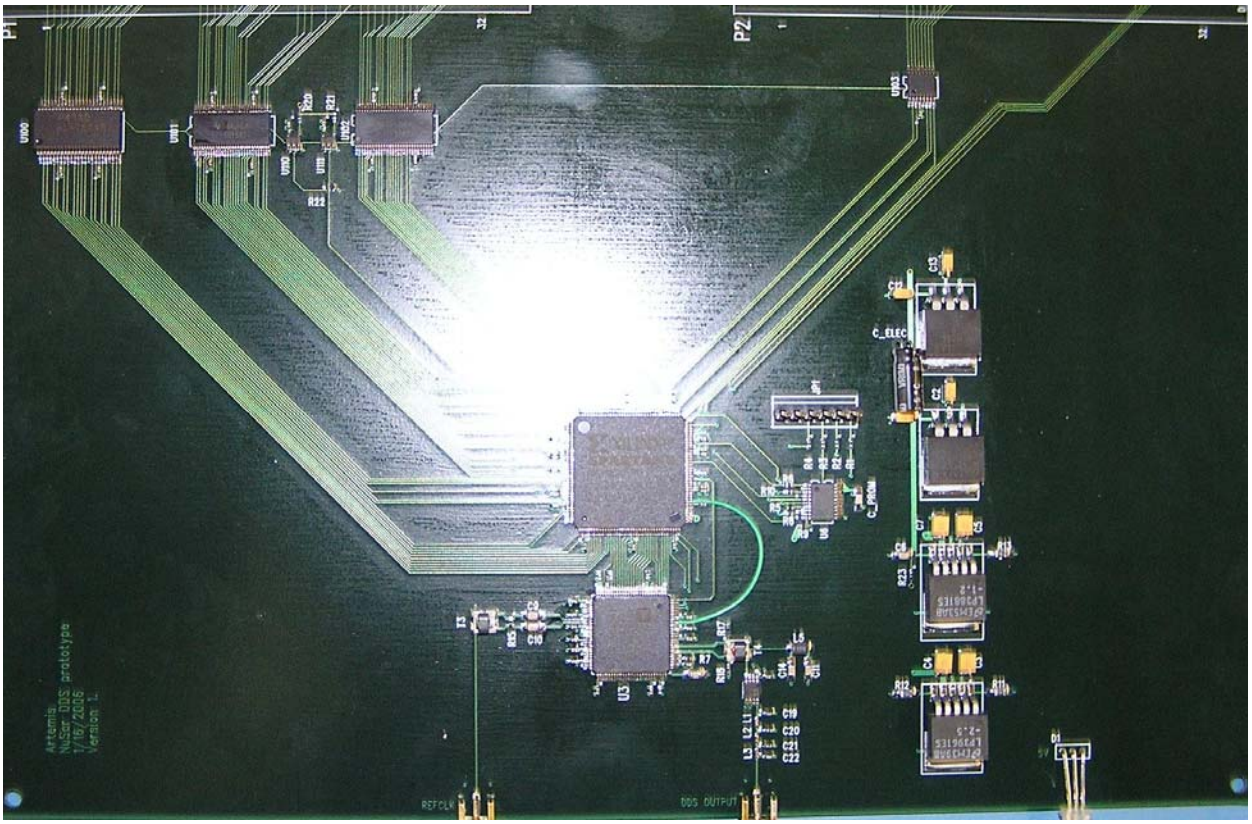

Figure 27: 5 Layer Printed Circuit Board

4.1.4. Final Verification:

After the build was complete the EEPROM was programmed with a final version of the VHDL code and each requirement was verified. Figure 28 shows the device attached to a Rohde&Schwarz spectrum analyzer with its output signal shown on the screen.

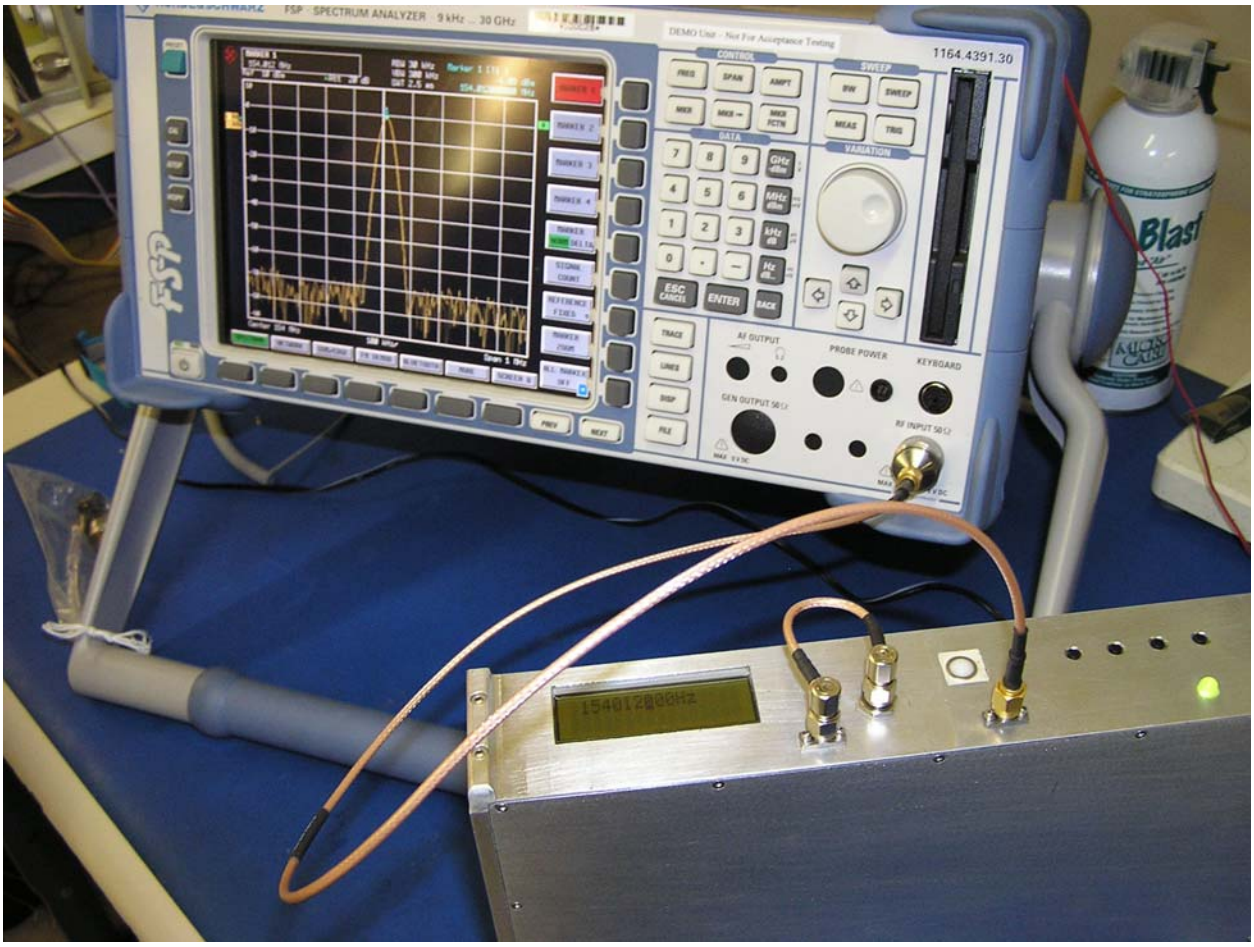

Figure 28: Output Frequency Verification

4.2. Requirements Verification:

4.2.1. Requirements Verification Chart

Requirement	Verification
The device must employ the DDS method of waveform synthesis.	An analog devices AD9858 Direct Digital Synthesizer chip was used to employ the DDS method.
All outputs signals must be available on standard SMA female connectors.	An SMA female connector was attached to the front panel of the device.
Four profiles will be available to set and select frequency configurations.	Four push buttons and the track ball select were used for profile set and select.
Frequency range should be from 0 MHz to 300 MHz.	The final tested output frequency ranged from 0 MHz to 350 MHz the output frequency was verified on a Rohde&Schwarz spectrum analyzer.
The user must be able to interactively select a frequency value into one of the user profiles.	The frequency storage was verified by storing frequencies in each profile and recalled.
The output spectrum should have a spurious free dynamic range (SFDR) below 40 decibles relative to one milliwatt (dBc).	The overall spectrum was verified with a Rohde&Schwarz spectrum analyzer.
The device must operate off a standard 120 volt AC wall outlet.	A 120 volt AC transform was used for the main power source.
The output frequency must be able to reach all multiples of 100 Hz between 0 and 300 MHz	The output frequency is accurate to a frequency resolution of 1 Hz.
The device must provide a quick and simple means (such as one push button for each profile) to select a frequency profile	One push button select is used to recall a stored frequency.
The device should display the frequency value stored within each profile	When a profile is selected the frequency value is written to the LCD screen.
The device's output power level must be between +10 dBm and -10 dBm. This is the voltage terminated into a 50 Ohm load	A Rohde&Schwarz spectrum analyzer verified that the output range was -2 dBm ± 2 dbm.
The generator should fit into a 4 inch X 12 inch X 12 inch volume	The dimensions were measured and verified.

Table 16: Requirements Verification Chart

5. Summary and Conclusions

5.1. Conclusion

Using the techniques outline in this document a continuous wave (CW) analog signal generator was developed. The overall project was completed successfully and all requirements were met. ARTEMIS had a chance to verify our design and was very impressed with the overall functionality of the system. A direct quote from ARTEMIS stated that “This is much cleaner than a standard generator“.

5.2. Further Improvements

If I was to redesign this project I’d only have a few improvements noted below.

- Use buttons for the frequency control instead of a track ball.
- Amplify the output and use an attenuator the to give the user control over the output power level
- Redesign the button interface using more stylish and easier to use buttons
- Shrink the PCB size so that the circuitry is denser and will fit into a smaller enclosure.
- Make similar feet used for the stands so that it can either stand up or lay on its side.

6. References:

[1] “Froogle” 2007. Google
< http://froogle.google.com/frghp?ie=UTF-8&oe=UTF-8&hl=en&tab=wf&q=> “Analog Deices

[2] “Filger Basics: Anti-Aliasing – Maxim Dallas” 2007. Maxim Integrated Products
<http://www.maxim-ic.com/appnotes.cfm/appnote_number/928>

[3] “AD9858 – 1 GSPS Direct Digital Synthesizer” 2007. Analog Devices
<http://www.analog.com/en/prod/0,,770_843_AD9858,00.html>

[4] “Diligent Inc. – Digital Design Engineer’s Source” 2007. Diligent Inc
<http://www.digilentinc.com/>

[5] “Xilinx Data Sheets Publications by Part” 2007. Xilinx
<http://direct.xilinx.com/bvdocs/publications/ds099.pdf>

Appendix A: Project Management Plan

A 1: Work Breakdown:

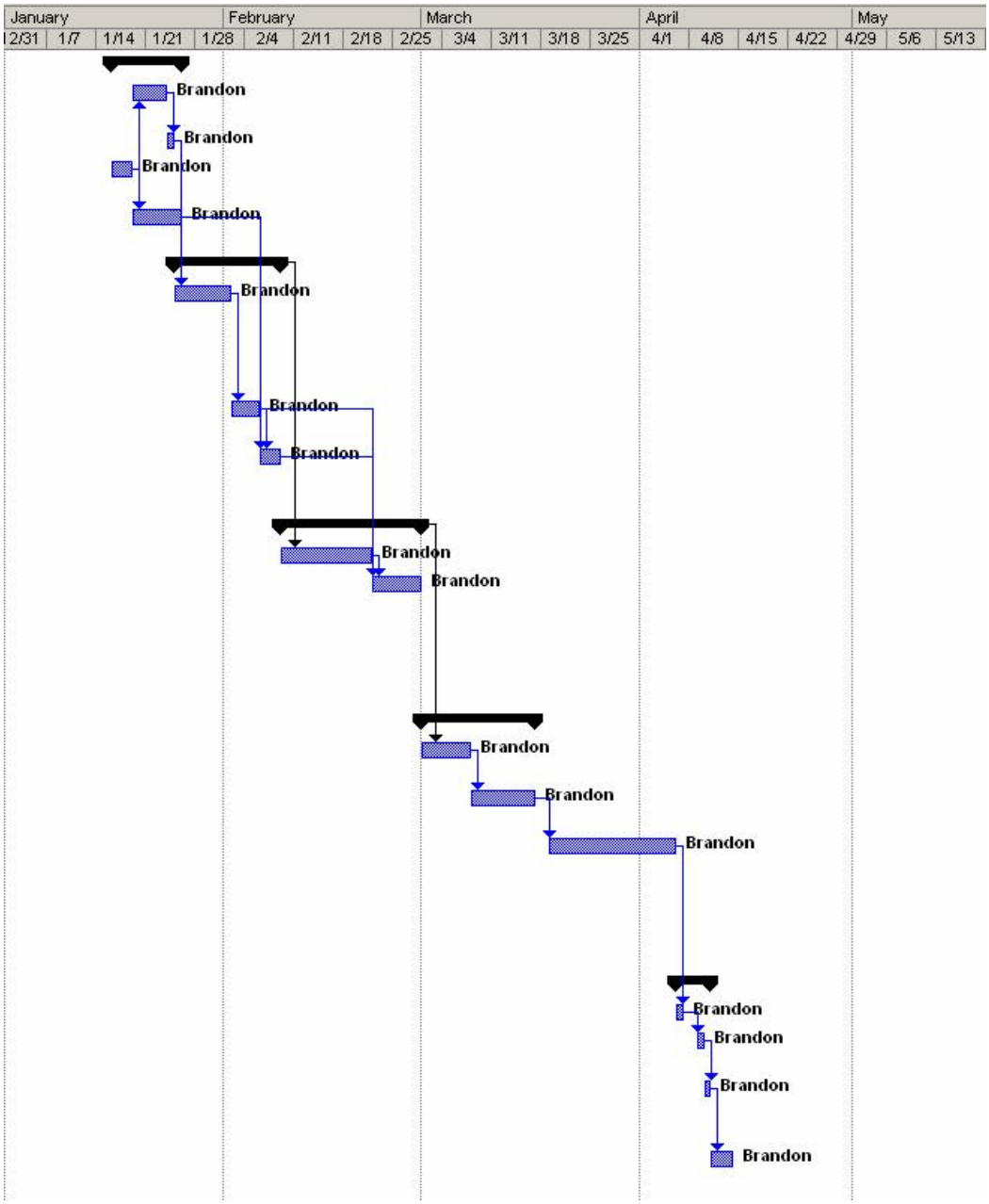

Figure 29: Work Time line.

		Task Name	Duration	Start	Finish	Predecessors	Deliverables	Resources
1		**Interface Circuitry**	**8 days**	**Tue 1/16/07**	**Thu 1/25/07**			
2		Design filter	3 days	Fri 1/19/07	Tue 1/23/07	4	Circiut Schematic, Simulation verification	PC, Microwave Office
3		Gather Parts	1 day	Wed 1/24/07	Wed 1/24/07	2	Sorted Parts	Artemis
4		Design user interface circuit	3 days	Tue 1/16/07	Thu 1/18/07		Circuit Schematic, Simulation verification	PC, SPICE simulator
5		Order parts for UI circuit	5 days	Fri 1/19/07	Thu 1/25/07	4	Place order, receive parts	Internet
6		**Proto-typing**	**11 days**	**Thu 1/25/07**	**Thu 2/8/07**			
7		Build filter prototype & test	6 days	Thu 1/25/07	Thu 2/1/07	3	Test data, Measurment of filter cutoff, verify functionality	ARTEMIS Filter housing, testbench, soldering iron, network analyzer
8		Integrate and test both eval boards with filter	2 days	Fri 2/2/07	Mon 2/5/07	7	Test data, verify functionality	Testbench
9		Build UI test setup	3 days	Tue 2/6/07	Thu 2/8/07	8,5	Test of FPGA communicating with LCD and buttons	Testbench, ModelSim, Xilinx
10		**Design & Test FSM**	**14 days**	**Fri 2/9/07**	**Wed 2/28/07**			
11		Design FSM for FPGA	9 days	Fri 2/9/07	Wed 2/21/07	6	ModelSim simulation	ModelSim, Xilinx
12		Test and revise FSM	5 days	Thu 2/22/07	Wed 2/28/07	11,9,8	Test data showing proper behavior between inputs and output with correct frequency range/resolution	ModelSim, Xilinx, Testbench
13		**PCB**	**12 days**	**Thu 3/1/07**	**Fri 3/16/07**			
14		Design PCB	5 days	Thu 3/1/07	Wed 3/7/07	10	Layout files, Gerber files	ORCAD Capture, ORCAD Layout
15		Order PCB	7 days	Thu 3/8/07	Fri 3/16/07	14	Place order, Receive parts	
16		System Integration and Test	14 days	Mon 3/19/07	Thu 4/5/07	15	Test data showing proper behavior between inputs and output with correct frequency range/resolution	Testbench, network analyzer
17		**Device Housing**	**3 days**	**Fri 4/6/07**	**Tue 4/10/07**			
18		Design device housing	1 day	Fri 4/6/07	Fri 4/6/07	16	3-D Model	AutoCad 2004
19		Fabricate housing	1 day	Mon 4/9/07	Mon 4/9/07	18	Finished casing	Artemis assembly department
20		Mount system into housing	1 day	Tue 4/10/07	Tue 4/10/07	19	Complete system	Screws, Screwdriver, Spacers
21		Final Test	3 days	Wed 4/11/07	Fri 4/13/07	20	Test data showing proper behavior between inputs and output with correct frequency range/resolution	Testbench, network analyzer

Figure 30: Work Breakdown Structure

A 2: Member Contributions:

Majority of this project was designed and executed by Brandon Wicks. Adam Elliott contributed with the initial preliminary designs and construction of this report. Since, Adam and myself either fell out of sync with the spring and fall semester, or in my case was offered a full time job, most of the work was unable to be completed as a team effort.

A 3: Development costs:

Part	Description	price	quantity	total price
ARTEMIS PLO	960 MHz Phase Lock Oscillator	0	1	0.00
LP3961ES-2.5	2.5 V regulator	3.04	1	3.04
LP3881ES-1.2	1.2 V regulator	4.45	1	4.45
LM1086IS-3.3	3.3 V regulator	2.42	2	4.84
TPSB686K010R0600	68 uF Tantalum	0.98	4	3.92
TPSA106K010R0900	10 uF Tantalum	0.61	4	2.44
xc3s400-4tq144I	spartan 3 FPGA	25.25	1	25.25
xcf02svo20c	PROM	4.25	1	4.25
PPPC202KFMS	40 pin female header	1.69	2	3.38
90120-0766	6 pin male header	2.27	1	2.27
TC1-1-13M	Transmittion transformer	1.49	1	1.49
TC1-1T	RF transformer	1.69	1	1.69
AD9858	DDS	79.64	1	79.64
	100 Ohm resisitor	0.2	5	1
	4.7 kOhm resisitor	0.2	3	0.6
	51 Ohm resistor	0.2	3	0.6
	.1 uF capacitor	0.2	5	1
C1005X7R1C103K	.01 uF capacitor	0.0126	40	0.504
	10 kOhm resistor	0.2	4	0.8
	33 nH inductor	0.2	2	0.4
	40 nH inductor	0.2	1	0.2
	12 pF capacitor	0.2	2	0.4
	22 pF capacitor	0.2	2	0.4
PCB Board	5 layer pcb board	108.5	1	108.5
			total	251.064

Table 17: Part List

Appendix B: Software

Button Debounce Code:

```
Entity DEBOUNCE is
  port  ( clk,reset,enable,RawKeyPress: in  std_logic;
          debounced: out std_logic);
end;

architecture behavior of debounce is
signal state: std_logic_vector(15 downto 0);
begin
debounce_process: process(clk,reset)
      begin
         if (reset='1') then
            state<=(others=>'0'); debounced<='0';
            elsif(clk'event and clk='1' and clk'LAST_VALUE='0') then
               if enable='1' then
                  state<=(state(14 downto 0)& not(RawKeyPress))or x"e000
                        if state=x"f000" then
                              debounced<='1';
                        else debounced<='0';
                  end if;
               end if;
            end if;
end process;
end architecture;

entity AD9858_fsm is
   port(clk,reset: in std_logic;
        sw: in std_logic;
          cw: out std_logic_vector(5 downto 0));
end entity;
```

AD9858 FSM Code:

```
architecture behavior of AD9858_fsm is
type state_type
is(rst,CFR_0,WR_0,CFR_1,WR_1,CFR_2,WR_2,CFR_3,WR_3,FTW0_A,WR_A,FTW0_B,WR_B,
                           FTW0_C,WR_C,FTW0_D,WR_D,FUD,delay);
signal state: state_type:=rst;

begin

state_process: process(clk,reset,sw)
      begin
         if (reset='1') then
            state<=rst;
            elsif(clk'event and clk='1' and clk'LAST_VALUE='0') then
                  case state is
                     when rst=> state<=CFR_0;
                     when CFR_0=> state<=WR_0;
                     when WR_0=> state<=CFR_1;
                     when CFR_1=> state<=WR_1;
                     when WR_1=> state<=CFR_2;
```

```
                    when CFR_2=> state<=WR_2;
                    when WR_2=> state<=CFR_3;
                    when CFR_3=> state<=WR_3;
                    when WR_3=> state<=FTW0_A;
            when FTW0_A=> state<=WR_A;
            when WR_A=> state<=FTW0_B;
            when FTW0_B=> state<=WR_B;
            when WR_B=> state<=FTW0_C;
            when FTW0_C=> state<=WR_C;
            when WR_C=> state<=FTW0_D;
            when FTW0_D=> state<=WR_D;
            when WR_D=> state<=FUD;
            when FUD=> state<=delay;
            when delay=>
                if sw='1' then
                     state<=FTW0_A;
                else
                     state<=delay;
                end if;
                  end case;
            end if;
end process;
output_process: process (state)
      begin
        case state is
            when rst=> cw<="010000";
               when CFR_0=> cw<="010000";
                  when WR_0=> cw<="000000";
                  when CFR_1=> cw<="010001";
                  when WR_1=> cw<="000001";
         when CFR_2=> cw<="010010";
         when WR_2=> cw<="000010";
         when CFR_3=> cw<="010011";
         when WR_3=> cw<="000011";
         when FTW0_A=> cw<="011010";
         when WR_A=> cw<="001010";
         when FTW0_B=> cw<="011011";
         when WR_B=> cw<="001011";
         when FTW0_C=> cw<="011100";
         when WR_C=> cw<="001100";
         when FTW0_D=> cw<="011101";
         when WR_D=> cw<="001101";
         when FUD=> cw<="111101";
         when delay=> cw<="011101";
     end case;
end process;
end architecture;
```

Appendix C: Definitions and Acronyms/Abbreviations

CPLD: Complex Programmable Logic Device

Daisy Chain: Set of hardware connected in series.

dBm: Power level relative to the carrier frequency

dBc: Power level relative to 1 mW

DDS: Direct Digital Synthesizer

FPGA: Field-Programmable Gate array

JTAG: Joint Tag Access Group

LO: Local Oscillator

PLL: Phase Lock Loop

PLO: Phase Lock Oscillator

Profile: waveform settings that distinctly defines a waveform

RXE: Receive Enable

TTL signal: Transistor-transistor logic (0 Volts – 5 Volts DC)

TXE: Transmit Enable

Upconverter: A device that converts a low frequency signal to a microwave (1GHz – 300 GHz) frequency.