ECE - 493 Final Report

High-Speed Analog Capture Device (HACD)

Team

Alex Wozneak
Rishub Nagpal
Ryan Meruvia

Faculty Supervisor

Dr. Jens-Peter Kaps

Class Instructor

Dr. Peter Pachowicz

Date of Submission: 4/28/2019

Executive Summary

High-speed oscilloscopes are useful for measuring high frequency and low power analog signals. There are a multitude of applications where high-speed oscilloscopes are useful and often required. The main problem the average user in need of a high-speed oscilloscope faces is the exceedingly high cost these oscilloscopes usually mandate. This is primarily an issue for users in academic and hobbyist communities where funding is often limited.

For example, a typical oscilloscope capable of measuring signals at 500-800 MHz can cost anywhere from $6,000 to $13,000 or more. This is why our goal is to design a low-cost, open-source high-speed oscilloscope daughterboard. We've designated the name of the open-source board to be The High-Speed Analog Capture Device (HACD).Throughout the following we will present what we've learned and the results we've obtained for HACD. Our faculty supervisor Dr. Jens-Peter Kaps has provided guidance and feedback throughout the project.

HACD is divided between two main components. The first is the digitizer daughterboard that interfaces via FMC connector with an FPGA (Zynq) development board, the Zedboard. The second component is the firmware and software needed for getting sample data off of our digitizer board. Each of these components were design around an 8-bit, 1 Gsps analog-to-digital converter (ADC). The cost of the parts for our design is around $220, and the cost of getting the pcb manufactured would be between $50 and $70. So together our solution would cost around $300 total, which is significantly less expensive than some of the alternatives.

Table of Contents:

1.0 Approach

1.1 Problem Analysis

There are many engineering applications especially in communications, digital circuit design, and cryptography when a user needs a device that can measure frequencies of 100 MHz or more. In particular, students and professors in academia or hobbyists who require devices of this nature face a problem, which is the exceedingly high cost of oscilloscopes that can measure signals up to 500 MHz. Currently there are several open-source oscilloscopes available online that can be utilized by users in communities previously mentioned. But most if not all of these open source designs do not performance specifications we aim to achieve with our design, specifically a bandwidth of 500 MHz. HACD will achieve a certain level of performance at a fraction of the cost of oscilloscopes with comparable performance that are currently on the market. Having such a device at an affordable cost opens up a variety of high frequency applications to a user. An ideal use case for our device would be for assisting graduate students and professors in their research.

We've identified the need for a lower cost measurement device that can accomplish the same bandwidth as more expensive equipment. Our design was never intended to compete with the feature rich high-end off the shelf oscilloscopes, but mainly to achieve the same bandwidth. HACD will accomplish this by providing a complete analog capture system at an affordable price. We'd like to see HACD act as a catalyst for promoting the expansion of open-source projects for high performance devices that open up research possibilities for any kind of researcher on a tight budget. The table below shows several other 500 MHz bandwidth oscilloscope on the market with their respective prices. As you can see the cost is significant.

Table 1: Commercially Available High-Speed Scopes

Picture	Bandwidth	Sample Rate	Price	Model/Manufacturer
	500 MHz	5 GS/S	$6,595	PicoTech PicoScope 6000 Series

	500 MHz	10 GS/S	$9,999	Rigol DS7054
	500 MHz	2.5 GS/S	$9,290	Tektronix MDO3052
	500 MHz	5 GS/S	$9,346	Keysight DSOX3052T

1.2 Design Methodology

We approached this design using a top-down method. This resulted in separating our system into two parts. The radio frequency (RF) daughterboard front-end is the first part which will deal with all of the analog signal manipulation. The second half is the firmware back-end which stores and processes the digital data on the Zedboard and sends it to the user computer. These two parts will be interfaced via a FPGA Mezzanine Card (FMC). A diagram of how we logically approached this project is shown in figure 1.0.

Our approach to the RF daughterboard was to conduct research on what typical high-speed oscilloscope front-ends look like and what components are used. One of the most useful reference designs we found was the TIDA-00826 from Texas Instruments (TI). While this design was a beneficial reference, the specs required for the HACD were different than what the TIDA-00826 was capable of. Due to these differences, more research needed to be conducted to find specific components that were rated for our specific specs. Once those were acquired, we were able to construct a flow graph of how the analog signals would travel to and from each integrated circuit (IC). The analog signal will enter in single-ended and then go through the RF circuitry and be fed to the ADC differentially. The ADC will then convert the data and send it to the firmware back-end via FMC.

Our backend design is a firmware based solution for ZYNQ-7000 SoCs. Specifically, we targeted the ZedBoard since this is one of the most ubiquitous entry level ZYNQ device available on the market. The design was based mostly on our specific ADC. The firmware will consist of a data forwarding pipeline which will take the input stream from the frontend via FMC and forward it to the Ethernet controller for

transmission to a PC. The input stream will first be deserialized using primitives build into the I/O banks. After, we plan to forward data within the FPGA using AXI format packets to DRAM. This allows us to maximize our usage of Xilinx IPs which will increase the portability of the firmware to other FPGA platforms.

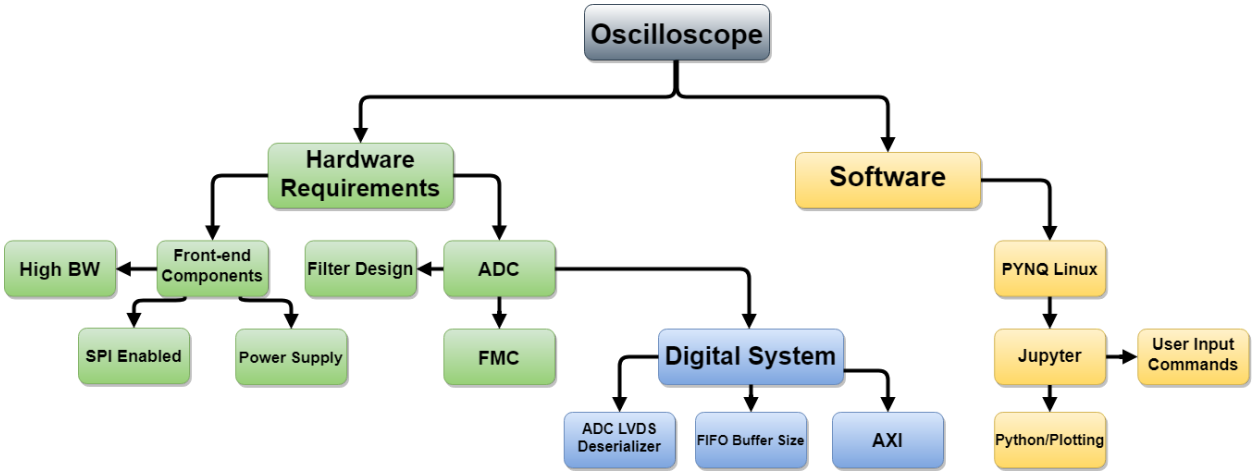


Figure 1.0: Approach Diagram

1.3 Alternative Designs

There are multiple different ways you could go about designing this open-scope. Each are able to obtain the same results with some different components or approaches.

One of the approaches is to purchase the evaluation board for the analog-to-digital converter (ADC) that will get the signal from a separate printed circuit board (PCB). This may also include getting the clock that drives the ADC from a separate board as well, or having the clock generated on the evaluation board as well. This approach simplifies the design of the PCB, but has its own drawbacks. Due to the evaluation board's inability to accept a differential signal, the analog signal will need to be converted back to single-ended right before the SubMiniature version A (SMA) connector. There is also the possibility of introducing noise to the signal that would not usually be there if every component was on the same PCB. This also causes issues with the control signals as some will need to be sent via FMC to the ADC and some via Peripheral Module (PMOD) to the RF daughterboard.

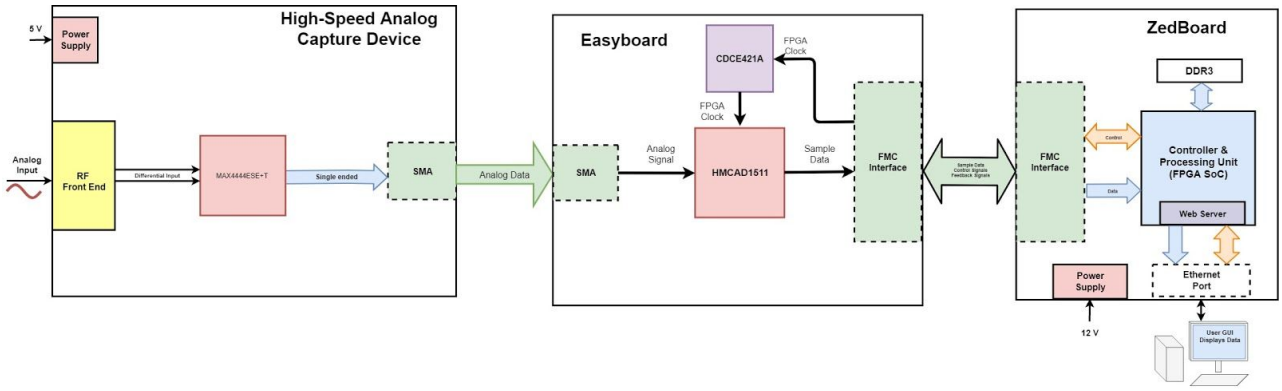


Figure 1.1: Alternative Design

Another approach would be to use a faster ADC with a higher resolution in our project. This would allow a lot more freedom, as well as allow sampling at higher rates if required. This may seem like a better approach and one that would make the project more efficient, it defeats a lot of the purpose of our implementation. We want to make a low cost open-scope and the ADC we found was the cheapest one that would sample at the rate we wanted. Any rate higher would cause the ADC to cost hundreds of dollars and then would not be realistic for hobbyists or students due to the high cost.

While there are multiple different ways this problem could have been approached, after weighing the pros and cons of each, we decided on our specific implementation. It seemed to be the most plausible for the time we were given for the project as well as the one that fit the requirements the most. The design would be simple to understand as well as not too expensive so it may be implemented as an open source solution.

1.4 Team Contributions

We were able to allocate the work effectively for this project by assigning tasks that best fit each group members experience and interests. The area we lacked the most knowledge in was RF PCB design, this is why two of the three team members were assigned to research how to efficiently design the RF daughterboard. The third member focused mostly on the firmware back-end of the project so we could obtain and display the data from our board.

Alex and Ryan spent a majority of their time learning about RF circuit design and how to approach designing the pcb. This began with conducting heavy research on oscilloscope front-ends and components rated for the necessary requirements. Through this research, we were able to find an extremely helpful reference design. This design was the TIDA-00826 from Texas Instruments (TI) which is a 2 GHz single channel oscilloscope daughtercard. Although this design’s specifications exceeded our

requirements, we were able to use many of their design methods in our front end. Using this reference design, we were able to determine the amplifiers and voltage regulators that we would use in our design. The abstracted diagram of our daughterboard was designed based on the framework provided by this reference design. Using this information, Ryan was able to conduct circuit simulations in TINA TI, which is TI's own circuit simulation software. This helped to confirm that the full system would work as expected. Once these tests were confirmed, the process of constructing the full schematic and bill of materials (BOM) began. Ryan and Alex split the work of the schematic up equally and reviewed the other's schematic to verify for correctness etc. This process allowed us to efficiently build the schematic and confirm that it followed the correct outline. Next came picking each of the small components which was a collaborative effort as well. Ryan would choose parts first and Alex would go over the parts chosen and confirm they were rated for our application and were readily available. Once this was finished, the process of designing the PCB layout began. Due to the lack of time and experience with designing PCBs of this scope, we contracted a freelancer who assisted us with the layout. Afterwards, Alex contacted multiple different companies for PCB fabrication and assembly quotes. The company that fabricated and assembled our pcb was called Imagineering inc. .

Rishub was the lead of working on the firmware back-end of this project. This involved conducting research on FMC connectors and the specific FPGA we would be using. Both were relatively new concepts so Rishub needed to analyze the potential issues using this specific FPGA may cause. After learning more about the specific equipment being used, Rishub needed to boot Linux onto the FPGA. This involved porting Pynq linux, a linux distro described in our initial requirements, to the Zedboard. The porting process involved rebuilding the kernel and boot files specifically for the Zedboard hardware. Once that was complete, Rishub spent time creating example software to validate the port.

The next step was to build the data forward pipeline to send the ADC data to DRAM. Due to Rishub's past experience in this area, this process was relatively straightforward. Rishub created a sample test generator which simulates the HMCAD1511 to verify the data forward pipeline.

The next task was to communicate with the HMCAD1511 evaluation board. Since the board peripherals use the SPI protocol, drivers needed to be written so that the hardware SPI block instantiated in the PL layer of the FPGA could be controlled through Python.

Finally, a data deserializer was created to convert the incoming ADC data into packets which could be converted to the AXI format. Since the HMCAD1511 provides data in a double data rate format at high speeds, built-in primitives within the I/O banks of the FPGA were utilized for maximum speed and performance.

Due to each members different experiences and exposures to the concepts of this project, we were able to allocate the work in the most efficient way possible. This allowed us to move at a good pace with this project and mostly stay on our schedule.

2.0 Technical Section

Oscilloscope Specifications Table

HACD Specifications	
Vertical	
Input Channels	1
Bandwidth (-3 dB)	500 MHz
Rise Time (10%-90%, calculated)	0.7ns
Input Type	Single-ended, SMA connector
Resolution	8 bits
Input Sensitivity	10 mV/div to 1 V/div at x1 zoom
Input Range	+/-50 mV to +/- 5 V
Input Coupling	AC, DC, 50 Ohm
Maximum system voltage gain	19.97 dB
Input characteristics	50 Ohm
Offset range	+5V to -3 V
Overvoltage protection	+/- 5.5V RMS
Horizontal	
Sampling Rate (1 Channel)	1 GS/s
Sampling Rate (2 Channel)	500 MS/s
Sampling Rate (4 Channel)	250 MS/s
Buffer memory	64 kB
Sampling Clock Generator	
Input clock range	4 kHz-500 MHz
Output clock range	4.25 MHz-1.175 GHz

2.1 Functional Decomposition

The first diagram is the top level view of our design. It is a simple black box representation showing the basic inputs and outputs of the system. This results in the system shown in figure 2.0.

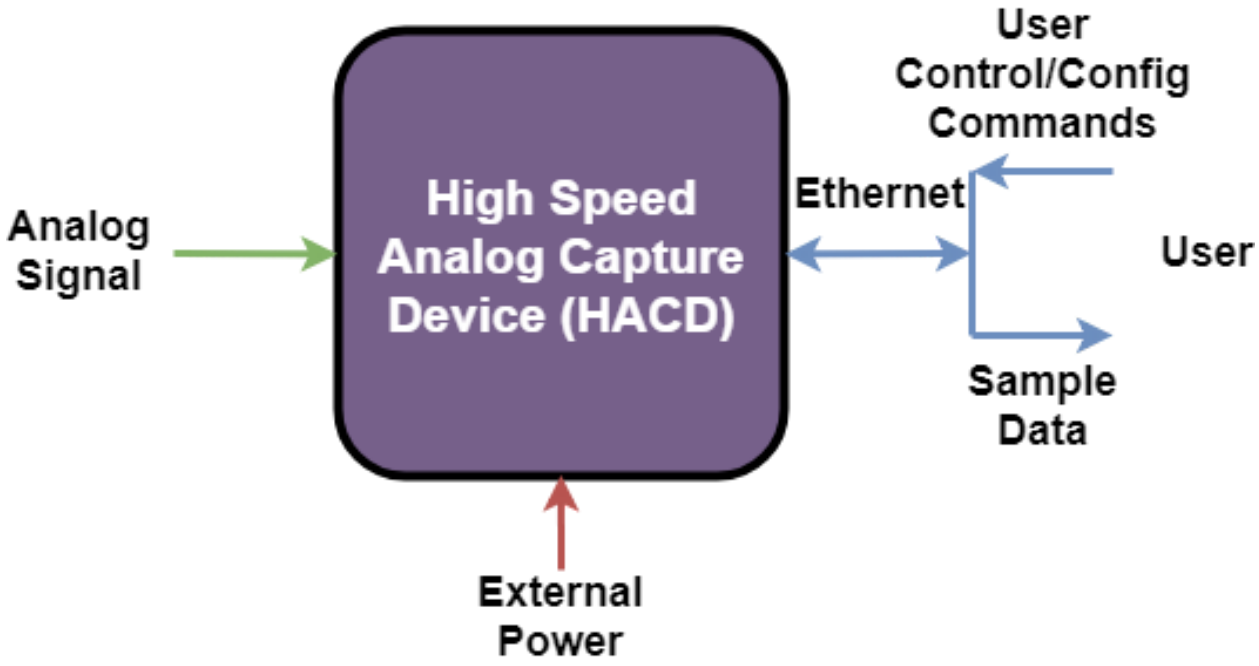


Figure 2.0: Black Box Diagram

There is one 50 ohm impedance analog input channel where the input signal will be applied. The outputs of this system that the user will see is the sample data accessed through a web server. The user control inputs are also facilitated using the same web server. Finally HACD will be powered by an external power source as the Zedboard isn't able to supply the proper voltages. These are all of the inputs and outputs of the black box system.

After analyzing the full black box, we were able to compartmentalize the primary subsystems implemented in our design.. We then connected the functions in such a way that it shows how the data will flow in the system. This functional decomposition can be seen in figure 2.1.

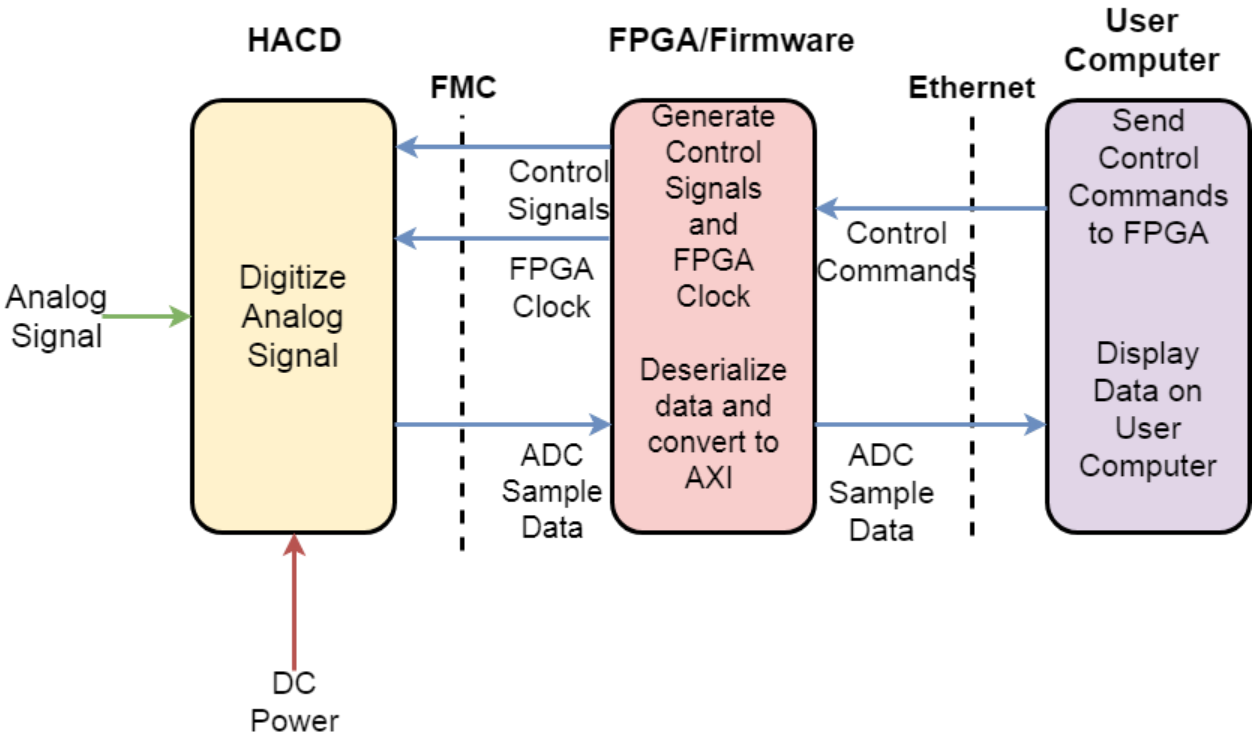


Figure 2.1: Functional Decomposition 1

A better sense of the full system is gained through each of the blocks in this decomposition. The analog data will be inputted to the front-end daughterboard. This daughterboard will manipulate the signal and finish by digitizing it. Once the digitized signal is obtained, it will be sent via FMC to the FPGA. The FPGA will deal with storing and processing the sample data. Once all of that is done, the data will be sent to the user computer via Ethernet, where it can be displayed. At the same time, the user computer will be able to send control commands to the FPGA. The FPGA will then generate the control signals as well as the FPGA clock which will be sent back to the front-end daughterboard via FMC. This allows the user to control all of the programmable devices on the daughterboard.

Once each of these functions was identified, as well as the flow of the data between each function, we then decomposed each of those to see a much more detailed view of each of the functions. In figure 2.2 below, we show the functional decomposition of digitizing the analog input waveform.

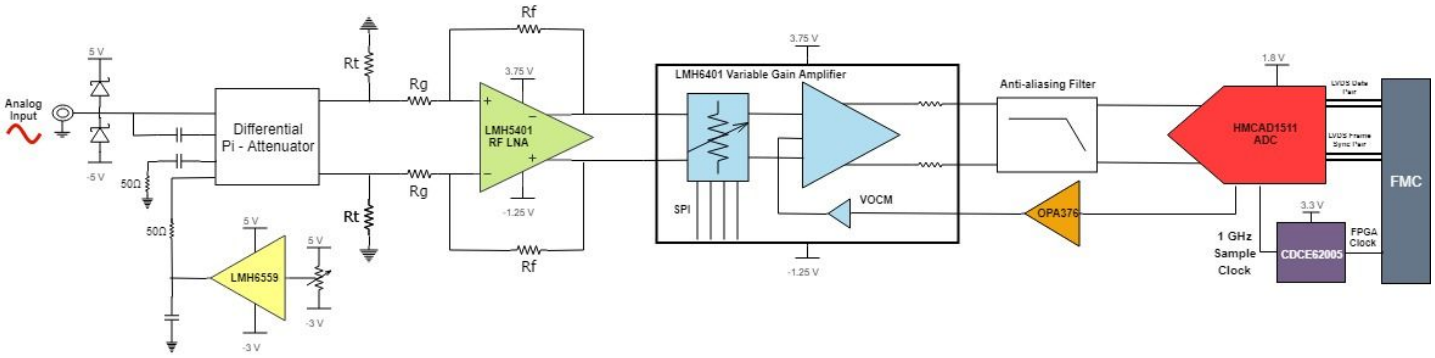


Figure 2.2: Front-End Circuit Diagram

This decomposition shows the flow of the analog signal through the front-end and identifies each of the main components that make up the front-end. The analog waveform will be inputted to the system by an SMA connector. Right after that are two schottky diodes with 5 volts and -5 volts. This serves as protection since we are inputting a maximum of 10 Vpp. The signal then feeds into the attenuator. This will drop the voltage of the input signal so it can be closer to the full-scale range (FSR) of the ADC. The LMH6559 and the potentiometer below the attenuator serve to deal with any DC offset on the input signal. Once through the attenuator, the signal will go through an LNA to amplify the signal a little bit and convert it to a differential signal. This then feeds to the VGA which will apply gain to the signal and raise it as close to the ADC FSR as possible. Next is the anti-aliasing filter which will filter out signals that are above the maximum input we are accepting. Our maximum input is 500 MHz so signals greater than that will be filtered out. Once filtered, the signal is fed to the ADC for sampling and quantizing. Once sampled and quantized, the digital data will then be sent to the FPGA via FMC. The ADC clock is being driven by the CDCE62005 PLL which obtains its reference clock from the FPGA. Lastly the OPA376 deals with the common mode voltage output of the ADC and feeds that to the VOCM pin on the VGA.

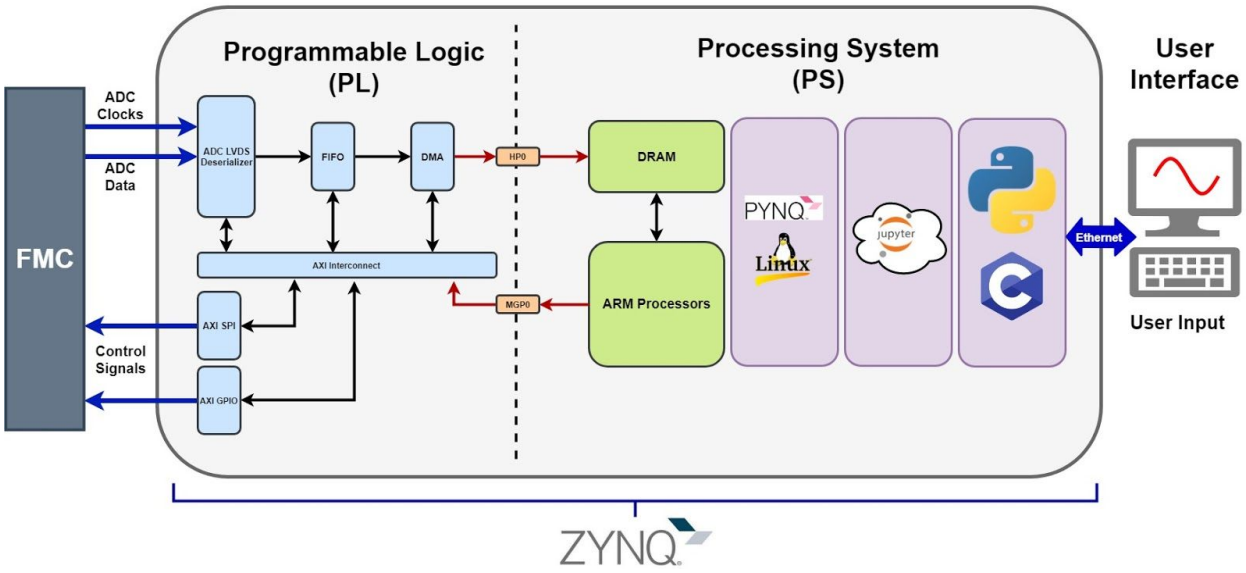


Figure 2.3: ZYNQ Architecture

Figure 2.3 above is the system diagram of the FPGA/Firmware and the software side of this project. The digital system used to deserialize and buffer the incoming ADC data is located on the Artix-7 FPGA located on the Programmable Layer (PL). PYNQ Linux is loaded onto the Processing System (PS). PYNQ Linux contains the Jupyter Notebook server which the user interacts with the control the system. The decompositions of the PL and PS system portions of our design are seen in figures 2.4 and figure 2.6.

Figure 2.4 shows the data flow on the PL side of this project. The ADC clocks and the ADC data are sent over from the RF front-end to the FPGA via FMC. Once it is sent over, the data needs to be deserialized. This is done through the HMCAD1511 LVDS Deserializer IP that was written for this project. The flow of the data through that IP is shown in figure 2.5. The IP deserializes the incoming data using the IDELAY2 and ISERDES2 primitives. The bit clock (DCO) and the frame clock (FCO) from the ADC are used to align the data samples correctly. The Bit Clock and Frame Clock Alignment blocks in Figure 2.5 contain state machines which automatically align the deserialized data when the SYNC signal is sent from the ADC. Since there are 8 data input channels, and each sample is 8-bits wide, the deserialized width of the data packet is 64-bits. Finally, the IP forwards this data packet along to a 64-bit wide AXI4-Stream output bus clocked on the FCO clock domain. The AXI bus is connected to a 64 kB AXI-Stream FIFO. This FIFO is used to cross from the FCO clock domain to the global 100 Mhz clock domain. The output of the FIFO is connected to the DMA which facilitates sending the data to main memory on the PS side through the HP0 interface port. All of these blocks are also connected to the AXI Interconnect bus which assigns

MMIO addresses to allow for user programming. The AXI SPI and AXI GPIO IPs are controlled via user programming and translate user commands into the appropriate SPI and control signals.

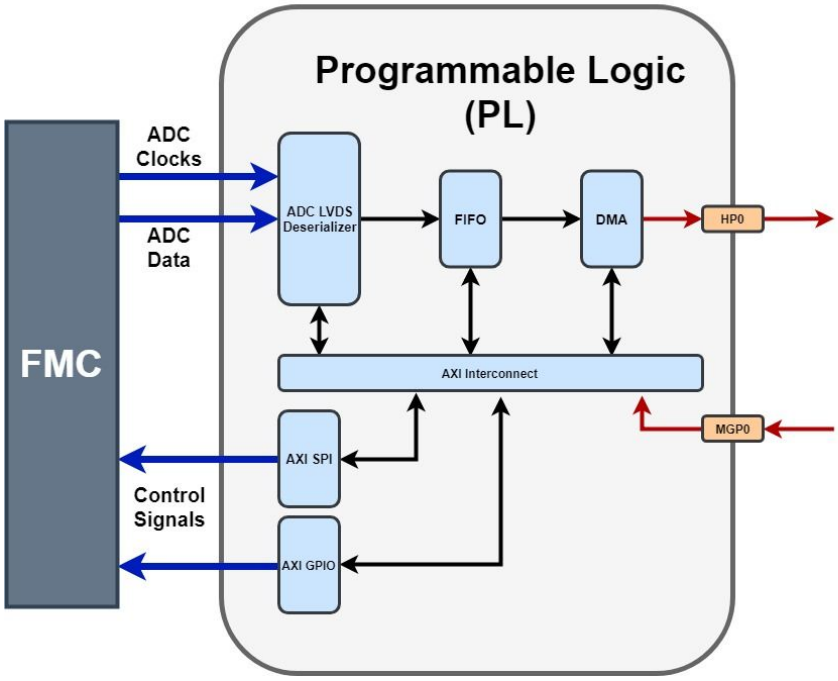


Figure 2.4: Programmable Logic

Figure 2.4 above shows what is in the PL side of this project. The HP0 and MGP0 ports facilitate communication with the PS.

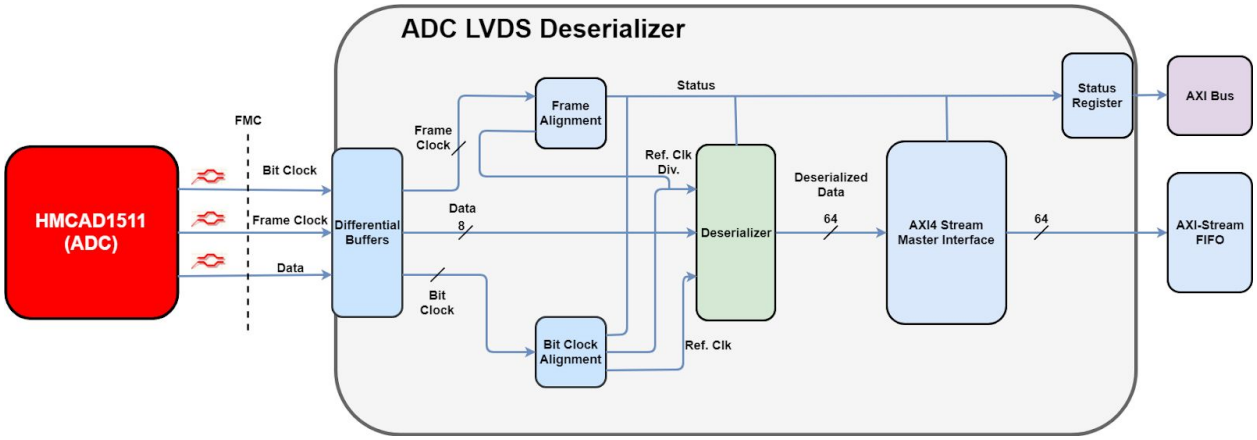


Figure 2.5: ADC LVDS Deserializer Block Diagram

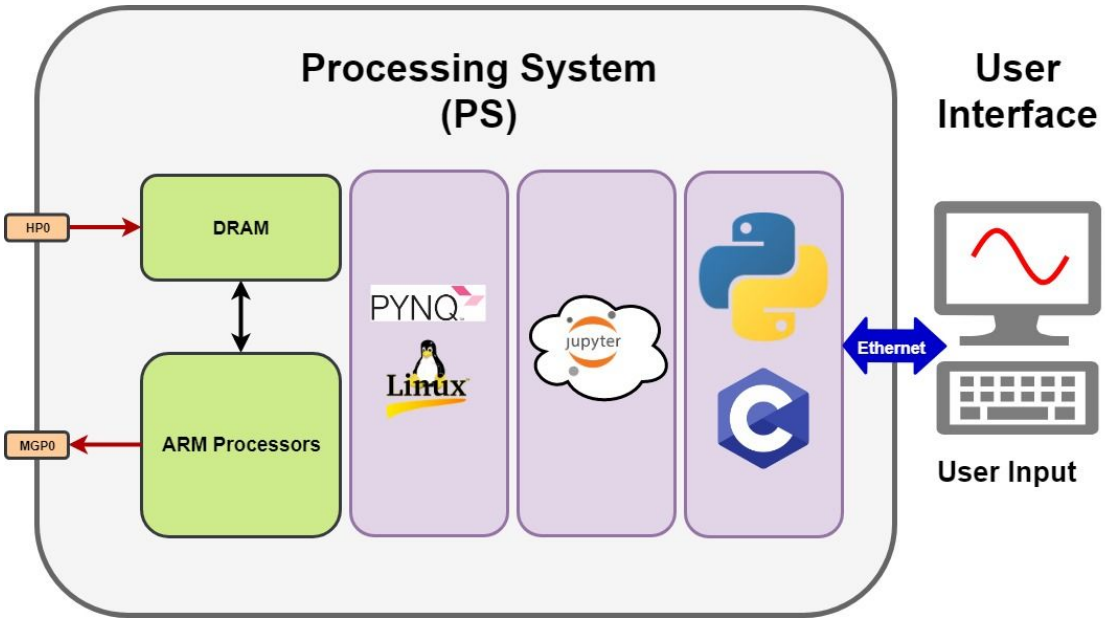


Figure 2.6: Processing System

Once each part of the system is fully decomposed, we get a system architecture that looks like the diagram in figure 2.7. We have the two parts of this project split up into their own sections, the RF Daughterboard and the Zedboard. The analog waveform is inputted to the RF Daughterboard and manipulated through the RF front end circuitry. The manipulated signal then goes to the ADC where it is sampled and quantized. This data is then sent to the Zedboard via FMC where it will be deserialized. Once deserialized, the data will go to the PS of the Zedboard and lastly sent to the user computer through ethernet. The user will also issue commands to control the SPI components on the RF daughterboard.

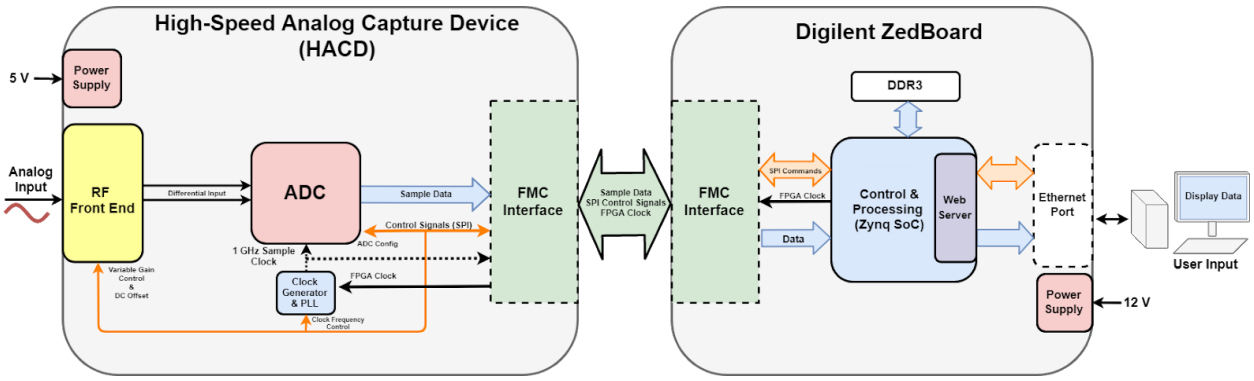
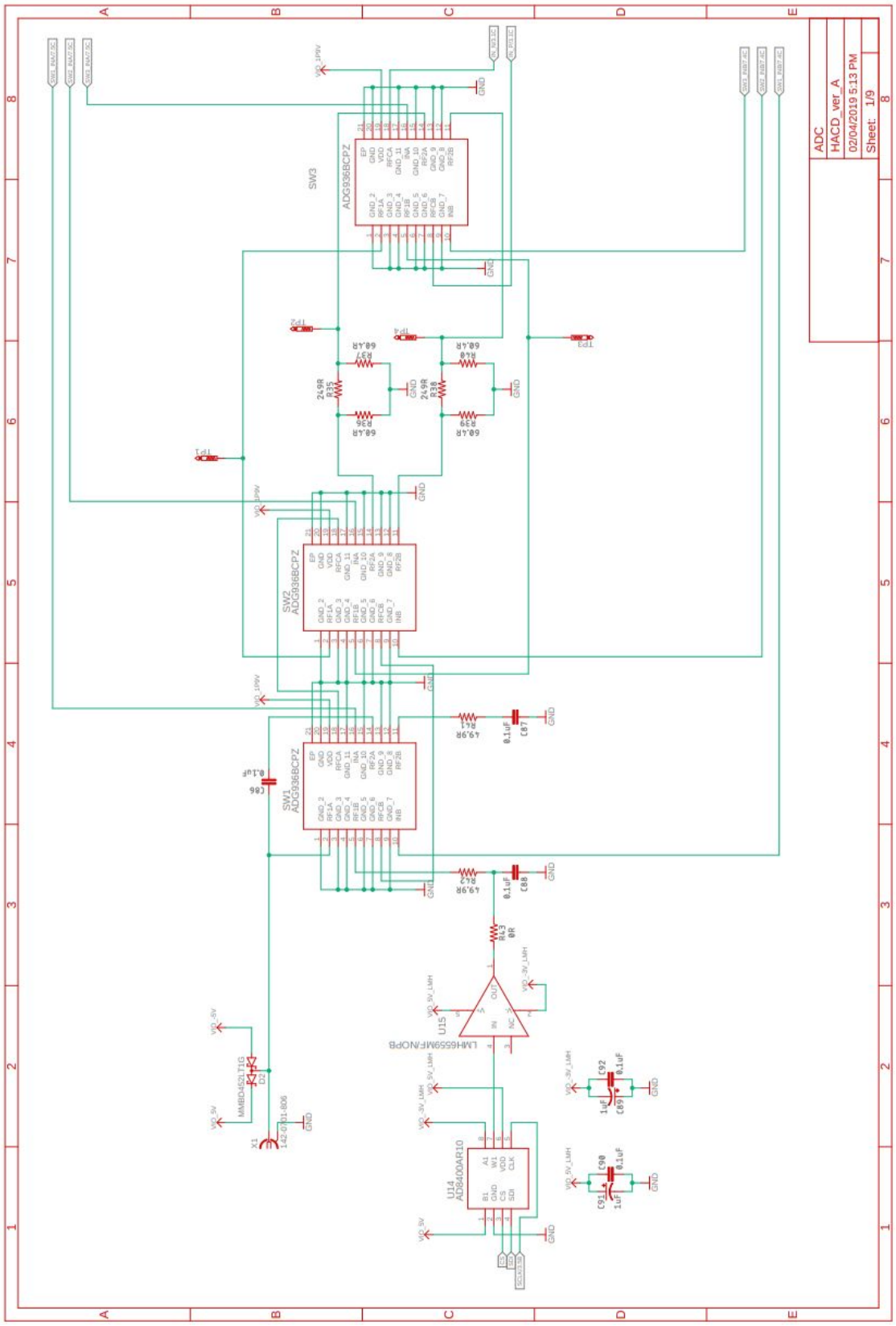
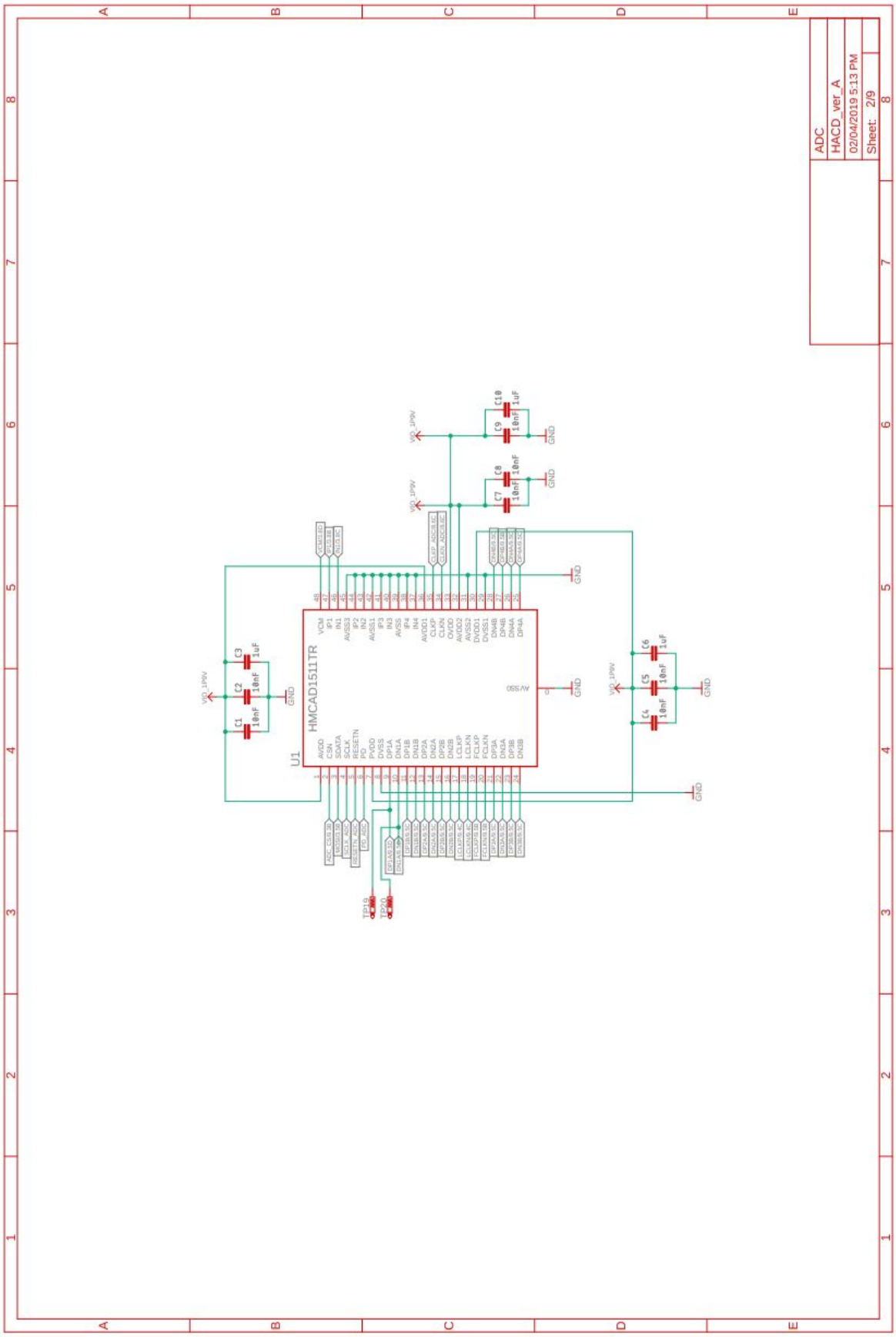
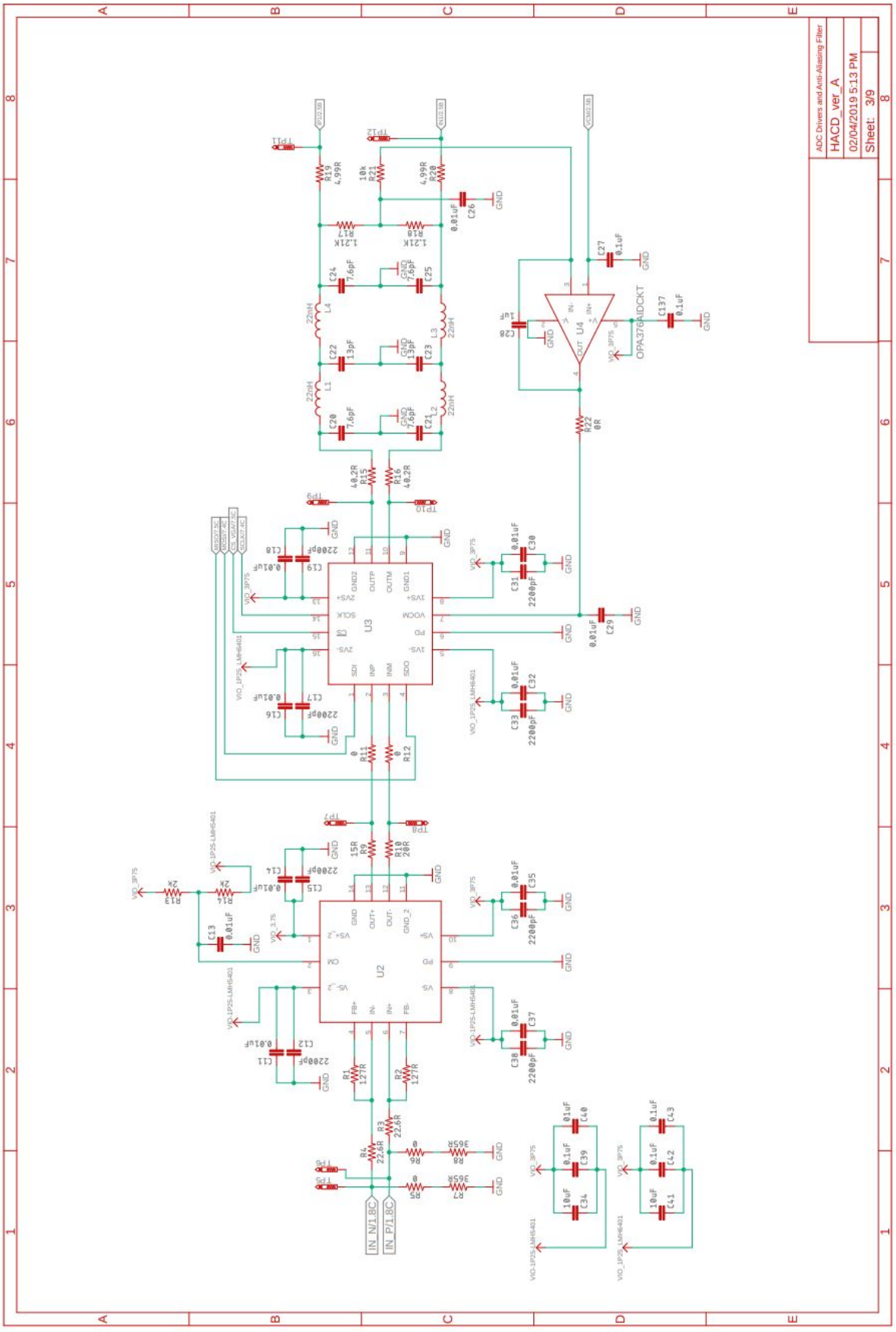


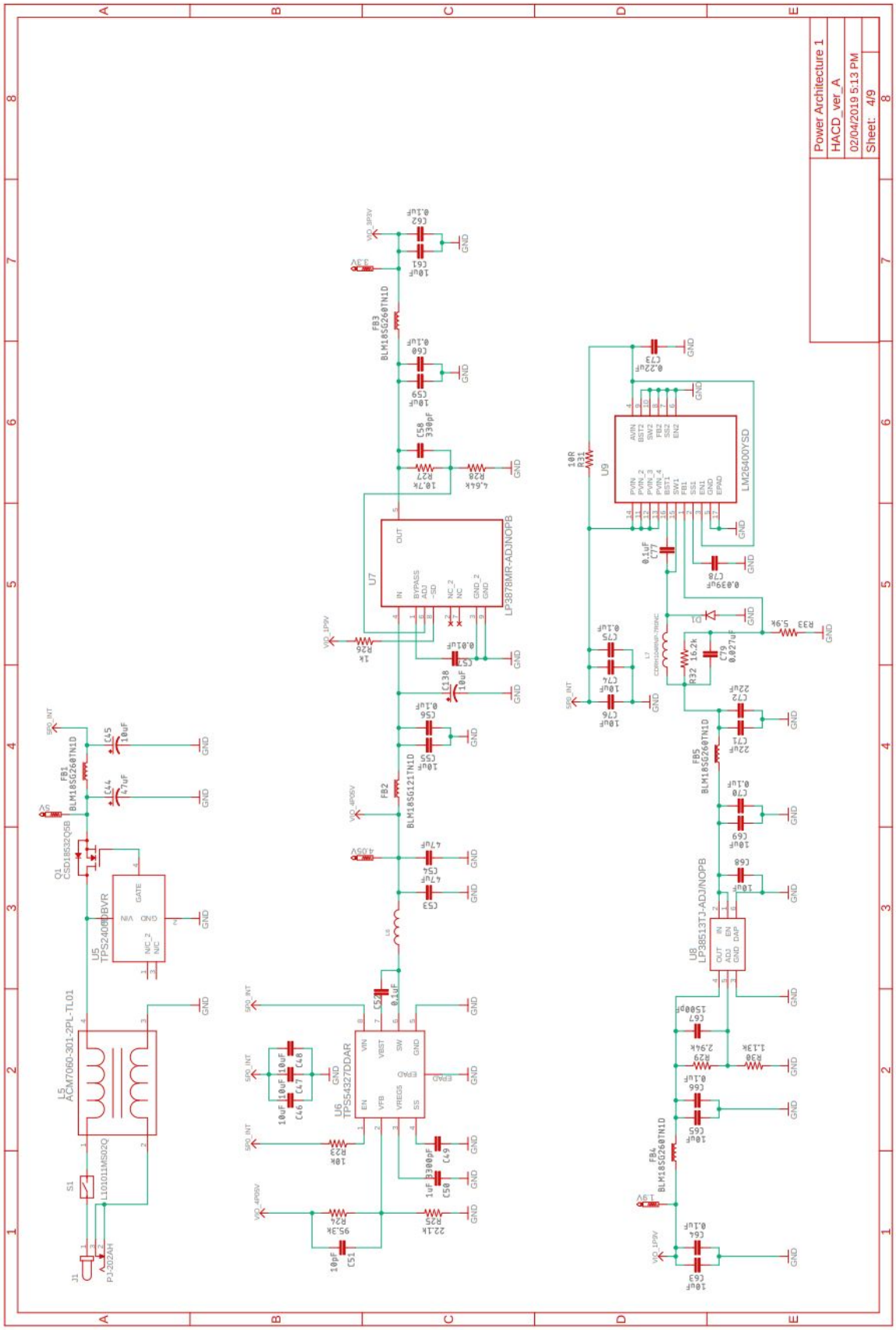
Figure 2.7: System Diagram

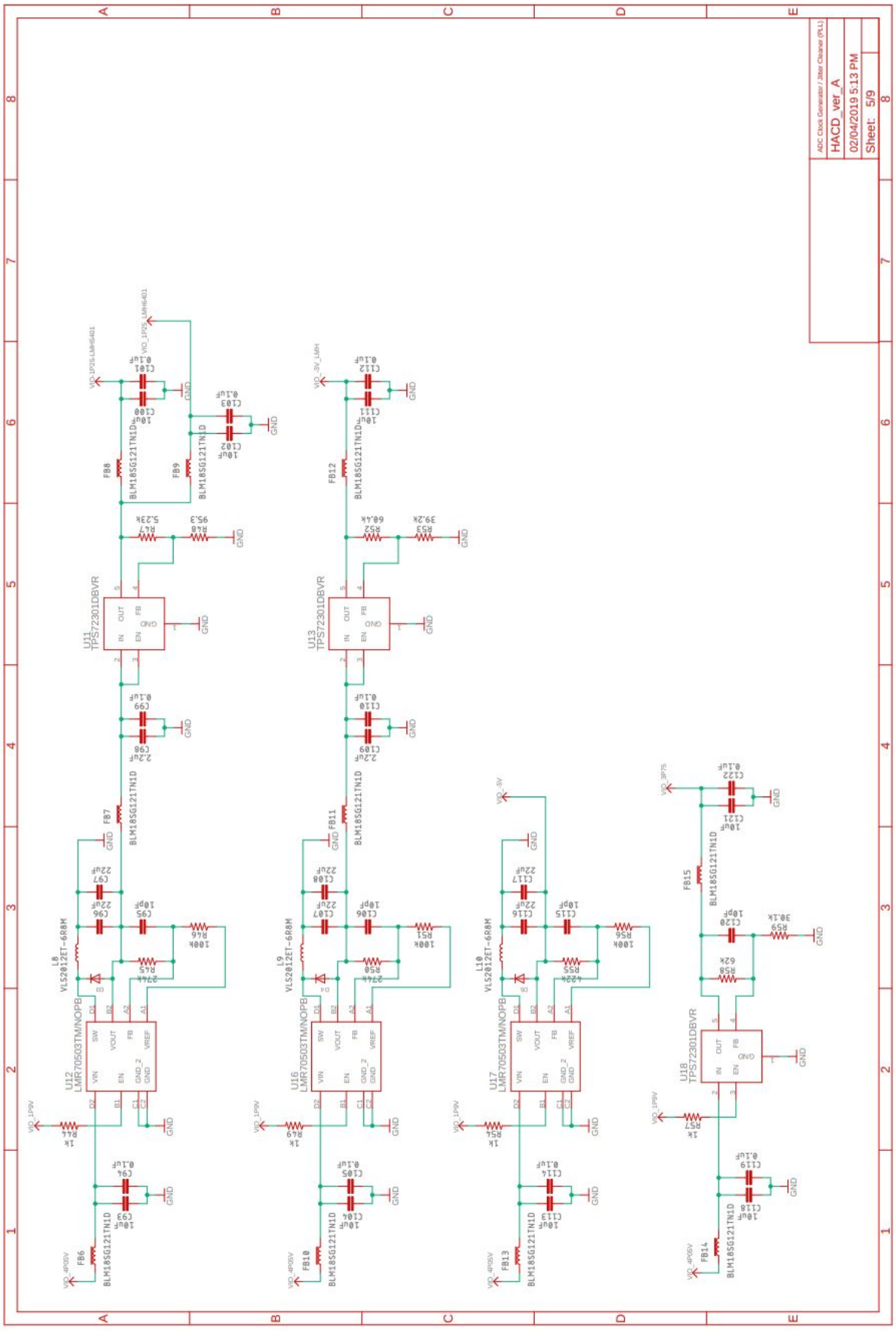
2.2 Schematics & PCB Layout

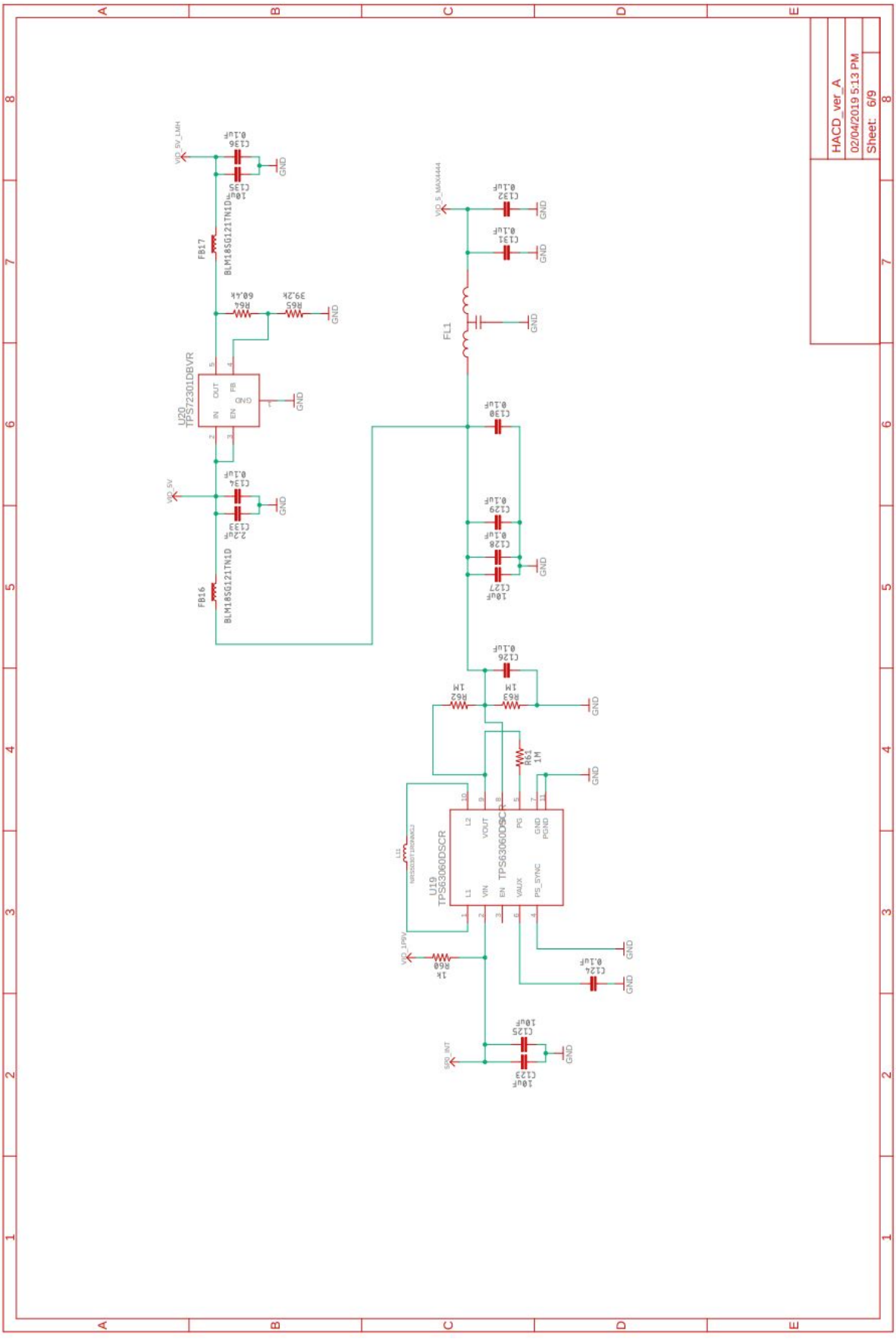


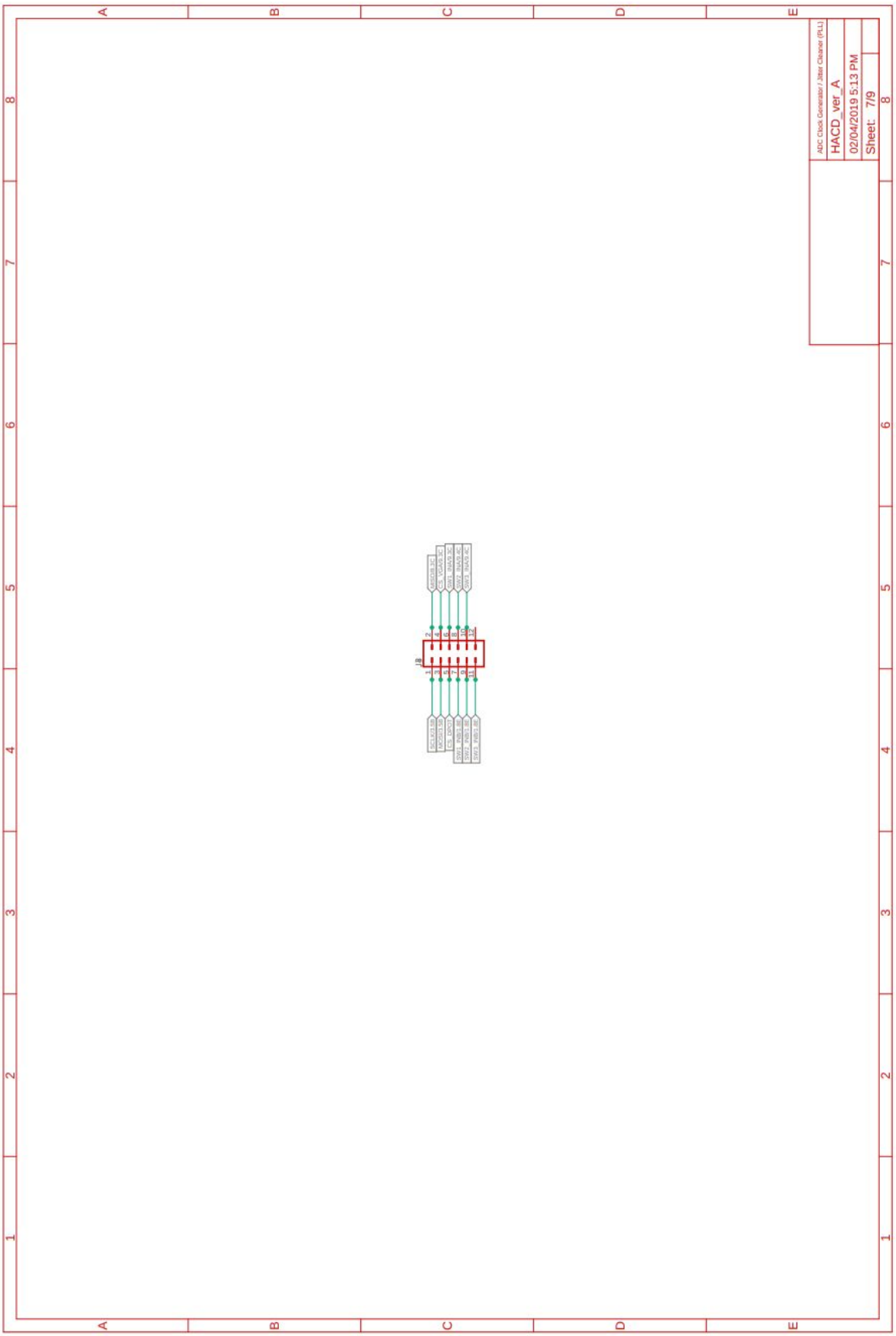

ADC
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 2/9
HMCAD1511TR
U1
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
10nF
1uF
GND
VIO_1P8V
TP19
TP20

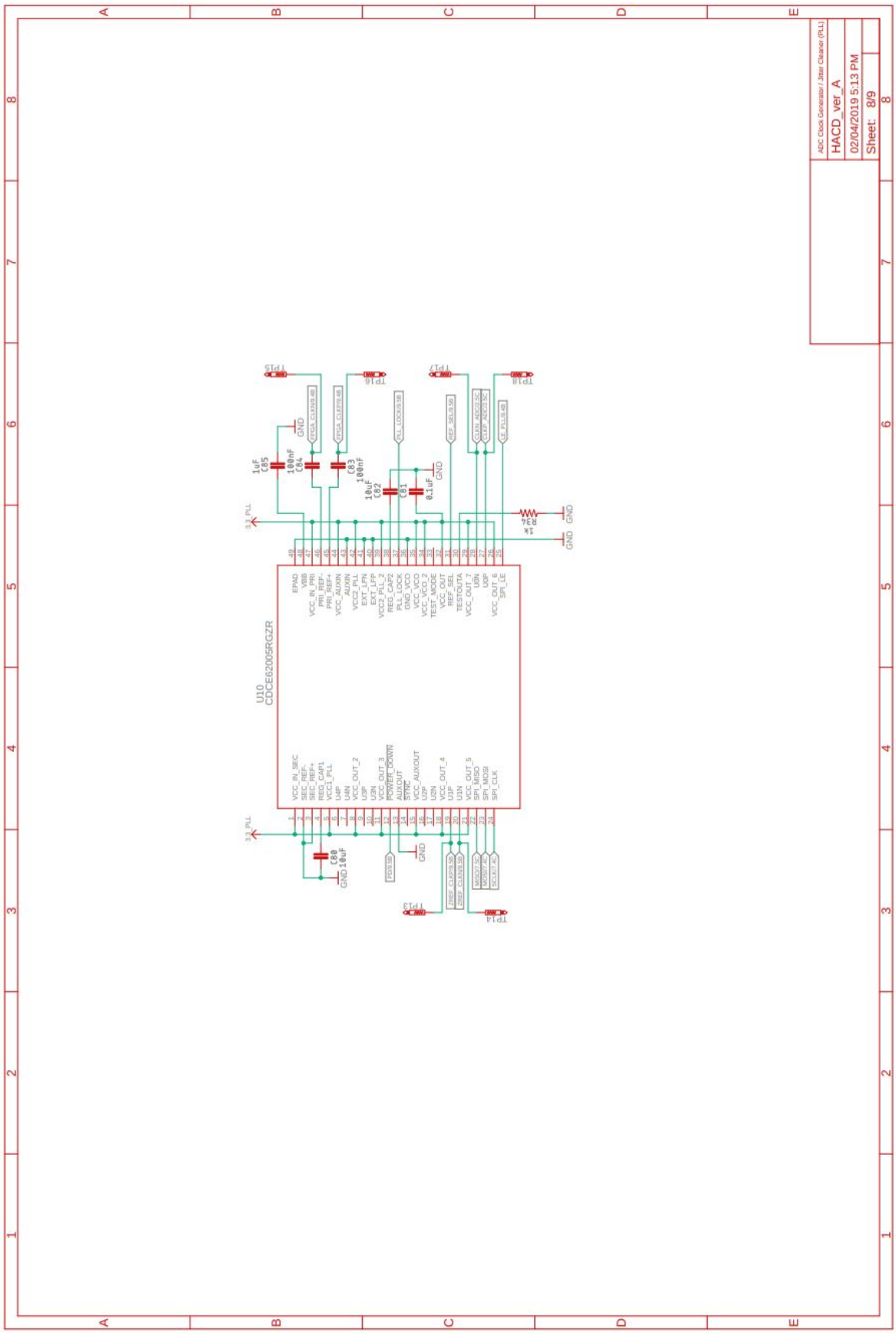

ADC Drivers and Anti-Aliasing Filter
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 3/9
U2
U3
U4
OPA376IDCKT
IN_N/1.8C
IN_P/1.8C

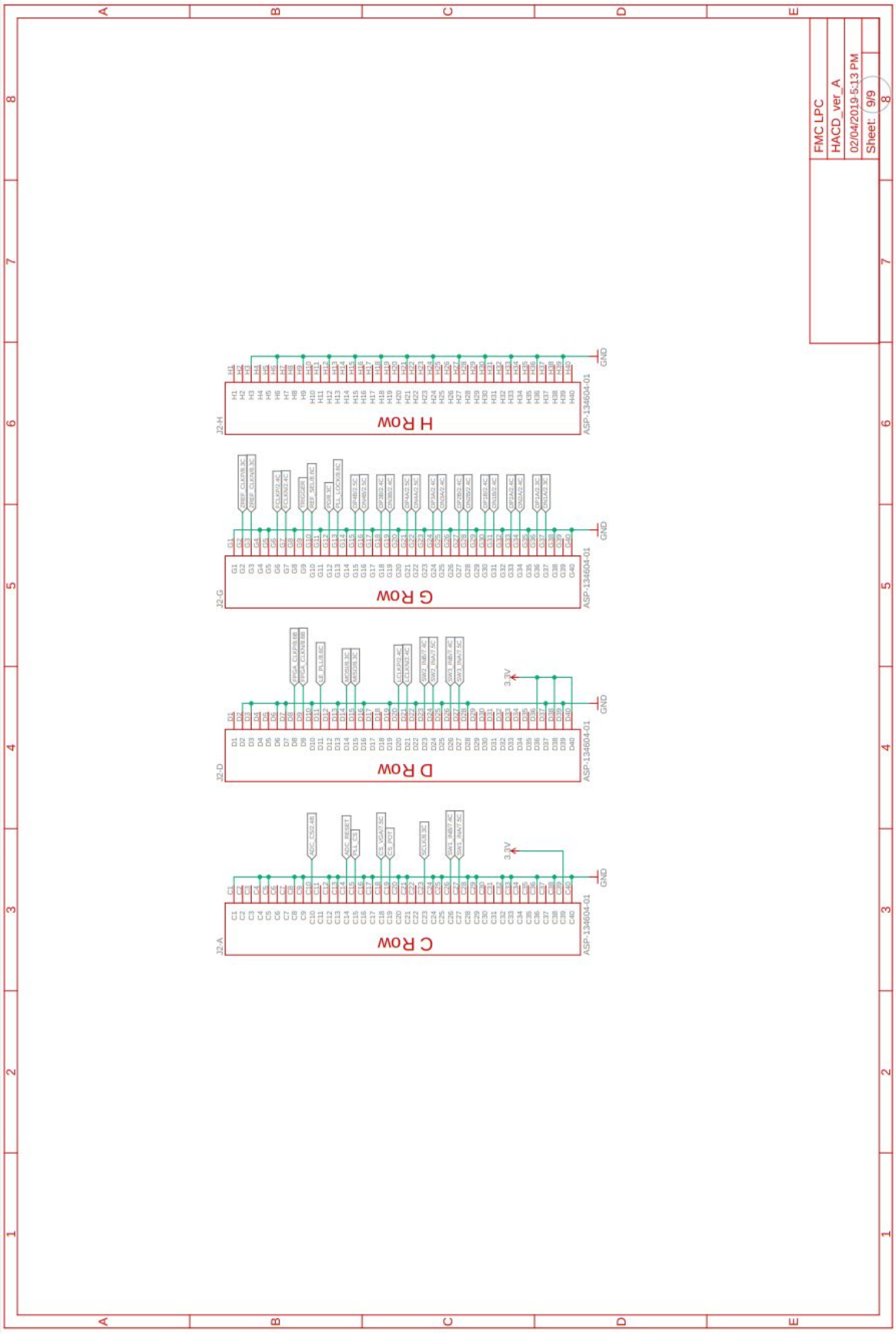

Power Architecture 1
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 4/9

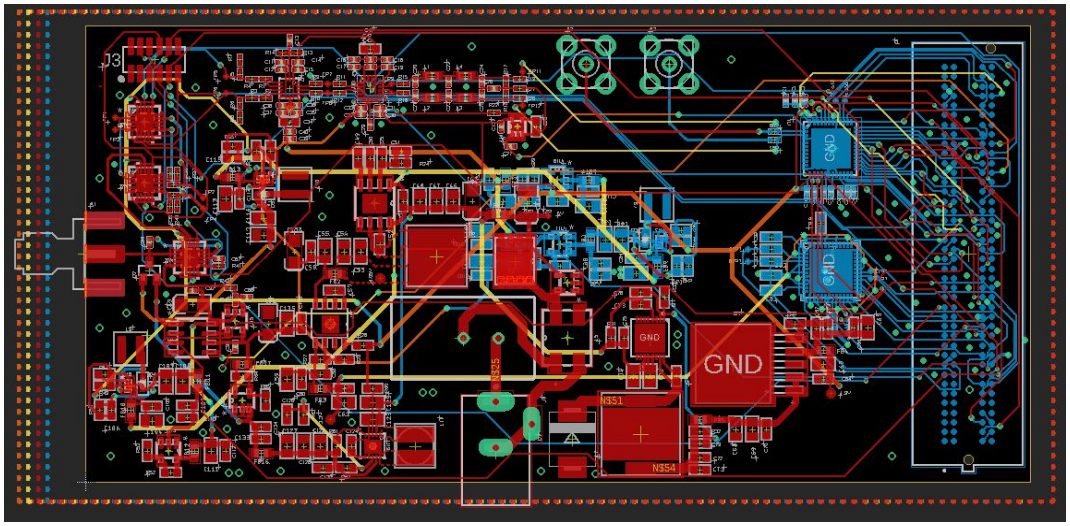

ADC Clock Generator/ Jitter Cleaner (PLL)
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 5/9

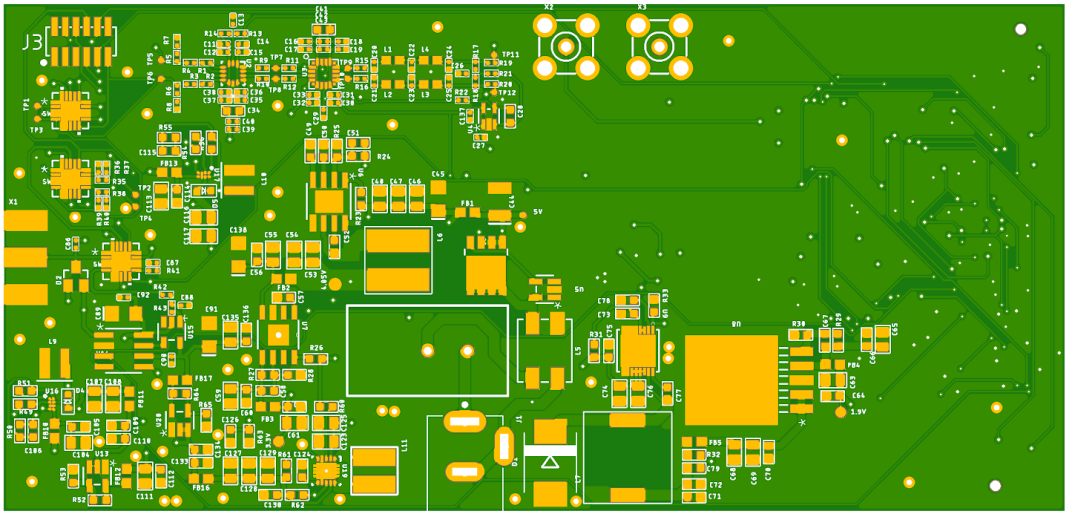

HACD_ver_A
02/04/2019 5:13 PM
Sheet: 6/9
U20
TPS72301DBVR
U19
TPS63060DSCR
FL1
FB17
FB16
BLM18SG121TN1D
GND


ADC Clock Generator / Jitter Cleaner (PLL)
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 7/9

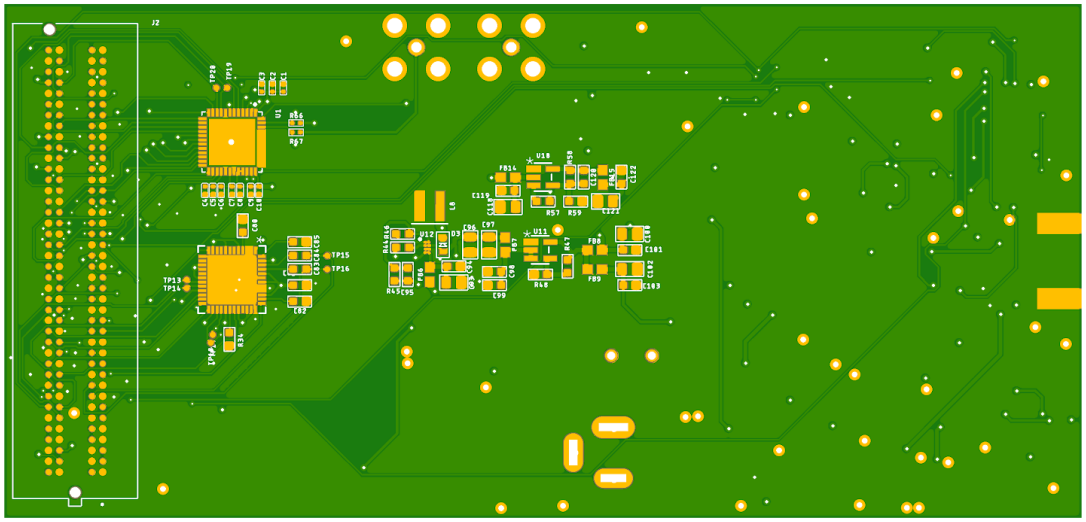

ADC Clock Generator / Jitter Cleaner (PLL)
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 8/9
U10
CDCE62005RGZR
C65 1uF
C84 100nF
C83 100nF
C82 10uF
C81 0.1uF
C80 10uF
R2 1k
3.3_PLL
GND


FMC LPC
HACD_ver_A
02/04/2019 5:13 PM
Sheet: 9/9
H Row
G Row
D Row
C Row
J2-H
J2-G
J2-D
J2-A
ASP-134604-01
GND
3.3V



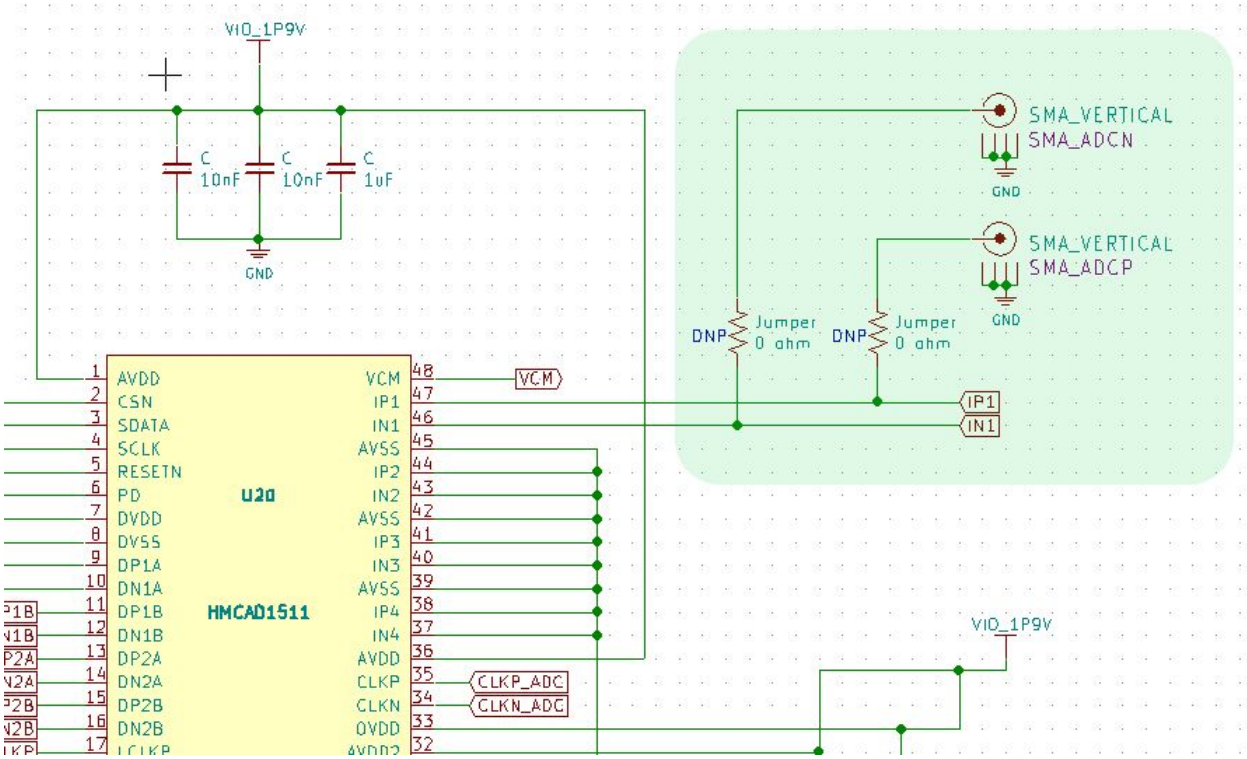


Top View

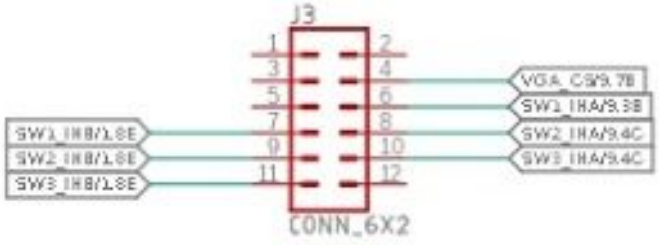


Bottom View

2.3 Design Changes



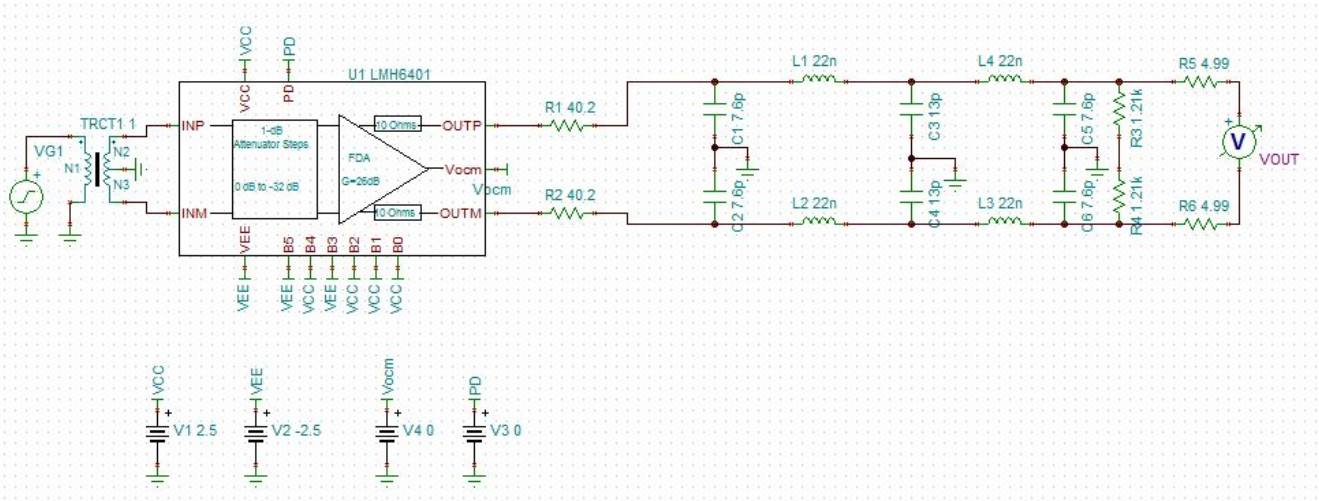
This change was added as a plan b in case the inputted waveform does not make it to the ADC. The two SMA connectors will allow a signal to be directly applied to the ADC from an external source. This gives the ability to still use the board and ADC if some error occurs with the signal in the circuitry prior.



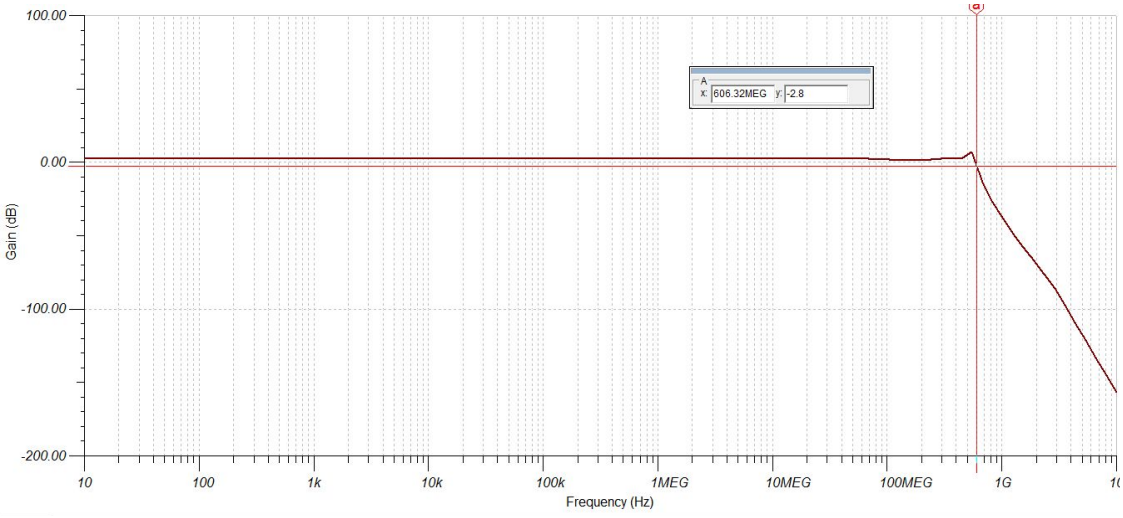
We also added a socket connector so that we could control the RF switches. These connections would interface with PMOD headers on the Zedboard. The switches would receive control signals from the FPGA and switch between the attenuator paths and the DC and AC coupling paths.

2.4 SPICE Simulations

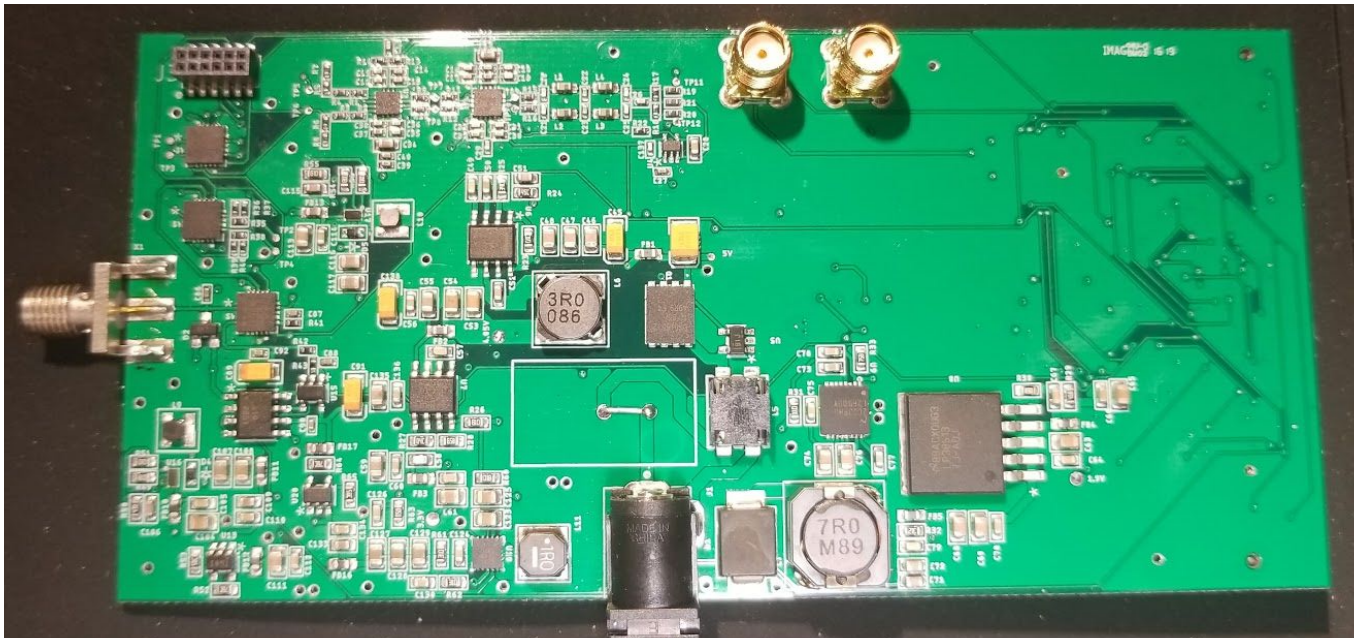
2.4.1 Anti-aliasing Chebyshev LPF



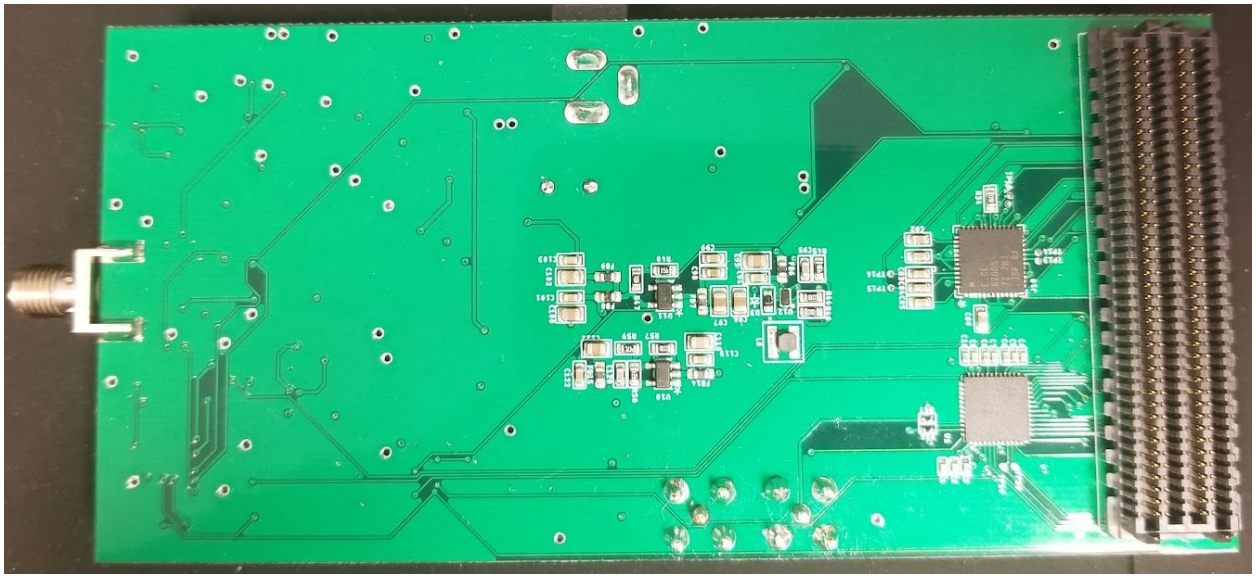
Gain vs. Frequency (LPF)



2.5 HACD PCB



PCB Top View



PCB Bottom View

3.0 Experimentation

3.1 Hardware Testing Plans:

3.1.1 RF front-end testing:

In this test we will input different waveform types from a function generator at 500 kHz, 1 MHz, 10 MHz, and 50 MHz. We will input a sine wave, square wave and sawtooth wave. We will use an oscilloscope to measure at the test points on our board. Test points are placed after each IC in the front end. In reference to the input signal we will be able to see how the different elements of the front end change the signal by attenuation, amplification and filtering. This test we also be verification that components in the front-end are powered on and operating correctly. From these tests we'll be able to verify the functionality of the following components the LNA, RF switches, DC offset digital potentiometer, passive attenuator, and VGA. Lastly, we will run a frequency sweep on the board and measure the response of the filter to confirm its cut-off frequency.

3.1.2 SPI Component testing:

Each of our SPI enabled components can be tested at their corresponding test points. We will measure the signals at the output of each of the SPI components and observe the changes made to the signal after programming the component to cycle through it's full range of settings. For examples we can test the DC offset control provided by the digital potentiometer by measuring with an oscilloscope at test points.

3.1.3 Power Testing:

These tests will involve measuring voltages at test points in the power architecture circuitry to verify that they are at the desired voltage levels. There are test points for 5V, 3.3 V, and 1.9V. To verify that the power circuitry is functioning as expected we need to confirm that all voltages are correct. If the voltages are significantly lower or higher than expected, it is likely that the system could be damaged or fail to properly function.

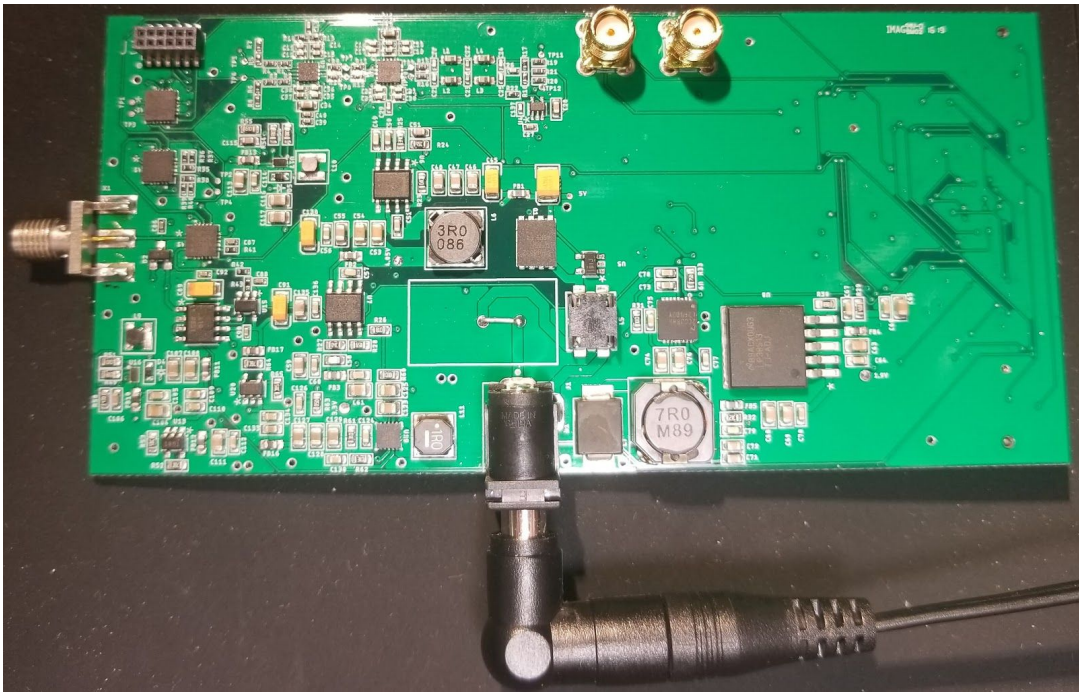


Figure 3.0: Power Testing Setup

3.2 Firmware Simulation & Testing:

3.2.1 Pynq Linux Port Test

In this test, an example Pynq overlay will be created with an AXI GPIO peripheral to control the onboard switches and LEDs. A script was created to flash the LEDs upon successful boot. This validates that the Pynq system can load a bit file onto the programmable layer of the FPGA, and validates that linux boots successfully. This test is based off of a built-in test on the Xilinx Pynq Z1 Board.

3.2.2 DMA Pipeline Test

In this test, a custom IP was created: HMCAD1511 Sample Generator. This IP mimics a portion of the functionality of the HMCAD1511. It produces a continuous ramp function on the output. This verifies the DMA pipeline and ADC-to-AXI converter. Since the sample generator produces predictable output, we can compare the expected results with the actual to verify the DMA pipeline.

3.2.3 LVDS Deserializer Hardware Test

In this test, we will use the HMCAD1511 evaluation board to send predictable data to the FPGA. This is similar to the test described in 3.2.2, but since the data arrives in a serialized, LVDS format, we can use this input to test the HMCAD1511 Deserializer IP. Again, since the input is a repeating pattern, we can compare the expected results

with the actual to verify the Deserializer. The ADC supports multiple test patterns: the SYNC pattern, which outputs 0xF0 along all channels, the DESKEW pattern, which outputs 0xAA along all channels, and the RAMP pattern which outputs the values of an 8-bit counter (0x00-0xFF) along all channels.

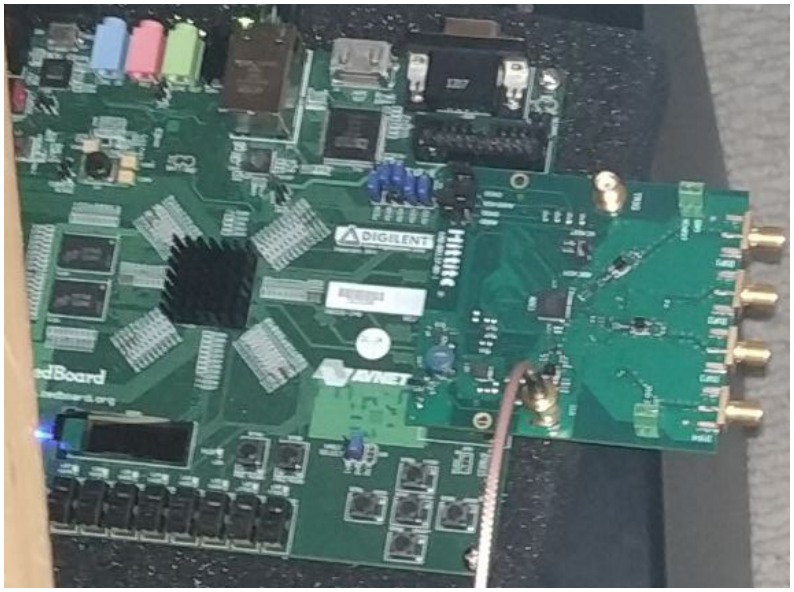


Figure 3.1: Eval board test

3.3 Full System Test:

For this test we plan to test a sinusoidal input with varying values on the SPI controlled devices. This assured that those components worked as expected and that the signal was behaving as expected. This also included testing an input with a DC offset and controlling the digital potentiometer to deal with the offset. Lastly, we conducted a frequency sweep to make sure that our system worked until a maximum of 500 MHz.

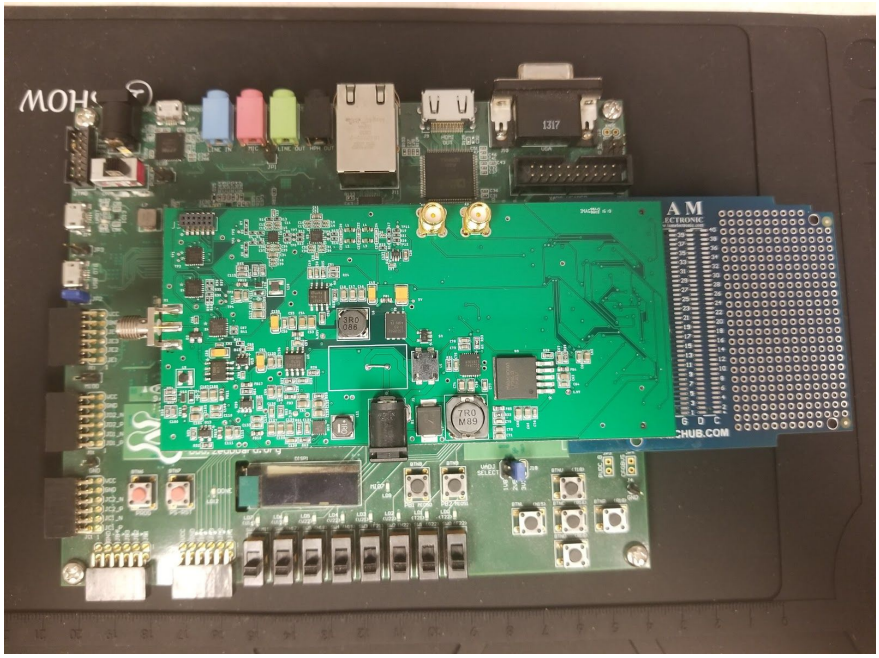


Figure 3.2:Full system setup

4.0 Experiment Validation

4.1 Firmware Simulations

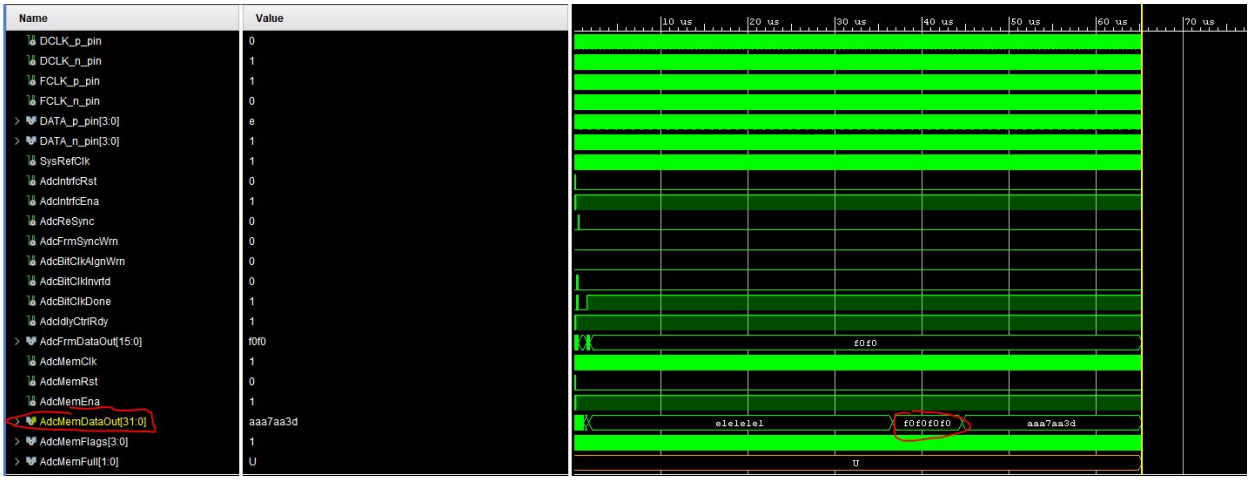


Figure 4.0: ADC LVDS Deserializer IP Simulation

Figure 4.0 and Figure 4.1 shows the output of the testbench created for the Deserialize IP. Initially, the deserializer reads a misaligned SYNC pattern across all channels (0xE1). When the SYNC command is given to the IP, the alignment state machines shift

the output to the correct sync pattern (0xF0). Once the user verifies that the Deserializer is synced to the ADC, the user can configure the ADC into normal operation mode.

Figure 4.1: Validating deserialization of predictable data in simulation

4.2 Easyboard Results

4.2.1 ILA Verification

Once the Deserializer IP was validated in simulation, experiment 3.2.3 was used to validate the IP in a live test. Xilinx Integrated Logic Analyzer IPs (ILAs) were used to verify the output of the system during the experiment. Figure 4.2 shows the result of the SYNC command in the ILA. 0xF0 across all channels is the expected result. Figure 4.3 shows the deserialized RAMP pattern measured in the ILA. This pattern counts from 0x00 to 0xFF. The measured ramp pattern is the expected output and was further validate manually by counting the output samples.

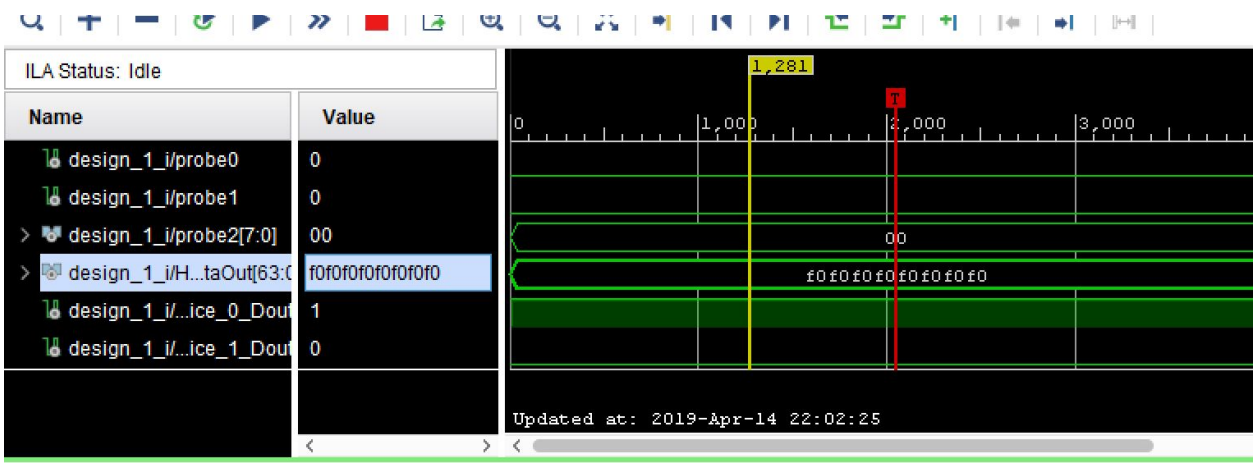


Figure 4.2

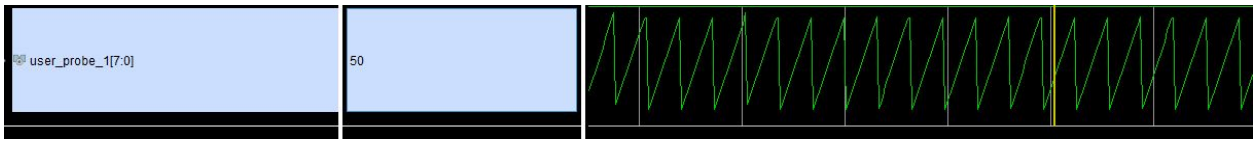


Figure 4.3

Resource	Utilization	Available	Utilization %
LUT	2912	53200	5.47
LUTRAM	242	17400	1.39
FF	4289	106400	4.03
BRAM	18	140	12.86
IO	25	200	12.50
BUFG	1	32	3.13

Figure 4.4: FPGA utilization reports

4.3 Jupyter Results

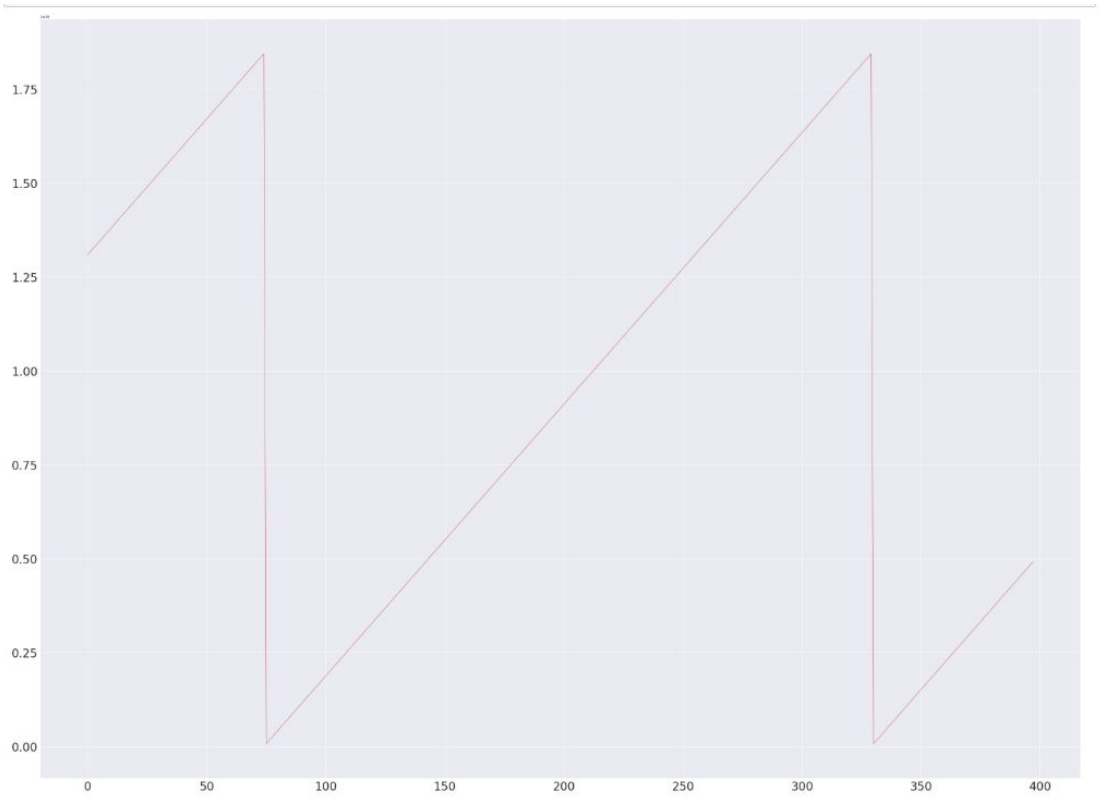


Figure 4.5: Ramp pattern output in Jupyter Notebook

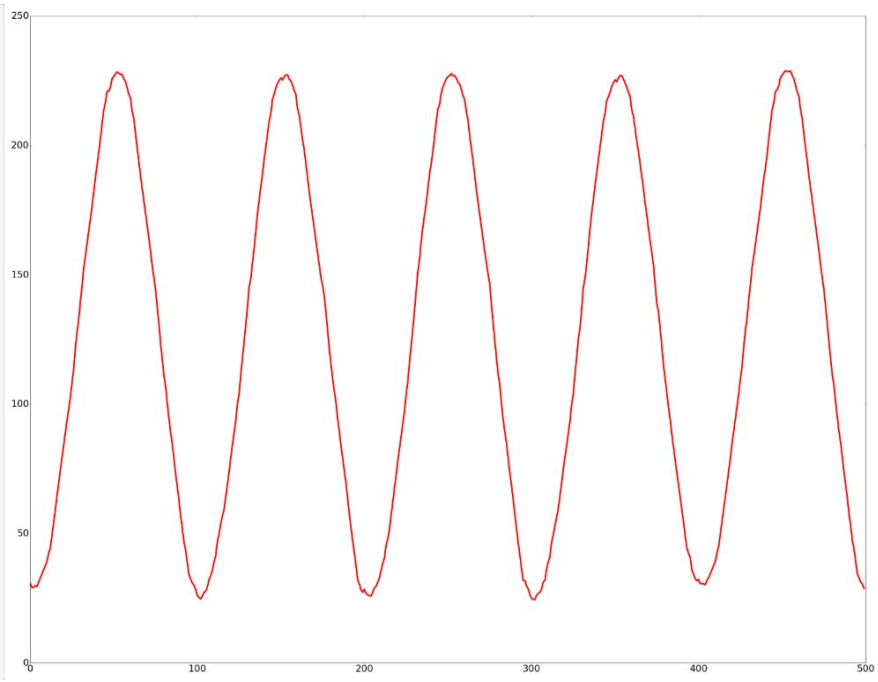


Figure 4.6: Measured 500-Khz sinewave in Jupyter

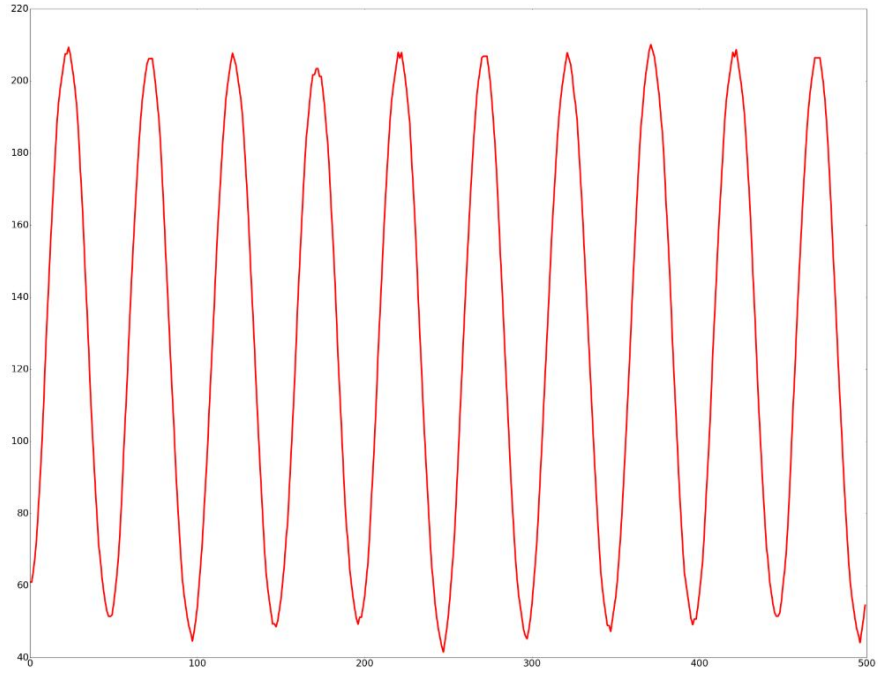


Figure 4.7: Measured 1-Mhz sine waveform in Jupyter

For these results, Experiment 3.3 was conducted. The Easyboard was used in place of the HACD board as the HACD board was unavailable at the time. Figure 4.5 shows the ramp pattern from the ADC. Figure 4.6 and Figure 4.7 shows the measured waveforms formatted in graphs created by Matplotlib within Jupyter Notebook. The measured data was filtered through a moving average function to remove random artifacts and to create nice graphs.

4.3 Voltage Tests

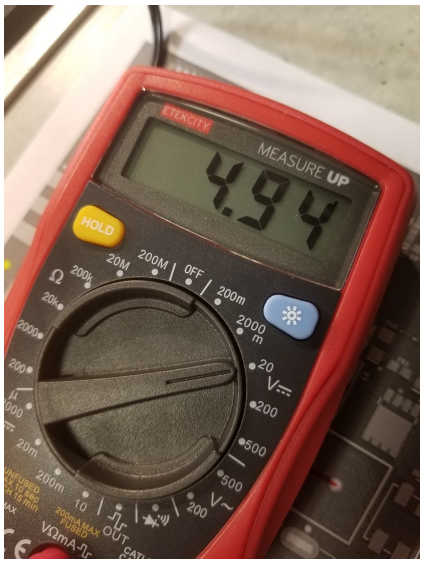


Figure 4.8: 5 V measurement

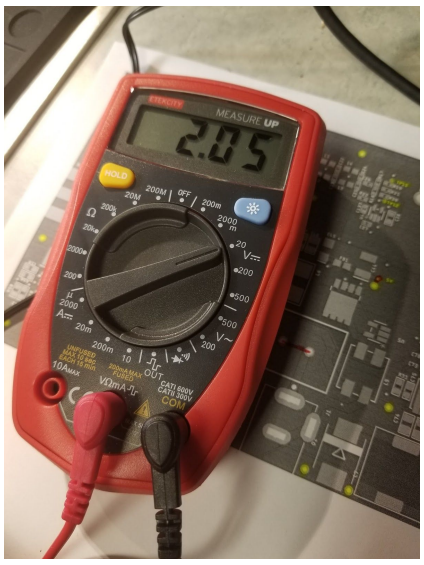


Figure 4.9: 1.9 V measurement

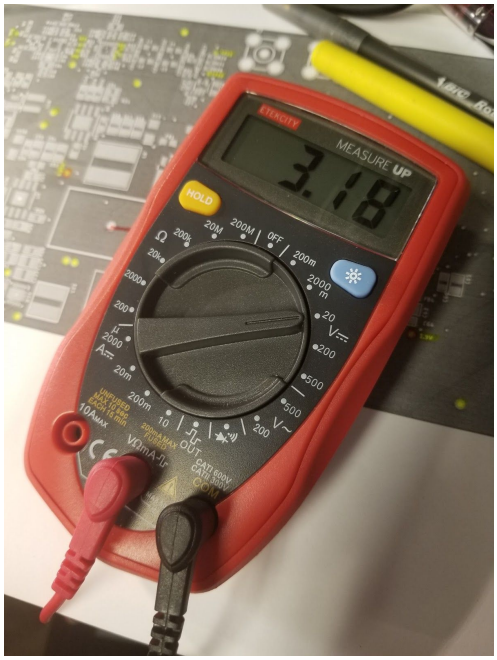


Figure 4.10: 3.3 V measurement

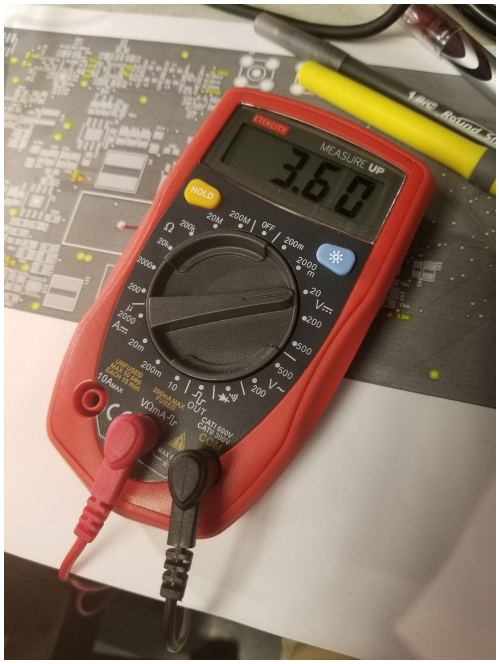


Figure 4.11: 4.05 V measurement

The results obtained from our power testing are shown in figures 4.8-4.11 above. Most of the voltages we obtained were correct, however the 4.05 V voltage was the farthest away from the expected value. After further investigation, it was found that one of the ICs we used in the power circuitry was mixed up in the layout process. This caused errors in other voltages. After some troubleshooting we were able to obtain more voltages that were correct. However, due to the wrong IC some voltages just were unable to be obtained. Due to this issue we were unable to run full system tests and could not achieve full system results. We hope to continue working on this project when we have time and get a full system test running.

5.0 Issues Encountered

5.1 Maintainability of Project

Our final design will be easily maintained. Due to their being only two main parts, there is not too much that will need to be maintained. The PCB will need to be cleaned every now and again to keep it operating as best as possible. This involves removing any dust and dirt that may have gotten on the PCB. This allows for easy maintainability for the hardware. The firmware and software are easily maintainable as they are stored with a git repository with revision history. The firmware and software come with documentation to help facilitate future work.

5.2 Retirement of Project

Once the project is done being used, it has the ability to be recycled. There are many companies that will recycle PCBs whether they are populated or not. This allows the resources to be reused. Also, due to our project not requiring batteries, none will be required to be thrown away. The FPGA would not be thrown away either since it could be used for many other applications. This allows our project to

Since our project was meant to be open source, it is easy to duplicate the project. People will have access to the schematics and the PCB layout and be able to recreate a new board if necessary. This also means that the project can be easily updated. The HACD board can serve as a basis for either a more complicated or more simplified design. Hobbyists and students who are interested in recreating the project could apply changes in areas they see fit. There are also plenty of ways this project could be built upon and thus increase the longevity of this project.

5.3 Encountered Issues & Future Technical Improvements

5.3.1 Hardware Issues

We encountered the following issues with our first revision of our PCB:

1. Missing heat vias for proper thermal regulation
    a. Some of the more power hungry ICs such as the PLL weren't properly laid out on the pcb. Some of them required thermal vias to help with cooling. This is why some of the ICs become very hot when testing. Overall more care and research should have been put into the thermal regulation of the various components on the pcb.
2. FMC connectors orientation in pcb layout is backwards
3. Power design should have been verified in simulation
4. Should have had separate analog and digital ground planes
5. Problem with the power design

    There was a mistake made with the power design that may have damaged our pcb, and impeded us from conducting further tests. The mistake was identified when we noticed that the original in our original schematics done in KiCAD for the 3.75 voltage we used the TPS79301DBVR, which is the correct voltage regulator.

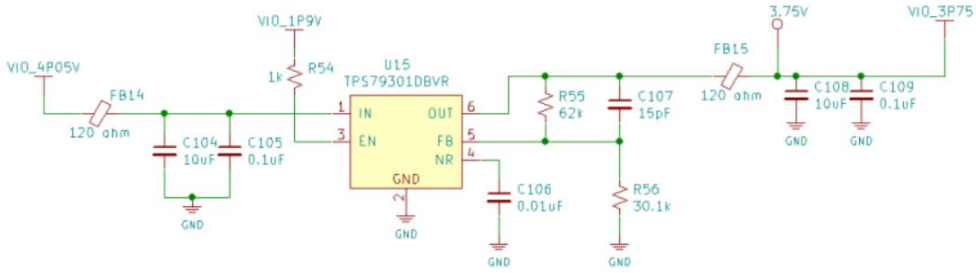


Figure 5.0

Then when we compared the schematic done in EAGLE done by the pcb layout freelancer we noticed he didn't use that voltage regulator.

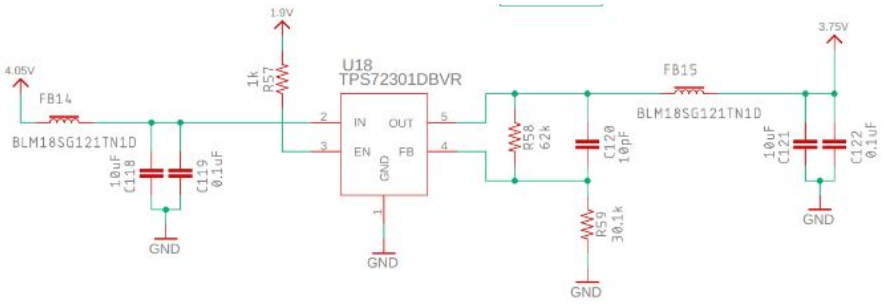


Figure 5.1

He used the TPS72301DBVR, which is the one used for the -1.25 voltage. This was the mistake made. This mistake may have been the cause of another

unexplained problem where the digipot was pulling excessively high amounts of current and heating up.

5.3.2 Firmware Improvements

1. The ADC LVDS Deserializer IP could be further optimized to better handle high speed data rates.
    a. Currently, the deserializer design utilizes static calibration to align the bit clock DCO. At full speed, the DCO operates at 500-Mhz. In an ideal case, this gives a sampling window of 2 ns, however, in real applications, this window is reduced to < 2ns due to clock jitter, PCB trace length and clock skew/uncertainty. The < 2ns sample window is too small to capture with static calibration, so dynamic calibration must be used. Dynamic calibration was attempted within the Bit Clock Alignment block in Figure 2.5, but was only validated in simulation. We were unable to validate this experimentally with the Easyboard as we did not have access to a low jitter high speed clock generator required by the ADC.
2. Digital Hardware Filtering
    a. Currently, the input data is filtered within software using algorithms written in Python. These could be translated into a digital hardware block so input data could be filtered in real time instead of in post processing. Real time streaming of measured data was outside the scope of this project.
3. Software library improvements
    a. Currently, Matplotlib is used to view the measured waveforms. This library is very slow and a more efficient one could be utilized for real time streaming.

6.0 Administrative Details

6.1 Project Progress

Over the course of the project we had been effectively utilizing the time of each group member which was how we had remained on track. The project was on schedule until about late March after running into overlooked issues that arose from design changes made in late February. The extra research time and effort trying to implement the design changes set the project back by 3 to 4 weeks. The design change was intended to cut costs and simplify our design by utilizing the evaluation board for our ADC, which would allow us to remove the ADC and several other expensive and complicated components. We proceeded with this design for several weeks before realizing major flaws with it. So all of the work done for this design including the

schematics for a separate clock generator pcb were no longer useful and became lost time. After this we decided to return to our original design which included the ADC on our pcb. This was the primary delay we experienced and is the reason we didn't have a pcb manufactured by the date we initially set as our goal. If we would have continued with our original design we would have likely been on schedule for pcb manufacturing the second semester.

6.2 Cost Breakdown and Discussion

Due to the problem we were trying to solve, our goal for this project was to keep it relatively low cost so that hobbyists or students could recreate the design. Due to the high ratings needed of the ICs on the PCB, they were the most expensive part of the design. The total cost of components in our BOM is about $220 and the total cost of manufacturing our PCB would be around $70. However, the manufacturing may be done at a cheaper cost depending on the company. Due to lack of time, we got ours manufactured and all of the components soldered on in the process. This is not necessary if somebody has more time and wants to solder each component to the board. Getting a PCB just manufactured without the components on is much cheaper. This makes the project perfect for hobbyists or students who have more time and are not under a time constraint.

Table 2: Cost of 1 Unit

Cost of 1 unit	
Item	**Cost**
Parts	$220
PCB	$70
Total	**$290**

Table 3: Cost per unit for 1000 boards

Unit Cost per 1000	
Item	**Cost**
Parts	$116.52
PCB	$5.70
Total	**$122.22**

6.3 Man-hours Breakdown

Each of our members dedicated many hours to produce the best solution. Every weekend was spent meeting as a group and researching the project and brainstorming ways to progress. These meetings lasted an average of about 9 hours on both Saturdays and Sundays. This is when we made the most progress during the project as we could all put our heads together and work as a team. During the weekdays, each member put in quite a few hours of their own time in as well. There was also an hour set aside every Thursday to meet with Dr. Kaps and discuss the progress we had made in the past week and how he would want us to progress. A more detailed breakdown of the man-hours is provided in Table 4 below.

Table 4: Man-hours Breakdown

Group Member	**Total Man-hours Worked**
Ryan	410
Alex	415
Rishub	405

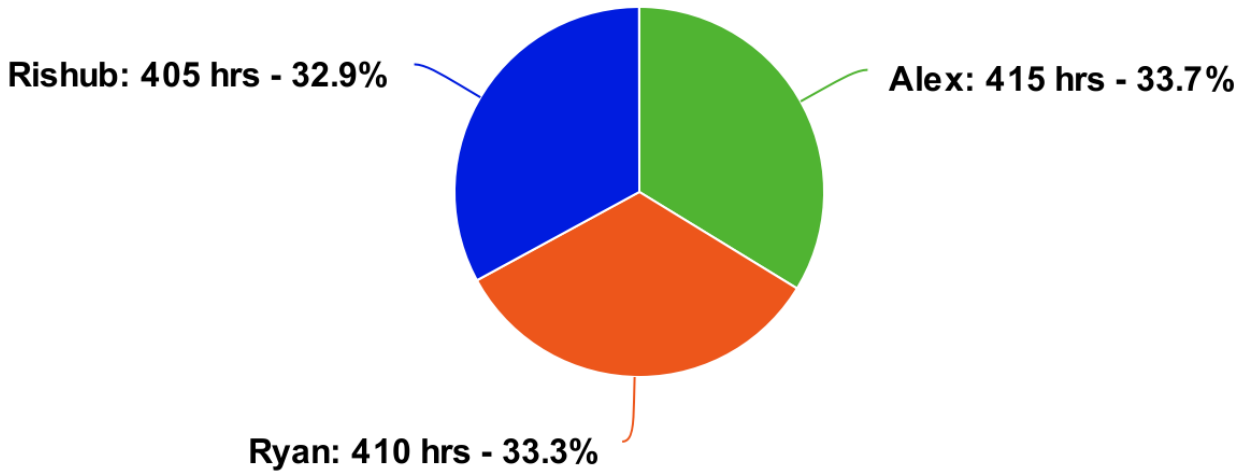


Figure 6.0: Breakdown of total man hours spent

7.0 Lessons Learned

7.1 Knowledge and Skills Learned

Going into this project there was a lot of things that were new to each of us. We were able to work and overcome a majority of these issues during the span of this project. There was especially a huge knowledge gap for the RF daughterboard front-end of the project. Each member left this project with a greater knowledge of the topics related to the project and gained certain skills that will be useful in the future.

For the RF front-end we learned how to properly design a full schematic and how to work with high frequency signals. Through this process we needed to familiarize ourselves with specific concepts of signals and also PCB design. One of the first things we needed to go over was the maximum frequency we could input due to our sampling rate. This was directly applying the Nyquist rate to our project and how we limited our frequencies to 500 MHz. We also wanted to limit the amount of signal reflection we would be getting throughout our circuit. This involved making sure that we impedance match across the circuit since we wanted to minimize the reflection. We also needed to familiarize ourselves with working with our specific ADC. The circuitry was required to work with the ADC that we chose for our implementation.

For the firmware backend, we learned how to design a functional system using the Vivado IP Integrator. We learned how to create our own custom IPs and to interface with the AXI protocol. We also learned about the intricacies of how an ADC works and

how deserializers operate. In addition, we learned about the inner details of an FPGA and how to optimize the placement and routing of a design for maximum speed.

There were many lessons that we learned during this project. One of the issues we ran into was not having backup plans for our project. We had one design which was not ideal in case it didn't work. Another oversight was not simulating our power circuitry. While we were confident in the design, it would have been beneficial to make sure the circuit worked in simulation. There were also times during this project where our meetings were not as structured as they should have been. This caused some communication errors and caused some meetings to not be as productive as they should have been. Throughout the design process we also realized that we needed to research hardware compatibility much more thoroughly. While we mostly researched it well, there were a few things we missed that caused some issues later on. Maintaining a certain level of transparency about what each member of a team is working on is important for ensuring that time isn't wasted and knowing whether more time is needed for certain tasks. At times we could have been better at communicating what the current status of certain tasks were. We consulted many online forums for answers to technical questions we didn't understand. Placing too much trust in this anonymous so called experts can lead you astray. So only take advice from online forums with a grain of salt. Lastly we learned something that should be avoided for any engineering project, and that is how major design changes should be avoided in the later stages of a project. If you already have a solution that will work don't insert new changes that require lots of time to implement, especially when time is one of your most valuable resources.

7.2 Team Experience

Throughout this project we were able to gain great team experience and a great sense of how an engineering team should function. We were able to allocate each of the tasks we had effectively and did not have any issues as a team. We made sure to meet regularly so that while we were working on different parts of the project, we could all still stay on the same page. This effective communication is what allowed us to stay mostly on track with our schedule. Each member was also open to accepting extra work if needed by the other members and it felt as if everybody was pulling equal weight. Since we were a smaller group compared to most other groups, allocating the work efficiently and being able to trust each group member was extremely important. Through this project, we learned to that everybody would be able to get their work done by the time we needed it to be done by. Overall this was a great environment for the project and our team was able to work effectively and efficiently together.

8.0 References

[1] “OPENADC Product Datasheet”, OpenADC datasheet.
[2] “*OpenADC Project*.” *Assembla*. (n.d.). [Online]. Available: https://app.assembla.com/spaces/openadc/wiki/Home [Accessed Nov. 29, 2018].
[3] Analog Devices, “High Speed Multi-Mode 8-Bit 30 MSPS to 1 GSPS A/D Converter,” HMCAD1511 datasheet.
[4] Hittite, “Hardware User Guide, EVAL01-HMCAD1520,” 584-EKIT01-HMCAD1511 datasheet.
[5] Texas Instruments, “8-GHz, Low-Noise, Low-Power, Fully-Differential Amplifier,” LMH5401 datasheet.
[6] Texas Instruments, “DC to 4.5 GHz, Fully-Differential, Digital Variable-Gain Amplifier,” LMH6401 datasheet.
[7] Texas Instruments, “50-Ohm 2-GHz Oscilloscope Front-end Reference Design,” TIDA-00826 Design Guide.
[8] Texas Instruments, “3:5 Clock Generator, Jitter Cleaner with Integrated Dual VCOs ,” CDCE62005 datasheet.
[9] Texas Instruments, “Low-Noise, Low Quiescent Current, Precision Operational Amplifier e-trim Series,” OPA376 datasheet.
[10] Texas Instruments, “High-Speed, Closed-Loop Buffer,” LMH6559 datasheet.
[11] Analog Devices, “Wideband 4 GHz, 36 dB Isolation at 1 GHz, CMOS, 1.65 V to 2.75 V, Dual SPDT,” ADG936 datasheet.
[12] Analog Devices, “1-/2-/4-Channel Digital Potentiometers,” AD8400 datasheet.
[13] Texas Instruments, “Overvoltage Protection Controller,” TPS2400 datasheet.
[14] Texas Instruments, “60-V N-Channel NexFET™ Power MOSFETs,” CSD18532Q5B datasheet.
[15] CUI Inc, “DC Power Jack,” PJ-202AH datasheet.
[16] Texas Instruments, “3-A Output Single Synchronous Step-Down Switcher With Integrated FET,” TPS54327 datasheet.
[17] Texas Instruments, “Micropower 800-mA Low-Noise "Ceramic Stable" Adjustable Voltage Regulator for 1-V to 5-V Applications,” LP3878-ADJ datasheet.
[18] Texas Instruments, “Dual 2-A, 500-kHz Wide Input Range Buck Regulator,” LM26400Y datasheet.
[19] Texas Instruments, “3A Fast-Transient Response Adjustable Low-Dropout Linear Voltage Regulator,” LP38513-ADJ datasheet.
[20] Texas Instruments, “SIMPLE SWITCHER® Buck-Boost Converter For Negative Output Voltage in µSMD,” LMR70503 datasheet.
[21] Texas Instruments, “200mA Low-Noise, High-PSRR Negative Output Low-Dropout Linear Regulators,” TPS72301DBVR datasheet.

[22] Texas Instruments, “High Input Voltage, Buck-Boost Converter With 2-A Switch Current,” TPS63060DSC datasheet.
[23] Texas Instruments, “Capacitor-Free, NMOS, 250-mA Low-Dropout Regulator With Reverse Current Protection,” TPS73201DBVR datasheet.
[24] ON Semiconductor, “Dual Schottky Barrier Diode,” D-MMBD352WT1 datasheet.
[25] ON Semiconductor, “Surface Mount Schottky Power Rectifier,” MBRS540T3G datasheet.
[26] C&K, “L Series Miniature Slide Switches,” L101011MS02Q datasheet.
[27] TDK, “Common mode filters For power line ACM series,” ACM7060-301-2PL-TL datasheet.
[29] Texas Instruments, “Low Phase Noise Clock Evaluation Module — up to 1.5 GHz,” CDCE62005EVM User’s Guide.
[30] “Digital potentiometer,” Resistor Guide. [Online]. Available: http://www.resistorguide.com/digital-potentiometer/. [Accessed: 29-Nov-2018].
[31] “Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing,” NI, 2016. [Online]. Available: ftp://ftp.ni.com/evaluation/pxi/Acquiring_Analog_Signal.pdf.
[32] “Data Acquisition Handbook,” Measurement Computing. [Online]. Available: https://www.mccdaq.com/handbook/chapt_2.aspx. [Accessed: 25-Nov-2018].
[33] J.-P. Busson, “Avoid Operational Amplifier Output Stage Saturation by Gain Control,” Texas Instruments, rep., 1999.
[34] J. Alexander, “Digitizing Oscilloscope Basics,” presented at UBC, April 23, 2003. [Online]. Available: https://www.ece.ubc.ca/~robertor/Links_files/Files/oscilloscopes_basics.pdf.
[35] C. Pearson, “High-Speed, Analog-to-Digital Converter Basics,” Texas Instruments, rep., 2011.
[36] AVNET, “Zynq™ Evaluation and Development Hardware User’s Guide,” Zedboard datasheet, August 1, 2012
[37] AVNET, “(Zynq™ Evaluation and Development) Hardware User’s Guide,” Zedboard User’s Guide, January 2014.
[38] L. Crockett, R. Elliot, M. Enderwitz, and R. Stewart, *The Zynq Book*. 1st ed. Glasgow, Scotland: Strathclyde Academic Media, 2014.
[39] Xilinx, “Zynq-7000 SoC Technical Reference Manual.” Zynq-7000 reference manual.
[40] Xilinx, “Clocking Wizard v5.3 LogiCORE IP Product Guide,” Clocking Wizard v5.3 product guide.
[41] M. Defossez, “XAPP 524: Serial LVDS High-Speed ADC Interface,” https://www.xilinx.com, 20-Nov-2012. [Online]. Available:

https://www.xilinx.com/support/documentation/application_notes/xapp524-serial-lvds-adc-interface.pdf.
[42] “XAPP 1017: LVDS Source Synchronous DDR Deserialization,” Xilinx.com, 22-Jul-2016. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp1017-lvds-ddr-deserial.pdf.
[43] “7 Series FPGAs SelectIO Resources,” xilinx.com, 08-May-2018. [Online]. Available:https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf.

Appendix A

Engineering Project Proposal

High-Speed Analog Capture Device (HACD)

Team
Alex Wozneak
Rishub Nagpal
Ryan Meruvia

Faculty Supervisor
Dr. Jens-Peter Kaps

Class Instructor
Dr. Peter Pachowicz

Oct. 5th 2018

1.0 Table of Contents

2.0 Executive Summary

The open-source OpenADC project gives users a device that can be used for a variety of analog capture applications one of which is an oscilloscope. An important task in computer security known as side-channel analysis can be performed with a oscilloscope, where the oscilloscope is used to measure things such as electromagnetic emissions (EM) from a computer system to determine sensitive information about the systems operation. With this being said, the OpenADC implementation is not suitable for capturing high frequency (min of around 400 MHz) EM signals needed for side-channel analysis. This is due to its low pre-amplifier bandwidth of 40 MHz and it's limited analog to digital conversion sampling rate of only 105 MSPS. We intend to enhance the design of this pre existing project so that a higher bandwidth and sample rate are achieved that would allow its use for EM side-channel analysis. Throughout the following we will discuss our proposed solution,the High-Speed Analog Capture Device (HACD). Working with Dr. Jens-Peter Kaps as our faculty supervisor, our team is tasked with enhancing the OpenADC project and meeting the following requested design criteria.

After meeting with Dr. Kaps and analyzing the previous project we were able to divide the project into two main sections, the analog front-end and the FPGA back-end. Each of these sections has a set of requirements that must be met. Firstly, for the analog front end:

1. *Low Noise Amplifier (LNA)*: Must achieve 400 MHz bandwidth and have variable gain
2. *Analog to Digital Converter (ADC)*: Must achieve a sample rate of 1 GSPS and have adjustable reference voltage
3. *Connection*: Be able to interface with the FPGA board
4. *Clock*: The ADC board needs an external clock

Secondly, the FPGA requirements:

1. *Data Rate*: The FPGA must handle a 1 GSPS data rate
2. *Connectors*: Connectors must also handle high data rate

These are the main requirements for both the front-end, and the back-end.

By using the specifications outlined by the customer, and the engineering specifications, we were able to identify the critical components needed to make the front-end function properly. Once smaller components are determined for a final PCB layout design, we will order a PCB layout from a manufacturer. Then once the board is

received the front-end PCB board will be assembled by us and tested for proper functionality.

Research has led us to conclude that regular PMOD connectors on most FPGAs development boards are insufficient to handle the data rate we are aiming to achieve with this design. This realization is what led to the decision to use an FPGA Mezzanine Card (FMC) connector for our design, since it can handle the high data rates we will require.

3.0 Problem Statement

3.1 Motivation

Side-channel analysis is a method of exploiting the hardware implementation of a computer system by using information that the system leaks through things like electromagnetic emissions (EM), power consumption, and computation timing to determine sensitive information such as a cryptographic key.

Non-invasive EM side-channel analysis can be done using measurements acquired from off-the-shelf equipment like an oscilloscope. In order to detect the often miniscule EM fluctuations a very sensitive oscilloscope that can measure high frequencies is required. The oscilloscopes suitable for this can cost something on the order of $10,000 or more, making them essentially unattainable to the average consumer or academic. Designing a low cost high-speed analog capture device (HACD) would enable users to perform EM side-channel analysis without spending a fortune.

3.2 Identification of Need

Due to the exceedingly high cost of measurement equipment typically used for EM side-channel analysis, we've identified the need for a lower cost measurement device that can accomplish the same task. The High-Speed Analog Capture Device (HACD) will meet the technical requirements to detect both low frequency inductive coupling EM emissions and high frequency capacitive coupling EM emissions providing a complete analog capture system at an affordable price. HACD will allow users to perform EM side-channel analysis for a fraction of the cost, which will be especially useful to users with limited funding. In particular graduate students would be able to use the device to conduct research on side channel-attacks. HACD could be a catalyst for promoting growth in this field of research at the graduate and possibly undergraduate levels.

3.3 Market/application Review

Typical side-channel analysis tools available to individuals can cost upwards of $5000 dollars, a substantial cost to students, researchers, and hobbyists. Our proposed solution will reduce the cost substantially, making side-channel work more affordable for this demographic. It will also provide a guide through how to properly construct the HACD for anyone who wants to recreate or build off of the project. Most people who will be interested in this project will either have an interest in learning about side channel analysis, or to conduct tests using the HACD.

4.0 Approach

4.1 Problem Analysis

The OpenADC project can generate data at a rate of 105 MSPS and has a bandwidth of 40 MHz. While this design is acceptable for general purpose applications, it does not allow the OpenADC to be useful in specialized applications like side-channel analysis.

4.2 Approach

We have selected a Top-Down approach for the design HACD. We have split our design into two systems: the “frontend” which houses the ADC and supporting hardware, and the “backend” which is the FPGA Development Board. These two components will communicate via a FPGA Mezzanine Connector (FMC).

Our frontend design is influenced by both the OpenADC project and by a manufacturer produced evaluation board that is specific to our selected ADC. The ADC board will consist of a high speed ADC which will take the differential inputs through four different SMA connectors. Two will contain transformers and two will contain low-noise amplifiers with variable gain. The amplifiers must have at least a 400 MHz bandwidth to allow for EM measurements. We plan to send the differential input to the ADC where our analog sample data will be converted. Then, with the use of FMC connectors, we will interface with the FPGA to send our newly converted data to the development board.

Our backend design is a firmware based solution for ZYNQ-7000 SoCs. Specifically, we will target the ZedBoard since this is one of the most ubiquitous entry level ZYNQ device available on the market. The firmware will consist of a data forwarding pipeline which will take the input stream from the frontend via FMC and forward it to the Ethernet controller for transmission to a PC. We plan to forward data within the FPGA using AXI format packets. This allows us to maximize our usage of

Xilinx IPs which will increase the portability of the firmware to other FPGA platforms. Refer to Figure 1 for more details.

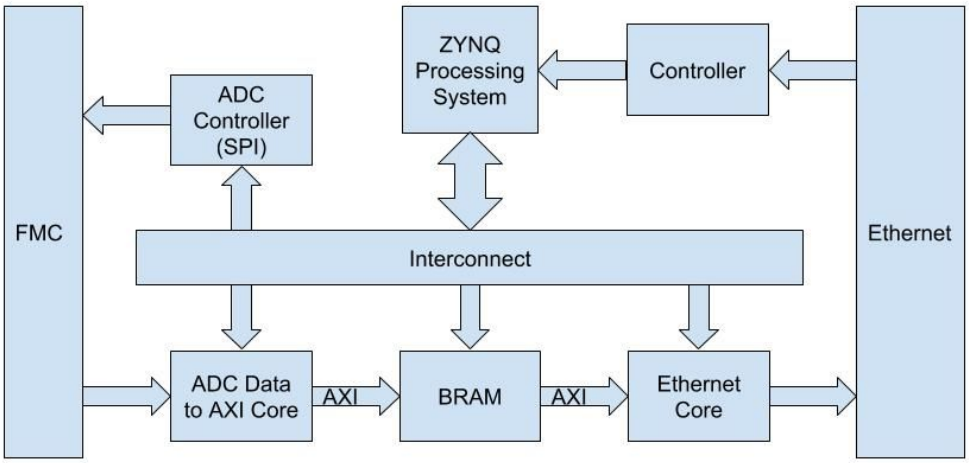


Figure 1: FPGA Firmware Diagram

4.3 Alternative Approaches

We selected our main approach from a variety of different methods we had brainstormed. Each achieved the same goal but with certain steps or components that were different.

One of these alternate approaches is soldering the FPGA to the same board as the ADC. This would work perfectly fine and achieve the same goal as in our main approach. However, the downside is the extra soldering. This could cause a pin on the FPGA to burn out, or a solder bridge could be created between two bridges. This approach, while possible, does increase the possibility of error in the project.

There was also the possibility of using a more forgiving ADC. Our current ADC samples at a rate of 1 GSPS, but we could have chosen one that could sample at a faster rate. While it would work and seems like a more efficient approach to this project, the cost for ADCs that sample at a faster rate is much greater than one that samples at 1 GSPS. This project is designed to be affordable not just for engineers and students, but also hobbyists who may not be willing to spend over $100 for an ADC.

The last alternate approach was our choice to have four differential inputs on our project. The ADC offers single, double, or quad differential inputs. The data rate is split as the number of inputs gets higher. We chose four differential inputs so that each input

had a rate of 250 MSPS going in. It is possible to use a single or dual ADC, it just may require some changes in the circuit.

4.4 Background Knowledge

The HACD requires an in depth understanding of two main topics, RF and analog circuit design and FPGAs. These concepts are split between the front end of the HACD and the back end.

The front end design for the high speed ADC board, requires knowledge in RF and analog circuit design. The board is taking in analog signals and converting it into a bitstream of sample data at an extremely fast rate. There are transformers and LNAs connected to the inputs. This allows the signal to be amplified but also reduces the amount of noise that is going into the ADC so that it can convert more accurately. While this part focuses more on the RF part of the board, an analog part is needed as well. This involves mostly resistors and capacitors for filtering and signal conditioning throughout the PCB for the circuit to work properly. As well as being able to design this circuit using RF and analog knowledge, a strong basis of soldering electrical components is needed. The components are required to be SMT soldered which is a very sensitive type of soldering, and can cause solder bridges or pins to burn out on ICs.

The FPGA development board we selected is the Digilent Zedboard which features the Xilinx Zynq-700 FPGA SoC. The firmware outline for the back end is detailed more in Figure 1. The Zynq SoC is the main element that is in between the user and the ADC board. VHDL code (firmware) will be written for various components that will be implemented on the Zynq's FPGA. These components will perform functions such as:

- Routing sample data from the FMC interface to the Zynq's internal memory
- Configuring the ADC through Serial Peripheral Interface (SPI) communication
- Sending and receiving data using an Ethernet protocol
- Accepting user commands and generating the corresponding control signals

4.5 Requirements

Mission Requirements

- The front-end device shall measure fluctuations in supply voltages or the electromagnetic radiation of a "black box".
- The front-end device will be designed in such a way that allows it to interface with any supported FPGA development board to act as the back end component.

- The back-end device will be able to process the input data from the front-end at the required speeds and send the results via Ethernet.
- The device shall be low cost (<$300).

Operational Requirements

Input/Output Requirements

- The device shall have at least two analog input channels.
- The front-end circuitry for one input channel shall utilize a CT transformer.
- The front-end circuitry for one input channel shall utilize a low-noise amplifier (LNA).
- The ADC output samples shall be no less than 8 bits.
- Shall be compatible with digital I/O voltages between 2.25 - 3.6V.
- An SMA connector for an external clock will be used.

External Interface Requirements

- The front-end device shall use an external clock.
- The front-end and back-end devices will be able to interface with each other via FMC connectors.

Functional Requirements

- The front-end device shall sample at least 1GSPS.
- The front-end device shall achieve at least 400 MHz bandwidth through the LNA.
- Target FPGA development board shall have device driver firmware for interfacing with the ADC board.
- Target FPGA development board shall have firmware for routing and storing ADC sample data in a memory device.

Technology and System-Wide Requirements

- The front-end device shall use an ADC chip.
- The ADC chip shall be soldered to a PCB.
- The front-end device should interface with and support ZYNQ FPGA for data.

5.0 System Design

This project is designed from top to bottom, so we needed to first perform a functional decomposition so that we can understand how this project properly works. Once we gained an understanding of each top level function and how they are related, we created a system architecture. The system architecture breaks down each module and conveys a top-level understanding of functionality and how the various components are connected.

5.1 Functional Decomposition

Below is a black-box representation as seen in Figure 2. This black box diagram solely focuses on the inputs, outputs, power supply, user commands, and what functions are executed inside of the HACD.

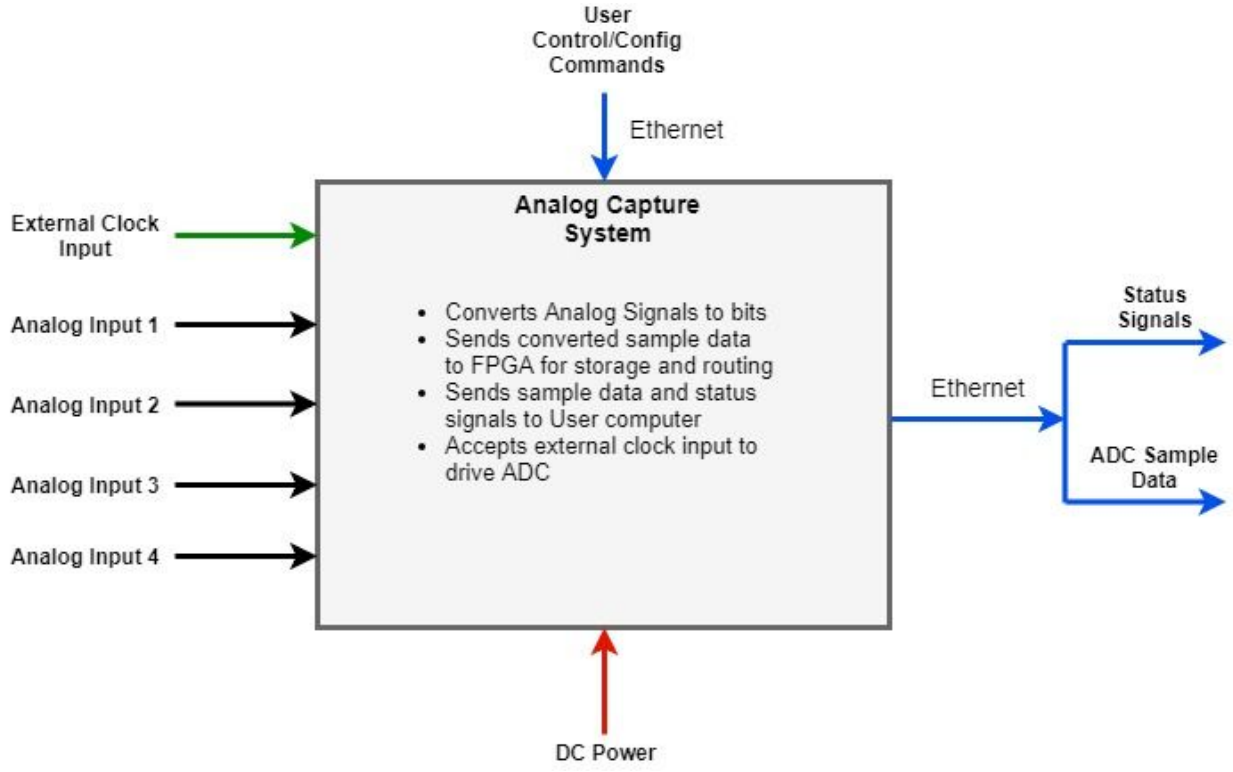


Figure 2: Black Box Diagram

Once we have analyzed the black box diagram and figured out the functions that will take place inside of the project, we need to break down how each of these functions are connected and what goes into each and what each outputs. This gives a little more insight of the flow of data throughout the project. This connection of functions can be seen in Figure 3. It shows the path of the input data and how it gets to the user's computer.

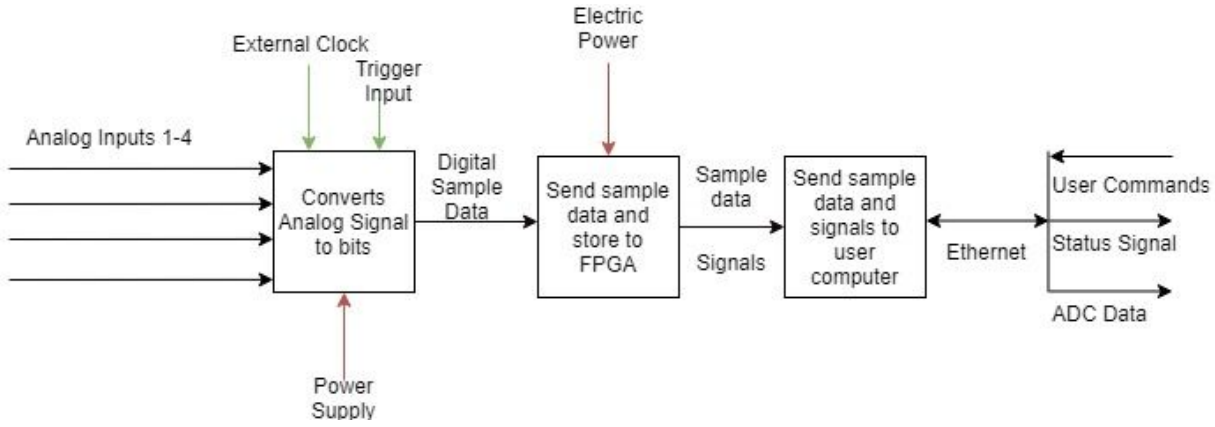


Figure 3: White box diagram

Once the flow of data over the larger functions is established, it is required to break down the larger functions down. Figures 4 through 6 show functional decompositions of each function. This is just to see the necessities of what each function needs to work, and what is needed to make the data flow through it.

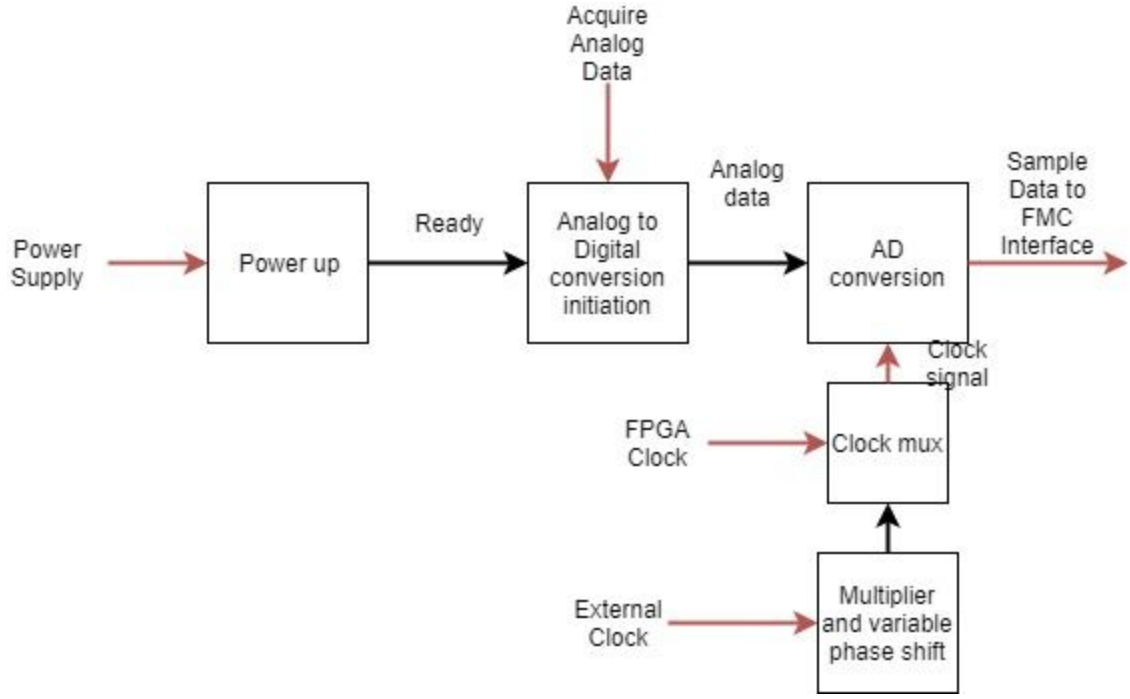


Figure 4: AD conversion decomposition

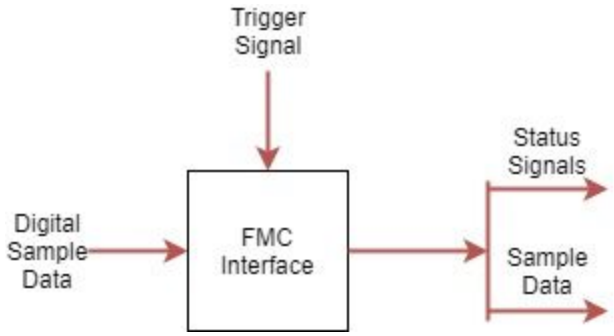


Figure 5: Send Data to FPGA decomposition

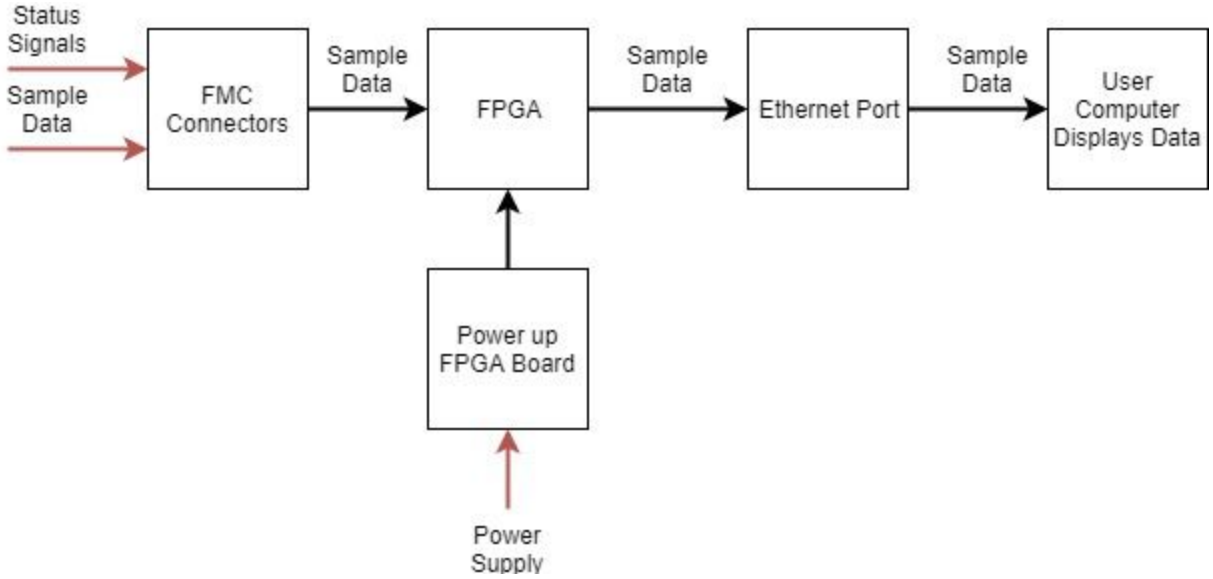


Figure 6: Data and signals to computer

5.2 System Architecture

The system architecture begins by doing a physical architecture of the project. Since we have two main parts, the front end ADC board, and a back end, the FPGA board, we did two separate physical architectures. The physical architecture for the ADC board can be seen in Figure 7. This goes over each of the modules within the high speed ADC board and what are the basic parts needed to make them work. Figure 8 shows the modules needed for the FPGA board. Table 1 contains a list of all main components that are needed to make the HACD work properly. Lastly in this section is the full system diagram seen in Figure 9 which shows how each of the parts actually connects and works together to perform its task.

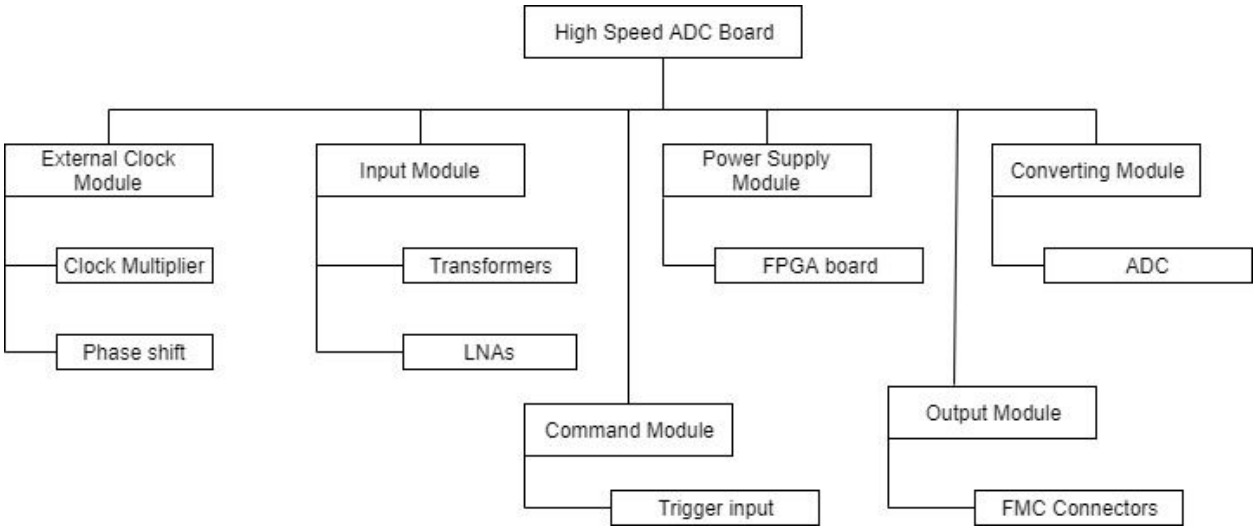


Figure 7: ADC Board Physical Architecture

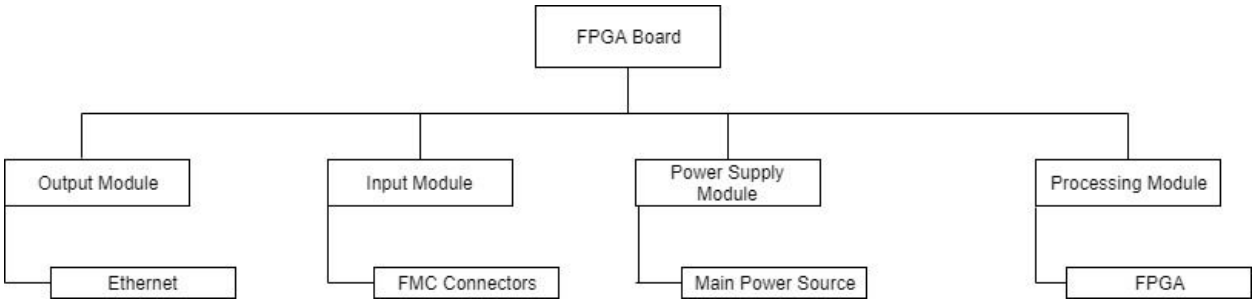


Figure 8: FPGA Board Physical Architecture

Component Name	Component Number/Type
Zedboard	Zedboard
ADC	584-HMCAD1511
Eval Kit	584-EKIT01-HMCAD1511
FMC connector	Low Pin Count
RF Transformer	TC1-1-13M-75X+
LNA	AD8376
Variable Phase Shifter	HMC936A
Clock Multiplier	HMC445

Table 1

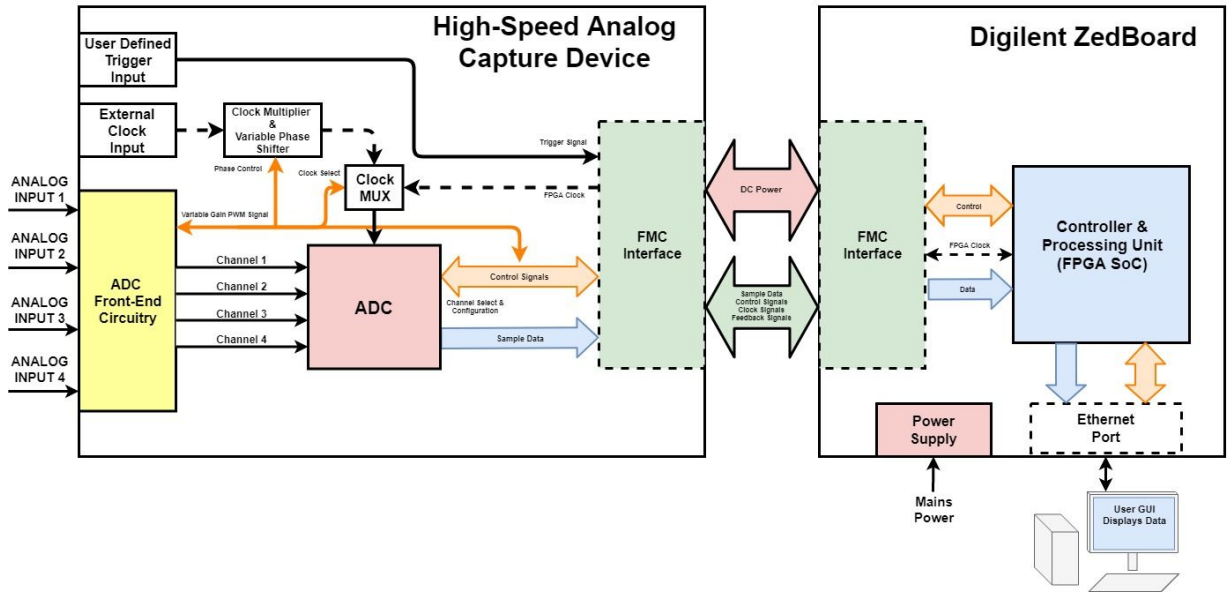


Figure 9: System Architecture

6.0 Preliminary Experimental Plan

To aid in the verification process, the software tool for our design will be able to run diagnostic and test routines to verify correctness. These routines will test specific functional aspects of the design, following the Test Driven Development (TDD) paradigm.

To verify that our solution meets the proposed specifications, we have selected the following requirements to validate our design.

1. The design must be able to handle a 1GSPS data rate
2. The design must be able to handle a 400 Mhz bandwidth through the LNA
3. The FPGA must be able to handle a 1 GBPS data rate

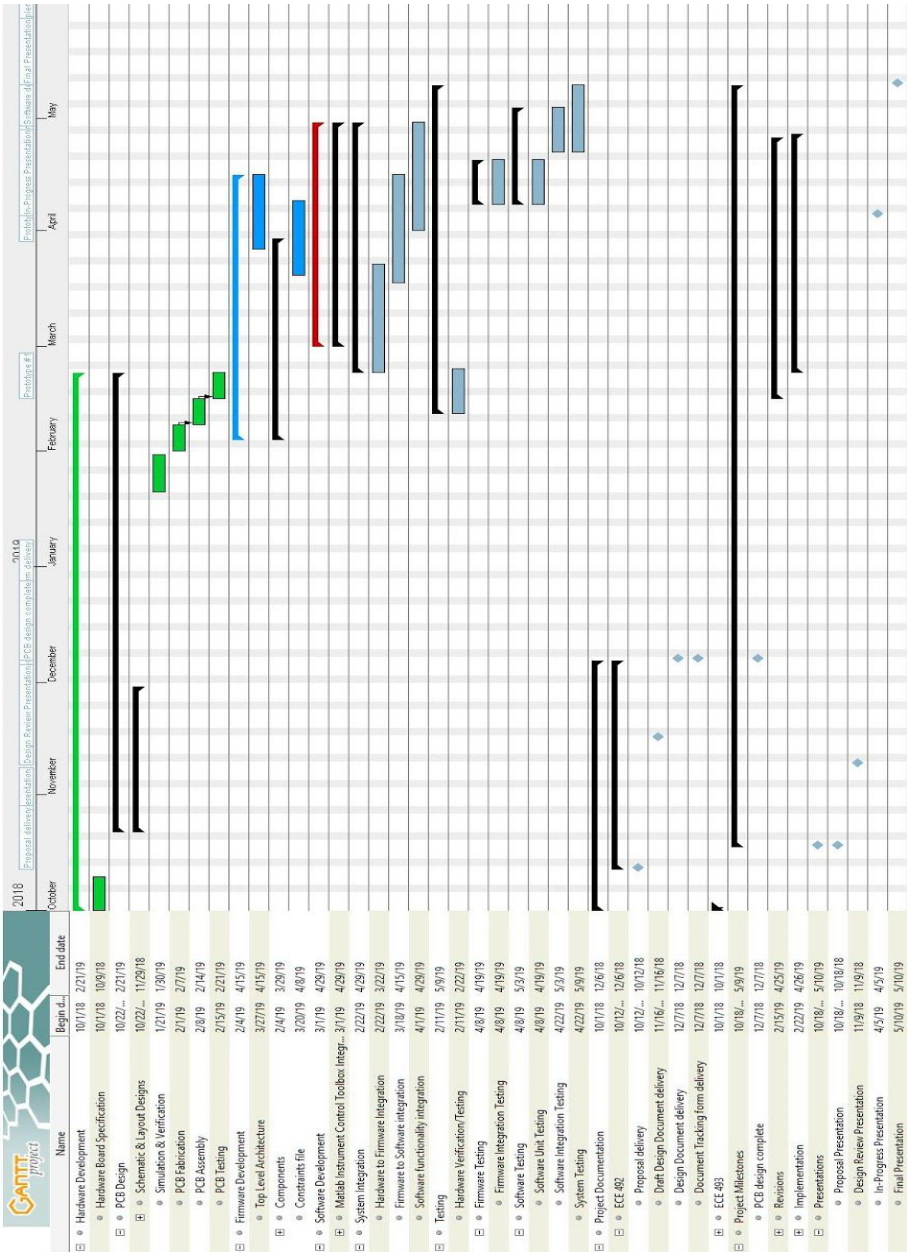
We have purchased the reference development board for our selected ADC (“EasyBoard”). Since this development board is functionally similar to our proposed device, we will use to verify the correctness of our implementation.

6.1 Proposed Experiments

The following is a tentative list of experiments. We hope to add to this list in the future once we have a better understanding of what is involved in hardware device testing

1. Sine wave test
    a. An oscilloscope probe will be attached to the device
    b. A function generator will create a sine wave across a resistance
    c. The device will measure the sine wave for one second
    d. We will verify the following from this test using MATLAB
        i. The correct amount of data will be collected from this test (1 billion samples)
        ii. The error introduced by the device by comparing the expected results from actual data
        iii. The bit error rate during ethernet transmission
2. Comparison Test
    a. An oscilloscope probe will be attached to the device and the EasyBoard
    b. A function generator will create a sine wave across a resistance
    c. The device and EasyBoard will measure the sine wave for one second
    d. Both data sets will be compared using MATLAB and will verify the following:
        i. Our device produces the same results as the EasyBoard, within a reasonable margin
        ii. The error rates of both devices are similar
        iii. The correctness of our device compared to the reference board

7.0 Preliminary Project Plan



8.0 Potential Problems

A large knowledge and skill gap will need to be filled if we are to complete this project. We lack sufficient expertise in the following topics:

- Analog Design
- PCB Design and Layout Methodology
- Real Time Data Analysis Software Programming
- Device Driver Development
- SMT soldering

8.1 Risk Analysis

The following risk sources have been identified: knowledge gap, time management, part degradation, and team motivation. Our team has a large knowledge gap that must be filled in order to be successful, and we have accordingly allocated time for research to fill this gap. Time management is related to our knowledge gap, due to our large research and experimentation time, there is some risk in falling behind schedule. We hope to mitigate this risk with proper planning and strong group cohesion. Part degradation refers to our components being damaged during the course of the project, either through human error (bad soldering) or an act of nature (static shock). Unfortunately, this risk is difficult to handle and the only solution is experience. Lastly, team motivation and morale is a risk source. During large projects, it's easy to feel "stuck" if progress stalls and this can be damaging to our group's emotional health. We hope to prevent this from happening through attainable goals, communication, planning, seeking our advisor, and maintaining focus throughout the project.

9.0 References

[1] “OPENADC Product Datasheet”, OpenADC datasheet.
[2] “*OpenADC Project*.” *Assembla*. (n.d.). [Online]. Available: https://app.assembla.com/spaces/openadc/wiki/Home [Accessed Sept. 19, 2018].
[3] Analog Devices, “High Speed Multi-Mode 8-Bit 30 MSPS to 1 GSPS A/D Converter,” HMCAD1511 datasheet.
[4] Texas Instruments, “Selecting Amplifiers, ADCs, and Clocks for High-Performance Signal Paths,” datasheet.
[5] J. Heath, “Choosing a reference voltage for an ADC,” *powerelectronictips.com*, Sept. 15, 2017. [Online]. Available: https://www.powerelectronictips.com/choosing-reference-voltage-adc-2/ [Accessed Sept. 18, 2018].
[6] T. Neu, “A checklist for designing RF-sampling receivers,” *edn.com*, June 22, 2016. [Online]. Available: https://www.edn.com/Home/PrintView?contentItemId=4442261 [Accessed Oct. 5, 2018].
[7] AVNET, “Zynq™ Evaluation and Development Hardware User’s Guide,” Zedboard datasheet, August 1, 2012.
[8] L. Crockett, R. Elliot, M. Enderwitz, and R. Stewart, *The Zynq Book*. 1st ed. Glasgow, Scotland: Strathclyde Academic Media, 2014.
[9] Hittite, “Hardware User Guide, EVAL01-HMCAD1520,” 584-EKIT01-HMCAD1511 datasheet.
[10] AVNET, “(Zynq™ Evaluation and Development) Hardware User’s Guide,” Zedboard User’s Guide, January 2014.
[11] Analog Devices, “Ultralow Distortion IF Dual VGA,” AD8376 datasheet.
[12] Analog Devices, “GaAs MMIC 6 BIT DIGITAL PHASE SHIFTERS, 1.2 - 1.4 GHz,” HMC936ALP6E datasheet.
[13] Analog Devices, “SMT GaAs HBT MMIC x16 ACTIVE FREQUENCY MULTIPLIER, 9.9 - 11.0 GHz OUTPUT,” HMC445LP4 / 445LP4E datasheet.

Appendix B: Design Document

ECE - 492 Design Document

High-Speed Analog Capture Device (HACD)

Team
Alex Wozneak
Rishub Nagpal
Ryan Meruvia

Faculty Supervisor
Dr. Jens-Peter Kaps

Class Instructor
Dr. Peter Pachowicz

Date of Submission: 12/10/2018

Table of Contents:

1.0 Problem Statement

1.1 Motivation

High-speed oscilloscopes are useful for measuring high frequency and low power analog signals. There are a multitude of applications where high-speed oscilloscopes are useful and often required. The main problem the average user in need of a high-speed oscilloscope faces is the exceedingly high cost these oscilloscopes usually mandate. This is primarily an issue for users in academic and hobbyist communities where funding is often limited.

For example, a typical oscilloscope capable of measuring signals at 500-800 MHz can cost anywhere from $4,000 to $13,000 or more. This is why our goal is to design a low-cost, open-source high-speed oscilloscope daughterboard. We've designated the name of the open-source board to be The High-Speed Analog Capture Device (HACD).

1.2 Identification of Need & Target Market

Currently there are several open-source oscilloscopes available online that can be utilized by users in communities previously mentioned. With that said there aren't many open-source oscilloscopes available that meet the performance specifications we aim to achieve with our design. HACD will achieve a certain level of performance at a fraction of the cost of oscilloscopes with comparable performance that are currently on the market. Having such a device at an affordable cost opens up a variety of high frequency applications to a user. An ideal use case for our device would be for assisting graduate students and professors in their research.

We've identified the need for a lower cost measurement device that can accomplish the same tasks as more expensive equipment. HCAD will accomplish this by providing a complete analog capture system at an affordable price. We'd like to see HACD act as a catalyst for promoting the expansion of open-source projects for high performance devices that open up research possibilities for any kind of researcher on a tight budget.

1.3 Open-Source Project

An important aspect of our project is to eventually make it completely open-source by providing all of our design documents and files online where users can use the information at their own discretion. This will likely be accomplished through creating a Git repository for the project. We will provide a user guide that will explanations,tutorials,examples and justification for some of our design choices, as well as suggestions for how our design could be

improved. We will also will provide instructions on how to manufacture the board with the files we provide. Ultimately we want to leave this project open-ended, meaning there will be multiple ways that it can be improved upon and added to in the future.

2.0 Requirements

2.1 General

- The design cost will be $300 or less.
- The front-end oscilloscope daughter-board will be designed in such a way that allows for easy interface with an FPGA development board via a low pin count FPGA Mezzanine Card (FMC LPC).
- The back-end firmware/software shall process and route the sample data to user via ethernet from a Linux web-server running on the Zynq SoC.

2.2 RF Front-end

- The front-end bandwidth will be no less than 500 MHz.
- The device will have one analog input channel.
- The ADC sample clock will be driven at 1 GHz by a PLL/VCO IC using an FPGA clock as the reference.
- The front-end circuitry will utilize an attenuator, low-noise amplifier (LNA), variable-gain amplifier (VGA), and anti-aliasing low-pass filter (LPF).
- The front-end ADC will be capable of a maximum sample rate of 1 GSPS.
- Front end programmable devices will be controlled/configured using Serial Peripheral Interface (SPI) protocol.
- The front-end will have a minimum input voltage of 50 mVpp and a maximum of 10 Vpp.

2.3 Firmware & Software

- Zynq system on a chip (SoC) will facilitate data flow from the ADC to the end user.
- A web-server running on the Zynq SoC will communicate to the user and display data.
- User will be able to issue command to the front end component through the web-server (underlying SPI protocol).

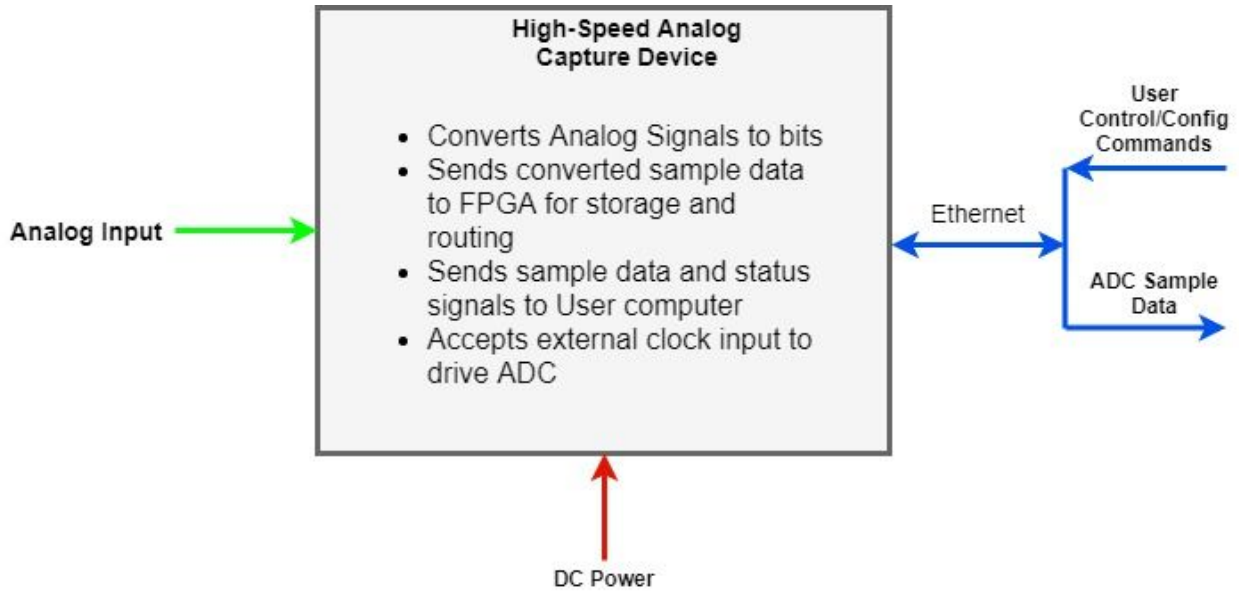
3.0 System Design

The HACD is designed in from top to bottom. We began with a black box scenario where we analyzed the inputs, outputs, user commands, and the main functions that will be executed. The functions are then decomposed and we are able to identify how each works on a smaller scale. Once the functions are understood we are able to construct a system architecture. This shows down to the component how each component works together and the flow of the data from start to finish.

3.1 Key System Specifications

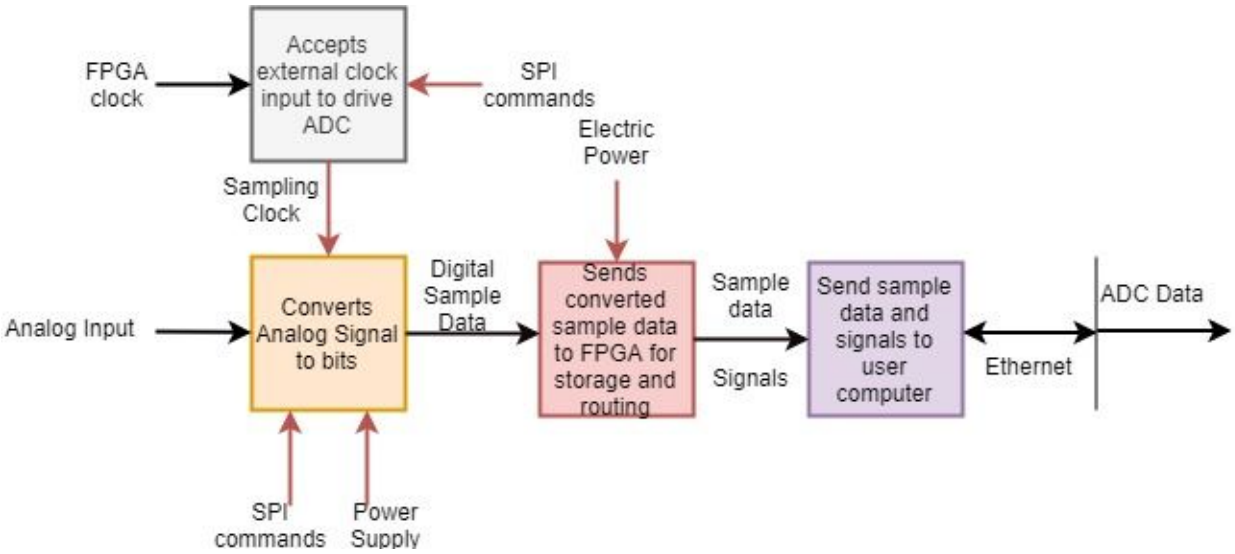
PARAMETER	SPECIFICATION
Input Channels	1
Analog Bandwidth with 50-□ inputs (fc = -3dB)	500 MHz
Maximum Sampling Rate	1 GSPS
Maximum Input voltage	± 5V
Maximum system voltage gain	19.97 dB
Calculated Rise Time (10% to 90%)	700 ps
Effective number of bits (ENOB)	7.9

3.2 Functional Decomposition



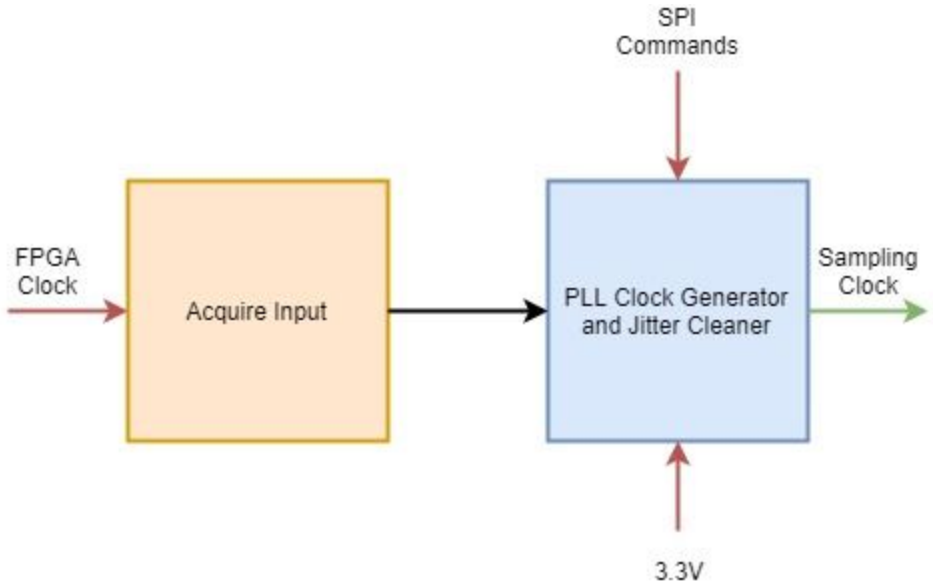
Black Box Design of HACD

Once the black box design is constructed we decompose the functions that are defined. This decomposition shows how each of the functions connect to each other as well as the specific inputs and outputs to each one. This gives a closer look to the system at a smaller scale and shows a much better idea of how the data will flow through the system. This decomposition can be see in the figure below.



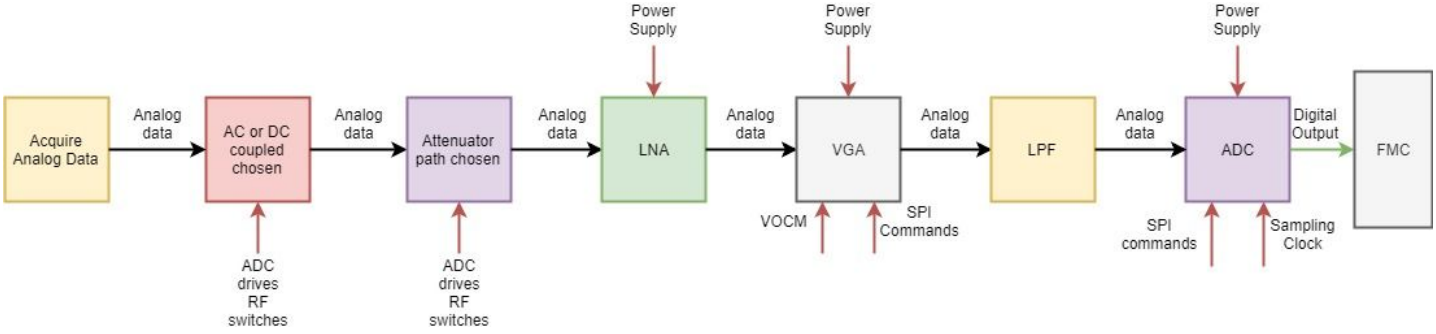
Functional Decomposition Level 1

The first functional decomposition shown above delves into how each of the functions provided in the black box design are connected and their specific inputs. The main inputs are power and SPI commands. Next we look deeper into each of the separate functions. First is the clock generator that feeds the sampling clock to the ADC. The FPGA clock will be fed into a PLL which will clean the jitter and output a 1 GHz clock for the ADC to sample with.



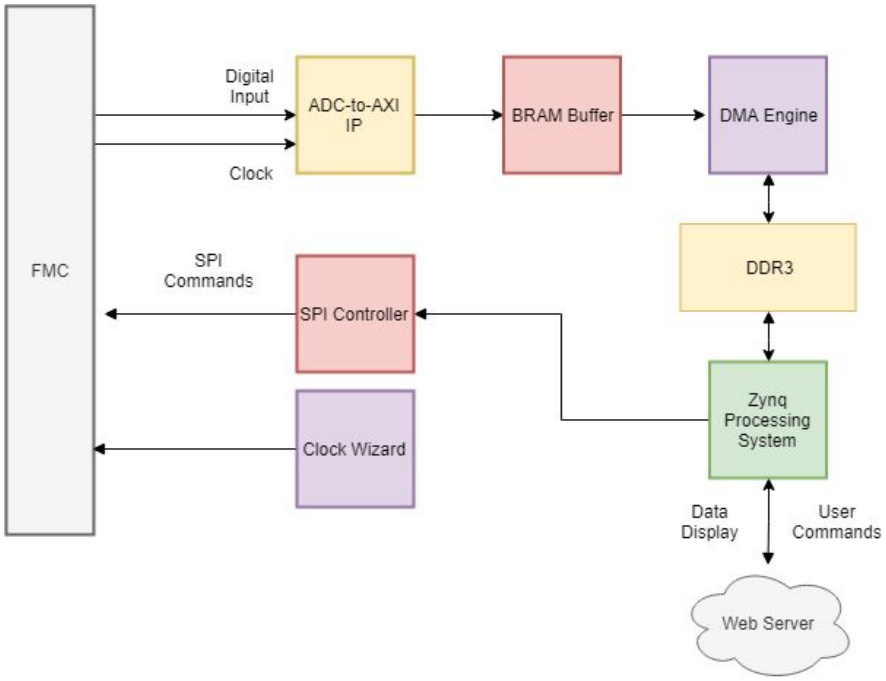
Functional Decomposition Level 2

The next figure is a look into the analog to digital conversion or the front-end of the system. The analog data is first acquired via a function generator which is then fed through different paths which can be chosen by RF switches driven by the FPGA. This signal is then sent through two amplifiers which amplify the signal to as close as the it can get to the ADC FSR. Once this is done any high frequency noise gets filtered out and the signal is fed to the ADC where it will be converted to digital data and sent to the FMC.



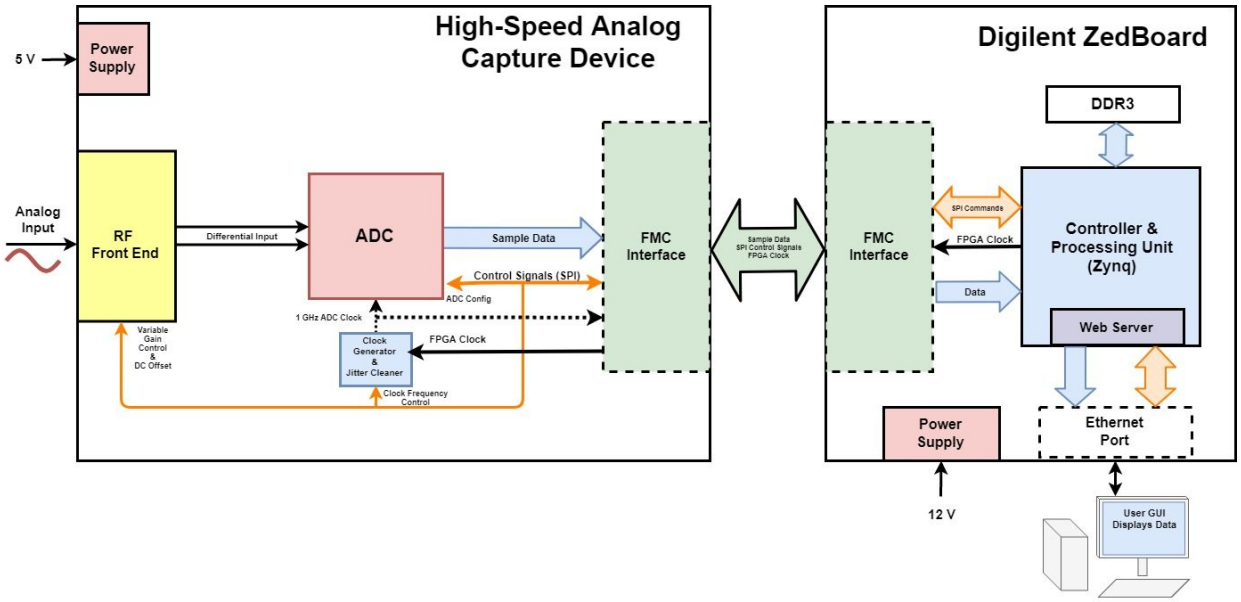
Analog to Digital Conversion Decomposition

The below figure details the firmware functional flow. The digitized data and timing signals from the ADC come in from the ADC to the ADC-to-AXI IP. This IP will be created by our team to convert the RAW input data into an AXI compliant packet format. The outgoing packets from this IP will then be buffered and sent to main memory, in our case DDR3. The Zynq Processing System contains ARM processors which will run a specialized distribution of Linux that will host a web server. The end user will be able to view the collected data through the web server, issue commands back to the front end board, and tweak various parameters.



Firmware Functional Diagram

3.3 System Architecture



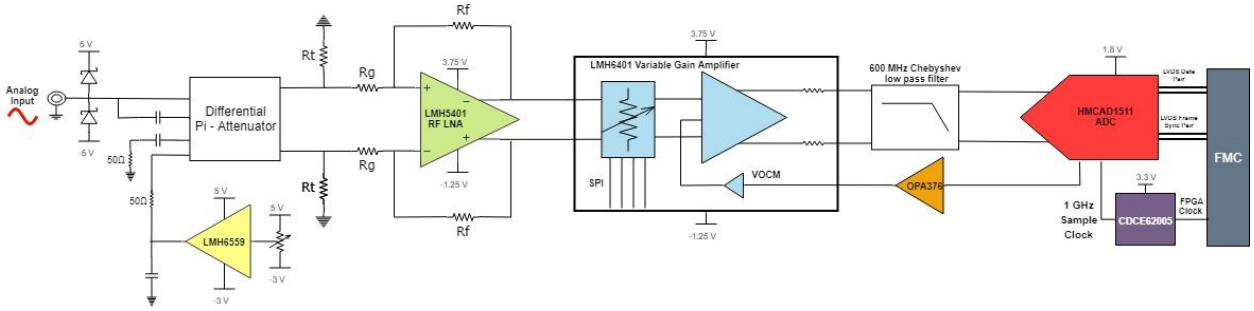
System Architecture Diagram

4.0 Background Knowledge

4.1 Overview

This section will cover the background knowledge that was employed to justify our choices and methods throughout the design process. This includes summaries of concepts that were taken into consideration for specific design elements, as well as supporting calculations that have been made to satisfy design requirements. This section is broken into two main categories, first being the RF front-end hardware design, and secondly the back-end firmware and software design.

4.2 RF Front-End Design:



Block Diagram of Front-End

System Dynamic Range & Full-Scale Input Range (FSR)

The System Dynamic Range for an oscilloscope is a figure that represents the maximum ratio of two signal levels simultaneously present at the input. Meaning it's a measure of the smallest and largest signal levels that can be measured at the same time without errors. The Full-Scale Range (FSR) of the input describes the maximum and minimum voltage swing the system is capable of accurately measuring. Below are calculations for these parameters.

Calculation of System Dynamic Range and Full-scale Input Range (FSR):

Dynamic Range of System:

$$Dynamic\ Range_{System} = Pi_{Attenuator_{Max}} + Dynamic\ Range_{LHM6401}$$

$$\boldsymbol{Dynamic\ Range_{System}} = 20\ dB + 32\ dB = 52\ dB$$

Max Voltage Gain at ADC Input

$$ADC\ FS_{in} = 500\ mV_{pp}$$

$$\boldsymbol{Max\ Voltage\ Gain\ at\ ADC\ Input} = LMH6401_{Max} + 20\log(0.500)$$

$$= 26\ dB + (-6.02)dB = 19.97\ dB$$

Minimum Oscilloscope FS Input:

$$19.97\ dB = 20\log\left(\frac{0.500}{Min\ Oscope\ FS_{in}}\right)$$

$$\boldsymbol{Min\ Oscope\ FS_{in}} = \frac{0.500}{10^{\frac{19.97}{20}}} = 0.0501\ V_{pp} \approx 50\ mV_{pp}$$

Maximum Oscilloscope FS Input:

$$Max\ System\ Attenuation = Max\ Gain - Dynamic\ Range$$

$$= 19.97 - 52 = -32.03\ dB$$

$$-32.03\ dB = 20\log\left(\frac{0.500}{Max\ Oscope\ FS_{in}}\right)$$

$$\boldsymbol{Max\ Oscope\ FS_{in}} \approx 20\ V_{pp}$$

***Note: We are limiting the max input to 10 Vpp to avoid saturating the LMH5401 (LNA) which would lead to clipping/distortion on the output signal. This is because the LMH5401 max power input is 5 V. This why there are two Schottky diodes in a clamping configuration on the input limiting the input voltage level to 10 Vpp.**

$$\boldsymbol{Effective\ Max\ Oscope\ FS_{in}} = 10\ V_{pp}$$

Rise Time(10%-90%)

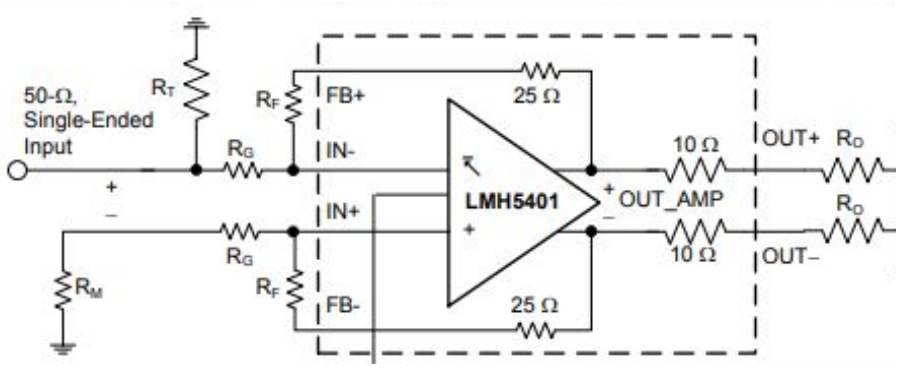
$$T_r = \frac{k}{BW}$$

The rise time for an oscilloscope is found by the equation above. The value k is normally equal to .35 for oscilloscopes with a bandwidth less than 1 GHz. Since our oscilloscope only has a BW of 500 MHz we can solve for the rise time.

$$\frac{.35}{500 \cdot 10^6} = 7 \times 10^{-10}$$

Single-Ended to Differential Conversion

In this configuration our low-noise amplifier (LNA) the LMH5401 is AC-coupled, single-ended source to differential. With a differential gain of 4-V/V, this circuit was for prototyping and calculations only.



LMH5401 Example Circuit

Solving the quadratic for the termination resistor RT, which removes DC currents from the feedback path between VOCM to ground.

$$R_T^2 - R_T \frac{2R_S\left(2R_F + \frac{R_S}{2}A_V^2\right)}{2R_F(2+A_V) - R_S A_V(4+A_V)} - \frac{2R_F R_S^2 A_V}{2R_F(2+A_V) - R_S A_V(4+A_V)} = 0$$

Here the quadratic can be solved after selecting:

- The target voltage gain Av.
- The desired input impedance at the junction of RT and RG1 to match the source impedance Rs.
- A value for the feedback resistor RF

To verify that the maximum gain is greater than the target gain after selection of RF and source Impedance Rs, use the following equation.

$$Av_{max} = \left(\frac{R_F}{R_S} - 2\right) \cdot \left[1 + \sqrt{1 + \frac{4\frac{R_F}{R_S}}{\left(\frac{R_F}{R_S} - 2\right)^2}}\right]$$

Gain resistors:

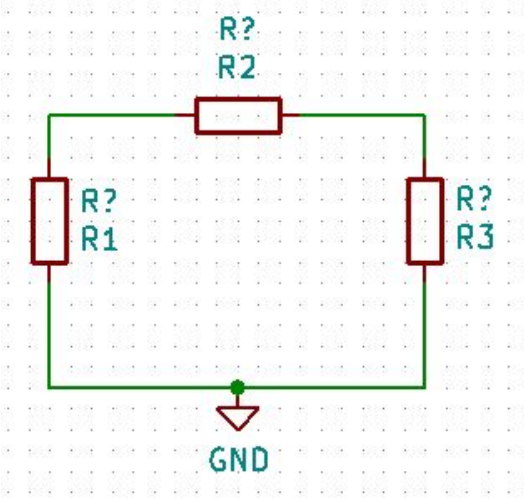
$$R_{G1} = \frac{2\frac{R_F}{A_V} - R_S}{1 + \frac{R_S}{R_T}} \qquad R_{G2} = \frac{2\frac{R_F}{A_V}}{1 + \frac{R_S}{R_T}}$$

Input impedance calculation:

$$ZA = R_{G1}\frac{\left(1+\frac{R_{G1}}{R_{G2}}\right)\left(1+\frac{R_F}{R_{G1}}\right)}{2+\frac{R_F}{R_{G2}}}$$

Attenuator

An attenuator is a voltage divider circuit that is used to step down higher voltages to lower voltages. This is beneficial when you have higher inputs going into a system but need to step the voltage down to have the system work most efficiently i.e. for an oscilloscope to achieve its FSR at the input of the ADC.



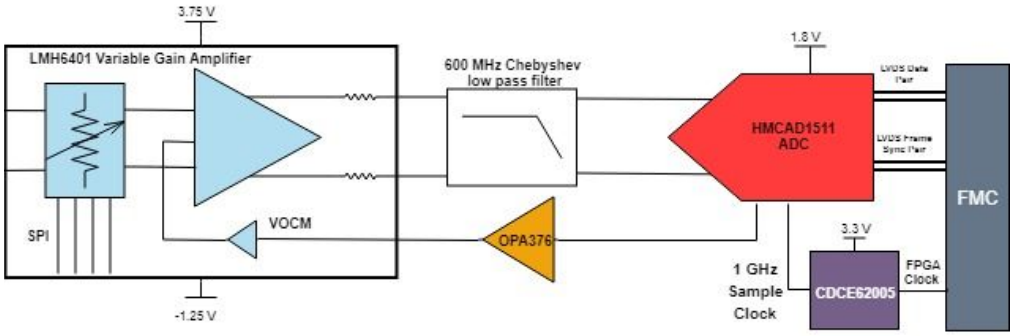
Equations:
R1 = R3 = Z(K+1/K-1)
R2 = Z((K^2 - 1)/2K)
K = 10^(dB/20)
V = 10^(-dB/20)

Above the input impedance is Z, K is the gain in volts, and V is the expected output value after attenuation.

User Controlled Gain

The Digitally-Controlled Variable Gain Amplifier (DVGA) the LMH6401 is configured to be the driver for the ADC in a gain control loop. Gain and attenuation control is controlled via SPI commands allowing for 1 dB steps between a range of -6 dB and 26 dB. The device provides an output common-mode control (VOCM) is provided in the device which allows the output common-mode voltage of the LMH6401 to match the optimal input common-mode voltage of the ADC. The UGC allows for find voltage level adjustments before entering the ADC so that the ADC's full-scale input is satisfied when

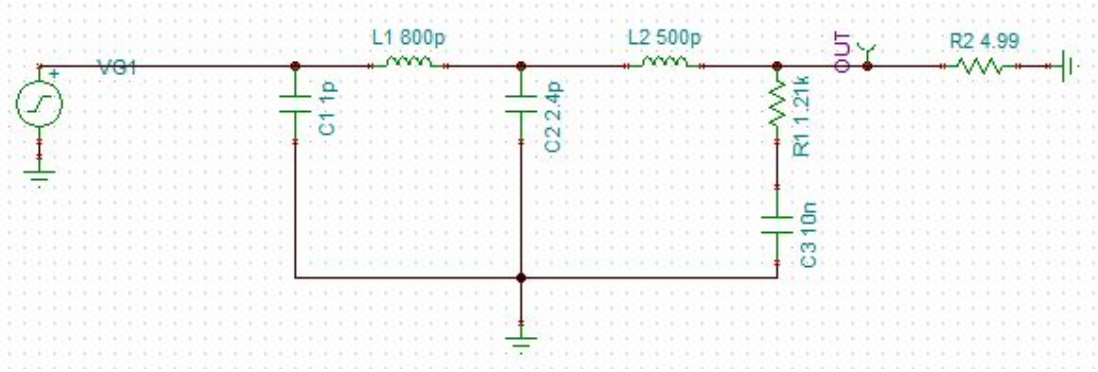
a signal is either below or above the ADC full-scale input. Below is a block diagram of the UGC loop.



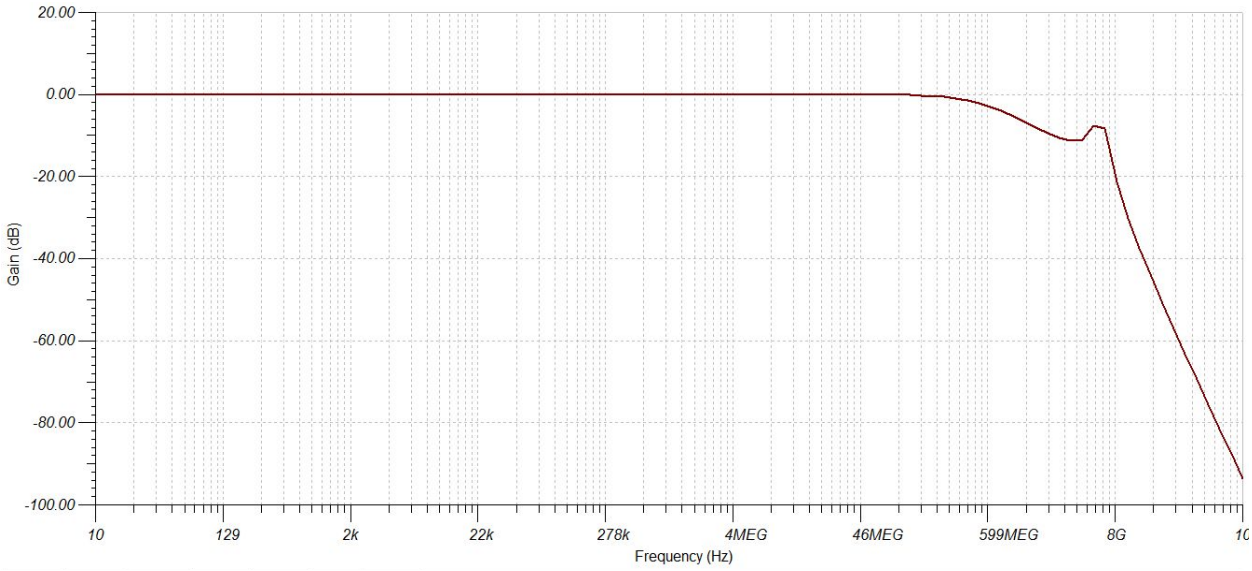
Gain Control Loop

Anti-Aliasing Filter

Anti-Aliasing Filters are used to cut out frequencies higher than the maximum input frequency. Due to the max input frequency being dependent on the sampling rate, it is not desirable to input frequencies higher than half of the sampling rate. This would cause aliasing and would not reconstruct the signal properly after filtering. We use a 5th order Chebyshev topology for our filter which is shown below. The filters frequency response graph is also shown which shows that our cut off frequency is right above 600 MHz which is the desired value.



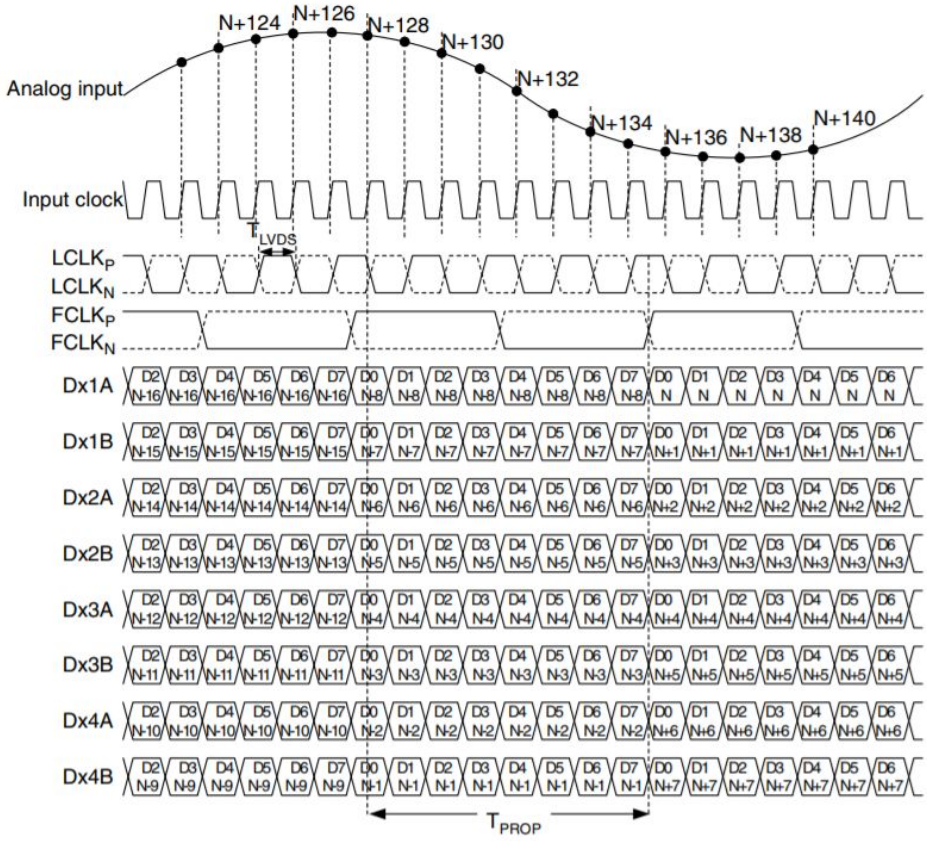
Filter Circuit



Frequency Response

Analog to Digital Converter (ADC)

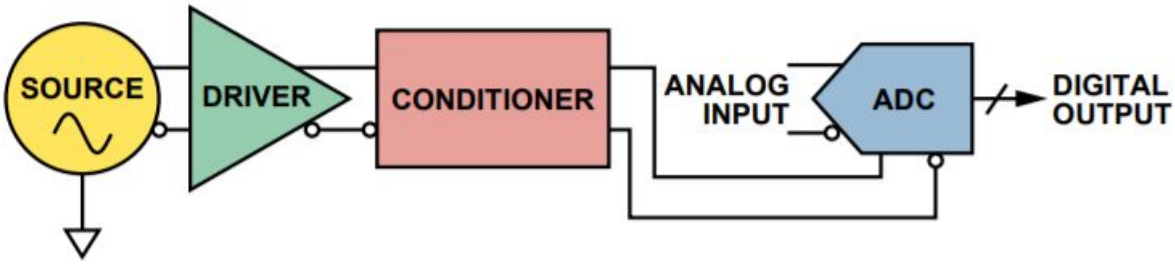
We are using an 8-bit 1 Gsps ADC the HMCAD1511 in single channel mode to maximize bandwidth. The output is serial low voltage differential signals (LVDS).



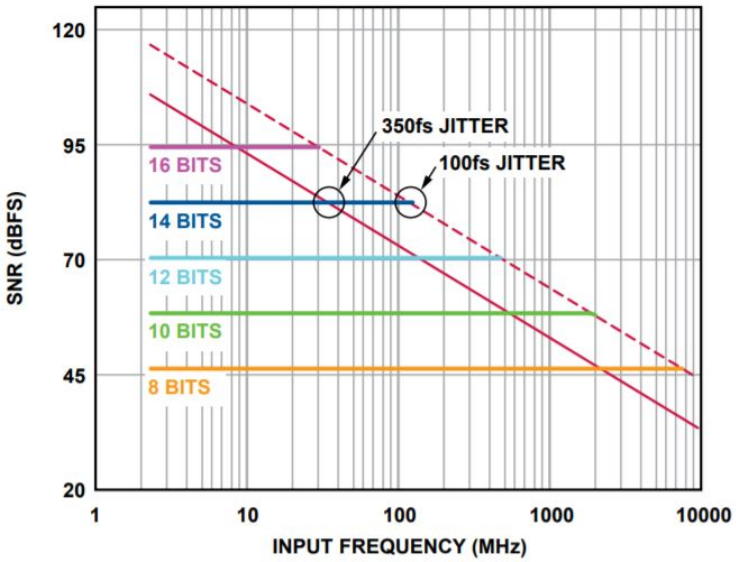
Single channel - LVDS timing 8-bit output (From HMCAD1511 datasheet)

ADC Sample Clock & Fractional/Integer-N Phase Lock Loop (PLL)

The sample clock that is being used to drive the ADC must be equivalent to the sampling rate. Since we are sampling at 1 GSPS we need to drive the ADC with a 1 GHz clock. To achieve this we will be connecting the FPGA clock to a PLL. The PLL will then multiply the input signal to 1 GHz, and that output will drive the ADC. We will specifically be using a PLL that cleans potential jitter from the FPGA clock. This will make sure that our ADC gets a clean clock to use for sampling rather than one that contains a lot of jitter and inconsistencies. Based off of the graph below, with our 8-bit ADC our maximum input of 500 MHz falls to the left of the line which means it is an acceptable input. The Nyquist theorem states that the sampling rate should be at least 2 times or greater than the maximum frequency of interest. Since our sample rate is 1 GSPS we limit our maximum frequency to 500 MHz = Fs/2.



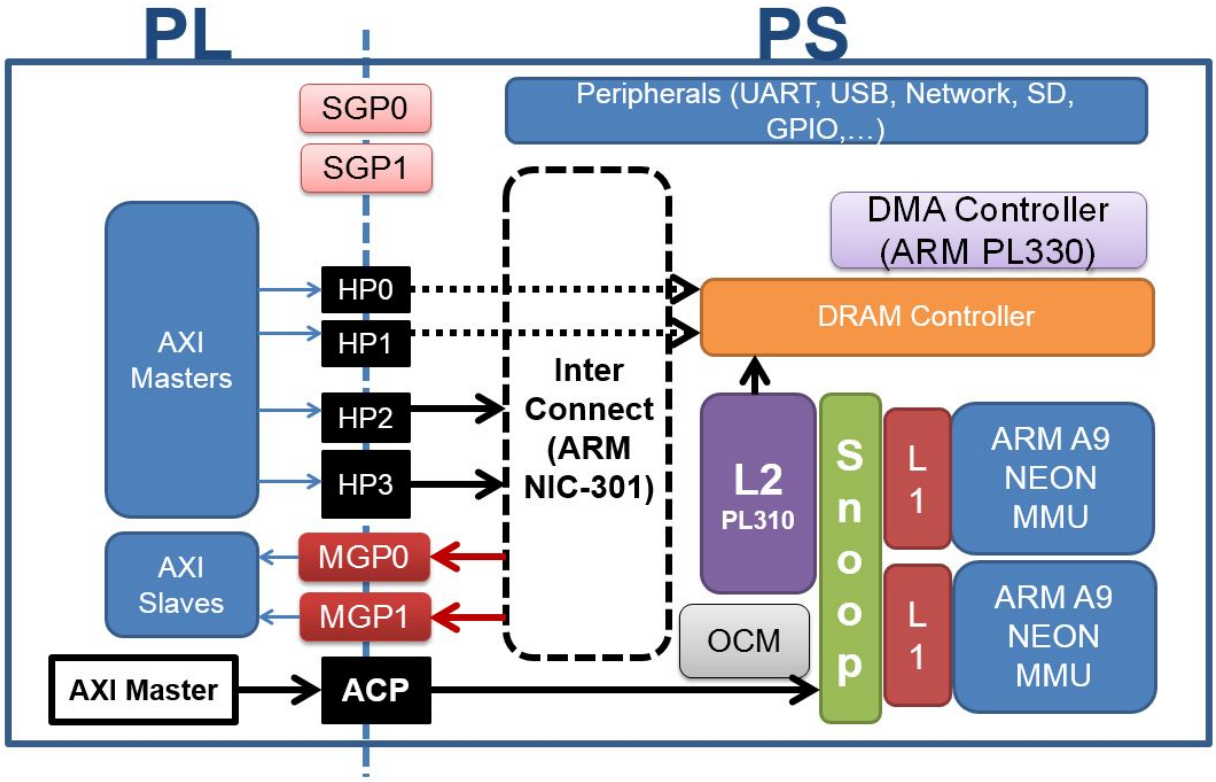
Abstracted Block diagram of Front-end



Graph indicates SNR performance of IDEAL ADCs

4.3 Firmware and Software Design:

Zynq SoC Architecture



Zynq Architecture

The Zynq SoC is composed of two components: the Processing System (PS) layer and Programmable Logic (PL) layer. The PS layer contains the ARM cores and supporting hardware, I/O Peripherals, a DRAM controller, and a DMA controller. The PL layer contains the FPGA programmable fabric which any hardware core can be implemented. The PL and PS layers communicate via ports. Our Zynq SoC exposes four high performance (HP) ports which allow for high data throughput between the PL and PS. We plan to utilize HP ports for transfering the incoming data stream from the ADC into main memory.

ADC to FPGA Datapath

Our ADC frontend will act as a daughter board for any FPGA development board equipped with an FMC header. Although developing an interface with PMOD functionality would allow greater compatibility, we found that the PMOD specification is unable to handle the fast data rates provided by our ADC.

Our datapath will consist of components which will buffer the incoming data from the ADC and store the contents into memory. The microprocessor included on our FPGA will run a linux based operating system which hosts a web server that will present

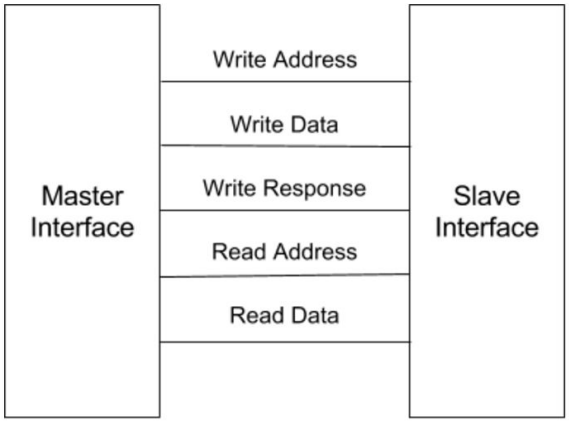
the received data to the user. This datapath design reduces the complexity of our project since we do not have to develop an interface to another PC for the user to receive our data.

Xilinx IP Cores

Xilinx provides specialized components (IP cores) for use in the design of datapaths. These IPs are specifically designed for Xilinx devices and Xilinx provides a library of these IPs for the most common functional units encountered in FPGA firmware design. We plan to utilize these IPs in our datapath to reduce the effort needed to implement our project, simplify our development cycle and increase turnaround time.

AXI Protocol

The AXI (Advanced eXtensible Interface) Protocol is the standard protocol Xilinx has adopted for data transfer within the programmable logic in FPGAs. We will adhere to this protocol since almost all Xilinx IPs adhere to this protocol and there is built in support for this protocol within Xilinx development tools.



AXI Protocol

ADC to AXI IP

In order to utilize Xilinx IPs discussed above, we must format the incoming data from the ADC to the AXI Protocol format. Therefore, we plan to create an IP core which will convert the incoming raw data to the AXI format. This IP will contain status and configuration registers which the Zynq processor can access and set via the AXI-lite protocol. This IP will contain a 64-bit internal memory register which will buffer eight 8-bit samples and send them out as a single 64-bit AXI formatted packet.

Memory Path

We will store the data received into the main memory via a DMA Controller and and then to the flash memory (SD card) of our FPGA Development board. This will

allow us to post process the data after it has been collected and present the data and history to the user via the web server. The maximum frequency of the DMA Controller will be a limiting factor of the maximum throughput available to our design. To overcome this limitation, a FIFO buffer will be implemented in the Programmable Logic (PL) layer of our SoC.The clock speed of the PL layer will be dependant on the size of this FIFO buffer. Our initial calculations for the buffer size and clock speed are detailed in the next section, but note, these calculations are not final and we plan to optimize the buffer size and clock speed once we begin the experimentation phase.

The following table details relevant data pertaining to our buffer and clock speed calculations

Parameter	**Value**
ADC Data Output Rate	8 Gbps
AXI-Stream Maximum Word Size	64-bit
AXI-to-ADC Data Output Rate	125M Words per Second
AXI DMA Maximum Frequency	150 Mhz
AXI DMA Maximum Throughput	64-bit * 150 Mhz = 9.6 Gbit/s

Since the ADC Data Output Rate < DMA Maximum Throughput, our SoC will be adequate for our application. Since sending each sample in real time to main memory would require the DMA to be clocked at 1 Ghz, Block RAM can be utilized as a FIFO buffer for the incoming data and be read from in large bursts. The size of the FIFO buffer will be set at 1024-bits, as this is the maximum burst size the DMA IP supports.

Linux OS & Web-Server

We plan to install a Linux operating system onto our device which will host a web server which the user can access the collected data. The OS will run on the ARM processors located in the PS layer of our SoC. The OS is optimized for our FPGA and will allow us to control the PL layer with a provided Python API.

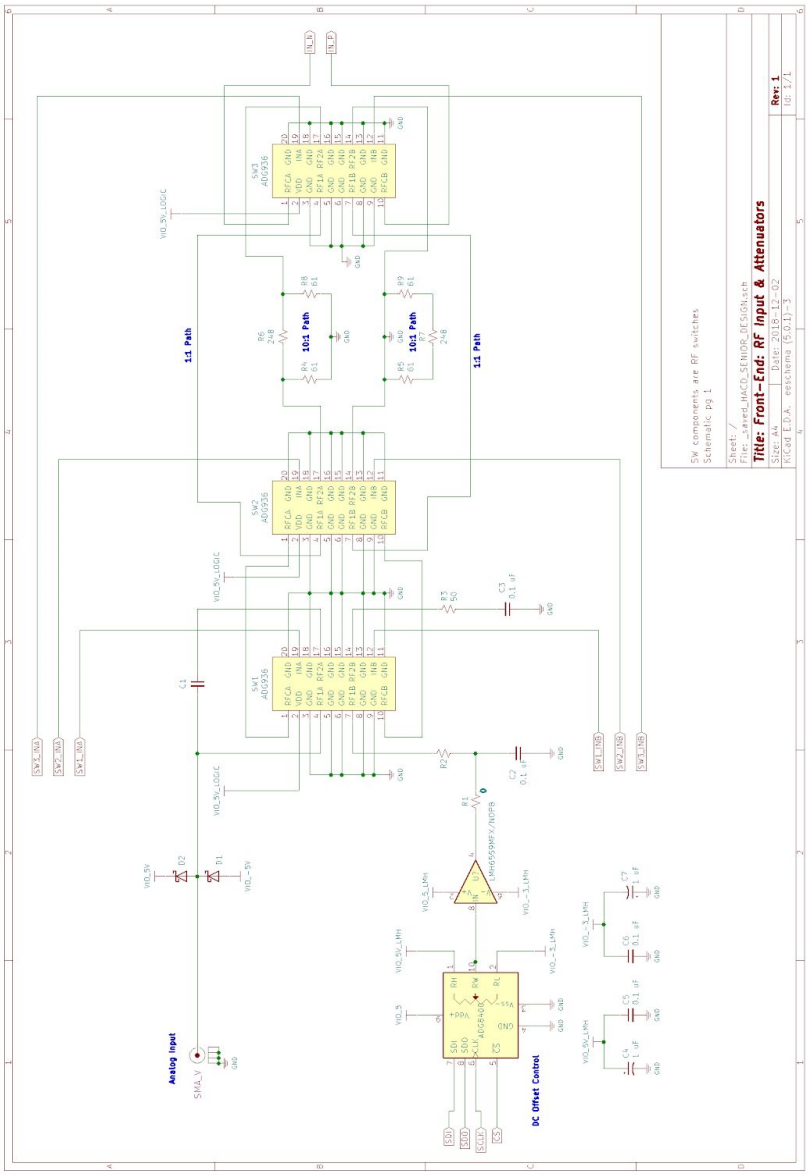
5.0 Detailed Design

5.1 Overview

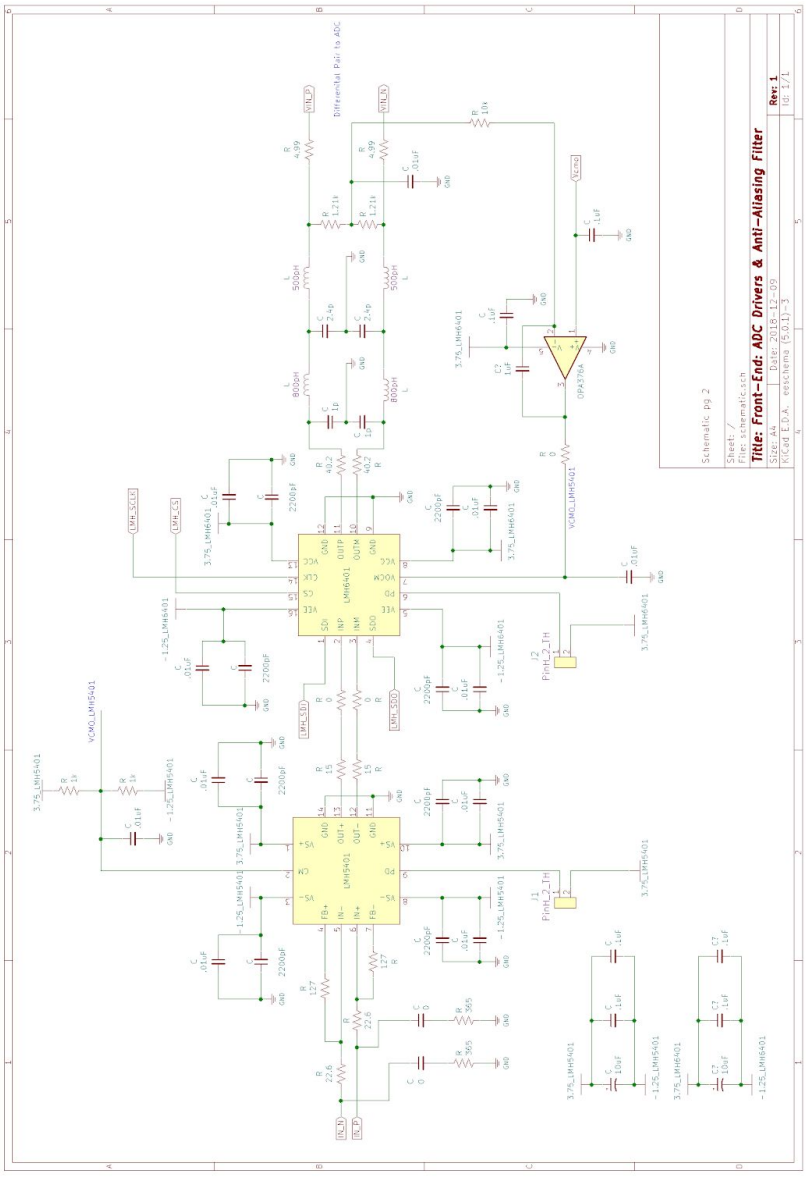
In the following sections we will be going into a more detailed design of our full system. This will include the full schematic for the front-end as well as component and schematic descriptions. It will then discuss the firmware and provide functional diagrams for the firmware.

5.2 Circuit Schematics

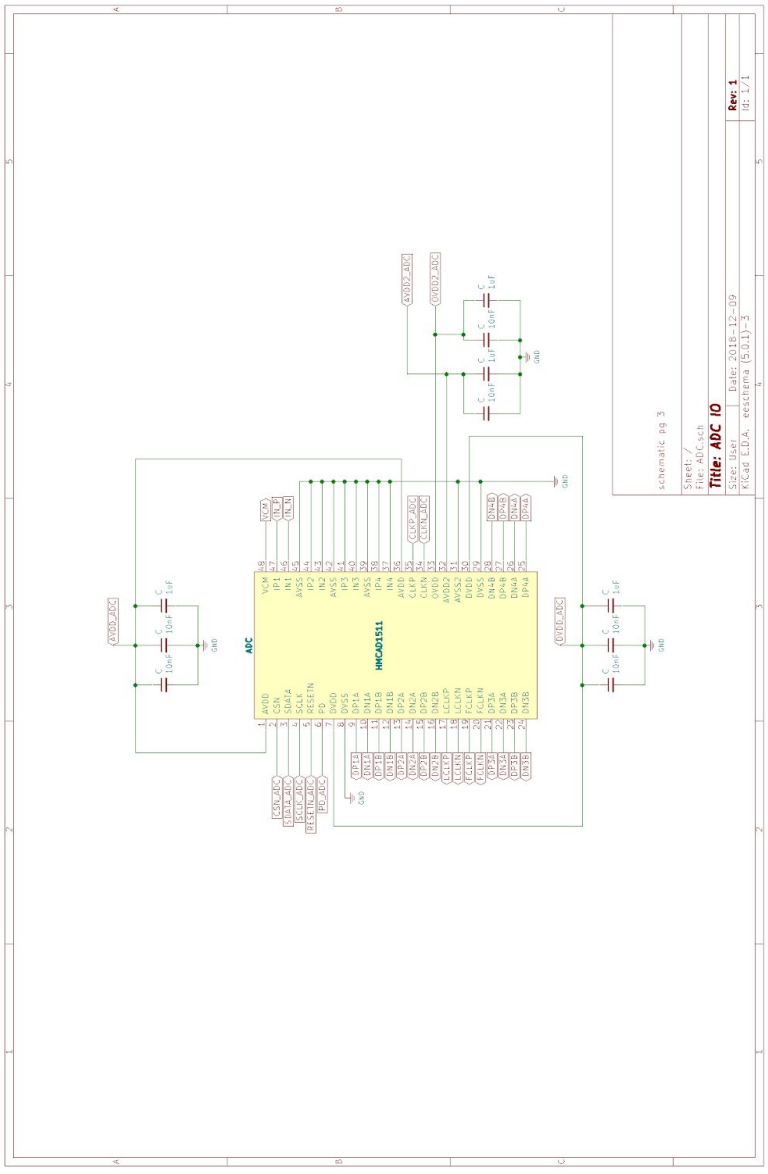
5.2.1 Front-End: RF Input & Attenuator



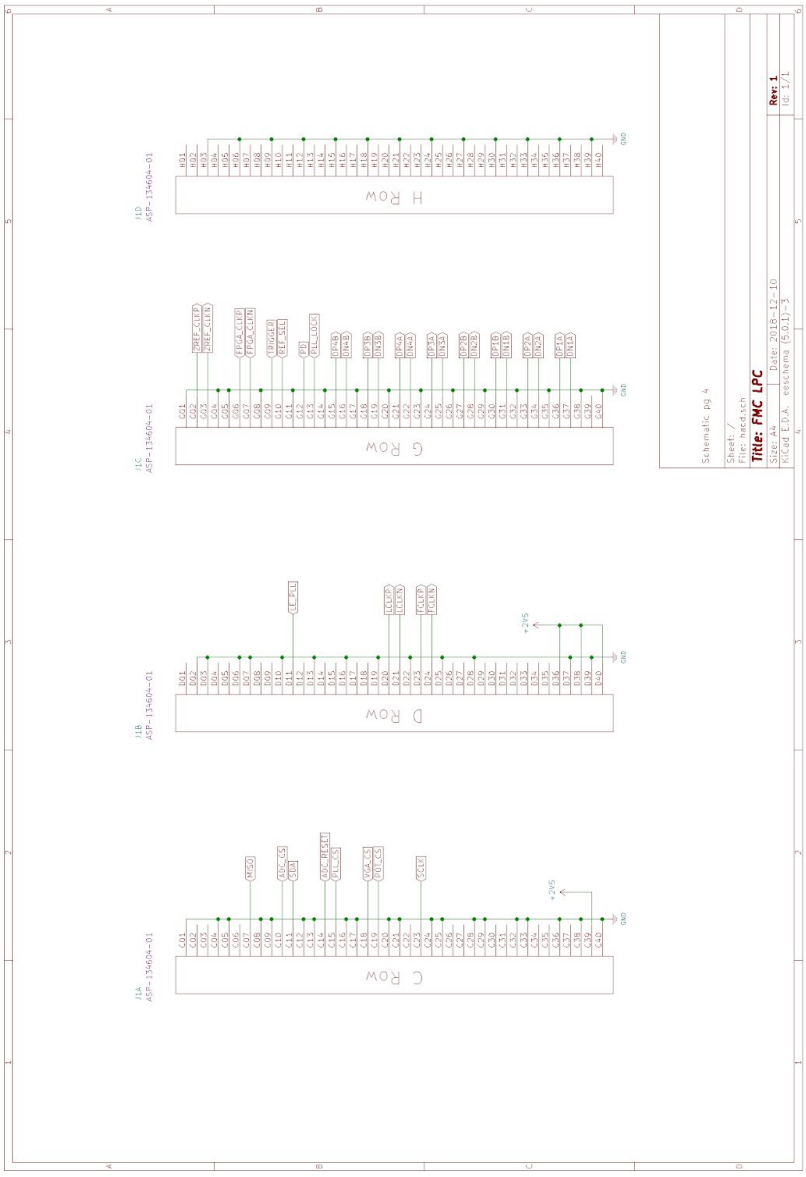
5.2.3 Front-End: ADC Drivers & Anti-Aliasing Filter



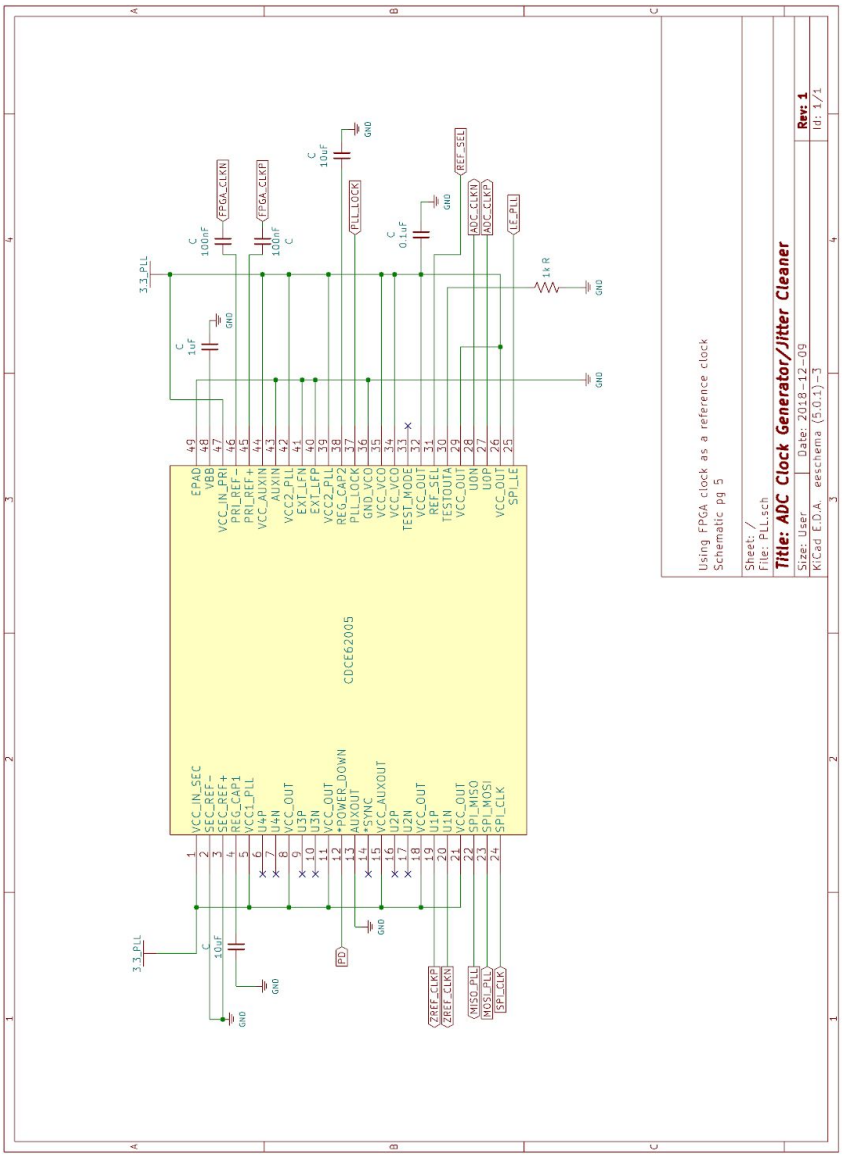
5.2.4 ADC IO


Title: ADC IO
Rev: 1
Date: 2018-12-09
Size: User
KiCad E.D.A. eeschema (5.1)-3
Sheet: /
File: ADC.sch
schematic pg 3
HMCAD1511
ADC
AVDD_ADC
DVDD_ADC
GND

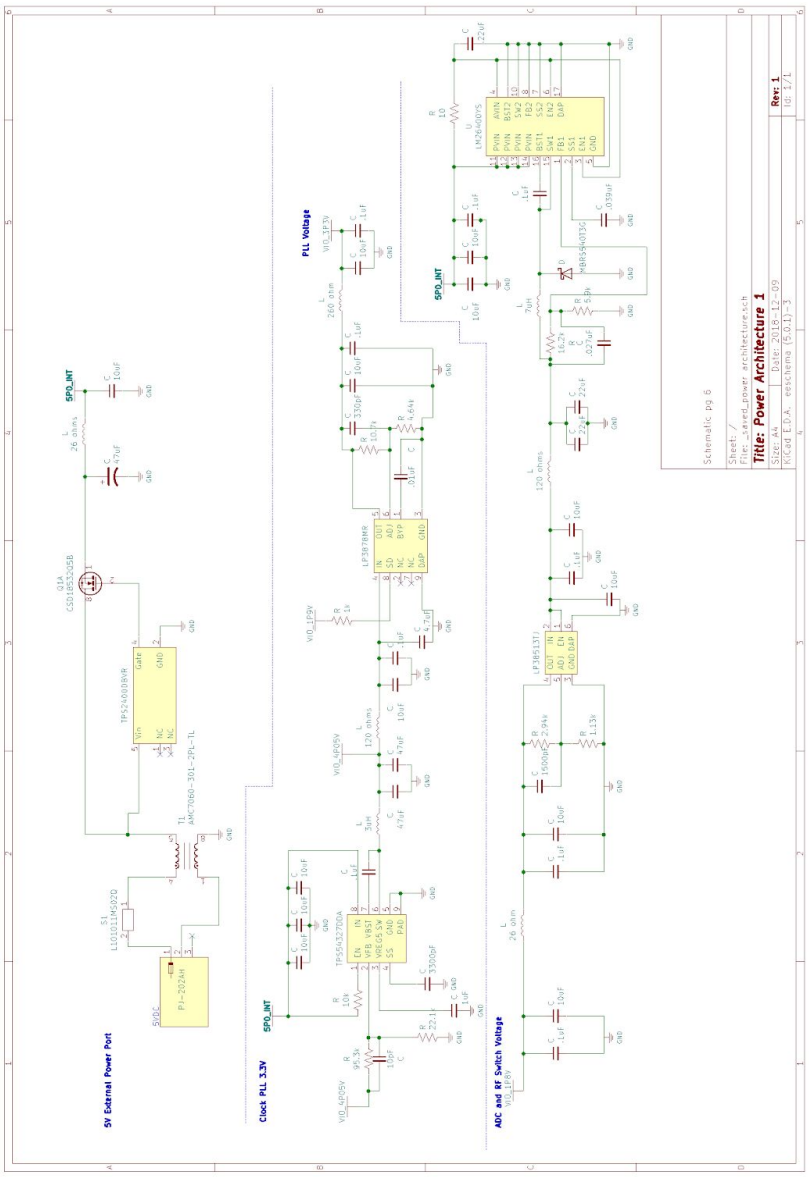
5.2.5 FMC LPC


H ROW
G ROW
D ROW
C ROW
Schematic pg 4
Sheet: /
File: hacd.sch
Title: FMC LPC
Size: A4
Date: 2018-12-10
KiCad E.D.A. eeschema (5.0.1)-3
Rev: 1
Id: 1/1

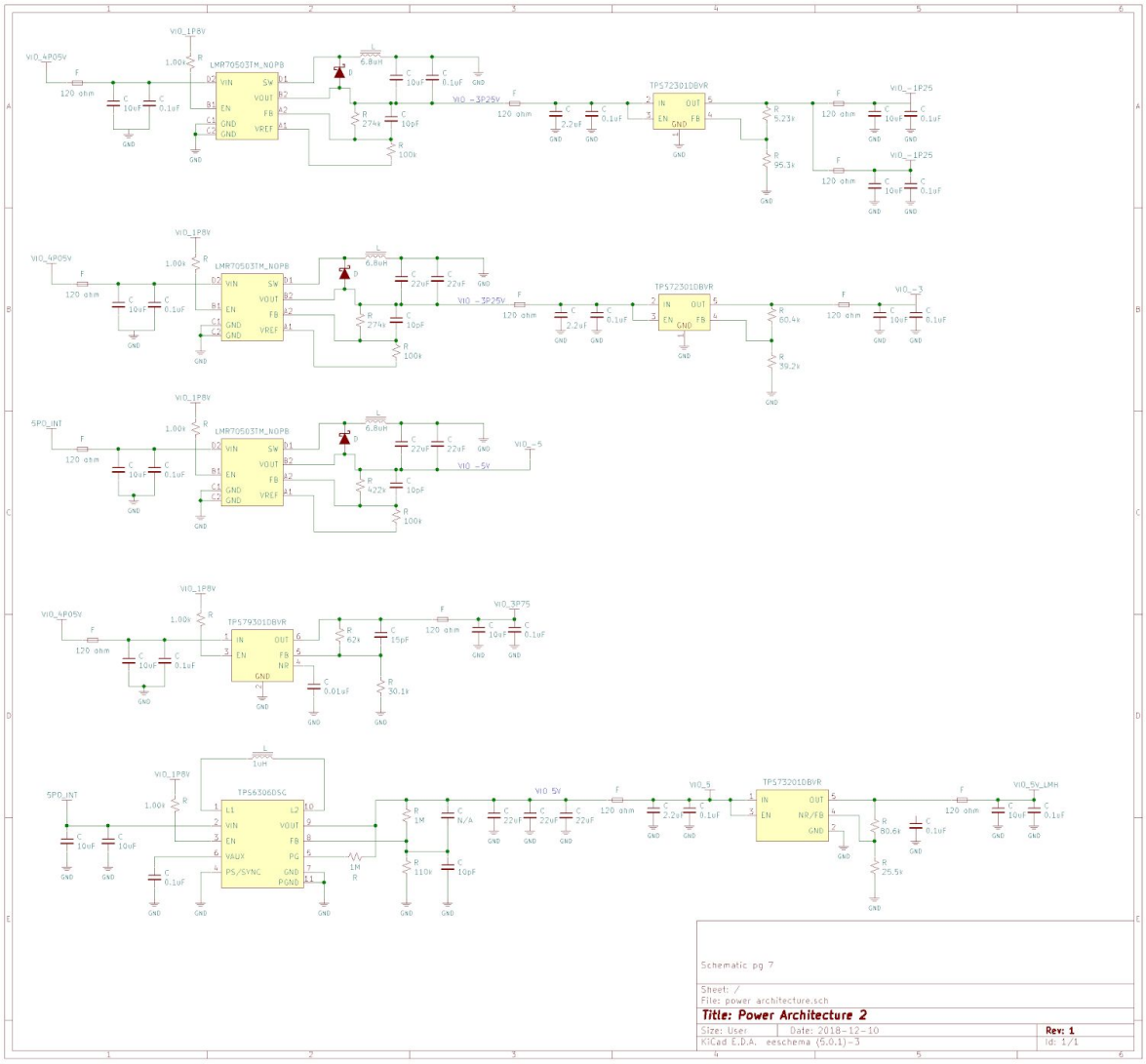
5.2.6 ADC Clock Generator/Jitter Cleaner



5.2.7 Power Architecture 1



5.2.8 Power Architecture 2



5.3 Schematic Description

5.3.1 Schematic pg 1: Front-End: RF Input & Attenuator

This is where the analog signal is inputted into the system. There will be an SMA connector attached to the beginning allowing easy connection of a function generator. There are two schottky diodes connected to positive and negative five

volts. This serves as protection against voltages larger than 10 Vpp, the maximum input voltage. Depending on what analysis is required to be done, AC or DC coupling can be used, which is chosen by the FPGA driving RF switches. If the DC coupling path is chosen, an adjustable DC offset is required. This offset is adjusted via a digital potentiometer that uses SPI. An attenuator path must then be chosen based off of the input voltage. If the input voltage is below the ADCs FSR, the 1:1 path should be chosen. However, if a larger voltage is inputted, the 10:1 path should be chosen. The attenuated signal is the sent out to the next section of the schematic.

5.3.2 Schematic pg 2: Front-End: ADC Drivers & Anti-Aliasing Filter

Once the signal is either attenuated or not, it will be fed into a LNA. The LNA allows the signal to be amplified a little bit while not introducing much noise into the signal. It is vital that we avoid adding in a lot of noise to the signal so when the signal is reconstructed it does not display excess noise. The output of the LNA is then fed to the VGA to amplify the signal once more. The VGA is controlled via SPI and may change its gain in 1 dB steps from -6-26 dB. This amplifier allows the signal to be amplified as close as it can be to the ADC's FSR. This will allow a much better reading from the ADC which is what we are aiming for. The VOCM pin receives a voltage from the ADC which compares the input common mode voltage of the ADC with the VCM output of the ADC. After the signal is amplified once more, it is fed into the anti-aliasing filter. This filters out frequencies above the maximum frequency in the system. Since our sample rate is 1 GSPS, the maximum input frequency is 500 MHz. Any frequency above that will cause aliasing in the output of the ADC and not reconstruct the signal properly. This filter also serves to filter out any high frequency noise that may have been introduced to the signal.

5.3.3 Schematic pg 3: ADC IO

Once the signal passes through the LPF it is fed into the ADC. While the HMCAD1511 allows for dual and quad channel inputs, we are only utilizing one set of differential inputs. The whole ADC is powered by 1.8 V which is provided via our power architecture. Each of these input voltages has capacitors connected to them to decouple the ADC. The clock that will be driving the ADC is outputted by the PLL chip which can adjust the sampling rate of the ADC. Our maximum is 1 GSPS but the sampling clock could be less than that. The ADC is also connected to the SPI pins on the FMC which allows the ADC to receive instructions from the user. The VCM output is fed to an amplifier and then to the

VOCM pin of the VGA. Lastly, while we are only using one differential pair for the inputs, we need to use all sixteen outputs of the ADC. This is due to the time-interleaving of the ADC, there are four ADCs inside of the HMCAD1511 and they each sample and then the output is interleaved.

5.3.4 Schematic pg 4: FMC LPC

The FMC LPC port will facilitate data transfer between the daughter board and the Zedboard. The FMC LPC connector has the capacity to support up to 68 user-defined, single-ended signals or 34 user-defined, differential pairs. This is more than enough for our application. The ADC bit and frame clocks, as well as the ADC data will come as an input through the FMC to the Zedboard. The Zedboard will provide clock reference signals and send SPI commands through the designated ports.

5.3.5 Schematic pg 5: ADC Clock Generator/Jitter Cleaner

The ADC Clock Generator/Jitter Cleaner will provide the sampling clock for the ADC. We will be inputting the FPGA clock into the PLL into the PRI_REF differential inputs. The FPGA clock is connected in series with 100 nF capacitors for decoupling. The REF_SEL pin will be then used to choose which input clock to be used. Since only one of the inputs is being used, REF_SEL will always be choosing the PRI_REF input. The PLL will be controlled via SPI as well which will allow the output to be adjusted so it is possible to sample at different rates. All of the PLL pins that require power are powered by 3.3 volts which is obtained in the power architecture circuit. Since we are only using one of the outputs we are allowed to leave the rest unconnected. The one differential output pair is then fed directly to the FPGA to give it the sampling rate.

5.3.6 Schematic pg 6: Power Architecture 1

The power architecture manipulates the input voltage to power each of the components in the front-end. Our whole system is running off of an initial input voltage of 5 volts which will be manipulated by regulators to properly power the components. The first regulator circuit is used to provide power to the PLL chip. The PLL chip runs off of 3.3 volts overall, the first regulator circuit manipulated the 5 volt input so the final outputted signal will be 3.3 volts. This is then fed directly to the PLL pins that required to be powered. The next regulator circuit is used to drive the ADC as well as the RF switches. The ADC requires a power supply of 1.8 volts and that voltage falls within the power supply range for the RF switches. Using a set of two regulators, the second one which defines the final

output voltage, 1.8 volts is obtained and routed both to the ADC as well as the RF switches.

5.3.7 Schematic pg 7: Power Architecture 2

This schematic is a continuation of the power architecture schematic. This schematic focuses on providing power to the amplifiers and the negative voltage provided to the schottky diodes for protection. The first regulator circuit is providing the negative voltage to both the LNA and VGA. This is achieved by using two negative voltage regulators. The next regulator circuit achieves a negative 3 volt output. This is then fed to the DC offset amplifier as well as the digital potentiometer. The third regulator circuit is used to acquire the negative 5 volts that is connected to one of the schottky diodes for protection. The last two regulator circuits also power amplifiers. The 3.75 volts is fed to the LNA and VGA for their positive rail while the positive five volts is fed to the digital potentiometer and op-amp for their positive rail and high input.

5.4 Descriptions of Major Components

5.4.1 Front-end Components

5.4.1.1 HMCAD1511

- 8-bit low power analog-to-digital converter (ADC).
- Time-interleaving is utilized between the 4 internal ADCs to increase sampling rate.
- Mode and configuration settings programmable through SPI interface
    - Adjustable low jitter clock divider
    - Digital fine gain adjustment

5.4.1.2 LMH5401

- Fully-differential amplifier for 50 ohm input single-ended to differential conversion
- 6 GHz bandwidth when configured for a single-ended to differential gain of 4 V/V
- Great linearity performance from DC-2 GHz
- Low input-voltage noise

5.4.1.3 LMH6401

- Digitally controlled variable-gain amplifier driving the ADC
- Gain control is carried out via a SPI interface from -6 dB to 26 dB
- Achieves a 3-dB bandwidth of 4.5 GHz at 26-dB gain
- Great linearity performance from DC-2 GHz

5.4.1.4 CDCE62005

- Clock Generator, Jitter Cleaner with Integrated Dual VCOs
- Input frequencies: 40kHz to 500 MHz
- Output Frequency Ranges from 4.25 MHz to 1.175 GHz in Synthesizer Mode

- Output Frequency up to 1.5 GHz in Fan-Out Mode
- Output frequency controlled by SPI

5.4.1.5 OPA376

- Low-Noise, Low Quiescent Current, Precision Operational Amplifier
- Gain bandwidth product is 5.5 MHz
- Rail-to-Rail Input and Output
- Used in a loop to compare the input common mode voltage of the ADC to the output common mode voltage of the ADC and applying that output to the output common mode control pin

5.4.1.6 ADG936

- Wideband RF Dual SPDT
- 4 GHz maximum frequency
- Low Power Consumption
- Low Insertion Loss
- CMOS/LVTTL control logic

5.4.1.7 AD8400

- 1kΩ, 10kΩ, 50kΩ, 100kΩ Digital Potentiometer
- 256 positions
- Controlled via SPI
- Used to adjust DC offset

5.4.1.8 LMH6559

- High-Speed, Closed-Loop Buffer
- 1750 MHz Small Signal Bandwidth
- 4580V/μs Slew Rate
- Used for DC offset

5.4.2 Power Components

5.4.2.1 TPS2400DBVR

- Overvoltage Protection Controller
- 100-V Overvoltage Protection
- Overvoltage Turnoff Time Less than 1 μs
- 1-mA Maximum Static Supply Current

5.4.2.2 TPS54327DDA

- Output Single Synchronous Step-Down Switcher
- Output Voltage Range: 0.76 V to 7 V
- Wide VIN Input Voltage Range: 4.5 V to 18 V
- Low-Output Ripple

5.4.2.3 LP3878MR-ADJ

- Adjustable Voltage Regulator
- Input Supply Voltage: 2.5 V to 16V
- Output Voltage Range: 1 V to 5.5 V
- Very Low Output Noise
- Overtemperature and Overcurrent Protection

5.4.2.4 LM26400YSDE

- Input Range Buck Regulator
- Input Voltage Range of 3 V to 20 V
- Output Voltage Down to 0.6 V
- 175-mΩ NMOS Switch
- Frequency Foldback Protection

5.4.2.5 LP38513TJ-ADJ

- Linear Voltage Regulator
- 2.25V to 5.5V Input Voltage Range
- Adjustable Output Voltage Range of 0.5V to • Servers 4.5V
- 3.0A Output Load Current
- Over-Temperature and Over-Current Protection

5.4.2.6 LMR70503TM

- Buck-Boost Converter For Negative Output Voltage
- 2.8 V to 5.5 V Input Voltage Range
- Adjustable Output Voltage: -0.9 V to -5.5 V
- 320 mA Switch Current Limit

5.4.2.7 TPS72301DBVR

- Negative Output Low-Dropout Linear Regulators
- Ultralow Noise
- High PSRR
- Thermal and Over-Current Protection

5.4.2.8 TPS63060DSC

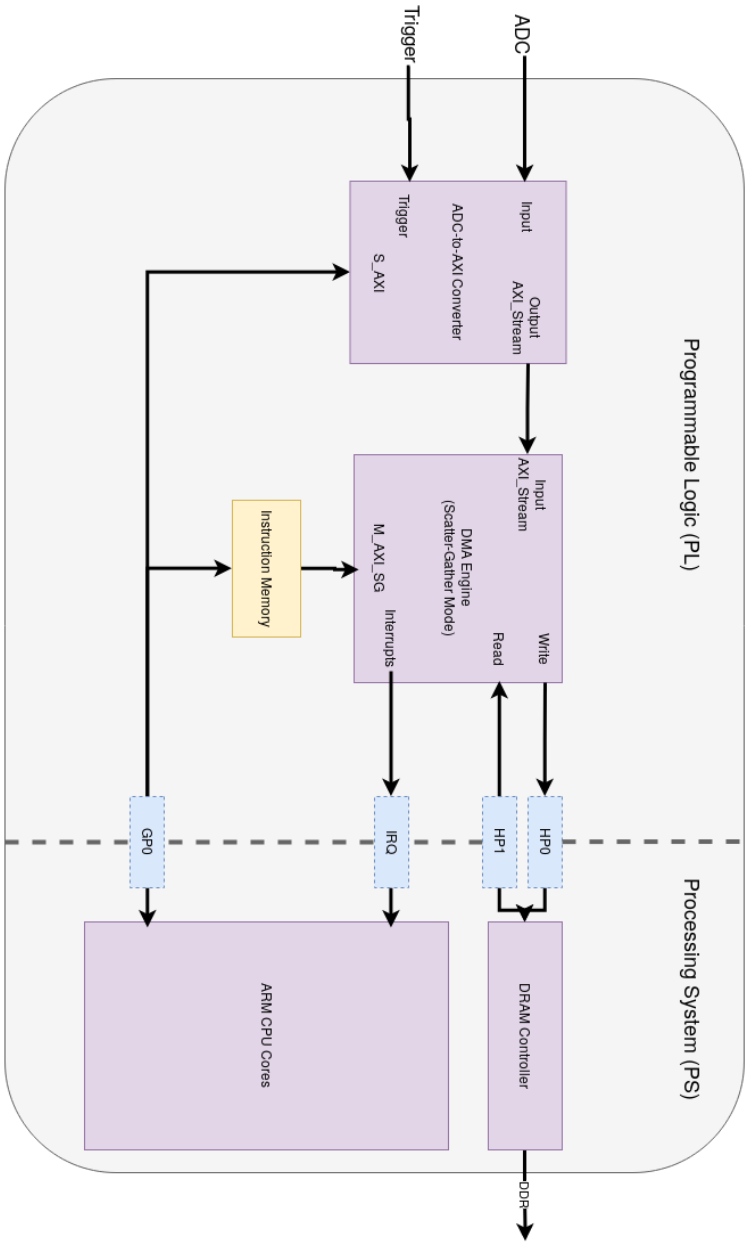
- Buck-Boost Converter
- Input Voltage Range: 2.5 V to 12 V
- Fixed and Adjustable Output Voltage Options from 2.5 V to 8 V
- Device Quiescent Current: < 30 µA
- Overtemperature and Overvoltage Protection

5.4.2.9 TPS73201DBVR

- Low-Dropout Regulator
- Input Voltage Range: 1.7 V to 5.5 V
- Low Noise

- Multiple Output Voltage Versions
    - Fixed Outputs of 1.2 V to 5 V
    - Adjustable Outputs from 1.2 V to 5.5 V
    - Custom Outputs Available

5.5 Firmware Block Diagram

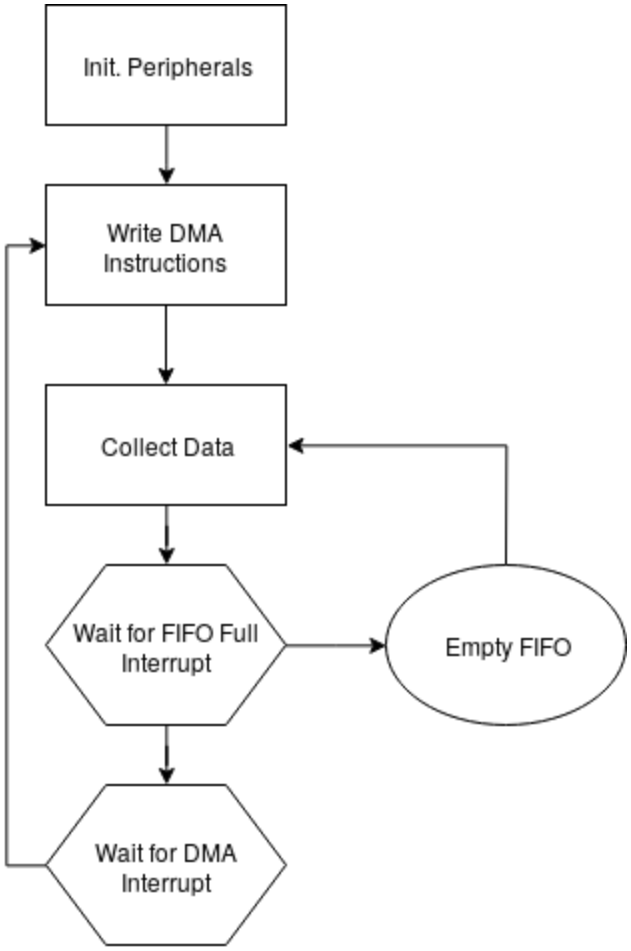


5.6 Firmware Description

The PL Layer will contain the ADC-to-AXI IP which will convert the incoming ADC data into an AXI Stream compliant format. The stream will then be sent to the DMA Engine which will transfer the data into DRAM via the PS's HP0 and HP1

ports. These ports provide high throughput data access. The DMA is programmed through the ARM CPU via the GP0 port. The ARM processor sends a group of read/write instructions to a BRAM block which the DMA engine can read from. Once the DMA has processed all the instructions, it can raise an interrupt to the processor, which will then send a new batch of instructions. This method reduces the amount of interrupts received by the CPU and allows the SoC to operate more efficiently.

5.7 Firmware Data Acquisition State Diagram



5.8 BOM & Cost Estimation

Description	Part #	Manufacturer	Cost

1 GSPS 8-bit Analog to Digital Converter	HMCAD1511	Analog Devices	$65
8 GHz Low Noise Amplifier	LMH5401	TI	$15
4.5 GHz Variable-Gain Amplifier	LMH6401	TI	$15
High-Speed Closed-Loop Buffer	LMH6559	TI	$3
Low-Noise Op-Amp	OPA376	TI	$2
Clock Generator/Jitter Cleaner	CDCE62005	TI	$9.94
RF Dual SPDT	ADG936	Analog Devices	$1.78
Overvoltage Protection Controller	TPS2400BVR	TI	$2.14
Switching Voltage Regulators	TPS54327DDA	TI	$2.02
LDO Voltage Regulators	LP3878MR-ADJ	TI	$2.49
Wide Input Range Buck Regulator	LM26400YSDE	TI	$4.89
LDO Voltage Regulators	LP38513TJ-ADJ	TI	$2.65
Switching Voltage Regulators	LMR70503TM	TI	$3.08
LDO Voltage Regulators	TPS72301DBVR	TI	$2.72
Switching Voltage Regulators	TPS63060DSC	TI	$2.34

LDO Voltage Regulators	TPS73201DBVR	TI	$1.48
Digital Potentiometer	AD8400	Analog Devices	$1.24
Schottky Diode	MMBD352WT1G	ON Semiconductor	$.28
Schottky Diodes & Rectifiers	MBRS540T3G	ON Semiconductor	$.56
Power MOSFET	CSD18532Q5B	TI	$2.18
Slide Switches	L101011MS02Q	C&K	$1.97
Power Jack	PJ-202AH	CUI Inc.	$.77
Common Mode Chokes	ACM7060-301-2PL-TL	TDK	$2.03
SMA Connector	5-1814832-1	TE Connectivity	$2.49

By summing the parts above we get a total of $147.05. There will be multiples of some of the smaller and cheaper components above, as well as resistors, capacitors, and inductors. Those parts are usually not too expensive so they would not add too much to the cost. With this information, we are able to say that we will stay under $300 for the components needed to make our board work.

5.9 PCB Fabrication & Assembly

We plan on beginning the PCB layout design over winter break so that we won't be rushed during and so that we will have plenty of time for design verification before fabrication. As of now we are considering using one of two different softwares for the PCB layout. Kicad being the first, and this option has plenty of online resources. The second software is PCB123 which was suggested to us by an employee at Lockheed Martin. We know less about this software so we would need to spend more time learning how to use it, but if it is better than Kicad it may be a better alternative. PCB123 also manufactures PCB boards so we would be able to send them our design once it is finished to be manufactured. While getting a board manufactured would not be overly expensive, the assembly cost is where the price can significantly increase. We are exploring several different options for board fabrication and assembly with the ideal

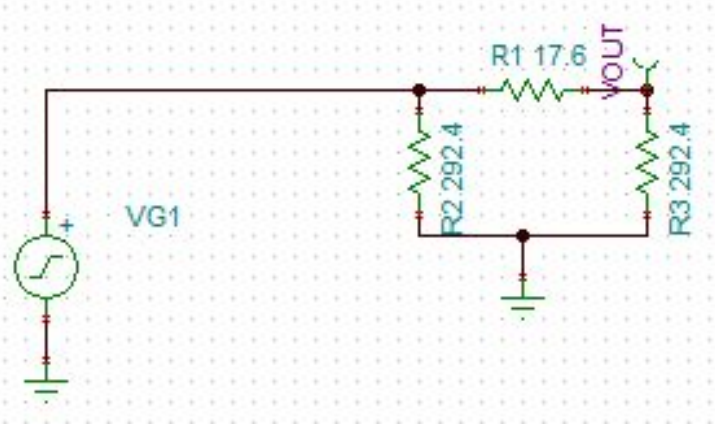
situation is having a company sponsor our project an making the boards for free. The more likely situation is we might have to pay a one time fixed price and a company would fabricate several boards for us when needed. If the free option doesn't pan out we still have several decent alternatives.

One of our biggest concerns for PCB assembly was our limited soldering experience. Especially when many of our components use the QFN form factor and are only several millimeters wide, we had little confidence in our ability to solder them correctly to a PCB. This is the primary reason we are looking into getting the PCB manufactured and assembled by a company. This presents a problem as the price for the full board would be much more expensive, over $100. However, this solution may actually save us money in the long run. Since our soldering is not as refined as experts in the field, we may end up ruining components and never knowing what we did wrong, thus requiring us to purchase the components again. Having a professional do this greatly reduces the risk but increases the cost. This is a situation we will be doing more research into to ensure the best decision is made.

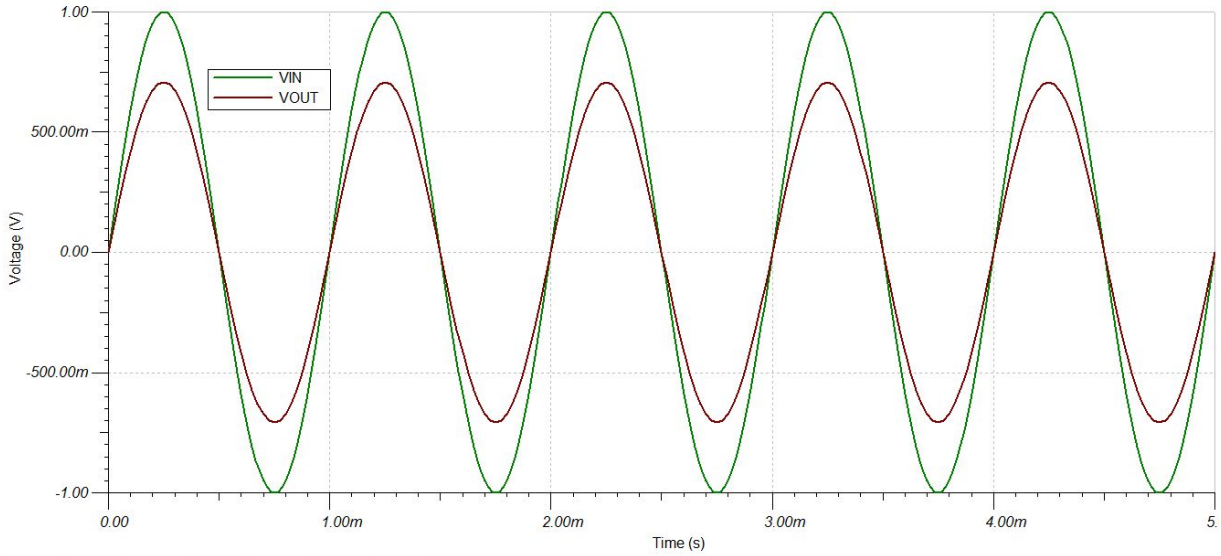
6.0 Prototyping Progress

6.1 SPICE Simulations

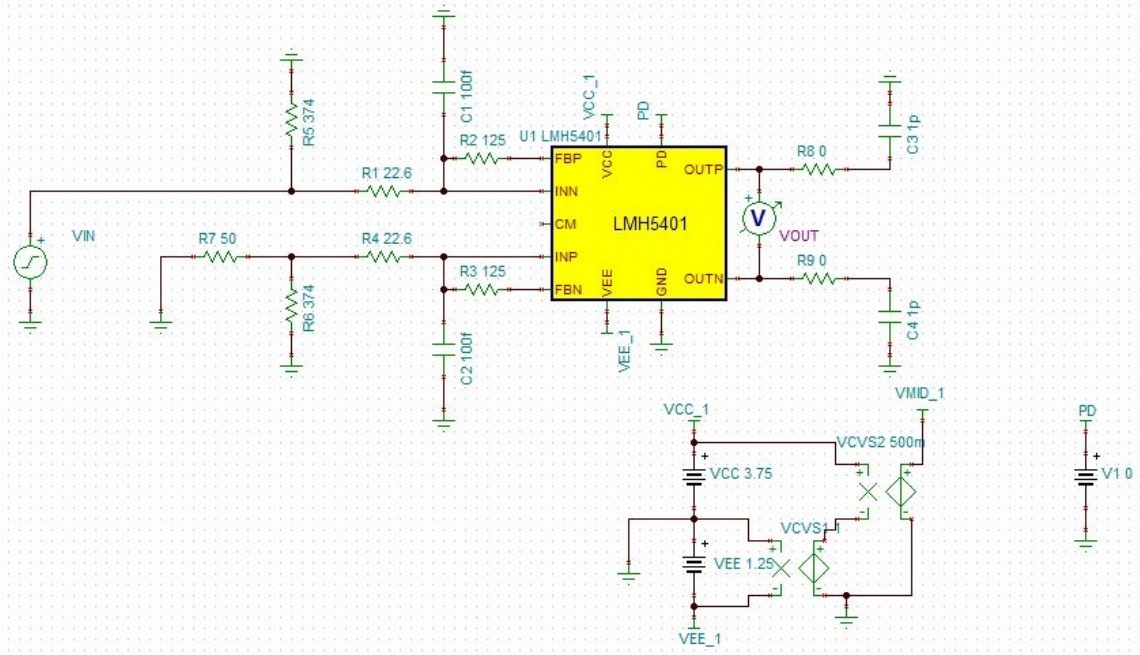
6.1.1 Attenuator



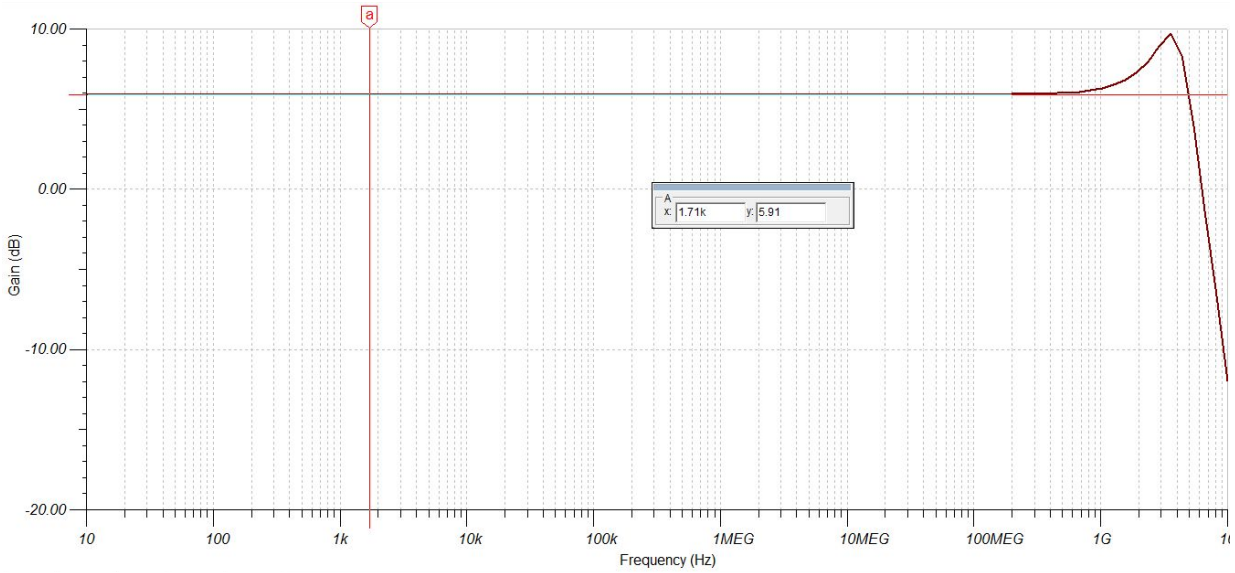
Voltage vs. Time (3dB attenuation)



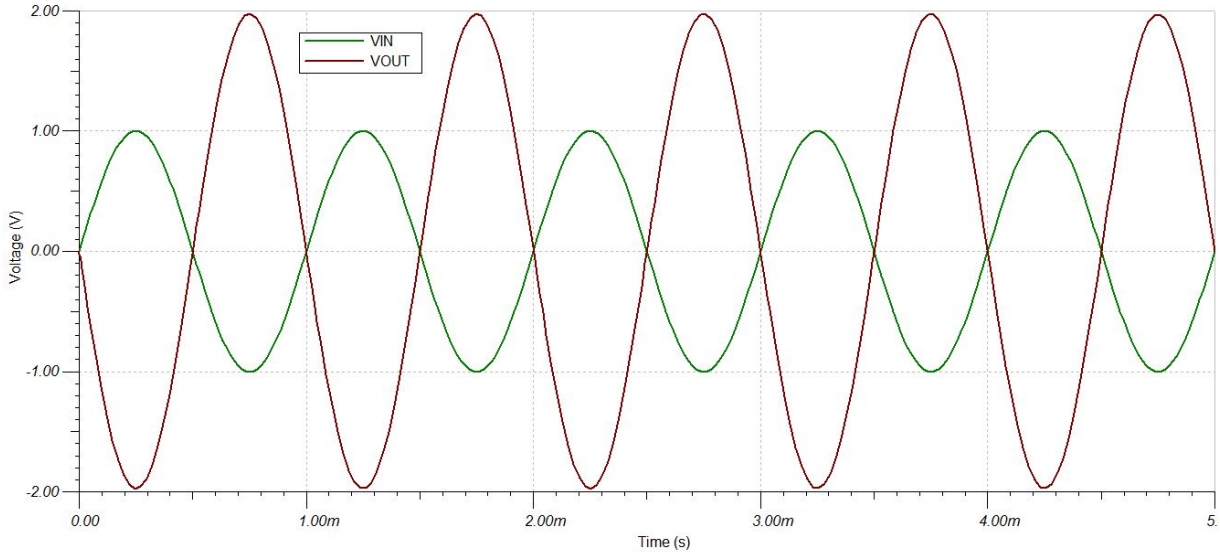
6.1.2 LMH5401



Gain vs. Frequency (LMH5401)

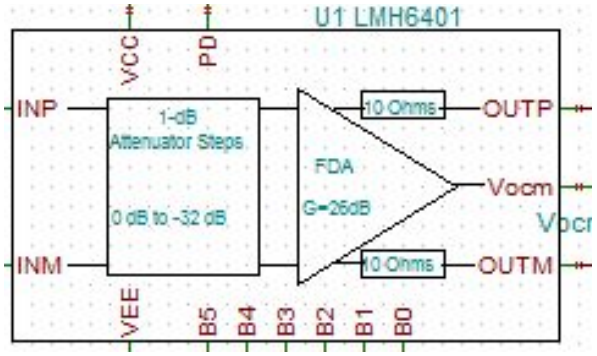


Voltage vs. Time (LMH5401)

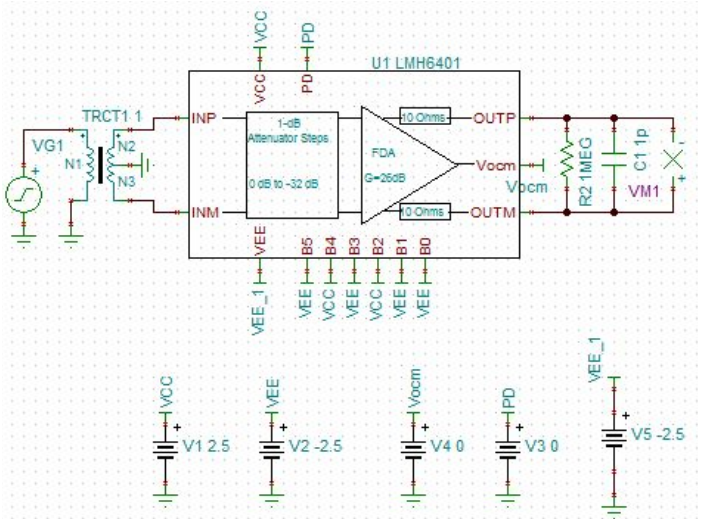


6.1.3 LMH6401

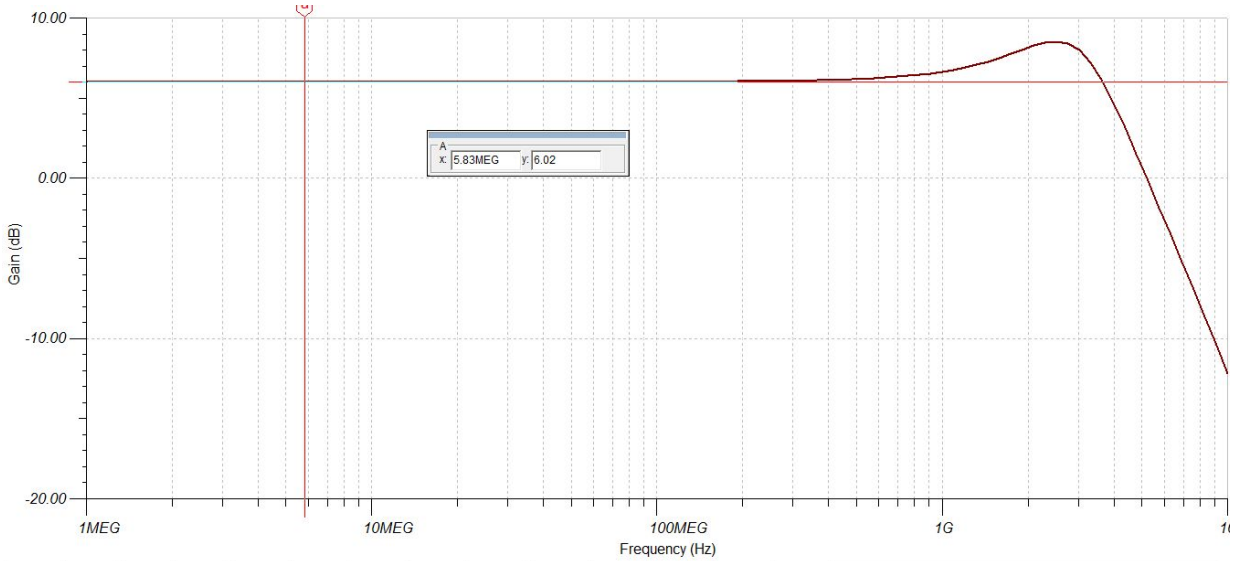
Tina-TI Symbol



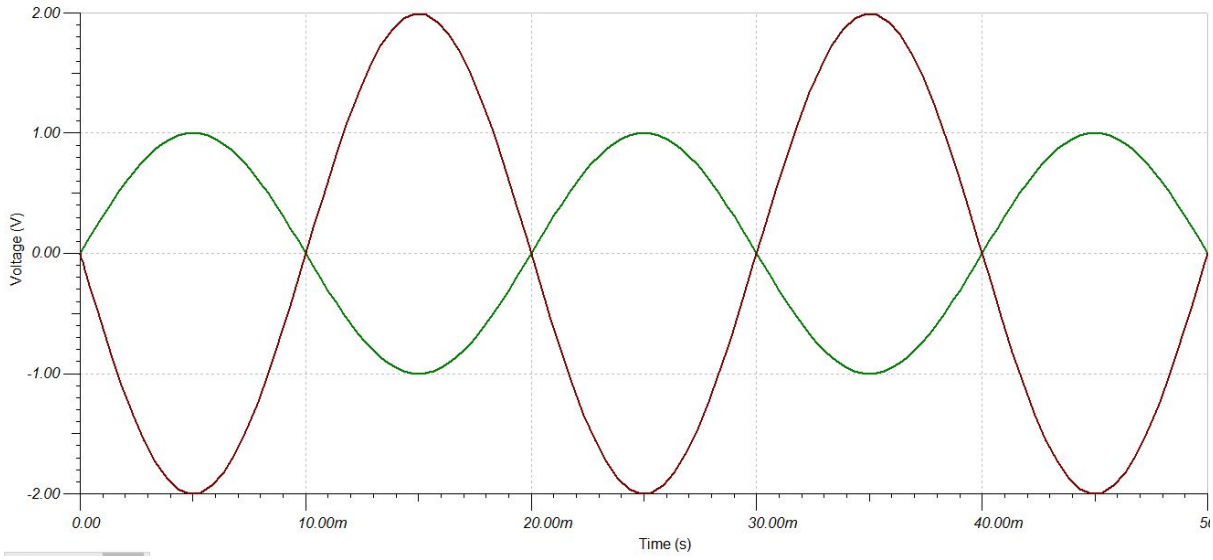
B0-B5 control gain of LMH6401

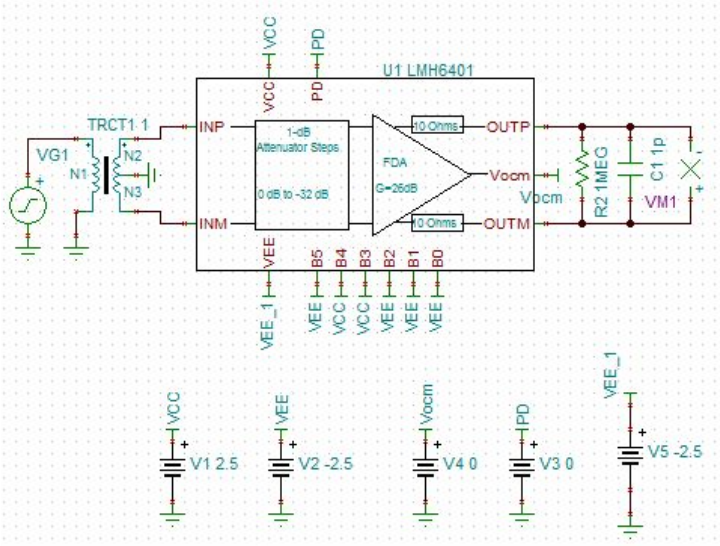


6 dB Gain vs. Freq (LMH6401)

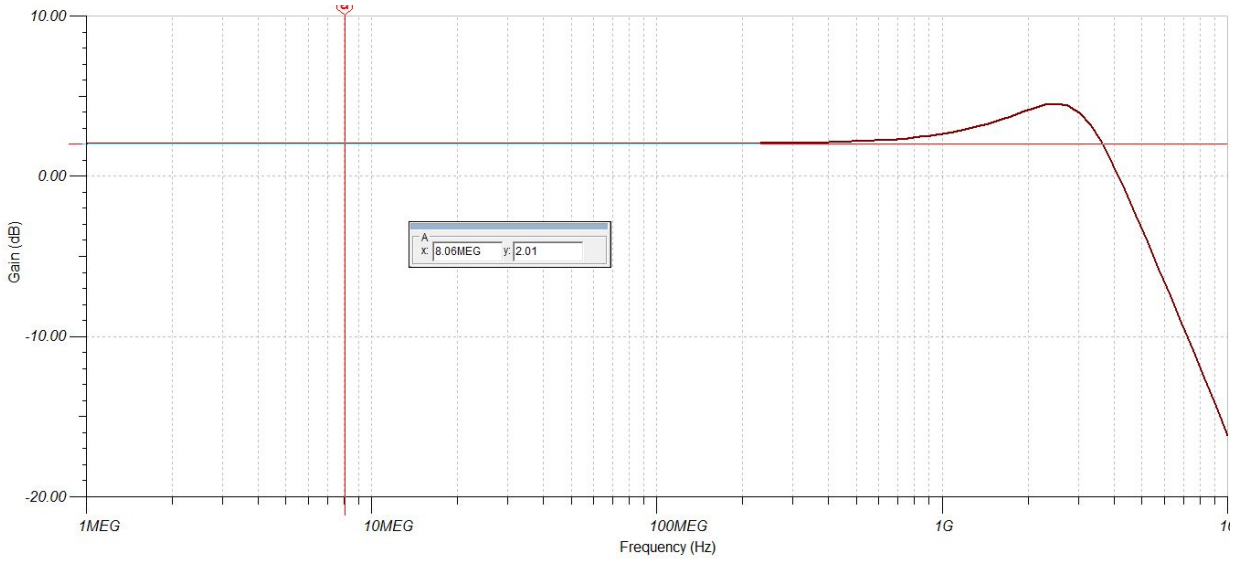


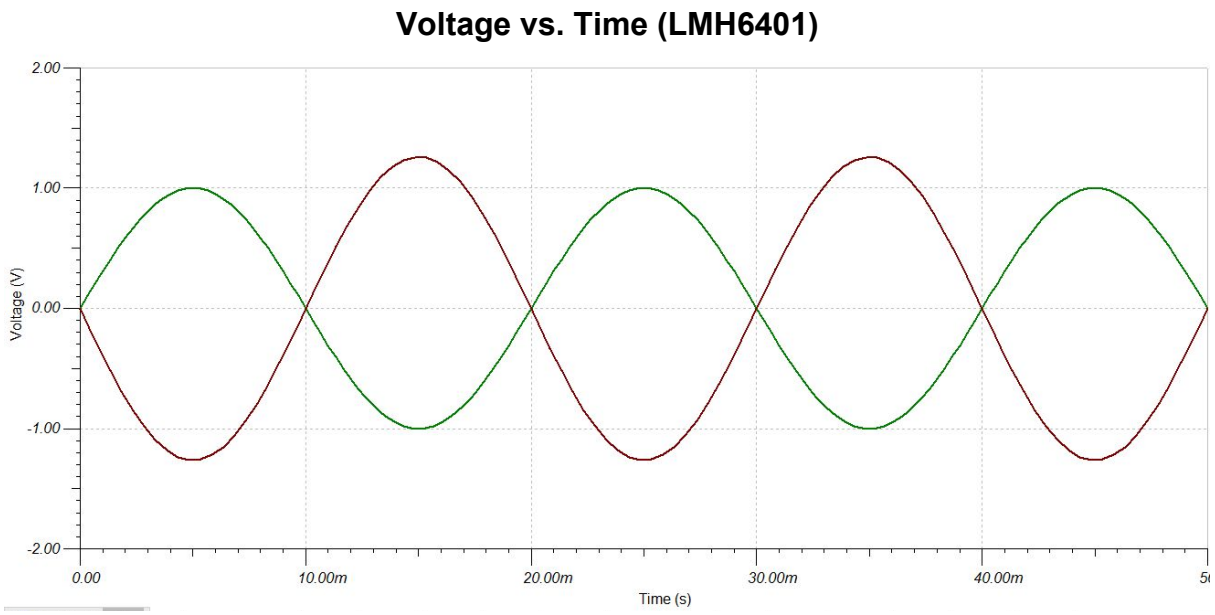
Voltage vs. Time (LMH6401)



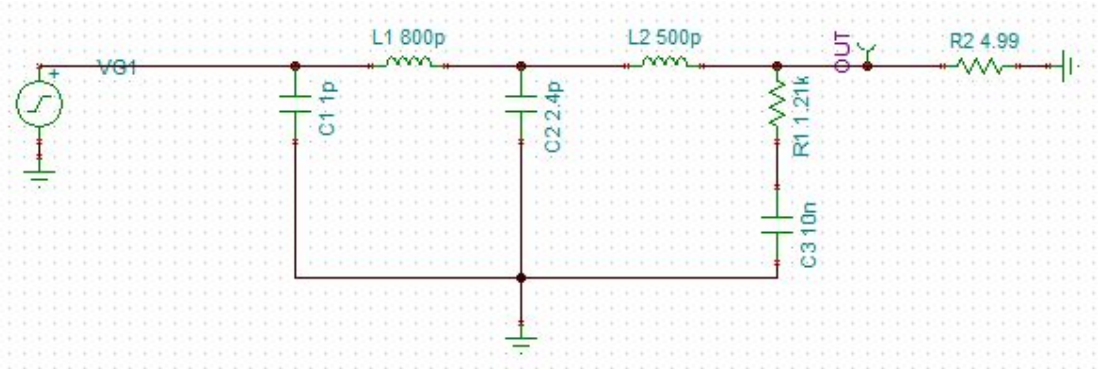


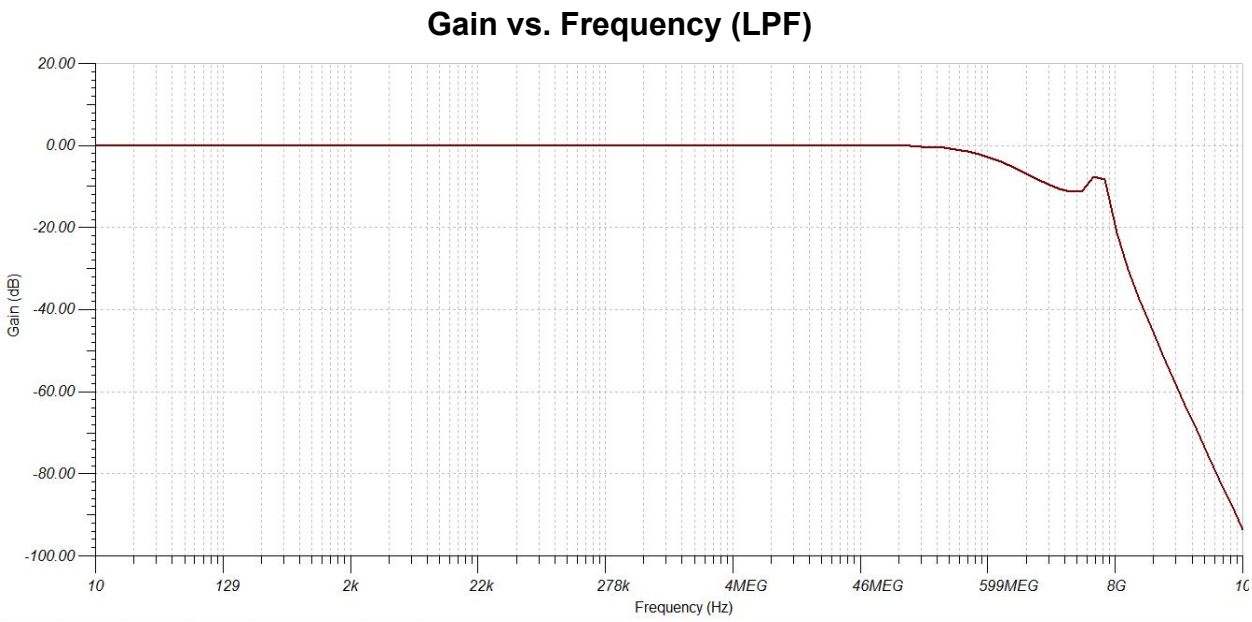
2 dB Gain vs. Frequency (LMH6401)



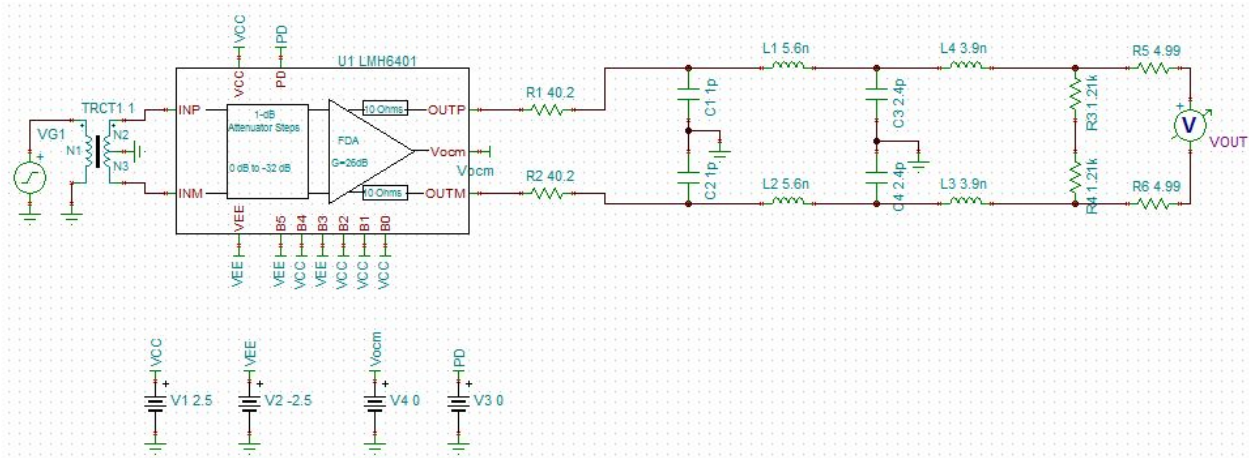


6.1.4 Non-aliasing Chebyshev LPF

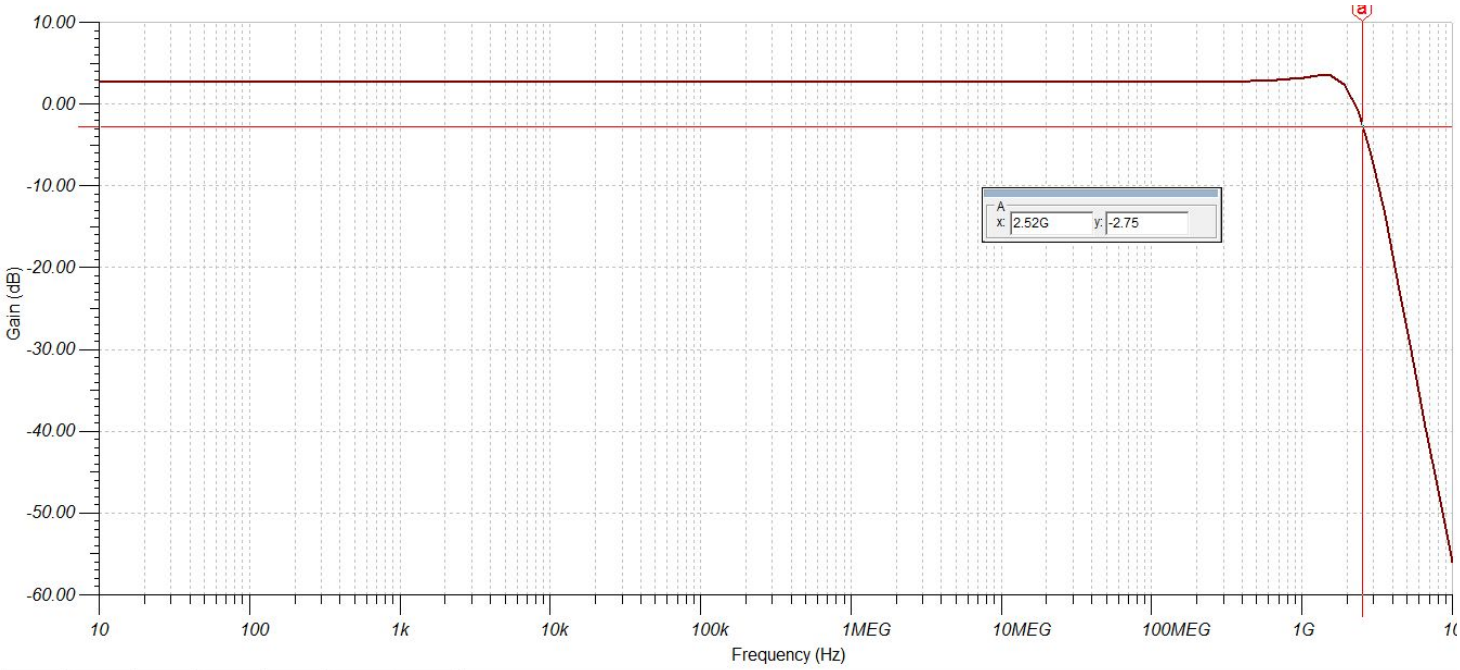




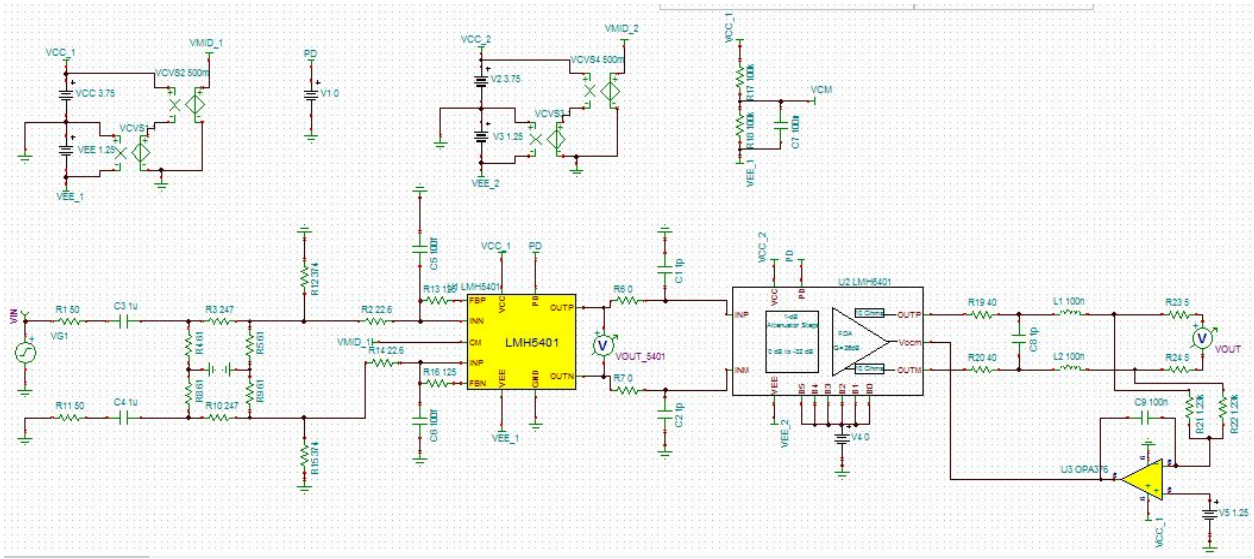
6.1.5 Aliasing Chebyshev LPF



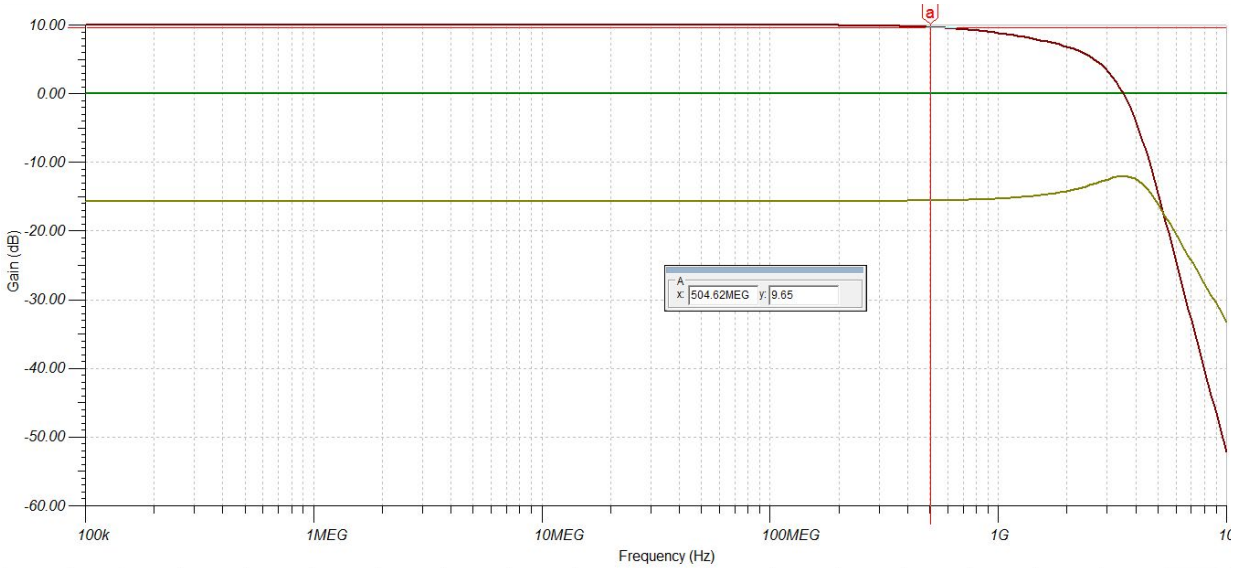
Gain vs. Frequency (Aliasing LPF)



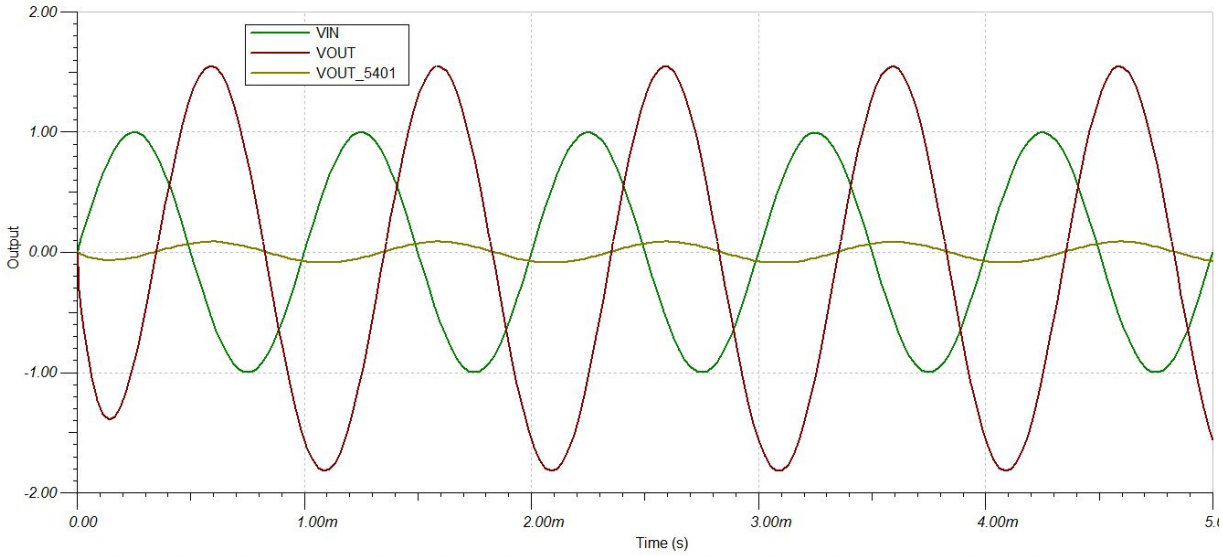
6.1.6 Full System AC Coupling



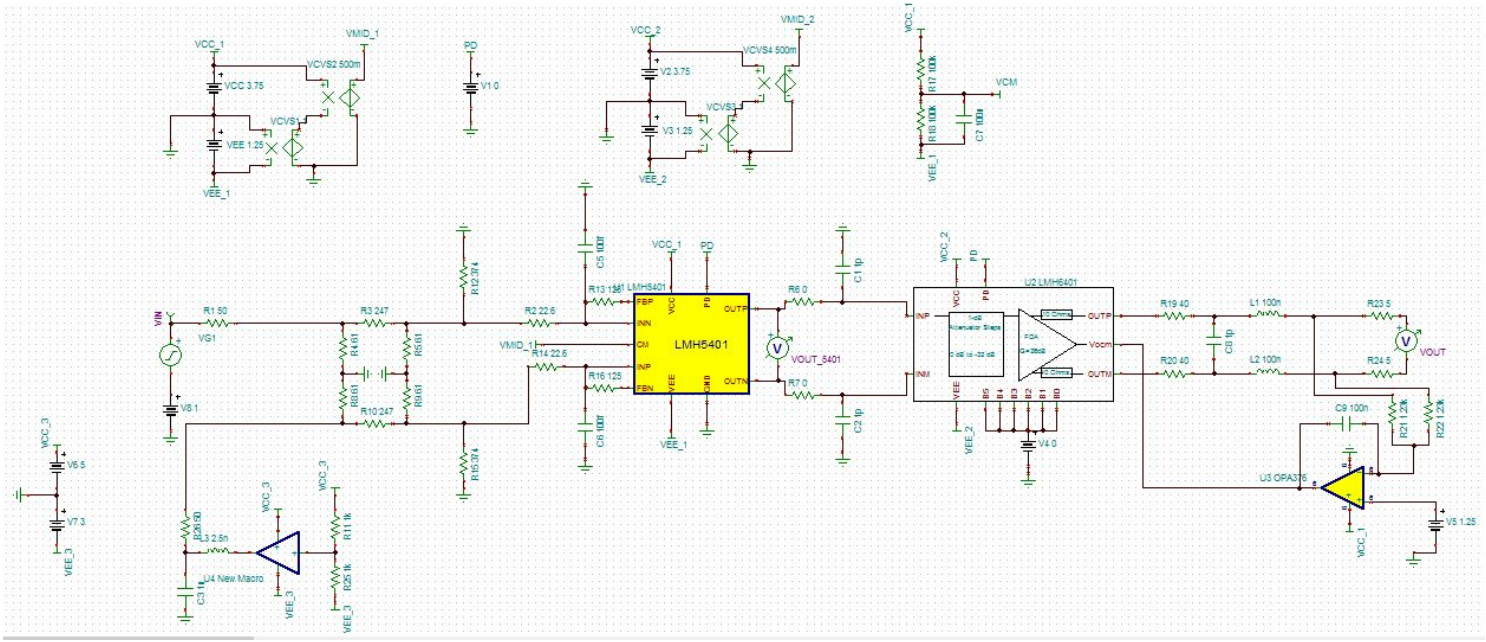
Gain vs. Frequency (AC Coupled)



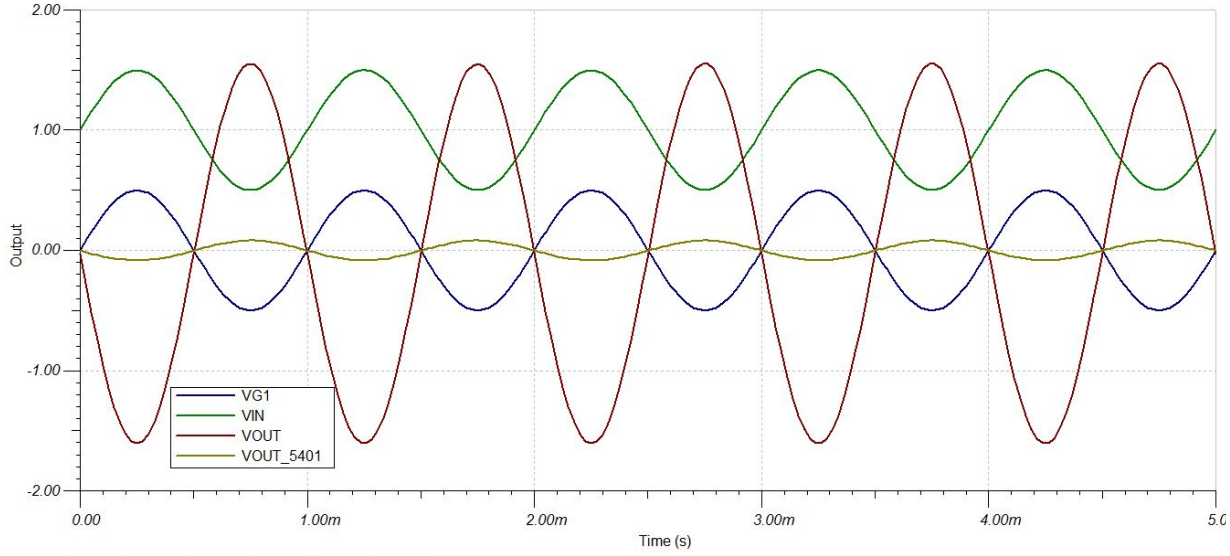
Voltage vs. Time (AC Coupled)



6.1.7 Full System DC Coupling



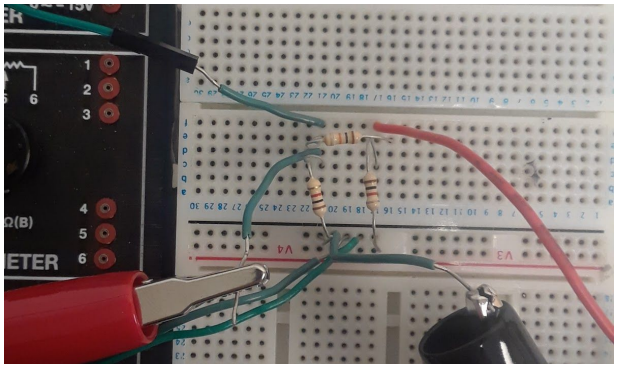
Voltage vs. Time (DC Coupled)



6.2 Component Evaluation Boards

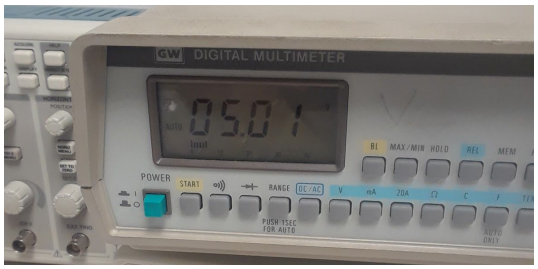
6.2.1 Attenuator

DC Test



Note: Resistor values used for this test will not be used for the full system

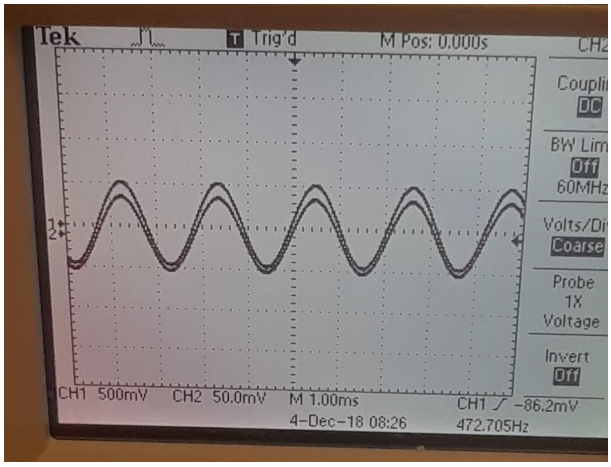
5 Volt Input Voltage



452 mV Output

AC Test

Note: Circuit stays the same



Larger signal is input voltage, around 1 Vpp. Lower signal is output.

7.0 Testing Plan

7.1 Component Specific Tests

Each component will have a test schematic and PCB layout designed for them. These evaluation models will allow us to conduct tests to confirm the performance of each component is as expected. The list of components and corresponding tests are as follows:

7.1.1 Attenuator Tests:

- Run DC voltage through attenuator and confirm drop in voltage.
- Run AC signal through attenuator and confirm same drop in voltage.

7.1.2 LNA LMH5401 Tests:

- Build test circuit as described in datasheet
- Run AC signal through LNA with varying frequencies and voltages
- Confirm LNA amplifies signal at output
- Confirm LNA works up to specified frequency and confirm at what voltage will the LNA saturate

7.1.3 VGA LMH6401 Tests:

- Build test circuit provided in datasheet
- Run AC signal through VGA
- Confirm that the output has been amplified by maximum gain
- Use SPI to vary the gain and confirm output signal varies as well
- Confirm the VGA can accept frequencies up to its maximum

7.1.4 PLL/Jitter Cleaner CDCE62005 Tests:

- Build circuit specified in eval board user guide
- Input external clock and vary phase and frequency
- Confirm that the output is multiplied by frequency
- Adjust clock multiplier via SPI
- Confirm that output frequency changes

7.1.5 ADC HMCAD1511 Tests:

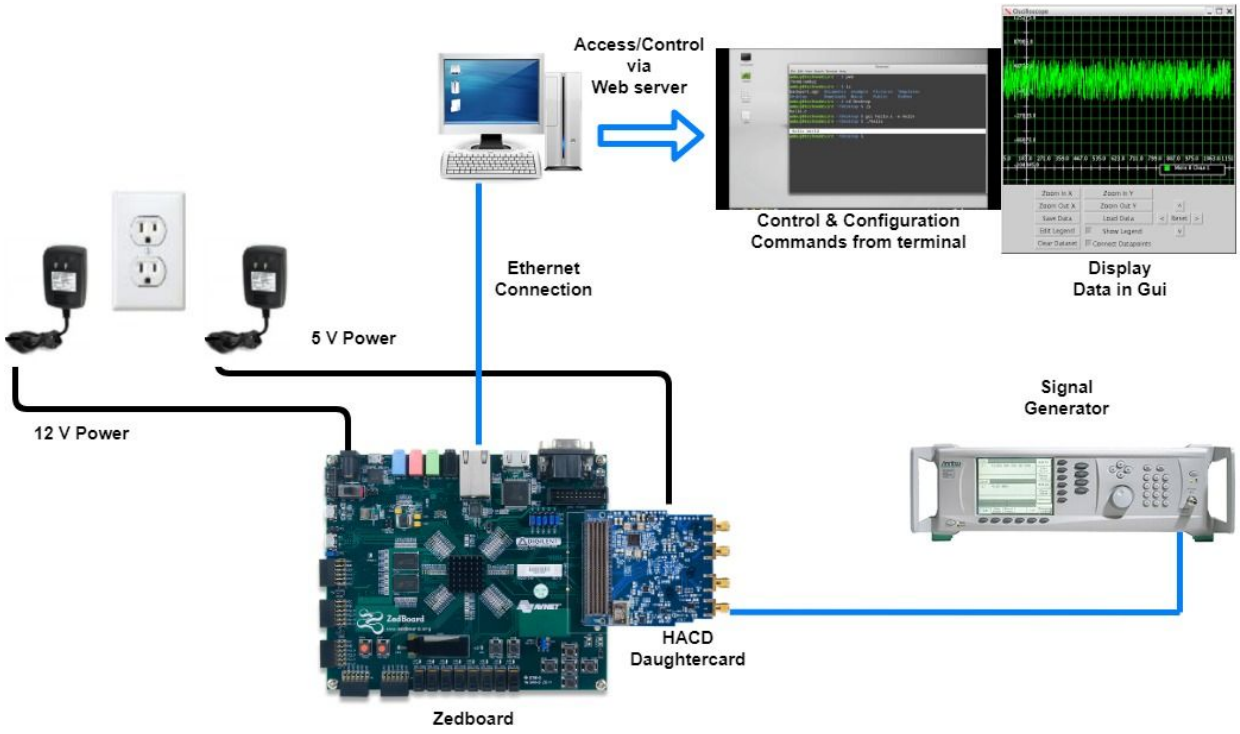
- Using eval board, input an AC signal
- Output of ADC is fed to FPGA with firmware
- FPGA processes data and shows output

7.2 Full System Test

Once the full system is built and connected we will run a variety of tests with different inputs to confirm that our open scope works under the maximum and minimum constraints we have provided. These tests will be helpful in observing that the full system works as expected. The list of tests are as follows:

- Input sine wave with 50 mVpp to confirm that is the minimum possible voltage
- Input sine wave with 10 Vpp to confirm that it is the maximum possible voltage
- Input varying frequencies from DC-500 MHz to confirm that we will not get any aliasing in that range
- Input different waves i.e. square and triangle waves and observe the output
- Adjust the input voltage and thus adjust the gain of the VGA to confirm the voltage can be adjusted to the FSR of the ADC
- Adjust the clock fed to the PLL and the multiplier to sample at different rates

7.3 System Test Setup



Example Test Setup

For our test setup we will be connecting the front-end board to the Zedboard via the FMC connectors. An example of how our setup will look like is in the picture above. We will have the PCB board being powered by an external 5 volt power supply and also feeding a signal in through an SMA connector. The Zeboard will also be powered by an external power supply of 12 volts and being connected to the user computer via an ethernet cable. With this setup we will be able to run the many tests that we have established in the previous section.

8.0 Tasks for ECE 493

8.1 Approach Description

In 493 we will be breaking up the building part of the project into separate parts. We will start over winter break by focusing on the front-end and all that needs to be done to get it fully functional. Simultaneously, we will working on the firmware to get that fully functional as well. Most of the front-end should be finished by the start of the spring semester, this gives us ample time to verify it all before manufacturing. Once the front-end is solidified, the focus will then shift to the firmware while the front-end is manufactured. We will need to run some extra tests on the firmware besides just using

the ADC Easyboard. Once we receive the PCB of the front-end, we will need to run tests and confirm that all the voltages at certain nodes are what we expect. When that is confirmed and the firmware finalized, we will then run a full system test.

8.2 Task Decomposition

8.2.1 Component Testing

- Reach out to companies for potential sample evaluation modules for larger components
- If received, run test for each component to verify the results match the theoretical results
- If not received, design small test circuits with PCB layouts, get these manufactured, and run tests
- Record all data received from these tests

8.2.2 PCB Design

- Go over PCB design for each component provided in the datasheet
- Using either PCB123 or Kicad, design the PCB for the full front-end
- Verify our design by posting to forums and reaching out to experienced professionals
- Send PCB to be manufactured and potentially soldered by another company

8.2.3 PCB testing

- Once PCB is received, run different signals and voltages through it
- Confirm voltages and current at certain nodes are as expected
- Confirm the output of the ADC is what we expect
- Try all variation of RF switches to make sure each path is accessible

8.2.4 Firmware design

- Build firmware on Easyboard to understand how the ADC will work. SInce our design is functionally similar to the Easyboard, we will not need to make major revisions to our firmware when transitioning to our final design.
- We will use Vivado Design Suite to create and test our firmware

8.2.5 Firmware Testing

- We will utilize the Easyboard and verify the functional aspects of our firmware.
- We will design a software sample generator IP to simulate our front end board to test our firmware
- We will create test benches to verify our custom IPs

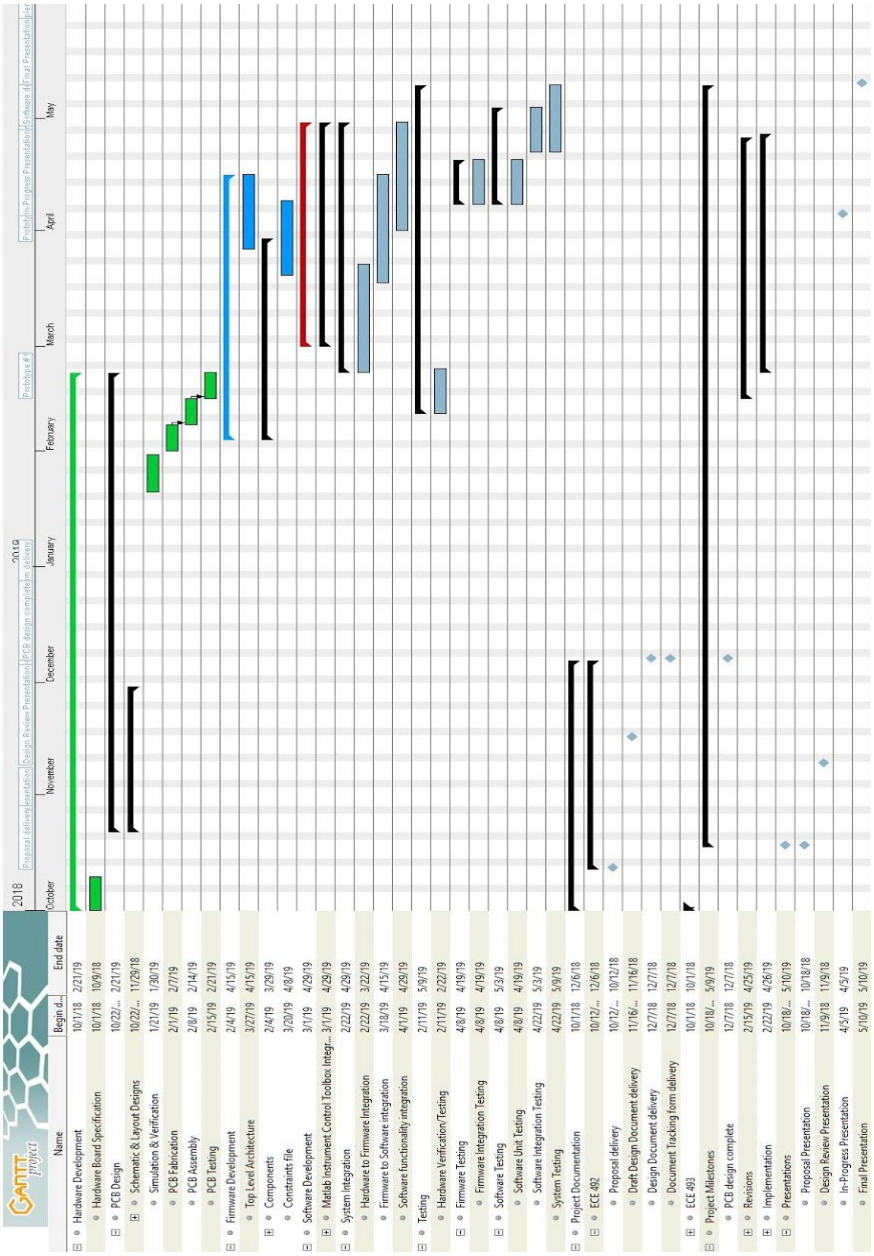
- We will use the software logic analyzers to verify our firmware design in real time

9.0 Schedule and Milestones

9.1 Major Milestones & Goals

One of our major milestones we plan to reach is the PCB design for the front-end to be finished by the end of winter break. That will then give us plenty of time to verify that the PCB design would be functional. We also plan to finish off component testing before the start of the semester so the rest of the semester can be putting it all together. Next, we plan on sending the PCB to be manufactured after we verify it. This should be around February and the manufacturing process will take from 1 to 2 weeks. The whole firmware design is planned to be done in February or March, this will be around the time the PCB of the front-end is done and we can begin running full system tests. We hope to be done with the full system and obtain results a month or so before the presentation. This will give us time to potentially enhance anything or debug any potential issues with the design.

9.2 Gantt Chart



10.0 References

[1] “OPENADC Product Datasheet”, OpenADC datasheet.
[2] “*OpenADC Project*.” *Assembla*. (n.d.). [Online]. Available: https://app.assembla.com/spaces/openadc/wiki/Home [Accessed Nov. 29, 2018].
[3] Analog Devices, “High Speed Multi-Mode 8-Bit 30 MSPS to 1 GSPS A/D Converter,” HMCAD1511 datasheet.

[4] Hittite, “Hardware User Guide, EVAL01-HMCAD1520,” 584-EKIT01-HMCAD1511 datasheet.
[5] Texas Instruments, “8-GHz, Low-Noise, Low-Power, Fully-Differential Amplifier,” LMH5401 datasheet.
[6] Texas Instruments, “DC to 4.5 GHz, Fully-Differential, Digital Variable-Gain Amplifier,” LMH6401 datasheet.
[7] Texas Instruments, “50-Ohm 2-GHz Oscilloscope Front-end Reference Design,” TIDA-00826 Design Guide.
[8] Texas Instruments, “3:5 Clock Generator, Jitter Cleaner with Integrated Dual VCOs ,” CDCE62005 datasheet.
[9] Texas Instruments, “Low-Noise, Low Quiescent Current, Precision Operational Amplifier e-trim Series,” OPA376 datasheet.
[10] Texas Instruments, “High-Speed, Closed-Loop Buffer,” LMH6559 datasheet.
[11] Analog Devices, “Wideband 4 GHz, 36 dB Isolation at 1 GHz, CMOS, 1.65 V to 2.75 V, Dual SPDT,” ADG936 datasheet.
[12] Analog Devices, “1-/2-/4-Channel Digital Potentiometers,” AD8400 datasheet.
[13] Texas Instruments, “Overvoltage Protection Controller,” TPS2400 datasheet.
[14] Texas Instruments, “60-V N-Channel NexFET™ Power MOSFETs,” CSD18532Q5B datasheet.
[15] CUI Inc, “DC Power Jack,” PJ-202AH datasheet.
[16] Texas Instruments, “3-A Output Single Synchronous Step-Down Switcher With Integrated FET,” TPS54327 datasheet.
[17] Texas Instruments, “Micropower 800-mA Low-Noise "Ceramic Stable" Adjustable Voltage Regulator for 1-V to 5-V Applications,” LP3878-ADJ datasheet.
[18] Texas Instruments, “Dual 2-A, 500-kHz Wide Input Range Buck Regulator,” LM26400Y datasheet.
[19] Texas Instruments, “3A Fast-Transient Response Adjustable Low-Dropout Linear Voltage Regulator,” LP38513-ADJ datasheet.
[20] Texas Instruments, “SIMPLE SWITCHER® Buck-Boost Converter For Negative Output Voltage in µSMD,” LMR70503 datasheet.
[21] Texas Instruments, “200mA Low-Noise, High-PSRR Negative Output Low-Dropout Linear Regulators,” TPS72301DBVR datasheet.
[22] Texas Instruments, “High Input Voltage, Buck-Boost Converter With 2-A Switch Current,” TPS63060DSC datasheet.
[23] Texas Instruments, “Capacitor-Free, NMOS, 250-mA Low-Dropout Regulator With Reverse Current Protection,” TPS73201DBVR datasheet.
[24] ON Semiconductor, “Dual Schottky Barrier Diode,” D-MMBD352WT1 datasheet.
[25] ON Semiconductor, “Surface Mount Schottky Power Rectifier,” MBRS540T3G datasheet.

[26] C&K, “L Series Miniature Slide Switches,” L101011MS02Q datasheet.
[27] TDK, “Common mode filters For power line ACM series,” ACM7060-301-2PL-TL datasheet.
[29] Texas Instruments, “Low Phase Noise Clock Evaluation Module — up to 1.5 GHz,” CDCE62005EVM User’s Guide.
[30] “Digital potentiometer,” Resistor Guide. [Online]. Available: http://www.resistorguide.com/digital-potentiometer/. [Accessed: 29-Nov-2018].
[31] “Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing,” NI, 2016. [Online]. Available: ftp://ftp.ni.com/evaluation/pxi/Acquiring_Analog_Signal.pdf.
[32] “Data Acquisition Handbook,” Measurement Computing. [Online]. Available: https://www.mccdaq.com/handbook/chapt_2.aspx. [Accessed: 25-Nov-2018].
[33] J.-P. Busson, “Avoid Operational Amplifier Output Stage Saturation by Gain Control,” Texas Instruments, rep., 1999.
[34] J. Alexander, “Digitizing Oscilloscope Basics,” presented at UBC, April 23, 2003. [Online]. Available: https://www.ece.ubc.ca/~robertor/Links_files/Files/oscilloscopes_basics.pdf.
[35] C. Pearson, “High-Speed, Analog-to-Digital Converter Basics,” Texas Instruments, rep., 2011.
[36] AVNET, “Zynq™ Evaluation and Development Hardware User’s Guide,” Zedboard datasheet, August 1, 2012
[37] AVNET, “(Zynq™ Evaluation and Development) Hardware User’s Guide,” Zedboard User’s Guide, January 2014.
[38] L. Crockett, R. Elliot, M. Enderwitz, and R. Stewart, *The Zynq Book*. 1st ed. Glasgow, Scotland: Strathclyde Academic Media, 2014.
[39] Xilinx, “Zynq-7000 SoC Technical Reference Manual.” Zynq-7000 reference manual.
[40] Xilinx, “Clocking Wizard v5.3 LogiCORE IP Product Guide,” Clocking Wizard v5.3 product guide.

Appendix C