ESE498
4-Channel USB Audio Oscilloscope

Samuel Powell and Michael Steinbock

Supervisor
Dr. Robert Morley

Submitted in Partial Fulfillment of the Requirement for the BSEE Degree, Electrical and Systems Engineering Department, School of Engineering and Applied Science, Washington University in St. Louis

May 2010

Abstract

In this paper, we present the design of a 4-channel, audio-bandwidth, USB based oscilloscope for hobby use. The scope consists of a hardware front-end connected to a host computer via USB. It has features common to most oscilloscopes, including the ability to trigger on any channel, AC or DC couple the inputs, adjust the vertical scale such that measurements of signals in the millivolt range and as large as 100 Vpp may be made, optionally filter high-frequency noise from the inputs, perform common math functions on the inputs, and make basic measurements of the inputs. The scope is currently still in the development stages. A preliminary prototype of the hardware front-end has been assembled and mostly tested. Firmware for the front-end has been developed and tested. No software interface for the oscilloscope has been developed yet. Michael Steinbock and Sam Powell have applied ethics to the design process and in the selection of the final proposed design. Further, the Washington University in St. Louis Academic Integrity Policy has been upheld throughout the process.

Acknowledgements

Ed Richter, David Zar, Dr. William D. Richard
## Contents

1 Problem Formulation .................................................. 1
2 Final Specification ..................................................... 1
3 Concept Synthesis ...................................................... 2

4 Analog Design and Analysis ........................................... 7
   4.1 Input Stage: Protection and Coupling .......................... 7
   4.2 Vertical Scaling Stage: Attenuate, Level Shift, and Amplify . 8
   4.3 Anti-aliasing Stage ............................................... 8
   4.4 Noise Analysis ................................................... 9

5 Analog Testing .......................................................... 13
   5.1 Level Shifter and Attenuator ................................... 13
   5.2 Programmable Gain Amplifier .................................. 14
   5.3 Anti-Aliasing Filter .............................................. 15
   5.4 Noise Performance .............................................. 15

6 Digital Design and Testing ............................................ 17
   6.1 Digital Architecture ............................................ 17
   6.2 PGA Handler .................................................... 17
   6.3 ADC Handler .................................................... 21
   6.4 Data Handler .................................................... 21
   6.5 USB Handler .................................................... 21

7 Conclusions ............................................................. 27
   7.1 Cost Analysis .................................................... 27
   7.2 Hazards and Safety .............................................. 27
   7.3 Project Completion Status ....................................... 32
   7.4 Design Process Reflections ..................................... 34
List of Figures

1. Sample Rate and Interfacing Chip Selection ........................................ 3
2. ADC and PGA Chip Selection ................................................................. 4
3. FTDI 2322H USB Mini-Module ............................................................... 5
4. Entire System Block Diagram ................................................................. 6
5. Input Stage Schematic ........................................................................... 7
6. Attenuate and Level Shift Schematic ...................................................... 8
7. Vertical Scaling Stage Schematic ............................................................ 10
8. Anti-Aliasing Filter Schematic ............................................................... 10
9. Anti-Aliasing Filter Frequency Response ............................................... 11
10. Anti-Aliasing Filter Step Response ......................................................... 11
11. PGA Frequency Response .................................................................... 13
12. Level Shifter and Attenuator Test ......................................................... 14
13. Anti-Aliasing Filter Actual and Theoretical Magnitude Response Comparison ................................................................. 15
14. Anti-Aliasing Filter Actual Phase Response .......................................... 16
15. Anti-Aliasing Filter Actual Step Response ............................................ 16
16. FPGA Internal Design Architecture ...................................................... 18
17. PGA Block Diagram ................................................................................ 19
18. PGA Timing Requirements .................................................................... 19
19. PGA_Handler VHDL Simulation ............................................................. 20
20. Verifies $t_1 = 40\text{ns}$, $t_3 = 100\text{ns}$, $t_4 = 60\text{ns}$ (Note: 100ns/div) ................................................................. 20
21. Verifies $t_7 = 40\text{ns}$. Also SHDN is also low to keep the part powered. ................................................................. 22
22. ADC Block Diagram ................................................................................ 22
23. ADC Timing Requirements .................................................................... 23
24. ADC SPI Sample Transfer Waveform .................................................... 23
25. ADC_Handler VHDL Simulation ............................................................. 24
26. Power-up Sequence: After reset CS_L falls, followed by CNVST staying high for roughly 2ms before ready. ................................................................. 24
27. Data timing from ADC meets Spec. Bits are shifted into the FPGA on rising edges of SCLK. ................................................................. 25
28. Samples streaming in on two channels. .................................................... 25
Digital noise coupled into our common ground plane, leaking into the analog signals before and after filtering. ................................. 33

List of Tables

1  Similar Oscilloscope Designs ............................................. 2
2  PGA Input-Referred Thermal Noise. ..................................... 10
3  Nominal vs. Actual PGA Gains .......................................... 14
1 Problem Formulation

Most industry oscilloscopes cost in the thousands of dollars, placing them out of reach from the casual hobbyist. There are a small number of USB based oscilloscopes with sub $300 prices, but are not very well known. The goal of this project was to design an oscilloscope to competitively target this hobbyist market. Initial requirements entering the design process were USB connectivity to the user’s computer and a standard set of oscilloscope features. From there, a specification evolved through comparing feature sets of these cheaper scopes. This process consisted of several compromises between cost, function, and implementation feasibility. For the prototype version, some concessions were made with the hopes of cutting initial development time which would not be ideal for a commercialized product. The final project constraint was to get initial results in the span of one school semester. Further development could incorporate lessons learned in a beta unit for final testing before commercialization.

After initial research of comparable products, we decided on the following minimum requirements.

- The scope must be able to measure signals from DC to at least audio bandwidths (~ 20kHz).
- The scope must have at least 2 channels and an external trigger channel.
- It must be capable of measuring signals at minimum in the millivolt range up to signals at least as high as 60 Vpp.
- It must be able to perform commonly used math functions on input signals including addition, subtraction, multiplication, inversion, integration, differentiation, averaging, and Fast Fourier Transformations.
- It must not require any power beyond what the USB interface provides.

2 Final Specification

We developed the following product specifications from the initial minimum requirements and the process described above.

- The scope will consist of two parts: a hardware front-end, and a host computer software interface. The hardware front-end will be used to sample the inputs and convey those samples via USB to the host computer. The host computer software interface will be responsible for displaying the captured waveforms to the user and for allowing the user to control the settings of the hardware front-end through a graphical user interface (GUI).
- The scope will support 4 channels.
  - The scope will be able to edge-trigger on any of the 4 channels, with adjustable trigger levels.
  - Each channel shall be either AC or DC coupled.
  - Each channel will have an input impedance of at least 1 MΩ.
Each channel will have selectable input ranges of 1, 2, 5, 10, 20, 50, or 100 Vpp.

Each channel will have an optional anti-aliasing filter that will ensure at least 8 bits of noise-free resolution in the audio bandwidths, regardless of gain settings.

Each channel will have less than 1% error with respect to the input range in magnitude in the audio bandwidth.

Each channel will be sampled at 1.25 MHz with 12 bit resolution.

- The hardware front-end of the scope will communicate with the host computer via a high speed (480 Mbps) USB interface.

- The hardware front-end of the scope will be solely powered by the USB interface.

- The software GUI will display the sampled waveforms.

- The GUI will be able to control the vertical range of each channel as well as the triggering settings.

- The GUI will be able to make the following measurements and apply the following mathematical operations on the waveforms:
  - Measurements: minimum, maximum, average, rms
  - Math: invert, add, subtract, multiply, integrate, differentiate, Fast Fourier Transform

3 Concept Synthesis

The specification listed in the previous section was not a trivial item to produce. As will be shown, we went through a series of decisions regarding sample rate and chip selection. The process was iterative in nature, as one decision would affect other decisions elsewhere in the architecture. We were constrained by three priorities throughout all decisions: cost, speed, and resolution. For a comparison, Table 1 lists the three closest competitors as we found. [1] – [3]

<table>
<thead>
<tr>
<th>Scope</th>
<th>Sample Rate</th>
<th>Resolution</th>
<th>Channels</th>
<th>Input Range</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>STINGRAY DUAL CHANNEL</td>
<td>1MS/s Aggregate</td>
<td>12 bits</td>
<td>2</td>
<td>+/-50V</td>
<td>$192.50</td>
</tr>
<tr>
<td>HobbyLab *</td>
<td>200KS/s</td>
<td>10 bits</td>
<td>2</td>
<td>+/-20V</td>
<td>$169.50</td>
</tr>
<tr>
<td>PicoScope 2203 **</td>
<td>1MS/s</td>
<td>8 bits</td>
<td>2</td>
<td>+/-100V</td>
<td>$262.35</td>
</tr>
<tr>
<td>Our Final Specifications</td>
<td>1.25MS/s</td>
<td>12 bits</td>
<td>4</td>
<td>+/-100V</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
- * Also 8MHz logic analyzer, spectrum analyzer, and 1MHz logic generator
- ** Also 5MHz spectrum analyzer and 100KHz waveform generator

Table 1: Similar Oscilloscope Designs
The first and primary decision to make was our intended sample rate. Tied very closely to this decision was the method of interfacing between the analog hardware and the computer. The options that we saw were either to use a microcontroller or an FPGA. The microcontroller would have the ADC’s and USB driver onboard, saving development and production costs. It would only require software programming to setup the interfacing, but there’s a catch. We knew that we would have bandwidth requirements over the Full-Speed USB (4.8Mbps), so we needed a microcontroller with support for the faster High-Speed USB standard (480Mbps). These are just starting to be released, and we could not find one in current production with the SPI port and ADC sampling requirements that we wanted at the time. Had we agreed upon the audio spectrum focus at that point we most likely would have tried to get a sample chip of one of these new microcontrollers. Instead, we decided to move forward with a FPGA to interface with external ADC and USB chips. FPGA’s have a much longer development time compared to a microcontroller, but they are vastly more functional in terms of speed, ports, and implementation possibilities. Cost-wise, the microcontrollers and FPGAs we expected to use are roughly equivalent at roughly $10 per chip. (Xilinx Spartan3E family FPGA, Atmel SAM9U series microcontroller) Figure 1 on page 3 shows an overview of this process.

Figure 1: Sample Rate and Interfacing Chip Selection

Still unanswered was our sample rate goal. We had made our decision on the FPGA without making any final decisions on the original goal. Looking back, this was a very poor route to have taken. Regardless, we moved forward in choosing a Programmable Gain Amplifier(PGA)/Analog to Digital Converter(ADC) combination. Figure 2 describes the iterative process we used to arrive with our final choices. Since the two chips were so closely linked, they had to be well paired, inducing an iterative selection process, as shown.

Along with the chip selection was the decision on number of analog channels to support. We wanted two fully functional channels in addition to an external trigger. As most chips we looked at came in packages of two channels, and an external trigger channel would require the same resolution and sample rate as a regular channel, we decided to make one layout for two channels and duplicate it for an additional two channels, each one selectable for triggering. This option did not require us to design a separate trigger circuit and added two functional input channels. We felt that this was a cost effective solution, as it limits development costs and adds value to the system with the added functionality of two more channels.

The final chip decision was the High-Speed USB interfacing. There are many USB interface chips available...
ADC
- Single Rail Power
- Input range to match PGA output range
- Serial configuration interface

PGA
- Single Rail Power (5V)
- Single-Ended Analog Input
- Serial configuration interface
- Low DC Offset Voltage
- Rail-to-rail output voltages

Design Priorities:
1) Cost
2) Speed (at least audio bandwidth)
3) Resolution (at least 8 bits)

Figure 2: ADC and PGA Chip Selection
from an array of manufacturers. Our decision came down to ease of development. We chose the FTDI FT2322H USB Mini-Module because it comes pre-packaged with several interfacing options ready to go. (See Figure 3) The FTDI part provides easy to use drivers, onboard power management, and a parallel interface to reduce the otherwise necessary high clock rate serial communication line. Finally, we had design experience with this part as well, further reducing development time. However, the module’s price is a major drawback at $27, but we would never use this in a commercial design. This allows us to quickly evaluate our oscilloscope concept, without spending much time on the computer interfacing. In a marketable product, we would use one of several USB chips we found costing only a few dollars. Alternate chips include, but not limited to: SMSC USB3320, TI TUSB6250, or TI TUSB6020. These would require substantially more up-front development time on the hardware and software side, but would save costs in sales.

![Image from: http://ftdichip.com](http://ftdichip.com)

Figure 3: FTDI 2322H USB Mini-Module

As a rule of thumb, we have been taught that high-precision analog to digital converters should have anti-aliasing filters to improve their performance. Without doing any initial cost/benefit analysis, we chose to include a 3rd order anti-aliasing filter for each channel because each filter would use a single opamp. This way we could use relatively inexpensive dual op-amp packages and share them between pairs of channels. After researching the specifications of professional oscilloscopes and talking to more experienced oscilloscope users, we decided that the most important requirements for the anti-aliasing filter would be a fast rise-time with little or no ripple, and a minimum amount of distortion in the scope’s bandwidth.

Our final major design decision was how to split up the oscilloscope math and triggering functions between the FPGA and computer. All of these functions could be implemented on either platform, but we decided to split the load. Triggering was put on the FPGA, operating in full parallel with the data streaming. The LSB of the fourth channel is reserved for trigger events. On the computer side, we were planning on implementing the software in LabView, which would allow for very simple coding of the remaining math functions.

With the above decisions made, Figure 4 on page 6 shows the design we ended up with.
Figure 4: Entire System Block Diagram
4 Analog Design and Analysis

The analog front-end consists of 3 stages before the ADC. As shown in Figure 4 on page 6, the first stage is the Input Stage, and consists of the input connectors as well as protection and AC/DC coupling circuitry. The second stage handles vertical scaling of the signal. It attenuates, level shifts, and then amplifies the signal so that it meets the input range requirements of the ADC. The last stage of the analog front-end is the anti-aliasing filter, used to reduce high frequency noise in the input before it’s sampled.

4.1 Input Stage: Protection and Coupling

The input connectors for each channel are standard female BNC jacks. The signal line of each is connected to ground via a Transient Voltage Suppressor (TVS) diode nominally rated to protect for voltages 48 V above or below ground. (The minimum and maximum breakdown voltages of the TVS diode are 53.3 V and 558.9 V, respectively.) If an overvoltage condition occurs on the input, these diodes will short to ground and dissipate any power that could damage components down the line. Following the TVS diode on each channel is an AC coupling capacitor and a jumper used to switch between AC and DC coupling. The AC coupling capacitor is rated for 200 V. It has a value of 0.1 μF, which combined with the 1MΩ nominal input impedance of the vertical scaling stage forms a high-pass filter with a cutoff frequency of 1.59 Hz. A schematic of this stage is shown below in Figure 5 on page 7.

Figure 5: Input Stage Schematic
4.2 Vertical Scaling Stage: Attenuate, Level Shift, and Amplify

The purpose of the vertical scaling stage is to scale and shift the input signal so that it will be appropriate for input to the ADC. The ADC we chose has a single-ended input and accepts inputs between 0 and 5 V. Thus, this stage attenuates by a factor of 20 (26 dB) to take our maximum input range of 100 Vpp down to 5 Vpp. The input signal is still centered about 0 V, so we have to shift it up by 2.5 V to meet the input requirements of the ADC. The op-amp circuit shown in Figure 6 on page 8 performs these functions.

![Figure 6: Attenuate and Level Shift Schematic](image)

With just the attenuator and level shifter, the least significant bit (LSB) of ADC would represent a voltage change of \( \frac{100 \text{ Vpp}}{4096} \approx 24 \text{ mV} \). This does not meet the design requirement that we must be able to measure signals in the millivolt range. To fix this, we add a programmable gain amplifier (PGA) after the attenuator and level shifter. The PGA’s gain can be set to one of 1, 2, 5, 10, 20, 50, or 100 V/V. Thus, at its maximum gain, the input voltage range is 1 Vpp and a single LSB represents a voltage change of \( \frac{1 \text{ Vpp}}{4096} \approx 240 \mu \text{V} \). The PGA we chose is compatible with the ADC in the sense that its input and output range match the ADC’s input range. It also amplifies about a voltage reference that is average of its supply rails. Fortunately, it outputs this voltage reference so that we can use it in our level shifter to properly center the signal. The reference level comes from a high-impedance resistive voltage divider with no buffering. To avoid loading down the divider with the level shifter circuit, we put a unity gain opamp buffer between the reference output and the level shifter. Each channel must have its own buffer to avoid cross-talk. The final schematic of the vertical scaling stage is shown in Figure 7 on page 10.

4.3 Anti-aliasing Stage

We include an anti-aliasing filter to decrease any noise that might alias into the audio bandwidth when the ADC samples the signal. The criteria we used to choose the transfer function for the filter were that it must have less than 1% attenuation or gain in the audio bandwidth and that it must have a very fast rise time with no ripple. We were able to design a transfer function to meet these requirements using the Filter Solutions software by Nuhertz Technologies, LLC. A schematic of the filter is shown in Figure 8 on page 10. To get the component values, we chose a resistance value of 10 kΩ for all of the resistors and used the Filter Solutions...
software to solve for the capacitor values. We then changed them to the closest standard capacitor values, re-evaluated the transfer function, and confirmed that it still met the specifications (there was negligible change in the transfer function coefficients). Our final filter transfer function (using real component values) is:

\[ H(s) = \frac{1}{6.033 \times 10^{-19} s^3 + 2.111 \times 10^{-12} s^2 + 2.33 \times 10^{-6} s + 1} \]

The frequency response and step response of the filter are shown in Figure 9 on page 11 and Figure 10 on page 11, respectively.

4.4 Noise Analysis

We performed a detailed thermal noise analysis of the 3 input stages to predict how many effective bits of resolution the system has. The first significant source of thermal noise is the attenuator and level shifter. The thermal noise power density due to the attenuator and level shifter, at the output of that circuit, is

\[
N_{\text{atten}} = (i_n R_2)^2 + \left( i_n \frac{R_3 R_4 R_1 + R_2}{R_1 R_3 + R_4} \right)^2 + \left( v_n \frac{R_1 + R_2}{R_1} \right)^2 + 4k_B T \left[ R_1 \left( \frac{R_2}{R_1} \right)^2 + R_2 + R_3 \left( \frac{R_4 R_1 + R_2}{R_1 R_3 + R_4} \right)^2 + R_4 \left( \frac{R_3 R_1 + R_2}{R_1 R_3 + R_4} \right)^2 \right] \text{W/Hz}
\]

where \( R_1, R_2, R_3, \) and \( R_4 \) are the resistances used in the attenuator and level shifter circuit as shown in Figure 7 on page 10. \( i_n \) and \( v_n \) are the standard deviations of the opamp input current and voltage noises, respectively. \( k_B \) is Boltzmann’s constant and \( T \) is the absolute temperature. From our circuit:

\[ R_1 = R_3 = 1 \text{ M} \Omega, \quad R_2 = R_4 = 51.1 \text{ k} \Omega \]

\[ i_n = 2 f_A / \sqrt{\text{Hz}}, \quad v_n = 7.5 \text{ nV}/\sqrt{\text{Hz}} \]

(The opamp parameters were taken from its datasheet.) These values yield an output noise power density of

\[ N_{\text{atten}} = 1.8419 \text{ fW/Hz} \]

for the attenuator and level shifter circuit.

The input-referred thermal noise densities of the PGA is listed in Table 2 on page 10. Since these noise power densities are at the same node in the circuit, we can add them directly. The combined values are also listed in Table 2.

The anti-aliasing filter also contributes noise to the system. It is, in the worst case, insignificant compared to the noise generated by the first two stages. This is because the noise due to the first two stages will be
Figure 7: Vertical Scaling Stage Schematic

Figure 8: Anti-Aliasing Filter Schematic

<table>
<thead>
<tr>
<th>PGA Gain V/V</th>
<th>$v_{n\text{PGA}}$ aV/$\sqrt{\text{Hz}}$</th>
<th>$N_{\text{PGA}}$ fW/Hz</th>
<th>$N_{\text{atten}} + N_{\text{PGA}}$ fW/Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>15.1</td>
<td>0.228</td>
<td>2.070</td>
</tr>
<tr>
<td>50</td>
<td>15.4</td>
<td>0.237</td>
<td>2.079</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>0.256</td>
<td>2.098</td>
</tr>
<tr>
<td>10</td>
<td>16.7</td>
<td>0.279</td>
<td>2.121</td>
</tr>
<tr>
<td>5</td>
<td>19.1</td>
<td>0.365</td>
<td>2.207</td>
</tr>
<tr>
<td>2</td>
<td>24.8</td>
<td>0.615</td>
<td>2.457</td>
</tr>
<tr>
<td>1</td>
<td>35.6</td>
<td>1.267</td>
<td>3.109</td>
</tr>
</tbody>
</table>

Table 2: PGA Input-Referred Thermal Noise.
Figure 9: Anti-Aliasing Filter Frequency Response

Figure 10: Anti-Aliasing Filter Step Response
amplified by the PGA before it is added to anti-aliasing filter’s noise. Because these are noise powers, to refer them to the output of the PGA, we have to multiply them by the PGA’s gain squared. This means that in the worst case, when the PGA is set to a gain of 100, the noise power will be amplified by a factor of 10000 and the noise power density at the input of the ADC will be about 20.7 $pW/Hz$. The anti-aliasing filter uses the same opamp as the attenuator and level shifter circuit and has comparable or smaller component values. Thus the thermal noise it generates at DC should be on the same order as the thermal noise of the attenuator and level shifter—it will be about 10000 times smaller than the total thermal noise power at the input of the ADC. For this reason, the thermal noise due to the anti-aliasing filter is ignored.

In order to calculate the maximum signal to noise ratio, $SNR_{max}$, and effective number of bits of resolution, $N_{eff}$, we need the effective bandwidth of the system. Without the anti-aliasing filter, the noise bandwidth is the same as the sample rate, $f_{NBW} = 1.25$ MHz. To find the effective noise bandwidth with the anti-aliasing filter in place, we find the noise power that the anti-aliasing filter will pass within the original noise bandwidth, then solve for the bandwidth of an ideal brick-wall filter that would pass the same noise power:

$$f_{ENBW} = \int_{0}^{f_{NBW}} \frac{|H(j2\pi f)|^2}{|H(0)|^2} df$$

In this case, $f_{ENBW} = 122.29$ kHz.

The next step in calculating the worst-case $SNR_{max}$ and $N_{eff}$, is to calculate the worst-case (when the PGA gain is 100 V/V) input-referred noise voltages. First we multiply the worst-case noise power density at the output of the attenuator by the bandwidth to get the noise power there. Then we take the square root to get the RMS noise voltage there, and finally multiply by the attenuation factor to move it to the input. The following equations show this calculation with and without the anti-aliasing filter, respectively:

$$v_n = 20\sqrt{(N_{atten} + N_{PGA}) f_{NBW}} = 1.017\ mV_{RMS}$$

$$v_{n,AAF} = 20\sqrt{(N_{atten} + N_{PGA}) f_{ENBW}} = 318.2\ \mu V_{RMS}$$

In the worst-case, the maximum input voltage range is 1 Vpp. This corresponds to an input voltage level of 354 mV$_{RMS}$. Thus, the worst-case maximum SNR with and without the anti-aliasing filter is

$$SNR_{max} = \frac{354\ mV}{1.017\ mV} = 347.64\ V/V = 50.823\ dB$$

$$SNR_{max,AAF} = \frac{354\ mV}{318.2\ \mu V} = 1.1111\ KV/V = 60.915\ dB$$

We can calculate the effective number of bits of resolution in the output using the following equations:

$$N_{eff} = \frac{SNR_{max}[dB] - 1.76}{6.02} = 8.15\ bits$$

$$N_{eff,AAF} = \frac{SNR_{max,AAF}[dB] - 1.76}{6.02} = 9.826\ bits$$
Note that this analysis covers the absolute worst-case thermal noise. For lower gains, the effective number of bits will increase, but the noise due to the anti-aliasing filter will become significant. Thus the noise powers for the other gains shown above are not sufficient for calculating the maximum $SNR$ or $N_{eff}$. Also, the frequency response of the PGA (shown in Figure 11 on page 13) will further attenuate high frequency noise and improve the noise performance.

![LTC6912-1 Frequency Response](image)

**Figure 11: PGA Frequency Response**

5 Analog Testing

5.1 Level Shifter and Attenuator

To test the level shifter and attenuator circuit, we applied a 100Hz 20Vpp sine wave to the signal input, set the analog ground voltage level to 2.5 V and examined the output. As shown in Figure 12 on page 14, the
output was attenuated by a factor of 20 to 1 Vpp and shifted up to be centered on the reference voltage.

![Waveforms](image)

**Figure 12: Level Shifter and Attenuator Test**

### 5.2 Programmable Gain Amplifier

To test the proper operation of the PGA, we applied a 100Hz sine wave of varying amplitudes to the signal input. The amplitude was set such that the output of the PGA would not saturate but would be as close to the maximum as our signal generator would allow. We then iterated through all of the nominal PGA gains and measured the peak-to-peak input and output magnitudes so that we could calculate the actual gain. These measurements are displayed in Table 3 on page 14. All of these gain measurements fall within the gain errors specified in the PGA datasheet.

<table>
<thead>
<tr>
<th>$A_{nom}$ V/V</th>
<th>$V_{in}$ mV</th>
<th>$V_{out}$ V</th>
<th>$A_{act}$ V/V</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>520</td>
<td>0.0005</td>
<td>0.001</td>
</tr>
<tr>
<td>1</td>
<td>520</td>
<td>0.519</td>
<td>0.99</td>
</tr>
<tr>
<td>2</td>
<td>520</td>
<td>1.04</td>
<td>2.00</td>
</tr>
<tr>
<td>5</td>
<td>520</td>
<td>2.58</td>
<td>4.96</td>
</tr>
<tr>
<td>10</td>
<td>262</td>
<td>2.58</td>
<td>9.85</td>
</tr>
<tr>
<td>20</td>
<td>107</td>
<td>2.07</td>
<td>19.34</td>
</tr>
<tr>
<td>50</td>
<td>94</td>
<td>4.47</td>
<td>47.55</td>
</tr>
<tr>
<td>100</td>
<td>44</td>
<td>4.08</td>
<td>92.73</td>
</tr>
</tbody>
</table>

**Table 3: Nominal vs. Actual PGA Gains**
5.3 Anti-Aliasing Filter

The anti-aliasing filter was tested independently of the other input stages. A 4.8 Vpp sine wave was applied to the input and swept from 10Hz to 1MHz and the both the output magnitude and the phase difference between the input and output was measured using an oscilloscope. A comparison of the magnitude response of the actual filter and the ideal transfer function is presented in Figure 14 on page 16. The phase response of the actual filter is shown in Figure 14 on page 16, and the step response is shown in Figure 15 on page 16.

![Bode Diagram](image)

Figure 13: Anti-Aliasing Filter Actual and Theoretical Magnitude Response Comparison

5.4 Noise Performance

To test the noise performance of the front-end, we shorted the input to ground and measured the RMS voltage at after each stage of the input. Perhaps due to our measurement set-up loading the circuit or something we overlooked in the way we processed the data, the measurements we made were inconsistent with linear circuit behavior. (For example, the noise floor we measured at the output of the anti-aliasing filter was lower than at the input, even though signal levels remain the same and our tests of the filter show that it has a DC gain of 0dB.) We have chosen not to include the noise measurements here because we are unsure which
Figure 14: Anti-Aliasing Filter Actual Phase Response

Figure 15: Anti-Aliasing Filter Actual Step Response
data are valid and where the non-linearities are coming from.

6 Digital Design and Testing

6.1 Digital Architecture

The digital design has the following functions:

- Sending gain settings to the PGA’s
- Acquiring data from the ADC’s
- Triggering
- Interleaving the 4 channels
- Streaming data to the computer
- Receiving commands from the computer

Our implementation of this can be seen in 16. We used a Xilinx Spartan3E 1600 FPGA on Digilent’s development board for this FPGA. We used the 50 input FX-2 connector on the development board to interface with our expansion board. This includes both power and data connections. The functions listed above were divided into blocks to handle similar tasks and to allow for code interchangeability if parts are swapped in the future. With the USB mini-module, the entire system can be powered over the USB bus 5V line. To ensure that we conformed to the USB power spec, we added a TI current-limited power switch (TPS2045A: 500mA) that was set to enable only after the FTDI part successfully enumerated. In the next iteration of this project, the device could be entirely turned off merely by placing the USB chip in a low power state, opening this power switch. The following sections explain more in-depth each block in the digital design and how they interact.

6.2 PGA Handler

The PGA_Handler is responsible for communicating with the PGA’s (Figure 17 on page 19) using an SPI interface defined in the PGA datasheet. [4] Each handler block is connected to a single PGA chip, with shared chip select, data clock, and shutdown lines going to the IC’s. Each block has an 8bit shift register containing the 4bit gain settings for both channels. This is set when the load input signal is driven high. This also starts the internal state machine which sends out the command sequence to the PGA chips. Because the PGA’s share all inputs except the data lines themselves, each PGA_Handler block is activated simultaneously when setting gains. Figure 18 on page 19 shows the SPI specification for communicating properly to the PGA’s from the datasheet. Figure 19 on page 20 shows VHDL simulation verifying output, and Figures 20 and 21 show the signals in hardware for gains of 50 and 1 respectively (Bit Sequences 0110 and 0001). We verified the timing requirements listed in the datasheet as well as correct operation on all channels of each part. Once in hardware, we verified each gain setting to be accurate to less than 0.5dB for a 1KHz input (most only off by 0.2dB).
Figure 16: FPGA Internal Design Architecture

Notes:
- Data FIFO feeds the computer_interface block. Originally planned to be USB connectivity, but due to issues, now a serial I/O interface (essentially two shift registers for data in and data out)
- State Machines drive unconnected signals
Figure 17: PGA Block Diagram

**LTC6912**

**SERIAL INTERFACE SPECIFICATIONS**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>CIN Valid to CLK Setup</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t2</td>
<td>DIN Valid to CLK Hold</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t3</td>
<td>CLK High</td>
<td></td>
<td>50</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t4</td>
<td>CLK Low</td>
<td></td>
<td>50</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t5</td>
<td>CS/LD Pulse Width</td>
<td></td>
<td>40</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t6</td>
<td>LSR_CLK to CS/LD</td>
<td></td>
<td>40</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t7</td>
<td>CS/LD Low to CLK</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t8</td>
<td>DOUT Output Delay</td>
<td></td>
<td>85</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t9</td>
<td>CLK Low to CS/LD Low</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Serial Interface Timing, Dual ±4.5V ±±5.5V Supplies (Note 10)

Figure 18: PGA Timing Requirements
Figure 19: PGA_Handler VHDL Simulation

Figure 20: Verifies $t_1 = 40\text{ns}$, $t_3 = 100\text{ns}$, $t_4 = 60\text{ns}$ (Note: 100ns/div)
6.3 ADC Handler

The ADC Handlers communicate directly with the pair of MAX1379 ADC’s, with one ADC_Handler per ADC. The MAX1379 is a dual channel, 12 bit, 1.25MS/s ADC with many different configuration options. We were forced to use single-ended input to match the output capabilities of the PGA. The chip allows for one serial data out line per channel, enabling easy separation of the incoming data to the FPGA. The digital drivers are powered from a separate 3.3V supply to conform to the FPGA input limits. See Figure 22 on page 22 for the manufacturer’s block diagram of the ADC. It uses an SPI interface to transfer samples back to the FPGA. Figures 23 and 24 show the timing requirements to be implemented in the ADC_Handler. Using one of the FPGA’s Digital Clock Managers (DCM), I step down the system 50MHz clock to 33.33MHz to drive the ADC_handler’s state machine. A locked signal out of the DCM is sent to the ADC_Handler, so that it does not start until the clock is stable. The state machine is very simple, in that it holds the idle state while in reset or waiting for the DCM. After that, it has a 2ms power-up sequence followed by looping through sample acquisition cycles. As setup, this block continually streams data until power-down or reset. To enable this, the CNVST signal is pulsed high through the 15th and 16th bits of each sample. 25 shows simulation of the VHDL. Figures 26 - 28 show the hardware control signals going to the ADC to verify timing constraints.

6.4 Data Handler

The Data_Handler can be seen in Figure 16 on page 18 as the hardware following the ADC Handlers. It is responsible for reading samples from the ADC_Handler FIFO’s and interleaving them out to 1 data channel. When a sample is available, each channel is selected through the data multiplexer from channel 1 through 4, and the multiplexed data is written out to the final data FIFO. This block will also pause writes if the data FIFO ever fills up so that no data is lost. Concurrently, triggering logic is initialized from computer commands with trigger channel, trigger level, and rising or falling edge. Each incoming sample on the selected channel is compared against the defined trigger level, and that output is fed into two successive flops enabled at each sample. By looking for changes between these flops, rising or falling edge triggers can be determined. The trigger event output replaces the LSB of channel 4’s data line, for the computer to extract and process. This actually does not cut out any usable data from the ADC data, as we pad each 12bit ADC sample with 4 bits, as described in the previous subsection. Figure 29 on page 26 shows simulation output of this block, verifying operation.

6.5 USB Handler

The USB_Handler forms the crucial connection between our peripheral and the host computer. Unfortunately we ran into issues with the FTDI Mini-module enumerating at all. We continued with other parts of the design, not wasting too much time on an unclear issue that is not the focus of this project. In the final day, we found that our development board shares the I/O ports to the expansion board differently than with an earlier version of the same development board. Specifically, the board’s configurable LED’s are shared on several of the mini-module’s configuration and data lines. This would add an excessive load, to an already close timing link.
Figure 21: Verifies $t_7 = 40$ns. Also SHDN is also low to keep the part powered.

Figure 22: ADC Block Diagram
### Dual, 12-Bit, 1.25Msps Simultaneous-Sampling ADCs with Serial Interface

#### TIMING CHARACTERISTICS (Figures 6, 10)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCLK Clock Period</td>
<td>tck</td>
<td></td>
<td>50</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Duty Cycle</td>
<td>tR/L</td>
<td></td>
<td>45</td>
<td>55</td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>SCLK Pulse-Width High</td>
<td>tCH</td>
<td></td>
<td>22.5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Pulse-Width Low</td>
<td>tCL</td>
<td></td>
<td>22.5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Rise to DOUT Transition</td>
<td>tDOUT</td>
<td>$C_L = 30pF, V_{IH} = 5V$</td>
<td>14</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_L = 30pF, V_{IH} = 3V$</td>
<td>17</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_L = 30pF, V_{IH} = 1.8V$</td>
<td>24</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>DOUT Remains Valid After SCLK</td>
<td>tD/Hold</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CNVST Fail to SCLK Fail</td>
<td>tSETUP</td>
<td>$C_L = 30pF$</td>
<td>10</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CNVST Pulse Width</td>
<td>tsetup</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Power-Up Time, Full Power-Down</td>
<td>tsleep</td>
<td>External load on REF x 3μF</td>
<td>2</td>
<td></td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>SEL to CNVST Fail</td>
<td>tsel_setup</td>
<td></td>
<td>100</td>
<td>120</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>SEL Hold to CNVST Fail</td>
<td>tsel_hold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CS Fail To CNVST Fail</td>
<td>tCSF</td>
<td>External load on REF x 3μF</td>
<td>2</td>
<td></td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>Restart Time, Partial Power-Down</td>
<td>tsycyc</td>
<td>No external load</td>
<td>16</td>
<td></td>
<td></td>
<td>CYCLES</td>
</tr>
</tbody>
</table>

**Note 1:** Relative accuracy is the deviation of the analog value at any code from its theoretical value after the gain error and the offset error have been nulled.

**Note 2:** Offset nulled.

**Note 3:** Conversion time is defined as the number of clock cycles (16) multiplied by the clock period. Clock has 50% duty cycle.

**Note 4:** At sample rates below 10ksps, the input full linear bandwidth is reduced to 5kHz.

**Note 5:** SCLK and CNVST not switching during measurement.

---

**Figure 23:** ADC Timing Requirements

**Figure 6:** Detailed Serial-Interface Timing Diagram

**Figure 24:** ADC SPI Sample Transfer Waveform
Figure 25: ADC_Handler VHDL Simulation

Figure 26: Power-up Sequence: After reset CS_L falls, followed by CNVST staying high for roughly 2ms before ready.
Figure 27: Data timing from ADC meets Spec. Bits are shifted into the FPGA on rising edges of SCLK.

Figure 28: Samples streaming in on two channels.
Figure 29: Data_Handler VHDL Simulation
For testing purposes, we were able to use one of the older development boards. The operational clock frequency of the FTDI mini module is static at 60 MHz. Due to the lengthy setup time of the FTDI part, 11 ns, with regards to its 16.666 ns period, other avenues had to be pursued in order for the FPGA to meet the required setup time (See Figure 30). Thus, a Digital Clock Manager onboard the FPGA was used to delay the output clock from the FTDI module to match the phase of the 60 MHz clock at all nodes within the FPGA. In addition, several parameters were set in the UCF file of the FPGA, such as setting the drive strength on the pins to 12 mA to provide faster point to point communication, increasing the slew rate to fast mode to reduce signal rise and fall times, and eliminating the delay on the input output buffers of the FPGA to decrease the latency in transmission. Finally, all of the signals coming from the FPGA to the FTDI part had to be flopped so as to be stable 11 ns before the FTDI clock. With all of these modifications, the FPGA was able to safely meet the 11 ns setup time required to clock the circuit correctly.

To interface with the chip, we implemented the design and state machine in Figures 31 and 32. This design puts a priority on reading commands from the computer before streaming data out. Reading data from the part required a straight-forward signal sequence shown in Figure 21. On the other hand, writing requires consideration of several cases where the writes would be interrupted: incoming reads, FTDI transmission buffer filling up, or the data FIFO emptying. All these cases were tested for in Simulation (See Figures 33 -35). Using a basic streaming FPGA and software design from another project we were able to sustain a 217Mbps, one-way stream to the computer without dropping data.

7 Conclusions

7.1 Cost Analysis

The estimated cost of the parts necessary for a production version of the scope when buying in quantities of at least 1000 is $55 to $65. A 9 square inch, 4-layer printed circuit board could cost as little as $4.73 per board as quoted by Sunstone Circuits. We were unable to find a quote from an assembly service for our board specifications, but extrapolated from the services that Screaming Circuits provides it would cost around $20 per board for automated assembly. This brings the total to somewhere around $85. This does not include the cost of shipping, enclosure, software distribution, or any other costs. To account for those, we estimate that the final production cost per board to be around $100.

7.2 Hazards and Safety

In the prototype state, there is a potential shock hazard if the scope is handled while high-voltage inputs are connected. This is because due to exposed circuit components and jumpers. The scope would be enclosed in a final production version, which would remove this hazard. In addition, there is the possibility of damaging the scope if the maximum ratings of the TVS diodes are exceeded. The TVS diodes are rated for a 600W peak power pulse for 1 ms at 25°C and a maximum of 100A for 8.3 ms at 25°C. The peak power pulse must be derated by 0.8%/°C.
4.4 FT245 Synchronous FIFO Interface Mode Description

When channel A is configured in an FT245 Synchronous FIFO interface mode the IO timing of the signals used are shown in Figure 4.4, which shows details for read and write accesses. The timings are shown in Table 4.1. Note that only a read or a write cycle can be performed at any one time. Data is read or written on the rising edge of the CLKOUT clock.

![Figure 4.4 FT245 Synchronous FIFO Interface Signal Waveforms](image)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MIN</th>
<th>NOM</th>
<th>MAX</th>
<th>Units</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>16.67</td>
<td>8ns</td>
<td>16.67</td>
<td>ns</td>
<td>CLKOUT period</td>
</tr>
<tr>
<td>t2</td>
<td>7.5</td>
<td>8ns</td>
<td>8.33</td>
<td>ns</td>
<td>CLKOUT high period</td>
</tr>
<tr>
<td>t3</td>
<td>7.5</td>
<td>8ns</td>
<td>8.33</td>
<td>ns</td>
<td>CLKOUT low period</td>
</tr>
<tr>
<td>t4</td>
<td>1</td>
<td>7.15</td>
<td>ns</td>
<td>CLKOUT to RXF#</td>
<td></td>
</tr>
<tr>
<td>t5</td>
<td>1</td>
<td>7.15</td>
<td>ns</td>
<td>CLKOUT to read DATA valid</td>
<td></td>
</tr>
<tr>
<td>t6</td>
<td>1</td>
<td>7.15</td>
<td>ns</td>
<td>OE# to read DATA valid</td>
<td></td>
</tr>
<tr>
<td>t7</td>
<td>1</td>
<td>7.15</td>
<td>ns</td>
<td>CLKOUT to OE#</td>
<td></td>
</tr>
<tr>
<td>t8</td>
<td>11</td>
<td>ns</td>
<td>11</td>
<td>RD# hold time</td>
<td></td>
</tr>
<tr>
<td>t9</td>
<td>10</td>
<td>ns</td>
<td>7.15</td>
<td>RD# hold time</td>
<td></td>
</tr>
<tr>
<td>t10</td>
<td>11</td>
<td>ns</td>
<td>7.15</td>
<td>Write DATA setup time</td>
<td></td>
</tr>
<tr>
<td>t11</td>
<td>0</td>
<td>ns</td>
<td>7.15</td>
<td>Write DATA hold time</td>
<td></td>
</tr>
<tr>
<td>t12</td>
<td>11</td>
<td>ns</td>
<td>7.15</td>
<td>WR# setup time (WR# low after TXE# low)</td>
<td></td>
</tr>
<tr>
<td>t13</td>
<td>0</td>
<td>ns</td>
<td>7.15</td>
<td>WR# hold time</td>
<td></td>
</tr>
<tr>
<td>t14</td>
<td>0</td>
<td>ns</td>
<td>7.15</td>
<td>WR# hold time</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 FT245 Synchronous FIFO Interface Signal Timings

Figure 30: USB Data Transfer Timing Constraints
Figure 31: USB Handler Block Diagram

29
Figure 32: USB Handler State Machine
Figure 33: USB_Handler Simulation Part 1

Figure 34: USB_Handler Simulation Part 2
7.3 Project Completion Status

Unfortunately at this point we are not as far as we had hoped entering this project. We have a prototype very close to completion, given 2-3 additional weeks, but at the same time we have the basic components in an initial form to show and learn from. Even with our current state, we believe this design has the potential to be competitive at market after further revisions are made to resolve issues we have uncovered in testing. Primary issues or things to be done are:

- Resolve analog and digital coupling
- Create an intuitive software interface
- Fully integrate USB_Handler and analyze performance
- Verify system functionality from input to data output
- Calibrate system

When we started initial testing of the ADC’s, we were very surprised to find significant amounts of high frequency noise even on the analog inputs to the ADC’s. At first we thought it was a measurement error or coupling, but looking at the FFT of the noise we noticed most of the energy was at 1MHz. At that point, the FPGA system was running off a 50MHz clock, and we divide that down to 33MHz from which we form the serial signals to the ADC’s. With a 33MHz input to the ADC_Handler, it’s sample rate drops to 1MS/s. Looking at the overlaid signals, seen in Figure 36 on page 33, we quickly realized the analog noise was coupling from the digital signals. The noise, at over 200mV peak-to-peak, is far from negligible. Expanding our FFT further, we saw the SCLK coupling in at roughly 16.67MHz. Excluding the effects of harmonics, we calculated the 16.67MHz signal to alias into our system at 370KHz, which could be theoretically filtered out digitally. However, the 1MHz coupling would alias into the DC and low frequency’s. Best case with an input signal at 100V peak-to-peak, we would have a 5V input to the ADC with the noise estimated at roughly a triangle wave (Estimation from Figure 36) with a 100mV amplitude. Using the same noise analysis presented above, with a 5V input range, we would have roughly 5.9 effective bits of resolution at best. Although, potentially not a fully fatal issue, it is unacceptable for an oscilloscope and needs to be fixed. We think the coupling is due to our unified analog and digital ground plane. With the ADC’s and PGA’s needing both analog and digital grounds, we thought (incorrectly) that it would not make a big difference to save time with a unified ground plane. Well... now we know what happens. Certainly one of the first changes made in a subsequent revision should be to separate these planes properly using a star ground separation layout.

Of course, we are completely missing a software interface to display the streamed data and relay commands to the hardware. As an Electrical Engineering project, this was secondary to the other project needs given our time restriction. Our intentions were to figure out how to get the real-time data stream into LabView which open the data to a rich signal processing toolbox. In a simple approach, the GUI could be quickly assembled, along with the planned math manipulations to be performed in software. Interfaces for setting the trigger level, trigger channel, edge selection, and vertical scale would be needed. Our approach to sending
Figure 35: USB_Handler Simulation Part 3

Figure 36: Digital noise coupled into our common ground plane, leaking into the analog signals before and after filtering.
configuration information to the hardware was to write a packet containing all settings at each change. Although this requires more bandwidth, the percentage of data streaming from the computer will still be a small percentage of the data streaming in from hardware. Further, this implementation greatly simplifies implementation in hardware and software.

The USB interfacing ran into major problems early on when the FTDI Mini-Module failed to enumerate at all with the computer. This issue was set aside and only reconsidered recently with the discovery of the LED’s loading several critical communication lines. With little time, we were unable to remove the LED’s, but we did verify the USB_Handler implementation using another project’s PCB utilizing the same FTDI part. In that test, we were able to successfully loopback data and stream unidirectional data at 217Mbps without dropping data. If this project is moved forward, we recommend staying with this part until there is clear evidence of enough demand to justify incorporating a cheaper USB interface chip. Looking back, more time should have been spent comparing alternatives from the start, but this chip does enable many features important to the system design. Other features we incorporated from the FTDI Mini-Module were the 60MHz clock, 3.3V regulated supply, and easy to setup driver interfacing.

Once we form a complete data path from analog input to digital output at the computer, the final steps would be to verify the digital signals match the expected values. Calibration will be needed to determine where 0V is digitally, since the ADC inputs are centered around 2.5V. This will also include verifying that none of the parts are limiting the signal more than expected. The system frequency and phase response will need to be compared against the expected response. Power consumption with the entire system running should be determined, although we were within the 500mA limit even powering the development board as well. The final change required for a final product is to integrate a minimal FPGA onto a single board. According to the resource usage on our current design, we can use the cheapest Xilinx Spartan3A FPGA at less than $5 each (much cheaper than the microcontrollers initially considered). All in all, this initial prototype has provided an important start on a product that has potential to compete commercially. We are aware of the current limitations and causes that need to be resolved.

7.4 Design Process Reflections

Looking back on the design process we used over the course of the project, we have several areas to learn from. First, and most importantly, is the importance of narrowing on an agreed specification early in the project. We had issues in agreeing upon the bandwidth of our scope, which greatly affected component selection for the rest of the system. On one side, all of the identified competing scopes sample at over 1MS/s, but we were not able to identify any signals of potential interest between 20KHz and radio frequencies. If the focus is only the audio spectrum, the ADC does not need to have a maximum sample rate of over 50KS/s, and money could potentially be saved with cheaper parts and lower design effort. For us, the specification was not fully agreed upon until roughly half-way through the project. This caused problems because we had already completed significant design work with parts that exceeded the needs of the specification. A second aspect of the design process that we learned is the importance of supporting all decisions with factual data. This was an issue related to our indecisiveness in agreeing upon a final bandwidth for the system. We wanted to accept the added cost of a higher bandwidth, but we did not have any justification for doing so. Also in development, it was clear that thoughts or opinions sometimes obstructed progress. Instead, hard
facts or well justified figures would have been helpful in forming more robust justification about important design decisions.

This project was an initial taste of what design work is like, and as such, it was an opportunity to learn from mistakes and issues along the way. We have learned a great deal from this semester’s experience and will surely take these lessons forward in future design work.