Ultrasound Brain Imaging

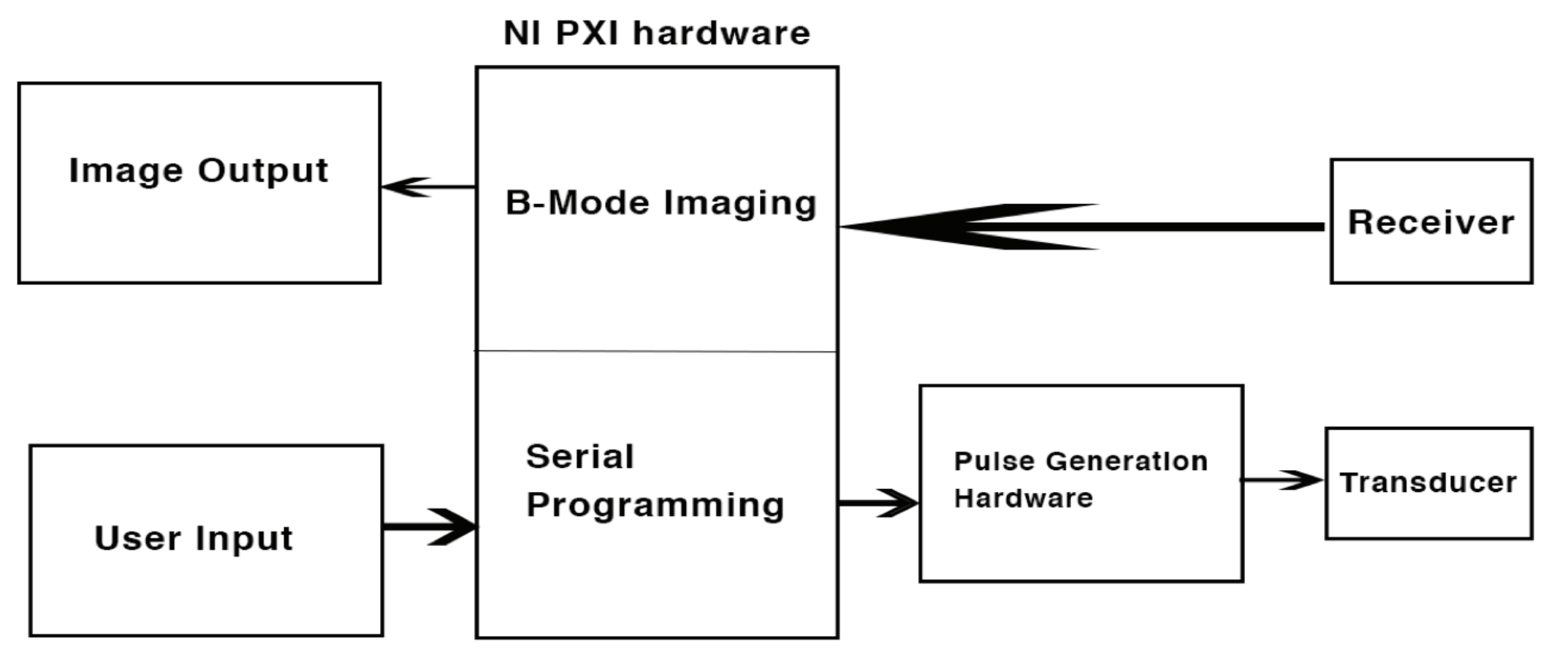
May 15-22
Jingyu Xie, Haoyu Wang, Aaron Tainter
Advisor/Client: Professor Bigelow
Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA

Introduction

This project was proposed by Professor Bigelow as a lab experiment. The goal is to create a system that can be used for ultrasound imaging of the brain. Creating the hardware needed for 256 channels is out of the budget of this project. Instead, our focus was on the design a system that was could be scaled.
In order to achieve our goal, we divided this project into two parts: software and hardware:
The hardware we created is used to produce a signal for the ultrasound transducer, reduce noise, amplify signals, and protect the transducer circuit. For the software part, we developed a system that could output serial instructions to program our custom hardware. Imaging algorithms were developed to translate received RF data into a B-Mode image.

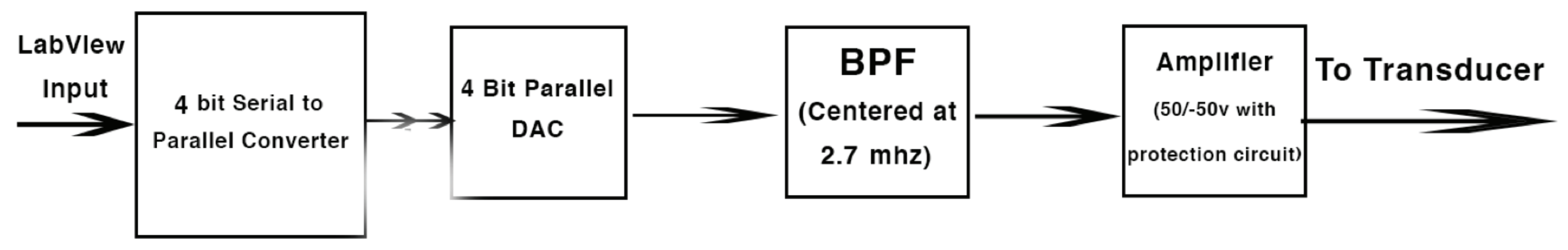
Design Approach

Block Diagram


NI PXI hardware
Image Output
B-Mode Imaging
Receiver
User Input
Serial Programming
Pulse Generation Hardware
Transducer

Block Diagram 1: This is block diagram of our overall design. First we input signals from users to serial programming system which will generate a series of output voltage levels (described in binary) that will be used by the DAC in the pulse generation hardware to create a waveform. The purpose of the pulse generation hardware is to converte a signal from digital to analog, reduce the noise in the generated waveform and amplify the signal across 256 channels. The receiver collects data sends the signals to B-Mode imaging system. The B-Mode imaging system generates images from input signals collected by the receiver. The image will show on the screen after running this system. This system is programmed by labview on NI PXI hardware. Ni PXI hardware was purchased by Professor Bigelow.

Hardware


LabView Input
4 bit Serial to Parallel Converter
4 Bit Parallel DAC
BPF (Centered at 2.7 mhz)
Amplifier (50/-50v with protection circuit)
To Transducer

Block Diagram 2 shows a detailed description of Pluse generation hardware in Block diagram 1. First we need to convert Labview serial input signal to parallel signal, because the input from Labview can only generate a 4 bit dataline, our DAC can only process 8 bits. This is why we have to convert serial signals to parallel signals. Then we convert the parallel digital signals to analog signals by using a DAC chip. Since there is typically some noise in the signal, the analog signal is processed by a bandpass filter with range of 500k hz to 5mhz. We set up a amplifier with a protection circuit to amplify the signal and make sure nothing would damage the hardware before entering the Transducer.

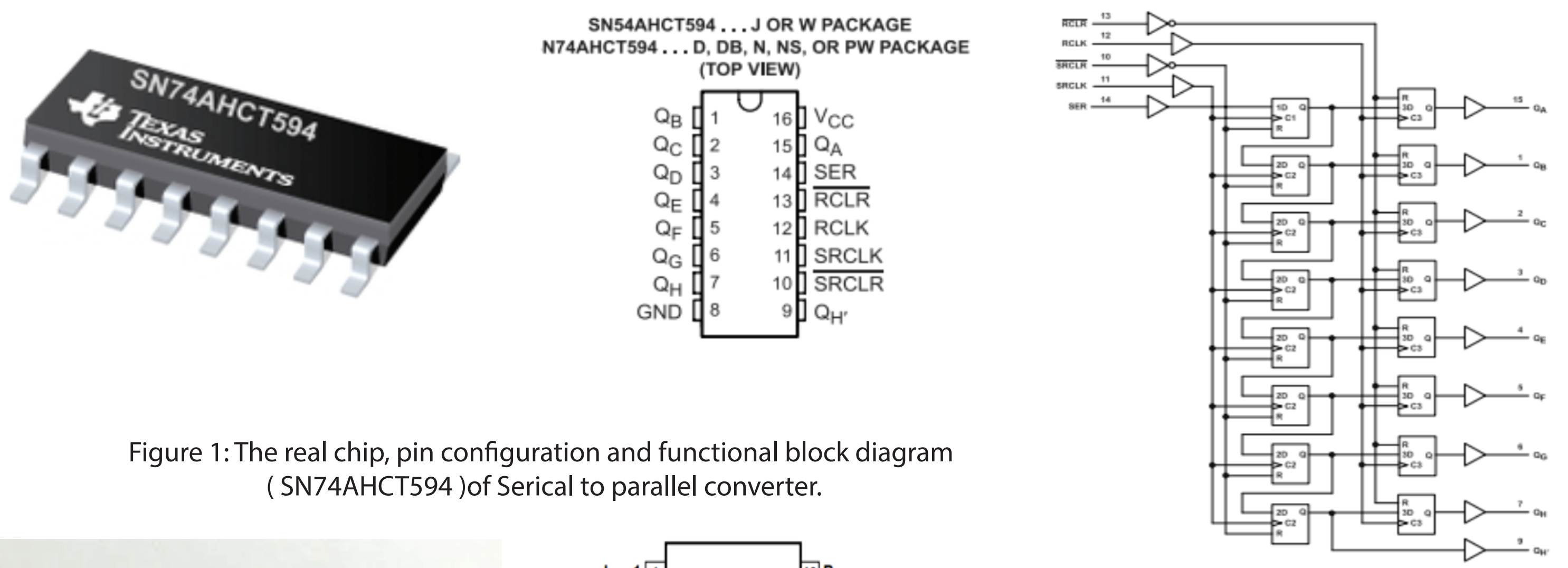


Figure 1: The real chip, pin configuration and functional block diagram (SN74AHCT594)of Serical to parallel converter.

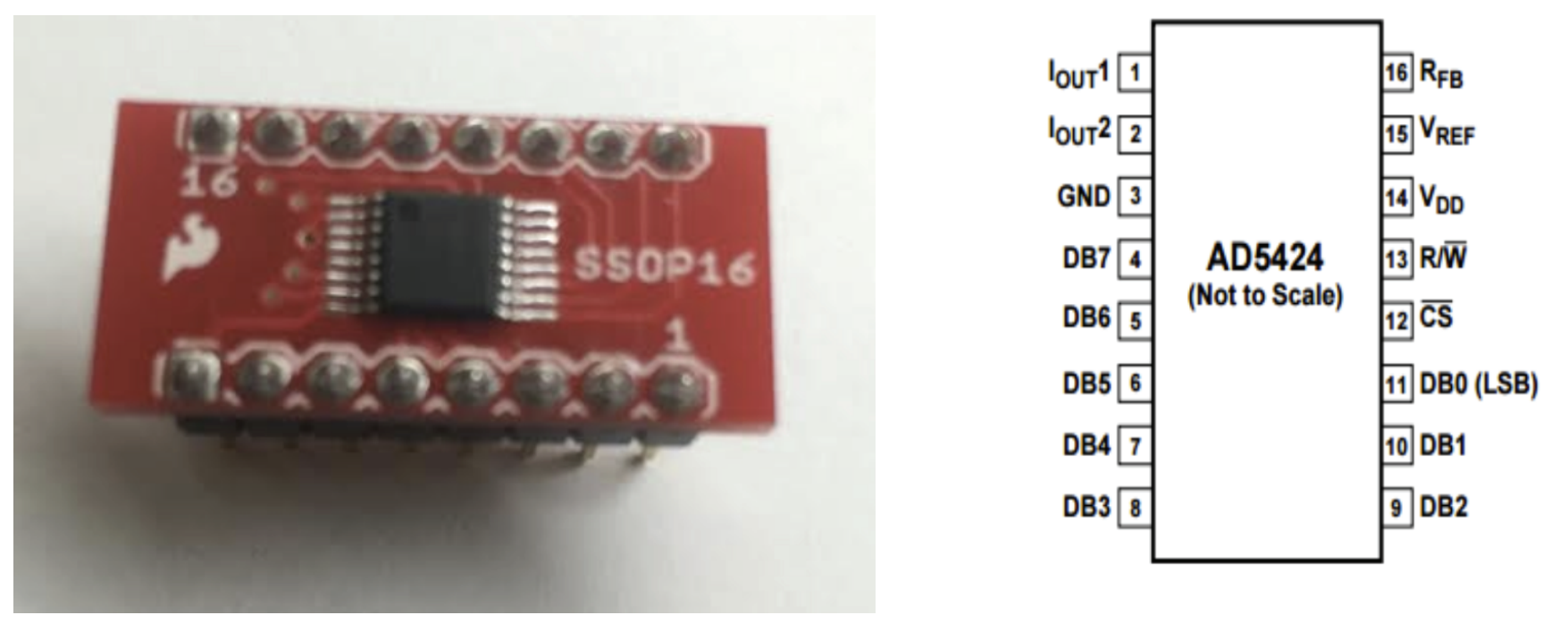


Figure 2: The real chip and the pin configuration of 4 bit Parallel DAC

We had run some simulations to make sure everything is working on properly. We did severial tests on Matlab-simulink as following

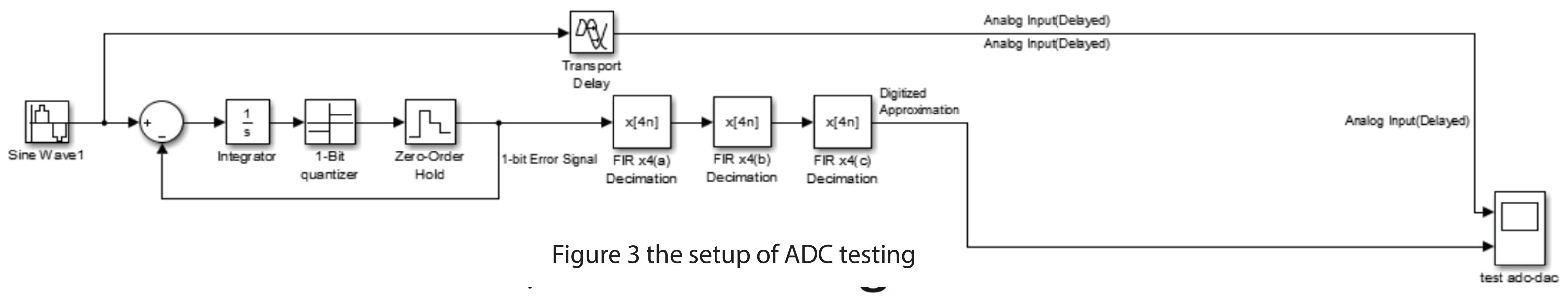


Figure 3 the setup of ADC testing

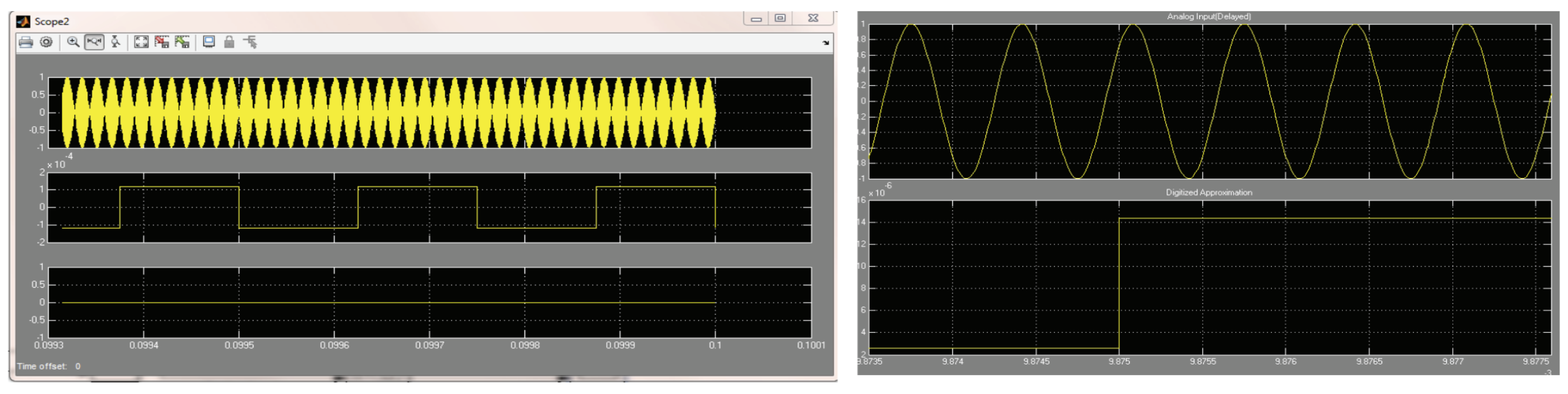


Figure 4 the result of ADC testing in Matlet-Simulink

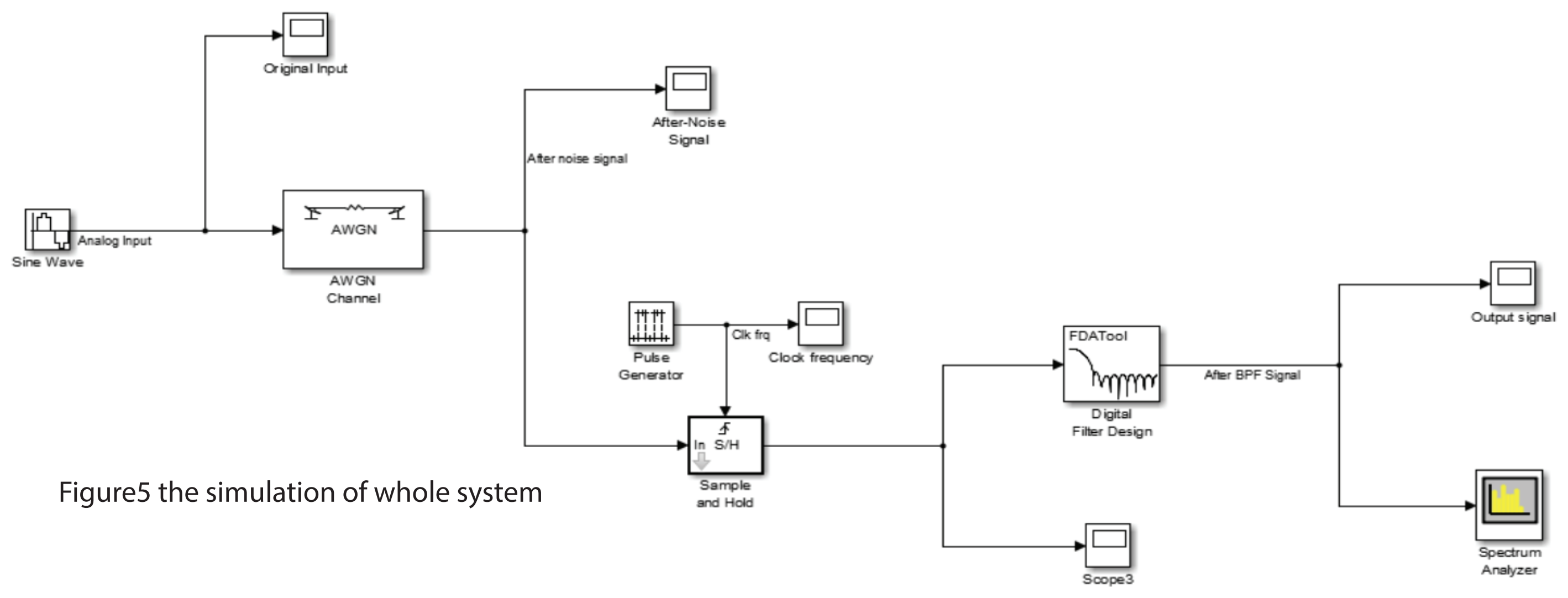


Figure5 the simulation of whole system

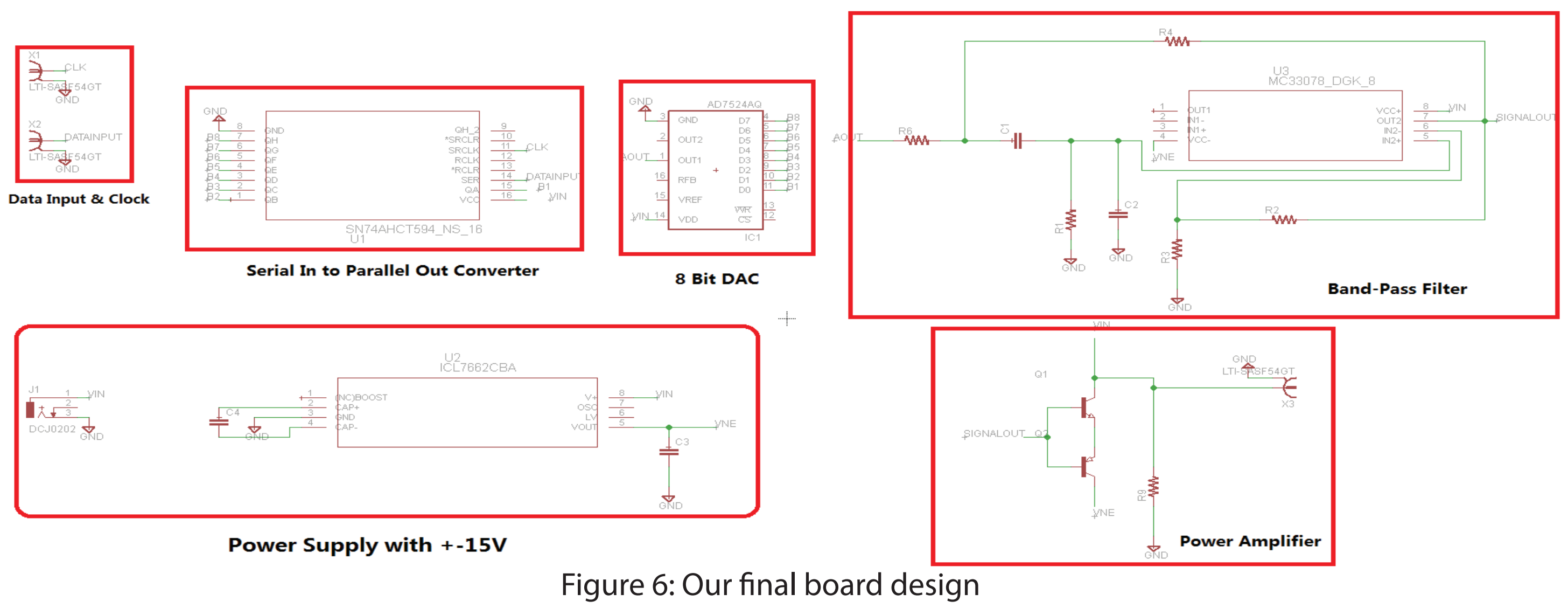


Figure 6: Our final board design

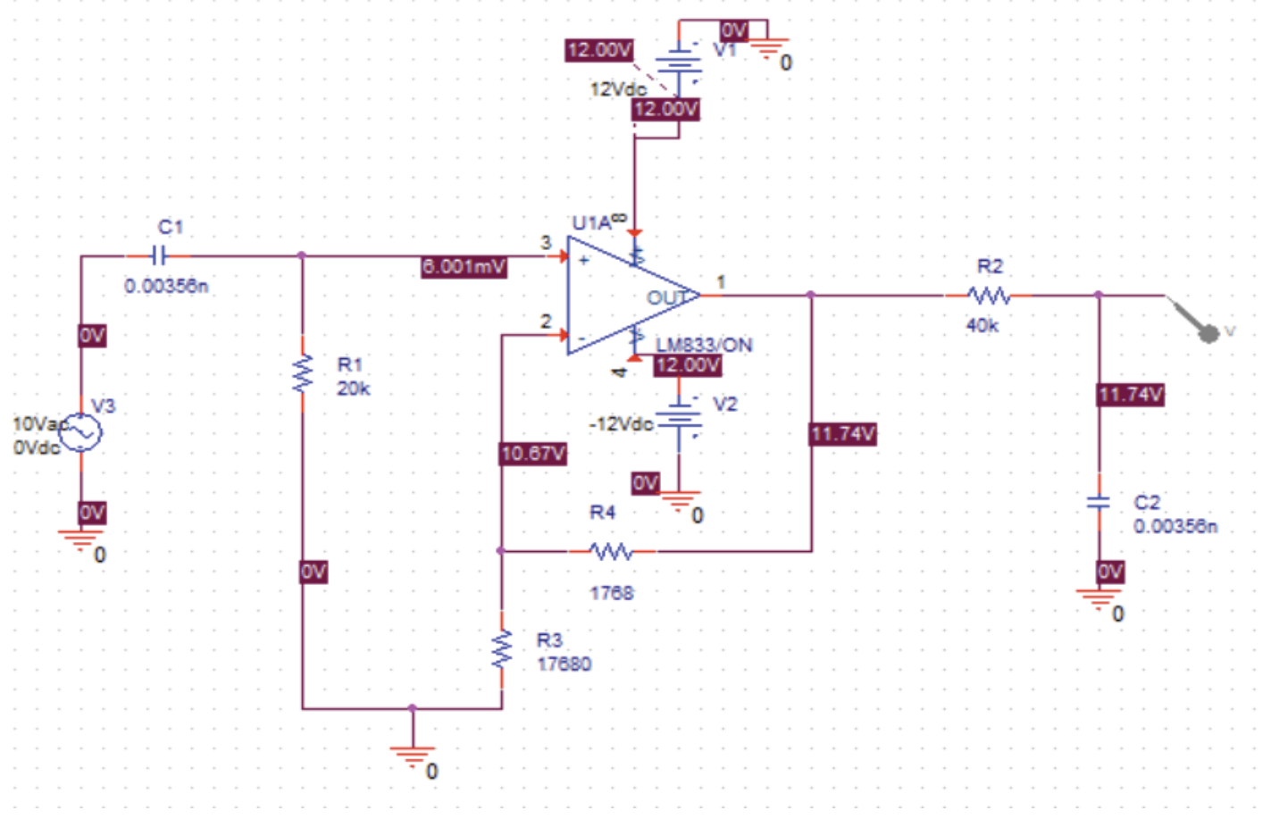


Figure 7: The sketch diagram of the banpass filter with the range of 500k hz to 5mhz

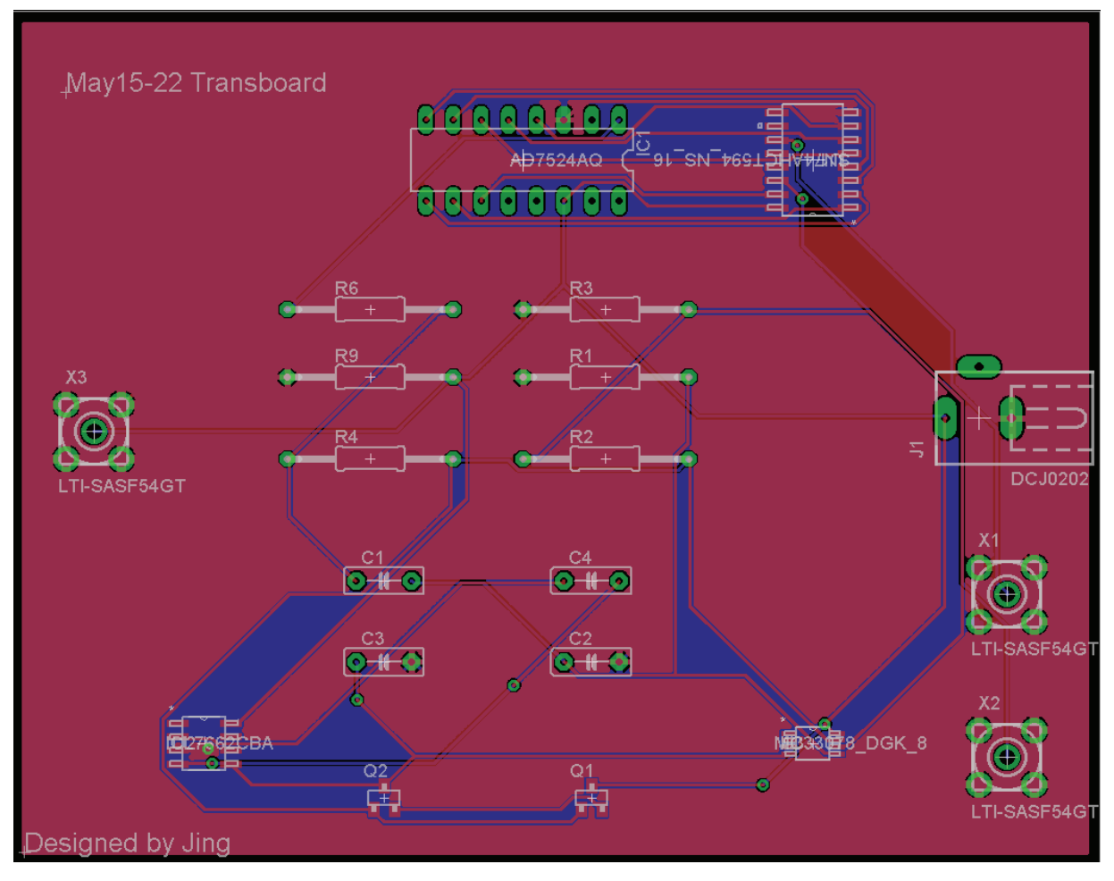
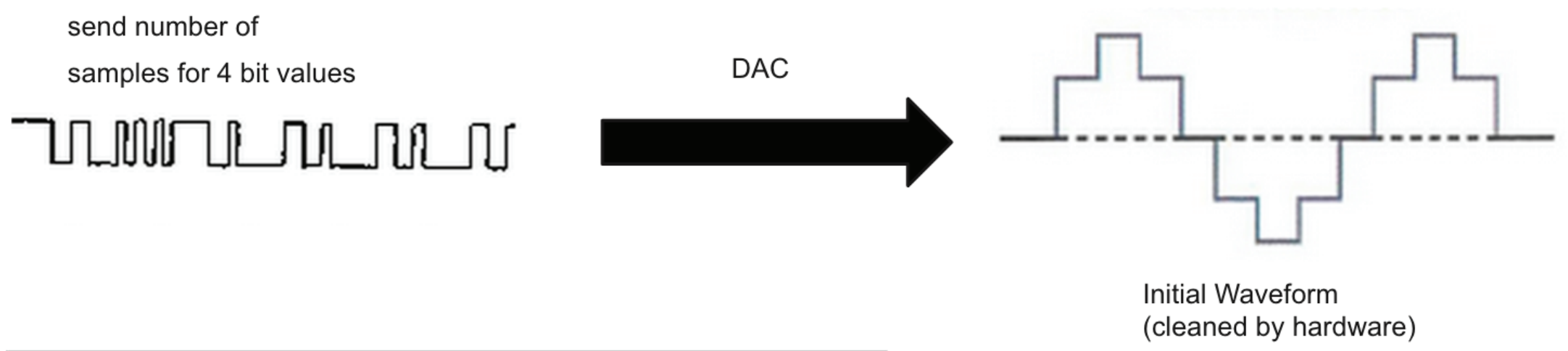


Figure 8: Our PCB board design

Software

The Software part consisted of three primary components all written in labview code: image processing, serial programming, and data acquisition. The as described in the hardware section, the serial programming module was used to create a analog waveform from a digital output. Figure 9 shows a basic representation of the differences in signal. Data aquisition protocols used the onboard NI 5752 cards' ADC to collecte data over the reflected wave's period shown in figure 10. Aquiring data over several channels resulted in "A-Lines" that were used for B-Mode imaging.

Figure 9: Serial Programming


send number of samples for 4 bit values
DAC
Initial Waveform (cleaned by hardware)

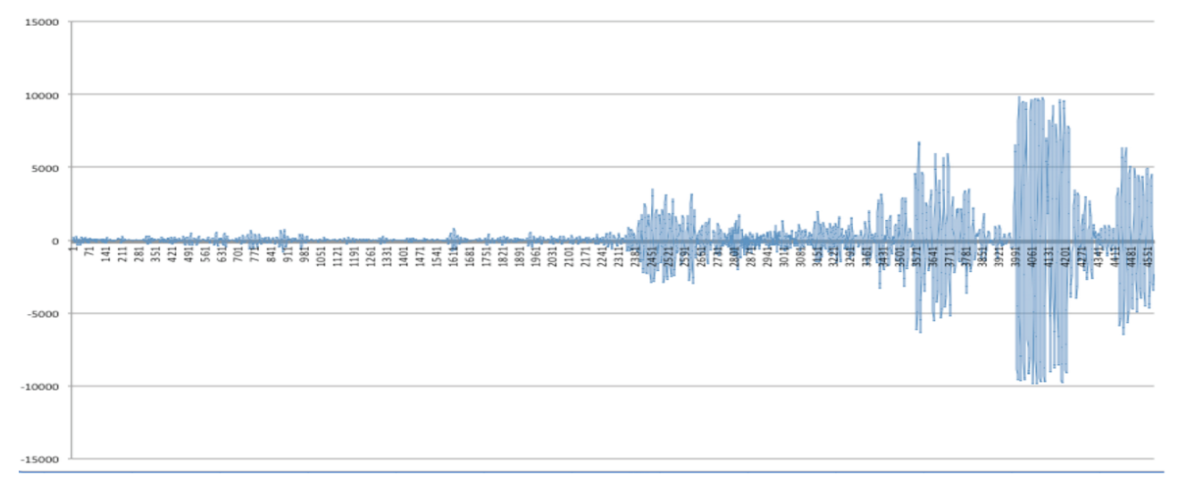


Figure 10: Example RF Input

The image processing portion is shown in the diagram in Figure 11. The RF input is demodulated with a Hilbert transform to remove negative frequencies. We then map the values into 8 bit space for black and white color. Since the data is captured over the period of about 6000 data points, the pixels are compressed into a smaller space so that the user can easily view the image.

Image Processing

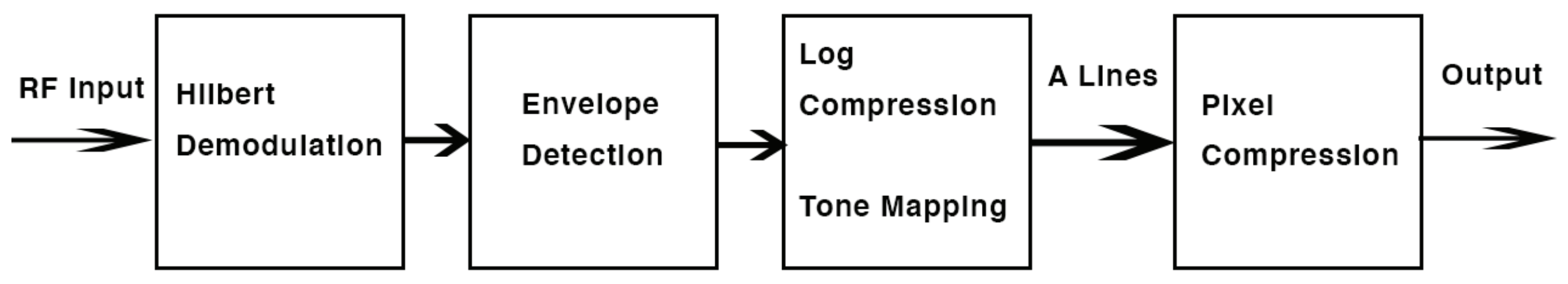

RF Input
Hilbert Demodulation
Envelope Detection
Log Compression
Tone Mapping
A Lines
Pixel Compression
Output

Figure 11: LabVIEW image processing block diagram.

Testing Results:

Testing the imaging system with a sample set of RF data yeilded the results shown in Figure 12. The sample data included results from a sytem running across 512 channels. After running tests on the timing of the imaging program, we found that the algorithm could produce an image in about 150ms. Which could produce video at a frame rate of about 8fps. This would work relatively well for a realtime system.

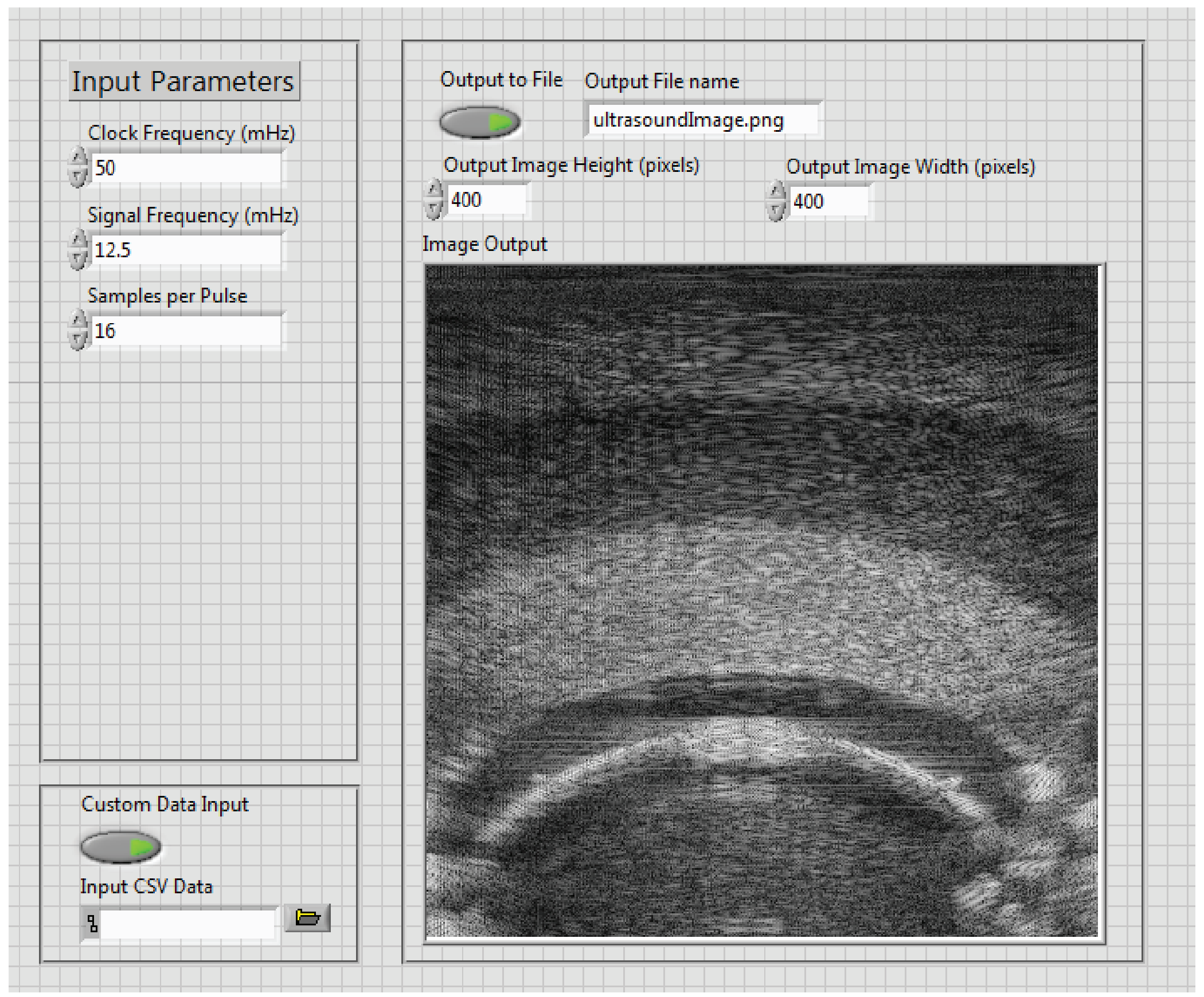

Input Parameters
Clock Frequency (mHz): 50
Signal Frequency (mHz): 12.5
Samples per Pulse: 16
Output to File
Output File name: ultrasoundImage.png
Output Image Height (pixels): 400
Output Image Width (pixels): 400
Image Output
Custom Data Input
Input CSV Data

Figure 12: The front panel of our imaging system. Most of the programming is abstracted behind the UI.

The images contain a bit of aliasing due to the pixel compression used to create a viewable image. More filters may be produced in the future to correct this issue. However, adding more filters to the imaging algorthm increases the time to produce a B-Mode image. Therefore, the framerate may drop.