Lab 1

UNIVERSITY of WASHINGTON

EE 443 4/6/16

Jake Garrison (1126716)
Jisoo Jung (1236577)

Problem 1

Description

For this task, we generated an 800hz sine tone with a sampling rate of 8khz in the form of a table and outputted it to the oscilloscope. We also generated a UART copy of this signal of length 256 and plotted it in matlab. The matlab plot is multiple periods since each period is 10 samples and we have 256 in our buffer.

This is the sine table we used:

```
short sine_f1[10]={0,588,951,951,588,0,-588,-951,-951,-588};
```

Results

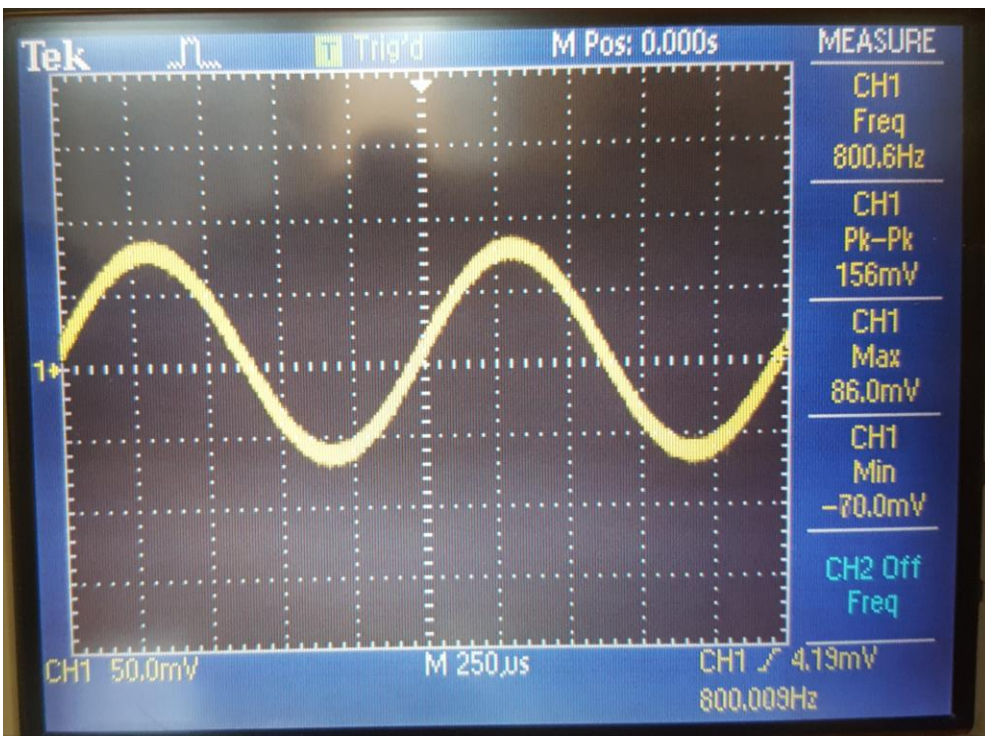

Figure 1. Demonstrates our sine wave outputted by the DAC is indeed ~800 Hz

Problem 2

In this task we used a tone generator (3k and 6khz) and sampled the tone in our DAC with a sampling rate of 8k. Since 6k exceeds the Nyquist limit (8k/2 < 6k) we observed aliasing on the 6k tone, but not the 3k tone.

3k Hz Tone Result

Below we show the time and frequency domain for the 3khz signal. No aliasing occurs, so the results are as expected

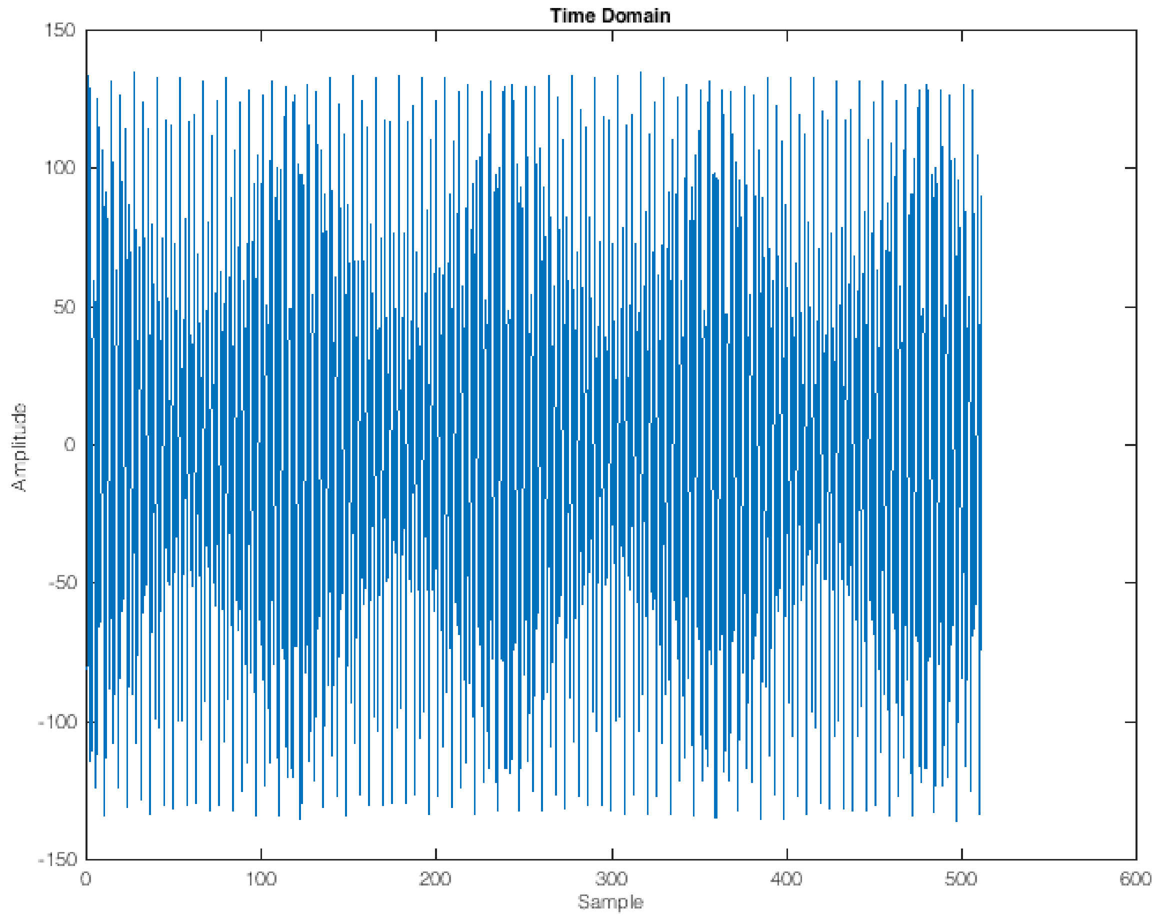

FIGURE 2. TIME DOMAIN REPRESENTATION OF THE 3KHZ INPUT SENT OVER UART

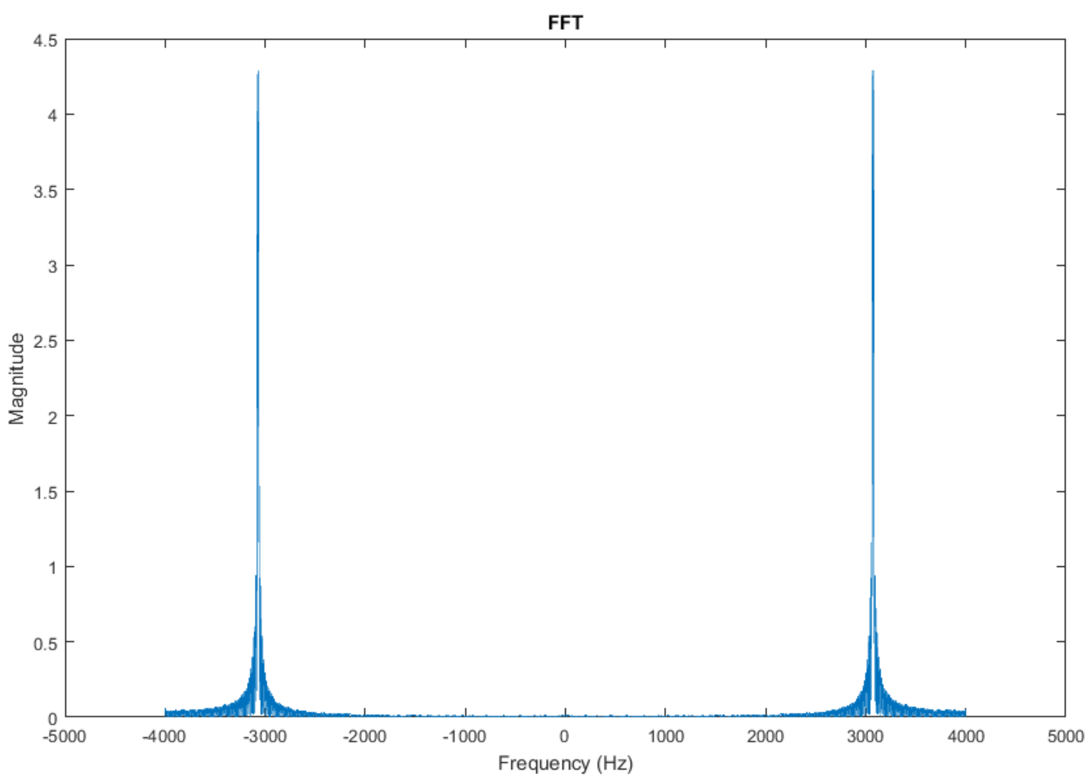

FIGURE 3. FFT RESULTS OF 3KHZ INPUT SIGNAL SENT OVER UART

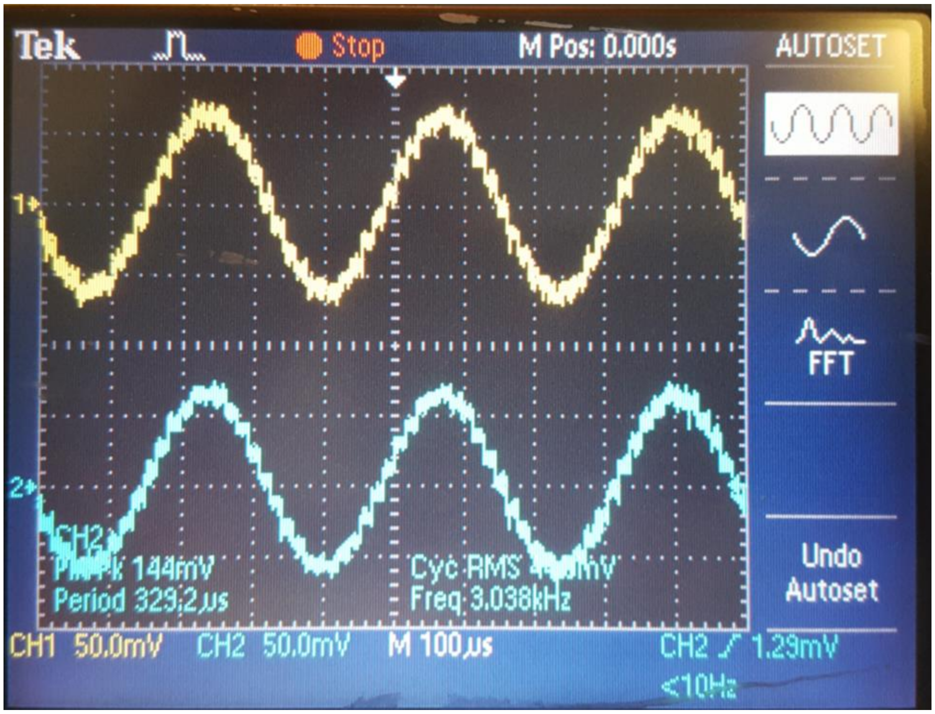

FIGURE 4. SHOWS THE STEREO 3K HZ INPUT TONE OUTPUTTED BY THE DAC

6k Hz Tone Results

As shown below, the results look nothing like what would be expected from a 6kHz sine wave. Since the Nyquist limit is exceeded, there is fold over in the frequency domain resulting in aliasing and a jumbled FFT response. This consequence is also present in the time domain representation as it is clear that it does not resemble a sine wave, or any periodic signal. In the oscilloscope picture, it is clear that our DAC is failing since the output is noise surrounded around 0.

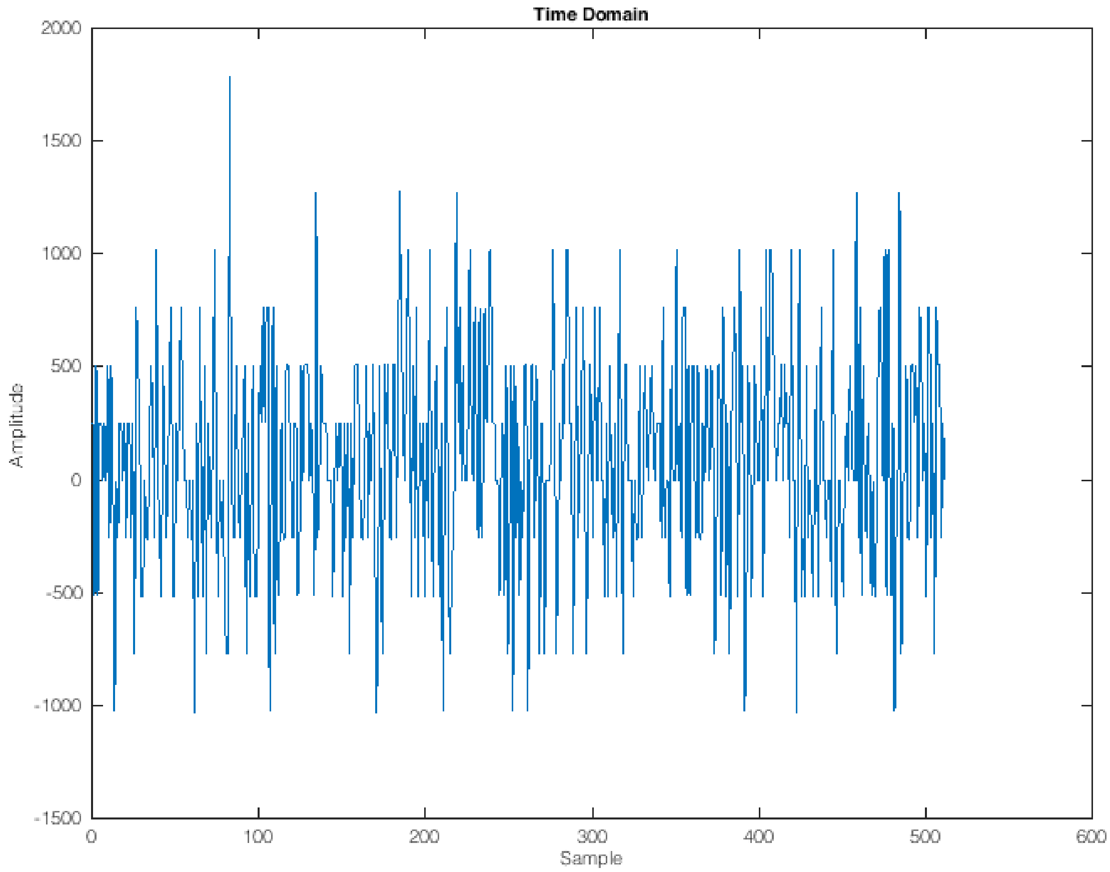

FIGURE 5. 6K HZ INPUT TONE SENT OVER UART. DEMONSTRATES ALIASING DUE TO HIGH FREQUENCY INPUT

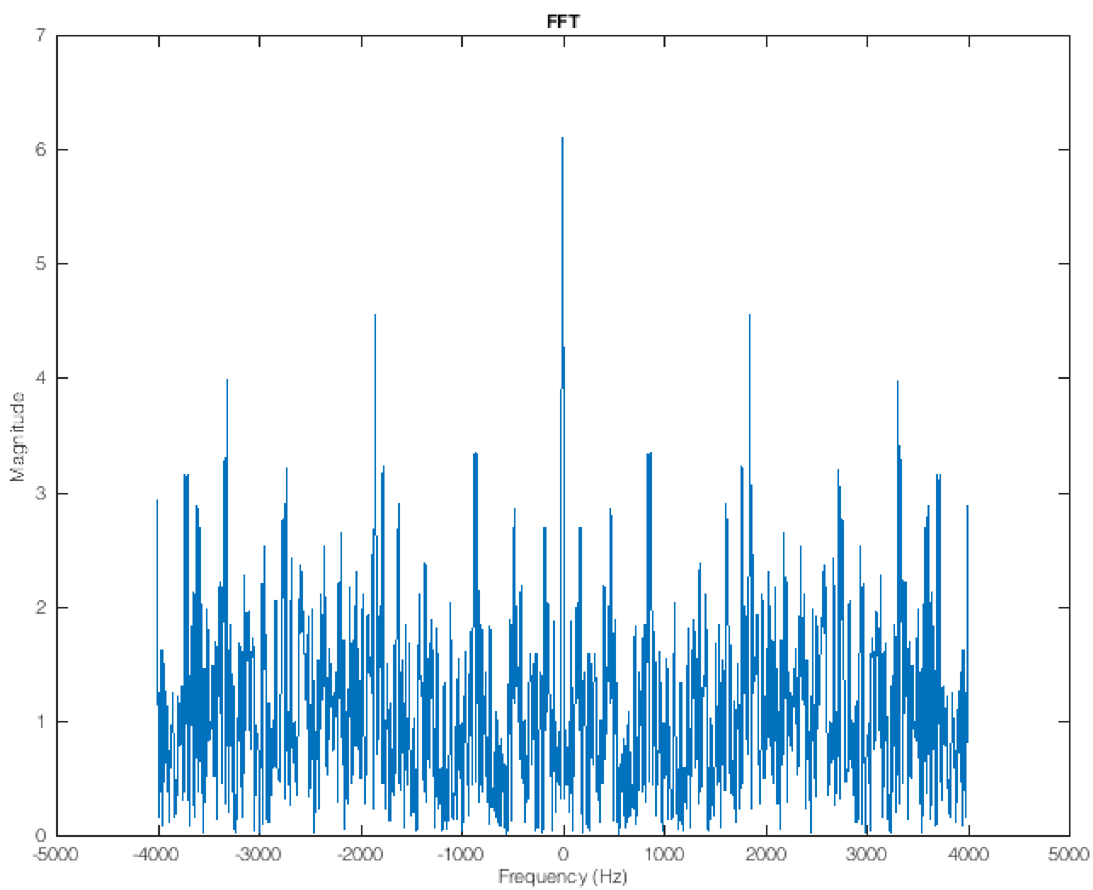

FIGURE 6. FFT OF ALIASED 6 HZ INPUT SIGNAL

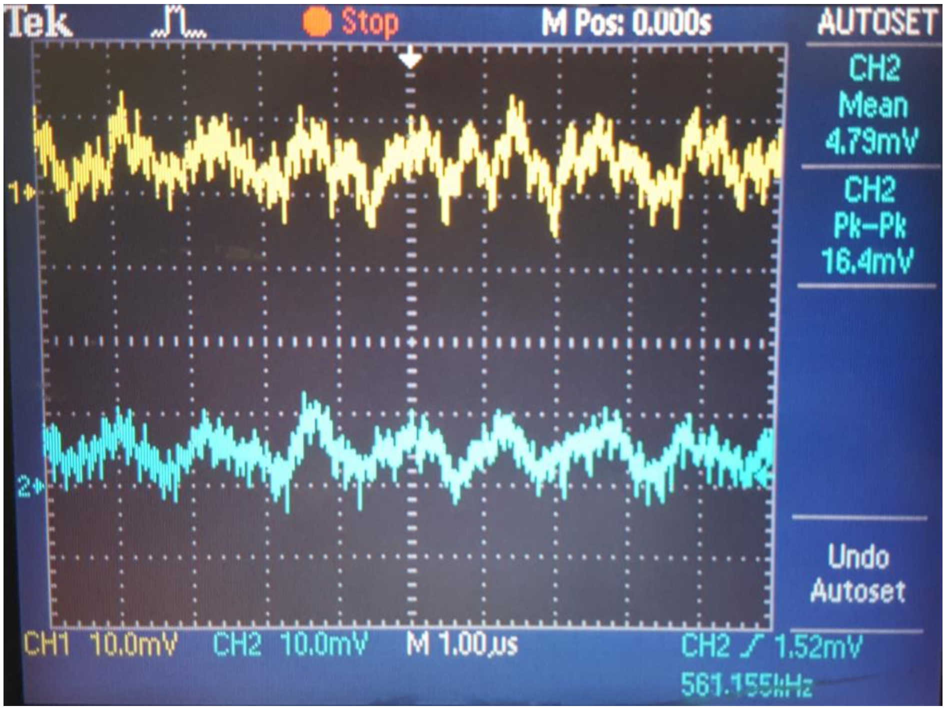

FIGURE 7. 6K HZ INPUT TONE OUTPUTTED FROM THE DAC. NOTICE IT IS JUST NOISE

Problem 3

In this task we utilized the switches to control the gain on three different sine waves. Our switches are configured such that each sine source has two switches and subsequently four bits. We used this to enable four gain states for each sine, 0, 1, 2, 3. The result is up to three sine sources at fixed, different frequencies with a gain based off the corresponding switch positions. To utilize 6 switches, we needed to initialize them in our main.c Our switch logic is shown below where alpha is the resulting gain. Two example configurations are shown below in the plots

```
alpha=(IORD_ALTERA_AVALON_PIO_DATA(SWITCH1_BASE) << 1)

      | IORD_ALTERA_AVALON_PIO_DATA(SWITCH0_BASE);
```

Our sampling frequency was 32khz and for our sine waves, the frequencies used were 3.2khz, 4.8khz and 8khz . For each frequency a sine table was generated to demonstrate a full period at the given frequency and sample rate. Different table sizes were needed to accommodate a full cycle depending on the frequency. The tables are shown below.

```
short sine_f1[10]={0,588,951,951,588,0,-588,-951,-951,-588};

short sine_f2[20]={0, 809, 951, 309,-588,-1000,-588, 309, 951, 809,

0,-809,-951,-309, 588, 1000, 588,-309,-951,-809};

short sine_f3[4]={0,1000,0,-1000};
```

Example 1 [100011]

In this example the switch configuration 100011 was used which means switch 0, 4 and 5 where on and the others off. This results in a gain of 1 for the 3.2k Hz signal, 0 for the 4.8k Hz and 3 for the 8k Hz. This is shown clearly in the FFT representation below.

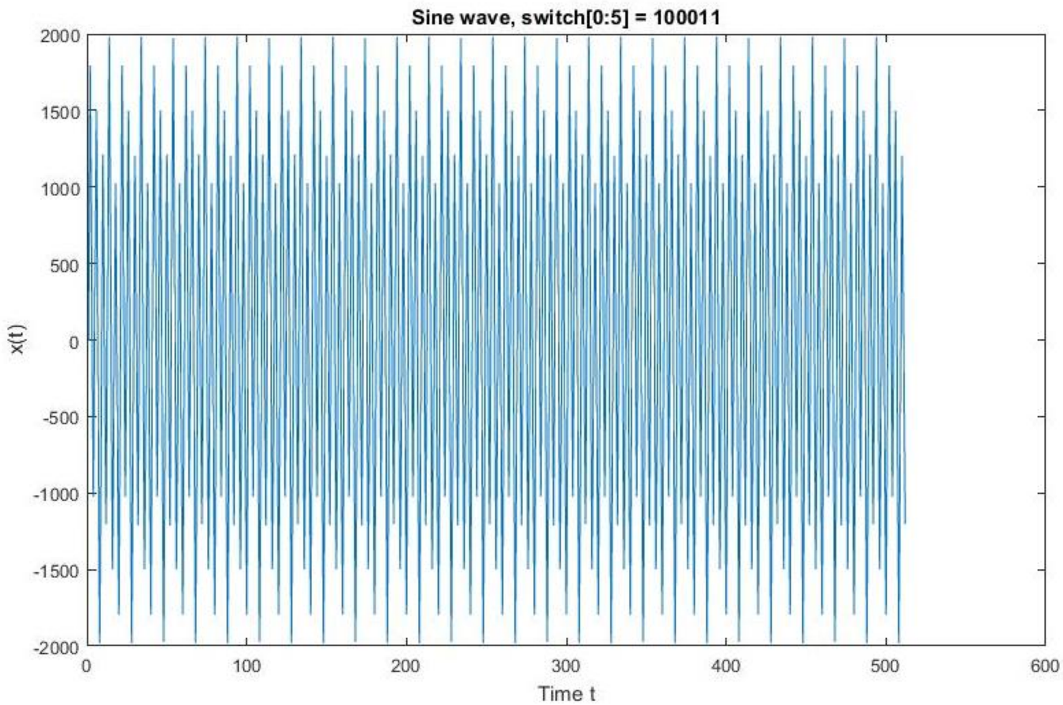

FIGURE 8. TIME DOMAIN SINUSOIDAL WAVE WITH ALPHA = 1, BETA = 0, GAMMA = 3

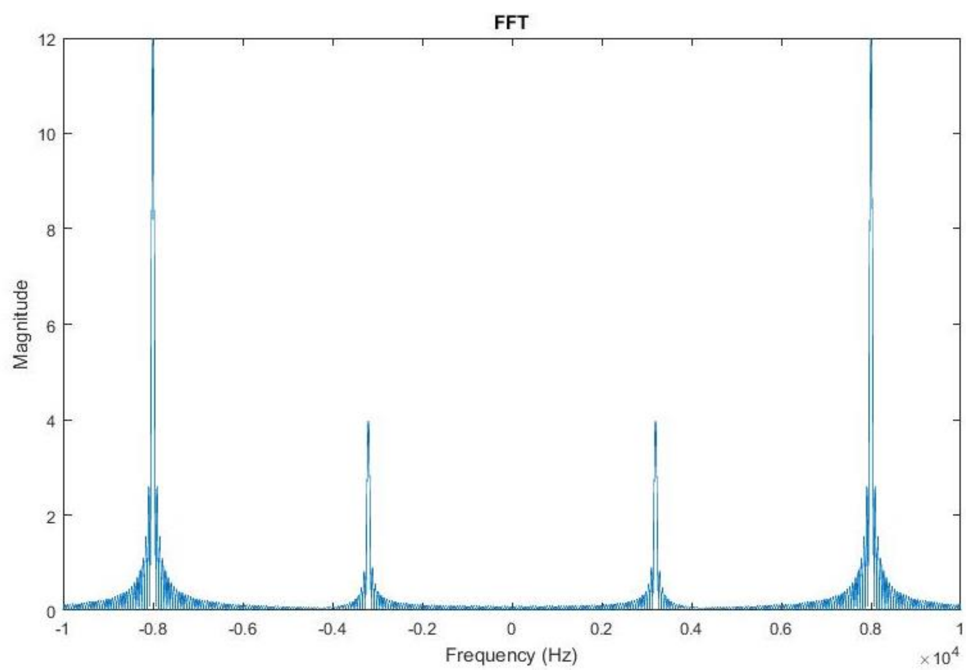

FIGURE 9. FREQUENCY DOMAIN OF SINUSOIDAL WAVE WITH ALPHA = 1, BETA = 0, GAMMA = 3

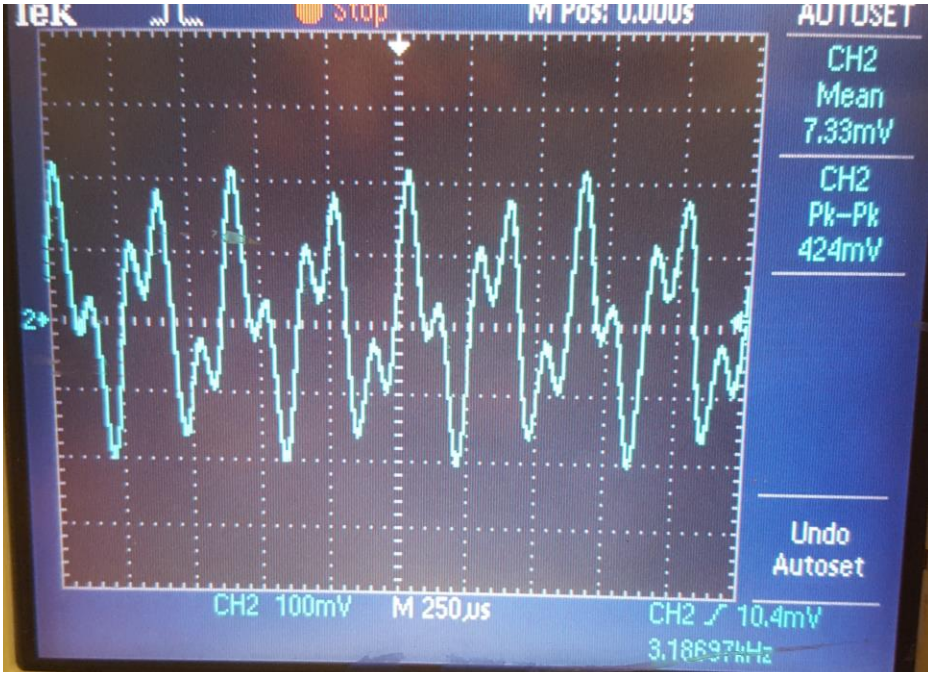

FIGURE 10. OSCILLOSCOPE OBSERVATION OF SINUSOIDAL WAVE IN TIME DOMAIN WITH ALPHA = 1, BETA = 0, GAMMA = 3

Example 2 [100100]

In this example the switch configuration 100100 was used which means switch 0 and 3 where on and the others off. This results in a gain of 1 for the 3.2khz signal, 2 for the 4.8khz and 0 for the 8khz. This is shown clearly in the FFT representation below.

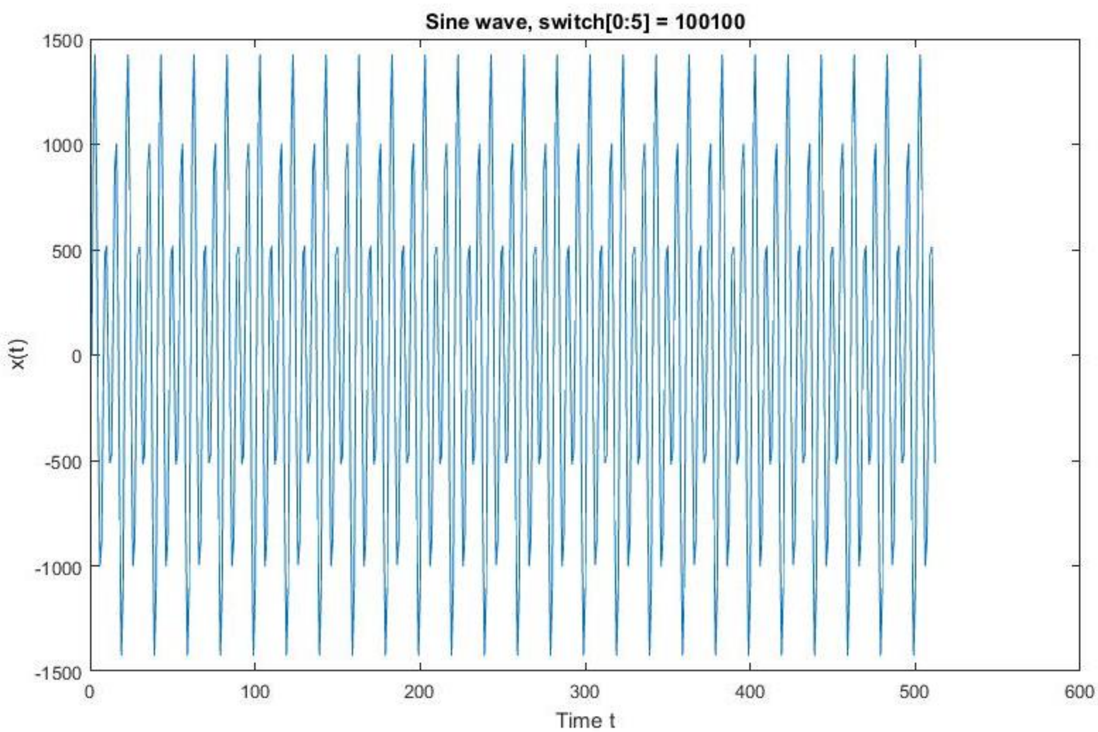

FIGURE 11. TIME DOMAIN SINUSOIDAL WAVE WITH ALPHA = 1, BETA = 2, GAMMA = 0

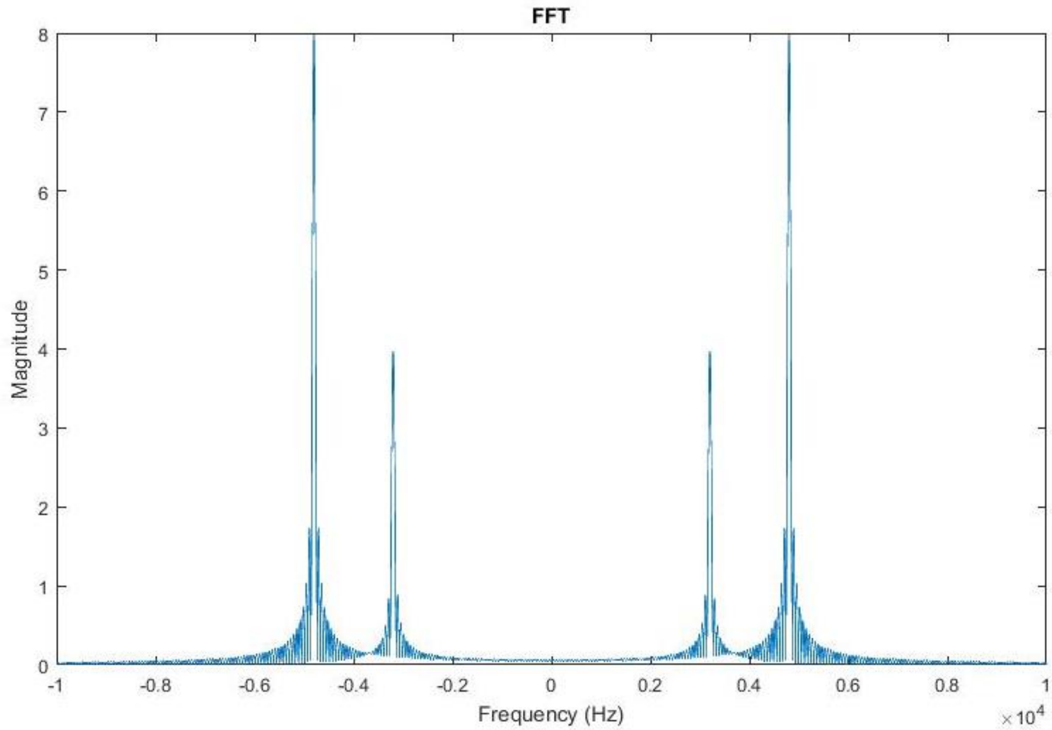

FIGURE 12. FREQUENCY DOMAIN OF SINUSOIDAL WAVE WITH ALPHA = 1, BETA = 2, GAMMA = 0

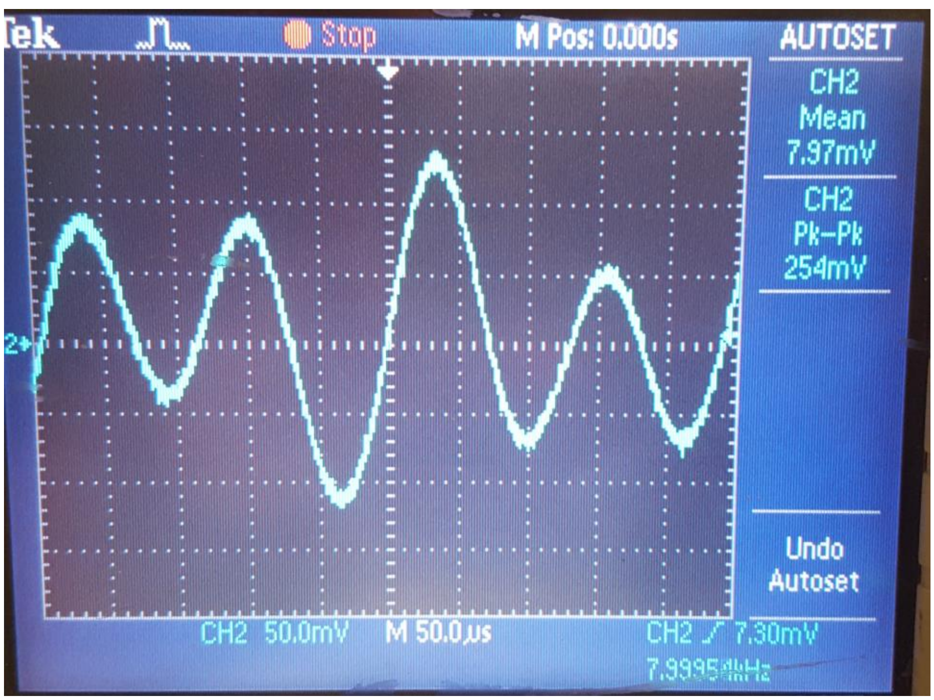

FIGURE 13. OSCILLOSCOPE OBSERVATION OF SINUSOIDAL WAVE IN TIME DOMAIN WITH ALPHA = 1, BETA = 2, GAMMA = 0

Problem 4

In this task we created channel mutes for incoming sound. We used two switches for the left and right channel respectively and used logic such that a switch off corresponded to muting that channel. In addition, an led corresponding to the channel was toggled. To implement this, we stored the state of the switch as a 0 or 1 and multiplied it by the corresponding channel, either muting or letting it pass. We used the switch 0 and1 interrupt to monitor the switch state, and the left/right interrupt handles to apply the gain constant (either 0 or 1). We also added an if statement to check the switch state and toggle the LEDs correctly.

Problem 5

In this task we were to convert an input sine wave into a square wave of equivalent frequency. In order to do this, we used an accumulator variable with a fixed step size. The logic is shown below

```
float stepSize = 2 * PI * f / Fs;
short sign = 1;
if (leftChannel < accum) {
      sign = -1;
}
accum = accum + sign * stepSize;
IOWR_ALTERA_AVALON_PIO_DATA(LEFTSENDDATA_BASE, sign * amplitude);
datatest[leftCount] = sign * amplitude;
```

Results

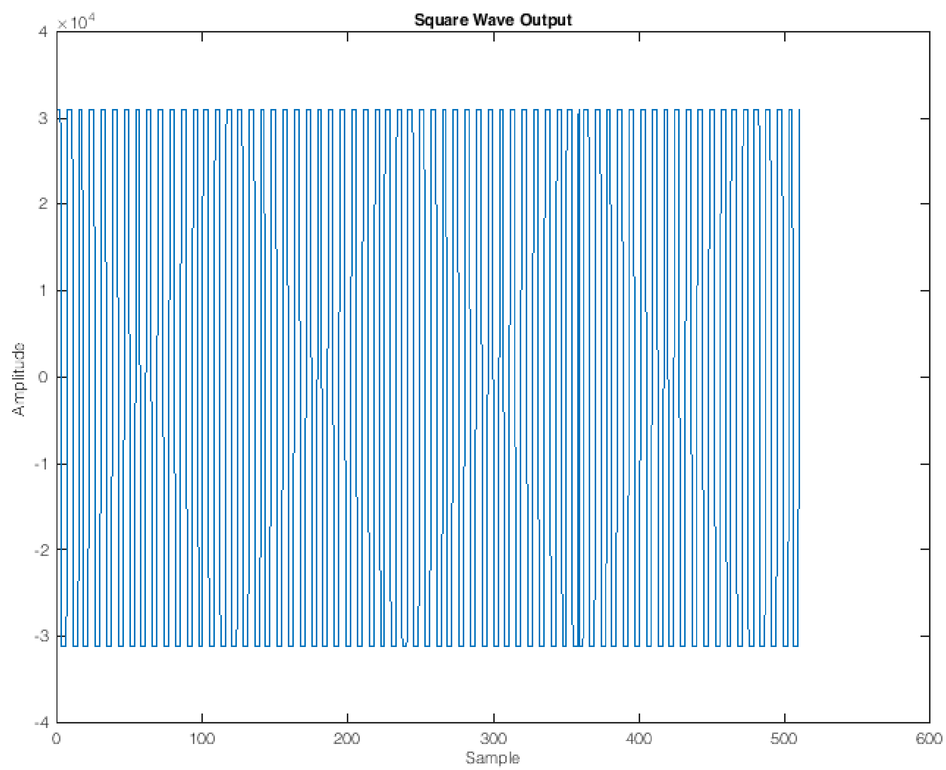

FIGURE 14. 1000HZ SQUARE WAVE OUTPUT

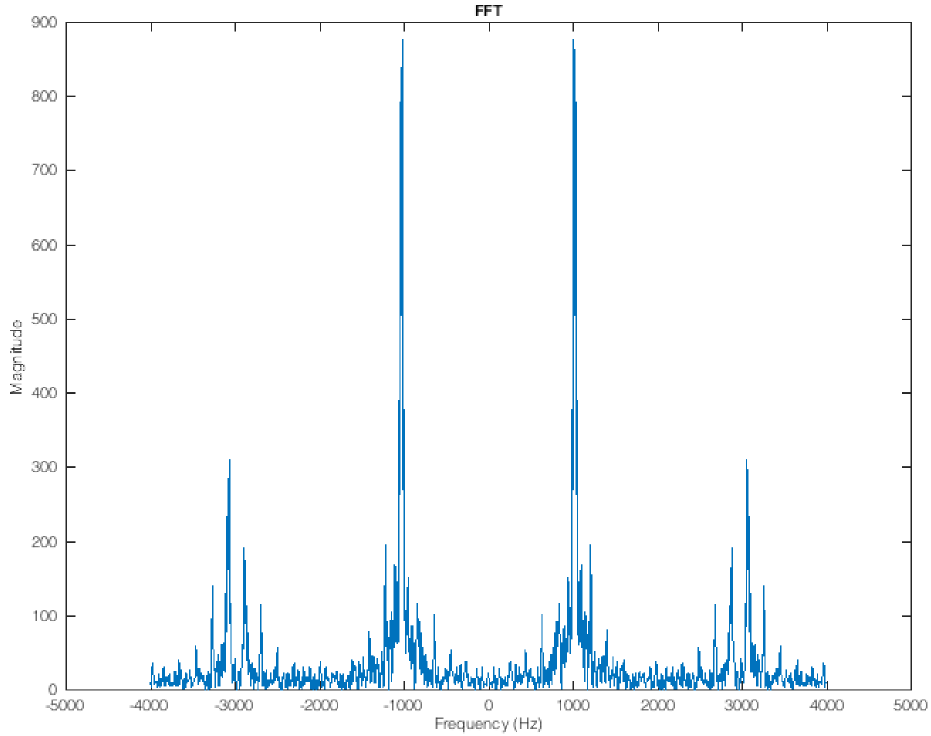

FIGURE 15. FFT SHOWS THE 1000HZ PEAK WITH THE HARMONICS NEEDED TO COMPLETE THE SQUARE WAVE

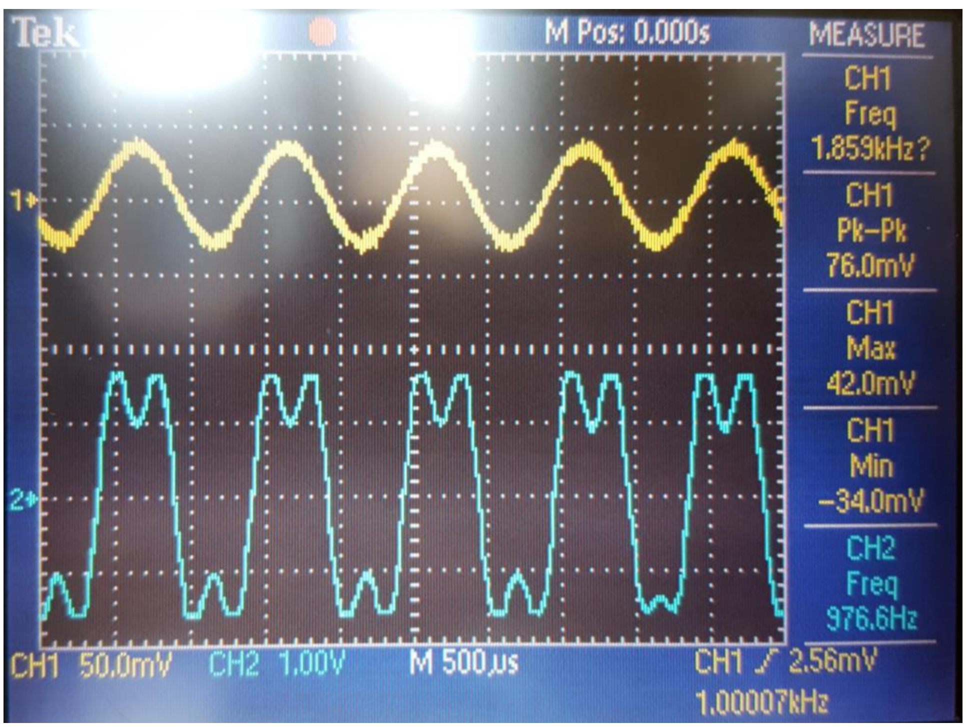

FIGURE 16. OSCILLOSCOPE SHOWING THE INPUT SINE AND OUTPUT SQUARE WAVE