REAL-TIME SOUND VISUALIZATION

By

Lin Le

Qian Chen

Xinyue Yu

Final Report for ECE 445 Senior Design, Spring 2018

TA: Dongwei Shi

2 May 2018

Project No. 43

Abstract

This report is about designing and building a small and affordable device which takes sound as an input and outputs the corresponding note of the pitch of the input sound on a connected VGA display. The device also has LEDs on it, which indicates in relative intensity of the input sound. This device allows people who is not good at music to record the music score of a certain music. To accomplish this goal, this system consists of several modules. There are controllers in the systems. The ARM Microcontroller(MCU) is responsible for performing Analog to Digital Conversion(ADC) and fast Fourier Transform (FFT). The pitch of input sound is determined by the frequency component from FFT. Then the result is sent to FPGA with a customized BUS. The FPGA decodes which node was been sent, then displays the image on VGA display through a VGA DAC. The FPGA also decodes the intensity of the input sound and displays the volume unit on LEDs, which works as a Volume Unit Meter.

Contents

I. Introduction

1.1. Objective

Nowadays, there are thousands of mobile phone application have the function of pitch detection to recognize pitch like Vocal Pitch Monitor in the Google Play, Pitch Tuner or Tenuto in the Apple Store [1]. But most of pitch detection application do not have the transcription option to show on and few that has the feature is really expensive to purchase in the apple store. Hence it is interesting to have a hardware version to combine all the fantastic features together and make it portable and affordable for musicians or music lovers with a cheap price.

This project creates a small embedded monophonic music transcription system that can transcribe one note at a time when there is an input. The next step is to store the melody into designed memory chip. A FFT analysis will be used inside our microcontroller. There will be LED effect accompanied with input sound when the sound is being detecting, which works as a volume unit meter. Once the musical notes has been detected, it will automatically display on a VGA display.

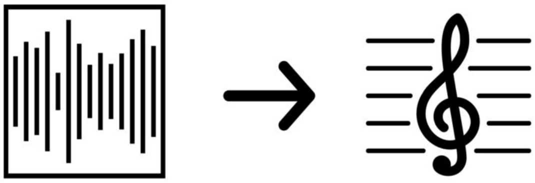


Figure 1. Basic Idea of Real-time Sound Visualization

1.2. High-level Requirement

- It should detect the correct pitch for notes produced by vocal or other instruments, and display the correct note on display.
- There should be no noticeable delay between incoming audio signal and digital output. The delay should be no larger than 250 ms (¼ second).
- The MCU drives a series of LEDs as a volume unit (VU) meter, which gives the intensity of input sound, like those on the recorder or CD player.

2. Design

2.1. Design Procedure

FFT vs Autocorrelation

The ultimate goal is to detect which note is playing. To achieve the goal of performing pitch detection, a frequency spectrum analyzation is mandatory. The method is to perform Fast Fourier Transform(FFT) on the sound data collected. Autocorrelation is also an feasible method. The FFT is chosen because Fourier Transform gives information of the whole frequency spectrum.

Choosing MCU

To do FFT, first a microphone is connected to the MCU. The microphone translates sound energy to electric energy. Then MCU sample data and digitize all the data and store in memory. Therefore, an ARM based MCU is chosen to do all the calculation. The chosen MCU, has built-in DSP module and capable of utilizing DSP module with DSP library provided by supplier.

Visual Output

The MCU is dedicated on performing FFT algorithm, therefore, FPGA is chosen to display result on a desktop monitor through VGA port. Moreover, FPGA is versatile, able to control LEDs and give light effect. The MCU writes the detected pitch in to databus between MCU and FPGA. Then the FPGA will read the data from the data bus and decode the data and decide what should be displayed on the VGA display.

Circuit Design

All the component will be mounted on a PCB (printed circuit board) except display and light effect LEDs. The FPGA is also responsible to control light effect LEDs. Which works as a volume unit meter; therefore, the use can easily notice if the volume of input sound is too low or too high.

2.2. Block Design

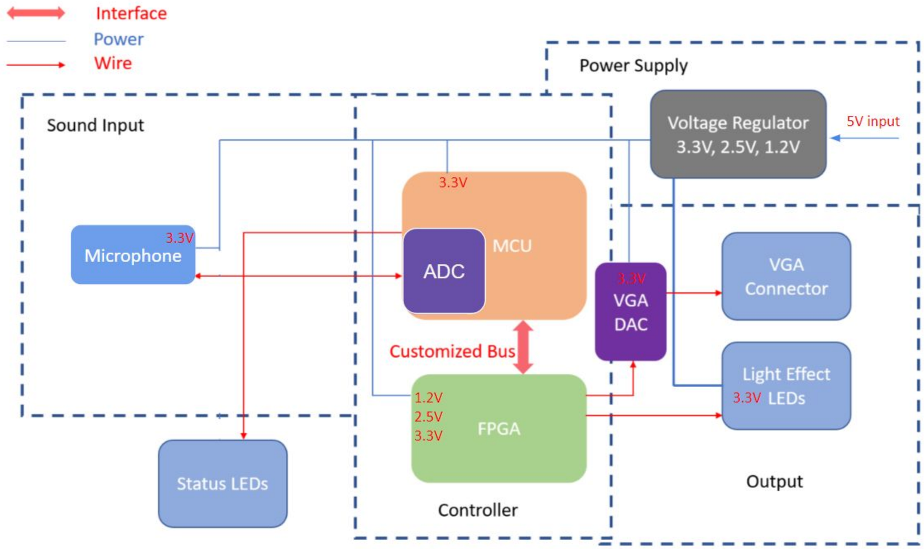


Figure 2. Block Diagram

2.2.1. Sound Input, Video Output

Microphone

The microphone is connected to the MCU. The A/D module in MCU takes voltage between 0v to 3.3v; therefore, the output of the microphone module must be between 0 to 3.3v.

Status LEDs

The status LEDs is controlled by the Microcontroller Unit and will display to the user if the MCU is working properly and whether the microphone and speaker is on or off.

VGA port

VGA is an analog signal, therefore the VGA port is driven by a VGA DAC. The DAC connect to FPGA by GPIO pins of FPGA. Then the pinout will be assigned through IDE. FPGA controls what to display on the VGA display.

2.2.2. Controller

MCU

The project uses a STM32F207 MCU [2], because we need a powerful MCU to perform FFT in real-time. The current MCU is ARM Cortex M3 based, and capable of doing DSP instruction. The MCU is also responsible to do A/D conversion, the MCU has build-in A/D convection module. MCU will collect data at sampling rate 22.1 kHz from A/D port, write data into memory and perform FFT.

FPGA

One of the important roles FPGA played in our system is to work as a VGA display controller. FPGA is responsible to display result on a desktop monitor through VGA port. Moreover, FPGA is also responsible to control LEDs. The LEDs display the input sound intensity as a VU meter. The FPGA used is Altera Cyclone III EP3CE144E22.

2.2.3. Other

IDE

Keil uVision as an IDE for STM32 MCU. The IDE is capable of compiling the code, uploading the compiled program in flash of MCU and debugging.

Quartus II is the official IDE realised by altera and works for FPGA. The simulation of FPGA will be done by modelsim.

2.2.4. Power Supply

A robust power supply circuit is implement for this project. There are three linear voltage regulators as power supply. FPGA has three input voltages of 3.3V, 2.5V, 1.25V. MCU takes 3.3V as power input.

2.3. Circuit Design

2.3.1. Circuit Schematics

The circuit is carefully designed and drawn by EAGLE. The full schematic is in appendix B. Microphone is connected to GPIO pins of MCU. VGA dac is connected to the GPIO pins of FPGA.

2.3.2. Internal circuit and block diagram of FPGA

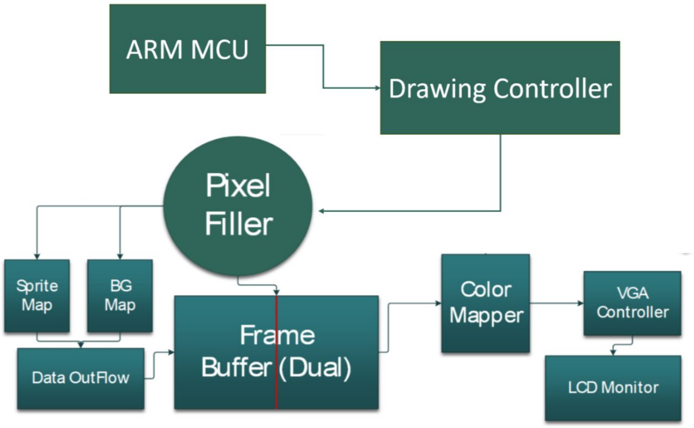


Figure 3. Block diagram of note drawing module in side FPGA.

The controller decode the pitch code sent by MCU, and draw the corresponding node on screen.

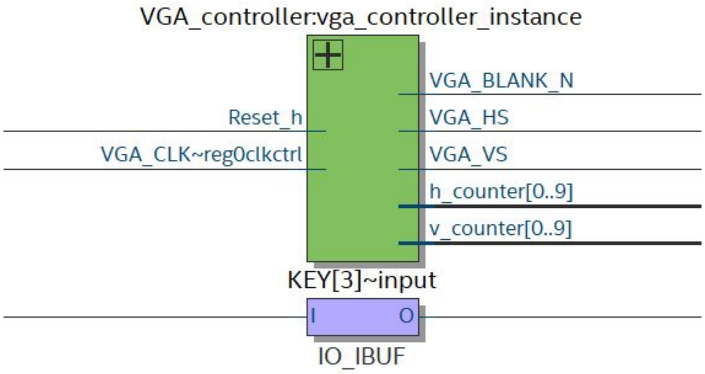


Figure 4. VGA controller

This module aims to control the timing of the VGA port, in which both the monitor vertical and horizontal synchronization signals are generated. Moreover, the 25 MHZ VGA_Clk signal is used as the input to this module, which aims to provide a timing standard for VGA output signals.

2.4. Algorithm

The software information in the project divides into two parts. The A/D conversion (Fg.12) below handles with the pitch detection inside the microcontroller unit. The algorithm will be briefly introduced in the following module. The Data Transmission between FPGA and Display part (Fg.13) deal with displaying final result as we expected like a flowing transcription.

2.4.1. A/D Conversion and FFT

The algorithm for pitch detection on MCU is shown below. MCU collects data from microphone and perform pre-processing and FFT. The result of FFT will be stored in memory. The highest peak in frequency domain will be considered as the fundamental frequency of the input sound.

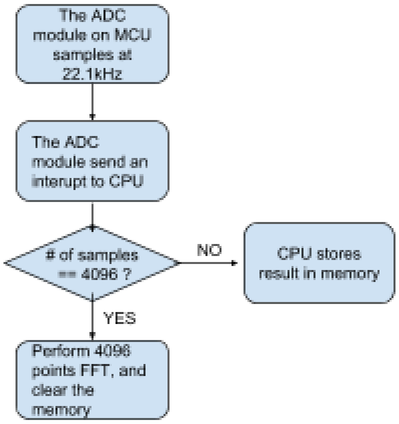

The ADC module on MCU samples at 22.1kHz
The ADC module send an interupt to CPU
# of samples == 4096 ?
NO
CPU stores result in memory
YES
Perform 4096 points FFT, and clear the memory

Figure 5. A/D conversion and FFT

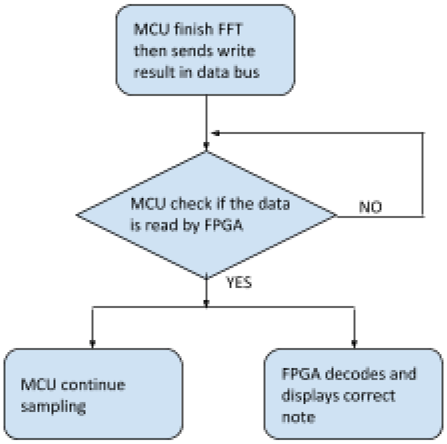

MCU finish FFT then sends write result in data bus
MCU check if the data is read by FPGA
NO
YES
MCU continue sampling
FPGA decodes and displays correct note

Figure 6. Data Transmission

2.4.2. Data Transmission between FPGA and MCU

The algorithm for data transmission between FPGA and MCU is shown above. Once the FPGA receives the signal from MCU, it will show on the display immediately and takes corresponding effect as we expect on display and LED.

2.5. Calculations

2.5.1. Cut-off frequency

The frequency response of filter design is a very important tolerance in this project. Before performing the low pass filtering, attenuation of the stop band should be high enough. Meanwhile, the cutoff frequency cannot be too low or too high, otherwise either lost information or induced aliasing will make all the data unacceptable.

In the old plan, we designed a low pass filter. The data went through a downsampling after sampling. The data has a decimation at down sampling rate 8:1. In this case, U = 8

$$x_u(m) = \begin{cases} x(n) & m = Un \\ 0 & else \end{cases}$$

It is clear that component in frequency domain with frequency $|\omega| > \omega_0/8 \ = 5.5rad$ will cause aliasing. If we want to use a averaging filter as a low pass filter with length of 8.

$$H[n] \ = \ [1/8,\ 1/8,\ 1/8,\ 1/8,\ 1/8,\ 1/8,\ 1/8,\ 1/8]$$

The frequency response is as plot.

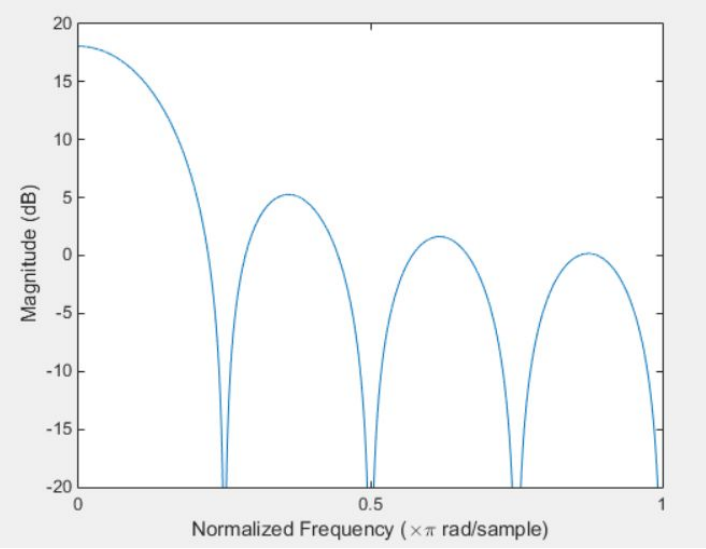


Figure 7. Frequency response of a averaging filter of length 8

From the frequency response, the half width of the main low is smaller than pi/4. Therefore the averaging is a good pre-downsampling filter. However, there was some minor problem, the magnitude of sidelobe is a bit too high. The highest value of the side lobe is at 5dB, which reduces the SNR (signal to noise ratio).

2.5.2 Accuracy of pitch detection

The accuracy of detected pitch is important to our project. According to Wikipedia of piano pitch frequency, “the following equation gives the frequency f of the nth key, as shown in the table[2]:

$$f(n) = \left(\sqrt[12]{2}\right)^{n-49} \times 440\,\mathrm{Hz}$$

Eq. 1

For example, A_5 has frequency 880.000 Hz, and its next pitch is Bb_5 whose frequency is 932.328 Hz. The tolerance of each pitch will be $\pm 3\%$, that means, pitches in range from 853.836 HZ to 906.164 Hz will all be recognized as A_5. Since the highest frequency after FFT will be tested as the pitch, there is 5% error rate saved for background noise.

3. Design verification

3.1. PCB (Printed circuit board)

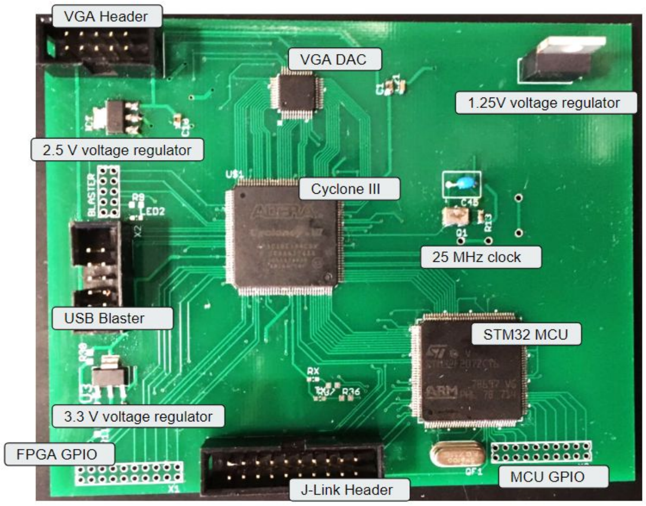


Figure 8. Completed Printed Circuit Board

3.2. FPGA and Display

The design of VGA controller circuit is accomplished well. The VGA controller gets the input from MCU through 16 bit bus. The first 8 bits carries the frequency information, which is going to be translated to music notation.The last 8 bits represents the intensity, which will be used to indicate the volume.Drawing controller is a NOIS II processor, which is in charge of decoding the frequency and the intensity. Main logic writing in C code runs on NOIS II. The Pixel Filler is similar to old Pixel Process Unit in old computers, which takes four input values, the starting coordinate of sprite, ending coordinate of the sprite, starting coordinate of the location to be filled and the ending coordinate of the location to be filled. Pixel filter firstly gets background color indices from background map. Then, it finds the music note from the correct place from the sprite map. Finally, it writes both background and sprite color indices into frame buffer.

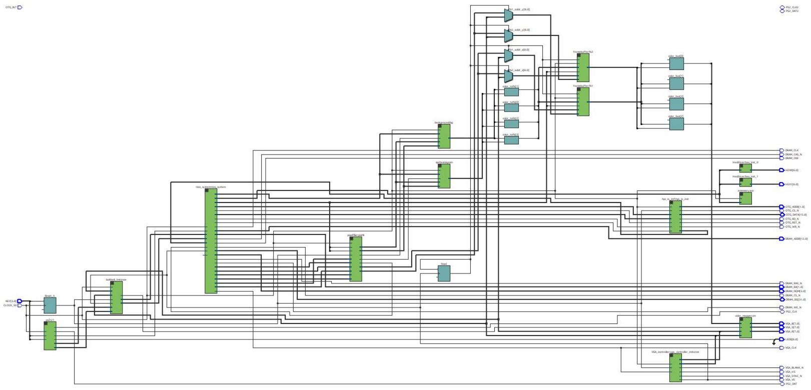
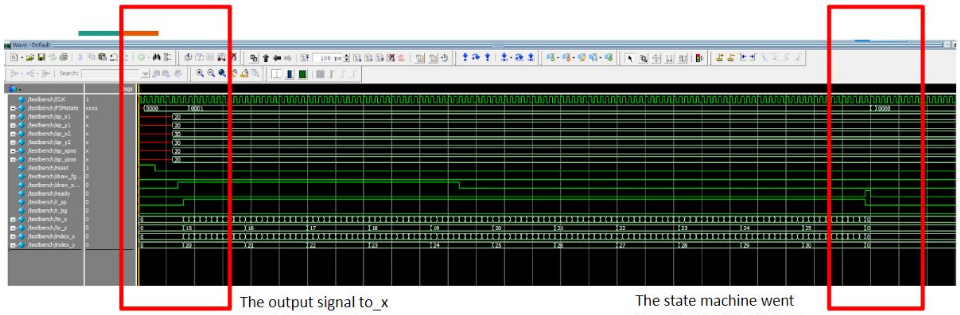


Figure 9 . The RTL synthesised from system verilog.



```
/testbench/CLK          1
/testbench/FSMstate     xxxx
/testbench/sp_x1        x
/testbench/sp_y1        x
/testbench/sp_x2        x
/testbench/sp_y2        x
/testbench/sp_xpos      x
/testbench/sp_ypos      x
/testbench/reset        1
/testbench/draw_fg...   0
/testbench/draw_s...    0
/testbench/ready        0
/testbench/r_sp         0
/testbench/r_bg         0
/testbench/to_x         0
/testbench/to_y         0
/testbench/index_x      0
/testbench/index_y      0
```

The output signal to_x to_y is incrementing and the output signal for index_x index_y is also incrementing.

The state machine went from IDLE to DR_SP then FIN. The ready signal rises at the FIN state.

Figure 10 . The simulation wave of Pixel filler

The background map is an array of indices representing RGB color. It is written in System Verilog. The image shown on screen are three rows of five line staves.

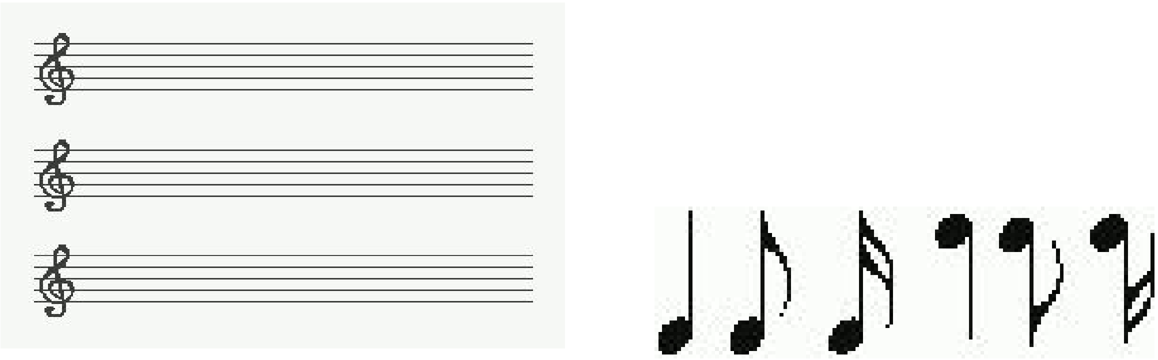
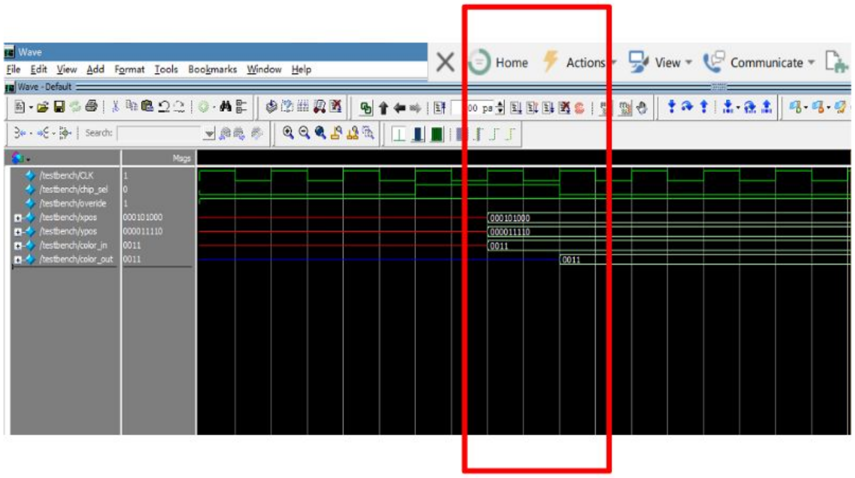


Figure 11. Image of background and sprite for drawing transcript

The frame buffer has two identical parts, which are responsible for storing odd and even number of frame. When writing in one buffer, the other buffer is used for displaying.



	Msgs
/testbench/CLK	1
/testbench/chip_sel	0
/testbench/overide	1
/testbench/xpos	000101000
/testbench/ypos	000011110
/testbench/color_in	0011
/testbench/color_out	0011

The data is write into x = 40, y = 30. And the same data is read out from color_out port.

Figure 12. The simulation wave of frame buffer

Color mapper reads data from frame buffer and maps indices to 30 bits RGB colors. Then it sends the colors to VGA DAC controller.

The VGA DAC controller also generates all the time signal for VGA DAC, including blank, Horizontal sync, VGA PSAVE, and VGA clk. The VGA clk is 25Mhz, generated by PPL in FPGA. The output resolution is 640×480 by pixel.

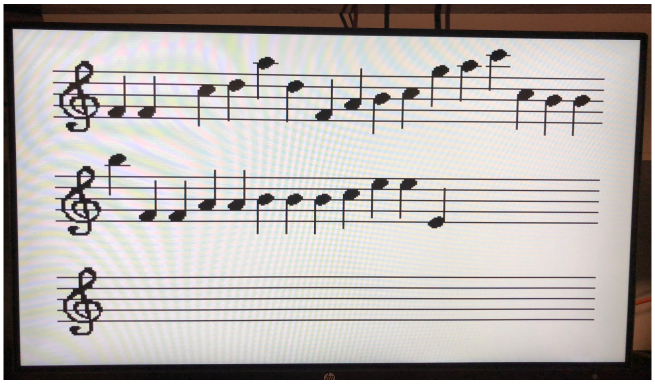


Figure 13. VGA Verification Result

3.3. Microphone input and output

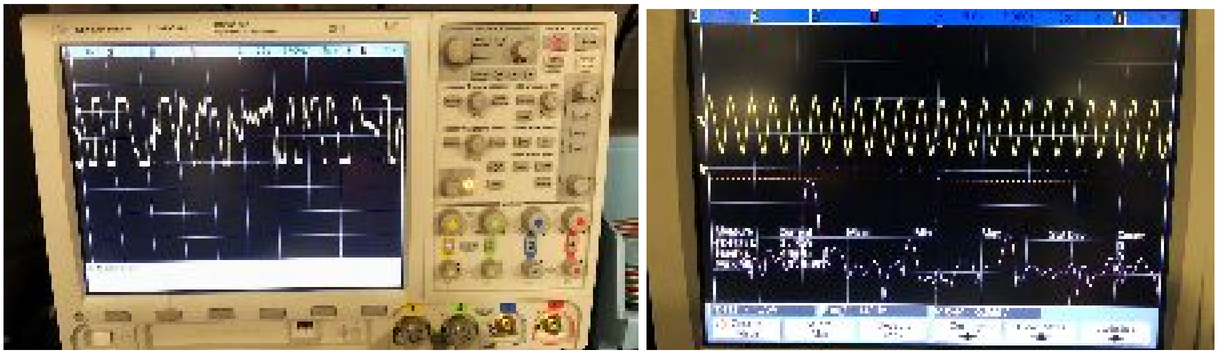
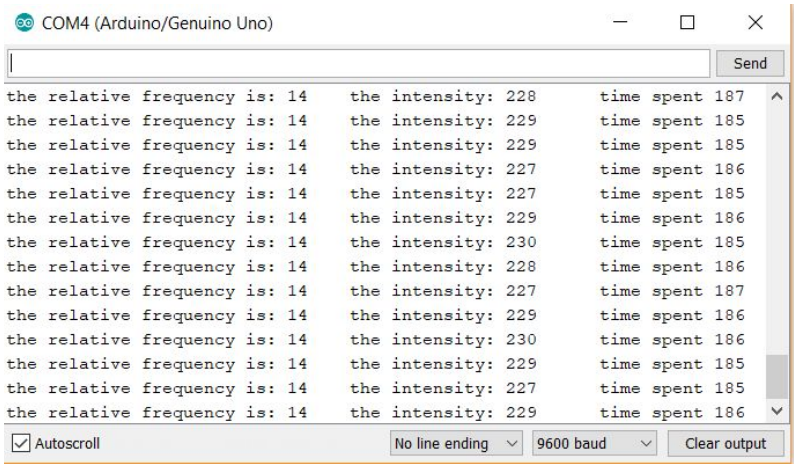


Figure 14. Microphone Verification Results

The microphone is connected to oscilloscope. In the left picture, a loud sound was sent to the microphone. The output of microphone saturated and maximized at 2.5V. On the right a 440 Hz sound is sent to the microphone. The yellow line is the waveform of time domain signal, the purple line is the frequency spectrum. The strongest peak is at 440 Hz with intensity -5.8 dBV. Without the input sound the highest peak in frequency spectrum is at -20dBV. The microphone has a SNR (Signal to Noise Ratio) about 15dB.

3.4. MCU, pitch detection and output

The microphone is connected to the GPIO pins of MCU. A 440 Hz sinusoidal sound is input into microphone. The sound is recorded by MCU and be processed with FFT. The MCU prints out the result through UART serial port.



```
the relative frequency is: 14    the intensity: 228     time spent 187
the relative frequency is: 14    the intensity: 229     time spent 185
the relative frequency is: 14    the intensity: 229     time spent 185
the relative frequency is: 14    the intensity: 227     time spent 186
the relative frequency is: 14    the intensity: 227     time spent 185
the relative frequency is: 14    the intensity: 229     time spent 186
the relative frequency is: 14    the intensity: 230     time spent 185
the relative frequency is: 14    the intensity: 228     time spent 186
the relative frequency is: 14    the intensity: 227     time spent 187
the relative frequency is: 14    the intensity: 229     time spent 186
the relative frequency is: 14    the intensity: 230     time spent 186
the relative frequency is: 14    the intensity: 229     time spent 185
the relative frequency is: 14    the intensity: 227     time spent 185
the relative frequency is: 14    the intensity: 229     time spent 186
```

Figure 15. The pitch detection algorithm is performed onto the input sound. The sound is 440 Hz. The relative frequency is 14, i.e. 440 hz. The time spent on each pitch detection is about 186 ms.

3.5. Power supplies

To test the output range and stability of power supplies. All power supplies are connect to a resistor box. The output voltage of power supply is recorded under different load. Smaller load has a larger output current. From the graph, it is clear that all three power supplies give stable output voltage. The output voltage drops when the output current exceeds the maximum output current.

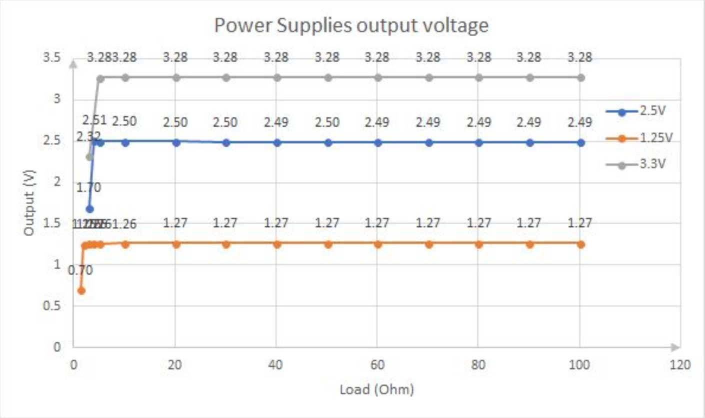


Figure 16. Power Supply Verification Result

4. Cost

4.1. Parts

Part	Manufacturer	Retail Price($)	Bulk Purchase Cost($)	Actual Cost($)
Microcontroller: ARM® 32-bit Cortex®-M3 STM32F207ZCT6	STMicroelectronics NV	$ 10.74	$ 6.09188	$ 40
FPGA: Altera Cyclone IV EP4CE10E22	Intel Altera	$ 24.00	$ 24.00	$ 96

ADV7123KSTZ50, VGA DAC	Analog Device	$ 10.99	$ 7.58	$ 44
Voltage regulators. LM1084IT-ADJ, LM317MDCY	Texas Instruments	$ 2.6	$ 1.27600	$ 10
SMD Capacitors and resistors	N/A	N/A	N/A	$ 50+
J-Link programmer, USB blaster	SEGGER, Altera	$ 20	N/A	$ 40
crystal clock oscillator	Abracon LLC	$ 1.06	$ 0.81180	
PCB	PCBWay	$ 25	< $ 1	$ 140 (two round + shipping)
Total		N/A	N/A	~ $ 450

Table 1. Cost of Parts

4.2. Labor

Our development cost for this project is to be $30/hour, 15 hours/week for three people. There are total 16 weeks for this semester and we committed 65% for our final project.

$30/hour * 15 hours/week* 16 weeks * 65% * 3 people = $14,040

The total cost will be the sum of labor cost and parts cost, which is $14490.

5. Conclusion

5.1. Accomplishments

The high level requirements are all met; although not perfectly.

- It correctly detect the pitch for sinusoidal monotonic sound produced wave generator, and displays the correct note on display. However, the multi-tonic sound produced by some instruments can not and correctly detected.
- There is no noticeable delay between incoming audio signal and digital output. The fourier transform takes less than 3ms. The output rate is the 1/4096 of the sampling rate, i.e. about 5.5Hz. There for the delay between incoming audio signal and digital output is less than 200ms.
- The FPGA drives a series of LEDs as a volume unit (VU) meter, which gives the intensity of input sound.

5.2. Uncertainties

One main uncertainty is considered in the progress of FFT. During the process of FFT analysis, higher harmonic and aliasing would make the results of pitch detection unacceptable.

5.3. Ethical consideration

Since it's a sound visualization model, there will be some music pieces and even an entire song envolved to test the functionality. It's necessary to follow the IEEE Code of Ethics, #6: "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others" [5]. To respect musician's work, music that has copyright will not be spreaded and will not be saved for commercial purpose neither.

In order to test human sound as input, some people with different voice will be invited to Help and we will record them sing and talk. Therefore, according to ACM code of Ethics and Professional Conduct, #1.7: "Respect the privacy of others." [6], we will always ask if they are willing to share their recording with us, and if it's good to use their recording during our demo. If any one of them says that it's not good to use his/her voice, we will remove his/her recording from the memory.

The main safety concern of this project is the DC power supply. It’s mentioned in the DC Power Electrical Safety Guidelines: “Ensure that the polarity of the DC input wiring is correct. Under certain conditions, connections with reversed polarity might trip the primary circuit breaker or damage the equipment.” [7] We will carefully mark the polarity before turning on the switch. Furthermore, according to the article, What's the Difference Between AC and DC Electric Shocks: “DC current will make a single continuous contraction of the muscles, and can cause fibrillation of the heart at high enough levels.” [5] We will always check if the switch is turned off before we debug our circuit. So that we won’t get injured by the power.

5.4. Future work

Due to limited time, final design is not perfect as we expect. More research and tests need to be done in order to have the perfect one. Hence, in the future work, autocorrelation algorithm will be used instead of FFT to avoid more noise factors . To increase the accuracy of pitch detection, energy of frequency components is considered in the verification section. Furthermore, since current design is restricted to monotone as sound input, it is necessary to increase the music input compatibility to expand users’ market. For instance, sound of piano, human and guitar sound could be recognized in the future design. Additionally, music rhythm will be accomplished in order to make more professional music scores for users.

6. Reference

[1] AppCrawlr, ‘Best pitch detection apps for ios (Top 100)’, [Online]. Available: http://appcrawlr.com/ios-apps/best-apps-pitch-detection [Accessed: 20- Feb- 2018]

[2] Wikipedia, ‘Piano Key Frequencies’, 2018, [Online]. Available: https://en.wikipedia.org/wiki/Piano_key_frequencies [Accessed: 22- Feb- 2018]

[3] Open Audio, ‘Benchmarking - FFT Speed’, 2016. [Online]. Available: http://openaudio.blogspot.com/2016/09/benchmarking-fft-speed.html [Accessed: 22- Feb- 2018]

[4] Life.Augmented, ‘System Workbench for STM32: free IDE on Windows, Linux and OS X’, 2018. [Online]. Available: http://www.st.com/en/development-tools/sw4stm32.html [Accessed: 20- Feb- 2018]

[5] IEEE, ‘IEEE Code of Ethics’, Section 7. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 20- Feb- 2018]

[6] ACM, ‘ACM code of Ethics and Professional Conduct’, 1992. [Online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct#sect4 [Accessed: 22- Feb- 2018]

[7] Juniper Networks, ‘DC Power Electrical Safety Guidelines’, 2015. [Online]. Available: https://www.juniper.net/documentation/en_US/release-independent/jsa/topics/reference/safety/dc-power-jsa-electrical-safety-guidelines.html [Accessed: 22- Feb- 2018]

[8] Bright Hub Engineering, ‘What's the Difference Between AC and DC Electric Shocks’, 2015. [Online]. Available:

http://www.brighthubengineering.com/power-plants/89792-ac-and-dc-shock- comparison/
[Accessed: 20- Feb- 2018]

[9] Carmine Noviello, ‘Build STM32 applications with Eclipse, GCC and STM32Cube’ 2015. [Online]. Available:
https://www.carminenoviello.com/2015/06/04/stm32-applications-eclipse-gcc-stcube/
[Accessed: 20- Feb- 2018]

[10] Altera, ‘Cyclone III Device Handbook’. [Online]. Available:
https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct#sect4

[Accessed: 22- Mar- 2018]

[4] STM, ‘ARM®-based 32-bit MCU, 150DMIPs, up to 1 MB Flash/128+4KB RAM, USB OTG HS/FS….’. [Online]. Available:
http://www.st.com/content/ccc/resource/technical/document/datasheet/bc/21/42/43/b0/f3/4d/d3/CD00237391.pdf/files/CD00237391.pdf/jcr:content/translations/en.CD00237391.pdf

[Accessed: 22- Mar- 2018]

Appendix A Requirement and Verification Table

	Requirement	Verification	Verification Status (Y or N)
Microphone	Microphone module outputs voltage between 0 to 3.3v. Microphone is sensitive to input sound. The microphone must have a output for inputs sound of intensity 40dB. The output of microphone must have a signal to noise ratio of at least 10db.	Connect the output of microphone module directly to the oscilloscope, read the output voltage, and ensure is output is between 0 to 3.3v. Connect the microphone to MCU though the GPIO. Use MCU to read data from microphone. Record without any input sound, record data as noise. Record input sound, record data as noise plus signal. The input sound is at 50 db. Calculate the SNR. The SNR should be at least 10 db.	Y
LEDs	The LED has noticeable light at input voltage 3.3v. Which is provided FPGA and MCU. One of the LED should be red, one of the led should be green.	Assemble the FPGA on PCB, connect all the required capacitors and resistors. Hard code FPGA, set the output GPIO pins to logic 1, to control LED blinking.	Y

VGA	Video signal should output with VGA at resolution 640 * 480.	Assemble the FPGA on PCB, connect all the required capacitors and resistors. Connect the VGA DAC to FPGA. Hardcoding testing image in to FPGA, test the output of VGA port. Connect a display through VGA port. The display should have test image on it.	Y
MCU	Sampling sound data with A/D module at sampling frequency 44.1kHz. Perform pitch detection, the delay of result is less than 250 ms, and the rate of sample possessed is more than 10k sample/sec.	Assemble the MCU on PCB, connect all the required capacitors and resistors. Connect the Wave generator to MCU through audio jack. MCU read data with its A/D module. Set the wave generator at 22kHz. Output the data through serial port, performing FFT with matlab. Ensure the highest peak is at 22kHz. Connect the GPIO pins of MCU to wave generator. Read input data from wave generator. Perform the pitch detection algorithm on the collected data. Communicate with MCU by serial port to ensure the result of pitch detection is correct. Record the time takes to perform pitch detection with build in time lib.	N

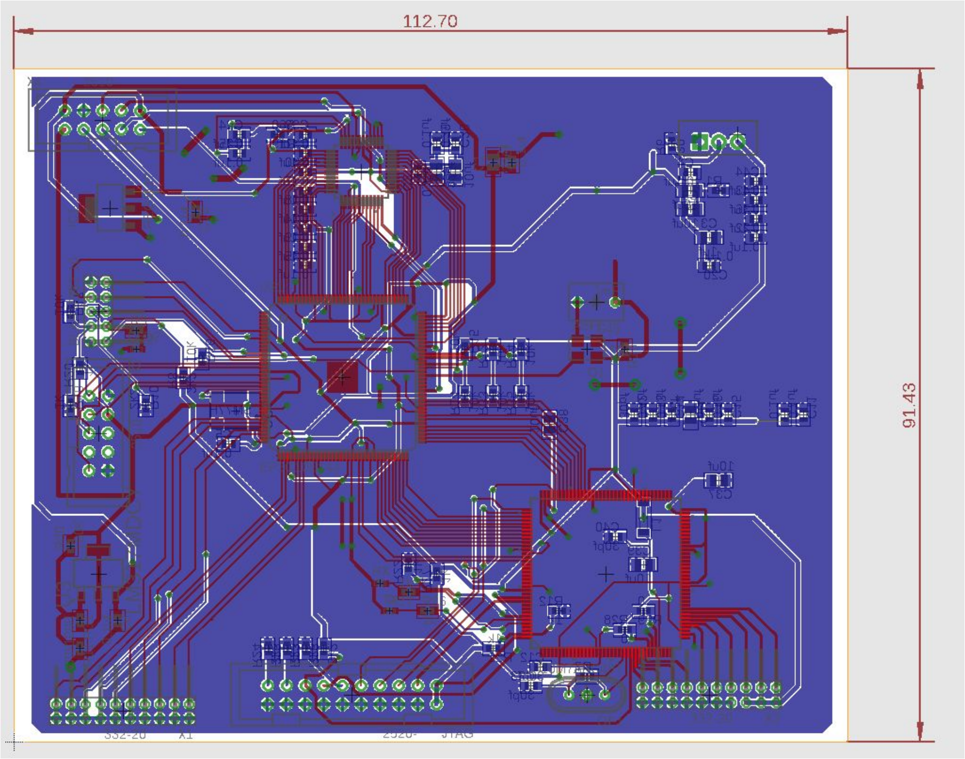
		Ensure the time is less than 250 millisecond.	
FPGA	Output the correct video and display it on VGA display. Read the data from MCU, decode the information. Blinks all the light effect LEDs correctly.	Assemble the FPGA on PCB, connect all the required capacitors and resistors. The FPGA require its dedicated Clock input. Check and make sure the FPGA is running at a desired clock speed by a oscilloscope Hard code a test image by quartus II. Upload the synthesised file. Ensure the output VGA signal is 640 * 480 at 60Hz frame per second Connect the GPIO pins of FPGA to GPIO pins of MCU. Read data to the GPIO pins by MCU. Read data from pins by FPGA. Then display the data through LEDs. Ensure the LED connected to pins of FPGA can be controlled with MCU through FPGA. The LEDs are not directly driven by FPGA, in that case the power consumption and current is too big for FPGA to handle. The pin of FPGA	Y

		is connected to a LED driver, then the LED driver drives LED. Ensure the LEDs blinks, and the intensity of LEDs are easily recognized by human.	
IDE	Compiles code in C for MCU. Synthesis systemverilog for FPGA.	Upload compiled program to STM32, communicate through serial port. Make sure the status LED of STM32 blinks. Upload synthesised file to FPGA, control LED and VGA. Read data from STM32, make sure the status LED of FPGA blinks.	Y
Power Supply	The power supply need to have output of 1.2v +/- 5%. 3.3V +/- 5%, 5V +/- 5%. The 1.2 volt power supply has the maximum output current at 0 to 500 mA. The 2.5v has a maximum current output of 1A. The 3.3v has a current output up to 800 mA. The 2.5v power supply has a maximum current output of 800mA. Maintain the temperature under 100C. The maximum operating	Mount all components of power supply to PCB. Connect all the required resistors and capacitors. Making sure the output reading is correct. Connect the output on the PCB to resistors. Read voltage across output and output current with multimeter. Ensure the power supplies meat all the requirements. Using infrared thermometer to read temperature. The temperature underload should be under 100 C.	Y

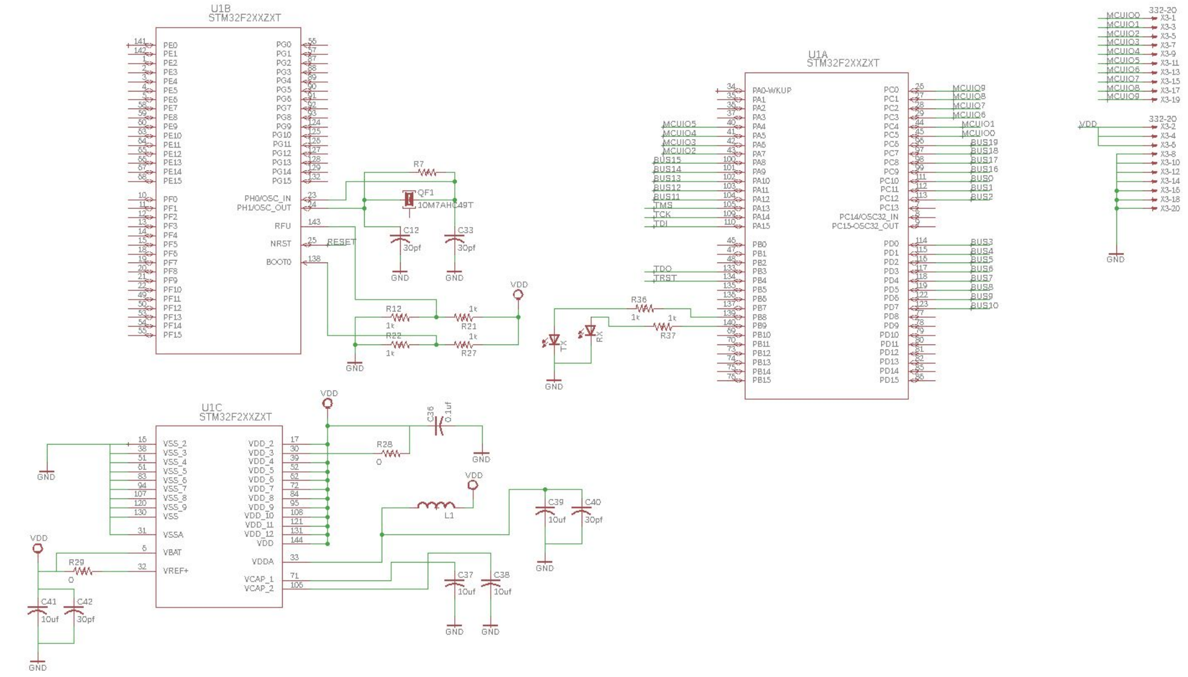
	temperature for voltage regulator	Otherwise add heatsink to the voltage regulator.	

Appendix B PCB Schematic and PCB Design

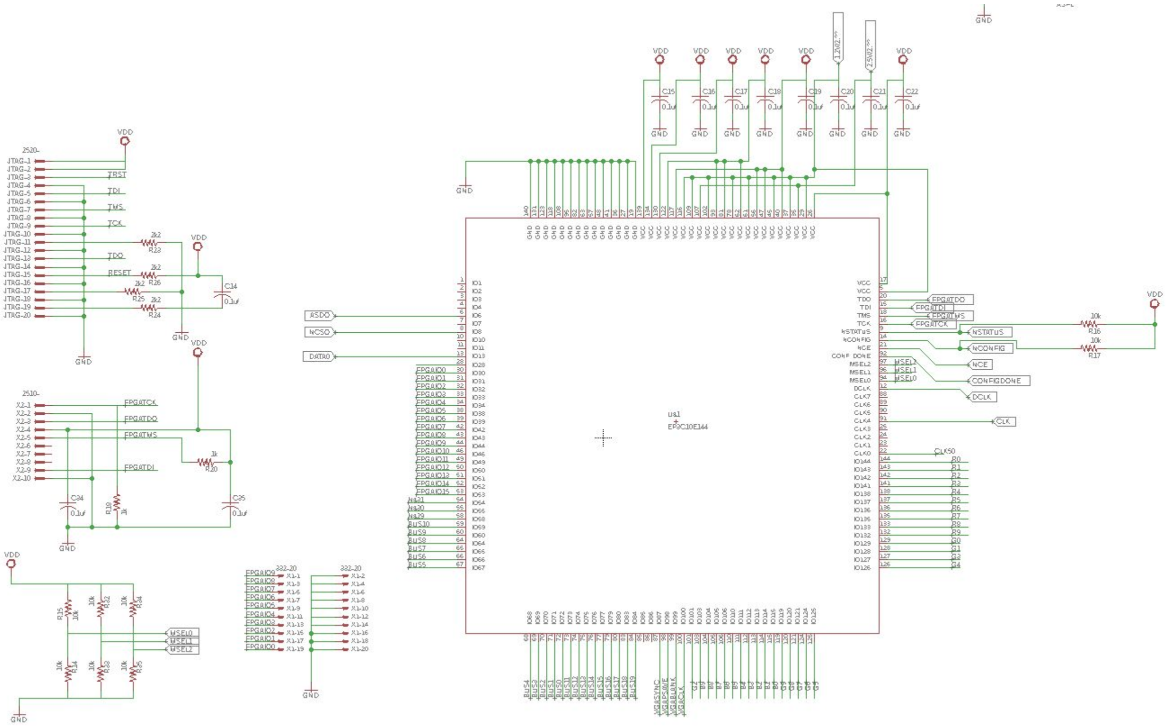
PCB design



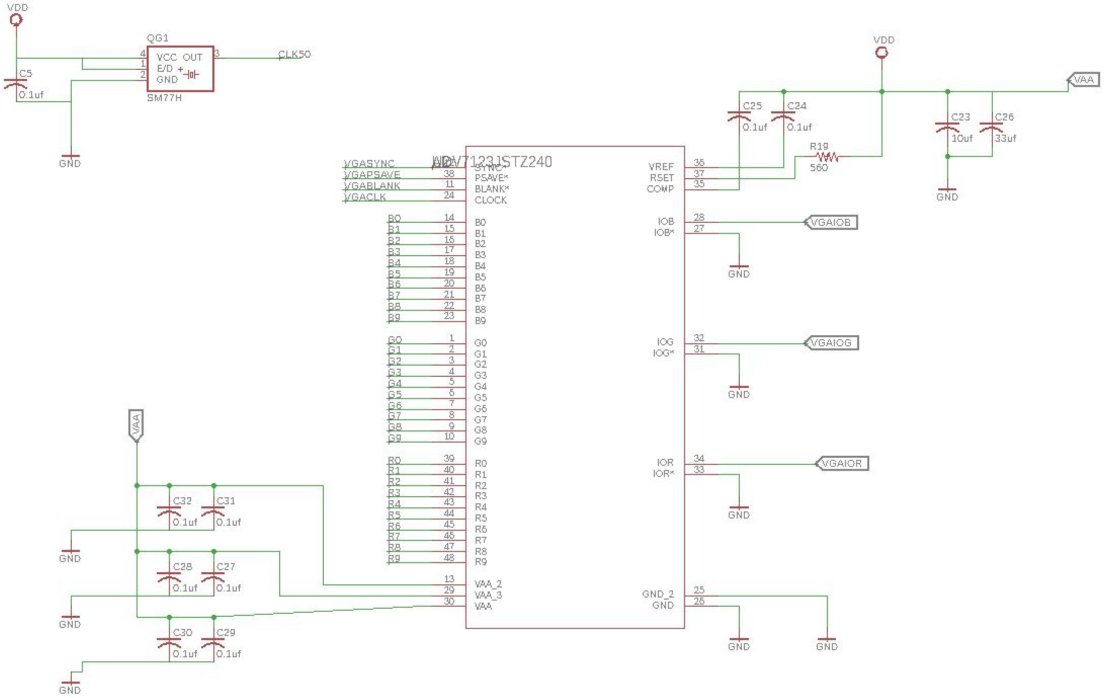
MCU Schematic



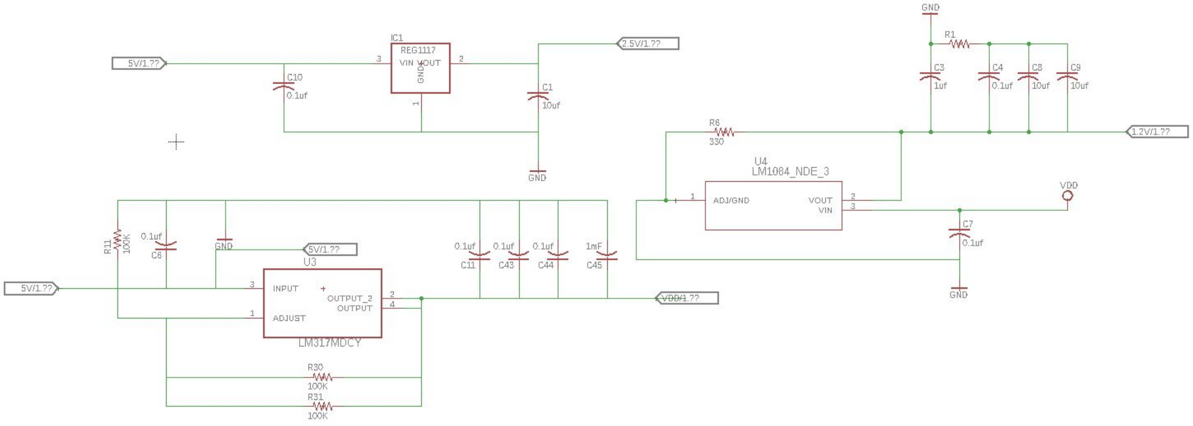
FPGA Schematic



VGA DAC Schematic



Power Supply Schematic


5V/1.??
IC1
REG1117
VIN VOUT
GND
2.5V/1.??
C10
0.1uf
C1
10uf
GND
GND
R1
C3
1uf
C4
0.1uf
C8
10uf
C9
10uf
R6
330
1.2V/1.??
U4
LM1084_NDE_3
ADJ/GND
VOUT
VIN
VDD
C7
0.1uf
GND
R11
100K
0.1uf
C6
GND
5V/1.??
U3
INPUT
OUTPUT_2
OUTPUT
ADJUST
LM317MDCY
5V/1.??
0.1uf
C11
0.1uf
C43
0.1uf
C44
1mF
C45
VDD/1.??
R30
100K
R31
100K