Music Synthesizer Design with

Field-Programmable Analog Arrays (FPAAs)

ECE 4011 Senior Design Project

Electronics for Music Applications

Dr. Jennifer Hasler

Chris Walds, EE, cwalds3@gatech.edu

Harrison Zhang, EE, hzhang462@gatech.edu

Jongheon Park, CmpE, jpark802@gatech.edu

Justin Kelley, CmpE, jkelley1@gatech.edu

Kristyn DiGiovanni, CmpE, kdigiovanni3@gatech.edu

Sriram Pulavarty, EE, spulavarty3@gatech.edu

Yewon Kim, EE, <u>ykim728@gatech.edu</u>

Submitted

November 18th, 2019

Table of Contents

Ex	ecuti	ve Summary (5 pts)	3
1.	Intr	roduction (10 pts)	5
	1.1	Objective	5
	1.2	Motivation	5
	1.3	Background	6
2.	Pro	ject Description and Goals (15 pts)	7
3.	Tec	hnical Specification (15 pts)	9
4.	Des	ign Approach and Details (20 pts)	
	4.1	Design Approach	9
	4.2	Codes and Standards	19
	4.3	Constraints, Alternatives, and Trade-Offs	20
	4.4	Engineering Analyses and Experiment	22
5.	Sch	edule, Tasks, and Milestones (10 pts)	23
6.	Pro	ject Demonstration (5 pts)	23
7.	Cos	t Analysis (5 pts)	24
8.	Cur	rrent Status (5 pts)	28
9.	Lea	dership Roles (5 pts)	28
10	. Ref	erences (5 pts)	30
Ap	opend	lix A	33
Ap	openc	lix B	34
		Appendix C	35

Executive Summary

Analog audio synthesizers are the dominant tools of music production. However, these devices come with downsides, such as a lack of portability, low-programmability, and high-power dissipation. Usage of a Field-Programmable Analog Array (FPAA) in a synthesizer design can help to improve on those drawbacks. The Electronics for Music Applications Team will do just that: design an ultra-low-power, real-time analog music synthesizer using the FPAA that takes in a user inputted MIDI signal and external hardware controls to produce analog audio outputs. The system will consist of a MIDI Shield board, a proprietary FPAA board, external hardware control devices, and a portable speaker. The data conversion board functions to ensure that the transmission of the MIDI data to the FPAA is valid and not distorted. Those signals, combined with external hardware inputs that will enable programmability in warping and distorting the final audio, will be read into the FPAA, which will be programmed with four major blocks to synthesize the audio output to the speaker. Those blocks are a Voltage-Controlled Oscillator (VCO) which generates the initial signal, a Voltage-Controlled Amplifier (VCA) which creates an amplitude envelope for the signal, a Low-Frequency Oscillator (LFO) which controls the audio output frequency, and a Voltage-Controlled Filter (VCF) which determines which frequencies to remove in order to minimize distortion. The outcome of this project is a portable analog synthesizer prototype that will produce clear audio signals that compare to those produced from common market devices but with the benefits of fitting into a small, portable package and low power utilization. Many high-end synthesizers cost more than \$2,000. However, with the usage of the FPAA, this synthesizer design is projected to cost approximately \$1,600.

Nomenclature

- ADSR Attack, Decay, Sustain, and Release
- FPAA Field-Programmable Analog Array
- LFO Low-Frequency Oscillator
- MIDI Musical Instrument Digital Interface
- PLD Programmable Logic Device
- VCA Voltage-Controlled Amplifier
- VCF Voltage-Controlled Filter
- VCO Voltage-Controlled Oscillator

Music Synthesizer Design w/ FPAAs

1. Introduction

The Electronics for Music Applications Team will build an ultra-low power, real-time analog music synthesizer using the Field Programmable Analog Array (FPAA). The team is requesting \$108 to develop a working prototype of this system and its peripherals.

1.1 Objective

This team will design and build a working prototype of an analog music synthesizer using the FPAA. This circuit would be able to receive many configurations of bias currents that would precisely adjust the character of a given note to produce a wide variety of sounds as audio output. A peripheral device or even an Android application will produce MIDI data as an input, which will be converted to an analog input signal and fed to the synthesizer circuit on the FPAA. This circuit will be tuned in real-time using external hardware and the FPAA's input/output capabilities, and the resulting synthesized signal will be routed to a speaker which will play the sound.

1.2 Motivation

Although synthesizers have existed for a while in many styles and forms, analog music synthesizers are preferred by many due to their ability to produce subtle distortions and variations in waveform shape, frequency, and amplitude [1]. Analog synthesizers, however, are often expensive, and sacrifice low-level programmability for predefined user configurations [2]. The FPAA provides an opportunity to eliminate these downsides while still producing an excellent sound quality that is standard in analog music synthesizers. The FPAA's programmability allows for the testing of many

different circuit configurations in a cheap and efficient manner, and its ultra-low power consumption will allow the FPAA to perform all functions of a music synthesizer at a much higher energy efficiency [3]. The FPAA will allow for the development of a custom-built synthesizer with operation standards superior in many aspects to music synthesizers in the modern industry. Most high-end commercial synthesizers sell for around \$2,000, but this system can be commercially developed for \$1,600. This project would most benefit musicians interested in the fields of music technology and signal processing, as this product would allow for the tuning of virtually every aspect of the synthesized sound. The synthesizer's basic components could be tweaked by musical experimentalists to produce many new kinds of sounds.

1.3 Background

Research into analog computation has significantly increased in recent years, and one device that utilizes this technology is the FPAA. The FPAA is a system-on-chip that integrates analog and digital configurable components into a single fabric controlled by a 16-bit MSP430 microprocessor. It allows for operation at up to 1000x the energy efficiency of its digital counterparts, and applications for the device range from signal processing to neuromorphic computation [3]. A large number of input/output pins are also available for interfacing with peripherals.

An open-source toolset named RASP Tools has been developed for compiling and programming circuit designs for the FPAA using configurable block diagrams. This tool allows a user to abstract certain circuit elements and build custom subunits for operation on the device. RASP Tools comes with a number of analog and mixed-signal components in its default library [4].

Investigation into subtractive music synthesis has already begun with the FPAA, and preliminary testing has been performed on a number of synthesizer components implemented on the

FPAA. In particular, a voltage-controlled oscillator, amplifier, and filter have all shown promising operation. While testing of music synthesis on the FPAA has only been conducted to show proof-of-concept, this project will realize a full implementation of the synthesizer, complete with hardware controls and an audio output [5].

There are a wide array of printed circuit boards which can act as reliable MIDI shields that would provide desired electrical protection. One such product is SparkFun's DEV-12898 ROHS MIDI Shield, which provides MIDI-IN and MIDI-OUT ports in addition to push buttons and potentiometers for tuning certain parameters [6]. This MIDI shield acts as an ideal circuit for providing input and output ports and electrical protection.

2. Project Description, Customer Requirements, and Goals

The goal of this project is to design a monophonic subtractive music synthesizer with a compact form factor. The system consists of a synthesis engine and a microcontroller. The synthesis engine is the circuit that produces the sound and will be designed by the team and be implemented on a FPAA board using the FPAA programming environment. The microcontroller processes user inputs from potentiometers and an external MIDI device. The synthesis engine consists of a VCO, VCF, VCA, ADSR envelope generator, and LFO. The VCO is a voltage controlled oscillator that produces sine, square, and triangle waveforms that will be modified by the rest of the components. The VCA is a voltage controlled amplifier that creates the envelope of the internal signal to control volume and change certain sound characteristics. The VCF is a voltage controlled low-pass filter that filters the waveform generated by the VCA to eliminate undesired frequencies. The ADSR envelope generator controls the volume of the sound as an input to the VCA to mimic the sound characteristic of natural instruments. The LFO is a low frequency oscillator that is assigned to modulate the cutoff frequency of the filter, which can create interesting musical effects such as a "wah" sounding effect. Once the

internal signal passes through the components of the synthesis engine, the output waveform is played

by a speaker.



Figure 3-1. QFD Diagram for the design of a FPAA synthesizer.

3. Technical Specifications

Important specifications from all major components of the project are shown below.

Many specifications of this project fall in line with the product specifications. However, there are

certain standards that are intended for the synthesizer system.

Table 1. Produc	et Specifications	
Synthesizer S	pecifications	
Feature	Specification	
Input-to-Sound Latency Time	0.2 seconds	
*Add something		
FPAA Spe	cifications	
Processor Speed	55 ns (Cycle Period)	
SRAM Size	16k x16 (Program), 16k x 16 (Data)	
Power (Active)	300 mW	
Voltage	3.3 V (+/- 0.3 V)	
MIDI Shield Specifications		
Voltage	5 V	
Dimensions	4" x 3" x 0.1"	
Number of Tuning Inputs	2	

4. Design Approach and Details 4.1 Design Approach *System Overview*

The portable music synthesizer system will consist of a MIDI shield circutary, a proprietary FPAA board, and a portable speaker. Figure 4 -1 shows a block diagram of the music synthesizer system.



Figure 4-1. Component-level block diagram for the music synthesizer system.

The music synthesis process begins with a peripheral device that generates MIDI data, such as a MIDI keyboard. This device sends MIDI data to the MIDI shield board, which shields the data from voltage aberrations and provides a stable interface for it to be sent to the FPAA board via a UART connection. The data are then synthesized into audio signals with the FPAA board, which houses its own microprocessor and is programmed with the music synthesizer's digital and analog circuitry. Input control signals tuned by potentiometers are also sent to the FPAA board to set certain electrical parameters, such as control voltages (CVs), which can vary the synthesizer's sound. The FPAA board's output audio signals are then played on the portable speaker. An in-depth discussion of each component is provided below.

Peripheral Devices and Android Application

The peripheral devices for the music synthesizer system are devices that can generate the MIDI data used for music synthesis. Any peripheral device that generates MIDI data is an allowed input for the system. These devices include MIDI keyboards, MIDI guitar controllers, and other electronic instruments because they can connect directly to the MIDI Shield board. However, it is currently undetermined if smartphones and tablets will also be allowed input devices for the music synthesizer system. An Android application that interfaces with the FPAA board via a USB interface has been proposed, but the application's functionality and graphical user interface have yet to be decided.

MIDI Shield Board

The MIDI Shield board is a printed circuit board that prevents ground loops and subsequent data errors by separating the transmitter circuitry and receiving circuitry with an opto-isolator [7]. It connects a MIDI In port to a serial UART output port. It also contains connections between a UART input and a MIDI Out port and connections between the MIDI In port and a MIDI Thru port. Figure 4-2 shows a schematic of the MIDI In port's connections, while Figure 4-3 shows selection ports that allow MIDI In data to be transmitted serially.



Figure 4-2. A schematic of the MIDI Shield's connections to the MIDI In port. The shield is connected to the UART port on the microcontroller [6].



Figure 4-3. A schematic of the MIDI Shield's serial selection ports. Port-Rx, which contains the MIDI Shield's desired output, can be selected by changing SJ1 [6].

In the music synthesizer system, the MIDI Shield board will be used to convert MIDI data from a peripheral device to a format that can easily be interfaced with the FPAA board's GPIO pins. This conversion can be done by connecting the peripheral device to the MIDI Shield board's MIDI In port and sending UART data through the board's Rx pin to the FPAA board. This data can then be processed and synthesized into audio signals using the FPAA's on-board microprocessor and programmed analog and digital circuitry.

FPAA Board

The FPAA board consists of a "fabric" of programmable digital and analog blocks, a 16-bit MSP430 open core processor, a register file, a series of SPI ports and GPIO pins, and a programming (DAC and ADC) infrastructure [3]. Figure 4-4 shows the layout of the FPAA SoC, while

Figure 4-5 provides a photo of the FPAA die.



Figure 4-4. Layout of the FPAA SoC, including digital and analog block array, MSP430

open core processor, register file, SPI ports, GPIO pins, DACs, and ADCs [3].

16k x 16 SRAM	NERAR BARSARA	Sc
μP	FPAA Fabric	C FPAA
Prog		-

Figure 4-5. Die photo of the FPAA SoC, which measures 12 mm \times 7 mm and is manufactured on a 350-nm CMOS process [3].

Music synthesis in the FPAA begins with the on-board microprocessor, which processes the MIDI data received from the MIDI Shield board and uses DACs to generate equivalent analog signals for the FPAA's programmable blocks. From there, the signals are used in three main voltage controlled components to generate audio signals via subtractive synthesis: the voltage controlled oscillator (VCO), the voltage controlled amplifier (VCA), and the voltage controlled filter (VCF) [5]. Figure 4-6 shows how these three components interact with each other in addition to their inputs and outputs.



Figure 4-6. Block diagram of subtractive synthesis with a VCO, VCA, and VCF. An ADSR (attack-decay-sustain-release) Generator and an LFO (low frequency oscillator) are used to provide CVs (control voltages) for the VCA and VCF, respectively.

In the VCO, a control voltage (CV) is taken in to produce an output analog waveform of a desired frequency. This waveform serves as the basis of the overall sound wave that is ultimately sent to the speakers [5]. Figure 4-7 shows how a VCO can be created from operational amplifiers and current starved inverters. Figure 4-8 shows the transistor layout for the VCO's current-starved inverters, which are slew-rate limited inverters that can be used to convert square waves to triangle waves.



Figure 4-7. Component-level schematic of a VCO made from CSIs (current starved inverters) and operational amplifiers. The reference voltages control the high and low voltages of internal square waves, while the CSIs' bias voltages and the value of the capacitor determine the slopes of internal triangle waves so that the frequency of the output waveform can be controlled [5].



Figure 4-8. Transistor layout of a CSI (current starved inverter). An additional PMOS is added to the pull up network and an additional NMOS is added to the pull down network so that the output current and, therefore, the slew rate of the output may be limited through the application of bias voltages V_{bp} and V_{bn} . This allows input square waves to be converted to triangle waves [5].

In the VCA, the VCO's output waveform and an envelope waveform are taken in to produce an amplified signal with a new envelope. Various types of envelopes can be chosen to produce different kinds of sounds, but one common envelope waveform is called ADSR (attack-decay-sustain-release). The ADSR waveform consists of a rapid rise from an initial voltage level (attack); a shorter, rapid fall

(decay); a constant voltage value (sustain); and a rapid fall back to the initial voltage level (release). It is notable for giving notes a "plucked string" sound [5]. Figure 4-9 shows how a VCA can be implemented using a Gilbert Multiplier, a circuit whose output is proportional to the product of its input voltages. Figure 4-10 shows the transistor layout of an ADSR Generator, which provides the input envelope for the VCA.



Figure 4-9. Transistor layout of a VCA made from a Gilbert Multiplier. The bias voltages V_{bn} and V_{bp} can be used to control the magnitude of the output voltage [5].



Figure 4-10. Transistor layout of an ADSR Generator made from a modified CSI. An additional NMOS is added to the pull down network to cut off current completely during the "sustain" phase of the signal. V_{rt} and V_{ft} can be used to control the rise and fall of the signal [5].

Finally, in the VCF, the VCA's output waveform and a cutoff frequency CV are taken in to produce a filtered output waveform without undesired frequency components. The waveform can be sent through several different VCF blocks in order to achieve the desired mixture of frequencies. The cutoff frequency CVs of different VCF blocks may be modulated by inputting sine waves instead of DC voltage values, allowing the synthesizer to produce a "wah wah" sound that can be similar to a glissando, or - if the cutoff frequency is modulated quickly enough - even vibrato. To produce these input sine waves, it is possible to include an LFO (low frequency oscillator) block (implemented the same way as a VFO) to produce low frequency signals [5]. Figure 4-11 shows how a VCF can be implemented with a transistor ladder filter.



Figure 4-11. A VCF implemented as a transistor ladder. The gain constants can be created by using operational amplifiers and performing sweeps of their bias currents. The cutoff and bias voltages of the circuit can be varied to subtract different frequencies from the input and to change the frequency response [5].

Once the output audio signals exit the VCF block, they can be sent in analog form to the portable speaker, which outputs the synthesized audio. This audio should match the notes represented by the peripheral device's MIDI data at the beginning of the synthesis process.

FPAA Programming Environment

While the FPAA programming environment is not an explicit component of our music synthesizer system, it is still crucial, for it allows the VCO, VCA, and VCF circuitry described in the previous section to be implemented. The FPAA programming environment is called RASP, and it consists of many layers. The user interacts directly with a high end graphical design environment built in Xcos that compiles into distinct analog, digital, and assembly components. The analog and digital components are converted into netlists with blif (Berkeley Library Interface), which are then compiled to a switch list for the FPAA's program unit with an open source tool called x2c. Meanwhile, the assembly component of the software is compiled into hex code, where it is also delivered to the FPAA's program unit [4]. Figure 4-12 shows a block diagram of the programming interface's different layers.

Figure 4-12. A block diagram of the different layers of the FPAA board's programming interface. The programming environment consists of a palette of graphical design blocks, which compile to lower analog, digital, and assembly software levels from the Xcos environment [4].

To program the FPAA board, a user simply drags pre-made graphical design blocks from a palette onto a blank design window. The premade design blocks range in complexity from simple PFET blocks to entire analog circuits, such as operational amplifiers. From the design window, the user can press a button to drag wires between the input and output ports of different design blocks before compiling the circuitry to the FPAA board. Programmed circuitry can be reloaded for subsequent uses of the FPAA board so that the locations of important cells do not change.

4.2 Codes and Standards

- 1. Universal serial bus (USB) is utilized to program the FPAA microcontroller. USB is also used in serial communication from the Android application to the microcontroller. USB features:
 - 480 Mbps data transfer rate.
 - Versatility in peripheral connections [7].
- 2. Universal Asynchronous Receiver Transmitter (UART) is a communication interface that serves to convert incoming and outgoing data into a serialized bit stream [8]. UART is utilized in the output of the MIDI shield as the input to the FPAA.
- 3. General Purpose Input/Output (GPIO) is a communication protocol used to connect the FPAA microcontroller with additional peripherals. GPIO will be used to provide the converted input to the FPAA from the MIDI shield [9].
- 4. Musical Instrument Digital Interface (MIDI) is a communication standard among electronic instruments, computers, and audio devices that contain information involving a note's notation, pitch, and velocity [10]. MIDI will be used as the initial input for the synthesizer.

5. The FPAA board also features a 3.5 mm headphone jack, which will be routed to a speaker in the final audio output [3].

4.3 Constraints, Alternatives, and Trade-Offs

Alternatives

The main alternative to an analog music synthesizer system is a digital synthesizer system. Digital synthesizers are often portable and easy to interface with other devices. However, analog music synthesizers are often considered to produce more natural, aurally pleasing sounds than their digital counterparts, for they have a tendency to produce distortion as well as subtle variations in waveform shape, frequency, and amplitude [1]. As a result, an FPAA board implementation of a music synthesizer seems to strike the best balance between portability and sound quality.

Constraints

In order for the music synthesizer system to be effective, it must be portable, energy-efficient, and fast. These constraints will enable the system to provide unique benefits to users who are unsatisfied with the large but rich-sounding analog synthesizers that are on the market today. Additionally, they will allow the synthesizer the possibility of interfacing with portable devices such as smartphones or tablets, which is a potential goal. As a result, the synthesizer system should be compact enough to fit on a device that connects to a smartphone via a micro-USB or USB-C connection. This goal can be achieved by combining the FPAA board, portable speaker, MIDI Shield board, and potentially a USB interface all onto a single package. To be consistent with the potential goal of interfacing with smartphones or tablets, this combined package should be able to draw power from the USB interface and be energy efficient enough to rely on a smartphone's or tablet's battery for an appreciable period of time.

Trade-Offs

Energy efficiency may trade-off with the performance of the music synthesizer, in terms of both its speed and its sound quality. Lower energy targets may force the FPAA board's microprocessor to operate less efficiently, resulting in slower conversion of the MIDI Shield board's output data to analog signals. Perhaps more significantly, they may also limit the maximum amplitudes allowed for the waveforms generated by the synthesizer, providing a bottleneck on the volume of any output audio. Overall, balancing energy efficiency with the desire for high performance may require sacrifices in the feature set of the music synthesizer system, especially if an Android application is eventually included in the design.

4.4 Engineering Analyses and Experiment

The majority of experimentation with synthesizers can simply be carried out by feeding the hardware a MIDI input and listening to the output, tuning certain circuit parameters on the MIDI shield as seen fit. The final product's working state can be easily assessed in this manner, and individual circuit parameters can be fine-tuned according to the observed results. However, individual synthesizer components on the FPAA will need to be tested before this stage of high-level experimentation is realized.

The individual components built on the FPAA can be tested through its various GPIO pins. One such device that interfaces with these pins is the Analog Discovery 2. This product, which communicates directly with the software *Waveforms* on a computer, is an all-in-one USB oscilloscope and instrumentation system which can generate input waveforms for the FPAA and read resulting output waveforms [11]. This device allows for quantitative and efficient testing of each circuit component. For example, testing the VCO would involve feeding the unit a square waveform and observing the characteristic triangular output waveform. Similar trials can be run with the VCA and VCF, with bias currents adjusted as needed along the way. Individual hardware components, along with their communication protocols will also need to be individually tested before the final design is built. For example, the MIDI shield needs to be thoroughly verified before integration into the large-scale hardware design, which can be carried out by routing a preexisting MIDI device, such as a keyboard, to a MIDI-out speaker. The Android application's external communication can be verified through various example programs.

5. **Project Demonstration**

At a minimum, the project demonstration will incorporate a MIDI keyboard that provides inputs to our FPAA music synthesizer which then sends its output to a speaker so the synthesized music can be played. The FPAA will be programmed at the site (with possibly multiple variations of our music synthesizer design) prior to use. A stretch goal would include a peripheral device, like a tablet or smartphone, which would control the FPAA with an app and send inputs over USB. The goal is to demonstrate that the FPAA provides an easily configurable and portable option for music synthesis.

This demonstration will validate project specifications by highlighting the fast and simple reprogrammability of the device. Because latency will be hard to test in a demo setting, we just expect to demonstrate programmability without a noticeable delay for user interaction. Other specifications (like clock speed, voltages, etc.) will be demonstrated by correct device operation.

6. Schedule, Tasks, and Milestones

The design team will be designing, testing, and implementing the music synthesizer over the semester until the major milestones in the second week of April. Appendix A is the comprehensive Gantt chart of the project. A partial Gantt Chart is shown in Appendix B; Appendix C contains the related Pert chart outlining the expected duration for each task.

PERT Chart Critical Path

Path Time

ABEFH~P	1.17+2.33+2.33+4.67+51.16 = 61.66
ABEGH~P	1.17+2.33+1.17+4.67+51.16 = 60.5
ACEFH~P	4.17+2.33+2.33+4.67+51.16 = 64.66
ACEGH~P	4.17+2.33+1.17+4.67+51.16 = 63.5
ADEFH~P	5.17+2.33+2.33+4.67+51.16 = 65.66
ADEGH~P	5.17+2.33+1.17+4.67+51.16 = 64.5

The estimated time of the critical path is 65.66 days. This estimate is excluding holidays and weekends of the number of days between the first day of class (1/6/20) and the Expo (4/10/20). The probability of finishing one week prior to the Senior Design Capstone Expo is 99.92 %.

7. Marketing and Cost Analysis Cost Estimate (Year 1)

A. Employee Services (12 hr/week)

	a.	Electrical/Computer Engineer - 7 @ \$40/hr	\$280.00
	b.	Project Manager - 1 @ \$40/hr	\$40.00
	c.	Fringe Benefits (25% of total salary)	\$14,400
	d.	Total (1 year)	\$72,000
B.	Materi	als/Supplies	
	a.	FPAA Components (per unit)	\$94.69
	b.	Supporting Components (per unit)	\$107.01
	c.	Total (10 testing units)	\$2,017
C.	Miscel	laneous Costs	
	a.	Spare parts (per month)	\$500.00

	b. Overtime - 8 @ \$45/hr (~2 hr/week)	\$720.00
	c. Total (1 year)	\$43,440
D. Tot	al Cost	\$117,457

Employee salaries were estimated based on the average salary for an electrical/computer engineer in the Atlanta area. Hours worked per week were based on ECE4012 credit hours and allocated lab hours (3 credit class * 4 hours = 12 hours/week).

Product cost breakdown below. Miscellaneous costs based on spare parts (in case of defective components) and need for overtime as required throughout the semester.

Component	Units	Cost
TI-MSP430	1	\$5.01
(Microprocessor) [12]		
70V261L25PFG (SRAM)	1	\$44.38
[13]		
Custom Analog/Digital Pl	_ 1	~\$25.00
TMS320C5515 (SPI I/O)	1	\$5.30
[14]		
Manufacturing	N/A	~\$15.00
Supporting Components		
Component	Units	Cost

FPAA Breakdown

Standard USB 3.0 Cable	1	\$10.00
USB 3.0 OTG Cable	1	\$6.00
DEV12898 (MIDI	1	\$21.95
Shield) [15]		
LPC1768	1	\$52.06
(Microcontroller) [16]		
Speaker	1	\$12.00
Potentiometers	5	\$5.00

Estimates for the custom PLD cost were determined by researching component costs for commonly used FPGA PLD chips from popular electronic part shopping sites. The estimate for the manufacturing cost was determined by researching custom PCB costs with the sizes/parameters for the current FPAA model we intend to use.

Development Cost

Parts	\$2,017
Labor (~180 hr/semester, 8 employees)	\$57,600
Fringe Benefits (25% total salary)	\$14,400
Subtotal	\$74,017
Overhead (120% of subtotal)	\$88,820

Total	\$162,837

Determination of Selling Price

Parts Cost (1 unit)	\$201.70
Assembly Labor (Manufacturing)	\$15.00
Testing Labor	\$135.00
Total Labor	\$150.00
Fringe Benefits (25% of labor)	\$37.50
Subtotal	\$539.20
Overhead (120% of subtotal)	\$647.04
Subtotal (Input Costs)	\$1,186.24
Sales Expense (10% selling price)	\$160.00
Subtotal (All Costs)	\$1,346.24
Profit	\$253.76
Selling Price	\$1,600

Assuming a selling period of 5 years and an approximate selling capacity of 120 units per year, to amortize the development cost, each unit must be sold at: [5 * (120 * (selling price - unit cost)) = \$162,837]. The calculated unit price comes out to be about \$1,600. Each unit provides a \$253.76 profit, or 15.9% of selling price profit.

8. Current Status

The group has defined a high-level design for the project with necessary components/resources identified. Currently, the group is focused on the analog designs and schematics for each component of the music synthesizer (VCO, VCA, VCF, etc.). The goal is to have everyone gain experience with the FPAA design software and analog components so that there is a common ground/understanding for technical communication. At this point, the group has the foundation planned for the project timeline moving forward as well as research conducted throughout the semester on project components (music synthesis, USB, FPAA, etc.). Based on the amount of work completed until now, the project is about 30% complete (as we still need to implement researched components, assemble product, and test).

9. Leadership Roles

- 1. Chris Walds: Expo Coordinator
 - a. Manages content to be presented at the Capstone Expo
- 2. Harrison Zhang: Webmaster
 - a. Maintains and updates the team website and Google Drive
- 3. Jongheon Park: Design Coordinator
 - a. Oversees design progress against schedule
 - b. Works with Dr. Hasler to implement corrective measures for identified issues
- 4. Justin Kelley: Team Coordinator
 - a. Arranges team meetings with Dr. Hasler

- 5. Kristyn DiGiovanni: Software Specialist
 - a. Reviews software development for synthesizer
- 6. Sriram Pulavarty: Documentation Coordinator
 - a. Creates and updates database with detailed explanations on functionality of synthesizer and design choices made
- 7. Yewon Kim: Hardware Specialist
 - a. Assists in the installation of the synthesizer
 - b. Analyzes and adjusts any hardware reconfigurations

10. References

- M. Doty, "What's Behind The Resurgence of Analog Synthesizers?", Performer Mag, July 14, 2016. [Online]. Available: https://performermag.com/best-instruments/best-music-keyboards-synth/resurgence-of-analog-synthesizers/. [Accessed Nov. 18, 2019]
- [2] V. Lazzarini and J. Timoney, "The Analogue Computer as a Voltage-Controlled Synthesiser", Maynooth University. April 25, 2019. [Online]. Available: https://arxiv.org/pdf/1904.10763v2.pdf [Accessed Nov. 18, 2019]
- [3] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan "A Programmable and Configurable Mixed-Mode FPAA SOC," IEEE Transactions on VLSI, January 2016. [Online]. Available: http://hasler.ece.gatech.edu/SoCFPAA/RASP30_SoCFPAA.pdf [Accessed Nov. 18, 2019]
- [4] M. Collins, J. Hasler, and S. George, "An Open-Source Toolset Enabling Analog–Digital– Software Codesign," *Journal of Low Power Electronics Applications*, January 2016. [Online]. Available: http://hasler.ece.gatech.edu/Published_papers/FPAA_Papers/Highlevel_tools_ jlpea_February2016.pdf. [Accessed Nov. 18, 2019].
- [5] S. Nease. "Neural and Analog Computation on Reconfigurable Mixed-Signal Platforms", Ph.
 D. dissertation, Georgia Institute of Technology, GA, 2014. [Online]. Available: https://smartech.gatech.edu/handle/1853/53999. [Accessed Nov. 18, 2019].
- [6] C. Taylor, *SparkFun_MIDI_Shield*, Sparkfun Electronics, 2015. [Online]. Available: http://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/Midi_Shieldv15.pdf [Accessed Nov. 17, 2019]
- [7] "USB 2.0, Hi-Speed USB FAQ," Everything USB, 20-Mar-2017. [Online]. Available: https://www.everythingusb.com/hi-speed-usb.html. [Accessed: Nov. 18, 2019].

- [8] Kumar, Kumar, and D. Rajani, "UART Communication Protocol How it works?," Codrey Electronics, 26-Jul-2018. [Online]. Available: https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/. [Accessed: 18-Nov-2019].
- [9] "What is GPIO?," Estimote Community Portal. [Online]. Available: https://community.estimote.com/hc/en-us/articles/217429867-What-is-GPIO-. [Accessed: 18-Nov-2019].
- [10] C. S. Sapp, MIDI Communication Protocol. [Online]. Available: http://www.ccarh.org/courses/253/handout/midiprotocol/. [Accessed: 18-Nov-2019].
- [11] "Analog Discovery 2 Specifications," Analog Discovery 2 Specifications
 [Reference.Digilentinc]. [Online]. Available: https://reference.digilentinc.com/reference/instrumentation/analog-discovery-2/specifications.
 [Accessed: 18-Nov-2019].
- Texas Instruments. "TMS320C5515 Fixed-Point Digital Signal Processor," Texas Instruments Incorporated, 2013. [Online]. Available: https://www.ti.com/lit/ds/symlink/tms320c5515.pdf
 [Accessed: 18-Nov-2019].
- [13] Integrated Device Technology. "HIGH-SPEED 3.3V 16K x 16 DUAL-PORT STATIC RAM,"
 July 2019. [Online]. Available: https://www.idt.com/us/en/document/dst/70v26-data-sheet
 [Accessed: 18-Nov-2019].
- [14] Texas Instruments. "16-Mb RADIATION-HARDENED SRAM," Texas Instruments Incorporated, 2014. [Online]. Available: http://www.ti.com/lit/ds/symlink/smv512k32-sp.pdf
 [Accessed: 18-Nov-2019].
- [15] Byron, J. "MIDI Shield Hookup Guide," Sparkfun Electronics. [Online]. Available:

https://learn.sparkfun.com/tutorials/midi-shield-hookup-guide/all [Accessed: 18-Nov-2019].

 [16] NXP Semiconductors. "LPC1769/68/67/66/65/64/63," May 4 2014. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/LPC1769_68_67_66_65_64_63.pdf
 [Accessed: 18-Nov-2019].

					NOVEMBER	-	IANUARY		H	RUARY		-	MARCH	т	_	AP	SIL	
	TASK TITLE	START DATE	FINISH DATE	TASK LEADER	MEEK 4 MEEK 3 MEEK 5 MEEK 4	MEEK 1	MEEK 3 MEEK 5	MEEK ¢	MEEK 3	MEEK 3	MEEK ¢	NEEK J	MEEK 3	MEEK #	MEEK 1	MEEK 5	MEEK 3	WEEK 4
-	Project Initiation																	
1.1	Research Codes	11/4/19	11/19/19	0,9		<u></u>		0							a C			
1.2	Research Market & Cost	11/4/19	11/19/19	Z														
1.3	Research Component Blocks	11/4/19	11/19/19	WKS,Y	3													
1.4	Meeting w/ Advisor 1	11/7/19	11/8/19	ALL														
1.5	Design Meeting	11/13/19	11/13/19	ALL														
1.6	Project Summary	11/11/19	11/18/19	ALL														
1.7	Project Proposal	11/12/19	11/19/19	ALL														
1.8	Meeting w/ Advisor 2	11/21/19	11/22/19	ALL											_			
2	Design and Test	1																
21	Meeting w/ Advisor 3	1/16/20	1/17/20	ALL														
22	Oral Project Presentation	1/17/20	1/17/20	ALL														
23	Research	1/23/20	1/29/20	ALL														
24	Project Specification	1/29/20	1/29/20	ALL														
25	Meeting w/ Advisor 4	1/30/20	1/31/20	ALL														
2.6	Hardware Design	1/31/20	2/14/20	W,Z,S,Y														
27	Software Design	1/31/20	2/14/20	P,K,D														
2.8	Lab Testing and Debugging	2/14/20	2/21/20	ALL														
2.9	Final Project Specification	2/21/20	3/9/20	ALL														
2.10	Project Report	3/11/20	3/18/20	ALL										- 33	-			
211	Completed Website	3/23/20	4/3/20	ALL		- 3		1								_		
3	Milestones					A.												
3.1	Proposal	1/23/20	1/29/20	ALL														
3.2	Final Project Presentation	4/6/20	4/6/20	ALL														
3.3	Final Project Demo	4/6/20	4/6/20	ALL														
3.4	Final Project Deliverables	4/6/20	4/9/20	ALL														
8		22	99 98		20	8	[8			85				8			
					LEGE	Q	T											
					Completed		1											
					Remaining		Ĩ											
					w/ Chris Walds													

ECE4011/4012 COMPREHENSIVE GANTT CHART

Music Synthesizer with FPAAs

Appendix A: Comprehensive GANTT Chart

Г

Appendix B: GANTT Chart



ECE4011/4012 GANTT CHART

Music Synthesizer with FPAAs

Appendix C: PERT Chart

