Critical Design Review

Ben Davey

Nathan Sweetman

Aaron Frishling

Andreas Wulf

Contents

Contents ... 1
Introduction ... 3
Purpose ... 3
Scope ... 3
Abbreviations and Acronyms ... 3
System Overview ... 4
Background & Significance of the project ... 5
History of the synthesizer ... 5
Technical overview of traditional analogue synthesizers ... 7
Significance of the project ... 8
Project Requirements ... 10
Controllable Oscillators ... 10
Functional Requirements ... 10
Non-functional Requirements ... 11
Controllable Filter ... 11
Controllable Envelope Generator ... 12
Functional Requirements ... 12
Non-functional requirements ... 12
Mixers ... 12
MIDI control ... 13
User Interface ... 13
Functional Requirements ... 13
Non-functional Requirements ... 14
Approach ... 15
Design Overview ... 15
Module specifications and development approach ... 16
Interfacing Control System ... 16
MIDI receiver ... 16
Serial Link and Serial-control-protocol ... 16

        Master Controller 16
        Gumstix Controller API 16
    Oscillators 17
    Controllable Filter 17
    Envelope Generator 18
    User Interface 18

Project Management 19

  Milestones 19
    1. Development and Understanding of Synthesis Components 19
    2. Development of Interconnection and Control Systems 20
    3. MIDI and Graphical User Interface (GUI) 21
    4. User Acceptance Testing and General Refinement 22
  Resources provided free-of-charge 24
  Budgeted resources 26
  Unacceptable audio quality 28
  Major component failure 28
  Incomplete System upon project deadline 29
  Project does not meet specified requirements 29
  Project goes over budget 29

Appendix A – Gantt Chart 31

References 32

Introduction

Purpose

This document provides a reference for the architecture, design, and time management plan for the Hybrid Synthesizer currently in development by Synthesia. It is intended to provide an overview of the system, as well as functional specifications of its constituent components

Scope

This document intends to provide both a design *overview* for the synthesizer, as well as a *design specification*. It describes the modules that will exist in the system, an overview of their functionality, and their interactions with other components to form a complete system.

Abbreviations and Acronyms

ADSR - Attack, Decay, Sustain & Release
COTS - Commercial Off The Shelf
DAC - Digital to Analogue Convertor
FPGA - Field Programmable Gate Array
GUI - Graphical User Interface
LCD - Liquid Crystal Display
MIDI - Musical Instrument Digital Interface
VHDL - VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
I/O - Input/Output

System Overview

Broadly, the purpose of this project is to create an *Analogue modelling musical synthesizer*; that is, a synthesizer implemented using both analogue and digital components that intends to emulate the sounds of traditional analogue synthesizers, whilst being controllable through modern-day digital protocols such as the Musical Instrument Digital Interface (MIDI).

The synthesizer platform will exist upon it's own standalone hardware, and may be controlled using a variety of existing MIDI-capable instruments, including keyboards, guitars and drums. Additionally, the unit will feature it's own innovative touch-screen interface which intends to provide simple-and-intuitive control of the system, but with an unprecedented ability to customize and control synthesizer parameters and sound generation.

Special attention has been placed upon the usability of the synthesizer, particularly within the realms of it being played by musicians in a live environment. Whilst easily combined with professional studio equipment and sequencing software such as Cubase VST, the unit will also be capable of being played live with minimal additional equipment.

Background & Significance of the project

History of the synthesizer

Synthesizers have been in development since the late 1800's. The first incarnation was created somewhat by accident by telephone engineer Elisha Grey when he stumbled upon the voltage controlled oscillator and used it to produce basic single-tone notes.

Perhaps the most famous and sought-after synthesizer in history, however, is the Moog mini, developed by Robert Moog in 1970.

When the mini was invented, the majority of synthesizers in existence were of a *modular configuration*, meaning that parts could be interconnected in interesting configurations through the use of patch cables, much like electronic components on a breadboard. This approach was popular in research due to the reconfigurable nature of the systems (and also because many of the synthesis modules were “borrowed” analogue computers and laboratory test equipment). However, it also meant that the systems were large and bulky, and usually only existed in large institutions, such as music departments at universities, making them inaccessible to the general populous. Additionally, the large learning curve involved in using synthesizers such as the RCA Mark II meant that an every-day musician without an electrical engineering degree would find the device exceptionally difficult to use.

Moog essentially took the modules that he considered to be the most important foundation elements of synthesis, and placed them into a single, highly portable system, known as the "mini". The mini became the benchmark for many synthesizers that followed it, and has become a well-recognized image in both the music world and pop culture in general.

Throughout the 1980s companies like Roland and Yamaha dropped the idea of the analogue synthesizer in favour of more digitized Pulse Code Modulation (PCM) based *sampling* synths. Essentially a sampling unit replaced oscillators, and much of the contour control offered by the envelope generator was removed from the system. Unfortunately this offered less control over the sound the unit created, and people began to realise that the sounds produced were not as nice as they had anticipated -- analogue synths had been popular not because they mimicked the sounds of other instruments, but were very good at making 'stupid' noises; the machines had become instruments in their own right.

In recent years, analogue synths have seen a huge resurgence in popularity. The 'electro' sound is once again becoming popular, with musicians and audiences beginning to favour the nostalgic analogue sound.

Technical overview of traditional analogue synthesizers

Our project has been designed using a traditional analogue synthesizer as a foundation. *Figure 1* provides a high-level block diagram for the operation of an analogue synthesizer.

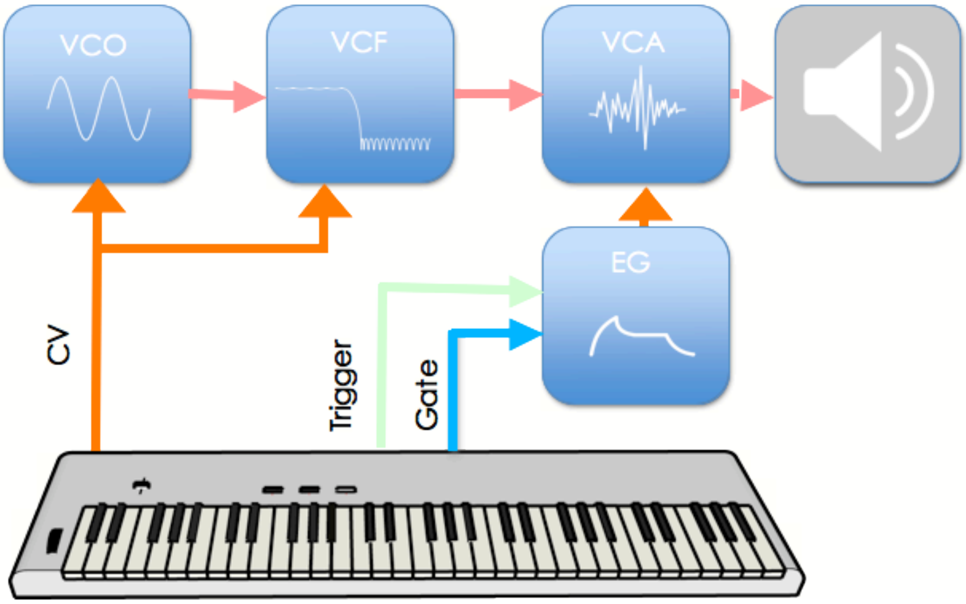


Figure 1. Traditional analogue synthesizer block diagram

A traditional analogue synthesizer consists of the following core components:

- Voltage Controlled Oscillator (VCO) – Creates an oscillating, periodic waveform (ie sine wave, square pulse train, sawtooth, etc) whose frequency is dependent upon input control voltage provided to it from the keyboard or control device

- Voltage Controlled Filter (VCF) – Filters the input waveform from the VCO to a cutoff-frequency that is dependent upon an input control voltage provided to it from the keyboard

- Envelope Generator (EG) – A module used to modulate the devices' volume over time. The device is activated by trigger and gate signals from the keyboard, which are provided when keys are pressed, changed or released on the keyboard. The contour of the envelope is often specified using the following parameters:

Attack – how quickly the sound reaches full volume after the sound is activated

Decay – how quickly the sound drops to the sustain volume after the initial peak

Sustain – the constant volume that the sound takes after decay until the note is released

Release – how quickly the sound fades away after a note is released

- Voltage Controlled Amplifier (VCA) – Mixes the contour signal generated by the EG with the filtered tones created by the VCO and VCF modules to create the final synthesized sound

In analogue designs, both the VCO and VCF are driven by Control Voltages from the keyboard, where a certain note frequency may correspond to a given voltage level.

Significance of the project

Whilst desired by many people for their distinctive and aurally appealing sound, traditional analogue synthesizers were not without their technical limitations and annoyances, particularly when compared to modern day devices. Such drawbacks include:

- **A high susceptibility to de-tuning**: Analogue synthesizers were often susceptible to whirring and sudden bends in pitch whilst being played due to fluctuation and noise in their power supplies, which lead to inconsistent control voltages which in-turn were passed to highly sensitive VCOs and filters and ultimately resulted in a less-than-desirable sound output under certain environmental conditions
- **Limited customisation options and scalability with other systems** – Traditional analogue synthesizers were commonly stand-alone units invented in a time when standardized interconnectivity protocols such as MIDI had not yet been conceived. This means that they are unable to be combined with other synthesizers and instruments, and could not be sequenced through software.

However, many traditional analogue synthesizers have the advantage that they can easily be played live, and have all of their parameter control easily within reach of the musician through the use of rotary knobs and sliders. This appears to have become a shortcoming of modern day soft synths and their host programs – Many of them have been designed for use in the studio, with over-complicated keyboard-and-mouse driven user interfaces that offer limited customization and control through tactile interfaces. This leads to clumsy or impossible live performances.

Our project will create a synthesizer that produces the highly sought-after “analogue sound”, whilst addressing the aforementioned shortcomings of traditional analogue synthesizers, and implementing modern and versatile technologies such as MIDI and a computerized user interface. This will allow for expansion and connection with other systems in a hardware package that is easy to use and control in real-time.

Project Requirements

The project deliverables have been broken down into sections detailing the individual requirements for each of the core system components.

Controllable Oscillators

Functional Requirements

ID	Description	Dependencies
CO001	Implement oscillators onto FPGA hardware using VHDL	
CO002	Project must implement a sine-wave oscillator	
CO003	Project must implement a periodic sawtooth oscillator	
CO004	Project must implement a rectangular pulse train oscillator	
CO005	Project must implement a triangular oscillator	
CO006	All oscillators must produce waveforms capable of being output as analogue signals of amplitude ± 1V	
CO007	All oscillators must produce signals of bit depth 16 bit	
CO008	All oscillators must output at sampling frequency fs ≥ 44100 Hz	

ID	Description	Dependencies
CO009	At least 3 of each oscillator must be implemented	
C0010	Oscillators must be tuneable to a user specified frequency, 22 Hz ≤ f ≤ 22050 Hz	

Non-functional Requirements

ID	Description	Dependencies
CO011	Oscillator waveform may be drawn in by using the touch-screen GUI	
CO012	Line-in may be used as a modulator input and linearly mixed with a combination of CO002-CO005 as carrier oscillators for the implementation of a vocoder	CO002 – CO005

Controllable Filter

ID	Description	Dependencies
CF001	Filter must accept an analogue input signal from the mixer output	
CF002	Filter must be capable of having a variable cut-off frequency	
CF003	Filter cut-off frequency must be digitally controllable	CF002

Controllable Envelope Generator

Functional Requirements

ID	Description	Dependencies
EG001	Envelope generator must be implemented as software on the gumstix platform	
EG002	Envelope generator must produce an output signal that can be mixed with output from the filter	CF00x
EG003	Envelope generator must produce an output contour with system specifiable input variables (both time and volume level) for Attack, Decay, Sustain, and Release	

Non-functional requirements

ID	Description	Dependencies
EG004	Envelope generator contour waveform may be drawn in using the touch-screen GUI	

Mixers

ID	Description	Dependencies
AM001	A mixer must be implemented to mix the output of at least 3 oscillators to a signal capable of being passed to the controllable filter	

ID	Description	Dependencies
AM002	A mixer must be implemented to mix the output of the controllable filter with the volume-contour-signal produced by the envelope generator	

MIDI control

ID	Description	Dependencies
MC001	The system must be capable of general MIDI input	
MC002	Control filter cut-off frequency must be controllable using MIDI-note input from a MIDI capable device (at a frequency comparable to the octave of the ‘note’ being triggered)	
MC003	Frequency of oscillators must be controllable using MIDI note input	
MC004	Parameters for oscillator and filter range, envelope ADSR values, and mixer input levels must be programmatically assignable to user specified MIDI controls (eg. rotary knobs on a keyboard controller)	

User Interface

Functional Requirements

ID	Description	Dependencies
UI001	User Interface software must be implemented onto the gumstix platform	

ID	Description	Dependencies
UI002	User Interface must be controllable using touch input gestures	
UI003	User Interface must provide ability for user to alter all controllable parameter values	

Non-functional Requirements

ID	Description	Dependencies
UI004	User interface may implement a graphical arpeggiator with user specifiable parameters	
UI005	User interface may implement a graphical pattern-based step sequencer capable of sending MIDI-note output to the audio-generation modules of the device	

Approach

Design Overview

The complete system consists of a number of modules that will be developed independently. Upon completion, they will be connected together and will be controllable using the Master Controller. Figure 2 illustrates an architectural abstraction of the complete device.

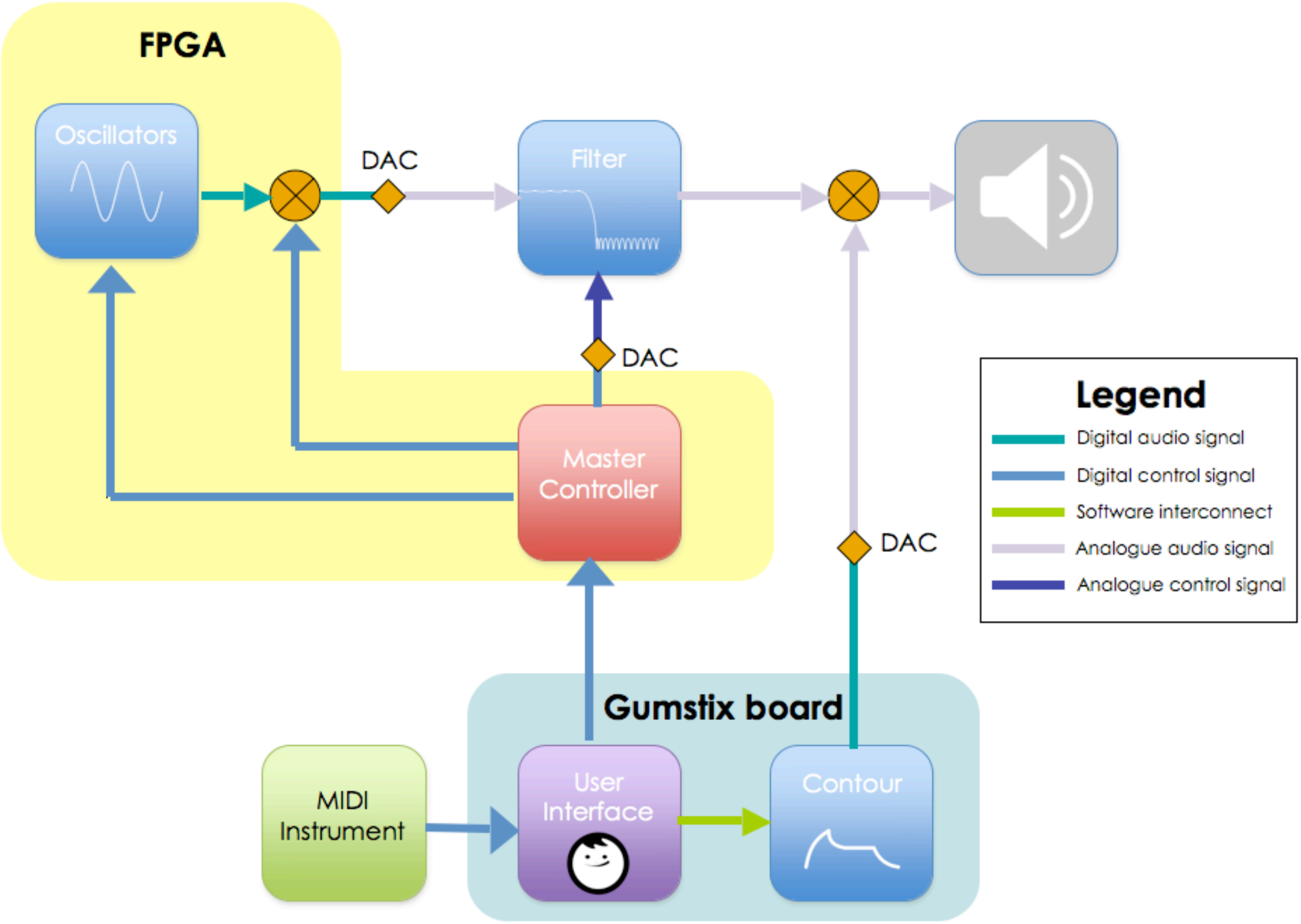


Figure 2. System block diagram

Module specifications and development approach

Interfacing Control System

The interfacing control system consists of the following components:

MIDI receiver

A MIDI receiver will be implemented using a combination of basic hardware and software connected to the FPGA development board. This will enable a MIDI compliant device to be connected to the FPGA.

Serial Link and Serial-control-protocol

An RS232 compliant serial link will be established between the master controller and the gumstix board. A control protocol and set of control instructions will be formulated to specify the manner in which data is transmitted between these devices and the means by which data integrity will be maintained.

Master Controller

The master controller will be implemented upon the FPGA development board as a soft-processor. It will be responsible for the following functions:

- Receiving raw MIDI input from the MIDI receiver hardware
- Converting MIDI input into system control signals and distributing them to the relevant synth module (e.g. conversion of MIDI note signals into to frequency control vectors to be sent to the oscillator)
- Receiving serial-control-protocol encoded messages from the gumstix board and converting this into control signals
- Encoding and sending serial-control-protocol messages to the gumstix board relating to UI event updates and synthesizer statistics e.g. MIDI control changes

Gumstix Controller API

The gumstix controller API will be a collection of functions enabling communication and control of synthesizer parameters between the user interface and the FPGA development board.

This collection will include:

- An event handling and distribution function for receiving serial-control-protocol encoded data from the FPGA and using it to trigger UI widgets and events
- Functions for the encoding of UI widget variables into serial-control-protocol messages to be sent to the FPGA

Oscillators

The oscillators will implemented digitally upon the FPGA development board using VHDL according to the following process:

1. Develop functions for rectangular, sawtooth, and triangle waves in terms of discrete sample, *n*, and carrier frequency, fc
2. Develop a lookup table function for the sine oscillators with frequency as an input parameter
3. Create a lookup table for the sine function using MATLAB and covert table’s values into 16-bit integers
4. Simulate functions using MATLAB to ensure their validity
5. Convert sine-lookup-table to a format such that it can be stored on the development board’s internal RAM
6. Implement all functions as VHDL
7. Make implementations controllable from the master controller
8. Test practical design

Controllable Filter

As part of the project’s budget, a pre-existing analogue voltage controlled filter has been purchased. This was chosen on account of its ease of implementation in comparison to the development of a digital filter.

The filter requires an analogue control voltage to set its cut-off frequency. The combination of a digital-to-analogue converter and a software algorithm on the FPGA will be required to enable its control digitally.

Envelope Generator

The envelope generator will be written as a software module upon the gumstix board. The software will function as follows

1. Convert numerical values for GUI ADSR timings and volume levels into a time-based digital waveform to be sent to the FPGA via serial link
2. Develop VHDL code on FPGA to perform any necessary conversions to received envelope output signal such that it is ready for conversion to analogue
3. Use a DAC to convert the envelope to an analogue signal
4. Mix envelope signal with output signal from controllable filter to produce output waveform

User Interface

A touch-screen graphical user interface will be developed as software on the gumstix board according to the following process:

1. Install embedded Linux operating system upon gumstix platform
2. Acquire, configure, and install pre-existing gumstix LCD-module kernel extensions (for display output and touch controller). Test using available applications
3. Investigate, select, configure, and install an appropriate Windowing toolkit for GUI I/O.
4. Research, design, and develop GUI widgets for the control of synthesizer parameters
5. Design a tabular full-screen interface for user navigation between categories of synthesizer control, e.g. a tab for filter parameters, a tab for oscillator selection.
6. Integrate developed widgets from *4* into *5*
7. Arrange for user acceptance test of incomplete GUI and implement some feedback
8. Add appropriate logic to glue UI widgets to control-functions and protocols developed as part of the interfacing control system.
9. Develop event listeners for the control of widgets using MIDI

Project Management

Milestones

Listed below are the milestones devised for this project

1. Development and Understanding of Synthesis Components

Commencement Date: Week 6, Semester 1

Completion Date: End of Week 12, Semester 1

Major Outcomes:

1. Development of various components necessary to synthesize music
2. Understanding how these components operate and may be controlled

Assigned To:

1 & 2. Ben Davey, Nathan Sweetman, Aaron Frishling, Andreas Wulf

Details:

This will include research into component operation as well as construction of either digital or analogue versions dependant upon the complexity of each and the expected quality of the components produced. However, if the components are significantly complex and a cheaper, higher quality COTS version is available, this may be purchased in order to lower development time.

Each component will be tested in order to verify that it is sufficient and if not, an improved version will be constructed, possibly changing from an analogue design to a digital design or vice versa.

A large proportion of this time will be devoted to gaining an understanding of the operation and control mechanisms for each of the components required. Components that are difficult to control will cause the system to be unnecessarily complicated and may reduce system reliability.

Each component can be developed, tested and researched in parallel as there is minimal interdependence between them. It is also possible to research the controllability of the components without them being operational.

2. Development of Interconnection and Control Systems

Commencement Date: Week 13, Semester 1

Completion Date: Supplementary Exam Week, Semester 1

Major Outcomes:

1. Interconnection of components
2. Control system

Assigned to:

1. Aaron Frishling, Andreas Wulf
2. Ben Davey, Nathan Sweetman

Details:

One of the requirements of the system is that it should allow the user to customise how the synthesis components are connected as much as possible. In order to allow for this, a significant amount of time and effort must be spent on researching and developing an interconnection system. It will be necessary to make sure that this does not affect quality or result in an overly complicated, uncontrollable system.

The control system for the components is a vital component of the overall project. This component will control all of the parameters of the audio synthesis components, along with their synchronisation and operation. Without implementing this, it will be impossible to develop a MIDI interface to the system and the customisability of the system will be significantly lowered.

The interconnection and control systems may be developed in parallel, but as it is intended that the interconnection of the audio synthesis components be digitally controlled, the interconnection system must be completed before the control system. It is necessary to know about the operational parameters and controllability of all of the components from the previous milestone before work on either of these systems is commenced.

3. MIDI and Graphical User Interface (GUI)

Commencement Date: Supplementary Exam Week, Semester 1

Completion Date: End of week 8, Semester 2

Major Outcomes:

1. MIDI Controller
2. Graphical User Interface

Assigned To:

1. Aaron Frishling, Nathan Sweetman
2. Ben Davey, Andreas Wulf

Details:

Using a MIDI interface will allow the synthesizer to be used similarly to any modern digital instrument. This allows for connection to the multitude of available MIDI output sources, e.g. keyboards, guitars, flutes or a computer with MIDI output capabilities. Implementing MIDI control will involve various subtasks due to the nature of the protocol and its features.

The GUI represents one of the significant differences between this system and many other audio synthesizers already available. It must be designed to be simple to use and visually appealing, while still providing as many configurable options as possible.

In order to achieve a design that meets these criteria, it will be necessary to thoroughly research and test the design. This section also requires familiarisation with both the hardware and software environments which will be used for the GUI implementation.

4. User Acceptance Testing and General Refinement

Commencement Date: Week 1 of Mid-Semester Break, Semester 2

Completion Date: Week 9, Semester 2

Major Outcomes

1. Completed System
2. Case to house the completed system

3. Additional Effects

Assigned To:

1. Ben Davey, Nathan Sweetman, Aaron Frishling, Andreas Wulf
2. Andreas Wulf, Aaron Frishling
3. Ben Davey, Nathan Sweetman, Aaron Frishling, Andreas Wulf

Details:

This milestone represents the completion of the synthesizer. As such, the tasks that are required in order to reach this milestone are not completely set in stone. Due to the nature of the proposed tasks involved in the milestone, many of them can be commenced before the completion of the previous milestone.

The synthesizer will be packaged as a discrete system. This will include the creation of a case for it and any further compnents which are necessary to ensure that it is stand-alone system.

The synthesizer must undergo user acceptance testing. This means that people of varying knowledge in the field of synthesised music will be brought in to attempt to use and evaluate the capabilities of the end product. This may result in some system changes in order to improve it in line with the gathered information.

If there is additional time remaining due to early completion of other tasks in the construction of the system, there is scope to create additional refinements to the system such as implementing means to improve sound quality and creating components to produce additional audio effects (such as echo and chorus).

Budget and Required Resources

The total budget for this project is $1000, whereby each of the four members have been allocated $250. The majority of the project relies on equipment provided by the department (The School of Electronic and Electrical Engineering) and the members, which significantly reduces the strain on the budget.

Resources provided free-of-charge

The no additional cost items are as follows:

Resource	Provider	Description
Bench space & electricity	EEE department	Location provided to perform the majority of project hardware development
Development PC	EEE department	Linux-based PC provided with bench space
Laptops	Group members	Each group member has provided their own laptops that will be used for the development of software.
Signal Generator, Oscilloscope & Regulated Power supply	EEE department	Used for the development and analysis of hardware components

Resource	Provider	Description
Rudimentary tools	Group members	Tools such as pliers, screwdrivers, wire strippers and oscilliscope probes will be used for hardware development
Bread board	EEE department	Used for the prototyping of electronic circuits prior to final implementation
FPGA Development board	EEE department	The department has provided a Spartan-III based Altium livedesign evaluation board. This will be used for the implementation of oscillators and the majority of the project's control system
MIDI Keyboard	Group members	Musical interface device required to drive the synthesizer
PCB printing	EEE department	The department provides facilities for the development of PCBs free of charge to project students
Development software	Free for evaluation/ open source	FPGA comes with Altium Designer 6, which contains development tools for the FPGA and for designing hardware such as PCBs. The gumstix uses open source software tools

Budgeted resources

The following table outlines budgeted resources and their prices:

Resource	Description	Cost ($AUS)
Gumstix LCD pack	Package includes: • 5.0 volt power adapter (supplies power to the gumstix system) • Serial null-modem cable (for interfacing/programming) • Screws & Spacers kit (for assembly) • gumstix verdex XL6P (the computer) • USB Gender Changer (adaptor) • netmicroSD-vx (daughter board the gumstix mainboard enabling storage of applications using SD-card memory) • Samsung™ 4.3" LCD touch panel (display/user interface) • consoleLCD16-vx (gumstix daughterboard providing LCD interface and serial I/O)	500
CEM3389	Integrated Circuit containing a Voltage Controlled Filter (VCF) with resonance and 4 Voltage Controlled Amplifiers (VCA).	50

Resource	**Description**	**Cost ($AUS)**
Rudimentary electronic components	Budget set-aside for commonly available ICs, resistors, capacitors, cables etc. for development of the hardware components	100
Project chassis	The chassis will provide a body to house all electrical components, switches, bezels, and mounts	50
Power supply unit	A second-hand ATX power supply from a PC will be adapted for use as a power supply for the system. It contains a +3.3volt line for the FPGA board, +5volt for the gumstix and +/-12volt for the remainder of the hardware.	10
	Total	710
	Remaining budget	290

The total budget allocated is $710 giving a padding of $290, which should be enough to cover an essential component would it fail out of warranty, or to purchase extra components if they are required in the future. The remaining budget can be spent on presentation such as a fancier case for the synthesizer or a permanent FPGA board as the FPGA development board will have to be surrendered once the project has concluded.

Risk Management Strategies

A number of risks have been identified in the project:

Unacceptable audio quality

Outcome: The synthesizer is meant to be an entertainment device, such that it produces sounds that are pleasant to a human's ear. Failing this requirement makes this product useless except as a concept device.

Likelihood: Low

Severity: Catastrophic

Reduction strategy: Creating the system in stages, whereby each stage will be tested and adjusted to improve audible quality and to reduce Total Harmonic Distortion (THD).

Action Plan: Focus the presentation of the project on its unique features and concept.

Major component failure

Outcome: As the project contains a few expensive components, if a component sustains any damage the cost of replacing it may be more than the budget allows.

Likelihood: Medium

Severity: Low

Reduction strategy: The most likely way to cause damage to components is by connecting them up incorrectly, thus double checking the designs and connections before powering up any devices is necessary to reduce the risk.

Action Plan: Allocate replacement into the budget or if replacements are too expensive find a substitute.

Incomplete System upon project deadline

Outcome: Interconnecting all the modules is a timely task, thus may not be completed by the deadline. An incomplete system may not function properly or work at all depending on what module is affected, thus not meeting requirements.

Likelihood: Medium

Severity: Low

Reducing Risk: Preventing feature creep and sticking to a well-planned work schedule will reduce this risk.

Action Plan: Focus presentation on completed parts and hide/cover-up the incomplete.

Project does not meet specified requirements

Outcome: When the system is complete but lacks all the requirements initially set out, may result in an irrelevant product.

Likelihood: High

Severity: Low

Reducing Risk: Ensure all development of the project lead to fulfilling the requirements.

Action Plan: Pretend that is what was meant to happen.

Project goes over budget

Outcome: If allocation of the budget has been too ambitious or funds dry out the project may not be completed. This may occur if the budget is poorly planned or if more components then initially planned for are required .

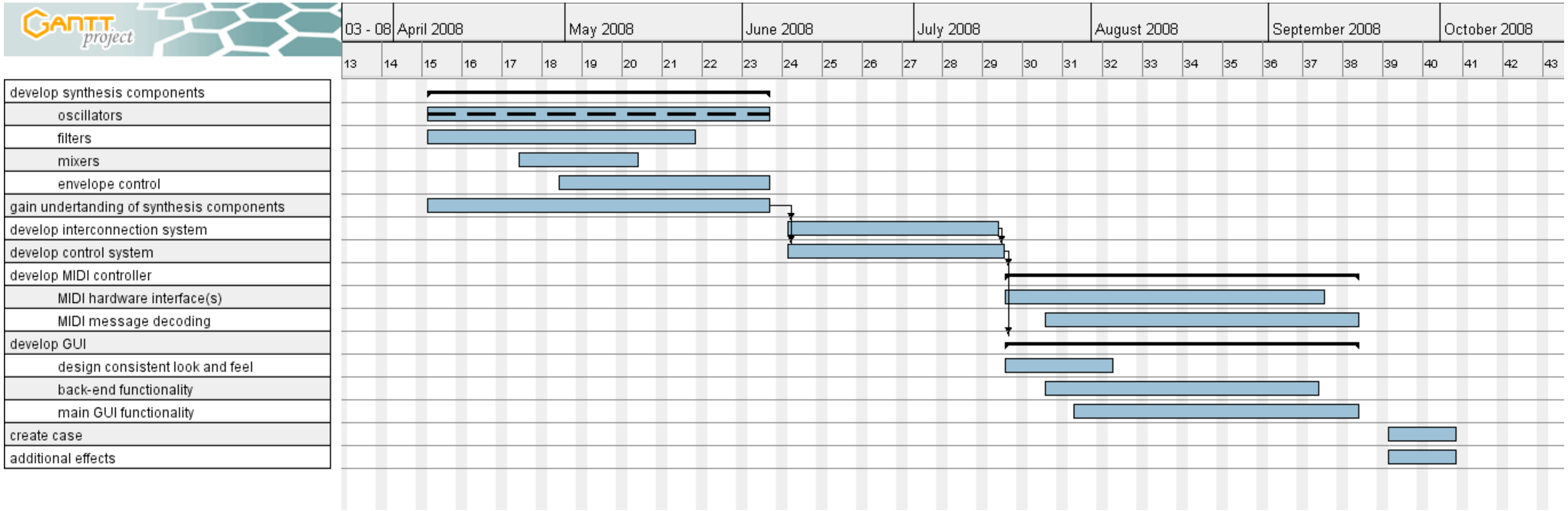
Likelihood: Medium

Severity: Medium

Reducing Risk: Have a well-planned budget and avoid expensive and unnecessary components/purchases.

Action Plan: The project can receive funding from the group members and use freely available items.

Appendix A – Gantt Chart



References

- Russ, Martin (2004) : Sound Synthesis and Sampling, Second Edition
- Pinch, Trevor (2004) : Analog Days: The Invention and Impact of the Moog Synthesizer
- Shapiro, Peter (2000) : Modulations : a history of electronic music. 2005
- Schmitz, Reinhard (2005): Analog Synthesis
- Gumstix company website : http://www.gumstix.com
- Qtopia windowing toolkit platform overview: http://trolltech.com/products/qtopia