Phase 1 Report: Project Proposal

Digital Guitar Effects Processor

A Report
Presented to
The Department of Electrical & Computer Engineering
Concordia University

In Partial Fulfillment
of the Requirements
of ELEC 490

By
Group 59

Pascal Everton ID: 4774442
Armen Forget ID: 4604156
Gunnar Beauregard ID: 4193857

Project Supervisor(s)
Dr. W. Lynch,
Dr. S. Kort

Concordia University
November 2004

1. PROJECT DESCRIPTION

1.1 Product Description

An audio effects processor serves to modify the input signal in various ways in order to obtain a variety of audio effects desired by musicians. The effects processor that we intend to create will be externally similar to existing units. It will be housed in a standard sized 19" rack-mount box. It will have similar controls to existing units. The main difference our product presents is that the signal processing will be performed in an FPGA and not in a DSP. Although it has been indicated that FPGAs are possibly faster than DSPs[1], the purpose of this project is not to display this speed; it is to serve as a proof of concept for processing audio signals in an FPGA.

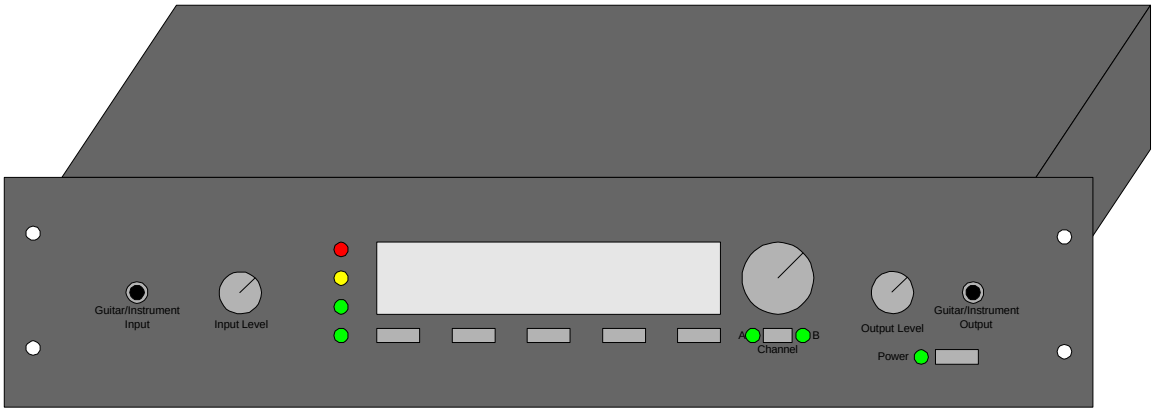

Figure 1 - FPGA based Effects Processor

As shown in figure 1, the system has a look similar to other units and the controls are comparable to many other systems. The main difference is on the back of the unit (figure 2) where there is an expansion port allowing the user to change the set of algorithms that the unit can use. This allows the complete sound of the system to be changed by simply inserting or replacing an expansion cartridge.

---

1 http://bwrc.eecs.berkeley.edu/Presentations/Retreats/Winter_Retreat_Jan_2000/Posters/pdf/Microsoft%20PowerPoint%20-%20hayden_poster.pdf p.10

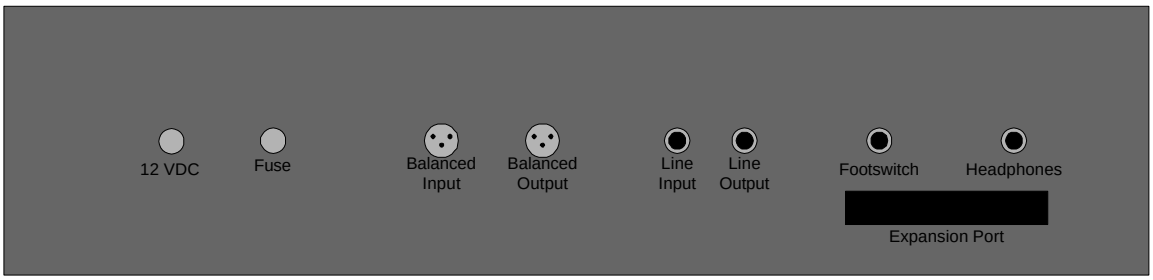

Figure 2 - FPGA based Effects Processor (Back Panel)

The operation of the system follows the block diagram shown in figure 3. From the initial input, the analog signal goes through a number of signal conditioning stages before it is converted to a digital signal. This digital signal is then applied to the input of the processor. The operation of the system is largely controlled by a microcontroller that acquires user input from a number of switches and provides feedback to the user with an LCD screen and several LEDs. After the signal has been processed in the FPGA, it is converted back into an analog signal and conditioned to provide three different outputs.

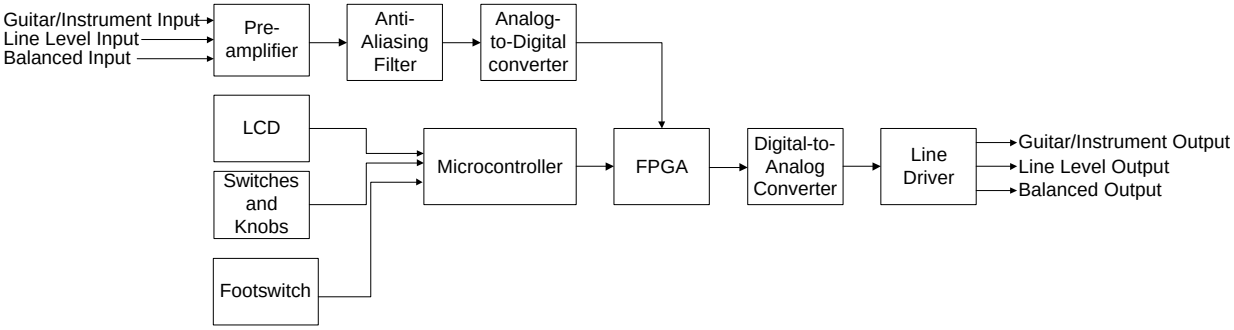

Figure 3 - System Level Block Diagram

The system will be developed using several different software tools. The signal processing algorithms will be developed using the tool chain shown in figure 4. The hardware will be developed using the tool chain shown in figure 5.

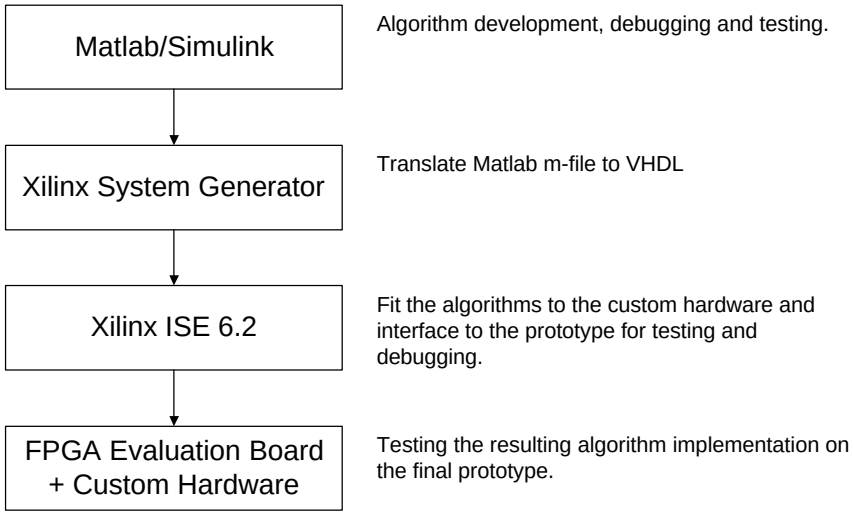

Figure 4 - Algorithm Development Tool Chain

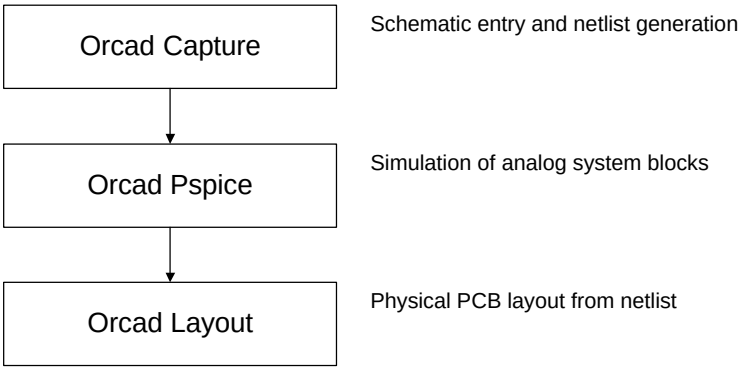

Figure 5 - Hardware Development Tool Chain

In summary, the main goals of the project are as follows:

- Create a hardware platform that serves as a proof of concept for audio signal processing using an FPGA
- Implement several audio effects of comparable audio quality to an industry standard unit based on DSPs

1.2 Product Specification

Inputs:

Description	Impedance	Level	Connector
Guitar/Instrument Input	1MO	-20 dBu	¼" Female
Balanced XLR Input	2kO	+20dBu	XLR Female
Line level Input	10kO	0 dBu	¼" Female

Outputs:

Description	Impedance	Level	Connector
Guitar/Instrument Output	1kO	-20 dBu	¼" Female
Balanced XLR Output	100O	+20dBu	XLR Male
Line Level Output	100O	0 dBu	¼" Female

User Interface:

Description	Usage
LCD Screen	Displays the configuration screens of the system menu structure
Input Level Indicator LEDs	Aids the user in adjusting the input level to set the optimum level
Power Switch	Switches the system on and off
Power LED	Indicates if the unit is powered on
Channel Switch	Switch between Channel A and Channel B
Footswitch	Switch between Channel A and Channel B while playing
Channel A and B LEDs	Indicates the present processing channel
User Selection Switches	Allows the user to make selections from the menus on the LCD screen
Input Level	Sets the input signal gain
Output Level	Sets the output signal gain
User Selection Dial	Allows the user to adjust the selected effects parameters in a quasi-analog fashion

2. TASK BREAKDOWN

2.1 Task Descriptions

2.1.1 Algorithm Development (Task ID 9)

Develop each of the algorithms that will be implemented. This will involve finding a mathematical model, translating and testing of each algorithm.

2.1.1.1 Find references for each effect (Task ID 10)

Find mathematical descriptions of each effect that we intend to implement in the time or frequency domain.

2.1.1.2 Implement each effect in Matlab (Task ID 19)

Translate each algorithm from a mathematical model into a Matlab™ m-file.

2.1.1.3 Test Algorithms in Matlab (Task ID 28)

Test each effect in Matlab™ to ensure that the desired effect is produced.

2.1.1.4 Translate m-file into VHDL using Xilinx System Generator (Task ID 38)

Use Xilinx System Generator to translate the algorithm from the representation in a Matlab™ m-file to a VHDL module capable of being transferred onto the FPGA.

2.1.2 Define User Interface

Implement an intuitive system of menus that will be displayed on the LCD screen that will allow the user to set all the adjustable parameters of each effect.

2.1.2.1 Define user settable variables for each effect (Task ID 40)

For each effect, there will be parameters that may be set by the user to control the way the sound is altered. These parameters need to be defined for each effect.

2.1.2.2 Design Menu Structure (Task ID 49)

Design the menu structure used to access the parameters of each effect.

2.1.2.3 Program Menu structure in ANSI compatible C (Task ID 50)

The menu structure will be implemented as a module coded in ANSI C to ensure portability.

2.1.2.4 Test menu structure on a PC (Task ID 51)

The operation of the menu structure will be simulated on a PC prior to being tested on the target hardware. This is possible because the module will have been coded in ANSI C so that it is portable.

2.1.3 Hardware Development

The system hardware used to make the system needs to be defined. Some of the elements will be implemented on an FPGA evaluation board.

2.1.3.1 Develop system block diagram (Task ID 54)

A block diagram of the system will be made that describes the system in terms of the main functional sub-systems and the interfaces between them.

2.1.3.2 Source the primary parts of the system (A/D, D/A, FPGA) (Task ID 55)

Select and purchase the main components of the system.

2.1.3.3 Define Expansion Port Interface (Task ID 142)

Define the interface between the expansion board and the main system.

2.1.3.4 Define System Power Supply (Task ID 143)

Determine the approximant power requirement of the system of the system and design a suitable power supply.

2.1.3.5 Make schematics for each system block (Task ID 56)

Enter the schematic of each system block into Orcad Capture™.

2.1.3.6 Simulate the analog section of the system (Task ID 64)

The analog sections of the system will be simulated using Orcad Pspice™ to verify their correct operation.

2.1.3.7 Define which section of the system will be implemented on the evaluation board and which will be on a custom PCB (Task ID 68)

As this is a prototype, a portion of the system will be implemented on an FPGA evaluation board. Consequently, the elements of the system that will be implemented on a custom PCB must be defined.

2.1.3.8 Layout PCB (Task ID 69)

Layout the sections of the system that are contained on the custom PCB using Orcad Layout™

2.1.3.9 Make PCB (Task ID 70)

Manufacture the custom PCB.

2.1.3.10 Test and Debug (Task ID 71)

Test and debug the integrated blocks of the system.

2.1.4 System Integration

2.1.4.1 Define PIC/FPGA Interface (Task ID 74)

The interface between the PIC and FPGA needs to be defined so that the configuration information entered by the user can be moved from the PIC into the FPGA.

2.1.4.2 Code VHDL Interface to A/D (Task ID 75)

Create the VHDL module used to interface to the A/D Converter.

2.1.4.3 Code VHDL Interface to the PIC (Task ID 76)

The VHDL module used to interface the FPGA to the PIC needs to be coded from the interface definition.

2.1.4.4 Code PIC Interface to FPGA (Task ID 77)

The C module used to interface the PIC to the FPGA needs to be coded from the interface definition.

2.1.4.5 Code VHDL Interface to D/A (Task ID 78)

Create the VHDL module used to interface to the D/A Converter.

2.1.4.6 Code Menu Driver / LCD Interface PIC (Task ID 79)

The driver routines specific to the target hardware must be written to run the user interface module.

2.1.4.7 Modify Prototype Enclosure (Task ID 80)

Modify the prototype enclosure to accommodate the prototype electronics.

2.1.4.8 Assemble Prototype (Task ID 81)

Assemble the prototype electronics in the enclosure.

2.1.5 Project Management (Bi-Weekly)

2.1.5.1 Modify Project Schedule to Meet Deadline (Task ID 84)

As the actual time to complete each task will likely vary from the estimated time, the project schedule must be periodically revised so that deadlines are satisfied.

2.1.5.2 Monitor Project Progress (Task ID 98)

The state of completion of each task needs to be monitored to provide accurate information with which to update the project schedule.

2.1.5.3 Maintain Project Schedule (Task ID 112)

The project schedule should contain current information concerning each task. As tasks are completed, this information needs to be entered into the schedule.

2.1.5.4 Generate Progress Update Documentation (Task ID 126)

Documentation information needs to be generated to communicate the project status to all concerned parties.

2.2 Task Assignments

2.2.1 Armen Forget – Task Assignments

Task Name	Task ID
Distortion - Define user variables	41
Compression - Define user variables	42
Reverberation – Define user variables	43
Equalization - Define user variables	44
Delay - Define user variables	45
Noise Gate - Define user variables	46
Tremolo - Define user variables	47
Chorus - Define user variables	48
Design Menu Structure	49
Program Menu structure in ANSI compatible C	50
Test menu structure on a PC	51
Preamplifier – Schematic	57
Anti-Aliasing Filter - Schematic	58
Line Driver – Schematic	63
Preamplifier – Simulate	65
Anti-Aliasing Filter - Simulate	66
Line Driver – Simulate	67
Make PCB	70
Test and Debug	71
Code VHDL Interface to A/D	75
Code VHDL Interface to the PIC	76
Code VHDL Interface to D/A	78
Modify Prototype Enclosure	80
Assemble Prototype	81

Table 1 - Task Assignments - Armen Forget

2.2.2 Pascal Everton – Task Assignments

Task Name	Task ID
Distortion - Find references	11
Compression - Find references	12
Reverberation - Find references	13
Equalization - Find references	14
Delay - Find references	15
Noise Gate - Find references	16
Tremolo - Find references	17
Chorus - Find references	18
Distortion - Implement in Matlab	20
Compression - Implement in Matlab	21
Reverberation - Implement in Matlab	22
Equalization - Implement in Matlab	23
Delay - Implement in Matlab	24
Noise Gate - Implement in Matlab	25
Tremolo - Implement in Matlab	26
Chorus - Implement in Matlab	27
Distortion - Test in Matlab	29
Compression - Test in Matlab	30
Reverberation - Test in Matlab	31
Equalization - Test in Matlab	32
Delay - Test in Matlab	33
Noise Gate - Test in Matlab	34
Tremolo - Test in Matlab	35
Chorus - Test in Matlab	36
Design Menu Structure	49
Test and Debug	71
Modify Prototype Enclosure	80
Assemble Prototype	81

Table 2 - Task Assignments - Pascal Everton

2.2.3 Gunnar Beauregard – Task Assignments

Task Name	Task ID
Translate Matlab m-file into VHDL using Xilinx System Generator	38
Develop system block diagram	54
Source the primary parts	55
A/D Converter - Schematic	59
FPGA and PROM - Schematic	60
LCD, PIC and User I/O - Schematic	61
D/A Converter - Schematic	62
Evaluation Board / Custom PCB	68
Layout PCB	69
Test and Debug	71
Define PIC/FPGA Interface	74
Code PIC Interface to FPGA	77
Code Menu Driver / LCD Interface PIC	79
Modify Prototype Enclosure	80
Assemble Prototype	81
Modify Project Schedule to Meet Deadline	85
Monitor Project Progress	99
Maintain Project Schedule	113
Generate Progress Update Documentation	127

Table 3 - Task Assignment - Gunnar Beauregard

3. SCHEDULE

As the schedule is not easily viewed on an 8.5” x 11” sheet, please see the attached large format schedule printout attached to the back of this document.

4. COMPONENT / TOOL LIST

4.1 Software Tools

• Matlab/Simulink	Initial algorithm development and testing
• Xilinx System Generator 6.2	Automatic translation tool specifically used to translate Matlab algorithms into VHDL optimized for Xilinx FPGAs.
• Xilinx ISE 6.2	Used to create hardware specific VHDL files that allow the VHDL algorithms to operate on the custom hardware
• CCS PCW	C Compiler used for Microchip PIC
• Microsoft Visual Studio .NET	Used to simulate the operation of the menu structure
• Orcad Capture 9.2	Used to enter the system schematic, create the PCB BOM and output the netlist used to create the PCB
• Orcad Layout 9.2	Used to design the custom PCB that corresponds to the netlist generated in Orcad Capture 9.2™
• Microsoft Word	Used to write project documentation
• Microsoft Excel	Used to create project budget
• Microsoft Project	Used to define, organize and track the project in terms of individual tasks and sub-tasks
• Microsoft Visio	Used to make project drawings and figures for use in the project documentation

4.2 Hardware Tools

- Oscilloscope — General system debugging and specification testing
- Spectrum Analyzer — Observing the frequency domain performance of certain algorithms
- Function Generator — Generation of test signal for debugging and specification testing.
- Laboratory Power Supply — To allow individual sections of the system to be tested prior to the assembly of the system power supply
- Soldering Iron — Used to assemble the custom PCB
- Miscellaneous Hand Tools — Used to assemble the prototype

4.3 Prototype Hardware

- Metal Enclosure
- FPGA Evaluation Board
- A/D and D/A converters
- LCD Display
- Microchip PIC microcontroller
- Various Integrated Circuits and Discrete Components
- Various connectors and switches

5. PRELIMINARY BUDGET

Projected Prototype BOM

Item	Cost
Enclosure (Digi-Key PN: 377-1205-ND)	$33.96
Rotary Encoder (Digi-Key PN: P10859-ND)	$2.08
Audio Connectors	Not yet specified
Potentiometers	Not yet specified
Mounting Hardware	Not yet specified
A/D Converter	Not yet specified
D/A Converter	Not yet specified
Misc. Electrical Components	Not yet specified
FPGA Evaluation Board (Provided by University)	$0.00
Microchip PIC18F252 (Provided by students)	$0.00
LCD Display (Provided by students)	$0.00
PCB Material (Provided by students)	$0.00

Tool Expenses

No new tools need to be purchased