Electronics and Computer Science

Faculty of Engineering and Physical Sciences

University of Southampton

Benjamin Parry

7th December 2022

An Audio Processor on an FPGA

Project supervisor: Tomasz Kazmierski

Second examiner: <2nd Examiner>

A project progress report submitted for the award of

Electrical and Electronic Engineering MEng

Abstract

No more than 200 words.

This template includes a suggested structure for your progress report.

The choice of section headings has been designed to map to the marking criteria given to supervisors. You can you these headings if you want to but they are not compulsory.

Contents

# Introduction

The outcome of this project is to design an audio processor that is implemented on an Altera De2-115 Cyclone IV FPGA. The processor will be based on an embedded microarchitecture designed using a RISC instruction set architecture and will be adapted to work with the on-board audio CODEC.

This scope of this project is to design and implement an embedded microarchitecture and to explore audio processing techniques and implement them onto the processor. Techniques such as equalisation, synthesis, and compression.

The processor will function as an equaliser which will take a phone input to provide the audio signal to be equalised. The audio will then be put through a parametric equaliser which can control the gain of frequency bands, the centre frequency and the cut off frequencies of the band in real time. This will be controlled by external i/o and the output will be played through a speaker to listen to the new sound.

If time permitting the processor will function as a synthesiser. The processor will use audio synthesis techniques and potentially MIDI to synthesis audio that is stored on the FPGA. Audio compression may also be added to the processor. This would be part of the equaliser and would compress the audio signal before it is passed to the equaliser.

At the point of writing, the first version of a RISC-V processor has been designed and tested which can carry out a small number of instructions. Two versions of an equaliser have been designed for the processor, only one will be implemented and this decision will be made later on depending on the capability of the designed processor.

# Background and report of literature search

## Embedded Microarchitecture/ISA

Talk about NIOS vs RISC -> MIPS vs RISC-V

Decisions why I chose to implement RISC-V over other ISAs

Talk about the reasons for the project?

## Equalisation

Talk about parametric equaliser, IIR vs FIR and the maths behind coefficient calculation.

Decisions behind parametric filter using IIR and why this may not be feasible

## Synthesis

Talk about different types of synthesis, additive, subtractive and FM.

# Report on Technical Progress

First version of RISC-V processor in system Verilog which can handle a limited instruction set.

Testing using Assembly language code that’s been written by hand.

First version of filters that has been written in C.

# Plan of remaining work

Finish implementing the processors remaining instructions

Potentially add in the multiplication and floating-point extensions

Finish coding the equaliser in C and then convert this to RISC-V assembly.

With an online interpreter/assembler or one that I design myself to convert C to assembly and then into 32 bit binary instructions.

Parametric equaliser is preferred but if not feasible a static one to be designed

Alter the design of the processor to communicate with the codec and i/o that is onboard the De2.

Finish the design of the synthesiser and audio compresser and then implement this in C and turn into binary.

References