Digital ASIC Fabrication

Jonathan Hess | Haris Khan | Samuel Heikens | Yuwei Tan

Group #: sdmay24-21

Advisor/Client: Dr. Henry Duwe

Introduction

- Design a custom digital **Application Specific Integrated Circuit (ASIC)**
- High barrier of entry with lack of resources or knowledge
- Our application is a guitar pedal with 3 effects: reverb, delay, and compression
- Implementing effects like reverb, compression, delay, and looping using digital hardware

What is Open MPW?

- Collaboration between Google, eFabless, and SkyWater foundry
- Allows for groups to create and submit open-source ASICs for FREE
- Typically expensive, makes more accessible to users without proprietary tools

Design Requirements

- 3 effects
  - delay, reverb, compression
- Audio sample size
  - 16 bits
- sampling rate
  - 10 kHz
  - 2x of 2637Hz (highest note)

Open-Source Tools Used

- EFabless Caravel Wrapper
- SkyWater 130 nm OpenPDK
- OpenLane
- KiCad
- Verilator

Users and Purposes

Users:

- Audio Engineers
- Musicians/Artists
- Future members Chip ISU

Purpose:

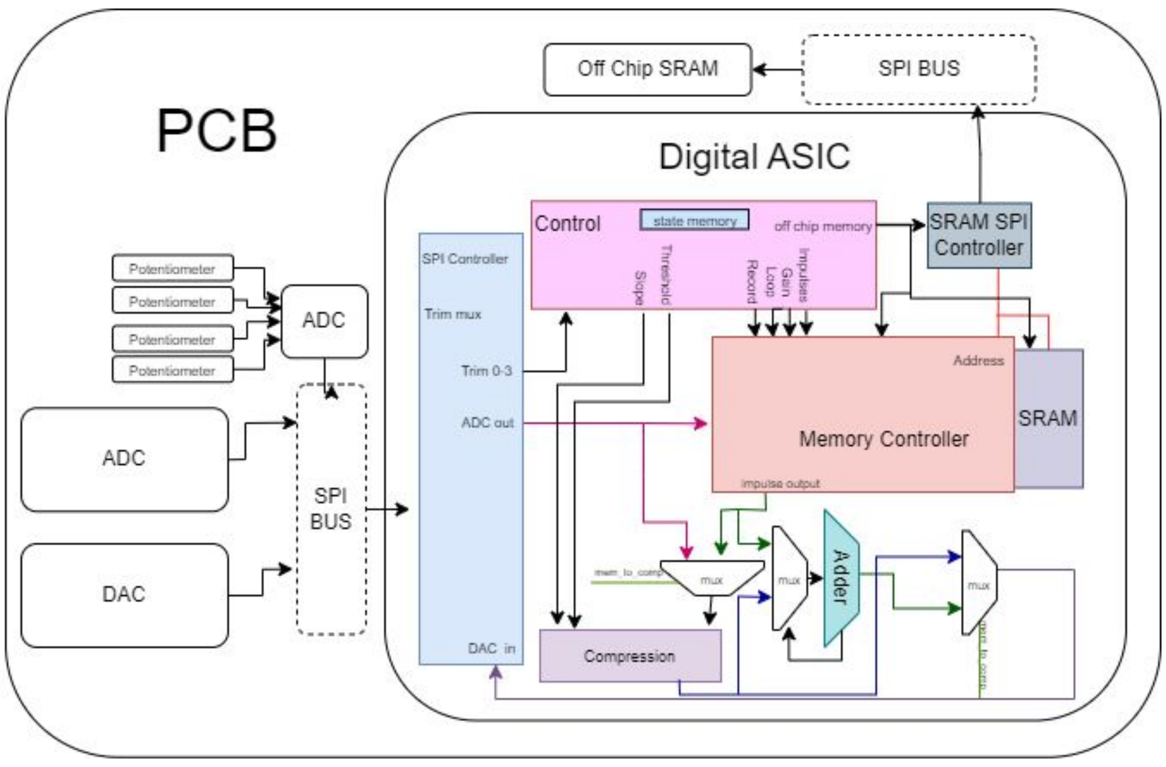
- To Provide future students ASICs to test and learn from
- Improve pedal usability for Musicians

Design Description

- We use a Finite Impulse Response (FIR), to give audio timbre
- This FIR design can implement any linear time invariant effect
- The FIR is created using data and impulses stored in the SRAM
- Using arithmetic operations, audio can be compressed giving non linear effects

Design Approach

- Modular design with multiple design modules
- Scalable design with configurable off-chip components
- Communicates over provided Wishbone Bus and SPI protocol
- Synthesize behavioral Verilog, custom cells, and memory macros



Design Modules

- Memory Controller
- SPI Controller
- SRAM
- Control
- Compression

Off-Chip

- SRAM
- ADC
- DAC
- Knobs
- Pre-amplifier

Technical Details

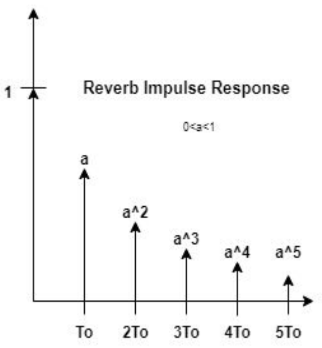
- Off-chip components are connected to ASIC via daughterboard
- Wishbone Test – Verify the provided Wishbone bus in the user area
- SPI – Communicate with other modules via external I/O

Testing

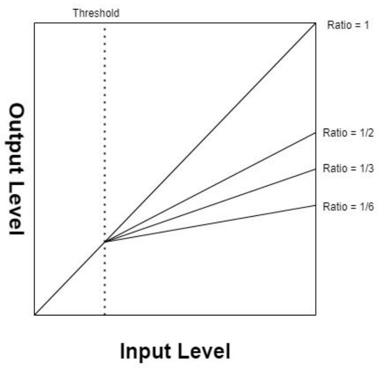
- RTL simulations of individual modules using Verilog
- Gate level simulations with synthesized design results
- Using Python to create test data and audio samples
- Signoff simulations with synthesized and placed macros
- Integrated top level verification using Verilog and C testbenches

Hardening

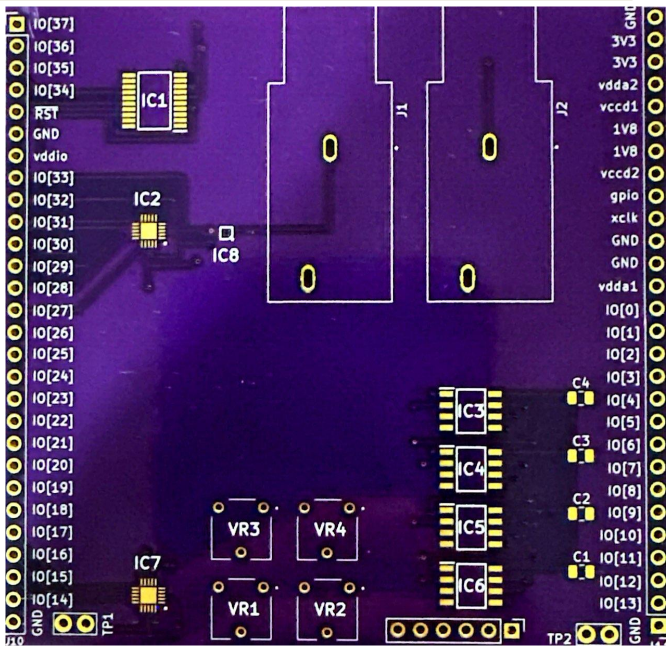
- Hardening is performed in Virtual Linux Machine
- Makefiles are used to setup environment and generate layout
- Run eFabless Precheck for final submission approval
- Includes DRC, LVS, and complexity checks



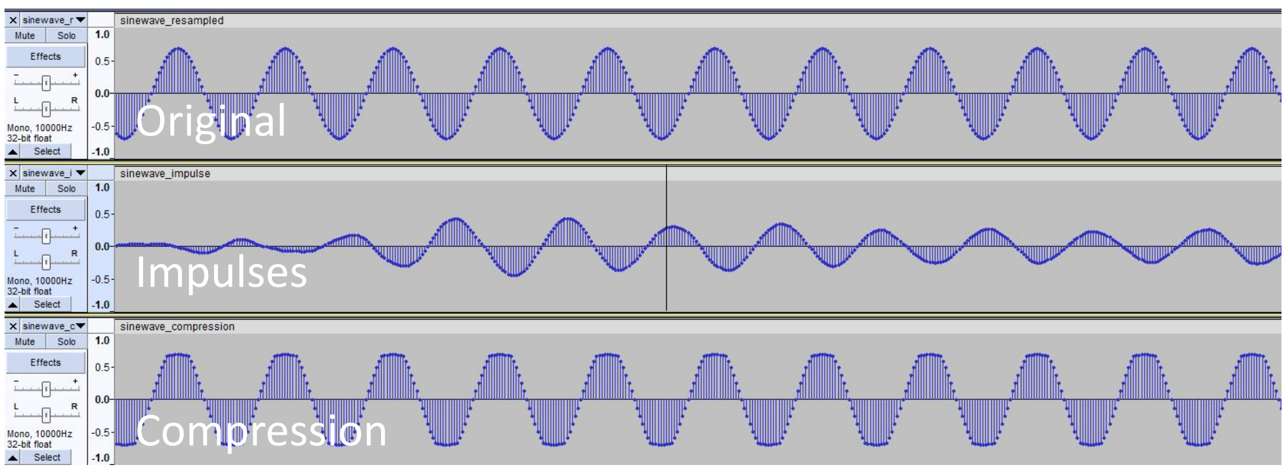
Impulse Response (linear effects)



Compression (non linear effect)



PCB - Daughterboard


sinewave_resampled
Original
sinewave_impulse
Impulses
sinewave_compression
Compression
Mute
Solo
Effects
Select
Mono, 10000Hz
32-bit float
1.0
0.5
0.0
-0.5
-1.0