

Digital Circuit Final Project Report

林軒霆, 蔡丞彥, 侯奕安

December 25, 2025

1 Introduction

For the final project, we implement a series of real-time guitar effect pedals to simulate a complete guitar pedal chain. The implemented effects range from simple modules such as noise gating and compression to more complex effects that require filtering or memory-dependent processing, including equalizers and delay effect. In addition, we implement a loop pedal that allows users to record and replay audio segments in the background in real time.

The effects chain follows real-world pedal ordering conventions commonly used by guitarists. Users can selectively enable individual pedals and configure their parameters according to their preferences. The system is operated and monitored through FPGA LEDs and control panels, which provide visual feedback and guide the user for smooth and intuitive operation.

2 File Structure

The code for our project is located in the `src/` folder.

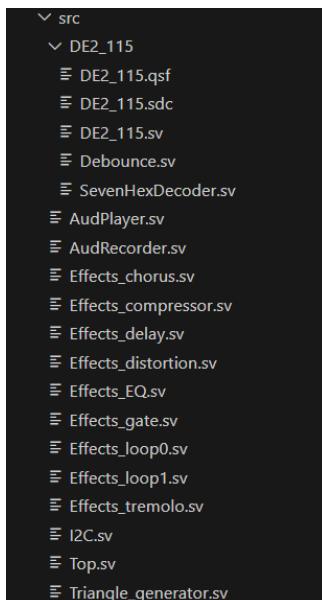


Figure 1: File Structure

3 System Architecture

The architecture of a guitar effects pedal is shown in the figure below.

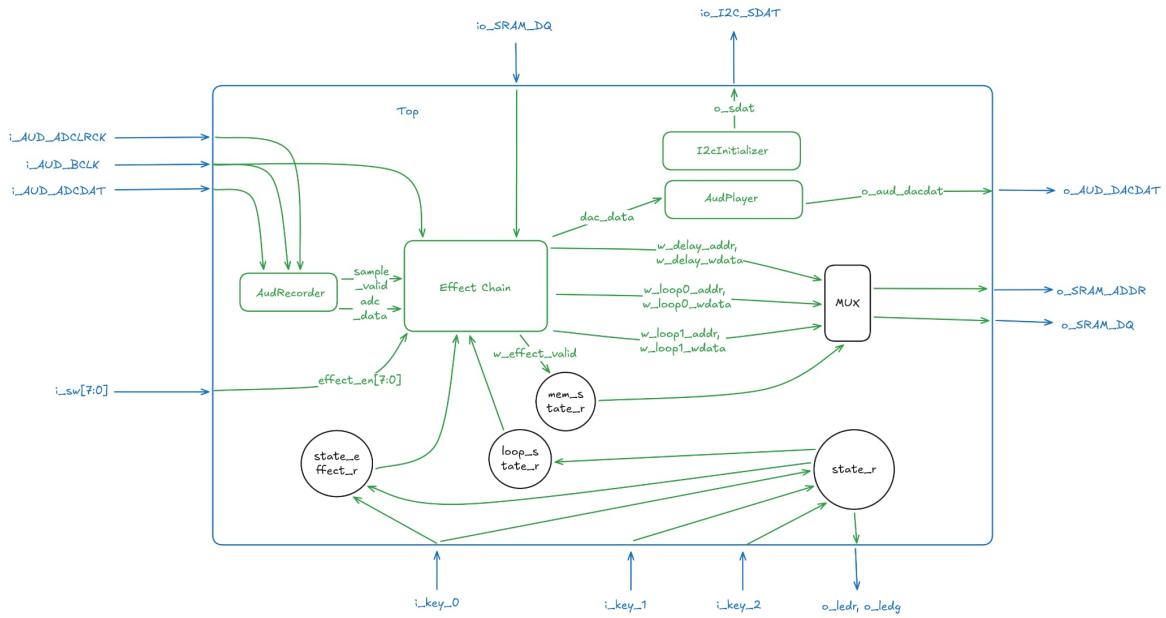


Figure 2: System Architecture

In the figure, some simple or less important signals (such as *i_clk* and *i_RST_N*) are omitted.

4 Hardware Scheduling

In this section, we describe the hardware architecture of our project, beginning with the top-level module that governs the overall operation of the effects pedal system, followed by the individual hardware components in the order in which they are applied to the audio signal.

4.1 Top Module

The Top module is the central connector of our design. It manages the state of the entire system, handles user inputs from keys and switches, and controls how different modules access the shared SRAM.

4.1.1 Finite State Machine

The main finite state machine for the top module is shown in Figure 3.

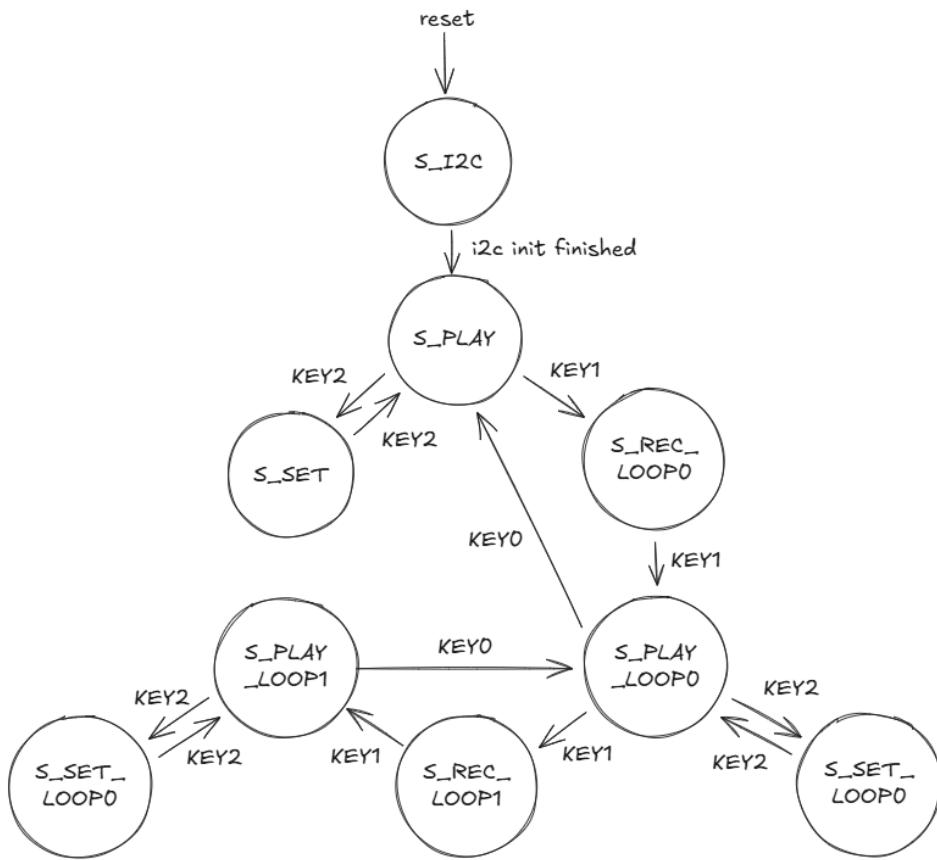


Figure 3: Finite state machine of the top module

After resetting and I2C initialization, the machine transitions to the **S_PLAY** state automatically. In this state, the system operates in its default mode, processing audio input in real-time through the effects chain and outputting it to the DAC. From the **S_PLAY** state, the user can navigate to two main functional branches using the push-buttons: the parameter setting mode (**S_SET**) and the looper mode.

Pressing **KEY2** transitions the FSM to the **S_SET** state. In this mode, the audio continues to play, but the user interface focus changes. The user can select specific effects using the switches and modify their parameters (such as distortion drive or delay level) using **KEY0**. Pressing **KEY2** again toggles the system back to **S_PLAY**, saving the current settings.

Pressing **KEY1** initiates the looper functionality, transitioning to **S_REC_LOOP0**. The system records audio into the first loop buffer until **KEY1** is pressed again or the recording finishes, at which point it switches to **S_PLAY_LOOP0** to playback the recorded audio in the background.

From the first loop playback state, the user has three options:

- Press **KEY1** to record a second layer (overdub), moving to **S_REC_LOOP1**.
- Press **KEY2** to enter **S_SET_LOOP0** to adjust effect pedal parameters or loop volume.
- Press **KEY0** to stop the loop and return to the main **S_PLAY** state.

When the system enters the second loop recording state (**S_REC_LOOP1**), it captures an overdub layer while continuing to play the first loop. This recording phase persists until **KEY1** is

pressed again or the memory buffer is filled, at which point the FSM transitions to the dual-loop playback state, `S_PLAY_LOOP1`.

In this final playback mode, both recorded loops are active. From here, the user has two control options:

- Press `KEY2` to enter `S_SET_LOOP1`. This mode allows the user to adjust the global effect parameters (such as distortion drive or EQ settings) or the loop playback volume while the loops continue to play in the background.
- Press `KEY0` to stop the second loop and return to the single-loop playback state (`S_PLAY_LOOP0`), effectively functioning as an “undo” operation for the overdub.

4.1.2 Audio Signal Chain

Inside the Top module, we connected the effects in a specific order, similar to how physical guitar pedals are chained together in real life. The audio data flows serially from one effect to the next. The chain starts with the `AudRecorder` (from Lab 3), passes through effects like the Gate, Compressor, and Distortion, and ends at the `AudPlayer`, which sends the sound to the speakers.

To make sure the data stays synchronized, we implemented a simple handshake system. When a module finishes processing a sample, it sends a `valid` signal to the next module in the chain. The next module waits for this signal before reading the input, processing it, and passing it on. This ensures that every effect processes the correct data at the right time.

The daisy chain architecture implemented in the Top module can be visualized in Figure 4.

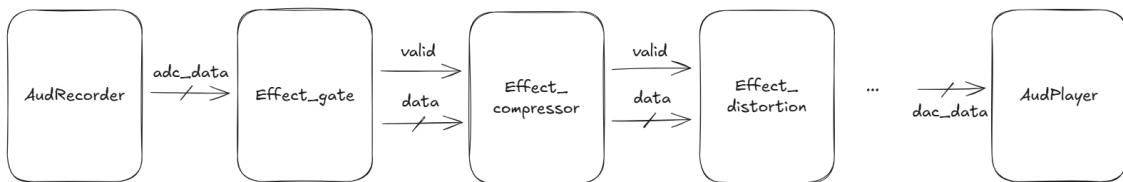


Figure 4: Effect Chain

4.1.3 SRAM Time-Multiplexing

There are multiple modules (Delay, Loop0, Loop1) that need access to SRAM to save and read data; however, there is only a single port that connects to the on-board SRAM chip. To resolve this, we implemented a time-multiplexing scheme controlled by the top module, and dedicated specific regions of SRAM to each module.

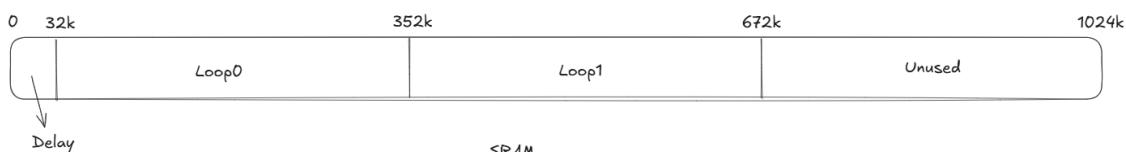


Figure 5: SRAM partition for delay and loops

Since the system clock is significantly faster than the audio sampling rate, the system has sufficient clock cycles to process multiple memory requests sequentially within a single audio sample period. We implemented a dedicated state machine to manage this access.

The process is triggered immediately after the Tremolo module finishes its calculation. The arbiter grants access in a fixed order:

1. **Delay Access (MEM_DEL):** The arbiter first connects the SRAM bus to the Delay module.
2. **Loop0 Access (MEM_LOOP0):** Once the Delay module finishes and signals that it is valid, the bus is handed over to the first Loop module.
3. **Loop1 Access (MEM_LOOP1):** Upon completion of the first loop, the bus is granted to the second Loop module.
4. **Idle (MEM_IDLE):** Finally, the system returns to the idle state until the next audio sample arrives.

To implement this, the Top module uses multiplexers to route the address, data, and write-enable signals. This ensures that only the active module is driving the SRAM interface at any specific time, preventing data corruption.

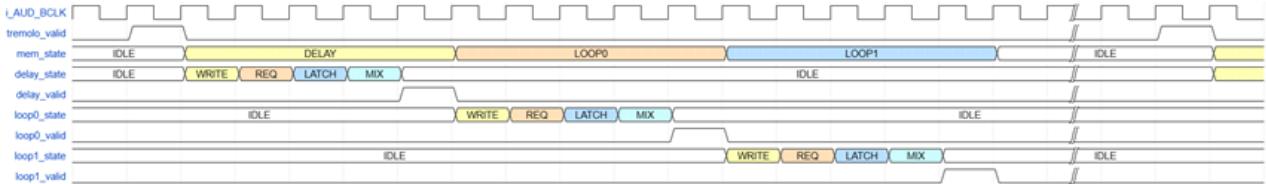


Figure 6: Memory control

4.1.4 User Interface

The user interface utilizes the on-board LEDs and 7-segment display to provide real-time feedback on the system's status and configuration. The visual feedback changes dynamically depending on whether the system is in Play mode or Parameter Setting mode.

The green LEDs (LEDG) serve as the primary status indicators for the finite state machine. They visually distinguish between the normal pass-through mode (S_PLAY), the recording phases for Loop 0 and Loop 1, and the active playback of the loops. This allows the user to confirm the current operational state at a glance.

The red LEDs (LEDR) function in two distinct modes:

- **In Play Mode:** The Red LEDs display the status of the effect bypass switches. If an LED is lit, the corresponding effect (e.g., Distortion or Delay) is enabled.
- **In Set Mode:** The Red LEDs function as a cursor. A single LED lights up to indicate which specific effect is currently selected for parameter adjustment (mapped to switches 17-15).

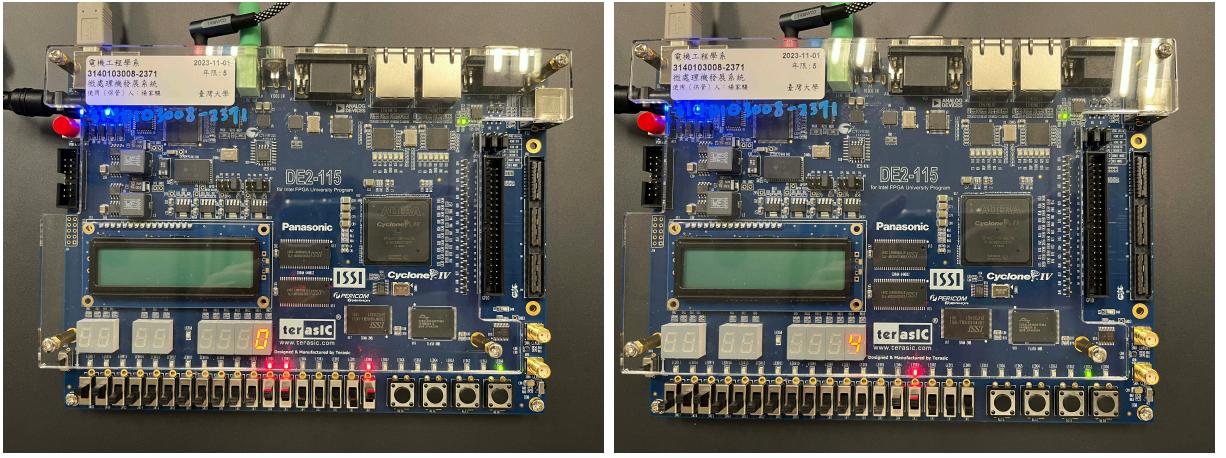


Figure 7: LED indicators during each state

The **HEX0** display provides numerical feedback for parameter adjustments. When the system is in **S_SET** mode, this display shows the current value (ranging from 0 to 7) of the selected effect's parameter. For example, if the Delay effect is selected, the display might show the current delay time level.

Switches and LEDRs 0—6 are mapped to the gate, compressor, distortion, bass EQ, treble EQ, tremolo, and delay modules, respectively.

4.2 Gate Pedal

The noise gate pedal is often overlooked due to its lack of dramatic audible effects; however, it plays a crucial role in suppressing background noise and eliminating unwanted buzzing to produce a clean signal.

We implement the noise gate using a hard-clipping-based thresholding algorithm, which effectively mutes all input signals whose amplitudes fall below a user-configurable threshold.

Let $x[n]$ denote the signed 16-bit input audio sample at time n , and let $y[n]$ denote the output sample. Let T be the user-defined threshold determined by the control level.

When the noise gate is enabled, the output signal is given by

$$y[n] = \begin{cases} 0, & |x[n]| < T \\ x[n], & |x[n]| \geq T \end{cases}$$

When the noise gate is disabled (bypass mode), the output is

$$y[n] = x[n].$$

4.3 Compressor Pedal

A compressor effects pedal evens out the dynamics of a signal. It can be used to align very loud and/or very quiet parts, smooth out attack sounds, give a signal a clean boost or add

sustain.

We implement the compressor using a simple hard-knee compression model with a fixed compression ratio applied above a user-defined threshold, followed by a level-dependent makeup gain.

Let $x[n]$ denote the signed 16-bit input audio sample at time n , and let $y[n]$ denote the output sample. Let T be the compression threshold determined by the control level, and let G denote the makeup gain factor.

The compressed magnitude of a signal is given by

$$|x_c[n]| = \begin{cases} T + \frac{|x[n]| - T}{4}, & |x[n]| > T \\ |x[n]|, & |x[n]| \leq T \end{cases}$$

The compressed signal is then amplified by the makeup gain and restored to its original sign

$$y[n] = \begin{cases} -G \cdot |x_c[n]|, & x[n] < 0 \\ G \cdot |x_c[n]|, & x[n] \geq 0 \end{cases}$$

4.4 Distortion Pedal

The distortion effect works by modifying the input signal based on amplitude thresholds to create a harmonically enriched sound.

Common distortion functions are continuous functions suitable for analog signal processing, such as $y[n] = 1/(1 + |x[n]|)$. These are not suitable for digital signal processing. Therefore, we use a piecewise function to process the signal. The distortion module has an adjustable parameter called *level*, which determines the maximum threshold $dist[0]$. After determining the maximum threshold, it is divided into 12 segments $dist[0 : 11]$. The relationship between input and output signals is as follows:

$$y[n] = \begin{cases} \text{sign}(x[n])dist[0], & |x[n]| \geq dist[0] \\ \text{sign}(x[n])dist[i], & |x[n]| \geq dist[i] \text{ and } |x[n]| < dist[i - 1] \forall i \in [1, 11] \\ 0, & \text{otherwise} \end{cases}$$

Since a total of 12 comparisons are required, binary search is employed in the implementation to reduce the critical path from 12 comparisons to $\lceil \lg(12) \rceil = 4$ comparisons.

The input-output relationship is illustrated in the figure below (Note: The amplitude has been normalized.). Signals exceeding the maximum threshold are hard-clipped, the intermediate region exhibits a stepped piecewise function, and signals below the minimum threshold pass through unmodified.

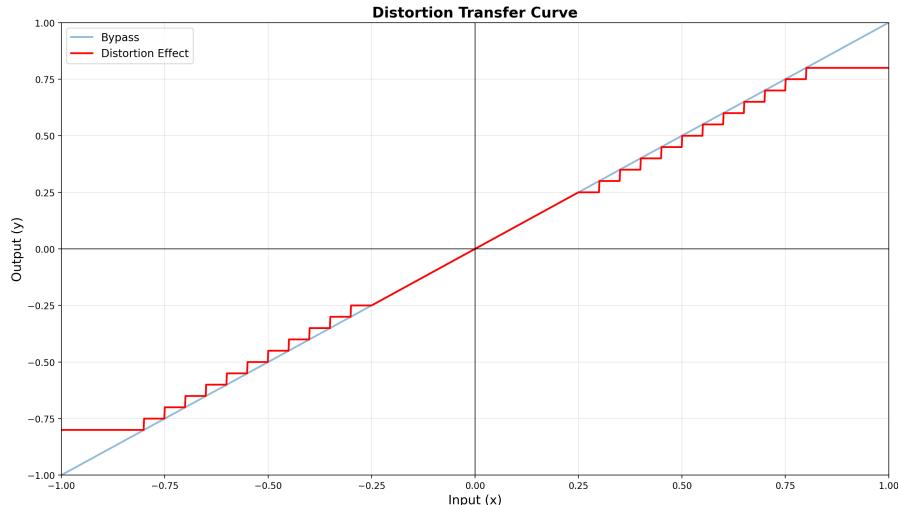


Figure 8: Distortion transfer curve

The figure below illustrates the output waveform when a sine wave is processed through the distortion effect.

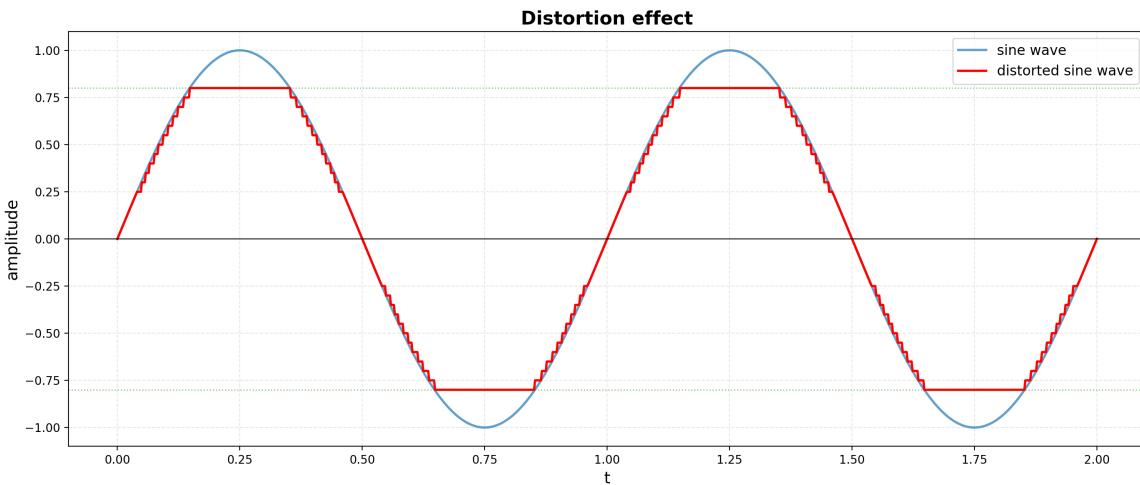


Figure 9: Distorted sine wave

4.5 Equalizer

An equalizer (EQ) pedal adjusts the frequency spectrum of an audio signal by boosting or attenuating selected frequency bands. This allows tonal shaping, such as emphasizing low-frequency bass components or reducing high-frequency treble content.

To simplify user operation, the equalizer provides independent bass and treble controls ranging from -12 dB to $+9\text{ dB}$ in 3 dB increments. The EQ is implemented as a cascade of two second-order filters: a low-shelf filter for bass adjustment followed by a high-shelf filter for treble adjustment.

Both filters are realized using biquad filter structures. The filter coefficients are precomputed based on the sampling rate, corner frequency, quality factor Q , and gain, and are selected through lookup tables according to the user-defined control levels.

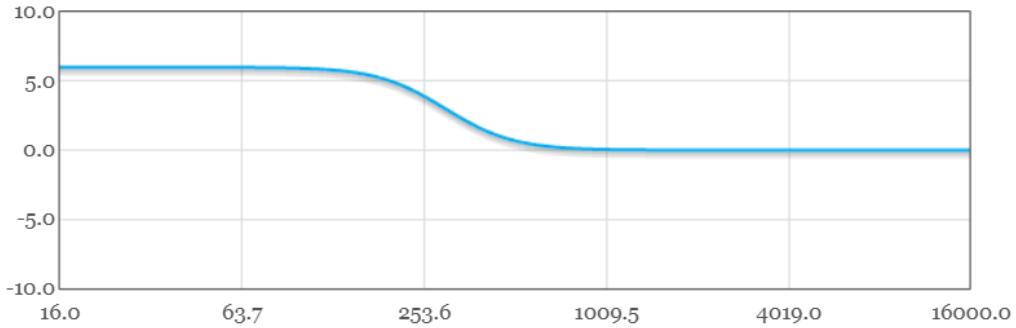


Figure 10: Low-shelf filter with a corner frequency of 250 Hz and a gain of 6 dB.

Each biquad filter is described by the difference equation

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] - b_1y[n-1] - b_2y[n-2],$$

where $x[n]$ is the input sample and $y[n]$ is the output sample.

The overall equalizer output is obtained by cascading the bass and treble filters:

$$y_{\text{bass}}[n] = \mathcal{B}_{\text{low-shelf}}\{x[n]\},$$

$$y[n] = \mathcal{B}_{\text{high-shelf}}\{y_{\text{bass}}[n]\},$$

where $\mathcal{B}_{\text{low-shelf}}(\cdot)$ and $\mathcal{B}_{\text{high-shelf}}(\cdot)$ denote the biquad operations with coefficients corresponding to the selected bass and treble gain levels.

All filter coefficients are represented in fixed-point Q4.28 format. Intermediate results are accumulated with extended precision and scaled back to 16-bit signed output samples with saturation to prevent overflow.

4.6 Tremolo Pedal

Tremolo effect is rather simple, just multiplying input signal with an LFO wave. In this project, for simplicity, we implemented a triangular wave.

$$y[n] = x[n] \times \text{tri}[n]$$

$$\text{tri}[n] = \begin{cases} \text{tri}[n-1] - \text{step}, & n \leq \frac{T}{2} \\ \text{tri}[n-1] + \text{step}, & n > \frac{T}{2} \end{cases}$$

The value of the triangular wave is represented in fixed-point Q8.24 format, and the result of multiplication is accumulated with extended precision and scaled back to 16-bit signed output samples with saturation check.

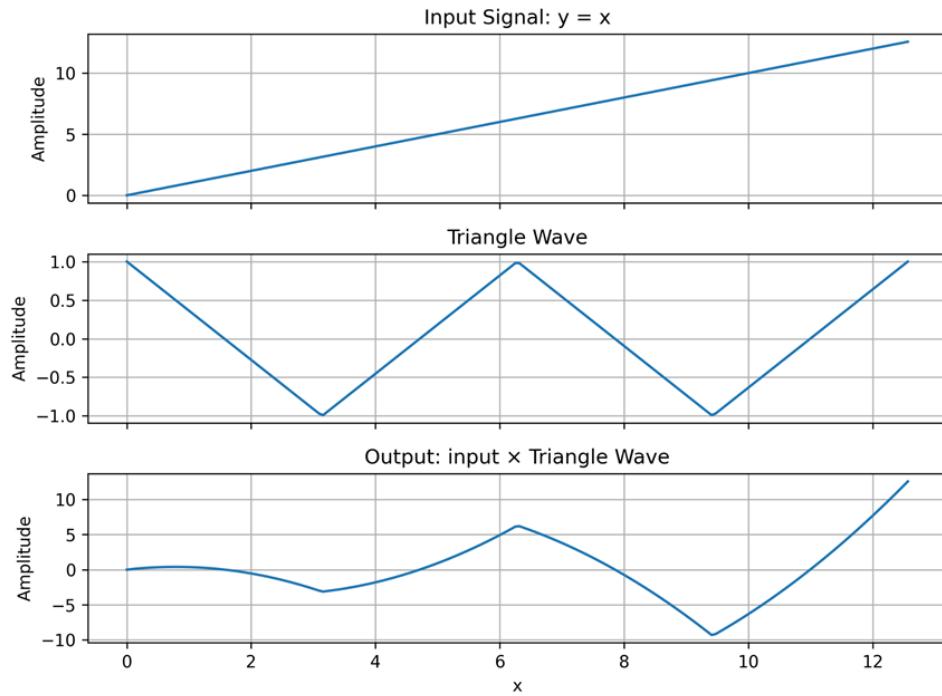


Figure 11: Demonstration of tremolo effect with linear input

4.7 Delay Pedal

Delay module's FSM consists of five states: `S_IDLE`, `S_WRITE`, `S_READ_REQ`, `S_READ_LATCH`, and `S_MIX`. When receiving a sample and valid signal from the previous module, it transitions from `S_IDLE` to `S_WRITE`, where the write enable signal is asserted, the SRAM write address is output, and the sample is provided as the data to be stored in SRAM. Next, it enters `S_READ_REQ` and `S_READ_LATCH`, where the write enable is deasserted and the SRAM read address is output, waiting one cycle to retrieve the SRAM data. Finally, it enters `S_MIX`, where it decides whether to output the sample directly bypassed from the previous module or mixed with the delayed sample from SRAM based on whether this module is enabled. At this point, it outputs the valid signal and final sample to the next module, then returns to `S_IDLE` to wait for the next processing cycle.

We designed the write address pointer (`write_ptr`) to reset to zero on reset, and subsequently increment by one each time a valid signal is received from the previous module. When it reaches the maximum value, it wraps back to zero, cyclically writing data into SRAM. The read address pointer (`read_ptr`) always lags behind `write_ptr` by a fixed offset value, which can be adjusted through `i_level` to ensure a consistent delay time. We allocated 32,000 samples of SRAM for the Delay module, which can achieve a maximum delay effect of one second.

The finite state machine of the delay module is shown in Figure 12.

4.8 Loop Pedal

Loop module is similar to the Delay module, with its FSM also having five states: `S_IDLE`, `S_WRITE`, `S_READ_REQ`, `S_READ_LATCH`, and `S_MIX`. Additionally, it determines whether the Loop

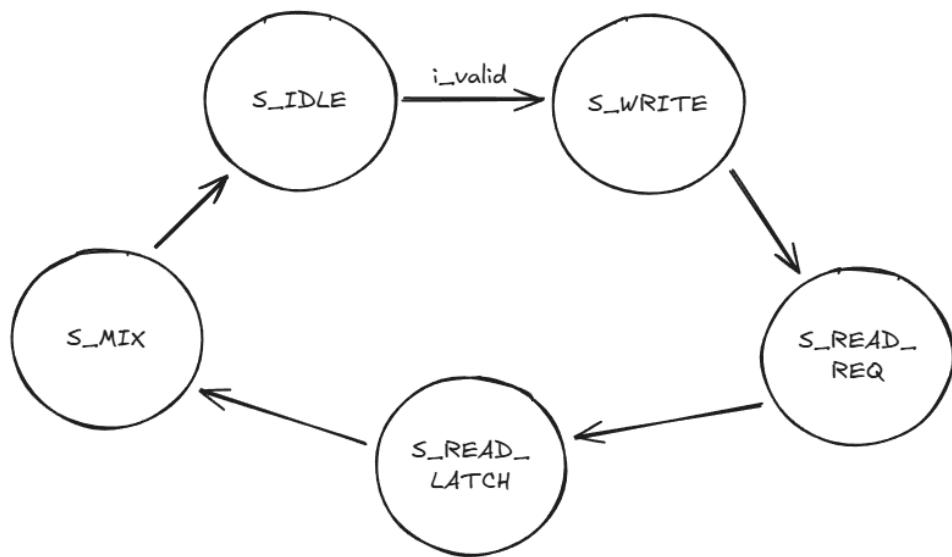


Figure 12: Finite state machine of the delay module

module operates in UNENABLED, RECORD, or PLAY state based on `i_state`, which is provided by the Top module. When `i_state = UNENABLED`, neither SRAM write nor read operations are performed, and the audio sample output is directly bypassed from the previous module. When `KEY_1` is pressed and `i_state = RECORD`, the write operation is enabled. Similar to the Delay module, `write_ptr` increments by one each time a valid signal is received from the previous module, and during `S_WRITE`, the audio sample and its write address are output. Simultaneously, the period parameter also increments by one. When `KEY_1` is pressed again and `i_state = PLAY`, the period remains fixed, and `read_ptr` continuously cycles between the start address and period, repeatedly retrieving samples stored in SRAM and adding them to the samples from the previous module before output. We can adjust the mixing ratio between the original input and looped samples through `i_level`.

5 Screenshot of the Fitter Summary

Fitter Summary	
Fitter Status:	Successful - Wed Dec 24 15:11:07 2025
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115
Top-level Entity Name	DE2_115
Family	Cyclone IV E
Device	EP4CE11F29C7
Timing Models	Final
Total logic elements	2,826 / 114,480 (2 %)
Total combinational functions	2,727 / 114,480 (2 %)
Dedicated logic registers	691 / 114,480 (< 1 %)
Total registers	691
Total pins	518 / 529 (98 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	48 / 532 (9 %)
Total PLLs	1 / 4 (25 %)

6 Screenshot of the Timing Analyzer

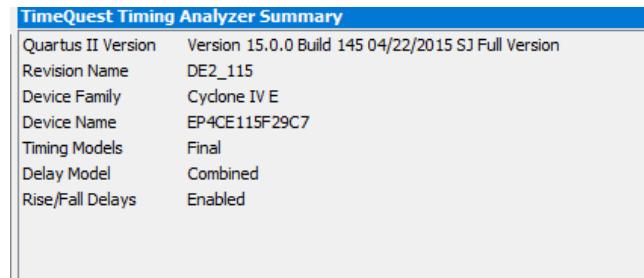


Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Filter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - Slow 1200mV 85C Model
 - Fmax Summary
 - Timing Closure Recommendations**
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - Slow 1200mV OC Model
 - Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - Multicorner Datasheet Report Summary

Timing Closure Recommendations

This design contains failing setup paths with a worst-case slack of -17.700 ns. Run [Report Timing Closure Recommendation](#) in the table below.

Top Falling Paths [Hide details]

Slack	From	To	Recommendations
1.-17.700	Top top0 state_EQ[1][0]	Top top0 Effect_E...[0]~ Duplicate_1	Report recommendations for this path
2.-17.697	Top top0 state_EQ[1][0]	Top top0 Effect_E...[0]~ Duplicate_1	Report recommendations for this path
3.-17.697	Top top0 state_EQ[1][0]	Top top0 Effect_E...[0]~ Duplicate_1	Report recommendations for this path
4.-17.695	Top top0 state_EQ[1][0]	Top top0 Effect_E...[0]~ Duplicate_1	Report recommendations for this path
5.-17.694	Top top0 state_EQ[1][0]	Top top0 Effect_E...[0]~ Duplicate_1	Report recommendations for this path

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Filter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - Slow 1200mV 85C Model
 - Fmax Summary
 - Timing Closure Recommendations**
 - Setup Summary**
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - Slow 1200mV OC Model
 - Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - Multicorner Datasheet Report Summary

Slow 1200mV 85C Model Setup Summary

	Clock	Slack	End Point TNS
1	AUD_BCLK	-17.700	-512.390
2	pl0 altpll_0 sd1 pl17 ck[1]	-4.211	-121.039
3	pl0 altpll_0 sd1 pl17 ck[0]	79.614	0.000

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Fitter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - > Slow 1200mV 8SC Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - > Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - > Slow 1200mV OC Model
 - > Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - > Multicorner Datasheet Report Summary

Slow 1200mV 8SC Model Minimum Pulse Width Summary

	Clock	Slack	End Point TNS
1	AUD_BCLK	-3.210	-1701.723
2	CLOCK_50	9.819	0.000
3	plio atpl_0 sd1 pl7 clk[0]	41.375	0.000
4	plio atpl_0 sd1 pl7 clk[1]	4999.619	0.000

Table of Contents

Slow 1200mV 8SC Model Settings - AUD_BCLK

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Fitter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - > Slow 1200mV 8SC Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - > Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - > Slow 1200mV OC Model
 - > Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - > Multicorner Datasheet Report Summary

Table of Contents

Slow 1200mV 8SC Model Settings - AUD_BCLK

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Fitter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - > Slow 1200mV 8SC Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - > Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - > Slow 1200mV OC Model
 - > Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - > Multicorner Datasheet Report Summary

Table of Contents

Slow 1200mV 8SC Model Settings - AUD_BCLK

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Fitter
- > Flow Messages
- > Flow Suppressed Messages
- > Assembler
- > TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - > Slow 1200mV 8SC Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - > Worst-Case Timing Paths
 - Datasheet Report
 - Metastability Summary
 - > Slow 1200mV OC Model
 - > Fast 1200mV OC Model
 - Multicorner Timing Analysis Summary
 - > Multicorner Datasheet Report Summary

6 Screenshot of the timing analyzer

6 Screenshot of the Timing Analyzer

Slow 1200mV 0C Model Setup Summary				
	Clock	Slack	End Point TNS	
1	AUD_BCLK	-15.998	-4681.718	
2	pll0 altpll_0 sd1 pl7 clk[1]	-3.715	-106.672	
3	pll0 altpll_0 sd1 pl7 clk[0]	79.875	0.000	

Table of Contents

- Summary
- Parallel Compilation
- SDC File List
- Clocks
- Slow 1200mV 85C Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary**
- Worst-Case Timing Paths
 - Setup: AUD_BCLK
 - Setup: 'pll0|altpll_0|sd1|pl7|clk[1']'
 - Setup: 'pll0|altpll_0|sd1|pl7|clk[0']'
 - Hold: 'pll0|altpll_0|sd1|pl7|clk[1']'
 - Hold: 'pll0|altpll_0|sd1|pl7|clk[0']'
 - Hold: 'AUD_BCLK'
 - Minimum Pulse Width: 'AUD_BCLK'**
 - Minimum Pulse Width: 'CLOCK_50'
 - Minimum Pulse Width: 'pll0|altpll_0|sd1|pl7|clk[1']'
 - Minimum Pulse Width: 'pll0|altpll_0|sd1|pl7|clk[0']'
- Datasheet Report
- Metastability Summary
- Slow 1200mV 0C Model
 - Fmax Summary
 - Setup Summary**
 - Hold Summary
 - Recovery Summary
 - Removal Summary

Table of Contents				Slow 1200mV OC Model Minimum Pulse Width Summary		
				Clock	Slack	End Point TNS
			Timing Closure Recommendations			
			Setup Summary			
			Hold Summary			
			Recovery Summary			
			Removal Summary			
			Minimum Pulse Width Summary			
>			Worst-Case Timing Paths			
			Setup: 'AUD_BCLK'			
			Setup: 'pll0[altpl_0 sd1 pll7 clk[1]]'			
			Setup: 'pll0[altpl_0 sd1 pll7 clk[0]]'			
			Hold: 'pll0[altpl_0 sd1 pll7 clk[1]]'			
			Hold: 'pll0[altpl_0 sd1 pll7 clk[0]]'			
			Hold: AUD_BCLK			
			Minimum Pulse Width: 'AUD_BCLK'			
			Minimum Pulse Width: 'CLOCK_50'			
			Minimum Pulse Width: 'pll0[altpl_0 sd1 pll7 clk[1]]'			
			Minimum Pulse Width: 'pll0[altpl_0 sd1 pll7 clk[0]]'			
>			Datasheet Report			
			Metastability Summary			
>			Slow 1200mV OC Model			
			Fmax Summary			
			Setup Summary			
			Hold Summary			
			Recovery Summary			
			Removal Summary			
			Minimum Pulse Width Summary			
>			Worst-Case Timing Paths			
>			Datasheet Report			
			Metastability Summary			
>			Fast 1200mV OC Model			
			Multicorner Timing Analysis Summary			

6 Screenshot of the Timing Analyzer

6 SCREENSHOT OF THE TIMING ANALYZER

Table of Contents

- > Dashboard Report
- > Metastability Summary
- > Slow 1200mV OC Model
 - Slow Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup pll0|altpll_0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Database Report
- > Metastability Summary
- > Fast 1200mV OC Model
 - Fast Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Clock Transfers
- > Report TCCS
- > Report SGM
- > Unresolved Paths
- Messages

Table of Contents

- > Dashboard Report
- > Metastability Summary
- > Slow 1200mV OC Model
 - Slow Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Database Report
- > Metastability Summary
- > Fast 1200mV OC Model
 - Fast Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Clock Transfers
- > Report TCCS
- > Report SGM
- > Unresolved Paths
- Messages

Table of Contents

- > Dashboard Report
- > Metastability Summary
- > Slow 1200mV OC Model
 - Slow Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Database Report
- > Metastability Summary
- > Fast 1200mV OC Model
 - Fast Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Clock Transfers
- > Report TCCS
- > Report SGM
- > Unresolved Paths
- Messages

Table of Contents

- > Dashboard Report
- > Metastability Summary
- > Slow 1200mV OC Model
 - Slow Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Database Report
- > Metastability Summary
- > Fast 1200mV OC Model
 - Fast Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Setup p0|alt0|sd1|pil7|dk[1]
 - Hold p0|alt0|sd1|pil7|dk[0]
 - Min. Pulse Width: CLOCK_SF
 - Min. Pulse Width: AUD_BCLK
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[1]
 - Min. Pulse Width: p0|alt0|sd1|pil7|dk[0]
- > Clock Transfers
- > Report TCCS
- > Report SGM
- > Unresolved Paths
- Messages

Fast 1200mV OC Model Setup Summary					
	Clock	Slack	End Point TNS		
1	AUD_BCLK	-8.234	-2199.903		
2	pll0 altpll_0 sd1 pil7 dk[1]	-1.703	-48.074		
3	pll0 altpll_0 sd1 pil7 dk[0]	81.500	0.000		

Fast 1200mV OC Model Minimum Pulse Width Summary

	Clock	Slack	End Point TNS
1	AUD_BCLK	-3.000	-1257.352
2	CLOCK_50	9.400	0.000
3	pll0 altpll_0 sd1 pll7 clk[0]	41.446	0.000
4	pll0 altpll_0 sd1 pll7 clk[1]	4999.780	0.000

Fast 1200mV OC Model Setup: 'AUD_BCLM'

Fast 1200mV DC Model Setup: 'AUD_BCLK'

Fast 1200mV OC Model Setup: 'AUD_BCLK'

6 Screenshot of the Timing Analyzer

Fast 1200mV OC Model Minimum Pulse Width: 'AUD_BCLK'							
Slack	Actual Width	Required Width	Type	Clock	Clock Edge		Target
1	-3.000	4.000	Port Rate	AUD_BCLK	Rise	AUD_BCLK	
2	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@bit_cnt_0]	
3	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@bit_cnt_1]	
4	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@bit_cnt_2]	
5	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@bit_cnt_3]	
6	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@bit_cnt_4]	
7	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@addr[reg_4]_pre_r]	
8	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_pre_r]	
9	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[10]	
10	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[11]	
11	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[12]	
12	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[13]	
13	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[14]	
14	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[15]	
15	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@shift[reg_4]]_reg[16]	
16	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[2]	
17	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[3]	
18	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[4]	
19	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[5]	
20	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[6]	
21	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[7]	
22	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[8]	
23	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudPlayer/player@0[reg_4]]_reg[9]	
24	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[0]	
25	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[10]	
26	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[11]	
27	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[12]	
28	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[13]	
29	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[14]	
30	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[15]	
31	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[16]	
32	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[17]	
33	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[18]	
34	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[19]	
35	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[1]	
36	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[2]	
37	-1.000	2.000	Min Period	AUD_BCLK	Rise	Top:top[AudRecorder/recorder@0[addr_0]]_reg[3]	

6 Screenshot of the Timing Analyzer

Fast 1200mV DC Model Minimum Pulse Width: 'AUD_BCLK'							
Slack	Actual Width	Required Width	Type	Clock	Clock Edge	Target	
34	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[9]
35	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[1]
36	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[2]
37	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[3]
38	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[4]
39	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[5]
40	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[6]
41	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[7]
42	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[8]
43	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[9]
44	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[10]
45	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[11]
46	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[12]
47	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[13]
48	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[14]
49	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[15]
50	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[16]
51	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[17]
52	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[18]
53	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[19]
54	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[10]
55	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[10]
56	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[11]
57	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[12]
58	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[13]
59	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[14]
60	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[15]
61	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[16]
62	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[17]
63	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[18]
64	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[19]
65	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[15]
66	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[16]
67	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[17]
68	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[18]
69	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[19]
70	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[0]AudRecorder:record0[address_1'[10]

Fast 1200mV OC Model Minimum Pulse Width: AUD_BCLK							
Sack	Actual Width	Required Width	Type	Clock	Clock Edge		Target
64	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][4]
65	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][5]
66	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][6]
67	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][7]
68	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][8]
69	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][9]
70	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][10]
71	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][10]
72	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][11]
73	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][12]
74	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][13]
75	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][14]
76	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][15]
77	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][16]
78	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][17]
79	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][18]
80	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][19]
81	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][19]
82	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][19]
83	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][17]
84	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][18]
85	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_data_][19]
86	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_state_].s_CLEAR
87	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_state_].s_IDLE
88	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_state_].s_RECORD
89	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]AudRecorder:record1[_state_].s_RECEIVED
90	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][0]
91	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][10]
92	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][11]
93	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][12]
94	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][13]
95	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][14]
96	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][15]
97	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][16]
98	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][17]
99	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][18]
100	-1.000	1.000	2.000	Min Period	AUD_BCLK	Rise	Top:[top0]Effect.Compressor:compress1[_data][19]

Multicorner Timing Analysis Summary						
	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	Worst-case Slack	-17.700	0.174	N/A	N/A	-3.210
1	AUD_BCLK	-17.700	0.201	N/A	N/A	-3.210
2	CLOCK_50	N/A	N/A	N/A	N/A	9.400
3	pll0[altpll_0]sd1[pll7][clk[0]	79.614	0.183	N/A	N/A	41.375
4	pll0[altpll_0]sd1[pll7][clk[1]	-4.211	0.174	N/A	N/A	4999.619
2	Design-wide TNS	-533.429	0.0	0.0	0.0	-1701.723
1	AUD_BCLK	-5212.390	0.000	N/A	N/A	-1701.723
2	CLOCK_50	N/A	N/A	N/A	N/A	0.000
3	pll0[altpll_0]sd1[pll7][clk[0]	0.000	0.000	N/A	N/A	0.000
4	pll0[altpll_0]sd1[pll7][clk[1]	-121.039	0.000	N/A	N/A	0.000

Unconstrained Paths			
	Property	Setup	Hold
1	Illegal Clocks	0	0
2	Unconstrained Clocks	0	0
3	Unconstrained Input Ports	34	34
4	Unconstrained Input Port Paths	1432	1432
5	Unconstrained Output Ports	63	63
6	Unconstrained Output Port Paths	1245	1245

7 Problems Encountered

1. The first challenge we encountered in this project was designing an effective way to integrate relatively complex control and display functions onto a single FPGA board. By identifying the key features and assigning dedicated buttons to them, while repurposing unused buttons in each operating mode for secondary functions, we were able to make the control of the effect pedal intuitive and user-friendly.
2. One design that took a long time to come up with was the memory-multiplexing. Since there are multiple modules that needed to access SRAM, some kind of selection method was needed. Fortunately, since the SRAM access time was very quick, and the modules that needed SRAM are sequentially placed, it is possible to assign SRAM control through time without conflicts. Implementation would be hard if we let each module access SRAM individually or if the modules were running in parallel, which would have required a complicated controller to stop them from fighting over the memory at the same time.