FPGA TR-909 Drum Machine Project Proposal

Jordan Lin (kl3758), Noel Gomez (ng2703), Aaron Zhu (azz2111)
CSEE4840 Spring 2026

1. Project Introduction

The goal of this project is to implement a hardware drum machine inspired by the Roland TR-909 using an FPGA. The system will allow a user to program rhythmic patterns using a visual beatgrid and step sequencer interface, select drum samples from a preloaded 909 sample bank, choose certain parameters for each sample such as pitch and volume, and play the resulting beat in real time out of a speaker. We expect to use a 64 or 128 step beat grid matched to 4 or 8 measures and allow the user to loop the full grid or a section of it in a playback mode.

The system combines hardware sequencing logic implemented in Verilog on the FPGA with software-based UI logic written in C to manage the on-screen beat grid and keyboard controls, as well as keep everything in time with the tempo chosen by the user. We would like to incorporate different ways of making beats such as manually MIDI mapping samples on the beatgrid vs. placing a sample on the grid in real time in a record mode (quantize on or off). This would require at least three modes of operation-sequencing, recording, and playing.

2. Systems Overview

The FPGA will handle:

- step sequencing
- audio sample playback
- timing and BPM control
- pattern memory
- mixing multiple drum sounds

The software will handle:

- graphical beat grid interface
- keyboard input
- communication with FPGA registers
- signal processing of drum samples

3. Anticipated Data Flow

User Input
↓
Software Beat Grid
↓
Write Pattern Data
↓
FPGA Pattern Memory

↓
Step Sequencer
↓
Sample Trigger
↓
Sample Playback
↓
Mixer
↓
Audio Output

4. Key Details

A finite state machine increments a step counter and the sequencer reads from pattern memory in this fashion:
Step → 1 2 3 4 5 6 7 8 ...
Kick X . . . X . . .
Snare . . X . . . X .
HiHat X X X X X X X X
(displayed by UI)

Sample selection and steps are navigated using keyboard controls.
Example:
SPACE → start/stop
Arrow → move grid cursor
Enter → toggle step
+/- → adjust BPM
TAB → open sample bank

Drum samples for kick, snare, high hats, toms, claps stored digitally in ROM. When a step triggers a drum sound, the sample playback engine outputs the waveform using a sample address counter, playback enable signal, ROM interface.

Signals must be mixed prior to output using hardware blocks for adding, gain scaling, and clipping control. Final digital signal is sent to DAC codec at 44.1kHz.

Multiple clocks will need to be managed for general system FPGA logic, BPM counter, Audio playback rate, etc.

Software will need to send signals to hardware to e.g. write pattern memory, set BPM, start/stop sequencer.

5. Expected Challenges

- synchronizing timing between sequencer and audio playback
- managing multiple simultaneous sample triggers
- maintaining real-time audio performance
- implementing efficient memory usage for samples
- toggling different use modes e.g. record live, looping

6. Additional Possible Unique Features

To make the project more interesting, additional features may include:

- quantize beat delay to create swing timing (groove control)
- effects panel e.g. roll, echo, reverb
- LED beat visualization
- random pattern generation
- Microphone input overlay