Piano Heroes

CSEE 4840: Embedded Systems

Spring 2025

Anita Bui-Martinez (adb2221)
Michael John Flynn (mf3657)
Zakiy Tywon Manigo (ztm2106)
Robel Wondwossen (rw3043)

Overview

Hardware

Software

Interface

Demo

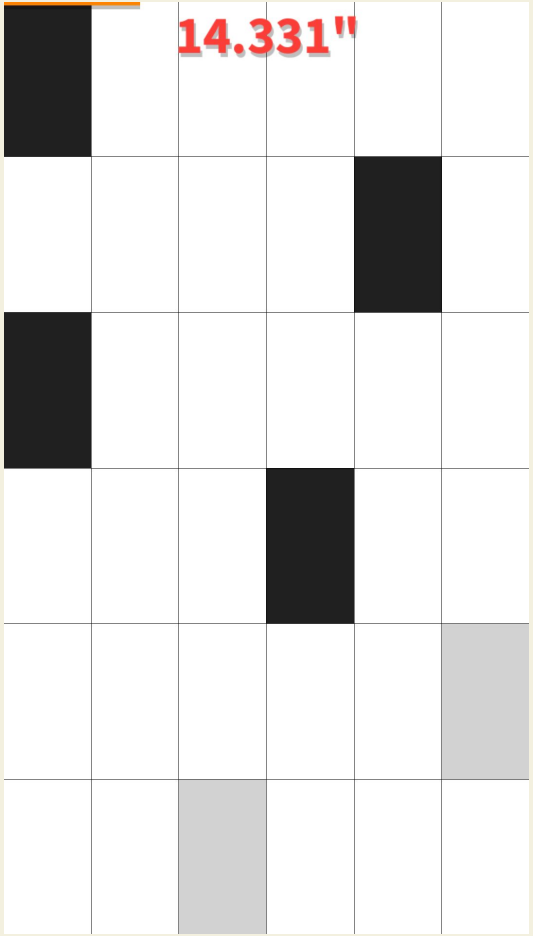
Overview

Inspiration

FPGA remake of “Piano Tiles” with real USB MIDI keyboard

Key Features:

- Two-octave support for richer gameplay
- Real-time VGA tile display + audio feedback
- Low-latency hit detection (< 1 ms)

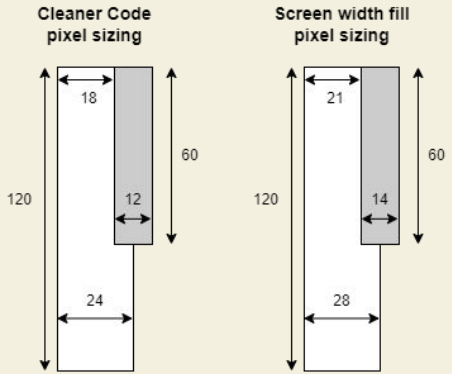


Our Goals

- Implement real-time, responsive note detection using FPGA logic.
- Provide accurate audio feedback for each key press using high-quality phase-shifted WAV samples.
- Design a visually appealing VGA interface for the falling note tiles and real-time score display.
- Achieve low latency between key press, note hit detection, and audio playback.

Hardware

Display



Game Engine:

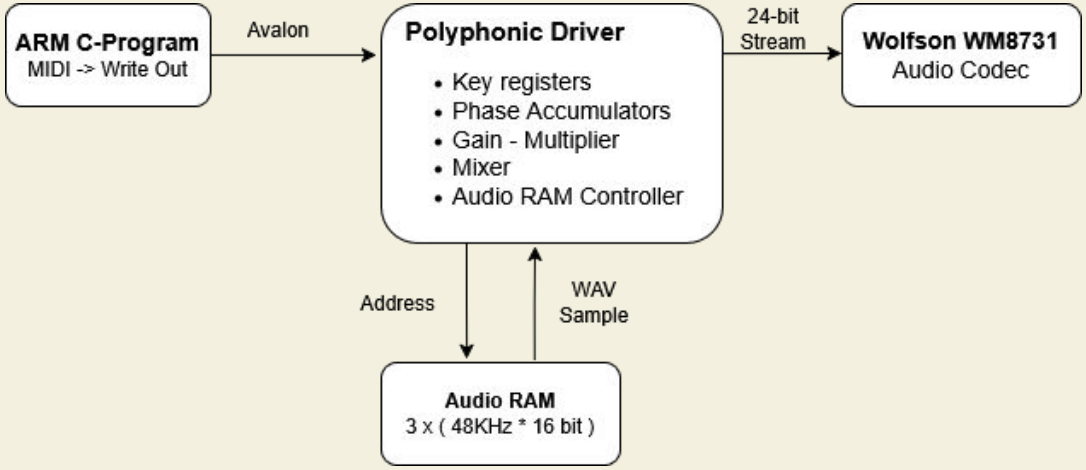
- Circular buffer of active tiles (column, spawn time, speed, color)
- Computes Y position each VGA frame
- Hit-line comparator + score counter

MIDI Capture IP:

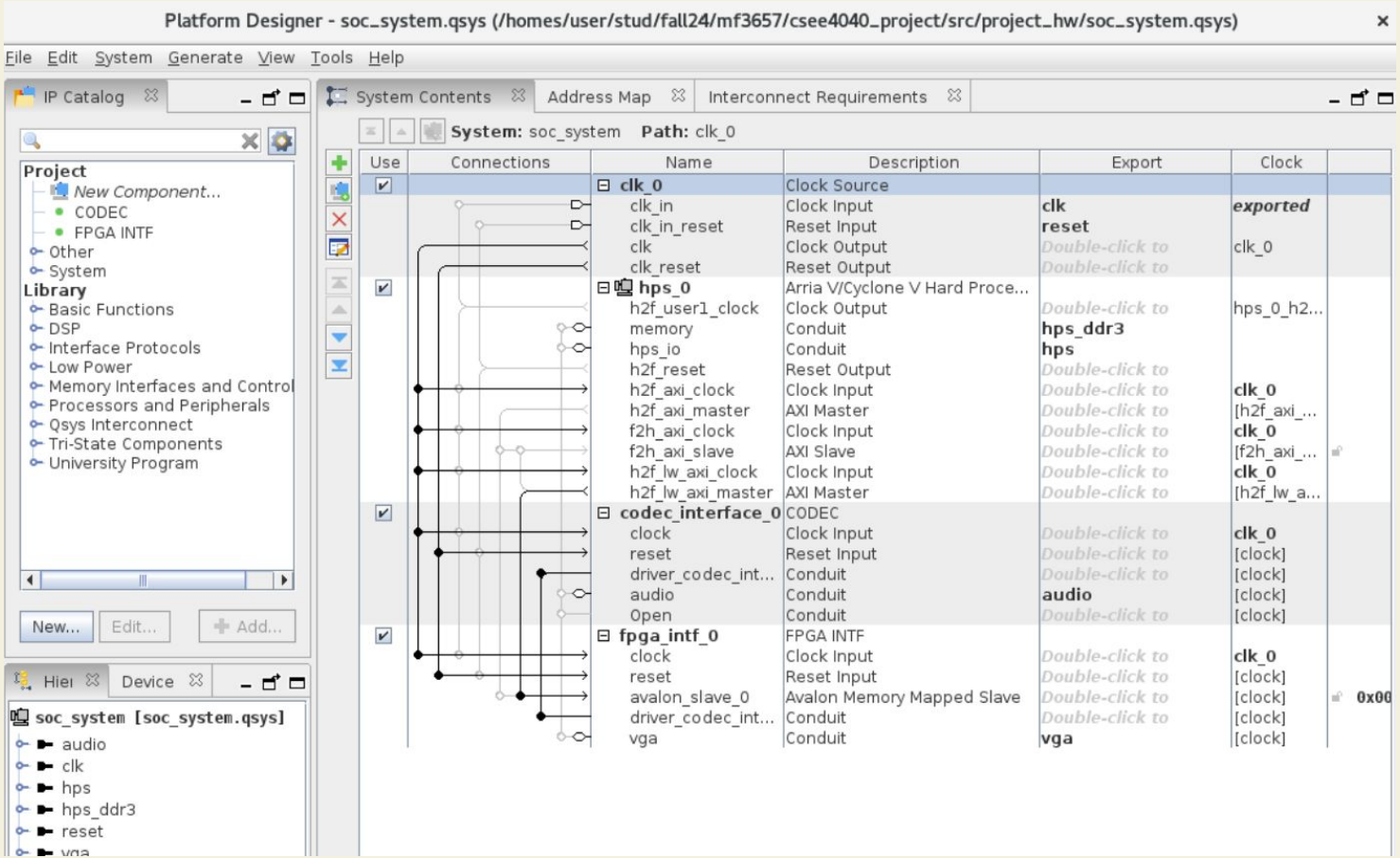
- Memory-mapped at 0xFF20_1000/1004
- Buffers 32-bit MIDI + timestamp, raises IRQ on new data

Audio

- Polyphonic Driver Peripheral
  - Up to 8 simultaneous voices (key registers)
  - Phase accumulators pitch-shift 2 base WAV samples (C3–C4)
  - Gain multipliers + mixer → single 24-bit stream
- On-Chip RAM
- Audio Codec



Audio


Platform Designer - soc_system.qsys (/homes/user/stud/fall24/mf3657/csee4040_project/src/project_hw/soc_system.qsys)
File Edit System Generate View Tools Help
IP Catalog
Project
New Component...
CODEC
FPGA INTF
Other
System
Library
Basic Functions
DSP
Interface Protocols
Low Power
Memory Interfaces and Control
Processors and Peripherals
Qsys Interconnect
Tri-State Components
University Program
New...
Edit...
Add...
Hier
Device
soc_system [soc_system.qsys]
audio
clk
hps
hps_ddr3
reset
vga
System Contents
Address Map
Interconnect Requirements
System: soc_system Path: clk_0
Use
Connections
Name
Description
Export
Clock
clk_0
Clock Source
clk_in
Clock Input
clk
exported
clk_in_reset
Reset Input
reset
clk
Clock Output
clk_0
clk_reset
Reset Output
hps_0
Arria V/Cyclone V Hard Proce...
h2f_user1_clock
Clock Output
hps_0_h2...
memory
Conduit
hps_ddr3
hps_io
Conduit
hps
h2f_reset
Reset Output
h2f_axi_clock
Clock Input
clk_0
h2f_axi_master
AXI Master
[h2f_axi_...
f2h_axi_clock
Clock Input
clk_0
f2h_axi_slave
AXI Slave
[f2h_axi_...
h2f_lw_axi_clock
Clock Input
clk_0
h2f_lw_axi_master
AXI Master
[h2f_lw_a...
codec_interface_0
CODEC
clock
Clock Input
clk_0
reset
Reset Input
[clock]
driver_codec_int...
Conduit
[clock]
audio
Conduit
audio
[clock]
Open
Conduit
[clock]
fpga_intf_0
FPGA INTF
clock
Clock Input
clk_0
reset
Reset Input
[clock]
avalon_slave_0
Avalon Memory Mapped Slave
[clock]
0x00
driver_codec_int...
Conduit
[clock]
vga
Conduit
vga
[clock]

Software

Software

- MIDI Reader Thread
  - Polls USB endpoint via libusb
  - Decodes “Note On,” timestamps with gettimeofday()
- Game State Handling
  - Tracks active tiles, score, combo, game over
  - Drives on-screen UI: start menu, difficulty select

Hardware-Software Integration

Integration

The plan was:

- Interrupt Flow
  - HPS writes MIDI
  - FPGA asserts IRQ
  - Userspace thread waits, updates game logic

Demo

Thank you!