

CSEE-4840

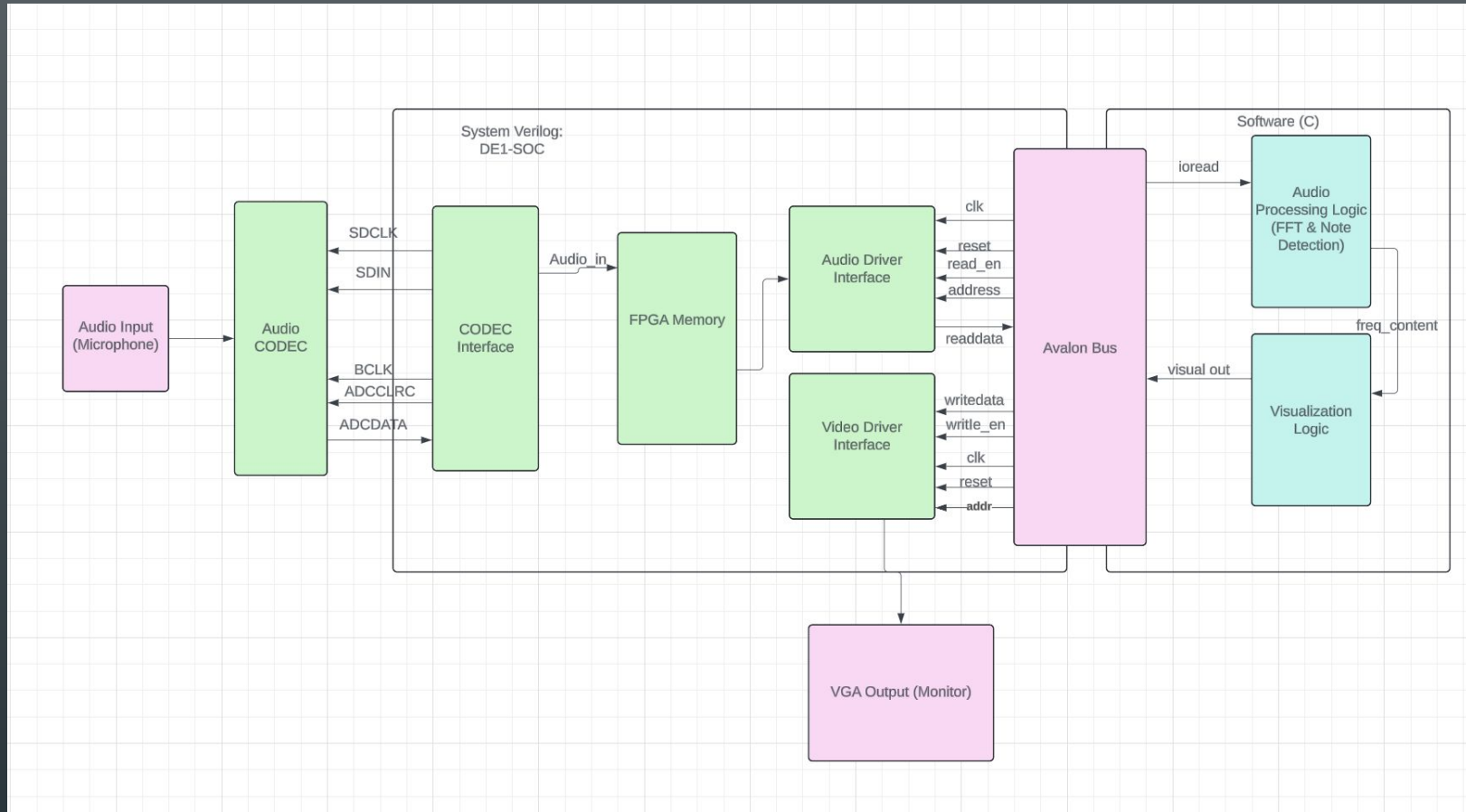
Audio Visualizer

Manas Pange, Yaagna Modi, Gaurav Agarwal, Max Lavey

Project Specifications

- The goal of our project is to capture notes from piano and then do FFT on the input data to identify the note that is being played.
- We are using HIYANCO microphone and connecting it to 'mic in' input port on FPGA board.
- Convert the analog data into the digital data and save it in an output buffer.
- The sampling rate is 48 kHz.

Block Diagram



Qsys

Platform Designer - soc_system.qsys* (/homes/user/stud/fall23/ykm2110/Downloads/hardwarefiles-20240515T00050Z-001/hardwarefiles/soc_system.qsys)

File Edit System Generate View Tools Help

IP Catalog System Contents Address Map Interconnect Requirements

System: soc_system Path: vga_driver

Use

Connections

Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
clk_0	Clock Source	clk	exported					
clk_in	Clock Input	reset	clk_0					
clk_in_reset	Reset Input	Double-click to						
clk	Clock Output	Double-click to						
clk_reset	Reset Output	Double-click to						
hps_0	Arria V10 Cyclone V Hard Proc...	hps_0_h2...						
h2f_user1_clock	Clock Output	Double-click to						
memory	Conduit	hps_ddr3						
h2f_reset	Reset Output	Double-click to						
h2f_axi_clock	Clock Input	Double-click to	clk_0					
h2f_axi_master	AXI Master	Double-click to						
f2h_axi_slave	AXI Slave	Double-click to						
h2f_axi_slave	Clock Input	Double-click to	clk_0					
h2f_axi_master	AXI Master	Double-click to						
f2h_irqp	Interrupt Receiver	Double-click to						
f2h_irqt	Interrupt Receiver	Double-click to						
avalon_slave_0	Avalon Memory Mapped Slave	Double-click to						
clock	Clock Input	Double-click to	clk_0	0x0001_0000	0x0007_ffff			
reset	Reset Input	Double-click to						
aud	Conduit	Double-click to						
fpga	Conduit	fpga						
hex1	Conduit	hex1						
hex2	Conduit	hex2						
hex3	Conduit	hex3						
hex4	Conduit	hex4						
hex5	Conduit	hex5						
key	Conduit	key						
filter_out	Conduit	Double-click to						
raw_data	Conduit	Double-click to						
filter_0	filter	Double-click to	clk_0					
clock	Clock Input	Double-click to						
reset	Reset Input	Double-click to						
filter_signal_out	Conduit	Double-click to						
filter_input	Conduit	Double-click to						
vga_driver	vga_driver	Double-click to	clk_0					
clock	Clock Input	Double-click to						
reset	Reset Input	Double-click to						
avalon_slave_0	Avalon Memory Mapped Slave	Double-click to						
vga	Conduit	Double-click to	clk_0	0x0000_0000	0x0003_ffff			

Current filter:

Messages

Type	Path	Message
2 Info Messages		
1	soc_system.hps_0	HPS Main PLL counter settings: n = 0 m = 73
1	soc_system.hps_0	HPS peripheral PLL counter settings: n = 0 m = 39

0 Errors, 0 Warnings

Generate HDL... Finish

Audio Codec

```
sound.txt
-1139
-1139
-1139
-1139
-1139
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-610
-93
-93
-93
-93
-93
-93
-93
-93
-93
-93
```

- The microphone connected to the FPGA provides analog data.
- Then the CODEC takes this data and converts it into digital data.

Verification of CODEC

- After integrating the CODEC and input analog data we connected CODEC's output to the speakers.
- By doing this we verified that the CODEC is working as expected.
- We saved the digital data to a buffer and saved the buffered data in a text file.
- Performed FFT on the data saved in txt file so as to verify that we are getting the right data and accuracy is good.

Hardware-Software Interface

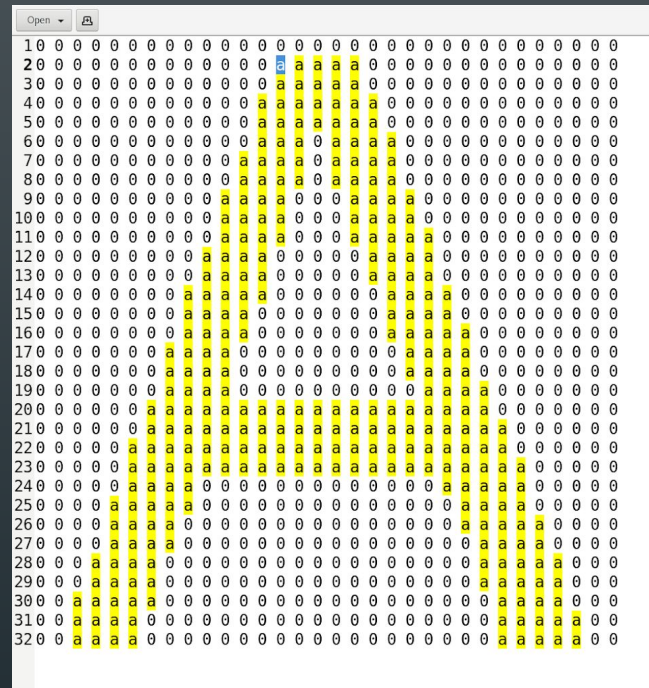
- The hardware of the DE1-SoC board interacts with the software logic to achieve its functionality.
- Writing driver or interface code, controlling memory access, and setting up peripherals are all part of this.
- Following the computation of the note detection, the software modifies the contents of the notes and frequency.
- This is done via a kernel program to the DE1-SoC board's registers.
- The VGA display is updated by the hardware once it reads the data from the registers.

FFT Algorithm

- The software will interface with the hardware and get the buffer data.
- Within the C program, audio samples are accumulated and stored in a time series array. They are then put through the FFT in software with a sampling frequency of 48kHz with 8192 bins.
- From this FFT output array, we are able to search the array for the index of the largest magnitude.
- This frequency is then plugged into a lookup table, where the corresponding note being played is understood.

VGA Interface

- We created custom sprites by drawing the pixel art for them.
- The image size is 32x32 pixels.
- These .txt files are read by Quartus to create ROM modules.



Note Detection and Display

- The VGA portion of the software, takes the identified note and determines the associated sprite, and note frequency to be sent back to the hardware.

