Autotune

CSEE W4840: Embedded Systems
Spring 2025 Presentation

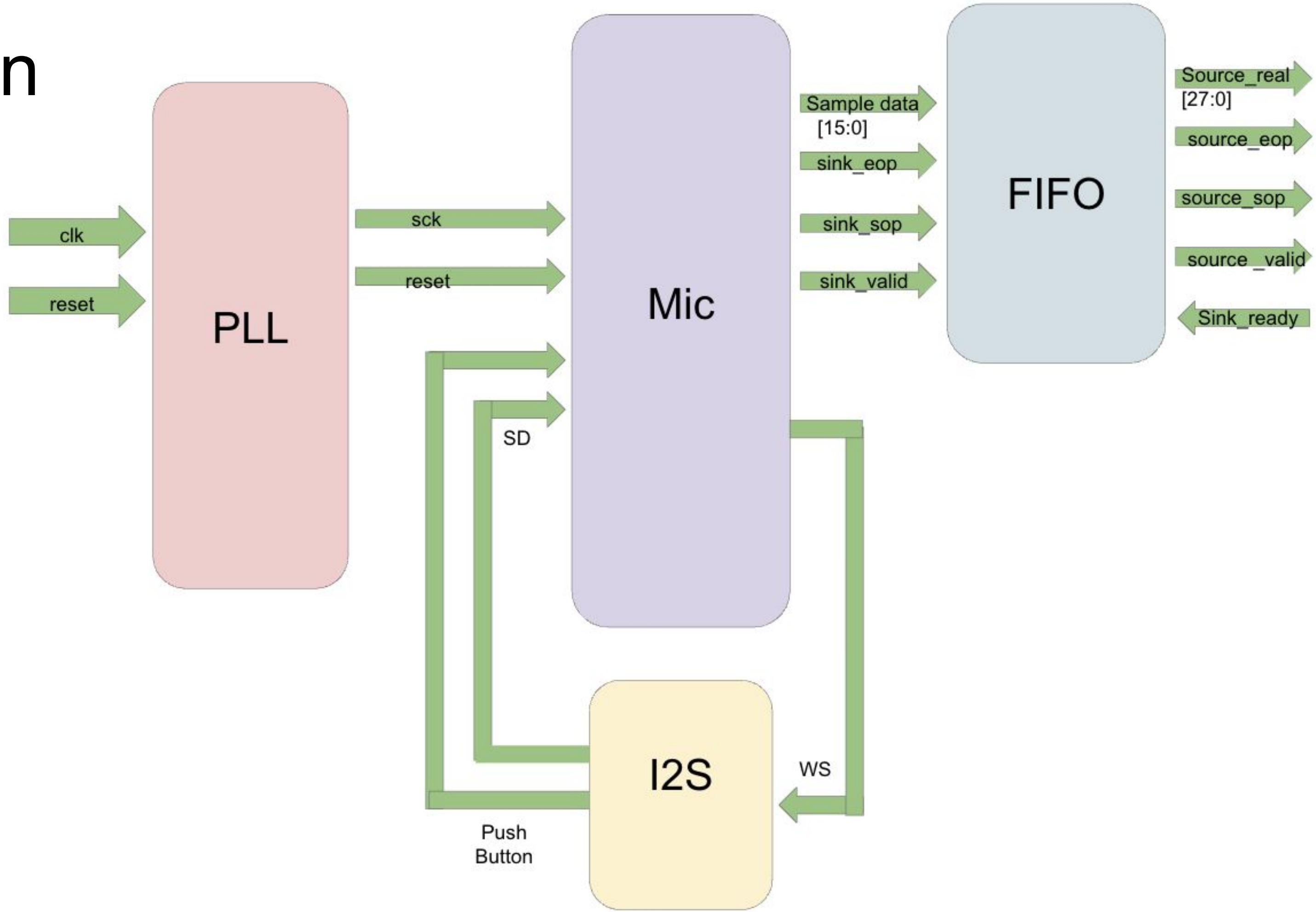
Charlie Mei (jm5912), Amanda Lee Jenkins(alj2155), Millie Chen(sc5405), Meng Fan Wang(mw3751)

Project Description

- Pitch correction system built on the DE1-SoC FPGA board, allowing users to sing into a microphone and hear autotuned playback through audio output.
- Integrates hardware-accelerated STFT, FFT/IFFT, and peak detection modules to identify and shift off-pitch frequencies to the nearest musical note.

Hardware

Audio-In



clk
reset
PLL
sck
reset
Mic
SD
Push
Button
I2S
WS
Sample data
[15:0]
sink_eop
sink_sop
sink_valid
FIFO
Source_real
[27:0]
source_eop
source_sop
source_valid
Sink_ready

Microphone

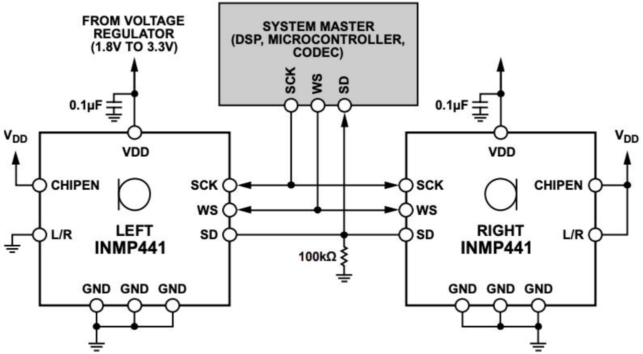
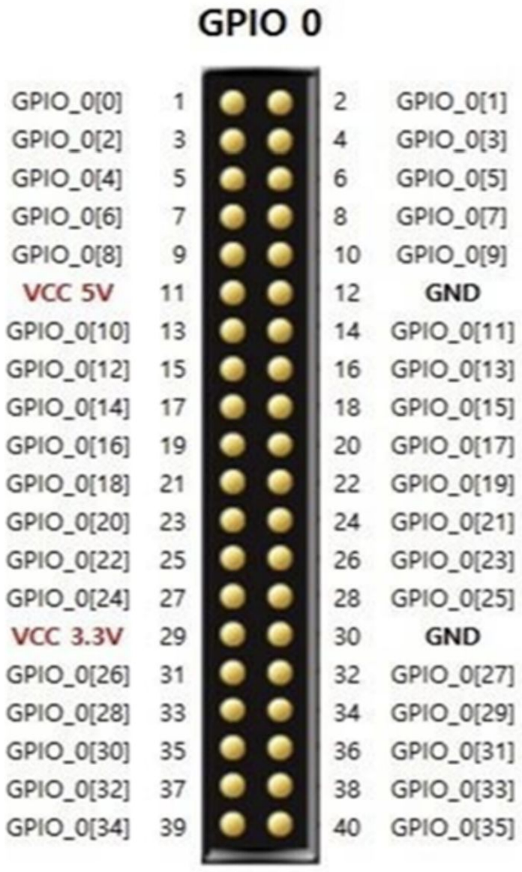


Figure 7. System Block Diagram

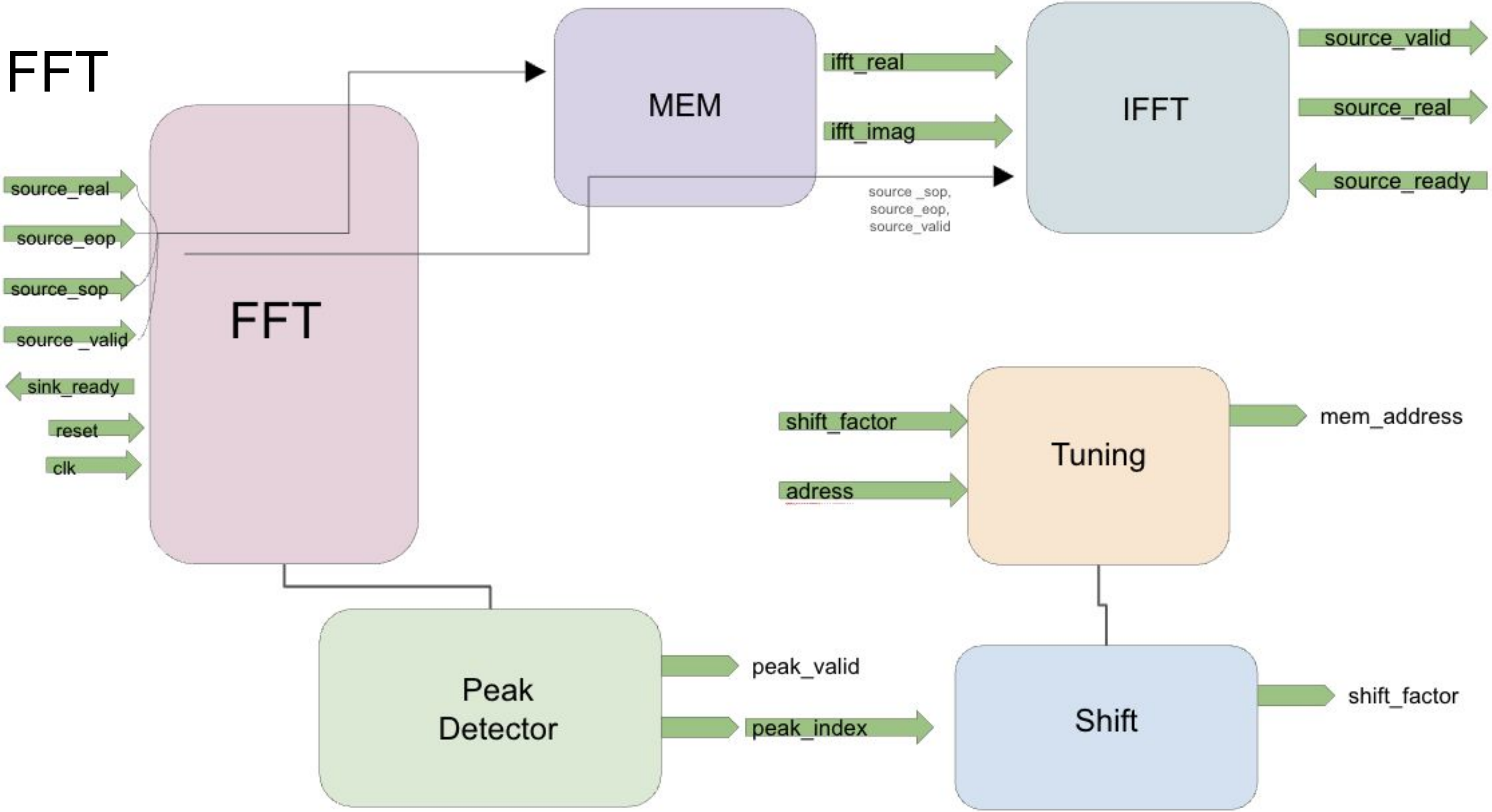
SCK

SD



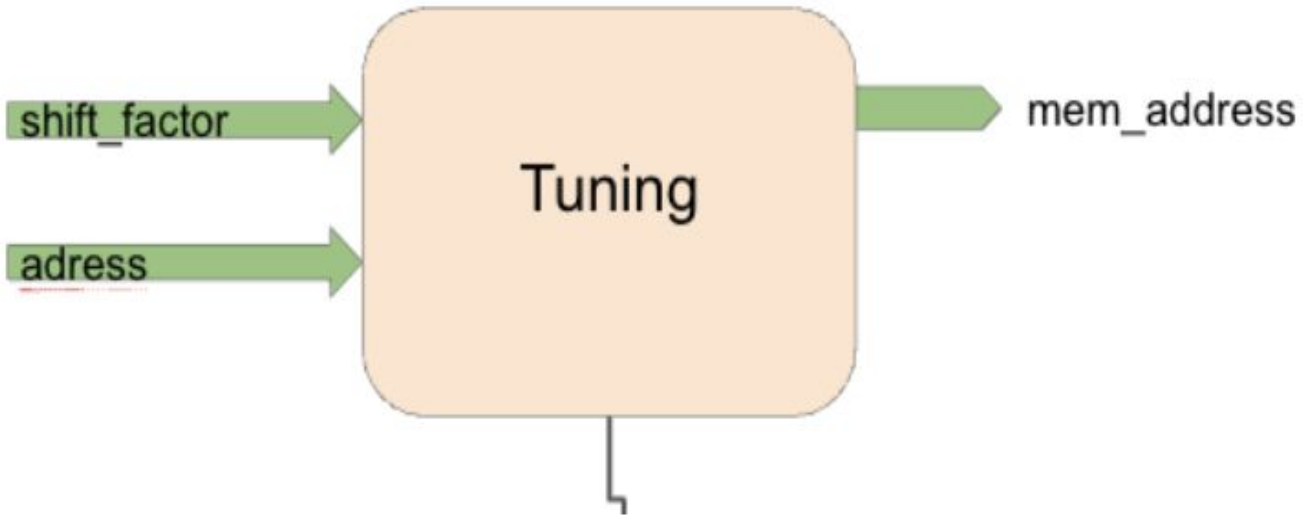
WS

FFT

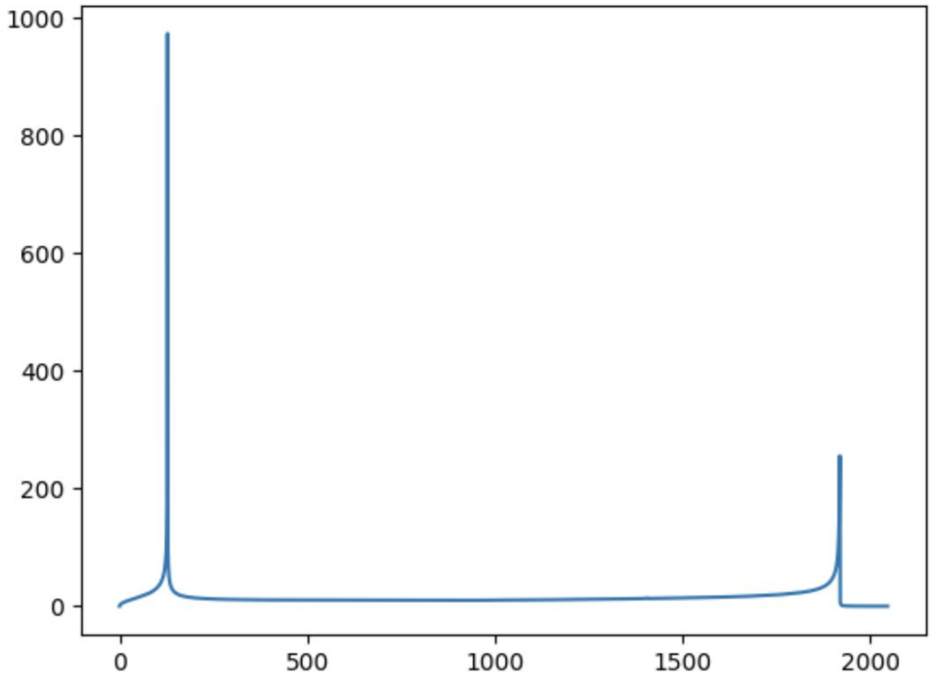


FFT
source_real
source_eop
source_sop
source _valid
sink_ready
reset
clk
MEM
ifft_real
ifft_imag
source _sop,
source_eop,
source_valid
IFFT
source_valid
source_real
source_ready
Tuning
shift_factor
adress
mem_address
Peak
Detector
peak_valid
peak_index
Shift
shift_factor

Tuning: we are not changing but shifting data. Tuning tells me where the new address is.



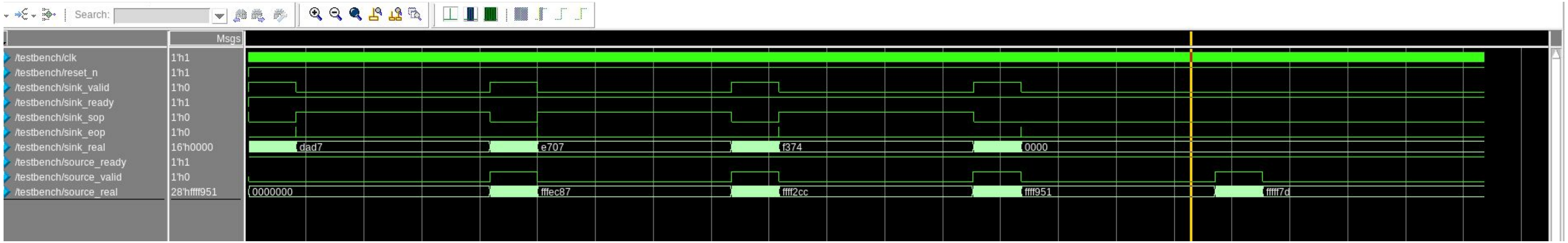
Pitch Shifting



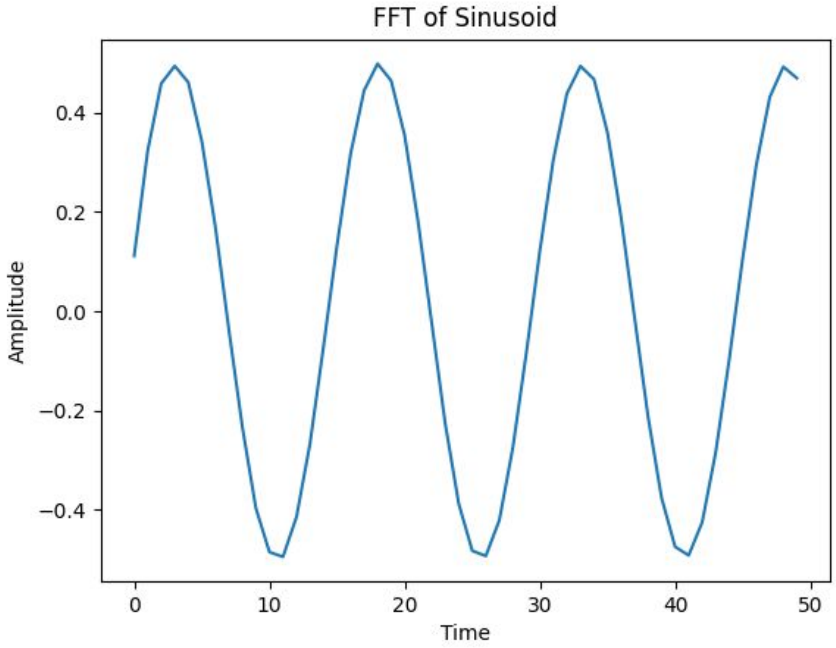
Real-time signal has symmetric magnitude in fourier domain. Compute cyclic shift in different direction for each half of the frequency bins.

-> shift_table: compute shift_factor in O(1) time

Verilog Simulation!



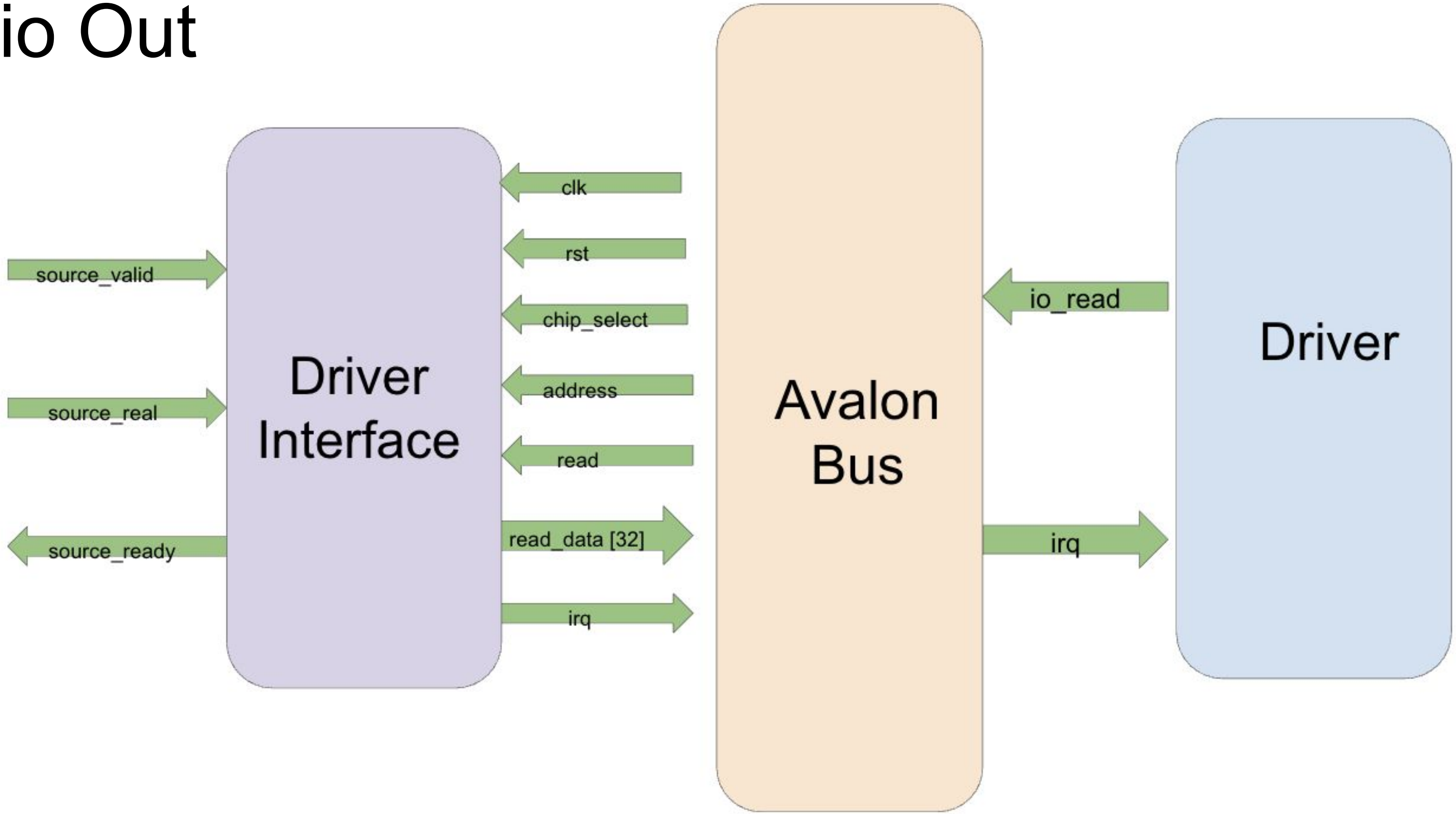
Output Data with input 500Hz sinusoid. # of data points goes from 16 to 15 period.



500 * (16/15) = 533Hz
500 *(2^(1/12)) = 529Hz.

Close Enough!!!

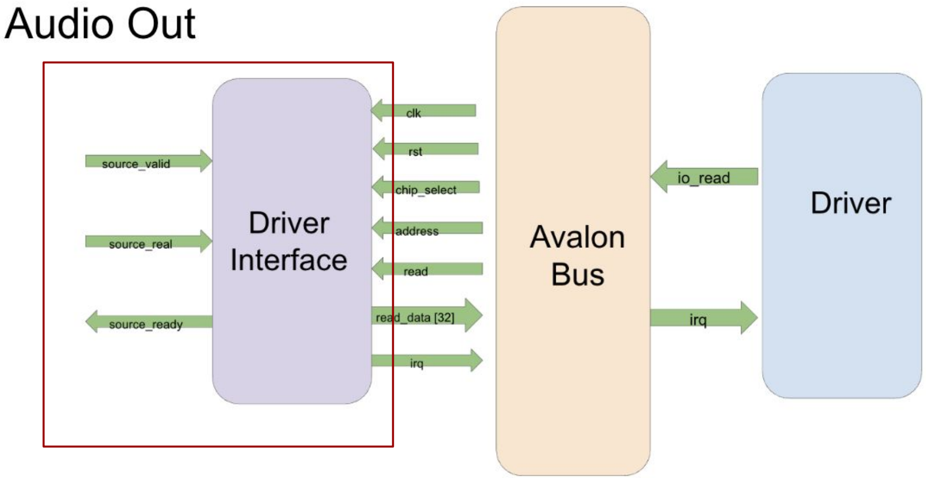
Audio Out



source_valid
source_real
source_ready
Driver Interface
clk
rst
chip_select
address
read
read_data [32]
irq
Avalon Bus
io_read
irq
Driver

FFT to Driver Interface

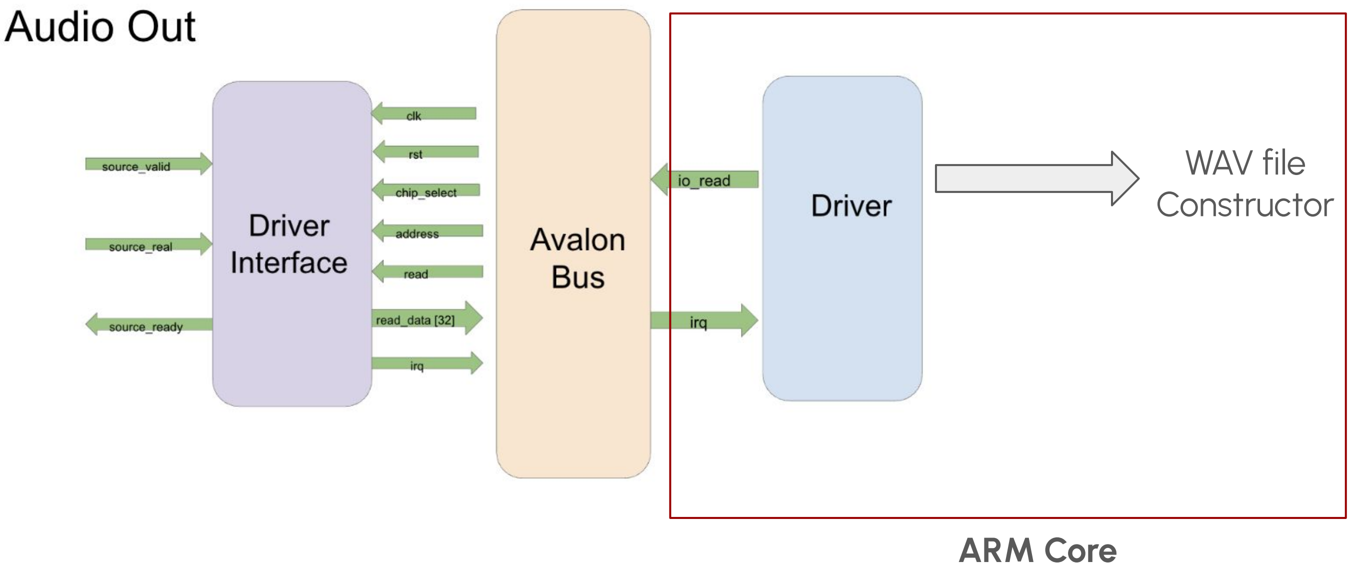
After the microphone input is digitized, the FPGA applies an FFT to analyze the frequency content. Custom pitch detection logic identifies the nearest musical note, and an IFFT reconstructs the pitch-corrected waveform. The resulting 28-bit audio samples are streamed from the IFFT module into the Driver Interface.



Software

Driver Interface to ARM Core

The hardware logic writes 2048-sample frames (from FFT and IFFT modules) to memory-mapped registers. The driver interface does not use interrupts, instead, the ARM core polls the device periodically at a ~8kHz sampling rate to read new audio data. The interface transfers 32-bit integer values over the Avalon bus to the Driver..



Device Driver to User Memory Space

Our Linux kernel module maps FPGA memory and exposes it as /dev/audio. On each ioctl call, the driver reads a block of 2048 samples using ioread32() and copies them from kernel to user space via a defined audio_arg_t structure.