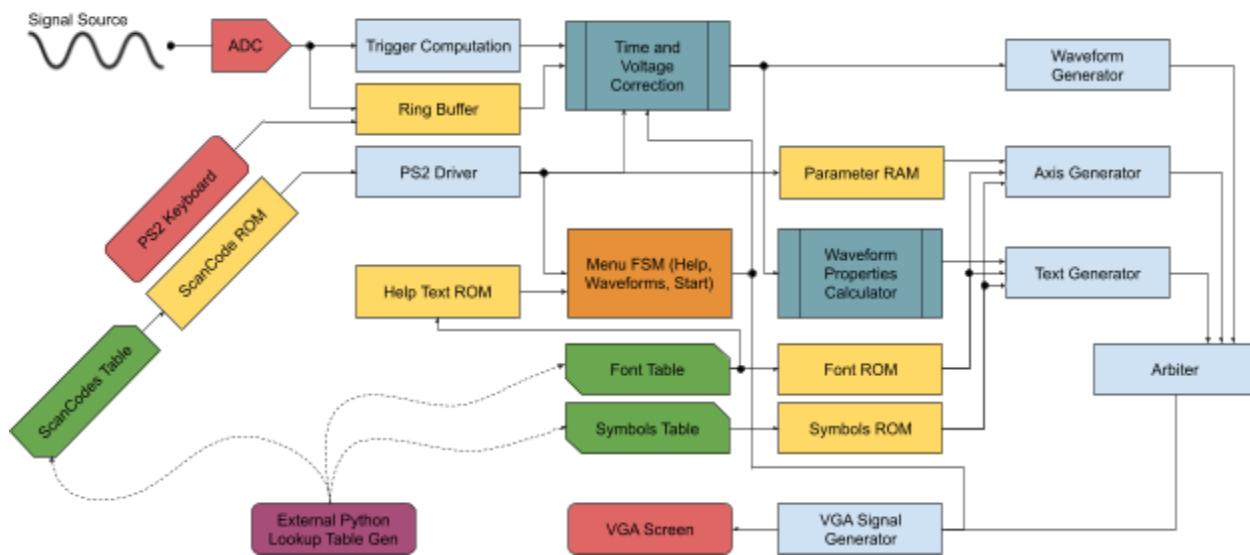


# Digital Oscilloscope ECE 241 Milestone #2

Rahil Harit-Singh and Ayan Ali

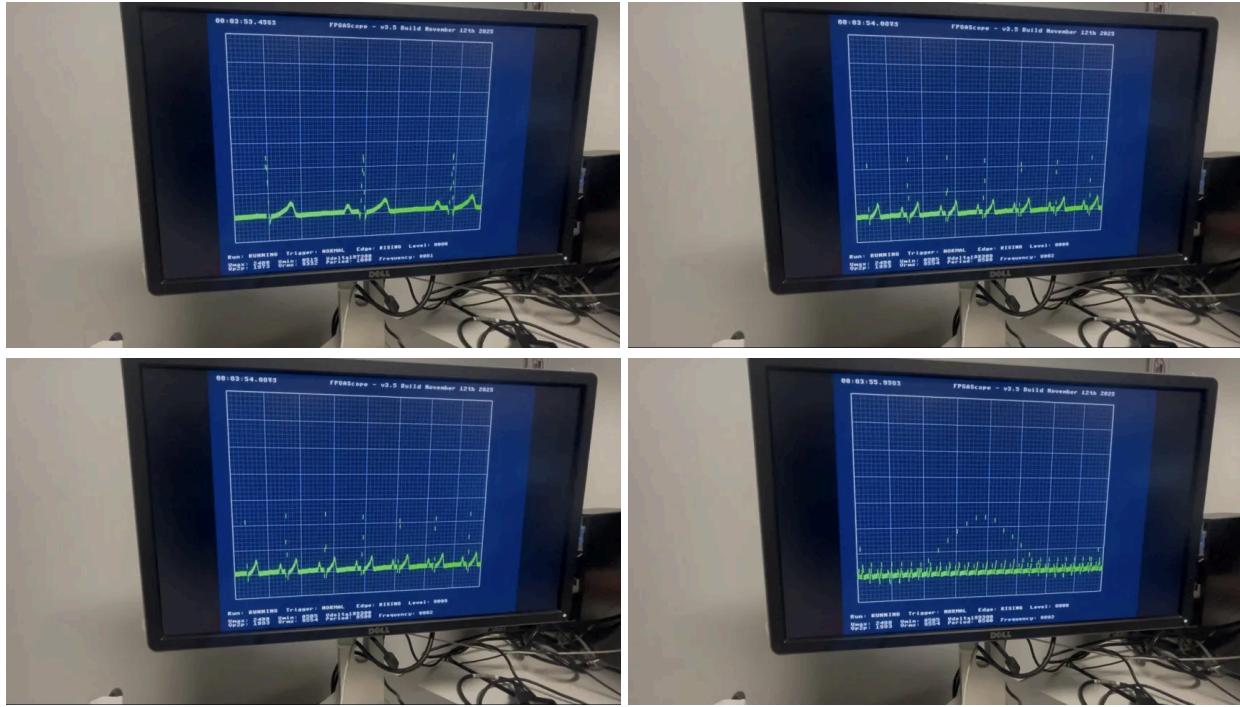
## Summary

This week involved getting the ADC to output a signal to the ringBuffer and research surrounding ways we can decimate and average our frame buffer to allow us to be able to sample waves of very high and very low frequencies. This week also involved writing the infrastructure behind a drop-down menu for our scope's controls (and eventual dual or quad channel). This week also involved the writing of time and voltage offset and sampling and the beginning of trigger logic so that waveforms do not scroll across the screen.

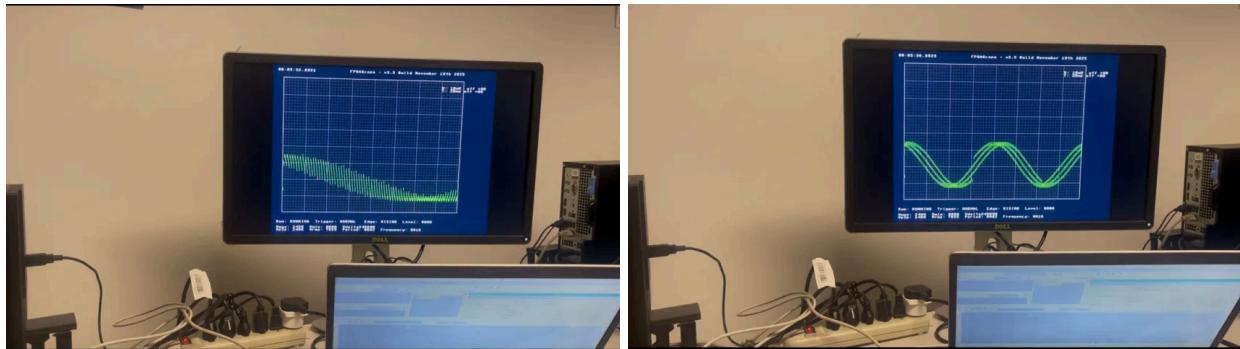


## Photos of Progress





Above images shows the zoom or the timescale variation ability of the scope.



The above images show the ability for the scope to be able to display a 10Hz signal as compared to a 70kHz signal as shown in the previous images. There is a visual artifact with three waves (which we plan to resolve before the final presentation) but the right image shows two waveforms when the timescale is set to 20ms which matches the expected period of a 10 Hz signal.

## New Ideas Developed

If there is time before the final project the following ideas might be interesting:

1. Instead of changing the sample rate of the ADC, keep the sample rate of the ADC as small as possible and then make a large buffer that can be sampled by the scope at different time scales.
2. Adding drop-down menus would just involve rendering a specific section of the screen and giving a transparent color to the rest of the screen, something that would look very good for selecting channels.

3. There is an issue where if a wave is too steep it will have gaps, the temporary way to fix this is make the line thicker, this reduces accuracy. Interpolation using  $\sin(x)/x$  or some sort of filtering would make scope more accurate.
4. Adding a second channel is quite straightforward as it is multiplying most modules x2 and handling the interconnections. We have used a small amount of resources, so this is feasible to do so. Perhaps even 6-channel is feasible given the 6-pin ADC header.

## Work Assignments

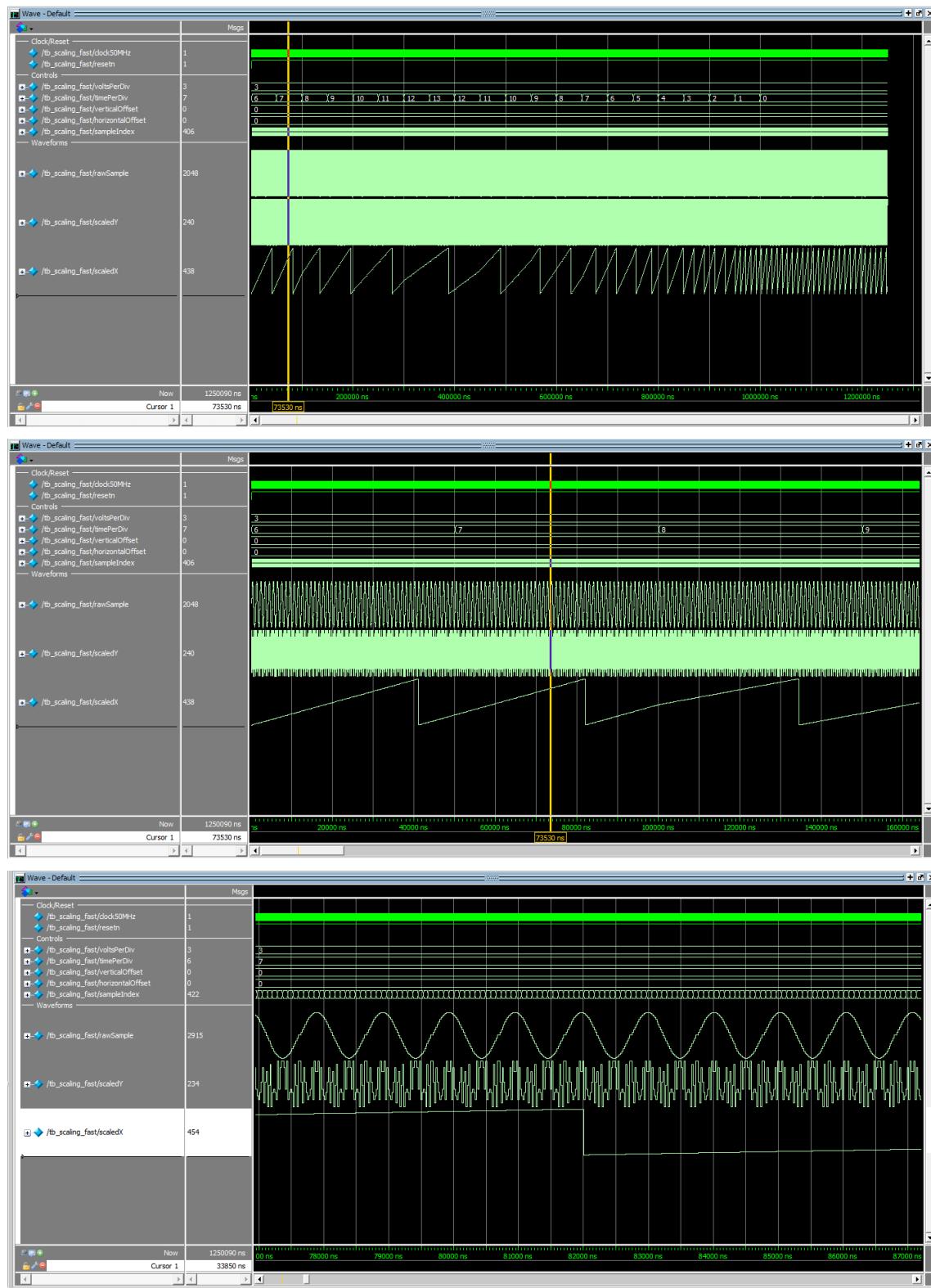
We had aimed to get the below work done in time for the halfway point between Milestone #2 and Milestone #3 (final presentation), hence some parts of the table below are in-progress. Given that during the actually writing of the verilog we discovered that we had to change our timeline slightly some work has been pushed to Milestone #3 and some Milestone #3 has been brought forward to Milestone #2.

Work labelled DONE is finished. Work labelled IN-PROGRE[SS] is work we aim to finish before the end of the work week.

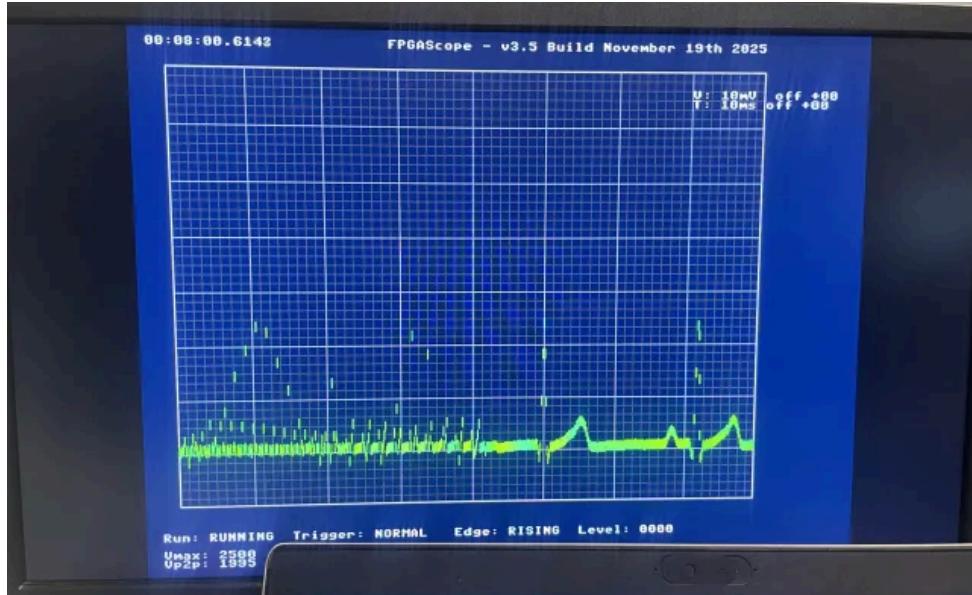
Given the nature of this week's work, some items in the list will be updated in the coming week, however for the basis of this week's progress we can call it complete.

Assignee	Work	Status
Ayan	Verilator Testbench ModelSim Testbenches  oscilloscopeTop.v waveformTrigger.v timeVoltageCorrection.v Itc23088Reader.v controlADC.v  Milestone #2 Writeup	DONE DONE  DONE DONE DONE DONE DONE DONE  DONE
Rahil	updated textComposer.v value changes for the timescale to be clean timeVoltageCorrection.v oscilloscopeTop.v waveformTrigger.v  Milestone #2 Writeup Draw Cool Logo + Parsing for Display	DONE DONE DONE DONE DONE  DONE IN-PROGRE

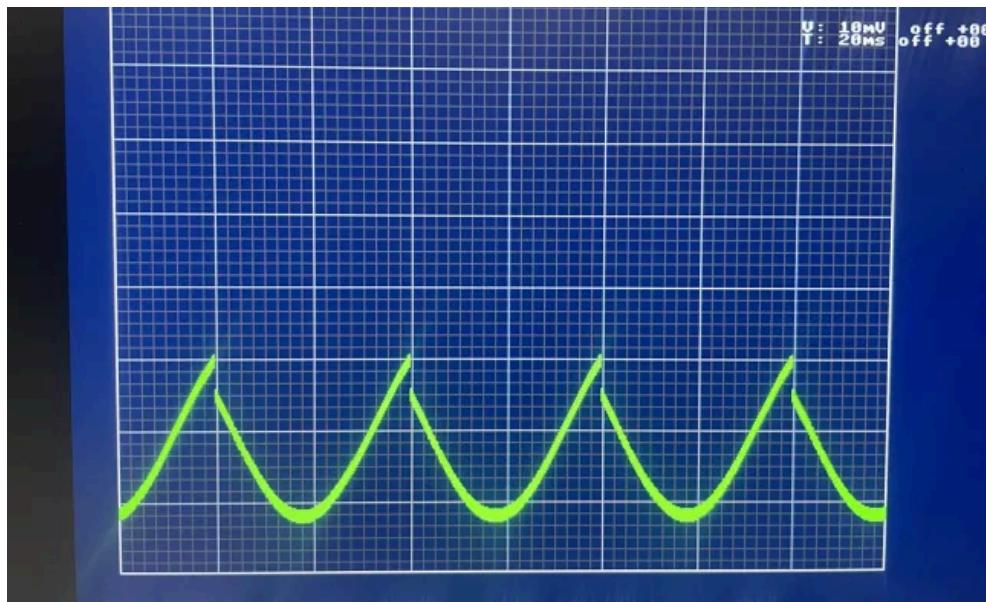
## ModelSim Testing for Scaling



## Spooky Bugs



This bug has happened when experimenting with a dynamic buffer size, the issue was that for every increase of the time-scale less data would be able to be displayed per pixel slice of the screen and so it would take smaller and smaller portions of the screen from the left hand side, hence you can see the most zoomed in section on the left.



In an effort to fix this, we tried to repeat the data across the screen to span the full area, but this would simply result in repetitions of the same waveform with cutoffs.

We resolved the final fix would be to actually increase the number of samples taken, i.e. increase the size of the sampling ring buffer and then sample as much or as little as needed.

The issue here is we run into an acquisition time issue, this can be solved by having a dynamic sample rate to only store as much as needed in the buffer.