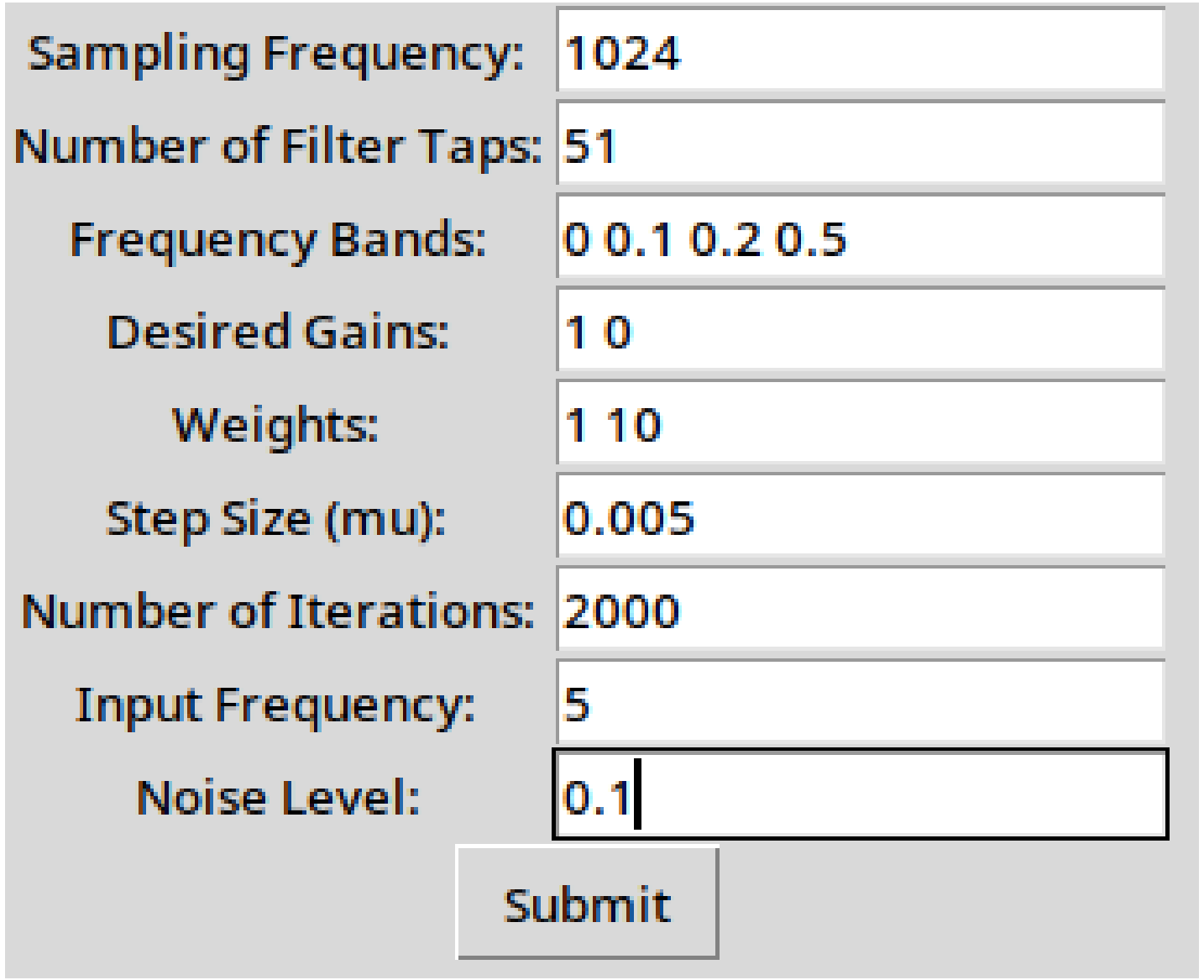
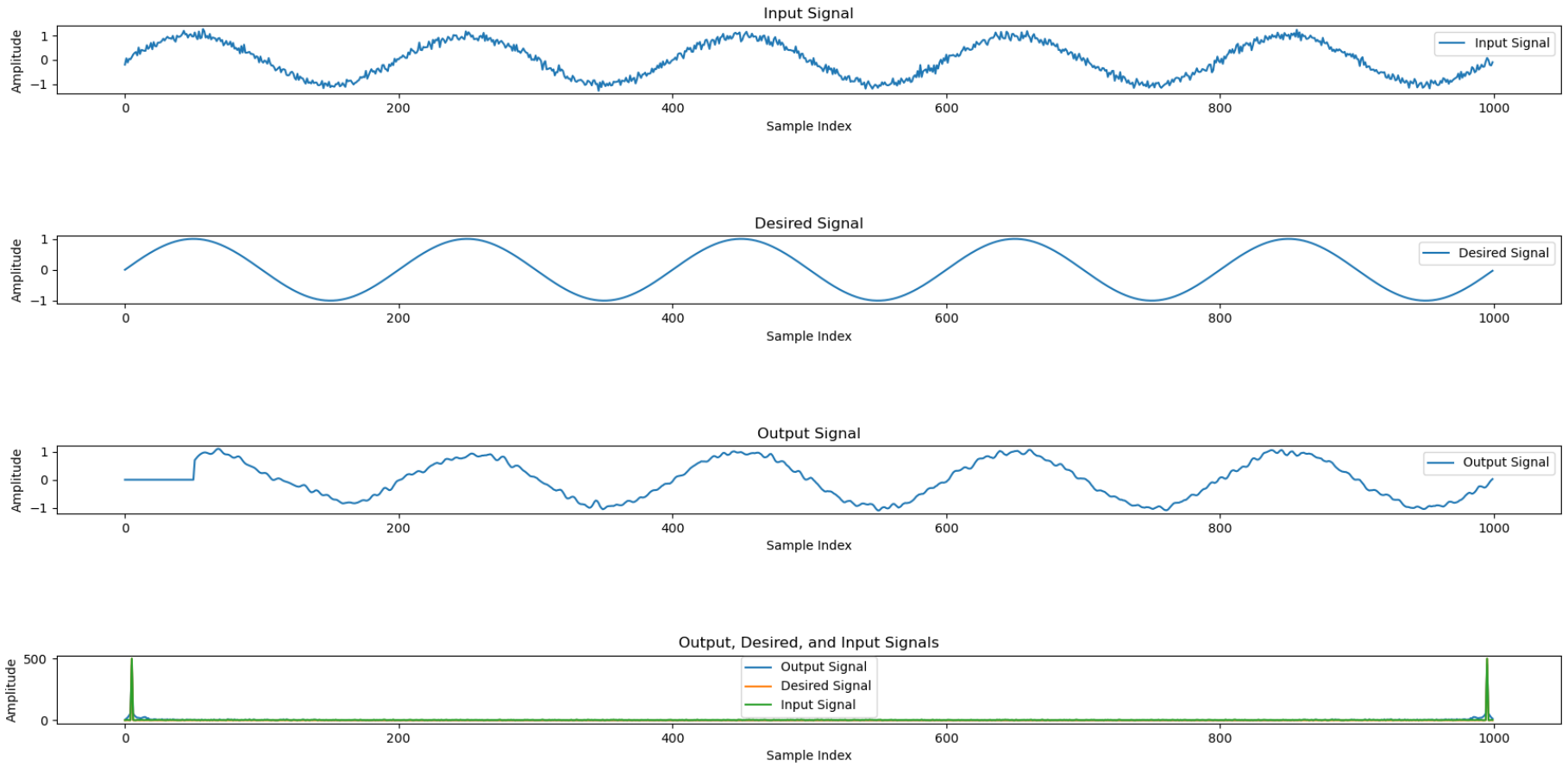
CHAIN: Week of June 24

Kayla Stafford

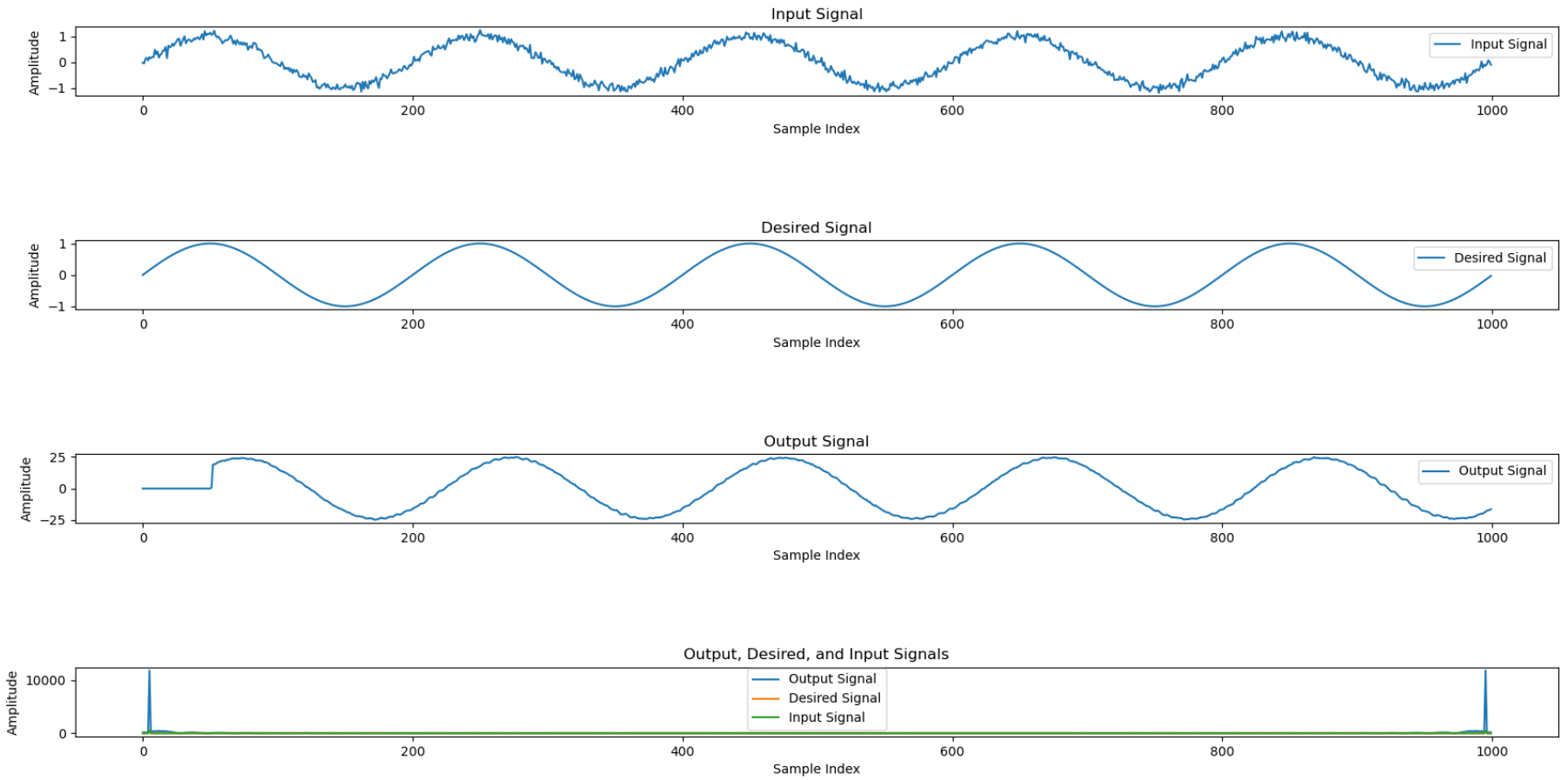
Parameters


Sampling Frequency:
1024
Number of Filter Taps:
51
Frequency Bands:
0 0.1 0.2 0.5
Desired Gains:
1 0
Weights:
1 10
Step Size (mu):
0.005
Number of Iterations:
2000
Input Frequency:
5
Noise Level:
0.1
Submit

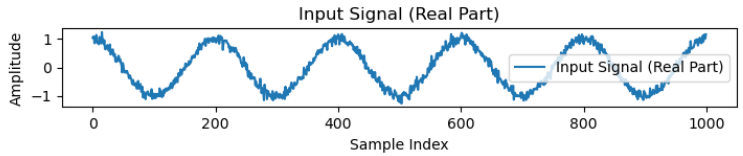
Nondigitizedadaptive.py

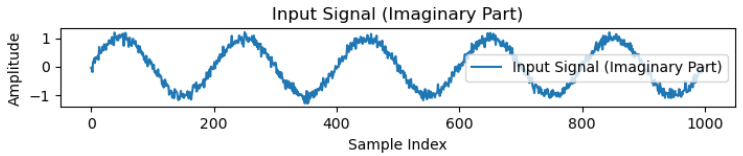


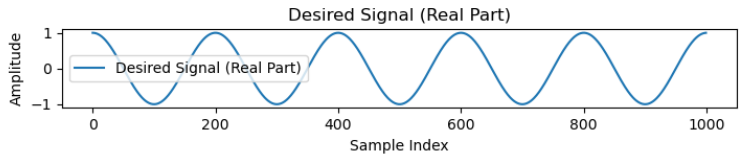
DigitizedAdaptive.py

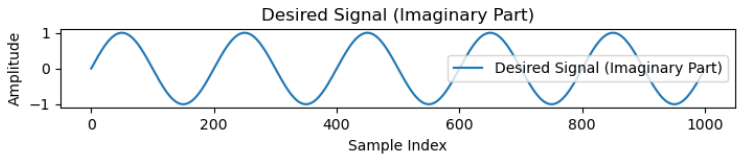


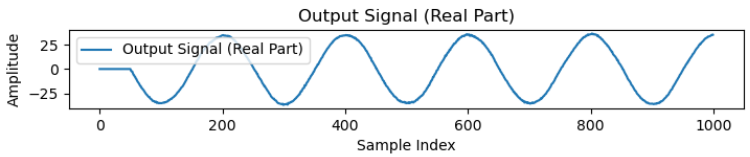
DigitizedComplex.py

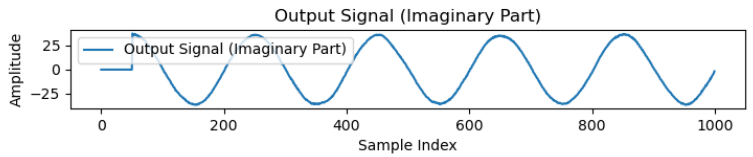


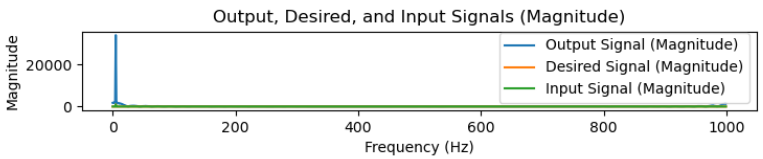


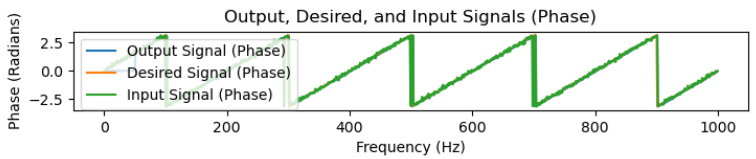




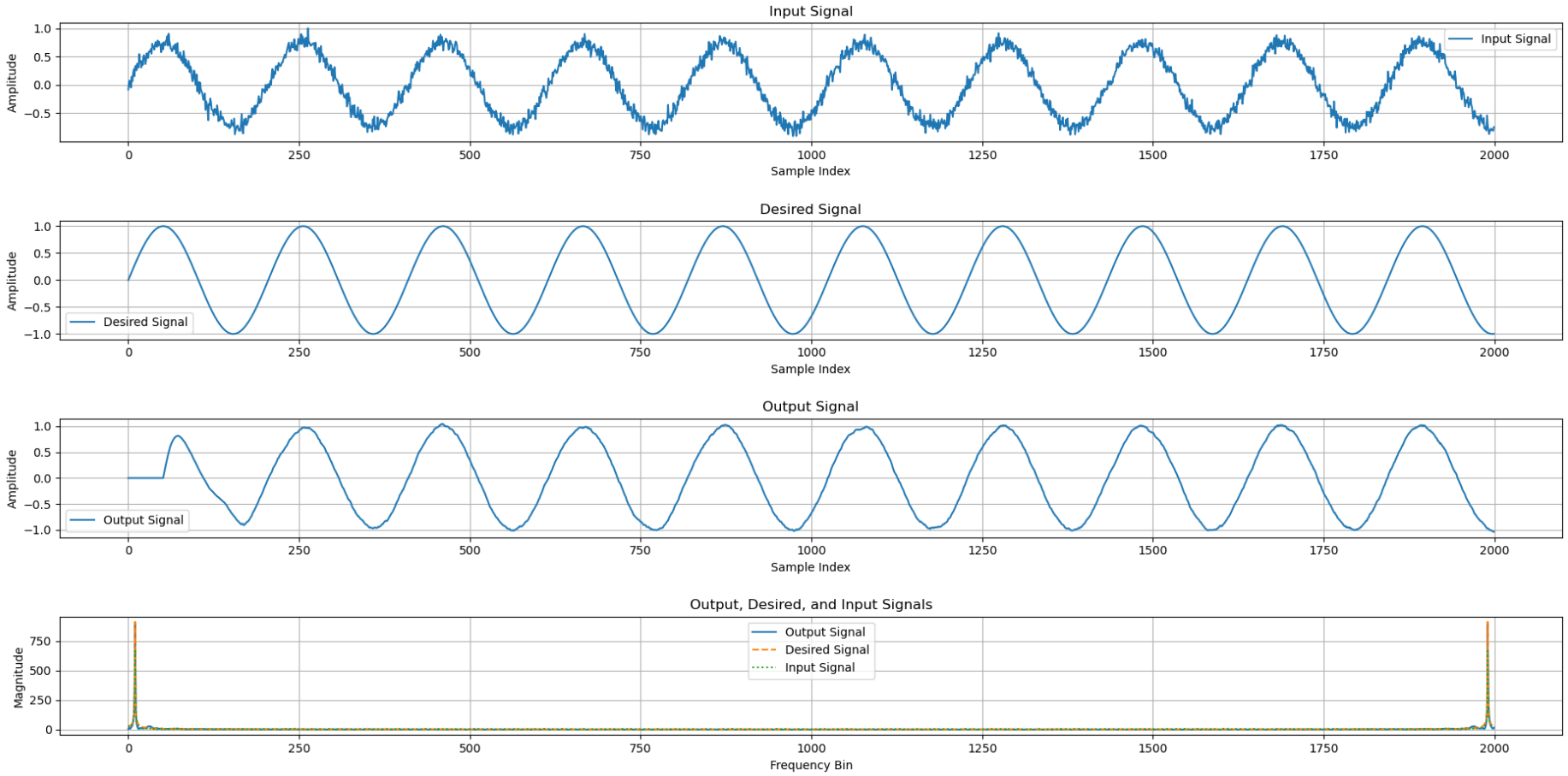








DumbFilter.py



Test.py

- Code is written, having trouble reading and executing the file.

June 24 2024

- Goals:
  - Install all required software on desktop
  - Create Python package
- What I learned:

June 25 2024

- Goals:
  - Implement DumbFilter.py in Verilog
  - Learn about Verilog syntax and program design
  - Field station visit
- What I learned:

June 26 2024

- Goals:
  - Write DumbFilter.py in Verilog
- What I learned:
  - Verilog design procedures

Implementing DumbFilter.py in Verilog

- The first fixed points are the memory reads.
- Each RAM must take one clock cycle to read where you do nothing but else with the value read.
- Firstly, add capability for user to update filter coefficients.
- Filter portlist: There's an i_reset strobe to clear the filter's internal memory, an i_tap_wr signal to indicate that the filter is being loaded with new taps through i_tap, and an i_ce signal to indicate both that a new sample is ready at i_sample and that a sample is ready to be read out at o_sample.
- Design process, debugging, pipelining

June 27 2024

Resources Consulted

- https://zipcpu.com/dsp/2018/05/17/slowsymf.html
- https://zipcpu.com/blog/2017/08/14/strategies-for-pipelining.html
- https://zipcpu.com/dsp/2017/11/04/genfil-tb.html
- https://zipcpu.com/blog/2017/10/19/formal-intro.html