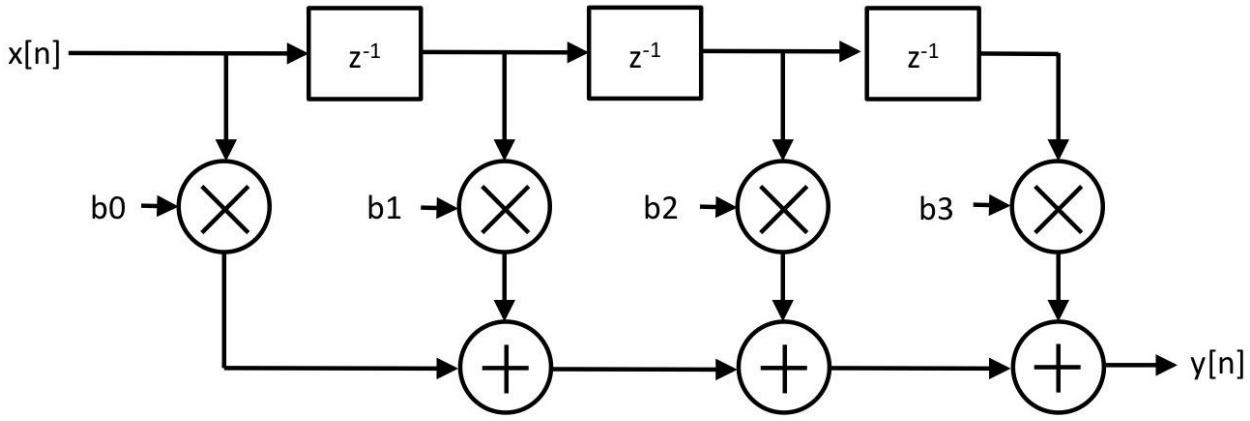
CS4362 Hardware Description Languages

Lab 1: Implement a Finite Impulse Response Filter using FPGA

Implement a FIR filter using the FPGA. FIR filter performs the following operation.



Finite Impulse Response (FIR) Filter is performing an elementwise multiply and add operations, on the samples which are delayed by a single time step (Z^{-1}). Multiplication happens between the element of the filter coefficients of the desired design (i.e. low pass / band pass or high pass etc.,) with the element of the input signal.

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \ldots$$

- $y[n]$ is the output signal at time n
- $y[n]$ is the input signal at the time n
- b_i are the filter coefficients.
- $n-1, n-2,$ … are (time delayed/ y axis offset) by 1,2, … cycles/pixels.

1. Determining Filter Coefficients.

First we have to determine the filter coefficients b_i and the number of "taps" (i.e. stages of the filter). We can use a filter design tool such as t-filter tool (http://t-filter.engineerjs.com/) or FDA Tool in Simulink (with Matlab) to generate suitable coefficients for a filter of our choice.

For this lab, you can design a low pass filter with the following characteristics using the t-filter tool.

from	to	gain	ripple/att.
0 Hz	400 Hz	1	5 dB
500 Hz	1000 Hz	0	-40 dB

Keep the sampling rate as 2000Hz and the number of desired taps as 25. Also change the output type to 16 bit integers from decimal. You will receive 25 filter coefficients to be used in your filter implementation.

If you like, you should be able to adjust these parameters and implement any type of filter.

2. **Implementation of the FIR filter.**

Use a hardware description language (VHDL or Verilog) to implement the FIR filter. Note you have integers as input values and your signals are also in signed integers.

3. **Prepare a testbench.**

Prepare a suitable automated test bench for testing your design.

Resources:

A C++ Floating Point Finite Impulse Response Filter

(You can create an account, login, and register for high level synthesis course so that you will get certain resources to complete High Level synthesis labs as well)

https://www.hackster.io/whitney-knitter/dsp-for-fpga-simple-fir-filter-in-verilog-91208d

FIR Filters for video sharpening.

https://www.youtube.com/playlist?list=PLGzeDuLmmxDr10_4zRujRBVnqij-PZyYb