ELC 433-L1

Lab 4 – Realization of FIR and IIR Filters

Brian Worts and Chris Jenson

11/8/20



**Introduction:**

This lab used the concepts from the previous labs to gain experience with Direct Forms and Transposed realizations of Finite Impulse Response (FIR) filters. The lab then expanded to include experience with Infinite Impulse Response (IIR) filters. The background for this lab is the equation for a Mth order FIR filter in combination with the direct form of a filter. In the direct form, the components represent the different aspects present in the equation. Linear network theory says that it is possible to draw a dual network that produces the same result as the original. For this lab, the dual network is the transposed realization. Only three components are required for a digital filter: a summation, a product, and a unit delay. Using these components, a filter will be designed as well as a transpose of that filter that produces the same result. The combination of knowledge about response filters and their properties with direct form are used to complete this lab and further the student’s understanding of the material.

**Procedure:**

This lab began with Step 1 which was to draw a block diagram like a signal-flow graph. The diagram represents the 13 module instances and 13 wires found in the direct\_form\_1 file supplied. It was completed for M=4 and is shown in the Results section of this report. All the desired files were then acquired from Canvas and added to a Vivado project for Steps 2 & 3.

With the project created, the project was then simulated for Step 4. The simulation resulted in a step response via the waveforms as expected. The input and output signals, din and dout had their waveform style changed to analog. Their analog settings were then swapped from Hold to Linear. Radix was also changed to signed decimal. The result of these changes is shown in Results. Using the waveform, the DC level was measured to estimate the DC gain. This was compared to the theoretical gain found based on the filter coefficients for Step 5.

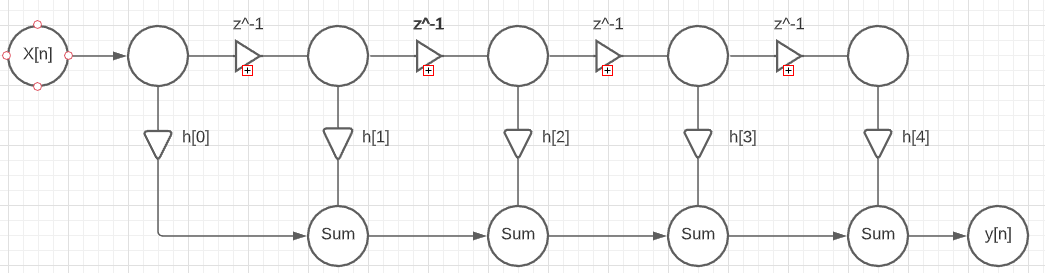
Step 6 involved using the block diagram from Step 1, but with notable changes. All the arrows were reversed, and the input and output were swapped. This was done to acquire the transposed version of the FIR filter with M=4. The splitter nodes became summing junctions. The next step, Step 7, was to implement this new transposed filter in Verilog. By adjusting the DUT instantized in the Testbench, the transposed filter was modeled. The results of this simulation are shown in Results, it is noted that the result matches the simulation of the form from Step 1 which was desired.

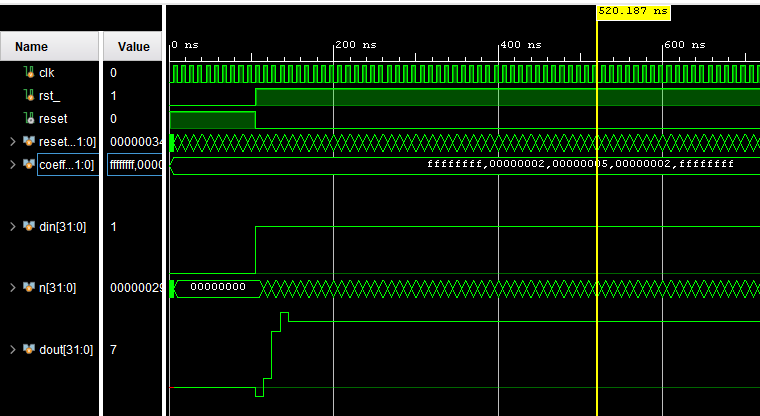
Step 8 involved a chirp signal. This was achieved by altering the testbench stimulus from a step to a chirp. The chirp is generated by a discrete-time sinusoid with linearly increasing instantaneous frequency. This means that the envelope of the output of a filter, given a chirp, is a representation of the magnitude of the response. The coefficients were then found by examining the testbench. MATLAB was used to plot the magnitude and phase response. The results of Step 8 can be found in Results.

Step 10 was top challenge us by creating a Verilog instantiation of the first order IIR filter from the lab handout. This required a new product module using fixed point, rounding, and shifting by 11 places to return to an integer value. Saturation was not needed as the signals were declared as integers.

**Results:**

Part 1) The block diagram representing the 13 module instances and 13 wires.

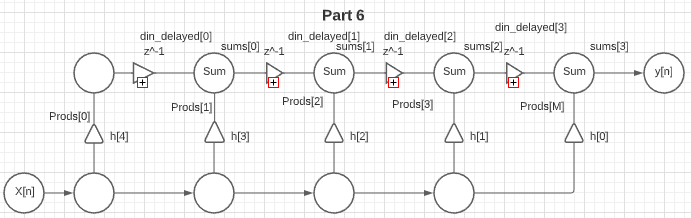


Part 4) Waveform showing the coefficients, measured dout value, and the din and dout outputs.

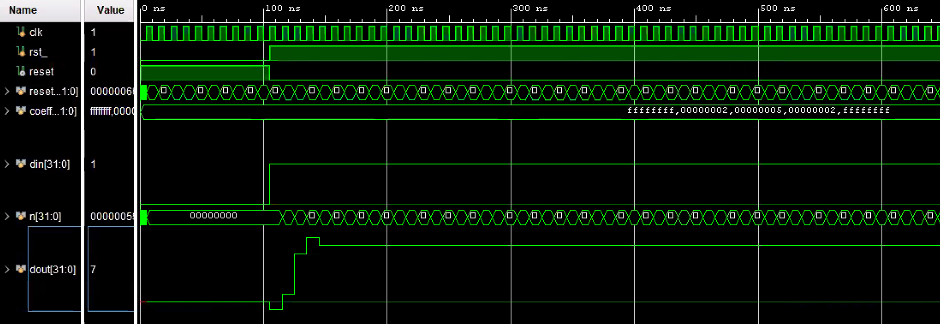
Part 5a) The DC level at the output of the step response was measured at 7 which is the estimate of DC gain.

b) The filter components were -1, 2, 5, 2, and -1. This sums to 7 and matches the estimate for DC gain.

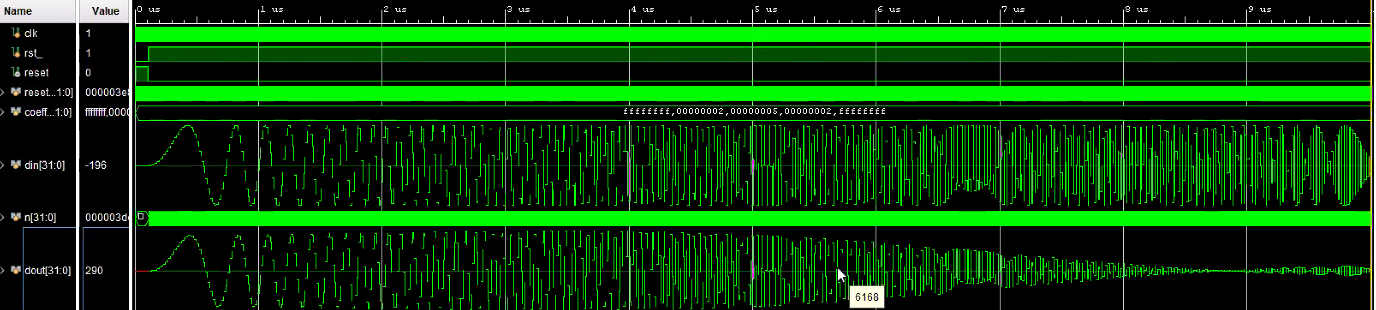
Part 6) Transposed realization of the FIR filter block diagram



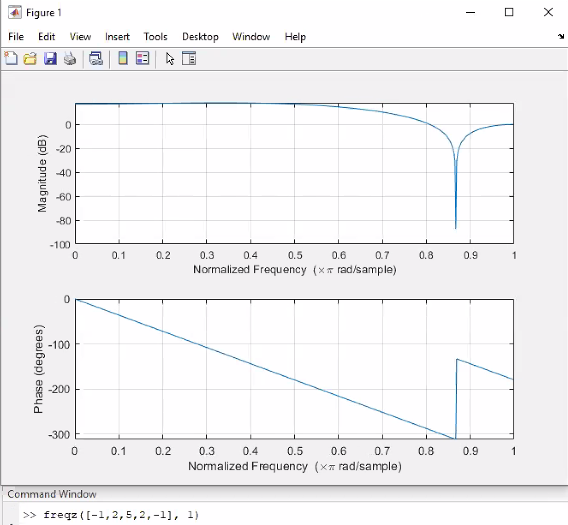
Part 7) Waveform showing the coefficients, measured dout value, and the din and dout outputs based on the transposed diagram.



Part 8)



1. The coeficients were found to be: -1,2,5,2,-1

b) MATLAB graphs of magnitude and phase:

c) Does the filter have linear phase response? If so, why?

The filter does have a linear phase response because there is no distortion other than a time delay.

Part 9) What flip-flop based logic circuit from ENG 312 is reminiscent of the chain of unit delay blocks in this design?

The chain of unit delay blocks is reminiscent of a shift register

**Engineering Work:**

Transposed DUT Verilog Code:

// Instantiate the delay line  
z\_minus\_1 u\_dm0( .din(prods[0]),.dout(din\_delayed[0]), .clk(clk), .reset(reset) );  
z\_minus\_1 u\_dm1( .din(sums[0]), .dout(din\_delayed[1]), .clk(clk), .reset(reset) );  
z\_minus\_1 u\_dm2( .din(sums[1]), .dout(din\_delayed[2]), .clk(clk), .reset(reset) );  
z\_minus\_1 u\_dm3( .din(sums[2]), .dout(din\_delayed[3]), .clk(clk), .reset(reset) );  
  
// Instantiate the gain blocks  
prod u\_prod0(   .din(din),   .coeff(coeffs[M]), .dout(prods[0]) );  
prod u\_prod1(   .din(din),   .coeff(coeffs[3]), .dout(prods[1]) );  
prod u\_prod2(   .din(din),   .coeff(coeffs[2]), .dout(prods[2]) );  
prod u\_prod3(   .din(din),   .coeff(coeffs[1]), .dout(prods[3]) );  
prod u\_prodNM1( .din(din),   .coeff(coeffs[0]), .dout(prods[M]) );  
  
// Instantiate the summing junctions  
sum u\_sum1( .x(prods[1]),   .y(din\_delayed[0]),  .z(sums[0]) );  
sum u\_sum2( .x(prods[2]),   .y(din\_delayed[1]),  .z(sums[1]) );  
sum u\_sum3( .x(prods[3]),   .y(din\_delayed[2]),  .z(sums[2]) );  
sum u\_sum4( .x(prods[M]),   .y(din\_delayed[3]),  .z(sums[M-1]) );  
  
// Last sum is the output signal  
assign dout = sums[3];

**Knowledge Gained:**

This lab focused on filters and their properties, specifically in direct form and the transpose of direct form. The students made graphs to visually comprehend what the filter was doing. During this process, it was extremely helpful to label the components in the block diagrams with labels that matched the Verilog components. This made implementing the transposed diagram much easier and helped with understanding. Comparing the regular direct form with the transposed waveforms visually confirmed that they were equivalent. In the MATLAB graphs, it was observed that the zero point was about 0.8 which corresponds to 800 microseconds in the chirp Verilog waveform and this matches what was expected. The magnitude graph saw an infinite dip at around 8.7, this corresponds with the time delay in the phase graph. All these components came together to further the students understanding of filters, their properties, and how to analyze advanced components.

**Who Did What:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student** | **Analysis** | **Development** | **Coding** | **Results** | **Writing** |
| Brian | 50 | 50 | 50 | 75 | 25 |
| Chris | 50 | 50 | 50 | 25 | 75 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Expectation** | **Max Pts.** | **Pts. Deducted** |
| Introduction | Brief overview | 1 |  |
| Procedure | Brief description of procedures | 1 |  |
| Design/Engineering Work |  |  |  |
|  | Mathematical analysis and code are correct. | 2 |  |
|  | Paste text of code, not images | 0.5 |  |
|  | Use SPACE not TAB characters. Strict indentation. Consistent placement of IF, ELSE, FOR, END | 0.5 |  |
|  | Full commenting | 1 |  |
| Results |  |  |  |
|  | Present all plots with informative, highly descriptive captions. What ENG 312 circuit is reminiscent of chain of unit delays? ANS=shift register. | 0.5 |  |
|  | Careful sketches of normal and transposed block diagrams | 0.5 |  |
|  | Measure DC gain, and compute the theoretical DC gain. | 0.5 |  |
|  | Chirp stimulus - compare against theoretical magnitude response | 0.5 |  |
| Knowledge Gained |  | 1 |  |
| Who Did What |  | 1 |  |
| **Total** |  | 10 |  |