Digital Communication Circuits Laboratory

Lab. 3

Digital Filter

I. Purpose

In this lab., we will learn to implement finite impulse response (FIR) filter.

II. Principle

Digital filter is an indispensable component in signal processing and communication systems. It can be classified into two categories, i.e. infinite impulse response (IIR) filter and finite impulse response (FIR) filter. The FIR filter is stable and has linear phase. Thus, it is widely used in various applications. The IIR filter has the feedback path, which results in good efficiency to obtain excellent frequency response with fewer arithmetic components.

Assume that the length of the FIR filter is L. The coefficient is denoted as $h[i]$, for $i = 0,1,\dots,L-1$. The Z-transform of FIR filter is represented by

$$H(z) = \sum_{i=0}^{L-1} h[i]z^{-i} \tag{1}$$

The impulse response of an FIR filter is its coefficients, i.e. the filter output signal is the convolution of the filter input signal and filter coefficients. If the input signal is denoted as $x[n]$ and output signal is denoted as $y[n]$, then

$$y[n] = \sum_{i=0}^{L-1} h[i]x[n-i]. \tag{2}$$

In other words,

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[L-1]x[n-L+1]. \tag{3}$$

From Eq. (3), we know that every output $y[n]$ needs the current and past inputs $x[n]{\sim}x[n-L+1]$ to complete the computation. Thus, storage of length L is required.

Generally speaking, the FIR filter can be implemented in two ways, namely direct form and transpose form. Direct form FIR filter is given in Fig. 1 based on Eq. (2). In the direct form architecture, there exists a long critical path consisting of a multiplier and several adders. Hence, it is not suitable for high-speed operation. However, all the register word-lengths are the same as that of the input.

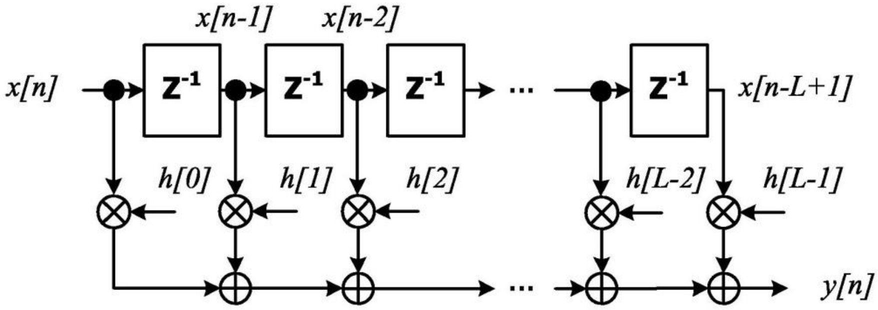

Fig. 1 FIR filter in direct form.

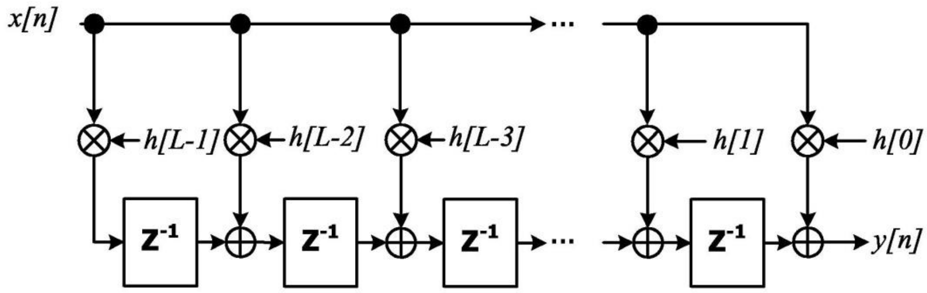

Fig. 2 FIR filter in transposed form.

We can also implement the FIR filter by the transposed form architecture, as shown in Fig. 2. In this architecture, the critical path is reduced to one multiplier and one adder. Thus, it is suitable for high-speed operation. However, after arithmetic operation, the word-length will increase. As a result, the requirement of registers will be a little more than that of the direct form architecture.

III. Procedures

1. Given a filter with the following continuous-time impulse response is given as follows.

$$h_p(t) = \frac{1}{\sqrt{T}}\left\{(1-\alpha)\,\mathrm{sinc}\left(\frac{(1-\alpha)t}{T}\right) + \alpha\left[\mathrm{sinc}\left(\frac{\alpha t}{T}+\frac{1}{4}\right)\cos\left(\frac{\pi t}{T}+\frac{\pi}{4}\right) + \mathrm{sinc}\left(\frac{\alpha t}{T}-\frac{1}{4}\right)\cos\left(\frac{\pi t}{T}-\frac{\pi}{4}\right)\right]\right\}$$

   Insert $t = nT_s$ and $T = 8T_s = 1$. Namely, 6 points are sampled in one symbol period. We will implement a $33(= 8 \times 4 + 1)$-tap causal FIR filter with $\alpha = 0.5$.

2. Feed the input signal given as follows into the filter

$$x[n] = \sin\left(-\frac{2\pi n}{64}\right) + \cos\left(\frac{2\pi n}{4}\right) \qquad \text{for } 0 \le n \le 128$$

   Observe the output time-domain waveform. Draw the spectrum of the input and output signals.

3. Please use direct form to implement 33-tap raised-cosine FIR filter. Decide the word-lengths of each data path after additions and multiplications so that the root mean squared error (RMSE) between the fixed-point output and the floating-point output is less than 2^{-a}.
4. Please use transposed form to implement 33-tap raised-cosine FIR filter. Decide the word-lengths of each data path after additions and multiplications so that the RMSE error between the fixed-point output and the floating-point output is less than 2^{-b}.

Remark: If $p = 0, (a, b) = (12,12)$. If $p = 1, (a, b) = (11, 13)$.
If $p = 2, (a, b) = (13,11)$, where $p = mod(d_1, 3)$ and d_1 is the last digit of your student ID.

==

5. Compare the frequency-domain magnitude responses of ideal floating-point representation in 1 and fixed-point representation in 3 and 4. Please given some explanations about their performances. (20%)
6. Please implement the direct form FIR. Use $x[n]$ as the input. Check the behavior and post-route simulation results. Compare the results with the Matlab floating-point results. (30%)
7. Please implement the transposed form. Note that modularity and parametric design can ease your loading. Use $x[n]$ as the input. Check the behavior and post-route simulation results. Compare the results with the Matlab floating-point results. (30%)
8. Find out the critical path of your design in Q5. Show the numbers of adders and multipliers in the critical path and list the timing information. (10%)
9. Find out the critical path of your design in Q6. Show the numbers of adders and multipliers in the critical path and list the timing information. (10%)
10. **(Bonus)** Programming your FIR filter into the FPGA board. Observe the measurement results. Compare with the behavior simulation results. (10%)

IV. Results

1. Please use Matlab to draw the impulse response and frequency response of the 33-tap square-root raised-cosine FIR filter. Note that you need to use scale in dB for the magnitude of the frequency response and use radian for the phase of frequency response versus normalized frequency. The x-axis must be marked with correct the label. Please explain whether the filter is high-pass, band-pass or low-pass and why.
2. Draw the time-domain waveform and frequency-domain response of the input and output in procedure 2.
3. To show how you determine the word-length, please use the word-length of data paths as the X-axis and error as the Y-axis. Scan the quantization error versus the

word-length. Mark the word-length settings in the block diagram of the direct form FIR filter.

a. Output error versus input word-lengths
b. Output error versus coefficient word-lengths
c. Output error versus word-lengths after multiplication
d. Output error versus word-lengths after addition

4. To show how you determine the word-length, please use the word-length of coefficients as the X-axis and error as the Y-axis. Scan the quantization error versus the word-length. Mark the word-length settings in the block diagram of the transposed form FIR filter.
   a. Output error versus input word-lengths
   b. Output error versus coefficient word-lengths
   c. Output error versus word-lengths after multiplication
   d. Output error versus word-lengths after addition

===

5. Draw the frequency-domain magnitude response for floating-point, fixed-point direct-form and transposed-form FIR filter.
6. Print out the behavior and post-route timing diagram. Show the errors of hardware outputs and Matlab floating-point results of direct from FIR by Matlab figures.
7. Print out the behavior and post-route timing diagram. Show the errors of hardware outputs and Matlab floating-point results of transposed from FIR by Matlab figures.
8. Mark the critical path of your direct-form FIR design in the block diagram. Also mark the input/output variable names and the word-lengths in the block diagram, which must be consistent with your Verilog codes. Print out the timing report. Show the numbers of adders and multipliers in the critical path
9. Mark the critical path of your transposed form FIR in the block diagram. Also mark the input/output variable names and the word-lengths in the block diagram, which must be consistent with your Verilog codes. Print out the timing report. Show the numbers of adders and multipliers in the critical path
10. **(Bonus)** Print out your measurement results and the comparison between measurement and RTL simulation. (10%)

V. Reference

[1] A. V Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall.