Homework 12

Topic covered in this homework is: fast Fourier transforms, digital filter structures, sample-rate conversion, and generalized linear-phase. Homework will be graded for (1) completion and (2) Three randomly picked problems will be graded. Submissions will be using gradescope. Please solve problems on your own in order to maximally benefit from this homework.

Problem 1: Radix-3 Decimation-in-Time (DIT) FFT

Assume $N = 3^v$ for the following parts:

(a) Derive a radix-3 DIT algorithm, i.e., express $\{X[k]\}_{k=0}^{N-1}$ in terms of three $\frac{N}{3}$-point DFT computations.

(b) Sketch the first stage of the signal flow-graph (SFG) of a $N = 9$ radix-3 FFT algorithm completely and trace the connections from the input to the output $X[0]$. Calculate the total number of complex multiply-adds needed. For simplicity, assume all twiddle factors require one complex multiplication even if they are equal to unity or -1.

(c) For a $N = 3^v$-point radix-3 FFT obtain: i) the number of stages; ii) the number of radix-3 butterflies per stage; and iii) the number of complex multiply-adds per butterfly. Using the results from i), ii) and iii) find the total computational complexity, i.e., the number of complex multiply-adds, of an $N = 3^v$-point radix-3 FFT? For simplicity, assume all twiddle factors require one complex multiplication even if they are equal to unity or -1.

(d) The DFT of a length-26 sequence $\{x[n]\}_{n=0}^{25}$ needs to be computed with a frequency resolution $\Delta\omega < \frac{\pi}{13}$. Compare the number of complex multiply-adds that would be required when using a radix-2 FFT and a radix-3 FFT.

Solution:

(a) To begin with, the N-point DFT is represented as

$$x[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi nk}{N}\right) = \sum_{n=0}^{N-1} x[n] W_N^{nk} \tag{1}$$

Then we split $\{x_n\}_{n=0}^{N-1}$ into three groups: (1) $n = 3m$, (2) $n = 3m+1$ and (3) $n = 3m+2$ with $m = 0, 1, \ldots, \frac{N}{3} - 1$. That is,

$$\begin{aligned} x[k] &= \sum_{m=0}^{\frac{N}{3}-1} x[3m] W_N^{3mk} + \sum_{m=0}^{\frac{N}{3}-1} x[3m+1] W_N^{3mk+1} + \sum_{m=0}^{\frac{N}{3}-1} x[3m+2] W_N^{3mk+2} \\ &= \sum_{m=0}^{\frac{N}{3}-1} x[3m] W_{\frac{N}{3}}^{mk} + W_N^k \sum_{m=0}^{\frac{N}{3}-1} x[3m+1] W_{\frac{N}{3}}^{mk} + W_N^{2k} \sum_{m=0}^{\frac{N}{3}-1} x[3m+2] W_{\frac{N}{3}}^{mk} \\ &= \boxed{F[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}] + W_N^k G[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}] + W_N^{2k} H[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}].} \end{aligned} \tag{2}$$

where $F[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}]$, $G[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}]$ and $H[\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}]$ are the three $\frac{N}{3}$-point DFTs of $\{x[3m]\}_{m=0}^{\frac{N}{3}-1}$, $\{x[3m+1]\}_{m=0}^{\frac{N}{3}-1}$ and $\{x[3m+2]\}_{m=0}^{\frac{N}{3}-1}$, respectively. Note that the $\langle\!\langle k \rangle\!\rangle_{\frac{N}{3}}$ is necessary since the range of k is $0, 1, \ldots, N-1$, which is larger than the range of $\frac{N}{3}$-point DFT.

(b) According to the derivation in (a), for the sequence $\{x[n]\}_{n=0}^{8}$, we divide it into three parts, which are $\{x[0], x[3], x[6]\}$, $\{x[1], x[4], x[7]\}$ and $\{x[2], x[5], x[8]\}$. Then we can sketch the first stage as following:

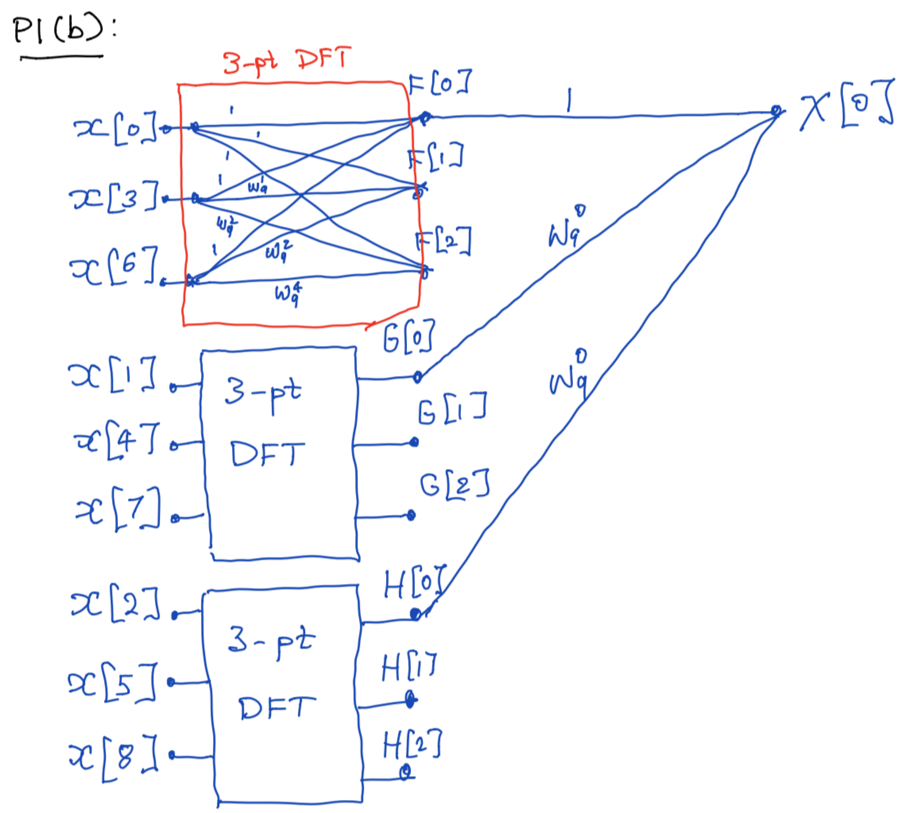


Figure 1: The first stage of the signal flow-graph of a $N = 9$ radix-3 FFT algorithm.

We can see that, if we include those factors equal to 1 or -1, in each radix-3 butterfly there will be 9 complex multiplications (since there are 9 connections) and 6 complex adds (since each output needs two complex adds, and there are three outputs), so in total each butterfly contains 9 complex multiply-adds. Since there are 6 butterflies (three butterflies in each stage, we have two stages), in total there are $\boxed{6 \times 9 = 54}$ complex multiplications and $\boxed{6 \times 6 = 36}$ complex adds needed. So the total complexity will be about $\boxed{54}$ number of complex multiply-adds.

(c) (i)Given $N = 3^v$, in each stage we divide it into three parts. Then there will be $\boxed{\log_3 N = v}$ number of stages; (ii) For each stage there will be $\boxed{N/3}$ number of radix-3 butterflies; (iii) According to the result in part (b), in each radix-3 butterfly there are 9 complex multiply-adds, and totally we have $\log_3 N \times \frac{N}{3} = \frac{N}{3}\log_3 N$ number of butterflies, the total complexity should be $\boxed{9 \times \frac{N}{3}\log_3 N = 3N\log_3 N}$ number of complex multiply-adds. The order of complexity should be $O(N\log_3 N)$.

(d) According to the requirement of the frequency resolution, the N-point of DFT should satisfy

$\frac{2\pi}{N} < \frac{\pi}{13}$. Then we get $N > 26$. Therefore, if we perform a radix-2 FFT, we need to zero-pad the sequence into length of 32. For a radix-3 FFT we need to zero-pad it into length of 27. According to the complexity of radix-3 FFT in (c), it will be $\boxed{3N \log_3 N|_{N=27} = 243}$ complex multiply-adds. For the radix-2 FFT, if we follow the same assumption that 1 or -1 are complex multiply, then each radix-2 butterfly unit will have four complex multiplication (since there are four connections) and two complex adds (one add for each output), so there are 4 complex multiply-adds for each radix-2 butterfly. Then the complexity for radix-2 FFT will be $\boxed{4 \times \frac{N}{2} \log_2 N = 2N \log_2 N|_{N=32} = 320}$ complex multiply-adds, which is even larger than the complexity of the radix-3 FFT algorithm in this case.

Remark: Here for the radix-2 FFT any complexity that is $O(N \log_2 N)$ will be acceptable. Anyway, the keypoint is that in some case radix-3 FFT can have less complexity than radix-2 FFT.

Problem 2: Throughput and Energy Consumption of an FFT Core

You have at your disposal a $N = 8$-point radix-2 DIT-based FFT core built from real multipliers and 2-input real adders with the following parameters: Delay ($T_m = 0.5\,\text{ns}$; $T_a = 0.25\,\text{ns}$) and energy consumption ($E_m = 400\,\text{fJ}$; $E_a = 100\,\text{fJ}$).

(a) Sketch the block diagram of a radix-2 butterfly unit in terms of complex multipliers and complex adders and compute its delay T_{BF} and energy consumption E_{BF}.

(b) If one butterfly unit of Part (a) is available, how much time T_{FFT} does it take to compute an 8-point FFT? What is its energy consumption E_{FFT}?

(c) If infinite number of butterfly units of Part(a) are available, how much time does it take to compute an 8-point FFT? What is its energy consumption E_{FFT}?

Solution:

(a) The block diagram of a radix-2 butterfly unit is following:

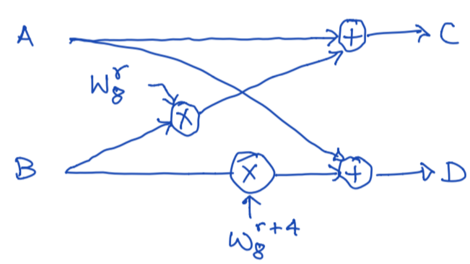


Figure 2: The block diagram of a radix-2 butterfly unit.

To determine the delay, note that the critical path will be a complex multiply and followed by a complex add. Consider we have enough number of adds and multipliers. For a complex multiply the delay is $T_m + T_a$ (we first multiply the real and imaginary part then sum then together), and for a complex add the delay is just T_a. Totally the delay will be $\boxed{T_{BF} = T_m + 2T_a = 1\text{ns}}$. To determine the energy consumption we just need to count how many real multiply and add operations we have. That is $\boxed{E_{BF} = 2 \times (4E_m + 2E_a) + 2 \times 2E_a = 4000\text{fJ} = 4\text{pJ}}$, where each complex multiply requires four real multiplies and twp real adds and we have two complex multipliers. Each complex add requires tow real adds. That is how the Energy cost comes from.

(b) We notice that the N-point radix-2 FFT has $\log_2 N$ stages and $\frac{N}{2} \times \log_2 N$ number of butterfly units. So for $N = 8$ we have three stages and $4 \log_2 8 = 12$ butterfly units. Here since only one butterfly unit is available. The time delay will be $\boxed{T_{FFT} = 12T_{BF} = 12\text{ns}}$. The energy consumption will be $\boxed{E_{FFT} = 12E_{BF} = 48\text{pJ}}$.

(c) Here since we have infinite number of butterfly units, the time delay will just depends on the number of stages. That is $\boxed{T_{FFT} = 3T_{BF} = 3\text{ns}}$. The energy consumption does not change since there are still 12 complex multiply-adds, that is $\boxed{E_{FFT} = 12E_{BF} = 48\text{pJ}}$.

Problem 3: Overlap-Add Method

You have at your disposal one $N = 256$-point radix-2 FFT core that can compute one 256-point FFT in $T_{FFT} = 24\,\text{ns}$ and while consuming $E_{FFT} = 7\,\text{nJ}$. Assume that you have an infinite number of complex multipliers and 2-input complex adders. Furthermore, a complex multiplier takes $T_{CM} = 2\,\text{ns}$ time and consumes $E_{CM} = 4\,\text{pJ}$ of energy, and a 2-input complex adder takes $T_{CA} = 1\,\text{ns}$ of time and consumes $E_{CA} = 1\,\text{pJ}$ of energy. This FFT core and the available arithmetic units is to be used to implement a linear convolution $y[n] = x[n] * h[n]$ of an infinite-length signal $x[n]$ and a $M = 120$-tap filter with impulse response $h[n]$ using the overlap-add method. Now answer the following:

(a) Determine the maximum length L of the finite-length segments $x_m[n]$ that would be needed.

(b) Determine the total delay T_{256} and energy E_{256} needed to compute 256 samples of $y[n]$.

(c) Estimate the delay and energy to compute 256 samples of the output $y[n]$ directly in the time-domain using a *transversal filter structure*. Compare your answer with the one obtained in Part (c).

(d) if $\{x[n]\}_{n=0}^{\infty} = \{1, -2, 3, \frac{1}{2}, 2, -7, \frac{1}{4}, \frac{1}{9}, \ldots\}$ and $\{h[n]\}_{n=0}^{2} = \{1, -1, 1\}$ use the overlap-add method to calculate the linear convolution $h[n] * x[n]$ using segments of length $L = 4$.

Solution:

(a) To compute the convolution of two sequences with length L and M respectively, the point of FFT should be greater or equal to $L + M - 1$. Therefore we have

$$\boxed{L + M - 1 = L + 119 \leq 256, \quad \Rightarrow \quad L \leq 137.} \tag{3}$$

Therefore $L = 137$ will be the maximum length.

(b) Suppose we use the maximum length $L = 137$, in order to compute 256 samples of $y[n]$, consider the following diagram:

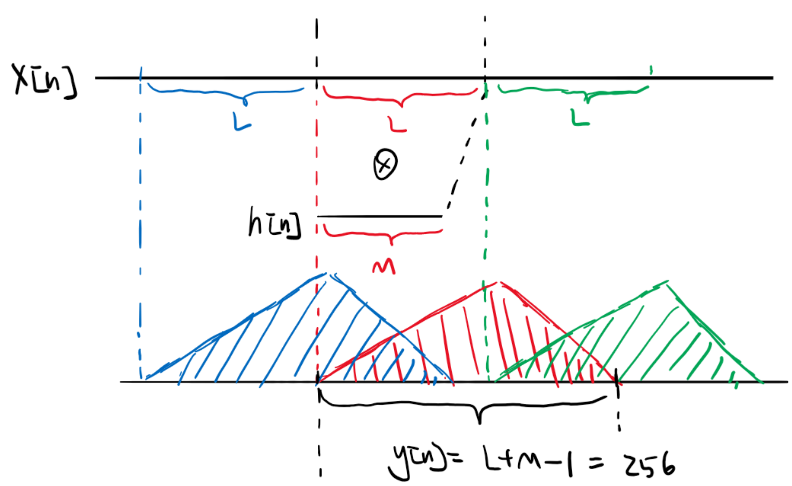


Figure 3: An illustration of determining a length of 256 $y[n]$ by overlap-add method.

In this figure we can see that, in order to determine the 256 samples $y[n]$ shown in black, we need to **at least** compute three convolutions shown in blue, red and green, respectively, which are overlapped and add them together. Therefore, for each convolution the time delay will be

$$T_Y = 3T_{FFT} + T_{CM} = 74\text{ns}, \tag{4}$$

where the $3T_{FFT}$ is because we need to perform 256-pt FFT on both $x_m[n]$ and $h[n]$ to get $x_m[k]$ and $h[k]$, and do the inverse 256-pt FFT on $x_m[k]h[k]$. The T_{CM} comes from the fact that we need to compute $x_m[k]h[k]$, here since we have infinite number of multipliers, the multiplication can be performed in parallel. To sum the three overlaps together, the total delay will be

$$\boxed{T_{256} = 3T_Y + T_{CA} = 223\text{ns},} \tag{5}$$

here we can perform the summation in parallel, then the time delay will just be T_{CA} due to the infinite number of adders.

To determine the energy we follow the same way. The energy cost for each convolution will be

$$E_Y = 3E_{FFT} + N \times E_{CM} = 31\text{nJ}, \tag{6}$$

where the factor $N = 256$ is because the operation $x_m[k]h[k]$ has N number of multiplications. Then the total energy will be

$$\boxed{E_{256} = 3E_Y + (M-1) \times E_{CA} \approx 93.1\text{nJ}.} \tag{7}$$

We can see that most energy is consumed by the FFTs.

Remark: Actually, note that in the solution above, we have computed the 256-pt FFT of the impulse response $h[n]$ three times. So a cleverer way to do this is to only compute it once. If we follow this strategy, we can save the time and energy for computing two 256-pt FFTs. The new time delay and energy will be

$$\boxed{T_{256,\text{new}} = T_{256} - 2T_{FFT} = 175\text{ns}, \quad E_{256,\text{new}} = E_{256} - 2E_{FFT} = 79.1\text{nJ}.} \tag{8}$$

Both results should be acceptable, but we will definitely advocate the one with less time delay and energy consumption.

(c)An example of the transversal filter structure will be as following:

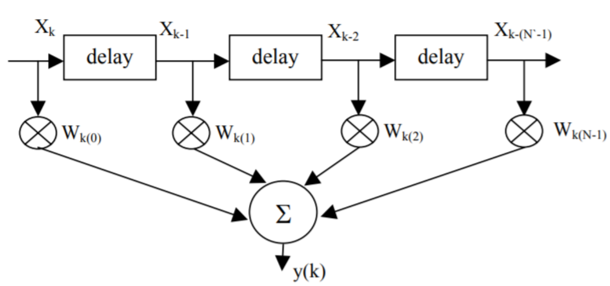


Figure 4: An example of the transversal filter structure. Figure copied from `https://www.ee.ucl.ac.uk/lcs/previous/LCS2002/LCS133.pdf`.

In the figure above the W_k will be $h[n]$ in our case. Therefore, to determine one sample of the 256 points $y[n]$, the time delay will be

$$T_s = T_{CM} + 119 \times T_{CA} = 121\text{ns}, \tag{9}$$

where the T_{CM} comes from the fact that we can do the multiplications in parallel, the factor 119 is because we need to sum the 120 terms together. So the total time delay for the 256 points sequence will be

$$\boxed{T_{\text{tran}} = 256 \times T_s = 31\mu\text{s}.} \tag{10}$$

To determine the energy consumption, we see that for each sample the energy cost will be $E_s = 120 \times E_{CM} + 119 \times E_{CA})$, then the total energy consupmtion will be

$$\boxed{E_{\text{tran}} = 256 \times E_s \approx 153.6\text{nJ}.} \tag{11}$$

We can see that the time-domain direct implementation will be $\frac{E_{\text{tran}}}{E_{256}} = 1.65\times$ more energy cost and $\frac{T_{\text{tran}}}{T_{256}} = 139\times$ slower!

Remark: Actually, when summing the 120 terms together, a better way to do this is to first sum those terms in pairs and then sum those pairs recurrently, this will reduce the time delay to $O(\log_2(N))$ where $N = 120$.

(d) Given $L = 4$, in order to determine the result $\{y[n]\}_{n=0}^{7}$ we just need two segments of length $L = 4$ in this case, which are

$$\{x_0[n]\}_{n=0}^{3} = \left\{1, -2, 3, \frac{1}{2}\right\}, \quad \{x_1[n]\}_{n=0}^{3} = \left\{2, -7, \frac{1}{4}, \frac{1}{9}\right\}. \tag{12}$$

Then we compute the convolution with $h[n]$, each result has length $L + M - 1 = 6$. That is,

$$\begin{aligned} \{y_0[n]\}_{n=0}^{5} &= \{x_0[n]\}_{n=0}^{3} * \{h[n]\}_{n=0}^{2} = \left\{1, -3, 6, -\frac{9}{2}, \frac{5}{2}, \frac{1}{2}\right\}, \\ \{y_1[n]\}_{n=0}^{5} &= \{x_1[n]\}_{n=0}^{3} * \{h[n]\}_{n=0}^{2} = \left\{2, -9, \frac{37}{4}, -\frac{257}{36}, -\frac{5}{36}, \frac{1}{9}\right\}. \end{aligned} \tag{13}$$

Next, overlapping the two convolutions by $M - 1 = 2$ samples, we get

$$\boxed{\{y[n]\}_{n=0}^{7} = \left\{1, -3, 6, -\frac{9}{2}, \frac{9}{2}, -\frac{17}{2}, \frac{37}{4}, -\frac{257}{36}\right\}.} \tag{14}$$

In order to determine $y[n]$ for $n \geq 8$ we need more information about the input $x[n]$. One can directly verify the answer by direct convolution between $\{x[n]\}_{n=0}^{7}$ and $\{h[n]\}_{n=0}^{2}$.

Problem 4: Filter Structures

We wish to implement an IIR filter with poles at $z = \frac{2}{3}e^{\pm j\frac{2\pi}{3}}, -\frac{2}{3}$ and zeros at $z = 2, -3$. Assume that each real multiplier takes $T_m = 2\,\text{ns}$ and each 2-input real adder takes $T_a = 1\,\text{ns}$ to complete. In the following, no complex multipliers or adders are to be used.

(a) Write down the transfer function $H(z)$ of this IIR filter in terms of first and second order terms with real-valued coefficients.

(b) Sketch the block diagrams of Direct Form-I, II, and the transpose structure and calculate their throughputs.

(c) Sketch the block diagram of a 2-stage cascade form structure using a cascade of two Direct Form II structures, and calculate its throughput.

Solution:

(a) By grouping the given poles and zeros, the transfer function can be given as

$$H(z) = \frac{(z-2)(z-3)}{\left(z+\frac{2}{3}\right)\left(z-\frac{2}{3}e^{j\frac{2\pi}{3}}\right)\left(z-\frac{2}{3}e^{-j\frac{2\pi}{3}}\right)} = \boxed{\frac{(z-2)(z+3)}{\left(z+\frac{2}{3}\right)\left(z^2+\frac{2}{3}z+\frac{4}{9}\right)}}. \tag{15}$$

(b) From the result in part (a), one can further express $H(z)$ as

$$H(z) = \frac{(z-2)(z+3)}{\left(z+\frac{2}{3}\right)\left(z^2+\frac{2}{3}z+\frac{4}{9}\right)} = \frac{\left(1+z^{-1}-6z^{-2}\right)z^{-1}}{\left(1+\frac{4}{3}z^{-1}+\frac{8}{9}z^{-2}+\frac{8}{27}z^{-3}\right)} = \frac{N(z)}{D(z)}. \tag{16}$$

Direct Form-I structure: Let us denote the middle stage $w(z) = x(z)N(z)$, then $y(z) = \frac{w(z)}{D(z)}$. Therefore we can perform inverse z-transform and get

$$w[n] = x[n-1] + x[n-2] - 6x[n-3], \quad y[n] = -\frac{4}{3}y[n-1] - \frac{8}{9}y[n-2] - \frac{8}{27}y[n-3] + w[n]. \tag{17}$$

Then the Direct-Form I structure is given as

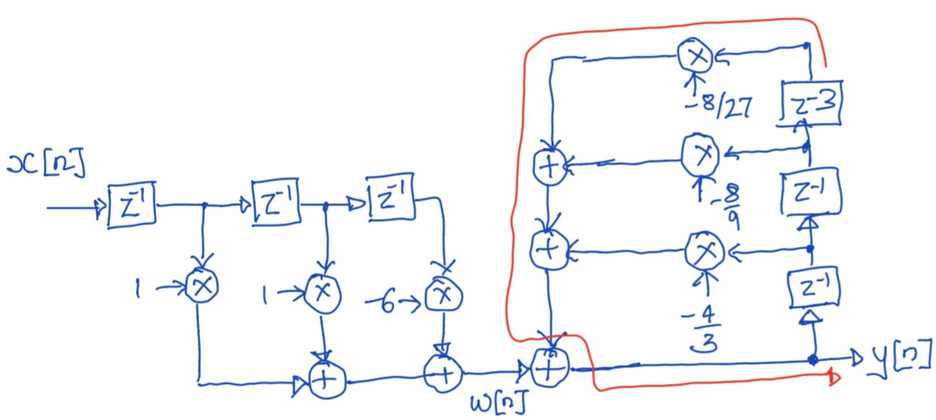


Figure 5: Direct-Form I structure.

The critical path is shown in red. The critical path delay is given as

$$\boxed{T_{cp} = T_m + 3T_a = 5\text{ns}, \quad \frac{1}{T_{cp}} = \frac{1}{5\text{ns}} = 200\text{MHz}.} \tag{18}$$

Direct Form-II structure: Similarly, in this case the middle sage is given as $W(z) = \frac{X(z)}{D(z)}$ and $Y(z) = W(z)N(z)$. Then by applying inverse z-transform we get

$$w[n] = x[n] - \frac{4}{3}w[n-1] - \frac{8}{9}w[n-2] - \frac{8}{27}w[n-3], \quad y[n] = w[n-1] + w[n-2] - 6w[n-3]. \tag{19}$$

Then the Direct-Form II structure is given as

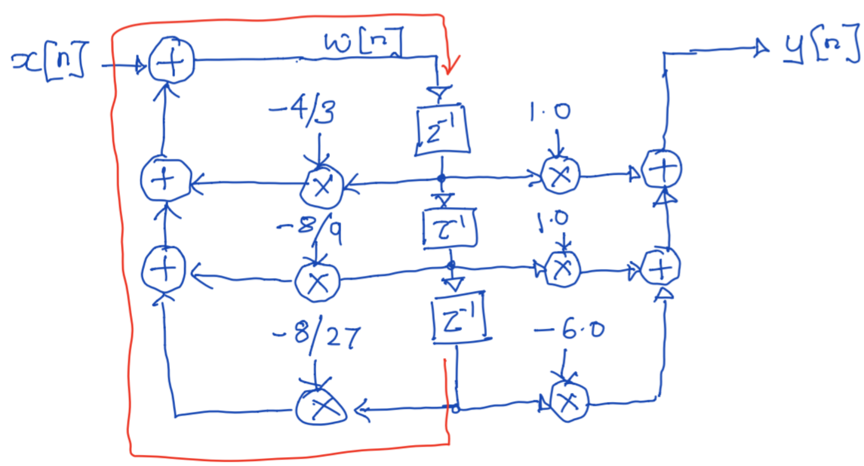


Figure 6: Direct-Form II structure.

The time delay for the critical path shown in red is

$$\boxed{T_{cp} = T_m + 3T_a = 5\text{ns}, \quad \frac{1}{T_{cp}} = \frac{1}{5\text{ns}} = 200\text{MHz}.} \tag{20}$$

Transpose structure: Let us consider $\frac{Y(z)}{X(z)} = \frac{N(z)}{D(z)}$ directly, the we perform inverse z-transform on $Y(z)D(z) = X(z)N(z)$ and get

$$y[n] + \frac{4}{3}y[n-1] + \frac{8}{9}y[n-2] + \frac{8}{27}y[n-3] = x[n-1] + x[n-2] - 6x[n-3]. \tag{21}$$

By simplifying we get

$$y[n] = x[n-1] + x[n-2] - 6x[n-3] - \frac{4}{3}y[n-1] - \frac{8}{9}y[n-2] - \frac{8}{27}y[n-3] \tag{22}$$

Then the transpose structure can be given as

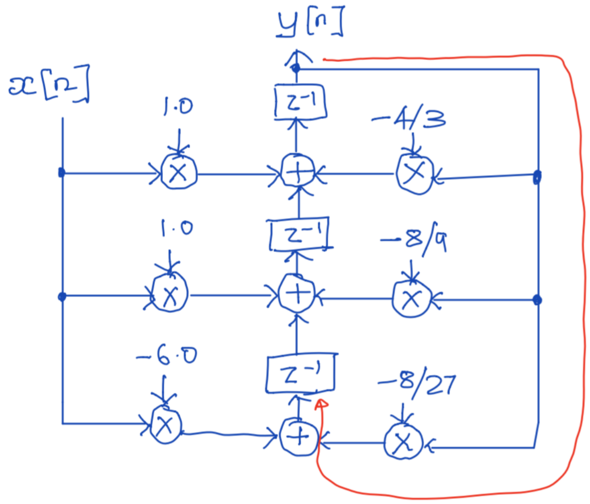


Figure 7: Transpose structure.

And the critical path delay is

$$\boxed{T_{cp} = T_m + T_a = 3\text{ns}, \quad \frac{1}{T_{cp}} = \frac{1}{3\text{ns}} = 333.3\text{MHz}.} \tag{23}$$

Note that the transpose structure has the highest throughput (333.3MHz), and this throughput is independent to the filter order.

(c) To give a 2-stage cascade form structure, we can write the transfer function $H(z)$ as a product of two factors as follows:

$$\begin{aligned} H(z) &= \frac{(z-2)(z-3)}{\left(z+\frac{2}{3}\right)\left(z^2+\frac{2}{3}z_{\frac{4}{9}}\right)} = \left(\frac{z-2}{z+\frac{2}{3}}\right) \times \left(\frac{z+3}{z^2+\frac{2}{3}z+\frac{4}{9}}\right) \\ &= \underbrace{\left(\frac{1-2z^{-1}}{1+\frac{2}{3}z^{-1}}\right)}_{H_1(z)} \times \underbrace{\left(\frac{z^{-1}+3z^{-2}}{1+\frac{2}{3}z^{-1}+\frac{4}{9}z^{-2}}\right)}_{H_2(z)} \\ &= H_1(z)H_2(z) \end{aligned} \tag{24}$$

Then we can employ the Direct Form-II structure for both $H_1(z)$ and $H_2(z)$ and get

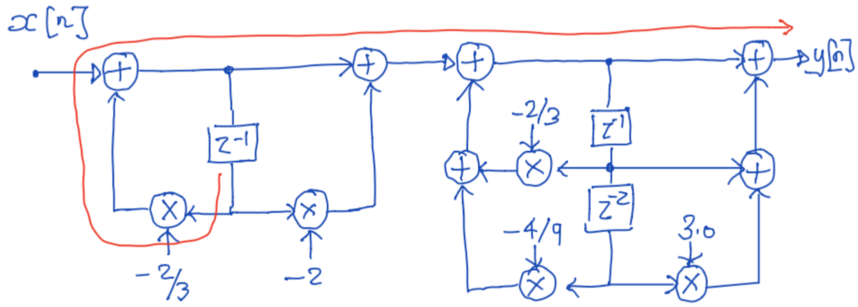


Figure 8: 2-stage cascade structure by using the Direct Form-II structure.

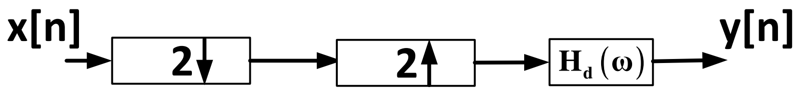
The critical path delay is given as

$$\boxed{T_{cp} = T_m + 5T_a = 7\text{ns}, \quad \frac{1}{T_{cp}} = \frac{1}{7\text{ns}} = 142.8\text{MHz}.} \tag{25}$$

Note that there are actually many different ways to group the numerator and denominator terms of $H(z)$ in a cascade structure. Answer with other ways should also be acceptable.

Problem 5:

Consider the system shown above. The frequency response $H_d(\omega)$ is given by,



$$H_d(\omega) = \begin{cases} 1 & |\omega| \leq \pi/2 \\ 0 & \text{otherwise} \end{cases}$$

Find the output $y[n]$ for the following input sequences,

1. $x[n] = \cos(\frac{\pi}{4}n)$
2. $x[n] = \cos(\frac{3\pi}{4}n)$

Solution:

1. Let us consider the frequency response in the intermediate stages as following:

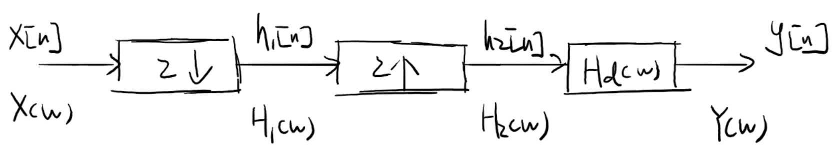


Figure 9: The intermediate stages

A good way to handle upsampling and downsampling will be on the frequency domain, so we would like to find out the frequency response of each intermediate stage. That is,

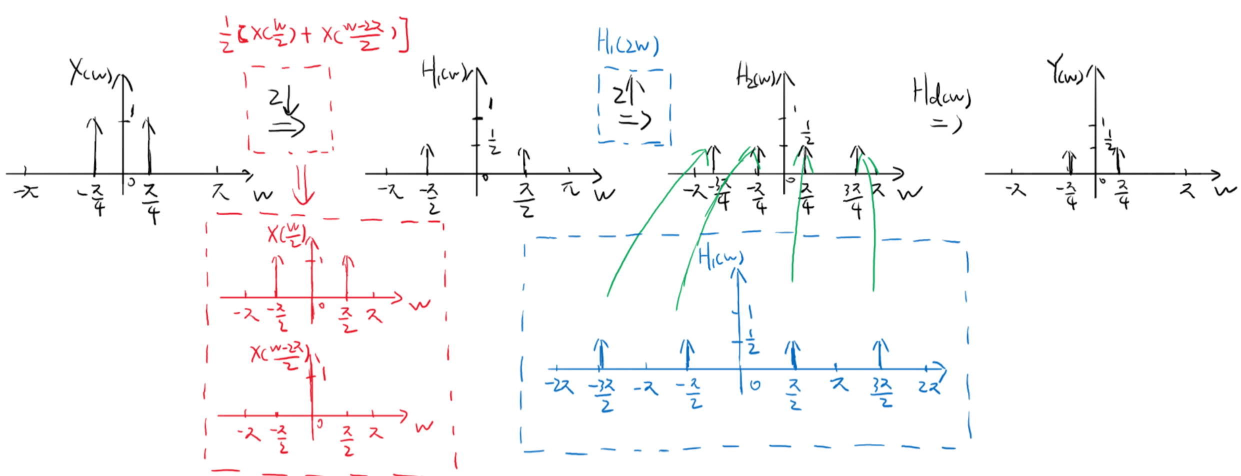


Figure 10: Part 1: The sketch of frequency response of the input, the output and the intermediate stages.

In the first stage $h_1[n]$ that $h_1[n] = x[2n]$, we know that the frequency response of $h_1[n]$ will be $H_1(\omega) = \frac{1}{2}\left(X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega - 2\pi}{2}\right)\right)$. Therefore we have drawn the two components in red in Figure 10. Note that the component $X\left(\frac{\omega-2\pi}{2}\right)$ has no contribution to the frequency range $[-\pi, \pi]$. Next we pass $h_1[n]$ through an upsampling by 2 and get $h_2[n]$, that will make $H_2(\omega) = H_1(2\omega)$, note that there will be two more copies emerge within the $[-\pi, \pi]$ due to the scaling. Then we pass $h_2[n]$ through the filter $H_d(\omega)$ and get the output $y[n]$. It is not hard to see that

$$\boxed{y[n] = \frac{1}{2}\cos\left(\frac{\pi}{4}n\right),} \tag{26}$$

which is the same as the input $x[n]$ with a factor $\frac{1}{2}$.

2. Similar to the part 1, we can do the same thing as following:

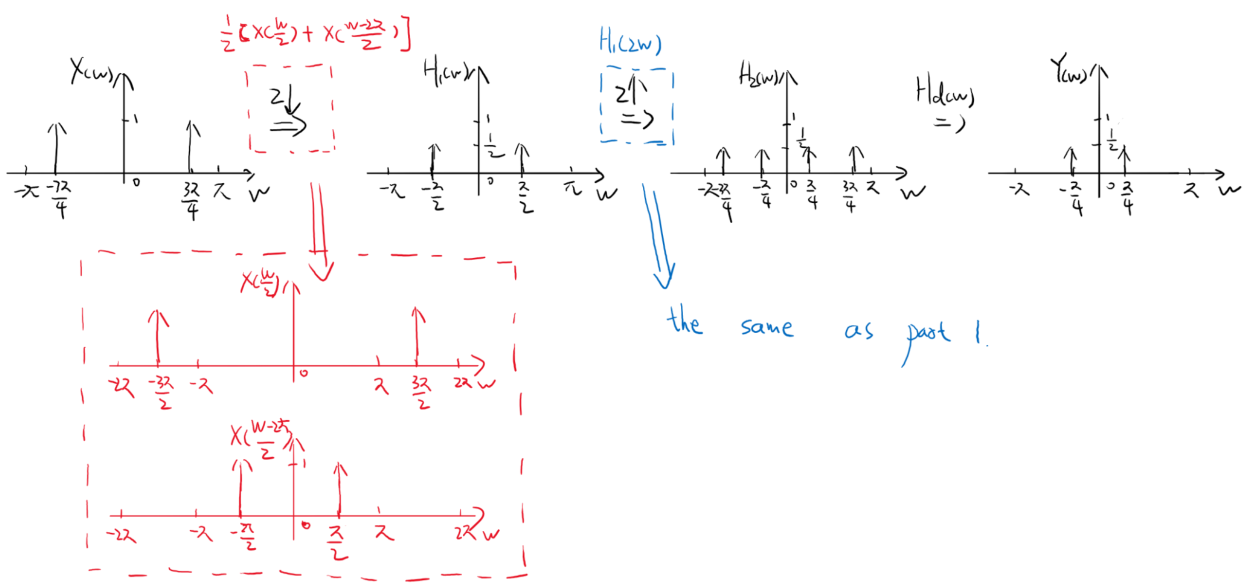


Figure 11: Part 2: The sketch of frequency response of the input, the output and the intermediate stages.

It is not hard to see that

$$\boxed{y[n] = \frac{1}{2}\cos\left(\frac{\pi}{4}n\right).} \tag{27}$$

which is the same as the part 1.

Problem 6:

Given an input $x[n]$, let $w[n]$ and $v[n]$ be given by,

$$w[n] = x[nD]$$

$$v[n] = \begin{cases} x[n/L] & n = Lk \\ 0 & otherwise. \end{cases}$$

Given that,

$$X_d(\omega) = \begin{cases} 1 - \frac{4|\omega|}{\pi} & |\omega| \leq \pi/4 \\ 0 & \pi/4 \leq |\omega| \leq \pi \end{cases}$$

1. Sketch $V_d(\omega)$ for $L = 3$.
2. Sketch $W_d(\omega)$ for $D = 4$.

Solution:

1. Notice that $v[n]$ is just the upsampling of $x[n]$ by $L = 3$, then we have $V_d(\omega) = X_d(L\omega) = X_d(3\omega)$. Therefore we have the following:

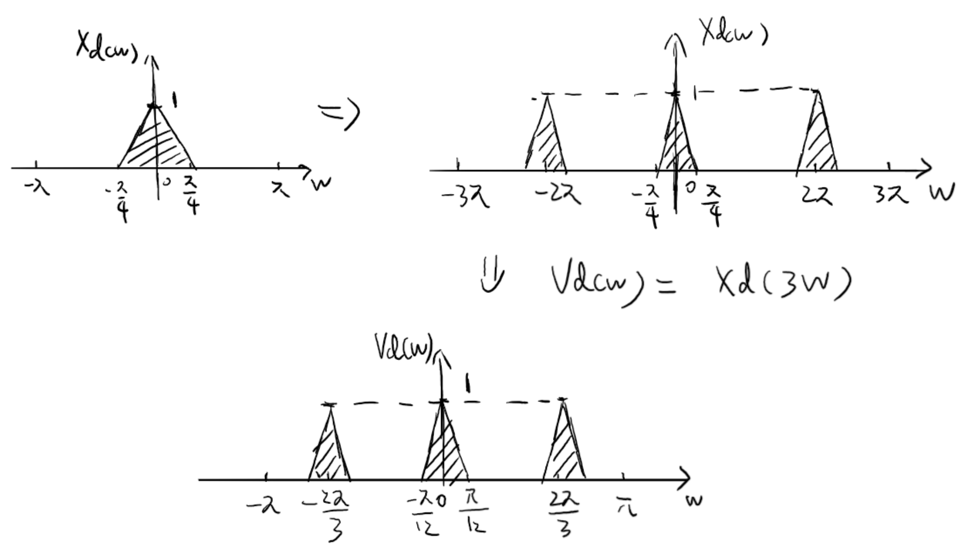


Figure 12: The sketch of $V_d(\omega)$

We see that within the range $[-\pi, \pi]$ of $V_d(\omega)$, the components come from $X_d(\omega)$ within the range $[-L\pi, L\pi]$, in our case that is $[-3\pi, 3\pi]$.

2. Notice that $w[n]$ is just the downsampling of $x[n]$ by $D = 4$. Then according to the formula:

$$W_d(\omega) = \frac{1}{D} \sum_{k=1}^{D-1} X_d\left(\frac{\omega - 2\pi k}{D}\right). \tag{28}$$

In this case that is,

$$V_d(\omega) = \frac{1}{4}\left[X_d\left(\frac{\omega}{4}\right) + X_d\left(\frac{\omega-2\pi}{4}\right) + X_d\left(\frac{\omega-4\pi}{4}\right) + X_e\left(\frac{\omega-6\pi}{4}\right)\right] \tag{29}$$

Based on this we can sketch the result as

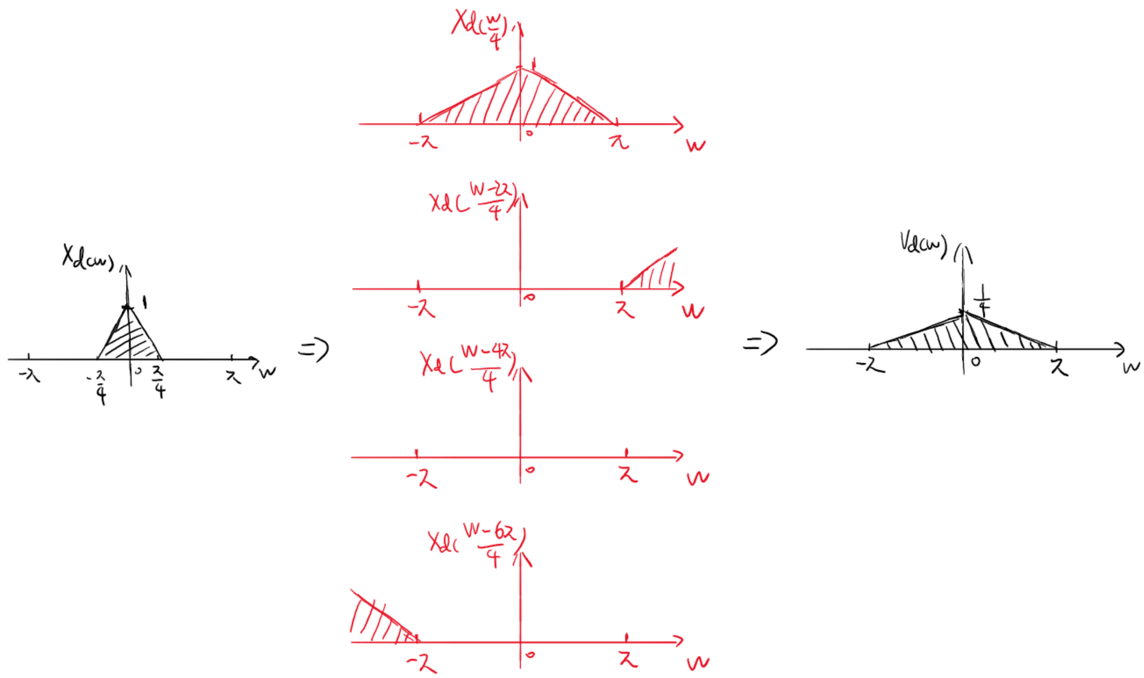
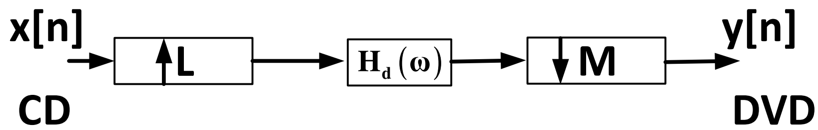


Figure 13: The sketch of $W_d(\omega)$

Note that the three components $X_d\left(\frac{\omega-2\pi}{4}\right)$, $X_d(\frac{\omega-4\pi}{4})$ and $X_d(\frac{\omega-6\pi}{4})$ do not have any contribution to the frequency within the range $[-\pi,\pi]$.

Problem 7:

You desire to transfer music stored on CD to a DVD. CD players operate at a rate of 44.1 kHz, while DVD players operate at 48 kHz. Consider the following system to perform this operation:



Find the smallest possible values for L and M, and sketch the frequency response $H_d(\omega)$ to perform this conversion.

Solution:

In this problem, sampling rate must be converted from 44.1kHz to 48kHz by using the upsampling and downsampling, note that the rate is not an integer, so we need to find out the integer L and M that make $\frac{L}{M} = \frac{48}{44.1}$. To do this, notice that:

$$48000 = 2^7 \times 3 \times 5, \quad 44100 = 2^2 \times 3^2 \times 5^2 \times 3^2. \tag{30}$$

Then we have

$$\frac{L}{M} = \frac{48000}{44100} = \frac{2^7 \times 3 \times 5}{2^2 \times 3^2 \times 5^2 \times 3^2} = \frac{160}{147}. \tag{31}$$

Therefore, the smallest possible value for L and M will be $\boxed{L = 160}$ and $\boxed{M = 147}$.

To determine the frequency response of $H_d(\omega)$, note that the input of $H_d(\omega)$ will be $X_d(L\omega) = X_d(160\omega)$, so we only need to keep the frequency response of $X_d(160\omega)$ within the range $[-\frac{\pi}{160}, \frac{\pi}{160}]$. Moreover, since when we do downsampling by M later the amplitude will be scaled by the factor $\frac{1}{M}$, so the filter amplitude should be scaled by M to compensate this effect. Then we can sketch $H_d(\omega)$ as

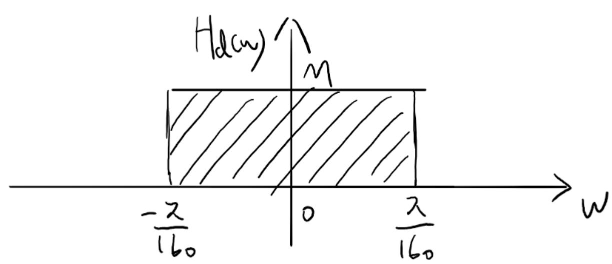


Figure 14: The sketch of the frequency response of $H_d(\omega)$.

Problem 8:

The frequency response of a generalized linear phase (GLP) filter can be expressed as $H_d(\omega) = R(\omega)e^{j(\alpha-\beta\omega)}$ where $R(\omega)$ is a real function and α and β are constants. For each of the following LTI systems, described by their impulse response, transfer function, or difference equation, determine whether it is a GLP filter. If it is, determine $R(\omega)$, α, and β, and indicate whether it is also a strictly linear phase filter.

(a) $y[n] = -x[n] + x[n-2]$

(b) $y[n] = 0.4y[n-1] + 0.4x[n]$

(c) $y[n] = 2x[n] + x[n-1] + x[n-2] + 2x[n-3]$

(d) $\{h[n]\}_{n=0}^{2} = \{4, 5, 6\}$

(e) $\{h[n]\}_{n=0}^{2} = \{3, 4, 3\}$

(f) $H(z) = -1 + 4z^{-1} + z^{-2}$

Solution:

(a) This filter **is GLP** since its impulse response has odd symmetry. To see this, we can first write the its z-transform as (or we can directly perform DTFT):

$$Y(z) = -X(z) + X(z)z^{-2}. \tag{32}$$

Therefore, the transfer function is given as:

$$H(z) = \frac{Y(z)}{X(z)} = -1 + z^{-2}. \tag{33}$$

One can see that the impulse response is $\{h[n]\}_{n=0}^{2} = \{-1, 0, 1\}$, so it has odd symmetry. The frequency response is then given as

$$\begin{aligned} H_d(\omega) = H(z)|_{z=e^{j\omega}} &= -1 + e^{-2j\omega} \\ &= -e^{-j\omega}(e^{j\omega} - e^{-j\omega}) \\ &= -e^{-j\omega}(2j\sin(\omega)) \\ &= 2\sin(\omega)e^{-j\left(\frac{\pi}{2}+\omega\right)}. \end{aligned} \tag{34}$$

Therefore,

$$\boxed{R(\omega) = 2\sin(\omega), \quad \alpha = -\frac{\pi}{2}, \quad \beta = -1}. \tag{35}$$

However, it **is not a strictly linear phase filter**, since $R(\omega)$ has a zero-crossing at $\omega = 0$, which corresponding to jump of π in the phase.

(b) Similarly, let us apply the z-transform and get the transfer function as

$$Y(z) = 0.4Y(z)z^{-1} + 0.4X(z), \Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{0.4}{1 - 0.4z^{-1}}. \tag{36}$$

Then we get the frequency response as

$$H_d(\omega) = H(z)|_{z=e^{j\omega}} = \frac{0.4}{1-0.4e^{-j\omega}} = \frac{0.4(1-0.4e^{j\omega})}{1.16-0.8\cos(\omega)}. \tag{37}$$

There are two ways check if the filter is GLP or not. First, one can see that the phase response of this filter is $P(\omega)$ that

$$P(\omega) = \arctan\left(-\frac{0.4\sin(\omega)}{1-0.4\cos(\omega)}\right). \tag{38}$$

Clearly, this is not a linear function of ω, this the filter **is not GLP**. Another way to check it is that, according to the transfer function $H(z)$, it has a nontrivial pole $z = 0.4$, therefore this filter is IIR, and an IIR filter cannot exhibit GLP.

(c) Again, this filter **is GLP** since its impulse response has even symmetry. To see this, we can check the frequency response as (or we can directly perform DTFT on both sides):

$$H(Z) = \frac{Y(z)}{X(z)} = 2 + z^{-1} + z^{-2} + 2z^{-3}, \Rightarrow H_d(\omega) = 2 + e^{-j\omega} + e^{-2j\omega} + 2e^{-3j\omega}. \tag{39}$$

The impulse response $\{h[n]\}_{n=0}^{3} = \{2, 1, 1, 2\}$, which has even symmetry. Then by further simplifying

$$\begin{aligned} H_d(\omega) &= e^{-1.5j\omega}(2e^{1.5j\omega} + e^{0.5j\omega} + e^{-0.5j\omega} + 2e^{-1.5j\omega}) \\ &= e^{-1.5j\omega}(4\cos(1.5\omega) + 2\cos(0.5\omega)) \\ &= 2(2\cos(1.5\omega) + \cos(0.5\omega))e^{-1.5j\omega}. \end{aligned} \tag{40}$$

Therefore,

$$\boxed{R(\omega) = 2(2\cos(1.5\omega) + \cos(0.5\omega)), \quad \alpha = 0, \quad \beta = \frac{3}{2}.} \tag{41}$$

However, similar to the part (a), this filter **is not a strictly linear phase filter**, since the there will be some zero-crossing points that makes a jump of π in phase.

(d) Clearly, this filter **is not GLP** since there is no symmetry in its impulse response.

(e) This filter **is GLP** since the impulse response has even symmetry. We apply the DTFT on the impulse response and get

$$\begin{aligned} H_d(\omega) &= 3 + 4e^{-j\omega} + 3e^{-2j\omega} \\ &= e^{-j\omega}(3e^{j\omega} + 4 + 3e^{-j\omega}) \\ &= (4 + 6\cos(\omega))e^{-j\omega}. \end{aligned} \tag{42}$$

Therefore,

$$\boxed{R(\omega) = 4 + 6\cos(\omega), \quad \alpha = 0, \quad \beta = 1.} \tag{43}$$

However, this filter **is not a strictly linear phase filter**, since $R(\omega)$ has some zero-crossing points which make jump of π on the phase response.

(f) One can get the impulse response from the transfer function as $\{h[n]\}_{n=0}^{2} = \{-1, 4, 1\}$, The impulse response $h[n]$ has odd symmetry but the middle coefficient is nonzero, which prevents $H_d(\omega)$ from being expressed as $H_d(\omega) = R(\omega)e^{j(\alpha-\beta\omega)}$. Therefore, the filter **is not GLP**.