Chapter 11

11.1 Analysis yields

\[ w_1[n] = \beta_1 \{ x[n-1] + \alpha_2 w_2[n-1] \}, \quad w_2[n] = w_1[n] + w_3[n], \]

\[ w_3[n] = \beta_1 \{ \alpha_2 w_2[n-1] + \alpha_2 w_4[n-1] \}, \quad w_4[n] = w_3[n] + w_5[n], \]

\[ w_5[n] = \beta_3 \{ \alpha_2 w_4[n-1] + \alpha_3 w_5[n-1] \}, \quad y[n] = w_1[n] + a_0 x[n]. \]

In matrix form the above equations can be written as

\[
\begin{bmatrix}
w_1[n] \\
w_2[n] \\
w_3[n] \\
w_4[n] \\
w_5[n] \\
y[n]
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_1[n] \\
w_2[n] \\
w_3[n] \\
w_4[n] \\
w_5[n] \\
y[n]
\end{bmatrix} +
\begin{bmatrix}
\alpha_1 \beta_1 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_2 & \alpha_2 \beta_2 & 0 & 0 & 0 \\
0 & \alpha_3 \beta_2 & \alpha_3 \beta_2 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_1[n-1] \\
w_2[n-1] \\
w_3[n-1] \\
w_4[n-1] \\
w_5[n-1] \\
y[n-1]
\end{bmatrix} +
\begin{bmatrix}
\beta_1 x[n-1] \\
0 \\
0 \\
0 \\
0 \\
\alpha_0 x[n]
\end{bmatrix}
\]

Here the \( F \) matrix is given by:

\[
F =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Since the \( F \) matrix contains nonzero entries above the main diagonal, the above set of equations are not computable.

11.2 A computable set of equations are given by:

\[ w_1[n] = \beta_1 \{ x[n-1] + \alpha_1 w_2[n-1] \}, \quad w_3[n] = \beta_1 \{ \alpha_2 w_2[n-1] + \alpha_2 w_4[n-1] \}, \]

\[ w_5[n] = \beta_3 \{ \alpha_2 w_4[n-1] + \alpha_3 w_5[n-1] \}, \quad w_2[n] = w_1[n] + w_3[n], \quad y[n] = w_1[n] + a_0 x[n], \]

\[ w_4[n] = w_3[n] + w_5[n]. \]

In matrix form the above equations can be written as

\[
\begin{bmatrix}
w_1[n] \\
w_2[n] \\
w_3[n] \\
w_4[n] \\
w_5[n] \\
y[n]
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_1[n] \\
w_2[n] \\
w_3[n] \\
w_4[n] \\
w_5[n] \\
y[n]
\end{bmatrix} +
\begin{bmatrix}
\alpha_1 \beta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_2 & \alpha_2 \beta_2 & 0 & 0 & 0 & 0 \\
0 & \alpha_3 \beta_2 & \alpha_3 \beta_2 & 0 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0 & 0 \\
0 & \alpha_2 \beta_3 & \alpha_3 \beta_3 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_1[n-1] \\
w_2[n-1] \\
w_3[n-1] \\
w_4[n-1] \\
w_5[n-1] \\
y[n-1]
\end{bmatrix} +
\begin{bmatrix}
\beta_1 x[n-1] \\
0 \\
0 \\
0 \\
0 \\
\alpha_0 x[n]
\end{bmatrix}
\]

Here the \( F \) matrix is given by:

\[
F =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]
Since the F matrix has no nonzero entries above the main diagonal, the new set of equations are computable.

11.3 Analysis yields

\[ w_1[n] = \beta_1 \{ \alpha_1 w_2[n-1] + \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \]
\[ w_2[n] = w_1[n] + \alpha x[n] \]
\[ w_3[n] = \beta_2 \{ \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \]
\[ w_4[n] = w_2[n] + w_3[n] \]
\[ w_5[n] = \beta_3 \{ \alpha_3 w_6[n-1] \} \]
\[ w_6[n] = w_4[n] + w_5[n] \]
\[ y[n] = \alpha_1 w_2[n] + \alpha_2 w_4[n] + \alpha_3 w_6[n] + \alpha_0 x[n] \]

In matrix form the above set of equations are given by

\[
\begin{bmatrix}
  w_1[n] \\
  w_2[n] \\
  w_3[n] \\
  w_4[n] \\
  w_5[n] \\
  w_6[n] \\
  y[n]
\end{bmatrix} =
\begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & a_1 & a_2 & a_3 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
  w_1[n-1] \\
  w_2[n-1] \\
  w_3[n-1] \\
  w_4[n-1] \\
  w_5[n-1] \\
  w_6[n-1] \\
  y[n-1]
\end{bmatrix} + 
\begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & a_1 & a_2 & a_3 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
  \alpha_0 x[n] \\
  \alpha_1 x[n] \\
  \alpha_2 x[n] \\
  \alpha_3 x[n] \\
  \alpha_5 x[n] \\
  \alpha_6 x[n] \\
\end{bmatrix}
\]

Here the F matrix is given by: \( F = \)

\[
\begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 1 & 0 & 0 \\
  a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Since the diagonal of the F matrix has all zeros, and no nonzero entries above the main diagonal, the new set of equations are computable.

11.4

Reduced signal flow graph obtained by removing the branches going out of the input node and the delay branches is:
The node sets of the precedence graph are as follows:
\( \{ \mathcal{N}_1 \} = \{ w_1[n], w_3[n], w_5[n] \}, \{ \mathcal{N}_2 \} = \{ w_2[n], w_4[n], y[n] \} \)

In the above precedence graph, \( \mathcal{N}_1 \) contains only outgoing nodes and \( \mathcal{N}_2 \) contains only incoming branches. The structure has no delay free loops. A valid computational algorithm is:

\[
\begin{align*}
\alpha_1 & = w_1[n] + w_2[n] + w_3[n] + w_4[n] + w_5[n] + w_6[n] \\
\alpha_2 & = w_2[n] + w_3[n] + w_4[n] + w_5[n] + w_6[n] + y[n] \\
\alpha_3 & = w_3[n] + w_4[n] + w_5[n] + w_6[n] + y[n] \\
\end{align*}
\]

In matrix form we have

The node sets of the precedence graph are as follows: \( \{ \mathcal{N}_1 \} = \{ w_1[n], w_3[n], w_5[n] \}, \{ \mathcal{N}_2 \} = \{ w_2[n] \}, \{ \mathcal{N}_3 \} = \{ w_4[n] \}, \{ \mathcal{N}_4 \} = \{ w_6[n] \}, \{ \mathcal{N}_5 \} = \{ y[n] \} \).

A valid computational algorithm is:

\[
\begin{align*}
\alpha_1 & = \beta_1 \{ \alpha_1 w_2[n-1] + \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \\
\alpha_2 & = \beta_2 \{ \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \\
\alpha_3 & = \beta_3 \{ \alpha_3 w_6[n-1] \} \\
\end{align*}
\]

\[
\begin{align*}
w_2[n] & = w_1[n] + x[n] \\
w_4[n] & = w_2[n] + w_3[n] \\
w_6[n] & = w_4[n] + w_5[n] \\
y[n] & = \alpha_1 w_2[n] + \alpha_2 w_4[n] + \alpha_3 w_6[n] + \alpha_0 x[n] \\
\end{align*}
\]

11.5 Reduced signal flow graph is:

A valid computational algorithm is:

\[
\begin{align*}
w_1[n] & = \beta_1 \{ \alpha_1 w_2[n-1] + \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \\
w_3[n] & = \beta_2 \{ \alpha_2 w_4[n-1] + \alpha_3 w_6[n-1] \} \\
w_5[n] & = \beta_3 \{ \alpha_3 w_6[n-1] \} \\
w_2[n] & = w_1[n] + x[n] \\
w_4[n] & = w_2[n] + w_3[n] \\
w_6[n] & = w_4[n] + w_5[n] \\
y[n] & = \alpha_1 w_2[n] + \alpha_2 w_4[n] + \alpha_3 w_6[n] + \alpha_0 x[n] \\
\end{align*}
\]

11.6 (a) Analysis yields \( w_1[n] = w_2[n] - k_1 w_1[n-1], \ w_2[n] = w_3[n] - k_2 s_2[n-1], \)

\[
\begin{align*}
w_3[n] & = x[n] - k_3 s_3[n-1], \ s_2[n] = k_1 w_1[n] + w_1[n-1], \ s_3[n] = k_2 w_2[n] + s_2[n-1], \\
y[n] & = k_3 w_3[n] + s_3[n-1]. \end{align*}
\]

In matrix form we have
As the diagonal elements of $F$-matrix are all zeros, there are no delay-free loops. However, the above set of equations are not computable as there are non-zero elements above the diagonal of $F$.

(b) The reduced signal flow-graph representation of Figure P11.3 is shown below:

From the above flow-graph we observe that the set composed of nodes with only outgoing branches is $\mathcal{N}_1 = \{w_3[n]\}$. The set of nodes with incoming branches from $\mathcal{N}_1$ and outgoing branches is $\mathcal{N}_2 = \{w_2[n]\}$. The set of nodes with incoming branches from $\mathcal{N}_1$ and $\mathcal{N}_2$ and outgoing branches is $\mathcal{N}_3 = \{w_1[n]\}$. Finally, the set of nodes with only incoming branches from $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$, and no outgoing branches $\mathcal{N}_4 = \{s_2[n], s_3[n], y[n]\}$. Therefore, one possible ordered set of equations that is computable is given by

$\begin{bmatrix} w_3[n] \\ w_2[n] \\ w_1[n] \\ s_2[n] \\ s_1[n] \\ y[n] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 & 0 & 0 \\ k_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_3[n] \\ w_2[n] \\ w_1[n] \\ s_2[n] \\ s_1[n] \\ y[n] \end{bmatrix}$

$+ \begin{bmatrix} -k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_3 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_3[n-1] \\ w_2[n-1] \\ w_1[n-1] \\ s_2[n-1] \\ s_1[n-1] \\ y[n-1] \end{bmatrix}$

Note: All elements on the diagonal and above diagonal of $F$ are zeros.
11.7 \[ H(z) = \frac{p_0 + p_1z^{-1} + p_2z^{-2}}{1 - 3z^{-1} + 1.5z^{-2}}. \] From Eq. (11.18) we get,
\[
\begin{bmatrix}
p_0 \\
p_1 \\
p_2
\end{bmatrix} = \begin{bmatrix}
3.2 & 0 & 0 \\
5.6 & 3.2 & 0 \\
7.0 & 5.6 & 3.2
\end{bmatrix} \begin{bmatrix}
1 \\
-3 \\
1.5
\end{bmatrix} = \begin{bmatrix}
3.2 \\
-4.0 \\
-5.0
\end{bmatrix}
\]
Hence, \[ H(z) = \frac{3.2 - 4z^{-1} - 5z^{-2}}{1 - 3z^{-1} + 1.5z^{-2}}. \]

11.8 From Eqn (11.15), \[ d_1 = \begin{bmatrix}
-4 & -2 \\
8 & 4
\end{bmatrix} \begin{bmatrix}
8 \\
-12
\end{bmatrix} = \begin{bmatrix}
-0.25 \\
3.5
\end{bmatrix} \]
Finally, from Eqn (11.21), \[ d_1 = \begin{bmatrix}
2 \\
1
\end{bmatrix} \]

11.9 From Eqn. (11.15)
\[
\begin{bmatrix}
p_0 \\
p_1 \\
p_2 \\
p_3
\end{bmatrix} = \begin{bmatrix}
2 & 0 & 0 & 0 \\
-4 & 2 & 0 & 0 \\
8 & -4 & 2 & 0 \\
-8 & 8 & -4 & 2
\end{bmatrix} \begin{bmatrix}
d_1 \\
d_2 \\
d_3
\end{bmatrix}. \]
Next, from Eqn. (11.20), \[ d_1 = \begin{bmatrix}
12 \\
16 \\
-8 \\
16
\end{bmatrix} \]
Finally, from Eqn. (11.21), \[ d_1 = \begin{bmatrix}
0.4 \\
-4.8 \\
-7.4
\end{bmatrix} \]

Hence,
\[ H(z) = \frac{2 - 3.2z^{-1} - 0.4z^{-3}}{1 + 0.4z^{-1} - 7.4z^{-3}}. \]
\[
\begin{bmatrix}
p_0 \\
p_1 \\
p_2 \\
p_3 \\
p_4
\end{bmatrix} = \begin{bmatrix}
2 & 0 & 0 & 0 & 1 \\
-4 & 2 & 0 & 0 & -0.6 \\
4 & -4 & 2 & 0 & 0.2 \\
-6 & -6 & -4 & 2 & 1.8 \\
-8 & 6 & 4 & -2 & 2
\end{bmatrix}\begin{bmatrix}
1 \\
\frac{1}{2} \\
\frac{1}{2} \\
\frac{1}{2} \\
\frac{1}{2}
\end{bmatrix} = \begin{bmatrix}
2 \\
-5.2 \\
6.8 \\
-5.6
\end{bmatrix}.
\]
Hence, \( P(z) = 2 - 5.2z^{-1} + 6.8z^{-2} - 5.6z^{-3} \).

\[
\begin{bmatrix}
p_0 \\
p_1 \\
p_2 \\
p_3 \\
p_4
\end{bmatrix} = \begin{bmatrix}
2 & 0 & 0 & 0 & 1 \\
-2 & 2 & 0 & 0 & \frac{1}{2} \\
4 & -2 & 2 & 0 & 0 \\
6 & 4 & -2 & 2 & 0 \\
-8 & 6 & 4 & -2 & 2
\end{bmatrix}\begin{bmatrix}
1 \\
\frac{1}{2} \\
\frac{1}{2} \\
\frac{1}{2} \\
-\frac{3}{2}
\end{bmatrix} = \begin{bmatrix}
2 \\
-\frac{7}{2} \\
2 \frac{1}{2} \\
4 \frac{1}{2} \\
-3 \frac{1}{2}
\end{bmatrix}.
\]
\( \{p_i\} = \{2, -\frac{7}{2}, 2 \frac{1}{2}, 4 \frac{1}{2}, -3 \frac{1}{2}\} \)

11.12 The \( k \)-th sample of an \( N \)-point DFT is given by \( X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} \). Thus, the computation of \( X[k] \) requires \( N \) complex multiplications and \( N - 1 \) complex additions. Now, each complex multiplication, in turn, requires 4 real multiplications and 2 real additions. Likewise, each complex addition requires 2 real additions. As a result, the \( N \) complex multiplications needed to compute \( X[k] \) require a total of \( 4N \) real multiplications and a total of \( 2N - 2 \) real additions. Therefore, each sample of the \( N \)-point DFT involves \( 4N \) real multiplications and \( 4N - 2 \) real additions. Hence, the computation of all DFT samples thus requires \( 4N^2 \) real multiplications and \( (4N - 2)N \) real additions.

11.13 Let the two complex numbers be \( \alpha = a + jb \) and \( \beta = c + jd \). Thus, \( \alpha\beta = (a + jb)(c + jd) = (ac - bd) + j(ad + bc) \), which requires 4 real multiplications and 2 real additions. Consider the product \( (a + b)(c + d), ac, \) and \( bd, \) which require 3 real multiplications and 2 real additions. The imaginary part of \( \alpha\beta \) can be formed from \( (a + b)(c + d) - ac - bd = ad + bc, \) which now requires 2 real additions. Likewise, the real part of \( \alpha\beta \) can be formed from \( ac - bd \) requiring an additional real additions. Hence, the complex multiplication can be computed using 3 real multiplications and 5 real additions.

11.14 Recall, \( W_N = e^{-j2\pi/N} = c + js, \) where \( c = \cos(2\pi/N) \) and \( s = -\sin(2\pi/N). \) Thus, \( c^2 + s^2 = 1. \) Now, \( \Psi_{r+1} = W_N \cdot \Psi_r = (c + js)(\Re\{\Psi_r\} + j\Im\{\Psi_r\}) \)
\[
= (c \cdot \Re\{\Psi_r\} - s \cdot \Im\{\Psi_r\}) + j(c \cdot \Im\{\Psi_r\} + s \cdot \Re\{\Psi_r\}).
\]
Thus, \( \Re\{\Psi_{r+1}\} = c \cdot \Re\{\Psi_r\} - s \cdot \Im\{\Psi_r\} \) and \( \Im\{\Psi_{r+1}\} = c \cdot \Im\{\Psi_r\} + s \cdot \Re\{\Psi_r\} \).

Figure P11.4 with internal node label is shown below. Its analysis yields

---

Not for sale
(1): \( U = \Re(\Psi_r) + \frac{c-1}{s} \Im(\Psi_r) \), \quad (2): \( \Im(\Psi_{r+1}) = \Im(\Psi_r) + sU \), and

(3): \( \Re(\Psi_{r+1}) = U + \frac{c-1}{s} \Im(\Psi_{r+1}) \). Substituting Eqs. (1) in Eq. (2) we get

(4): \( \Im(\Psi_{r+1}) = \Im(\Psi_r) + s\left(\frac{c-1}{s} \Im(\Psi_r) + \Re(\Psi_r)\right) = c \cdot \Im(\Psi_r) + s \cdot \Re(\Psi_r) \). Next, substituting Eqs. (1) and (4) in Eq. (3) we get (5):

(5): \( \Re(\Psi_{r+1}) = \frac{c-1}{s}\left(c \cdot \Im(\Psi_r) + s \cdot \Re(\Psi_r)\right) + \frac{c-1}{s} \Im(\Psi_r) + \Re(\Psi_r) \)

\[
= \frac{c^2 - 1}{s} \Im(\Psi_r) + c \cdot \Re(\Psi_r) = \frac{-s^2}{s} \cdot \Im(\Psi_r) + c \cdot \Re(\Psi_r) = c \cdot \Re(\Psi_r) - s \cdot \Im(\Psi_r).
\]

It thus follows that the structure of Figure P11.4 implements the multiplication of a complex signal with the twiddle factor using only 3 real multiplications.

In the case of multiplication by complex twiddle factor \( W_N^{-1} = c - js \), we have

\[
\Psi_{r+1} = W_N^{-1} \cdot \Psi_r = (c - js) \left( \Re(\Psi_r) + j \Im(\Psi_r) \right)
\]

\[
= (c \cdot \Re(\Psi_r) + s \cdot \Im(\Psi_r)) + j(c \cdot \Im(\Psi_r) - s \cdot \Re(\Psi_r)).
\]

Thus, here

\[
\Re(\Psi_{r+1}) = c \cdot \Re(\Psi_r) + s \cdot \Im(\Psi_r) \quad \text{and} \quad \Im(\Psi_{r+1}) = c \cdot \Im(\Psi_r) - s \cdot \Re(\Psi_r).
\]

It follows then that the real and imaginary parts are simply obtained by reversing the sign of \( s \) of the real and imaginary parts derived in the case of multiplication by \( W_N \). As a result, the corresponding structure is obtained by cascading an inverter to the real multipliers in Figure P11.4 as indicated below:

11.15 The center frequency bin \( k : f_c(k) = \frac{kF_T}{N} \), where \( N = \# \) of bins, and \( F_T \) is the sampling frequency. Inverting we have \( f(k) = \left\lfloor \frac{f F_T}{N} \right\rfloor \). Therefore, the absolute difference from one of the given four tones (150 Hz, 375 Hz, 620 Hz, and 850 Hz) to the center of its bin is given by

\[
dist(N, f) = \left| f - \frac{F_T}{N} \cdot \left\lfloor \frac{f}{F_T} \right\rfloor \right|.
\]

It follows from this equation that the distance goes to zero if

\[
\frac{F_T}{N} \cdot \left\lfloor \frac{f}{F_T} \right\rfloor = f \quad \text{or} \quad \frac{f}{F_T} \text{ is an integer.}
\]
The total distance is reduced to zero if \( \frac{f_i}{F_T} \) is an integer for \( i = 1, \ldots, 4 \). The minimum value of \( N \) for which this true is 500. However, the total distance can be small, but nonzero, for significantly smaller values of \( N \).

11.16 \( H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \). Hence, \( Y(z) = \frac{X(z)}{1 - W_N^{-k} z^{-1}} = \frac{1 + z^{-N/2}}{1 - W_N^{-k} z^{-1}} = V(z) + z^{-N/2} V(z) \), where \( V(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \). Or, in other words, \( y[n] = v[n] + v[n - N/2] \).

Consider \( k = 1 \): Then \( V(z) = \frac{1}{1 - W_N^{-1} z^{-1}} \). This implies,

\[
v[n] = W_N^{-n} \mu[n] = \left\{ 1, \ W_N^{-1}, \ W_N^{-2}, \ \ldots, \ W_N^{-N/2}, \ W_N^{-(N+1)/2}, \ \ldots \right\}
\]

\[
= \left\{ 1, \ W_N^{-1}, \ W_N^{-2}, \ \ldots, \ W_N^{-N-1}, \ \ldots \right\}
\]
since \( W_N^{-N/2} = -1, W_N^{-(N+1)/2} = -W_N^{-1} \), and so on. Thus, \( v[n - N/2] = \{0, 0, 0, \ldots, 1, W_N^{-1}, W_N^{-2}, \ldots \} \). Hence,

\[
y[n] = v[n] + v[n - N/2] = \left\{ 1, \ W_N^{-1}, \ W_N^{-2}, \ \ldots, \ W_N^{-(N-1)/2}, \ 0, 0, 0, \ \ldots \right\}
\]

Now, consider \( k = N/2 \): \( V(z) = \frac{1}{1 - W_N^{-N/2} z^{-1}} \). This implies,

\[
v[n] = W_N^{-(N/2)n} \mu[n] = \left\{ 1, \ W_N^{-N/2}, \ W_N^{-N}, \ \ldots, \ W_N^{N/2}, \ W_N^{-(N+1)/2} N/2, \ \ldots \right\}
\]

Thus, \( v[n - N/2] = \{0, 0, 0, \ldots, 0, 1, W_N^{-N/2}, W_N^{-N}, \ldots \} \)

Now, \( W_N^{N/2} = (-1)^{N/2}, W_N^{-N/2} = -1, W_N^{-N} = 1, W_N^{-(N+1)/2} N/2 = (-1)^{N/2} (-1) \), etc. Hence,

\[
y[n] = \begin{cases} 1, & -1, \ 1, & -1, \ \ldots, \ 1 + (-1)^{N/2}, & 1 + (-1)^{N/2}, \ \ldots \ \end{cases}
\]

Therefore, if \( N/2 \) is even, \( y[n] = \left\{ 1, \ -1, \ 1, \ -1, \ \ldots, \ 2, \ -2, \ 2, \ \ldots \right\} \), and if \( N/2 \) is odd,
\[ y[n] = \begin{cases} 1, & n = N/2 \\ -1, & n = N/2 + 1 \\ 1, & n = N \\ -1, & n = N + 1 \end{cases} \]

11.17


\[ W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \]


11.18


\[ W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \quad W_N^0 \]


11.19

\[ X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \]

\[ = \sum_{n=0}^{N/r_1} x[n r_1] W_N^{nk r_1} + \sum_{n=0}^{N/r_1} x[n r_1 + 1] W_N^{nk r_1 + k} + \ldots + \sum_{n=0}^{N/r_1} x[n r_1 + r_1 - 1] W_N^{nk r_1 + (r_1 - 1)k} \]
Thus, if the \((N / \tau_1)\)-point DFT has been calculated, we need at the first stage an additional \((\tau_1 - 1)\) multiplications to compute one sample of the \(N\)–point DFT \(X[k]\) and, as a result, additional \((\tau_1 - 1)N\) multiplications are required to compute all \(N\) samples of the \(N\)–point DFT. Decomposing it further, it follows that additional \((\tau_2 - 1)N\) multiplications are needed at the second stage, and so on. Therefore, the total number of multiply (add) operations
\[
= (\tau_1 - 1)N + (\tau_2 - 1)N + \ldots + (\tau_v - 1)N = \left(\sum_{i=1}^{v} \tau_i - v\right)N.
\]

11.20 An examination of the flow-graph of the 8-point DIT FFT algorithm shown in Figure 11.24 reveals that in the first stage all twiddle factors are \(W_N^0 = 1\) and in the second stage the twiddle factors are either \(W_N^0 = 1\) or \(W_N^{N/4} = j\). Hence, there are no twiddle factors with nonunity magnitudes in the first two stages. In all succeeding stages, one of the twiddle factors is \(W_N^0 = 1\) and another one is \(W_N^{N/4} = j\). The number of twiddle factors with nonzero magnitudes in the \(i\)-th stage, \(i \geq 3\), is \(2^i - 2\). Hence, the total number of twiddle factors with nonzero magnitudes in an \(N\)-point radix-2 FFT algorithm is
\[
R(v) = \left(\sum_{i=3}^{v} 2^{i-1}\right) - 2(v - 2), \ v \geq 3, \ \text{where} \ N = 2^v.
\]

11.21 Direct computation of \(M\) samples of an \(N\)-point DFT requires \(M^2\) multiplications, whereas, the Radix-2 FFT algorithm requires \(\frac{N}{2} \log_2 N\) multiplications. In order for a \(N\)-point radix-2 FFT algorithm to be computationally more efficient than a direct computation of \(M\) samples of an \(N\)-point DFT, the following inequality must hold:
\[
M > \left\lfloor \frac{\sqrt{\log_2 N}}{2} \right\rfloor.
\]
a) \(N = 512, M = 49\), b) \(N = 1024, M = 72\), c) \(N = 2048, M = 107\)

11.22 \(X(z) = X_0(z^3) + z^{-1}X_1(z^3) + z^{-2}X_2(z^3)\). Thus, the \(N\)–point DFT can be expressed as
\[
X[k] = X_0[(k)_{N/3}] + W_N^k X_1[(k)_{N/3}] + W_N^{2k} X_2[(k)_{N/3}].
\]
Hence, the structural interpretation of the first stage of the radix-3 DFT is as indicated below:
11.23  \[ X[k] = \sum_{n=0}^{8} x[n]W_9^n = \left( x[0]W_9^{0-k} + x[3]W_9^{3k} + x[6]W_9^{6k} \right) \]
\[ + \left( x[2]W_9^{2k} + x[5]W_9^{5k} + x[8]W_9^{8k} \right), \]
where \( G_0(\langle k \rangle_3) = x[0]W_3^{0-k} + x[3]W_3^{3k} + x[6]W_3^{6k} \)
and \( G_1(\langle k \rangle_3) = x[1]W_3^{1-k} + x[4]W_3^{4k} + x[7]W_3^{7k} \),
and \( G_2(\langle k \rangle_3) = x[2]W_3^{2k} + x[5]W_3^{5k} + x[8]W_3^{8k} \),
are 3-point DFTs. A flow-graph representation of this radix-3 DIT FFT computation scheme is shown below,

where the twiddle factors for computing the DFT samples are indicated below for a typical DFT sample:
In the above diagram, the 3-point DFT computation is carried out as indicated below:

\[
X[k] = \sum_{n=0}^{14} x[n]W_{15}^{nk} = \left( x[0]W_{15}^{0k} + x[3]W_{15}^{3k} + x[6]W_{15}^{6k} + x[9]W_{15}^{9k} + x[12]W_{15}^{12k} \right) \\
= G_0(\langle k \rangle_5) + G_1(\langle k \rangle_5)W_{15}^{k} + G_2(\langle k \rangle_5)W_{15}^{2k}, \text{ where}
\]

\[
G_0(\langle k \rangle_5) = x[0]W_{5}^{0k} + x[3]W_{5}^{3k} + x[6]W_{5}^{6k} + x[9]W_{5}^{9k} + x[12]W_{5}^{12k},
\]

\[
G_1(\langle k \rangle_5) = x[1]W_{5}^{1k} + x[4]W_{5}^{4k} + x[7]W_{5}^{7k} + x[10]W_{5}^{10k} + x[13]W_{5}^{13k}, \text{ and}
\]

\[
\]

A flow-graph representation of this mixed-radix DIT FFT computation scheme is shown below:
Now, by definition, \( q[n] = \text{Im}[X[n]] + j\text{Re}[X[n]] \). Its \( N \)-point DFT is given by

\[
Q[k] = \sum_{n=0}^{N-1} q[n]W_N^{nk}
\]

Thus,
(1): \( \text{Re}\{Q(k)\} = \sum_{m=0}^{N-1} \text{Im}\{X[n]\} \cos\left(\frac{2\pi mk}{N}\right) + \text{Re}\{X[n]\} \sin\left(\frac{2\pi mk}{N}\right) \bigg|_{k = n} \),

(2): \( \text{Im}\{Q[k]\} = \sum_{m=0}^{N-1} -\text{Im}\{X[n]\} \sin\left(\frac{2\pi mk}{N}\right) + \text{Re}\{X[n]\} \cos\left(\frac{2\pi mk}{N}\right) \bigg|_{k = n} \).

From the definition of the inverse DFT we observe \( x[k] = \frac{1}{N} \sum_{m=0}^{N-1} X[m]W_{N}^{-mnk} \). Hence,

(3): \( \text{Re}\{x[k]\} = \frac{1}{N} \sum_{m=0}^{N-1} \left[ \Re\{X[m]\} \cos\left(\frac{2\pi mk}{N}\right) - \Im\{X[m]\} \sin\left(\frac{2\pi mk}{N}\right) \right] \bigg|_{k = n} \),

(4): \( \text{Im}\{x[k]\} = \frac{1}{N} \sum_{m=0}^{N-1} \left[ \Im\{X[m]\} \cos\left(\frac{2\pi mk}{N}\right) + \Re\{X[m]\} \sin\left(\frac{2\pi mk}{N}\right) \right] \bigg|_{k = n} \).

Comparing Eqs. (2) and (3) we get \( \text{Re}\{x[n]\} = \frac{1}{N} \cdot \text{Im}\{Q[k]\}\big|_{k = n} \), and comparing Eqs. (1) and (4) we get \( \text{Im}\{x[n]\} = \frac{1}{N} \cdot \text{Re}\{Q[k]\}\big|_{k = n} \).

11.27 \( r[n] = X[(-n)_{N}] = \begin{cases} X[0], & \text{if } n = 0, \\ X[N-n], & \text{if } n \neq 0. \end{cases} \) Therefore,

\[
R[k] = \sum_{n=0}^{N-1} r[n]W_{N}^{nk} = r[0] + \sum_{n=1}^{N-1} r[n]W_{N}^{nk} = X[0] + \sum_{n=1}^{N-1} X[n]W_{N}^{(N-n)k} = X[0] + \sum_{n=1}^{N-1} X[n]W_{N}^{-nk} = \sum_{n=0}^{N-1} X[n]W_{N}^{-nk} = N \cdot x[k]. \]

Thus, \( x[n] = \frac{1}{N} \cdot R[k]\big|_{k = n} \).

11.28 Let \( y[n] \) denote the result of convolving a length-\( L \) sequence \( x[n] \) with a length-\( N \) sequence \( h[n] \). The length of \( y[n] \) is then \( L + N - 1 \). Here \( L = 16 \) and \( N = 9 \), hence length of \( y[n] \) is 24.

Method #1: Direct linear convolution - For a length-\( L \) sequence \( x[n] \) and a length-\( N \) sequence \( h[n] \),

\[
\# \text{ of real mult.} = 2 \sum_{n=1}^{N} n + N(L - N - 1) = 2 \sum_{n=1}^{9} n + 9(16 - 9 - 1) = 135.
\]

Method #2: Linear convolution via circular convolution - Since \( y[n] \) is of length 24, to get the correct result we need to pad both sequences with zeros to increase their lengths to 24 before carrying out the circular convolution.

\[
\# \text{ of real mult.} = 24 \times 24 = 576.
\]
Method #3: Linear convolution via radix-2 FFT - The process involves computing the 16-point FFT $G[k]$ of the length-16 complex sequence $g[n] = x[n] + j h_e[n]$, where $h_e[n]$ is obtained by zero-padding $h[n]$ to length 16. Then, the 16-point DFTs, $X[k]$ and $H_e[k]$, of $x[n]$ and $h_e[n]$, respectively, are recovered from $G[k]$. Finally, the IDFT of the product $Y[k] = X[k] \cdot H_e[k]$ yields $y[n]$.

Now, the first stage of the 16-point radix-2 FFT requires 0 complex multiplications, the second stage requires 0 complex multiplications, the third stage requires 4 complex multiplications, and the last stage requires 6 complex multiplications, resulting in a total of 10 complex multiplications.

- # of complex mult. to implement $G[k] = 10$
- # of complex mult. to recover $X[k]$ and $H_e[k]$ from $G[k] = 0$
- # of complex mult. to form $Y[k] = 16$
- # of complex mult. to form the IDFT of $Y[k] = 10$

Hence, the total number of complex mulit. = 36

A direct implementation of a complex multiplication requires 4 real multiplications resulting in a total of $4 \times 36 = 144$ real multiplications for Method #3. However, if a complex multiply can be implemented using 3 real multiplies (see Problem 11.13), in which case Method #3 requires a total of $3 \times 36 = 108$ real multiplications.

11.29 Let $y[n]$ denote the result of convolving a length-$L$ sequence $x[n]$ with a length-$N$ sequence $h[n]$. The length of $y[n]$ is then $L + N - 1$. Here, $L = 16$ and $N = 10$, hence length of $y[n]$ is 25.

Method #1: Direct linear convolution - For a length-$L$ sequence $x[n]$ and a length-$N$ sequence $h[n]$,

- # of real mult. = $2 \sum_{n=1}^{N} n + N(L - N - 1) = 2 \sum_{n=1}^{10} n + 10(16 - 10 - 1) = 160$.

Method #2: Linear convolution via circular convolution - Since $y[n]$ is of length 24, to get the correct result we need to pad both sequences with zeros to increase their lengths to 24 before carrying out the circular convolution.

- # of real mult. = $25 \times 25 = 625$.

Method #3: Linear convolution via radix-2 FFT - The process involves computing the 16-point FFT $G[k]$ of the length-16 complex sequence $g[n] = x[n] + j h_e[n]$, where $h_e[n]$ is obtained by zero-padding $h[n]$ to length 16. Then, the 16-point DFTs, $X[k]$ and $H_e[k]$, of $x[n]$ and $h_e[n]$, respectively, are recovered from $G[k]$. Finally, the IDFT of the product $Y[k] = X[k] \cdot H_e[k]$ yields $y[n]$.

Now, the first stage of the 16-point radix-2 FFT requires 0 complex multiplications, the second stage requires 0 complex multiplications, the third stage requires 4 complex multiplications, and the last stage requires 6 complex multiplications, resulting in a total of 10 complex multiplications.
multiplications, and the last stage requires 6 complex multiplications resulting in a total of 10 complex multiplications.

# of complex mult. to implement \( G[k] = 10 \)
# of complex mult. to recover \( X[k] \) and \( H_e[k] \) from \( G[k] = 0 \)
# of complex mult. to form \( Y[k] = 16 \)
# of complex mult. to form the IDFT of \( Y[k] = 10 \)

Hence, the total number of complex mult. = 36

A direct implementation of a complex multiplication requires 4 real multiplications resulting in a total of \( 4 \times 36 = 144 \) real multiplications for Method #3. However, if a complex multiply can be implemented using 3 real multiplies (see Problem 11.13), in which case Method #3 requires a total of \( 3 \times 36 = 108 \) real multiplications.

**11.30 (a)** Since the impulse response of the filter is of length 72, the transform length \( N \) should be greater than 72. If \( L \) denotes the number of input samples used for convolution, then \( L = N - 71 \). So for every \( L \) samples of the input sequence, an \( N \) -point DFT is computed and multiplied with an \( N \) -point DFT of the impulse response sequence \( h[n] \) (which needs to be computed only once), and finally an \( N \) -point inverse of the product sequence is evaluated. Hence, the total number \( R_M \) of complex multiplications required (assuming \( N \) is a power-of-2) is given by

\[
R_M = \left\lceil \frac{2048}{N-71} \right\rceil (N \log_2 N + N) + \frac{N}{2} \log_2 N.
\]

It should be noted that in developing the above expression, multiplications due to twiddle factors of values \( \pm 1 \) and \( \pm j \) have not been excluded. The values of \( R_M \) for different values of \( N \) are as follows:

1) For \( N = 128 \), \( R_M = 37312 \),
2) For \( N = 256 \), \( R_M = 28672 \),
3) For \( N = 512 \), \( R_M = 27904 \),
4) For \( N = 1024 \), \( R_M = 38912 \).

Hence, \( N = 512 \) is the appropriate choice for the transform length requiring 27904 complex multiplications or equivalently, \( 27904 \times 3 = 83712 \) real multiplications.

Since the first stage of the FFT calculation process requires only multiplications by \( \pm 1 \), the total number of complex multiplications for \( N = 128 \) is actually

\[
R_M = \left\lceil \frac{2048}{N-71} \right\rceil (N \log_2 N + N) + \frac{N}{2} \log_2 N - \frac{N}{2} = 27648,
\]

or equivalently, \( 27648 \times 3 = 82944 \) real multiplications.

**11.31 (a)**

\[
2 \sum_{n=1}^{N} n + N(N - L - 1) = 2 \sum_{n=1}^{72} n + 72(2048 - 72 - 1) = 147456.
\]
As can be seen from the above, multiplication by each matrix $V_k$, $k = 1, 2, 3$, requires at most 8 complex multiplications.

(b) The transpose of the matrices given in Part (a) are as follows:

$V_8^t = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & W_8^0 & 0 & 0 & 0 & W_8^0 & 0 & 0 \\
0 & W_8^0 & 0 & 0 & 0 & W_8^0 & 0 & 0 \\
0 & 0 & W_8^2 & 0 & 0 & 0 & W_8^2 & 0 \\
0 & 0 & 0 & W_8^3 & 0 & 0 & 0 & W_8^3 \\
\end{bmatrix}$

$V_4^t = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & W_8^2 & 0 & W_8^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}$

$V_2^t = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
W_8^0 & W_8^4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & W_8^0 & W_8^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & W_8^0 & W_8^4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & W_8^0 & W_8^4 \\
\end{bmatrix}$

$E^t = E = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}$

It is easy to show that the flow-graph representation of $D_8 = E^tV_2^tV_4^tV_8^t$ is precisely the 8-point DIF FFT algorithm of Figure 11.28.
\[ X[2\ell] = \sum_{n=0}^{N-1} x[n] W_N^{2\ell n} = \sum_{n=0}^{N-1} x[n] W_N^{2\ell n} + \sum_{n=N/2}^{N-1} x[n] W_N^{2\ell n}, \quad 0 \leq \ell \leq \frac{N}{2} - 1. \]

Replacing \( n \) by \( n + \frac{N}{2} \) in the right-most sum we get
\[ X[2\ell] = \sum_{n=0}^{N-1} x[n] W_N^{2\ell n} + \sum_{n=0}^{N-1} x[n + \frac{N}{2}] W_N^{2\ell n} = \sum_{n=0}^{N-1} \left( x[n] + x[n + \frac{N}{2}] \right) W_N^{2\ell n}/2, \quad 0 \leq \ell \leq \frac{N}{2} - 1. \]

Likewise,
\[ X[4\ell + 1] = \sum_{n=0}^{N-1} x[n] W_N^{(4\ell+1)n} + \sum_{n=0}^{N-1} x[n + N/4] W_N^{(4\ell+1)n} + \sum_{n=0}^{N/2} x[n] W_N^{(4\ell+1)n} + \sum_{n=-3N/4}^{-1} x[n] W_N^{(4\ell+1)n}, \]
where \( 0 \leq \ell \leq \frac{N}{4} - 1 \). Replacing \( n \) by \( n + \frac{N}{4} \) in the second sum, \( n \) by \( n + \frac{N}{2} \) in the third
sum, and \( n \) by \( n + \frac{3N}{4} \) in the fourth sum, we get
\[ X[4\ell + 1] = \sum_{n=0}^{N/4-1} x[n] W_N^{4\ell n} W_N^n + \sum_{n=0}^{N/4-1} x[n + N/4] W_N^{4\ell n} W_N^n + \sum_{n=0}^{N/4-1} x[n + 3N/4] W_N^{4\ell n} W_N^{3N/4}. \]

Now, \( W_N^{2\ell N} = W_N^{3\ell N} = 1, W_N^{N/4} = -j, W_N^{N/2} = -1, \) and \( W_N^{3N/4} = +j \). Therefore,
\[ X[4\ell + 1] = \sum_{n=0}^{N/4-1} \left\{ \left( x[n] - x[n + N/2] \right) - j \left( x[n + N/4] - x[n + 3N/4] \right) \right\} W_N^{\ell n} W_N^{n}/4, \quad 0 \leq \ell \leq \frac{N}{4} - 1. \]

Similarly, \( X[4\ell + 3] = \sum_{n=0}^{N/4-1} x[n] W_N^{4\ell+3n} + \sum_{n=0}^{N/4-1} x[n + N/4] W_N^{4\ell+3n} W_N^{(4\ell+3)N/4} \]
\[ + \sum_{n=0}^{N/4-1} x[n + N/2] W_N^{4\ell+3n} W_N^{(4\ell+3)N/2} + \sum_{n=0}^{N/4-1} x[n + 3N/4] W_N^{4\ell+3n} W_N^{(4\ell+3)3N/4} \]
\[ = \sum_{n=0}^{N/4-1} x[n] W_N^{4\ell n} W_N^{3n} + \sum_{n=0}^{N/4-1} x[n + N/4] W_N^{4\ell n} W_N^{3n} W_N^{\ell n} W_N^{3N/4}. \]
\[
N-1 \\
\sum_{n=0}^{4} x[n + \frac{N}{2}] W_N^{4\ell n} W_N^{3n} W_N^{2\ell N} W_N^{6N/4} + \sum_{n=0}^{N-1} x[n + \frac{N}{4}] W_N^{4\ell n} W_N^{3n} W_N^{\ell N} W_N^{3N/4} \\
= \sum_{n=0}^{N-1} \left\{ (x[n] - x[n + \frac{N}{2}]) + j \left( x[n + \frac{N}{4}] - x[n + \frac{3N}{4}] \right) \right\} W_N^{3n} W_N^{\ell n}, 0 \leq \ell \leq \frac{N}{4} - 1.
\]

The butterfly here is as shown below which is seen to require two complex multiplications.

11.33 From the flow-graph of the 8-point split-radix FFT algorithm given below it can be seen that the total number of complex multiplications required is 2. On the other hand, the total number of complex multiplications required for a standard DIF FFT algorithm is also 2.

11.34 If multiplications by ±j, ±1 are ignored, the flow-graph shown below requires 8 complex multiplications = 24 real multiplications. A radix-2 DIF 16-point FFT algorithm, on the other hand requires 10 complex multiplications = 30 real multiplications.
11.35 (a) \( N = 12 \). Choose \( N_1 = 4 \) and \( N_2 = 3 \). Thus, \( n = n_1 + 4n_2, \begin{cases} 0 \leq n_1 \leq 3, \\ 0 \leq n_2 \leq 2 \end{cases} \), and
\[ k = 3k_1 + k_2, \begin{cases} 0 \leq k_1 \leq 3, \\ 0 \leq k_2 \leq 2 \end{cases} \]
The corresponding index mappings are indicated below:

\[
\begin{array}{c|cccc}
 n_2 & 0 & 1 & 2 & 3 \\
\hline
\end{array}
\begin{array}{c|cccc}
 k_1 & 0 & 1 & 2 & 3 \\
\hline
\end{array}
\]

(b) \( N = 15 \). Choose \( N_1 = 3 \) and \( N_2 = 5 \). Thus, \( n = n_1 + 3n_2, \begin{cases} 0 \leq n_1 \leq 2, \\ 0 \leq n_2 \leq 4 \end{cases} \), and
\[ k = 5k_1 + k_2, \begin{cases} 0 \leq k_1 \leq 2, \\ 0 \leq k_2 \leq 4 \end{cases} \]
The corresponding index mappings are indicated below:

\[
\begin{array}{c|cccc}
 n_2 & 0 & 1 & 2 \\
\hline
\end{array}
\begin{array}{c|cccc}
 k_1 & 0 & 1 & 2 \\
\hline
\end{array}
\]
(c) \( N = 21 \). Choose \( N_1 = 7 \) and \( N_2 = 3 \). Thus, \( n = n_1 + 7n_2, \begin{cases} 0 \leq n_1 \leq 6, \\ 0 \leq n_2 \leq 2, \end{cases} \) and
\[
k = 3k_1 + k_2, \begin{cases} 0 \leq k_1 \leq 6, \\ 0 \leq k_2 \leq 2. \end{cases}
\]
The corresponding index mappings are indicated below:

<table>
<thead>
<tr>
<th>( n_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>( k_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

(d) \( N = 35 \). Choose \( N_1 = 7 \) and \( N_2 = 5 \). Thus, \( n = n_1 + 7n_2, \begin{cases} 0 \leq n_1 \leq 6, \\ 0 \leq n_2 \leq 4, \end{cases} \) and
\[
k = 5k_1 + k_2, \begin{cases} 0 \leq k_1 \leq 6, \\ 0 \leq k_2 \leq 4. \end{cases}
\]
The corresponding index mappings are indicated below:

<table>
<thead>
<tr>
<th>( n_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>( x[21] )</td>
<td>( x[22] )</td>
<td>( x[23] )</td>
<td>( x[24] )</td>
<td>( x[25] )</td>
<td>( x[26] )</td>
<td>( x[27] )</td>
</tr>
<tr>
<td>4</td>
<td>( x[28] )</td>
<td>( x[29] )</td>
<td>( x[30] )</td>
<td>( x[31] )</td>
<td>( x[32] )</td>
<td>( x[33] )</td>
<td>( x[34] )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( k_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

11.36 (a) \( N = 12 \). Choose \( N_1 = 4 \) and \( N_2 = 3 \). \( A = 3, \ B = 4, \ C = 3(3^{-1})_4 = 9, \)
\[ D = 4(4^{-1})_3 = 4. \]
Thus, \( n = \langle 3n_1 + 4n_2 \rangle_{12}, \begin{cases} 0 \leq n_1 \leq 3, \\ 0 \leq n_2 \leq 2, \end{cases} \)
\[ k = \langle 9k_1 + 4k_2 \rangle_{12}, \begin{cases} 0 \leq k_1 \leq 3, \\ 0 \leq k_2 \leq 2. \end{cases} \]
The corresponding index mappings are indicated below:

<table>
<thead>
<tr>
<th>( n_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_2 )</td>
<td>0</td>
<td>( x[0] )</td>
<td>( x[3] )</td>
<td>( x[6] )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( k_1 )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_2 )</td>
<td>0</td>
<td>( X[0] )</td>
<td>( X[9] )</td>
</tr>
</tbody>
</table>

(b) \( N = 15 \). Choose \( N_1 = 3 \) and \( N_2 = 5 \). \( A = 5, \ B = 3, \ C = 5(5^{-1})_3 = 10, \)
\[ D = 3(3^{-1})_5 = 6. \]
Thus, $n = \langle 5n_1 + 3n_2 \rangle_{15}$, \[
\begin{cases}
0 \leq n_1 \leq 2, \\
0 \leq n_2 \leq 4,
\end{cases}
\quad k = \langle 10k_1 + 6k_2 \rangle_{15}, \quad \begin{cases}
0 \leq k_1 \leq 2, \\
0 \leq k_2 \leq 4.
\end{cases}
\]

The corresponding index mappings are indicated below:

\[
\begin{array}{c|ccc}
& n_1 & 0 & 1 & 2 \\
\hline
\end{array}
\quad
\begin{array}{c|ccc}
& k_1 & 0 & 1 & 2 \\
\hline
k_2 & 0 & X[0] & X[10] & X[5] \\
\end{array}
\]

(c) $N = 21$. Choose $N_1 = 7$ and $N_2 = 3$. $A = 3$, $B = 7$, $C = 3(3^{-1})_4 = 3 \times 5 = 15$,

\[D = 7(7^{-1})_3 = 7 \times 1 = 7.\] Thus, $n = \langle 3n_1 + 7n_2 \rangle_{21}$, \[
\begin{cases}
0 \leq n_1 \leq 6, \\
0 \leq n_2 \leq 2,
\end{cases}
\]

$k = \langle 15k_1 + 7k_2 \rangle_{21}, \quad \begin{cases}
0 \leq k_1 \leq 6, \\
0 \leq k_2 \leq 2.
\end{cases}$ The corresponding index mappings are indicated below:

\[
\begin{array}{c|cccccc}
& n_1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\end{array}
\quad
\begin{array}{c|cccccc}
& k_1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\end{array}
\]

(d) $N = 35$. Choose $N_1 = 7$ and $N_2 = 5$. $A = 5$, $B = 7$, $C = 5(5^{-1})_7 = 5 \times 3 = 15$,

\[D = 7(7^{-1})_5 = 7 \times 3 = 21.\] Thus, $n = \langle 5n_1 + 7n_2 \rangle_{35}$, \[
\begin{cases}
0 \leq n_1 \leq 6, \\
0 \leq n_2 \leq 4,
\end{cases}
\]

$k = \langle 15k_1 + 21k_2 \rangle_{35}, \quad \begin{cases}
0 \leq k_1 \leq 6, \\
0 \leq k_2 \leq 4.
\end{cases}$ The corresponding index mappings are indicated below:

\[
\begin{array}{c|cccccc}
& n_1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\end{array}
\quad
\begin{array}{c|cccccc}
& k_1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\end{array}
\]
11.37 \( N = 12 \). Choose \( N_1 = 4 \) and \( N_2 = 3 \). \( A = 3, B = 4, C = 3(3^{-1})_4 = 9, D = 4(4^{-1})_3 = 4 \).

Thus, \( n = (3n_1 + 4n_2)_{12}, \begin{cases} 0 \leq n_1 \leq 3, \\ 0 \leq n_2 \leq 2, \end{cases} \) \( k = (9k_1 + 4k_2)_{12}, \begin{cases} 0 \leq k_1 \leq 3, \\ 0 \leq k_2 \leq 2. \end{cases} \)

The corresponding index mappings are indicated below:

\[
\begin{array}{cccc|cccc}
\hline
n_2 & 0 & 1 & 2 & 3 & k_1 & 0 & 1 & 2 & 3 \\
\hline
\hline
\end{array}
\]

Alternately, \( k = (9n_1 + 4n_2)_{12}, \begin{cases} 0 \leq n_1 \leq 3, \\ 0 \leq n_2 \leq 2, \end{cases} \) \( k = (3k_1 + 4k_2)_{12}, \begin{cases} 0 \leq k_1 \leq 3, \\ 0 \leq k_2 \leq 2. \end{cases} \)

The corresponding index mappings are indicated below:

\[
\begin{array}{cccc|cccc}
\hline
n_2 & 0 & 1 & 2 & 3 & k_1 & 0 & 1 & 2 & 3 \\
\hline
\hline
\end{array}
\]

Hence, \( X[2k] = Y[2k] \), and \( X[2k + 1] = Y[(6 + (2k + 1))_{12}], k = 0, 1, \ldots, 5 \).

11.38 (a) \( N = 6 \). Choose \( N_1 = 2 \) and \( N_2 = 3 \). \( A = 3, B = 2, C = 3(3^{-1})_2 = 3, D = 2(2^{-1})_3 = 4 \).

Thus, \( n = (3n_1 + 2n_2)_{6}, \begin{cases} 0 \leq n_1 \leq 1, \\ 0 \leq n_2 \leq 2, \end{cases} \) \( k = (3k_1 + 4k_2)_{6}, \begin{cases} 0 \leq k_1 \leq 1, \\ 0 \leq k_2 \leq 2. \end{cases} \)

The corresponding index mappings are indicated below:

\[
\begin{array}{cccc|cccc}
\hline
n_2 & 0 & 1 & & k_1 & 0 & 1 & \\
\hline
0 & x[0] & x[3] & & 0 & G[0,0] & G[1,0] & \\
\hline
\end{array}
\]

\[
\begin{array}{cccc|cccc}
\hline
k_2 & 0 & 1 & & k_1 & 0 & 1 & \\
\hline
\hline
\end{array}
\]

Not for sale
(b) $N = 10$. Choose $N_1 = 2$ and $N_2 = 5$. $A = 5$, $B = 2$, $C = 5(5^{-1})_2 = 5$, $D = 2(2^{-1})_5 = 6$.

Thus, $n = \langle 5n_1 + 2n_2 \rangle_{10}$, $0 \leq n_1 \leq 4$, $0 \leq n_2 \leq 2$, $k = \langle 5k_1 + 6k_2 \rangle_{10}$, $0 \leq k_1 \leq 4$, $0 \leq k_2 \leq 2$.

The corresponding index mappings are indicated below:

\[
\begin{array}{c|cc}
  n_1 & 0 & 1 & 2 \\
  \hline
\end{array}
\quad
\begin{array}{c|cc}
  k_1 & 0 & 1 & 2 \\
  \hline
  0 & G[0,0] & G[1,0] & G[2,0] \\
  1 & G[0,1] & G[1,1] & G[2,1] \\
  2 & G[0,2] & G[1,2] & G[2,2] \\
  3 & G[0,3] & G[1,3] & G[2,3] \\
\end{array}
\quad
\begin{array}{c|cc}
  k_1 & 0 & 1 & 2 \\
  \hline
\end{array}
\]

(c) $N = 12$. Choose $N_1 = 3$ and $N_2 = 4$. $A = 4$, $B = 3$, $C = 4(4^{-1})_3 = 4$, $D = 3(3^{-1})_4 = 3$.

Thus, $n = \langle 4n_1 + 3n_2 \rangle_{12}$, $0 \leq n_1 \leq 2$, $0 \leq n_2 \leq 3$, $k = \langle 4k_1 + 3k_2 \rangle_{12}$, $0 \leq k_1 \leq 2$, $0 \leq k_2 \leq 3$.

The corresponding index mappings are indicated below:

\[
\begin{array}{c|cc|c}
  n_1 & 0 & 1 & 2 \\
  \hline
\end{array}
\quad
\begin{array}{c|cc|c}
  k_1 & 0 & 1 & 2 \\
  \hline
  0 & G[0,0] & G[1,0] & G[2,0] \\
  1 & G[0,1] & G[1,1] & G[2,1] \\
  2 & G[0,2] & G[1,2] & G[2,2] \\
  3 & G[0,3] & G[1,3] & G[2,3] \\
\end{array}
\quad
\begin{array}{c|cc|c}
  k_1 & 0 & 1 & 2 \\
  \hline
\end{array}
\]

Not for sale
(d) \( N = 15 \). Choose \( N_1 = 5 \) and \( N_2 = 3 \). \( A = 3, B = 5, C = 3(3^{-1})_5 = 10, \)
\( D = 5(5^{-1})_3 = 6 \). Thus, \( n = \langle 5n_1 + 3n_2 \rangle_{10}, \) \( 0 \leq n_1 \leq 4, \) \( 0 \leq n_2 \leq 2, \)
\( k = \langle 10k_1 + 6k_2 \rangle_{10}, \) \( 0 \leq k_1 \leq 4, \) \( 0 \leq k_2 \leq 2. \)

The corresponding index mappings are indicated below:

\[
\begin{array}{cccccc}
\hline
n_2 & 0 & 1 & 2 & 3 & 4 \\
\hline
\hline
\end{array}
\]

\[
\begin{array}{cccccc}
\hline
k_2 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & G[0,0] & G[1,0] & G[2,0] & G[3,0] \\
1 & G[0,1] & G[1,1] & G[2,1] & G[3,1] \\
2 & G[0,2] & G[1,2] & G[2,2] & G[3,2] \\
\hline
\end{array}
\]

\[
\begin{array}{cccccc}
\hline
k_1 & 0 & 1 & 2 & 3 & 4 \\
\hline
\hline
\end{array}
\]
11.39 Note that $1536 = 256 \times 6$. Now an $N$-point DFT, with $N$ divisible by 6, can be computed as follows: 

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} = X_0[\langle k \rangle_{N/6}] + W_N^k X_1[\langle k \rangle_{N/6}] + W_N^{2k} X_2[\langle k \rangle_{N/6}] + W_N^{3k} X_3[\langle k \rangle_{N/6}] + W_N^{4k} X_4[\langle k \rangle_{N/6}] + W_N^{5k} X_5[\langle k \rangle_{N/6}],$$

where 

$$X_\ell[\langle k \rangle_{N/6}] = \sum_{r=0}^{N/6-1} x[6r + \ell]W_N^{rk}, \quad 0 \leq \ell \leq 5.$$ 

For $N = 1536$, we thus get 

$$X[k] = X_0[\langle k \rangle_{256}] + W_{1536}^k X_1[\langle k \rangle_{256}] + W_{1536}^{2k} X_2[\langle k \rangle_{256}] + W_{1536}^{3k} X_3[\langle k \rangle_{256}] + W_{1536}^{4k} X_4[\langle k \rangle_{256}] + W_{1536}^{5k} X_5[\langle k \rangle_{256}],$$

where $X_\ell[\langle k \rangle_{256}] = \sum_{r=0}^{511} x[6r + \ell]W_{256}^{rk}, \quad 0 \leq \ell \leq 5.$
Now an $N$-point FFT algorithm requires $\frac{N}{2} \log_2 N$ complex multiplications and $N \log_2 N$ complex additions. Hence, an $\frac{N}{6}$-point FFT algorithm requires $\frac{N}{12} \log_2 (N/6)$ complex multiplications and $\frac{N}{6} \log_2 (N/6)$ complex additions. In addition, we need $5N$ complex multiplications and $5N$ complex additions to compute the $N$-point DFT $X[k]$. Hence, for $N = 1536$, the evaluation of $X[k]$ using six 256-point FFT modules requires $\frac{N}{6} \log_2 (N/6) + 5N = 128 \times \log_2 (256) + 5 \times 1536 = 8704$ complex multiplications and $\frac{N}{6} \log_2 (N/6) + 5N = 256 \times \log_2 (256) + 5 \times 1536 = 9728$ complex additions. It should be noted that a direct computation of the 3072-point DFT would require $6431296$ complex multiplications and $2357760$ complex additions.

11.40 (a) # of zero-valued samples to be added is $512 - 498 = 14$.

(b) Direct computation of a 512-point DFT of a length-498 sequence requires $(498)^2 = 248004$ complex multiplications and $497 \times 498 = 247506$ complex additions.

(c) A 512-point Cooley-Tukey type FFT algorithm requires $2304 \times \log_2 512 = 2304$ complex multiplications and $46308 \times \log_2 512 = 46308$ complex additions.

11.41 $z_\ell = \alpha^\ell$. Hence, $A_o V_0^{-\ell} e^{j\theta_o} e^{-j\phi_o} = \alpha^\ell$. Since $\alpha$ is real, we have $A_o = 1, V_o = 1/\alpha, \theta_o = 0, \phi_o = 0$.

11.42 (a) $Y(z) = H(z)X(z)$ or $y[0] + y[1]z^{-1} + y[2]z^{-2} = (h[0] + h[1]z^{-1})(x[0] + x[1]z^{-1})$. Now, $Y(z_0) = Y(-1) = y[0] - y[1] + y[2] = H(-1)X(-1) = (h[0] - h[1])(x[0] - x[1]), Y(z_1) = Y(\infty) = y[0] = H(\infty)X(\infty) = h[0]x[0], Y(z_2) = Y(1) = y[0] + y[1] + y[2] = H(1)X(1) = (h[0] + h[1])(x[0] + x[1]).$

From Eqs. (6.114) and (6.115), we arrive at

$$Y(z) = \frac{I_0(z)}{I_0(z_0)} Y(z_0) + \frac{I_1(z)}{I_1(z_1)} Y(z_1) + \frac{I_2(z)}{I_2(z_2)} Y(z_2),$$

where $I_0(z) = (1 - z_1z^{-1})(1 - z_2z^{-1}), I_1(z) = (1 - z_0z^{-1})(1 - z_2z^{-1}), I_2(z) = (1 - z_0z^{-1})(1 - z_1z^{-1})$. Therefore,

$$\frac{I_0(z)}{I_0(z_0)} = \frac{(1 - z_1z^{-1})(1 - z_2z^{-1})}{(1 - z_1z_0^{-1})(1 - z_2z_0^{-1})} = -\frac{1}{2} z^{-1}(1 - z^{-1}),$$

$$\frac{I_1(z)}{I_1(z_1)} = \frac{(1 - z_0z^{-1})(1 - z_2z^{-1})}{(1 - z_0z_1^{-1})(1 - z_2z_1^{-1})} = (1 - z^{-2}),$$

and

$$\frac{I_2(z)}{I_2(z_0)} = \frac{(1 - z_0z^{-1})(1 - z_1z^{-1})}{(1 - z_0z_2^{-1})(1 - z_1z_2^{-1})} = \frac{1}{2} z^{-1}(1 + z^{-1}).$$

Hence,
\( Y(z) = -\frac{1}{2} z^{-1} (1 - z^{-1}) Y(z_0) + (1 - z^{-2}) Y(z_1) + \frac{1}{2} z^{-1} (1 + z^{-1}) Y(z_2) \)
\[ = Y(z_1) + \left( -\frac{1}{2} Y(z_0) + \frac{1}{2} Y(z_2) \right) z^{-1} + \left( \frac{1}{2} Y(z_0) - Y(z_1) + \frac{1}{2} Y(z_2) \right) z^{-2} \]
\[ = h[0] x[0] + \left( -\frac{1}{2} (h[0] - h[1])(x[0] - x[1]) + \frac{1}{2} (h[0] + h[1])(x[0] + x[1]) \right) z^{-1} \]
\[ + \left( \frac{1}{2} (h[0] - h[1])(x[0] - x[1]) - h[0] x[0] + \frac{1}{2} (h[0] + h[1])(x[0] + x[1]) \right) z^{-2} \]
\[ = h[0] x[0] + (h[0] x[1] + h[1] x[0]) z^{-1} + h[1] x[1] z^{-2}. \]

Ignoring the multiplications by \( \frac{1}{2} \), computation of the coefficients of \( Y(z) \) require the values of \( Y(z_0), Y(z_1), \) and \( Y(z_2) \), which can be evaluated using only 3 multiplications.

(b) \( Y(z) = H(z) X(z) \) or
\[ y[0] + y[1] z^{-1} + y[2] z^{-2} + y[3] z^{-3} + y[4] z^{-4} = \left( h[0] + h[1] z^{-1} + h[2] z^{-2} \right) \left( x[0] + x[1] z^{-1} + x[2] z^{-2} \right) \]

Now,
\[ Y(z_0) = Y\left( -\frac{1}{2} \right) = (h[0] - 2h[1] + 4h[2])(x[0] - 2x[1] + 4x[2]), \]
\[ Y(z_1) = Y(-1) = (h[0] - h[1] + h[2])(x[0] - x[1] + x[2]), \quad Y(z_2) = Y(\infty) = h[0] x[0], \]
\[ Y(z_3) = Y(1) = (h[0] + h[1] + h[2])(x[0] + x[1] + x[2]), \]
\[ Y(z_4) = Y\left( \frac{1}{2} \right) = (h[0] + 2h[1] + 4h[2])(x[0] + 2x[1] + 4x[2]). \]

From Eqs. (6.114) and (6.115), we arrive at
\[ Y(z) = \frac{I_0(z)}{I_0(z_0)} Y(z_0) + \frac{I_1(z)}{I_1(z_1)} Y(z_1) + \frac{I_2(z)}{I_2(z_2)} Y(z_2) + \frac{I_3(z)}{I_3(z_3)} Y(z_3) + \frac{I_4(z)}{I_4(z_4)} Y(z_4), \]
where
\[ I_0(z) = (1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1}), \]
\[ I_1(z) = (1 - z_0 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1}), \]
\[ I_2(z) = (1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1}), \]
\[ I_3(z) = (1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_4 z^{-1}), \]
\[ I_4(z) = (1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1}). \]

Therefore,
\[ \frac{I_0(z)}{I_0(z_0)} = \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1})}{(1 - z_1 z_0^{-1})(1 - z_2 z_0^{-1})(1 - z_3 z_0^{-1})(1 - z_4 z_0^{-1})} = \frac{1}{12} z^{-1}(1 - \frac{1}{2} z^{-1})(1 - z^{-2}), \]
\[ \frac{I_1(z)}{I_1(z_1)} = \frac{(1 - z_0 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1})}{(1 - z_0 z_1^{-1})(1 - z_2 z_1^{-1})(1 - z_3 z_1^{-1})(1 - z_4 z_1^{-1})} = -\frac{2}{3} z^{-1}(1 - z^{-1})(1 - \frac{1}{4} z^{-2}), \]
\[ \frac{I_2(z)}{I_2(z_2)} = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_3 z^{-1})(1 - z_4 z^{-1})}{(1 - z_0 z_2^{-1})(1 - z_1 z_2^{-1})(1 - z_3 z_2^{-1})(1 - z_4 z_2^{-1})} = (1 - z^{-2})(1 - \frac{1}{4} z^{-2}), \]
\[ \frac{I_3(z)}{I_3(z_3)} = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_4 z^{-1})}{(1 - z_0 z_3^{-1})(1 - z_1 z_3^{-1})(1 - z_2 z_3^{-1})(1 - z_4 z_3^{-1})} = \frac{2}{3} z^{-1}(1 + z^{-1})(1 - \frac{1}{4} z^{-2}), \]
\[
\frac{I_4(z)}{I_4(z_4)} = \frac{(1 - z_0 z^{-1})(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})}{(1 - z_0 z_4^{-1})(1 - z_1 z_4^{-1})(1 - z_2 z_4^{-1})(1 - z_3 z_4^{-1})} = -\frac{1}{12} z^{-1} \left( 1 + \frac{1}{2} z^{-1} \right)(1 - z^{-2}).
\]

Hence,
\[
Y(z) = \frac{1}{12} \left( z^{-1} - \frac{1}{2} z^{-2} - z^{-3} + \frac{1}{4} z^{-4} \right) Y(z_0) - \frac{2}{3} \left( z^{-1} - \frac{1}{2} z^{-2} - z^{-3} + \frac{1}{4} z^{-4} \right) Y(z_1)
\]
\[
+ \left( 1 - \frac{5}{4} z^{-2} + \frac{1}{4} z^{-4} \right) Y(z_2) + \frac{2}{3} \left( z^{-1} + z^{-2} - \frac{1}{4} z^{-3} - \frac{1}{4} z^{-4} \right) Y(z_3)
\]
\[
- \frac{1}{12} (z^{-1} + \frac{1}{2} z^{-2} - z^{-3} - \frac{1}{2} z^{-4}) Y(z_4)
\]
\[
= Y(z_2) + \left( \frac{1}{12} Y(z_0) - \frac{2}{3} Y(z_1) + \frac{2}{3} Y(z_3) - \frac{1}{12} Y(z_4) \right) z^{-1}
\]
\[
+ \left( \frac{1}{24} Y(z_0) + \frac{3}{4} Y(z_1) - \frac{5}{4} Y(z_2) + \frac{3}{4} Y(z_3) - \frac{1}{24} Y(z_4) \right) z^{-2}
\]
\[
+ \left( \frac{1}{12} Y(z_0) + \frac{1}{6} Y(z_1) - \frac{1}{6} Y(z_3) + \frac{1}{12} Y(z_4) \right) z^{-3}
\]
\[
+ \left( \frac{1}{24} Y(z_0) - \frac{1}{6} Y(z_1) + \frac{1}{6} Y(z_3) - \frac{1}{24} Y(z_4) \right) z^{-4}.
\]

Substituting the expressions for \( Y(z_0), Y(z_1), Y(z_2), Y(z_3), \) and \( Y(z_4), \) in the above equation, we then arrive at the expressions for the coefficients \{y[n]\} in terms of the coefficients \{h[n]\} and \{x[n]\}. Thus, \( y[0] = Y(z_2) = h[0] x[0], \)
\[
y[1] = \frac{1}{12} Y(z_0) - \frac{2}{3} Y(z_1) + \frac{2}{3} Y(z_3) - \frac{1}{12} Y(z_4) = h[0] x[1] + h[1] x[0],
\]
\[
y[2] = \frac{1}{24} Y(z_0) + \frac{3}{4} Y(z_1) - \frac{5}{4} Y(z_2) + \frac{3}{4} Y(z_3) - \frac{1}{24} Y(z_4) = h[0] x[2] + h[1] x[1] + h[2] x[0],
\]
\[
y[3] = \frac{1}{12} Y(z_0) + \frac{1}{6} Y(z_1) - \frac{1}{6} Y(z_3) + \frac{1}{12} Y(z_4) = h[1] x[2] + h[2] x[1],
\]
\[
y[4] = \frac{1}{24} Y(z_0) - \frac{1}{6} Y(z_1) + \frac{1}{6} Y(z_2) - \frac{1}{24} Y(z_4) = h[2] x[2].
\]

Hence, ignoring the multiplications by \( \frac{1}{12}, \frac{1}{24}, \frac{5}{6}, \frac{1}{4}, \frac{1}{4}, \) and \( \frac{1}{6}, \) computation of the coefficients of \( Y(z) \) require the values of \( Y(z_0), Y(z_1), Y(z_2), Y(z_3), \) and \( Y(z_4), \) which can be evaluated using only 5 multiplications.

11.43 \( Y(z) = H(z) X(z) \) or \( y[0] + y[1] z^{-1} + y[2] z^{-2} = \left( h[0] + h[1] z^{-1} \right) \left( x[0] + x[1] z^{-1} \right) \)
\[
= h[0] x[0] + (h[0] x[1] + h[1] x[0]) z^{-1} + h[2] x[2] z^{-2}. \] Hence, \( y[0] = h[0] x[0], \)
\[
\[
(h[0] + h[1])(x[0] + x[1]) - h[0] x[0] - h[1] x[1] = h[0] x[1] + h[1] x[0] = y[1]. \] As a result, evaluation of \( H(z) X(z) \) requires the computation of 3 products, \( h[0] x[0], h[1] x[1], \) and \( h[0] + h[1])(x[0] + x[1]). \) In addition, it requires 4 additions, \( h[0] + h[1], x[0] + x[1], \) and \( (h[0] + h[1])(x[0] + x[1]) - h[0] x[0] - h[1] x[1]. \)

11.43 Let the two length- \( N \) sequences be denoted by \{h[n]\} and \{x[n]\}. Denote the sequence
generated by the linear convolution of \( h[n] \) and \( x[n] \) as \( y[n] \). Let 
\[
H(z) = \sum_{n=0}^{N-1} h[n]z^{-n}
\]
and 
\[
X(z) = \sum_{n=0}^{N-1} x[n]z^{-n}.
\]
Rewrite \( H(z) \) and \( X(z) \) in the form 
\[
H(z) = H_0(z) + z^{-N/2} H_1(z),
\]
and 
\[
X(z) = X_0(z) + z^{-N/2} X_1(z),
\]
where 
\[
H_0(z) = \sum_{n=0}^{(N/2)-1} h[n]z^{-n}, \quad X_0(z) = \sum_{n=0}^{(N/2)-1} x[n]z^{-n},
\]
\[
H_1(z) = \sum_{n=0}^{(N/2)-1} (n + \frac{N}{2}) z^{-n}, \quad X_1(z) = \sum_{n=0}^{(N/2)-1} x[n + \frac{N}{2}] z^{-n}.
\]
Therefore, we can write 
\[
Y(z) = \left( H_0(z) + z^{-N/2} H_1(z) \right) \left( X_0(z) + z^{-N/2} X_1(z) \right)
\]
\[
= H_0(z)X_0(z) + z^{-N/2} \left( H_0(z)X_1(z) + H_1(z)X_0(z) \right) + z^{-N} H_1(z) X_1(z)
\]
\[
= Y_0(z) + z^{-N/2} Y_1(z) + z^{-N} Y_2(z),
\]
where 
\[
Y_0(z) = H_0(z)X_0(z), \quad Y_1(z) = H_0(z)X_1(z) + H_1(z)X_0(z), \quad Y_2(z) = H_1(z)X_1(z).
\]
The degree of each polynomial is \( N/2 \), and hence, requires \( \left( \frac{N}{2} \right)^2 \) multiplications each. 

Now, we can write 
\[
Y_1(z) = \left( H_0(z) + H_1(z) \right) \left( X_0(z) + X_1(z) \right) - Y_0(z) - Y_1(z).
\]
As a result, \( Y(z) = H(z)X(z) \) can be computed using \( \left( \frac{N}{2} \right)^2 \) multiplications instead of \( N^2 \) multiplications. If \( N \) is a power-of-2, \( \frac{N}{2} \) is even, and the same procedure can be applied to compute \( Y_0(z), Y_1(z), \) and \( Y_2(z) \), reducing further the number of multiplications. This process can be continued until the sequences being convolved are of length 1 each.

Let \( R(N) \) denote the total number of multiplications required to compute the linear convolution of two \( N \)-length sequences. Then, in the method outlined above, we have 
\[
R(N) = 3 \cdot R(N/2) + R(1).
\]
A solution of this equation is given by 
\[
R(N) = 3 \log_2 N.
\]

11.45 The dynamic range of a signed \( B \)-bit integer \( \eta \) is given by 
\[
-(2^{(B-1)} - 1) \leq \eta < (2^{(B-1)} - 1)
\]
which for \( B = 32 \) is given by 
\[
-2^{31} \leq \eta < 2^{31}.
\]

(a) For \( E = 6 \) and \( M = 25 \), the value of a 32-bit floating-point number is given by 
\[
\eta = (-1)^E 2^{E-31} (M).
\]
Hence, the value of the largest number is \( \approx 2^{32} \), and the value of the smallest number is \( \approx -2^{32} \). The dynamic range is therefore \( \approx 2 \times 2^{32} \).

(b) For \( E = 7 \) and \( M = 24 \), the value of a 32-bit floating-point number is given by 
\[
\eta = (-1)^E 2^{E-63} (M).
\]
Hence, the value of the largest number is \( \approx 2^{64} \), and the value of the...
smallest number is \( \approx -2^{64} \). The dynamic range is therefore \( \approx 2 \times 2^{64} \).

(c) For \( E = 8 \) and \( M = 23 \), the value of a 32-bit floating-point number is given by 
\[ \eta = (-1)^s \times 2^{E-127} (M) \]
Hence, the value of the largest number is \( \approx 2^{128} \), and the value of the smallest number is \( \approx -2^{128} \). The dynamic range is therefore \( \approx 2 \times 2^{128} \).

Hence, the dynamic range in a floating-point representation is much larger than that in a fixed-point representation with the same wordlength.

11.46 A 32-bit floating-point number in the IEEE Format has \( E = 8 \) and \( M = 23 \). Also, the exponent \( E \) is coded in a biased form as \( E - 127 \) with certain conventions for special cases such as \( E = 0, 255, \) and \( M = 0 \) (See page 637 of text).

Now, a positive 32-bit floating-point number represented in the “normalized” form have an exponent in the range \( 0 < E < 255 \), and is of the form 
\[ \eta = (-1)^s \times 2^{E-127} (1 \Delta M) \]
Hence, the smallest positive number that can be represented will have \( E = 1 \), and \( M = 0 \ldots 0 \),
and has therefore a value given by \( 2^{-126} \approx 1.18 \times 10^{-38} \). For the largest positive number, \( E = 254 \), and \( M = 1 \ldots 1 \). Thus, here 
\[ \eta = (-1)^0 \times 2^{127} (1 \Delta 1 \ldots 1) \approx 2^{127} \times 2^{22} \approx 3.4 \times 10^{38} \]

Note: For representing numbers less than \( 2^{-126} \), IEEE format uses the “de-normalized” form where \( E = 0 \), and 
\[ \eta = (-1)^s \times 2^{-126} (0 \Delta M) \]
In this case, the smallest positive number that can be represented is given by 
\[ \eta = (-1)^0 \times 2^{-126} (0 \Delta 0 \ldots 0 1) \approx 2^{-149} \approx 1.4013 \times 10^{-45} \]

11.47 For a two’s-complement binary fraction \( s \Delta a_{-1}a_{-2} \ldots a_{-b} \), the decimal equivalent for 
\( s = 0 \) is simply \( \sum_{i=1}^{b} a_{-i} 2^{-i} \). For \( s = 1 \), the decimal equivalent is given by
\[ -\left( \sum_{i=1}^{b} (1-a_{-i}) 2^{-i} + 2^{-b} \right) = -\sum_{i=1}^{b} 2^{-i} + \sum_{i=1}^{b} a_{-i} 2^{-i} = -(1-2^{-b}) + \sum_{i=1}^{b} a_{-i} 2^{-i} = -1 + \sum_{i=1}^{b} a_{-i} 2^{-i} . \]
Hence, the decimal equivalent of \( s \Delta a_{-1}a_{-2} \ldots a_{-b} \) is given by
\[ -s + \sum_{i=1}^{b} a_{-i} 2^{-i} . \]

11.48 For a ones’-complement binary fraction \( s \Delta a_{-1}a_{-2} \ldots a_{-b} \), the decimal equivalent for
\[ s = 0 \text{ is simply } \sum_{i=1}^{b} a_{-i} 2^{-i}. \] For \( s = 1 \), the decimal equivalent is given by

\[ -\left( \sum_{i=1}^{b} (1 - a_{-i}) 2^{-i} \right) = -\sum_{i=1}^{b} 2^{-i} + \sum_{i=1}^{b} a_{-i} 2^{-i} = -(1 - 2^{-b}) + \sum_{i=1}^{b} a_{-i} 2^{-i} \]

\[ = -1 + \sum_{i=1}^{b} a_{-i} 2^{-i}. \] Hence, the decimal equivalent of \( s \Delta a_{-1} a_{-2} \ldots a_{-b} \) is given by

\[ -s(1 - 2^{-b}) + \sum_{i=1}^{b} a_{-i} 2^{-i}. \]

11.49 \( \eta_1 = 0.78125_{10} = 0_{\Delta} 11001, \eta_2 = 0.68750_{10} = 0_{\Delta} 10110, \eta_3 = 0.53125_{10} = 1_{\Delta} 01111. \)

\( \eta_1 + \eta_2 = 0_{\Delta} 11001 + 0_{\Delta} 10110 = 1_{\Delta} 01111. \)

Dropping the overflow bit and adding \( (\eta_1 + \eta_2) \) to \( \eta_3 \):

\( (\eta_1 + \eta_2) + \eta_3 = 0_{\Delta} 01111 + 1_{\Delta} 01111 = 0_{\Delta} 11110 = 0.9375_{10}, \)

where we have dropped the carry bit in the MSB location. Note that the final sum is correct inspite of the overflow.

11.49 The transformation \( \cos \omega = \alpha + \beta \cos \hat{\omega} \) is equivalent to \( \frac{e^{j\omega} + e^{-j\omega}}{2} = \alpha + \beta \left( \frac{e^{j\hat{\omega}} + e^{-j\hat{\omega}}}{2} \right) \),

which by analytic continuation can be expressed as \( \frac{z + z^{-1}}{2} = \alpha + \beta \left( \frac{\hat{z} + \hat{z}^{-1}}{2} \right) \). Now, \( H(z) \) let be a Type 1 linear-phase FIR transfer function of degree \( 2M \). As indicated in Eq. (8.128),

\[ H(z) \text{ can be expressed as } H(z) = z^{-M} \sum_{n=0}^{M} a[n] \left( \frac{z + z^{-1}}{2} \right)^n \] with a frequency response

given by \( H(e^{j\omega}) = e^{-jM\omega} \sum_{n=0}^{M} a[n](\cos \omega)^n \), with \( \tilde{H}(\omega) = \sum_{n=0}^{M} a[n](\cos \omega)^n \) denoting the amplitude function or the zero-phase frequency response. The amplitude function or the zero-phase frequency response of the transformed filter obtained by applying the mapping \( \cos \omega = \alpha + \beta \cos \hat{\omega} \) is therefore given by \( \tilde{H}(\hat{\omega}) = \sum_{n=0}^{M} a[n](\alpha + \beta \cos \hat{\omega})^n \). Or, equivalently, the transfer function of the transformed filter is given by

\[ H(\hat{z}) = z^{-M} \sum_{n=0}^{M} a[n] \left( \alpha + \beta \frac{\hat{z} + \hat{z}^{-1}}{2} \right)^n \] A convenient way to realize \( H(\hat{z}) \) is to consider the realization of the parent transfer function \( H(z) \) in the form of a Taylor structure as outlined in Problem 8.17 which is obtained by expressing \( H(z) \) in the form

\[ H(z) = \sum_{n=0}^{M} a[n] z^{-M+n} \left( \frac{1+z^{-2}}{2} \right)^n \]. Similarly, the transformed filter can be realized by
replacing each block \( \frac{1+z^{-2}}{2} \) in the Taylor structure by the block \( \alpha \hat{z}^{-1} + \beta \frac{1+z^{-2}}{2} \).

Now, for a lowpass-to-lowpass transformation we can impose the condition 
\[
\tilde{H}(\omega) \bigg|_{\omega=0} = \hat{H}(\omega) \bigg|_{\omega=0}.
\]
This condition is met if \( \alpha + \beta = 1 \) and \( 0 \leq \alpha < 1 \). In this case, the transformation reduces to \( \cos \omega = \alpha + (1-\alpha) \cos \hat{\omega} \). From the plot of the mapping given below, it follows that as \( \alpha \) is varied between 0 and 1, \( \hat{\omega}_c < \omega_c \).

On the other hand if \( \hat{\omega}_c > \omega_c \) is desired along with a lowpass-to-lowpass transformation, we can impose the condition 
\[
\tilde{H}(\hat{\omega}) \bigg|_{\hat{\omega}=\pi} = \hat{H}(\omega) \bigg|_{\omega=\pi}.
\]
This condition is met if \( \beta = 1 + \alpha \) and \( -1 < \alpha \leq 1 \). The corresponding transformation is now given by \( \cos \omega = \alpha + (1+\alpha) \cos \hat{\omega} \). From the plot of the mapping given below, it follows that as \( \alpha \) is varied between 0 and 1, \( \hat{\omega}_c > \omega_c \).

11.51 (a) \( X(z) = \sum_{n=0}^{N-1} x[n] z^{-n} \). \( X(\tilde{z}) = X(z) \bigg|_{z=\tilde{z}} = -\alpha + \tilde{z}^{-1} = \sum_{n=0}^{N-1} x[n] \left( \frac{-\alpha + \tilde{z}^{-1}}{1-\alpha \tilde{z}^{-1}} \right)^n = \frac{P(\tilde{z})}{D(\tilde{z})} \),

where \( P(\tilde{z}) = \sum_{n=0}^{N-1} p[n] \tilde{z}^{-n} = \sum_{n=0}^{N-1} x[n] (1-\alpha \tilde{z}^{-1})^{N-1-n} (-\alpha + \tilde{z}^{-1})^n \), and

\( D(\tilde{z}) = \sum_{n=0}^{N-1} d[n] \tilde{z}^{-n} = (1-\alpha \tilde{z}^{-1})^{N-1} \).
(b) \[ \tilde{X}[k] = X(\tilde{z})\big|_{\tilde{z} = e^{j2\pi k/N}} = \frac{P(\tilde{z})}{D(\tilde{z})}\big|_{\tilde{z} = e^{j2\pi k/N}} = \tilde{P}[k] = P(\tilde{z})\big|_{\tilde{z} = e^{j2\pi k/N}} \]

is the \( N \)–point DFT of the sequence \( p[n] \) and \( D[k] = D(\tilde{z})\big|_{\tilde{z} = e^{j2\pi k/N}} \) is the \( N \)–point DFT of the sequence \( d[n] \).

(c) Let \( P = [p[0] \ p[1] \ \cdots \ p[N-1]]^T \) and \( X = [x[0] \ x[1] \ \cdots \ x[N-1]]^T \). Without any loss of generality, assume \( N = 4 \) in which case \n
\[
P(\tilde{z}) = \sum_{n=0}^{3} p[n]\tilde{z}^{-n} = (x[0] - \alpha x[1] + \alpha^2 x[2] - \alpha^3 x[3]) + \\
+ \left( -3\alpha x[0] + (1 + 2\alpha^2) x[1] - \alpha (2 + \alpha^2) x[2] + 3\alpha^2 x[3] \right) \tilde{z}^{-1} + \\
+ \left( 3\alpha^2 x[0] - \alpha (2 + \alpha^2) x[1] - (1 + \alpha^2) x[2] - 3\alpha x[3] \right) \tilde{z}^{-2} + \\
+ \left( -\alpha^3 x[0] + \alpha^2 x[1] - \alpha x[2] + \alpha x[3] \right) \tilde{z}^{-3}.
\]

Equating like powers of \( \tilde{z}^{-1} \) we can write

\[
P = Q \cdot X \quad \text{or} \quad \begin{bmatrix} p[0] \\ p[1] \\ p[2] \\ p[3] \end{bmatrix} = \begin{bmatrix} 1 & -\alpha & \alpha^2 & -\alpha^3 \\ -3\alpha & 1 + 2\alpha^2 & -\alpha (2 + \alpha^2) & 3\alpha^2 \\ 3\alpha^2 & -\alpha (2 + \alpha^2) & 1 + 2\alpha^2 & -3\alpha \\ -\alpha^3 & \alpha^2 & -\alpha & 1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.
\]

It can be shown that the elements \( q_{r,s}, 0 \leq r, s \leq 3 \), of the \( 4 \times 4 \) matrix \( Q \) can be determined as follows:

(i) The first row is given by \( q_{0,s} = (-\alpha)^s \),

(ii) The first column is given by \( q_{r,0} = 3 C_r (-\alpha)^r = \frac{3!}{r!(3-r)!} (-\alpha)^r \), and

(iii) The remaining elements can be obtained using the recurrence relation \( q_{r,s} = q_{r-1,s-1} - \alpha q_{r,s-1} + \alpha q_{r-1,s} \).

In the general case, we only change the computation of the elements of the first column using the relation \( q_{r,0} = N^{-1} C_r (-\alpha)^r = \frac{(N-1)!}{r!(N-1-r)!} (-\alpha)^r \).

11.52 The highpass transfer function can be expressed as \( H(z) = \frac{1}{2} \left[ A_0(z) - A_1(z) \right] \), where \( A_0(z) = \frac{0.3038 + z^{-1}}{1 + 0.3038 z^{-1}} \) and \( A_1(z) = \frac{0.639 + 0.4012 z^{-1} + z^{-2}}{1 + 0.4012 z^{-1} + 0.639 z^{-2}} \). The tunable highpass transfer function is obtained by applying the lowpass-to-lowpass transformation of Eq. (11.115) where the tuning parameter is given by \( \alpha = \frac{\sin((0.6\pi - \hat{\omega}_p)/2)}{\sin((0.6\pi + \hat{\omega}_p)/2)} \). The tunable highpass transfer function is then given by \( H(z,\alpha) = \frac{1}{2} \left[ \hat{A}_0(z) - \hat{A}_1(z) \right] \) where from Eqn.
(11.119) \[ \hat{A}_0(z) = \frac{[0.3038 - 0.9077\alpha] + z^{-1}}{1 + [0.3038 - 0.9077\alpha]z^{-1}} \] and from Eqn. (11.123)
\[ \hat{A}_1(z) = \frac{[0.639 - 0.1448\alpha] + [0.4012 + 3.439\alpha]z^{-1} + z^{-2}}{1 + [+0.4012 + 3.439\alpha]z^{-1} + [0.639 - 0.1448\alpha]z^{-2}}. \]

11.53 The lowpass transfer function can be expressed as \[ H(z) = \frac{1}{2} \left[ A_0(z) + A_1(z) \right] \] where
\[ A_0(z) = \frac{-0.1584 + z^{-1}}{1 - 0.1584z^{-1}} \] and \[ A_1(z) = \frac{0.3554 - 0.4191z^{-1} + z^{-2}}{1 - 0.4191z^{-1} + 0.3554z^{-2}}. \] The tunable bandpass transfer function is obtained by applying the lowpass-to-bandpass transformation of Eq. (11.124) where the tuning parameter is given by \[ \beta = \cos \omega_o \] with \( \omega_o \) denoting the center frequency of the bandpass filter. The tunable bandpass transfer function is then given by
\[ H(z, \alpha) = \frac{1}{2} \left[ \hat{A}_0(z) + \hat{A}_1(z) \right] \] where
\[ \hat{A}_0(z) = \frac{-0.1584 - z^{-1}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right)}{1 + 0.1584z^{-1}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right)} \] and \[ \hat{A}_1(z) = \frac{0.3554 + 0.4191z^{-1}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right) + z^{-2}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right)^2}{1 + 0.4191z^{-1}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right) + 0.3554z^{-2}\left(\frac{\beta + z^{-1}}{1 + \beta z^{-1}}\right)^2}. \]

M11.1 (a) Numerator coefficients =
\[
[0.0528 \ 0.0797 \ 0.1295 \ 0.1295 \ 0.0797 \ 0.0528]
\]
Denominator coefficients =
\[
[1.0000 \ -1.8107 \ 2.4947 \ -1.8801 \ 0.9537 \ -0.2336]
\]

(b) Numerator coefficients =
\[
[0.0084 \ -0.0335 \ 0.0502 \ -0.0335 \ 0.0084]
\]
Denominator coefficients =

Not for sale
M11.2 The modified MATLAB program is given below:

```matlab
n = 0:60;
w = input('Normalized angular frequency vector = ');  
num = input('Numerator coefficients = ');  
den = input('Denominator coefficients = ');  
x1 = cos(w(1)*pi*n); x2 = cos(w(2)*pi*n);  
x = x1+x2;  
% Generate the output sequence by filtering the input  
y = filter(num,den,x);  
% Plot the input and the output sequences  
figure(1)  
stem(n,x);  
xlabel('Time index n'); ylabel('Amplitude');  
title('Input sequence');  
figure(2)  
stem(n,y);  
xlabel('Time index n'); ylabel('Amplitude');
```

(c) Numerator coefficients = [0.0003 0 -0.0019 0 0.0057 0  
-0.0095 0 0.0095 0 -0.0057 0 0.0019 0 -0.0003]  
Denominator coefficients = [1.0000  1.7451 4.9282 6.1195  
2.2601 0.8470 0.4167 0.0856 0.0299]
The plots generated by the above program for the filter of Example 9.14 for an input composed of a sum of two sinusoidal sequences of angular frequencies, $0.3\pi$ and $0.6\pi$, are given below:

The blocking of the high-frequency signal by the lowpass filter can be demonstrated by replacing the statement `stem(n,x);` with `stem(n,x2);` and the statement `y=filter(num,den,x);` with `y=filter(num,den,x2);`. The plots of the high-frequency input signal and the corresponding output are shown below:

M11.3 The plots generated by the program of Exercise M11.2 for the filter of Example 9.15 for an input composed of a sum of two sinusoidal sequences of angular frequencies, $0.3\pi$ and $0.6\pi$, are given below:
The blocking of the low-frequency signal by the highpass filter can be demonstrated by replacing the statement \texttt{stem(n,x);} with \texttt{stem(n,x1);} and the statement \texttt{y=filter(num,den,x);} with \texttt{y=filter(num,den,x1);}.

The plots of the low-frequency input signal and the corresponding output are shown below:

![Input sequence](image1)

![Output sequence](image2)

\textbf{M11.4} \% The factors for the transfer function of the lowpass filter of Example 9.14 are
\% num1 = [0.0528 0.0528 0];
\% den1 = [1.0000 -0.4909 0];
\% num2 = [1.0000 0.6040 1.0000];
\% den2 = [1.0000 -0.7624 0.5390];
\% num3 = [1.0000 -0.0949 1.0000];
\% den3 = [1.0000 -0.5574 0.8828];

\texttt{N = input('Total number of sections = ')};
\texttt{for k = 1:N;}
\hspace{1em}\texttt{num(k,:) = input('Numerator factor = ')};
\hspace{1em}\texttt{den(k,:) = input('Denominator factor = ')};
\texttt{end}
\texttt{n = 0:60;}
\texttt{w = input('Normalized angular frequency vector = ')};
\texttt{x1 = cos*w(1)*pi*n); x2 = cos*w(2)*pi*n)};
\texttt{x = x1 + x2;}
\% Plot the input sequence
\texttt{figure(1)}
\texttt{stem(n,x);}
\texttt{xlabel('Time index n'); ylabel('Amplitude');}
\texttt{title('Input sequence');}
\texttt{si = [0 0];}
\texttt{for k = 1:N;}
\hspace{1em}\texttt{y(k,:) = filter(num(k,:),den(k,:),x,si);}
\hspace{1em}\texttt{x = y(k,:);}.
\texttt{end}
\% Plot the output sequence
\texttt{figure(2)}
\texttt{stem(n,y(N,:));}
\texttt{xlabel('Time index n'); ylabel('Amplitude');}
\texttt{title('Output sequence');}
The plots generated by the above program for the filter of Example 9.14 for an input composed of a sum of two sinusoidal sequences of angular frequencies, $0.3\pi$ and $0.6\pi$, are given below:

M11.5 % The factors for the transfer function of 
% the highpass filter of Example 9.15 are 
% num1 = [0.0084 -0.0167 0.0084]; 
% den1 = [1.0000 1.3101 0.5151]; 
% num2 = [1.0000 -2.0000 1.0000]; 
% den2 = [1.0000 1.0640 0.7966]; 

The plots generated by the program of Exercise M11.4 for the filter of Example 9.15 for an input composed of a sum of two sinusoidal sequences of angular frequencies, $0.3\pi$ and $0.6\pi$, are given below:

M11.6 To apply the function direct2 to filter a sum of two sinusoidal sequences, we replace the statement 
`y = filter(num,den,x,si);` in the MATLAB program given in the solution of Exercise M11.2 with the statement 
`y = direct2(num,den,x,si);`. The plots generated by running the modified program for the data given in this problem are given below:
The MATLAB program that can be used to compute all DFT samples using the function `goertzel` and the function `fft` is as follows:

```matlab
clear
N = input('Desired DFT length = '); x = input('Input sequence = '); for j = 1:N
    Y(j)=goertzel(x,j);
end
disp('DFT samples computed using goertzel are ');
disp(Y);
disp('DFT samples computed using fft are ');
X = fft(x); disp(X);
```

Results obtained for two input sequences of lengths 8 and 12, respectively, are given below:

Desired DFT length = 8
Input sequence = [1 2 3 4 4 3 2 1]

DFT samples computed using goertzel are
Columns 1 through 4
20.0000  -5.8284-2.4142i  -0.0000-0.0000i  -0.1716-0.4142i
Columns 5 through 8
0         -0.1716+0.4142i  0.0000+0.0000i  -5.8284+2.4142i

DFT samples computed using fft are
Columns 1 through 4
20.0000  -5.8284-2.4142i  0                 -0.1716-0.4142i
Columns 5 through 8
0         -0.1716+0.4142i  0                 -5.828+2.4142i

Desired DFT length = 12
Input sequence = [1 2 4 8 10 12 7 3 -4 5 0 6]

DFT samples computed using goertzel are
Columns 1 through 4
54.0000  -13.0622-21.0885i  1.5000+19.9186i  -4.0000-2.0000i
Columns 5 through 8

Desired DFT length = 12
Input sequence = [1 2 4 8 10 12 7 3 -4 5 0 6]
The MATLAB program that can be used to verify the plots of Figure 11.43 is given below:

```matlab
% Program_11_8.m

[z,p,k] = ellip(5,0.5,40,0.4);
a = conv([1 -p(1)],[1 -p(2)]); b = [1 -p(5)];
c = conv([1 -p(3)],[1 -p(4)]);
w = 0:pi/255:pi;
alpha = 0;
an1 = a(2) + (a(2)*a(2) - 2*(1 + a(3)))*alpha;
an2 = a(3) + (a(3) -1)*a(2)*alpha;
g = b(2) - (1 - b(2)*b(2))*alpha;
cn1 = c(2) + (c(2)*c(2) - 2*(1 + c(3)))*alpha;
cn2 = c(3) + (c(3) -1)*c(2)*alpha;
a = [1 an1 an2]; b = [1 g]; c = [1 cn1 cn2];

h1 = freqz(fliplr(a),a,w); h2 = freqz(fliplr(b),b,w);
h3 = freqz(fliplr(c),c,w);

ha = 0.5*(h1.*h2 + h3); ma = 20*log10(abs(ha));
alpha = 0.1;
an1 = a(2) + (a(2)*a(2) - 2*(1 + a(3)))*alpha;
an2 = a(3) + (a(3) -1)*a(2)*alpha;
g = b(2) - (1 - b(2)*b(2))*alpha;
cn1 = c(2) + (c(2)*c(2) - 2*(1 + c(3)))*alpha;
cn2 = c(3) + (c(3) -1)*c(2)*alpha;
a = [1 an1 an2]; b = [1 g]; c = [1 cn1 cn2];

h1 = freqz(fliplr(a),a,w); h2 = freqz(fliplr(b),b,w);
h3 = freqz(fliplr(c),c,w);

hb = 0.5*(h1.*h2 + h3); mb = 20*log10(abs(hb));
alpha = -0.25;
an1 = a(2) + (a(2)*a(2) - 2*(1 + a(3)))*alpha;
an2 = a(3) + (a(3) -1)*a(2)*alpha;
g = b(2) - (1 - b(2)*b(2))*alpha;
cn1 = c(2) + (c(2)*c(2) - 2*(1 + c(3)))*alpha;
cn2 = c(3) + (c(3) -1)*c(2)*alpha;
```

M11.8
a = [1 an1 an2]; b = [1 g]; c = [1 cn1 cn2];
h1 = freqz(fliplr(a),a,w); h2 = freqz(fliplr(b),b,w);
h3 = freqz(fliplr(c),c,w);
hc = 0.5*(h1.*h2 + h3); mc = 20*log10(abs(hc));
plot(w/pi,ma,'r-',w/pi,mb,'b--',w/pi,mc,'g-.'); axis([0 1 -80 5]);
xlabel('Normalized frequency'); ylabel('Gain, dB');
legend('b--','\alpha = 0.1 ','r-','\alpha = 0 ','g-.','\alpha = -0.25 ');

M11.9 The MATLAB program that can be used to verify the plots of Figure 11.44 is given below:
%Program_11_9.m
w = 0:pi/255:pi;
w = 0:pi/255:pi;
f = [0 0.36 0.46 1]; m = [1 1 0 0];
f = [0 0.36 0.46 1]; m = [1 1 0 0];
b1 = remez(50, f, m);
b1 = remez(50, f, m);
h1 = freqz(b1,1,w);
h1 = freqz(b1,1,w);
m1 = 20*log10(abs(h1));
m1 = 20*log10(abs(h1));
n = -25:-1;
n = -25:-1;
c = b1(1:25)./sin(0.41*pi*n);
c = b1(1:25)./sin(0.41*pi*n);
d = c.*sin(wc2*n); q = (b1(26)*wc2)/(0.4*pi);
d = c.*sin(wc2*n); q = (b1(26)*wc2)/(0.4*pi);
b2 = [d q fliplr(d)];
b2 = [d q fliplr(d)];
h2 = freqz(b2,1,w);
h2 = freqz(b2,1,w);
m2 = 20*log10(abs(h2));
m2 = 20*log10(abs(h2));
w = 0:pi/255:pi;
w = 0:pi/255:pi;
f = [0 0.36 0.46 1]; m = [1 1 0 0];
f = [0 0.36 0.46 1]; m = [1 1 0 0];
b1 = remez(50, f, m);
b1 = remez(50, f, m);
h1 = freqz(b1,1,w);
h1 = freqz(b1,1,w);
m1 = 20*log10(abs(h1));
m1 = 20*log10(abs(h1));
n = -25:-1;
n = -25:-1;
c = b1(1:25)./sin(0.41*pi*n);
c = b1(1:25)./sin(0.41*pi*n);
d = c.*sin(wc3*n); q = (b1(26)*wc3)/(0.4*pi);
d = c.*sin(wc3*n); q = (b1(26)*wc3)/(0.4*pi);
b3 = [d q fliplr(d)];
b3 = [d q fliplr(d)];
h3 = freqz(b3,1,w);
h3 = freqz(b3,1,w);
m3 = 20*log10(abs(h3));
m3 = 20*log10(abs(h3));
plot(w/pi,m1,'r-',w/pi,m2,'b--',w/pi,m3,'g-.');
plot(w/pi,m1,'r-',w/pi,m2,'b--',w/pi,m3,'g-.');
axis([0 1 -80 5]);
axis([0 1 -80 5]);
xlabel('Normalized frequency'); ylabel('Gain, dB');
xlabel('Normalized frequency'); ylabel('Gain, dB');
legend('b--','\omega_c = 0.3\pi ','r-','\omega_c = 0.41\pi ','g-.','\omega_c = 0.51\pi ');
legend('b--','\omega_c = 0.3\pi ','r-','\omega_c = 0.41\pi ','g-.','\omega_c = 0.51\pi ');

M11.10 The MATLAB program to evaluate Eqs. (11.155) is given below:
%Program_11_10.m
x = 0:0.001:0.5;
xs = 3.140625*x + 0.0202636*x.^2 - 5.325196*x.^3 + 0.5446778*x.^4 + 1.800293*x.^5;
xs = 3.140625*x + 0.0202636*x.^2 - 5.325196*x.^3 + 0.5446778*x.^4 + 1.800293*x.^5;
xl = pi*x;
xl = pi*x;
z = sin(xl);
z = sin(xl);
plot(x,y);
plot(x,y);
xlabel('Normalized angle, radians'); ylabel('Amplitude');
xlabel('Normalized angle, radians'); ylabel('Amplitude');
title('Approximate sine values');
title('Approximate sine values');
grid; axis([0 0.5 0 1]);
grid; axis([0 0.5 0 1]);
pause
pause
plot(x,y-z);
plot(x,y-z);
xlabel('Normalized angle, radians'); ylabel('Amplitude');
xlabel('Normalized angle, radians'); ylabel('Amplitude');
title('Error of approximation');
title('Error of approximation');
grid; grid;

The plots generated by the above program are as indicated below:
M11.11 The MATLAB program to evaluate Eqs. (11.156) and (11.135) is given below:

```matlab
%Program_11_11.m
k = 1;
for x = 0:.01:1
    op1 = 0.318253*x+0.00331*x^2-0.130908*x^3+0.068524*x^4-
        0.009159*x^5;
    op2 = 0.999866*x-0.3302995*x^3+0.180141*x^5-
        0.085133*x^7+0.0208351*x^9;
    arctan1(k) = op1*180/pi;
    arctan2(k) = 180*op2/pi;
    actual(k) = atan(x)*180/pi;
    k = k+1;
end
subplot(211)
x = 0:.01:1;
plot(x,arctan2);
ylabel('Angle, degrees');xlabel('Tangent Values');
subplot(212)
plot(x,actual-arctan2,'--');
ylabel('Tangent Values');xlabel('error, radians');
```

The plots generated by the above program are as indicated below: