9.1. There are several possible approaches to this problem. Two are presented below.

Solution #1: Use the program to compute the DFT of $X[k]$, yielding the sequence $g[n]$.

$$g[n] = \sum_{k=0}^{N-1} X[k]e^{-j2\pi kn/N}$$

Then, compute

$$x[n] = \frac{1}{N} g[((N-n))_N]$$

for $n = 0, \ldots, N-1$. We demonstrate that this solution produces the inverse DFT below.

$$\begin{aligned} x[n] &= \frac{1}{N} g[((N-n))_N] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{-j2\pi k(N-n)/N} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N} \end{aligned}$$

Solution #2: Take the complex conjugate of $X[k]$, and then compute its DFT using the program, yielding the sequence $f[n]$.

$$f[n] = \sum_{k=0}^{N-1} X^*[k]e^{-j2\pi kn/N}$$

Then, compute

$$x[n] = \frac{1}{N} f^*[n]$$

We demonstrate that this solution produces the inverse DFT below.

$$\begin{aligned} x[n] &= \frac{1}{N} f^*[n] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N} \end{aligned}$$

9.2. Multiplying out the terms, we find that

$$(A - B)D + (C - D)A = AD - BD + AC - AD = AC - BD = X$$

$$(A - B)D + (C + D)B = AD - BD + BC + BD = AD + BC = Y$$

Thus, the algorithm is verified.

9.3. First, we derive a relationship between the $X_1(e^{j\omega})$ and $X(e^{j\omega})$ using the shift and time reversal properties of the DTFT.

$$\begin{aligned} x_1[n] &= x[32-n] \\ X_1(e^{j\omega}) &= X(e^{-j\omega})e^{-j32\omega} \end{aligned}$$

Looking at the figure we see that calculating $y[32]$ is just an application of the Goertzel algorithm with $k = 7$ and $N = 32$. Therefore,

$$\begin{aligned} y[32] &= X_1[7] \\ &= X_1(e^{j\omega})\big|_{\omega=\frac{2\pi 7}{32}} \\ &= X(e^{-j\omega})e^{-j\omega 32}\big|_{\omega=\frac{7\pi}{16}} \\ &= X(e^{-j\frac{7\pi}{16}})e^{-j(\frac{7\pi}{16})32} \\ &= X(e^{-j\frac{7\pi}{16}}) \end{aligned}$$

Note that if we put $x[n]$ through the system directly, we would be evaluating $X(z)$ at the conjugate location on the unit circle, i.e., at $\omega = +7\pi/16$.

9.4. The figure corresponds to the flow graph of a second-order recursive system implementing Goertzel's algorithm. This system finds $X[k]$ for $k = 7$, which corresponds to a frequency of

$$\omega_k = \frac{14\pi}{32} = \frac{7\pi}{16}$$

9.5 . $X(e^{j6\pi/8})$ corresponds to the $k = 3$ index of a length $N = 8$ DFT. Using the flow graph of the second-order recursive system for Goertzel's algorithm,

$$\begin{aligned} a &= 2\cos\left(\frac{2\pi k}{N}\right) \\ &= 2\cos\left(\frac{2\pi(3)}{8}\right) \\ &= -\sqrt{2} \\ b &= -W_N^k \\ &= -e^{-j6\pi/8} \\ &= \frac{1+j}{\sqrt{2}} \end{aligned}$$

9.6. (a) The "gain" along the emphasized path is $-W_N^2$.

(b) In general, there is only one path between each input sample and each output sample.

(c) $x[0]$ to $X[2]$: The gain is 1.
$x[1]$ to $X[2]$: The gain is W_N^2.
$x[2]$ to $X[2]$: The gain is $-W_N^0 = -1$.
$x[3]$ to $X[2]$: The gain is $-W_N^0 W_N^2 = -W_N^2$.
$x[4]$ to $X[2]$: The gain is $W_N^0 = 1$.
$x[5]$ to $X[2]$: The gain is $W_N^0 W_N^2 = W_N^2$.
$x[6]$ to $X[2]$: The gain is $-W_N^0 W_N^0 = -1$.
$x[7]$ to $X[2]$: The gain is $-W_N^0 W_N^0 W_N^2 = -W_N^2$, as in Part (a).
Now

$$\begin{aligned}
X[2] &= \sum_{n=0}^{7} x[n] W_8^{2n} \\
&= x[0] + x[1]W_8^2 + x[2]W_8^4 + x[3]W_8^6 + x[4]W_8^8 + x[5]W_8^{10} + x[6]W_8^{12} \\
&\quad + x[7]W_8^{14} \\
&= x[0] + x[1]W_8^2 + x[2](-1) + x[3](-W_8^2) + x[4](1) + x[5]W_8^2 \\
&\quad + x[6](-1) + x[7](-W_8^2)
\end{aligned}$$

Each input sample contributes the proper amount to the output DFT sample.

9.7. (a) The input should be placed into A[r] in bit-reversed order.

$$
\begin{aligned}
A[0] &= x[0]\\
A[1] &= x[4]\\
A[2] &= x[2]\\
A[3] &= x[6]\\
A[4] &= x[1]\\
A[5] &= x[5]\\
A[6] &= x[3]\\
A[7] &= x[7]
\end{aligned}
$$

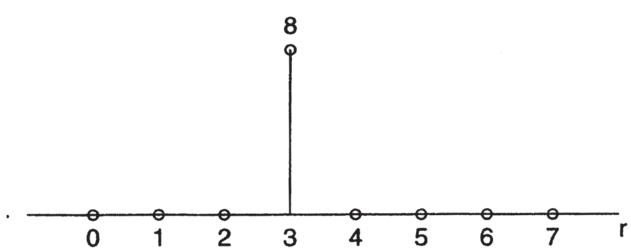
The output should then be extracted from D[r] in sequential order.

$$X[k] = D[k], \quad k = 0, \ldots, 7$$

(b) First, we find the DFT of $(-W_N)^n$ for $N = 8$.

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{7} (-W_8)^n W_8^{nk}\\
&= \sum_{n=0}^{7} (-1)^n W_8^n W_8^{nk}\\
&= \sum_{n=0}^{7} (W_8^{-4})^n W_8^n W_8^{nk}\\
&= \sum_{n=0}^{7} W_8^{n(k-3)}\\
&= \frac{1 - W^{k-3}}{1 - W_8^{k-3}}\\
&= 8\delta[k-3]
\end{aligned}
$$

A sketch of D[r] wis provided below.



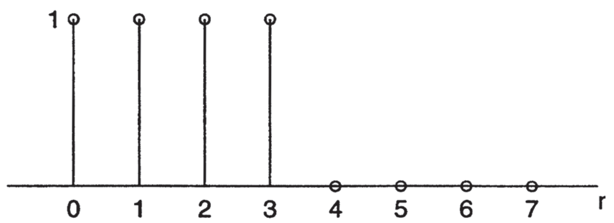
(c) First, the array $D[r]$ is expressed in terms of $C[r]$.

$$
\begin{aligned}
D[0] &= C[0] + C[4]\\
D[1] &= C[1] + C[5]W_8^1\\
D[2] &= C[2] + C[6]W_8^2\\
D[3] &= C[3] + C[7]W_8^3\\
D[4] &= C[0] - C[4]\\
D[5] &= C[1] - C[5]W_8^1\\
D[6] &= C[2] - C[6]W_8^2\\
D[7] &= C[3] - C[7]W_8^3
\end{aligned}
$$

Solving this system of equations for $C[r]$ gives

$$\begin{aligned}
C[0] &= (D[0] + D[4])/2 \\
C[1] &= (D[1] + D[5])/2 \\
C[2] &= (D[2] + D[6])/2 \\
C[3] &= (D[3] + D[7])/2 \\
C[4] &= (D[0] - D[4])/2 \\
C[5] &= (D[1] - D[5])W_8^{-1}/2 \\
C[6] &= (D[2] - D[6])W_8^{-2}/2 \\
C[7] &= (D[3] - D[7])W_8^{-3}/2
\end{aligned}$$

for $r = 0, 1, ..., 7$. A sketch of C[r] is provided below.



9.8. (a) In any stage, $N/2$ butterflies must be computed. In the mth stage, there are 2^{m-1} different coefficients.

(b) Looking at figure 9.10, we notice that the coefficients are

1st stage:	W_8^0
2nd stage:	W_8^0, W_8^2
3rd stage:	$W_8^0, W_8^1, W_8^2, W_8^3$

Here we have listed the *different* coefficients only. The values above correspond to the impulse response

$$h[n] = W_{2^m}^n u[n]$$

which can be generated by the recursion

$$y[n] = W_{2^m} y[n-1] + x[n]$$

Using this recursion, we only generate a sequence of length $L = 2^{m-1}$, which consists of the different coefficients. Then, the remaining $\frac{N}{2} - L$ coefficients are found by repeating these L coefficients.

(c) The difference equation from Part (b) is periodic, since

$$\begin{aligned} h[n] &= W_{2^m}^n u[n] \\ &= e^{-j2\pi n/2^m} u[n] \end{aligned}$$

has a period $R = 2^m$. Thus, the frequency of this oscillator is

$$\omega = \frac{2\pi}{2^m}$$

9.9. *Answer 3*

Decimation in Time:	**The figure is the basic butterfly with $r = 2$.**
Decimation in Frequency:	**The figure is the end of one butterfly and the start of a second with $r = 2$.**

9.10. This is an application of the causal version of the chirp transform with

$$\begin{aligned}
N &= 20 \quad \text{The length of } x[n] \\
M &= 10 \quad \text{The number of desired samples} \\
\omega_0 &= \tfrac{2\pi}{7} \quad \text{The starting frequency} \\
\Delta\omega &= \tfrac{2\pi}{21} \quad \text{The frequency spacing between samples}
\end{aligned}$$

We therefore have

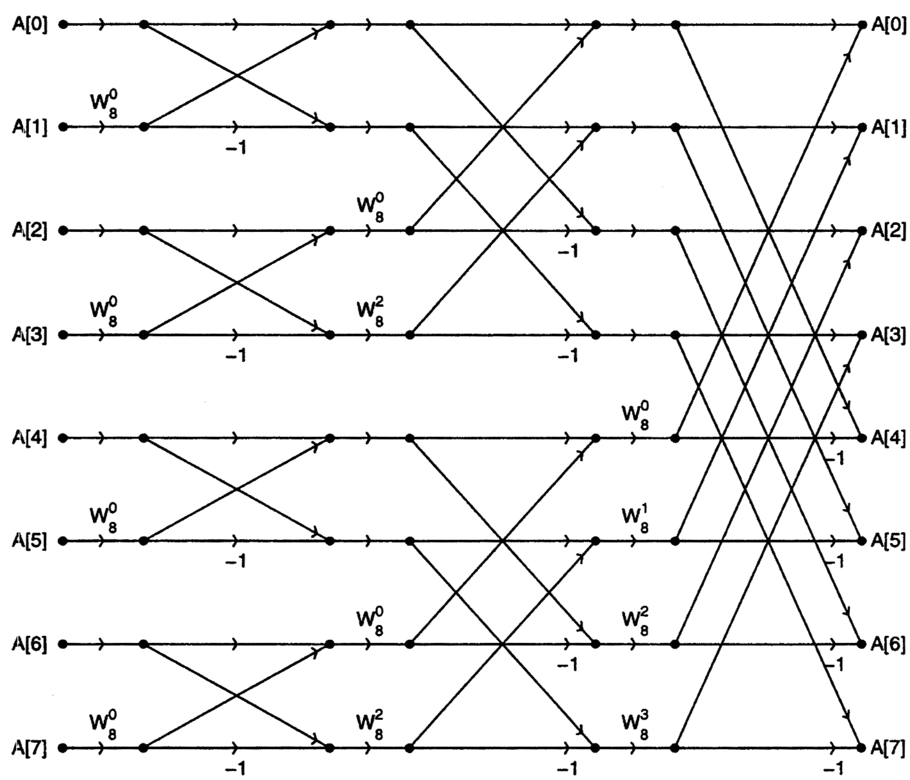
$$y[n+19] = X(e^{j\omega_n}), \quad n = 0, \ldots, 9$$

for $\omega_n = \omega_0 + n\Delta\omega$ or

$$y[n] = X(e^{j\omega_n}), \quad n = 19, \ldots, 28$$

for $\omega_n = \omega_0 + (n-19)\Delta\omega$.

9.11. In this problem, we are using butterfly flow graphs to compute a DFT. These computations are done in place, in an array of registers. An example flow graph for a $N = 8$, (or $v = \log_2 8 = 3$), decimation-in-time DFT is provided below.



(a) The difference between ℓ_1 and ℓ_0 can be found by using the figure above. For example, in the first stage, the array elements $A[4]$ and $A[5]$ comprise a butterfly. Thus, $\ell_1 - \ell_0 = 5 - 4 = 1$. This difference of 1 holds for all the other butterflies in the first stage. Looking at the other stages, we find

$$\text{stage m} = 1: \quad \ell_1 - \ell_0 = 1$$
$$\text{stage m} = 2: \quad \ell_1 - \ell_0 = 2$$
$$\text{stage m} = 3: \quad \ell_1 - \ell_0 = 4$$

From this we find that the difference, in general, is

$$\ell_1 - \ell_0 = 2^{m-1}, \qquad \text{for } m = 1, \ldots, v$$

(b) Again looking at the figure, we notice that for stage 1, there are 4 butterflies with the same twiddle factor. The ℓ_0 for these butterflies are 0, 2, 4, and 6, which we see differ by 2. For stage 2, there are two butterflies with the same twiddle factor. Consider the butterflies with the W_8^0 twiddle factor. The ℓ_0 for these two butterflies are 0 and 4, which differ by 4. Note that in the last stage, there are no butterflies with the same twiddle factor, as the four twiddle factors are unique. Thus, we found

$$\text{stage m} = 1: \quad \Delta\ell_0 = 2$$
$$\text{stage m} = 2: \quad \Delta\ell_0 = 4$$
$$\text{stage m} = 3: \quad \text{n/a}$$

From this, we can generalize the result

$$\Delta\ell_0 = 2^m, \qquad \text{for } m = 1, \ldots, v - 1$$

9.12. This is an application of the causal version of the chirp transform with

$$\begin{aligned} N &= 12 && \text{The length of } x[n] \\ M &= 5 && \text{The number of desired samples} \\ \omega_0 &= \tfrac{2\pi}{19} && \text{The starting frequency} \\ \Delta\omega &= \tfrac{2\pi}{10} && \text{The distance in frequency between samples} \end{aligned}$$

Letting $W = e^{-j\Delta\omega}$ we must have

$$r[n] = e^{-j\omega_0 n} W^{n^2/2} = e^{-j\frac{2\pi}{19}n} e^{-j\frac{2\pi}{10}n^2/2}$$

9.13. Reversing the bits (denoted by $\rightarrow$) gives

$$\begin{array}{ccccccc}
0 & = & 0000 & \rightarrow & 0000 & = & 0 \\
1 & = & 0001 & \rightarrow & 1000 & = & 8 \\
2 & = & 0010 & \rightarrow & 0100 & = & 4 \\
3 & = & 0011 & \rightarrow & 1100 & = & 12 \\
4 & = & 0100 & \rightarrow & 0010 & = & 2 \\
5 & = & 0101 & \rightarrow & 1010 & = & 10 \\
6 & = & 0110 & \rightarrow & 0110 & = & 6 \\
7 & = & 0111 & \rightarrow & 1110 & = & 14 \\
8 & = & 1000 & \rightarrow & 0001 & = & 1 \\
9 & = & 1001 & \rightarrow & 1001 & = & 9 \\
10 & = & 1010 & \rightarrow & 0101 & = & 5 \\
11 & = & 1011 & \rightarrow & 1101 & = & 13 \\
12 & = & 1100 & \rightarrow & 0011 & = & 3 \\
13 & = & 1101 & \rightarrow & 1011 & = & 11 \\
14 & = & 1110 & \rightarrow & 0111 & = & 7 \\
15 & = & 1111 & \rightarrow & 1111 & = & 15
\end{array}$$

The new sample order is 0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15.

9.14. *False.* It is possible by rearranging the order in which the nodes appear in the signal flow graph. However, the computation cannot be carried out in-place.

9.15. Only the $m = 1$ stage will have this form. No other stage of a $N = 16$ radix-2 decimation-in-frequency FFT will have a W_{16} term raised to an odd power.

9.16. The possible values of r for each of the four stages are

$$
\begin{aligned}
m &= 1, \qquad r = 0 \\
m &= 2, \qquad r = 0, 4 \\
m &= 3, \qquad r = 0, 2, 4, 6 \\
m &= 4, \qquad r = 0, 1, 2, 3, 4, 5, 6, 7
\end{aligned}
$$

9.17. Plugging in some values of N for the two programs, we find

N	Program A	Program B
2	4	20
4	16	80
8	64	240
16	256	640
32	1024	1600
64	4096	3840

Thus, we see that a sequence with length $N = 64$ is the shortest sequence for which Program B runs faster than Program A.

9.18. The possible values for r for each of the four stages are

$$\begin{aligned} m &= 1, & r &= 0 \\ m &= 2, & r &= 0, 4 \\ m &= 3, & r &= 0, 2, 4, 6 \\ m &= 4, & r &= 0, 1, 2, 3, 4, 5, 6, 7 \end{aligned}$$

where W_N^r is the twiddle factor for each stage. Since the particular butterfly shown has $r = 2$, the stages which have this butterfly are

$$m = 3, 4$$

9.19. The FFT is a decimation-in-time algorithm, since the decimation-in-frequency algorithm has only W_{32}^0 terms in the last stage.

9.20. If the $N_1 = 1021$ point DFT was calculated using the convolution sum directly it would take N_1^2 multiplications. If the $N_2 = 1024$ point DFT was calculated using the FFT it would take $N_2 \log_2 N_2$ multiplications. Assuming that the number of multiplications is proportional to the calculation time the ratio of the two times is

$$\frac{N_1^2}{N_2 \log_2 N_2} = \frac{1021^2}{1024 \log_2 1024} = 101.8 \approx 100$$

which would explain the results.

9.21. (a) Assume $x[n] = 0$, for $n < 0$ and $n > N-1$. From the figure, we see that

$$y_k[n] = x[n] + W_N^k y_k[n-1]$$

Starting with $n = 0$, and iterating this recursive equation, we find

$$\begin{aligned}
y_k[0] &= x[0] \\
y_k[1] &= x[1] + W_N^k x[0] \\
y_k[2] &= x[2] + W_N^k x[1] + W_N^{2k} x[0] \\
&\vdots \\
y_k[N] &= x[N] + W_N^k x[N-1] + \cdots + W_N^{k(N-1)} x[1] + W_N^{kN} x[0] \\
&= 0 + \sum_{\ell=0}^{N-1} W_N^{k(N-\ell)} x[\ell] \\
&= \sum_{\ell=0}^{N-1} W_N^{-k\ell} x[\ell] \\
&= \sum_{\ell=0}^{N-1} x[\ell] W_N^{(N-k)\ell} \\
&= X[N-k]
\end{aligned}$$

(b) Using the figure, we find the system function $Y_k(z)$.

$$\begin{aligned}
Y_k(z) &= X(z)\frac{1 - W_N^{-k} z^{-1}}{1 - 2z^{-1}\cos(\frac{2\pi k}{N}) + z^{-2}} \\
&= X(z)\frac{1 - W_N^{-k} z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^{k} z^{-1})} \\
&= \frac{X(z)}{1 - W_N^k z^{-1}}
\end{aligned}$$

Therefore, $y_k[n] = x[n] + W_N^k y_k[n-1]$. This is the same difference equation as in part (a).

9.22. To compute the DTFT of $x[n]$ at a particular frequency point we need the impulse response of the LTI filter to be a complex exponential. If $a = W_N^{-k} = e^{j\frac{2\pi k}{N}}$, we can write

$$y[M] = \sum_{n=0}^{M} x[n] W_N^{-k(M-n)}.$$

We need the output sample to be equal to the DTFT of $x[n]$ evaluated at $\omega = 2\pi/60$, i.e.,

$$\sum_{n=0}^{M} x[n] W_N^{nk} W_N^{-Mk} = \sum_{n=0}^{89} x[n] W_{60}^{n}.$$

We can see that we need $M \geq 89$, otherwise samples of $x[n]$ will be disregarded in the computation. If M is chosen to be an integer multiple of N, then $W_N^{-Mk} = 1$ and we eliminate that term. All that remains is to choose k and N such that $\frac{k}{N} = \frac{1}{60}$.

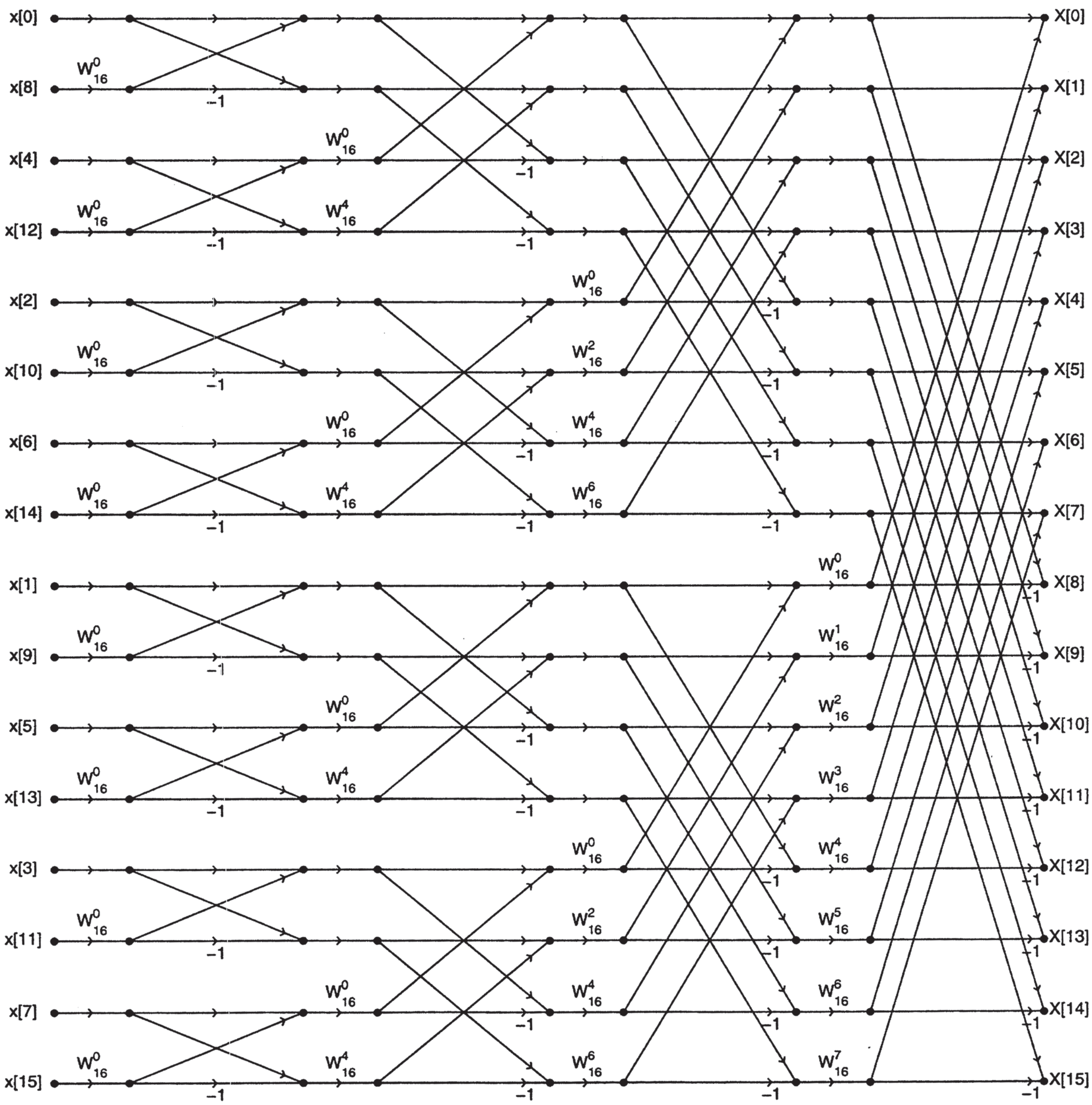
If $k = 1$, $N = 60$, and $M = 120$ we have

$$y[M] = \sum_{n=0}^{120} x[n] W_{60}^{n} W_{60}^{-120} = \sum_{n=0}^{89} x[n] W_{60}^{n},$$

so we can use $a = W_{60}^{-1} = e^{j\frac{2\pi}{60}}$ and $M = 120$.

In fact, any M that is a multiple of 60 and is greater than 89 will work with this choice of a, due to the periodicity of W_{60}^{n}.

9.23. The flow graph for 16 point radix-2 decimation-in-time FFT algorithm is shown below.



To determine the number of real multiplications and additions required to implement the flow graph, consider the number of real multiplications and additions introduced by each of the coefficients W_N^k:

W_{16}^0 :	0 real multiplications + 0 real additions	$(W_{16}^0 = 1)$
W_{16}^4 :	0 real multiplications + 0 real additions	$(W_{16}^4(a+jb) = b - aj)$
W_{16}^2 :	2 real multiplications + 2 real additions	$(W_{16}^2(a+jb) = \frac{\sqrt{2}}{2}(a+b) + j\frac{\sqrt{2}}{2}(b-a))$
W_{16}^6 :	2 real multiplications + 2 real additions	similarly
W_{16}^1 :	4 real multiplications + 2 real additions	
W_{16}^3 :	4 real multiplications + 2 real additions	
W_{16}^5 :	4 real multiplications + 2 real additions	
W_{16}^7 :	4 real multiplications + 2 real additions	

The contribution of all the W_N^k's on the flow graph is 28 real multiplications and 20 real additions. The butterflies contribute 0 real multiplications and 32 real additions per stage. Since there are four stages, the butterflies contribute 0 real multiplications and 128 real additions. In total, 28 real multiplications and 148 real additions are required to implement the flow graph.

9.24. Let $X[k]$ be the DFT of the N-point sequence $x[k]$.

1) Swap the real and imaginary parts of $X[k]$. This gives $Y[k] = jX^*[k]$.
2) Applying the FFT to $Y[k]$ gives

$$\begin{aligned} y[n] &= \sum_{k=0}^{N-1} Y[k] e^{-j2\pi\frac{kn}{N}} \\ &= \sum_{k=0}^{N-1} jX^*[k] e^{-j2\pi\frac{kn}{N}} \\ &= j\sum_{k=0}^{N-1} X^*[k] e^{-j2\pi\frac{kn}{N}}. \end{aligned}$$

3) Swap the real and imaginary part of $y[n]$. This gives $z[n] = jy^*[n]$. That is,

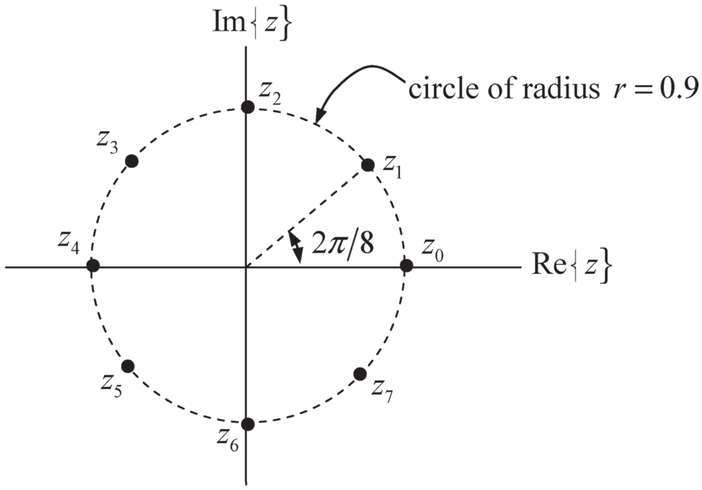
$$\begin{aligned} z[n] &= jy^*[n] \\ &= j\left(j\sum_{k=0}^{N-1} X^*[k] e^{-j2\pi\frac{kn}{N}} \right)^* \\ &= \sum_{k=0}^{N-1} X[k] e^{j2\pi\frac{kn}{N}}. \end{aligned}$$

4) Scale by $\frac{1}{N}$. This gives

$$x[n] = \tfrac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j2\pi\frac{kn}{N}},$$

which is the IDFT of $X[k]$. The procedure works as claimed.

9.25. A.



B.

$$X(z_k) = \sum_{n=0}^{N-1} x[n] z_k^{-n} = \sum_{n=0}^{N-1} x[n] r^{-n} e^{-j\frac{2\pi}{N}kn} = \tilde{X}[k],$$

where $\tilde{x}[n] = x[n] r^{-n}, \quad 0 \le n \le N-1$.

C. 1. For $n = 0$ to $N-1$

$$\tilde{x}[n] = r^{-n} x[n]$$

2. $\tilde{X}[k] = \text{fft}\{\tilde{x}[n]\}$

3. Done.

9.26. Let

$$y[n] = e^{-j2\pi n/627}x[n]$$

Then

$$Y(e^{j\omega}) = X(e^{j(\omega+\frac{2\pi}{627})})$$

Let $y'[n] = \sum_{m=-\infty}^{\infty} y[n+256m], \quad 0 \leq n \leq 255$, and let $Y'[k]$ be the 256 point DFT of $y'[n]$. Then

$$Y'[k] = X\left(e^{j\left(\frac{2\pi k}{256}+\frac{2\pi}{627}\right)}\right)$$

See problem 9.30 for a more in-depth analysis of this technique.

9.27. (a) The problem states that the effective frequency spacing, Δf, should be 50 Hz or less. This constrains N such that

$$\begin{aligned} \Delta f &= \frac{1}{NT} \leq 50 \\ N &\geq \frac{1}{50T} \\ &\geq 200 \end{aligned}$$

Since the sequence length L is 500, and N must be a power of 2, we might conclude that the minimum value for N is 512 for computing the desired samples of the z-transform.

However, we can compute the samples with N equal to 256 by using time aliasing. In this technique, we would zero pad $x[n]$ to a length of 512, then form the 256 point sequence

$$y[n] = \begin{cases} x[n] + x[n+256], & 0 \leq n \leq 255 \\ 0, & \text{otherwise} \end{cases}$$

We could then compute 256 samples of the z-transform of $y[n]$. The effective frequency spacing of these samples would be $1/(NT) \approx 39$ Hz which is lower than the 50 Hz specification.
Note that these samples also correspond to the *even*-indexed samples of a length 512 sampled z-transform of $x[n]$. Problem 9.30 discusses this technique of time aliasing in more detail.

(b) Let

$$y[n] = (1.25)^n x[n]$$

Then, using the modulation property of the z-transform, $Y(z) = X(0.8z)$ and so $Y[k] = X(0.8e^{j2\pi k/N})$.

9.28. A. The DFT of the 4-point sequence $x[0], x[1], x[2], x[3]$ is given by

$$X[k] = X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi k}{4}}, \quad k = 0,1,2,3,$$

where

$$X\left(e^{j\omega}\right) = \sum_{n=0}^{3} x[n] e^{-j\omega n}.$$

The 8-point DFT of the sequence $x[0], x[1], x[2], x[3], 0, 0, 0, 0$ is given by

$$\hat{X}[k] = X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi k}{8}}, \quad k = 0,\ldots,7,$$

where $X\left(e^{j\omega}\right)$ is as above. We see that

$$X[k] = \hat{X}[2k], \quad k = 0,1,2,3.$$

The cost of the system is the cost of the 8-point DFT, which is $1.

9.29. (a) Note that we can write the even-indexed values of $X[k]$ as $X[2k]$ for $k = 0, \ldots, (N/2) - 1$. From the definition of the DFT, we find

$$\begin{aligned} X[2k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi(2k)n/N} \\ &= \sum_{n=0}^{N/2-1} x[n] e^{-j\frac{2\pi}{(N/2)}kn} \\ &\quad + \sum_{n=0}^{N/2-1} x[n+(N/2)] e^{-j\frac{2\pi}{(N/2)}kn} e^{-j\frac{2\pi}{(N/2)}(N/2)k} \\ &= \sum_{n=0}^{N/2-1} (x[n] + x[n+(N/2)]) e^{-j\frac{2\pi}{(N/2)}kn} \\ &= Y[k] \end{aligned}$$

Thus, the algorithm produces the desired results.

(b) Taking the M-point DFT $Y[k]$, we find

$$\begin{aligned} Y[k] &= \sum_{n=0}^{M-1} \sum_{r=-\infty}^{\infty} x[n+rM] e^{-j2\pi kn/M} \\ &= \sum_{r=-\infty}^{\infty} \sum_{n=0}^{M-1} x[n+rM] e^{-j2\pi k(n+rM)/M} e^{j2\pi(rM)k/M} \end{aligned}$$

Let $l = n + rM$. This gives

$$\begin{aligned} Y[k] &= \sum_{l=-\infty}^{\infty} x[l] e^{-j2\pi kl/M} \\ &= X(e^{j2\pi k/M}) \end{aligned}$$

Thus, the result from Part (a) is a special case of this result if we let $M = N/2$. In Part (a), there are only two r terms for which $y[n]$ is nonzero in the range $n = 0, \ldots, (N/2) - 1$.

(c) We can write the odd-indexed values of $X[k]$ as $X[2k+1]$ for $k = 0, \ldots, (N/2) - 1$. From the definition of the DFT, we find

$$\begin{aligned} X[2k+1] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi(2k+1)n/N} \\ &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi n/N} e^{-j2\pi(2k)n/N} \\ &= \sum_{n=0}^{(N/2)-1} x[n] e^{-j2\pi n/N} e^{-j\frac{2\pi}{(N/2)}kn} + \sum_{n=0}^{(N/2)-1} x[n+(N/2)] e^{-j2\pi[n+(N/2)]/N} e^{-j\frac{2\pi}{N/2}k[n+(N/2)]} \\ &= \sum_{n=0}^{(N/2)-1} \left[(x[n] - x[n+(N/2)]) e^{-j\frac{2\pi}{N}n} \right] e^{-j\frac{2\pi}{(N/2)}kn} \end{aligned}$$

Let

$$y[n] = \begin{cases} (x[n] - x[n+(N/2)]) e^{-j(2\pi/N)n}, & 0 \le n \le (N/2) - 1 \\ 0, & \text{otherwise} \end{cases}$$

Then $Y[k] = X[2k+1]$. Thus, The algorithm for computing the odd-indexed DFT values is as follows.

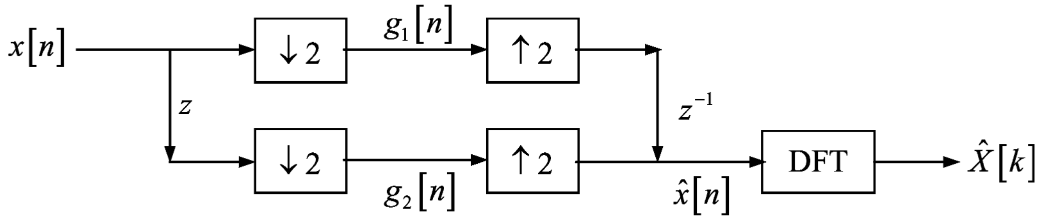
step 1: Form the sequence

$$y[n] = \begin{cases} (x[n] - x[n+(N/2)]) e^{-j(2\pi/N)n}, & 0 \le n \le (N/2) - 1 \\ 0, & \text{otherwise} \end{cases}$$

step 2: Compute the $N/2$ point DFT of $y[n]$, yielding the sequence $Y[k]$.

step 3: The odd-indexed values of $X[k]$ are then $X[k] = Y[(k-1)/2]$, $k = 1, 3, \ldots, N-1$.

9.30. Swapping $g_1[n]$ and $g_2[n]$ corresponds to the following system:



$$G_1(e^{j\omega}) = \tfrac{1}{2}\left[X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right]$$

$$G_2(e^{j\omega}) = \tfrac{1}{2}\left[e^{j\frac{\omega}{2}}X\left(e^{j\frac{\omega}{2}}\right) + e^{j\left(\frac{\omega}{2}-\pi\right)}X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right]$$

$$= \frac{e^{j\frac{\omega}{2}}}{2}\left[X\left(e^{j\frac{\omega}{2}}\right) - X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right].$$

Then

$$\hat{X}(e^{j\omega}) = \frac{e^{-j\omega}}{2}\left[X(e^{j\omega}) + X\left(e^{j(\omega-\pi)}\right)\right] + \frac{e^{j\omega}}{2}\left[X(e^{j\omega}) - X\left(e^{j(\omega-\pi)}\right)\right]$$

$$= \tfrac{1}{2}\left[\left(e^{-j\omega} + e^{j\omega}\right)X(e^{j\omega}) + \left(e^{-j\omega} - e^{j\omega}\right)X\left(e^{j(\omega-\pi)}\right)\right].$$

This gives

$$\hat{X}[k] = \tfrac{1}{2}\left[\left(W_N^k + W_N^{-k}\right)X[k] + \left(W_N^k - W_N^{-k}\right)X\left[((k-N/2))_N\right]\right],$$

where $W_N^k = e^{-j\frac{2\pi k}{N}}$.

9.31. (a) Setting up the butterfly's system of equations in matrix form gives

$$\begin{bmatrix} 1 & 1 \\ W_N^r & -W_N^r \end{bmatrix} \begin{bmatrix} X_{m-1}[p] \\ X_{m-1}[q] \end{bmatrix} = \begin{bmatrix} X_m[p] \\ X_m[q] \end{bmatrix}$$

Solving for

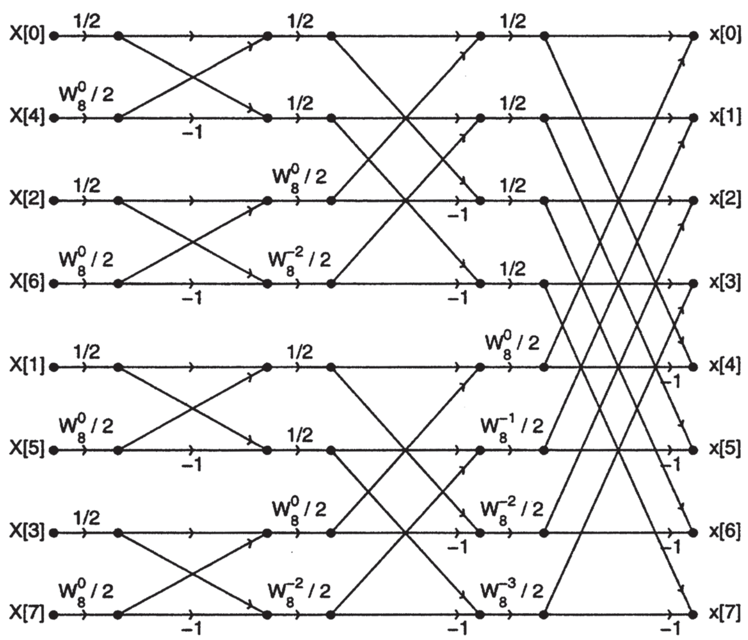
$$\begin{bmatrix} X_{m-1}[p] \\ X_{m-1}[q] \end{bmatrix}$$

gives

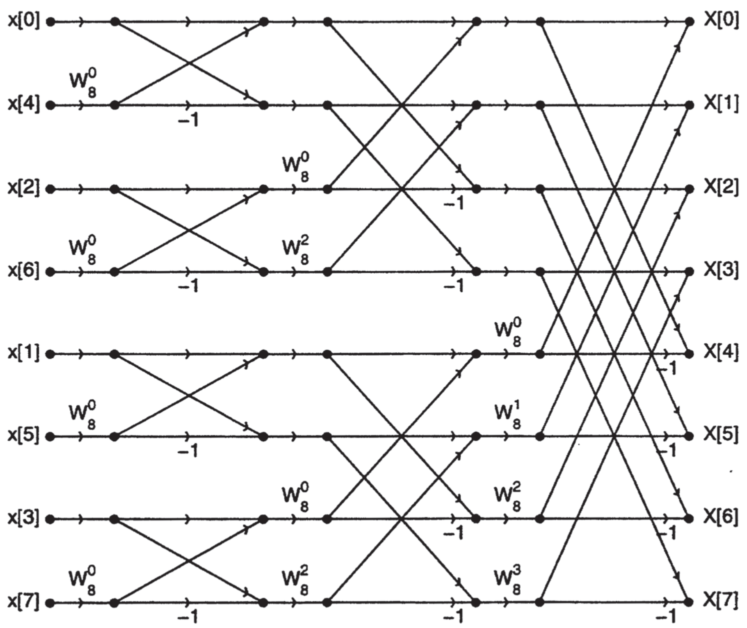
$$\begin{bmatrix} X_{m-1}[p] \\ X_{m-1}[q] \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}W_N^{-r} \\ \frac{1}{2} & -\frac{1}{2}W_N^{-r} \end{bmatrix} \begin{bmatrix} X_m[p] \\ X_m[q] \end{bmatrix}$$

which is consistent with Figure P9.6-2.

(b) The flow graph appears below.



(c) The modification is made by removing all factors of $1/2$, changing all W_N^{-r} to W_N^r, and relabeling the input and the output, as shown in the flow graph below.



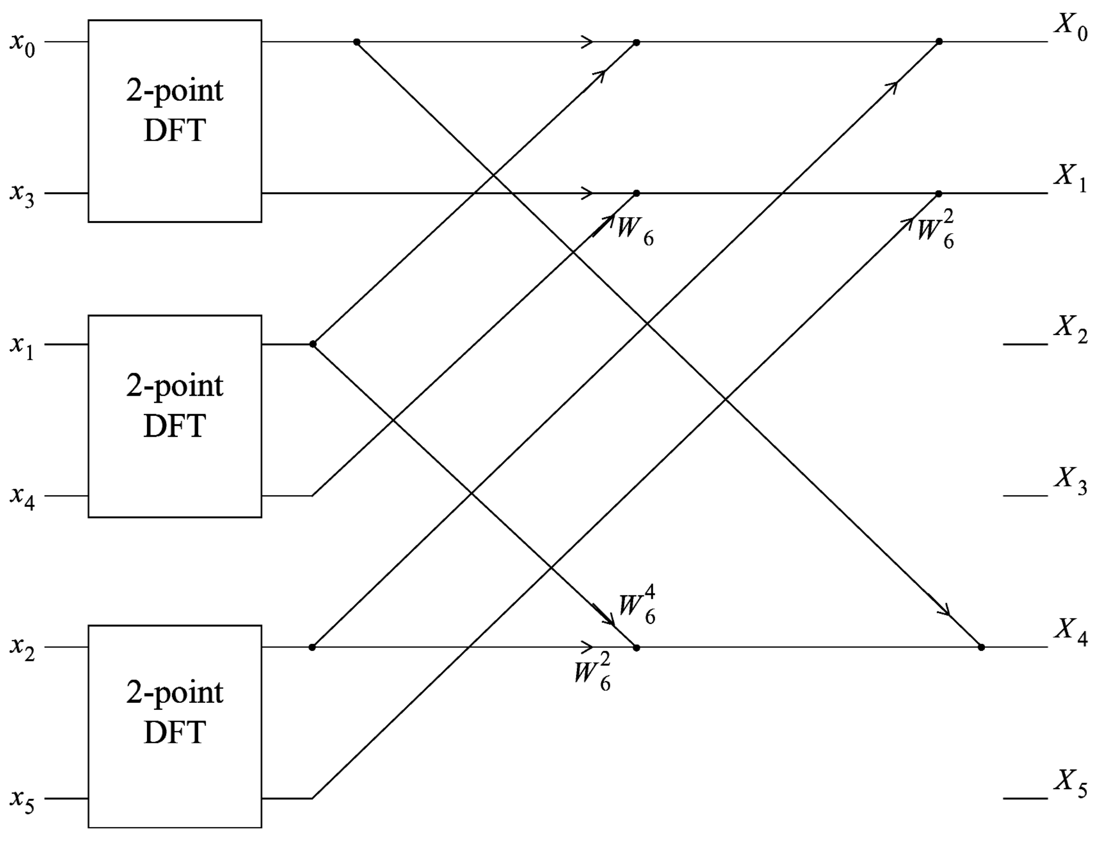
(d) *Yes.* In general, for each decimation-in-time FFT algorithm there exists a decimation-in-frequency FFT algorithm that corresponds to interchanging the input and output and reversing the direction of all the arrows in the flow graph.

9.32. Problem 2 in Spring 2003 Final exam.
Appears in: Spring04 PS8, Fall03 PS8.

Problem

We want to implement a 6-point decimation-in-time FFT using a mixed radix approach. One option is to first take three 2-point DFTs, and then use the results to compute the 6-point DFT. For this option:

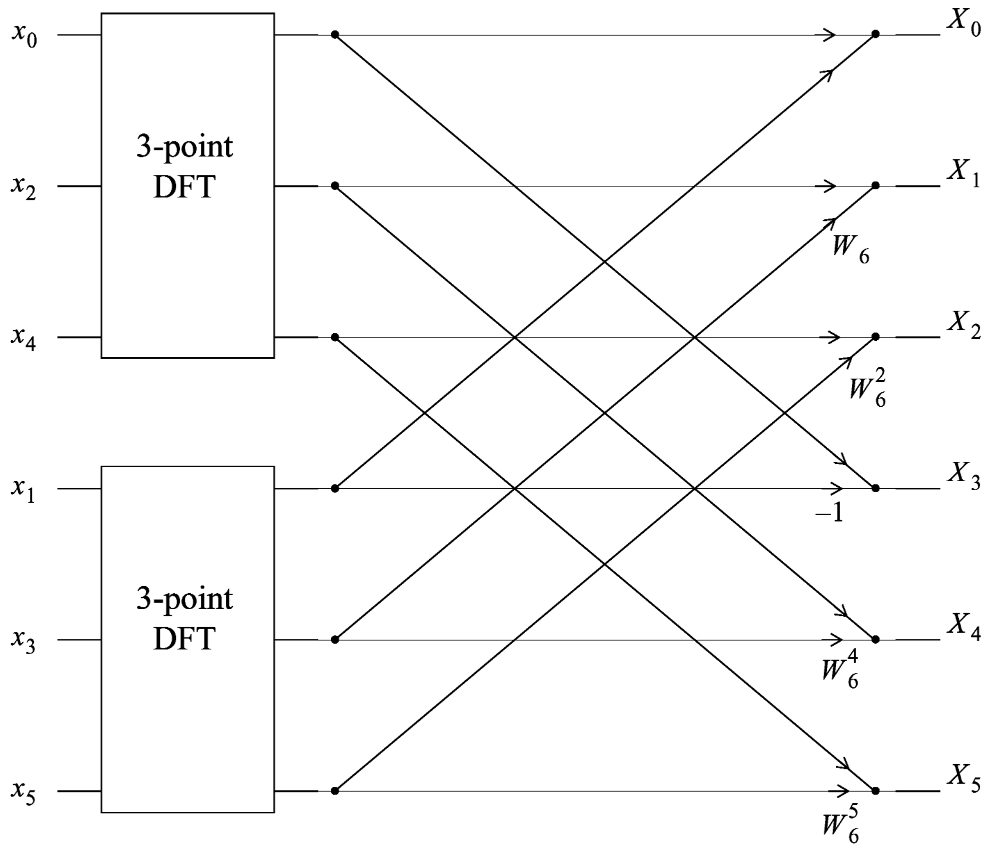
(a) Draw a flowgraph to show what a 2-point DFT calculates. Also, fill in the parts of the flowgraph below involved in calculating the DFT values X_0, X_1, and X_4. Use the definition $W_N = e^{-j2\pi/N}$. Note that due to this function's properties, there is no need to write W_N^p where $p \geq N$ because the function can be rewritten.



(b) How many complex multiplications does this option require? (Multiplying a number by -1 does not count as a complex multiplication.)

A second option is to start with two 3-point DFTs, and then use the results to compute the 6-point DFT.

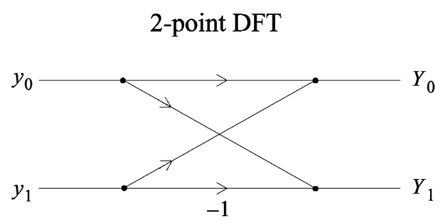
(c) Draw a flowgraph to show what a 3-point DFT calculates. Also, fill in all of the following flowgraph and briefly explain how you derived your implementation:

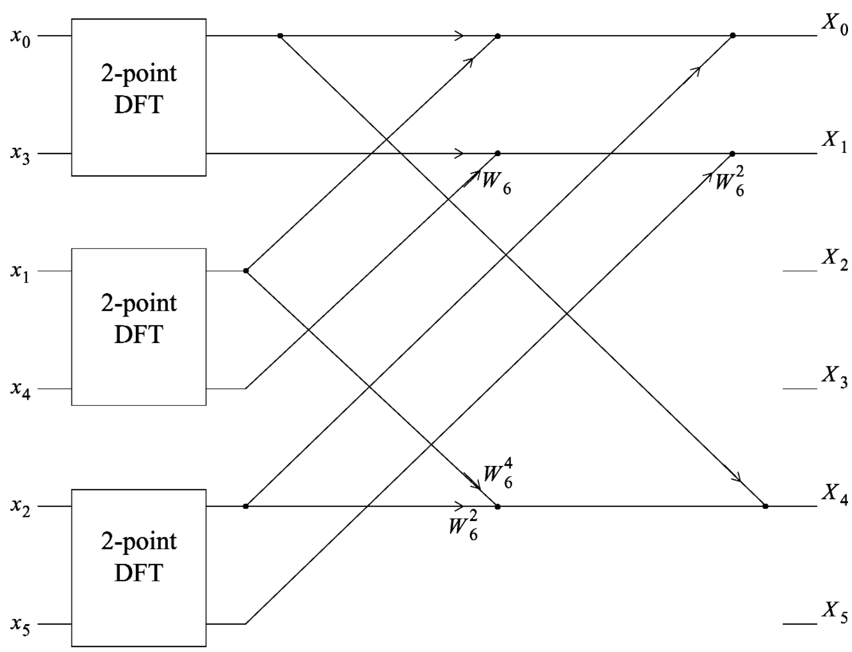


(d) How many complex multiplications does this option require?

Solution from Spring04 PS8

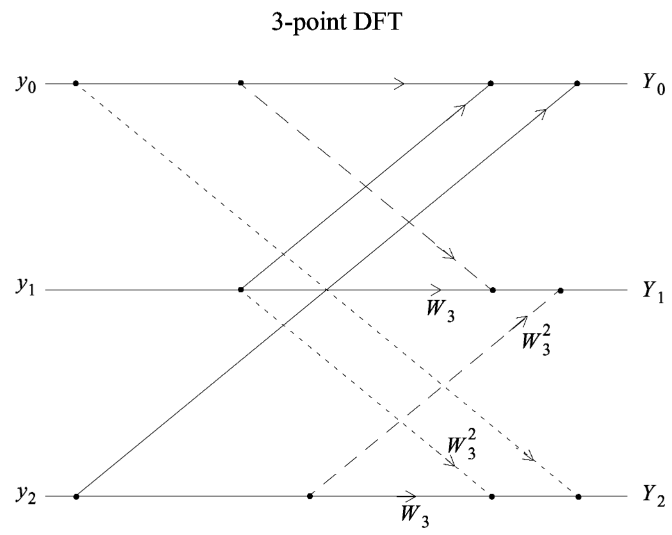
(a)

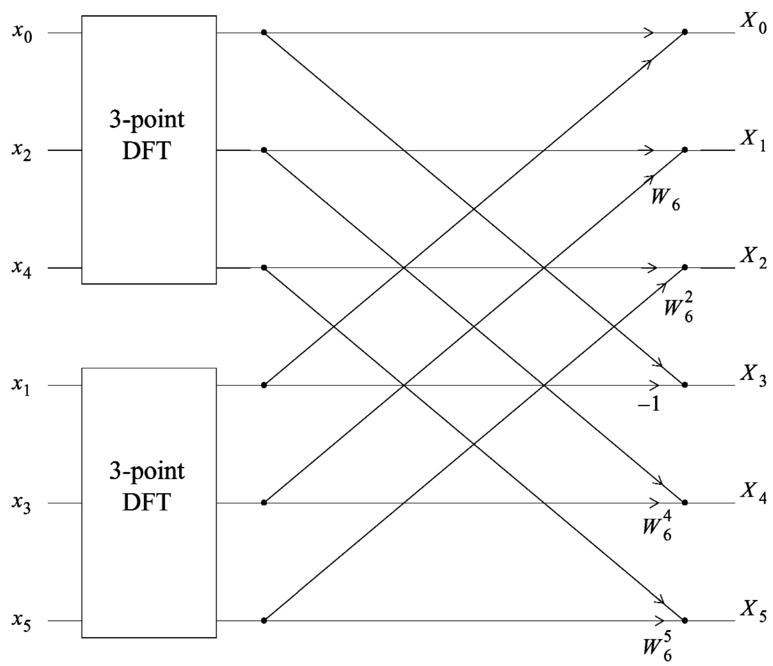




(b) 8 complex multiplications needed

(c)

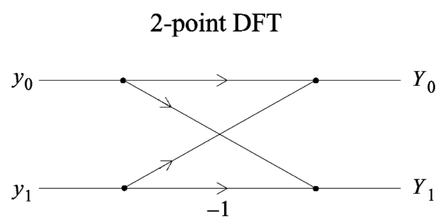


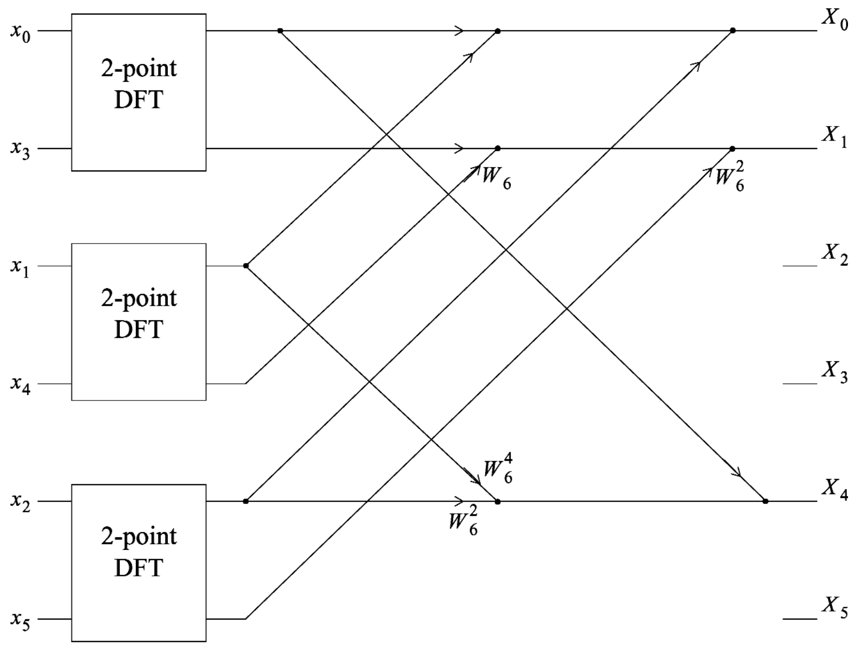


(d) 12 complex multiplications needed

Solution from Fall03 PS8

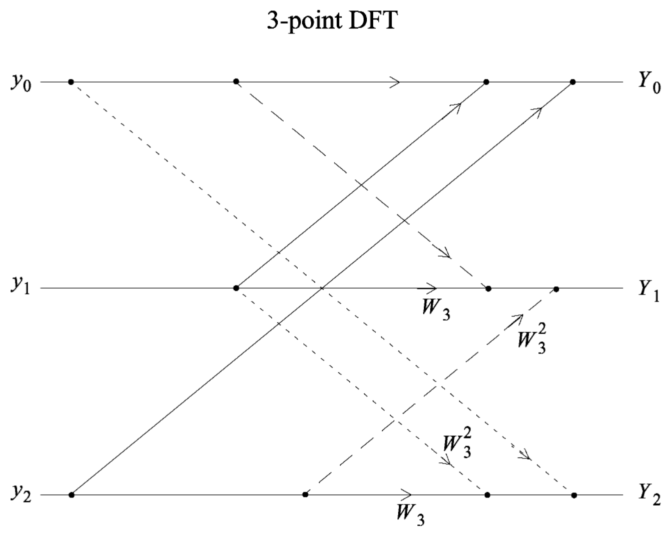
(a)

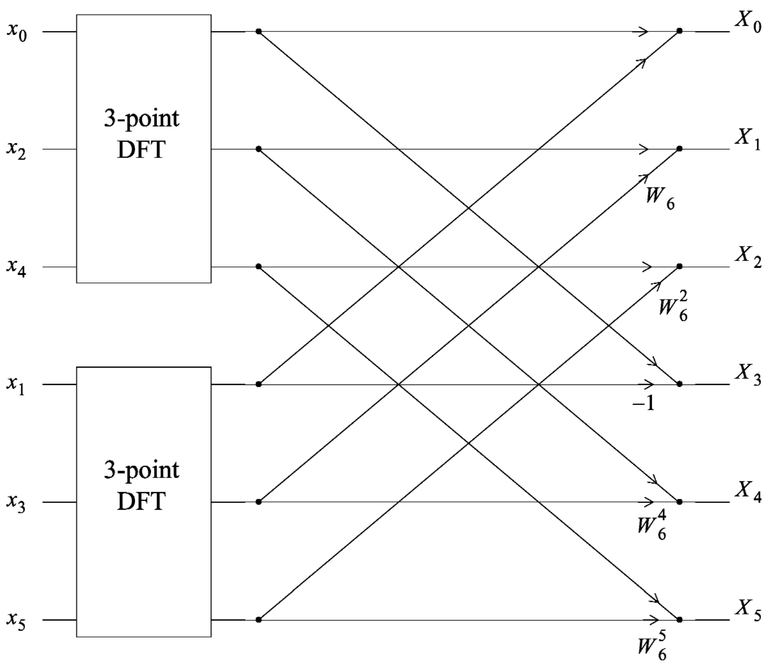




(b) 8 complex multiplications needed

(c)

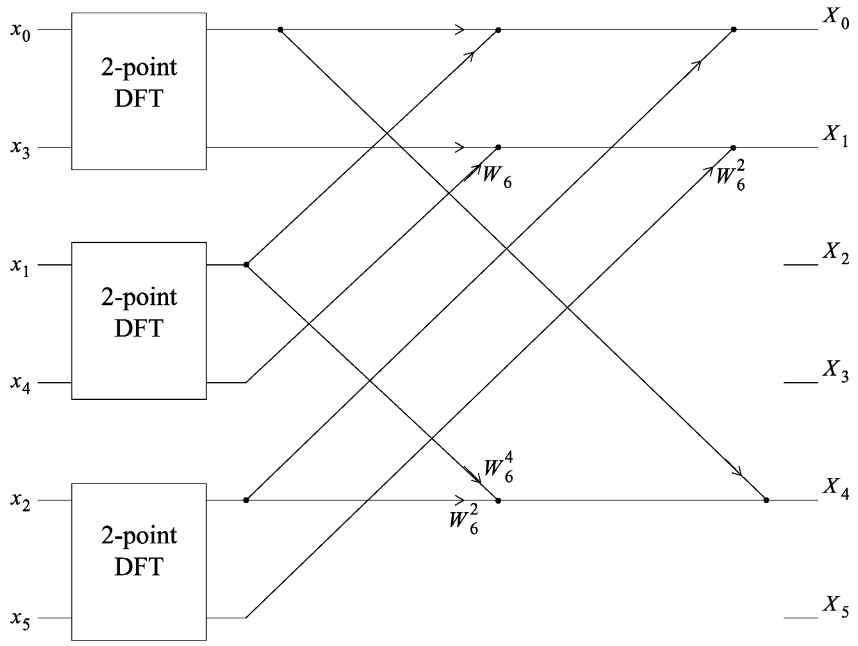


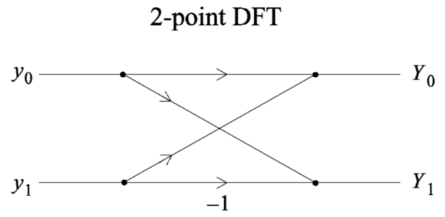


(d) 12 complex multiplications needed

Solution from Spring2003 Final

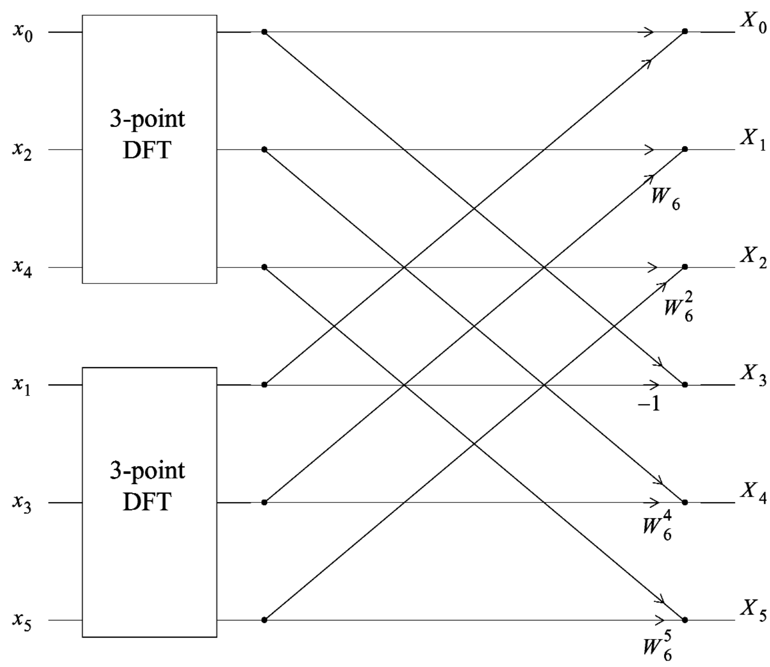
(a) The flowgraphs are:

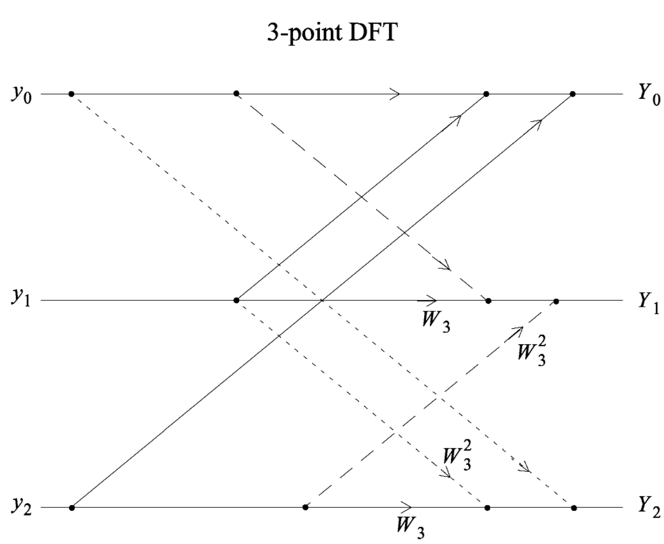




(b) 8 complex multiplications needed

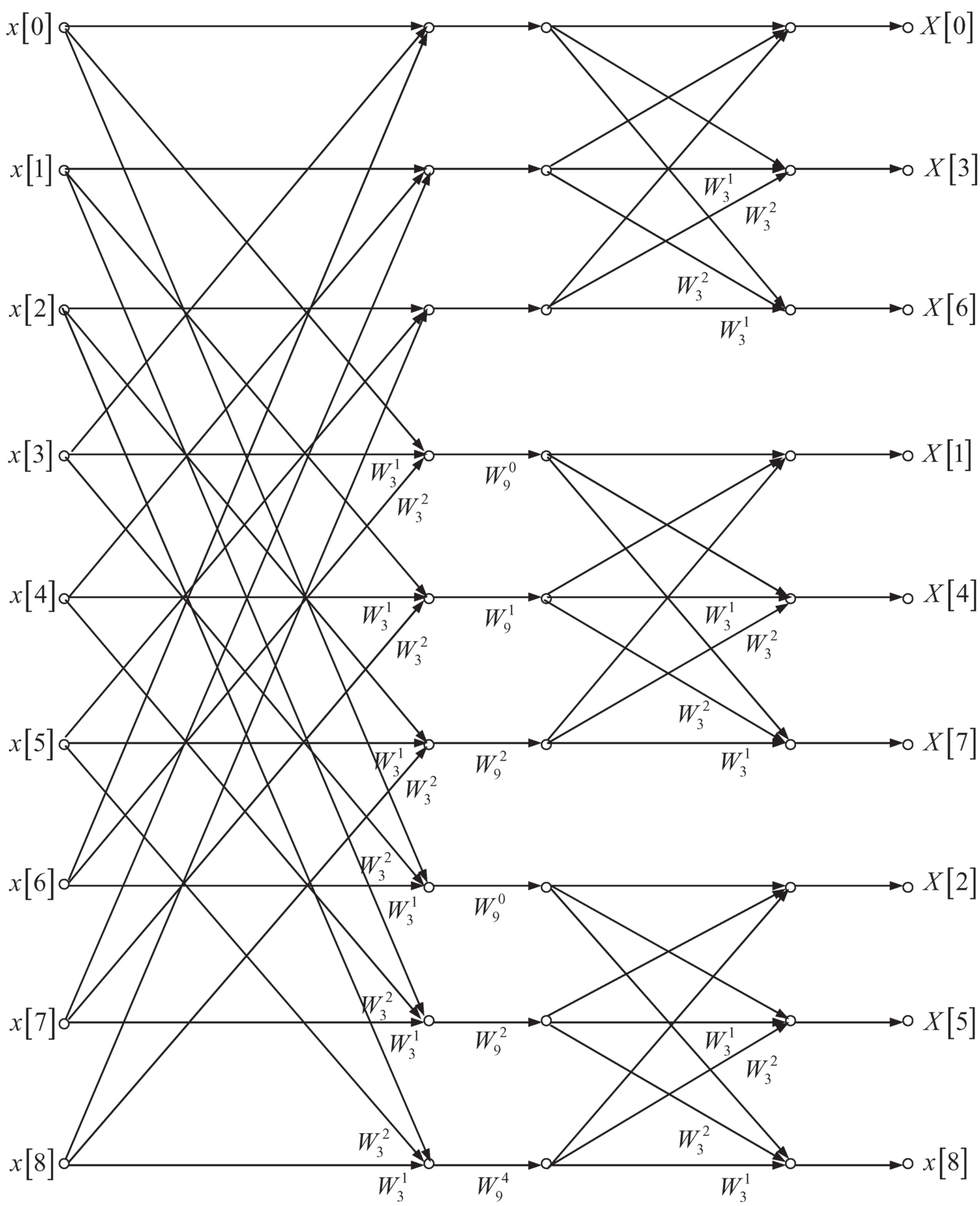
(c) The flowgraphs are:





(d) 12 complex multiplications needed

9.33. 1.



2. For $N = 3^\nu$ approximately $2N\log_3(N)$ complex multiplications are required.

3. In-place computation results only if the arrangement of nodes is such that the input and output nodes for each butterfly computation are horizontally adjacent. That is the case for the diagram of part 1, so in-place computation can be used.

9.34. (a) Using the figure, it is observed that each output $Y[k]$ is a scaled version of $X[k]$. The scaling factor is $W[k]$, which is found to be

$$\begin{array}{lcllllllll} k & = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ W[k] & = & 1 & G & G & G^2 & G & G^2 & G^2 & G^3 \end{array}$$

Using this $W[k]$, $Y[k] = W[k]X[k]$.

(b) $W[k] = G^{p[k]}$, where $p[k]$ = the number of ones in the binary representation of index k.

(c) A procedure for finding $\hat{x}[n]$ is as follows.

step 1: Form $W'[k] = 1/W[k]$.
step 2: Take the inverse DFT of $W'[k]$, yielding $w'[n]$.
step 3: Let $\hat{x}[n]$ be the circular convolution of $x[n]$ and $w'[n]$.

If $\hat{x}[n]$ is input to the modified FFT algorithm, then the output will be $X[k]$, as shown below.

$$\begin{aligned} Y[k] &= W[k]\hat{X}[k] \\ &= W[k]X[k]W'[k] \\ &= X[k] \end{aligned}$$

9.35. Let z_k be the z-plane locations of the 25 points uniformly spaced on an arc of a circle of radius 0.5 from $-\pi/6$ to $2\pi/3$. Then

$$z_k = 0.5e^{j(\omega_0+k\Delta\omega)}, \quad k = 0, 1, \ldots, 24$$

where

$$\begin{aligned} \omega_0 &= -\frac{\pi}{6} \\ \Delta\omega &= \left(\frac{5\pi}{6}\right)\left(\frac{1}{24}\right) \\ &= \frac{5\pi}{144} \end{aligned}$$

From the definition of the z-transform,

$$X(z_k) = \sum_{n=0}^{N-1} x[n]z_k^{-n}$$

Plugging in z_k, and setting $W = e^{-j\Delta\omega}$,

$$X(z_k) = \sum_{n=0}^{N-1} x[n](0.5)^{-n}e^{-j\omega_0 n}W^{nk}$$

This is similar to the expression for $X(e^{jw})$ using the chirp transform algorithm. The only difference is the $(0.5)^{-n}$ term. Setting

$$g[n] = x[n](0.5)^{-n}e^{-j\omega_0 n}W^{n^2/2}$$

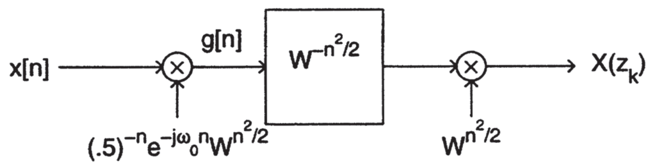
we get

$$X(z_k) = W^{k^2/2}\sum_{n=0}^{N-1} g[n]W^{-(k-n)^2/2}$$

using the result of the chirp transform algorithm. A procedure for computing $X(z)$ at the points z_k is then

- Multiply the sequence x[n] by the sequence $(0.5)^{-n}e^{-j\omega_0 n}W^{n^2/2}$ to form $g[n]$.
- Convolve $g[n]$ with the sequence $W^{-n^2/2}$.
- Multiply this result by the sequence $W^{n^2/2}$ to form $X(z_k)$.

A block diagram of this system appears below.



Solution from Spring05 final

The basic intuition is that since we want the DFT of the even samples and the decimation in time FFT algorithm computes the DFT of the even samples as an intermediate step, we can undo the last stage of a decimation in time FFT algorithm.

Let $N = 1024$. Using the IDFT equation for $x \leftrightarrow X$ and the fact that $x_e[n]$ is a downsampled version of $x[n]$,

$$\begin{aligned}
x_e[n] &= x[2n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-k2n} \\
&= \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_{N/2}^{-kn} \\
&= \frac{1}{N}\sum_{k=0}^{(N/2)-1} \left(X[k] + X[k+N/2]\right) W_{N/2}^{-kn} \quad \text{since } W_{N/2}^{-kn} \text{ is a periodic function of } k \text{ with period } N/2.
\end{aligned}$$

Comparing to the IDFT equation for $x_e \leftrightarrow X_e$

$$x_e[n] = \frac{1}{(N/2)}\sum_{k=0}^{(N/2)-1} X_e[k]W_{N/2}^{-kn},$$

we conclude

$$X_e[k] = \frac{1}{2}\left(X[k] + X[k+(N/2)]\right).$$

Thus an algorithm for finding $X_e[k]$ does not require any DFTs or IDFTs—just the $N/2$ additions and $N/2$ multiplications by $1/2$ above.

Another useful intuition is that since $x_e[n]$ is $x[n]$ downsampled by 2,

$$X_e(e^{j\omega}) = \frac{1}{2}\left(X(e^{j\omega/2}) + X(e^{j(\omega/2+\pi)})\right).$$

Since the DFT is samples of the DTFT, a similar equation holds for the DFT.

Solution from Spring05 final

The basic intuition is that since we want the DFT of the even samples and the decimation in time FFT algorithm computes the DFT of the even samples as an intermediate step, we can undo the last stage of a decimation in time FFT algorithm.

Let $N = 1024$. Using the IDFT equation for $x \leftrightarrow X$ and the fact that $x_e[n]$ is a downsampled version of $x[n]$,

$$\begin{aligned} x_e[n] &= x[2n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-k2n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_{N/2}^{-kn} \\ &= \frac{1}{N}\sum_{k=0}^{(N/2)-1} \left(X[k] + X[k+N/2]\right) W_{N/2}^{-kn} \quad \text{since } W_{N/2}^{-kn} \text{ is a periodic function of } k \text{ with period } N/2. \end{aligned}$$

Comparing to the IDFT equation for $x_e \leftrightarrow X_e$

$$x_e[n] = \frac{1}{(N/2)}\sum_{k=0}^{(N/2)-1} X_e[k]W_{N/2}^{-kn},$$

we conclude

$$X_e[k] = \frac{1}{2}\left(X[k] + X[k+(N/2)]\right).$$

Thus an algorithm for finding $X_e[k]$ does not require any DFTs or IDFTs—just the $N/2$ additions and $N/2$ multiplications by $1/2$ above.

Another useful intuition is that since $x_e[n]$ is $x[n]$ downsampled by 2,

$$X_e(e^{j\omega}) = \frac{1}{2}\left(X(e^{j\omega/2}) + X(e^{j(\omega/2+\pi)})\right).$$

Since the DFT is samples of the DTFT, a similar equation holds for the DFT.

9.37. (a) Since $x[n]$ is real, $x[n] = x^*[n]$, and $X[k]$ is conjugate symmetric.

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x^*[n]e^{-j\frac{2\pi}{N}kn} \\ &= \left(\sum_{n=0}^{N-1} x[n]e^{j\frac{2\pi}{N}kn}e^{-j\frac{2\pi}{N}Nn}\right)^* \\ &= X^*[N-k] \end{aligned}$$

Hence, $X_R[k] = X_R[N-k]$ and $X_I[k] = -X_I[N-k]$.

(b) In Part (a) it was shown that the DFT of a real sequence $x[n]$ consists of a real part that has even symmetry, and an imaginary part that has odd symmetry. We use this fact in the DFT of the sequence $g[n]$ below.

$$\begin{aligned} G[k] &= X_1[k] + jX_2[k] \\ &= (X_{1ER}[k] + jX_{1OI}[k]) + j(X_{2ER}[k] + jX_{2OI}[k]) \\ &= \underbrace{X_{1ER}[k] - X_{2OI}[k]}_{\text{real part}} + j\underbrace{(X_{1OI}[k] + X_{2ER}[k])}_{\text{imaginary part}} \end{aligned}$$

In these expressions, the subscripts "E" and "O" denote even and odd symmetry, respectively, and the subscripts "R" and "I" denote real and imaginary parts, respectively.

Therefore, the even and real part of $G[k]$ is

$$G_{ER}[k] = X_{1ER}[k]$$

the odd and real part of $G[k]$ is

$$G_{OR}[k] = -X_{2OI}[k]$$

the even and imaginary part of $G[k]$ is

$$G_{EI}[k] = X_{2ER}[k]$$

and the odd and imaginary part of $G[k]$ is

$$G_{OI}[k] = X_{1OI}[k]$$

Having established these relationships, it is easy to come up with expressions for $X_1[k]$ and $X_2[k]$.

$$\begin{aligned} X_1[k] &= X_{1ER}[k] + jX_{1OI}[k] \\ &= G_{ER}[k] + jG_{OI}[k] \\ X_2[k] &= X_{2ER}[k] + jX_{2OI}[k] \\ &= G_{EI}[k] - jG_{OR}[k] \end{aligned}$$

(c) An $N = 2^\nu$ point FFT requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions. This is equivalent to $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions.

(i) The two N-point FFTs, $X_1[k]$ and $X_2[k]$, require a total of $4N\log_2 N$ real multiplications and $6N\log_2 N$ real additions.

(ii) Computing the N-point FFT, $G[k]$, requires $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions. Then, the computation of $G_{ER}[k]$, $G_{EI}[k]$, $G_{OI}[k]$, and $G_{OR}[k]$ from $G[k]$ requires approximately $4N$ real multiplications and $4N$ real additions. Then, the formation of $X_1[k]$ and $X_2[k]$ from $G_{ER}[k]$, $G_{EI}[k]$, $G_{OI}[k]$, and $G_{OR}[k]$ requires no real additions or multiplications. So this technique requires a total of approximately $2N\log_2 N + 4N$ real multiplications and $3N\log_2 N + 4N$ real additions.

(d) Starting with

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

and separating $x[n]$ into its even and odd numbered parts, we get

$$X[k] = \sum_{n \text{ even}} x[n]e^{-j2\pi kn/N} + \sum_{n \text{ odd}} x[n]e^{-j2\pi kn/N}$$

Substituting $n = 2\ell$ for n even, and $n = 2\ell + 1$ for n odd, gives

$$\begin{aligned}
X[k] &= \sum_{\ell=0}^{(N/2)-1} x[2\ell]e^{-j2\pi k\ell/(N/2)} + \sum_{\ell=0}^{(N/2)-1} x[2\ell+1]e^{-j2\pi k(2\ell+1)/N} \\
&= \sum_{\ell=0}^{(N/2)-1} x[2\ell]e^{-j2\pi k\ell/(N/2)} + e^{-j2\pi k/N} \sum_{\ell=0}^{(N/2)-1} x[2\ell+1]e^{-j2\pi k\ell/(N/2)} \\
&= \begin{cases} X_1[k] + e^{-j2\pi k/N}X_2[k], & 0 \le k < \frac{N}{2} \\ X_1[k-(N/2)] - e^{-j2\pi k/N}X_2[k-(N/2)], & \frac{N}{2} \le k < N \end{cases}
\end{aligned}$$

(e) The algorithm is then

step 1: Form the sequence $g[n] = x[2n] + jx[2n+1]$, which has length $N/2$.
step 2: Compute $G[k]$, the $N/2$ point DFT of $g[n]$.
step 3: Separate $G[k]$ into the four parts, for $k = 1, \ldots, (N/2) - 1$

$$\begin{aligned}
G_{OR}[k] &= \frac{1}{2}(G_R[k] - G_R[(N/2) - k]) \\
G_{ER}[k] &= \frac{1}{2}(G_R[k] + G_R[(N/2) - k]) \\
G_{OI}[k] &= \frac{1}{2}(G_I[k] - G_I[(N/2) - k]) \\
G_{EI}[k] &= \frac{1}{2}(G_I[k] + G_I[(N/2) - k])
\end{aligned}$$

which each have length $N/2$.

step 4: Form

$$\begin{aligned}
X_1[k] &= G_{ER}[k] + jG_{OI}[k] \\
X_2{}'[k] &= e^{-j2\pi k/N}(G_{EI}[k] - jG_{OR}[k])
\end{aligned}$$

which each have length $N/2$.

step 5: Then, form

$$X[k] = X_1[k] + X_2{}'[k], \qquad 0 \le k < \frac{N}{2}$$

step 6: Finally, form

$$X[k] = X^*[N-k], \qquad \frac{N}{2} \le k < N$$

Adding up the computational requirements for each step of the algorithm gives (approximately)

step 1: 0 real multiplications and 0 real additions.
step 2: $2\frac{N}{2}\log_2\frac{N}{2}$ real multiplications and $3\frac{N}{2}\log_2\frac{N}{2}$ real additions.
step 3: $2N$ real multiplications and $2N$ real additions.
step 4: $2N$ real multiplications and N real additions.
step 5: 0 real multiplications and N real additions.
step 6: 0 real multiplications and 0 real additions.

In total, approximately $N\log_2\frac{N}{2} + 4N$ real multiplications and $\frac{3}{2}N\log_2\frac{N}{2} + 4N$ real additions are required by this technique.

The number of real multiplications and real additions required if $X[k]$ is computed using one N-point FFT computation with the imaginary part set to zero is $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions.

9.38. (a) The length of the sequence is $L + P - 1$.

(b) In evaluating $y[n]$ using the convolution sum, each nonzero value of $h[n]$ is multiplied once with every nonzero value of $x[n]$. This can be seen graphically using the flip and slide view of convolution. The total number of real multiplies is therefore LP.

(c) To compute $y[n] = h[n] * x[n]$ using the DFT, we use the procedure described below.

step 1: Compute N point DFTs of $x[n]$ and $h[n]$.
step 2: Multiply them together to get $Y[k] = H[k]X[k]$.
step 3: Compute the inverse DFT to get $y[n]$.

Since $y[n]$ has length $L + P - 1$, N must be greater than or equal to $L + P - 1$ so the circular convolution implied by step 2 is equivalent to linear convolution.

(d) For these signals, N is large enough so that circular convolution of $x[n]$ and $h[n]$ and the linear convolution of $x[n]$ and $h[n]$ produce the same result. Counting the number of complex multiplications for the procedure in part (b) we get

DFT of $x[n]$	$(N/2)\log_2 N$
DFT of $h[n]$	$(N/2)\log_2 N$
$Y[k] = X[k]H[k]$	N
Inverse DFT of $Y[k]$	$(N/2)\log_2 N$
	$(3N/2)\log_2 N + N$

Since there are 4 real multiplications for every complex multiplication we see that the procedure takes $6N\log_2 N + 4N$ real multiplications. Using the answer from part (a), we see that the direct method requires $(N/2)(N/2) = N^2/4$ real multiplications.

The following table shows that the smallest $N = 2^\nu$ for which the FFT method requires fewer multiplications than the direct method is 256.

N	Direct Method	FFT method
2	1	20
4	4	64
8	16	176
16	64	448
32	256	1088
64	1024	2560
128	4096	5888
256	16384	13312

9.39. (a) For each L point section, $P-1$ samples are discarded, leaving $L-P+1$ output samples. The complex multiplications are:

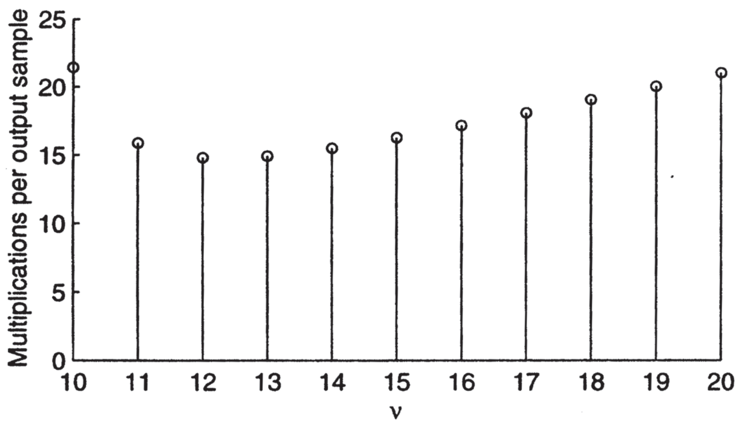
L point FFT of input:	$(L/2)\log_2 L = \nu 2^\nu/2$
Multiplication of filter and section DFT:	$L = 2^\nu$
L point inverse FFT:	$(L/2)\log_2 L = \nu 2^\nu/2$
Total per section:	$2^\nu(\nu+1)$

Therefore,

$$\frac{\text{Complex Multiplications}}{\text{Output Sample}} = \frac{2^\nu(\nu+1)}{2^\nu - P + 1}.$$

Note we assume here that $H[k]$ has been precalculated.

(b) The figure below plots the number of complex multiplications per sample versus ν. For $\nu = 12$, the number of multiplies per sample reaches a minimum of 14.8. In comparison, direct evaluation of the convolution sum would require 500 complex multiplications per output sample.



Although $\nu = 9$ is the first valid choice for overlap-save method, it is not plotted since the value is so large (in the hundreds) it would obscure the graph.

(c)

$$\begin{aligned}\lim_{\nu\to\infty} \frac{2^\nu(\nu+1)}{2^\nu - P + 1} &= \lim_{\nu\to\infty} \frac{\nu+1}{1+\frac{-P+1}{2^\nu}} \\ &= \nu\end{aligned}$$

Thus, for $P = 500$ the direct method will be more efficient for $\nu > 500$.

(d) We want

$$\frac{2^\nu(\nu+1)}{2^\nu - P + 1} \le P.$$

Plugging in $P = L/2 = 2^{\nu-1}$ gives

$$\frac{2^\nu(\nu+1)}{2^\nu - 2^{\nu-1} + 1} \le 2^{\nu-1}.$$

As seen in the table below, the FFT will require fewer complex multiplications than the direct method when $\nu = 5$ or $P = 2^4 = 16$.

	Overlap/Save	Direct
ν	$\frac{2^\nu(\nu+1)}{2^\nu-2^{\nu-1}+1}$	$2^{\nu-1}$
1	2	1
2	4	2
3	6.4	4
4	8.9	8
5	11.3	16

9.40. This problem asks that we find eight equally spaced *inverse* DFT coefficients using the chirp transform algorithm. The book derives the algorithm for the forward DFT. However, with some minor tweaking, it is easy to formulate an inverse DFT. First, we start with the inverse DFT relation

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi nk/N}$$
$$x[n_\ell] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi n_\ell k/N}$$

Next, we define

$$\Delta n = 1$$
$$n_\ell = n_0 + \ell\Delta n$$

where $\ell = 0, \dots, 7$. Substituting this into the equation above gives

$$x[n_\ell] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi n_0 k/N}e^{j2\pi\ell\Delta nk/N}$$

Defining

$$W = e^{-j2\pi\Delta n/N}$$

we find

$$x[n_\ell] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi n_0 k/N}W^{-\ell k}$$

Using the relation

$$\ell k = \frac{1}{2}[\ell^2 + k^2 - (k-\ell)^2]$$

we get

$$x[n_\ell] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi n_0 k/N}W^{-\ell^2/2}W^{-k^2/2}W^{(k-\ell)^2/2}$$

Let

$$G[k] = X[k]e^{j2\pi n_0 k/N}W^{-k^2/2}$$

Then,

$$x[n_\ell] = \frac{1}{N}W^{-\ell^2/2}\left(\sum_{k=0}^{N-1} G[k]W^{(k-\ell)^2/2}\right)$$

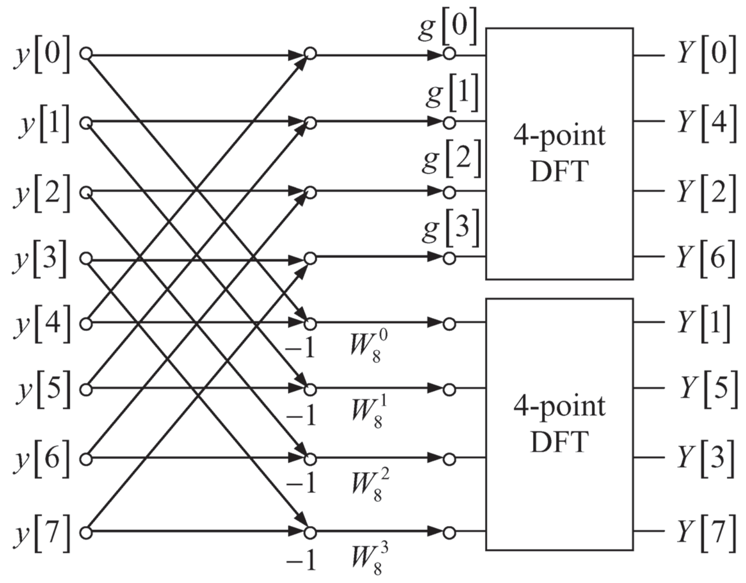
From this equation, it is clear that the inverse DFT can be computed using the chirp transform algorithm. All we need to do is replace n by k, change the sign of each of the exponential terms, and divide by a factor of N. Therefore,

$$m_1[k] = e^{j2\pi kn_0/N}W^{-k^2/2}$$
$$m_2[k] = W^{-k^2/2}$$
$$h[k] = \frac{1}{N}W^{k^2/2}$$

Using this system with $n_0 = 1020$, and $\ell = 0, \dots, 7$ will result in a sequence $y[n]$ which will contain the desired samples, where

$$\begin{aligned} y[0] &= x[1020] \\ y[1] &= x[1021] \\ y[2] &= x[1022] \\ y[3] &= x[1023] \\ y[4] &= x[0] \\ y[5] &= x[1] \\ y[6] &= x[2] \\ y[7] &= x[3] \end{aligned}$$

9.41. B. The figure below shows the calculation of an 8-point DFT using the decimation in frequency technique.

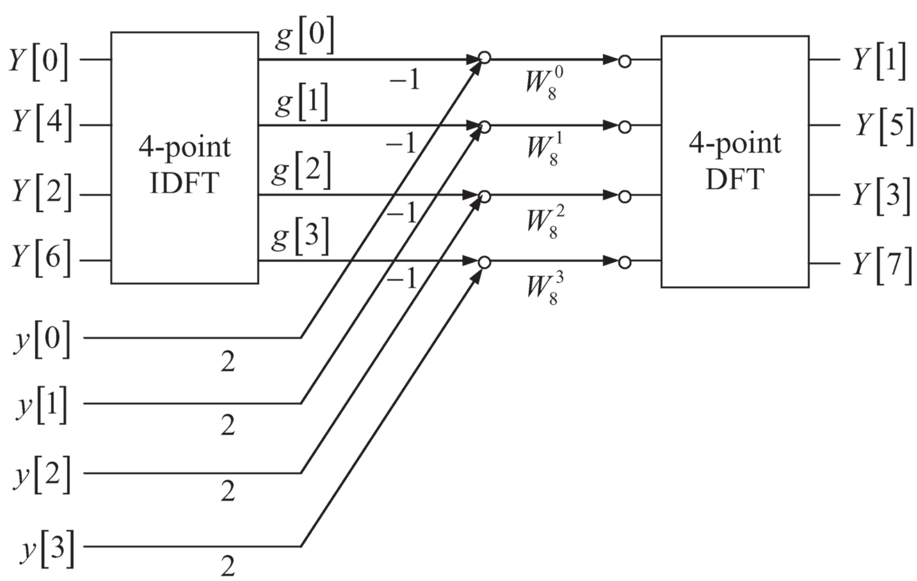


1. If $Y[0]=Y[2]=Y[4]=Y[6]=2$, then a 4-point IDFT gives $g[0]=2, g[1]=g[2]=g[3]=0$. We then have

$$y[4]=g[0]-y[0]=2-1=1$$
$$y[5]=g[1]-y[1]=0$$
$$y[6]=g[2]-y[2]=0$$
$$y[7]=g[3]-y[3]=0.$$

   The 8-point DFT now gives $Y[1]=Y[3]=Y[5]=Y[7]=0$.

2. Now here is my plan:

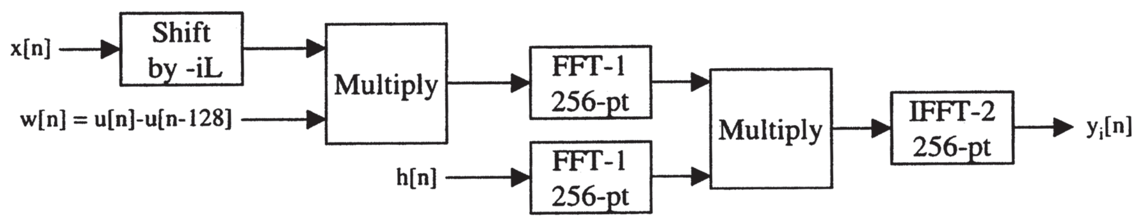


The diagram above is a generalization of the method used in part 1 above. The system shown requires 7 multipliers and 4 subtractors for a total cost of $110.

9.42. First note that

$$
\begin{aligned}
x_i[n] &= \begin{cases} x[n], & iL \leq n \leq iL + 127, \\ 0, & \text{otherwise} \end{cases} \\
&= \begin{cases} x[n+iL], & 0 \leq n \leq 127, \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

Using the above we can implement the system with the following block diagram.

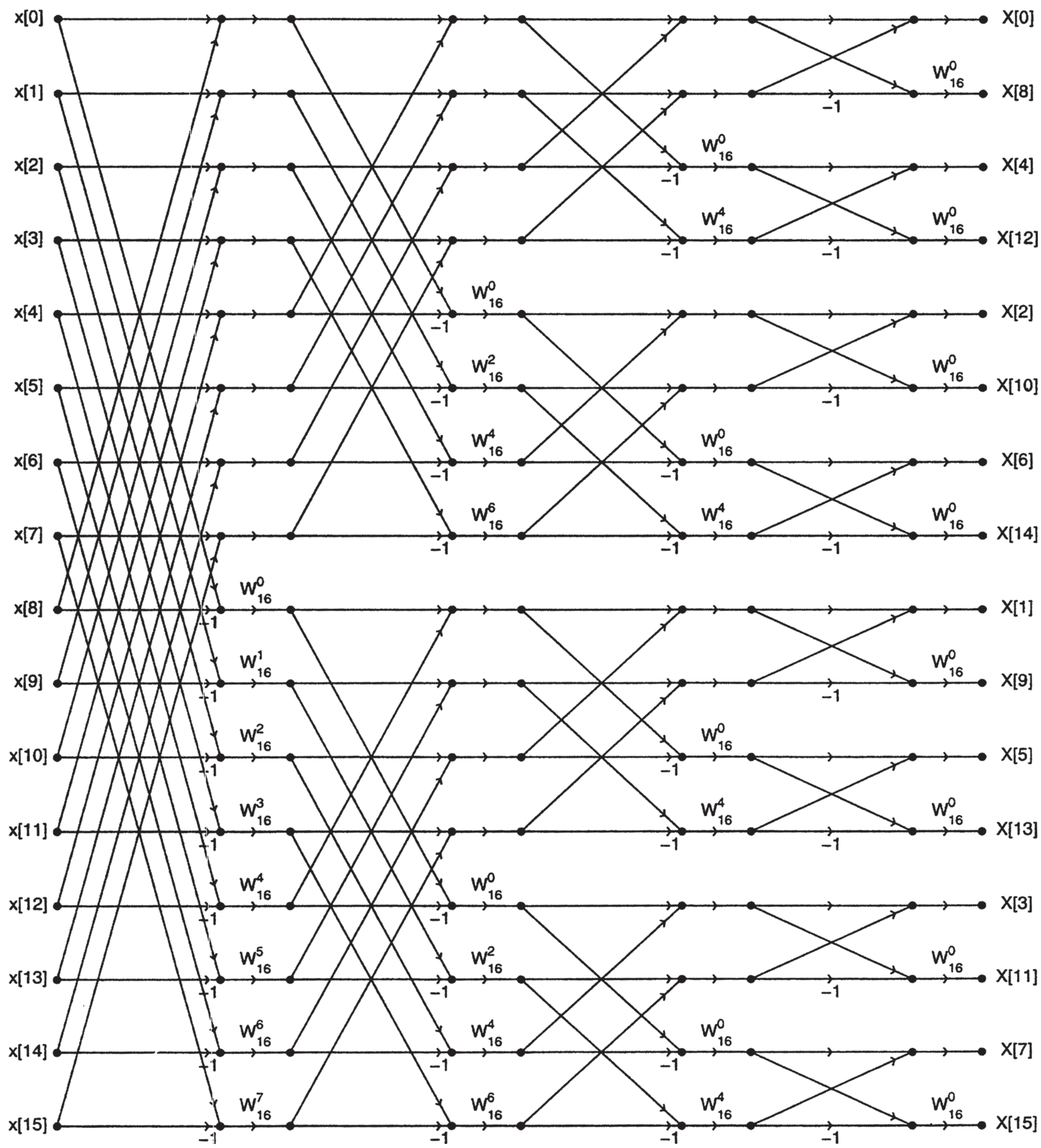


The FFT size was chosen as the next power of 2 higher than the length of the linear convolution. This insures the circular convolution implied by multiplying DFTs corresponds to linear convolution as well.

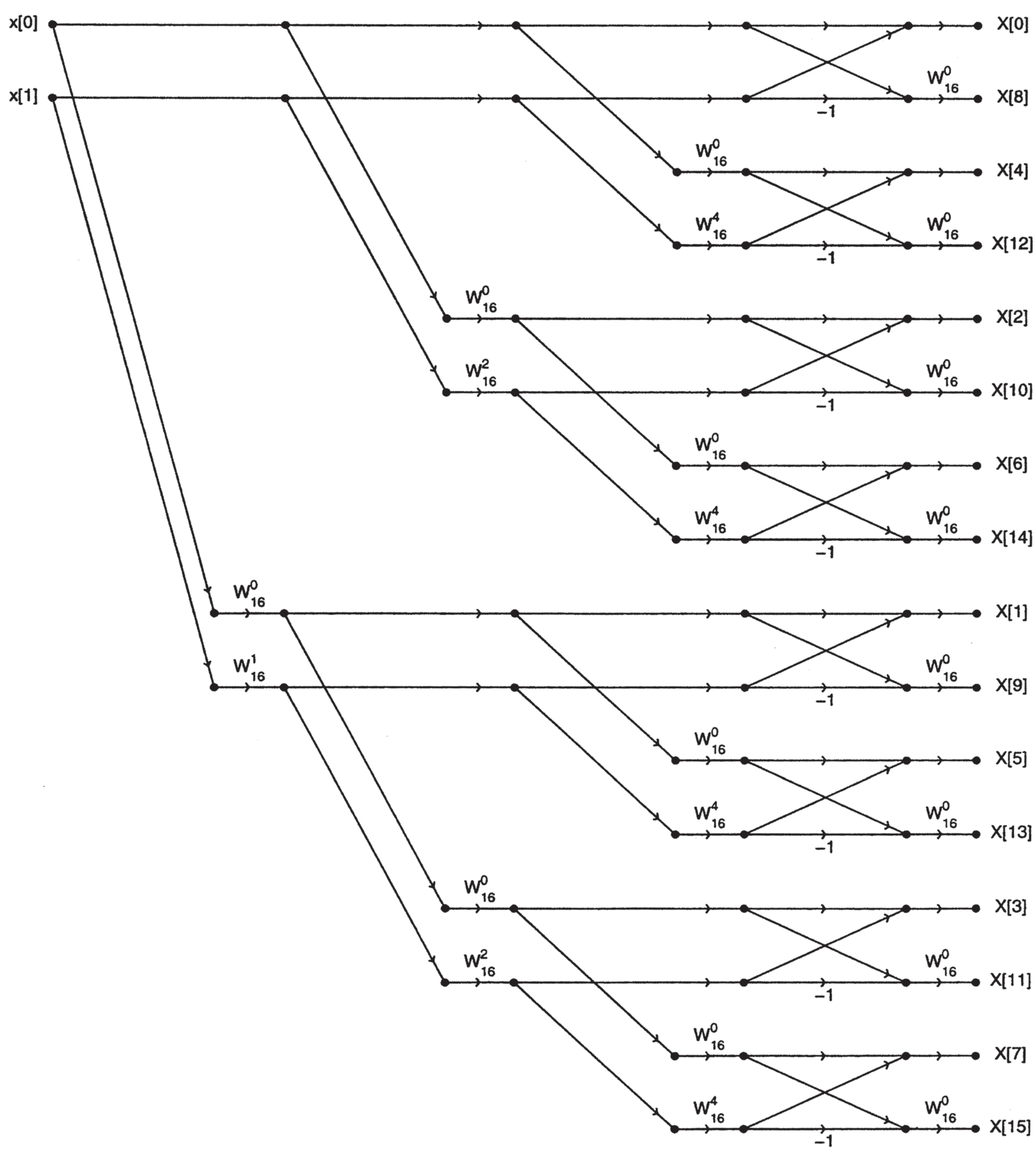
$$
\begin{aligned}
N_{\text{Conv}} &= N_{x_i} + N_{\text{h}} - 1 \\
&= 128 + 64 - 1 \\
&= 191
\end{aligned}
$$

$$N_{\text{FFT}} = 256$$

9.43. (a) The flow graph of a decimation-in-frequency radix-2 FFT algorithm for N = 16 is shown below.



(b) The pruned flow graph is shown below.



(c) The pruned butterflies can be used in $(\nu - \mu)$ stages. For simplicity, assume that $N/2$ complex multiplies are required in each unpruned stage. Counting all W_N^k terms gives

$$
\begin{aligned}
\text{Number of multiplications} &= \text{(Unpruned multiplications)} + \text{(Pruned multiplications)} \\
&= \mu \cdot \frac{N}{2} + \sum_{k=1}^{\nu-\mu} 2^k \\
&= \mu \cdot \frac{2^\nu}{2} + \sum_{k=0}^{\nu-\mu} 2^k - 1 \\
&= \mu \cdot 2^{\nu-1} + \frac{1 - 2^{\nu-\mu+1}}{1-2} - 1 \\
&= \mu \cdot 2^{\nu-1} + 2^{\nu-\mu+1} - 2
\end{aligned}
$$

9.44. (a) Starting with the equation

$$f[n] = x[2n+1] - x[2n-1] + Q, \quad n = 0, 1, \ldots, \frac{N}{2} - 1,$$

where

$$Q = \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} x[2n+1],$$

we note that $x[2n+1] = h[n]$, and $x[2n-1] = h[n-1]$ for $n = 0, 1, \ldots, \frac{N}{2} - 1$. We then get

$$f[n] = h[n] - h[n-1] + Q, \quad n = 0, 1, \ldots, \frac{N}{2} - 1.$$

Taking the $N/2$ point DFT of both sides gives

$$\begin{aligned} F[k] &= H[k] - W_{N/2}^{k} H[k] + Q \sum_{n=0}^{\frac{N}{2}-1} W_{N/2}^{kn} \\ &= H[k](1 - W_N^{2k}) + \frac{N}{2} Q \delta[k] \end{aligned}$$

So

$$\begin{aligned} F[0] &= \frac{N}{2} Q \\ &= \frac{N}{2} \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \\ &= H[0] \end{aligned}$$

Therefore,

$$\begin{aligned} X[k] &= G[k] + W_N^k H[k] \\ X[0] &= G[0] + H[0] = G[0] + F[0] \\ X[N/2] &= G[N/2] + W_N^{N/2} H[N/2] \\ &= G[0] + W_N^{N/2} H[0] \\ &= G[0] - H[0] = G[0] - F[0] \end{aligned}$$

(b) The equation,

$$F[k] = H[k](1 - W_N^{2k}) + \frac{N}{2} Q \delta[k]$$

for $k \neq 0$ becomes

$$\begin{aligned} F[k] &= H[k](1 - W_N^{2k}) \\ &= H[k] W_N^k (W_N^{-k} - W_N^k) \end{aligned}$$

So

$$H[k] = \frac{F[k]}{W_N^k (W_N^{-k} - W_N^k)}$$

Therefore,

$$\begin{aligned} X[k] &= G[k] + W_N^k H[k] \\ &= G[k] + \frac{F[k]}{W_N^{-k} - W_N^k} \\ &= G[k] - \frac{j}{2} \frac{F[k]}{\sin(2\pi k/N)} \end{aligned}$$

Clearly, we need to compute $X[0]$ and $X[N/2]$ with a separate formula since the $\sin(2\pi k/N) = 0$ for $k = 0$ and $k = N/2$.

(c) For each stage of the FFT, the equations

$$\begin{aligned} X[0] &= G[0] + F[0] \\ X[N/2] &= G[0] - F[0] \end{aligned}$$

require 2 real additions each, since the values $G[0]$ and $F[0]$ may be complex. We therefore require a total of 4 real additions to implement these two equations per stage.

For a single stage, the equation

$$X[k] = G[k] - \frac{1}{2}j\frac{F[k]}{\sin(2\pi k/N)} \quad k \neq 0, N/2$$

requires $(N-2)/2$ multiplications of the purely imaginary "twiddle factor" terms by the complex coefficents of $F[k]$ for $k \neq 0, N/2$. The number of multiplications were halved using the symmetry $\sin(2\pi(k + N/2)/N) = -\sin(2\pi k/N)$ and the fact that $F[k]$ is periodic with period $N/2$. Since multiplying a complex number by a purely imaginary number takes 2 real multiplies, we see that the equation requires a total of $(N-2)$ real multiplies per stage.

We also need $(N-2)$ complex additions to add the $G[k]$ and modified $F[k]$ terms for $k \neq 0, N/2$. Since a complex addition requires two real additions, we see that the equation takes a total of $2(N-2)$ real additions per stage.
Putting this all together with the fact that there are $\log_2 N$ stages gives us the totals

$$\begin{aligned} \text{Real Multiplications} &= (N-2)\log_2 N \\ \text{Real Additions} &= 2N\log_2 N \end{aligned}$$

Note that this is approximately half the computation of that of the standard FFT.

(d) The division by $\sin(2\pi k/N)$ for k near 0 and $N/2$ can cause $X[k]$ to get quite large at these values of k. Imagine a signal $x_1[n]$, and signal $x_2[n]$ formed from $x_1[n]$ by adding a small amount of white noise. Using this FFT algorithm, the two FFTs $X_1[k]$ and $X_2[k]$ can vary greatly at such values of k.

9.45. (a)

$$\begin{aligned}
X[2k] &= \sum_{n=0}^{N-1} x[n]W_N^{2kn} \\
&= \sum_{n=0}^{(N/2)-1} \left(x[n]W_N^{2kn} + x[n+(N/2)]W_N^{2k(n+(N/2))}\right) \\
&= \sum_{n=0}^{(N/2)-1} \left(x[n] + x[n+(N/2)]\right) W_N^{2kn}
\end{aligned}$$

In the derivation above, we used the fact that $W_N^{kN} = 1$. Since $W_N^{2kn} = W_{N/2}^{kn}$, $X[2k]$ has been expressed as an $N/2$ point DFT of the sequence $x[n] + x[n+(N/2)]$, $n = 0, 1, \ldots, (N/2)-1$.

(b)

$$\begin{aligned}
X[4k+1] &= \sum_{n=0}^{N-1} x[n]W_N^{(4k+1)n} \\
&= \sum_{n=0}^{(N/4)-1} \Big(x[n]W_N^{n}W_N^{4kn} + x[n+(N/4)]W_N^{n+(N/4)}W_N^{4k(n+(N/4))} \\
&\quad + x[n+(N/2)]W_N^{n+(N/2)}W_N^{4k(n+(N/2))} + x[n+(3N/4)]W_N^{n+(3N/4)}W_N^{4k(n+(3N/4))}\Big) \\
&= \sum_{n=0}^{(N/4)-1} \{(x[n] - x[n+(N/2)]) - j(x[n+(N/4)] - x[n+(3N/4)])\}W_N^{n}W_N^{4kn}
\end{aligned}$$

In the derivation above, we used the fact that $W_N^{N/4} = -j$, $W_N^{3N/4} = j$, $W_N^{N/2} = -1$, and $W_N^{kN} = 1$. Since $W_N^{4kn} = W_{N/4}^{kn}$, $X[4k+1]$ has been expressed as a $N/4$ point DFT. But we need to multiply the sequence $(x[n] - x[n+(N/2)]) - j(x[n+(N/4)] - x[n+(3N/4)])$ by the twiddle factor W_N^n, $0 \le n \le (N/4)-1$ *before* we compute the $N/4$ point DFT.

The other odd-indexed terms can be shown in the same way to be

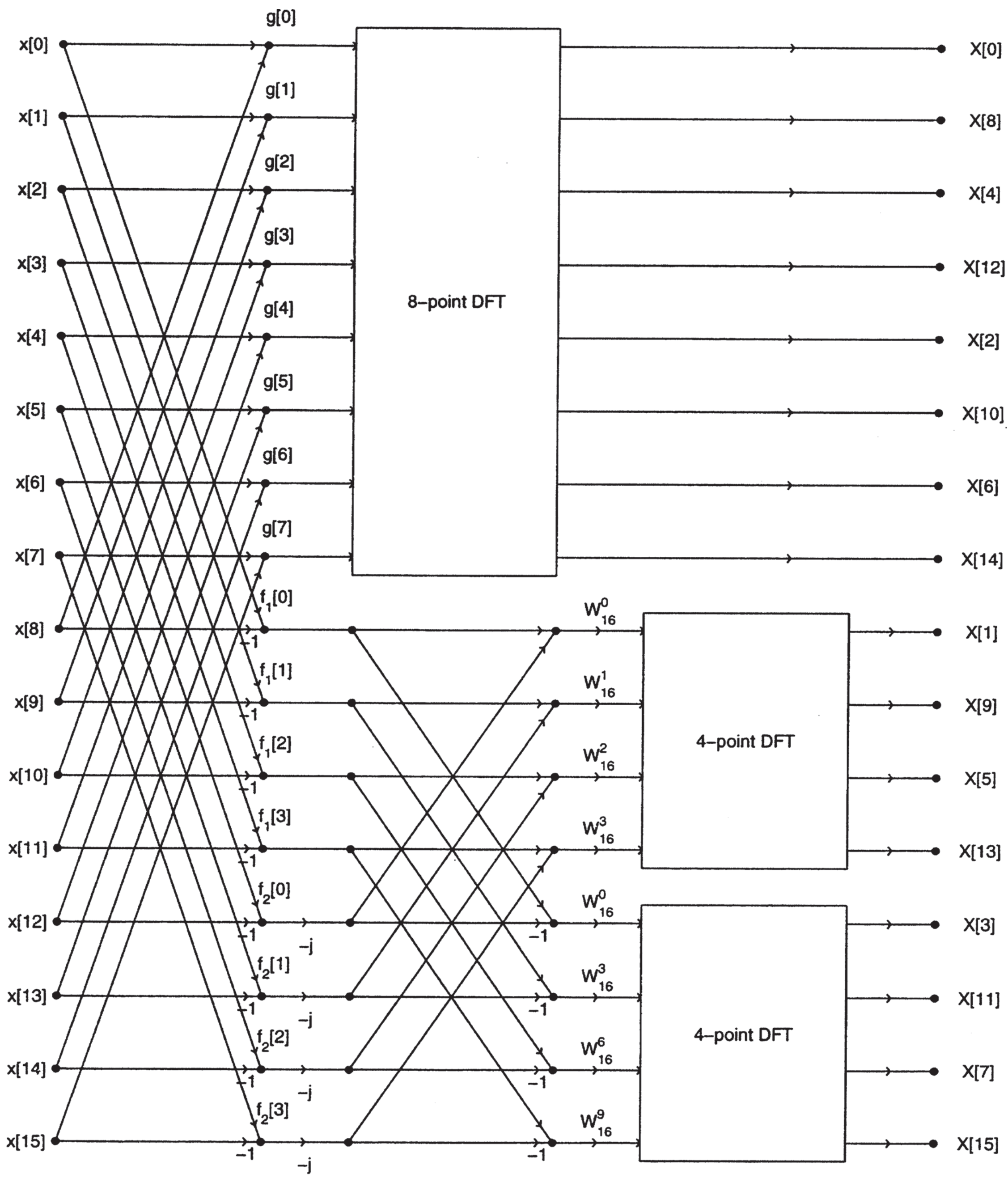
$$\begin{aligned}
X[4k+3] &= \sum_{n=0}^{(N/4)-1} \{(x[n] - x[n+(N/2)]) \\
&\quad + j(x[n+(N/4)] - x[n+(3N/4)])\}W_N^{3n}W_N^{4kn}, \quad k = 0, 1, \ldots, (N/4)-1.
\end{aligned}$$

Parts (a) and (b) show that we can replace the computation of an N point DFT with the computation of one $N/2$ point DFT, two $N/4$ point DFTs, and some extra complex arithmetic.

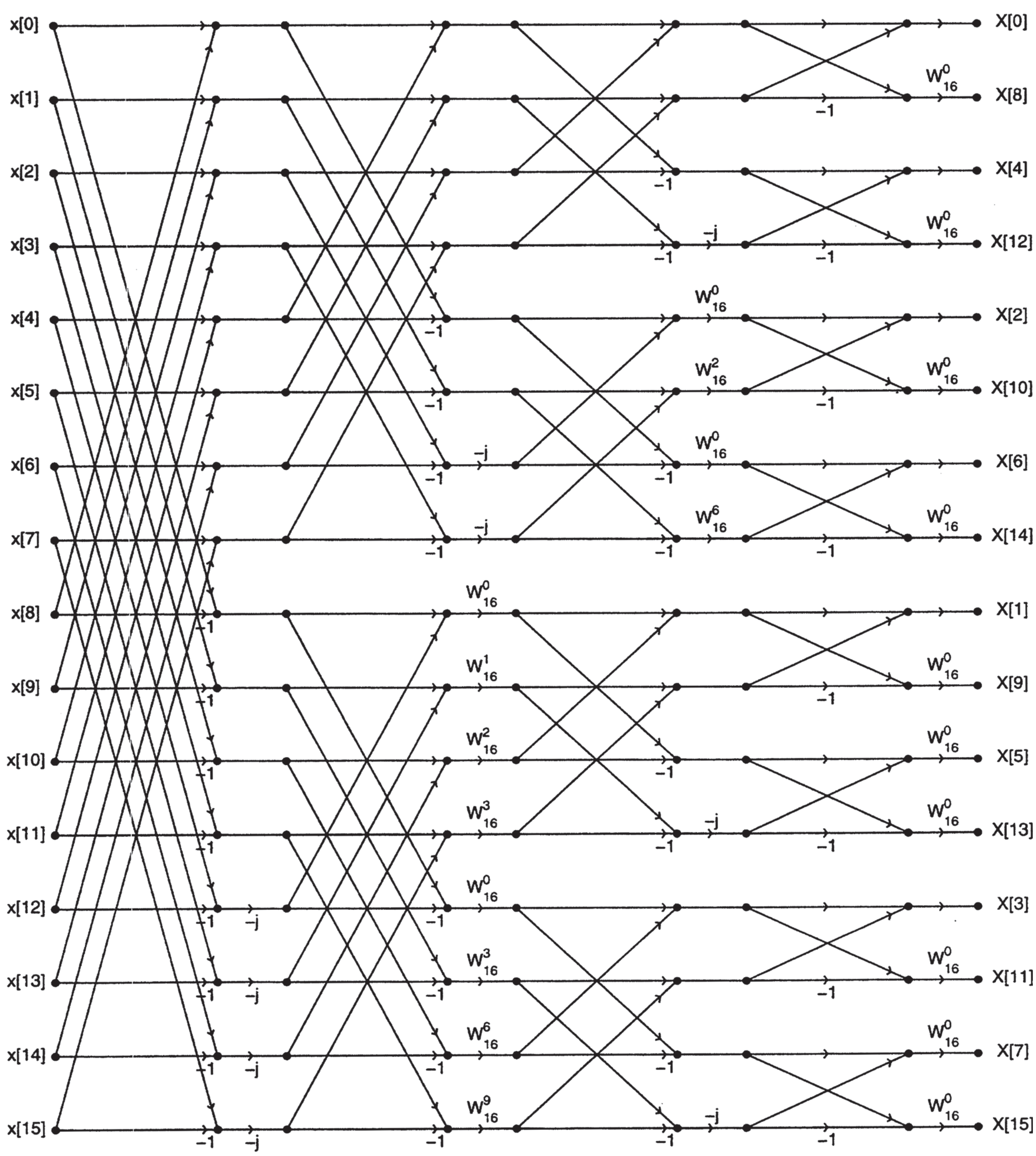
(c) Assume $N = 16$ and define

$$\begin{aligned}
g[n] &= x[n] + x[n+(N/2)], \quad n = 0, 1, \ldots, (N/2)-1 \\
f_1[n] &= x[n] - x[n+(N/2)], \quad n = 0, 1, \ldots, (N/4)-1 \\
f_2[n] &= x[n+(N/4)] - x[n+(3N/4)], \quad n = 0, 1, \ldots, (N/4)-1
\end{aligned}$$

A diagram for computing the values of $X[k]$ looks like:



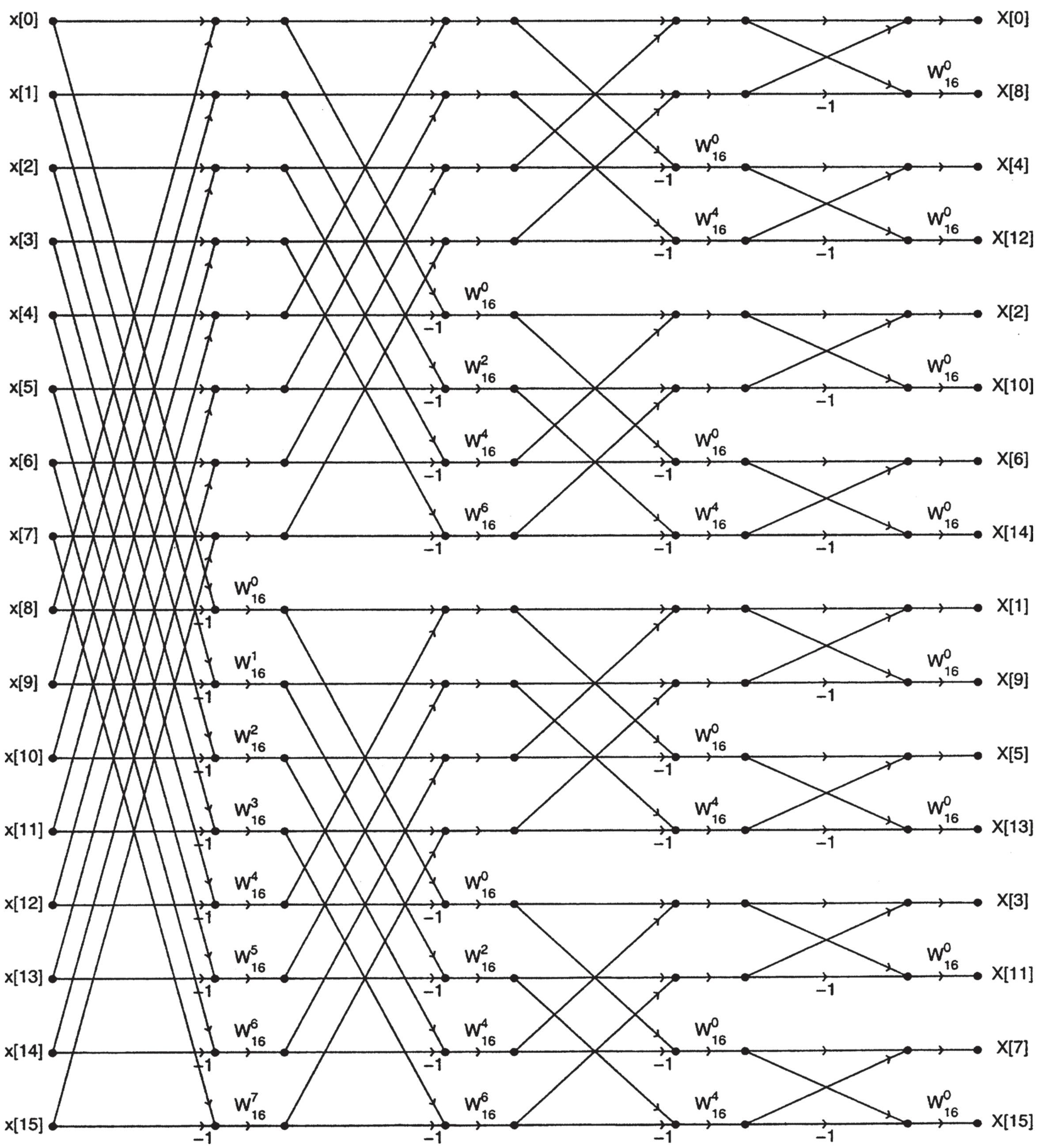
If we carry on applying the split-radix principle, we get the next diagram:



(d) The flow diagram for the regular radix-2 decimation-in-frequency algorithm is shown in the next figure for $N = 16$. Not counting trivial multiplications by W_N^0, we find that there are 17 complex multiplications total. Of these 17 complex multiplications, 7 are multiplications by $W_{16}^4 = -j$. Since a multiplication by $-j$ can be done with zero real multiplications, and a complex multiplication requires 4 real multiplications, we find that the total number of real multiplications for the decimation-in-frequency algorithm to be $(10)(4) = 40$.

Taking a look at the split-radix algorithm, we find again that there are 17 complex multiplications.

In this case, however, 9 of these are by $W_{16}^4 = -j$. Thus, it takes a total of $(8)(4) = 32$ real multiplications to implement this flow graph.



9.46. (a) Noting that

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots \qquad |x| < 1$$

$$\theta_i = \arctan(2^{-i}) = 2^{-i} - \frac{2^{-3i}}{3} + \frac{2^{-5i}}{5} - \frac{2^{-7i}}{7} + \ldots$$

we notice that, to first order, the θ_i and θ_{i+1} differ by a factor of 2. (Note that these formulae are in radians). This approximate factor of 2 for sucessive θ_i is confirmed by looking at some values of θ_i: $\theta_0 = 45°$, $\theta_1 = 26.6°$, $\theta_2 = 14.0°$, $\theta_3 = 7.1°$. So we have a set of angles whose values are decreasing by about a factor of 2.

You can add and subtract these θ_i angles to form any angle $0 < \theta < \pi/2$. The error is bound by $\theta_M = \arctan(2^{-M})$, the angle that would be included next in the sum. If the error were greater than θ_M, then one of the α_i terms must have been incorrect. The inclusion of the Mth term must bring the sum closer to θ.

(b) An algorithm to compute θ_i is described below.

$\alpha_0 = +1$
$\hat{\theta} = \alpha_0\theta_0$
for $i = 1$ to $M - 1$
    if $(\hat{\theta} > \theta)$
        $\alpha_i = -1$
    else
        $\alpha_i = +1$
    endif
    $\hat{\theta} = \alpha_i\theta_i + \hat{\theta}$
end for

Using this algorithm, the sequence α_i is found to be

i	0	1	2	3	4	5	6	7	8	9	10
α_i	1	−1	1	1	−1	−1	1	1	−1	−1	−1

(c) Note that $(X + jY)(1 + j\alpha_i 2^{-i}) = (X - \alpha_i Y 2^{-i}) + j(Y + \alpha_i X 2^{-i})$. Hence, the recursion is simply multiplying by M complex numbers of the form $(1 + j\alpha_i 2^{-i})$. These can be represented in polar form:

$$\begin{aligned}(1 + j\alpha_i 2^{-i}) &= \sqrt{1 + 2^{-2i}}\, e^{j\alpha_i \arctan(2^{-i})} \\ &= G_i e^{j\alpha_i\theta_i}\end{aligned}$$

Multiplication of polar numbers produces a sum of the phases, $\alpha_i\theta_i$.

$$\hat{\theta} = \sum_{i=0}^{M-1} \alpha_i\theta_i$$

(d) Multiplication of polar numbers produces a product of the magnitudes, G_i.

$$G_M = \prod_{i=0}^{M-1} \sqrt{1 + 2^{-2i}}$$

9.47. (a) Starting with the definition of the DFT,

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

$$\begin{aligned}X[3k] &= \sum_{n=0}^{N-1} x[n] W_N^{3nk} \\ &= \sum_{n=0}^{N/3-1} x[n] W_N^{3nk} + \sum_{n=N/3}^{2N/3-1} x[n] W_N^{3nk} + \sum_{n=2N/3}^{N-1} x[n] W_N^{3nk}\end{aligned}$$

Substituting $m = n - N/3$ into the second summation, and $m = n - 2N/3$ into the third summation gives

$$\begin{aligned}X[3k] &= \sum_{n=0}^{N/3-1} x[n] W_N^{3nk} + \sum_{m=0}^{N/3-1} x[m + N/3] W_N^{3mk} W_N^{Nk} + \sum_{m=0}^{N/3-1} x[m + 2N/3] W_N^{3mk} W_N^{2Nk} \\ &= \sum_{n=0}^{N/3-1} (x[n] + x[n + N/3] + x[n + 2N/3]) W_N^{3nk} \\ &= \sum_{n=0}^{N/3-1} (x[n] + x[n + N/3] + x[n + 2N/3]) W_{N/3}^{nk}\end{aligned}$$

Define the sequence

$$x_1[n] = x[n] + x[n + N/3] + x[n + 2N/3]$$

The 3-point DFT of $x_1[n]$ is $X_1[k] = X[3k]$.

(b) This part is similar to part (a). First, $x_2[n]$ is found. Starting again with the definition of the DFT,

$$\begin{aligned}X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\ X[3k+1] &= \sum_{n=0}^{N-1} x[n] W_N^{n(3k+1)} \\ &= \sum_{n=0}^{N/3-1} x[n] W_N^{n(3k+1)} + \sum_{n=N/3}^{2N/3-1} x[n] W_N^{n(3k+1)} + \sum_{n=2N/3}^{N-1} x[n] W_N^{n(3k+1)}\end{aligned}$$

Substituting $m = n - N/3$ into the second summation, and $m = n - 2N/3$ into the third summation gives

$$\begin{aligned}X[3k+1] &= \sum_{n=0}^{N/3-1} x[n] W_N^{n(3k+1)} + \sum_{m=0}^{N/3-1} x[m + N/3] W_N^{(m+N/3)(3k+1)} \\ &\quad + \sum_{m=0}^{N/3-1} x[m + 2N/3] W_N^{(m+2N/3)(3k+1)} \\ &= \sum_{n=0}^{N/3-1} x[n] W_N^{n(3k+1)} + \sum_{m=0}^{N/3-1} x[m + N/3] W_N^{m(3k+1)} W_N^{Nk} W_N^{N/3} \\ &\quad + \sum_{m=0}^{N/3-1} x[m + 2N/3] W_N^{m(3k+1)} W_N^{2Nk} W_N^{2N/3} \\ &= \sum_{n=0}^{N/3-1} (x[n] + x[n + N/3] W_N^{N/3} + x[n + 2N/3] W_N^{2N/3}) W_N^{n(3k+1)}\end{aligned}$$

$$= \sum_{n=0}^{N/3-1} (x[n] + x[n+N/3]W_N^{N/3} + x[n+2N/3]W_N^{2N/3})W_N^n W_{N/3}^{kn}$$

Define the sequence

$$x_2[n] = (x[n] + x[n+N/3]W_N^{N/3} + x[n+2N/3]W_N^{2N/3})W_N^n$$

The 3-point DFT of $x_2[n]$ is $X_2[k] = X[3k+1]$. Next, $x_3[n]$ is found in a similar manner. Starting with the definition of the DFT,

$$\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n]W_N^{nk} \\
X[3k+2] &= \sum_{n=0}^{N-1} x[n]W_N^{n(3k+2)} \\
&= \sum_{n=0}^{N/3-1} x[n]W_N^{n(3k+2)} + \sum_{n=N/3}^{2N/3-1} x[n]W_N^{n(3k+2)} + \sum_{n=2N/3}^{N-1} x[n]W_N^{n(3k+2)}
\end{aligned}$$

Substituting $m = n - N/3$ into the second summation, and $m = n - 2N/3$ into the third summation gives

$$\begin{aligned}
X[3k+2] &= \sum_{n=0}^{N/3-1} x[n]W_N^{n(3k+2)} + \sum_{m=0}^{N/3-1} x[m+N/3]W_N^{(m+N/3)(3k+2)} \\
&\quad + \sum_{m=0}^{N/3-1} x[m+2N/3]W_N^{(m+2N/3)(3k+2)} \\
&= \sum_{n=0}^{N/3-1} x[n]W_N^{n(3k+2)} + \sum_{m=0}^{N/3-1} x[m+N/3]W_N^{m(3k+2)}W_N^{Nk}W_N^{2N/3} \\
&\quad + \sum_{m=0}^{N/3-1} x[m+2N/3]W_N^{m(3k+2)}W_N^{2Nk}W_N^{4N/3} \\
&= \sum_{n=0}^{N/3-1} (x[n] + x[n+N/3]W_N^{2N/3} + x[n+2N/3]W_N^{4N/3})W_N^{n(3k+2)} \\
&= \sum_{n=0}^{N/3-1} (x[n] + x[n+N/3]W_N^{2N/3} + x[n+2N/3]W_N^{4N/3})W_N^{2n}W_{N/3}^{kn}
\end{aligned}$$

Define the sequence

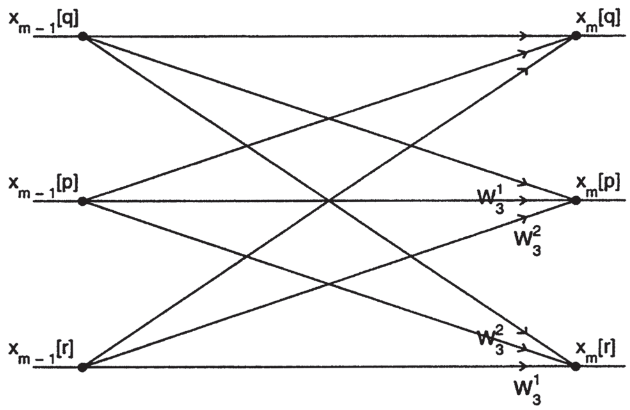
$$x_3[n] = (x[n] + x[n+N/3]W_N^{2N/3} + x[n+2N/3]W_N^{4N/3})W_N^{2n}$$

The 3-point DFT of $x_3[n]$ is $X_3[k] = X[3k+2]$.

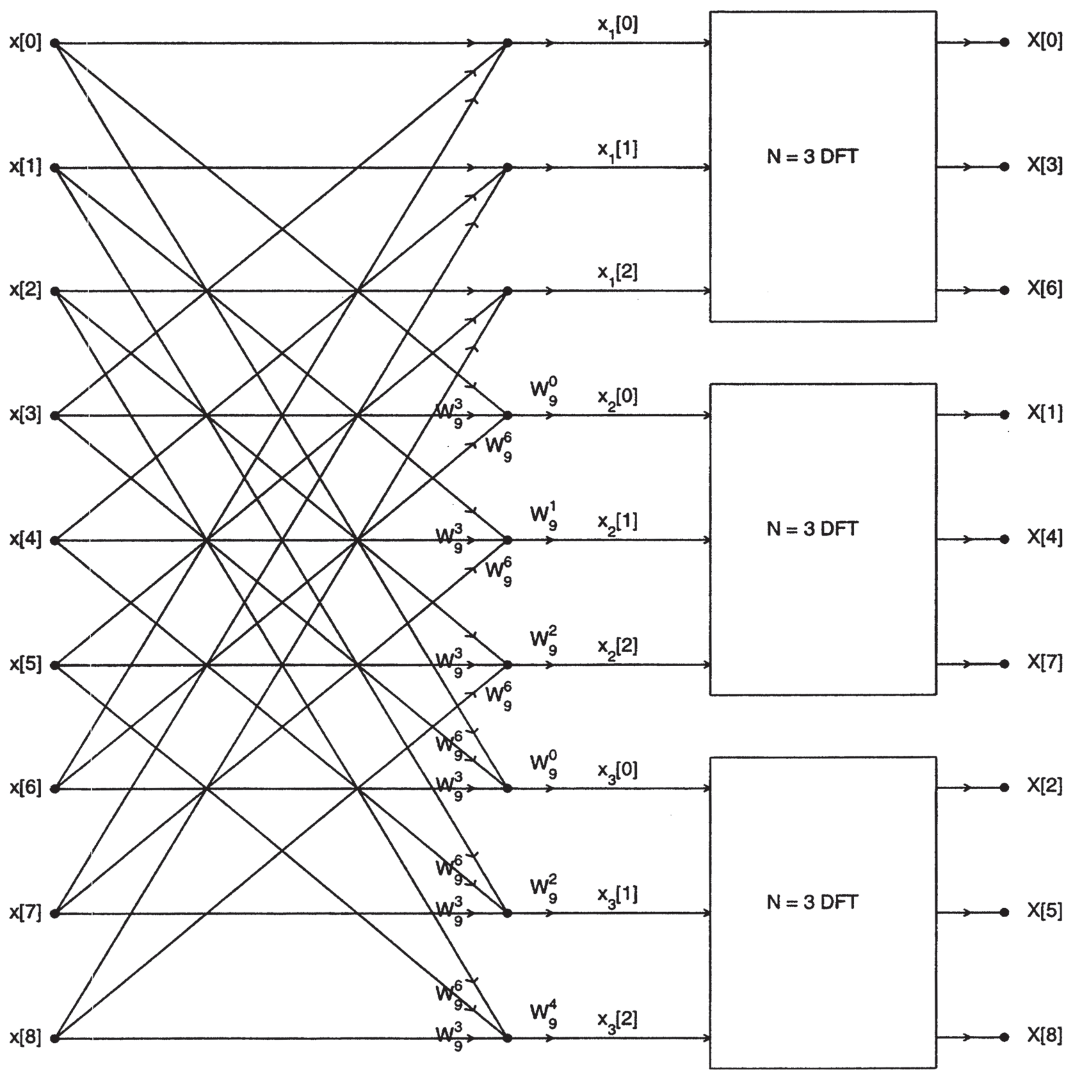
(c) To draw the radix-3 butterfly, it helps to derive the output of the butterfly first. From the definition of the DFT,

$$\begin{aligned}
X[k] &= \sum_{n=0}^{2} x[n]W_3^{nk} \\
X[0] &= x[0] + x[1] + x[2] \\
X[1] &= x[0] + x[1]W_3^1 + x[2]W_3^2 \\
X[2] &= x[0] + x[1]W_3^2 + x[2]W_3^4 = x[0] + x[1]W_3^2 + x[2]W_3^1
\end{aligned}$$

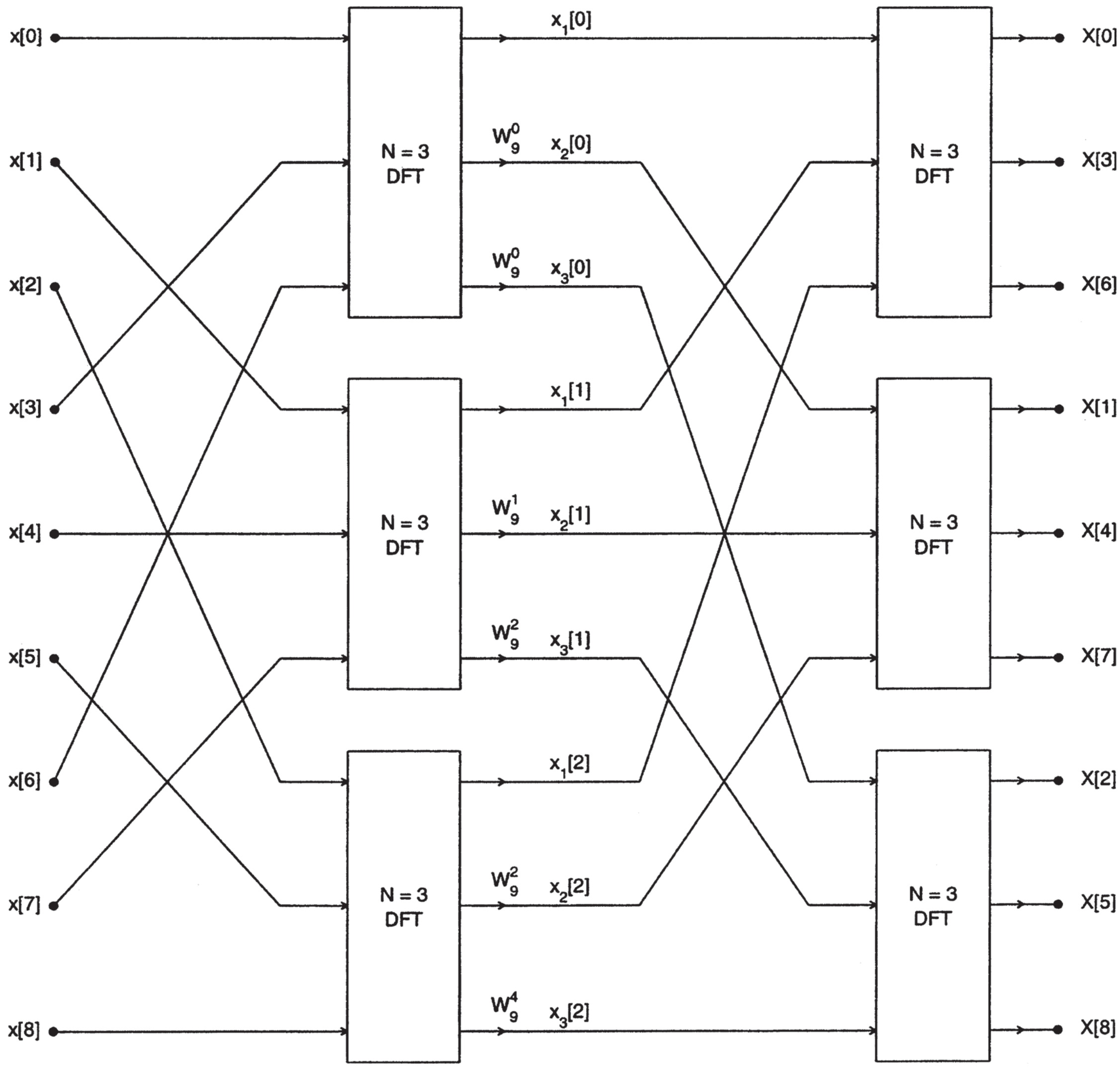
The butterfly for the 3 point DFT is drawn below.



(d) Using the results from parts (a) and (b), the flow graph is drawn below.



(e) The system consisting entirely of $N = 3$ DFTs is drawn below.



(f) A direct implementation of the 9 point DFT equation requires $9^2 = 81$ complex multiplications. The system in part (e), in constrast, requires 4 complex multiplications for each 3 point DFT, and an additional 4 from the twiddle factors, if we do not count the trival W_N^0 multiplications. In total, the system in part (e) requires 28 complex multiplications. In general, a radix-3 FFT of a sequence of length $N = 3^\nu$ requires *approximately*

$$\text{Number of complex multiplications} = \left(4\frac{\text{complex multiplications}}{\text{3-pt DFT}}\right)\left(\frac{N}{3}\frac{\text{3-pt DFTs}}{\text{stage}}\right)(\nu \text{ stages}) + \left(\frac{\text{N twiddle factors}}{\text{stage with twiddle factors}}\right)(\nu - 1 \text{ stages with twiddle factors})$$

Replacing ν with $\log_3 N$, and simplifying this formula gives

$$\text{Number of complex multiplications} = \frac{7}{3}N\log_3 N - N$$

Note that this formula for a radix-3 FFT is of the form $N\log_3 N$. The constant multiplier, $\frac{7}{3}$, is significantly larger than that of a radix-2 FFT. This is because a radix-2 butterfly has no complex multiplications, while in part (c) we found that a radix-3 butterfly has 4 complex multiplications. Also note that this formula is an upper bound, since some of the N twiddle factors in the $\nu - 1$ stages will be trivial. However, the formula is a good estimate.

9.48. (a)

$$\begin{aligned} X[k] &= h^*[k]\sum_{n=0}^{N-1}(x[n]h^*[n])h[k-n] \\ &= e^{-j\pi k^2/N}\sum_{n=0}^{N-1}x[n]e^{-j\pi n^2/N}e^{j\pi(k^2-2kn+n^2)/N} \\ &= \sum_{n=0}^{N-1}x[n]e^{-j2\pi nk/N} \end{aligned}$$

(b)

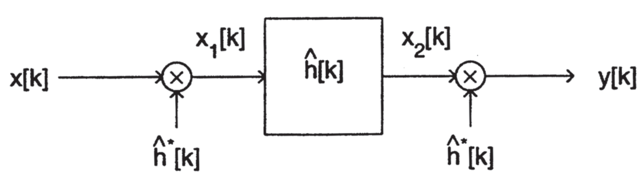
$$X[k+N] = h^*[k+N]\sum_{n=0}^{N-1}x[n]h^*[n]h[k+N-n]$$

$$\begin{aligned} h^*[k+N] &= e^{-j\pi(k+N)^2/N} \\ &= e^{-j\pi(k^2+2kN+N^2)/N} \\ &= e^{-j\pi k^2/N}e^{-j\pi N} \\ &= h^*[k]e^{-j\pi N} \end{aligned}$$

So

$$\begin{aligned} X[k+N] &= h^*[k]e^{-j\pi N}\sum_{n=0}^{N-1}x[n]h^*[n]h[k-n]e^{j\pi N} \\ &= X[k] \end{aligned}$$

(c) From the figure



we define the signals $x_1[k]$ and $x_2[k]$ to be

$$\begin{aligned} x_1[k] &= x[k]\hat{h}^*[k] \\ x_2[k] &= \sum_{\ell=-\infty}^{\infty}x_1[\ell]\hat{h}[k-\ell] \\ &= \sum_{\ell=0}^{N-1}x[\ell]e^{-j\pi\ell^2/N}e^{j\pi(k-\ell)^2/N} \\ & \quad k\in[0,\ldots,2N-1] \end{aligned}$$

Therefore,

$$\begin{aligned} y[k] &= \hat{h}^*[k]x_2[k] \qquad k\in[N,\ldots,2N-1] \\ &= e^{-j\pi k^2/N}\sum_{\ell=0}^{N-1}x[\ell]e^{-j\pi\ell^2/N}e^{j\pi(k-\ell)^2/N} \qquad k\in[N,\ldots,2N-1] \\ &= \sum_{\ell=0}^{N-1}x[\ell]e^{-j2\pi k\ell/N} \qquad k\in[N,\ldots,2N-1] \\ &= X[k+N] \qquad k\in[0,\ldots,N-1] \\ &= X[k] \end{aligned}$$

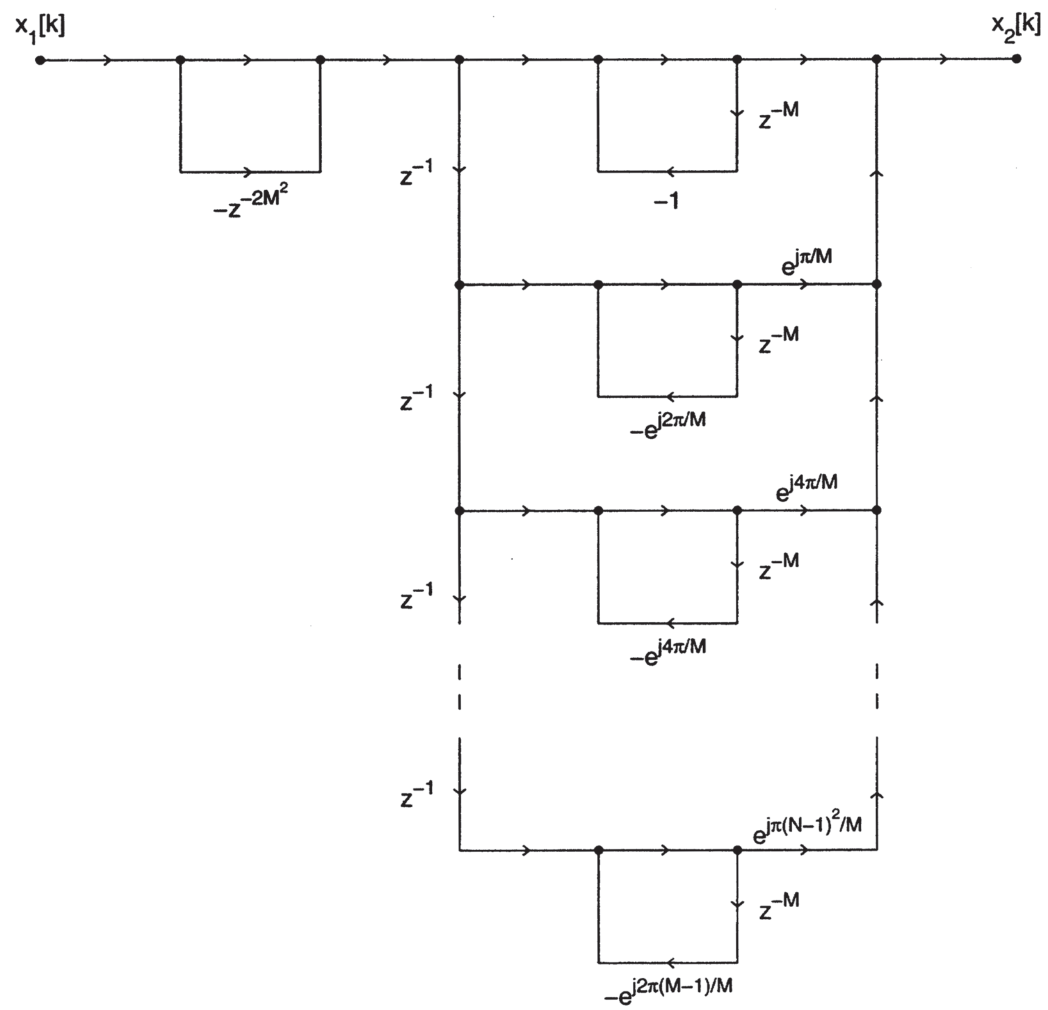
(d)

$$\begin{aligned}
\hat{H}(z) &= \sum_{k=0}^{2N-1} e^{j\pi k^2/N} z^{-k} \\
&= \sum_{r=0}^{M-1} \sum_{\ell=0}^{2M-1} e^{j\pi(r+\ell M)^2/N} z^{-(r+\ell M)} \\
&= \sum_{r=0}^{M-1} \sum_{\ell=0}^{2M-1} e^{j\pi r^2/N} e^{j2\pi r\ell/M} e^{j\pi \ell^2} z^{-r} z^{-\ell M} \\
&= \sum_{r=0}^{M-1} e^{j\pi r^2/N} z^{-r} \left[\sum_{\ell=0}^{2M-1} (e^{j2\pi r/M} z^{-M})^{\ell} (-1)^{\ell^2} \right] \\
&= \sum_{r=0}^{M-1} e^{j\pi r^2/N} z^{-r} \left[\sum_{\ell=0}^{2M-1} (e^{j2\pi r/M} z^{-M})^{\ell} (-1)^{\ell} \right] \\
&= \sum_{r=0}^{M-1} e^{j\pi r^2/N} z^{-r} \left[\sum_{\ell=0}^{2M-1} (-e^{j2\pi r/M} z^{-M})^{\ell} \right] \\
&= \sum_{r=0}^{M-1} e^{j\pi r^2/N} z^{-r} \left[\frac{1 - e^{j2\pi r(2M)/M} z^{-2M^2}}{1 + e^{j2\pi r/M} z^{-M}} \right] \\
&= \sum_{r=0}^{M-1} e^{j\pi r^2/N} z^{-r} \left[\frac{1 - z^{-2M^2}}{1 + e^{j(2\pi/M)r} z^{-M}} \right]
\end{aligned}$$

(e) The flow graph for the system,

$$\hat{H}(z) = (1 - z^{-2M^2}) \sum_{r=0}^{M-1} \frac{z^{-r} e^{j\pi r^2/M}}{1 + e^{j2\pi r/M} z^{-M}}$$

is drawn below.



(f) **Complex multiplications:** Since we are only interested in $y[k]$ for $k = N, N+1, \dots, 2N-1$ we do not need to calculate the complex multiplications on the output side of each parallel branch until $k \geq N$. Thus,

Operation	Complex Multiplications
$x_1[k] = x[k]\hat{h}^*[k]$	N
Poles of $\hat{H}(z)$ for $k = 0, \dots, N-1$	NM
Poles and branch exponentials of $\hat{H}(z)$ for $k = N, \dots, 2N-1$	$2NM$
$y[k] = x_2[k]\hat{h}^*[k]$	N

$$= 3MN + 2N$$
$$= 3M^3 + 2M^2$$

We are including the multiplication of -1 and 1 from the first branch here for simplicity.
Direct evaluation requires $N^2 = M^4$ complex multiplications.
Note that since we are only interested in $y[k]$ for $k = N, N+1, \dots, 2N-1$, the initial delay of $2M^2$ is unneccessary; we have obtained all interesting output before the first delayed sample appears.

Complex additions: The complex additions on the output side of each parallel branch do not need to be computed until $k \geq N$. Thus,

Operation	Complex Additions
Poles of $\hat{H}(z)$ for $k = 0, \dots, N-1$	NM
Poles and branch exponentials of $\hat{H}(z)$ for $k = N, \dots, 2N-1$	$2NM$

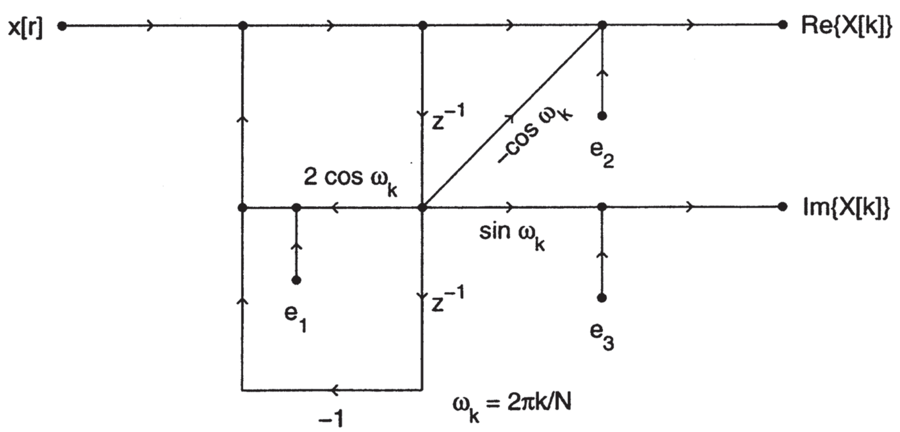
$$= 3MN$$
$$= 3M^3$$

Direct evaluation requires $N(N-1) = M^2(M^2-1)$ complex additions.

9.49. (a) We separate the system function $H(z)$ into two pieces; one corresponding to $h_R[n] = Re\{h[n]\}$ and another corresponding to $h_I[n] = Im\{h[n]\}$.

$$\begin{aligned}
H(z) &= \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \\
H_R(z) &= \frac{1 - \cos(2\pi k/N)z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \\
h_R[n] &= \cos(2\pi kn/N)u[n] \\
H_I(z) &= \frac{\sin(2\pi k/N)z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \\
h_I[n] &= \sin(2\pi kn/N)u[n]
\end{aligned}$$

Since $x[n]$ is real, the real and imaginary parts of $X[k] = y_k[N]$ are computed using the following flowgraph.



(b)

$$\begin{aligned}
\text{Re}\{X[k]\} &= \text{Re}\{y_k[N]\} \\
\text{Im}\{X[k]\} &= \text{Im}\{y_k[N]\}
\end{aligned}$$

Since the output of interest is the Nth sample, we need only consider the variance at time N. The noise $e_1[n]$ is input to both $h_R[n]$ and $h_I[n]$. Using the techniques from chapter 6, we find the variance of the noise is

$$\begin{aligned}
\sigma_R^2[N] &= \sigma_{e_2}^2 + \sigma_{e_1}^2 \sum_{n=0}^{N} h_R^2[n] \\
\sigma_I^2[N] &= \sigma_{e_3}^2 + \sigma_{e_1}^2 \sum_{n=0}^{N} h_I^2[n].
\end{aligned}$$

Let $\theta = 2\pi k/N$.

$$\begin{aligned}
\sum_{n=0}^{N} h_R^2[n] &= \sum_{n=0}^{N} \cos^2 \theta n \\
&= \frac{1}{4}\sum_{n=0}^{N}(e^{j\theta n} + e^{-j\theta n})^2 \\
&= \frac{1}{4}\sum_{n=0}^{N}(e^{j\theta 2n} + 2 + e^{-j\theta 2n}) \\
&= \frac{1}{4}\left(\frac{1-e^{j2\theta(N+1)}}{1-e^{j2\theta}} + 2(N+1) + \frac{1-e^{-j2\theta(N+1)}}{1-e^{-j2\theta}}\right) \\
&= \frac{1}{4}(1 + 2(N+1) + 1) \\
&= \frac{N}{2} + 1
\end{aligned}$$

Similarly, $\sum_{n=0}^{N} h_I^2[n] = N/2$.
Therefore,

$$\begin{aligned}
\sigma_R^2[n] &= \frac{2^{-2B}}{12}(1 + (N/2) + 1) \\
&= \frac{2^{-2B}}{12}(N+4)/2 \\
\sigma_I^2[n] &= \frac{2^{-2B}}{12}(1 + (N/2)) \\
&= \frac{2^{-2B}}{12}(N+2)/2
\end{aligned}$$

9.50.

$$X[k] = \sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N) - j\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N).$$

For $k \neq 0$, there are $N-1$ multiplies in the computation of the real part and the imaginary part:

$$\sigma_R^2 = (N-1)\sigma^2 = \sigma_I^2,$$

where $\sigma^2 = 2^{-2B}/12$. For $k = 0$ there are no multiplies, and therefore $\sigma_R^2 = 0 = \sigma_I^2$.

$$\sigma_R^2 = \sigma_I^2 = \begin{cases} 0 & k = 0 \\ (N-1)\sigma^2 & k \neq 0 \end{cases}$$

9.51. (a)

$$|X_{m-1}[q]W_N^r| = |X_{m-1}[q]||W_N^r| = |X_{m-1}[q]|$$

$$|X_m[p]| \leq |X_{m-1}[p]| + |X_{m-1}[q]| < \frac{1}{2} + \frac{1}{2} = 1$$

implies that $|\text{Re}\{X_m[p]\}| < 1$ and $|\text{Im}\{X_m[p]\}| < 1$. A similar argument holds for $|X_m[q]|$.

(b) The conditions are not sufficient to guarantee that overflow cannot occur.

$$\begin{aligned} |\text{Re}\{X_m[p]\}| &\leq |\text{Re}\{X_m[p]\}| + |\text{Re}\{W_N^r X_{m-1}[q]\}| \\ &\leq \frac{1}{2} + \left|\cos\left(\frac{2\pi r}{N}\right)\text{Re}\{X_{m-1}[q]\} - \sin\left(\frac{2\pi r}{N}\right)\text{Im}\{X_{m-1}[q]\}\right| \end{aligned}$$

Consider the worst case, when $r = 3N/8$. Then,

$$\begin{aligned} |\text{Re}\{X_m[p]\}| &\leq \frac{1}{2} + \left|\frac{1}{\sqrt{2}}\text{Re}\{X_{m-1}[q]\} + \frac{1}{\sqrt{2}}\text{Im}\{X_{m-1}[q]\}\right| \\ &\leq \frac{1}{2} + \frac{1}{\sqrt{2}} \not< 1. \end{aligned}$$

Therefore, overflow can occur.

9.52. (a) First, note that each stage has $N/2$ butterflies. In the first stage, all the multiplications are $W_N^0 = +1$. In the second stage, half are $+1$ and the other half are $W_N^{N/4} = -j$. Successive stages have half the number of the previous stage. In general,

$$\text{Number of } +1 \text{ multiplications in stage } m = \frac{N}{2^m}, \qquad m = 1, \ldots, v$$

and

$$\text{Number of } -j \text{ multiplications in stage } m = \begin{cases} 0, & m = 1 \\ N/2^m, & m = 2, \ldots, v \end{cases}$$

(b) If we assume that all the $+1$ and $-j$ multiplications are done noiselessly, then the noise variance will be different at each output node. This is easily seen by looking at Figure 9.10, where we see for example that $X[0]$ will be noise-free, while $X[1]$ will not be noise-free. Thus, a noise analysis would be required for each output node separately. A somewhat simpler approach would be to assume that since the first two stages consist of only $+1$ and $-j$ multiplications, these two stages can be performed noiselessly. Each output node is connected to all $N/2$ butterflies in the first stage and to $N/4$ butterflies in the second stage. Thus, if the first two stages are performed noiselessly, a better estimate of the number of independent noise sources contributing to the output is

$$N - 1 - \frac{N}{2} - \frac{N}{4} = \frac{N}{4} - 1.$$

Note that all the odd indexed outputs will have exactly $(N/4) - 1$ of these noise sources, while the even indexed outputs will have less. In fact, $X[0]$, $X[N/4]$, $X[N/2]$, $X[3N/4]$ will be noiseless. $X[N/8]$, $X[3N/8]$, $X[5N/8]$, and $X[7N/8]$ will have one noise source. It is possible to continue this analysis for all $X[k]$, but clearly, a complicated formula would be required to describe the number of noise sources for all even k. We have shown that

$$\begin{aligned} \text{Number of noise sources} &= (N/4) - 1, & k \text{ odd} \\ \text{Number of noise sources} &< (N/4) - 1, & k \text{ even} \end{aligned}$$

Thus, the number of noise sources is upper bounded $(N/4) - 1$. Using this bound, we can get a more optimistic output noise variance.

$$\begin{aligned} \mathcal{E}[|F[k]|^2] &\leq (\frac{N}{4} - 1)\sigma_B^2 \\ &\lesssim \frac{N}{4}\sigma_B^2 \quad \text{for large } N \end{aligned}$$

When the scaling is done at the input, an upper bound on the noise-to-signal ratio is found to be

$$\frac{\mathcal{E}[|F[k]|^2]}{\mathcal{E}[|X[k]|^2]} \lesssim \frac{\frac{N}{4}\sigma_B^2}{\frac{1}{3N}} = \frac{3}{4}N^2\sigma_B^2 = \frac{N^2 2^{-2B}}{4}$$

This equation is similar to Eq. 9.65, but scaled by $\frac{1}{4}$. This implies that Eq. 9.65 is about 1 bit too pessimistic. However, note that the N^2 dependence is still present.

Another approach to this noise analysis is to compute the average noise at the output by using the average number of noisy butterflies connected to an output node. This style of analysis is used in Weinstein.

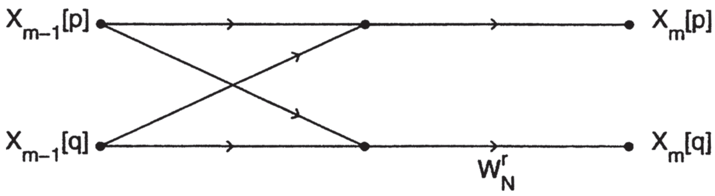
(c) Now assume as before that the first two stages are noiseless. Thus, equation 9.67 would not include the first two stages.

$$
\begin{aligned}
\mathcal{E}[|F[k]|^2] &\leq \sigma_B^2 \sum_{m=2}^{\nu-1} 2^{(\nu-m)} (\frac{1}{2})^{2\nu-2m-2} \\
&= \sigma_B^2 \sum_{m=2}^{\nu-1} (\frac{1}{2})^{\nu-m-2} \\
&= 2\sigma_B^2 \sum_{k=0}^{\nu-3} (\frac{1}{2})^k \\
&= 2\sigma_B^2 \left(\frac{1-(\frac{1}{2})^{\nu-2}}{1-\frac{1}{2}} \right) \\
&= 4\sigma_B^2 (1-(\frac{1}{2})^{\nu-2}) \\
&= 4\sigma_B^2 (1-\frac{4}{N})
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\frac{\mathcal{E}[|F[k]|^2]}{\mathcal{E}[|X[k]|^2]} &\leq 12N\sigma_B^2 \frac{N-4}{N} \\
&\lesssim 12(N-4)\sigma_B^2
\end{aligned}
$$

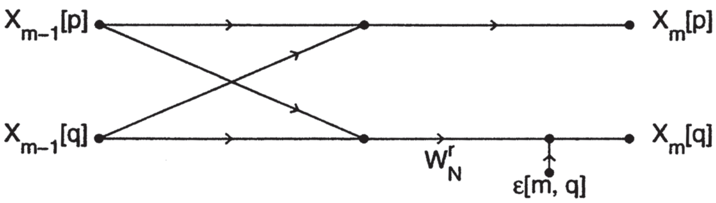
9.53. The butterfly for decimation-in-frequency is drawn below.



where

$$\begin{aligned} X_m[p] &= X_{m-1}[p] + X_{m-1}[q] \\ X_m[q] &= (X_{m-1}[p] - X_{m-1}[q])W_N^r \end{aligned}$$

The statistical model is drawn below.



As in a decimation-in-time FFT, each output node connects to $N-1$ butterflies in a decimation-in-frequency FFT, each of which introduces a noise source whose variance is

$$\mathcal{E}[|\epsilon[m,q]|^2] = \sigma_B^2 = \frac{1}{3}2^{-2B}$$

Thus the output noise propagated through the flowgraph is

$$\mathcal{E}[|F[k]|^2] = (N-1)\sigma_B^2$$

since each noise source propagates along a unity gain path to the output nodes.

Thus, the results for the decimation-in-frequency FFT are identical to those for the decimation-in-time FFT. This is true for both cases of scaling either at the input of the FFT by $1/N$, or at the input of each stage of the FFT by $1/2$.

9.54. Recall the following symmetry properties of the DFT:

$$\begin{array}{lcl} x[n] \text{ real} & \Longleftrightarrow & X[k] = X^*[N-k] \\ x[n] = x^*[N-n] & \Longleftrightarrow & \text{Im}\{X[k]\} = 0 \\ x[n] = -x^*[N-n] & \Longleftrightarrow & \text{Re}\{X[k]\} = 0 \end{array}$$

Thus, $x[n]$ real and even $\Longleftrightarrow$ $X[k]$ real and even; and $x[n]$ real and odd $\Longleftrightarrow$ $X[k]$ imaginary and odd.

(a)

$$y_1[n] = x_1[n] + x_3[n]$$

$x_1[n]$ is real and even; $x_3[n]$ is real and odd.

$$Y_1[k] = \text{Re}\{Y_1[k]\} + j\text{Im}\{Y_1[k]\} = X_1[k] + X_3[k]$$

$\text{Re}\{Y_1[k]\}$ is real and even; $j\text{Im}\{Y_1[k]\}$ is imaginary and odd; $X_1[k]$ is real and even; $X_3[k]$ is imaginary and odd. It follows that

$$\begin{array}{rcl} X_1[k] & = & \text{Re}\{Y_1[k]\} \\ X_3[k] & = & j\text{Im}\{Y_1[k]\} \end{array}$$

(b)

$$y_3[n] = y_1[n] + jy_2[n]$$

$$\begin{array}{rcl} Y_3[k] & = & Y_1[k] + jY_2[k] = [\text{Re}\{Y_1[k]\} + j\text{Im}\{Y_1[k]\}] + j[\text{Re}\{Y_2[k]\} + j\text{Im}\{Y_2[k]\}] \\ & = & \underbrace{\text{Re}\{Y_1[k]\}}_{\text{real}} - \underbrace{\text{Im}\{Y_2[k]\}}_{\text{odd}} + j[\underbrace{\text{Re}\{Y_2[k]\}}_{\text{even}} + \underbrace{\text{Im}\{Y_1[k]\}}_{\text{odd}}] \\ Y_3[k] & = & [\text{Ev}\{\text{Re}\{Y_3[k]\}\} + \text{Od}\{\text{Re}\{Y_3[k]\}\}] + j[\text{Ev}\{\text{Im}\{Y_3[k]\}\} + \text{Od}\{\text{Im}\{Y_3[k]\}\}] \end{array}$$

Thus we have

$$\begin{array}{rcl} \text{Re}\{Y_1[k]\} & = & \text{Ev}\{\text{Re}\{Y_3[k]\}\} \\ \text{Re}\{Y_2[k]\} & = & \text{Ev}\{\text{Im}\{Y_3[k]\}\} \\ \text{Im}\{Y_1[k]\} & = & \text{Od}\{\text{Im}\{Y_3[k]\}\} \\ \text{Im}\{Y_2[k]\} & = & -\text{Od}\{\text{Re}\{Y_3[k]\}\} \end{array}$$

and so

$$\begin{array}{rcl} Y_1[k] & = & \frac{1}{2}[\text{Re}\{Y_3[k]\} + \text{Re}\{Y_3[N-k]\}] \\ & & +\frac{j}{2}[\text{Im}\{Y_3[k]\} - \text{Im}\{Y_3[N-k]\}] \\ Y_2[k] & = & \frac{1}{2}[\text{Im}\{Y_3[k]\} + \text{Im}\{Y_3[N-k]\}] \\ & & -\frac{j}{2}[\text{Re}\{Y_3[k]\} - \text{Re}\{Y_3[N-k]\}] \end{array}$$

and from part (a)

$$\begin{aligned} X_1[k] &= \frac{1}{2}[\text{Re}\{Y_3[k]\} + \text{Re}\{Y_3[N-k]\}] \\ X_2[k] &= \frac{1}{2}[\text{Im}\{Y_3[k]\} + \text{Im}\{Y_3[N-k]\}] \\ X_3[k] &= \frac{j}{2}[\text{Im}\{Y_3[k]\} - \text{Im}\{Y_3[N-k]\}] \\ X_4[k] &= \frac{-j}{2}[\text{Re}\{Y_3[k]\} - \text{Re}\{Y_3[N-k]\}] \end{aligned}$$

(c)

$$u_3[n] = x_3[((n+1))_N] - x_3[((n-1))_N]$$

$$u_3[((N-n))_N] = x_3[((N-n+1))_N] - x_3[((N-n-1))_N]$$

Since $x_3[n] = x_3[((N-n))_N]$,

$$u_3[((N-n))_N] = x_3[((n-1))_N] - x_3[((n+1))_N] = -u_3[n]$$

For $n = 0$, we have $u_3[0] = -u_3[0]$, which is only satisfied for $u_3[0] = 0$.

(d) Using the circular shift property of the DFT we have

$$\begin{aligned} U_3[k] &= X_3[k]e^{j(2\pi/N)k} - X_3[k]e^{-j(2\pi/N)k} \\ &= 2j\sin(2\pi k/N)X_3[k] \end{aligned}$$

(e) From part (a), since $u_3[n]$ is antisymmetric,

$$X_1[k] = \text{Re}\{Y_1[k]\}$$

$$U_3[k] = j\text{Im}\{Y_1[k]\}$$

and so

$$X_3[k] = \frac{\text{Im}\{Y_1[k]\}}{2\sin\frac{2\pi k}{N}} \quad k \neq 0, \ \frac{N}{2}$$

Note that $X_3[0]$ cannot be recovered using this technique, and if N is even, neither can $X_3[N/2]$.

(f) In part (b) replace X_3 and X_4 with U_3 and U_4 and use the result of (d) to give

$$\begin{aligned} X_1[k] &= \frac{1}{2}[\text{Re}\{Y_3[k]\} + \text{Re}\{Y_3[N-k]\}] \\ X_2[k] &= \frac{1}{2}[\text{Im}\{Y_3[k]\} + \text{Im}\{Y_3[N-k]\}] \\ X_3[k] &= \frac{1}{4}[\text{Im}\{Y_3[k]\} - \text{Im}\{Y_3[N-k]\}]/\sin(2\pi k/N), \quad k \neq 0, N/2 \\ X_4[k] &= -\frac{1}{4}[\text{Re}\{Y_3[k]\} - \text{Re}\{Y_3[N-k]\}]/\sin(2\pi k/N), \quad k \neq 0, N/2 \end{aligned}$$

9.55. First, we find an expression for samples of the system function $H(z)$.

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{1 - \sum_{\ell=1}^{N} a_\ell z^{-\ell}}$$

$$H(e^{j2\pi k/N}) = \frac{\sum_{r=0}^{M} b_r e^{-j2\pi kr/N}}{1 - \sum_{\ell=1}^{N} a_\ell e^{-j2\pi k\ell/N}}$$

Now assume $N,\ M \leq 511$. Let $b[n] = b_n$ and

$$a[n] = \begin{cases} 1, & n = 0 \\ a_n, & 1 \leq n \leq N \end{cases}$$

Let $B[k]$, $A[k]$ be the 512 pt DFTs of $b[n]$, and $a[n]$. Then

$$H(e^{j2\pi k/512}) = \frac{B[k]}{A[k]}$$

9.56. (a) It is interesting to note that (linear) convolution and polynomial multiplication are the same operation. Many mathematical software tools, like Matlab, perform polynomial multiplication using convolution. Here, we replace

$$p(x) = \sum_{i=0}^{L-1} a_i x^i, \qquad q(x) = \sum_{i=0}^{M-1} b_i x^i$$

with

$$p[n] = \sum_{i=0}^{L-1} a_i \delta[n-i], \qquad q[n] = \sum_{i=0}^{M-1} b_i \delta[n-i]$$

Then,

$$r[n] = p[n] * q[n].$$

The coefficients in $r[n]$ will be identically equal to those of $r(x)$. We can compute $r[n]$ with circular convolution, instead of linear convolution, by zero padding $p[n]$ and $q[n]$ to a length $N = L+M-1$. This zero padding ensures that linear convolution and circular convolution will give the same result.

(b) We can implement the circular convolution of $p[n]$ and $q[n]$ using the following procedure.

step 1: Take the DFTs of $p[n]$ and $q[n]$ using the FFT program. This gives $P[k]$ and $Q[k]$.
step 2: Multiply to get $R[k] = P[k]Q[k]$.
step 3: Take the inverse DFT of $R[k]$ using the FFT program. This gives $r[n]$.

Here, we assumed that the FFT program also computes inverse DFTs. If not, it is a relatively simple matter to modify the input to the program so that its output is an inverse DFT. (See problem 9.1).

While it may seem that this procedure is more work, for long sequences, it is actually more efficient. The direct computation of $r[n]$ requires approximately $(L+M)^2$ real multiplications, since a_i and b_i are real. Assuming that a length $L+M$ FFT computation takes $[(L+M)/2]\log_2(L+M)$ complex multiplications, we count the complex multiplications required in the procedure described above to be

Operation	Complex Multiplications
FFTs of $p[n]$ and $q[n]$	$2[(L+M)/2]\log_2(L+M) = (L+M)\log_2(L+M)$
$R[k] = P[k]Q[k]$	$L+M$
Inverse FFT of $R[k]$	$[(L+M)/2]\log_2(L+M)$

$$= [3(L+M)/2]\log_2(L+M) + (L+M)$$

Since a complex multiplication is computed using 4 real multiplications, the number of real multiplications required by this technique is $6(L+M)\log_2(L+M) + 4(L+M)$. Plugging in some values for $(L+M) = 2^\nu$, we find

$L+M$	Direct	FFT
2	4	20
4	16	64
8	64	176
16	256	448
32	1024	1088
64	4096	2560

Thus, for $(L+M) \geq 64$, the FFT approach is more efficient.

(c) The binary integers u and v have corresponding decimal values, which are

$$u_{\text{decimal}} = \sum_{i=0}^{L-1} u_i 2^i$$

$$v_{\text{decimal}} = \sum_{i=0}^{M-1} v_i 2^i$$

Note the resemblance to $p(x)$ and $q(x)$ of part (a). We form the signals

$$u[n] = \sum_{i=0}^{L-1} u_i \delta[n-i]$$

$$v[n] = \sum_{i=0}^{M-1} v_i \delta[n-i]$$

and use the procedure described in part (b). This computes the product $u \cdot v$ in binary. For $L = 8000$ and $M = 1000$, this procedure requires approximately

$$\begin{aligned}\#\text{ real multiplications} &= 6(8000+1000)\log_2(8000+1000) + 4(8000+1000) \\ &= 7.45 \times 10^5\end{aligned}$$

In contrast, the direct computation requries 8.1×10^7 real multiplications.

(d) For the (forward and inverse) FFTs, the mean-square value of the output noise is $(L+M)\sigma_B^2$. While σ_B^2 will be small, as there are 16 bits, the noise can be significant, since $L+M$ is a large number.

9.57. (a) Using the definition of the discrete Hartley transform we get

$$\begin{aligned} H_N[a+N] &= C_N[a+N] + S_N[a+N] \\ &= \cos(2\pi a/N + 2\pi) + \sin(2\pi a/N + 2\pi) \\ &= \cos(2\pi a/N) + \sin(2\pi a/N) \\ &= C_N[a] + S_N[a] \\ &= H_N[a] \end{aligned}$$

$$\begin{aligned} H_N[a+b] &= C_N[a+b] + S_N[a+b] \\ &= \cos(2\pi a/N + 2\pi b/N) + \sin(2\pi a/N + 2\pi b/N) \\ &= \cos(2\pi a/N)\cos(2\pi b/N) - \sin(2\pi a/N)\sin(2\pi b/N) \\ &\quad + \sin(2\pi a/N)\cos(2\pi b/N) + \cos(2\pi a/N)\sin(2\pi b/N) \end{aligned}$$

Grouping the terms in the last equation one way gives us

$$\begin{aligned} H_N[a+b] &= [\cos(2\pi a/N) + \sin(2\pi a/N)]\cos(2\pi b/N) \\ &\quad + [\cos(-2\pi a/N) + \sin(-2\pi a/N)]\sin(2\pi b/N) \\ &= H_N[a]C_N[b] + H_N[-a]S_N[b] \end{aligned}$$

while grouping the terms another way gives us

$$\begin{aligned} H_N[a+b] &= [\cos(2\pi b/N) + \sin(2\pi b/N)]\cos(2\pi a/N) \\ &\quad + [\cos(-2\pi b/N) + \sin(-2\pi b/N)]\sin(2\pi a/N) \\ &= H_N[b]C_N[a] + H_N[-b]S_N[a] \end{aligned}$$

(b) To obtain a fast algorithm for computation of the discrete Hartley transform, we can proceed as in the decimation-in-time FFT algorithm; i.e.,

$$\begin{aligned} X_H[k] &= \sum_{r=0}^{(N/2)-1} x[2r]H_N[2rk] + \sum_{r=0}^{(N/2)-1} x[2r+1]H_N[(2r+1)k] \\ &= \sum_{r=0}^{(N/2)-1} x[2r]H_N[2rk] + \sum_{r=0}^{(N/2)-1} x[2r+1]H_N[2rk]C_N[k] \\ &\quad + \sum_{r=0}^{(N/2)-1} x[2r+1]H_N[((-2rk))_N]S_N[k] \end{aligned}$$

Now since $H_N[2rk] = H_{N/2}[rk]$, we have

$$\begin{aligned} X_H[k] &= \sum_{r=0}^{(N/2)-1} x[2r]H_{N/2}[rk] + \sum_{r=0}^{(N/2)-1} x[2r+1]H_{N/2}[rk]C_N[k] \\ &\quad + \sum_{r=0}^{(N/2)-1} x[2r+1]H_{N/2}[((-rk))_{N/2}]S_N[k] \\ &= F[k] + G[k]C_N[k] + G[((-k))_{N/2}]S_N[k] \end{aligned}$$

where

$$F[k] = \sum_{r=0}^{(N/2)-1} x[2r]H_{N/2}[rk]$$

is the $N/2$-point DHT of the even-indexed points and

$$G[k] = \sum_{r=0}^{(N/2)-1} x[2r+1]H_{N/2}[rk]$$

is the $N/2$-point DHT of the odd-indexed points. As in the derivation of the decimation-in-time FFT algorithm, we can continue to divide the sequences in half if N is a power of 2. Thus the indexing will be exactly the same except that we have to access $G[((-k))_{N/2}]$ as well as $G[k]$ and $F[k]$; i.e., the "butterfly" is slightly more complicated. The fast Hartley transform will require $N \log_2 N$ operations as in the case of the DFT, but the multiplies and adds will be real instead of complex.

9.58. (a)

$$\mathbf{F}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad \mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_8^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_8^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_8^0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_8^2 \end{bmatrix}$$

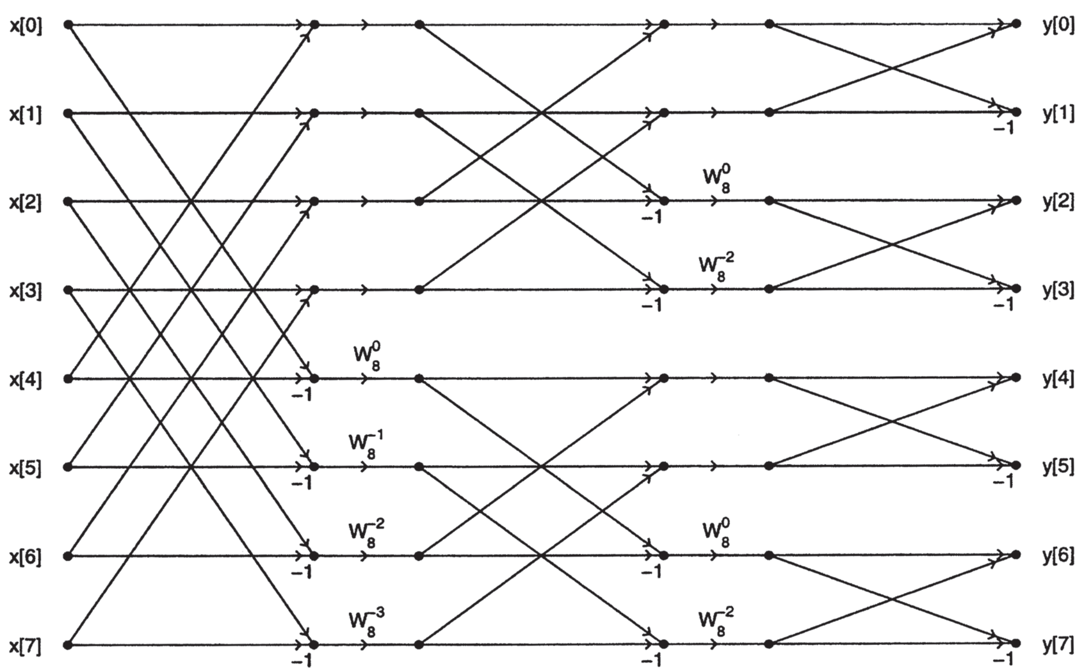
$$\mathbf{F}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_8^0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_8^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_8^3 \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

(b)

$$\begin{aligned}\mathbf{Q}^H &= \mathbf{F}_1^H\mathbf{T}_1^H\mathbf{F}_2^H\mathbf{T}_2^H\mathbf{F}_3^H \\ &= \mathbf{F}_1\mathbf{T}_1^*\mathbf{F}_2\mathbf{T}_2^*\mathbf{F}_3\end{aligned}$$

where $*$ denotes conjugation. Drawing the flow graph, we get



This structure is the decimation in frequency FFT with the twiddle factors conjugated and therefore calculates

$$N \cdot \text{IDFT}\{x[n]\}$$

(c) Knowing that $\mathbf{Q}$ calculates the DFT and $\frac{1}{N}\mathbf{Q}^H$ calculates the IDFT, we should realize that cascading the two should just return the original signal. More formally we have

$$\mathbf{F}_1^H\mathbf{F}_1 = 2\mathbf{I} \quad \mathbf{F}_2^H\mathbf{F}_2 = 2\mathbf{I} \quad \mathbf{F}_3^H\mathbf{F}_3 = 2\mathbf{I}$$

$$\mathbf{T}_1^H\mathbf{T}_1 = \mathbf{I} \quad \mathbf{T}_2^H\mathbf{T}_2 = \mathbf{I}$$

$$\begin{aligned}(1/N)\mathbf{Q}^H\mathbf{Q} &= (1/N)\left(\mathbf{F}_1^H\mathbf{T}_1^H\mathbf{F}_2^H\mathbf{T}_2^H\mathbf{F}_3^H\right)\left(\mathbf{F}_3\mathbf{T}_2\mathbf{F}_2\mathbf{T}_1\mathbf{F}_1\right) \\ &= (1/N)(N\mathbf{I}) \\ &= \mathbf{I}\end{aligned}$$

where $N = 8$ in this case.

9.59. (a) First, we derive the circular convolution property of the DFT. We start with the circular convolution of $x[n]$ and $h[n]$.

$$y[n] = \sum_{m=0}^{N-1} x[m]h[((n-m))_N]$$

Taking the DFT of both sides gives

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x[m]h[((n-m))_N]W^{nk} \\ &= \sum_{m=0}^{N-1} x[m]\sum_{n=0}^{N-1} h[((n-m))_N]W^{nk} \end{aligned}$$

Using the circular shift property of the DFT,

$$\begin{aligned} Y[k] &= \sum_{m=0}^{N-1} x[m]H[k]W_N^{mk} \\ &= H[k]\sum_{m=0}^{N-1} x[m]W_N^{mk} \\ &= H[k]X[k] \end{aligned}$$

Next, the orthogonality of the basis vectors is shown to be a necessary requirement for the circular convolution property. We start again with the circular convolution of $x[n]$ and $h[n]$

$$y[n] = \sum_{m=0}^{N-1} x[m]h[((n-m))_N]$$

Substituting the inverse DFT for $x[m]$ and $h[((n-m))_N]$ gives

$$\begin{aligned} y[n] &= \sum_{m=0}^{N-1}\left(\frac{1}{N}\sum_{k_1=0}^{N-1} X[k_1]W_N^{-k_1 m}\right)\left(\frac{1}{N}\sum_{k_2=0}^{N-1} H[k_2]W_N^{-k_2((n-m))_N}\right) \\ &= \sum_{m=0}^{N-1}\left(\frac{1}{N}\sum_{k_1=0}^{N-1} X[k_1]W_N^{-k_1 m}\right)\left(\frac{1}{N}\sum_{k_2=0}^{N-1} H[k_2]W_N^{-k_2(n-m)}\right) \\ &= \frac{1}{N^2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\left(X[k_1]H[k_2]W_N^{-k_2 n}\sum_{m=0}^{N-1} W_N^{-k_1 m}W_N^{k_2 m}\right) \end{aligned}$$

With orthogonal basis vectors,

$$\sum_{n=0}^{N-1} W_N^{nk} = \begin{cases} N, & k=0 \\ 0, & k\neq 0 \end{cases}$$

so the right-most summation becomes

$$\begin{aligned} \sum_{m=0}^{N-1} W_N^{-k_1 m}W_N^{k_2 m} &= \sum_{m=0}^{N-1} W_N^{(k_2-k_1)m} \\ &= N\delta[k_1-k_2] \end{aligned}$$

Therefore,

$$\begin{aligned} y[n] &= \frac{1}{N^2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\left(X[k_1]H[k_2]W_N^{-k_2 n}N\delta[k_1-k_2]\right) \\ &= \frac{1}{N}\sum_{k_1=0}^{N-1} X[k_1]H[k_1]W_N^{-k_1 n} \\ &= \sum_{m=0}^{N-1} x[m]h[((n-m))_N] \end{aligned}$$

Therefore, the circular convolution property holds as long as the basis vectors are orthogonal.

(b)

$$\left(\left(\sum_{n=0}^{3} 4^{nk}\right)\right)_{17} = ((1 + 4^k + 16^k + 64^k))_{17}$$

$$\begin{aligned}
k &= 0: & ((1+1+1+1))_{17} &= ((4))_{17} &= 4 \\
k &= 1: & ((1+4+16+64))_{17} &= ((85))_{17} &= 0 \\
k &= 2: & ((1+16+256+4096))_{17} &= ((4369))_{17} &= 0 \\
k &= 3: & ((1+64+4096+262144))_{17} &= ((266305))_{17} &= 0
\end{aligned}$$

The relation holds for $P = 17$, $N = 4$, and $W_N = 4$.

(c)

$$\begin{aligned}
X[k] &= \left(\left(\sum_{n=0}^{3} x[n]4^{nk}\right)\right)_{17} \\
&= ((1 \cdot 1 + 2 \cdot 4^k + 3 \cdot 16^k + 0 \cdot 64^k))_{17} \\
&= ((1 + 2 \cdot 4^k + 3 \cdot 16^k))_{17}
\end{aligned}$$

Using this formula for $X[k]$,

$$\begin{aligned}
X[0] &= ((1+2+3))_{17} &= ((6))_{17} &= 6 \\
X[1] &= ((1+2\cdot 4+3\cdot 16))_{17} &= ((57))_{17} &= 6 \\
X[2] &= ((1+2\cdot 16+3\cdot 256))_{17} &= ((801))_{17} &= 2 \\
X[3] &= ((1+2\cdot 64+3\cdot 4096))_{17} &= ((12417))_{17} &= 7
\end{aligned}$$

$$\begin{aligned}
H[k] &= \left(\left(\sum_{n=0}^{3} h[n]4^{nk}\right)\right)_{17} \\
&= ((3 \cdot 1 + 1 \cdot 4^k + 0 \cdot 16^k + 0 \cdot 64^k))_{17} \\
&= ((3 + 4^k))_{17}
\end{aligned}$$

Using this formula for $H[k]$,

$$\begin{aligned}
H[0] &= ((3+1))_{17} &= ((4))_{17} &= 4 \\
H[1] &= ((3+4))_{17} &= ((7))_{17} &= 7 \\
H[2] &= ((3+16))_{17} &= ((19))_{17} &= 2 \\
H[3] &= ((3+64))_{17} &= ((67))_{17} &= 16
\end{aligned}$$

Multiplying terms $Y[k] = X[k]H[k]$ gives

$$\begin{aligned}
Y[0] &= ((24))_{17} &= 7 \\
Y[1] &= ((42))_{17} &= 8 \\
Y[2] &= ((4))_{17} &= 4 \\
Y[3] &= ((112))_{17} &= 10
\end{aligned}$$

(d) By trying out different values,

$$\begin{aligned} N^{-1} &= 13 \\ W_N^{-1} &= 13 \end{aligned}$$

$$((N^{-1}N))_{17} = ((W_N^{-1}W_N))_{17} = ((13 \cdot 4))_{17} = ((52))_{17} = 1$$

$$\begin{aligned} y[n] &= \left(\left(13\sum_{k=0}^{3} Y[k]13^{nk}\right)\right)_{17} \\ &= ((13[7 \cdot 1 + 8 \cdot 13^n + 4 \cdot 169^n + 10 \cdot 2197^n]))_{17} \end{aligned}$$

Using this formula for $y[n]$,

$$\begin{aligned} y[0] &= ((13[7+8+4+10]))_{17} &&= ((377))_{17} &&= 3 \\ y[1] &= ((13[7 \cdot 1 + 8 \cdot 13 + 4 \cdot 169 + 10 \cdot 2197]))_{17} &&= ((295841))_{17} &&= 7 \\ y[2] &= ((13[7 \cdot 1 + 8 \cdot 169 + 4 \cdot 28561 + 10 \cdot 4826809]))_{17} &&= ((628988009))_{17} &&= 11 \\ y[3] &= ((13[7 \cdot 1 + 8 \cdot 2197 + 4 \cdot 4826809 + 10 \cdot 10604499373]))_{17} &&= ((1378836141137))_{17} &&= 3 \end{aligned}$$

Performing manual convolution $y[n] = x[n] * h[n]$ gives

$$\begin{aligned} y[0] &= 3 \\ y[1] &= 7 \\ y[2] &= 11 \\ y[3] &= 3 \end{aligned}$$

The results agree.

9.60. (a) The tables below list the values for n and k obtained with the index maps.

$n_1 \backslash n_2$	0	1	2
0	0	1	2
1	3	4	5

$k_1 \backslash k_2$	0	1	2
0	0	2	4
1	1	3	5

As shown, the index maps only produce $n = 0, \ldots, 5$ and $k = 0, \ldots, 5$.

(b) Making the substitution we get

$$\begin{aligned} X[k] &= X[k_1 + 2k_2] \\ &= \sum_{n=0}^{5} x[n] W_6^{(k_1+2k_2)n} \\ &= \sum_{n_2=0}^{2} \sum_{n_1=0}^{1} x[3n_1 + n_2] W_6^{(k_1+2k_2)(3n_1+n_2)} \end{aligned}$$

(c) Expanding out the W_6 terms we get

$$\begin{aligned} W_6^{(k_1+2k_2)(3n_1+n_2)} &= W_6^{3k_1n_1} W_6^{6k_2n_1} W_6^{k_1n_2} W_6^{2k_2n_2} \\ &= W_2^{k_1n_1} W_6^{k_1n_2} W_3^{k_2n_2} \end{aligned}$$

(d) Grouping the terms we get

$$X[k_1 + 2k_2] = \sum_{n_2=0}^{2} \left[\left(\sum_{n_1=0}^{1} x[3n_1 + n_2] W_2^{k_1n_1} \right) W_6^{k_1n_2} \right] W_3^{k_2n_2}$$

The interpretation of this equation is as follows

(i) Let $G[k_1, n_2]$ be the $N = 2$ point DFTs of the inner parenthesis; i.e.,

$$G[k_1, n_2] = \sum_{n_1=0}^{1} x[3n_1 + n_2] W_2^{k_1n_1}, \qquad \begin{cases} 0 \le k_1 \le 1, \\ 0 \le n_2 \le 2. \end{cases}$$

This calculates 3 DFTs, one for each column of the index map associated with n. Since the DFT size is 2, we can perform these with simple butterflies and use no multiplications.

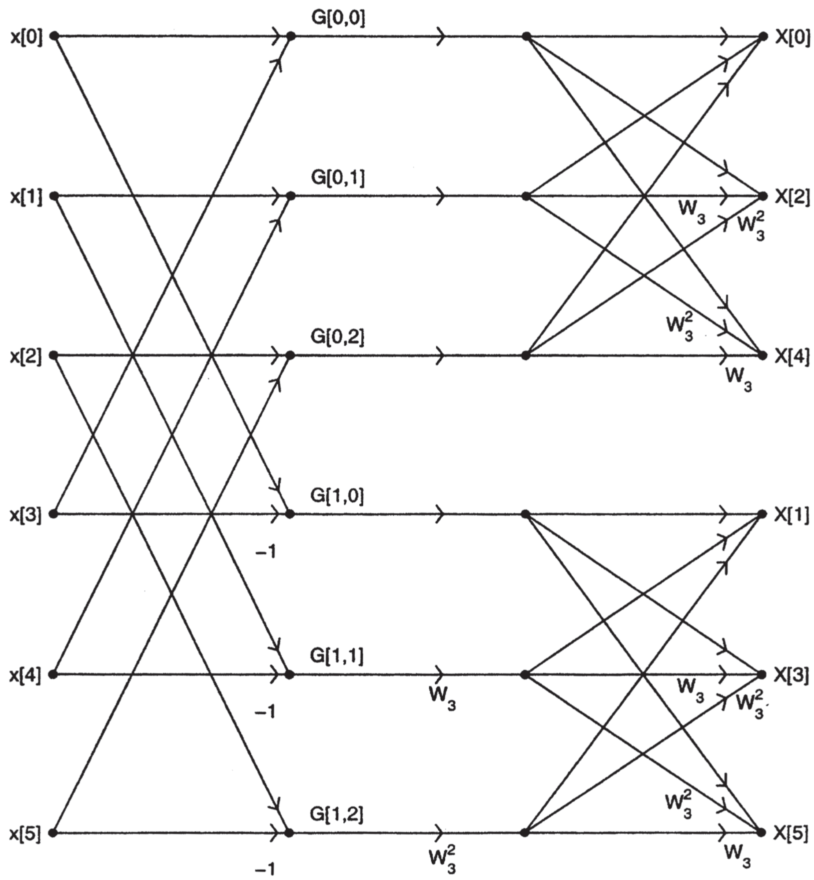
(ii) Let $\tilde{G}[k_1, n_2]$ be the set of 3 column DFTs multiplied by the twiddle factors.

$$\tilde{G}[k_1, n_2] = W_6^{k_1n_2} G[k_1, n_2], \qquad \begin{cases} 0 \le k_1 \le 1, \\ 0 \le n_2 \le 2. \end{cases}$$

(iii) The outer sum calculates two $N = 3$ point DFTs, one for each of the two values of k_1.

$$X[k_1 + 2k_2] = \sum_{n_2=0}^{2} \tilde{G}[k_1, n_2] W_3^{k_2n_2}, \qquad \begin{cases} 0 \le k_1 \le 1, \\ 0 \le k_2 \le 2. \end{cases}$$

(e) The signal flow graph looks like



The only complex multiplies are due to the twiddle factors. Therefore, there are 10 complex multiplies. The direct implementation requires $N^2 = 6^2 = 36$ complex multiplies (a little less if you do not count multiplies by 1 or -1).

(f) The alternate index map can be found be reversing the roles of n and k; i.e.,

$$n = n_1 + 2n_2 \quad \text{for } n_1 = 0, 1;\ n_2 = 0, 1, 2$$
$$k = 3k_1 + k_2 \quad \text{for } k_1 = 0, 1;\ k_2 = 0, 1, 2$$