9.27. 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.

9.30. (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

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

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

step 1: Form the sequence

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

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 $X[k] = Y[(k-1)/2]$, $k = 1, 3, \ldots, N-1$.

9.31. (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.

9.32. (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

4.42. (a) The Nyquist criterion states that $x_c(t)$ can be recovered as long as

$$\frac{2\pi}{T} \geq 2 \times 2\pi(250) \Longrightarrow T \leq \frac{1}{500}.$$

In this case, $T = 1/500$, so the Nyquist criterion is satisfied, and $x_c(t)$ can be recovered.

(b) Yes. A delay in time does not change the bandwidth of the signal. Hence, $y_c(t)$ has the same bandwidth and same Nyquist sampling rate as $x_c(t)$.

(c) Consider first the following expressions for $X(e^{j\omega})$ and $Y(e^{j\omega})$:

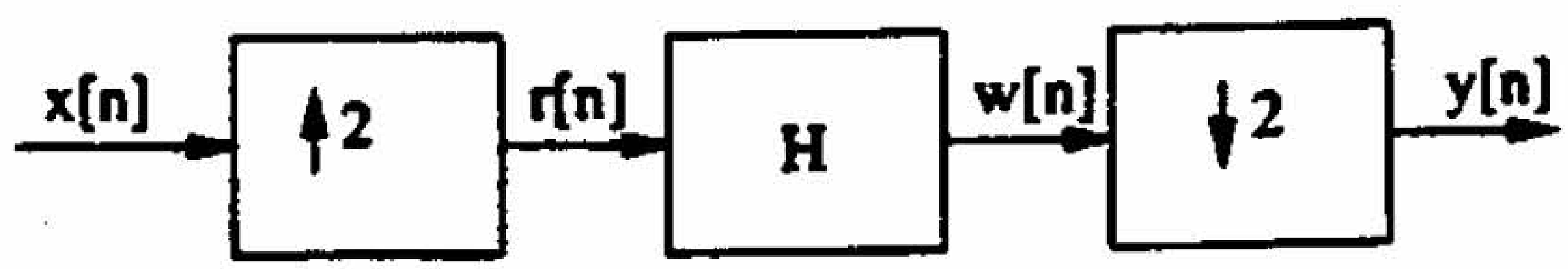
$$\begin{aligned}
X(e^{j\omega}) &= \frac{1}{T}X_c(j\Omega)\,|_{\Omega=\frac{\omega}{T}} = \frac{1}{500}X_c(j500\omega) \\
Y(e^{j\omega}) &= \frac{1}{T}Y_c(j\Omega)\,|_{\Omega=\frac{\omega}{T}} = \frac{1}{T}e^{-j\Omega/1000}X_c(j\Omega)\,|_{\Omega=\frac{\omega}{T}} \\
&= \frac{1}{500}e^{-j\omega/2}X_c(j500\omega) \\
&= e^{-j\omega/2}X(e^{j\omega})
\end{aligned}$$

Hence, we let

$$H(e^{j\omega}) = \begin{cases} 2e^{-j\omega}, & |\omega| < \frac{\pi}{2} \\ 0, & \text{otherwise} \end{cases}$$

Then, in the following figure,

$$\begin{aligned}
R(e^{j\omega}) &= X(e^{j2\omega}) \\
W(e^{j\omega}) &= \begin{cases} 2e^{-j\omega}X(e^{j2\omega}), & |\omega| < \frac{\pi}{2} \\ 0, & \text{otherwise} \end{cases} \\
Y(e^{j\omega}) &= e^{-j\omega/2}X(e^{j\omega})
\end{aligned}$$



(d) Yes, from our analysis above,

$$H_2(e^{j\omega}) = e^{-j\omega/2}$$

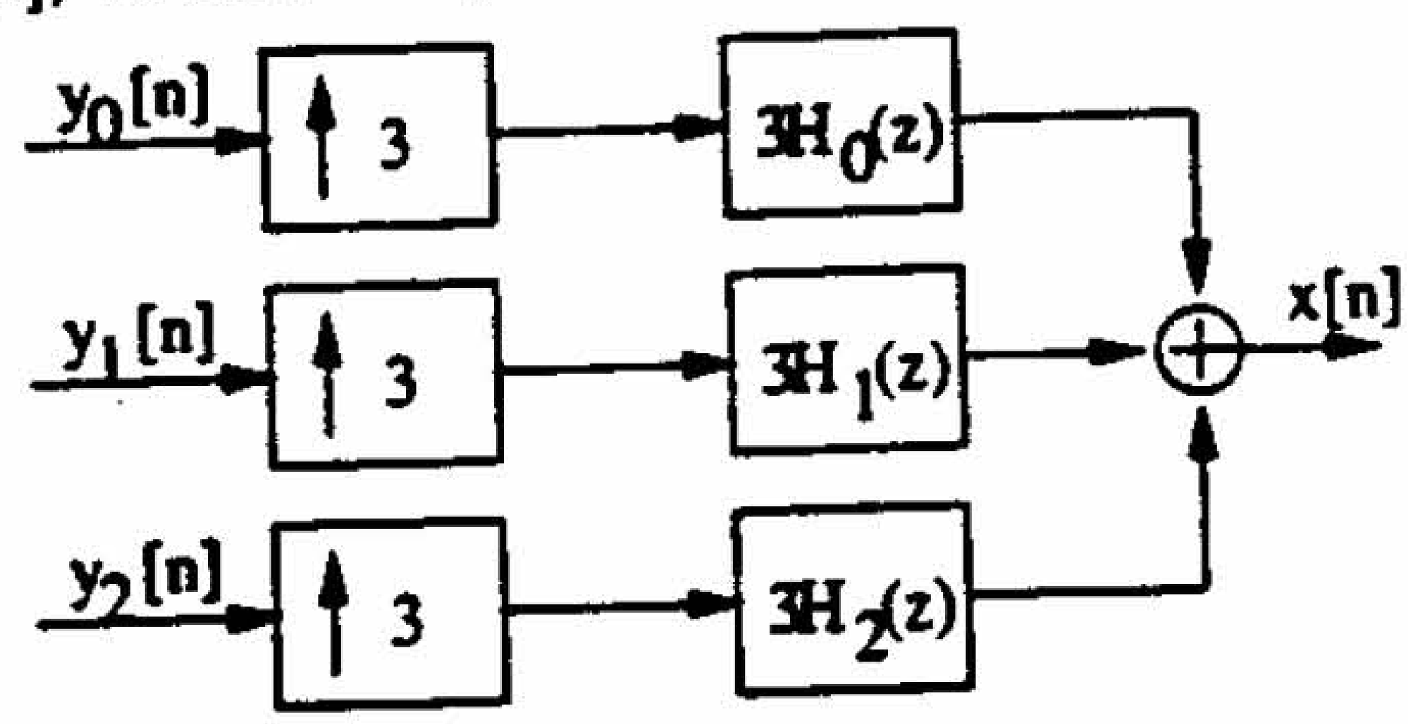
4.46. (a) Notice that

$$\begin{aligned} y_0[n] &= x[3n] \\ y_1[n] &= x[3n+1] \\ y_2[n] &= x[3n+2], \end{aligned}$$

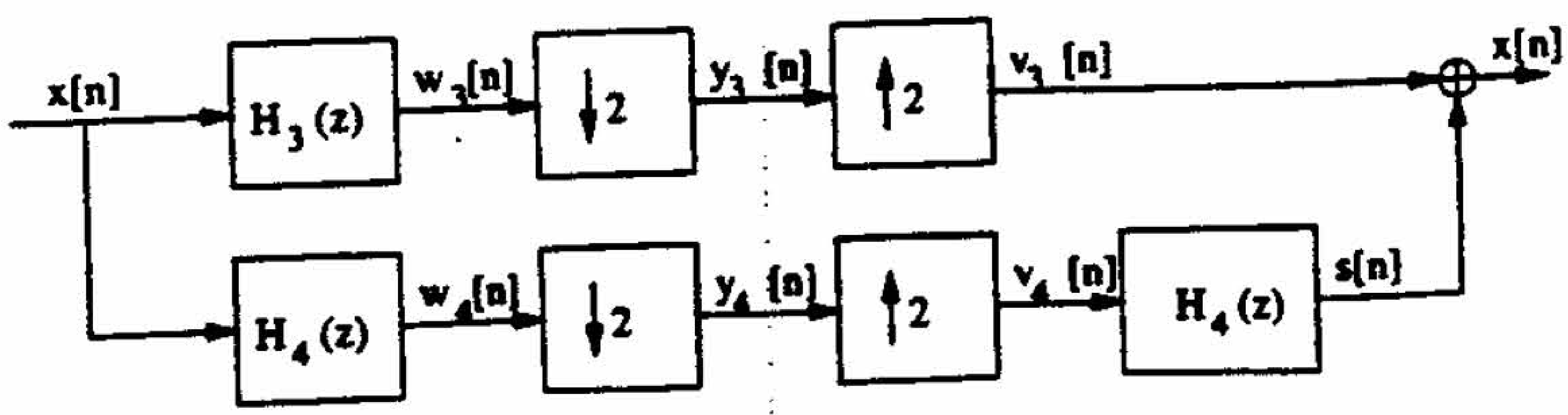
and therefore,

$$x[n] = \begin{cases} y_0[n/3], & n = 3k \\ y_1[(n-1)/3], & n = 3k+1 \\ y_2[(n-2)/3], & n = 3k+2 \end{cases}$$

(b) Yes. Since the bandwidth of the filters are $2\pi/3$, there is no aliasing introduced by downsampling. Hence to reconstruct $x[n]$, we need the system shown in the following figure:



(c) Yes, $x[n]$ can be reconstructed from $y_3[n]$ and $y_4[n]$ as demonstrated by the following figure:



In the following discussion, let $x_e[n]$ denote the even samples of $x[n]$, and $x_o[n]$ denote the odd samples of $x[n]$:

$$\begin{aligned} x_e[n] &= \begin{cases} x[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \\ x_o[n] &= \begin{cases} 0, & n \text{ even} \\ x[n], & n \text{ odd} \end{cases} \end{aligned}$$

In the figure, $y_3[n] = x[2n]$, and hence,

$$\begin{aligned} v_3[n] &= \begin{cases} x[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \\ &= x_e[n] \end{aligned}$$

Furthermore, it can be verified using the IDFT that the impulse response $h_4[n]$ corresponding to $H_4(e^{j\omega})$ is

$$h_4[n] = \begin{cases} -2/(j\pi n), & n \text{ odd} \\ 0, & \text{otherwise} \end{cases}$$

Notice in particular that every other sample of the impulse response $h_4[n]$ is zero. Also, from the form of $H_4(e^{j\omega})$, it is clear that $H_4(e^{j\omega})H_4(e^{j\omega}) = 1$ and hence $h_4[n] * h_4[n] = \delta[n]$. Therefore,

$$\begin{aligned} v_4[n] &= \begin{cases} y_4[n/2], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \\ &= \begin{cases} w_4[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \\ &= \begin{cases} (x * h_4)[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \\ &= x_o[n] * h_4[n] \end{aligned}$$

where the last equality follows from the fact that $h_4[n]$ is non-zero only in the odd samples. Now, $s[n] = v_4[n] * h_4[n] = x_o[n] * h_4[n] * h_4[n] = x_o[n]$. and since $x[n] = x_e[n] + x_o[n]$, $s[n] + v_3[n] = x[n]$.