Student: Marco Defilippi

ES1:

4.1 Thinking about the FFT:

(a) How do the results differ between performing an N-point FFT and performing an N-point discrete Fourier transform (DFT) on the same set of time samples?

(b) What is the restriction on the number of time samples, N, in performing an N-point radix-2 FFT?

Answer:

The FFT and the DFT are mathematically equivalent.

a) Both transform a sequence of N time-domain samples into an N-point frequency-domain representation.

There are some considerations about the complexity. DFT has a complexity $O(N^2)$, therefore becomes computationally expensive for large N. The FFT uses divide and conquer strategies and is implemented with specific algorithms to achieve faster computation times. This makes the FFT practical for large datasets and real time signal processing applications.

b) The radix-2 FFT algorithm is one of the most common and efficient implementations of the FFT. It has specific requirements regarding the number of time samples, N.

   The number of time samples N must be a power of two. This means N can be expressed as $N=2^k$ where k is any integer.

   The radix-2 FFT algorithm divides the problem into smaller sub-problems, each of size N/2. This recursive division is efficient when N is a power of two because it ensures that the problem can be continuously divided until the base case of size 1 is reached.

   If N is not a power of two, the algorithm cannot evenly divide the sequence, leading to inefficiencies and complicating the implementation.

ES2:

4.2 Assume we want to compute an N-point FFT of an $x(n)$ audio signal from a compact disc (CD), with the FFT's output frequency-domain sample spacing no greater than 1 Hz. If $x(n)$'s sample rate is f_s = 44.1 kHz, what is the number of necessary time samples, N, applied to the FFT?

Answer:

To determine the number of necessary time samples N to achieve a frequency-domain sample spacing no greater than 1 Hz, it is possible by using the following formula:

Frequency resolution:

Delta_f = fs/N, where fs is the sampling rate and N is the number of time samples.

Delta_f <= fs/N <= 1Hz

N>= fs/delta_f = 44100 Hz/1 Hz = 44100.

Therefore, the number of necessary time samples N applied to the FFT must be at least 44100 to achieve a frequency-domain sample spacing of no greater than 1 Hz.

ES3:

4.3 Assume we have an $x(n)$ time-domain sequence, whose length is 3800 samples, on which we want to perform an FFT. The 3800 time samples represent a total signal collection-interval duration of 2 seconds.

(a) How many zero-valued samples must be appended (*zero padding*) to $x(n)$ in order to implement an FFT?

(b) After the FFT is performed, what is the spacing, measured in Hz, between the frequency-domain FFT samples?

(c) In the case of lowpass sampling, what is the highest-frequency spectral component permitted in the original analog $x(t)$ signal such that no aliasing errors occur in $x(n)$?

Answer:

a) To perform an FFT efficiently, it is often desirable to have the number of points be a power of 2. Given the length of the sequence x(n) is 3800 samples, it is needed to determine the smallest power of 2 that is greater than or equal to 3800.

The powers of 2 around 3800 are: 2

2^{11} = 2048

2^{12} = 4096

Since 2048 is less than 3800, the smallest power of 2 that is greater than 3800 is 4096.

The number of zero-valued samples to append is : 4096 – 3800 = 296

b)

The FFT frequency resolution is determined by the sampling frequency fs and the number of points N used in the FFT.

The frequency resolution delta_f is:

Delta_f = fs/N

Since the total time duration is given by N*Ts = 2 seconds:

Ts=2/N = 2/3800 = 1/1900 s

Fs=1/Ts = 1900 Hz.

Since the number of points for the FFT is now N=4096:

Delta_f=fs/N = 1900/4096 = 0.463 Hz.

Thus, the spacing between the frequency-domain FFT samples is approximately 0.463.

c)

To avoid aliasing, the sampling frequency must be at least twice the highest frequency component in the original signal, according to the Nyquist theorem.

The highest frequency fmax that can be accurately represented without aliasing is given by:

Fmax= fs/2

The sampling frequency fs is determined by the number of samples collected over the total duration. Given 3800 samples over 2 seconds, the sampling frequency is:

Fs=3800/2 = 1900Hz

Therefore, the highest-frequency spectral component permitted to avoid aliasing is:

Fmax= fs/2 = 1900/2 = 950 Hz.

ES4:

4.4 This problem illustrates the computational savings afforded by the FFT over that of the discrete Fourier transform (DFT). Suppose we wanted to perform a spectrum analysis on a time-domain sequence whose length is 32768 (2^{15}) samples. Estimate the ratio of the number of complex multiplications needed by a 32768-point DFT over the number of complex multiplies needed by a 32768-point FFT. (Assume that one of the text's *optimized* Figure 4–14(c) butterflies, requiring one complex multiply per butterfly operation, is used to implement the FFT.)

Answer:

The DFT requires N^2 complex multiplications for an N point sequence.

The FFT is an optimized version of the DFT and significantly reduces the number of operations. For an N point sequence, the FFT requires approximately (N/2) $\log_2(N)$ complex multiplications.

If N= 32768 = 2^{15}, means that:

- for DFT, N^2 = N^{30} = 1073741824 operations.
- for FFT, (N/2)$\log_2(N)$ = (32768/2) log2(32768) = 245760 operations.

The ratio of the number of complex multiplications needed by the DFT to the number needed by the FFT is:

Ratio = DFT multiplications/FFT multiplications = 1073741824 / 245760 = 4369.92

So, the number of complex multiplications needed by a 32768 point DFT is approximately 4369.92 times greater than that needed by a 32768 point FFT.

ES5:

4.5 Think about the system in Figure P4–5 using an FFT to measure the amplitude of the $p(t)$ signal. The output of the mixer, the product $p(t)q(t)$, contains the sum of two sinusoids whose amplitudes are proportional to the peak value of $p(t)$. The frequencies of those sinusoids are 50 Hz and 2050 Hz. The lowpass filter rejects the 2050 Hz signal. Due to imperfections in the mixer, signal $p(t)q(t)$ is riding on a constant DC (zero Hz) bias represented as value D. This scenario results in an $x(n)$ time sequence whose average value is 17.

(a) What is the minimum value for the analog-to-digital converter's f_s sample rate to satisfy the Nyquist criterion?

(b) If we collect 2048 filter output samples and perform a 2048-point FFT, what will be the magnitude of the FFT's $X(0)$ sample?

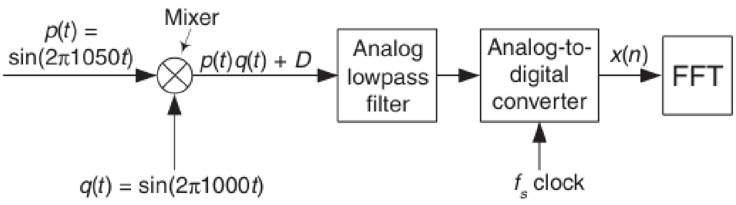


Figure P4–5

Answer:

a)

To determine the minimum sampling rate fs for the analog-to-digital converter to satisfy the Nyquist criterion, it is needed to consider the highest frequency component in the signal after the lowpass filter.

The mixer output p(t)xq(t) contains two frequencies:

- 50 Hz (from $\sin(2\pi 1050t)\cdot\sin(2\pi 1000t)$)
- 2050 Hz

The lowpass filter rejects the 2050 Hz component, leaving only the 50 Hz component and a DC bias D.

According to the Nyquist criterion, the sampling rate fs must be at least twice the highest frequency component in the signal.

The highest frequency component in the signal after filtering is 50 Hz.

Therefore, the minimum sampling rate fs is:

$fs \geq 2\times 50\text{ Hz}=100\text{ Hz}$

b)

The X(0) sample of the FFT represents the DC component of the signal. The average value of the x(n) sequence is given as 17.

When performing an FFT, the DC component is represented by the magnitude of the X(0) term, which is the sum of the samples divided by the number of samples.

Given:

- The average value of x(n) is 17
- 2048 samples collected

The signal x(n) is the output of the analog-to-digital converter after the lowpass filter.

The lowpass filter rejects the 2050 Hz component, leaving the 50 Hz component and the DC bias D. Therefore, the DC bias D is added to the signal.

The magnitude of the X(0) sample in the FFT is the average value of the signal multiplied by the number of samples:

X(0) = $\sum_{n=0}^{N-1} x(n) = 17x2048 = 34816$

Therefore, the magnitude of the FFT's X(0) sample is 34816.

ES6:

4.6 Assume you've purchased a high-performance commercial *real-time* spectrum analyzer that contains an analog-to-digital converter so that the analyzer can accept analog (continuous) $x(t)$ input signals. The analyzer can perform a 1024-point FFT in 50 microseconds and has two banks of memory in which the analog-to-digital converter samples are stored as shown in Figure P4–6(a). An FFT is performed on 1024 $x(n)$ signal samples stored in Memory Bank 1 while 1024 new $x(n)$ time samples are being loaded into Memory Bank 2.

At the completion of the first FFT, the analyzer waits until Memory Bank 2 is filled with 1024 samples and then begins performing an FFT on the data in that second memory. During the second FFT computation still newer $x(n)$ time samples are loaded into Memory Bank 1. Thus the analyzer can compute 1024 FFT results as often as once every 50 microseconds, and that is the

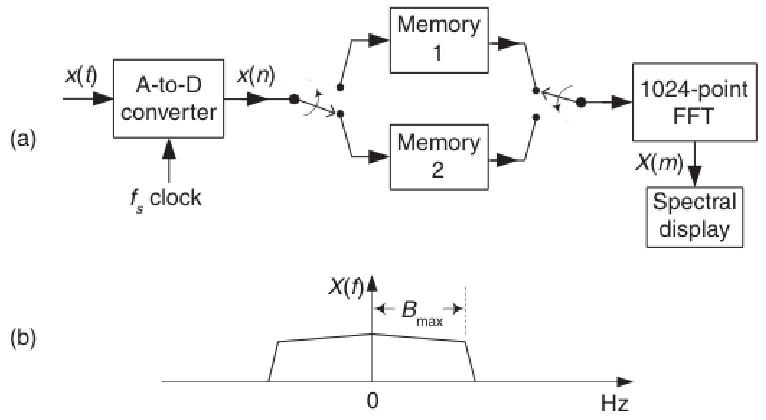


Figure P4–6

meaning of the phrase "real-time spectrum analyzer." Here's your problem: In a *lowpass sampling* scenario what is the maximum one-sided bandwidth B_{max} of the analog $x(t)$ input signal for which the analyzer can perform real-time FFTs without discarding (ignoring) any discrete $x(n)$ samples? (The definition of bandwidth B_{max} is shown in Figure P4–6(b).)

Answer:

The analyzer can perform a 1024 point FFT in 50 microseconds.

The analyzer alternates between two memory banks, each holding 1024 samples.

During the 50 microseconds required for an FFT computation, 1024 new samples are collected.

To find the sampling rate fs, it is needed to determine how many samples are collected per second.

Since 1024 samples are collected in 50 microseconds:

$fs=1024$ samples/50×10^{-6} seconds = 20.48×10^{6} Hz=20.48 MHz.

To avoid aliasing in a lowpass sampling scenario, the sampling rate fs must be at least twice the maximum frequency Bmax of the input signal.

$fs\geq2Bmax$

Given the sampling rate fs:

$20.48\ MHz\geq2Bmax$

$Bmax\leq20.48\ MHz/2$

$Bmax\leq10.24\ MHz$

The maximum one-sided bandwidth Bmax of the analog x(t) input signal for which the analyzer can perform real-time FFTs without discarding any discrete x(n) samples is 10.24 MHz

ES7:

4.7 Here's an interesting problem. Assume we performed lowpass sampling of an analog $x(t)$ signal, at a sample rate of $f_s = 20$ kHz, obtaining a discrete sequence $x_1(n)$. Next we perform an FFT on $x_1(n)$ to obtain the $|X_1(m)|$ FFT magnitude results presented in Figure P4–7(a). There we see our signal of interest in the range of 0 to 4 kHz, but we detect a high-magnitude narrowband spectral noise signal centered at 5 kHz.

Experimenting, as every good engineer should, we change the sampling rate to $f'_s = 19$ kHz, obtaining a new discrete sequence $x_2(n)$. Performing an FFT on $x_2(n)$, we obtain the $|X_2(m)|$ FFT magnitude results presented in Figure P4–7(b). In our new spectral results we see our signal of interest remains in the frequency range of 0 to 4 kHz, but the narrowband spectral noise signal is now centered near 4 kHz! (If this ever happens to you in practice, to quote Veronica in the 1986 movie *The Fly*, "Be afraid. Be very afraid.") Describe the characteristic of the analog $x(t)$ that would account for the unexpected shift in center frequency of the narrowband noise in the $|X_2(m)|$ FFT results.

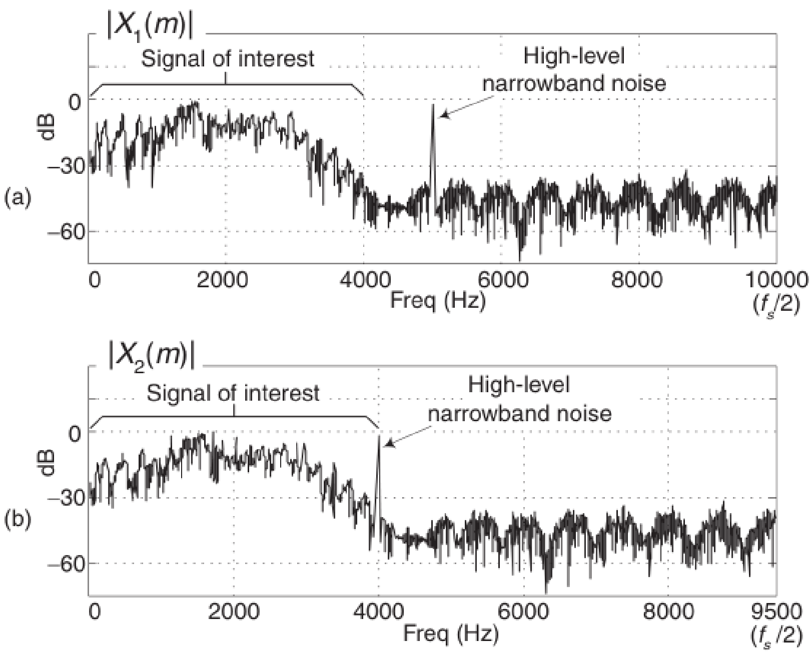


Figure P4–7

Answer:

The Nyquist criterion states that to accurately sample a signal without aliasing, the sampling rate fs must be at least twice the highest frequency component fmax of the signal:

fs≥2fmax

When a signal is sampled below its Nyquist rate, higher frequency components can fold back into the lower frequencies, creating distortions called aliases. This can be calculated as:

$f_{alias} = |f - kfs|$, where k is an integer.

Initial Sampling is at 20 kHz:

The FFT magnitude plot |X1(m)| shows a signal of interest between 0 to 4 kHz and a narrowband noise centered at 5 kHz. Given that the sampling rate fs = 20 kHz, the Nyquist frequency is fs/2 = 10 kHz. Thus, a 5 kHz noise component is correctly represented in the sampled data without aliasing.

Changed Sampling Rate to 19 kHz:

When the sampling rate is reduced to 19 kHz, the Nyquist frequency becomes fs'=9.5 kHz. This is still above 5 kHz, so direct aliasing of the 5 kHz component is not the issue. However, due to the reduced sampling rate, the high-magnitude narrowband noise shifts in the FFT magnitude plot |X2(m)|.

The shift of the narrowband noise from 5 kHz to near 4 kHz when changing the sampling rate from 20 kHz to 19 kHz is a problem of aliasing. When the signal was sampled at 19 kHz, the noise component at 5 kHz gets aliased according to the relationship:

$f_{alias} = |f_{signal} + k\ fs|$

For a 5 kHz signal sampled at 19 kHz, the aliased frequency can be calculated as:

$f_{alias} = |5000+k\ 19000|$

Considering the nearest Nyquist zone (k = -1), we get:

$f_{alias} = |5000-19000|=|-14000|=14000$ Hz.

Since 14000 Hz is above the Nyquist frequency (9.5 kHz), it folds back into the baseband. The resulting frequency within the baseband (0 to 9.5 kHz) appears closer to 4 kHz due to spectral wrapping around the Nyquist frequency.

To calculate the folding back frequency 14000 Hz into the Nyquist range (0 to 9.5 kHz), it is needed to subtract the Nyquist frequency multiples until it falls within the baseband.

Folding freq:

14000−9500=4500 Hz.

Therefore, the observed shift in the narrowband noise in the FFT from 5 kHz to near 4 kHz when the sampling rate changes from 20 kHz to 19 kHz is due to aliasing effects, where higher frequency components fold back into the baseband due to undersampling.

ES8:

4.8 In the text's derivation of the radix-2 FFT, to simplify the algebraic notation we represented unity-magnitude complex numbers (what we called "twiddle factors") in the following form:

$$\alpha = W_N^k.$$

If $k = 3$ and $N = 16$:

(a) Express α as a complex number in polar (complex exponential) form.
(b) Express α as a complex number in rectangular form.

Answer:

To solve this problem, it is needed to derive the twiddle factor α both in polar (complex exponential) form and rectangular form for given values of k=3 and N = 16.

a) The twiddle factor α is represented as:
   $\alpha = W_N^k$, where W_N is defined as: $W_N = e^{-j2\pi/N}$

   For N = 16 and k=3: $W_{16} = (e^{-j2\pi/16})^k = e^{-j6\pi/16} = e^{-j3\pi/8}$

   So, in polar form, α is: $\alpha = e^{-j3\pi/8}$

b) To express α in rectangual form, it is needed to convert the complex exponential form to its rectangular form.
   $e^{j\Theta} = \cos(\Theta) + j\sin(\Theta)$

   For $\Theta = -3\pi/8$: $\alpha = e^{-j3\pi/8} = \cos(-3\pi/8) + j\sin(-3\pi/8)$

   Since cos is an even function and sin is an odd function:

The twiddle factor in rectangular form is:

$\alpha = e^{-j3\pi/8} = \cos(3\pi/8) - j\sin(3\pi/8) = 0.382 - j0.9238$

ES9:

4.9 Reviewing the 8-point FFT signal-flow diagram in the text's Figure 4–5:

(a) Which $x(n)$ input samples affect the value of the FFT's $X(2)$ output sample?

(b) Which $x(n)$ input samples affect the value of the FFT's $X(5)$ output sample?

Answer:

a)

- X(2) is influenced by the intermediate values A(2) and B(2) in the second stage.

  $X(2) = A(2) + W^2_8 B(2)$

- A(2) is influenced by x(0), x(4), x(2) and x(6).

  $A(2)=x(0) + x(4) +(x(2) + x(6))\, W^2_4$

- B(2) is influenced by x(1), x(5), x(3) and x(7).

  $B(2) = x(1) + x(2) + (x(3) + x(5))\, W^2_4$

So, X(2) is affected by the input samples x(0), x(1), x(2), x(4), x(5) x(6),x(3), and x(7), therefore all inputs.

b)

- X(5) is influenced by A(1) and B(1)

  $X(5) = A(1) + B(1)\, W^5_8$

- A(1) is influenced by x(0), x(4), x(2) and x(6).

  $A(1) = x(0) – x(4) + (x(2) – x(6))\, W^1_4$

- B(1) is influenced by x(1), x(5), x(3) and x(7).

  $B(1) = x(1) – x(5) + (x(3)-x(7))\, W^1_4$

So, X(5) is affected by the input samples x(0), x(1), x(2), x(5), x(3), x(4), x(6) and x(7).

Therefore even X(5) depends on all input samples.

ES10:

4.10 Figure P4–10 shows a 4-point FFT using standard decimation-in-time butterflies. Redraw that FFT using *optimized* decimation-in-time butterflies as shown in the text's Figure 4–14(c). In your drawing provide the correct indices for the $X(m)$ output samples.

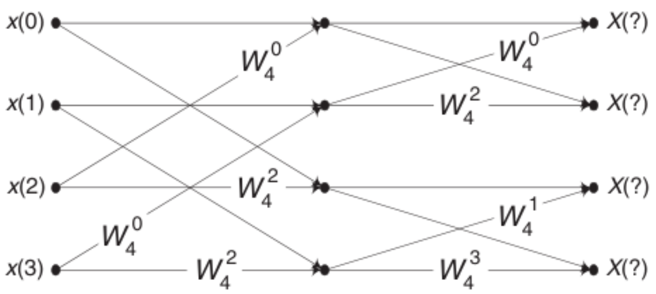


Figure P4–10

Answer:

To redraw the 4-point FFT using optimized decimation-in-time butterflies as shown in Figure 4-14(c), it is needed to modify the original FFT structure by applying the optimizations where the multiplication by W_N^k and -1 is done as indicated in figure 4-14(c) in the book.

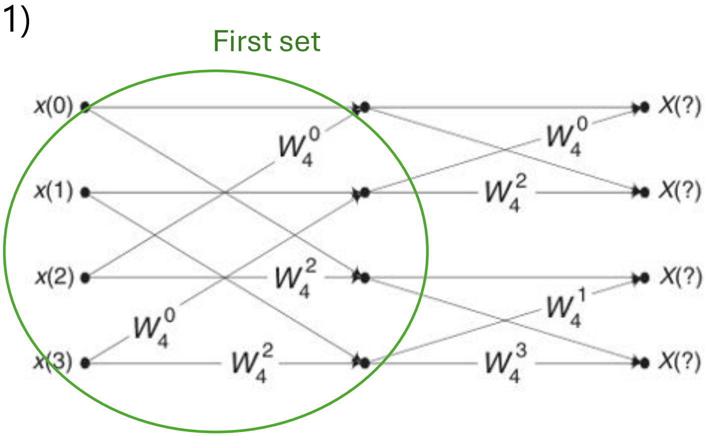
The multipliers in the figure are W_4^0, W_4^1, W_4^2, and W_4^3.

In the optimized structure, the multipliers and negations are simplified.

Each butterfly now only involves a multiplication by W_N^k and a possible multiplication by −1.

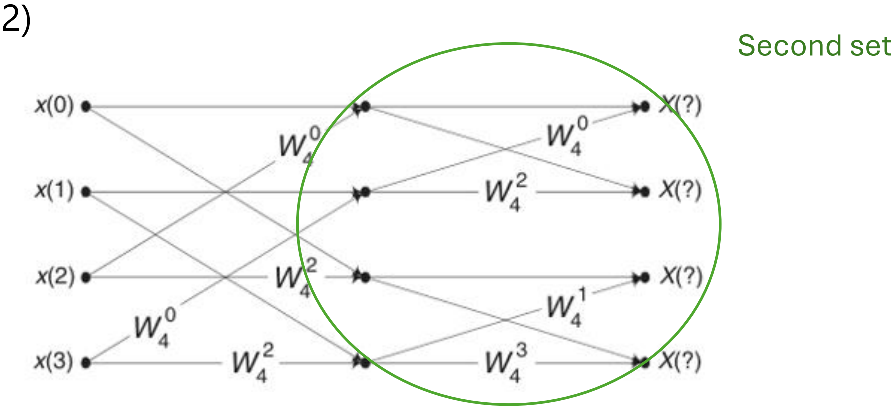
Steps for redrawing:

1)

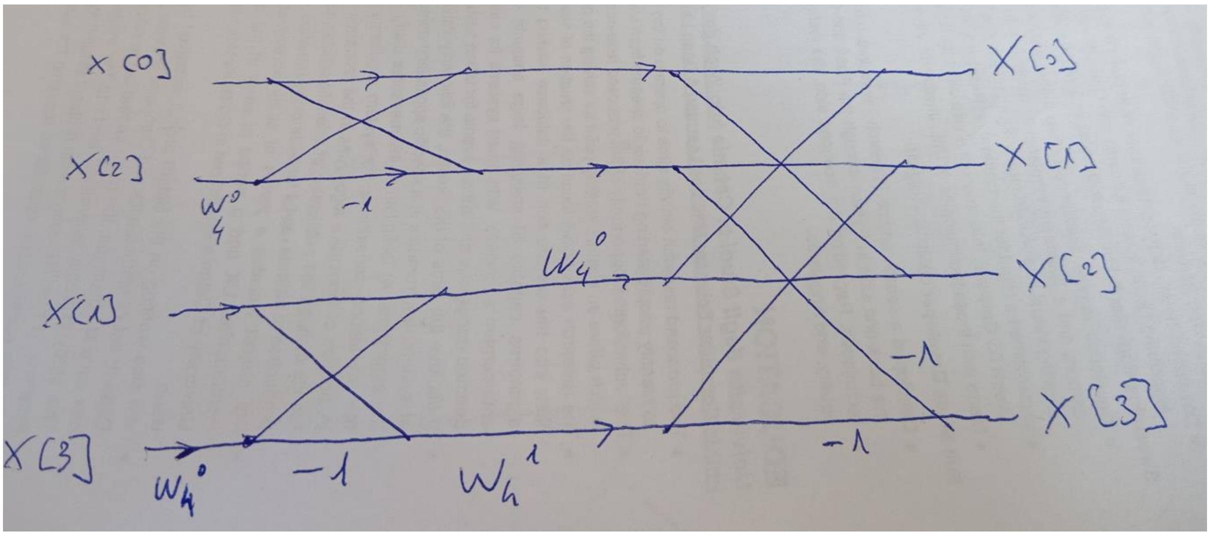


- Use the optimized butterfly structure for the first set of butterflies.
- Inputs: x(0) and x(2), x(1) and x(3).
- Outputs: x’(0), x’(1), x’(2), x’(3).

2)



- Apply the optimized butterfly structure again for the second set of butterflies.
- Inputs: x’(0), x’(1), x’(2), x’(3).
- Outputs: X(0), X(1), X(2), X(3).



This drawing incorporates the optimized butterfly operations, showing how the signals flow through the FFT stages. The multipliers and their positions are simplified according to the optimized structure from Figure 4-14(c).