5. Complex-valued DFT. FFT

1. Parsevals' relation
2. Fourier transform pairs
3. Complex-valued DFT
4. Fast Fourier transform
5. Efficient use of DFT

[Smith, ch. 11, 12, 30 and 31]

1. Parseval's relation

Since the time and frequency domains are equivalent representations of the same signal, they must have **the same *energy*** (Parseval's relation).

For the DFT, Parseval's relation is expressed as:

$$\sum_{i=0}^{N-1} x[i]^2 = \frac{2}{N} \sum_{k=0}^{N/2} (Mag\bar{X}[k])^2$$

where $x[i]$ is a time domain signal (N samples) and $\bar{X}[k]$ is its modified frequency spectrum (N/2 + 1 sinusoids) (i.e. after dividing the X[0] and X[N/2] by the $\sqrt{2}$).

Energy is proportional to the *amplitude squared*.

- The left side of this equation is the total energy contained in the time domain signal, found by summing the energies of its N samples.
- The right side is the energy contained in the frequency domain, found by summing the energies of the $N/2 + 1$ sinusoids.

- Recall that the Inverse DFT requires first to divide the first and last points (sample 0 and $N/2$) by 2, and then dividing all of the points by $N/2$.
- This provides the amplitudes of the sinusoids, not the *root-mean-square* (rms) amplitude needed for energy calculations.
- In a sinusoid, the peak amplitude is converted to rms by dividing by $\sqrt{2}$.
- This correction must be made to all of the frequency domain values, *except* sample 0 and $N/2$.
- This is because these two sinusoids are unique; one is a constant value, while the other alternates between two constant values. For these two special cases, the *peak* amplitude is already equal to the *rms* value.
- The last step is to divide the summed value by N, to account for each sample in the frequency domain being converted into a sinusoid that covers N values in the time domain.

2. Fourier Transform Pairs

For every *time domain* waveform there is a corresponding *frequency domain* waveform, and vice versa.

For example, a rectangular pulse in the time domain coincides with a sinc function (sin(x)/x) in the frequency domain.

Duality provides that the reverse is also true; a rectangular pulse in the frequency domain matches a sinc function in the time domain.

Waveforms that correspond to each other in time- and frequency domain are called **Fourier transform pairs**.

1) Delta Function Pairs

A **delta function** in the time domain has a frequency spectrum, where the **magnitude** is a constant value, while the **phase** is entirely zero.

Shifting the time domain waveform does not affect the magnitude, but **adds a linear component** to the **phase**.

The horizontal axes in the frequency domain run from -0.5 to 0.5. That is, they show the ***negative* frequencies** in the spectrum, as well as the ***positive*** ones. The negative frequencies are **redundant** information.

The frequency domain of the delta function in **rectangular form** :

- the real and imaginary parts are **sinusoidal oscillations** that are difficult to attach a meaning to;
- this documents again the **duality** of the DFT - each **sample** in the time domain corresponds to **sinusoids** in the frequency domain.

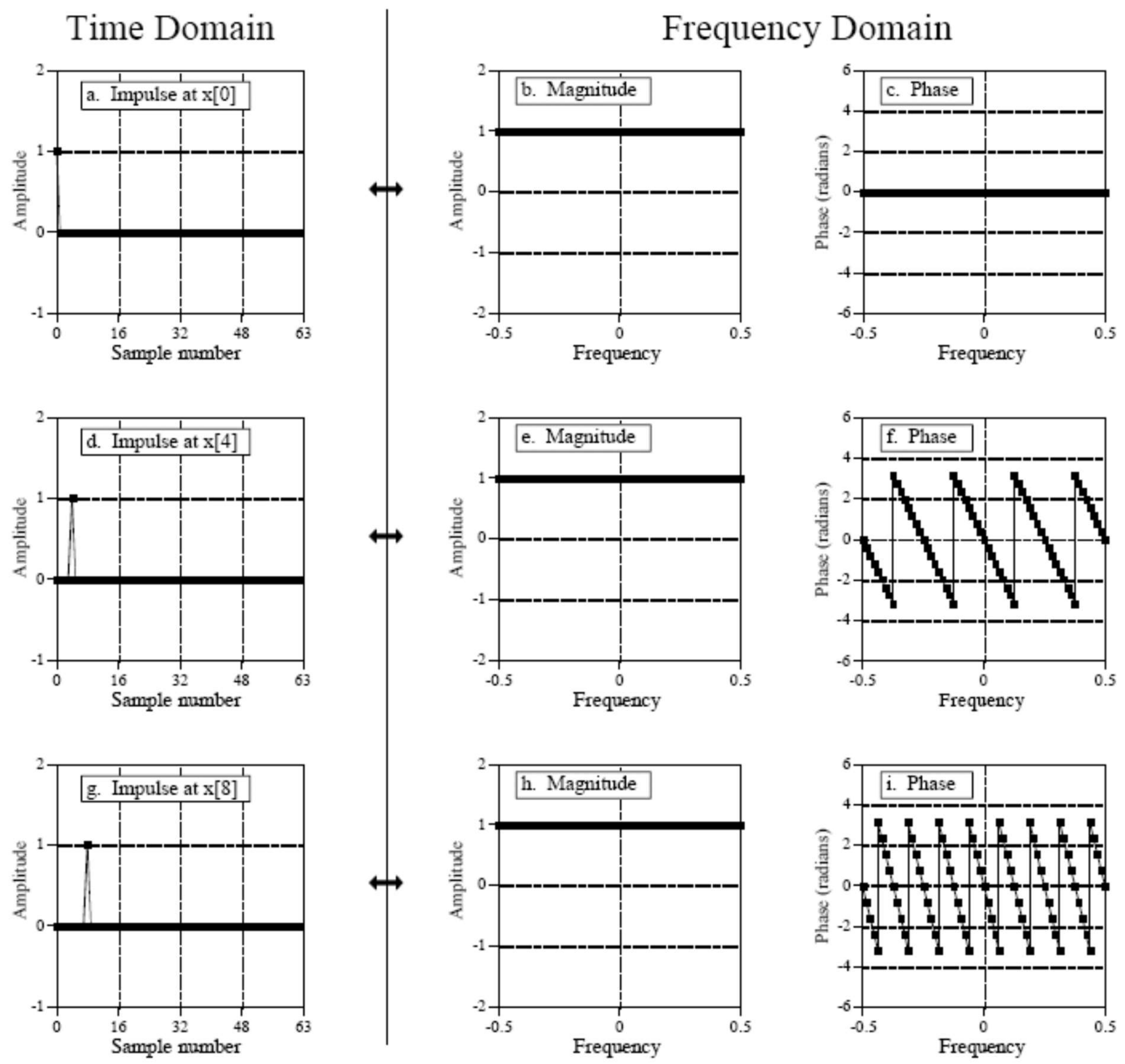


Fig. 1 Delta function pairs in *polar form*. An impulse in the time domain corresponds to a constant magnitude and a linear phase in the frequency domain.

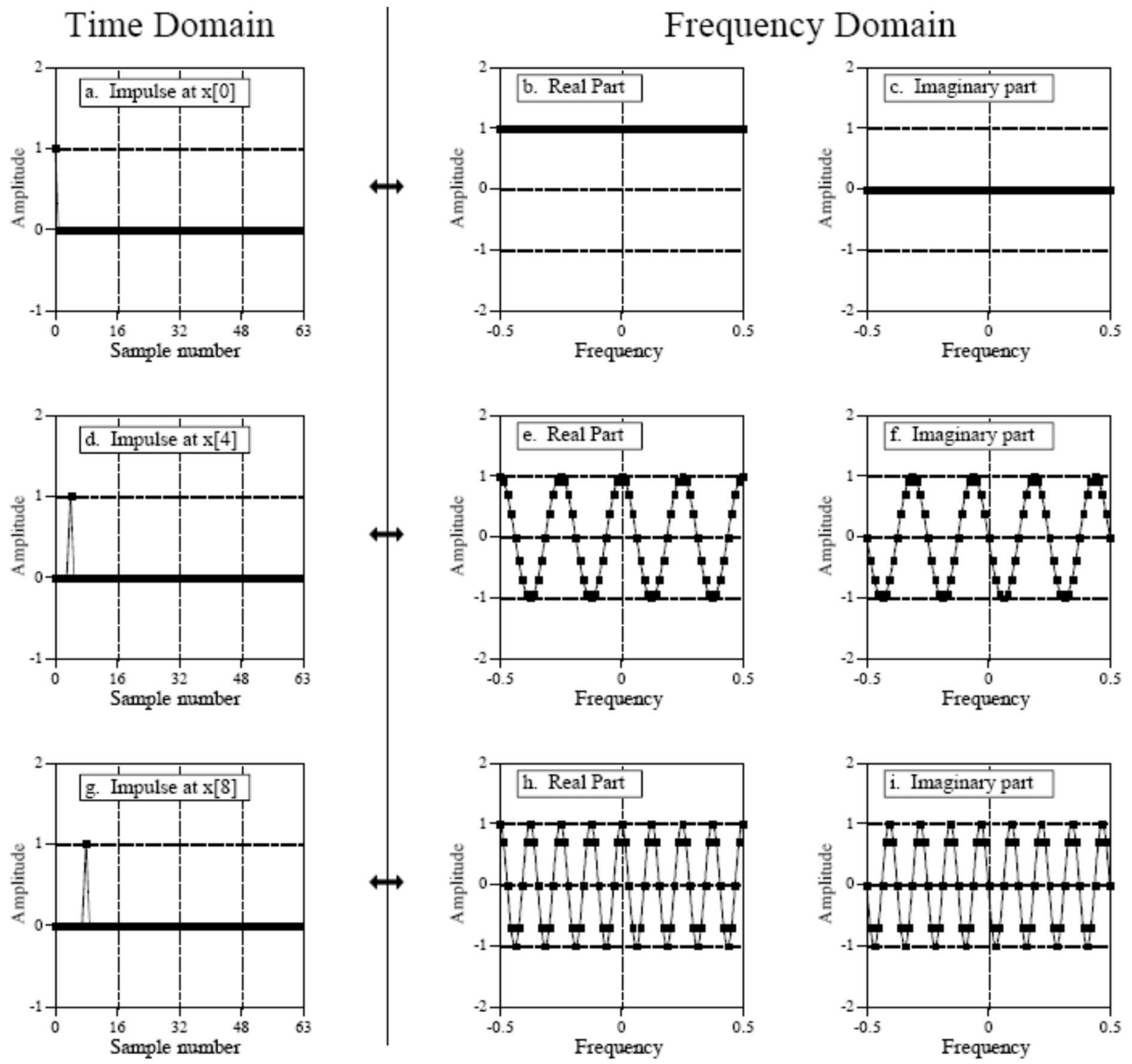


Fig. 2 Delta function pairs in *rectangular form*. Each impulse sample in the time domain results in a cosine wave in the real part, and a negative sine wave in the imaginary part.

Alternative DFT method

Observation

An impulse at sample number 4 results in the real part of the frequency spectrum in 4 cycles of a cosine wave being added in the time domain.
An impulse at sample number 4 results in the imaginary part of the frequency spectrum in 4 cycles of a negative sine wave in the time domain.

This observation leads to another way to **calculate the DFT** (alternatively to correlating the time domain with sinusoids):

1. Each **sample** in the time domain results in a **cosine wave** being added to the real part of the frequency domain, and a **negative sine wave** being added to the imaginary part.
2. The **amplitude** of each sinusoid is given by the **amplitude** of the time domain sample.
3. The **frequency** of each sinusoid is provided by the **sample number** of the time domain point.

2) The Sinc Function

Figure 3 illustrates a common transform pair:

- the **rectangular pulse** and
- the **sinc** function.

The normalized sinc function is defined as:

$$\mathrm{sinc}(x) = \sin(\pi x)/ (\pi x) \text{ , where } \mathrm{sinc}(0) = 1.$$

I.e. the normalized sinc is a sine wave that decays in amplitude as $1/(\pi x)$.

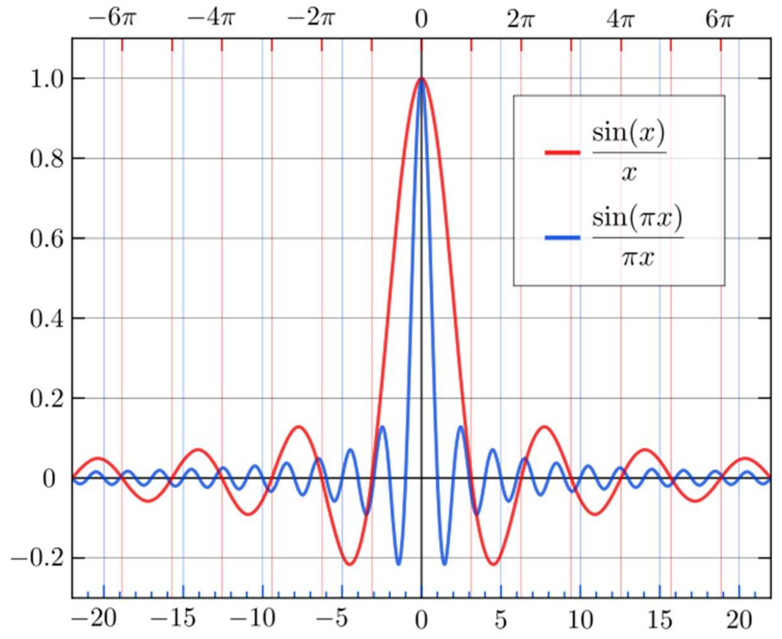


Fig. 3 The sinc function (from Wikipedia)

Unwrapped magnitude of polar form

We are using the term **unwrapped magnitude** to indicate that it can have both positive and negative values.

The sinc function in frequency domain:

- The **magnitude** is an oscillation that decreases in amplitude with increasing frequency.
- The **phase** is composed of all zeros, as the time domain signal is symmetrical around sample number zero.

But by definition, the **magnitude** must always be positive.

In Fig. 4 (d, e) the magnitude is **made all positive** by introducing a **phase shift of** π at all frequencies where the unwrapped magnitude is negative.

In Fig. 4 (f), the signal is shifted and no longer centered on sample number zero. This doesn't change the magnitude of the frequency domain, but it adds a linear component to the phase.

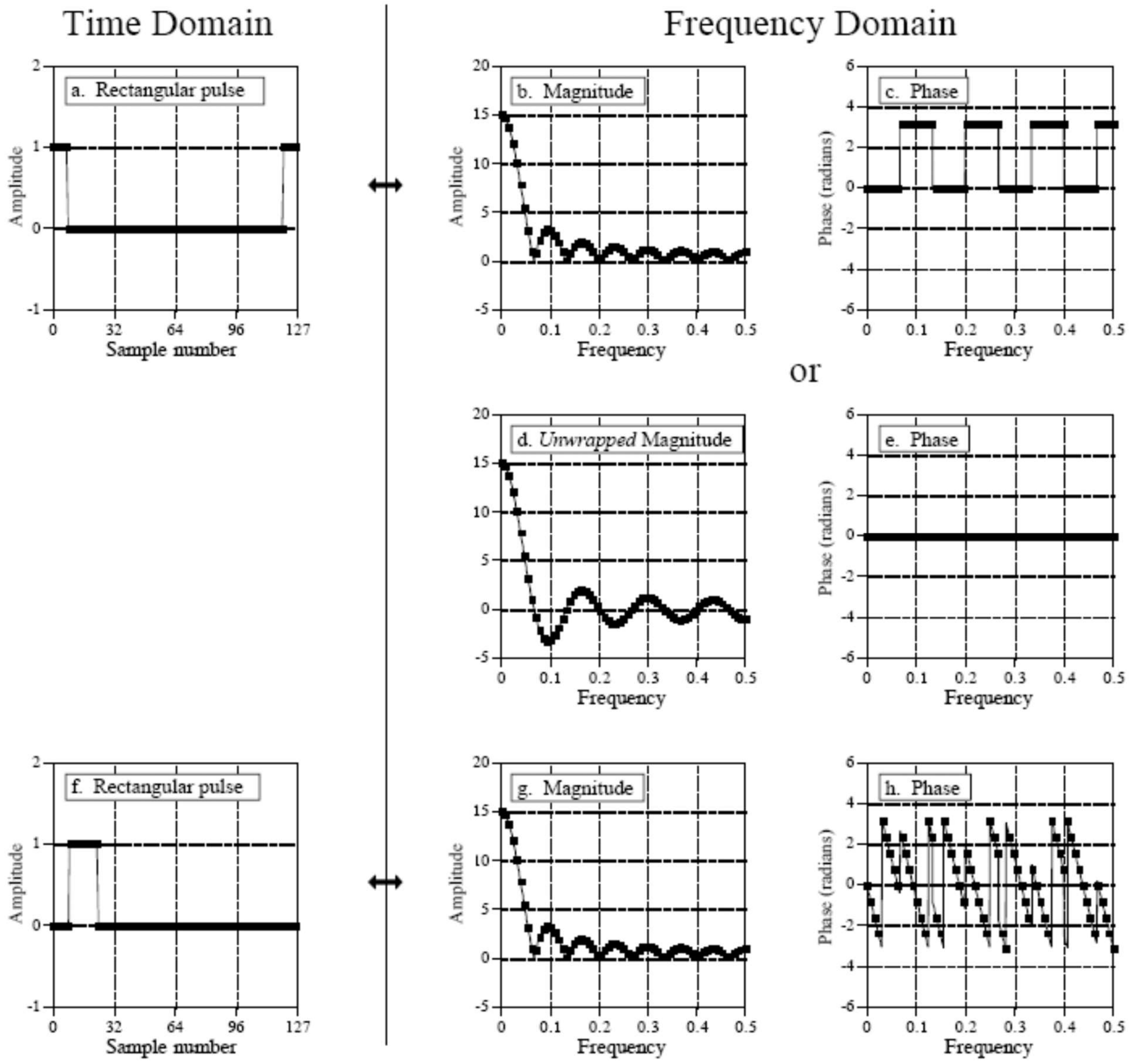


Fig. 4. A rectangular pulse in time domain corresponds to a *sinc* function in the frequency domain.

The sinc spectrum

An N point time domain signal that contains a **unity amplitude rectangular pulse, *M* points wide**, has a **DFT**-frequency spectrum as:

$$MagX[k] = \left| \frac{\sin(\pi k M / N)}{(\pi k M / N)} \right|$$

(to avoid the division by zero, use $X[0] = M$)

This equation provides ($N/2 + 1$) ***samples*** in the frequency spectrum,

The **Discrete-time FT** can be used to express the frequency spectrum as a **fraction of the sampling rate *f***, running continuously from 0 to 0.5:

$$MagX(f) = \left| \frac{sin(\pi f M)}{(\pi f M)} \right|$$

This equation provides a ***continuous curve***.

Aliasing effects for sinc

The amplitude of the oscillation **does not decay to zero** before a frequency of 0.5 is reached. The waveform **continues into the next period** where it is **aliased**.

When the frequency spectrum of the rectangular pulse **is *not* aliased** (because the time domain signal is continuous), it is of the general form:

sin (*x*)/ *x*, i.e., a sinc function.

- For **continuous** signals, the **rectangular pulse** and the sinc **function** are Fourier transform pairs.
- For **discrete** signals this is only an **approximation**, with the error being due to aliasing.

Zeros of the sinc function

A key question of the sinc function is the location of the **zero crossings**.

These occur at frequencies where an **integer number** of the sinusoid's cycles **fit into the rectangular pulse**.

For example, if the pulse is 20 points wide, the **first zero** in the frequency domain is at the frequency that makes **one complete cycle** in 20 points. The second zero is at the frequency that makes **two complete cycles** in 20 points, etc.

Sinc function in the time domain

By duality of the two domains, the following FT-pair exists:

- a rectangular pulse in the frequency domain corresponds to
- a sinc function (plus aliasing) in the time domain.

Including the effects of aliasing, the time domain signal is given by:

$$\boxed{x[i] = \frac{1}{N}\frac{\sin(2\pi i(M-1/2)/N)}{\sin(\pi i/N)}}$$

Assumptions

1. In the frequency domain, the pulse has an amplitude of one, and runs from sample number 0 through sample number M-1.
2. The parameter N is the length of the DFT, and $x[i]$ is the time domain signal with i running from 0 to N-1.
3. To avoid the division by zero, use

$$x[0] = (2M-1)/N.$$

Now imagine that the frequency domain is so finely sampled that it turns into a **continuous curve**. This makes the time domain infinitely long with **no periodicity (no aliasing)**.
The Fourier transform is used here:

$$\boxed{x[i] = \frac{\sin(2\pi f_c i)}{i\pi}}$$

In the frequency domain, the pulse has an amplitude of one, and runs from zero frequency to the cutoff frequency, f_c , between 0 and 0.5.
To avoid the division by zero, use $x[0] = 2f_c$.

The **rectangular pulse** in the frequency domain is the perfect **low-pass filter**. The sinc function is the filter kernel for the perfect low-pass filter.

➔ A very useful class of digital filters is the **windowed sinc filter**.

3) Other Transform Pairs

Triangular pulse ←→ sinc squared

A **triangular pulse** in the time domain corresponds to a sinc **function squared** (plus aliasing) in the frequency domain.

A $2M$ - 1 point triangle in the time domain can be formed by **convolving** an M point **rectangular pulse with itself**.

Since convolution in the time domain results in multiplication in the frequency domain, convolving a waveform with itself will **square** the frequency spectrum.

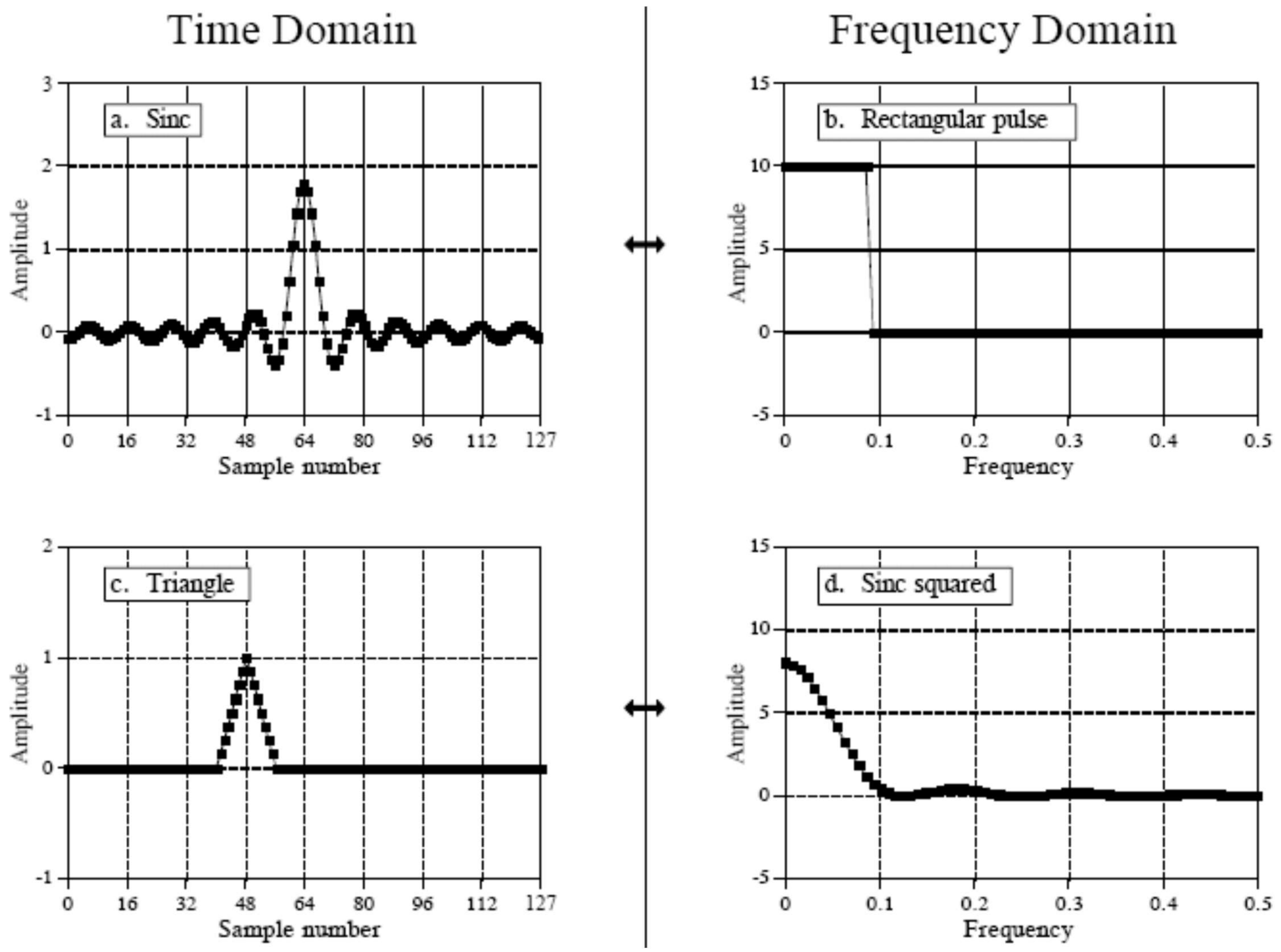


Fig. 5. The (triangular pulse – sinc squared) pair.

Gaussian ←→ Gaussian

The **Gaussian** is the only waveform that is its **own** Fourier Transform.

Remark

This is only true if we **ignore aliasing**.

The relationship between the standard deviation of the time domain and frequency domain is given by:

$$\boxed{2\pi\sigma_f = 1/\sigma_t} \,.$$

Gaussian burst - amplitude modulation of a Gaussian signal

A **Gaussian burst** formed by **multiplying a sine wave by a Gaussian**, corresponds in the frequency domain to a **Gaussian** centered somewhere other than zero frequency.

In the frequency: convolving an impulse centered at the frequency of the sine wave with a Gaussian at zero frequency produces a Gaussian centered at the frequency of the sine wave.

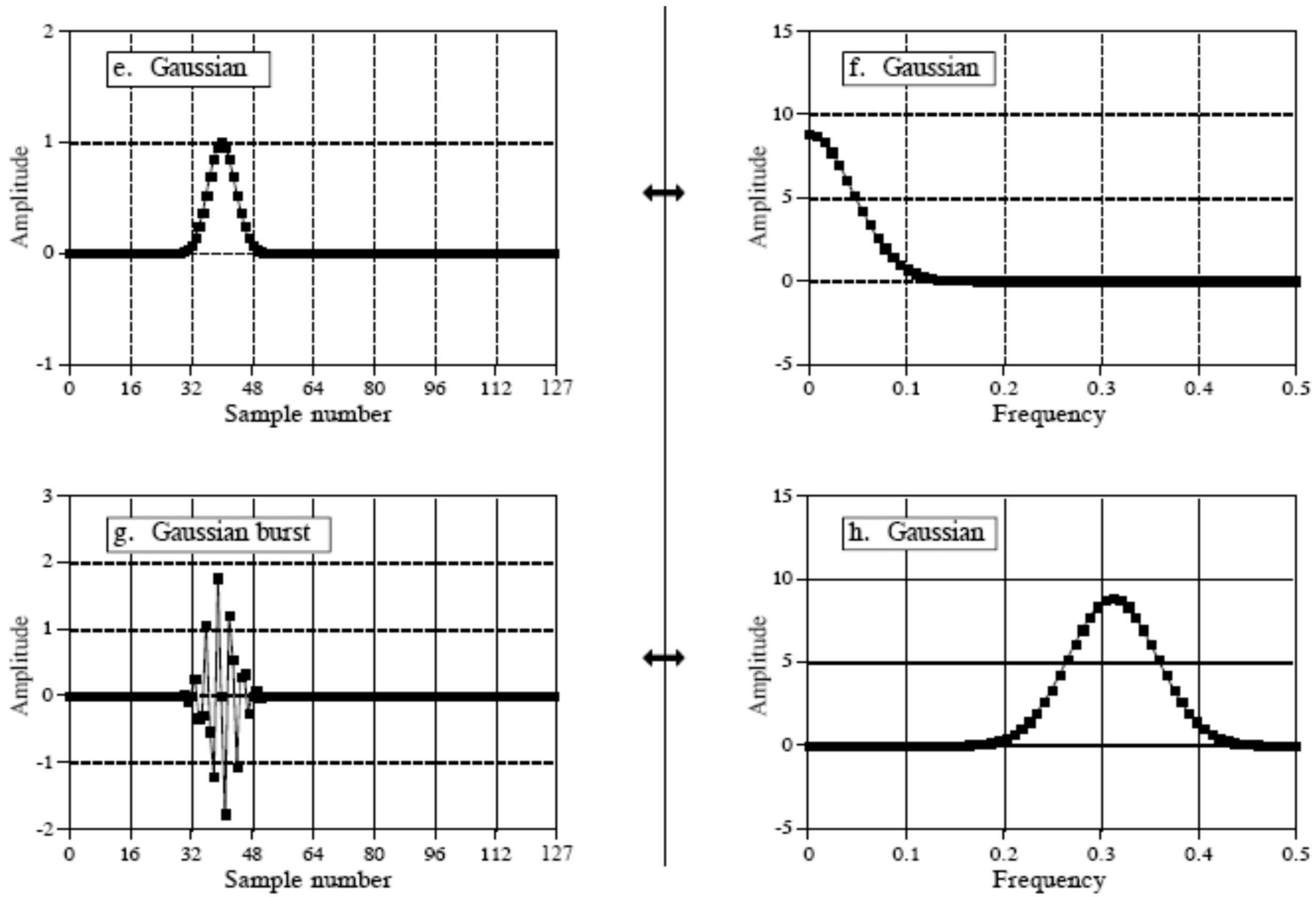


Fig. 6. Gaussian transform pairs (e,f).
Amplitude modulation of a Gaussian signal (g, h).

4) Harmonics

- If **a signal is periodic** with frequency *f*, the only frequencies composing the signal are integer **multiples of *f*,** i.e., *f*, 2*f*, 3*f*, 4*f*, etc.
- These frequencies are called **harmonics**.
  The **first harmonic** is *f*, the **second harmonic** is 2*f*, the **third harmonic** is 3*f*, and so forth.
- The first harmonic (i.e., *f*) has also a name, the **fundamental frequency**.

Harmonics can be of **any amplitude** - they usually become smaller as they increase in frequency.

If the signal is **symmetrical** around a **horizontal** axis, i.e., the top lobes are mirror images of the bottom lobes, all of the **even harmonics** will have a **value of zero**, i.e. the only frequencies contained in the signal are the fundamental, the third harmonic, the fifth harmonic, etc.

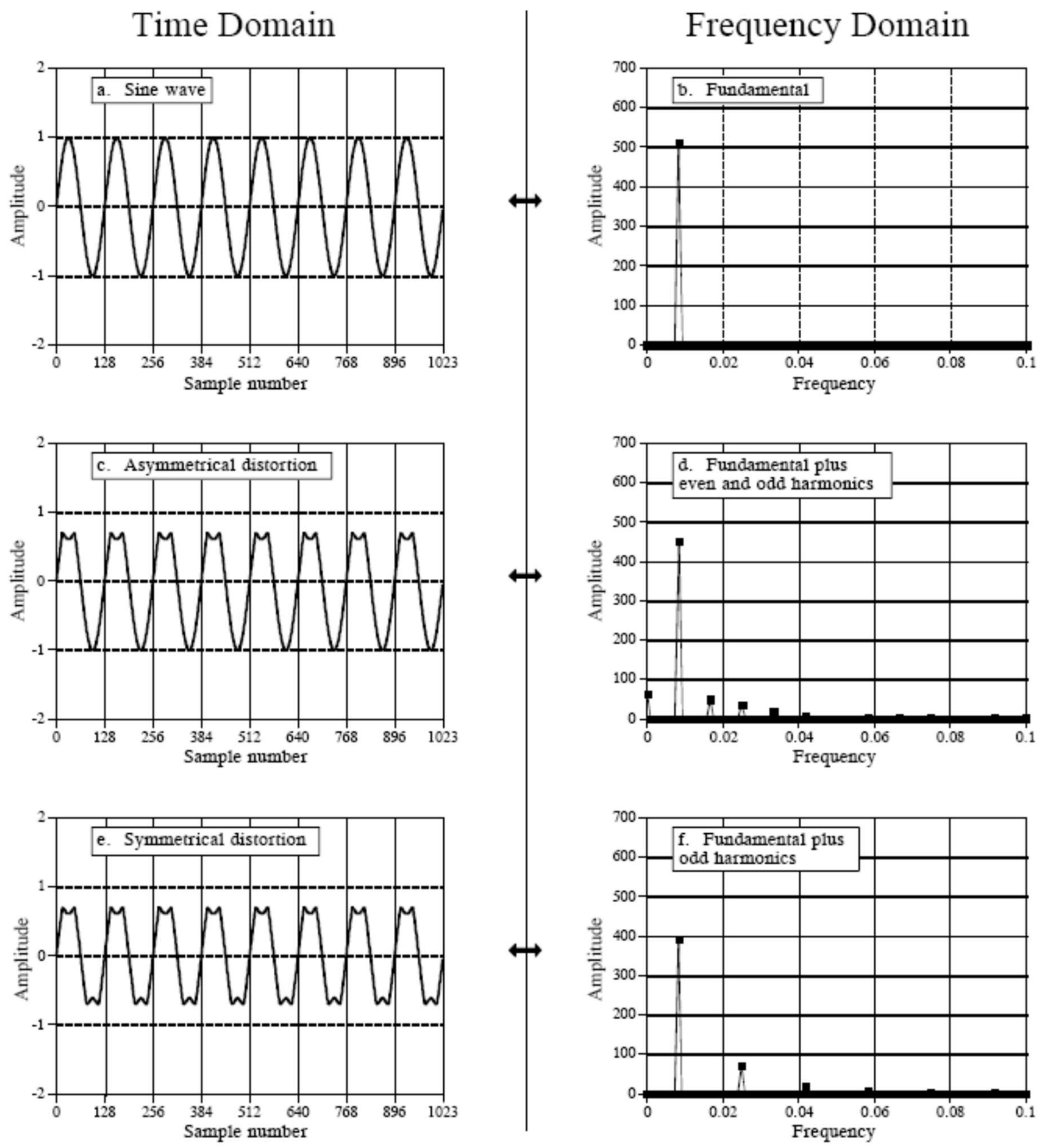


Fig. 7. Example of harmonics.
(a, b) Non-distorted sine wave and its frequency magnitude.
(c) Asymmetrical distortion results in (d) even and odd harmonics.
(e) Symmetrical distortion (produces only (f) odd harmonics.

All **continuous** periodic signals can be represented as a summation of harmonics.

Discrete periodic signals may have **a problem** of **aliasing**.

Harmonic aliasing is only a problem when nonlinear operations are performed directly on a discrete signal. Even then, the amplitude of these aliased harmonics is often low enough that they can be ignored.

Example on fig. 8 involves distorting a signal ***after*** its **digitization.**

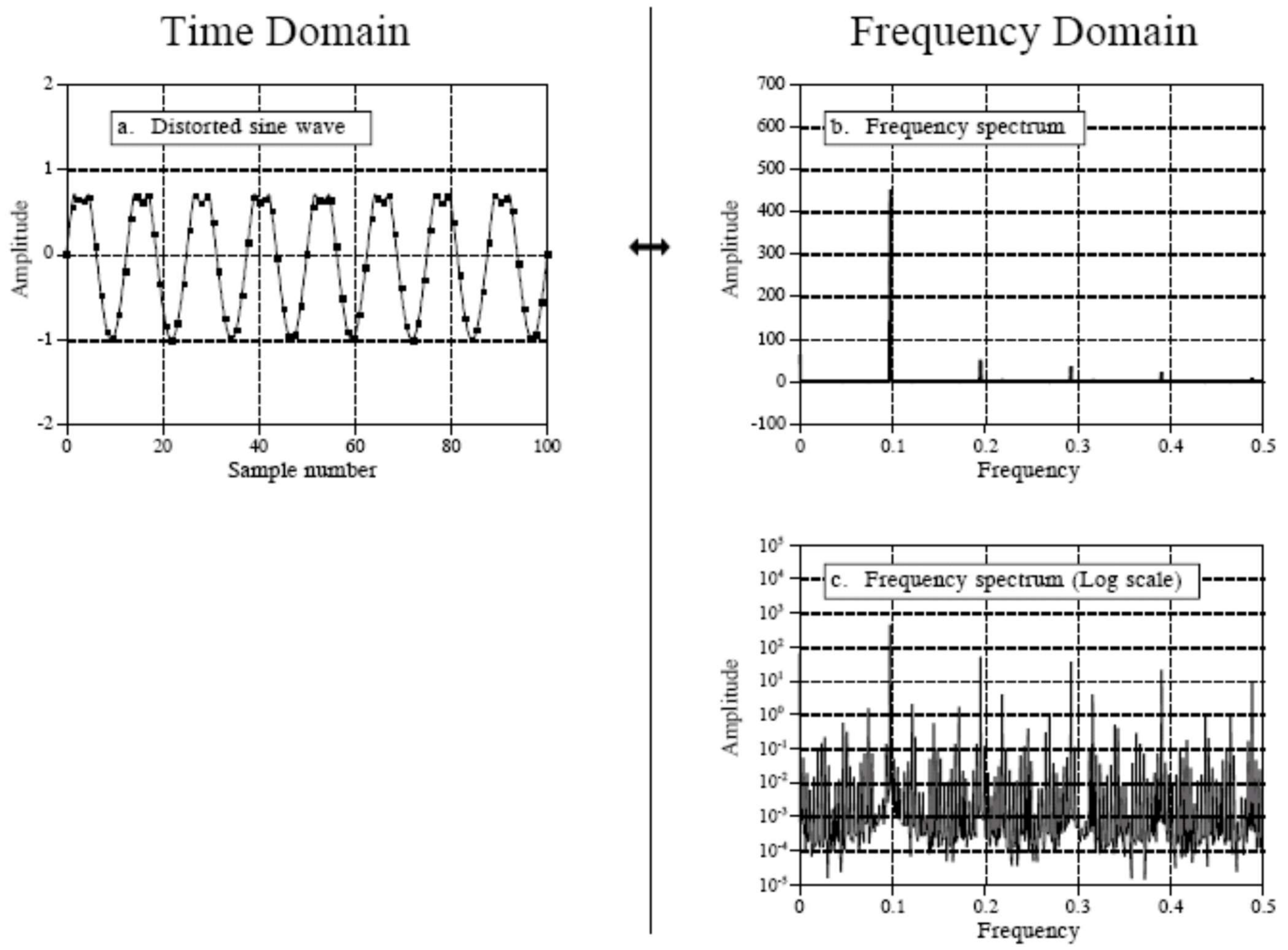


Fig. 8. Harmonic aliasing. (a) and (b): a distorted sine wave and its frequency spectrum, respectively. (c) Harmonics with a frequency greater than 0.5 will become aliased to a frequency between 0 and 0.5. Fig. (c) shows the same frequency spectrum on a logarithmic scale, revealing many aliased peaks.

The concept of harmonics is also useful for another reason:
it explains why **the DFT** views both domains as **periodic**.

In the frequency domain, an N point DFT consists of $N/2+1$ equally spaced frequencies. The frequencies **between** these samples as:
(1) having a value of zero, or
(2) not existing.

Hence a **discrete** frequency spectrum consists of **harmonics**, rather than a continuous range of frequencies.

If one domain is **discrete**, the other domain must be **periodic**.

3. Complex DFT

1) Complex numbers

j is a constant symbol representing $\sqrt{-1}$.

Polar notation

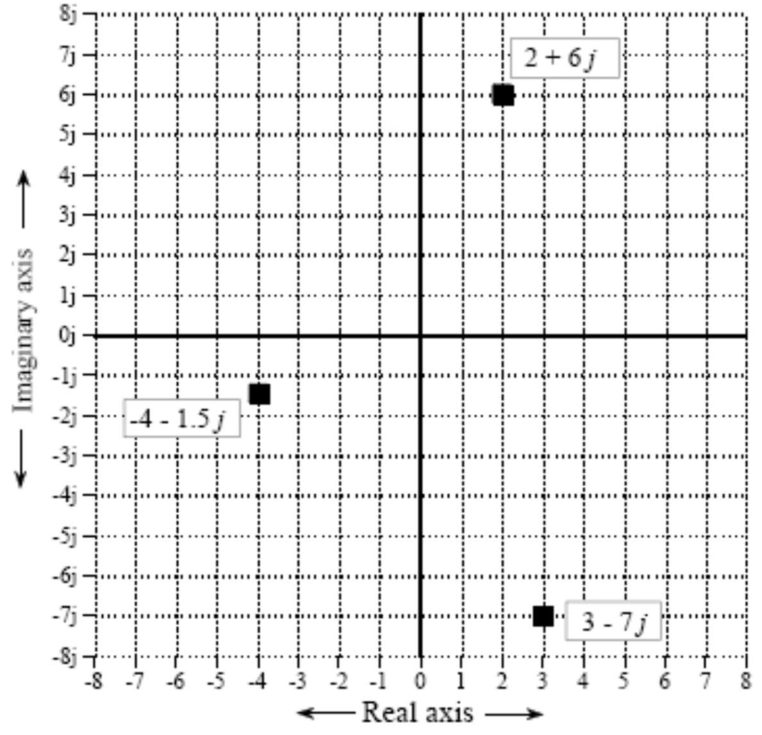


Fig. 1 The complex plane. Every complex number has a unique location in the complex plane.

The algebra of complex numbers

Addition, subtraction, multiplication, division:

$$(a+jb)+(c+jd)=(a+c)+j(b+d)$$

$$(a+jb)-(c+jd)=(a-c)+j(b-d)$$

$$(a+jb)(c+jd)=(ac-bd)+j(bc+ad)$$

$$\frac{(a+jb)}{(c+jd)}=\left(\frac{ac+bd}{c^2+d^2}\right)+j\left(\frac{bc-ad}{c^2+d^2}\right)$$

Switching the sign of the imaginary part of a complex number is called taking the **complex conjugate** (denoted by a star at the upper index of the variable), e.g.

if $Z=a+bj$, then $Z^*=a-bj$

Polar notation

The **magnitude** and the **phase angle**:

Rectangular-to-polar conversion:

$$\boxed{\begin{aligned} M &= \sqrt{(Re\ \mathrm{A})^2 + (Im\ \mathrm{A})^2} \\ \theta &= \arctan\left[\frac{Im\ \mathrm{A}}{Re\ \mathrm{A}}\right] \end{aligned}}$$

Polar-to-rectangular conversion:

$$\boxed{\begin{aligned} Re\ \mathrm{A} &= M\cos(\theta) \\ Im\ \mathrm{A} &= M\sin(\theta) \end{aligned}}$$

$$\boxed{a + jb = M(\cos\theta + j\sin\theta)}$$

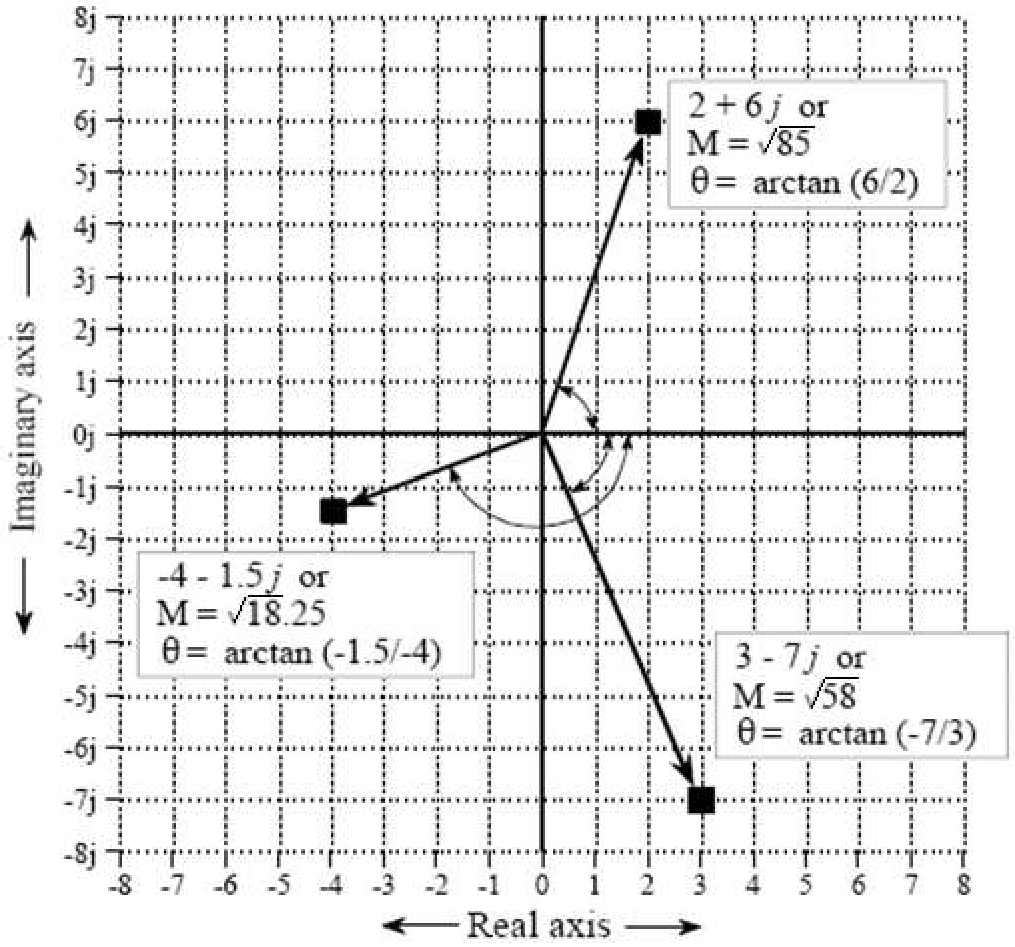


Fig. 2 Complex numbers in polar form.

Exponential form

The **Euler relation**:

$$e^{jx} = \cos(x) + j\sin(x)$$

The complex number in exponential form:

$$a + jb = M e^{j\theta}$$

This form simplifies the multiplication and division operations:

$$M_1 e^{j\theta_1} M_2 e^{j\theta_2} = M_1 M_2 e^{j(\theta_1 + \theta_2)}$$

$$\frac{M_1 e^{j\theta_1}}{M_2 e^{j\theta_2}} = \left[\frac{M_1}{M_2}\right] e^{j(\theta_1 - \theta_2)}$$

The addition and subtraction is better to perform in rectangular notation.

Representing physical problems by complex numbers

We ***change*** the physical problem into a **complex number** form, **manipulate** the complex numbers, and then ***change*** **back** into a physical answer.

Physical problems can be represented using complex numbers by:

1. a simple method of **substitution**, and
2. **mathematical equivalence**.

Substitution takes two real physical parameters and places one in the real part of the complex number and one in the imaginary part. This allows the two values to be manipulated as a single entity, i.e. a single complex number. After the desired mathematical operations, the complex number is separated into its real and imaginary parts, which again correspond to the physical parameters we are concerned with.

2) Complex Representation of Sinusoids

Complex numbers are a compact way to represent and manipulate the **sine** and **cosine waves**.

The conventional way to represent a sinusoid is:

$$M \cos(\omega t + \phi)$$

or

$$A \cos(\omega t) + B \sin(\omega t),$$

in **polar** and **rectangular form**.

Using **substitution** we change the **sinusoidal** representation to **a complex number** :

$$A \cos(\omega t) + B \sin(\omega t) \leftrightarrow \quad a + jb$$

where $A \leftrightarrow a$ and $B \leftrightarrow -b$.

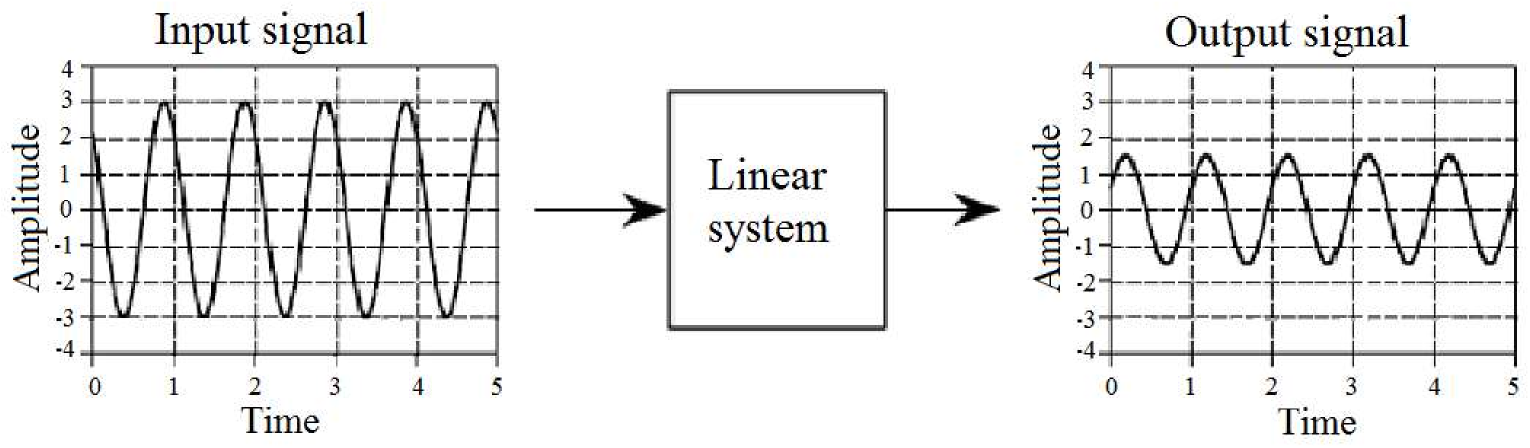
This substitution can also be applied in polar (exponential) form:

$$M \cos(\omega t + \phi) \quad \leftrightarrow \quad \mathrm{M\,e}^{j\theta}$$

where $M \leftrightarrow M$ and $\theta \leftrightarrow -\phi$.

Complex representation of systems

In fig. 3, the system is described by the complex number, $0.5\, e^{j\,3\pi/8}$, i.e. the amplitude is reduced by 0.5, while the phase angle is changed by -3π/8.



$$3\cos(\omega t + \pi/4) = 2.1213\cos(\omega t) - 2.1213\sin(\omega t)$$

$$1.5\cos(\omega t - \pi/8) = 1.3858\cos(\omega t) + 0.5740\sin(\omega t)$$

Physical representation

$\updownarrow$ $\updownarrow$

Complex numbers

$$3\,e^{-j\pi/4} \cdot 0.5\,e^{j3\pi/8} = 1.5\,e^{j\pi/8}$$

$$2.1213 + j\,2.1213 \cdot 0.1913 - j\,0.4619 = 1.3858 - j\,0.5740$$

Fig. 3 Sinusoids represented by complex numbers - sinusoidal input and output signals and the ***change*** that a linear system makes to a sinusoid.

3) Complex DFT

Recall the Euler relation:

$$e^{jx} = \cos(x) + j\sin(x)$$

We can rearrange it into:

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2}; \quad \sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$$

Expressing sinusoids as complex numbers:

$$\cos(\omega t) = \frac{e^{j(-\omega t)} + e^{j\omega t}}{2}; \quad \sin(x) = \frac{je^{j(-\omega)t} - je^{j\omega t}}{2}$$

The above expressions are each the sum of two exponentials:

- one containing a *positive* frequency (ω), and
- the other containing a *negative* frequency (-ω).

The positive and negative frequencies are treated with an equal status; it requires one-half of each to form **a complete waveform**.

Complex DFT definition

Let $\boldsymbol{x}$ =[x_0 , x_1, ... , x_{M-1}] be complex-valued samples in the input signal. The complex-valued **Discrete Fourier Transform (DFT)** of order M is a linear transformation of complex numbers:

$$F_k = \sum_{t=0}^{M-1} x_t \left(e^{-j2\pi\frac{k}{M}}\right)^t , \qquad \text{for} \quad k = 0, \dots, M-1 \tag{6.1}$$

It takes the explicit matrix form: $F = D_M \cdot x$

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{M-1} \end{bmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & m & m^2 & \cdots & m^{M-1} \\ 1 & m^2 & m^4 & \cdots & m^{2(M-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & m^{M-1} & m^{2(M-1)} & \cdots & m^{(M-1)^2} \end{pmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{M-1} \end{bmatrix} \tag{6.2}$$

with $m = e^{-j2\pi/M}$

Complex- or real-valued input

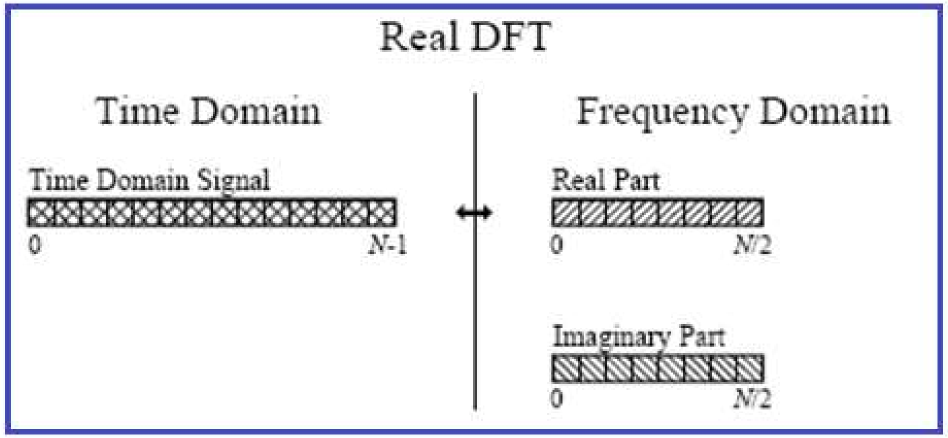
- In general the DFT converts a sampled complex-valued function of time into a sampled complex-valued function of frequency.
- Usually we operate on real-valued input functions, i.e. all the imaginary parts of the input signal are assumed to be zero.

The inverse DFT

$$D_M^{-1} = \frac{1}{M} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & m^{-1} & m^{-2} & \cdots & m^{-(M-1)} \\ 1 & m^{-2} & m^{-4} & \cdots & m^{-2(M-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & m^{-(M-1)} & m^{-2(M-1)} & \cdots & m^{-(M-1)^2} \end{pmatrix} \tag{6.3}$$

with $m = e^{-j 2\pi / M}$

How to transfer ***real DFT*** data into and out of the ***complex DFT*** format



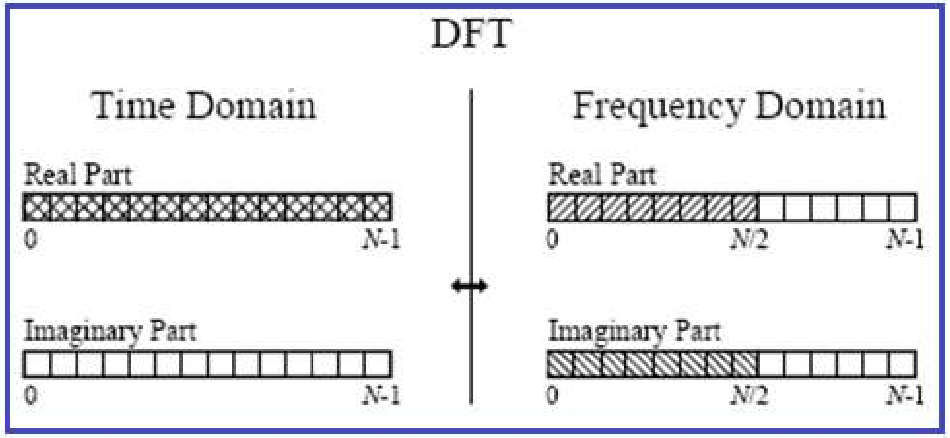


Fig. 4. Comparing the real and complex DFTs. The real DFT takes an N point time domain signal and creates two $N/2 + 1$ point frequency domain signals. The complex DFT takes an *N complex* point time domain signal and creates an *N complex* point frequency domain signal.

4. The Fast Fourier Transform

Credit for bringing the FFT to the world:

J.W. Cooley and J.W. Tukey: "An algorithm for the machine calculation of complex Fourier Series," *Mathematics Computation*, Vol. 19, 1965, pp 297-301.

The great German mathematician Karl Friedrich Gauss (1777-1855) had used the method more than a century earlier (~ 1805).

The FFT is calculating the **complex(-valued) DFT**.

1) Principle: divide-and-conquere

As the DFT is of quadratic complexity O(N^2) with respect to the size N of an input signal, it is possible to reach essential calculation acceleration if we manage to reduce the N-point DFT to two (N/2)-point DFT:

- this leads to halving the number of operations, but
- additional operations of halving the sequence and association of two (N/2)-point DFTs into one N-point are required.

At the same time each of the (N/2)-point DFT can also be calculated by replacing it with two (N/4)-point DFTs, which in their turn can be calculated by means of (N/8)-point DFTs. This recursion can be continued, so far it is possible to divide the input sequence into two.

If $N = 2^k$, where *k* is a positive integer, we can halve the sequence *k* times.

The FFT algorithms, designed for data (signal) of length $N = 2^k$, are called “radix-2 FFT algorithms”. Next, consider two ways of dividing and conquering:

1. decimation in time and
2. decimation in frequency.

2) Radix-2 decimation in time FFT

In complex DFT, the time and frequency domains each contain *one signal* made up of *N complex points*.

Radix-2 decimation in time FFT:

1. Decompose an *N* point time domain signal into *N* time domain signals each composed of a single point.
2. Calculate the *N* frequency spectra corresponding to these *N* time domain signals.
3. The *N* spectra are synthesized into a single frequency spectrum.

STEP 1. An interlaced decomposition is used each time a signal is broken in two, that is, the signal is separated into its even and odd numbered samples.
This pattern continues until there are *N* signals composed of a single point.

There are **$\log_2 N$ stages** required in this decomposition

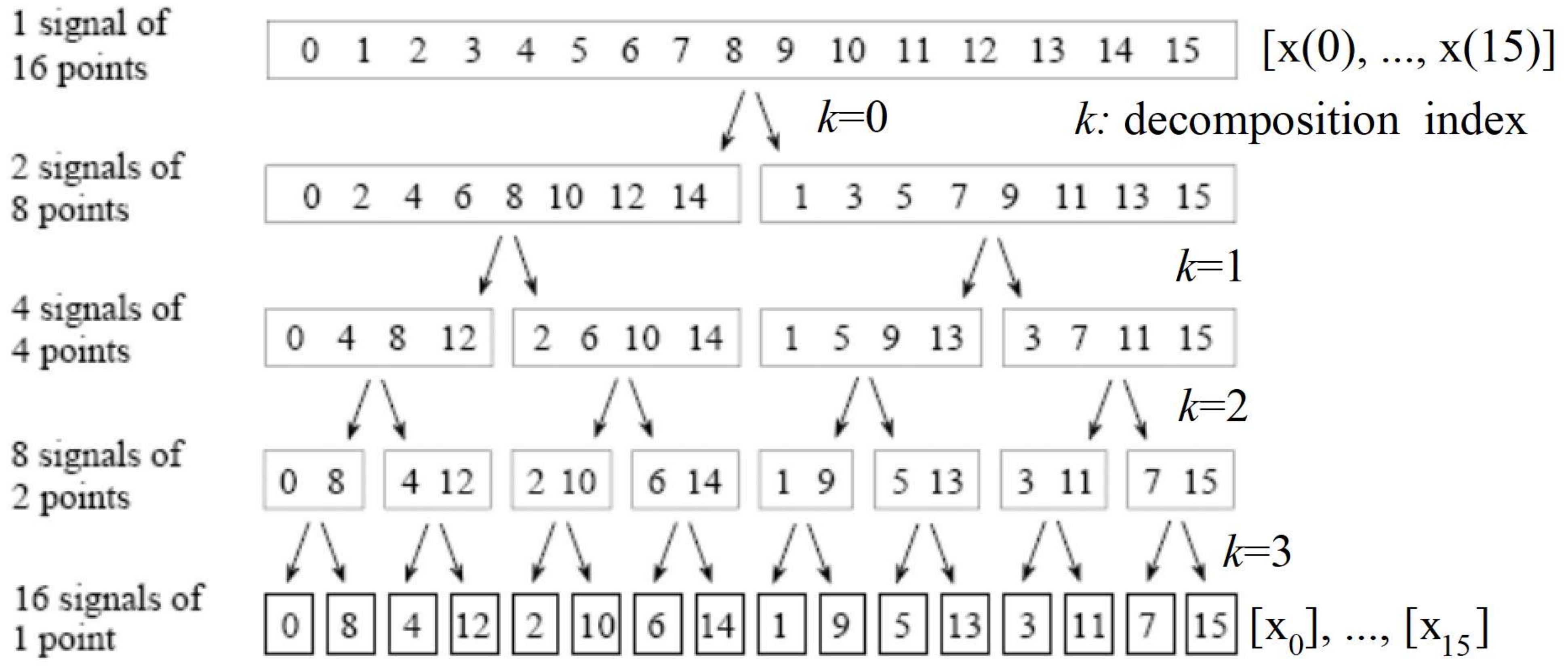


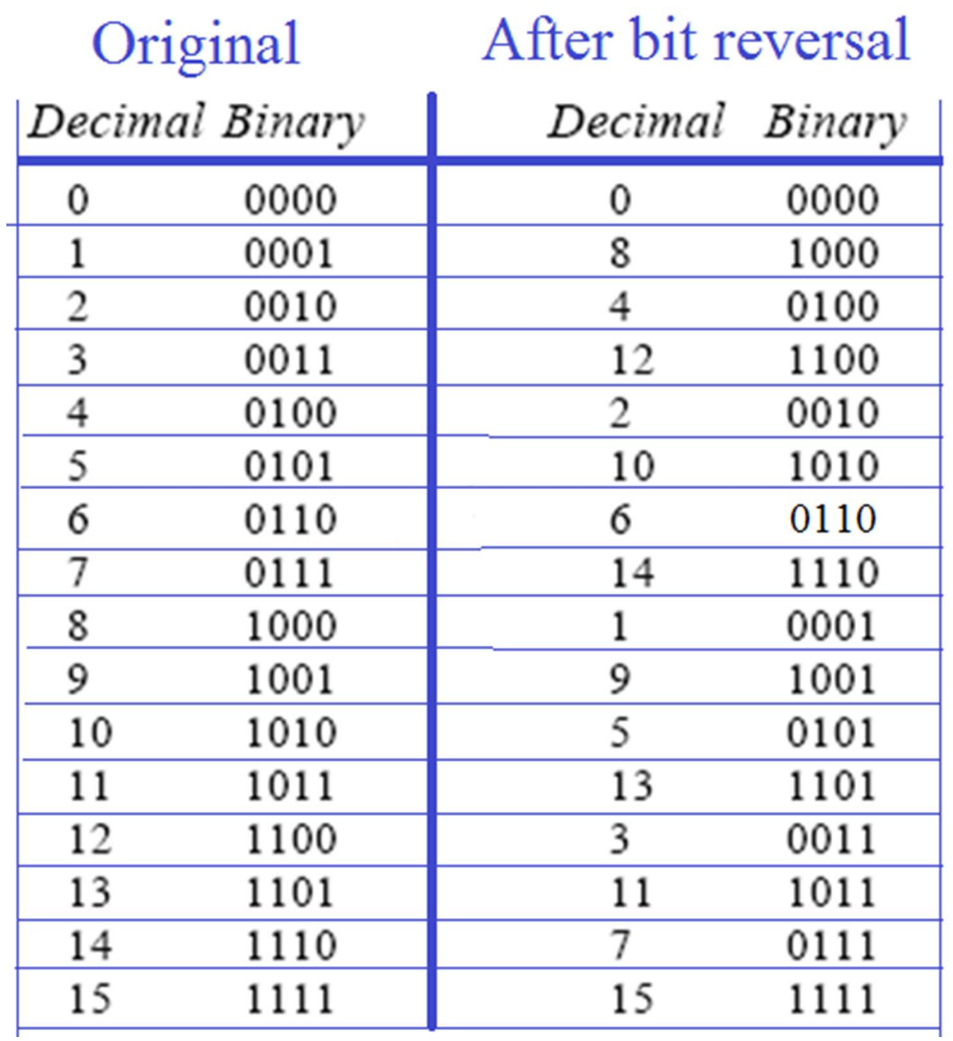
Fig. 5. Example of FFT decomposition (N=16). An *N* point signal is decomposed into *N* signals each containing a single point. Each stage uses an ***interlace decomposition***, separating the even and odd numbered samples.

The decomposition is a ***reordering* of the samples** in the signal.

The important idea is that the **binary** numbers are the ***reversals*** of each other.

Conclusion

The FFT time domain decomposition is carried out by a **bit reversal sorting** algorithm.
This involves rearranging the order of the N time domain samples by counting in binary with the bits flipped left-for-right.



Original		After bit reversal	
Decimal	Binary	Decimal	Binary
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0110
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011
14	1110	7	0111
15	1111	15	1111

Fig. 6 The FFT bit reversal sorting. The FFT time domain decomposition can be implemented by sorting the samples according to bit reversed order.

STEP 2. Finding the frequency spectra of the **1 point** time domain signals.

The frequency spectrum of a 1 point signal is **equal to *itself***. This means that *nothing* is required to do this step. Each of the 1 point signals is now a frequency spectrum, and not a time domain signal.

STEP 3. Combining the *N* frequency spectra in the exact reverse order that the time domain decomposition took place.
Unfortunately, the bit reversal shortcut is not applicable, and we must go back one stage at a time.
Example

- In the first stage, N=16 frequency spectra (1 point each) are synthesized into 8 frequency spectra (2 points each).
- In the second stage, the 8 frequency spectra (2 points each) are synthesized into 4 frequency spectra (4 points each), and so on.
- The last stage results in the output of the FFT, a N=16 point frequency spectrum.

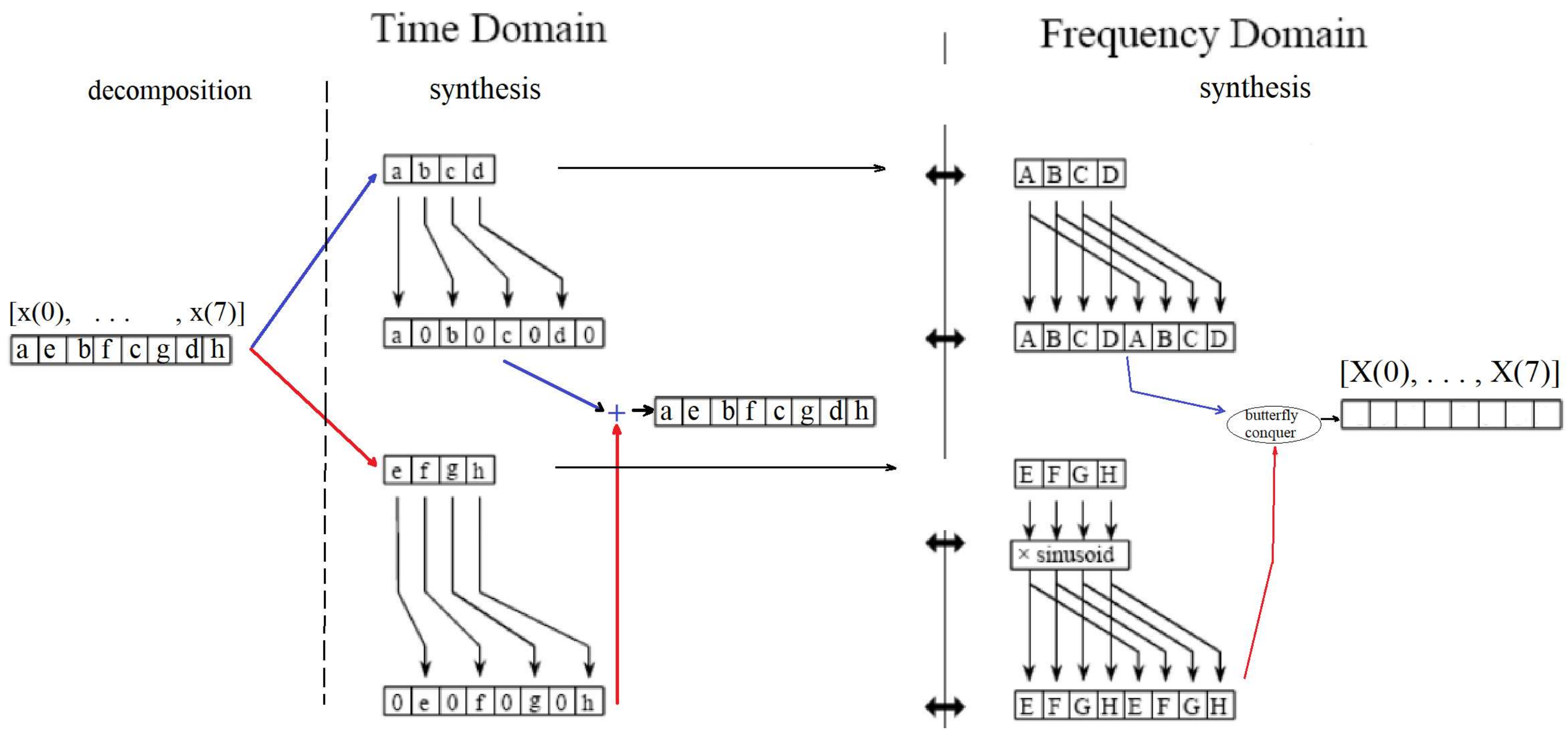
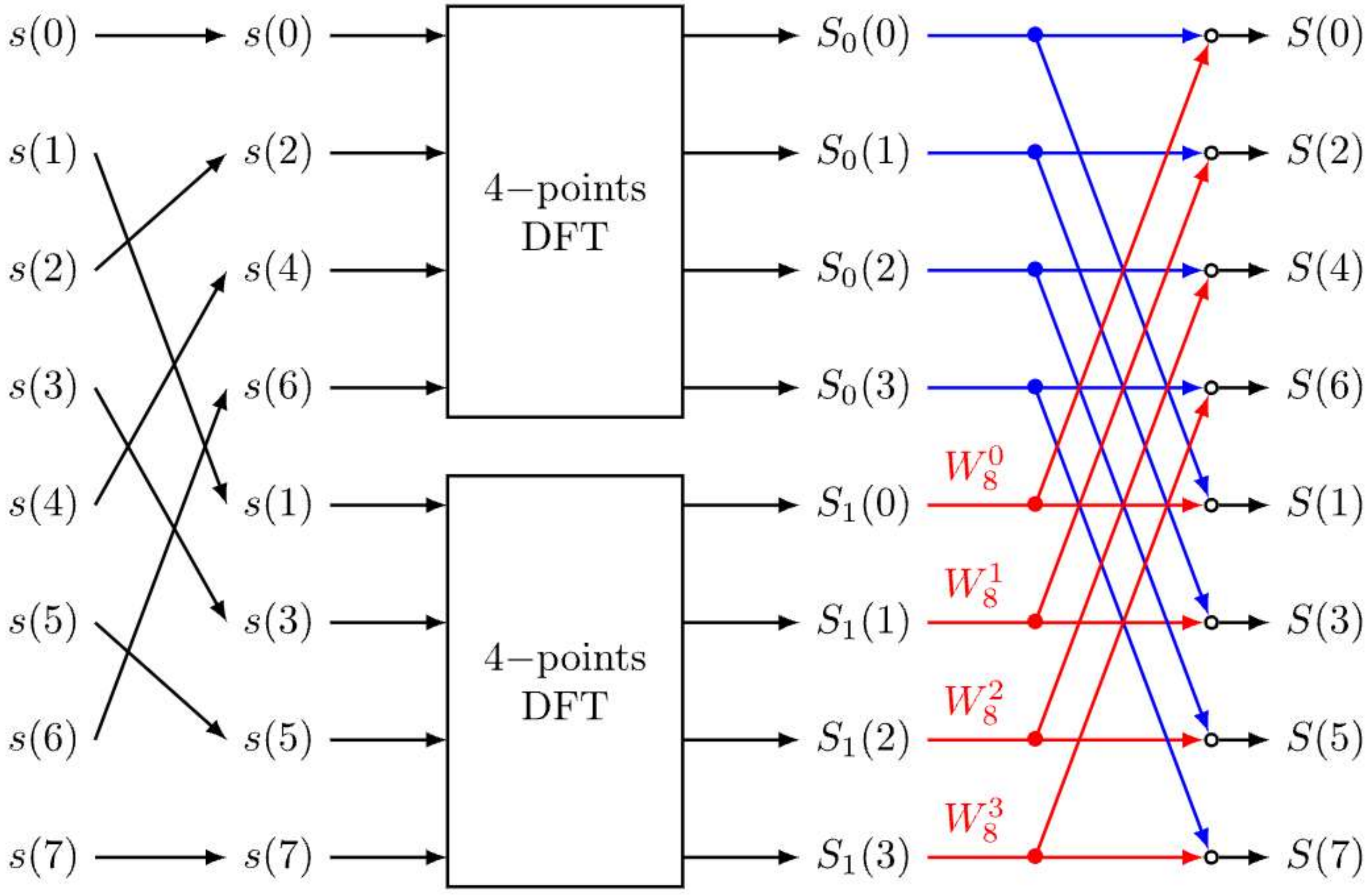


Fig. 7 Illustration of one synthesis stage in FFT (synthesis of an 8-element DFT from two 4-element DFTs). When a time domain signal is diluted with zeros, the frequency domain is duplicated. If the time domain signal is also shifted by one sample during the dilution, the spectrum will additionally be multiplied by a sinusoid.

Example of one synthesis stage (2x4 DFT → 1x8 DFT)



- **Diluting** the time domain with zeros corresponds to a ***duplication*** of the frequency spectrum. Therefore, the frequency spectra are combined in the FFT by duplicating them, and then adding the duplicated spectra together.

- In order to match up when added, the two time domain signals are diluted with zeros in a slightly different way:
  - In one signal, the *odd points* are zero, while in the other signal, the *even points* are zero. In other words, one of the time domain signals is shifted to the right by one sample.
  - This time domain shift corresponds to **multiplying** the spectrum by a ***sinusoid***.

Recall that a shift in the time domain is equivalent to convolving the signal with a shifted delta function. This multiplies the signal's spectrum with the spectrum of the shifted delta function. The spectrum of a shifted delta function is a sinusoid.

The shift (*twiddle*) factors are:

$$W_N^k = (m_N)^k = e^{-i(2\pi/N)\cdot k}, \qquad k = 0, \dots, N-1$$

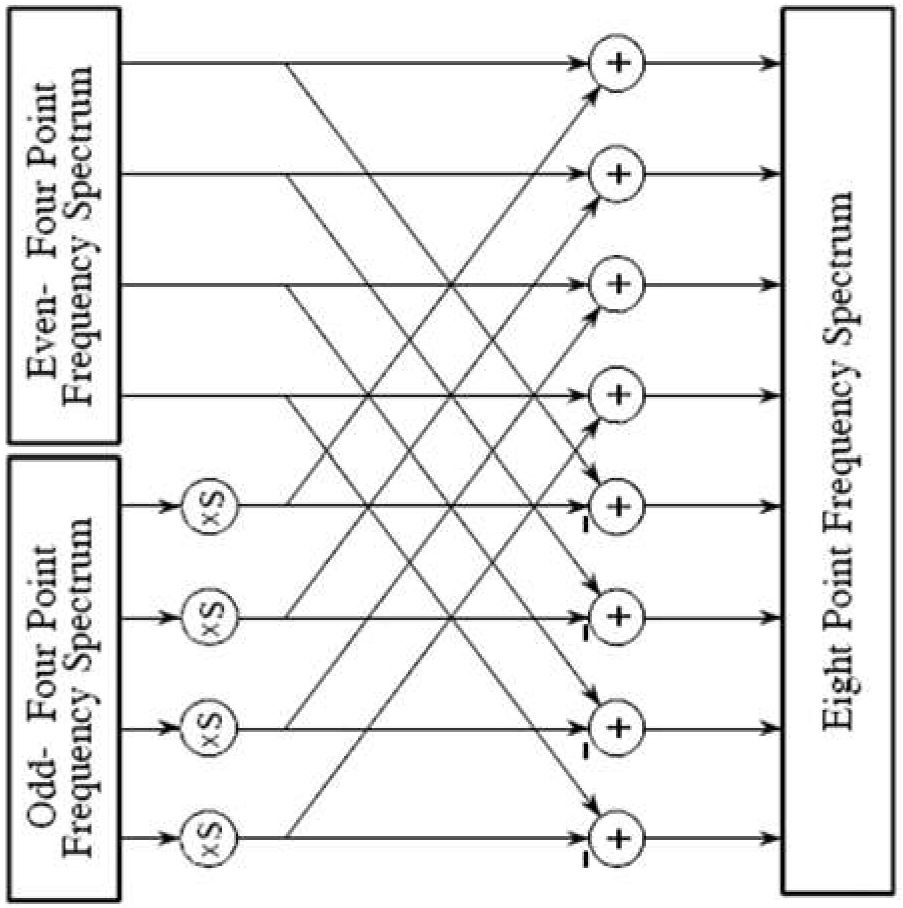

Even- Four Point Frequency Spectrum
Odd- Four Point Frequency Spectrum
xS
+
-
Eight Point Frequency Spectrum

Fig. 8 **FFT synthesis** flow diagram. This shows the method of combining two 4 point frequency spectra into a single 8 point frequency spectrum. The ×S operation means that the signal is multiplied by a sinusoid with an appropriately selected frequency.

The butterfly is the basic computational element of the FFT, transforming two complex points into two other complex points.

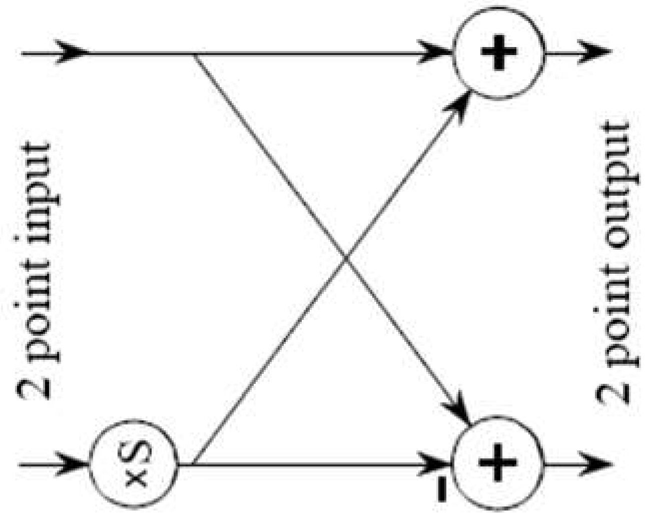
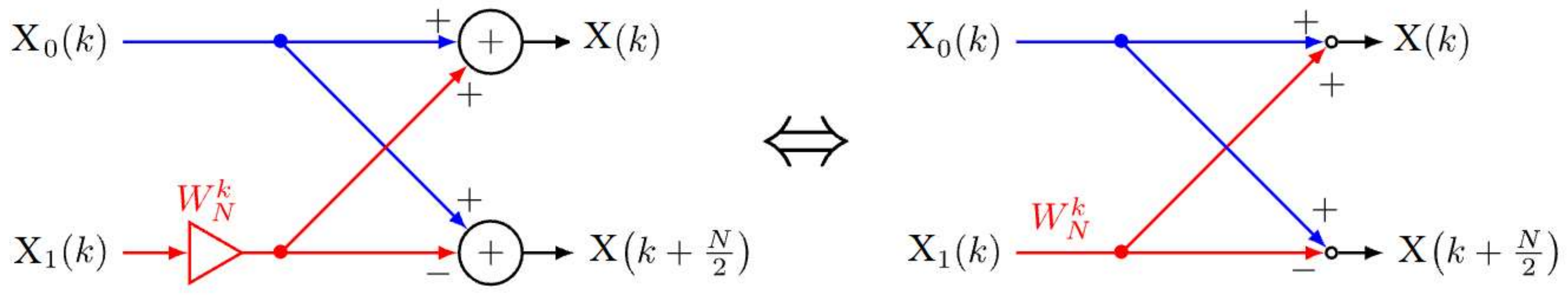


Fig. 9 The FFT butterfly.
This is the basic calculation element in the FFT, taking two complex points and converting them into two other complex points.



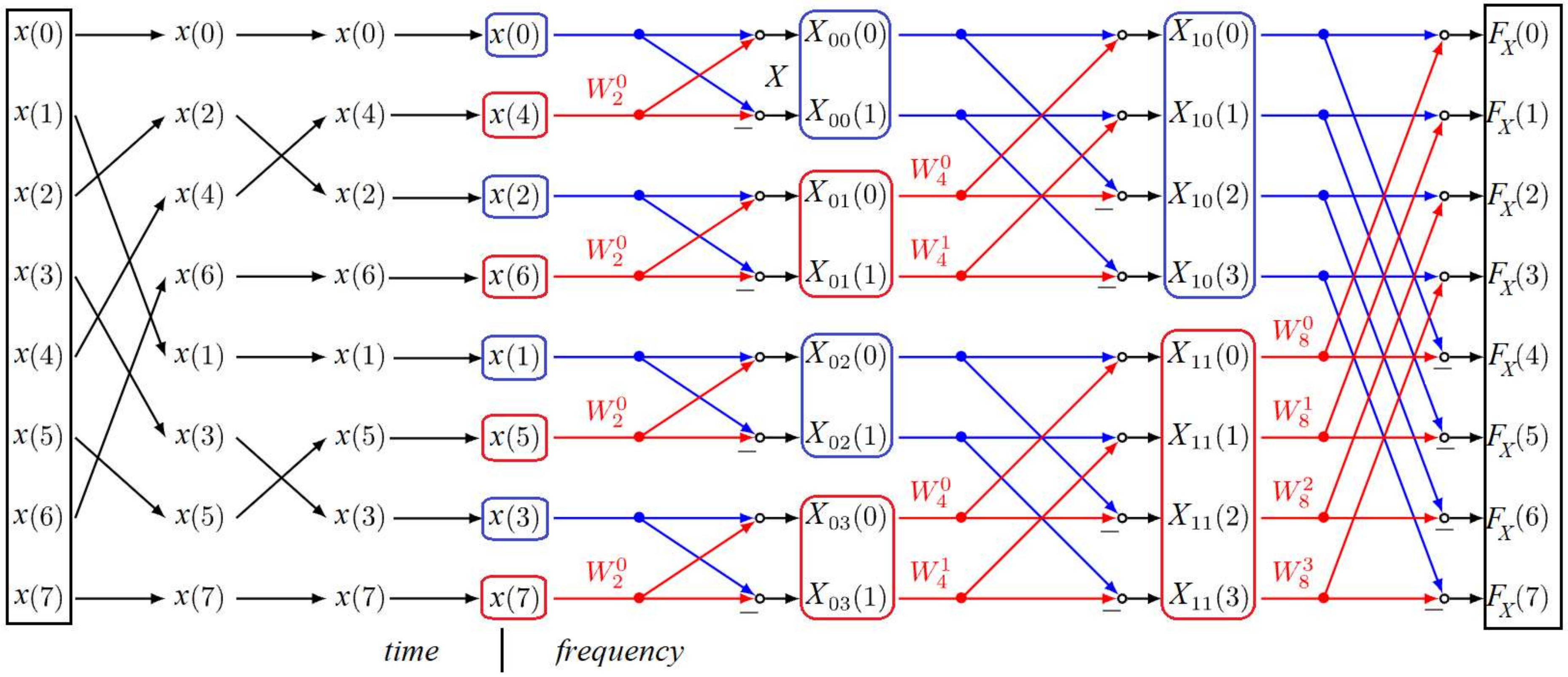

$x(0)$ $x(1)$ $x(2)$ $x(3)$ $x(4)$ $x(5)$ $x(6)$ $x(7)$
$x(0)$ $x(2)$ $x(4)$ $x(6)$ $x(1)$ $x(3)$ $x(5)$ $x(7)$
$x(0)$ $x(4)$ $x(2)$ $x(6)$ $x(1)$ $x(5)$ $x(3)$ $x(7)$
W_2^0
X
$X_{00}(0)$ $X_{00}(1)$ $X_{01}(0)$ $X_{01}(1)$ $X_{02}(0)$ $X_{02}(1)$ $X_{03}(0)$ $X_{03}(1)$
W_4^0 W_4^1
$X_{10}(0)$ $X_{10}(1)$ $X_{10}(2)$ $X_{10}(3)$ $X_{11}(0)$ $X_{11}(1)$ $X_{11}(2)$ $X_{11}(3)$
W_8^0 W_8^1 W_8^2 W_8^3
$F_X(0)$ $F_X(1)$ $F_X(2)$ $F_X(3)$ $F_X(4)$ $F_X(5)$ $F_X(6)$ $F_X(7)$
time
frequency

Fig. 10 The scheme of FFT with decimation in time, for M = $2^3 = 8$ (blue line - the component is added, red line - the component is added or subtracted).

2) Computational complexity of DFT

The DFT formula requires O(M^2) operations, whereas the **Fast Fourier Transform** (FFT) will be of complexity O(M * $\log_2(M)$).

Usually we use the FFT instead of DFT. However, for a small subset of the frequency spectrum (say two or three frequency samples), the DFT formula would be of practical use.

3) The FFT with decimation in frequency

In this implementation of FFT, called "**decimation in frequency**", there is a final bit **reversal sorting** performed and not the first step - the decimation in time domain.

Step 1) The M individual samples are transformed into the frequency domain.
Step 2) Butterfly computations in the frequency domain.
For $M = 2^k$ samples k iterations of the main loop are needed.
In the processing flow this corresponds to k layers.
The input consists of M samples and the output consists of the Fourier coefficients in a reverse order of binary represented indices, i.e.

$X_z^{[j]} = F_s$, for $s=0,\ldots,M$-1;

with $s = (s_{k-1} \ldots s_0)_2$ and $z = (s_0 \ldots s_{k-1})_2$.

In every iteration step the vector of M complex-valued samples is updated "in-place".

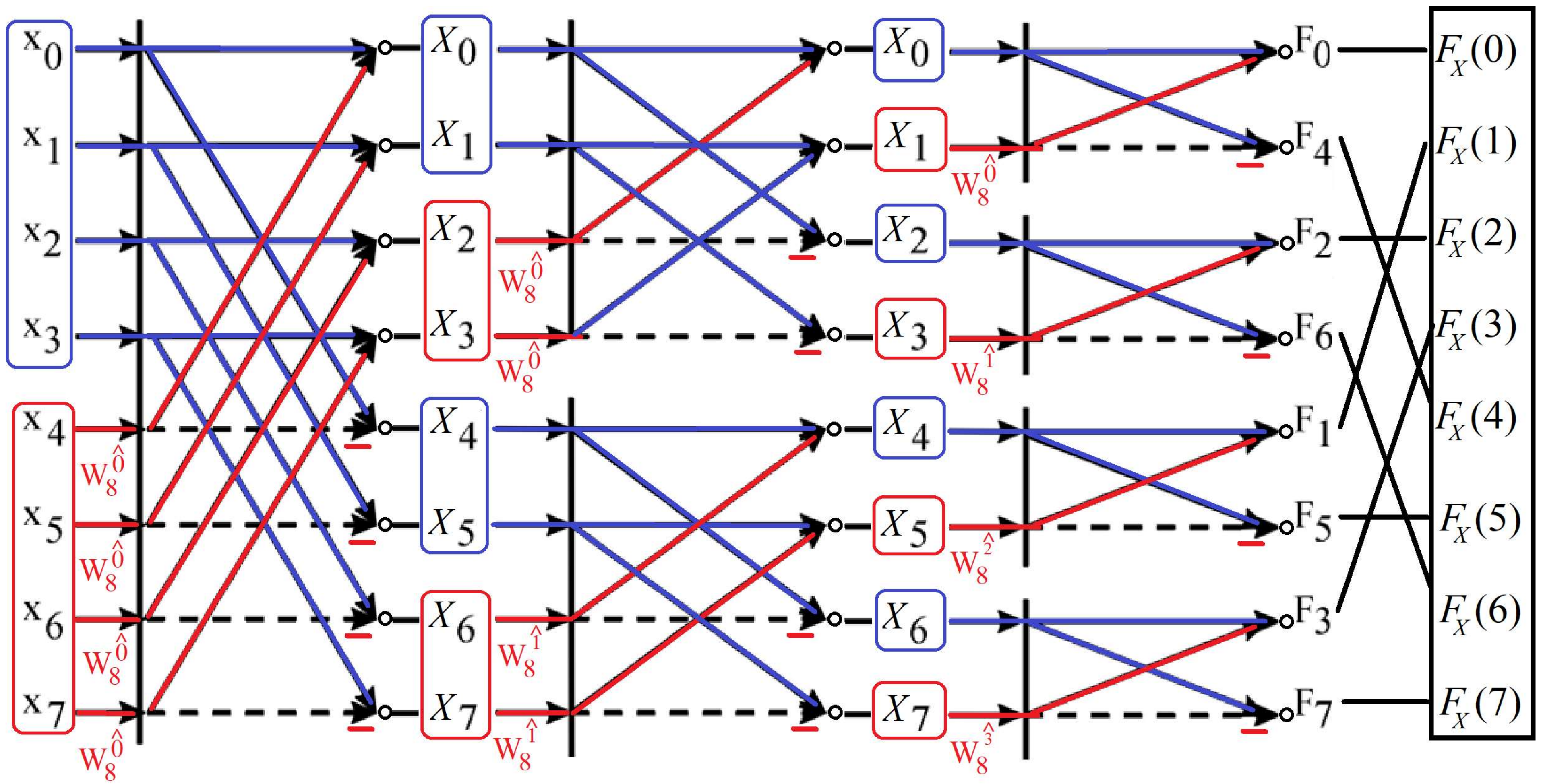


Fig. 10 The scheme of FFT with decimation in frequency, for $M = 2^3 = 8$ (solid line - the component is added, broken line - the component is subtracted).

Let $X_t^{[0]} = x_t + i \cdot 0$, for $t = 0, \dots, M-1$. - the input vector ***X*** is filled with real values of the signal, whereas the imaginary part is set to zero.

Let $j=0,1,\dots, k-1$; be the layer (main loop) index

"Butterfly pair":

- Each "butterfly pair" has two inputs $(a,b)^{[j]}$ and the same two outputs $(a,b)^{[j+1]}$. The indices of (a,b) in the j-th layer differ only by the $(k-j)$-th binary digit.
- The weights of these input links are $(-1)^{pq} m^{q\hat{u}}$. They can be pre-computed and stored in a look-up table: $M[q,u] = m^{q\hat{u}}$.
  But $M[0,u] = [1,\dots,1]$.
  It is sufficient to store, $M[1,u] = m^{\hat{u}}$, for every u-th group and element index $q=1$.

FFT loop

FOR ($j = 0, \dots , k\text{-}1$) DO

- Modify the elements of vector ***X*** as:

$$X^{[j+1]}_{(s_{k-1}\dots s_{k-j}t_{k-j-1}t_0)_2} = \sum_{t_{k-j}=0}^{1} m^{t_{k-j}(s_{k-j}\dots s_{k-1})_2 \cdot 2^{k-j}} \cdot X^{[j]}_{(s_{k-1}\dots s_{k-j+1}t_{k-j}\dots t_0)_2}$$

or given in equivalent form (with explicit indices):

$$X^{[j+1]}[u\cdot 2^{k-j} + p\cdot 2^{k-j-1} + v] = \sum_{q=0}^{1} (-1)^{pq} m^{q\hat{u}} \cdot X^{[j]}[u\cdot 2^{k-j} + q\cdot 2^{k-j-1} + v],$$

where: $\boxed{u = 0,\dots,2^{j} - 1\ ;\ v = 0,\dots,2^{k-j-1} - 1\ ;\ p = 0,1\ ;}$

and $\hat{u}$ is the reverse bit order operation on u.

Other formulation of the FFT loop:

```
FOR ( $j = 0, \ldots , k-1$ ) DO
        FOR ($u = 0, \ldots , 2^j -1$) DO (for every group)
          FOR every "butterfly" pair ( $a, b$ ) (indexed by $v$) of group $u$  DO
          begin
               $S = M[1, u]$ ;
               $X^{[j+1]}[a] = X^{[j]} [a] + S \cdot X^{[j]} [b]$;
               $X^{[j+1]}[b] = X^{[j]} [a] - S \cdot X^{[j]} [b]$;
          end
```

Reverse the bit order of binary digits:

$X_z^{[j]} \rightarrow F_s$. for $s=0,\ldots,M-1$; with $s = (s_{k-1} \ldots s_0)_2$ and $z = (s_0 \ldots s_{k-1})_2$.

4) Inverse FFT implementation

The Inverse DFT is nearly identical to the Forward DFT.
The easiest way to calculate an *Inverse FFT* is to calculate a *Forward FFT*, and then adjust the data.

INVERSE FFT (a subroutine in FORTRAN)

Input:
Let N is the number of points in the IDFT,
REX[ ] and IMX[ ] contain the real and imaginary parts of the complex frequency domain.
Output:
Upon return, RE_x[ ] and IM_x[ ] contain the complex time domain signal (from 0 to N-1).

$x \leftarrow \mathrm{IFFT}(X)$

Steps:

```
1) for K = 0 to N-1 with K++  // Change the sign of IMX[ ]
       IMX[K] = - IMX[K];

2) [ RE_x, IM_x ] = FFT([REX, IMX]); // Calculate forward FFT

3) for I = 0 to N-1 with I++ // Divide the time domain by N
       RE_x[I] = RE_x[I]/N ;
       IM_x[I] = - IM_x[I]/N ; // and change the sign of IMX[ ]

4) return [ RE_x, IM_x] ;
```

5. Efficient use of DFT

Even and odd symmetry in frequency domain

Let us assume that the signal is composed of an **even number** of samples, and that the indexes run from 0 to N-1.

- An N point signal is said to have **even symmetry** if it is a mirror image around point $N/2$. That is,

  sample $x[N/2+1]$ must be equal to $x[N/2-1]$,

  sample $x[N/2+2]$ must be equal to $x[N/2-2]$, etc.
- Similarly, **odd symmetry** occurs when the matching points have equal magnitudes but are opposite in sign, such as:

  $x[N/2+1] = -x[N/2-1]$,

  $x[N/2+2] = -x[N/2-2]$, etc.

Example. **A real-valued signal** (Fig. 11) vs. an **imaginary-valued** signal (Fig. 12).

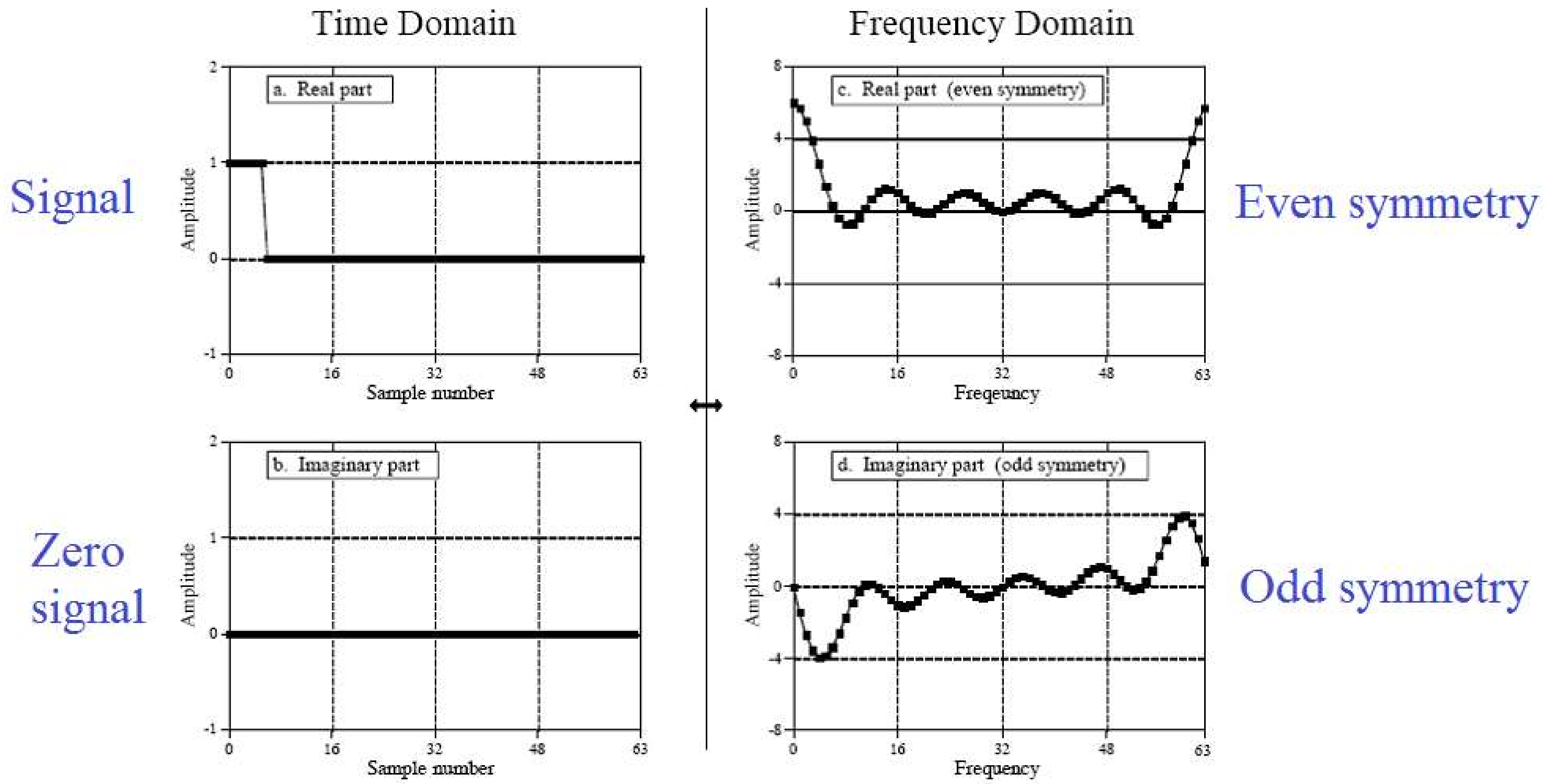


Fig. 11 **Real-valued** signal and its frequency-domain **symmetry**. A time-domain signal that consists of **real part only** has in the frequency domain: **even symmetry** of the real part and **odd symmetry** of the imaginary part.

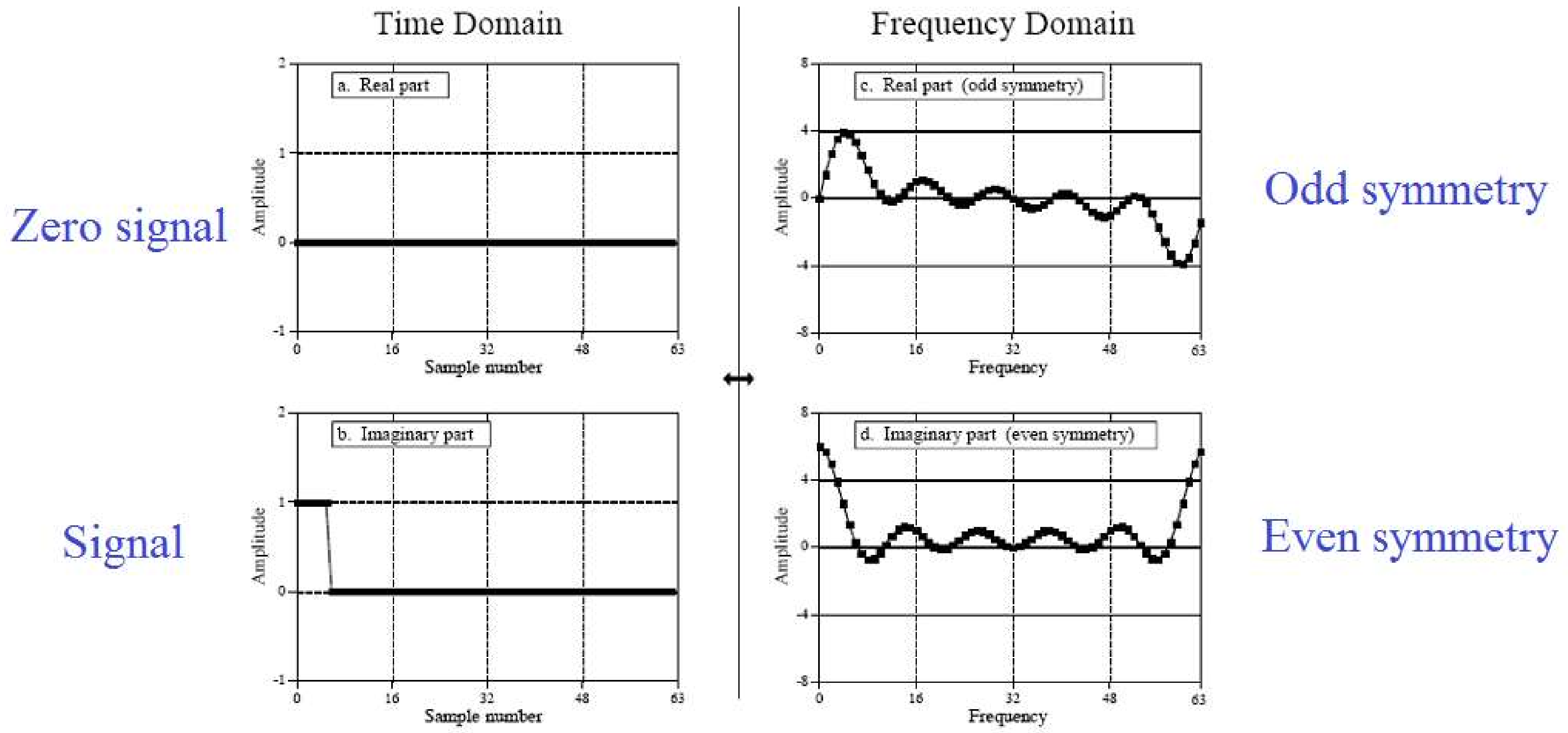


Fig. 12. **Imaginary** signal and its frequency-domain **symmetry**. The pulse is in the imaginary part of the time domain, and the real part has all zeros. The symmetry in the frequency domain is *reversed*: the **real part is odd**, while the **imaginary part is even**.

Two real DFT in a single FFT

Let there is a signal in ***both parts*** of the time domain. By *additivity*, the frequency domain will be **the *sum*** of the two signal’s frequency spectra.

The key element:

- A frequency spectrum composed of these **two types of symmetry** can be perfectly separated into the two component signals.
- This is achieved by the **even/odd decomposition**.

In other words, **two real DFT's** can be calculated for the price of a single FFT. One of the signals is placed in the real part of the time domain, and the other signal is placed in the imaginary part.

After calculating the complex DFT (e.g. via the FFT), the spectra are separated using the **even/odd decomposition**.

The **even/odd decomposition**, breaks a signal into **two** component signals, one having **even symmetry** and the other having **odd symmetry**.

The decomposition is calculated from the relations:

$$\boxed{\begin{aligned} x_{even}[n] &= \frac{x[n]+x[N-n]}{2} \\ x_{odd}[n] &= \frac{x[n]-x[N-n]}{2} \end{aligned}}$$

The zero-th samples in the even and odd signals are calculated as:

$$x_{even}[0] = x[0],$$
$$x_{odd}[0] = 0.$$

- This decomposition is part of **circular symmetry**: viewing the *end* of the signal as connected to the *beginning* of the signal, i.e. point $x[N\text{-}1]$ is next to point $x[0]$.
- When two or more signals need to be passed through the FFT, this technique reduces the execution time by about 40%.

Real DFT in half-size FFT

- The input signal is broken in half by using an interlaced decomposition. The $N/2$ even points are placed into the real part of the time domain signal, while the $N/2$ odd points go into the imaginary part.
- An $N/2$ point FFT is then calculated, requiring ca. one-half the time as an N point FFT.
- The resulting frequency domain is then separated by the even/odd decomposition, resulting in the frequency spectra of the two interlaced time domain signals.
- These two frequency spectra are then combined **into a single spectrum**, just as in the last synthesis stage of the FFT (with initial decimation in time-domain).

Exercises 5

Exercise 5.1

Prove the following statements:

A) "The DFT transformation matrix $\boldsymbol{D}_M$ is a symmetric and unitary matrix (i.e. $\boldsymbol{D}_M \boldsymbol{D}_M^* = M\,\mathbf{I}$)."

B) "If the input signal $\boldsymbol{x}(t)$ consists only of a real part, then the vector F of Fourier coefficients is a symmetric conjugate around M/2, i.e.

$$F_{M/2-k} = F^*_{M/2+k}\ , \text{ for } k = 0, \ldots, M/2 - 1.$$ "

C) "Furthermore for a real-valued $\boldsymbol{x}(t)$ if M is an even number then F_0 and $F_{M/2}$ are real numbers too."

Exercise 5-2

Simulate the steps of FFT computation by the wide-butterfly method for M=8 and the following input signal $x(t)$:

t	0	1	2	3	4	5	6	7
$x[t]$	20	10	5	5	5	0	-10	-10

Exercise 5.3

Prove the correctness of the Inverse FFT implementation, by using the straight FFT code, i.e. that:

$$\mathrm{IFFT}(X) = [\mathrm{FFT}(X^*)]^*$$