Chapter 4
Integer Fast Fourier Transform

4.1 Introduction

Since the floating-point operation is very expensive, numbers are quantized to a fixed number of bits. The number of bits at each internal node in the implementation of FFT is fixed to a certain number of bits. Denote this number as N_n. The most significant bits (MSBs) of the result after each operation is kept up to N_n bits, and the tail is truncated. Thus this conventional fixed-point arithmetic affects the invertibility of the DFT because DFT coefficients are quantized.

Integer fast Fourier transform (IntFFT) is an integer approximation of the DFT [I-6]. The transform can be implemented by using only bit shifts and additions but no multiplications. Unlike the fixed-point FFT (FxpFFT), IntFFT is power adaptable and reversible. IntFFT has the same accuracy as the FxpFFT when the transform coefficients are quantized to a certain number of bits. Complexity of IntFFT is much lower than that of FxpFFT, as the former requires only integer arithmetic.

Since the DFT has the orthogonality property, the DFT is invertible. The inverse is just the complex conjugate transpose. Fixed-point arithmetic is often used to implement the DFT in hardware. Direct quantization of the coefficients affects the invertibility of the transform. The IntFFT guarantees the invertibility/perfect-reconstruction property of the DFT while the coefficients can be quantized to finite-length binary numbers.

Lifting factorization can replace the 2×2 orthogonal matrices appearing in fast structures to compute the DFT of input with length of $N = 2^n$ for n an integer such as split-radix, radix-2 and radix-4. The resulting transforms or IntFFTs are invertible, even though the lifting coefficients are quantized and power-adaptable, that is, different quantization step sizes can be used to quantize the lifting coefficients.

K.R. Rao et al., *Fast Fourier Transform: Algorithms and Applications*,
Signals and Communication Technology,
DOI 10.1007/978-1-4020-6629-0_4, © Springer Science+Business Media B.V. 2010

4.2 The Lifting Scheme

The lifting scheme is used to construct wavelets and perfect reconstruction (PR) filter banks [I-1, I-4, I-6]. Biorthogonal filter banks having integer coefficients can be easily implemented and can be used as integer-to-integer transform.

The two-channel system in Fig. 4.1 shows the lifting scheme. The first branch is operated by A_0 and called *dual lifting* whereas the second branch is operated by A_1 and is called *lifting*. We can see that the system is PR for any choices of A_0 and A_1. It should be noted that A_0 and A_1 can be nonlinear operations like rounding or flooring operations, etc. Flooring a value means rounding it to the nearest integer less than or equal to the value.

4.3 Algorithms

Integer fast Fourier transform algorithm approximates the twiddle factor multiplication [I-6, I-7]. Let $x = x_r + jx_i$ be a complex number. The multiplication of x with a twiddle factor $W_N^{-k} = \exp\left(\frac{j2\pi k}{N}\right) = e^{j\theta} = \cos\theta + j\sin\theta$, is the complex number, $y = W_N^{-k}x$ and can be represented as

$$y = (1,\, j)\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x_r \\ x_i \end{bmatrix} = (1,\, j)[R_\theta]\begin{bmatrix} x_r \\ x_i \end{bmatrix} \tag{4.1}$$

where

$$[R_\theta] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.2}$$

The main difficulty in constructing a multiplier-less or integer transform by using the sum-of-powers-of-two (SOPOT) representation of $[R_\theta]$ is that *once the entries of* $[R_\theta]$ *are rounded to the SOPOT numbers, the entries of its inverse cannot be represented by the SOPOTs* (IntFFT covered in the first half of this chapter

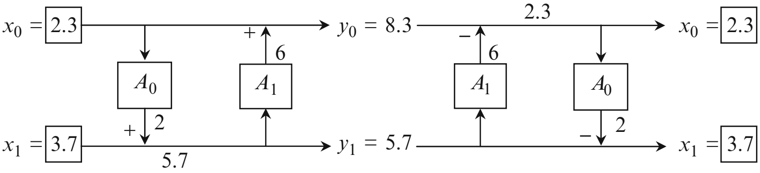

Fig. 4.1 Lifting scheme guarantees perfect reconstruction for any choices of A_0 and A_1. Perfect reconstruction means the final output equals to the input. Here A_0 and A_1 are rounding operators. [I-6] © 2002 IEEE

resolves this difficulty). In other words, if $\cos\theta$ and $\sin\theta$ in (4.1) are quantized and represented as α and β in terms of SOPOT coefficients, then an approximation of $[R_\theta]$ and its inverse can be represented as

$$[\tilde{R}_\theta] = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \tag{4.3}$$

$$[\tilde{R}_\theta]^{-1} = \frac{1}{\sqrt{\alpha^2+\beta^2}} \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \tag{4.4}$$

As α and β are SOPOT coefficients, the term $\sqrt{\alpha^2+\beta^2}$ cannot in general be represented as SOPOT coefficient. The basic idea of the integer or multiplier-less transform is to decompose $[R_\theta]$ into three lifting steps.

If $\det([A]) = 1$ and $c \neq 0$ [I-4],

$$[A] = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & (a-1)/c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \begin{bmatrix} 1 & (d-1)/c \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

From (4.5), $[R_\theta]$ is decomposed as

$$\begin{aligned} [R_\theta] &= \begin{bmatrix} 1 & \dfrac{\cos\theta-1}{\sin\theta} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & \dfrac{\cos\theta-1}{\sin\theta} \\ 0 & 1 \end{bmatrix} = [R_1][R_2][R_3] \\ &= \begin{bmatrix} 1 & -\tan\left(\dfrac{\theta}{2}\right) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan\left(\dfrac{\theta}{2}\right) \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{4.6}$$

$$\begin{aligned} [R_\theta]^{-1} &= [R_3]^{-1}[R_2]^{-1}[R_1]^{-1} \\ &= \begin{bmatrix} 1 & -\dfrac{\cos\theta-1}{\sin\theta} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\dfrac{\cos\theta-1}{\sin\theta} \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{4.7}$$

The coefficients in the factorization of (4.6) can be quantized to SOPOT coefficients to form

$$[R_\theta] \approx [S_\theta] = \begin{bmatrix} 1 & \alpha_\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta_\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & \alpha_\theta \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

where α_θ and β_θ are respectively SOPOT approximations to $(\cos\theta - 1)/\sin\theta$ and $\sin\theta$ having the form

$$\alpha_\theta = \sum_{k=1}^{t} a_k\, 2^{b_k} \tag{4.9}$$

where $a_k \in \{-1, 1\}$, $b_k \in \{-r, \ldots, -1, 0, 1, \ldots, r\}$, r is the range of the coefficients and t is the number of terms used in each coefficient. The variable t is usually limited so that the twiddle factor multiplication can be implemented with limited number of addition and shift operations. The integer FFT converges to the DFT when t increases.

The lifting structure has two advantages over the butterfly structure. First, the number of real multiplications is reduced from four to three, although the number of additions is increased from two to three (see Figs. 4.2 and 4.3). Second, the structure allows for quantization of the lifting coefficients and the quantization does not affect the PR property. To be specific, instead of quantizing the elements of $[R_\theta]$ in (4.2) directly, the lifting coefficients, s and $(s-1)/c$ are quantized and therefore, the inversion also consists of three lifting steps with the same lifting coefficients but with opposite signs.

Example 4.1 In case of the twiddle factor, W_8^1, $\theta = -\pi/4$, $(\cos\theta - 1)/\sin\theta = \sqrt{2} - 1$ and $\sin\theta = -1/\sqrt{2}$. If we round these numbers respectively to the right-hand one digit of the decimal point, then $\alpha_\theta = 0.4$ and $\beta_\theta = 0.7$.

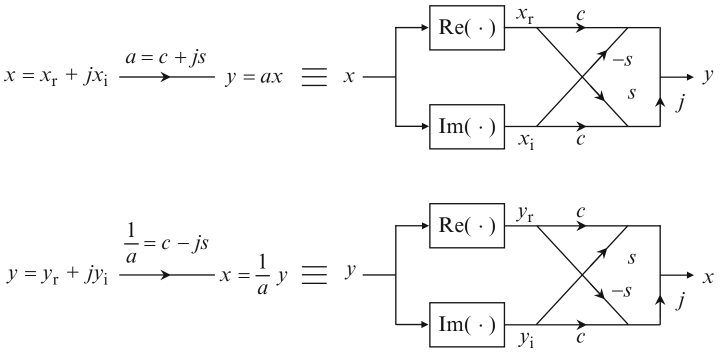

Fig. 4.2 Butterfly structure for implementing a complex multiplication above and its inverse below where $s = \sin\theta$ and $c = \cos\theta$. [I-6] © 2002 IEEE

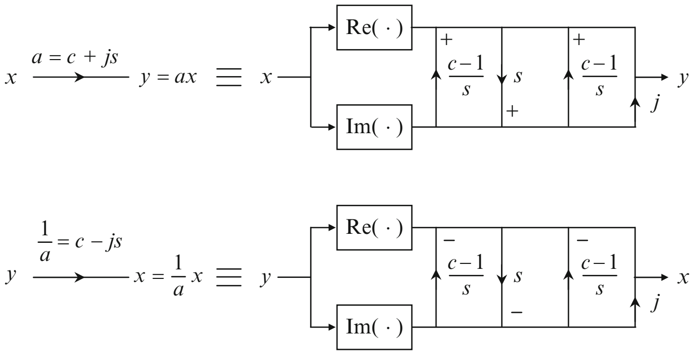

Fig. 4.3 Lifting structure for implementing a complex multiplication above and its inverse below where $s = \sin\theta$ and $c = \cos\theta$. [I-6] © 2002 IEEE

$$[S_\theta] = \begin{bmatrix} 1 & 0.4 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -0.7 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.4 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.72 & 0.688 \\ -0.7 & 0.72 \end{bmatrix} \tag{4.10a}$$

$$[S_\theta]^{-1} = \begin{bmatrix} 1 & -0.4 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.7 & 1 \end{bmatrix} \begin{bmatrix} 1 & -0.4 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.72 & -0.688 \\ 0.7 & 0.72 \end{bmatrix} \tag{4.10b}$$

The lifting scheme defined in (4.8) and (4.10) works for any numbers (real and complex) of α_θ and β_θ. $[S_\theta]$ is no more orthogonal ($[S_\theta]^{-1} \neq [S_\theta]^T$), but developing its inverse transform is as easy as for the case of the orthogonal transform as the entries of $[S_\theta]$ and $[S_\theta]^{-1}$ are the same with different signs except 1s on the diagonal (Figs. 4.2 and 4.3), while both schemes guarantee perfect inverse or perfect reconstruction as $[S_\theta]^{-1}[S_\theta] = [I]$ (biorthogonal) and $[R_\theta]^T[R_\theta] = [I]$ (orthogonal, see [4.2]), respectively.

In summary, in implementing a complex number multiplication, a twiddle factor in matrix form has a butterfly structure and if we round the coefficients, its inverse is computationally complex, but if we decompose the twiddle factor into a lifting structure, the twiddle factor has a perfect inverse even if we round the coefficients. Once the coefficients are rounded in the lifting structure, the twiddle factor may have either a lifting or butterfly structures for perfect inverse but the lifting structure has one less multiplication.

An eight-point integer FFT based on the split-radix structure is developed in [I-6]. Figure 4.4 shows the lattice structure of the integer FFT, where the twiddle factors W_8^1 and W_8^3 are implemented using the factorization. Another integer FFT based on the radix-2 decimation-in-frequency is covered in [I-7]. At the expense of precision, we can develop computationally effective integer FFT algorithms.

When an angle is in I and IV quadrants, (4.6) is used. If $\theta \in (-\pi, -\pi/2) \cup (\pi/2, \pi)$, then $|(\cos\theta - 1)/\sin\theta| > 1$ as $\cos\theta < 0$ for II and III quadrants. Thus the absolute values of the lifting coefficients need to be controlled to be less than or equal to one by replacing R_θ by $-[R_{\theta+\pi}]$ as follows:

$$\begin{aligned}[R_\theta] = -[R_{\theta+\pi}] &= -\begin{bmatrix} -\cos\theta & \sin\theta \\ -\sin\theta & -\cos\theta \end{bmatrix} \\ &= -\begin{bmatrix} 1 & (c+1)/s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -s & 1 \end{bmatrix} \begin{bmatrix} 1 & (c+1)/s \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{4.11}$$

When an angle is in I and II quadrants, we can have another choice of lifting factorization as follows:

$$\begin{aligned}[R_\theta] &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & (s-1)/c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \begin{bmatrix} 1 & (s-1)/c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{aligned} \tag{4.12}$$

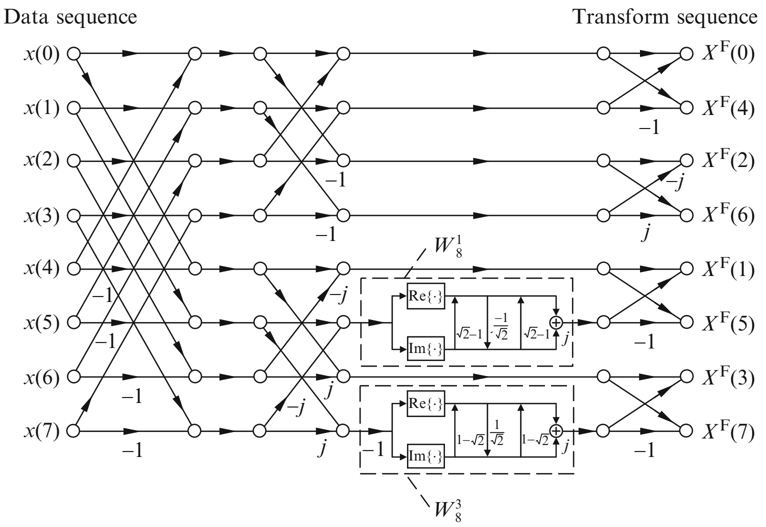

Fig. 4.4 Lattice structure of eight-point integer FFT using split-radix structure (see also [I-9]). For example, the twiddle factors are quantized/rounded off in order to be represented as 16-bit numbers (N_c bits). Multiplication results are again uniformly quantized to N_n bits. N_n the number of bits required to represent internal nodes is determined only by FFT of size N and N_i, the number of bits required to represent input signal. [I-6] © 2002 IEEE

Table 4.1 For each value of θ, only two out of four possible lifting factorizations have all their lifting coefficients falling between −1 and 1. [I-6] © 2002 IEEE

Quadrant	Range of θ	Lifting factorization
I	(0, π/2)	(4.6) and (4.13)
II	(π/2, π)	(4.11) and (4.13)
III	(−π, −π/2)	(4.11) and (4.15)
IV	(−π/2, 0)	(4.6) and (4.15)

However, if $\theta \in (-\pi, 0)$, then $\sin\theta < 0$ for III and IV quadrants and $(\sin\theta - 1)/\cos\theta$ will be greater than one. Thus $[R_\theta]$ should be replaced by $-[R_{\theta+\pi}]$ as follows (Table 4.1):

$$[R_\theta] = -\begin{bmatrix} -\cos\theta & \sin\theta \\ -\sin\theta & -\cos\theta \end{bmatrix} = -\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} -\sin\theta & \cos\theta \\ -\cos\theta & -\sin\theta \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$
$$= -\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & (s+1)/c \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -c & 1 \end{bmatrix}\begin{bmatrix} 1 & (s+1)/c \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{4.13}$$

For example, suppose we are given the twiddle factor, $W_8^3 = e^{-j6\pi/8}$. Then $\theta = -3\pi/4$. Then we have the two options, (4.11) and (4.13). We select (4.11). Then

$$[R_\theta] = -[R_{\theta+\pi}] = -\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$
$$= -\begin{bmatrix} 1 & 1-\sqrt{2} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 1/\sqrt{2} & 1 \end{bmatrix}\begin{bmatrix} 1 & 1-\sqrt{2} \\ 0 & 1 \end{bmatrix} \tag{4.14}$$

Substitute (4.14) in (4.1) to obtain a lifting structure for a multiplication of a complex number and the twiddle factor W_8^3. Figure 4.4 shows lifting/lattice structure of eight-point integer FFT using split-radix structure, where the two twiddle factors W_8^1 and $W_8^3(= -W_8^7)$ are implemented using lifting scheme. Inverse integer FFT is as usual the conjugate of the integer FFT whose block diagram is shown in Fig. 4.4.

4.3.1 Fixed-Point Arithmetic Implementation

One of the factors that primarily affects the cost of the DSP implementation is the resolution of the internal nodes (the size of the registers at each stage). In practice, it is impossible to retain infinite resolution of the signal samples and the transform coefficients. Since the floating-point operation is very expensive, these numbers are often quantized to a fixed number of bits. Two's-complement arithmetic for fixed-point representation of numbers is a system in which negative numbers are represented by the two's complement of the absolute value; this system represents signed integers on hardware, DSP and computers (Table 4.2).

Each addition can increase the number of bits by one, whereas each multiplication can increase the number of bits by $2n$ bits for the multiplication of two n-bit numbers. The nodes in latter stages require more bits than those in earlier stages to store the output after each arithmetic operation without overflows. As a result, the number of bits required to store the results grows cumulatively as the number of stages increases. In general, the number of bits at each internal node is fixed to a certain number of bits. The most significant bit (MSB) of the result after each operation will be kept up to a certain number of bits for each internal node, and the tail will be truncated. However, this conventional fixed-point arithmetic affects the

Table 4.2 Four-bit two's-complement integer. Four bits can represent values in the range of −8 to 7

Two's complement		Decimal
0	111	7
0	110	6
0	001	1
0	000	0
1	111	−1
1	001	−7
1	000	−8

The sign of a number (given in the first column) is encoded in the most significant bit (MSB) invertibility of the transform because the DFT coefficients are quantized. Lifting scheme is a way to quantize the DFT coefficients that preserves the invertibility property [I-6].

The integer FFT and fixed-point FFT are compared in noise reduction application (Table 4.3). At low power, i.e., the coefficients are quantized to low resolution, the IntFFT yields significantly better results than the FxpFFT, and they yield similar results at high power [I-6].

While for two and higher dimensions, the row-column method, the vector-radix FFT and the polynomial transform FFT algorithms are commonly used fast algorithms for computing multidimensional discrete Fourier transform (M-D DFT). The application of the integer approach to the polynomial transform FFT for the ($N \times N$) two-dimensional integer FFT is described in [I-34] using radix-2. The proposed method can be readily generalized to the split vector-radix and row-column algorithms.

The radix-2^2 algorithm is characterized by the property that it has the same complex multiplication computational complexity as the radix-4 FFT algorithm, but still retains the same butterfly (BF) structures as the radix-2 FFT algorithm (Table 4.4). The multiplicative operations are in a more regular arrangement as the non-trivial multiplications appear after every two BF stages. This spatial regularity provides a great advantage in hardware implementation if pipeline behavior is taken into consideration.

In the widely used OFDM systems [O2], the inverse DFT and DFT pair are used to modulate and demodulate the data constellation on the sub-carriers. The input to the IDFT at the transmitter side is a set of digitally modulated signals. Assuming the 64-QAM scheme is adopted, then the input levels are ±1, ±3, ±5, and ±7, which

Table 4.3 Computational complexities (the numbers of real multiplies and real adds) of the split-radix FFT and its integer versions (FxpFFT and IntFFT) when the coefficients are quantized/rounded off to $N_c = 10$ bits at each stage. [I-6] © 2002 IEEE

N	Split-radix FFT		FxpFFT		IntFFT	
	Multiplies	Adds	Adds	Shifts	Adds	Shifts
16	20	148	262	144	202	84
32	68	388	746	448	559	261
64	196	964	1,910	1,184	1,420	694
128	516	2,308	4,674	2,968	3,448	1,742
256	1,284	5,380	10,990	7,064	8,086	4,160
512	3,076	12,292	25,346	16,472	18,594	9,720
1024	7,172	27,652	57,398	37,600	41,997	22,199

Table 4.4 Number of nontrivial complex multiplies. A set of complex multiply is three real multiplies. [I-33] © 2006 IEEE

N	Radix-2	Radix-2^2	Split-radix
16	10	8	8
64	98	76	72
256	642	492	456
1,024	3,586	2,732	2,504

can be represented by a six-bit vector. The output of the IDFT consists of the time-domain samples to be transmitted over the real channel. Accordingly, at the receiver side, the DFT is performed.

The input sequence uses 12 bits for both real and imaginary parts. The internal word length and the word length for the lifting coefficients and twiddle factors are set to 12 bits.

Based on the IntFFT, a VLSI feasible radix-2^2 FFT architecture is proposed and verified by Chang and Nguyen [I-33]. The most important feature of the IntFFT is that it guarantees the invertibility as well as provides accuracy comparable with the conventional FxpFFT. The required number of real multipliers is also reduced because the lifting scheme (LS) uses one fewer multiplier than general complex multipliers. Compared to FxpFFT designs, the system simulations prove that IntFFT-based architecture can also be adopted by OFDM systems [O2] and yield comparative bit error rate (BER) performance, even if the noisy channel is present.

4.4 Integer Discrete Fourier Transform

Integer Fourier transform approximates the DFT for the fixed-point multiplications [I-5]. The fixed-point multiplications can be implemented by the addition and binary shifting operations. For example

$$7 \times a = a \ll 2 + a \ll 1 + a$$

where a is an integer and $<<$ is a binary left-shift operator.

Two types of integer transforms are presented in this section. Forward and inverse transform matrices can be the same and different. They are referred to as *near-complete* and *complete* integer DFTs.

4.4.1 Near-Complete *Integer DFT*

Let $[F]$ be the DFT matrix, and let $[F_{\mathrm{i}}]$ be an integer DFT. Then for integer DFT to be orthogonal and, hence, be reversible, it is required that

$$[F_{\mathrm{i}}]^*[F_{\mathrm{i}}]^T = [F_{\mathrm{i}}][F_{\mathrm{i}}]^H = \operatorname{diag}(r_0, r_1, \ldots, r_7) = [C] \tag{4.15}$$

where $[F_{\mathrm{i}}]^H$ denotes the transpose of $[F_{\mathrm{i}}]^*$ and $r_l = 2^m$, where m is an integer. Thus it follows that

$$[C]^{-1}[F_{\mathrm{i}}][F_{\mathrm{i}}]^H = [I] \tag{4.16}$$

To approximate the DFT, integer DFT $[F_\mathrm{i}]$ keeps all the signs of entries of $[F]$ as follows.

$$[F_\mathrm{i}] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a_1 & a_2 - ja_2 & -ja_1 & -a_2 - ja_2 & -a_1 & -a_2 + ja_2 & ja_1 & a_2 + ja_2 \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ b_1 & -b_2 - jb_2 & jb_1 & b_2 - jb_2 & -b_1 & b_2 + jb_2 & -jb_1 & -b_2 + jb_2 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ b_1 & -b_2 + jb_2 & -jb_1 & b_2 + jb_2 & -b_1 & b_2 - jb_2 & jb_1 & -b_2 - jb_2 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ a_1 & a_2 + ja_2 & ja_1 & -a_2 + ja_2 & -a_1 & -a_2 - ja_2 & -ja_1 & a_2 - ja_2 \end{bmatrix} \tag{4.17}$$

In order for (4.15) to be satisfied, the complex inner products of the following row pairs of $[F_\mathrm{i}]$ should be zero. When Row 2 represents the second row

$$\langle \text{Row 2},\ \text{Row 6} \rangle = 0 \qquad \langle \text{Row 4},\ \text{Row 8} \rangle = 0 \tag{4.18}$$

Here a complex inner product is defined by

$$\langle \underline{z}, \underline{w} \rangle = \underline{w}^H \underline{z}$$

for complex vectors $\underline{z}$ and $\underline{w}$. The vector $\underline{w}^H$ is the transpose of $\underline{w}^*$. From (4.18)

$$a_1 b_1 = 2a_2 b_2 \Rightarrow \quad a_1 \geq a_2 \quad b_1 \geq b_2 \tag{4.19}$$

From (4.15)

$$r_0 = r_2 = r_4 = r_6 = N$$

$$r_1 = r_7 = (N/2)\ a_1^2 + N\ a_2^2$$

$$r_3 = r_5 = (N/2)\ b_1^2 + N\ b_2^2$$

Some possible choices of the parameters of eight-point integer DFT are listed in Table 4.5.

Table 4.5 Some sets of parameter values of eight-point integer DFT. [I-5] © 2000 IEEE

a_1	2	3	4	5	8	10	17	99	500
a_2	1	2	3	3	5	7	12	70	353
b_1	1	4	3	6	5	7	24	140	706
b_2	1	3	2	5	4	5	17	99	500

$[F_{\mathrm{i}}]$ Only row vectors are orthogonal.
$\Downarrow$
Thus $[F_{\mathrm{i}}][F_{\mathrm{i}}]^H =$ diagonal matrix; $[F_{\mathrm{i}}][F_{\mathrm{i}}]^H \neq$ diagonal matrix
$\Downarrow$
$[\tilde{F}_{\mathrm{i}}][\tilde{F}_{\mathrm{i}}]^H =$ diagonal matrix; $[\tilde{F}_{\mathrm{i}}]^H[\tilde{F}_{\mathrm{i}}] =$ diagonal matrix
where $[\tilde{F}_{\mathrm{i}}]$ is$[F_{\mathrm{i}}]$ normalized by the first column as defined in (4.34).
$\Downarrow$

$$[\tilde{F}_{\mathrm{i}}] = ([C]^{1/2})^{-1}[F_{\mathrm{i}}] \tag{4.20}$$

where $[C]$ is defined in (4.15). Then $[\tilde{F}_{\mathrm{i}}]^{-1} = [\tilde{F}_{\mathrm{i}}]^H$, i.e., $[\tilde{F}_{\mathrm{i}}]$ is unitary.

4.4.2 Complete *Integer DFT*

Let $[F]$ be the DFT matrix, and let $[F_{\mathrm{i}}]^T$ the transpose of (4.17) and $[IF]^*$ be the forward and inverse integer DFTs.

$$[IF] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a_3 & a_4 - ja_4 & -ja_3 & -a_4 - ja_4 & -a_3 & -a_4 + ja_4 & ja_3 & a_4 + ja_4 \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ b_3 & -b_4 - jb_4 & jb_3 & b_4 - jb_4 & -b_3 & b_4 + jb_4 & -jb_3 & -b_4 + jb_4 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ b_3 & -b_4 + jb_4 & -jb_3 & b_4 + jb_4 & -b_3 & b_4 - jb_4 & jb_3 & -b_4 - jb_4 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ a_3 & a_4 + ja_4 & ja_3 & -a_4 + ja_4 & -a_3 & -a_4 - ja_4 & -ja_3 & a_4 - ja_4 \end{bmatrix} \tag{4.21}$$

Then for integer DFT to be orthogonal and, hence, be reversible, it is required that

$$[IF]^*[F_{\mathrm{i}}]^T = \mathrm{diag}(r_0, r_1, \ldots, r_7) = [D] = \text{diagonal matrix} \tag{4.22}$$

where $r_l = 2^m$ and m is an integer. Since $[D]$ is a diagonal matrix, it follows that

$$[D]^{-1}[IF]^*[F_{\mathrm{i}}]^T = [I] \tag{4.23}$$

From the constraint of (4.22), the complex inner products of the following pairs should be zero.

$$\begin{aligned} &\langle \text{Row 2 of } [F_{\mathrm{i}}], \text{Row 6 of } [IF] \rangle = \langle \text{Row 8 of } [F_{\mathrm{i}}], \text{Row 4 of } [IF] \rangle = 0 \\ &\langle \text{Row 4 of } [F_{\mathrm{i}}], \text{Row 8 of } [IF] \rangle = \langle \text{Row 6 of } [F_{\mathrm{i}}], \text{Row 2 of } [IF] \rangle = 0 \\ &a_1 b_3 = 2a_2 b_4 \qquad a_3 b_1 = 2a_4 b_2 \end{aligned} \tag{4.24}$$

Since the inner product of the corresponding rows of $[F_{\rm i}]$ and $[IF]$ should be the power of two from the constraint of (4.22),

$$a_1a_3 + 2a_2a_4 = 2^k \qquad b_1b_3 + 2b_2b_4 = 2^h \tag{4.25}$$

$$a_1 \geq a_2 \quad b_1 \geq b_2 \quad a_3 \geq a_4 \quad b_3 \geq b_4 \tag{4.26}$$

From (4.24), we set

$$b_3 = 2a_2 \qquad b_4 = a_1 \qquad a_3 = 2b_2 \qquad a_4 = b_1 \tag{4.27}$$

Then (4.25) becomes

$$2(a_1b_2 + a_2b_1) = 2^k \qquad 2(b_1a_2 + c_2a_1) = 2^h \tag{4.28}$$

1. Choose a_1 and a_2 such that they are integers and

$$2a_2 \geq a_1 \geq a_2$$

2. Choose b_1 and b_2 such that they are integers and

$$2b_2 \geq b_1 \geq b_2 \qquad a_1b_2 + a_2b_1 = 2^n$$

where n is an integer.

3. Set a_3, a_4, b_3, b_4 as

$$b_3 = 2^{h+1}a_2 \qquad b_4 = 2^h a_1 \qquad a_3 = 2^{h+1}b_2 \qquad a_4 = 2^h b_1$$

where h is an integer.

Substitute (4.17) and (4.21) into (4.22).

$$r_0 = r_2 = r_4 = r_6 = N$$

$$r_1 = r_7 = (N/2)a_1a_3 + Na_2a_4$$

$$r_3 = r_5 = (N/2)b_1b_3 + Nb_2b_4$$

Some possible choices of the parameters of eight-point integer DFT are listed in Table 4.6. The eight-point integer DFT retains some properties of the regular DFT.

Table 4.6 Some sets of parameter values of eight-point integer DFT. [I-5] © 2000 IEEE

a_1	2	7	3	4	4	5	10
a_2	1	5	2	3	3	4	7
b_1	2	13	17	12	44	17	18
b_2	1	9	10	7	31	12	13
a_3	1	18	34	7	31	24	13
a_4	1	13	10	6	22	17	9
b_3	1	10	4	3	3	8	7
b_4	1	7	3	2	2	5	5

4.4.3 Energy Conservation

Only the rows of $[F_{\mathrm{i}}]$ are orthogonal and the columns are not.

$$[F_{\mathrm{i}}][IF]^H = [D] \tag{4.29}$$

where $[D]$ defined in (4.22) is diagonal and its entries are integers.

Let $\underline{X} = [F_{\mathrm{i}}]^T \underline{x}$ and $\underline{Y} = ([D]^{-1}[IF])^T \underline{y}$. Then the energy conservation property is as follows.

$$\underline{x}^T \underline{y}^* = \underline{X}^T \underline{Y}^* \tag{4.30}$$

Proof.

$$\underline{X}^T \underline{Y}^* = \left([F_{\mathrm{i}}]^T \underline{x}\right)^T \left([IF]^T [D]^{-1} \underline{y}\right)^* = \underline{x}^T [F_{\mathrm{i}}][IF]^H [D]^{-1} \underline{y}^* = \underline{x}^T \underline{y}^* \tag{4.31}$$

4.4.4 Circular Shift

Let

$$X^{\mathrm{i}}(k) = \text{intDFT}\,[x(n)] \qquad Y^{\mathrm{i}}(k) = \text{intDFT}\,[x(n+h)] \tag{4.32}$$

where $x(n+h)$ is defined in (2.17). Then

$$Y^{\mathrm{i}}(k) \approx X^{\mathrm{i}}(k) W_N^{-hk} \quad \text{when both } k \text{ and } h \text{ are odd} \tag{4.33a}$$

$$Y^{\mathrm{i}}(k) = X^{\mathrm{i}}(k) W_N^{-hk} \quad \text{otherwise} \tag{4.33b}$$

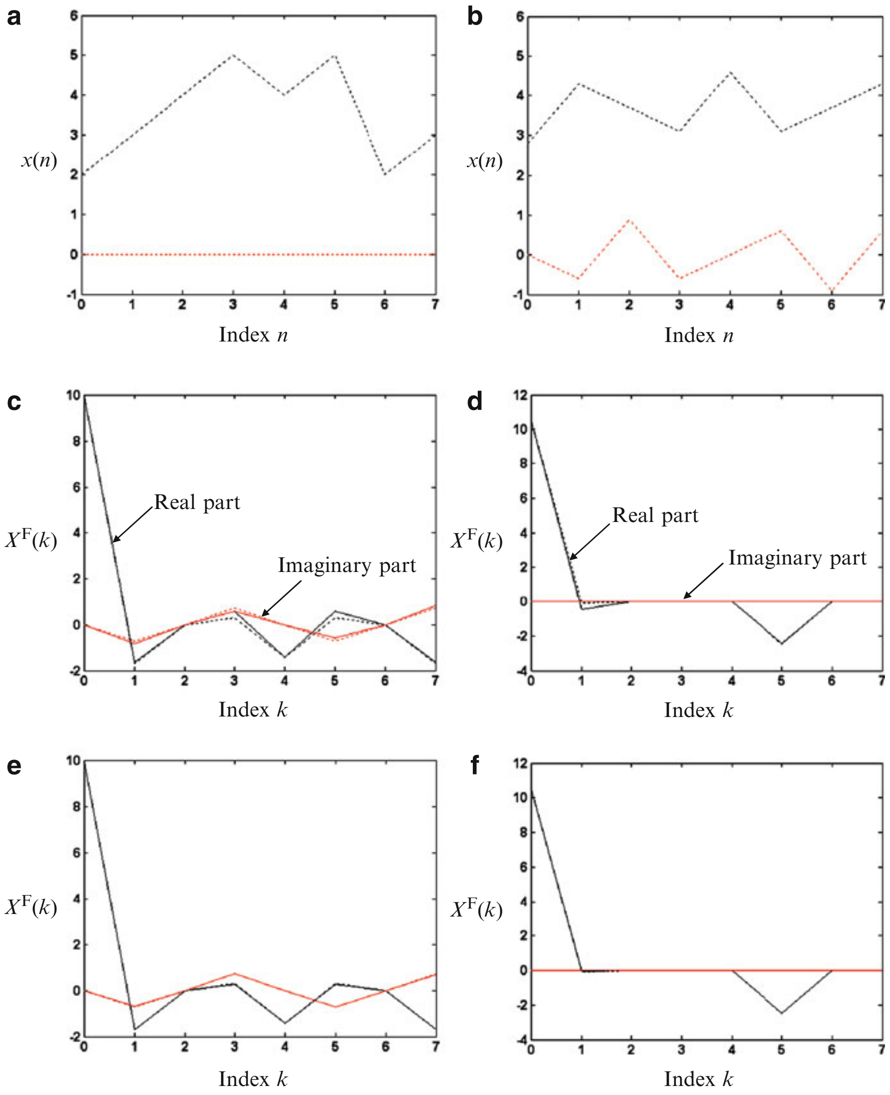

Fig. 4.5 The regular and integer DFTs of input signals are represented by *dashed* and *solid lines*. **a**, **b** Input signals, $\underline{x}_1, \underline{x}_2$. **c**, **d** *Near-complete* integer DFTs of $\underline{x}_1, \underline{x}_2$. **e**, **f** *Complete* integer DFTs of $\underline{x}_1, \underline{x}_2$. [I-5] © 2000 IEEE

Example 4.2 Figure 4.5 shows the near-complete and complete integer DFTs of two random input vectors $\underline{x}_1$ and $\underline{x}_2$.

$$\begin{aligned}\underline{x}_1 &= (2, 3, 4, 5, 4, 5, 2, 3)^T\\ \underline{x}_2 &= (2.8,\quad 4.3 - j0.6,\quad 3.7 + j0.9,\quad 3.1 - j0.6,\\ &\qquad 4.6,\quad 3.1 + j0.6,\quad 3.70 - j0.9,\quad 4.3 + j0.6)^T\end{aligned}$$

A parameter set is chosen for the near-complete integer DFT:

$$\{a_1 = 2,\ a_2 = 1,\ b_1 = 1, b_2 = 1\}$$

A parameter set is chosen for the complete integer DFT:

$$\{a_1 = 7,\ a_2 = 5,\ b_1 = 13, b_2 = 9, a_3 = 18, a_4 = 13, b_3 = 10, b_4 = 7\}$$

Entries $F_{\rm i}(k,n)$ of $[F_{\rm i}]$ are normalized by the first column $F_{\rm i}(k,0)$ as

$$\tilde{F}_{\rm i}(k,n) = F_{\rm i}(k,n)/F_{\rm i}(k,0) \qquad k,\ n = 0,\ 1\ ,...,\ N-1 \tag{4.34}$$

Integer DFT vector is computed for both the near-complete and complete integer DFTs as follows.

$$\underline{X}_1^T = [\tilde{F}_{\rm i}]^T \underline{x}_1^T \tag{4.35}$$

$[F_{\rm i}]^T$ can be normalized differently using $[IF][F_{\rm i}]^H = [D]$ of (4.29) to get the normalized integer DFT $[\tilde{F}_{\rm i}]^T$ as

$$[\tilde{F}_{\rm i}] = \left([D]^{1/2}\right)^{-1}[F_{\rm i}] \tag{4.36}$$

where diagonal matrix $[D]$ is defined in (4.22). Similarly

$$[I\tilde{F}] = \left([D]^{1/2}\right)^{-1}[IF] \tag{4.37}$$

Then $[\tilde{F}_{\rm i}]^{-1} = [I\tilde{F}]^H$, i.e., $[\tilde{F}_{\rm i}]$ and $[I\tilde{F}]$ are biorthogonal.

4.5 Summary

This chapter has developed the integer FFT (IntFFT) based on the lifting scheme. Its advantages are enumerated. A specific algorithm (eight-point IntFFT) using split-radix structure is developed. Extension of the 1-D DFT to the multi-D DFT (specifically 2-D DFT) is the focus of the next chapter. Besides the definitions and properties, filtering of 2-D signals such as images and variance distribution in the DFT domain are some relevant topics.

4.6 Problems

4.1 If $\det[A] = 1$ and $b \neq 0$,

$$[A] = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ (d-1)/b & 1 \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (a-1)/b & 1 \end{bmatrix} \tag{P4.1}$$

Assume $c \neq 0$. Derive (4.5) from (P4.1).

4.2 Develop a flow-graph for implementing eight-point inverse integer FFT using split-radix structure (see Fig. 4.4).

4.3 Repeat Problem 4.2 for $N = 16$ for forward and inverse integer FFTs.

4.4 List five other parameter sets for the integer DFT than those in Table 4.5. What equation do you need?

4.7 Projects

4.1 Repeat the simulation described in Example 4.2 about integer DFTs and obtain the results shown in Fig. 4.5.