

이산 및 고속 Fourier 변환

이 책 ‘1절 이산 푸리에 변환’과 관련하여 기본 개념을 보충 설명하고, 행렬 표현 접근을 통해 DFT가 일종의 직교 변환임을 보였다.

책의 ‘3절 원형 컨벌루션’과 관련하여 원형 컨벌루션의 개념을 보충 설명하고 선형 컨벌루션과 어떤 관계인지를 보충 설명하였다. 또한 중복-더하기 방법 및 중복-저장 방법의 알고리즘을 정리하여 나타내었다.

책의 ‘4절 고속 푸리에 변환’에 대해서는 신호 흐름 선도의 성질을 이용한 다양한 변형 FFT 선도를 소개하고, 역 FFT의 수식적 유도를 추가하였으며, 간단한 FFT 계산 예제를 보충하였다. 또한 밑수 2 FFT보다 일반화된 소인수 분해에 의한 FFT에 대해 비교적 상세하게 원리와 예를 소개하였다. 이 주제는 조금 더 고급의 FFT 학습이 필요한 사람들에게 유용할 것이다.

이상과 같은 보충 자료를 잘 학습하면 이 책 2부의 주제인 주파수 영역 해석의 실용적 방법인 DFT와 FFT에 대한 이해와 활용이 힘들지 않을 것이다.

7.1 이산 Fourier 변환(DFT)

7.1.1 이산 Fourier 변환(DFT)의 기본 개념

푸리에 급수 및 변환이나 z 변환과 같은 주파수 영역 변환 기법들은 시간 영역에서는 파악이 어려운 특성들을 주파수 영역에서 볼 수 있게 해 주므로 신호와 시스템의 분석과 취급에 매우 유용하다. 실제로 디지털 필터의 출력 계산을 비롯한 다양한 응용문제에서 신호의 스펙트럼이나 시스템의 주파수 응답 또는 전달 함수를 사용하거나 구해야 할 때가 많다. 그러나 간단한 수식 전개로, 즉 손으로 해결할 수 있는 경우는 거의 없으므로 부득이 컴퓨터를 이용하여 처리를 해야 하는데, 두 가지 장애 요소에 부딪치게 된다.

하나의 대부분의 대상 신호가 시간에서 연속, 즉 아날로그 신호라는 점이고, 다른 하나는 일반적으로 주기 신호와 같은 특수한 경우를 제외하고는 스펙트럼이 주파수에 대한 연속 함수라는 것이다.

그런데, 전자의 문제는 샘플링이라는 수단을 이용하여 해결할 수 있음을 이미 4장에서 배워 알고 있다. 이렇게 샘플링에 의해 이산 신호로 바뀌면 DTFT를 적용할 수 있으나, 스펙트럼이 주파수의 연속 함수라는 문제는 여전히 남는다. 게다가 DTFT는 모든 시간 순간에 대한 신호 값을 알고 있다는 전제 하에 수학적인 이론 전개가 이루어진 것이기 때문에 유한개의 데이터 그것도 적은 수의 데이터에 대한 작업인 경우가 대부분인 실제 문제에 적용하기에는 맞지 않는 부분이 있다.

후자의 문제에 대해서도 스펙트럼이 이산 함수가 되어야 한다는 결과는 이미 알고 있지만 어떻게 논리적인 일관성을 유지하면서 그렇게 할 수 있을까 하는 것이 관건이다. 우리는 이미 푸리에 변환의 성질에서 시간 영역에서 신호가 이산이면 주파수 영역에서 스펙트럼이 주기적이며, 또한 시간 영역에서 신호가 주기적이면 주파수 영역에서 스펙트럼이 이산으로 주어진다는 것을 알고 있다. 그러므로 이 둘을 결합한다면 시간 영역에서 이산이며 주기적인 신호에 주파수 영역에서도 이산이며 주기적인 스펙트럼이 대응될 것이다. 이것이 바로 6.2 절에서 다룬 DTFS이다.

이와 비슷하게, 시간 영역에서 연속인 신호를 샘플링을 통해 이산 신호로 만들 듯이 주파수 영역에서 연속인 스펙트럼을 등간격으로 샘플링한 것과 같은 효과가 나도록 만든다면, 다시 말해, 실제로는 그렇지 않지만 주어진 이산 신호를 마치 동일한 데이터가 주기적으로 반복되는 주기 신호인 것처럼 취급하여 이를 푸리에 변환하게 되면 스펙트럼은 주파수에 대한 이산 함수가 되어 컴퓨터를 이용한 처리에 아무런 장애가 없게 될 것이다.

이러한 시도의 결과로 유한한 개수의 이산 데이터로부터 유한한 개수의 이산 스펙트럼을 생성하여 컴퓨터에 의한 수치 계산이 가능한 새로운 종류의 푸리에 변환이 얻어지며, 이것이 바로 이산 푸리에 변환(DFT)이다. 이산 푸리에 변환은 이론적 프레임이 아니라 컴퓨터를 이용한 효율적 수치 계산을 위한 도구적 프레임이라는 것을 명심해야 한다.

7.1.2 DFT의 행렬 표현

(책0식 (7.3)의 DFT를 (책)[예제 7-1]의 풀이 과정에서 했던 것처럼 $k=0, 1, \dots, N-1$ 에 대해 풀어 쓰면 DFT는 결국 N 개의 미지수 $X[0], X[1], \dots, X[N-1]$ 에 대해 N 개의 연립방정식을 푸는 문제와 같게 된다.

DFT를 행렬 형태로 정리하여 나타내면 다음과 같다.

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & \cdots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (C7-1)$$

$$\mathbf{X} = \mathbf{W} \mathbf{x} \quad (C7-2)$$

식 (C7-1)에서 회전 인자로 이루어진 계수 행렬 \mathbf{W} 를 DFT행렬이라고 한다. 회전 인자 $W_N^{kn} = W_N^{nk}$ 이므로 DFT행렬 \mathbf{W} 는 대칭 행렬이다. 즉 $\mathbf{W}^T = \mathbf{W}$ 이다.

한편 IDFT를 행렬 형태로 나타내면, $W_N^{-kn} = (W_N^{nk})^*$ 이므로 다음과 같다.

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \frac{1}{N} \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & \cdots & W_N^0 \\ W_N^0 & W_N^{-1} & W_N^{-2} & \cdots & \cdots & W_N^{-(N-1)} \\ W_N^0 & W_N^{-2} & W_N^{-4} & \cdots & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ W_N^0 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & \cdots & W_N^{-(N-1)^2} \end{bmatrix} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} \quad (C7-3)$$

$$\mathbf{x} = \frac{1}{N} \mathbf{W}^* \mathbf{X} \quad (C7-4)$$

그런데, 식(C7-2)를 \mathbf{x} 에 대해서 풀면 $\mathbf{x} = \mathbf{W}^{-1} \mathbf{X}$ 가 되므로 다음이 성립한다.

$$\mathbf{W}^{-1} = \frac{1}{N} \mathbf{W}^* \quad (C7-5)$$

\mathbf{W} 가 대칭 행렬이므로, 식 (C7-5)은 다음과 같이 다시 쓸 수 있다.

$$\mathbf{W}^{*T} \mathbf{W} = N \mathbf{I} \quad (C7-6)$$

일반적으로 행렬 \mathbf{A} 가 $\mathbf{A}^{*T}\mathbf{A}=\mathbf{I}$ 를 만족하면 유니터리 행렬 ^{unitary matrix}이라 한다. 유니터리 행렬은 $\mathbf{A}^T\mathbf{A}=\mathbf{I}$ 를 만족하는 실수 직교행렬 ^{orthogonal matrix}을 복소 행렬에 확장시킨 개념이다.

식 (C7-6)에서 보면 행렬 \mathbf{W} 가 엄밀히 말해 유니터리 행렬이 아니지만 IDFT의 계수 행렬에 곱해진 $1/N$ 을 쪼개어 DFT와 IDFT 양쪽의 계수 행렬에 $1/\sqrt{N}$ 을 똑같이 곱하여 이를 계수 행렬로 사용하면 식 (C7-2)와 식 (C7-4)는 각각 $\mathbf{X}=\mathbf{W}\mathbf{x}$ 와 $\mathbf{x}=\mathbf{W}^*\mathbf{X}$ 가 되므로 \mathbf{W} 는 유니터리 행렬이 된다. 따라서 **DFT는 유니터리 변환, 즉 직교 변환이다**. 직교변환의 유용성에 대해서는 5장에서 이미 상세히 설명한 바 있다.

7.2 이산 푸리에 변환의 특성

7.3 원형 컨벌루션

7.3.1 원형 컨벌루션의 개념

원형 컨벌루션의 개념을 이해하기 위해 다음과 같은 두 주기 신호의 컨벌루션을 살펴보자.

$$x[n]=[1, 2, 3, 4, 5, 6], \quad h[n]=[a, b, c, d, e, f]$$

잘 알다시피 컨벌루션은 하나의 신호는 고정시키고 다른 신호를 뒤집어서 오른쪽으로 미끄러뜨리면서 매 스텝마다 연산을 하는 동작이다. 따라서 $x[n]$ 을 고정시키고 $h[n]$ 을 뒤집어서 이동시키는 동작은 다음과 같이 나타낼 수 있다.

$\begin{array}{cccccc} \cdots & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \cdots & a & f & e & d & c & b \\ \cdots & b & a & f & e & d & c \\ \cdots & c & b & a & f & e & d \\ \cdots & d & c & b & a & f & e \\ \cdots & e & d & c & b & a & f \\ \cdots & f & e & d & c & b & a \\ \hline \cdots & a & f & e & d & c & b \\ \cdots & b & a & f & e & d & c \end{array}$	$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline a & f & e & d & c & b \\ b & a & f & e & d & c \\ c & b & a & f & e & d \\ d & c & b & a & f & e \\ e & d & c & b & a & f \\ f & e & d & c & b & a \\ \hline a & f & e & d & c & b \\ b & a & f & e & d & c \end{array}$	$\begin{array}{ccc} 1 & 2 & \cdots \\ \hline a & f & \cdots \\ b & a & \cdots \\ c & b & \cdots \\ d & c & \cdots \\ e & d & \cdots \\ f & e & \cdots \\ \hline a & f & \cdots \\ b & a & \cdots \end{array}$
--	---	---

(C7-1)

식 (C7-1)에서 보면, 주기 신호의 컨벌루션은 이동 신호가 한 주기만큼 이동하면 정확히 같은 패턴을 반복한다. 또한 선으로 구분한 한 주기 블록 내의 신호 값들만 살펴보면, 한

스텝 이동하면 제일 앞의 신호가 사라지는 게 아니라 마치 동그란 고리로 연결된 것처럼 맨 뒤로 나타난다. 따라서 (책)[그림 7-6]과 같이 두 개의 동심원 위에 컨벌루션할 두 신호의 한 주기 값들을 각각 늘어놓고 하나를 한 클릭씩 돌리는 것과 같은 형태로 계산이 진행되는 것이다. 물론 전체 구간의 값은 일반적으로 무한대가 되지만 한 주기 구간만 떼어 놓고 본다면 유한한 값이 된다는 것을 알 수 있다.

이에 반해 $x[n]$ 과 $h[n]$ 이 비주기 신호인 경우, 두 신호에 대한 선형 컨벌루션은 아래와 같이 시간의 경과에 따라 이동 신호가 지나가면 그 자리로 다시 나타나지 않고 사라진다.

$$\begin{array}{cccccc|cccccc}
 & & & & & & 1 & 2 & 3 & 4 & 5 & 6 \\
 f & e & d & c & b & & a & & & & & \\
 & f & e & d & c & & b & a & & & & \\
 & & f & e & d & & c & b & a & & & \\
 & & & f & e & & d & c & b & a & & \\
 & & & & f & & e & d & c & b & a & \\
 & & & & & f & e & d & c & b & a & \\
 & & & & & & f & e & d & c & b & a \\
 & & & & & & & f & e & d & c & b \\
 & & & & & & & & f & e & d & c \\
 & & & & & & & & & f & e & d \\
 & & & & & & & & & & f & e \\
 & & & & & & & & & & & f
 \end{array} \quad (C7-2)$$

그런데 식 (C7-1)과 식 (C7-2)를 비교해보면 흥미로운 사실을 알 수 있다. 식 (C7-2)의 선형 컨벌루션의 이동 신호의 첫 번째 행과 일곱 번째 행을 더하면 식 (C7-1)의 원형 컨벌루션의 이동 신호의 첫 번째 행이 되고, 식 (C7-2)의 이동 신호의 두 번째 행과 여덟 번째 행을 더하면 식 (C7-1)의 이동 신호의 두 번째 행이 된다. 식 (C7-1)의 이동 신호의 나머지 행들에 대해서도 마찬가지로, 여섯 번째 행의 경우는 식 (C7-2)의 여섯 번째 행에 0 벡터를 더한 것과 같다.

이상의 관찰 결과로부터 두 N -주기 신호의 원형 컨벌루션은 두 신호의 선형 컨벌루션 결과를 길이 반으로 자른 다음 더한 것과 같다는 것을 알 수 있다. 즉 두 신호의 선형 컨벌루션 $z[n] = x[n] * h[n]$ 은 길이 $2N-1$ 인 신호이므로, 원형 컨벌루션 $y[n] = x[n] \otimes h[n]$ 은 다음과 같게 된다.

$$y[n] = [z[0] \ z[1] \ z[2] \ \cdots \ z[N-1]] + [z[N] \ z[N+1] \ \cdots \ z[2N-2] \ 0] \quad (C7-3)$$

7.3.2 원형 컨벌루션을 이용한 긴 데이터의 컨벌루션

(1) 중복-더하기 방법

중복-더하기 방법을 정리하면 다음과 같다.

1. $h[n]$ 에 $L-1 = N-M$ 개의 영점 채우기를 시행한다.

$$h_a[n] = \begin{cases} h[n], & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases} \quad (C7-4)$$

2. $x[n]$ 을 중복 없이 길이 L 로 나누어 각각 $M-1$ 개의 영점 채우기를 시행한다.

$$x_i[n] = \begin{cases} x[n], & iL \leq n \leq iL+L-1 \\ 0, & iL+L \leq n \leq iL+N-1 \end{cases} \quad (C7-5)$$

3. DFT와 IDFT를 이용하여 N 점 원형 컨벌루션을 수행한다.

$$y_i[n] = x_i[n] \otimes h_a[n], \quad iL \leq n \leq iL+N-1 \quad (C7-6)$$

4. 각각의 원형 컨벌루션 수행 결과를 그대로 더하여 출력으로 낸다.

$$y[n] = x[n] * h[n] = \sum_{i=0}^{\infty} y_i[n] \quad (C7-7)$$

위의 알고리즘에서 $y_i[n]$ 은 $x[n]$, $iL \leq n \leq iL+L-1$ 에 의한 응답 성분이고 $y_{i+1}[n]$ 은 $x[n]$, $iL+L \leq n \leq iL+N-1$ 에 의한 응답 성분이므로 두 응답이 겹쳐지는 부분은 과거의 입력에 의한 응답과 현재의 입력에 의한 응답이 따로 계산되어 합쳐지는 부분에 해당된다. 따라서 출력이 식 (C7-7)과 같이 계산되는 것은 타당하다.

(2) 중복-저장 방법

중복-저장 방법을 정리하면 다음과 같다.

1. $h[n]$ 에 $L-1 = N-M$ 개의 영점 채우기를 시행한다.

$$h_a[n] = \begin{cases} h[n], & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases} \quad (C7-8)$$

2. $x[n]$ 을 $M-1$ 샘플씩 중복되게 길이 N 으로 나눈다. 이때 첫 번째의 경우는 중복 샘플이 없으므로 맨 앞부분에 $M-1$ 개의 영점을 채운다.

$$x_0[n] = \begin{cases} 0, & -M+1 \leq n \leq -1 \\ x[n], & 0 \leq n \leq L-1 \end{cases} \quad (C7-9)$$

$$x_i[n] = x[n], \quad iL-M+1 \leq n \leq iL+L-1 \quad (C7-10)$$

3. DFT와 IDFT를 이용하여 N 점 원형 컨벌루션을 수행한다.

$$y_i[n] = x_i[n] \otimes h_a[n], \quad iL - M + 1 \leq n \leq iL + L - 1 \quad (C7-11)$$

4. 각각의 원형 컨벌루션 수행 결과에서 앞의 $M-1$ 샘플을 버린 뒤 더하여 출력으로 낸다.

$$\bar{y}_i[n] = \begin{cases} 0, & iL - M + 1 \leq n \leq iL - 1 \\ y_i[n], & iL \leq n \leq iL + L - 1 \end{cases} \quad (C7-12)$$

$$y[n] = x[n] * h[n] = \sum_{i=0}^{\infty} \bar{y}_i[n] \quad (C7-13)$$

위에서 $y_i[n]$ 을 만드는 입력을 $x[n]$, $iL - M + 1 \leq n \leq iL - 1$ 과 $x[n]$, $iL \leq n \leq (i+1)L - 1$ 의 둘로 나누어보면, $iL - M + 1 \leq n \leq iL - 1$ 에서는 그 구간의 입력, 즉 전자에 의한 응답만이 나오므로 불완전하다. $iL \leq n \leq (i+1)L - 1$ 은 전자의 입력이 들어오기 시작한 때로부터 임펄스 응답의 지속 시간만큼 지난 이후이므로 전자의 영향과 현 구간의 입력, 즉 후자의 영향이 모두 반영된 응답이 만들어지기 때문에 정상적인 출력이다. $y_{i+1}[n]$ 을 만들어내는 입력의 앞부분은 $y_i[n]$ 을 만들어내는 입력과 임펄스 응답의 지속 시간만큼 겹쳐 있는데, 이 구간의 컨벌루션 결과 값은 조금 전에 설명한 것처럼 불완전할 뿐만 아니라 어차피 $y_i[n]$ 에서 올바르게 계산된 결과가 이미 있기 때문에 불필요하다. 따라서 각 $y_i[n]$ 의 앞부분 $M-1$ 샘플을 버리고 더하여 출력으로 내는 동작이 바로 식 (C7-13)이다.

7.4 고속 Fourier 변환 (FFT)

7.4.1 FFT 알고리즘의 여러 가지 변형

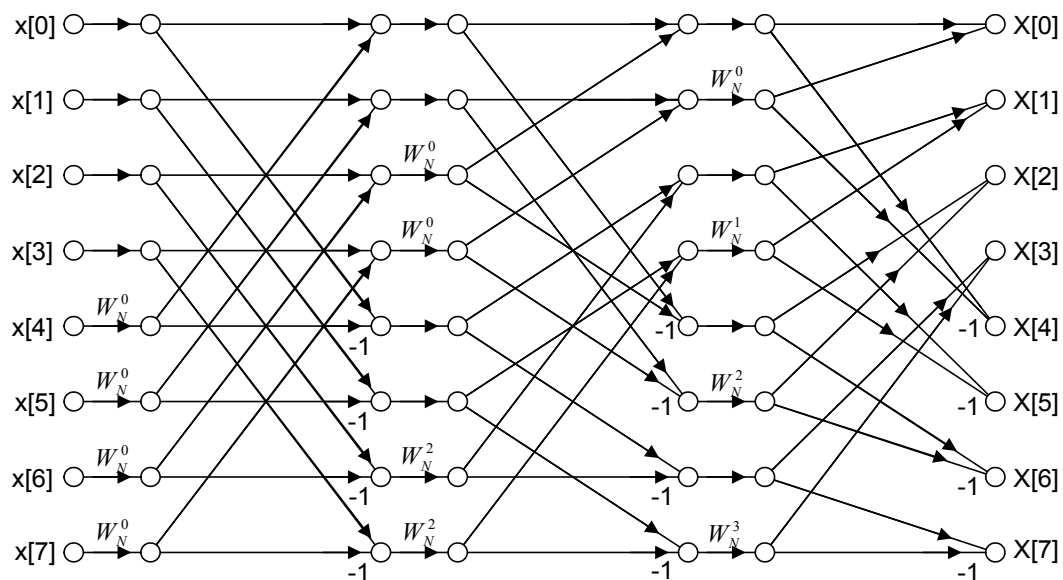
(1) 시분할 FFT의 변형

신호 흐름 선도의 중요한 특성은 마디의 연결 관계만 바뀌지 않으면 그림의 형태가 바뀌어도 동일한 대상을 표현한 것이라는 점이다. 그러므로 FFT 선도에서 마디의 연결만 달라지지 않으면 그림에 나타낸 마디의 순서가 바뀌어도 계산 결과는 같게 된다.

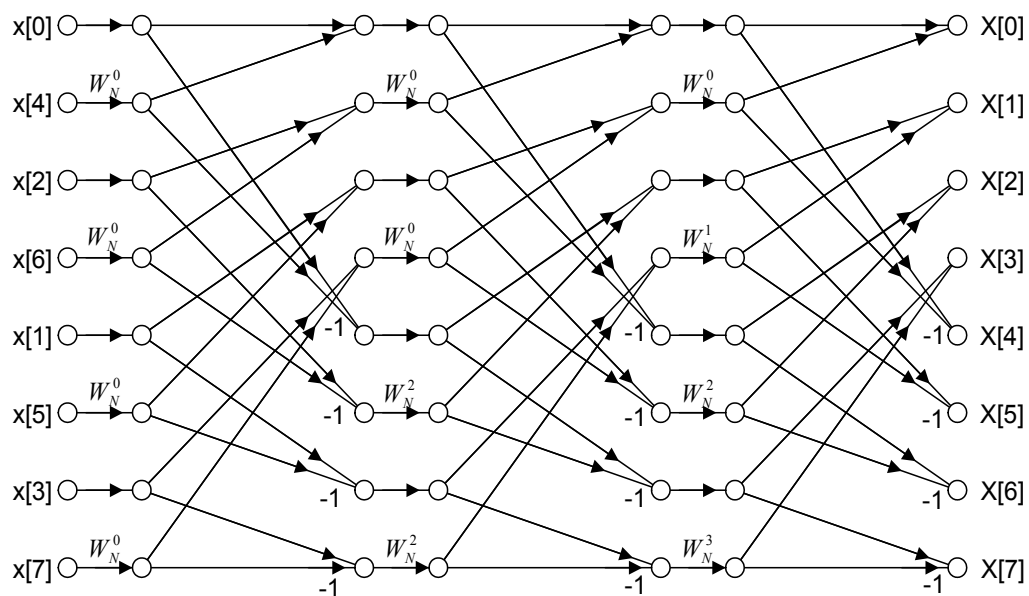
따라서 밑수 2 시분할 FFT를 몇 가지 형태로 변형할 수 있다. 우선 (책)[그림 7-23]의 입력 역 비트순 형에서 $x[4] \leftrightarrow x[1]$, $x[6] \leftrightarrow x[3]$ 의 위치를 교환하면 (책)[그림 7-25]와 같이 입력 $x[n]$ 은 정상 순서, 출력 $X[k]$ 가 역 비트순인 형태로 바꿀 수 있다.

필요에 따라서는 [그림 C7-1]과 같이 입력 $x[n]$ 과 출력 $X[k]$ 모두 정상 순서인 형태로 바꿀 수도 있다. 또 다른 변형으로는 [그림 C7-2]와 같이 각 단계의 기하 구조가 동일한 형

태로 만드는 경우를 들 수 있다. 앞에서 살펴본 세 가지 FFT 선도는 각 단계의 기하 구조가 달라서 데이터의 순차적인 액세스(access)가 안 되지만, 이 경우는 각 단계의 기하 구조를 같게 하여 데이터의 순차적 액세스를 가능하게 함으로써 프로그래밍이나 하드웨어 구현에 매우 유리하다. 왜냐하면 하나의 단계에 대한 프로그램이나 하드웨어를 변형 없이 그대로 반복하여 활용할 수 있기 때문이다.



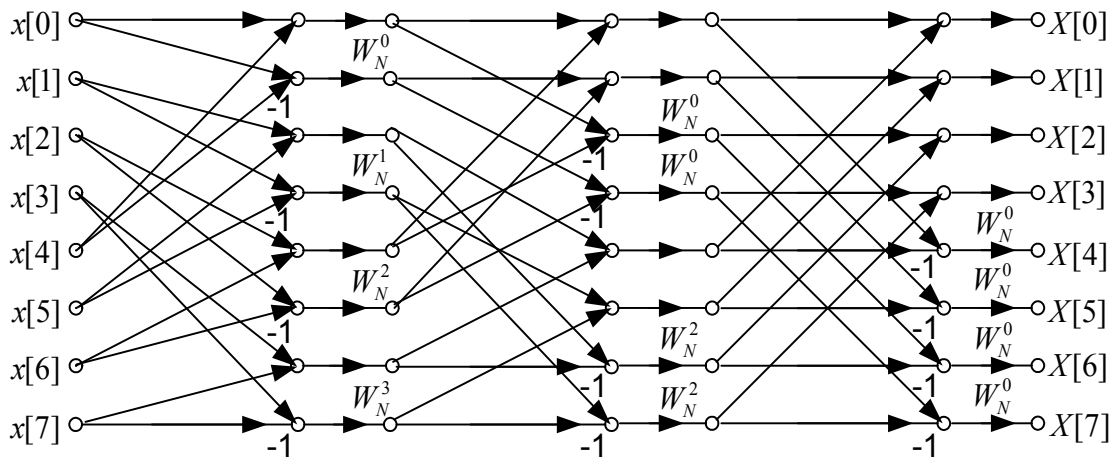
[그림 C7-1] 입력과 출력이 모두 정상순인 시분할 FFT 선도 (N=8)



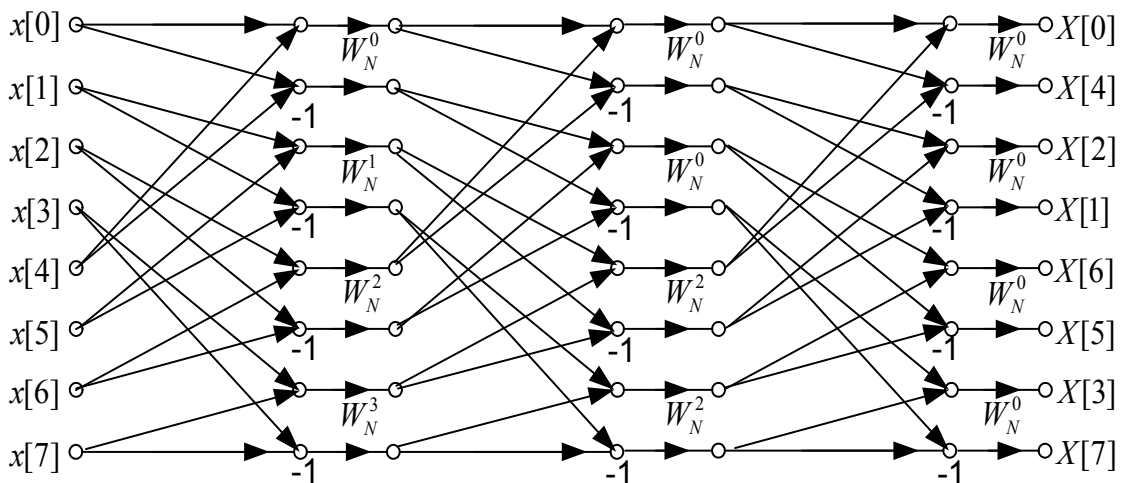
[그림 C7-2] 각 단계의 기하 구조가 같은 시분할 FFT 선도 (N=8)

(2) 주파수 분할 FFT의 변형

주파수 분할 FFT의 경우에도 시분할 FFT와 마찬가지로 여러 가지 변형이 가능하다. (책) [그림 7-28]의 출력 역 비트순과 (책) [그림 7-29]의 입력 역 비트순 외에도 [그림 C7-3]과 같이 입출력이 모두 정상순인 FFT 선도와 각 단계별 기하 구조가 같은 [그림 C7-4]의 FFT 선도로 변형할 수 있다.



[그림 C7-3] 입출력 정상순 주파수 분할 FFT (N=8)



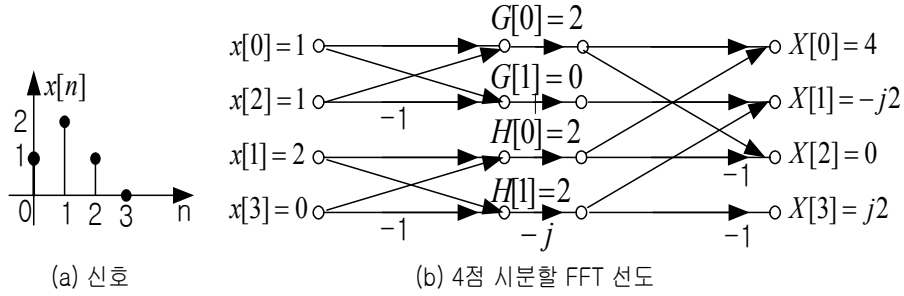
[그림 C7-4] 동일 기하 구조 주파수 분할 FFT 선도 (N=8)

■ 예제 C7-1 : 4점 밑수 2 시분할 FFT 계산

[그림 7-5(a)]와 같은 신호 $x[n]$ 에 대해 4점 밑수 2 시분할 FFT 선도를 그리고 이를 이용하여 샘플 스펙트럼을 구하라.

<풀이>

주어진 신호에 대해 4점 시분할 FFT 선도를 그리면 [그림 7-5(b)]와 같고, 이로부터 샘플 스펙트럼을 계산하면 다음과 같다.



[그림 C7-5] [예제 C7-1]의 신호 및 FFT 선도

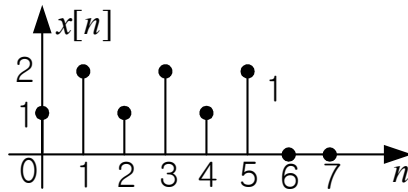
$$\begin{aligned}
 G[0] &= x[0] + x[2] = 2 & \Rightarrow & X[0] = G[0] + H[0] = 4 \\
 G[1] &= x[0] - x[2] = 0 & & X[1] = G[1] + (-j) \cdot H[1] = -j2 \\
 H[0] &= x[1] + x[3] = 2 & & X[2] = G[0] - H[0] = 0 \\
 H[1] &= x[1] - x[3] = 2 & & X[3] = G[1] - (-j) \cdot H[1] = j2
 \end{aligned}$$

■ 예제 C7-2 : 8점 밑수 2 시분할 FFT 계산

[그림 7-6]과 같은 신호 $x[n]$ 에 대해 8점 밑수 2 시분할 FFT 선도를 그리고 이를 이용하여 샘플 스펙트럼을 구하라.

<풀이>

8점 FFT를 수행하기 위해 필요한 회전 인자는 W_8^m , $m = 0, 1, 2, 3$ 이므로 이를 구하면 다음과 같다.



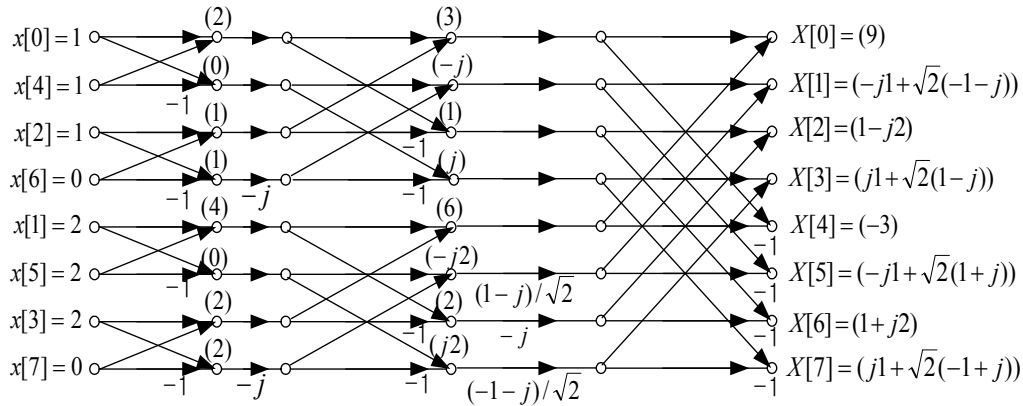
$$\begin{cases}
 W_8^0 = 1 \\
 W_8^1 = \frac{1}{\sqrt{2}}(1 - j) \\
 W_8^2 = -j \\
 W_8^3 = \frac{1}{\sqrt{2}}(-1 - j)
 \end{cases}$$

[그림 C7-6] [예제 C7-2]의 신호

주어진 신호에 대해 8점 시분할 FFT 선도를 그리면 [그림 7-7]과 같고, 이로부터 샘플 스펙트럼을 계산하면 다음과 같다.

$$\begin{aligned}
 B_1[0] &= x[0] + x[4] = 2 & \Rightarrow & G[0] = B_1[0] + B_2[0] = 3 \\
 B_1[1] &= x[0] - x[4] = 0 & & G[1] = B_1[1] + (-j)B_2[1] = -j \\
 B_2[0] &= x[2] + x[6] = 1 & & G[2] = B_1[0] - B_2[0] = 1 \\
 B_2[1] &= x[2] - x[6] = 1 & & G[3] = B_1[1] - (-j)B_2[1] = j \\
 B_3[0] &= x[1] + x[5] = 4 & & H[0] = B_3[0] + B_4[0] = 6 \\
 B_3[1] &= x[1] - x[5] = 0 & & H[1] = B_3[1] + (-j)B_4[1] = -j2 \\
 B_4[0] &= x[3] + x[7] = 2 & & H[2] = B_3[0] - B_4[0] = 2 \\
 B_4[1] &= x[3] - x[7] = 2 & & H[3] = B_3[1] - (-j)B_4[1] = j2
 \end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad X[0] &= G[0] + H[0] = 9 \\
X[1] &= G[1] + H[1] = -j + \sqrt{2}(-1-j) = -\sqrt{2} - j(\sqrt{2}+1) \\
X[2] &= G[2] + H[2] = 1 - j2 \\
X[3] &= G[3] + H[3] = j + \sqrt{2}(1-j) \\
X[4] &= G[0] + H[0] = -3 \\
X[5] &= G[1] - \left(\frac{1-j}{\sqrt{2}}\right)H[1] = -j + \sqrt{2}(1+j) = \sqrt{2} + j(\sqrt{2}-1) \\
X[6] &= G[2] - (-j)H[2] = 1 + j2 \\
X[7] &= G[3] - \left(\frac{-1-j}{\sqrt{2}}\right)H[3] = jj + \sqrt{2}(-1+j) = -\sqrt{2} + j(\sqrt{2}+1)
\end{aligned}$$



[그림 C7-7] [예제 C7-2]의 시분할 FFT 선도

7.4.2 역 FFT(IFFT)의 유도

$X[k]$ 를 짝수 항과 홀수 항으로 분할하면

$$\begin{aligned}
x[n] &= \sum_{k=0}^{N/2-1} \frac{X[k]}{N} W_N^{-kn} = \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} W_N^{-2rk} + \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} W_N^{-(2r+1)k} \quad (C7-14) \\
&= \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} (W_N^2)^{-kr} + W_N^{-k} \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} (W_N^2)^{-kr} \\
&= \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} W_{\frac{N}{2}}^{-kr} + \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} W_{\frac{N}{2}}^{-kr} \cdot W_N^{-k} \\
&= \sum_{r=0}^{N/2-1} C_0[k] + \sum_{r=0}^{N/2-1} W_N^{-k} C_1[k]
\end{aligned}$$

식 (C7-14)를 다시 짝수 항과 홀수 항으로 계속 분할해나가면 시분할 FFT의 유도 때와 마찬가지로 궁극적으로는 다음과 같은 2점 IDFT로 나뉜다.

$$U[k] = \sum_{i=0}^1 \frac{X[v(i)]}{N} W_2^{-ik}, \quad k=0, 1 \quad (\text{C7-15})$$

여기서 $W_2^{-0k} = 1$, $W_2^{-1k} = e^{j\frac{2\pi}{2}k} = \begin{cases} 1, & k=0 \\ -1, & k=1 \end{cases}$ 이다.

따라서 식 (C7-15)를 다시 쓰면 다음과 같다.

$$\begin{cases} U[0] = \frac{X[v(0)]}{N} + \frac{X[v(1)]}{N} \\ U[1] = \frac{X[v(0)]}{N} - \frac{X[v(1)]}{N} \end{cases} \quad (\text{C7-16})$$

7.4.3 일반화된 FFT

$N \neq 2^q$ 일 경우에도 N 을 소수의 곱, 즉 소인수 분할에 의해 쪼개어 FFT 구조로 만들 수 있다. 이와 같이 일반화된 FFT를 어떻게 유도할 수 있는지 살펴보기로 하자.

$N = p_1 \cdot q_1$ 이라면 입력 수열을 p_1 개의 q_1 샘플 수열로 분할하여 N 점 DFT를 p_1 개의 q_1 점 DFT로 바꿀 수 있다. 그러면 N 점 DFT는 p_1 개의 q_1 점 DFT를 계산하여 W_N^{lk} 곱해서 더하면 된다. 따라서 이 경우에는 $p_1 q_1^2 + N(p_1 - 1)$ 의 곱셈이 필요하다.

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{kn} \\ &= \sum_{r=0}^{q_1-1} x[p_1 r] W_N^{p_1 r k} + \sum_{r=0}^{q_1-1} x[p_1 r + 1] W_N^k W_N^{p_1 r k} + \dots \\ &\quad + \sum_{r=0}^{q_1-1} x[p_1 r + p_1 - 1] W_N^{(p_1-1)k} W_N^{p_1 r k} \\ &= \sum_{l=0}^{p_1-1} W_N^{lk} \sum_{r=0}^{q_1-1} x[p_1 r + l] W_N^{p_1 r k} = \sum_{l=0}^{p_1-1} W_N^{lk} G_l[k] \end{aligned} \quad (\text{C7-17})$$

여기서

$$G_l[k] \equiv \sum_{r=0}^{q_1-1} x[p_1 r + l] W_N^{p_1 r k} = \sum_{r=0}^{q_1-1} x[p_1 r + l] W_{q_1}^{rk} \quad (\text{C7-18})$$

식 (C7-18)은 $W_N^{p_1 r k} = e^{-j \frac{2\pi}{q_1} r k} = W_{q_1}^{r k}$ 임을 이용한 것이다.

예를 들어 $N = 12 = 3 \times 4$ 으로 나눈다면, 다음과 같이 3개의 4점 DFT로 분할이 될 것이다.

$$\begin{array}{cccc} x[0] & x[3] & x[6] & x[9] \\ x[1] & x[4] & x[7] & x[10] \\ x[2] & x[5] & x[8] & x[11] \end{array} \quad (C7-19)$$

위와 같은 분할에서 같은 그룹의 신호 샘플은 $x[p_1 r + l]$, $r = 0, \dots, q_1 - 1$ 이다. 이때 그룹은 $l = 0, \dots, p_1 - 1$ 에 따라 분류된다.

다시 $q_1 = p_2 q_2$ 로 두어 q_1 점 DFT를 p_2 개의 q_2 점 DFT로 분할하게 되면, p_2 개의 q_2 점 DFT는 $p_2 q_2^2 + q_1(p_2 - 1)$ 의 곱셈이 필요하다. 따라서 N 점 DFT는 $N(p_1 - 1) + N(p_2 - 1) + p_1 p_2 q_2^2$ 의 곱셈이 필요하게 된다. 끝까지 분할하면 $N = p_1 p_2 \dots p_\nu$ 이 되므로 p_ν 점 DFT로 시작하는 ν 단계를 갖는 FFT로 변환된다. 이때 필요한 곱셈의 양은 다음과 같다.

$$N(p_1 - 1) + N(p_2 - 1) + \dots + N(p_{\nu-1} - 1) + p_1 p_2 \dots p_{\nu-1} p_\nu^2 = N(p_1 + \dots + p_\nu - (\nu - 1)) \quad (C7-20)$$

N 을 분할할 때 더 이상 나누어지지 않도록 소인수 분해하는 것이 계산량이 더 적게 된다. 예를 들어 $N = p_1 p_2 \dots p_\nu$ 로 분할했을 때 $p_i = r_i s_i$ 라면, $N = p_1 \dots p_{i-1} r_i s_i p_{i+1} \dots p_\nu$ 이므로 더 분할된 FFT의 곱셈 연산량은 $N(p_1 + \dots + p_{i-1} + r_i + s_i + p_{i+1} + \dots + p_\nu - (\nu - 1))$ 이 된다. 그런데 $p_i = r_i s_i > r_i + s_i$ 이므로 다음과 같이 덜 쪼갠 때의 곱셈 연산량이 더 크다.

$$N(p_1 + \dots + p_i + \dots + p_\nu - (\nu - 1)) > N(p_1 + \dots + r_i + s_i + \dots + p_\nu - (\nu - 1)) \quad (C7-21)$$

이와 같이 소인수 분해에 의해 DFT를 분할할 때 어떤 규칙성을 가지고 식 (C7-19)와 같이 신호 샘플들이 그룹화되는지 살펴보자.

N 을 $N = p_1 p_2 \dots p_\nu$ 로 소인수 분해하면 시간 변수 n 은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} n &= p_1 q_1 + r_1 = p_1(p_2 q_2 + r_2) + r_1 = p_1(p_2(p_3 q_3 + r_3) + r_2) + r_1 = \dots \\ &= (p_1 p_2 \dots p_{\nu-1}) q_{\nu-1} + (p_1 p_2 \dots p_{\nu-2}) r_{\nu-1} + \dots + (p_1) r_2 + r_1 \\ &= (p_1 p_2 \dots p_{\nu-1}) n_1 + (p_1 p_2 \dots p_{\nu-2}) n_2 + \dots + (p_1) n_{\nu-1} + n_\nu \end{aligned} \quad (C7-22)$$

따라서 시간 변수 n 과 주파수 변수 k 는 다음과 같이 나타낼 수 있다.

$$\begin{cases} n = (p_1 p_2 \dots p_{\nu-1}) n_1 + (p_1 p_2 \dots p_{\nu-2}) n_2 + \dots + (p_1) n_{\nu-1} + n_\nu \\ k = (p_\nu p_{\nu-1} \dots p_2) n_\nu + (p_\nu p_{\nu-1} \dots p_3) n_{\nu-1} + \dots + (p_\nu) n_2 + n_1 \end{cases} \quad (C7-23)$$

여기서 $0 \leq n_1 \leq p_\nu - 1$, $0 \leq n_2 \leq p_{\nu-1} - 1$, \dots , $0 \leq n_\nu \leq p_1 - 1$ 로서 각 자리^{digit}의 값이다.

식 (C7-23)은 샘플 신호 $x[n]$ 과 샘플 스펙트럼 $X[k]$ 가 밑수 2 FFT때와 비슷하게 역 자리순으로 대응됨을 말해준다.

예를 들어 $N=18$ 의 경우, $N=3 \cdot 3 \cdot 2 = p_1 p_2 p_3$ 로 소인수 분할된다. 따라서 n 과 k 는 다음과 같이 나타낼 수 있다.

$$\begin{cases} n = p_1 p_2 n_1 + p_1 n_2 + n_3 \\ k = p_3 p_2 n_3 + p_3 n_2 + n_1 \end{cases} \quad (\text{C7-24})$$

이로부터 $n=16$ 에 대해 따져보면 $n=16=9 \cdot 1 + 3 \cdot 2 + 1$ 이므로 $n_1=1$, $n_2=2$, $n_3=1$ 이다. 따라서 $k=6 \cdot 1 + 2 \cdot 2 + 1=11$ 에 대응된다. 또한 $n=11$ 은 $n=11=9 \cdot 1 + 3 \cdot 0 + 2$ 이므로, $n_1=1$, $n_2=0$, $n_3=2$ 가 되어 $k=6 \cdot 2 + 2 \cdot 0 + 1=13$ 에 대응된다.

이상의 결과는 밑수 2 FFT를 일반화한 것이므로 당연히 밑수 2 FFT에서도 만족된다. 예를 들어 $N=4=2 \cdot 2=p_1 \cdot p_2$ 의 경우 $n=2n_1+n_2$, $\leftrightarrow k=n_1+2n_2$, $0 \leq n_1, n_2 \leq 1$ 이므로 다음과 같이 변환 관계를 그림으로 나타낼 수 있다.

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

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

n_1	n_2	n	n_2	n_1	k
0	0	0	0	0	0
1	0	2	0	1	1
0	1	1	1	0	2
1	1	3	1	1	3

[그림 C7-8] 소인수 분해 FFT의 인덱스 대응 관계

그림에서 보면, 식 (C7-23)의 대응 관계를 적용하여 얻은 결과임에도 입력은 역 비트순 출력은 정상순으로 대응되고 있다. 따라서 식 (C7-23)이 보다 일반적인 관계임을 알 수 있다.

■ 예제 C7-3 : 16점 밑수 4 FFT의 DFT 분할 관계

$N=16$ 일 때 밑수 4 FFT의 시간 신호 및 주파수 스펙트럼 분할 관계를 구하라.

<풀이>

$N=16=4 \cdot 4=p_1 \cdot p_2$ 이므로 $n=4n_1+n_2$, $\leftrightarrow k=n_1+4n_2$, $0 \leq n_1, n_2 \leq 3$ 이 된다. 따라서 샘플 신호와 샘플 스펙트럼의 분할 대응 관계는 [그림 C7-9]와 같이 된다.

그림에서 $\{x[0], x[4], x[8], x[12]\}$, $\{x[1], x[5], x[9], x[13]\}$, $\{x[2], x[6], x[10], x[14]\}$,

$\{x[3], x[7], x[11], x[15]\}$ 의 4개의 4점 DFT로 분할되어 정상순인 스펙트럼에 대응된다.

$n_2 \backslash n_1$	0	1	2	3
0	0	4	8	12
1	1	5	9	13
2	2	6	10	14
3	3	7	11	15

$n_1 \backslash n_2$	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

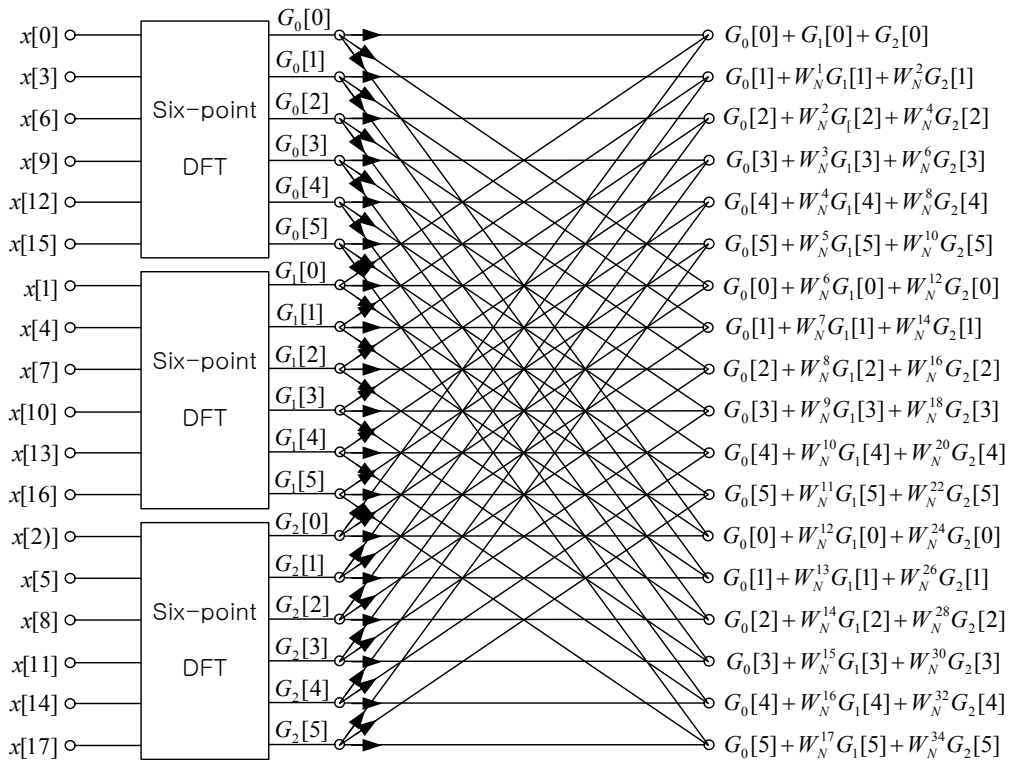
[그림 C7-9] [예제 C7-3]의 인덱스 대응 관계

이상과 같은 소인수 분해 FFT 원리를 이용하여 $N=18$ 일 경우의 FFT 선도를 구해보자. 우선 $N = 3 \cdot 6$, 즉 3개의 6점 DFT로 분할하면 다음과 같다.

$$\begin{array}{cccccc} x[0] & x[3] & x[6] & x[9] & x[12] & x[15] \\ x[1] & x[4] & x[7] & x[10] & x[13] & x[16] \\ x[2] & x[5] & x[8] & x[11] & x[14] & x[17] \end{array} \quad (C7-25)$$

이를 수식으로 나타내면 다음과 같고, 이에 대한 FFT 선도는 [그림 C7-10]에 나타내었다.

$$X[k] = \sum_{l=0}^2 W_{18}^{lk} \sum_{r=0}^5 x[3r+l] W_{18}^{3rk} = G_0[k] + W_{18}^k G_1[k] + W_{18}^{2k} G_2[k] \quad (C7-26)$$



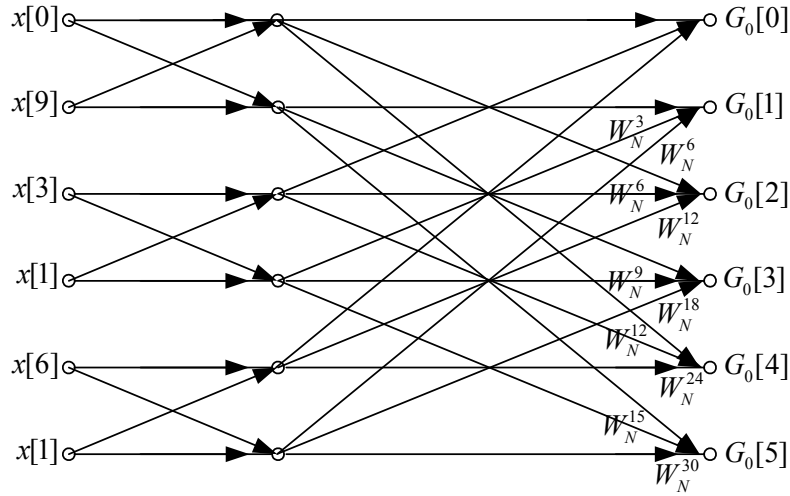
[그림 C7-10] 3개의 6점 DFT로 분할한 FFT 선도 ($N=18$)

[그림 C7-10]의 6점 DFT를 3개의 2점 DFT로 변환하면 다음과 같다.

$$\begin{array}{ccc} x[0] & x[9] & x[1] & x[10] & x[2] & x[11] \\ x[3] & x[12] & x[4] & x[13] & x[5] & x[14] \\ x[6] & x[15] & x[7] & x[16] & x[8] & x[17] \end{array} \quad (C7-27)$$

이를 수식으로 나타내면 다음과 같고, 첫 번째 블록에 대한 FFT 선도를 [그림 C7-11]에 나타내었다.

$$G_l[k] = \sum_{r=0}^5 x[3r+l] W_6^{rk} = \sum_{s=0}^2 W_6^{sk} \sum_{p=0}^1 x[9p+3s+l] W_6^{3pk} \quad (C7-28)$$



[그림 C7-11] 3개의 2점 DFT로 분할한 FFT 선도 ($N=6$)