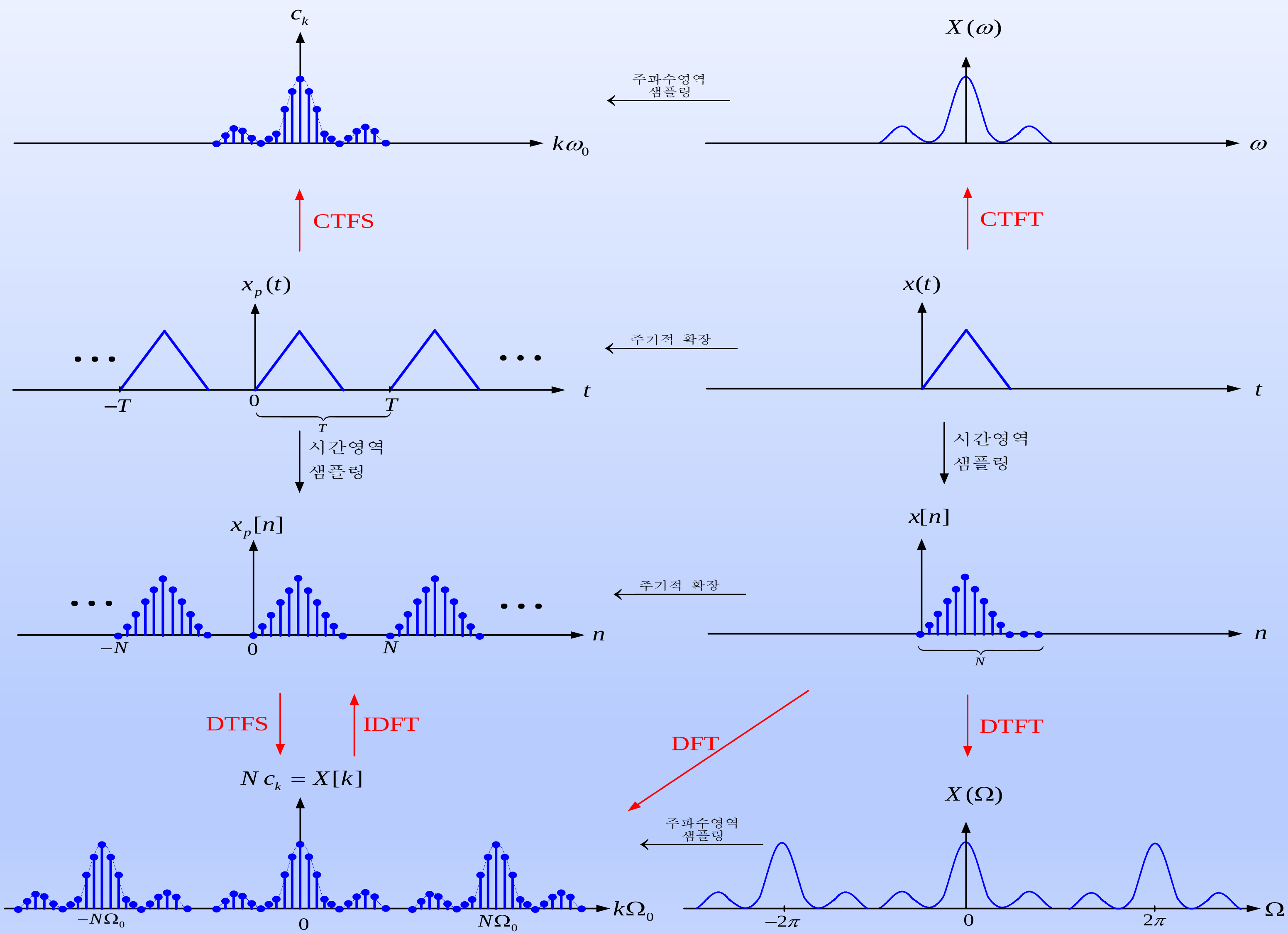
Discrete Fourier Transform

DFT와 DTFS의 관계



DFT와 DTFS의 관계

- DFT와 DTFS : 1/*N* 인수를 제외하고 동일

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-jk\frac{2\pi}{N}n},\quad k=0,1,\cdots,N-1 \quad \text{(DFT)}$$

$$C_k=\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{-jk\Omega_o n}=\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{-jk\frac{2\pi}{N}n} \quad \text{(DTFS 계수)}$$

$$\text{where}\ \ \Omega_0=\frac{2\pi}{N}\ \ \text{[rad], the fundamental (radian) frequency}$$

$$\therefore\ X[k]=N\,C_k$$

Fast Fourier Transform (FFT)

- DFT

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn},\qquad k=0,1,\cdots,N-1$$

Define $W_N \triangleq e^{-j\frac{2\pi}{N}}$, then

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{kn},\qquad k=0,1,\cdots,N-1$$

$$=x[0]W_N^{0}+x[1]W_N^{k}+x[2]W_N^{2k}+\cdots+x[N-1]W_N^{(N-1)k}$$

- 계산량
  - 각각의 $X[k]$: N 번의 복소 곱셈과 N 번의 복소 덧셈
  - N 개의 $X[k]$: N^2 번의 복소 곱셈과 N^2 번의 복소 덧셈

❑ FFT 알고리즘

- 짧은 길이의 신호로 분할 : 여러 DFT를 각각 계산하고 통합
- 길이가 절반인 두 신호로 분할 : 곱셈 수 1/4 씩, 총 1/2 감소
- FFT 알고리즘 : 데이터 길이 $N = 2^L$
- DIT(Decimation-In-Time) 알고리즘 : 입력 $x[n]$을 절반씩 분할
- DIF(Decimation-In-Frequency) 알고리즘 : 출력 $X[k]$를 절반씩 분할

❑ DIT Algorithm

- $x[n]$을 길이 $N/2$인 짝수-인덱스 수열과 홀수-인덱스 수열로 분할

$$X[k] = \sum_{n \text{ even}} x[n] W_N^{nk} + \sum_{n \text{ odd}} x[n] W_N^{nk}$$

- 짝수와 홀수를 각각 $n=2r$, $n=2r+1$로 표현

$$\begin{aligned} X[k] &= \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k} \\ &= \sum_{r=0}^{N/2-1} g[r] W_N^{2rk} + W_N^k \sum_{r=0}^{N/2-1} h[r] W_N^{2rk} \\ &= \sum_{r=0}^{N/2-1} g[r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} h[r] W_{N/2}^{rk} \end{aligned}$$

where $g[r] = x[2r],\ h[r] = x[2r+1]$

and $W_N^2 = \left[e^{-j\frac{2\pi}{N}}\right]^2 = e^{-j\frac{4\pi}{N}} = e^{-j\frac{2\pi}{N/2}} = \left[e^{-j\frac{2\pi}{N/2}}\right]^1 = W_{N/2}^1$

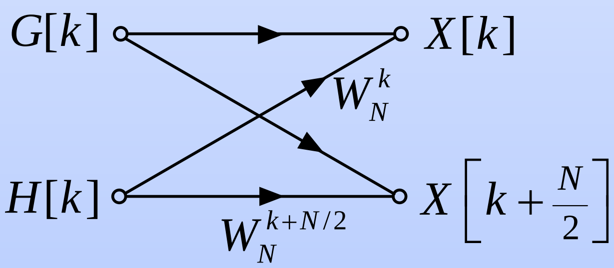
$$\therefore \quad X[k] = G[k] + W_N^k H[k], \quad k = 0, 1, \cdots, N-1$$

- $G[k]$와 $H[k]$의 주기는 $N/2$:

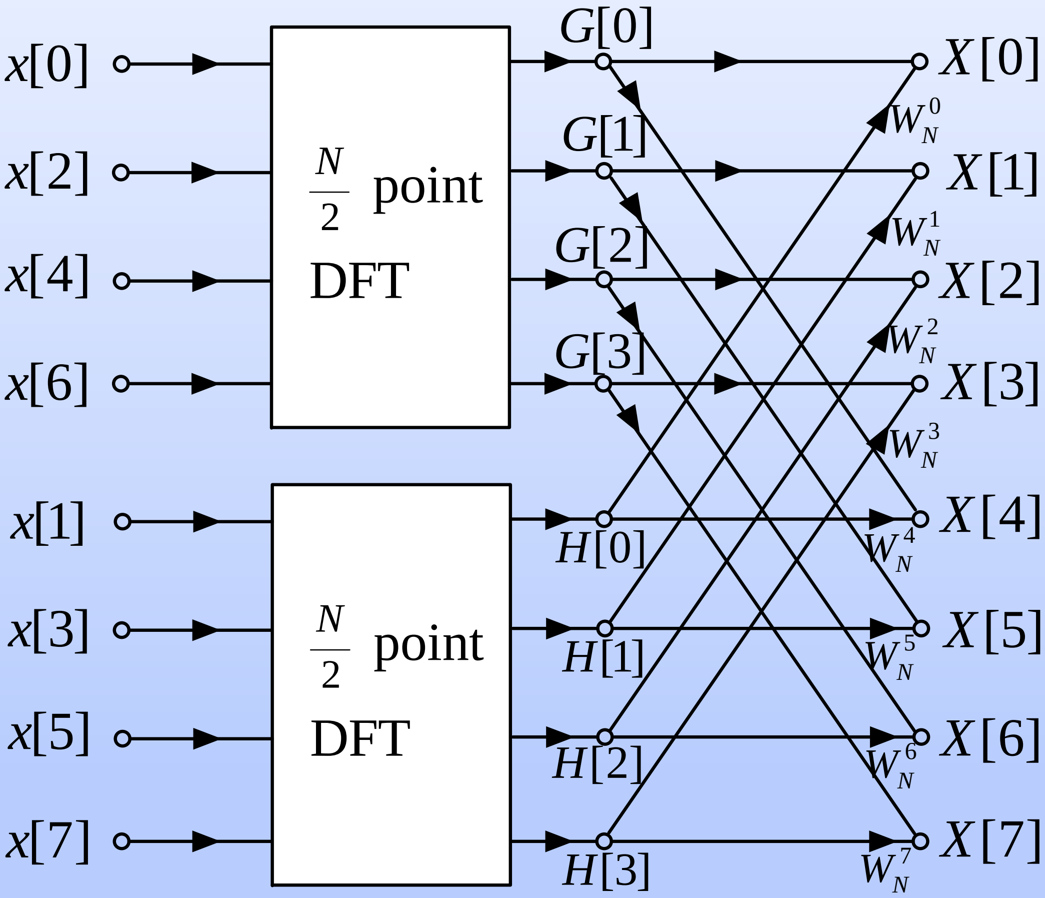
$$X[k]=G[k]+W_N^k H[k], \qquad k=0,1,\cdots,\frac{N}{2}-1$$

$$X\left[k+\frac{N}{2}\right]=G[k]+W_N^{k+N/2}H[k], \qquad k=0,1,\cdots,\frac{N}{2}-1$$

- N-point DFT : 두 개의 $N/2$-point DFT로 계산

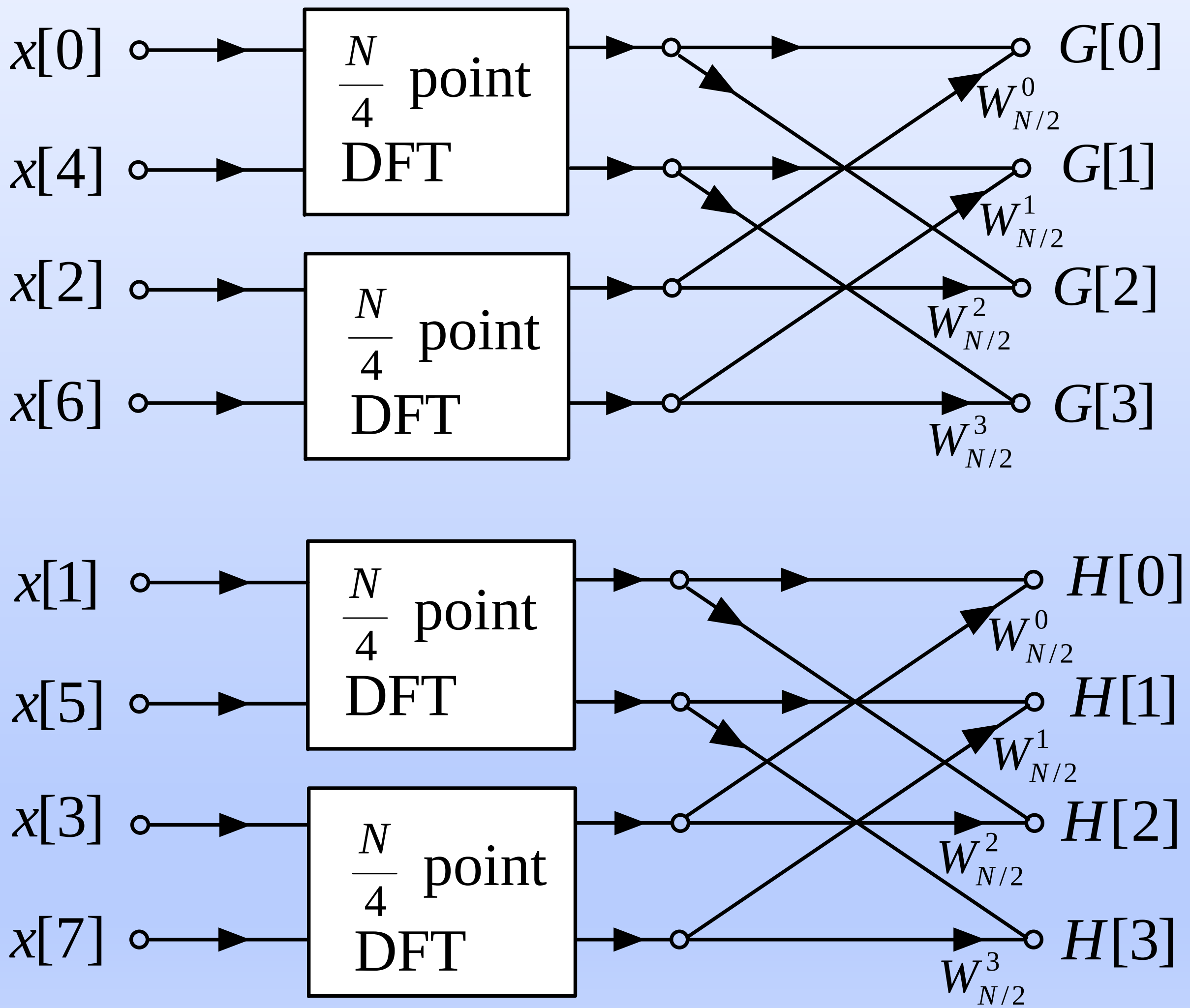


- $N = 8$ 인 경우:



x[0]
x[2]
x[4]
x[6]
x[1]
x[3]
x[5]
x[7]
N/2 point DFT
N/2 point DFT
G[0]
G[1]
G[2]
G[3]
H[0]
H[1]
H[2]
H[3]
X[0]
X[1]
X[2]
X[3]
X[4]
X[5]
X[6]
X[7]
W_N^0
W_N^1
W_N^2
W_N^3
W_N^4
W_N^5
W_N^6
W_N^7

❑ *N*/2-point DFT : 두 개의 *N*/4-point DFT로 분활



❑ 반복 적용

- 가장 작은 단위의 DFT는 2-point DFT : 곱셈 연산 불필요

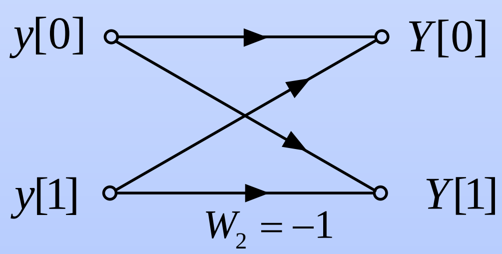
$$Y[k] = \sum_{n=0}^{1} y[n] W_2^{kn}, \qquad k = 0,1$$

$$= y[0] + y[1] W_2^{k}$$

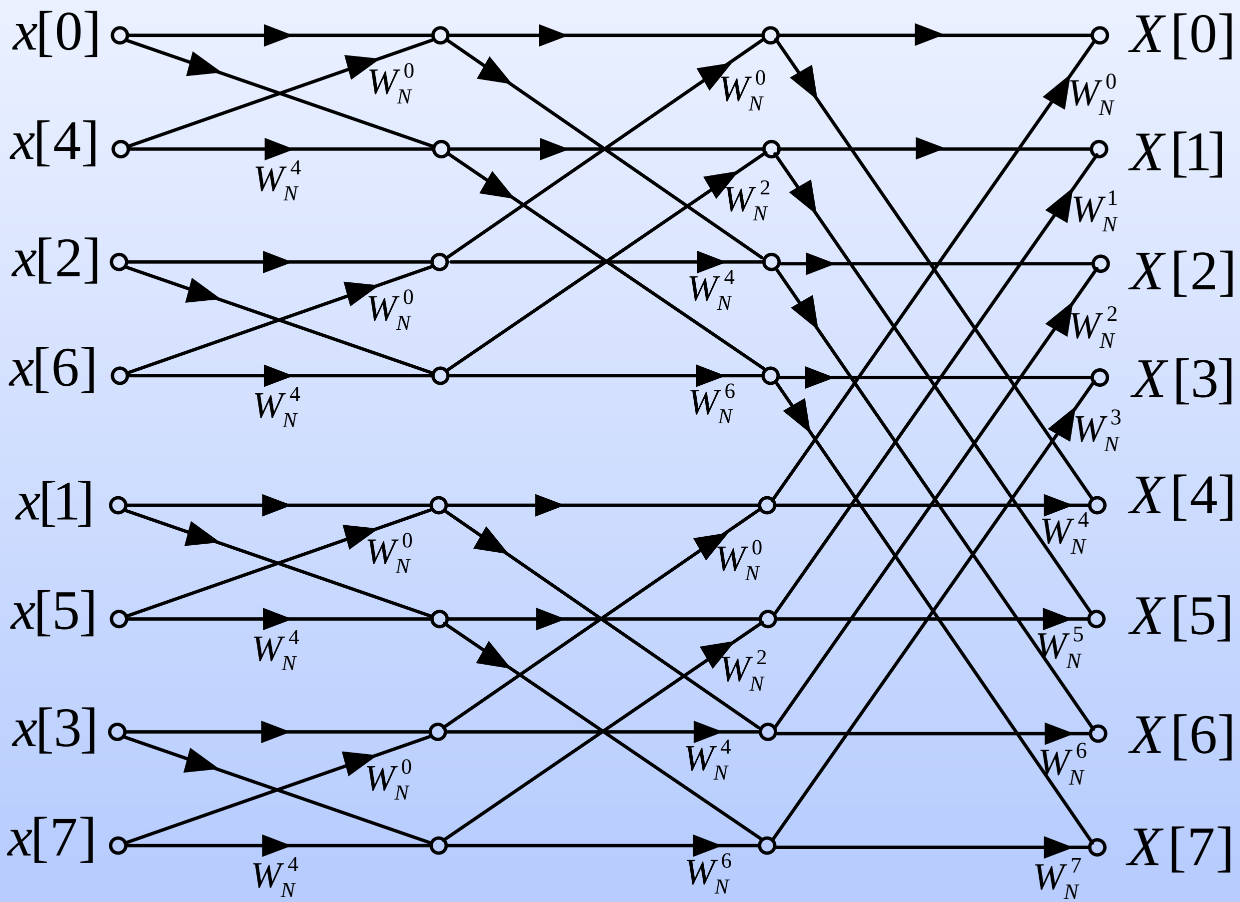
Since $W_2^0 = 1,\ W_2^1 = e^{-j\frac{2\pi}{2}} = -1$

$$Y[0] = y[0] + y[1]$$

$$Y[1] = y[0] - y[1]$$



- $\log_2 N$ 개의 stage ($N = 8 = 2^3$ 이면 3개의 stage)



- Stage마다 N 개의 곱셈과 덧셈 : 총 $N\log_2 N$ 곱셈과 덧셈

- $N^2 : N \log_2 N = \left(2^{10}\right)^2 : 2^{10}(10) = 1024 : 10 \quad if\ \ N = 1024 = 2^{10}$