高速 Fourier 變換(I)

李 太 遠

高麗大學校 電子工科大學 教授(工博)

1. 머 릿 말

FFT라는 이름으로 알려져 있는 高速 Fourier 變換의 앨고리즘은 1965年에 J. W. Cooley 와 J. W. Tukey에 의해서 발표되었다.[1] FFT를 이용하므로서, 종래에 수십분이나 걸렸던 1,024 쎔플에 대한 數値的 Fourier 變換(DFT)를 단지 수십초에 끝낼 수 있게 되었는데 이 技法은 波形解析이 필요한 研究의 발전에 공헌하고 있을 뿐만 아니라 相關函數를 구하는 계산, 디지탈필터링같은 분야에서 實用的인 시간내에 계산을 실행할 수 있게끔한 意義가 크다.

FFT의 원리를 설명하기 위하여 우선 連續函數의 Fourier 變換式에서 離散的 Fourier 變換(DFT)라고 하는 數値計算에 有用한 計算式을 유도한다. 다음으로 그것을 바탕으로 하여 高速 Fourier 變換計算의 원리를 설명하고 프로그램 作成에 必要한 프로우챠트, 실제로 프로그램作成에서의 고려할 점과 프로그램을 설명하고져 한다.

2. 離散的 Fourier 変換(Discrete Fourier Transform, DFT)

電算機에 의한 數値的인 Fourier 變換을 다루기 위한 준비단계로서 Fourier 變換의 公式을 출발점으로 하여 離散的 Fourier 變換(DFT)와 그 逆變換(IDFT)의 定義를 설명하고져 한다.

Fourier 變換에서 時間 t 의 函數인 $g(t)$와 周波數 f 의 函數 $G(f)$ 와의 사이의 關係式을 다음과 같이 주고 있다.

$$\left.\begin{aligned} G(f) &= \int_{-\infty}^{\infty} g(t)\,exp(-j2\pi ft)\,dt \\ g(t) &= \int_{-\infty}^{\infty} G(f)\,exp(j2\pi ft)\,df \end{aligned}\right\} \qquad (1)$$

$G(f)$는 時間函數(波形) $g(t)$의 周波數스펙트럼이며 처음의 식을 Fourier 變換, 다음의 것을 Fourier 逆變換이라고 한다.

만일 $g(t)$가 $0 \leq t \leq T$의 時間區間밖에서는 항상 零이고 주파수스펙트럼 $G(f)$는 FHz 以下로 帶域이 제한되어 있다면 式(1)의 첫式의 積分區間을 $0 \sim T$로 하고 두번째式의 적분구간을 $-F \sim +F$로 할 수 있다. 이 경우의 時間函數 $g(t)$의 Nyquist 쌤플値 $g_k\ (k=0,1,2,\cdots,N-1)$는

$$g_k = g(k/2F) \quad k=0,1,2,\cdots,N-1 \quad (2)$$

로 표시할 수 있다. $g(t)$를 周期가 T초인 波形이라고 보면 周波數스펙트럼은 $1/T$ Hz마다의 線스펙트럼이 되며 $g(f)$의 $1/T$ Hz마다의 값 Gp는

$$Gp = G(p/T) \qquad (3)$$

가 되며 여기서

$$p = -\frac{N}{2},\ -\frac{N}{2}+1,\ \cdots,\ -1,0,1,\cdots,\frac{N}{2}-1$$

이다. $g(t)$가 非週期函數가 아닌 경우에도 式(3)으로 주어지는 Gp의 값이 정하여지면 周波數의 連續函數로서의 $G(f)$의 형태는 一意的으로 결정된다. 샘플링定理를 만족하는 샘플링의 총수 N과 周波數帶域 F 및 時間幅 T와의 사이에는 다

음에 관계가 있다.

$$N = 2FT \tag{4}$$

이들의 관계를 式(1)에 넣고 $dt \rightarrow \frac{1}{2F}$, $df \rightarrow \frac{1}{T}$라고 하면

$$\left.\begin{aligned} G_p &= \frac{1}{2F}\sum_{k=0}^{N-1} g_k\, exp(-j2\pi pk/N) \\ g_k &= \frac{1}{T}\sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} G_p\, exp(j2\pi pk/N) \end{aligned}\right\} \tag{5}$$

라고 할 수 있다. 그런데

$$exp\left\{j2\pi k(-\frac{N}{2}+r)/N\right\} = exp(-j\pi k)\cdot exp(j2\pi kr/N)$$

또한 k는 整數이므로

$$exp(-j\pi k) = exp(j\pi k)$$

따라서

$$exp\left\{j2\pi k(-\frac{N}{2}+r)/N\right\} = exp\left\{j2\pi k(\frac{N}{2}+r)/N\right\}$$

式(5)의 두번째式의 G_p에 대하여 첫式을 사용하여 위에서와 같이 고쳐 쓰면

$$\begin{aligned} G_{-\frac{N}{2}+r} &= \frac{1}{2F}\sum_{k=0}^{N-1} g_k\, exp\left\{-j2\pi(-\frac{N}{2}+r)\,k/N\right\} \\ &= \frac{1}{2F}\sum_{k=0}^{N-1} g_k\, exp\left\{-j2\pi(\frac{N}{2}+r)\,k/N\right\} \\ &= G_{\frac{N}{2}+r} \end{aligned}$$

따라서 式(5)의 두개의 式을 통합하여

$$\begin{aligned} g_k &= \frac{1}{T}\sum_{p=0}^{N-1} G_p\, exp(j2\pi pk/N) \\ &= \frac{1}{N}\sum_{p=0}^{N-1}\sum_{l=0}^{N-1} g_l\, exp\left\{j2\pi\frac{p}{N}(k-l)\right\} \end{aligned} \tag{6}$$

와 같이 정리할 수 있다.

이것을 두개로 나누어서 DFT와 IDFT를 다음과 같이 定義한다.

$$\left.\begin{aligned} DFT: G_p &= \sum_{k=0}^{N-1} g_k W^{-pk} \\ IDEF: g_k &= \frac{1}{N}\sum_{p=0}^{N-1} G_p W^{pk} \end{aligned}\right\} \tag{7}$$

여기서

$$W = exp(j\frac{2\pi}{N})$$

式(7)을 正弦波函數로서 고쳐 쓰면

$$\left.\begin{aligned} DFT: G_p &= \sum_{k=0}^{N-1} g_k(cos\frac{2\pi pk}{N} - j\, sin\frac{2\pi pk}{N}) \\ IDET: g_k &= \frac{1}{N}\sum_{k=0}^{N-1} g_k(cos\frac{2\pi pk}{N} + j\, sin\frac{2\pi pk}{N}) \end{aligned}\right\} \tag{8}$$

時間函數나 周波數函數는 複素數라도 무방하지만 時間函數가 實數일 경우에는 周波數函數G_p를 實數部 G_{rp}와 虛數部 G_{ip}로 나누어서 다음과 같이 나타낼 수 있다.

$$\left.\begin{aligned} DFT: G_{rp}+j\,G_{ip} &= \sum_{k=0}^{N-1} g_k(cos\frac{2\pi pk}{N} - j\, sin\frac{2\pi pk}{N}) \\ IDET: g_k &= \frac{1}{N}\sum_{p=0}^{N-1}(G_{rp}\; cos\frac{2\pi pk}{N} - G_{ip}\; sin\frac{2\pi pk}{N}) \end{aligned}\right\} \tag{9}$$

3. 高速 Fourier 變換(Fast Fourier Transform, FFT)

高速Fourier 變換은 앞에서 유도한 式(7), (8) 또는 (9)의 數値計算을 효율적으로 실행하는 앨고리즘이다. 예를 들어서 Fourier變換으로서 式(7)의 첫 式에서 Gp의 한 값을 구하기 위해서는 g_k와 W^{-pk}와의 複素數의 積의 計算을 N回 실행하여야 하며, 이 작업을 p의 모든 값(O~N−1)에 대해서 실행하기 위해서는 결국 N^2回

의 積의 計算이 필요하다. 그러나 이 積의 計算의 回數는 다음의 설명으로 알 수 있듯이, N이 클 경우에는 대폭적으로 감소시킬 수 있다.

샘플의 총수 N이 두개의 整數 n_1과 n_2와의 積으로 되어 있다고 가정하면 $N = n_1 \cdot n_2$이므로 式(7)의 k와 p를 다음과 같이 쓸 수 있다.

$$k = k_1 n_1 + k_0 \begin{cases} k_0 = 0, 1, 2, \cdots, n_1 - 1 \\ k_1 = 0, 1, 2, \cdots, n_2 - 1 \end{cases}$$

$$p = p_1 n_1 + p_0 \begin{cases} p_0 = 0, 1, 2, \cdots, n_1 - 1 \\ p_1 = 0, 1, 2, \cdots, n_2 - 1 \end{cases}$$

이것을 이용하면 式(7)을 다음과 같이 고쳐 쓸 수 있다.

$$G_p = \sum_{k_0=0}^{N_1-1} \sum_{k_1=0}^{N_2-1} g_{k_1 n_1 + k_0} \; exp\left\{-j2\pi\left(k_1 p_1 + \frac{k_0 p_1}{n_1} + \frac{k_1 p_0}{n_2} + \frac{k_1 p_0}{n_1 n_2}\right)\right\}$$

k_1과 p_1는 整數이므로

$$exp(-j2\pi k_1 p_1) = 1$$

따라서

$$G_p = \sum_{k_0=0}^{N_1-1} \sum_{k_1=0}^{N_2-1} g_{k_1 n_1 + k_0} \; exp\left\{-j2\pi\left(\frac{k_0 p_1}{n_1} + \frac{k_1 p_0}{n_2} + \frac{k_0 p_0}{n_1 n_2}\right)\right\}$$

$$= \sum_{k_0=0}^{N_1-1} g(k_0, p_0) \; exp\left\{-j2\pi\left(\frac{k_0 p_1}{n_1} + \frac{k_0 p_0}{n_1 n_2}\right)\right\} \quad (10)$$

여기서

$$g(k_0, p_0) = \sum_{k_1=0}^{N_2-1} g_{k_1 n_1 + k_0} \; exp\left(-j2\pi \frac{k_1 p_0}{n_2}\right)$$

式(8)에서 積의 計算回數는 k_1에 대해서 n_2回 실행한 것을 k_0와 p_0에 대해서 각각 n_1回, n_2回 실행하게 되므로 전체적으로 $n_1 n_2^2 = n_2$ N回만큼 積의 計算이 필요하다. 그 결과를 式(10)에 넣어서 計算할 경우에도 積의 計算回數는 먼저와 같이 해서 n_1 N回가 필요하다. 합계하면 積의 計算의 總回數는 $(n_1 + n_2)$N回이며 이 回數는 式(7)의 첫 式을 직접 計算하는 것보다 일반적으로 적다. 式(7)의 두번째 式에 대해서도 積의 計算回數는 $(n_1 + n_2)$N回가 된다.

나아가서 n_1과 n_2가 각각

$$n_1 = n_{11} \cdot n_{12}, \quad n_2 = n_{21} \cdot n_{22}$$

와 같이 整數의 積의 형태로 표현될 경우에는 n_1^2回의 積의 計算回數는 $(n_{11} + n_{12})n_1$回로, n_2^2回의 積의 回數는 $(n_{21} + n_{22})n_2$回로 압축되어, 積의 計算의 總回數는 $(n_{11} + n_{12} + n_{21} + n_{22})$N回로 줄어드는데 이러한 과정을 계속하여 해나갈 수 있다.

$N = r^m$으로 나타낼 때 이러한 計算의 궁극적인 回數는 mrN回, 즉

$$r N \log_r N回$$

가 되는데 이것과 式(7)의 첫 式을 직접 計算했을 경우의 積의 計算回數와의 比를 R이라 하면

$$R = \frac{r N \log_r N}{N^2} = \frac{m}{r^{m-1}}$$

가 된다. r이나 m도 整數이므로 샘플의 總數 $N = r^m$은 어떠한 數라도 되는 것은 아니다. 그러나 샘플의 사간간격이 샘플링定理를 만족한다는, DFT를 위한 필요조건을 만족하기만 하면 위의 제한은 실용상에 문제가 되지 않는다. r의 값의 선택에 있어 $r^m = N$이 바람직한 어떤 數가 되도록 하는 것보다 오히려 위의 式의 R을 작게 한다는 기준으로 생각할 필요가 있다.

表1에 r = 2 ~ 8에 대해서 각각의 m에 값에 대한 N과 R을 나타내었다. 이 表에서는 r = 2 ~ 5의 범위에서 거의 차이가 없다. 그러나 r ≥ 7에서는 확실히 불리하다. 따라서 N에 대한 특별한 요구가 없는 한, r = 2로 하는 것이 보통이다. 따라서 앞으로는 r = 2의 경우만을 고려하기로 한다.

표 1. FFT를 이용한 DFT와 직접계산과의 積의 計算回數의 比 R의, r과 m에 따른 차이

N	r^m	R	r^m	R	r^m	R	N	r^m	R	r^m	R	r^m	R
32	2^5	0.316					243	3^5	0.0617				
36	6^2	0.333					256	2^8	0.0625	4^4	0.0625		
49	7^2	0.286					343	7^3	0.0612				
64	2^5	0.187	4^3	0.187	8^2	0.25	512	2^0	0.035			8^3	0.0469
81	3^4	0.148					625	5^4	0.032				
125	5^3	0.120					729	3^6	0.0247				
128	2^7	0.109					1024	2^{10}	0.0195	4^5	0.0195		
216	6^3	0.0834					1296	6^4	0.0185				

4. 信号흐름図에 의한 検討

앞에서 설명하였듯이 여러 개의 샘플値에 대한 DFT를 少數個의 샘플마다 나누어서 실행하므로서 DFT를 위한 積의 計算의 總回數를 줄일 수 있는 것이 FFT의 算法의 원리이다. 앞에서의 설명만으로는 프로그램을 작성하는 구체적인 방법을 생각해내기가 어려우므로 J. W. Cooley와 J. W. Tukey가 제시한 방법[2)]으로 샘플總數N이 8인 경우를 예로 하여 計算의 절차를 설명하기로 한다.

우선 그림 1과 같이 N = 8개의 샘플値로 주어지는 時系列 g_k($k = 0, 1, 2, \cdots, 6, 7$ (N−1)) 를 하나 걸러 b_k와 c_k로 나눈다. b_k는 偶數번째, c_k는 奇數번째의 샘플値이다. 즉

$$\begin{aligned} b_k &= g_{2k} \\ c_k &= g_{2k} \end{aligned} \qquad k = 0, 1, 2, \cdots, \frac{N}{2} - 1 \qquad (11)$$

b_k와 c_k는 각각 다음의 식으로 주어지는 DFT를 가진다.

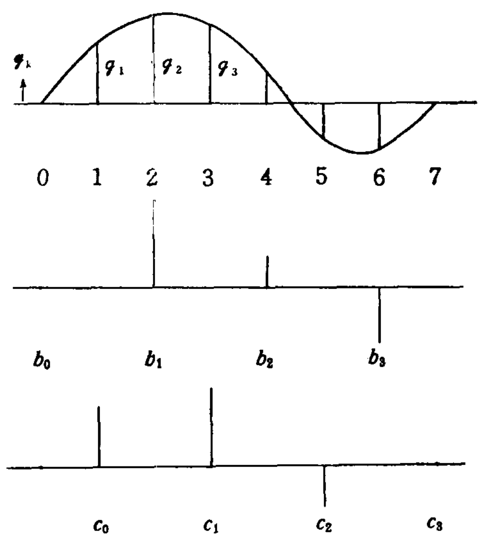

그림 1. N = 8개의 샘플値系列 g_k를 하나 걸러 b_k와 c_k로 나눈다.

$$\left.\begin{aligned} B_r &= \sum_{k=0}^{\frac{N}{2}-1} b_k W^{-2rk} \\ C_r &= \sum_{k=0}^{\frac{N}{2}-1} c_k W^{-2rk} \end{aligned}\quad r = 0, 1, 2, \cdots, \frac{N}{2} - 1 \right\} (12)$$

여기서 $W=exp(j\frac{2\pi}{N})$이며 g_k의 Fourier變換(DFT) G_r를 구하는데 있어서 다음과 같이 분해하여 B_r과 C_r로서 나타낼 수 있다.

$$\begin{aligned}G_r&=\sum_{k=0}^{N-1} g_k W^{-rk}\\&=\sum_{k=0}^{\frac{N}{2}-1}\left\{b_k W^{-2rk}+c_k W^{-r(2k+1)}\right\}\\&=\sum_{k=0}^{\frac{N}{2}-1}\left\{b_k W^{-2rk}+W^{-r}c_k W^{-2rk}\right\} \qquad (13)\\&=B_r+C_r W^{-r}\\&=B_r+C_r\, exp(-j2\pi r/N)\end{aligned}$$

$$r=0,1,2,\cdots,\frac{N}{2}-1$$

$r\geq(N/2)$의 범위의 r의 값에 대해서는 B_r과 C_r은 $0\leq r\leq\frac{N}{2}-1$에서의 값을 반복하여 巡回的인 값을 가지는 것으로 하면

$$\left.\begin{aligned}C_{\frac{N}{2}+r}&=B_r+C_rW^{-(\frac{N}{2}+r)}=B_r+C_r\,exp\left\{-j2\pi(\frac{N}{2}+r)/N\right\}\\&=B_r-C_r\,exp\left\{-j2\pi r/N\right\}=B_r-C_rW^{-r}\\r&=0,1,2,\cdots,\frac{N}{2}-1\end{aligned}\right\}(14)$$

에 따른 $r\geq(N/2)$의 범위의 DFT가 주어진다.

즉 g_k를 하나 걸러 취한 b_k와 c_k의 DFT B_r과 C_r만을 알면 B_r과 C_r을 平面벡터로 생각하고 C_r가 $exp(-j2\pi r/N)\equiv W^{-r}$만큼 回轉한 벡터를 B_r에 가하거나, B_r로부터 뺌으로서 g_k $(k=0,1,2,\cdots,N-1)$의 DFT G_r $(r=0,1,2,\cdots,N-1)$를 구할 수 있다. C_r의 회전은 円周를 그림 2에서와 같이 N등분하여 시계방향으로 r/N回轉시키는 것으로 이해할 수 있다.

이러한 절차를 信號의 흐름圖로 나타내면 그림 3이 된다.

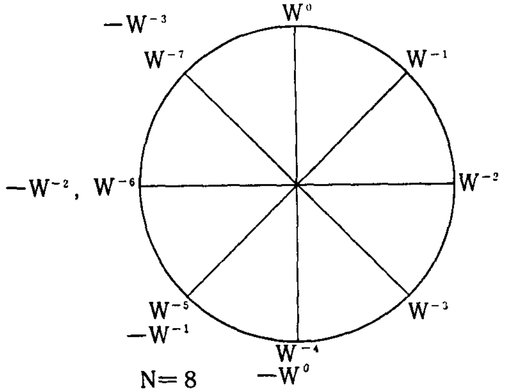

그림 2. 位相項W^{-r}은 r의 증가에 따라 시계방향으로 회전한다. $r\geq N/2+n$에서는 $r=n$때와 반대방향이 된다.

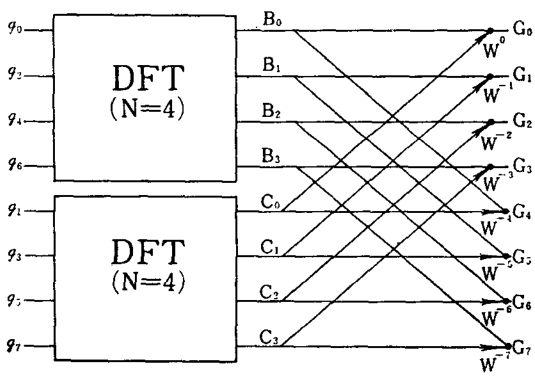

그림 3.

이 그림에서 左端의 g_k $(k=0,1,2,\cdots,7)$를 偶數번째와 奇數번째의 것으로 나누어서 각각에 대하여 DFT를 실행한 것이 B_k $(k=0,1,2,3\,(=\frac{N}{2}-1))$와 $C_k(k=0,1,2,3)$이다. 式(13)과 같이 B_0, B_1, B_2, B_3의 각각에 $C_rW^{-r}(r=0,1,2,3)$를 더한것이 G_0, G_1, G_2, G_3이며 式(14)와 같이 B_0, B_1, B_2, B_3의 각각에서 $C_rW^{-r}(r=0,1,2,3)$를 감한 것이 G_4, G_5, G_6, G_7이다. 이러한 과정을 數式으로 나타내면 다음과 같다.

$$\left.\begin{aligned}
G_0 &= B_0 + W^{-0} C_0 \\
G_1 &= B_1 + W^{-1} C_1 \\
G_2 &= B_2 + W^{-2} C_2 \\
G_3 &= B_3 + W^{-3} C_3 \\
G_4 &= B_0 + W^{-4} C_0 = B_0 - W^{-0} C_0 \\
G_5 &= B_1 + W^{-5} C_1 = B_1 - W^{-1} C_1 \\
G_6 &= B_2 + W^{-6} C_2 = B_2 - W^{-2} C_2 \\
G_7 &= B_3 + W^{-7} C_3 = B_3 - W^{-3} C_3
\end{aligned}\right\} \quad (15)$$

B_r과 $C_r(r=0,1,2,3)$을 구하는 計算도 위에서와 같은 방식으로 g_1, g_2, g_4, g_6의 奇數번째(g_0, g_4)의 DFT D_0, D_1과 偶數번째(g_2, g_6)의 DFT E_0, E_1에 W^0, W^{-2}을 곱한 것의 合과 差라는 형태로 구할 수 있다. 이것을 信號흐름圖로 나타내면 그림 4와 같다. 이 과정을

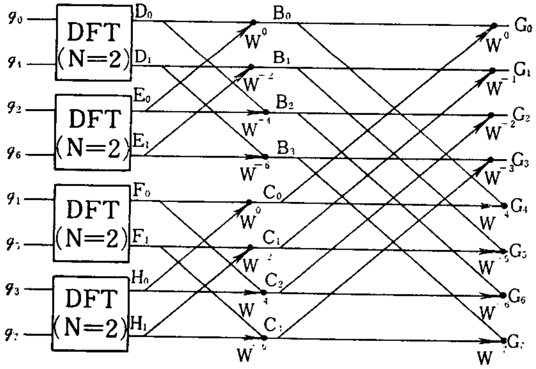

그림 4.

이 과정을 數式으로 나타내면 다음과 같다. 그림 4와 다음의 式(16)에서 $F_r, H_r\ (r=0,1)$는 각각 g_1, g_3, g_5, g_7의 奇數번째(g_1, g_5)와 偶數번째(g_3, g_7)의 DFT이다.

$$\left.\begin{aligned}
B_0 &= D_0 + W^{-0} E_0 \\
B_1 &= D_1 + W^{-2} E_1 \\
B_2 &= D_0 + W^{-4} E_0 = D_0 - W^{-0} E_0 \\
B_3 &= D_1 + W^{-6} E_1 = D_0 - W^{-2} E_1 \\
C_0 &= F_0 + W^{-0} H_0 \\
C_1 &= F_1 + W^{-2} H_1 \\
C_2 &= F_0 + W^{-4} H_0 = F_0 - W^{-0} H_0 \\
C_3 &= F_1 + W^{-6} H_1 = F_0 - W^{-2} H_1
\end{aligned}\right\} \quad (16)$$

마지막으로 $D_r, E_r, F_r, H_r\ (r=0,1)$은 g_k ($k=0,4,2,6,1,5,3,7,$)의 두개씩에서부터 직접 구할 수 있다. 이 과정을 信號흐름圖로 나타내면 그림 5와 같으며 여기서 실행되는 D_r, E_r, F_r, H_r을 구하는 計算式은 다음과 같다.

$$\left.\begin{aligned}
D_0 &= g_0 + g_4 & D_1 &= g_0 - g_4 \\
E_0 &= g_2 + g_6 & E_1 &= g_2 - g_6 \\
F_0 &= g_1 + g_5 & F_1 &= g_1 - g_5 \\
H_0 &= g_3 + g_7 & H_1 &= g_3 - g_7
\end{aligned}\right\} \quad (17)$$

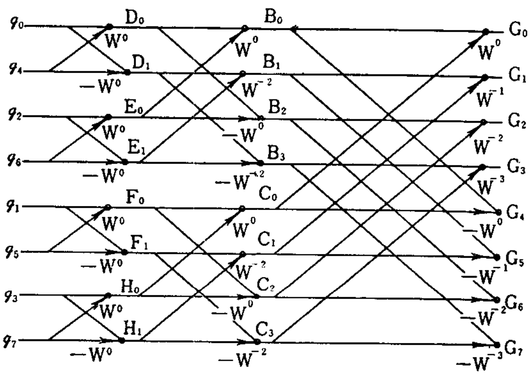

그림 5.

그림 5의 左端의 g_r의 添字 r의 순서가 0,4,2,6,1,5,3,7로 되어 있어서 불규칙적으로 보이지만 이것은 0,1,2,3,4,5,6,7을 奇數번째, 偶數번째의 순서로 차례로 나열순서를 바꾸어 놓은 결과이다. 이것을 2進數로 바꾸면 表 2와 같다. 결과

표 2. 0～7의 逆빗트의 순서나열 바꿈

r	2進數	첫번째 순서바꿈		두번째 순서바꿈	10進數
0	000	0	000	000	0
1	001	2	010	100	4
2	010	4	100	010	2
3	011	6	110	110	6
4	110	1	001	001	1
5	101	3	011	101	5
6	110	5	101	011	3
7	111	7	111	111	7

만을 쓰면 $0 \sim N-1(=2^M-1)$의 수를 M자리의 2進數로 나타내고 그 左右의 자리의 순서를 반대로 한것을 10進數로 바꾸면 최종적인 순서를 기계적으로 알아낼 수 있다. 이 과정을 逆빗트순서에 따른 나열순서바꿈이라 한다.

지금까지의 과정을 정리하면, $g_0, g_1, \cdots, g_7$라는 系列로 주어진 샘플值의 순서를 우선 逆빗트의 순서에 따른 나열바꿈을 한 다음 式(17)에 의하여 $D_r, E_r, F_r, H_r (r=0,1)$를 구하고 나아가서 式(16)에 의하여 $B_r, C_r (r=0,1,2,\cdots 7)$을 구하므로서 g_r이라는 샘플系列의 DFT를 計算한다.

積의 計算은 $W^r (r=0,1,2,\cdots,\frac{N}{2}-1)$과 샘플値, 또는 중간적인 DFT의 값과의 積의 計算이 된다. 그림 5에서 積의 計算回數를 조사하면 그림속의 화살의 개수와 같아서 24回가 된다. 또한 같은 計算이 두번씩 있음에 주의하여 한번計算한 積을 記憶裂置에 넣어두었다가 사용하는 방식으로 하면 積의 計算의 回數는 절반이 되어 12回가 된다. 이것을 앞의 式(10)에서 사용한 回數 $rN\log_r N$回에, 이 경우의 값을 대입한 $rN\log_r N = 2\times8\times3=48$回와 비교하면 1／4이 되어 $r=2$로 놓고 실행하는 FFT에서의 積의 計算의 回數는 $\frac{1}{2}N\log_2 N$回까지 줄일 수 있다.

그림 5에 나타낸 방법으로 FFT의 계산을 하려면 우선 逆빗트의 순서나열바꿈을, 入力샘플系列에 대해서 하여야 한다. 이 과정이 불편하다면 逆빗트의 순서나열바꿈을 최종과정으로 돌릴 수 있다. 그렇게 하기위해서는 그림 3과 같이 入力샘플系列 g_k를 奇數번째것과 偶數번째것으로 나누어서 DFT하는 것과는 반대로 그림 6과 같이 g_k의 DFT G_k를 奇數번째것과 偶數번째것으로 나누어서 그 각각이 $N=4$의 DFT로 구성되어 있으므로 그림 7, 그림 8의 순서대로 진행한다.

그림 8의 右端에는 $G_0, G_4, G_2, G_6, G_1, G_5, G_3, G_7$의 차례로 나열되어 있으므로 이것을 逆빗트순서나열바꿈을 함으로써 $G_0, G_1, G_2, \cdots, G_7$로 나열된 DFT를 구할 수 있다. 積의 計算의 回數등은 먼저의 경우와 같다.(다음號에 계속)

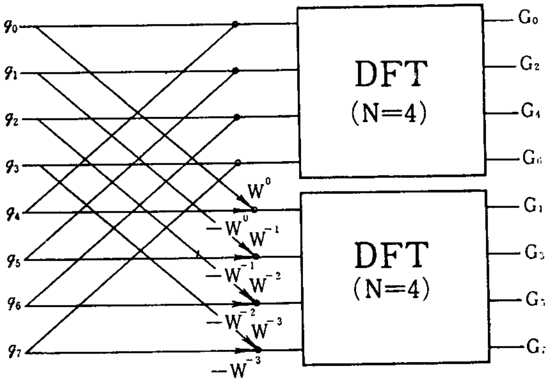

그림 6.

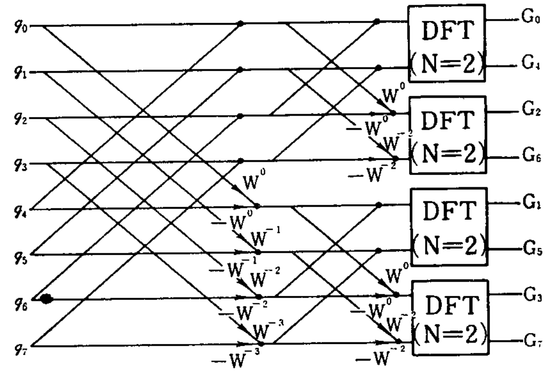

그림 7.

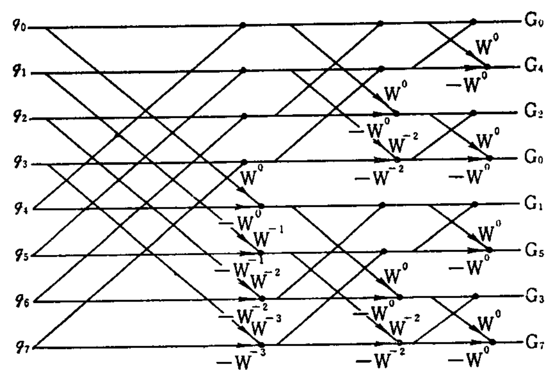

그림 8.