CALCOLO EFFICIENTE DELLA DFT (FFT)

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

La formula richiede $N(N-1)$ prodotti e $N(N-1)$ somme complesse.
Quindi l'ordine di complessità è $O(N^2)$.
Grazie alle proprietà del kernel $e^{-j\frac{2\pi}{N}kn}$ è possibile semplificare notevolmente il calcolo.

Assumiamo che N sia una potenza intera di 2, ovvero che $N=2^b$. Questo non è di fatto una limitazione alla applicabilità dell'algoritmo perché, dato un certo N_0, si può trovare l'$N=2^b$ immediatamente superiore e introdurre $N-N_0$ zeri nella sequenza $x[n]$ senza introdurre alcun cambiamento nel risultato.
La ragione per la scelta $N=2^b$ è che l'algoritmo efficiente è basato su dimezzamenti successivi.
Vediamo come.

ALGORITMO DELLA DECIMAZIONE NEL TEMPO

Riscriviamo l'espressione della DFT dividendo il contributo dei campioni di indice pari e di quelli di indice dispari.

$$X[k] = \sum_{n\ \text{pari}}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn} + \sum_{n\ \text{dispari}}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{r=0}^{\frac{N}{2}-1} x[2r]\, e^{-j\frac{2\pi}{N}k2r} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]\, e^{-j\frac{2\pi}{N}(2r+1)k}$$

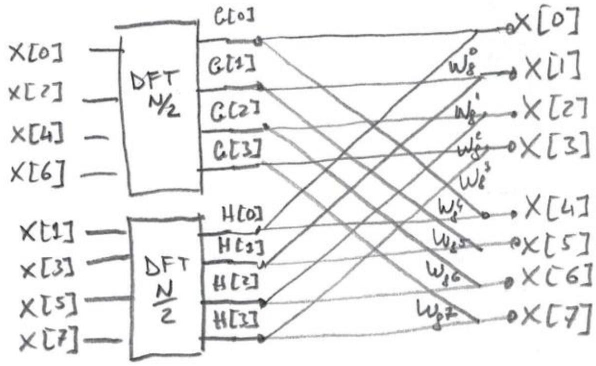
$$= \underbrace{\sum_{r=0}^{\frac{N}{2}-1} x[2r]\, e^{-j\frac{2\pi}{N/2}kr}}_{\text{DFT di ordine } \frac{N}{2} \text{ sui campioni pari.}} + e^{-j\frac{2\pi}{N}k} \underbrace{\sum_{r=0}^{\frac{N}{2}-1} x[2r+1]\, e^{-j\frac{2\pi}{N/2}kr}}_{\text{DFT di ordine } \frac{N}{2} \text{ sui campioni dispari.}}$$

$$= G[k] + e^{-j\frac{2\pi}{N}k} H[k]$$

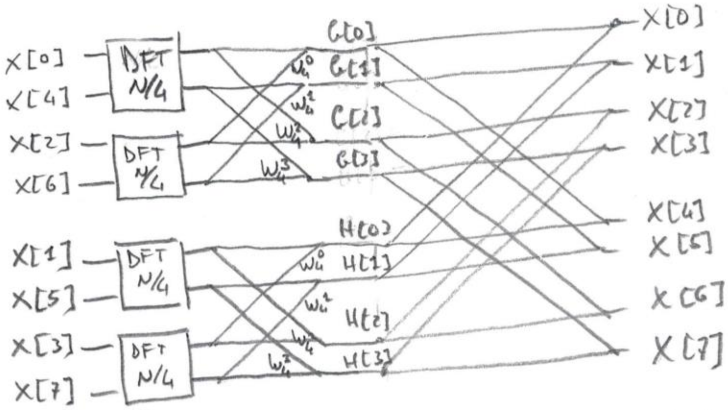
L'operazione può essere scomposta come nel grafo di figura.
Ricordiamo che $G[k]$ e $H[k]$ sono periodiche con periodo $\frac{N}{2}$, ovvero

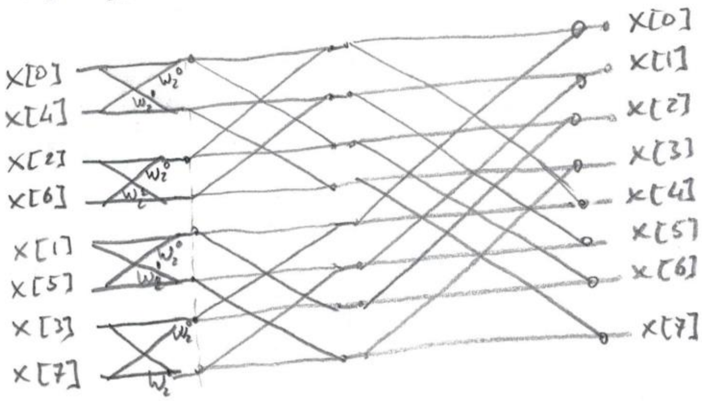
$G[k]=G[k+\frac{N}{2}]$, $H[k]=H[k+\frac{N}{2}]$; $W_N \triangleq e^{-j\frac{2\pi}{N}}$ FFT.2

Visualizzazione per $N=8=2^3$



Dividendo ulteriormente.



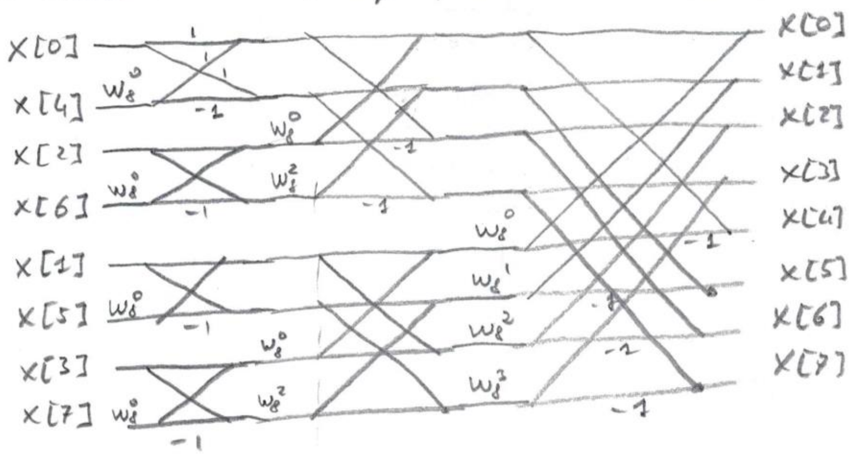


bastano $b=\log_2 N$ stadi per raggiungere la dimensione minima che è una DFT di ordine 2

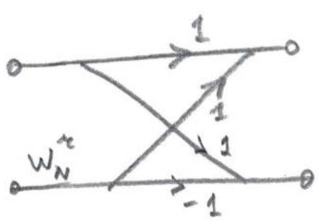
Notando che $W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{N}(k+\frac{N}{2})} = -W_N^k$

$$\begin{cases} X[k] = G[k] + W_N^k H[k] & k = 0, \dots, \frac{N}{2}-1 \\ X[k] = G[k] - W_N^k H[k] & k = \frac{N}{2}, \dots, N-1 \end{cases}$$

La struttura si semplifica ulteriormente



La "farfalla" base è del tipo



Per ogni farfalla serve

1 moltiplicazione complessa

2 somme complesse.

(ricordiamo che i valori sono in generale tutti complessi)

Per ogni stadio abbiamo $\frac{N}{2}$ farfalle, quindi in totale serve

$$\begin{cases} \frac{N}{2} \lg_2 N \text{ moltiplicazioni complesse} \\ N \lg_2 N \text{ somme complesse} \end{cases}$$

L'ordine di complessità è pertanto

$$O(N \lg N)$$

L'ordine in cui i campioni all'ingresso sono presentati nel grafo appare peculiare. Esso è il risultato di selezionare pari e dispari e poi pari e dispari in ogni sottoinsieme. Per $N=8$ abbiamo ottenuto

	indice in BCD	
$x[0]$	000	000
$x[4]$	100	001
$x[2]$	010	010
$x[6]$	110	011
$x[1]$	001	100
$x[5]$	101	101
$x[3]$	011	110
$x[7]$	111	111
	ordine "bit-reversal"	ordine BCD

Questa osservazione può essere utilizzata per progettare un algoritmo che riorganizza i campioni all'ingresso ed esegue le operazioni descritte nel grafo.

Nota che una moltiplicazione complessa

$$(a+jb)(c+jd) = \Rightarrow \text{4 moltiplicazioni reali e 2 somme reali.}$$

$$= ac - bd + j(bc+ad)$$

una somma complessa

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

$$\Rightarrow \text{2 somme reali.}$$

in totale quindi:

$\frac{N}{2}\log_2 N$ moltiplicazioni complesse $\rightarrow 4\frac{N}{2}\log_2 N$ molt. reali $+ 2\frac{N}{2}\log_2 N$ somme reali

$N\log_2 N$ somme complesse $\Rightarrow 2N\log_2 N$ somme reali.

$2N\log_2 N$ Molt reali.

$3N\log_2 N$ somme reali

La tabellina seguente riporta i valori di N^2 e $N\log_2 N$ per varie lunghezze della finestra, mostrando il notevole risparmio computazionale.

b	N	N^2	$N\log_2 N$
1	2	4	2
4	16	256	64
8	256	65536	2048
10	1024	1'048'576	10240
12	4096	16'777'216	49152
16	65536	$4.29\,10^9$	1'048'576

ALGORITMO DELLA DECIMAZIONE IN FREQUENZA

Una strategia duale per il dimezzamento successivo divide gli N campioni di partenza nei primi $\frac{N}{2}$ e nei secondi $\frac{N}{2}$ iterando la suddivisione.
Più in particolare

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

$$= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{N}kn} + \sum_{l=0}^{\frac{N}{2}-1} x[l+\tfrac{N}{2}]\, e^{-j\frac{2\pi}{N}k\left(l+\frac{N}{2}\right)}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{N}kn} + \underbrace{e^{-j\frac{2\pi}{N}k\frac{N}{2}}}_{(-1)^k} \sum_{l=0}^{\frac{N}{2}-1} x[l+\tfrac{N}{2}]\, e^{-j\frac{2\pi}{N}kl}$$

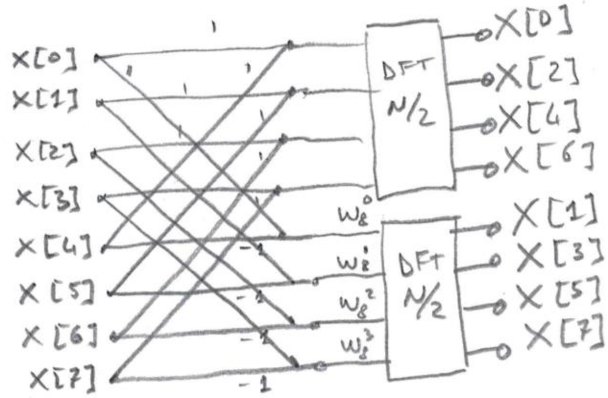
$$= \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + (-1)^k x[n+\tfrac{N}{2}]\right) e^{-j\frac{2\pi}{N}kn}$$

Consideriamo K pari e K dispari separatamente:

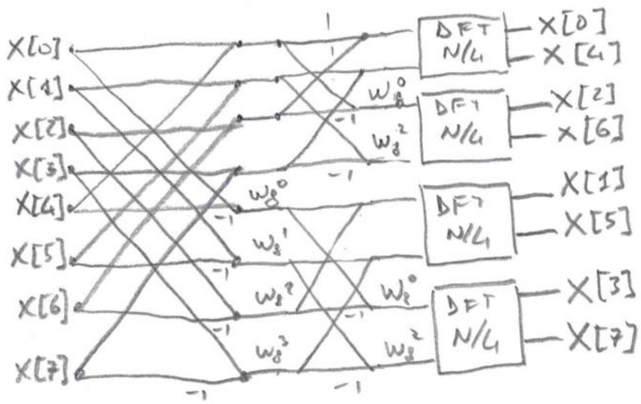
$$\begin{cases} X[2r] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x[n+\frac{N}{2}]\right) e^{-j\frac{2\pi}{N}2rn} & r=0,\dots,\frac{N}{2}-1 \\ X[2r+1] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] - x[n+\frac{N}{2}]\right) e^{-j\frac{2\pi}{N}(2r+1)n} & r=0,\dots,\frac{N}{2}-1 \end{cases}$$

Noto che $\quad W_N^{2rn} = e^{-j\frac{2\pi}{N}2rn} = e^{-j\frac{2\pi}{\frac{N}{2}}rn} = W_{N/2}^{rn}$

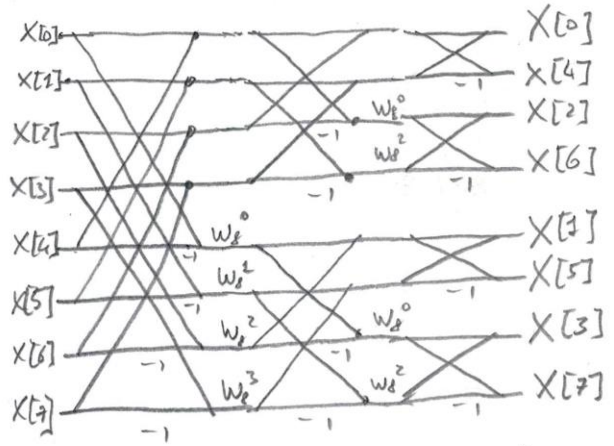
$$\begin{cases} X[2r] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x[n+\frac{N}{2}]\right) W_{N/2}^{rn} & r=0,\dots,\frac{N}{2}-1 \\ X[2r+1] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] - x[n+\frac{N}{2}]\right) W_N^{n}\, W_{\frac{N}{2}}^{rn} & r=0,\dots,\frac{N}{2}-1 \end{cases}$$



Dividendo ulteriormente le DFT (N/2) in DFT su N/4



Il grafo risultante diventa



Valori ordinati in bit-reversal.

Il grafo della decimazione in frequenza è duale a quello della decimazione nel tempo. La complessità computazionale è la stessa. I campioni all'ingresso sono nell'ordine originale mentre quelli all'uscita sono in bit-reversal.

È possibile derivare altri algoritmi FFT basati su altre partizioni dei dati. I due algoritmi presentati sono i più tipici e i più utilizzati.

Si noti anche che la strategia di partizione del problema in sottoproblemi di ordine inferiore corrisponde, nella formulazione matriciale, ad una cascata di prodotti con matrici sparse.

$$\underline{X} = \underline{\underline{W}}\,\underline{x}$$

$$\underline{X} = \underline{\underline{A}}_1\,\underline{\underline{A}}_2 \cdots \underline{\underline{A}}_b\,\underline{x}$$

matrici sparse corrispondenti ai vari stadi dell'algoritmo.

Nell'analisi numerica la struttura a cascata è un paradigma comune per risolvere problemi di fattorizzazione spettrale.