Algoritmi di FFT basati sulla decimazione in frequenza

(*Decimation* in the *Frequency* domain: FFT-DF)

Algoritmi di FFT basati sulla *decimazione in frrquenza* (FFT-DF)

Suddivisione della sequenza $X(k)$ in sottosequenze sempre più piccole:

Ipotesi: $N = 2^\nu \Rightarrow N$ pari

$$X(k) \underset{\substack{\uparrow \\ \text{suddividendo } x(n) \text{ in due parti}}}{=} \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk}$$

$$\underset{\substack{\uparrow \\ \boxed{n' = n - \frac{N}{2} \text{ in } 2^a \sum}}}{=} \underbrace{\sum_{n=0}^{(N/2)-1} x(n) W_N^{nk}}_{\substack{\neq \text{ da DFT su } N/2 \text{ punti} \\ \text{perché c'è } W_N^{nk}! \\ (\text{e non } W_{N/2}^{nk})}} + \underbrace{\sum_{n'=0}^{(N/2)-1} x\left(n' + \frac{N}{2}\right) W_N^{n'k} \cdot W_N^{(N/2)k}}_{\substack{\neq \text{ da DFT su } N/2 \text{ punti} \\ \text{perché c'è } W_N^{n'k} \text{ e non } W_{N/2}^{n'k}!}}$$

Algoritmi di FFT-DF

$$\boxed{W_N^{(N/2)k} = (-1)^k}$$

$$X(k) \underset{\uparrow}{\overset{\downarrow}{=}} \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{nk}$$

accorpando le $\sum$

$n = n'$ in 2^a $\sum$

$\Downarrow$ separando k pari da k dispari

$$X(2r) \underset{\uparrow}{=} \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{2rn}$$

$$\boxed{(-1)^k = 1}$$

$$X(2r+1) \underset{\uparrow}{=} \sum_{n=0}^{(N/2)-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^{n} \cdot W_N^{2rn}$$

$$\boxed{(-1)^k = -1}$$

$$r = 0, 1, \ldots, (N/2 - 1)$$

Algoritmi di FFT-DF

Così espresse, le $X(2r)$ e $X(2r+1)$ sono due DFT su $N/2$ punti, essendo:

- $g(n) = x(n) + x(n+N/2)$ somma di 1^a e 2^a metà di $x(n)$
- $h(n) = [x(n) - x(n+N/2)]$ differenza tra 1^a e 2^a metà di $x(n)$
- $h(n)$ è moltiplicata per W_N^n: $h(n)W_N^n$ e $X(2r+1)$ è sua DFT
- $g(n)$ e $h(n)$: lunghe $N/2$
- $\sum$ corrispondono a DFT su $N/2$ punti dal momento che:

$$W_N^{2rn} = W_{N/2}^{rn}$$

Algoritmi di FFT-DF

$$X(2r) \rightarrow \underbrace{\left(\frac{N}{2}\right)^2}_{\text{FFT}} \circledast/\oplus \text{ complesse } + \frac{N}{2} \text{ somme complesse per costruire } g(n)$$

$$X(2r+1) \rightarrow \underbrace{\left(\frac{N}{2}\right)^2}_{\text{FFT}} \circledast/\oplus \text{ complesse } + \frac{N}{2} \text{ somme complesse per costruire } h(n)$$

$$+ \frac{N}{2} \text{ moltiplicazioni complesse per } W_N^n$$

$\Rightarrow$ In totale:

$$\oplus \quad \frac{N}{2} + \frac{N}{2} + 2\left(\frac{N}{2}\right)^2 = N + 2\left(\frac{N}{2}\right)^2$$

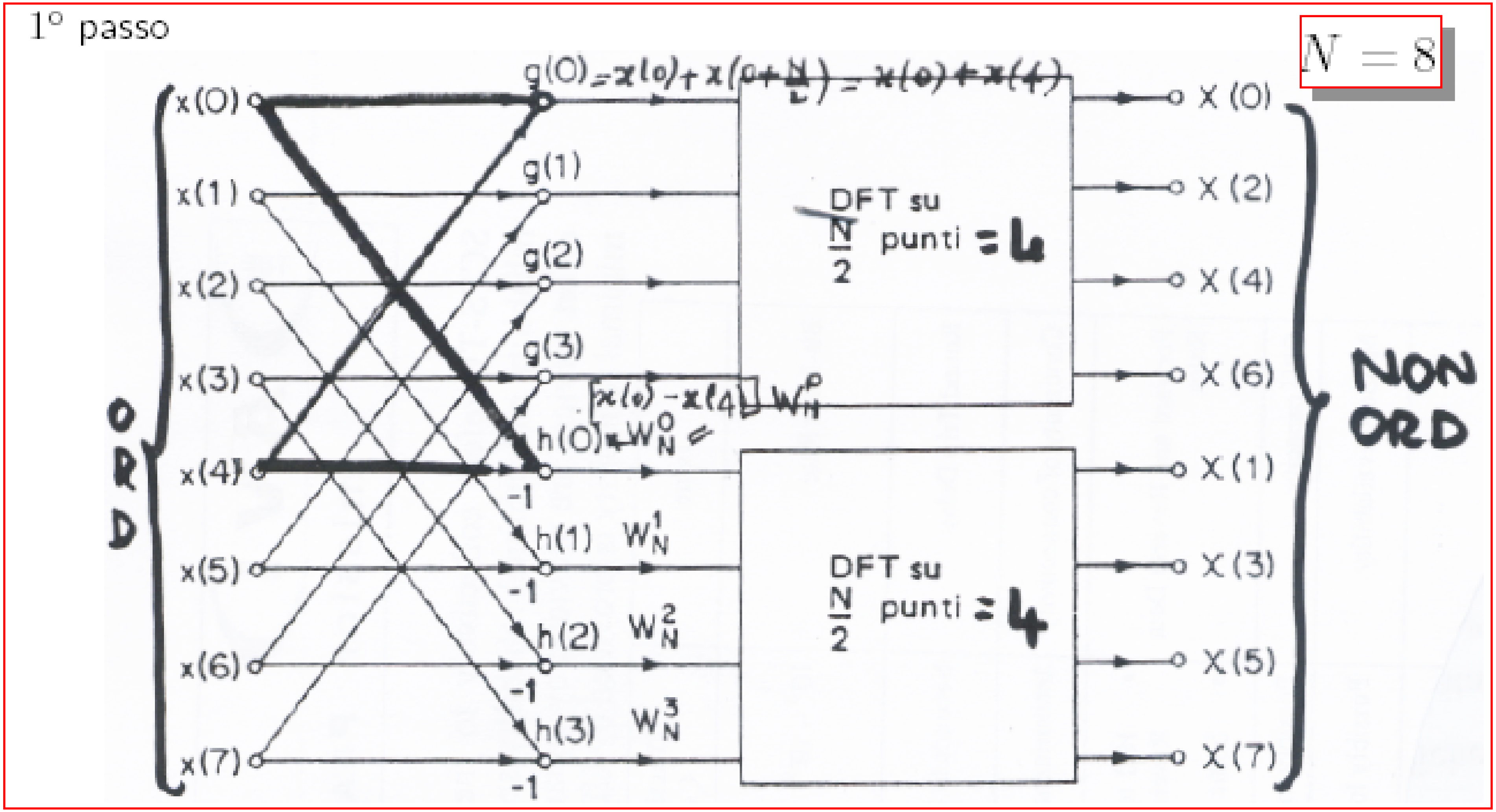
$$\circledast \; \frac{N}{2} + 2\left(\frac{N}{2}\right)^2 = \frac{N}{2} + 2\left(\frac{N}{2}\right)^2$$

operazioni complesse

$\Rightarrow$ iterando la decimazione:

$$\oplus \rightarrow \underbrace{N + N + \cdots + N}_{\nu \text{ volte}} \quad \circledast \rightarrow \underbrace{\frac{N}{2} + \frac{N}{2} + \cdots + \frac{N}{2}}_{\nu \text{ volte}}$$

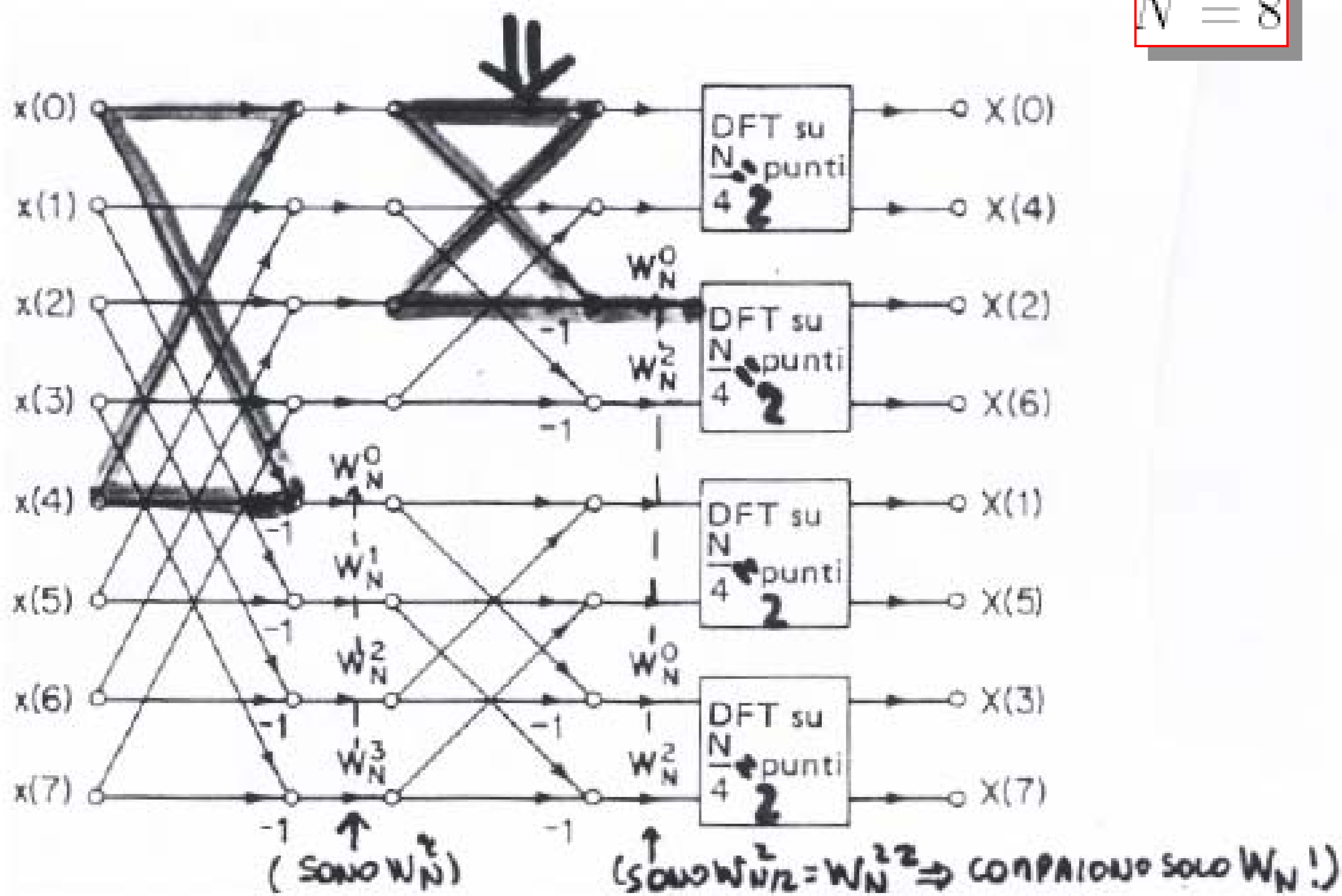
Algoritmi di FFT-DF



Algoritmi di FFT-DF

2° passo

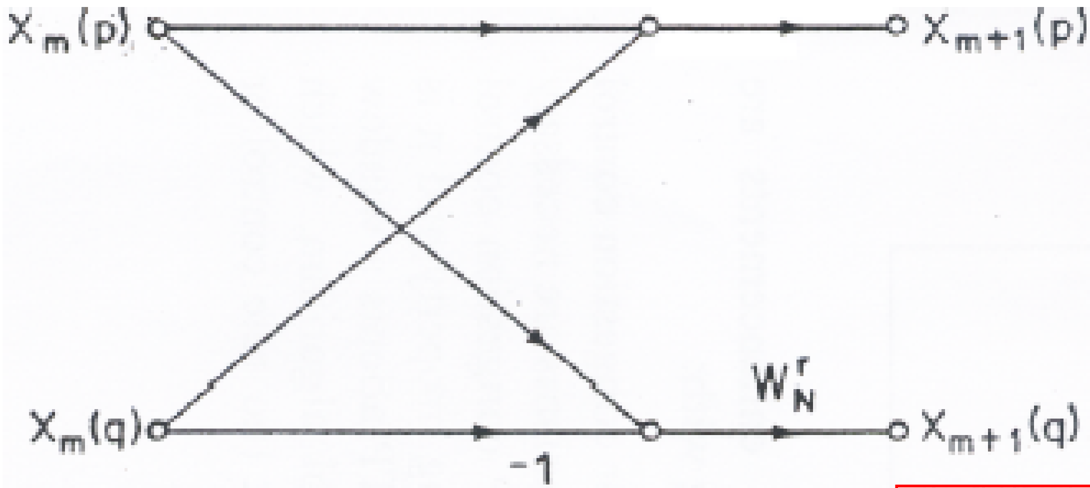
$N = 8$



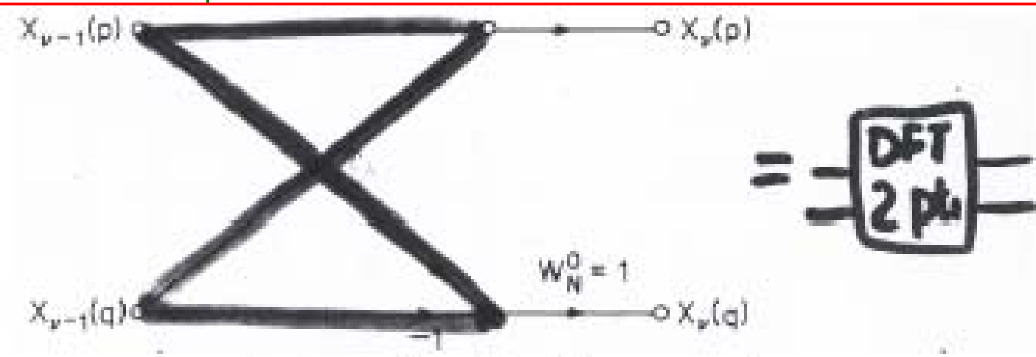
Algoritmi di FFT-DF

3° passo

$$\left\{ \begin{array}{l} X_{m+1}(p) = X_m(p) + X_m(q) \\ X_{m+1}(q) = [X_m(p) - X_m(q)]W_N^r \end{array} \right\}$$ calcoli diversi da DT



butterfly del generico stadio



butterfly di ultimo stadio

Algoritmi di FFT-DF

$N = 8$