❶❸ Algoritmi fundamentali de tip FFT

Algoritmul lui Goertzel — Prima variantă de calcul a TFD

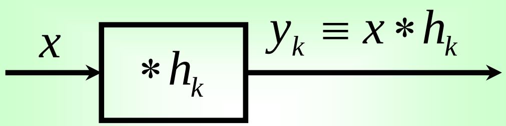
- **Exprimare echivalentă a TFD**

$$X[k] = \sum_{n=0}^{N-1} x[n]\overline{w}_N^{k(N-n)}, \quad \forall k \in \overline{0, N-1}$$

$$\left(w_N^{kN} = 1 \quad \overline{\forall k \in 0, N-1} \right)$$

periodicitate   sumă de convoluţie

Ieşirea la momentul N a unui sistem liniar



$$h_k[p] \overset{def}{=} \overline{w}_N^{kp} u_0[p], \ \forall p \in \mathbb{Z}$$

$$\left(Supp\ x = \overline{0, N-1} \right)$$

$$y_k[p] = \sum_{n \geq 0} x[n] h_k[p-n] = \sum_{n=0}^{N-1} x[n]\overline{w}_N^{k(p-n)}, \quad \forall p \in \overline{0, N-1}$$

$$X[k] = y_k[N], \quad \forall k \in \overline{0, N-1}$$

- **Funcţia de transfer a sistemului**

$$H_k(z) \overset{def}{=} \sum_{p \geq 0} h_k[p] z^{-p} = \sum_{p \geq 0} \left(\overline{w}_N^k z^{-1} \right)^p = \frac{1}{1 - \overline{w}_N^k z^{-1}}, \quad \forall k \in \overline{0, N-1}$$

$$|z| > 1$$

Ecuaţia recusivă a ieşirii

$$y_k[n] - \overline{w}_N^k y_k[n-1] = x[n], \quad \forall n \in \mathbb{Z}$$

Teorema întîrzierii

$$z^{-1}\mathscr{Z}(f)(z) = \mathscr{Z}(q^{-1} f)(z)$$

Ecuaţia recusivă a TFD

$$\begin{cases} y_k[0] = x[0] \\ y_k[1] = \overline{w}_N^k y_k[0] + x[1] \\ \vdots \\ y_k[N] = \overline{w}_N^k y_k[N-1] + x[N] = \overline{w}_N^k y_k[N-1] = X[k] \end{cases}$$

$$\forall k \in \overline{0, N-1}$$

Algoritmul lui Goertzel

A doua variantă de calcul a TFD (îmbunătăţită)

- Exprimare echivalentă ecuaţiei recursive anterioare

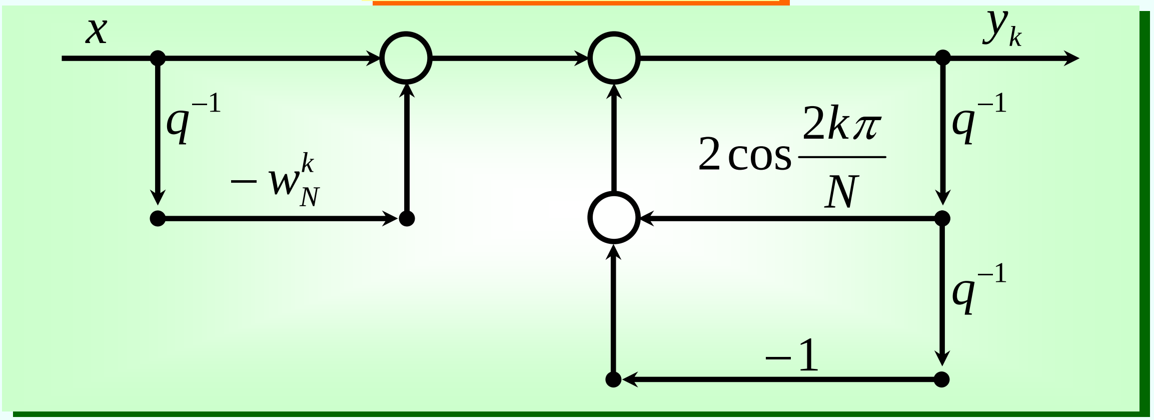
$$\left(1-\overline{w}_N^k q^{-1}\right) y_k[n] = x[n], \quad \forall n \in \overline{0,N}, \ \forall k \in \overline{0,N-1}$$

înmulţire forţată cu $\left(1-w_N^k q^{-1}\right)$

$$\left(1-2\cos\frac{2k\pi}{N}q^{-1}+q^{-2}\right) y_k[n] = \left(1-w_N^k q^{-1}\right) x[n], \quad \forall n \in \overline{0,N}, \ \forall k \in \overline{0,N-1}$$

$$y_k[n] = 2y_k[n-1]\cos\frac{2k\pi}{N} - y_k[n-2] + x[n] - w_N^k x[n-1], \quad \forall n \in \overline{0,N}, \ \forall k \in \overline{0,N-1}$$

Schema de calcul



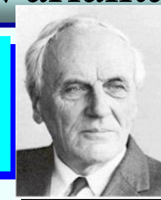
Iniţializare

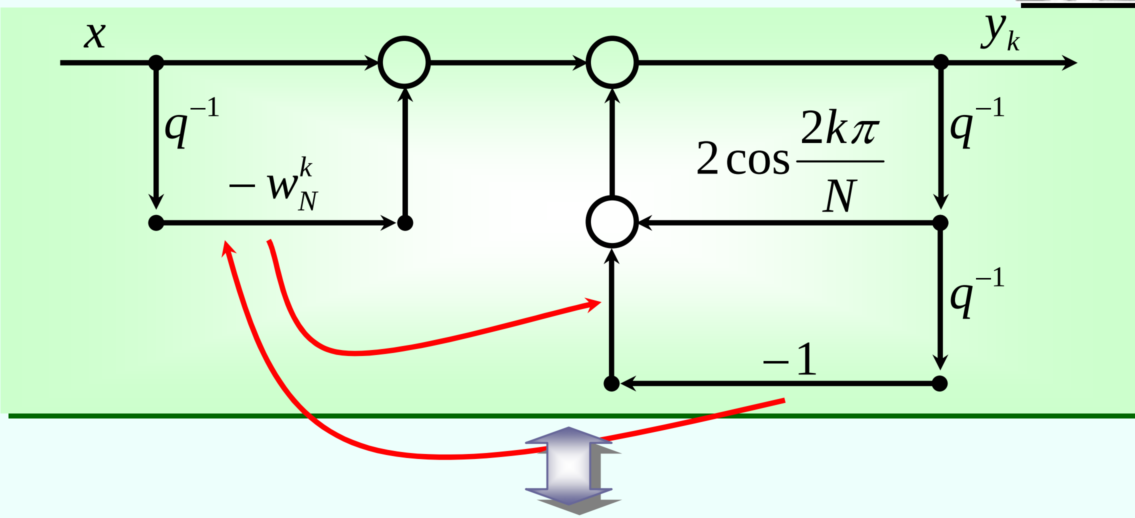
$$y_k[-1] = y_k[-2] = 0$$
$$\forall k \in \overline{0,N-1}$$

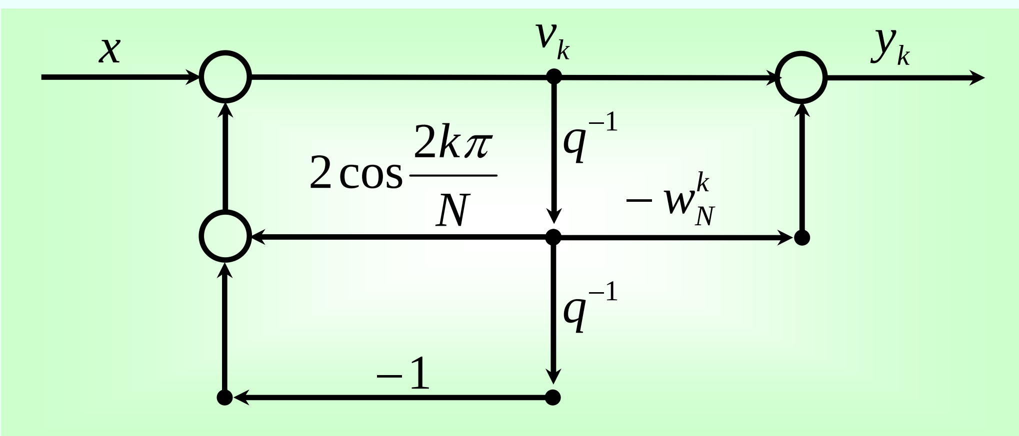
Algoritmul lui Goertzel

Varianta eficientă de calcul a TFD

Schema anterioară de calcul se poate transforma echivalent, folosind *Teorema lui TELLEGEN*.







Algoritmul lui Goertzel

$$\begin{cases} v_k[-2] = v_k[-1] = 0 \\ \vdots \\ v_k[n] = 2\cos\frac{2k\pi}{N} v_k[n-1] - v_k[n-2] + x[n] \\ \vdots \\ v_k[N] = 2\cos\frac{2k\pi}{N} v_k[N-1] - v_k[N-2] \end{cases}$$

$$X[k] = y_k[N] = v_k[N] - w_N^k v_k[N-1]$$

$$\forall k \in \overline{0, N-1}$$

Număr de operații

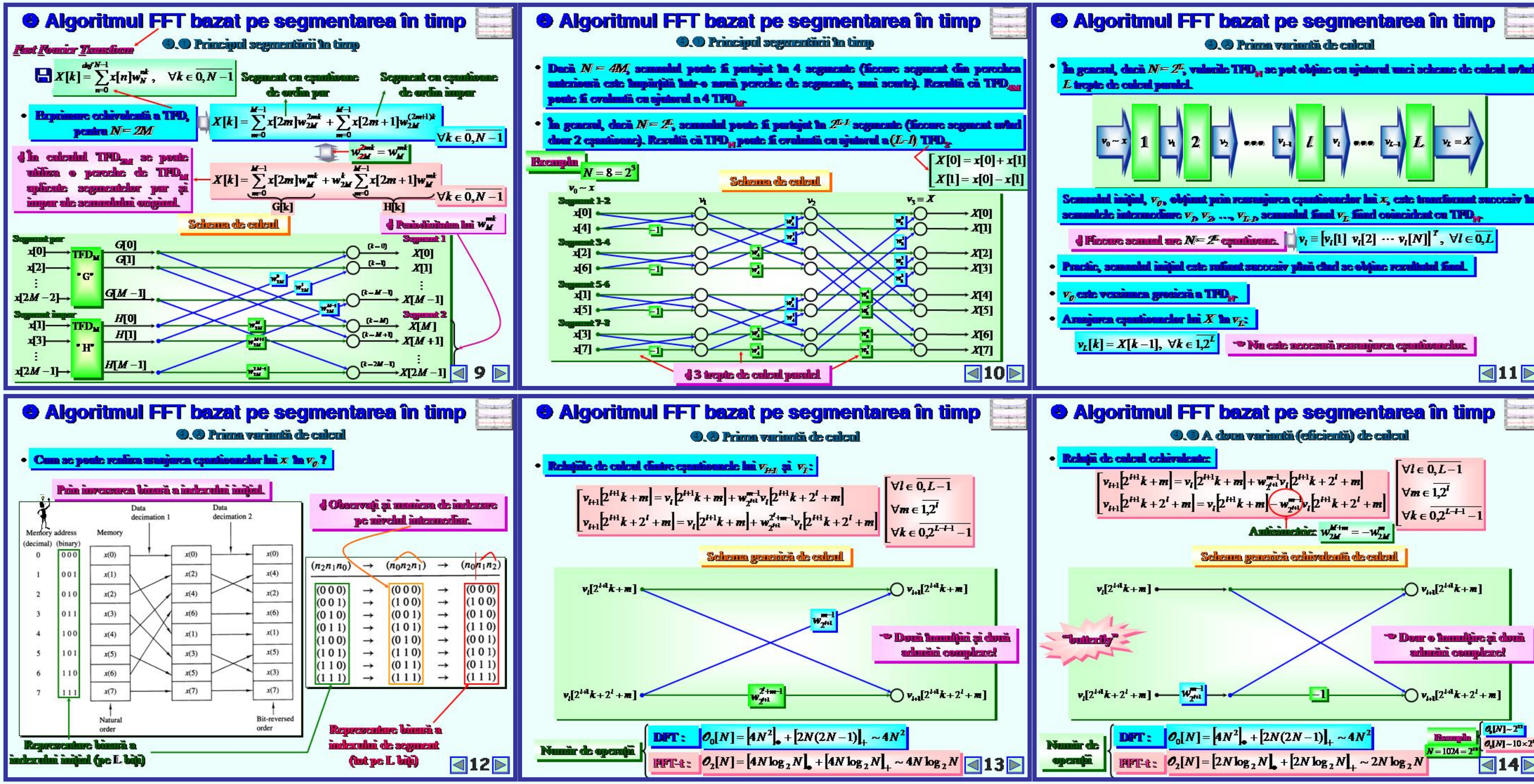
$$\mathcal{O}_1[N] = \left[\left\lceil\frac{N-1}{2}\right\rceil(2N+5)\right]_\bullet + [4N(N+1)]_+ \sim N^2$$

☝ de 4 ori mai mic

❶❸ Algoritmi fundamentali de tip FFT

Algoritmul fundamental bazat pe segmentarea în timp (FFT-t)

• Descris tot în cadrul lucrărilor de laborator ale pachetului **FFT**.



Număr de operații

$$\mathcal{O}_{FFT-t}[N] = \left[2N\log_2 N\right]_{\bullet} + \left[2N\log_2 N\right]_{+} \sim 2N\log_2 N$$

❶❸ Algoritmi fundamentali de tip FFT

Fast Fourier Transform

Algoritmul FFT-t — Principul segmentării în timp

$$X[k] \overset{def}{=} \sum_{n=0}^{N-1} x[n] w_N^{nk}, \quad \forall k \in \overline{0, N-1}$$

- **Exprimare echivalentă a TFD, pentru $N = 2M$**

Segment cu eşantioane de ordin par — Segment cu eşantioane de ordin impar

$$X[k] = \sum_{m=0}^{M-1} x[2m] w_{2M}^{2mk} + \sum_{m=0}^{M-1} x[2m+1] w_{2M}^{(2m+1)k}, \quad \forall k \in \overline{0, N-1}$$

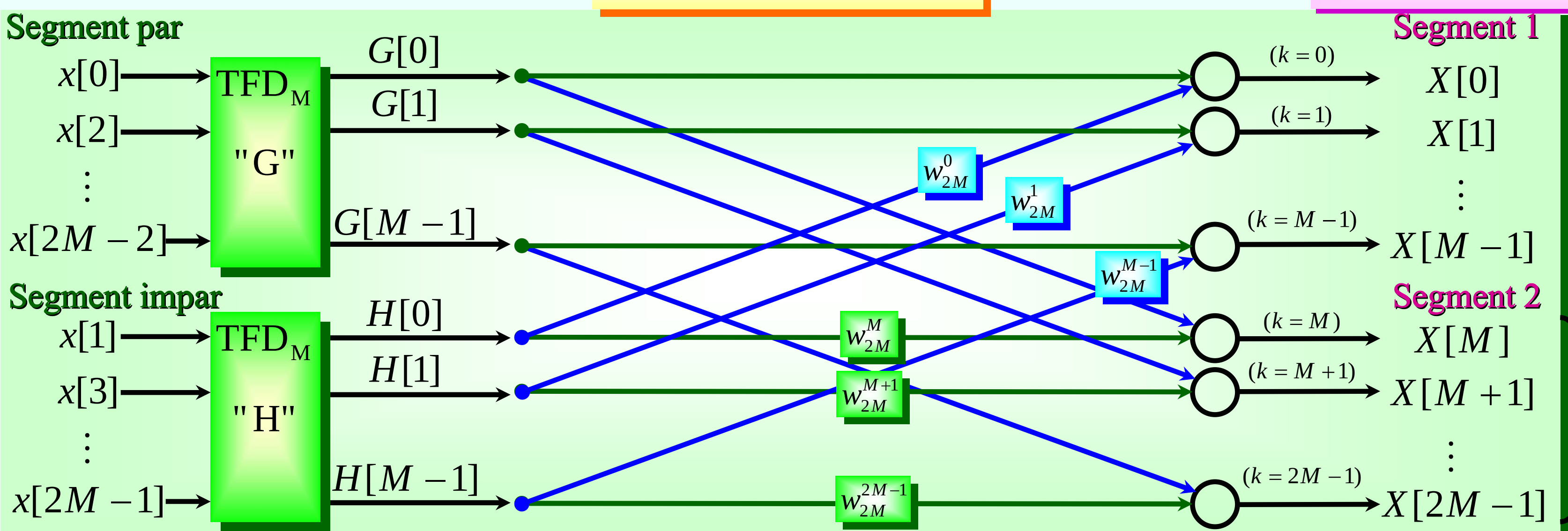
$$\cancel{w_{2M}^{2mk}} = w_M^{mk}$$

$$X[k] = \underbrace{\sum_{m=0}^{M-1} x[2m] w_M^{mk}}_{G[k]} + w_{2M}^{k} \underbrace{\sum_{m=0}^{M-1} x[2m+1] w_M^{mk}}_{H[k]}, \quad \forall k \in \overline{0, N-1}$$

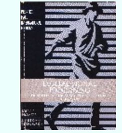
☝ În calculul TFD_{2M} se poate utiliza o pereche de TFD_M aplicate segmentelor par şi impar ale semnalului original.

Schema de calcul

☝ Periodicitatea lui w_M^{mk}



➊➌ Algoritmi fundamentali de tip FFT



Algoritmul FFT-t | Principul segmentării în timp

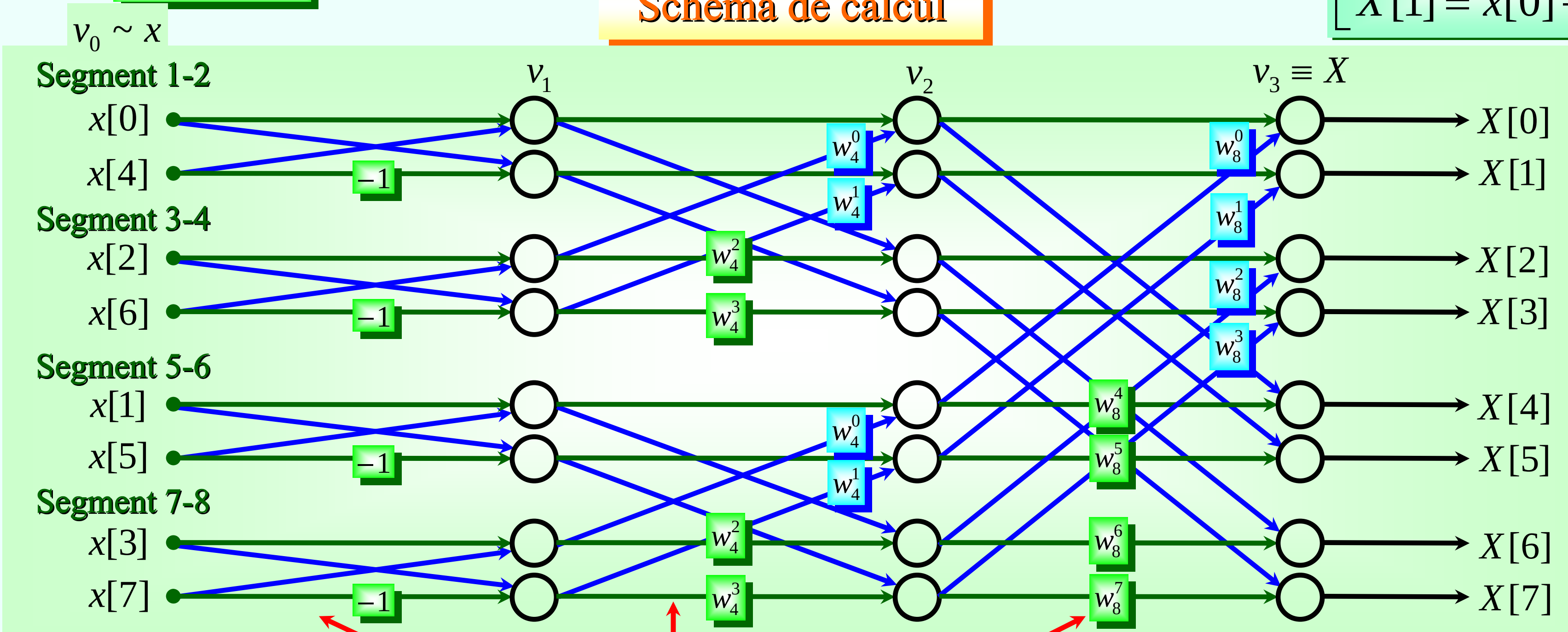
- Dacă $N = 4M$, semnalul poate fi partajat în 4 segmente (fiecare segment din perechea anterioară este împărțită într-o nouă pereche de segmente, mai scurte). Rezultă că TFD_{4M} poate fi evaluată cu ajutorul a 4 TFD_{M}.

- În general, dacă $N = 2^{L}$, semnalul poate fi partajat în 2^{L-1} segmente (fiecare segment avînd doar 2 eşantioane). Rezultă că TFD_{N} poate fi evaluată cu ajutorul a $(L-1)$ TFD_{2}.

$$\begin{cases} X[0] = x[0] + x[1] \\ X[1] = x[0] - x[1] \end{cases}$$

Exemplu $N = 8 = 2^{3}$

Schema de calcul

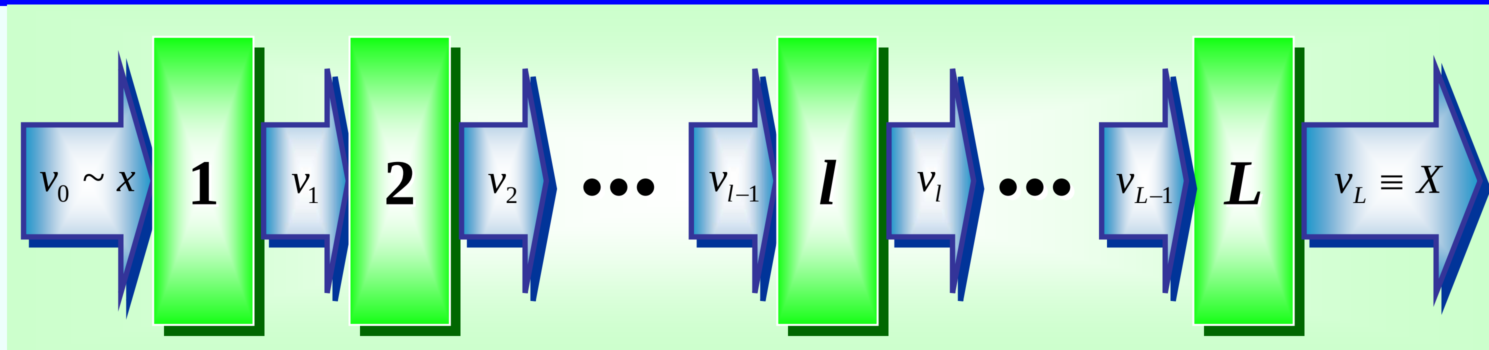


☝ 3 trepte de calcul paralel

❶❸ Algoritmi fundamentali de tip FFT

Algoritmul FFT-t | Prima variantă de calcul

- În general, dacă $N = 2^L$, valorile TFD_N se pot obţine cu ajutorul unei scheme de calcul avînd L trepte de calcul paralel.



Semnalul iniţial, v_0, obţinut prin rearanjarea eşantioanelor lui x, este transformat succesiv în semnalele intermediare v_1, v_2, …, v_{L-1}, semnalul final v_L fiind coincident cu TFD_N.

☝ Fiecare semnal are $N = 2^L$ eşantioane. ⇨ $v_l \equiv \left[v_l[1] \;\; v_l[2] \;\; \cdots \;\; v_l[N]\right]^T, \;\; \forall l \in \overline{0,L}$

- Practic, semnalul iniţial este rafinat succesiv pînă cînd se obţine rezultatul final.
- v_0 este versiunea grosieră a TFD_N.
- Aranjarea eşantioanelor lui X în v_L:

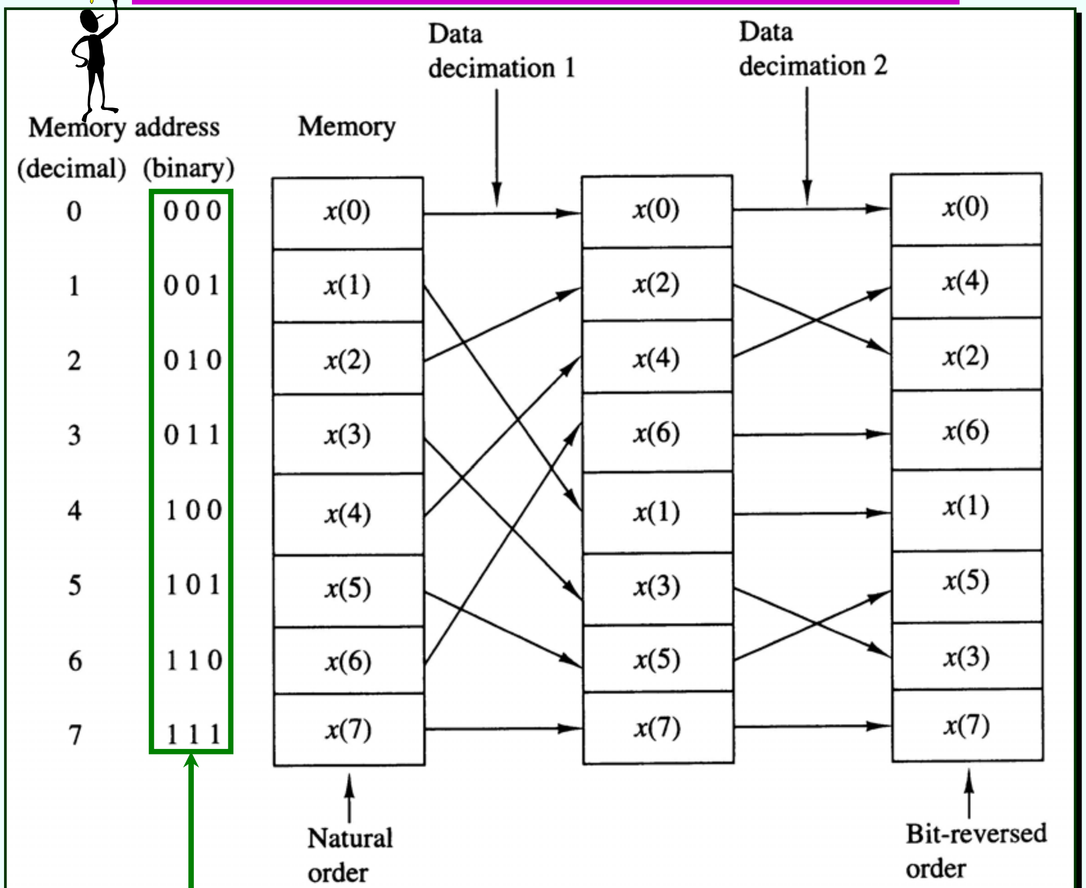
$v_L[k] = X[k-1], \;\; \forall k \in \overline{1,2^L}$ ☜ Nu este necesară rearanjarea eşantioanelor.

❶❸ Algoritmi fundamentali de tip FFT

Algoritmul FFT-t | Prima variantă de calcul

Cum se poate realiza aranjarea eşantioanelor lui x în v_0 ?

Prin inversarea binară a indexului iniţial.

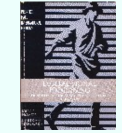


☝ Observaţi şi maniera de indexare pe nivelul intermediar.

$(n_2 n_1 n_0)$	→	$(n_0 n_2 n_1)$	→	$(n_0 n_1 n_2)$
(0 0 0)	→	(0 0 0)	→	(0 0 0)
(0 0 1)	→	(1 0 0)	→	(1 0 0)
(0 1 0)	→	(0 0 1)	→	(0 1 0)
(0 1 1)	→	(1 0 1)	→	(1 1 0)
(1 0 0)	→	(0 1 0)	→	(0 0 1)
(1 0 1)	→	(1 1 0)	→	(1 0 1)
(1 1 0)	→	(0 1 1)	→	(0 1 1)
(1 1 1)	→	(1 1 1)	→	(1 1 1)

Reprezentare binară a indexului iniţial (pe L biţi)

Reprezentare binară a indexului de segment (tot pe L biţi)



❶❸ Algoritmi fundamentali de tip FFT

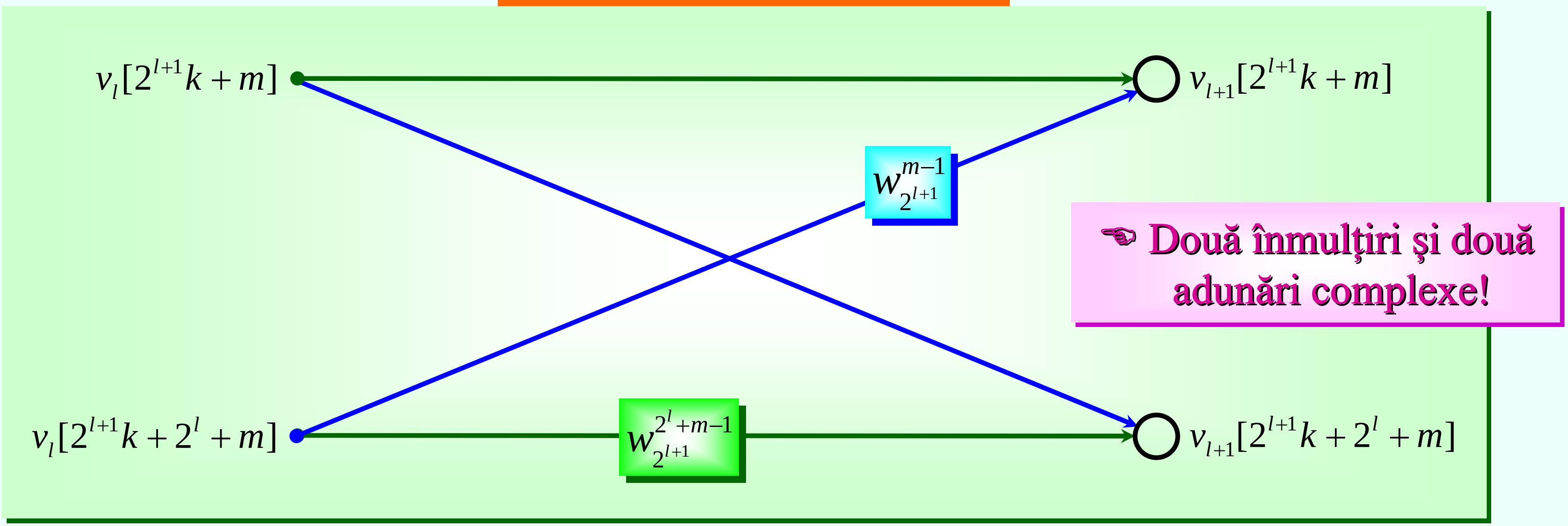
Algoritmul FFT-t | Prima variantă de calcul

- **Relaţiile de calcul dintre eşantioanele lui v_{l+1} şi v_l :**

$$\begin{cases} v_{l+1}\left[2^{l+1}k+m\right]=v_l\left[2^{l+1}k+m\right]+w_{2^{l+1}}^{m-1}v_l\left[2^{l+1}k+2^l+m\right] \\ v_{l+1}\left[2^{l+1}k+2^l+m\right]=v_l\left[2^{l+1}k+m\right]+w_{2^{l+1}}^{2^l+m-1}v_l\left[2^{l+1}k+2^l+m\right] \end{cases}$$

$$\begin{cases} \forall l\in\overline{0,L-1} \\ \forall m\in\overline{1,2^l} \\ \forall k\in\overline{0,2^{L-l-1}-1} \end{cases}$$

Schema generică de calcul



Număr de operaţii

DFT : $\mathcal{O}_0[N]=\left[4N^2\right]_\bullet+\left[2N(2N-1)\right]_+\sim 4N^2$

FFT-t : $\mathcal{O}_2[N]=\left[4N\log_2 N\right]_\bullet+\left[4N\log_2 N\right]_+\sim 4N\log_2 N$

❶❸ Algoritmi fundamentali de tip FFT

Algoritmul FFT-t | A doua variantă (eficientă) de calcul

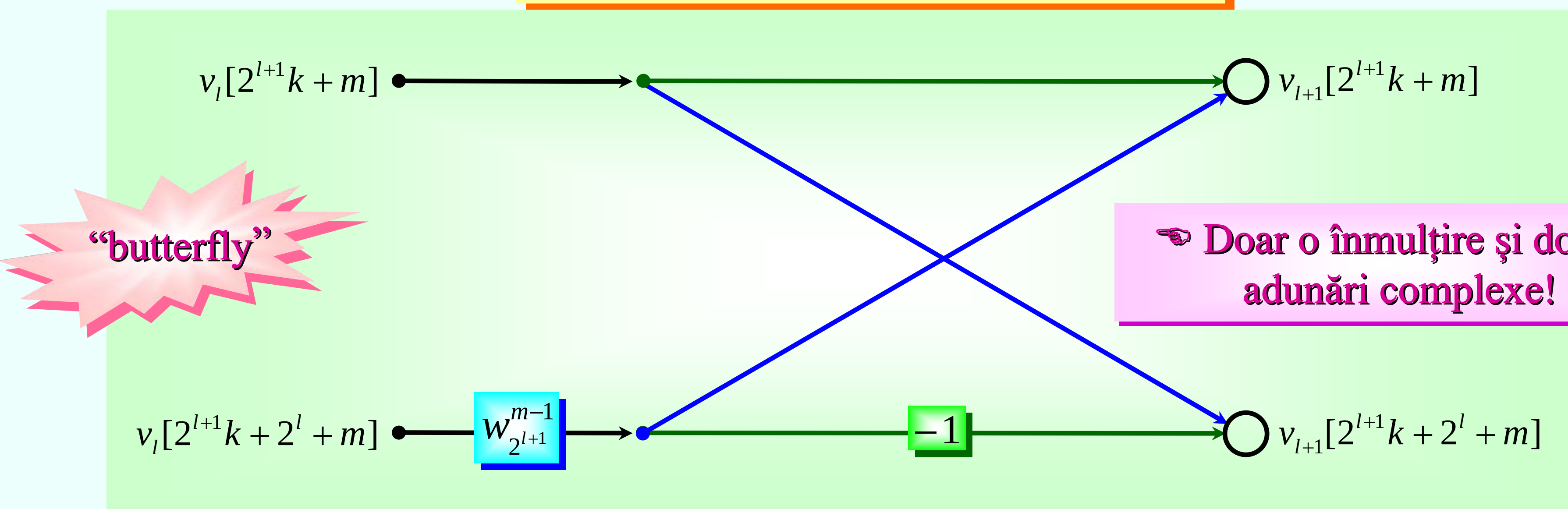
- **Relaţii de calcul echivalente:**

$$\begin{cases} v_{l+1}\left[2^{l+1}k+m\right]=v_l\left[2^{l+1}k+m\right]+w_{2^{l+1}}^{m-1}v_l\left[2^{l+1}k+2^l+m\right] \\ v_{l+1}\left[2^{l+1}k+2^l+m\right]=v_l\left[2^{l+1}k+m\right]-w_{2^{l+1}}^{m-1}v_l\left[2^{l+1}k+2^l+m\right] \end{cases}$$

$$\begin{cases} \forall l\in\overline{0,L-1} \\ \forall m\in\overline{1,2^l} \\ \forall k\in\overline{0,2^{L-l-1}-1} \end{cases}$$

Antisimetrie: $w_{2M}^{M+m}=-w_{2M}^{m}$

Schema generică echivalentă de calcul



☜ Doar o înmulţire şi două adunări complexe!

Număr de operaţii

$$\begin{cases} \text{DFT}: & \mathcal{O}_0[N]=\left[4N^2\right]_\bullet+\left[2N(2N-1)\right]_+\sim 4N^2 \\ \text{FFT-t}: & \mathcal{O}_2[N]=\left[2N\log_2 N\right]_\bullet+\left[2N\log_2 N\right]_+\sim 2N\log_2 N \end{cases}$$

Exemplu $N=1024=2^{10}$

$$\begin{cases} \mathcal{O}_0[N]\sim 2^{22} \\ \mathcal{O}_2[N]\sim 10\times 2^{11} \end{cases}$$