

① ③ Algoritmi fundamentali de tip FFT



Algoritmul FFT-f

Principul segmentării în frecvență


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

Segment cu eșantioane
de ordin $0-(M-1)$

Segment cu eșantioane
de ordin $M-(2M-1)$

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

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

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

👉 Calculul TFD_{2M} se poate efectua folosind 2 semnale cu suporturi de lungime M .

$$w_{2M}^{Mk} = w_2^k = (-1)^k$$

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

$$\begin{aligned} g[m] &\stackrel{\text{def}}{=} x[m] + x[M+m], \text{ pentru } k = \text{par} \\ h[m] &\stackrel{\text{def}}{=} x[m] - x[M+m], \text{ pentru } k = \text{impar} \end{aligned} \quad \forall m \in \overline{0, M-1}$$

$$\begin{aligned} X[2k] &= \sum_{m=0}^{M-1} g[m] w_M^{mk} \\ X[2k+1] &= \sum_{m=0}^{M-1} w_{2M}^m h[m] w_M^{mk} \end{aligned} \quad \forall k \in \overline{0, M-1}$$

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

👉 În calculul TFD_{2M} se poate utiliza o pereche de TFD_M aplicate semnalelor g și $w_{2M}^\bullet h$.

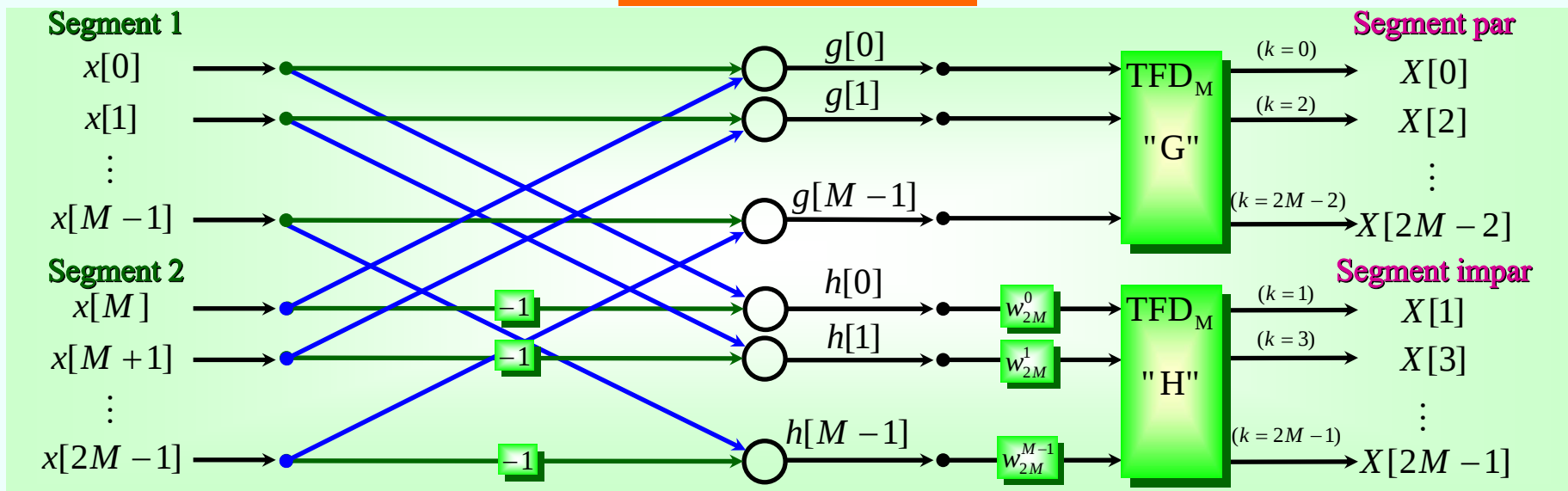
①③ Algoritmi fundamentali de tip FFT



Algoritmul FFT-f

Principul segmentării în frecvență

Schema de calcul



- 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 eșantioane). Rezultă că TFD_N poate fi evaluată cu ajutorul a $(L-1)$ TFD_2 .

① ③ Algoritmi fundamentali de tip FFT

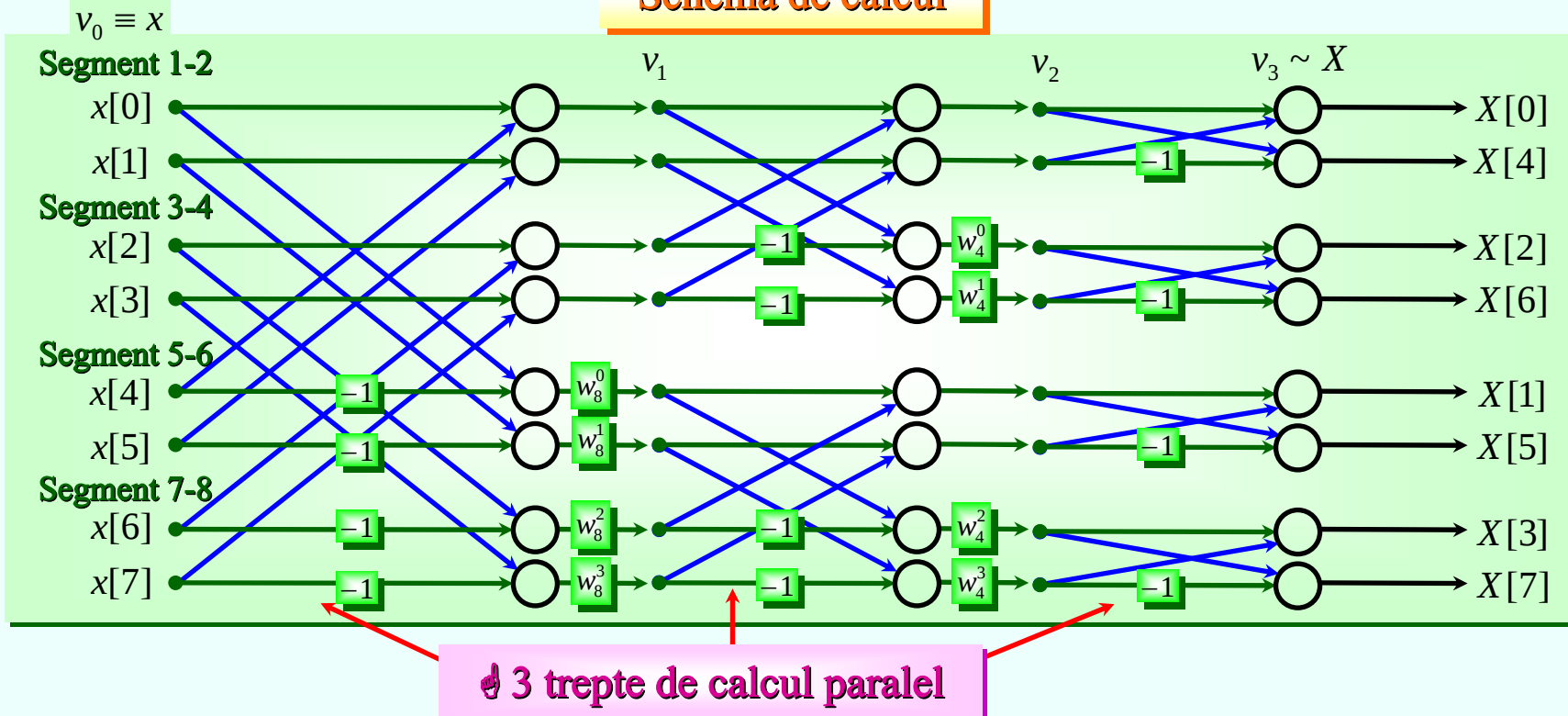
Algoritmul FFT-f

Principul segmentării în frecvență

Exemplu

$$N = 8 = 2^3$$

Schema de calcul



- Această schemă poate fi obținută din cea aferentă Algoritmului FFT cu segmentarea semnalului în timp și celule elementare de calcul de tip “fluture” (*butterfly*), aplicînd 2 operații: **intrările se inversează cu ieșirile; toate arcele își schimbă sensurile.**

Aranjarea semnalelor în vectorii v_0 și v_3 este similară algoritmului precedent.

①③ Algoritmi fundamentali de tip FFT

Algoritmul FFT-t

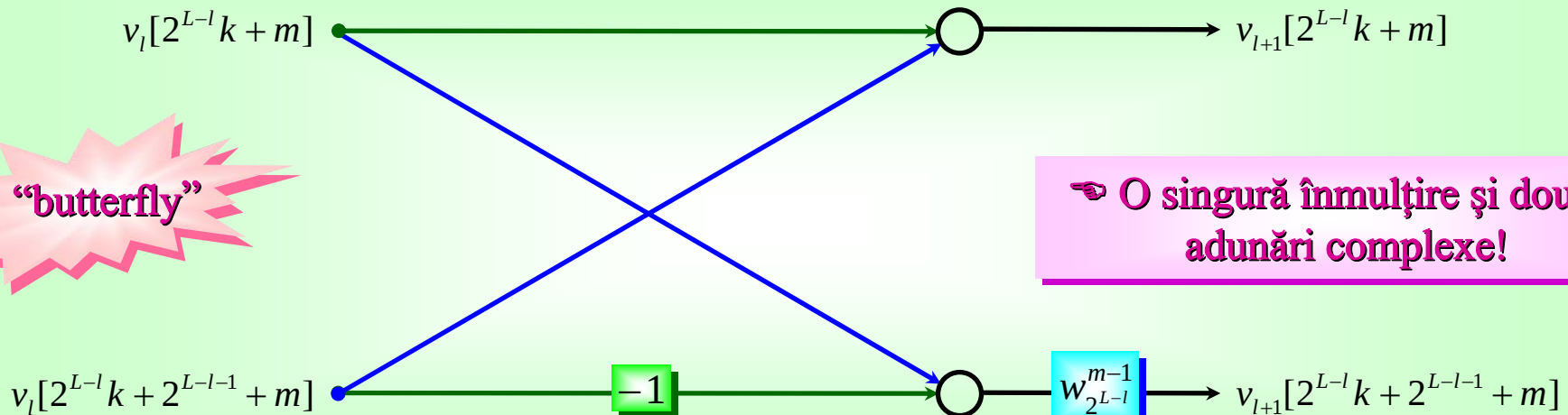
Varianta eficientă de calcul

- Relațiile de calcul între semnalele intermediare succesive:

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

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

Schema generică de calcul



Număr de
operații

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

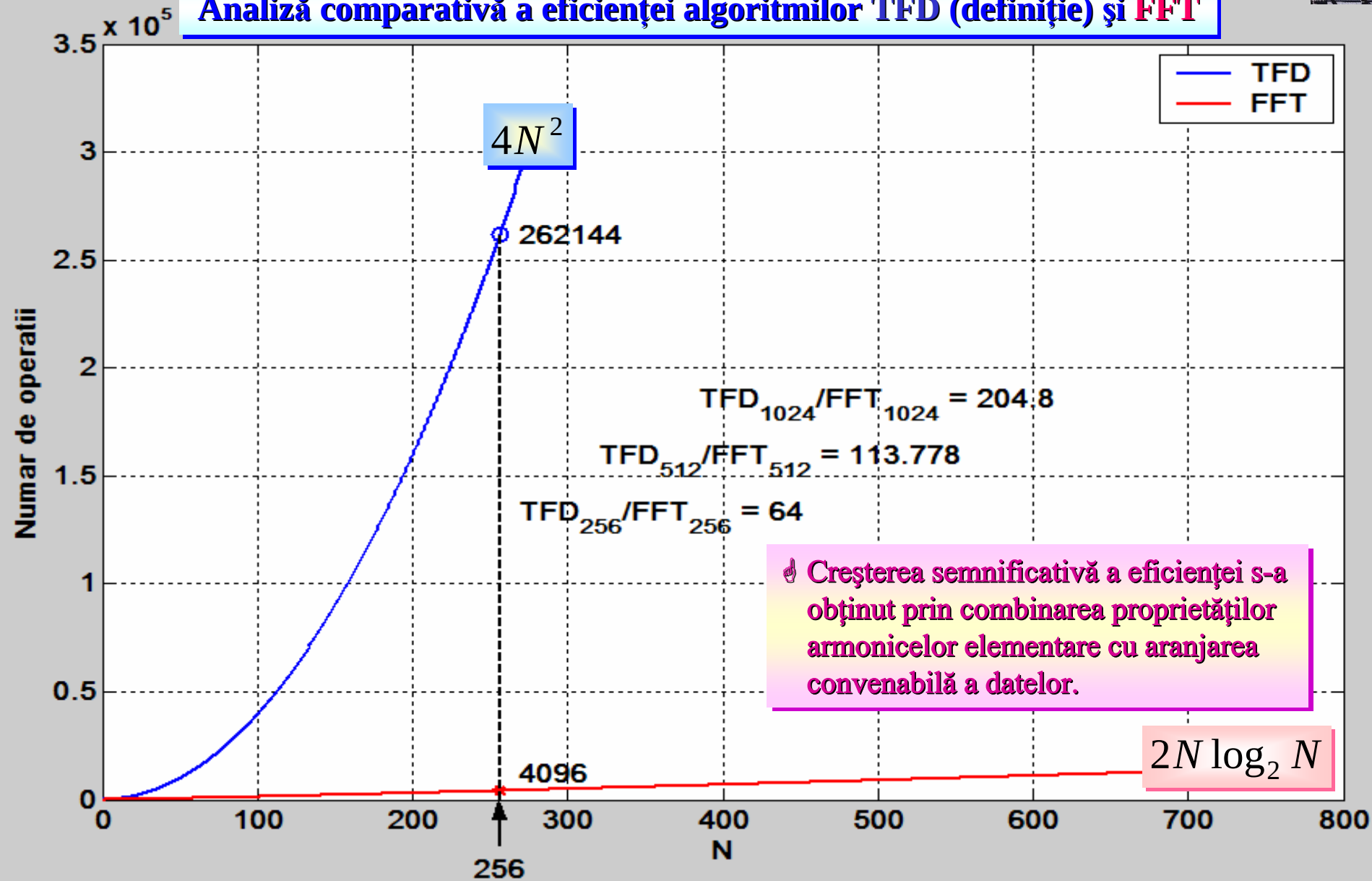
FFT-f : $\mathcal{O}_2[N] = [2N \log_2 N]_{\bullet} + [2N \log_2 N]_{+} \sim 2N \log_2 N$

➡ Același.

⚡ Se recomandă totuși implementarea bazată pe o funcție auto-recursivă.

① ③ Algoritmi fundamentali de tip FFT

Analiză comparativă a eficienței algoritmilor TFD (definiție) și FFT



①③ Algoritmi fundamentali de tip FFT

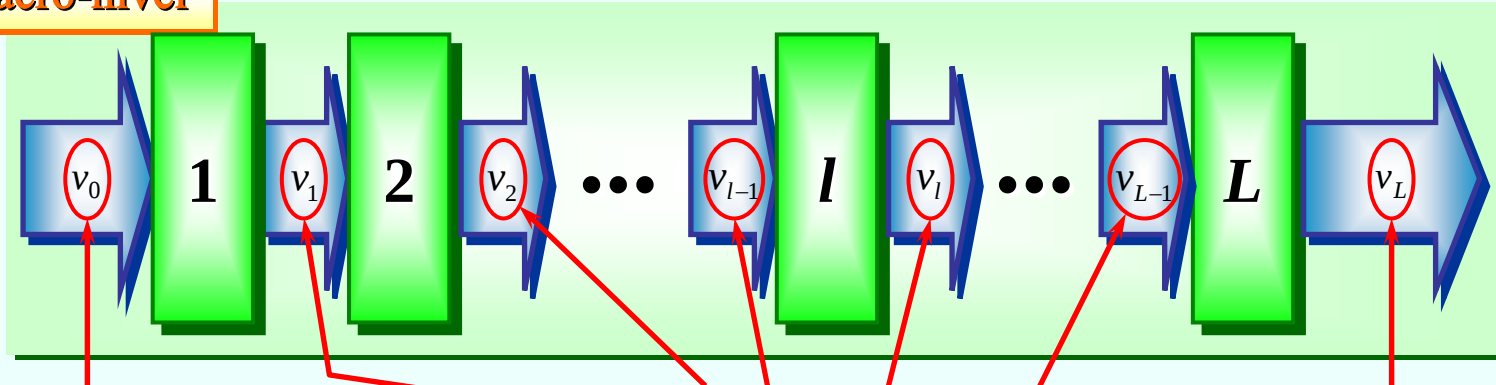
Caracteristici principale ale algoritmilor fundamentali de tip FFT

→ Durata semnalului discret trebuie să fie o putere a lui 2.

$$N = 2^L$$

→ Organizarea calculului pe două nivele ierarhice: $\left\{ \begin{array}{l} \text{➤ Macro-nivel: } L \text{ trepte secvențiale de calcul.} \\ \text{➤ Micro-nivel: } N/2 \text{ blocuri paralele de calcul.} \end{array} \right.$

Macro-nivel



semnal de intrare

(identic cu semnalul original sau obținut din acesta prin rearanjarea eșantioanelor)

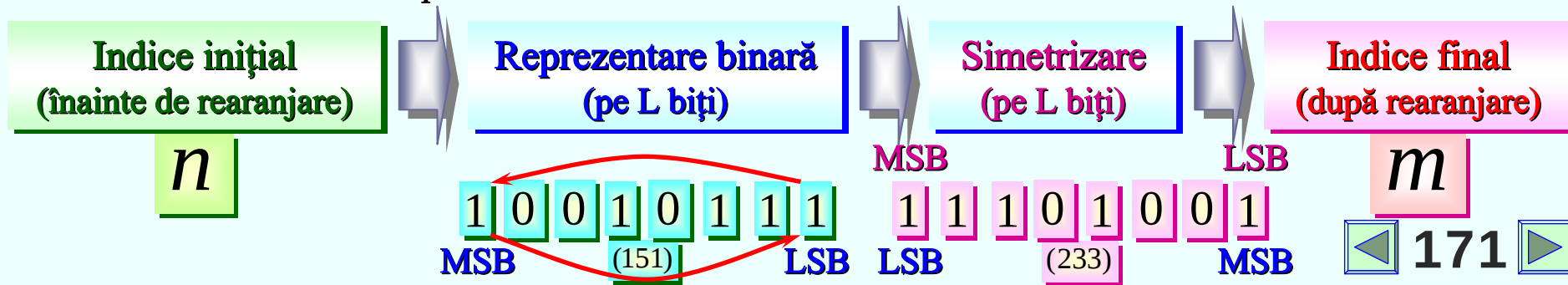
semnale intermediare

(versiuni din ce în ce mai rafinate ale semnalului de intrare)

semnal de ieșire

(identic cu TFD sau obținut din aceasta prin rearanjarea eșantioanelor)

- Rearanjarea eșantioanelor este o **operație esențială**.
- Ea se realizează prin **Tehnica simetrizării indicilor binari**.



① ③ Algoritmi fundamentali de tip FFT

Caracteristici principale ale algoritmilor fundamentali de tip FFT (continuare)

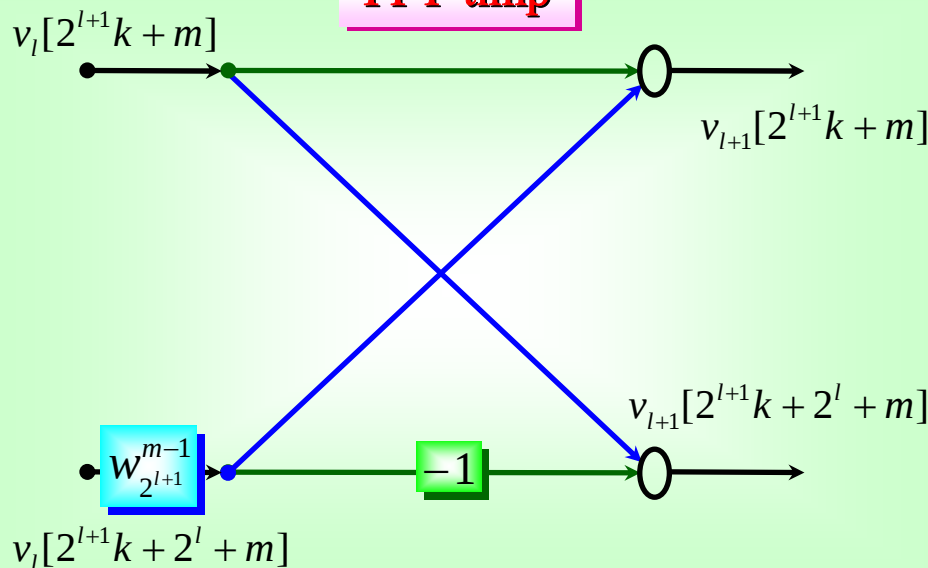
Micro-nivel

Blocuri paralele de calcul

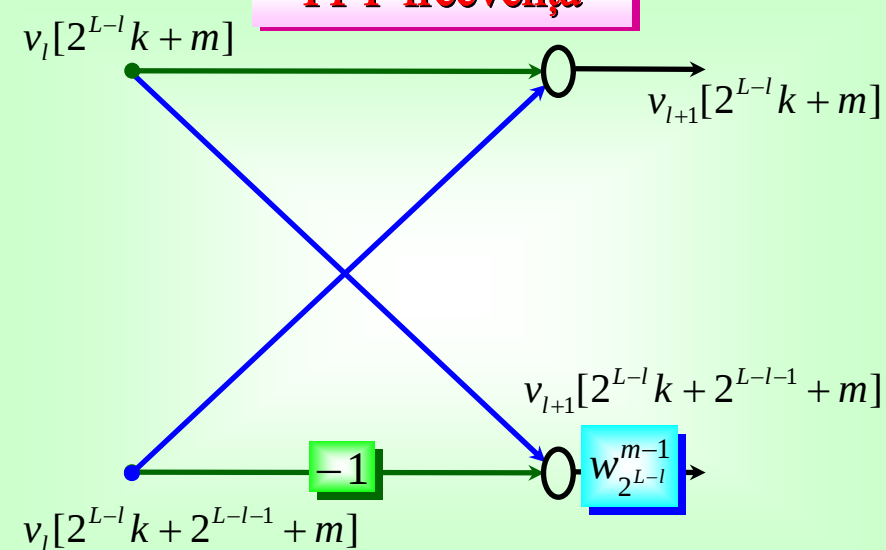
“butterfly”

← schema cea mai eficientă

FFT-timp



FFT-frecvență



- Nu este necesară rearanjarea eșantioanelor semnalelor intermediare, ci doar a eșantioanelor semnalului de intrare (FFT-timp) sau de ieșire (FFT-frecvență).

➔ Geometria blocurilor paralele de calcul variază de-a lungul treptelor secvențiale de calcul, dar nu este deformată.

- FFT-timp: “aripile” se deschid odată cu rafinarea semnalelor.
- FFT-frecvență: “aripile” se închid odată cu rafinarea semnalelor.

☞ Pentru a evita fie rearanjarea eșantioanelor fie geometria variabilă a blocurilor paralele de calcul, au fost proiectate alte variante ale algoritmilor de tip FFT.

①③ Algoritmi fundamentali de tip FFT

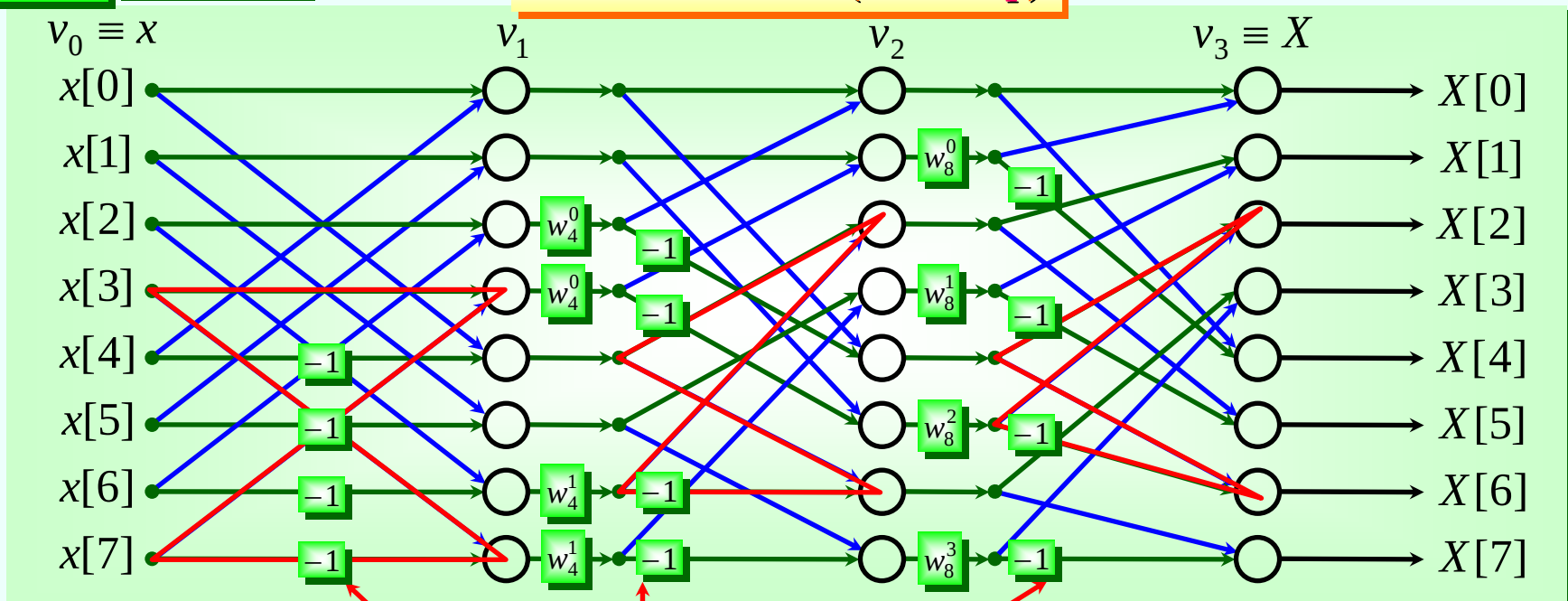
Evitarea reindexării semnalelor de intrare și ieșire

Algoritmul fundamental cu structură internă variabilă

- Semnalul de intrare v_0 coincide cu semnalul original, iar semnalul de ieșire v_L – cu TFD a sa, X .
- Geometria blocurilor paralele de calcul este **deformată și variabilă**.

Exemplu $N = 8 = 2^3$

Schema de calcul (FFT-timp)



3 trepte de calcul paralel

Se poate observa geometria deformată și variabilă de-a lungul treptelor de calcul.

- Acesta este prețul plătit pentru evitarea rearanjării semnalelor de intrare și ieșire.

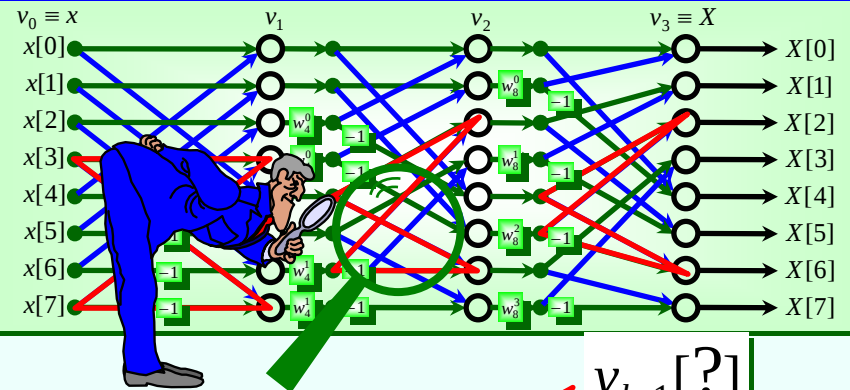
① ③ Algoritmi fundamentali de tip FFT

Evitarea reindexării semnalelor de intrare și ieșire (continuare)

Cum se poate implementa structura variabilă de-a lungul treptelor de calcul?

Trebuie analizată cu atenție **indexarea** intrărilor și ieșirilor blocului paralel generic de calcul (de tip “butterfly”).

Algoritmul fundamental cu structură internă variabilă



Ieșirile blocului de calcul sunt întotdeauna de forma

$$\begin{cases} v_{l+1}[m] \\ v_{l+1}[2^{L-1} + m] \end{cases}$$

$$m \in \overline{0, 2^{L-1} - 1}$$

Semnalul de ieșire este segmentat în **două componente succesive.**

$$\begin{aligned} &v_{l+1}[0] \\ &v_{l+1}[1] \\ &\vdots \\ &v_{l+1}[m] \\ &\vdots \\ &v_{l+1}[2^{L-1} - 1] \\ &v_{l+1}[2^{L-1}] \\ &\vdots \\ &v_{l+1}[2^{L-1} + m] \\ &\vdots \\ &v_{l+1}[2^L - 1] \end{aligned}$$

Indexarea ieșirilor este **independentă** de indicele treptei de calcul curente.

$$v_l[?]$$

$$v_l[?]$$

Așadar

$$v_l[?]$$

$$v_l[?]$$

$$v_{l+1}[?]$$

$$v_{l+1}[?]$$

$$v_{l+1}[m]$$

$$v_{l+1}[2^{L-1} + m]$$

$$l \in \overline{0, L-1}$$

$$\begin{aligned} &m \in \overline{0, 2^{L-1} - 1} \\ &l \in \overline{0, L-1} \end{aligned}$$

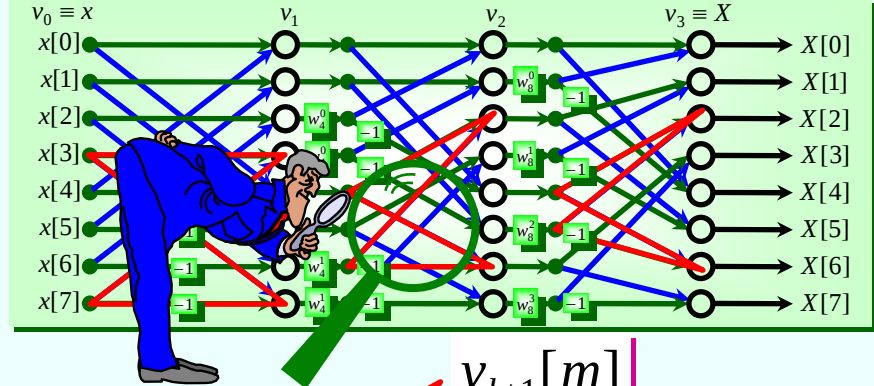
①③ Algoritmi fundamentali de tip FFT

Evitarea reindexării semnalelor de intrare și ieșire (continuare)



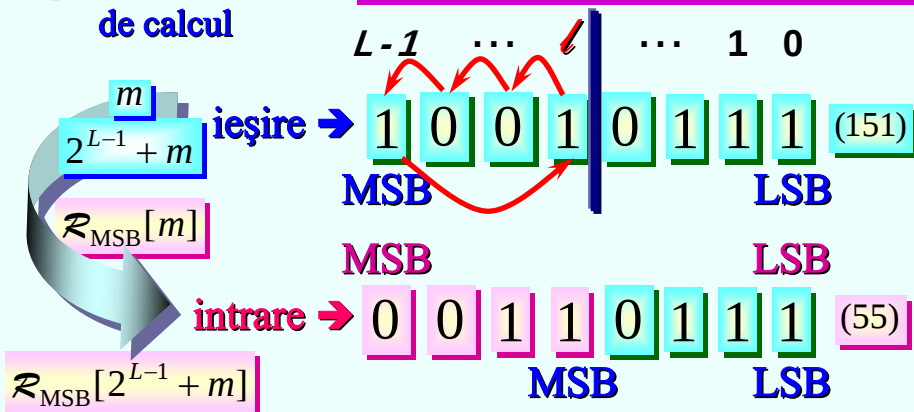
Algoritmul fundamental cu structură internă variabilă

Intrările blocului de calcul sunt indexate prin **Tehnica relocării bitului celui mai semnificativ (MSB)** în reprezentările binare ale indicilor celor 2 ieșiri.



$l = 0$
prima treaptă de calcul
Indicii intrărilor coincid cu indicii ieșirilor.

$l \in \overline{1, L-2}$
treptele intermediare de calcul
MSB este relocat în poziția l (unde LSB are poziția 0).



$l = L - 1$
ultima treaptă de calcul
MSB este relocat în poziția LSB, adică în poziția 0.

