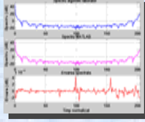
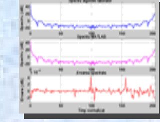


Sumar

- ✓ **Bibliografie**
- ✓ ❶ Notaţii şi convenţii
- ✓ ❷ Obiectivul lucrărilor de laborator
- ✓ ❸ Algoritmul lui Goertzel
- ☞ ❹ Algoritmul FFT bazat pe segmentarea semnalului în timp
- ❺ Algoritmul FFT bazat pe segmentarea semnalului în frecvenţă
- ❻ Date de intrare, prezentarea rezultatelor şi punctaje

❹ Algoritmul FFT bazat pe segmentarea în timp



Fast Fourier Transform

❹.❶ 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 $N = 2M$**

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

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

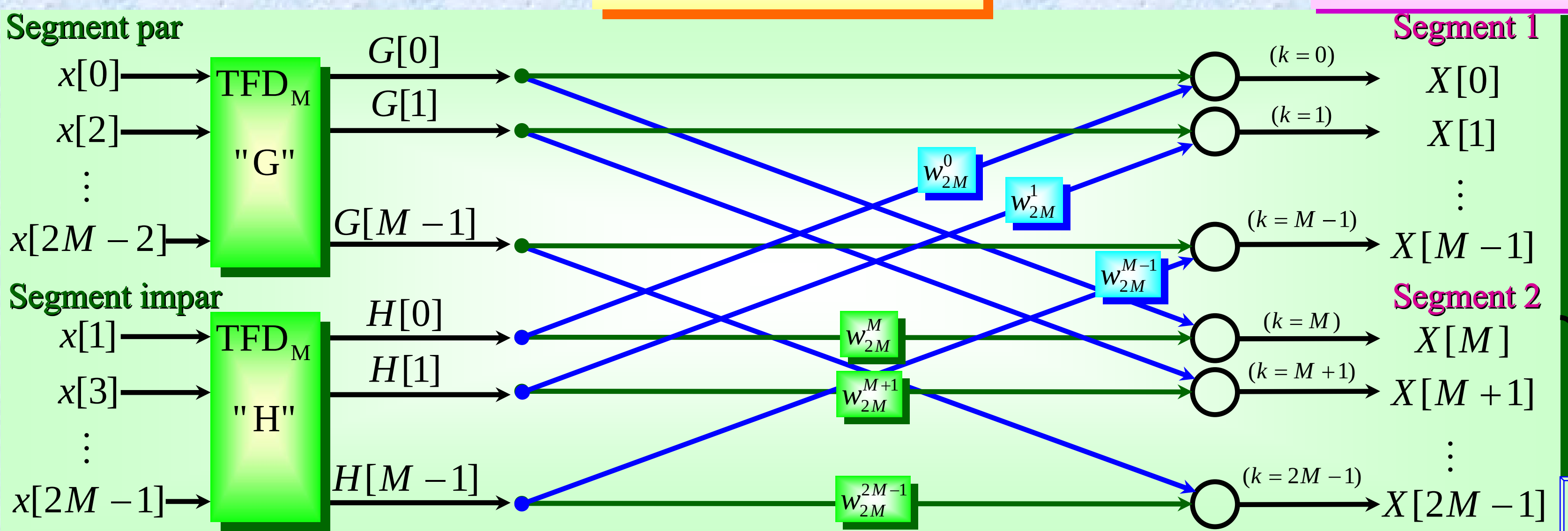
$$w_{2M}^{2mk} = w_M^{mk}$$

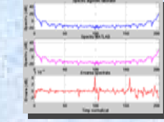
$$X[k] = \underbrace{\sum_{m=0}^{M-1} x[2m] w_M^{mk}}_{G[k]} + w_{2M}^{k} \underbrace{\sum_{m=0}^{M-1} x[2m+1] w_M^{mk}}_{H[k]} \quad \forall k \in \overline{0, N-1}$$

☝ 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}





❹ Algoritmul FFT bazat pe segmentarea în timp

❹.❶ Principiul segmentării în timp

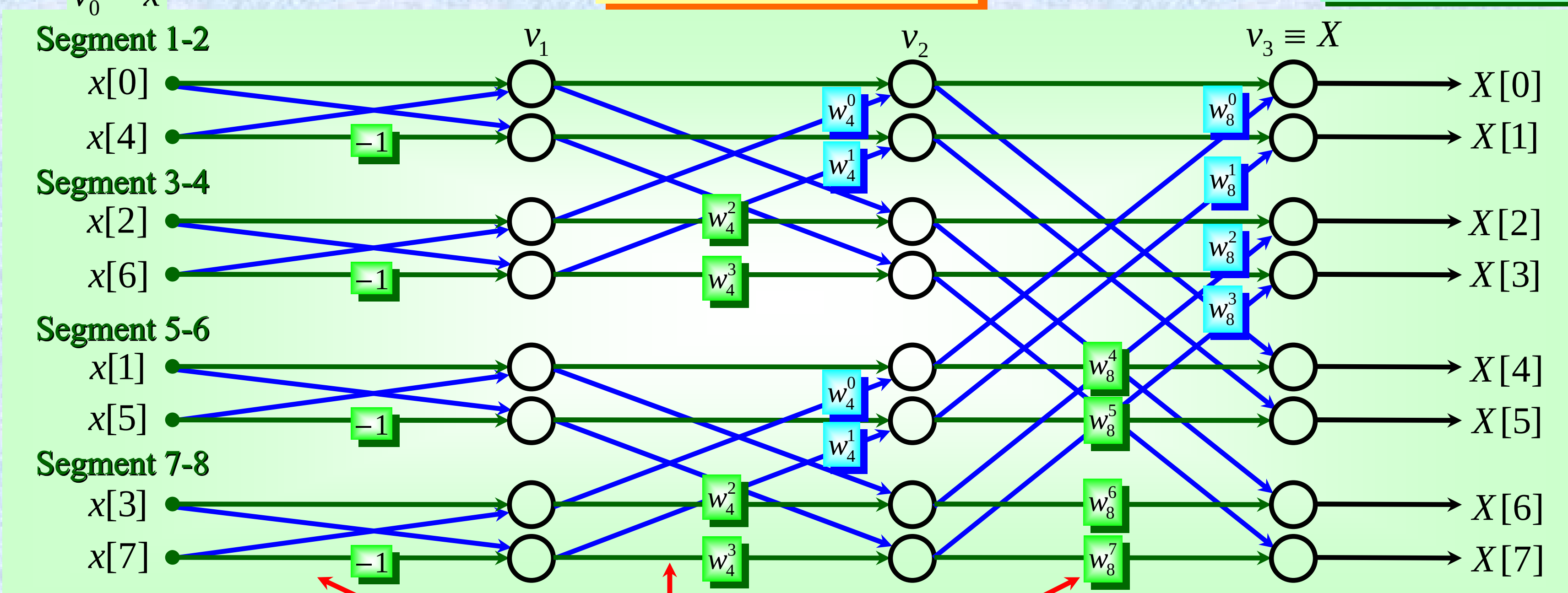
- Dacă $N = 4M$, semnalul poate fi partajat în 4 segmente (fiecare segment din perechea anterioară este împărțită într-o nouă pereche de segmente, mai scurte). Rezultă că TFD_{4M} poate fi evaluată cu ajutorul a 4 TFD_{M}.
- În general, dacă $N = 2^L$, semnalul poate fi partajat în 2^{L-1} segmente (fiecare segment având doar 2 eșantioane). Rezultă că TFD_N poate fi evaluată cu ajutorul a $(L-1)$ TFD_2.

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

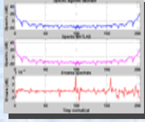
Exemplu $N = 8 = 2^3$

Schema de calcul

$v_0 \sim x$



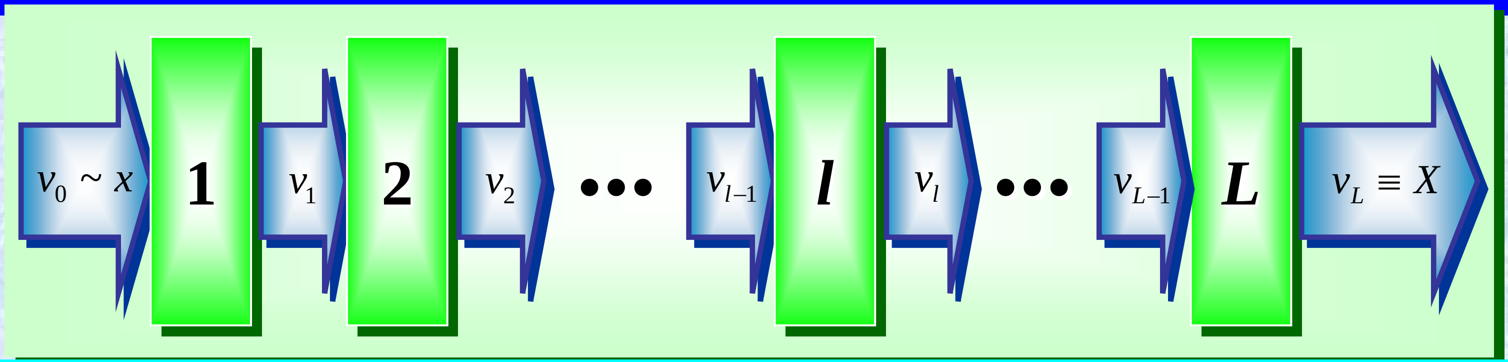
☝ 3 trepte de calcul paralel



❹ Algoritmul FFT bazat pe segmentarea în timp

❹.❷ 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$_N$.

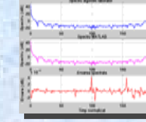
☝ Fiecare semnal are $N = 2^L$ eşantioane.

$$v_l \equiv \left[v_l[1] \;\; v_l[2] \;\; \cdots \;\; v_l[N] \right]^T, \;\; \forall l \in \overline{0,L}$$

- 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.

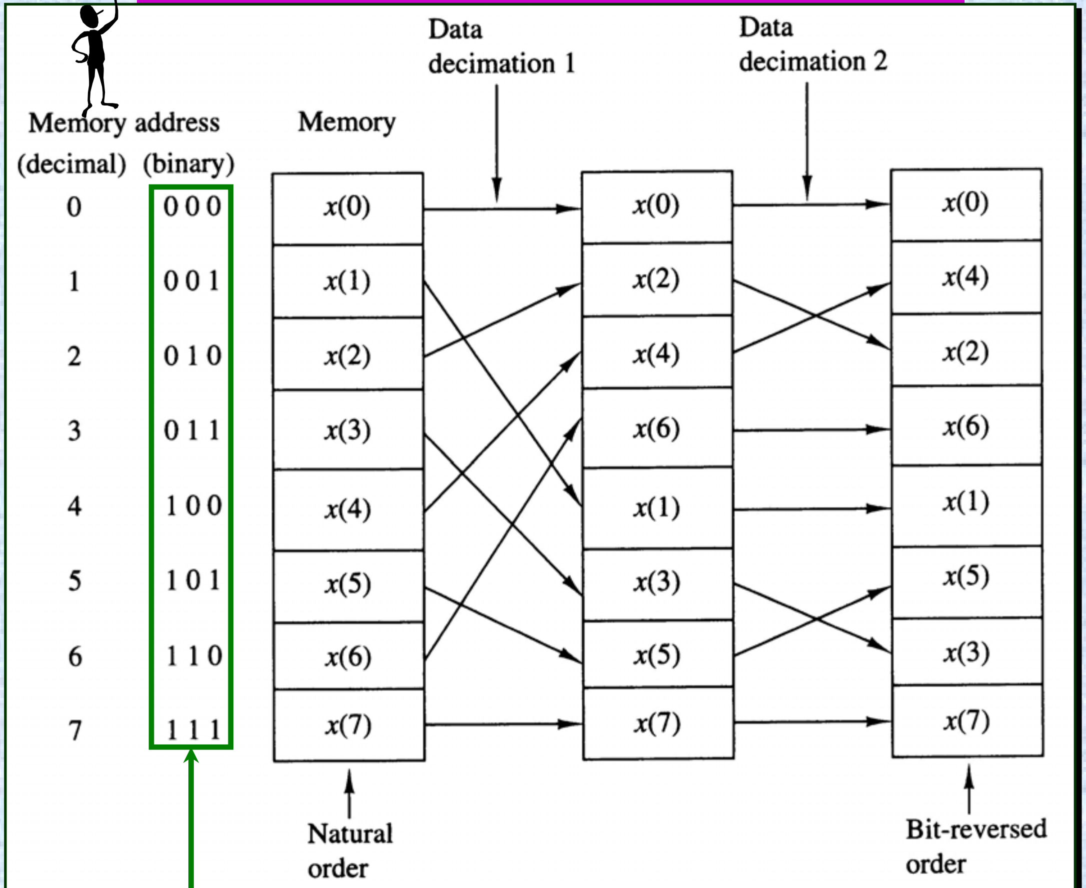


❹ Algoritmul FFT bazat pe segmentarea în timp

❹.❷ 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)

❹ Algoritmul FFT bazat pe segmentarea în timp

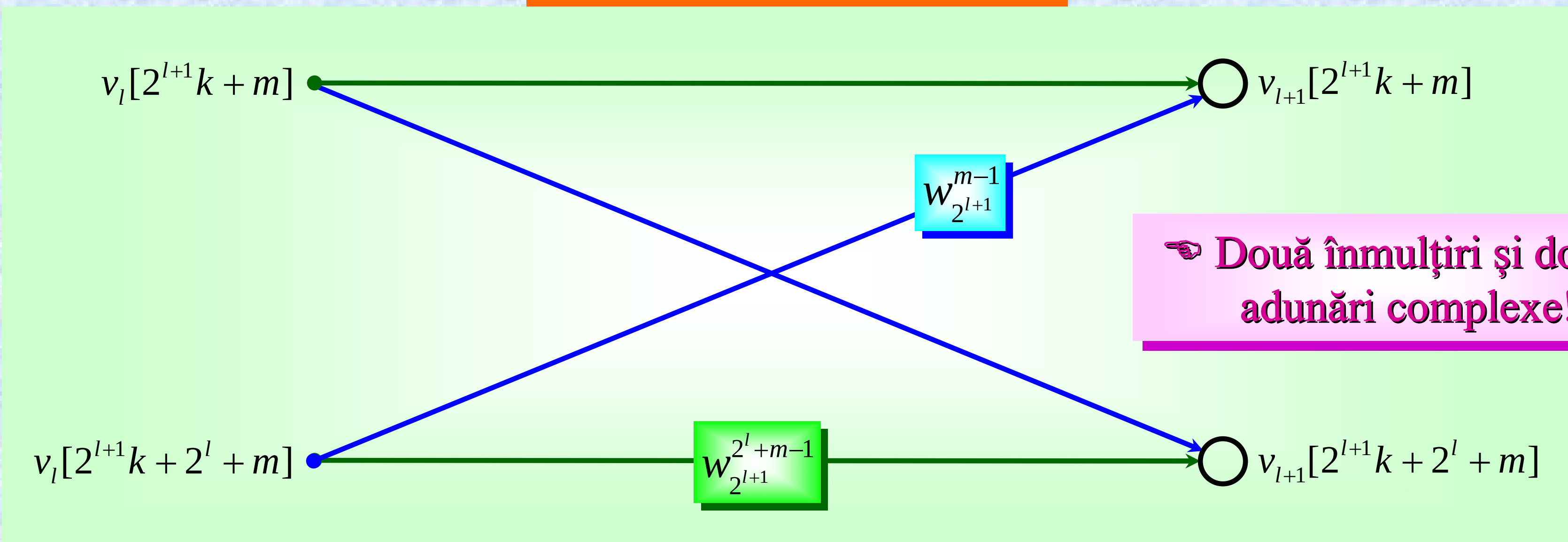
❹.❷ Prima variantă de calcul

- **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



☞ Două înmulţiri şi două adunări complexe!

Număr de operaţii

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

❹ Algoritmul FFT bazat pe segmentarea în timp

❹.❸ A doua variantă (eficientă) de calcul

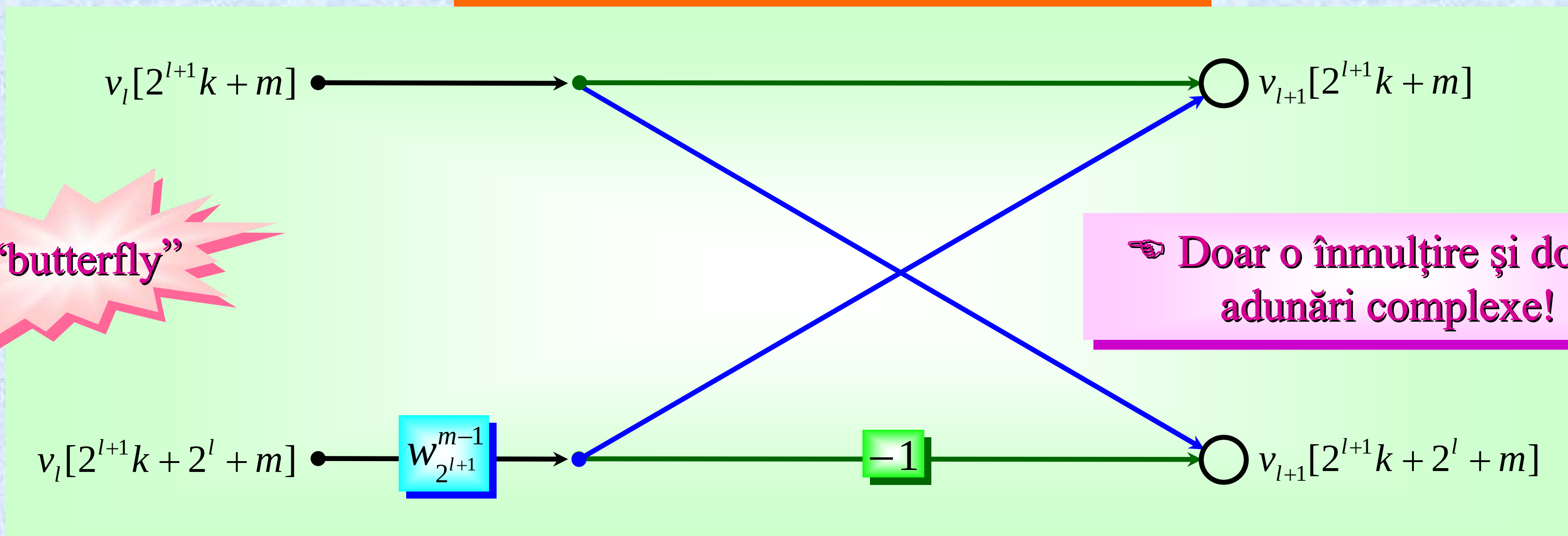
- **Relații de calcul echivalente:**

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

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

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

Schema generică echivalentă de calcul



"butterfly"

☜ Doar o înmulțire și două adunări complexe!

Număr de operații

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

Exemplu

$N = 1024 = 2^{10}$

$\mathcal{O}_0[N] \sim 2^{22}$

$\mathcal{O}_2[N] \sim 10 \times 2^{11}$