DCA 0118 – Processamento Digital de Sinais
Tópico 7.2: Transformada rápida de Fourier

Tiago Barros [1]

[1] ⟨tbarros@dca.ufrn.br⟩

Departamento de Engenharia de Computação e Automação (DCA)
Centro de Tecnologia (CT)
Universidade Federal do Rio Grande do Norte (UFRN)

2022.1

Programa

Conteúdo

- Introdução;
- Dizimação no tempo;
- Dizimação em frequência;
- Considerações práticas;

Bibliografia

Livro texto

Oppenheim, A.V. and Schafer, R.W., 2012. Processamento em tempo discreto de sinais. Tradução Daniel Vieira. 3ª ed.-São Paulo: Pearson Education do Brasil.

- Capítulo 9:
  - Seções 9.0, 9.1 (exceto 9.1.2), 9.2, 9.3, 9.4.

Transformada discreta de Fourier

- Transformada direta (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, \; k = 0, \ldots, N-1$$

- Transformada inversa (IDFT):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk}, \; n = 0, \ldots, N-1$$

Onde

$$W_N = e^{-j2\pi/N}$$

Transformada discreta de Fourier

Cálculo direto:

$$x[n]W_N^{kn}$$

- 1 multiplicação complexa:
  - 4 multiplicações, 2 somas.

Para $X[k]$, $k = 0, 1, \ldots, N-1$:

$\Longrightarrow N^2$ multiplicações complexas.

$\Longrightarrow N(N-1)$ somas complexas.

$$\approx N^2 \text{ multiplicações e somas (MADS)}$$

Transformada rápida de Fourier (FFT)

$\Longrightarrow$ Fatora-se $N = p_1 \times p_2 \times \cdots \times p_\nu$

$\Longrightarrow$ MADS $\propto N[p_1 + p_2 + \cdots + p_\nu]$

$\Longrightarrow$ Se $p_1 = p_2 = \cdots = p_\nu = 2, \longrightarrow N = 2^\nu$ (Radix-2).

Com algoritmo de FFT, MADS $\approx N \log_2 N$.

Estudaremos dois tipos algoritmos:

- FFT com dizimação no tempo.
- FFT com dizimação em frequência.

FFT com dizimação no tempo

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\ &= \sum_{n\,\text{par}} x[n] W_N^{nk} + \sum_{n\,\text{ímpar}} x[n] W_N^{nk} \end{aligned}$$

$\Longrightarrow$ n par: $n = 2r$

$\Longrightarrow$ n ímpar: $n = 2r + 1$

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

$$X[k] = \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k}$$

FFT com dizimação no tempo

Simplificações:

- $W_N^{(2r+1)k} = W_N^k W_N^{2rk}$
- $W_N^2 = e^{-j(2\pi/N)2} = e^{-j2\pi/(N/2)} \longrightarrow W_N^2 = W_{N/2}$

Reescrevendo:

$$
\begin{aligned}
X[k] &= \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k} \\
&= \sum_{r=0}^{N/2-1} x[2r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] W_{N/2}^{rk}
\end{aligned}
$$

FFT com dizimação no tempo

$\Longrightarrow G[k] = \sum_{r=0}^{N/2-1} x[2r] W_{N/2}^{rk}$ é DFT de $N/2$ pontos.

$\Longrightarrow H[k] = \sum_{r=0}^{N/2-1} x[2r+1] W_{N/2}^{rk}$ é DFT de $N/2$ pontos.

Podemos escrever

$$X[k] = G[k] + W_N^k H[k]$$

Observações:

$\Longrightarrow$ MADS $= 2\left(\frac{N}{2}\right)^2 + N = N + \frac{N^2}{2}$

- Para $N > 2 \longrightarrow N + N^2/2 < N^2$

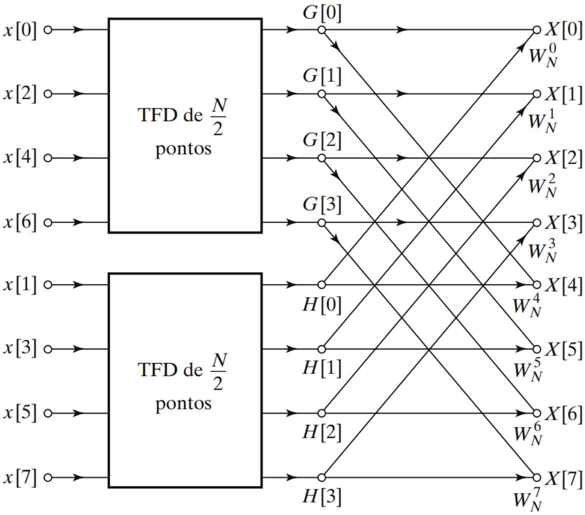
$\Longrightarrow$ DFT de $N/2$ pontos é periódica em k com período $N/2$:

- $G[k + N/2] = G[k]$
- $H[k + N/2] = H[k]$

FFT com dizimação no tempo para $N = 8$ pontos

$$
\begin{aligned}
X[0] &= G[0] + W_N^0 H[0] \\
X[1] &= G[1] + W_N^1 H[1] \\
X[2] &= G[2] + W_N^2 H[2] \\
X[3] &= G[3] + W_N^3 H[3] \\
X[4] &= G[4] + W_N^4 H[4] = G[0] + W_N^4 H[0] \\
X[5] &= G[5] + W_N^5 H[5] = G[1] + W_N^5 H[1] \\
X[6] &= G[6] + W_N^6 H[6] = G[2] + W_N^6 H[2] \\
X[7] &= G[7] + W_N^7 H[7] = G[3] + W_N^7 H[3]
\end{aligned}
$$

FFT com dizimação no tempo para $N = 8$ pontos



FFT com dizimação no tempo para $N = 8$ pontos

Pode-se decompor novamente em DFT's de $N/4$ pontos:

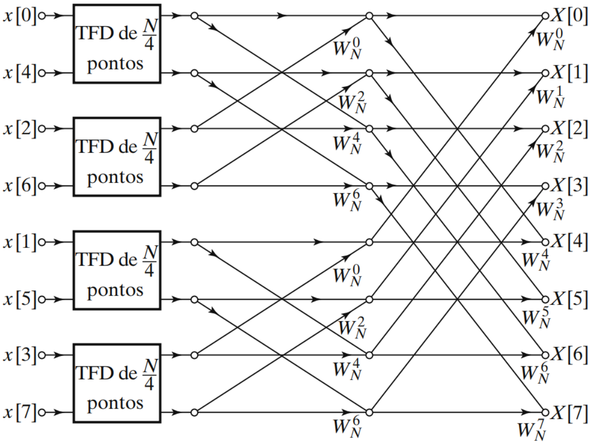
$$\begin{aligned} G[k] &= \sum_{l=0}^{N/4-1} g[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} g[2l+1] W_{N/4}^{lk} \\ H[k] &= \sum_{l=0}^{N/4-1} h[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} h[2l+1] W_{N/4}^{lk} \end{aligned}$$

Pode-se decompor ainda mais em DFT's de $N/8$ pontos e assim sucessivamente, até ter-se apenas DFT's de 2 pontos.

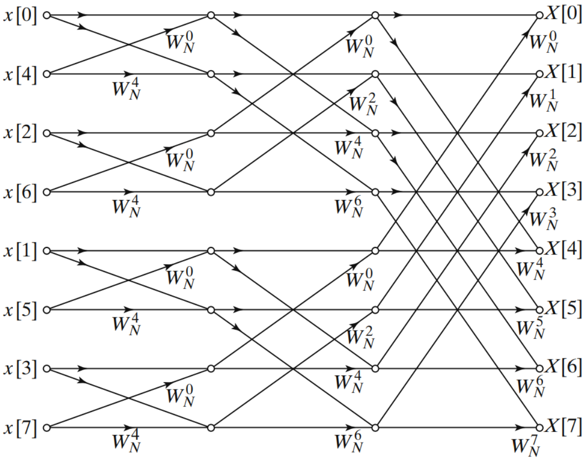
Requer $\nu = \log_2 N$ estágios de cálculo de DFT's. A complexidade resultante é:

$$N_\nu = N \log_2 N \text{ MADS.}$$

FFT com dizimação no tempo para $N = 8$ pontos

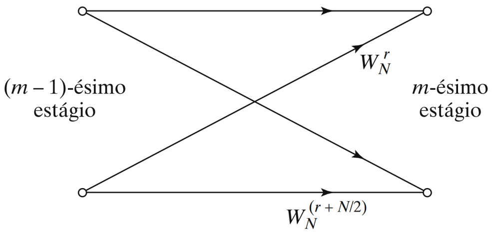


FFT com dizimação no tempo para $N = 8$ pontos



FFT com dizimação no tempo

Diagrama de fluxo da operação básica (borboleta) da FFT com dizimação no tempo:



Simplificação:

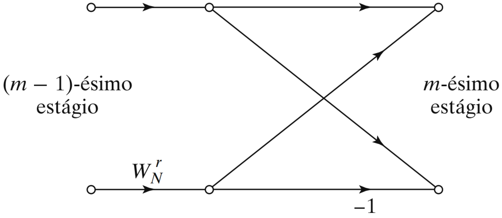
- $W_N^{N/2} = e^{-j(2\pi/N)N/2} = e^{-j\pi} = -1$

Logo:

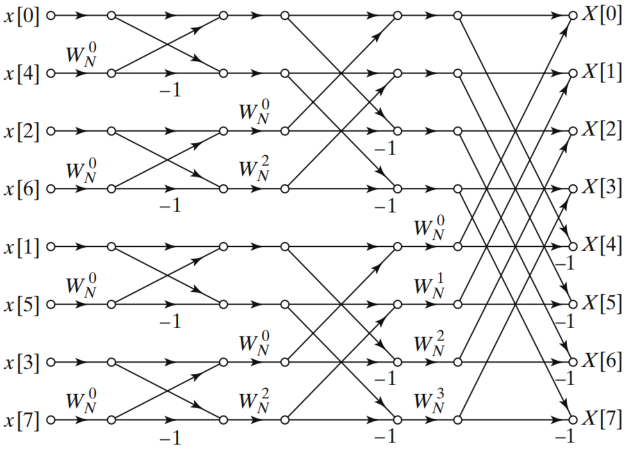
- $W_N^{r+N/2} = W_N^{N/2} W_N^r = -W_N^r$

FFT com dizimação no tempo

Diagrama de fluxo da operação básica (borboleta ou treliça) da FFT com dizimação no tempo, após simplificação:



FFT com dizimação no tempo para $N = 8$ pontos, após simplificação



FFT com dizimação no tempo

Cálculos realizados localmente (*in-place*):

- Cada estágio do cálculo toma um conjunto de N números complexos e os transforma em outro conjunto de N números complexos por meio de operações básicas da treliça (borboleta).
- Ao implementar os cálculos da FFT, podemos imaginar o uso de dois vetores (complexos) de registradores de armazenamento, um para o vetor que está sendo calculado e um para os dados que estão sendo usados no cálculo.
- Denotamos a sequência de números complexos resultantes do m-ésimo estágio do cálculo como $X_m[l]$, em que $l = 0, 1, \ldots, N-1$, e $m = 1, 2, \ldots, \nu$. Por conveniência, definimos o conjunto de amostras de entrada como $X_0[l]$.

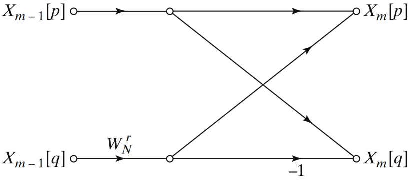
FFT com dizimação no tempo

$\Longrightarrow$ Seja $X_{m-1}[l]$ entrada de cada nó da borboleta no estágio m.

$\Longrightarrow$ Seja $X_m[l]$ saída de cada nó da borboleta no estágio m.

Escrevemos:

$$
\begin{aligned}
X_m[p] &= X_{m-1}[p] + W_N^r X_{m-1}[q] \\
X_m[q] &= X_{m-1}[p] - W_N^r X_{m-1}[q]
\end{aligned}
$$



FFT com dizimação no tempo

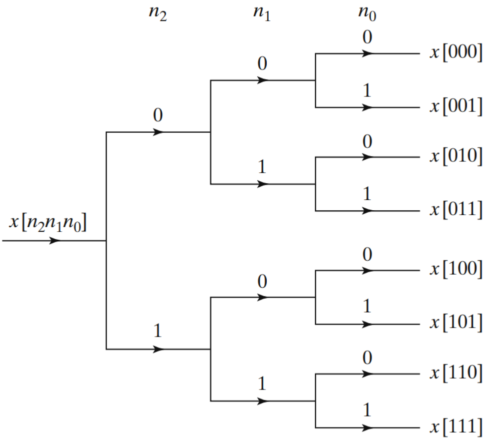
Para que o cálculo possa ser realizado localmente, a sequência de entrada precisa ser armazenada (ou, pelo menos acessada) em uma ordem não sequencial.

A ordem em que os dados de entrada são armazenados e acessados é conhecida como ordem *bit-reversa*. Para FFT com $N = 8$, entrada do primeiro estágio é:

$$
\begin{aligned}
X_0[0] &= X_0[000] = x[000] = x[0],\\
X_0[1] &= X_0[001] = x[100] = x[4],\\
X_0[2] &= X_0[010] = x[010] = x[2],\\
X_0[3] &= X_0[011] = x[110] = x[6],\\
X_0[4] &= X_0[100] = x[001] = x[1],\\
X_0[5] &= X_0[101] = x[101] = x[5],\\
X_0[6] &= X_0[110] = x[011] = x[3],\\
X_0[7] &= X_0[111] = x[111] = x[7],
\end{aligned}
$$

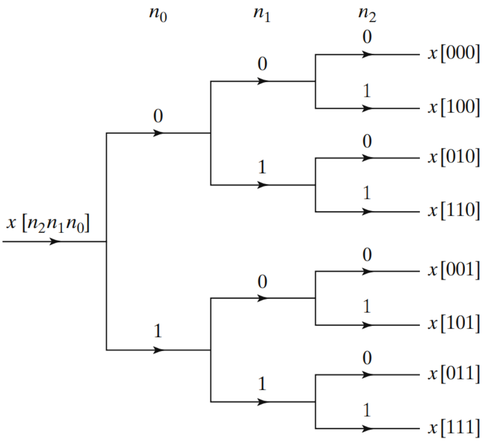
FFT com dizimação no tempo

Diagrama de árvore que representa a ordenação normal
$(X_0(n_0, n_1, n_2) = x(n_2, n_1, n_0))$.



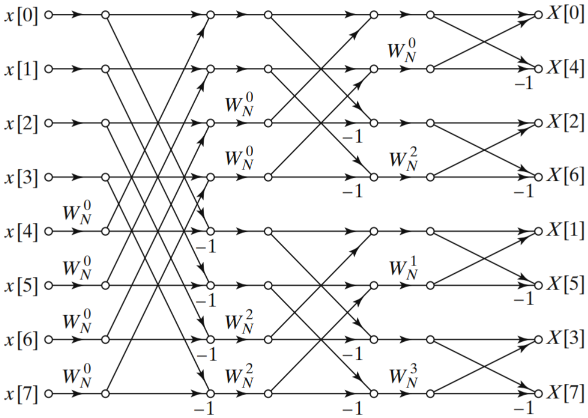
FFT com dizimação no tempo

Diagrama de árvore que representa a ordenação bit-reversa ($X_0(n_0, n_1, n_2) = x(n_2, n_1, n_0)$).



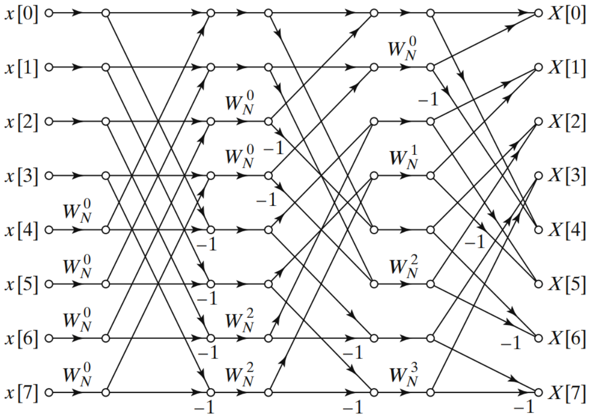
FFT com dizimação no tempo – Formas alternativas

Entrada na ordem normal e saída na ordem bit-reversa (*in-place*).



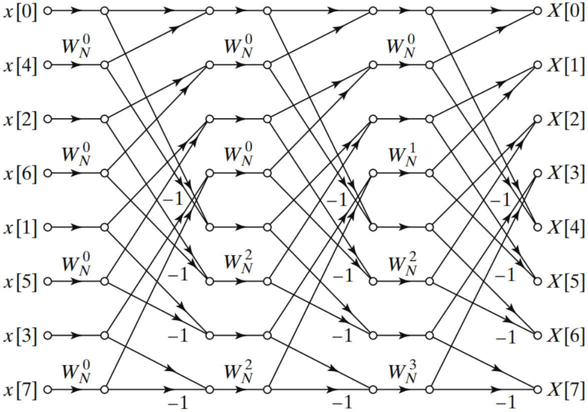
FFT com dizimação no tempo – Formas alternativas

Entrada na ordem normal e saída na ordem normal (não é *in-place*, indexação complicada).



FFT com dizimação no tempo – Formas alternativas

Entrada na ordem bit-reversa e saída na ordem normal (não é *in-place*, indexação idêntica em todos estágios).



FFT com dizimação em frequência

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
&= \sum_{n=0}^{N/2-1} x[n] W_N^{nk} + \sum_{n=N/2}^{N-1} x[n] W_N^{nk}
\end{aligned}
$$

Substituição de variáveis no segundo somatório:

$$
\begin{aligned}
\sum_{n=N/2}^{N-1} x[n] W_N^{nk} &= \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{(n+N/2)k} \\
&= W_N^{(N/2)k} \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{nk}
\end{aligned}
$$

Simplificação:

$$
W_N^{(N/2)} = e^{-j(2\pi/N)(N/2)} = e^{-j\pi} = -1
$$

FFT com dizimação em frequência

Com isso, rescrevemos

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

Separamos saída da DFT em amostras para valores pares e ímpares de k:

$\Longrightarrow$ k par:

$$X[2r] = \sum_{n=0}^{N/2-1} [x[n] + x[n+N/2]]\, W_N^{2rn}, \; r = 0, 1, \ldots, N/2-1$$

$\Longrightarrow$ k ímpar:

$$X[2r+1] = \sum_{n=0}^{N/2-1} [x[n] - x[n+N/2]]\, W_N^{n} W_N^{2rn}, \; r = 0, 1, \ldots, N/2-1$$

FFT com dizimação em frequência

Simplificação:

$$W_N^{2rn} = e^{-j(2\pi/N)(2rn)} = e^{-j(2\pi/N/2)(rn)} = W_{N/2}^{rn}$$

Podemos reescrever:

$\Longrightarrow$ k par:

$$X[2r] = \sum_{n=0}^{N/2-1} g[n] W_{N/2}^{rn}, \; r = 0, 1, \ldots, N/2-1,$$

com $g[n] = x[n] + x[n+N/2]$.

$\Longrightarrow$ k ímpar:

$$X[2r+1] = \sum_{n=0}^{N/2-1} \left[h[n] W_N^{n}\right] W_{N/2}^{rn}, \; r = 0, 1, \ldots, N/2-1,$$

com $h[n] = x[n] - x[n+N/2]$

FFT com dizimação em frequência

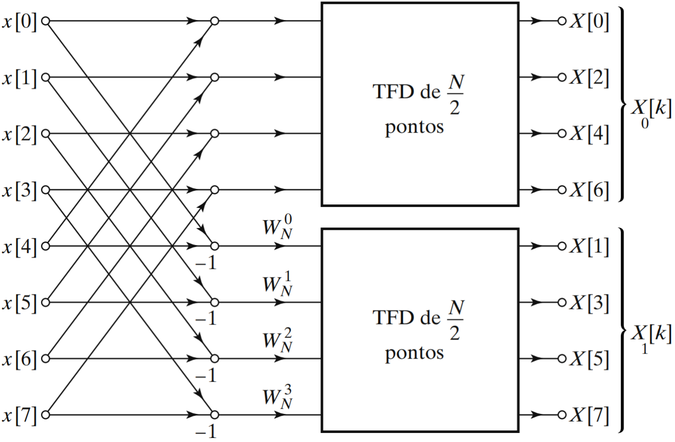
DFT pode ser calculada com algoritmo:

1. Formar sequências $g[n]$ e $h[n]$;
2. Calcular $h[n]W_N^n$;
3. Calcular DFT de $N/2$ pontos de $g[n]$ e $h[n]W_N^n$;

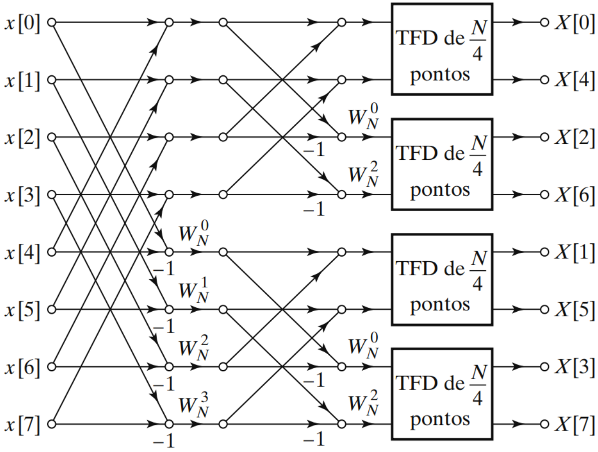
Isso é feito sucessivamente até calcularmos a DFT de 2 pontos (análogo à dizimação no tempo).

Total de cálculos é o mesmo do que na dizimação no tempo ($N \log_2 N$).

FFT com dizimação em frequência para $N = 8$ pontos

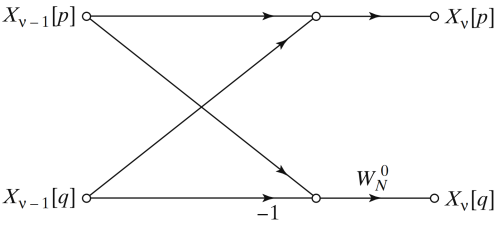


FFT com dizimação em frequência para $N = 8$ pontos



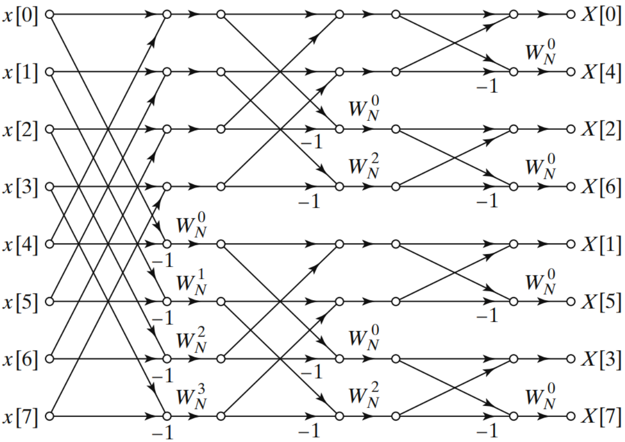
FFT com dizimação em frequência

Diagrama de fluxo da operação básica (borboleta) da FFT de 8 pontos com dizimação em frequência:



$\Longrightarrow$ FFT com dizimação em frequência é a forma transposta da FFT com dizimação no tempo.

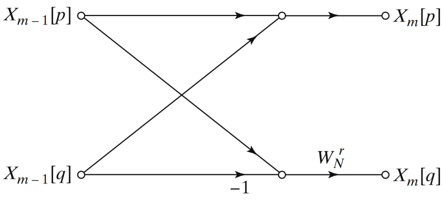
FFT com dizimação em frequência para $N = 8$ pontos



FFT com dizimação em frequência – cálculos realizados localmente

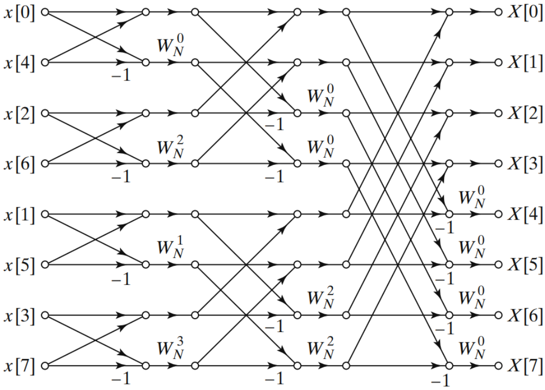
Se denotarmos a sequência de números complexos resultante do m-ésimo estágio do cálculo como $X_m[l]$, em que $l = 0, 1, \ldots, N-1$, e $m = 1, 2, \ldots, \nu$, então a operação borboleta básica tem a forma

$$\begin{aligned} X_m[p] &= X_{m-1}[p] + X_{m-1}[q] \\ X_m[q] &= (X_{m-1}[p] - X_{m-1}[q])W_N^r \end{aligned}$$



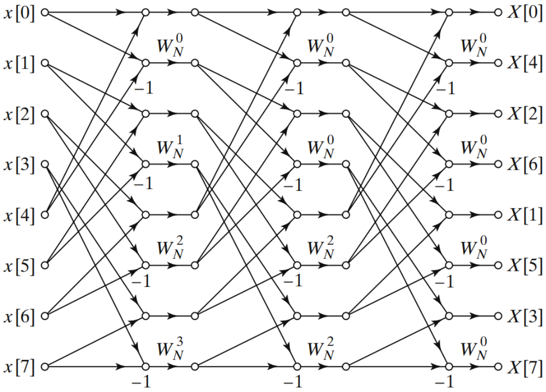
FFT com dizimação em frequência – cálculos realizados localmente (formas alternativas)

Entrada na ordem bit-reversa e saída na ordem normal (computação *in-place*, indexação diferente em cada estágio).



FFT com dizimação – formas alternativas

Entrada na ordem normal e saída na ordem bit-reversa (não é *in-place*, indexação idêntica em todos estágios).



Considerações práticas

1) DFT inversa:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk}$$

DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$$

Implementamos IFFT com FFT, fazendo as seguintes alterações:

- Multiplicamos resultado por $1/N$.
- $W_N = e^{-j2\pi/N} \implies W_N^* = e^{j2\pi/N}$:
  - Conjugamos os coeficientes.

Considerações práticas

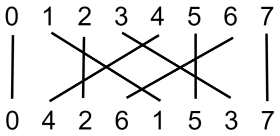
2) Ordenação bit-reversa

Para FFT's com entrada na ordem bit reversa, primeiro passo é ordenação.

- Ordenação é realizada por dois contadores, um na ordem normal e outro na ordem bit-reversa.

$\Longrightarrow$ Cálculos implementados localmente (*in-place*):

- Trocar endereços de armazenamento de entrada e saída (conectar transversalmente).



Considerações práticas

3) Coeficientes:

Duas possibilidades:

- Armazenar em uma tabela coeficientes pré-calculados.
- Calcular em tempo real.

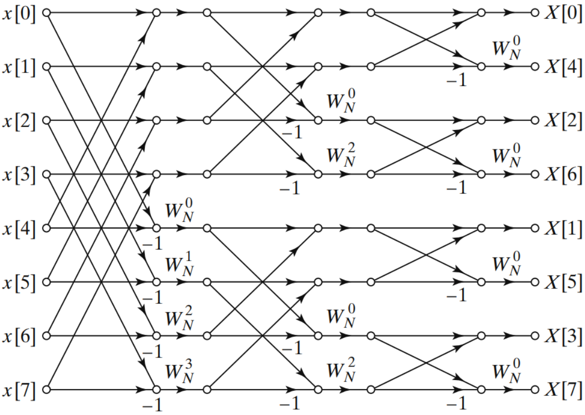
$\Longrightarrow$ Coeficientes W_N podem ser armazenados em ordem normal ou bit-reversa, dependendo do algoritmo.

$\Longrightarrow$ Geração é mais simples na ordem normal.

Considerações práticas

FFT com dizimação em frequência (entrada na ordem normal e saída na ordem bit-reversa):

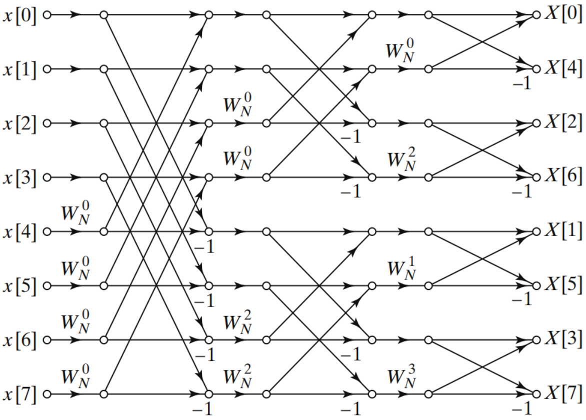
- Potências dos coeficientes estão na ordem normal.



Considerações práticas

FFT com dizimação no tempo (entrada na ordem normal e saída na ordem bit-reversa):

- Potências dos coeficientes estão na ordem bit-reversa.



Considerações práticas

4) Aplicações de FFT:

Exemplo

Implementação da convolução (ou correlação):

- FFT + multiplicação em frequência + IFFT.

$\Longrightarrow$ Exemplo – FFT:

- Entrada: ordem normal.
- Saída: ordem bit-reversa.

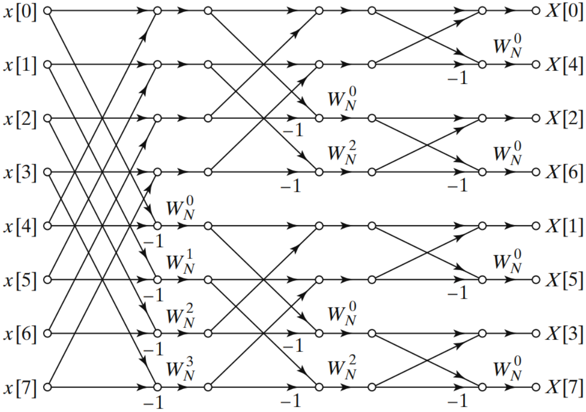
$\Longrightarrow$ Exemplo – IFFT:

- Entrada: ordem bit-reversa.
- Saída: ordem normal.

$\Longrightarrow$ Organizar adequadamente: combinar algoritmos de dizimação no tempo e dizimação em frequência.

Considerações práticas

FFT – algoritmo de dizimação em frequência (entrada na ordem normal, saída na ordem bit-reversa e coeficientes na ordem normal):



Considerações práticas

IFFT – algoritmo de dizimação no tempo (entrada na ordem bit-reversa, saída na ordem normal e coeficientes na ordem normal):