پردازش سیگنال دیجیتال

درس ۲۳

تبدیل سریع فوریه

Fast Fourier Transforms

کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

http://courses.fouladi.ir/dsp

Fast Fourier Transforms

Discrete Fourier Transform

- The DFT pair was given as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}$$

- Baseline for **computational complexity**:
  - Each DFT coefficient requires
    - N complex multiplications
    - $N-1$ complex additions
  - All N DFT coefficients require
    - N^2 complex multiplications
    - $N(N-1)$ complex additions
- Complexity in terms of **real operations**
    - $4N^2$ real multiplications
    - $2N(N-1)$ real additions
- Most fast methods are based on symmetry properties
  - Conjugate symmetry
  - Periodicity in n and k

$$e^{-j(2\pi/N)k(N-n)} = e^{-j(2\pi/N)kN} e^{-j(2\pi/N)k(-n)} = e^{j(2\pi/N)kn}$$

$$e^{-j(2\pi/N)kn} = e^{-j(2\pi/N)k(n+N)} = e^{j(2\pi/N)(k+N)n}$$

The Goertzel Algorithm

- Makes use of the periodicity

$$W_N^{-kN} = e^{j(2\pi/N)Nk} = e^{j2\pi k} = 1,$$

- Multiply DFT equation with this factor

$$X[k] = W_N^{-kN} \sum_{r=0}^{N-1} x[r] W_N^{kr} = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)}.$$

- Define

$$y_k[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(n-r)} u[n-r].$$

- With this definition and using $x[n] = 0$ for $n < 0$ and $n > N-1$

$$X[k] = y_k[n]\Big|_{n=N}.$$

- $X[k]$ can be viewed as the output of a filter to the input $x[n]$

$$y_k[n] = W_N^{-k} y_k[n-1] + x[n],$$

  - Impulse response of filter:

$$W_N^{kn} u[n] = e^{j(2\pi/N)kn} u[n]$$

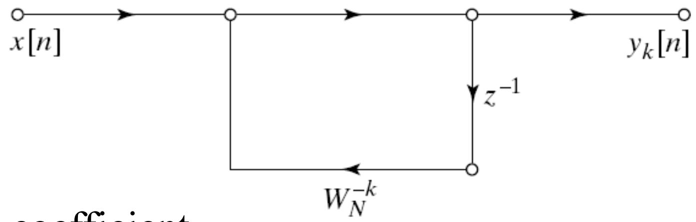
  - $X[k]$ is the output of the filter at time $n = N$

The Goertzel Filter

Goertzel Filter

$$y_k[n] = W_N^{-k} y_k[n-1] + x[n],$$

$$H_k(z) = \frac{1}{1 - e^{j\frac{2\pi}{N}k} z^{-1}}$$



- Computational complexity for each coefficient
  - $4N$ real multiplications
  - $2N$ real additions
  - Slightly less efficient than the direct method

- Multiply both numerator and denominator

$$H_k(z) = \frac{1 - e^{-j\frac{2\pi}{N}k} z^{-1}}{\left(1 - e^{j\frac{2\pi}{N}k} z^{-1}\right)\left(1 - e^{-j\frac{2\pi}{N}k} z^{-1}\right)} = \frac{1 - e^{-j\frac{2\pi}{N}k} z^{-1}}{1 - 2\cos\frac{2\pi k}{N} z^{-1} + z^{-2}}$$

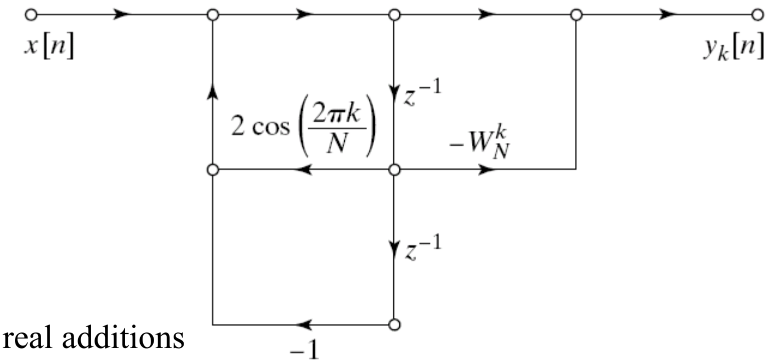
Second Order Goertzel Filter

Second order Goertzel Filter

$$v_k[n] = 2\cos(2\pi k/N)v_k[n-1] - v_k[n-2] + x[n].$$

$$X[k] = y_k[n]\Big|_{n=N} = v_k[N] - W_N^k v_k[N-1].$$

$$H_k(z) = \frac{1 - e^{-j\frac{2\pi}{N}k} z^{-1}}{1 - 2\cos\frac{2\pi k}{N} z^{-1} + z^{-2}}$$



- Complexity for one DFT coefficient
  - Poles: $2N$ real multiplications and $4N$ real additions
  - Zeros: Need to be implement only once
    - 4 real multiplications and 4 real additions
- Complexity for all DFT coefficients
  - Each pole is used for two DFT coefficients
    - Approximately N^2 real multiplications and $2N^2$ real additions
- Do not need to evaluate all N DFT coefficients
  - Goertzel Algorithm is more efficient than FFT if
    - less than M DFT coefficients are needed
    - $M < \log_2 N$

Decimation-In-Time FFT Algorithms

- Makes use of both **symmetry** and **periodicity**
- Consider special case of N an integer power of 2
- Separate $x[n]$ into two sequence of length $N/2$
  - **Even** indexed samples in the first sequence
  - **Odd** indexed samples in the other sequence

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} = \sum_{\text{n even}}^{N-1} x[n] e^{-j(2\pi/N)kn} + \sum_{\text{n odd}}^{N-1} x[n] e^{-j(2\pi/N)kn}$$

- Substitute variables $n = 2r$ for n even and $n = 2r + 1$ for odd

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

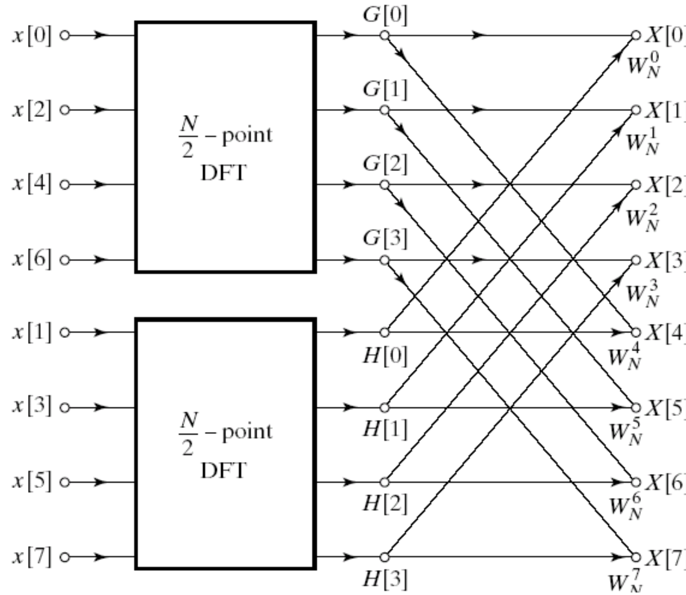
- $G[k]$ and $H[k]$ are the $N/2$-point DFT's of each subsequence

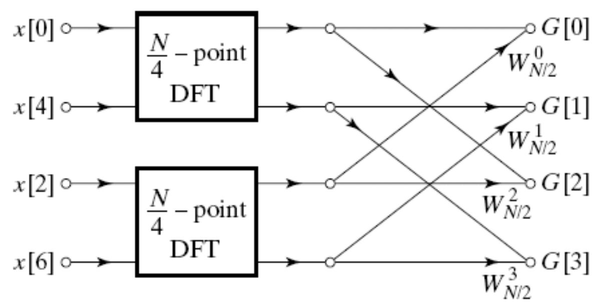
- 8-point DFT example using decimation-in-time:

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

- Two $N/2$-point DFTs
  - $2(N/2)^2$ complex multiplications
  - $2(N/2)^2$ complex additions
- Combining the DFT outputs
  - N complex multiplications
  - N complex additions
- Total complexity
  - $N^2/2 + N$ complex multiplications
  - $N^2/2 + N$ complex additions
  - More efficient than direct DFT

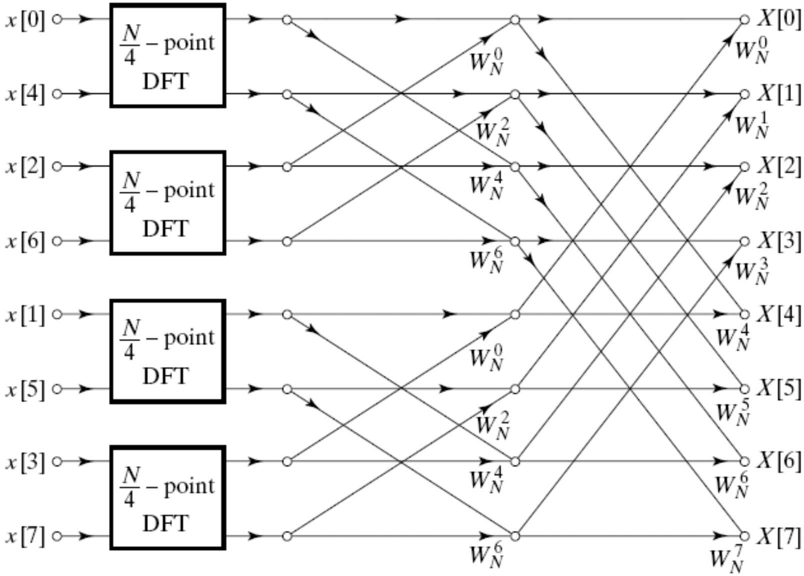
- Repeat same process
  - Divide $N/2$-point DFTs into
    - Two $N/4$-point DFTs
    - Combine outputs



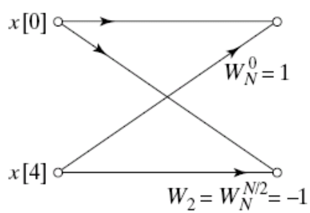


Decimation In Time Cont'd

- After two steps of decimation in time



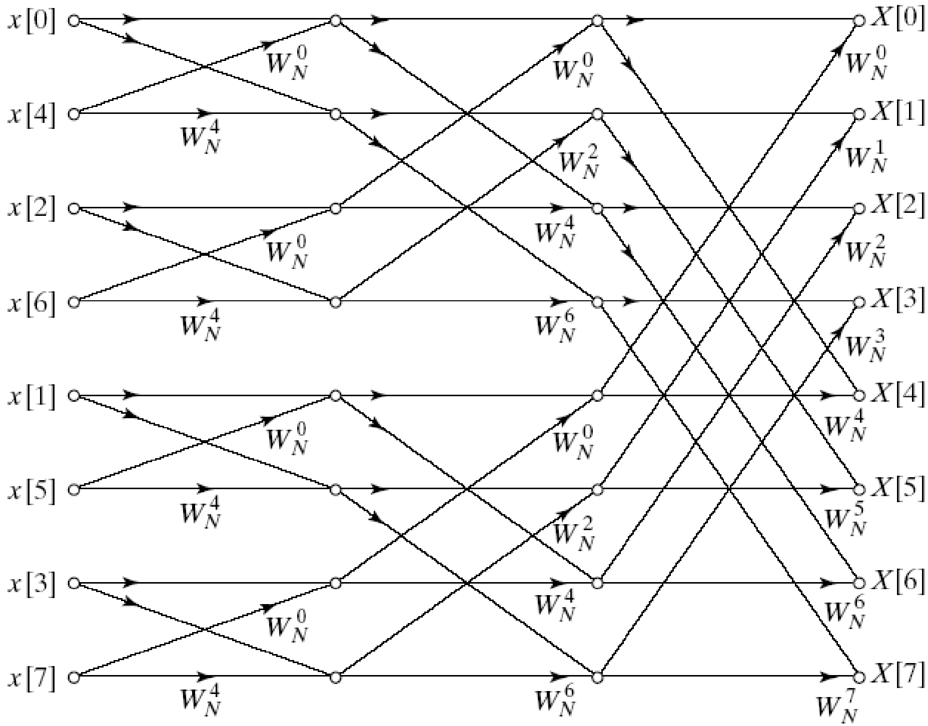
- Repeat until we're left with two-point DFT's



$$\boxed{X[k] = G[k] + W_N^k H[k]}$$

Decimation-In-Time FFT Algorithm

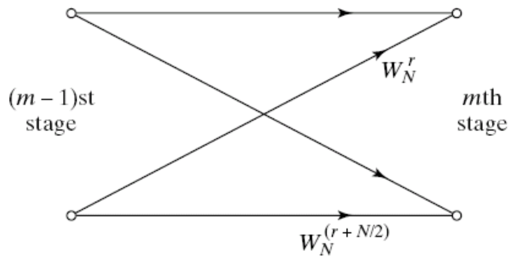
- Final flow graph for 8-point decimation in time



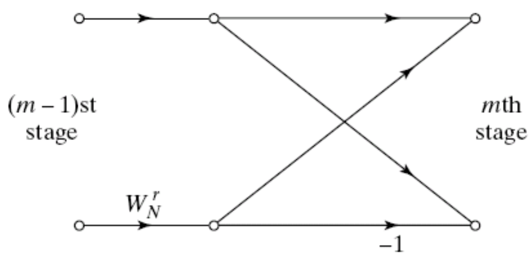
- Complexity:
  - $N\log_2 N$ complex multiplications and additions

Butterfly Computation

- Flow graph constitutes of butterflies



- We can implement each butterfly with one multiplication



- Final complexity for decimation-in-time FFT
  - $(N/2)\log_2 N$ complex multiplications and additions

In-Place Computation

- Decimation-in-time flow graphs require two sets of registers
  - **Input** and **output** for each stage
- Note the arrangement of the input indices
  - Bit reversed indexing

$$X_0[0] = x[0] \leftrightarrow X_0[000] = x[000]$$
$$X_0[1] = x[4] \leftrightarrow X_0[001] = x[100]$$
$$X_0[2] = x[2] \leftrightarrow X_0[010] = x[010]$$
$$X_0[3] = x[6] \leftrightarrow X_0[011] = x[110]$$
$$X_0[4] = x[1] \leftrightarrow X_0[100] = x[001]$$
$$X_0[5] = x[5] \leftrightarrow X_0[101] = x[101]$$
$$X_0[6] = x[3] \leftrightarrow X_0[110] = x[011]$$
$$X_0[7] = x[7] \leftrightarrow X_0[111] = x[111]$$

Decimation-In-Frequency FFT Algorithm

- The DFT equation

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

- Split the DFT equation into even and odd frequency indexes

$$X[2r] = \sum_{n=0}^{N-1} x[n] W_N^{n2r} = \sum_{n=0}^{N/2-1} x[n] W_N^{n2r} + \sum_{n=N/2}^{N-1} x[n] W_N^{n2r}$$

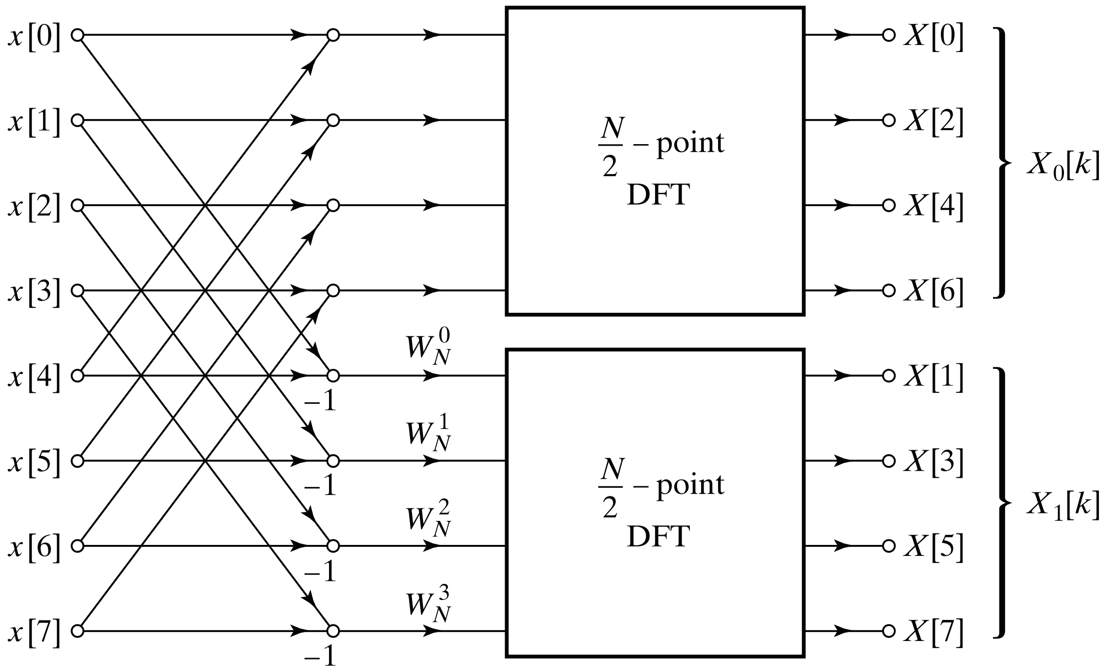
- Substitute variables to get

$$X[2r] = \sum_{n=0}^{N/2-1} x[n] W_N^{n2r} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{(n+N/2)2r} = \sum_{n=0}^{N/2-1} \left(x[n] + x[n+N/2] \right) W_{N/2}^{nr}$$

- Similarly for odd-numbered frequencies

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

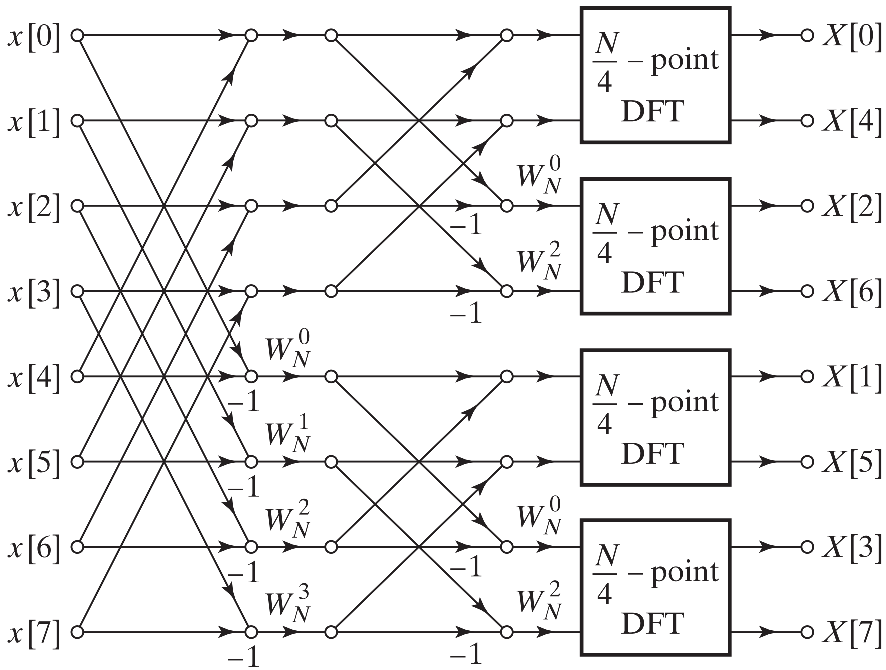
Decimation-In-Frequency FFT Algorithm



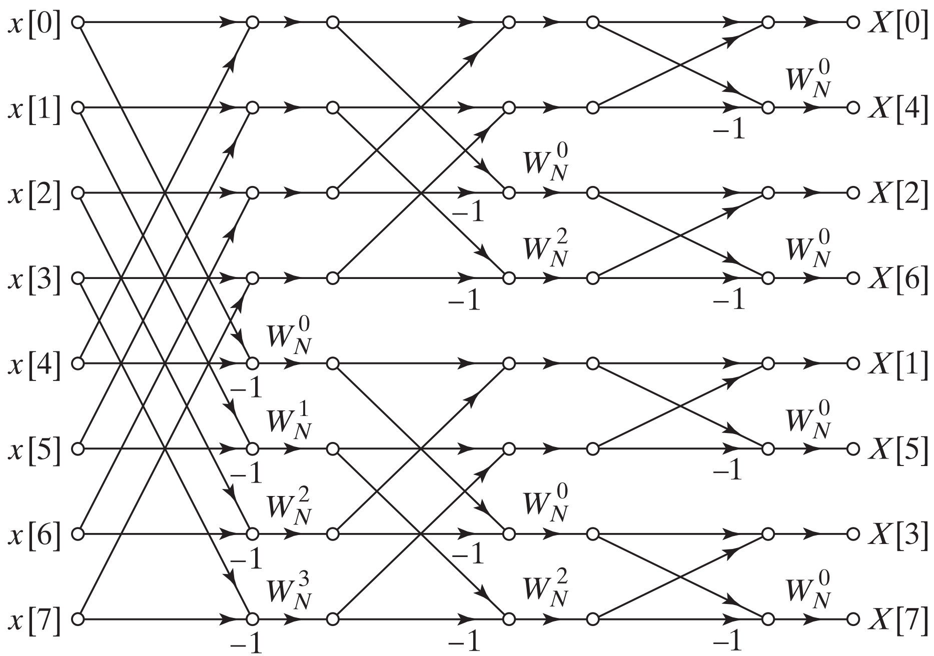
Flow graph of decimation-in-frequency decomposition of an N-point DFT computation into two $(N/2)$-point DFT computations ($N = 8$).

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

Decimation-In-Frequency FFT Algorithm



Decimation-In-Frequency FFT Algorithm



- Final flow graph for 8-point decimation in frequency