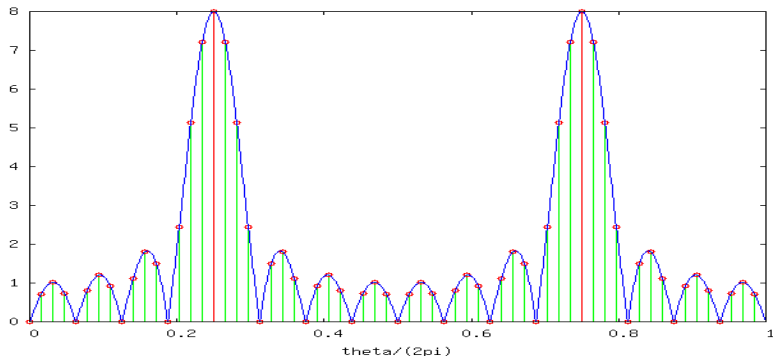


EDISP (NWL3)
(English) Digital Signal Processing
DFT Windowing, FFT

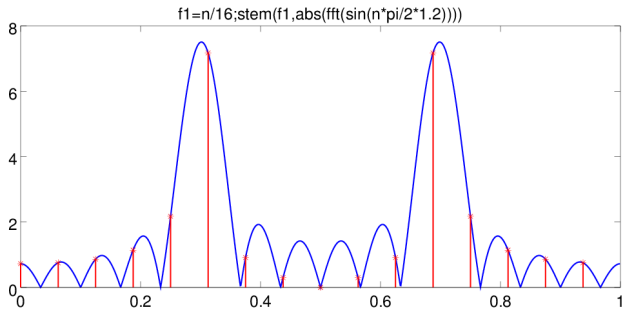
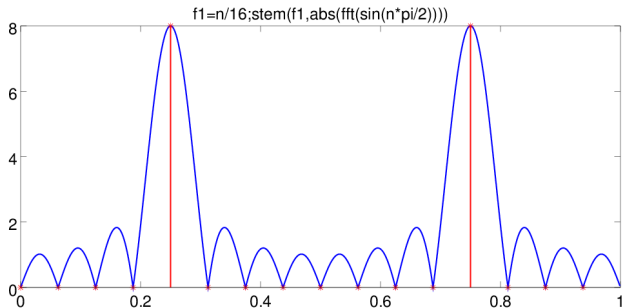
March 24, 2016

DFT resolution 1

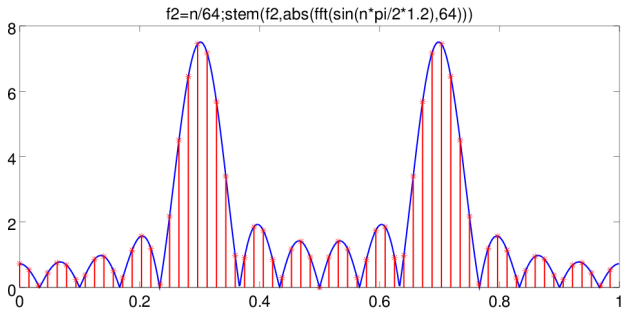
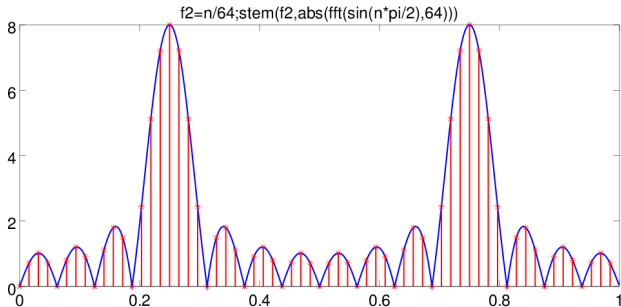
- ▶ N-point DFT \rightarrow frequency sampled at $\theta_k = \frac{2\pi k}{N}$, so the resolution is f_s/N
- ▶ If we want more, we use $N_1 > N$ filling with zeros (zero-padding)
- ▶ but IDFT will give N_1 -periodic signal
- ▶ and the spectrum will have *sidelobes*



DFT resolution 2a: 16pt DFT



DFT resolution 2b: 64pt DFT (zero-padded 16pt signal)



Limited observation time

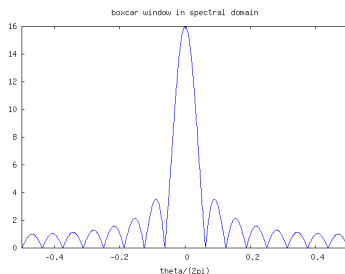
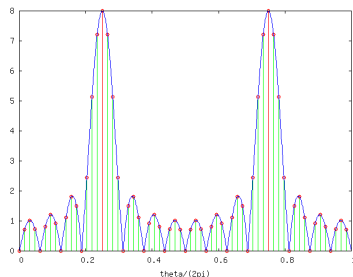
For DFT we used to cut a fragment of the signal

$$x_0[n] = x[n]g[n], \text{ where } g[n] = \begin{cases} 1 & \text{for } n = 0, 1, \dots, N-1 \\ 0 & \text{for other } n \end{cases}$$

$g[n]$ is a window function. Here - a *boxcar window*

Window effect:

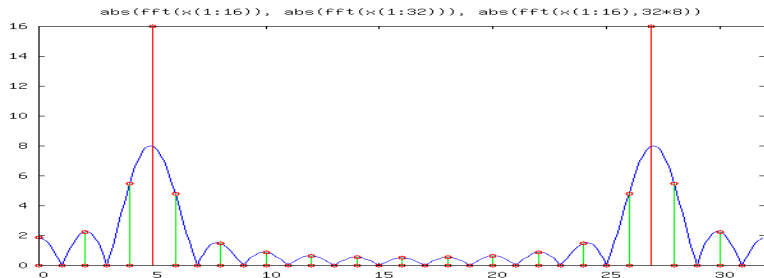
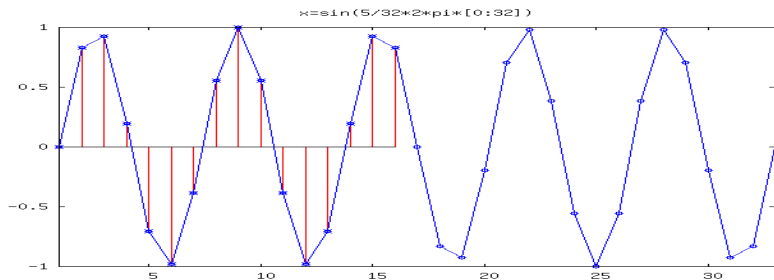
- ▶ selection of a signal fragment
- ▶ $x[n] \cdot g[n]$ in time $\longrightarrow X(\theta) * G(\theta)$ in spectral domain \longrightarrow *sidelobes* or *spectral leakage*



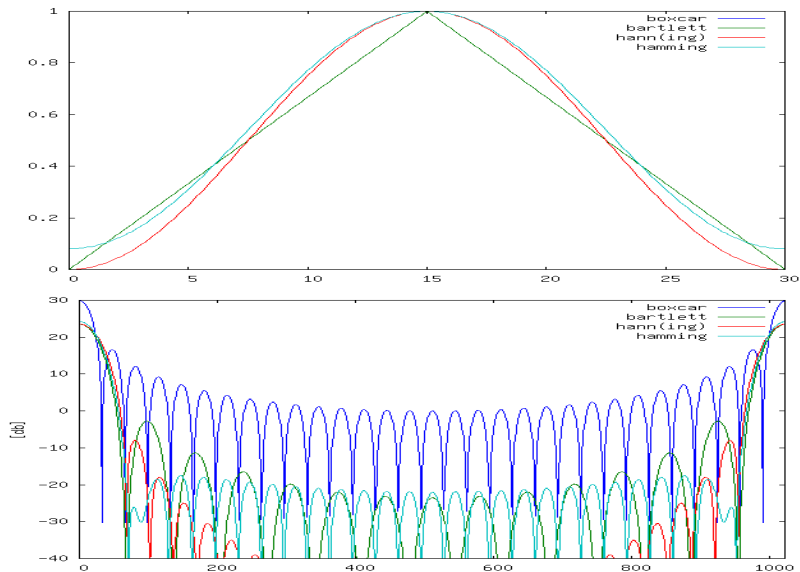
Windowing a pure cosine

Example to be done on slide, temporarily on blackboard (:-).

Leakage example



Window (apodization) functions



Raised cosine window family

- ▶ Hann window: Julius von Hann, 1839 – 1921, Austrian meteorologist; *hanning* is a verb form (*to hann*) $w(n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$
- ▶ Hamming window: Richard Hamming, 1915 – 1998, American mathematician; $w(n) = 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N-1}\right)$
- ▶ Blackman window $w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$

Kaiser window

(D. Slepian, H.O. Pollak, H.J. Landau, around 1961, *Prolate spheroidal wave functions* ...)

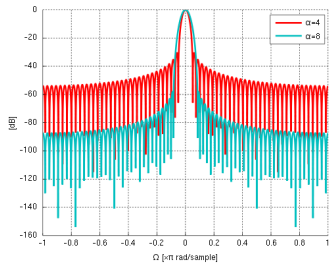
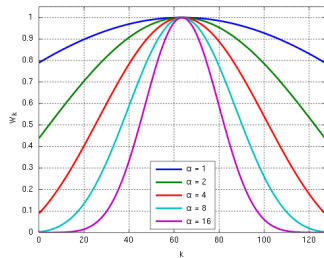
- ▶ time limited sequence with energy concentrated in finite frequency interval
- ▶ a family of windows with many degrees of freedom
- ▶ Kaiser (1974) – an approximation to optimal window: standard method to compute the optimal window was numerically ill-conditioned.

$$w_n = \begin{cases} \frac{I_0\left(\alpha\sqrt{1-\left(\frac{2n}{N}-1\right)^2}\right)}{I_0(\alpha)} & \text{if } 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

I_0 – zeroth order modified Bessel function of the first kind,

- ▶ α (real number) determines the shape of the window:
 - ▶ $\alpha = 0$ gives Boxcar,
 - ▶ $\alpha = 4$ gives -30 dB first sidelobe, -50 asymptotic,
 - ▶ $\alpha = 8$ gives -60 dB first sidelobe, -90 asymptotic,

Kaiser window



Fast DFT algorithms \longrightarrow FFT

- ▶ Direct computation with pre-computed *twiddle factors*

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

$$X(e^{j\theta_k}) = \sum_{n=0}^{N-1} x(n)(W_N)^{kn}$$

\longrightarrow complexity: N^2 complex multiplications & additions

- ▶ Goertzel algorithm: $X(k) = y_k(N)$, where

$$y_k(n) = \sum_{r=0}^{N-1} x(r)W_N^{-k(n-r)}$$

\longrightarrow filtering: $y_k(n) = x(n) + y_k(n-1) \cdot W_N^{-k}$

Also N^2 , but after decomposition majority is real \times real (see next slide).

Useful when not all N frequencies are needed.

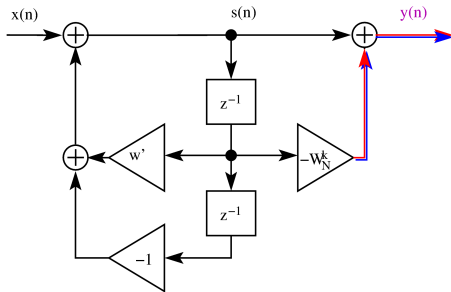
- ▶ Divide-by-two (or decimation) in time \longrightarrow FFT algorithm, complexity $N \log_2(N)$

Goertzel algorithm (1958)

Calculate a single sample of DFT (at $\omega = \omega_k$) by filtering

Gerald Goertzel (1919 – 2002), theoretical physicist, worked with Manhattan Project and later Sage Instruments and IBM

- ▶ A convolution with sinusoid: $s(n) = x(n) + 2 \cos(\theta_k)s(n-1) - s(n-2)$
- ▶ After N samples $X(k)$ is computed as $X(k) = y(N) = s(n) - e^{-j\theta_k} s(n-1)$



$$W_N^k = (e^{-j\frac{2\pi}{N}})^k = e^{-j\theta_k}$$

$$w' = W_N^{-k} + W_N^k = 2 \cos\left(\frac{2\pi k}{N}\right) = 2 \cos(\theta_k)$$

$$-1 = W_N^k \cdot W_N^{-k}$$

... and many versions with special tricks

Fast DFT algorithms \longrightarrow FFT

Decimation in time **FFT** (first stage):

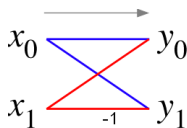
$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \\ &= \sum_{n \text{ even}} x(n) W_N^{nk} + \sum_{n \text{ odd}} x(n) W_N^{nk} = \\ &= \sum_{r=0}^{N/2-1} x(2r) (W_{N/2})^{rk} + W_N^k \sum_{r=0}^{N/2-1} x(2r+1) (W_{N/2})^{rk} \end{aligned}$$

radix-2 FFT

$$X(k) = \sum_{n \text{ even}} x(n) W_N^{nk} + \sum_{n \text{ odd}} x(n) W_N^{nk} =$$

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

- ▶ If $N = 2^L \dots$ We can continue with this trick - decimating each half into sub-halves, each sub-half into sub-sub \dots L times
- ▶ for $k > N/2$, $W_N^k = -W_N^{k-N/2}$ and $\text{FT}_{N/2}$ is periodic with period $N/2$
- ▶ DFT with size 1 is rather trivial

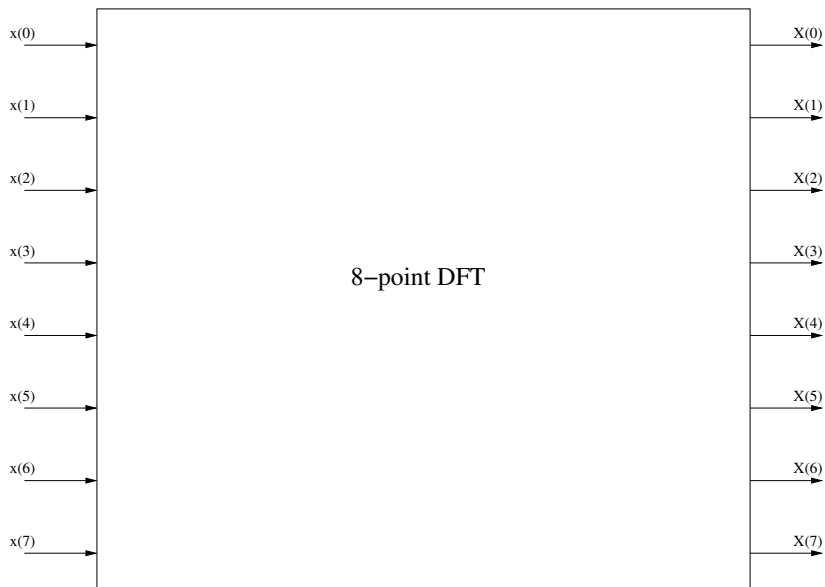


Effect: We have L layers of $N/2$ butterflies. Each butterfly is one multiplication, one addition, one subtraction. In the result, we have $O(N \log_2 N)$ operations

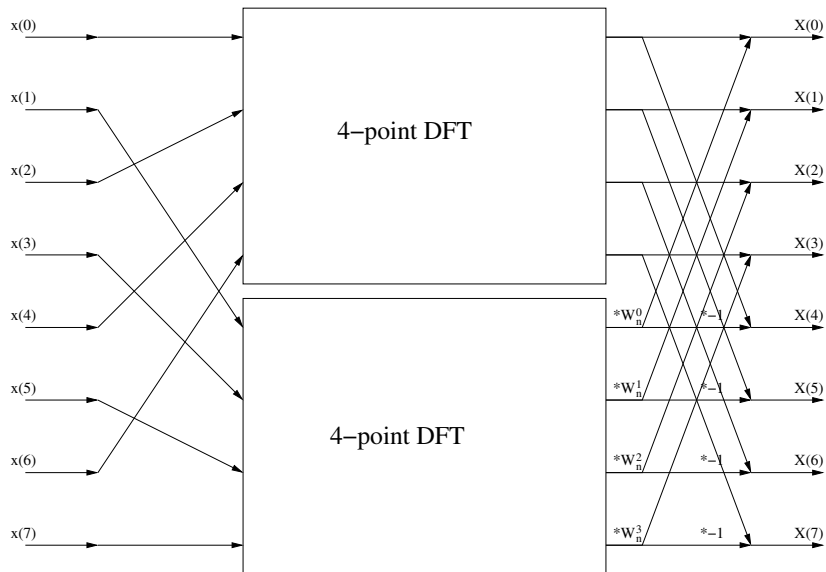
FFT inventors

James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. 19, pp. 297-301 (1965).

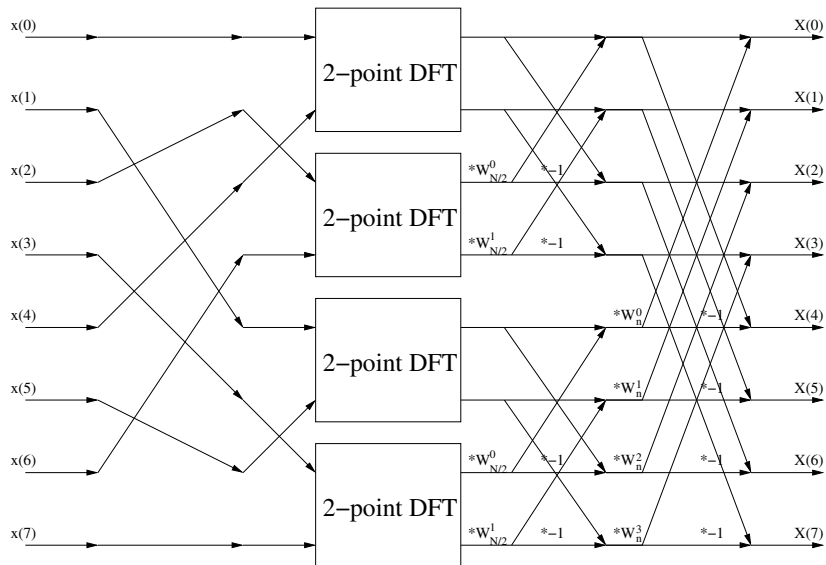
8-point radix-2 FFT



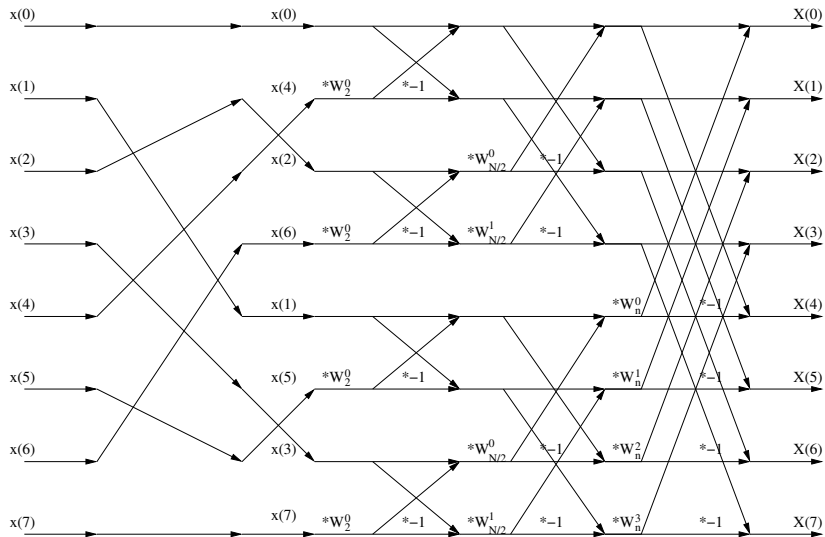
8-point radix-2 FFT



8-point radix-2 FFT

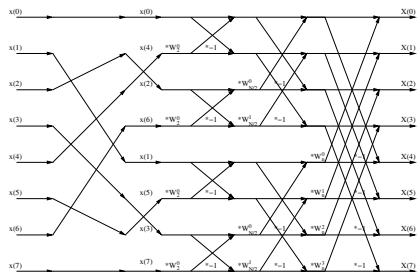


8-point radix-2 FFT



Indexing for FFT

How to describe the sequence of numbers: 0, 4, 2, 6, 1, 5, 3, 7?



- ▶ $0 = 000_2$ is at position 000_2
- ▶ $4 = 100_2$ is at position 001_2
- ▶ $2 = 010_2$ is at position 010_2
- ▶ $6 = 110_2$ is at position 011_2
- ▶ $1 = 001_2$ is at position 100_2
- ▶ $5 = 101_2$ is at position 101_2
- ▶ $3 = 011_2$ is at position 110_2
- ▶ $7 = 111_2$ is at position 111_2

→ bit-reversal does the job!

Processors designed for FFT do have the bit-reversal mode of indexing. (And they do a butterfly in one or two cycles)

Decimation in frequency FFT

- ▶ We split the definition formula for k even ($= 2r$) or odd ($= 2r + 1$)
 - ▶ We note that $W_N^{2nr} = W_{N/2}^{nr}$ or $W_N^{n(2r+1)} = W_N^n \cdot W_{N/2}^{nr}$
 - ▶ Further, for $n > N/2$ $W_N^n = -W_N^{n-N/2}$
 - ▶ and so on - please sketch the DIF FFT diagram by yourselves
- here, we need to re-index the frequencies...

Specials

- ▶ Non-radix2 FFT - slower than radix2, but still faster than direct
- ▶ Chirp-z transform - one use of it is to calculate FT for θ 's not equal to $2\pi/N$
- ▶ Non-uniform FFT ...
- ▶ FFTW - the Fastest FFT in the West - a free library, used by many free and commercial products (Frigo & Johnson from MIT)
- ▶ FFTS - in the South - (New Zealand)

Summary

Fourier transforms:

- ▶ DTFT - spectrum of a discrete-time signal (defined for a limited-energy signal or a limited mean power signal in a different manner) *periodic, continuous or discrete function of θ*
- ▶ DFT - samples of DTFT of a limited duration signal (or a segment....) *periodic, discrete $X(k)$*
- ▶ FFT - a trick (method[s]) to compute DFT efficiently

To window or not to window?

- ▶ If we need to *analyse* the signal - YES,
- ▶ If we need to manipulate spectrum and then reconstruct the signal back - NO.