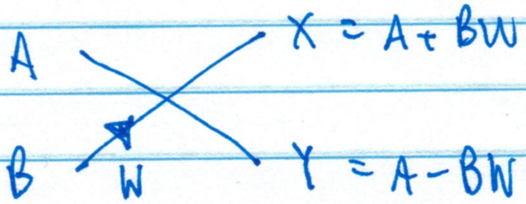
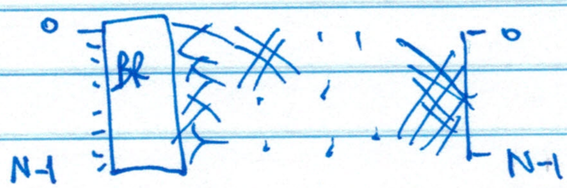
EEC 281

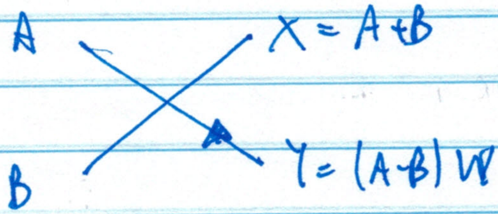
FFT

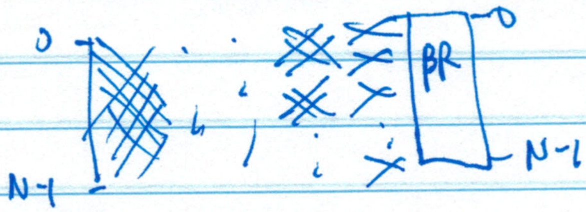
- Radix-2 Dec. In Time (DIT)





- Radix-2 Dec. In Freq (DIF)





- Radix-4 DIT

- Radix-4 DIF

Matlab

indexes from 1 - ...

can not index (0)

Read an array : $b = data(Addr \underline{+1})$

Write an array : $data(Addr \underline{+1}) = a + c$

I. Common-Factor FFTs

- Most common
- Cooley-Tukey FFTs
- Factors of N to decompose the DFT.

A) Radix-r

- $N = r^k$, k = positive integer
- All butterflies are the same
- Radix-r butterflies
- N/r butterflies per stage
- $k = \log_r N$ stages

radix-2

N = 64, 1024, ...

B) Mixed-Radix

- Different types of butterflies
- Is necessary if $N \neq r^k$

Ex: radix-4, N = 32

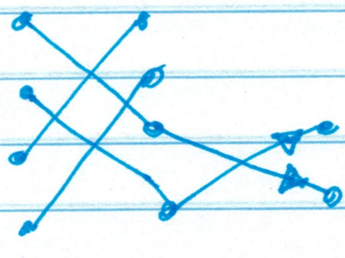
R4 butterfly + R4 + R2

II. Prime Factor FFTs

- N = product of relatively prime numbers
- Ex: $7 \times 11 \times 13 = 1003$
- Advantage: no W_N twiddle factors!
- Disadvantages:
  - Input data irregularly sorted
  - Output " " "
  - Butterfly addr " "

III. Other FFTs

- Split Radix, 1989 Vetterli



- FFTW

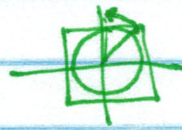
FFT signal growth

- DFT $\sum_{0}^{N-1}$ → growth by N

1) Floating point
   expensive: area, power (energy), throughput

2) Fixed point with scaling by $\frac{1}{r}$ each stage

- First stage is a special case



3) Block Floating Point

Calc. IFFT out = IFFT(in)
(Assume we have an FFT processor)

0) Design a seperate IFFT

1) Swap real and imag parts

a = fft(imag(in) + j·real(in))

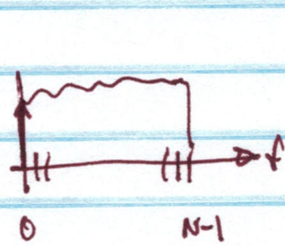
out = imag(a) + j·real(a)

2) Using conjugates

a = fft(conj(in))

out = conj(a)

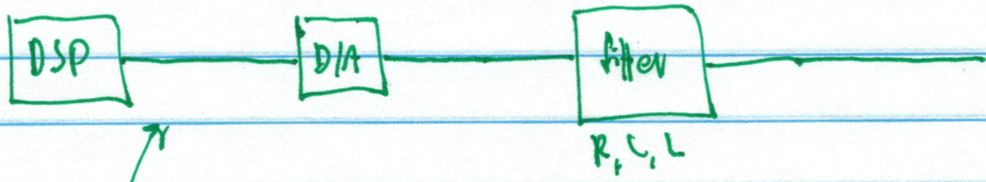
3) A simple indexing change



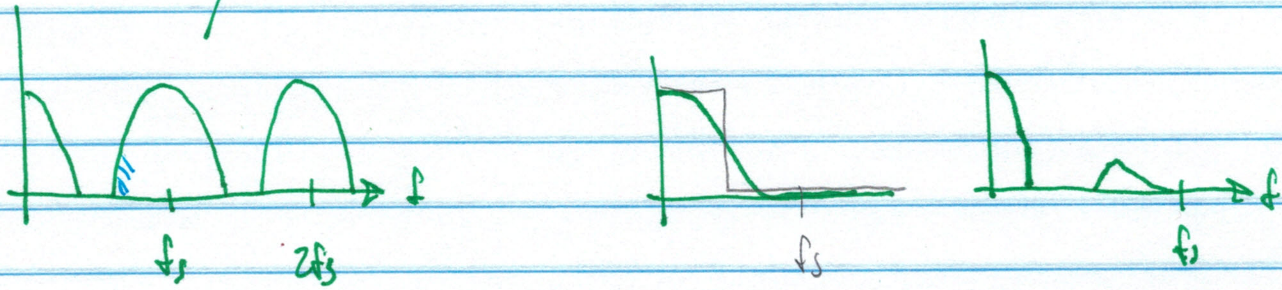
a = fft(in)

out = [a(0) a(N-1 : -1 : 1)]

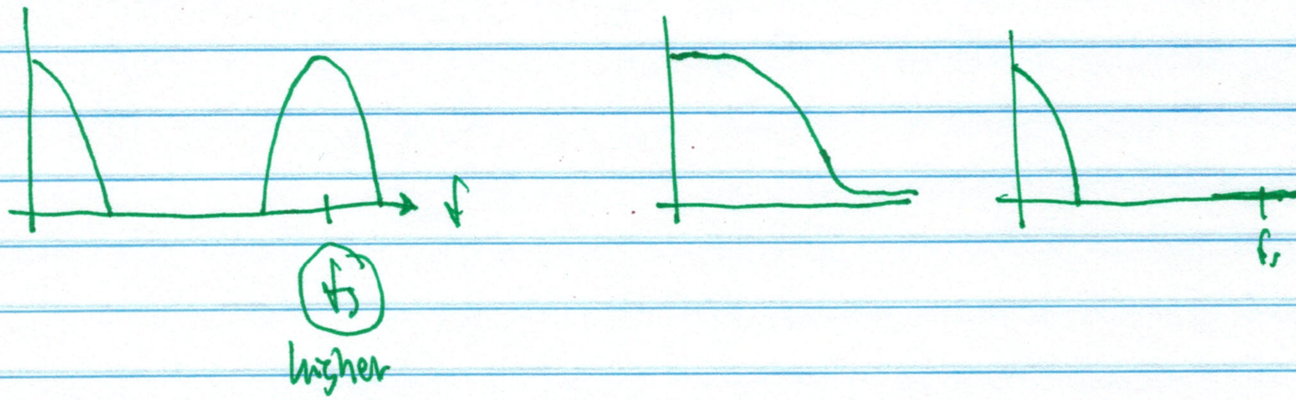
Multi-Rate Signal Processing



(1)



(2)



Upsampling = operation that increases the sample rate

Downsampling = " " decreases " " "

Ref: DeFatta

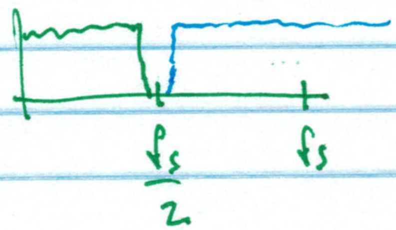
Advantages of lower sample rates:

- Likely/May require less processing
- May require less power
- Likely require less storage

Advantages of higher sample rates:

- May simplify computation
- " " analog circuitry

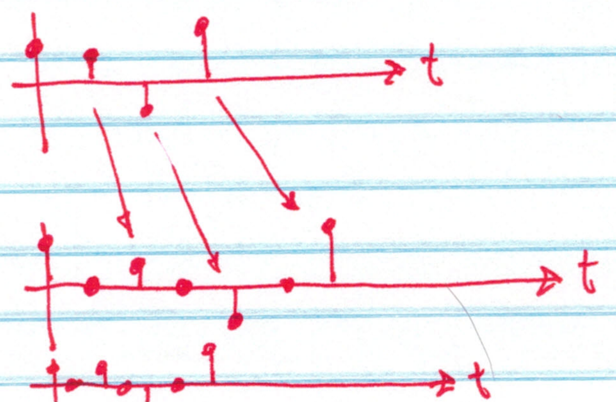
Nyquist



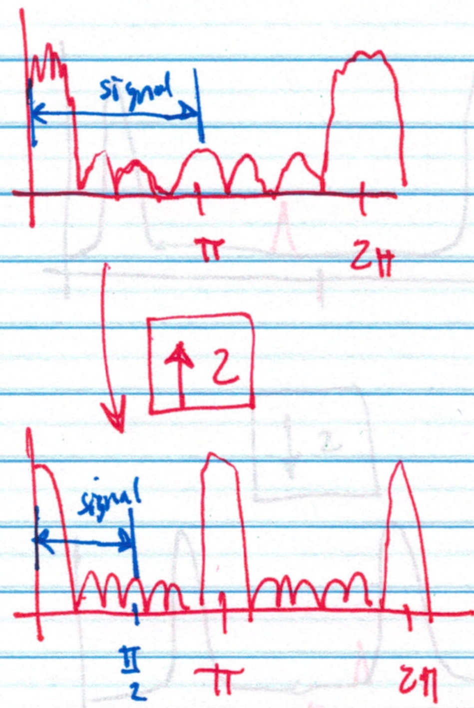
Upsampling or Interpolation

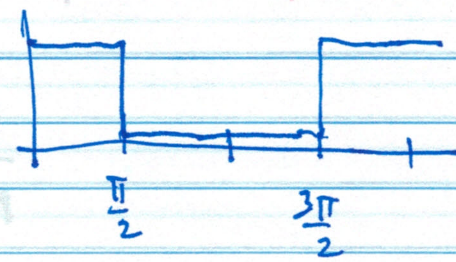
To upsample by a factor I, add I-1 zeros between the original samples

$\uparrow I$, Ex: $\uparrow 2$



Ex:



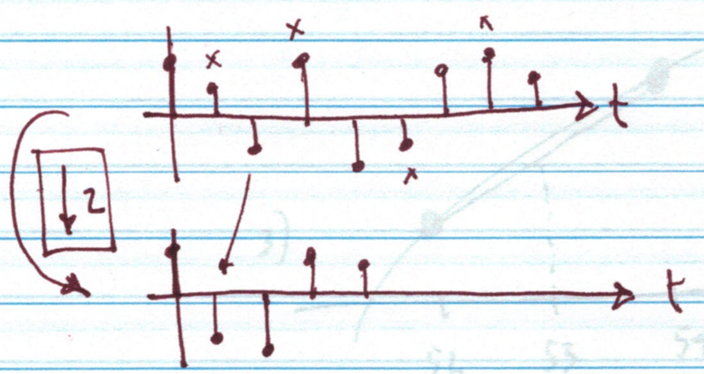


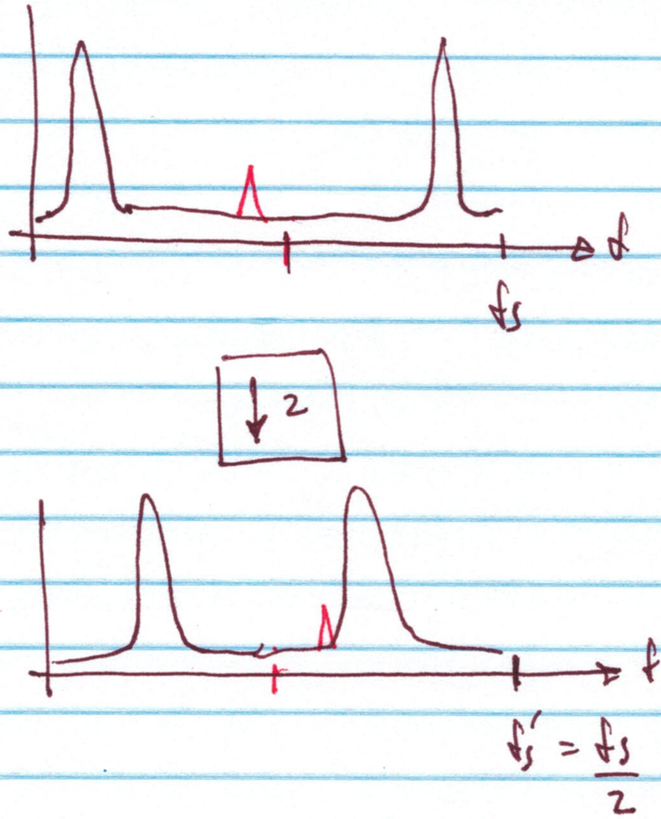
Anti-Image Filter

Downsampling or Decimate

To downsample by a factor D, keep one of every D samples on a periodic basis

$\downarrow D$ $\downarrow 2$





Homework / Project 4

output_ver.m

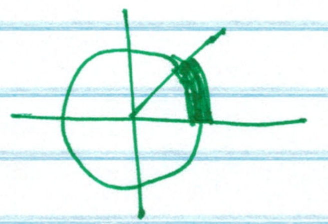
out = 12417 + j * 48112;

matlab

out2 = out / 2^16;

diff(_ _)

2)



3)