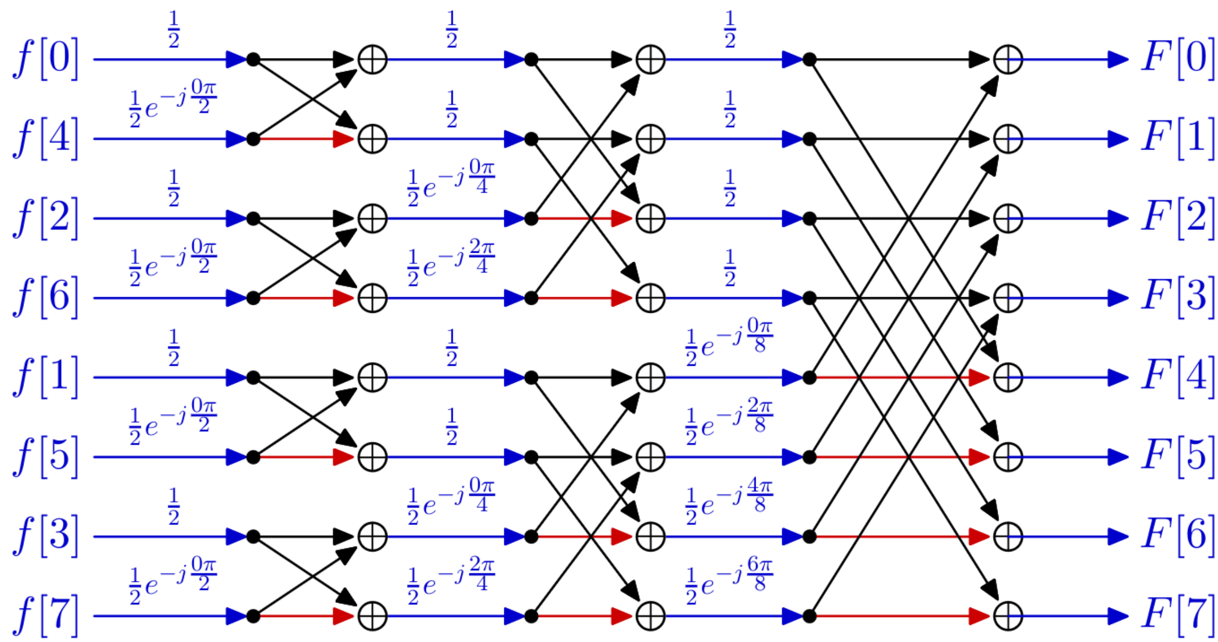
6.300: Signal Processing

Tutorial Topic Fast Fourier Transform
Author Titus K. Roesler (tkr@mit.edu)

Schematic: Decimation-in-Time FFT Algorithm



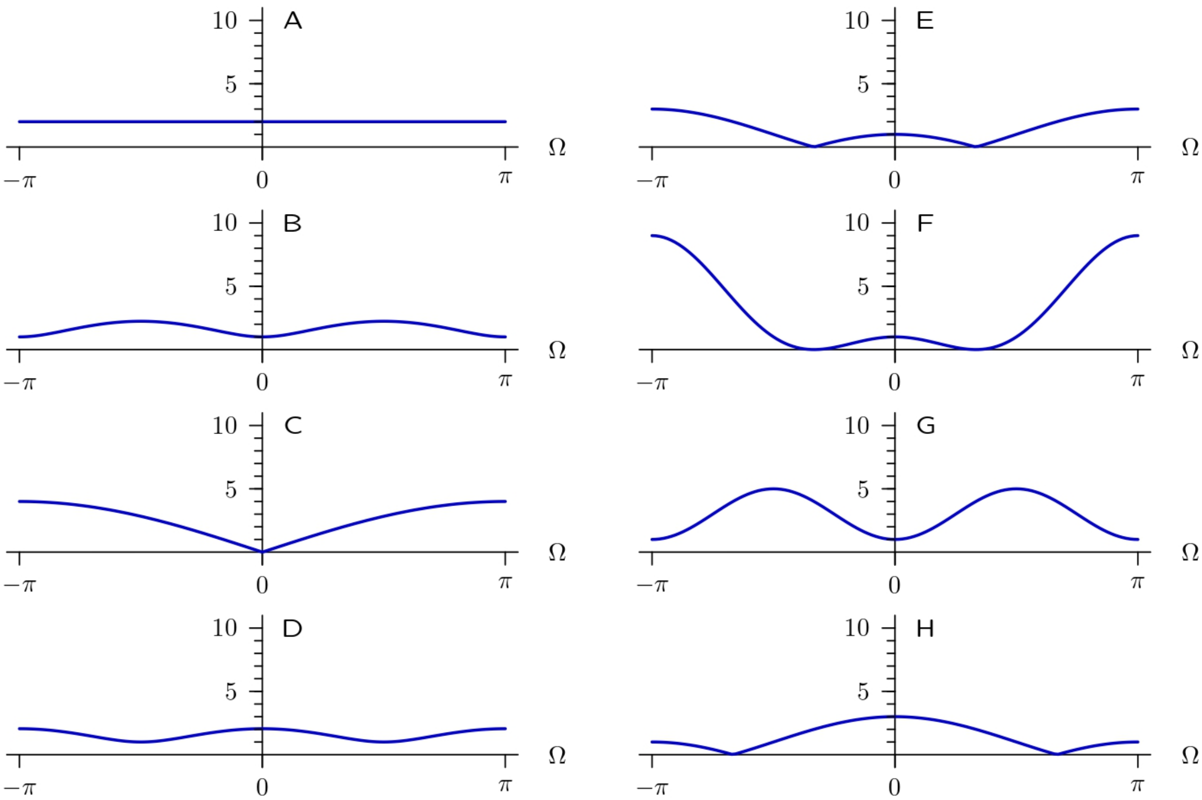
Inverse FFT

In lecture, we discussed FFT algorithms for computing the DFT. Because the analysis and synthesis equations for the DFT are so similar, we need only make a few tweaks to derive inverse FFT algorithms for the computing the inverse DFT. How would you change the code below to compute the inverse DFT?

```
from math import e, pi
def FFT(x):
    N = len(x)
    if N != 2 * N//2:
        print('N must be a power of 2')
        exit(1)
    if N == 1:
        return x
    xe = x[::2]
    xo = x[1::2]
    Xe = FFT(xe)
    Xo = FFT(xo)
    X = []
    for k in range(N//2):
        X.append((Xe[k] + e**(-2j*pi*k/N)*Xo[k]) / 2)
    for k in range(N//2):
        X.append((Xe[k] - e**(-2j*pi*k/N)*Xo[k]) / 2)
    return X
```

Quiz Review: Composite Systems

The following plots should the magnitudes of the frequency responses of eight discrete-time, linear, time-invariant systems.



Quiz Review: Composite Systems

Match each unit-sample response below

- $h_1[n] = g_1[n] = \delta[n] - \delta[n-1] + \delta[n-2]$
- $h_2[n] = g_2[n] = \delta[n] + \delta[n-1] - \delta[n-2]$
- $h_3[n] = g_1[n] + g_2[n]$
- $h_4[n] = g_1[n] - g_2[n]$
- $h_5[n] = g_1[n] \times g_1[n]$
- $h_6[n] = g_1[n] \times g_2[n]$
- $h_7[n] = (g_1 * g_1)[n]$
- $h_8[n] = (g_2 * g_2)[n]$

to the plot above that shows the magnitude of the corresponding frequency response.

Hint: Two unit-sample responses may match to the same frequency-response magnitude plot. The difference between the corresponding frequency responses would appear in the phase plot.