The Fast Fourier Transform (FFT) is a method to compute the Discrete Fourier Transform (DFT) efficiently.

The DFT is defined as:

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

- For each $k \rightarrow N$ adds.
- $N$ values of $k \rightarrow N^2$ adds.
- Direct way of computing DFT. $O(N^2)$
Decimation in time

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

\[ n = 2r \quad r: 0 \rightarrow \frac{N}{2} - 1 \]

\[ n = 2r + 1 \quad r: 0 \rightarrow \frac{N}{2} - 1 \]

\[ X(k) = \sum_{r=0}^{N/2-1} x(2r) e^{-j \frac{2\pi kr}{N}} \]

\[ X(k) = \sum_{r=0}^{N/2-1} g(r) e^{-j \frac{2\pi kr}{N/2}} \]

\[ X(k) = \sum_{r=0}^{N/2-1} h(r) e^{-j \frac{2\pi kr}{N/2}} \]

\[ X(k) = \text{N/2 pt DFT of } g(r) \]

\[ X(k) = \text{N/2 pt DFT of } h(r) \]
\[ G\left(\frac{N}{2}\right) = G(0) \]
\[ G\left(\frac{N}{2} + 1\right) = G(1) \]
\[ G\left(\frac{N}{2} + k\right) \triangleq G(K) \]

\[ H(K) = H\left(\frac{N}{2} + k\right) \]
\[
X(n) = G(n) + e^{j\frac{2\pi n}{N}} H(n)
\]

\[
G(n) = G(3), \quad H(n) = H(3)
\]
Repeat the above process until I get to a 2 pt DF.

\[ x(n) = e^{-\text{link} \cdot \frac{k}{2}} x(0) \]

until

\[ n = 0 \]

then

\[ x(n) = e^{-\frac{\text{link} \cdot k}{2}} x(0) \]

for

\[ k = 0, 1 \]

The link is

\[ \text{link} = \frac{2}{\sqrt{2}} \]

\[ x(0) = x(0) + x(1) \]

\[ x(1) = x(1) - x(0) \]

\[ x(0) = \frac{2 \sqrt{2}}{\sqrt{2}} \]

\[ x(1) = 1 \]
New Notation

\[ W_N^k \triangleq e^{-j2\pi k/N} \]

Twiddle Factor.

Use twiddle factor to redraw basic flow graph for Dec. in Time.

Fig 9.3 of 02S.

q of 02S.
The following butterfly flowgraph is made of butterflies of the form:

\[
\frac{1}{n^2} (2^n + 1) \left( \frac{1}{n^2} + \frac{1}{n^2} \right) - \frac{2}{n^2} \left( \frac{1}{n^2} + \frac{1}{n^2} \right) \]

Observe that:

\[
\frac{1}{n!} = \frac{1}{2^n} \left( \frac{1}{n^2} + \frac{1}{n^2} \right) - \frac{2}{n^2} \left( \frac{1}{n^2} + \frac{1}{n^2} \right)
\]
\[ w_2^2 = \frac{r^2 + 1}{w_2} \]

Diagram:
- 2 input paths
- 2 output paths
- 2 multiplies
- 2 adds
- 1 multiply
- 2 adds
- New improved butterfly
- Show Fig. 9.10
From 9.10, observations:

1. In-place computation. Start with an array of N #s and keep re-reusing it, overwriting it.

2. Embarrassingly parallel.

3. log₂ N stages.

4. Each stage has N/2 butterflies.

5. Each processor does one of the N/2 butterflies in each stage.

Parallelize it to N/2 processes.
6. Each butterfly \( \frac{1}{2} \text{ adds} \) \( \frac{1}{2} \text{ mult.} \)

Total:

Each stage:

\( \frac{N}{2} \) butterflies

Each butterfly \( \frac{1}{2} \text{ mult} \)

\( \frac{1}{2} \text{ add} \)

\[ \log_2 N \text{ stage} \]

\[ \frac{N}{2} \log_2 N \text{ mult.} \]

\[ N \log_2 N \text{ add} \]

\[ \text{Total:} \]

\[ \frac{N}{2} \log_2 N \text{ mult.} \]

\[ N \log_2 N \text{ add} \]
\[ E \quad N = 64 \times 10^6 \]
\[ N^2 = 64 \times 64 \times 10^{12} \]
\[ N \log_2 N = 64 \times 10^6 \times (6 + 20) = 64 \times 26 \times 10^6 \]
\[ \frac{N^2}{N \log_2 N} = \frac{64 \times 64 \times 10^{12}}{64 \times 26 \times 10^6} = 2 \times 10^6 \]

\[ \frac{2000000}{4.6 \times 10^6} = 0.5 \text{ weeks} \]
\[ \frac{500 \text{ hours}}{4 \text{ weeks}} = 1 \text{ sec} \]
Fig 9.10 \to input order is messed up.

Bit Reversing

\[
\begin{array}{cccc}
X(0) & 0 & 000 & 0 \\
X(4) & 1 & 001 & 4 \\
X(12) & 2 & 010 & 2 \\
X(16) & 3 & 011 & 6 \\
X(1) & 4 & 100 & 1 \\
X(15) & 5 & 101 & 5 \\
X(3) & 6 & 110 & 3 \\
X(7) & 7 & 111 & 7 \\
\end{array}
\]