Fast Fourier Transform using Cooley Tukey algorithm

Pseudo-code:

```
For each stage
        For each group of butterfly
                For each butterfly
                        Compute butterfly
                end
        end
end
```

Implementation details:

Consider the DFT algorithm for an integer power of 2, $N = 2^{\nu}$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N{}^{nk} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}; \quad W_N = e^{-j2\pi/N}$$

Create separate sums for even and odd values of *n*:

$$X[k] = \sum_{n\ \ even} x[n] W_N{}^{nk} + \sum_{n\ \ odd} x[n] W_N{}^{nk}$$

Letting $n{=}2r$ ***for n even and*** $n{=}2r{+}1$ ***for n odd, we obtain***

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

Splitting indices in time , we have obtained

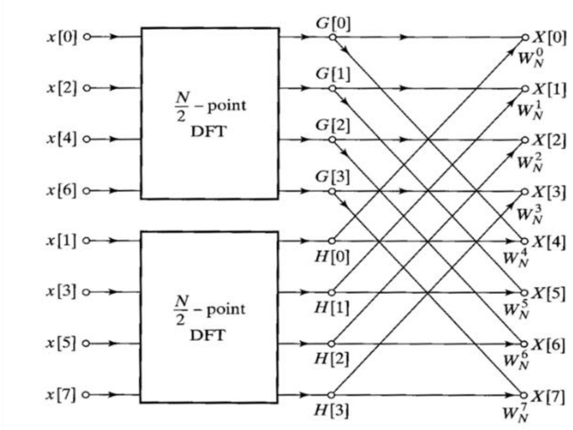
$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] W_N{}^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1] W_N{}^{(2r+1)k}$$

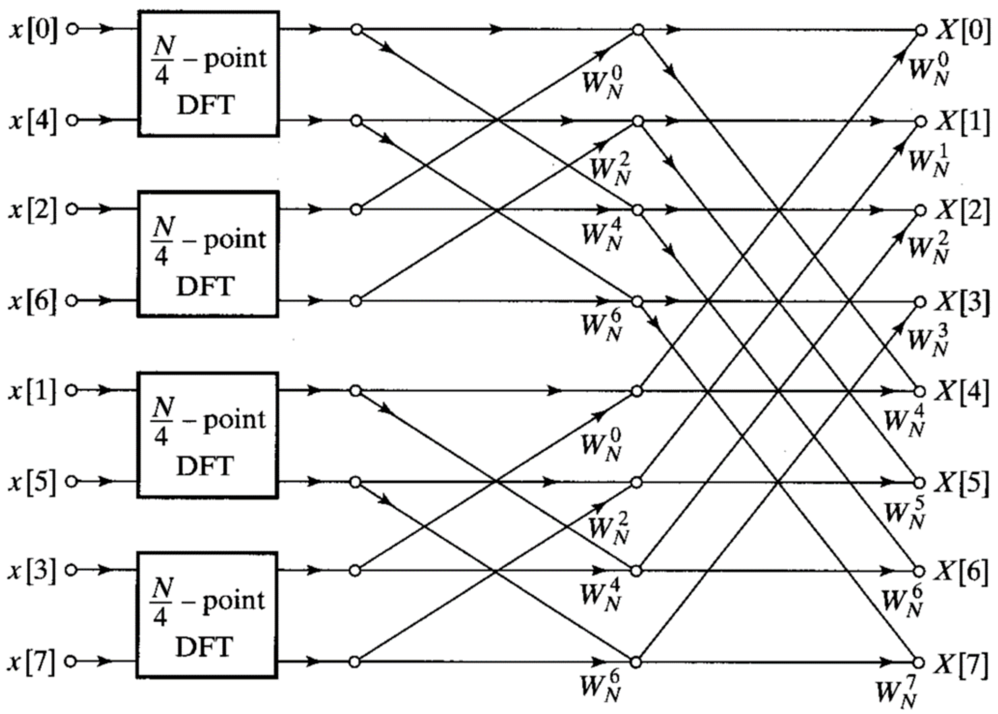
But $W_N^2 = e^{-j2\pi 2/N} = e^{-j2\pi/(N/2)} = W_{N/2}$ **and** $W_N^{2rk} W_N^k = W_N^k W_{N/2}^{rk}$

So $$X[k] = \underbrace{\sum_{n=0}^{(N/2)-1} x[2r] W_{N/2}^{rk}}_{N/2\text{-point DFT of } x[2r]} + W_N^k \underbrace{\sum_{n=0}^{(N/2)-1} x[2r+1] W_{N/2}^{rk}}_{N/2\text{-point DFT of } x[2r+1]}$$

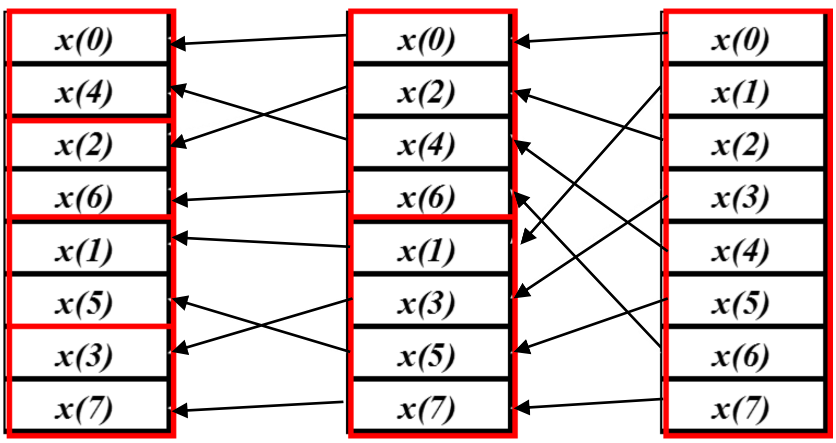
Recall that the DFT is now of the form $X[k] = G[k] + W_N^k H[k]$

The DFT in (partial) flowgraph notation:





index	address
000	000
100	001
010	010
110	011
001	100
101	101
011	110
111	111



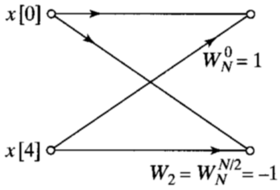
The expression for the 2-point DFT is:

$$X[k] = \sum_{n=0}^{1} x[n] W_2^{nk} = \sum_{n=0}^{1} x[n] e^{-j2\pi nk/2}$$

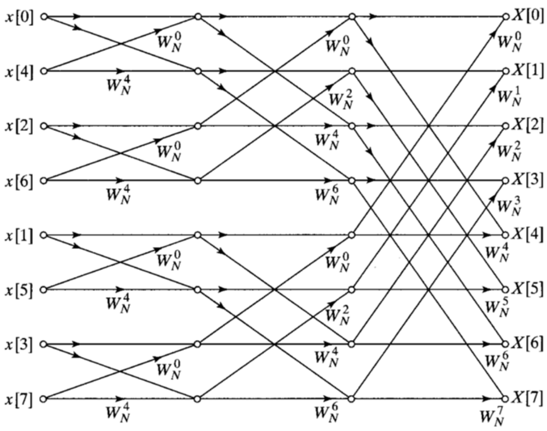
Evaluating for $k = 0,1$ ***we obtain*** $X[0] = x[0] + x[1]$

$$X[1] = x[0] + e^{-j2\pi 1/2} x[1] = x[0] - x[1]$$

which in signal flow graph notation looks like …



This topology is referred as the <u>BASIC BUTTERFLY</u>



- *Let* $N = 2^{\nu}$ where $\nu = \log_2(N)$

($\log_2(N)$ columns)($N/2$ butterflies/column)(2 mults/butterfly)

or $N \log_2(N)$ multiplies