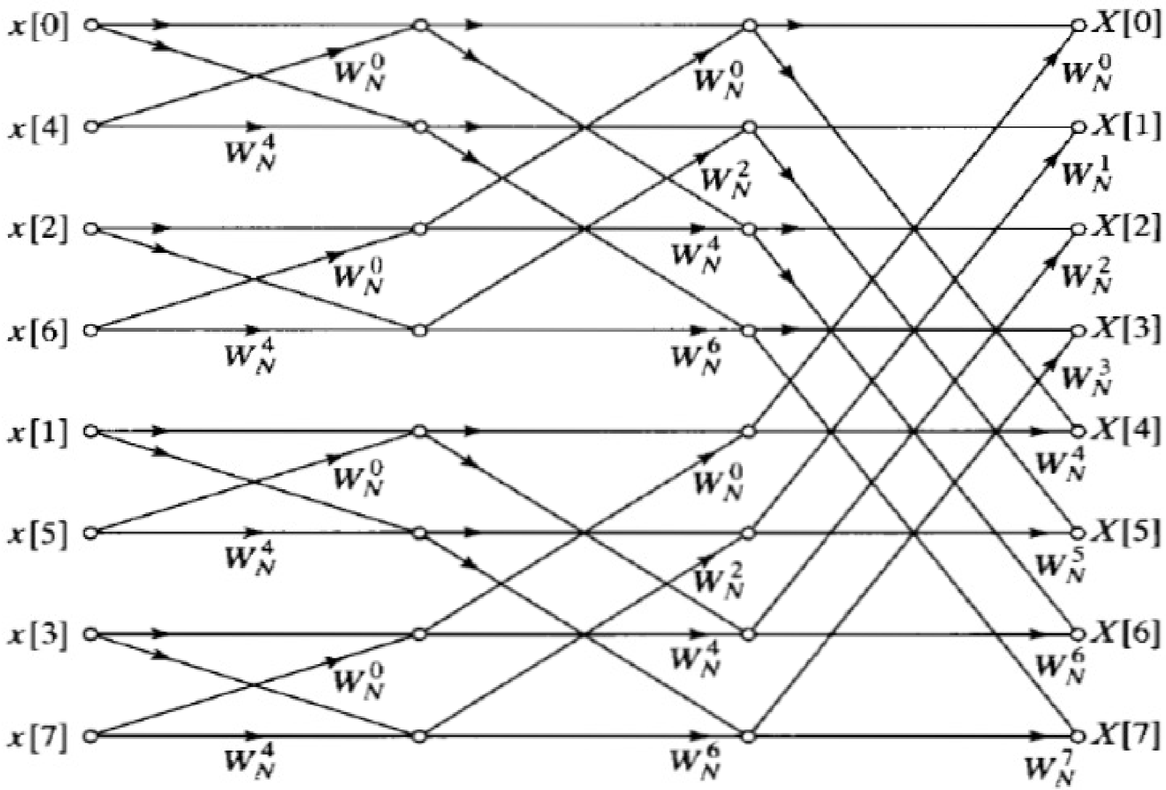
EE 5330 Nov. 5, 2013

Note Title 05-11-2013

Butterfly Diagram for an 8-pt FFT (DIT Algorithm):



From "Discrete-Time Signal Processing" by Oppenheim & Schafer

For $N = 8$, the order in which the i/p appears is,

$$x_0 \quad x_2 \quad x_4 \quad x_6 \quad x_1 \quad x_3 \quad x_5 \quad x_7$$

$$\downarrow$$

$$g_0 \quad g_1 \quad g_2 \quad g_3 \quad h_0 \quad h_1 \quad h_2 \quad h_3$$

$$\downarrow$$

$$g_0 \quad g_2 \quad g_1 \quad g_3 \quad h_0 \quad h_2 \quad h_1 \quad h_3$$

$$\downarrow$$

$$x_0 \quad x_4 \quad x_2 \quad x_6 \quad x_1 \quad x_5 \quad x_3 \quad x_7$$

This order is nothing but the one obtained by (i) representing the index in binary form, and (ii) bit reversing the representation.

index :	0	1	2	3	4	5	6	7
binary Form :	000	001	010	011	100	101	110	111
bit reversal :	000	100	010	110	001	101	011	111
bit reversed index :	0	4	2	6	1	5	3	7

Note that the final stage consists of 2-point DFTs.

If $\{p_0, p_1\}$ is the two point sequence, then,

$$P_0 = p_0 + p_1$$

$$P_1 = p_0 + p_1 e^{-j\pi} = p_0 - p_1$$

Similar to the Decimation in Time (DIT) algorithm, there exists the Decimation in Frequency (DIF) algorithm, wherein computational savings are obtained by dividing the sequence into its first and second halves successively (rather than into its odd and even indices).

We get the same computational savings, but the X_k now appear in bit reversed order.

For N that is not a power of 2, FFT algorithms exist by factoring N into its prime factors (resulting in the so-called Prime Factor Algorithm).

When $N = 2^\gamma$, such FFT algorithms are called radix 2 algorithms.

Efficient algorithms exist even when N is prime!

Since the structure of the Inverse DFT (IDFT) is fundamentally similar to that of the DFT, FFT algorithms can be applied for the efficient computation of the IDFT also with only trivial modifications.

Because of the existence of the FFT class of algorithms, for $N > 60$ (roughly), it is more efficient to realize convolution of two sequences by multiplying their respective transforms and computing the inverse transform of the product.