Efficient Computation of the DFT: FFT Algorithms

Hunter Mills

July 2, 2023

These notes will only cover sections 8.1.1 and 8.1.3.

1 Efficient Computation of the DFT: FFT Algorithms

The computational problem for the DFT is to compute the sequence $K(k)$ of N complex numbers given $x(n)$. For each value of k the DFT involves N complex multiplies and $N-1$ complex additions which gives the algorithms $O(N^2)$. The two properties that are levarged are the symmetry property and the periodicity property.

$$\text{Symmetry}: W_N^{k+N/2} = -W_N^k \tag{1}$$

$$\text{Periodicity}: W_N^{k+N} = W_N^k \tag{2}$$

2 Radix-2 FFT Algorithms

These algorithms are derived in the book pg 520. I will include figures and important equations here. The first figure shows when you only split the DFT sum into two parts but it can continually be split until the resulting sequences are reduced to one point sequences.

2.1 Decimation in Time

$$f_1(n) = x(2n) \tag{3}$$

$$f_2(n) = x(2n+1) \tag{4}$$

Figure 1 shows the first step in the Radix 2 algorithm when you split the DFT into two terms. This is the decimation in time version.

Figure 2 shows the an 8pt FFT using the radix-2 algorithm.

The butterfly unit is the main part of the FFT and is shown in figure 3. It includes a complex multiply and sum complex additions.

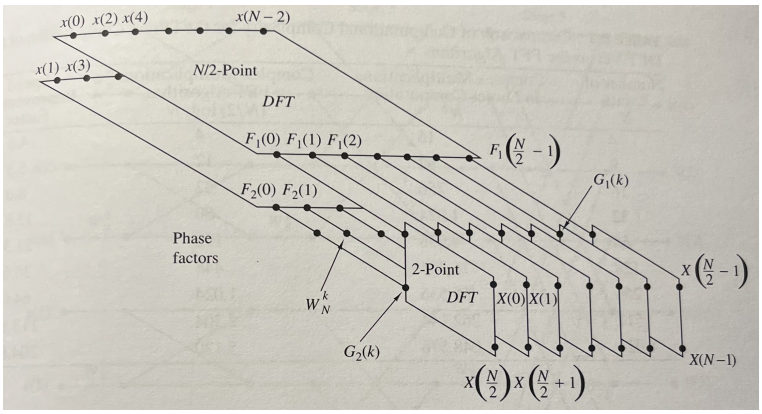

Figure 1: Two summation Radix 2

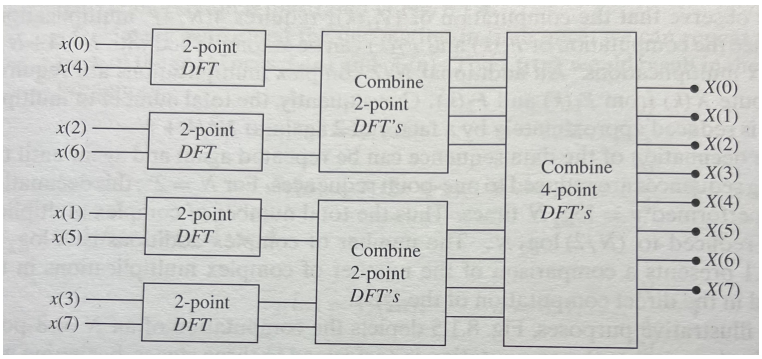

Figure 2: Radix 2 Algorithm

Figure 4 is the full algorithm for a 8-point Radix-2 algorithm with all the butterfly units.

Figure 5 shows how you can reverse the bits and calculate the FFT in an algorithm.

2.2 Decimation in Frequency

In this section, first find $X(k)$ then decimate that in frequency the same way as it was done in time

$$X(2k) \text{ and } X(2k+1) \tag{5}$$

Figure 6 e shows the Radix-2 Decimation in frequency algorithm.

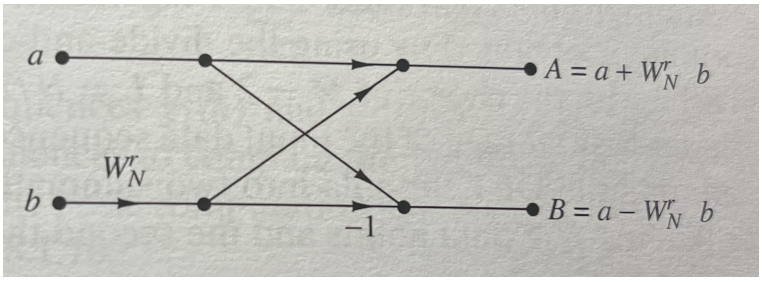

Figure 3: Butterfly Unit

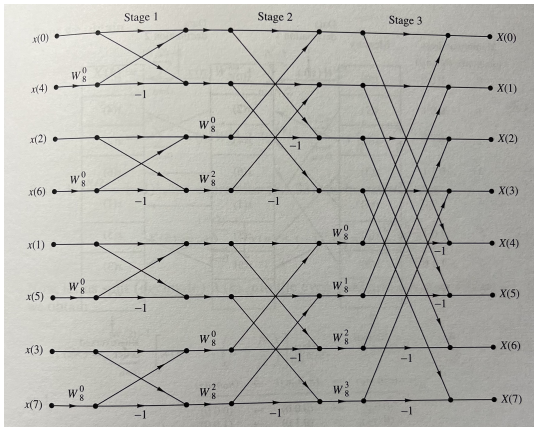

Figure 4: Full Radix 2 Algorithm

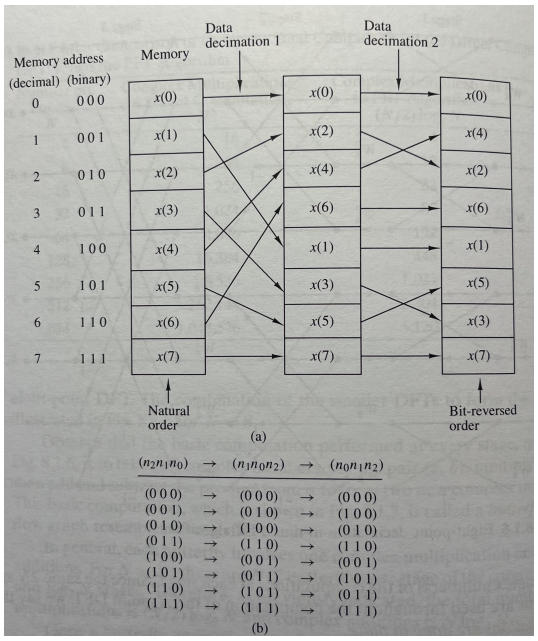

Figure 5: DIT Bitflip Algorithm

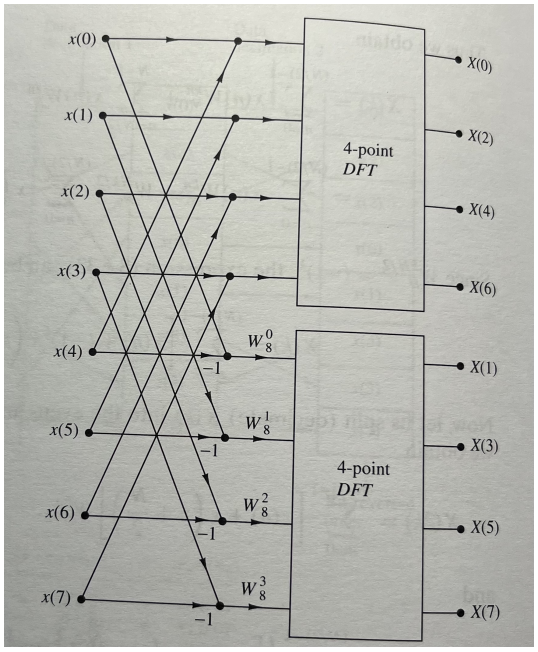

Figure 6: Radix-2 FFT DIF

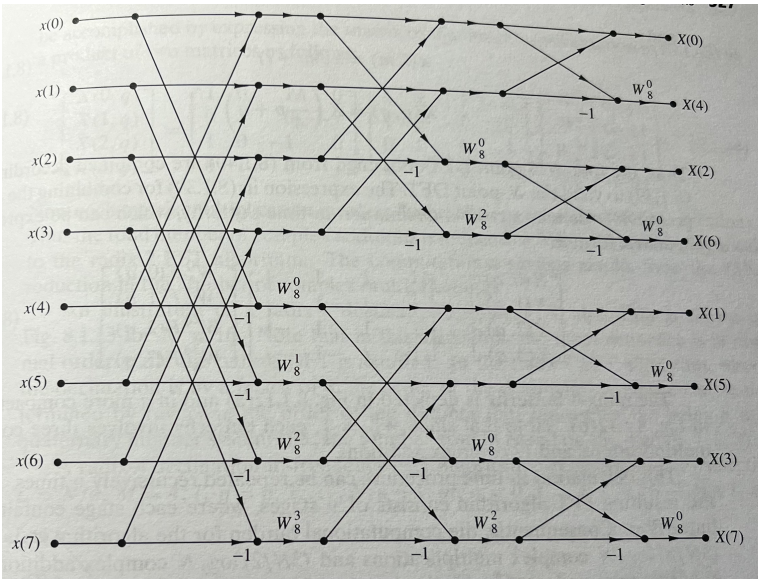

Figure 7: Full Radix-2 DIF Algorithm