DFT. The frst one is a DFT of the even-numbered elements, and the second of the odd-numbered elements. The frst DFT is combined with the second multiplied by the complex exponential



The half-length transforms are each evaluated at frequency indices *k* = 0, ..., *N* - 1. Normally, the number of frequency indices in a DFT calculation range between zero and the transform length minus one. The **computational advantage** of the FFT comes from recognizing the periodic nature of the discrete Fourier transform. The FFT simply reuses the computations made in the half-length transforms and combines them through additions and the multiplication by e, which is not periodic over



[Figure 5.12](#_bookmark387) (Length-8 DFT decomposition) illustrates this decomposition. As it stands, we now compute



transforms



multiply one of them by the complex exponential (complexity *O(N)*), and add the results (complexity *O(N)*). At this point, the total complexity is still dominated by the half-length DFT calculations, but the proportionality coefcient has been reduced.

Now for the fun. Because *N=2L*, each of the half-length transforms can be reduced to two quarter-length transforms, each of these to two eighth-length ones, etc. This decomposition continues until we are left with length-2 transforms. This transform is quite simple, involving only additions. Thus, the frst stage of the FFT has



length-2 transforms (see the bottom part of [Figure 5.12](#_bookmark387) (Length-8 DFT decomposition)). Pairs of these transforms are combined by adding one to the other multiplied by a complex exponential. Each pair requires 4 additions and 2 multiplications, giving a total number of computations equaling



This number of computations does not change from stage to stage. Because the number of stages, the number of times the length can be divided by two, equals log2*N*, the number of arithmetic operations equals



which makes the complexity of the FFT O(*N*log2*N*).