Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 5
Spectrum Analysis -- DFT

Verigy Japan
September 2008

Preface to the Series

ADC and DAC are the most typical mixed signal devices. In mixed signal testing, analog stimulus signal is generated by an arbitrary waveform generator (AWG) which employs a D/A converter inside, and analog signal is measured by a digitizer or a sampler which employs an A/D converter inside. The stimulus signal is created with mathematical method, and the measured signal is processed with mathematical method, extracting various parameters. It is based on digital signal processing (DSP) so that our test methodologies are often called DSP-based testing.

Test/application engineers in the mixed signal field should have thorough knowledge about DSP-based testing. FFT (Fast Fourier Transform) is the most powerful tool here. This corner will deliver a series of fundamental knowledge of DSP-based testing, especially FFT and its related topics. It will help test/application engineers comprehend what the DSP-based testing is and assorted techniques.

Editor's Note

For other articles in this series, please visit the Verigy web site at www.verigy.com/go/gosemi.

1. Introduction

In the previous go/semi articles, there are several figures of frequency spectrum shown as Figure 1 without any explanation.

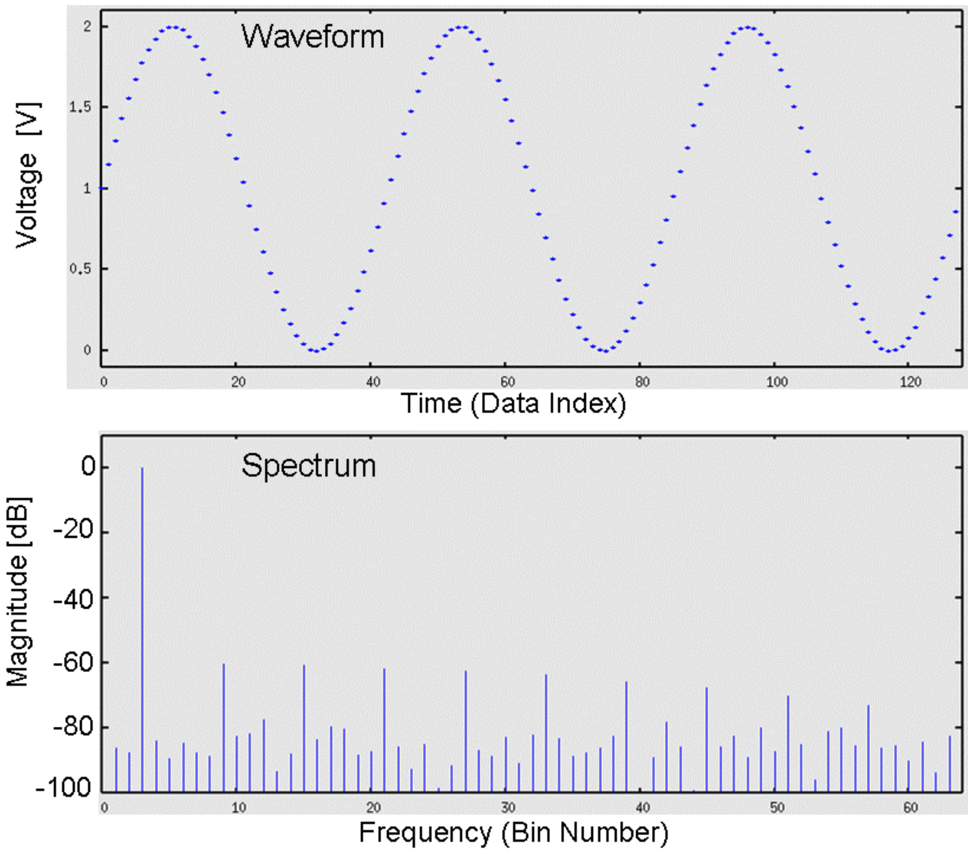

Figure 1. Sampled Signal Waveform and Spectrum

Figure 1 shows a sampled discrete signal waveform and its frequency spectrum, which is created by applying DFT(FFT). In ATE field, 99% of signal analyses mean spectrum analyses. For application engineers who are going to analyze a test signal in the frequency domain, the most powerful and useful tool is DFT or rather FFT. In IC design area, DFT sometimes stands for design for test(-ability). Here in this series of go/semi articles, DFT is the short-hand for discrete Fourier transform. The turbo version of DFT is called fast Fourier transform (FFT). In this issue DFT is discussed in general, and in the next issue practical FFT will be discussed.

2. Discrete Fourier Transform (DFT)

The signal we take care of in tests is a real number signal. A sinusoidal signal can often be expressed Acos(ωt+φ) which is a real number, but it can be expressed with the summation of complex numbers as follows.

$$A\cos(\omega t+\phi)=\frac{A}{2}\cdot\left(e^{j(\omega t+\phi)}+e^{-j(\omega t+\phi)}\right) \qquad (1)$$

Figure 2 explains the meaning of Equation (1). Figure 2 means that the signal Acos(ωt+φ) can be expressed with the summation of a pair of rotating vectors – one rotates to the positive direction, and the other rotates to the negative direction. Each vector rotates with the angular velocity of (+ω) and (-ω) respectively. When ω=2πf, the frequency of the pair of vectors is (+f) and (−f) respectively. The pair of vectors is represented with complex numbers of Ae+j(ωt+φ) and Ae-j(ωt+φ). They are always complex conjugates with each other. Consequently a real number signal can always be expressed with the complex conjugate pair as Equation (1).

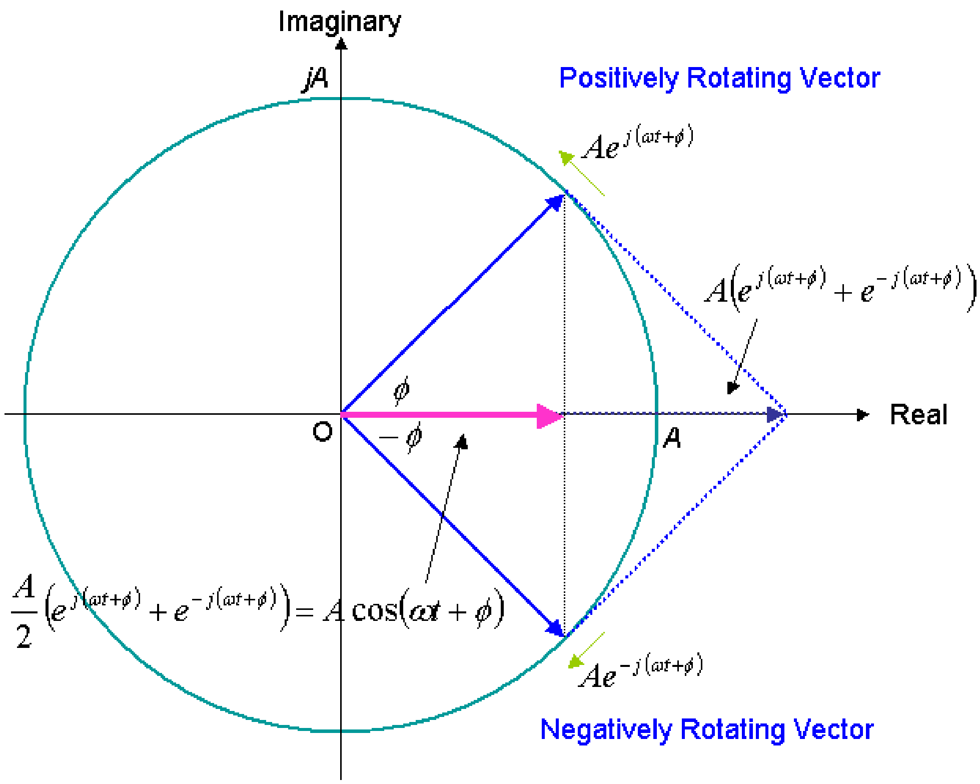

Figure 2. Real Number Signal Vector

Since the test signal Acos(ωt+φ) is expressed with a pair of conjugate vectors, the spectrum of the conjugate pair can be shown as Figure 3.

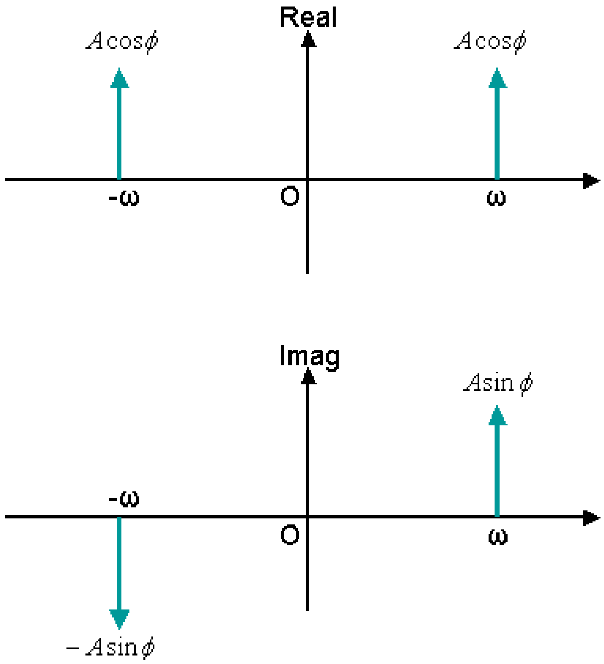

Figure 3. Frequency Spectrum (Complex Conjugate)

The point is that a real signal is always expressed with complex conjugate spectrum. The real parts are equal, and the imaginary parts have the same length but the opposite sign.

Frequency spectrum means how much each frequency component contains in the test signal. In other words the frequency analysis is equivalent to "correlation" to frequency components. Figure 4 shows the primitive 0 to 3-cycle cosine and sine waveforms contained in the unit test period (UTP). When the number of points in the UTP is N, the UTP can contain the sinusoidal waveforms from 1 up to (N/2–1) cycles. Considering the positive frequency and the negative frequency, the UTP can contain from –(N/2-1) to (N/2-1) cycles components including 0 cycle or DC component, that goes total N components. Each component is actually pairs of cosine and sine waveforms.

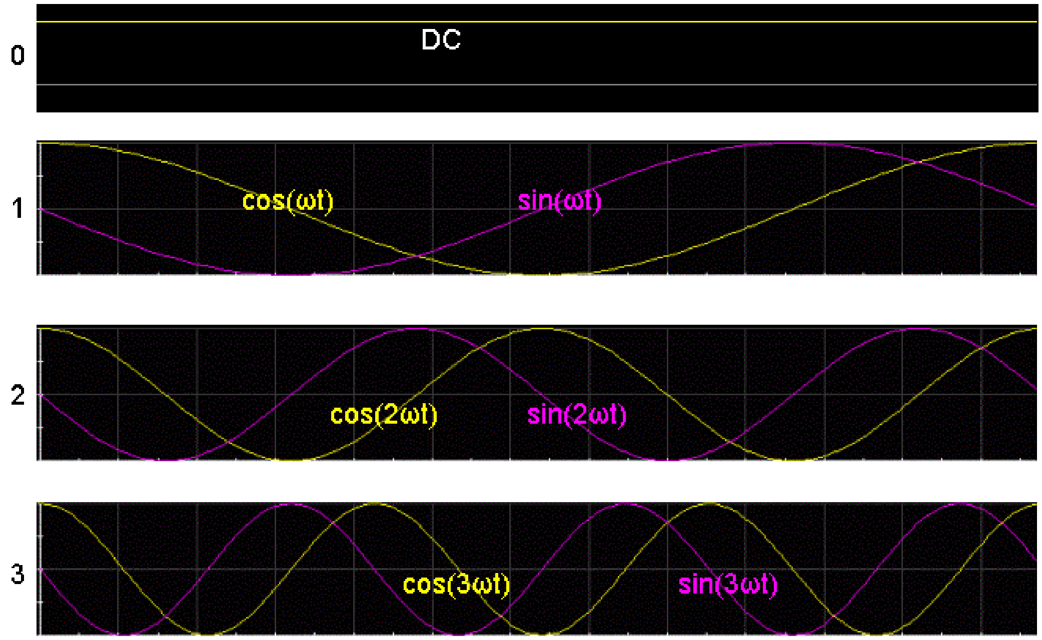

Figure 4. Cosine & Sine Waveforms In UTP

Look at Figure 5, which describes about the correlation data processing of the 3-cycle component in the unit test period (UTP). The waveform (a) shows the test signal waveform. The first operation is the simple summation and averaging of all values. The derived mean value is the DC component of the signal, which corresponds the bin #0 spectrum. The 3-cycle cosine and sine waveforms in (b) are multiplied to the test signal waveform of (a) respectively, generating waveforms in (c). The mean values of the waveforms in (c) actually represent the spectrum of bin #3. The value derived by cosine multiplication becomes the real part of the spectrum, and the value derived by sine multiplication becomes the imaginary part of the spectrum.

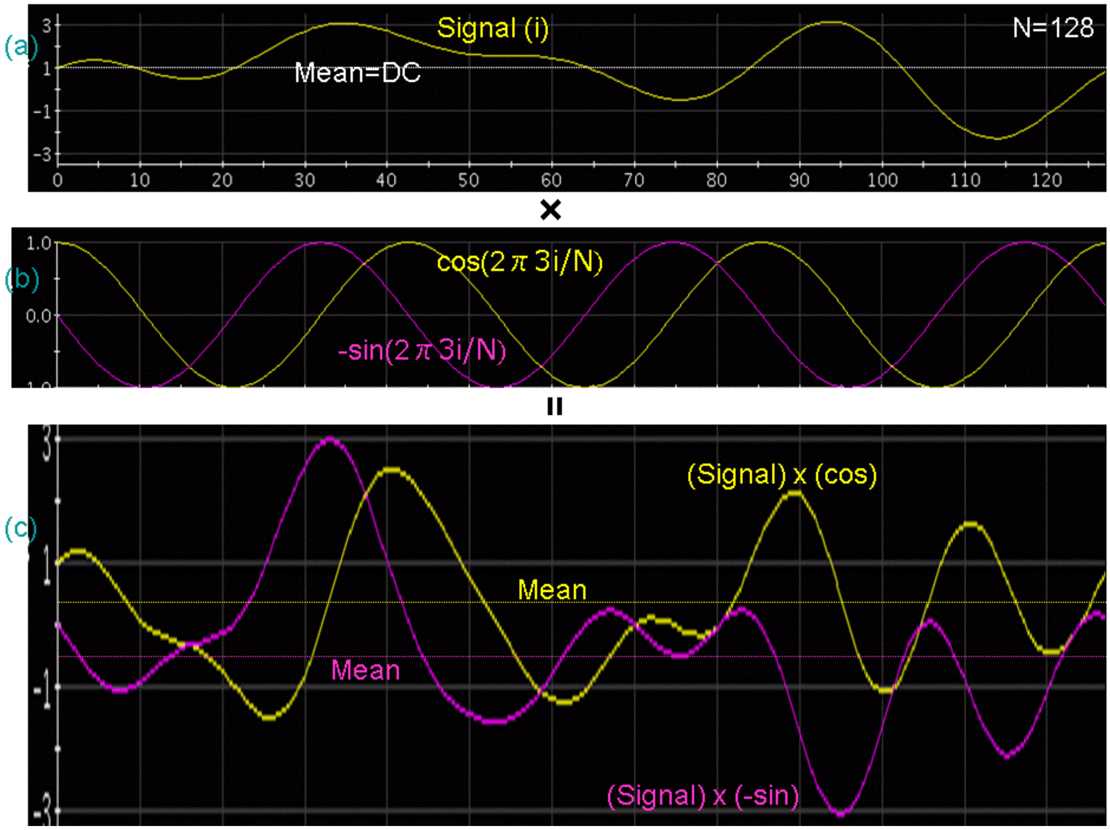

Figure 5. 3-Cycle Cosine And Sine Correlation

Figure 5 shows the data processing of deriving the bin #3 spectrum. This is the way the entire frequency components are derived as Figure 6 for the test signal shown in Figure 5 (a). It shows complex conjugate symmetrical spectrum as discussed previously.

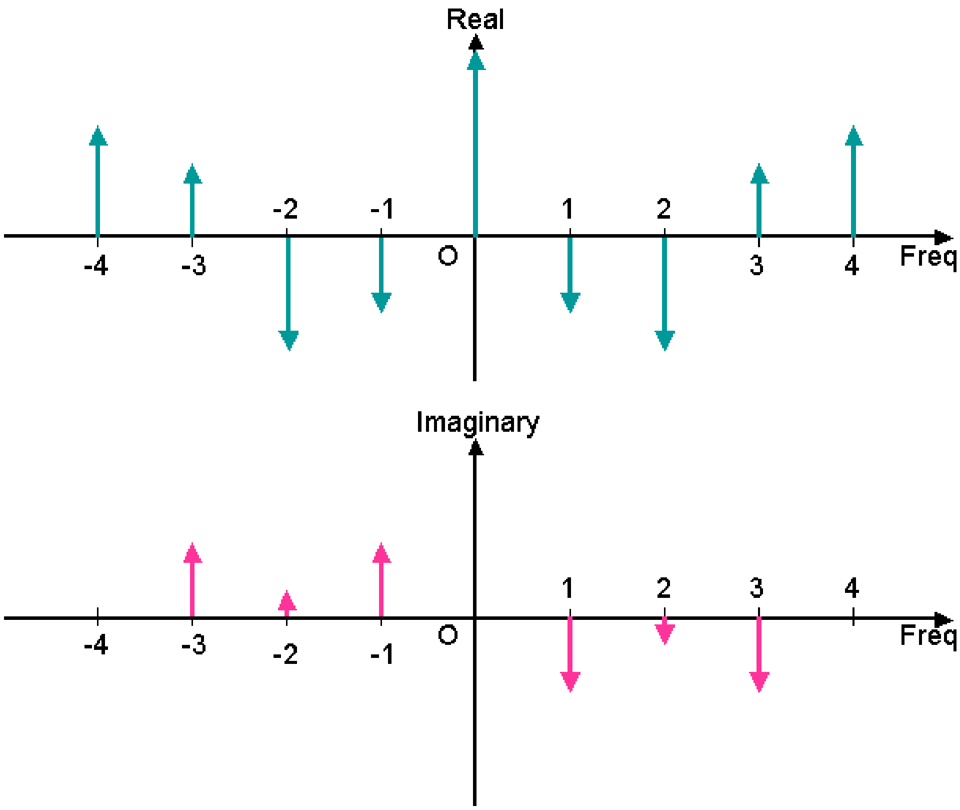

Figure 6. Frequency Analysis by Correlation

The correlation processing described above is the essence of DFT in short. Consequently very basic DFT can be described as follows.

$$X_k = \frac{1}{N}\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} x_i e^{-j\frac{2\pi k i}{N}} = \frac{1}{N}\left(\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} x_i \cos\frac{2\pi k i}{N} - j\cdot\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} x_i \sin\frac{2\pi k i}{N}\right) \qquad (2)$$

where k=-N/2 to (N/2-1). {xi} is the series of sampled discrete data in the time domain. N is the number of the elements of {xi}. Xk is the mean values of multiplied cosine and sine waveforms to the sampled data series. {Xk} is the series of the mean values, and represents the frequency spectrum of the data.

Figure 7 shows a very noisy and distorted signal that is sampled with N points of data.

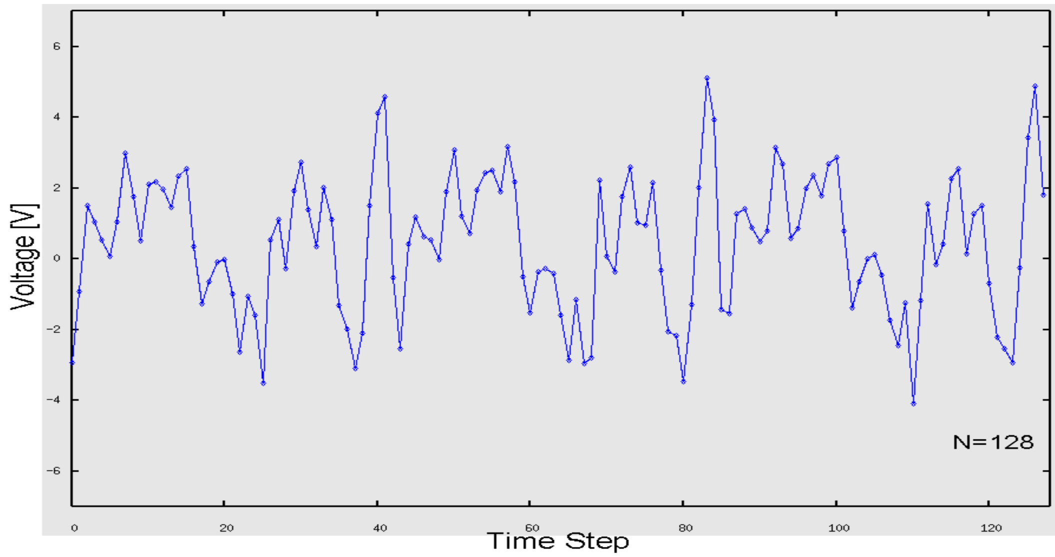

Figure 7. Noisy and Distorted Signal Waveform

This signal is analyzed based on Equation (2), and its frequency spectrum is derived as Figure 8, which shows complex conjugate spectral lines.

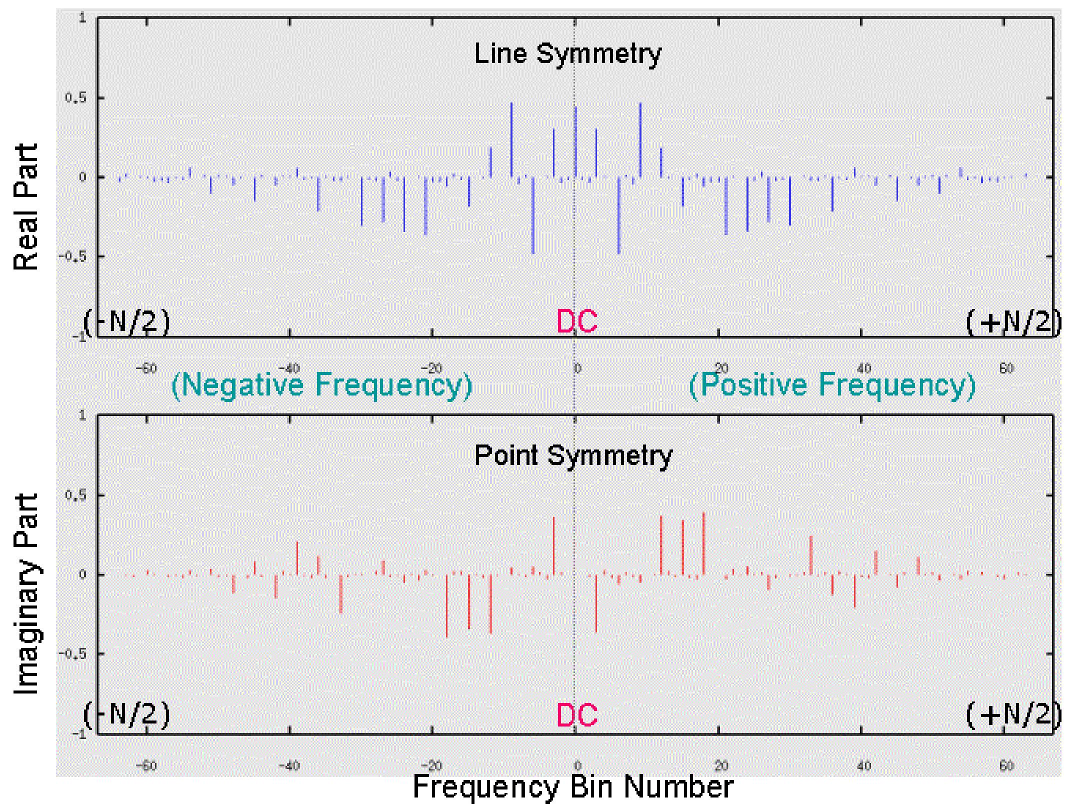

Figure 8. Frequency Spectrum

The definition of the DFT is as simple as Equation (2) that you can create your own DFT program by yourself as follows.

```
INT               i,k,N;
DOUBLE           dP,dRe,dIm;
ARRAY_D          dWave;
ARRAY_COMPLEX   CSp;

dWave.resize(N);// Input Waveform Data
CSp.resize(N);  // Output Spectrum Data (Complex Number)
dP=2.0*M_PI/N;
for (k=0;k<N;k++) {
     dRe=0.0; dIm=0.0;
     for (i=0;i<N;i++) {
           dQ=i*k*dP;
           dRe=dRe+dWave[i]*cos(dQ);
           dIm=dIm-dWave[i]*sin(dQ);
       }
       dRe=dRe/N;  dIm=dIm/N;
       CSp[k].real()=dRe;  CSp[k].imag()=dIm;
}
```

List 1. Program Code of DFT

In List 1, index k goes with 0 to (N-1) instead of N/2 to (N/2-1) for programming simplicity. The sinusoidal waveforms from −π to +π are equivalent to the waveforms from 0 to 2π. Applying this code to the signal waveform, the frequency spectrum is derived as Figure 9. The difference between Figures 8 and 9 is that the left and the right half areas are exchanged. The area from (N/2) to (N-1) in Figure 9 corresponds to the negative frequency spectrum in Figure 8 actually.

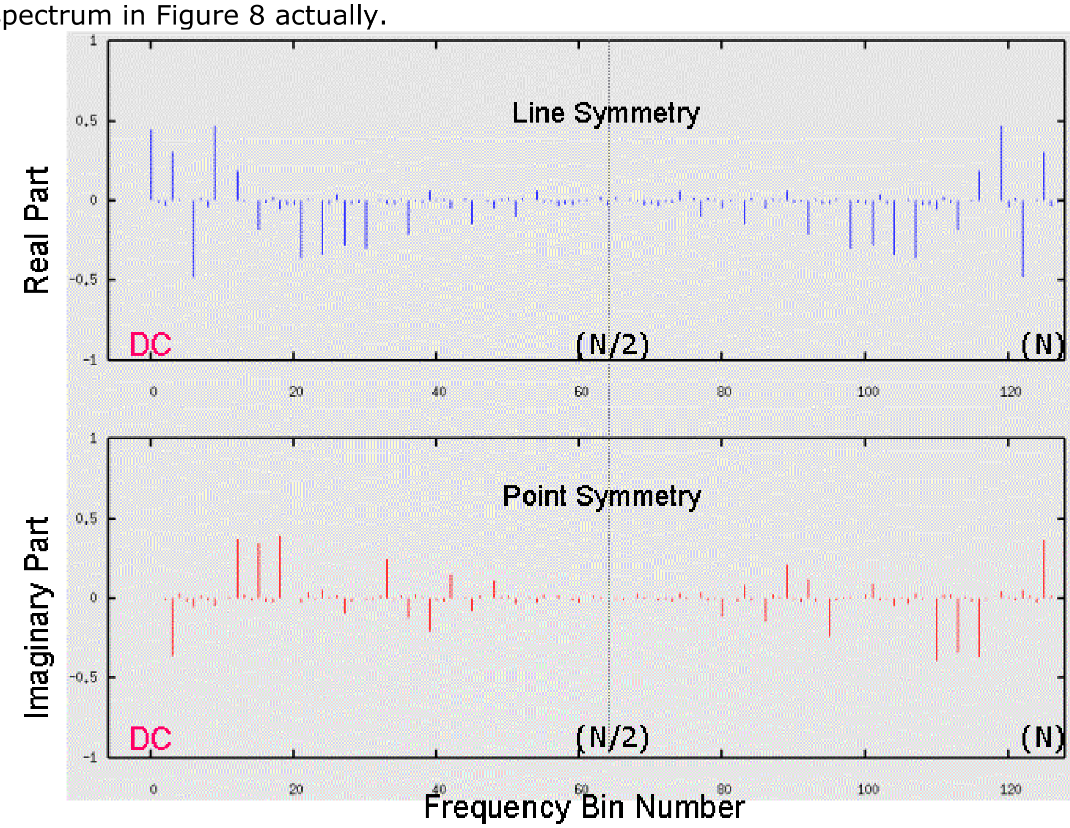

Figure 9.   Spectrum Calculated By List 1

Since the frequency domain spectrum Xk is complex conjugate with XN-k , the right half can always be derived by the left half. Therefore the right half information is redundant. The left half from bin#0 to bin#(N/2-1) is good enough for spectrum analyses. So the source code of List 1 can be modified as follows.

```
INT              i,k,N,Nsp;
DOUBLE          dP,dRe,dIm;
ARRAY_D         dWave;
ARRAY_D         dSp,dPha;

dWave.resize(N);// Input Waveform Data

  Nsp=N/2;            // Half Size of Data
dSp.resize(Nsp);  // Output Spectrum Data (Half Page)
dPha.resize(Nsp); // Phase of the signal

dP=2.0*M_PI/N;
for (k=0;k<Nsp;k++) {
     dRe=0.0; dIm=0.0;
     for (i=0;i<N;i++) {
            dQ=i*k*dP;
            dRe=dRe+dWave[i]*cos(dQ);
            dIm=dIm-dWave[i]*sin(dQ);
       }
       dRe=dRe/N;  dIm=dIm/N;               // Complex
       dSp[k]=2.0*sqrt(dRe*dRe+dIm*dIm); // Magnitude
       dPha[k]=atan2(dIm,dRe);             // Phase
}
```

List 2. Program Code of DFT (Half Page Mode)

Each spectrum magnitude is doubled in List 2 to match the total power of the spectrum to the total power of List 1. When calculating the spectrum with List 2, the result becomes Figure 10, which shows the magnitude and phase instead of the real and imaginary parts. This expression is actually popular and consistent to the display on spectrum analyzers.

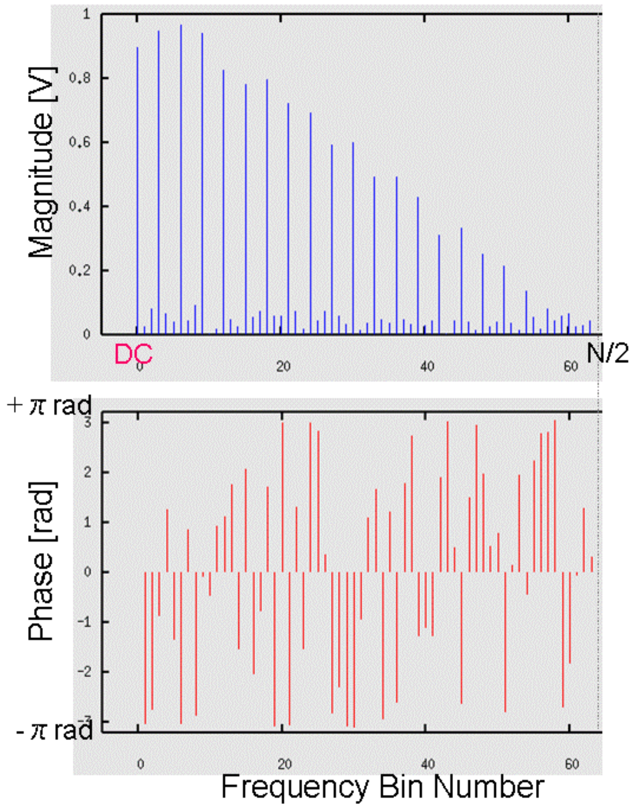

Figure 10. Frequency Spectrum Calculated By List 2

DFT is simple data processing described in Lists 1 and 2, and it can give you entire information of the test signal -- what frequency components are contained in the signal, their magnitude and phase. It tells you about the distortion, noise and spurious in the signal. This is really a useful and powerful tool for application engineers. So application engineers, especially who need to deal with mixed signal applications, must definitely be familiar with the DFT/FFT operations, what they mean, how they work and their restrictions.