Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 49
S-parameter De-embedding

ADVANTEST Corporation
April 2013

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

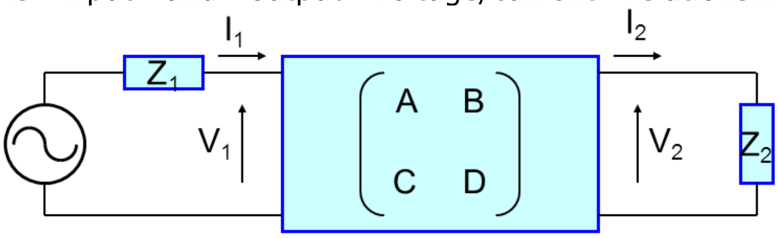
For other articles in this series, please visit the Advantest web site at www.verigy.com/go/gosemi.

Preface

A data rate of Several Gbps is not surprising in highspeed digital signal these days. In such high frequency range, when a DUT output signal goes through the transmission line on the DUT board to the input terminal of the measurement instrument, the signal would easily be attenuated and distorted by the frequency characteristics of the transmission line. Therefore in order to look at the true output signal at the DUT output terminal, the measured signal is often compensated for recovering the degradation by de-embedding the frequency response of the path. The S-parameters of the transmission line is used to perform compensation. The topic of this article is the de-embedding of the network S-parameters.

F-matrix to S-matrix

Two-port network and F-matrix (ABCD-matrix) were discussed in the Lecture Series 40. The input and output voltage/current relationship is described in Figure 1.



$$\begin{cases} V_1 = A \cdot V_2 + B \cdot I_2 \\ I_1 = C \cdot V_2 + D \cdot I_2 \\ V_2 = I_2 \cdot Z_2 \end{cases}$$

$$V_1 = A \cdot V_2 + B \cdot \frac{V_2}{Z_2} = \left(A + \frac{B}{Z_2} \right) \cdot V_2$$

Figure 1: F-matrix and Two-Port Network

In the discussion of de-embedding, V_1 is the true outout of a DUT and V_2 is the measured signal by an instrument. So $V_1=(A+B/Z_2)\cdot V_2$ is actually the equation for de-embedding if you know the parameters of A, B and Z_2. S-parameters are popular in RF network and you can measure S-parameters of the target circuit by using a network analyzer. If the S-parameters of the network is available, the equation should be modified by S-parameters instead of ABCD parameters. Figure 2 describes the relationship of ABCD-parameters and S-parameters. By substituting ABCD-parameters with S-parameters, the equation of the input/output relationship in Figure 1 can be modified as Figure 3 illustrates. This shows that the input V_1 can be estimated by the output V_2 with using s_{21} and s_{11}.

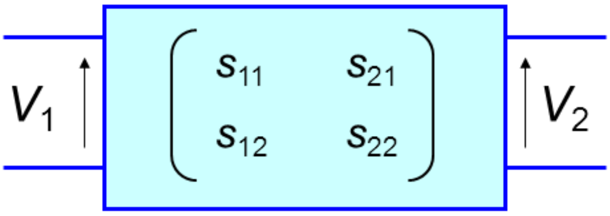
$$A = \frac{(1+s_{11})\cdot(1-s_{22})+s_{21}\cdot s_{12}}{2s_{21}}$$

$$B = \frac{(1+s_{11})\cdot(1+s_{22})-s_{21}\cdot s_{12}}{2s_{21}}\cdot Z_2$$

$$C = \frac{(1-s_{11})\cdot(1-s_{22})-s_{21}\cdot s_{12}}{2s_{21}}\cdot\frac{1}{Z_1}$$

$$D = \frac{(1-s_{11})\cdot(1+s_{22})+s_{21}\cdot s_{12}}{2s_{21}}\cdot\frac{Z_2}{Z_1}$$

Figure 2: **ABCD-parameters by S-parameters**



$$V_1 = \left(A+\frac{B}{Z_2}\right)\cdot V_2 = \left(\frac{(1+s_{11})(1-s_{22})+s_{21}s_{12}}{2s_{21}}+\frac{\frac{(1+s_{11})(1+s_{22})-s_{21}s_{12}}{2s_{21}}\cdot Z_2}{Z_2}\right)\cdot V_2$$

$$V_1 = \frac{1+s_{11}}{s_{21}}\cdot V_2$$

Figure 3: **V1 and V2 Relationship based on S-parameters**

Experimental Result

Figure 4 shows a photo of a strip line used in the experiment. A weaved trace image is printed on the board surface, but the actual zigzag trace of the strip line is burried inside the inner layers of the board. This is a 50-ohm transmission line and 25 cm long. It is intentionally designed zigzag to see how it degrades the waveform. The SMA connector in the lefthand side is the input port J121. Another SMA connector in the righthand side is the output port J120. One of the PS9G pin driver in the V93000 generates a PRBS bit stream, which is fed into the input port (J121). The output signal at the output port (J120) is monitored by another PS9G pin receiver. This configuration is illustrated in Figure 5. Ro is 50 ohms here.

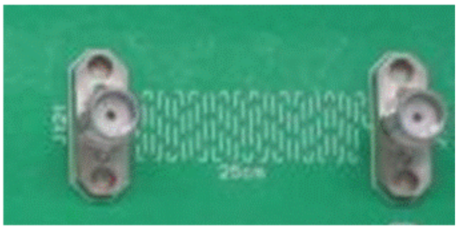


Figure 4: Zigzag Strip Line

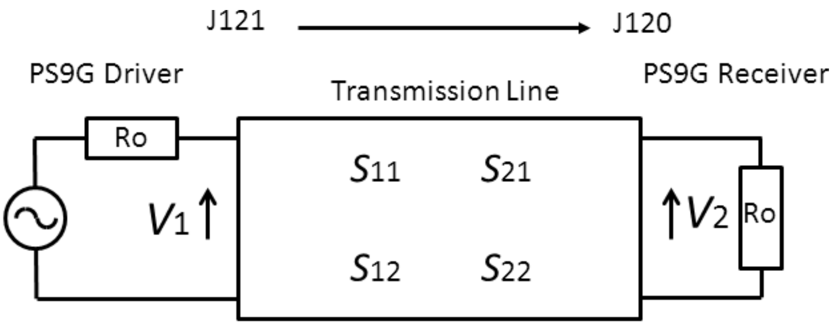


Figure 5: Test Configuration

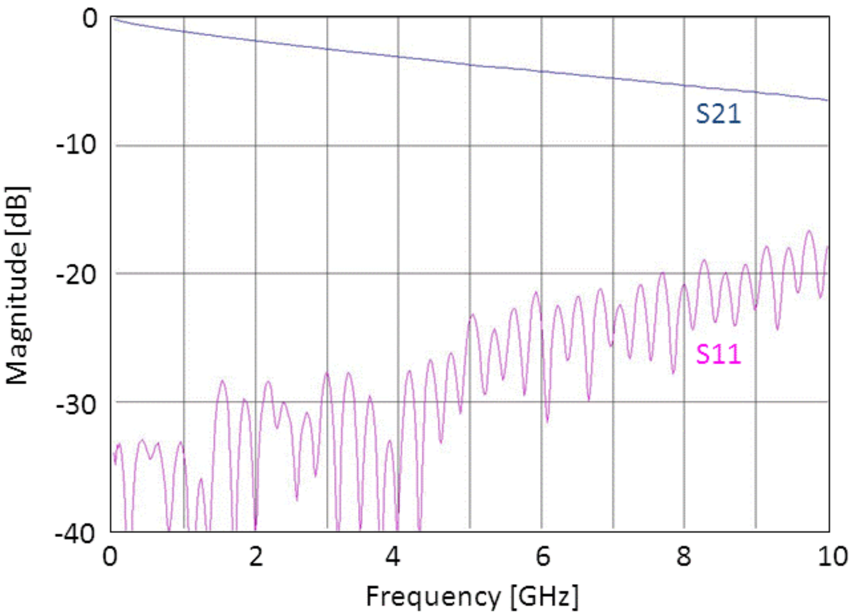


Figure 6: Measured S-parameters (Magnitude Only)

The S-parameters of the strip line is measured by using a network analyzer from 20 MHz to 10 GHz every 20 MHz. Its |S21| and |S11| is plotted in Figure 6. If the network is ideal, S21=1.0 (0 dB) and S11=0.0 (-∞ dB). However, in reality, S21 is gradually decreasing and S11 is gradually increasing.

The test signal is 127-bit PRBS7 data stream running at the rate of 5 Gbps, whose waveform is precisely monitored by using DIGITAL_WAVEFORM() function. It is shown in Figure 7. The waveform is constructed by 5080 points of data at the step of 5 ps, so it is equivalent to the rate of 200 Gsps. As you can see, the narrow pulses have much less amplitude than thick pulses. The waveform distortion is mainly caused by the gradually attenuating frequency characteristics of S21.

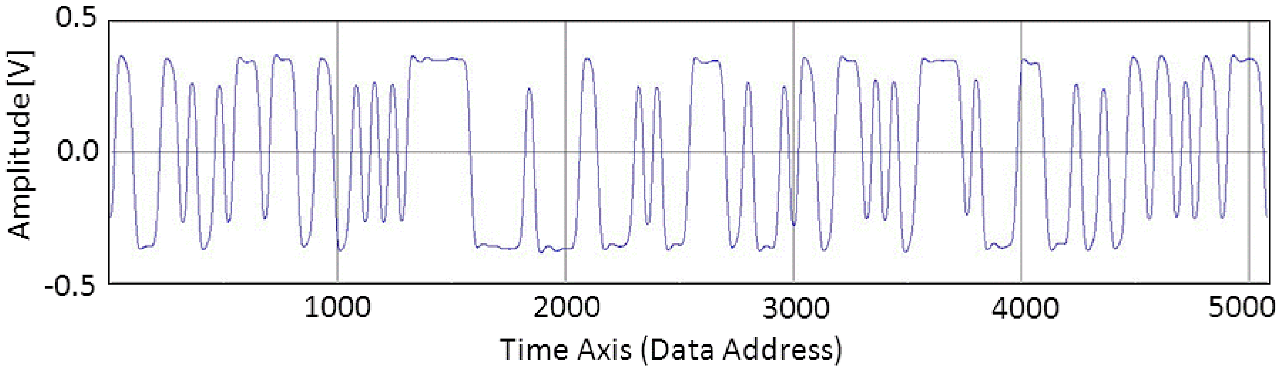


Figure 7: Captured Waveform (V2) 127-bit PRBS7 @ 5 Gbps

The frequency spectrum of the waveform is illustrated in Figure 8. It is constructed by the frequency resolution of 39.37 MHz (=200 GHz/5080) which does not exactly meet the resolution that the S-parameter is measured.

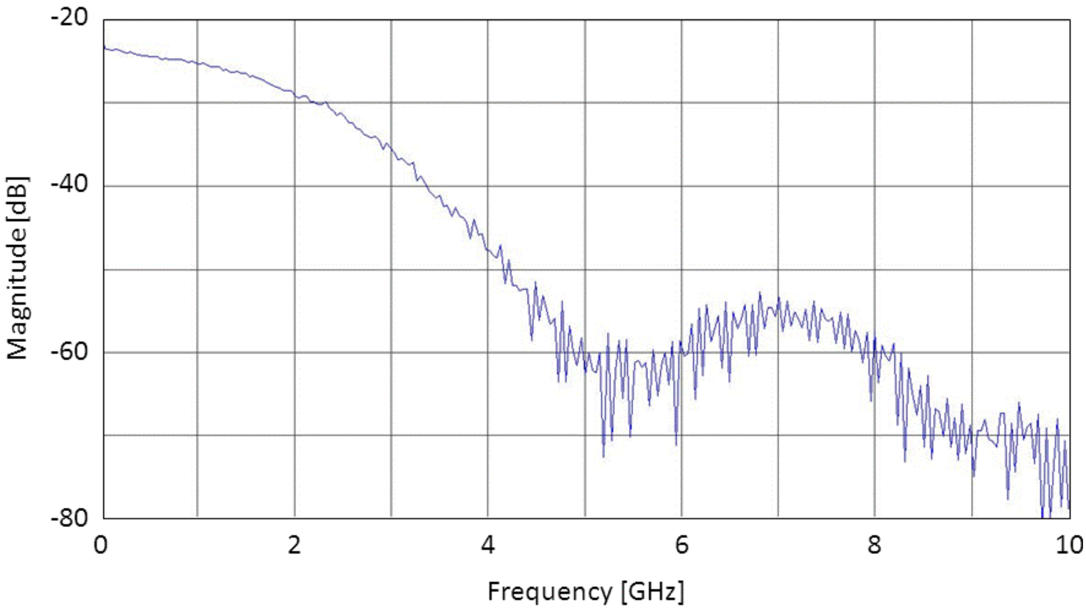


Figure 8: Frequency Spectrum of PRBS7

The complex number of S-parameters in Figure 6 is displayed in the polar coordinates in Figure 9. The blue dots shows the measured points. The red dots shows the interpolated points based on the Lagrange interpolation method introduced in the previous Lecture Sieries 48. Each one of the red points is estimated by using the third order polynomial curve with looking at two points before and two points after the target frequency so the interpolated location can accurately estimate the correct location.

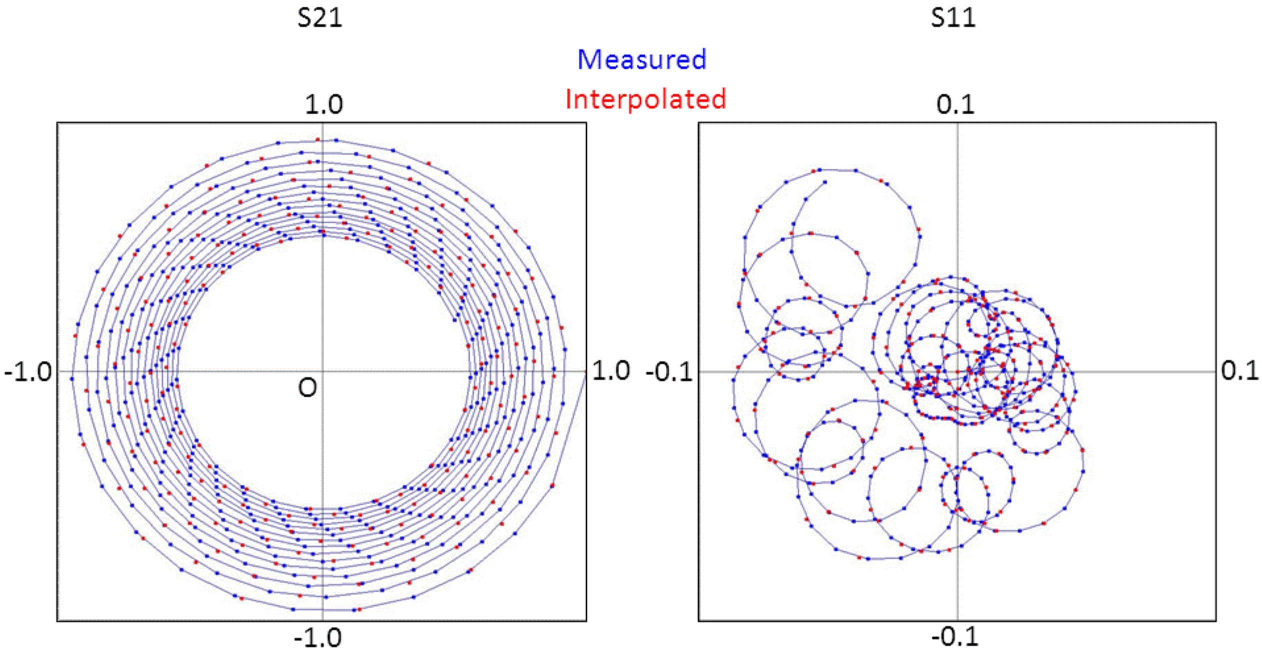


Figure 9: Measured S-parameters (Blue Dots) and Interpolated Values (Red Dots)

By involving the estimated S-parameters into the equation of $V_1=\{(1+S_{11})/S_{21}\}*V_2$, the de-embedded PRBS waveform is derived as Figure 10 shows. The narrow pulses certainly become longer than before and the leading edges are emphasized.

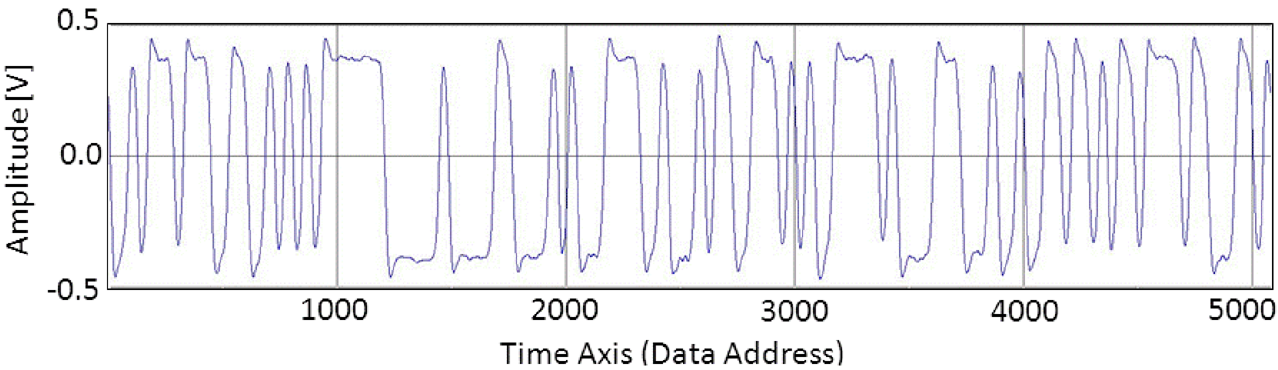


Figure 10: De-embedded Waveform (V1)

The frequency spectra shown in Figure 11 really display how the signal is improved from the spectrum's point of view. As you can notice, the red spectrum is intentionally cut out at 8 GHz. De-embedding processing is apt to boost high frequency noise more than necessary, resulting the de-embedded waveform would contain extreme spikes. So limiting the frequency band at a reasonalbe range is a very important point in this processing.

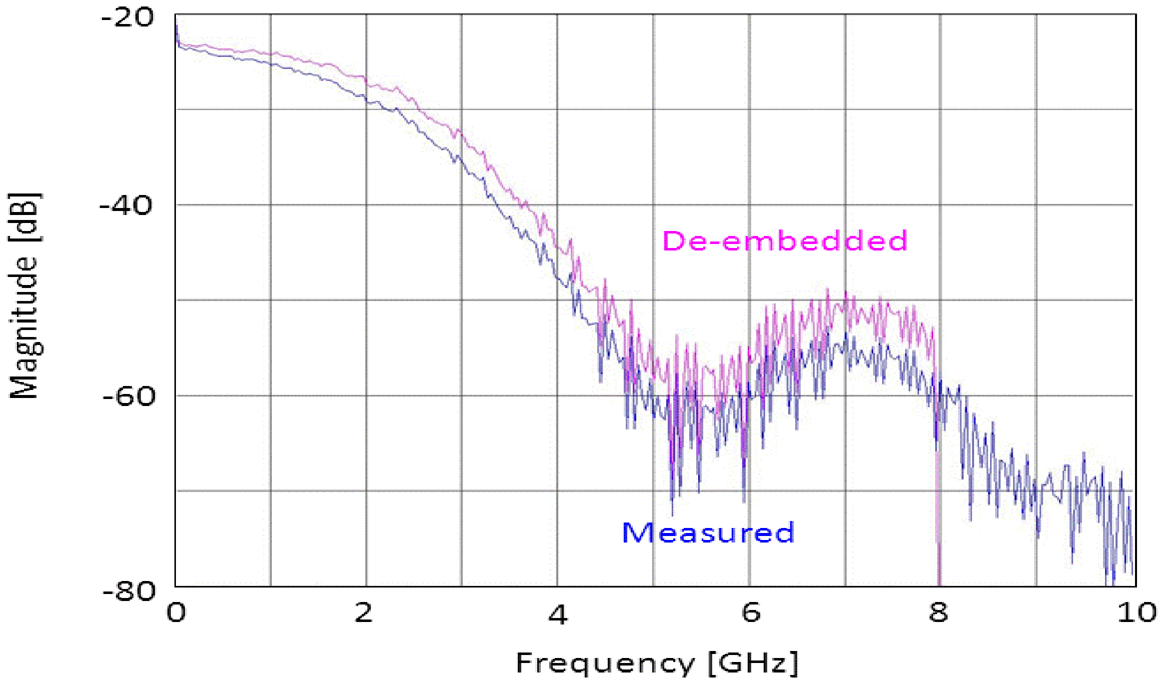


Figure 11: PRBS Spectrum (Measured and De-embedded)

Finally the waveforms V_2 and V_1 in Figures 7 and 10 are converted to the eye diagrams shown in Figure 12, which clearly depicts how the de-embedded (equalized) waveform is improved.

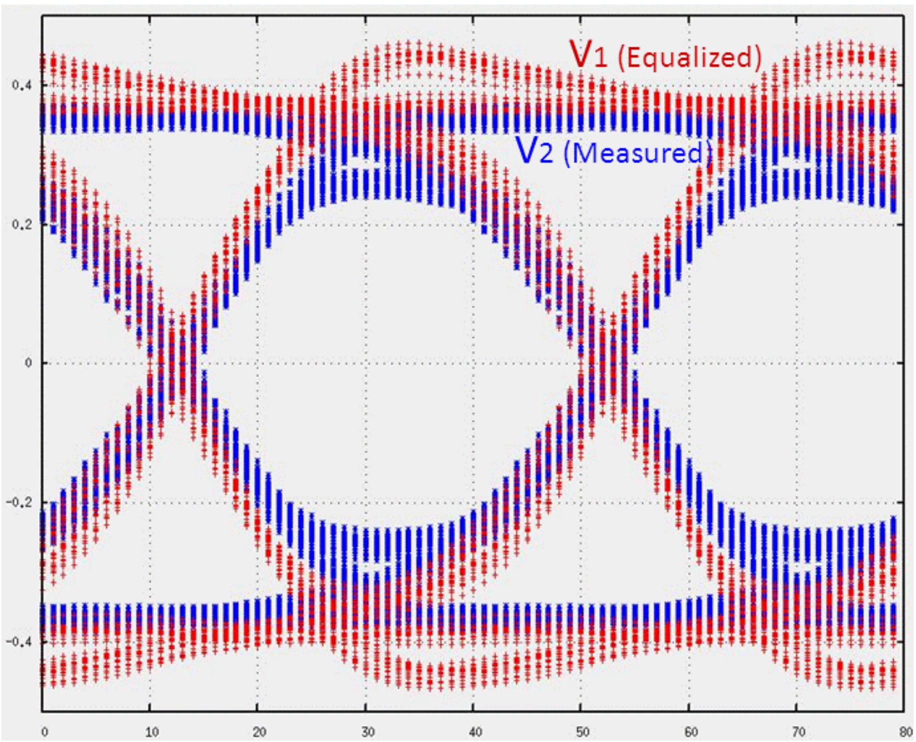


Figure 12: Eye Pattern (Measured and De-embedded)

This de-embedding calculation is performed by using the sample code in List 1. The original waveform is supposed to be in the array "dWave2[]". The subroutine "CLagrangeInterpolation()" is specifically discussed in the previous Lecture Series #48. Refer the article for detail.

```
int            i,N,Nbw;
double         dTsmp,dFsmp,dFresln,dBW,dFreq;
COMPLEX        s11,s21;
COMPLEX        CZero=COMPLEX(0.0,0.0);
COMPLEX        COne=COMPLEX(1.0,0.0);
ARRAY_D        dWave1;                 // De-embedded Signal Waveform (V1)
ARRAY_D        dWave2;                 // Measured Signal Waveform (V2)
ARRAY_D        FREQ;                   // Frequencies of S11 & S21 data
ARRAY_COMPLEX CS11,CS21;               // Measured S11 and S21
ARRAY_COMPLEX CWave1,CWave2;
ARRAY_COMPLEX CSp1,CSp2;
N=5080;                                 // # of data in the waveform here
dTsmp=5.0 ps;                           // Equivalent sampling step
dFsmp=1.0/dTsmp;                        // Equivalent sampling rate
dFresln=dFsmp/N;                        // Spectrum frequency resolution
dBW=8.0 GHz;                            // De-embedding bandwidth
Nbw=(int)(dBW/dFresln);                 // Bin # of de-embedding bandwidth
DSP_CONV_D_C(dWave2,CWave2,1.0,0.0);  // (double)→(COMPLEX); Measured V2
DSP_FFT(CWave2,CSp2,RECT);              // Then size of CSp2 = N  (not N/2)
CSp1.resize(N);                         // Input Signal Spectrum Container
CSp1.init(CZero);                       // To make "Nbw" to "N/2" as (0,0)
CSp1[0]=CSp2[0];                        // DC
for (i=1;i<Nbw;i++) {                             // De-embedding upto #Nbw
   dFreq=dFresln*i;
   //  Refer Lecture Series 48 about "CLagrangeInterpolation()"
   s11=CLagrangeInterpolation(dFreq,FREQ,CS11);  // S11 and S21 Estimat.
   s21=CLagrangeInterpolation(dFreq,FREQ,CS21);  // at Frequency "dFreq"
   CSp1[i]  =(COne+s11)/s21*CSp2[i];                  // V1=V2*(1+S11)/S21
   CSp1[N-i]=COMPLEX(CSp1[i].real(),-CSp1[i].imag());// Complex Conjugate
}
DSP_IFFT(CSp1,CWave1);
dWave1=CWave1.getReal();                          // De-embedded Result V1
```

List 1 :   Sample Code