Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 42
Jitter Calculation by Spectrum

Verigy Japan
November 2011

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.

Preface

When testing clock signals, jitter is the second typical test item after the frequency. Jitter is usually measured by using a time measurement resource. However, it can be evaluated by using an analog approach such as spectrum analysis because jitter is a kind of phase noise. The theme of this article is how to calculate jitter by looking at a frequency spectrum.

Jitter as Phase Noise

Equation (1) shows a generic sinusoidal waveform $v(t)$ containing a small phase noise $\phi(t)$. As it shows, $v(t)$ can be approximated and separated as signal and noise components at the end.

$$\begin{aligned} v(t) &= A\sin\left(\omega_s t + \phi(t)\right) \\ &= A\cos\phi(t)\sin(\omega_s t) + A\sin\phi(t)\cos(\omega_s t) \\ &\begin{bmatrix} \phi(t) \approx 0 \\ \cos\phi(t) \approx 1 \\ \sin\phi(t) \approx \phi(t) \end{bmatrix} \\ &\approx A\sin(\omega_s t) + \phi(t)A\cos(\omega_s t) \\ &= Signal + Noise \end{aligned} \tag{1}$$

Then the rms phase noise is defined as the power ratio of the noise and the signal as follow;

$$PhaseNoise\,RMS = \sqrt{\frac{NoisePower}{SignalPower}} \tag{2}$$

The time that the signal $v(t)$ crosses over the threshold of 0 can be solved as follows;

$$\begin{aligned} &v(t) = A\sin(\omega_s t + \phi(t)) = 0 \\ &\omega_s t + \phi(t) = n\pi \qquad (n = 0,\pm1,\pm2,\ldots) \\ &\therefore t = \frac{n\pi}{\omega_s} - \frac{\phi(t)}{\omega_s} \end{aligned} \tag{3}$$

Since the first term is a constant, the second term represents jitter based on the phase noise $\phi(t)$. So the jitter can be defined as the phase noise divided by the angular frequency as follows;

$$JitterRMS[s.rms] = \frac{PhaseNoise\,RMS[rad.rms]}{\omega_s[rad/s]} = \sqrt{\frac{NoisePower}{SignalPower}} \cdot \frac{1}{\omega_s} \tag{4}$$

Equation (4) suggests how to calculate jitter from signal spectrum. Firstly FFT processes a measured signal waveform into a frequency spectrum as Figure 1 illustrates. Then collecting the phase noise components, and calculating the ratio of the noise and the signal, and finally the ratio should be divided by the angular frequency of the test signal.

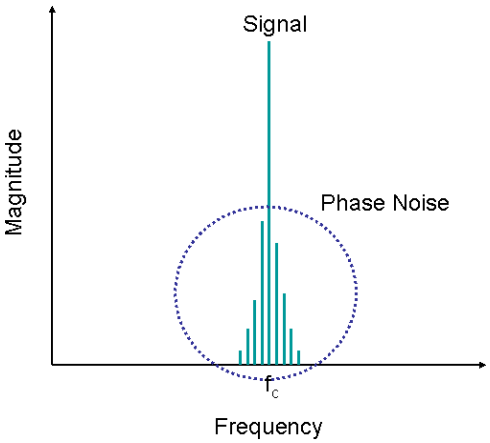

Figure 1 Phase Noise Spectrum

Simulation – Clock Signal

Figure 2 shows a 600 MHz clock in which 100 ps.pp of phase modulation is applied[1]. The original waveform is constructed with 256 clock periods in the 8192 points of data. The sampling frequency is 19.2Gsps so that the frequency resolution is 2.34375 MHz. The edge error is +/-50 ps with a single cycle of sinusoid as the bottom graph in Figure 2 shows.

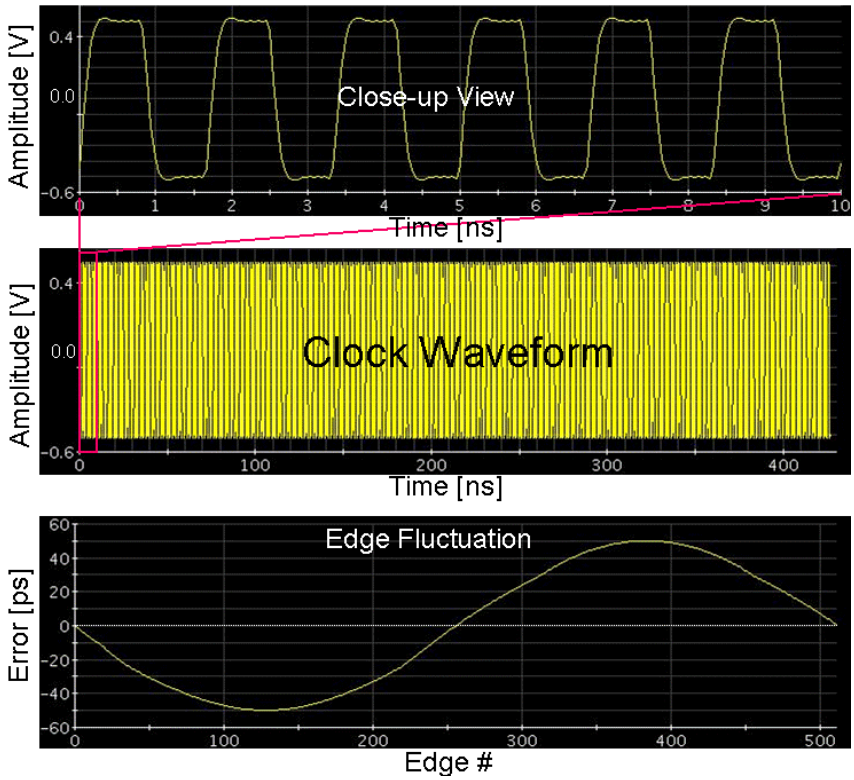

Figure 2 Jitter Injected Clock Waveform

[1] DSP-Based Testing Fundamental 30 – Jitter Injection

The waveform of Figure 2 is sliced as 256 pieces and overlaid to a single clock waveform, which is shown in Figure 3.

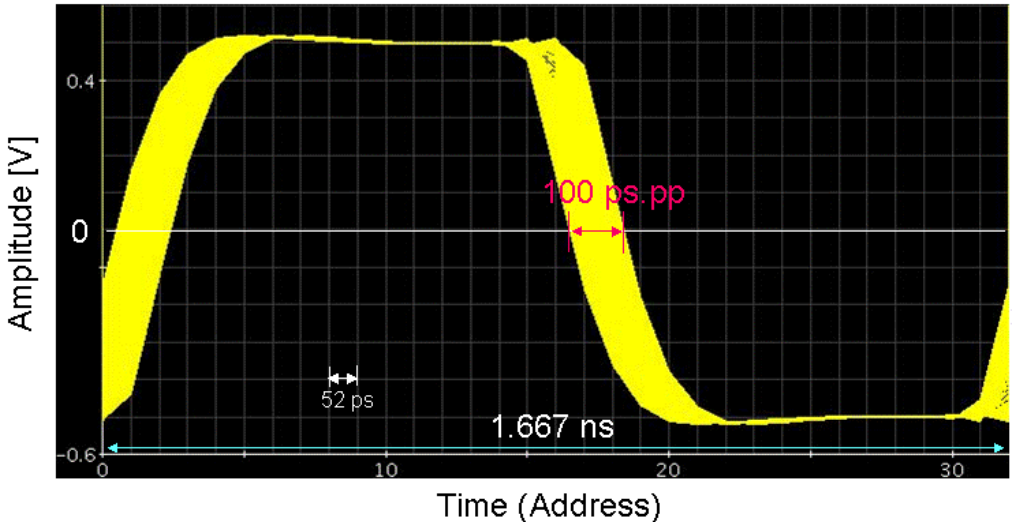

Figure 3 **Overlaid Jitter Injected Clock**

The jitter in Figure 3 is 100 ps.pp. As shown in Figure 2, this is a sinusoidal jitter so that the RMS jitter is equal to 50.0 ps/sqrt(2.0)=35.3 ps.rm.
Figure 4 illustrates the frequency spectrum of the clock signal in Figure 2. The phase modulation components can be found besides the fundamental tone.

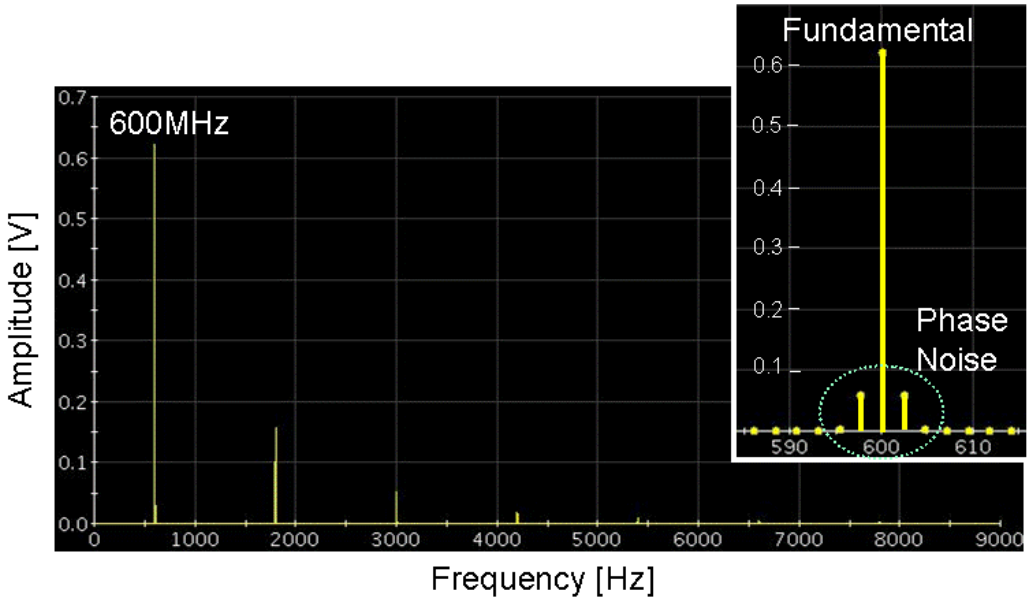

Figure 4 **Jitter Injected Clock Spectrum**

According to Equation (4), you can calculate the jitter from the spectrum. List 1 describes the calculation process.

```
10:   INT     i,k;
11:   INT     N=8192;                          // # of sampling points
12:   INT     Ncycles=256;                     // # of clock periods in UTP
13:   DOUBLE  dFs=19.2 GHz;                    // Sampling Frequency
14:   DOUBLE  dFrequency;                      // Clock Frequency (600 MHz)
15:   DOUBLE  dFresln=dFs/N;                   // Frequency Resolution
16:   DOUBLE  dSP,dNP;
17:   DOUBLE  dJitterRMS;                      // RMS Jitter Container
18:   ARRAY_D dWave;                           // Sampled Waveform
19:   ARRAY_D dSp;                             // FFT Spectrum Container
20:
21:
22:   dFrequency=Ncycles*dFresln;              // Clock Frequency
23:   DSP_SPECTRUM(dWave,dSp,VOLT,1.0,RECT,0);
24:
25:   k=10;                                    // Decide Experimentally
26:   dNP=0.0;                                 // Noise Power Container
27:   for (i=(Ncycles-k);i<=(Ncycles-1);i++)
28:         dNP=dNP+dSp[i]*dSp[i];
29:   dSP=dSp[Ncycles]*dSp[Ncycles];           // Signal Power
30:   for (i=(Ncycles+1);i<=(Ncycles+k);i++)
31:         dNP=dNP+dSp[i]*dSp[i];
32:
33:   dJitterRMS=sqrt(dNP/dSP)/(2.0*M_PI*dFrequency);
34:   dJitterRMS=dJitterRMS/(1.0 ps);          // [ps.rms]
35:
```

List 1 **Example Source Code**

The points in this processing are as follows;

- DSP_SPECTRUM() must be used in the VOLT mode.
- Collect the phase noise components around the Fundamental tone. How many bins should be collected is decided experimentally. In List 1, it is k=10.
- Sum up the noise power ("dNP") around the fundamental tone. (Lines 25..31)
- As Line 33, the square root of (dNP/dSP) must be divided by $\omega=2\pi$*dFrequency in order to convert the rms phase noise into the rms jitter.

The processing in List 1 is applied to the data in Figure 2, providing 35.5 ps.rms of jitter which is the same as the edge error analysis in Figures 2 and 3. This is the way you can estimate the rms jitter by analyzing the frequency spectrum of the clock.