Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 9
Under Sampling 2

Verigy Japan
January 2009

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.

Under-sampling – Practical Conditioning (1)

The fundamentals of under-sampling were discussed in a previous article. In this and subsequent issues, practical conditioning strategy of under-sampling is discussed in various applications. There are two types of analysis methods available. One is waveform analysis in the time domain, and the other is spectrum analysis in the frequency domain. The waveform analyses are discussed in this issue.

Extended Coherency Equation

Equation (1) is the coherency equation for under-sampling as discussed in the previous issue. Usually *Ft* and *N* are decided at first. Then *M* or rather *Mx* should be decided according to the test strategy, and the sampling rate of the sampler would be settled finally.

$$\frac{Ft}{Fs} = \frac{M}{N} = K + \frac{Mx}{N}$$

Ft: Signal Frequency M: Number of Cycles K: Zone Number

Fs: Sampling Frequency N: Number of Points Mx: Aliased Bin Number (1)

This equation is important to settle your sampling condition appropriately. When you make up your test plan, recall this equation.

What kind of signals samplers can see

The sampling rate of a sampler is limited; however the analog input range of the sampler is much greater than the sampling rate. As discussed in the previous issue, the sampler takes extremely broad range signals, aliasing them in the baseband which is from DC to the half of sampling frequency. Entire signals fall in the narrow band. Under-sampling loses the frequency information so that you have to carefully make up your sampling condition. Otherwise signals would be mixed up in the baseband plane and you cannot distinguish each signal.

Let's think of what kind of signals samplers can deal with. Recall the degenerated frequency domain diagram in Figure 1, which depicts four typical situations. If the actual sampling rate of the sampler is expressed *Fs*, the baseband is from DC to *Fs*/2. You should set up your condition that the test signal bins appropriately alias in the baseband without overlapping with each other. In the example of Figure 1 the number of data *N* is 32 so that the baseband spectrum contains 16 bins, meaning entire signal spectrum would be aliased in the bin #0 to #15 in the case.

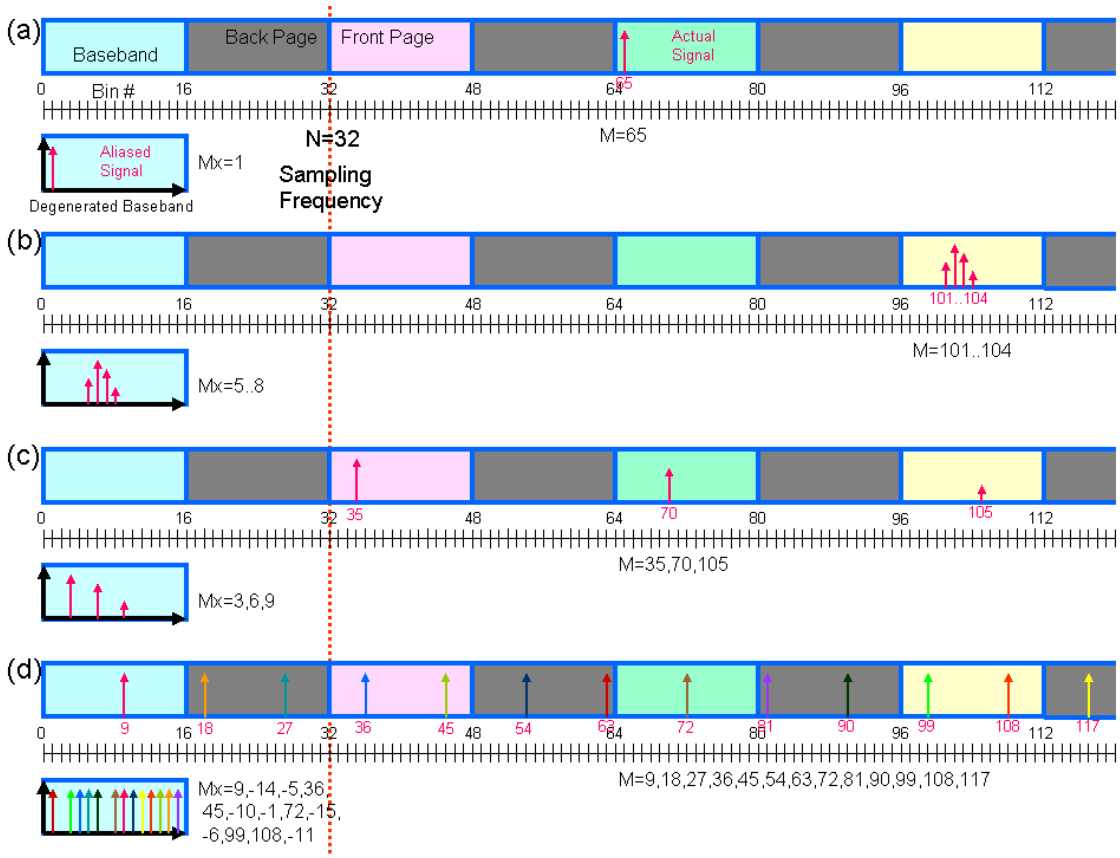


Figure 1: Various Situations of Aliasing

(a) Single Tone

A single tone analysis is the most typical application of a sampler. There are many situations that you want to analyze precise waveform. For instance, rise/fall time, duty cycle ratio, settling time, glitch energy and pulse template tests are typical applications. When analyzing the precise waveform of the signal, you should settle the coherency condition as *Mx*=1. Then the captured waveform automatically restores the original waveform. In Figure 1(a), the signal tone is located at *M*=65 so that *Mx*=1, which is an appropriate condition. For some reason, you could not settle *Mx*=1 in your plan, for instance *M*=101, the tone would fall in the bin #5 (=101-96). In that case you can reconstruct the original waveform with employing DSP_SHUFFLE(Key Number=5).

(b) Band-limited Signal

When the test signal is localized within a narrow bandwidth, for example, a modulated signal such as AM or FM, or dual-tone inter-modulation test, samplers can successfully replicate the original spectrum in the baseband. In Figure 1(b) the signal is located at the bins #101, 102, 103 and 104 so that they would be aliased in the bins #5, 6, 7 and 8 in the baseband. In other words, sampling is a frequency conversion. High frequency signals can be shifted to the baseband. As you already notice, the test signal should be localized within the bandwidth of the baseband. And you should adjust your sampling frequency with managing Equation (1) as your test signal would be allocated in one of the front pages of the extended frequency domain (colored pages in Figure 1).

(c) Harmonics Signal

This is actually similar to (a). When the test signal at the bin #35 is not a good sinusoidal waveform, the harmonics would be located at the multiple of 35, such as #70 and #105. Therefore each component would fall in the bins #3, 6 and 9 in the baseband. The point in this case is that you should make a plan to align the fundamental and its harmonics components in the so-called front pages that are colored in Figure 1. Then all the components would fall in the baseband with keeping the same order, replicating a natural spectrum.

(d) Multi-tone Signal

Usually a sampler is not good at dealing with a wide-band signal, because entire spectral components fall in the limited baseband and some of them occupy the same bin locations. When falling in the same location, the information of each component would be mixed up and could not be separated with each other anymore. However, if you carefully make a plan of bin allocation, you can do wider band frequency analysis than the bandwidth of the baseband. Figure 1(d) may be an example of multi-tone filter testing. You can see 13 tones in the picture. Each of them does not fall in the same aliased bin, i.e. absolute value of Mx. In Figure 1 there are 15 bins available in the baseband excluding DC bin so that maximum 15 tones could be allocated in the multi-tone. In this situation you cannot avoid allocating some tones in the back pages colored gray in the figure. Therefore the baseband bin location would be scrambled; however, yourself program the test signal multi-tone so that you can figure out which tone falls in which bin in the baseband by the coherency Equation (1).

Sampler Applications – Waveform Analyses

The following test items are typical waveform analyses.

- Rise time, fall time, duty cycle ratio, and jitter of a clock signal
- Settling time and glitch energy of a DAC
- Template tests of data communication transceivers

For these items, a single cycle of the target waveform should be reconstructed in the time domain. If you would like to reconstruct a single cycle waveform directly, you should make a plan as $Mx=1$ in the coherency condition. When you make up the condition as $Mx>1$ for some reason, you can reconstruct the single-cycle waveform with employing the DSP API "DSP_SHUFFLE()" with the key number Mx.

(1) DAC Settling Time Measurement

Let's look at an example of DAC settling time measurement, which is a typical waveform analysis that needs a wide analog bandwidth, because the waveform looks square wave containing lots of harmonics components up to very high frequency. Figure 2 depicts the test configuration. The 8-bit DAC is asserted the all 0's and all 1's data alternately, generating voltages of the zero-scale (code 0) and the full-scale (code 255). It looks as a square wave clock signal whose frequency is half of the DAC sampling rate.

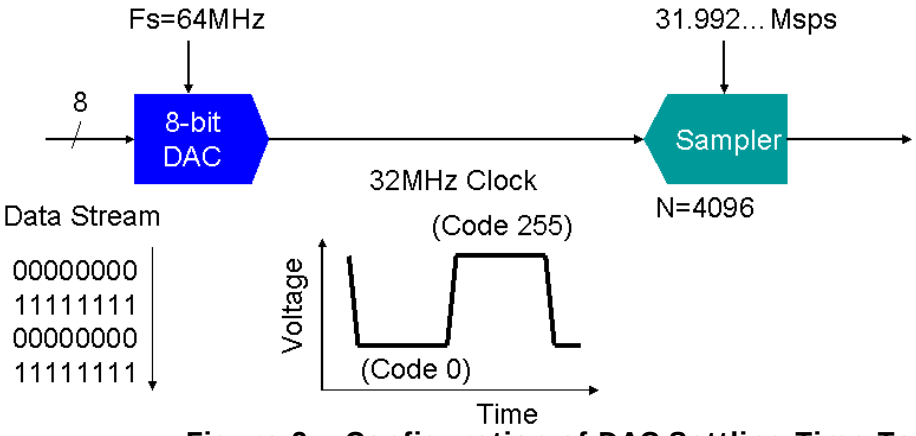


Figure 2: Configuration of DAC Settling Time Test

The sampler should capture an exact single cycle of the clock waveform. The data size *N* is 4096 here. The coherency condition can be calculated with using Equation (1). The Coherency Calculator in Figure 3 can help you manage Equation (1). When you put in the DAC condition as *Fs*=64MHz, *N*=2 and *M*=1 in the top section of the tool, the test frequency *Ft* is figured out as 32MHz, which is the frequency of the clock-like waveform. A sampling rate would temporarily be set as 31.9MHz in the bottom section, which should be slightly lower than *Ft* (=32MHz). *N* is filled as 4096, and press the button labeled "Calculate". Then adjust *Mx*=1 and calculate again. Finally the tool tells you the precise *Fs* as 31.992189…MHz, which can be copy-and-pasted in the analog setup tool in the SmarTest.

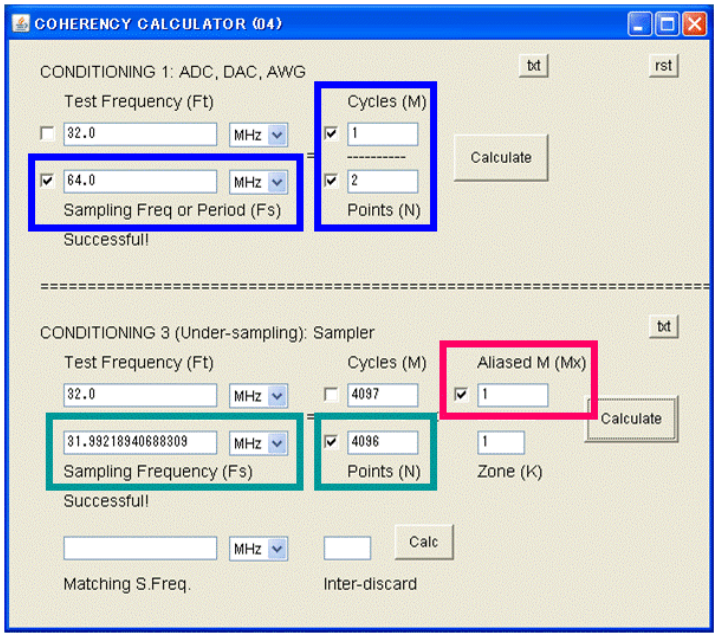


Figure 3: Coherency Calculator1 (on MS Windows XP®)

[1] The coherency calculator is a personal Java tool that Author developed to resolve the coherency Equation (1) when planning test condition. If you are interested in it or would like to try to use it, consult your local application engineer.

An example measurement result of a DAC settling time looks as Figure 4. After you successfully capture a single cycle clock waveform, you can analyze the settling time with using a DSP API "DSP_SETTLING()."

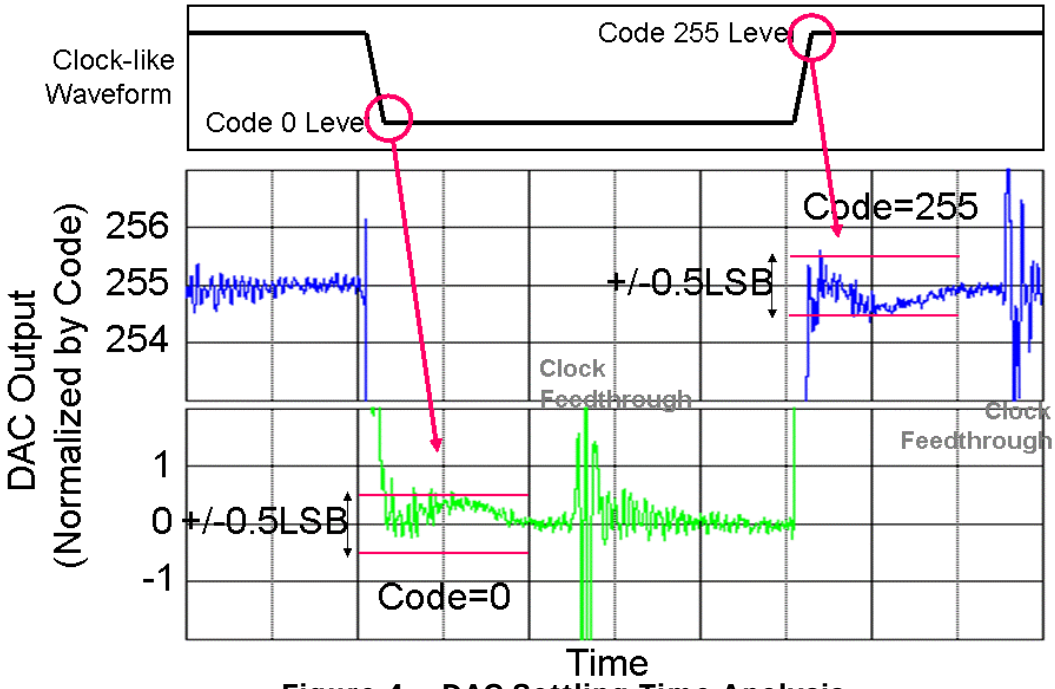


Figure 4: DAC Settling Time Analysis

(2) Clock Waveform Analysis

For some reason, if you need to settle *Mx* more than 1, and still you need a single cycle waveform, you could reconstruct the waveform with using an API DSP_SHUFFLE() with the number of cycles captured in the original waveform data.

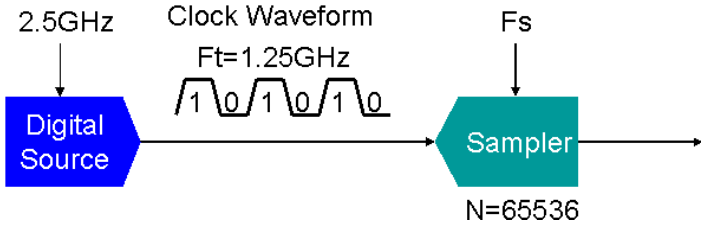


Figure 5: Configuration of Clock Waveform Analysis

In Figure 5, a digital source generates toggling data 101010 at the rate of 2.5GHz. It becomes 1.25GHz clock waveform. The signal is measured by a sampler, capturing *N*=65536 points. With using the coherency calculator, the condition is settled as *Mx*=3001, and the sampling frequency is resolved 103.770680…Msps as Figure 6.

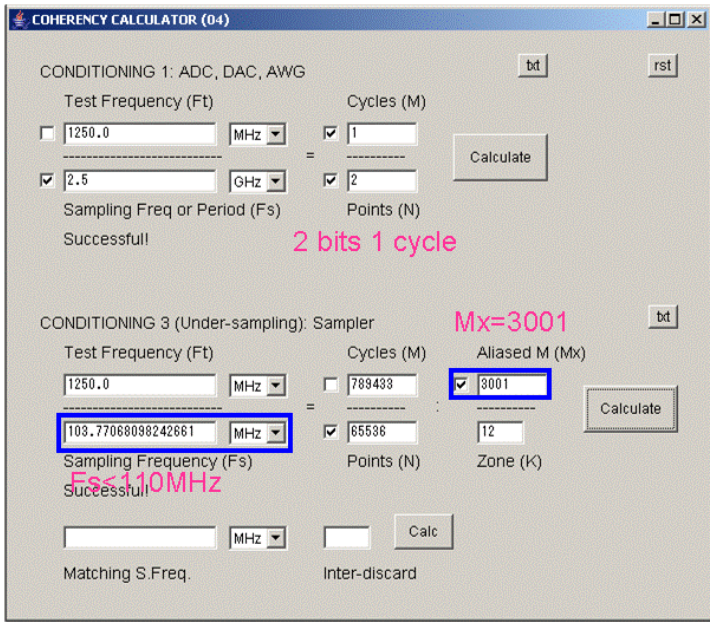


Figure 6: Coherency Calculator (on MS Windows 2000®)

Figure 7(a) is a measured waveform, and applying DSP_SHUFFLE() with the key number *Mx*=3001. A single-cycle clock waveform is reconstructed as (b). Then with slicing the voltage span at 10% and 90%, you can analyze the rise and fall times. In the vicinity of the mid levels in Figure 7(b), you can see lots of data points scattered at the rising and falling edges. Regressing ideal straight lines in the rising/falling edges, you can estimate jitter with the method depicted in Figure 8. The voltage difference from the scattered data points to the ideal straight line can be interpreted into the time errors with referring the slew rate of the rising/falling edges. Consequently with processing the time error, the peak-to-peak jitter and the rms jitter can be derived.[1]

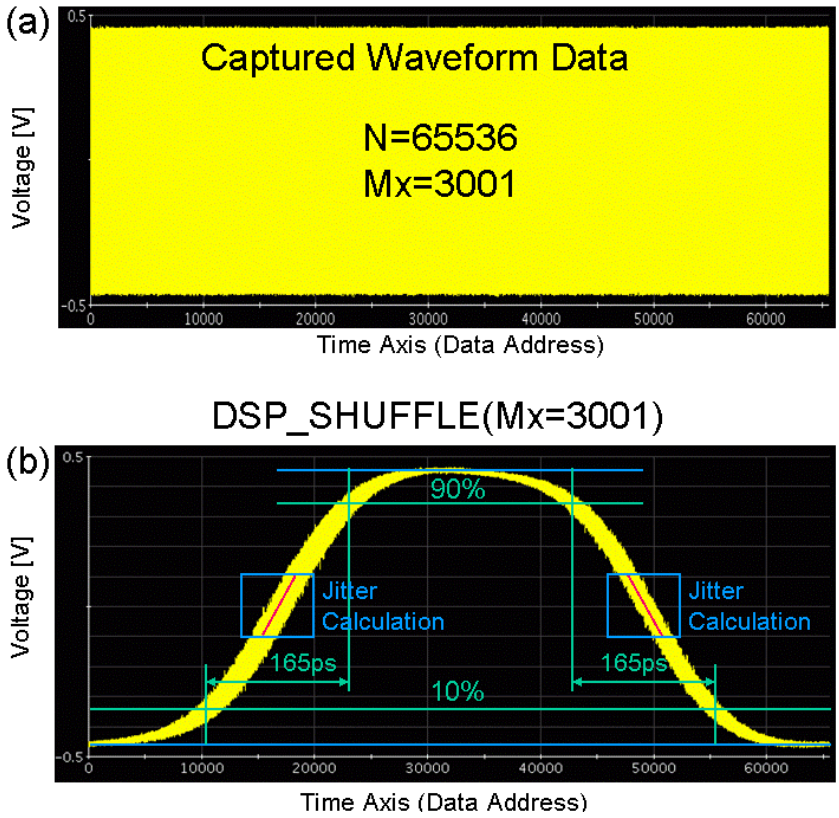


Figure 7: Measured Waveform and Reconstructed Waveform

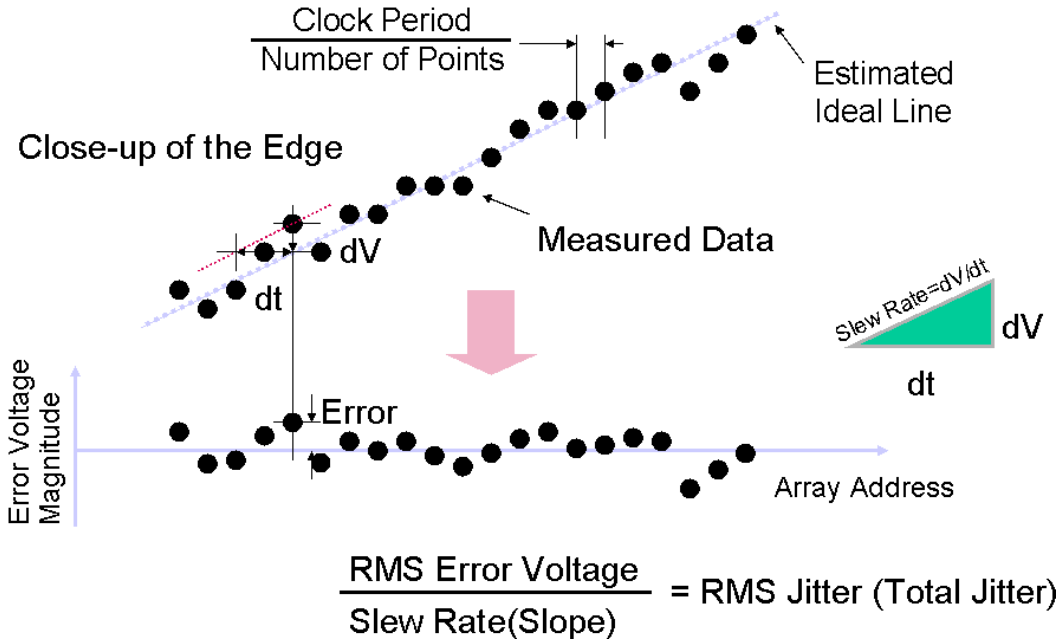


Figure 8: Jitter Calculation (Waveform Method)

(3) PRBS Eye Pattern Analysis

Samplers deal with repetitive signal only; however when measuring an eye pattern by using a pseudo random bit stream, there is a tricky method available.[2] Figure 9

is the test configuration; a digital transmitter generates 127-bit PRBS at 2.5Gbps, which is measured by a sampler.

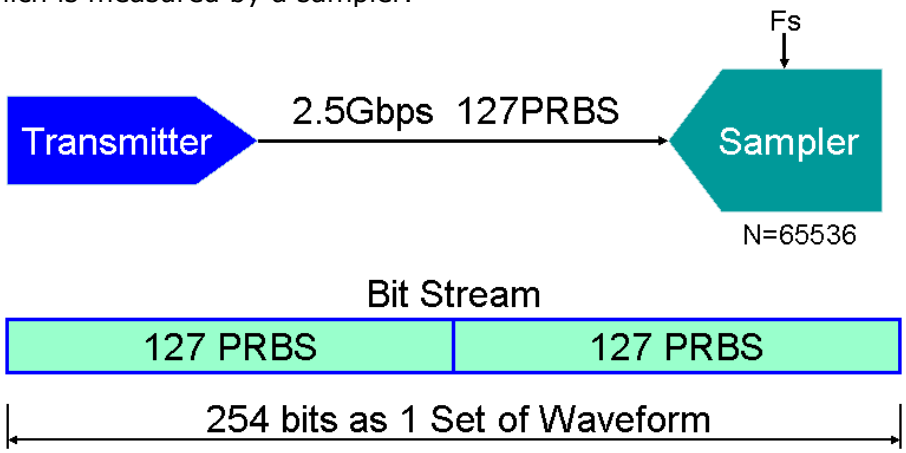


Figure 9: Configuration of Eye Pattern Measurement with PRBS

The transmitter repeats 127 bits PRBS set pattern. The sampler is programmed to capture exact 2 sets of the PRBS pattern i.e. total 254 bits with *N*=65536 points of data. The purpose of capturing 2 sets as a test waveform is to create a 2-UI period eye pattern with applying the reshuffling DSP API DSP_SHUFFLE(). The sampling rate is calculated as Figure 10 shows.

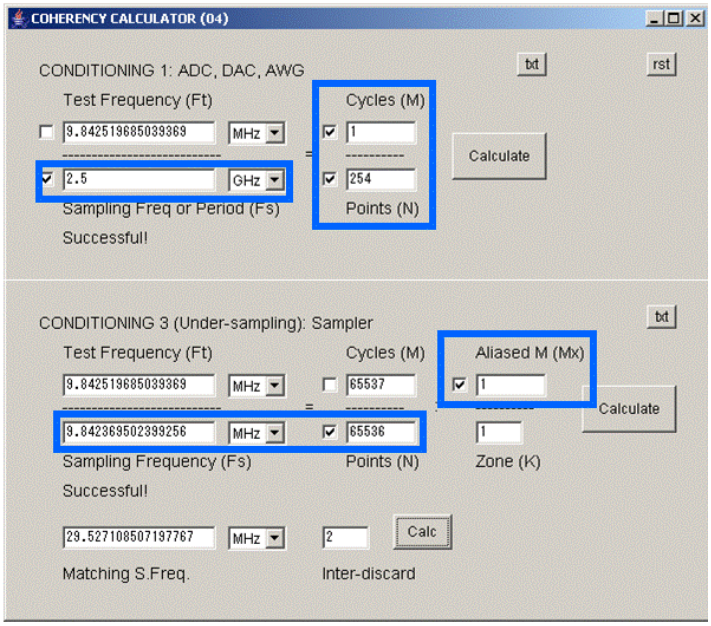


Figure 10: Coherency Calculator

Then the sampler runs at the rate of 9.8423695…Msps. If it is the 6G Quad Sampler, the sampling rate is settled as 29.5271…Msps and the initial discard number should be set 2.

Consequently the captured waveform looks as Figure 11(a), which contains 2 sets of 127-bit PRBS waveform exactly. Applying the DSP API DSP_SHUFFLE() with a key number "127" (not 254 for 2-UI period eye pattern; 127 and 254 are mutually prime), the eye pattern is reconstructed as Figure 11(b). Then you can do an eye mask test and jitter calculation.

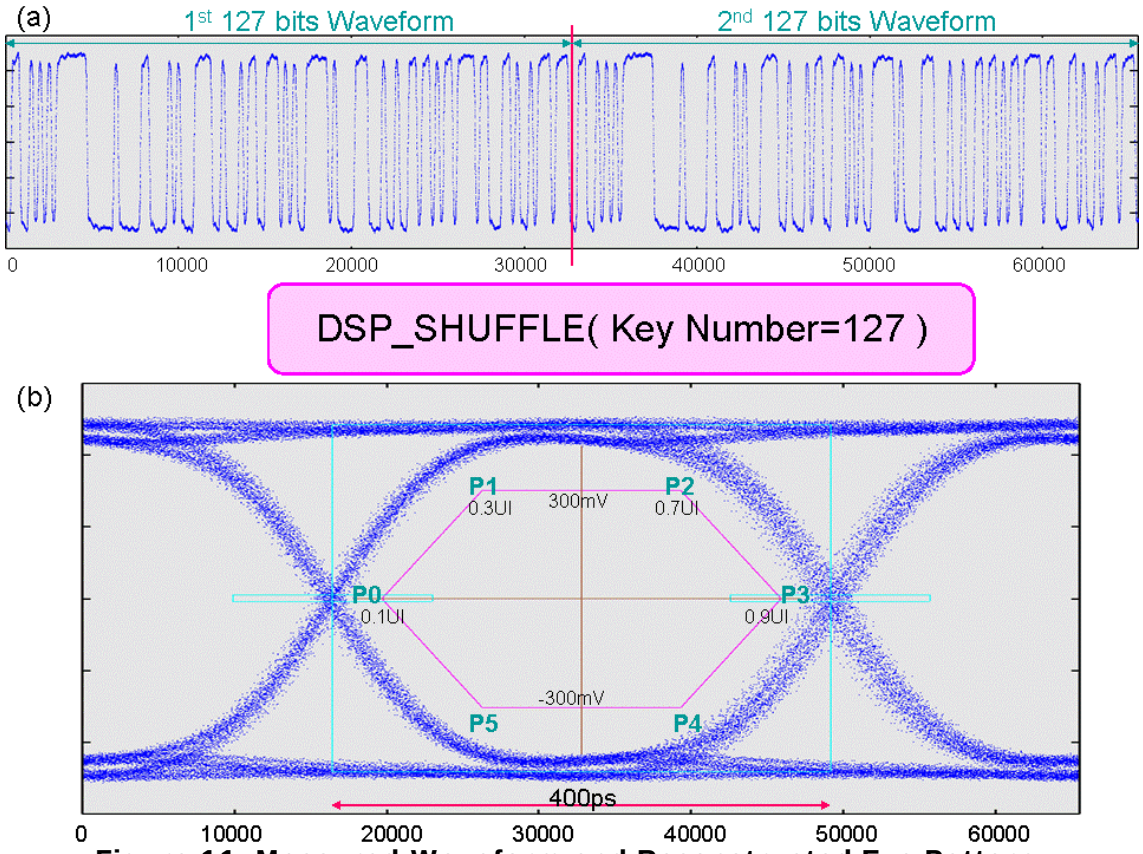


Figure 11: Measured Waveform and Reconstructed Eye Pattern

(4) Another Eye Pattern Analysis

In the previous section, whole waveform of a PRBS pattern is captured as if it would be a single cycle signal. In this section, just a two-bit period is measured by a sampler. This strategy is applicable to a true random bit stream not a PRBS stream.

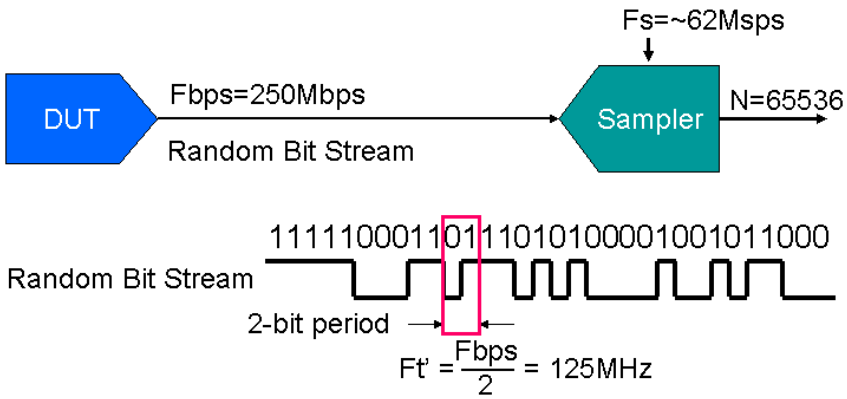


Figure 12: Configuration of Eye Pattern Measurement with Random Data

The test configuration is shown in Figure 12. The device under test (DUT) generates 250Mbps random bit stream, which is measured with a sampler. If you want to construct two-bit window eye pattern, you should take two bits period as a measurement window.

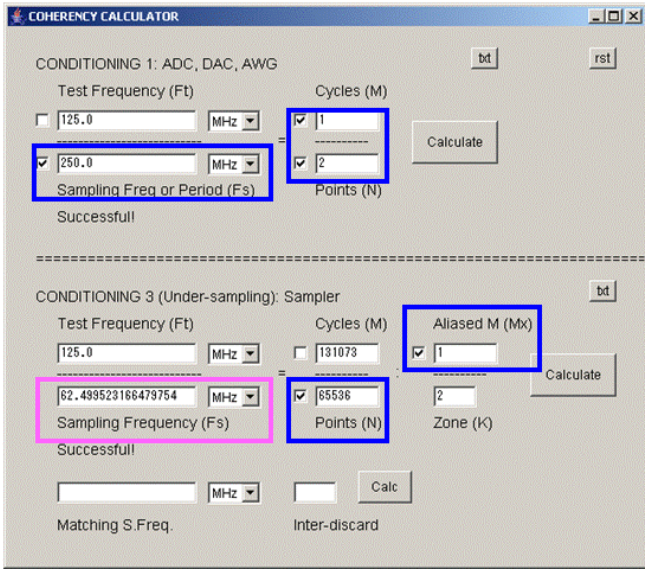


Figure 13: Coherency Calculator

Conditioning by the coherency calculator is shown in Figure 13. The top section of the tool shows that the measurement window is the 125MHz period. When you have the 6G Quad Sampler "MCC" whose maximum sampling rate is 110Msps, starting a temporary *Fs*=110MHz and *N*=65536 in the bottom section, and adjusting *Mx*=1 and *K*=2, then *Fs* is resolved as 62.499523…MHz.

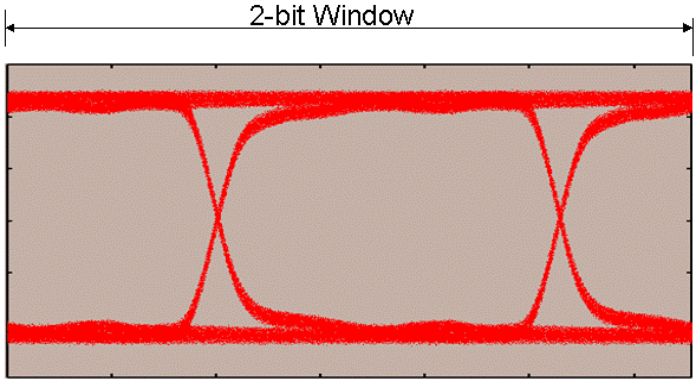
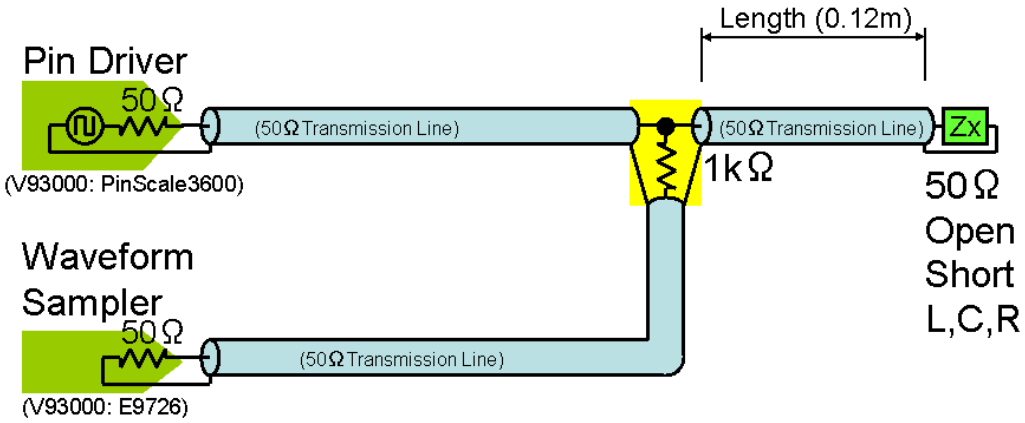


Figure 14: Measured Eye Pattern

Figure 14 shows the constructed eye pattern. Then you can analyze the eye opening and jitter from the eye pattern.

(5) TDR: Time-Domain Reflectometry

This is an interesting experiment. Figure 15 depicts the test configuration. A digital pin driver stimulates 50MHz clock to a DUT that is L, C or R through 50Ω transmission line. A 1kΩ resistor is directly connected to the transmission line at the point 12cm away from the DUT, and senses the combined forward and reflected clock signal. The composite signal is eventually captured by a waveform sampler.



Figuer 15: Configuration of TDR Experiment

The coherency condition is shown in Figure 16. The clock frequency is 50MHz so that the sampling rate is resolved as 49.9512195… MHz for Mx=1;

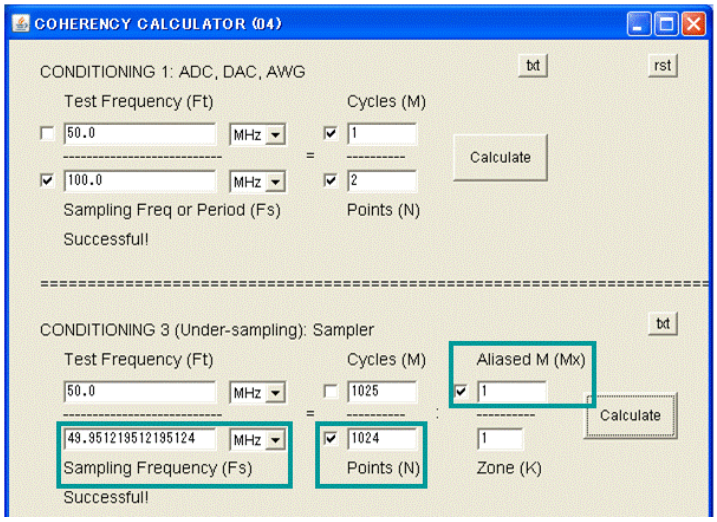


Figure 16: Coherency Calculator

The test result is shown in Figure 17. The resistive probe reduces the signal as only 5% of the original level so that the captured signal looks too noisy. Therefore the sampler runs with the averaging mode for getting clear image of the signal. The reflection appears at 1.4ns delayed for the 12cm offset distance. The signal locus depends on the L/C/R. The trace exists between the short and open loci, and you can estimate an approximate value of the LCR. The accuracy would be 10% to 20%.

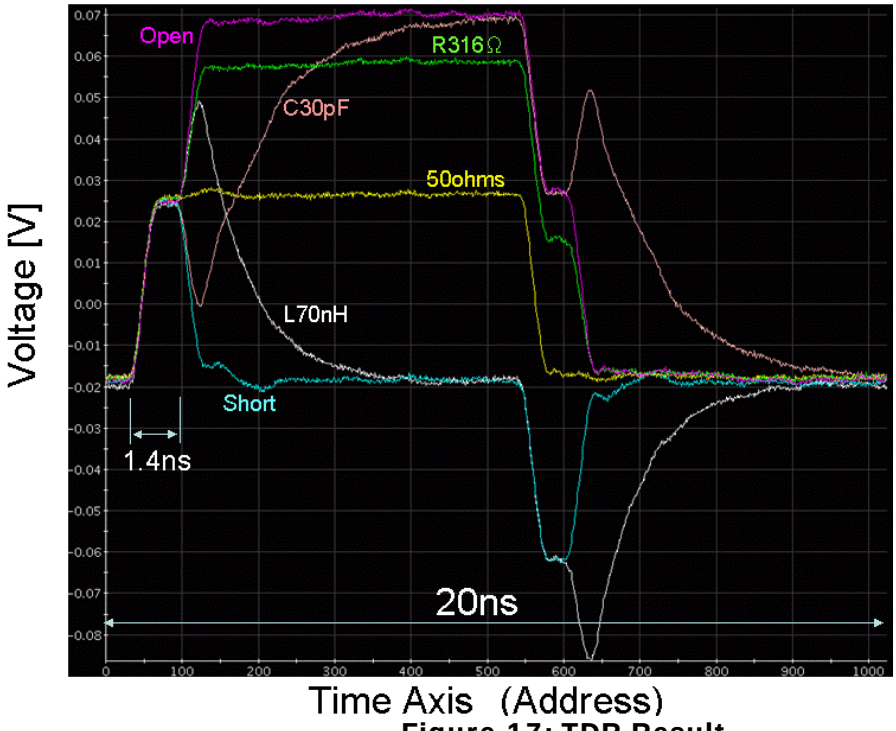


Figure 17: TDR Result

REFERENCE

[1] Hideo Okawara, "Novel Clock Waveform Analyses by a Waveform Sampler", Verigy Technical Symposium VOICE 2008
[2] Hideo Okawara, "Novel Eye Pattern Test Method by Waveform Sampler", Verigy Technical Symposium VOICE 2007