



## **Hideo Okawara's Mixed Signal Lecture Series**

### **DSP-Based Testing – Fundamentals 44 Analog Signal Generation by Digital Pin Driver I**

*Verigy Japan  
February 2012*

#### **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](http://www.verigy.com/go/gosemi).

## Preface

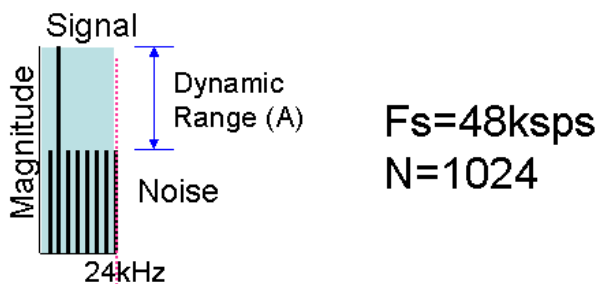
When we have only a purely digital tester available, we may encounter a situation that a sinusoidal signal is required for test. For instance, microcontrollers are usually tested with low cost digital testers. However, sometimes the DUTs integrate ADC which may be required to test with using a sinusoidal stimulus. In that case you need to provide sinusoidal stimuli with using a digital pin. The topic of this article is how to generate a good enough sinusoid with minimum external components by using a digital pin driver.

## Over-sampling

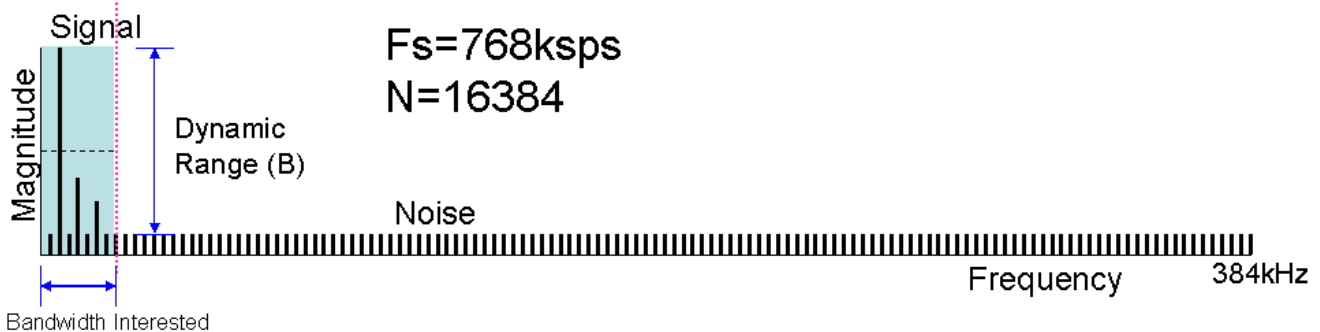
Today's precision audio ADC and DAC often employs 1-bit converters which have only two codes of 0 and 1. They are called delta-sigma type converters. The sampling rate of such devices is usually 44.1 kHz, 48 kHz, 96 kHz or 192 kHz. However, the conversion circuit inside the devices runs at more than several tens or hundreds times higher sampling rate than the nominal rate. In spite of only one-bit, the actual performance reaches more than 20 bits. There are two points here. One is over-sampling and the other is noise-shaping.

A linear n-bit converter can theoretically provide the maximum SNR of  $6.02 \cdot n + 1.76$  [dB] based on the quantization noise. Figure 1 illustrates the effect of over-sampling. The figure shows two cases of sampling rate. The spectrum in (a) shows the quantization noise of 48 ksps and 1024-point sampling. The spectrum in (b) shows the noise of 768 ksps and 16384-point sampling. The unit test period of two conditions is the same —  $1024/(48 \text{ kHz})$  and  $16384/(768 \text{ kHz})$ .

### (a) Regular Sampling



### (b) Over-sampling



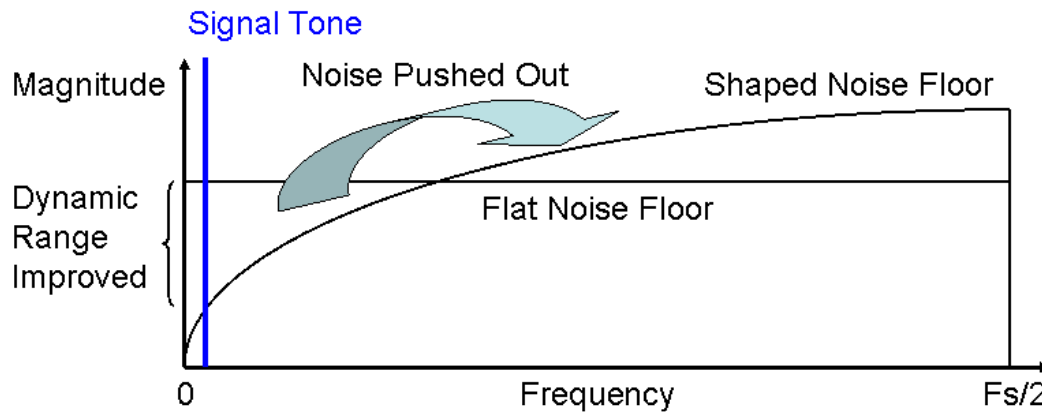
**Figure 1: Over-sampling Effect**

Since the quantization noise power is constant, the SNR based on the quantization noise is the same in both cases. When  $N = 1024$ , the quantization noise energy is distributed in the 512 bins. When  $N = 16384$ , the same energy is distributed in the 8192 bins. Theoretically the quantization

noise floor is located at  $10 \cdot \log(N/2)$  [dB] below the level of SNR value. The higher the sampling rate is, the lower the noise floor and the wider the dynamic range become. This is the effect of over-sampling.

### Noise Shaping

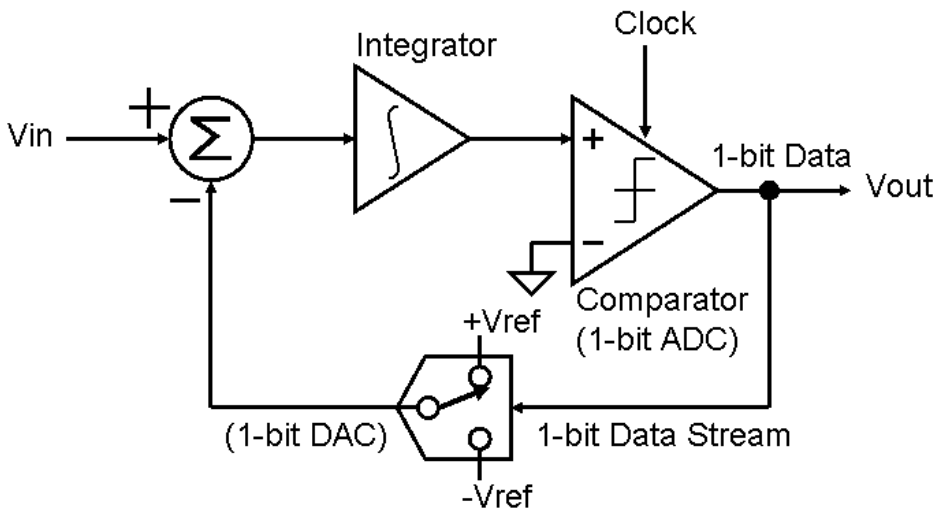
The quantization noise is distributed all over the frequency range and the noise floor looks flat or white in Figure 1. By using some digital processing method, you can push out the noise in the low frequency range to the high frequency range. This is called noise shaping. See Figure 2. Then if you can remove the noise in the high frequency range, you can earn more dynamic range in the low frequency area.



**Figure 2:** Noise Shaping Effect

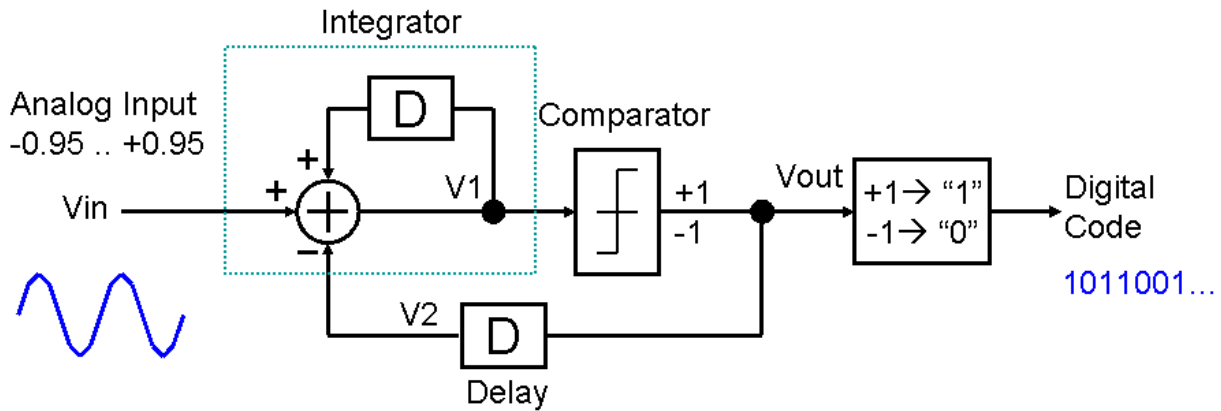
### Delta-sigma Modulator

The over-sampling and the noise shaping are the key techniques of 1-bit DAC which is known as delta-sigma converter. One-bit DAC uses only two levels of data so that the code generated for a 1-bit DAC can be downloaded directly to a digital pin electronics which can perform as a 1-bit DAC. The simplest delta-sigma modulation is depicted as Figure 3.



**Figure 3:** Delta Sigma Modulator

This circuit diagram can be converted into a block diagram of signal processing as Figure 4. "D" in the diagram means one-clock delay of the data. The input analog signal is normalized between +1 and -1. The comparator discretizes the integrator output with the threshold level of zero as +1 or -1, which becomes the output digital data stream. The comparator output is one-clock delayed and made a feedback to the input.



**Figure 4: First Order Noise Shaping Block Diagram**

List 1 shows an example signal processing procedure based on the diagram in Figure 4. Lines 24 to 29 and 32 to 38 are the noise-shaping loop. Lines 26 and 34 have the same variables "dV1" in both sides of "=" so it may look peculiar. But this realizes a one-clock delay. The "dV1" in the left-hand side is the current value and the one in the right-hand side is the last or one-clock delayed value. "dV1" and "dV2" should be initialized at Lines 22 and 23 in advance, and then Lines 24 to 29 initialize and stabilize the system. The actual digital bit stream is generated at Lines 32 to 38.

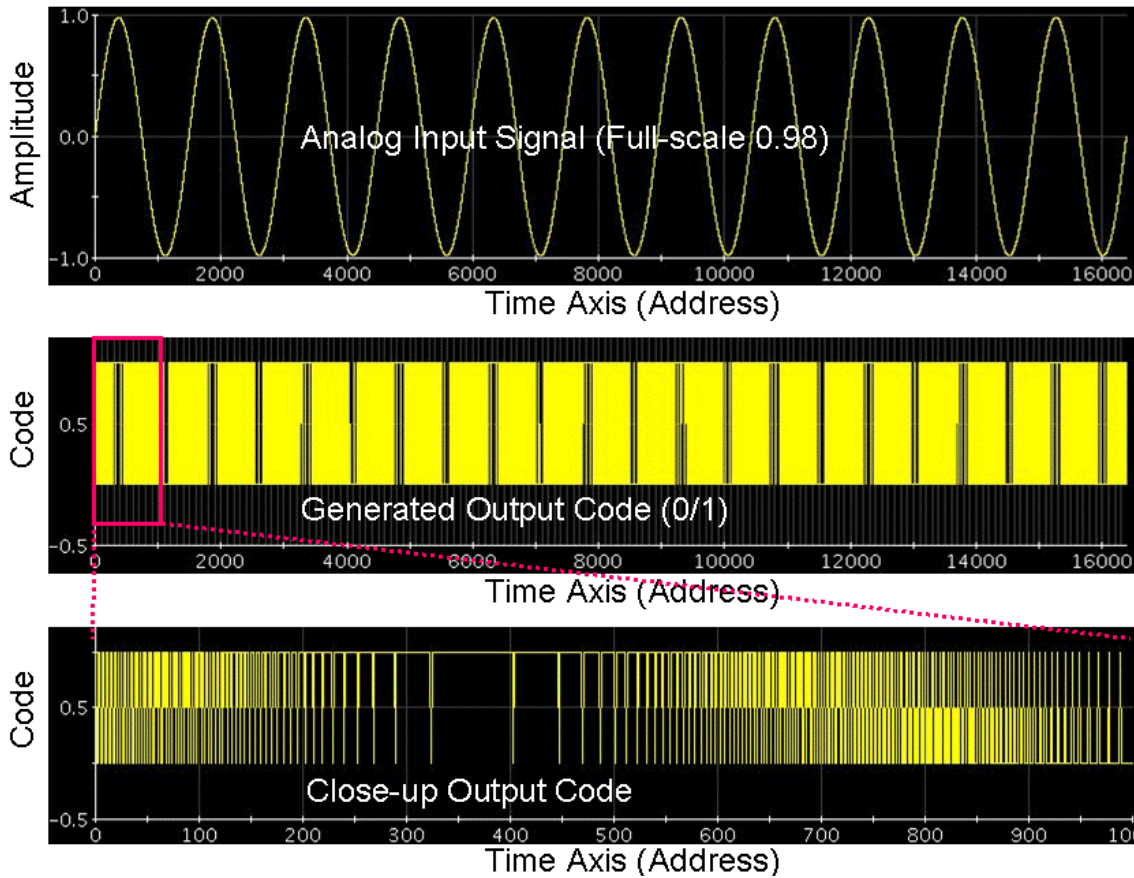
```

10:  INT      i,N,M;
11:  DOUBLE   dv1,dv2,dvin,dvout,dA,dP;
12:  ARRAY_I  iCode;                               // Digital Code Container
13:  ARRAY_D  dwave;                               // Original Waveform Container
14:
15:  N=16384;                                     // # of data
16:  M=11;                                        // # of sinusoid cycles
17:  dA=0.95;                                    // Relative Full-scale Amplitude
18:  dP=2.0*M_PI*M/N;
19:  dwave.resize(N);
20:  for (i=0;i<N;i++) dwave[i]=dA*sin(dP*(i+0.25)); // Sinusoidal waveform
21:
22:  dv1=0.0;                                     // Initialize (Don't forget)
23:  dv2=0.0;                                     // Initialize (Don't forget)
24:  for (i=0;i<N;i++) {                          // Initialize (Don't forget)
25:      dvin=dwave[i];
26:      dv1=dvin+dv1-dv2;                         // Right v1 is 1-clock delayed.
27:      if (dv1>0.0) dvout=1.0; else dvout=-1.0; // Comparator
28:      dv2=dvout;                                // 1-clock delay
29:  }
30:
31:  iCode.resize(N);
32:  for (i=0;i<N;i++) {
33:      dvin=dwave[i];
34:      dv1=dvin+dv1-dv2;
35:      if (dv1>0.0) dvout=1.0; else dvout=-1.0;
36:      dv2=dvout;
37:      if (dvout>0.0) iCode[i]=1; else iCode[i]=0;
38:  }
39:

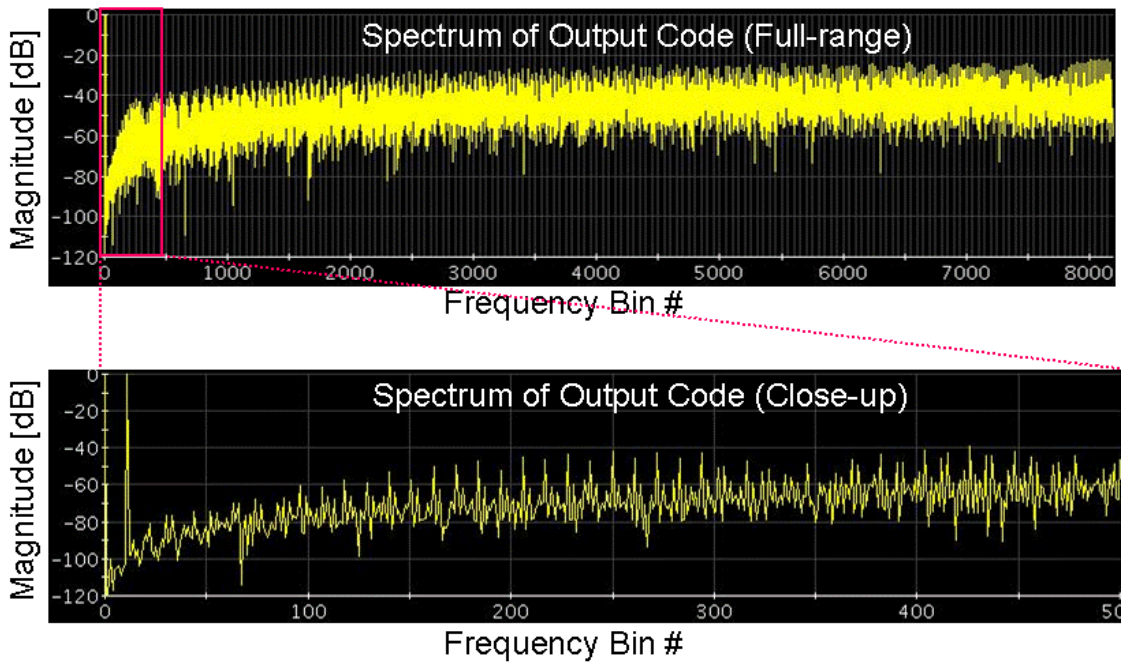
```

**List 1: First Order Noise Shaping Program Code Example**

Figure 5 illustrates the input analog signal and the output codes. The input is a 11-cycle sinusoid whose amplitude is 0.98. (Line 17 in List 1:  $dA=0.98$ ) The output code is a series of 1s and 0s. You can see condensational and rarefactional areas alternately. This is a kind of pulse density modulation. However the data is not a simple pulse width modulation proportional to the frequency. There are some noise components superimposed in it. By performing the FFT (Fast Fourier Transform) directly to the bit stream, you can see the frequency spectrum as Figure 6. You can clearly see how the noise is integrated in the frequency spectrum of the bit stream. The noise floor is not flat. The DC to very low frequency area has low noise floor, but the higher the frequency is, the higher the noise level becomes. The close-up view of the low frequency area clearly shows the noise shaping effect. The data stream is 0s and 1s codes so that the amplitude is less than 0.5. The FFT spectrum is referenced to the full-scale of amplitude 0.5. So the fundamental tone is located at the bin 11 and its magnitude is nearly 0 dB. The 2<sup>nd</sup> harmonics is almost -80 dBc and the 3<sup>rd</sup> harmonics is almost -75 dBc. If you employ a low pass filter to remove these harmonics, you could expect to have a noise floor of -90 dBc in this case. If you need more dynamic range, you may want to apply higher order noise shaping. It will be reported in the next month article with actual performance experimented by using the PS1600 pin electronics. Stay tuned.



**Figure 5: Input and Output of First Order Noise Shaping (Calculation)**



**Figure 6: Spectrum of First Order Noise Shaping (Calculation)**