VERIGY

Hideo Okawara's

Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 2

Waveform Generation

Verigy Japan

June 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.

Waveform Generation

D/A converters (DAC) are the most typical device under test (DUT) in mixed signal testing. Moreover, arbitrary waveform generators (AWG) are the analog signal sources in the modern digital signal processor (DSP) based mixed signal testers. AWG contains a DAC and waveform memory. This paper focuses on waveform generation.

DAC Data Generation

(1) How many points minimum do you need to program in a sine wave signal?

A linear n-bit D/A converter has 2n levels of output voltages. When you test a D/A converter, you need to stimulate all codes from 0 to (2n-1). When you drive a DAC from 0 to (2n-1), the signal becomes a simple ramp and is monotonic. It may be a less severe test for the device. If you want to play a dynamic test with a realistic clock speed, you can stimulate the DAC with the code of a sinusoidal waveform. In that case, how many points minimum do you need to program the DAC? This section answers this question.

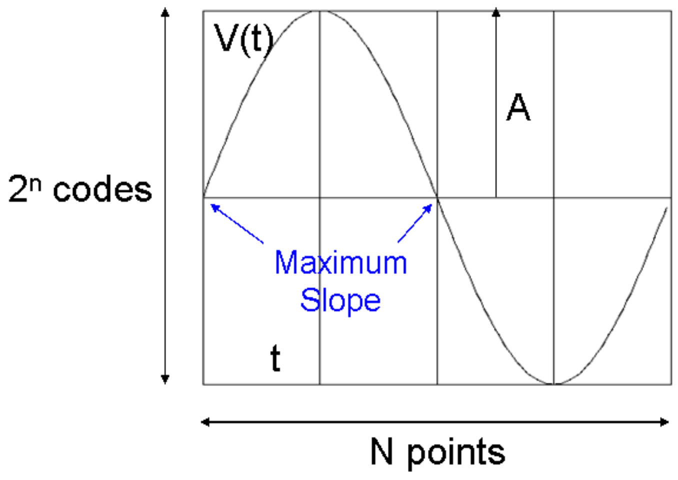

Figure 6. Full-scale sine wave.

Figure 6 depicts a single cycle of sine wave distributed in N points of DAC data. N should be greater than 2n. The practical full-scale range of DAC is (2n-1)-0 = (2n-1).
Let's define the sine wave as below.

$$V(t) = A \bullet \sin(\omega t) \tag{5}$$

where A:amplitude, ω:2пf and f:frequency.
Then (2A) should fit into the full-scale range (2n-1). Differentiating the Equation (5), the maximum slope is located at ωt=kп where k is any integer number, and it is derived as Equation (7).

$$\frac{dV}{dt} = A\omega \bullet \cos(\omega t) \tag{6}$$

$$\left.\frac{dV}{dt}\right|_{\omega t = k\pi} = A\omega \tag{7}$$

When ΔV represents the DAC 1LSB voltage and Δt represents the DAC 1 clock period, the maximum DAC voltage step occurs at ωt=kп where the curve crosses over the mid level, and it is expressed:

$$\Delta V = A\omega \cdot \Delta t \tag{8}$$

The maximum slope is high-lighted with a red line in Figure 7. As Figure 7 shows, the sine wave is symmetrical so that it is enough to consider a half cycle of sine wave.

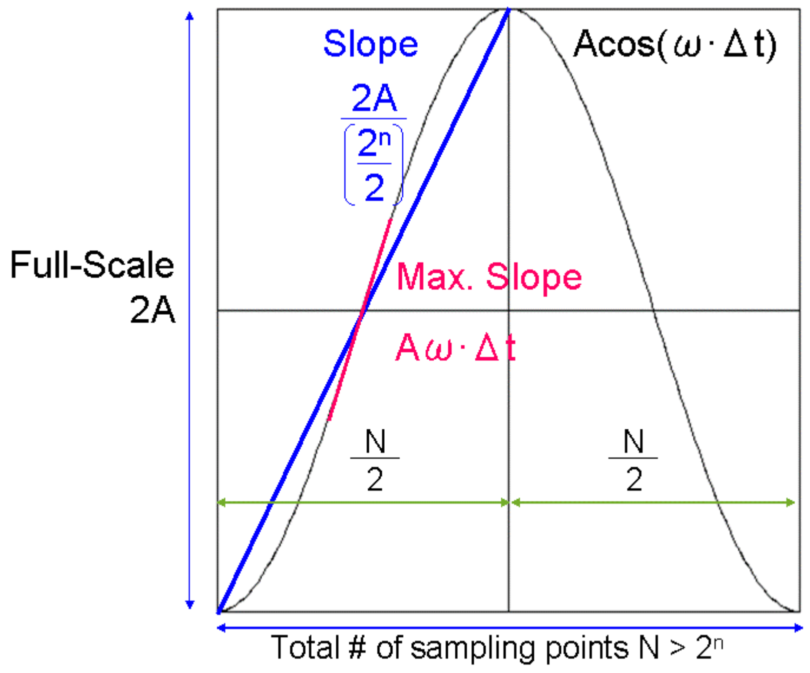
Slope
$\frac{2A}{\left(\frac{2^n}{2}\right)}$
Acos(ω·Δt)
Full-Scale
2A
Max. Slope
Aω·Δt
$\frac{N}{2}$
$\frac{N}{2}$
Total # of sampling points N > 2ⁿ

Figure 7. Full-scale sine wave.

The full-scale span 2A must be covered with minimum N/2 points of data. The average voltage step of the triangle in Figure 7 is shown with a blue line and it can be described as below.

$$\frac{2A}{\left(\frac{N}{2}\right)} = \frac{2A}{\left(\frac{2^n}{2}\right)} \qquad (9)$$

This slope should be greater than the maximum slope of the sine wave so that Equations (8) and (9) are combined as follows.

$$A\omega \bullet \Delta t \le \frac{2A}{\left(\frac{2^n}{2}\right)} \qquad (10)$$

$$\omega \bullet \Delta t \le \frac{1}{2^{n-2}} \qquad (11)$$

$$2\pi f \bullet \Delta t \le \frac{1}{2^{n-2}} \qquad (12)$$

$$\left(f = \frac{1}{N\Delta t} \right) \qquad (13)$$

$$N \ge \pi \bullet 2^{n-1} \approx 2^{n+1} \qquad (14)$$

$$\left(\pi = 3.14... < 4 = 2^2 \right) \qquad (15)$$

$$N > 2^{n+1} \qquad (16)$$

The exact result is derived as Equation (14), but N should be rounded up to 2n+1 (Equation (16)), because it is always convenient when applying FFT for some reason. The conclusion is that if your DAC is n-bit, you should create your stimulus sine wave with minimum 2n+1 points of data.

(2) Programming the DAC sine wave.

Let’s think of creating a sine wave data with an example of four-bit DAC, and check whether Equation (16) is correct or not. According to the equation, the minimum number of points should be 24+1=25=32. A single cycle of sine wave can be programmed as the code listed below.

```
    INT         I,Ndata,Ncycles;
    ARRAY_I   iWave;
    DOUBLE     dP,dO,dA;

    Ndata=32;                         //   2^(4+1)=2^5=32
    Ncycles=1;                        //   Single cycle
    dP=2.0*M_PI*Ncycles/Ndata;// Phase increment
  dA=7.5;                             // Fullscale/2=15/2
```

```
          dO=dA;                              // Offset
          iWave.resize(Ndata);
          for (i=0;i<Ndata;i++)
              iWave[i]=(INT)(dO+dA*sin(dP*i)+0.5);
```

LIST 1. DAC data programming (1).

Casting by INT with offset 0.5 is a typical practice of rounding to make it an integer. Look at Figure 8 showing the data created with this program.

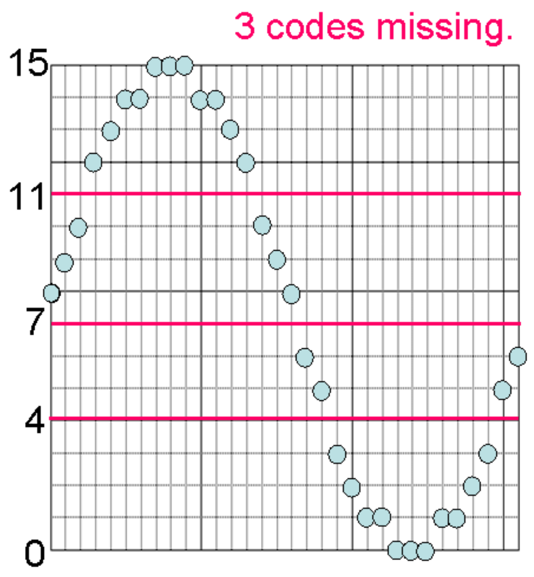

Figure 8. DAC Data programming (1).

All available codes from 0 to 15 should be created, but Figure 8 shows there are three missing codes: 4, 7 and 11. What happened? Is the conclusion of Equation (16) wrong? No, it isn't. This is a programming problem. You can see that Figure 8 is symmetrical, meaning the data hit the same levels as shown in the left graph in Figure 9. This causes the problem.

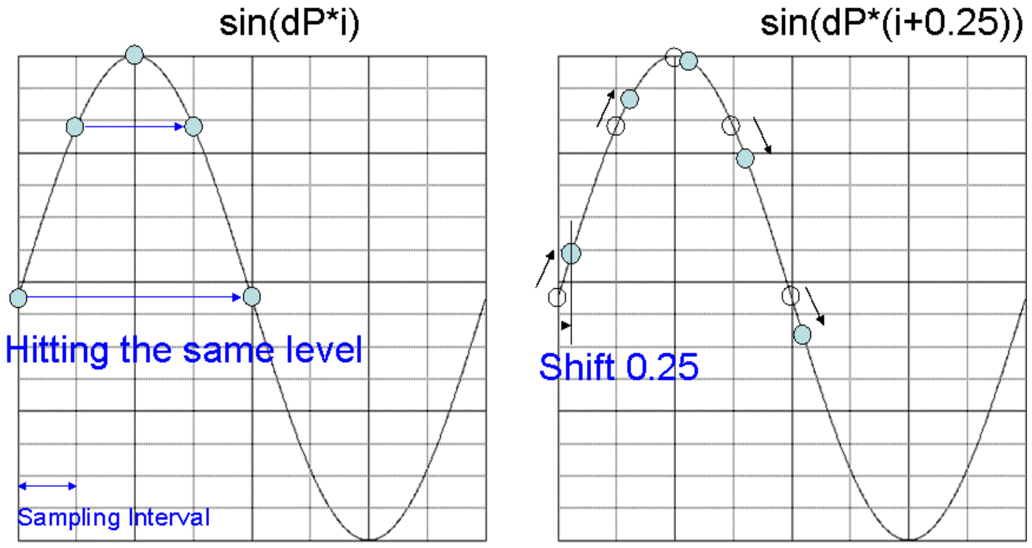

Figure 9. Phase shift 0.25.

You should modify it by applying the phase offset 0.25 in the sampling period as shown in the right graph in Figure 9. The sampling points are slightly shifted, and do not hit the same level with each other. This is the point. The source code should be modified as below.

```
INT         I,Ndata,Ncycles;
ARRAY_I     iWave;
DOUBLE      dP,dO,dA;

Ndata=32;                      //  2^(4+1)=2^5=32
Ncycles=1;                     //  Single cycle
dP=2.0*M_PI*Ncycles/Ndata;// Phase increment
dA=7.5;                        // Fullscale/2=15/2
dO=dA;                         // Offset
iWave.resize(Ndata);
for (i=0;i<Ndata;i++)
      iWave[i]=(INT)(dO+dA*sin(dP*(i+0.25)+0.5);
```

LIST 2. DAC data programming (1).

With this modification, you can successfully program all available codes of the DAC as shown in Figure 10.

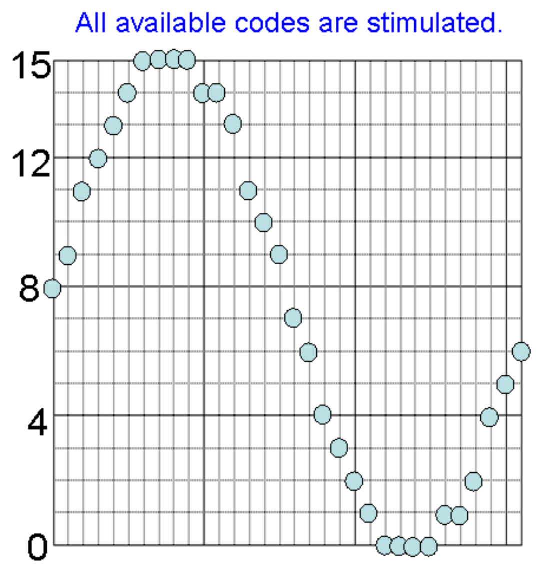

Figure 10. DAC data programming (2).

Here is a tip for creating DAC code appropriately with a minimum number of codes. When you create DAC stimulus codes for regular DAC tests, check to see whether you stimulate all available codes or not.

To be continued.