VERIGY

Application Note

Method for Calculating SNR for Non-integer Cycle Sine Wave

Koji Karube
Verigy Japan

Table of Contents

Introduction

Sine waves are the most fundamental waveform used in analog tests. In the mixed-signal IC testers which use DSP (digital signal processing), they are used to calculate test parameters such as THD (total harmonic distortion), SNR (signal-to-noise ratio) and frequency characteristics with FFT (fast Fourier transform). However, if the frequency of the measured sine wave is slightly different from expected frequency, and the number of cycles of the captured waveform ceases to be an integer, then FFT processing will cause leakage and the measurement accuracy will decline.

In general, when a waveform with a non-integer cycle is introduced to FFT, the ends of the measured waveform are taken as zero to eliminate leakage from joint of waveform when regarded as a repeated waveform, then window processing is performed in the time domain before FFT processing. The most common type of window is the Hanning window, but Hamming, Bartlett (Triangular), Blackman-Harris, and Flat-top windows also exist. Each has different characteristics and is used for a different purpose. If the object is to determine level, a flat-top window which offers poor frequency resolution but high level accuracy is used. If dynamic ranges such as SNR are to be emphasized, a Blackman-Harris window is used.

DTMF is the example

As the complexity of ICs increases, so does the frequency of cases in which mixed-signal IC testers must use non-integer cycle sine waves as the measured signal. Tests of ICs for telecommunications equipment in particular frequently run into this problem. A typical example is the measurement of tone-phone dial signals (dual-tone multi-frequency, or DTMF signals). DTMF signals are recommended under CCITT (ITU-T) Q.23 *(www.itu.int)*. They consist of a dual tone composed of one of four frequencies from the low group and one of four frequencies from the high group shown below. The frequency groups are as follows:

```
Low-group frequencies:  697, 770, 852, 941 Hz ±1.8 %
High-group frequencies:  1209, 1336, 1447, 1633 Hz ±1.8 %
```

Even so, if all the frequencies are to be measured in units of integer cycles, the observation time must be one second. However, given the price of DTMF ICs, an observation time of one second leads to prohibitively high test costs. Moreover, the signal output from the actual IC differs slightly from those of the recommendations and even if an observation time of one second is used in measurements, it is not possible to obtain a waveform with an integer cycle. Figure 1 shows the hypothetical block diagram of the IC where the DTMF signal is produced. In both the low group and high group a crystal oscillation frequency of 10.24 MHz is divided, and 128 points of sine wave data recorded on a ROM are read and the DTMF signal is synthesized using D/A converter output. The actual frequencies in this situation are as follows:

```
697 Hz:  10.24 MHz / 115 / 128 points = 695.6521739 ... Hz
770 Hz:  10.24 MHz / 104 / 128 points = 769.2307692 ... Hz
852 Hz:  10.24 MHz /  94 / 128 points = 851.0638298 ... Hz
941 Hz:  10.24 MHz /  85 / 128 points = 941.1764706 ... Hz
1209 Hz:  10.24 MHz / 66 / 128 points = 1212.121212 ... Hz
1336 Hz:  10.24 MHz / 60 / 128 points = 1333.333333 ... Hz
1447 Hz:  10.24 MHz / 57 / 128 points = 1403.508772 ... Hz
1633 Hz:  10.24 MHz / 49 / 128 points = 1632.653061 ... Hz
```

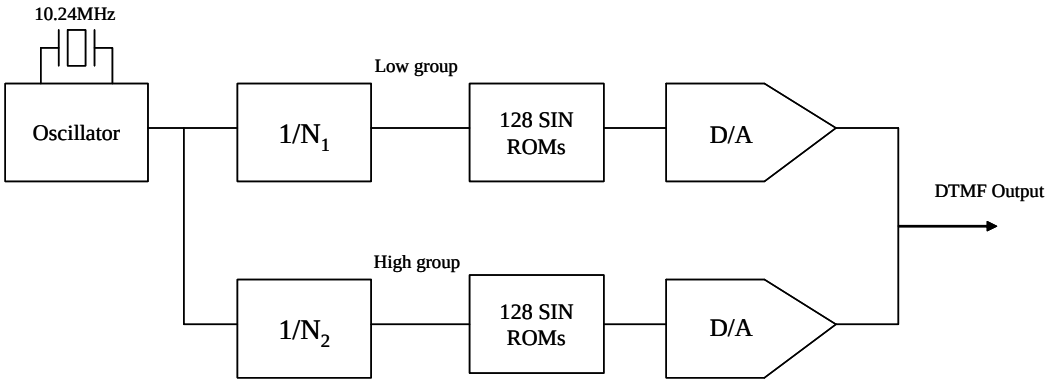

Figure 1. Mechanism By Which DTMF Signals Are Produced Inside The IC.

In the old generation of mixed-signal IC testers which did not use DSP, selective level meters and other devices were used to measure the signal levels for the low and high groups respectively. Using a notch filter to eliminate the two fundamental wave components, and inputting the noise component to a voltmeter, made it possible to measure the noise component with great accuracy but the measurement time required was long. Filter design should allow similar effects in signal processing, but even if a narrow band filter could be realized by calculation, such a long settling time would be required that the advantages of DSP would be small. Moreover, as in the previously mentioned case of the DTMF signal, even if the frequencies are specified, because the method of realizing the interior mechanism of the IC will lead to slight differences in the frequency, narrow band filters would offer poor adaptability in relation to the labor involved in their development. It was therefore decided to adopt a calculation method that would allow automatic alignment and the elimination of the fundamental wave component.

At one time, a method became popular in which FFT was replaced by chirp-Z to resolve the complex limitations relating to frequencies in tests, but chirp-Z was only designed to relieve the restriction of the FFT of the Nth power of 2 and had no effect on non-integer cycle signals. It is likely that it was almost never used in tests where it was sufficient simply to confirm the frequency and level of signals such as DTMF. If frequency and level must be accurately determined, under normal circumstances the measurement time is long in relation to the accuracy obtained, but Tabei, et al. have proposed an extremely effective formula for this task. [1]

Detail of the method

Let us briefly discuss the method of determining the frequency and level of DTMF signals using this formula. By applying a Hanning window and FFT to a DTMF signal digitized at 16 KHz and 256 points as shown in Figure 2, one obtains the spectrum shown in Figure 4. A comparison of Figure 3, in which FFT has been applied without the use of the window, reveals that in Figure 4, the DTMF signal is surrounded by the side-lobes typical of the Hanning window. But if the peak value of the spectrum in Figure 4 and the second highest peak value to its left or right are inserted into the formula, the true frequency and level can be derived. Thus, even if the measurement time is approximately 16 ms as in Figure 2, the frequency and level can be accurately determined even at frequencies of below 1 Hz.

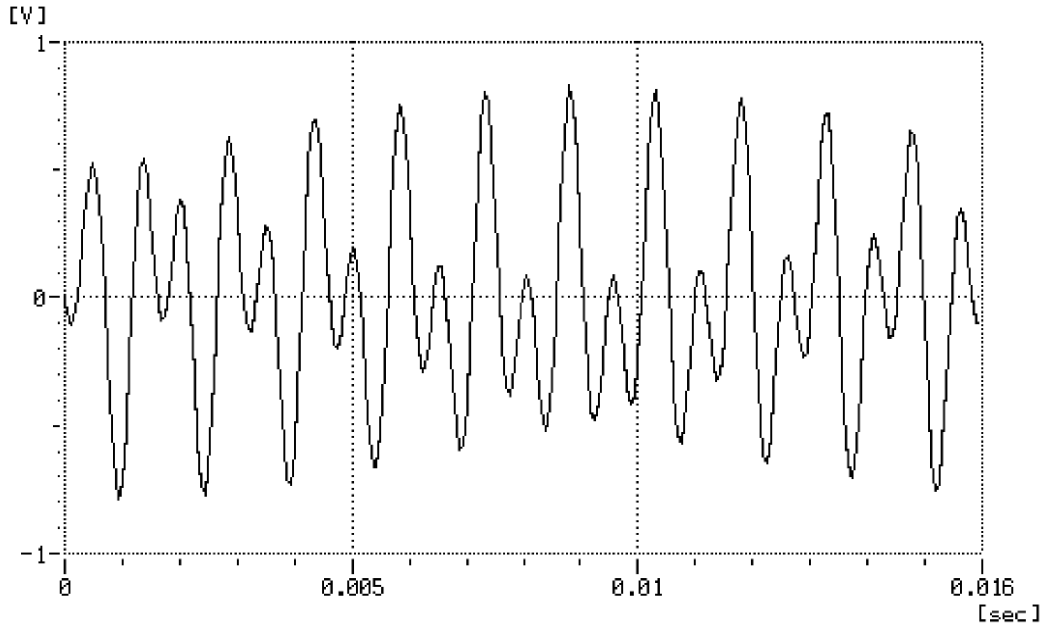

Figure 2. DTMF Signal (fs = 16KHz, 256 points).

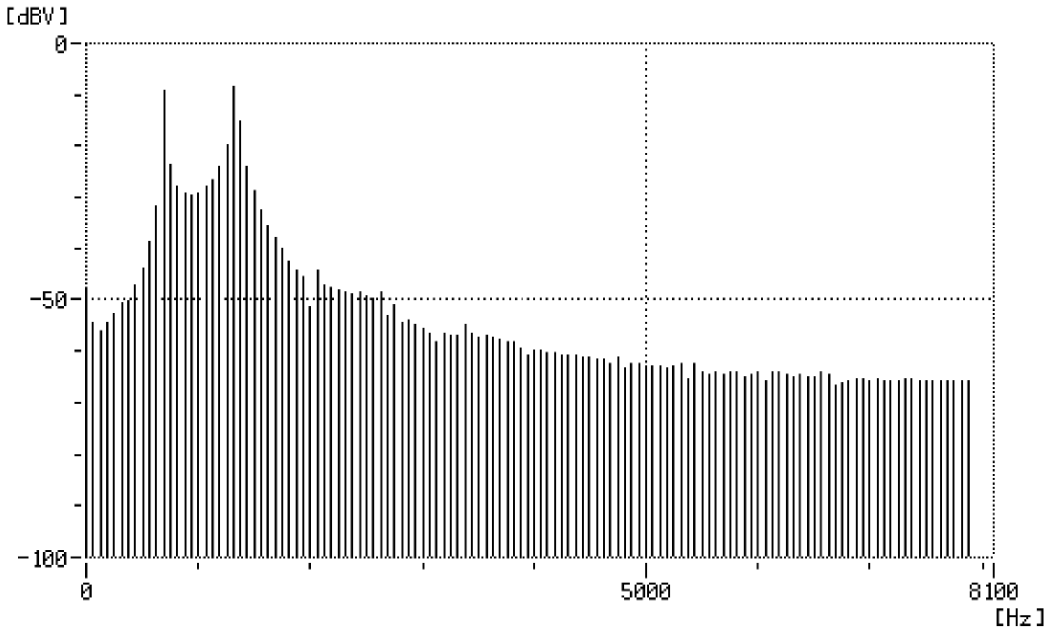

Figure 3. FFT Applied to DTMF Signal Without Window.

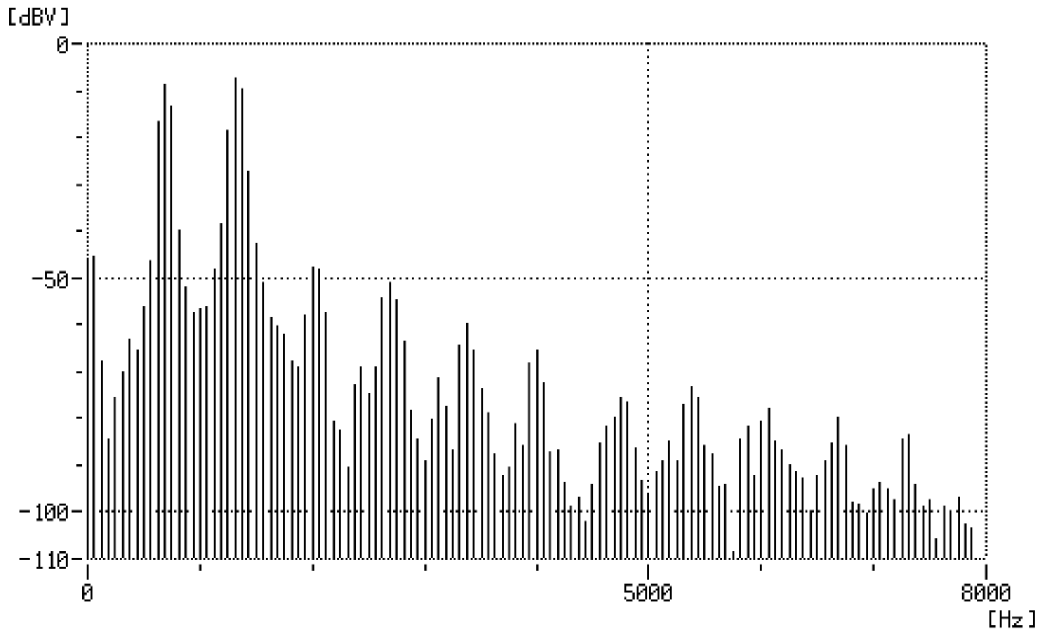

Figure 4. FFT Applied to DTMF Signal With Hanning Window.

Finally, this formula using the characteristics of the Hanning window also allows precise calculation of the phase of the target signal, so that given the structure of the subtraction-type distortion meter in Figure 5, we judged that it might be possible to perform the previously mentioned SNR measurement by automatic alignment.

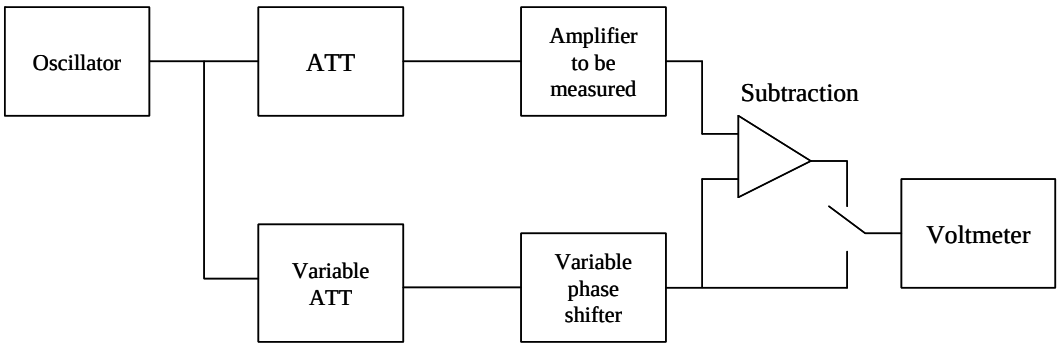

Figure 5. Subtraction-Type Distortion Meter.

In conventional distortion meters, a notch filter is used to eliminate the fundamental wave. Here, an attenuator and phase device are adjusted to produce the same waveform as the fundamental wave component of the measured signal. The produced waveform is subtracted from the measured signal, leaving as residual components the harmonics and noise, so that voltage can be measured to obtain the distortion. As shown in Figure 6, if the procedure for adjustment of the attenuator and phase device are applied to the previously mentioned formula, the distortion factor -- in other words the SNR -- can be obtained. However, Figure 6 shows the results for processing a single tone and if, as in the previously mentioned DTMF signal, the waveform includes several signal components, then the number of supplementary formula blocks and blocks which reproduce the waveforms by cosine are increased in proportion to the number of signal components to be eliminated. Moreover, it is essential

that, after the fundamental wave is removed, the voltage be calculated without FFT and using the root mean square (rms) of the time domain (in practice, sigma is substituted to eliminate the DC component). If FFT is used, it will be seen from the side lobes in Figure 4 that the voltage value increases. Moreover, since the phase obtained from the supplementary formula is the real part, the waveform is produced using cos rather than sin.

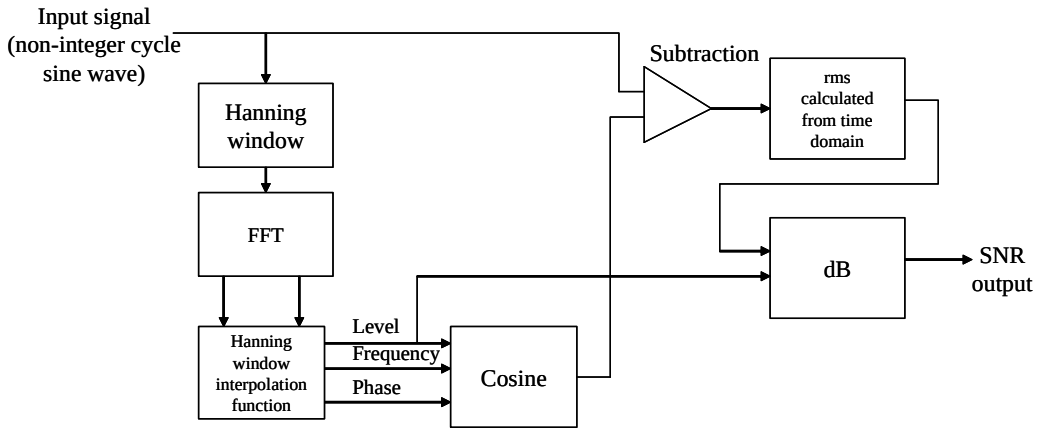

Figure 6. Measurement of SNR of a Non-Integer Cycle Sine Wave.

SNR Calculation Method for a Non-Integer Cycle Sine Wave

The calculation procedure is as follows:

1. Estimate level, frequency, and phase of fundamental signal by performing FFT with Hanning window.
2. Recover pure fundamental sine wave.
3. Subtract pure recovered fundamental wave from input signal.
4. Calculate noise rms in time-domain.
5. Calculate SNR using results of steps 1 and 4.

This SNR calculation method has the following features:

- Can get accurate SNR value for asynchronous distorted signal.
- Can use rough frequency-resolution to shorten test time.
- Can ignore leaked energy from fundamental frequency, principle harmonics, and unexpected spurious signals.

References

[1]. Makoto Tabei and Mitsuhiro Ueda, *A Method of High Precision Frequency Detection with FFT*, IEICE (Japan) Vol. J70-A No.5 pp.708-805 1987.5

[2]. R.parker and S.A.Stoneman, *On the Use of Fast Fourier Transform When High Frequency Resolution is Required*, J. Sound and Vibration, 104, 1,pp.75-79 (1986)

[3]. J.P. Burg, *Maximum entropy spectral analysis, Paper presented at the 37th Annual International Meeting*, Soc. of Explor. Geophys., Oklahoma City (1967)

[4]. S.M. Kay and S.L. Marple Jr, *Spectrum analysis-A modern perspective*, Proc. IEEE, 69, 11, pp.1380-1419 (1981)

Calculation Program

The following shows an example function program for calculating SNR for a non-integer cycle sine wave.

```
int async_SNR(ARRAY_D& wave, DOUBLE fdig, DOUBLE faim, DOUBLE* freq, DOUBLE* level, DOUBLE* dist) {
/*****************************************************************************
 * ARRAY_D& wave : Input Parameter: Digitized input waveform
 * DOUBLE fdig   : Input Parameter: Digitizing frequency
 * DOUBLE faim   : Input Parameter: Target Frequency
 * DOUBLE* freq  : Output Parameter: Frequency of detected sine wave
 * DOUBLE* level : Output Parameter: Level of detected sine wave
 * DOUBLE* dist  : Output Parameter: SNR of detected sine wave
 *****************************************************************************/

   ARRAY_D noise_wave;
   ARRAY_COMPLEX Spect;
   int i, index, index1, index2;
   DOUBLE fstep, s1, s2, a, b, c, d;
   ARRAY_COMPLEX e(1);
   ARRAY_D f(1), g(1);
   double phase, phase_step;
   double mean, sigma, noise;

/*****************************************************************************
 * In preparation for estimation of frequency/level using the Hanning
 * window, the index on the spectrum array closest to the target fundamental
 * wave and the adjoining index are obtained.
 *****************************************************************************/
   fstep = fdig / (DOUBLE)(wave.size());
   index1 = (int)(faim / fstep);
   index2 = index1 + 1;

/*****************************************************************************
 * FFT is applied via the Hanning window and the power is obtained from
 * the average of the squares of the real and imaginary parts for each of
 * the two indexes.
 *****************************************************************************/
   Spect.resize(wave.size()/2);
   DSP_FFT(wave, Spect, HANNING);

   s1 = sqrt(Spect[index1].real() * Spect[index1].real() + Spect[index1].imag() * Spect[index1].imag());
   s2 = sqrt(Spect[index2].real() * Spect[index2].real() + Spect[index2].imag() * Spect[index2].imag());

/*****************************************************************************
 * Data are separated into cases according to the ratio of the size of
 * the powers of the tow indexes, a formula approximating the characteristics
 * of a Hanning window is applied and accurate estimates are made of
```

```
 * the frequency and level of the target fundamental wave.
 ********************************************************************************/
   if ((s1 != 0.0) && (s2 != 0.0)) {
      if (s1 < s2) {
         index = index2;
         a = s1 / s2;
         b = (1.0 - 2.0 * a) / (1.0 + a);
         c = -s2 * M_PI * b / sin(M_PI * b) * (b - 1.0) * (b + 1.0);
         d = (DOUBLE)index + b;
      }
      else {
         index = index1;
         a = s2 / s1;
         b = -(1.0 - 2.0 * a) / (1.0 + a);
         c = -s1 * M_PI * b / sin(M_PI * b) * (b - 1.0) * (b + 1.0);
         d = (DOUBLE)index + b;
      }
      *level = 20.0 * log10(c / M_SQRT2);
      *freq = d * fstep;
   }
   else {
      *level = -3.40282347e+38;
      *freq = 0.0;
   }

/*******************************************************************************
 * In the same way, the formula approximating the characteristics of
 * the Hanning window is used to make an accurate estimate of the phase in
 * the cosine of the target fundamental wave.
 ********************************************************************************/
   e[0].real() = Spect[index].real() * cos(M_PI * b) + Spect[index].imag() * sin(M_PI * b);
   e[0].imag() = -(Spect[index].real()) * sin(M_PI * b) + Spect[index].imag() * cos(M_PI * b);
   DSP_RECT_POL(e, f, g);

/*******************************************************************************
 * Conversion to polar coordinates, input of one point in "e" array,
 * output of gain in "f" array and phase in "g" array.
 ********************************************************************************/
   phase = g[0];

/*******************************************************************************
 * The fundamental wave estimated above is subtracted from the original signal
 * leaving a signal with high frequency and noise only, and the rms value
 * without the DC component is calculated from the sigma of the noise data.
 ********************************************************************************/
   phase_step = *freq / fdig * 2.0 * M_PI;
   noise_wave.resize(wave.size());
   for (i = 0; i < wave.size(); i++) {
      noise_wave[i] = wave[i] - c * cos(i * phase_step + phase) - Spect[0].real();
   }
   DSP_MEAN(noise_wave, &mean, &sigma);
   noise = sqrt((double)(wave.size() - 1) / (double)(wave.size())) * sigma;

/*******************************************************************************
 * From the high frequency wave, the rms value of the noise and the level
 * of the fundamental wave, an SNR without leakage is obtained.
 ********************************************************************************/
   if ((c > 0.0) && (noise > 0.0)) {
      *dist = 20.0 * log10((double)(noise / (c / M_SQRT2)));
   }
   else {
      *dist = 3.40282347e+38;
   }

   return 0;
}
```