# Hideo Okawara's
# Mixed Signal Lecture Series

# DSP-Based Testing – Fundamentals 7
# Coherent Condition

*Verigy Japan*

*November 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.

## 1. Coherent Condition

In mixed signal testers or the DSP-based testing, test signal waveforms are digitized with a waveform digitizer/sampler or an A/D converter, and the captured signal is processed with DFT/FFT basically, and a frequency spectrum is created to point out the particular signal components. DFT/FFT is the main tool to analyze signals. The most important factor in the DSP-based testing is coherent condition, which is the theme of this article.

## 2. Whole Number of Cycles in the UTP

For the successful Fourier transform, the test signals are supposed to be infinitely continuous. The unit test period (UTP) is a choice cut of the continuous test signal. Figure 1 shows that a unit of captured waveform is concatenated serially three times. The total waveform looks continuous in (a), but the waveform (b) does not. The UTP in (a) exactly captures a whole number cycles of the signal. However, in (b) the UTP contains a fractional cycle. (*M* in the pictures is the number of cycles in the UTP.)

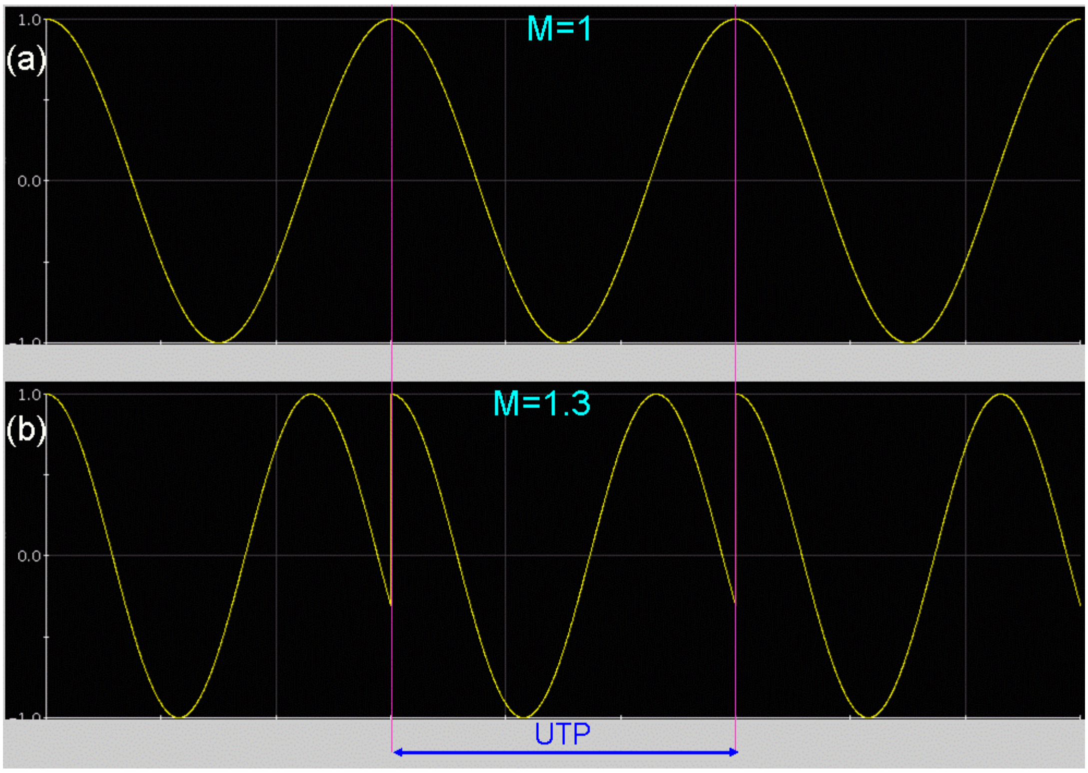


**Figure 1: Continuous Waveform**

Figure 2 shows the two waveforms, which are captured data of 8-bit AD converters. Two kinds of 5 MHz signals are captured at the sampling rate of 110 Msps. It captures 512 points of data. There are two lines in the graph. The frequency of the signal colored green is 4.941406...MHz, and exactly 23 cycles of the signal matches the UTP. On the other hand, the signal colored yellow is exactly 5 MHz so that the UTP cannot contain the whole number cycles of the signal.

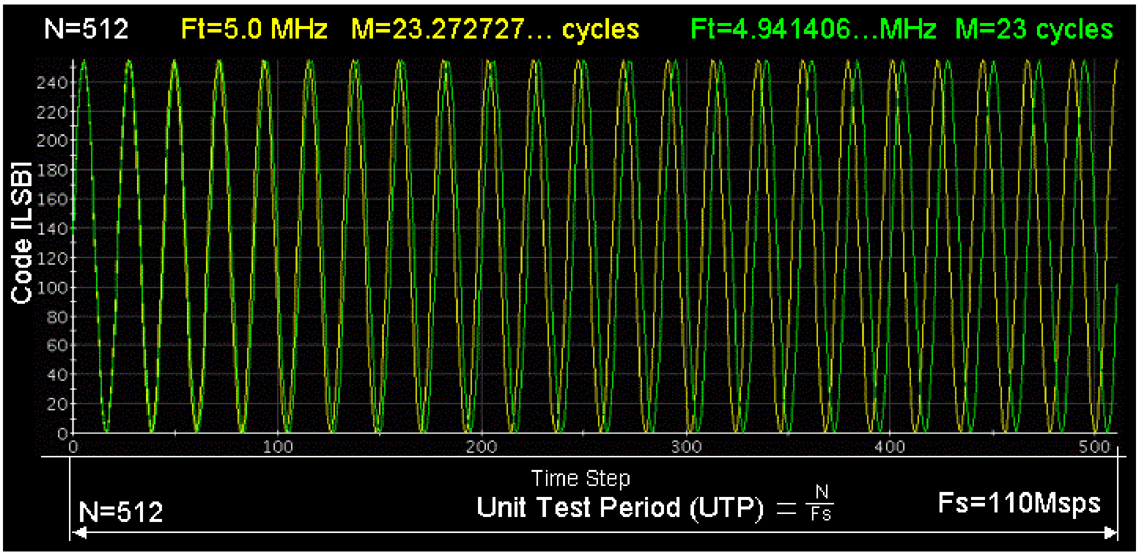


**Figure 2: 8-bit ADC Output Waveforms**

When FFT is applied to each waveform, the frequency spectra are displayed in Figure 3. The color matches Figure 2. The green spectrum clearly shows the fundamental component and the quantization noise. However, the yellow one shows abnormal shape.

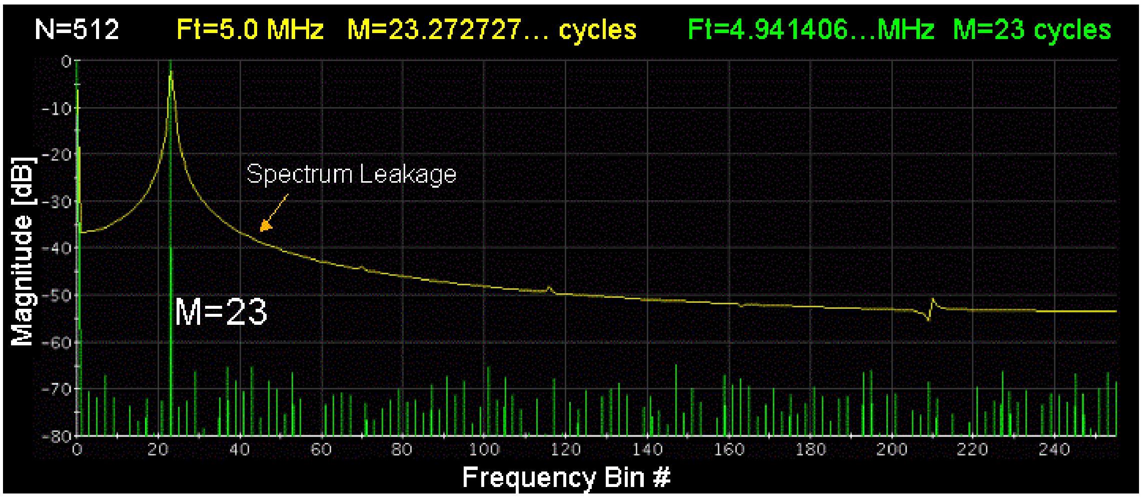


**Figure 3: Spectrum of the Waveforms**

This is a significant effect of the fractional cycles of the captured waveform. The Fourier transform is based on the infinitely continuous waveform. The final data point of the waveform should be smoothly continuous to the start point of the waveform. If there is discrepancy between them, the spectrum would be smeared.

If you would like to get a beautiful frequency spectrum in the DSP-based testing, you must set up your test condition strictly coherently. A whole number of signal cycles should exactly fit the UTP. It can be expressed as the simple equation as follows.

$$\underbrace{\frac{F_t}{F_s}}_{\substack{\text{Test Signal Frequency} \\ \text{Sampling Frequency}}} = \underbrace{\frac{M}{N}}_{\substack{\text{Number of Cycles} \\ \text{Number of Points}}} \qquad (1)$$

This equation shows the fundamental coherent condition. M and N must be integer numbers, and M and N must be "mutually prime" or they must not have common divisors. For FFT restriction the number of points N is usually set as $2^n$ so that you may settle M as an appropriate odd number. This is definitely important condition so that you can often find this equation here and there.

Figure 4 shows that 65 cycles of sinusoidal waveform is captured with 512 points of data. (The upper picture) "65" and "512" have no common divisors. There is a very handy DSP API available to check if it is a good coherent condition or not.

**DSP_SHUFFLE(InputWaveform,OutputWaveform,Ncycles);**

This API reshuffles the input data according to the number of cycles, and reorders each one of the data points based on the primitive phase of the sine waveform. It creates a single cycle of the waveform as the lower picture of Figure 4, which shows that each one of the data points looks distributed all over the sinusoidal trace within the range from 0 to 255. If FFT processes the waveform, the spectrum appears as in Figure 5.

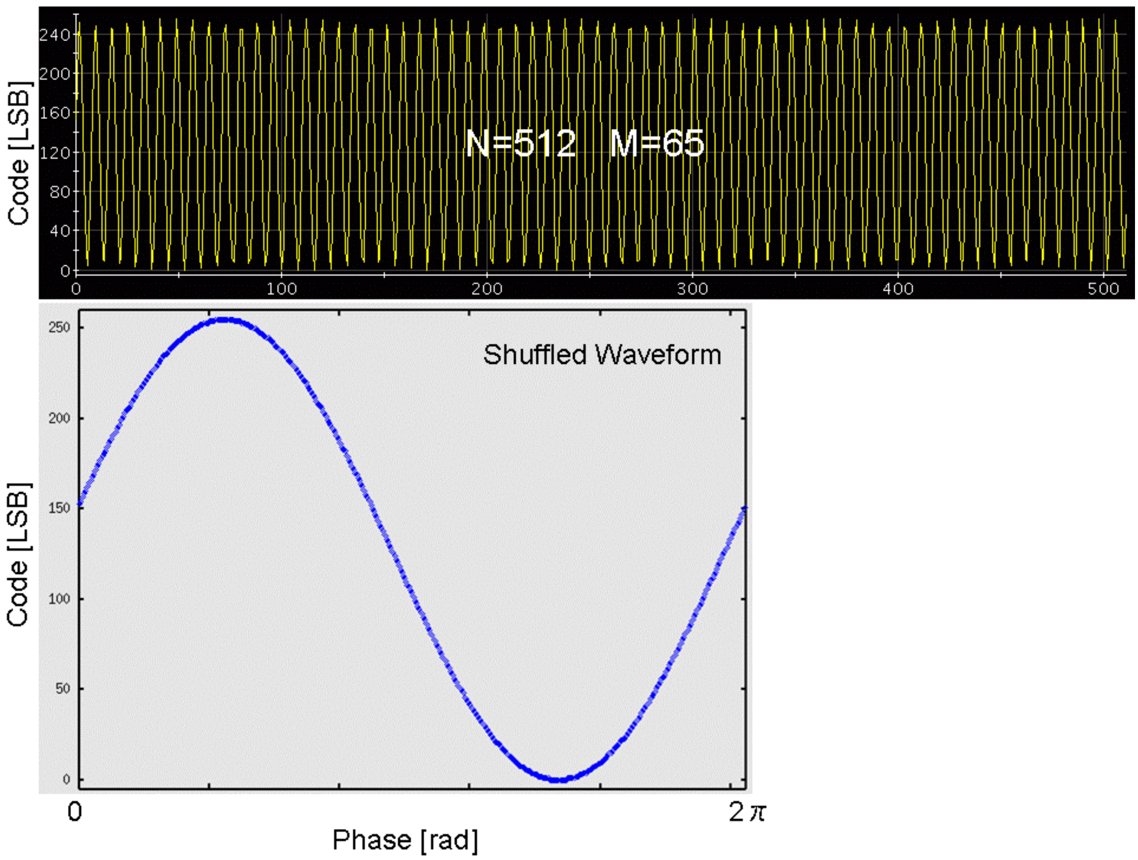

**Figure 4: Good Coherent Condition**

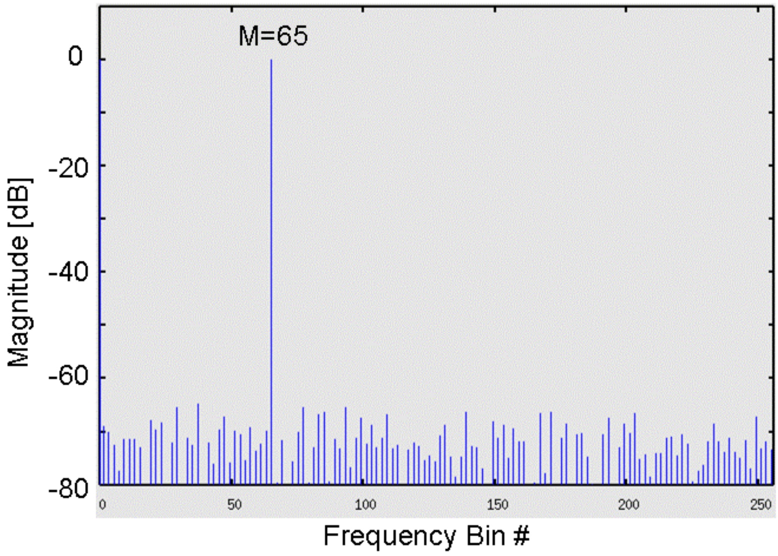


**Figure 5: Spectrum with Good Coherent Condition**

On the other hand, when 64 cycles of sinusoidal waveform are captured with 512 points of data, "64" and "512" are not mutually prime. Figure 6 shows the waveform (upper picture) and its reshuffled data (lower picture). As you notice, the reshuffled waveform hits only 8 particular codes out of 256 codes multiple times. (512/64=8)

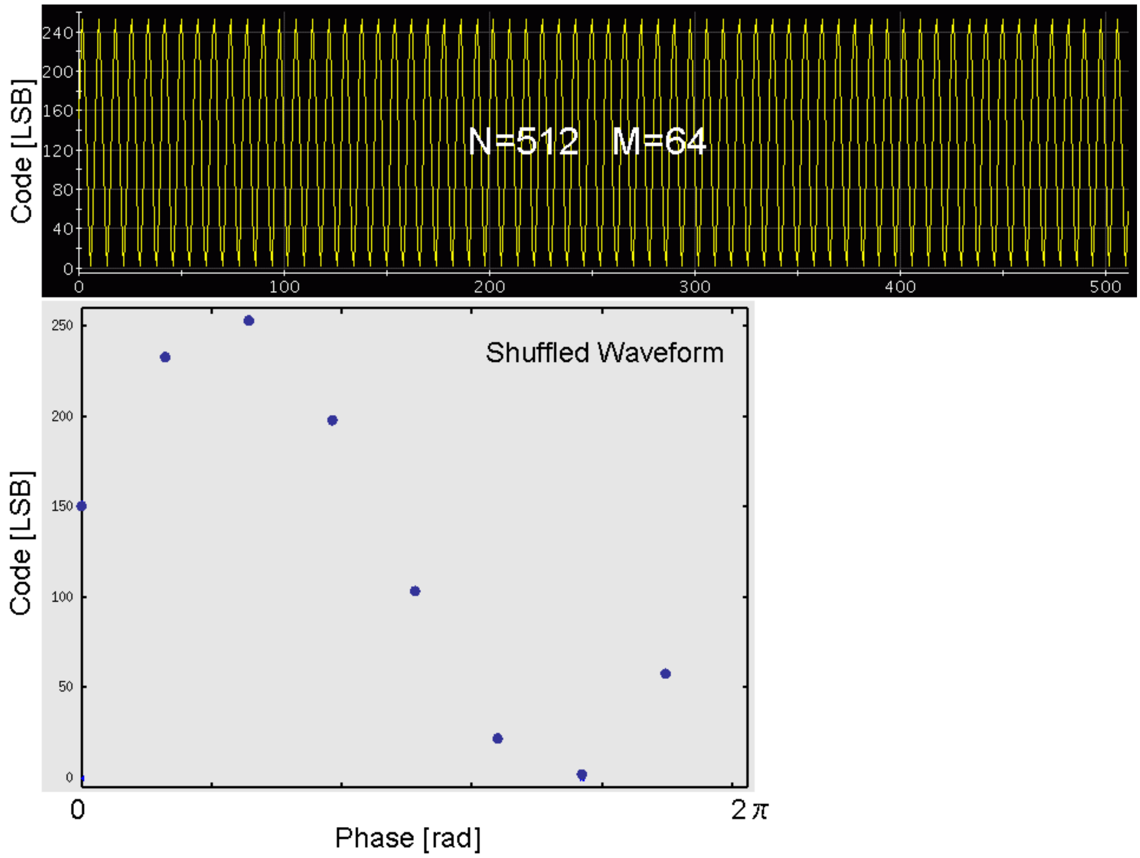

**Figure 6: Wrong Coherent Condition**

When applying FFT to this waveform, the frequency spectrum appears as in Figure 7, which is an extremely abnormal spectrum. All the quantization noise concentrates on the single bin. This comes from the violation of the rule of M and N mutually prime. When you design your test condition, you should make up the coherent condition carefully.

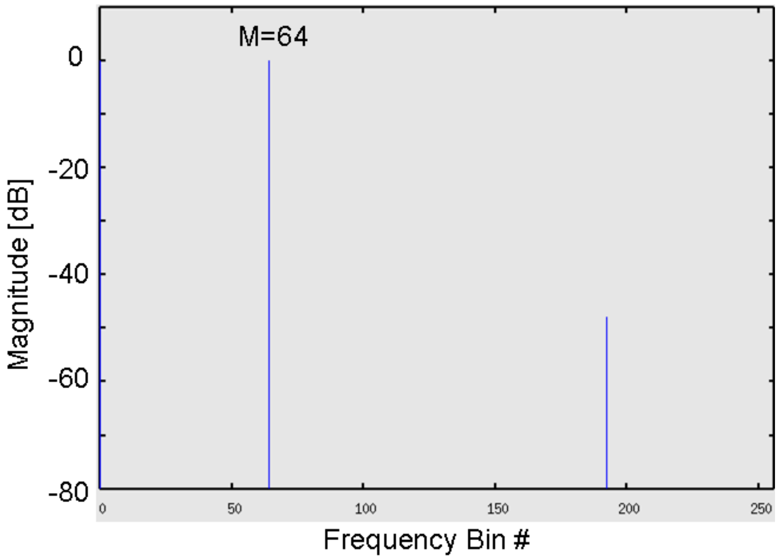


**Figure 7: Spectrum with Wrong Coherent Condition**

## Exercise -- Distinctive Spectrum

You will see various strange spectrum appearances in this section. Consider what the cause of the problem may be.

The model device is an ideal 8-bit ADC, which takes 2048 samples of data. At first you see the reference spectrum and the waveform in Figures 7 and 8.

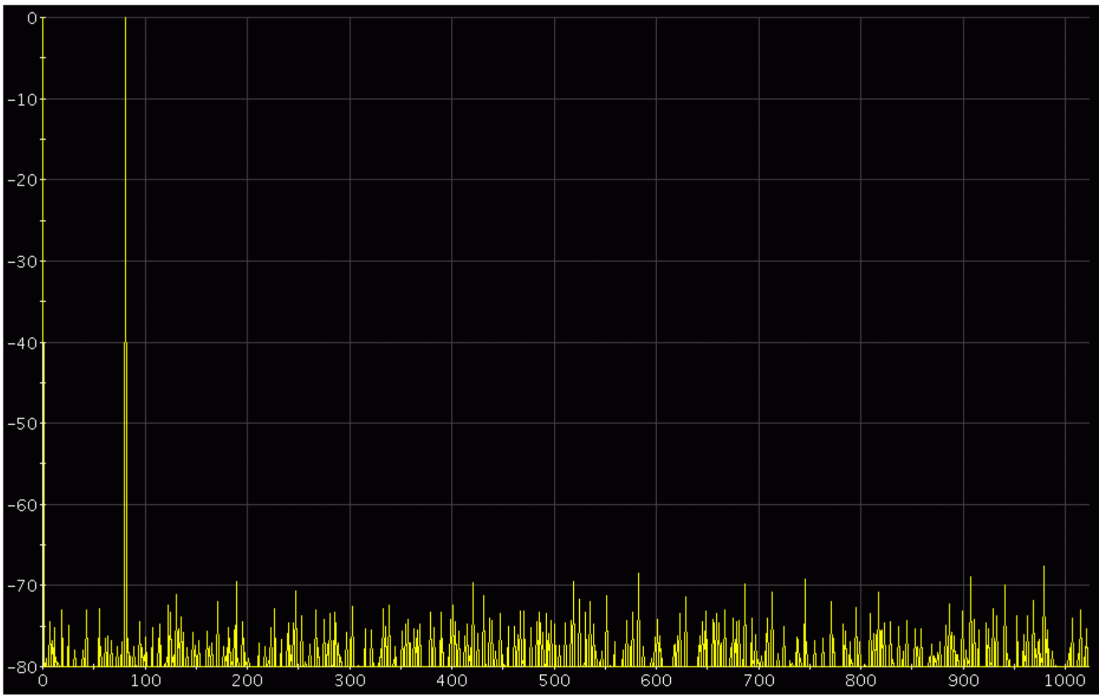


**Figure 7: Reference Spectrum**

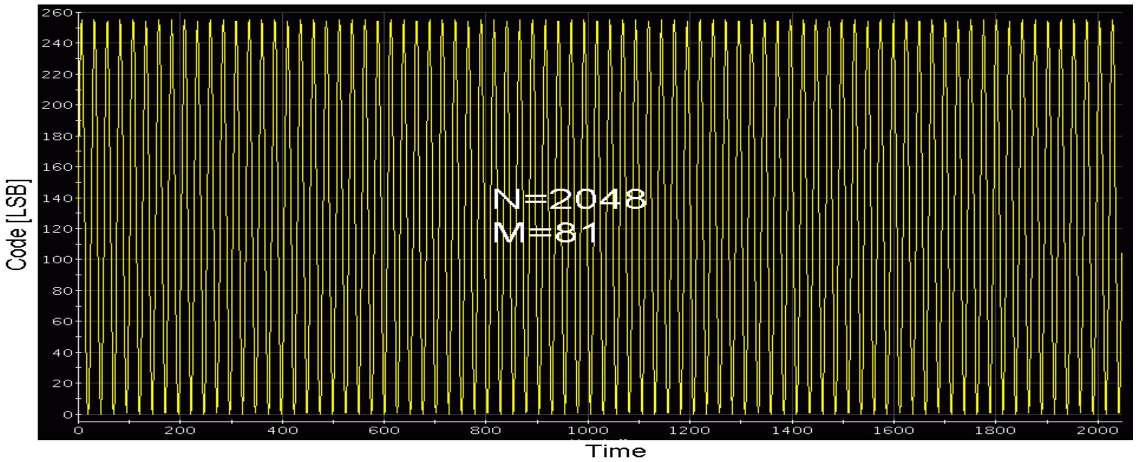


**Figure 8: Reference Waveform**

**[Problem 1]**

Figure 9 is the first problem spectrum. What do you think is the cause of the problem? I believe this is the easiest question for the readers.

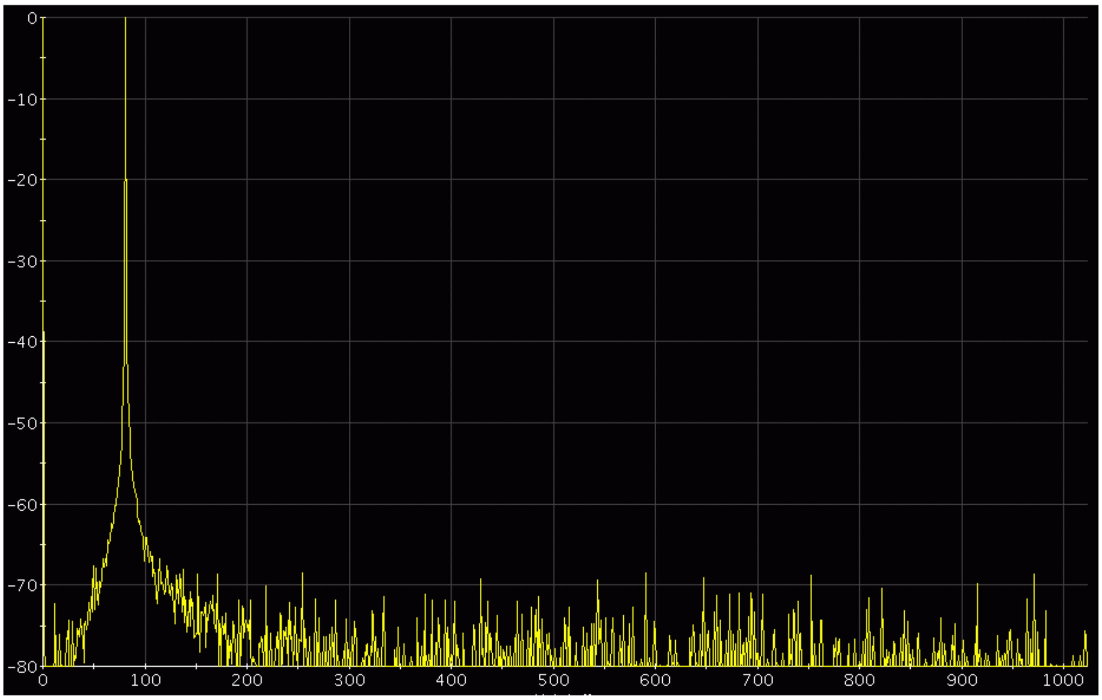


**Figure 9: Abnormal Spectrum (1)**

**[Answer 1]**

The fundamental component has smeared skirt. This is a very typical phenomenon with the imperfect coherency problem. Actually the waveform for this spectrum is constructed by the number of cycles M=81.01 cycles. This is a typical violation of an integer number M.

**[Problem 2]**

The spectrum in Figure 10 contains heavy harmonics components. What do you think is the cause of the problem?

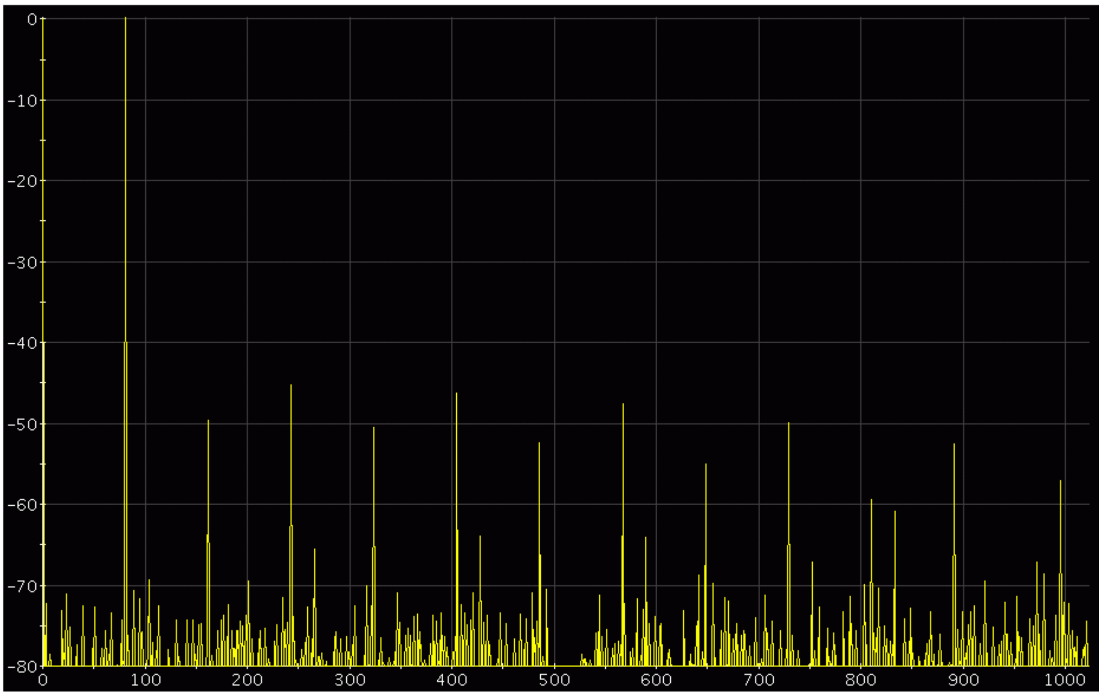


**Figure 10: Abnormal Spectrum (2)**

**[Answer 2]**

In general, harmonics mean that the waveform is distorted. The original waveform is shown in Figure 11, which does not show clearly that the waveform is distorted.

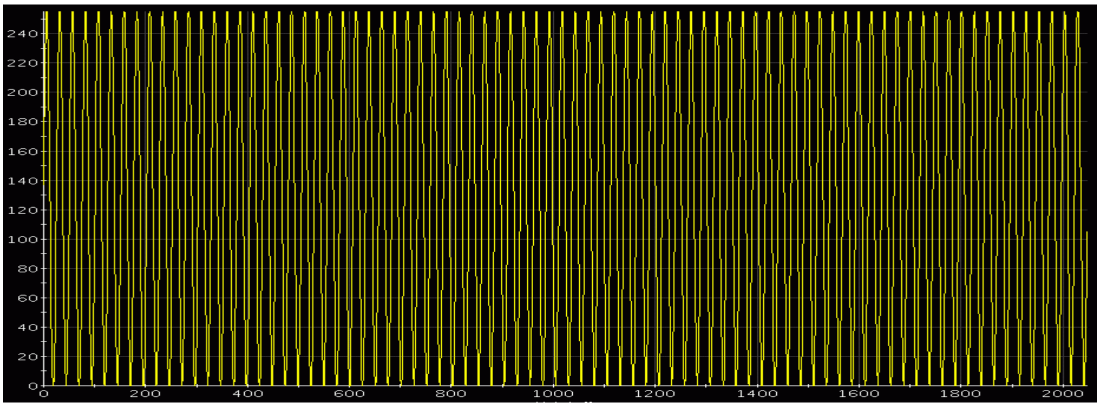


**Figure 11: Waveform (2)**

Let's play DSP_SHUFFLE(M=81) in the case like this. (Figure 12)


**Figure 12: Reshuffled Waveform (2)**

Now you can clearly find out the problem point -- the saturation or overload of the signal.

**[Problem 3]**

This may be slightly more difficult. The noise floor of Figure 13 is a little higher than the reference Figure 7.

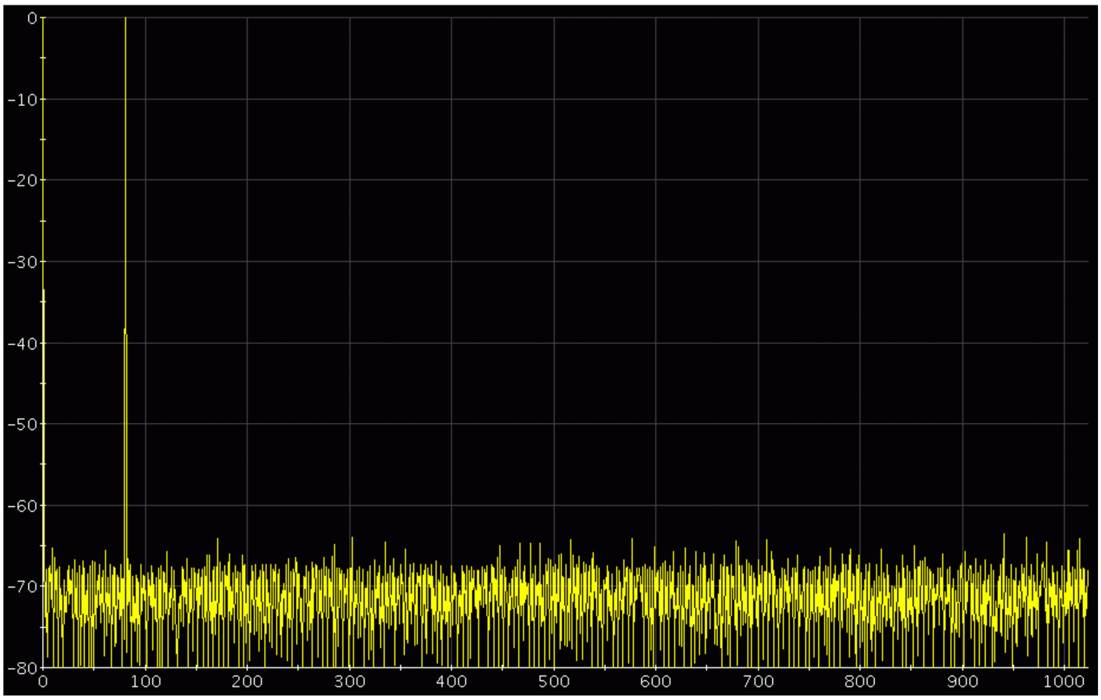


**Figure 13: Abnormal Spectrum (3)**

**[Answer 3]**

The waveform of this spectrum is shown in Figure 14, and it looks normal as is.

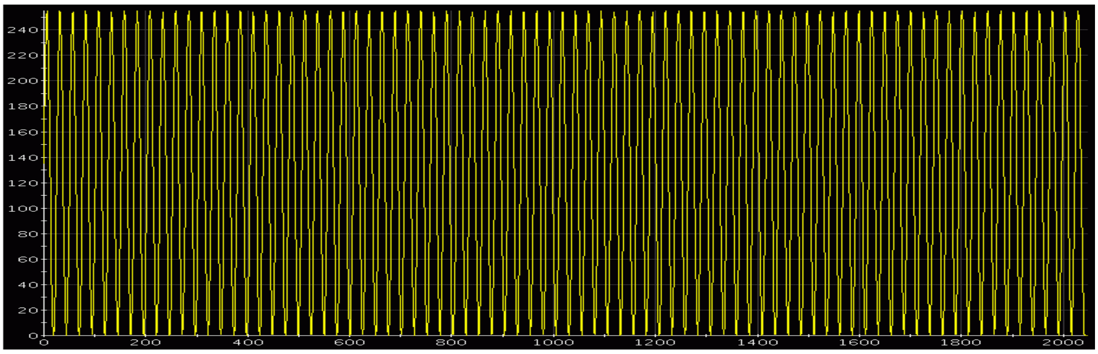


**Figure 14: Waveform (3)**

Play the conventional approach -- DSP_SHUFFLE(M=81).


**Figure 15: Reshuffled Waveform (3)**

Now you can clearly understand the problem. The captured waveform has a couple of defects for some reason. You would need to make further analysis about what made the defects.

By the way, the noise floor based on the quantization noise can be described as the theoretical SNR and the number of samples.

$$\text{Noise\_Floor [dB]} = (6.02*n+1.76) + 10*\log(N/2) \qquad (2)$$

where N is the number of samples. The first term is the famous signal to quantization noise ratio (SNR) and the second term may be called the noise improving factor (NIF). NIF shows the effect of over-sampling. It means the more data you capture, the lower the noise floor becomes. In the discrete signal environment, you can estimate the theoretical noise floor level. In actual measurements, usually you cannot reach the theoretical level because of various noise and spurs. If your test result would get a larger dynamic range than the theoretical number, it means there might be some problem in your total test method or a trick in the method. See the previous *go/semi* newsletter for more about "over-sampling".

**[Problem 4]**

Let's look at Figure 16, which shows a strange noise floor. What do you think the cause of the problem is? Probably there would be some defects in the captured waveform too.

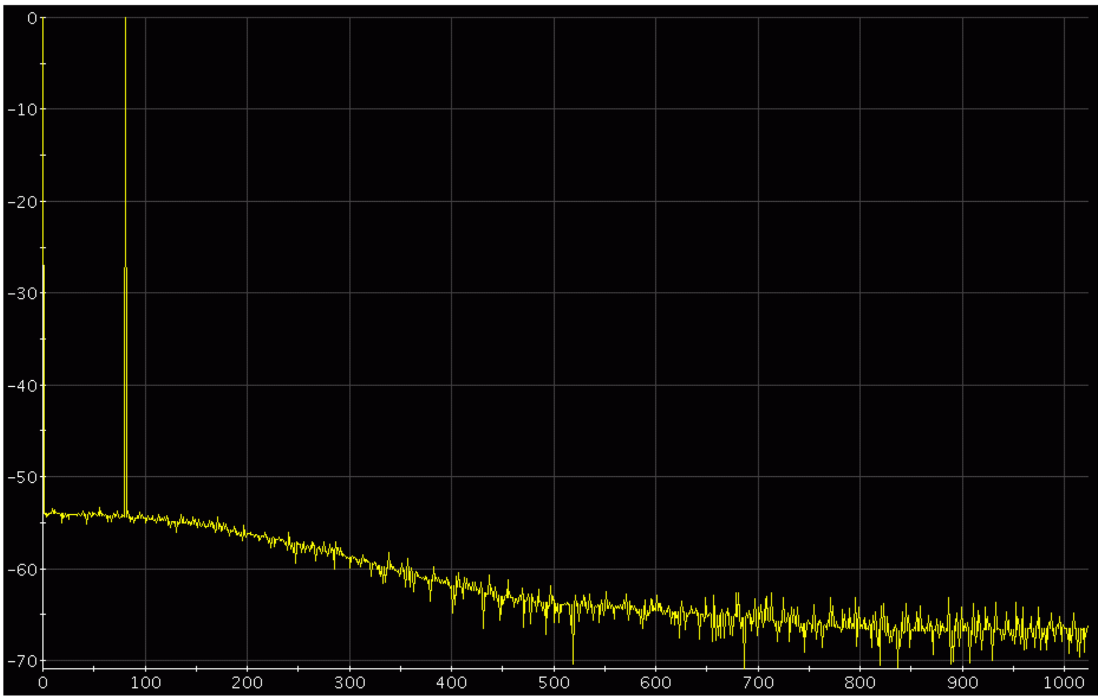


**Figure 16: Abnormal Spectrum (4)**

**[Answer 4]**

Let's check the waveform. (Figure 17)

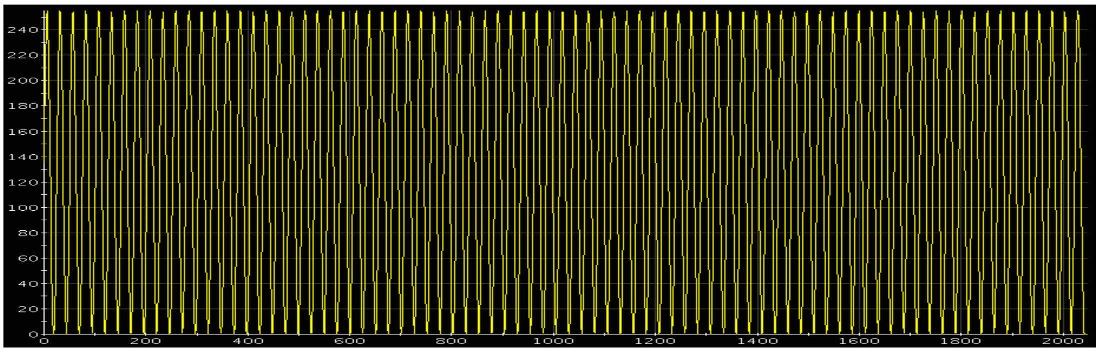


**Figure 17: Waveform (4)**

It seems to be normal at a glance. As a good practice, let's apply DSP_SHUFFLE(M=81), deriving Figure 18.

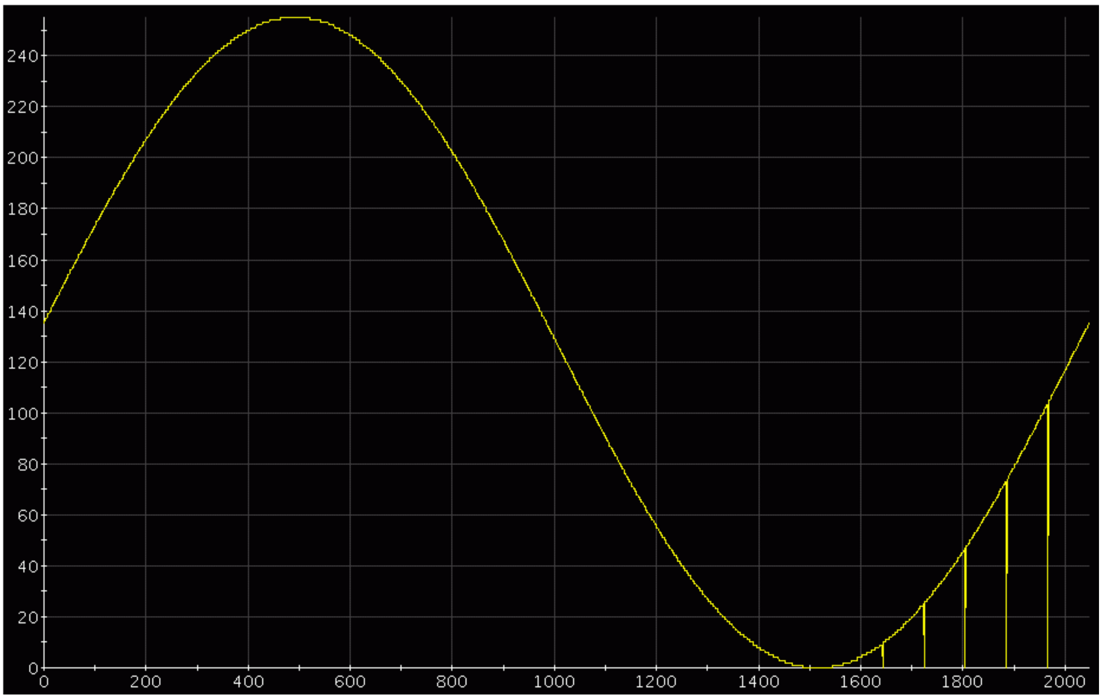


**Figure 18: Reshuffled Waveform (5)**

You can easily find five ugly samples occurred regularly in there. The result of DSP_SHUFFLE() does not tell you time information directly. Anyway you should understand there are five broken samples in the waveform. So take a closer look at the original waveform in Figure 17 again. Then you may notice that the final five points of data are not captured correctly, if your eyes are very good. They stay at zero. Probably the ADC was not supplied enough sampling clocks to complete the necessary number of points for some reason.

The starting area of the captured waveform would be another check point as well. It could relate to a waiting time problem and/or the trigger problem.

As you see in the exercises, if you experience any strange spectrum, you should carefully examine the original waveform. You should recognize typical appearances. DSP_SHUFFLE() is a handy tool to check the waveform.

**[Problem 5]**
The final abnormal spectrum may be slightly difficult, but engineers who have long carrier may have experienced this kind of situation. The signal itself looks perfect in this case; however the noise floor gradually rises in the DC direction.

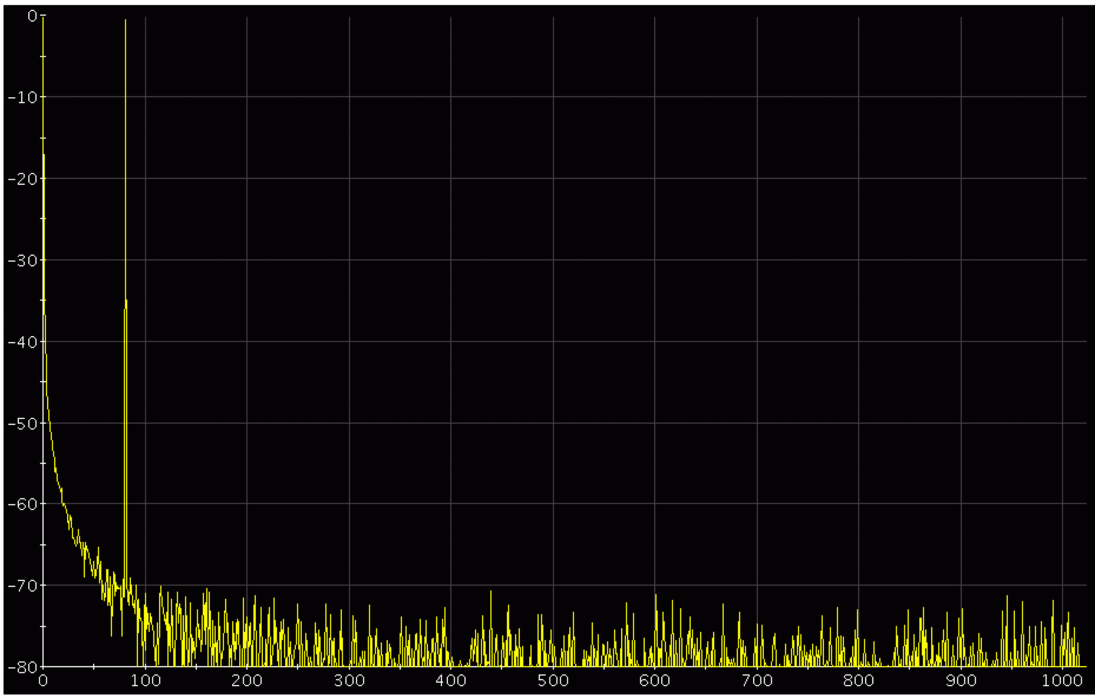


**Figure 19: Abnormal Spectrum (6)**

**[Answer 5]**
Let's look at the original waveform in Figure 20. It looks normal at a glance. But don't you think something is strange? The trend of the waveform is slightly inclined upward over time. The signal contains DC drift. If the input signal circuit is AC-coupled or any DC blocking capacitors are integrated in the path, this phenomenon may occur. The circuit is not balanced yet, or you need more wait time to capture samples. As other possible reasons, the device or peripheral circuit may not be stabilized in terms of temperature, or the input signal source might have 1/f type noise inherently. Anyway you need to detect the cause of the problem, and resolve it for taking a good spectrum. That is one of the thrills of online debug.

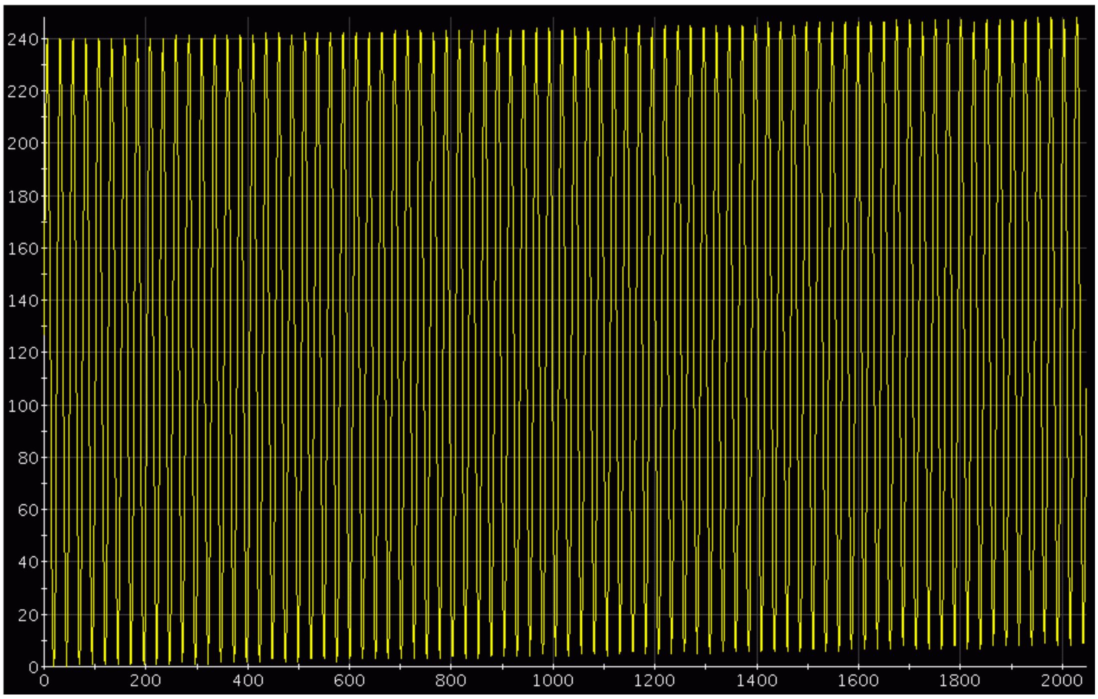


**Figure 20: Waveform (6)**