Create a stimulus-response system with an AWG and digitizer

Arthur PiniGreg Tate,Oliver Rovini, - March 02, 2016

Many devices and systems don't require a stimulus signal for you to test them. But, when it comes to amplifiers, filters, transmitters, receivers, digital interfaces, or any system that requires an input signal, you'll have to generate it and measure the output. Modular digitizers and arbitrary waveform generators (AWGs) let you test one or more devices that need input signals. Combining the two products in one system provides very cost effective and efficient way to meet an extensive range of automated test requirements.

Characterize a filter

AWG's let you generate almost any waveform, analog or digital. Consider a simple test for determining the frequency characteristics of an amplifier or filter. The test requires a signal source with a bandwidth greater than that of the device under test. Furthermore, the source needs to deliver a constant output level over the entire testing bandwidth. A swept sinewave or impulse function waveforms offer broadband output with flat spectral characteristics. Either can be generated by an AWG. The swept sine offers a greater dynamic range for the measurement. **Figure 1** shows the result of a swept sine frequency response measurement of a 36 MHz low-pass filter.

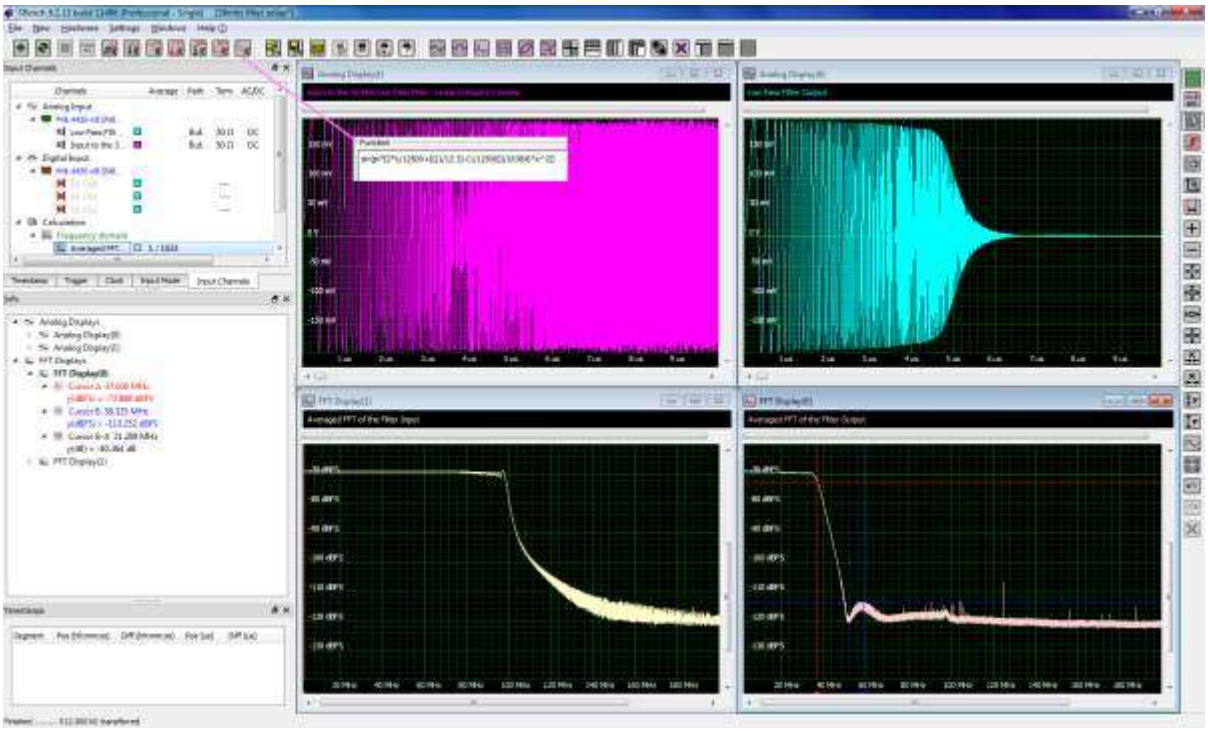

Figure 1. The frequency response of a 36 MHz filter shows the filter's input and output in

the time domain (top left & right, respectively, and in the frequency domain (bottom left & right, respectively).

The upper left grid shows the swept sine that we applied to the filter's. For this test, we created the test signal using the equation:

$$V_{OUT} = sin\left[\pi \times \left(\frac{2X}{X_S} + \frac{\frac{1}{X_E} - \frac{1}{X_S}}{X_{MAX}}\right) \times x^2\right]$$

Where X=Sample value
X_S=Start period in samples
X_E=End period in samples
X_{MAX}=Duration of the waveform in samples

In this example, we used:

$$V_{OUT} = sin\left[\pi \times \left(\frac{2x}{12500} + \frac{\frac{1}{12.5} - \frac{1}{12500}}{16384}\right) \times x^2\right]$$

The bottom-left grid shows the FFT (Fast Fourier Transform) of the filter. The swept sine was flat out to 100 MHz. The upper-right grid shows the filter's time response. The filter's output decreases rapidly above its 36 MHz cutoff frequency. The matching frequency response is shown in the grid on the lower right. You can clearly see the filter's bandwidth, in-band flatness, and stop-band attenuation.

Matched diodes
Some circuits require diodes with matched I-V curves. While you can make these measurements with a curve tracer, you can also make automated measurements using an AWG and digitizer, shown in **Figure 2**. Note that the maximum current supplied to the diode must be less than the maximum AWG output current otherwise an amplifier may be required. This works well for the 5082-2811 Schottky Barrier diode used in this example.

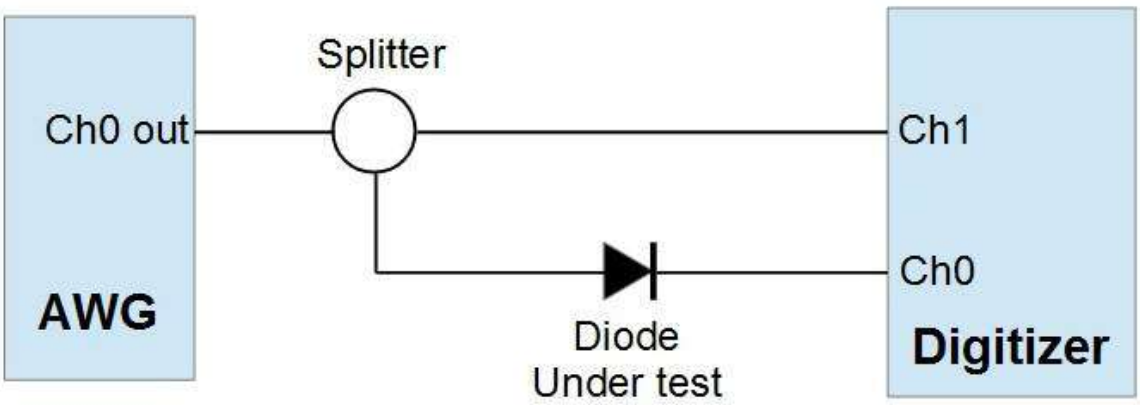

Figure 2: a simple setup for quickly characterizing signal diodes using a ramp function from the AWG. The 50 Ω input termination of the digitizer serves as the load.

Diode measurements, modulation, part simulation

Figure 3 shows the result of the measurement. The AWG is setup to output a ±2 V ramp waveform. This waveform is applied, through a splitter, to the digitizer's channel Ch1. The other leg of the splitter connects to the diode under test and then on to the digitizer Ch0. Both channels are set to 50 Ω termination. The voltage on channel Ch0 is proportional to the current through the diode. Rescaling is applied to that channel so that the readings for Ch0 will display in milliamps. The voltage across the diode is calculated by Ch1 - Ch0.

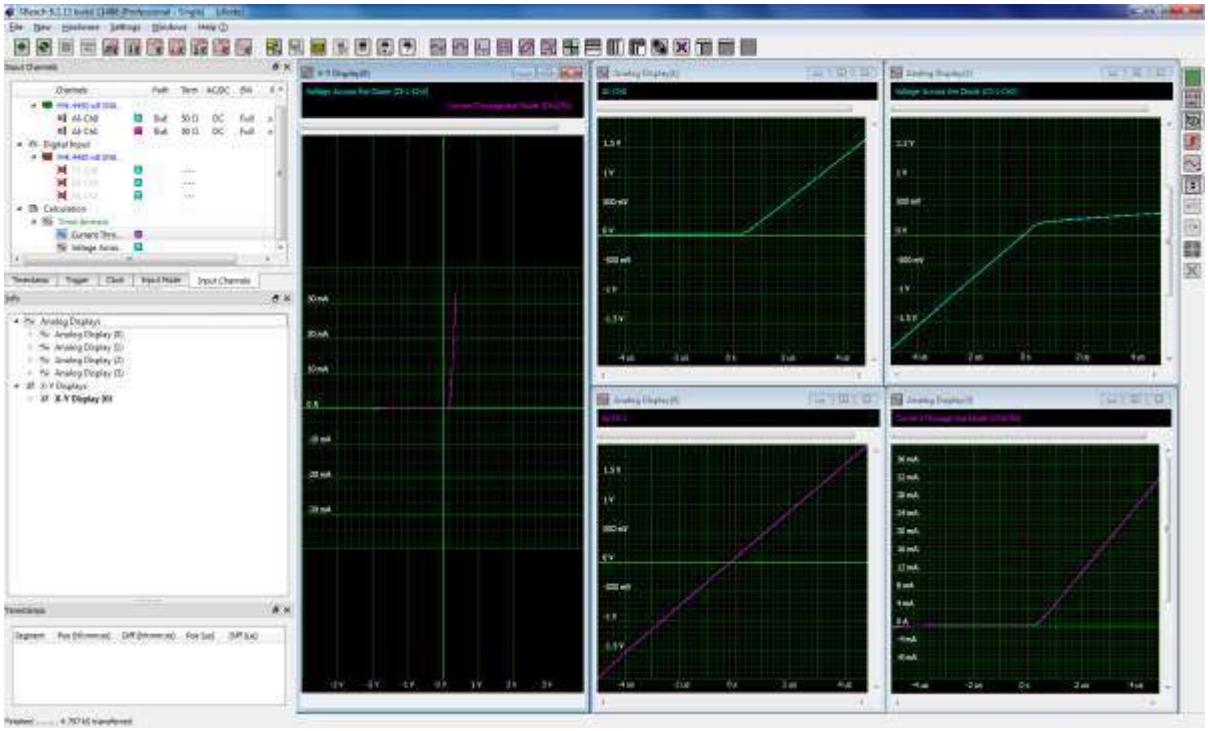

Figure 3. With some basic math, you can derive the current through the diode and display it.

The center bottom grid is the ramp waveform from the AWG shown in Ch1 of the digitizer. The center-top grid shows the Ch0 waveform, which is proportional to the diode's forward current. Subtracting Ch0 from Ch1 yields the voltage across the diode (upper-right grid). The lower-right grid is the shows the diode current directly in milliamps. The X-Y plot in the leftmost grid is the diode's Current-Voltage (I-V) plot of the diode. To find matching diodes, look for closely matching I-V plots.

Although this is a very simple example of device characterization, many AWG's can perform more complex small-signal device testing. Testing devices at higher voltage and current levels may require amplifiers.

Modular modulation

Because an AWG can generate almost any waveform, you can perform communications tests by generating modulated signals. Amplitude, phase, and frequency modulation can be achieved as was demonstrated in the first example on frequency response testing where a linear frequency swept sinewave was used. In **Figure 4**, we used quadrature signal techniques to generate a "chirp"

waveform at an intermediate frequency.

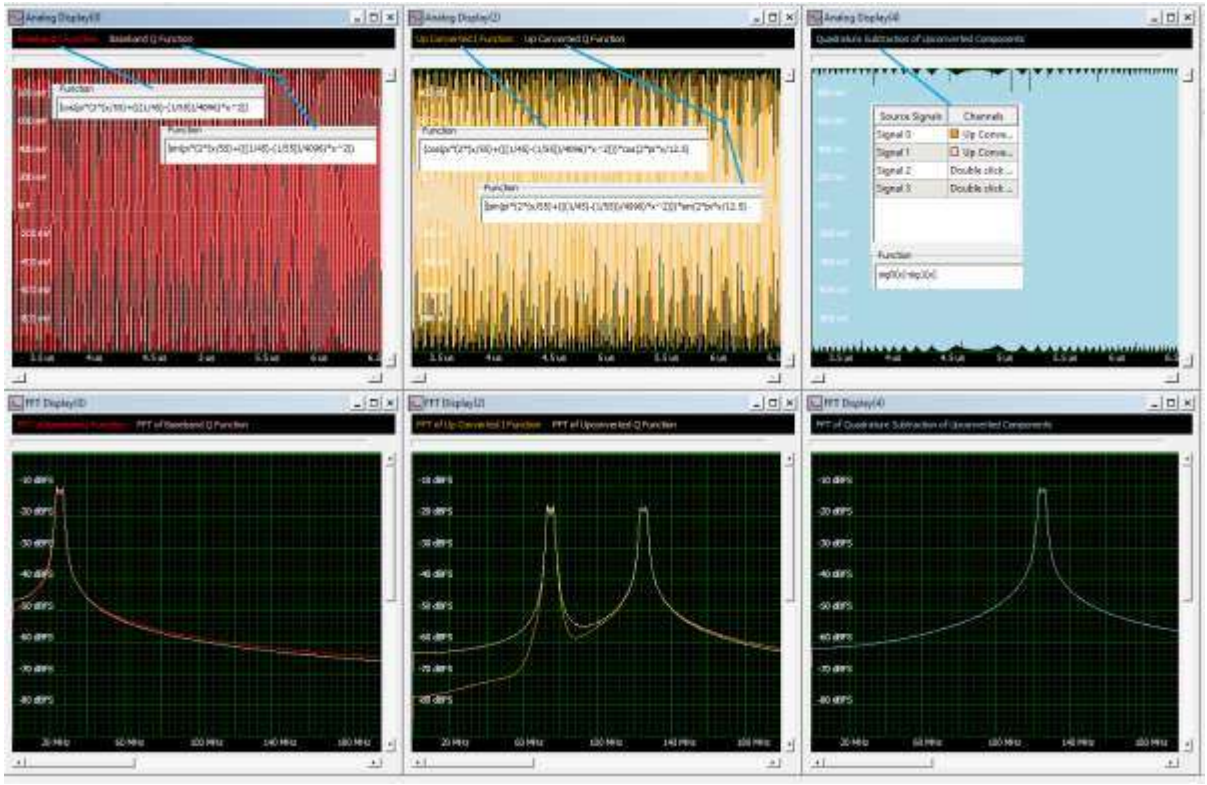

Figure 4: Using quadrature signal processing to generate a "chirp" waveform at an intermediate frequency.

Figure 4 shows the steps in creating the chirp. It starts in the upper left grid where linearly swept sine and cosine waveforms are shown in the time domain along with the equations used to create them. These signals are in quadrature representing the I (in-phase) and Q (quadrature) components of a quadrature modulation, but the phase difference doesn't show in the FFT's of these waveforms, which appear in the lower left grid. The FFT shows the sweep range from 22.7 MHz to 27.7 MHz. This is a baseband signal.

We then up-convert the baseband quadrature components by multiplying the I and Q baseband signals by the cosine and sine of the intermediate frequency, respectively. The intermediate frequency is 100 MHz. The upper central grid shows the time domain view of the up converted signals. The lower central grid shows the FFT's. Note that the multiplication by a sinusoid has converted the baseband signals to double sideband suppressed carrier signals centered at 100 MHz with upper and lower sidebands extending from 22.7 MHz to 27.7 MHz on either side of the center frequency. The amplitude FFT doesn't, however, show the phase of the two quadrature components.

In the final step, we subtract the quadrature IF components, I and Q. The right-hand grids show the results. The lower-right grid shows that, as a result of the subtraction, the lower sideband has been cancelled leaving only the upper sideband. This signal sweeps from 122.7 MHz to 127.7 MHz.

Simulate missing parts

The lack of a critical component can often delay testing and product development. If, however, you have access to the missing component's response waveform, you can generate it with an AWG. You can capture real-world waveforms with a digitizer or oscilloscope, then import the waveform into the AWG for playback. One such example is a CAN (controller area network) bus, a serial data stream generated by a steering angle sensor (**Figure 5**). This waveform was acquired using an oscilloscope and transferred to the AWG in ASCII file format.

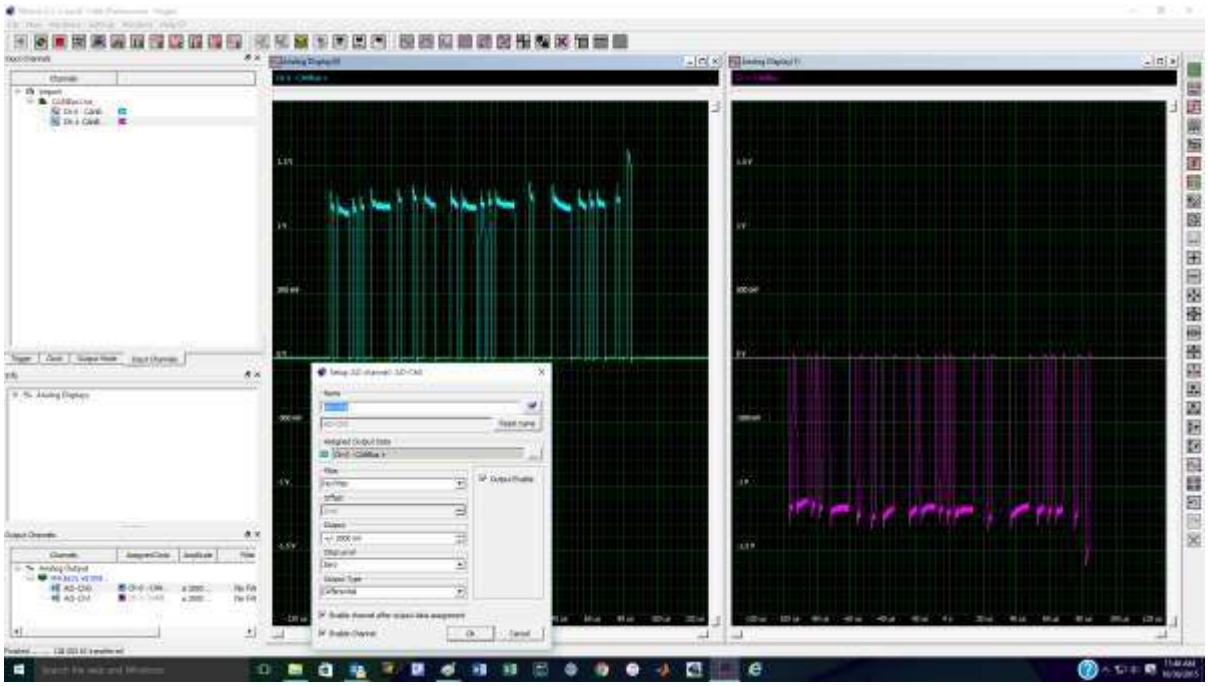

Figure 5. An AWG can simulate a serial data stream such as those from a CAN bus.

A CAN bus signal is dual-channel waveform that represents the bus' differential (+ and –) components. To generate this signal, you need a two-channel AWG that's set to output the CAN bus waveform as a differential signal.

Once the AWG receives a trigger signal, it can generate the waveform so you can test the system. Additionally, these waveforms can be modified for margin testing of both amplitude and timing.

Conclusion

These few examples show the great diversity that is possible when using an AWG as a signal source for testing. AWG's can generate standard function generator waveforms such as sine, square, triangle and ramp signals. They can generate modulated waveforms and serial data patterns. They can even be used to replay real world signals that have been acquired by digitizers and oscilloscopes. AWGs can be paired with an accompanying digitizer and programmed using a manufacturer's supplied software suite such as SBench 6, commonly available system integration software like MATLAB, or LabVIEW, or custom programmed in the language of your choice through an API (application programming interface).

See EDN collection: Oscilloscope articles by Arthur Pini