EDN NETWORK

[Modular AWGs: How they work and how to use them]

[Arthur Pini](#)[Greg Tate](#),[Oliver Rovini](#), - June 16, 2016

The AWG (arbitrary waveform generator), with its near universal selection of waveforms, has become a popular signal source for test systems. Modular AWGs let you add standard or custom waveforms to PCs as part of an automated test station. With an AWG, you can create waveforms by using equations, by capturing waveforms from digitizers or digital oscilloscopes, or you can create your own waveform with manufacturer supplied or third-party software. Waveform sequencing lets you switch among predefined waveforms.

Today's modular AWGs offer extended bandwidth, higher sampling rates, and longer waveform memory than previous models. In addition, they offer advanced operating modes and the ability to stream large amounts of waveform data from a PC's main memory. Before selecting a modular AWG, learn how they work and what they can do.

How AWGs work

AWGs are digital signal sources. They operate like digitizers or digital oscilloscopes in reverse. The AWG has a numeric description of the waveform stored in waveform memory. Selected samples from the memory are sent to a DAC (digital to analog converter) and then, after filtering and signal conditioning, samples are output as an analog waveform. **Figure 1** contains a conceptual block diagram of an AWG.

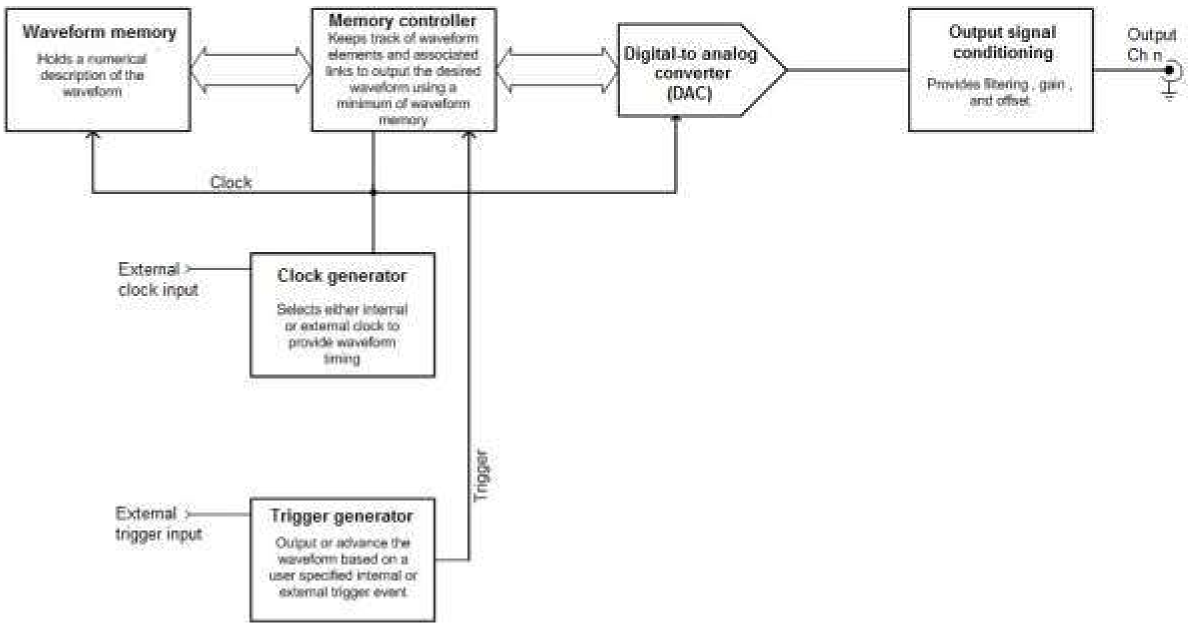

Figure 1. An AWG consists of several functions blocks that take a digital representation of a waveform and produce a filtered analog signal.

Data representing the waveform first gets loaded into the waveform memory. Normally, the waveform memory has a larger data width and is clocked with a divided sampling clock compared to the DAC. An FPGA in between demultiplexes the samples and generates a waveform data stream at the DAC's sampling clock. When commanded by the memory controller, the contents of the waveform memory are sent to the DAC for conversion into an equivalent analog voltage. Some DACs allow additional interpolation, which results in a higher update rate at the output than supplied by the waveform memory.

The raw DAC output is rich in harmonics and requires filtering. An AWG's output stage filters and conditions the signal by adjusting gains and offsets to meet the user's waveform specification.

The memory controller keeps track of the elements of each waveform component in the waveform memory, and any associated links, and outputs them in the correct order. To save memory space, the memory controller can loop on repetitive elements so that these elements need be listed only once in the waveform memory. A clock generator uses either an internal or an external clock to provide a common timebase. The trigger generator, which causes the waveform to be output or advanced based on a user specified event, provides synchronization. In addition to internal or external trigger events, the AWG can be linked to another modular AWG or digitizer.

The actual implementation of the functions above vary with specific models, but all AWGs have similar elements.

AWG Specifications

An AWG's specifications are quite different from standard signal generators because of the AWG's output waveform selection and its digital nature.

The key parameters, just as with a digitizer, are bandwidth and sampling rate. The *bandwidth* determines the highest sinewave frequency that the AWG can output with a loss less than 3 dB. Because many of the waveforms that an AWG can create are rich in harmonics, the bandwidth limit will determine the highest frequency waveform that can be generated. For example, a square wave generally must pass the fifth harmonic to be recognizable. For a given bandwidth, the highest frequency square wave is typically one fifth of the AWG's bandwidth.

An AWG's *sampling rate* is related to its bandwidth. According to sampling theory, the sampling rate has to be at least twice the bandwidth. The sampling rate also determines the AWG's horizontal resolution. This defines the smallest time increment that can be set within a waveform.

The size of the waveform memory determines the longest waveform that the AWG can produce without repeating (looping) any waveform components. The limit of signal duration, without looping, is memory length times the sample period. An AWG that has 2 Gsamples of waveform memory and a maximum sample rate of 1.25 Gsamples/s can produce a waveform that's 1.6 s long. Looping lets you repeat redundant waveform components, which can greatly increase the maximum waveform length.

Amplitude resolution specifies the minimum output signal level the AWG can generate, which is also the minimum amplitude step between adjacent samples. The amplitude resolution of the AWG is determined by the number of bits of resolution of the DAC and memory. In general, there is a tradeoff between DAC resolution and sampling rate. That is, the greater the number of bits in the DAC, the lower the maximum sampling rate. An AWG that has 14-bit resolution has a theoretical dynamic range of 16384:1 An AWG with 16-bit resolution has a theoretical dynamic range of 65536:1. Noise and other factors reduce dynamic range, just as they do with digitizers.

The *maximum output amplitude* that the AWG can generate is determined by the output amplifier stage. In general, there is also a trade-off between an AWG's sampling rate and output amplitude, with faster AWG's having a lower maximum output amplitude. The minimum full-scale output range depends on the internal attenuators in the output stage. On any given full scale range, the theoretical minimum value is the full output divided by the amplitude resolution (an AWG with a 10 V_{P-P} range and 16-bit resolution has a minimum output step of 10/65,536 = 152.5 µV). Internal noise and non-linearity limit the practical minimum signal output.

An AWG's *number of channels* typically ranges from one to four, but AWGs can often be synchronized to provide additional channels.

Output filtering improves the signal to noise ratio of the AWG output. Generally the types and cutoff frequencies of the filters can be specified.

All AWGs can create *modulated waveforms* by creating them analytically, in software, using manufacturer's operating software like Spectrum's SBench 6 or other third-party math software, and downloading them into the AWG's waveform memory.

Another useful feature is having a trigger input to initiate the output or to advance the waveform through multiple segments. We'll cover that in more detail below.

AWGs can also produce an output trigger or marker output synchronous with the waveform output. These signals can then be used to trigger a digitizer, oscilloscope, or other instrument at appropriate times during the waveform.

Operating Modes

AWGs may incorporate multiple operating modes that affect how they store and replay waveforms. The ability to repeat (loop) selected segments of the waveform and advance between segments based on triggers or gating signals adds flexibility and reduces the amount of memory required for complex waveforms. Here is a summary of common operating modes:

- *Single shot*: The programmed waveform is played once for each external or software trigger. After the first trigger, subsequent triggers are ignored.
- *Repeated (continuous) output*: The programmed waveform is played continuously for a pre-programmed number of times or until a stop command is executed. The trigger source can be either an external hardware trigger input or a software trigger. After the first trigger, additional trigger events will be ignored.
- *Single Restart replay*: This mode outputs the waveform data of the on-board memory once after each trigger event. The trigger source can be either hardware or software.
- *FIFO*: Some AWG's offer a FIFO (first in first out) operating mode designed for continuous data transfer between the host computer's memory or hard disk and the AWG. The AWG's on-board memory serves for buffering data, making continuous streaming extremely reliable. In this mode, the available waveform memory is limited by the host computer's memory.

  AWGs that support FIFO streaming can further extend waveforms by utilizing the host computer's memory. In FIFO mode, the AWG uses its on-board memory as a high-speed buffer between system memory and the DAC. This frees the AWG from the limits of its internal memory. Combining FIFO streaming with looping and linking functions lets you produce even longer waveforms.
- *Multiple replay*: The multiple-replay mode (**Figure 2**) provides fast output of waveforms on multiple trigger events without restarting the hardware. The on-board memory is divided into several equal size segments. Each segment can contain different waveform data, each of which is output with the occurrence of each trigger event. This mode allows very fast repetition rates.

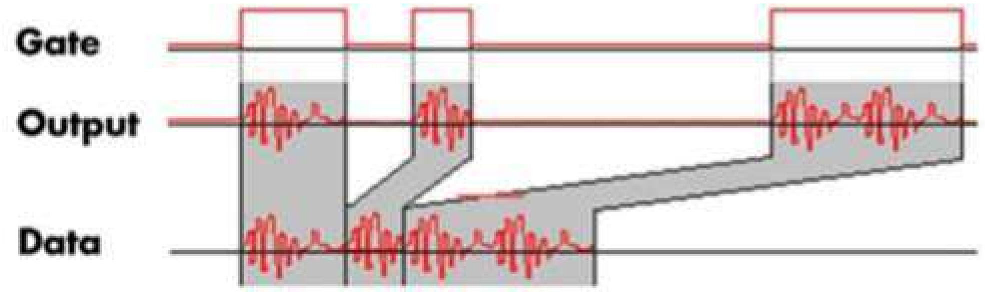

Figure 2: Using the multiple replay mode to output three waveform segments upon trigger input.

- *Gated Replay*: The gated sampling mode outputs waveform data controlled by an external gate signal (**Figure 3**). Data is only replayed if the gate signal is at a preprogrammed level.

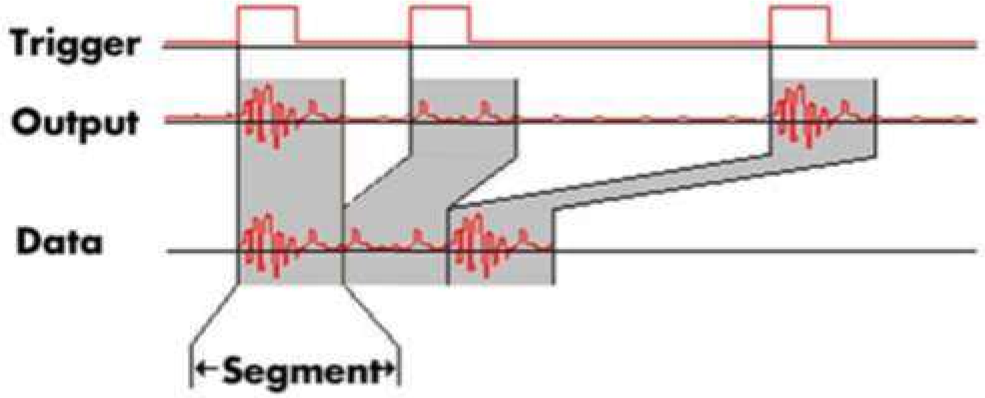

Figure 3. Use the multiple-replay mode to output three waveform segments upon a trigger input.

- *Sequence mode*: The sequence mode (**Figure 4**) splits the internal card memory into a number of data segments of different lengths. These data segments are chained in a user set order using an additional sequence memory. The sequence memory determines the order that segments are output as well as the number of loops for each segment. Trigger conditions can be defined to advance from segment to segment. Using sequence mode, an AWG can switch between replay waveforms by a simple software command or it can redefine waveform data for segments simultaneously while other segments are being replayed.

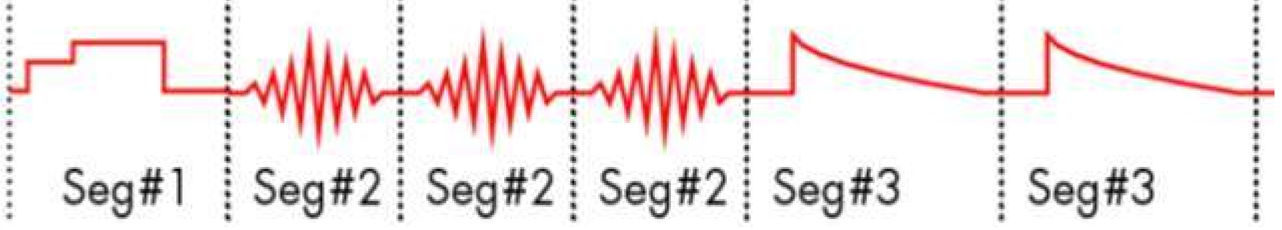

Figure 4. Sequence mode outputs waveform segments in the order specified in the sequence control memory.

Selecting an AWG

Selecting an AWG
The selection of an AWG requires matching the AWG specifications your test specification.

The basic starting point is generally the AWG's bandwidth, the highest frequency that can be produced. This frequency must be greater than or equal to the maximum frequency required for testing. Keep in mind that harmonic rich waveforms require bandwidths three to five times the frequency required. For example: a 100 MHz bandwidth AWG can output a 100 MHz sine but will only support a 20 MHz square wave.

As previously described, the maximum sampling rate of the AWG must be at least twice the required bandwidth. This is the Nyquist limit and, in practice, it is generally better to over sample by a factor of three or four. The sampling rate determines the smallest increment of time that can be programmed. Note that AWG’s often restrict the minimum number of samples required to create a waveform. Generally they require waveforms to contain an even number of samples or be multiples of a fixed number of samples (e.g. 4, 8, 16, etc.). These requirements arise because many AWG's use multiplexed memory to obtain high sample rates. If they multiplex 4:1, then waveforms have to be multiples of 4 samples.

The waveform memory length determines the longest duration, non-repeating signal that the AWG can support. Operating modes that support ‘looping’ or repeating redundant elements in the waveform decrease the amount of waveform memory required.

The maximum output level of the AWG must match the test requirement. If not, it may require an external amplifier with a bandwidth that exceeds the AWG’s so that the bandwidth of the combination equals the specified bandwidth.

The ratio of the highest test signal amplitude to the minimum amplitude at the same time determines the dynamic range requirements of the test. This is determined by the amplitude resolution of the AWG expressed in bits. Note that noise and non-linearity in the AWG limit the dynamic range to less than the theoretical values. The actual performance is often characterized as the ENOB (effective number of bits).

The AWG requires software for waveform generation and operational control. Almost all AWGs are shipped with drivers for common operating systems. In the case of the Spectrum AWGs, drivers for Windows and Linux are supplied. Drivers allow you to write your own software using common programming languages and platforms including C/C++, IVI, .NET, Delphi, Visual Basic, and Python. Drivers also support third party software such as LabVIEW (Windows), MATLAB (Windows and Linux), and LabWindows/CVI. AWG's often include full-featured software packages such as SBench 6.

Creating waveforms

If you use an AWG's waveform-generation software, you can select from several predefined waveforms such as sine, rectangular, triangle, sawtooth, SINC, and DC waveforms. The frequency, phase, and amplitude of each waveform is user adjustable, as is the duty cycle for the rectangular, triangle and saw tooth waveforms. **Figure 5** shows a typical example of such software control.

Figure 5: AWG software often includes sine, rectangular, triangle, sawtooth (ramp), SINC, and DC waveforms for each AWG channel. The amplitude, frequency, and phase are user adjustable.

The most accurate way to create waveforms is to base them on mathematical equations. They are precise and repeatable and offer a great range of test signals. Waveform creation software includes a function editor that supports the generation of waveforms using text-based equations as shown in **Figure 6**. The function generator editor accepts the equation in text format and allows for the selection of the sampling rate, amplitude, and duration of the waveform.

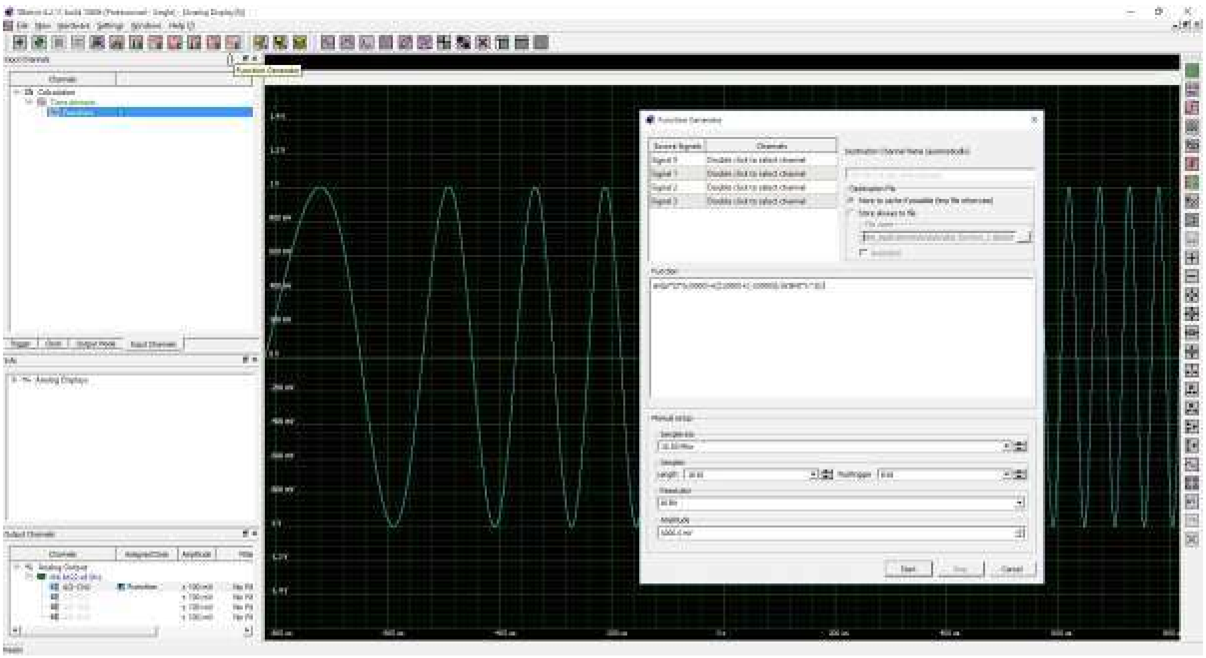

Figure 6. A function generator editor lets you waveforms based on equations, such as this swept sine wave.

Importing Waveforms

Waveforms can also be created or acquired from other sources including instruments suck as digitizers and digital oscilloscopes and software tools including spreadsheets, math programs, and system-integration software. Waveforms from these sources can be imported into SBench 6 and sent to the AWG using any of the formats listed in **Table 1**.

Table 1. Typical waveform sources and the data formats supported by SBench 6.

Waveform Source	File Format for transfer
Spectrum Digitizer	.sb6dat, ASCII, Binary
Spectrum AWG	.sb6dat, ASCII, Binary
Spectrum digitizerNETBOX	.sb6dat, ASCII, Binary
Digitizer (Non-Spectrum)	ASCII, Binary, Wave
Digital Oscilloscope	ASCII, Binary, Wave
Spreadsheet (Excel)	ASCII
Math Program (MATLAB)	ASCII, Binary
System Integration Software (LabVIEW)	ASCII, Binary
Audio Recording	.WAV

Signal Processing

Signal processing within the AWG support software lets you combine multiple waveforms using waveform arithmetic that supports sum, difference, product, and ratio. Moving average and filtering can be used to reduce noise and improve signal-to-noise ratio on imported signals.

Conclusion

Modular arbitrary function generators are excellent signal sources for test systems providing small size, configuration flexibility, and easy integration. With bandwidths of up to 400 MHz, sampling rates to 1.25 GS/s and with 16 bits of amplitude resolution they offer a broad range of test solutions.

Also see

- Create high-performance stimulus test systems using MATLAB and arbitrary waveform generators
- Maximize a waveform generator's memory
- The year of the waveform generator