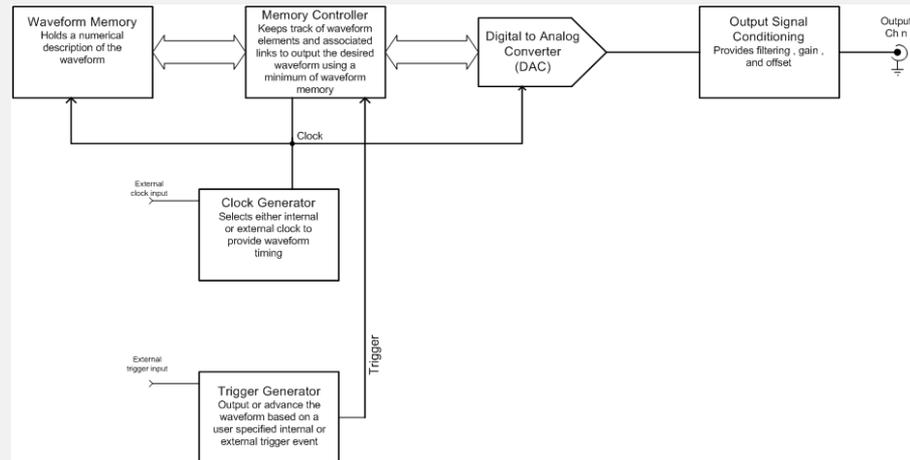


► **Application Note**

## Using Arbitrary Waveform Generator Operating Modes Effectively

### Introduction

One of the great powers of Arbitrary Waveform Generators (AWG's) is that they can generate an almost infinite number of waveform shapes. The AWG's operating mode controls the timing of how these waveforms are output. In this application note we will investigate the effective use of the operating modes for the Spectrum M4i.66xx series of AWG's, concentrating on the sequence mode which offers the ability to control the selection of the output waveshapes in near real-time.



**Figure 1: AWG Conceptual block diagram showing the data path from memory to output**

AWG's operate like a digitizer in reverse. They store a waveform in numerical form in the waveform memory and then a waveform controller routes the digital data to a digital to analog converter (DAC) which converts it into an analog voltage. This is illustrated in Figure 1.

### Waveform Memory Operating Modes

The waveform memory in the Spectrum M4i.66xx series AWG's is used in two distinct modes. The first is Standard mode in which the waveform data is stored entirely within the waveform memory, this limits the waveform duration to the length of the waveform memory. Note that for this product series the waveform memory is a healthy 2 GSamples providing waveform durations up to 1.6 seconds at the fastest (1.25 GS/s) clock rate with no part of the waveform being repeated.

The second mode is First In-First Out (FIFO) which streams waveform data from the host computer using a PCIe x8 serial interface at rates of up to 2.8 GB/s. The internal waveform memory is used as a streaming buffer. This mode allows waveform durations to be extended with the limitation now being the available memory in the host computer. PC systems using RAID based data storage, such as Spectrum's Streaming Systems, can even be used to enable uninterrupted waveform generation for several hours.

### AWG Operating Modes

The operating modes determine when, and what part of the waveform, is output. The operating mode works in conjunction with the AWG trigger which can be internally or externally generated.

The operating modes are summarized below:

- **Single** - data in the waveform memory will be replayed once after the first trigger event

## ► Application Note

and subsequent triggers are ignored.

- **Single restart** – The same data of the waveform memory will be replayed once after each trigger event.
- **Repetitive (continuous)** - data of the waveform memory is played continuously for a programmed number of times or until a stop command is issued.
- **Gated** – allows the waveform output to be controlled by the state of an external gate signal.
- **Multiple** – Memory is divided into segments of the same length. One segment is output per trigger event.
- **Sequence** – Memory is divided into segments. Waveforms of differing lengths are loaded into the segments. A sequence file determines the order of output, loop count, and segment to be output.

Multiple and sequence mode both segment the waveform memory with each segment containing one waveform or part of a waveform. Each trigger then advances the segment number and outputs the selected waveform. The sequence mode adds the ability to loop the segment contents for a user selectable number of times as well as selecting the next segment to be output.

## Sequence Mode

The whole sequence process is explained in Figure 2

The sequence mode, as mentioned previously, segments the AWG waveform into several data segments.

Each of the waveforms loaded in each segment can be of a different size. The data segments are linked in a user chosen order using an additional sequence memory, as shown at the bottom of Figure 2. The sequence memory links the memory step with a specific segment, specifies the number of loops for each segment, and defines the next segment. Importantly, the switching between waveform segments occurs with no dead time between segments.

In our example 4 steps within the sequence memory have been defined. Three of them (Step #1, Step #3, and Step #4) perform an endless loop that will be repeated continuously. The output of the card will contain data segment #2 repeated 10 times, data segment #3 repeated 100 times, and data segment #7 repeated only once. At this point the AWG loops back to Step 1 and begins the cycle over again.

In sequence mode it is possible to change the replayed waveforms by a simple software command

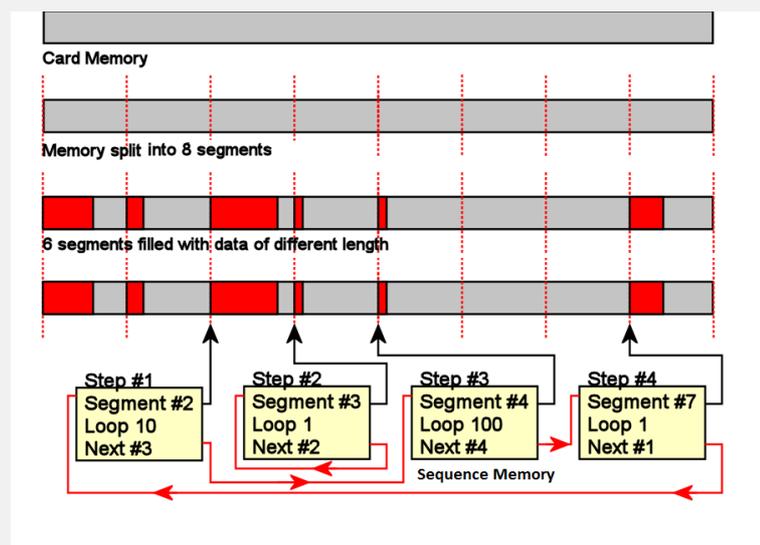


Figure 2: Waveform memory splitting and sequence memory linking

### ▶ Application Note

or to redefine the waveform data for segments at the same time other segments are being replayed. This allows for near real-time programming of the waveform output.

The strict separation of the waveform and sequence memories makes it possible to change the sequence memory during runtime. If we look again at the example, the sequence has an unused step, step #2. In our example 3 steps have been defined prior to the AWG start. At first these are not changed. Step#2 is set up to repeat itself. That is it links back to itself as the next step, but due to the defined sequence order it is normally not used. Due to the nature of the sequence memory (read-before-write) it is possible to write to any step register

in the sequence memory during runtime without corrupting it. By addressing a certain step and changing, for example, its next parameter, it is possible by software to switch between two sequences.

Assuming in our example above, that we change the next parameter of Step #4 from Next=1 to Next=2, the infinitely executed 3-step sequence (that has been repeated since the AWG start) will be left the next time that the replay finishes the last sample of the pattern associated with Step #4 (which in this case is Segment #7) It will then jump to step #2 and seamlessly continue replaying with the first sample of the associated segment #3. As step #2 links back to itself it will generate the data segment #3 in an endless loop until being stopped by the user.

Any of the three step parameters „Next“, „Segment“ and „Loop“ of any step in the sequence memory can be changed during runtime, without corruption of the sequence memory. However, once a step is entered it will be executed and the current settings, such as outputting the associated pattern and repeating it the programmed number of times, will be completed.

## Advantages of Sequence Mode

The Sequence mode for the Spectrum M4i.66xx series AWG's offers considerable advantages. The first is that it makes more effective use of internal memory. Recurrent waveform elements can be stored once and repeated as necessary reducing the required storage space.

There is great flexibility in the ordering of the output waveform selection. All the waveforms that are necessary for a test procedure can be loaded immediately and then selected as needed. This greatly increases the testing speed.

Additionally, there is near real-time control of waveform segments making it easy to provide adaptive response to test needs. Measured test results can change the sequence order and this can occur without stopping the test. This is the most powerful benefit because it enables adaptive testing where test conditions can be changed based on measured performance.

Consider the following generalized scenario:

- Preload all test waveforms.
- Connect the device under test (DUT) to an AWG and digitizer.
- Output test waveform 1
- Acquire and analyze the effect of test waveform 1 on the DUT with the digitizer
- Based on the measurement result select the next test
- Replay test waveform 2

### ► Application Note

- Acquire and analyze the effect of test waveform 2 on the DUT with the digitizer
- Based on the measurement result select the next test
- Continue until the test is complete

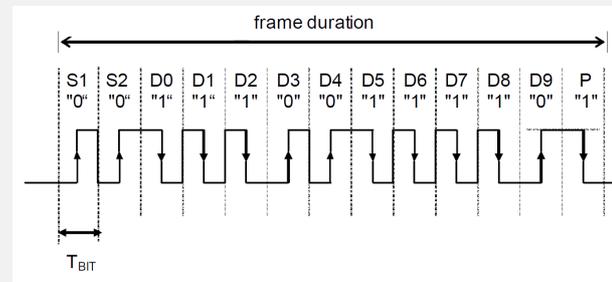
This ability to store all test waveforms and call them up as needed by editing the sequence memory on the fly is a unique capability. Combined with the AWG's ability to generate almost any wave shape, it constitutes an ideal test generator.

## An Example of Using Sequence Mode

Consider testing the receiver for a Manchester encoded serial data stream as shown in Figure 3.

Manchester coding is used in many common serial data links including RFID, PSI 5, and Ethernet.

Manchester code always places a transition at the middle of each bit period. It may (depending on the information to be transmitted) have a transition at the start of the period as well. The direction of the mid-bit transition indicates the data. Transitions at the period boundaries do not carry information. They exist only to place the signal in the correct state to allow the mid-bit transition. The guaranteed transitions permit the signal to be self-clocking.



**Figure 3: example of a Manchester encoded serial data stream that should be generated by the AWG**

This compound waveform consists of three waveform components shown in Figure 4:

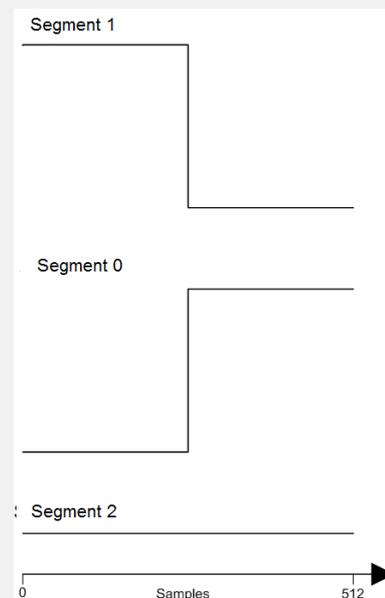
#### Waveform Segments

- "1" - a high to low transition
- "0" – a low to high transition
- Baseline – 0 volt DC level.

By defining three waveform segments using these components any combination of data patterns can be synthesized. This means that by rearranging the order of these three segments the content of the packet can be changed. 1

In the example that follows the segments will be set to a length of 512 samples with a clock rate of 50 MS/s so that the duration of each component (TBit in Figure 3) will be 10.24  $\mu$ s. Packets are separated by a baseline signal lasting more than two bit clock periods.

The Spectrum M4i.6631-x8 AWG was controlled using a MATLAB script which constructed four different data patterns for test purposes. Spectrum provides drivers for both Windows and Linux operating systems. The drivers support most common programming languages and third party system integration software like LabVIEW, MATLAB or LabWindows/CVI. All drivers come with detailed documentation and working examples.



**Figure 4: 3 data segments are used to generate the Manchester code**

The MALAB code controlling the sequence memory contents are shown below:

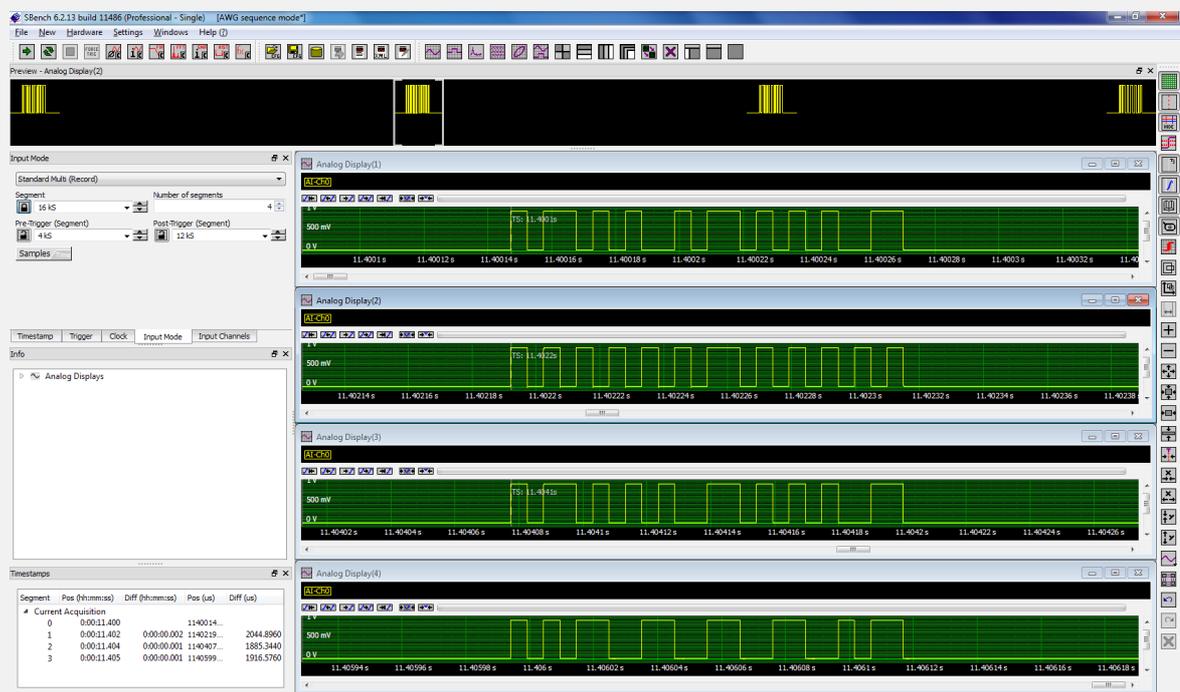
► **Application Note**

```
%----- set sequence steps -----
%
spcMSetupSequenceStep (cardInfo, step, nextStep, segment, loops, condition)
spcMSetupSequenceStep (cardInfo, 0, 1, 2, 2, 0);
spcMSetupSequenceStep (cardInfo, 1, 2, 1, 1, 0);
spcMSetupSequenceStep (cardInfo, 2, 3, 1, 1, 0);
spcMSetupSequenceStep (cardInfo, 3, 4, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 4, 5, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 5, 6, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 6, 7, 1, 1, 0);
spcMSetupSequenceStep (cardInfo, 7, 8, 1, 1, 0);
spcMSetupSequenceStep (cardInfo, 8, 9, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 9, 10, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 10, 11, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 11, 12, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 12, 13, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 13, 14, 0, 1, 0);
spcMSetupSequenceStep (cardInfo, 14, 0, 2, 2, 2);
```

Note: Condition Definitions: (0 => End loop always, 1 => End loop on trigger, 2 => End sequence)

The waveform is built up in fourteen steps using only the three components that have been described above.

The MATLAB script contains four such tables, each loading a different combination of segments. When the AWG is run under this MATLAB script it produces four different data patterns. The results of the data output is shown in Figure 5.



**Figure 5: Acquisition of different segments of the serial data packets using a M4i.4450-x8 Digitizer**

A Spectrum M4i.4450 -x8 digitizer was used to acquire four segments of the serial data packets in the multiple acquisition mode. The top display is the data pattern that has been discussed. The other three are different patterns showing how the AWG can switch the data content just by changing the sequence memory content.



## ▶ Application Note

### Conclusion

The AWG operating modes control the choice and timing of waveforms. Sequence mode is the most versatile mode with the ability to segment the memory and select the waveform segment to be output, set the number of times the segment is to be repeated, and then determine the next segment. The programming of the segments is controlled by the sequence memory which can be changed while the AWG is running. This means that the waveforms be output can be changed in near real-time providing the ability to do adaptive programming of tests.

### Authors

*Arthur Pini – Independent Consultant*

*Oliver Rovini- Technical Director, Spectrum GmbH*

*Greg Tate - Asian Business Manager, Spectrum GmbH*