Agilent M9330A/M9331A Series PXI-H Arbitrary Waveform Generator Modules

User's Guide

M9330-90004

Agilent Technologies

Notices

Manual Part Number

M9330-90004

Edition

Edition, May 15, 2012

Printed in USA

Agilent Technologies, Inc.
1400 Fountaingrove Pkwy
Santa Rosa, CA 95403

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

1 Introducing the Agilent Technologies M933x Series AWG Modules

The M933x Series AWG modules are wideband arbitrary waveform generators (AWGs) capable of creating high-resolution waveforms for radar, satellite, and frequency agile communication systems.

Each channel of the M933x Series AWG modules operate at 1.25 GS/s.

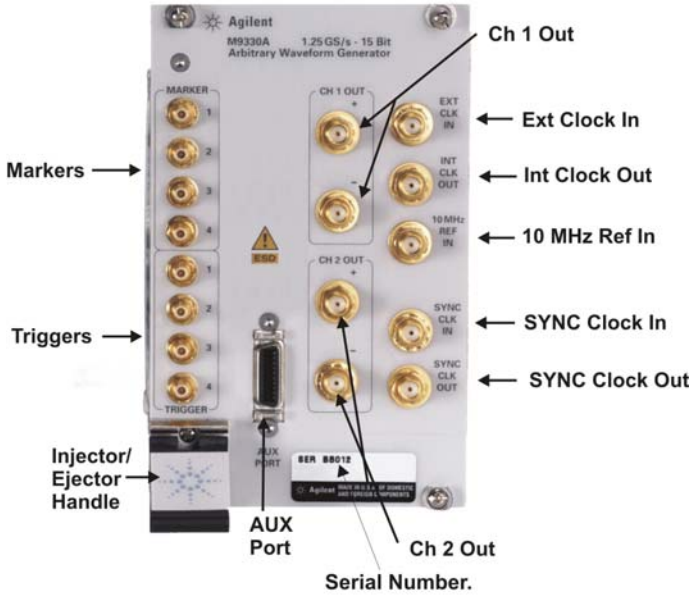
- The M933x Series AWG has 15 bits of vertical reM933xsolution.
- The M9331A Series AWG module has 10 bits of vertical resolution.

Both M933x Series AWG modules are 4-slot 3U PXI modules that offer dual differential output channels to drive both single-ended and balanced designs. The M933x Series AWG modules include a complete software suite to speed waveform development and system integration supporting MATLAB, VEE, LABVIEW, and IVI-C programmatic interfaces. In addition, the following two options are available:

- "Option 300 Dynamic Sequencing" described on page 55
- "Option 330 Direct Digital Synthesis" described on page 59

Agilent M933x Series AWG Modules

Front Panel Interface



MARKERS

There are four SMB female marker output connectors that can be used for triggering or system synchronization. The connectors are 3.3 V TTL/CMOS 30 ohm series terminated. The output is capable of driving a 50 ohm load.

TRIGGERS

There are four SMB female trigger input connectors that are used to control the waveforms in the sequencer and create event-based signal simulation. The connectors support TTL/CMOS, ECL, and PECL logic levels.

Injector/Ejector Handle

This injector/ejector handle is used during installation and removal of M933x Series AWG modules.

CH 1/CH 2 OUT

The CH 1 OUT and CH 2 OUT positive (+) connectors are used for single-ended operation. Use both the positive (+) and negative (–) connectors for differential operation.
Refer to "Signal Conditioning" described on page 47 for more information.

AUX PORT

The AUX port enables "Option 300 Dynamic Sequencing" described on page 55.

EXT CLK IN

Use this 50 ohm SMA external clock in connector to input an external sample clock. It will accept clock rates in the range of 100 MS/s through 1.25 GS/s. Refer to "External Clock" described on page 35 for more information.

An error message will appear if the clock rate does not match the hardware setting, or an external clock is not present.

INT CLK OUT

Use this 50 ohm SMA internal clock out connector to route the internal 1.25 GS/s clock to other test instruments or devices.

10 MHz REF IN

Use this 50 ohm SMA 10 MHz reference in connector to input an external 10 MHz reference.

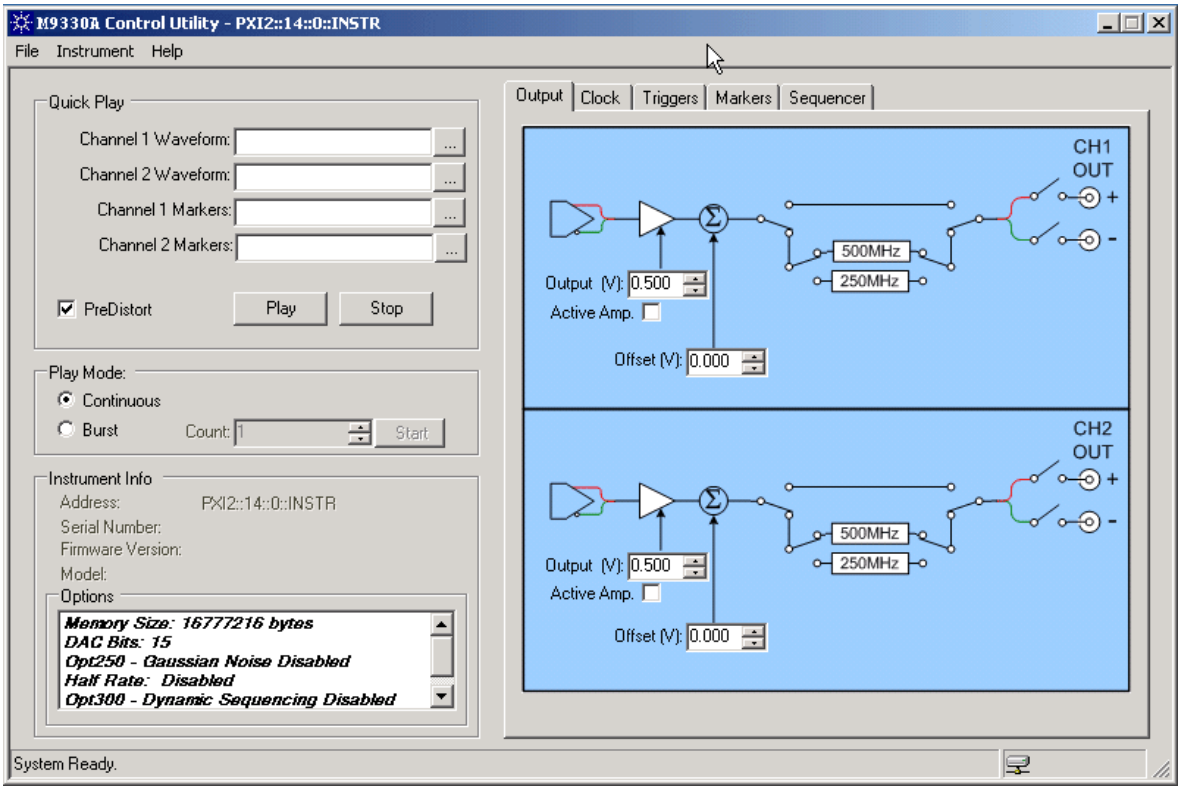
SYNC CLK IN/SYNC CLK OUT

These connectors support synchronization of multiple modules. Refer to "Multiple Module Synchronization" described on page 50.

Graphical User Interface (GUI)

The tab-based graphical interaction of the GUI gives instant access to parameters for the M933x Series AWG modules, making it easy to configure signal output. Each tab is labeled with its contents, enabling quick access to all functions. Figure 1 displays the first level of the GUI. For more information on the GUI, refer to the Agilent M933x Series AWG Modules Online Help. Access this from the application Help menu or in Windows: **Start** > **Programs** > **Agilent** > **M933x** > **Soft Front Panel**.

Figure 1 Graphical User Interface



Getting Started

System Requirements

Hardware

PXI compliant chassis with documentation

PXI compliant embedded controller

or

PXI interface link with related documentation

Agilent E4440A spectrum analyzer or equivalent (system verification)

Supported Operating Systems

- Windows XP, 32-bit
- Windows Vista, 32-bit, 64-bit
- Windows 7, 32-bit, 64-bit

Software Requirements

- Windows .NET Framework, Version 3.5 or greater Redistributable Package
- IVI Compliance Package, Version 4.1 or greater, which includes the IVI Shared Components

Installation of Agilent M933x Series AWG Modules

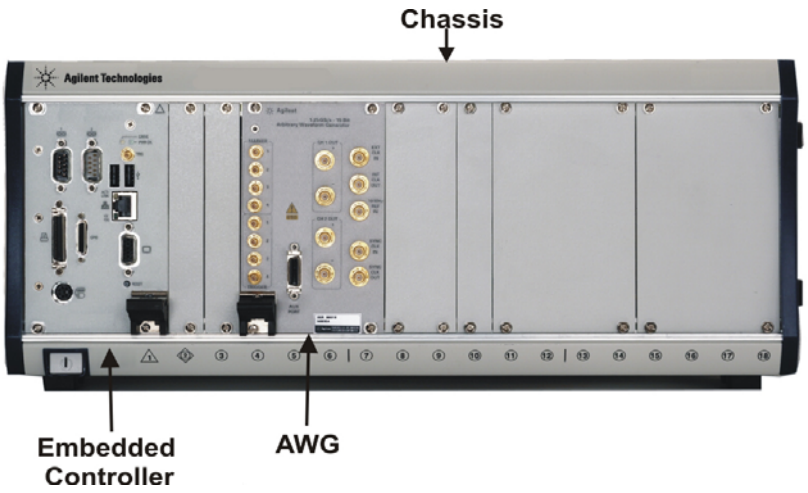
Electrostatic discharge (ESD) can damage the highly sensitive components in M933x Series AWG modules. ESD damage is most likely to occur as M933x Series AWG modules are being installed or when cables are connected or disconnected. Protect the circuits from ESD damage by wearing a grounding strap that provides a high resistance path to ground. Alternatively, ground yourself to discharge any static charge built-up by touching the outer shell of any grounded instrument chassis before touching the port connectors.

Modules in a Chassis with an Embedded Controller

Check the shipment:

- chassis
- embedded controller
- M933x Series AWG modules
- M9330A CD
- *Agilent M933x Series AWG Modules, User's Guide*

This configuration is ready to use. The embedded controller, modules, and all the software were installed prior to shipment.



At least one EMC filler panel must separate the controller from the M933x Series AWG modules. EMC filler panels must be installed in all open slots to ensure proper chassis airflow; this enables effective cooling.

Modules in a Pre-Existing Chassis with Controller

Check the shipment:

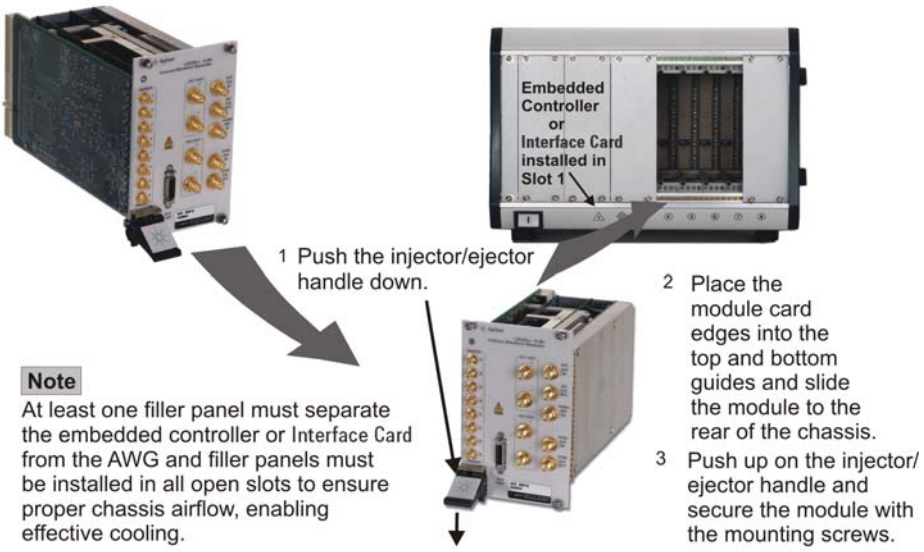
- M933x Series AWG modules
- M9330A CD
- *Agilent M933x Series AWG Modules, User's Guide*

Install the Software:

- Insert the M9330A CD into the CD drive and follow the instructions.
- If the install application fails to come up, navigate to the CD drive and double-click setup.bat.

Once the installation is complete, turn off the controller and make sure the chassis is switched off. Then install the module in the chassis as shown in the figure below.

NOTE: Please make sure that the software is properly installed before installing the module in the chassis



Verifying System Operation

System Set Up

Embedded Controller

1 Start with the controller (PC) turned off.

2 Connect the keyboard, mouse, and monitor to the controller front panel interface.

3 Connect the power cord to the chassis and turn the power on.

Waveform Playback

An Agilent E4440A spectrum analyzer or equivalent is required to view the waveforms.

1 Connect a 10 MHz reference to the M933x Series AWG modules front panel connector. If you are using a PXI chassis, use the backplane 10 MHz reference.

2 Connect the channel 1 positive (+) output to the spectrum analyzer RF input connector.

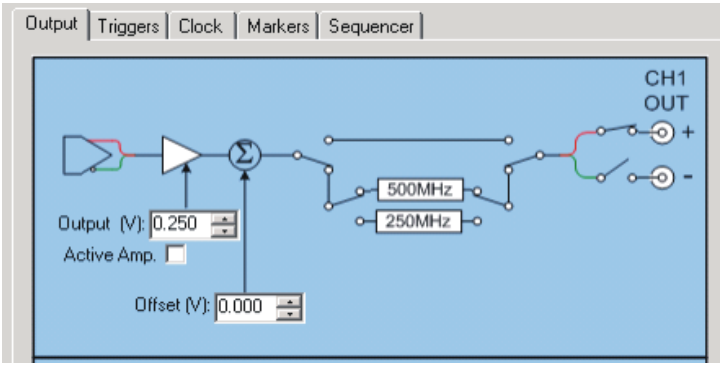
3 Open the user interface by double-clicking the **M9330A Control Utility** icon placed on the desktop during installation.

If there is no icon, go the **Start** > **Programs** > **Agilent** > **M933x** and select **Control Utility** to open the user interface.

4 In the **Output** tab (Figure 2), configure the signal conditioning path to include the 500 MHz reconstruction filter through CH1 OUT (toggle the switches you want to connect) on channel 1 and channel 2. The connection will automatically enable differential mode. Click on the

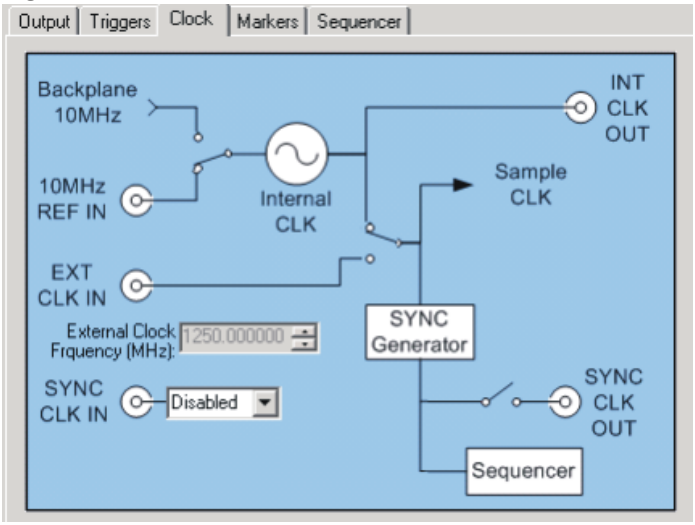
negative (–) node to enable single-ended mode. Notice that the Output drops to 0.250 volts.

Figure 2 Output Tab



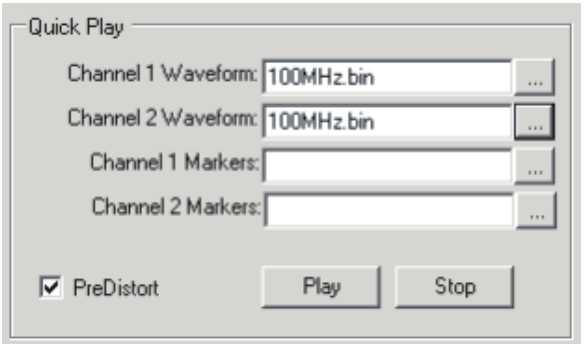
5 Select the **Clock** tab (Figure 3) and configure the internal clock to **10 MHz REF IN**. If you are using a PXI chassis, leave the clock set to the default Backplane 10 MHz.

Figure 3 Clock Tab



6 In the **Quick Play** section of the user interface, browse and select the **100MHz.bin** waveform file for channels 1 and 2.

Figure 4 Quick Play



7 Click **Play.**

The spectrum analyzer cabled to channel 1 should display a spurious free dynamic range (SFDR):

- of at least –65 dBc for the M9330A Series AWG module, as shown in Figure 5
- of at least –50 dBc for the M9331A Series AWG module, as shown in Figure 6

Figure 5 M9330A Series AWG Module

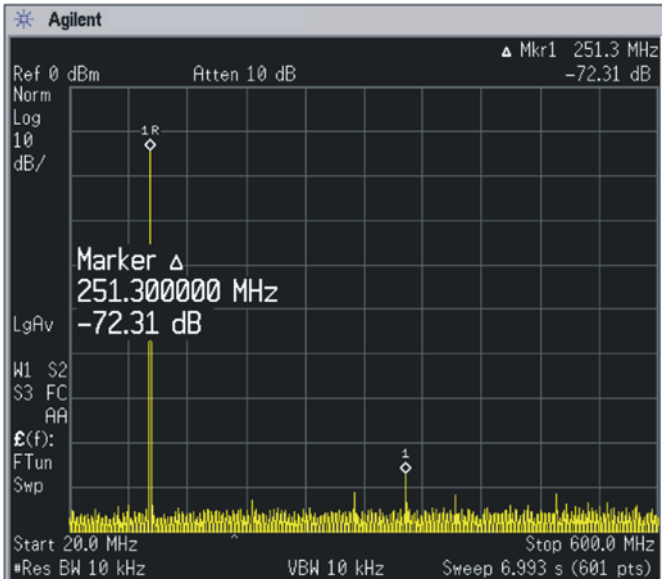
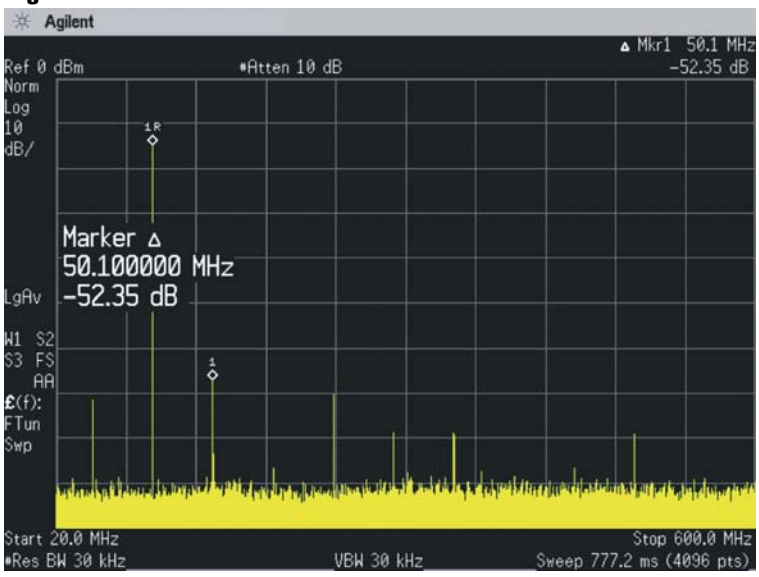


Figure 6 M9331A Series AWG Module



The spectrum analyzer should also display an SFDR of at least –65 dBc for the M9330A Series AWG module and at least –50 dBc for the M9331A Series AWG module when channel 2 positive (+) is connected to the spectrum analyzer RF input connector.

Shutting Down the System

1. Close the M9330A Control Utility.
2. Shut down Windows (**Start > Shut Down**).
3. When Windows is completely shut down, power off the chassis.

Maintenance

To prevent electrical shock, disconnect the instrument and/or system from mains before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally.

2 Basic Operation

This chapter guides you through the basic operation of the M9330A Series AWG modules. Prior to following these procedures, the M9330A Series AWG modules must be installed in a PXI chassis and software must be installed on the controller or PC. Refer to "Getting Started" described on page 11 for complete instructions on how to complete these tasks.

Using the Graphical User Interface

Generating a Single Tone Signal

A spectrum analyzer is required to display the waveforms.

Use the following procedure as a guide to basic single-ended waveform playback with M9330A Series AWG modules. All waveform parameters need to be set prior to waveform playback.

1 Connect a 10 MHz reference from the spectrum analyzer to the front panel connector of the M9330A Series AWG modules. If you are using a PXI chassis, use the Backplane 10 MHz reference.

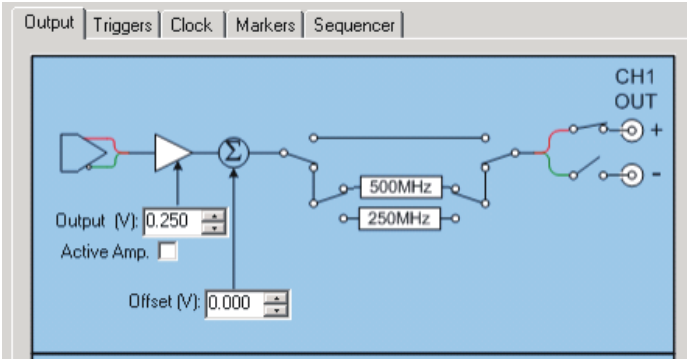
2 Connect the channel 1 positive (+) output to the spectrum analyzer RF input connector.

3 Open the user interface by double-clicking the M9330A icon placed on the desktop during installation.

   If an icon was not placed on the desktop, go to:
   Start > **Programs** > **Agilent** > **M933x** > **M933x SFP**

4 Select the **Output** tab (Figure 7) and connect a single-ended signal conditioning path to CH1 OUT (+) (click on the node that you want to connect).

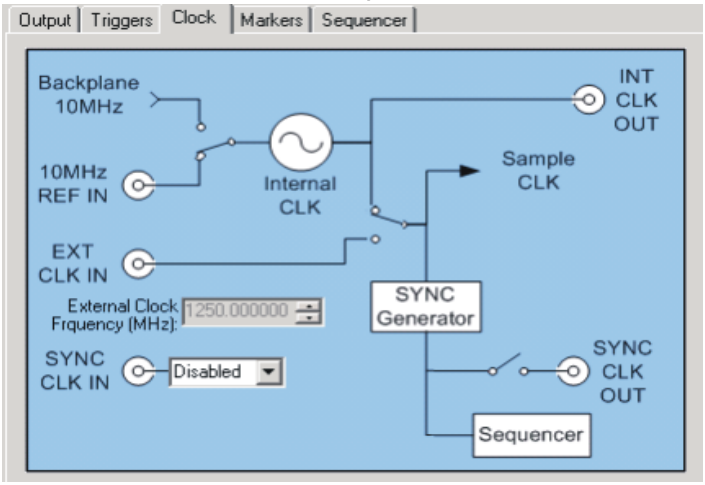
Figure 7 Output Tab



   The connection will automatically enable differential mode. Click on the negative (–) node to open this path and enable single-ended mode. Notice that the default gain value was 0.500 volts. Once you select single-ended mode, the value drops to 0.250 volts. These are maximum values for the modes indicated.

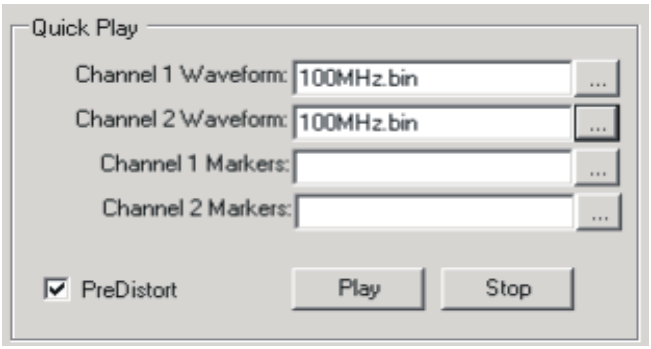
5 Select the **Clock** tab (Figure 5) and configure the **10 MHz REF IN**. For this example, we utilized the 10 MHz reference from the

E4440 spectrum analyzer in step 1. If using a PXI chassis, leave the clock set to the default Backplane 10 MHz.Clock Tab



6 In the **Quick Play** section of the user interface (Figure 8), browse and select the desired single tone waveform file for Channel 1 Waveform. The M9330A Series AWG module accepts data formatted as 16-bit signed integers ignoring the LSB.

Figure 8 Quick Play



Different waveforms can be loaded into channel 1 and 2, but the length of the waveforms must be the same.

Use the default setting for the play mode and predistortion.

Click **Play**.

Figure 9 displays a 100 MHz waveform played back on the M9330A Series AWG module. The SFDR is greater than –70.0 dBc.
Figure 10 displays a 100 MHz waveform played back on the M9331A Series AWG module. The SFDR is greater than –50.0 dBc.

Figure 9 M9330A Series AWG Module

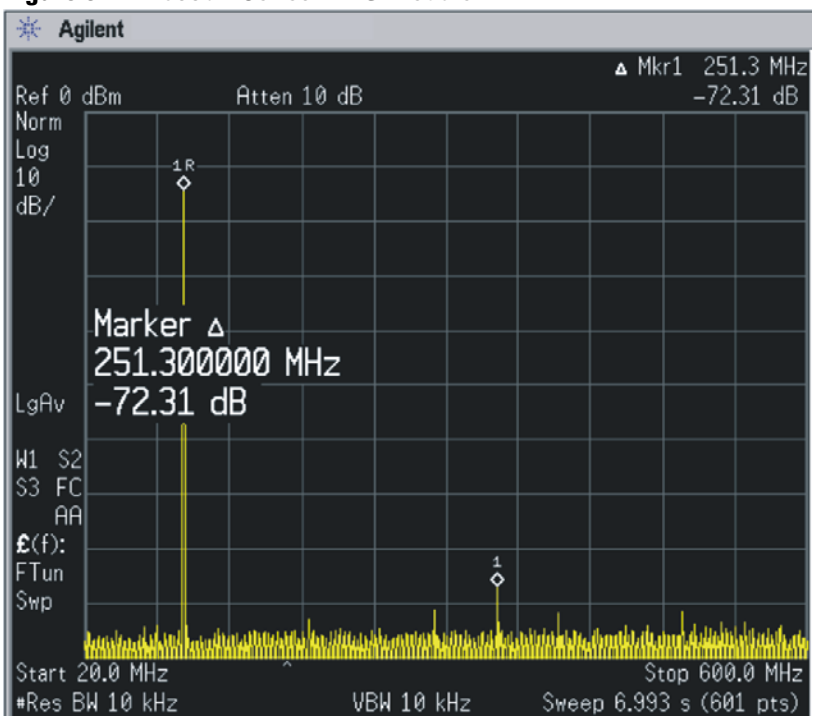
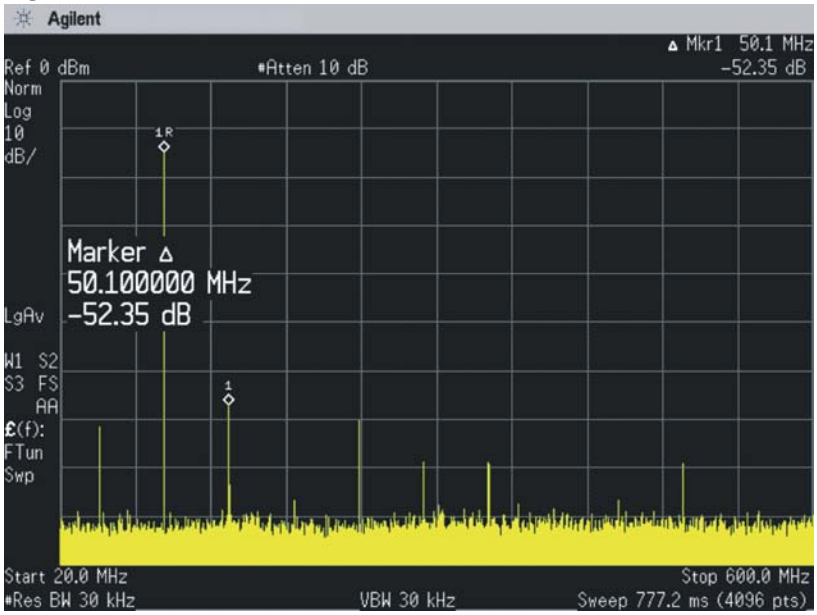


Figure 10 M9331A Series AWG Module



Generating a Multi-tone Signal

Follow steps 1-5 of "Generating a Single Tone Signal" described on page 20 to configure the signal path and clock reference.

In the **Quick Play** section of the Control Utility, browse and select the desired multi-tone waveform for channel 1.

Use the default setting for play mode and predistortion.

Click **Play**.

For this example, a waveform with five tones was used. The intermodulation distortion produced by the five tones played back on the M9330A Series AWG module is less than 60.0 dB, Figure 11. The intermodulation distortion produced by the five tones played back on the M9331A Series AWG module is less than 45.0 dB, Figure 12.

Figure 11 M9330A Series AWG Module, 5 Tone

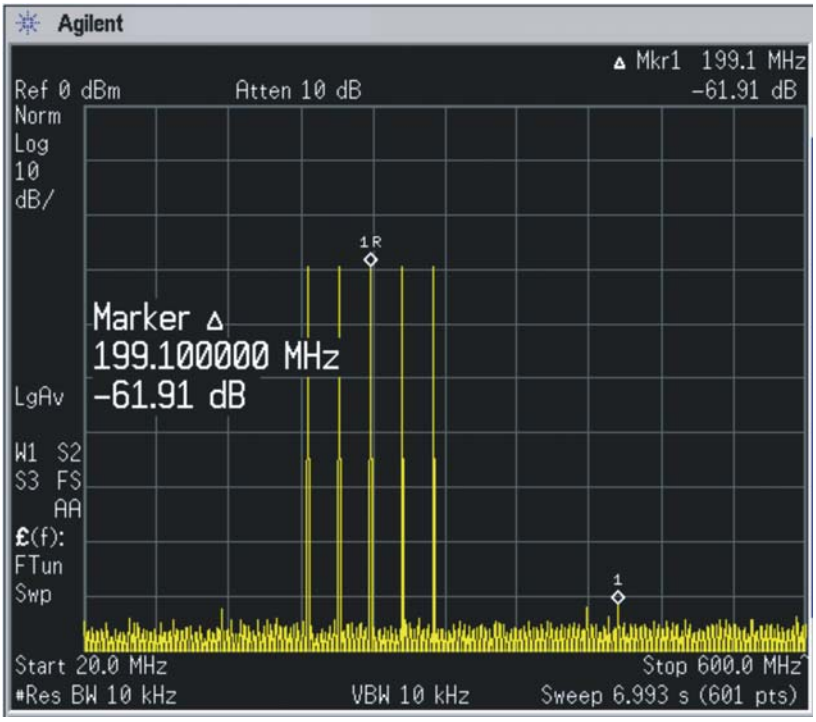
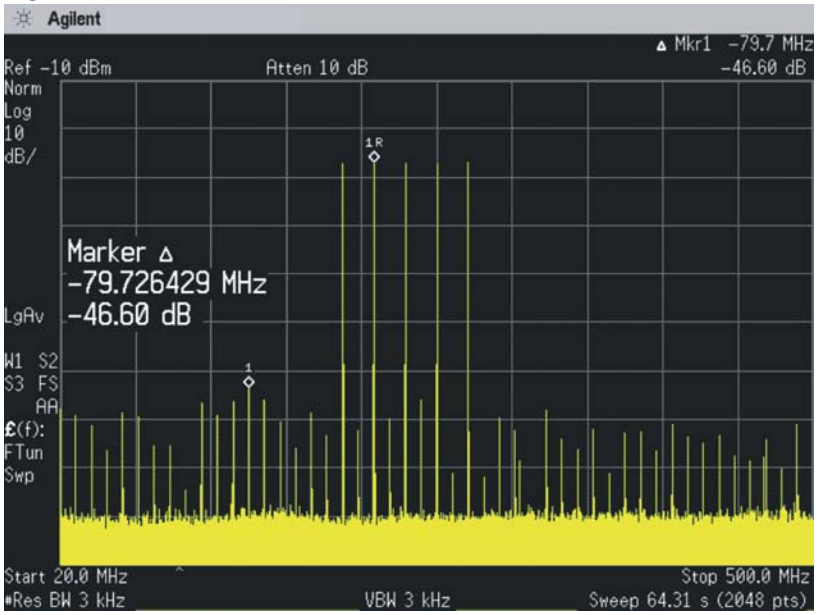


Figure 12 M9331A Series AWG Module, 5 Tone



Creating and Playing a Sequence

Follow steps **1**-**5** of "Generating a Single Tone Signal" described on page 20 to configure the signal path and clock reference.

1 Select the **Sequencer** tab.

2 From the **Segment List** select **Add**. This brings up a **Segment Information** window.

3 Browse and select the **100 MHz** waveform, then click **OK**.

For dual channel sequencing, add the same waveform to both channel 1 and channel 2. Currently, the software does not support independent channel sequencing.

4 Repeat steps **2** and **3** twice, selecting the **2tone** and **5_tone** waveforms.

5 In the **Segment List**, select the **100 MHz** waveform.

6 In the **Sequence Definition** area, select **Add**. This will bring up the **Enter Repetition Count** window.

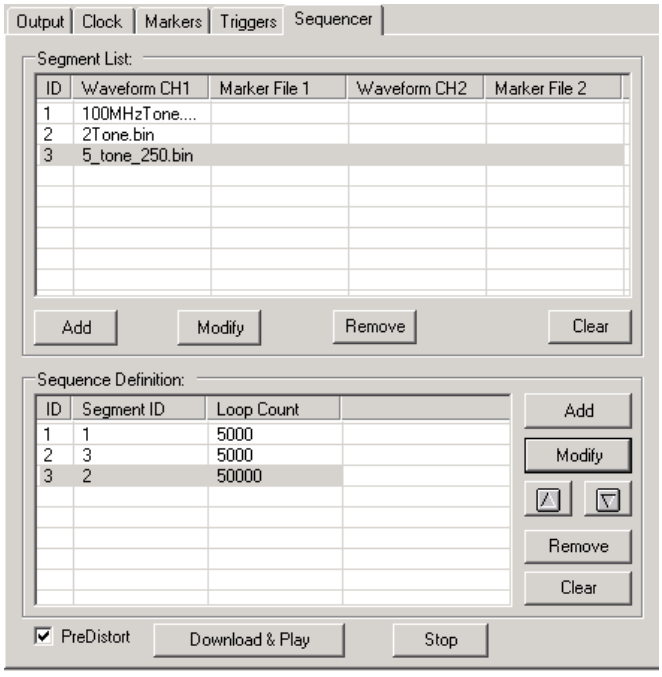
7 Enter **5000** repetitions and click **OK**.

8 Repeat steps **5**, **6**, and **7** for the **2tone** and **5_tone** waveforms.

9 In the **Sequence Definition** area, select **Segment ID 2** and move it below **Segment ID 3** using the down arrow.

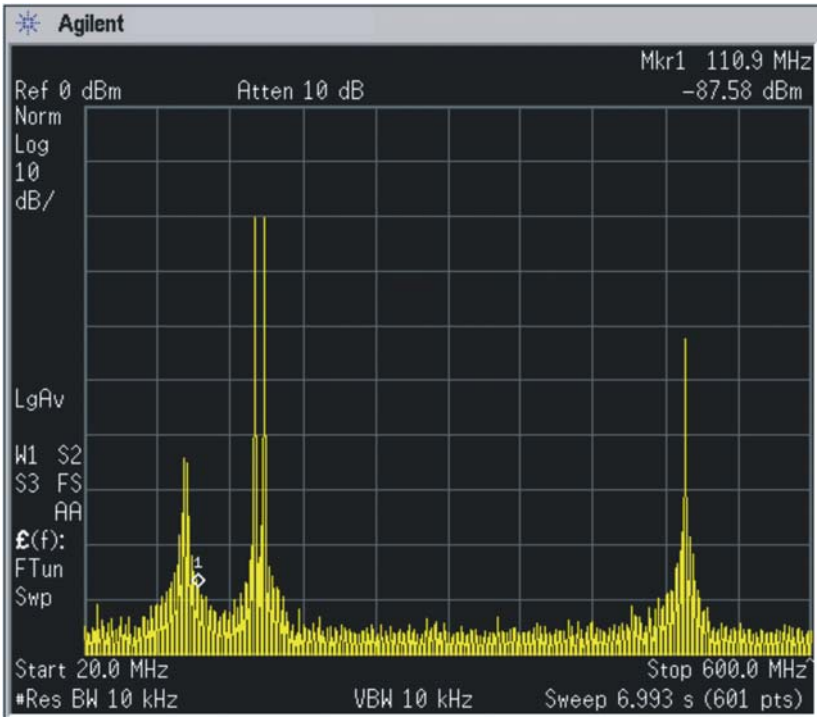
10 Click **Modify** and change the count to **50000**. The sequencer tab should look like Figure 13.

Figure 13 Sequencer Tab



11 Click **Download & Play**. The spectrum of the sequence should be similar to the one shown in Figure 14.

Figure 14 Playback of a Sequence



Synchronizing Two Agilent M9330A Series AWG Modules

Internal Clock Synchronization Using Continuous Mode

When synchronizing two modules using the internal clock, one unit is designated as the Master and the other unit is designated as the Slave. The Master unit sources the sample clock and the sync clock signals. These signals are split and fed to the synchronized modules (the Master as well as the Slave).

The internal sample clock operates at 1.25 GHz and provides the final retiming of the analog output from each M9330A Series AWG modules. Any skew in the sample clock cable delays between the modules will result in the same skew in the analog outputs.The sample clock signal is split with a matched passive divider and the cable lengths are matched. The resulting skew is small and repeatable.

Required Equipment

- M9330A Series AWG modules
- PXI chassis
- Embedded controller

Cable Kit

- SMB Cable Assembly, (3 each)
- SMB Adapter Tee M-M-M (1 each)
- SMA Cable Assembly, 10 in (4 each
- Power Divider, 11636B (2 each)

Customer Furnished Cables

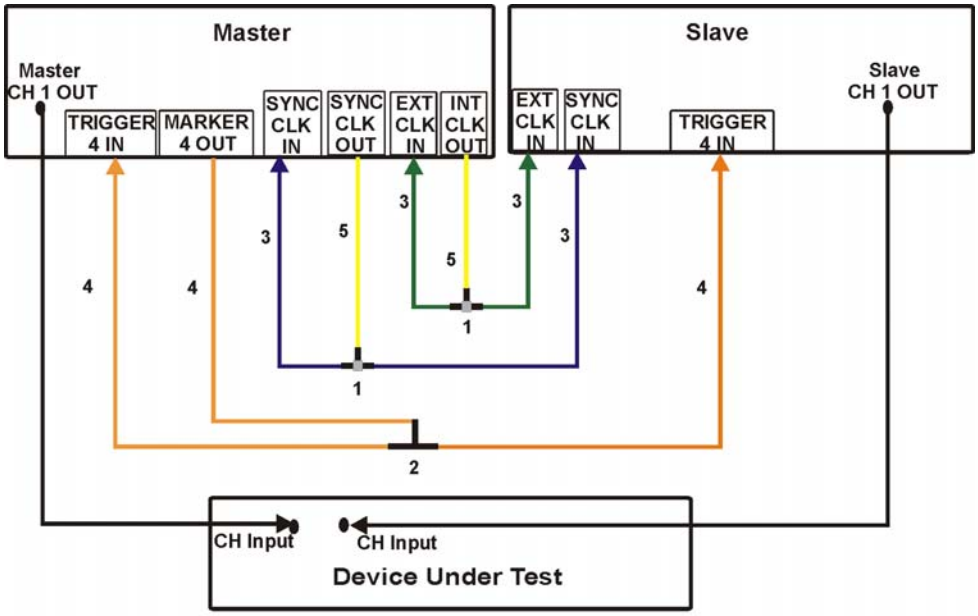
- SMA to BNC Cable (2 each, equal length)

Procedure Using a Software Marker

1 Start with the system turned off.

2 Connect equipment cables as shown in Figure 15.

Figure 15 Cabling for Two Module Synchronization



1— Power divider
2— SMB Adapter Tee
3— SMA Cable Assembly 10 in
4— SMB Cable Assembly
5— Adapter

Note: The CH 1 OUT customer furnished cables from the Master and Slave modules must be of equal length.

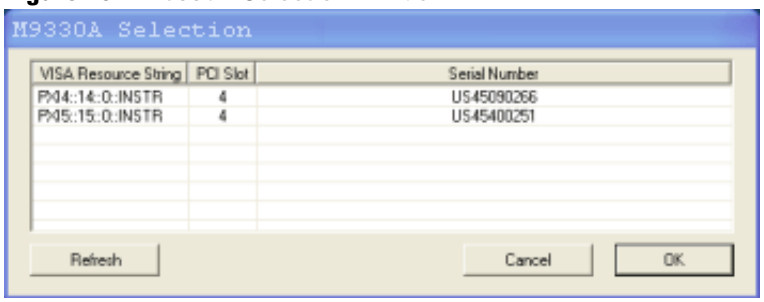
3 Turn the system on.

Selecting the Master Unit

1 Open an **M9330A Control Utility** session (double-click the M933x SFP icon on the desktop).

2 Highlight the **Master** unit in the **M9330A Selection** window list and click **OK**.

Figure 16 M9330A Selection Window



3 Select the desired signal conditioning path.

4 Select the desired waveform file.

5 Select the **Clock** tab.

6 From the **SYNC CLK IN** drop-down list, select **Master**.

Notice the following changes to the graphical user interface that are automatically configured when the Master unit is assigned:

Clock Tab

- the internal clock is no longer driving the sample clock
- the sample clock and sync clock out are driven by the external clock in signal
- the sync clock in signal communicates with the sequencer

Marker Tab

- Marker 4 is assigned to a Software marker and is grayed out

Trigger Tab

- Start trigger is assigned to Trigger 4 and is grayed out

Selecting the Slave Unit

1 Open an M9330A Control Utility session (double-click the M933x SFP icon on the desktop).

2 Highlight the **Slave** unit in the **M9330A Selection** window list and click **OK**.

3 Select the desired signal conditioning path.

4 Select the desired waveform file.

5 Select the **Clock** tab.

6 From the **SYNC CLK IN** drop-down list, select **Slave**.

Notice the following changes to the graphical user interface that are automatically configured when the Slave unit is assigned:

Clock Tab

- the internal clock is disabled
- the sample clock is driven by the external clock in signal
- the sync clock out is disabled
- the sync clock in signal communicates with the sequencer

Trigger Tab

- Start trigger is assigned to Trigger 4 and is grayed out

Initiating Synchronous Playback

1 In the **Quick Play** area of the **Slave** GUI, select **Play**. This arms the waveform playback.

2 In the **Quick Play** area of the **Master** GUI, select **Play**. This initiates synchronous waveform playback.

3 You can view the output on an oscilloscope by setting Marker 1 on the Master module to Waveform Start and cabling the marker output to trigger the oscilloscope.

Using the Programmatic Interfaces

IVI-C Driver Functionality

The IVI Foundation's class driver specification for function generators has been the model for the features in M9330A Series AWG modules. This includes the recommended method to incorporate attributes for instrument-specific functions.

Please refer to IVI-4.3 IviFgen Class Specification and IVI-3.1 Driver Architecture Specification for more information. These can be found at:

www.ivifoundation.org/Downloads/Specifications.htm

A set of API Functions and Attributes can be found in the M9330A Help system. Go to:

Start > **Programs** > **Agilent IVI Drivers** > **AgM933x > Documentation**

MATLAB Interface

Requirements for Using the AgM933x in MATLAB

The following components are required to be installed (at the time the AgM933x driver installer runs), in order to use the AgM933x driver in MATLAB:

- MATLAB R2009a (or later)
- MATLAB Instrument Control Toolbox

Note: Mathworks, Inc does not currently support IVI-COM type drivers in it's 64-bit MATLAB product. For this reason the Agilent MD1 driver is not supported under MATLAB 64-bit. However MATLAB 32-bit continues to be compatible with the MD1 driver (even on Windows 64-bit platforms). For more information, please contact Agilent support via:
www.agilent.com/find/assist

Accessing Class-Compliant Functionality

Access to class-compliant functionality in an IVI driver is provided by the MATLAB Class objects. You use the appropriate MATLAB Class to create an instrument object.
For more information please refer to the IVI Driver help.

Accessing Instrument-Specific Functionality

In order to access the instrument-specific functionality of the AgM933x driver, the driver must be directly referenced. Note that directly referencing the AgM933x driver provides access to the class-compliant functionality as well.

Programming examples

Programming examples can be found in:

Program Files\IVI Foundation\IVI\Driver\M933x\Examples.

3 Theory of Operation

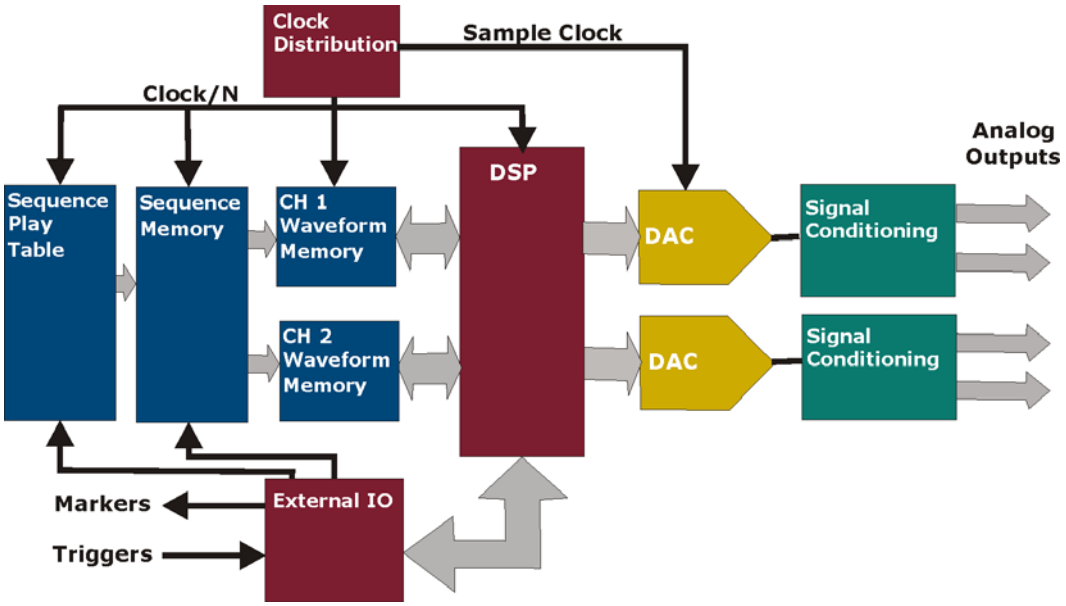
The M9330A Series AWG modules are capable of creating high-resolution waveforms for radar, satellite and frequency agile communication systems. Each channel of the M9330A Series AWG modules operate at 1.25 GS/s. The M9330A Series AWG module features 15 bits of vertical resolution and the M9331A Series AWG module features 10 bits of vertical resolution. The M9330A Series AWG modules are 4-slot 3U PXI that offer dual differential output channels to drive both single-ended and balanced designs.

The M9330A Series AWG modules include a complete software suite to speed waveform development and system integration supporting MATLAB, VEE, LABVIEW, and IVI-C programmatic interfaces. In addition, the following options are available:

- "Option 300 Dynamic Sequencing" described on page 55
- "Option 330 Direct Digital Synthesis" described on page 59

Theory of Operation

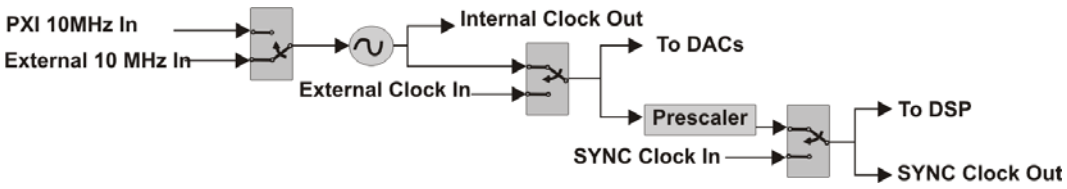
Block Diagram



N is the Sync clock prescaler divide ratio. Refer to Synchronous Triggers.

Clock I/O

External 10 MHz In



A 10 MHz reference is required when using the internal clock.

The Backplane 10 MHz is only available when using a PXI chassis. Compact PCI typically do not have a 10 MHz backplane reference.

Internal Clock

The high-performance 1.25 GHz oscillator provides the internal sample clock for the M9330A Series AWG modules.

External Clock

An external sample clock can be input through the EXT CLK IN connector. The external sample rate must be between 100 MS/s and 1.25 GS/s. To achieve the optimal signal performance on the analog output of M9330A Series AWG modules, use an external clock with a phase noise floor below –155 dBc/Hz and a power setting of approximately 0 dBm.

An error message will appear if the clock rate does not match the hardware setting, or an external clock is not present.

Internal Clock Out

The 1.25 GS/s low noise internal sample clock is output through the INT CLK OUT connector and routed to other M9330A Series AWG modules or test equipment.

SYNC Clock In/SYNC Clock Out

The SYNC CLOCK IN and SYNC CLOCK OUT are used for the synchronization of multiple modules. Refer to "Synchronization Using an Internal Clock" described on page 50 and "Synchronization Using an External Clock" described on page 52.

Waveform Playback

Waveforms

Single waveforms are played back in one of two modes:

- **Continuous**
  The waveform repeats indefinitely.
- **Burst**
  Once a trigger is received, the waveform repeats a specified number of times.

Waveform Sequencer Function

Sequencing provides a method of waveform memory compression using a play table, sequencer memory, and waveform memory. The sequencer controls how waveforms are accessed and performs the following functions:

- determines the order of play for waveforms stored in memory
- enables the construction of long waveforms from shorter or repeated segments
- responds to external triggers
- offers several modes of segment advance
- outputs markers

Sequencer Memory

The sequencer memory contains instructions on how to play through the waveform memory. It can hold up to one million segments (waveforms with a specified loop count).

Waveform Memory

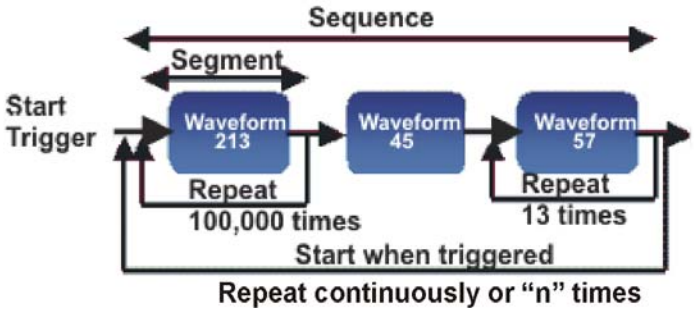
The waveform memory contains Channel 1 and Channel 2, and output marker data.

Basic Sequencing

A sequence is a sequential list of segments and may occur anywhere in the sequence memory.

The waveform playback of each channel is directly controlled by the sequencer. The sequencer supplies the memory pointers necessary to create analog signals from the digital data stored in memory. In addition, the sequencer provides the capability to create sequences made of multiple waveform segments. This is helpful when constructing long waveforms with repeating segments. A long waveform might consist of repetitive data that can be stored as single segments and repeated in the sequencer. This extends the waveform play time achievable with the available memory.

Figure 17 Sequence Example



Basic sequencing can be done using the software Control Utility GUI or through the programatic interfaces.

The M9330A Control Utility GUI only supports basic sequencing. Advanced sequencing features can only be accessed through the programatic interfaces.

Playback

There are two playback modes for basic sequencing:

- **Continuous**
  The sequence repeats indefinitely or until an stop trigger is received.
- **Burst**
  The sequence is repeated a predefined number of times. This mode requires a start trigger.

A total of 32,768 unique waveform sequences can be defined. Segments have a minimum length of 128 samples and a granularity of 8 samples. A sequence must contain at least one segment and can have up to a maximum number of 1 million (2^{20}-8) segments. Each waveform segment is played out according to its segment definition. Each segment can be

configured to repeat over 1 million times (2^{20}). After the last segment loop is executed, the entire sequence can repeat continuously or for the predefined burst count.

Advanced Sequencing

Advanced sequencing is only available through the programatic interfaces.

Advanced sequencing enables the grouping of sequences into scenarios in a way that is similar to how segments are grouped in sequencing. With scenarios you gain more control of waveform playback.

Scenario Pointer Source

A scenario handle is written to by the host processor that addresses scenarios. The handle can be written to at any time including while a scenario is playing. A valid Start trigger or Jump trigger starts the scenario specified by the handle.

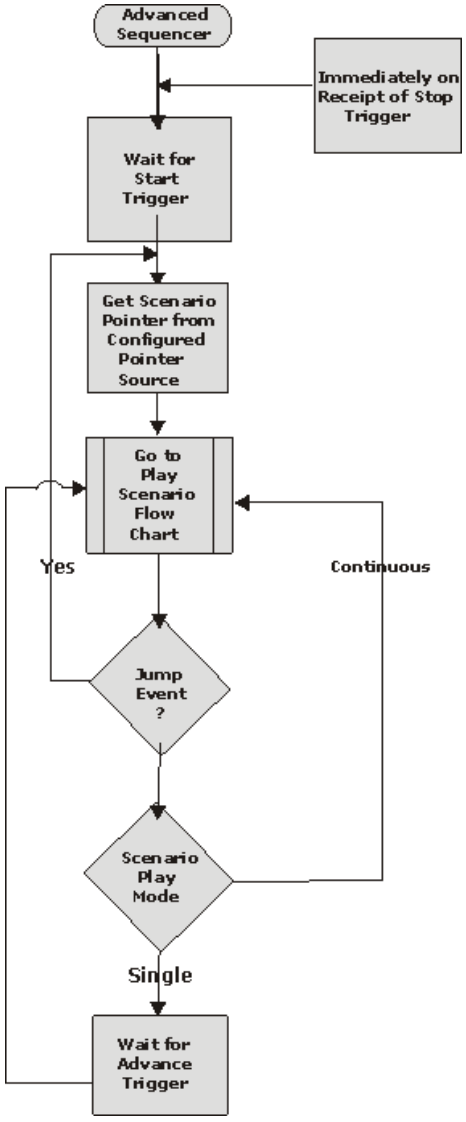
Scenario Advance Mode

The M9330A Series AWG modules can be configured to play a scenario once or continuously after starting.

- **Single**
  The scenario plays once and then waits for an advance trigger. While waiting for a trigger, the value of the last waveform continues to play. After receiving an advance trigger, the scenario is then played again.

- **Continuous**
  The scenario repeats indefinitely until it is stopped or a scenario jump trigger is received.

Figure 18 Advanced Sequencer Flow Chart



Waveform Advancement

In basic sequencing, waveforms always advance to the next waveform automatically after the specified number of repetitions. With advanced sequencing, waveforms can be configured to advance in one of four ways.

- **Automatic**
  The waveform automatically advances to the next waveform after completing the specified number of loop repetitions.

- **Continuous**
  The waveform is played continuously until a waveform jump trigger is received. After a trigger is received, waveform playback completes before advancing to the nest segment. The waveform loop repetition count is ignored.
- **Single**
  The waveform plays once and waits at the end of the waveform playback for a trigger. The waveform is played for each trigger until the number of waveform loop repetitions is met. The next trigger will advance to the next waveform. When the waveform loop repetition count is one, a single trigger will advance to the next waveform. While waiting for a trigger, the last value of the waveform loop continues to play.
- **Repeat**
  The waveform plays repeatedly until the number of waveform loop repetitions is met, then waits for a trigger. The next trigger will advance to the next waveform.

Scenario Jump Mode

The scenario jump mode determines how the AWG responds to a scenario jump input. A scenario jump has very predictable behavior. There are three types of jump modes:

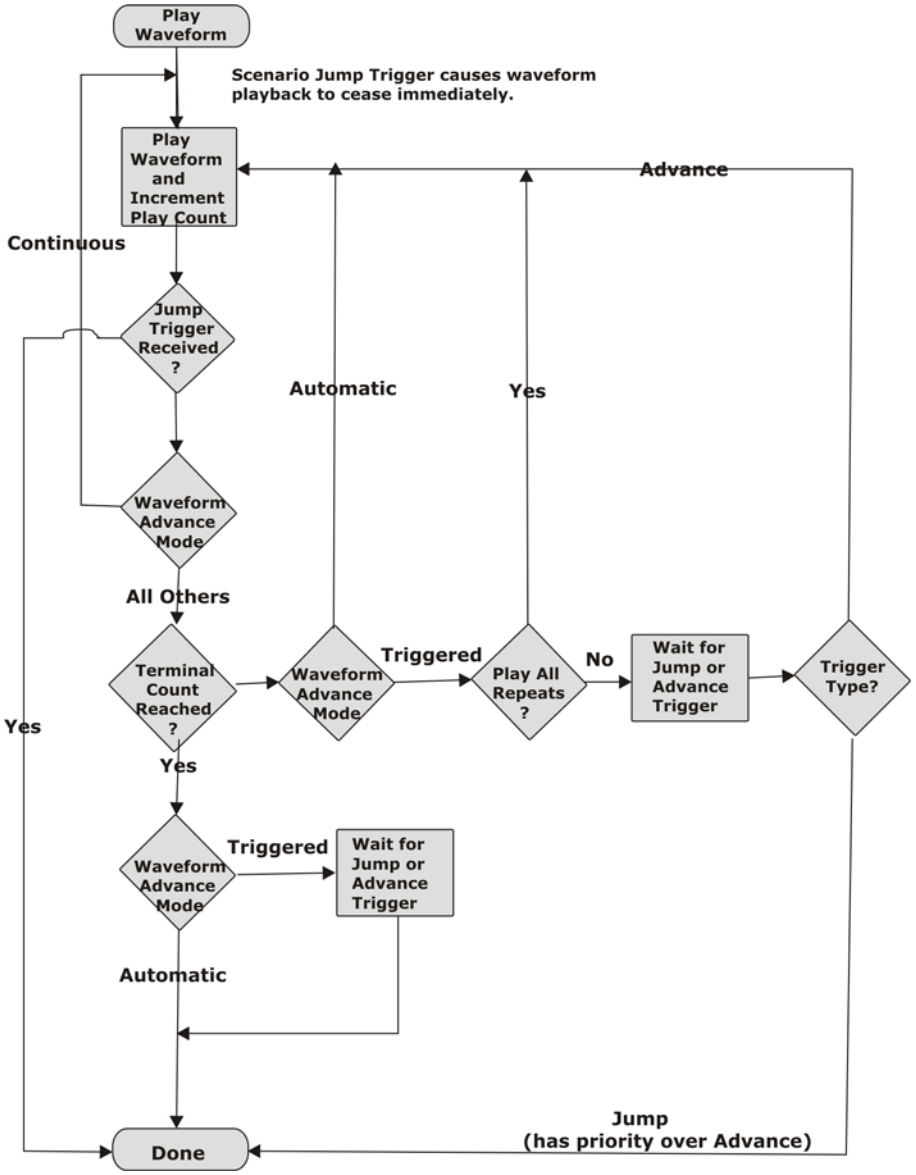
- **Immediate**
  The scenario starts or jumps immediately (with latency).
- **End of Waveform**
  The current waveform, including repeats, is completed before jumping to the new scenario.
- **End of Scenario**
  The current scenario is completed before jumping to the new scenario.

Scenario Start/Jump Trigger Source

It is possible to start a scenario, or to jump to a new scenario using one of five inputs. There are four external trigger inputs and a host trigger source. The host trigger source is a register in the play table that can be written to by the host processor. The host processor provides the user a way to start the scenario, or create a jump event. The latency for a scenario jump is established by the jump mode.

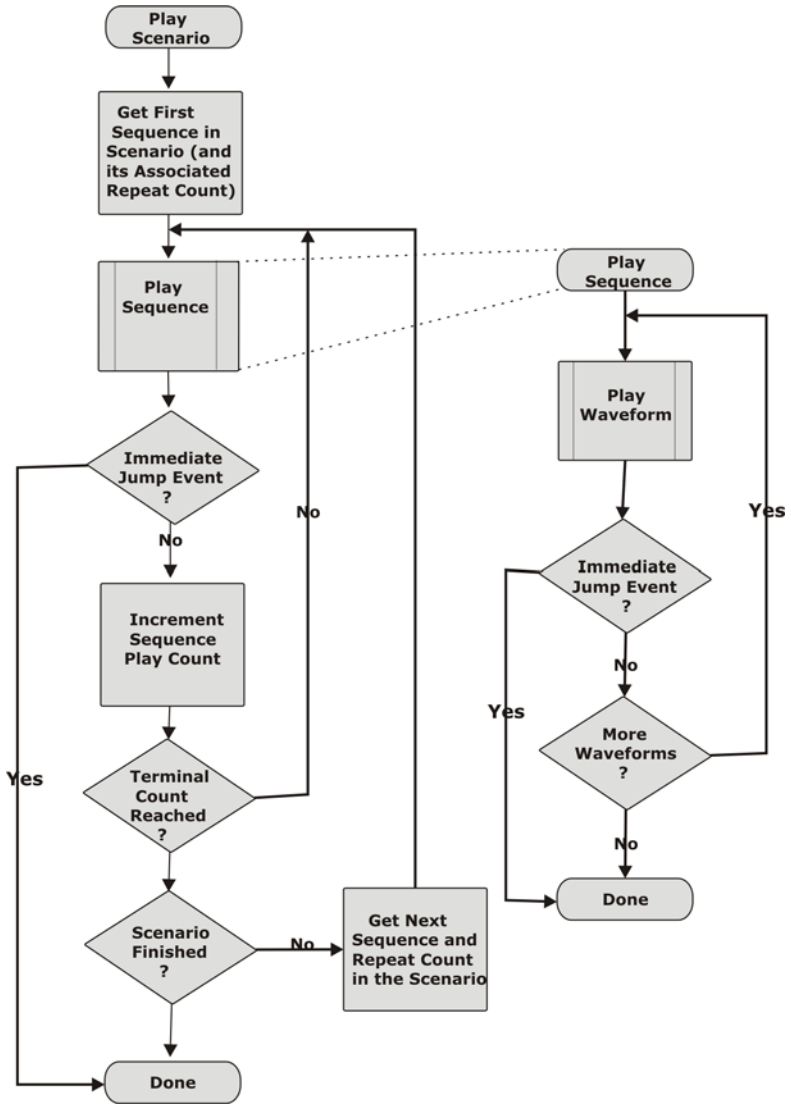
Refer to Figure 19, “Waveform Play Flow Chart,” on page 41, and Figure 20, “Scenario and Sequence Play Flow Charts,” on page 41.

Figure 19 Waveform Play Flow Chart



The Jump Trigger condition is satisfied either by a waveform jump event, or by a scenario jump trigger event when the scenario jump mode is set to “End of Waveform.”

Figure 20 Scenario and Sequence Play Flow Charts


Play Scenario
Get First Sequence in Scenario (and its Associated Repeat Count)
Play Sequence
Immediate Jump Event ?
No
Yes
Increment Sequence Play Count
Terminal Count Reached ?
No
Yes
Scenario Finished ?
No
Get Next Sequence and Repeat Count in the Scenario
Done
Play Sequence
Play Waveform
Immediate Jump Event ?
No
Yes
More Waveforms ?
No
Yes
Done

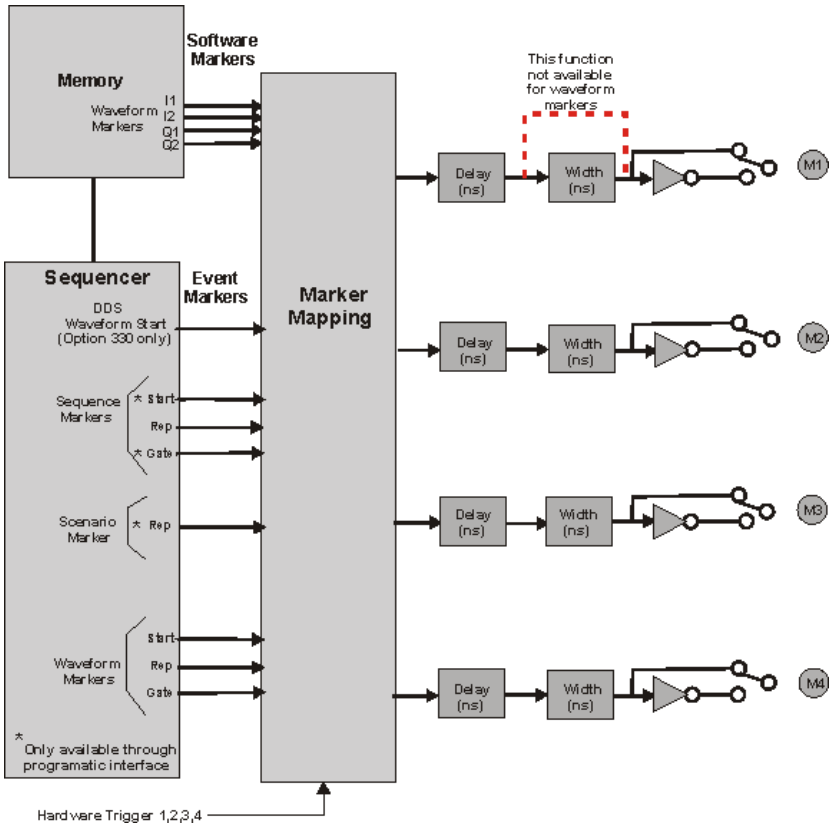
Markers

M9330A Series AWG modules provide four marker output connectors that can be used for system synchronization and triggering.

The following markers can be enabled:

- Ch 1 Memory Marker 1 and Memory Marker 2
- Ch 2 Memory Marker 1 and Memory Marker 2
- Waveform Start, Repeat, and Gate
- Sequence Start, Repeat, and Gate
- Scenario Repeat
- Software
- Hardware Trigger 1-4
- DDS Waveform Start (Option 330 only)

Figure 21 Marker Block Diagram



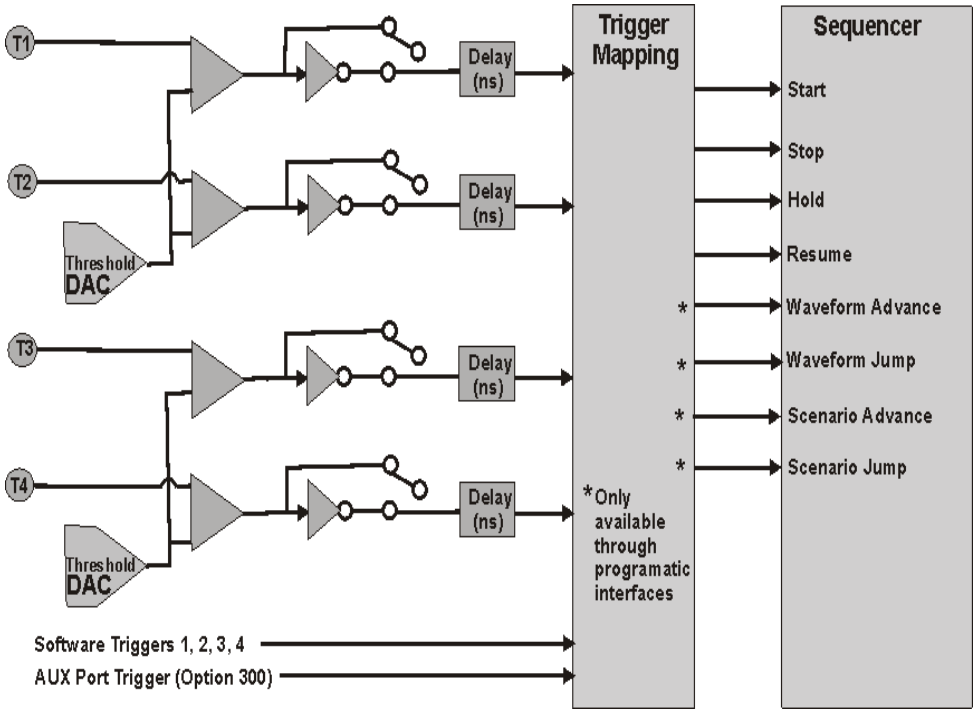
Marker outputs are aligned with the analog output of the AWG. Marker. Markers can be set in the sequencer to be at any point in the data with a positive or negative polarity. Marker widths, except those derived from waveform memory, can be set in increments of the SYNC clock (–8 to

247 clocks). The marker delay function uses the input value to calculate the delay to the nearest 1/4 SYNC clock cycle. The sequencer is capable of outputting nine markers, which can be multiplexed to the four marker outputs. The Sequence Start, Sequence Gate, and Scenario Repeat markers are only available through the programatic interfaces.

Triggers

M9330A Series AWG modules have four trigger inputs that can be used to control waveforms in the sequencer. Hardware trigger inputs may be configured to generate events on the rising or falling SYNC clock edges, but not both at the same time. The trigger threshold can be set between –4.5 and +4.5V. Ports 1 and 2 have a common threshold, and ports 3 an 4 have a common threshold. These two common thresholds are not shared and can be set independently.

Figure 22 Trigger Block Diagram



Trigger delays can be set in increments of the SYNC clock (0-255 clocks).

The trigger input can be configured to initiate the following events through the software Control Utility:

- **Start** - Starts playback at the beginning of the waveform
- **Hold** - Holds at the end of the waveform
- **Stop** - Stops playback
- **Resume** - Resumes playback at the point in the waveform that play was held or stopped

The Waveform Advance, Scenario Advance, Waveform Jump, and Scenario Jump triggers are only available through the programatic interfaces.

Synchronous Triggers

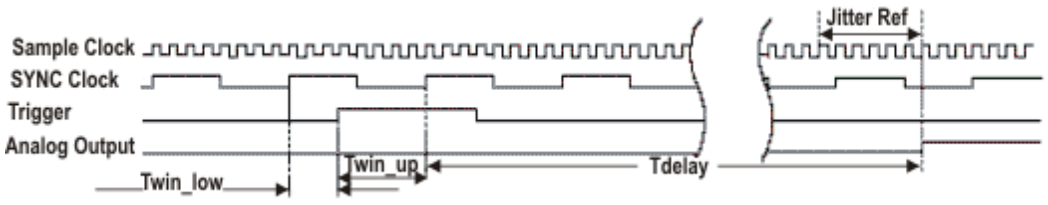
Triggers are registered into M9330A Series AWG modules using the SYNC clock. The SYNC clock is nominally at the sample clock frequency divided by 8. However, at lower sample rates, an internal variable modulus prescaler selects other binary divide ratios: 8, 4, 2, and 1. In general, the SYNC clock frequency is always in the range of 78.13 MHz to 156.25 MHz. The input clock frequency ranges and prescaler divide ratios are as specified in Table 1.

Table 1 Synchronous Triggers

Sample Clock Frequency	SYNC Clock Prescaler Divide Ratio
625 MHz - 1.25 GHz	8
312.5 MHz – 625 MHz	4
156.25 MHz – 312.5MHz	2
100 MHz – 156.25 MHz	1

It is necessary to insure that the correct timing relationships are achieved to guarantee consistent synchronous trigger operation. The trigger input must occur within a valid window with respect to the SYNC clock. The window is specified by two times: Twin_low —the minimum trigger delay after the prior SYNC clock edge; and Twin_up — the minimum trigger setup before the next SYNC clock edge. These are specified for the trigger input relative to the SYNC clock output. The trigger must be a minimum of two SYNC clock cycles long. The trigger timing is specified relative to the rising edge of the SYNC clock. The analog output from M9330A Series AWG modules is then produced a fixed number of sample clock cycles (plus a small fixed propagation delay) after the first rising edge of the SYNC clock after the trigger goes active. Since the analog output is retimed by the sample clock, the reference for jitter measurements is the sample clock, as shown in Figure 23.

Figure 23 Synchronous Trigger Timing Diagram

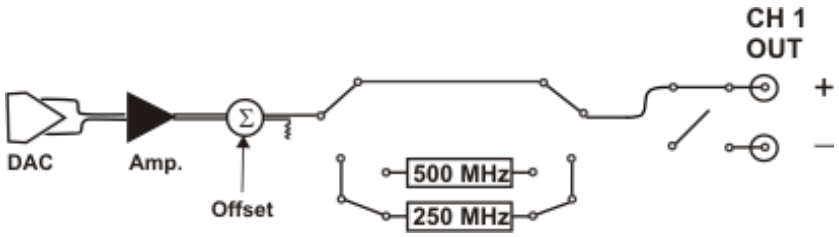


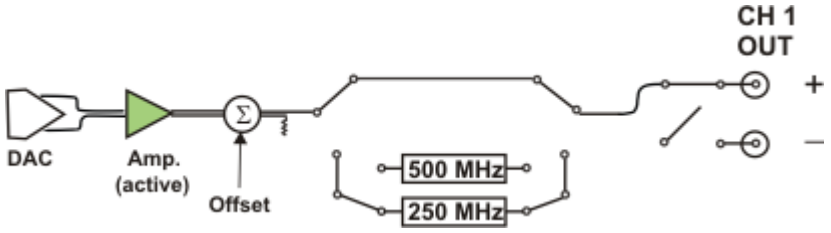
Signal Conditioning

Single-Ended Mode

Single-ended mode has two modes of operation with signal output through the positive (+) port. The negative port (–) is reserved for differential mode.

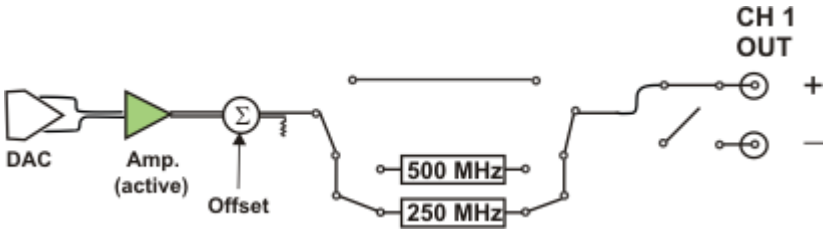
Passive mode has an adjustable output level of up to 0.5 Vp-p
This mode gives the greatest single-ended signal fidelity because there is a balun in the path that suppresses the second order harmonic.





Active mode has an output level of up to 1.0 Vp-p and +0.2 Vp-p offset range when the amplifier is activated
The active mode trades off signal fidelity for an increase in signal power.

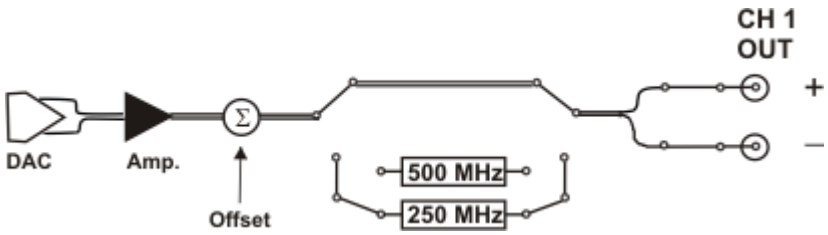
There are two internal reconstruction filters, 250 MHz and 500 MHz, that can be inserted in the signal path of either mode.



Differential Mode

The differential mode has an output level of up to 0.5 Vp-p. This mode provides exceptional signal fidelity into true differential inputs (which provide common mode rejection). A larger differential output voltage is also obtained without the use of the amplifier. To preserve signal purity, the active amplifier cannot be used in differential mode. Differential mode is not recommended when driving single-ended loads since the second order distortion is degraded. If you choose to drive single-ended loads, you must terminate the negative (–) port of the channel with a 50 ohm load.

Adjustable output voltage and offsets as well as reconstruction filters can be used in differential model.



Digital Predistortion

The predistortion function compensates for the variation in the magnitude of the output response as a function of frequency. This variation is the result of the sin x/x (sinc) roll-off of the internal DAC and the frequency response of the reconstruction filter. The correction method uses filters to level the amplitude response and to create a linear phase response at the front panel of M9330A Series AWG modules. This precess attenuates the signal as a function of frequency, but cannot increase the signal above the maximum output voltage. Therefore, it is necessary to attenuate the lower frequency signals. This results in a reduced output voltage and dynamic range at all frequencies, but with uniform response across the full frequency range.

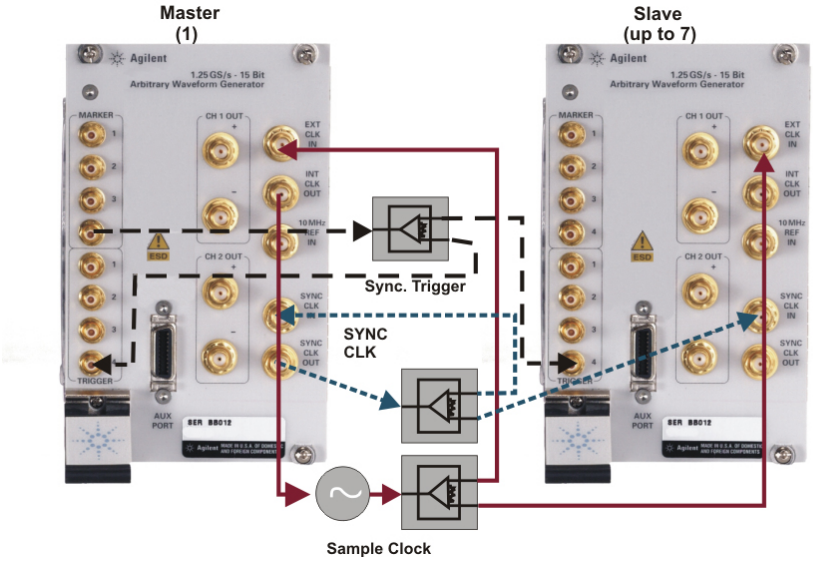
Multiple Module Synchronization

Within the M9330A Series AWG modules, the two channels are synchronized by design. Some systems, such as phased array radar, require more than two synchronized channels. The M9330A Series AWG modules are designed to support the synchronization of up to 16 channels through the use of eight M9330A Series AWG modules. Synchronization of multiple modules can be achieved using either the internal clock or an external clock.

Synchronization Using an Internal Clock

In synchronizing multiple modules using the internal clock, one unit is designated as the Master and the other units are designated as Slave units. The Master unit sources the
following signals: Sample clock, SYNC clock, and the Sync Marker. These signals are all split and fed to each of the synchronized modules (the Master as well as the Slaves).

Figure 24 Master/Slave Synchronization Using an Internal Clock



The internal sample clock is at 1.25 GHz. The sample clock provides the final retiming of the analog output from each of the M9330A Series AWG modules. Any skew in the sample clock cable delays between the multiple modules will result in the same skew in the analog outputs. Typically, the sample clock signal is split with a matched passive splitter and the cable lengths are matched to better than 5 mm. The resulting skew is small and repeatable. If desired, the skew can be measured and calibrated (along with any phase shifts in cables on the AWG outputs) by adding fixed delay offsets to the waveforms.

The SYNC clock is used internal to M9330A Series AWG modules to clock the internal data generator and to clock in the synchronous triggers. When using the internal clock, the SYNC clock has a frequency of 1/8th the sample clock rate (156.25 MHz). When synchronizing multiple units, the SYNC clock output must be enabled in software (in the Master) and the external SYNC clock input selected in all the modules. The SYNC clock signal is split passively and distributed with low skew. The SYNC clock output level and the input sensitivity support up to a 1 to 8 split (fan-out) using matched 50 Ohm splitters (6 dB loss per 1 to 2 splitter). There is a specific SYNC cable length that is required as a function of the sample clock frequency. Several different lengths can be used, provided they are integer multiples of one half of a SYNC clock period.

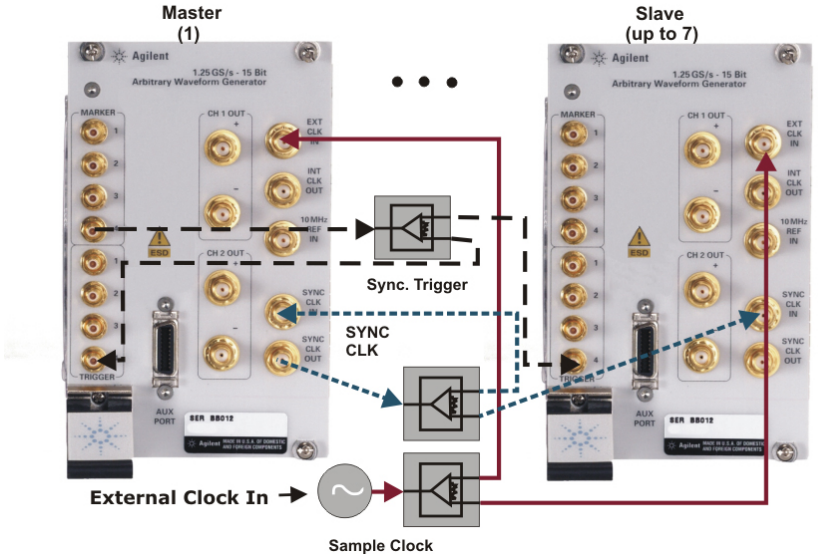
The trigger cables should all be the same length. The trigger inputs are high impedance and several inputs can be driven in parallel without matched passive splitters. The synchronous trigger timing can be determined in the same way as any synchronous trigger into M9330A Series AWG modules. The timing is specified relative to the SYNC clock out. This is easily observed on the slave modules, where the SYNC clock out is unconnected.

The multiple M9330A Series AWG modules are configured to have an internal start trigger to begin play. A software start marker event is used to initiate the synchronized play. Marker 4 and Trigger 4 are used for this purpose.

Synchronization Using an External Clock

In synchronizing multiple modules using an external clock, one unit is designated as the Master and the other units are designated as Slave units. The external clock is split with low skew and distributed to all units. The Master unit sources the following signals: SYNC clock, and the Sync Marker. These signals are all split and fed to each of the synchronized modules (the Master as well as the Slaves).

Figure 25 Master/Slave Synchronization Using an External Clock



The external Sample clock can be in the range of 625 MHz to 1.25 GHz. The Sample clock provides the final retiming of the analog output from each of the M9330A Series AWG modules. Any skew in the Sample clock cable delays between the multiple modules will result in the same skew in the analog outputs. Typically, the sample clock signal is split with a matched passive splitter and the cable lengths are matched to better than 5 mm. The resulting skew is small and repeatable. The skew can be measured and calibrated, along with any phase shifts in cables on the ARB outputs, by adding fixed delay offsets to the waveforms of the M9330A Series AWG modules. The SYNC clock is used internal to M9330A Series AWG modules to clock the internal data generator and to clock in the synchronous triggers. When using the internal clock, the SYNC clock has a frequency of 1/8 of the sample clock rate (156.25 MHz). When synchronizing multiple units, the SYNC clock output must be enabled in software (in the Master) and the external SYNC clock input selected in all the modules. The SYNC clock signal is split passively and distributed with low skew. The SYNC clock output level and the input sensitivity support up to a 1 to 8 split (fan-out) using matched 50 Ohm splitters (6 dB loss per 1 to 2 splitter). There is a specific SYNC cable length that is required as a

function of the Sample clock frequency. Several different lengths can be used, provided they are integer multiples of one half of a SYNC clock period.

Multiple Module Synchronous Trigger Timing

Triggers are registered into M9330A Series AWG modules using the SYNC clock. The SYNC clock is nominally at the sample clock frequency divided by 8. However at lower sample rates an internal variable modulus prescaler selects other binary divide ratios: 8, 4, 2, and 1.

Table 2 SYNC Clock Frequency Ranges

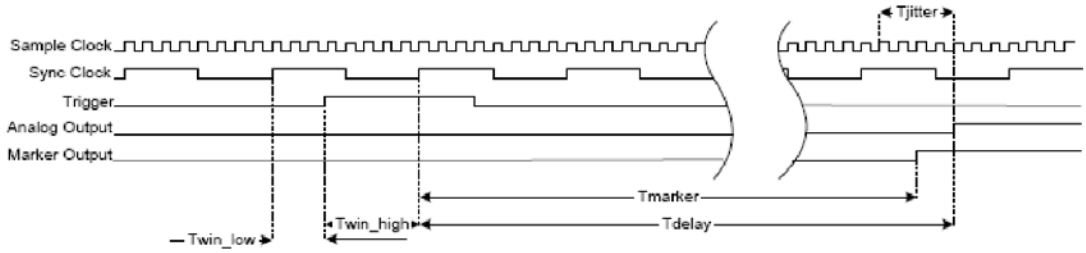
Frequency Range	SYNC Clock Prescaler Divide Ratio
625 MHz-1.25 GHz	8
312.5 MHz-625 MHz	Multi-Module Synchronization
156.25 MHz-312.5 MHz	Not Supported
100 MHz-156.25 MHz	

Multiple synchronization of M9330A Series AWG modules is only supported in the 625 MHz – 1.25 GHz frequency range. The input clock frequency ranges and prescaler divide ratios are as specified in Table 2. It is necessary to insure that the correct timing relationships are achieved to guarantee consistent synchronous trigger operation. The trigger input must occur within a valid window with respect to the SYNC clock. The window is specified by two times: Twin_low -- the minimum trigger delay after the prior SYNC clock edge; and Twin_high -- the minimum trigger setup before the next SYNC clock edge. These are specified for the trigger Input relative to the SYNC clock Output. The trigger must be a minimum of two SYNC clock cycles long. The trigger timing is specified relative to the rising edge of the SYNC clock by default, as shown in Figure 26, "Multiple Module Synchronous Trigger Timing Diagram," on page 54. To guarantee proper synchronous trigger operation with arbitrary length cables, it is possible to configure the trigger inputs to register the trigger event with respect to the falling edge of the SYNC clock, under software control. In this way there is always a setting for the trigger input timing which will operate reliably for any chosen cable. The typical specifications for the trigger window using the internal clock at 1.25 GS/s is (these values will vary at other clock frequencies):

Twin_high > 3.4 ns

Twin_low > -2.8 ns (the trigger can occur slightly before the prior SYNC clock edge)

Figure 26 Multiple Module Synchronous Trigger Timing Diagram



Cable Length and Skew

The cabling requirements are as follows:

Sample Clock

Skew less than 10 mm between modules. The absolute SYNC cable length is given by the following formula as a function of the Sample clock frequency:

Sample Clock Skew Formula 1 **(1)**

$$Length = [(n \times 686 \times (1250MHz)/f) - 394]$$

Expressed in millimeters, where *n* is an arbitrary integer and *f* is the sample clock frequency in MHz.

It should be noted that n is the number of 1/2 SYNC clock cycles of total delay between the modules.

This can also be expressed in terms of delay:

Sample Clock Skew Formula 2 **(2)**

$$Cabledelay = \lceil (n \times 3.29 \times (1250MHz)/f) - 1.89 \rceil$$

Expressed in nanoseconds, where n is an arbitrary integer and f is the sample clock frequency in MHz.

For the external Sample clock the formulas apply over the frequency range of 625 MHz to 1.25 GHz

Marker and Trigger Cables

The Marker Out to Trigger In cable should be less than 305 mm (12 in). With the 1.25 GHz internal clock, the trigger is falling edge triggered.

4
Option 300 Dynamic Sequencing

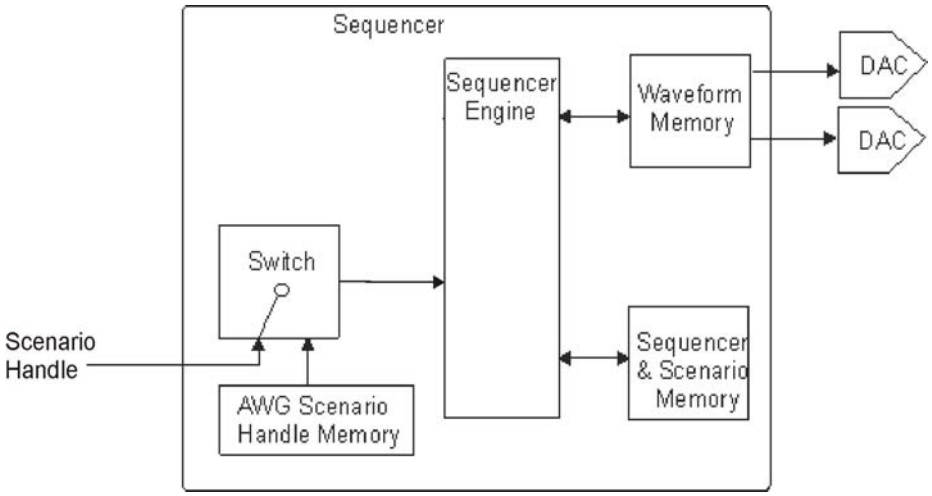
The dynamic sequencing option enables you to access up to eight thousand previously stored scenarios through a 16-bit interface. This functionality gives you the ability to build custom signal scenarios to simulate dynamically changing environments.

Dynamic Sequencing

Dynamic sequencing is a mode where M9330A Series AWG modules scenario handle memory is bypassed and scenarios are selected from an external source. You must first load the data into memory of M9330A Series AWG modules, then, in real-time, provide the scenario handles through the AUX PORT input connector.

The dynamic sequencing option is only available through the programmatic interfaces since it operates in the advanced sequencing mode that is not available through the Control Utility.
Refer to "Advanced Sequencing" described on page 38.

Figure 27 Dynamic Sequencing Block Diagram



AUX PORT Connector

- Description: Receptacle, Mini D
- Number of Contacts: 20
- Manufacturer: 3M
- Part Number: 10220-0210EC

Figure 28 AUX PORT Pin Outs

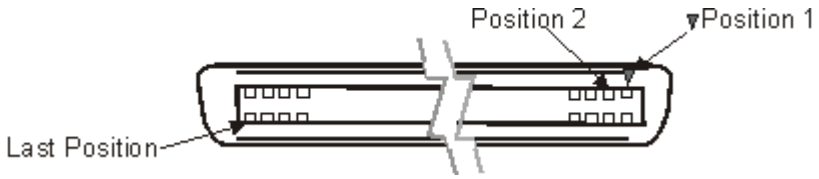


Table 3 Pin Assignment

Pin No.	Signal Assignment
1	Trigger
2	Ground
3	Data Valid
4	CH 1/CH 2 (Reserved, set low)
5	D0
6	D1
7	Ground
8	D2
9	D3
10	D4
11	D5
12	D6
13	D7
14	Ground
15	D8
16	D9
17	D10
18	D11
19	Ground
20	D12

Signal Levels

All pins are configured as 2.5 V, LVCMOS inputs. The logic levels must be within the following ranges:

- Low: –0.2 to +0.5 V
- High: +2.0 to +2.8 V

Signal Descriptions

Data Input

The input data represents a handle to the next scenario to be played by M9330A Series AWG modules. Only the first 8,192 scenarios are available. The scenario handle must be divided by 2 before being written to the AUX port. For example, to play the scenario with a handle of 72, write the value 36 to the AUX port. All scenario handles are even numbers.

Data Valid

When Data Valid is asserted high, it indicates that the data present on the Data pins is valid and can be latched into the channel 1 and channel 2 next sequence register.

Trigger

Trigger input can be configured to be either rising-edge or falling-edge, with a programmable delay. Refer to "Triggers" described on page 45.

The latency between trigger assertion and sequence playback is the same as that for the front panel trigger inputs, a resolution of one SYNC clock.

5
Option 330 Direct Digital Synthesis

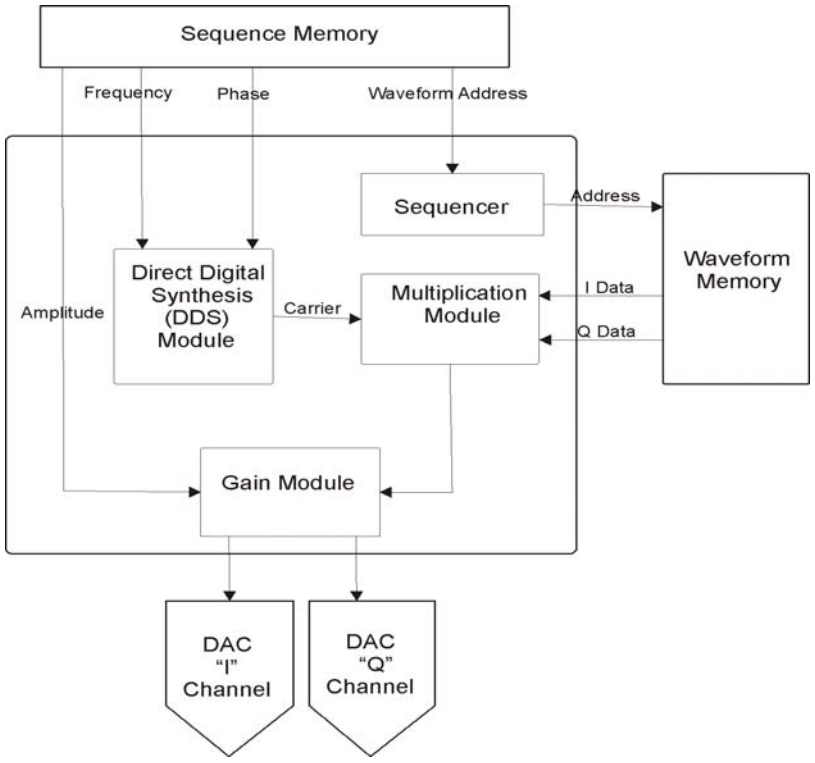
The direct digital synthesis (DDS) architecture in M9330A Series AWG modules enables you to create basic waveforms in the AWG memory and then modify the behavior of the waveforms with profiles for amplitude, phase, and frequency modulations.

Direct Digital Synthesis (DDS)

The direct digital synthesis (DDS) application can be managed through the Control Utility graphical user interface (GUI) or one of the supported programmatic interfaces. Accessing DDS through the GUI is the easiest way to view the functionality as many details are handled by the software in the background.

As an introduction, we will step through using DDS with the Control Utility. Figure 29 displays a high level DDS block diagram.

Figure 29 DDS Block Diagram



Direct Digital Synthesis Using the Control Utility

A spectrum analyzer is required to display the waveform.

Configuring the Equipment

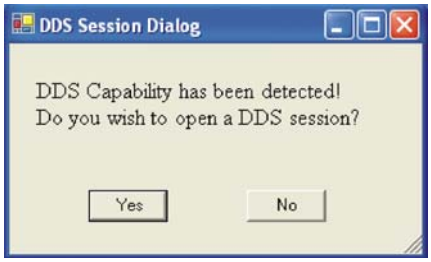
1 Connect a 10 MHz reference from the spectrum analyzer to a front panel connector on one of the M9330A Series AWG modules. If you are using a PXI chassis, use the backplane 10 MHz reference.

2 Connect the channel 1 positive (+) output to the spectrum analyzer RF input connector.

Selecting the DDS Option

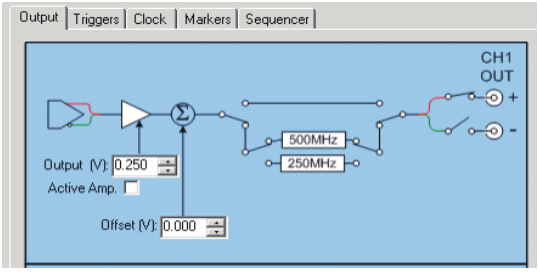
Open the **Control Utility** by double-clicking the icon on the desktop.

In the **DDS Session Dialog** box, select **Yes**.



Configuring the Signal Conditioning Path

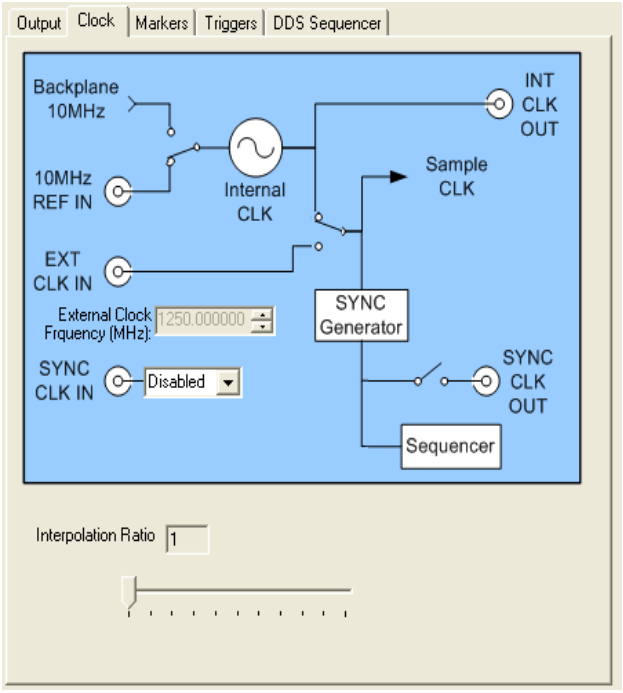
1 Select the **Output** tab and connect a single-ended signal conditioning path to CH1 OUT (+) (click on the node that you want to connect).



The connection will automatically enable differential mode. Click on the negative (–) node to open this path and enable single-ended mode.

Configuring the Clock

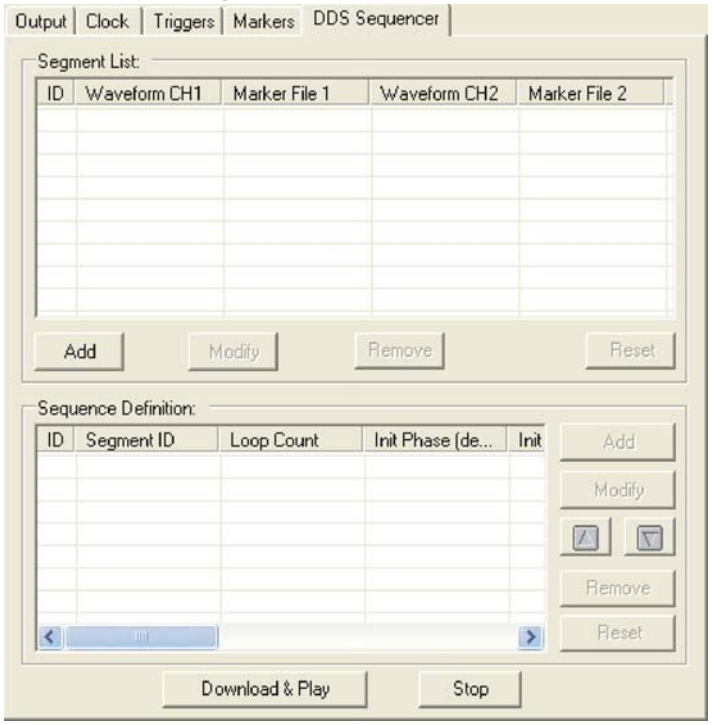
Select the **Clock** tab and configure the 10 MHz REF IN. For this example, we utilized the 10 MHz reference from the spectrum analyzer in step 1. If you are using a PXI chassis, leave the clock set to the default Backplane 10 MHz.



Use the default setting for the **Interpolation Ratio**.

Configuring the Sequencer

Select the **DDS Sequencer** tab.



From the **Segment List** select **Add**. This brings up a **Segment Information** window.

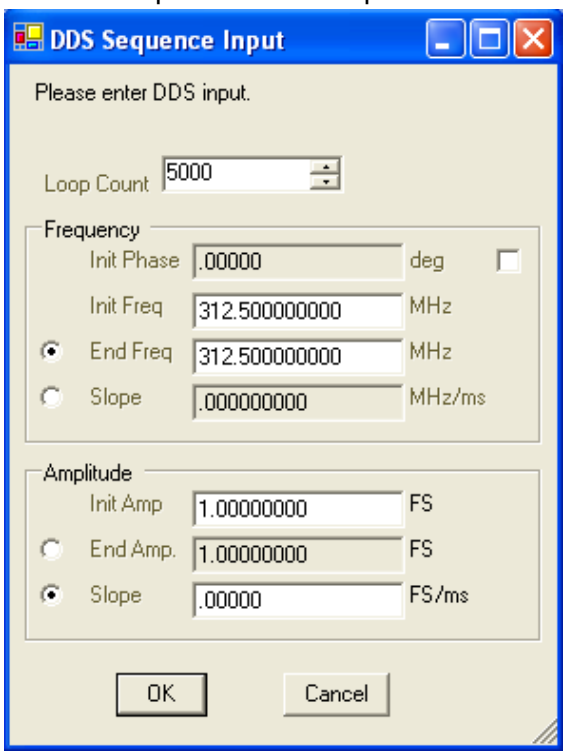
Browse and select the **DDS_All_Ones** waveform from the **Demo Waveform DDS** folder included on the M9330A Series CD then click **OK**.

For dual channel sequencing, add waveforms of the same length to both channel 1 and channel 2. The software does not support independent channel sequencing.

In the **Segment List**, select the **DDS_All_Ones** waveform.

In the **Sequence Definition** area, select **Add**. This brings up the **DDS Sequence Input** window.

Enter **5000** repetitions and accept all default settings. Click **OK**.

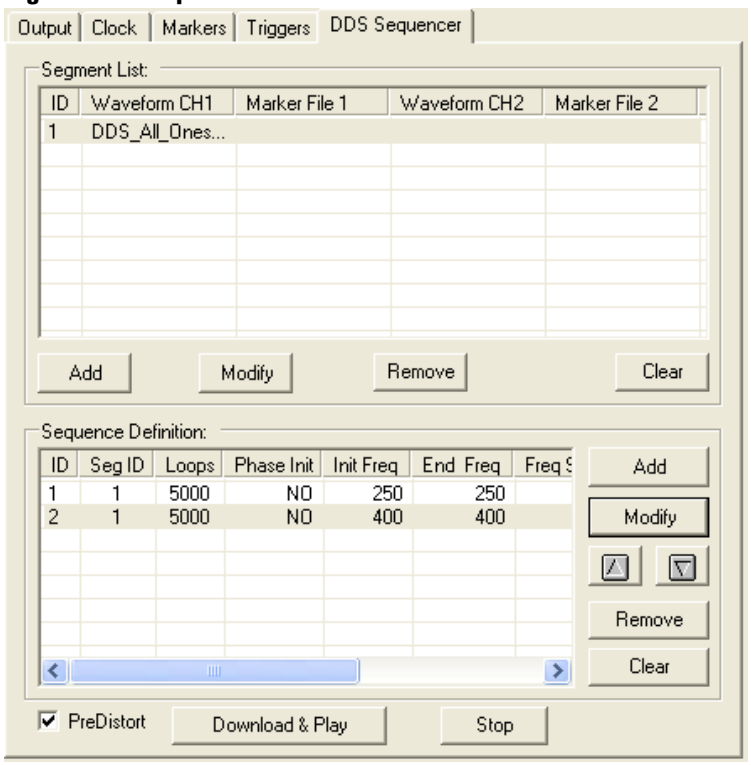


The values entered in the DDS Sequence Input window are recorded in the sequence definition area of the Sequencer tab. This enables you to review the values after the DDS Sequence Input window is closed.

Repeat steps **4**, **5**, and **6** using a 400 MHz Init Freq Value.

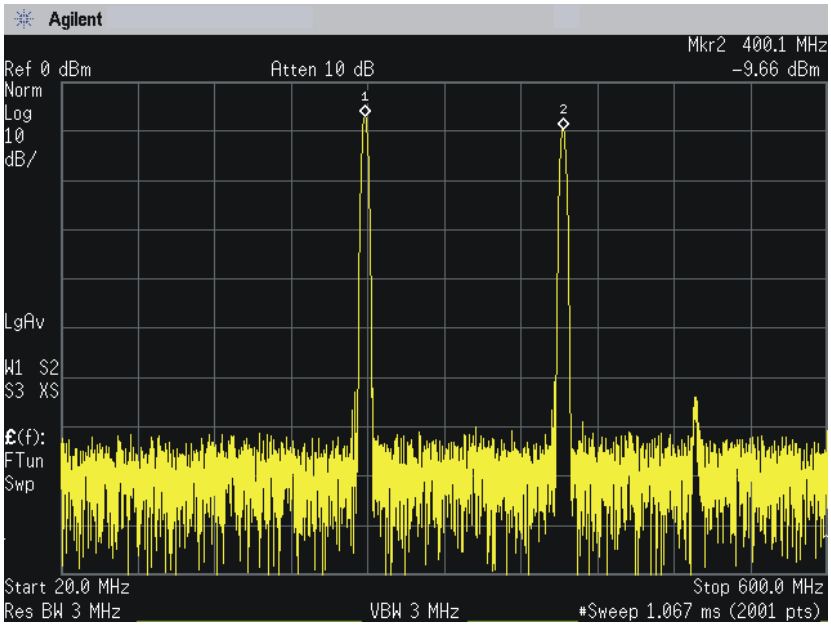
The sequencer tab should look like Figure 30.

Figure 30 Sequencer Tab



Click **Download & Play**. The spectrum of the sequence should be similar to the one shown in Figure 31.

Figure 31 Playback of a Sequence

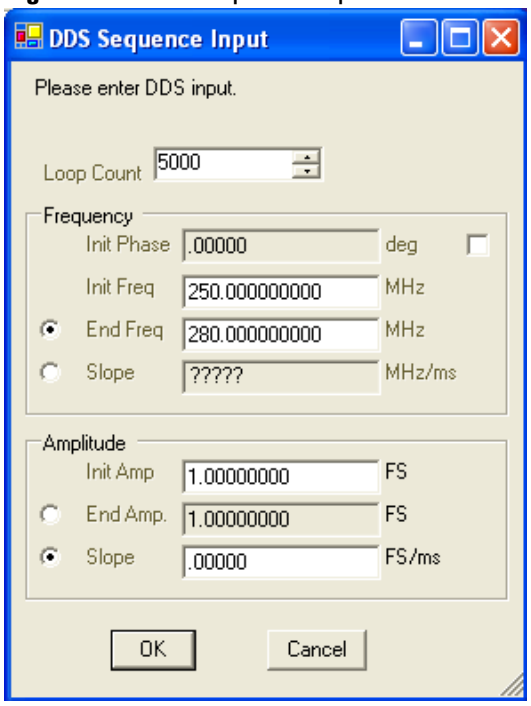


The 250 MHz carrier (marker 1) and the 400 MHz carrier (marker 2) are combined with a waveform composed of all ones. This illustrates how the DDS engine produces sine waves when a constant frequency is specified.

Out of Range Input Values

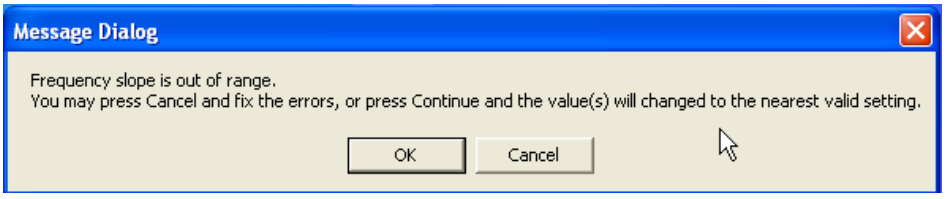
Some values may cause an 'out of range' condition. Refer to Figure 32.

Figure 32 DDS Sequence Input Window



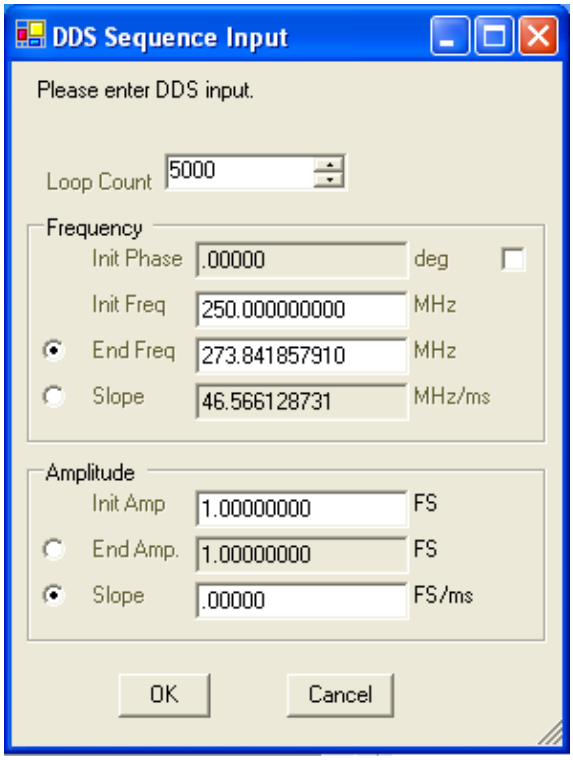
Notice the question marks in the **Frequency Slope** box. This is occurs when the combination of the loop count, the initial frequency, and the end frequency cannot be calculated correctly. If you select **OK**, a message window comes up.

Figure 33 Message Dialog



For this example, selecting **OK** resulted in values shown in Figure 34.

Figure 34 Calculated Valid Settings



The end frequency value was adjusted to enable the slope count.

This type of 'out of range' condition may also occur with amplitude settings.

Theory of Operation

The Direct Digital Synthesis, Option 330, is a powerful tool for those customers who are using M9330A Series AWG modules to synthesize waveforms best expressed in the frequency domain. Traditionally, waveforms are expressed in the time domain, sampled, and then stored in waveform memory for eventual playback. This approach is completely generic and applicable to any describable waveform. However, many waveforms can be described as a combination of information content and simple sinusoids. For these waveforms, most of the available waveform memory gets used up storing the sinusoids, leaving little space for the information content.This is an inefficient utilization of waveform memory.

For example, in communications, the waveform can be described as a carrier (sinusoid) modulated with data (information content). Because M9330A Series AWG modules have such high dynamic range, the modulated carrier can be generated with very good equivalent error vector magnitude (EVM) performance. But, because the carrier has to be stored in waveform memory along with the modulation, limited playback time can be achieved. Another important example of a frequency domain waveform is wideband radar chirps. Again, the waveform consists of a combination of a sinusoid and a frequency chirp profile, both of which must be traditionally stored in waveform memory, resulting in limited playback time.

To address this issue, Option 330 allows the AWG to generate the sinusoidal portion of the waveform real time, and then modulate the sinusoid with the information content stored in waveform memory (see Figure 35). This is done by adding a direct digital synthesizer (DDS) to the main FPGA in the AWG. The DDS implemented has a frequency resolution of 1.1369 mHz, and can synthesize sinusoids from DC to 400 MHz (using the 1.25 GHz internal clock). The DDS can be linearly ramped in frequency, with a frequency ramp rate resolution of 1.3552 Hz/s, and a maximum frequency ramp rate of 46.566 GHz/s. The frequency ramp rate can be positive or negative.

The initial phase of the DDS can be set to a known value, with a phase resolution of 21.458 μdegrees. Alternatively, through the use of the initial phase control field, the DDS can be operated in phase continuous mode; for example, the initial phase not initialized. This is useful for applications requiring phase continuous frequency hopping. The DDS generates both sine and cosine outputs for use in the complex modulator. Refer to Figure 35.

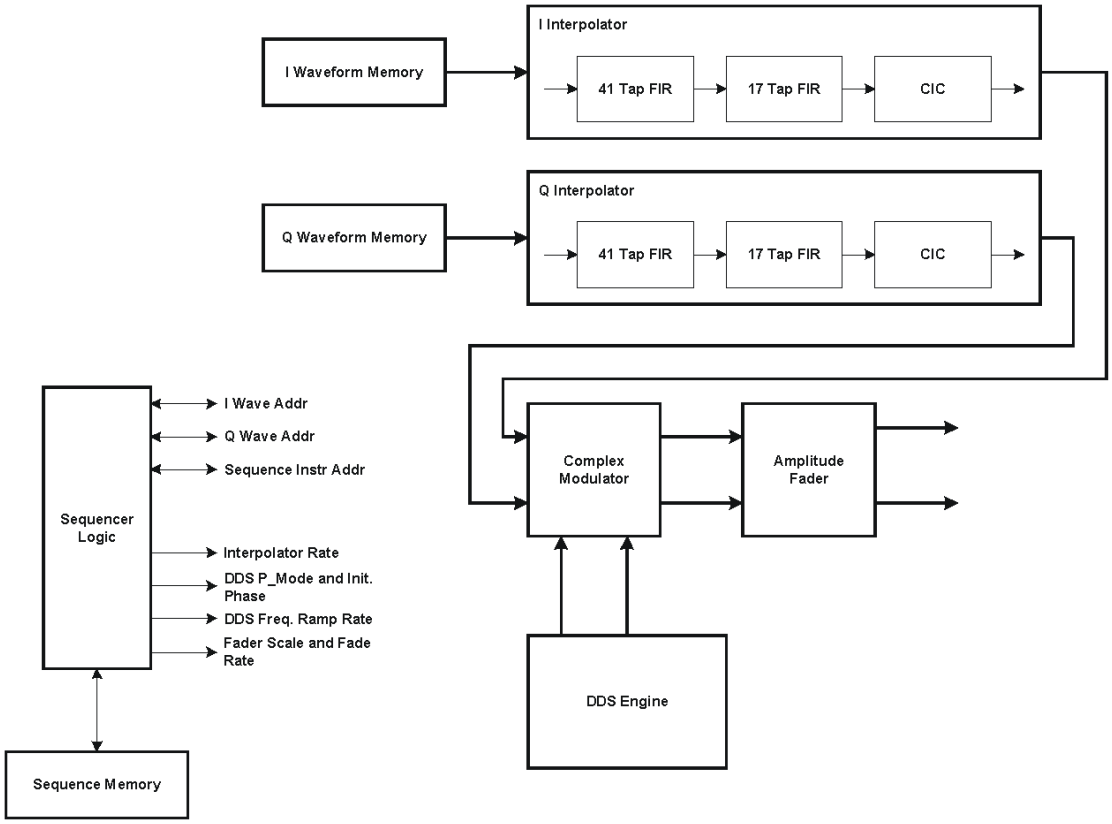
To allow waveform memory to be played back at a rate slower than the AWG sample rate, interpolation filters have been added to the main FPGA, for both channels. The interpolation filters can be configured to interpolate by integer powers of two: 2, 4, 8. . ., up to a value of 1024. Image rejection for the filters is better than 65 dBc for all interpolation rates, and flatness

is compensated for automatically in software. By setting the interpolation filters to 1024, the waveform memory can be played back at a rate over a thousand times slower than the AWG sample rate. The interpolation filters can also be bypassed.

The sine and cosine outputs of the DDS, and the interpolated outputs of waveform memory are sent to a complex (or I/Q) modulator for upconversion. If the channel 1 interpolated memory is represented symbolically by "I", and the channel 2 interpolated memory by "Q", then the channel 1 analog output can be expressed as I*cos(wt) – Q*sin(wt). Channel 2 can be expressed as I*sin(wt) + Q*cos(wt). Each analog output represents a carrier (DDS output) I/Q modulated by data (channel 1 and 2 interpolated waveform memory). Alternatively, if both analog channels are subsequently used to drive an external I/Q modulator, they are configured to provide for upper sideband SSB conversion at the output of the external modulator.

Both internally modulated outputs can be linearly faded in amplitude within the AWG. The linear fade function occurs after the complex modulator. Amplitude fade rate resolution is set to 1.819% full scale per second, with a maximum fade rate of 62.5% full scale per nanosecond. Fade rates can be positive or negative.

Figure 35 DDS



6
Application Note, Predistortion

This application note explains the predistortion feature available in M9330A Series AWG modules.

The following topics are included in this chapter:

Overview

Predistortion of a waveform is performed during the waveform creation (download) process. (Download occurs when “Play” or “Arm” is pressed in the Control Utility, and when agt awg storewaveform is called in Matlab.) The resulting pre-distorted waveform is stored in memory inside M9330A Series AWG modules. Predistortion is intended to remove the effects of non-ideal components in the analog circuitry (or analog path) of M9330A Series AWG modules.

For example, Figure 36 shows the frequency response of hypothetical M9330A Series AWG modules. This is for purposes of illustration only and is not meant to represent the actual response of M9330A Series AWG modules. Also shown is the amplitude spectrum of a multi-tone signal being played on M9330A Series AWG modules. Although the tones were created with equal amplitudes in the digital waveform, they are generated at varying amplitudes due to the non-ideal frequency response of M9330A Series AWG modules.

An important observation here is that while the lowest frequency tones are generated at the expected amplitude (This is specified by the IVI-C “arb gain” attribute), the higher frequency tones are not at the expected amplitude. This is an unavoidable effect created by non-ideal components in the analog path.

We’ll assume the multi-tone waveform has been scaled to be as large as possible, to maximize dynamic range. We now wish to pre-distort the waveform so the multi-tone signal is “flat” over frequency. Since, we’ve already produced the maximum possible amplitude at the high frequencies, the only option is to reduce the amplitude of the lower frequency tones to equal the lowest amplitude (high frequency) tone. Figure 37 shows the result. The amplitude of low frequency tones has

suffered in order to achieve a flat signal.

Figure 36 AWG Gain versus Frequency

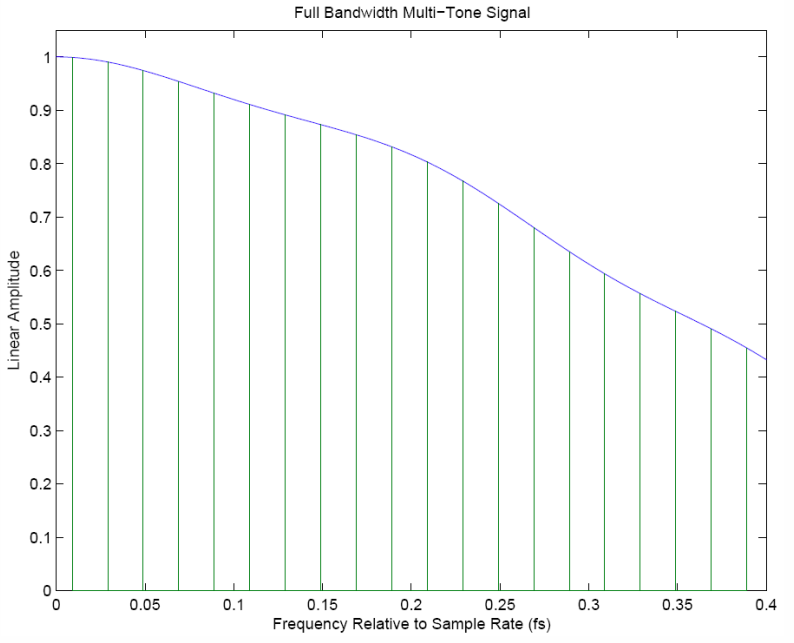
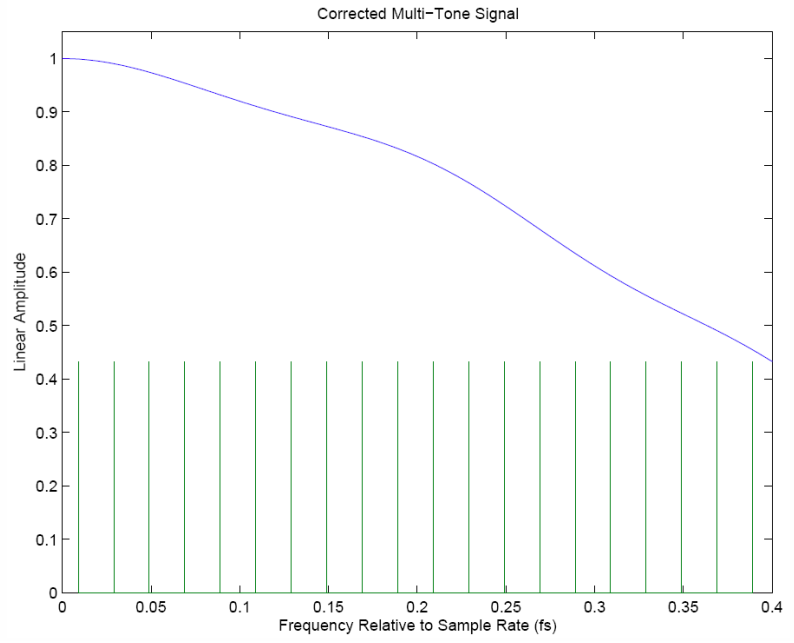


Figure 37 Corrected Signal



Information about the current analog path configuration is used during predistortion to ascertain the exact corrections which are to be applied. There are several settings (or in IVI-C jargon, attributes) which can alter the analog path between DAC output and the front panel connectors. Below is a list of the settings which determine the analog path:

- Output configuration (single-ended, differential or amplified)
- Output filter enabled
- Output bandwidth

Analog path behavior is of course, a function of (absolute) frequency. Frequency components in a waveform are however also dependent on the sample rate at which that waveform is played. Therefore, waveform predistortion must also take into account the chosen sample rate.

Predistortion *- is invalid for new settings!*

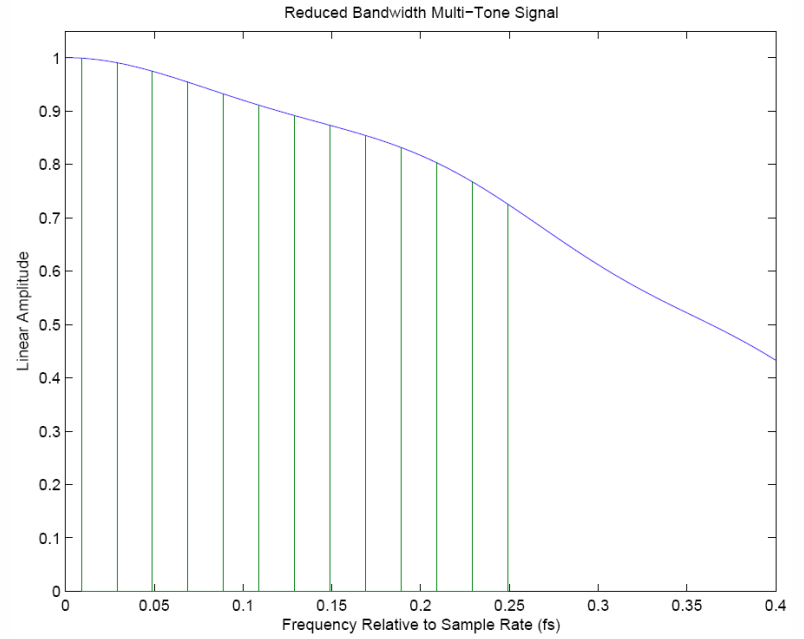
Once a waveform has been pre-distorted to correct for a specific analog path and sample rate, routing it through a different analog path or playing it at a different sample rate will not give the desired results – *the predistortion is invalid for the new settings*.

The analog path attributes and sample rate of M9330A Series AWG modules must be set prior to downloading any waveforms that are to be pre-distorted.

Waveform Scaling

As noted above, pre-distorted waveforms are smaller and do not provide the amplitude expected in accordance with the ARB GAIN attribute setting. We assume during predistortion that the waveform contains frequency components up to 40% of the sample rate which must be corrected. Waveforms which do not use this entire frequency range will be reduced in amplitude more than necessary. Figure 38 shows such a waveform. Using this assumption, all components of this signal would be reduced in amplitude below a level of about 0.45, which is clearly not necessary to achieve a flat response in this case.

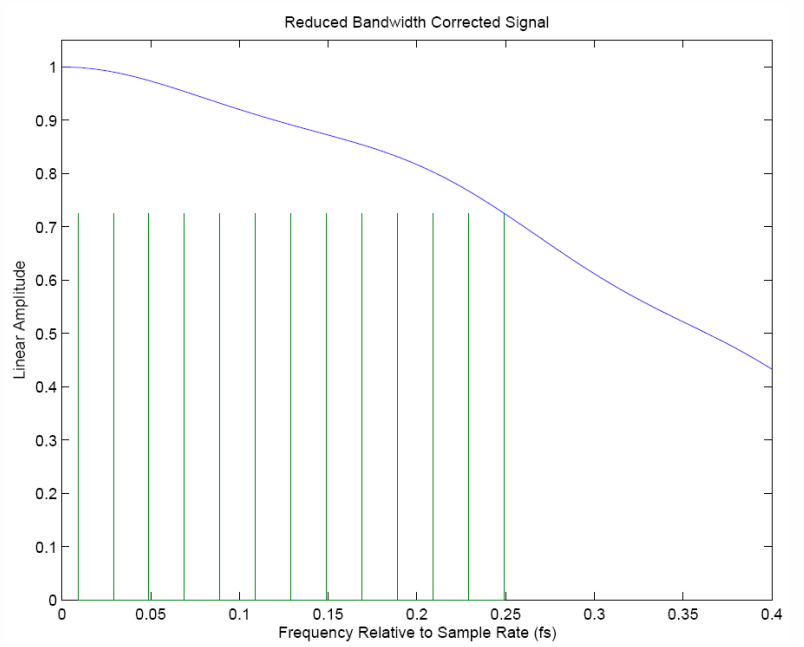
Figure 38 Band-Limited Signal



The IVI-C driver provides the ability to specify the actual bandwidth of the waveform being pre-distorted. By using this feature, unnecessary reductions in signal amplitude can be avoided for waveforms with limited bandwidths. Figure 39 shows the waveform of Figure 38 after being corrected using this feature.

The individual tone amplitudes are now above 0.7 instead of 0.45 as would have occurred otherwise.

Figure 39 Band-Limited Signal



There is an additional factor which can also reduce the amplitude of pre-distorted waveforms. Predistortion is accomplished by passing the waveform through a FIR digital filter. During this operation, care must be taken that waveform clipping does not occur. This task is made more difficult by the fact that the software has no prior knowledge of the waveform to be filtered. To guarantee that clipping does not occur we must assume a worst-case scenario when scaling the result. The effect is that most waveforms will be reduced in amplitude more than necessary by the predistortion process. The alternative of providing automatic optimal scaling for each waveform would result in un-predictable and mismatched amplitudes between different waveforms. Again, the IVI-C driver does allow more control over this aspect of scaling.

Examples of Advanced Waveform Scaling Controls Usage

Presented here are several examples which demonstrate the increase in output level that is possible by making use of advanced waveform scaling controls.

Scaling behavior is examined with four different signals,

- A broadband 500 MHz wide chirp. This utilizes the full bandwidth of the AWG.
- A QPSK signal which occupies bandwidth up to about 450 MHz.
- A QPSK signal with 225 MHz of bandwidth.
- A multi-tone signal which extends up to 65 MHz.

Four scaling scenarios are examined for each signal,

- No predistortion at all. Although this results in the maximum output level, the signal is of lower quality due to amplitude and phase variations at higher frequencies.
- Standard predistortion with scaling guaranteed safe for any signal.
- Predistortion scaling bandwidth adjusted to match the signal's frequency content.
- Scaling bandwidth specified, plus optimum scaling computed for each waveform.

Table 4 shows the measured peak-to-peak output voltage for each waveform with each type of scaling. Voltages are in milli-volts, peak-to-peak; divide them by two for peak voltages as would be specified via the ARB GAIN IVI-C attribute. As shown by the data, it is often possible to achieve an output voltage level at least twice (for example, +6 dB) the safe value provided by default predistortion scaling. In some cases the improvement can be as much a factor of four (+12 dB).

Table 4 Output Voltages for Various Scaling Scenarios

Signal	Predistortion Off	Normal	BW Spec'd	Optimum
500 MHz Chirp	385	105	105	200
450 MHz QPSK	335	93	153	287
225 MHz QPSK	287	68	210	282
65 MHz Multi-tone	443	97	342	443

Reconstruction Filters

The reconstruction filters used in the AWG are designed to have 3 dB of attenuation at the advertised cutoff frequency. When using the internal clock, the 500 MHz filter is most often used. At 500 MHz, the internal DACs (digital-to-analog converters) have about 6 dB of loss at 500 MHz, compared to low frequencies or DC. The internal filter adds another 3 dB to this, resulting in a total of about 9 dB at 500 MHz.

In cases where this amount of high frequency roll-off is too much, the internal filter may be bypassed and higher performance external filters can be used. One such filter is from RLC Electronics, part number F-30-600-R. This filter will result in approximately 6dB of total roll-off at 500 MHz, and the lowest alias component (at 750 MHz) will be attenuated by at least 55 dB. In this case, predistortion should be disabled in the AWG; frequency and phase corrections will need to be performed external to the AWG. It is possible to leave predistortion enabled, which will correct for amplitude and phase variations at the AWG's output port, but variations due to the external filter will need to be corrected elsewhere.

DDS

Due to the unpredictable frequency content of signals produced in DDS mode, predistortion is not performed for DDS waveforms.

Operational Details

To aid in dealing with these restrictions, IVI-C attributes and functions have been provided.

AGM933X ATTR PREDISTORTION ENABLED

By setting this attribute to zero or false, predistortion of waveforms is completely disabled. Lower frequency waveform components will have the expected amplitude, but high-frequency components will not, due to gain roll-off in the analog path. For the same reason, the phase of high-frequency components will not be as expected.

AGM933X ATTR PREDISTORTION VALID

This attribute is a read-only boolean. When queried, it indicates whether the currently configured waveform or sequence is properly pre-distorted for the cur- rent AWG state (analog path and sample rate). If the AWG is in sequence, or advanced sequence mode, it will check all waveforms referenced by the currently configured sequence and return the logical "and" of validity for each waveform.

A waveform is configured to play (either directly, or as part of a sequence) by setting the waveform or sequence handle attribute. If the waveform or sequence thus configured contains predistortion that is invalid for the current AWG state, the following warning code is returned from the set-attribute routine.

AGN6030A WARN PREDISTORTION INVALID

No warning is returned however when a change to the analog path or sample rate causes existing predistortion to become invalid. A separate check must be made in this case.

AGM933X ATTR PREDISTORTION SCALE

When queried, this floating point attribute will return the scale factor used in the predistortion of the most-recently created waveform. Since this is a multi-channel attribute, it returns the scale value for the most recently created waveform on the specified channel. The actual output gain for this waveform is the current AWG gain multiplied by this scale factor. If the waveform was not pre-distorted, a value of "1.0" is returned.

There is also an IVI-C function which will return the predistortion scaling associated with a particular waveform handle:

AGN6030A GetPredistortionScale

AGM933X ATTR PREDISTORTION BANDWIDTH

This floating point attribute is normally set to zero, which causes the IVI-C driver to assume waveforms use the full bandwidth available (40% of the sample rate). By setting this to an absolute value (in Hertz), predistortion software will only correct waveforms at and below the indicated frequency. This will result in more signal amplitude when a waveform has smaller frequency content.

AGM933X ATTR PREDISTORTION AUTOSCALE

To gain more control over waveform predistortion, set this boolean attribute to zero or false. When auto-scaling is disabled, the user must explicitly specify the scaling value to use when waveforms are pre-distorted.

AGM933X ATTR PREDISTORTION SCALE

When auto-scaling is disabled, the user must set the desired scaling value using this attribute. Since this is a multi-channel attribute, a value must be set for each channel (although the same value is often used for both channels).

Using an overly large scaling value will cause clipping while using an unnecessarily small value wastes output amplitude and dynamic range. Therefore, the IVI-C driver provides the ability to determine the optimum scaling for any given waveform.

AgM933x_Predistortion GetOptimumScaleRaw

This function will analyze a [raw] waveform and return the largest possible scaling value which will avoid clipping. If a single waveform is being played, this value can be used directly to set the predistortion scale.

When a pair of waveforms are played on the AWG's two channels simultaneously, it is often desirable to maintain a known relationship between the amplitude of each channel. In this case the optimum scale should be obtained for each waveform (being careful to specify the proper channel for each), and the smaller of the two scale values used to pre-distort both waveforms.

When a sequence is being played, it may be desirable to obtain the optimum scale value for each waveform in the sequence, then use the minimum of all scale values returned as the optimum setting.

Performance Issues

The internal process of computing the optimum scale value requires that most of the predistortion process be performed on a waveform. For large waveforms this will result in a time penalty. Unfortunately, there is no easy way to avoid this and still achieve optimal amplitude scaling.

The time penalty can be avoided if a guess is made at a larger safe scaling value. The guess should be based on analyzing several typical waveforms ahead of time to get a feel for the range of optimal scaling values. The guessed scaling value can be used for all waveforms. The return code received from AgM933x_CreateArbWaveform or other functions will contain a warning value if the scaling factor is too large, and resulted in the waveform being clipped.

7
Application Note, Waveform Download & Sequencing

This application note provides detailed information on downloading waveforms and waveform sequencing for the M9330A AWG.

The following topics are included in this chapter:

Overview

The process of downloading waveforms into these two-channel AWGs is easier to manage with an understanding of the IVI function generator (or FGEN) definition. The FGEN definition makes assumptions about an AWG's internal architecture which are slightly different than the actual architecture of the M9330A AWG. Understanding these differences are helpful when downloading waveforms and creating sequences.

The IVI FGEN standard assumes waveform memory is not channel specific. In other words, once downloaded, a waveform may be played on any AWG channel. As a consequence, the decision of which channel a given waveform will be played on is made after the waveform is downloaded.

Each channel in the M9330A AWG has separate, dedicated memory. Separate physical memory exists for channel 1 and channel 2. Therefore, as part of the download process, the playback channel must be chosen. Obviously, once downloaded, a waveform can only be played on the channel which was chosen during the download process.

This application note explains specific behaviors which are necessary for the M9330A AWG to conform to the FGEN standard. Understanding these behaviors makes programming the AWG a more straightforward process. An example of how not to download waveforms will be followed by a description of the proper method.

Downloading Waveforms

When a waveform is downloaded into the AWG, a channel assignment must be made. However, the FGEN standard does not permit this. To solve this problem, the IVI-C driver contains a memory manager which automatically makes a channel assignment. Although it is possible to determine which channel a downloaded waveform was assigned to, it is not necessary and should be avoided when necessary.

Memory Manager

The memory manager's operation is not complicated. When a waveform is downloaded, memory of both channels is searched for the first available segment large enough to hold the waveform. The segment with the lowest address is chosen to hold the waveform. If both available segments have the same address, the segment associated with channel 1 is chosen.

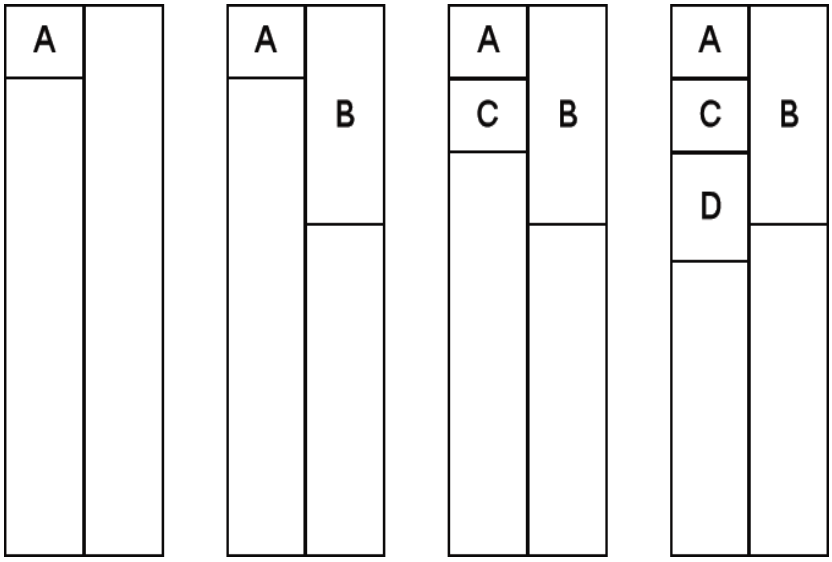
Several examples are shown below (see Figure 40) to demonstrate the memory manager's operation. The first example shows the process of downloading several waveforms of varying lengths.

For the first waveform (A), one segment is available in each channel at the start of memory. Since the start addresses are identical, the waveform is stored in channel 1's memory.

The second waveform can be placed either after A in channel 1 memory or at the start of channel 2 memory. Since the channel 2 address is lower, channel 2 memory is chosen.

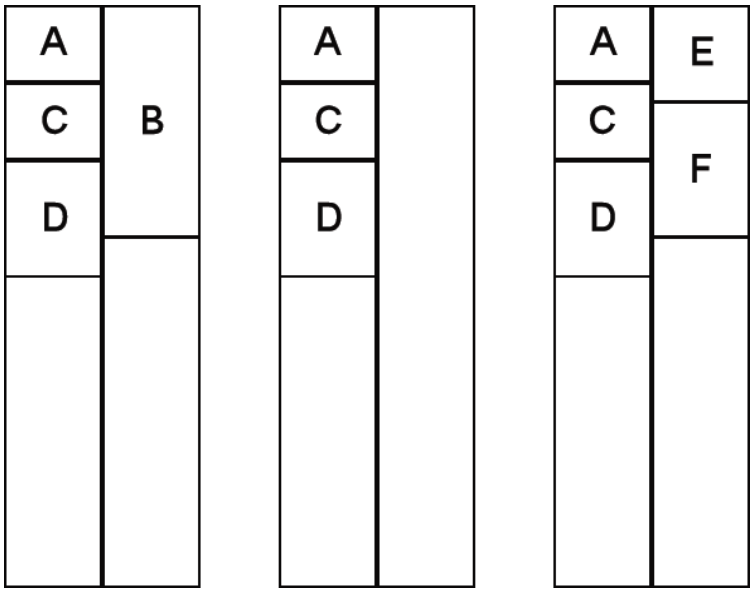
Waveforms C and D are stored in channel 1's memory since their starting addresses are lower than would be achieved in channel 2's memory (following waveform B). It should also be apparent that the next waveform will be stored in channel 2's memory.

Figure 40 Downloading waveforms of differing lengths



The memory manager's behavior is unchanged after waveforms are deleted. Figure 41 shows an example of this. After deleting waveform B, the next two waveforms are stored in channel 2's memory since they have lower starting addresses there.

Figure 41 Waveform deletion



Sometimes, deleting a waveform (such as waveform C in figure 2) results in a "hole". The memory manager will subsequently use this hole if the waveform to be downloaded is the same size or smaller than the hole, and the start address is lower than the first available segment in the other channel's memory.

Finally, it is possible to wind up memory fragmentation if a lot of waveforms are created; some are deleted, and so on. This is not a performance problem, but may result in a "memory full" error when there is enough free memory, but it is not in a single contiguous chunk. When this occurs, waveform memory must be cleared and re-loaded. There is no function available to de-fragment memory.

Waveform Playback

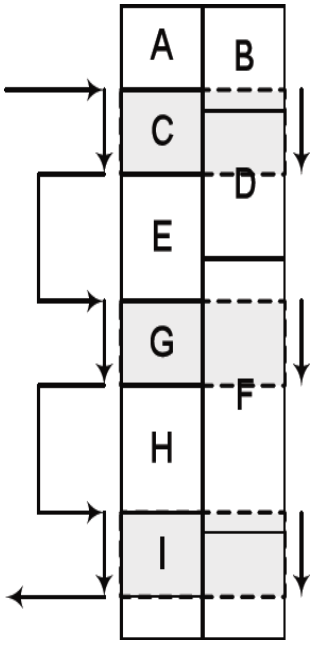
When defining a sequence, the waveform handles supplied should all be associated with the same channel. For example, in figure 3, (C, G, I) is a valid sequence, but (A, B, H) is invalid.

When individual waveforms are played, or sequenced, identical addresses are always accessed simultaneously in both channels' memory. This does not always result in meaningful output from both channels.

Figure 42 shows an example of this. Here, several waveforms of unequal length have been downloaded, and a sequence has been defined using waveform handles from channel 1. Although the desired waveforms from channel 1's memory (C, G, and I) are played correctly, the output from channel 2 contains only various un-aligned chunks of different waveforms. This is almost certainly not what was intended.

As this example demonstrates, mixing unequal length waveforms between channel 1 and 2 results in confusing behavior which is hard to predict and should be avoided.

Figure 42 Sequencing unmatched waveforms



Downloading Waveforms with Predictable Results

If a very simple rule is followed when downloading waveforms, none of the confusion described above will ever occur, and predictable results are always obtained.

Always download and delete waveforms in 2-channel, equal-length pairs.

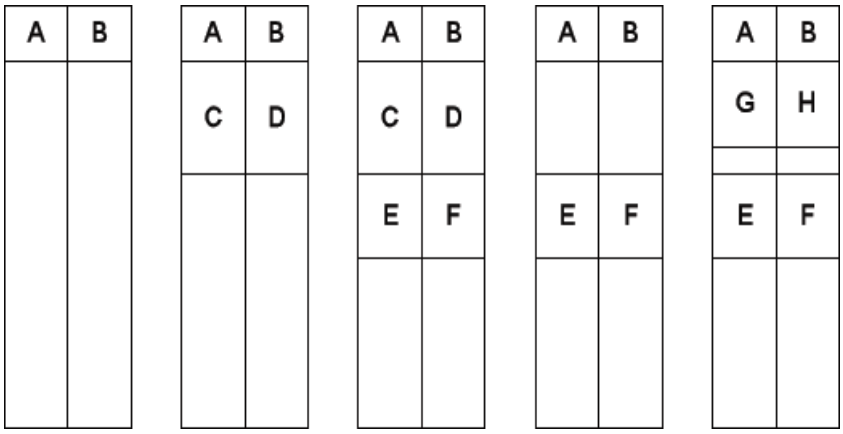
The first waveform of the pair will always be assigned to channel 1, the second to channel 2. In cases where no waveform is required for one of the two channels, an identical-length waveform of zeros must be downloaded.

Following this procedure has the added benefit that the software created should also work with other compliant IVI-FGEN drivers.

The channel assigned to a waveform can be verified by examining the waveform handle; channel 1 handles are even and channel 2 handles are odd. This should be avoided if possible though, because it is an extension to the FGEN standard and results in less portable software. The IVI-C driver will automatically check this when a waveform is assigned to a channel. If the wrong channel is specified, an error will be reported.

Figure 43 shows the outcome when the above rule is followed. In this scenario, three waveform pairs (six waveforms) are first downloaded. Next the second waveform pair is deleted. The next pair of downloaded waveforms is inserted into the hole created when waveforms (C, D) were deleted.

Figure 43 Correct waveform download



Building Sequences

As mentioned above, sequences must be built with waveform handles from one channel only. Either channel can be used, but waveform handles from both channels can not be mixed in a single sequence. Referring to figure 4, a sequence can be build from any combination of waveforms (A, G, E) or (B, H, F).

When a sequence is configured to play, one of the two channels must be specified. The IVIC driver will check that all of the waveforms included in the sequence are associated with the indicated channel; an error is reported if there is a mismatch. In figure 4, a sequence built from waveforms (A, G, E) must be assigned to channel 1, while a sequence which uses waveforms (B, H, F) must be assigned to channel 2.

8 Application Note, Advanced Sequencing

This application note further explains the advanced sequencing features available in M9330A Series AWG modules. This information is intended to supplement the User's Guide and assumes the reader is familiar with that publication. Specifically, it is assumed the reader understands basic sequencing features as defined in the IVI-Fgen specification, and further explained in this user's guide.

The following topics are included in this chapter:

Overview

Advanced sequencing enhances basic sequencing in three distinct ways:

- Scenarios are added, which provide a way to repeat and order sequences, much in the same way that sequences provide the ability to repeat and order waveforms.
- Additional flexibility is provided in triggering the playback of scenarios. In fact, all playback must be triggered – unlike the basic sequencing mode. Triggering can be varied on a waveform-by-waveform basis within each sequence.
- New marker events are available. Both sequence and scenario events can be masked on a per-waveform and per-sequence basis.

To create a scenario, at least one waveform and sequence must first be created. The simplest possible scenario contains a single sequence, and that sequence contains a single waveform. In the advanced sequencing mode, sequences are referred to as advanced sequences to distinguish them from sequences created in basic sequencing mode. References to sequences in this application note should be taken to mean advanced sequences, unless otherwise noted.

Often, users will use only a subset of advanced sequencing capabilities. For example, the extra triggering capabilities are needed, but the additional power of scenarios is not needed. In this case, the user would create waveforms and sequences as necessary, and degenerate scenarios containing a single sequence would be created to allow the sequences to be played.

For the sake of brevity, references to IVI-C attributes and enumerated values will omit the prefixes AGM933xA ATTR. For example a reference to the OUTPUT MODE attribute should be taken to mean AGM933xA ATTR OUTPUT MODE.

Please refer to the advanced sequencing flowcharts in the “Advanced Sequencing” section of the “Theory of Operation” chapter in the User’s Guide for further clarification of advanced sequencing.

Scenarios

The global behavior of scenarios is specified through IVI-C attributes.

Scenario Selection

There are two ways to specify which scenario the AWG should play. This is done with the SCENARIO_ID_SOURCE attribute and it has two possible settings:

- ID SOFTWARE: The scenario is chosen by setting the ARB SCENARIO HANDLE attribute.
- ID AUX PORT: Data presented at the front panel auxiliary port is used to choose a new scenario. This setting is only valid when the dynamic sequencing option has been installed in the AWG.

As mentioned above, when scenario ID source is set to “software”, the ARB SCENARIO HANDLE attribute is used to specify the scenario to be played.

Starting and Changing Scenarios

Simply setting the scenario handle to be played will not begin playback. If the sequencer has been aborted (with the AbortGeneration() IVI-C function), the InitiateGeneration() function must be called to arm the AWG for playback.

At this point, a start trigger must be issued to begin scenario playback. The TRIGGER SOURCE attribute is used to select a source for the start trigger. When the sequencer receives a start trigger, it will retrieve a scenario handle from the *configured source* and begin playing that scenario. (The configured source is either the last value written to the ARB SCENARIO HANDLE attribute, or the front panel auxiliary data port.)

If the sequencer is already busy playing a scenario, the transition to a new scenario is first controlled by the SCENARIO PLAY MODE attribute. When this attribute is set to PLAY SINGLE, the sequencer will finish playing the current scenario and then pause, waiting for a scenario advance trigger. Upon receiving the advance trigger, the sequencer will read the current scenario ID from the configured source (software or auxiliary port), and begin playing that scenario.

It is not necessary to update the scenario ID – the previous scenario will simply play again if an advance trigger is received in this case.

The other setting for SCENARIO PLAY MODE is PLAY CONTINUOUS. In this state, the configured scenario will play indefinitely until the sequencer is either stopped (with a stop trigger or a call to AbortGeneration), or a scenario jump trigger is received.

The scenario jump trigger is mainly intended for two purposes:

1 To transition to the next scenario when the scenario play mode is continuous. In the continuous mode, this is the only way to get to another scenario without a stop trigger or aborting generation.

2 To prematurely exit a scenario and begin playing a new scenario.

The sequencer's response to a scenario jump trigger is controlled by the SCENARIO JUMP MODE attribute. This has three possible settings,

- When JUMP END SCENARIO is selected, the current scenario will finish playing completely before starting the next scenario. In the continuous play-back mode, it is possible that a jump trigger received close to the end of a scenario will not be recognized until the next repetition of the scenario.
- The JUMP END WFM setting will allow the waveform currently playing to finish before jumping. Because of latency issues, if a waveform is close to finishing, the next waveform may be allowed to finish before the jump is taken.
- With JUMP IMMEDIATE, the sequencer will jump as soon as possible, although there will be some latency prior to the jump.

Scenario Creation

The IVI-C function used to create scenarios is CreateArbScenario. Similar to the function to used create basic sequences (CreateArbSequence), a list of (advanced) sequence handles and repeat counts is provided to define the scenario.

In addition to this information, a list of marker masks is also provided. As explained on the section on markers below, the advanced sequencer generates marker events each time a sequence within a scenario is played. Marker masks are used to selectively enable or disable these events for each sequence within the scenario. There are three mask values which can be logically OR'd together to enable any combination of the three events:

- SEQ START MASK enables the sequence start event.
- SEQ REPEAT MASK enables the sequence repeat event.
- SEQ GATE MASK enables the sequence gate event.

A mask value of zero will disable all events. As a short-cut, a null pointer may be supplied instead which will enable all of these events for all sequences in the scenario.

Advanced Sequences

Advanced sequences are created with the CreateAdvancedSequence() IVI-C function. As with basic sequences, a list of waveform handles and repeat counts is provided to define the sequence. In addition, two new lists of information are provided. For each waveform in the advanced sequence, a waveform advance mode is specified as well as a marker mask value for waveform marker events.

Waveform Advance Mode

The playback behavior of each waveform in the sequence can be independently configured in this list. There are four choices for the waveform advance mode:

- WFM ADV AUTO: All repetitions of the waveform are played, then playback of the next waveform in the sequence begins automatically without delay.
- WFM ADV CONTINUOUS: The waveform's repeat count is ignored in this case.

The waveform plays continuously, until a waveform jump trigger is received. The jump trigger will cause the next waveform in the sequence to begin playing (using the waveform advance mode associated with the next waveform).

- WFM ADV PLAY ALL REPS All specified repetitions of the waveform are played.

  The sequencer then pauses and waits for a waveform advance trigger before playing the next waveform in the sequence (using the waveform advance mode associated with the next waveform).
- WFM ADV PLAY ONE REP One repetition of the waveform is played.

  The sequencer pauses and waits for a waveform advance trigger. The next repetition is then played, or if all repetitions have been played, then playback of the next waveform is begun.

  When the sequencer transitions to a new waveform, that waveform is always played through once before the waveform advance mode is examined. What happens next is then controlled by the waveform advance mode setting.

Waveform Marker Mask

Just as sequence events can be masked when a scenario is defined, waveform events can be masked on a per-waveform basis when defining an advanced sequence. There are three event mask values:

- • WFM START MASK enables the waveform start event.
- • WFM REPEAT MASK enables the waveform repeat event.
- • WFM GATE MASK enables the waveform gate event.

Triggering

Up to nine different trigger events (also called trigger sources) can be used to control the AWG's operation. The advanced sequencer has eight different trigger inputs which perform various operations within the sequencer. This is depicted in Figure 22 on page 45.

Trigger mapping functions allow the various trigger events to be routed (or connected) to trigger inputs on the sequencer. This is a many-to-many mapping. It is possible for one trigger event to be routed to multiple sequencer trigger inputs, and each sequencer trigger input can be connected to multiple trigger events by OR'ing attribute values together. When OR'ing multiple trigger events, be sure to use only constants which end with "FLAG".

The hardware trigger inputs are all edge-triggered, with adjustable threshold, polarity and delay. Two threshold settings are provided for the four trigger inputs. The "A" threshold is applied to inputs 1 and 2, while the "B" threshold is used for inputs 3 and 4. The IVI-C attributes TRIGGER THRESHOLD A and TRIGGER THRESHOLD B control these two threshold levels.

Polarity and delay is adjustable for each of the four inputs independently. The TRIGGER POLARITY and TRIGGER DELAY attributes are used for this purpose.

Since there are four markers, and only one of each attribute, the ACTIVE TRIGGER string attribute must first be set (to "1", "2", "3" or "4"); this specifies which of the four markers' polarity or delay is being adjusted.

Routing a single trigger event to a nonsensical combination of sequencer trigger inputs will result in undefined behavior. For example, routing hardware trigger 1 to the sequencer's start and stop triggers at the same time may not give the expected result. No warning will be issued by the IVI-C driver if this is done.

On the other hand, routing the same trigger source to both the start and waveform advance trigger inputs on the sequencer will work just fine. In this case the user will not be informed whether the a particular trigger is treated as a start or waveform advance event.

The increment with which trigger delay may be adjusted can be discovered by querying the read-only IVI-C attribute TRIGGER DELAY QUANTA.

Trigger inputs are sampled by the AWG's internal sync clock. It is possible to alter which edge of the sync clock is used to sample trigger inputs with the TRIGGER SYNC POLARITY attribute. This is for advanced usage only, and will be addressed in application notes where applicable.

Markers

There are about 20 different marker events available within the AWG when the advanced sequencing mode is used. With only four front panel marker outputs, it is obvious that only a subset of these can be used at any given time. Various IVI-C attributes are used to control which marker events are routed to which front panel marker output. Only one marker event can be routed to a specific front panel output at one time. See figure 3-5 in the User's Guide.

Marker Events

- Waveform Data Markers. Two marker bits can be stored along with each waveform. Marker bits have a granularity of eight samples. In other words, for each group of eight waveform samples, there is only one unique marker sample. In fact, the number of marker samples provided to CreateArbWaveformWithMarkers must be exactly one eighth of the number of waveform samples.
- Waveform Event Markers (see below).
- Sequence Event Markers (see below).
- Scenario Event Marker (see below).
- Hardware Trigger Event Markers. Trigger events resulting from hardware trigger inputs can be connected to a marker output. This has the benefit of synchronizing the hardware input with the AWG's internal sync clock.

Sequencer Generated Event Markers

During scenario playback, the sequencer generates various marker events which can be routed to hardware marker outputs. These events occur as the sequencer makes various transitions during playback. Events are generated as waveforms, sequences and scenarios are played. Waveform and sequences both generate three separate events during playback.

- The start event occurs once as playback of a new waveform in the sequence begins. This can be considered to be a true event that occurs at a point in time, but has no real duration.
- The repeat event also occurs as the playback of a new waveform begins, but is repeated for every repetition of the waveform as well. This should also be considered to be an event with no real duration. For a waveform with a repeat count of one, there is no difference between the start and repeat events.
- The gate event is not so much an event as it is a condition. This condition is true or asserted the entire time a waveform is being played, including all repetitions of that waveform.

Each of these events or conditions can be enabled or disabled on a per-waveform and per-sequence basis when sequences and scenarios are created.

The sequencer also provides a repeat event at the scenario level. No scenario start or gate events are available.

Gate Markers

Most sequencer event markers are truly events, which occur at some point in time and are then gone. Gate markers are different however – they are more like a state, which exists the entire time a waveform or sequence is being played.

Unexpected results may occur if the user is not aware of this fact. For example, assume two adjacent waveforms in a sequence have mask values which enable the waveform gate marker. Also assume the waveform gate marker event is routed to a front panel marker output. When the marker output is observed, it will be a constant high level the entire time both waveforms are playing, regardless of any repeat counts. There will be no brief transition to a low level between waveforms, which some users might expect. In fact, if every waveform in a sequence enables the waveform gate event, the marker will remain high at all times during the sequence, which might or might not be what was intended.

This example illustrates one of the uses for marker event masks. Assume there are three waveforms in a sequence. Disabling the waveform gate event on either side of the waveform which enables it will result in a high-going pulse with a duration equal to that of the central waveform.

Marker Routing

A marker output can only be connected to a single marker event at one time, although a single marker event can be connected to multiple marker outputs.

Marker Pulse Width

For true marker events, the output pulse width of the hardware marker is adjustable via the MARKER PULSE WIDTH attribute. However, for gate markers to be useful, the internal pulse generator should be bypassed so the actual gate appears on the marker. This is done by setting the marker pulse width to exactly zero.

Examples

Assume it is desired to play a sequence of waveforms where the playback of each waveform must be triggered by hardware trigger input 1. The nested sequencing capability of scenarios is not required. Marker output is not required.

The advanced sequencer must be used to implement this example. Call the AbortGeneration() function and set the OUTPUT MODE attribute to OUTPUT ADV SEQ.

As with basic sequencing, a set of waveforms would be created. The corresponding waveform handles will be used to define an advanced sequence. In addition to the lists (arrays) of waveform handles and repeat counts, two new arguments need to be passed to the CreateAdvancedSequence() IVI-C function. The waveform marker mask array passed as a null pointer; this enables all sequence events which is acceptable, since marker output will not be enabled.

An array of waveform advance mode values must be created, containing the same number of entries as the waveform handle and repeat count arrays. Every value in this array will be set to WFM ADV PLAY ONE REP.

Using the handle of the new advanced sequence, a one-element scenario is now created with a loop count of one. The sequence marker mask array can be passed as a null pointer since markers are not being used in this example. This takes care of creating the scenario.

To configure the scenario, first set the SCENARIO ID SOURCE attribute to ID SOFTWARE. Then set the ARB SCENARIO HANDLE attribute to the handle of the scenario that was just created.

Set the SCENARIO PLAY MODE attribute to PLAY SINGLE. The jump mode need not be set since jump triggers are not used in this example.

To get things started, two sequencer triggers need to be configured. First, set the TRIGGER SOURCE attribute to VAL SOFTWARE 1. The software trigger will be the start trigger used to kick off the sequencer.

Set the WFM ADV TRIGGER SOURCE to VAL EXTERNAL 1. Adjust the threshold (TRIGGER THRESHOLD A) as desired. Next set the ACTIVE TRIGGER attribute to one – this specifies that settings for trigger input 1 will be adjusted. Now set trigger polarity and delay as desired.

Finally, call the InitiateGeneration() function to arm the advanced sequencer. Now, issuing a software trigger (with SendSoftwareTrigger() or SendNumberedSoftwareTrigger() will issue a start trigger to the sequencer.

The first waveform in the sequence will play and the sequencer will pause until the an input is received on hardware trigger 1.

9 Application Note, Synchronization of Multiple AWGs

This application note provides detailed information about multiple AWG synchronization.

To successfully synchronize multiple M9330A Series AWG modules, it is helpful to understand a bit of the AWG's internal architecture. The AWG utilizes two clock signals during operation. The sample clock is used to strobe data into the digital-to-analog converters (DACs), and determines the AWG's sample rate. The hardware which orchestrates waveform playback and sequencing does not run at the full sample rate. Instead, a slower sync clock is derived from the sample clock and is used for this purpose. For two or more AWGs to play waveform data in unison, all AWGs must share identical copies of both sample and sync clocks. Just what the word identical means in this context is discussed in detail below.

The following topics are included in this chapter:

The Sequencer

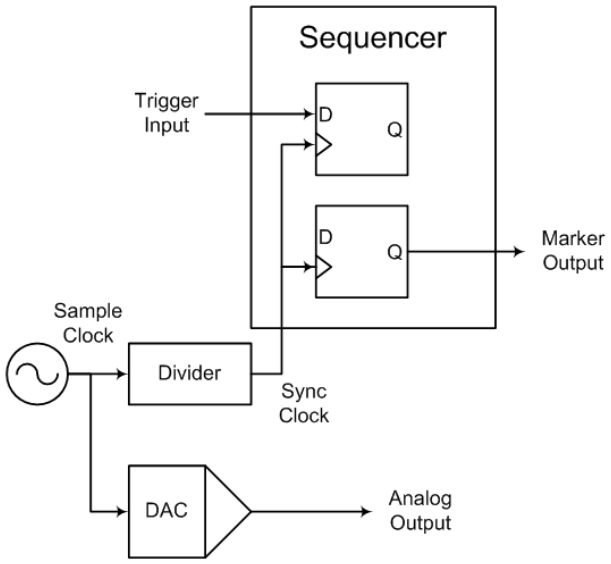
Hardware which coordinates waveform playback and sequencing of waveforms can be collectively referred to as the sequencer. Its responsibilities include generation of marker outputs and responding to trigger inputs.

Much of the sequencer's operations are based on the sync clock. For synchronization purposes, there are three important details about the sequencer's use of the sync clock.

- The point in time at which the sequencer changes the state of marker outputs is based on the sync clock.
- Trigger inputs are latched on the user-specified edge the sync clock, and the sequencer samples the latched result periodically based on the sync clock.
- For proper sequencer operation, a specific phase relationship must exist between sequencer and sample clocks.

Figure 44 provides a grossly over-simplified, but conceptually useful illustration of the sequencer's use of the sync clock.

Figure 44 Sequencer's Use of the Sync Clock



External Clock Usage

The discussion up to now has assumed the internal sample clock is in use. Synchronization of multiple AWG's is also possible with a user-supplied sample clock.

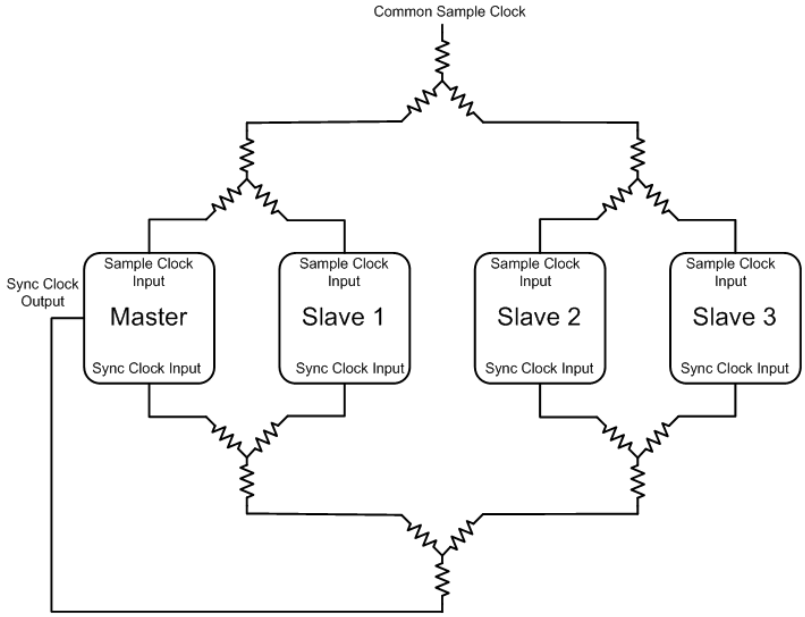
Regardless of the sample clock source, the sync clock is generated by dividing the sample clock. Sync clock frequency is maintained within certain limits to guarantee proper sequencer operation. When using an external sample clock, it is important to accurately specify the actual sample rate; this allows the AWG to keep the sync clock's frequency within proper limits.

Clock Distribution

Synchronizing multiple AWGs requires common sample and sync clocks be distributed to each AWG. This is achieved by designating one AWG as the master. All other AWGs are considered slaves. Figure 45 depicts the way clocks should be distributed in order to synchronize four AWGs.

A common sample clock is distributed to all AWGs. This clock can come from the master, or be externally generated. The relative phase of the sample clock at each AWG must be matched. The actual delay between the common clock and each AWG is not important, but the delay should be the same for all AWGs. Be sure to take into account the delay within each splitter if they are not identical.

Figure 45 Common Sample Clock with Master and Slaves



The master AWG produces a sync clock output with a controlled phase relative to its sample clock input. This sync clock must be split and distributed to the sync clock input of all AWGs, including the master. For proper operation, the delay between the master's sync clock output and the sync clock input on each AWG must be maintained within certain limits. This delay is a function of frequency; as such, changing the sample rate may necessitate altering the cable lengths used to distribute sync clocks. Agilent provides kits consisting of cables and splitters which are known to produce the proper delay two different ranges of sample clock frequency.

Clock amplitudes are such that an 8-way split will still deliver adequate amplitudes to each AWG's sample and sync clock inputs.

Triggering

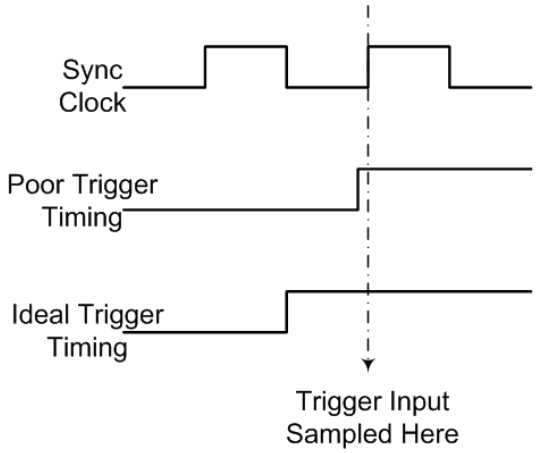
Once sample and sync clocks are properly distributed, AWG playback can be started synchronously. This is accomplished by distributing a common hardware trigger to each AWG.

As noted earlier, each AWG samples its trigger inputs at points in time determined by the sync clock. If the trigger input changes at a point in time close to when the input is sampled, it can result in erratic triggering. Therefore, it is important that trigger inputs be synchronized to the sync clock. There are two ways to accomplish this.

- Because marker outputs are also synchronized with the sync clock, they can be successfully used as a trigger if the proper polarity is selected.
- The sync clock can be used to externally synchronize trigger inputs.

Figure 46 shows examples of proper and improper timing of hardware trigger inputs.

Figure 46 Example of Proper and Improper Timing



This is not the entire story, however. Figure 46 shows the timing as seen by the trigger latching hardware. In reality there is internal cabling inside the AWG which results in time delays between the front panel and the hardware trigger latch. The timing of markers produced on the front panel is such that everything works correctly if cable lengths between marker outputs and trigger inputs are kept shorter than about 30 cm. The timing of markers versus the sync clock at the front panel will not look like Figure 46, because of these cabling delays.

Trigger Configuration Options

When the basic arb or sequence modes are used, the IVI-C driver will automatically configure one marker output (#4) on the master. Also, one trigger input (#4) and trigger voltage level on all AWGs is preset for this purpose. It is expected that marker #4 from the master AWG will be connected in parallel to the trigger #4 input on all AWGs (including the master). Trigger inputs are high impedance; 50-ohm splitting is not required. The user cannot alter the marker/trigger setup.

IVI-C requires that the InitiateGeneration function cause playback to begin, as long as the operation mode is "continuous" (as opposed to "burst"). When the continuous operation mode is selected, marker #4's source is forced to be a software marker (#4). The call to initiate generation behaves differently for the master and its slaves. When slaves are told to initiate, they are made ready to play, waiting for the pre-configured start trigger. The master is also made ready by the initiate call, after which an output is generated on marker #4 that will cause all AWGs to start playing.

In burst mode, a source for the master's marker #4 output must be chosen. Here, the call to initiate generation works identically for slaves and the master; the user must cause an output on marker #4 to start playback. Typically, the source for marker #4 will be set to one of the following:

- One of the software marker sources. This allows playback to be started directly by the software.
- One of the remaining three hardware trigger inputs. This allows playback to be started by an external hardware source. Since proper trigger timing is guaranteed through marker #4, timing of the external trigger input is not critical.

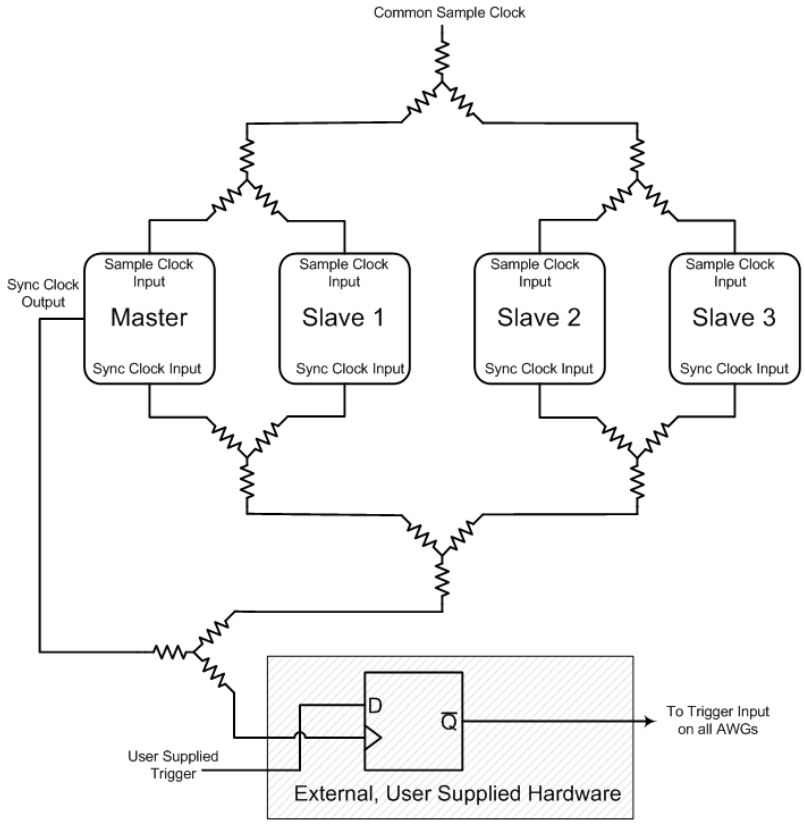
External Trigger Synchronization

It is also possible to externally synchronize a trigger, as shown in Figure 47. In this example, notice that the sync clock passes through three splitters before arriving at each AWGs' sync clock input. This reduces the sync clock amplitude to the minimum acceptable level. Further reduction in amplitude may give erratic results. We've assumed the external sync hardware has a 50-ohm input. Other arrangements are also possible.

When externally synchronizing a trigger in this fashion, the user must ensure proper timing of the synchronized trigger. Due to internal cabling delays in the instrument it is not possible to specify an exact timing. Instead, the user must empirically determine the proper delay. One approach is to empirically locate the timing point which gives erratic triggering, and add or subtract 180 degrees of delay at the sync clock frequency.

Figure 47 is intended to be a conceptual diagram only. In practice, users may find it necessary to double-latch the trigger signal to deal with the possibility of exciting meta-stable states in the first latch.

Figure 47 Externally Synchronize a Trigger



Matlab

The discussion till now has assumed the user is using the IVI-C driver directly or through LabView. Synchronization requirements when controlling the AWG with Matlab are identical.

The Control Utility (GUI)

When using the control utility, a software trigger is generated, producing an output on marker #4 at the front panel, which starts playback. This occurs when the "Download and Play" button is clicked.

Advanced Sequencing

If more flexibility in triggering is desired, advanced sequencing mode must be used. In this mode, triggers and markers are not pre-configured; the user must take care of all triggering configuration. The following steps must be taken prior to starting playback:

- Select and configure an appropriate start trigger source for each AWG. This will usually be one of the front panel hardware trigger inputs, although it need not be the same input on each AWG.
- Configure and connect a common trigger signal to each AWG's configured trigger input. This will typically be one of the marker outputs, although there are other options as shown in Figure 47.

In any case, it is crucial that the actual trigger input be received at each AWG during the same sync clock cycle, and displaced in time from the active sync clock edge. If these requirements are not met, The AWG's may not begin playback at the same time, and will not be synchronized. If this occurs, increasing or decreasing trigger cable lengths by a few inches or several cm will usually correct the situation.

10
Frequently Asked Questions

The following is a list of Frequently Asked Questions (FAQs):

1. **Q. [Power On] If the M9330A Series AWG modules is powered on after the PC is already powered on, does the PC recognize the M9330A Series AWG modules.**

   A. The M9330A Series AWG modules must be powered on before the PC is powered on. The PC looks for PXI devices (the M9330A Series AWG modules) only once during PC power-on. So, unless the M9330A Series AWG modules is powered on prior to the PC, the PC will not find it.

2. **Q. [Reset] How do I reset the M9330A Series AWG modules?**

   A. When using the IVI-C command "agn6030a_init", pass a "1" to the "resetDevice" input.

3. **Q. [Self-Test] How do I run a self-test on the M9330A Series AWG modules and How can I tell if the self-test command ("agn6030a_self_test") is actually running?**

   A. It can be done programmatically using IVI-C commands.

   A self-test is required by the IVI standard and there is a command in the driver to run it, but there is no real self-test to be run; it is a stub-out.

   The real self-test is to initialize the unit by starting the M9330A Control Utility. If it initializes without error, all the digital circuitry is verified. Stuck bits in the DACs and signal path fidelity through all possible paths following the DACs are not checked; neither are markers or triggers. These checks aren't possible anyway without fancy loopback circuitry, canned waveforms, and extra cables or internal switching.

4. **Q. [Initialize Unit] Do I need to initialize the unit when programming using IVI-C commands?**

   A. Yes. When using the IVI-C command "agn6030a_init", pass a "1" to the "resetDevice" input.

   Note: Starting the M9330A Control Utility automatically resets the device.

5. **Q. [IO Libraries] What version of Agilent IO Libraries Suite should I use?**

   A. IO Libraries Suite version 16.0 and above should be used for all supported operating systems (Windows XP, Vista and 7).

   For every individual Agilent instrument, Agilent IO card (GPIB card, converters) and development copy of Agilent T&M Toolkit or VEE Pro that you own, you are entitled to one non-exclusive license of this product at no charge.

   Please visit our web site for other information, including how to purchase additional copies, at www.agilent.com/find/iosuite.

   If you need further assistance, please visit www.agilent.com/find/assist to talk to an Agilent engineer.

6. **Q. [IVI Compliance] What version of the IVI Compliance package should I be using?**

   A. Version 3.2 or above of the IVI compliance package should be used with M9330A Control Utility and IVI-C Drivers.

   NI IVI Compliance Package can be downloaded from National Instruments at: ftp://ftp.ni.com/support/softlib/ivi/ICP

7. **Q. [Control Utility] Can the N8241A or N8242A use the M9330A Control Utility?**

   A. The M9330A Control Utility can not be used with the N8241A/N8242A and the N8241A/N8242A Control Utility can not be used with the M9330A Series AWG modules!

- If you installed the M9330A Control Utility on your PC, un-install it before installing the N8241A/N8242A Control Utility.
- If you installed the N8241A/N8242A on your PC, un-install it before installing the M9330A Control Utility.

  Both the M9330A Control Utility and the N8241A/N8242A Control Utility can not be installed on the same PC because they share a common .dll name which has different content depending on model. The "Control Utility and IVI-C Driver" download can be found at the following:

  - Go to: http://www.agilent.com/find/AWG
  - Select from models: M9330A and M9331A
  - Select the Technical Support/Drivers & Software tab.
  - Download and install the newest version of the Control Utility and IVI-C driver

before using your AWG.

8. **Q. [Correction File] Is there an M9330A Correction file?**

   A. If you would like to insert correction coefficients into the M9330A Series AWG modules, as is done with spectrum analyzers or signal generators where the user can go to "amplitude corrections" and enter a list of coefficients, you can't because the M9330A Series AWG modules doesn't have any internal memory for corrections. There is no way to insert correction coefficients using the M9330A Control Utility. (Refer to the next FAQ for information about using Matlab and a Correction File.)

9. **Q. [Matlab / Correction File] Can I add an M9330A Series AWG modules Correction file using Matlab when the M9330A Series AWG modules is connected to an Agilent IQ synthesizer which is connected to an Agilent spectrum analyzer.**

   The M9330A Series AWG modules can be tested at different frequencies to find the correction for Amplitude and Phase at each frequency that gives the best SFDR.

   How do I add this correction to my Matlab unknown wave if the user is creating a wave and I want to add the correction to the wave?

   A. Perform an FFT on the wave, add the correction, and then perform an inverse FFT.

- One way to apply correction data: Use an FFT on the waveform, add the correction values, and perform an inverse FFT to apply the correction, but there may be unintended problems and so it is not a safe way to add corrections.
- Another way to apply correction data in Matlab is by creating a FIR filter and applying it to the unknown waveform before downloading to the M9330A Series AWG modules.

  If the correction data was measured with predistortion on, predistortion will need to be on with the FIR filter.

  Perform the following steps with Matlab open:
  - Click on “Start” in the lower left corner.
  - “Toolboxes”
  - “Signal Processing”
  - “Filter Design & Analysis Tool”
  - In the “Filter Design & Analysis Tool” window:
  - Response Type – select “Arbitrary Magnitude”
  - FIR – select “Least-squares”

  Then add in the Frequency and Magnitude Specifications.

10. **Q. [Matlab] How fast can the M9330A Series AWG modules switch between two sequences programmed with Matlab?**

    A. This is a way to achieve frequency hopping if you know the order of the desired frequencies. If you use an external method of determining the order, Option 300 Dynamic Sequencing may be a better way to frequency hop. Using either method, the change in frequency is <10 ns.

    This number can be less depending on how it is measured. It depends on how many zero crossings are counted and if there is an amplitude change. A sequence must contain at least two waveforms. The total number of samples in a waveform must be greater than or equal to 128. The total number of samples in a waveform must be a multiple of 8. The fall time is less than 1 ns.

11. **Q. [Markers] Can the M9330A Series AWG modules markers be programmed on a sample by sample basis?**

    For example, could I use a marker as a 1 bit arb running at the same clock rate as the ADCs?

    If the markers are not arbitrarily programmable, what is the fastest (highest repetition rate) 50% duty cycle signal that can be programmed (assuming the use of the internal 625 MHz clock and a 5 ms duration waveform)?

    A. No, the M9330A Series AWG modules markers cannot be programmed on a sample by sample basis.

    - The markers are generated in the FPGA which runs at fs/8 (which for an M9330A Series AWG modules is {1250 MSa/s}/8 or {625 MSa/s}/8).
    - The low voltage TTL chip that distributes the marker signal is not a high speed chip.

12. **Q. [Security] Does the M9330A Series AWG modules have a cleaning procedure for DOD security?**

    A. The M9330A Series AWG modules does not have any non-volatile RAM or media that stores waveform information or settings. The security procedure is to turn off the power to the M9330A Series AWG modules and turn it back on. The FPGA code gets reloaded and the SRAM is reset with zeros loaded into the SRAM when the M9330A Control Utility is started.

13. **Q. [Synchronization] How do I synchronize two M9330A Series AWG modules?**

A. Using the recommended Agilent cables and power dividers, use the internal 1.25 GS/s clock and play a 500 MHz tone. The outputs of M9330A Series AWG modules 1 and M9330A Series AWG modules 2 should be within 30 picoseconds of each other. Use a DSO81304A oscilloscope (or equivalent to verify) with equal length cables from the M9330A Series AWG modules outputs to the oscilloscope. (Refer to M9330A Series AWG Modules Option Y1176A Synchronization Cable Kits Installation Note, PN M9330-90007.)

14. **Q. [Sequencing] What is the difference between Advanced Sequencing and Dynamic Sequencing?**

    A. Dynamic Sequencing is Option 300. Its only contribution is to allow the scenario addresses and triggers defined by Advanced Sequencing to be directed through the AUX connector on the front panel. This provides a low-latency address strobe in addition to the hardware trigger which allows near-real-time selection of pre-defined scenarios and triggers.

15. **Q. [Sequencing] What is the shortest waveform that can be defined in a memory segment?**

    A. The shortest waveform that can be defined in a memory segment is 102.4 ns long. This is 128 samples of 800 ps duration each. If waveforms shorter than 102.4 ns are required, they must be concatenated together to reach the 128 sample minimum and then stored in memory. There are no exceptions to this rule. The number of sequences available is 32767 (32 k); the number of scenarios available is 16383 (16 k).

    To get start and stop trigger modes to support the customer use model, advanced sequencing must be used. This is included in every instrument. We may be able to discover a way to fit the pre-defined sequences into the allowed space with some creativity, in other words, there may not be more than 16 k unique scenarios required, just lots of combinations that are reused. Remember, scenarios are made up of sequences, which are made up of memory segments (32 k x 16 k is a big number).

    The M9330A Series AWG modules clock must be stable. Any attempt to apply/un-apply, move around, or interrupt the M9330A Series AWG modules clock will result in excessive irregularities.

    So, the first thing to do is determine if combinations of 10 ns segments that will satisfy the 128 sample rule can be defined. Once we can determine this, we can move onto determining if the number of unique sequences and scenarios to be addressed exceed the limits.

16. **Q. [Triggering] The M9330A Series AWG modules trigger in specifies a maximum input voltage of 4.5 V. When I measure the trigger out of the M9330A Series AWG modules into a 4.7 k ohm load it's 5 V. Does this mean the trigger out of one M9330A Series AWG modules can possibly exceed the safe trigger in level of another M9330A Series AWG modules? Is the device incompatible with its own trig in/ trig out?**

    A. There's really no problem here. Marker outputs are generated by logic circuits supplied from ground and +5 V. There are 30 ohm series resistors on the outputs and the "internal" end of the resistor is diode-clamped to the supply voltages (in case someone was to attempt driving the marker output connector with a source). Trigger inputs have a 10 ohm series resistor with the "internal" end of the resistor clamped between –5 V and +5 V. Damage will not occur until you pass enough current through the 10 ohm resistor to pop one of the clamp diodes. The main point is that it is perfectly safe to connect our markers directly to trigger inputs -- we intended it to work this way. The 4.5 V number is the limit of adjustment for trigger levels (plus and minus). In most cases, it is acceptable to run markers un-terminated to trigger inputs (they have a 2 k ohm input impedance). There will be reflections on the cable, but this is usually not important for triggering to function correctly. If this becomes a problem, the marker output should be treated as a 50 ohm source, split if necessary, and 50 ohm terminations should be provided with tee connectors as close as possible to trigger inputs. Trigger levels should be lowered accordingly. This is often not necessary, but it is something to investigate if triggering problems arise.

17. **Q. [Option 300] Does Agilent provide a cable as part of Option 300 for plugging into the AUX Port?**

    Option 300 uses a 3M Mini-D connector.

    A. No, Agilent does not provide a cable that plugs into the AUX Port.

18. **Q. [NIMax] Why doesn't the M9330A Series AWG modules show up in NIMax as a PXI device?**

    A. NIMax checks the configuration register of the M9330A Series AWG modules and since NIMax doesn't know how to control the M9330A Series AWG modules, it does not show up.

19. **Q. [Predistortion] What are the details related to using predistortion?**

    A. The built-in predistortion scales the data to guarantee that you never get a DAC over-range condition. Typically, it gives up 2-3 dB of range. This is important because less power out of the M9330A Series AWG modules means less leveled output power

from the PSG with Option 016. Making sure that M9330A Series AWG modules use the full range of the DAC ensures that the PSG can provide a minimum leveled output signal of 0 dBm! Higher leveled output power of 5-10 dBm (depending on the signal and the RF frequency) is achievable.

20. **Q. [PXI Address String] Why does the PXI address string have an extra number in it (PXI1::13::0::INSTR as opposed to just PXI1::13::INSTR)?**

    A. The PXI Address string has the extra number for "sub devices" which the M9330A Series AWG modules doesn't have.
    If you use "PXI1::13:INSTR" as the M9330A Series AWG modules address, it works the same as "PXI1::13::0:INSTR".

11 Troubleshooting

The following topics are included in this chapter.

Software

Removing the Software

If it is necessary to remove the **M933x** software, go to:

Start > **Settings** > **Control Panel** > **Add/Remove Programs**

Remove **Windows Driver Package - Agilent Technologies Agilent M933x Device Driver**.

Remove **Agilent M933x** Software.

Remove **AgM933x IVI Driver xx.x**.

You may need to restart the controller, but this will completely remove all of the M933x files.

Moving the Software

If it becomes necessary to move the **Agilent M933x** software, complete the instructions documented in "Removing the Software". Then, using the M9330A Series CD, reinstall the software where it is needed.

Updating the Software

To resolve an error message you get while attempting to upgrade the **M933x** software, take the following steps in the listed order:

Go to: **Start** > **Settings** > **Control Panel**.

Double-click **Add/Remove Programs**.

Select **Windows Driver Package-Agilent Technologies Agilent M933x Device Driver**, and select **Remove**.

Select **Agilent M933x Arbitrary Waveform Generator**, and select **Remove**.

Reinstall the **M933x** software.

Contacting Agilent

Agilent on the Web

You can find information about technical and professional services, product support, and equipment repair and service on the Web:

http://www.agilent.com

Click on the Test & Measurement link then click on Select a Country. Click on the Contact Us link for contact information.

11 Troubleshooting

12
Characteristics

The following topics are included in this chapter.

Technical Characteristics

Channels

Two independent channels available as baseband or IF outputs

CH1: Single-ended and differential

CH2: Single-ended and differential

Modulation Bandwidth

500 MHz per channel (1 GHz IQ bandwidth)

Resolution

M9330A Series AWG module operates at 1.25 GS/s, 15 bits of vertical resolution
M9331A Series AWG module operates at 1.25 GS/s, 10 bits of vertical resolution

Output Spectral Purity - (CH1 and CH2)

Harmonic Distortion: 1 kHz to 500 MHz
M9330A Series AWG module: < –65 dBc for each channel
M9331A Series AWG module: < –50 dBc for each channel

Non-Harmonic Spurious: 1 kHz to 500 MHz
M9330A Series AWG module: < –65 dBc for each channel
M9331A Series AWG module: < –65 dBc for each channel

Noise Floor:
M9330A Series AWG module: < –150 dBc/Hz across the channel bandwidth
M9331A Series AWG module: < –150 dBc/Hz across the channel bandwidth

Sample Clock

Internal: Fixed 1.25 GS/s

Internal clock output: +3 dBm nominal into 50 ohm load

External clock input: Tunable 100 MS/s to 1.25 GS/s

External clock input (power): –15 to +5 dBm, 0 dBm nominal

Phase noise characteristics:

1 kHz	-95 dBc/Hz
10 kHz	-115 dBc/Hz
100 kHz	-138 dBc/Hz
1 MHz	-150 dBc/Hz
Noise Floor	-150 dBc/Hz

Accuracy:
Same as 10 MHz timebase input

Frequency reference

Input drive level: +2 to +12 dBm into 50 ohms (+2 dBm nominal)

Waveform length

8 MS per channel (16 MS with option M16)
Minimum waveform length: 128 samples
Waveform granularity: 8 samples

Segments

From 1 to 32,768 unique segments can be defined consisting of waveform start and stop address, repetitions and marker enable flags.

Sequences

Up to 16,384 total unique waveform segments can be combined with separate loop counts to form a sequence.

External Triggers

Number of inputs:	5 each (4 SMB female front-panel connectors plus one software trigger over PCI backplane from host processor)
Trigger polarity:	Negative/positive
Trigger impedance:	2k ohms
Maximum input level:	4.3V
Input sensitivity:	250 mV
Trigger threshold:	-4.5 to + 4.5 V
Trigger timing resolution:	Clock/8 (6.4 ns at full rate)
Trigger uncertainty:	< 50 ps
Minimum trigger width:	12.8 ns at full clock rate
Trigger delay:	resolution of 1 SYNC clock

External Markers

Markers can be defined for each waveform segment.

Number of outputs:	4 each SMB female
Marker polarity:	Negative, positive
Output type:	3.3V CMOS with 30 ohm series termination
Marker low level:	100 mV nominal (high impedance load)
Marker high level:	3.2V nominal (high impedance load)
Marker timing resolution:	Clock/8 (6.4 ns at full rate)
Marker delay:	resolution of 1 sample clock
Marker width:	resolution of 1 SYNC clock

Module Synchronization

Hardware supports synchronization of multiple modules with future software enhancements.

Sync clock output level: 800 mV p-p 50 ohm output impedance, AC coupled
Sync clock input level: 100 mV p-p into 50 ohms AC coupled

Analog Output

Output connector: SMA female
Output impedance: ~50 ohms

Analog Output Levels

The following output levels are specified into 50 ohms:

	Single-Ended	Differential
Passive Mode	0.5Vp-p	1Vp-p
Active Mode	1Vp-p with +/-0.2Vp-p	N/A

Uncorrected passband flatness: +/- 1 dB DC - 200 MHz; +/- 3.5 dB DC - 500 MHz (with 1.25 GHz clock)
Uncorrected passband group delay: +/- 500 ps DC - 200 MHz; +/- 1 ns DC - 500 MHz (with 1.25 GHz clock)

Reconstruction filters

500 MHz and 250 MHz realized as 7-pole elliptical filters plus thru-line output

General Characteristics

Power

Supply	Typical Operation (Watts)
+3.3 VDC	11.2
+5 VDC	22
+12 VDC	5
-12 VDC	5
Total Power	43.2

Environmental

Samples of this product have been type tested in accordance with the Agilent Environmental Test Manual and verified to be robust against the environmental stresses of Storage, Transportation, and End-use; those stresses include but are not limited to temperature, humidity, shock vibration, altitude, and power line conditions. Test methods are aligned with IEC 60068-2 and levels are similar to MIL-PRF-28800F Class 3.

Operating temperature:	0 to +55 degrees C
Storage temperature:	-20 to +70 degrees C
Relative Humidity:	
Type tested:	10 to 90% at 40 degrees C (non-condensing)
Altitude:	0 to 2000m (6500 ft) above sea level

Safety

Complies with European Low Voltage Directive 2006/95/EC
* IEC/EN 61010-1 2nd Edition
* Canada: CSA C22.2 No. 61010-1
* USA: UL 61010-1 2nd Edition

EMC

Complies with European EMC Directive 2004/108/EC
* IEC/EN 61326-1 or IEC/EN 61326-2-1
* CISPR Pub 11 Group 1, class A
* AS/NZS CISPR 11
* ICES/NMB-001

This ISM device complies with Canadian ICES-001.

Cet appareil ISM est conforme a la norme NMB-001 du Canada.

[EMC compliance tested with EMC shielded filler panels (Agilent P/N N6030-80007, kit of 6) separating the controller and M9330A Series AWG modules, and in all open slots. The RFI gaskets must be oriented to the right.]

Weight

1.14 kg (2.5 lb)

Security

All user data stored in volatile memory

Dimensions

3U, 4 slot PXI module: 8.1 x 13 x 21.6 cm (3.2 x 5.1 x 8.5 inches)

Test Certificate

This instrument went through a complete functional test.
A Functional Test Certificate is provided with this instrument.

ISO compliance

This modular instrument is manufactured in an ISO-9001 registered facility in concurrence with Agilent Technologies, Inc. commitment to quality.

Options

M9330A Series AWG module and
M9331A Series AWG module

M9330A-M16: Waveform memory expansion to 16 MS per channel
M9330A-300 Dynamic Sequencing
M9330A-330 Direct Digital Synthesis (DDS)

M9330A-350 Function Generator Software

Declaration of Conformity

The Declaration of Conformity (DOC) is on file. If a copy is required, please contact an Agilent Sales Representative or the closest Agilent Sales Office. Alternately, contact Agilent at:

http://www.agilent.com/find/assist

The Agilent Technologies M9330A Series AWG Modules share the same hardware covered by this DOC.

Index

Index

S

T

U

W