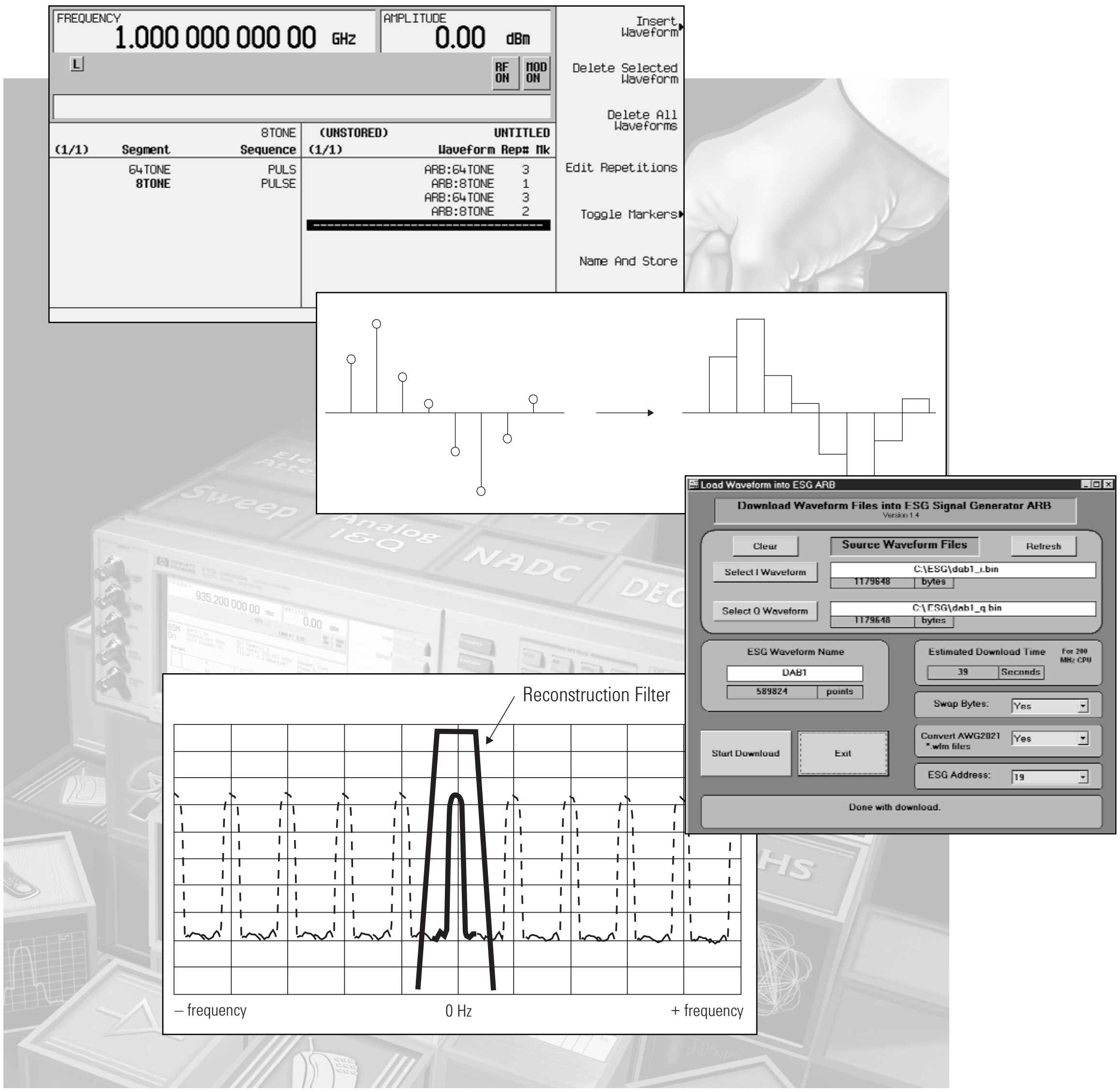
Agilent Option UND

Generating Digital Modulation with the Agilent ESG-D Series Dual Arbitrary Waveform Generator

Product Note



Agilent ESG-D Series RF Signal Generators

Agilent Technologies

Table of Contents

Introduction

Option UND is one of two internal baseband generators for the Agilent Technologies ESG family. This dual arbitrary waveform generator provides extremely flexible baseband generation for the most complex RF waveforms. With the capability to drive the ESG-D's I/Q modulator, the internal dual arbitrary waveform generator provides the power to simulate complex, nonstandard, or proprietary modulated RF signals. These modulating waveform files can be generated in a variety of external simulation tools or by the Agilent ESG-D's available built-in firmware personalities.

This product note will introduce the hardware structure and features of Option UND, dual arbitrary waveform generator, then follow with techniques for creating I and Q waveforms to download. Hardware limitations will also be addressed. Program examples and utilities for creating digital waveforms are included in Appendices and are available for download from www/agilent/com/find/esg/.

Complementary product note

A related product note on applying the techniques described in this note was planned for June 1999. The proposed topic was using MATLAB to add interfering signals to NADC signals. For updates on availability, please check the Agilent ESG website.

The two baseband generators in the Agilent ESG-D

The Agilent ESG-D series of RF signal generators offers two highly flexible, complementary baseband generators for complex digitally modulated signals. With either of these baseband generators, you can easily simulate existing communications standards, modify existing digital protocols, define or create digitally modulated signals, or intentionally impair the baseband signals. Because both baseband platforms are programmable, new digital modulation formats can be added in the future by loading new firmware into the Agilent ESG-D.

The two baseband architectures available in the ESG-D series are Option UN8, the real-time I/Q baseband generator and Option UND, the internal dual arbitrary waveform generator. It is convenient to think of Option UN8 as the in-channel baseband generator better suited to receiver test applications, and Option UND as the out-of-channel generator well matched to component test applications.

Option UN8 easily simulates a single communications channel by giving the user access to the building blocks of digital modulation. From the instrument front panel, you can select from a variety of modulation types, FIR filters, and symbol rates. Data sources can be generated internally, downloaded, or supplied real-time to an external input. For more information on using the Option UN8, real-time I/Q baseband generator, please refer to the Agilent product note, *Customize Digital Modulation with the Agilent ESG-D Series Real-time I/Q Baseband I/Q Generator, Option UN8*, literature number 5966-4096E.

Option UND gives completely arbitrary I/Q waveform generation capability without the ability to modulate real-time data. Typical applications with Option UND include:

- simulating digitally modulated signals with up to 20 MHz bandwidth
- generating two or more CW tones with one ESG-D
- constructing multiple communications channels
- generating a signal that includes noise or other impairments
- creating multiple modulated RF carriers, such as mixed NADC and CDMA carriers for base-station amplifier testing.

How to use this note

This product note describes how Option UND works, its design features for digital modulation, and how to use it to create arbitrary digital modulation signals. This note assumes a basic understanding of digital modulation concepts and instrument programming with Standard Commands for Programmable Instruments (SCPI). Program examples included are based on MATLAB from the MathWorks.

The product note is made up of sections specific to applications. A description of each section follows.

Dual arbitrary waveform generator. This section provides an overview of the internal hardware and features of the ESG-D, including the dual arbitrary waveform generator and the I/Q modulator. This is useful for those who would like an overview of the basic capabilities and operational concept of a signal generator with vector modulation and an internal dual arbitrary waveform generator.

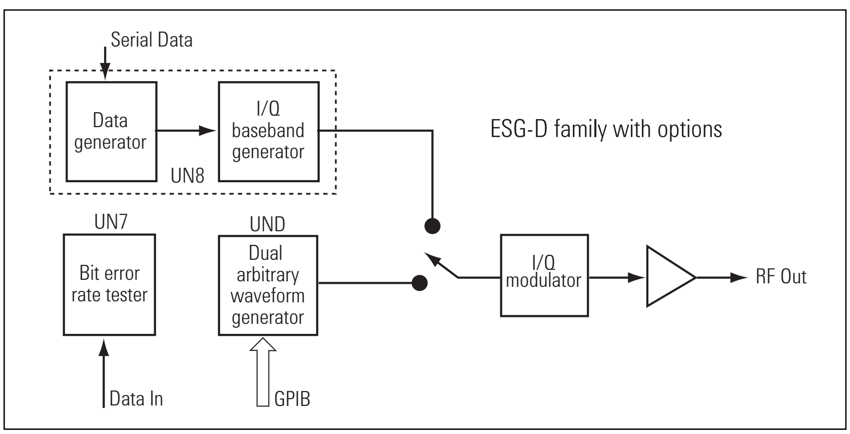


Figure 1. Complementary digital baseband generation and bit-error-rate test in the Agilent ESG-D series

Introduction, continued

Waveform generation. This section describes the waveform generation process in detail, including concentration on the following topics:

Basic digital transmitter. This section describes a generic digital transmitter to be used as the framework for the ensuing discussion of creating waveforms.

Hardware constraints. This section addresses some of the constraints of the hardware, and their effect on waveform generation. This information is important for any waveform generation.

Formatting and downloading data. This section describes the data format used to store waveform data in the ESG-D. Those users who will use one of the formatting and downloading utilities found at www.agilent.com/find/esg/ may not need to read this section.

Advanced techniques. This section contains advanced techniques that can be used to create waveforms specific to an application.
They include:

- *phase continuity* for creating waveforms that will be repeated continuously in the ESG-D.
- *multicarrier signals* for simulating multple modulated or unmodulated carriers with a single ESG-D.
- *maximizing effective I/Q bandwidth* for wide bandwidths (typically >10 MHz).
- *design in the frequency domain* for modlation formats (like OFDM, or orthogonal frequency division multiplexing) that involve symbol building in the frequency domain, followed by an inverse Fourier transform.

Resources. This section lists support sources and where to get more information from Agilent (and others).

The minimum configuration for the examples presented in this note is:

- Agilent ESG-D
- Options UND and UN5
- Option H99 (recommended for optimal adjacent channel power)
- A PC with MATLAB 5.0 or higher with the Signal Processing Toolbox, and a GPIB interface.

Examples. The examples in this product note use the following convention: Softkeys (redefined by context) are denoted by **bold type**. Hardkeys are denoted by underlined type.

For additional information about these and other topics, the following application and product notes are available through your local Agilent sales office or at www.agilent.com

- *Customize Digital Modulation with the Agilent ESG-D Series Real-time I/Q Baseband Generator, Option UN8*, literature number 5966-4096E.
- *Digital Modulation in Communication Systems–An Introduction*, literature number 5965-7160E.

More general references are listed in a separate “References” section at the end of this note.

Dual arbitrary waveform generator

The internal dual arbitrary waveform generator in the Agilent ESG-D series of RF signal generators is used in a similar way to external arbitrary waveform generators used for digital modulation. To understand the operation of the dual arbitrary waveform generator, it is important to understand how it fits in the overall block diagram of the ESG.

Agilent ESG block diagram

Refer to Figure 2. The dual arbitrary waveform generator delivers I and Q signals that drive the I/Q modulator on the output board, which modulates the synthesized LO. The automatic level control (ALC) then adjusts this signal for an extremely accurate power level.

I/Q modulation signals

The I/Q modulator of the ESG accepts inputs from various sources. In addition to I and Q signals from the dual arbitrary waveform generator, the I/Q modulator can use as inputs external I and Q inputs (which can be swapped internally), an internal calibration signal, or inputs from another baseband generator, such as Option UN8, real-time I/Q baseband generator.

These I and Q signals can then be adjusted for I and Q offset or I and Q gain before they are applied to the I/Q modulator. In the I/Q modulator they are applied, in quadrature phase offset, to the carrier LO. The quadrature phase relationship of the I and Q signals in the modulator can also be adjusted. The LO used in the modulator is a 250 MHz to 4 GHz (depending on the model's frequency range) signal provided by the frequency synthesis section of the ESG.

Frequency synthesis

While most stages of frequency synthesis in the ESG are of little importance in the use of the dual arbitrary waveform generator, there are two distinct bands of interest: carriers at or above 250 MHz, and carriers below 250 MHz (heterodyne band).

For carriers greater than or equal to 250 MHz, a LO at that frequency is directly modulated by the I/Q modulator and passed on through the output section.

For carriers below 250 MHz, frequency synthesis is finalized after I/Q modulation by heterodyning the modulated signal with a 1-GHz LO from the reference section. Since the modulated carrier before heterodyning is set between 750 MHz and 1 GHz, the resulting modulated RF carrier is an image of the baseband signal, which is reversed in frequency.

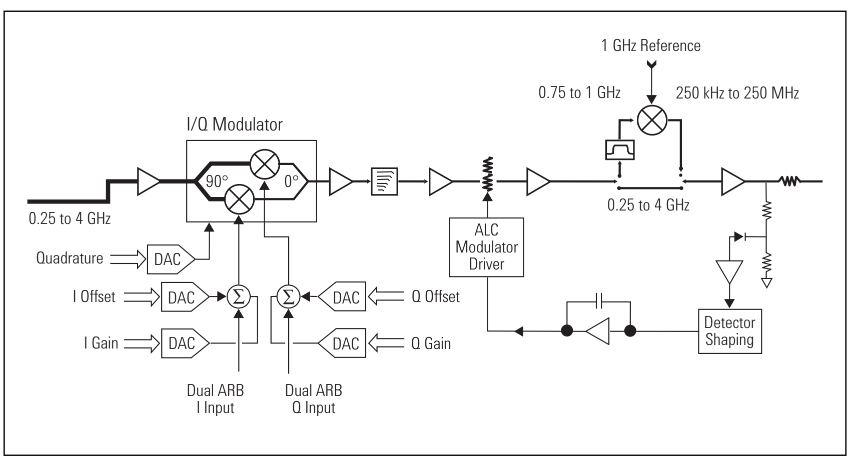


Figure 2. Block diagram of the Agilent ESG-D with dual arbitrary waveform generator

Dual arbitrary waveform generator, continued

Modulated signals at IF frequencies
This method of heterodyning will swap I and Q in signals that are modulated on carriers below 250 MHz. This can be overcome by connecting the rear panel I and Q outputs to the front panel I and Q inputs, respectively, and selecting external I/Q as the modulating signal. The firmware of the ESG will compensate by switching these signals for a heterodyne band carrier. The firmware does not compensate for this effect when using I and Q signals from the dual arbitrary waveform generator.

Automatic level control (ALC)
The automatic level control of the ESG maintains a calibrated repeatable power level at the RF output. It is programmed to account for varying power spectral densities of modulated signals, but it can treat low-rate modulation as output level fluctuation and try to correct for it. Modulating signals with a baseband bandwidth up to 100 kHz may experience unwanted amplitude modulation as the ALC tries to compensate for this fluctuation. This distortion is dependent on modulation format and usually results in degraded EVM.

To prevent amplitude fluctuations in response to low-rate modulation, turn off the ALC (Ampln→ **ALC Off**), and use the ESG's Power Search function to ensure an accurate output power level.

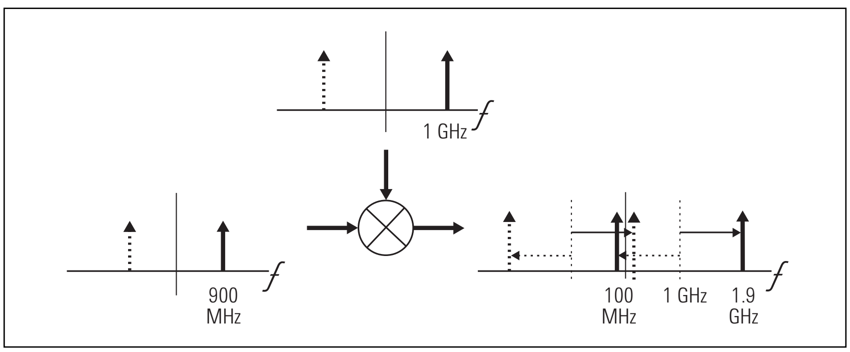


Figure 3. Spectrum "reversal" after heterodyning a complex-modulated signal to achieve a 100 MHz carrier

Dual arbitrary waveform generator, continued

Dual arbitrary waveform generator block diagram

The dual arbitrary waveform generator is designed to provide optimized I and Q signals to the Agilent ESG's internal I/Q modulator. It consists of three major blocks: a digital signal processor (DSP), a sequencer with RAM, and digital/analog converters (DACs) and reconstruction filters. The dual arbitrary waveform generator is analogous to a compact disc (CD) player with recorded music. A CD contains stored binary data that can be sequenced, converted to analog signals, and played through an amplifier and speakers. Likewise, the dual arbitrary waveform generator's RAM stores two channels of binary data (I and Q), which undergo digital/analog conversion and is used to modulate an RF carrier which is "played" through the RF output of the ESG. The blocks of the dual arbitrary waveform generator shown in Figure 4 are discussed below in detail.

Digital signal processor

The digital signal processor is used by optional personalities (e.g. CDMA, W-CDMA, CDMA2000) that generate and store I/Q modulation data in ARB RAM. The multitone personality, for instance, generates a sequence of samples that, when modulated on a carrier, simulates multiple CW tones. Personalities based on the dual arbitrary waveform generator are discussed later in this product note.

It is convenient to think of the DSP as something outside the dual arbitrary waveform generator. The user has no direct control over the DSP. However, users can generate waveforms externally and store them in ARB (volatile playback) RAM in a parallel process described below.

Sequencer and RAM

Through use of the dual arbitrary waveform generator's personalities, or by downloading external data, the user can write I and Q waveforms to ARB RAM. There are two types of RAM on the dual arbitrary waveform generator: volatile ARB RAM, and nonvolatile NVARB RAM. There are four one-Msample (1,048,576 samples) banks of RAM. The I and Q channels each have one Msample of ARB RAM and one Msample of NVARB RAM.

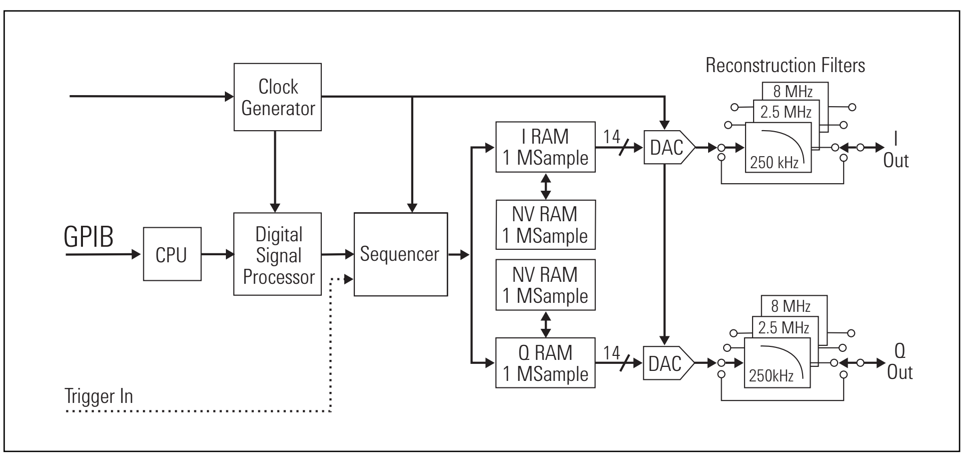


Figure 4. Agilent ESG series internal dual arbitrary waveform generator block diagram

Dual arbitrary waveform generator, continued

IARB RAM is used for waveform playback. When a waveform segment is stored in ARB RAM it is immediately available to be activated and used to modulate the RF carrier. NVARB RAM is used to store waveforms for later recall. While waveforms stored in NVARB RAM cannot be used to modulate a carrier, they can be copied quickly to ARB RAM. Waveforms stored in NVARB RAM remain when the ESG is powered off, preset or unplugged.

The dual arbitrary waveform generator's ARB RAM is directly controlled by the sequencer. The sequencer provides the memory pointers necessary to create analog signals from the digital data stored in RAM. In addition, the sequencer gives the capability to create sequences made of multiple waveform segments, or files. This is helpful when constructing long waveforms with repeating segments. A long waveform that might not fit in the available ARB RAM might consist of repetitive data that can be stored as single segments and repeated in the sequencer. Figure 5 demonstrates this concept.

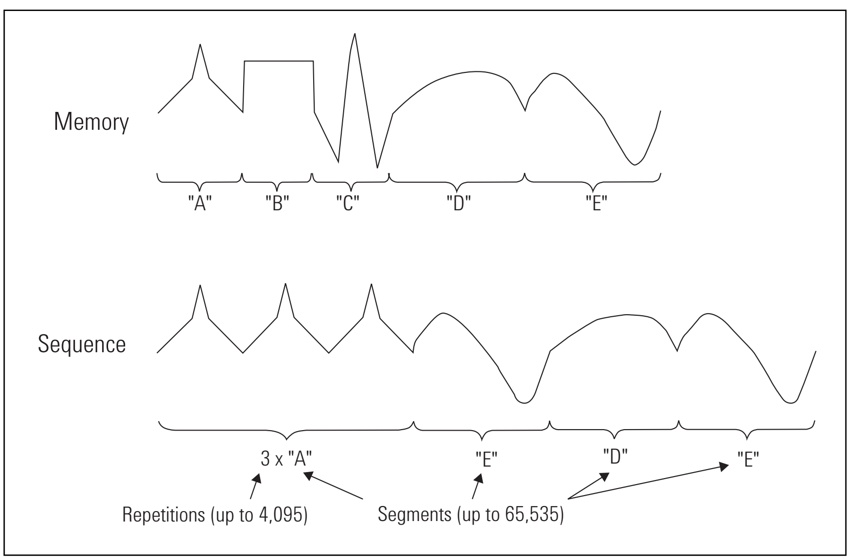


Figure 5. Using sequencing to conserve memory in the dual arbitrary waveform generator

Sequences are easily created in the sequence table editor shown in Figure 6. For each segment selected (up to 65,535) the user can turn markers on or off and select a number of repetitions, up to 4,095. For added flexibility, the user can embed another sequence as a segment in a sequence.

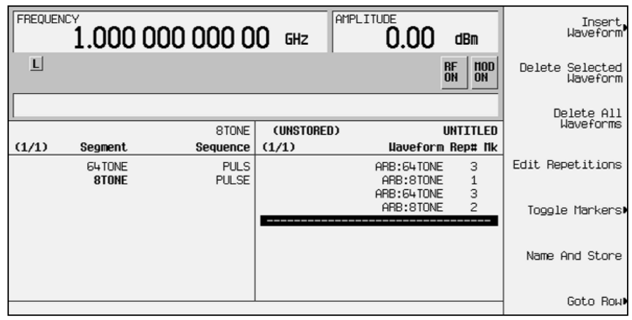


Figure 6. Agilent ESG sequence editor

Dual arbitrary waveform generator, continued

DACs and reconstruction filters
When waveforms are accessed for playback as standalone segments or as parts of a sequence, the binary data is provided to digital-to-analog converters (DACs), which build analog voltage signals that drive the I/Q modulator.

When the sequencer accesses data stored in ARB RAM, it is applied to a DAC for one sample period (set by the sample clock frequency). During that sample period, a discrete voltage level is generated at the DAC output and held until the next sample period. The DACs produce a series of quantized steps representing analog signals. The DAC used in the ESG's internal dual arbitrary waveform generator has 14-bit resolution, allowing up to 16,384 quantized voltage levels.

The quantized steps produced by the DAC have the same baseband frequency response as the signal that was mathematically "sampled" to produce discrete values. However, the effect of sampling in the time domain is repetition in the frequency domain. Each frequency image is separated by the sample rate. This is demonstrated in Figure 7.

To remove these frequency images, the DAC output is applied to reconstruction filters. These low-pass filters are intended to transmit the baseband signal while rejecting the higher frequency images. The ESG's internal dual arbitrary waveform generator allows the user to select among three reconstruction filters (250 kHz, 2.5 MHz, and 8 MHz) or a through path for an external reconstruction filter.

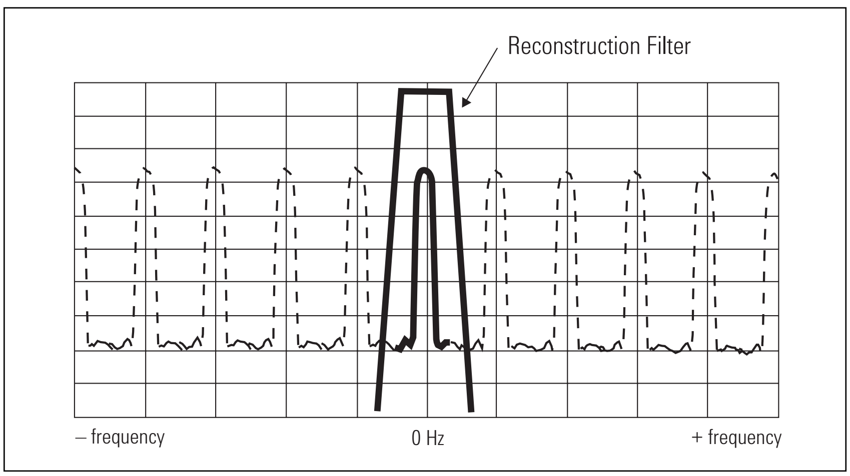


Figure 7. The frequency-domain effect of time-domain sampling

Dual arbitrary waveform generator, continued

Reconstruction filter selection is a function of two variables: signal bandwidth and sample rate. The reconstruction filter must be broad enough to accurately transmit the entire baseband signal, but its cutoff must be low enough to sufficiently reject the first image at the sample rate. Given the available reconstruction filters, care must be taken in the design of a waveform to allow for effective signal reconstruction. This is discussed in more detail in the section on generating waveforms.

The personalities based upon the dual arbitrary waveform generator automatically activate the appropriate reconstruction filter for each waveform they generate.

Triggers and markers

In a test and measurement environment, users might want to synchronize the playback of waveforms in the dual arbitrary waveform generator with external test equipment or trigger other measurements at certain points in the playback of a waveform. The Agilent ESG-D provides the capability to do both.

Triggering waveforms

Users may want to control the playback of waveform segments or sequences so that their timing is synchronized with some external event. Four types of triggers are provided for this purpose: continuous, single, gated, and segment advance. These are described individually below. Triggering can receive its input from the front-panel trigger key, the GPIB bus, or an external TTL or CMOS signal.

Continuous trigger

Continuous triggering is the default triggering mode, and results in a waveform that repeats continuously, triggering to begin every time the waveform completes playback.

Single trigger

Single triggering results in a waveform segment or sequence that plays one time for each trigger signal received. For users who need to have predefined waveforms play at a specific time, single triggering allows them to synchronize these waveforms with external events.

Gated trigger

Gated triggering is the only mode that allows interruption of a segment's playback. In gated triggering, an external trigger is used to control the playback of the segment or sequence. When the external signal is at the "active level," which can be set to "high" or "low," the waveform plays back normally. When the signal moves to the "inactive level," playback is suspended until the signal returns to the active level.

Segment advance

Segment advance triggering is available only when the active waveform is a sequence. When segment advance triggering is active, the dual arbitrary waveform generator will continuously play the current segment of the sequence. When a trigger signal is received the current segment will be played to its end; then the sequencer will advance to the next segment in the sequence.

Dual arbitrary waveform generator, continued

Application example: CDMA frame error rate measurements
CDMA base-station manufacturers perform sensitivity measurements on their receivers by transmitting patterns of CDMA data with error-detecting coding, and calculating a frame error rate (FER). The user can employ the techniques described in the section on “Waveform generation” to generate a reverse traffic channel signal with full coding (long code of 0’s), including interleaving and convolutional encoding. “Single trigger” mode will accept the base station’s “even second” clock to synchronize the transmitted waveform with CDMA system time. The base station will then calculate a FER from the received signal.

Using markers
Waveform markers are signals embedded in dual arbitrary waveform generator signals that can be used to trigger events either externally or internally to the ESG. Their location is defined in a segment during the waveform generation process, or using the marker editor function from the ESG front panel. For the multichannel CDMA personality, Option UN5, the even-second system-synchronization signal is created by a marker in the waveform that is generated by the personality. The section on generating waveforms for the dual arbitrary waveform generator describes how to place markers in externally generated signals.

Markers can be activated or deactivated using the sequence table editor. For each segment or embedded sequence in the table editor, the user can choose to independently activate one of two markers. In addition, the user can select positive or negative marker polarity, or tie a marker to the RF blanking feature of the ESG to simulate bursted TDMA signals.

TDMA bursting
To simulate bursted TDMA signals, such as GSM, marker 1 can be linked to the RF blanking of the ESG. Since the absence of I/Q modulation will result simply in an unmodulated RF carrier, due to small I/Q imbalances in the hardware, this capability provides the best means to actually turn off the RF carrier by using the dual arbitrary waveform generator. An inactive marker will allow the I/Q-modulated signal to be generated normally. However, for those portions of the waveform that should simulate inactive timeslots, with no RF carrier, marker 1 can be set to active, which results in RF blanking.

Markers are effective as synchronization and control signals when using the ESG’s internal dual arbitrary waveform generator. They can only be added to waveforms from the front panel or during the generation process. The discussion follows with a description of this process and with an extensive example that uses MATLAB.

Waveform generation

Basic digital transmitter

As mentioned earlier, the onboard DSP of the dual arbitrary waveform generator has the capability to generate waveforms for the Agilent ESG's many personalities. For those who wish to generate custom waveforms, the ESG with Option UND provides the capability to download waveforms directly to the ARB RAM.

These waveforms can be generated in a variety of ways, including low-level programming languages such as BASIC or C++, general-purpose simulation tools like MATLAB or Agilent VEE, and high-level CAE applications like the Advanced Design System. Virtually any application capable of generating a sequence of numbers can generate waveforms for the ESG.

Since the ESG is often used to simulate all or part of a digital communications transmitter, the following discussion of waveform generation is conducted in the context of a generic digital transmitter.

Figure 8 depicts the block diagram of a basic digital transmitter. The whole chain can be simulated with a properly designed waveform and the ESG with Option UND, the dual arbitrary waveform generator. The I/Q modulation and RF transmission components are performed by the ESG hardware. The blocks preceding I/Q modulation, however, can be simulated externally and downloaded in the form of a sampled waveform to ARB RAM.

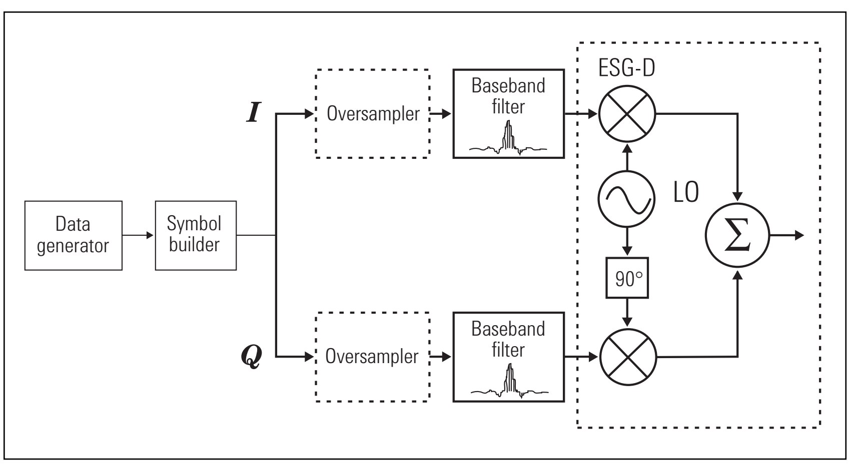


Figure 8. Block diagram of a basic digital transmitter

Data generator

The basic blocks of the digital transmitter are the data generator, the symbol builder and the baseband filter.

For the purposes of this block diagram, data generation includes steps such as data framing, cyclic redundancy check (CRC) encoding, and interleaving. The information that passes to the symbol builder consists of binary data that represents all of the logical manipulations performed before that stage.

An example of a MATLAB M-file that generate data sequences that follow the pattern of a linear feedback shift register can be found in Appendix A, under `lfsr.m`. The following example generates a 511-bit PN9 sequence, repeated twice.

```
>> taps = [1 0 0 0 1 0 0 0 0 1];
>> seed = [1 1 1 1 1 1 1 1 1];
>> data = lfsr(9, taps, 1022, seed);
```

Waveform generation, continued

Symbol builder

The symbol builder in a basic digital transmitter takes the bits produced in the data-generation process, collects them into symbols and creates I and Q waveforms that map the instantaneous or differential phase and magnitude of the modulating signal to these symbols. The output of a symbol builder consists of two waveforms, I and Q.

In the following example, we create a QPSK symbol builder that collects two bits per symbol and maps them to the four quadrants of the I/Q plane. The function of a QPSK symbol builder is illustrated in Figure 9.

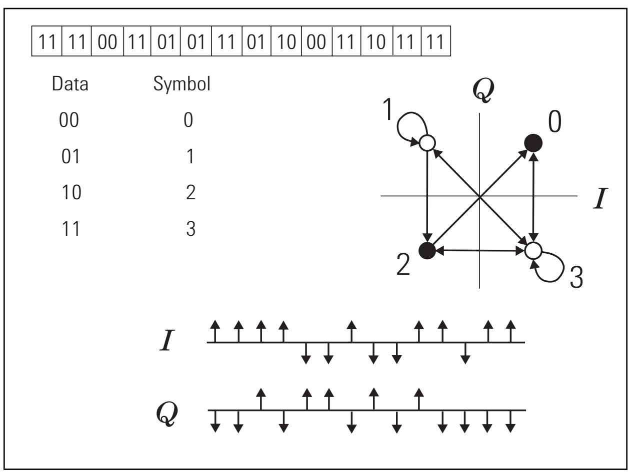


Figure 9. Building QPSK symbols from binary data

The M-file `qpsk.m` in Appendix A demonstrates an implementation of a QPSK symbol builder. The following example generates 511 QPSK symbols from the PN9 data generated above.

```
>> symbols = qpsk(data);
```

Baseband filter

A baseband filter is applied to reduce the transmitted bandwidth, increasing spectral efficiency. For signals generated with digital signal processing, these filters are often finite impulse response (FIR) filters with “taps” that represent the sampled impulse response of the desired filter.

Waveform generation, continued

The following example uses a root Nyquist filter with the impulse response and transfer function as shown in Figure 10. Basic FIR filtering can be accomplished using the mathematical concept of convolution.

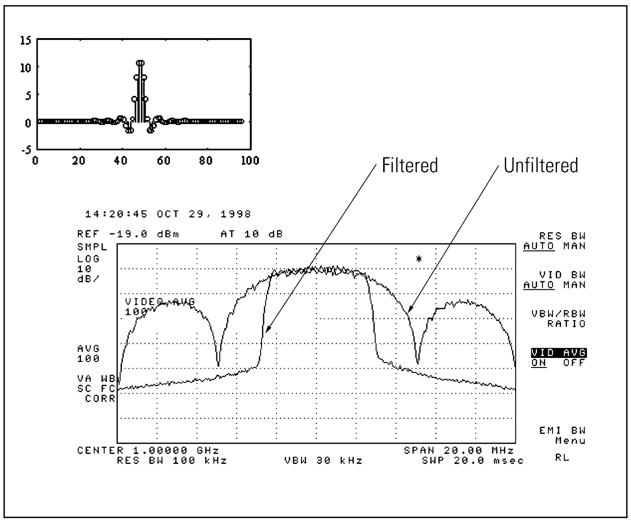


Figure 10. Impulse response and frequency response of root Nyquist filter

Oversampling

Before a FIR filter is applied, some degree of *oversampling* is usually applied to the signal. Oversampling is the process of increasing the number of samples per symbol. The QPSK modulator shown above produces one sample for each symbol (two bits). An oversample ratio of four results in four samples per symbol, and a longer waveform. Oversampling relaxes the requirements for a reconstruction filter in the actual digital-to-analog conversion of the waveform in the dual arbitrary waveform generator.

A discussion of oversampling and how to choose an appropriate oversample ratio that is based on the capabilities of the ESG's internal dual arbitrary waveform generator is included in the following section.

The following MATLAB commands perform specific tasks, using the MATLAB M-files that are shown after the commands:

- Perform 5X oversampling on the I/Q waveform
- Create a 5X oversampled root raised cosine filter with α=0.35
- Perform FIR filtering using convolution

```
>> qpsk5x = oversamp(symbols,5);
>> rtnyq5x = rtnyq(24,5,0.35);
>> qpsk5xfilt = conv(qpsk5x,rtnyq5x);
```

The M-files `oversamp.m` and `rtnyq.m` are listed in Appendix A.

Waveform generation, continued

Hardware constraints

The Agilent ESG with an internal dual arbitrary waveform generator is a powerful simulation tool. Since the user is given access to the most basic elements of digital synthesis, some consideration must be made for the hardware to generate a useful signal.

Basic digital signal processing concepts that relate to sampling and reconstruction must be taken into account. In addition, hardware constraints, such as memory length and maximum sample rate need to be considered when designing a waveform. These criteria result in a trade-off between oversample ratio and waveform length.

Oversample ratio

The oversample ratio of a signal is the ratio of the sample rate to the Nyquist rate of the signal. For most signals, the Nyquist rate is estimated at the symbol rate as discussed below. Increasing the oversample ratio of a signal separates sampling images while maintaining the baseband signal's bandwidth. As the images move further away in frequency, the gap between images broadens, which allows for better rejection.

Figure 11 shows a signal sampled at the Nyquist rate. Two problems arise for this level of oversampling:

1. Since the Nyquist rate in this example is set at the symbol rate, no guardband is allowed for the actual filtered bandwidth.
2. An unrealizable "brick wall" reconstruction filter would be required to accurately transmit the baseband signal while sufficiently attenuating the next sampling image.

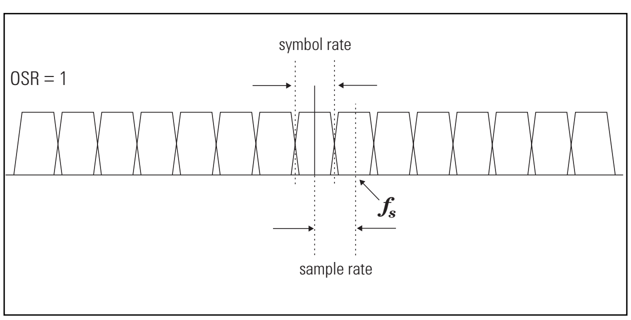


Figure 11. Frequency response of a signal sampled at the Nyquist rate

For these reasons, an oversample ratio (OSR) of four is recommended in most cases. This reduces the chance of "aliasing" (overlap between the baseband signal and sampling images).

Nyquist rate, Nyquist frequency, and symbol rate

The Nyquist sampling theorem is part of digital-signal-processing theory. It states that a sampled signal, band-limited to the *Nyquist frequency*, is uniquely determined by its samples if the sample rate is twice the Nyquist frequency. This sample rate (twice the Nyquist frequency) is referred to as the *Nyquist rate*. For practical (non-bandlimited) signals, a reasonable Nyquist frequency can be determined, beyond which signal power is negligible. Figure 11 depicts a signal sampled at the Nyquist rate in the frequency domain.

For most single-carrier digitally modulated signals, the Nyquist rate is close to the *symbol rate*, which allows for a guardband to account for the rolloff of baseband filtering. For example, the symbol rate of a cdmaOne carrier is 1.2288 MHz. With a guardband the RF bandwidth is 1.25 MHz, equating to 625 kHz at baseband (the Nyquist frequency). A valid Nyquist rate for this signal would be 1.25 MHz.

Due to constraints in reconstruction filters and the convenience of integer oversample ratios, most digital waveform synthesis requires an oversample ratio of at least two. This corresponds to a sample rate that is approximately twice the Nyquist rate, or two times the symbol rate.

Waveform generation, continued

The following steps are helpful in selecting an oversample ratio for waveform generation:

1. Determine the real baseband bandwidth of the signal. This will be approximately one half the symbol rate for most signals. Figure 12 depicts the baseband spectrum of an experimental signal operating at a symbol rate of 500 kHz, with an actual transmitted bandwidth of 600 kHz, or 300 kHz, at baseband.
2. Select a reconstruction filter with sufficient bandwidth to pass the entire baseband signal. The 250-kHz filter shown in Figure 12 has a cutoff that is too low to transmit the entire signal. The 2.5-MHz filter is the best choice for this application.
3. Determine the appropriate oversample ratio (sample rate / symbol rate) for the chosen reconstruction filter. An OSR of six centers the first carrier at 500 kHz * 6 = 3 MHz. At this offset, part of the image is not sufficiently attenuated by the reconstruction filter and causes distortion in the I/Q signal. At a higher OSR of eight, the carrier is centered at 4 MHz. This moves the edge of the image well beyond the filter cutoff point, which minimizes aliasing.

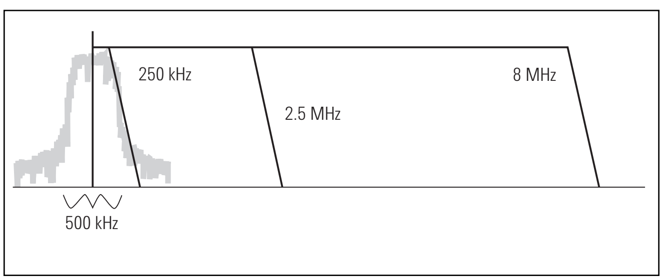


Figure 12. Experimental signal with a 500 kHz symbol rate

In the example above, we selected the minimum reconstruction filter that transmits the in-channel signal, as well as the minimum oversample ratio. The reconstruction filter with the lower cutoff allows for a smaller oversample ratio, and the smaller oversample ratio results in less memory usage for a given length of data in symbols or time.

Waveform length

The number of samples occupied by a given amount of data (symbols or time) is determined by the oversample ratio. The Agilent ESG with Option UND, the dual arbitrary waveform generator, can accommodate up to 1,048,576 samples of data in each channel (I and Q). Signals that require a long-time record of data can occupy all of the available ARB RAM. For example, one frame of IS-95A CDMA data requires 24,560 symbols (chips) of data. This corresponds to 122,800 samples with an OSR of five. Eight such frames can be stored in ARB RAM.

Total sample memory and other constraints on waveform length are summarized below:

- Maximum number of samples: 1,048,576
- Minimum number of samples: 16
- Number of samples must be even
- I and Q samples must be of equal length or Q must be empty

Formatting and downloading data

Once the I and Q waveforms are created in the simulation environment, they must be prepared for use in the dual arbitrary waveform generator, and downloaded to the Agilent ESG.

Dual arbitrary waveform generator binary data format

The waveform data of the dual arbitrary waveform generator is stored in ARB RAM in sixteen-bit integer format. Fourteen bits of each word determine the value of the sample itself. The remaining two bits are used for markers in the I waveform, and are reserved in the Q waveform. The binary storage representation of the data is shown in Figure 13.

Scaling

Since the samples stored in ARB RAM are unsigned fourteen bit integers, the samples created during the simulation of a digital transmitter must be rescaled before they can be downloaded. For I and Q, the possible values are integers in the range from zero to 16,383, with 8,192 corresponding to zero volts after level-shifting on the output board.

The algorithm for proper scaling follows:

1. Calculate the scale factor as follows:

$$\textit{scale factor} = \frac{8191}{\max(|I|_{\max}, |Q|_{\max})}$$

2. Scale and offset all values (I and Q) by:

$$(\textit{scale factor}) \text{ x } \textit{value} + 8192$$

Note: Use a fraction of full scale for better ACP performance.

Each of the methods above is intended to use the full range of the 14-bit DAC while creating an accurate signal. However, driving the I/Q modulator at the maximum level can cause nonlinear distortion in later amplifier stages, causing distortion. This can degrade the usefulness of the ESG for out-of-channel measurements such as adjacent channel power (ACP). To maximize ACP performance, it is sometimes necessary to scale the signal to a fraction of full scale to reduce the drive level of the modulator and subsequent amplifier stages. The ideal fraction of full scale to use is best determined experimentally. As an example, Option UN5, the IS-95A CDMA personality, reduces drive level by approximately 6 dB to optimize ACP performance.

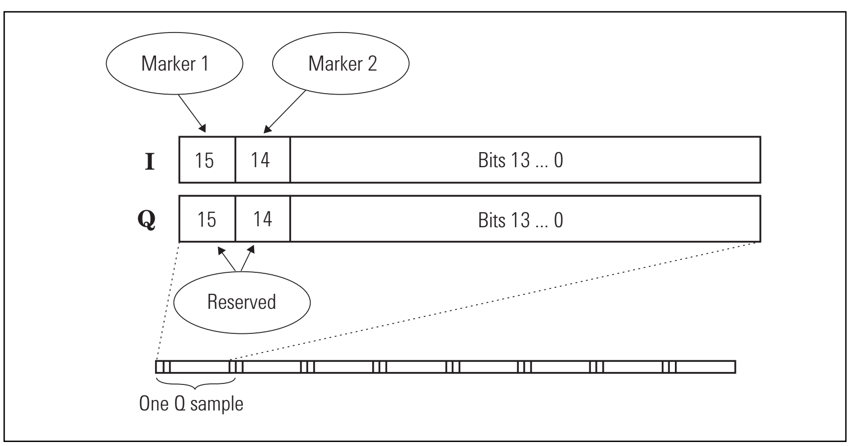


Figure 13. Format of binary data stored in ARB RAM

Markers

Once the data is scaled to 14-bit integers, you can modify the two most significant bits in the I channel to activate markers. As described above, markers can be used for synchronization signals, triggers to external test equipment, or burst control for TDMA timeslots. Marker 1 is determined by bit 15, and marker two by bit 14. Any *I* waveform sample scaled to a 14-bit integer can have a marker added by adding the appropriate "power-of-two" value (2^{15} = 32,768 for marker one or 2^{14} = 16,384 for marker 2).

A sample MATLAB M-file for scaling waveform data and adding markers is demonstrated in `arbsave.m`, in Appendix A. The following exam-ple stores the filtered QPSK waveform generated above to two files (i.bin and q.bin), activates marker one at the first sample of the file, and scales the waveform to 70% of full scale.

```
>> arbsave(qpsk4x,1,0,.7);
```

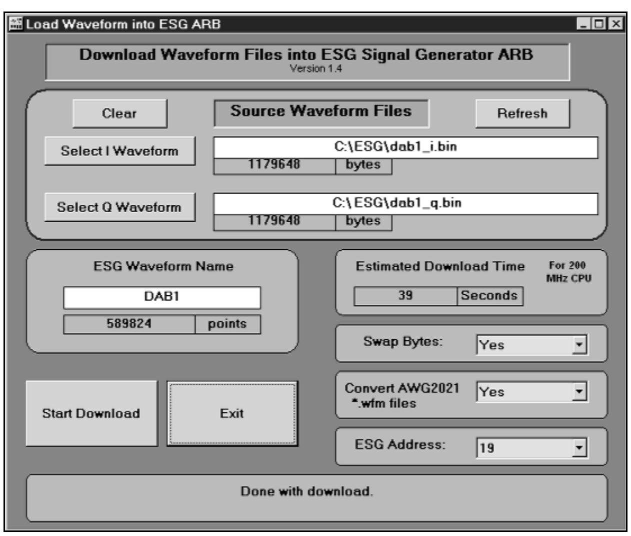


Figure 14. Windows-based download utility for Agilent ESG arbitrary waveform files

Utilities

Agilent has developed some utilities to simplify the process of downloading waveforms to the ESG's internal dual arbitrary waveform generator. These can be downloaded from the ESG website at www.agilent.com/find/esg/. The first utility (shown in Figure 14), which runs in Windows NT®or Windows 95®,[1] loads waveform files stored in 16-bit unsigned integer format and transfers them via GPIB to the ESG's ARB RAM.

This utility requires data files that are stored in the format generated by `arbsave.m`, listed in Appendix A.

1. Windows NT® and Windows 95® are U.S. registered trademarks of Microsoft Corporation.

Agilent also provides a download utility that works directly from the MATLAB command line in Windows NT® and Windows 95®[1] environments. This program scales waveform data and performs the download via GPIB.

More details about these utilities and supported PC hardware configurations can be obtained from the ESG website at www.agilent.com/find/esg/.

Advanced techniques

The process described above allows a user to generate a basic waveform and download it to the ESG's internal dual arbitrary waveform generator. This section describes some advanced techniques to improve waveform performance and create a greater variety of signals.

Phase continuity

Most waveforms generated for the dual arbitrary waveform generator are repeated continuously in the ESG. A discontinuity between the end of a waveform and the beginning of the next repetition can lead to periodic spectral regrowth that distorts measurements. This section discusses the factors that lead to phase discontinuities in waveforms and some methods for avoiding them.

Consider the sinewave shown in Figure 15. Note that this signal is an accurate sinewave in the time period of interest (one waveform length). However, if this waveform is repeated, as is likely to happen in the ESG, a discontinuity is induced at the point where the waveform repeats. The spectrum of the sinewave with a discontinuity shows a dramatic increase in spectral components away from the impulse functions that should represent the spectrum of a sinewave alone. This is one form of phase discontinuity that can be avoided by simulating an integer number of cycles.

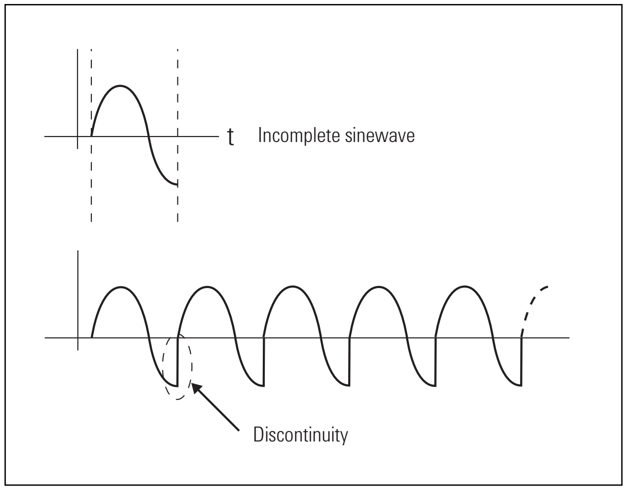


Figure 15. Demonstration of the spectral effect of adding a discontinuity to a sinewave

1. Windows NT® and Windows 95® are U.S. registered trademarks of Microsoft Corporation

Waveform generation, continued

The effects of FIR filtering induce a form of phase discontinuity that is seen often during the simulation of a digital transmitter. The addition of filter delay, which must be removed for proper playback timing, can result in a discontinuity between the beginning and end of the truncated waveform if it is repeated. Refer to Figure 16 for an illustration of this effect. If the waveform will be repeated continuously once downloaded to the ESG, the use of circular convolution will result in a waveform that more realistically simulates a true digital transmitter and eliminates phase discontinuities. Figure 17 demonstrates this technique.

A circular convolution algorithm for creating a continuously filtered signal is demonstrated in the M-file `circfilt.m` in Appendix A. The following example duplicates the `qpsk5x` data generated above to create an even number of samples, then performs circular convolution with the 5X oversampled root Nyquist filter.

```
>> qpsk5x2 = [qpsk5x qpsk5x];
>> qpskfilt = circfilt(qpsk5x2,rtnyq5x);
```

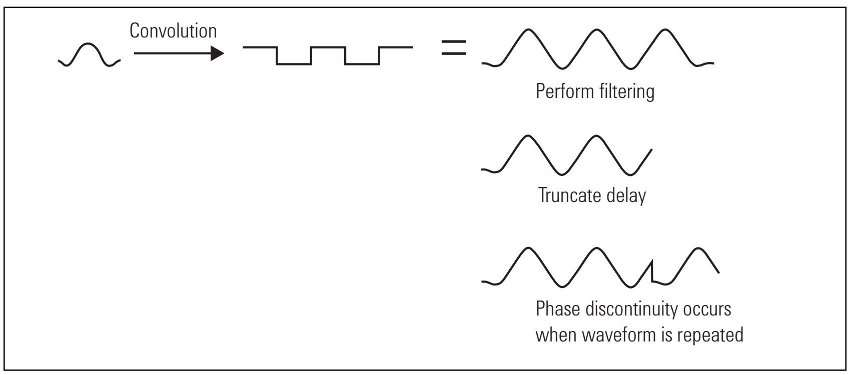


Figure 16. Phase discontinuity generated by truncation after FIR filtering

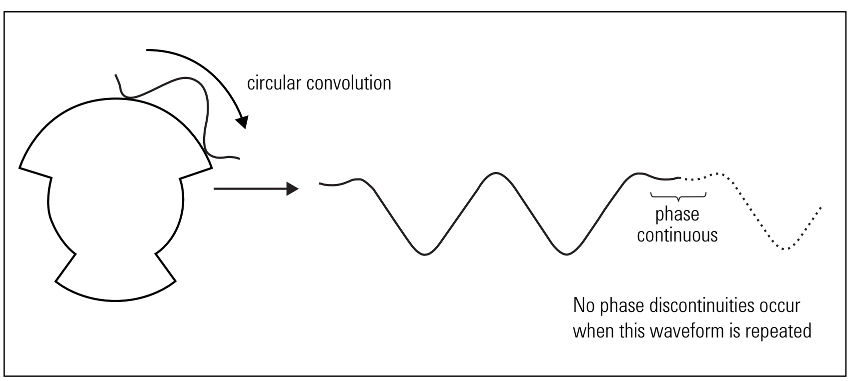


Figure 17. Using circular convolution to eliminate filter delay and phase discontinuities for FIR filtering

Waveform generation, continued

Generating multicarrier signals

The ESG has only one synthesized RF signal that can be modulated as a carrier. However, through baseband frequency translation, multiple RF carriers can be simulated with the dual arbitrary waveform generator.

The basic process for creating multicarrier signals is listed below and illustrated in Figure 18.

Step 1. Generate independent carriers using the techniques mentioned earlier.
Step 2. Translate carriers to relative offsets in frequency.
Step 3. Add translated carriers together for complete multicarrier baseband signal.

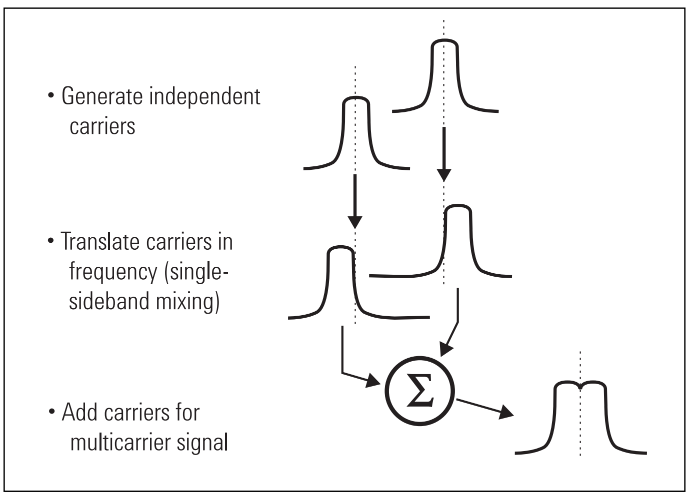


Figure 18. Multicarrier signal generation process

I/Q baseband signals for each carrier should be generated independently from data generation to baseband filtering. Once this is done, the signals can be assigned to separate carriers, which is determined by their offset from a center frequency. At baseband, this center frequency is represented by dc. Carriers that will be lower than the center frequency at RF should be placed at negative frequencies at baseband and those that will be above the center frequency should be at positive frequencies.

Waveform generation, continued

Frequency translation is accomplished by mixing with a complex sinusoid. Mixing with a real sinusoid, such as a cosine, would result in translation of a carrier both up and down in frequency. However, a complex sinusoid (cos x ± i·sin x) performs a one-sided frequency translation due to the phase relationship between sine and cosine in the frequency domain. Figure 19 illustrates this point. This concept is discussed in more detail in Appendix B.

Bandwidth of multicarrier signals
Translating modulated carriers in frequency can quickly transform narrowband single carrier signals into broadband multicarrier signals. This must be taken into account when generating the original baseband signals before they are translated and summed together.

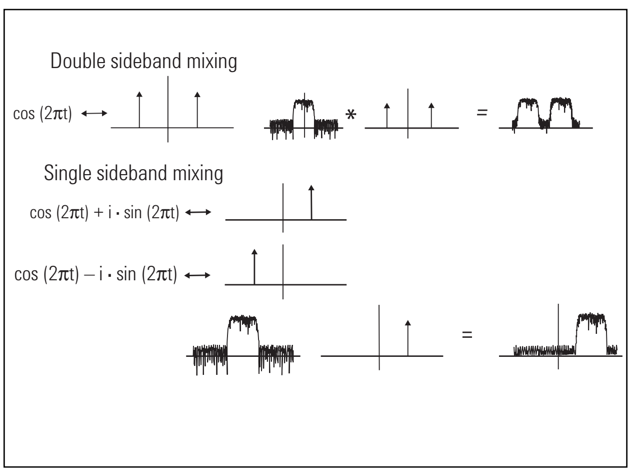


Figure 19. Using a complex sinusoid to achieve one-sided frequency translation

As an example, consider an NADC (IS-136) signal, as shown in Figure 20. While the baseband signal occupies 30 kHz of bandwidth, a multichannel version can occupy an arbitrary bandwidth depending on spacing. A sample rate of 480 kHz, which is fine for single carrier NADC (with an oversample ratio [sf2]of 16), would be too low for a multicarrier signal consisting of 10 adjacent carriers (OSR is 480/300 = 1.6). Therefore, it would be more appropriate to sample each NADC carrier at 1.2 MHz (OSR = 40), for instance, in order to achieve an OSR of four when these carriers are translated and combined into a ten carrier waveform.

The creation of a multicarrier signal can be accomplished using the M-file `cplxmix.m` in Appendix A. The following example creates two carriers with `qpskfilt` on each, offset 625 kHz above and below the set carrier frequency, assuming a sample rate of 6.25 MHz. These signals could easily be entirely different, as long as their sample rates and waveform lengths match.

```
>> multicarrier = cplxmix(qpskfilt,
  -625000, 6250000)
+ cplxmix(qpskfilt, 625000, 6250000);
```

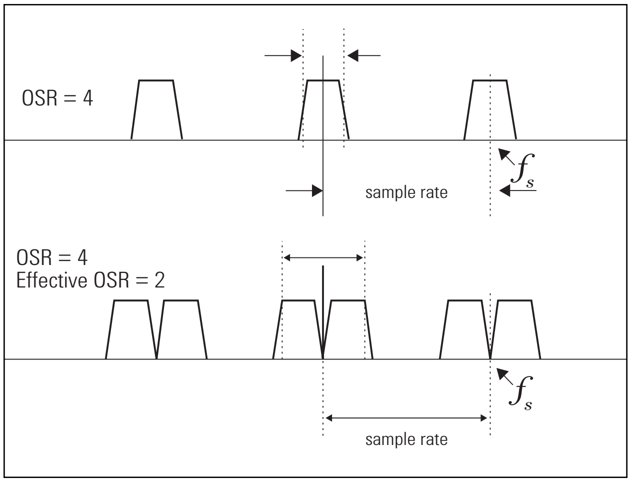


Figure 20. Increased oversampling requirement for multi-carrier signals

Maximizing effective I/Q bandwidth

The recommended minimum OSR is four in most cases. However, for signals with a bandwidth above 10 MHz, a lower OSR is necessary. For a 20 MHz W-CDMA signal, for example, an OSR of two is required. In addition to the additional constraints placed on the reconstruction filter response, an OSR of two requires the waveform designer to accommodate for the “sample-and-hold” output of the DACs.

The output of the 14-bit DACs is a signal for which each sample is output and held for the duration of a sample clock period. In the next sample period the next sample is output and held. The resulting signal looks like a “staircase.”

The sample-and-hold signal is equivalent to the convolution of the ideal impulse-train output with a delayed pulse function with a frequency response (for sample rate f_s) of:

$$\frac{\sin\left[\pi \bullet \frac{f}{f_s}\right]}{\pi \bullet \frac{f}{f_s}}$$

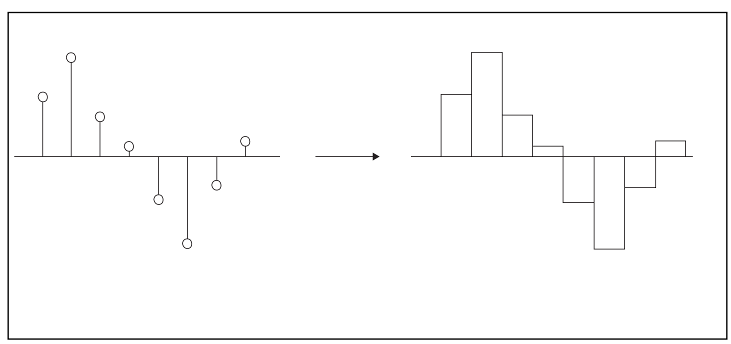


Figure 21. Sample-and-hold DAC output compared to ideal digital synthesis

This response results in a gradual rolloff near the center of the baseband signal that increases dramatically at offsets close to the sample rate. With an oversample ratio of four, the frequency response at the edge of the transmitted bandwidth (offset at ½ the symbol rate) is:

$$H_{dB} = 10^{*}\log\left[\operatorname{sinc}\left[\frac{1}{8}\right]\right] = -0.11 \text{ dB}$$

However, for an oversample ratio of two, this rolloff increases to –0.46 dB, which can significantly degrade the in-channel performance of a digitally modulated signal.

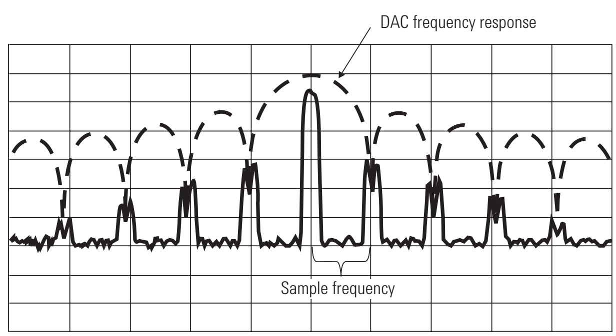


Figure 22. Frequency response induced by sample-and-hold output of DACs

To generate useful signals with an oversample ratio below four, the waveform designer should compensate for this rolloff in simulation by preemphasizing the band edges with the inverse response of the sinc (sin x / x) function. This can be accomplished simply by filtering the baseband signal with a preemphasis function. The appropriate filter has the following frequency response for the passband of the signal:

$$H_{preemph} = \frac{\pi \bullet \dfrac{f}{f_s}}{\sin\left[\pi \bullet \dfrac{f}{f_s}\right]}$$

This inverse sinc function will multiply with the DAC frequency response, resulting in a net response of one across the transmitted signal's bandwidth. Note that with this technique the absolute minimum OSR is two. As the preemphasis function approaches the sample rate, its magnitude approaches infinity as the DAC response approaches zero. Preemphasis with such a large gain will quickly exhaust the dynamic range of the DACs as the edges of the signal become multiple orders of magnitude larger in simulation than the signal's center. To avoid this, define the preemphasis filter only over the passband of the signal.

Waveform generation, continued

$$H\ \text{preemph} = \frac{\pi \cdot \frac{f}{f_s}}{\sin\left[\pi \cdot \frac{f}{f_s}\right]}$$

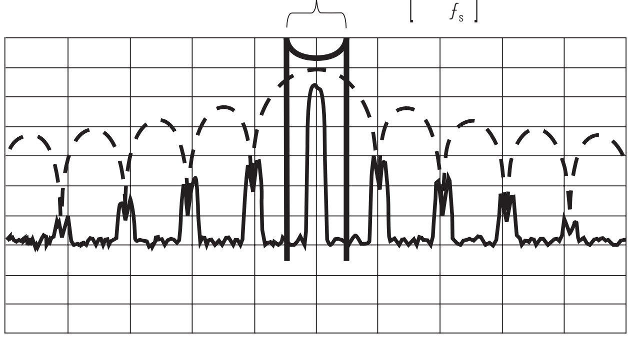


Figure 23. Preemphasis filter used to cancel effects of sample-and-hold DAC rolloff for wide baseband signals

An M-file that applies the correct preemphasis filter to correct for DAC rolloff can be found under `daccorr.m` in Appendix A. The following example preemphasizes `qpskfilt` to account for possible DAC rolloff.

```
>> preem = daccorr(qpskfilt,75,0.7,1);
```

Design in the frequency domain

Up to this point, all waveform examples have been generated in the time domain. The resulting I/Q signals are time domain signals, and all discussion of the frequency domain has been for clarification of the concepts of waveform generation and optimization. Some modulation formats, like orthogonal frequency division multiplexing (OFDM), are easier to design and simulate in the frequency domain.

Waveform generation, continued

Discrete-time frequency
Since the signals developed in simulation are dependent on the sample rate to determine absolute frequency, it is easier to think of their frequency response in the discrete-time frequency domain. The discrete-time frequency representation accounts for the periodicity of discrete-time signals as seen in the images that appear spaced by the sample rate, as described above. The discrete-time frequency domain is limited to the frequencies between zero and the sample rate, normalized to ½ the sample rate.

In this domain, frequencies are relative, and all frequencies are expressed in terms of the sample rate. Frequency components located at points along the discrete-time frequency domain of the frequency axis will translate to that position times the sample rate divided by two when actually synthesized by the dual arbitrary waveform generator. This includes discrete-time frequency components between one and two. If the discrete-time frequency domain is directly translated to continuous time frequency, the upper sideband of the baseband signal is paired with the lower sideband of the first sampling image that is centered at the sample rate. Since discrete-time signals are periodic in the frequency domain, a copy of this lower sideband will also appear as the lower sideband of the actual baseband signal. As a result, the final reconstructed baseband signal has an upper sideband derived from the discrete-time frequency values from zero to one, and a lower sideband corresponding to frequency values between one and two.

Orthogonal frequency-division multiplexing (OFDM)
OFDM is an example of a modulation format that is best constructed in the frequency domain. OFDM, used in digital video broadcast (DVB) often consists of hundreds of carriers, modulated individually. In designing an OFDM signal, each carrier is normally assigned to a fast Fourier transform (FFT) "bin." Each element in the array of data that represents the frequency content of a discrete-time signal is referred to as a bin. These bins can contain real and imaginary components, which allows direct simulation of amplitude- and phase-modulated signals.

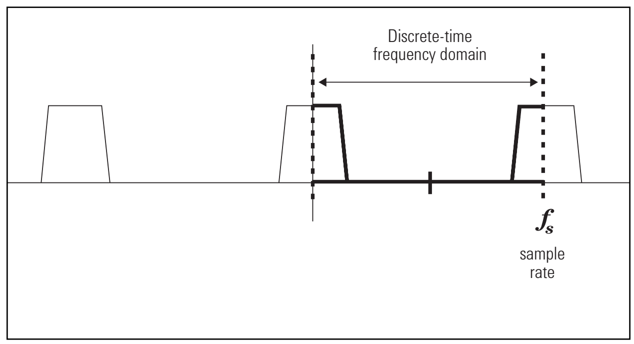


Figure 24. Correspondence between discrete-time frequency and continuous time frequency

Waveform generation, continued

For example, one OFDM scheme might consist of 500 carriers, each modulated with 64 QAM. To simulate a symbol of data on one of these carriers, the waveform designer simply would need to pick real and imaginary components that correspond to the correct amplitude and phase for the data in question. Assigning such symbols to each of 500 carriers in a series of frequency bins results in the instantaneous frequency response of the OFDM signal for one “symbol” period. In the case of 500 carriers with 64 QAM, one symbol represents 6 (bits per symbol) x 500 (symbols) = 3,000 bits of data.

Just as with any other modulation format, OFDM requires a sufficient oversample ratio to allow reconstruction. This can be accomplished in the frequency domain by padding the discrete-time frequency data with zeros. Instead of interlacing the data with zeros as in time domain oversampling, the appropriate method is to insert a block of zeros between the upper and lower sidebands of the signal. This effectively moves the sample rate to a higher level relative to the baseband bandwidth of the OFDM signal, which is the definition of oversampling.

Once this signal is constructed in the frequency domain it must be transformed to a time-domain signal that can be applied to an I/Q modulator in the ESG-D. This is accomplished with an IFFT. Since the resulting data represents only one “snapshot” of data on the 500 carriers that are modulated, this process must be repeated for each subsequent block of data that must be transmitted. Each block of data translates to an OFDM symbol with a number of time samples equal to the number of FFT “bins” or points in the frequency domain. For 4X oversampling, and 500 carriers, this would result in 4 x 500 = 2,000 samples.

In many implementations of OFDM, “guard data” is inserted between such blocks of data to avoid interference between one symbol and the next. Data can be copied from the beginning of a block and appended in the time domain before the next block is generated.

Since OFDM does not normally require baseband filtering, the data assembled as described above is ready to be downloaded to the dual arbitrary waveform generator’s RAM and played back at the proper sample rate.

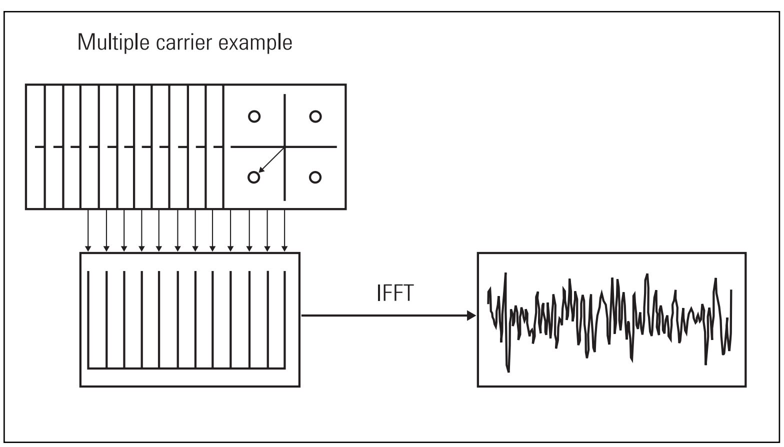


Figure 25. Constructing an OFDM signal

Resources

This product note outlines the basic steps required to generate and download waveforms to the Agilent ESG-D's internal dual arbitrary waveform generator. It provides a foundation on which an experienced designer can apply techniques to generate real-world signals from simulation. The following section outlines Agilent's commitment to supporting customers who want assistance in waveform generation or would like more information on the general topics of digital signal processing and digital modulation.

Agilent-provided applications support

Several levels of ESG assistance are available.

Agilent provides the following services, included in the price of the instrument. See the ESG website at www.agilent.com/find/esg/ (under "Related Info") or contact your sales office for more information.

- Assistance in downloading waveform files to the ESG-D via RS-232 or GPIB using utilities provided by Agilent, or example code published in Agilent literature.
- Assistance in using waveform files distributed via the ESG website, or as examples in Agilent literature.
- Assistance using sample code (including MATLAB M-files and programming examples) distributed via the customer website, or as examples in Agilent literature.
- Assistance using general ESG-D features, including those of the dual arbitrary waveform generator.

Agilent can provide additional professional services for a fee. Some examples are listed below.

- Waveform generation services or consultation.
- Download support using tools or utilities other than those provided (in the form of a software program or programming example) by Agilent.
- Test integration of the ESG-D with other test equipment.

Agilent ESG website: www.agilent.com/find/esg

For more information about the Agilent ESG family of RF signal generators, including Option UND, the dual arbitrary waveform generator, please consult the ESG website, www.agilent.com/find/esg/. In addition to general product information and links to related literature, the following Option UND-specific resources are available.

Downloadable waveforms, MATLAB examples, and utilities

Waveforms can be downloaded to the ESG-D for common communications standards or basic signal-generator tests using the utility described above.

The MATLAB example M-files used in this product note can be downloaded from this website.

The Windows- and MATLAB-based download utilities can also be downloaded from this website.

Resources, continued

References

The following references can provide the reader with more information on digital signal processing, digital communications and measurement issues for RF digital communications systems.

Agilent Literature

Agilent provides a wide selection of application and product notes about RF and microwave measurement techniques and technologies. These are all available through your Agilent sales office, or from the Agilent ESG web page, www.agilent.com/find/esg/

Using MATLAB

For a complete list of references using MATLAB examples for signal processing, please consult the MathWorks website: www.mathworks.com

Digital communications

Cellular Radio Systems, ed. Balston and Macario, Artech House 1993

Kamilo Feher, Wireless Digital Communications, Prentice-Hall, 1995

Garg and Wilkes, Wireless and Personal Communications Systems, IEEE Press, 1996

Jerry Gibson, The Communications Handbook, IEEE Press, 1997

Jerry Gibson, The Mobile Communications Handbook, IEEE Press, 1996

Simon Haykin, Digital Communications, Wiley 1988

Harry Young, Wireless Basics 2nd Edition, Telephony Books, 1996

Raymond Macario, Cellular Radio Principles and Design, McGraw-Hill 1993

Madisetti and Williams, The Digital Signal Processing Handbook, IEEE Press, 1998

Asha Mehrotra, Cellular Radio Analog and Digital Systems, Artech House 1994

Rappaport, Wireless Communications, Principles & Practices, Prentice-Hall, 1996

Reed, Rappaport and Woerner, Wireless Personal Communications, Klewar, 1997

Bernard Sklar, Digital Communications, Fundamentals and Applications, Prentice-Hall, 1988

Resources, continued

Digital signal processing

Abraham and Baldwin, et al, Programs for Digital Signal Processing, IEEE Press, 1979

Douglas Elliot, Handbook of Digital Signal Processing Engineering Applications, Academic Press, 1987

Lonnie Lundeman, Fundamentals of Digital Signal Processing, Harper & Row, 1986

Marven and Ewers, A Simple Approach to Digital Signal Processing, Wiley, 1996

Morgera and Krishna, Digial Signal Processing Applications to Communications and Algebraic Coding Theories, Academic Press, 1989

Oppenhiem and Willsky, Signals and Systems, Prentice-Hall, 1983

Oppenheim and Shafer, Digital Signal Processing, Prentice-Hall, 1975

Alan Oppenheim, Applications of Digital Signal Processing, Prentice-Hall, 1978

Sophocles Orphandis, Introduction to Signal Processing, Prentice-Hall, 1996

Ifeachor and Jervis, Digital Signal Processing, A Practical Approach, Addison-Wesley 1993.

Proakis and Manolakis, Introduction to Digital Signal Processing, Macmillan 1988.

Westall and Ip, Digital Signal Processing in Telecommunications, Chapman & Hall, 1993

Widrow & Stearns, Adaptive Signal Processing, Prentice-Hall, 1985

William Stanley, Digital Signal Processing, Reston 1975

Ziemer and Trantner, Principles of Communications, Systems, Modulation, and Noise, Fourth Edition, John Wiley & Sons, 1995

Appendix A, MATLAB M-files

arbsave.m

```
function arbsave(v,mkr1,mkr2,scale)
%    arbsave(v,mkr1,mkr2,scale)
%
%    Converts the vector v into I and Q. Scales these
%    two vectors into integers lying between 0 and
%    +16383 for 14 bit dac values.
%
%    Activates markers 1 and 2, based on mkr1 and mkr2
     states.
%
%    Scales data to maximum range by 'scale'
%
%    After conversion, the I values are stored in i.bin, %      and the
Q values are stored in q.bin.
%

i = real(v);
q = imag(v);

mx = max([max(abs(i)) max(abs(q))]);
scaleint = round(8192*scale)-1;
i = i/mx*scaleint + 8191;     % Make 14 bit unsigned
     integers
q = q/mx*scaleint + 8191;
i = round(i);
q = round(q);

i = min(i,16383);           % Just to be safe
i = max(i,0);
q = min(q,16383);
q = max(q,0);

i(1)=i(1)+mkr1*16384+mkr2*32768; % Set markers to begin
segment

fid = fopen('i.bin','w');
num = fwrite(fid,i,'unsigned short');
fclose(fid);
fid = fopen('q.bin','w');
num = fwrite(fid,q,'unsigned short');
fclose(fid);
```

cplxmix.m

```
     function mixed = cplxmix(bbsignal, fmix, fs)

% mixed = cplxmix(bbsignal, fmix, fs)
%
%    Mixes (with complex mixing) a signal up or down by a specific
     frequency.
%
%    'bbsignal' is the complex signal you wish to translate in
     frequency.
%    'fmix' is the LO frequency for      mixing (with +/- corresponding
to mixing up
%     or down in frequency, respectively)
%    'fs' is the sample frequency for 'bbsignal' and 'mixed'
%    Note: fmix + the baseband bandwidth MUST BE < fs / 2 !!
%    'mixed' is the output IF signal

updn = sign(fmix);

% Calculate "integer cycles" shifted mixing frequency
Nfrac = length(bbsignal)*abs(fmix)/fs;
N = round(Nfrac);
fmixmod = N*fs/length(bbsignal)

% Calculate discrete-time frequency equivalent
nyqratio = fmixmod / fs;
digfreq = nyqratio * 2 * pi;

% Create mixing signal
t = 1:length(bbsignal);
mixsig = cos(digfreq*t) + updn*i*sin(digfreq*t);

%  Mix
mixed = mixsig .* bbsignal;
```

daccorr.m

```
function y = daccorr(x,N,bw,samprate)

% y = daccorr(x,N,bw,samprate)
%
% Returns 'y' which is 'x' corrected for the rolloff response of a
% sample-and-hold DAC.  The transfer function of the filter is
%  1/sinc in the passband, preemphasizing the frequencies subject to
%  attenuation.  The maximum passband is at 1/4 the Nyquist rate,
% corresponding to a minimum OSR of 2.
%
% N is the order of the equalizing filter and should be an odd
  integer
%.
%
```

```
% 'bw' is the bandwidth of the passband, including a
     guardband for
% baseband filter rolloff.  For example, for a signal with a
1.2288 MHz
% symbol rate, a good bandwidth might be 1.5 MHz.  A minimal
     bandwidth
% is desirable to reduce the dynamic range requirements of
     the inverse
% sinc filter.
%
% 'samprate' is the sample rate that will be used with the
     signal being
% corrected.
%
% NOTE: bw / samprate must be less than 0.9.

bandedge = bw/samprate;
h = cremez(N, [-1 -.9 -bandedge bandedge .9 1], {'invsinc',
     .5});
y = filtcont(x,h);
```

circfilt.m

```
     function y=circfilt(x,h)

% y = circfilt(x,h)
%
% Uses convolution to filter signal 'x' with filter 'h.'
Removes
% filter-induced delay and eliminates the phase discontinuity
% problem that arises when the signal "wraps" to repeat in a            dual
% arbitrary waveform generator.
%
% 'x' must be larger than 'h' and 'h' must have an even
     number of
% taps.

% Replicate data
hlength=length(h);
datalong=zeros(1,length(x) + hlength);
front=x(1:hlength);
datalong=[x front];

% Filter...
y=conv(datalong, h);

% Shed copied data and added convolution samples
 delay=round(hlength/2) - 1;
y(1:delay)=[];
y(1:delay + 1)=y(length(x)+1 :length(x)+ delay + 1);
y(length(x)+1:length(y))=[];
```

lfsr.m

```
function a = lfsr(n,taps,m,seed)

%  This function generates a maximal length sequence which
%  matches that from a linear feedback shift register.
%
%    Call:    a = lfsr(n,taps,m,seed)
%
%    Where:   a   is the returned array
%                 size is m rows by n columns
%                (see m below)
%                rows -> successive states
%                columns -> individual
%              register states
%              bit output is the last
%               column: a(:,n)
%         n     is the number of stages
           taps  is a row vector of length n+1
%                  showing tap locations as a
%                  binary sequence. The
%                  leftmost element is the
%             coefficient of D^n;
%                   the rightmost element is the
%                coefficient of D^0.
%
%              e.g.  for n=12:
%                P = D^12 + D^10 + D^9 + 1
%                taps = [1 0 1 1 0 0 0 0 0
%                0 0 0 1]
%
%               Note that the first and last
%             elements must be 1.
%
%         m       is the number of records to
%                 generate. If this is not
%                  included in the call, the
%                    number of records assumes a
%                 maximal
%                   sequence (2^n-1).  Long
%                  execution times occur
%                    for n>16 maximal sequences.
%
```

```
%               seed  is an optional parameter.
%                     The default is
%                     1. If this is used, it
%                   should contain the
%                      initial states of each of the
%                      n stages in a row vector
%                      of length n. The rightmost
%                      element is the first to be
%                     shifted out of the lfsr.
%

%
if exist('m') < .5               % If m wasn't passed in
  m = 2^n-1;
end

a = zeros(1,2^n-1);               % Initialize the vector

if exist('seed')
  if length(seed)~=n
    error('The seed must be of length n.')
  else
    a(1:n) = seed;
  end
else
  a(n) = 1;                     % Arbitrary seed
end

if length(taps) ~= n+1
  error('The taps vector must be of length n+1.')
  return
end
taps = taps(1:n);               % Drop the first element

for i=(n+1):m
  a(i) = rem(sum(taps.*a(i-n:i-1)),2);
end
```

oversamp.m

```
   function x = oversamp(signal, ratio)

% x = oversamp(signal, ratio)
%
%  Oversamples 'signal' by 'ratio', using "zero-stuffing".
%
%  e.g.
%   oversamp([1 -1 -1 -1 1 1], 2) = [1 0 -1 0 -1 0 -1 0
     1 0 1 0]

% Interpolate with zeros
pad=zeros(ratio - 1,length(signal));
x=[signal; pad];
x=x(:).';
```

qpsk.m

```
function IQdata = qpsk(data)

% IQdata = qpsk(data)
%
% Creates IQdata, a complex signal with 1X OSR QPSK symbols
%  with the following mapping:
%
% Data   I  Q
%  ——————
%  00    +1 +1
%  01    -1 +1
%  10    -1 -1
%  11    +1 -1

%       I Q
%       —— -
IQmap = [
        +1  +1
        -1  +1
        -1  -1
        +1  -1
        ];

nsyms = length(data) / 2; % Symbol count is half the bit
     count

tempdata = reshape(data,2,nsyms); % Columns are symbols denoted
                        by % two-bit pairs
syms = zeros(1,nsyms);              % Initialize symbol index
     array
IQdata = zeros(1,nsyms);         % Initialize output array

syms = 2*tempdata(1,:) + tempdata(2,:);  % Map binary data
to symbols

syms = syms + 1; % to get the index right
map = IQmap(:,1) + i.*IQmap(:,2);
IQdata = map(syms);
IQdata = reshape(IQdata,1,length(syms));
```

rtnyq.m

```
     function [taps,time] = rtnyq(nsyms, osr, alpha)

% taps = rtnyq(nsyms, osr, alpha)
%
% Generates root Nyquist filter withs ‘nsyms’ symbols.
% ‘osr’ is the oversampling ratio (i.e. samples per symbol)
% ‘alpha’ is the filter alpha characteristic
%

ntaps = nsyms*osr

% (C code) time=(array_index-array_size/2+0.5)/osr;

time = linspace((-ntaps/2+.5)/osr,(ntaps-1-ntaps/2+.5)/osr,ntaps);
% Insert “bad_time” stuff?
taps = 10*((4*alpha)/pi)*(cos((pi*time)*(1+alpha))+(sin(
(pi*time)*(1-alpha)))./(4*alpha*time))./(1-(4*alpha*time).^2);
```

Appendix B, Complex mixing

The equation for the mixing signal in one-sided frequency translation is cos $\omega t \pm i \cdot sin\omega t$, where ω is the frequency offset in radians. Since this is dependent on the sample rate when the waveform is activated in the dual arbitrary waveform generator, this should be treated as a discrete-time frequency, described later in this document. For a known sample rate, f_s, and frequency offset f_{offset}, the mixing signal $m(n)$ is calculated as:

$$m(n) = \cos\left[\frac{|f_{offset}|}{f_s} \bullet 2\pi \bullet n\right] + \operatorname{sgn}(f_{offset}) \cdot i \cdot \sin\left[\frac{|f_{offset}|}{f_s} \bullet 2\pi \bullet n\right]$$

The imaginary sine term is positive for positive frequency translation and negative for negative frequency translation. Point-by-point multiplication of this complex sinusoid by the carrier to be translated will result in a signal translated to the specified frequency offset.

Phase continuity is also an issue for frequency translation when mixing with a complex sinusoid. A discontinuity in the translating signal from end to beginning will cause the same effects as those described above. Therefore, it is necessary to use a complex sinusoid with an integer number of cycles. The easiest way to accomplish this is to alter the mixing frequency slightly to obtain a frequency that completes an integer number of cycles with the number of samples in the original waveform. The equation for calculating the adjusted frequency f_{mod} for a signal with l samples is:

$$f_{mod} = \operatorname{round}\left[l \bullet \frac{f_{offset}}{f_s}\right] \bullet \frac{f_s}{l}$$

This will result in a maximum frequency error $f_{err\text{-max}}$ relative to the intended frequency offset, calculated as:

$$|f_{err\text{-}\max}| = \frac{f_s}{2 \cdot l}$$

This error can be minimized by increasing waveform length of lowering the oversample ratio to decrease the sample rate.

Appendix C, Related literature

Agilent ESG Family of RF Signal Generators, Data Sheet, literature number 5965-3096E

IntuiLink Software, Data Sheet, literature number 5980-3115EN

Agilent ESG Family of RF Signal Generators, Configuration Guide, literature number 5965-4973E

Generating and Downloading data to the ESG-D RF Signal Generator for Digital Modulation, Product Note, literature number 5966-1010E

Customize Digital Modulation with ESG-D Series Real-Time IQ Baseband Generator, Option UND, Product Note, literature number 5966-4096E

Multi-channel CDMA Personality for Component Test, Option UN5, Product Note, literature number 5968-2981E

Using the ESG-D Series of RF Signal Generators and the 8922 GSM Test Set for GSM Applications, Product Note, literature number 5965-7158E

Generating Digital Modulation with the ESG-D Series Dual Arbitrary Waveform Generator, Option UND, Product Note, literature number 5966-4097E

Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is available for at least five years beyond the production life of the product. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage."

Our Promise

"Our Promise" means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, at no extra cost upon request. Many self-help tools are available.

Your Advantage

"Your Advantage" means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

By internet, phone, or fax, get assistance with all your test and measurement needs.

Online Assistance

www.agilent.com/find/assist

Phone or Fax

United States:
(tel) 1 800 452 4844

Canada:
(tel) 1 877 894 4414
(fax) (905) 282 6495

Europe:
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Japan:
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Latin America:
(tel) (305) 269 7500
(fax) (305) 269 7599

Australia:
(tel) 1 800 629 485
(fax) (61 3) 9210 5947

New Zealand:
(tel) 0 800 738 378
(fax) (64 4) 495 8950

Asia Pacific:
(tel) (852) 3197 7777
(fax) (852) 2506 9284

Product specifications and descriptions in this document subject to change without notice.

Printed in U.S.A. March 6, 2002
5966-4097E

Agilent Technologies