CompuScope Digitizer
HANDBOOK

Revision 5.08 – August 2024

Copyright © 2024

All Rights Reserved

Gage Contact Information	
Toll Free:	1-800-567-GAGE
Tel:	1-514-633-7447
Sales Email:	sales@gage-applied.com
Support Email:	tech-support@gage-applied.com
Web:	http://www.gage-applied.com

Gage is a Product Brand of:

Vitrek, LLC
900 North State Street
Lockport, Illinois 60441-2200
USA

Toll Free: 1-800-DATA-NOW
Tel: 1-815-838-0005

http://www.vitrek.com

Copyright © 2024 Vitrek, LLC. All Rights Reserved, including those to reproduce this publication or parts thereof in any form without permission in writing from Vitrek, LLC.

COMPUSCOPE, GAGESCOPE, AND COMPUGEN are trademarks or registered trademarks of Vitrek, LLC.

C#, Visual C/C++, .NET, Visual Basic, Microsoft Windows are trademarks or registered trademarks of Microsoft Corporation. LabVIEW, LabWindows/CVI are trademarks or registered trademarks of National Instruments Corporation. MATLAB is a registered trademark of The MathWorks Inc. Delphi is a trademark or registered trademark of Borland Software Corporation.

Other company and product names mentioned herein may be trademarks or trade names of their respective owners.

Changes are periodically made to the information herein; these changes will be incorporated into new editions of the publication. Vitrek, LLC may make improvements and/or changes in the products described in this publication at any time.

How to reach Gage Product Support
Toll-free phone: (800) 567-4243

To reach Gage Product Support from outside North America
Tel: +1-514-633-7447

E-mail: prodinfo@gage-applied.com
Web site: http://www.gage-applied.com
On-line Support Request Form: http://www.gage-applied.com/support/support-form.php

Table of Contents

INTRODUCTION 3

1 | COMPUSCOPE DIGITIZER OVERVIEW 3

1.1 Analog Oscilloscopes 3
1.2 DSOs & Digitizers 4

2 | COMPUSCOPE CHANNELS 7

2.1 CompuScope Modes 7
2.2 Channels and ADCs 7
2.3 Ping-ponging 8
2.4 Channel Enumeration 8

3 | COMPUSCOPE SAMPLING 9

4 COMPUSCOPE VERTICAL RESOLUTION 11

5 COMPUSCOPE ON-BOARD MEMORY 12

6 | COMPUSCOPE SIGNAL CONDITIONING 13

6.1 Input Analog Bandwidth 13
6.2 Input termination 14
6.3 Input Ranges 14
6.4 CompuScope Channel Auto-calibration 14
6.5 CompuScope Factory Calibration 15
6.6 User DC Offset 15
6.7 Input Coupling 16

7 | COMPUSCOPE TRIGGERING 17

7.1 Trigger Settings: Source, Level and Slope 17
7.2 Trigger Hysteresis 18
7.3 Internal Versus External Triggering 18
7.4 Triggered CompuScopes Acquisition 19
7.5 Segment Size, Pre-Trigger Data and Post-Trigger Data 20
7.6 Trigger Timeout 21
7.7 Trigger Delay and Trigger HoldOff 21
7.8 Complex Triggering 21

8 | COMPUSCOPE MEMORY MODE ACQUISITION 23

8.1 Single Record Acquisition 23
8.2 Multiple Record Acquisition 24
8.3 Time Stamping 24

9 | COMPUSCOPE AUXILIARY CONNECTIONS 26

9.1 Trigger In (External Trigger) 26
9.2 Clock In 26
9.3 Synchronous Triggering 28
9.4 Trigger Out 29
9.5 Clock Out 30

10 | COMPUSCOPE SOFTWARE 31

11 | MULTI-COMPUSCOPE SYSTEMS 33

12 | COMPUSCOPE EXPERT FIRMWARE 35

13 | COMPUSCOPE STREAMING MODE ACQUISITION 37
14 | COMPUSCOPE DSP OPERATION 41
15 | OTHER GAGE INSTRUMENTS 42

Introduction

This manual provides an extensive overview of all Gage CompuScopes digitizer functionality and is the best starting point for new CompuScope users. All operational details, such as software installation methods, safety precautions and detailed instructions upon using functionality are discussed elsewhere. Generally, only functionality that is common to all CompuScope models is presented in this manual.

1 | CompuScope Digitizer Overview

1.1 Analog Oscilloscopes

Analog oscilloscopes were first developed in the early 20th century. An example is shown in Figure 1. Oscilloscopes display analog signals by writing them with a scanning electron beam on a phosphor screen, which serves as the oscilloscope's main display. The persistence of light emitted by the phosphor allows the display screen to act as a short-term analog storage medium for acquired waveforms.

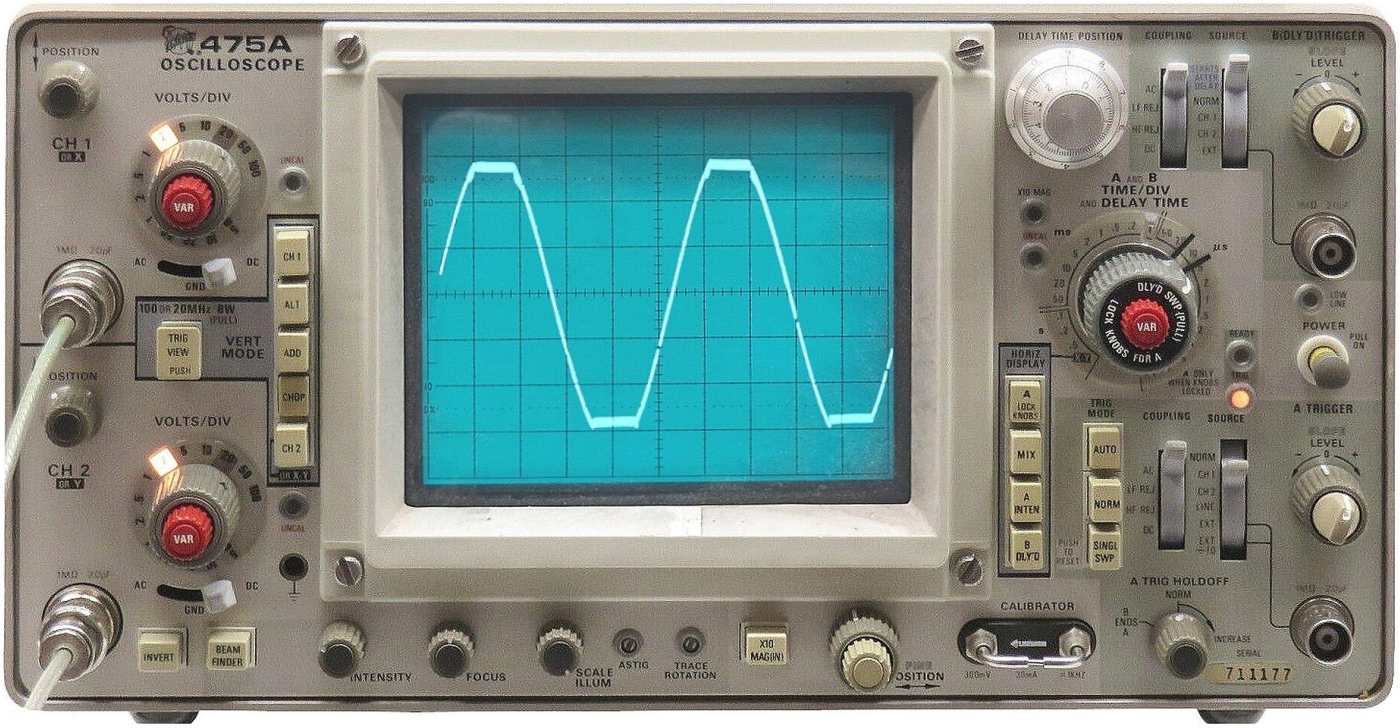


Figure 1: An analog Tektronix Oscilloscope from the 1970s [Wikipedia]

Because of the rapid decay of the phosphor image, only signals that repeat at high frequency are easily viewed on an analog oscilloscope. Triggered to display at the same position within each waveform, the repeating signal rapidly overwrites the signal image on the screen.

Analog oscilloscopes rely upon the repetitive nature of many electrical signals to continuously refresh their displays. Accordingly, it is not easy to view single-shot signals that only occur once on analog oscilloscopes. For this purpose, inconvenient options existed to attach cameras to oscilloscope display screens so that only a single occurrence of a signal could be photographed.

1.2 DSOs & Digitizers

The late 20th century saw the advent of the *Digital Storage Oscilloscope* (DSO). DSOs are equipped with *Analog-to-Digital Conversion* (ADC) chips that transform analog input signals into digital data that may be stored indefinitely within digital memory – in contrast to the temporary storage of an analog oscilloscope's phosphor screen. The inset of Figure 2 shows an ADC chip with its analog input signal and its digital output. The clocking signal is a square wave, whose rising edges determine when a new digital sample is taken by the ADC.

Figure 2 shows a plot of an analog signal input to an ADC chip together with the resulting output signal. The plot illustrates that digitization by the ADC misrepresents the signal in both horizontal and vertical dimensions – time and Voltage – by forcing the digitized samples to reside on the 2D grid indicated. This grid spacing may be decreased in the horizontal and vertical dimensions by increasing the ADC's sampling rate and its vertical resolution, respectively.

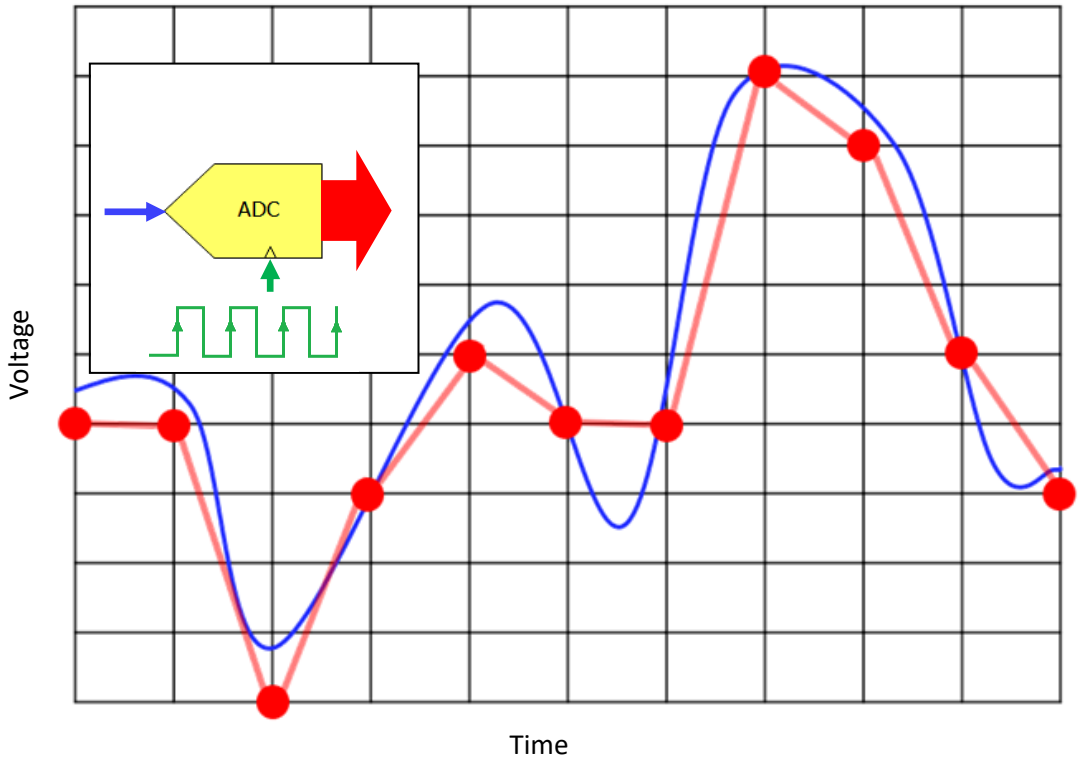


Figure 2: Illustration of digitized version (red) of an analog signal (blue). The inset shows an ADC chip with input, output and clocking signals shown on the left, right and bottom of the ADC chip, respectively.

In addition to allowing acquisition of one-shot signals without elaborate camera equipment, DSOs are often equipped with digital interface cable connections, such as GPIB. This connection allows waveform data digitized by the DSO to be transferred to a remote computer for display, analysis, or printing – a novel feature.

One definition of a digitizer is simply a DSO without a display and with a cabled interface connection to a separate computer. Indeed, such dedicated box digitizers became available shortly after the introduction of DSOs. The 1980s saw the emergence of low-cost Personal Computers (PCs) that contained dedicated slots for add-on circuit cards. These slots were adopted as a platform for PC-based digitizers, such as Gage CompuScopes.



Figure 3: A Gage CompuScope PC-based digitizer

While both the DSO and digitizer use the same core electrical device – the ADC chip - they have evolved to be optimized for different applications. DSOs are ideal instruments for probing unknown signals on an electrical circuit. Accordingly, DSOs are optimized for signal visualization are equipped with a fast display, the ability to acquire signals with a wide range of signal amplitudes, powerful voltage probe technology and elaborate triggering methods.

By contrast, a digitizer is optimized for usage within *Automated Test Equipment (ATE)*, wherein the broad nature of probe signals is understood. In an automated fashion, the ATE monitors small changes in signals over time, position, or some other independent variable. While lower 8-bit vertical resolution is usually sufficient for visualization within a DSO, signals from a digitizer are usually analyzed programmatically and so profit from usage of ADCs with higher vertical resolution. Similarly, digitizer waveform memory is often much larger than the memory available on a DSO.

There are two fundamental advantages of PC-based digitizers that cannot be matched by DSOs. Firstly, while modern DSOs are often housed within a customized PC, it is not an open PC. For example, a user cannot install a separate instrument card within a DSO. A PC-based digitizer, however, is installed within an open PC into which multiple other add-on cards may be installed. In this way, the user can construct an elegant and compact ATE

system all within a single host PC. Secondly, because a PC-based digitizer resides on a PC add-on card bus (e.g. PCI) within the PC, it is in direct and intimate contact with the host CPU, which is only a few inches away. By contrast, cabled communication between a PC and a remote DSO is orders of magnitude slower. The faster data throughput capability of the PC-based digitizer directly enables it to attain far faster repetitive waveform acquisition rates than a DSO.

Since the late 1980s, the *Gage CompuScope* product line has provided high-performance PC-based digitizers. Gage CompuScopes are designed to be easily installed within the add-on slots of almost any commercial PC. Different CompuScope models generally operate identically and vary principally in their three banner specifications, which are: *channel count* (the number of input channels), *maximum sampling rate* and *vertical resolution* (in bits). Models may also vary in other respects, such as their input analog bandwidth or selection of multiple input ranges. Specific information on individual CompuScope models is available elsewhere. This manual introduces core concepts necessary for understanding CompuScope operation.

2 | CompuScope Channels

2.1 CompuScope Modes

The most important attribute of a CompuScope is its *channel count* or number of input channels, which is always a power of two (i.e. 1 channel, 2 channels, 4 Channels...). While each input channel can accept a different input signal, there is one single acquisition engine for all channels on a CompuScope. Accordingly, all channels on a CompuScope must always be sampled and triggered together. Users can only initiate different acquisition sequences on separate and independent CompuScopes.

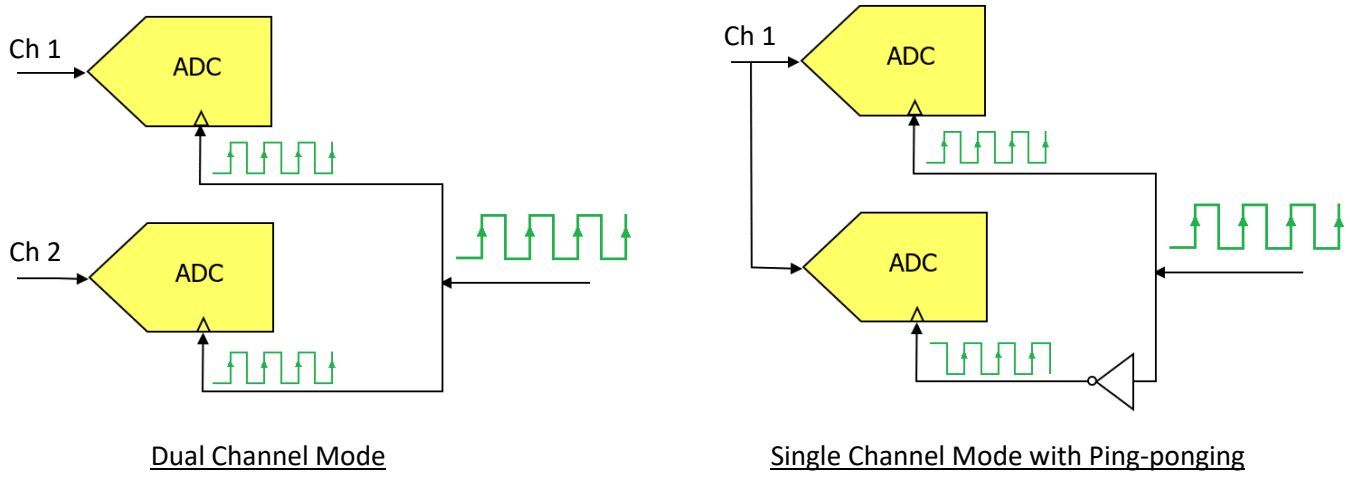
Every CompuScope allows acquisition on a number of channels that is a power of 2. These numbers are given CompuScope Mode names that are listed in the table below:

CompuScope Mode	Number of Channel to be Acquired.
SINGLE	1 Channel
DUAL	2 Channels
QUAD	4 Channels
OCTAL	8 Channels

A a CompuScope cannot be configured, for example, to acquire only on 3 channels and the user must acquire on 4 channels and ignore data from one channel. The purpose of different CompuScope Modes is to optimize on-board CompuScope memory usage. For example, a CompuScope model with 4 GigaSamples (GS) of memory can acquire up to 4 GS of waveform data in SINGLE channel mode but can only acquire up to 1 GS on each channel in QUAD mode.

2.2 Channels and ADCs

Lower-speed waveform acquisition instruments can multiplex multiple input channels into the same ADC chip. This approach cannot be used at Gage speeds because settling times are not sufficient. Consequently, Gage CompuScopes always uses *at least one ADC per channel*, as shown in Figure 4.



Dual Channel Mode

Single Channel Mode with Ping-ponging

Figure 4: CompuScope ADC Configuration in Dual Channel Mode and in Single Channel Mode with Ping-ponging

The left image of Figure 4 shows a 2-channel CompuScope operating in DUAL channel mode. Each channel uses an identical ADCs whose input signal come from a CompuScope input channel connector, after passing through signal conditioning circuitry. As shown, the clocking signal is connected to each ADC through identical circuit paths. This way, the rising edges of the clocking signals relayed to each ADC are highly simultaneous (to within a few picoseconds). The resultant sampling is simultaneous to within picoseconds.

2.3 Ping-ponging

Some CompuScope models use *ping-ponging* or *channel interleaving*. This technique uses two ADCs operating in tandem to double the maximum sampling rate. The right image from Figure 4 shows a 2-channel CompuScope operating in SINGLE channel mode. In this case, the signal from Channel 1 is relayed to both ADCs, which are clocked 180^{o} out of phase, as indicated by the inverter component in the path of one of the clocking signals in Figure 4. In this way, sampling alternates between the two ADCs so that when their output data sets are interleaved, the user obtains a waveform sampled at twice the sampling rate of the ADCs. Switching from DUAL to SINGLE Mode on this CompuScope, therefore, adds a 2X ping-ponging sampling rate in addition to devoting all on-board memory to Channel 1.

2.4 Channel Enumeration

When activating different CompuScope Modes, active channels are always distributed uniformly across all channels. The table below indicates the active channels set for different Modes on CompuScopes with various channel counts.

	Active Channel Numbers			
CompuScope Channels	SINGLE Mode	DUAL Mode	QUAD Mode	OCTAL Mode
2-Channel	1	1, 2	-	-
4-Channel	1	1, 3	1, 2, 3, 4	-
8-Channel	1	1, 5	1, 3, 5, 7	1, 2, 3, 4, 5, 6, 7, 8

3 | CompuScope Sampling

For a given channel, the most important digitizer specification is its *sampling rate*. While its units could be specified in *MHz* or *GHz*, Gage documentation always uses *MegaSamples/Second (MS/s)* or GigaSamples/second (GS/s) (sometimes written elsewhere as *MSPS* and *GSPS*). These units for the sampling rate clearly distinguish it from other frequency values in discussion.

As its name suggests, the sampling rate determines the number of samples acquired by a CompuScope in a specific time. The inverse of the sampling rate is the *sampling interval*, which is equal to the horizontal grid interval in Figure 2. For example, a 2 GS/s sampling rate is equivalent to a 0.5 nanoseconds sampling interval.

For a signal that contains a maximum frequency f_{MAX}, the *Nyquist Criterion* requires a sampling rate of at least $2f_{MAX}$ to avoid misrepresentation of the signal. Figure 5 shows the consequences of sampling a high-speed signal wave with a frequency that is far above the *Nyquist frequency*, which is half the sampling rate. Clearly, the apparent frequency of the digitized signal is *aliased* to a much lower frequency than the actual signal frequency. Aliasing is acceptable only in certain communication applications where correct representation of a signal's carrier frequency is unimportant. A more complete description of the frequency aliasing that results from violation of the Nyquist criterion requires discussion in the frequency domain.

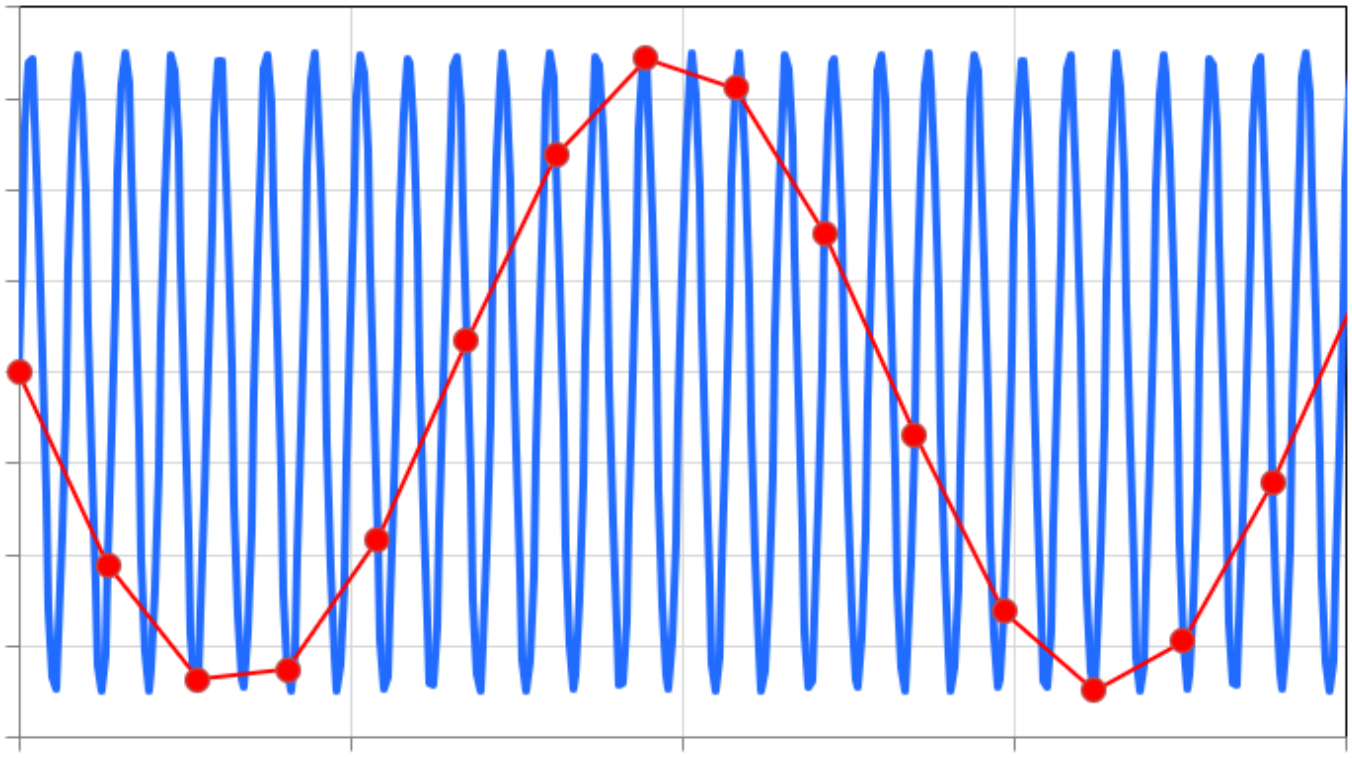


Figure 5: A high speed sine wave signal (blue) with whose frequency is far above the Nyquist frequency (half the sampling rate). This violation of the Nyquist Criteria causes the frequency of the digitized signal(red) to be misrepresented or aliased to a lower frequency.

While sampling at twice the signal frequency is the minimum acceptable in most applications, a higher sampling rate is generally always better. Figure 6 shows a simulated ultrasonic echo sampled at 2X, 5X and 10X the central frequency of the echo. While the 2X acquisition meets the Nyquist criterion, it traces out only the most rudimentary outline of the echo signal. By contrast, the 10X digitized signal clearly follows the subtle nuances of the echo signal and so 10X is the recommended over-sampling factor in ultrasonics.

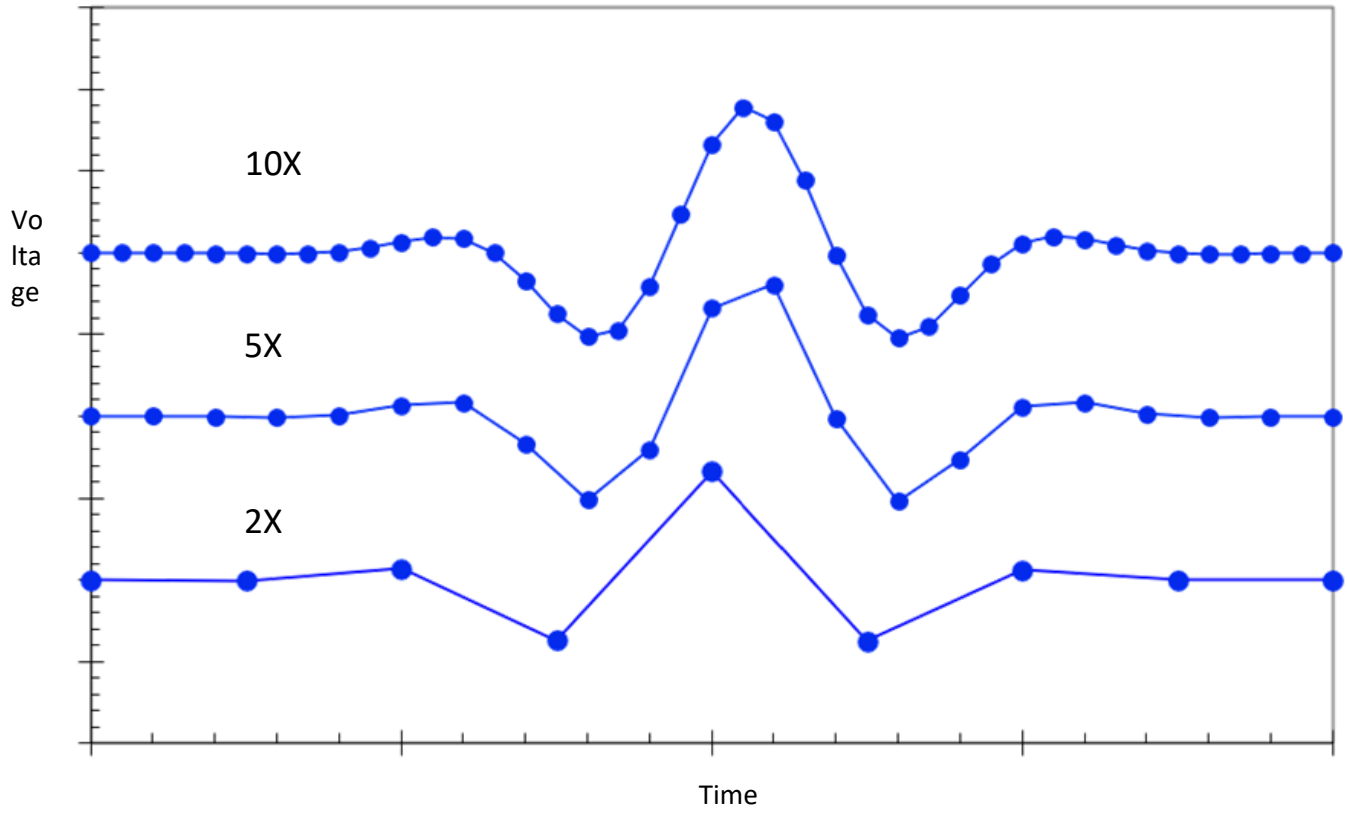


Figure 6: A signal digitized using a sampling rate that is 2X, 5X and 10X the signal frequency.

Generally, the only disadvantage of using a higher sampling rate is increased data volume, which leads on-board CompuScope memory to run out more quickly. Gage CompuScopes always operate their ADCs at near the maximum sampling rate, which allows for optimal tuning of circuitry to a narrow range of sampling rates for best performance. Most lower sampling rates are achieved by *decimation*, in which samples are periodically dropped by a CompuScope after acquisition by its ADCs. Using decimation, most CompuScope provide sampling rates below 1 MS/s. High Sample rate accuracies of order 1 part-per-million (ppm) are achieved using a high-accuracy internal 10 MHz reference source.

4 CompuScope Vertical Resolution

The vertical resolution of a CompuScope determines the precision with which vertical voltage measurements are made by its ADC chips. An ADC resolution of N bits divides the vertical voltage range into 2^N different discrete levels. Different CompuScopes models have resolutions of 8, 12, 14, 16-bits and so divide the vertical range into 256, 4096, 16384 and 65536 levels, respectively. The voltage amplitude that corresponds to the height of each ADC level is sometime called the *Least Significant Bit (LSB)*

CompuScopes with higher resolution are important for signals with *high dynamic range*, which are signals that contain both high amplitude and low amplitude components. This is because high amplitude signal components in a high dynamic range signal limit its amplification, which could otherwise enhance low amplitude signal components.

The importance of high resolution is illustrated in Figure 7, which shows a ringing square wave – a high dynamic range signal. The high amplitude of the square wave and its initial overshoot inhibit the user from lowering the CompuScope's input range to enhance small signal features. The inset of Figure 7 shows the signal magnified by a factor of 25X as digitized by both an 8-bit CompuScope and a 12-bit CompuScope. The 8-bit acquisition clearly shows the truncating effect of low-resolution that results because the input range is only divided into 256 levels. Ringing oscillations are so small beyond the first half of the inset that they are missed completely by the 8-bit CompuScope. By contrast, 12-bit digitization provides 16X more levels than the 8-bit digitization (since $2^{(12-8)} = 16$) and so the decaying oscillations can be followed throughout the inset.

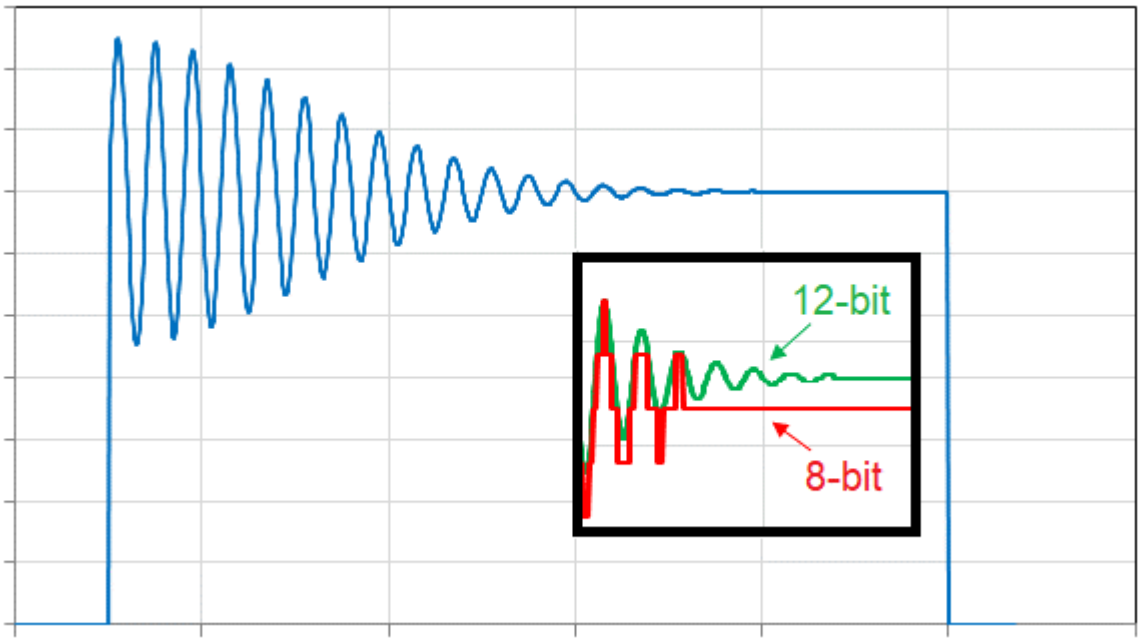


Figure 7: A ringing square wave signal. Inset shows a vertically-magnified view of its 8-bit and 12-bit digitization.

With regard to the size of Samples, Gage CompuScopes may be divided into two classes: 8-bit CompuScopes, for which 1 digitizer Sample is equal to 1 Byte and non- 8-bit CompuScopes (12-bit, 14-bit, 16-bit) for which one Sample is equal to 2 Bytes. Generally, 12-bit and 14-bit CompuScopes use 16-bit Samples and left-justify the data so that the lowest 4 and 2 bits are always 0, respectively. Only in rare cases are 12-bit CompuScopes samples packed data into 1.5 Bytes.

5 CompuScope On-board Memory

All CompuScope are equipped with fast on-board acquisition memory whose capacity is usually at least 4 GigaBytes. This memory is dedicated as a CompuScope acquisition target and cannot be used by the host PC like regular PC RAM. The amount of acquisition memory used is related to the acquisition time and the Sampling rate as follows:

Memory Used = (Sampling Rate) X (Acquisition Time)

Consider a 500 MS/s acquisition that lasts for 50 microseconds.
In this case:

Memory Used = (500 MS/s) X (50 microseconds) = 25,000 Samples

Alternately, the equation may be used to determine the maximum CompuScope acquisition time. For example, a 500 MS/s acquisition on a CompuScope with 4 GigaSamples of memory allows a maximum acquisition time of:

Acquisition Time = (4 GigaSamples) / (500 MS/s) = 8 seconds

6 | CompuScope Signal Conditioning

For most CompuScope models, input signals are not merely connected to ADC chips directly. Instead, the signal first passes through signal conditioning circuitry designed to improve the signal fidelity of digitization. CompuScopes generally allow adjustment of several configuration settings associated with signal conditioning, such as choice of input range and input termination.

6.1 Input Analog Bandwidth

Some signal conditioning is always applied by a CompuScope due to its *input analog bandwidth* - an often-overlooked attribute. If we consider that a CompuScope collectively acts as a low-pass filter, then the analog input bandwidth is equivalent to this filter's -3 dB roll-off frequency. The frequency response of an example CompuScope is shown in Figure 8.

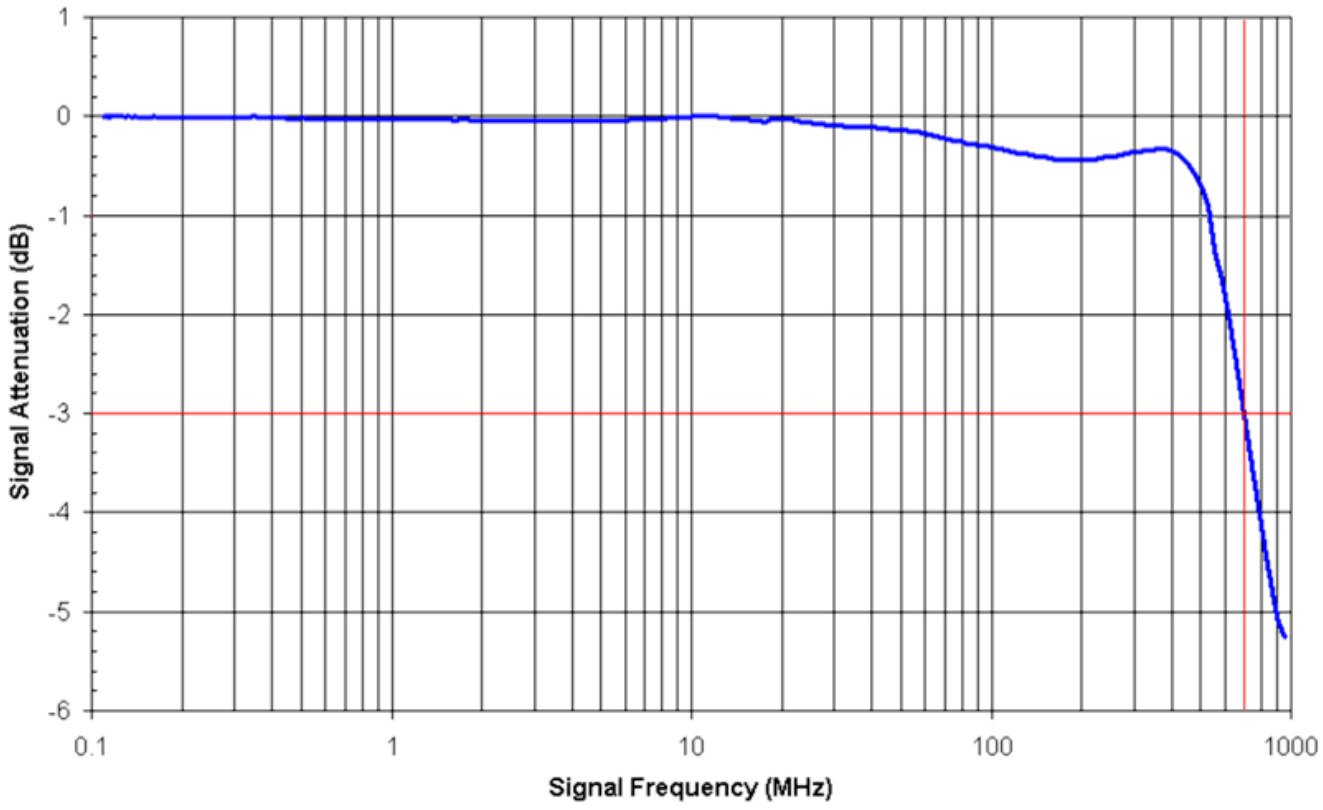


Figure 8: Frequency Response curve of CompuScope with its 700 MHz analog input bandwidth indicated.

Each point in the frequency response curve of Figure 8 is obtained by connecting a sine wave of fixed amplitude and frequency to a CompuScope and measuring the amplitude of the resulting acquisition. The input analog bandwidth of the CompuScope of Figure 8 is 700 MHz since the frequency response drops below -3 dB for higher frequencies. The frequency response is flat within ±1 dB up to over 500 MHz signal frequency. The input bandwidth depends slightly upon a CompuScope's input range.

The user must ensure that their input signal frequencies of interest are below the CompuScope bandwidth since higher frequencies will be severely attenuated. CompuScopes use their maximum sampling rate as a banner specification, oscilloscopes are in contrast always specified by their analog input bandwidth.

6.2 Input termination

At low signal frequencies, electrical instruments are generally designed to have the highest possible terminating input impedance. This is because high terminating impedance imposes minimal disturbance upon the circuit under test.

High terminating impedance is not the best choice signal frequencies significantly above 10 MHz. This is because the wavelength of such signals becomes comparable to the length of cables that relay signals to the CompuScope. In this case, mismatch between the cables' impedance and the CompuScope terminating impedance leads to distorting signal reflections.

Electrical transmission line theory reveals that if the terminating impedance can be *matched* to the cabling impedance, distorting signal reflections disappear. Since 50 Ω coaxial cable is used to connect to their inputs, all CompuScopes are equipped with matching 50 Ω input terminations. Most high-speed signal generators are designed for output termination by a 50 Ω load. Signals from generators not designed for a 50 Ω terminating load may be severely attenuated by a CompuScope's 50 Ω input.

Some CompuScope models are equipped with 1 MΩ termination – a high impedance termination like on lower speed instruments. A CompuScope's 1 MΩ termination should be used only with lower-speed signals that do not create significant signal reflections or with signals that are buffered by an appropriate active probe.

6.3 Input Ranges

The most important signal conditioning feature of Gage CompuScopes is their portfolio of multiple input ranges. Most CompuScopes are equipped with the following set of input ranges:

±5 Volts, ±2 Volts, ±1 Volts, ±500 milliVolts, ±200 milliVolts, ±100 milliVolts.

These six ranges are constructed as a combination of direct connection to the lower three ranges and connection through a divide-by-10 attenuator to achieve the upper three ranges.

The purpose of a CompuScope's input ranges is to allow the user to fully exploit the resolution of the ADC chips. If the user selects an input range that is too small, then their signal will saturate the input and will be clipped, which is severe distortion. Conversely, if the user selects an input range that is too large, the signal will then only span a small fraction of the input range and so the resolution of the ADC is not fully employed. Ideally, the user should select an input range so that the input signal spans 50% to 90% of the input range.

The input range portfolio of a CompuScope may be extended using external components. Gage in-line passive attenuator components effectively provide a CompuScope with higher input ranges. Similarly, for lower input ranges, Gage can recommend low-noise amplifiers to effectively provide a CompuScope with lower input ranges.

6.4 CompuScope Channel Auto-calibration

By default, any time the user changes its channel configuration setting (such as input range or input termination), a CompuScope subjects itself to an on-board autocalibration sequence, that may take several seconds to execute. During auto-calibration, the CompuScope switches its input away from the channel input connector and over to an on-board DC calibration source, without changing any of the input configuration settings.

The DC calibration source consists of an adjustable DC Zener reference source that can provide high-accuracy DC voltages over a wide range. Auto-calibration consists of two distinct operations. The first operation - *null calibration* - injects a 0 Volt signal into the input. The CompuScope then acquires this 0V signal and adjusts internal offset amplifiers until the corresponding ADC output corresponds to the center of its numeric range. The second operation, *gain calibration*, injects DC voltages near *+ FullScale* and near *–Fullscale* and then subtracts the correspondingly measured ADC digital output levels to calculate an apparent signal swing. The calibration algorithm then adjusts internal gain amplifiers until this swing corresponds to the known voltage swing provided by the Zener reference. The two operations – null and gain calibration - may not be completely independent and so the two operations may require execution in an iterative fashion until the optimal settings are achieved.

Auto-calibration is provided on CompuScopes to compensate against drifts in electrical component characteristics with temperature or component aging. Often a user works with the same CompuScope channel configuration for hours at a time and can easily suffer a few seconds of auto-calibration at the beginning of their measurements. Some users who construct *Automated Test Equipment(ATE)* , however, must execute multiple tests with different channel configurations (e.g. different input ranges). These users cannot suffer several seconds of auto-calibration delay with each test. For these users can be provided with Gage Fast Calibration setting that require the CompuScopes to auto-calibrate only once after reboot upon first encountering a specific set of channel configurations settings. Thereafter, the CompuScope remembers the former calibration thereby avoiding the need for unnecessary repetitive auto-calibrations.

6.5 CompuScope Factory Calibration

Before shipment, every CompuScope is subjected to the *Gage Performance Verification System*. This system uses a calibrated NIST-traceable calibrator unit that injects high-accuracy AC and DC signals into a CompuScope under test. Signal levels acquired by the CompuScope are required to agree with calibrator values within published CompuScope tolerances. In addition to vertical accuracy, the Performance Verification system tests other CompuScope functionalities, such as analog input bandwidth, trigger level accuracy and noise level.

If a CompuScope passes all Performance Verification tests, then it is shipped to the customer with a *Certificate of NIST-Traceable Calibration*. Some customers require, for example, yearly factory calibration of their CompuScope hardware. While Gage calibration services are available, they usually entail only verification of a CompuScope's compliance using the Performance Verification System. Since the board auto-calibrates itself during normal operation, CompuScope calibration drift only occurs due to significant drift in the on-board Zener references, which almost never happens. In practice, therefore, a CompuScope should virtually never require factory recalibration.

6.6 User DC Offset

In addition to offsets that internally adjust their vertical calibration, CompuScopes also provide a separate user-controlled DC offset that spans ±100 % of the input range. This user DC offset allow the user to change a CompuScope's bi-bolar input range into a unipolar input range. Consider a user with a unipolar positive signal that consists only of positive voltages. In this case, the negative half of a CompuScope's normal bi-polar input range will be wasted. However, using a DC offset of 1 Volt on the bi-polar ±1 Volt range, for example, the user effectively obtains a uni-polar 0 to 2 Volt range. This way, a 0 to 2 Volt signal will span the entire input range instead of wasting half of the ±2 Volt range.

6.7 Input Coupling

The *AC-coupling* of a signal refers to its high-pass filtering so that its DC component is removed. This removal may allow the signal's remaining AC components to be enhanced by amplification. While some older CompuScope models provide software selection of AC coupling, this feature has not been popular. Consequently, this feature was removed to increase signal fidelity through simplification of the circuitry. Instead, modern CompuScopes are offered with the option of AC-coupling only. A more flexible option is to use inexpensive in-line *DC Blocker* components, which may be optionally connected to AC-couple a signal.

7 | CompuScope Triggering

One of the most important considerations in CompuScope operation is *triggering*, which provides a signal timing reference called the *Trigger Position*. As described below, CompuScopes trigger in a fashion similar to an oscilloscope.

7.1 Trigger Settings: Source, Level and Slope

The three main controls for CompuScope triggering are the *Trigger Source*, the *Trigger Level,* and the *Trigger Slope*. The Source refers to the signal that is used to trigger CompuScope acquisition. The Source can be *Internal*, which means one of the CompuScope input channels, *External*, which means the *Trigger In* Input, or *Disabled,* which means that there is no Source. Figure 12 illustrates triggering internally on the input signal with a positive level and on both positive and negative slopes.

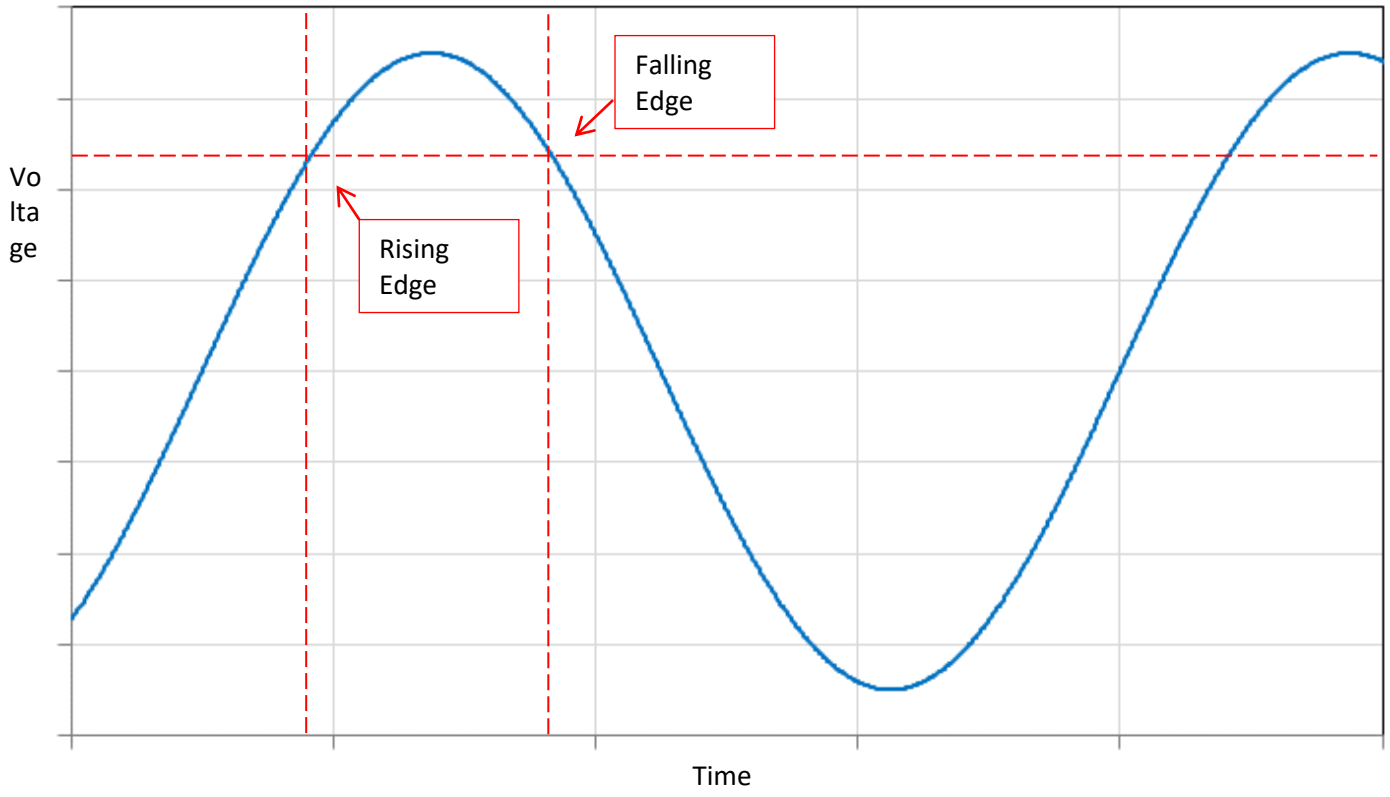


Figure 9: A digitized signal with the trigger level indicates as a horizontal dashed line. Trigger positions resulting from triggering on the signal's Rising Edge and the Falling Edge are indicated.

7.2 Trigger Hysteresis

Users may notice that a CompuScope cannot trigger on a signal below a certain amplitude. This behavior results from a feature called *Trigger Hysteresis* that suppresses false triggering. The effect of Trigger Hysteresis is illustrated in Figure 10.

The figure shows a noisy sine wave acquired by a CompuScope that is configured to trigger with trigger level shown and on a positive slope. A valid trigger is illustrated on the rising edge of the sine wave. Without Trigger Hysteresis, the invalid trigger illustrated in Figure 10 would also be detected as a trigger. To suppress such false triggers, the CompuScope qualifies the trigger event using a second *Hysteresis Threshold* that is slightly below the trigger level – typically by a few percent of the input range. Trigger Hysteresis requires that the signal must swing through both the trigger Level and the Hysteresis Threshold to be considered a valid trigger event. Figure 10 Illustrates that indeed the valid trigger event passes through both the trigger level and the hysteresis threshold, while the invalid trigger passes only though one. Consequently, trigger hysteresis has indeed eliminated the false trigger.

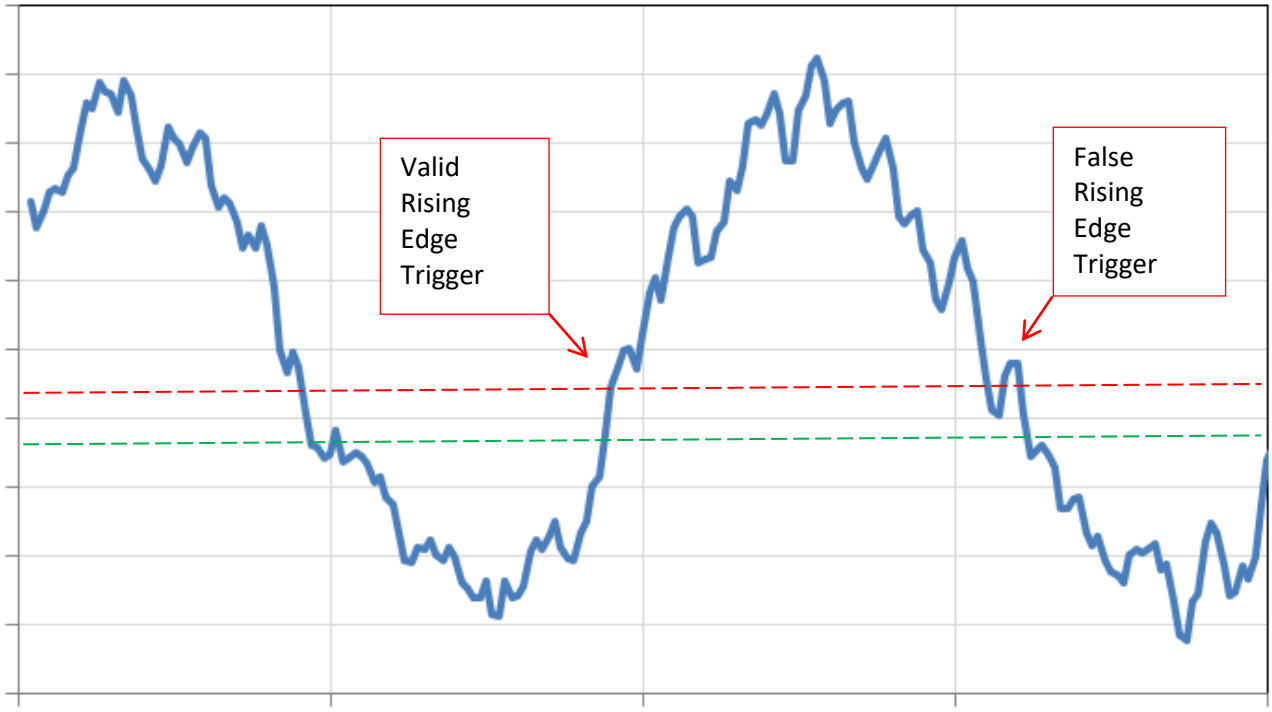


Figure 10: A noisy digitized signal indicating a valid and false rising edge trigger. The red dashed line indicates the trigger level, and the green dashed line is the hysteresis threshold.

7.3 Internal Versus External Triggering

Figure 9 and Figure 10 show examples of internal triggering on the input signal itself. However, most Gage users conduct *stimulus-response experiments*, illustrated in Figure 11, which always require external triggering.

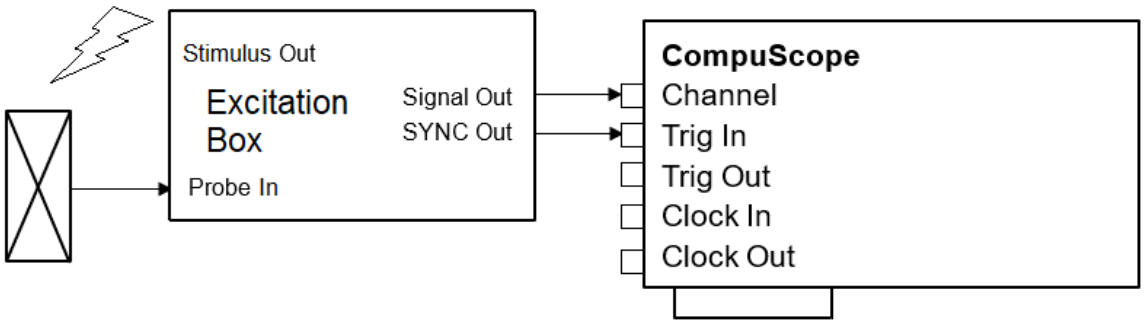


Figure 11: A CompuScope externally triggering in an idealized stimulus response experiment.

Figure 11 shows a probe connected to an excitation box that provides a conveniently-scaled version of the probe signal from its *Signal Out* output that is connected to a CompuScope channel. The box also creates a stimulus excitation of some sort that is illustrated by a lightning bolt. Simultaneous with the stimulus, the box produces a short pulse on its SYNC Out output. This pulse is connected to the *Trig In (External Trigger*) input of the CompuScope and is used to externally trigger it. In this way, all response signals received by the probe are referenced to the stimulus excitation event, which serves as the *Time=0* reference within the acquisition– much as the firing of a starter pistol serves as the Time=0 reference during a foot race.

The majority of CompuScope applications are stimulus-response experiments that use external triggering. These applications include ultrasonics, lidar, radar, communications, and many others. Generally, the only applications that use internal channel triggering are those where the event of interest is initiated by nature or by a third party, such as lightning monitoring, nuclear particle detection and communications monitoring.

7.4 Triggered CompuScopes Acquisition

A deeper understanding of CompuScope triggering requires a more detailed description of CompuScope waveform acquisition. A diagram of a simplified CompuScope is shown in Figure 12.

Figure 12 shows the ADC chip at the heart of the CompuScope that performs the digitization. The signal from the CompuScope channel input is connected to the analog input of the ADC chip through signal conditioning circuitry. A fixed frequency oscillator provides a clocking signal, whose every rising edge creates a new sample at the digital output of the ADC. These digital samples are relayed to CompuScope on-board acquisition memory.

In addition to clocking the ADC samples, the same clocking signal feeds a counter, as shown in Figure 12. This counter's output is connected to the address line of the on-board CompuScope memory. In this way, the memory address is incremented with each new sample generated by the ADC, starting from 0, so that successive ADC samples are stored in successive memory locations.

Figure 12 shows that the Trigger In signal is connected to a comparator that turns it into a digital rising edge. This rising edge latches the current counter value (which is also the current memory address) into the *Trigger Address Register*. This register therefore holds the address of the Trigger Position, which is the reference Sample within the waveform acquisition that corresponds to occurrence of the Trigger event.

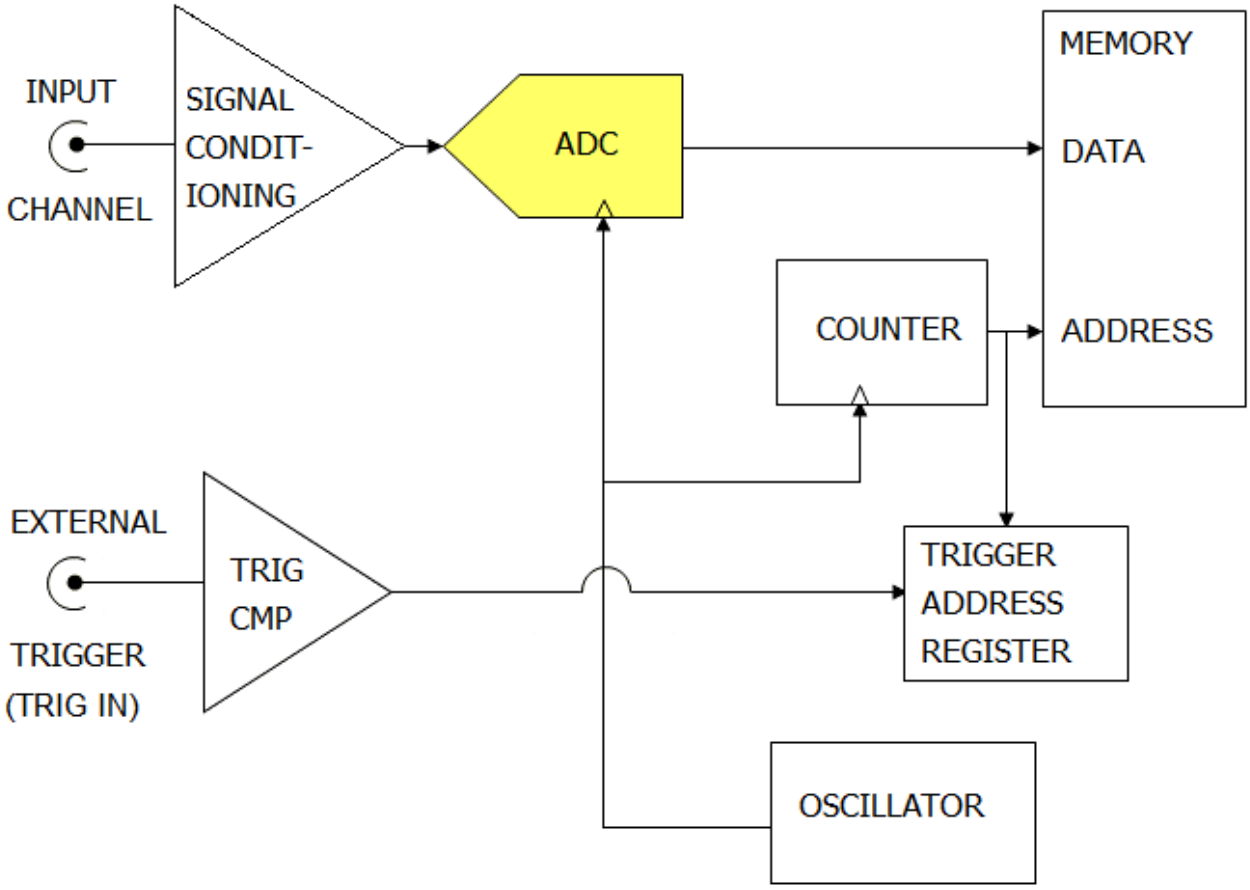


Figure 12: A block diagram of a simplified CompuScope digitizer.

An old analog oscilloscope managed triggering in a fundamentally different fashion from a CompuScope. On an analog oscilloscope, the trigger event is what initiates scanning of the electron beam across the phosphor screen, which writes the signal on the display. On an analog oscilloscope, therefore, the trigger event *initiates* waveform acquisition. By contrast, for a CompuScope, the trigger event does not initiate the acquisition, which was begun by a software command. Rather, the trigger event merely bookmarks the trigger position as the acquisition progresses. This approach enables the acquisition of *pre-trigger data*, which is not available on an analog oscilloscope.

Latching of the Trigger Position in the Trigger Address Register initiates the countdown of the *Post-Trigger Depth Counter* (not shown in Figure 12), whose expiration ultimately ends the acquisition. Once the acquisition is complete, the acquired waveform data may be downloaded from on-board CompuScope mode to PC RAM. Once there, waveform data may be displayed, analyzed, or stored to drive.

7.5 Segment Size, Pre-Trigger Data and Post-Trigger Data

The total number of samples acquired during a CompuScope waveform acquisition is referred to as the *Segment Size*. It is equal to the sum of the *Pre-Trigger Data* and the *Post-Trigger Depth:*

Segment Size = Pre-Trigger Data + Post-Trigger Depth

The Post-Trigger Depth must be a multiple of an increment (for example 128) that is specific for each CompuScope model. While the Post-Trigger Depth may be set to large values up to the size of the CompuScope memory, the Pre-Trigger Data volume is limited by the size of a dedicated buffer whose size depends upon the CompuScopes model. The buffer size is typically less than 1 MegaSample and is shared amongst the number of active channels. The smaller maximum size of the pre-trigger data volume is not a serious limitation in most applications since users are usually interested in pre-trigger data only immediately before the trigger event.

7.6 Trigger Timeout

Every CompuScope is equipped with a dedicated on-board counter that is used to time the *Trigger Timeout*, which is the time during which the CompuScope will await a trigger event. After the Trigger Timeout time expires, the board will issue an automatic trigger and will trigger immediately. Generally, the user should set to the Trigger Timeout so that its value far exceeds the expected time for a trigger to occur.

The purpose of the Trigger Timeout functionality is to safeguard against a trigger event that is later than anticipated or that never occurs at all. In these cases, the Trigger Timeout inhibits the controlling software application from freezing as it awaits the delinquent trigger.

Of course, the automatic trigger created by the expiration of the Trigger Timeout is not correlated with the signal. Consequently, signal acquisition initiated by the Trigger Timeout typically does not contain useful data.

In applications like lightning testing, a trigger event may take days to occur. In these applications, the user must ensure that the Trigger Timeout is set to infinity so that the CompuScope will correctly wait forever for the trigger event.

7.7 Trigger Delay and Trigger HoldOff

An auxiliary trigger settings is the *Trigger Delay*. This value specifies the period after the trigger event by which the CompuScope will delay initiating countdown of the post-trigger depth counter. In this way, the user can inhibit acquisition of data for a programmable amount of time after the trigger event.

In ultrasonics, the External Trigger event is simultaneous with excitation of an ultrasonic transducer. A long interval may occur after excitation before the first ultrasonic echo of interest returns to the transducer. In this case, the Trigger Delay may be used to inhibit unnecessary acquisition of this uninteresting data before the first echo's arrival.

The *Trigger Holdoff* specifies the period after waveform acquisition begins during which trigger events will be ignored. The *Trigger Holdoff* must be set to be equal to at least the amount of pre-trigger data. This will ensure that pre-trigger acquisition is not prematurely terminated by a trigger before all the requested pre-trigger data has been acquired. Larger values of the *Trigger Holdoff* may be used to inhibit triggering in obscure applications.

7.8 Complex Triggering

The vast majority of CompuScope applications are solved using simple triggering off a single Trigger Source with a single Level and Slope. Gage CompuScopes are, however, also capable of *Complex Triggering*, which is the ability to trigger by logically ORing together multiple independent trigger events. Each potential trigger source – all input channels and the Trig In input - are equipped with two distinct *trigger engines*. These outputs of these two engines for all sources are ORed together to determine if the CompuScope should trigger. For example, one

might configure a CompuScope to trigger if it detects a rising edge that crosses a 0.5 Volt level on Channel 1 OR if it detects a falling edge that crosses a 0.25 Volt level on Channel 3.

The most important example of complex triggering is a particle physics experiment. In a typical particle physics experiment, a high-speed particle impacts a target. As a result, scattered particles travel away from the target. Whenever a particle impinges upon a particle detector, it creates a particle pulse whose amplitude increases with the kinetic energy of the particle. Imagine an array of such detectors surrounding the target. The signal from each detector is connected to the input channels of a CompuScope. In this example case, the user might want to trigger acquisition on all channels if the amplitude *of any one* of the detectors' pulse amplitudes exceeds 0.5 Volts. To implement this requirement, the user must simply enable a single trigger engine for each CompuScope input channel with a positive *Slope* and a *Level* of 0.5 V. Signals will then be acquired on all channels if any one of the channels detects a pulse with an amplitude of greater than 0.5 Volts – exactly as the user requires. This triggering scheme may be extended to multiple CompuScopes by appropriately connecting and configuring their Trig In and Trig Out connections.

The second important application of complex triggering is Windowed Triggering. In this case, the user must trigger their CompuScope if and only if the signal amplitude exits a window delimited by two voltages – say 0.5 Volts and 0.75 Volts. In this example, the user must set only the two trigger engines for the active channel. The two levels are set to 0.5 Volts and 0.75 Volts with falling and rising slopes, respectively. A special case of windowed triggering is absolute value triggering, where the two levels are equal and opposite.

8 | CompuScope Memory Mode Acquisition

Traditional CompuScope acquisition occurs in *Memory Mode*, in which all CompuScope acquisition operations occur with no simultaneous communication to the host PC required. Such communication only occurs after acquisition is complete and waveform data are downloaded from the CompuScope to the PC. This chapter describes the two Memory Mode methods of acquiring multiple waveforms from a CompuScope, which are: *Single Record Mode* and *Multiple Record Mode*.

8.1 Single Record Acquisition

The simplest CompuScope waveform acquisition is in Single Record Mode. All high-level operations required to complete a Single Record acquisition are listed and described below:

1. Initialization
2. Configuration
3. Start CompuScope acquisition
4. Await occurrence of Trigger Event
5. Await completion of acquisition
6. Download waveform data to host PC.

These operations are manifested within the simplest CompuScope SDK program called *GageAcquire*. Step 1 is a simple software subroutine call to initialize both the CompuScope hardware and the CompuScope software drivers. Step 2 is the configuration of all CompuScope *configuration settings*, which fall into three groups: *Acquisition settings* (such as sampling rate), *Channel Settings* (such as channel input range) and *Trigger Settings* (such as trigger source). Configuration occurs in two phases. First, the controlling software application obtains the required variable setting values (for example, from a configuration file or from a software GUI) and assigns these values to corresponding internal driver variables. Second, the application communicates with the CompuScope hardware and implements these configuration settings using the now-assigned internal driver variables.

In Step 3, the CompuScope starts acquiring pre-trigger data from the CompuScopes ADCs and storing the results in CompuScope on-board memory. In Step 4, the CompuScope awaits a trigger event. Upon its occurrence, the CompuScope proceeds to Step 5 and awaits completion of the acquisition. Step 4 is often dropped because it is not strictly necessary since if the acquisition completes, then the trigger event must have occurred.

After Step 5, the acquisition is complete, and all acquired waveform data reside in CompuScope on-board memory. In Step 6, the controlling software application downloads the acquired data to PC RAM, from where they may be displayed, digitally processed, or stored permanently to drive.

The user can acquire multiple waveforms in Single Record Mode signal by adding a final line loop back to Step 3. Steps 1 and 2 are avoided since they need only be executed once at the beginning of the repetitive acquisition sequence.

Because of the extremely fast data transfer of the CompuScope hardware, a repetitive Single Record acquisition can keep up with a trigger rate (or *Pulse Repetition Frequency (PRF))* of up to 10,000 waveforms per second without missing triggers. By contrast, cable-connected oscilloscopes can typically provide PRFs only of order 100 Hz. Its ability to handle much higher PRFs is a key advantage that a CompuScope provides over a traditional oscilloscope. PRFs naturally decrease with increased channel count and waveform length (Segment Size). PRF performance curves are available for all CompuScope models.

8.2 Multiple Record Acquisition

While repetitive acquisition in Single Record Mode allows for very rapid triggering, even faster rates can be achieved using *Multiple Record Mode*, for two important reasons. Firstly, multiple records (or waveforms) are stacked within CompuScope on-board memory, as illustrated in Figure 13. Between acquisitions in Single Record Mode, the user must download the waveform data after each acquisition. Accordingly, no data download is necessary between record acquisitions and so the maximum attainable trigger rate (PRF) is higher than in Single Record Mode.

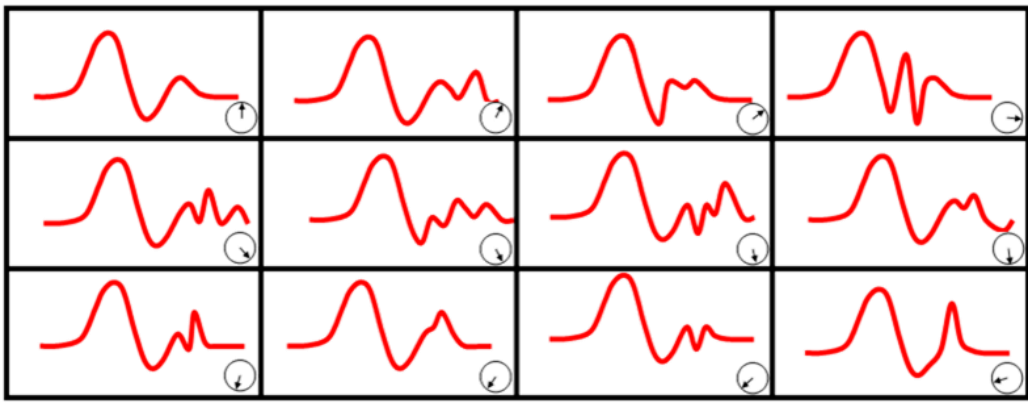


Figure 13: Contents of CompuScope on-board memory after a Multiple Record acquisition of multiple short waveforms (records). Time Stamps are indicated as stopwatches in the lower right of each waveform.

The second reason that Multiple Record Mode can provide faster repetitive triggering is due to its lightning-fast hardware re-arm. Between Single Record acquisitions, a CompuScope must be re-armed with a software command, which introduces several microseconds of software latency. By contrast, in Multiple Record mode, the CompuScope automatically re-arms itself in hardware with no software commands required. This hardware re-arm occurs within a microsecond, which enables trigger rates (PRFs) of over 100 kHz.

The number of records acquired in a Multiple Record Acquisition is called the *Segment Count*. Roughly, the maximum segment count is equal to the amount of CompuScope on-board memory available divided by the Segment Size. For example, consider a CompuScope with 4 GigaSamples of Memory and a user who wants to acquire 10,000 Sample waveforms on 2 channels. In this case, the maximum Segment Count is roughly 4 GS/(2 Ch)/(10,000 S) = 200,000 records. The actual maximum Segment Count is slightly less since we have neglected to count memory occupied by Time Stamps and other small factors.

Once a Multiple Record acquisition is complete, all acquired records must be downloaded, which will take longer than the download of a single record. Consequently, Multiple Record is a powerful feature for the acquisition of a burst of triggers, after which a break occurs when all data may be downloaded. For example, a 2D scanning system may combine multiple 1D scans, between which there is a break during repositioning to download the data from each 1D Multiple Record acquisition.

8.3 Time Stamping

Multiple Record mode acquisitions conserve on-board CompuScope memory by only acquiring the data of interest after each trigger, rather than acquiring 100% of the time. This suppression of uninteresting inter-trigger data, however, destroys the knowledge of how much time elapsed between acquired records. Recovery

of the inter-trigger time in Multiple Record mode is achieved by the CompuScope *Time-Stamping* functionality. The CompuScope is equipped with a high-speed counter whose value is latched every time a trigger event occurs. This Time Stamp value is then appended as a footer after the corresponding waveform data in CompuScope memory. Tiem Stamps are illustrated in Figure 13 as little stopwatches associated with each record. The Time Stamp clock may be referenced to absolute time by, for example, triggering on an IRIG signal and using the corresponding Time Stamp value as a reference.

Time stamps are particularly useful for waveforms whose repetition interval is not constant. This is usually not the case in stimulus-response experiments where the user typically initiates stimuli at a uniform frequency. Usually non-uniform triggers are initiated by nature or a third party, as in particle physics, acoustic emission, lightning testing, or communications monitoring. Time Stamps provide the occurrence time of each trigger so that the user can determine exactly how much time elapsed between successive triggers.

9 | CompuScope Auxiliary Connections

Each CompuScope model is equipped with four auxiliary connections: Trigger In (or External Trigger), Trigger Out, Clock In, and Clock Out, which are illustrated in Figure 14.

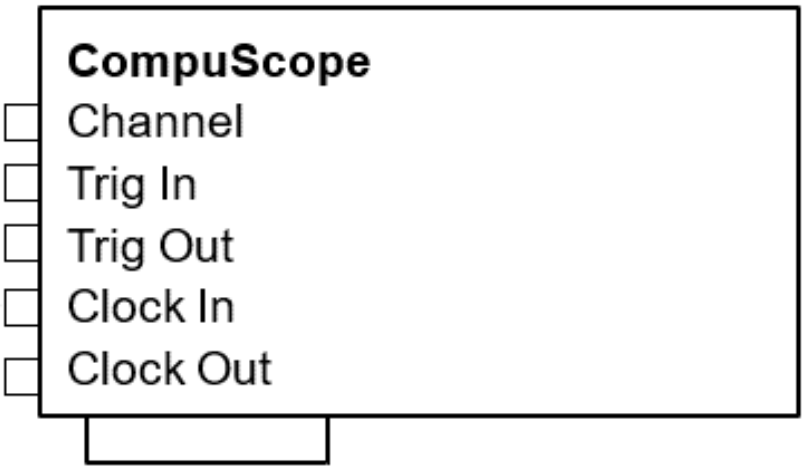


Figure 14: Diagram of CompuScope indicating four Auxiliary Connections: Trigger In (or External Trigger), Trigger Out, Clock In (or External Clock), Clock Out.

9.1 Trigger In (External Trigger)

Of the four connections, *Trigger In (or External Trigger)* is by far the most important. As discussed in Section 7.3, the majority of CompuScope applications use external triggering. Generally, the best external trigger signal is a sharp square pulse of a few microseconds duration that has an amplitude of about 3 Volts. The pulse must also be able to drive the high impedance load presented by the Trig In input. The user's External Trigger Pulse is correlated with their input signals so that the interesting region of the signal is close to the Trigger Position.

9.2 Clock In

Every CompuScope model is equipped with a portfolio of internal sampling rates that may be selected by the user. Users' signals typically bear no timing relationship to the CompuScope's internal sampling clock and there is no way to impose such a relationship. In this case, we say that the signal and the sampling clock are *asynchronous*, which means that they are unrelated.

In some applications, the customer has an auxiliary timing signal that is *synchronous* or related to their input signal. Such a timing signal may be used to make the CompuScope's sampling clock also synchronous to the input signal. The CompuScope's ADC clocking architecture is shown in Figure 15. Switches (MUX) are shown that allow for selection of multiple ADC clocking configurations.

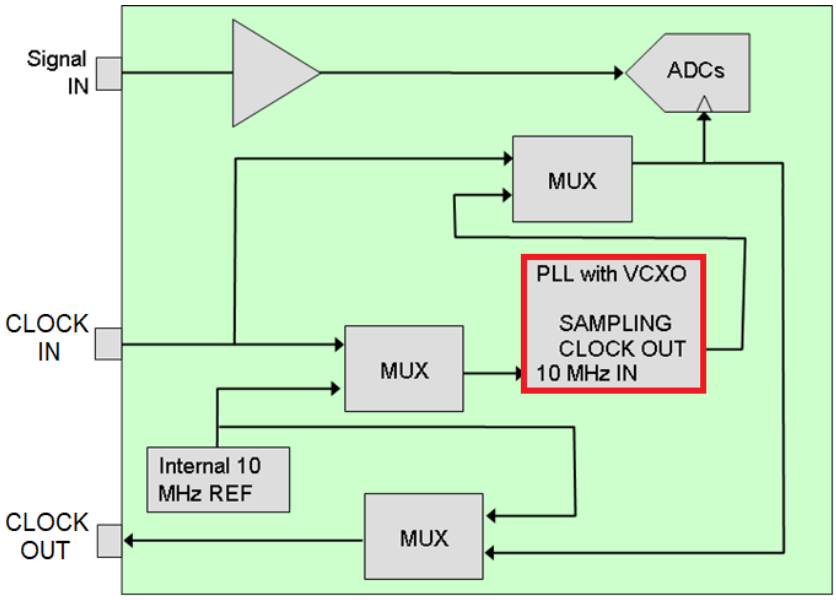


Figure 15: CompuScope ADC clocking architecture

Internal sampling rates are provided to the CompuScope's ADCs by a *Phase Lock Loop (PLL)* that includes a *Voltage Controlled Crystal Oscillator (VCXO*). This PLL is *disciplined* by an on-board *10 MHz reference oscillator* whose 10 MHz frequency is accurate within a few parts-per-million (PPM). If the ADC's clocking frequency was designed to be 1000 MHz, for example, then the PLL's internal feedback would ensure that the frequency of its output signal was at a exactly 100X the 10 MHz reference frequency. In this way, the internal 1000 MHz ADC clocking is synchronous with the 10 MHz reference frequency and inherits its PPM accuracy.

The first alternative to internal clocking is for the user to provide a high-speed external clocking signal to a CompuScope's Clock In input, which is then connected directly to the clocking input of the ADC chips. For example, if the user provides a 275 MHz external clocking signal, then the CompuScope will sample at 275 MS/s. The user must ensure that their external clocking signal has sufficient amplitude, can drive the 50 Ω load and is within the clocking frequency limits of the ADC chips, which is published for each CompuScope model. The external clocking signal can never become inactive since this corresponds to a disallowed 0 Hz frequency.

The utility of external clocking application is illustrated by an example. Consider a CCD camera that continuously clocks out analog pixel amplitudes together with a 100 MHz *pixel clock*. Even if the CompuScopes were configured to internal sample at a 100 MS/s (100 MHz) rate, this 100 MHz frequency would not be equivalent to the 100 MHz pixel clock frequency, since they are generated by separate and unconnected circuits. Accordingly, the CompuScope's 100 MS/s sampling is not synchronous to the pixel clock and its edges would drift with respect to the pixel clock edges over time. Clearly, the best way to sample the CCD signal is to use the synchronous 1000 MHz pixel clock signal as an external sampling clock source for the CompuScope.

The second alternative to internal clocking to provide a 10 MHz reference to a CompuScope's Clock In input. Instead of the CompuScope's on-board internal 10 MHz reference, this external 10 MHz signal now disciplines the PLL of Figure 15. The CompuScope still samples by selecting from the portfolio of internal sampling rates. However, now these sampling rates are disciplined by the user's external 10 MHz reference (instead of by the CompuScope's internal 10 MHz signal) and so are synchronous to it.

External high-accuracy 10 MHz reference signals can improve the accuracy of CompuScope clocking. Examples include atomic standards or GPS standards like IRIG - both of which can provide sub-PPM accuracies. Some research facilities distribute a common high-accuracy 10 MHz signal for all researchers.

External 10 MHz reference outputs are often available on high-speed signal generators. Unlike a high-accuracy absolute reference, the generator provides a relative reference that is synchronous to the generator's output signal.

9.3 Synchronous Triggering

As we have seen, a CompuScope's Clock In input can accommodate two different modes: External Clock Mode, where a high-speed clock signal drives ADC clocking directly, and 10 MHz Reference Mode, where a 10 MHz external signal disciplines the CompuScope's internal sampling rates. Both techniques are employed for one or both of these same two reasons:

1. To externally impose absolute accuracy upon CompuScope ADC sampling so that it may be compared with universal standards or with measurements from separate instruments.
2. To externally synchronize CompuScope ADC sampling to a user's signal and trigger. This relative synchronization is unrelated to the absolute sampling accuracy, which may improve or degrade in consequence.

To further illustrate synchronous sampling, we consider ADC sampling around the Trigger Position, represented in Figure 16.

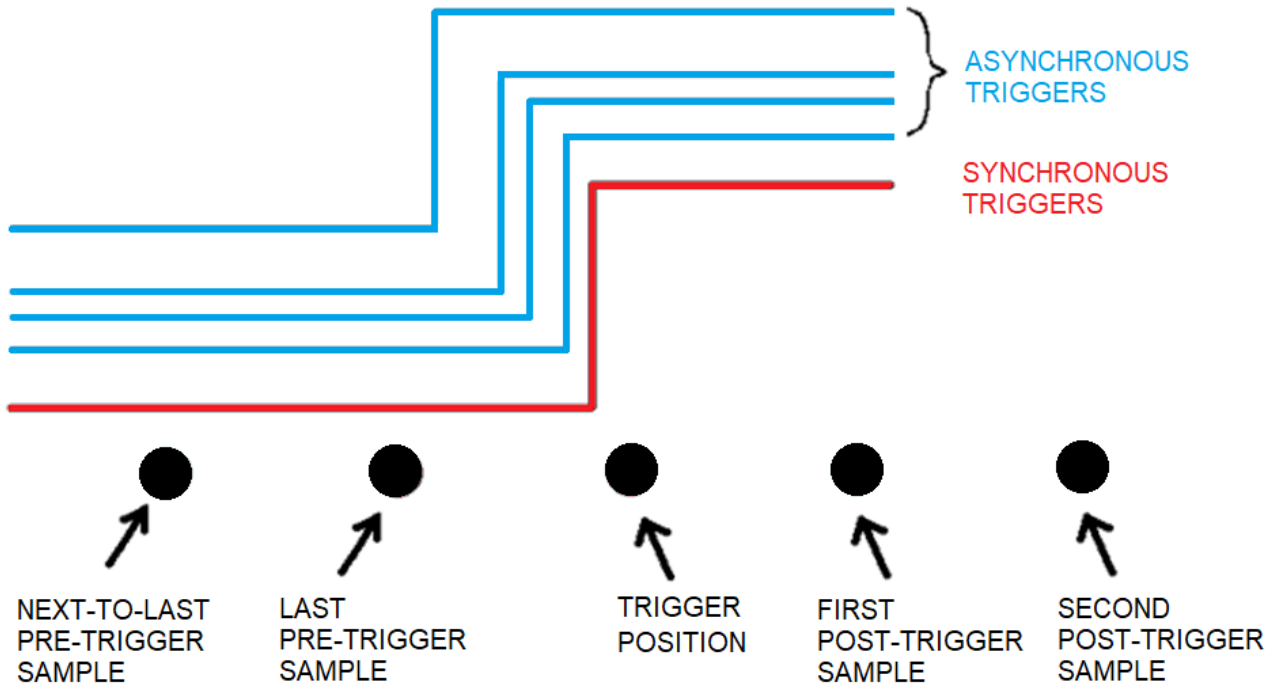


Figure 16: Synchronous and asynchronous trigger pulses and their relation to digitized samples near the Trigger Position.

Suppose that a repetitive external trigger signal's rising edge is coincident with excitation in a stimulus-response experiment, as described in Section 7.3. If the user has made no provisions for synchronous sampling, then their trigger edges will naturally be asynchronous or unrelated to the CompuScope's internal sampling clock. Accordingly, separate edges will occur at different phases of the sampling cycle. Nonetheless, all triggers edges that occur between the last pre-trigger Sample and the Trigger Sample (Or Trigger Position) will all be logged with this same Trigger Position. In consequence, when waveforms from successive acquisitions are compared, they will appear to jitter by exactly one Sample. This 1-Sample jitter is no failing of the CompuScope hardware but is a fundamental consequence of an asynchronous clock and trigger. The consequences of this 1-Sample jitter are minimal in many applications and are diminished by using the highest possible sampling rate.

The 1-Sample jitter endemic to asynchronous acquisition may be greatly improved by making the acquisition synchronous. Often this is achieved by the user, who prepares synchronous external clock and external trigger signals. In Figure 16, multiple synchronous triggers are indistinguishable since they all occur upon the same phase of the sampling clock. Synchronous sampling thus overcomes 1-sample jitter and the residual jitter of synchronous sampling results only from the electrical jitter of the CompuScope hardware components, which are typically of picosecond order. This enormous improvement is powerful in stimulus response experiments like ultrasonics and radar where determining the exact timing of the reflected echoes is of paramount importance.

9.4 Trigger Out

Once configured to be active, the CompuScope *Trigger Out* connection outputs a digital pulse with a width below 1 microsecond upon every CompuScope trigger. No further software commands are required to control the Trigger Out function. The Trigger Out pulse is subject to short, fixed propagation delays and so typically occurs a few hundred nanoseconds after the trigger event.

Figure 11 shows an excitation box that initiates a stimulus and simultaneously externally triggers a CompuScope from its SYNC output. Typically, the box will emit stimuli and accompanying trigger pulses at a fixed repetition frequency. There are two immediate disadvantages of this set-up. First, although the CompuScopes is very adept at fast repetitive triggering, if triggers from the box are too fast, the CompuScope will miss them. Secondly, if the trigger repetition rate is slow, then the CompuScope will waste time awaiting triggers. Both disadvantages are overcome using Trigger Out.

If the excitation box of Figure 11 had its own Trigger In input (not shown), then the CompuScope's Trigger Out pulse could drive this input. In this way, the direction of the triggering arrow is reversed - making the CompuScope *Trigger Master,* instead of the box. In this scenario, the CompuScope is triggered from software, which causes emission of a Trigger Out pulse that externally triggers the excitation box. Trigger Out usage immediately solves the two disadvantages of regular external triggering. Since Trigger Out pulses are generated only once it is ready, the CompuScope will never miss triggers. Secondly, the CompuScope does not have to waste time awaiting triggers since the software can generate the next trigger as soon as possible.

The previous section shows synchronous triggering reduces repetitive acquisition jitter from one Sample to well below one Sample. The third and most important advantage of using Trigger Out is that it makes acquisition synchronous without requiring the user to prepare and provide synchronous external clocking and triggering signals.

The Trigger Out pulse is created synchronous to the CompuScope's sampling clock – like the synchronous triggers shown in Figure 16. If it externally triggers the excitation box, then the resulting excitation will inherit the synchronicity of Trigger Out pulse and so will also be synchronous to the sampling clock. In this way, the excitation and its dependent response will also be synchronous to the CompuScope's sampling clock and so will not suffer 1-Sample jitter.

Triggering with Trigger Out makes acquisition synchronous if and only if the excitation box is analog – like an analog ultrasonic pulser/receiver. In this case, excitation pulses become nothing more than a complex, delayed version of the synchronous Trigger Out pulse. On the other hand, some excitation boxes are digital and create excitations using a *Digital-to-Analog Convertor* (DAC) with its own clock. This second clock source effectively destroys the synchronicity of the Trigger Out pulse. In this case, there is no alternative to using a common clock signal for both the CompuScope and the excitation box.

9.5 Clock Out

Clock Out is the least useful of the four CompuScope auxiliary connections. It can generate a high-speed clock. However, for reasons of internal architecture, this is usually not the ADC sampling clock but is a divided down version of it. Alternately, the Clock Out connection may be configured to output the CompuScope's 10 MHz reference clock. While this signal may be used to synchronize external instruments, it is not as accurate as other available 10 MHz sources.

10 | CompuScope Software

Gage CompuScopes are always controlled by the host PC and so are equipped with a powerful library of controlling software. The low-level CompuScope drivers reside at the *kernel level* of the operating system. In the kernel, the drivers take care of detailed CompuScope low-level register programming and harmonize differences between different CompuScope models. Controlling software applications (which reside at the *Applications Level*) call the CompuScope drivers through a *Dynamically Linked Library (DLL)* using the Gage CompuScope *Applications Programming Interface (API).* This API is a set of subroutine calls that are universal for all CompuScope models. Accordingly, software applications written using this API will automatically control all CompuScope models, and the user can switch between CompuScope models with no reprogramming required.

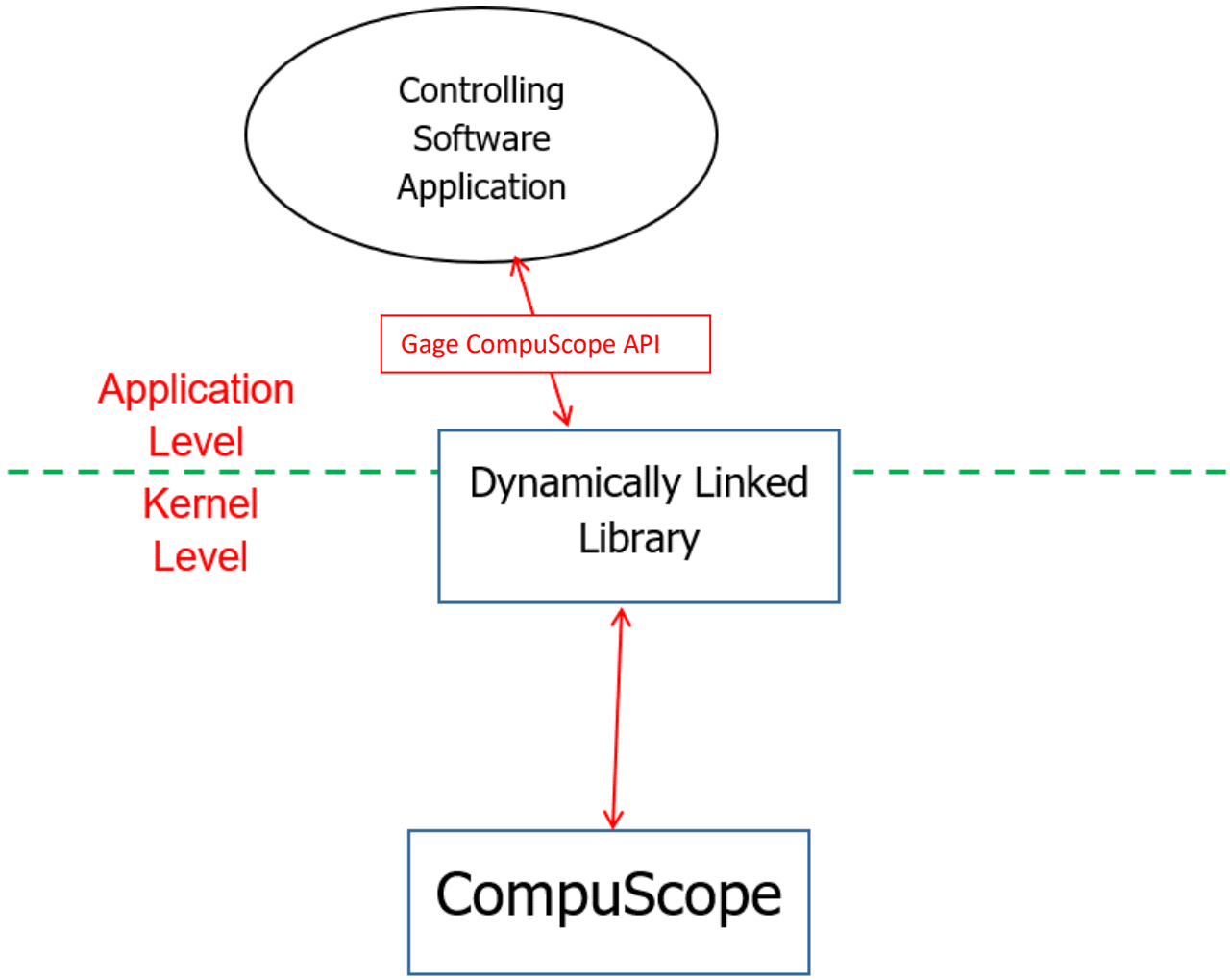


Figure 17: Structure of Gage CompuScope drivers, which operate at the operating system's kernel level and communicate with software applications through a Dynamically Linked Library (DLL).

Gage CompuScope drivers are available for select versions of the Microsoft Windows operating system. Support is also available for certain versions of the Linux operating system.

The CompuScope Windows drivers are installed with two software utilities. *CompuScopeManager* allows for detection of the CompuScope hardware and set-up of a virtual (or simulated) CompuScope. CompuScopeManager also reports on hardware, firmware, and software versions in addition to providing error reporting tools. *CsTest* is a simple oscilloscope utility that provides basic waveform acquisition and display capabilities.

GageScope is Gage's flagship oscilloscope software. It operates any CompuScope with no programming required. From GageScope, users can acquire, display and store waveforms acquired by their CompuScope hardware.

While GageScope is a powerful software application that allows users to test out their CompuScope hardware and to troubleshoot their experiment, it is rare that GageScope provides a complete solution to a user requirement. This is because:

1. GageScope does not allow control of other devices that may be required for the user's measurement system.
2. Since GageScope displays each acquired waveform, it is not optimized for the fastest repetitive capture performance.
3. GageScope does not extract application-specific parameters from raw waveforms, which may be required for the application.

Gage CompuScope *Software Development Kits* (SDKs) allow programmers to integrate their CompuScopes are modular instruments into a master controlling software application that perfectly addresses their requirement. Each SDK provides a complete manual and several sample programs illustrating usage of CompuScopes in different operating modes. Each sample program is structured to provide the simplest possible starting point for a user's applications. Samples programs are also available that illustrate eXpert firmware options, described in Section 12. The main three CompuScopes SDKs are available for *C/C#,* for *LabVIEW* and for *MATLAB*. Support is also available in the C/C# SDK for *Python*, *VisualBasic.NET* and *Delphi*.

11 | Multi-CompuScope Systems

Sometimes, users' channel count requirements exceed the number of available CompuScope channels. In these cases, users must employ multiple CompuScopes to achieve their channel count requirements. For example, a 16-channel system may be constructed using four 4-channel CompuScopes in the same PC. Sometimes these cards are required to work independently - wherein each CompuScope independently triggers and acquires waveforms. More often, however, multiple CompuScopes must trigger and acquire together as if they were all channels were on one CompuScope.

Formerly, CompuScopes were available configured in Master/Slave Mode, wherein an internal bridgeboard connected and synchronized the clocking and triggering of multiple CompuScopes. While this configuration is no longer available, testing has shown that equivalent synchronization may be achieved by appropriately distributing clocking and triggering signals via external cabling to multiple CompuScopes.

Figure 18 shows two CompuScopes externally synchronized through external cabling. Both the trigger and clock source signals are split by *RF power splitters* whose two outputs are connected to the Trigger In (External Trigger) and Clock In (External Clock) of each CompuScopes. True RF power splitters are used since simple co-axial Tees would not maintain 50 Ohm impedance matching and so would cause distorting signal reflections. Further, the two RF splitter outputs must be connected to the CompuScopes through equal length co-axial cables for identical signal propagation times.

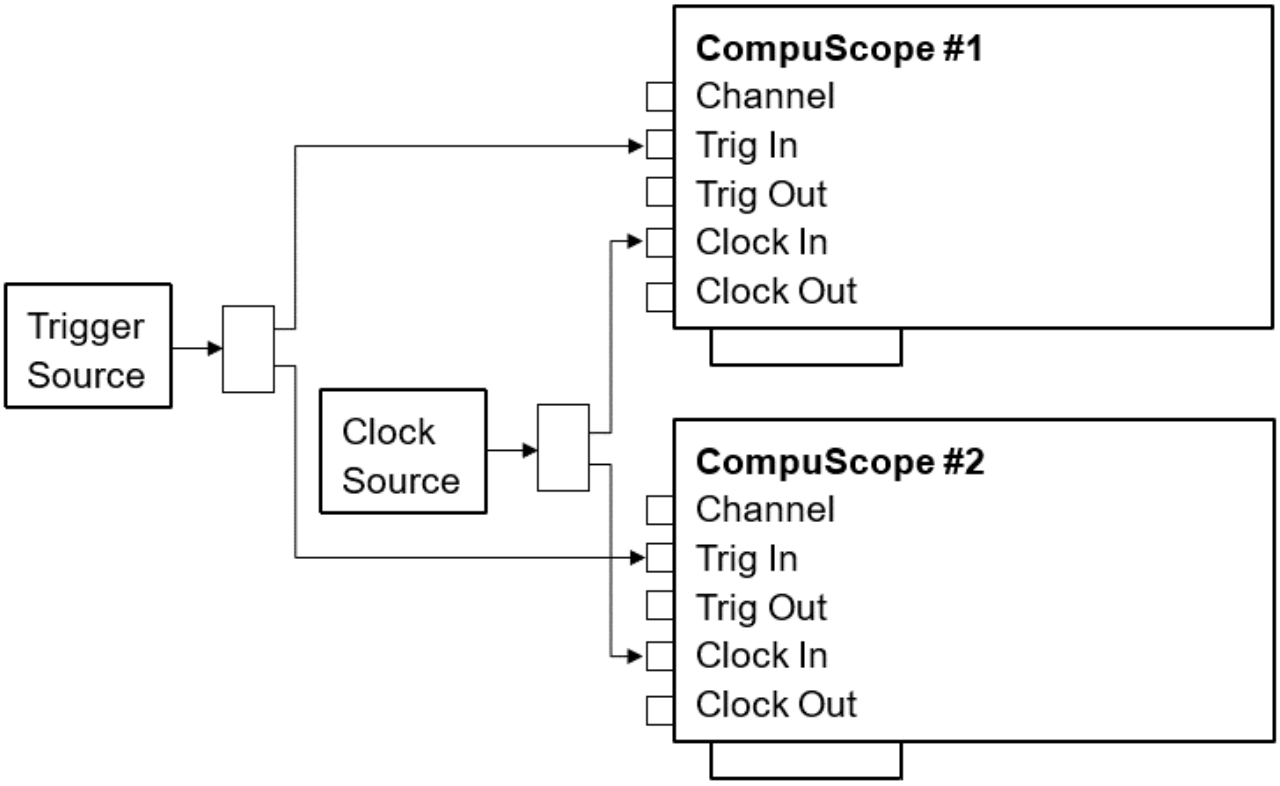


Figure 18: Simultaneous Multi-CompuScope operation using Fan-Out method.

The multiple CompuScope synchronization method of Figure 18 is called the *Fan-out* method and is recommended for most applications. The method may be extended to synchronize more than two CompuScopes using multi-output RF splitters – even across different PC chassis. With enough outputs, the insertion losses of the RF splitters will excessively attenuate timing signal, which will require the addition of booster amplifiers.

A second method of synchronizing multiple CompuScopes is the *Cascade Method*, illustrated in Figure 19. As before, the clocking signal is distributed using the Fan-Out method. However, the trigger signal is introduced only to the first CompuScope. The Trigger Out pulse from this first CompuScope is then used to trigger the second CompuScope, through its Trig In input. As with the Fan-Out method, the Cascade Method can be extended to more than two CompuScopes.

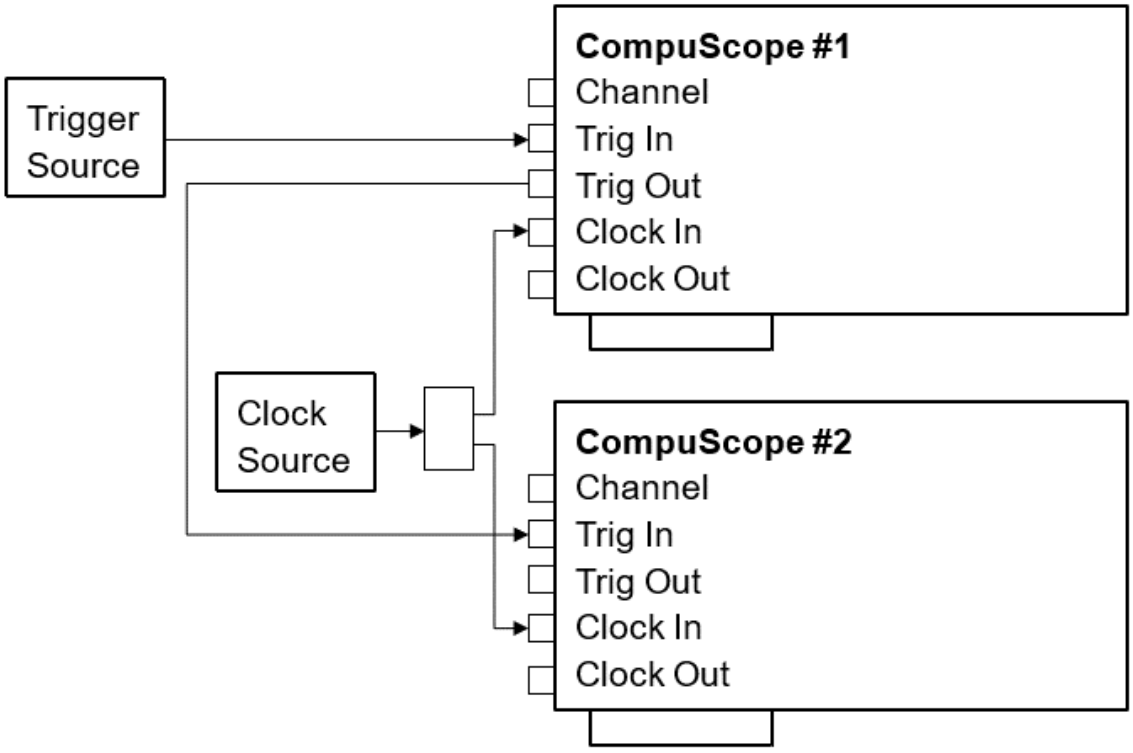


Figure 19: Simultaneous Multi-CompuScope operation using Cascade method.

The Cascade method does not provide as good trigger synchronization as the Fan Out method because of the microsecond-order propagation delay suffered by the Trigger Out pulse. The user can compensate for this delay by performing an external timing calibration, in which a high-speed signal is broadcast to a channel on all CompuScopes through an RF power splitter. This way, an identical and simultaneous signal will be acquired by all CompuScopes. The trigger delays suffered by each successive CompuScope, will show up as shifts for which the user can compensate in software. Since the Trigger Out propagation delay is constant, this external timing calibration need only be performed once.

The main advantage of the Cascade Method manifests in applications with very fast sporadic triggers, as in lightning testing or particle pulse experiments. In these cases, a trigger may take a long time to occur or may occur immediately after the last. Since the CompuScopes are armed to acquire sequentially in the Fan-Out method, it is possible for a trigger to be acquired by one armed CompuScope but missed by another not-yet-armed CompuScope. In the Cascade method, however, if the first CompuScope is armed last, then this errant scenario cannot occur.

12 | CompuScope eXpert Firmware

Lower-speed instruments are generally controlled by an embedded microprocessor – much like a PC. Microprocessors, however, are too slow to allow for the high-speed data transfer required within a CompuScope. Consequently, CompuScopes are controlled by an on-board *Field Programmable Gate Array (FPGA*), which can move data orders of magnitude faster than a microprocessor. FPGAs are loaded with *firmware*, which is configurable digital circuitry. Figure 20 shows the high-level architecture of the CompuScope with the controlling FPGA connected to the digitizing ADC chips, the on-board acquisition memory, and the PCI Express bus. Arrows indicate the direction of data flow to and from these three components.

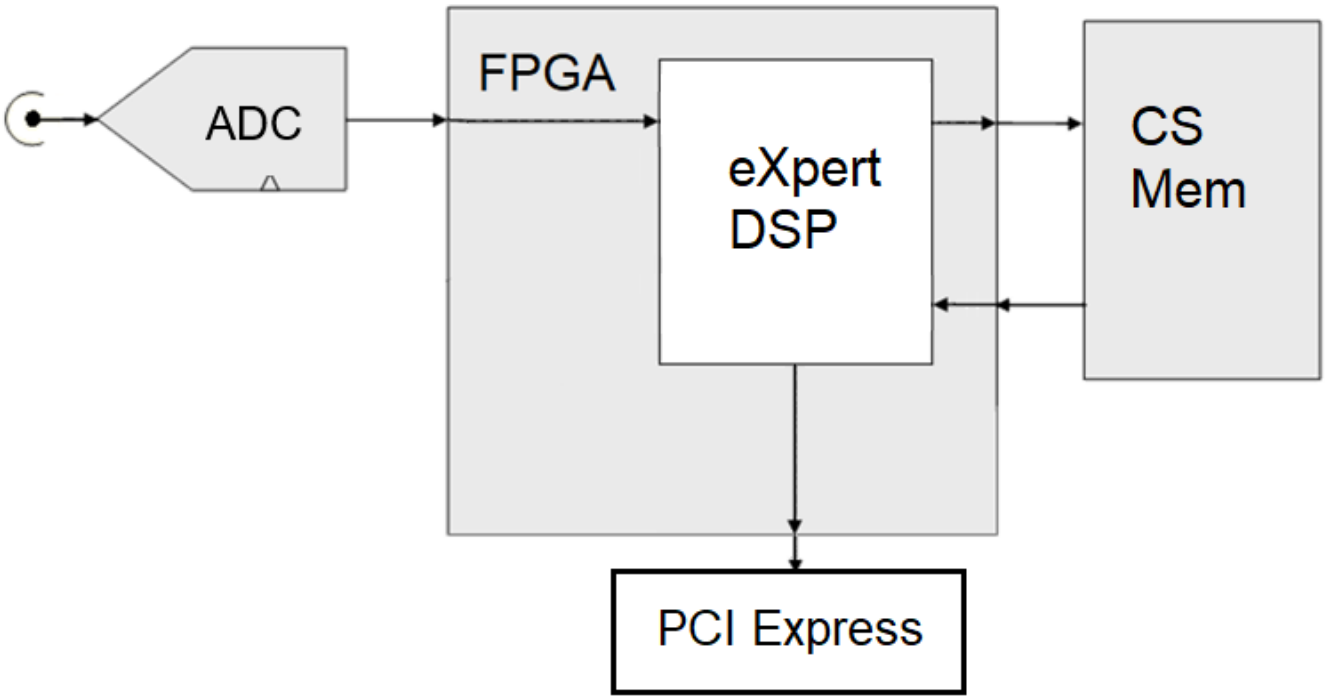


Figure 20: High-level architecture of Gage CompuScope including controlling FPGA. An eXpert DSP firmware module is shown that processes incoming raw waveform data.

In conventional operation, the FGPA moves raw ADC data among the three components. Additionally, Gage CompuScopes can upload various *eXpert firmware modules* that apply *Digital Signal Processing (DSP)* to the raw waveform data. The results of the DSP are then relayed to the user, rather than just raw waveform data.

The on-board DSP applied by CompuScope eXpert firmware provides two advantages. First, the expert firmware frees up the CPU in the host PC from the burden of performing DSP tasks. More importantly, in many cases eXpert DSP can apply significant *data reduction* to the waveform data. The resulting reduced volume of data may be transferred to the host CPU in less time than all the raw waveform data so that the overall measurement rate is increased. Accordingly, eXpert firmware works best when it applies fixed-point data operations with a high data reduction factor. Final floating-point operations may then be applied to the reduced data set within the CPU.

A great example of optimal firmware DSP is CompuScope *eXpert Signal Averaging firmware*. This firmware sums corresponding samples within *N* separately acquired waveforms. Specifically, Sample #1 from Waveform #1 is added to Sample #1 from Waveform #2 et cetera. For example, the firmware may acquire 1000 waveforms and

sum them together to make a single waveform. The resulting data reduction factor is 500-to-1, (since the summed data occupies double space).

The purpose of Signal Averaging is to coherently sum signal features that recur with each trigger while eliminating random noise, which does not. The firmware sums the raw waveforms together but does not calculate the average, which requires division by N. This division is a floating-point operation, which is best done within the CPU. The eXpert Signal Averaging firmware thus illustrates optimal task division, where the eXpert firmware reduces the data with simple fixed-point operations while final floating-point operations are performed by the host CPU.

Other eXpert firmware like *Digital Down Conversion (DDC)* results in significant data reduction factors while firmwares like *Fast Fourier Transform (FFT)* do not. In several cases, custom CompuScope eXpert firmware has been provided to select customers.

13 | CompuScope Streaming Mode Acquisition

A very important eXpert firmware is CompuScope *eXPert PCIe Data Streaming*. As discussed in Section 8, traditional CompuScope acquisition occurs in Memory Mode, where the CompuScope acquires waveform data, stops, and then downloads these data to the host PC. In CompuScope Streaming Mode, data acquisition and data download occur simultaneously. It is therefore possible to stream data indefinitely with no breaks. CompuScope data streaming always requires a *Streaming Target*, that consumes the streamed data. Several possible Streaming targets such as PC RAM, a storage drive, and a Graphical Processing Unit (GPU) are indicated as targets in Figure 21. While a CompuScope is capable of streaming at very high rates through the PCI Express bus, this may not be true of the target. The user must ensure that the target is able to consume data at the stream rate for the streaming acquisition to be successful.

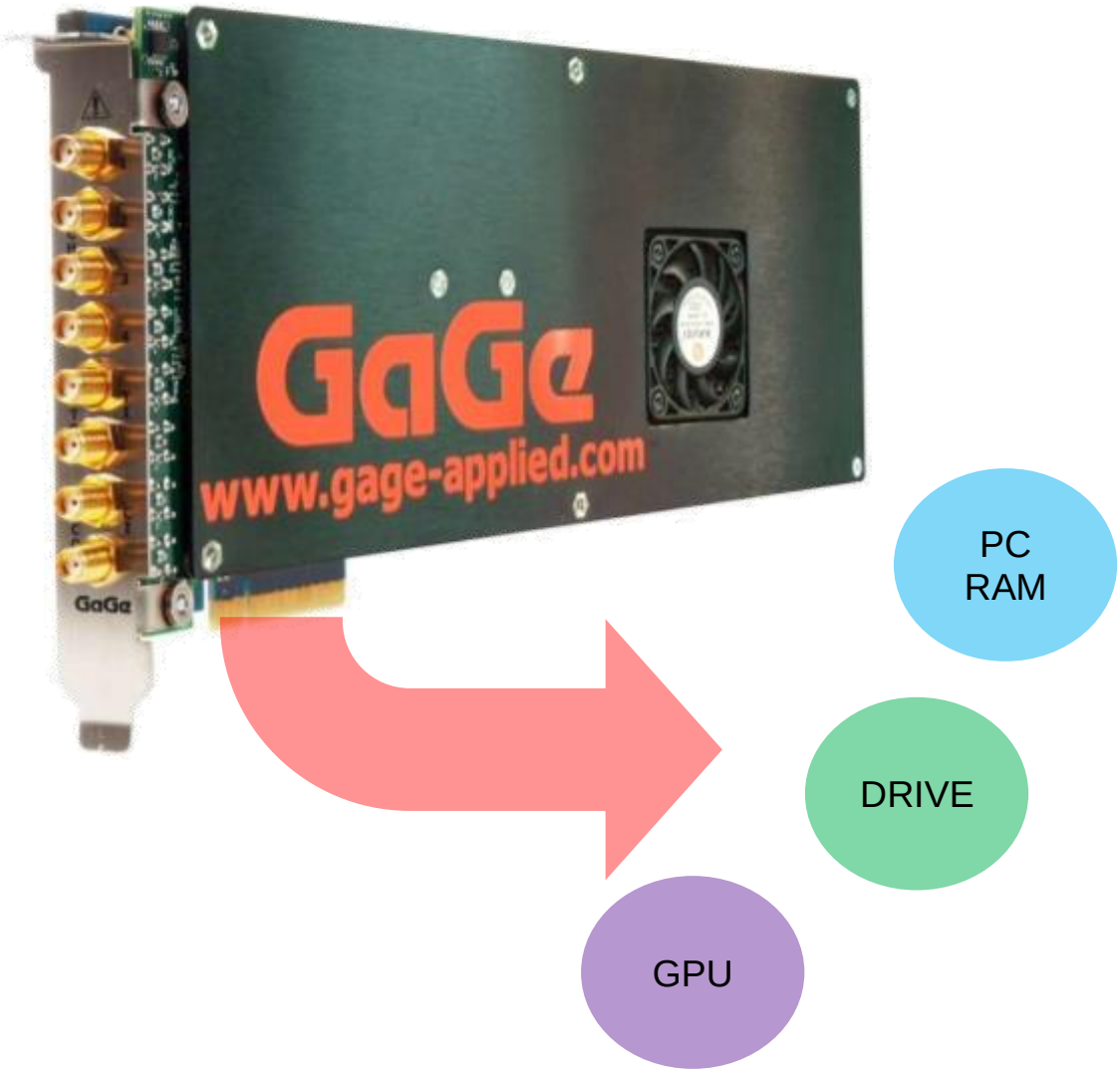


Figure 21: Illustration of data Streaming and potential Streaming Targets: PC RAM, Storage Drive, and a GPU.

The architecture of the streaming acquisition is shown in Figure 22. As shown, the data stream passes through CompuScope on-board memory that serves as a *FIFO (First In First Out)* buffer and which can typically hold a few seconds of data. Small interruptions of data flow through the PCIe bus will cause the on-board memory to fill up a little. These data will be flushed out once the interruption subsides. In cases where the target cannot keep up with the stream flow rate, the on-board memory will eventually fill up, which will cause a FIFO OVERFLOW error to occur. This error therefore indicates that the target is not fast enough.

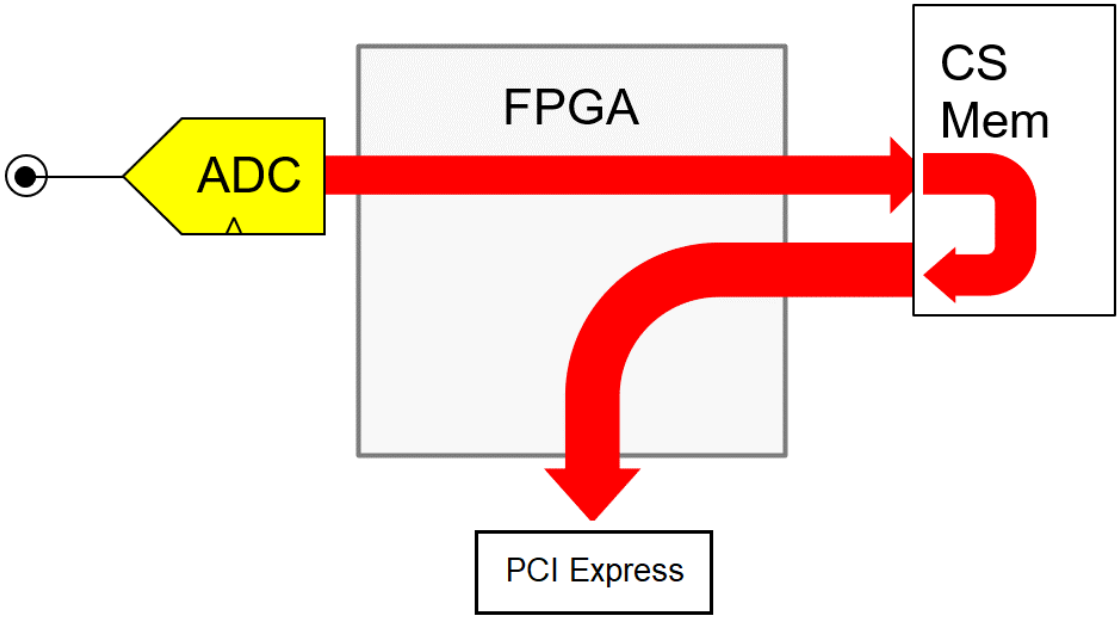


Figure 22: Architecture of Streaming Data acquisition.

Figure 23 shows how the software operates in a streaming acquisition. Low-level CompuScope drivers operate at the kernel level and controlling software applications operate at the Application level. The primary streaming data targets consists of two toggling buffers illustrated in Figure 23 that are accessible from both the Application and Kernel levels. These buffers are special and must be composed of physical (not virtual), contiguous memory that is locked down so that its physical location does not change over time. Gage provides special memory allocation sub-routines for toggling buffers and their size typically cannot exceed a few Megabytes, which is no limitation in practice. In contrast to these buffers, Application-level buffers, which can be much larger, allocate *virtual RAM* using conventional sub-routines. Virtual RAM buffers are not contiguous and can change physical location over time.

During a streaming acquisition, the CompuScope fills up one of the toggling buffers while the contents of the other buffer are transferred to the final target. Once both operations are complete, the buffers are swapped and the process repeats. The two buffers toggle in this way repeatedly throughout the acquisition. If the final target is unable to consume the data from one toggling buffer faster than the CompuScope fills the other toggling buffer with new waveform data, then the CompuScope FIFO memory will overflow, producing an error. Conversely, if this error does not occur, the user may be confident that all streamed data were transferred to the target with no data loss.

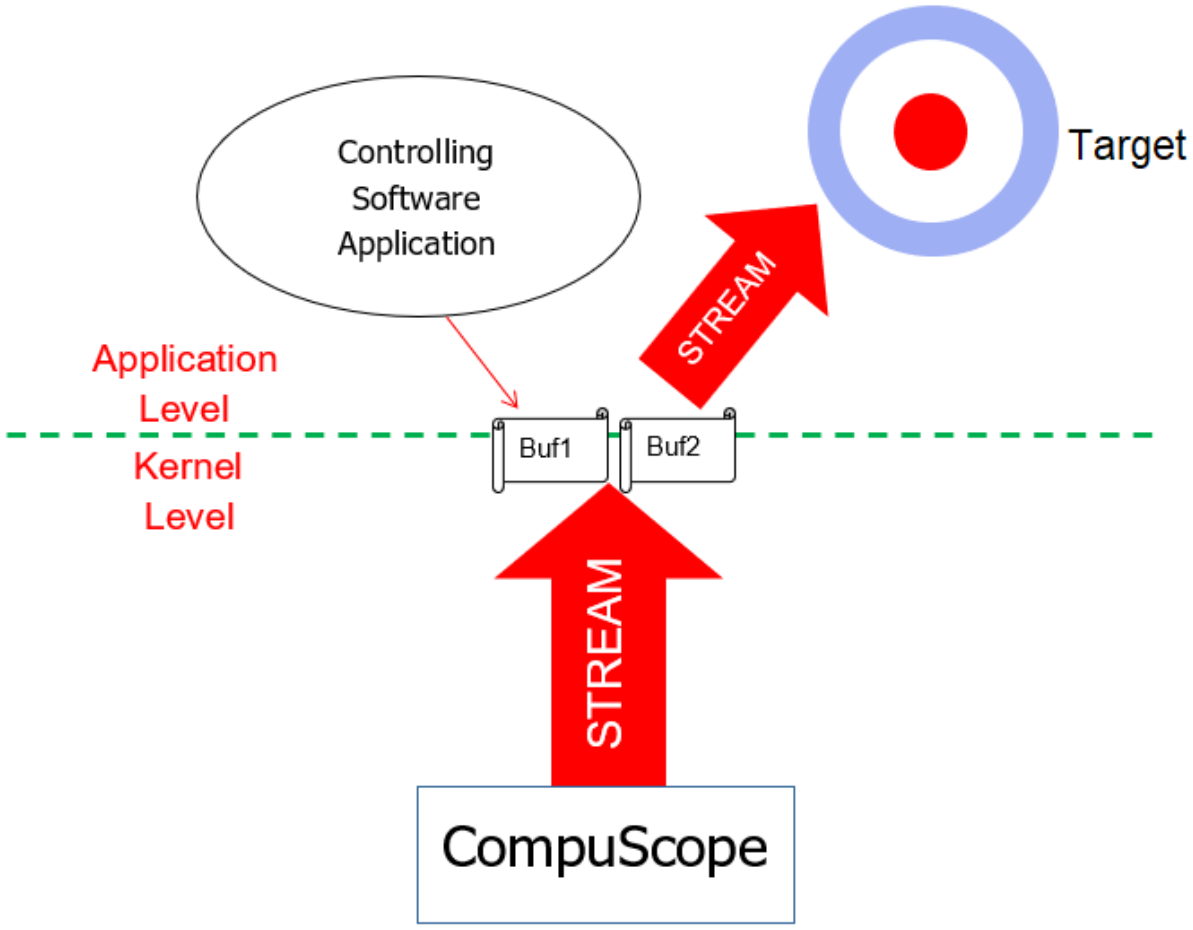


Figure 23: Streaming Software Architecture

The high-performance data transfer operations associated with a CompuScope streaming acquisition limits the user to controlling a streaming acquisition from the C, C++ or C# programming environments. While Gage provides support for CompuScope Memory Mode acquisition in other programming environments like LabVIEW or MATLAB, these languages do not provide sufficient data transfer or processing performance to enable high-speed streaming targets and so these SDKs are not provided with steaming examples.

Users must ensure that their CompuScope data streaming rate does not exceed the maximum speed at which the PCI Express bus can sustain data transfer. For example, the Gen3 X8 PCI Express bus has a maximum theoretical data transfer rate of 8 GigaBytes/sec, however, Gen3 CompuScopes indefinitely sustain rates of up to 5 or 6 GB/s, depending upon the system. A 2-channel streaming acquisition on a 16-bit CompuScope at a sampling rate of 1 GS/s creates an aggregate streaming data rate of:

Stream Rate = 2 Channel X (2 Bytes/Sample) X 1 GS/s = 4 GB/s.

Since this 4 GS/s is below the 5 or 6 GB/s, this acquisition can be indefinitely sustained by the CompuScope. Doubling the number of channels to four, however, brings the aggregate rate to 8 GS/s, which exceeds the limit and so is unsustainable.

Continuous streaming acquisition acquires data 100% of the time and is initiated by a single trigger. Alternately, a streaming acquisition can be in Multiple Record Mode where multiple triggers are accepted throughout the streaming action after which only short waveforms are acquired. Consider an example radar application where the user acquired 20 microsecond waveforms at a trigger rate of 10 kHz – for a trigger interval of 100

microseconds. In this case, data are acquired only 20 µs/100 µs = 20% of the time. For a 4-channel, 16-bit, 1 GS/s acquisition, this Multiple Record streaming data rate would be:

Stream Rate = 20% X 4 Channel X (2 Bytes/Sample) X 1 GS/s = 1.6 GB/s.

Consequently, while 100% continuous streaming cannot be sustained under these conditions, usage of Multiple Record streaming reduces the rate by 5X, which can be sustained indefinitely.

Once the user has confirmed that their CompuScope can sustain the proposed streaming acquisition rate, they must then ensure that their streaming target can keep up. The most obvious streaming target is PC RAM, which is very fast and can always keep up. While the two toggling buffers of Figure 23 are limited in size, the size of a regular application RAM buffer can be hundreds of Gigabytes. Streaming to PC RAM, therefore, easily extends CompuScope acquisition memory from a few Gigabytes of on-board CompuScope memory to several hundred Gigabytes of PC RAM.

CompuScope waveform recording capacity may be extended even further by using a high-speed storage drive system, whose capacity can be 100 Terabytes or more, which allows continuous recording for several hours. The user must ensure that the sustained write speed of the storage disk system is sufficient. This may require construction of a RAID drive system, in which several drives are configured in parallel to increase throughput. Gage Recording Systems are delivered with pre-configured storage drive arrays so that user can immediately stream to multi-Terabyte drives with no programming required.

14 | CompuScope DSP Operation

Some users need to perform *Digital Signal Processing (DSP)* on raw waveform data as they are being acquired. In most cases, they can dispense with the raw data. Sometimes, the processed results are required to make decisions during acquisition, which defines a *real-time application.*

Section 12 describes how eXpert DSP may be applied to raw waveform data within a CompuScope's FPGA. A second option is to perform DSP on a host PC's CPU. A third option is to employ a *Graphical Processing Unit (GPU)*. Designed for the parallel processing of vectors associated with high-speed graphics, GPU cards may be employed to parallel process CompuScope waveforms. The Gage C SDK provides sample programs that will operate NVidia GPU cards using NVidia's *CUDA library*, which the user can easily modify to implement their own DSP calculations.

.

15 | Other Gage Instruments

In addition to Gage CompuScope digitizers, Gage products include PCs in three formats: Rackmount PCs, Desktop PCs, and Lunchbox PCs. Gage expansion chassis are also available that may be controlled from host PC or laptops through cabled interfaces, such as Thunderbolt. Customers can purchase turnkey Gage systems with all CompuScope hardware and software installed and activated. See your Gage sales agent for more details.

Gage CompuScope Recording Systems are equipped with fast drive systems that are qualified with sufficient speed and capacity for a customer's recording requirement. Recording Systems may be configured to allow continuous recording of waveform data from multiple CompuScope channels for hours or even days.

Gage can provide a*nalog downconverter* units that are positioned in between raw RF communications signals and CompuScope channel inputs. As the name suggests, the down-converter shifts the carrier frequency of a communications signal down to a lower frequency that is accessible to the CompuScope. For example, a raw communications signal with a bandwidth of 1 GHz from a receiving antenna might contain frequencies between 21 and 22 GHz. The down-convertor can shift this frequency spectrum down so that it contains lower frequencies between 0 and 1 GHz. This transformation has the effect of shifting down the high carrier frequency while not altering the information content of a communications signal. Now that its frequency spectrum sits below 1 GHz, several CompuScope models may be employed to acquire it.