www.tek.com/keithley

S540 Parametric Test System

High-Voltage Library Reference Manual

S540-908-01 Rev. A / September 2017

S540-908-01A

KEITHLEY

A Tektronix Company

S540

High-Voltage Library (HVLib)

Reference Manual

Document number: S540-908-01 Rev. A / September 2017

Safety precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with nonhazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the user documentation for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product warranty may be impaired.

The types of product users are:

Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

Maintenance personnel perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

Service personnel are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Keithley Instruments products are designed for use with electrical signals that are measurement, control, and data I/O connections, with low transient overvoltages, and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II (as referenced in IEC 60664) connections require protection for high transient overvoltages often associated with local AC mains connections. Certain Keithley Instruments measuring instruments may be connected to mains. These instruments will be marked as category II or higher.

Unless explicitly allowed in the specifications, operating manual, and instrument labels, do not connect any instrument to mains.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before: connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

For safety, instruments and accessories must be used in accordance with the operating instructions. If the instruments or accessories are used in a manner not specified in the operating instructions, the protection provided by the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories. Maximum signal levels are defined in the specifications and operating information and shown on the instrument panels, test fixture panels, and switching cards.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as protective earth (safety ground) connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

The ⚠ symbol on an instrument means caution, risk of hazard. The user must refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

The symbol on an instrument means warning, risk of electric shock. Use standard safety precautions to avoid personal contact with these voltages.

The symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The symbol indicates a connection terminal to the equipment frame.

If this Hg symbol is on a product, it indicates that mercury is present in the display lamp. Please note that the lamp must be properly disposed of according to federal, state, and local laws.

The **WARNING** heading in the user documentation explains dangers that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in the user documentation explains hazards that could damage the instrument. Such damage may invalidate the warranty.

The **CAUTION** heading with the ⚠ symbol in the user documentation explains hazards that could result in moderate or minor injury or damage the instrument. Always read the associated information very carefully before performing the indicated procedure. Damage to the instrument may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits — including the power transformer, test leads, and input jacks — must be purchased from Keithley Instruments. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. The detachable mains power cord provided with the instrument may only be replaced with a similarly rated power cord. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keithley Instruments to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley Instruments office for information.

Unless otherwise noted in product-specific literature, Keithley Instruments instruments are designed to operate indoors only, in the following environment: Altitude at or below 2,000 m (6,562 ft); temperature 0 °C to 50 °C (32 °F to 122 °F); and pollution degree 1 or 2.

To clean an instrument, use a cloth dampened with deionized water or mild, water-based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., a data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

Safety precaution revision as of June 2017.

Table of contents

Section 1

General information

In this section:

High-Voltage Library overview

The Keithley High-Voltage Library (HVLib) is a set of commands you can use to make measurements on an S540 Power Semiconductor Test System using the Keithley Test Environment (KTE) software.

You can use these commands in two- and three-terminal measurement applications to measure capacitance-voltage (C-V), calculate compensation constants, do open, load, and short compensation, and do breakdown voltage tests.

This manual contains general information about doing basic and advanced high-voltage C-V measurements and detailed descriptions of the HVLib commands, including usage prototypes, parameter definitions, and examples.

Manual structure

The content of this manual is organized into the following sections:

- General information (on page 1-1) (this section)
- Introduction to high-voltage C-V measurements with HVLib (on page 2-1)
- Advanced high-voltage C-V measurements (on page 3-1)
- High-Voltage Library command reference (on page 4-1)

Contact information

If you have any questions after you review the information in this documentation, please contact your local Keithley Instruments office, sales partner, or distributor. You can also call the corporate headquarters of Keithley Instruments (toll-free inside the U.S. and Canada only) at 1-800-935-5595, or from outside the U.S. at +1-440-248-0400. For worldwide contact numbers, visit the Keithley Instruments website (http://www.tek.com/keithley).

Introduction to HV C-V measurements with HVLib

In this section:

Introduction

To bias transistors to high voltage and measure capacitance, any high-voltage capacitance-voltage (C-V) measurement technique must use bias tees to protect instrumentation and devices under test (DUTs) from damage.

Bias tees allow you to mix high-voltage DC bias voltage with an AC signal. A drawback of using bias tees is that it causes degradation of the AC pathway and increased measurement errors. To solve this problem, you can apply compensation factors to measurements to negate the effect of bias tees in high-voltage measurement applications.

The S540 Parametric Test System High-Voltage Library (HVLib) gives you the ability to make basic high-voltage capacitance-voltage measurements with fixed system-level compensation factors and more accurate, advanced high-voltage C-V measurements using preprogrammed system-level compensation factors and user-generated device-level compensation factors.

CMTRs in the S540 system

A capacitance meter (CMTR) is a vector impedance meter that evaluates AC impedance of a device, including both real and imaginary (reactive) parts of the complex impedance.

The AC drive signal is supplied to the device under test (DUT) from the high side of the CMTR, and an automatically balanced bridge on the low side of the CMTR measures the amplitude and phase of the current.

The S540 Power Semiconductor Test System supports several configurations:

- System with no capacitance meters (CMTRs)
- System with one low-voltage CMTR (CMTR1)
- System with one low-voltage CMTR (CMTR1) and one high-voltage CMTR (CMTR2)

For more information about complex AC impedance, see AC impedance (on page 3-1).

Basic and advanced commands

The most commonly used commands are provided in the HVLib in both basic and advanced versions. These commands are described in the following table.

Basic command	Advanced command	Purpose
`hvcv_3term_basic`	`hvcv_3term`	This command measures output capacitance (C_{oss}), input capacitance (C_{iss}), or short-circuit reverse transfer capacitance (C_{rss}) of three-terminal devices.
`hvcv_sweep_basic`	`hvcv_sweep`	This command does a high-voltage capacitance-voltage (C-V) sweep.
`hvcv_test_basic`	`hvcv_test`	This command makes a high-voltage capacitance-voltage (C-V) measurement at a single frequency.

Differences between basic and advanced commands

The main difference between the two versions of these commands is that several parameters are fixed in the basic commands to make it simpler to make high-voltage capacitance-voltage (C-V) measurements. These parameters and their fixed values are:

Parameter	Description	Fixed value
Dut	Device under test; valid options: • `dut` = Test the DUT itself with the high-voltage capacitance meter (CMTR) • `open` = Characterize the open device using the high-voltage CMTR; this can be done with the chuck down (pins not in contact with the device) • `short` = Characterize the short device using the high-voltage CMTR • `load` = Characterize the load device using the high-voltage CMTR • `shortEx` = Characterize the short device using the low-voltage capacitance meter (CMTR); this data is used to do `CompShort` compensation of the *loadEx* device • `loadEx` = Measure the load device using the low-voltage CMTR to get the expected value of the `loadEx` device • `openEx` = Characterize the open device using the low-voltage CMTR; this data is used to do `CompOpen` compensation of the `loadEx` device	`dut`
Comp_mode	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *dut* parameter is set to anything other than `dut`) • `CompOpen` • `CompShort` • `CompLoad` • `CompOpenLoad` • `CompShortOpen` • `CompShortLoad` • `CompShortOpenLoad`	`compNone`
doComp	Specifies whether to do system-level compensation: • `0` = Do not do system-level compensation • `1` = Do system-level compensation • 2 = Do user-created compensation (not available for hvcv_3term and hvcv_3term_basic commands)	`1`
doRetest	Specifies whether to remeasure compensation data: • `0` = Do not remeasure compensation data • `1` = Remeasure compensation data once, then reuse that measurement in any additional calls	`1`

For a thorough discussion of advanced high-voltage C-V measurement topics, see Advanced high-voltage C-V measurements (on page 3-1). For detailed descriptions of each of the HVLib commands, see High-Voltage Library command reference (on page 4-1).

Advanced high-voltage C-V measurements

In this section:

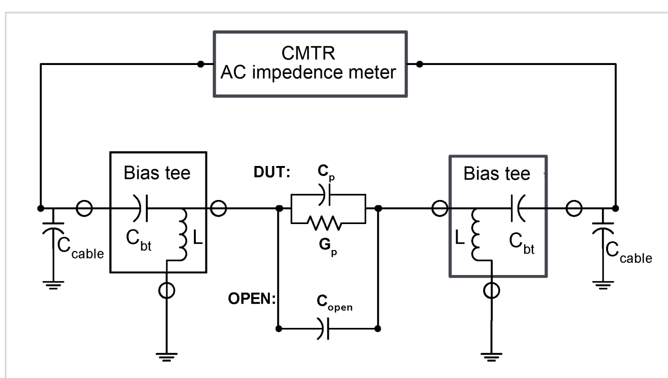
AC impedance

The ratio of the complex AC voltage vector to the current vector provides a complex impedance, which is then converted to specified values using the selected impedance model. The most common impedance models are parallel capacitance and conductance (C_p and G_p) and magnitude and series phase (Z and theta). The High-Voltage Library (HVLib) software automatically makes the calculations to convert the raw CMTR data to the model you choose.

Bias tees and compensation in a two-terminal AC model

Capacitance-voltage (C-V) measurements made with bias tees in the circuit have significant error, and this error must be addressed using compensation factors. For example, if capacitance measurements are made using Keithley Instruments S530-RBT-3KV bias tees with no compensation, measurement error can be as high as three to four percent.

Figure 1: Two-terminal high-voltage capacitance measurement



An impedance analysis shows that the measured value of the impedance (Z_{meas}) can be related to impedance of the device under test (Z_{dut}) using following equation:

$$Z_{meas} = k \times Z_{dut} \,||\, Z_{open} + Z_{short}$$

Where:

- Z_{meas} = Measured impedance
- k = Gain error, $1 + (C_{cable}/C_{bt})^2$
- Z_{dut} = Actual device impedance
- Z_{open} = Measured open parasitic impedance
- Z_{short} = Measured impedance of short device, $2 \times (1/jw \times C_{bt}) \times (1+ C_{cable}/C_{bt})$

You can use this equation to build a compensation model that calculates device capacitance and removes the effects of parasitic capacitances and bias tees. This calculation requires values for the open device (Z_{open}) and short device (Z_{short}) and the value of the gain compensation (k).

Using this model, the effect of the bias tees in the circuit is the same as the gain error. Compensation for the gain error requires measurement of a load standard. In this context, a load standard is a device under test (DUT) of approximately the same impedance as the one you are measuring.

Gain error is determined by the ratio of the cable capacitance to the bias tee capacitance, which should not change much across the range of frequencies. The S540 system can have individual compensation constants for any requested frequency, in the event that the ratio varies.

Three-terminal capacitance measurements

Three-terminal capacitance measurements (C_{iss}, C_{oss}, and C_{rss}) are some of the most challenging, yet commonly used measurements in power device characterization. Making C_{iss}, C_{oss}, and C_{rss} measurements allows you to evaluate required transistor switching characteristics such as speed, energy, and charge.

This type of measurement historically has been done using bench setups. However, Keithley Instruments has developed a procedure for the S540 system that uses its high-voltage matrix to automate these measurements. Keithley High-Voltage Library (HVLib) and Linear Parametric Test Library (LPTLib) commands are used with the Keithley Test Environment (KTE) Software to automate this procedure.

C_{iss}, C_{oss}, and C_{rss} measurements are typically made in the off state, with gate voltage at 0 V DC and at high drain voltage. A bias tee must be connected to each terminal because varying high-voltage biases are applied to each terminal (gate, drain, and source). This connection configuration differs from the standard two-terminal configuration shown in the figure in Bias tees and compensation in a two-terminal AC model (on page 3-1).

Input (C_{iss}) and output (C_{oss}) capacitances are measured in a similar way. Each of the three terminals (gate, drain, and source) must have an independent DC bias. In the AC frequency domain, two terminals are connected to each other in the high-voltage matrix and impedance is measured against the third terminal.

For example, for C_{iss}, drain voltage is usually high, the source is DC grounded, and the gate voltage ensures the off state of the transistor. Then the gate is AC-tied to the sense (low) terminal of the capacitance meter (CMTRL) and the source pin is AC-tied with the drain to the high (AC-drive) side of the CMTR (CMTRH).

Guard challenge for Crss

C_{rss} capacitance measurement is more difficult than the measurement of C_{iss} or C_{oss}. Impedance measurement is done between the gate and drain with AC guarding at the source pin.

In AC guarding, the guarded pin is held as close as possible to AC ground by providing a low-impedance connection to the AC ground, or by applying an active AC signal to the guarded pin. This ensures minimal AC voltage on the guarded pin. However, the ability to minimize AC voltage on the guarded pin is limited by interconnects, and becomes progressively less effective at high frequencies.

The S540 system guards the source by connecting it directly to ground (GND), which bypasses the bias tees in the system. This technique works sufficiently at 100 kHz, but has reduced accuracy at 1 MHz (see the S540 specifications for details). This only affects the quality of the C_{rss} measurement.

Automated high-voltage C-V measurements

Three-terminal capacitance measurements require careful application of complex connections to the capacitance meter (CMTR), bias tees, and DC instruments. The S540 3-kV high-voltage matrix facilitates software-controlled connections, automating these connections for high-voltage capacitance-voltage (C-V) measurements.

The High-Voltage Library (HVLib) `hvcv_3term` command uses these software-controlled connections to make automated three-terminal capacitance measurements. You can use this command without changes, or you can copy it under a different name and customize it for your application. It can be used to measure C_{iss} and C_{oss} parameters and individual capacitances (C_{rss}). The routine uses system-level compensation when the *`doComp`* parameter is enabled.

The `hvcv_3term` command can do device-level compensation, including most of the known compensation models. For example, to run ShortOpenLoad device-level compensation, you must first collect compensation data from the open, short, and load devices.

High-voltage C-V compensation methods

S540 systems with two capacitance meters (CMTRs) can make high-voltage capacitance-voltage (C-V) measurements using two different compensation methods:

- System-level compensation (on page 3-3)
- Device-level compensation (on page 3-8)

The following topics describe each of these methods. For examples of how these methods can be used to make high-voltage C-V measurements, see High-voltage C-V usage scenarios (on page 3-10).

System-level compensation

System-level compensation applies compensation factors to negate the effect of bias tees in the system. It uses a single, preprogrammed set of compensation factors for any pin combination in the system. System-level compensation factors are set at the factory and stored in the system. They can also be recreated at your site if necessary.

System-level compensation is not as accurate as device-level (user-specified) compensation because it does not account for the subtle pin-to-pin differences caused by probe cards, cabling, or test fixtures in the system.

To use this level of compensation, set the *doComp* parameter to 1 or 2 in the hvcv_test, hvcv_sweep, or hvcv_3term high-voltage library (HVLib) commands.

System-level compensation factors are stored in the cvCALsystem.ini file. If system-level compensation is enabled, the hvcv_test, hvcv_sweep, or hvcv_3term commands use hvcv_intcg function instead of the standard Linear Parametric Test Library (LPTLib) intgcg function.

The hvcv_intcg command uses one of two different sets of compensation factors:

- When the *doComp* parameter is set to 1, the hvcv_intcg command uses compensation factors stored in the cvCALsystem.ini file.
- When the *doComp* parameter is set to 2, the hvcv_intcg command uses compensation factors stored in the cvCAL.ini file. This setting does run-time ShortOpenLoad compensation of the raw data.

Procedure overview

Following is a summary of what happens during system-level compensation:

- Preprogrammed compensation factors are retrieved, or new compensation factors can be created at run-time using a custom fixture and load standard (probe card with connection for the discrete capacitors). The default procedure uses preprogrammed compensation factors.
- Compensation factors are stored in the opt/kiS530/cvCAL.ini file.
- The hvcv_intgcg command reads the compensation factors, and if appropriate, does ShortOpenLoad compensation.

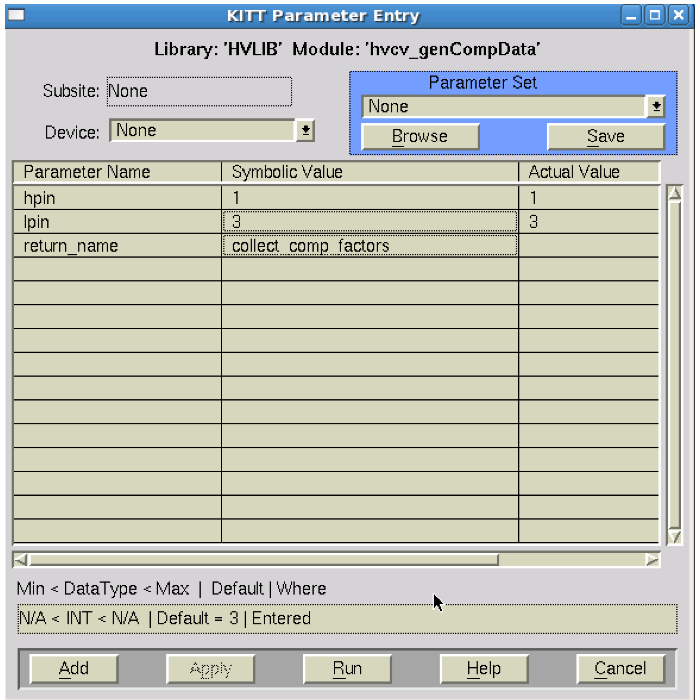
System-level compensation can be enabled or disabled by setting the *doComp* parameter to 1 or 0 in the following commands: hvcv_test, hv_sweep, and hvcv_3term.

For examples of specific usage scenarios, see High-voltage C-V usage scenarios (on page 3-10).

Recreating system-level compensation factors

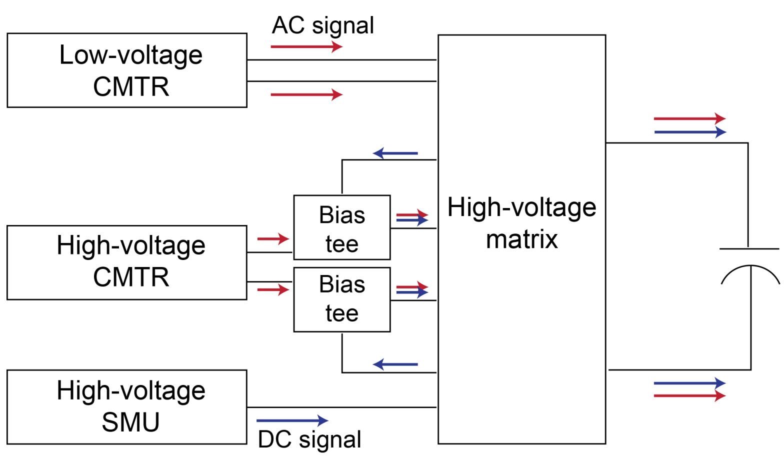
To recreate system-level compensation factors, use the hvcv_genCompData command. This command prompts you to connect open, short, and load devices to a selected pin pair. You must use a custom fixture or probe card that allows you to insert a load standard and short device. The following figure shows the selected pin-pair of pin 1 and pin 3. For a load-device or load standard, Keithley recommends using a discrete capacitor with a range of 100 pF to 1 nF.

Figure 2: Compensation data collection function: hvcv_genCompData



System-level compensation uses the algorithm shown in the following figure. If the S540 system has a high-voltage capacitance-voltage (C-V) configuration (high-voltage capacitance meter (CMTR) with bias tees), it also has a low-voltage CMTR, as shown in the following figure.

Figure 3: Using a low-voltage CMTR for high-voltage compensation



The high-voltage CMTR is connected to the high-voltage matrix through bias tees, which allows a DC bias voltage of up to 3 kV applied to the device.

The low-voltage CMTR is connected directly to high-voltage matrix with no bias-tees and does not exhibit AC signal degradation caused by bias-tees. The low-voltage CMTR is protected from high voltages by protection modules (not shown in the previous figure).

When the `hvcv_genCompData` command is collecting compensation data, the high-voltage and low-voltage CMTRs characterize the open, short, and load devices. The low-voltage CMTR makes the measurements and calculates compensation factors for frequencies from 1e4 to 2e6.

The `hvcv_genCompData` routine then writes the compensation factors for each frequency (10 kHz to 2 MHz) to the `opt/kiS530/cvCAL.ini` file. The following figure shows an example of the compensation factors for the 100 KHz and 1 MHz frequencies. In this figure, `ShortCs` and `ShortRs` characterize the impedance of the short. Short resistance (`ShortRs`) should be below 10 Ω. Values for `OpenCp` are usually below 10 pF. Gain compensation factors (`GainR` and `GainX`) are real and imaginary (reactive) components of the load compensation. `GainX` is usually 0.10 or below, and `GainR` is close to 1.00 (from 0.95 to 1.10). Note that units of `GainX` and `GainR` are dimensionless.

Figure 4: Open, short, and load compensation factors in cvCAL.ini

```
#Created 100kHz 8/16/16
#Load is 1nF
<HVCV100000>
ShortCs=7.3527e-08
ShortRs=3.82049
OpenCp=1.40695e-11
OpenGp=2.27076e-05
GainR=0.988447
GainX=-5.94805e-05

#Created 1MHz 8/16/16
#Load is 1nF
<HVCV1000000>
ShortCs=-3.4795e-09
ShortRs=5.97945
OpenCp=3.07177e-12
OpenGp=3.59237e-06
GainR=0.962777
GainX=0.125604
```

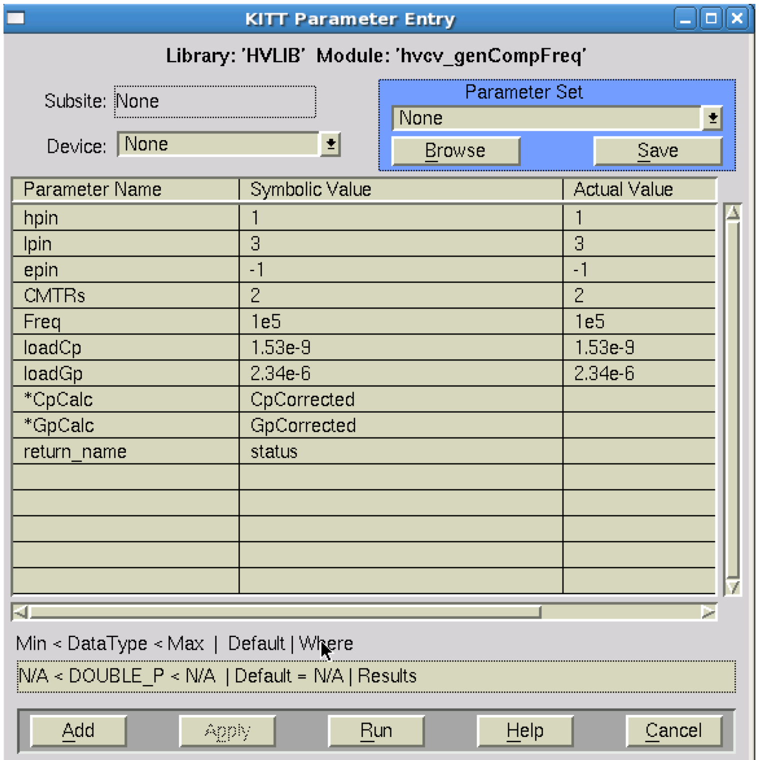
Generating compensation factors for a single frequency

The S540 High-Voltage Library (HVLib) `hvcv_genCompFreq` command generates compensation factors for a single frequency. You can use this command to debug the compensation algorithms, and it prompts you to insert the open, short, and load devices when appropriate.

The `hvcv_genCompFreq` command has two modes of operation based on the number of capacitance meters (CMTRs) that are available in the system. If the *`CMTRs`* parameter is set to `1`, the routine only uses the high-voltage CMTR and uses values specified by the *`loadCP`* and *`loadGP`* parameters for the load device. If the *`CMTRs`* parameter is set to `2`, the second CMTR (low-voltage CMTRL) is used to create reference load data and provided data for the load is ignored.

The following figure shows example parameters for the command; definitions of the parameters follow the figure.

Figure 5: Compensation data collection function: hvcv_genComp_Freq



This example uses the following parameters:

- *`hpin`*: Pin connected to the capacitance meter high-side (for low-voltage CMTR1H and high-voltage CMTR2H)
- *`lpin`*: Pins connected to the CMTR low-side (for low-voltage CMTR1L and high-voltage CMTR2L)
- `epin`: Extra pin connected to CMTR high-voltage guard terminal (CMTR2G)
- *`CMTRs`*: The number of CMTRs to use for compensation measurements. If the number is 2, the low-voltage CMTR and the high-voltage CMTR are used. If the number is `1`, only the high-voltage CMTR is used, and the *`loadCp`* and `loadGp` are used for the load.
- *`Freq`*: Specified frequency
- *`loadCp`* and *`loadGp`*: Independently known values of the load device, expressed as capacitance and conductance, according to the parallel model representation
- *`CpCalc`* and *`GpCalc`*: Values of the load device after measurement and compensation; these values should be very close to the loadCp and loadGp values when the value of the *`CMTRs`* parameter is `1`
- *`Return_name`*: Status of the measurement; negative value for failure

Device-level compensation

Device-level compensation applies compensation factors to negate the effect of bias tees, cabling, probe cards, or test fixtures in the system. You create the compensation factors just before testing a device. This results in more accurate high-voltage capacitance-voltage (C-V) measurements.

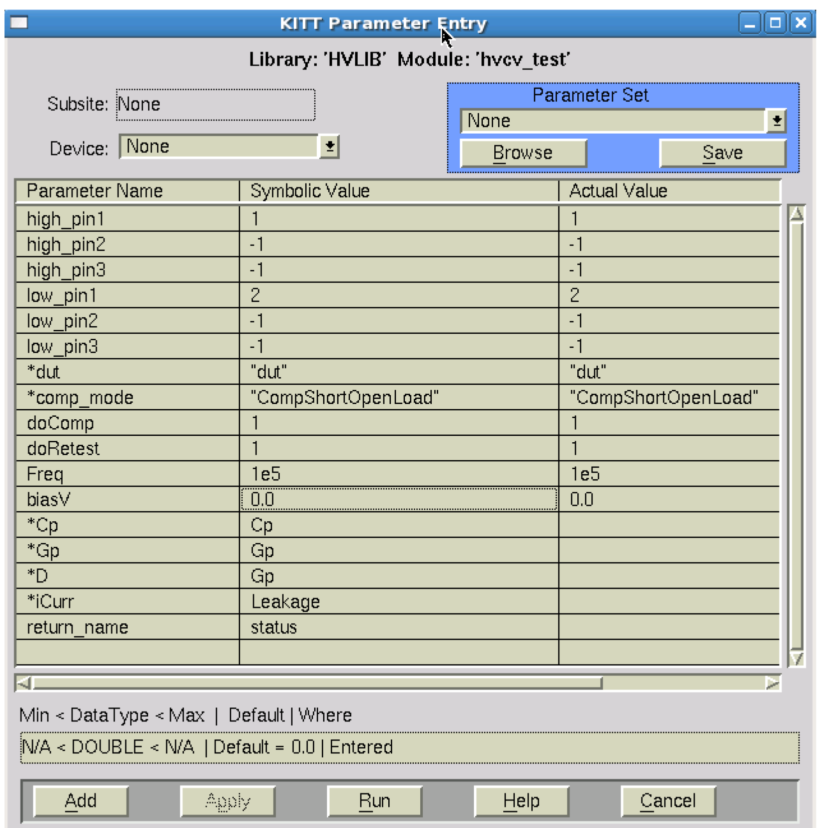
There are several ways to acquire device-level of compensation factors, discussed in the following topics. For examples of specific usage scenarios, see High-voltage C-V usage scenarios (on page 3-10).

Device-level compensation allows you to do compensation for each individual pin-pair at run time during automated testing on the wafer. Available compensation methods are Open, Short, Load, OpenLoad, ShortOpen, and ShortOpenLoad.

Open measurement is done when the chuck is down. Short measurements can be made on any metal surface or connected pads on the wafer.

Selection of the known load device on the wafer can be difficult because it requires C-V characterization of the capacitor on the wafer with no bias tee connection. If your S540 system is configured with a high-voltage capacitance meter (CMTR) and a low-voltage CMTR, you can use the low-voltage CMTR (CMTR1) to measure the expected value of the load.

Figure 6: Example of the hvcv_test in KITT



Procedure overview

The S540 High-Voltage Library (HVLib) contains several device-level commands that you can configure for capacitance measurement:

- `hvcv_test`: Makes a two-terminal single DC bias measurement (the figure in Device-level compensation (on page 3-8) shows example settings for this command)
- `hvcv_sweep`: Collects capacitance-voltage (C-V) data and does a DC bias sweep
- `hvcv_3term`: Measures output capacitance (C_{oss}), input capacitance (C_{iss}), and short-circuit reverse transfer capacitance, common source (C_{rss}) of three-terminal devices

These device-level, user-configured commands are structured similarly to allow for device-level compensation.

The following example of device-level ShortOpenLoad compensation shows how device-level compensation can improve data accuracy. This is useful in situations where impedance of the device is small (large capacitances) or for larger frequencies (for example, 1 MHz).

This example makes a two-terminal capacitance measurement and does full ShortOpenLoad compensation using the `hvcv_test` command.

1. Select a short device with pins 1 and 2 connected.
2. Execute the following test:
```
doComp = 1;
doRetest = 0;
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "short", "CompNone", doComp,
doRetest, Freq, DcBias, Cp, Gp, Gp, Leakage);
```
3. On the load device (select a device with an impedance close to the tested impedance), execute the following test:
```
doComp = 1;
doRetest = 0;
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "load", "CompNone", doComp,
doRetest, Freq, DcBias, Cp, Gp, Gp, Leakage);
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "loadEx", "CompNone", doComp,
doRetest, Freq, DcBias, Cp, Gp, Gp, Leakage);
```
4. On the device under test (DUT), execute the following test:
```
doComp = 1;
doRetest = 0;
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "open", "CompNone", doComp,
doRetest, Freq, DcBias, Cp, Gp, Gp, Leakage);
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "dut", "CompShortOpenLoad",
doComp, doRetest, Freq, DcBias, Cp, Gp, Gp, Leakage);
```

The sequence of tests shown above does the following:

- Enables all measurements to run through system-level compensation one time (*doComp* parameter set to 1).
- Forces all compensation data (short, open, load, and loadEx) to be collected only once (*doRetest* parameter set to 0). To force data to be collected again, you can set the *doRetest* parameter to 1.
- Sets the *comp_mode* parameter to CompShortOpenLoad to do ShortOpenLoad compensation.
- Sets the *dut* parameter to short to collect data on the short device. This type of compensation is useful when there is very large capacitance with small impedance.
- Sets the *dut* parameter to load to collect gain compensation data with the high-voltage CMTR.
- Sets the *dut* parameter to loadEx to collect gain compensation data with the low-voltage CMTR.
- Sets the *dut* parameter to open to move the chuck down and collect compensation data.
- Enables characterization of the device under test (DUT) for short, open, and load devices by setting the *comp_mode* parameter to CompShortOpenLoad. This step uses previously collected data; if short, open, or load data was not previously collected, the test fails (even if the appropriate compensation mode was specified).
- Stores compensation data in memory; this data is only available within the same process. This means that if, for example, lines of code are executed one at a time using the Keithley Interactive Test Tool (KITT), compensation data will not be available and the test will fail. For this debug scenario, use the hvcv_storeData command to store required data in the data pool.

High-voltage C-V usage scenarios

The following topics contain example high-voltage capacitance-voltage C-V measurement applications using system-level or device-level compensation factors.

Two-terminal HV C-V measurement with system-level compensation

This is the simplest type of high-voltage capacitance-voltage (C-V) measurement. One of the following S540 High-Voltage Library (HVLib) commands is used:

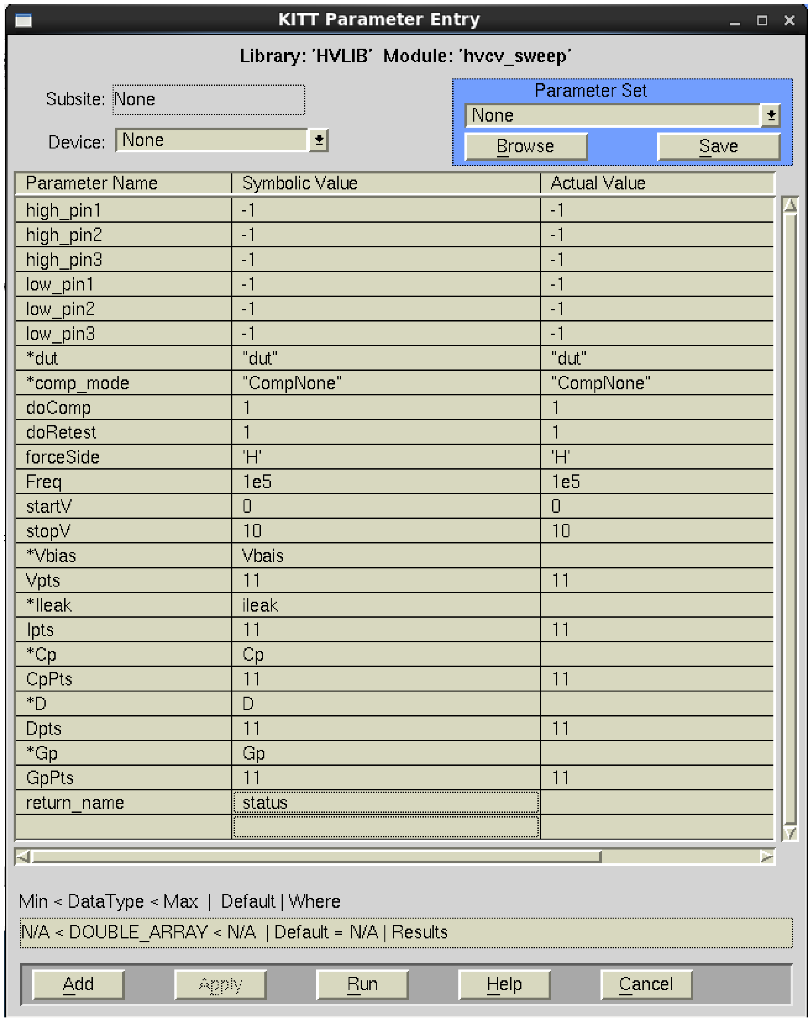
- hvcv_test: A single measurement is made at one voltage bias level
- hvcv_sweep: An array of measurements is made using an array of voltage biases

The following code is an example of using the hvcv_sweep command:

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "dut", "CompNone", 1, "H", 0, 1e5, 0,
   10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In the example above, "dut" indicates that the test will run on the device under test. "CompNone" indicates that device-level compensation will not be used. The value following "CompNone" (the *doComp* parameter) is set to 1 to specify that the test uses system-level compensation factors supplied by the factory.

Figure 7: High-voltage C-V sweep test



Two-terminal HV C-V measurement with device-level compensation

This example shows how you can run bench-top tests on a single device using the Keithley Interactive Test Tool (KITT). There are two parts to this process:

1. Run device-level compensation using the `hvcv_genCompData` user module in KITT:
   a. Input the system pin numbers to use to test your device.
   b. Run the module.
   c. Following the prompts in KITT, connect the pins to an open device, a short device, and a load device. The load device should be a known good device with capacitance properties similar to the ones you expect from the device you intend to test.
   The S540 system uses both capacitance meters (high-voltage and low-voltage CMTRs) in the system to measure this device with and without connections through bias tees.
   When the test is complete, compensation data is stored to a file on the system for later retrieval.
2. Make the measurement using either the `hvcv_test` command (for a single measurement at one voltage bias level) or the `hvcv_sweep` command (for an array of measurements using an array of voltage biases).

The following code is an example using the `hvcv_sweep` command.

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "dut", "CompNone", 2, "H", 0, 1e5, 0,
    10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In the example above, `"dut"` indicates that the test will run on the device under test. `"CompNone"` indicates that automated device-level compensation will not be used. The value following `"CompNone"` (the *doComp* parameter) is set to 2 to specify that the test uses the device-level compensation data that you just created using the `hvcv_genCompData` module in step 1.

NOTE

The compensation factors that you created using the `hvcv_genCompData` module contain data for all frequencies available on the system at the time that you ran the module. They are specific to the pins and setup configured when the module was run. These compensation factors remain on the system for reuse until the `hvcv_genCompData` module runs again. When the module runs again, the data is overwritten with new compensation factors based on the configuration at that time.

Automated two-terminal HV C-V measurement with device-level compensation

If you are using an automated test plan module to test a series of devices, you start by collecting device-level compensation data. The process of doing this is slightly different than testing at the bench level.

In an automated setting, you can choose any combination of the following types of device-level compensation:

- Open
- Short
- Load

Each type of device-level compensation data is collected separately using one of the following S540 High-Voltage Library (HVLib) commands:

- `hvcv_test`: A single measurement is made at one voltage bias level
- `hvcv_sweep`: An array of measurements is made using an array of voltage biases

The following code is an example of each type of device-level compensation using the `hvcv_sweep` command.

Open compensation:

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "open", "CompNone", 0, "H", 0, 1e5, 0,
    10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

Short compensation:

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "short", "CompNone", 0, "H", 0, 1e5, 0,
    10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

Load compensation:

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "load", "CompNone", 0, "H", 0, 1e5, 0,
    10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "loadEX", "CompNone", 0, "H", 0, 1e5,
    0, 10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In each of the examples above, the *dut* parameter specifies the type of compensation device to be used (open, short, or load).

NOTE

For the load type of compensation, you must run the test twice: Once using *dut* parameter load and once using the *dut* parameter loadEx. This tells the system to run the test first using the high-voltage capacitance meter (CMTR) connected through bias tees, then to run the test using the low-voltage CMTR with no connection through bias tees. The same parameters should be used for both the load and loadEx DUT tests.

In the examples above, "CompNone" indicates that no compensation is done as you are collecting compensation data. The value following "CompNone" (the *doComp* parameter) is set to 0 so that the routine does not do ShortOpenLoad compensation as you are collecting compensation data.

The *doRetest* parameter is set to 0 so that if the hvcv_sweep command is called again during the same run of the test plan module (with the identical pin configuration), the retest is skipped. This saves time by using compensation data that was previously collected and stored in the data pool. This is useful when you are testing multiple devices or wafers. If you want to collect new compensation data for each pin configuration (even if it is identical to a previous pin configuration), set the *doRetest* parameter to 1.

NOTE

Use prober commands to ensure that the pins are connected to an appropriate device (or not connected to anything if the open compensation mode is selected) before each of the compensation commands is run. See the *S530/S540 Prober and Prober Driver Manual* (part number S530-911-01) for descriptions of prober commands.

Once compensation data has been collected, measure the device using either the hvcv_test command (for a single measurement at one voltage bias level), or the hvcv_sweep command (for an array of measurements using an array of voltage biases). Following is an example using the hvcv_sweep command.

```
return_name = hvcv_sweep(1, -1, -1, 2, -1, -1, "dut", "CompShortOpenLoad", 0, "H", 0,
    1e5, 0, 10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In this example, "dut" indicates that you are testing the device under test. "CompShortOpenLoad" specifies that all three types of compensation are used. Alternatively, you could use "CompShortOpen" to use short and open compensation only, or "CompOpen" to use only open compensation, and so on. The *doComp* parameter is set to 0 so that automated device-level compensation is used instead of system-level or bench-level compensation.

Three-terminal HV C-V measurement with device-level compensation

There are several steps that must be completed in a single test run for this scenario:

- Collect device-level compensation factors
- Store compensation factors in the data pool
- Test the device

Following is an example of three-terminal high-voltage capacitance-voltage (C-V) measurement using device-level compensation.

Step 1: Collect device-level compensation factors

Do the device-level compensation for the type of devices you are testing (open, short, or load devices, or any combination of those devices) using the `hvcv_3term` module in the Keithley Interactive Test Tool (KITT). For example:

Open compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "open", "CompNone", 1e5, 0, 0, drainV,
    11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

Short compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "short", "CompNone", 1e5, 0, 0,
    drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

Load compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "load", "CompNone", 1e5, 0, 0, drainV,
    11, drainI, 11, Cp, 11, D, 11, Gp, 11)
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "loadEx", "CompNone", 1e5, 0, 0,
    drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

In each of the examples above, the `dut` parameter indicates the type of compensation device to use (open, short, or load).

NOTE

For the load type of compensation, you must run the test twice: Once using *`dut`* parameter `load` and once using the *`dut`* parameter `loadEx`. This tells the system to run the test first using the high-voltage capacitance meter (CMTR) connected through bias tees, then to run the test using the low-voltage CMTR with no connection through bias tees. The same parameters should be used for both the `load` and `loadEx` DUT tests.

`"CompNone"` indicates that no compensation is done as you are collecting compensation data. When the module is run in this mode, only one actual value is returned for Cp and Gp, and all the other values in the array are 0. This is because compensation is run at a 0 V bias only. Record the `Cp` and `Gp` values returned. These are the compensation factors that you use as inputs in step 2.

When this step is complete, record these values.

Step 2: Store compensation factors in the system data pool

Use the `hvcv_storeData` command as shown below to store compensation factors in the data pool.

Open compensation:

```
return_name = hvcv_storeData("D1_G4_S5_Mode:Ciss", "open", 1e5,3.85523e-12,
    -4.50236e-8)
```

Short compensation:

```
return_name = hvcv_storeData("D1_G4_S5_Mode:Ciss", "short", 1e5,8.3487E-8, 0.0136743)
```

Load compensation:

```
return_name = hvcv_storeData("D1_G4_S5_Mode:Ciss", "load", 1e5, 9.61043e-10,
    2.87691e-6)
return_name = hvcv_storeData("D1_G4_S5_Mode:Ciss", "loadEx", 1e5, 9.61043e-10,
    2.87691e-6)
```

The value of the *label* parameter in each of the code examples above is determined by the compensation mode you are using (`Ciss`, `Coss`, or `Crss`) and the pin numbers used to test the device. In this example, the label parameter is `"D1_G4_S5_Mode:Ciss"`, which means the compensation mode is C_{iss}, the drain is connected to pin 1, the gate is connected to pin 4, and the source is connected to pin 5.

Step 3: Run the hvcv_3term module in KITT

Immediately after storing the compensation factors in the data pool, run the `hvcv_3term` module in KITT to test the device.

NOTE

The data pool only retains information for the duration of a single test run in KITT. Because of this, you must run all `hvcv_storeData` modules and the `hvcv_3term` module in a single KITT test run.

The following is an example of how to complete this step.

```
return_name1 = hvcv_storeData("D1_G4_S5_Mode:Ciss", "open", 1e5,3.85523e-12,
    -4.50236e-8)
return_name2 = hvcv_storeData("D1_G4_S5_Mode:Ciss", "short", 1e5,8.3487E-8,
    0.0136743)
return_name3 = hvcv_storeData("D1_G4_S5_Mode:Ciss", "load", 1e5, 9.61043e-10,
    2.87691e-6)
return_name4 = hvcv_storeData("D1_G4_S5_Mode:Ciss", "loadEx", 1e5, 9.61043e-10,
    2.87691e-6)
return_name5 = hvcv_3term(1, 4, 5, 0, 0, 10, "Ciss", "dut", "CompShortOpenLoad", 1e5,
    0, 1, V, 11, I, 11, Cp, 11, D, 11, Gp, 11)
```

In the `hvcv_3term` example above, `"dut"` indicates that the test will run on the device under test. `"CompShortOpenLoad"` specifies that all three modes of compensation will be used. You could instead use any of the other compensation modes, for example: `"CompShortOpen"` for short and open only, or `"CompOpen"` for open only. The `doComp` parameter is set to `0` to specify that automated device-level compensation is used instead of system-level compensation.

Automated three-terminal HV C-V measurement with device-level compensation

This type of testing is similar to automated two-terminal testing, except instead of using the `hvcv_test` or `hvcv_sweep` commands, you use the `hvcv_3term` command.

Open compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "open", "CompNone", 1e5, 0, 0, drainV,
    11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

Short compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "short", "CompNone", 1e5, 0, 0,
    drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

Load compensation:

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "load", "CompNone", 1e5, 0, 0, drainV,
    11, drainI, 11, Cp, 11, D, 11, Gp, 11)
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "loadEx", "CompNone", 1e5, 0, 0,
    drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

In each of the examples above, the *dut* parameter specifies the type of compensation device to be used (`open`, `short`, or `load`).

NOTE

For the load type of compensation, you must run the test twice: Once using *dut* parameter `load` and once using the *dut* parameter `loadEx`. This tells the system to run the test first using the high-voltage capacitance meter (CMTR) connected through bias tees, then to run the test using the low-voltage CMTR with no connection through bias tees. The same parameters should be used for both the `load` and `loadEx` DUT tests.

`"CompNone"` indicates that no compensation is done as you are collecting compensation data. The value following `"CompNone"` (the *doComp* parameter) is set to 0 so that the routine does not do ShortOpenLoad compensation as you are collecting compensation data.

The *doRetest* parameter is set to `0` so that if the `hvcv_3term` command is called again during the same run of the test plan module (with the identical pin configuration), the retest is skipped. This saves time by using compensation data that was previously collected and stored in the data pool. This is useful when you are testing multiple devices or wafers. If you want to collect new compensation data for each pin configuration (even if it is identical to a previous pin configuration), set the *doRetest* parameter to `1`.

NOTE

Use prober commands to ensure that the pins are connected to an appropriate device (or not connected to anything if the open compensation mode is selected) before each of the compensation commands is run. See the *S530/S540 Prober and Prober Driver Manual* (part number S530-911-01) for descriptions of prober commands.

Once compensation data has been collected, measure the device using the `hvcv_3term` command, as shown below.

```
return_name = hvcv_3term(1, 2, 3, 0, 0, 10, "Ciss", "dut", "CompShortOpenLoad", 1e5,
    0, 0, drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

In this example, `"dut"` indicates that you are testing the device under test. `"CompShortOpenLoad"` specifies that all three types of compensation are used. Alternatively, you could use `"CompShortOpen"` to use short and open compensation only, or `"CompOpen"` to use only open compensation, and so on. The *doComp* parameter is set to `0` so that automated device-level compensation is used instead of system-level or bench-level compensation.

Section 4

High-Voltage Library command reference

In this section:

How to use the library reference

The commands in the S540 high-voltage library (HVLib) are in the C programming language. Each command is presented in a standard format that follows the pattern below:

- **Purpose statement:** The first line of text under the command heading contains a brief explanation of what the command does.

Figure 8: Example purpose statement

hvcv_comp

This command does complex mathematical calculations to implement specified impedance compensation models.

- **Usage:** A line of code representing the prototype of the command, followed by a table listing the input and output parameters for the command.

  Parameters that you specify are shown in *`monospace italic`* font. Parameters preceded by an asterisk (`*`) are character parameters that are passed into the function (input) or pointers to information that is returned (output).

  Each parameter is preceded by one of the following declarations that specifies the data type for the parameter: `int` (integer), `double` (double-precision floating-point), and `char` (a character string).

Figure 9: Example syntax and parameter description

Usage

```
int hvcv_comp(char *label, char *comp_mode, double Freq, double *CpComp, double
    *GpComp, double *DComp)
```

label	Input	Device name (label), for example, `DUT1` or `p1_3`
comp_mode	Input	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *dut* parameter is set to anything other than `dut`) • `CompOpen` • `CompLoad` • `CompShort` • `CompOpenLoad` • `CompShortEx` • `CompShortOpen` • `CompShortOpenLoad`
Freq	Input	Frequency (`1e4` to `2e6`)
CpComp	Output	Compensated capacitance (*Cp*) value after correction
GpComp	Output	Compensated conductance (G*p*) value after correction
DComp	Output	Compensated dissipation factor after correction
cal	Output	Value of calibration constants

- **Details:** Additional information about using the command.

Figure 10: Example details

Details

This command does the following:

- Verifies input conditions
- Gets capacitance-voltage (*Cp*, *Gp*) data for the specified device label and all specified device types (`dut`, `open`, `short`, `load`)
- Runs compensation model specified by the *`comp_mode`* parameter
- Reports corrected *Cp* and *Gp* values

This command does separate `CompOpen`, `CompShort`, and `CompLoad` compensation or any combination of these modes (for example, `CompOpenLoad`, `CompShortOpen`, `CompShortOpenLoad`). If compensation data is not already stored in the data pool when device testing is done or incorrect labels are used, an error is returned.

- **Example:** Lines of code showing what a call to the command might look like in actual use.

Figure 11: Example command call

Example

```
stat = hvcv_comp("pin1_pin2", "CompShortOpen", 1e5, CpComp, GpComp, DComp, Cal)
```

Does ShortOpen compensation on the device named `pin1_pin2` and returns the Cp, Gp, and D values after compensation.

- **Also see:** Cross-references to other related commands and topics, where applicable.

Figure 12: Example cross-references

Also see

hvcv_measure (on page 3-14)
hvcv_comp (on page 3-7)

High-Voltage Library commands

The S540 High-Voltage Library (HVLib) commands are described in detail in the following topics.

gate_charge

This command measures the gate charge required to switch on the power transistor.

Usage

```
int gate_charge(int gate, int drain, int source, double Vds, double drainLimitI, double
    gateCurrent, double gateMaxV, double timeOut, int measDrain, double *timeArray, int
    timeArraySize, double *VgArray, int VgArraySize, double *VgCharge, int VgChargeSize,
    double *VdArray, int VdArraySize, double *Slope, int SlopeSize, double Coffset,
    double *Ceff, double *Vpl, double *T1, double *T2, double *Qgs, double *Qgd)
```

gate	Input	The gate pin
drain	Input	The drain pin
source	Input	The source pin
Vds	Input	Drain voltage
drainLimitI	Input	Current limit for the drain instrument; 1 A maximum
gateCurrent	Input	Amount of current to force into the gate
gateMaxV	Input	Voltage compliance limit for the gate; 200 V maximum
timeOut	Input	Timeout value for the test; 200 seconds maximum
measDrain	Input	Flag to enable (1) or disable (0) drain voltage measurement
timeArray	Output	Array to store timestamps
timeArraySize	Input	Size of the *timeArray* parameter
VgArray	Output	Array to store gate voltage
VgArraySize	Input	Size of the *VgArray* parameter
VgCharge	Output	Array to store gate charge
VgChargeSize	Input	Size of the *VgCharge* parameter
VdArray	Output	Array to store drain voltage
VdArraySize	Input	Size of the *VdArray* parameter
Slope	Output	Array to store slope (gate voltage (Vg) versus time) values
SlopeSize	Input	Size of the *Slope* parameter array
Coffset	Input	Parasitic capacitance of the gate cable
Ceff	Output	Ratio of total gate charge to maximum gate voltage
Vpl	Output	Gate voltage of the plateau area
T1	Output	Timestamp where the plateau area begins
T2	Output	Timestamp where the plateau area ends
Qgs	Output	Gate to source charge; calculate according to the JEDEC standard JESD 24-2
Qgd	Output	Gate to drain charge; calculate according to the JEDEC standard JESD 24-2

Details

This test does the following:

- Verifies input conditions
- Sets up a gate SMU and a drain SMU (Model 2636s):
  - Sets gate SMU voltage limit to *gateMaxV*
  - Sets gate SMU current range to the fixed 10**gateCurrent* range
  - Sets drain SMU current compliance and range to *drainLimitI*
- Makes connections
- Forces *GateCurrent* into the gate
- Measures gate voltage as a function of time
- Optionally (depending on the value of the *measDrain* parameter, `0` or `1`) measures drain voltage
- Determines the two points of inflection on the curve of Vg = Vg(Time) and reports it
- Adjusts values of the gate charge using the specified parasitic capacitance (*Coffset*)
- Returns effective capacitance, defined as gateMaxV/total-gate-charge
- Calculates and returns the gate charges (*Qgs* and *Qgd*) values according to the JEDEC standard JESD 24-2
- Returns the test status

This command returns a status:

- 1 Success
- –1 Gate voltage compliance limit exceeds 200 V
- –2 Drain current limit exceeds 1 A
- –3 *VgArraySize* is not equal to *timeArraySize*
- –4 *VdArraySize* is not equal to *timeArraySize*
- –5 Timeout exceeds maximum: 200 s
- –6 Test time exceeds timeout value
- –7 Number of measurements exceeds maximum allowed (10000)
- –8 *SlpSize* is not equal to *timeArraySize*
- –9 Compliance or test error
- –10 Error calculating S1, correlation factor < 0.9
- –11 Error calculating S2, correlation factor < 0.9
- –12 Power limit (current should be less than 0.1 A if voltage is greater than 20 V) exceeded

Example

```
pts = 600;
stats = gate_charge(3, 4, 5, 20, 0.1, 3e-8, 10, 10, 1, Time, pts, Vg, pts, Vq, pts,
      Vd, pts, Slp, pts, 3e-10, Coff, Vpl, T1, T2, Qgs, Qgd)
```

This test implements the JEDEC standard JESD 24-2.

Also see

None

hv_bvsweep

This command does a breakdown voltage sweep.

Usage

```
int hv_bvsweep(int high1, int high2, int high3, int low1, int low2, int low3, double
    vStart, double vStop, double vStep, double stepDelay, double trigCurrent, double
    compl, double ratio, double *bV, double *bVR, double *LeakR, double *Vbias, int
    VbiasPts, double *Imeas, int ImeasPts)
```

high1	Input	High pin 1
high2	Input	High pin 2
high3	Input	High pin 3
low1	Input	Low pin 1
low2	Input	Low pin 2
low3	Input	Low pin 3
vStart	Input	The start voltage of the sweep
vStop	Input	The stop voltage of the sweep
vStep	Input	The voltage step size for the sweep
stepDelay	Input	The step delay, in seconds
trigCurrent	Input	The current trigger level
compl	Input	The current limit
ratio	Input	The ratio of voltage to breakdown voltage at which voltage and current are reported
bV	Output	Breakdown voltage
bVR	Output	Voltage at a specified percent (*ratio*) of the breakdown voltage
LeakR	Output	The current level at *bVR*
Vbias	Output	The forced voltage bias
VbiasPts	Input	The number of voltage bias points
Imeas	Output	The measured current
ImeasPts	Input	The number of current measure points

Details

This command does the following:

- Verifies input conditions
- Configures source-measure unit (SMU) and trigger levels
- Sweeps voltage from *vStart* to *vStop*
- Reports breakdown voltage (bV) at the trigger point
- Reports leakage at a specified ratio of breakdown voltage

The *VbiasPts* and *ImeasPts* parameters are the number of points to use; they should be equal to or greater than (Vstop – Vstart)/Vstep + 1.

This command returns a status:

- `1` = Success
- `-1` = Invalid high pins
- `–2` = Invalid low pins
- `–3` = Invalid *`Vstart`* and *`Vstop`* values
- `–4` = Invalid ratio; valid range is 0.01 to 0.99
- `–5` = Invalid *`trigCurrent;`* valid range is 1e-9 A to 0.001 A
- `–6` = Invalid *`vStep;`* valid range is 0.1 V to 20 V
- `–7` = Invalid *`stepDelay;`* valid range is 0.001 s to 0.5 s
- `–8` = Wrong number of points; should be equal to or larger than (Vstop – Vstart)/Vstep + 1
- `–9` = Low-voltage pins are used for high-voltage test
- `–10` = Test is too fast; slow it down or increase the *`trigCurrent`* level

Example

```
startV = 0
stopV = 2500.0
status = hv_bvsweep(pin1, -1, -1, pin2, -1, -1,startV, stopV, 5, 0.1,1e-6,1e-5,
     0.85, BV, BVR, Leak, Vbias, 501, Imeas, 501)
```

Measures breakdown voltage by sweeping 0 V to 2500 V in 5 V steps.

Also see

None

hvcv_3term

This command measures output capacitance (C_{oss}), input capacitance (C_{iss}), or short-circuit reverse transfer capacitance (C_{rss}) of three-terminal devices.

Usage

```
int hvcv_3term(int drain, int gate, int source, double gateV, double startV, double
    stopV, char *mode, char *dut, char *comp_mode, double Freq, int doComp, int doRetest,
    double *drainV, int drainVPts, double *drainI, int drainPts, double *Cp, int CpPts,
    double *D, int DPts, double *Gp, int GpPts)
```

drain	Input	Drain pin
gate	Input	Gate pin
source	Input	Source pin
gateV	Input	Gate voltage
startV	Input	Start voltage of drain voltage sweep
stopV	Input	Stop voltage of drain voltage sweep
mode	Input	Bias connections mode (`Ciss`, `Coss`, or `Crss`)
dut	Input	Device under test; valid options: • `dut` = Test the DUT itself with the high-voltage capacitance meter (CMTR) • `open` = Characterize the open device using the high-voltage CMTR; this can be done with the chuck down (pins not in contact with the device) • `short` = Characterize the short device using the high-voltage CMTR • `load` = Characterize the load device using the high-voltage CMTR • `shortEx` = Characterize the short device using the low-voltage capacitance meter (CMTR); this data is used to do `CompShort` compensation of the *`loadEx`* device • `loadEx` = Measure the load device using the low-voltage CMTR to get the expected value of the `loadEx` device • `openEx` = Characterize the open device using the low-voltage CMTR; this data is used to do `CompOpen` compensation of the `loadEx` device
comp_mode	Input	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *`dut`* parameter is set to anything other than `dut`) • `CompOpen` • `CompShort` • `CompLoad` • `CompOpenLoad` • `CompShortOpen` • `CompShortLoad` • `CompShortOpenLoad`
Freq	Input	Frequency; recommended frequency is `1e5` Hz
doComp	Input	Specifies whether to do system-level compensation: • `0` = Do not do system-level compensation • `1` = Do system-level compensation

doRetest	Input	Specifies whether to remeasure compensation data: • `0` = Do not remeasure compensation data • `1` = Remeasure compensation data once, then reuse that measurement in any additional calls See **Details** for more information
drainV	Output	Drain voltage array
drainVPts	Input	Number of drain voltage points in the array
drainI	Output	Drain current array
drainIPts	Input	Number of drain current points in the array
Cp	Output	Capacitance
CpPts	Input	Compensated capacitance points
D	Output	Dissipation factor
DPts	Input	Compensated dissipation factor points
Gp	Output	Compensated conductance
GpPts	Input	Compensated conductance points

Details

This command can also do open compensation of the device under test (DUT), defined by the *comp_mode* parameter. This includes separate `CompOpen`, `CompShort`, and `CompLoad` compensation or any combination of these modes (for example, `CompOpenLoad`, `CompShortOpen`, `CompShortOpenLoad`). If compensation data (`open`, `short`, `load`, `loadEx`, `openEx`, `shortEx`) is not available before device testing, an error is returned.

For best results measuring C_{rs}, suppress the AC signal at the source terminal by connecting the high-voltage ground (HV GND) terminal to the source. In a system configured with a high-voltage matrix and long high-voltage cables, passive AC guarding (GND) provides superior performance over AC guarding using bias tees.

This command also collects compensation data for `open`, `short`, `load`, `loadEx`, `openEx`, and `shortEx`. Compensation data for `openEx` and `loadEx` is collected using the low-voltage CMTR, bypassing the bias-tees.

The *doComp* parameter provides a switch that enables or disables system-level compensation. To do ShortOpenLoad compensation using a system-level compensation file that is stored on the system (`cvCALsystem.ini`), set this parameter to `1`. To do ShortOpenLoad compensation using a user-generated compensation file (`cvCAL.ini`), set this parameter to `2`.

The *doRetest* parameter provides a switch that enables or disables remeasurement of the compensation data.

This command returns a status:

- `-1` = Arrays have a different number of output points; all arrays must have the same number of points
- `-2` = Gate, drain, or source pins are not defined
- `-3` = Invalid *dut* parameter name; valid names are `dut`, `open`, `load`, `loadEx`, `openEx`, and `shortEx`

- `-4` = Invalid compensation mode (*`comp_mode`*) name; valid names are `CompNone`, `CompOpen`, `CompShort`, `CompLoad`, `CompOpenLoad`, `CompShortOpen`, `CompShortEx`, and `CompShortOpenLoad`
- `-5` = Error when moving chuck down
- `-6` = Invalid *`mode`* parameter name; valid names are `Crss`, `Coss`, and `Ciss`
- `-7` = Low voltage pin is used for high-voltage test

Example

```
status = hvcv_3term(drain, gate, source, 0, 0, 10, "Ciss", "dut", "CompOpen", 1e5,
    1, 1, drainV, 11, drainI, 11, Cp, 11, D, 11, Gp, 11)
```

Measures C_{iss} of a three-terminal device.

Also see

hvcv_3term_basic (on page 4-9)

hvcv_3term_basic

This command measures output capacitance (C_{oss}), input capacitance (C_{iss}), or short-circuit reverse transfer capacitance (C_{rss}) of three-terminal devices.

Usage

```
int hvcv_3term_basic(int drain, int gate, int source, double gateV, double startV,
    double stopV, char *mode, double Freq, double *drainV, int drainVPts, double *drainI,
    int drainPts, double *Cp, int CpPts, double *D, int DPts, double *Gp, int GpPts)
```

drain	Input	Drain pin
gate	Input	Gate pin
source	Input	Source pin
gateV	Input	Gate voltage
startV	Input	Start voltage of drain voltage sweep
stopV	Input	Stop voltage of drain voltage sweep
mode	Input	Bias connections mode (`Ciss`, `Coss`, or `Crss`)
Freq	Input	Frequency; recommended frequency is `1e5` Hz
drainV	Output	Drain voltage array
drainVPts	Input	Number of drain voltage points in the array
drainI	Output	Drain current array
drainIPts	Input	Number of drain current points in the array
Cp	Output	Capacitance
CpPts	Input	Compensated capacitance points
D	Output	Dissipation factor
DPts	Input	Compensated dissipation factor points
Gp	Output	Compensated conductance
GpPts	Input	Compensated conductance points

Details

For best results measuring C_{rs}, suppress the AC signal at the source terminal by connecting the high-voltage ground (HV GND) terminal to the source. In a system configured with a high-voltage matrix and long high-voltage cables, passive AC guarding (GND) provides superior performance over AC guarding using bias tees.

This command returns a status:

- `0` = Skip system-level compensation
- `-1` = Arrays have a different number of output points; all arrays must have the same number of points
- `-2` = Gate, drain, or source pins are not defined
- `-3` = Invalid *`dut`* parameter name; valid names are `dut`, `open`, `load`, `loadEx`, `openEx`, and `shortEx`
- `-5` = Error when moving chuck down
- `-6` = Invalid *`mode`* parameter name; valid names are `Crss`, `Coss`, and `Ciss`
- `-7` = Low voltage pin is used for high-voltage test

Example

```
status = hvcv_3term(drain, gate, source, 0, 0, 10, "Ciss", 1e5, drainV, 11, drainI,
    11, Cp, 11, D, 11, Gp, 11)
```

Measures C_{iss} of a three-terminal device.

Also see

hvcv_3term (on page 4-7)

hvcv_comp

This command does complex mathematical calculations to implement specified impedance compensation models.

Usage

```
int hvcv_comp(char *label, char *comp_mode, double Freq, double *CpComp, double *GpComp,
    double *DComp)
```

label	Input	Device name (label), for example, `DUT1` or `p1_3`
comp_mode	Input	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *dut* parameter is set to anything other than `dut`) • `CompOpen` • `CompShort` • `CompLoad` • `CompOpenLoad` • `CompShortOpen` • `CompShortLoad` • `CompShortOpenLoad`
Freq	Input	Frequency (`1e4` to `2e6`)
CpComp	Output	Compensated capacitance (*Cp*) value after correction
GpComp	Output	Compensated conductance (G*p*) value after correction
DComp	Output	Compensated dissipation factor after correction
cal	Output	Value of calibration constants

Details

This command does the following:

- Verifies input conditions
- Gets capacitance-voltage (*Cp*, *Gp*) data for the specified device label and all specified device types (`dut`, `open`, `short`, `load`)
- Runs compensation model specified by the *comp_mode* parameter
- Reports corrected *Cp* and *Gp* values

This command does separate `CompOpen`, `CompShort`, and `CompLoad` compensation or any combination of these modes (for example, `CompOpenLoad`, `CompShortOpen`, `CompShortOpenLoad`). If compensation data is not already stored in the data pool when device testing is done or incorrect labels are used, an error is returned.

This command returns a status:

- `1` = Success
- `-1` = Device label is NULL
- `-2` = Device label is less than 2 characters or more than 64
- `-3` = Invalid compensation mode; valid modes are `CompNone`, `CompOpen`, `CompLoad`, `CompShort`, `CompOpenLoad`, `CompShortEx`, `CompShortOpen`, or `CompShortOpenLoad`

- `-4` = Frequency is out of valid range (`1e4` to `2e6`)
- `-5` = Failed on data retrieval for DUT
- `-6` = Failed on data retrieval for open
- `-7` = Failed on data retrieval for short
- `-8` = Failed on data retrieval for load
- `-9` = Failed on data retrieval for loadEx

Example

`stat = hvcv_comp("pin1_pin2", "CompShortOpen", 1e5, CpComp, GpComp, DComp, Cal)`
Does ShortOpen compensation on the device named `pin1_pin2` and returns the Cp, Gp, and D values after compensation.

Also see

None

hvcv_genCompData

This command generates correction factors for system-level high-voltage capacitance-voltage (C-V) compensation.

Usage

```
int hvcv_genCompData(int hpin, int lpin)
```

hpin	Input	Pin for the capacitance meter (CMTR) high signal
lpin	Input	Pin for CMTR low signal

Details

The correction factors generated by this command are saved as calibration constants in `/opt/kiS530/cvCAL.ini`. These calibration constants are used by the `hvcv_intgcg` command.

`CompOpen`, `CompShort`, and `CompLoad` devices must be connected to run this procedure. Select a `CompLoad` device with a value close to the capacitance you are measuring. If you are configuring three-terminal capacitance measurements, use a 1 nF to 2 nF capacitor.

This command uses the low-voltage CMTR as a calibration tool to provide load values for high-voltage CMTR characterization.

This command returns a status:

- 1 = Success
- -1 = Low or high pins are not defined
- -2 = Open measurement failed
- -3 = Open correction canceled
- -4 = Short measurement failed
- -5 = Short correction canceled
- -6 = Load measurement failed

- -7 = Load correction canceled
- -8 = DUT/load measurement failed
- -9 = Failed to open `/opt/kiS530/cvCAL.ini` file
- -10 = Failed running compensation calculations

Example

`status= hvcv_genCompData(pin1, pin2)`
Generates compensation factors for pin 1 and pin 2.

Also see

None

hvcv_genCompFreq

This command generates compensation factors for system-level capacitance compensation for a single, specified frequency.

Usage

```
int hvcv_genCompFreq(int hpin, int lpin, int epin, int CMTRs, double Freq, double Cp,
    double Gp, double *CpCalc,  double *GpCalc)
```

hpin	Input	Pin for the capacitance meter (CMTR) high signal
lpin	Input	Pin for CMTR low signal
epin	Input	Extra pin
CMTRs	Input	Number of CMTRs to use to do capacitance-voltage compensation: 1 = Use only high-voltage CMTR for compensation measurements; for load values, use the *CP* and *GP* parameters 2 = Use both high-voltage and low-voltage CMTRs to obtain *Cp* and *Gp* values (user-specified values using the *Cp* and *Gp* parameters are ignored); the low-voltage CMTR obtains the *Cp* and the high-voltage CMTR obtains the *Gp*.
Freq	Input	Frequency
Cp	Input	Expected or known value for compensated capacitance; use when low-voltage CMTR cannot be used to collect *Cp* value
Gp	Input	Expected or known value for compensated conductance; use when low-voltage CMTR cannot be used to collect *Gp* value
CpCalc	Output	Corrected value for compensated capacitance; should be close to the expected, known, and measured values on low-voltage CMTR
GpCalc	Output	Corrected value for compensated conductance; should be close to the expected, known, and measured values on low-voltage CMTR

Details

This command can be used in two different CMTR configurations, as specified by the *`CMTRs`* parameter:

- 1 = Using only a high-voltage CMTR connected through bias tees; you must provide values for the load device, compensated capacitance, and compensated conductance
- 2 = Using a low-voltage CMTR and a high-voltage CMTR; the low-voltage CMTR bypasses the bias tees and provides data for the load device (user-specified values using the *`Cp`* and *`Gp`* parameters are ignored)

When the command runs successfully, correction factors are displayed on the computer. You can then add these values to the `/opt/kiS530/cvCAL.ini` file.

NOTE

The correction factors are not automatically added to the `/opt/kiS530/cvCAL.ini` file; you must add them.

This command returns a status:

- 1 = Success
- -1 = Low or high pins are not defined
- -2 = Open measurement failed
- -3 = Open correction canceled
- -4 = Short measurement failed
- -5 = Short correction canceled
- -6 = Load measurement failed
- -7 = Load correction canceled
- -8 = DUT or load measurement failed
- -9 = Low-voltage measurement of DUT failed
- -10 = Storing expected value failed
- -11 = CompShortOpenLoad compensation routine failed
- -12 = Correction data does not match expected data

Example

```
Freq = 1e5
Cp = 1.23e-9
Gp = 4.5e-6
CMTRs = 2
status = hvcv_genCompFreq(pin1, pin2, -1, CMTRs, Freq, Cp, Gp, CpCalc, GpCalc);
```

Collects the compensation factor for one frequency.

Also see

None

hvcv_getData

This command gets compensated capacitance (Cp) and compensated conductance (Gp) data from the data pool.

Usage

```
int hvcv_getData(char *label, char *dut, double Freq, double *Cp, double *Gp)
```

label	Input	Device name (label), for example, `DUT1` or `p1_3`
dut	Input	Device type (`dut`, `open`, `short`, `load`, `loadEx`, `shortEx`, or `openEx`)
Freq	Input	Frequency (`1e4` to `2e6`)
Cp	Output	Compensated capacitance value
Gp	Output	Compensated conductance value

Details

This command gets *Cp* and *Gp* data from the data pool using a keyword that specifies which device to get the data from. The keyword is derived by combining the device name (label), device type, and frequency. For example, the keyword `trans1_dut_10000` identifies the device named `trans1`, with a device type of `dut`, at a frequency of 1e+4 Hz.

This command returns a status:

- `1` = Success
- `-1` = Device label is NULL
- `-2` = Device label is less than 2 characters or more than 64
- `-3` = DUT is not one of the following: `dut`, `open`, `short`, `load`, `loadEx`, `shortEx`, or `openEx`
- `-4` = Frequency is out of valid range (`1e4` to `2e6`)
- `-5` = Failed to read values from data pool

Example

`status = hvcv_getData("pin1_pin2", "dut", 1e5, Cp, Gp)`
Gets capacitance-voltage (C-V) data with the label `pin1_pin2` from a data pool.

Also see

None

hvcv_intgcg

This command measures capacitance and does system-level ShortOpenLoad compensation on the high-voltage capacitance meter (CMTR).

Usage

```
void hvcv_intgcg(int instr, int doComp, double Freq, double *Cp, double *Gp)
```

instr	Input	High-voltage CMTR (`CMTR2`)
doComp	Input	Specifies whether to do ShortOpenLoad compensation: • `0` = Do not do ShortOpenLoad compensation • `1` = Do ShortOpenLoad compensation using a system-level file installed on the system (`cvCALsystem.ini`) • `2` = Do ShortOpenLoad compensation using a user-created file (`cvCAL.ini`)
Freq	Input	Frequency
Cp	Input	Capacitance value, according to the parallel capacitor model
Gp	Input	Conductance value, according to the parallel capacitor model

Details

This command does the following:

- Reads compensation `CompOpen`, `CompShort`, and `gain` correction parameters from `/opt/kiS530/cvCAL.ini`
- Makes a standard capacitance-voltage (C-V) measurement using the `intgcv` LPT command
- Does `CompOpen`, `CompShort`, and `CompLoad` compensation on the C-V measurements

Use this command instead of the `intgcg` Linear Parametric Test (LPT) command when you need to compensate for connections through bias tees.

The `hvcv_intgcg` command measures capacitance like the standard `intgcg` command, but it also does system-level compensation using a single set of constants stored in the `/opt/kiS530/cvCAL.ini` file. These constants are created using the `hvcv_genCompData` and `hvcv_genCompFreq` commands.

The instrument specified by the `instr` parameter must be a high-voltage CMTR (`CMTR2`).

Example

```
hvcv_intgcg(CMTR2, 1, 1e5, Cp, Gp)
```

Measures capacitance and does system-level ShortOpenLoad compensation on the high-voltage capacitance meter.

Also see

hvcv_genCompData (on page 4-12)
hvcv_genCompFreq (on page 4-13)

hvcv_measure

This command measures and stores compensated capacitance (Cp) and compensated conductance (Gp) values.

Usage

```
int hvcv_measure(int instr, char *label, char *dut, double Freq, double ACV, double PLC,
    int doComp, double *Cp, double *Gp, double *D)
```

instr	Input	Capacitance meter (CMTR) instrument ID
label	Input	Device name (label), for example, `DUT1` or `p1_3`
dut	Input	Device under test; valid options: • `dut` = Test the DUT itself with the high-voltage capacitance meter (CMTR) • `open` = Characterize the open device using the high-voltage CMTR; this can be done with the chuck down (pins not in contact with the device) • `short` = Characterize the short device using the high-voltage CMTR • `load` = Characterize the load device using the high-voltage CMTR • `shortEx` = Characterize the short device using the low-voltage capacitance meter (CMTR); this data is used to do `CompShort` compensation of the *loadEx* device • `loadEx` = Measure the load device using the low-voltage CMTR to get the expected value of the `loadEx` device • `openEx` = Characterize the open device using the low-voltage CMTR; this data is used to do `CompOpen` compensation of the `loadEx` device
Freq	Input	Frequency (`1e4` to `2e6`)
ACV	Input	AC amplitude level
PLC	Input	Power line integration time (recommend `1` to `3` PLC)
doComp	Input	Specifies whether to do system-level compensation: • `0` = Do not do ShortOpenLoad compensation • `1` = Do ShortOpenLoad compensation using `cvCALsystem.ini` • 2 = Do ShortOpenLoad compensation using a user-created file (`cvCAL.ini`)
Cp	Output	Compensated capacitance value
Gp	Output	Compensated conductance value
D	Output	Dissipation factor

Details

This command does the following:

- Verifies input conditions
- Configures the CMTR with the specified *ACV*, *Freq*, and *PLC*
- Disables all compensation operations on the CMTR
- Configures a parallel measurement model (CpGp)
- If the CMTR is high-voltage and requires system-level compensation, this command calls the `hvcv_intgcg` command, which does system-level ShortOpenLoad compensation; if the CMTR is low-voltage, this command calls the `intgcg` LPT command, which does not do system-level compensation
- Makes capacitance measurements
- Using the `hvcv_storeData` command, stores Cp and Gp data with the specified *label*, *dut*, and *freq* parameters in the data pool

This command returns a status:

- `1` = Success
- `-1` = Device label is NULL
- `-2` = Device label is less than 2 characters or more than 64
- `-3` = The *dut* parameter is not one of the following: `dut`, `open`, `short`, `load`, `loadEx`, `shortEx`, or `openEx`
- `-4` = Frequency is out of the valid range (`1e4` to `2e6`)
- `-5` = The *ACV* parameter value exceeds 0.1 V or less than 0.01 V
- `-6` = The *PLC* parameter value is out of range (`0.1` to `30`)

Example

```
ACV = 0.1
doComp = 1
status = hvcv_measure(CMTR1, "pin1_pin2", "dut", 1e5, ACV, 1, doComp, Cp, Gp, D)
```

Measures and stores Cp and Gp values from CMTR1 to the data pool under the label `pin1_pin2`.

Also see

hvcv_intgcg (on page 4-16)

hvcv_storeData

This command stores compensated capacitance (Cp) and compensated conductance (Gp) data in the data pool.

Usage

`int hvcv_storeData(char *label, char *dut, double Freq, double Cp, double Gp)`

label	Input	Device name (label), for example, `DUT1` or `p1_3`
dut	Input	Device type (`dut`, `open`, `short`, `load`, `loadEx`, `shortEx`, or `openEx`)
Freq	Input	Frequency (`1e4` to `2e6`)
Cp	Input	Compensated capacitance value
Gp	Input	Compensated conductance value

Details

This command stores *Cp* and *Gp* data in the data pool under a keyword that identifies a specific device. The keyword is derived by combining the device name (label), device type, and frequency. For example, the keyword `trans1_dut_10000` identifies the device named `trans1`, with a device type of `dut`, at a frequency of `1e+4` Hz.

This command returns a status:

- `1` = Success
- `-1` = Device label is NULL
- `-2` = Device name is less than two characters or more than 64
- `-3` = The *dut* parameter is not one of the following values: `dut`, `open`, `short`, `load`, `loadEx`, or `openEx`
- `-4` = Frequency is out of valid range (`1e4` to `2e6`)

Example

```
status = hvcv_storeData("pin1_pin2", "dut", 1e5,12.2e-12,1.56e-8)
```

Stores capacitance-voltage (C-V) data in the data pool with the label `pin1_pin2`.

Also see

None

hvcv_sweep

This command does a high-voltage capacitance-voltage (C-V) sweep.

Usage

```
int hvcv_sweep(int high_pin1, int high_pin2, int high_pin3, int low_pin1, int low_pin2,
    int low_pin3, char *dut, char *comp_mode, int doComp, char forceSide, int doRetest,
    double Freq, double startV, double stopV, double *Vbias, int Vpts, double *Ileak,
    int Ipts, double *Cp, int CpPts, double *D, int Dpts, double *Gp, int GpPts)
```

high_pin1	Input	First pin to connect to high-voltage capacitance meter (CMTR) high side
high_pin2	Input	Second pin to connect to high-voltage CMTR high side
high_pin3	Input	Third pin to connect to high-voltage CMTR high side
low_pin1	Input	First pin to connect to high-voltage CMTR low side
low_pin2	Input	Second pin to connect to high-voltage CMTR low side
low_pin3	Input	Third pin to connect to high-voltage CMTR low side
dut	Input	Device under test; valid options: • `dut` = Test the DUT itself with the high-voltage capacitance meter (CMTR) • `open` = Characterize the open device using the high-voltage CMTR; this can be done with the chuck down (pins not in contact with the device) • `short` = Characterize the short device using the high-voltage CMTR • `load` = Characterize the load device using the high-voltage CMTR • `shortEx` = Characterize the short device using the low-voltage capacitance meter (CMTR); this data is used to do `CompShort` compensation of the *loadEx* device • `loadEx` = Measure the load device using the low-voltage CMTR to get the expected value of the `loadEx` device • `openEx` = Characterize the open device using the low-voltage CMTR; this data is used to do `CompOpen` compensation of the `loadEx` device
comp_mode	Input	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *dut* parameter is set to anything other than `dut`) • `CompOpen` • `CompShort` • `CompLoad` • `CompOpenLoad` • `CompShortOpen` • `CompShortLoad` • `CompShortOpenLoad`

doComp	Input	Specifies whether to do ShortOpenLoad compensation: • `0` = Do not do ShortOpenLoad compensation • `1` = Do ShortOpenLoad compensation using a system-level file installed on the system (`cvCALsystem.ini`) • `2` = Do ShortOpenLoad compensation using a user-created file (`cvCAL.ini`)
forceSide	Input	Side used to force DC bias voltage: `"H"` = High (CMTR1H, CMTR2H) `"L"` = Low (CMTR1L, CMTR2L)
doRetest	Input	Specifies whether to remeasure compensation data: • `0` = Do not remeasure compensation data • `1` = Remeasure compensation data once, then reuse that measurement in any additional calls See **Details** for more information
Freq	Input	Frequency (`1e4` to `2e6`)
startV	Input	Sweep start bias
stopV	Input	Sweep stop bias
Vbias	Output	Sweep of voltage bias points
Vpts	Input	Sweep size; must be same as *Ipts*, *CpPts*, *Dpts*, *GpPts*
Ileak	Output	Leakage current
Ipts	Input	Sweep size; must be same as *Vpts*, *CpPts*, *Dpts*, *GpPts*
Cp	Output	Compensated capacitance value
CpPts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *Dpts*, *GpPts*
D	Output	Compensated dissipation factor
Dpts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *CpPts*, *GpPts*
Gp	Output	Compensated conductance value
GpPts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *CpPts*, *Dpts*

Details

This command does the following:

- Verifies input conditions and checks pins
- Checks whether the compensation mode (*comp_mode*) is valid
- Makes connections to CMTR1 and CMTR2
- Uses the high-voltage CMTR (CTMR2) for the *dut* parameter options `dut`, `open`, `short`, and `load`; uses the low-voltage CMTR (CMTR1) for *dut* parameter options `loadEx`, `openEx`, and `shortEx`
- Forces sweep voltage and measures current
- Calls the `hvcv_measure` command to measure *Cp* and *Gp*
- When the *dut* parameter is set to `open` or `openEx`, the routine moves the chuck down, measures, and moves the chuck up again
- Runs compensation according to the compensation mode (*comp_mode*)

This command returns a status:

- `1` = Success
- `–1` = No valid pins
- `–2` = Wrong number of points
- `–3` = No valid compensation mode is specified; valid options are `CompNone`, `CompOpen`, `CompLoad`, `CompShort`, `CompShortOpen`, `CompShortEx`, `CompShortOpenLoad`
- `–4` = No valid DUT is specified; valid options are `dut`, `open`, and `short`
- `–5` = Frequency is out of valid range (`1e4` to `2e6`)
- `–6` = Error with prober chuck moving down
- `–7` = Error in the compensation procedure
- `–8` = Low-voltage pin is used for high-voltage test

Use the *`doRetest`* parameter to save time when you are characterizing a compensation device (open, short, or load) in an automated setting where test macros are repeated multiple times on a wafer or group of wafers. When this parameter is set to `1`, the compensation device is retested once the first time the test macro is encountered. Any further calls to the test macro during the test plan run automatically use the value from the retest.

Example

```
status = hvcv_sweep(pin1, -1, -1, pin2, -1, -1, "dut", "CompNone", 1, "H", 1,
        1e5, 0, 10, Vbias, 11, Ileak, 11, Cp, 11, D, 11, Gp, 11)
```

Performs an 11-point high-voltage C-V sweep.

Also see

hvcv_comp (on page 4-11)
hvcv_measure (on page 4-17)
hvcv_sweep_basic (on page 4-23)

hvcv_sweep_basic

This command does a high-voltage capacitance-voltage (C-V) sweep.

Usage

```
int hvcv_sweep(int high_pin1, int high_pin2, int high_pin3, int low_pin1, int low_pin2,
    int low_pin3, char forceSide, double Freq, double startV, double stopV, double
    *Vbias, int Vpts, double *Ileak, int Ipts, double *Cp, int CpPts, double *D, int Dpts,
    double *Gp, int GpPts)
```

high_pin1	Input	First pin to connect to high-voltage capacitance meter (CMTR) high side
high_pin2	Input	Second pin to connect to high-voltage CMTR high side
high_pin3	Input	Third pin to connect to high-voltage CMTR high side
low_pin1	Input	First pin to connect to high-voltage CMTR low side
low_pin2	Input	Second pin to connect to high-voltage CMTR low side
low_pin3	Input	Third pin to connect to high-voltage CMTR low side
forceSide	Input	Side used to force DC bias voltage: `"H"` = High (CMTR1H, CMTR2H) `"L"` = Low (CMTR1L, CMTR2L)
Freq	Input	Frequency (`1e4` to `2e6`)
startV	Input	Sweep start bias
stopV	Input	Sweep stop bias
Vbias	Output	Sweep of voltage bias points
Vpts	Input	Sweep size; must be same as *Ipts*, *CpPts*, *Dpts*, *GpPts*
Ileak	Output	Leakage current
Ipts	Input	Sweep size; must be same as *Vpts*, *CpPts*, *Dpts*, *GpPts*
Cp	Output	Compensated capacitance value
CpPts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *Dpts*, *GpPts*
D	Output	Compensated dissipation factor
Dpts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *CpPts*, *GpPts*
Gp	Output	Compensated conductance value
GpPts	Input	Sweep size; must be same as *Vpts*, *Ipts*, *CpPts*, *Dpts*

Details

This command does the following:

- Verifies input conditions and checks pins
- Forces sweep voltage and measures current
- Calls the `hvcv_measure` command to measure *Cp* and *Gp*

This command returns a status:

- `1` = Success
- `–1` = No valid pins
- `–2` = Wrong number of points
- `–5` = Frequency is out of valid range (`1e4` to `2e6`)
- `–6` = Error with prober chuck moving down
- `–8` = Low-voltage pin is used for high-voltage test

Example

```
status = hvcv_sweep(pin1, -1, -1, pin2, -1, -1, "H", 1e5, 0,
        10, Vbias, 11, Ileak, 11, Cp, 11, D, 11, Gp, 11)
```

Performs an 11-point high-voltage C-V sweep.

Also see

hvcv_measure (on page 4-17)
hvcv_sweep (on page 4-20)

hvcv_test

This command makes a high-voltage capacitance-voltage (C-V) measurement at a single frequency.

Usage

```
int hvcv_test(int high_pin1, int high_pin2, int high_pin3, int low_pin1, int low_pin2,
    int low_pin3, char *dut, char *comp_mode, int doComp, int doRetest, double Freq,
    double biasV, double *Cp, double *Gp, double *D, double *iCurr)
```

high_pin1	Input	First pin to connect to high-voltage capacitance meter (CMTR) high side
high_pin2	Input	Second pin to connect to high-voltage CMTR high side
high_pin3	Input	Third pin to connect to high-voltage CMTR high side
low_pin1	Input	First pin to connect to high-voltage CMTR low side
low_pin2	Input	Second pin to connect to high-voltage CMTR low side
low_pin3	Input	Third pin to connect to high-voltage CMTR low side
dut	Input	Device under test; valid options: • `dut` = Test the DUT itself with the high-voltage capacitance meter (CMTR) • `open` = Characterize the open device using the high-voltage CMTR; this can be done with the chuck down (pins not in contact with the device) • `short` = Characterize the short device using the high-voltage CMTR • `load` = Characterize the load device using the high-voltage CMTR • `shortEx` = Characterize the short device using the low-voltage capacitance meter (CMTR); this data is used to do `CompShort` compensation of the *`loadEx`* device • `loadEx` = Measure the load device using the low-voltage CMTR to get the expected value of the `loadEx` device • `openEx` = Characterize the open device using the low-voltage CMTR; this data is used to do `CompOpen` compensation of the `loadEx` device
comp_mode	Input	Compensation type: • `CompNone` (use this if you do not want to run any compensation or if the *`dut`* parameter is set to anything other than `dut`) • `CompOpen` • `CompShort` • `CompLoad` • `CompOpenLoad` • `CompShortOpen` • `CompShortLoad` • `CompShortOpenLoad`
doComp	Input	Specifies whether to do system-level compensation: • `0` = Do not do ShortOpenLoad compensation • `1` = Do ShortOpenLoad compensation using `cvCALsystem.ini` • 2 = Do ShortOpenLoad compensation using a user-created file (`cvCAL.ini`)

doRetest	Input	Specifies whether to remeasure compensation data: • `0` = Do not remeasure compensation data • `1` = Remeasure compensation data once, then reuse that measurement in any additional calls See **Details** for more information
Freq	Input	Frequency (`1e3` to `3e6`)
biasV	Input	Voltage bias
Cp	Output	Compensated capacitance value
Gp	Output	Compensated conductance value
D	Output	Compensated dissipation factor
iCurr	Output	Leakage current at the biasV voltage

Details

This command does the following:

- Verifies input conditions and checks pins
- Checks whether the compensation mode (*comp_mode*) is valid
- Makes connections to CMTR1 and CMTR2
- Uses the high-voltage CMTR (CTMR2) for the *dut* parameter options `dut`, `open`, `short`, and `load`; uses the low-voltage CMTR (CMTR1) for *dut* parameter options `loadEx`, `openEx`, and `shortEx`
- Forces *biasV*
- Measures Cp and Gp by calling the `hvcv_measure` command
- When the *dut* parameter is set to `open` or `openEx`, the routine moves the chuck down, measures, and moves the chuck up again
- Runs compensation as specified by the *comp_mode* parameter

This command returns a status:

- `0` = Skip system-level compensation
- `1` = Success
- `–1` = No valid pins
- `–2` = No valid compensation mode is specified; valid options are `CompNone`, `CompOpen`, `CompLoad`, `CompShort`, `CompShortOpen`, `CompShortEx`, `CompShortOpenLoad`
- `–3` = No valid *dut* parameter is specified; valid options are `dut`, `open`, or `short`
- `–4` = Frequency is out of valid range (`1e3` to `3e6`)
- `–5` = Error with PrChuck
- `–6` = Error in the compensation procedure
- `–7` = Low-voltage pin is used for high-voltage test

If compensation data (`open`, `short`, `load`, `loadEx`, `openEx`, `shortEx`) is not available before DUT testing, an error is generated.

This command collects `dut`, `open`, `short`, or `load` data with a high-voltage CMTR on the `dut`, `open`, `short`, or `load` device.

This command collects `openEx`, `loadEx`, or `shortEx` data with a low-voltage CMTR on an `open`, `load`, or `short` structure.

The *doComp* parameter provides a switch that enables or disables system-level compensation. To do ShortOpenLoad compensation using a system-level compensation file that is stored on the system (`cvCALsystem.ini`), set this parameter to `1`. To do ShortOpenLoad compensation using a user-generated compensation file (`cvCAL.ini`), set this parameter to `2`.

Example

```
doRetest = 1
doComp = 1
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, "open", "CompNone", doComp,
        doRetest, 1e5, 0.0, Cp, Gp, D, ICurr)
```

Makes a single-point C-V measurement.

Also see

hvcv_comp (on page 4-11)
hvcv_measure (on page 4-17)
hvcv_test_basic (on page 4-27)

hvcv_test_basic

This command makes a high-voltage capacitance-voltage (C-V) measurement at a single frequency.

Usage

```
int hvcv_test(int high_pin1, int high_pin2, int high_pin3, int low_pin1, int low_pin2,
   int low_pin3, double Freq, double biasV, double *Cp, double *Gp, double *D, double
   *iCurr)
```

high_pin1	Input	First pin to connect to high-voltage capacitance meter (CMTR) high side
high_pin2	Input	Second pin to connect to high-voltage CMTR high side
high_pin3	Input	Third pin to connect to high-voltage CMTR high side
low_pin1	Input	First pin to connect to high-voltage CMTR low side
low_pin2	Input	Second pin to connect to high-voltage CMTR low side
low_pin3	Input	Third pin to connect to high-voltage CMTR low side
Freq	Input	Frequency (`1e3` to `3e6`)
biasV	Input	Voltage bias
Cp	Output	Compensated capacitance value
Gp	Output	Compensated conductance value
D	Output	Compensated dissipation factor
iCurr	Output	Leakage current at the biasV voltage

Details

This command does the following:

- Verifies input conditions and checks pins
- Makes connections to CMTR2
- Forces biasV
- Measures Cp and Gp by calling the `hvcv_measure` command

This command returns a status:

- `1` = Success
- `−1` = No valid pins
- `−4` = Frequency is out of valid range (`1e3` to `3e6`)
- `−5` = Error with PrChuck
- `−7` = Low-voltage pin is used for high-voltage test

Example

```
status = hvcv_test(pin1, -1, -1, pin2, -1, -1, 1e5, 0.0, Cp, Gp, D, ICurr)
```

Makes a single-point C-V measurement.

Also see

hvcv_measure (on page 4-17)
hvcv_test (on page 4-25)

Keithley Instruments
Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-800-935-5595 • www.tek.com/keithley

KEITHLEY
A Tektronix Company