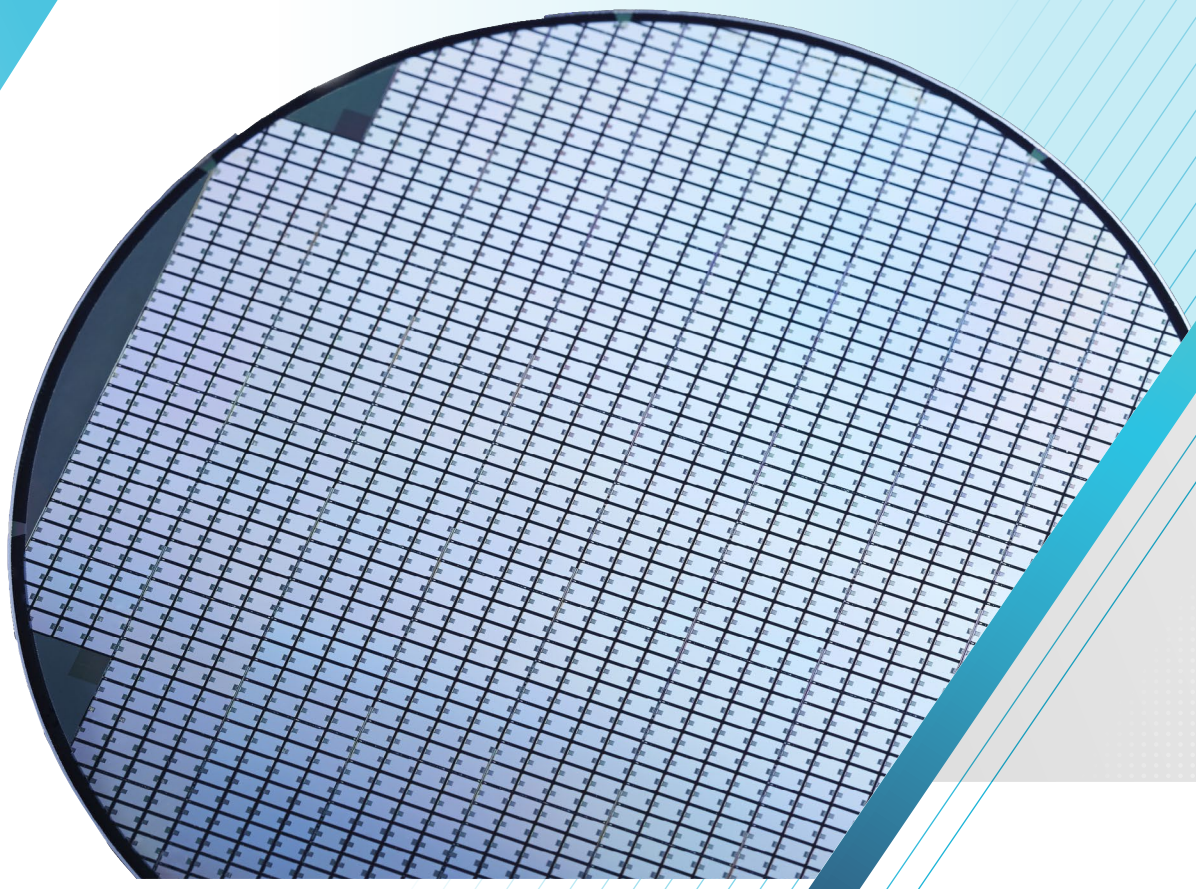
High Voltage Wafer Testing in a Production Environment with the HV S540 Parametric Test System

APPLICATION NOTE



KEITHLEY
A Tektronix Company

Tektronix®

Due to the complexities typically associated with high voltage (HV) wafer-level testing, such as instrumentation setup, cabling, probing, automation, and safety, on-wafer HV testing is usually limited to characterization labs or manual benchtop setups that are separate from a fab's standard production workflow. This application note contains implementation details on the integration of HV testing in a production environment.

Keithley has developed several measurement techniques and approaches that enable automated HV wafer level characterization on multiple pins without sacrificing low voltage performance or throughput. These techniques include integration methods that allow sensitive transistor characterization and low current leakage tests to run in the same process flow as HV breakdown and HV capacitance tests. For example, in one automated test sequence, the transistor I_{off} current is measured, followed by the threshold voltage (V_{th}) measurement. Next, the drain current (I_{on}) is measured when both the gate and drain are biased above 1 kV. Then, capacitance measurements are performed with a 2 kV bias level. Last, breakdown tests are run at 3 kV levels.

Keithley has also developed a run-time open/short/load impedance compensation technique that supports making accurate on-wafer HV capacitance measurements. This application note will explore these and other HV measurement issues, as well as share results and experiences in the emerging field of HV wafer-level testing.

Why HV Testing Is Necessary

Power semiconductor transistors are commonly used in a variety of industries, including home appliances and automobiles, as well as in various power applications. Demand for faster, more powerful devices and switches that can handle more current and voltage will continue to grow.

Materials

A variety of semiconductor devices can be used to control power. See **Figure 1** for an overview of the history of power semiconductor devices. Initially, they were limited to Si-based bipolar devices like thyristors and diodes, in such applications as power rectifiers. Later, bipolar power BJTs, GTOs, IGCTs, IGBTs, and power MOSFET transistors were introduced to address power handling needs.

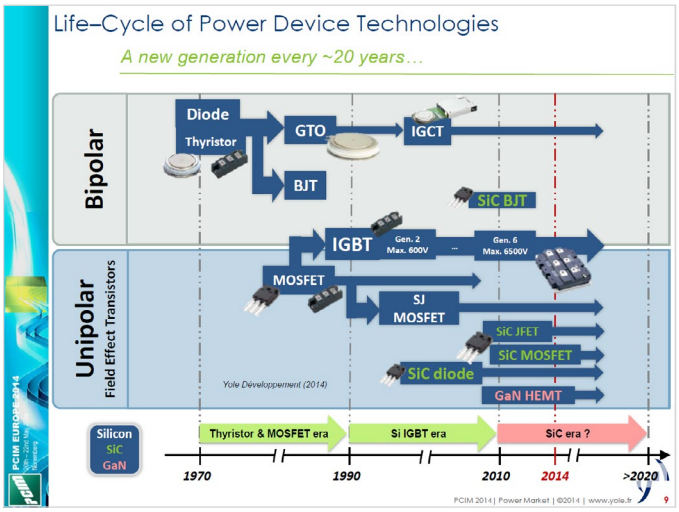


Figure 1: History of power devices [1].

Continuous technology progress and new market drivers require new materials. Sometimes, power devices are characterized using the figure of merit (FOM), which is defined as $R_{dson} * Q_{gate}$, the product of on-resistance multiplied by the gate charge required for transistor switching. Si-based devices have reached their material limit, and other materials with higher mobility (GaN) or better heat conductivity and superior electrical properties (SiC) are used for the new generation of power devices (**Figure 2**). SiC and GaN have higher bandgaps and significantly higher breakdown fields than Si. The high mobility of GaN devices led to the development of High Electron Mobility Transistors (HMETs).

Property	Units	Si	GaAs	4-SiC	GaN
Bandgap	eV	1.1	1.42	3.26	3.39
Relative dielectric constant	-	11.8	13.1	10	9
Electron mobility	cm^2/Vs	1350	8500	700	1200-2000
Breakdown field	10^6 V/cm	0.3	0.4	3	3.3
Saturation electron velocity	-	1	1	2	2.5
Thermal conductivity	K	1.5	0.43	3-3-4.5	1.3

Figure 2: Electrical and thermal properties of materials for power devices [2].

Any material technology and device developments require extensive characterization, first in the labs on test benches, and later, as technology matures, on the production floor with automated test systems. Before Keithley Instruments developed the HV S540 system, no automated test system available on the market was capable of HV parametric testing.

HV Characterization Parameters

To offer a clear description of the HV S540 system, it's important to understand the parameters typically measured to characterize power devices. For useful tutorials on power device characterization parameters, refer to [3], [4] and [5].

Maximum Ratings: $V_{(br)dss}$, V_{gs} and I_d, I_{dm}

Maximum voltage ratings are defined as the maximum voltage that can be applied before avalanche breakdown in the transistor or when the gate is damaged. V_{dss} is the maximum voltage between drain (collector) and source, with the transistor in the off-state. V_{gs} is the maximum voltage between gate and the source (emitter) before the gate can be damaged. The S540 system can perform breakdown tests at up to 3 kV.

I_d is the maximum continuous drain (collector) current that a device can sustain with no damage. It is controlled by the R_{dson} and the thermal power dissipation capability of the DUT.

I_{dm} is the maximum pulse current rating that the device can handle in the pulse mode. Usually, it is larger than Id and is determined by the pulse duration and shape.

I_{dss}

This is drain (collector) leakage current of the off-state at a specified drain voltage. Current usually is small and can be accurately characterized by the S540 system for V_{ds} up to 3 kV and leakages down to tens of picoamps.

$V_{gs(th)}$ Threshold Voltage

Threshold voltage measurement for power devices is the characterization of the off/on transition. Standard V_{th} techniques, such as extrapolated V_t (based on maximum transconductance evaluation), cannot not be used easily here because of the high current required. Instead, on the S540 system, threshold voltage can be obtained as the gate-to-source (or gate-to-emitter) voltage, which yields the known value of the drain (collector) current. The amount of the critical current varies, and it is based on the characterized device. Threshold current values as low as 250 µA or 1 mA are common. For standard LV (low voltage) transistors, trigger currents between 0.1 µA and 1 µA are commonly used.

R_{dson}

Probably one of the most important parameters of a power transistor is R_{dson}, the drain-to-source resistance in the on-state. This parameter controls maximum current in the on-state. Values for R_{dson} usually are small, in the range of tens of milliohms, and include not only the resistance of the channel and drift region, but also, in the test environment, parasitic resistance of the pads, contact resistance, and external interconnect resistance. Resistance is measured at relatively small drain voltages (< 20 V) as a function of drain current. Drain current can be as high as several tens of amps, depending on the application. With the S540 system, R_{dson} is measured using a 2636B SMU SourceMeter® Instrument, with drain current of up to 1.5 A.

Transfer Characteristics

Transfer characteristic usually is defined as the electrical characteristic relating drain current to gate voltage. For non-power transistors, the ratio of change in the drain current to the change in the gate voltage is defined as transconductance (gm), and determines the amplification of the device. Power devices are used primarily as switches, and Id/Vg data determines their transfer characteristics. Depending on the power device, this current can be in the tens of amps.

Gate Charges: Q_{gs}, Q_{gd}, and Q_g

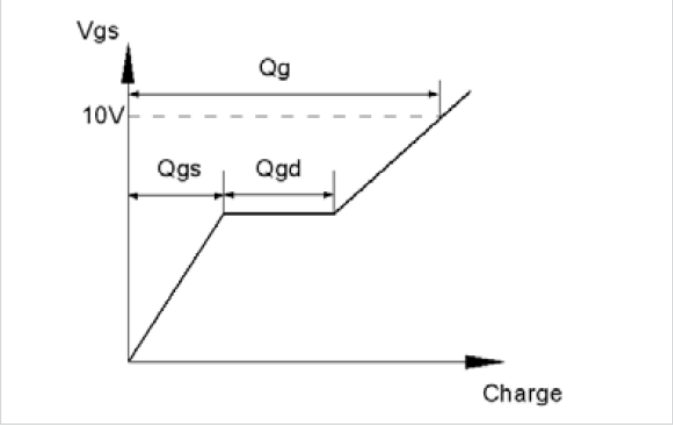


Figure 3: Gate Charges: Q_{gs}, Q_{gd}, and Q_g.

Gate charge is charge accumulated at the gate sufficient to turn on the device. It depends on the parasitic capacitance and determines switching time and the energy required to switch on the transistor. Devices start switching on at the beginning of the plateau, and completely turn on at the

right side of the plateau. To perform this measurement, the instrumentation setup should be able to sustain high current (I_{don}).

Device Capacitances: C_{gd}, C_{gs}, C_{ds}, and C_{iss}, C_{oss}, C_{rss}

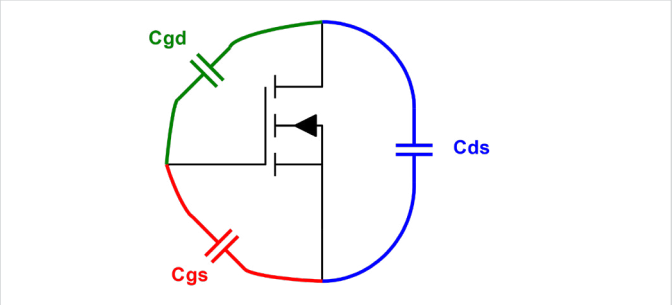


Figure 4: Transistor capacitances: C_{gd}, C_{gs}, and C_{ds}.

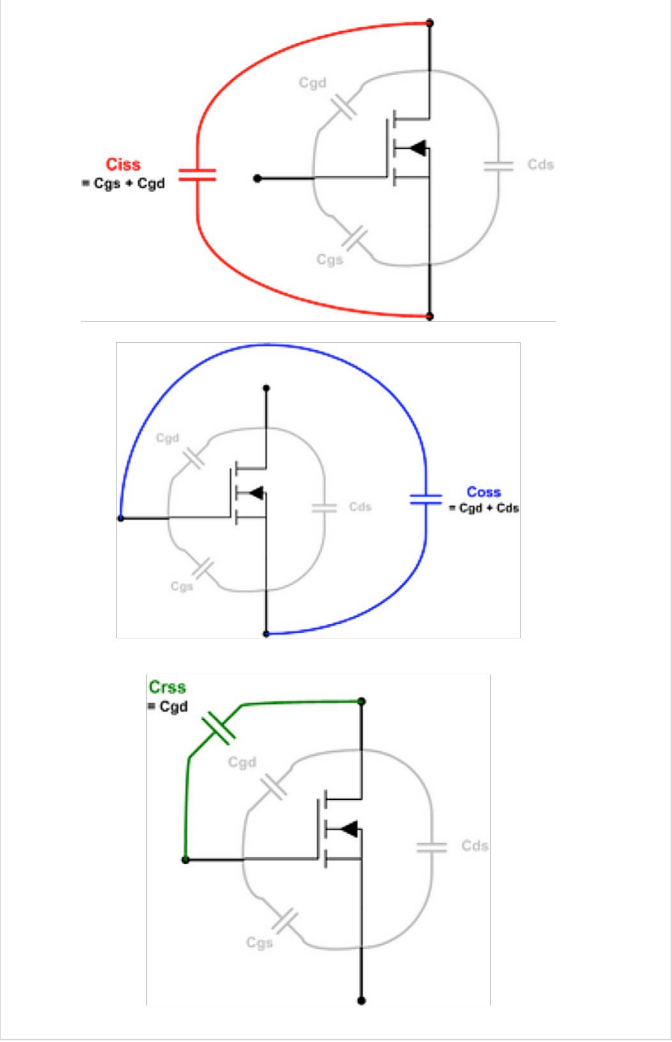


Figure 5: C_{iss}, C_{oss}, and C_{rss}.

Switching speed of the power devices is controlled by the device capacitances. This includes capacitances between gate and drain (C_{gd}), gate and source (C_{gs}), and drain and source (C_ds). The combination of these capacitances (C_{iss}, C_{oss}, C_{rss}) characterizes input and output transient performance. C_{iss} is equal to C_{gs} plus C_{gd}. It is the capacitance that has to be charged for the transistor to be switched on, and is the one that controls the speed of the turn-on switching. C_{oss} = C_{ds} + C_{dg} is an output capacitance. It affects circuit resonance and dynamic behavior. C_{rss} (C_{gd}) is the reverse transfer capacitance, sometimes called Miller capacitance. This capacitance controls turn-off timing.

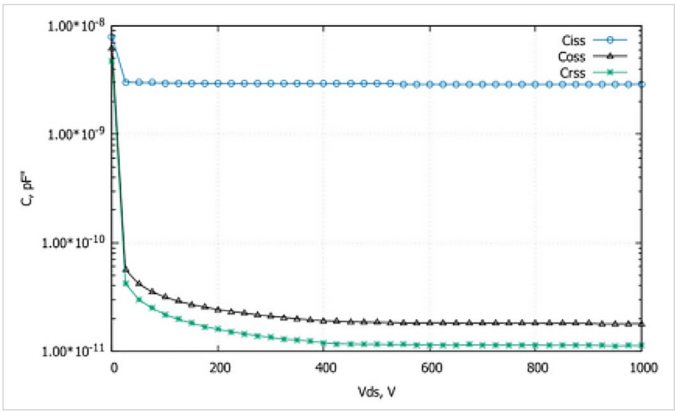


Figure 6: C_{iss}, C_{oss}, and C_{rss} as a function of V_{ds}, as measured by the S540 system.

C_{iss}, C_{oss}, and C_{rss} are measured in the off-state, when V_{gs} = 0, for different drain (V_{ds}) biases (**Figure 6**).

Characterization and Production PCM Systems Requirements

The S540 system was designed for two scenarios: the first is its use in process integration labs; the second is process control monitoring (PCM) with automated testing. Each scenario has slightly different requirements. In process integration, all parameters usually need to be characterized, and complexity in the setup and testing is acceptable. The amount of data taken, throughput, and simplicity usually are not significant factors. The flexibility of the test setup is the primary factor. In PCM, only a subset of parameters is collected. Throughput, simplicity and automation are the primary factors.

HV S540 Tests

Breakdown Test

HV breakdown tests usually are used to outline the usage boundary of the device and to evaluate the stability of the power devices in various scenarios. Breakdown happens in various media, but most of the interest is in the breakdown of material junctions, such as at the junction of the drain to gate or substrate. Breakdown voltages depend on the materials and the structure design. The S540 system is designed to evaluate breakdown voltages of up to 3 kV.

Pad Pitch Distance Limitation

For packaged devices, isolation between HV voltage terminals is relatively easy and usually is achieved through the use of greater separation and insulating materials.

The situation is more complicated when the device is still on the wafer as in PCM (Process Control Monitoring) testing. Minimizing the amount of wafer real estate needed for test structures is in conflict with the continuous drive for higher breakdown voltages. Usually, the minimum pad pitch (the distance between two adjacent pads) should be sufficient to prevent arcing in the air. Maximum sustainable electrical field in the dry air can be estimated be close to 40 kV/cm, or 4 V/µm. To sustain 1.8 kV, for example, pad pitch size has to be equal to or greater than 425 µm.

Electrical Field with Needles, Spherical, Cylindrical Field Concentration, Polymer Protection of the Needles

The maximum electrical field between flat metal pads can be estimated to be close to V/d. The electrical field around the metal needle is defined by the cylindrical geometry of the needle. The electrical field will be at its maximum close to the needle, and can be estimated to be close to V/R, where R is the radius of the needle. Given that the radius of the needle (25 µm, for example) is much smaller than the HV pad pitch distance (300–600 µm) typically used, there is amplification of the electrical field at the needles, by the ratio of d/R (see **Figure 7**). For this particular case, the maximum electrical field will be 12 to 24 times larger than the field estimated by dividing the voltage by the pad distance. This illustrates the potential challenges in the breakdown tests on the wafer, when sufficient pad pitch size can still lead to air ionization and air breakdown. Polymer or any other isolating coating on the metal surface of the needles can suppress this effect, and this technique is used by some probe card makers (Celadon Systems, for example).

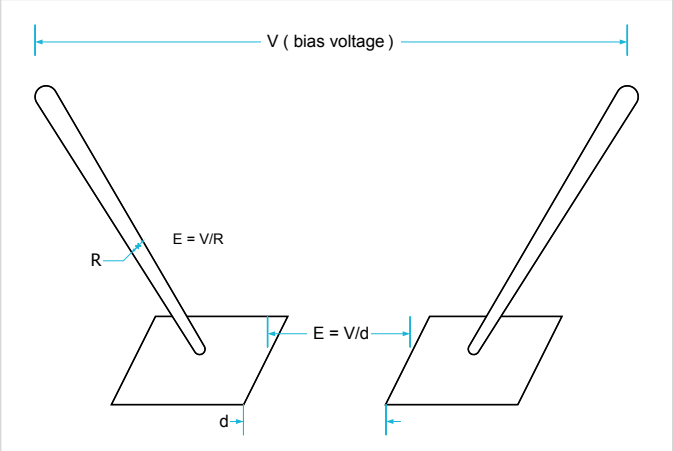


Figure 7: Pin to pad connection geometry.

Wafer Surface Damage

Another complication of HV on-wafer testing is caused by the presence of moisture on the surface of the wafer. Breakdown test is supposed to test maximum voltage in the bulk of the semiconductor material; however, surface breakdown often prevents reaching these voltage levels.

Surface moisture significantly decreases the electrical strength of the air, and moisture has to be suppressed by the environmental control. The most common technique is to inject nitrogen or CDA (Clean Dry Air) into the test area. This allows for somewhat increased electrical strength of the air; however, all dry gases have comparable electrical strength and at high enough voltages at given device geometries, breakdown is unavoidable and happens before breakdown in the bulk of the semiconductor material (**Figure 8**). Pins to the left are for source and gate connections; pads to the right are drain/bulk connections. The distance between source and gate pads is about 600 µm. Visual inspection of the damage indicates air breakdown between source/gate needles to the drain connection on the right.

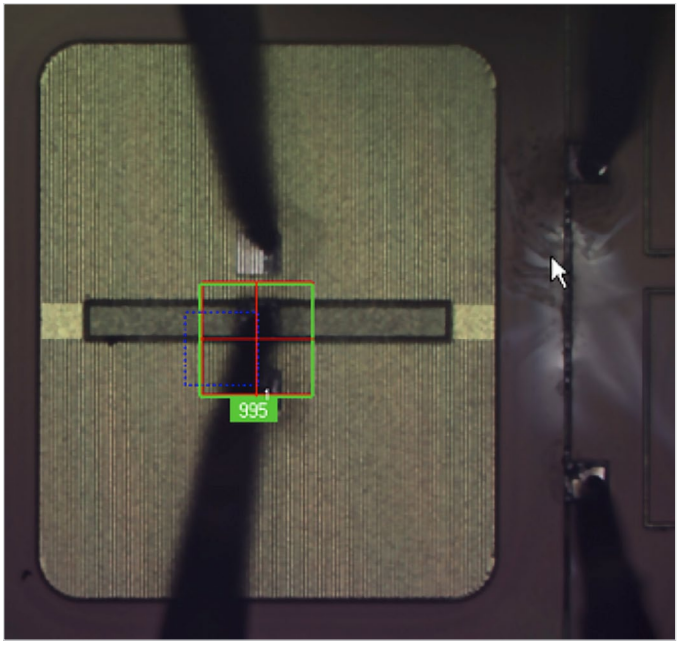


Figure 8: HV surface breakdown.

Fluorinert vs. CDA (Clean Dry Air) / N_2

The breakdown capabilities of the environment can be increased through the use of special electronic liquids from the group of chemicals named collectively known as “Fluorinert.” Examples of these liquids include Fluorinert FC-40 ($C_{21}F_{48}N_2$) from 3M or HT-110 ($CF_3O[-CF(CF_3)CF_2O-]_x(-CF_2O-)_yCF_3$) from Galden. The dielectric strength of these liquids is at least 16 kV per mm, or 16 V/µm, which is about 4× the strength of dry air. Fluorinert is a transparent electronic liquid that evaporates leaving no traces behind. Fluorinert is a good tool for achieving high voltages and provides excellent electrical isolation; however, its use is rarely acceptable in clean room or even characterization lab environments. Any production-worthy solution should exclude usage of Fluorinert, but measures must be taken to eliminate the moisture from the air by the use of either N_2 or CDA (Clean Dry Air).

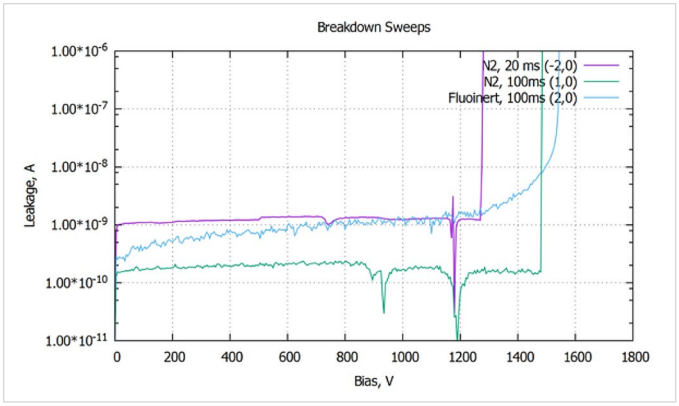


Figure 9: Breakdown sweep.

Figure 9 shows examples of the breakdown test, when voltage is ramped up to 1800 V at two different ramp rates, 20 ms and 100 ms per step. High ramp rate (slow delay time) increased the measured current from 100 pA to 1 nA. At the higher ramp rate (20 ms per step), most of the current is the displacement current (~1 nA).

Another interesting feature of this data set is that it shows the susceptibility of the test setup used (probes, environment and wafer) to the surface discharge. The expected breakdown voltage of the structure is about 1750 V. These voltage levels (~1700 V) are reliably achieved in experiments with Fluorinert use. In tests with no Fluorinert, surface breakdowns occur occasionally at lower voltages (1200–1400 V). These breakdowns occasionally are preceded by current spikes at even lower voltage (1000–1100 V), which do not cause runaway damage to the device under test. Presence of these spikes can be used to determine a test “fail” scenario when device spacing is not sufficient to have a breakdown in the bulk. One of the sweeps (**Figure 9**) was performed with the Fluorinert encapsulating the pads, pins, and surrounding area on the wafer (**Figure 10**).

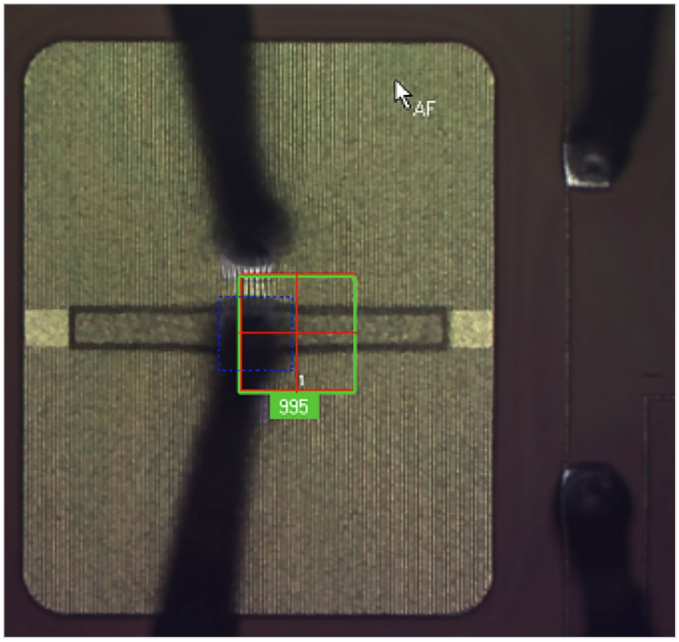


Figure 10: Test area on the wafer encapsulated in the protective Fluorinert liquid.

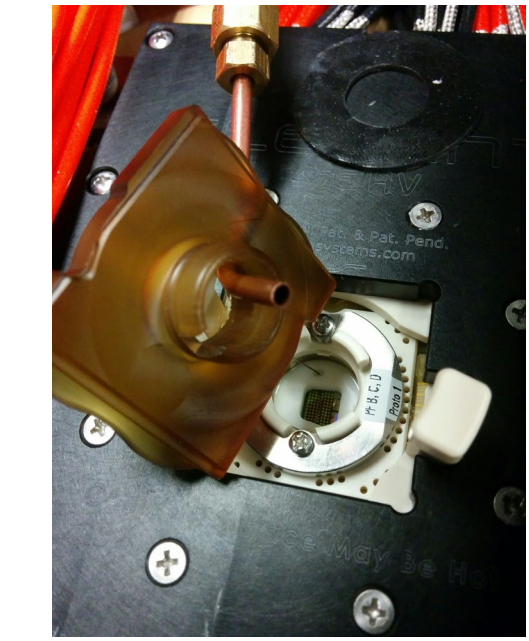
Breakdown Measurement with Pressurized Cavities

Fluorinert works well in that it suppresses all discharges in the air and allows measuring the intrinsic breakdown of the devices; however, it is not a production solution. There is an alternative to the use of electronic insulating liquids: pressurized cavities above the probe needle area (**Figure 11**).

An encapsulating cover on top of the probe card adaptor creates a pressurized cavity filled with N_2 or CDA, and forces a stream of the gas along the probe needles.

It turns out that the pressurized cavity approach works just as well as the Fluorinert approach, producing the same breakdown voltages and I-V data (**Figure 12**). The pressurized cavity yields the same MEAN and STDEV for the breakdown voltage as the Fluorinert does.

The use of pressurized cavities to suppress discharges is relatively common in the industry, and the usual explanation is that this suppression is the result of excessive pressure, which increases the breakdown voltage.



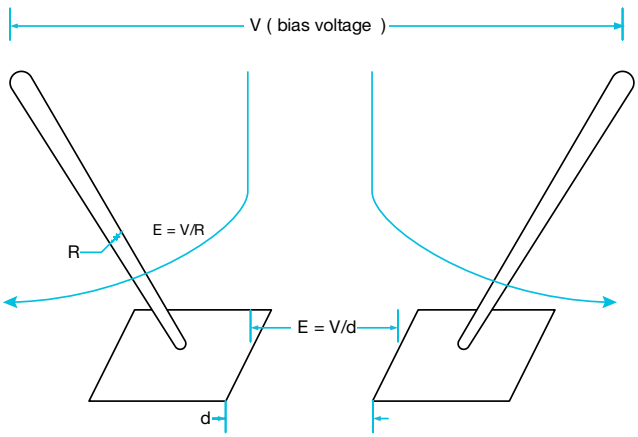
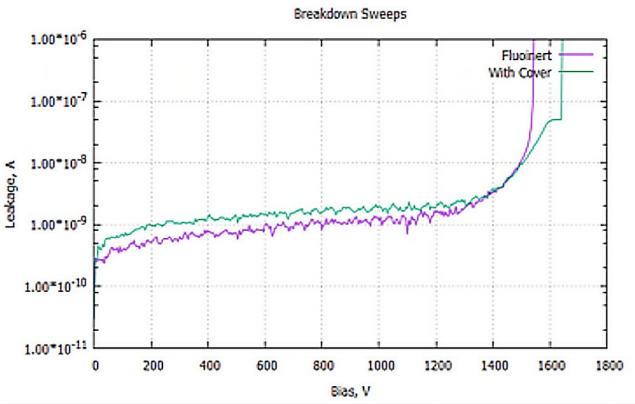


Figure 11: Pressurized cavity on top of probe card insert (Celadon Systems).



	N2 Stream Only	N2 stream and Pressurized cavity	With Fluorinert
MAX	1470	1530	1510
MIN	1285	1435	1445
MEAN	1382	1496	1465
STDEV	70	29	24

Figure 12: I-V data collected with Fluorinert and pressurized cavity.

Energy Dissipation and the Needle Tip

Another aspect of the breakdown test with high voltages (>1000 V) relative to LV (<200 V) is higher energy dissipation, which has an impact on the durability and stability of the pins and pin contacts. First, let's estimate the total electrical energy, which is dissipated during the breakdown. It is given by:

$$E = \frac{1}{2} C V^2$$

Equation 1: Energy of the capacitance

Here, C is the cable capacitance and V is the maximum breakdown voltage. For 3 m-long Kelvin triaxial cables, capacitance can be estimated to be close to 1 nF. With breakdown voltage of 1700 V, the total energy stored in the cable parasitic capacitance is about 1.4 mJ, which is 400 times larger than the energy dissipated during 85 V breakdown. This is an energy difference of several orders of magnitude and presents a very different usage scenario for the probe tips. During breakdown, this energy will dissipate in different parts of the test setup, for example, at the tested drain/substrate junction or at the pin-to-pad contact. Any bad contact, small prober overdrive, or the gap between probe tip and the pad would lead to the concentration of the energy dissipation to the probe tip, which may lead to pin degradation or even melting. Assuming tungsten needles are used, and with the specific heat of tungsten (0.134 J/gm K), density of tungsten of 19.2 gm/cm^3, and volume of the impacted tip defined by the radius of 50 µm (or 5e-3 cm), a simple estimate yields 10,000 K. For 85 V breakdown, for example, heating would be limited only to 25 K, or 400 times smaller. Of course, this calculation overestimates the heating of the needle because it assumes that all energy released would go to the tip; nevertheless, this estimate does illustrate the potential issue, and the fact that, at high voltages, pins can be easily melted, oxidized, or degraded.

The design of the Prober Card Adapter (PCA) with built-in elements may help to limit the breakdown current and damage to the needle tips.

High Voltage C-V Measurements

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

Bias tees allow mixing high DC bias voltage with an AC signal. One drawback of using bias tees is that it causes degradation of the AC pathway and increased measurement errors. To solve this problem, compensations can be applied to negate the effect of bias tees in high voltage measurement applications.

The Keithley High Voltage Library (HVLib) allows making high voltage C-V measurements using pre-programmed compensation factors and more accurate user-generated device-level compensation factors.

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)

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; an automatically balanced bridge on the low side of the CMTR measures the amplitude and phase of the current.

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 (*Cp* and *Gp*) 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 chosen.

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 3 to 4 percent.

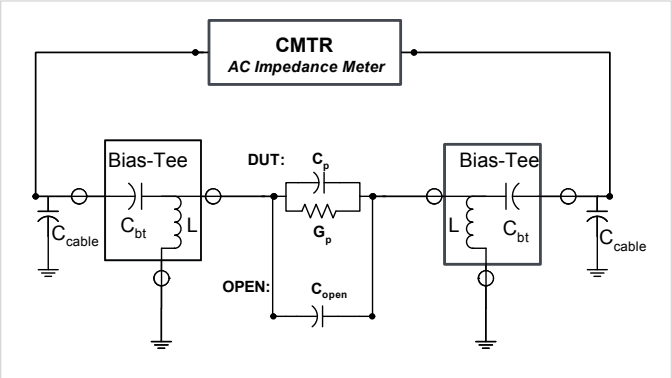


Figure 13: Two-terminal HV capacitance measurement.

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

$$Z_{meas} = k * Z_{dut} \parallel Z_{open} + Z_{short}$$

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, $2 \times (1/jw \times C_{bt}) \times (1+ C_{cable}/C_{bt})$

This equation can be used 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 being measured.

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 evaluating 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 **Figure 13**.

Input (C_{iss}) and output (C_{oss}) capacitances are measured in a similar way. Each of the three terminals (gate/drain/source) needs to have an independent DC bias. In the AC frequency domain, two terminals are AC-tied together 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 CMTRL (sense terminal) and the source pin is AC-tied with the drain to CMTRH (High/AC-drive side). AC impedance is measured between Low (gate) and High (drain and source), and $C_{iss} = C_{gd} + C_{gs}$.

Guard Challenge for C_{rss}

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 it has reduced accuracy at 1 MHz (refer to the S540's specifications for details). This only affects the quality of the Crss 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. This command can be used without changes or it can be copied under a different name and customized for a specific application. It can be used to measure C_{iss} and C_{oss} parameters and individual capacitances (C_{rss}, C_{ds}, and C_{gs}). 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, compensation data from the open, short, and load devices must first be collected.

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
- Device-level compensation

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 **Appendix 2**.

System-level compensation

System-level compensation applies compensation factors to negate the effect of bias tees in the system. It uses a single pre-programmed 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 on-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 the *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.

See **Appendix 1** for a procedure overview for system-level compensation.

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. Compensation factors should be created just before testing a device. This results in more accurate high voltage capacitance-voltage (C-V) measurements.

Several ways to acquire device-level of compensation factors are discussed in the following topics. For examples of specific usage scenarios, see **Appendix 2**.

Device-level compensation allows applying 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 the S540 system is configured with a high voltage capacitance meter (CMTR) and a low voltage CMTR, the low voltage CMTR (CMTR1) can be used to measure the expected value of the load.

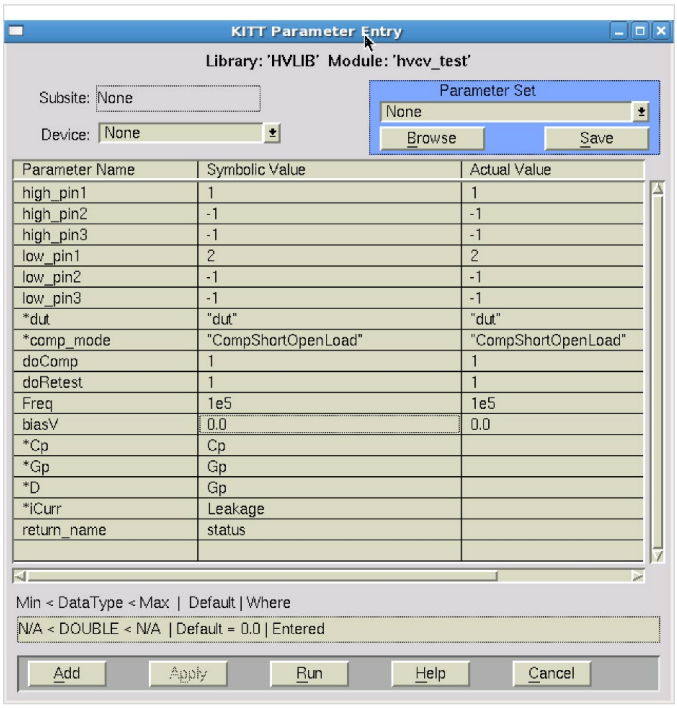


Figure 14: Example of the *hvcv_test* in KITT.

See **Appendix 1** for a procedure overview for device-level compensation.

Effectiveness of compensation models

Table 1 shows the effectiveness of the different compensation models. A 103 pF capacitor was measured in different scenarios. This included measurements performed with the S540 system and no correction, with *Open* correction only, *Short*, *OpenShort*, *OpenShortLoad* corrections, etc. With no compensation, the S540 system measures 106.6 pF (3.4% error). Just '*Open*', '*Short*' or '*OpenShort*' compensation does not decrease the error. The table indicates the two load devices that were used: 1 nF and resistive load 11 kΩ. For all these loads, '*ShortOpenLoad*' compensation works well, with total error reduced to less than 0.2%.

	Load Device	Cp	Error
CAPACITANCE		1.03E-10	
S540		1.07E-10	3.40%
Open		9.97E-11	-3.30%
Short		1.07E-10	3.80%
OpenShort		1.00E-10	-2.90%
OpenLoad 1nF	1 nF	1.06E-10	2.90%
ShortOpenLoad 1nF	1 nF	1.03E-10	0.20%
Load 11 kΩ	11 kΩ	1.09E-10	5.70%
OpenLoad 11 kΩ	11 kΩ	1.03E-10	0.20%
ShortOpenLoad 11 kΩ	11 kΩ	1.03E-10	0.00%

Table 1: Correction effectiveness for different compensation models (*Load*, *OpenLoad*, and *ShortOpenLoad*).

A short description of the test code, which enables run-time compensation per any pin, together with an example test routine (*hvcv_test*), is provided in **Appendix 3**, which provides details for the HV application test library (HVLib).

The *hvcv_3term* routine from the HVLib was designed to take advantage of the HV matrix's capability to automate three-terminal capacitance measurements. This test is provided as both a verified test and as a template for customization. It can be used to measure such parameters as C_{iss}, C_{oss} and individual capacitances such as C_{rss}, C_{ds}, and C_{gs}. The routine will use system-level compensation when the "*doComp*" flag is enabled. Also, the routine will perform additional device-level compensations, including most of the known compensation models. For example, to run "*ShortOpenLoad*" device-level compensation, data needs to be collected from the Open, Short and Load devices.

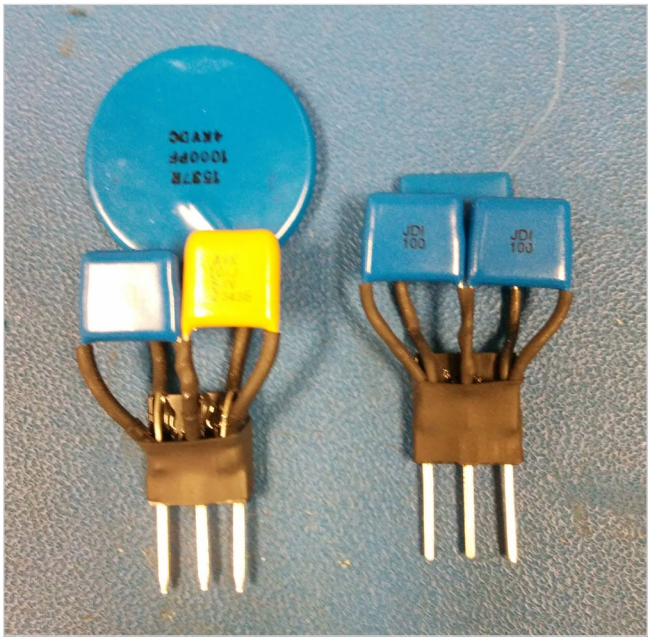


Figure 15: Test parts for three-terminal C-V evaluation.

The S540 system's three-terminal HV C-V procedure was evaluated using discrete parts, similar to the ones shown in **Figure 15**. Test capacitors were pre-evaluated and were built into several transistor configurations, as shown in **Table 2**.

100KHz			Expected	Actual	% Error
Cgs	9.73000E-10	Ciss	9.8381E-10	9.9800E-10	*1.4%*
Cds	1.02340E-10	Coss	1.1315E-10	1.1800E-10	*4.3%*
Cgd	1.08100E-11	Crss	1.0810E-11	1.1400E-11	*5.5%*
100KHz			Expected	Actual	% Error
Cgs	9.46000E-10	Ciss	1.0496E-09	1.0490E-09	*-0.1%*
Cds	9.39000E-10	Coss	1.0426E-09	1.0470E-09	*0.4%*
Cgd	1.03600E-10	Crss	1.0360E-10	1.0180E-10	*-1.7%*
100KHz			Expected	Actual	% Error
Cgs	4.66000E-09	Ciss	4.6709E-09	4.7600E-09	*1.9%*
Cds	1.08610E-11	Coss	2.1740E-11	2.2700E-11	*4.4%*
Cgd	1.08790E-11	Crss	1.0879E-11	1.0670E-11	*-1.9%*

Table 2: Three-terminal HV C-V evaluation data.

Table 2 shows data for three capacitor configurations (C_{gs}, C_{gd}, and C_{ds}). Three configurations were selected for various ratios of C_{gs}:C_{ds}:C_{gd} capacitances. Ratios tested here: 1:0.1:0.01, 1:1:0.1 and 1:0.02:0.02. The larger the ratio, the less accuracy can be expected from smaller capacitance measurements. Expected values of C_{iss}, C_{oss} and C_{rss} are calculated and compared against measured values. The measurement procedure included system-level *OpenShortLoad* compensation, and DUT-level *Open* compensation. The last column shows the deviation of the measured values from those expected.

Figure 6 shows the HV C-V data for C_{rss}, C_{iss} and C_{oss} measured at various drain biases, ranging from 0 V to 1000 V. Data were collected using the S540 system, with all interconnect changes and AC guarding setup performed by the HV matrix and under automated software control.

Automated Test Sequence

LV, HV and C-V measurements to any pin

The S540 system was designed to run standard parametric measurements (LV characterization, C-V, etc.) in a single pass together with HV measurements (HV breakdown test, HV C-V, three-terminal HV C-V).

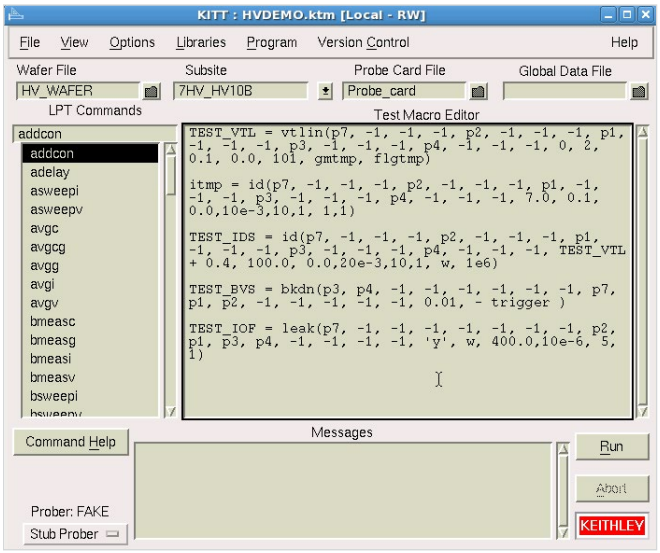


Figure 16: An example of an S540 test sequence.

Figure 16 shows an example of a test sequence performed on one of the subsites (scribe test module). This sequence includes standard LV transistor characterization (threshold voltage: *hvlib_vtlin*, drain current: *hvlib_id*) and HV tests (breakdown tests: *hvlib_bkdn*, *hvlib_bvswep*; and leakage test: *hvlib_leak*). The names of these tests and library were taken from one of the demonstration tests. Due to the destructive nature of the HV tests, these tests usually are done at the end of the sequence. From the user's perspective, there is no differentiation between HV, LV and C-V tests. The system handles all the switching, interconnect issues, protection of the user and instrumentation, and interfacing to the wafer with the probe card adaptor.

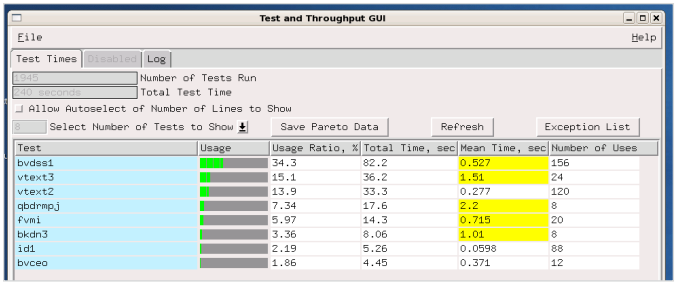


Figure 17: Throughput analysis.

The S540 system is designed as a production system. In addition to the requirement to run all tests (LV, HV and HV C-V) in a single pass, it's also desirable to run the tests as fast as possible to satisfy production throughput requirements. HV tests often take longer to run due to the capacitive nature of the DUT and higher required voltage values. Nevertheless, because throughput is always important, the S540 system includes a tool for throughput analysis and optimization. **Figure 17** shows a throughput analysis performed on production log data using the *ptlog* utility.

Low Voltage/Sensitive Measurement through HV Matrix and HV Interconnects

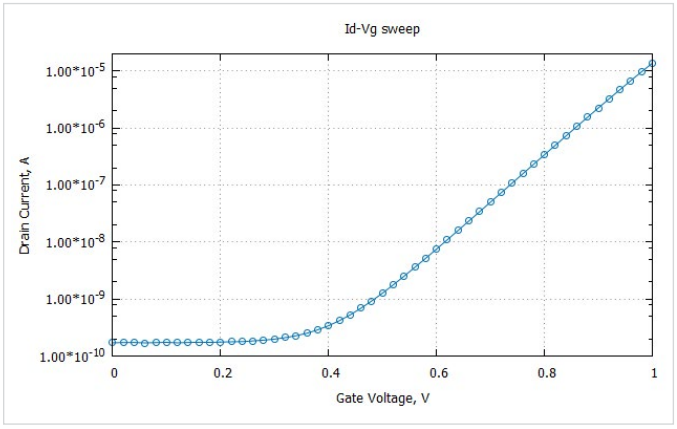


Figure 18: Id-Vg data. Vds = 0.1 V.

The addition of the 3 kV HV capability to this production system does not degrade the accuracy of its sensitive LV parametric measurements. The S540 system includes both HV instrumentation (2657A High Power System SourceMeter® SMU Instrument) and LV sensitive SMUs (2636B Dual-channel System SourceMeter SMU Instrument with 0.1 fA sensitivity). **Figure 18** shows Id/Vg data taken at small drain voltage (0.1 V) and demonstrates low current capability. I_{doff} current for this device is about 100 pA. With the chuck down,

the current would drop to the noise floor of the system, which is in the range of less than 10 picoamps.

S540 System Configurations

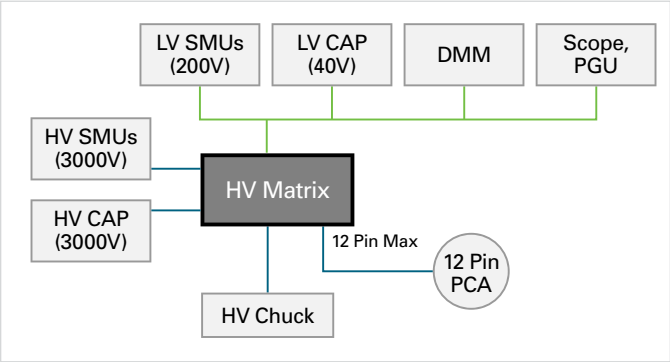


Figure 19: 3 kV HV system.

The S540 system comes in two versions: “3 kV only system” and “Hybrid” system. The 3 kV system (**Figure 19**) contains a 12×12 HV matrix, with 12 input rows and 12 output columns. Each output is 3 kV-enabled, and can be connected either to an HV Probe Card Adapter or an HV chuck connection. It can also be used to provide HV bias to up to three HV bias tees.

3 kV HV System Only, HV matrix with 12 pins

The S540 system can be configured with up to three bias tees for HV C-V measurements. As discussed previously, capacitance measurements made with bias tees have to be corrected to account for their presence. This can be done on the system level and per specific device on the user level. In addition to the CMTR (installed in the 4200-SCS Parametric Analyzer), the S540 system contains two HV SMUs (2657As), and up to six 2636B SMUs. Protection modules allow using low voltage SMUs (like the 2636B with <200 V) together with HV for sensitive measurements. A low patch panel provides a ground reference point for the all instruments. HV, LV, and C-V signals can be provided to any of the probe pins.

Hybrid 3 kV HV and Sensitive System

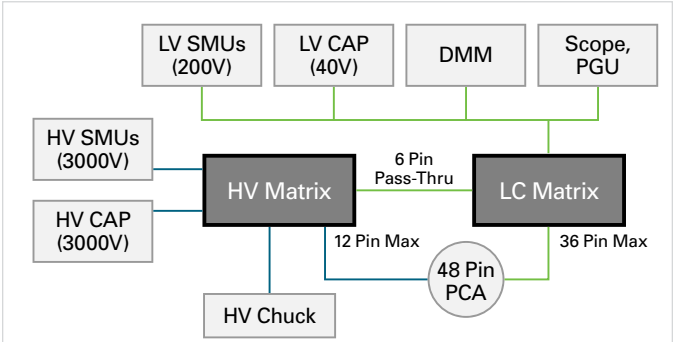


Figure 20: Hybrid system: 3 kV + LV.

The 3 kV HV matrix has 12 HV pins. For example, for a system with three bias tees, with an HV chuck and a total 12 HV matrix columns, there are eight HV pins available for the Probe Card Adapter (PCA). For some customers in a production environment, it is desirable to run a larger set of output pins. The hybrid system (**Figure 20**) was designed for this requirement. In addition to a few (8–12) HV pins, an S540 hybrid system allows using up to 36 standard LV pins for automated testing.

Appendix 1: Procedure overviews

System-level compensation

The following summarizes what happens during system-level compensation:

- Pre-programmed compensation factors are retrieved, or new compensation factors can be created at runtime using a custom fixture and load standard (probe card with connection for the discrete capacitors). The default procedure uses pre-programmed 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 **Appendix 2**.

Recreating system-level compensation factors

To recreate system-level compensation factors, use the *hvcv_genCompData* command. This command prompts the user to connect open, short, and load devices to a selected pin pair. A custom fixture or probe card that allows inserting a load standard and short device must be used. **Figure 21** 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.

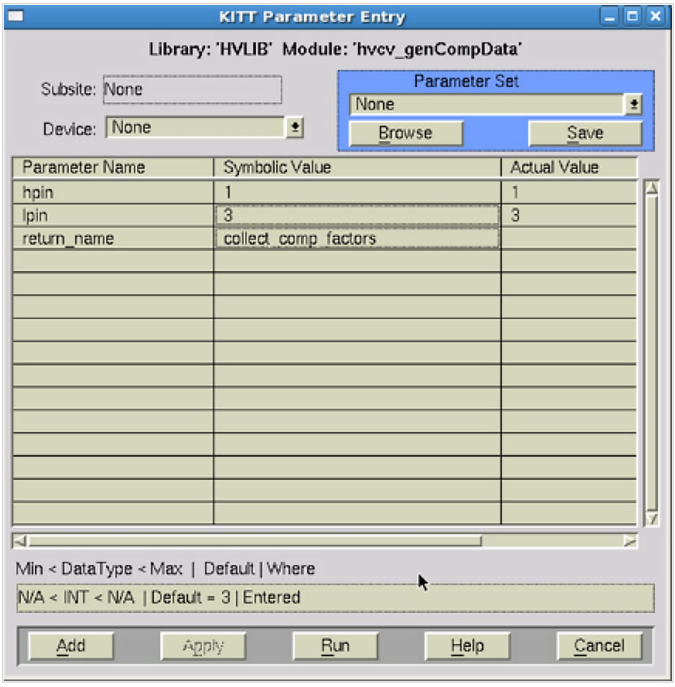


Figure 21: Compensation data collection function: *hvcv_genCompData.*

System-level compensation uses the algorithm shown in **Figure 22**. 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 **Figure 22**.

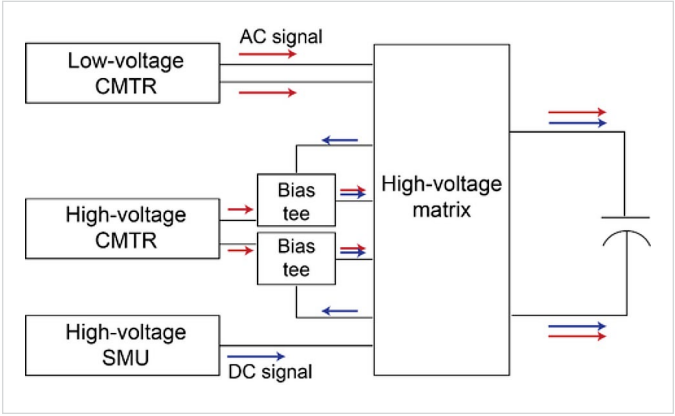


Figure 22: Using a low voltage CMTR for high voltage compensation.

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 to be 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 **Figure 22**).

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. **Figure 23** 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 less than 10 Ω. Values for OpenCp are usually less than 10 pF. Gain compensation factors (GainR and GainX) are real and imaginary (reactive) components of the load compensation. GainX is usually 0.10 or less, and GainR is close to 1.00 (from 0.95 to 1.10). Note that units of GainX and GainR are dimensionless.

```
#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
```

Figure 23: Open, short, and load compensation factors in cvCAL.ini.

Generating compensation factors for a single frequency

The S540 High Voltage Library (HVLib) *hvcv_genCompFreq* command generates compensation factors for a single frequency. This command can be used to debug the compensation algorithms, and it prompts the user 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. **Figure 24** shows example parameters for the command; definitions of the parameters follow the figure.

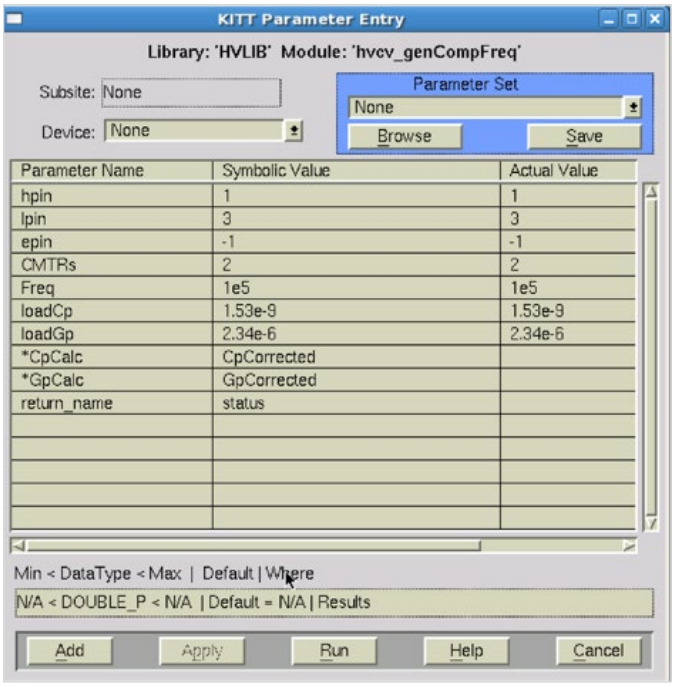


Figure 24: Compensation data collection function: *hvcv_genComp_Freq*.

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

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

- *hvcv_test*: Makes a two-terminal single DC bias measurement
- *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}), gate drain capacitance (C_{gd}), 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 previously 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, 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.

Appendix 2: 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

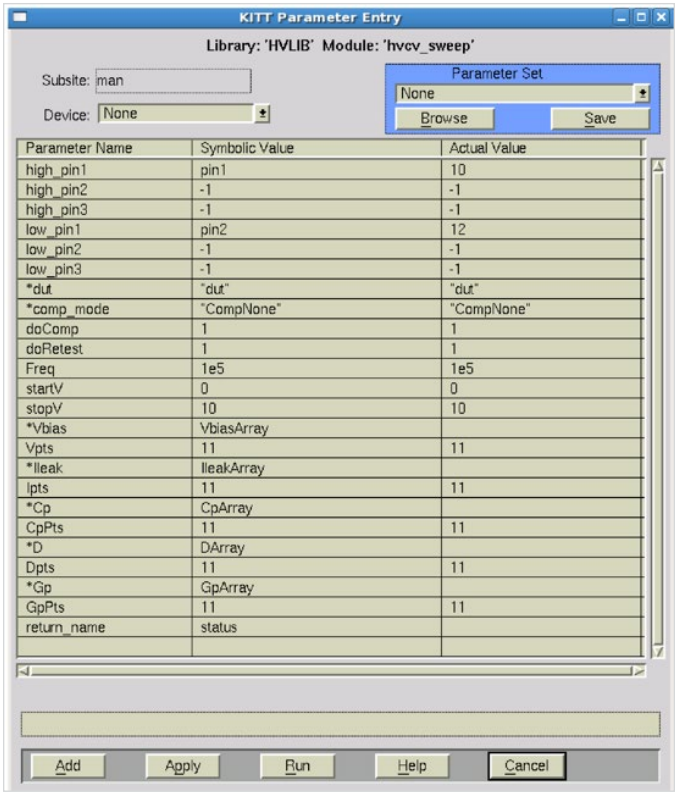
This is the simplest type of high voltage capacitance-voltage (C-V) measurement. One of the following S540 High Voltage Library (HVLib) commands is used:

- *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, 0, 1e5, 0, 10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In the preceding example, “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. Example of the call is also **Figure 25**.



Parameter Name	Symbolic Value	Actual Value
high_pin1	pin1	10
high_pin2	-1	-1
high_pin3	-1	-1
low_pin1	pin2	12
low_pin2	-1	-1
low_pin3	-1	-1
*dut	"dut"	"dut"
*comp_mode	"CompNone"	"CompNone"
doComp	1	1
doRetest	1	1
Freq	1e5	1e5
startV	0	0
stopV	10	10
*Vbias	VbiasArray	
Vpts	11	11
*Ileak	IleakArray	
Ipts	11	11
*Cp	CpArray	
CpPts	11	11
*D	DArray	
Dpts	11	11
*Gp	GpArray	
GpPts	11	11
return_name	status	

Figure 25: High voltage C-V sweep test.

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

This example shows how to run benchtop tests on a single device using the Keithley Interactive Test Tool (KITT). This process has three parts:

- Run device-level compensation using the *hvcv_genCompData* user module in KITT.
- Input the system pin numbers to use to test the device.
- Run the module.

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 expected from the device to be tested. 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.

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, 0, 1e5, 0, 10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

In the preceding example, "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 just created using the *hvcv_genCompData*.

> **Note:** The compensation factors created using the *hvcv_genCompData* module contain data for all frequencies available on the system at the time the module was run. 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

When using an automated test plan module to test a series of devices, 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, it's possible to 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, 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, 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, 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, 0, 1e5, 0, 10, Vbias, 11, ILeak, 11, Cp, 11, D, 11, Gp, 11)
```

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

> **Note:** For the load type of compensation, the test must be run 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 preceding examples, "*CompNone*" indicates that no compensation is done as the compensation data is collected. The value following "*CompNone*" (the *doComp* parameter) is set to 0 so that the routine does not do *ShortOpenLoad* compensation as the compensation data is collected.

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 testing multiple devices or wafers. 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, 0, 1e5, 0, 10, Vbias, 11, ILeak,
11, Cp, 11, D, 11, Gp, 11)
```

In this example, “dut” indicates that the system is testing the device under test. “*CompShortOpenLoad*” specifies that all three types of compensation are used. Alternatively, “*CompShortOpen*” could be used 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

Several steps 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

The 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 being tested (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 preceding examples, the dut parameter indicates the type of compensation device to use (*open*, *short*, or *load*).

> **Note:** For the load type of compensation, the test must be run 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 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 while compensation data is being collected. 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 to be used 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 in the following 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 preceding code examples is determined by the compensation mode being used (C_{iss}, C_{oss}, or C_{rss}) 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, all *hvcv_storeData* modules and the *hvcv_3term* module must be run 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 preceding *hvcv_3term* example, "dut" indicates that the test will run on the device under test. "*CompShortOpenLoad*" specifies that all three modes of compensation will be used. One 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, the *hvcv_3term* command is used.

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

> **Note:** For the load type of compensation, the test must be run 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 compensation data is being collected. The value following "*CompNone*" (the *doComp* parameter) is set to 0 so that the routine does not do *ShortOpenLoad* compensation as compensation data is being collected.

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 testing multiple devices or wafers. 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 in the following.

```
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 the device under test is being tested. "*CompShortOpenLoad*" specifies that all three types of compensation are used. Alternatively, "*CompShortOpen*" could be used 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.

Appendix 3: HV Library

This appendix provides a short description of the HVLib Library, which was developed by Keithley Instruments for HV testing on the S540 system with KTE software. Most of the functions were developed for use in three-terminal HV C-V measurements and designed to measure C-V, calculate compensation constants, and perform *Open/Short/Load* compensation.

hv_bvsweep

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

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

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.

hvcv_3term

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)

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

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 when measuring C_{rss}, 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 re-measurement of the compensation data.

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.

hvcv_comp

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

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, *CompOpen*Load, *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:

```
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.

hvcv_genCompData

int hvcv_genCompData(*int* hpin, *int* lpin)

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 being measured. If 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.

Example:

```
status= hvcv_genCompData(pin1, pin2)
```

Generates compensation factors for pin 1 and pin 2.

hvcv_genCompFreq

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

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; it's essential to 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. These values can then be added to the /opt/kiS530/cvCAL.ini file

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.

hvcv_getData

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

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.

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.

hvcv_intgcg

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

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 it's necessary 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.

hvcv_measure

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

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

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.

hvcv_storeData

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

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.

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.

hvcv_sweep

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*, *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)

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*)

Example:

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

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

hvcv_test

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)

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
- 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.

References

[1] D. K. Madjour, Silicon Carbide market update: From discrete devices to modules. PCIM Europe 2014. Nuremberg. [Online]. Available: http://apps.richardsonrfpd.com/Mktg/Tech-Hub/pdfs/YOLEPCIM_2014_SiC_Market_ARROW_KMA_Yole-final.pdf. Accessed on: September 30, 2016.

[2] Customized probe card for on-wafer testing of AlGaN/GaN power transistors. R. Venegas. COMPASS, 2015. [Online]. Available: http://www.celadonsystems.com/wp-content/uploads/2016/06/Final_COMPASS2015_imecKA090715.pdf. Accessed on: September 30, 2016.

[3] Power MOSFET Tutorial. Jonathan Dodge, P.E. Advanced Power Technology. Jonathan Dodge, P.E. (2006). Power MOSFET Tutorial. Bend, OR 97702: Advanced Power Technology. Retrieved from http://www.microsemi.com/document-portal/doc_view/14692-mosfet-tutorial.

[4] Power MOSFET Basics. (2008). Alpha and Omega Semiconductor. Retrieved from http://www.aosmd.com/res/application_notes/mosfets/Power_MOSFET_Basics.pdf

[5] Understanding power MOSFET data sheet parameters. (2014). NXP. Retrieved from http://assets.nexperia.com/documents/application-note/AN11158.pdf

Contact Information:

Australia* 1 800 709 465

Austria 00800 2255 4835

Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777

Belgium* 00800 2255 4835

Brazil +55 (11) 3759 7627

Canada 1 800 833 9200

Central East Europe / Baltics +41 52 675 3777

Central Europe / Greece +41 52 675 3777

Denmark +45 80 88 1401

Finland +41 52 675 3777

France* 00800 2255 4835

Germany* 00800 2255 4835

Hong Kong 400 820 5835

India 000 800 650 1835

Indonesia 007 803 601 5249

Italy 00800 2255 4835

Japan 81 (3) 6714 3010

Luxembourg +41 52 675 3777

Malaysia 1 800 22 55835

Mexico, Central/South America and Caribbean 52 (55) 56 04 50 90

Middle East, Asia, and North Africa +41 52 675 3777

The Netherlands* 00800 2255 4835

New Zealand 0800 800 238

Norway 800 16098

People's Republic of China 400 820 5835

Philippines 1 800 1601 0077

Poland +41 52 675 3777

Portugal 80 08 12370

Republic of Korea +82 2 6917 5000

Russia / CIS +7 (495) 6647564

Singapore 800 6011 473

South Africa +41 52 675 3777

Spain* 00800 2255 4835

Sweden* 00800 2255 4835

Switzerland* 00800 2255 4835

Taiwan 886 (2) 2656 6688

Thailand 1 800 011 931

United Kingdom / Ireland* 00800 2255 4835

USA 1 800 833 9200

Vietnam 12060128

*** European toll-free number. If not accessible, call:** +41 52 675 3777

KEITHLEY
A Tektronix Company

Find more valuable resources at **TEK.COM**

010918 SBG 1KW-60936-2