Techniques for Proper and Efficient Characterization, Validation, and Reliability Testing of Power Semiconductor Devices

APPLICATION GUIDE

KEITHLEY
A Tektronix Company

Contents

Testing Power Semiconductor Devices with Keithley High Power System SourceMeter® SMU Instruments

Introduction

The proliferation of electronic control and electronic power conversion into a variety of industries (e.g., energy generation, industrial motor drives and control, transportation, and IT) has spurred growth in power semiconductor device design and test. To demonstrate technology improvements, new device capabilities must be compared with those of existing devices. The use of semiconductor materials other than silicon demands the use of new processes. And, to be sustainable, these new processes must be tuned to deliver consistent results and high production yield. As new device designs are developed, reliability measurements are performed on many devices over long periods. Therefore, test engineers must identify test equipment that is not only accurate but scalable and cost-effective.

Power module design engineers—the consumers of the discrete power semiconductor components—work at the other end of the semiconductor device testing spectrum. They integrate the discrete components into designs for DC-DC converters, inverters, LED controllers, battery management chips, and many other devices. Driven by demands for higher energy efficiency, these engineers need to qualify the devices they receive from their vendors to ensure that they can withstand use in the application, predict how the efficiency of the power modules may be affected by the device, and finally validate the performance of the end product.

Keithley's SourceMeter SMU instruments give both device test engineers and power module design engineers the tools they need to make the measurements they require. Whether they're familiar with curve tracers, semiconductor parameter analyzers, or oscilloscopes, they can obtain accurate results simply and quickly. This application note highlights some of the most commonly performed tests, the challenges associated with them, and how Keithley SMU instruments can simplify the testing process, especially when integrated into a Keithley Parametric Curve Tracer (PCT) configuration.

Background on Power Device Characterization

The switching power supply is one common electrical circuit element used in power management products. In its simplest form (***Figure 1***), its main components include a semiconductor such as a power MOSFET, a diode, and some passive components, including an inductor and a capacitor. Many also include a transformer for electrical isolation between the input and output. The semiconductor switch and diode alternatively switch on and off at a controlled duty cycle to produce the desired output voltage.

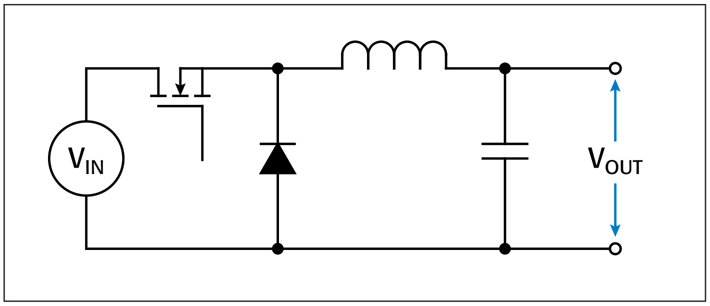


Figure 1. A simple schematic of a type of switching power supply.

When evaluating energy efficiency, it's important to understand the *switching loss* (energy loss that occurs during the short periods when the device is changing states) and *conduction loss* (energy losses that occur when the device is either on or off). Keithley SMU instrument-based solutions can help test engineers evaluate the device parameters that affect conduction loss.

Semiconductor devices are often used to ensure circuit protection. For example, some thyristor devices are used for overvoltage protection. To achieve that objective, such devices must trigger at the appropriate intended voltage and current, must withstand the intended voltage, and must behave in circuit with minimal current draw. High power instrumentation is required to qualify these devices properly.

This note focusses on the characterization of static power device parameters.[1] These parameters can be divided into two broad categories: those that determine the performance of the device in its ON state and those that determine the performance in its OFF state. ***Table 1*** lists common ON-state and OFF-state parameters for several power semiconductor devices that Keithley SMU instruments support. Many tests involve the use of multiple SMU instruments. Keithley's ACS Basic Edition software simplifies the test configuration by managing the configuration and data collection of all SMU instruments in the test system. Unlike general-purpose start-up software, ACS Basic Edition is designed specifically for semiconductor device characterization and includes a library of tests; users can focus on the test and device parameters rather than the SMU instrument configuration. The test results included in this note were acquired using ACS Basic Edition software, which is included in our PCT configuration.

[1] Tektronix solutions are available for transient characterization of power devices. For more information, visit www.tek.com.

Table 1. Common power semiconductor devices and tests.

Test Category	Devices and Tests					Keithley SMU I-V Capability
	Diode	MOSFET	BJT	IGBT	Thyristor-Class Devices (e.g., SCR, Triac)	
ON-state	V_F–I_F	V_{DS}–I_D V_{TH} V_{GS}–ID $R_{DS(on)}$	V_{CE}–I_C Gummel plot	V_{CE}–I_C V_{GE}–I_C	V_T I_H I_L	Voltage: –40V to +40V Current: Up to 100A (pulse)
OFF-state	I_R	I_{GSS} I_{DSS} BV_{DSS} BV_{DG}	I_{CEO} I_{CES} BV_{CES} BV_{CEO} BV_{CBO}	I_{CEO} I_{CES} BV_{CES} BV_{CEO}	V_{bo} V_{DRM} V_{RRM}	Voltage: –3kV to +3kV Current: Down to 1fA

ON-State Characterization

ON-state characterization is typically performed using a high current instrument capable of sourcing and measuring low-level voltages. If the device has three terminals, then a second SMU instrument is used at the device control terminal to place the device in the ON state. For a typical configuration for characterizing the ON-state parameters of a power MOSFET, see ***Figure 2***.

Keithley SMU instruments cover a wide range of currents for power semiconductor devices. Series 2600A and 2600B System SourceMeter SMU instruments include at least 1.5A DC and 10A pulse capability for DC characterization. For very high current devices, two Model 2651A High Power SourceMeter SMU instruments can be used in parallel to generate current pulses up to 100A.

Let's examine the configuration details and measurement challenges of a few ON-state parameters.

Output characteristics

One of the most readily recognizable set of test results for a semiconductor device is a plot of its output characteristics. Output characteristics are normally shown in graphical form on the device's data sheet and depict the relationship between the output voltage and current. For a gated power semiconductor switch, such as a MOSFET, IGBT, or BJT, output characteristics are commonly referred to as the "family of curves." ***Figure 3*** shows the results for a power IGBT as generated by ACS Basic Edition software.

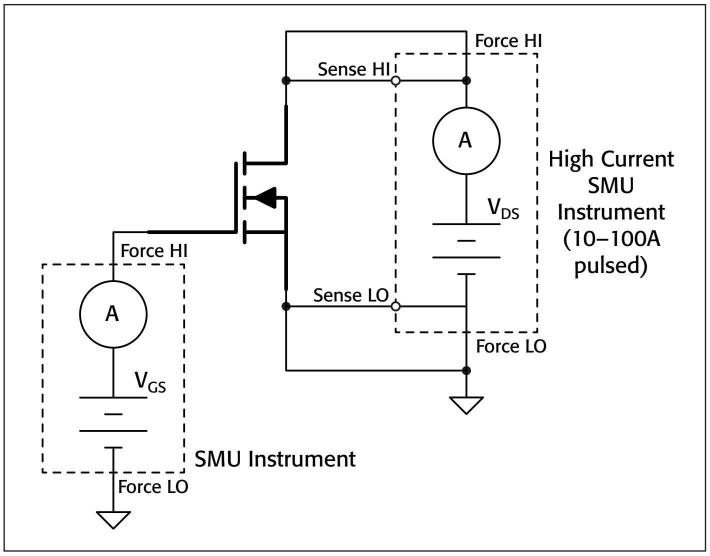


Figure 2. Typical SMU configuration for ON-state characterization of power devices.

Pulsed testing is common in power semiconductor test because DC testing can cause the device to self-heat, which can alter the measured characteristics. Implementing a pulsed test with multiple SMU instruments requires precise control of the timing of source adjustments and measurements and is often coordinated by means of a computer program. For consistent results, it is important to validate the system. This can be done by sourcing a single pulse through the system and measuring the response at the device. Capturing the complete pulse waveform as a function of time allows selecting appropriate source and measure delays so that the device turns on properly and measurements are made after the system is settled. The high speed A/D converters in Series 2650A High Power System SourceMeter SMU instruments are useful for monitoring the voltages and currents at the device as they relate to time. A diagnostic feature in ACS Basic Edition 2.0 software allows capturing the pulse shape of a single point in the family of curves easily. ***Figure 4*** depicts the results of a pulse transient characterization of collector voltage and current vs. time of an IGBT. For this particular example, delaying the measurement 100μs after the start of the pulse ensures the system has settled prior to measurement, allowing for more consistent results between tests.

Power semiconductor devices are often high gain devices; oscillation is common when characterizing such devices and will result in erratic measurements. The high speed A/D converters in the Series 2650A SMU instruments and the pulse transient feature of ACS Basic Edition 2.0 software are useful for verifying the presence of oscillation. Resolving this

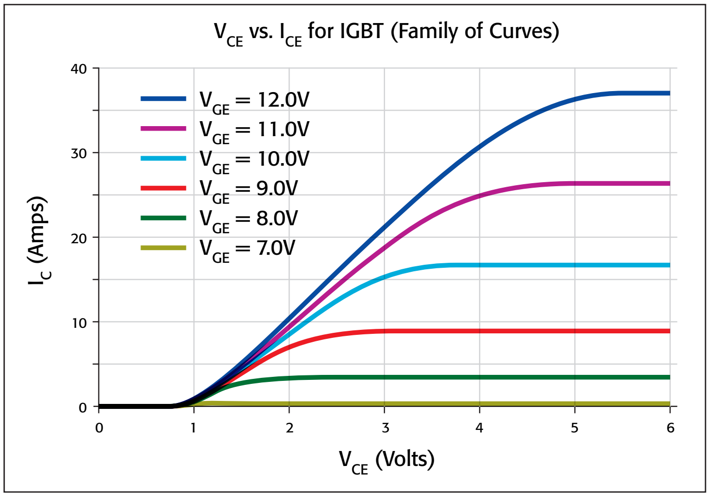


Figure 3. Measured output characteristics for commercially available IGBT.

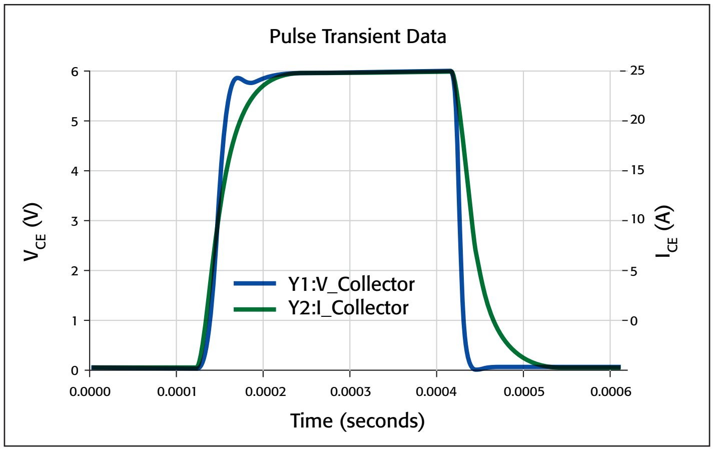


Figure 4. Pulse transient data of collector voltage and current vs. time for an IGBT.

oscillation involves adding a resistor in series with the device control or input terminal, for example, the gate of a MOSFET or IGBT. The Keithley Model 8010 High Power Test Fixture easily accommodates the addition of a discrete resistor.

ON-state voltage

The semiconductor device's ON-state voltage directly impacts the conduction loss. Examples of ON-state voltages include the forward voltage of a power diode (V_F), the ON-state saturation voltage of a BJT or an IGBT (V_{CEsat}), and the ON-state voltage of a thyristor (V_T). Power consumed by or lost in the device is equal to the product of the ON-state voltage and the load current. This power is not delivered to the device. Typically, device manufacturers want to characterize how the ON-state voltage and, by extension, the conduction loss, varies with temperature and load current. Keithley SMU instruments are commonly used in these characterizations.

To measure the ON-state voltage, the high current SMU instrument is configured to output current to the device and measure voltage. For BJTs and IGBTs, a second, lower-power SMU instrument is used at the base or gate terminal to place the device in the ON state. Because power semiconductors are typically high current devices, ON-state voltage is generally measured using a pulsed stimulus to avoid any change in device parameters as a result of device self-heating due to a DC test current.

Two key elements help ensure a successful ON-state voltage test: (1) accurate voltage measurement and (2) proper cabling and connections. Accurate voltage measurements are crucial because ON-state voltage varies with temperature. For instance, a few millivolts of difference in the forward voltage of a power diode can indicate a change of several degrees in the temperature at the device. High speed A/D converters in the Keithley Model 2651A High Power System SourceMeter SMU Instrument let it make very accurate voltage measurements at 1μs intervals with pulse widths as short as 100μs.

Proper cabling and connections are equally key to minimizing voltage errors. For power diodes, BJTs, and IGBTs, typical test currents can range from 100mA to tens of amps while ON-state voltages of 1–3V are very common. Thyristors are ideal for use in ultra-high-power applications because they have very low ON-state voltages (<2V) while conducting currents that could be greater than 100A. During testing, such high currents can generate voltage drops across test leads and test lead connections between the instrument and the DUT. These additional voltages create errors in the voltage measurement. Four-wire or Kelvin connections eliminate most of these voltage errors from the measurement by using separate test leads for the voltmeter. Minimal current flows in these leads, creating virtually no voltage drops between the instrument and DUT, so the instrument measures the voltage seen at the DUT.

The use of low inductance cables helps ensure good pulse fidelity (i.e., short rise and fall times) when testing high current devices, which provides more time for measurement within a given pulse width. The Model 2651A High Power System SourceMeter SMU Instrument comes with a specialized low resistance, low inductance cable assembly appropriate for generating 100μs pulses at 50A.

Transfer characteristics

A device's transfer characteristics allow evaluating its transconductance and therefore its current carrying capability. Transfer characteristics have an indirect relationship to determining switching time and estimating switching losses. The transfer characteristics are often monitored vs. temperature in order to gauge temperature's effect on the device's maximum current handling capability. Two SMU instruments are required for measuring the transfer characteristics: one sweeps the input voltage on the control terminal of the device and the second biases the output terminal and measures the output current. Typical transfer characteristic measurements include the gate voltage vs. drain current plot for a MOSFET (V_{DS}–I_D), the gate voltage vs. collector current plot for an IGBT (V_{GE}–I_C), and the Gummel plot for a BJT (V_{BE} vs. I_C, I_B).

In some cases, a wide range of output current is measured. This is especially true for the Gummel plot of a BJT, where several orders of magnitude of current are traversed. In these cases, the Model 2651A is very useful because it can measure currents from the nanoamp range up to 50A. ***Figure 5*** depicts a Gummel plot generated using a Model 2651A on the collector and a Model 2636B on the base.

ON-resistance

The figure of merit for a power MOSFET is the product of ON-resistance ($R_{DS(on)}$) and gate charge (Q_G). The ON-resistance is the key determinant of the conduction loss of the power MOSFET. The conduction loss is equal to $I_D * R_{DS(on)}$. Newer devices have ON-resistances in the range of a few milliohms to tens of milliohms at high current. This requires very sensitive voltage measurement capability at the drain terminal. Measuring ON-resistance requires the use of two SMU instruments: one SMU instrument drives the gate into the ON state and a second SMU instrument pulses a defined current at the drain and

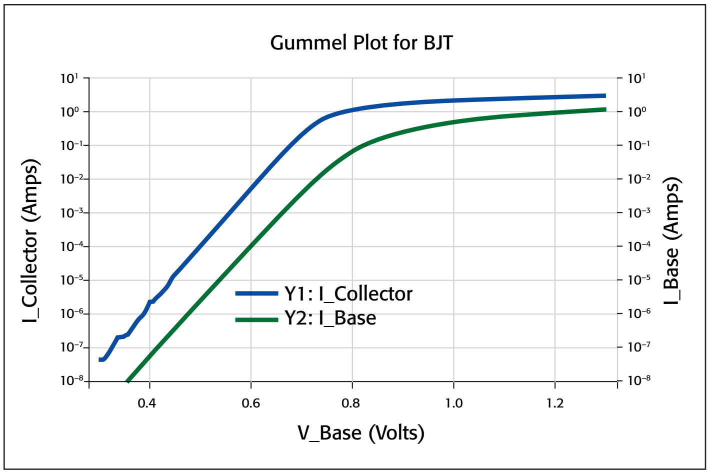


Figure 5. Gummel plot for typical power BJT generated using Keithley Models 2651A and 2636B SMU instruments.

measures the resulting voltage. The ON-resistance is calculated using Ohm's Law and the programmed drain current and measured drain voltage. Such a calculation can be automatically configured in software.

ON-resistance is often characterized as a function of drain current or gate voltage. Using software, both SMU instruments can be triggered and swept so that this measurement is performed within a single test. ***Figure 6*** shows the calculated $R_{DS(on)}$ vs. drain current results. This was all completed during a single run of the $R_{DS(on)}$ test. For very high current devices, two Model 2651As can be used in parallel to generate current pulses up to 100A. ACS Basic Edition software manages the configuration of both SMU instruments and the data collection.

ON-resistance increases with breakdown voltage, so any process adjustments made to increase the breakdown voltage will involve later testing of ON-resistance in order to assess the overall impact of process changes. Obtaining more efficient devices at higher voltages is one of the drivers for further research on SiC and GaN devices, which offer ON-resistances smaller than silicon devices' at high withstand voltages.

OFF-State Characterization

To gain an adequate understanding of overall product efficiency, the impact of the device on the overall circuit when the device is turned off must be investigated. For high power devices, OFF-state characterization often involves the use of a high voltage instrument capable of sourcing hundreds or thousands of volts and measuring small currents. OFF-state characterization is often performed between two device terminals (regardless of the total number of device terminals), so a single SMU instrument is often sufficient to perform the measurement. However, an additional SMU instrument can be used to force the device into its OFF state or to add certain stress to certain terminals.

Keithley SourceMeter SMU instruments cover a wide range of voltages and currents for characterizing the OFF state of power semiconductor devices. The Model 2635B and 2636B SMU instruments offer 200V characterization with current measurement capability down to the femtoamp level. The Model 2657A SMU instrument extends high voltage characterization to 3kV while providing very low current measurement resolution and accuracy.

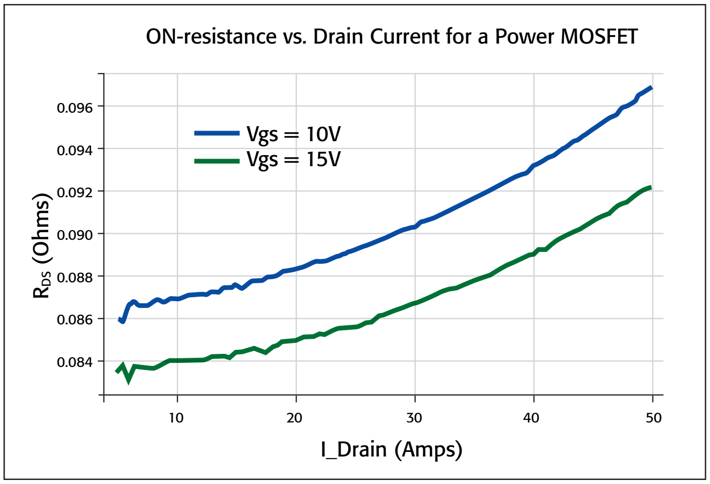


Figure 6. Results for ON-resistance of a power MOSFET as measured as a function of drain current for two gate voltages.

Two primary DC tests are performed when the device is off: breakdown voltages and leakage currents. Let's consider these individually.

Breakdown voltages

A device's OFF-state breakdown voltage determines the maximum voltage that can be applied to it. The primary withstand voltage of interest to power management product designers is the breakdown voltage between drain and source of a MOSFET or between the collector and emitter of an IGBT or BJT. For a MOSFET, the gate can be either shorted or forced into a "hard" OFF state, such as by applying a negative voltage to an n-type device or a positive voltage to a p-type device. This is a very simple test that can be performed using one or two SMU instruments. The lower power SMU instrument is connected to the gate and forces the transistor off. It can force 0V for a gate shorted test or force a user-specified bias voltage. A high voltage SMU instrument, such as the Model 2657A, forces current into the drain and measures the resulting drain voltage.

Most MOSFETs typically have breakdown voltages higher than the value specified on the data sheet. Therefore, it is useful to define the voltage limit of the drain SMU instrument to a value higher than the specified breakdown voltage. Additionally, as the drain terminal is driven closer and closer toward avalanche, the behavior of the currents and voltages of the device under test (DUT) may begin to change. These cases can take advantage of the high speed A/D converters of the Series 2650A SMU instruments. Without the need for any extra equipment, it's possible to catch a quick look at both the transient behavior of current and voltage at the DUT. ***Figure 7*** is an example of transient characterization of a test of the breakdown voltage between drain and source of a commercially available 600V silicon power MOSFET. The Model 2657A is used to pulse a

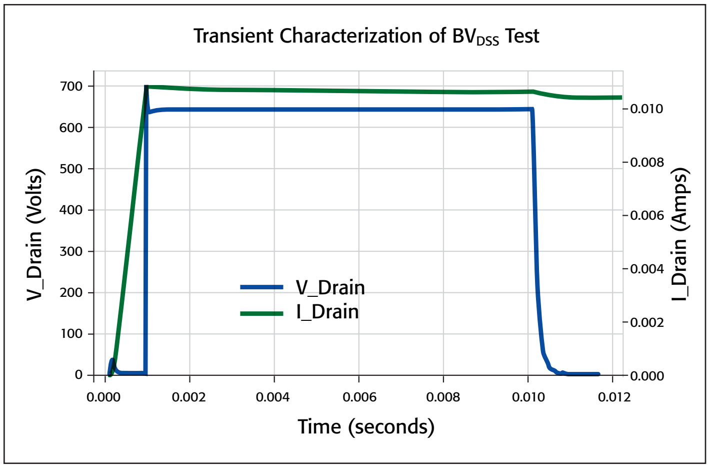


Figure 7. Transient capture of voltages and currents of BV$_{DSS}$ test using the Model 2657A's fast A/D converter. Sampling interval is 10µs.

current into the MOSFET and then measure the voltage and current at 10μs intervals. The plot shows that the device actually breaks down at approximately 680V.

Another way of characterizing breakdown voltages involves forcing a voltage across the terminals of interest (e.g., drain and source of a MOSFET) and measuring the resulting current. The breakdown voltage is defined as the voltage at which the current exceeds a specified threshold, for example, 1mA. Limit the maximum current in order to prevent destruction of the device during testing. Unlike traditional curve tracers and power supplies, Keithley SourceMeter SMU instruments include built-in programmable features to limit the maximum voltage and current to the device precisely and quickly. As with any protection device, the limit control of the SMU instrument has a finite response time. Some devices may have extremely fast and hard breakdown behavior in which the device impedance changes by several orders of magnitude in a very short period. When the device breaks down faster than the SMU instrument can respond, use series resistors to limit the total maximum current through the device.

Safety must be one of the first considerations for high voltage characterization of power semiconductor devices. Keep in mind the voltage ratings for all terminals, connectors, and cables. For example, Keithley SourceMeter SMU instruments are electrically floating, which means that measurement common is not tied to protective earth (safety ground). Unless the user ties measurement common to safety ground, then he must take high voltage precautions at *all* terminals if the SMU instrument can generate more than 42V.

When configuring a test system, it's important to protect the operator from electric shock. One primary way to do this is to use a safe test enclosure that surrounds the DUT and any exposed connections. The Keithley Model 8010 High Power Test Fixture permits safe testing of packaged high power semiconductor devices at up to 3kV. Pairing a safe enclosure with a safety interlock also minimizes risk of electric shock when the user changes test connections. Keithley SMU instruments are equipped with a safety interlock; when properly installed, this interlock ensures that the hazardous voltages are disconnected whenever a user opens the test fixture or accesses the wafer in a probe station.

In addition to protecting the operator, it is also important to consider the interactions between all the instruments connected to the device terminals. If a lower voltage SMU instrument is connected to the device during a breakdown characterization, a device fault can result in high voltage being present across this lower voltage SMU instrument. Keithley's Model 8010 High Power Test Fixture includes built-in protection to protect the lower voltage SMU instrument in such applications.

Leakage currents

Leakage current is the level of current that flows through two terminals of a device even when the device is off. It factors into the standby current of the end product. In most cases, temperature and the voltage across the terminals of interest will affect leakage current. Minimizing leakage current minimizes power loss when the device is off. This power is consumed by the device and is not output to the load and therefore contributes to power inefficiency. When using a transistor or diode to switch or rectify, it's important to make a clear distinction between ON and OFF states; therefore, a lower leakage current means having a better switch.

While testing a device's OFF state, it is generally desirable to test the gate leakage current and drain or collector leakage current. For power devices, these values are typically within the nanoamp and microamp ranges, so they can be measured using the sensitive current measurement capability of Keithley SMU instruments. This capability can be greatly beneficial when testing devices made of wide bandgap materials such as silicon carbide, gallium nitride, and aluminum nitride, which typically have higher breakdown voltages and lower leakage currents than do silicon devices. ***Figure 8*** is a plot of OFF-state drain voltage vs. drain current results for a commercially available SiC power MOSFET.

Triaxial cables are essential to achieving accurate low current measurements in part because they permit carrying the guard terminal. Guarding eliminates the effect of system leakage currents by routing them away from the measurement terminal. Additionally, guarding reduces settling time in high voltage applications by virtually eliminating the need to charge the cable capacitance. Using a guarded test fixture or prober allows the benefits of guarding to be extended to the DUT.[2] Keithley offers triaxial cables and connections for all its SMU instruments that are tailored for low current measurements, such as the Model 2636B and Model 2657A SMU instruments. The specialized high voltage triaxial cables for the Model 2657A allow measurements at 3kV with 1fA resolution. The Model 8010 test fixture includes guarded connections to the device test board to permit current measurements down to tens of picoamps.

[2] For more details on guarding, review Keithley Application Note #3163, *Creating Multi-SMU Systems with High Power System SourceMeter SMU Instruments.*

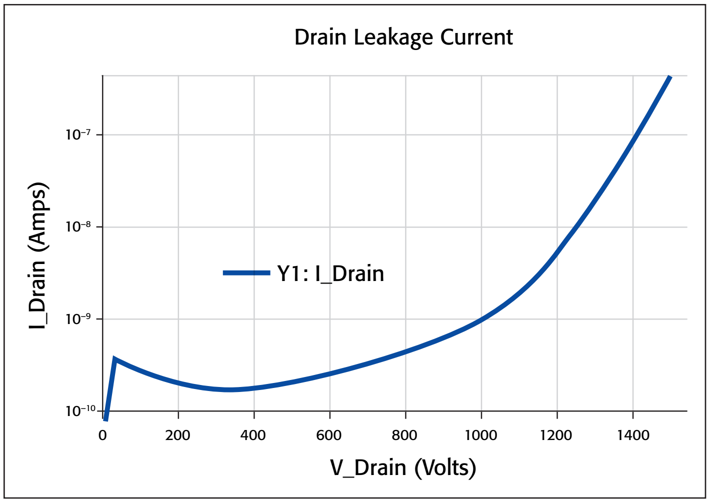


Figure 8. A look at the drain leakage current as the drain voltage is swept while the transistor is in the OFF state.

Electrostatic shielding is another important consideration for low current measurements. An electrostatic shield is a metal enclosure that surrounds the circuit and any exposed connections. The shield is connected to measurement common to shunt electrostatic charges away from the measurement node. When testing devices within the Model 8010 High Power Test Fixture, the test fixture chassis serves as an electrostatic shield.

Finally, system validation is important for low current characterization to ensure that the measurement is settled. Settling time increases as the expected current decreases. Keithley products have auto delay settings that set reasonable delays to achieve good measurements while taking the instrument's settling characteristics into account. However, to account for any unguarded system capacitance, perform a measurement validation by stepping voltage and measuring the resulting current through the system. Use the results from the validation to set additional source and measure delays in order to achieve consistent measurements.

Conclusion

Keithley SourceMeter SMU Instrument solutions can be used with ACS Basic Edition software to provide a comprehensive solution for testing high power semiconductor discrete devices. ACS Basic Edition includes a library of tests for a variety of power devices including FETs, BJTs, IGBTs, diodes, resistors, and thyristors. Additionally, Keithley has the appropriate cabling accessories and test fixtures that allow safe, accurate, and reliable testing. Although these instruments and accessories can be purchased separately, they are also available as part of Keithley's Parametric Curve Tracer Configurations. Refer to Keithley's website (www.keithley.com) or contact your local Keithley representative for further information or system configuration assistance.

Breakdown and Leakage Current Measurements on High Voltage Semiconductor Devices Using Keithley Series 2290 High Voltage Power Supplies and Series 2600B System SourceMeter® Source Measure Unit (SMU) Instruments

Increased attention to energy efficiency has resulted in electronics with higher power density. In grid-connected and industrial applications, such as AC motor control, uninterruptible power supplies (UPS,) and traction control (large hybrid and electric transport vehicles,) the need to keep manageable cable sizes pushes power conversion to higher voltages. For such voltages, the semiconductor device of choice has historically been the thyristor. Technological advances in device fabrication and material processing is enabling the development of IGBTs and MOSFETs with voltage ratings of thousands of volts. In applications where possible, using IGBTs or even MOSFETs in place of thyristors permits power conversion at high switching frequencies. The migration to higher frequency reduces the size of passive components used in the design and, thereby, improves energy efficiency.

Keithley has long had a strong presence in high power semiconductor device test with its high voltage source-measure products, including the Models 237, 2410, and 2657A SMU instruments. Most recently, Keithley released the Model 2290-5 5kV and Model 2290-10 10kV High Voltage Power Supplies. This note considers the application of these power supplies to high voltage semiconductor device testing.

High Voltage Device Tests

Basic characterization of high voltage semiconductor devices typically involves a study of the breakdown voltage and leakage current. These two parameters help the device designer to quickly determine whether the device was correctly manufactured and whether it can be effectively used in the target application.

Breakdown Voltage Measurements

Measuring breakdown voltage is done by applying an increasing reverse voltage to the device until a certain test current is reached that indicates that the device is in breakdown. ***Figure 1*** depicts a breakdown measurement on a high voltage diode using a Series 2290 High Voltage Power Supply. Note that the Series 2290 Power Supplies are unipolar supplies and must be connected to the diode's cathode in order to apply a reverse voltage.

In qualifying breakdown voltage, measurements are typically made well beyond the expected rating of the device to ensure that the device is robust and reliable. The models 2290-5 and 2290-10 Power Supplies have a voltage range wide enough to test many of the industry's future devices.

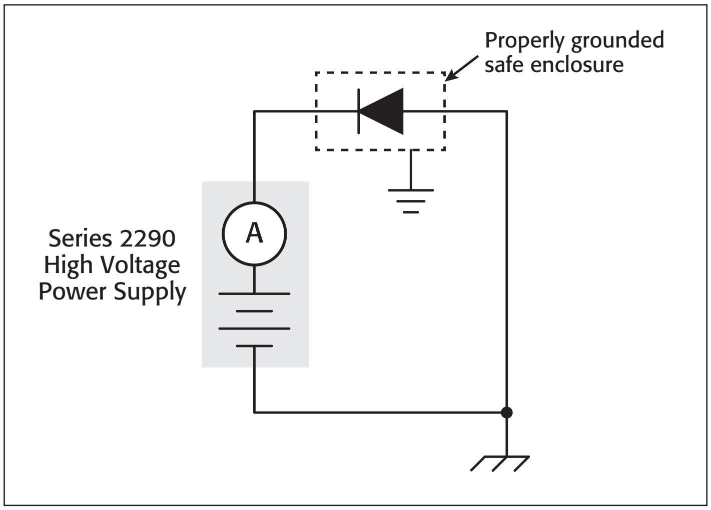

Properly grounded safe enclosure
Series 2290 High Voltage Power Supply
A

Figure 1. Typical breakdown voltage measurement of a high voltage diode using the Series 2290 High Voltage Power Supply.

Safety Considerations

When testing at high voltage, safety is of utmost concern. The Series 2290 Power Supplies generate voltage up to 10kV, so precautions must be taken to ensure that the operator is not exposed to unsafe voltage:

- Enclose the device under test (DUT) and any exposed connections in a properly grounded fixture.
- Use the safety interlock. The Series 2290 Power Supplies are fully interlocked so that the high voltage output is turned off if the interlock is not engaged (interlock switch closed.) The interlock circuit of the power supply should be connected to a normally-open switch that closes only when the user access point in the system is closed to ensure that operators cannot come in contact with a high voltage connection to the DUT. For example, opening the lid of the test fixture should open the switch/relay that disengages the interlock of the Series 2290 Power Supply.
- Use cables and connectors rated to the maximum voltage in the system. Series 2290 Power Supplies provide a number of appropriately-rated accessories that the test system designer can use to interface to the device under test (DUT).

Leakage Current Measurements

In a typical power conversion application, the semiconductor device is used as a switch. Leakage current measurements indicate how closely the semiconductor performs to an ideal switch. Also, when measuring the reliability of the device, leakage current measurements are used to indicate device degradation and to make predictions of device lifetime.

Semiconductor researchers are finding materials to make higher quality switches and produce devices with very small leakage currents. Such currents may fall below the measurement capability of the Series 2290 Power Supplies. In such cases, couple the accurate sourcing ability of the Series 2290 Power Supply with the precision low current measurement ability of a Keithley SMU instrument. Using Keithley SMU instruments improves the low current measurement resolution and accuracy and also improves the accuracy of the current limit. As an example, Keithley Models 2635B and 2636B SourceMeter® SMU instruments have four current ranges at 1μA and below. The current limit of the Keithley SMU instrument can be configured as small as 10% of a range.[1]

To prevent unwanted measurement error when measuring currents less than 1μA, use triaxial cables and electrostatic shielding. Triaxial cables are essential in part because they permit carrying the guard terminal from the current measurement instrument. Guarding eliminates the effect of system leakage currents by routing them away from the measurement terminal. Use an electrostatic shield to shunt electrostatic charges away from the measurement terminal. An electrostatic shield is a metal enclosure that surrounds the circuit and any exposed connections. The safe test enclosure may serve as an electrostatic shield. For more tips on optimizing low current measurements, refer to Keithley's *Low Level Measurements Handbook, 7th Edition*.

Safety Considerations

Review system safety whenever a new element, in this case the SMU instrument, is added to the test circuit. In addition to the safety issues considered under the topic of breakdown voltage testing, the Series 2600B SMU Instrument is also capable of generating voltages up to 200V. Like the Series 2290 power supplies, Keithley Series 2600B SMU instruments have a safety interlock to ensure operator safety during changes in the test setup. For optimum system safety, the interlock of the Series 2600B SMU Instrument should be wired in parallel with the Series 2290 Power Supplies. An example of this is shown in ***Figure 2***.

As a part of the system safety review, consider all potential consequences of device failure. In a setup where both a Series 2290 Power Supply and a Series 2600B SMU Instrument are employed, a device breakdown could result in high voltage appearing at the input terminals of the SMU instrument. Because the SMU instrument is not designed to handle these higher voltages, it must be protected against possible damage by the high voltage power supply. The Model 2290-PM-200 Protection Module can be used for this purpose. The same module can be used regardless of whether a Model 2290-5 5kV or Model 2290-10 10kV High Voltage Power Supply is used in the test circuit (see ***Figure 3***). ***Figure 4*** illustrates the placement of the Model 2290-PM-200 in the test circuit.

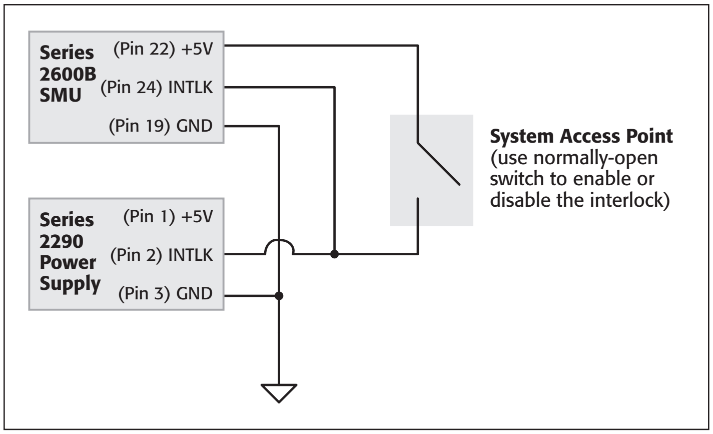


Figure 2. Correct wiring of interlocks from a Series 2600B SMU instrument and a Series 2290 Power Supply to a single system access point, e.g. test fixture lid.

Figure 3. The Model 2290-PM-200 Protection Module permits safe connection of a single 200V SMU instrument into the test circuit. It is designed to be used in a test circuit with either the Model 2290-5 5kV or the Model 2290-10 10kV Power Supply.

Using the test setup shown in ***Figure 4***, the actual test results when measuring leakage current of a high voltage diode are displayed in ***Figure 5***. The diode has a maximum specified reverse current of 10μA when 3300V is applied at room temperature. The results show that the diode meets its specification. The reverse current grows at a faster rate as the reverse voltage increases, indicating that the diode is approaching breakdown.

Figure 6 depicts the actual test results when measuring the collector-emitter cutoff current of a 4000V IGBT. In this test, the gate and emitter terminals are shorted to ensure that the device remains off (***Figure 7a***). An SMU instrument can also be used to actively program the gate voltage. Using an SMU instrument is useful if the leakage current measurements are desired with the device in hard cutoff (with a bias less than 0V at the gate terminal). ***Figure 7b*** depicts the setup using two SMU instruments and a Series 2290 Power Supply.

This particular IGBT has a maximum specified cutoff current of 100μA at 4000V. The performance of this IGBT is much better

[1] The current limit of an SMU instrument is an active current limit and has a finite response time. To limit the maximum possible current in a circuit, use a series resistor.

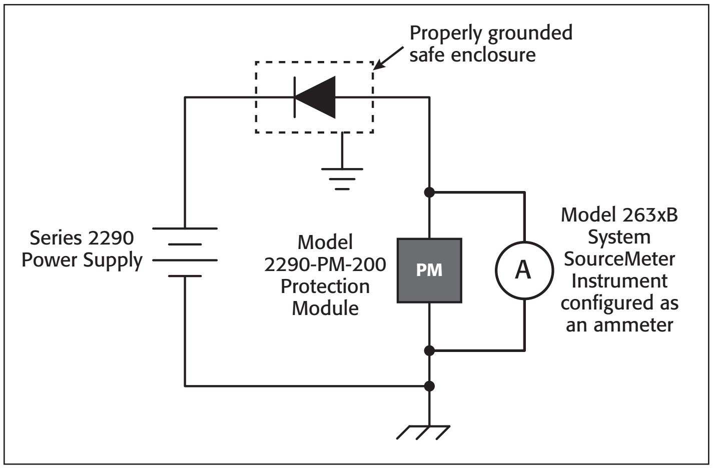


Figure 4. Characterizing the leakage current of a high voltage diode using a Series 2290 Power Supply with a Model 263xB SMU Instrument. Using the SMU instrument enhances the resolution and accuracy of both the current measurement and current limit.

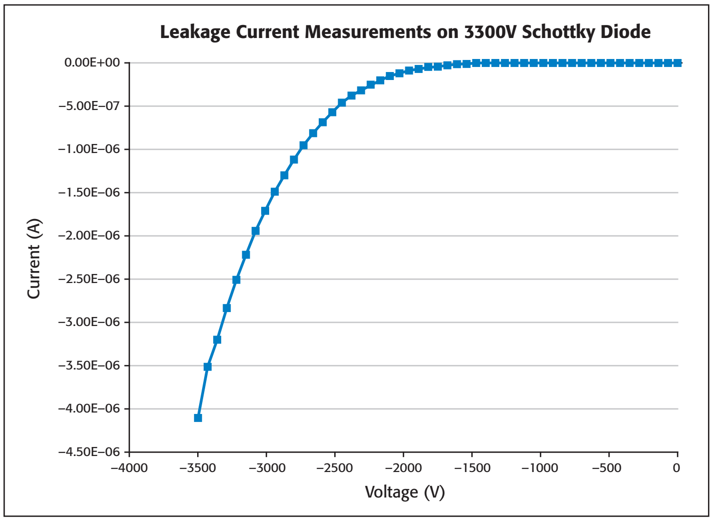


Figure 5. Measurements of 3300V Silicon Carbide Schottky diode. Voltage is applied with the Model 2290-5 5kV Power Supply and current is measured with the Model 2636B System SourceMeter SMU Instrument.

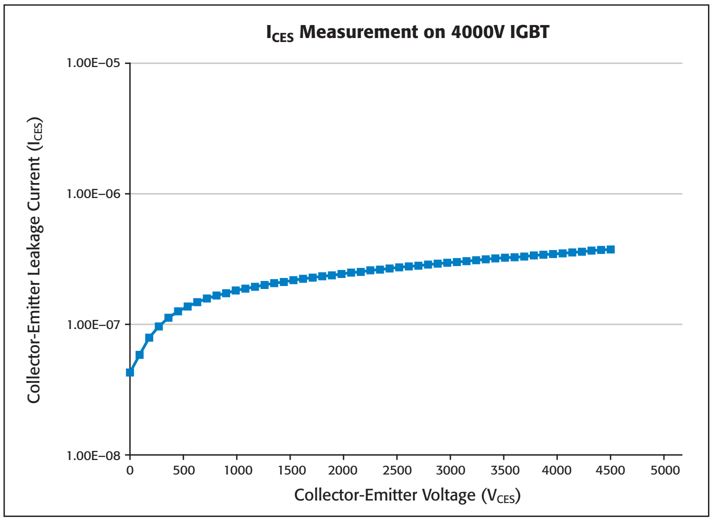


Figure 6. Ices measurements of 4000V IGBT. V_{CES} is applied with the Model 2290-5 Power Supply, and I_{CES} is measured with the Model 263xB SourceMeter SMU instrument. Gate and source terminals are shorted.

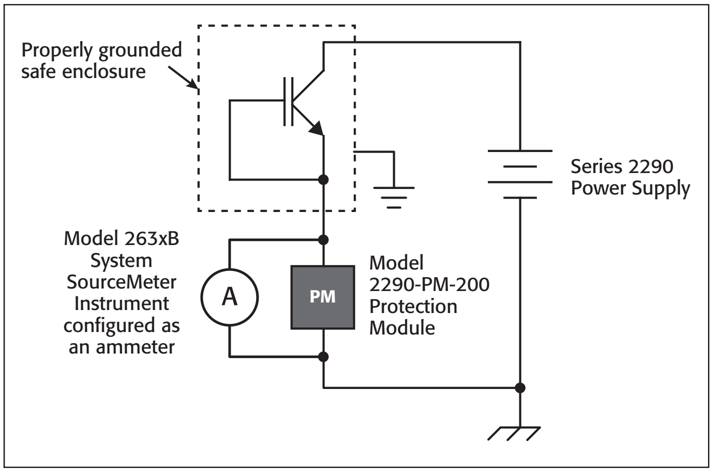


Figure 7a. Test setup using a Series 2290 Power Supply and the Model 263xB SourceMeter SMU Instrument to measure the cutoff current (I_{CES}) of an IGBT. The short between the gate and emitter terminals keeps the device in the off-state.

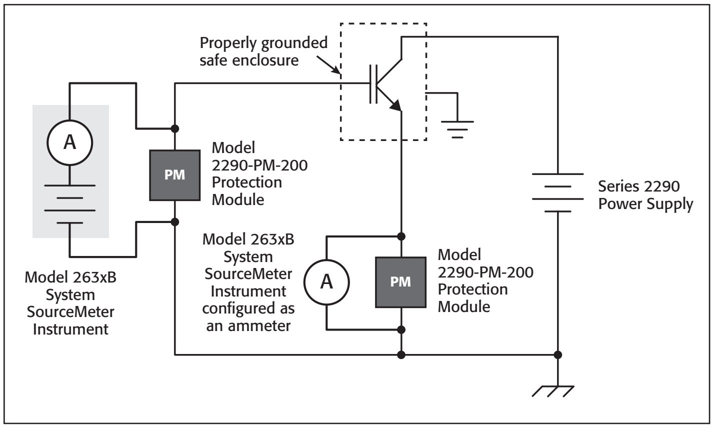


Figure 7b. Test setup using a Series 2290 Power Supply and two Model 263xB SourceMeter SMU instruments to measure the cutoff current (i_{CES}) of an IGBT. The SMU instrument connected to the gate terminal can be used to place a certain bias on the gate, e.g., to drive the device into hard cutoff.

than the specification. In fact, even at 4500V, the cutoff current is not increasing rapidly, thereby indicating that the device is not yet in breakdown.

The command sequence to generate the results shown in both ***Figures 5*** and ***6*** using either the Model 2290-5 or the Model 2290-10 and the Model 2636B SourceMeter SMU Instrument is included in the Appendix. Note that open source language Python™ [2] is used to send the information to the GPIB interface.

Conclusion

Testing high voltage semiconductor devices involves a consideration of test system safety, wide voltage range, and accurate current measurement. Coupling a Keithley Series 2290 Power Supply with a Keithley SourceMeter SMU instrument and their associated accessories meets all these needs and further facilitates research of high voltage materials and semiconductor devices.

[2] Find out more about Python programming language at http://www.python.org

Appendix

```
# Import pyVisa and time modules into the Python
environment
import visa
import time

# Open a VISA session with the 2290 at GPIB address 14
# and 263xB at GPIB address 26
ki2290 = visa.instrument("GPIB::14")
ki263x = visa.instrument("GPIB::26")

# Reset and clear the status of the 263xB
ki263x.write("reset()")
ki263x.write("*CLS")

# Reset and clear any errors of the 2290
ki2290.write("*RST")
ki2290.write("*CLS")
ki2290.write("*RCL 0")

# Configure the 263xB as an ammeter, set the current
limit
# and current measurement range
ki263x.write("smua.source.rangev = 0.2")
ki263x.write("smua.source.levelv = 0")
ki263x.write("smua.source.limiti = 1e-3")
ki263x.write("smua.source.autorangei = 1")
ki263x.write("smua.measure.lowrangei = 100e-9")

# Configure the display of the 263xB and turn on the
output
ki263x.write("display.screen = display.SMUA")
ki263x.write("display.smua.measure.func = display.
MEASURE_DCAMPS")
ki263x.write("smua.source.output = smua.OUTPUT_ON")

# Define sweep variables for the programmed output
voltage
# and measured current readings
voltage = 0
currReading = ""
currRdgList = []
```

```
# Turn on the output of the 2290
time.sleep(1)
ki2290.write("HVON")

print "Running Sweep . . . "

# Perform a sweep from 0 to 4500V and make current
measurements
# at each point of the sweep
for n in range(0,51):
    ki2290.write("VSET " + str(voltage))
    time.sleep(2) # Allow new voltage level to stabilize
    currReading = ki263x.ask("print(smua.measure.i())")
    time.sleep(1) # Allow measurement to be taken
    currReading = float(currReading)
    currRdgList.append(currReading)
    voltage = voltage + 100

# Set the voltage of the 2290 to 0V and turn off its
output
ki2290.write("VSET 0")
ki2290.write("HVOF")

# Turn off the output of the Model 263xB
ki263x.write("smua.source.output = smua.OUTPUT_OFF")

# Print the current measurements
print "Sweep Complete. Current Measurements: ",
currRdgList
```

Creating Multi-SMU Systems with High Power System SourceMeter® Instruments

Introduction

The design and configuration of test systems for DC characterization of power semiconductor devices using high voltage and high current source measurement units (SMUs) involves several steps:

- Selecting equipment to meet test demands
- Selecting cabling and fixturing to connect the instruments to the device under test (DUT)
- Verifying system safety and instrument protection
- Optimizing the instrument setup to ensure measurement integrity
- Controlling the instrumentation hardware

Power semiconductor discrete devices are designed so that in the ON state, a device delivers a lot of power to the load and consumes minimal power from the power source (high efficiency); in its OFF state, the device delivers nearly zero power to the load and consumes minimal power from the power source (standby current is small). Therefore, characterization or DC parametric test of power semiconductors can be broken into two categories: ON-state and OFF-state characterization. This application note considers the application of test to these two categories. Specific examples of test systems built with a variety of Keithley SourceMeter® Source Measurement Unit (SMU) instruments will also be presented.

Select Equipment to Meet Test Demands

Power devices typically require high power instrumentation at only one or two terminals. For example, characterizing the OFF-state of a high voltage n-channel MOSFET requires a high voltage supply at the drain; all other terminals are driven with lower voltage supplies. Conversely, when characterizing the ON-state performance, high current flows from drain to source, thereby requiring that only those two terminals be rated for maximum power. Test researchers who are making the transition from testing lower power devices to higher power devices can reuse some of their existing test equipment at the gate and substrate terminals. Being able to use the same test equipment for multiple devices allows users to maximize their return on investment.

In order to select appropriate test equipment, it's essential to know the minimum and maximum current and voltages that will be necessary to source and measure. If at all possible, select equipment that has the capability to extend beyond these values in order to accommodate the development of new devices.

Keithley's Series 2650A and 2600B SMUs were designed with evolving test systems in mind. The TSP-Link® inter-unit communication bus supports creating mainframe-less systems while still allowing sub-microsecond synchronization of multiple SMU channels.

One of the most powerful features of the Series 2650A and 2600B is the ability it offers to build systems that address all of the application's test requirements while maintaining seamless system performance. The Series 2650A and 2600B families includes 11 models that offer a variety of functions and capabilities:

- Up to 50A pulse at 2000W (100A possible with two SMUs)
- Up to 3kV source at 60W, 1500V at 180W
- Sub-picoamp measurement capability
- Up to 1A or 3A DC on lower-power SMUs. This is ideal when testing high power BJTs with large base currents.

This level of capability is generally unavailable in an off-the-shelf commercial test mainframe and would have once required configuring a custom or semi-custom ATE. Moreover, using stand-alone instruments allows the test engineer to add new capabilities as new test needs evolve. Stand-alone high power SMUs can extend the current and voltage capabilities of semiconductor parametric analyzers and, therefore, the scope of devices that can be tested.

Selecting Cabling and Fixturing to Connect the Instruments to the Device

Determine the Interface to the Device

In the past, most power semiconductor manufacturers had to package a device in order to test it because there was no widely available technology that allowed delivering tens of amps or thousands of volts to a device on wafer.

The availability of commercial prober solutions is allowing many manufacturers to seize the opportunity to lower their cost of test by testing devices on wafer.

Deciding whether to test packaged devices or devices on wafer is a balance between the large capital costs of a prober versus the smaller (but repeated) costs of packaging devices prior to test. Keithley solutions apply to both packaged test and wafer-level testing.

For testing packaged devices, system developers should take advantage of commercial test fixtures, paying attention to the supported device packages and any opportunity for customization. Keithley offers the Model 8010 High Power

Test Fixture, which is equipped to measure devices in TO-220 and TO-247 packages using supplied and optional device test boards. Additionally, a device test board supplied with the Model 8010 allows customization for connecting to a variety of device packages. System developers can create a custom circuit or can use clips and an insulating block to connect to a device in an unsupported package.

The Model 8010 supports the following Keithley SMUs:

- Up to two Model 2651A High Current High Power SourceMeter Instruments (connected in parallel to allow up to a 100A pulse)
- One Keithley Model 2657A High Voltage High Power SourceMeter Instrument
- Up to two of the following SMUs: Models 2611/12B, Models 2635/6B, Model 4200-SMU, or Model 4210-SMU

Overvoltage protection circuitry for the lower voltage SMUs is installed in the Model 8010 so that instruments are protected in the event that a device failure results in an overvoltage from the Model 2657A. With the Model 8010, SMUs may be connected to up to three terminals on the device.

For some, a custom fixture may be necessary to handle packages or device types/terminals that commercial fixtures don't support. In such cases, consider the following guidelines:

- Plan for the variety of device package types to be tested.
- Find or design a socket that is rated for the maximum voltages and currents in the system.
- For high current testing, use a socket that has Kelvin connections. This ensures that four-wire connections are made all the way to the device pins and provides a true measure of the device characteristics without voltage errors induced by lead resistance in the socket.
- For high voltage testing, ensure that the socket is built with high quality insulators to ensure that measurements can be made to sufficiently low current levels.
- Use triaxial connections if possible for high voltage testing. For the Model 2657A High Power System SourceMeter Instrument, use the Keithley Model HV-CA-571-3 panel-mount HV triax to unterminated cable assembly to create a test fixture with high voltage triaxial connections. Ensure that cables are properly terminated for high voltage creepage and clearance specifications.
- Connect the enclosures of conductive fixtures to safety earth ground. Ensure that non-conductive fixture enclosures are rated to a voltage that's twice the maximum voltage of the test system. Follow all other safety precautions.

For testing devices on wafer, select a prober vendor with experience in testing with high voltage and high current on wafer. Power semiconductor devices are typically vertical devices with high voltage or current connections terminals on the backside of the wafer. This vertical orientation allows device designers to achieve higher breakdown voltages. The wafers are typically ultra-thin, which results in devices with lower ON resistance. Knowledgeable prober vendors have experience with a variety of chuck materials and chuck designs in order to achieve low contact resistance during ON-state characterization. In addition, they will have sufficient expertise in obtaining good low current measurements through the large chuck surface and will be able to provide guidance on selecting the appropriate cabling to meet specific needs.

Connecting the Instrumentation

Connections from the instrumentation to the device under test (DUT) are crucial to obtaining meaningful results. When using the Keithley Model 8010 High Power Test Fixture, connecting the instrument using Keithley supplied cables is straightforward. Review ***Figure 22*** in the "Example Systems" section of this document for a connection diagram. If connections are made to a prober or to a custom fixture instead, keep these guidelines in mind:

1. **Select the best cables for high current test.**

   Ensure that the cables used during test are rated for the maximum current in the test system. Use cables designed to achieve performance required for the high current, low voltage measurements commonly encountered during ON-state characterization.

   In high current testing, pay attention to lead resistance and lead inductance to avoid voltage source and measurement errors.

   Lead resistance

   Some power devices have ON resistances in the range of a few milliohms. Therefore, lead resistance can be on the same order as the parameter under test. When high current is applied, a small amount of lead resistance can result in voltage errors. Small amounts of offset or noise in the voltage measurement can lead to large errors in the ON-resistance result.

   Note the example of an RdsOn measurement of a MOSFET setup in ***Figure 1***, which depicts the instrument configuration. ***Figure 2*** illustrates how the lead resistance is large relative to the device resistance and how the lead resistance results in an 80% measurement error.

   To eliminate errors, use separate cables for the voltage measurement as depicted in ***Figure 3***; use additional cables to connect from the sense terminals of the instrument to the DUT. The test current travels in one set of cables and the voltage measurement is made through the sense lines in which nearly zero current is flowing.

   In high current testing, four-wire connections, also called Kelvin connections, are a must for accurate low voltage and low resistance measurements. To maintain good measurements, monitor force lead resistance to avoid exceeding the SMU's specification for the maximum voltage drop between the force and the sense leads.

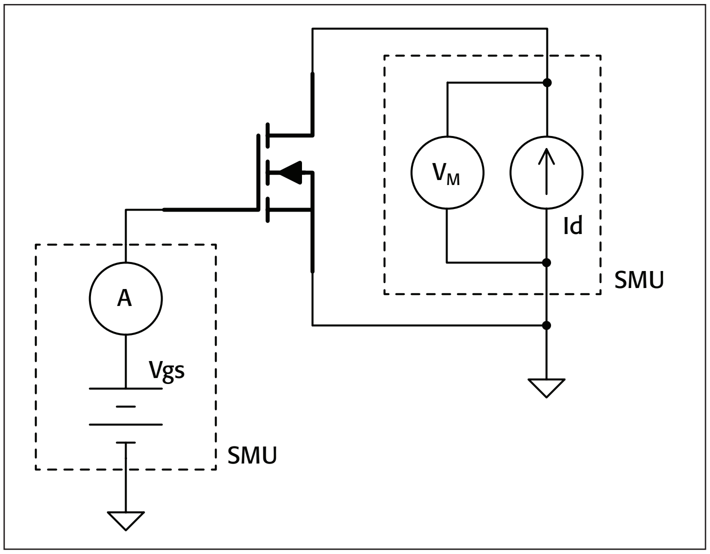


Figure 1. Typical instrument configuration for measuring RdsOn of a power MOSFET.

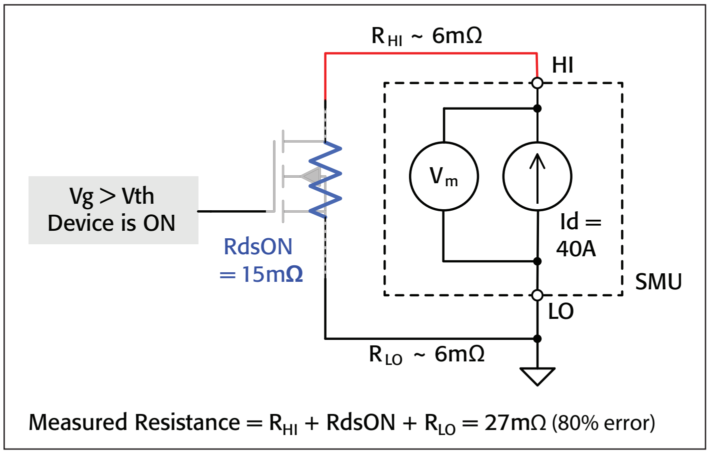


Figure 2. Resistance of test leads is large relative to the DUT resistance. Because the voltage measurement is made at the instrument's output terminals in a two-wire configuration, the measurement includes the sum of the test lead resistance and the DUT resistance.

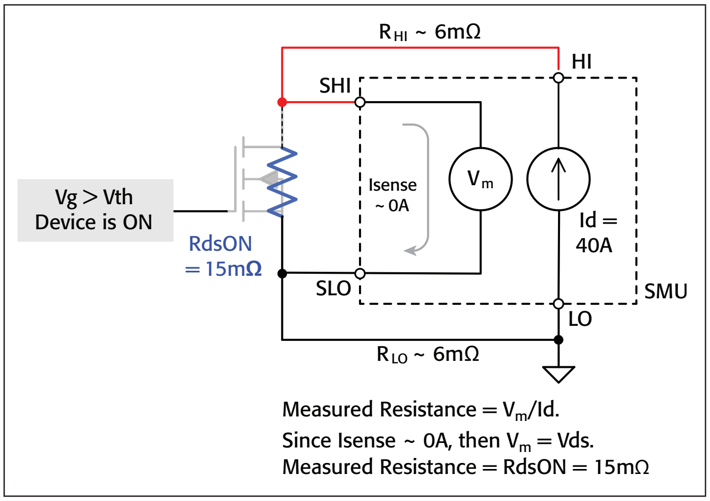

R_HI ~ 6mΩ
HI
SHI
Vg > Vth
Device is ON
Isense
~ 0A
V_m
Id =
40A
RdsON
= 15mΩ
SLO
SMU
LO
R_LO ~ 6mΩ
Measured Resistance = V_m/Id.
Since Isense ~ 0A, then V_m = Vds.
Measured Resistance = RdsON = 15mΩ

Figure 3. Use separate test leads to connect the device to the instrument's sense terminals. In this way, the voltage measured is only that across the device. The resulting resistance measurement will be a true measurement of the DUT resistance.

Lead inductance

Excessive inductance results in voltage overshoots when there is a large change in current over a short period of time (*V = L[di/dt]*). This is especially important for pulse testing, when *dt* can be small. Excessive inductance requires the instrument to force more voltage in order to achieve the desired voltage at the DUT.

Keithley Model 2651A-KIT-1 cables are designed to have low resistance and inductance and are recommended whenever the Model 2651A High Power System SourceMeter Instrument is used.

2. **Select the best cables for high voltage test.**

Ensure that cables used during test are rated for the maximum voltage in the test system. Use cables designed to achieve the performance required for the high voltage, low current measurements common in OFF-state characterization.

In high voltage testing, ensure sufficient insulation and minimize the effect of leakage currents and system capacitance.

Proper insulation

Use a cable rated for at least the maximum voltage of the test system. Use high quality insulators in test fixtures to achieve low current measurements. The insulation resistance is in parallel with the resistance of the DUT and induces measurement errors. (See ***Figure 4***.) With the Model 2657A, up to 3kV may appear across the test circuit, so the amount of current generated through these insulators can be significant relative to the current to be measured through the DUT. For good results, ensure that the insulation resistance is several orders of magnitude higher than the resistance of the DUT.

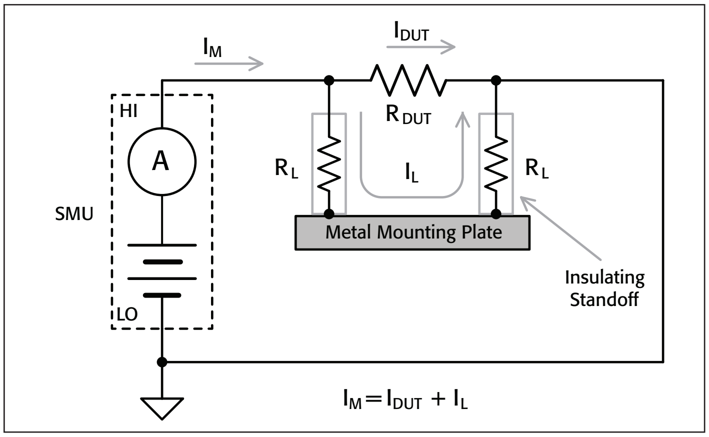


Figure 4. The current generated in the insulators affects the DUT current measurement. To minimize measurement error, ensure that the resistance of insulators (R_L) is much higher than the resistance of the device (R_{DUT}).

Leakage currents and system capacitances

Use guarding to minimize the effect of insulators in the test circuit. Guard is a low impedance terminal that is at the same potential as the high impedance terminal. In ***Figure 4***, the leakage from the insulators is still present, even with high quality insulators. This leakage may be problematic when measuring currents in the nanoamp range. Note how guarding improves the measurement in ***Figure 5***. The leakage currents are routed away from the high impedance measurement terminal (HI), which eliminates them from the measurement.

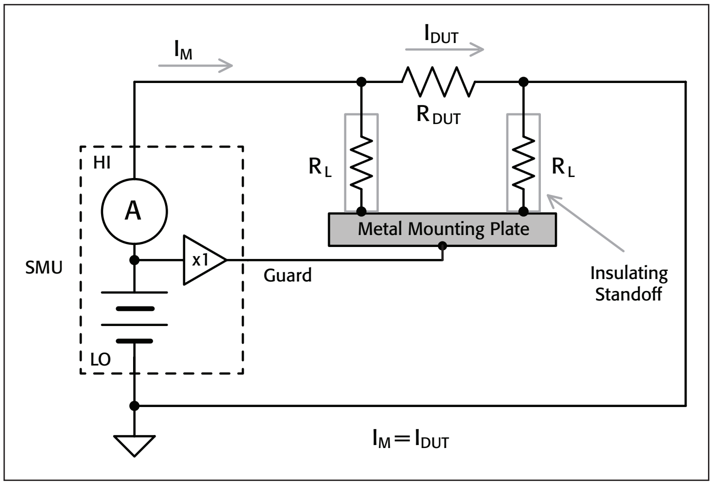
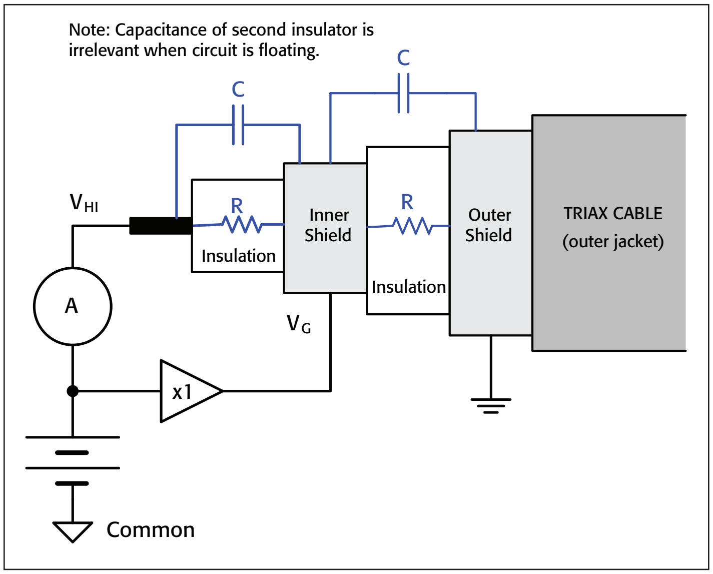


Figure 5. Guard is used to reduce leakage currents by reducing the voltage across the insulators in the circuit to nearly 0V. Any remaining leakage is routed away from the HI terminal where the measurement occurs.

Because the guard terminal is at the same potential as the high impedance terminal, the guard voltage is a hazardous voltage. Use triaxial cabling to carry guard and protect operators from risk of an electric shock. In a triaxial cable, the high impedance terminal is connected to the center conductor, guard is carried on the inner shield, and the outer shield is connected to ground. ***Figure 6*** shows the cross-section of a triaxial cable.

Guarding can also minimize the effect of system capacitances. System capacitances impact the settling of the voltage source and the current measurement. The test setup must allow for the charging of such capacitances and the settling of currents at or below the expected noise floor of the device measurement. The high impedance nature of these setups results in long settling times. ***Figure 6*** illustrates how guarding can reduce the impact of cable capacitance. A typical triaxial cable has a capacitance of about 40pF per foot. This can quickly result in several hundred picofarads of capacitance for a cable that is two or three meters long, and tens of milliseconds of voltage settling time, depending on the maximum current in the test setup. Placing guard on the inner shield of the cable means there is no voltage drop across the cable's insulator. Therefore, the capacitance of this insulator does not need to be charged. In steady-state conditions, the Model 2657A is specified so that the guard voltage is within 4mV of the high impedance terminal (HI). The Keithley Model HV-CA-554 is a high voltage triaxial cable that safely carries signal and guard voltages up to 3280V. Keithley Model HV-CA-554 cables are designed to meet the demands of a 3kV, low current measurement system.

To minimize settling times and leakage current, carry the guard from the SMU all the way to the device pins. Doing so eliminates the need to charge other system capacitances. Because the guard voltage can be up to 3kV, make sure that the guard is terminated a safe distance away from other conductors.



Note: Capacitance of second insulator is irrelevant when circuit is floating.
C
C
V_{HI}
R
Insulation
Inner Shield
R
Insulation
Outer Shield
TRIAX CABLE (outer jacket)
A
V_G
x1
Common

Figure 6. When $V_{HI} \approx V_G$, the voltage drop across the capacitor and resistor is 0V. Guarding virtually eliminates leakage current due to cable insulation and minimizes response time by eliminating the need to charge cable capacitance.

Converting to coaxial connection is a necessity in some systems. For high voltage testing, SHV is the industry standard coaxial connector. Keithley offers Model SHV-CA-553 cable assemblies that allow adapting from high voltage triaxial to SHV. These cable assemblies use a triaxial cable so that the guard is carried as far as possible before it must be terminated so that the connection can be made to SHV. Using coaxial connections results in degraded performance because the benefits of guarding are lost from the point at which the guard is terminated. This means that the remaining cable capacitance and the capacitance within the test system must be charged.

When designing a test fixture, a user can take steps to minimize the capacitances by shortening the length of the cabling and device connections after the conversion to coax from triax.

The effects of converting to coax become even more significant on a probe station, where cabling and connections are a function of the size of the wafer and device orientation (vertical or lateral). When cable capacitances are taken into account, capacitances in a probe station can easily be on the order of nanofarads, resulting in significant capacitive charge time and measurement settling time.

3. **Establish a common reference for the SMUs.**

Few test system issues are as misunderstood as grounding. Here, a "ground" is defined as a connection to earth ground.

However, many people often use the word “ground” to refer to the reference point for the SMUs in the test circuit. In this note, this reference point will be referred to as “circuit common.”

Earth ground

For safety, most systems have a connection to earth ground to ensure that any faults within the instrument or test system do not expose the user to the risk of electric shock. For similar reasons, in high voltage systems, connect conductive test fixtures and their associated accessories to earth ground.

Circuit common

Identifying circuit common is important in order to obtain accurate source values and measurements. When connecting multiple supplies to a DUT, it is important that the supplies be referenced to the same point so that the desired voltages appear at each DUT terminal. Consider the example shown in ***Figure*** 7.

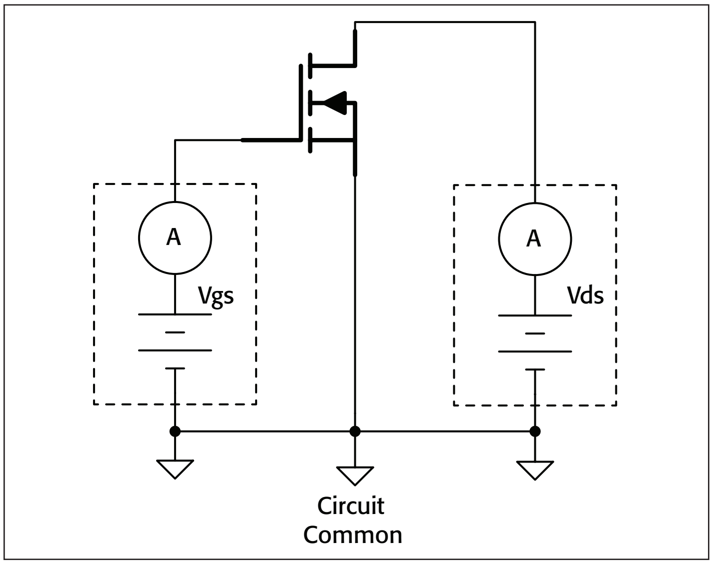


Figure 7. When using stand-alone instruments, the outputs must have the same reference so that the correct voltages and current appear at the device under test (DUT). In this example, the source terminal of the FET must be connected both to the LO of the gate supply and the LO of the voltage supply so that V_{GS} and V_{DS} are accurate. Because the LO terminals of both instruments are connected to the source terminal, this becomes circuit (or measurement) common.

Performance of the device is specified based on the relationship between V_{DS} and V_{GS}. We will consider connections to circuit common in the light of the two test configurations: ON-state characterization with a Model 2651A SMU on the drain and OFF-state characterization with a Model 2657A SMU on the drain.

Creating circuit common when performing ON-state characterization using the Model 2651A high current SMU

The section titled “Selecting Cabling and Fixturing to Connect the Instruments to the Device” addresses why four-wire connections are required for the high current instrument. Four-wire connections are also recommended for the SMU connected to the base of a BJT or gate of a MOSFET or an IGBT, even when there is little current flowing through the gate. Let’s examine the reasons for this recommendation as it relates to the connections to circuit common.

Note the measurement configuration in ***Figure 8*** where ON-state characterization is performed on a power MOSFET. This configuration could apply for generating a family of curves for the MOSFET. Circuit common is created when the LO of the gate SMU (SMU 1) is tied to the LO of the drain SMU (SMU 2). Because little to no current is flowing in the gate-to-source loop, the gate SMU is measuring and correcting the output voltage based on the measurement at its force terminals, which is the difference between the voltage at the gate terminal and the voltage at circuit common, denoted as node S´ (prime) in ***Figure 8***. Circuit common is tied to the Source terminal of the FET (node S in ***Figure 8***) through a test lead that has a resistance R_{slead}. Because a high current (up to 50A pulse) is flowing in the drain-to-source loop, we cannot ignore R_{slead}. Even a 1mΩ resistance here may cause a difference between V_{GS} and V_{GS}´ (prime) of 50mV. Some devices are very sensitive to changes in the gate-to-source voltage. A 50mV difference in V_{GS} may cause hundreds of milliamps or even an amp or more difference in the drain current. To compensate for the voltage drop between the connection in circuit common and the actual device terminal, connect the sense terminals of the gate SMU separately to the DUT as shown in ***Figure 9*** [1]. Because nearly zero current is flowing in the sense leads, the gate SMU accurately measures the device voltage at the FET’s Source terminal and can correct its output voltage to maintain the desired V_{GS} at the device.

In some cases, the response of the gate SMU must be slowed to compensate for ringing or oscillation in the gate circuit. This is done when high capacitance mode is enabled on the gate SMU. However, this slower response time can slow the feedback between the measurement of the sense voltage and the correction of the output voltage. In such cases, connect both the LO and Sense LO terminals of the gate SMU to the Sense LO of the drain SMU. Because little to no current is flowing in the gate-to-source loop, no voltage measurement errors occur. However, this should not be done when testing BJTs because significant current can flow between base and emitter.

Creating circuit common when performing OFF-state characterization using the Model 2657A high voltage SMU

For OFF-state characterization, connections between the gate and drain SMUs and the DUT are made as shown in ***Figure*** 7. If four-wire connections are desired, then simply connect the Sense LO terminals of the gate and drain SMUs.

Device faults can result in the presence of high voltages at lower voltage terminals. Therefore, the connection to the gate, source, and substrate must be made with high

[1] As nearly zero current flows in the sense circuit, the Sense LO of the gate SMU can be tied to the Sense LO terminal of the drain SMU without any impact on the voltage measurements.

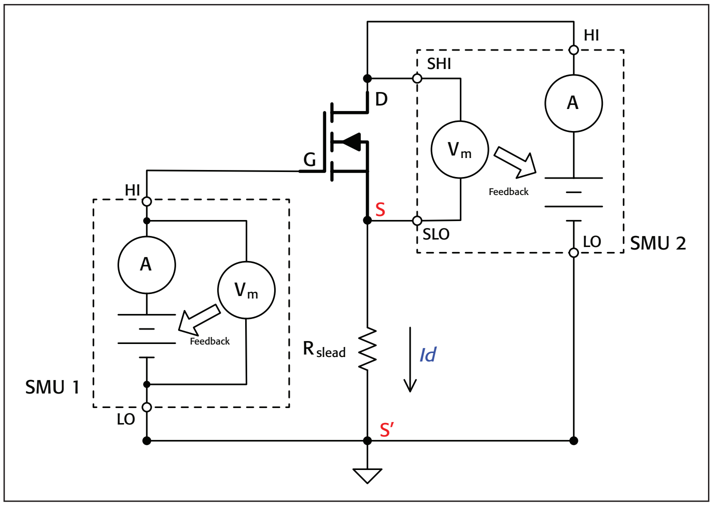


Figure 8. Because high current flows through circuit common, the resistance between circuit common and the FET's source terminal (R_{slead}) will result in voltage differences between measurements made at circuit common and measurements made at the source terminal of the FET. Therefore $V_{GS} \neq V_{GS}'$ (prime) when two-wire connections are used to connect the gate SMU (SMU 1) to the DUT.

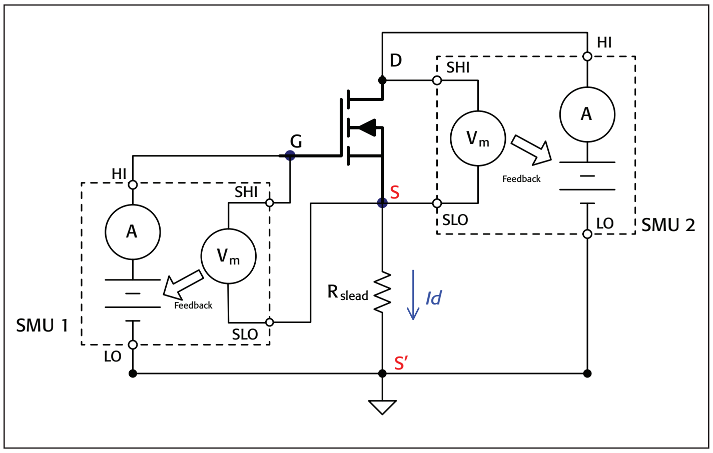


Figure 9. Using four-wire connections to the gate SMU negates the voltage errors that arise because of R_{slead}. In this way, the gate SMU can correct its output voltage to maintain the desired V_{GS}.

voltage connectors. To facilitate connections between the LO and Sense LO of the two instruments, Keithley offers the Model 2657A-LIM-3 LO Interconnect Module as an optional accessory. The LO and Sense LO of three SMUs can be easily routed through the Model 2657A-LIM-3. Additional SMUs can be routed with slight modifications in connections. (Consult Keithley's online FAQs or Keithley field applications engineers for more details.)

Creating circuit common for systems that will use both Models 2651A and 2657A High Power SMUs

Given that complete testing of a power semiconductor device involves both ON-state and OFF-state characterization, it is likely that a test setup will involve both the Model 2651A and the Model 2657A. To ensure the integrity of measurements in both configurations, route the Model 2651A's LO terminal separately from the device under test. Route the Model 2651A's Sense LO through the Model 2657A-LIM-3 so that it is common with the other SMUs in the test setup. Connect the Output LO terminal of the Model 2657A-LIM-3 to the LO terminal of the Model 2651A as close as possible to the DUT. These connections are illustrated in detail in ***Figure 21*** in the section of this note titled "Example Systems."

When performing ON-state characterization to devices on wafer, the connections recommended in the previous paragraph would result in three probe needles touching down on the pad that connects to the Source terminal of the FET. However, making three connections may be a problem because there may be insufficient space on the pad for three probe needles and because pad lifetime decreases as the number of probe touchdowns increases. This problem can be resolved by using the autosense resistor in the Model 2657A-LIM-3. The autosense resistor connects the output Sense LO of the Model 2657A-LIM-3 to the output LO of the Model 2657A-LIM-3 through a 100kΩ resistor. (See ***Figure 10***.) Although true four-wire connections are not maintained to the device under test, there is little impact on the voltage delivered to the FET or IGBT gate terminal because the gate current is small and the Model 2651A's LO is routed through a separate connection to circuit common[2]. ***Figure 21*** in the "Example Systems" section of this application note assumes that the autosense resistor of the Model 2657A-LIM-3 is used to connect to its Output LO and Sense LO terminals.

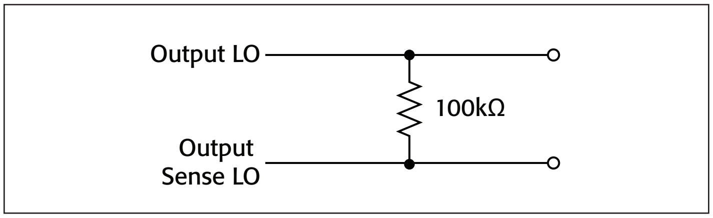


Figure 10. In the Model 2657A-LIM-3, a 100kΩ resistor is used to connect Output Sense LO to Output LO. This allows for quasi-Kelvin sense connections in cases where there is insufficient space to place four connections on the DUT. In cases where full Kelvin is desired, simply connect separate cables to Output Sense LO and Output LO, and the 100kΩ resistor is essentially ignored.

Reviewing System Safety and Instrument Protection

When considering cabling and fixturing design, it is also important to consider system safety. Think through a variety of fault conditions, including those due to operator error and those due to a change in the device state, in order to determine what dangers are presented to the operator and to the instrumentation.

One of the potential dangers of high current testing is the risk of fire or burns. Enclose the device under test to protect users from the potential for fire or flying projectiles if the device fails during high current tests.

[2] If true four-wire connections are not made to the DUT, voltage errors may appear when testing BJTs with high base currents.

The risk of electric shock is one potential danger for high voltage testing. The risk of electric shock exists anytime an instrument (or device) has the ability to output more than 42VDC. Proper test system configurations must include mechanisms for protecting the operator and untrained users from electric shock.

While power is applied, such users should be unable to access systems in which high voltage is present. The safety interlock is one mechanism for limiting access. All modern Keithley SMUs include an interlock so that high power is only enabled when the interlock line is engaged. Interlock is intended to be used with a normally open switch at each system access point. When there are multiple instruments that present a risk of electric shock, wire the instruments so that the outputs of all instruments are disabled if the system access point is opened. When there are multiple access points, each point requires a separate switch and all such switches should be wired in series. This is illustrated in ***Figure 11***.

The 5V supplies from all instruments are combined, externally routed through the switch and used to enable the interlock. When the 5V supplies from multiple SMUs are used, use Schottky diodes to prevent the 5V supplies from each of the SMUs from back-driving each other. Schottky diodes are preferred because they have low forward voltage; therefore, they are less likely to affect the voltage needed to enable the interlock line.

Depending upon the number of SMUs in the test setup, the 5V supply from one SMU may be able to drive the interlock lines of the other SMUs in the system. Review the instrument specifications for the safety interlock pin to determine the amount of current required to drive the interlock pin of each SMU. Also review the specifications for the 5V power supply pin to determine its current capability. Alternatively, an external power supply may be used to drive the interlock lines of the SMUs.

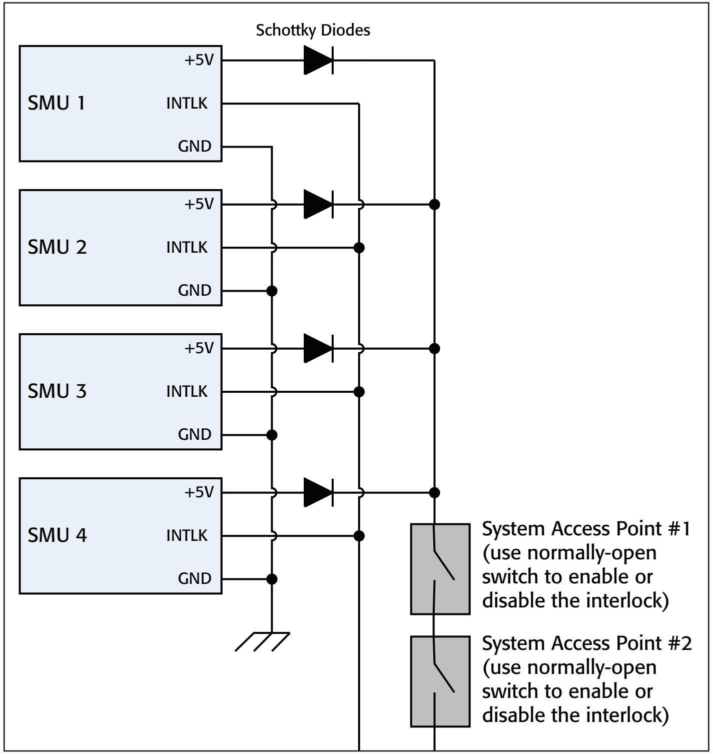

Schottky Diodes
SMU 1
+5V
INTLK
GND
SMU 2
+5V
INTLK
GND
SMU 3
+5V
INTLK
GND
SMU 4
+5V
INTLK
GND
System Access Point #1 (use normally-open switch to enable or disable the interlock)
System Access Point #2 (use normally-open switch to enable or disable the interlock)

Figure 11. Configuration for connecting multiple SMUs to a system with multiple access points.

In addition to operator safety, it is important to protect the investment in the system's test instrumentation. Carefully consider the effect of potential device failures. The test configuration shown in ***Figure 12*** is typical for measuring OFF-state leakage current of an n-channel FET near its specified drain-to-source breakdown voltage. The lower voltage SMU at the gate terminal ensures that the device is in the OFF state. The lower voltage SMU at the source terminal provides a direct measurement of the current at the source terminal. However, if there is a sudden breakdown from drain to source, then there is potential for a 200V SMU to be damaged by the 3kV SMU. The same potential for damage exists between the gate and drain terminals.

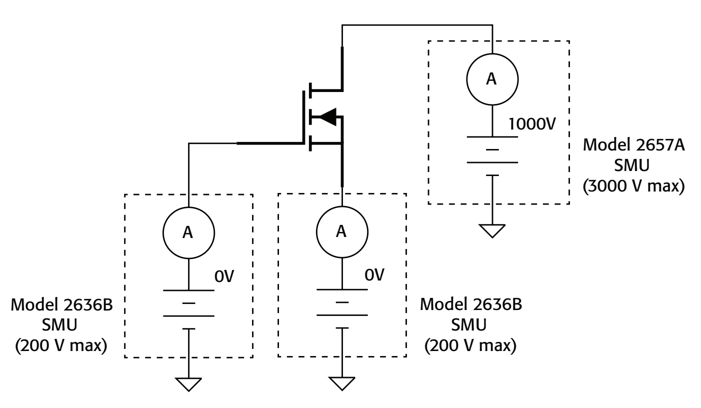

A
1000V
Model 2657A SMU (3000 V max)
A
0V
Model 2636B SMU (200 V max)
A
0V
Model 2636B SMU (200 V max)

Figure 12. In the event of a device breakdown or failure, there is potential for the Model 2657A SMU to damage either Model 2636B SMU.

Use an overvoltage protection device to protect the lower voltage SMU in the event of device breakdown or device failure. The overvoltage protection device should have little effect on the test circuit under normal conditions but will clamp the voltage during an overvoltage condition. The Keithley Model 2657A-PM-200 200V Protection Module is designed for use in test systems that contain both the Model 2657A SMU and lower voltage SMUs. In the event of an overvoltage, the protection module will clamp the external high voltage source to approximately 2V in microseconds. In an unclamped condition, picoamp-level current source and measure capability is maintained on the lower voltage SMU[3].

Instrument damage can also occur when device failures subject the instruments to high current. Ensure that the

[3] The overvoltage protection of the Model 2657A-PM-200 protection module is triggered at ~ 220V. Therefore, it is not recommended for use in SMUs with maximum output voltage capability of less than 200V. Additionally, the Model 2657A-PM-200 is not recommended for use in protecting the HI and LO terminals of the Model 2651A. The protection module is not rated for the maximum output current of the Model 2651A.

instruments at all terminals are capable of handling currents normally present at the specific device terminals. To limit instrument damage when device failure results in higher currents, use series resistors to limit the maximum current that can flow into any instrument. During characterization of breakdown voltage or leakage current, the user must be careful to limit the amount of current through the device so that the testing is not destructive. Although SMUs have programmable current limits, these active limits take some finite time to respond to changes in load (known as the "transient response time.") When the load impedance changes very quickly, then currents above the programmable limit may flow. Adding a series resistor enforces a maximum current at that terminal even under transient conditions.

Make sure to select a resistor rated for the maximum current, voltage, and power in the test system. Such resistors often serve a dual purpose. During ON-state characterization of transistors, resistors are often added in series with the gate to minimize the oscillation and ringing in the gate voltage commonly seen in high gain devices. In addition, during OFF-state characterization, resistors are often used to limit the maximum current that can flow during a breakdown, preventing premature device failure.

Optimizing the Instrument Setup to Obtain Good Measurements

Once the system setup is complete, it's time to test its functionality and optimize instrument setup to obtain the best measurements.

For ON-state Characterization

In most ON-state testing performed today, the device is pulsed on in order to minimize heating. In addition, the end application for many power semiconductor devices involves operating under pulsed conditions. Qualify the test system for pulsing by outputting a pulse of the SMU through the test system and capturing the response at the device terminals. The Model 2651A High Power System SourceMeter instrument includes high speed analog-to-digital converters (ADC) that allow the simultaneous digitization of current and voltage waveforms. These ADCs are useful in determining system pulse performance. If anomalies in the pulse shape occur, review cabling to make sure that lead inductance is minimized. Use the Keithley-supplied low impedance coaxial cabling wherever possible and minimize the inductive loop area of other leads.

For OFF-state Characterization

Understand the source and measure settling time of the system. High voltage sources are generally used to perform OFF-state measurements of the device, where currents are low and the device is in a high impedance state. However, there may be system capacitances that cannot be guarded out and the device itself will have some capacitance. Power semiconductor transistors typically have output capacitances on the order of 100pF or more. Device resistance in the OFF state can be 1GΩ or more, resulting in a single RC time constant of 100 milliseconds or longer. Note the plot of voltage vs. time for a charging capacitor in ***Figure 13***. In order to produce settled readings, it's important to wait at least four to six time constants (four time constants = 99%), which may mean nearly a second of settling time is required. Account for this settling time in estimating production throughput or the time required for long tests, such as device reliability testing.

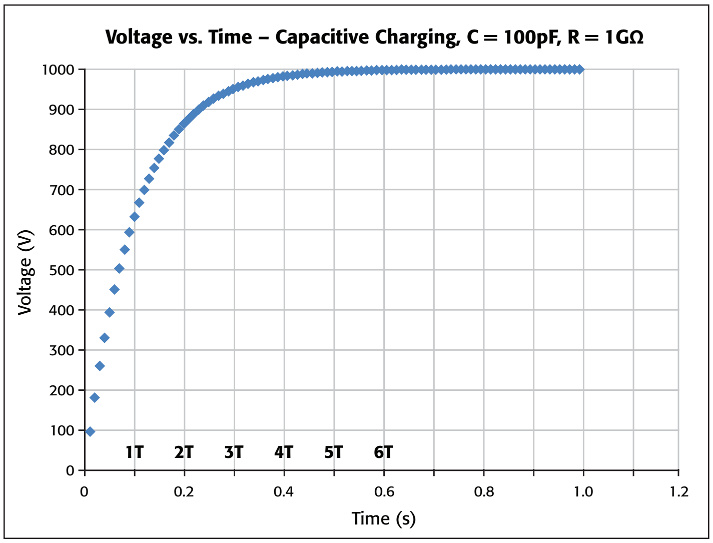


Figure 13. A simulated plot of voltage vs. time for a 100pF capacitor while it is being charged. Readings are settled at four to six time constants. T = RC time constant.

If coaxial connections are used instead of triaxial, be aware that additional settling time will be required to obtain accurate and repeatable low current measurements. In addition to the RC of the device, the settling time must now include the RC time constant of the cable and the probe station or fixture. Characterize the total settling time by applying a voltage step at the system input and measuring the current decay over time at the output. The fast analog-to-digital converter (ADC) of the Model 2657A or Model 2651A affords the user a quick and easy way to capture settling time information. Select a measurement delay by observing the time at which the current falls below the expected noise floor for the device measurement.

Controlling the Instrumentation Hardware

Coordinating the source and measure sequence of the device under test is not trivial when multiple instruments are in use. Therefore, it is advantageous to use available software solutions to simplify and/or eliminate extensive programming.

Use the supplied free start-up software to validate the test system configuration and functionality. Series 2600B and Series 2650A SMUs are provided with TSP® Express software, free start-up software that is served from the web interface of the instrument. When the TSP-Link interface is used to create a network of instruments, TSP Express software can be used to control all of the instruments on the network. See ***Figure 14*** for an example diagram of the communication setup between three SMUs. Using TSP Express software, the user can easily set up DC and pulse sweeps on one or more SMUs and have other SMUs assigned to output static bias voltage or program a step change in bias voltages at each sweep. Built-in simplified graphing software allows the user to plot results quickly and assess whether all cabling, connections, and parameters have been correctly configured. ***Figure 15*** and ***Figure 16*** illustrate the capabilities TSP Express software.

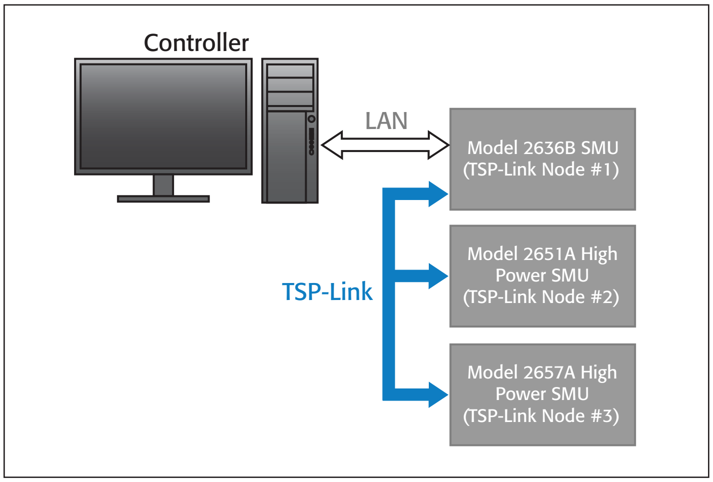

Controller
LAN
Model 2636B SMU (TSP-Link Node #1)
TSP-Link
Model 2651A High Power SMU (TSP-Link Node #2)
Model 2657A High Power SMU (TSP-Link Node #3)

Figure 14. TSP-Link technology provides mainframe-less system expansion and enables communication between the instruments, precision timing, and tight channel synchronization.

TSP Express software can also access the fast ADC capability of the Model 2651A and Model 2657A high power SMUs. Use TSP Express software to output a single pulse from the Model 2651A through the test system and capture its response. These results can be used to set the required source and measure delays to ensure that data is taken during the settled portion of the pulse.

Simplify the testing of discrete semiconductor devices by using software designed for parametric test. Such software includes test libraries with pre-defined tests that simplify gathering data for various devices. Keithley's ACS Basic Edition software is recommended for semiconductor device testing with multiple-SMU systems in which the device is contacted using a manual prober or test fixture. Keithley's ACS Standard software is recommended when interfacing with devices using a semiautomatic or automatic prober.

Example Systems

This section provides detailed connections for several example configurations. For more information on how to apply any of these examples to a specific application, please contact your local Keithley field applications engineer. Worldwide contact information is available at the end of this application note or on www.keithley.com.

Packaged Device Testing Using a Test Fixture

The Keithley Model 8010 High Power Device Test Fixture completes the solution for testing power semiconductor devices with Model 2651A and Model 2657A High Power System SourceMeter instruments. ***Figure 22*** depicts the connections from the instrumentation to the Model 8010. Example device testing configurations are considered in detail in the Model 8010 Interconnect Reference Guide (IRG), which can be downloaded from www.keithley.com.

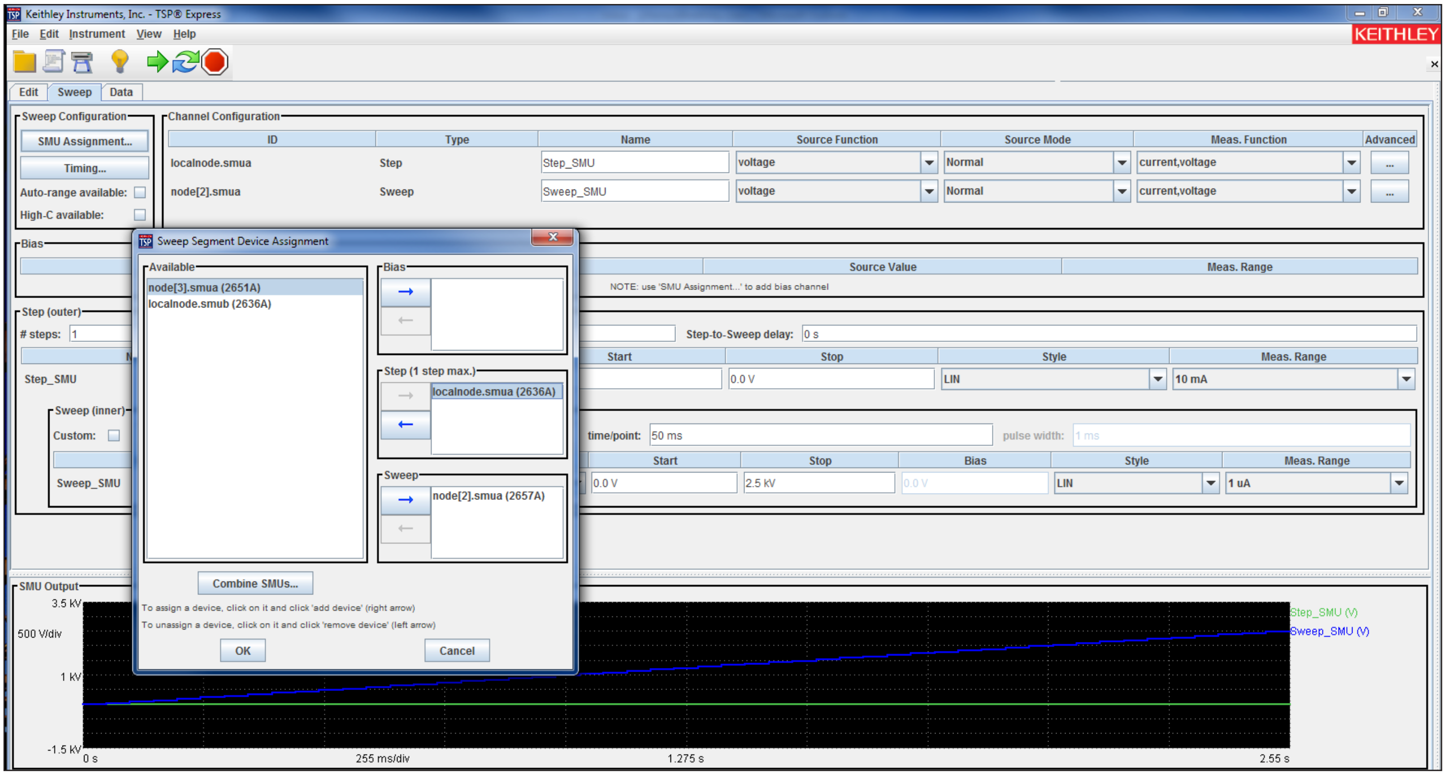
Custom test fixtures can be designed to incorporate any number of SMUs for testing. For high current testing, connections may be easily adapted from the Phoenix screw-terminal connectors provided on the Model 2651A. For highest integrity high voltage measurements using the Model 2657A, Keithley offers a bulkhead version of the custom HV triax connector, already assembled with a triax cable that is unterminated at one end. It is designed to be installed in a safe enclosure. Connections can easily be soldered to the device or adapted to another connector appropriate for the setup. This is illustrated in ***Figure 17***. The legend in ***Figure 18*** applies to this custom fixture connection example.

Wafer-Level Device Testing on a Probe Station

A few probe station vendors offer commercial solutions for high power semiconductor device testing and other vendors create custom solutions. Consult with a probe station vendor to determine the types of probing solutions available for the current voltage, and power required for a specific test application. Prober manufacturers can contact Keithley Instruments for information on how to use our custom high voltage triaxial cable and connector.

As part of a recent review of probe station solutions, we found that banana plugs and jacks are commonly used for high current testing and that SHV is an industry standard for high current connections. The high current Phoenix screw-terminal connectors can be easily adapted to banana connectors. To adapt to SHV solutions, Keithley offers the Model SHV-CA-553, a cable assembly that has a high voltage triaxial connector on one end and an SHV (coaxial) connector on the other end. The diagrams that follow illustrate three example configuration connections to a wafer-level device using banana and SHV connections.

The legend in ***Figure 18*** applies to the wafer-level device connection examples.



Keithley Instruments, Inc. - TSP® Express

Sweep Configuration

SMU Assignment...

Timing...

Auto-range available: □

High-C available: □

Channel Configuration

ID	Type	Name	Source Function	Source Mode	Meas. Function	Advanced
localnode.smua	Step	Step_SMU	voltage	Normal	current,voltage	...
node[2].smua	Sweep	Sweep_SMU	voltage	Normal	current,voltage	...

Bias

Source Value | Meas. Range

NOTE: use 'SMU Assignment...' to add bias channel

Step (outer)

steps: 1

Step-to-Sweep delay: 0 s

Step_SMU — Start | Stop: 0.0 V | Style: LIN | Meas. Range: 10 mA

Sweep (inner)

Custom: □

time/point: 50 ms

pulse width: 1 ms

Sweep_SMU — Start: 0.0 V | Stop: 2.5 kV | Bias: 0.0 V | Style: LIN | Meas. Range: 1 uA

Sweep Segment Device Assignment

Available

node[3].smua (2651A)

localnode.smub (2636A)

Bias

Step (1 step max.)

localnode.smua (2636A)

Sweep

node[2].smua (2657A)

Combine SMUs...

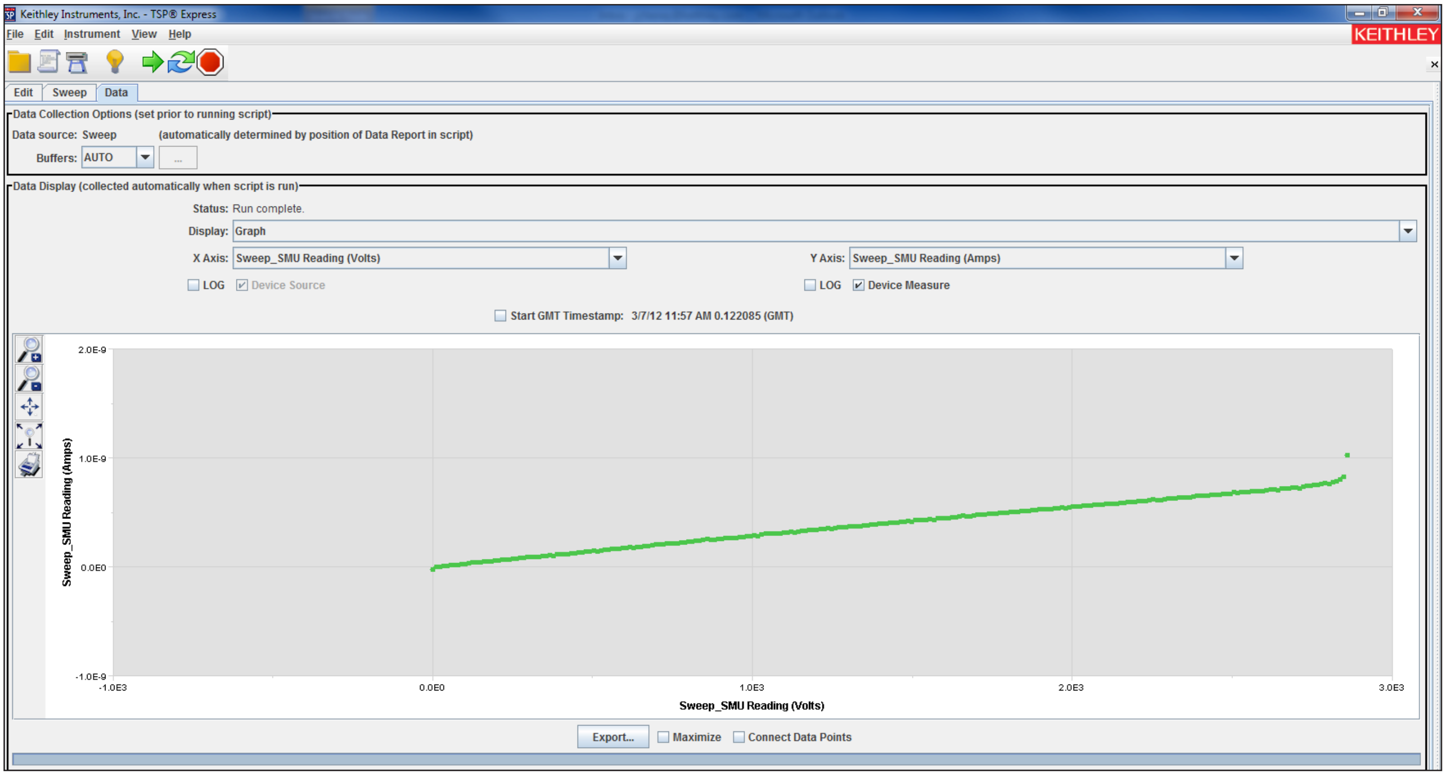
To assign a device, click on it and click 'add device' (right arrow)

To unassign a device, click on it and click 'remove device' (left arrow)

OK Cancel

SMU Output

Figure 15. TSP Express software allows quick setup of source and measurement parameters of all instruments connected over the TSP-Link network.



Keithley Instruments, Inc. - TSP® Express

Data Collection Options (set prior to running script)

Data source: Sweep (automatically determined by position of Data Report in script)

Buffers: AUTO

Data Display (collected automatically when script is run)

Status: Run complete.

Display: Graph

X Axis: Sweep_SMU Reading (Volts)

Y Axis: Sweep_SMU Reading (Amps)

□ LOG ☑ Device Source

□ LOG ☑ Device Measure

□ Start GMT Timestamp: 3/7/12 11:57 AM 0.122085 (GMT)

Export... □ Maximize □ Connect Data Points

Figure 16. TSP Express software allows quick and easy plotting of measurements, such as the V_{ces} vs. I_c plot shown here for an IGBT with a rated BV_{ces} of 2500V.

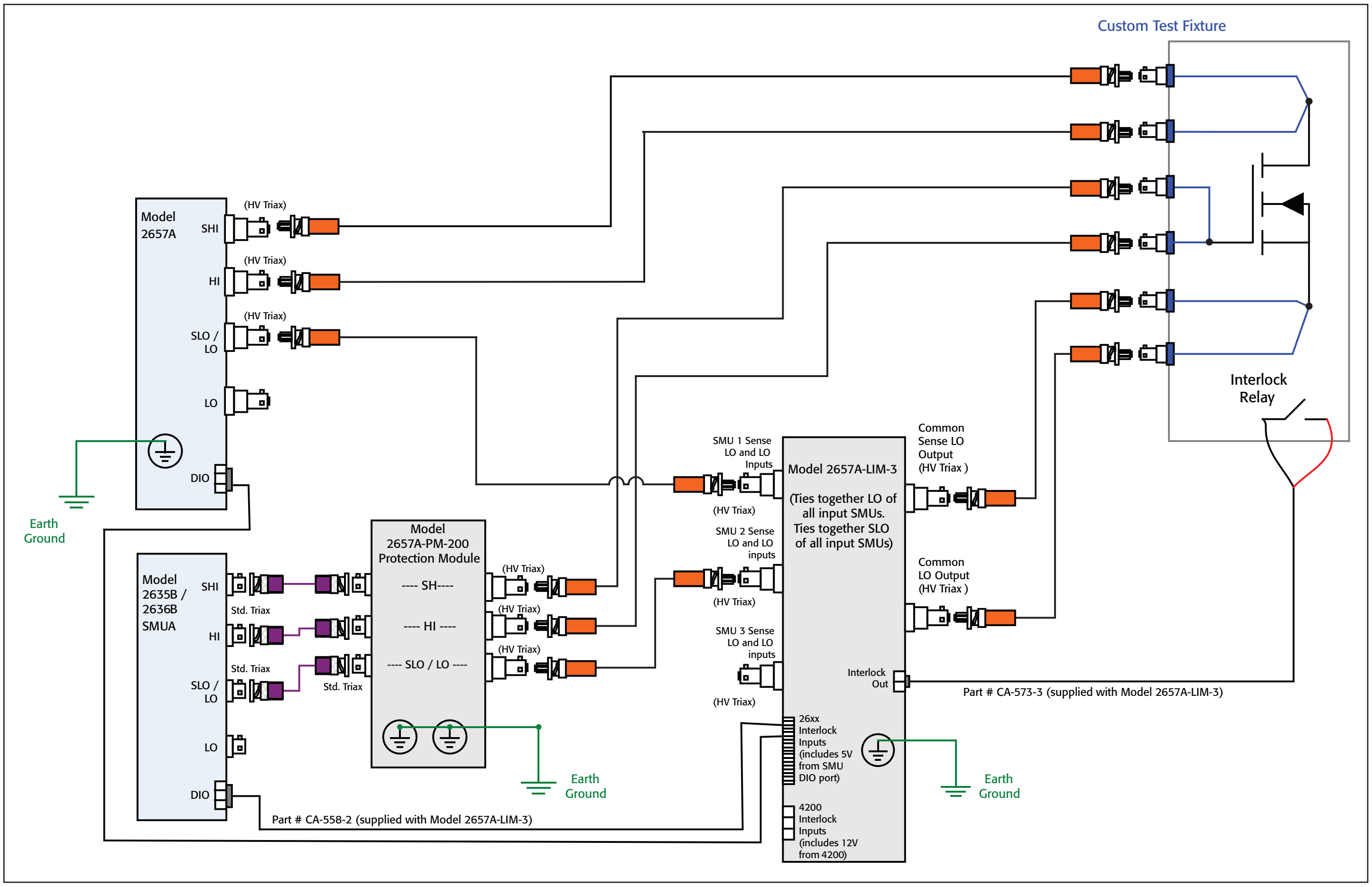

Custom Test Fixture
Model 2657A
(HV Triax)
SHI
(HV Triax)
HI
(HV Triax)
SLO / LO
LO
DIO
Earth Ground
Model 2635B / 2636B SMUA
SHI
Std. Triax
HI
Std. Triax
SLO / LO
LO
DIO
Std. Triax
Model 2657A-PM-200 Protection Module
---- SH----
---- HI ----
---- SLO / LO ----
(HV Triax)
(HV Triax)
(HV Triax)
Earth Ground
SMU 1 Sense LO and LO Inputs
(HV Triax)
SMU 2 Sense LO and LO inputs
(HV Triax)
SMU 3 Sense LO and LO inputs
(HV Triax)
Model 2657A-LIM-3
(Ties together LO of all input SMUs. Ties together SLO of all input SMUs)
Common Sense LO Output (HV Triax)
Common LO Output (HV Triax)
Interlock Out
Part # CA-573-3 (supplied with Model 2657A-LIM-3)
26xx Interlock Inputs (includes 5V from SMU DIO port)
4200 Interlock Inputs (includes 12V from 4200)
Earth Ground
Part # CA-558-2 (supplied with Model 2657A-LIM-3)
Interlock Relay

Figure 17. Custom fixture connection example for high voltage testing.

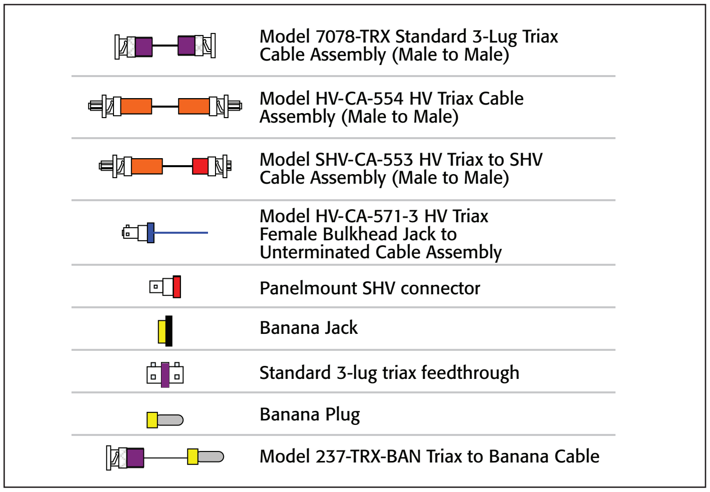

Model 7078-TRX Standard 3-Lug Triax Cable Assembly (Male to Male)
Model HV-CA-554 HV Triax Cable Assembly (Male to Male)
Model SHV-CA-553 HV Triax to SHV Cable Assembly (Male to Male)
Model HV-CA-571-3 HV Triax Female Bulkhead Jack to Unterminated Cable Assembly
Panelmount SHV connector
Banana Jack
Standard 3-lug triax feedthrough
Banana Plug
Model 237-TRX-BAN Triax to Banana Cable

Figure 18. Legend for wafer-level device connection example diagrams.

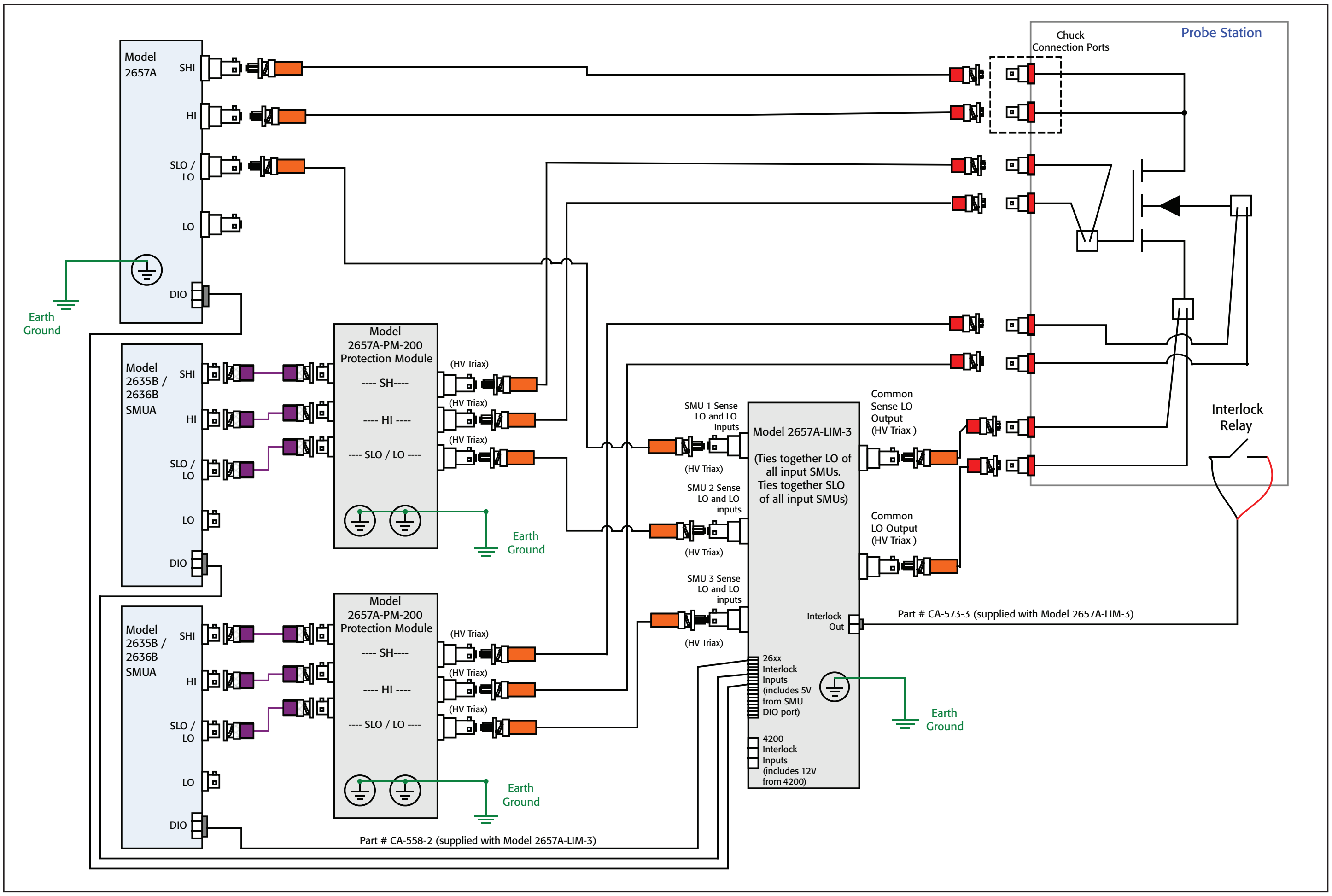

Model 2657A
SHI
HI
SLO / LO
LO
DIO
Earth Ground
Model 2635B / 2636B SMUA
SHI
HI
SLO / LO
LO
DIO
Model 2657A-PM-200 Protection Module
---- SH----
---- HI ----
---- SLO / LO ----
(HV Triax)
(HV Triax)
(HV Triax)
Earth Ground
Model 2635B / 2636B SMUA
SHI
HI
SLO / LO
LO
DIO
Model 2657A-PM-200 Protection Module
---- SH----
---- HI ----
---- SLO / LO ----
(HV Triax)
(HV Triax)
(HV Triax)
Earth Ground
Part # CA-558-2 (supplied with Model 2657A-LIM-3)
SMU 1 Sense LO and LO Inputs
(HV Triax)
SMU 2 Sense LO and LO inputs
(HV Triax)
SMU 3 Sense LO and LO inputs
(HV Triax)
Model 2657A-LIM-3
(Ties together LO of all input SMUs. Ties together SLO of all input SMUs)
Common Sense LO Output (HV Triax)
Common LO Output (HV Triax)
Interlock Out
26xx Interlock Inputs (includes 5V from SMU DIO port)
4200 Interlock Inputs (includes 12V from 4200)
Earth Ground
Part # CA-573-3 (supplied with Model 2657A-LIM-3)
Chuck Connection Ports
Probe Station
Interlock Relay

Figure 19. Connections to wafer-level device for high voltage only testing using SHV connections.

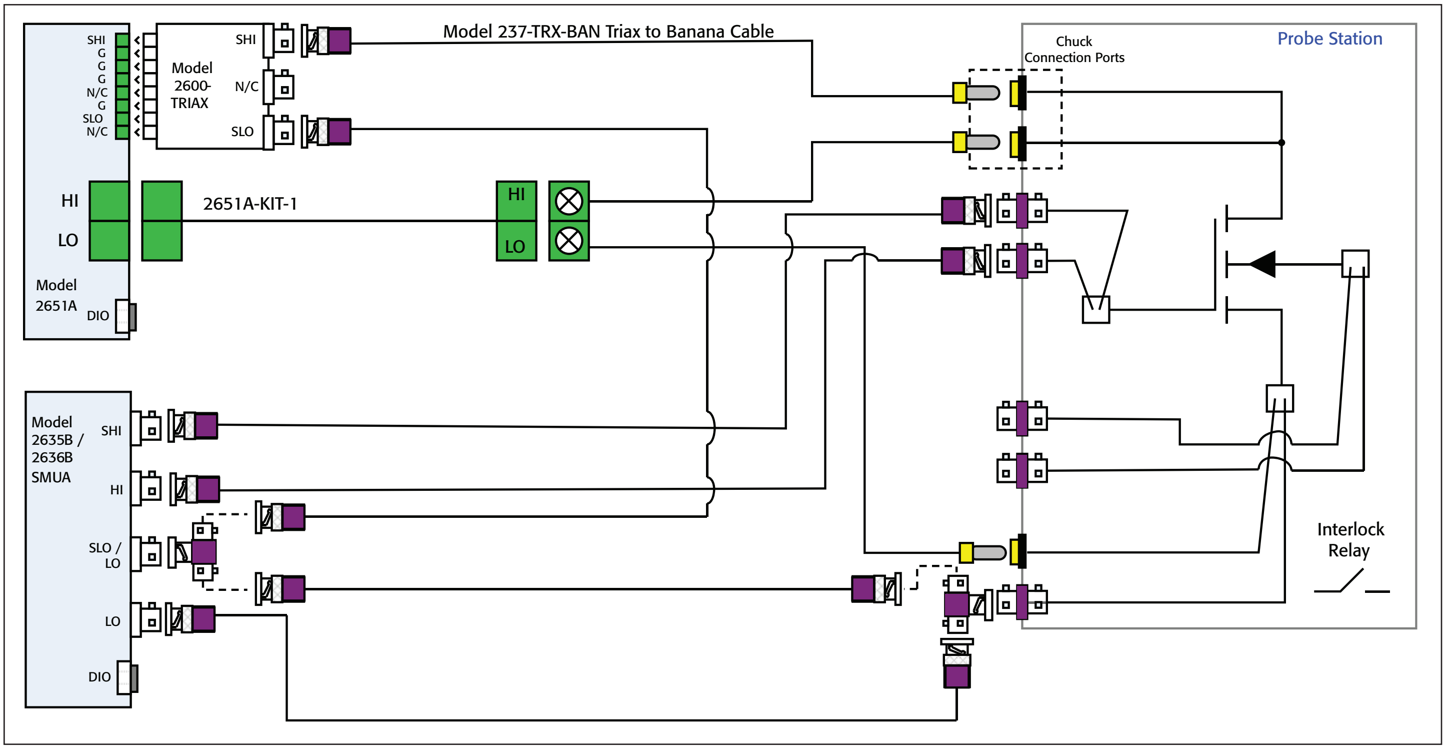


Figure 20. Connections to wafer-level device for high current only testing using banana connections.

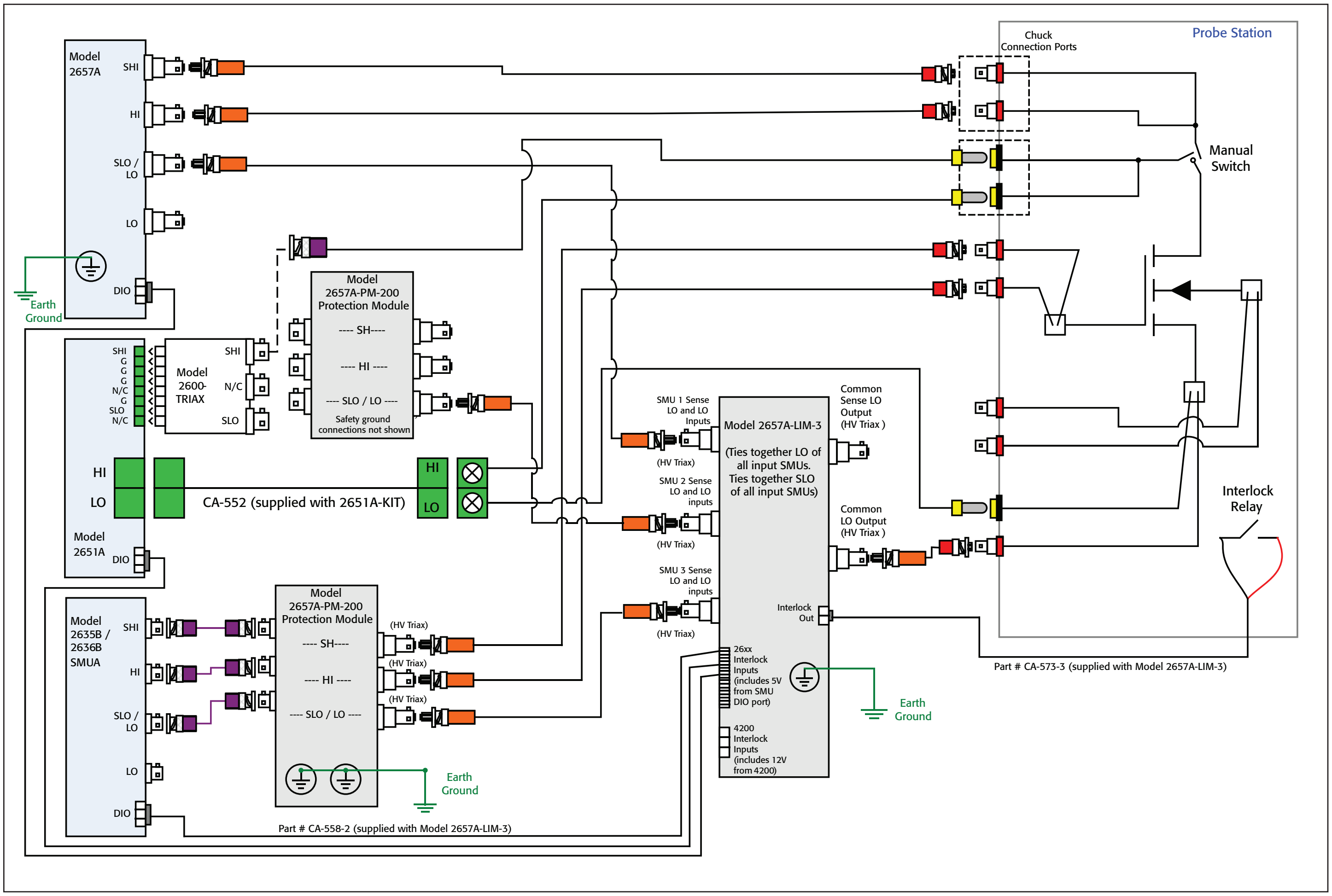

Probe Station
Chuck Connection Ports
Manual Switch
Interlock Relay
Model 2657A
SHI
HI
SLO / LO
LO
DIO
Earth Ground
Model 2657A-PM-200 Protection Module
---- SH----
---- HI ----
---- SLO / LO ----
Safety ground connections not shown
Model 2600-TRIAX
SHI
N/C
SLO
SHI G G G N/C G SLO N/C
HI
LO
CA-552 (supplied with 2651A-KIT)
Model 2651A
DIO
Model 2635B / 2636B SMUA
SHI
HI
SLO / LO
LO
DIO
Model 2657A-PM-200 Protection Module
---- SH----
---- HI ----
---- SLO / LO ----
(HV Triax)
(HV Triax)
(HV Triax)
Earth Ground
Part # CA-558-2 (supplied with Model 2657A-LIM-3)
SMU 1 Sense LO and LO Inputs
(HV Triax)
SMU 2 Sense LO and LO inputs
(HV Triax)
SMU 3 Sense LO and LO inputs
(HV Triax)
Model 2657A-LIM-3
(Ties together LO of all input SMUs. Ties together SLO of all input SMUs)
Common Sense LO Output (HV Triax)
Common LO Output (HV Triax)
Interlock Out
26xx Interlock Inputs (includes 5V from SMU DIO port)
4200 Interlock Inputs (includes 12V from 4200)
Earth Ground
Part # CA-573-3 (supplied with Model 2657A-LIM-3)

Figure 21. Connections to wafer-level device for high voltage and high current test.

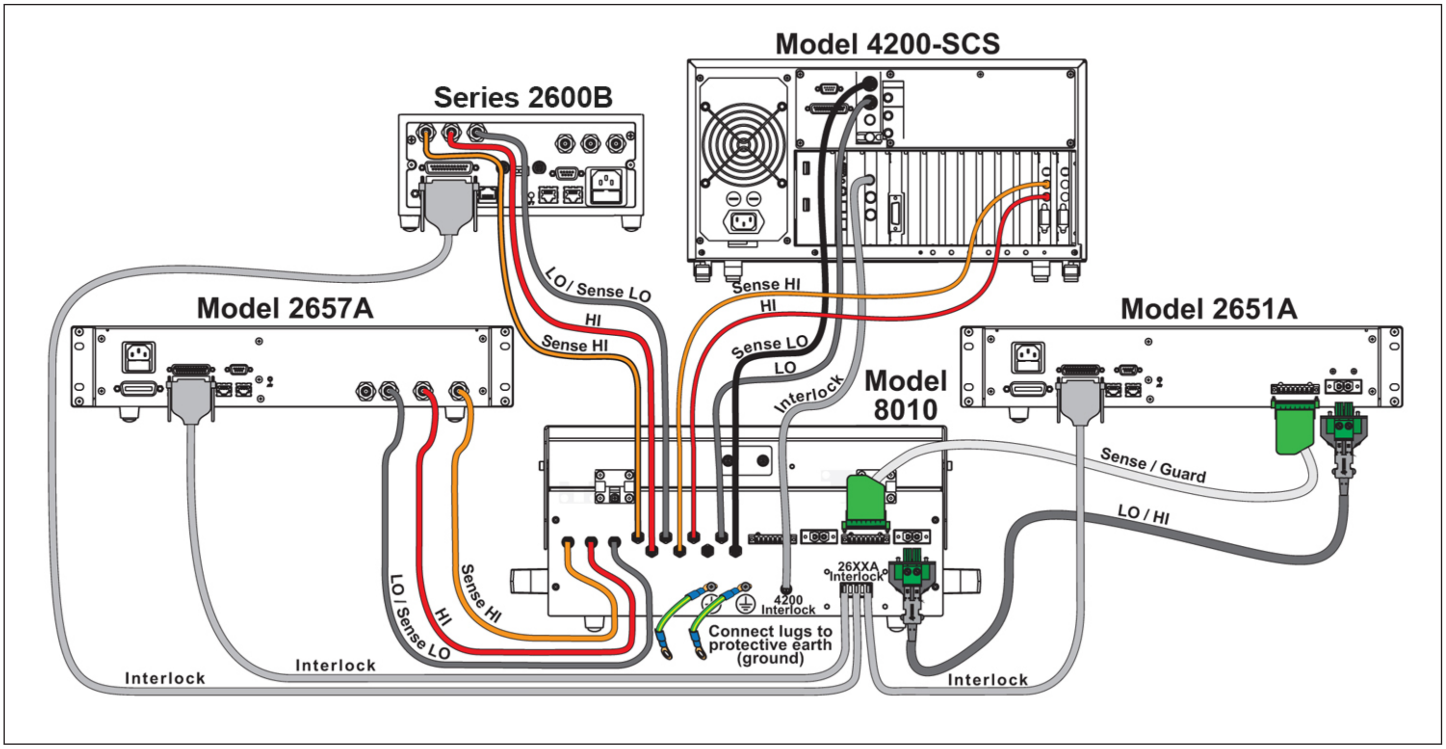


Figure 22. Connections from SMU instruments to Model 8010 High Power Device Test Fixture.

Simplifying FET Testing with Series 2600B System SourceMeter® SMU Instruments

Introduction

Field effect transistors (FETs) are important semiconductor devices with many applications because they are fundamental components of many devices and electronic instruments. Some of the countless applications for FETs include their use as amplifiers, memory devices, switches, logic devices, and sensors. The most commonly used FET is the MOSFET, which is the basis for digital integrated circuits.

Characterizing FETs' current-voltage (I-V) parameters is crucial to ensuring they work properly in their intended applications and meet specifications. Some of these I-V tests may include gate leakage, breakdown voltage, threshold voltage, transfer characteristics, drain current, on-resistance, etc. FET testing often involves the use of several instruments, including a sensitive ammeter, multiple voltage sources, and a voltmeter. Programming and synchronizing multiple instruments, however, can be tedious and time consuming. The use of a turnkey semiconductor characterization system is one alternative approach that solves the integration problem and offers other important benefits, but systems of this type typically cost tens of thousands of dollars. A third approach involves using Source Measurement Units (SMUs) to perform parameter testing on FETs and other semiconductor devices. An SMU is an instrument that can quickly and accurately source and measure both current and voltage. The number of SMUs required in the test usually depends on the number of FET terminals that must be biased and/or measured.

This application note explains how to simplify I-V measurements on FETs using a Series 2600B System SourceMeter SMU Instrument with the TSP® Express embedded software tool. The Series 2600B family of one- and two-channel SMUs offers a range of instruments ideal for electrical characterization of FETs that can source and measure over a wide range of voltage and current. These SMUs have current resolution to 0.1fA and can be current limited to prevent damage to the device. The TSP Express software tool simplifies performing common I-V tests on FETs and other semiconductor devices, without programming or installing software. ***Figure 1*** illustrates a typical test setup.

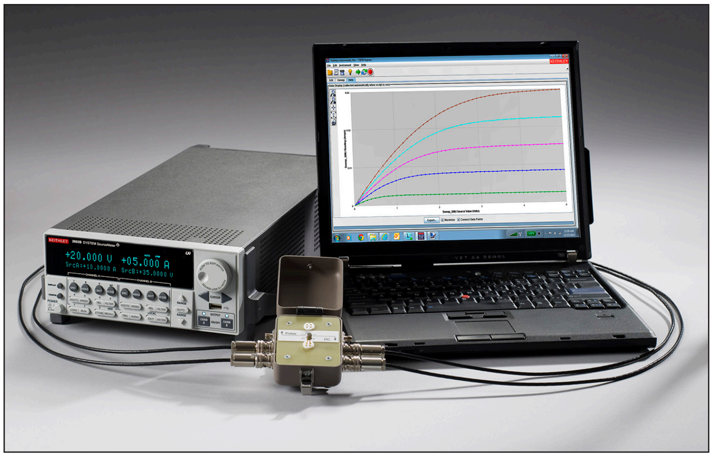


Figure 1. Model 2636B with the TSP Express software tool generating a drain family of curves on a MOSFET

Field Effect Transistors

The field effect transistor is a majority charge-carrier device in which the current-carrying capability is varied by an applied electric field. The FET has three main terminals: the gate, the drain, and the source. A voltage applied to the gate terminal (V_G) controls the current that flows from the source (I_S) to the drain (I_D) terminals.

The many types of FETs include the MOSFET (metal-oxide-semiconductor), MESFET (metal-semiconductor), JFET (junction), OFET (organic), GNRFET (graphene nano-ribbon), and CNTFET (carbon nanotube). These FETs differ in the design of their channels. ***Figure 2*** illustrates the MOSFET, CNTFET, and JFET.

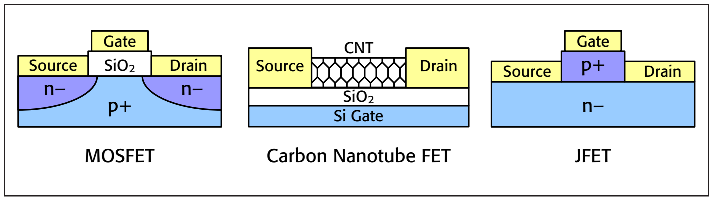


Figure 2. MOSFET (insulated gate), CNTFET (carbon nanotube channel), and JFET (junction FET)

Using a Series 2600B SMU for FET Testing

A FET's I-V characteristics can be used to extract many parameters about the device, to study the effects of fabrication techniques and process variations, and to determine the quality of the contacts. ***Figure 3*** illustrates a DC I-V test configuration for a MOSFET using a two-channel Series 2600B SMU. In this configuration, the Force HI terminal of Channel 1 (SMU CH1) is connected to the gate of the MOSFET and the Force HI terminal Channel 2 (SMU CH2) is connected to the drain. The source terminal of the MOSFET is connected to the Force LO terminals of both SMU channels or to a third SMU if it is necessary to source and measure from all three terminals of the MOSFET.

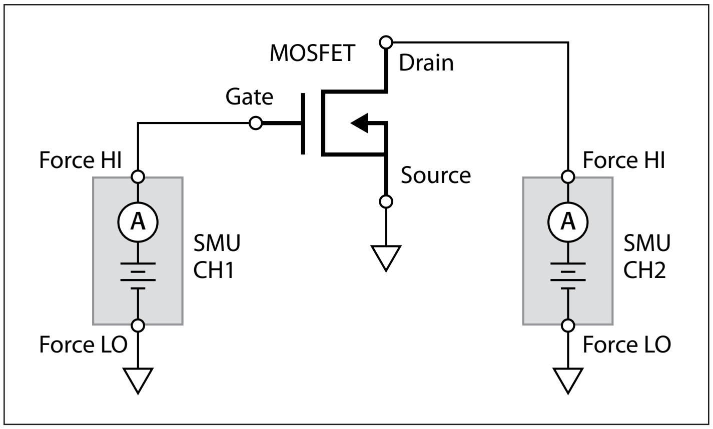


Figure 3. MOSFET test configuration using two SMU channels

Once the device is set up and connected to the SMU, the control software must be configured to automate the measurements, such as with the TSP Express software tool embedded into Series 2600B SMUs. As shown in ***Figure 4***, it's simple to connect the instrument to any computer with the supplied Ethernet cable. Entering the IP address of the SMU into the URL line of any web browser will open the instrument's web page. From that page, the user can launch TSP Express and configure the desired test using the project wizard. These tests, or projects, can be saved and recalled for future use.

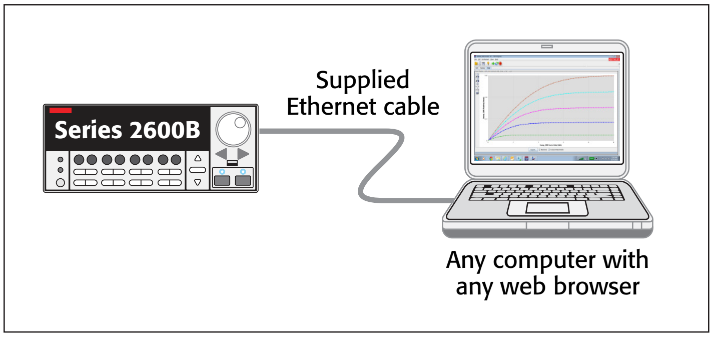


Figure 4. Connecting a Series 2600B SMU to a computer

A common I-V test performed on a MOSFET is the drain family of curves (V_{DS}-I_D). With this test, SMU CH1 steps the gate voltage (V_G) while SMU CH2 sweeps the drain voltage (V_D) and measures the resulting drain current (I_D). Once the two SMUs are configured to perform the test, the data can be generated and plotted on the screen in real time. ***Figure 5*** shows a screen capture of a MOSFET drain family of curves created by using the two-channel Model 2636B SMU with the TSP Express software. This I-V data can be exported directly to a .csv file with a single button click, then imported into a spreadsheet for further analysis.

The data in the software can also be displayed in a table view (***Figure 6***). For each SMU channel, the current, voltage, and time appear in the table. This data can also be saved to a file.

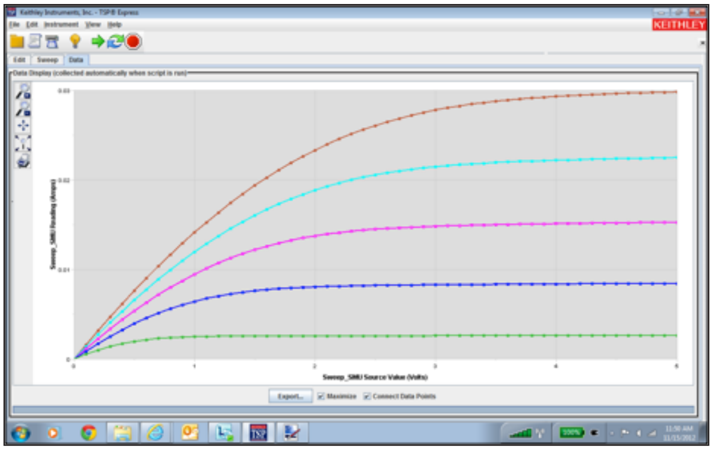
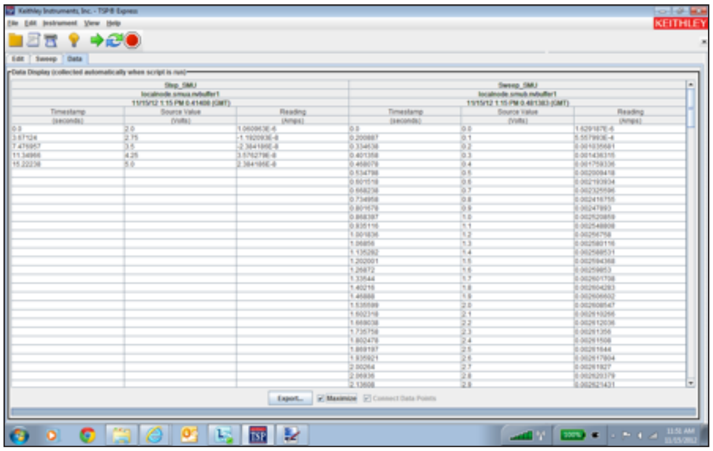

Keithley Instruments, Inc. - TSP® Express
File
Edit
Instrument
View
Help
KEITHLEY
Edit
Sweep
Data
Data Display (collected automatically when script is run)
0.03
0.02
0.01
0
Sweep_SMU Reading (Amps)
1
2
3
4
5
Sweep_SMU Source Value (Volts)
Export...
Maximize
Connect Data Points

Figure 5. Screen capture of MOSFET drain family of curves measured by the two-channel Model 2636B using TSP Express software



Data Display (collected automatically when script is run)

Step_SMU			Sweep_SMU		
localnode.smua.nvbuffer1			localnode.smub.nvbuffer1		
11/15/12 1:15 PM 0.41408 (GMT)			11/15/12 1:15 PM 0.481383 (GMT)		
Timestamp (seconds)	Source Value (Volts)	Reading (Amps)	Timestamp (seconds)	Source Value (Volts)	Reading (Amps)
0.0	2.0	1.060963E-8	0.0	0.0	1.629187E-6
3.67124	2.75	-1.192093E-8	0.200887	0.1	5.557993E-4
7.476957	3.5	-2.384186E-8	0.334638	0.2	0.001035681
11.34966	4.25	3.576279E-8	0.401358	0.3	0.001436315
15.22238	5.0	2.384186E-8	0.468078	0.4	0.001759336
			0.534798	0.5	0.002008418
			0.601518	0.6	0.002193934
			0.668238	0.7	0.002325596
			0.734958	0.8	0.002416755
			0.801678	0.9	0.00247993
			0.868397	1.0	0.002520859
			0.935116	1.1	0.002548808
			1.001836	1.2	0.00256758
			1.06856	1.3	0.002580116
			1.135282	1.4	0.002588531
			1.202001	1.5	0.002594368
			1.26872	1.6	0.00259853
			1.33544	1.7	0.002601708
			1.40216	1.8	0.002604283
			1.46888	1.9	0.002606602
			1.535599	2.0	0.002608547
			1.602318	2.1	0.002610266
			1.669038	2.2	0.002612036
			1.735758	2.3	0.00261356
			1.802478	2.4	0.00261508
			1.869197	2.5	0.00261644
			1.935921	2.6	0.002617804
			2.00264	2.7	0.00261827
			2.06936	2.8	0.002620379
			2.13608	2.9	0.002621431

Figure 6. MOSFET I-V curve data displayed in a table in TSP Express

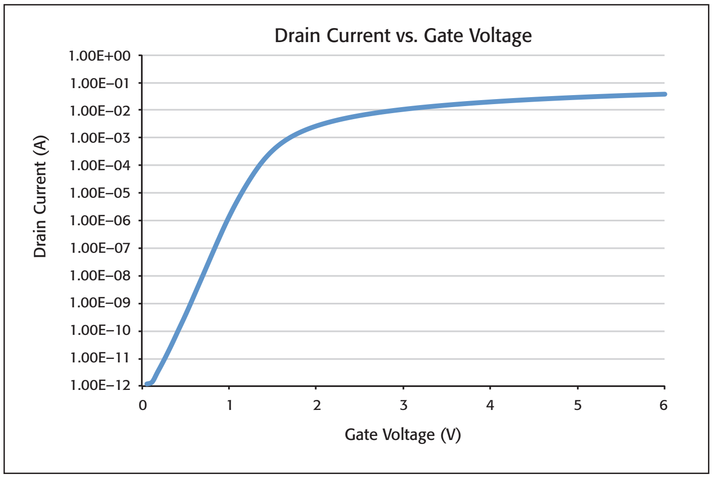


Figure 7. Drain current as a function of the gate voltage of a MOSFET generated by the Model 2636B and the TSP Express software

Without changing connections to the device, the TSP Express software allows users to perform other common I-V FET tests such as the Drain Current (I_D) as a function of Gate Voltage (V_G). For this test, the gate voltage is swept and the resulting drain current is measured at a constant drain voltage. ***Figure 7*** shows the results of an I_D-V_G curve at a constant drain voltage. However, in this case, the generated data was exported to a file and plotted on a semi-log graph. This test can be easily reconfigured to step the drain voltage as the gate voltage is swept.

The I_D-V_G data shows the many decades of drain current that were measured by the Model 2636B, from 1E-12 to 1E-2 amps. The Model 2636B is a low current SMU with 0.1fA resolution, so it's suitable for other low current FET test applications such as gate leakage current measurements. When measuring low current, it is important to employ low level measurement techniques, such as shielding and guarding, to prevent errors. Further information on optimizing low current measurements is available in Keithley's *Low Level Measurements Handbook*.

Conclusion

Parameter testing of FETs is often performed using multiple Source Measurement Units (SMUs). To simplify I-V characterization of FETs, Series 2600B SMUs with embedded TSP Express software allow generating I-V sweeps quickly and easily. With their wide dynamic range of current and voltage sourcing and measuring, Series 2600B SMUs are ideal tools for semiconductor device characterization.

Measuring Pulsed Waveforms with the High Speed Analog-to-Digital Converter in the Model 2651A High Power System SourceMeter® Instrument

Green initiatives and energy efficiency standards worldwide have motivated engineers to find ways to design more efficient semiconductor devices and integrated circuits. In industrial applications, engineers are trying to improve the efficiency of switching power supplies and power inverters. In commercial and residential applications, the push for LEDs (light emitting diodes) drives the design of AC-DC converters to make these DC devices operate on AC power and use pulse width modulation as a light dimming technique.

For these inherently pulsed applications, it is important to test the discrete components that make up the end product under pulsed conditions. Test instruments with only DC capability can deliver an amount of power to a device that causes enough heat dissipation to alter its characteristics. The desire to measure the true state of the device without the effects of self-heating is another motivation for pulsed characterization.

The use of a pulsed stimulus demands faster measurements. The Model 2651A meets this need with its high speed ADCs (analog-to-digital converters). Coupled with the ability to measure asynchronously from the source, this feature makes the Model 2651A suitable for many transient characterization applications. The following demonstrates how to configure the ADCs to perform measurements on pulsed waveforms and considers techniques to obtain optimal results.

High Speed ADC vs. Integrating ADC

Traditional precision SMUs (source-measure units) use integrating ADCs. An integrating ADC averages the signal over a certain time interval known as the integration time. ***Figure 1*** depicts a simplified dual-slope integrating ADC. This type of ADC operates by charging a capacitor with the unknown signal and then discharging the capacitor using a reference voltage. The ratio of the charge and discharge times is proportional to the ratio of the unknown signal to the reference signal.

While having the advantage of high accuracy and excellent noise immunity, this ADC technology does not lend itself to high speed measurements. The charge-discharge cycles on the capacitor result in long inter-measurement intervals. For example, though the smallest integration interval for the Model 2651A is 0.001PLC (16.67µs for 60Hz, 20µs for 50Hz), the smallest inter-measurement interval is 50µs.

In addition to the two integrating ADCs for voltage and current, the Model 2651A also includes two high speed ADCs with the capability of sampling signals at burst rates of up to 1MHz.[1] These ADCs use sampling technology similar to an oscilloscope and take snapshots of the signal over time. A high speed ADC of the Model 2651A has higher resolution (18 bits) than an oscilloscope (typically 8 bits) resulting in more precise transient characterization in comparable bandwidths.

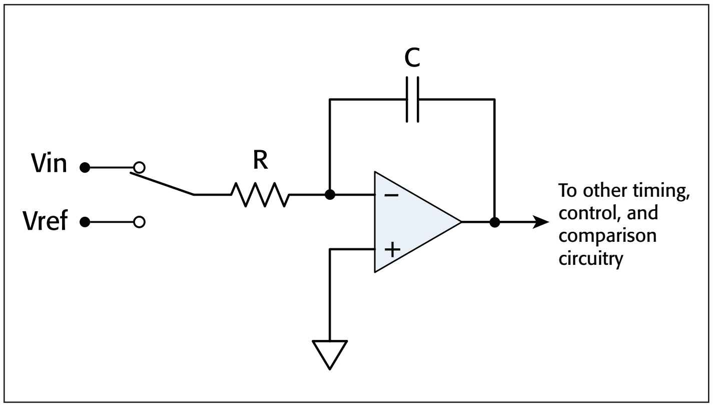


Figure 1. Simplified diagram of dual-slope integrating ADC.

Figure 2 illustrates the difference between the integrating and high speed ADCs. While returning more readings, the measurements performed by the high speed ADC are less accurate and less repeatable than those performed by the integrating ADC. For applications that demand higher throughput, the lower accuracy can be tolerated, or if needed, improved by averaging several readings. Typically, integrating

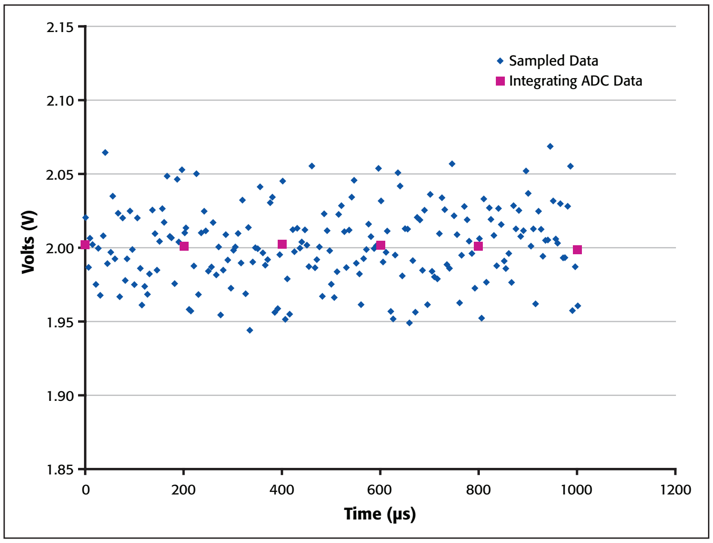


Figure 2. Comparison of possible results from integrating and sampling ADC technologies.

[1] Up to 5,000 readings can be acquired at the maximum acquisition rate.

ADC measurements with integration rates of 0.01PLC or faster can be made with similar accuracy using the high speed ADC.

Having two high speed ADCs ensures that voltage and current measurement can be made simultaneously. The ability to sample current is a unique feature of the Model 2651A and may replace the need for a current probe and an oscilloscope in some applications.

The combination of the Model 2651A's high speed ADCs and trigger model supports precisely timed measurements on pulsed signals. Additionally, the Model 2651A introduces a feature that allows the user to trigger measurements asynchronously from source operations, such as before, during, or after a pulse. This capability can also be used with the integrating ADCs. ***Figure 3*** diagrams five examples of pulsed signals and measurements that can be made with the Model 2651A. The sections below discuss how to configure the Model 2651A to perform each of these examples.

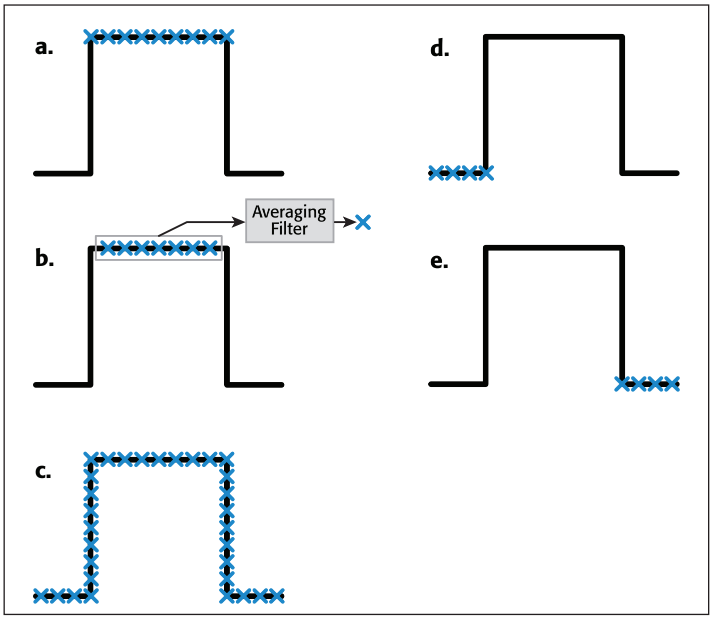


Figure 3a. Example #1, measuring at the top of the pulse.

Figure 3b. Example #2, performing a spot mean measurement at the top of the pulse.

Figure 3c. Example #3, digitizing the entire pulse.

Figure 3d. Example #4, triggering measurements to begin before the pulse.

Figure 3e. Example #5, triggering measurement to begin after the pulse.

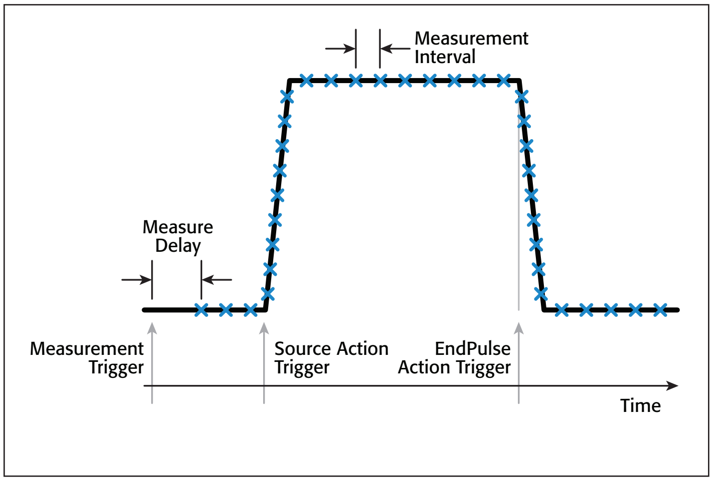


Figure 4. Specifying a pulse using the Model 2651A.

Figure 4 illustrates the relationship between a pulse, the triggering conditions that create the pulse, and the definition of high speed ADC measurement parameters.

General Information Regarding the Examples

In the sections that follow, each of the examples makes use of a few common functions. The code for these functions is in ***Appendix A***.

Example #1: Digitizing the Top of a Pulse

Potential Uses

For some applications, such as thermal impedance of power diodes and LEDS, characterizing the slope of the measured voltage at the top of the pulse is important. This capability is also useful for characterizing pulse amplitude flatness. The high speed ADCs can digitize the top of the pulse when the measurements are made synchronously with the source.

How the Measurement is Made

The pulses are timed using trigger timers. Measurements are triggered at the beginning of the pulse, but delayed to the settled part of the pulse by programming a measure delay.

The trigger model setup is depicted in ***Figure 5***. The Test Script Processor (TSP®) script for Example #1 is located in ***Appendix B*** along with the commands for executing the test and obtaining the results. The sample results taken using a 0.1Ω resistor are shown in ***Figure 6***.

Example #2: Performing a Spot Mean Measurement at the Top of the Pulse

Potential Uses

Often, analysis software is used to average sampled data to improve accuracy. The Model 2651A can automatically perform averages on measurements.

How the Measurement is Made

The averaging and median filters of the Model 2651A can be used on the high speed ADC readings, making it possible to return spot mean measurements. The same test performed in Example #1 can be modified to return a spot mean measurement instead of the raw sample data by changing a few lines of code.

The trigger model configuration is the same as used for Example #1. ***Appendix C*** contains the TSP script for Example #2 and the commands for executing the test and obtaining the results. The test results are shown in ***Figure 6*** next to the raw sample data and are also listed in ***Table 1***.

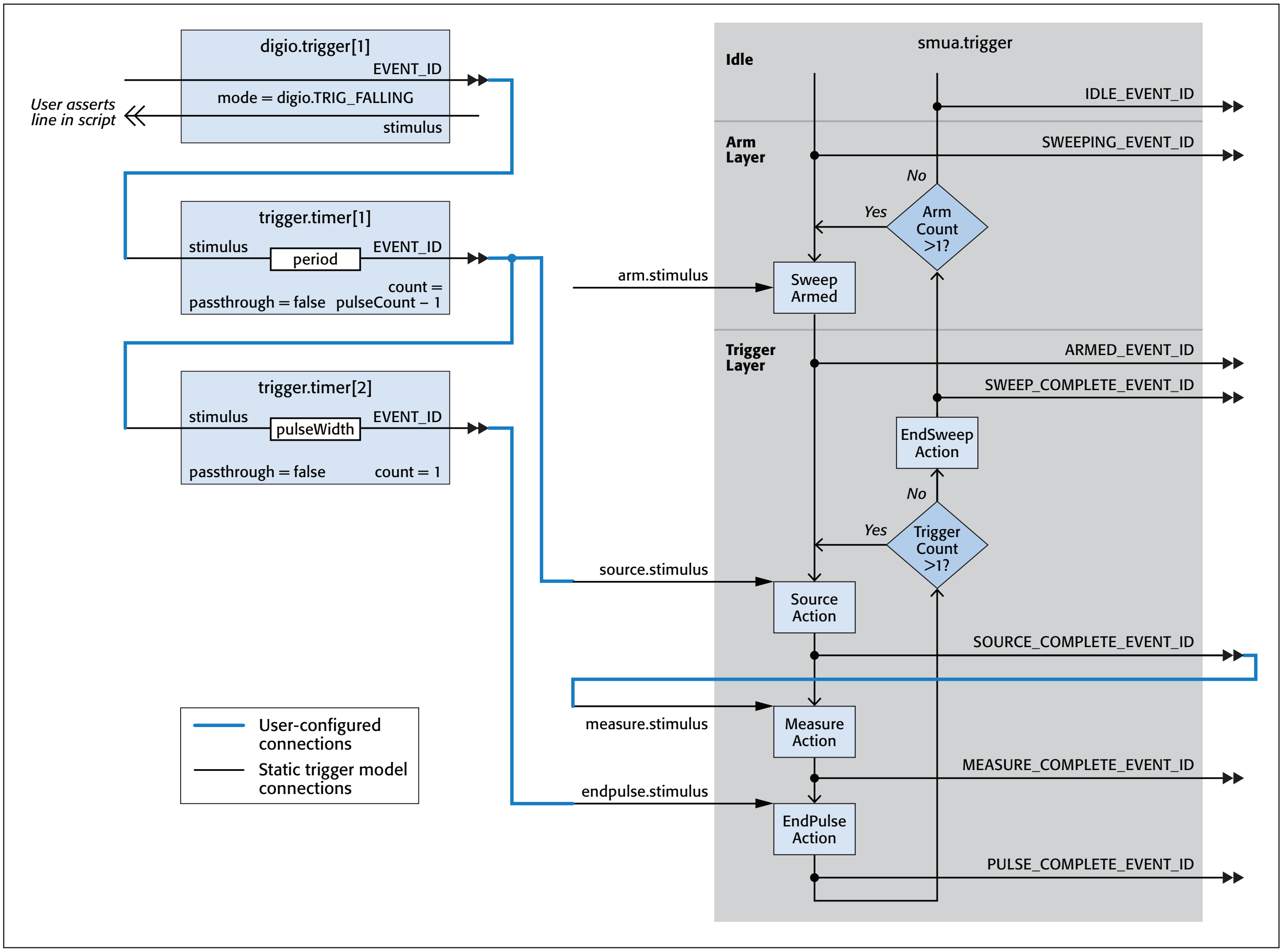


Figure 5. Trigger model configuration for Examples #1 and #2.

Table 1. Spot mean measurement results for Example #2.

Current (A)	Voltage (V)
19.7816	1.97854
19.8421	1.98492

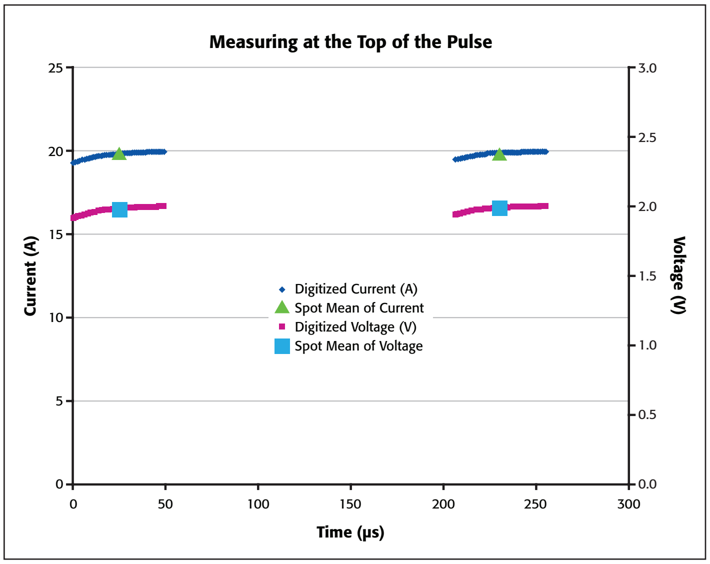


Figure 6. Results of Examples #1 and #2. The load is a 0.1Ω resistor.

Example #3: Digitizing the Entire Pulse Including the Rising and Falling Edges

Potential Uses

At times, it is useful to characterize how a pulse is transmitted through a device or system. These applications require that the entire pulse be digitized, including the rising and falling edges. This measurement is possible using the high speed ADCs to measure asynchronously to the source operation.

How the Measurement is Made

Timers are again used to trigger the start and end of the pulse. The trigger used to start the pulse is also used to start the measurement process.

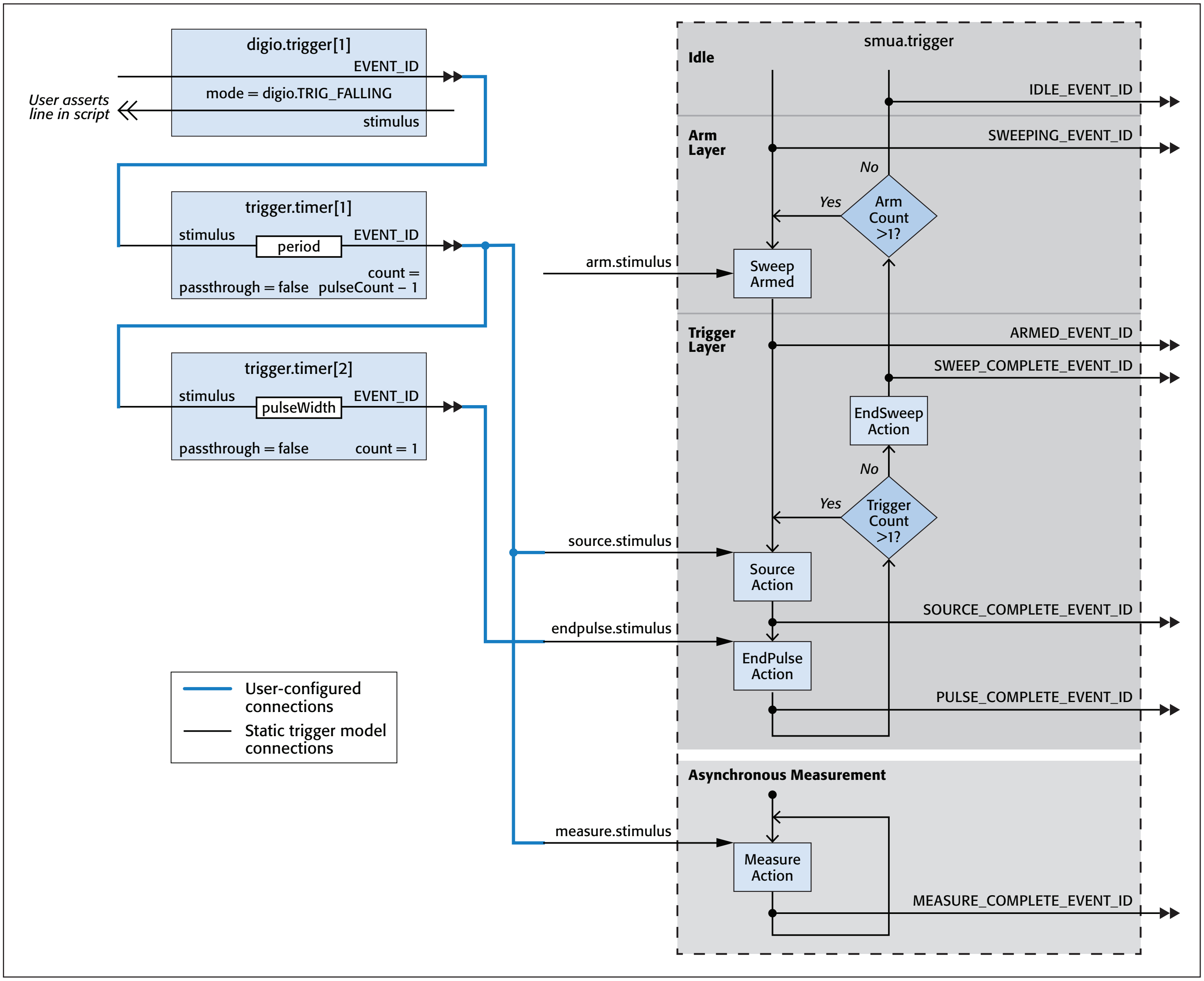


Figure 7. Trigger model configuration for Example #3.

Figure 7 diagrams the trigger model configuration. ***Appendix D*** contains the TSP script for Example #3 and the commands for executing the test and obtaining the results. The results for the example data are shown in ***Figure 8***.

Example #4: Pre-Pulse Characterization: Triggering Measurements before the Pulse

Potential Uses

Pulses are sometimes used to provide power stresses to the device. It is useful to note the device state before the stress is applied. This can be done by programming a pulse with a non-zero idle level and triggering the measurements before the pulse.

How the Measurement is Made

This example is arranged so that the user can specify how long before the pulse the measurements should occur. Timers are

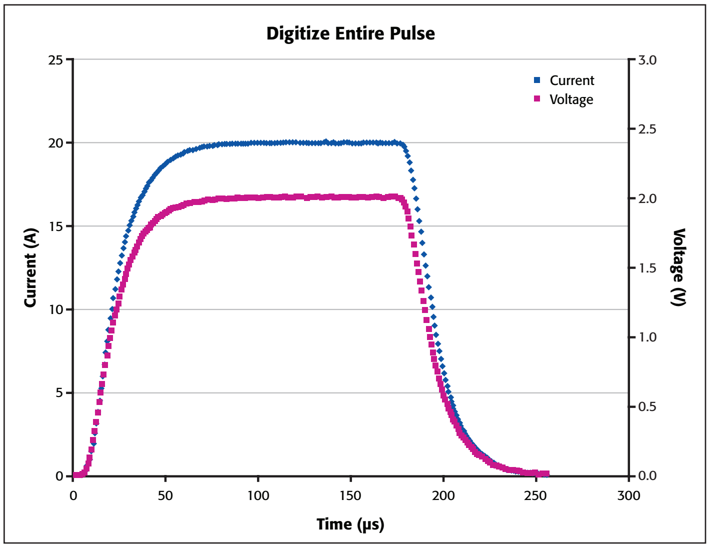


Figure 8. Results from Example #3 showing the voltage and current measurements as performed by the high speed ADCs. There is a 0.1Ω resistive load.

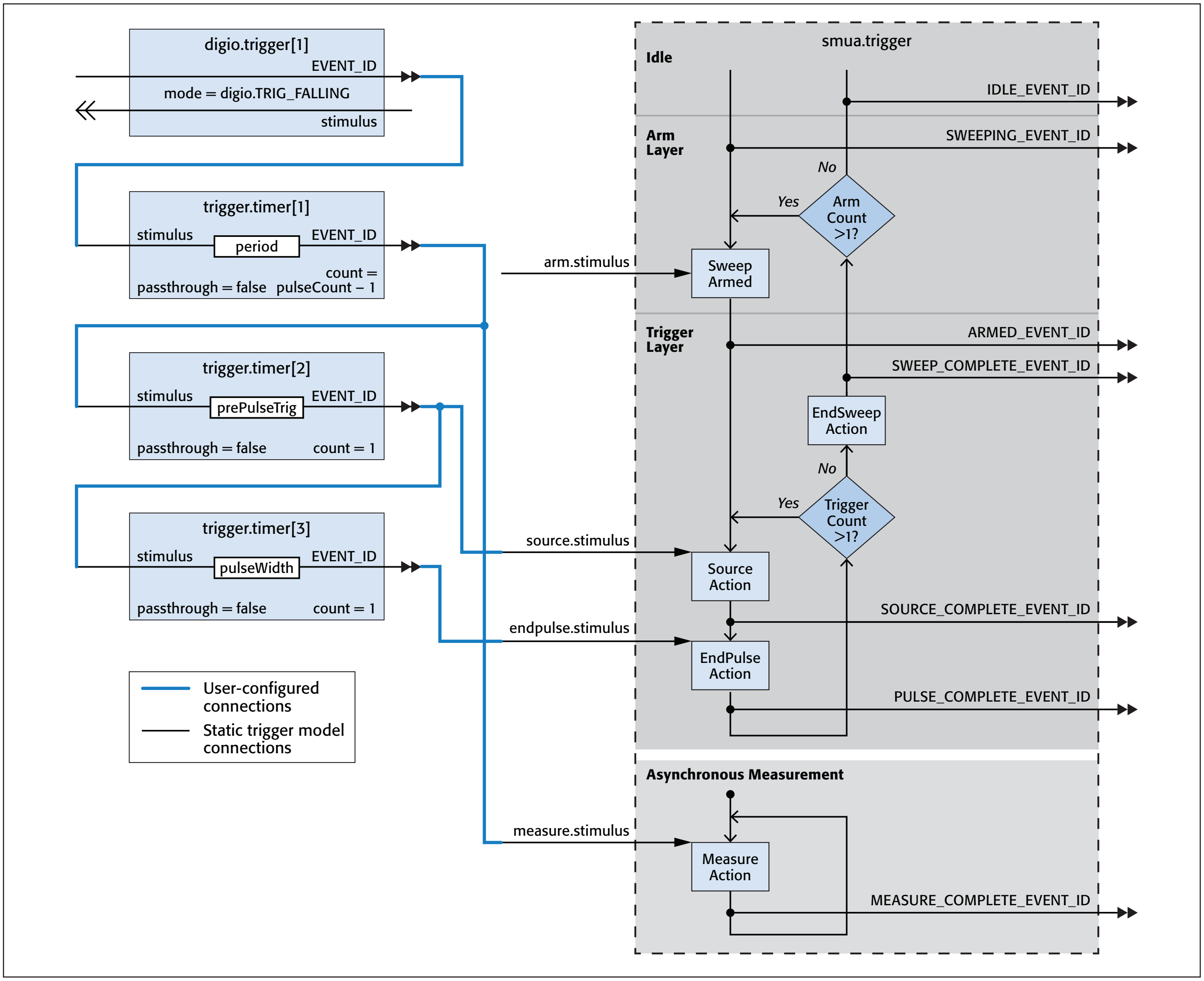


Figure 9. Trigger model configuration for Example #4.

used to program the start of the measurement and the beginning and end of the pulse.

The trigger model configuration for this example is shown in ***Figure 9***. ***Appendix E*** contains the TSP script of Example #4 and the commands for executing the test and obtaining the results. The results for the example code are shown in ***Figure 10***.

Example #5: Post-Pulse Characterization: Triggering Measurements after the Pulse

Potential Uses

When using pulse testing to stress a device, the device must also be characterized after the stress is applied. This is typically done by sourcing a pre-defined test voltage or current after the pulse. The test level is chosen so as not to cause any additional thermal or electrical stress to the device. The measurement can be made by sourcing a pulse with a non-zero idle level and using the high speed ADCs to perform the measurement. The results

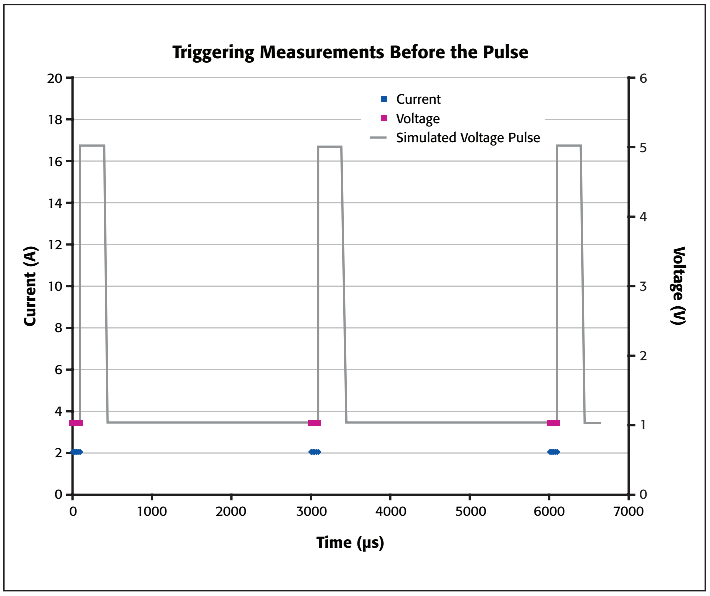


Figure 10. Results from Example #4. There is a 0.5Ω resistive load.

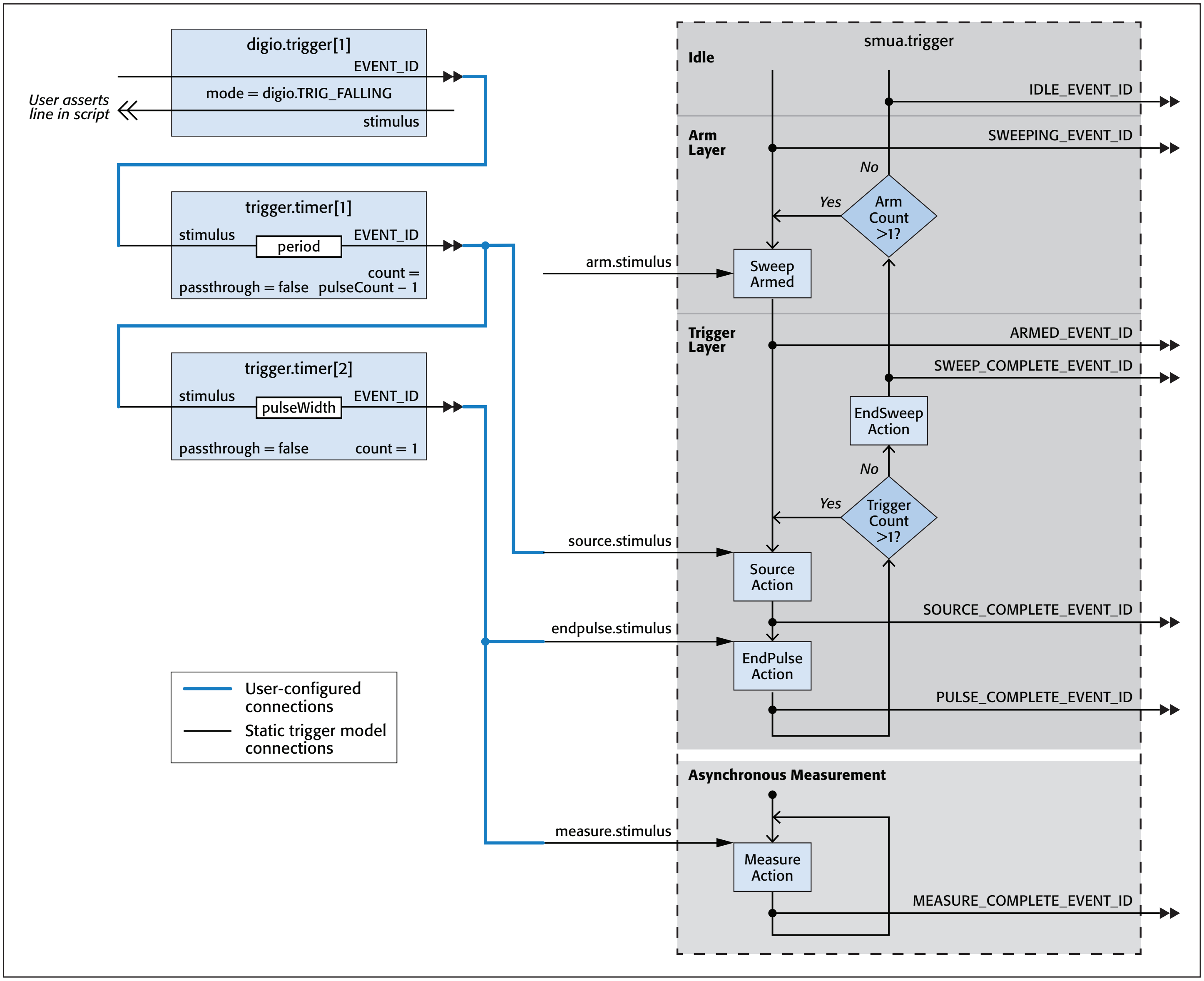


Figure 11. Trigger model configuration for Example #5.

from the high speed ADCs indicate how the device recovers from the stress.

How the Measurement is Made

The pulses are timed using trigger timers. The measurement is triggered by the EndPulse event of the trigger model, which causes the pulse to return to the idle level. If desired, a measurement delay can be used to postpone the start of the measurements until after the falling edge occurs.

Figure 11 diagrams the trigger model configuration for this example. *Appendix F* contains the TSP script of Example #5 and the commands for executing the test and obtaining the results. The results for the example code are shown in *Figure 12*.

Getting Pulses and Measurements on Time

The ability to pulse in pulse-only current and voltage regions coupled with the ability to make measurements with the high

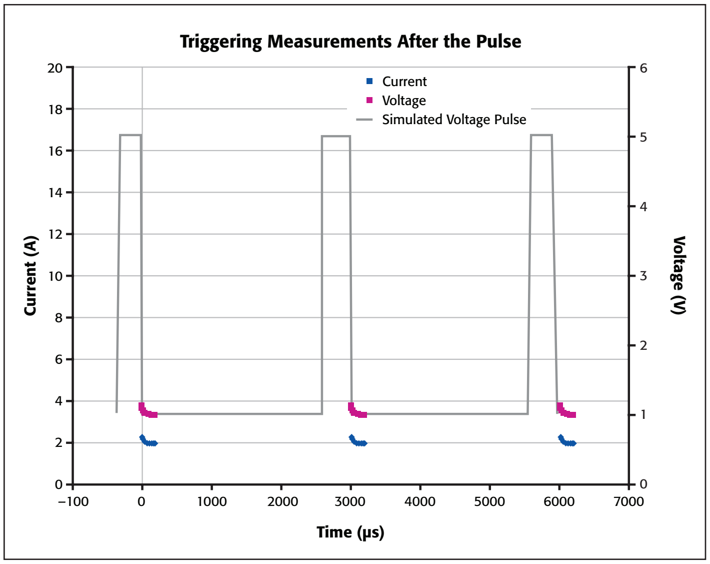


Figure 12. Results from Example #5. There is a 0.5Ω resistive load.

speed ADCs means that the user must carefully consider test timing. In the Model 2651A, the taking of measurements is given priority over source and display operations. Therefore, display updates may not occur and source timing can be compromised if the system is busy making or processing measurements. Consider the following precautions in order to avoid erratic pulse timing. These precautions can also help to prevent damaging devices from the excessive power that can be generated from the Model 2651A's ability to output very high currents in pulse and DC regions.

- For precise timing, always use fixed ranges for source and measurement functions. (Note: Asynchronous measurements mandate the use of fixed ranging for both the integrating and high speed ADCs.)
- When operating in the extended pulse-only operation region, obey all maximum duty cycle and pulse width limitations. The instrument will turn off the output to prevent thermal runaway. However, continuing to issue triggers for source or EndPulse actions may result in source and/or EndPulse action overruns.
- Sustained high speed data acquisition rates may result in undesired pulse timing. This is more likely when the measure interval for the high speed ADC is smaller than 10µs and when a pulse train has a high duty cycle. If a particular configuration results in undesired pulse timing, then abort the present test and perform one of the following to obtain the desired timing: reduce the measurement trigger frequency, reduce the measurement count, or increase the measurement interval.
- In asynchronous operation, all expected measurements must be triggered before the End Sweep action occurs. Refer to Example #5 for an example of how to use a measure delay to postpone the start of measurements after the trigger.
- Any source polarity changes during a sweep incur a 100µs delay before the source level is set. The number 0 is considered a positive value.
  - For negative-going pulses that start from zero, use a 'negative zero', which is a negative number that is very near zero, e.g., -1e-12. See the code for Examples #1, #2, and #3.
  - When the source polarity must be changed during a sweep, the user must account for the polarity change delay in the TSP script in order to obtain proper pulse timing.

A user can monitor the state of the instrument using its status model. Action overruns set bits in different registers of the status model according to the trigger object in which the action overrun is generated. (Each time a register is read, all bits are cleared.) For the SMU trigger object, an action overrun occurs when a new input trigger is detected before the previously triggered action has been started[2]. ***Figure 13*** shows the SMU trigger overrun operation status register set. Note that each action block (Arm, Source, Measure, and EndPulse) has a corresponding overrun bit in the status model. ***Appendix G*** contains the TSP script demonstrating how to monitor the Model 2651A's operation overrun Event register for overruns to the Source, Measure, or EndPulse action blocks.

Each of the SMU action blocks latches (or remembers) one trigger, even if it cannot immediately act upon that trigger. An action overrun is not generated by the latched trigger. However, if multiple triggers are issued while an action is in process, then overruns are generated. For instance, if the Model 2651A is busy making a large number of high speed measurements, it may not be able to respond to the EndPulse event. The first time the EndPulse event is triggered, the SMU will hold off ending the pulse and the result is a pulse with a longer width than expected. If the SMU is still making or processing measurements when the next EndPulse event is triggered, an EndPulse action overrun is generated and the next pulse does not end.

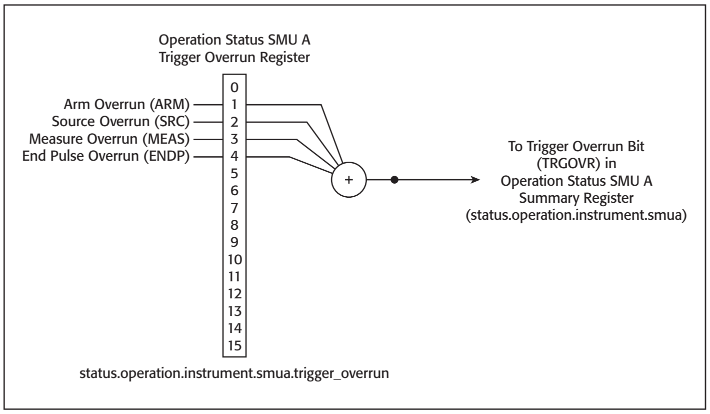


Figure 13. SMU trigger overrun operation status register set.

Conclusion

The high speed ADCs in Keithley's Model 2651A provide a more detailed look at measurements of pulsed waveforms. They can be used for a variety of transient characterization applications previously not possible with an SMU.

[2] For asynchronous measurement mode, a measurement action overrun is generated each time a new measurement is triggered while the SMU is taking measurements. No triggers are latched for asynchronous measurements.

Appendix A. Common Functions, runPulse() and printData()

The examples in the following appendices use common functions. The TSP scripts for these functions are listed below. These lines of code may be copied, saved, and run as a script so that the functions are available.

```
--Runs the pulse after it is configured
function runPulse()
        smua.source.output = 1
        delay(0.001)
        smua.trigger.initiate()
        digio.trigger[1].assert()
        waitcomplete()
        smua.source.output = 0

end
--Prints the results from the reading buffers.
function printData()
        if smua.nvbuffer1.n == 0 then
                print("No readings in buffer")
        else
                print("Timestamps\tCurrent\tVoltage")
                for i = 1, smua.nvbuffer1.n do
print(string.format("%g\t%g\t%g", smua.nvbuffer1.timestamps[i], smua.nvbuffer1.readings[i], smua.nvbuffer2.
readings[i]))
                end
        end
end
```

Appendix B. Code for Example #1, Digitizing the Top of a Pulse

This is the TSP script that is used to configure the Model 2651A SourceMeter instrument to digitize the top of the pulse when the measurements are made synchronously with the source. The commands for executing the code and obtaining the results are listed below the TSP script.

```
function digitizeTopSync(iAmpl, vLimit, pulseWidth, period, pulseCount, sampleInterval, measCount, measDelay)
        reset()
        errorqueue.clear()
        status.reset()

        local l_iZero --Variable for the idle level of the sweep

        --[[
        The ‘if’ statement below controls the idle level of the sweep. Set to 0 for positive pulses. Set to
‘negative zero’ for
        negative pulses. Eliminates polarity change delay. See “Getting Pulses and Measurements On Time” for more
information.
        ]]
        if iAmpl <0 then
                local l_iZero = -1e-13
        else
                local l_iZero = 0
        end

        -- Prepare the reading buffers
        smua.nvbuffer1.clear()
        smua.nvbuffer1.appendmode = 0
        smua.nvbuffer1.collecttimestamps = 1
        smua.nvbuffer1.collectsourcevalues = 0
        smua.nvbuffer2.clear()
        smua.nvbuffer2.appendmode = 0
        smua.nvbuffer2.collecttimestamps = 1
        smua.nvbuffer2.collectsourcevalues = 0

        --Set up the source
        smua.source.func = smua.OUTPUT_DCAMPS
        smua.sense = smua.SENSE_REMOTE --Enables remote sense (4W measurements)
        smua.source.rangei = iAmpl
        smua.source.leveli = l_iZero --Sets the pulse off value.
        smua.source.limitv = vLimit -- Sets the DC voltage limit
        smua.trigger.source.listi({iAmpl}) --Sets the values for the source sweep
        smua.trigger.source.limitv = vLimit --Sets the voltage limit
        smua.trigger.source.action = smua.ENABLE --Enables the source sweep

        --Set up the measurements
        smua.measure.rangev = vLimit -- Set the measure voltage range
        smua.measure.adc = smua.ADC_FAST --Configures the SMU to use the high speed ADC
        smua.measure.interval = sampleInterval --Sets the measurement interval
        smua.measure.delay = measDelay --Sets the initial measurement delay
        smua.measure.count = measCount --Sets the measurement count
        smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2) -- Configure the SMU to measure both voltage and
current
        smua.trigger.measure.action = smua.ENABLE -- Enables measurements in sync with the source sweep

        --Set up a digital I/O line to trigger the start of the period timer
        digio.trigger[1].mode = digio.TRIG_FALLING
        digio.trigger[1].clear()

        -- Timer 1 controls the pulse period by triggering the pulse to begin
        trigger.timer[1].delay = period
        --The timer should always have a count of 1. The ‘if’ statement below configures the timer count.
        if pulseCount > 1 then
                trigger.timer[1].count = (pulseCount - 1)
        else
                trigger.timer[1].count = 1
        end
        trigger.timer[1].passthrough = true --Timer issues an event at the start of the first interval.
        trigger.timer[1].stimulus = digio.trigger[1].EVENT_ID -- Period timer is triggered by a digital I/O line
        trigger.timer[1].clear()

        -- Timer 2 controls the pulse width
        trigger.timer[2].delay = pulseWidth
        trigger.timer[2].count = 1
        trigger.timer[2].passthrough = false --Pulse width timer only issues event when interval elapses
        trigger.timer[2].stimulus = trigger.timer[1].EVENT_ID --Timer starts when period timer issues event
        trigger.timer[2].clear()

        -- Return to the bias level at the end of the pulse/sweep
        smua.trigger.endpulse.action = smua.SOURCE_IDLE
        smua.trigger.endsweep.action = smua.SOURCE_IDLE
        smua.trigger.count = pulseCount --Sets the number of pulses to generate
        smua.trigger.arm.count = 1 --Sets the number of times to iterate through the entire sweep
        smua.trigger.arm.stimulus = 0
        smua.trigger.source.stimulus = trigger.timer[1].EVENT_ID --Source starts when period timer generates event
        smua.trigger.measure.stimulus = smua.trigger.SOURCE_COMPLETE_EVENT_ID -- Start measuring when the source
action is complete
        smua.trigger.endpulse.stimulus = trigger.timer[2].EVENT_ID -- Start EndPulse action as soon as the pulse
width timer ends

end
```

Executing the Code

Send the following lines to configure and execute two 20A, 100µs pulses with a 200µs period and a voltage limit of 10V. Fifty measurements are requested, taken at 1µs intervals with an initial measurement delay of 50µs.

```
digitizeTopSync(20, 10, 100e-6, 200e-6, 2, 1e-6, 50, 50e-6)
runPulse()
```

Results

The raw sample data is stored in the dedicated nonvolatile buffers, smua.nvbuffer1 and smua.nvbuffer2. Use the following command to print the data through the remote command interface:

```
printData()
```

Sample results taken using a resistor are shown in ***Figure 6***.

Appendix C. Code for Example #2, Performing a Spot Mean Measurement at the Top of the Pulse

Below is the example code to configure the Model 2651A to perform a spot mean measurement at the top of the pulse. Highlighting is used to distinguish the command lines that are different from those in Example #1 (Appendix B). The commands for executing the code and obtaining the results are listed below the TSP script.

```
function spotMeanTopSync(iAmpl, vLimit, pulseWidth, period, pulseCount, sampleInterval, filtCount, measDelay)
        reset()
        errorqueue.clear()
        status.reset()

        local l_iZero --Variable for the idle level of the sweep

        --[[
        The 'if' statement below controls the idle level of the sweep. Set to 0 for positive pulses. Set to
'negative zero' for
        negative pulses. Eliminates polarity change delay. See "Getting Pulses and Measurements On Time" for more
information.
        ]]
        if iAmpl <0 then
                local l_iZero = -1e-13
        else
                local l_iZero = 0
        end

        -- Prepare the reading buffers
        smua.nvbuffer1.clear()
        smua.nvbuffer1.appendmode = 0
        smua.nvbuffer1.collecttimestamps = 1
        smua.nvbuffer1.collectsourcevalues = 0
        smua.nvbuffer2.clear()
        smua.nvbuffer2.appendmode = 0
        smua.nvbuffer2.collecttimestamps = 1
        smua.nvbuffer2.collectsourcevalues = 0

        --Set up the source
        smua.source.func = smua.OUTPUT_DCAMPS
        smua.sense = smua.SENSE_REMOTE --Enables remote sense (4W measurements)
        smua.source.rangei = iAmpl
        smua.source.leveli = l_iZero --Sets the pulse off value.
        smua.source.limitv = vLimit -- Sets the DC voltage limit
        smua.trigger.source.listi({iAmpl}) --Sets the values for the source sweep
        smua.trigger.source.limitv = vLimit --Sets the voltage limit
        smua.trigger.source.action = smua.ENABLE --Enables the source sweep

        --Set up the measurements
        smua.measure.rangev = vLimit -- Set the measure voltage range
        smua.measure.adc = smua.ADC_FAST --Configures the SMU to use the high speed ADC
        smua.measure.interval = sampleInterval --Sets the measurement interval
        smua.measure.delay = measDelay --Sets the initial measurement delay
        smua.measure.count = 1 --Sets the number of measurements to take each time the measure action block is
triggered
        smua.measure.filter.type = smua.FILTER_REPEAT_AVG
        smua.measure.filter.count = filtCount
        smua.measure.filter.enable = smua.FILTER_ON
        smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2) -- Configure the SMU to measure both voltage and
current
        smua.trigger.measure.action = smua.ENABLE --Enables measurements in sync with the source sweep

        --Set up a digital I/O line to trigger the start of the period timer
        digio.trigger[1].mode = digio.TRIG_FALLING
        digio.trigger[1].clear()

        -- Timer 1 controls the pulse period by triggering the pulse to begin
        trigger.timer[1].delay = period
        --The timer should always have a count of 1. The 'if' statement below configures the timer count.
        if pulseCount > 1 then
                trigger.timer[1].count = (pulseCount - 1)
        else
                trigger.timer[1].count = 1
        end
        trigger.timer[1].passthrough = true --Timer issues an event at the start of the first interval.
        trigger.timer[1].stimulus = digio.trigger[1].EVENT_ID -- Period timer is triggered by a digital I/O line
        trigger.timer[1].clear()

        -- Timer 2 controls the pulse width
        trigger.timer[2].delay = pulseWidth
        trigger.timer[2].count = 1
        trigger.timer[2].passthrough = false --Pulse width timer only issues event when interval elapses
        trigger.timer[2].stimulus = trigger.timer[1].EVENT_ID --Timer starts when period timer issues event
        trigger.timer[2].clear()

        -- Return to the bias level at the end of the pulse/sweep
        smua.trigger.endpulse.action = smua.SOURCE_IDLE
        smua.trigger.endsweep.action = smua.SOURCE_IDLE
        smua.trigger.count = pulseCount --Sets the number of pulses to generate
        smua.trigger.arm.count = 1 --Sets the number of times to iterate through the entire sweep
        smua.trigger.arm.stimulus = 0
        smua.trigger.source.stimulus = trigger.timer[1].EVENT_ID --Source starts when period timer generates event
        smua.trigger.measure.stimulus = smua.trigger.SOURCE_COMPLETE_EVENT_ID -- Start measuring when the source
action is complete
        smua.trigger.endpulse.stimulus = trigger.timer[2].EVENT_ID -- Start EndPulse action as soon as the pulse
width timer ends

end
```

Executing the Code

Send the following lines to configure and execute two 20A, 100μs pulses with a 200μs period and a voltage limit of 10. Two measurements are requested, each with an average of 50 readings taken at 1μs intervals with an initial measurement delay of 50μs.

```
spotMeanTopSync(20, 10, 100e-6, 200e-6, 2, 1e-6, 50, 50e-6)
runPulse()
```

Results

The raw sample data is stored in the dedicated nonvolatile buffers, smua.nvbuffer1 and smua.nvbuffer2. Use the following command to print the data through the remote command interface:

```
printData()
```

The results are shown in ***Figure 6*** next to the raw sample data and are also listed in ***Table 1***.

Appendix D. Code for Example #3, Digitizing the Entire Pulse Including the Rising and Falling Edges

The example code below lists the commands necessary to digitize one to two pulses at the maximum sample rate. (Note: To increase the pulse count, the sample interval or pulse period must be increased to allow sufficient time for data processing and to avoid erratic source timing. For more details, see “Getting Pulses and Measurements on Time” in this application note.) The commands for executing the code and obtaining the results are listed after the TSP script.

```
function digitizePulse(iAmpl, vLimit, pulseWidth, period, pulseCount, sampleInterval)
        reset()
        errorqueue.clear()
        status.reset()

        --[[
        The ‘if’ statement below controls the idle level of the sweep. Set to 0 for positive pulses. Set to 
‘negative zero’ for
        negative pulses. Eliminates polarity change delay. See “Getting Pulses and Measurements On Time” for more 
information.
        ]]
        local l_iZero

        if iAmpl <0 then
                l_iZero = -1e-13
        else
                l_iZero = 0
        end

-- Prepare the reading buffers
        smua.nvbuffer1.clear()
        smua.nvbuffer1.appendmode = 0
        smua.nvbuffer1.collecttimestamps = 1
        smua.nvbuffer1.collectsourcevalues = 0
        smua.nvbuffer2.clear()
        smua.nvbuffer2.appendmode = 0
        smua.nvbuffer2.collecttimestamps = 1
        smua.nvbuffer2.collectsourcevalues = 0

        --Set up the source
        smua.source.func = smua.OUTPUT_DCAMPS
        smua.sense = smua.SENSE_REMOTE --Enable remote sense (4W measurements)
        smua.source.rangei = iAmpl
        smua.source.leveli = l_iZero --Sets the pulse off value.
        smua.source.limitv = vLimit -- Sets the DC voltage limit
        smua.trigger.source.listi({iAmpl}) -- Sets the amplitude values for the pulse sweep
        smua.trigger.source.limitv = vLimit --Sets the voltage limit
        smua.trigger.source.action = smua.ENABLE --Enables the source sweep

        --Set up the measurements
        smua.measure.rangev = vLimit -- Set the measure voltage range
        smua.measure.adc = smua.ADC_FAST -- Configures SMU to use high speed aDC
        smua.measure.interval = sampleInterval --Set measurement interval
        smua.measure.delay = 0 --Configures no initial measurement delay
        -- Set the measure count to be 50% greater than the width of the pulse to ensure we capture rising and 
falling edges of pulse
        smua.measure.count = pulseWidth / smua.measure.interval * 1.5
        smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2) -- Configure the SMU to measure both voltage and 
current
        smua.trigger.measure.action = smua.ASYNC -- Configure SMU measurements to occur asynchrnously with source 
sweep

        --Set up a digital I/O line to trigger the start of the period timer
        digio.trigger[1].mode = digio.TRIG_FALLING
        digio.trigger[1].clear()

        -- Timer 1 controls the pulse period by triggering the pulse to begin
        trigger.timer[1].delay = period

        --The timer should always have a count of 1. The ‘if’ statement below configures the timer count.
        if pulseCount > 1 then
                trigger.timer[1].count = (pulseCount - 1)
        else
                trigger.timer[1].count = 1
        end
        trigger.timer[1].passthrough = true --Timer issues an event at the start of the first interval.
        trigger.timer[1].stimulus = digio.trigger[1].EVENT_ID -- Period timer is triggered by a digital I/O line
        trigger.timer[1].clear()

        -- Timer 2 controls the pulse width
        trigger.timer[2].delay = pulseWidth
        trigger.timer[2].count = 1
        trigger.timer[2].passthrough = false --Pulse width timer only issues event when interval elapses
        trigger.timer[2].stimulus = trigger.timer[1].EVENT_ID --Timer starts when period timer issue event
        trigger.timer[2].clear()

        -- Return to the bias level at the end of the pulse/sweep
        smua.trigger.endpulse.action = smua.SOURCE_IDLE
        smua.trigger.endsweep.action = smua.SOURCE_IDLE
        smua.trigger.count = pulseCount -- Configures number of pulses to complete
        smua.trigger.arm.count = 1
        smua.trigger.arm.stimulus = 0
        smua.trigger.source.stimulus = trigger.timer[1].EVENT_ID -- Trigger source sweep using events generated by
period timer
        smua.trigger.measure.stimulus = trigger.timer[1].EVENT_ID -- Start measuring when the period timer starts
the source action
        smua.trigger.endpulse.stimulus = trigger.timer[2].EVENT_ID -- Start EndPulse action as soon as the pulse
width timer ends

end
```

Executing the Code

Send the following lines to configure and execute a single 20A, 200µs pulse with a voltage limit of 10V and take high speed ADC measurements at 1µs intervals.

```
digitizePulse(20, 10, 200e-6, 2e-3, 1, 1e-6)
runPulse()
```

Results

The raw sample data is stored in the dedicated nonvolatile buffers, smua.nvbuffer1 and smua.nvbuffer2. Use the following command to print the data through the remote command interface:

```
printData()
```

The results for the example data are shown in ***Figure 8***. The load is a 0.1Ω resistor.

Appendix E. Code for Example #4, Pre-Pulse Characterization: Triggering Measurements before the Pulse

Below is the TSP script for Example #4. The commands for executing the code and obtaining the results are listed after the TSP script.

```
function measThenPulse(vPulse, vIdle, iLimit, pulseWidth, period, pulseCount, prePulseTrig, measCount,
sampleInterval)
        reset()
        errorqueue.clear()
        status.reset()

        --If the programmed pulse is negative and biased from zero, then program negative zero
        if vIdle == 0 and vPulse <0 then
                vIdle = -1e-12
        else
                vIdle = vIdle
        end


        -- Prepare the reading buffers
        smua.nvbuffer1.clear()
        smua.nvbuffer1.appendmode = 0
        smua.nvbuffer1.collecttimestamps = 1
        smua.nvbuffer1.collectsourcevalues = 0
        smua.nvbuffer2.clear()
        smua.nvbuffer2.appendmode = 0
        smua.nvbuffer2.collecttimestamps = 1
        smua.nvbuffer2.collectsourcevalues = 0

        --Set up the source
        smua.source.func = smua.OUTPUT_DCVOLTS --Set source function to DC Volts
        smua.sense = smua.SENSE_REMOTE -- Enable remote sense (4W measurements)
        smua.source.rangev = vPulse
        smua.source.levelv = vIdle --Sets the pulse off value.
        --The 'if' statement below sets the DC current limit
        if iLimit > 20 then
                smua.source.limiti = 20
        else
                smua.source.limiti = iLimit
        end
        smua.trigger.source.listv({vPulse}) -- Configures amplitude levels for source sweep
        smua.trigger.source.limiti = iLimit --Sets current limit during pulsing
        smua.trigger.source.action = smua.ENABLE --Enables source sweep

        --Set up the measurements
        smua.measure.rangei = iLimit -- Set the measure current range
        smua.measure.adc = smua.ADC_FAST -- Configures SMU to use high speed ADC
        smua.measure.interval = sampleInterval -- Sets measurement interval
        smua.measure.delay = 0
        smua.measure.count = measCount
        smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2) -- Configure the SMU to measure both voltage and
current
        smua.trigger.measure.action = smua.ASYNC -- Configures measurements to occur asynchrnously from source sweep

        --Set up a digital I/O line to trigger the start of the period timer
        digio.trigger[1].mode = digio.TRIG_FALLING
        digio.trigger[1].clear()


        -- Timer 1 controls the start of the measurement and serves as system clock
        trigger.timer[1].delay = period
        --Timer must always have a count of 1 or more. The following 'if' statement sets the timer count
appropriately.
        if pulseCount > 1 then
                trigger.timer[1].count = (pulseCount - 1)
        else
                trigger.timer[1].count = 1
        end

        trigger.timer[1].passthrough = true --Timer issues an event at the start of the first interval.
        trigger.timer[1].stimulus = digio.trigger[1].EVENT_ID -- Period timer is triggered by a digital I/O line
        trigger.timer[1].clear()
        -- Timer 2 controls the delay from the start of measurement to the start of the pulse
        trigger.timer[2].delay = prePulseTrig
        trigger.timer[2].count = 1
        trigger.timer[2].passthrough = false --Only issue event when timer interval has elapsed
        trigger.timer[2].stimulus = trigger.timer[1].EVENT_ID
        trigger.timer[2].clear()

        --Timer 3 controls the pulse width
        trigger.timer[3].delay = pulseWidth
        trigger.timer[3].count = 1
        trigger.timer[3].passthrough = false -- Only issue event when timer interval has elapsed
        trigger.timer[3].stimulus = trigger.timer[2].EVENT_ID
        trigger.timer[3].clear()


        -- Return to the bias level at the end of the pulse/sweep
        smua.trigger.endpulse.action = smua.SOURCE_IDLE
        smua.trigger.endsweep.action = smua.SOURCE_IDLE
        smua.trigger.count = pulseCount --Sets number of pulses to perform
        smua.trigger.arm.count = 1
        smua.trigger.arm.stimulus = 0
        smua.trigger.source.stimulus = trigger.timer[2].EVENT_ID -- Start next pulse once prePulseTrigger time
interval has elapsed
        smua.trigger.measure.stimulus = trigger.timer[1].EVENT_ID -- Start measuring when the period timer issues an
event
        smua.trigger.endpulse.stimulus = trigger.timer[3].EVENT_ID -- Start EndPulse action as soon as the pulse
width timer ends

end
```

Executing the Code

Send the following lines to configure and execute three pulses which pulse from 1 to 5V with a width of 300µs and a period of 3ms with a current limit of 20A. Ten measurements taken at 10µs intervals are taken 100µs before the pulse begins.

```
measThenPulse(5, 1, 20, 300e-6, 3e-3, 3, 100e-6, 10, 10e-6)
runPulse()
```

Results

The raw sample data is stored in the dedicated nonvolatile buffers, smua.nvbuffer1 and smua.nvbuffer2. Use the following command to print the data through the remote command interface:

```
printData()
```

The results for the example code are shown in ***Figure 10***. The load is a 0.5Ω resistor.

Appendix F. Code for Example #5, Post-Pulse Characterization: Triggering Measurements after the Pulse

Below is the TSP script that can be used to program the Model 2651A to perform Example #5. The commands for executing the code and obtaining the results are listed after the TSP script.

```
function pulseThenMeas(vPulse, vIdle, iLimit, pulseWidth, period, pulseCount, measDelay, measCount, sampleInterval)
        errorqueue.clear()
        reset()
        status.reset()

        --If the programmed pulse is negative and biased at zero, then program negative zero
        if vIdle == 0 and vPulse <0 then
                vIdle = -1e-12
        else
                vIdle = vIdle
        end

        -- Prepare the reading buffers
        smua.nvbuffer1.clear()
        smua.nvbuffer1.appendmode = 1
        smua.nvbuffer1.collecttimestamps = 1
        smua.nvbuffer1.collectsourcevalues = 0
        smua.nvbuffer2.clear()
        smua.nvbuffer2.appendmode = 1
        smua.nvbuffer2.collecttimestamps = 1
        smua.nvbuffer2.collectsourcevalues = 0

        --Set up the source
        smua.source.func = smua.OUTPUT_DCVOLTS --Set source function to DC Volts
        smua.sense = smua.SENSE_REMOTE -- Enable remote sense (4W measurements)
        smua.source.rangev = vPulse
        smua.source.levelv = vIdle --Sets the pulse off value.
        --The ‘if’ statement below sets the DC current limit
        if iLimit > 20 then
                smua.source.limiti = 20
        else
                smua.source.limiti = iLimit
        end
        smua.trigger.source.listv({vPulse}) -- Configures amplitude levels for source sweep
        smua.trigger.source.limiti = iLimit --Sets current limit during pulsing
        smua.trigger.source.action = smua.ENABLE --Enables source sweep

        --Set up the measurements
        smua.measure.rangei = iLimit -- Set the measure current range
        smua.measure.adc = smua.ADC_FAST --Configures SMU to use high speed ADC
        smua.measure.interval = sampleInterval -- Sets measurement interval
        smua.measure.delay = measDelay --Set the time from the end of the pulse to the start of measurements using
measure delay
        smua.measure.count = measCount --Sets number of measurements
        smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2) -- Configure the SMU to measure both voltage and
current
        smua.trigger.measure.action = smua.ASYNC -- Configures measurements to occur asynchronously with the source
sweep

        --Set up a digital I/O line to trigger the start of the period timer
        digio.trigger[1].mode = digio.TRIG_FALLING
        digio.trigger[1].clear()

        -- Timer 1 controls the start of the measurement and serves as system clock
        trigger.timer[1].delay = period
        --Timer must always have a count of 1 or more. The following ‘if’ statement sets the timer count
appropriately.
        if pulseCount > 1 then
                trigger.timer[1].count = (pulseCount - 1)
        else
                trigger.timer[1].count = 1
        end

        trigger.timer[1].passthrough = true --Timer issues an event at the start of the first interval.
        trigger.timer[1].stimulus = digio.trigger[1].EVENT_ID -- Period timer is triggered by a digital I/O line
        trigger.timer[1].clear()
        -- Timer 2 controls the pulse width
        trigger.timer[2].delay = pulseWidth
        trigger.timer[2].count = 1
        trigger.timer[2].passthrough = false --Only issue events after timer interval elapses
        trigger.timer[2].stimulus = trigger.timer[1].EVENT_ID -- Start timer interval when period timer issues event
        trigger.timer[2].clear()


        -- Configure SMU Trigger Model for Sweep

        -- Return to the bias level at the end of the pulse/sweep
        smua.trigger.endpulse.action = smua.SOURCE_IDLE
        smua.trigger.endsweep.action = smua.SOURCE_IDLE
        smua.trigger.count = pulseCount --Set number of pulses to complete
        smua.trigger.arm.count = 1
        smua.trigger.arm.stimulus = 0
        smua.trigger.source.stimulus = trigger.timer[1].EVENT_ID --Trigger source action when period timer generates
event
        smua.trigger.measure.stimulus = trigger.timer[2].EVENT_ID -- Trigger measurements when the end pulse action
is started
        smua.trigger.endpulse.stimulus = trigger.timer[2].EVENT_ID -- Start EndPulse action as soon as the pulse
width timer ends
end
```

Executing the Code

Send the following lines to configure and execute three pulses that pulse from 1V to 5V with a width of 300µs and a period of 3ms with a current limit of 20A. Twenty measurements taken at 10µs intervals are taken 100µs after the pulse ends.

```
pulseThenMeas(5, 1, 20, 300e-6, 3e-3, 3, 100e-6, 50, 10e-6)
runPulse()
```

Results

The raw sample data is stored in the dedicated nonvolatile buffers, smua.nvbuffer1 and smua.nvbuffer2. Use the following command to print the data through the remote command interface:

```
printData()
```

The results for the example code are shown in ***Figure 12***. The load is a 0.5Ω resistor.

Appendix G. Monitoring the Operation Overrun Event Register

The following TSP script demonstrates how to monitor the Model 2651A's operation overrun Event register for overruns to the Source, Measure, or EndPulse action blocks.

```
function checkSwpStatus()

        trigOvrEvent = status.operation.instrument.smua.trigger_overrun.event
        print("Overall smu trigger overrun event register value", trigOvrEvent)

        --For more detail look at bits 2, 3, 4
        --bit 2 = source, bit 3 = measure, bit 4 = endpulse
        --The bit.test function returns a boolean: true if set, false if not set
        sourceOvr = bit.test(trigOvrEvent, 3)
        print("Source Overrun bit:", sourceOvr)
        measOvr = bit.test(trigOvrEvent, 4)
        print("Measure Overrun bit:", measOvr)
        endPulseOvr = bit.test(trigOvrEvent, 5)
        print("EndPulse Overrun bit:", endPulseOvr)

        --If any bit in the register is set, then return a Boolean set to true for overrun occurred.
        if trigOvrEvent > 0 then
                ovrBool = true
        else
                ovrBool = false
        end
        return ovrBool
end
```

Methods to Achieve Higher Currents from I-V Measurement Equipment

The most flexible test equipment for sourcing and measuring current (I) and voltage (V) are source-measure units (SMUs) such as Keithley's Series 2600B System SourceMeter® instruments. The Series 2600B also includes three new benchtop models that offer best-in-class value and performance. These specialized instruments are high performance I-V source-measure instruments that are designed for use either as bench-top I-V characterization tools or as building block components of multi-channel I-V test systems. Each Series 2600B SourceMeter instrument combines a precision power supply, a true current source, a DMM, an arbitrary waveform generator with measurement, an electronic load, and a trigger controller – all in one instrument. In short, they can source I or V, and then measure V or I, simultaneously. They also support both polarities of I and V (sinking and sourcing power), referred to as "four quadrant operation."

By design, there is a limit to the maximum current or voltage that a single SMU can source and measure. This paper will present methods to achieve current levels during test sequencing that are higher than the published DC (direct current) specifications of a single SMU. Two techniques will be explored:

1. Pulse sweeps
2. Combining multiple SMU channels together

These techniques can be used to source and measure currents up to 40A for high-power applications such as:

- Solar cells and other photovoltaics
- Power management devices such as power MOSFETs and IGBTs
- High brightness light emitting diodes
- RF power transistors

Pulse sweeps

There is a limit to the DC maximum current or voltage that a single SMU can source and measure. This limit is a function of the inherent equipment design and is typically dependent on design parameters such as the maximum output of the power supply internal to the SMU itself, the safe operating area (SOA) of the discrete components used in the SMU, the spacing of the metal lines on the SMU's internal printed circuit board, etc. Some of these design parameters are constrained by maximum current limits, some by maximum voltage limits, and some by maximum power limits (I×V). A typical expression of the DC I-V limits of a four quadrant SMU is shown in ***Figure 1***. It shows a maximum DC current of 3A (point A in the figure) and a maximum voltage of 40V (point B). The maximum power the SMU can output is 40W, which is achieved at point B (1A×40V). At point A the power is lower at 18W. The difference can be explained, for example, that the maximum at point B is constrained by the maximum allowed power output of the on-board power supply, whereas at point A the limit is based on the maximum current (not power) that a key component can handle.

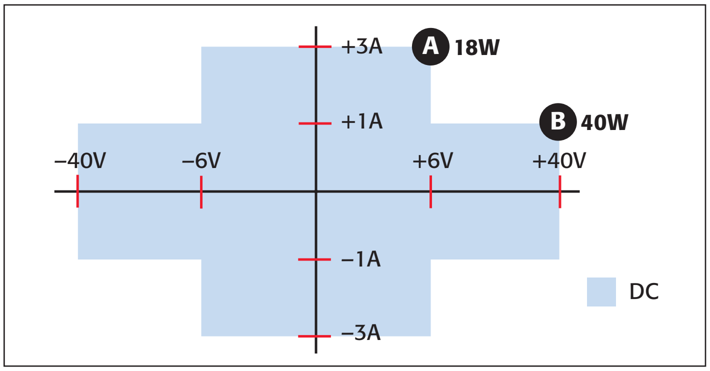


Figure 1

Figure 1 shows the DC (or continuous wave, CW) I-V limits, or performance envelope. Now consider if the SMU could produce a time-varying waveform such as a pulse. If the pulse waveform had 40V amplitude, 1ms pulse width, and a 50% duty cycle, then the effective CW power averaged over several seconds is 20W, not 40W. Depending on its design, it may be possible for that SMU to source higher current in pulse mode than in DC mode – the instantaneous maximum peak power in pulse mode is higher than DC peak power, but the CW power dissipation during pulse mode is less on average than in DC mode.

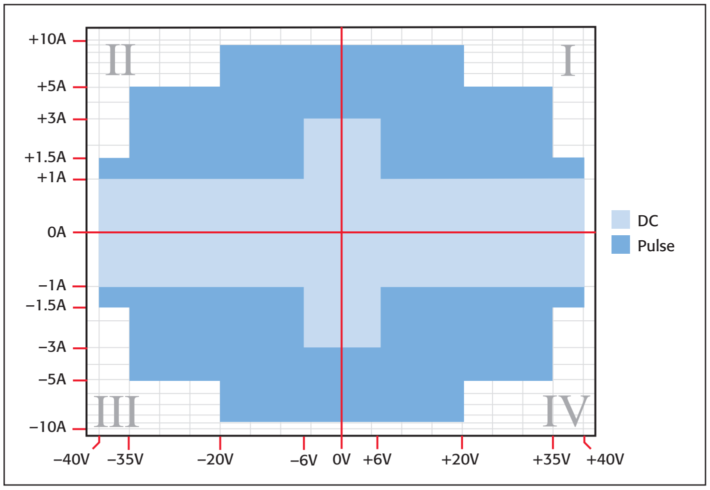


Figure 2

As an example, the pulsed I-V envelope is shown in ***Figure 2*** for the same SMU model shown in ***Figure 1***. There are constraints on the allowed pulse width and duty cycle, but by pulsing the instantaneous power can be as high as 200W (10A×20V). Although the instantaneous power may be high, the CW power based on allowed pulse width and duty cycle is below the DC power limit of 40W.

The higher instantaneous power when pulsing can be applied to achieve higher power I-V sweeps. Consider a standard scenario where a voltage bias is applied to a DUT (Device Under Test). The voltage values are swept over time, from low voltage to high voltage. Intermittently, the current is measured (schematic shown in ***Figure 3***). This generates I-V pairs that, when plotted, give a typical I-V sweep such as that shown in ***Figure 4*** for a P-N junction diode (1N5400 component). P-N diodes are encountered when measuring a solar cell or other photovoltaic (PV) device, or high-brightness light emitting diodes (HB-LEDs). In this sweep, voltage increments of 0.02V were used during the sweep. As shown in ***Figure 2***, when the applied voltage is less than 6V, that particular SMU's maximum allowed DC current is 3A, as demonstrated in ***Figure 4***.

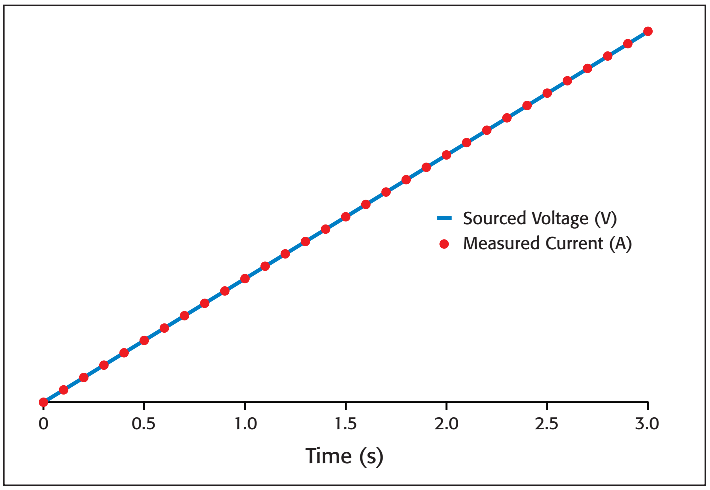


Figure 3

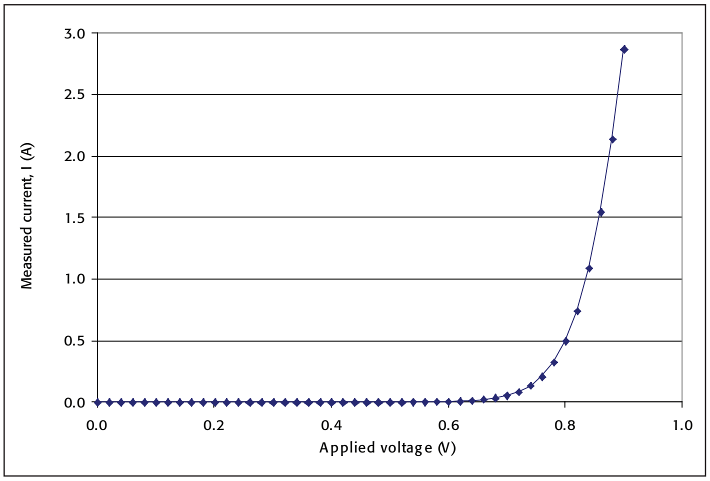


Figure 4

The schematic for a pulsed voltage sweep is shown in ***Figure 5***, which is equivalent to the DC sweep shown in ***Figure 3***. When current is not being measured, the sourced voltage values simply return to zero to keep the averaged CW power within the allowed limits. As seen in ***Figure 5***, the V bias values at which the I values are measured and the I sampling rates are identical to those in the DC sweep shown in ***Figure 3***. Performing pulse sweeps in this manner allows identical I-V values as DC to be achieved in the lower power regions (***Figure 6***), while allowing I-V curves in the higher power region up to 10A to be achieved ***(Figure 7***).

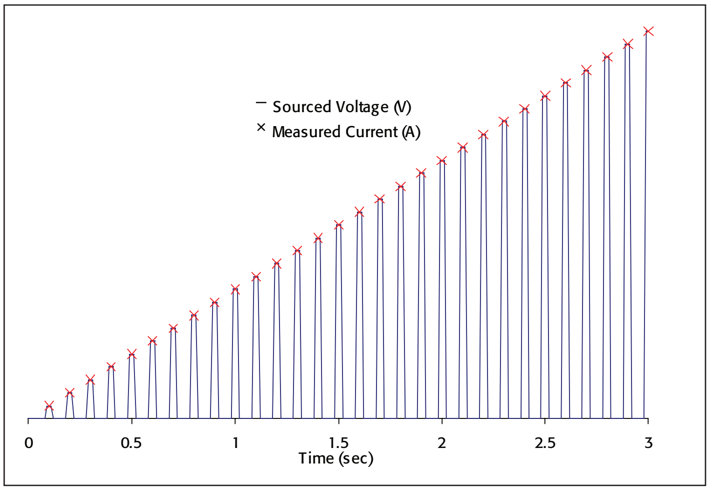


Figure 5

In ***Figure 6***, the DC and pulse sweep I-V curves are so closely overlaid that it is difficult to visually discern the differences between the curves. Therefore to quantify the excellent correlation, the relative percent difference between the curves is also calculated and shown in ***Figure 6***. At the higher current values of most interest, ±2% correlation is achieved between the DC and pulse sweeps (using the DC sweep values as the reference). With excellent correlation established at the lower current levels, we then use pulse sweeps to extend the I-V curves to 3× higher currents of 10A maximum than could be achieved with DC sweeps. This is shown in ***Figure 7***.

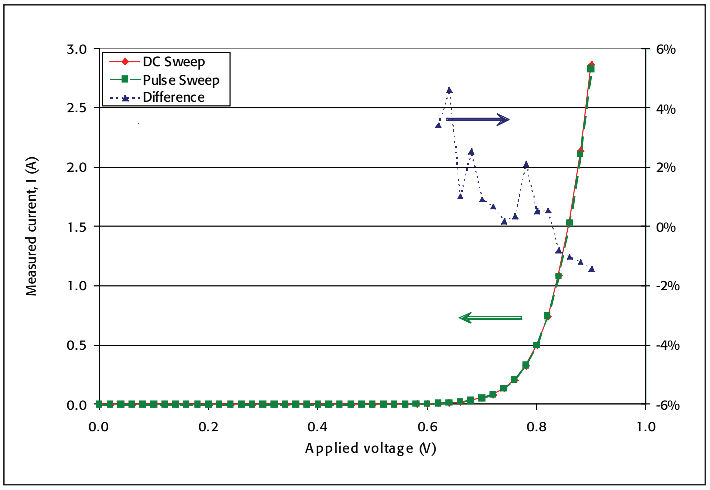


Figure 6

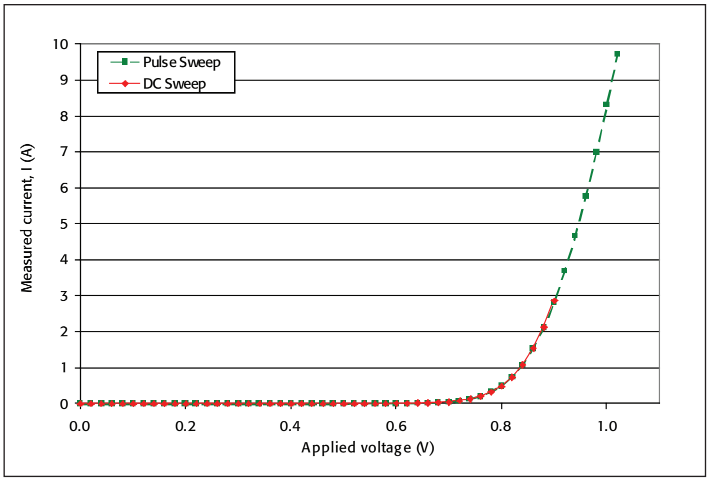


Figure 7

For many DUT types, a pulsed sweep can be substituted for a DC sweep to achieve higher power I-V sweeps with little impact on results. DUT types for which pulsed sweeps may not adequately correlate to DC sweeps are those DUTs potentially impacted by displacement current (second term in Maxwell's equation for current, $J_{tot} = J + \partial D/\partial t$). Large displacement currents can be generated at the sharp edges of the voltage pulse, and DUTs such as capacitors can have their electrical properties changed by large displacement currents.

However, there are many high power devices where pulsed I-V testing ***must*** be performed to get optimal results. The reason is that during high power CW testing, the semiconductor material itself starts to dissipate the applied power via thermal heating. As the material in the device heats up, the conduction current decreases as the carriers have more collisions with the vibrating lattice (phonon scattering). Therefore, the measured current is erroneously too low, due to so-called self-heating effects (http://www.keithley.com/data?asset=50742) caused by Joule heating. Because devices such as these are typically run in pulsed mode, intermittently, or AC and not run continuously on, the erroneously-low DC-measured currents are not an accurate characterization of their performance. In this case, pulsed testing must be used, and pulse width and duty cycle are explicitly stated in the device's published datasheet (see for example the inset in Figure 2 at http://www.fairchildsemi.com/ds/1N%2F1N5400.pdf). Example devices that require pulse testing are high-power RF power amplifiers and even low-power nanoscale devices.

The primary tradeoffs when migrating from a DC sweep to a pulse sweep are as follows:

- The pulse width must be wide enough to allow time for the device transients, cabling and other interfacing circuitry to settle so a stable, repeatable measurement can be made.
- The pulse width cannot be so wide so as to violate the test instruments' maximum pulse width and duty cycle limits, which would exceed the allowed power duty cycle of the instrument.

Combining multiple SMU channels to achieve higher DC current

The most commonly-used method of combining SMU channels to achieve higher DC currents is to put the current sources in parallel across the DUT, as shown in ***Figure 8***.

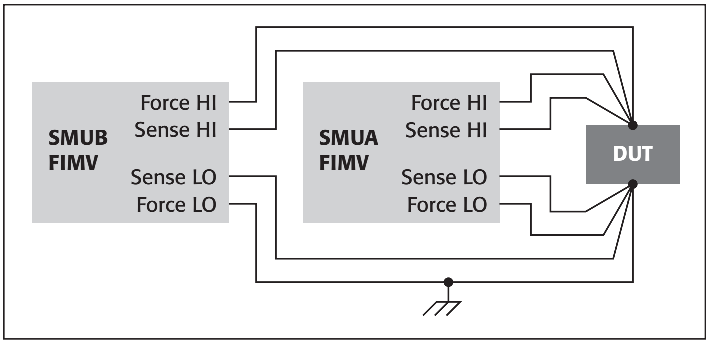


Figure 8

This test setup takes advantage of the well-known electrical principal that two current sources connected to the same circuit node in parallel will have their currents added together (Kirchhoff's current law). In ***Figure 8***, both SMUs are sourcing current and measuring voltage. The HI terminals refer to the high impedance terminals of the SMU and the LO terminals are the low impedance terminals. The "FORCE" terminals are the ones forcing current, and the SENSE lines are used for the four-wire voltage measurements. four-wire configuration is a mandatory requirement when high values of current are involved and is discussed later in this document. All of the LO terminal (FORCE and SENSE) of both SMUs are tied to earth ground.

Characteristics of this particular configuration are:

- Source Current:
  $I_{DUT} = I_{SMU\ A} + I_{SMU\ B}$
- Load Voltage:
  $V_{DUT} = V_{SMU\ A} = V_{SMU\ B}$
- Maximum Source Current:
  $I_{MAX} = I_{MAX\ SMU\ A} + I_{MAX\ SMU\ B}$
- Maximum Voltage: Limited to the smaller of the two SMUs maximum voltage capabilities. This is a result of very small variations between units of the maximum voltage that can be output when sourcing current. V_{MAX} = Smaller of $V_{MAX\ CMPL\ SMU\ A}$ and $V_{MAX\ CMPL\ SMU\ B}$.

Other notes:

- Set SMU A and SMU B output currents to the same polarities to obtain maximum output.
  - – While not absolutely required, the source polarity is generally the same for the two SMUs in this configuration.
- When possible, have one SMU in a fixed source configuration at a time and the other SMU performing the sweep. This is preferable to having both sweeping simulataneously. If both SMUs are sweeping, their output impedances are naturally changing, for example, as the meter autoranges. In

addition, the DUT's output impedance may also be changing significantly, for example, from high resistance off state to low resistance on state. With so many of the impedance elements in the circuit changing, this could increase overall circuit settling time at each bias point. Although this is a transient effect that damps out, nonetheless, fixing one SMU's source and sweeping the other usually results in more stable and faster-settling transient measurements, therefore higher test throughput.

- One ramification of having one SMU's current fixed and the others sweeping current is that for current levels well below I_{MAX}, one SMU will be sourcing or sinking much more current than the other. The current levels are not balanced between the two SMUs, but this does not cause any accuracy or precision issues when done with high quality SMUs. There is no particular reason to try to keep the SMUs at approximately the same current throughout the sweep.
- Therefore, to sweep from 0A to I_{MAX}, set both SMUs to source 0A and then sweep SMU B from 0A to $+I_{MAX\ SMU\ B}$. Next sweep SMU A from 0A to $+I_{MAX\ SMU\ A}$. Similar approaches can be used to sweep from I_{MAX} to 0A, or 0A to $-I_{MAX}$, or $-I_{MAX}$ to 0A.
- To sweep from $-I_{MAX}$ to $+I_{MAX}$, first set SMU A to $-I_{MAX\ SMU\ A}$ and SMU B to $-I_{MAX\ SMU\ B}$. Sweep SMU B from $-I_{MAX\ SMU\ B}$ to $+I_{MAX\ SMU\ B}$ and then sweep SMU A from $-I_{MAX\ SMU\ A}$ to $+I_{MAX\ SMU\ A}$. Again, there is no particular reason to try to keep the SMUs at approximately the same current. A similar approach can be used to sweep from $+I_{MAX}$ to $-I_{MAX}$.

- Both SMUs are sourcing current, but only let one SMU limit the maximum output voltage via the compliance setting. For example, set the I-source voltage compliance of SMU B greater than the compliance of SMU A, i.e. $V_{LIMIT\ SMU\ B} > V_{LIMIT\ SMU\ A}$.
  - If the DUT is an active source, the compliance setting of SMU A must be greater than the maximum voltage the DUT can source, to avoid putting the circuit and SMUs into an unknown state. For example, if the DUT is a 9V battery but SMU A's compliance is set to 5V, the results will be unpredictable and unstable.
  - Set the voltage readback measurement range of SMU B equal to its compliance range. There is no special requirement for SMU A's measurement range, but be aware of range compliance if the measurement range is less than compliance range and the instrument allows differences between real compliance and range compliance (such as Keithley Model 24xx and 6430).

Now we apply this technique of combining SMUs in parallel using the SMU models whose DC I-V power envelope are shown in ***Figure 2***. The same P-N diode DUT is used whose results are shown in ***Figure 4*** for a single SMU. By combining two SMUs in parallel, we expect to be able to double the maximum DC current measured from 3A to 6A.

This is confirmed by the results shown in ***Figure 9***. Up to 3A, the single-SMU and dual-SMU results are so closely correlated that is difficult to visually discern any differences between the results. As before, the relative percent difference between the results is calculated and plotted in the figure and, in most cases, shows ±1% correlation is achieved between the single-SMU and dual-SMU sweeps (using the single-SMU sweep values as the reference). With excellent correlation established at the lower current levels, we then use dual-SMU sweeps to extend the I-V curves to 2× higher currents of 6A maximum than could be achieved with single-SMU sweeps.

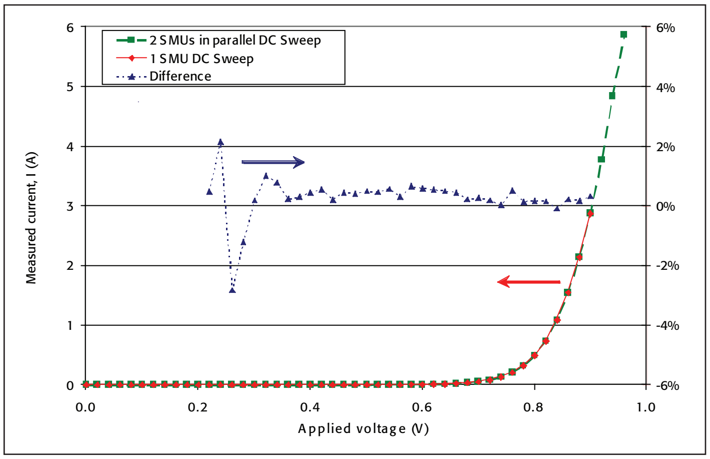


Figure 9

Pulse sweeps while combining multiple SMU channels

In this section we combine power-enhancement techniques of the Pulse Sweep method with the method of combining multiple SMU channels in parallel. Furthermore, we increase the number of SMU channels from two to four by using two dual channel SMUs such as the Keithley Model 2602B. As seen in ***Figure 2***, this SMU can achieve a maximum 10A pulse for DUT bias less than 20V. Therefore the maximum current now achievable is 40A (4×10A), which is more than 13× higher than the 3A that can be achieved by using a single SMU with DC sweeps.

Not surprisingly, great care must be taken when implementing this testing method. First, there is a personnel safety aspect: When dealing with hazardous voltages, it is critical to insulate or install barriers to prevent user contact with live circuits. Failure to exercise these precautions could result in electric shock or death.

There is also an aspect related to avoiding damage to the measurement equipment or the DUT. The multiple pulses must be tightly synchronized in time (on the nsec scale) so that one piece of equipment is not applying power and damaging units that are not turned on yet. Most SMUs on the market simply do not have the capability to synchronize on sufficiently short time scales and therefore are not suitable for implementing this type of test methodology; however, the Keithley Series 2600B SMUs have been intrinsically designed to do this. There are other important considerations when using more than two SMUs together, which will be discussed in the next section.

To set a baseline before combining SMUs, we perform a 10A pulse using a single SMU and observe the results on an oscilloscope. A high power precision resistor (0.01Ω, ±025%, KRL R-3274) was used as the test DUT and a pulse width of 300μs was programmed. We expect the oscilloscope to show a nearly square waveform of amplitude 0.1V (10A × 0.01Ω) and 300μs width, and, in fact, those are the results we see in ***Figure 10***. As also shown in ***Figure 10***, combining four SMUs in parallel to pulse 40A across the same DUT results in the expected 0.4V magnitude with excellent synchronization (low jitter) between the channels.

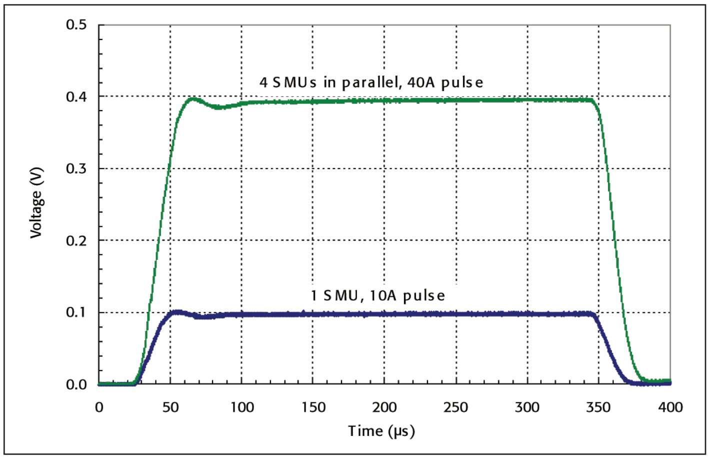


Figure 10

Using the pulse waveform shown in ***Figure 10*** (40A amplitude, 300μs width, generating 0.4V across a 0.01Ω ±0.25% resistor), repeatability testing was done to verify pulse consistency. This is a particularly stringent test that simultaneously checks both the high I sourcing performance along with low V measurement performance. The results are shown in ***Figure 11***, with a 3σ standard deviation of 0.045% observed across 25 repetitions done in quick succession.

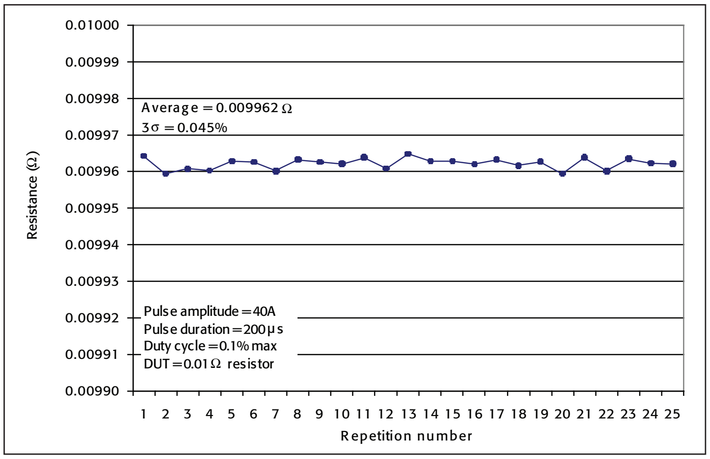


Figure 11

With the pulse performance verified, we program a pulse sweep combining four SMUs and repeat the I-V curve on the P-N diode test DUT. The results are shown in ***Figure 12***. We see excellent correlation with the 1-SMU DC sweeps up to 3A, and with the 1-SMU pulse sweep up to 10A. Then, we extend the achievable I-V curve up to 40A.

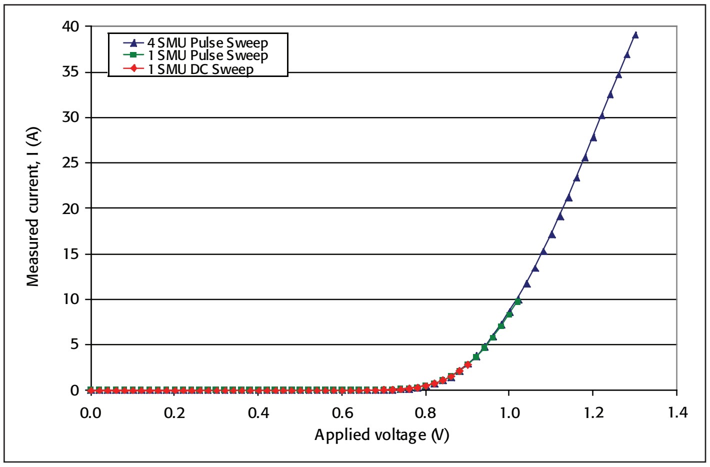


Figure 12

With the results of this technique (combining four SMU channels and pulsing to achieve 40A) verified on two-terminal devices (resistor and diode), the technique is next applied to a three-terminal device, a high-power MOSFET (IRFP240, datasheet available at www.datasheetcatalog.org/datasheet/fairchild/IRFP240.pdf). Such devices have high operating drain current $I_{D(ON)}$ (>20A at V_{DS}>20V, V_{GS}>10V), and low drain-source on resistance $r_{DS(ON)}$ (<0.2Ω at V_{GS}=10V, I_D=10A). Typical electrical parameters that are measured on such a device include:

- I_D–V_{DS} curves for a variety of V_{GS} values
- $r_{DS(ON)}$–I_D curves for different V_{GS} values
- Threshold voltage (gate to source), $V_{GS(TH)}$
- Drain-to-source breakdown voltage, BV_{DSS}
- Continuous source-to-drain current and voltage, I_{SD} and V_{SD}

For the three-terminal measurements, four SMUs are connected in parallel across the drain-source nodes to enable 40A pulsed currents; a fifth SMU is connected across the gate-source nodes to provide the gate bias.

The I_D–V_{DS} curves for a variety of V_{GS} values are shown in ***Figure 13***. The measurements and curves are the expected results, with the effects of minor device self-heating observed at around V_{DS}=10V for the V_{GS}=7V and V_{GS}=8V curves. Self-heating is expected for the pulse width of 1000μs that was used. Note that nearly 800W peak power (20V × 40A) is achieved with this tests setup.

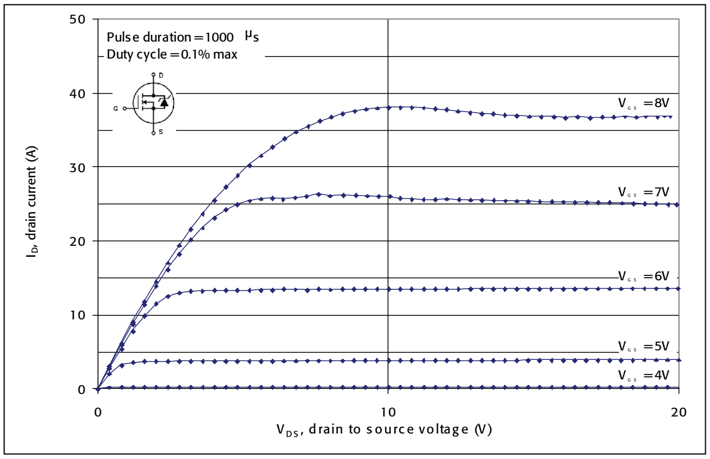


Figure 13

The $r_{DS(ON)}$–I_D curve was measured for V_{GS}=10V up to the maximum measurable drain current of I_D=40A. The results are shown in ***Figure 14***, and, for comparison purposes, the data from the device's published datasheet (Figure 8 in www.datasheetcatalog.org/datasheet/fairchild/IRFP240.pdf) are also shown. The correlation is excellent when using this multiple-SMU technique combined with pulse sweeping. With the standard, single-SMU DC sweep, the curve would have ended at 3A, which is not sufficient to properly characterize the device.

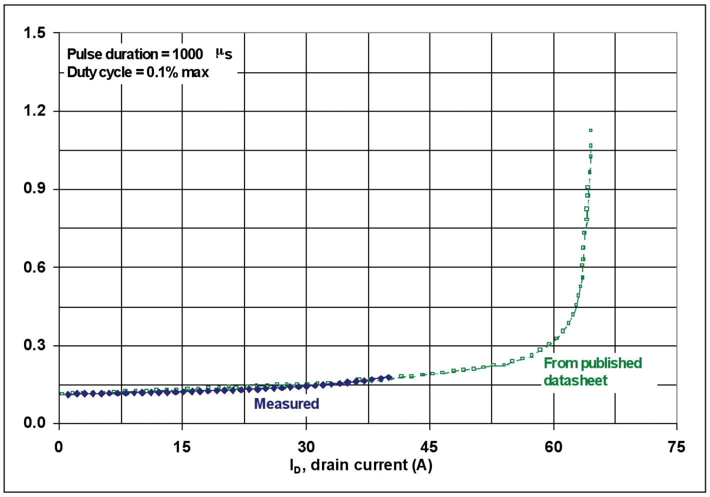


Figure 14

Finally, the I_{SD}–V_{SD} curves for V_{GS}=0V are shown in ***Figure 15***. These are compared to the results from the device's published datasheet (Figure 13 in www.datasheetcatalog.org/datasheet/fairchild/IRFP240.pdf), and, again, correlation is very good.

With a modern smart SMU like the Keithley Series 2600B, which has its own microprocessor, onboard memory, and math and logic programming functions via an embedded open-source scripting language, it is easy to run the SMU via a graphical user-interface for benchtop applications, or have the scripts resident on the SMU with no need for a control PC for high-speed parallel test production applications. Also, because the Keithley Series 2600B SMUs are mainframeless, the exact number of channels can be expanded to the desired number without having to incur the additional cost or power limitations of a mainframe.

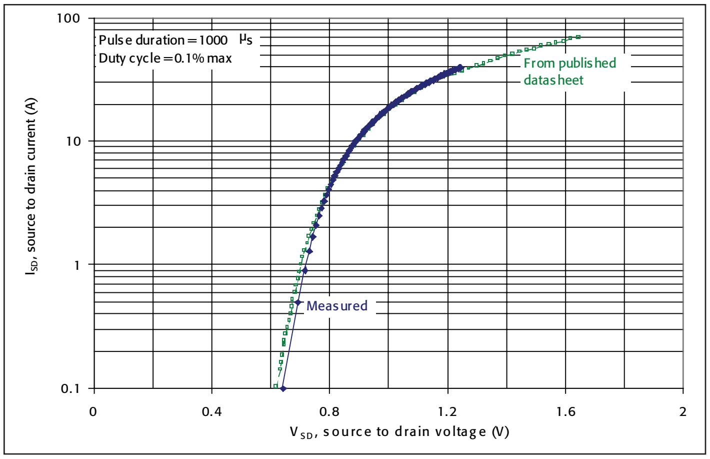


Figure 15

Important test implementation details

This section describes key implementation details that significantly improve the accuracy and precision of the results obtained using this multi-SMU pulsed sweep approach.

Source readback

Consider the case when a test applies a voltage to a DUT and measures a current. Because an SMU has both source and measure functions built into the same unit, it can also read back the actual value of the applied voltage using its measurement circuitry. This is a source-measure-measure sequence not just source-measure: Source voltage, measure (readback) applied voltage, measure resulting current across the device. A typical reason why the programmed value for the source voltage is not the same as the voltage applied to the DUT is that the DUT is sinking a large current, which slightly loads the voltage source. In that case, the actual measured voltage values that are read back are typically slightly lower than the values that were programmed. A comparison was done for the P-N diode DUT used previously, and the results are shown in ***Figure 16***. At the maximum point of the I-V (current about 40A), the programmed voltage is 1.3000V and the actual measured sourced value is 1.2917V, a small 0.64% error, which may or may not be impactful depending on the actual application. The error manifests itself primarily as a small offset on the voltage axis (left-right shift on the X-axis in ***Figure 16***) with little impact to the measured current values (little shift on the Y-axis).

Four-wire measurements

Four-wire (Kelvin) measurements must be used when doing high current testing. A four-wire measurement bypasses the voltage drop in the test leads by bringing two very high impedance voltage sense leads out to the DUT. With very little current flowing into the SENSE leads, the voltage seen by the SENSE

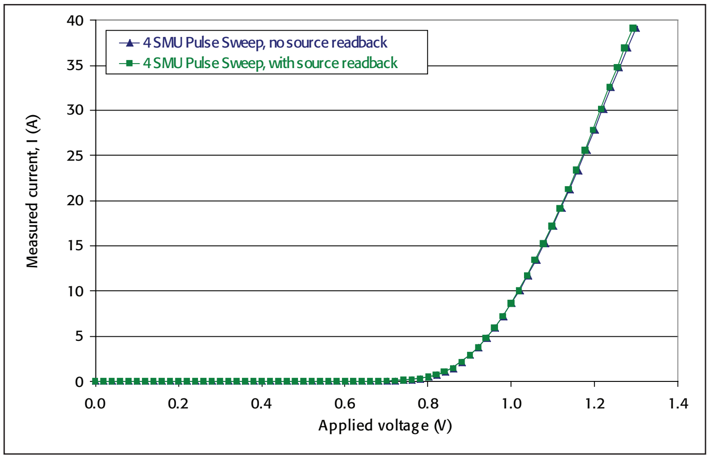


Figure 16

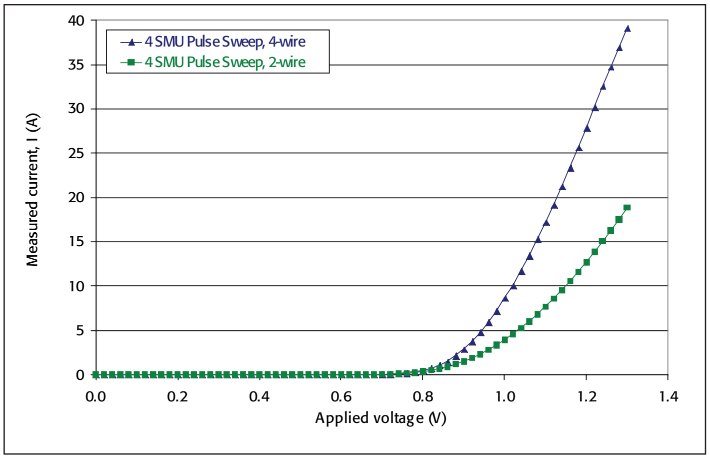


Figure 17

terminals is the virtually same as the voltage developed across the unknown resistance.

This is very important when high currents are being tested. At 40A levels, even a small resistance such as 10mΩ in the test cable can generate a voltage drop of 0.4V. So if the SMU is forcing 1V at 40A current and the cable resistance is 10mΩ and there are two test leads, the DUT might only receive a voltage of 0.2V, with 0.8V dropped across the test cables. A more detailed discussion of four-wire measurements can be found at http://www.keithley.com/data?asset=10636.

Unlike source readback, which primarily impacted just the source values, implementing four-wire measurements will result in significantly more accuracy on both the sourced and measured values. That is because most good-quality modern SMUs have an analog feedback control loop; in other words, if it is programmed to source, say, 1V, but the measured value at the DUT using four-wire is only 0.2V, then the SMU will increase the current it sources to compensate for the loads and voltage drops in the circuit, until the four-wire voltage value reaches the programmed source value (within the loop exit limits). The DUT's bias will be closer to the desired value; hence, the measured value will be more accurate. Therefore, by enabling four-wire measurement capability, both the sourced values and the measured values will be more accurate (impacts both axes in an I-V curve).

The results of two-wire versus the more accurate four-wire results are shown in ***Figure 17*** for the P-N diode used previously. As seen in that figure, due to the voltage drops in the test leads, in two-wire mode the DUT sees only a small fraction of the intended applied voltage, and, therefore, the forward current is lower. At 1.3V bias in 2-wire mode (bearing in mind that the DUT will get less voltage around 1.1V due to uncompensated voltage drops from the resistance in the test leads), the measured current is less than 20A, half of the real value (40A) when four-wire mode is used. This is a significant error, justifying the benefit of four-wire full Kelvin (not quasi-Kelvin) testing.

Of course, to achieve best results, every effort must be made to place the test leads for the four-wire Kelvin connection as close to the DUT as is possible.

Maximum of one voltage source at each DUT node

It is common in many test sequences to perform voltage sweeps (force voltage) and measure current (FVMI). In the case of more than one SMU connected in parallel to a single terminal of the device, the obvious implementation would be to have all of the SMUs in V-source mode and measure I. However, three factors must be considered:

- SMUs when sourcing voltage are in a very low-impedance state.
- DUTs can have impedances higher than an SMU that's in V-source mode. The DUT's impedance can be static or dynamic, changing during the test sequence.
- Even when all SMUs in parallel are programmed to output the same voltage, small variations between SMUs related to the instruments' voltage source accuracy means that one of the SMU channels will be at a slightly lower voltage (mV order of magnitude) than the others.

So, if three SMUs are connected in parallel to one terminal of a DUT, and each SMU is forcing voltage and outputting near-maximum currents, and the DUT is in a high impedance state, then all current will go to the SMU which is sourcing the slightly lower voltage. This will most likely damage that SMU. Therefore, when connecting SMUs in parallel to a single terminal of a DUT, only one SMU should be sourcing V, as shown, for example, in ***Figure 18*** for the multiple SMUs connected to the drain of the MOSFET whose results were shown in ***Figures 13–15***.

Even if the configuration in ***Figure 18*** is used, extreme care must be taken so that throughout the sweep only one SMU is in FVMI mode and that none of the SMUs in FIMV mode automatically or inadvertently change to FVMI mode. An SMU can change from FIMV to FVMI mode, for example, when an SMU in FIMV mode reaches its programmed voltage level for source compliance. When an SMU is sourcing current to a programmed level in FIMV mode and the DUT sinks current, the SMU will automatically increase its output voltage via an analog feedback loop so as to maintain the programmed output current. The maximum voltage it will use is set by the user via a source compliance voltage level. When an SMU is in FIMV mode and

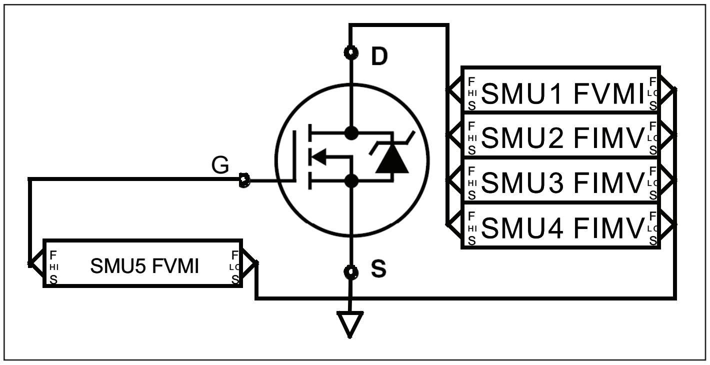


Figure 18

source voltage compliance is reached, the SMU switches modes and becomes a low-impedance voltage source and is at risk to being damaged. To prevent that from happening:

- Voltage compliance levels should be set appropriately (typically as high as possible)
- Control code can be written to monitor results during the sweep and take corrective action to avoid compliance as the instrument approaches compliance levels.

One practical implementation of a maximum of 1V-source at each DUT node is to have no SMUs in V-source, all in I-source. The sweep then would be entirely I-bias (not V-bias). While this is easiest to implement, this method suffers from the fact that I-V data will not be equally spaced on the voltage axis; they will be equally spaced on the current axis. This might complicate or confuse some standard analysis algorithms. A demonstration of this is shown in ***Figure 19***, where all SMUs are FIMV and none are V-source. These results should be compared to ***Figure 13***, where one SMU is in V-source mode and, therefore, the data points are equally spaced in voltage.

The method shown in ***Figure 18*** works on the basis that one SMU controls the output voltage while the rest of the SMUs supplement the current. To do this, one SMU is configured as a V source while the rest are configured as current sources for the entire sweep. The SMUs supplying current are supplying it at the level measured by all SMUs at the previous bias point.

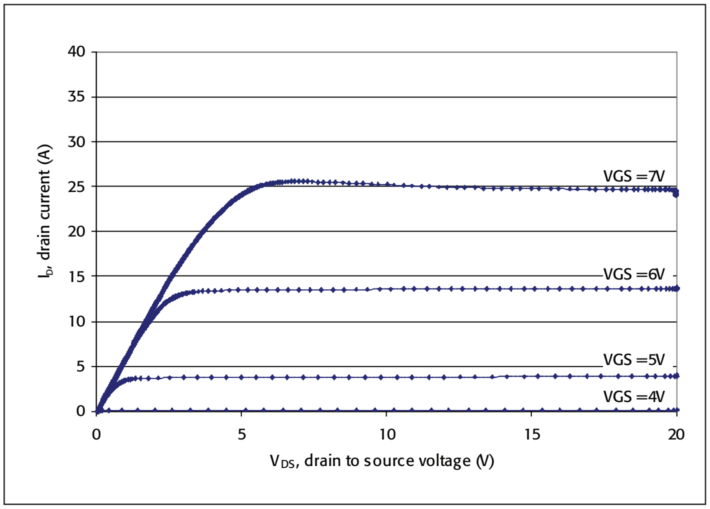


Figure 19

So in the case of four SMUs for the results shown in ***Figures 12–17***:

- Put SMU1 in V-source mode; SMU2, SMU3, SMU4 in I-source mode
- Determine V_{step}
- Initially SMU1 sources 0V (or a voltage level that will result in a DUT current that is less than the maximum current that SMU1 can handle on its own). SMU2–4 source 0A.
- Sweep loop:
  - – All 4 SMUs measure I.
  - – Calculate Itotal = $I_{SMU1} + I_{SMU2} + I_{SMU3} + I_{SMU4}$
  - – $I_{bias} = I_{total}/3$
  - – Set SMU1 to source voltage at the level of the previous voltage plus V_{step}
  - – Set SMU2, SMU3, SMU4 to source current each at a level of I_{bias}
  - – Repeat loop until exit condition is reached

Mitigating excessive energy dissipation due to device breakdown

When two SMUs of the same capability are connected in parallel to a single node in the circuit, one SMU is always capable of sinking all of the current being output by the other SMU. This scenario can occur, for example, when a DUT breaks down, becomes an open (near-infinite impedance), and is no longer a continuity path where current can flow. There is a short time during which 1 SMU has to sink all the current from the other.

However, when there are more than 2 SMUs connected in parallel at a single circuit node, 1 SMU cannot sink all of the current coming from the other SMUs. The SMU(s) that will be forced to sink current if the DUT breaks down are the SMUs at the lowest voltage or lowest impedance, most likely the ones sourcing voltage. In order to protect the signal input of the SMU forcing voltage, a diode such as the 1N5820 can be used. This limits the amount of current that can go into the SMU. A diode is preferable, because a fuse is too slow, and a resistor will cause too large of a voltage drop a across it. A diode has a much faster response than a fuse, and the diode has a much smaller maximum voltage drop across it (typically around 1V) than a resistor.

Adding diode protection to the test setup previously shown in ***Figure 18*** results in the circuit schematic shown in ***Figure 20***. Although SMU5 is also in FVMI mode, it does not require input protection because its Force-HI connection is connected to the high-impedance gate node. Also, in the test setup shown in ***Figure 20***, extra test code has been implemented to ensure that SMUs 2–4 will not reach compliance during the test sequence, to ensure they don't switch to FVMI mode. If this extra test code is not used, hardware protection should be added to the inputs of SMUs 2–4 in case any of them reach source voltage compliance.

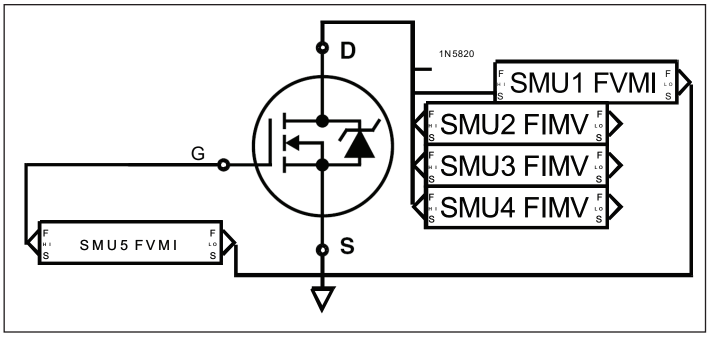


Figure 20

To demonstrate little impact on results of adding diode protection to SMU1, we repeat the $r_{DS(ON)}$–I_D curve measured on the IRFP240 part for V_{GS}=10V up to the maximum measurable drain current of I_D=40A (results previously shown in ***Figure 14***), both with and without diode protection. The results are shown in ***Figure 21***. The results overlay so well they are almost indistinguishable, and, upon calculating the relative percent difference between the two curves and plotting, in most cases, shows ±1% correlation is achieved between the no-diode and diode cases. This confirms that diode protection can and should be used.

It is also important to ground the LO terminals of both SourceMeter units, as shown in ***Figures 8***, ***18***, and ***19***. If the DUT becomes grounded and the steps above are not followed carefully, the SMUs could be damaged.

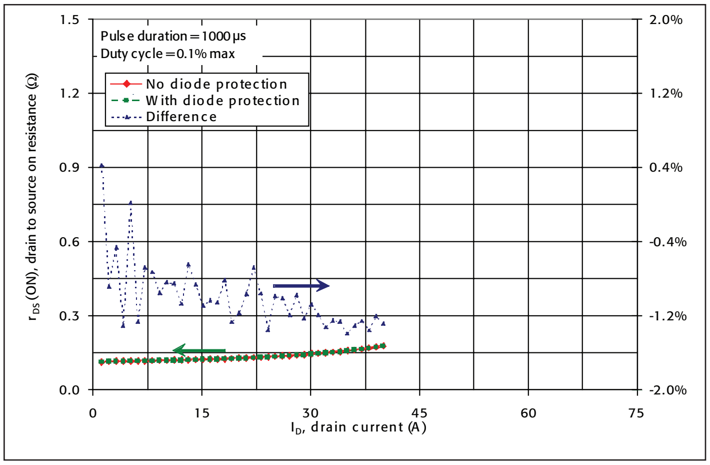


Figure 21

Cabling and test fixture considerations

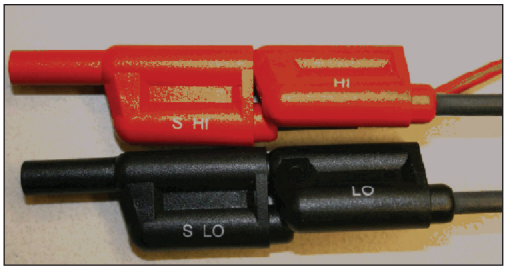
In general, test cabling and test connections must all be designed to minimize resistance (R), capacitance (C), and inductance (L), between the DUT and SMU.

To minimize resistance, use thick gauge wire (14 gauge is acceptable, 12 gauge is better) wherever possible, and definitely within the test fixture itself. Cable resistance can range from 30-300 mΩ/meter and higher, so obviously choose cabling at the lower end of that range. Keep cable lengths a short as possible and in no case longer than one meter.

Cables used in this document were Keithley Model 2600B-BAN: 1m (3.3 ft) banana test leads/adapter cable, which provide safety banana connections to Hi, Sense Hi, Lo, Sense Lo, and guard. The test leads were connected to the DUTs using alligator clips with boots (barrels accept standard banana plugs), such as those in Keithley Model 5804 Test Lead Set.

It is generally thought that guarding can minimize the effects of cable charging, but this is typically more of a concern for high voltage testing and not for high current testing. Guarding was not used in the results shown in this paper.

Four-wire Kelvin connections must be as close to the DUT as possible (every mm matters). So if using banana test leads and "piggybacking" the jacks, the Sense leads should be in front of the Force leads, as shown in the photo below. Putting the Sense leads behind the Force leads will degrade the results. While "piggybacking" is acceptable for the SMUs forcing current, the SMU forcing voltage must have its sense leads separated and right at the DUT (for example using alligator clips) in order to have proper 4-wire operation. Also, it should be noted that the voltage readback should be done with the SMU forcing voltage, because the current SMU's voltage readings will all vary quite a bit due to the connections, and will be different then what is actually at the DUT.



The jacks used on the test fixture should be high quality. In particular, some red jacks use high amounts of ferrous (Fe, iron) content to produce the red coloring, and these jacks can have unacceptably high levels of leakage due to conduction. The resistance between the plugs to the case should be as high as possible and in all cases >10^{10}Ω.

Many published test setups recommend to add a resistor between the SMU and the device's gate in the case of test a FET or IGBT. For example in ***Figure 20***, a 10kΩ resistor would be added between SMU5 and the gate node. This resistor can stabilize measurements, and, because the gate does not draw much current, the resistor does not cause a significant voltage drop.

If voltages in excess of 40V will be used during the test sequence, the test fixture and SMUs must have the proper interlock installed and be operational according to normal safety procedures.

Summary

Methods were shown how to increase from 3A to 40A the maximum current level that can be measured:

1. Pulse sweeps
2. Combining multiple SMU channels in parallel to achieve higher current

Example results using these techniques were given for commercially-available devices, and the results show excellent correlation with the published datasheets.

In addition, important test implementation factors were discussed in detail, including source readback, four-wire measurements, single V-source at each DUT node, and mitigating excessive energy dissipation due to device breakdown.

Combining Keithley Model 2651A High Power SourceMeter® Instruments for 100A Operation

Introduction

Source-measure units (SMUs), such as the Keithley Model 2651A High Power System SourceMeter instrument, are the most flexible and most precise equipment for sourcing and measuring current and voltage. Because of this, they are widely used to test semiconductor devices such as MOSFETs, IGBTs, diodes, high brightness LEDs, and more.

With today's focus on green technology, the amount of research and development being done to create semiconductor devices for power management has increased significantly. These devices, with their high current/high power operating levels, as well as their low On resistances, require a unique combination of power and precision to be tested properly. A single Keithley Model 2651A is capable of sourcing up to 50A pulsed and 20A DC. For applications requiring even higher currents, Model 2651As can be combined to extend their operating range to 100A pulsed.

This application note demonstrates how to collect Rds (on) measurement data from a power MOSFET device by using a pulsed current sweep to test up to 100A (see ***Figure 1***); however, it can be easily modified for use in other applications. The document is divided into three sections: theory, implementation, and example.

Theory

Kirchhoff's Current Law says that the sum of the currents entering a node is equal to the sum of the currents leaving the node. In ***Figure 2***, two current sources representing SMUs and a device under test (DUT) are connected in parallel.

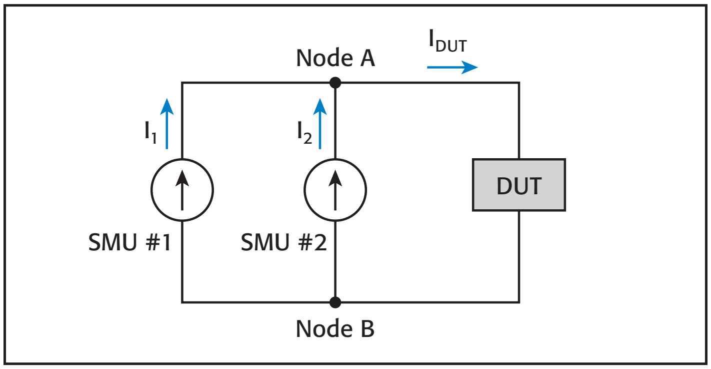


Figure 2: The sum of the currents entering the node equals the sum of the currents leaving the node.

In ***Figure 2***, we can see that two currents, I_1 and I_2, are entering Node A and a single current, I_{DUT}, is leaving Node A. Based on Kirchhoff's Current Law we know that:

$$I_{DUT} = I_1 + I_2$$

This means that the current delivered to the DUT is equal to the sum of the currents flowing from each SMU. With two SMUs connected in parallel, we can deliver to the DUT twice the amount of current that can be delivered by a single SMU. Using this method with two Model 2651As, we can deliver up to 100A pulsed.

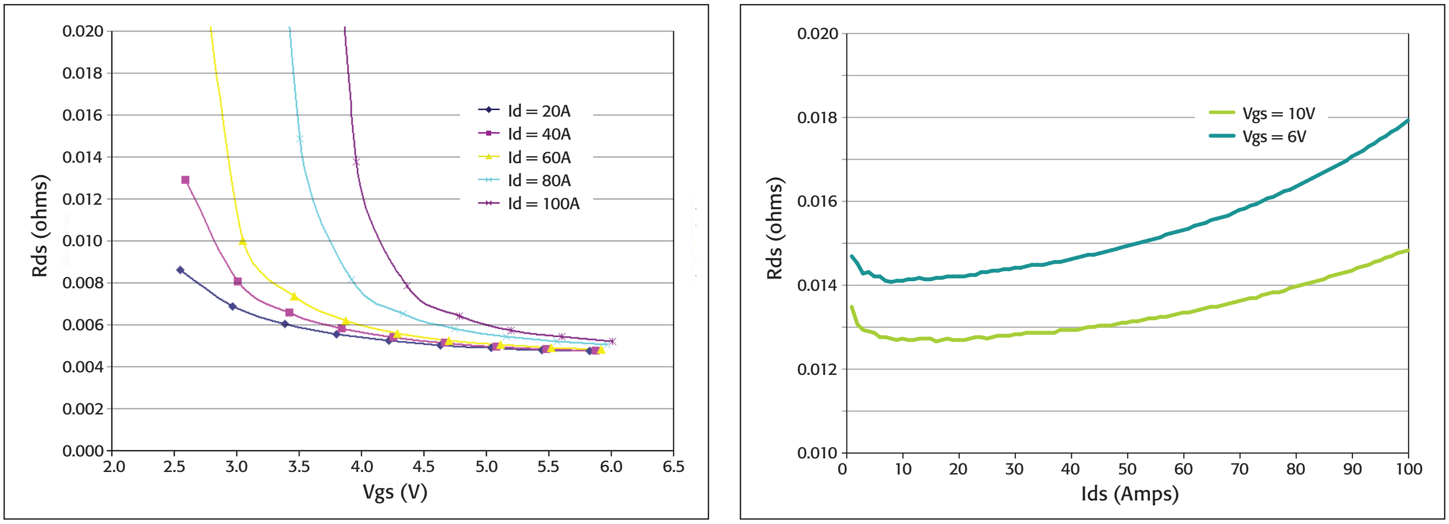


Figure 1: Example results after performing a pulsed Rds(on) current sweep (500µs pulse width and 0.01 NPLC) to test up to 100A on a power MOSFET device using two Model 2651A SourceMeter instruments connected in parallel.

Implementation

In order to create a current source capable of delivering more current than a single SMU can provide, we put two SMUs, both configured as current sources, in parallel. Below is a quick overview of what needs to be done to successfully combine two Model 2651As so that together they can source up to 100A pulsed. The following sections explain each item in detail.

1. Use two Model 2651As, running the same version of firmware.
2. Use the same current range for both SMUs.
3. Use the same regions of the power envelope (***Figure 3***) for both SMUs.
4. Use 4-wire mode on both SMUs with Kelvin connections placed as close to the DUT as possible.
5. Use the Keithley supplied cables. If this is not possible, ensure your cabling matches the specifications of the Keithley-supplied cable.
6. Set the voltage limit of both SMUs. (When the output of an SMU reaches its voltage limit, it goes into compliance.) The voltage limit of one SMU should be set 10% lower than the other SMU.
7. Select the output off-mode of each SMU. This determines whether an SMU will function as a voltage source set to 0V or as a current source set to 0A when the output is turned off. When two SMUs are functioning in parallel as current sources:
   - The SMU with the lower voltage limit should have its output off-mode set to NORMAL with the off function set to voltage, and
   - The SMU with the higher voltage limit should have its output off-mode set to NORMAL with the off function set to current.

Identical Model

Both SMUs MUST be the same model, the Model 2651A. This ensures that if the SMUs are forced into a condition in which one SMU must sink all of the current from the other SMU, the SMU that is sinking is capable of sinking all the current. For this reason, combining different model SourceMeter instruments in parallel is NOT recommended. In addition, both SMUs should be running the same version of firmware to ensure that both SMUs perform the same.

Source Current Range

Both SMUs should be set to the same source current range. How an SMU responds to a change in current level can vary with the current range on which it is being sourced. By configuring both SMUs to source on the same current range, both SMUs will respond similarly to changes in current levels. This reduces the chances for overshoots, ringing, and other undesired SMU-to-SMU interactions.

Region of Power Envelope

Both SMUs should be configured to operate in the same region of the power envelope (see ***Figure 3***). In order for one SMU to sink all the current of the other SMU, the sinking SMU must be operating in an equivalent region of the power envelope as the sourcing SMU.

When configured as a current source, the region of the power envelope in which the SMU is operating is determined by the source current range and the voltage limit value. When combining SMUs in parallel, each SMU should be set to the same source current range so the final determining factor for the region is the voltage limit. As can be seen in ***Figure 3***, the Model 2651A has three ranges of voltage limit values that determine the operating region: >0V to ≤10V, >10V to ≤20V, and >20V to ≤40V. For example, if one SMU's voltage limit is set to 20V, then the other SMU's voltage limit should be set to a value that is less than 20V and greater than 10V in order to keep both SMUs in the same operating region.

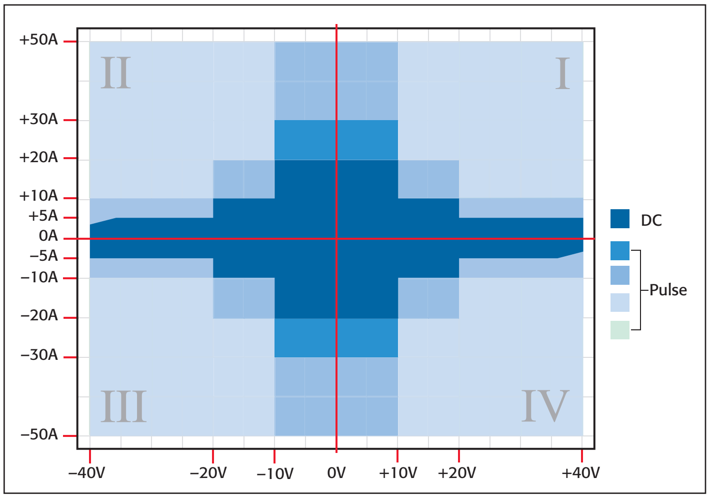


Figure 3: Power envelope for a single Model 2651A.

Connections

A simple connection diagram for combining two SMUs in parallel for higher current can be seen in ***Figure 4***.

Because the Model 2651A can produce such high currents, test leads (even those with very little resistance) can produce significant voltage drops and create errors in the voltage measurements. To eliminate these errors, use 4-wire mode on both SMUs with Kelvin connections placed as close to the DUT as possible.

Cabling Considerations

Cables capable of supporting the high levels of current that the Model 2651A can produce should be used to obtain the desired performance. The cable provided by Keithley with the Model 2651A is designed for both low resistance and low inductance. We recommend using this cable from the Model 2651A to as close to the DUT as possible. If the Keithley cable cannot be used, use wiring with as low a resistance and inductance as

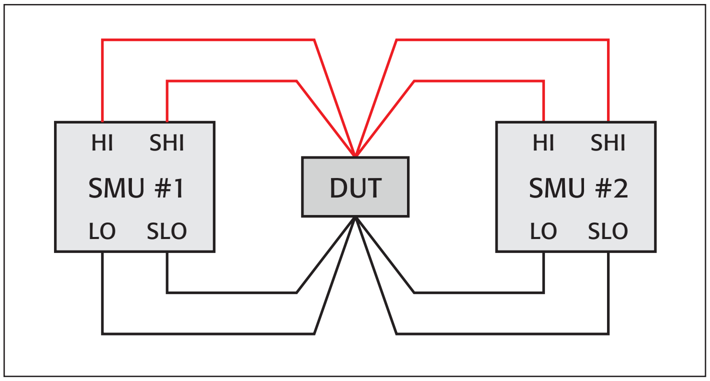


Figure 4: Wiring diagram for connecting two SMUs in parallel using 4-wire mode.

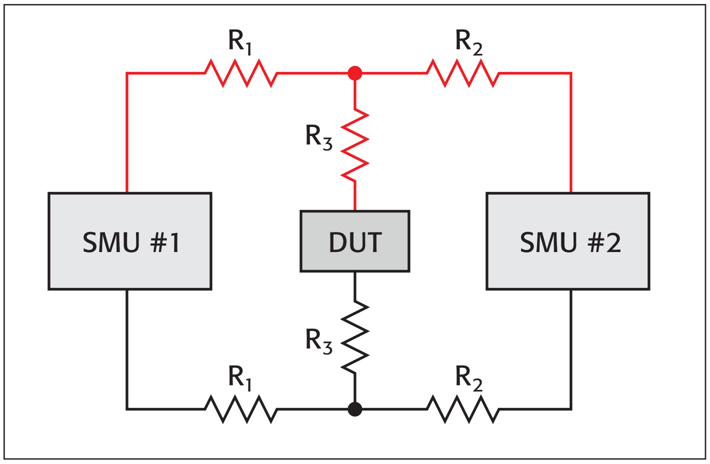


Figure 5: Model for test lead resistances.

possible. We recommend that wire of 12 AWG or thicker be used with a single Model 2651A. When combining SMUs for greater current, 10 AWG or thicker wire should be used. Guidelines for cabling should be taken seriously since wiring not rated for the current being sourced can affect the performance of the SMU and could also create a potential fire hazard.

The following sections discussing resistance and inductance are provided to help you verify that the cables you are using will allow your system to function properly.

Resistance

The Model 2651A has the ability to compensate for errors caused by voltage drops due to resistance in the force leads when large currents are flowing. This allows the Model 2651A to deliver or measure the proper voltage at the DUT rather than at the output of the instrument. This is done by using Kelvin connections.

The resistance of any cabling and connections between the SMUs' output and the DUT should be kept as low as possible to avoid excessive voltage drops across the force leads. This is because there is a limit to how large a voltage drop an SMU is capable of compensating for without adversely affecting performance. In the Keithley Model 2651A, this limit is 3V per source lead, which is imposed by the Kelvin connections.

A single Model 2651A is capable of sourcing up to 50A pulsed and up to 20A DC. Using Ohm's Law we can calculate the maximum resistance allowed in our test leads so as not to exceed the 3V limit under these maximum conditions. Ohm's Law states:

$$V = I \cdot R$$

where: V is voltage, I is current, and R is resistance. If we rewrite this equation, solving for R we get:

$$R = V/I$$

To find the maximum resistance values allowed in our test leads, we can substitute our limits for V and I.

$$\frac{3\text{V}}{20\text{A DC}} = 0.15\Omega \qquad \frac{3\text{V}}{50\text{A Pulse}} = 0.06\Omega$$

Based on these calculations, the resistance of each source lead should not exceed 150mΩ when only DC testing is used and should not exceed 60mΩ when pulse testing is used.

For example, in ***Figure 5*** the length of the test lead represented by R_3 should be as short as possible in order to minimize its resistance value (and thus the voltage drop across R_3). In this configuration, the current that flows through R_3 is the sum of the current flowing through R_1 and R_2. If we assume $R_1 = R_2 = R_3$ and that both SMU #1 and SMU #2 are delivering the same amount of current to the circuit, then the voltage drop across R_3 is twice as large as the voltage drop across R_1 or R_2 because twice as much current is flowing through R_3 as there is through R_1 or R_2. The voltage drop that each SMU sees is the sum of the voltage drop across R_3 and the voltage drop across its own lead resistance, R_1 or R_2.

Inductance

The Model 2651A also has the ability to compensate for errors caused by voltage drops due to inductance in the force leads. As mentioned in the discussion about resistance, this allows the Model 2651A to deliver or measure the proper voltage at the DUT rather than at the output of the instrument. Inductance in connections resists changes in current and tries to hold back the current by creating a voltage drop. This is similar to resistance in the leads. However, inductance only plays a role while the current is changing, whereas resistance plays a role even when current is steady.

The inductance of connections between the SMUs' outputs and the DUT should be kept as low as possible to minimize impacting SMU performance. To drive fast rising pulses, the Model 2651A must have enough voltage overhead to compensate for the voltage drop created by the inductance. If the supply does not have enough overhead, the inductance can slow the rise time of the pulse.

Another reason why the inductance of connections between the SMUs' outputs and the DUT should be kept as low as possible is that if the inductance causes a voltage drop large enough to exceed the 3V source-sense lead drop limit of the Kelvin connections, readings could be affected. If the 3V limit is exceeded, readings taken during the rising or falling edge of the pulse could be invalid. However, readings taken during the stable part of the pulse will not be affected.

On the Model 2651A, the amount of overhead in the power supply varies depending on the operating region in the power envelope (see Model 2651A datasheet at www.keithley.com/data?asset=55786 for more detail); but, in general, the amount of voltage drop caused by inductance should be kept under the 3V source-sense lead drop limit of the Kelvin connections. We can calculate the maximum amount of inductance allowed in our connections by using the equation:

$$V = L \cdot \frac{di}{dt}$$

where: V is the voltage in volts, L is the inductance in henries, and di/dt is the change in current over the change in time. If we rewrite the equation solving for L we get:

$$L = V / \frac{di}{dt}$$

As an example, let's assume that with zero inductance the Model 2651A produces a 50A pulse through our DUT with a rise time of 35µs. In order to not exceed the 3V limit while maintaining this rise time, the max amount of inductance per test lead is:

$$3V / \frac{50A}{35\mu s} = 2.1\mu H$$

In this example (35µs rise time for a 50A pulse), to not exceed the 3V limit we must ensure that our test leads have less than 2.1µH of inductance per lead.

NOTE: The Model 2651A specifications indicate a maximum inductive load of 3µH, thus the total inductance for both HI and LO leads must be less than 3µH under all conditions.

Set the Compliance

In parallel configurations, like the one shown in ***Figure 5***, the voltage limit of one SMU should be set 10% lower than the voltage limit of the other SMU. This allows only one SMU to go into compliance and become a voltage source.

Definition

An SMU, or any real current source for that matter, has a limit as to how much voltage it can output in order to deliver the desired current. When the voltage limit in an SMU is reached, the SMU goes into compliance and becomes a voltage source set to that voltage limit. When the compliance on one SMU is set lower than the compliance on the other SMU, the voltage limit can only be reached by one of the SMUs. In other words, when the SMU with the lower voltage limit goes into compliance, it becomes a voltage source with low impedance and begins to sink the current from the other SMU. With the SMU in compliance sinking current, the other SMU can now source its programmed current level and thus never go into compliance.

Setting Correct Voltage Limits

In a parallel SMU configuration, setting voltage limits properly is important. If both SMUs were to go into compliance and become voltage sources, then we would have two voltage sources in parallel. If this condition occurs, an uncontrolled amount of current could flow between the SMUs, possibly causing unexpected results and/or damage to the DUT. This condition can also occur if the DUT becomes disconnected from the test circuit. Fortunately, this condition can easily be avoided by setting the compliance for one of the SMUs lower than the compliance of the other SMU.

For example, in ***Figure 6*** we have two Model 2651As configured as 20A current sources that are connected in parallel to create a 40A current source. The voltage limit of SMU #1 is configured to 10V and the voltage limit of SMU #2 is configured to 9V and they are sourcing into a 10mΩ load. If one of the leads disconnects from the DUT during the test, each SMU would ramp up its output voltage trying to force 20A until SMU #2 reaches its voltage limit of 9V and goes into compliance. SMU #1 continues to raise its output voltage until 20A are flowing from it into SMU #2. This condition can be seen in ***Figure 7***. Because the SMUs are the same model, SMU #2 can sink the 20A current SMU #1 is delivering to it. Note that operating in this condition will cause SMU #2 to heat up quickly and will cause it to shut off if it heats up too much. This over-temperature protection is a safety feature built into the Model 2651A to help prevent accidental damage to the unit.

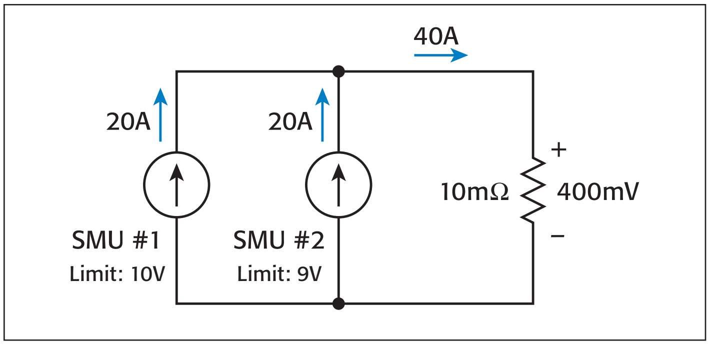


Figure 6: Example of two current source SMUs connected in parallel and functioning under normal operation.

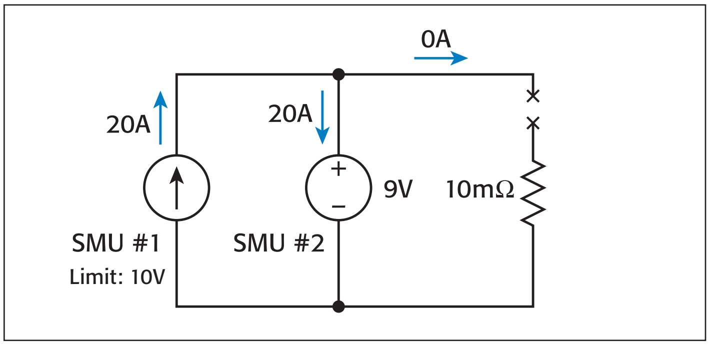


Figure 7: Example of two current source SMUs connected in parallel and functioning under compliance operation (for example, if a lead disconnects).

Set the Output Off-Mode

Introduced with the Model 2651A are new features to the NORMAL output off-mode of Series 2600A instruments. Previously, under the NORMAL output off-mode, when the output was turned off, the SMU was reconfigured as a voltage source set to 0V. This would happen whether the SMU's on state was configured as a current source or a voltage source. This is still the default configuration for the NORMAL output off-mode; however, the NORMAL output off-mode can now have its off function configured as a current source. With the off function set to current, when the output is turned off the SMU is reconfigured as a 0A current source. This happens whether the SMU's on state was configured as a current or voltage source.

When putting two SMUs configured as current sources in parallel, the SMU whose On State voltage limit is set lower should be configured using an output off-mode of NORMAL with an off function of voltage and its Off State current limit should be set to 1mA. The other SMU, whose On State voltage limit is higher, should be configured using an output off-mode of NORMAL with an off function of current and its Off State voltage limit should be set to 40V. To illustrate this, let's use ***Figure 6*** as an example. For this configuration, both SMU's output off-mode should be set to NORMAL. Also, SMU #1 should have its off function set to current with an off limit of 40V and SMU #2 should have its off function set to voltage with an off limit of 1mA. (The 40V and 1mA off limits are provided in the configuration guidelines in the reference manual of the Model 2651A.)

Setup of this new output off mode configuration is done through two new ICL commands:

- `smua.source.offfunc`
- `smua.source.offlimitv`

`smua.source.offfunc` is used to select the off function, for example:

```
smua.source.offfunc = smua.OUTPUT_DCVOLTS
-- Sets the off function to voltage

smua.source.offfunc = smua.OUTPUT_DCAMPS
-- Sets the off function to current
```

`smua.source.offlimitv` is used to set the voltage limit of the Off State configuration when the off function is current. It is similar to the command `smua.source.offlimiti`, which sets the current limit for the off state when the off function is voltage. Example usage follows:

```
smua.source.offlimitv = 10
-- Sets the off state voltage limit to 10V
```

Correctly Setting the Output Off-Mode

If you configure an SMU as a current source and do not change the off-mode, then when you turn the output off, the SMU will switch its source function from current to voltage and begin sourcing 0V. If you did not anticipate this switch, you could have a problem as the SMU essentially becomes a short to whatever is connected to it. If you had two SMUs in parallel and the SMU whose output was still on was operating as a voltage source when the other SMU's output was turned off, you would have two voltage sources in parallel, which could result in excessive current flow and could potentially damage the SMU.

Figure 7 shows what would happen if a connection to the DUT were severed. SMU #2, whose voltage limit is lower, would go into compliance and SMU #1, with a higher voltage limit, would deliver all of its current to SMU #2. If SMU #1's output were to be shut off unexpectedly and its output mode turned it into a 0V voltage source, then we would have a 0V voltage source in parallel with a 9V voltage source. In this case, SMU #2 would come out of compliance and switch back to a current source. However, uncontrolled current may flow before this switch occurs.

If SMU #1 had its output off function configured as a current source, the unexpected shut off of SMU #1's output would not have resulted in two voltage sources in parallel. Instead, SMU #1 would have simply dropped to a 0A current source. Because SMU #1's voltage limit was set higher than SMU #2's voltage limit, SMU #2 would remain in compliance but now no current would flow in the system since SMU #1 is still in control and forcing 0A.

If the opposite situation were to occur and SMU #2's output turned off unexpectedly, the situation would still be safe. SMU #2, whose off function was configured as a voltage source, would simply drop down from the 9V state to 0V. This is not a problem as SMU #1 is still a current source and holds the current to the 20A it was sourcing. The system is still not settled, however, since SMU #2 is configured with an off limit of 1mA. Because of this, SMU #2 goes into compliance, becomes a 1mA current source, and begins to raise its output voltage to try to limit current to 1mA. At this state, we have two current sources in parallel. As SMU #2 continues to ramp its output voltage, SMU #1 goes into compliance at 10V and becomes a 10V voltage source. In this state, SMU #2, a current source at this time, is in control and only 1mA of current is flowing.

Example

This example is designed to collect Rds(on) measurement data from a power MOSFET device by using a pulsed current sweep to test up to 100A, however, it can be easily modified for use in other applications.

Required Equipment

This example requires the following equipment:

- Two Model 2651A High Power System SourceMeter Instruments that will be connected in parallel to source up to 100A pulsed through the drain of the DUT
- One Model 26xxA System SourceMeter Instrument to control the gate of the DUT
- Two TSP-Link® cables for communications and precision timing between instruments

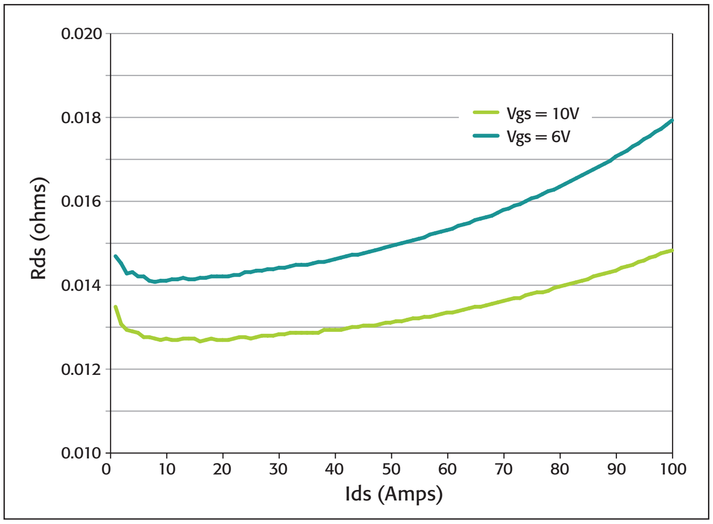


Figure 8: Example results.

- One GPIB cable or one Ethernet cable to connect the instruments to a computer

Communications Setup

The communication setup is illustrated in ***Figure 9***. GPIB is being used to communicate with the PC, but this application can be run using any of the supported communication interfaces. The TSP-Link connection enables communication between the instruments, precision timing, and tight channel synchronization.

To configure the TSP-Link communication interface, each instrument must have a unique TSP-Link node number. Configure the node number of Model 2651A #1 to 1, Model 2651A #2 to 2, and Model 26xxA to 3.

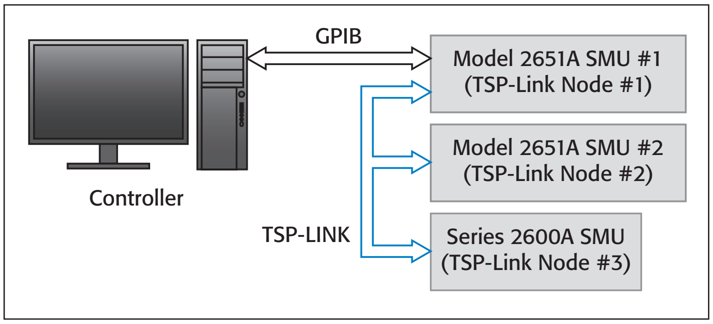


Figure 9: Communications setup for examples.

To set the TSP-Link node number using the front panel interface of either instrument:

1. Press MENU.
2. Select TSPLink.
3. Select NODE.
4. Use the navigation wheel to adjust the node number.
5. Press ENTER to save the TSP-Link node number.

On Model 2651A #1, perform a TSP-Link reset to alert Model 2651A #1 to the presence of Model 2651A #2 and Model 26xxA:

NOTE: You can also perform a TSP-Link reset from the remote command interface by sending `tsplink.reset()` to Model 2651A #1.

1. Press MENU.
2. Select TSPLink.
3. Select RESET.

NOTE: If error 1205 is generated during the TSP-Link reset, ensure that Model 2651A #2 and Model 26xxA have unique TSP-Link node numbers.

Device Connections

Connections from the SourceMeter instruments to the DUT can be seen in ***Figure 10***. Proper care should be taken to ensure good contact through all connections.

NOTE: For best results, all connections should be left floating and no connections should be tied to ground. Also, all connections should be made as close to the device as possible to minimize errors caused by voltage drops between the DUT and the points in which the test leads are connected.

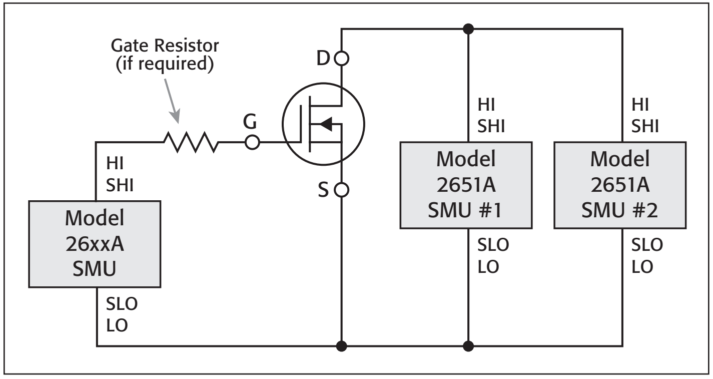


Figure 10: Connections for dual SMU Rds(on) sweep.

NOTE: During high current pulsing, the gate of your DUT may begin to oscillate, creating an unstable voltage on the gate and thus unstable current through the drain. To dampen these oscillations and stabilize the gate, a resistor can be inserted between the gate of the device and the Force and Sense Hi leads of the Model 26xxA. If the gate remains unstable after inserting a dampening resistor, enable High-C mode on the Model 26xxA (leaving the dampening resistor in place).

Configuring the Trigger Model

In order to achieve tight timing and 100A pulses with two Model 2651As, the advanced trigger model must be used. Using the trigger model, we can keep the 50A pulses of the two Model 2651As synchronized to within 500ns to provide a single 100A pulse. ***Figure 11*** illustrates the complete trigger model used in this example.

In this example, Model 2651A #1 is configured to control the overall timing of the sweep while Model 2651A #2 is configured to wait for signals from Model 2651A #1 before it can generate a pulse. The Model 26xxA is controlled by script in this example, so its trigger model is not used.

Model 2651A #1 Trigger Model Operation

In Model 2651A #1's trigger model (***Figure 12***), Timer 1 is used to control the period of the pulse while Timer 2 is used to control the pulse width. TSP-Link Trigger 1 is used to tell Model 2651A #2 to output its pulse.

When the trigger model of Model 2651A #1 is initialized, the following occurs:

1. The SMU's trigger model leaves the Idle state, flows through the Arm Layer, enters the Trigger Layer, outputs the ARMED event trigger, and then reaches the Source Event where it waits for an event trigger.
2. The ARMED event trigger is received by Timer 1, which begins its countdown and passes the trigger through to be received by TSP-Link Trigger 1, and the SMU's Source Event.
3. TSP-Link Trigger 1 receives the event trigger from Timer 1 and sends a trigger through the TSP-Link to Model 2651A #2 to instruct it to output the pulse.
4. The SMU's Source Event receives the event trigger from Timer 1, begins to output the pulse, waits the programmed source delay, if any, outputs the SOURCE_COMPLETE event to Timer 2, and then lets the SMU's trigger model continue.
5. Timer 2 receives the SOURCE_COMPLETE event trigger from Timer 1 and begins to count down.
6. The SMU's trigger model continues to the Measure Event where it waits a programmed measure delay, if any, takes a measurement, and then continues until it hits the End Pulse Event where it waits for an event trigger.
7. Timer 2's countdown expires and Timer 2 outputs an event trigger to the SMU's End Pulse Event.
8. The SMU's End Pulse Event receives the event trigger from Timer 2, outputs the falling edge of the pulse, then lets the SMU's trigger model continue.
9. The SMU's trigger model then compares the current Trigger Layer loop iteration with the trigger count.
   a. If the current iteration is less than the trigger count, then the trigger layer repeats and the SMU's trigger model reaches Source Event where it waits for another trigger from Timer 1. Because Timer 1 had its count set to the trigger count minus one, Timer 1 will continue to output a trigger for each iteration of the Trigger Layer loop. The trigger model then repeats from Step 3.
   b. If the current iteration is equal to the trigger count, then the SMU's trigger model exits the Trigger Layer, passes through the Arm Layer, and returns to the Idle state.

Model 2651A #2 Trigger Model Operation

In Model 2651A #2's trigger model (***Figure 13***), Timer 1 is used to control the pulse width and is programmed with the same delay as Model 2651A #1's Timer 2. The pulse period is controlled by TSP-Link Trigger 1, which receives its triggers from Model 2651A #1's Timer 1, thus the pulse period for Model 2651A #2 is controlled by the same timer as the Model 2651A #1.

When the trigger model of Model 2651A #2 is initialized, the following occurs:

1. The SMU's trigger model leaves the Idle state, flows through the Arm Layer, enters the Trigger Layer, and then reaches the Source Event where it waits for an event trigger.
2. TSP-Link Trigger 1 receives a trigger from TSP-Link and outputs an event trigger to the SMU's Source Event.
3. The SMU's Source Event receives the event trigger from TSP-Link Trigger 1, begins to output the pulse, waits for a programmed source delay, if any, outputs the SOURCE_COMPLETE event to Timer 1, and then lets the SMU's trigger model continue.
4. Timer 1 receives the SOURCE_COMPLETE event trigger from TSP-Link Trigger 1 and begins its countdown.
5. The SMU's trigger model continues until it reaches the Measure Event where it waits for a programmed measure delay, if any, takes a measurement, and then continues until it hits the End Pulse Event where it stops and waits for an event trigger.
6. Timer 1's countdown expires and Timer 1 outputs an event trigger to the SMU's End Pulse Event.
7. The SMU's End Pulse Event receives the event trigger from Timer 1, outputs the falling edge of the pulse, then lets the SMU's trigger model continue.
8. The SMU's trigger model compares the current Trigger Layer loop iteration with the trigger count.
   a. If the current iteration is less than the trigger count, then the trigger layer repeats and the SMU's trigger model reaches Source Event where it waits for another trigger from TSP-Link Trigger 1. The trigger model then repeats from Step 2.
   b. If the current iteration is equal to the trigger count, then the SMU's trigger model exits the Trigger Layer, passes through the Arm Layer, and returns to the Idle state.

Example Program Code

NOTE: The example code is designed to be run from Test Script Builder or TSB Embedded. It can be run from other programming environments such as Microsoft® Visual Studio or National Instruments LabVIEW®, however, modifications may be required.

The TSP script for this example contains all the code necessary to perform a pulsed Rds(on) sweep up to 100A using two Model 2651A High Power System SourceMeter instruments

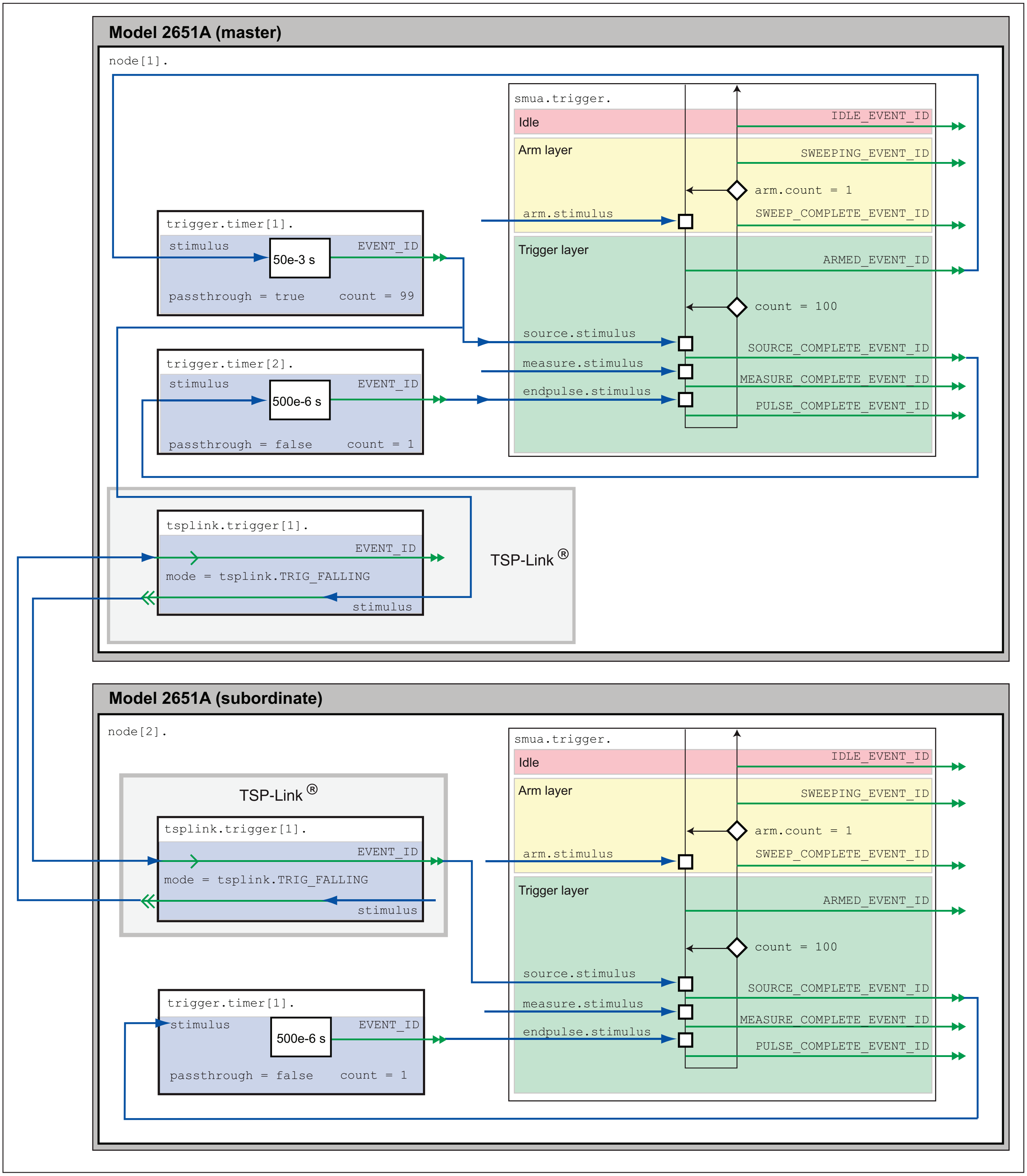


Figure 11: Example of a complete trigger model for Rds(on) sweep up to 100A.

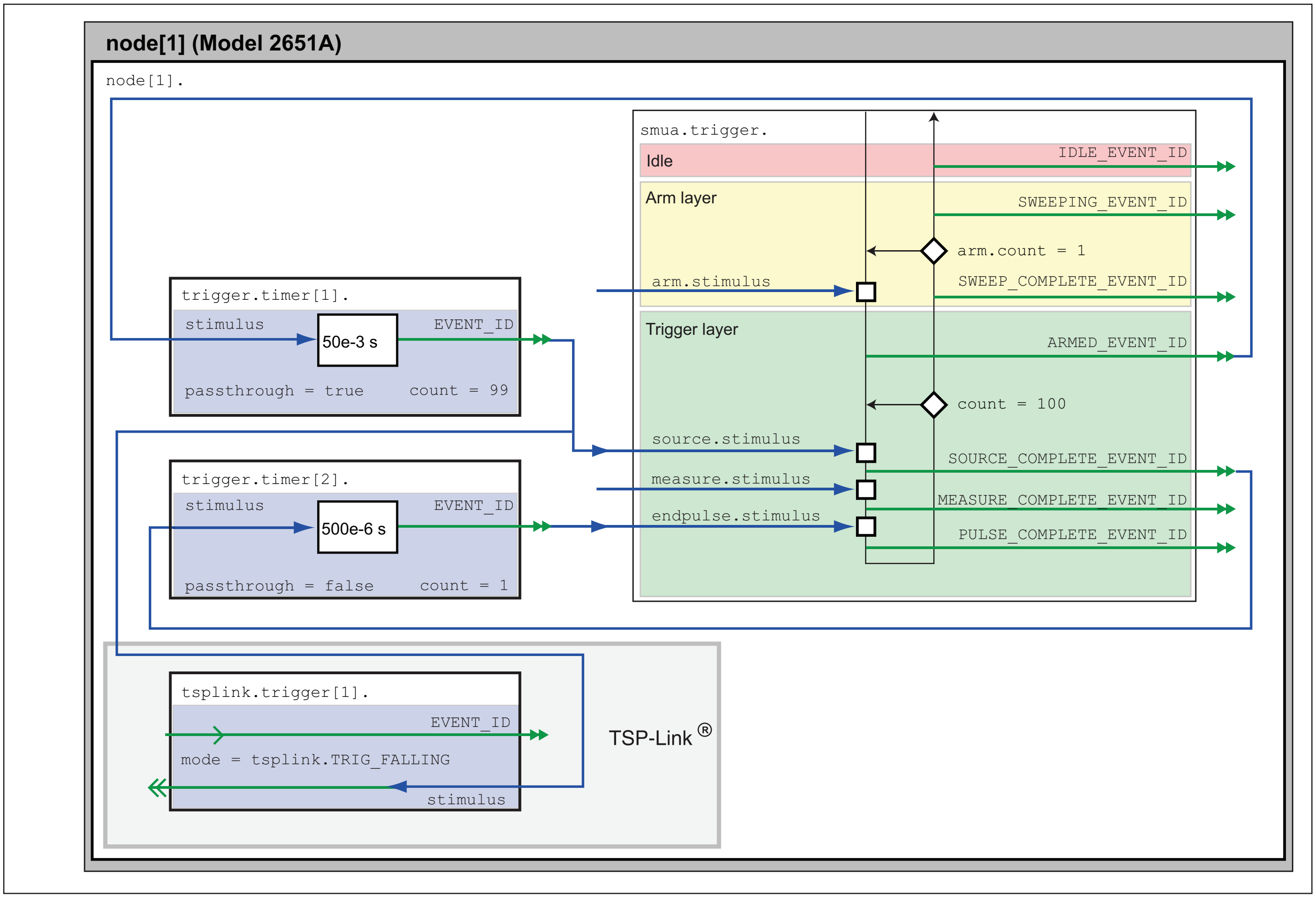


Figure 12: Example of a trigger model for 2651A #1 for Rds(on) sweep up to 100A.

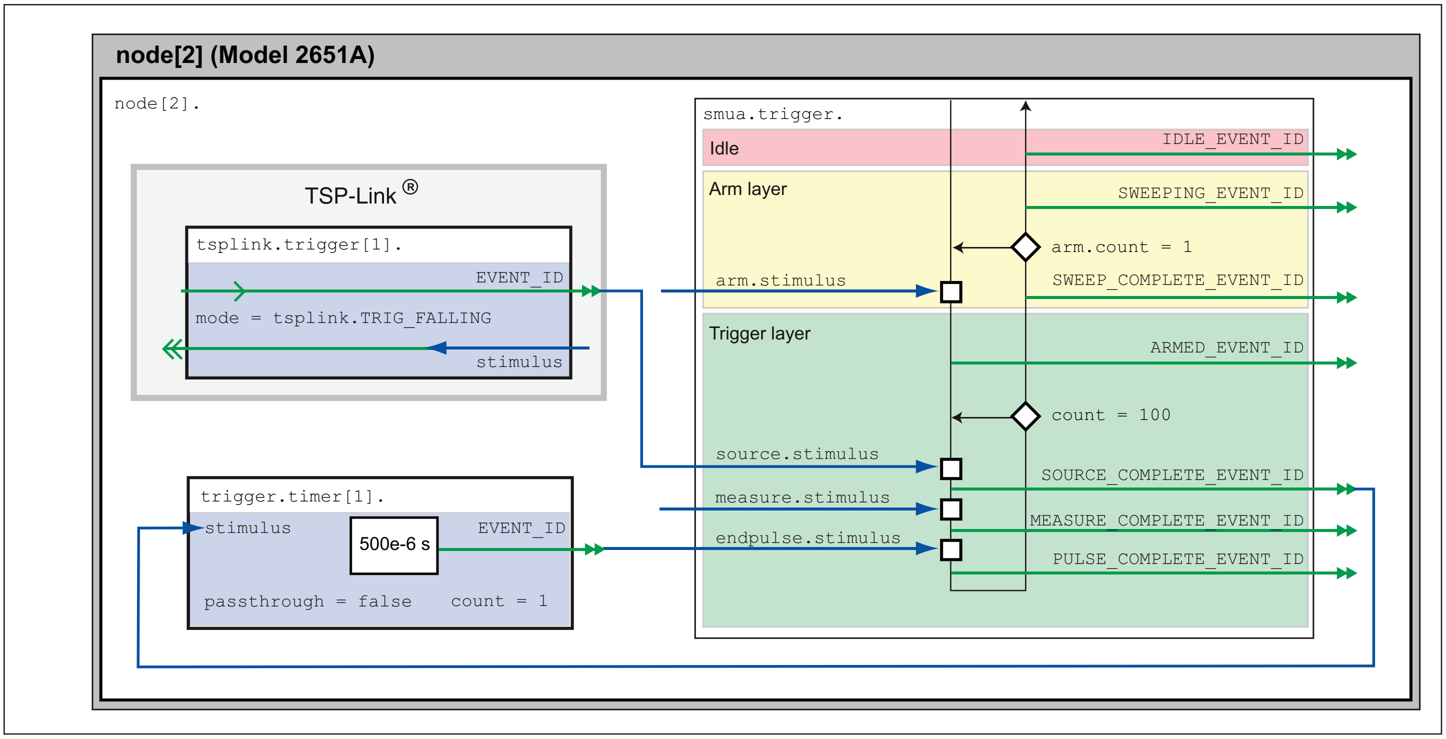


Figure 13: Example of a trigger model for 2651A #2 for Rds(on) sweep up to 100A.

and a Model 26xxA System SourceMeter instrument. This script can also be downloaded from Keithley's website at www.keithley.com/base_download?dassetid=55808.

The script performs the following functions:

- Initializes the TSP-Link connection
- Configures all the SMUs
- Configures the trigger models of the two Model 2651As
- Prepares the readings buffers
- Initializes the sweep
- Processes and returns the collected data in a format that can be copied and pasted directly into Microsoft Excel®

The script is written using TSP functions rather than a single block of inline code. TSP functions are similar to functions in other programming languages such as C or Visual Basic and must be called before the code contained in them is executed. Because of this, running the script alone will not execute the test. To execute the test, run the script to load the functions into Test Script memory and then call the functions. Refer to the documentation for Test Script Builder or TSB Embedded for directions on how to run scripts and enter commands using the instrument console.

Within the script, you will find several comments describing what is being performed by the lines of code as well as documentation for the functions contained in the script. Lines starting with

```
node[2].
```

are commands that are being sent to Model 2651A #2 through the TSP-Link interface. Lines starting with

```
node[3].
```

are commands that are being sent to the Model 26xxA through the TSP-Link interface. All other commands are executed on the Model 2651A #1.

Example Program Usage

The functions in this script are designed such that the sweep parameters of the test can be adjusted without needing to rewrite and re-run the script. A test can be executed by calling the function

```
DualSmuRdson()
```

with the appropriate values passed in its parameters.

Parameters of the function DualSmuRdson()

Parameter	Units	Description
gateLevel	Volts	Voltage level to which the gate will be held during the test
dstart	Amps	Level of the first step in the drain sweep
dstop	Amps	Level of the last step in the drain sweep
dsteps	N/A	Number of steps in the drain sweep
pulseWidth	Seconds	Width of the pulse in the drain sweep
pulsePeriod	Seconds	Time between the start of consecutive pulses in the drain sweep
pulseLimit	Volts	Voltage limit of the pulses in the drain sweep

This is an example call to function `DualSmuRdson()`.

```
DualSmuRdson(10, 1, 100, 100, 500e-6, 50e-3, 10)
```

This call sets the gate SMU output to 10V, then sweeps the drain of the DUT from 1A to 100A in 100 points. The points of this sweep will be gathered using pulsed measurements with a pulse width of 500μs and a pulse period of 50ms for a 1% duty cycle. These pulses are limited to a maximum voltage of 10V. At the completion of this sweep, all SMU outputs will be turned off and the resulting data from this test will be returned in an Excel compatible format for graphing and analysis.

Example Test Script Processor (TSP®) Script

```
--[[
   Title:   Combining SMUs for 100A Example
   Description:   This script is designed to perform an Rds(on)sweep on a power
   MOSFET device. It combines two 2651A SMUs in parallel to perform a current
   sweep up to 100A. Data collected from the sweep is then returned in a
   Microsoft Excel compatible format for plotting and analysis.

   Equipment needed:
      2x 2651A
      1x 26xxA
      2x TSP-Link Cable

   TSP-Link Configuration:
   ----------------------
   Unit          |   Node #
   2651A #1      |   1
   2651A #2      |   2
   26xxA         |   3

   Master Node (PC Interface): Node 1
]]

--[[
   Name:   DualSmuRdson(gateLevel, dstart, dstop, dsteps, pulseWidth,
   pulsePeriod, pulseLimit)
   Description:   This function uses two 2651A SMUs to perform a pulsed Rds(on)
   sweep with currents up to 100A.

   Parameters:
      gateLevel:   The gate level to be used during the sweep
      dstart:      The starting current level of the drain sweep
      dstop:       The ending current level of the drain sweep
      dsteps:        The number of steps in the drain sweep
      pulseWidth:  The width of the drain pulse in seconds
      pulsePeriod: The time from the start of one drain pulse to
         the next in seconds
      pulseLimit:  The voltage limit of the drain pulse in volts
         Note: Actual pulse limit will be 10% lower than setting
         to protect SMUs in a compliance condition

   Example Usage:
      DualSmuRdson(10, 1, 100, 100, 500e-6, 50e-3, 10)
]]

function DualSmuRdson(gateLevel, dstart, dstop, dsteps, pulseWidth, pulsePeriod, pulseLimit)
   tsplink.reset(3) -- Verify that at least three nodes are present
   reset()

   -- Configure 2651A #1 (Drain SMU 1)
   ---------------------------------
   smua.reset()
   smua.source.func            = smua.OUTPUT_DCAMPS
   smua.sense                  = smua.SENSE_REMOTE
   smua.source.offmode = smua.OUTPUT_NORMAL
   smua.source.offfunc = smua.OUTPUT_DCVOLTS
   smua.source.offlimiti       = 1e-3  -- Set off limit

  -- SMU #1 will be a 0V voltage source with 1mA limit when its
   -- output is turned off. SMU #2 will be a 0A current source with
   -- a 10V limit when the output is turned off. These settings keep
   -- the parallel combination safe in case one SMU is turned off.

   smua.source.rangei = math.max(math.abs(dstart / 2), math.abs(dstop / 2))
   smua.source.leveli          = 0            -- Sets the DC bias level
   smua.source.limitv          = 9            -- Sets the DC bias limit
   -- SMU #2 will have a voltage limit of 10V. By setting the voltage
   -- limit 10% lower than that of SMU #2, we can ensure that only
   -- one of the two SMUs will ever go into compliance and become a
   -- voltage source. This is desirable, because if both SMUs went
   -- into compliance, there would be two voltage sources in parallel,
   -- which is an unsafe condition.
```

```
 smua.measure.nplc           = 0.005
 smua.measure.rangev         = pulseLimit
 smua.measure.autozero               = smua.AUTOZERO_ONCE
 smua.measure.delay              =
        pulseWidth - ((1 / localnode.linefreq) * smua.measure.nplc)) - 20e-6

-- Set the delay so that the measurement is near the end of the pulse

 -- Prepare the reading buffers
 smua.nvbuffer1.clear()
 smua.nvbuffer1.collecttimestamps    = 1
 smua.nvbuffer1.collectsourcevalues  = 1
 smua.nvbuffer1.fillmode                     = smua.FILL_ONCE
 smua.nvbuffer2.clear()
 smua.nvbuffer2.collecttimestamps    = 1
 smua.nvbuffer2.collectsourcevalues  = 1
 smua.nvbuffer2.fillmode                     = smua.FILL_ONCE

 -- Configure TSP-Link Trigger 1
 tsplink.trigger[1].clear()
 tsplink.trigger[1].mode                     = tsplink.TRIG_FALLING
 tsplink.trigger[1].stimulus         = trigger.timer[1].EVENT_ID
 -- TSP-Link Trigger 1 signals 2651A #2 to pulse

-- Timer 1 controls the pulse period by triggering the pulse to begin
 trigger.timer[1].count             = dsteps - 1
 trigger.timer[1].delay             = pulsePeriod
 trigger.timer[1].passthrough  = true
 trigger.timer[1].stimulus      = smua.trigger.ARMED_EVENT_ID
 trigger.timer[1].clear()

 -- Timer 2 controls the pulse width
 trigger.timer[2].count             = 1
 trigger.timer[2].delay             = pulseWidth - 3e-6
 trigger.timer[2].passthrough  = false
 trigger.timer[2].stimulus      = smua.trigger.SOURCE_COMPLETE_EVENT_ID
 trigger.timer[2].clear()

-- Configure SMU Trigger Model for Sweep
 -- Each unit will source half the current, so divide the start
 -- and stop values by 2
 smua.trigger.source.lineari(dstart / 2, dstop / 2, dsteps)
 smua.trigger.source.limitv   = pulseLimit - (pulseLimit * 0.1)
 -- Again, keep the limit SMU #1 lower than the limit of SMU #2
 -- to prevent parallel V-sources
 smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2)
 smua.trigger.measure.action  = smua.ENABLE

 -- Return to the bias level at the end of the pulse/sweep
 smua.trigger.endpulse.action = smua.SOURCE_IDLE
 smua.trigger.endsweep.action = smua.SOURCE_IDLE

 smua.trigger.count           = dsteps
 smua.trigger.arm.stimulus    = 0
 smua.trigger.source.stimulus = trigger.timer[1].EVENT_ID
 smua.trigger.measure.stimulus        = 0
 smua.trigger.endpulse.stimulus = trigger.timer[2].EVENT_ID
 smua.trigger.source.action   = smua.ENABLE

-- Configure 2651A #2 (Drain SMU 2)
 -----------------------------------
 node[2].smua.reset()
 node[2].smua.source.func    = node[2].smua.OUTPUT_DCAMPS
 node[2].smua.sense          = node[2].smua.SENSE_REMOTE
 node[2].smua.source.offmode = node[2].smua.OUTPUT_NORMAL
 node[2].smua.source.offfunc = node[2].smua.OUTPUT_DCAMPS
 node[2].smua.source.offlimitv       = 10    -- Set off limit
 -- SMU will be a 0A current source with 10V limit when output is turned off
 node[2].smua.source.rangei     =
         math.max(math.abs(dstart / 2), math.abs(dstop / 2))
 node[2].smua.source.leveli  = 0     -- Sets the DC bias level
 node[2].smua.source.limitv  = 10    -- Sets the DC bias limit

node[2].smua.measure.nplc    = 0.005
```

```
 node[2].smua.measure.rangev = pulseLimit
 node[2].smua.measure.autozero       = node[2].smua.AUTOZERO_ONCE
 node[2].smua.measure.delay  = (pulseWidth -
        ((1 / node[2].linefreq) * node[2].smua.measure.nplc)) - 20e-6
 -- Set the delay so that the measurement is near the end of the pulse

-- Prepare the reading buffers
 node[2].smua.nvbuffer1.clear()
 node[2].smua.nvbuffer1.collecttimestamps      = 1
 node[2].smua.nvbuffer1.collectsourcevalues    = 1
 node[2].smua.nvbuffer1.fillmode               = node[2].smua.FILL_ONCE
 node[2].smua.nvbuffer2.clear()
 node[2].smua.nvbuffer2.collecttimestamps      = 1
 node[2].smua.nvbuffer2.collectsourcevalues    = 1
 node[2]. smua.nvbuffer2.fillmode              = node[2].smua.FILL_ONCE

 -- Configure TSP-Link Trigger 1
 node[2].tsplink.trigger[1].clear()
 node[2].tsplink.trigger[1].mode      = node[2].tsplink.TRIG_FALLING

 -- Timer 1 controls the pulse width
 node[2].trigger.timer[1].count              = 1
 node[2].trigger.timer[1].delay              = pulseWidth - 3e-6
 node[2].trigger.timer[1].passthrough        = false
 node[2].trigger.timer[1].stimulus    =
       node[2].smua.trigger.SOURCE_COMPLETE_EVENT_ID
 node[2].trigger.timer[1].clear()

 -- Configure SMU Trigger Model for Sweep
 node[2].smua.trigger.source.lineari(dstart / 2, dstop / 2, dsteps)
 node[2].smua.trigger.source.limitv          = pulseLimit
 node[2].smua.trigger.measure.iv(node[2].smua.nvbuffer1, node[2].smua.nvbuffer2)
 node[2].smua.trigger.measure.action         = node[2].smua.ENABLE

-- Return the output to the bias level at the end of the pulse/sweep
 node[2].smua.trigger.endpulse.action   = node[2].smua.SOURCE_IDLE
 node[2].smua.trigger.endsweep.action   = node[2].smua.SOURCE_IDLE
 node[2].smua.trigger.count            = dsteps
 node[2].smua.trigger.arm.stimulus     = 0
 node[2].smua.trigger.source.stimulus   = node[2].tsplink.trigger[1].EVENT_ID
 node[2].smua.trigger.measure.stimulus  = 0
 node[2].smua.trigger.endpulse.stimulus = node[2].trigger.timer[1].EVENT_ID
 node[2].smua.trigger.source.action    = node[2].smua.ENABLE

-- Configure the 26xxA (Gate SMU)
 --------------------------------
 node[3].smua.reset()
 node[3].smua.source.func    = node[3].smua.OUTPUT_DCVOLTS
 node[3].smua.sense          = node[3].smua.SENSE_REMOTE
 node[3].smua.source.levelv  = gateLevel
 node[3].smua.source.highc   = node[3].smua.ENABLE
 -- If you find your gate oscillating even with a dampening resistor
 -- in place, try enabling high-C mode to help stabilize the gate.

 -- Prepare the reading buffers
 node[3].smua.nvbuffer1.clear()
 node[3].smua.nvbuffer1.collectsourcevalues  = 1
 if node[3].smua.nvbuffer1.fillmode           ~= nil then
    node[3].smua.nvbuffer1.fillmode           = node[3].smua.FILL_ONCE
 end
 node[3].smua.nvbuffer2.clear()
 node[3].smua.nvbuffer2.collectsourcevalues  = 1
 if node[3].smua.nvbuffer2.fillmode           ~= nil then
    node[3].smua.nvbuffer2.fillmode           = node[3].smua.FILL_ONCE
 end

-------------------------
 -- Ready to begin the test
 -------------------------
 -- Outputs on
 node[3].smua.source.output          = node[3].smua.OUTPUT_ON
 node[2].smua.source.output          = node[2].smua.OUTPUT_ON
 smua.source.output          = smua.OUTPUT_ON
```

```
   if errorqueue.count > 0 then
      print(“Errors were encountered”)
      reset()
      return
   end
   -- Give the gate some time to settle before starting the sweep
   delay(0.001)
   node[3].smua.measure.iv(node[3].smua.nvbuffer1, node[3].smua.nvbuffer2)

   -- Start the 2651A #2 trigger model
   node[2].smua.trigger.initiate()
   -- Start the 2651A #1 trigger model
   smua.trigger.initiate()
   -- Wait until test is complete
   waitcomplete()

   -- Outputs off
   node[3].smua.source.output          = node[3].smua.OUTPUT_OFF
   smua.source.output          = smua.OUTPUT_OFF
   node[2].smua.source.output          = node[2].smua.OUTPUT_OFF

   -- Print back data
   PrintDualSmuRdsonData()
end

--[[
   Function:   PrintDualSmuRdsonData()
   Description:
      This function processes the data stored in the SMU reading buffers by
   function DualSmuRdson() and prints back the individual SMU data and the
   combined SMU data and Rds(on) readings in a format that is copy and paste
   compatible with Microsoft Excel.
]]
function PrintDualSmuRdsonData()
   -- Print the gate SMU readings
   print(“Gate SMU\r\nSource Value\tVoltage\tCurrent”)
   print(string.format(“%0.2f\t%g\t%g\r\n”,
      node[3].smua.nvbuffer1.sourcevalues[1],
      node[3].smua.nvbuffer2[1],
       node[3].smua.nvbuffer1[1]))

   -- Print column headers
   print(“Timestamp\tSource Value\tVoltage 1\tCurrent 1\tVoltage
      2\tCurrent 2\tVoltage\tCurrent\tRds(on)”)
   -- Loop through the reading buffer printing one row at a time
   for i = 1,smua.nvbuffer1.n do
       -- Combined Source Level = SMU1 source level + SMU2 source level
      sourceLevel = smua.nvbuffer1.sourcevalues[i] +
         node[2].smua.nvbuffer1.sourcevalues[i]

  -- Combined Voltage = Average(SMU1 Voltage reading, SMU2 Voltage reading)
   combinedVoltage = (smua.nvbuffer2[i] + node[2].smua.nvbuffer2[i]) / 2

   -- Combined Current = SMU1 Current reading + SMU2 Current reading
   combinedCurrent = smua.nvbuffer1[i] + node[2].smua.nvbuffer1[i]

   -- Rds(on) = Combined Voltage / Combined Current
   rdson = combinedVoltage / combinedCurrent

   -- Print a row of data
   print(string.format(“%g\t%g\t%g\t%g\t%g\t%g\t%g\t%g\t%g”,
      smua.nvbuffer1.timestamps[i],
      sourceLevel,
      smua.nvbuffer2[i],
      smua.nvbuffer1[i],
      node[2].smua.nvbuffer2[i],
      node[2].smua.nvbuffer1[i],
      combinedVoltage,
      combinedCurrent,
      rdson))
   end
end
```

Optimizing Reliability Testing of Power Semiconductor Devices and Modules with Keithley SMU Instruments and Switch Systems

Introduction

To minimize early defect rates and to continuously improve the overall reliability and lifetime of power semiconductors, a variety of important tests are performed by both manufacturers and end-use designers. Many of these tests are outlined in JEDEC Standards such as JESD22-A108D "Temperature, Bias, and Operating Life," JESD22-A110D "Highly Accelerated Temperature and Humidity Stress Test (HAST)," or JESD236 "Reliability Qualification of Power Amplifier Modules." This application brief discusses methods to optimize reliability testing of silicon and wide band gap (WBG) power semiconductor devices, modules, and materials by using Keithley SourceMeter® Source Measure Unit (SMU) Instruments and Switch Systems (***Figures 1 and 2***).

KEITHLEY
+50.0000 A
SrcA:+10.0000 V LimA:20.0000 A
2651A HIGH POWER SYSTEM SourceMeter® LXI
KEITHLEY
+0.01000nA
SrcA:+3.0000kV LimA:1.00000mA
2657A HIGH POWER SYSTEM SourceMeter® LXI

Figure 1. Keithley Series 2650A High Power SourceMeter SMU Instruments.

Figure 2. Keithley Series 3700A and 707B Series Switch Systems.

Typical Reliability Tests

Typical reliability tests involve stressing a batch or batches of sample devices for hundreds or thousands of hours with bias voltages that are greater than or equal to their normal operating voltages while subjecting them to temperatures that are well beyond normal operating conditions. During this stress, a variety of key operating parameters are measured at specific time intervals. Some of the more popular reliability tests for power semiconductors are HTOL (High Temperature Operating Life), ELFR (Early Life Failure Rate), HTFB (High Temperature Forward Bias), HTRB (High Temperature Reverse Bias), and HAST (Highly Accelerated Temperature & Humidity Stress Test). These tests will either use a continuous bias (***Figure 3***) or cycled bias (***Figure 4***). A continuous bias can be a fixed voltage or a staircase ramp. A cycled bias will typically vary the duty cycle and/or frequency of the bias voltage. In both cases, key device parameters will be tested continuously or at specific time intervals.

Reliability Testing Challenges

Reliability testing of today's WBG power semiconductors presents several key challenges for engineers and test system designers. Most importantly, since most of these devices are being targeted for energy-efficiency applications, they have much lower leakage and on-resistance specifications compared to traditional silicon. The test instrumentation must therefore be capable of providing the necessary accuracy, resolution, and stability to meet the electrical requirements of these devices. In addition, since WBG devices exhibit failure mechanisms that are different from silicon, effective reliability testing per JEDEC standards requires larger sample sizes and longer stress durations to adequately predict important reliability parameters. This requires test instrumentation that is capable of supplying enough power to test many devices in parallel, while maintaining the accuracy and resolution mentioned above. Finally, the test instrumentation must be able to respond to the high speed behaviors associated with these devices and produce the masses of data associated with testing devices in parallel. Each instrument in the system must be fast, and all units must operate in a highly synchronized manner.

Keithley Solutions

Keithley reliability test solutions based on the Series 2600B and 2650A System SourceMeter SMU Instruments meet these demanding requirements (***Figure 5***). Keithley's Series 2600B and 2650A are modular, independent, isolated SMUs that provide up to 200W per channel with measurement resolutions to sub-pA levels.

In addition, all Series 2600B and 2650A SMU Instruments contain Keithley's Test Script Processor (TSP®) and TSP-Link® technology for ultra- high speed operation and parallel test.

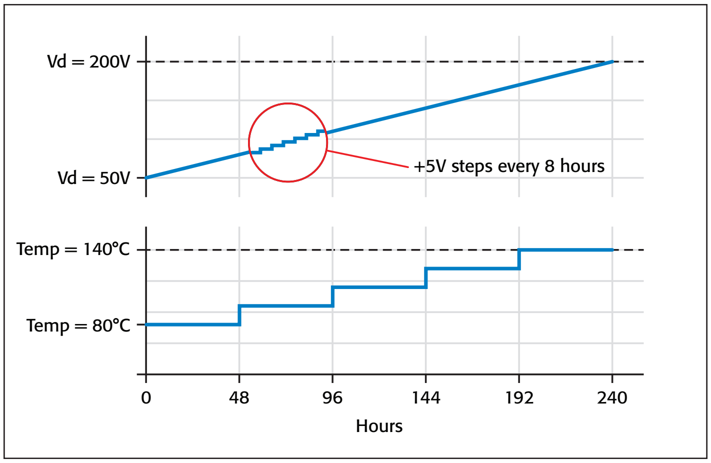


Figure 3. Continuous Bias Test. Vd=50V. Step temperature 15°C every 48 hours from 80°C to 140°C. Step Vd 5V every 8 hours from 50V to 200V. Collect I-V curve data before and after each voltage step. Test 72 devices per batch.

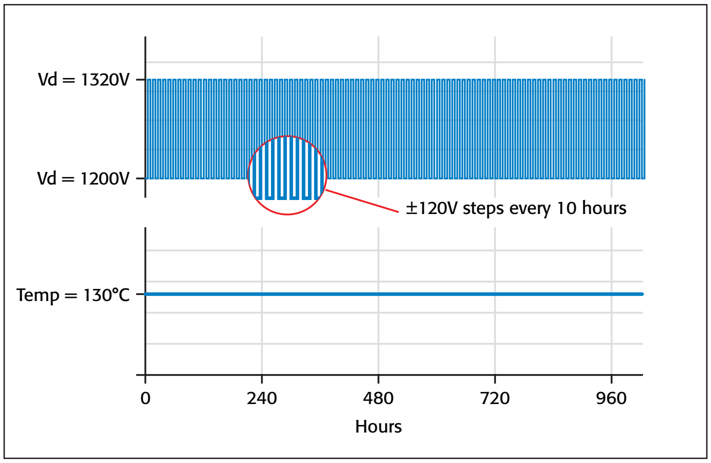


Figure 4. Cycled Bias Test. Vd cycles between 1200V to 1320V every 10 hours with a 50% duty cycle. Maintain temperature of 130°C for 1000 hours, and continuously measure Id. Test 240 devices per batch.

Keithley's TSP technology enables the instrument to perform advanced tests without PC intervention through the use of embedded test scripts. These scripts are complete test subroutines executed from the instrument's non-volatile memory that can perform conditional branching, flow control, advanced calculations, pass-fail testing, and more. On-board memory buffers can store over 140,000 readings, and can be queried while data is being written to them. For larger, multi-channel applications, Keithley's TSP-Link technology works together with TSP technology to enable high-speed, multi-channel, SMU-per-pin parallel testing or seamless integration with Keithley's Series 3700A or Series 700 Switch Mainframes (***Figure** 6*). The Keithley Series 3700A Mainframes also have a temperature measurement option that can be used for monitoring internal oven temperature at multiple locations. Synchronization with external equipment such as temperature chambers is enabled by the built-in digital I/O capability. Because Series 2600B and 2650A instruments have fully isolated channels that do not require a mainframe, they can easily be reconfigured and re-deployed as reliability test requirements evolve.

Solution Details

Reliability test algorithms are typically very specific to the device under test (DUT) and its market requirements. In this application brief, we are referring to the cycled bias test example in ***Figure** 4*.

This test cycles the drain voltage (Vd) from 1200V to 1320V every 10 hours with a 50% duty cycle, and maintains a temperature of 130°C for 1000 hours while continuously measuring the devices' off-state leakage current (Id). A batch size of 240 is tested, so the most economical approach is to use a switch system to route the power of one SMU instrument to all DUTs.

An example system block diagram is shown in ***Figure** 7**. The Keithley Model 2657A High Voltage SMU Instrument provides power to the drain terminal of all 240 DUTs, which are connected in parallel. Maximum current available from the Model 2657A at voltages of 1200V–1320V is 120mA, so there is plenty of capacity. In fact, the Model 2657A can supply power to multiple batches. Current limiting resistors are placed in series with each DUT in case of device failure.

The Keithley Model 2635B SMU Instrument is used to measure the leakage current (Id) of each device at the source terminal, and is connected to each device through the Keithley Model 3706A Switch System with Model 3720 Switch Cards. A low-leakage diode is also placed at the source terminal in parallel with each switch. When a given switch is open, any leakage current from the DUT will flow through this diode. When a given switch is closed, the leakage current for that DUT will flow through the Keithley Model 2635B (since its output is 0V when measuring current). The Model 3720 Switch Cards are configured

Model 2657A: 180W, High Voltage

- Source and measure up to 3000V @ 20mA or 1500V @ 120mA.
- Basic voltage source accuracy of 0.03% with 5mV programming resolution.
- Basic current measure accuracy of 0.02%. 1fA resolution.

Model 2636B: Dual Channel, Low Current

- Source and measure up to 200V @ 100mA or 20V @ 1.5A per channel.
- Basic voltage source accuracy of 0.03% with 5μV programming resolution.
- Basic current measure accuracy of 0.02%. 0.1fA resolution.

Model 2651A: 200W, High Current

- Source and measure up to 40V @ 5A or 10V @ 20A
- Basic voltage source accuracy of 0.02% with 5μV programming resolution.
- Basic current measure accuracy of 0.02%. 1pA resolution.

Figure 5. Key specifications of Keithley SourceMeter SMU instruments.

Model 3706A: High Speed Switch System

- Up to 576 channels of 2-pole switching
- Multiplexer or matrix configurations with 300V, 1A capability
- Built-in temperature measurement option.

Model 707B: Low Current Switch System

- Up to 576 matrix cross-points
- 1300V, 1A switching with <1pA offset
- 200V, 2A switching with <100fA offset

Figure 6. Key specifications of Keithley switch systems.

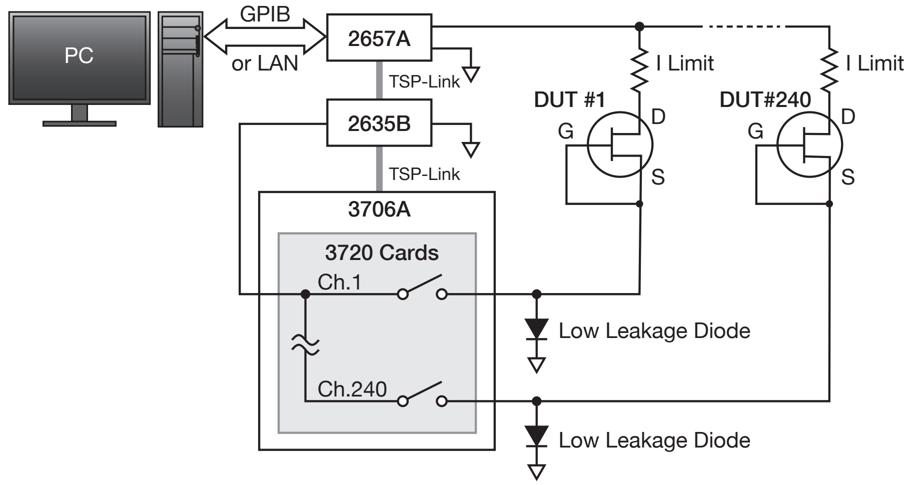


Figure 7.

*This example is for illustrative purposes only. Full test system design and adherence to proper safety guidelines and procedures are the responsibility of the designer and operator.

for single-pole switching, and each card has a 60-channel capacity; therefore, four switch cards are used (the Model 3706A Switch System chassis has a total capacity of six switch cards). The scanning speed of the Model 3720 Switch Card is 120 channels per second.

For maximum system speed and programming simplicity, the TSP-Link intercommunication bus is used to connect the three instruments together. Only one connection is needed to the PC controller, and this is through the Model 2657A over GPIB or LAN (LXI). The PC controller contains the main test program that calls the TSP scripts (subroutines) that reside inside the Model 2657A's non-volatile memory. Keithley's TSP technology enables all instrument control and most data management to be performed at the instrument level, thus eliminating the typical GPIB or LAN traffic delays that slow system-level throughput in instrument-based systems.

In this example, Keithley's Series 2600B and 2650A System SourceMeter SMU Instruments and Series 3700A Switch Systems are used to meet the accuracy, resolution, power, and speed requirements of today's semiconductor reliability test applications.

V_{DS} Ramp and HTRB Reliability Testing of High Power Semiconductor Devices with Automated Characterization Suite (ACS) Software

Introduction

Wide bandgap semiconductor materials such as silicon carbide (SiC) and gallium nitride (GaN) offer physical properties superior to those of silicon (Si) for power device applications, enabling devices based on these materials to withstand high voltages and temperatures, as well as permitting higher frequency response, greater current density, and faster switching [1]. These emerging power devices have great potential; however, the technologies necessary to create and refine them are still under development and therefore less mature than silicon technology. This creates some big challenges associated with designing and characterizing these devices, as well as process monitoring and reliability issues [2].

Before they can gain commercial acceptance, the reliability of wide bandgap devices must be proven and there is a demand for higher reliability requirements. The continuous drive for greater power density at the device and package levels creates consequences in terms of higher temperatures and temperature gradients across the package. New application areas often mean more severe ambient conditions. For example, in automotive hybrid traction systems, the cooling liquid for the combustion engine may reach temperatures as high as 120°C. In order to provide sufficient margin, this means the maximum junction temperature (T_{JMAX}) must be increased from 150°C to 175°C [4]. In safety-critical applications such as aircraft, the zero-defect concept has been proposed to meet stricter reliability requirements.

V_{DS} Ramp and HTRB Reliability Tests

The V_{DS} ramp and the High Temperature Reverse Bias (HTRB) tests are among the most common reliability tests for power devices. In a V_{DS} ramp test, as the drain-source voltage is stepped from a low voltage to a voltage that's higher than the rated maximum drain-source voltage, specified device parameters are evaluated. The test is useful for tuning the design and process conditions, as well as verifying that devices deliver the performance specified on their data sheets. For example, Dynamic $R_{DS(ON)}$, monitored using a V_{DS} ramp test, provides a measurement of how much a device's ON-resistance increases after being subjected to a drain bias [5]. Although a V_{DS} ramp test is generally used as a quick form of parametric verification, an HTRB test evaluates long-term stability under high drain-source bias. During an HTRB test, the device samples are stressed at or slightly less than the maximum rated reverse breakdown voltage (usually 100% or 80% of V_{RRM}) at an ambient temperature close to their maximum rated junction temperature (T_{JMAX}) over a period of time (usually 1,000 hours)[3][5][6][7]. The leakage current is continuously monitored throughout the test and a fairly constant leakage current is generally required to pass the test. Because it combines electrical and thermal stress, this test can be used to check the junction integrity, crystal defects and ionic-contamination level, which can reveal weaknesses or degradation effects in the field depletion structures at the device edges and in the passivation [8].

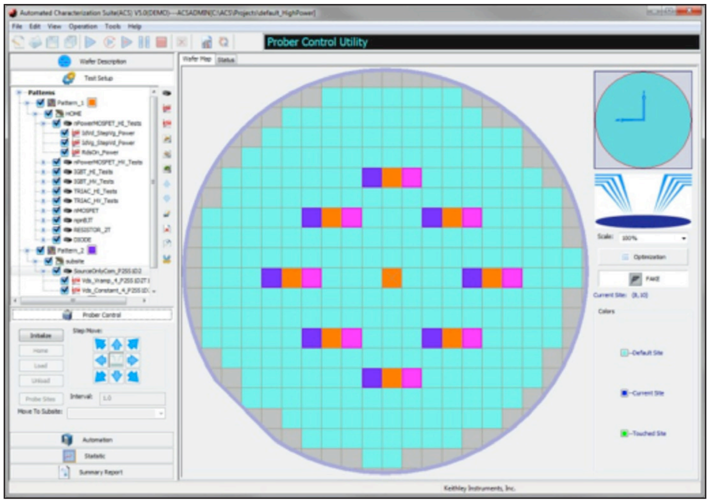


Figure 1. Automated Characterization Suite (ACS) graphical user interface

Test Instrumentation and Measurement Considerations

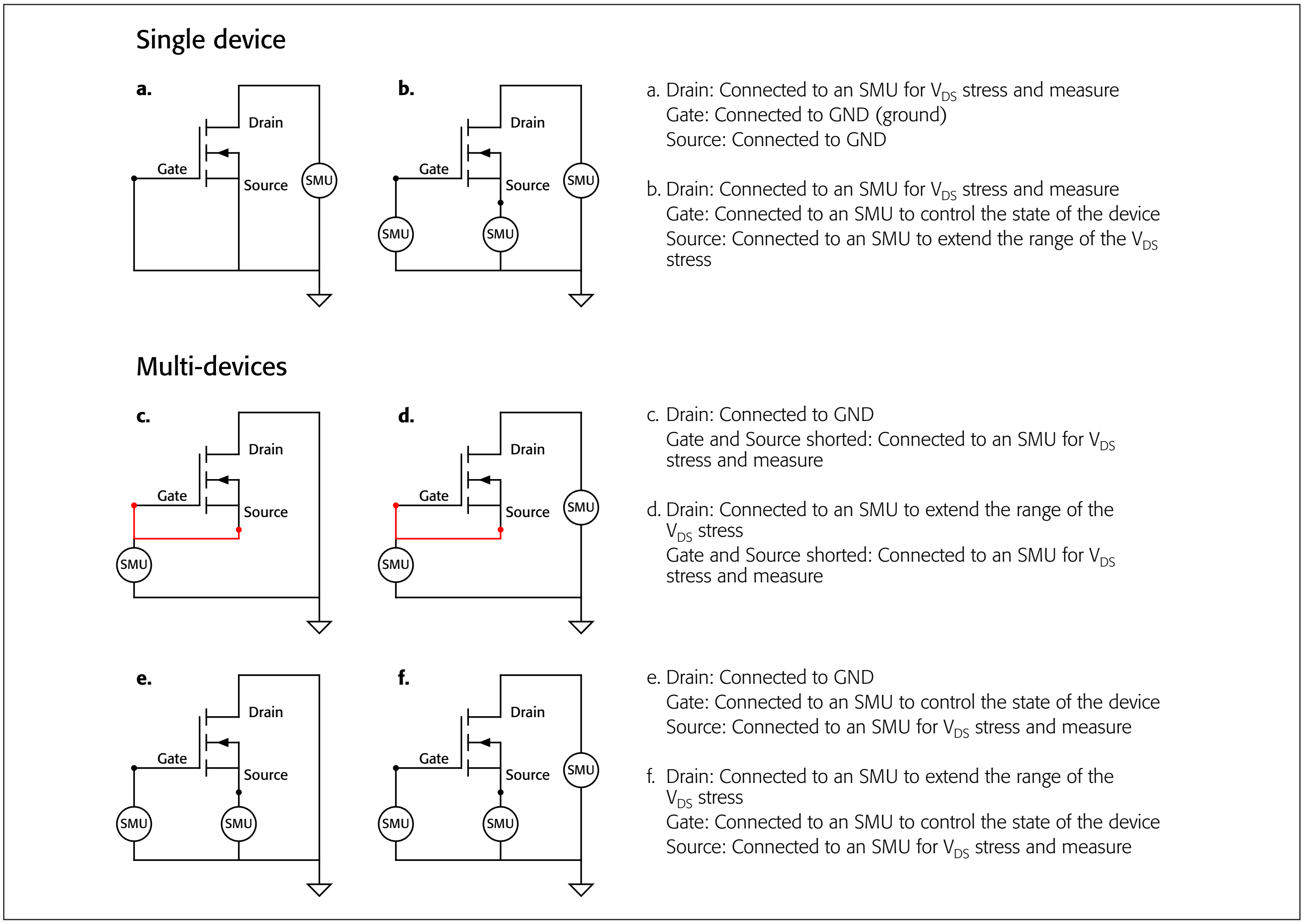
Power device characterization and reliability testing require test instrumentation with higher voltage as well as more sensitive current measurement capability than ever before [2]. During operation, the devices undergo both electrical and thermal stress: when in the ON state, they have to pass tens or hundreds of amps with minimal loss (low voltage, high current); when they are OFF, they have to block thousands of volts with minimal leakage currents (high voltage, low current). Additionally, during the switching transient, they are subjected to a brief period of both high voltage and high current. The high current experienced during the ON state generates a large amount of heat, which may degrade device reliability if it is not dissipated efficiently [1].

Reliability tests typically involve high voltages, long test times, and often multiple devices under test (wafer level testing). As a result, well-designed test systems and measurement plans are essential to avoid breaking devices, damaging equipment, and losing test data. Consider the following factors when executing V_{DS} ramp and HTRB reliability tests:

- *Device connections:* When testing vertical devices with a common drain, proper connections are required to prevent stress termination in case of a single device breakdown.
- *Current limit control:* Current limit should allow for adjustment at breakdown to avoid damage to the probe card and device.
- *Stress control:* The high voltage stress must be well controlled to avoid overstressing the device, which can lead to unexpected device breakdown.
- *Proper test abort design:* The test program must be designed in a way that allows the user to abort the test (that is, terminate the test early) without losing the data already acquired.
- *Data management:* Effective data collection is essential to accommodate the large datasets due to long time and multi-site testing

A comprehensive hardware and software solution is essential to address these test considerations effectively. Keithley's Automated Characterization Suite (ACS) software supports semiconductor characterization at the device, wafer, and cassette levels. It offers users maximum flexibility for performing applications, so they can switch easily between manual operation for lab use and fully automated operation for production settings, using the same test plan. Its integrated test plan and wafer description function allow setting up single or multiple test plans on one wafer and selectively executing them later, either manually or automatically. ACS is compatible with many of Keithley's most advanced Source Measure Unit (SMU) instruments, including the Model 2636B (capable of sourcing up to 200V and measuring with 0.1fA resolution) and the high power Model 2657A (capable of sourcing up to 3kV and measuring with 1fA resolution).

Depending on the needs of the test environment, ACS systems can be configured to create anything from a simple system with a few instruments on a benchtop to an integrated, fully automated rack of instruments on a production floor. In addition to controlling Keithley SourceMeter® SMUs, other instruments, and switching hardware, ACS can control most standard automatic probers, execute the tests and return the resulting data for parameter extraction and analysis, all through a graphical user interface ***(Figure 1)***. The V_{DS} breakdown reliability test module for V_{DS} ramp and HTRB testing described in this application note is included in ACS Version 5.0 [9].



a. Drain: Connected to an SMU for V_{DS} stress and measure
Gate: Connected to GND (ground)
Source: Connected to GND

b. Drain: Connected to an SMU for V_{DS} stress and measure
Gate: Connected to an SMU to control the state of the device
Source: Connected to an SMU to extend the range of the V_{DS} stress

c. Drain: Connected to GND
Gate and Source shorted: Connected to an SMU for V_{DS} stress and measure

d. Drain: Connected to an SMU to extend the range of the V_{DS} stress
Gate and Source shorted: Connected to an SMU for V_{DS} stress and measure

e. Drain: Connected to GND
Gate: Connected to an SMU to control the state of the device
Source: Connected to an SMU for V_{DS} stress and measure

f. Drain: Connected to an SMU to extend the range of the V_{DS} stress
Gate: Connected to an SMU to control the state of the device
Source: Connected to an SMU for V_{DS} stress and measure

Figure 2. Various device connection options used with V_{DS} breakdown test module

ACS V_{DS} Breakdown Test Module

ACS's V_{DS} breakdown test module applies two different stress tests across the drain and source of the MOSFET structure (or across the collector and emitter of an IGBT) for V_{DS} ramp and HTRB reliability assessment:

Vds_Vramp – Applies ramped stress. This test sequence can be used as a quick method of parametric verification. The devices tested are evaluated on whether they meet the performance specifications listed on their data sheets.

Vds_Constant – Applies constant stress. This test sequence can be set up for reliability testing over an extended period and at elevated temperature, such as an HTRB test.

Device Connection

Depending on the number of instruments and devices, or the probe card type, users can implement various connection schemes ***(Figure 2)*** to achieve the desired stress configurations. When testing a single device, a user can apply voltage at the drain only for V_{DS} stress and measure, which requires only one SMU per device. Alternatively, a user can connect each gate and source to a SMU for more control in terms of measuring current at all terminals, extend the range of V_{DS} stress, and set voltage on the gate to simulate a practical circuit situation. For example, to evaluate the device in the OFF state (including HTRB test), V_{GS} might be set to $V_{GS} < 0$ for an N-channel depletion device, $V_{GS} > 0$ for a P-channel device, or $V_{GS} = 0$ for an enhancement-mode device.

For multi-device testing, careful consideration of device connections is essential. In a vertical device structure, the drain is common; therefore, it is not used for stress sourcing so that stress will not be terminated in case a single device breaks down. Instead, the source and gate are used to control stress.

Test Sequence

The *Vds_Vramp* test sequence has three stages: pre-test, main stress-measure, and post-test. Pre-test and post-test are optional. During the pre-test, a constant voltage is applied to verify the initial integrity of the body diode of the MOSFET; if the body diode is determined to be good, the test proceeds to the main stress-measure stage. Starting at a lower level, the drain-source voltage stress is applied to the device and ramps linearly to a point higher than the rated maximum voltage or until the user-specified breakdown criteria is reached. If the tested device is not broken at the main stress stage, the test proceeds to the next step, the post-test, in which a constant voltage is applied to evaluate the state of the device, similar to the pre-test. The measurements throughout the test sequence are made at both source and gate for multi-device testing (or drain for the single device case) and the breakdown criteria will be based on the current measured at source (or drain for a single device). The *Vds_Vramp* test sequence is useful for evaluating the effect of a drain-source bias on the device's parameters. ***Figure 3*** shows an example of a stress vs. time diagram for the main stress-measure stage of a *Vds_Vramp* test. (Pre-test and post-test are not shown.)

Although the *Vds_Vramp* sequence is generally used as a quick method of parametric verification, the *Vds_Constant* test sequence is suitable for long-term stability evaluations, such as HTRB, which normally run for an extended period at an elevated temperature. The *Vds_Constant* test sequence has a structure similar to that of the *Vds_Vramp* (that is, optional pre-test, main stress-measure, and optional post-test), with a constant voltage stress applied to the device during the stress stage and different breakdown settings. The stability of the leakage current I_{DSS} is monitored throughout the test. ***Figure 4*** shows an example stress vs. time diagram for the main stress-measure stage of a *Vds_Constant* test. (Pre-test and post-test are not shown.) Because this test requires evaluation over an extended period, the test module includes two features developed to prevent damage to the probe and device and to reduce the size of the dataset created: dynamic limit change and data compression.

Features to Solve Test Challenges

Several features of the ACS V_{DS} breakdown test module are designed to address the challenges commonly associated with V_{DS} ramp and HTRB reliability testing, namely the risk of device and equipment damage, device overstress and unexpected device breakdowns, the potential for data loss when aborting a test, and the management of large datasets. The following features offer solutions to these challenges: dynamic limit change, soft abort/bias, real-time post data, data compression, and graphic update suppression.

- **Dynamic limit change (current limit control)**
  A current limit can be set for the SMUs when applying the voltage. The output current will be clamped to the limit (compliance value) to prevent damage to the DUT. The high-level limit is usually set by estimating the maximum current during stress (for example, the current at the beginning of the stress). For most of the stress time, however, the current is much lower than the limit. When a breakdown occurs, there is no need to keep a high-level limit current running to that particular device because, over an extended time, the high-level current may melt the probe card tips and damage the devices. Dynamic limit change allows the current limit to vary according to the settings. The limit value can be set as a multiple of the measured current; in some cases, it can reduce the current by three orders of magnitude, which corresponds to 10^6 times lower in terms of power. This feature is implemented in the *Vds_Constant* test sequence (***Figure 6***).
- **Soft bias and soft abort (stress control and proper test abort design)**
  Soft bias/abort is an enhanced feature for protecting the measured device. This function allows the forced voltage or current to reach the desired value by ramping gradually at the start or the end of the stress, or when aborting the test, instead of changing suddenly. It helps to prevent in-rush currents and unexpected device breakdowns. In addition, it serves as a timing control over the process of applying stress.

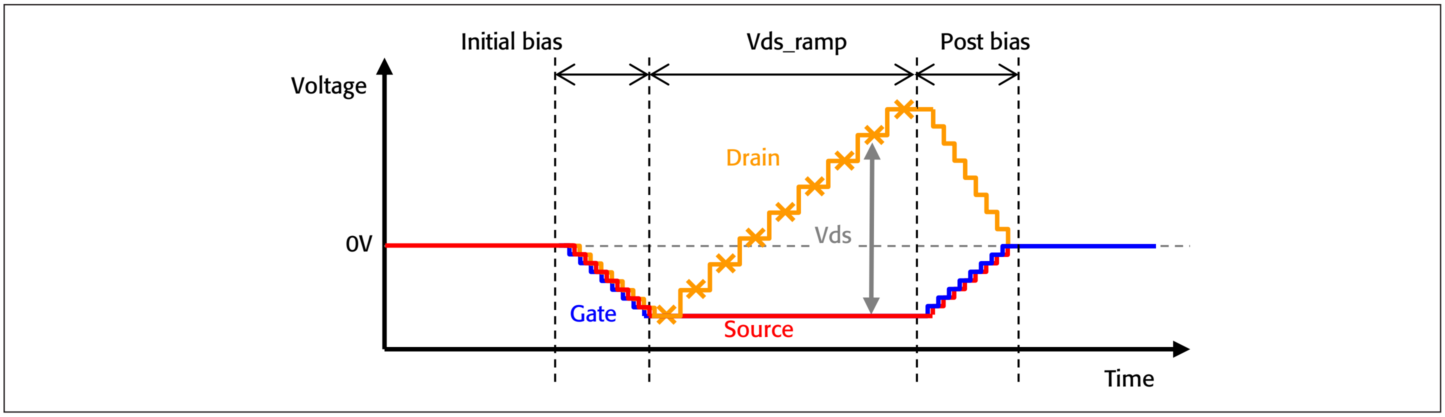


Figure 3. Stress vs. time diagram for *Vds_Vramp* test for a single device. Drain, gate and source are each connected to a Series 2600B SMU respectively. The drain is used for V_{DS} stress and measure; the V_{DS} range is extended by a positive bias on drain and a negative bias on source. A soft bias (gradual change of stress) is enabled at the beginning and end of the stress (initial bias and post bias). This test is used in conjunction with the device connection shown in *Figure 2b.* Measurements are performed at the "x" points.

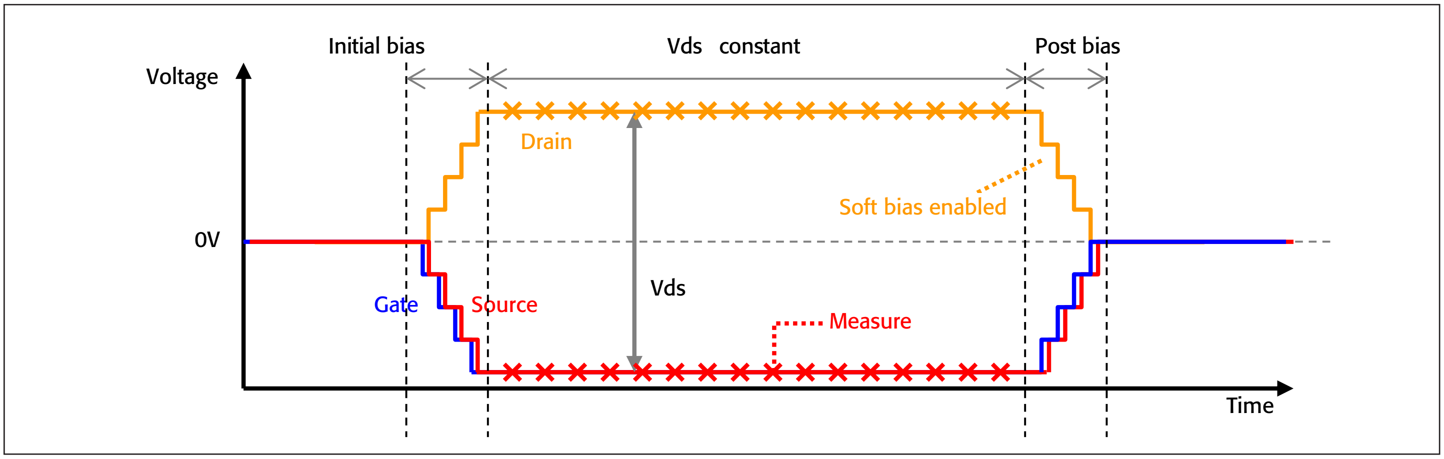


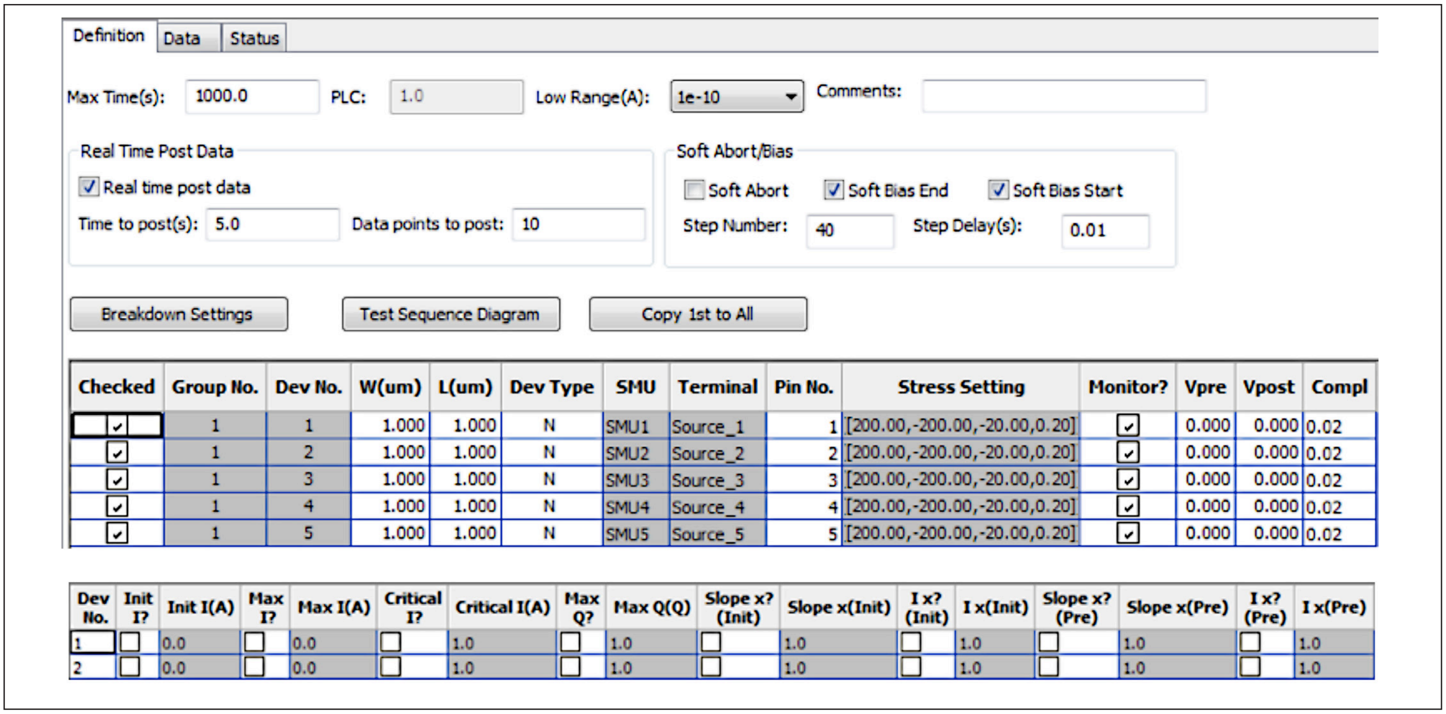
Figure 4. Stress vs. time diagram for *Vds_Constant* test sequence. The example is shown for vertical structure and multi-device case. Common drain, gate and source are each connected to a Series 2600B SMU respectively. The source is used for V_{DS} stress and measure; the V_{DS} range is extended by a positive bias on the drain and a negative bias on the source. A soft bias (gradual change of stress) is enabled at the beginning and end of the stress (initial bias and post bias). This test is used in conjunction with the device connection shown in *Figure 2f.* Measurements are performed at the "x" points.

As seen in *Figures 3 and 4*, the soft bias is enabled at the beginning and end of the stress.

The soft abort function works with a digital I/O module installed and connected to the test instrument. The digital I/O bit is set to LO after the user presses the abort button, and the LO bit input is sent to the software to invoke the soft abort function. The data will not be lost due to the data formatting at the termination of the test program. This is especially useful when the user does not want to continue the test as planned. For instance, imagine that 20 devices are being evaluated in a breakdown test for 10 hours and one of the tested devices exhibits abnormal behavior (such as substantial leakage current). The user will want to stop the test and redesign the test plan without losing the data that was taken for 10 hours. This feature is implemented in both the *Vds_Vramp* and *Vds_Constant* test modules ***(Figures 5* and *6).***

- **Real-time post data (data management and stress control** This feature enables the user to select the test data to be posted and sent from the instruments to the ACS software after a specified time interval or when the desired number of points have been accumulated. When the *Vds_Vramp* sequence is used in a large-scale ramp test, which may use up to eight SMUs or other instruments, these settings ensure safe data logging and transfer. This is because data transfer via GPIB takes longer in multi-site testing than in single device testing. In addition, the master instrument controls data transactions between all other instruments and the host PC, as well as controlling the ramping rate of the stress. The time needed for communication might prevent ramping at the user-specified ramping rate.

Initially, test data is stored in instrument memory during testing, then transferred after the test is completed, which helps ensure the system uses the desired ramp rate. However, in some instances, the user will want to see the data graphed



Max Time(s): 1000.0 PLC: 1.0 Low Range(A): 1e-10 Comments:

Real Time Post Data

☑ Real time post data

Time to post(s): 5.0 Data points to post: 10

Soft Abort/Bias

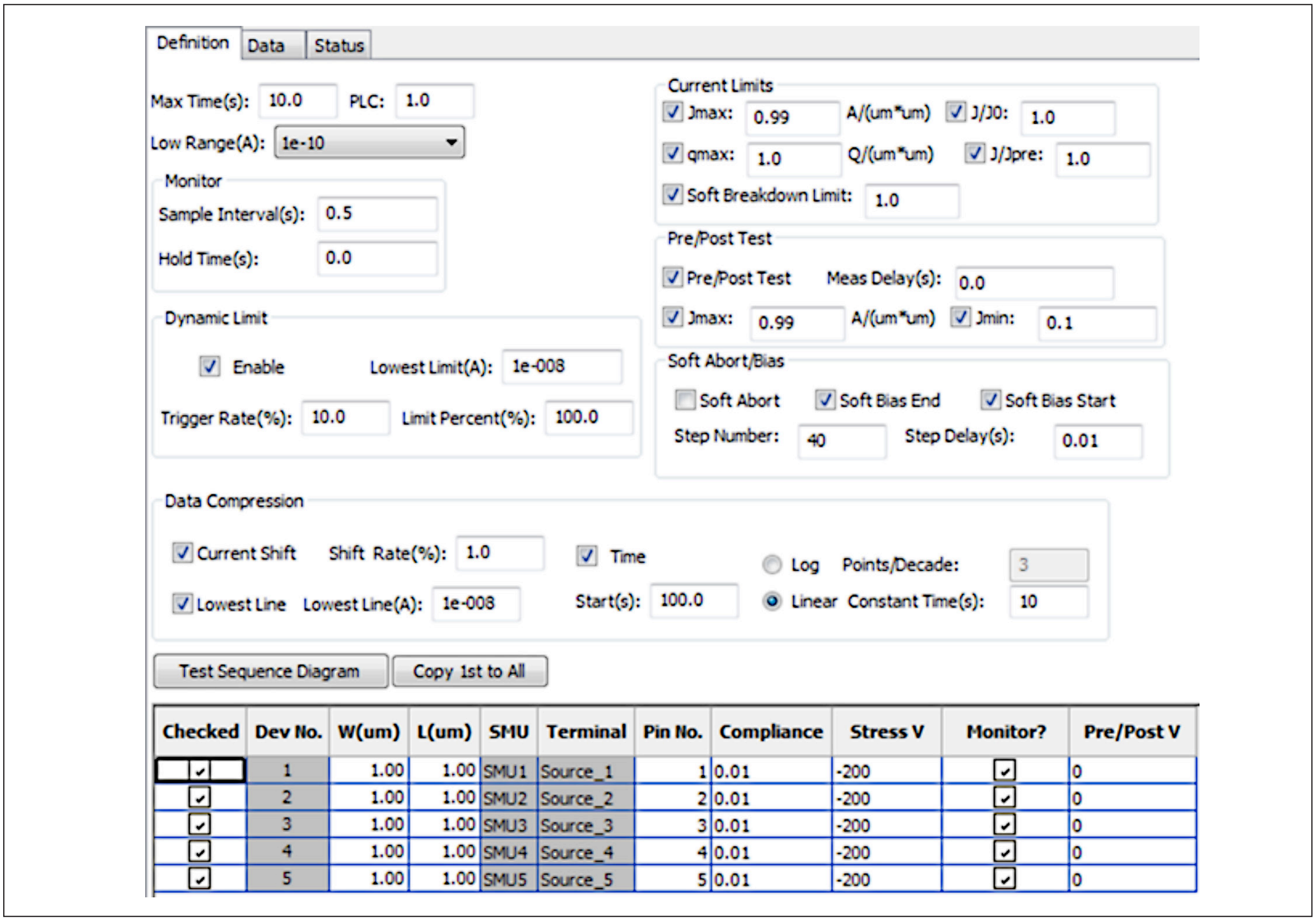
☐ Soft Abort ☑ Soft Bias End ☑ Soft Bias Start

Step Number: 40 Step Delay(s): 0.01

Checked	Group No.	Dev No.	W(um)	L(um)	Dev Type	SMU	Terminal	Pin No.	Stress Setting	Monitor?	Vpre	Vpost	Compl
☑	1	1	1.000	1.000	N	SMU1	Source_1	1	[200.00,-200.00,-20.00,0.20]	☑	0.000	0.000	0.02
☑	1	2	1.000	1.000	N	SMU2	Source_2	2	[200.00,-200.00,-20.00,0.20]	☑	0.000	0.000	0.02
☑	1	3	1.000	1.000	N	SMU3	Source_3	3	[200.00,-200.00,-20.00,0.20]	☑	0.000	0.000	0.02
☑	1	4	1.000	1.000	N	SMU4	Source_4	4	[200.00,-200.00,-20.00,0.20]	☑	0.000	0.000	0.02
☑	1	5	1.000	1.000	N	SMU5	Source_5	5	[200.00,-200.00,-20.00,0.20]	☑	0.000	0.000	0.02

Dev No.	Init I?	Init I(A)	Max I?	Max I(A)	Critical I?	Critical I(A)	Max Q?	Max Q(Q)	Slope x? (Init)	Slope x(Init)	I x? (Init)	I x(Init)	Slope x? (Pre)	Slope x(Pre)	I x? (Pre)	I x(Pre)
1	☐	0.0	☐	0.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0
2	☐	0.0	☐	0.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0	☐	1.0

Figure 5. ***Vds_Vramp*** **test sequence**



Max Time(s): 10.0 PLC: 1.0

Low Range(A): 1e-10

Monitor

Sample Interval(s): 0.5

Hold Time(s): 0.0

Dynamic Limit

☑ Enable Lowest Limit(A): 1e-008

Trigger Rate(%): 10.0 Limit Percent(%): 100.0

Current Limits

☑ Jmax: 0.99 A/(um*um) ☑ J/J0: 1.0

☑ qmax: 1.0 Q/(um*um) ☑ J/Jpre: 1.0

☑ Soft Breakdown Limit: 1.0

Pre/Post Test

☑ Pre/Post Test Meas Delay(s): 0.0

☑ Jmax: 0.99 A/(um*um) ☑ Jmin: 0.1

Soft Abort/Bias

☐ Soft Abort ☑ Soft Bias End ☑ Soft Bias Start

Step Number: 40 Step Delay(s): 0.01

Data Compression

☑ Current Shift Shift Rate(%): 1.0 ☑ Time ○ Log Points/Decade: 3

☑ Lowest Line Lowest Line(A): 1e-008 Start(s): 100.0 ● Linear Constant Time(s): 10

Checked	Dev No.	W(um)	L(um)	SMU	Terminal	Pin No.	Compliance	Stress V	Monitor?	Pre/Post V
☑	1	1.00	1.00	SMU1	Source_1	1	0.01	-200	☑	0
☑	2	1.00	1.00	SMU2	Source_2	2	0.01	-200	☑	0
☑	3	1.00	1.00	SMU3	Source_3	3	0.01	-200	☑	0
☑	4	1.00	1.00	SMU4	Source_4	4	0.01	-200	☑	0
☑	5	1.00	1.00	SMU5	Source_5	5	0.01	-200	☑	0

Figure 6. ***Vds_Constant*** **test sequence**

in real time, in order to check for irregularities and avoid human error or accidents.

The "real-time post data" option allows for manual control of data transfer. This balanced solution gives the user the advantage of both a real-time data check and ramp rate control. This feature is implemented in the *Vds_Vramp* test sequence ***(Figure 5)***.

- **Data compression (data management)**
  Reliability tests can run over many hours, days, or weeks, and have the potential to amass enormous datasets. Instead of collecting all the data produced, this function allows users to log only the data important to their particular work. Users can choose when to start data compression and how the data will be recorded by using the measured value or time settings. Data points can be logged when the current shift exceeds a specified percentage as compared to previously logged data and when the current is higher than a specified noise level. If the "time" setting is selected, the mean value of the data over the specified period or each value of the specified number of points per decade will be logged. This feature is implemented in the *Vds_Constant* test sequence ***(Figure 6).***
- **Suppress redundant graphic update (data management)**
  This practical function reduces the risks of working with large datasets by allowing users to specify that results graphics should NOT be refreshed automatically. This offers an effective way to avoid continuously and redundantly plotting and replotting a huge amount of data. The graph is updated only when the user requests it by pressing the "Refresh" button.

Conclusion

The V_{DS} ramp and HTRB tests are important power device reliability tools for parametric verification and long-term stability evaluation. The V_{DS} breakdown test module in ACS Version 5.0 includes two stress test sequences for V_{DS} ramp and HTRB reliability evaluation, which are enhanced with features that address common test challenges. The software and instrumentation in an ACS system are configurable for use in a wide variety of applications at the device, wafer, or cassette level, and in settings from a simple benchtop test setup to an automated, integrated rack of instruments. An optional ACS-2600-RTM package from Keithley offers additional reliability tests, including Hot Carrier Injection (HCI), Negative Bias Temperature Instability (NBTI), Time Dependent Dielectric Breakdown (TDDB), voltage ramp (V_{Ramp}) test, and ramped current (J_{Ramp}) tests. Refer to the ACS-2600-RTM User's Manual for more information on these tests [10].

References

1. R. J. Kaplar, M. J. Marinella, S. DasGupta, M. A. Smith, S. Atcitty, M. Sun, and T. Palacios, "Characterization and Reliability of SiC- and GaN Based Power Transistors for Renewable Energy Applications" IEEE EnergyTech, 2012
2. L. Stauffer and M. Cejer, "Demand for Higher Power Semi Devices Will Require Pushing Instrumentation to New Extremes," Keithley Instruments documentation 2012
3. J. Lutz, H Schlangenotto, U. Scheuermann, and R. De Doncker, Semiconductor Power Devices: Physics, Characteristics, Reliability, Springer, 2011.
4. R. Singh, A. R. Hefner, and T. R. McNutt, " Reliability Concerns in Contemporary SiC Power Devices"
5. EPC GaN Transistor Application Reliability Report (Readiness: Phase One Testing)
6. International Rectifier RBV BRIDGE SERIES QUALIFICATION REPORT
7. Cree Z-RECTM 1200V Qualification Report
8. Power Semiconductor Reliability Handbook. Alpha and Omega Semiconductor
9. ACS Reference Manual
10. ACS-2600-RTM User's Manual

Testing High Brightness LEDs under Pulse Width Modulation Using the Model 2651A High Power System SourceMeter® Instrument

Introduction

With a worldwide focus on energy conservation, the high power consumption of traditional light sources has come under heavy scrutiny and governments are demanding improvements in the energy efficiency of lighting sources. This demand has lead to heavy investment in and development of alternatives to the incandescent bulb. Compact Fluorescent (CFL) bulbs have become prevalent in the marketplace and, although they are much more efficient and last longer than traditional bulbs, they still fall short of being ideal replacements. High Brightness Light Emitting Diodes (HBLEDs), however, have proven themselves to be a much better option. Like incandescent bulbs, they reach full brightness immediately and do not contain any difficult-to-dispose-of chemicals. They also offer advantages over incandescents: they have incredibly long lifetimes and efficiency levels are continuously being increased.

Unfortunately, the cost of HBLEDs is still too high for most consumers to select them over cheaper technologies, despite the large reduction in energy consumption they offer. In order to reduce the cost of HBLEDs to consumers, manufacturers are constantly working to improve yields and further increase efficiency levels. Meeting this demand requires proper testing and proper testing requires the proper test equipment. Manufacturers demand a lot from their test equipment today to test high brightness LEDs properly. This application note explores some of the electrical test requirements and how the Model 2651A High Power System SourceMeter® instrument can be applied to meet these demands.

The Demand for More Power

HBLEDs are commonly defined as LEDs that operate at 1W of power or higher, with typical operating ranges of from 1W to 3W. Instead of running on 10–30mA of current and being packaged in small 3mm or 5mm plastic domes, HBLEDs run at 300mA–1A or more and are mounted on a small, thermally conductive board designed to draw heat away from the LED's junction. Despite how bright a single HBLED can be, it is typically not bright enough to be used by itself in most lighting applications. Instead, multiple HBLEDs are commonly combined to create a single luminaire, whether for an LED light bulb for a retrofit application or an entire lighting fixture. Even though they are combined in actual use, production testing is typically performed at the individual package level, requiring modest power delivery capability, which is well within the capabilities of most modern instrument-based source-measure units (SMUs), such as the Keithley Series 2400, 2600B and 2651A SourceMeter instruments.

For many applications, combining multiple LEDs into a single luminaire works well when it's desirable to have the light spread in multiple directions and the size of the luminaire provides sufficient space to fit multiple LEDs. However, in other applications where space is limited and/or the light must be directional, this approach is either undesirable or simply will not work. This demand for a lot of light in a small package has lead to the development of high power LED modules, which consist of one or more large-die LEDs. When multiple die are present, they're either wired in parallel or in series, depending on the application and the available power source. The die of these LEDs are much larger than the die of typical HBLEDs and can handle much larger currents as well. It's common for a single die to be required to withstand current levels as high as 10A.

Testing high power HBLED modules properly demands test equipment that can deliver a lot of power to the DUT. Although SMUs, given their ability to source and measure in a single instrument, are the best type of test equipment for testing LEDs, most SMUs on the market simply can't deliver the level of power required. High power HBLED modules often require 100W of power or more, but most instrument-based SMUs are capable of delivering only 40W or less. Keithley's Model 2651A High Power System SourceMeter instrument is capable of delivering up to 200W of continuous DC power and up to 2000W of pulsed power, making it more than powerful enough to test both today's and tomorrow's high power modules (***Figure 1***).

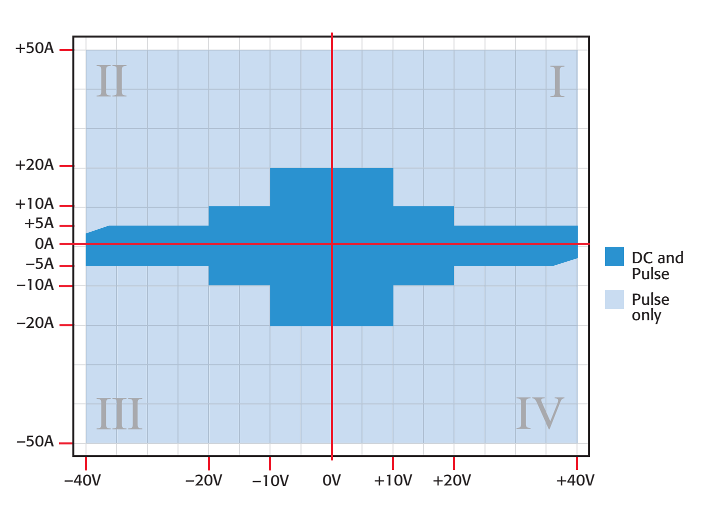


Figure 1: Power envelope for the Model 2651A High Power System SourceMeter Instrument.

Pulse Width Modulation

Pulse width modulation is a common method of controlling the brightness of LEDs. When using this technique, the current through the LED is pulsed at a constant frequency with a constant pulse level, but the width of the pulse is varied (***Figure 2***). Varying the width of the pulse changes the amount of time the LED is in the ON state, as well as the perceived level of brightness. In this drive scheme, the LED is actually flashing, but the frequency of the flashing is so high that it's indistinguishable to the human eye from a constant light level. Although it's possible to control the brightness of an LED simply by lowering the forward drive current, pulse width modulation is the preferable technique for a number of reasons.

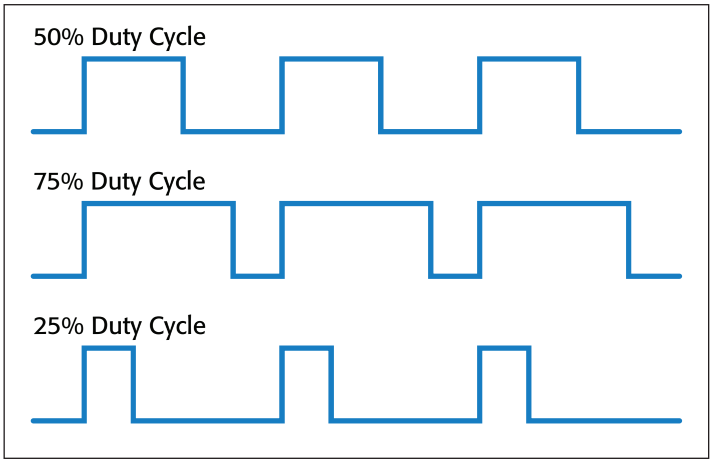


Figure 2: In pulse width modulation, the pulse level and frequency remain constant but the duty cycle is varied.

The first and arguably the most important reason for using pulse width modulation is to maintain consistency of the color of the light as the LED's brightness is reduced. In an LED, the color of the light it emits is related to the forward voltage at which it operates. Although the forward voltage of an LED will remain relatively constant as the forward current is changed, it actually does vary by as much as tens to even hundreds of millivolts. This occurs especially at lower current levels (***Figure 3***). This slight variation in forward voltage equates to a slight variation in light color, which is undesirable for the end user. If heating effects are ignored, in the pulse width modulation technique, the LED is pulsed using exactly the same current level on every pulse, so the forward voltage is the same for every pulse; therefore, the color of the light emitted will not vary.

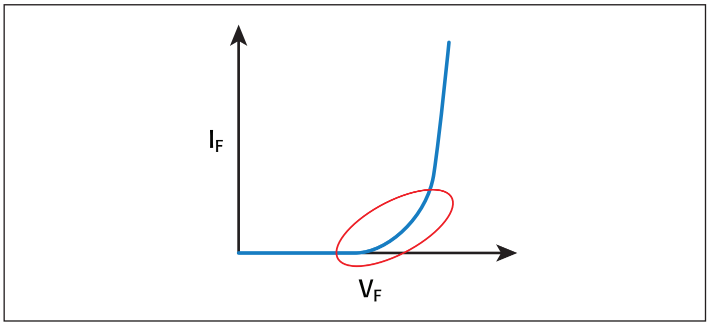


Figure 3: Forward voltage changes significantly when the forward current is low and becomes relatively constant as forward current becomes large.

Another important reason pulse width modulation is preferable is because this technique provides linear control over brightness. The amount of light an LED emits is not linearly related to the amount of current used to drive it. In other words, reducing the drive current by 50% will not cut the light output by 50%; instead, it will drop by some other amount. This would make a dimming scheme based on varying current difficult to apply because it would be necessary to characterize each LED's light output vs. forward current then calibrate the drive scheme to that curve. Using pulse width modulation is a much simpler way to get linear control over brightness. With pulse width modulation, in order to make the LEDs output 50% as much light, all that's necessary is to reduce the duty cycle by 50%. If the LEDs are only ON for half as long, only half as much light will be produced.

Power efficiency is another advantage of the pulse width modulation approach. Because pulse width modulation uses a constant current level for each pulse, it's possible to select a pulse level where the LED operates most efficiently, that is, where the lumen output per watt is the greatest. That means the LED is operating at maximum efficiency no matter what brightness level is used. Another way in which pulse width modulation enhances efficiency is that LEDs will actually output more light for a given drive current when pulsed rather than at DC. Many manufacturers' datasheets include a graph of forward current vs. luminous flux. If the manufacturer has characterized the LED under both pulsed and DC drive currents, one can observe that the pulsed characterization curve lies above the DC characterization curve. This is due to the reduced self-heating that the pulsed drive current produces. Finally, pulse width modulation enhances power efficiency even in the circuitry that drives it. The switching circuitry used in Pulsed Width Modulation wastes very little power. When the switch circuitry is turned off, virtually no current flows and virtually zero power is being used. When the switch is turned on, due to the very low on state resistances, nearly all the power is delivered to the LED and very little is consumed by the switch. In a variable current drive scheme, power to the LED is often reduced by consuming the excess power elsewhere in the circuit.

Pulse Width Modulation with the Model 2651A High Power System SourceMeter Instrument

NOTE: The techniques described here for outputting a pulse width modulated waveform with the Model 2651A are applicable to all members of the Series 2600B System SourceMeter instrument family.

Given that LEDs are often used with pulse width modulation, it's only appropriate that they be tested with pulse width modulation techniques. As part of Pulsed Width Modulation testing, an LED is usually tested by running a series of pulses through it while using a spectrometer to take an integrated measurement of the light output over the course of many pulses. This measurement may take tens or hundreds of milliseconds to complete. During the pulsed output, the forward voltage is measured on every pulse to look for changes as the temperature of the LED rises. ***Figure 4*** illustrates this test.

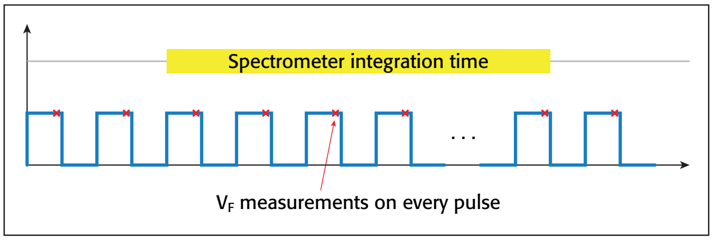

Spectrometer integration time
. . .
V_F measurements on every pulse

Figure 4: The LED is pulsed with a series of pulses and a V_F measurement is taken at each pulse. A spectrometer concurrently takes optical measurements.

The Model 2651A High Power System SourceMeter instrument is capable of outputting a pulse width modulated waveform with up to 100% duty cycle from 0–20A, 50% duty cycle from 20–30A, and 35% duty cycle from 30–50A. The Model 2651A's advanced trigger model allows for precision pulse widths and duty cycles and tight synchronization with other instruments. These synchronization features can be used to combine two Model 2651As to achieve a Pulsed Width Modulation waveform with pulse current levels twice as high as a single Model 2651A allows with the same duty cycle. This note details how to configure the Model 2651A to output a 30A, 50% duty cycle waveform with a digital I/O output trigger for triggering a spectrometer. It also details how to combine two Model 2651As to increase the current to 60A for this same test.

Required Equipment

Performing this test requires the following equipment:

- PC with GPIB or Ethernet adapter
- GPIB cable or RJ45 LAN Crossover Ethernet cable
- Model 2651A High Power System SourceMeter Instrument
- Model 2651A-KIT-1 Low-Impedance/High-Current Coaxial Cable
- 8-pin signal control cable
- 2-pin terminal block extender for use with the Model 2651A-KIT-1
- 8-pin terminal block extender for use with 8-pin signal control cable
- Digital I/O DB-25 Male Connector Kit Hardware
- 12 AWG or thicker cabling to connect from terminal blocks to device

Add the following equipment to perform this test at up to 100A with a 35% duty cycle or at up to 60A with a 50% duty cycle:

- Model 2651A High Power System SourceMeter Instrument
- Model 2651A-KIT-1 Low-Impedance/High-Current Coaxial Cable
- 8-pin Signal Control Cable
- 2-pin terminal block extender for use with the Model 2651A-KIT-1
- 8-in terminal block extender for use with 8-pin signal control cable
- TSP-Link RJ45 LAN Crossover Cable

Communications

Single SourceMeter Instrument

To perform this test using a single 2651A SourceMeter instrument configuration, connect the instrument to the computer via GPIB or Ethernet as illustrated in ***Figure 5***.

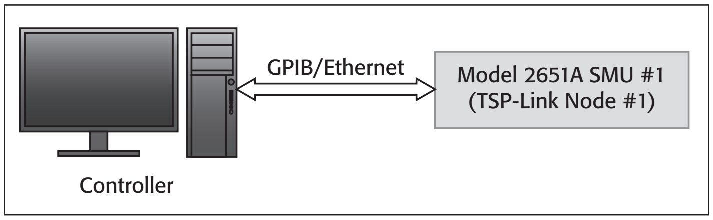


Figure 5: Communications setup for a single SourceMeter instrument.

Dual SourceMeter Instruments

For a dual 2651A SourceMeter instrument configuration, connect the first Model 2651A to the computer via GPIB or Ethernet. Connect the second Model 2651A to the first Model 2651A via the TSP-Link connection. Assign the first Model 2651A to Node #1 and the second Model 2651A to Node #2. These connections are illustrated in ***Figure 6***.

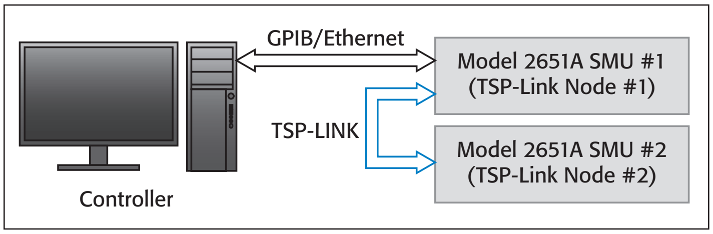


Figure 6: Communications setup for two Model 2651A instruments.

Connecting the Spectrometer to the Digital I/O

In order for the Model 2651A to trigger the spectrometer, the spectrometer's start of test trigger line must be connected to the Model 2651A's digital I/O port. Connect the trigger line from the spectrometer to pin #1 of the 25-pin D-Sub connector on the back panel of the Model 2651A. Ground connections can be

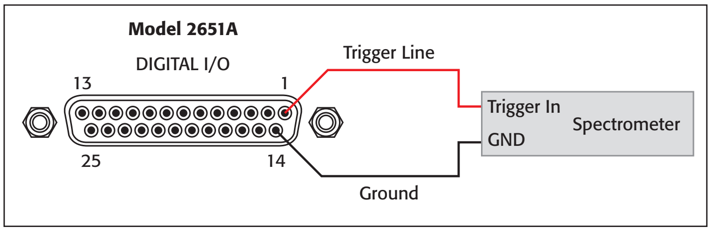


Figure 7: Connections from Model 2651A digital I/O port to spectrometer.

found on any of pins 15 through 21. The proper connections are illustrated in ***Figure*** *7*.

NOTE: In dual SourceMeter instrument configurations, connect only to the digital I/O port of SMU #1. The digital I/O port of SMU #2 is not used.

Device Connections

Figures 8 and ***9*** illustrate the connections from the SourceMeter instruments to the LED device under test. ***Figure 8*** illustrates the connections for a single SourceMeter instrument; ***Figure 9*** illustrates the connections for dual SourceMeter instruments.

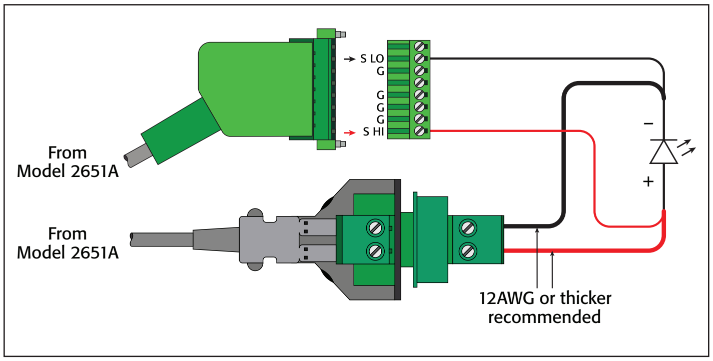


Figure 8: Connections from a single Model 2651A SourceMeter instrument to LED device under test.

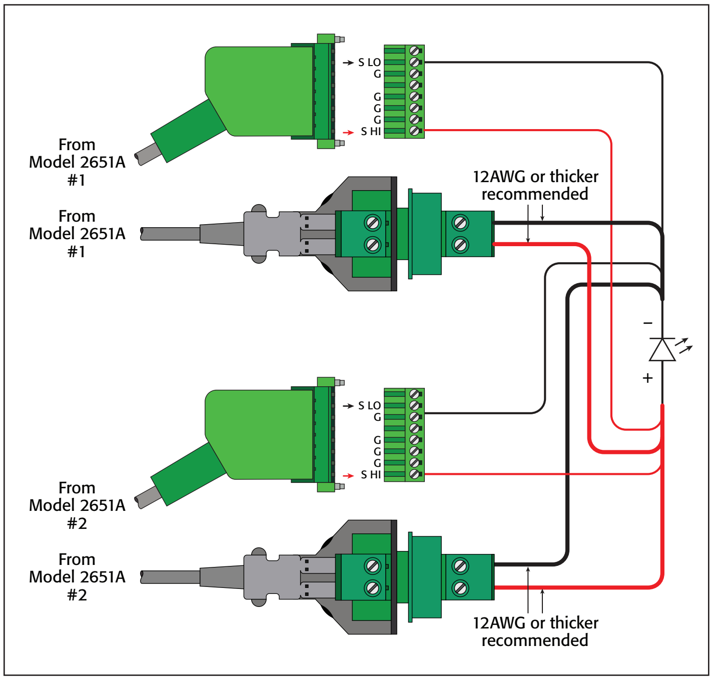


Figure 9: Connections from dual Model 2651A SourceMeter instruments to LED device under test.

When using dual Model 2651A SourceMeter instruments, the two SMUs are connected in parallel to create a single source with twice as much current capacity. Read and understand the section titled "Combining SMU outputs" in the *Model 2651A High Power System SourceMeter Instrument Reference Manual* before performing this test.

NOTE: Wiring from the 2-pin terminal block extender that is connected to the end of the low-impedance/high-current coaxial cable provided with the Model 2651A to the device under test should be made with 12 AWG wire or heavier in order to support the high current levels. Also, the length of these wires should be kept to a minimum to minimize inductance.

Configuring the Trigger Model

The advanced trigger model must be used to source current levels above those available in the DC operating regions of the Model 2651A. This same trigger model gives the Model 2651A its precise pulse widths and tight synchronization and makes accurate pulse width modulation possible. The following sections illustrate how to configure the trigger model to output a pulse width modulated waveform and trigger a spectrometer with the Model 2651A.

Configuring the Trigger Model for a Single Model 2651A SourceMeter Instrument

The trigger model shown in ***Figure 10*** will perform a pulse width modulation test on an LED using a single Model 2651A. In this configuration, Timer 1 controls the pulse period, Timer 2 controls the pulse width and Timer 3 inserts a delay between the start of the pulse and the start of the measurement. Timer 4 creates a delay between the start of the waveform output and the start of the spectrometer measurement. When Timer 4 expires, it triggers Digital I/O Trigger 1, which sends the start trigger to the spectrometer.

Configuring the Trigger Model for Dual Model 2651A SourceMeter Instruments

The trigger model shown in ***Figure 11*** will perform a pulse width modulation test on an LED using two Model 2651A SourceMeter instruments with their outputs connected together in parallel. Timer 1 of Model 2651A #1 controls the pulse period of both SMUs. It does this by relaying its output trigger signal on TSP-Link® Trigger 1, which is then sent to Model 2651A #2. Due to the extremely low latency of TSP-Link triggering, Model 2651A #2 is kept in sync with Model 2651A #1 to within 500ns. Because this signal is sent once every time Timer 1 expires, long-term synchronization between the SMUs is ensured because the two SMUs are synchronized at the start of every pulse cycle.

As in the single SMU configuration, Timer 2 controls the pulse width and Timer 3 inserts a delay between the start of the pulse and the start of the measurement. Each SMU uses its own Timer 2 and Timer 3 to control its pulse width and measure delay. Given the accuracy of the timers and the synchronization at the start of every pulse, it's certain that both SMUs will output a pulse with the same pulse width and take measurements at the same time. Finally, once again Timer 4 on Model 2651A #1 creates a delay between the start of the pulse waveform and the start of the spectrometer measurement by delaying the output of Digital I/O Trigger 1.

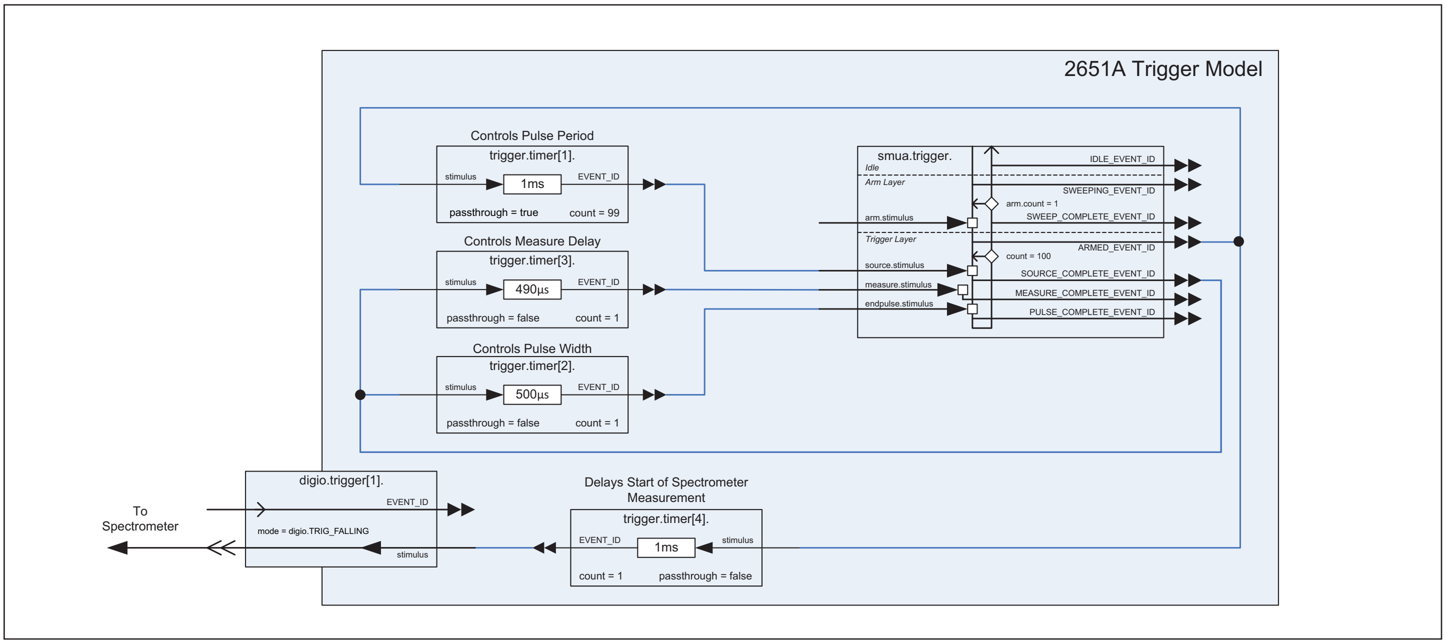


Figure 10: Trigger model for pulsed width modulation test on LED with a single Model 2651A System SourceMeter instrument.

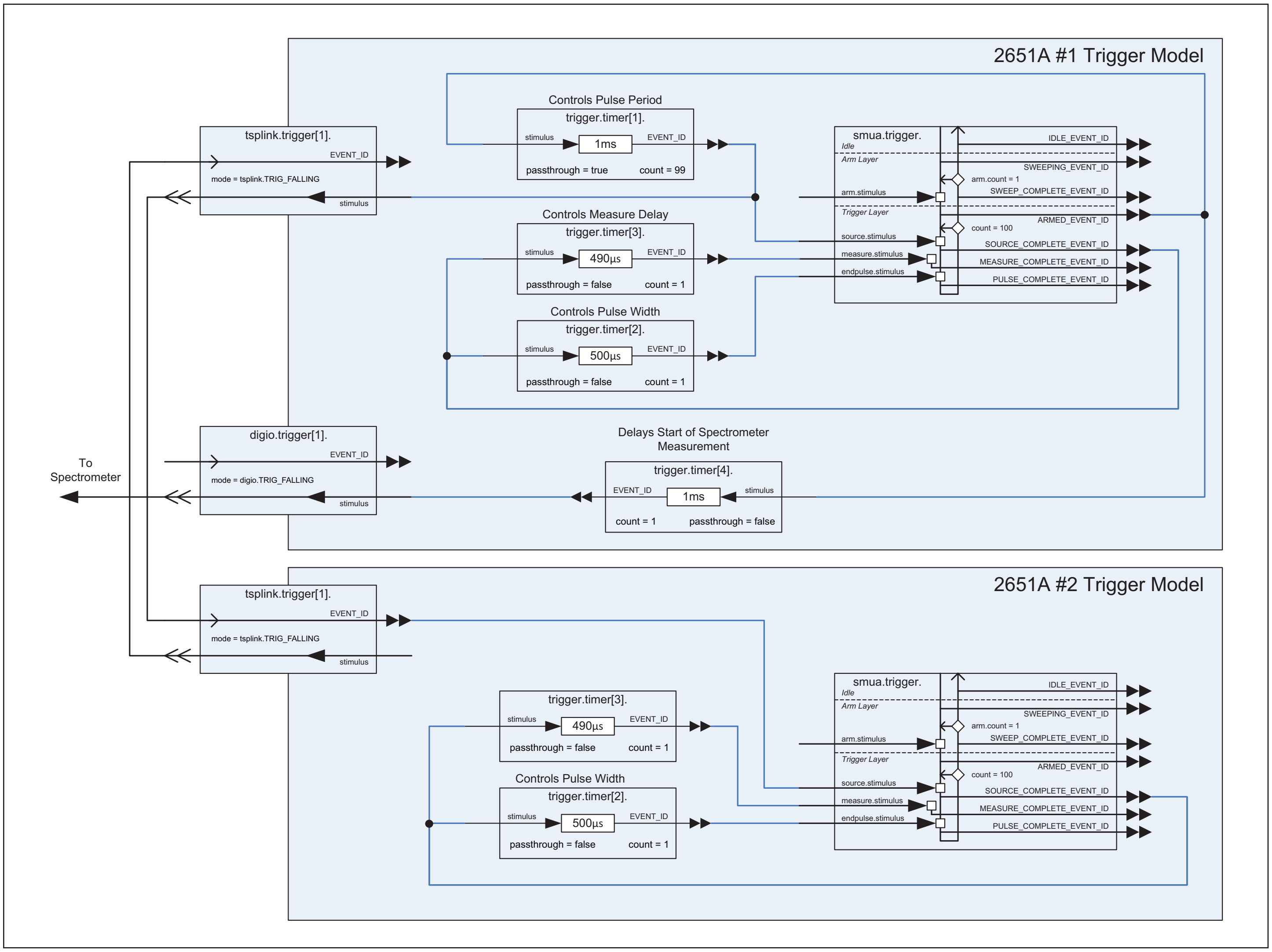


Figure 11: Trigger model for pulsed width modulation test on LED with dual Model 2651A System SourceMeter instruments.

Configuring the Frequency and Duty Cycle

In both the single and dual SMU configurations, configuring the waveform for a particular frequency and duty cycle requires setting the appropriate pulse period and pulse width in the trigger model. Frequency (f) is related to pulse period (P) by the equation P = 1 / f; therefore, it's possible to control the frequency of the waveform by setting the appropriate pulse period. For a 1kHz waveform, P = 1/1kHz = 1ms. Duty cycle (D.C.) is the ratio of pulse width (PW) over pulse period (P) or D.C. = PW / P; therefore, the duty cycle can be set by setting the appropriate value for pulse width; this value can be calculated using the equation D.C. * P = PW.

Modulating the Pulse Width

The trigger model diagrams in ***Figures 10*** and ***11*** show the pulse width being controlled by Timer 2, which has a fixed timeout value. This is represented in code by calling the ICL command

```
trigger.timer[2].delay = pulseWidth
```

where **`pulseWidth`** is a fixed delay value in seconds. Having a fixed timeout value means that the pulse width for every cycle in a single waveform output will be the same for the entire length of the waveform. Modulating the pulse width during a single waveform output requires that the timeout value of Timer 2 be variable. To facilitate this, the timers of the advanced trigger model can be assigned a delay list rather than a single value. This can be done by calling the ICL command

```
trigger.timer[2].delaylist = pulseWidthTable
```

where **`pulseWidthTable`** is a table containing multiple delay values in seconds. With a delay list assigned to Timer 2, each pulse in the waveform can be assigned a different pulse width; therefore, the waveform can be pulse width modulated.

NOTE: The example script in this application note allows for both fixed and variable pulse widths. See the function documentation for details.

Example Program Code

NOTE: The Test Script Processor (TSP®) Script in this application note is for demonstration purposes only and is not optimized for fastest production throughput. Please contact a Keithley Applications Engineer for system throughput optimization considerations.

NOTE: The TSP Script in this application note is designed to be run from Test Script Builder. It can be run from other programming environments such as Microsoft® Visual Studio or National Instruments LabVIEW®; however, modifications may be required.

The TSP script provided in this application note contains all the code necessary to perform a pulse width modulation test with optical measurement on a high brightness LED using one or two Model 2651A High Power System SourceMeter instruments. The code for this script can be found in Appendix A: Source Code.

The script performs the following functions:

- Initializes the TSP-Link connection (only when using two units)
- Configures the SMU(s) ranges and measurement settings
- Configures the trigger model(s)
- Prepares the readings buffers for data
- Outputs the PWM waveform
- Returns the collected data to the instrument console in a format that can be copied and pasted directly into a Microsoft Excel® spreadsheet.

The script is written using TSP functions rather than a single block of inline code. TSP functions are similar to functions in other programming languages such as C or Visual Basic and must be called before the code contained in them is executed. Because of this, running the script alone will not execute the test. To execute the test, first run the script to load the functions into Test Script memory and then call the functions. Refer to the documentation for Test Script Builder for directions on how to run scripts and enter commands using the instrument console.

Within the script, there are several comments describing what is being performed by the lines of code, as well as documentation for the functions contained in the script. Lines starting with

```
node[2].
```

are commands that are being sent to Model 2651A #2 through the TSP-Link interface. All other commands are executed on Model 2651A #1.

Example Program Usage

This script contains two functions for outputting a pulse width modulated waveform: one for use with a single Model 2651A High Power System SourceMeter instrument and one for use with two Model 2651A High Power System SourceMeter instruments. These functions contain parameters whose values are used to configure the waveform, allowing the user to set properties of the waveform such as the frequency and duty cycle without needing to rewrite any code. The following sections provide explanations of each function and its parameters.

PWM_Test_Single()

```
PWM_Test_Single(pulseLevel, pulseLimit, frequency,
dutyCycle, numpulses, specDelay)
```

This function will output a pulse width modulated waveform using a single Model 2651A High Power System SourceMeter instrument. A forward voltage measurement will be taken with the Fast ADC on every pulse in the waveform and measurements will be placed 10μs before the falling edge of the pulse. Use the parameters of this function to configure the properties of the Pulsed Width Modulation waveform and to set the delay between the start of the waveform and the start of the spectrometer measurement. If any parameters are left blank, a default value will be used.

Parameter	Units	Description
`pulseLevel`	Amps	The current level to pulse to during the test Min: –50 Max: +50 Default: 1 ***Comments:*** The value set here goes into determining what operating region the SMU is in and will therefore have an effect on the maximum duty cycle value that can be set without error.
`pulseLimit`	Volts	Voltage limit of the pulses during the test Max: 40 Default: 1 ***Comments:*** The value set here goes into determining what operating region the SMU is in and will therefore have an effect on the maximum duty cycle value that can be set without error.
`frequency`	Hz	The number of pulses per second Min: 0.1 Max: 10,000 Default: 100 ***Comments:*** Frequency and Duty Cycle determine the pulse width of the waveform. The minimum frequency that can be used without error may be higher than the value shown if the operating region for the test is outside of the DC operating area.
`dutyCycle`	%	The on time of the pulse as a percentage of the pulse period Min: 0.01 Max: 99 Default: 1 ***Comments:*** This parameter may be assigned a single value or a table of values. If it is assigned a single value, then all pulses in the waveform will have the same duty cycle. If this parameter is assigned a table of values, then the duty cycle for each pulse will be determined by a corresponding entry in the table. For example, if the table has the values 50, 25, and 40, then the first pulse in the waveform will have a duty cycle of 50%, the second pulse will have a duty cycle of 25%, and the third pulse a duty cycle of 40%. If there are more pulses in the waveform than there are values in the table, then when the last value in the table is reached, the values in the table will be reused from the beginning of the table starting with the next pulse. Using the previous example, if the number of pulses to output is 5, then the duty cycles of the pulses will be 50%, 25%, 40%, 50%, and 25%. Frequency and Duty Cycle determine the pulse width of the waveform. Pulse and duty cycle of the instrument are limited depending on the region of the power envelope the SMU is operating in. The minimum and maximum duty cycle that can be used without error may be higher or lower than the values shown depending on the operating region and selected frequency for the test. If the duty cycle specified results in a pulse width/duty cycle that is too large for the operating region, the SMU will limit the pulse width/duty cycle itself to its maximum allowable value. This will appear in the output waveform in the form of pulses that are cut short or are missing. See the Model 2651A Specification for details on maximum pulse widths and duty cycles.
`numpulses`	N/A	The number of pulses in the waveform Min: 2 Max: 100,000 or greater Default: 10
`specDelay`	Seconds	The time between the start of the PWM output and the output of the Digital I/O trigger to start the spectrometer measurement. Min: 0 Default: 0

The following is an example call to function **PWM _ Test _ Single()**.

```
PWM_Test_Single(30, 10, 1000, 50, 100, 1e-3)
```

This call will output a pulse width modulated waveform of 100 pulses with a pulse level of 30A, a 10V voltage limit, a frequency of 1kHz, and a duty cycle of 50%. Spectrometer measurements will begin 1ms after the start of the waveform output. At the completion of the test, the SMU output will be turned off and the forward voltage measurements collected during the test will be printed to the instrument console in a format compatible with copying and pasting into a Microsoft Excel spreadsheet. An example of this output can be seen in ***Figure 12***.

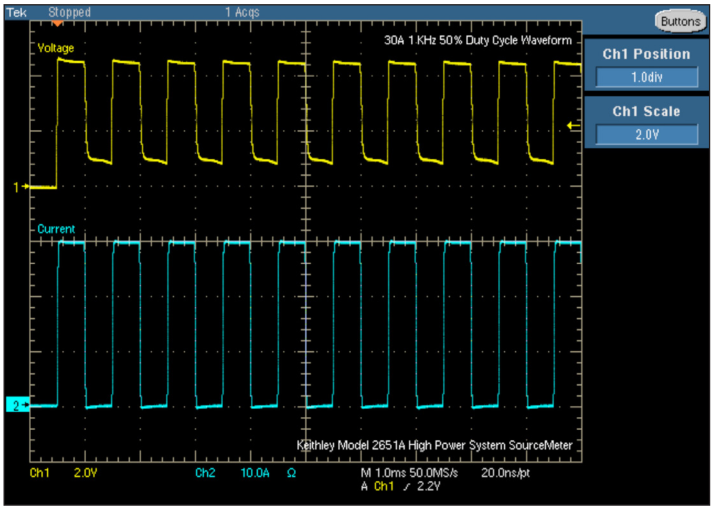


Figure 12: 30A 1kHz pulsed waveform with 50% duty cycle from a Keithley Model 2651A into a high power LED module.

In the previous example, the duty cycle of the waveform was fixed at 50%. To create a waveform where the pulse width varies for each pulse, it's necessary to pass the function a table of duty cycles. The following example calls demonstrate this.

```
dutyTable = {20, 40, 60, 80, 60, 40, 20, 40, 60
PWM_Test_single(20, 10, 1000, dutyTable, 9, 1e-3)
```

This call will output a pulse width modulated waveform of 9 pulses with a pulse level of 20A, a 10V voltage limit, a frequency of 1kHz and a varying duty cycle. The duty cycle for each pulse will be read from the table. The first pulse will have a duty cycle of 20%, the second pulse a duty cycle of 40%, the third pulse a duty cycle of 60% and so on. Spectrometer measurements will begin 1ms after the start of the waveform output. At the completion of the test, the SMU output will be turned off and the forward voltage measurements collected during the test will be printed to the instrument console in a format compatible for copy and pasting into Microsoft Excel. An example of this output can be seen in ***Figure 13***.

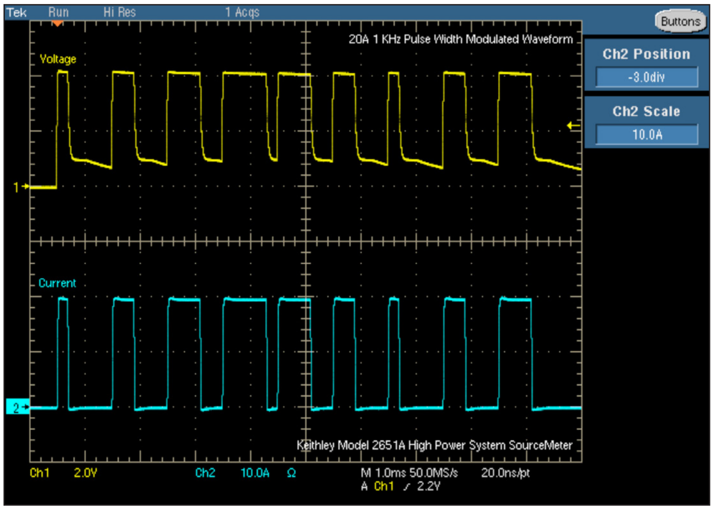


Figure 13: 20A 1kHz pulse width modulated waveform from a Keithley Model 2651A into a high power LED module.

PWM_Test_Dual()

```
PWM_Test_Dual(pulseLevel, pulseLimit, frequency,
dutyCycle, numpulses, specDelay)
```

This function uses two Model 2651A High Power System SourceMeter instruments to output a pulse width modulated waveform with as much as twice as much current as a single Model 2651A is capable of.

This function will output a pulse width modulated waveform using two Model 2651A High Power System SourceMeter instruments connected together via TSP-Link. By combining the SMU's outputs in parallel, twice the current of a single Model 2651A can be delivered to the device under test. Just like the single SMU test, a forward voltage measurement will be taken on every pulse in the waveform and measurements will be taken with the Fast ADC, placed 10μs before the falling edge of the pulse. Use the parameters of this function to configure the properties of the Pulsed Width Modulation waveform and to set the delay between the start of the waveform and the start of the spectrometer measurement. If any parameters are left blank, a default value will be used.

Parameter	Units	Description
`pulseLevel`	Amps	The current level to pulse to during the test Min: –100 Max: +100 Default: 1 ***Comments:*** The value set here goes into determining what operating region the SMU is in and will therefore have an effect on the maximum duty cycle value that can be set without error.
`pulseLimit`	Volts	Voltage limit of the pulses during the test Max: 40 Default: 1 ***Comments:*** The value set here goes into determining what operating region the SMU is in and will therefore have an effect on the maximum duty cycle value that can be set without error.
`frequency`	Hz	The number of pulses per second Min: 0.1 Max: 10,000 Default: 100 ***Comments:*** Frequency and Duty Cycle determine the pulse width of the waveform. The minimum frequency that can be used without error may be higher than the value shown if the operating region for the test is outside of the DC operating area.
`dutyCycle`	%	The on time of the pulse as a percentage of the pulse period Min: 0.01 Max: 99 Default: 1 ***Comments:*** This parameter may be assigned a single value or a table of values. If it is assigned a single value then all pulses in the waveform will have the same duty cycle. If this parameter is assigned a table of values then the duty cycle for each pulse will be determined by a corresponding entry in the table. For example, if the table has the values 50, 25, 40 then the first pulse in the waveform will have a duty cycle of 50%, the second pulse will have a duty cycle of 25% and the third pulse a duty cycle of 40%. If there are more pulses in the waveform than there are values in the table, then when the last value in the table is reached, the values in the table will be reused from the beginning of the table starting with the next pulse. Using the previous example, if the number of pulses to output is 5 then the duty cycles of the pulses will be 50%, 25%, 40%, 50%, 25%. Frequency and Duty Cycle determine the pulse width of the waveform. Pulse and duty cycle of the instrument are limited depending on the region of the power envelope the SMU is operating in. The minimum and maximum duty cycle that can be used without error may be higher or lower than the values shown depending on the operating region and selected frequency for the test. If the duty cycle specified results in a pulse width/duty cycle that is too large for the operating region, the SMU will limit the pulse width/duty cycle itself to its maximum allowable value. This will appear in the output waveform in the form of pulses that are cut short or are missing. See the Model 2651A Specification for details on maximum pulse widths and duty cycles.
`numpulses`	N/A	The number of pulses in the waveform Min: 2 Max: 100,000 or greater Default: 10
`specDelay`	Seconds	The time between the start of the PWM output and the output of the Digital I/O trigger to start the spectrometer measurement. Min: 0 Default: 0

The following is an example call to function **`PWM_Test_Dual()`**.

```
PWM_Test_Dual(60, 10, 1000, 50, 100, 1e-3)
```

This call will output a pulse width modulated waveform of 100 pulses with a pulse level of 60A, a 10V voltage limit, a frequency of 1kHz, and a duty cycle of 50%. Spectrometer measurements will begin 1ms after the start of the waveform output. At the completion of the test, the SMU output will be turned off, and the forward voltage measurements collected during the test will be printed to the instrument console in a format compatible with copy and pasting into Microsoft Excel. An example of this output can be seen in ***Figure 14***.

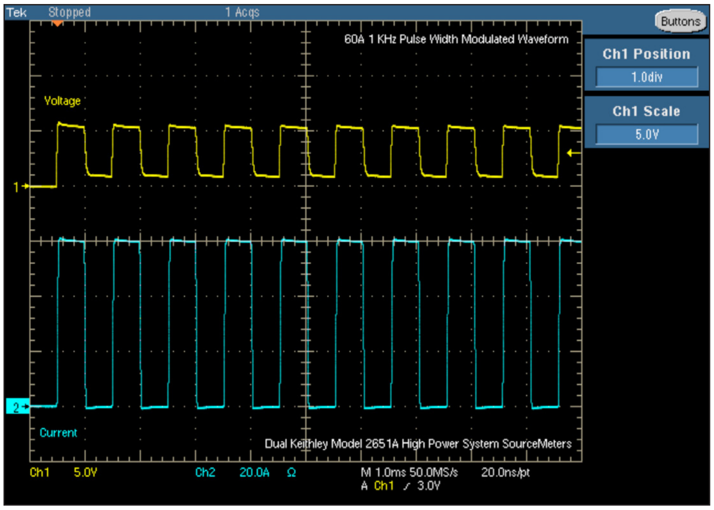


Figure 14: 60A 1kHz pulsed waveform with 50% duty cycle from dual Keithley Model 2651As into a high power LED module.

Just like the single SMU function, this function allows for a variable duty cycle as well. To create a waveform where the pulse width varies for each pulse it's necessary to pass the function a table of duty cycles. The following example calls demonstrate this.

```
dutyTable = {20, 40, 60, 80, 60, 40, 20, 40, 60}
PWM_Test_Dual(40, 10, 1000, dutyTable, 9, 1e-3)
```

This call will output a pulse width modulated waveform of 9 pulses with a pulse level of 40A, a 10V voltage limit, a frequency of 1kHz, and a varying duty cycle. The duty cycle for each pulse will be read from the table. The first pulse will have a duty cycle of 20%, the second pulse a duty cycle of 40%, the third pulse a duty cycle of 60% and so on. Spectrometer measurements will begin 1ms after the start of the waveform output. At the completion of the test, the SMU output will be turned off and the forward voltage measurements collected during the test will be printed to the instrument console in a format compatible for copy and pasting in to Microsoft Excel. An example of this output can be seen in ***Figure 15***.

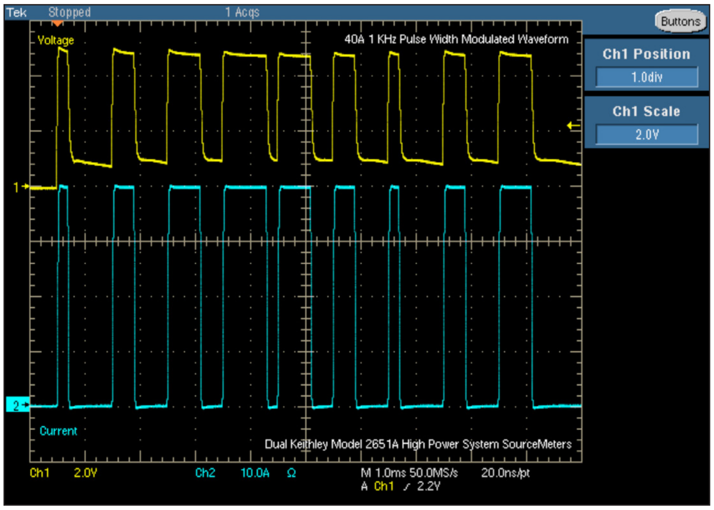


Figure 15: 40A 1kHz pulse width modulated waveform from dual Keithley Model 2651As into a high power LED module.

Conclusion

HBLEDs are advancing at an incredible pace and manufacturers are working hard to make them the lighting source choice of the future. In order to get there, LED manufacturers must jump several hurdles, ever trying to reduce the cost of manufacturing LEDs while simultaneously trying to increase their efficiency and light output. At the center of meeting these objectives, manufacturers require accurate, reliable, and repeatable source and measurement equipment with the power and flexibility to adapt to their ever-changing testing needs.

Series 2600B System SourceMeter instruments offer the features and flexibility to keep up with LED manufacturers' testing needs. Innovative features like the advanced trigger model allow Series 2600B instruments to make repeatable measurements accurately and reliably using complex drive schemes such as pulse width modulated waveforms, AC waveforms, and even arbitrary waveforms. The newest addition to this product line, the Model 2651A High Power System SourceMeter Instrument, provides additional power to handle even the brightest high power LED modules while still maintaining low current accuracy. Series 2600B System SourceMeter instruments, with their flexible output capabilities and their ability to source and measure accurately in a single instrument, make them the perfect choice for testing HBLEDs.

Appendix A: Source Code

NOTE: The code in this script will work without modification only with the Model 2651A High Power System SourceMeter instrument. However, this script can also work with other Series 2600B System SourceMeter instruments with only minor modifications.

```
--[[
   Title:       Pulse Width Modulation Script
   Desription:  The purpose of this script is to generate a pulse width
   modulated waveform for use in testing High Brightness LED modules.
   Users of this script should call the functions in the User Functions
   section.  Functions in the Utility Functions section are used by the
   User Functions to execute the test.

   System Setup:
      PWM_Test_Single()
         1x Model 2651A
      PWM_TEST_Dual()
         2x Model 2651A
         1x TSP-Link Cable

         Node 1: 2651A #1 (Master)
         Node 2: 2651A #2 (Slave)
]]--

--=================
-- User Functions
--=================
--[[ PWM_Test_Single()

   This function uses a single SMU to output a pulse width modulated waveform.
--]]
function PWM_Test_Single(pulseLevel, pulseLimit, frequency, dutyCycle, numPulses, specDelay)
   if (pulseLevel == nil) then pulseLevel = 1 end
   if (pulseLimit == nil) then pulseLimit = 1 end
   if (frequency == nil) then frequency = 100 end
   if (dutyCycle == nil) then dutyCycle = 1 end
   if (numPulses == nil) then numPulses = 10 end
   if (specDelay == nil) then specDelay = 0 end

   local pulsePeriod
   local pulseWidth
   local measDelay
   -- Calculate the timing parameters from the frequency and duty cycle
   pulsePeriod,pulseWidth,measDelay = CalculateTiming(frequency, dutyCycle)

   -- Do a quick check on the input parameters
   f,msg = SimpleRegionCheck(pulseLevel, pulseLimit, dutyCycle, 1)
   if (f == false) then
      print(msg)
      quit()
   end

   reset()
   smua.reset()
   smua.source.func                               = smua.OUTPUT_DCAMPS
   smua.sense                                            = smua.SENSE_REMOTE
   smua.source.autorangei                  = 0
   smua.source.rangei                             = pulseLevel
   smua.source.leveli                             = 0
   -- Set the DC bias limit.  This is not the limit used during the pulses.
   smua.source.limitv                             = 1
   smua.measure.autozero                   = smua.AUTOZERO_ONCE
   smua.measure.autorangev                 = 0
   smua.measure.rangev                            = pulseLimit
   -- The fast ADC allows us to place the measurements very close to the falling edge of
   -- the pulse allowing for settled measurements even when pulse widths are very small
   smua.measure.adc                               = smua.ADC_FAST
   smua.measure.count                             = 1
   smua.measure.interval                   = 1e-6
   -- Uncomment the following lines to turn on measure filtering.  When enabled, the SMU
   -- will take multiple measurements and average them to produce a single reading.
   -- Because the Fast ADC can take one measurement every microsecond, several measurements
   -- can be aquired in a small time to produce an averaged reading.
   --smua.measure.filter.count          = 5
   --smua.measure.filter.enable         = smua.FILTER_ON

   -- This measure delay sets the delay between the measurement trigger being received
   -- and when the actual measurement(s) start.  This is set to 0 because we will be
   -- delaying the trigger itself and do not need additional delay.
   smua.measure.delay                             = 0

   -- Setup the Reading Buffers
   smua.nvbuffer1.clear()
   smua.nvbuffer1.appendmode            = 1
   smua.nvbuffer1.collecttimestamps= 1
   smua.nvbuffer2.clear()
   smua.nvbuffer2.appendmode            = 1
   smua.nvbuffer2.collecttimestamps= 1

   -- Configure the Trigger Model
   --============================

   -- Timer 1 controls the pulse period
   trigger.timer[1].count                  = numPulses > 1 and numPulses - 1 or 1
   trigger.timer[1].delay                  = pulsePeriod
   trigger.timer[1].passthrough= true
   trigger.timer[1].stimulus            = smua.trigger.ARMED_EVENT_ID

   -- Timer 2 controls the pulse width
   trigger.timer[2].count                  = 1
   if (type(pulseWidth) == “table”) then
     -- Use a delay list if the duty cycle will vary for each pulse
     trigger.timer[2].delaylist         = pulseWidth
   else
     -- else every pulse will be the same duty cycle
     trigger.timer[2].delay             = pulseWidth
   end
   trigger.timer[2].passthrough= false
   trigger.timer[2].stimulus            = smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- Timer 3 controls the measurement
   trigger.timer[3].count                  = 1
   if (type(measDelay) == “table”) then
     -- If the duty cycle is variable then the measure delay will be as well
     trigger.timer[3].delaylist         = measDelay
   else
     trigger.timer[3].delay             = measDelay
   end
   trigger.timer[3].passthrough= false
   trigger.timer[3].stimulus            = smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- Configure SMU Trigger Model for Sweep
```

```
   smua.trigger.source.lineari(pulseLevel, pulseLevel, numPulses)
   smua.trigger.source.limitv            = pulseLimit
   smua.trigger.measure.action           = smua.ASYNC
   smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2)
   smua.trigger.endpulse.action= smua.SOURCE_IDLE
   smua.trigger.endsweep.action= smua.SOURCE_IDLE
   smua.trigger.count                              = numPulses
   smua.trigger.arm.stimulus             = 0
   smua.trigger.source.stimulus= trigger.timer[1].EVENT_ID
   smua.trigger.measure.stimulus         = trigger.timer[3].EVENT_ID
   smua.trigger.endpulse.stimulus        = trigger.timer[2].EVENT_ID
   smua.trigger.source.action            = smua.ENABLE

   -- Configure the Digital I/O trigger
   ConfigureSpectrometerTrigger(specDelay)

   -- Start the Test
   --================
   -- Turn the output on
   smua.source.output                              = 1
   -- Start the trigger model execution
   smua.trigger.initiate()

   -- While the trigger model is outputing the waveform and collecting the
   -- measurements, the script will scan the status model for any overruns
   -- that may occur as a result of using improper settings.
   local ovr = false
   local msg = ""
   while ((status.operation.sweeping.condition ~= 0) and (ovr == false)) do
      ovr, msg = CheckForOverRun(localnode)
   end
   if (ovr == true) then
      smua.abort()
      print(msg)
   end
   -- Turn the output off
   smua.source.output                              = 0
   -- Return the data
   PrintData()
end

--[[ PWM_Test_Dual()

   This function uses two SMUs connected together in parallel to ouput a pulse width
   modulated wavform.  By using two SMUs higher current levels/duty cycles can be achieved.
--]]
function PWM_Test_Dual(pulseLevel, pulseLimit, frequency, dutyCycle, numPulses, specDelay)
   if (pulseLevel == nil) then pulseLevel = 1 end
   if (pulseLimit == nil) then pulseLimit = 1 end
   if (frequency == nil) then frequency = 100 end
   if (dutyCycle == nil) then dutyCycle = 1 end
   if (numPulses == nil) then numPulses = 10 end
   if (specDelay == nil) then specDelay = 0 end

   local pulsePeriod
   local pulseWidth
   local measDelay

   -- Calculate the timing parameters from the frequency and duty cycle
   pulsePeriod,pulseWidth,measDelay = CalculateTiming(frequency, dutyCycle)

   -- Do a quick check on the input parameters
   f,msg = SimpleRegionCheck(pulseLevel, pulseLimit, dutyCycle, 2)
   if (f == false) then
      print(msg)
      quit()
   end

   -- Initialize the TSP-Link
   errorqueue.clear()
   tsplink.reset()
   errcode,errmsg,stat = errorqueue.next()
   if (errcode ~= 0) then
      print(errmsg)
      exit()
   end
   reset()
   ConfigureLocalSMU(pulseLevel, pulseLimit, pulsePeriod, pulseWidth, measDelay, numPulses)
   ConfigureRemoteSMU(pulseLevel, pulseLimit, pulsePeriod, pulseWidth, measDelay, numPulses)

   -- Start the Test
   --================
   -- Turn the output on
   smua.source.output                                          = 1
   node[2].smua.source.output                    = 1
   -- Start the trigger model execution
   node[2].smua.trigger.initiate()
   smua.trigger.initiate()

   -- While the trigger model is outputing the waveform and collecting the
   -- measurements, the script will scan the status model for any overruns
   -- that may occur as a result of using improper settings.
   local ovr1 = false
   local ovr2 = false
   local msg1 = “”
   local msg2 = “”
   -- Loop until the sweep is either complete, or an overrun condition is detected
   while (((status.operation.sweeping.condition ~= 0) or (node[2].status.operation.sweeping.condition ~=
0)) and (ovr1 == false) and (ovr2 == false)) do
      ovr1, msg1 = CheckForOverRun(localnode)
      ovr2, msg2 = CheckForOverRun(node[2])
   end
   if ((ovr1 == true) or (ovr2 == true)) then
      smua.abort()
      node[2].smua.abort()
      print(“SMU#1:”, msg1)
      print(“SMU#2:”, msg2)
   end
   -- Turn the output off
   node[2].smua.source.output                    = 0
   smua.source.output                                          = 0
   -- Return the data
   PrintDataDual()
end


--====================
-- Utility Functions
--====================
function ConfigureLocalSMU(pulseLevel, pulseLimit, pulsePeriod, pulseWidth, measDelay, numPulses)
   smua.reset()
   smua.source.func                              = smua.OUTPUT_DCAMPS
   smua.sense                                           = smua.SENSE_REMOTE
   smua.source.autorangei                 = 0
   smua.source.rangei                             = pulseLevel/2
   smua.source.leveli                             = 0
   -- Set the DC bias limit.  This is not the limit used during the pulses.
   smua.source.limitv                             = 1
   smua.source.offmode                            = smua.OUTPUT_NORMAL
   smua.source.offfunc                            = smua.OUTPUT_DCVOLTS
   smua.source.offlimiti                  = 1e-3

   smua.measure.autozero                  = smua.AUTOZERO_ONCE
   smua.measure.autorangev                = 0
   smua.measure.rangev                            = pulseLimit
   -- The fast ADC allows us to place the measurements very close to the falling edge of
   -- the pulse allowing for settled measurements even when pulse widths are very small
   smua.measure.adc                               = smua.ADC_FAST
   smua.measure.count                             = 1
   smua.measure.interval                  = 1e-6
   -- Uncomment the following lines to turn on measure filtering.  When enabled, the SMU
   -- will take multiple measurements and average them to produce a single reading.
   -- Because the Fast ADC can take one measurement every microsecond, several measurements
   -- can be aquired in a small time to produce an averaged reading.
   --smua.measure.filter.count          = 5
   --smua.measure.filter.enable         = smua.FILTER_ON

   -- This measure delay sets the delay between the measurement trigger being received
   -- and when the actual measurement(s) start.  This is set to 0 because we will be
   -- delaying the trigger itself and do not need additional delay.
   smua.measure.delay                             = 0

   -- Setup the Reading Buffers
   smua.nvbuffer1.clear()
   smua.nvbuffer1.appendmode            = 1
   smua.nvbuffer1.collecttimestamps= 1
   smua.nvbuffer2.clear()
   smua.nvbuffer2.appendmode            = 1
   smua.nvbuffer2.collecttimestamps= 1

   -- Configure the Trigger Model
   --============================

   -- Timer 1 controls the pulse period
   trigger.timer[1].count                 = (numPulses > 1) and numPulses - 1 or 1
   trigger.timer[1].delay                 = pulsePeriod
   trigger.timer[1].passthrough= true
   trigger.timer[1].stimulus            = smua.trigger.ARMED_EVENT_ID

   -- Timer 2 controls the pulse width
   trigger.timer[2].count                 = 1
   if (type(pulseWidth) == “table”) then
      -- Use a delay list if the duty cycle will vary for each pulse
      trigger.timer[2].delaylist        = pulseWidth
   else
      -- else every pulse will be the same duty cycle
      trigger.timer[2].delay            = pulseWidth
   end
   trigger.timer[2].passthrough= false
   trigger.timer[2].stimulus            = smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- Timer 3 controls the measurement delay
   trigger.timer[3].count                 = 1
   if (type(measDelay) == “table”) then
      -- If the duty cycle is variable then the measure delay will be as well
      trigger.timer[3].delaylist        = measDelay
   else
      trigger.timer[3].delay            = measDelay
   end
   trigger.timer[3].passthrough= false
   trigger.timer[3].stimulus            = smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- TSP-Link Trigger 1 is used to synchronize the SMUs by telling
   -- the second SMU when to pulse.
   tsplink.trigger[1].clear()
   tsplink.trigger[1].mode                 = tsplink.TRIG_FALLING
   tsplink.trigger[1].stimulus          = trigger.timer[1].EVENT_ID

   -- Configure SMU Trigger Model for Sweep
   smua.trigger.source.lineari(pulseLevel/2, pulseLevel/2, numPulses)
   smua.trigger.source.limitv           = pulseLimit
   smua.trigger.measure.action          = smua.ASYNC
   smua.trigger.measure.iv(smua.nvbuffer1, smua.nvbuffer2)
   smua.trigger.endpulse.action= smua.SOURCE_IDLE
   smua.trigger.endsweep.action= smua.SOURCE_IDLE
   smua.trigger.count                             = numPulses
   smua.trigger.arm.stimulus            = 0
   smua.trigger.source.stimulus= trigger.timer[1].EVENT_ID
   smua.trigger.measure.stimulus        = trigger.timer[3].EVENT_ID
   smua.trigger.endpulse.stimulus       = trigger.timer[2].EVENT_ID
   smua.trigger.source.action           = smua.ENABLE
end

function ConfigureRemoteSMU(pulseLevel, pulseLimit, pulsePeriod, pulseWidth, measDelay, numPulses)
   node[2].smua.reset()
   node[2].smua.source.func                       = node[2].smua.OUTPUT_DCAMPS
   node[2].smua.sense                                           = node[2].smua.SENSE_REMOTE
   node[2].smua.source.autorangei                 = 0
   node[2].smua.source.rangei                     = pulseLevel/2
   node[2].smua.source.leveli                     = 0
   -- Set the DC bias limit.  This is not the limit used during the pulses.
   node[2].smua.source.limitv                     = 1
   node[2].smua.source.offmode                    = node[2].smua.OUTPUT_NORMAL
   node[2].smua.source.offfunc                    = node[2].smua.OUTPUT_DCAMPS
   node[2].smua.source.offlimitv                  = 40

   node[2].smua.measure.autozero                  = node[2].smua.AUTOZERO_ONCE
   node[2].smua.measure.autorangev                = 0
   node[2].smua.measure.rangev                    = pulseLimit
   -- The fast ADC allows us to place the measurements very close to the falling edge of
   -- the pulse allowing for settled measurements even when pulse widths are very small
   node[2].smua.measure.adc                       = node[2].smua.ADC_FAST
   node[2].smua.measure.count                     = 1
   node[2].smua.measure.interval                  = 1e-6
   -- Uncomment the following lines to turn on measure filtering.  When enabled, the SMU
   -- will take multiple measurements and average them to produce a single reading.
   -- Because the Fast ADC can take one measurement every microsecond, several measurements
   -- can be aquired in a small time to produce an averaged reading.
   --node[2].smua.measure.filter.count     = 5
   --node[2].smua.measure.filter.enable    = node[2].smua.FILTER_ON

   -- This measure delay sets the delay between the measurement trigger being received
   -- and when the actual measurement(s) start.  This is set to 0 because we will be
   -- delaying the trigger itself and do not need additional delay.
   node[2].smua.measure.delay                     = 0

   -- Setup the Reading Buffers
   node[2].smua.nvbuffer1.clear()
   node[2].smua.nvbuffer1.appendmode       = 1
   node[2].smua.nvbuffer1.collecttimestamps= 1
   node[2].smua.nvbuffer2.clear()
   node[2].smua.nvbuffer2.appendmode       = 1
   node[2].smua.nvbuffer2.collecttimestamps= 1

   -- Configure the Trigger Model
   --============================

   -- Timer 2 controls the pulse width
   node[2].trigger.timer[2].count                 = 1
   if (type(pulseWidth) == “table”) then
     -- Use a delay list if the duty cycle will vary for each pulse
     node[2].trigger.timer[2].delaylist= pulseWidth
   else
     -- else every pulse will be the same duty cycle
     node[2].trigger.timer[2].delay        = pulseWidth
   end
   node[2].trigger.timer[2].passthrough = false
   node[2].trigger.timer[2].stimulus       = node[2].smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- Timer 3 controls the measurement delay
   node[2].trigger.timer[3].count                 = 1
   if (type(measDelay) == “table”) then
     -- If the duty cycle is variable then the measure delay will be as well
     node[2].trigger.timer[3].delaylist= measDelay
   else
     node[2].trigger.timer[3].delay        = measDelay
   end
   node[2].trigger.timer[3].passthrough = false
   node[2].trigger.timer[3].stimulus       = node[2].smua.trigger.SOURCE_COMPLETE_EVENT_ID

   -- TSP-Link Trigger 1 is used to synchronize the SMUs.  SMU #2 receives
   -- its trigger to pulse from SMU #1
   node[2].tsplink.trigger[1].clear()
   node[2].tsplink.trigger[1].mode                   = node[2].tsplink.TRIG_FALLING
   -- Release the trigger line when the pulse is complete
   node[2].tsplink.trigger[1].stimulus     = 0

   -- Configure SMU Trigger Model for Sweep
   node[2].smua.trigger.source.lineari(pulseLevel/2, pulseLevel/2, numPulses)
   node[2].smua.trigger.source.limitv      = pulseLimit
   node[2].smua.trigger.measure.action     = node[2].smua.ASYNC
   node[2].smua.trigger.measure.iv(node[2].smua.nvbuffer1, node[2].smua.nvbuffer2)
   node[2].smua.trigger.endpulse.action = node[2].smua.SOURCE_IDLE
   node[2].smua.trigger.endsweep.action = node[2].smua.SOURCE_IDLE
   node[2].smua.trigger.count                     = numPulses
   node[2].smua.trigger.arm.stimulus       = 0
   node[2].smua.trigger.source.stimulus = node[2].tsplink.trigger[1].EVENT_ID
   node[2].smua.trigger.measure.stimulus  = node[2].trigger.timer[3].EVENT_ID
   node[2].smua.trigger.endpulse.stimulus  = node[2].trigger.timer[2].EVENT_ID
   node[2].smua.trigger.source.action      = node[2].smua.ENABLE
end

function ConfigureSpectrometerTrigger(specDelay)
   -- Digital I/O line 1 triggers the spectrometer measurements
   -- Timer 4 puts a delay between the start of the pulse train and the
   -- output of the digital IO trigger on Digital I/O line 1
   digio.trigger[1].clear()
   digio.trigger[1].mode                = digio.TRIG_FALLING

   -- If the delay value is > 0 then configure a timer to provide the delay
   if specDelay > 0 then
      trigger.timer[4].count               = 1
      trigger.timer[4].delay               = specDelay
      trigger.timer[4].passthrough       = false
      trigger.timer[4].stimulus          = smua.trigger.ARMED_EVENT_ID

      digio.trigger[1].stimulus          = trigger.timer[4].EVENT_ID
   else
      -- Else bypass the timer and trigger the digital I/O immediately
      -- Configure the Digital I/O pin that will trigger the spectrometer
      digio.trigger[1].stimulus          = smua.trigger.ARMED_EVENT_ID
   end
end

function CheckForOverRun(pNode)
   -- Check SMUA Trigger Overruns
   if (bit.bitand(pNode.status.operation.instrument.smua.trigger_overrun.condition, 2) == 2) then
      return true, “smua arm trigger is overrun”
   end
   if (bit.bitand(pNode.status.operation.instrument.smua.trigger_overrun.condition, 4) == 4) then
      return true, “smua source trigger is overrun”
   end
   if (bit.bitand(pNode.status.operation.instrument.smua.trigger_overrun.condition, 8) == 8) then
      return true, “smua measure trigger is overrun”
   end
   if (bit.bitand(pNode.status.operation.instrument.smua.trigger_overrun.condition, 16) == 16) then
      return true, “smua endpulse trigger is overrun”
   end

   local CFORi = 0
   -- Check Timers for Overrun
   if (pNode.status.operation.instrument.trigger_timer.trigger_overrun.condition > 0) then
      return true, string.format(“Timer trigger is overrun: 0x%x”, CFORi)
   end

   -- Check Blenders for Overrun
   if (pNode.status.operation.instrument.trigger_blender.trigger_overrun.condition > 0) then
      return true, string.format(“blender trigger is overrun: 0x%x”, CFORi)
   end

   -- Check TSP-Link Triggers for Overrun
   if (pNode.status.operation.instrument.tsplink.trigger_overrun.condition > 0) then
      return true, string.format(“TSP-Link trigger is overrun: 0x%x”, CFORi)
   end

   -- Check DIGIO Triggers for Overrun
   if (pNode.status.operation.instrument.digio.trigger_overrun.condition > 0) then
      return true, string.format(“digio trigger is overrun: 0x%x”, CFORi)
   end

   -- Check LAN Triggers for Overrun
   if (pNode.status.operation.instrument.lan.trigger_overrun.condition > 0) then
      return true, string.format(“LAN trigger is overrun: 0x%x”, CFORi)
   end

   return false, “no overrun detected”
end

function PrintData()
   print(“Timestamp\tVoltage\tCurrent”)
   for i=1,smua.nvbuffer1.n do
      print(smua.nvbuffer1.timestamps[i], smua.nvbuffer2[i], smua.nvbuffer1[i])
   end
end

function PrintDataDual()
   local voltage
   local current
   print(“Timestamp\tVoltage\tCurrent”)
   for i=1,smua.nvbuffer1.n do
      voltage = (smua.nvbuffer2[i] + node[2].smua.nvbuffer2[i])/2
      current = smua.nvbuffer1[i] + node[2].smua.nvbuffer1[i]
      print(smua.nvbuffer1.timestamps[i], voltage, current)
   end
end

function CalculateTiming(frequency, dutyCycle)
   local pulsePeriod                              = 1/frequency
   local pulseWidth
   local measDelay

   -- If duty cycle was a table then we need to create delay lists for the timers
   if (type(dutyCycle)==”table”) then
      pulseWidth = {}
      measDelay  = {}
      for i=1,table.getn(dutyCycle) do
         if ((dutyCycle[i] > 99) or (dutyCycle[i] < 0.01)) then
            print(string.format(“Error: dutyCycle[%d] must be between 0.01% and 99%.”, i))
            exit()
         end
         -- Calculate pulse width from period and duty cycle.  Subtract 3us of overhead
         pulseWidth[i]          = pulsePeriod  * (dutyCycle[i]/100) - 3e-6
         -- Set measure delay so measurement happen 10us before the falling edge of the pulse
         measDelay[i]           = pulseWidth[i] - 10e-6
      end
   else  -- Duty cycle was a single value so we only need a single delay value for the timers
      if ((dutyCycle > 99) or (dutyCycle < 0.01)) then
         print(“Error: dutyCycle must be between 0.01% and 99%.”)
         exit()
      end
      pulseWidth                         = pulsePeriod  * (dutyCycle/100) - 3e-6
      measDelay                          = pulseWidth - 10e-6
   end
   return pulsePeriod, pulseWidth, measDelay
end

function SimpleRegionCheck(pulseLevel, pulseLimit, dutyCycle, SMUs)
   -- This function only serves as a quick check that the entered parameters are
   -- within the max allowable duty cycles for the operating regions.  This function
   -- does not check that the pulse widths are within the maximums as well.

   local pLev = math.abs(pulseLevel)
   f = true
   msg = “Checks passed.”
   if ((pulseLimit >= 10e-3) and (pulseLimit <= 10)) then
      if ((pLev > 30*SMUs) and (dutyCycle > 35)) then
         msg = string.format(“Duty Cycle too high for pulse region 5.  Duty cycle must be 35%% or less
for pulse levels above %dA.”, 30*SMUs)
         f = false
      elseif (((pLev > 20*SMUs) and (pLev <= 30*SMUs)) and (dutyCycle > 50)) then
         msg = string.format(“Duty Cycle too high for pulse region 2.  Duty cycle must be 50%% or less
for pulse levels between %dA and %dA.”, 20*SMUs, 30*SMUs)
         f = false
      end
   elseif ((pulseLimit > 10) and (pulseLimit <= 20)) then
      if ((pLev > 20*SMUs) and (dutyCycle > 10)) then
         msg = string.format(“Duty Cycle too high for pulse region 6.  Duty cycle must be 10%% or less
for pulse levels above %dA.”, 20*SMUs)
         f = false
      elseif (((pLev > 10*SMUs) and (pLev <= 20*SMUs)) and (dutyCycle > 40)) then
         msg = string.format(“Duty Cycle too high for pulse region 3.  Duty cycle must be 40%% or less
for pulse levels between %dA and %dA.”, 10*SMUs, 20*SMUs)
         f = false
      end
   elseif (pulseLimit > 20) and (pulseLimit <= 40) then
      if ((pLev > 10*SMUs) and (dutyCycle > 1)) then
         msg = string.format(“Duty Cycle too high for pulse region 7.  Duty cycle must be 1%% or less
for pulse levels above %dA.”, 10*SMUs)
         f = false
      elseif (((pLev > 5*SMUs) and (pLev <= 10*SMUs)) and (dutyCycle > 40)) then
         msg = string.format(“Duty Cycle too high for pulse region 4.  Duty cycle must be 40%% or less
for pulse levels between %dA and %dA.”, 5*SMUs, 10*SMUs)
         f = false
      end
   else
      msg = “Error: pulseLimit out of range.  pulseLimit must be between 10mV and 40V.”
      f = false
   end

   return f,msg
end


--PWM_Test_Single(1, 2, 100, 1, 10, 0)
-- duty = {20, 40, 60, 80, 60, 40, 20, 40, 60}
--PWM_Test_Single(20, 10, 1000, duty, 9, 1e-3)
--PWM_Test_Dual(40, 10, 1000, duty, 9, 1e-3)
```

Contact Information:

ASEAN / Australia	(65) 6356 3900
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
Italy	00800 2255 4835
Japan	81 (3) 6714 3010
Luxembourg	+41 52 675 3777
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
Norway	800 16098
People's Republic of China	400 820 5835
Poland	+41 52 675 3777
Portugal	80 08 12370
Republic of Korea	001 800 8255 2835
Russia / CIS	+7 (495) 6647564
South Africa	+41 52 675 3777
Spain	00800 2255 4835
Sweden	00800 2255 4835
Switzerland	00800 2255 4835
Taiwan	886 (2) 2656 6688
United Kingdom / Ireland	00800 2255 4835
USA	1 800 833 9200

Rev. 01/16

TEK.COM

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit tek.com

01/16 1KW-17996-1