

# Model 4200-SCS Semiconductor Characterization System

## User's Manual

4200-900-01 Rev. F / September 2010



**Model 4200-SCS**  
**Product Test Software License Agreement**

NOTICE TO USERS: CAREFULLY READ THE FOLLOWING 4200 PRODUCT TEST SOFTWARE LICENSE AGREEMENT (THIS "AGREEMENT"). BY INSTALLING, COPYING OR OTHERWISE USING ALL OR ANY PORTION OF THE LICENSED PRODUCT (AS DEFINED BELOW), YOU ACCEPT ALL TERMS AND CONDITIONS OF THIS AGREEMENT. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU. THIS AGREEMENT IS ENFORCEABLE AGAINST YOU AND ANY LEGAL ENTITY THAT OBTAINED THE LICENSED PRODUCT AND ON WHOSE BEHALF IT IS USED. **IF YOU DO NOT AGREE, DO NOT INSTALL OR USE THE LICENSED PRODUCT.** IF YOU ACQUIRED THE LICENSED PRODUCT ON TANGIBLE MEDIA (E.G., CD) WITHOUT AN OPPORTUNITY TO REVIEW THIS LICENSE AND YOU DO NOT ACCEPT THIS AGREEMENT, YOU MAY OBTAIN A REFUND OF ANY AMOUNT YOU ORIGINALLY PAID FOR THE LICENSED SOFTWARE, IF YOU: (A) DO NOT USE THE LICENSED PRODUCT AND (B) RETURN IT, WITH PROOF OF PAYMENT, TO THE LOCATION FROM WHICH IT WAS OBTAINED WITHIN THIRTY (30) DAYS OF THE PURCHASE DATE.

1. **Definitions.** "**Licensed Product**" means all of the contents of the files, disk(s), CD-ROM(s) or other media with which this Agreement is provided, including but not limited to (i) the current version of the KTEI software ("Software"), and (ii) any related explanatory written materials and manuals pertaining to the Software provided by Keithley with the Software ("Documentation"), and means all upgrades, modified versions, updates, and additional features and functionality relating to the Software, if any, licensed to you by Keithley pursuant to Section 8 of this Agreement ("Upgrades").

"**4200 Product**" shall mean the 4200-SCS Semiconductor Characterization System.

"**You**" means the individual using the Licensed Product as well as your employer ("Company").

2. **Grant of License.** Subject to Licensee's full compliance with the terms and conditions of this Agreement, and payment of any fees owed to Keithley (if applicable, which payment shall be non-refundable), Keithley hereby grants to you a non-exclusive and nontransferable license to use the Software solely for the purpose of testing and demonstrating the functionality of 4200 Product, and the Software shall be used only on a permitted maximum number of central processing units (CPUs) as stated in an invoice or order confirmation issued by Keithley. If a maximum number of CPUs is not so indicated, the Software may be stored and used on no more than four CPUs. The Licensed Product is licensed solely for internal business use within your organization and should not be used for the benefit of any other person or entity. The license is not a site or enterprise-wide license. The Licensed Product is licensed for use by a single organization and may not be used for processing of any third party data as a service bureau.
3. **Ownership.** Keithley and its third party suppliers own all right, title and interest in and to the Licensed Product. You acknowledge that all right, title and interest in and to the Licensed Product will remain the exclusive property of Keithley and its suppliers, and you will not acquire any rights in or to the Licensed Product except as expressly set forth in this Agreement. The Licensed Product is protected by law, including without limitation the copyright laws of the United States and other countries, trade secret laws, and by international treaty provisions. Except as expressly stated herein, this Agreement does not grant you any intellectual property rights in the Licensed Product and all rights not expressly granted are reserved by Keithley and its third party suppliers.
4. **Limitations on Use.** You will not remove any copyright, trademark or other proprietary notices from the Licensed Product or any media relating thereto. You agree that you will not (i) attempt to reverse compile, reverse engineer, modify, translate, adapt or disassemble the Licensed Product, nor attempt to discover, create the source code from the object code for the Licensed Product, in whole or in part, and (ii) rent, copy, lease, assign or transfer your rights in the Licensed Product, or authorize all or any portion of the Licensed Product to be copied onto any products or another user's computer except as may be expressly permitted herein.
5. **Termination.** Keithley shall have the right to terminate this Agreement at any time by providing notice of termination. This Agreement hereunder may be terminated by Keithley: (i) if Licensee fails to comply with section titled "Export Restriction", (ii) in the event of a material breach by Licensee of any provision of this Agreement where Licensee fails to correct such breach within ten (10) days of written notice, (iii) for failure to pay any required fees, or (iv) upon the insolvency, bankruptcy, reorganization, or assignment for the benefit of creditors of Licensee. Within three (3) days after termination of this Agreement, Licensee shall furnish to Keithley written notice certifying that Licensee has erased or otherwise destroyed all copies of the Licensed Software that are fixed or resident in the memory or hard disks of computers owned or controlled by Licensee.



6. **Export Restrictions.** You warrant, agree and give assurance to Keithley that you will not at any time (i) sell, transmit, export, re-export, release or otherwise disclose, either directly or indirectly, the Licensed Product, or any part thereof, including without limitation any source code, or technology relating to the Licensed Product, (ii) permit Licensed Product to be sold, transmitted, disclosed, released, exported, re-exported to countries or nationals of countries in Country Groups D:1 or E:2 set forth in Supplement 1 to Part 740 of the United States Export Administration Regulation ("EAR"), or (iii) permit use of the Licensed Product in any manner prohibited by the EAR or any other export laws, restrictions or regulations. All rights to use the Licensed Product are granted on condition that such rights shall be immediately forfeited if you fail to comply with the terms of this Section 6.
7. **U.S. Government Restricted Rights.** The Licensed Product is "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (i) only as Commercial Items and (ii) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States to Keithley Instruments, 28775 Aurora Rd., Solon, Ohio 44139, USA.
8. **No Support and Upgrades. NOTHING IN THIS AGREEMENT ENTITLES YOU TO ANY SUPPORT, MAINTENANCE, OR UPGRADES OF THE SOFTWARE.** You may contact Keithley to determine the availability of support, maintenance, and/or Upgrades of the Software, and the fees, terms, and conditions that may apply. Except as otherwise expressly provided herein, this Agreement will govern any Upgrades that may be provided to you by Keithley, unless such Upgrades are provided under separate license agreement. You may use such Upgrades only in conjunction with your then-existing Software licensed under this Agreement. The Software and Upgrades are licensed as a single product and such Upgrades may not be separated from the Software to exceed the scope of your original license. Notwithstanding anything to the contrary set forth in this Agreement, the Upgrades shall be used only on such number of CPUs as specified in any purchase order of terms and conditions of purchase of such Upgrades. Any other software that may be provided with the Software that is associated with a separate end-user license agreement is licensed to you under the terms of that separate license agreement.
9. **NOTICE.** THE LICENSED PRODUCT IS NOT INTENDED OR LICENSED FOR USE IN ANY HAZARDOUS OR HIGH RISK ACTIVITY.
10. **DISCLAIMER OF WARRANTY.** THE LICENSED PRODUCT IS PROVIDED "AS IS" AND "AS AVAILABLE" WITHOUT ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND. KEITHLEY SPECIFICALLY DISCLAIMS ALL REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, USAGE, CUSTOM, TITLE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS, INTEGRATION, ACCURACY OF ANY TEST RESULTS, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE AND IMPLIED WARRANTY ARISING FROM COURSE OF PERFORMANCE OR COURSE OF DEALING. NEITHER KEITHLEY NOR ITS THIRD PARTY SUPPLIERS REPRESENT OR WARRANT THAT THE LICENSED PRODUCT WILL SATISFY YOUR REQUIREMENTS OR THAT IT IS WITHOUT DEFECT OR ERROR OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED OR ERROR FREE.
11. **EXCLUSION OF CERTAIN DAMAGES; LIMITATION OF LIABILITY.** TO THE FULL EXTENT PERMITTED BY LAW, NEITHER KEITHLEY NOR ITS THIRD PARTY SUPPLIERS SHALL IN ANY CASE BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY LOSS OF PROFITS, REVENUE, BUSINESS, OR DATA, COST OF COVER, EQUIPMENT ARISING FROM OF RELATING TO THIS AGREEMENT OR THE USE OR INABILITY TO USE THE LICENSED PRODUCT, REGARDLESS OF THE FORM OF ACTION AND EVEN IF KEITHLEY HAS BEEN ADVISED OF SUCH DAMAGE. Some jurisdictions do not allow certain exclusions or limitations of liability, so some of these may not apply to you. THE AGGREGATE LIABILITY OF KEITHLEY OR ITS THIRD PARTY SUPPLIERS ARISING FROM OR RELATING TO THIS AGREEMENT OR THE USE OR INABILITY TO USE THE LICENSED PRODUCT (REGARDLESS OF THE FORM OF ACTION OR CLAIM, WHETHER CONTRACT, WARRANTY, TORT AND/OR OTHERWISE) SHALL NOT EXCEED THE LESSER OF (i) AMOUNT OF ALL LICENSE FEES PAID BY YOU FOR THE LICENSED PRODUCT LICENSE(S) GRANTED HEREUNDER, OR (ii) US\$500.
12. **Compliance with Licenses.** If you are a business or organization, you agree that upon request from Keithley or Keithley's authorized representative, you will within thirty (30) days after receiving notice from Keithley fully document and certify that use of any and all Licensed Product at the time of the request is in conformity with your valid licenses from Keithley. You agree and Company shall make all applicable records available for review by Keithley during Company's normal business hours so as to permit Keithley (upon reasonable notice to you) to verify your and Company's compliance with the terms and conditions of this Agreement. Keithley may (upon written notice) inspect your use of the Licensed Products during Company's normal business hours to ensure your compliance with this Agreement.



13. **Miscellaneous.** In the event of invalidity of any provision of this Agreement, the parties agree that such invalidity shall not affect the validity of the remaining portions of this Agreement. This Agreement shall be governed by and construed in accordance with the laws of the state of Ohio, exclusive of any provisions of the United Nation Convention on the International Sale of Goods and without regard to any principle of conflicts of laws. This is the entire agreement between you and Keithley and supersedes any prior agreement or understanding, whether written or oral, relating to the subject matter of this license. To the extent the terms of any purchase order, or any Keithley policies or programs for providing any support services for the Licensed Product conflict with the terms of this Agreement, the terms of this Agreement shall control. This Agreement may only be modified by a writing signed by an authorized officer of Keithley. Any waiver by either party of any provision of this Agreement shall not constitute or be deemed a subsequent waiver of that or any other provision. The provisions of the Sections titled Ownership, Limitations on Use, Export Restrictions, Limited Warranty, Limitation of Liability, Compliance with Licenses, and Miscellaneous shall survive the termination of this Agreement.

Should you have any questions concerning this Agreement, or if you desire to contact Keithley Instruments for any reason, please call Keithley at 1-800-552-1115, or write at Keithley Instruments, 28775 Aurora Rd., Solon, Ohio, USA 44139.

#### **Limited Hardware Warranty**

Keithley warrants to you that the Keithley manufactured portion of the hardware (the "Keithley Hardware") purchased by you will substantially perform in accordance with the specifications set forth in this manual for a period of one (1) year after your receipt of the Keithley Hardware (the "Warranty Period"); provided the Keithley Hardware is used on the products for which it is intended and in accordance with the documentation. This limited warranty shall be null and void upon (1) any modifications of the Keithley Hardware, unless approved in writing by Keithley, (2) any operation of the 4200-Semiconductor Characterization System (the "4200-SCS") with third party software, unless the software is explicitly approved and supported by Keithley, and (3) any operation of the 4200-SCS on an operating system not explicitly approved and supported by Keithley.

If the Keithley Hardware is not performing as warranted during the Warranty Period, as determined in Keithley's sole discretion (a "Nonconformity"), your exclusive remedy under this limited warranty is the repair or replacement of the Keithley Hardware, at Keithley's option. The portions of the hardware not developed and owned by Keithley shall not be covered by this limited hardware warranty, and Keithley shall have no duty or obligation to enforce a third party supplier's warranties on your behalf. The failure to notify Keithley of a Nonconformity during the Warranty Period shall relieve Keithley of its obligations and liabilities under this limited hardware warranty.

EXCEPT FOR THE FOREGOING, THE HARDWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

KEITHLEY'S SOLE LIABILITY OR OBLIGATION UNDER THIS LIMITED HARDWARE WARRANTY IS THE REPAIR OR REPLACEMENT OF THE KEITHLEY HARDWARE. IN NO EVENT SHALL KEITHLEY BE LIABLE FOR ANY DAMAGES. WITHOUT LIMITING THE FOREGOING, KEITHLEY SHALL NOT BE LIABLE OR ASSUME LIABILITY FOR: (1) ECONOMICAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, PUNITIVE OR EXEMPLARY DAMAGES, WHETHER CLAIMED UNDER CONTRACT, TORT OR ANY OTHER LEGAL THEORY, (2) LOSS OF OR DAMAGE TO YOUR DATA OR PROGRAMMING, (3) PENALTIES OR PENALTY CLAUSES OF ANY DESCRIPTION, OR (4) INDEMNIFICATION OF YOU OR OTHERS FOR COSTS, DAMAGES OR EXPENSES RELATED TO THE GOODS OR SERVICES PROVIDED UNDER THIS LIMITED HARDWARE WARRANTY.



# Model 4200-SCS Semiconductor Characterization System User's Manual

## KTE Interactive Version 8.1

©2010, Keithley Instruments, Inc.

All rights reserved.

Any unauthorized reproduction, photocopy, or use of the information herein, in whole or in part, without the prior written approval of Keithley Instruments, Inc. is strictly prohibited.

All Keithley Instruments product names are trademarks or registered trademarks of Keithley Instruments, Inc. Other brand names are trademarks or registered trademarks of their respective holders.

Document Number: 4200-900-01 Rev. F / September 2010



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

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

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

The types of product users are:

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

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

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

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

Keithley Instruments products are designed for use with electrical signals that are rated Measurement Category I and Measurement Category II, as described in the International Electrotechnical Commission (IEC) Standard IEC 60664. Most measurement, control, and data I/O signals are Measurement Category I and must not be directly connected to mains voltage or to voltage sources with high transient over-voltages. Measurement Category II connections require protection for high transient over-voltages often associated with local AC mains connections. Assume all measurement, control, and data I/O connections are for connection to Category I sources unless otherwise marked or described in the user documentation.

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

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

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

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



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

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

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


The instrument and accessories must be used in accordance with its specifications and operating instructions, or the safety of the equipment may be impaired.


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


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

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

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


If a  screw is present, connect it to safety earth ground using the wire recommended in the user documentation.

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

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

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

The  symbol indicates a connection terminal to the equipment frame.

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

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

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

Instrumentation and accessories shall not be connected to humans.

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

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

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



Section	Topic	Page
<b>1</b>	<b>Getting Started</b>	1-1
	Installation and system connections	1-3
	Unpacking the Model 4200-SCS	1-3
	Shipment contents	1-3
	Environmental Considerations	1-4
	Powering up the Model 4200-SCS	1-5
	System connections	1-7
	Model 4200-SCS Hardware Overview	1-11
	Front panel	1-11
	Rear Panel	1-12
	DC source-measure unit (SMU)	1-14
	DC preamp	1-20
	Multi-frequency capacitance / voltage unit (CVU)	1-23
	Introduction	1-23
	Model 4210-CVU card	1-23
	Force-measure timing	1-26
	Pulse generator card	1-27
	About the pulse generator card	1-28
	Firmware upgrade for the Model 4200-PG2	1-28
	Standard pulse	1-28
	Segment Arb™ waveform	1-29
	Remote bias tee (RBT) and 3-port power divider	1-32
	SCP2 (Oscilloscope)	1-35
	Ground unit (GNDU)	1-39
	Basic circuit configurations	1-40
	Ground unit terminals and connectors	1-41
	Connecting DUTs	1-43
	Test fixtures	1-43
	Probers	1-45
	Advanced connections	1-45
	How to run a basic test	1-45
	Boot the system and log in	1-45
	Open KITE	1-46
	Locate and run the vds-id test module	1-48
	Test definition	1-49
	Run vds-id test	1-50
	View and save the sheet data	1-50
	View and save the graph data	1-51
<b>2</b>	<b>Model 4200-SCS Software Environment</b>	2-1
	Understanding KITE	2-3
	KITE project structure	2-3
	KITE interface	2-3
	Project navigator	2-5
	Project defined	2-6
	Project components	2-7
	ITMs versus UTMs	2-8
	Defining an ITM	2-10
	Defining a UTM	2-11
	How to create your own ITMs	2-13
	Understanding the ITM definition tab	2-13
	Understanding the ITM forcing functions	2-13
	Understanding dual sweep	2-15
	Understanding pulse mode	2-16
	How to use the definition tab to configure ITM parameters	2-17
	Basic test execution	2-21



Project navigator check boxes .....	2-21
Executing an individual test .....	2-22
How to display and manage test results .....	2-24
Data file management .....	2-24
How to manage numeric test results in Sheet tab .....	2-31
How to manage graphical test results in the Graph tab .....	2-38
KITE library management .....	2-45
Submitting devices, ITMs, and UTMs to libraries .....	2-45
Submitting tests to a library .....	2-48
<b>3 Common Device Characterization Tests .....</b>	<b>3-1</b>
How to perform an I-V test on my device .....	3-4
Default project overview .....	3-4
How to perform a C-V test on my device .....	3-13
KITE ITM configuration .....	3-13
Definition tab .....	3-13
Forcing functions and measure options .....	3-15
Selecting the forcing function .....	3-16
CVU ITM examples .....	3-19
CVU Voltage Sweep .....	3-20
CVU Voltage List Sweep .....	3-21
CVU Frequency Sweep (bias) .....	3-22
CVU Frequency Sweep (step) .....	3-24
How to perform a Pulsed I-V test on my device .....	3-26
Introduction (PIV-A and PIV-Q) .....	3-26
Pulse IV for CMOS:Model 4200-PIV-A .....	3-27
4200-PIV-A test connections .....	3-28
Using the PulseIV-Complete project for the first time .....	3-35
Pulse IV UTM descriptions .....	3-46
cal_pulseiv .....	3-46
vdsid_pulseiv .....	3-48
Vdld_Pulse_DC_Family_pulseiv .....	3-50
vgsid_pulseiv .....	3-53
Vgld_DC_Pulse_pulseiv .....	3-55
scopeshot_cal_pulseiv .....	3-59
scopeshot_pulseiv .....	3-61
vdsid_pulseiv_demo .....	3-63
vgsid_pulseiv_demo .....	3-63
scopeshot_pulseiv_demo .....	3-63
How to perform a Quiescent-point Pulsed I-V test (PIV-Q) on my device .....	3-64
Q-Point Pulse IV – Model 4200-PIV-Q .....	3-64
What is the PIV-Q package .....	3-64
How to perform a reliability (stress-measure) tests on my device .....	3-66
Connecting devices for stress / measure cycling .....	3-66
Overviewing the cycling-related tabs .....	3-67
Configuring subsite cycling .....	3-67
Configuring device stress properties .....	3-71
How to perform AC stress for wafer level reliability (WLR) .....	3-75
Segment Stress / Measure Mode .....	3-81
Segment Arb™ stressing .....	3-82
Segment Stress / Measure Mode configuration .....	3-83
Executing subsite cycling .....	3-86
Subsite cycling data sheets .....	3-86
Subsite cycling graphs .....	3-90
Configuration sequence for subsite cycling .....	3-92
How to perform a flash memory test on my device .....	3-94
Introduction .....	3-94
Theory of operation .....	3-94
Flash connections .....	3-108
Direct connection to single DUT .....	3-112
Direct connection to array DUT for disturb testing .....	3-114
Switch matrix connection to array DUT .....	3-115
Flash projects .....	3-116
Flash-NAND tests .....	3-118
Flash-NOR tests .....	3-123



	Flash-switch tests .....	3-123
	Running any Flash Project for the first time .....	3-124
	Running the Flash-NAND, Flash-NOR or Flash-Switch Project .....	3-125
	Running the Program or Erase UTM .....	3-126
	Running the Fast-Program-Erase UTM .....	3-128
	Running the SetupDC UTM .....	3-129
	Running the Vt-MaxGm ITM .....	3-129
	Running the ConPin-Pulse or ConPin-DC UTM (Switch projects only) .....	3-129
	FlashEndurance-NAND tests .....	3-130
	Running a FlashEndurance or FlashDisturb project .....	3-137
	Running endurance or disturb looping .....	3-139
	FlashDisturb tests .....	3-140
	Explanation of flash UTM parameters .....	3-141
	Error codes .....	3-143
	Troubleshooting .....	3-144
	How to perform Charge Pumping .....	3-145
	How to perform a Charge Trapping test .....	3-146
	Slow single pulse charge trapping high K gate stack .....	3-146
	Charge trapping procedure .....	3-146
<b>4</b>	<b>How to Control Other Instruments with the Model 4200-SCS .....</b>	<b>4-1</b>
	How to control external equipment .....	4-2
	Controlling external equipment overview .....	4-2
	Keithley Configuration Utility (KCON) .....	4-5
	How to control a switch matrix .....	4-7
	KCON setup .....	4-8
	Open KITE and the ivswitch project .....	4-11
	Running test sequences .....	4-11
	The connect test .....	4-13
	How to control a probe station .....	4-15
	Prober control overview .....	4-16
	Test system connections .....	4-17
	KCON setup .....	4-18
	Probe station configuration .....	4-20
	Open the probesubsites project .....	4-21
	Open the project plan window .....	4-21
	Test descriptions .....	4-22
	Running the test sequence .....	4-25
	Test data .....	4-26
	Running individual plans or tests .....	4-27
	How to control an external pulse generator .....	4-27
	Test system connections .....	4-28
	KCON setup .....	4-28
	Open the ivpgswitch project .....	4-31
	Description of tests .....	4-32
	First id-vg test .....	4-32
	pgu-trigger test .....	4-34
	Compare the test results .....	4-35
	How to control an external CV analyzer .....	4-37
	Connections .....	4-37
	KCON setup .....	4-38
	Create a new project .....	4-39
	Add a subsite plan .....	4-40
	Add a device plan .....	4-41
	Add a UTM .....	4-42
	Modifying the cvsweep UTM .....	4-43
	Executing the test .....	4-44
	What if my equipment is not listed in KCON .....	4-44
<b>5</b>	<b>How to Generate Basic Pulses .....</b>	<b>5-1</b>
	Keithley Pulse Application (KPulse) .....	5-2
	KPulse: Getting started .....	5-2
	Triggering .....	5-3
	Standard pulse waveforms .....	5-4
	Segment Arb™ waveforms .....	5-6



	Custom file arb waveforms (full-arb).....	5-8
<b>6</b>	<b>How to Control the 4200-SCS Remotely from a PC</b> .....	6-1
	Remote control overview (GPIB, Ethernet).....	6-2
	GPIB standards.....	6-2
	Communication connections.....	6-2
	Setting remote control mode (GPIB versus Ethernet).....	6-3
	KXCI control interface.....	6-4
	Starting KXCI and the GPIB command interpreter.....	6-4
<b>7</b>	<b>How to Create UTMs Using C Programming</b> .....	7-1
	Keithley User Library Tool (KULT).....	7-3
	Introduction.....	7-3
	KULT window.....	7-3
	Understanding the module identification area.....	7-4
	Understanding the module parameter display area.....	7-5
	Understanding the module code entry area.....	7-5
	Understanding the terminating brace area.....	7-6
	Understanding the tab areas.....	7-6
	Parameters tab area.....	7-6
	Includes tab area.....	7-8
	Description tab area.....	7-9
	Build tab area.....	7-10
	Understanding the status bar.....	7-10
	Understanding the menus.....	7-10
	File menu.....	7-11
	Edit menu.....	7-12
	Options menu.....	7-13
	Help menu.....	7-14
	KULT Tutorials.....	7-14
	Tutorial 1: Creating a new user library and user module.....	7-15
	Entering parameters.....	7-18
	Entering header files.....	7-20
	Tutorial 2: Creating a user module that returns data arrays.....	7-28
	Tutorial 3: Calling one user module from within another.....	7-34



---

## Section 1

# Getting Started

In this section:

Topic	Page
<b>Installation and system connections</b>	<b>1-3</b>
Unpacking the Model 4200-SCS	1-3
Inspection for damage	1-3
Shipment contents	1-3
Manual package	1-3
Repacking for shipment	1-3
Environmental Considerations	1-4
Shipping and storage environment	1-4
Operating environment	1-4
Powering up the Model 4200-SCS	1-5
Line power	1-5
Line power connection	1-6
Line frequency setting	1-6
Line fuses	1-7
Warm-up period	1-7
System connections	1-7
Connecting the keyboard and mouse	1-7
Connecting GPIB instruments	1-8
Connecting a probe station	1-9
Connecting a printer	1-10
Connecting a LAN	1-10
<b>Model 4200-SCS Hardware Overview</b>	<b>1-11</b>
Front panel	1-11
Instrument panels	1-11
Front panel	1-11
Rear Panel	1-11
Activate LAN2 connection	1-13
<b>DC source-measure unit (SMU)</b>	<b>1-13</b>
Introduction	1-13
Models 4200-SMU and 4210-SMU overview	1-13
Basic characteristics	1-13
Basic SMU circuit configuration	1-14
SMU terminals and connectors	1-16
SMU with Model 4200-PA overview	1-17
Basic characteristics	1-17
Basic SMU/preamp circuit configuration	1-18
<b>DC preamp</b>	<b>1-19</b>
PreAmp terminals and connectors	1-19
FORCE terminal	1-20
SENSE terminal	1-21
PreAmp CONTROL connector	1-21
<b>Multi-frequency capacitance / voltage unit (CVU)</b>	<b>1-21</b>
Introduction	1-21
Model 4210-CVU card	1-21
Measurement overview	1-21
Measurement functions	1-22



Test signal .....	1-23
DC bias function and sweep characteristics .....	1-23
Force-measure timing .....	1-24
Bias function .....	1-24
Sweep function .....	1-25
<b>Pulse cards .....</b>	<b>1-25</b>
About the pulse cards .....	1-26
Firmware upgrade for the Model 4200-PG2 .....	1-27
Standard pulse .....	1-27
Segment Arb™ waveform .....	1-27
Full arb .....	1-29
Pulse card settings .....	1-30
Remote bias tee (RBT) and 3-port power divider .....	1-30
RBT .....	1-30
3-port power divider .....	1-31
Using an RBT and power divider .....	1-31
SCP2 (Oscilloscope) .....	1-32
Digital storage oscilloscope card .....	1-32
Scope card settings .....	1-33
Ground unit (GNDU) .....	1-36
Basic characteristics .....	1-36
Basic circuit configurations .....	1-37
Ground unit connections .....	1-37
Ground unit DUT connections .....	1-38
Ground unit terminals and connectors .....	1-39
FORCE terminal .....	1-39
SENSE terminal .....	1-39
COMMON terminal .....	1-40
Chassis ground .....	1-40
<b>Connecting DUTs .....</b>	<b>1-40</b>
Test fixtures .....	1-40
Testing with less than $\pm 20$ V .....	1-41
Testing with more than $\pm 20$ V .....	1-41
Probers .....	1-42
Advanced connections .....	1-42
<b>How to run a basic test .....</b>	<b>1-42</b>
Boot the system and log in .....	1-42
Open KITE .....	1-43
Locate and run the vds-id test module .....	1-44
Test definition .....	1-45
Run vds-id test .....	1-46
View and save the sheet data .....	1-46
View and save the graph data .....	1-47



# Installation and system connections

## Unpacking the Model 4200-SCS

### Inspection for damage

After unpacking the Keithley Instruments Model 4200 Semiconductor Characterization System (SCS), carefully inspect the unit for any shipping damage. Report any damage to the shipping agent, because such damage is not covered by the warranty.

## Shipment contents

The following items are included with the Model 4200-SCS:

- Model 4200-SCS with any ordered source-measure units (SMUs) factory-installed
- Ordered Model 4200-PA modules factory-installed
- Ordered Model 4220-PGU pulse generator cards factory-installed
- Ordered Model 4225-PMU pulse/measure cards factory-installed
- Ordered Model 4225-RPM remote pulse (and switch) modules
- Ordered pulse application packages (For more information refer to the [Reference Manual, Pulsing: Source and measure options, page 1-9](#))
- Cables, connectors, adapters and other accessories that are supplied with the pulse generator, scope, and pulse application packages. (For more information about the lists of supplied accessories for the pulsing options refer to the [Reference Manual, Pulsing: Source and measure options, page 1-9](#)).
- Line cord
- Model 4200-Semiconductor Characterization System User's Manual
- Model 4200-SCS technical data
- Miniature triaxial cables, two per Model 4200-SMU or 4210-SMU, 2 m (6 ft)<sup>1</sup>
- Triaxial cables, two per Model 4200-PA, 2 m (6 ft)
- Interlock cable
- Keyboard with integrated pointing device
- System software and manuals on CD-ROM
- Microsoft® Windows® XP Professional
- Microsoft® Visual Studio® 2008 Standard Edition

### Manual package

All 4200 manuals are provided on a CD-ROM and are preinstalled on the hard drive. If a complete set of printed manuals is required, order the optional manual package (Keithley Instruments part number 4200-MAN). The manual package includes any pertinent addenda. Because the manuals are provided in PDF format, they can be printed from any computer that is connected to a printer by using Adobe® Reader®.

### Repacking for shipment

Should it become necessary to return the Model 4200-SCS for repair, carefully pack the entire unit in its original packing carton or the equivalent, and follow these instructions:

- Call Keithley Instruments' repair department at 1-888-KEITHLEY (1-888-534-8453) for a Return Material Authorization (RMA) number.

---

1. Not included when SMU is ordered with a Model 4200-PA.



- Let the repair department know the warranty status of the Model 4200-SCS Semiconductor Characterization System.
- Write ATTENTION REPAIR DEPARTMENT and the RMA number on the shipping label.

Complete and include the Service Form located at the back of this manual.

## Environmental Considerations

### Shipping and storage environment

To avoid possible damage or deterioration, the Model 4200-SCS should be shipped and stored within the following environmental limits:

- Temperature: -10° C to +60° C
- Relative humidity: 5% to 90%, non-condensing

### Operating environment

#### Temperature and humidity

The Model 4200-SCS should be operated within the following environmental limits:

- Temperature: +15° C to +40° C
- Relative humidity: 5% to 80%, non-condensing

---

**NOTE** *SMU and preamp accuracy specifications are based on operation at 23° C  $\pm$ 5° C and between 5% and 60% relative humidity. See the product specifications for additional temperature and humidity derating factors outside these ranges.*

---

#### Proper ventilation

To avoid overheating, the Model 4200-SCS should be operated in an area with proper ventilation. Allow at least eight inches of clearance at the back of the mainframe to assure sufficient airflow.



- 
- 
- CAUTION** To prevent damaging temperatures and other harmful environmental conditions that could degrade specified performance, follow these precautions:
- Keep the venting holes and fan free of dust, dirt, and contaminants, so that the unit's ability to dissipate heat is not impaired.
  - Keep the fan vents and cooling vents from becoming blocked.
  - Do not position any devices that force air (heated or unheated) adjacent to the unit into cooling vents. This additional airflow could compromise accuracy performance.
  - When rack mounting the unit, make sure there is adequate airflow around the sides, bottom, and back to ensure proper cooling.
  - Rack mounting high-power dissipation equipment adjacent to the Model 4200-SCS could cause excessive heating to occur.
  - To ensure proper cooling in rack situations with convection cooling only, place the hottest equipment (the, power supply) at the top of the rack. Precision equipment, such as the Model 4200-SCS, should be placed as low as possible in the rack where temperatures are the coolest.
  - Add spacer panels below the unit will help ensure adequate airflow.
- 
- 

---

---

**CAUTION** A large system (for example, multiple SMUs, multiple pulse generators, and a scope) draws more power than a small system, causing the internal power supply to generate more heat. Because of this, it is imperative that systems of any size have proper ventilation. Even a small system with inadequate ventilation can be damaged by excess heat.

---

---

### Cleanliness

To avoid internal dirt buildup that could degrade performance and affect longevity, the Model 4200-SCS should be operated in a clean, dust-free environment.

## Powering up the Model 4200-SCS

The following information covers power requirements for the Model 4200-SCS power connections, power-up characteristics, and warm-up requirements.

### Line power

The Model 4200-SCS operates from a line voltage in the range of 100 V to 240 V at a frequency of 50 Hz or 60 Hz. Line voltage is automatically sensed, but line frequency is not (For more information see the [Reference Manual, Line frequency setting, page 2-17](#). Check to ensure the operating voltage in your area is compatible.

---

---

**CAUTION** Operating the instrument on an incorrect line voltage may cause damage, possibly voiding the warranty.

---

---



---

**NOTE** To avoid possible problems caused by electrical transients or line voltage fluctuations, the Model 4200-SCS should be operated from a dedicated power source.

---

## Line power connection

Perform the following steps to connect the unit to line power and turn it on:

1. Before plugging in the power cord, make sure the front panel power switch is in the off position.
2. Connect the female end of the supplied power cord to the AC receptacle on the rear panel (see [Figure 1-1](#)).

---

**WARNING** The large diameter line cord (supplied) must be used to power the Model 4200-SCS. DO NOT use a different line cord. Using a different line cord may result in personal injury or death due to electric shock.

---

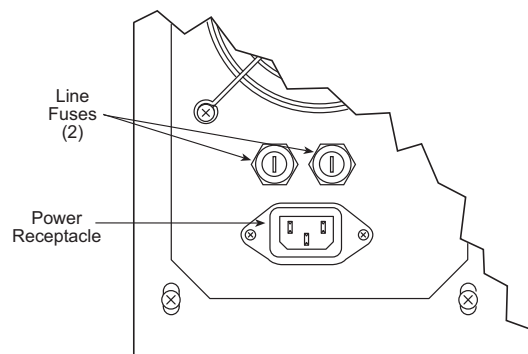
3. Connect the other end of the supplied line cord to a grounded AC line power receptacle.

---

**WARNING** The power cord supplied with the unit contains a separate ground for use with grounded outlets. When proper connections are made, the instrument chassis is connected to power line ground through the ground wire in the power cord. Failure to use a grounded outlet may result in personal injury or death due to electric shock.

---

Figure 1-1  
Line power receptacle and line fuses location



## Line frequency setting

The Model 4200-SCS can be operated either from 50 Hz or 60 Hz power line sources, but it does not automatically sense the power line frequency when it is powered up. You can change the line frequency setting using the KCON utility. See the [Reference Manual, Keithley CONFIGuration Utility \(KCON\)](#), page 7-1 for details.

---

**NOTE** Operating the Model 4200-SCS with the wrong line frequency setting may result in noisy readings because the line frequency setting affects SMU line frequency noise rejection.

---



## Line fuses

Rear-panel fuses protect the power line input of the unit.

If the line fuses need to be replaced, perform the following steps:

---

---

**WARNING**    *Turn off the power and disconnect the line cord before replacing the fuses. Failure to turn off the power and disconnect the line cord before replacing the fuses may result in personal injury or death due to electric shock.*

---

---

1. The fuses are located in two fuse holders above the AC receptacle (see [Figure 1-1](#)).
2. Using a small slotted screwdriver to remove each fuse holder, push the fuses in and rotate them counterclockwise to remove.
3. Remove the fuses from the fuse holders and replace them with 250 V, 15 A, 5 × 20 mm, slow-blow fuses.

---

---

**CAUTION**    **For continued protection against fire or instrument damage, replace the fuses only with the type and rating shown above. If the instrument repeatedly blows fuses, locate and correct the cause of the problem before replacing the fuses.**

---

---

## Warm-up period

The Model 4200-SCS can be used immediately after being turned on. However, the unit should be allowed to warm up for at least 30 minutes to achieve rated measurement accuracy.

## System connections

### Connecting the keyboard and mouse

The keyboard is connected to the Model 4200-SCS with a USB, and can be plugged into any of the four USB ports (two in front and two in back; see [Figure 1-3](#)). The keyboard is shown in [Figure 1-2](#). To ensure proper operation, make sure the keyboard is connected to one of the four USB ports prior to power up. [Figure 1-3](#) shows the keyboard connections to the rear panel of the Model 4200-SCS.

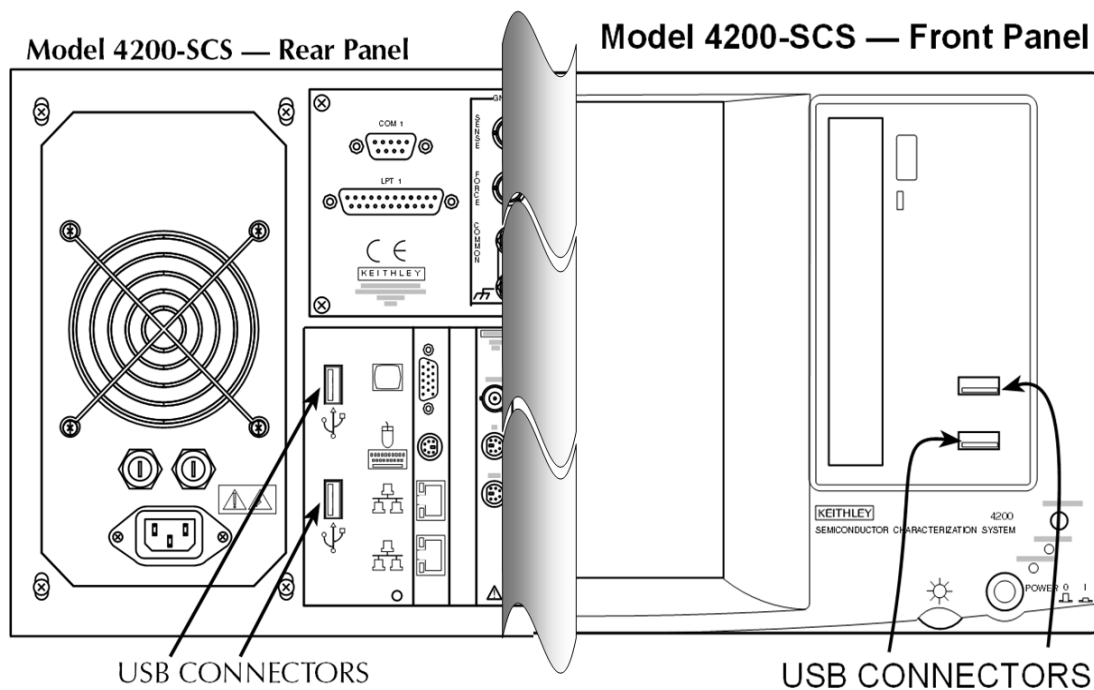


Figure 1-2  
**Model 4200-SCS keyboard**



If you wish to use an optional mouse, connect a USB mouse into any of the four Model 4200-SCS USB ports.

Figure 1-3  
**Keyboard connections**

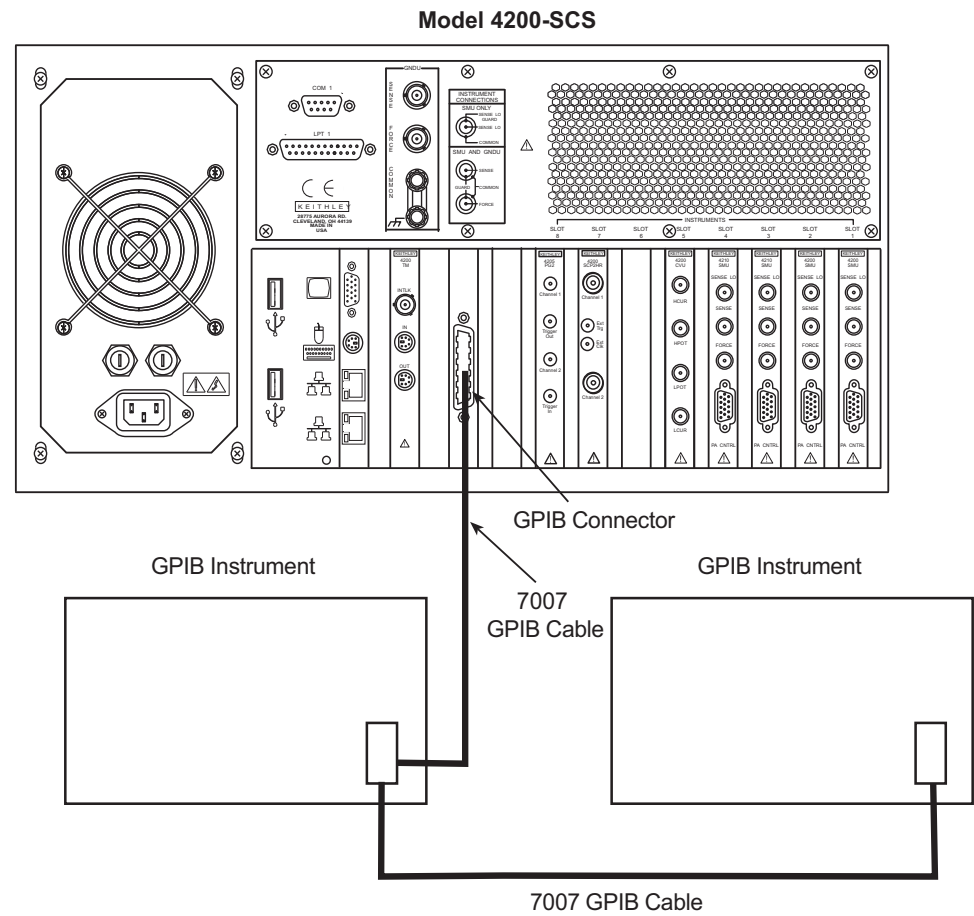


### Connecting GPIB instruments

The Model 4200-SCS can control one or more external instruments by way of the IEEE-488 General Purpose Instrument Bus (GPIB). Instruments typically used in a test system with the Model 4200-SCS include a switch matrix and a C-V meter. [Figure 1-4](#) shows how to connect GPIB instruments to the Model 4200-SCS.



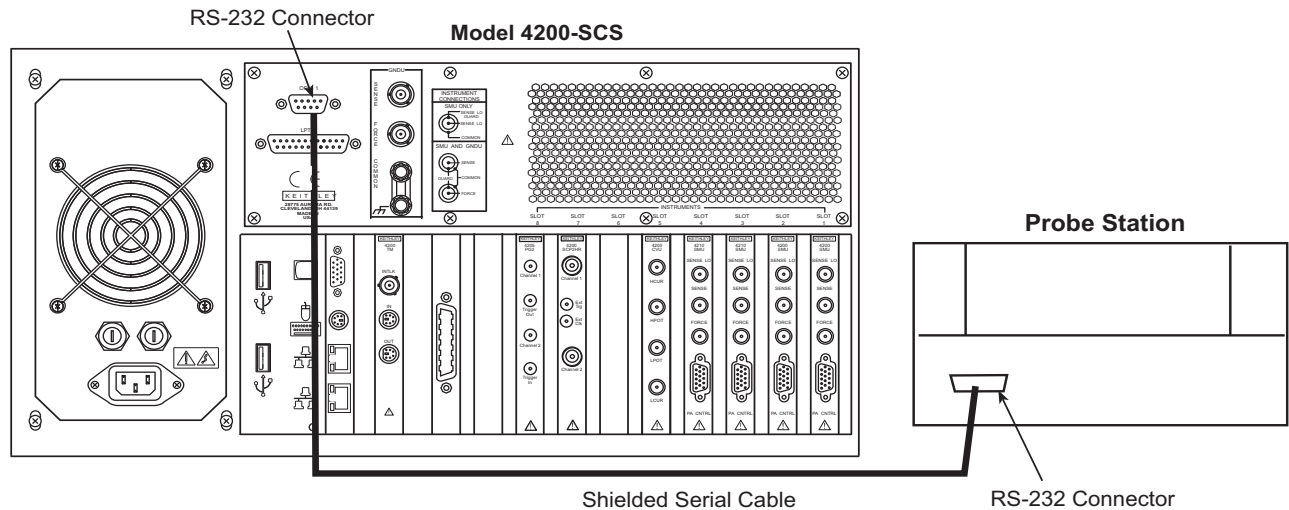
Figure 1-4  
**GPIB instrument connections**



**Connecting a probe station**

A probe station can be controlled through the RS-232 interface connected to the Model 4200-SCS, as shown in [Figure 1-5](#).

Figure 1-5  
**Probe station connections**

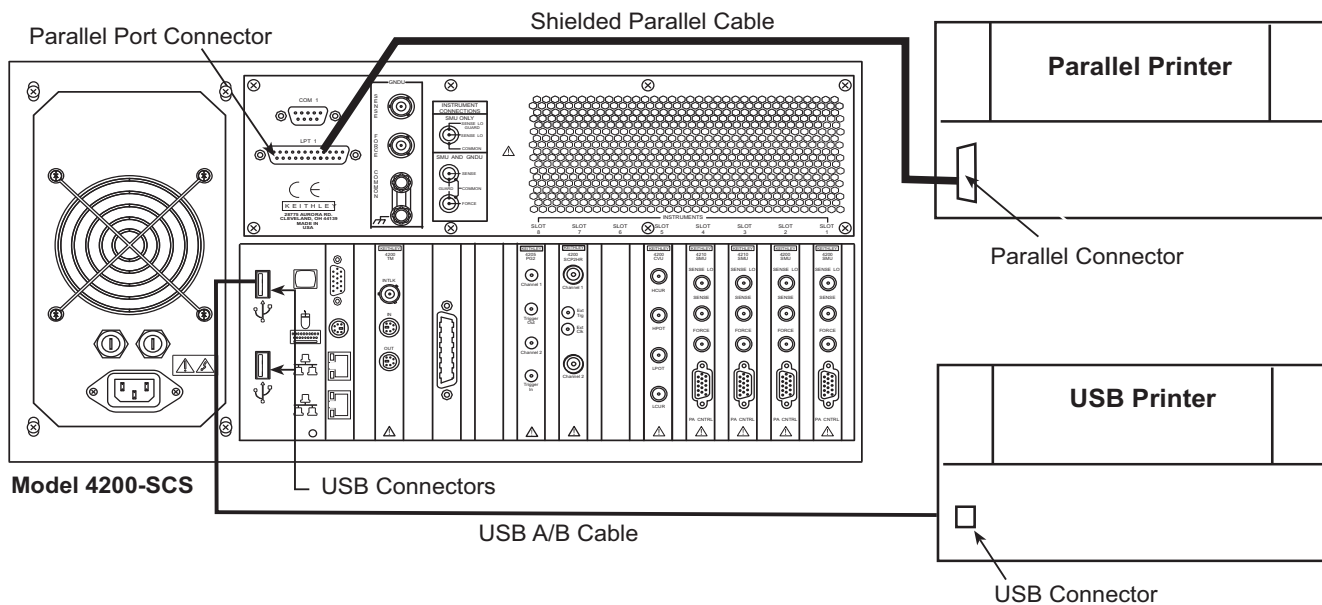




## Connecting a printer

As shown in [Figure 1-6](#), a printer can be connected to the parallel port of the Model 4200-SCS. If you are using a USB printer, connect it to one of the v2.0 USB connectors.

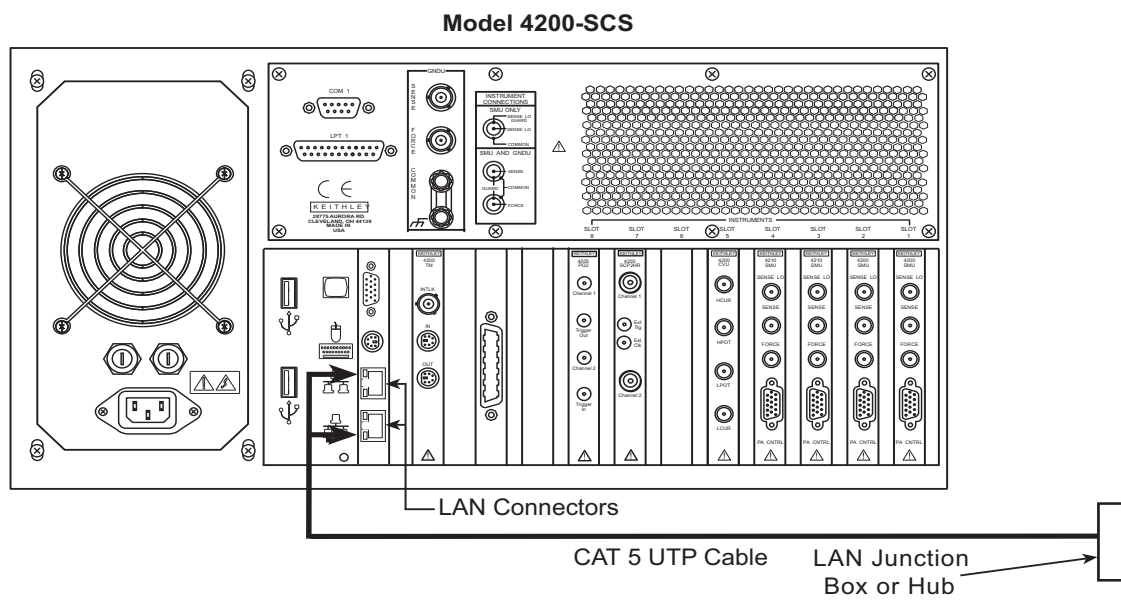
Figure 1-6  
Printer connections



## Connecting a LAN

The two LAN connectors on the Model 4200-SCS are standard RJ-45 connectors intended for use with unshielded twisted pair (UTP) cable. For best results, use only CAT 5 UTP cables equipped with RJ-45 connectors to connect your LANs, as shown in [Figure 1-7](#). If IP addresses are statically assigned, a different IP address will be needed for each of the two LAN ports.

Figure 1-7  
LAN connections





# Model 4200-SCS Hardware Overview

## Front panel

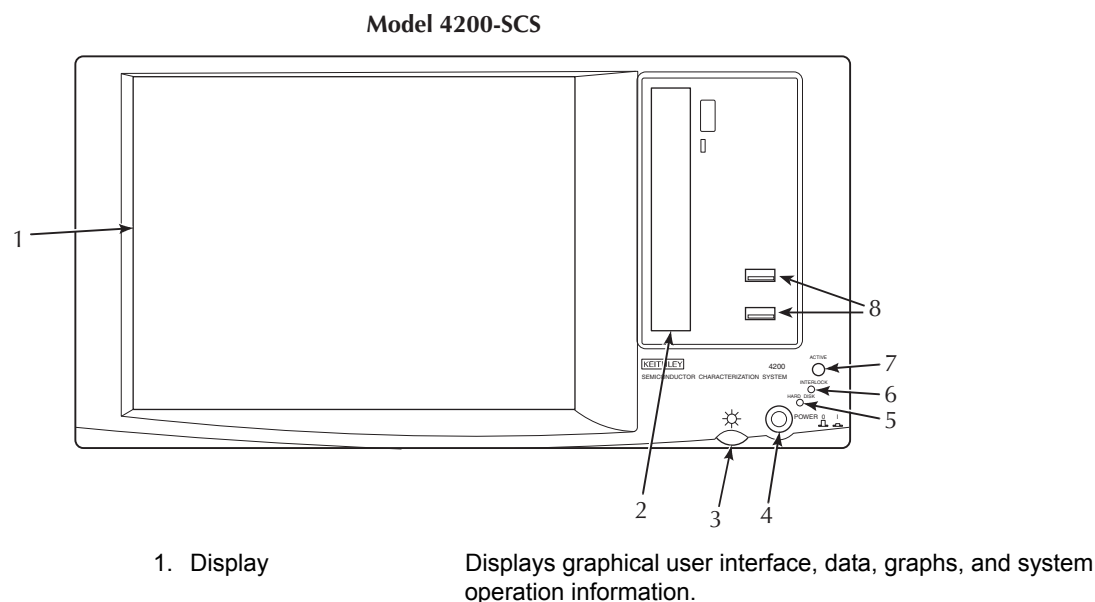
### Instrument panels

All operator interfaces are on the front panel of the Model 4200-SCS, and all connection interfaces are on the rear panel. The next two topics describe the front and rear panels.

### Front panel

Figure 1-8 shows the front panel of the Model 4200-SCS. The various components are summarized below the figure.

Figure 1-8  
Front panel




---

**NOTE** Model 4200-SCS / C has no display and requires an external CRT / monitor.

---

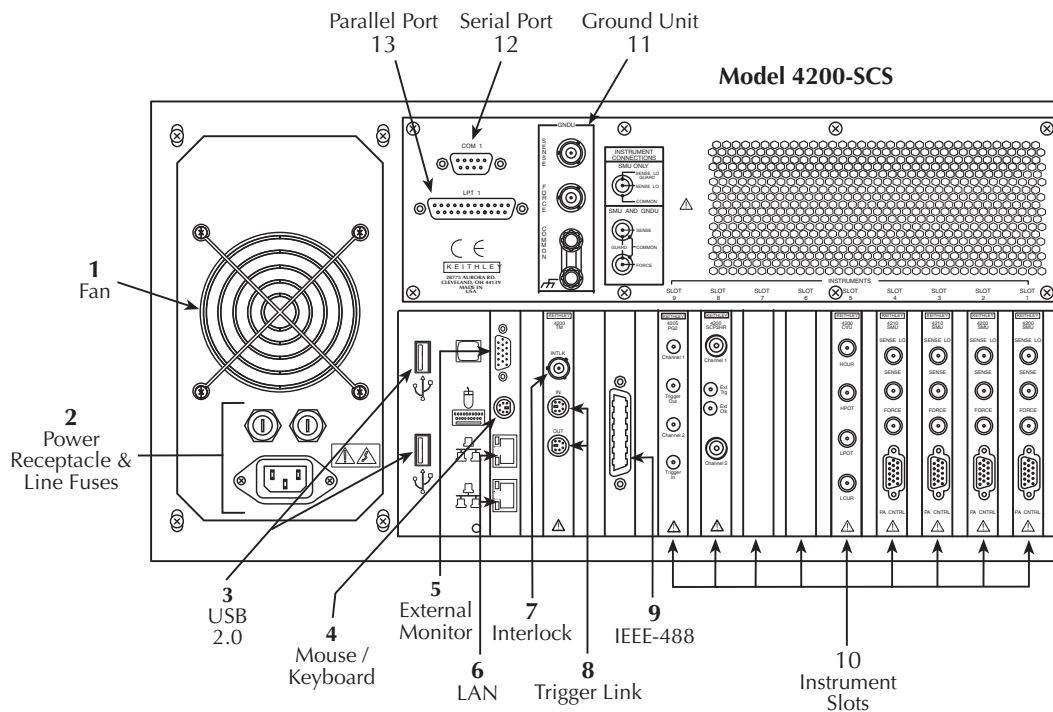
- |                        |   |
|------------------------|---|
| 2. DVD / CD-RW drive   | Provides a means to install or update system software, manuals, and utilities.  |
| 3. Display brightness  | Allows you to set the FPD display to the desired brightness, and turn off the FPD backlight.                                    |
| 4. POWER switch        | Turns main system power on or off.  |
| 5. HARD DISK indicator | Illuminates when the hard disk is being accessed.   |
| 6. INTERLOCK indicator | Illuminates when the test fixture interlock is closed.  |
| 7. ACTIVE indicator    | Illuminates when any internal cards are energized.  |
| 8. Two v2.0 USB ports  | Interfaces to peripherals (for example, pointing devices, printers, scanners, flash drives, external hard drives, and CD-ROMs). |

## Rear Panel

Figure 1-9 shows the rear panel of the Model 4200-SCS mainframe. The various components are summarized below the figure.



Figure 1-9  
Rear panel



- |                                    |  |
|------------------------------------|--|
| 1. Fan                             | Provides system cooling.   |
| 2. Power receptacle and line fuses | Connects to line power through supplied line cord. Two line fuses protect the unit.  |
| 3. Two v2.0 USB connectors         | Interfaces to peripherals (for example, pointing devices, printers, scanners, flash drives, external hard drives, and CD-ROM drives).            |
| 4. Mouse and keyboard connector    | Included Y-cable to connect the mouse or other pointing device and the system keyboard. (see the <a href="#">Reference Manual, Figure 2-1</a> ). |
| 5. External monitor port           | Used to connect an external CRT or other monitor.  |
| 6. Two LAN connectors              | Interfaces the unit to an Ethernet local area network.   |
| 7. Interlock connector             | Connects to test fixture or probe safety interlock.  |
| 8. Trigger Link connectors         | <b>For future use only. Do not use.</b>  |

---

**WARNING** *Using the Trigger Link connectors can cause malfunction or damage to the Model 4200-SCS. These connectors are for future use only and should not be used.*

---

- |                       |   |
|-----------------------|---|
| 9. IEEE-488 connector | Connects to peripherals or computer with GPIB interface.  |
| 10. Instrument slots  | Any of the nine slots can be used for a SMU. Pulse generator cards are installed starting in Slot 9 and continuing to the right. A scope card can be installed in the slot next to the last pulse generator card. A CVU card will be located just after the last SMU. In <a href="#">Figure 1-9</a> , a pulse generator card is installed in slot 9 and a scope card is installed in slot 8. SMUs are installed in slots 1 through 4. |
| 11. Ground unit       | Provides a convenient way to make system-level COMMON and SENSE circuit connections.  |
| 12. Serial port       | Connects to RS-232 peripherals, such as a probe.  |
| 13. Parallel port     | Used to interface to printer or other parallel device.  |



---

**NOTE** The actual rear panel layout may vary slightly from the diagram shown in [Figure 1-9](#).

---

## Activate LAN2 connection

The LAN2 connection may be deactivated by default.

### To enable the LAN2 connection:

- Go to Windows Device Manager.
  - To change network adapter settings in the Device Manager:
1. Double-click **Network adapters**.
  2. Right-click the network adapter for which you want to change settings, and then click **Properties**.
  3. On the **Advanced** tab, make the desired changes.
  4. Device properties for network adapters are device dependent. Determine which settings you need to change to enable device.

## DC source-measure unit (SMU)

### Introduction

This section provides detailed information about several Model 4200-SCS hardware components, and is arranged as follows:

- **Models 4200-SMU and 4210-SMU overview:** Discusses Models 4200-SMU and 4210-SMU basic source and measure characteristics, basic circuit configurations, operating boundaries, and connectors.
- **SMU with Model 4200-PA overview:** Describes how the Model 4200-PA extends Models 4200-SMU and 4210-SMU dynamic range, and covers source and measure characteristics, basic circuit configurations, operating boundaries, connectors, and mounting methods.
- **Ground unit (GNDU):** Provides basic information about using the ground unit, including basic characteristics and connectors.

### Models 4200-SMU and 4210-SMU overview

The following paragraphs discuss the basic characteristics of both the Models 4200-SMU and 4210-SMU.

### Basic characteristics

#### Current characteristics

Current characteristics for both SMUs are summarized in [Table 1-1](#).

Table 1-1  
**Models 4200-SMU and 4210-SMU current characteristics**

Function	4200-SMU	4210-SMU
Current source ranges (full-scale / set resolution)	105 nA / 5 pA	105 nA / 5 pA
	1.05 $\mu$ A / 50 pA	1.05 $\mu$ A / 50 pA
	10.5 $\mu$ A / 500 pA	10.5 $\mu$ A / 500 pA
	105 $\mu$ A / 5 nA	105 $\mu$ A / 5 nA
	1.05 mA / 50 nA	1.05 mA / 50 nA
	10.5 mA / 500 nA	10.5 mA / 500 nA
	105 mA / 5 $\mu$ A	105 mA / 5 $\mu$ A
	-	1.05 A / 50 $\mu$ A



Table 1-1  
**Models 4200-SMU and 4210-SMU current characteristics**

Function	4200-SMU	4210-SMU
Current measurement ranges (full-scale / nominal resolution)	105 nA / 1 pA 1.05 $\mu$ A / 10 pA 10.5 $\mu$ A / 100 pA 105 $\mu$ A / 1 nA 1.05 mA / 10 nA 10.5 mA / 100 nA 105 mA / 1 $\mu$ A -	105 nA / 1 pA 1.05 $\mu$ A / 10 pA 10.5 $\mu$ A / 100 pA 105 $\mu$ A / 1 nA 1.05 mA / 10 nA 10.5 mA / 100 nA 105 mA / 1 $\mu$ A 1.05 A / 10 $\mu$ A

### Voltage characteristics

Table 1-2 summarizes SMU voltage characteristics.

Table 1-2  
**Models 4200-SMU and 4210-SMU voltage characteristics**

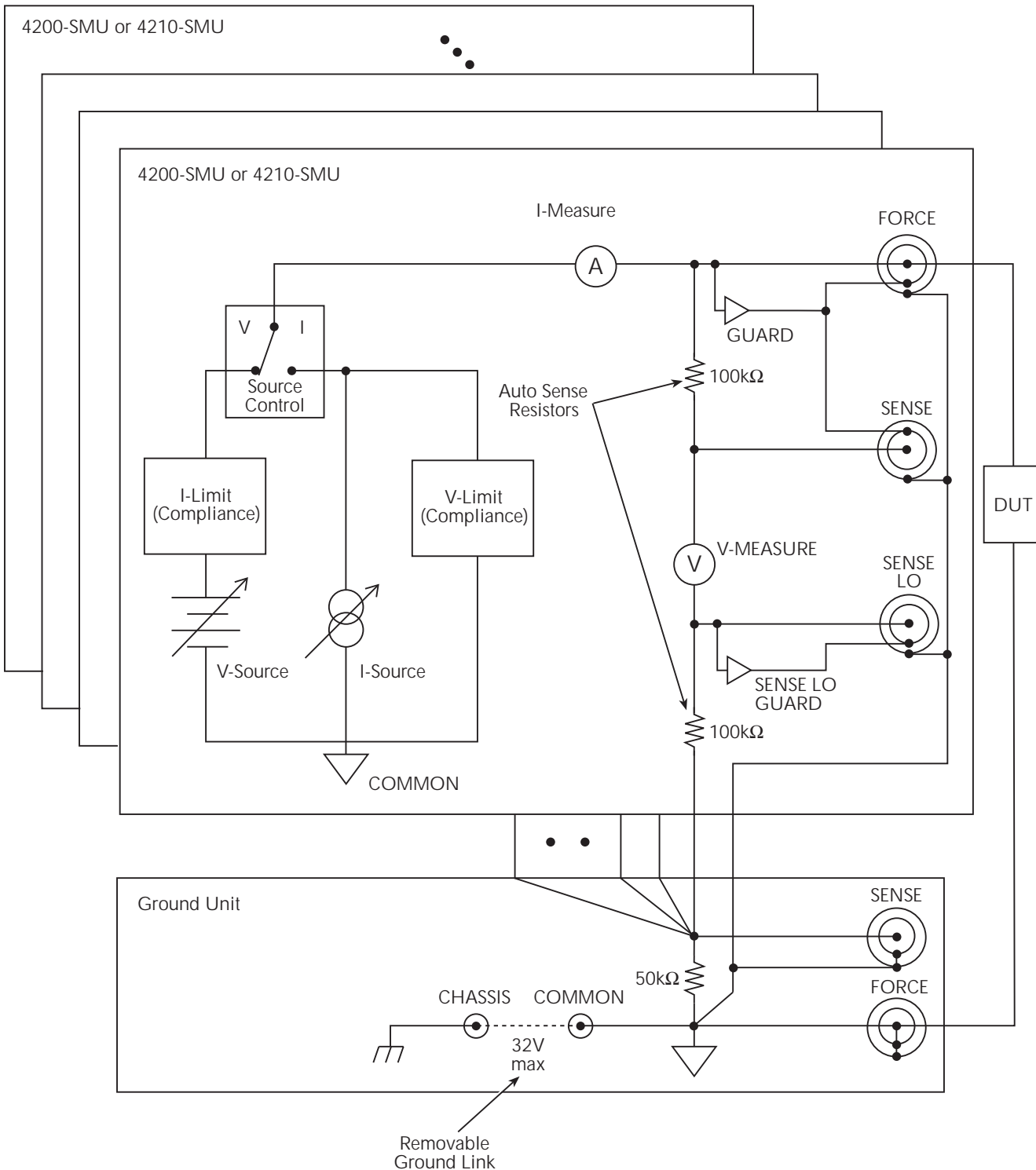
Function	4200-SMU	4210-SMU
Voltage source ranges (full-scale / set resolution)	210 mV / 5 $\mu$ V 2.1 V / 50 $\mu$ V 21 V / 500 $\mu$ V 210 V / 5 mV	210 mV / 5 $\mu$ V 2.1 V / 50 $\mu$ V 21 V / 500 $\mu$ V 210 V / 5 mV
Voltage measurement ranges (full-scale / nominal resolution)	210 mV / 1 $\mu$ V 2.1 V / 10 $\mu$ V 21 V / 100 $\mu$ V 210 V / 1 mV	210 mV / 1 $\mu$ V 2.1 V / 10 $\mu$ V 21 V / 100 $\mu$ V 210 V / 1 mV

### Basic SMU circuit configuration

The basic SMU circuit configuration is shown in [Figure 1-10](#). The SMU operates as a voltage or current source (depending on source function) in series with an I-Meter, and connected in parallel with a V-Meter. The voltage limit (V-limit) and current limit (I-limit) circuits limit the voltage or current to the programmed compliance value. In this local sensing example, the SMU FORCE terminal is connected to device-under test (DUT) HI, while the DUT LO is connected to COMMON. See the [Reference Manual, Connections and Configuration, page 4-1](#), and [Reference Manual, Source-Measure Concepts, page 5-1](#) for details.



Figure 1-10  
**Basic SMU source-measure configuration**





## SMU terminals and connectors

The locations and configuration of the Models 4200-SMU and 4210-SMU terminals are shown in [Figure 1-11](#). Basic information about these terminals is summarized below. Refer to the [Reference Manual, Connections and Configuration, page 4-1](#) for additional information regarding SMU signal connections.

---

**WARNING**    *Asserting the interlock will allow the SMU and preamp terminals to become hazardous, exposing the user to possible electrical shock that could result in personal injury or death. SMU and preamp terminals should be considered hazardous even if the outputs are programmed to be low voltage. Precautions must be taken to prevent a shock hazard by surrounding the test device and any unprotected leads (wiring) with double insulation for 250 V, Category I.*

---

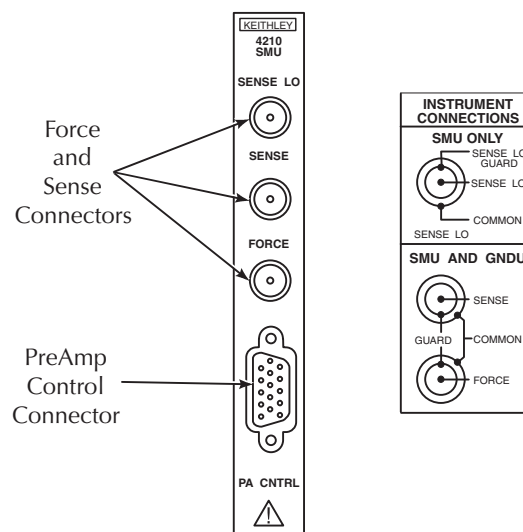


---

**CAUTION**    The maximum allowed voltage between COMMON and chassis ground is  $\pm 32$  V DC.

---

Figure 1-11  
Models 4200-SMU and 4210-SMU connectors



### FORCE terminal

The FORCE terminal is a miniature triaxial connector used to apply the SMU FORCE signal to the DUT when a preamp is not being used.

- The center pin is FORCE
- The inner shield is GUARD
- The outer shield is circuit COMMON

### SENSE terminal

The SENSE terminal is a miniature triaxial connector used to apply the SMU SENSE signal to the DUT in a remote sense application when the preamp is not being used.

- The center pin is SENSE



- The inner shield is GUARD
- The outer shield is circuit COMMON

Nominal internal auto-sense resistance appears between SENSE and FORCE.

---

**NOTE** *The SENSE terminal does not need to be connected to the DUT for the SMU to operate correctly. Remote sensing is automatic. If SENSE is connected to the DUT, errors due to voltage drops in the FORCE path between the SMU and the DUT will be eliminated; and, the SMU will sense locally.*

---

### SENSE LO terminal

The SENSE LO terminal is a miniature triaxial connector used to apply the SMU SENSE LO signal to the DUT in a full-Kelvin remote sense application.

- The center pin is SENSE LO
- The inner shield is SENSE GUARD
- The outer shield is circuit COMMON

Nominal internal auto-sense resistance appears between SENSE LO GUARD and COMMON.

---

**NOTE** *The remote sense capability of the ground unit should be used instead of the SENSE LO of a SMU. If it is necessary to use the SENSE LO terminal of an SMU, the SENSE LO terminals of all SMUs being used in that Model 4200-SCS should be connected to the DUT.*

---

### PA CNTRL connector

The PA CNTRL (preamp control) terminal is a 15-pin D connector that provides both power and signal connections to the Model 4200-PA remote preamp. The preamp can either be mounted and connected directly to the SMU, or it can be connected to the SMU using a cable (Model 4200-RPC-X) when mounted remotely. Refer to the Model SMU with Model 4200-PA overview for more information about the preamp.

## SMU with Model 4200-PA overview

### Basic characteristics

#### Current characteristics

Current characteristics of the Models 4200-SMU and 4210-SMU, when used with the Model 4200-PA, are summarized in [Table 1-3](#). The preamp extends the current source-measure dynamic range of the Models 4200-SMU and 4210-SMU downward by five decades. The lowest current range available without the preamp is 100nA full-scale, while the lowest range with the preamp is 1pA full-scale.



Table 1-3  
**SMU with Model 4200-PA current characteristics**

Function	4200-SMU with 4200-PA	4210-SMU with 4200-PA
Current source ranges (full-scale / set resolution)	1.05 pA / 50 aA 10.5 pA / 500 aA 100.5 pA / 5 fA 1.05 nA / 50 fA 10.5 nA / 500 fA 105 nA / 5 pA 1.05 $\mu$ A / 50 pA 10.5 $\mu$ A / 500 pA 105 $\mu$ A / 5 nA 1.05 mA / 50 nA 10.5 mA / 500 nA 105 mA / 5 $\mu$ A -	1.05 pA / 50 aA 10.5 pA / 500 aA 100.5 pA / 5 fA 1.05 nA / 50 fA 10.5 nA / 500 fA 105 nA / 5 pA 1.05 $\mu$ A / 50 pA 10.5 $\mu$ A / 500 pA 105 $\mu$ A / 5 nA 1.05 mA / 50 nA 10.5 mA / 500 nA 105 mA / 5 $\mu$ A 1.05 A / 50 $\mu$ A
Current measurement ranges (full-scale / nominal resolution)	1.05 pA / 10 aA 10.5 pA / 100 aA 100.5 pA / 1 fA 1.05 nA / 10 fA 10.5 nA / 100 fA 105 nA / 1 pA 1.05 $\mu$ A / 10 pA 10.5 $\mu$ A / 100 pA 105 $\mu$ A / 1 nA 1.05 mA / 10 nA 10.5 mA / 100 nA 105 mA / 1 $\mu$ A -	1.05 pA / 10 aA 10.5 pA / 100 aA 100.5 pA / 1 fA 1.05 nA / 10 fA 10.5 nA / 100 fA 105 nA / 1 pA 1.05 $\mu$ A / 10 pA 10.5 $\mu$ A / 100 pA 105 $\mu$ A / 1 nA 1.05 mA / 10 nA 10.5 mA / 100 nA 105 mA / 1 $\mu$ A 1.05 A / 10 $\mu$ A

### Voltage characteristics

Table 1-4 summarizes a SMU with Model 4200-PA voltage characteristics that are identical to those for the SMUs alone.

Table 1-4  
**SMU with Model 4200-PA voltage characteristics**

Function	4200-SMU with 4200-PA	4210-SMU with 4200-PA
Voltage source range (full-scale / set resolution)	210 mV / 5 $\mu$ V 2.1 V / 50 $\mu$ V 21 V / 500 $\mu$ V 210 V / 5 mV	210 mV / 5 $\mu$ V 2.1 V / 50 $\mu$ V 21 V / 500 $\mu$ V 210 V / 5 mV
Voltage measurement range (full-scale / nominal resolution)	210 mV / 1 $\mu$ V 2.1 V / 10 $\mu$ V 21 V / 100 $\mu$ V 210 V / 1 mV	210 mV / 1 $\mu$ V 2.1 V / 10 $\mu$ V 21 V / 100 $\mu$ V 210 V / 1 mV

### Basic SMU/preamp circuit configuration

Basic SMU/preamp circuit configuration is shown in [Figure 1-12](#). This configuration is similar to the SMU configuration discussed earlier, exception the preamp, which adds low-current source-measure capabilities.



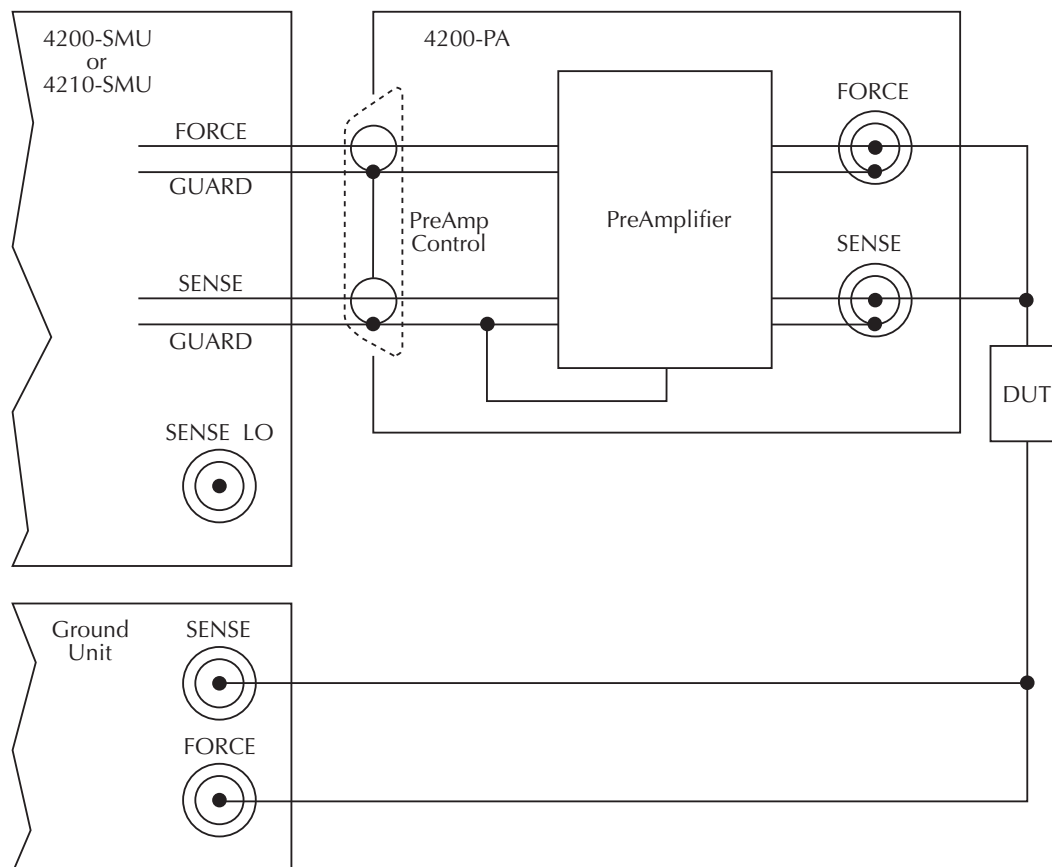
---

**NOTE** The preamp FORCE terminal is connected to DUT HI, while DUT LO is connected to COMMON.

---

See the [Reference Manual, Basic source-measure connections, page 4-3](#), and [Source-Measure Concepts, Section 5](#) for more source-measure details.

Figure 1-12  
Basic SMU/preamp source-measure configuration



## DC preamp

### PreAmp terminals and connectors

The locations and configuration of the Model 4200-PA terminals are shown in [Figure 1-13](#). Basic information about these terminals is summarized below. For additional information about making preamp signal connections, refer to the [Reference Manual, Basic source-measure connections, page 4-3](#).

---

**WARNING** The preamp terminals can carry exposed hazardous voltages that could result in personal injury or death if the safety interlock is asserted. See the [Reference Manual, Control and data connections, page 4-21](#) for additional information about safety interlock connections.

---



**CAUTION** The maximum allowed voltages between the preamp signals are as follows:

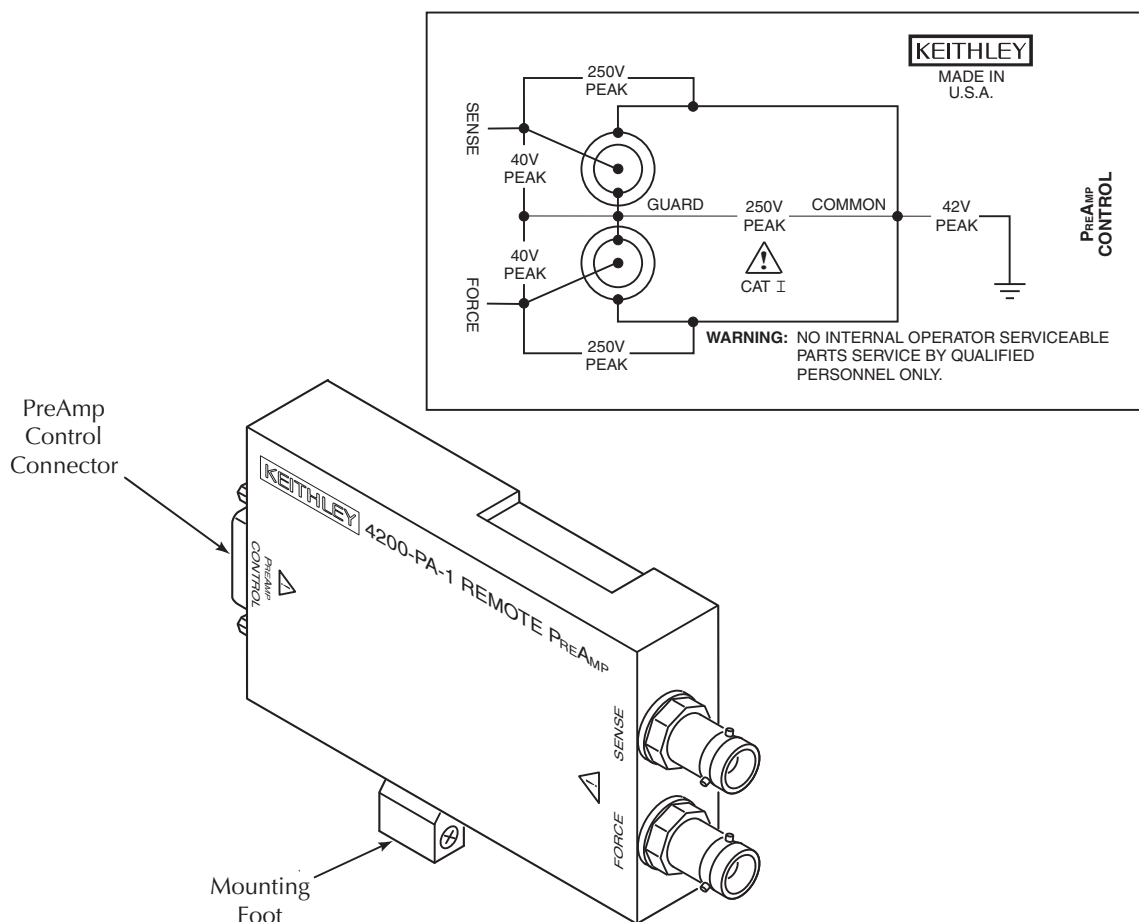
- **COMMON** to chassis ground: 32 V peak
- **GUARD** to **COMMON**: 250 V peak
- **SENSE** or **FORCE** to **GUARD**: 40 V peak

### FORCE terminal

The FORCE terminal is a standard triaxial connector used to apply the preamp FORCE signal to the DUT.

- The center pin is FORCE
- The inner shield is GUARD
- The outer shield is circuit COMMON

Figure 1-13  
Model 4200-PA connectors





## SENSE terminal

The SENSE terminal is a standard triaxial connector used to apply the preamp SENSE signal to the DUT in a remote sense application.

- The center pin is SENSE
- The inner shield is GUARD
- The outer shield is circuit COMMON

Nominal internal auto-sense resistance appears between SENSE and FORCE.

---

**NOTE** *The SENSE terminal does not need to be connected to the DUT for the preamp to operate correctly. Remote sensing is automatic. If SENSE is connected to the DUT, errors due to voltage drops in the FORCE path between the preamp and the DUT will be eliminated; and, the preamp will sense locally.*

---

## PreAmp CONTROL connector

The preamp CONTROL connector connects to the SMU PA CNTRL connector and provides both power and signal connections from the Models 4200-SMU or 4210-SMU to the Model 4200-PA preamp.

# Multi-frequency capacitance / voltage unit (CVU)

## Introduction

The Model 4210-CVU<sup>1</sup> is a multi-frequency (1 kHz to 10 MHz) impedance measurement card that is installed in the Model 4200-SCS mainframe. The AC test signal (10 mV RMS to 100 mV RMS) can be DC voltage biased from -30 V to +30 V.

The CVU measures impedance by sourcing an AC voltage across the device under test (DUT), and then measures the resulting AC current and phase difference. The capacitance and conductance are derived parameters from the measured impedance and phase.

## Model 4210-CVU card

### Measurement overview

AC impedance measurement ( $Z_{DUT}$ ) of the device under test (DUT) is performed by sourcing an AC test voltage across the device and measuring the resulting AC current.

The AC current is measured as shown in [Figure 1-14](#). The Model 4210-CVU uses an auto balance bridge (ABB) technique to achieve accurate impedance measurements. The purpose of the ABB is to create a virtual ground at the DUT to minimize measurement error. Every CVU measurement is taken with ABB active. The ABB will always attempt to **lock** the low side of the DUT to virtual ground.

#### If the ABB fails to lock:

- The measurement will still be taken but may be out of specification.
- The returned data will be flagged and colored yellow in the data sheet.
- The graph will display an **ABB Not Locked** message.

---

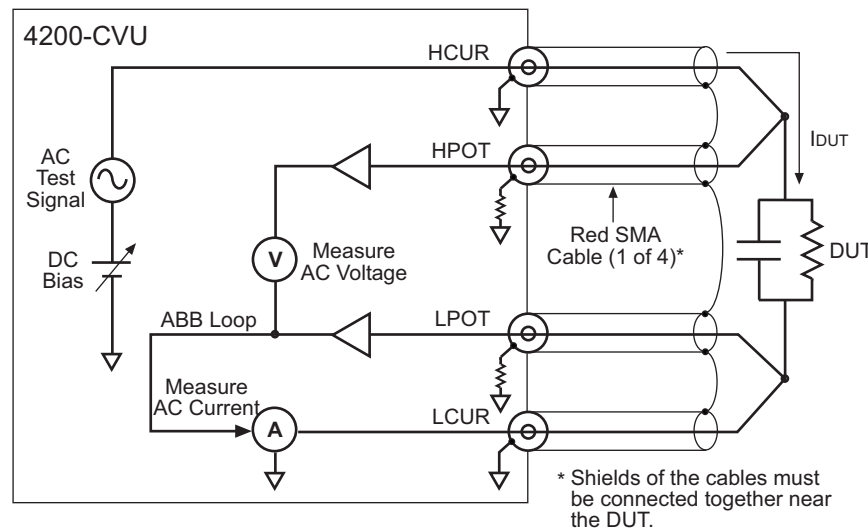
1. In February 2009, the 4210-CVU replaced the 4200-CVU. The 4210-CVU is identical to the 4200-CVU **except** that it adds the 1 kHz frequency. (1 kHz - 9 kHz in 1 kHz increments)



**Most common causes of ABB not locked are as follows:**

- Mismatched physical cable lengths
- Mismatched physical cable lengths versus the programmed cable length in Keithley Interactive Test Environment (KITE)
- Improperly torqued SMA cables
- Sub-optimal I-range setting
- Too much parasitic load on the low side of DUT

Figure 1-14

**Measurement circuit (simplified)**

The capacitive impedance (and conductance) are calculated based on the measured AC impedance and phase.

The capacitance is calculated from the capacitive impedance and the test frequency using the following formula:

$$C_{DUT} = \frac{I_{DUT}}{2\pi f V_{AC}}$$

$C_{DUT}$  = Capacitance of the DUT (f)  
 $f$  = Test frequency (Hz)  
 $V_{AC}$  = Measured AC voltage (V)

**Measurement functions**

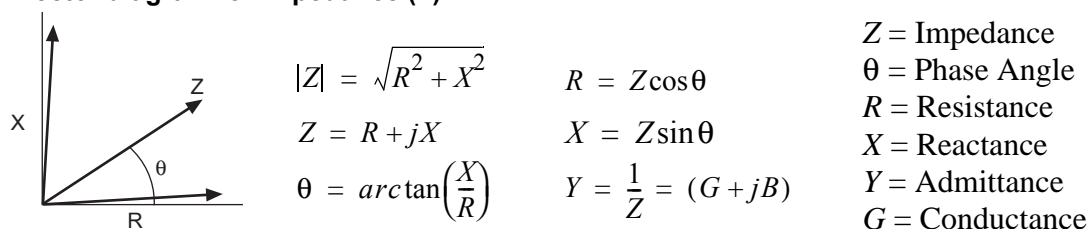
The Model 4210-CVU can measure the following parameters:

- |            |   |
|------------|---|
| • Z, Theta | Impedance and Phase Angle                   |
| • R + jX   | Resistance and Reactance                    |
| • Cp-Gp    | Parallel Capacitance and Conductance        |
| • Cs-Rs    | Series Capacitance and Conductance          |
| • Cp-D     | Parallel Capacitance and Dissipation Factor |
| • Cs-D     | Series Capacitance and Dissipation Factor   |

Figure 1-15 shows the vector diagram and fundamental equations for impedance.

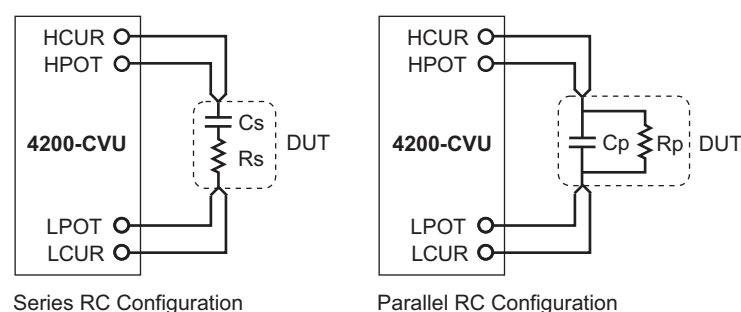


Figure 1-15

**Vector diagram for impedance (Z)**

The simplified model of a DUT is a resistor and a capacitor. As shown in [Figure 1-16](#), the Model 4210-CVU can measure the DUT as a series configuration of the resistor-capacitor (RC), or as a parallel RC configuration.

Figure 1-16

**Measure models (simplified)****Test signal**

The test signal can be set for the following frequencies:

- 1 kHz through 10 kHz in 1 kHz increments
- 10 kHz through 100 kHz in 10 kHz increments
- 100 kHz through 1 MHz in 100 kHz increments
- 1 MHz through 10 MHz in 1 MHz increments

The AC signal output level can be set from 10 mV RMS to 100 mV RMS (1 mV resolution). The output impedance is 100  $\Omega$  (typical).

There are three current measurement ranges available to measure current: 1  $\mu\text{A}$ , 30  $\mu\text{A}$  or 1 mA. With auto range selected, range selection will be performed automatically.

**DC bias function and sweep characteristics**

The AC test signal can be biased with a static DC level (-30 V to +30 V), or a voltage sweep (up or down).

**You can also perform a frequency sweep (up or down):**

- DC bias waveform: The DC bias is set to 0V, but can be set to any valid DC bias level (you specify the number of measurements to perform). (see [Figure 1-17](#))
- DC voltage sweep: You specify the start voltage, stop voltage and step voltage. The number of data (measurement) points is calculated by the Model 4210-CVU. (see [Figure 1-18](#))
- Frequency sweep: You specify the start frequency and the stop frequency; the number of data (measurement) points is calculated by the Model 4210-CVU. (see [Figure 1-19](#))



- Voltage list sweep: You specify the voltage levels for the sweep. (not shown)
- Step frequency sweep: Includes voltage stepping. A voltage sweep will be performed for every frequency point. (not shown)

---

**NOTE** Refer to the [Forcing functions and measure options, page 3-15](#) for details on the bias and sweep forcing functions.

---

Figure 1-17

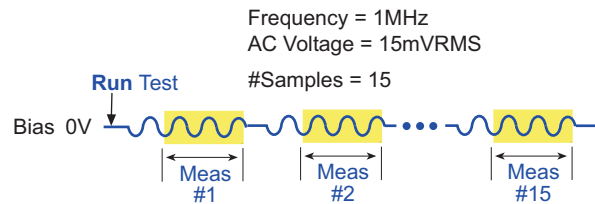
**DC bias waveform (example)**

Figure 1-18

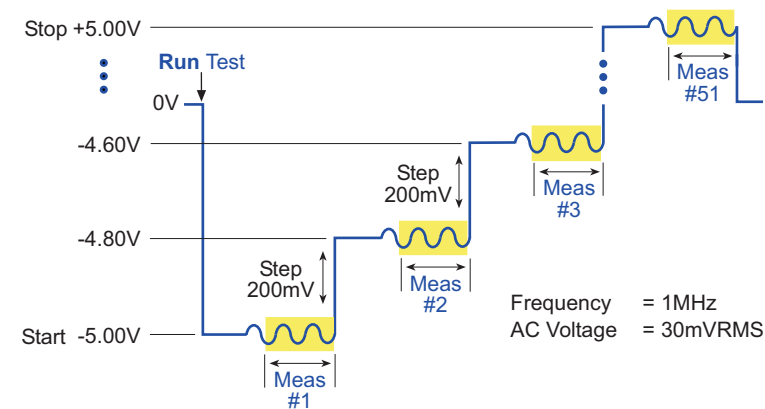
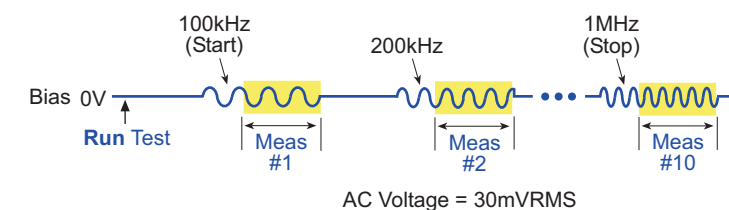
**DC voltage sweep (example)**

Figure 1-19

**Frequency sweep (example)**

## Force-measure timing

### Bias function

Timing for the force-measure process for a bias function is shown in [Figure 1-20](#).

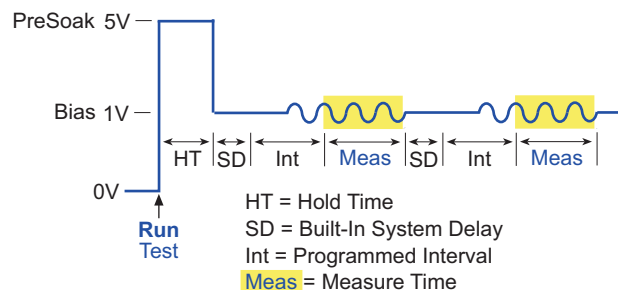
**When the test is started, the following timing sequence takes place:**

1. The DC source outputs the presoak voltage for the hold time period.
2. The DC source goes to the DC bias voltage.



3. After the built-in system delay and time Interval periods, the Model 4210-CVU performs a measurement. The AC test signal is applied just before the start of the measurement. AC drive is turned off after the measurement is completed.
4. Step 3 is repeated for every measurement.

Figure 1-20

**Force-measure timing****Sweep function**

Force-measure timing for a sweep function is similar to the timing for a bias function (shown in Figure 1-20), with the following differences:

- The hold time is repeated at the beginning each subsequent sweep step.
- A programmed delay is used in place of the interval.

**Pulse cards**

The Keithley Instruments pulse cards are two-channel, high speed, voltage pulse generator cards that provide the following types of output:

There are two pulse generator instrument cards available for Model 4200-SCS:

- Model 4220-PGU Pulse Generator Unit
- Model 4225-PMU Ultra-Fast IV Module.

Both cards offer:

- Two output channels
- Standard (2-level) pulse
- Segment Arb<sup>TM</sup> waveform
- Full Arb

Each output channels has two output ranges:

- 10 V (into high impedance, 5 V into 50  $\Omega$ )
- 40 V (into high impedance, 20 V into 50  $\Omega$ )

The Model 4220-PGU is a 2-channel voltage pulse generator. The Model 4225-PMU is also a 2-channel voltage pulse generator, but includes integrated simultaneous current and voltage measurement with two A/D converters for each channel.

Both can be isolated from the DUTs by a high endurance output relay (HEOR). The HEOR is typically used for applications that require high-speed, high-volume switching of the output.

A pulse card can be programmed for continuous pulse output or set to output a finite number of pulses (burst or trig burst triggering modes). The pulse amplitude can be set from 100 mV to 40 V. The pulse period can be set from 20 ns to 1 s with a minimum pulse width of



10 ns. Transition times (pulse rise and pulse fall) can be set independently. Refer to [Pulse card settings](#) for details on all pulse card settings.

**NOTE**

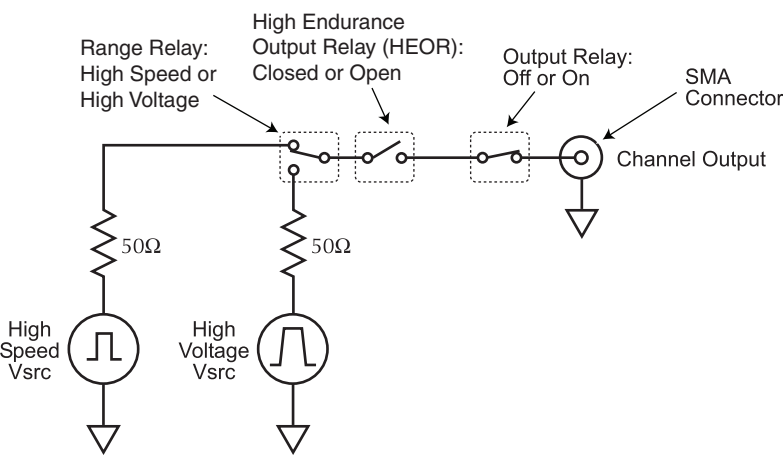
Pulse amplitude can be set as high as 80 V depending on the pulse high and low levels, pulse output range, and DUT load settings.

Refer to the [Reference Manual, Pulse source-measure connections, page 11-34](#) for details on pulse card connectors and connections to the DUT.

[Figure 1-21](#) shows a simplified schematic the Model 4220-PGU pulse card single channel output. The range relay chooses between the high-speed and high-voltage output ranges. The schematic for the Model 4225-PMU is similar except it also includes measure circuitry for both current and voltage (see [Figure 16-2](#) in the Reference Manual).

The HEOR provides fast, unlimited, open/close cycles for demanding tests such as flash memory endurance. The HEOR is also known as the SSR (solid state relay), See [Segment Arb™ waveform](#) for more details about the typical use of the HEOR, which is a solid-state relay for connecting or disconnecting a pulse channel from a device terminal.

Figure 1-21  
Simplified schematic of each Model 4220-PGU channel



## About the pulse cards

The following chart shows a comparison of features between the Model 4220-PGU) and the Model 4225-PMU):

Table 1-5  
Feature comparison of pulse cards

Feature	Model 4220-PGU	Model 4225-PMU
Standard (2-level) Pulse	Yes	Yes
Segment Arb™	Yes	Yes
Full Arb (source only)	Yes	Yes
High Endurance Output Relay (solid-state relay; SSR	Yes	Yes
Integrated Current and Voltage Measure	No	Yes



## Firmware upgrade for the Model 4200-PG2

The firmware can be upgraded to allow the Model 4200-PG2 to configure and output Segment Arb™ and full arb waveforms. However, since the Model 4200-PG2 does not have the HEORs and an input trigger connector, the related operations cannot be performed. These exceptions will be noted where appropriate in this section.

The instructions to upgrade the firmware of the Model 4200-PG2 to KITE V6.2, are available by clicking on the Model 4200-SCS Complete Reference icon on the Model 4200-SCS desktop. Follow the links for release notes, then look for the firmware upgrade procedure for the pulse card firmware.

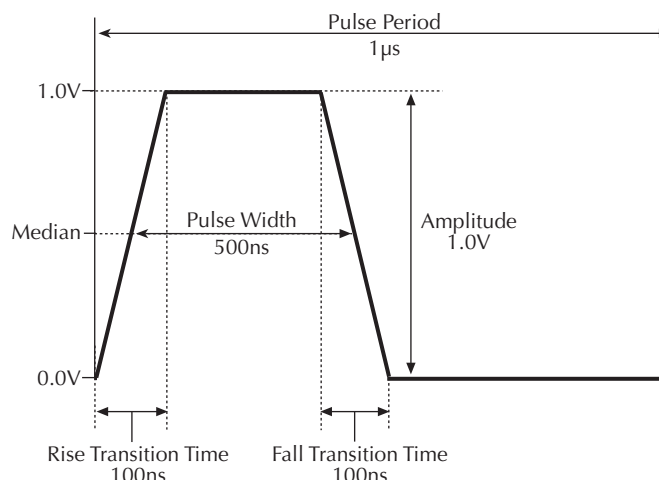
## Standard pulse

Each channel of a pulse card can be configured for standard pulse output. [Figure 1-22](#) shows an example of standard pulse output.

A pulse card is a dual-channel pulse generator. Each channel can output high speed (low voltage) or high voltage (medium speed) pulses. The basic pulse characteristics of the pulse card are listed in the specification sheet.

Figure 1-22

**Standard pulse example (pulse high = 1 V, pulse low = 0 V)**



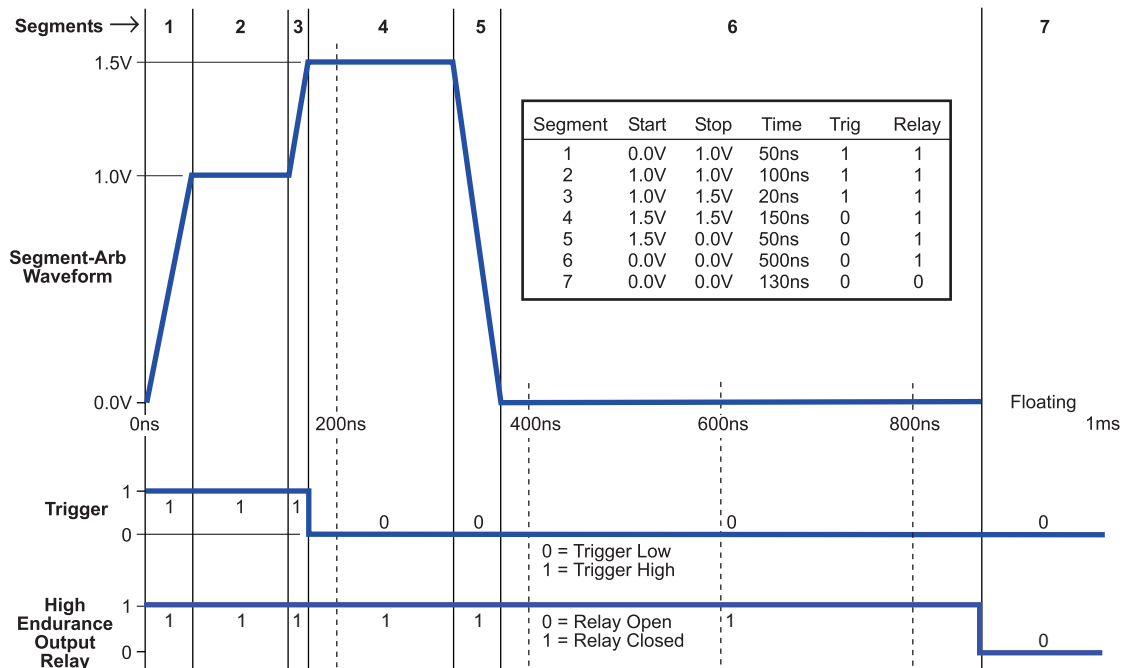
## Segment Arb™ waveform

Each channel of a pulse card can be configured to output its own unique Segment Arb™ waveform. A Segment Arb™ waveform is composed of user-defined line segments (up to 1024 for the Model 4205-PG2 or 2048 for the Models 4220-PGU and 4225-PMU). Each segment can have a unique time interval, start value, stop value, output trigger level (TTL high or low) and output relay state (open or closed).

[Figure 1-23](#) shows an example of a Segment Arb™ waveform that contains seven segments. It also shows the programmed trigger levels and open/closed states for the output relay.



Figure 1-23  
**Segment Arb™ waveform example**



#### Start, stop, and time restrictions:

- The start level of the first segment and the stop level of the last segment must be the same. In Figure 1-23, segment 1 start and segment 7 stop are both set for 0.0 V.
- The stop level for a segment must be the same as the start level for the next segment. In Figure 1-23, the stop level for Segment 1 is 1.0 V, as is the start level for Segment 2 (no discontinuities are allowed).
- The minimum time per segment is 20 ns, with increments of 10 ns.

**Trigger levels:** The segment trigger levels are available at the TRIGGER OUT connector. When set high (1), a TTL high level will be present at TRIGGER OUT during that time interval. When set low (0), the trigger goes low for that segment. In Figure 1-23, trigger is set high for the first three segments, and low for the rest of the segments.

**NOTE** If both channels of a pulse card are being used, the segment trigger levels for CHANNEL 1 will be seen at the TRIGGER OUT connector. The trigger levels for CHANNEL 2 are ignored.

**High-endurance output relay (HEOR):** Each output channel of a pulse card has a high-speed, solid-state output relay. When this relay is closed, the waveform segment is output. When opened, the channel output is electrically isolated (floating) from the DUT. In Figure 1-24, the output relay is opened during segment seven. This puts the output in a floating condition. The minimum time for a segment with a HEOR transition (open-to-close or close-to-open) is 100  $\mu$ s for the Model 4205-PG2 or 25  $\mu$ s for the Models 4220-PGU and 4225-PMU.



---

**NOTE** *If the firmware for the Model 4200-PG2 has been upgraded to KITE V6.2, it can be used to configure and output Segment Arb™ waveforms (see [Firmware upgrade for the Model 4200-PG2](#) earlier in this section). However, the Model 4200-PG2 does not have output relays (HEOR). Therefore, relay control will be ignored.*

---

#### **seg\_arb\_define:**

This function is used to define a Segment Arb™ waveform. This function includes parameters to specify the number of segments (`nSegments`), and arrays for start (`startvals`), stop (`stopvals`), and time values (`timevals`). It also includes arrays for trigger levels (`triggervals`) and output relay states (`outputRelayVals`). For more information, refer to the [Reference Manual, seg\\_arb\\_define, page 8-99](#).

#### **seg\_arb\_file:**

This function is used to load a Segment Arb™ waveform into a pulse card. For more information, refer to the [Reference Manual, seg\\_arb\\_file, page 8-100](#).

#### **seg\_arb\_sequence and seg\_arb\_wavform:**

These more advanced functions can be used by the Models 4220-PGU and 4225-PMU cards to define a Segment Arb waveform (see [seg\\_arb\\_sequence](#) and [seg\\_arb\\_wavform](#) in [Section 8](#) of the Reference manual).

---

**NOTE** *Because of resources necessary to run the Segment Arb™ engine, an additional 10 ns interval is added to the end of the last segment of a Segment Arb™ waveform. During this interval, the output voltage, HEOR, and trigger output values remain the same as the final value reached in the last segment.*

---

### **Full arb**

---

**NOTE** *If the firmware for the Model 4200-PG2 has been upgraded to KITE V6.2, it can be used to configure and output full arb waveforms (see [Firmware upgrade for the Model 4200-PG2](#) earlier in this section).*

---

Each channel of the pulse generator can be configured to generate its own unique full arb waveform. A full arb waveform is made up of user-defined points (up to 262,144).

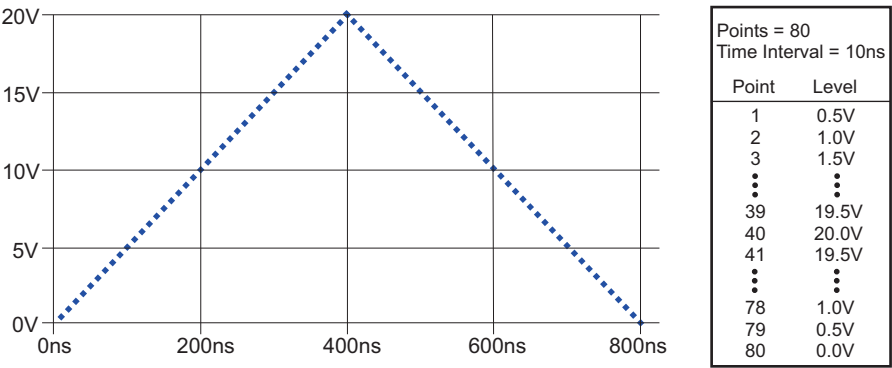
Each waveform point can have its own unique voltage value. A time interval is set to control the time spent at each point in the waveform. [Figure 1-24](#) shows an example of a user-defined full arb waveform. The waveform is made up of 80 voltage points, with the time interval between each point set to 10 ns.

The `arb_array` function is used to define a full arb waveform. This function includes parameters to specify the number of waveform points (`length`), the time interval (`TimePerPt`), an array of voltage levels (`levelArr`), and a file name (`fname`). For more information, refer to the [Reference Manual, arb\\_array, page 8-81](#).

The `arb_file` function is used to load the defined full arb waveform into the pulse generator. For more information, refer to the [Reference Manual, arb\\_file, page 8-82](#).



Figure 1-24  
Full arb waveform example



KPulse full arb waveforms

The Keithley Pulse tool (KPulse) is a virtual front panel software application used to control the optional pulse generator cards. KPulse can be used to create, save and output full arb waveforms, and provides a collection of basic full arb waveform types such as sine, square, triangle, noise, Gaussian, and calculation. After configuring one of the basic waveform types, you can save it as a .kaf file. For more information, refer to the “How to Generate Basic Pulses” in Section 5.

Once a full arb waveform is saved as a .kaf file, it can later be imported back into KPulse. The waveform can also be loaded into the pulse generator card using the arb\_file function. For more information, refer to the Reference Manual, arb\_file, page 8-82.

Pulse card settings

Settings and features for the pulse card are summarized in Section 11 of the reference manual (see Table 11-1 in the reference manual). For more details see the “How to Generate Basic Pulses” in Section 5. In addition to short descriptions and default settings, the table includes the following:

- **LPT function:** The Keithley Instruments Linear Parametric Test Library (LPTLib) function used for each setting or feature. Refer to the Reference Manual, LPT Library Function Reference, page 8-22 for details.
- **Access level:** The access level for each setting. If a setting can be independently set for each pulse generator channel, its access level is “channel.” Otherwise, the access level is “card” to indicate that both channels are affected.
- **Pulse mode:** A checkmark (✓) is used to indicate which pulse mode is associated to the setting or feature. The “n/a” notation indicates that the pulse mode is not applicable to that setting or feature.

Remote bias tee (RBT) and 3-port power divider

The Model 4205-RBT and power divider are used for the Keithley Instrument’s PulseIV-Complete and Demo-PulseIV projects. Two RBT adapters and one 3-port power divider are included with the Models 4200-PIV-A solution bundle. Also included are two Model 4200-MAG-BASE mounts (that attach bias tee adapters to the probe platen magnetically). For more information, refer to the Reference Manual, PulseIV-Complete and Demo-PulseIV projects, page 12-4.

RBT

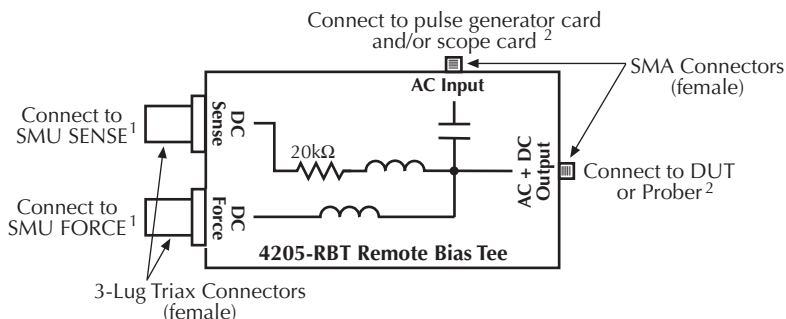
The RBT adapter (see Figure 1-25) is a coupler for DC bias from a SMU, and pulse output from a Model 4205-PG2 pulse generator channel. The output of the RBT provides pulse output riding on the DC V bias.



As shown in [Figure 1-25](#), the RBT has two three-lug female triax connectors for connection to a SMU (FORCE and SENSE), and two female SMA connectors, one for AC inputs, such as a pulse generator card and scope card (Model 4200-SCP2HR or 4200-SCP2), and one for AC+DC output connection to a probe or directly to a DUT.

[Figure 1-25](#) also shows the simplified schematic of the RBT. The capacitor allows pulses from the pulse generator card to pass through to the output, while blocking DC from the SMU. The inductors allow DC from the SMU to pass through to the output, while blocking pulses from the pulse generator.

Figure 1-25

**Model 4205-RBT simplified schematic**

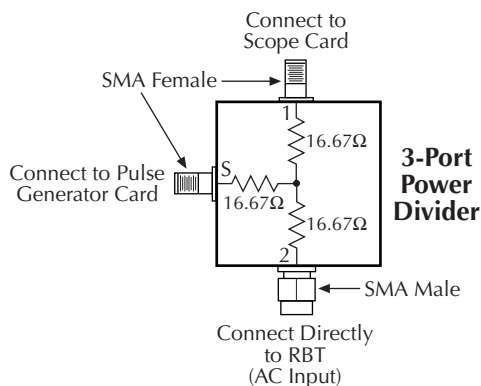
- 1) When using a SMU PreAmp, use 4200-TRX-X cables for connections. When NOT using a PreAmp, use 4200-MTRX-X cables for connections.
- 2) Use SMA cables (male-to-male) for connections.

**3-port power divider**

The 3-port power divider divides the electrical power equally among its three connectors using a 16.67 W resistor in each “leg” (see [Figure 1-26](#)). The power divider is used on the gate of a FET to provide an impedance matched signal (pulse) path (50 Ω).

As shown in [Figure 1-26](#), the power divider is equipped with two SMA female connectors and one SMA male connector. The SMA male connector allows the power divider to connect directly to the RBT (AC input).

Figure 1-26

**3-port power divider****Using an RBT and power divider**

[Figure 1-27](#) shows a block diagram of the pulse IV (PIV) test system that uses two SMUs, a pulse generator card (one channel), a scope card (both channels), two RBTs, and the power divider.

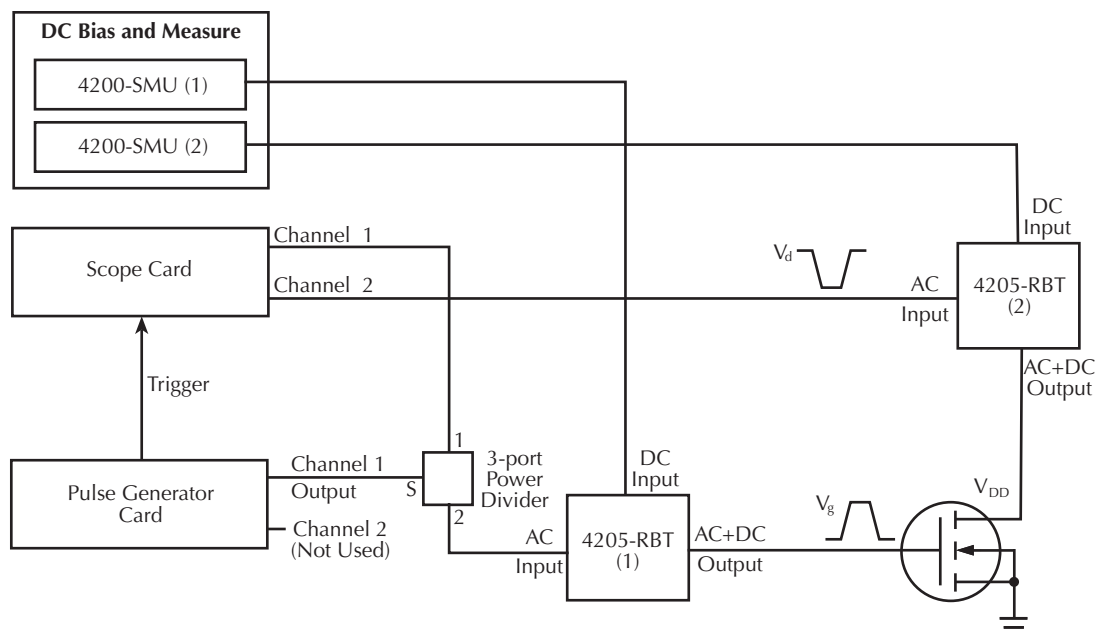


The power divider provides impedance matching; an RBT functions as a coupler for DC bias from a SMU, and pulse output (AC) from a pulse generator card. The output of an RBT provides pulse output that rides on the DC bias level. The scope card is used to capture pulse waveforms or pulse readings. The DUT is typically a wafer site (using prober) or a discrete device.

The capacitor for an RBT functions as a low-impedance component for high-speed pulses, and as a high-impedance element for DC. This allows the high-speed pulses from the pulse generator card to pass through to the output, while blocking DC from the SMU.

The inductors of an RBT function as low-impedance components for DC, and as high-impedance components for high-speed pulses. This allows the DC bias from the SMU to pass through to the output, while blocking the high-speed pulses from the pulse generator.

Figure 1-27  
**Block diagram of a PIV test system**



## SCP2 (Oscilloscope)

### Digital storage oscilloscope card

Keithley Instruments offers two scope cards: Models 4200-SCP2HR and 4200-SCP2. However, only one scope card at a time can be used in the Model 4200-SCS system.

The scope card is a modular, dual-channel, high-speed digital storage oscilloscope (DSO). It uses a high-speed memory digitizer (DC to 700 MHz) and an embedded digital signal processor (DSP). The scope card has two input channels to capture and analyze a variety of time-varying signals.

KScope, a soft front-panel software used to view pulse waveforms, is included with the Models 4200-SCP2HR and 4200-SCP2 (see the [“How to Generate Basic Pulses”](#) in Section 5 for details). KScope provides full control of the DSO and allows export of waveform data in a Microsoft® Excel®-compatible format. The plug-and-play drivers can be used in most application programming environments (for example, National Instruments (NI™) LabVIEW™, Microsoft® Visual Basic®, and Microsoft® Visual C++®).

The primary differences between the two scope cards are ADC resolution and sample rate.



- **Model 4200-SCP2HR:** 16-bit resolution, single-channel sampling rate of 400-million samples per second (400 MS / s)
- **Model 4200-SCP2:** 8-bit resolution, single-channel sampling rate of 2.5 billion samples per second (2.5 GS / s)

Basic pulse characteristics of the two scope cards are listed in [Table 1-6](#). See the supplied ZTEC User's Manual for complete specifications of the scope card. <sup>1</sup>

---

**NOTE** Refer to the [Reference Manual, Pulse source-measure connections, page 11-34](#) for details about scope card connectors and connections to DUT.

---

Table 1-6  
Scope card characteristics

Scope card characteristic	Specifications	
	Model 4200-SCP2HR (ZT410-50K)	Model 4200-SCP2 (ZT450-50K)
Dual channel	Simultaneous sampling of both channels	Simultaneous sampling of both channels
ADC resolution	16-bit	8-bit
Bandwidth	50 $\Omega$ : DC to 250 MHz 1 M $\Omega$ : DC to 125 MHz	50 $\Omega$ : DC to 1 GHz 1 M $\Omega$ : DC to 350 MHz
Maximum input	50 $\Omega$ : $\pm 5$ V DC 1 M $\Omega$ : $\pm 25$ V DC (derated 20 dB / decade above 10 MHz)	50 $\Omega$ : $\pm 5$ V DC 1 M $\Omega$ : $\pm 150$ V DC (derated 20 dB / decade above 1 MHz)
Coupling	DC or AC	DC or AC
AC coupling	50 $\Omega$ : 200 kHz high-pass 1 M $\Omega$ : 10 Hz high-pass	50 $\Omega$ : 200 kHz high-pass 1 M $\Omega$ : 10 Hz high-pass
Probe attenuation	0.9 to 1000:1	0.9 to 1000:1
Analog filter	N/A	20 MHz or bypass
Total memory	Up to 1 M S per channel Up to 2 M S per channel (1 channel interleaved)	Up to 1 M S per channel Up to 2 M S per channel (1 channel interleaved)
Sample (S) rate	10 k S / s to 200 M S / s 10 k S / s to 400 M S / s (1 channel interleaved)	2.5 k S / s to 1.25 G S / s 2.5 k S / s to 2.5 G S / s (1 channel interleaved)
Acquisition time range	250 ns to 3355 s	50 ns to 419 s (2 M sample memory)

---

**NOTE** All specifications are subject to change; for the latest specifications, visit ([www.ztecstruments.com](http://www.ztecstruments.com)).

---

## Scope card settings

The following information summarizes the most frequently used settings for the scope. For detailed information about all scope settings.

Keithley Instruments user modules are used to control waveform acquisition operations of the scope. New user modules can be created, or existing user modules can be modified (see the [Reference Manual, Keithley Interactive Test Environment \(KITE\), page 6-1](#) for details). For more information about ZTEC, refer to the Model 4200-SCS Complete Reference, ZTEC User's Manual.

---

1. The ZTEC User's Manual is located on your Model 4200-SCS Complete Reference Product Information CD.



### Input impedance, input voltage range, and input voltage offset

Table 1-7 lists the input impedances, voltage ranges, and voltage offsets that can be set for each input channel. As shown in the table, the setting for each of these parameters depends on the settings of the other two parameters. For example, to select 50  $\Omega$  input impedance, range must already be set to one of the eight ranges listed in the table for 50  $\Omega$ , and voltage offset must not be set greater than  $\pm 10$  V.

To avoid settings conflicts, first set voltage offset to 0 V, and then select the 10 V range. These settings are compatible with both impedance settings. Now you can set impedance, range, and then offset, in that order.

Table 1-7  
Scope impedance, range, and offset settings

1M Ohm impedance				50 Ohm impedance			
Model 4200-SCP2HR		Model 4200-SCP2		Model 4200-SCP2HR		Model 4200-SCP2	
Range	Offset	Range	Offset	Range	Offset	Range	Offset
50 V pp	0 V	100 V pp	$\pm 50$ V	10 V pp	0 V	10 V pp	$\pm 5$ V
25 V pp	$\pm 12.5$ V	50 V pp	$\pm 25$ V	5 V pp	$\pm 2.5$ V	5 V pp	$\pm 2.5$ V
10 V pp	$\pm 5$ V	20 V pp	$\pm 10$ V	2 V pp	$\pm 1$ V	2 V pp	$\pm 1$ V
5 V pp	$\pm 5$ V	10 V pp	$\pm 5$ V	1 V pp	$\pm 1$ V	1 V pp	$\pm 0.5$ V
2.5 V pp	$\pm 5$ V	5 V pp	$\pm 2.5$ V	0.5 V pp	$\pm 1$ V	0.5 V pp	$\pm 0.25$ V
1.25 V pp	$\pm 5$ V	2.5 V pp	$\pm 1.25$ V	0.25 V pp	$\pm 1$ V	0.25 V pp	$\pm 0.125$ V
0.5 V pp	$\pm 5$ V	1 V pp	$\pm 0.5$ V	0.1 V pp	$\pm 1$ V	0.1 V pp	$\pm 0.05$ V
0.25 V pp	$\pm 5$ V	0.5 V pp	$\pm 0.25$ V	0.05 V pp	$\pm 1$ V	0.05 V pp	$\pm 0.025$ V
—	—	0.2 V pp	$\pm 0.1$ V	—	—	—	—
—	—	0.1 V pp	$\pm 0.05$ V	—	—	—	—

### Input coupling

Input coupling, which is used to pass or block the DC component of an input signal, can be set to AC or DC:

- DC coupling passes all frequencies
- AC coupling blocks low frequencies; with high-input impedance (1 M  $\Omega$ ) selected, AC coupling attenuates frequencies below 10 Hz; with low input impedance (50  $\Omega$ ) selected, AC coupling attenuates frequencies below 200 kHz

### Input filter

For the Model 4200-SCP2 scope, A 20 MHz low-pass analog filter can be applied to the input signal of each channel. The 20 MHz setting applies the filter, and the bypass setting bypasses the filter. The Model 4200-SCP2HR does not have the low-pass filter.

### Input attenuation

The input signal for each channel can be attenuated by a factor of 0.9 to 1000.0.

### Acquisition sample rate

The acquisition sample rate for the two input channels can be set in 1, 2.5, or 5 steps:

- **Model 4200-SCP2HR:** 10 k S / s to 200 M S / s (one interleaved channel can be sampled at 400 M S / s)
- **Model 4200-SCP2:** 2.5 k S / s to 1.25 G S / s (one interleaved channel can be sampled at 2.5 G S / s)



### Sweep mode (triggering)

There are two sweep modes to trigger scope measurements: Normal or auto. In the normal mode, an internal or external trigger (for example, trigger output from the Model 4205-PG2) is used to trigger measurements. In the auto mode, triggering is provided automatically by the scope in the absence of a trigger event.

For normal triggering, trigger initiation can be provided by internal and external sources. The scope can be set to be triggered by a leading-edge or falling-edge trigger from the Model 4205-PG2. For more information about trigger and arm controls, refer to the Model 4200-SCS Complete Reference, ZTEC User's Manual.

### Sweep offset reference

The offset reference determines when sampling occurs in relationship to the trigger event. Offset reference can be set from 0.0 (0%) to 1.0 (100%). For the following examples, assume the record size (sweep points) is 100 samples:

- **Post-trigger sampling:** With offset reference set to 0.0 (0%), sampling of all 100 sweep points will occur after the trigger event.
- **Pre-trigger sampling:** With offset reference set to 1.0 (100%), sampling of all 100 sweep points will occur before the trigger event.
- **Pre-trigger and post-trigger sampling:** With offset reference set to 0.5 (50%), 50 samples will be acquired before the trigger event, and 50 samples will be acquired after the trigger.

A sweep offset time can also be used to affect when sampling occurs.

### Sweep offset time

The sweep offset time is the time period between the trigger event and the sweep offset reference. Offset time can be set from 0 to 665 seconds. For the following examples, assume the record size (sweep points) is 100 samples and offset time is set to 1 second:

- **Delayed sampling from 0% offset reference:** With offset reference set to 0.0 (0%), sampling for all 100 sweep points will start one second after the trigger event.
- **Delayed sampling from 100% offset reference:** With offset reference set to 1.0 (100%), all 100 samples will have been acquired at the "trigger plus one second" point in time.
- **Delayed sampling from 50% offset reference:** With offset reference set to 0.5 (50%), 50 samples will have been acquired at the "trigger plus one second" point in time. Sampling will continue for the other 50 sweep points.

### Average type (acquisition)

There are four waveform acquisition types that can be set for the scope: Normal, average, envelope, or equivalent time:

**Normal:** In normal mode, a single waveform is captured.

**Average:** In average mode, multiple captured waveforms are averaged.

**Envelope:** In envelope mode, the minimum and maximum waveform points from multiple acquisitions are combined to form a waveform (an envelope) that shows min/max changes over time.

**Equivalent time:** In equivalent time mode, a picture of a repetitive waveform is constructed by capturing a small amount of information from each repetition. Because the points appear randomly along the waveform, it is important to note that an entire waveform may not be constructed unless there are sufficient repetitions. Unfilled points will be constructed using a zero-order hold and are flagged with a "1" in the least significant bit (LSB) of the 16-bit waveform code. Also, the number of points per point can be set using average equivalent time points to increase the resolution of the waveform.



### Average equivalent time points

When using the equivalent time acquisition mode, the number of user-defined points per point for equivalent time sampling of a waveform can be set. Average equivalent time points can be set from 2 to 100.

When using equivalent time sampling, any signal up to the analog bandwidth of the scope can be acquired, regardless of the sample rate. The scope gathers the necessary number of samples across several triggers. For more information about average equivalent time points command, refer to the Model 4200-SCS Complete Reference, ZTEC User's Manual.

### Reference channels

Up to four waveforms can be stored in nonvolatile flash memory as reference channels. The stored waveforms are retained when power is removed. These waveforms are limited to a record size of 32K samples.

### Calculate functions

The scope has two calculation channels to create new waveforms mathematically.

**The following calculate functions can be performed:**

- Add
- Subtract
- Multiply
- Copy
- Invert
- Integral
- Derivative
- Absolute Value
- Limit test
- Mask test
- Frequency transform
- Time domain transform

The waveforms for a calculation can be input channels (2) and waveforms that are stored in memory as reference channels (4). For more information about ZTEC, refer to the Model 4200-SCS Complete Reference, ZTEC User's Manual, Chapter 2.

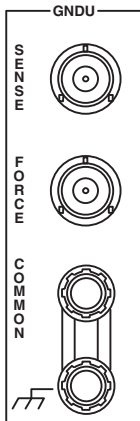
## Ground unit (GNDU)

### Basic characteristics

The ground unit (see [Figure 1-28](#)) provides convenient access to circuit COMMON, which is the measurement ground signal shared by all installed Model 4200-SCS instrumentation. In addition, the GNDU SENSE terminal provides access to the SMU SENSE LO signals.



Figure 1-28  
Ground unit



Basic ground unit characteristics are summarized in [Table 1-8](#).

Table 1-8  
Basic ground unit characteristics

Characteristic	Description
Maximum current (FORCE triax connector)	2.6 A
Maximum current (COMMON binding post connector)	9 A
Maximum FORCE path/cable resistance	1 $\Omega$
Maximum SENSE path/cable resistance	10 $\Omega$

## Basic circuit configurations

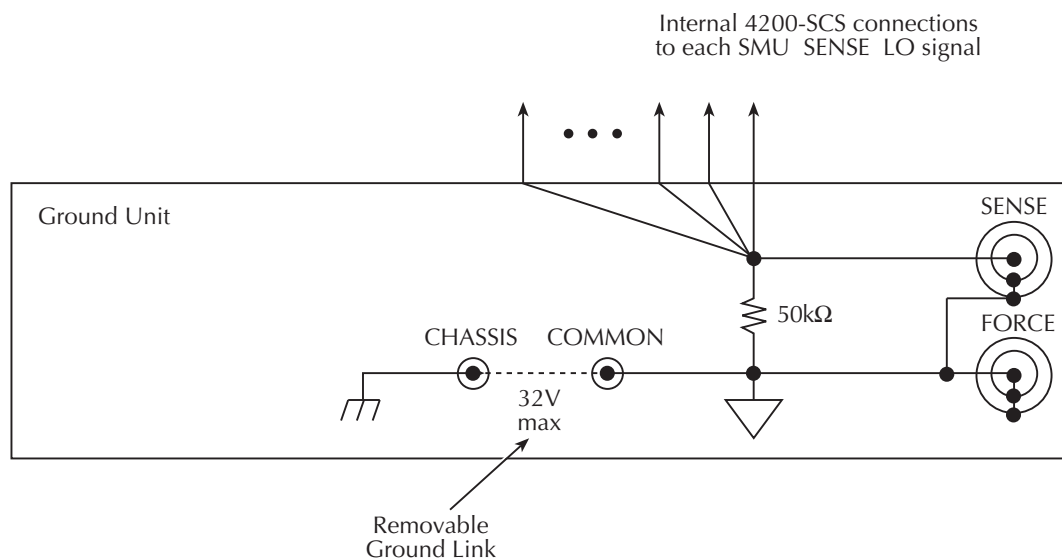
### Ground unit connections

[Figure 1-29](#) shows how the various GNDU signals are related to the SMU signals. Note that the GNDU FORCE signal is circuit COMMON. The GNDU SENSE terminal is connected to each SMU SENSE LO signal through a unique auto-sense resistor.

When the GNDU SENSE signal is connected to a DUT, all measurements will be made relative to this DUT connection.



Figure 1-29  
Ground unit



### Ground unit DUT connections

Figure 1-30 shows the connections necessary to use the GNDU in conjunction with a SMU to make full-Kelvin remote sense measurements. Similarly, Figure 1-31 includes the preamp.

As shown in these figures, the GNDU FORCE signal provides the return path for SMU or preamp FORCE current. For detailed information about the ground unit, SMU, and preamp connections, refer to the [Reference Manual, Basic source-measure connections, page 4-3](#).

Figure 1-30  
Full-Kelvin SMU/ground unit connections

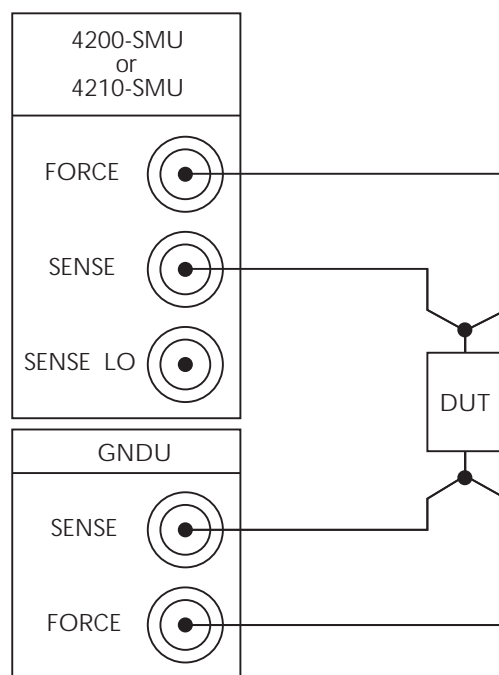
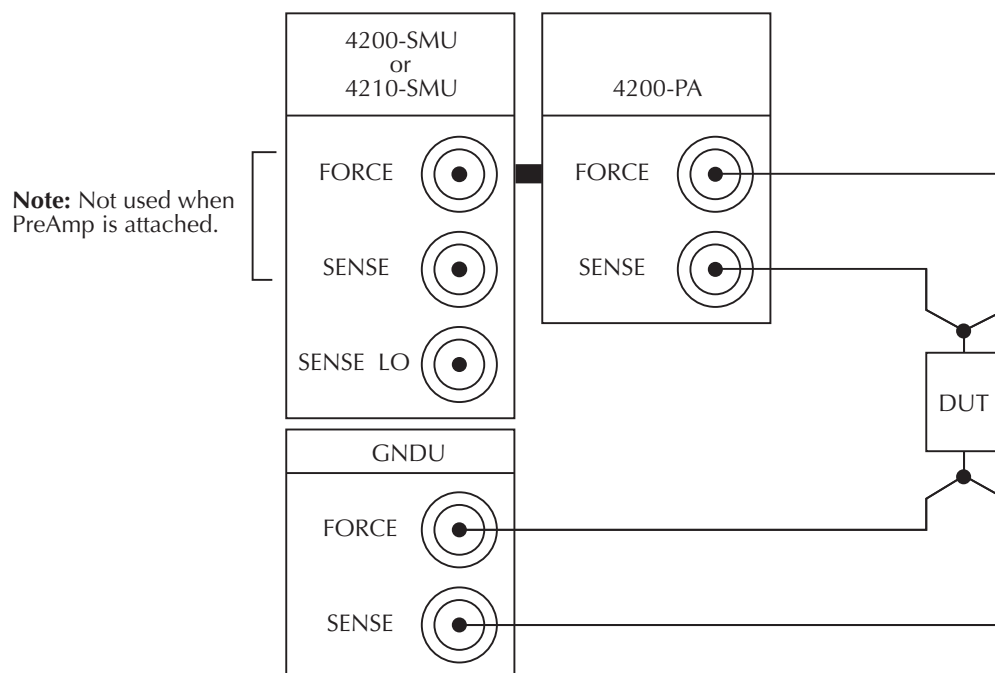




Figure 1-31  
Full-Kelvin preamp/ground unit connections



## Ground unit terminals and connectors

The locations and configuration of the GNDU terminals are shown in [Figure 1-28](#). Basic information about these connectors is summarized below. For more information about ground unit signal connections, refer to the [Reference Manual, Basic source-measure connections, page 4-3](#).

---

**CAUTION** The maximum allowed voltage between circuit **COMMON** and chassis ground is  $\pm 32$  V DC.

---

### FORCE terminal

The FORCE terminal is a standard triaxial connector used as a return path for the SMU or preamp FORCE current.

- The center pin is FORCE
- The inner shield is GUARD
- The outer shield is circuit COMMON

---

**NOTE** The ground unit FORCE and GUARD signal terminals are connected to circuit COMMON.

---

### SENSE terminal

The SENSE terminal is a standard triaxial connector used to apply the ground unit SENSE signal to the DUT in a remote sense application.

- The center pin is SENSE
- The inner shield is GUARD



- The outer shield is circuit COMMON

When the ground unit SENSE signal is connected to a DUT, all SMU/preamp measurements will be made relative to this DUT connection.

## COMMON terminal

The COMMON terminal is a binding post that provides access to circuit COMMON.

---

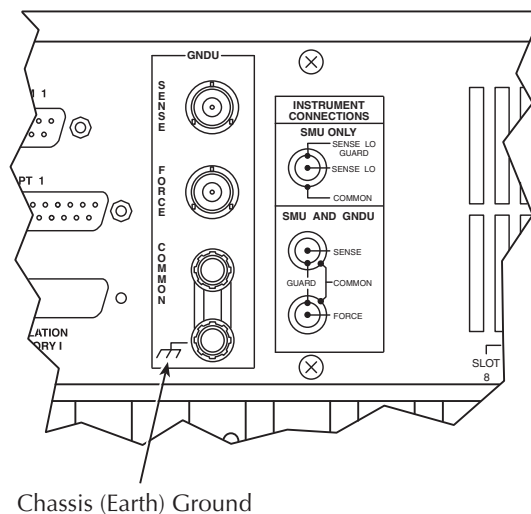
**NOTE** Normally, a link is connected between ground unit COMMON and chassis ground, but it may be necessary to remove the link to avoid measurement problems caused by ground loops or electrical interference. Refer to the [Reference Manual, Interference, page 5-25](#) for details.

---

## Chassis ground

This binding post provides a convenient connecting point to system chassis ground for the purpose of shielding a test fixture.

Figure 1-32  
Chassis ground



## Connecting DUTs

### Test fixtures

There are two types of test fixtures for the Model 4200-SCS:

- Low-voltage fixtures (less than  $\pm 20$  V) and high-voltage (greater than  $\pm 20$  V).
- High-voltage fixtures require extra precautions to ensure there are no dangerous shock hazards.



---

**WARNING** *To avoid exposure to high voltages that could result in personal injury or death, whenever the interlock of the Model 4200-SCS is asserted, the FORCE and GUARD terminals of the SMUs and preamps should be considered high-voltage, even if they are programmed to a non-hazardous voltage current.*

---

### Testing with less than $\pm 20$ V

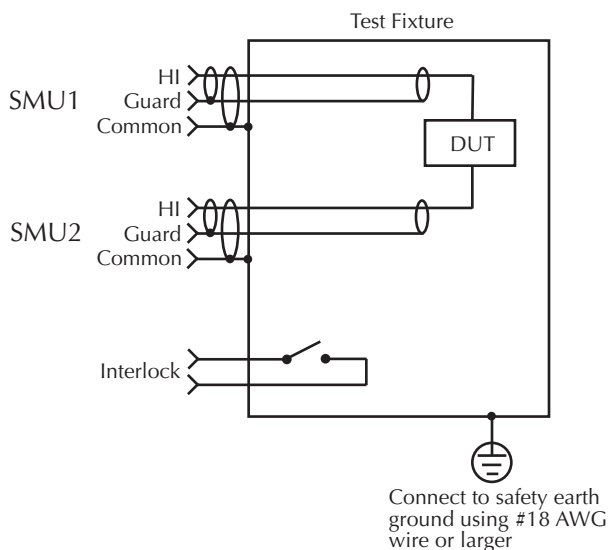
A test fixture equipped with three-lug triax connectors is necessary to connect the Model 4200-SCS discrete device for testing. Figure 1-33 shows a basic test fixture to use with a two-terminal device. For best performance when testing with less than  $\pm 20$  V follow these standard industry practices:

- Use a metal test fixture
- Connect the metal fixture to COMMON
- Mount the DUT on high-resistivity terminals (for example, Teflon)
- Guarding will reduce leakage and parasite capacitance that degrades measurement quality

The Keithley Instruments Low Level Measurements Handbook provides an in-depth discussion about guarding and other techniques that are useful for building quality test fixtures.

Contact a Keithley Instruments sales or service office to obtain a copy.

Figure 1-33  
Typical test fixture




---

**NOTE** *The Model 4200-SCS will function on all current ranges and up to  $\pm 20$  V without the interlock being asserted. The maximum voltage on the SMU and preamp terminals is not hazardous when the interlock is not asserted.*

---

### Testing with more than $\pm 20$ V

If voltages greater than  $\pm 20$  V are required for testing, follow these steps:

- Ensure that hazardous voltages are not present when the fixture's exterior enclosure is open.



---

**NOTE** *The Model 4200-SCS voltage output will be higher when the fixture's exterior enclosure is closed.*

---

- Add an interlock switch to the fixture.
- Connect the exterior enclosure to COMMON or safety ground using #18AWG wire or greater.
- Ensure that the wiring (FORCE, GUARD, and SENSE) within the fixture does not contact the exterior enclosure.

For more details about the Model 4200-SCS interlock system, see the [Reference Manual, Control and data connections, page 4-21](#).

---

**CAUTION** **Asserting the interlock will allow the SMU and preamp terminals to become hazardous, possibly exposing the user to high-voltage that could result in personal injury or death.**

**SMU and preamp terminals should be considered hazardous even if the outputs are programmed to be low voltage.**

**Precautions must be taken to prevent a shock hazard by surrounding the test device and any unprotected leads (wiring) with double insulation for 250 V, Category I.**

---

## Probers

To learn how probe stations are controlled by Keithley Instruments Model 4200-SCS through user modules refer to the [Reference Manual, Using a Probe Station, Appendix G](#).

## Advanced connections

To learn more about advanced connections for controlling a switch matrix, sequencing test on multiple devices, and customizing a user test module (UTM) refer to the [Reference Manual, Advanced Applications, Appendix O](#).

## How to run a basic test

### Boot the system and log in

**To boot the system and log in:**

1. A. Make sure the power switch is in the **O** (out) position. The POWER switch is located on the front panel in lower right-hand corner.
2. B. Plug the male end of the line cord into a properly grounded AC line power receptacle.
3. C. Turn on the Model 4200-SCS by pushing in the POWER switch to the **I** (in) position.
4. D. When prompted, simultaneously press Ctrl + Alt + Del.
5. E. At the KIUSER prompt, press ENTER. There is no password for this account.

---

**CAUTION** **When first starting a KTE-Interactive software tool, you must answer "YES" to an on-screen license agreement. Answering "No" makes your system nonfunctional until you reinstall the software.**

---



## Open KITE

To start KITE, open the default project, and select the vds-id test:

1. Start KITE by double-clicking the **KITE** icon  on your Windows® desktop.
2. When KITE starts, the default project will open automatically if it has been set as the default project. The vds-id test will be opened in the Definition tab. (see [Figure 1-35](#))
3. If a different project opens, perform the three steps in [Figure 1-36](#) to open the default project. The project navigator for the default project is shown in [Figure 1-35](#).

---

**NOTE** If the project navigator is not displayed when KITE is started, click the **View** menu and select the **Project Navigator** item. The **View** menu is located at the upper left-hand side of the KITE window. For more information about the project navigator, refer to the [Reference Manual, Project Navigator, page 6-12](#).

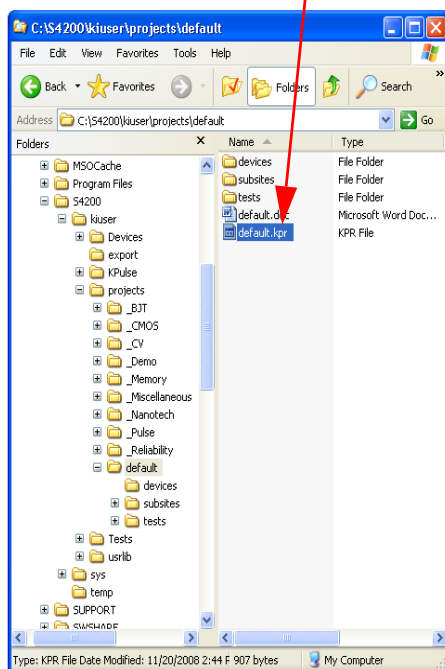
---

To open a KITE default project, follow the steps in [Figure 1-34](#).

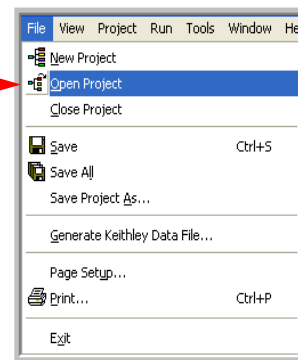
Figure 1-34

### Default project directory

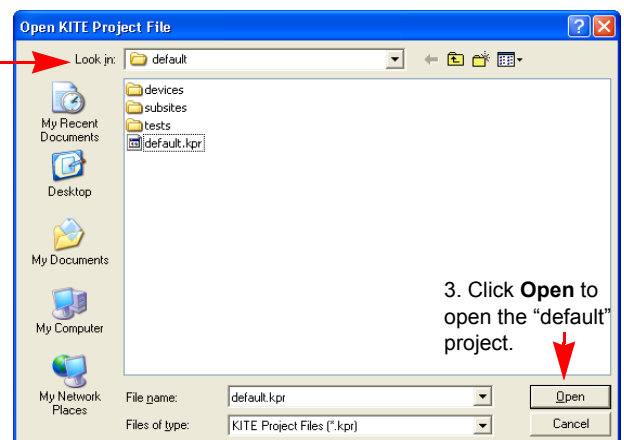
From your windows browser, use the following directory path to locate the default.kpr project file: C:\S4200\kiuser\projects\default.



1. From the **File** menu, click **Open Project**.



2. Use the browser to select the **default** project.



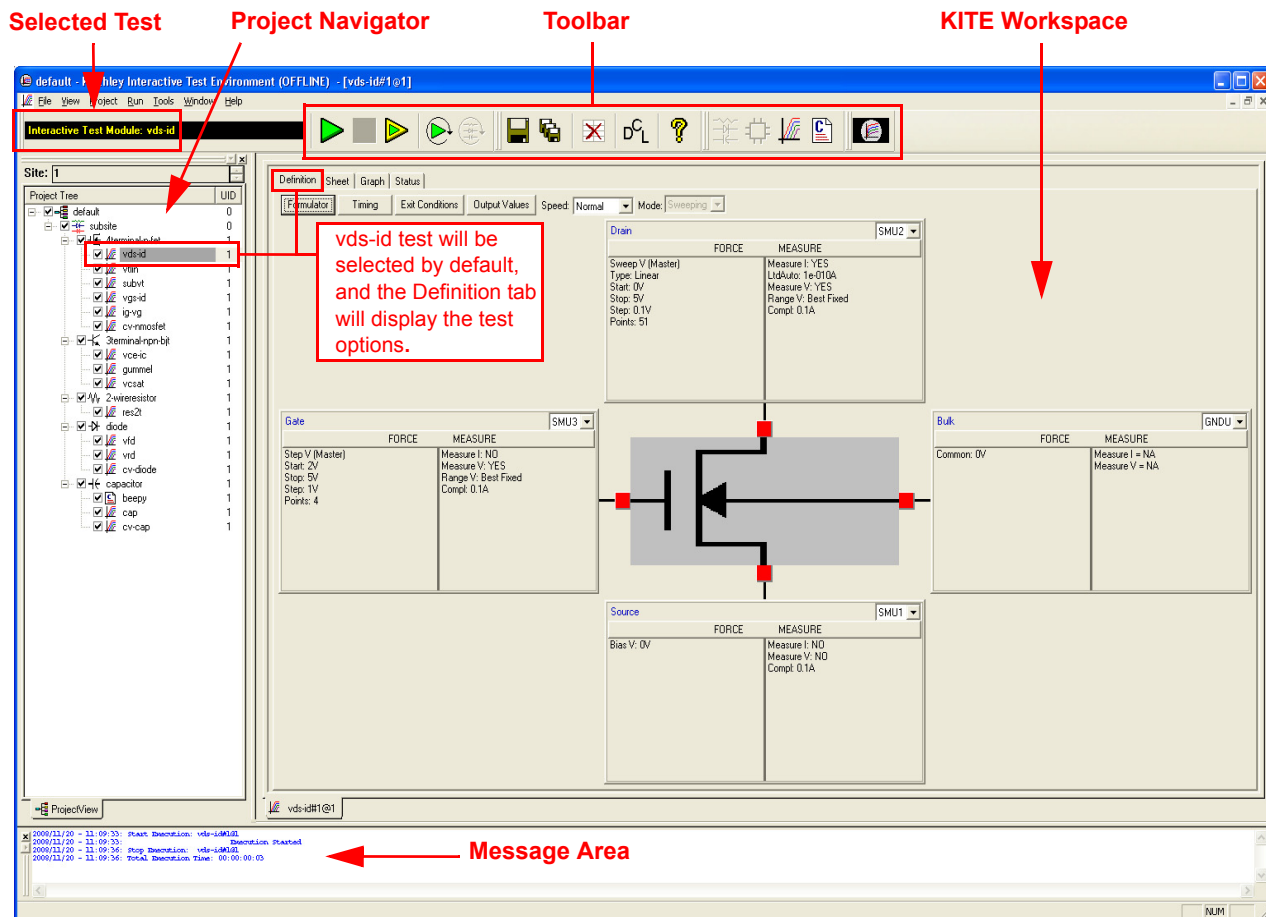
3. Click **Open** to open the "default" project.



Figure 1-35 is an example of the KITE interface default project with the vds-id test selected.

Figure 1-35

KITE interface default project shown with vds-id test selected



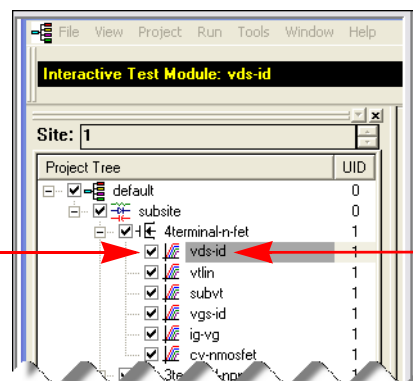
## Locate and run the vds-id test module

To locate the vds-id test module go to the project navigator as shown in Figure 1-36.

Figure 1-36

Interactive Test Module vds-id

The checkbox for vds-id must be checked in order to run the test. If unchecked, click the checkbox to insert a ✓.



Double click the vds-id test icon, and the test Definition tab as shown in Figure 1-35 is displayed.



## Test definition

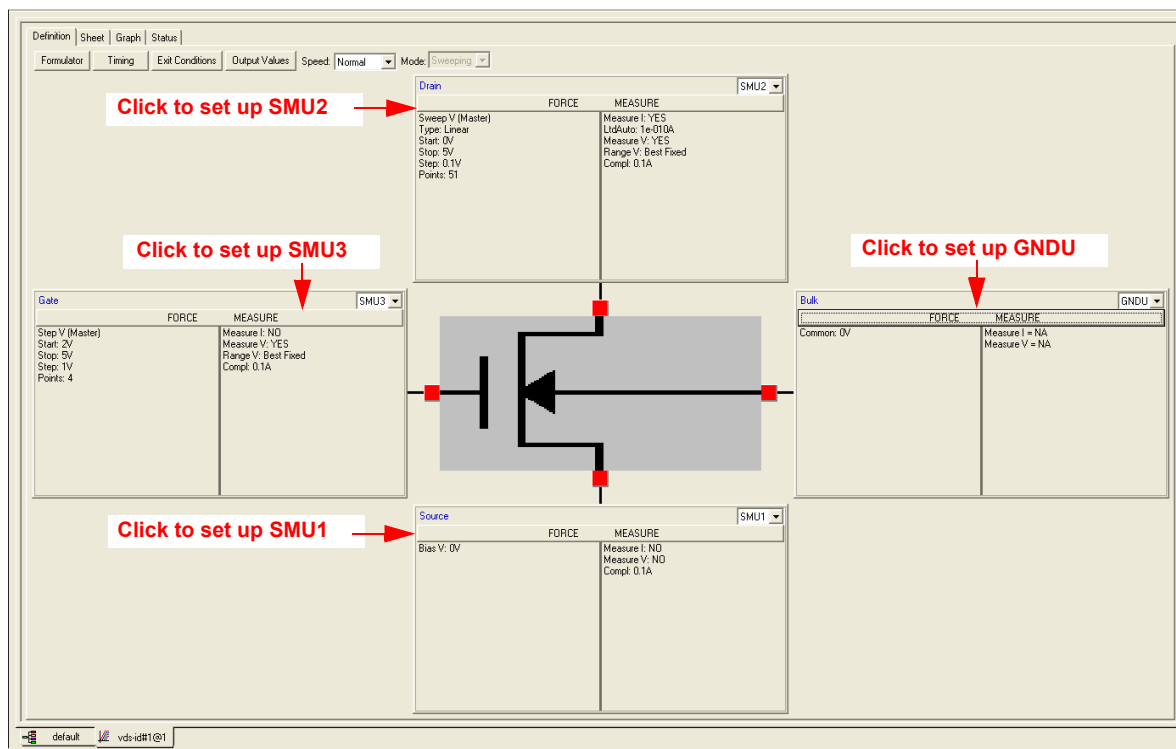
The test is defined from the test Definition tab shown in [Figure 1-37](#). As shown in the Definition tab, the device is connected to three SMUs and one Ground Unit (GNDU). In general, SMU3 is used as a voltage step function to provide four different gate voltages (2V, 3V, 4V, and 5V). SMU2 is used to perform a 51 point sweep of drain voltage (0V to 5V) at each gate voltage. A current measurement is performed at each voltage sweep point.

### To define a test:

- A. If desired, the setup for SMUs and the GNDU can be changed. A settings window is displayed by clicking the appropriate force measure bar as shown in [Figure 1-37](#).

Figure 1-37

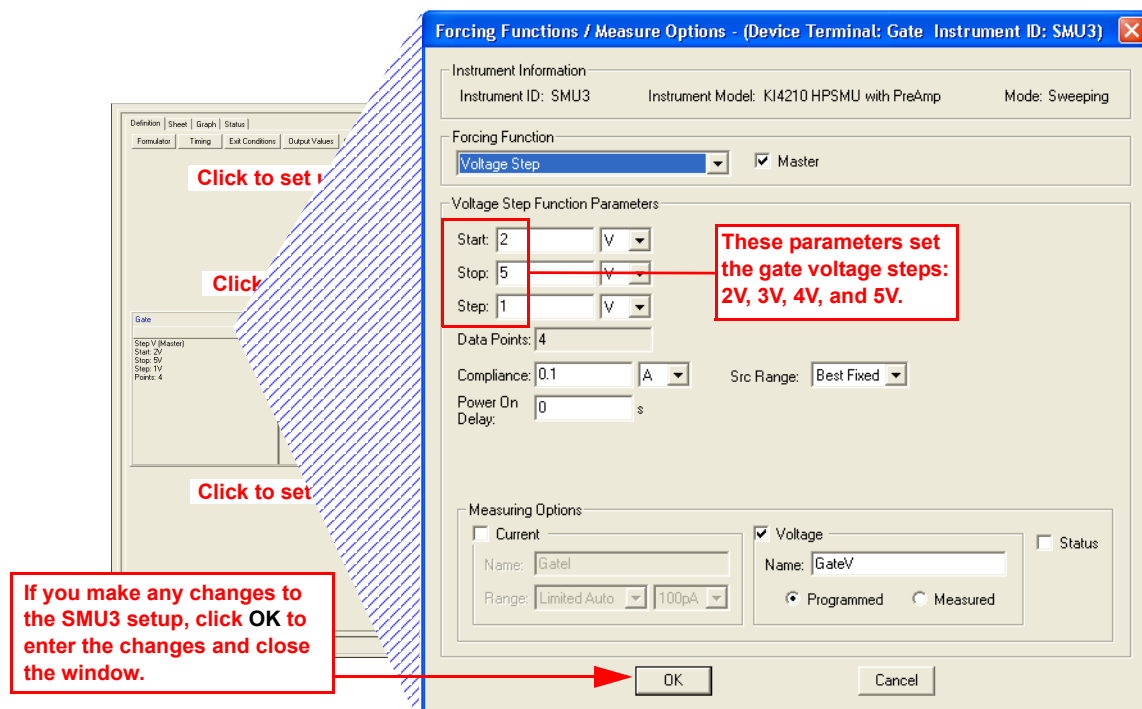
**Vds-id Definition tab - How to display a setup window for SMUs and GNDU**




- B. [Figure 1-38](#) shows the settings dialog box for SMU3. The settings dialog box for the other SMUs and GNDU are similar.



Figure 1-38  
Setup for SMU3




- C. After making any changes to the test definition, click **Save All**  on the toolbar to save the settings.

## Run vds-id test

In the project navigator (see [Figure 1-36](#)), make sure the vds-id test is highlighted and the checkbox is checked.

On the toolbar, click the green **Run Test**  button to run the test one time.

**NOTE** While the test is running, the Run Test button turns gray and the Abort Test button turns red . Also, the ACTIVE indicator light (located on the lower right-hand corner of the front panel of the Model 4200-SCS) will be on while the test is running. When the test is finished, the Run test button turns green.

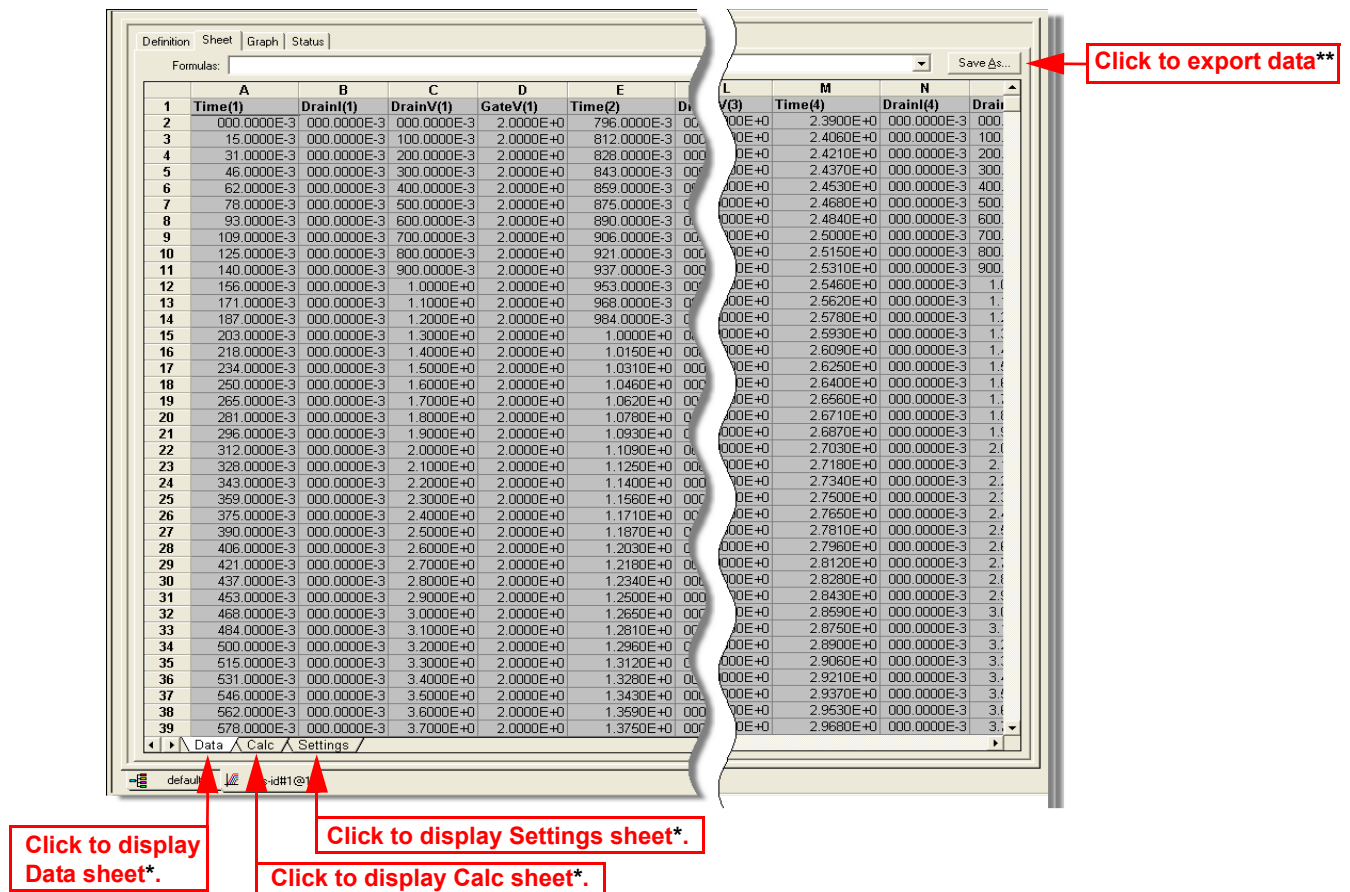
## View and save the sheet data

View data sheet for the vds-id test is displayed by clicking the Sheet tab for the test. Use the tabs at the bottom of the Sheet to display the data type. A sample data sheet for the vds-id test is shown in [Figure 1-39](#).

The data sheet for the vds-id test is displayed by clicking the Sheet tab.



Figure 1-39  
Sample data sheet for vds-id test



\* To select more than one sheet for selective printing, hold down the **Ctrl** key and then click the tab.

\*\* The data is saved as a Microsoft® Excel® ( \*.xls ) document.

#### To save the Data sheet:

1. In the KITE workspace, click the **Sheet** tab to display the test data.
2. In the Sheet tab, click the **Save As** button as shown in [Figure 1-39](#).
3. From the **Save As** dialox box, specify a file name and path, and click **Save**.

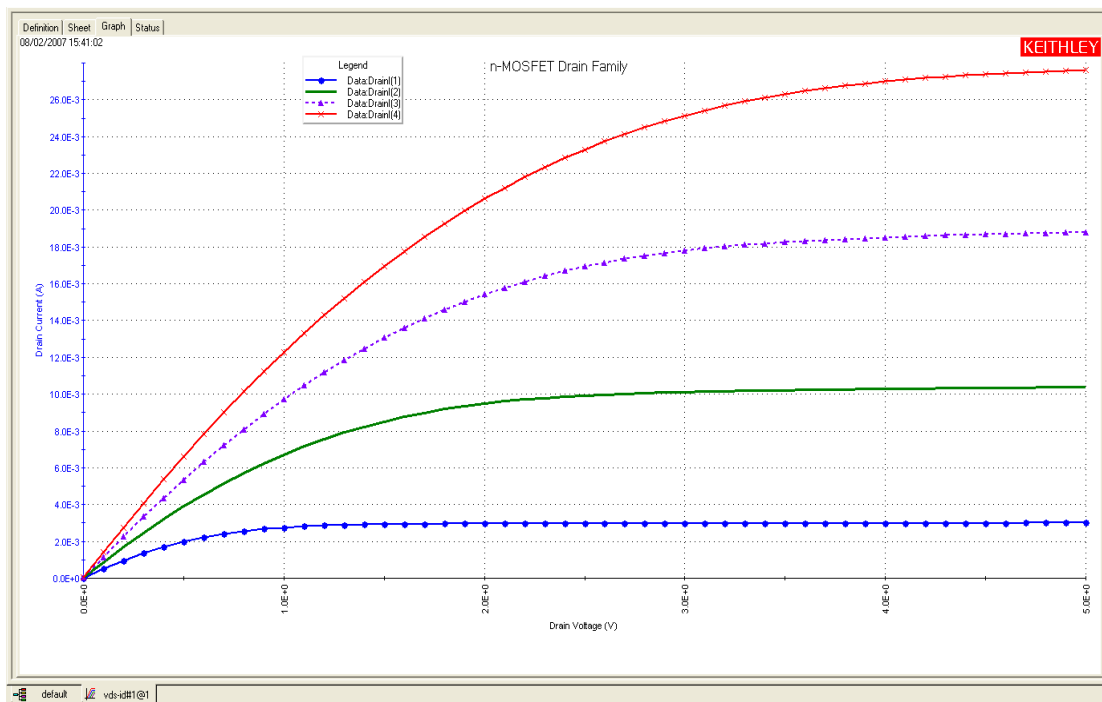
**NOTE** The default directory path for exporting data is *C:\S4200\kiuser\export*.

## View and save the graph data

The graph for the vds-id test is displayed by clicking the Graph tab for the test. A sample graph for the vds-id test is shown in [Figure 1-40](#). As shown, there are four I-V curves - one for each gate voltage. The graph was customized to include the Legend box and use different colors for the graph series. The Graph Settings menu (shown in [Figure 1-41](#)) was used to select the Legend box and change series colors. For more Graph tab information Refer to the [Reference Manual, Viewing ITM or UTM results graphically: The Graph tab, page 6-21](#).

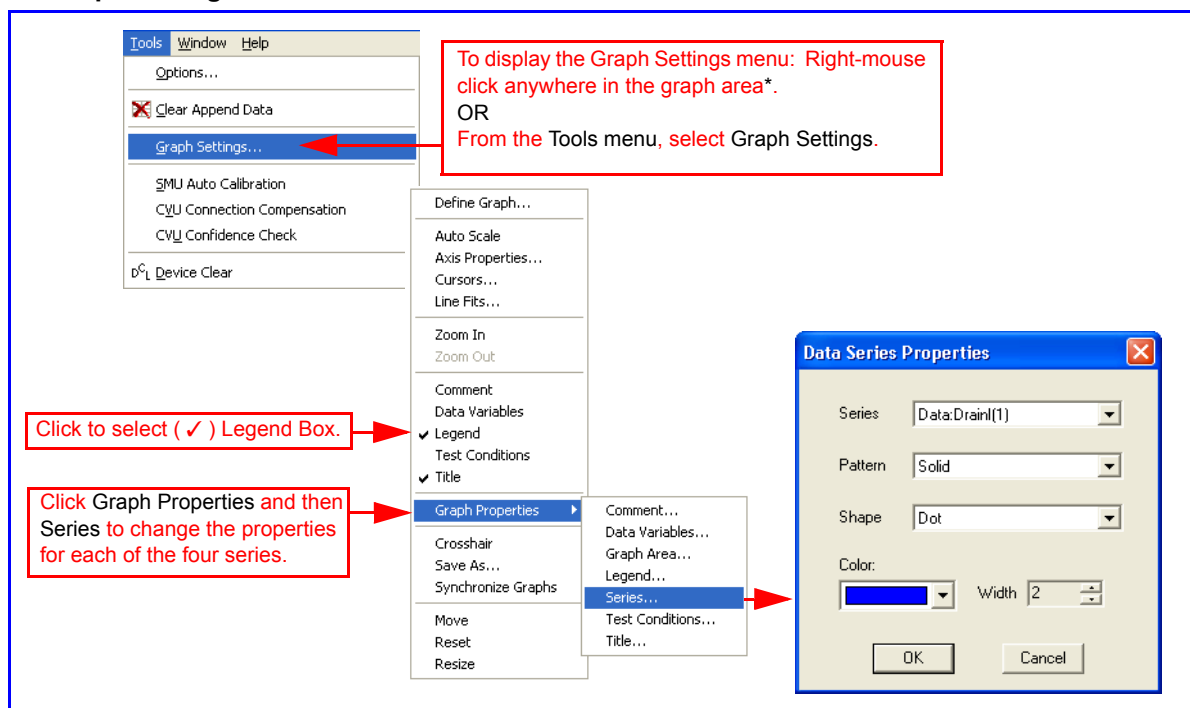


Figure 1-40  
Sample graph for vds-id test



In Figure 1-40 you can see that the line colors, line patterns, plot symbols, and line widths are different. To learn how to define the graph line properties see Figure 1-41 below.

Figure 1-41  
Graph settings menu



\* To learn more about Data Series Properties refer to the [Reference Manual](#), Defining the plot properties of the graph: colors, line patterns, symbols, line widths, page 6-224.



---

## Model 4200-SCS Software Environment

In this section:

Topic	Page
<b>Understanding KITE</b> .....	2-3
<b>KITE project structure</b> .....	2-3
KITE interface .....	2-3
Graphical User Interface .....	2-4
Project navigator .....	2-5
Project defined .....	2-6
Project components .....	2-7
Sites .....	2-7
Subsites .....	2-7
Devices .....	2-7
Tests .....	2-7
<b>ITMs versus UTMs</b> .....	2-8
Defining an ITM .....	2-10
Defining a UTM .....	2-11
<b>How to create your own ITMs</b> .....	2-13
Understanding the ITM definition tab .....	2-13
Understanding the ITM forcing functions .....	2-13
Understanding dual sweep .....	2-15
Understanding pulse mode .....	2-16
How to use the definition tab to configure ITM parameters .....	2-17
Match the physical and virtual connections: .....	2-17
Configuring forcing functions for each device terminal .....	2-17
Configuring pulse mode .....	2-20
<b>Basic test execution</b> .....	2-21
Project navigator check boxes .....	2-21
Tests (ITMs and UTMs) .....	2-21
Device plan .....	2-22
Subsite plan .....	2-22
Initialization and termination steps .....	2-22
Project plan .....	2-22
Executing an individual test .....	2-22
Selecting a test .....	2-22
Running the test .....	2-23
<b>How to display and manage test results</b> .....	2-24
Data file management .....	2-24
Using file and test-result directories .....	2-24
Default user director: C:\S4200\kiuser .....	2-24
Devices subdirectory .....	2-25
Understanding device libraries .....	2-25
How to create and add a new device .....	2-27
Projects subdirectory .....	2-28
Tests subdirectory .....	2-29
Usrlib subdirectory .....	2-31
System directory: C:\S4200\sys .....	2-31
How to manage numeric test results in Sheet tab .....	2-31
Displaying and analyzing data using Sheet tab .....	2-31



Opening a Sheet tab .....	2-32
Understanding and using the Data worksheet of a Sheet tab . . .	2-34
Understanding the formula box of the Data worksheet .....	2-34
Understanding the data-source identifier .....	2-35
Saving a worksheet .....	2-35
How to manage graphical test results in the Graph tab .....	2-38
Opening a Graph tab .....	2-38
Accessing the Graph tab windows .....	2-39
Opening the graph settings menu .....	2-39
Understanding the graph settings menu .....	2-40
Defining data to be graphed .....	2-42
Understanding table columns in the Graph Definition window .	2-42
Understanding buttons in the Graph Definition window .....	2-43
Opening and using the Graph Definition window .....	2-43
<b>KITE library management .....</b>	<b>2-45</b>
Submitting devices, ITMs, and UTMs to libraries .....	2-45
Submitting devices to a library .....	2-45
Submitting tests to a library .....	2-48



## Understanding KITE

The Keithley Interactive Test Environment (KITE) is the main software component of the KTE Interactive software tool set. KITE is the primary user interface for the Keithley Instruments Model 4200 Semiconductor Characterization System (SCS). KITE is a versatile tool that facilitates interactive characterization of an individual parametric test device or automated testing of an entire semiconductor wafer.

### Two additional KTE Interactive software tools augment the capabilities of KITE:

- **Keithley User Library Tool (KULT):** Used to create test modules, using the C programming language. These test modules can then be executed by KITE.
- **Keithley CONfiguration utility (KCON):** Used to manage the configuration and interconnections between all of the test system components that are controlled by KITE.

A fourth KTE Interactive software tool, the Keithley External Control Interface (KXCI) allows the Model 4200-SCS to be controlled remotely by an external GPIB controller.

---

**NOTE** *KXCI and KITE, cannot run simultaneously.*

---

Beginning with KTE Interactive 6.0, two optional KTE Interactive tools have been added:

- **Keithley Pulse tool (KPulse):** A virtual front panel software application used to control the optional pulse generator cards. The dual-channel pulse generator cards are integrated inside the Model 4200-SCS mainframe.
- **Keithley Scope tool (KScope):** A virtual front panel software application used to control the optional scope card. The scope card is a dual-channel digital storage oscilloscope that is integrated inside the Model 4200-SCS mainframe.

---

**NOTE** *Although KScope and KPulse can be launched at the same time as KITE, KScope / KPulse and KITE cannot communicate with hardware simultaneously.*

---

## KITE project structure

This subsection overviews the primary features of KITE. These features allow you to create, execute, and evaluate tests and complex test sequences, interactively and without programming. This subsection also overviews use of an essential companion tool, KULT, that allows you to create libraries of specialized user test modules (UTMs) that run in KITE (KULT is discussed in detail in the [Reference Manual, Keithley User Library Tool \(KULT\), page 8-1](#)).

## KITE interface

**The KITE application consists of a variety of graphical user interfaces (GUIs) that allow you to do the following:**

- Customize existing / supplied interactive test modules (ITMs) or create new ITMs from existing templates.
- Create UTMs from supplied or user-programmed C-code modules.
- Automatically execute tests and associated operations (switch matrix connections, probe movements, and so on), including:
  - a. A single test for one selected device (transistor, diode, resistor, capacitor, and so on).
  - b. A sequence of tests for one selected device.
  - c. A sequence of tests for multiple devices, for example, all of the devices contacted by a probe at a given touchdown or subsite location on a semiconductor wafer.

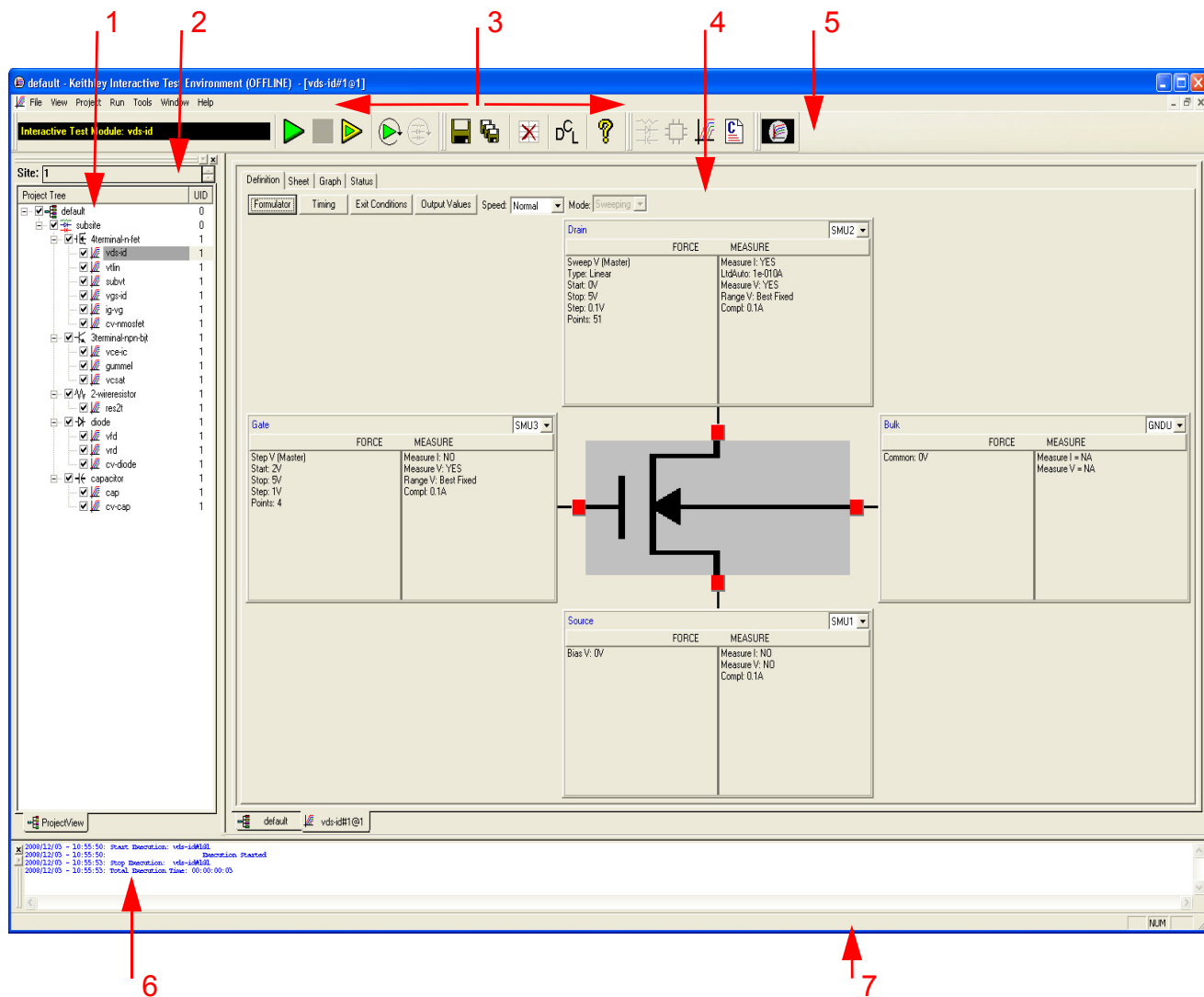


- d. A sequence of tests for an entire project plan, which may include multiple prober touchdowns for a single semiconductor site (or die):
  - For one site
  - For multiple sites
- View test results numerically and graphically.
- Analyze test results using built-in parameter extraction tools.
- View the analysis results numerically and graphically.
- Interactively build and edit test / execution sequences using the project navigator.

## Graphical User Interface

Figure 2-1 shows the KITE GUI interface. The various parts of the GUI are summarized below the figure.

Figure 2-1  
KITE interface overview



### KITE interface descriptions:

1. **Project navigator** Where a project plan is assembled, edited, displayed, and executed. A project plan defines a series of tests, of various devices, at one or more locations. Double-clicks



- here open the definition, configuration, and tool screens. A selection here defines the starting location when only part of the project plan is to be executed.
2. Site navigator Displays the current site. Typically a die on a semiconductor wafer that is being evaluated by the project plan. Allows selection of the single site to be evaluated when only part of the project plan is to be executed. A unique ITM / UTM test window and data exists for each different site.
  3. Menu area Provides choice of icons to select KITE functions.
  4. KITE workspace Displays the variety of screens, windows, tabs, message boxes, and so on:
    - Configure all project plan components
    - Observe evaluation results
    - Analyze evaluation results.
  5. Toolbar area Displays a variety of icons that can be used to:
    - Start and stop all or part of a project plan
    - Verify project plan execution
    - Insert project plan components
    - Save and print project plan files
    - View KITE help.
  6. Message area Displays KITE error, warning, and execution messages.
  7. Status bar Displays descriptions of menu and toolbar items.

## Project navigator

The project navigator is the primary interface for building, editing, viewing, specifying, and accessing project plan components:

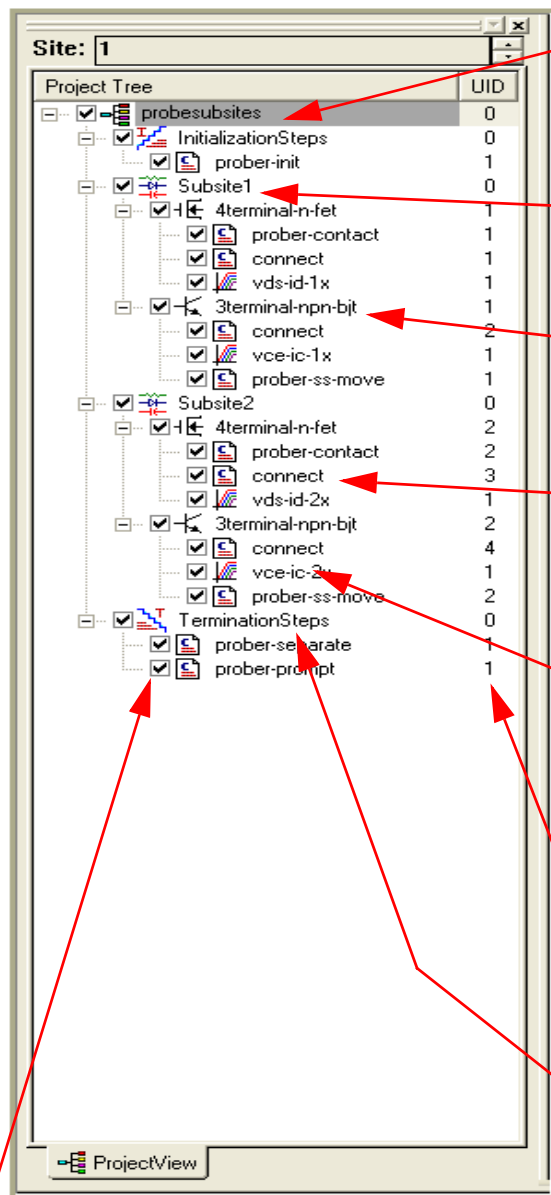
- Each project plan component may be added, sequentially or nonsequentially using menu items or toolbar buttons.
- Single-clicking a navigator component selects it as one of the following:
  - A location where a new component may be added or an existing component may be deleted.
  - An individual test, device, or series of devices for which only part of the project plan may be executed.
- Double-clicking a project navigator component opens access to configuration screens for the component and, as appropriate, test results, analysis tools, and status information.

Figure 2-2 describes typical project plan components that are displayed in a project navigator, using the **example** project plan for illustration. Subsection [ITMs versus UTMs on page 2-8](#) describes the ITM and UTM components in more detail.

For details about building a project plan using the project navigator, refer to the [Reference Manual, Building, modifying, and deleting a Project Plan, page 6-47](#).



Figure 2-2  
Project navigator



**Project plan:**

- Defines and sequences all subsites to be tested.
- This node allows resequencing of all subsite nodes.
- There is **only one** project plan per KITE project.

**Subsite plan:**

- Defines and sequences all device plans to be tested.
- This node allows resequencing of all device nodes.

**Device plan:**

- Defined and sequences all tests for a specific device.
- This node allows resequencing of all tests.

**Interactive test module (ITM):**

- A graphical test definition that requires no programming.
- All test definition, data, and graphing is located here.
- Can be used for all instrumentation internal to the 4200.

**User test module (UTM):**

- A test definition that is a call into user module written in C.
- All test definition, data, and graphing is located here.
- Can be used for all instrumentation internal and external to the 4200.
- Typically used for switch-matrix connections, prober control, and all other external GPIB and Ethernet instruments.

**Device unique ID number (UID):**

A number assigned to each instance of a same-named project component (for example: ITM, UTM, Device, or Subsite node) and only once per KITE Project.

**NOTE** If UID=0, this means that this component can occur only once.

**Initialization and Termination Steps:**

A special UTM that can be run just like any other UTM, however, during looping, these nodes are called **only once** at the beginning (Init Steps) or at the end (Term Steps) of a test looping sequence.

**Project navigator check boxes:**

These are used to enable and disable each project node and all of its children. If a particular project node is unchecked, the project run button will be grayed / disabled and this particular project node cannot be executed.

## Project defined

Users interact with KITE in the context of project. A project specifies the start-to-finish, repetitive and nonrepetitive actions and test locations involved in evaluating a semiconductor wafer (or other collection of circuits). Projects are both created and executed using the KITE GUI.



---

**NOTE** Refer to [Project plan](#) later in this section. The term “project” is sometimes used to refer to a “project plan.”

---

## Project components

Because KITE is most valuable for automatic characterization of semiconductor wafers, KITE projects are organized in a manner consistent with the organization of a modern semiconductor wafer. A project visits and evaluates locations on the wafer in the following logical hierarchy:

- Project
  - Sites (Virtual, can be switched using the [Sites](#) Navigator)
    - Subsites
      - Devices
      - Tests

These are the primary components of a project. Two other components, initialization steps and termination steps, are discussed in the [Reference Manual](#), [Project structure](#), page 6-38. These components are defined contextually in the next subsections.

### Sites

At the macroscopic level, one or more semiconductor dies are built up at a given wafer location. This location is comprised not only of end product dies, but usually has one or more parametric test structures or subsites. KITE refers to such a repeating pattern of dies and test structures as a “site.”

---

**NOTE** In KITE, sites can be switched using the site navigator. All KITE sites have the same subsite, device, and test definitions, and are not repeated in the Project Tree.

---

### Subsites

The terminals of each device on a test structure are connected to a uniformly-spaced series of contact pads. These pads are used to connect the devices to the probes of a prober. Any single wafer location that the prober moves to and contacts is called a “subsite” (sometimes referred to as a test element group (TEG)).

The Model 4200-SCS hardware / KITE software combination was optimized to evaluate test structures, though it can be effectively used to evaluate dies and discrete components. KITE refers to each such test structure (or combination of test devices that are tested as a group) as a subsite.

### Devices

As described in context under [Sites](#), each test structure contains a series of devices to be characterized: Transistors, diodes, resistors, capacitors, and so on. A switch matrix is used to connect the Model 4200-SCS sequentially if the SMUs cannot be connected to all devices simultaneously.

A “device” is also referred to as a test element, because subsites are often referred to as test structures or test element groups (TEGs), which are composed of devices.

### Tests

Once the device is in position, KITE automatically conducts one or more specified tests for each device on the test structure. Each test generates data and, if desired, parametric curves. A test includes the following for each terminal of a device:



- The desired voltage or current forcing functions (stimuli).
- The desired voltage or current measurements.
- The associated data analyses and parameter extractions.

The combination of forcing functions and measurements is referred to as the “test definition.”

There are two classes of tests or test modules in KITE: ITMs and UTMs. Both ITMs and UTMs share common data analysis functions, such as a Microsoft® Excel®-compatible data sheet and a real-time graph tool.

**Key differences between ITMs and UTMs include the following:**

- **Interactive test module (ITM):** An ITM allows the you to define a test interactively using a GUI and no coding.
- **User test module (UTM):** A UTM is defined through “C” programming of its connected KULT created user module, but allows the user to configure key test parameters using a GUI.

Differences between ITMs and UTMs were discussed in more detail under [Understanding KITE](#).

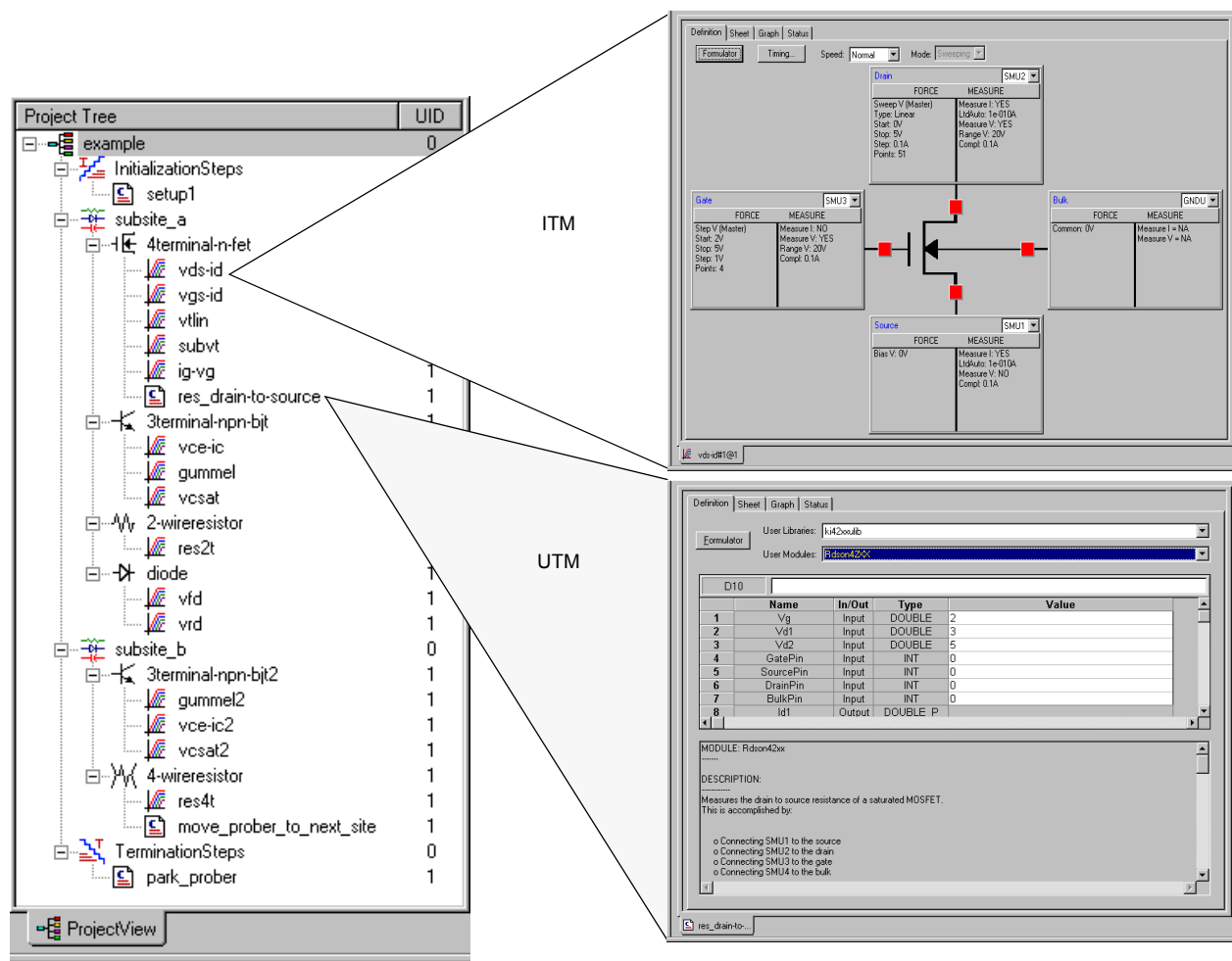
## ITMs versus UTMs

KITE tests and operations are performed through ITMs and UTMs, as shown in [Figure 2-2](#). [Figure 2-3](#) shows the location of the configuration windows of the `vds-id` ITM and the `res_drain-to-source` UTM to their respective locations in the example KITE project plan.

These windows and some associated windows are examined in more detail in the next two subsections, [Defining an ITM](#), and [Defining a UTM](#).



Figure 2-3  
ITMs and UTM in the project navigator



The primary differences between ITMs and UTMs are summarized in [Table 2-1](#).

Table 2-1  
Primary differences between an ITM and a UTM

ITM	UTM
Is always configured using a series of systematic, interactive GUIs, without programming.	Is created and configured by connecting the UTM name to a user module and entering or modifying the input parameter values.
Is flexible. Keithley Instruments provides default ITM configurations for most standard devices and tests; you can perform many of your evaluations with no changes (or very few) to the default parameters. However, you can create a new ITM, or customize any existing ITM, to perform a wide variety of static and dynamic evaluations. You can even create an ITM for a generic “n-terminal” device.	Is task-specific. However, you can modify the source code for a user module that is connected to a UTM and recompile it to create a new user module. Keithley Instruments provides the source code for most of the user modules that are shipped with the Model 4200-SCS. User modules are modified using KULT.



Table 2-1 (continued)

**Primary differences between an ITM and a UTM**

ITM	UTM
Performs exclusively tasks on internal Model 4200-SCS instrumentation. <sup>a</sup>	Performs tasks on internal Model 4200-SCS instrumentation and external instrumentation that is connected to the Model 4200-SCS IEEE-488 bus or the Model 4200-SCS RS-232 port.
Is used exclusively for parametric testing.	May be used to perform almost any test-related task.
Generated data updates the Data worksheet <sup>b</sup> in real time, as the test executes.	Generated data updates the Data worksheet after test execution is complete. Beginning with KTE Interactive v5.0, you have the ability to add function calls to new and existing user modules (UTMs) that provide real-time data and graphing capabilities (see the <a href="#">Reference Manual, Enabling real time plotting for UTMs, page 6-16</a> ).

a. The pulse generator card (Model 4205-PG2) and scope card (Model 4200-SCP2 or 4200-SCP2HR) are not supported by ITMs at this time.

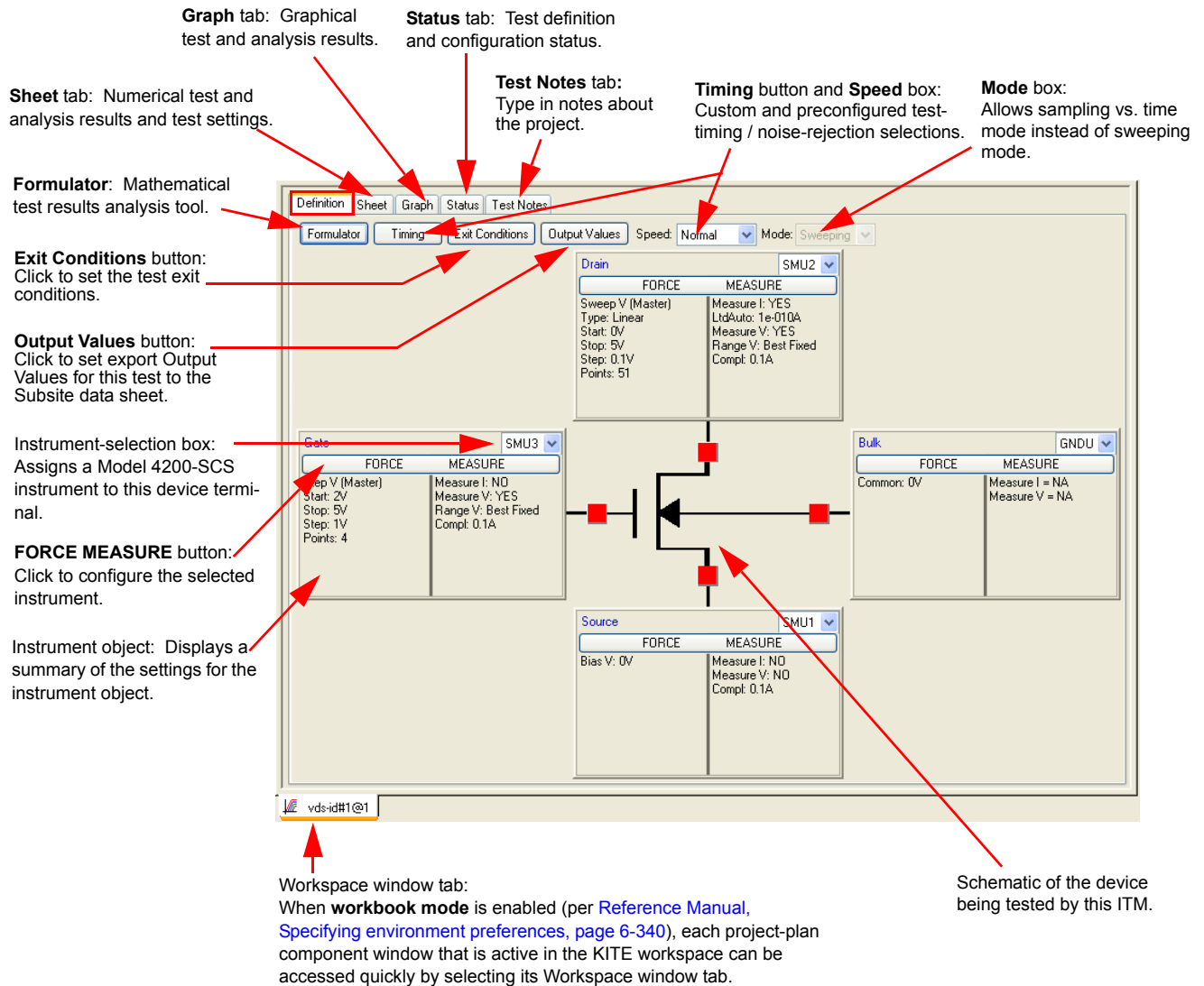
b. Refer to the [Reference Manual, Viewing ITM or UTM results numerically: The Sheet tab Data worksheet, page 6-20](#).

## Defining an ITM

An ITM is defined by the ITM definition tab (displayed by double-clicking the ITM name in the project navigator) illustrates and explains the ITM definition tab ([Figure 2-4](#) defines the `vds-id` ITM, one of the ITMs in the example project plan shown in [Figure 2-2](#) and [Figure 2-3](#)). See [Reference Manual, Specifying environment preferences, page 6-340](#).



Figure 2-4  
ITM definition tab



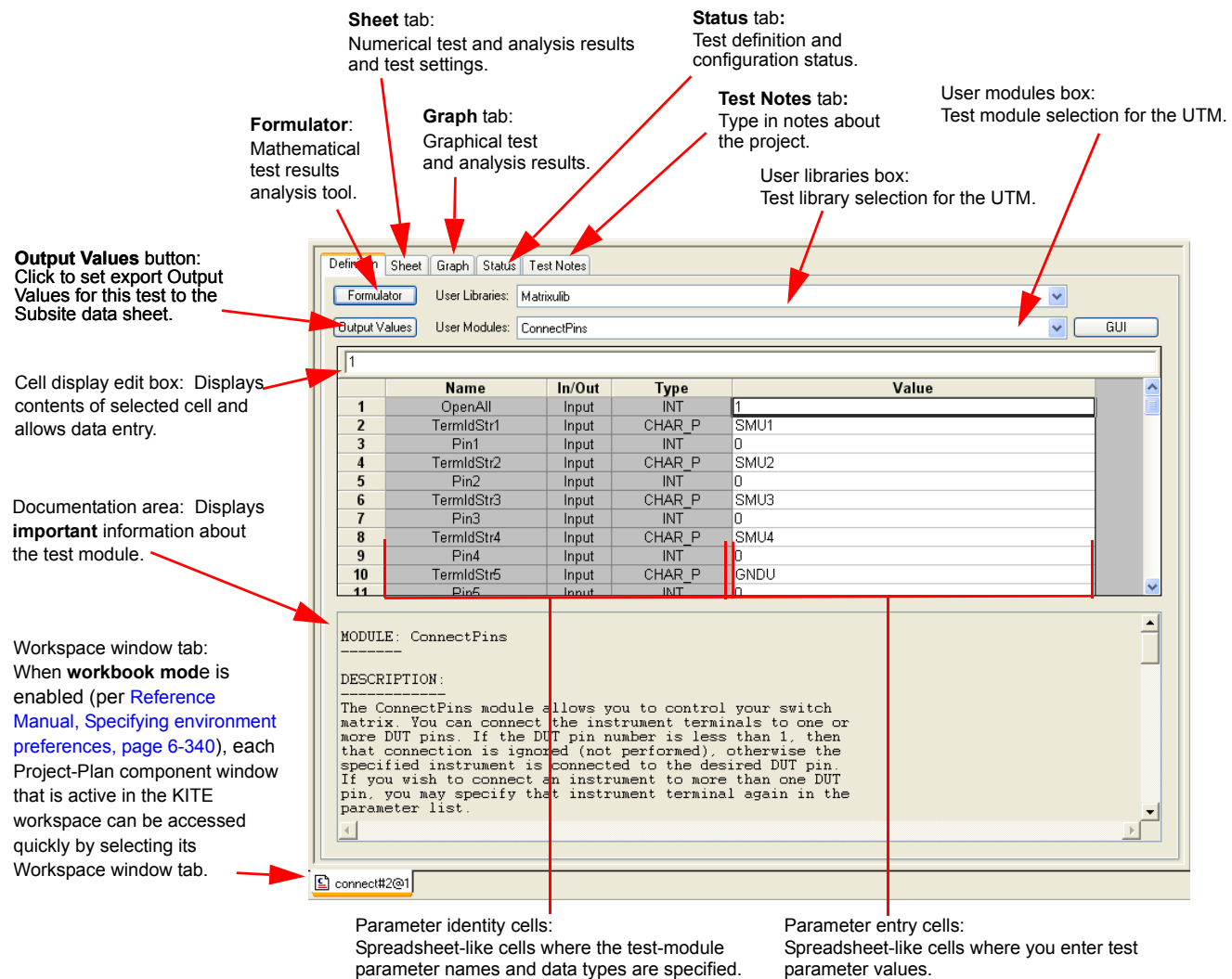
For details about defining and configuring an ITM, refer to the [Reference Manual, Configuring the Project Plan ITMs, page 6-86](#).

## Defining a UTM

A UTM is defined using the UTM definition tab (displayed by double-clicking the UTM name in the project navigator). [Figure 2-5](#) illustrates and explains the UTM definition tab ([Figure 2-5](#) defines the `res_drain-to-source` UTM, one of the UTMs in the example project plan that was shown in [Figure 2-2](#) and [Figure 2-3](#)).



Figure 2-5  
UTM definition tab



A UTM is created and configured by selecting a user library and user module, and then entering parameter values. For details about defining and configuring a UTM, refer to the [Reference Manual, Configuring the UTMs, page 6-145](#).



## How to create your own ITMs

### Understanding the ITM definition tab

To define an ITM, use the ITM definition tab. Display it by double-clicking the ITM name in the project navigator. [Figure 2-4](#) shows and describes the ITM definition tab, in this case for the `vds-id` ITM, which is part of the example project (shown in [Figure 2-2](#) and [Figure 2-3](#)).

**An ITM definition tab defines the ITM as follows:**

- Schematically displays the type of device to be tested by the ITM (FET, BJT, capacitor, and so on).
- Next to each terminal of the device, displays an “instrument object,” which acts as follows:
  - Identifies the terminal (for example, gate, drain, source, collector, anode, and so on).
  - Identifies and allows assignment / reassignment of the terminal to match the SMU, GNDU, or open circuit that is physically connected to the terminal during the test.
  - Displays the present forcing-function and measurement options for the terminal.
  - Identifies and allows assignment and configuration / reconfiguration of SMU forcing function and measuring options. A single-click of the force measure button displays the Forcing Functions / Measure Options window for the terminal.
- Provides access to the Formulator, which allows in-test and post-test data computations.
- Allows setting of preconfigured speed or custom timing parameters for the ITM.
- Allows exit conditions to be set if the source goes into compliance.
- Allows you to select the measured readings (output values) that you want exported to the Subsite Data sheet.
- Displays the present test mode:
  - Sweeping
  - Sampling

### Understanding the ITM forcing functions

[Table 2-2](#) summarizes the available ITM “forcing functions,” which tells the Model 4200-SCS how to apply static or dynamic voltage or current conditions to device terminals.

Table 2-2  
**SMU Forcing function summary**

General type	Name	Description and graphical illustrations
Static	Open	Maintains a zero-current state at the terminal, subject to the maximum voltage compliance of the connected SMU.
	Common	Maintains a zero-voltage state at the terminal, subject to the maximum current compliance of the connected SMU.
	Current bias	Maintains a selected constant-current state at the terminal, subject to the user-specified voltage compliance for the connected SMU.
	Voltage bias	Maintains a selected constant-voltage state at the terminal, subject to a user-specified current compliance of the connected SMU.



Table 2-2 (continued)  
SMU Forcing function summary

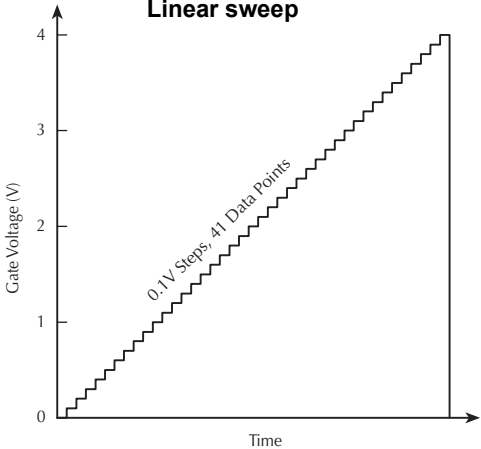
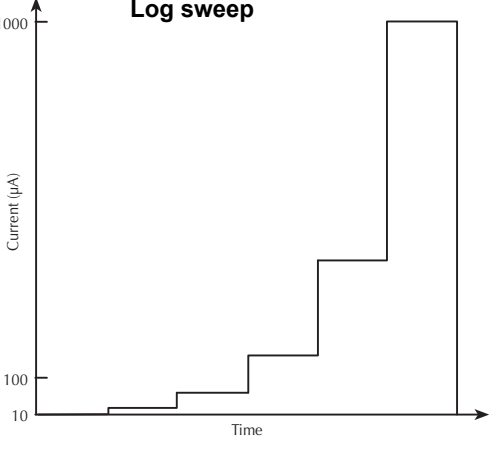
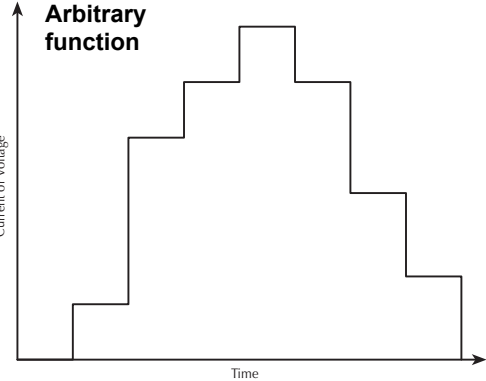
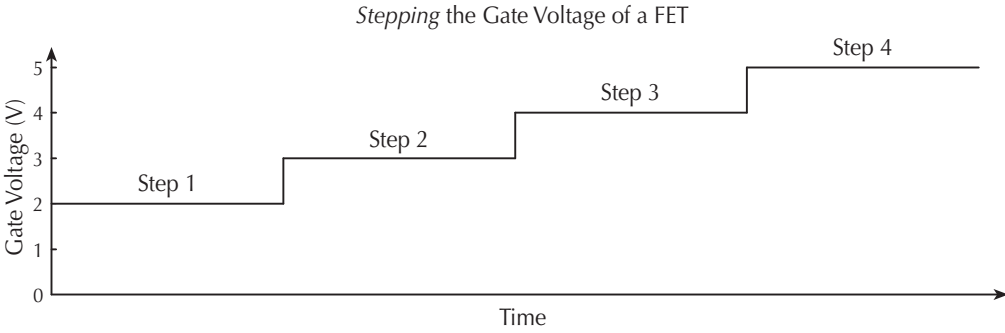
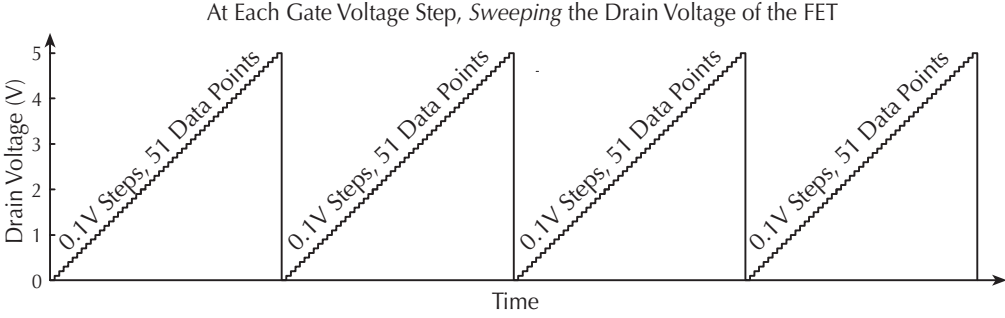
General type	Name	Description and graphical illustrations
Sweep	Current sweep	<p>Increments a series of current values or voltage values at a rate that is determined by the timing and speed settings in the ITM definition tab. Generates parametric curve data that is recorded in the Sheet tab Data worksheet for the ITM and can be plotted in the ITM Graph tab.</p> <div><div><p><b>Linear sweep</b></p></div><div><p><b>Log sweep</b></p></div></div>
	Voltage sweep	
List sweep	Current List Sweep	<p>Steps through a list of user-specified current values or voltage values, at a rate that is determined by the timing and speed settings in the ITM definition tab. Generates parametric data that is recorded in the ITM Sheet tab Data worksheet and can be plotted in the ITM Graph tab, if appropriate.</p> <div><p><b>Arbitrary function</b></p></div>
	Voltage List Sweep	



Table 2-2 (continued)  
SMU Forcing function summary

General type	Name	Description and graphical illustrations
Step	Current step	<p>Increments a current or voltage to two or more levels, each of which is held constant during the progress of a current sweep, a voltage sweep, a current list sweep, or a voltage list sweep at another terminal. For each current step or voltage step level, parametric curve data is recorded in the ITM Sheet tab Data worksheet. The combined data can be plotted in the ITM Graph tab, resulting in a series (family) of curves.</p> <div><p><i>Stepping the Gate Voltage of a FET</i></p></div> <div><p><i>At Each Gate Voltage Step, Sweeping the Drain Voltage of the FET</i></p></div>
	Voltage step	

For forcing-function details, refer to the [Reference Manual, The ForcingFunctionName function parameters area, page 6-102](#).

### Understanding dual sweep

A SMU that is configured to perform a linear or log sweep, can also be set to perform a dual sweep. With dual sweep enabled, the SMU will perform two sweeps. The first sweep steps from the start level to the stop level. The SMU then continues with the second sweep, which steps from the stop level back to the start level. With dual sweep disabled, the SMU performs a single sweep that steps from start to stop.

A dual sweep for a slave SMU is typically used with a master SMU that is also set to perform a dual sweep. The master SMU does not have to be set for dual sweep in order to use dual sweep for a slave SMU. In this case, setting the master SMU's sweep points to an even number will ensure that the slave's dual sweep is symmetrical. Setting the master SMU count to an odd number, will cause the slave SMU to repeat the last sweep point.

**NOTE**

*The slave SMUs will not automatically set for dual sweep when dual sweep is enabled for the master SMU. Dual sweep must be enabled individually for each SMU.*

To compare a single sweep to a dual sweep, refer to the [Reference manual, Figure 6-135](#).



## Understanding pulse mode

To avoid device overheating in some tests, voltages or currents can be applied to a device only for brief periods at widely spaced intervals. For sweep (linear, log, and list) and bias forcing functions, an SMU can be set to provide pulse output.

**With pulse mode enabled, the following pulse parameters can be set:**

- On Time
- Off Time
- Base Voltage (or Base Current)

The base is the level the SMU goes to between sweep points. Pulse “on” and “off” times determine pulse period and pulse width as follows:

- Pulse period = On Time + Off Time + cumulative measure time (if set to measure)
- Pulse width = On Time

Pulse mode can be selected only when source and measure ranges are fixed. Pulse mode is disabled if the source or measure range is set to **AUTO**.

Pulse “on” and “off” times can be set from 5 ms to 10 s. The base voltage (or current) that can be set is dependent upon the present source range.

An example pulse output for the voltage bias forcing function is shown in [Figure 2-6](#). Pulse output goes to the specified pulse level during the pulse “on” time. If the SMU is set to measure, the measurement will occur after the “on” time expires and before the transition to the “off” time level. This effectively increases the “on” time by the amount of time required to make the measurement. Minimize this extra time by choosing “custom” in the timing tab and setting delay and filter factor to 0, and A / D Integration factor to 0.01. This is the fastest (but least accurate) measurement timing scheme. If not set to measure, the output will transition from “on” to “off.”

During pulse “off” time, the pulse output returns to the specified base voltage level. After the “off” time expires, the output returns to 0 V.

For a sweep forcing function, pulse output steps to the sweep step levels during the pulse “on” times. During the “off” times, pulse output returns to the specified base voltage (or base current) level. If set to measure, the measurement will occur after each “on” time period expires and before the pulse transitions to the “off” time level.

The voltage sweep in [Figure 2-7](#) is a single sweep. If dual sweep is enabled, the test will continue by going back to the stop level and then step down to the start level. For details, see [Understanding dual sweep, on page 2-15](#).

Figure 2-6

**Pulse mode example: Voltage bias; 2V level, 1V base**

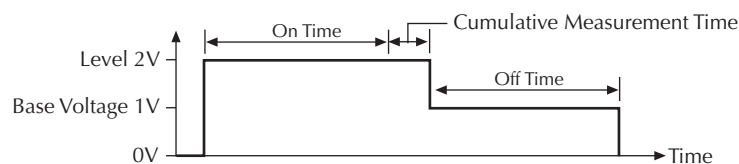
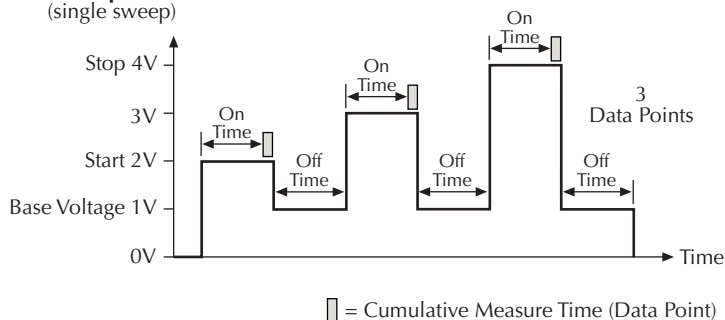


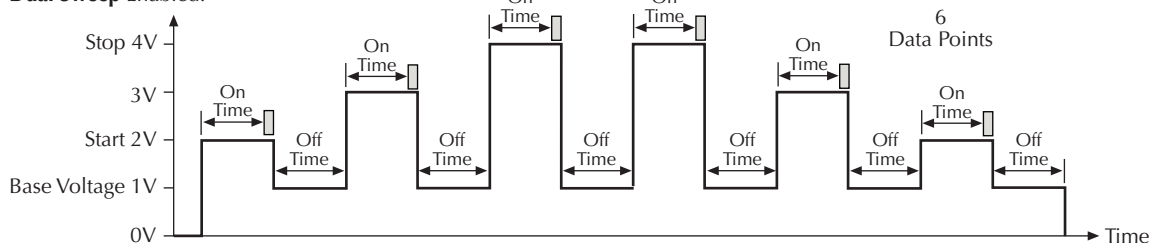


Figure 2-7  
Pulse mode examples: Single and dual sweep

Dual Sweep Disabled:  
(single sweep)



Dual Sweep Enabled:



## How to use the definition tab to configure ITM parameters

This topic describes parameter configuration of a library ITM, in which connections and test modes are preconfigured. For discussions of general ITM configuration, including creation of new ITMs or customization of existing ITMs, refer to the [Reference Manual, Configuring the Project Plan ITMs, page 6-86](#).

After inserting library ITMs into your project, configure the setup for each ITM as described in the subsections that follow in the order in which they appear.

### Match the physical and virtual connections:

1. In the project navigator double-click the ITM that you wish to configure. The definition tab of the ITM window opens by default (see [Figure 2-4](#)).
2. In the definition tab, review the virtual connections for each device terminal, as listed in the "instrument object" for that terminal (see [Figure 2-4](#)).
3. Ensure that the physical device connections match the virtual ( definition tab) device connections. If necessary, shut down the instrumentation and correct the physical connections.

---

**CAUTION** Physical device-terminal connections must accurately match virtual connections to avoid bad test results and potential device damage.

---

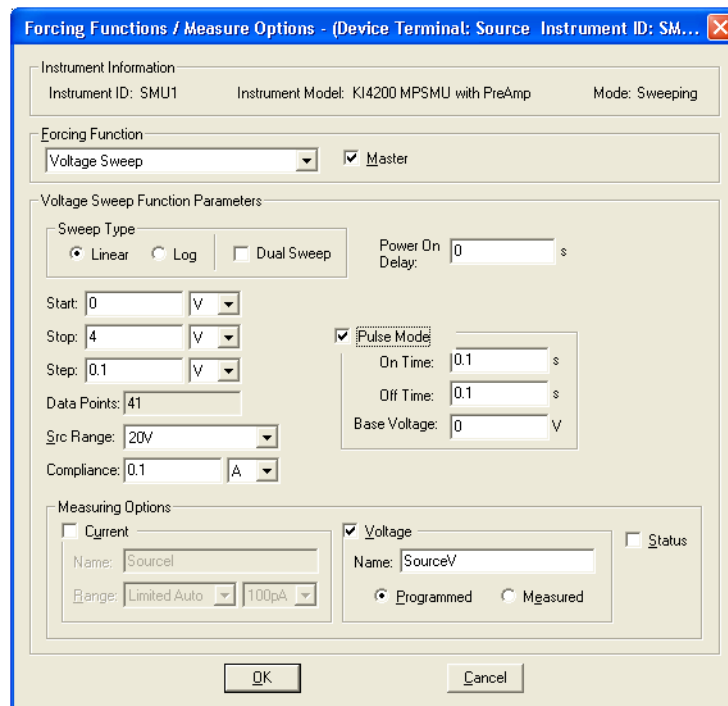
### Configuring forcing functions for each device terminal

With the definition tab for the ITM open, do the following for each device terminal:

1. On the instrument object for the terminal ([Figure 2-4](#)), click the **FORCE MEASURE** button. The corresponding Forcing Function / Measure Options window appears. The Forcing Functions / Measure Options window in [Figure 2-8](#) illustrates typical window features.

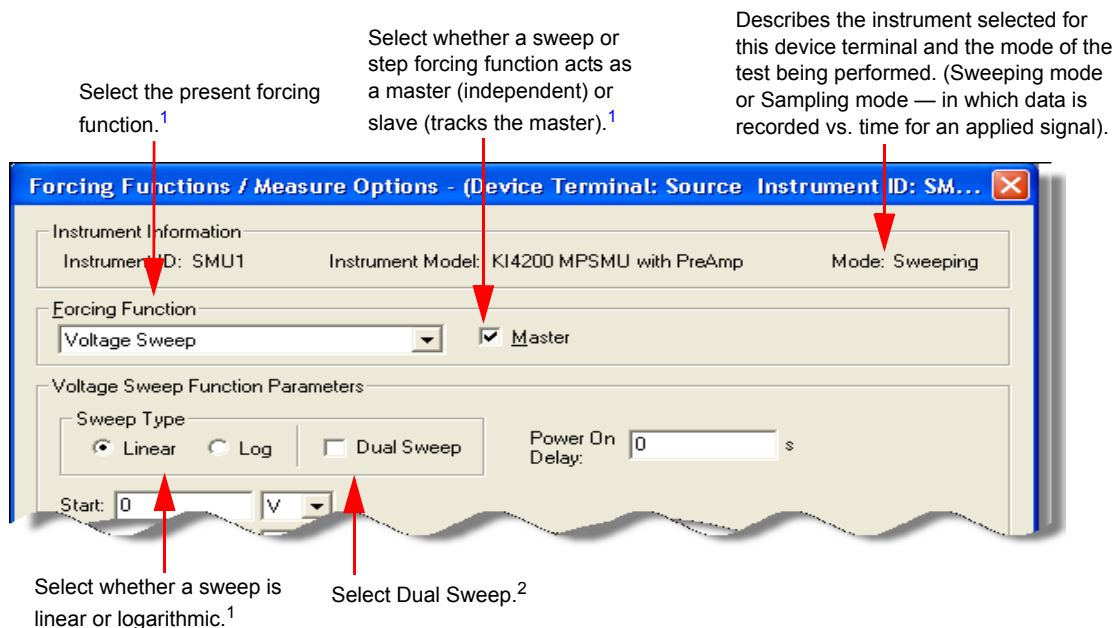


Figure 2-8  
**Typical SMU forcing functions / measure options window for an existing library ITM**



Figures 2-9, 2-10, and 2-11 illustrate basic options and functions for each part of the Forcing Functions / Measure Options user interface.

Figure 2-9  
**Instrument information, forcing functions, and voltage sweep function parameters**



<sup>1</sup> Do not change these parameters, unless you want to customize the ITM — which is beyond the scope of this User's manual.

<sup>2</sup> Dual sweep — with dual sweep enabled, the SMU will sweep from start to stop, and then sweep from stop back to start. When disabled, the SMU will sweep from start to stop. For details, see [Understanding dual sweep on page 2-15](#).



Figure 2-10

**Linear sweep or step, SMU range, Pulse Mode, and SMU current compliance**

For a **Linear sweep** or **step** **Start** is the voltage / current at the start of the sweep, **Stop** is the voltage / current at the end of the sweep, and **Step** is the voltage / current change between steps.<sup>3</sup> **Data Points** — calculated automatically from the **Start**, **Stop**, and **Step** values — is disabled. For a **Log sweep**, you specify the **Data Points** value. The **Step** value — calculated automatically — is disabled.

Select and configure **Pulse Mode**.<sup>4</sup>

Select the SMU range to be used when forcing the specified voltage or current. Select dynamically optimized range (Auto), single best range for entire sweep (Best Fixed), or manually specified numerical range.

The screenshot shows a software window for configuring a sweep. At the top, there are tabs for 'Linear', 'Log', and 'Dual Sweep'. Below these are input fields for 'Start' (0 V), 'Stop' (4 V), 'Step' (0.1 V), and 'Data Points' (41). A 'Src Range' dropdown is set to '20V'. Below that is a 'Compliance' field set to '0.1 A'. A 'Pulse Mode' checkbox is checked, and its sub-options are visible: 'On Time' (0.1 s), 'Off Time' (0.1 s), and 'Base Voltage' (0 V). A 'Measuring Options' section is partially visible at the bottom. Red arrows point to the 'Src Range' dropdown and the 'Pulse Mode' checkbox.

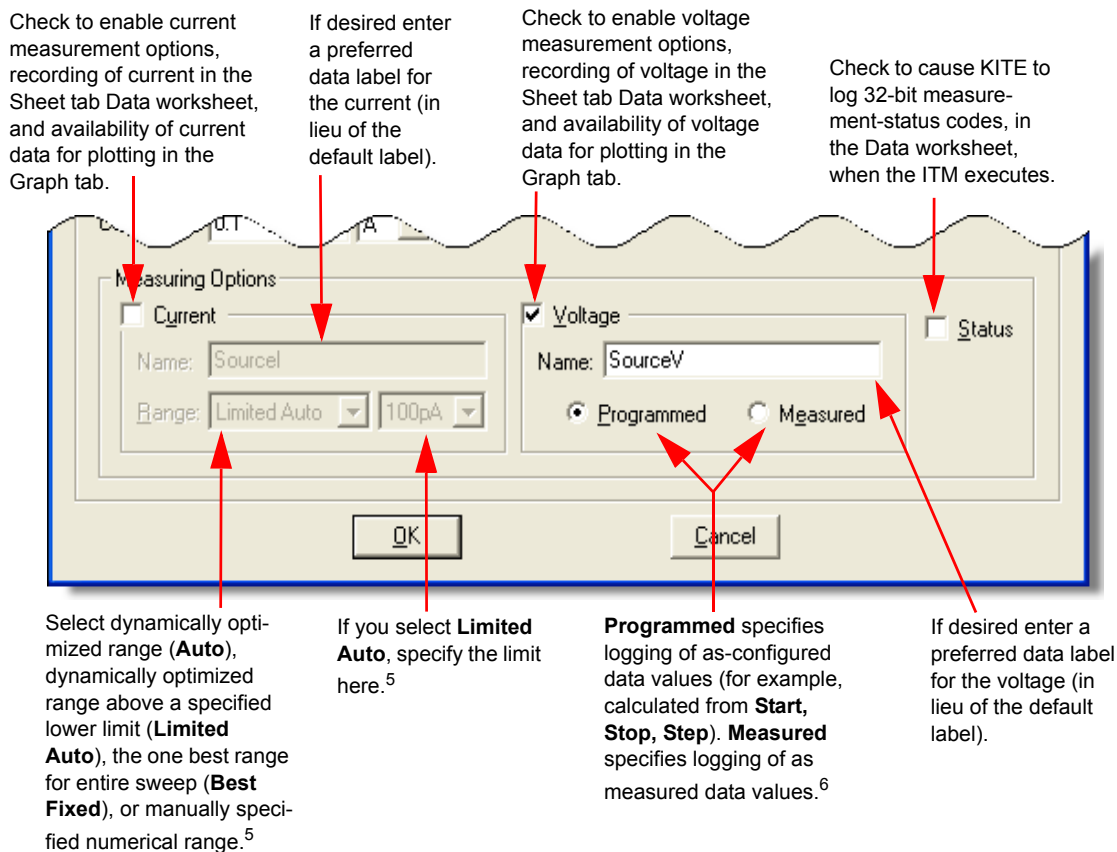
Select the SMU current compliance for a voltage sweep or the voltage compliance for a current sweep.

<sup>3</sup> In a list sweep configuration window, you enter a list of discrete voltages or currents, instead of start, stop, and step values. In a current bias or voltage bias configuration window, you enter a fixed level value, instead of start, stop, and step values.

<sup>4</sup> Pulse mode — Select **Pulse Mode** to provide pulse output for sweep (linear, log and list) and bias forcing functions. For details, see [Understanding pulse mode on page 2-16](#).



Figure 2-11  
Measuring Options



<sup>5</sup> If sweeping / stepping voltages If sweeping / stepping currents, replaced with programmed and measured buttons, as in right panel.

<sup>6</sup> If sweeping / stepping voltages If sweeping / stepping currents, buttons replaced with range settings similar to those in left panel.

2. Referring to the setting explanations in [Figure 2-8](#), [2-9](#), [2-10](#), and [2-11](#) above, configure the following types of parameters for the device terminal's forcing function, as appropriate:
  - Linear or log for a sweep-type forcing function.
  - Dual sweep – sweep from start to stop, and then sweep from stop to start.
  - The current or voltage value(s) to be forced:
    - Level for a static forcing function
    - Start, stop, and step for a step or linear sweep forcing function
    - Start, stop, and data points for a log sweep forcing function
    - Data points value and list of amperes or volts values for a list sweep
      - The default or desired Src Range and compliance
      - The default or desired measuring options
3. Click **OK**. The configuration for this device terminal takes effect and the Forcing Functions / Measure Options window closes.
4. If using pulse mode, see [Configuring pulse mode](#).
5. Repeat Steps 1 through 3 for the remaining device terminals.

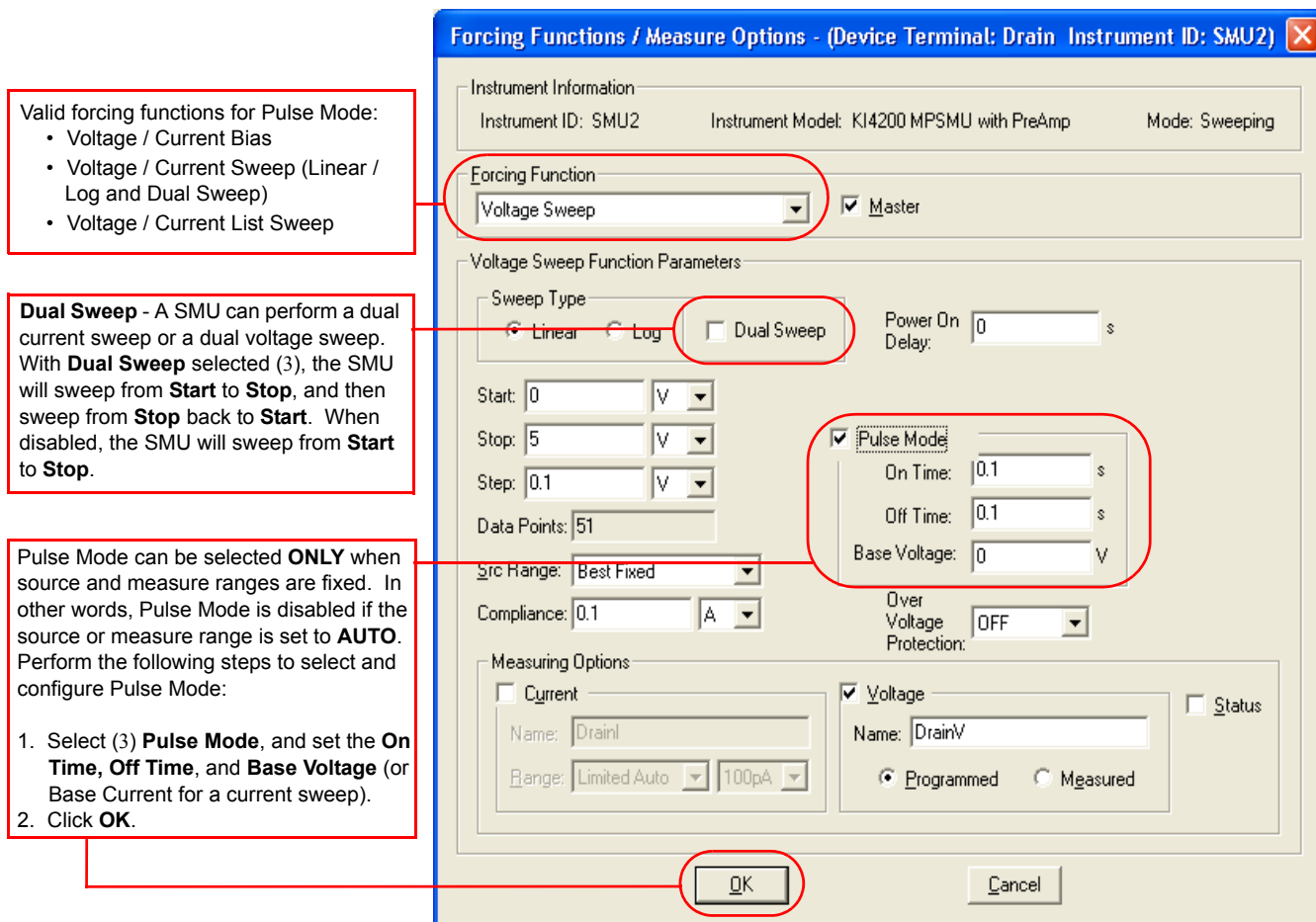
## Configuring pulse mode

With a valid forcing function selected, a SMU can be configured to provide pulse output.



Figure 2-12 explains how to use the pulse mode. The settings in the Forcing Functions / Measure Options window shown in Figure 2-12 configures a voltage sweep. The pulse mode controls for the other valid forcing functions are similar.

Figure 2-12  
Pulse mode configuration (voltage sweep)



## Basic test execution

### Project navigator check boxes

As shown in Figure 2-13, each component of the project plan has a check box. A check mark in a box indicates that the test or plan is enabled. The absence of a check mark indicates that the test or plan is disabled. Clicking a check box either inserts a check mark to enable or removes a check mark to disable. Only enabled (check marked) tests or plans can be run.

**There is interaction between the project navigator check boxes and is explained by the following actions:**

#### Tests (ITMs and UTMs)

- A check mark can be inserted or removed for any test.
- Inserting a check mark for a test also inserts a check mark for its device plan, its subsite plan, and the project plan.



## Device plan

- Clearing a check mark for a device plan also clears the check marks for all of its subordinate tests.
- Inserting a check mark for a device plan also inserts check marks for all of its tests.
- Removing the check marks for all the tests in the device plan, also removes the check mark for the device plan.

## Subsite plan

- Removing a check mark for a subsite plan also removes the check marks for all of its device plans and tests.
- Inserting a check mark for a subsite plan also inserts check marks for all of its device plans and tests.
- Removing the check marks for all the tests in the subsite plan, also removes the check mark for the subsite plan.

## Initialization and termination steps

- Removing a check mark for initialization or termination steps also removes the check marks for all of its UTMs.
- Inserting a check mark for initialization or termination steps also inserts check marks for all of its UTMs.
- Removing the check marks for all the UTMs in the initialization or termination steps, also removes the check mark for the initialization or termination steps.

## Project plan

- Removing a check mark for a project plan also removes the check marks for all of its plans and tests.
- Inserting a check mark for a project plan also inserts check marks for all of its plans and tests.
- Removing the check marks for all the tests in the project, also removes the check mark for the project plan.


---

**NOTE** *Reference manual, Figure 6-16 shows an example of project plan structure that shows a mix of enabled and disabled tests.*

---

## Executing an individual test

### Selecting a test

An enabled (check marked) ITM or UTM is selected by clicking the test in the project navigator (see [Figure 2-13](#)). The Run Test / Plan button  turns green to indicate that the test is enabled and ready to be run. Also, the selected-test name is displayed in the Test / Plan Indicator box located above the project navigator.

The test can also be selected by double-clicking it in the project navigator. The double-click action places the appropriate ITM or UTM window in the KITE workspace. The ITM and UTM definition tabs show the test configurations.\*<sup>1</sup> See [Figure 2-4](#) and [Figure 2-5](#).

---

1. For details about using the ITM and UTM definition tabs, see [Reference Manual, Configuring the Project Plan ITMs, page 6-86](#), and [Configuring the UTMs](#).




---

**NOTE** Before executing a test for which data must be labeled with a specific site number, refer to the [Reference Manual, Assigning a site-number label to individual test and test-sequence data](#), page 6-28.

---

## Running the test

### To run a selected test:

1. Click the green **Run Test / Plan** button .
2. Select **Run** in the Run menu, or press the **F6** keyboard key.

While the test is running, test data is placed the data sheet. In ITMs data is placed in the data sheet real time as data is being acquired. In UTMs by default, data is not placed in the data sheet until after the test has finished running.

### In the message area of the KITE window, time stamps indicate:

- Start time
- Stop time
- Total execution time

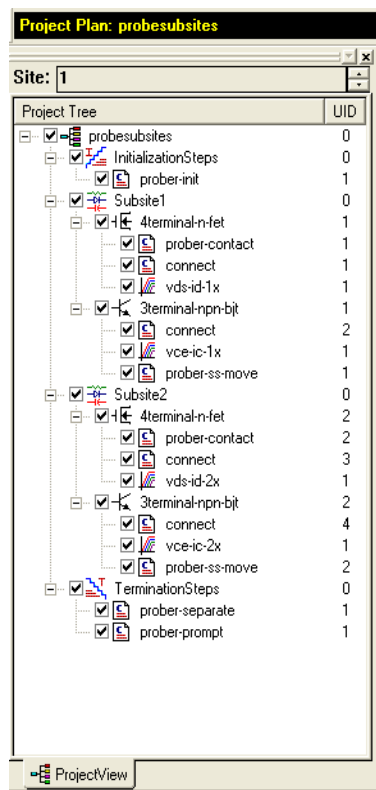
---

**NOTE** You can also start a test by pressing the **F6** keyboard key. You can abort a test by clicking the red **Abort Test / Plan** toolbar button, by selecting **Abort** in the Run menu, or by pressing the **PAUSE / BREAK** keyboard key.

---

For detailed information about running individual tests, see [Run execution of individual tests and test sequences](#) in Section 6 of the Reference Manual..

Figure 2-13  
Example project plan





## How to display and manage test results

### Data file management

#### Using file and test-result directories

KITE application files and test results are stored on the Model 4200-SCS hard drive by default. However, KITE projects and various other KITE application files can be stored and utilized on any available hard drive, except CD, CD-R, CD-RW drives and write-protected drives or directories.

---

**NOTE** *Storing application files on a congested network drive can degrade overall test sequence performance. The best Model 4200-SCS system performance is obtained by storing all KITE Interactive application files on the Model 4200-SCS internal hard drive. There, sweep and sampling measurement speed is not affected perceptibly by network traffic or any other embedded PC system activities.*

---

This subsection provides useful information regarding the default organization of KITE application and data files. Additional information regarding KITE Interactive file management and system administration can be found in the [Reference Manual, Managing multiple users and systems, page 10-6](#).

---

**CAUTION** **Never directly edit KITE application files using a test editing program (with one exception, noted in [How to create and add a new device on page 2-27](#)) because unexpected results and application crashes can occur.**

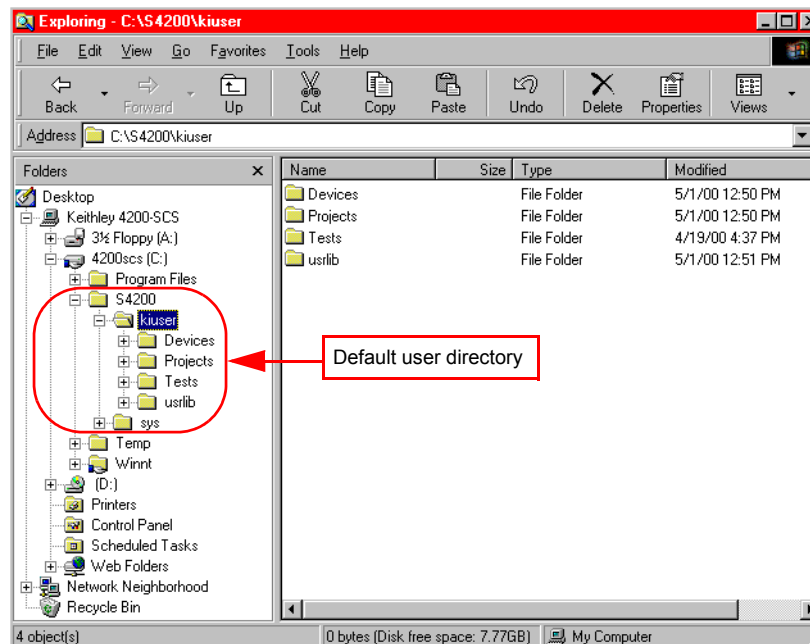
---

#### Default user director: C:\S4200\kiuser

By default, all of the sample projects and standard libraries included with KITE Interactive are stored in the C:\S4200\kiuser directory, as illustrated in [Figure 2-14](#). This folder, is referred to as the "default user directory."



Figure 2-14  
Default user directory



Note however, that KITE projects, device libraries, and test libraries can be stored and shared on any accessible disk drive, including a network drive (except CD, CD-R, CD-RW drives and write-protected drives or directories).

The default user directory contains several subdirectories. Each of these subdirectories is discussed below under a separate heading.

### Devices subdirectory

By default, the `Devices` subdirectory contains the KITE Device Library that is provided with each version of KITE Interactive. Also by default, you can access this Device Library when operating KITE. You can copy devices from this library to their project(s) or submit devices from their project(s) to this library.

---

**NOTE** For more information about submitting devices to libraries, refer to the [Reference Manual, Submitting devices, ITMs, and UTMs to libraries](#), page 6-151.

---

### Understanding device libraries

A Device Library is comprised of devices stored in folders that are organized by device category. To create a new device category, create a new folder in the `C:\S4200\kiuser\Devices` directory.\*<sup>2</sup>

**To provide project access to additional device libraries, or to change the KITE Device Library that appears by default, use the KITE Options window:**

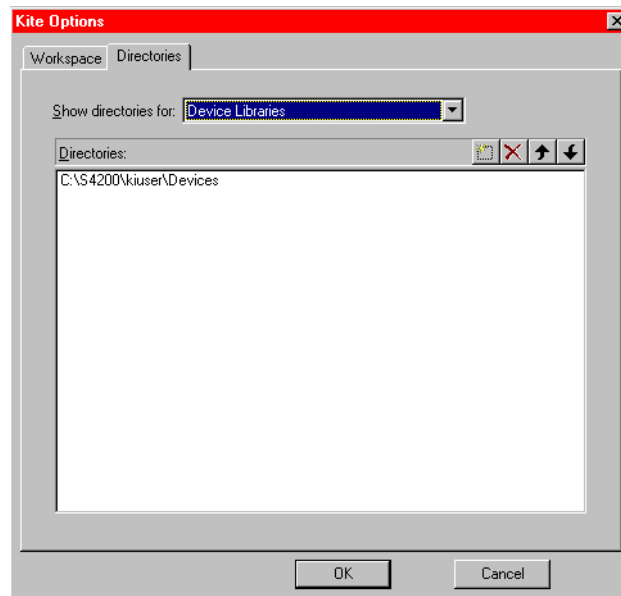
- Select **Options** in the **Tools** menu.
- On the **Directories** tab of the KITE Options window that appears, choose **Device Libraries** in the **Show Directories for:** box. See [Figure 2-15](#).

---

2. The `C:\S4200\kiuser\Devices` is the factory-default `Devices` directory. You can also create a new folder in another `Devices` directory, (for example, `C:\S4200\YourName\Devices`).



Figure 2-15  
Device library access selection



---

**NOTE** For more information about Device Library access selection, refer to the [Reference Manual, Customizing directory options, page 6-342](#).

---

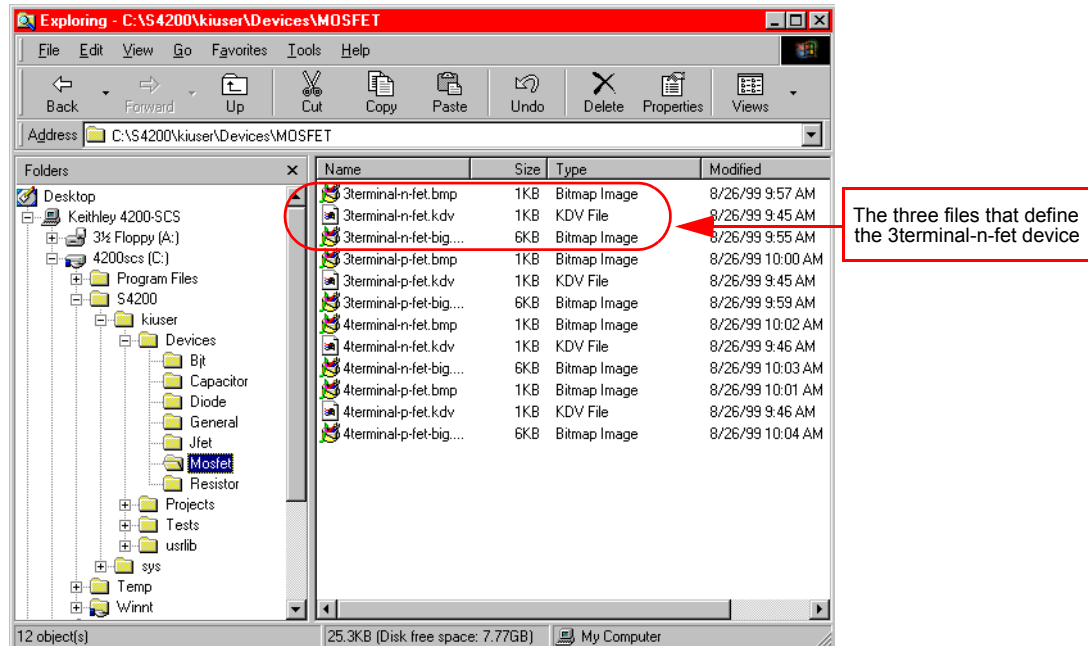
**Each device stored in a Device Library contains the following three types of files:**

- A Keithley device (.kdv) file that follows the Microsoft® Windows® .ini file format.
- A small bitmap; 16 x 16 dpi (.bmp) file for the device icon that is displayed in the project navigator.
- A large bitmap; 120x100 dpi (.bmp) file for the device graphic that is displayed on the definition tab of each ITM that tests the device.



Figure 2-16 shows the files in the `Mosfet` Device Library folder. Note that a `.kdv` file and two bitmap files are listed for each device.

Figure 2-16  
Device files



## How to create and add a new device

To create a new device, you must create three files:

- The Keithley device (`.kdv`) file.
- The small bitmap; 16 x 16 dpi (`.bmp`) file for the project navigator device icon.
- The large bitmap; 120x100 dpi (`.bmp`) file for the ITM definition tab device graphic.

The `.kdv` file can be created or modified using text editing software, such as Microsoft® Notepad®. The `.bmp` files can be created or modified using bitmap editing software, such as Microsoft® Paint®.

The following procedure illustrates how to add a new device, named `new-mosfet`, to the default Device Library:

1. In the `C:\S4200\kiuser\Devices\MOSFET` directory, locate the following three files, which define the existing library device called `3terminal-n-fet`:
  - `3terminal-n-fet.kdv`.
  - `3terminal-n-fet.bmp` for the project navigator device icon.
  - `3terminal-n-fet-big.bmp` for the ITM definition tab device graphic.
2. Copy `3terminal-n-fet.bmp` to a new file called `new-mosfet.bmp`.
3. If required, modify `new-mosfet.bmp` using Microsoft® Paint®.
4. Copy `3terminal-n-fet-big.bmp` file to a new file called `new-mosfet-big.bmp`.
5. If required, modify the new bitmap using Microsoft® Paint®.
6. Copy the `3terminal-n-fet.kdv` file to a new file called `new-mosfet.kdv`.



7. Edit the `new-mosfet.kdv` file with **Notepad®** by replacing all occurrences of “`3terminal-n-fet`” with “`new-mosfet`”.

The edited `new-mosfet.kdv` file should appear as shown in [Figure 2-17](#). All six lines are required.

Figure 2-17  
Contents of the Keithley Device file `new-mosfet.kdv`



[Table 2-3](#) describes each line of the `new-mosfet.kdv` file that appears in [Figure 2-17](#).

Table 2-3  
Line-item descriptions for a `.kdv` file

Line-item example	Description
[Bitmaps]	Location of bitmap file information.
Small=new-mosfet.bmp	Name of file to use when displaying the device in the KITE project navigator.
Big=new-mosfet-big.bmp	Name of file to use when displaying the device on a KITE ITM definition tab.
[Terminals]	Location of terminal-label information.
Number=3	Number of device terminals (8 maximum).
Orientation=;N:Drain;S:Source;W:Gate	Geographic, or screen, location and name of each device terminal. Valid locations are as follows: <ul style="list-style-type: none"><li>• N North or top</li><li>• NE Northeast or upper right</li><li>• E East or right</li><li>• SE Southeast or lower right</li><li>• S South or bottom</li><li>• SW Southwest or lower left</li><li>• W West or left</li><li>• NW Northwest or upper left</li></ul>

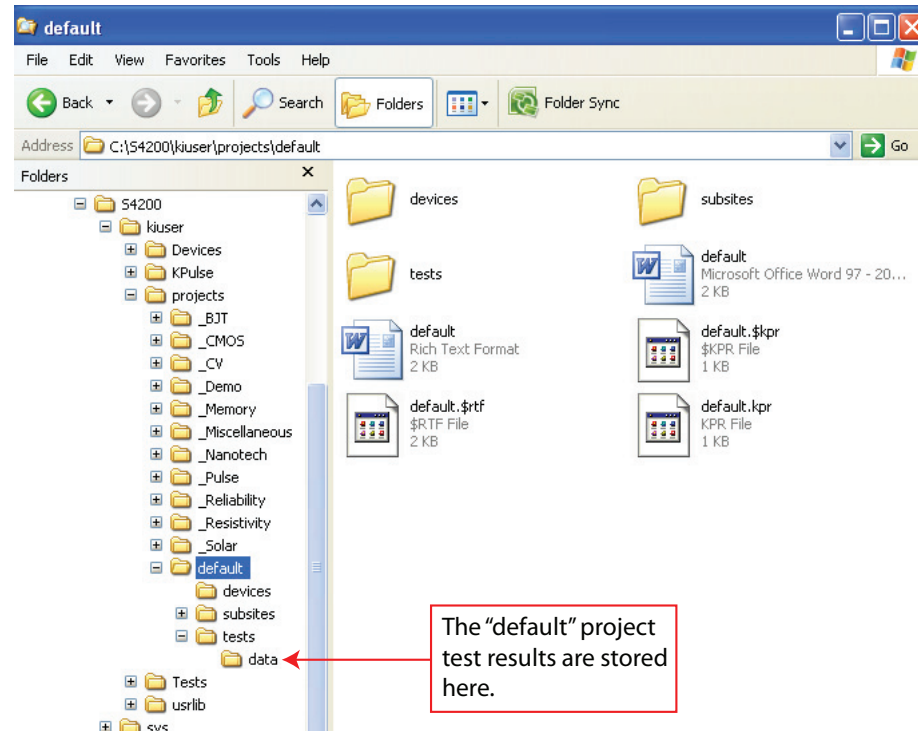
Projects subdirectory

The `Projects` subdirectory contains the default KITE project library that is provided with each version of KTE Interactive. By default, you can store KITE projects in this directory. However, KITE projects can be stored in any location, using the KITE **File > Save Project As** menu.

Projects are comprised of multiple files stored in a predefined directory structure. All of the project components are stored in a project folder. [Figure 2-18](#) shows the folders of KITE projects that are included with KTE Interactive. The expanded `default` project folder shows the pre-defined project file structure.



Figure 2-18  
KITE project folders



For each project, test results files (.xls worksheet and .kgs graph) are stored in a project specific data folder, as illustrated in [Figure 2-18](#).

---

**NOTE** Refer to the [Reference Manual, Test data, page 6-29](#) for more information about results files and KITE file-naming conventions.

---

Projects can be moved from one location to another as long as the entire project folder (with all of its contents) are relocated.

### Tests subdirectory

By default, the Tests subdirectory contains the KITE test library that is provided with each version of KITE Interactive. Also by default, you can access this test library when operating KITE. You can copy tests from this library to their projects or submit tests from projects to this library.

---

**NOTE** For more information about submitting tests to libraries, refer to the [Reference Manual, Submitting devices, ITMs, and UTMs to libraries, page 6-151](#).

---

This test library is comprised of tests that are stored in folders organized by device category. To create a new test category, create a new folder in the C:\S4200\kiuser\Tests directory.\*<sup>3</sup>

**To provide project access to additional test libraries in other directories or to change the KITE test library that appears by default, use the KITE Options window:**

1. Select **Options** in the **Tools** menu.

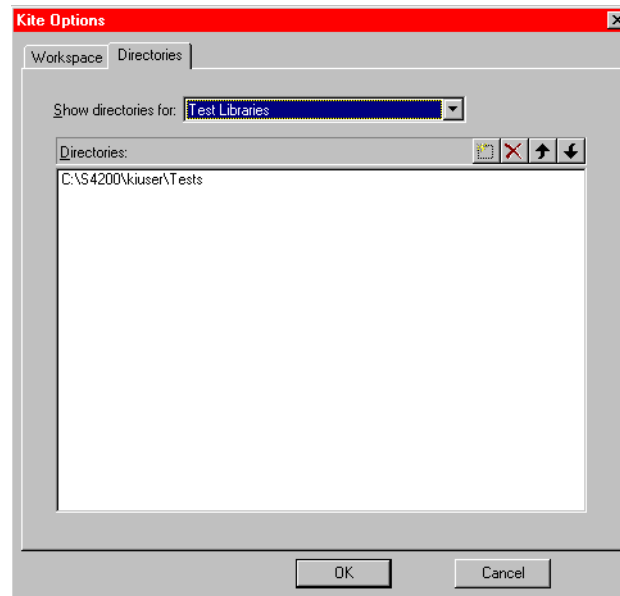
---

3. The C:\S4200\kiuser\Tests is the factory default Tests directory. You can create a new folder in another Tests directory, such as C:\S4200\YourName\Tests.



2. On the **Directories** tab of the KITE Options window that appears, choose **Test Libraries** in the **Show Directories for:** Combo box. See [Figure 2-19](#).

Figure 2-19

**Test library access selection**

---

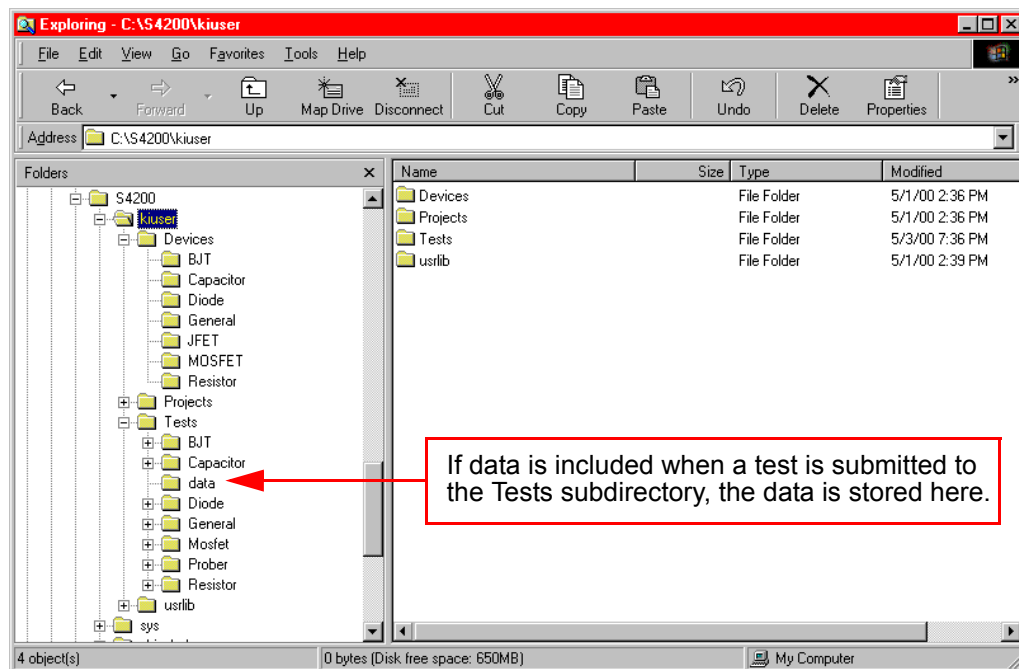
**NOTE** For more information about test-library access selection, refer to the [Reference Manual, Customizing directory options, page 6-342](#).

---

Tests can be submitted to a library with or without including measurement data. By default, data is included when a test is submitted. Test results files (.xls data and .kgs graph files) are stored in the test library data folder, as shown in [Figure 2-20](#).



Figure 2-20  
Test library results folder



### Usrlib subdirectory

By default, the `usrlib` subdirectory contains the KULT user libraries that are provided with each version of KTE Interactive. Also by default, you can access all Model 4200-SCS user libraries when operating KITE and KULT. For more information about this directory, refer to the [Reference Manual, Managing user libraries, page 8-2](#).

### System directory: C:\S4200\sys

All binary and executable files that KTE Interactive needs to control the Model 4200-SCS are stored in the `sys` folder (directory).

---

**NOTE** *The files stored in the `sys` folder (directory) must not be modified, not even by system administrators. This folder must reside on the Model 4200-SCS hard drive.*

---

## How to manage numeric test results in Sheet tab

### Displaying and analyzing data using Sheet tab

The Sheet tab of an ITM or UTM window is used to record and manipulate numerical test data and settings. There is a Sheet tab corresponding to every ITM / UTM for each site. All data in the worksheets of the Sheet tab is exportable in Microsoft® Excel® format.

**A Sheet tab is the same as a Microsoft® Excel®-compatible workbook that always contains at least the following three worksheets:**

- **Data worksheet:** The Data worksheet of the Sheet tab records all of the numerical test data that is generated every time you execute an ITM or a UTM at a given site. The Sheet tab Data worksheet also records data generated by the formulator.



- **Calc worksheet:** The Sheet tab Calc worksheet provides a spreadsheet for local data analysis. If there are multiple same-named instances of an ITM or UTM in a project plan, the Calc worksheet equations are unique for each instance.
- **Settings worksheet:** The Sheet tab settings worksheet documents the test configuration and site number.

A Sheet tab may also contain one or more Append worksheets (Append1, Append2, ... and so on), as discussed in the [Append execution of tests, test sequences, and Project Plans](#) in Section 6 of the Reference Manual. Each Append worksheet behaves like a Data worksheet. However, its data cannot be plotted on a separate Graph tab graph, only on the same graph as the Data worksheet data. For more information about Sheet tabs refer to [Understanding and using the Data worksheet of a Sheet tab](#).

**Each worksheet contains the following controls:**

- A data-source identifier.
- Save as button.

## Opening a Sheet tab

**To open a Sheet tab:**

1. In the site navigator, enter the site number where the ITM or UTM was executed, using the spin button controls (the little arrows at the right).
2. In the project navigator, double-click the name of the ITM or UTM that acquired the data. An ITM or UTM window appears displaying the definition tab for the selected ITM or UTM.

---

**NOTE** *If the project plan contains multiple instances of an ITM or UTM under the same name, each instance generates its own data and has its own unit identification (UID). Ensure that you select the correct instance of the ITM or UTM.*

---

3. Click the **(ITM or UTM) Sheet** tab. The Data worksheet of the Sheet tab appears, as well as tabs that provide access to the corresponding Calc and Settings worksheets.

[Figure 2-21](#) is the Data worksheet of a Sheet tab for the `vds-id` ITM, showing data for multiple sweeps. [Figure 2-22](#) is the Data worksheet of a Sheet tab for the `vgs-id` ITM, showing formulator calculation results, in addition to test data.



Figure 2-21

Data worksheet of a Sheet tab containing data for multiple sweeps

	A	B	C	D	E	F	G
1	DrainCurrent	DrainVolt(1)	GateVolt(1)	DrainCurrent	DrainVolt(2)	GateVolt(2)	DrainCurrent D
2	2.69733E-11	0.00000E-01	2.00000E+00	2.35055E-11	0.00000E-01	3.00000E+00	2.48973E-11
3	8.59179E-04	1.00000E-01	2.00000E+00	1.25070E-03	1.00000E-01	3.00000E+00	1.56022E-03
4	1.65599E-03	2.00000E-01	2.00000E+00	2.45347E-03	2.00000E-01	3.00000E+00	3.07960E-03
5	2.39495E-03	3.00000E-01	2.00000E+00	3.61199E-03	3.00000E-01	3.00000E+00	4.56253E-03
6	3.06948E-03	4.00000E-01	2.00000E+00	4.71752E-03	4.00000E-01	3.00000E+00	5.99838E-03
7	3.68147E-03	5.00000E-01	2.00000E+00	5.77335E-03	5.00000E-01	3.00000E+00	7.39149E-03
8	4.23041E-03	6.00000E-01	2.00000E+00	6.77790E-03	6.00000E-01	3.00000E+00	8.73960E-03
9	4.71815E-03	7.00000E-01	2.00000E+00	7.73332E-03	7.00000E-01	3.00000E+00	1.00459E-02
10	5.14285E-03	8.00000E-01	2.00000E+00	8.63258E-03	8.00000E-01	3.00000E+00	1.12998E-02
11	5.50827E-03	9.00000E-01	2.00000E+00	9.47832E-03	9.00000E-01	3.00000E+00	1.25073E-02
12	5.81830E-03	1.00000E+00	2.00000E+00	1.02721E-02	1.00000E+00	3.00000E+00	1.36691E-02
13	6.07518E-03	1.10000E+00	2.00000E+00	1.10084E-02	1.10000E+00	3.00000E+00	1.47774E-02
14	6.28476E-03	1.20000E+00	2.00000E+00	1.16913E-02	1.20000E+00	3.00000E+00	1.58351E-02
15	6.45256E-03	1.30000E+00	2.00000E+00	1.23199E-02	1.30000E+00	3.00000E+00	1.68412E-02
16	6.58471E-03	1.40000E+00	2.00000E+00	1.28972E-02	1.40000E+00	3.00000E+00	1.77992E-02
17	6.68605E-03	1.50000E+00	2.00000E+00	1.34201E-02	1.50000E+00	3.00000E+00	1.87017E-02
18	6.76269E-03	1.60000E+00	2.00000E+00	1.38916E-02	1.60000E+00	3.00000E+00	1.95522E-02
19	6.81990E-03	1.70000E+00	2.00000E+00	1.43148E-02	1.70000E+00	3.00000E+00	2.03532E-02
20	6.86246E-03	1.80000E+00	2.00000E+00	1.46891E-02	1.80000E+00	3.00000E+00	2.10999E-02
21	6.89466E-03	1.90000E+00	2.00000E+00	1.50178E-02	1.90000E+00	3.00000E+00	2.17957E-02
22	6.91982E-03	2.00000E+00	2.00000E+00	1.53037E-02	2.00000E+00	3.00000E+00	2.24406E-02
23	6.94024E-03	2.10000E+00	2.00000E+00	1.55508E-02	2.10000E+00	3.00000E+00	2.30381E-02
24	6.95734E-03	2.20000E+00	2.00000E+00	1.57602E-02	2.20000E+00	3.00000E+00	2.35848E-02
25	6.97220E-03	2.30000E+00	2.00000E+00	1.59371E-02	2.30000E+00	3.00000E+00	2.40841E-02

Figure 2-22

Data worksheet of a Sheet tab containing both data and formulator results

Formulator results

	A	B	C	D	E	F
1	DrainI	GateV	GM	SQRID	IDLIN	VT
2	6.74225E-12	0.00000E-01	#REF	2.59658E-06	-1.02731E-02	1.19814E+00
3	3.93329E-11	1.00000E-01	3.25906E-10	6.27159E-06	-9.41566E-03	
4	2.33063E-10	2.00000E-01	1.93730E-09	1.52664E-05	-8.55824E-03	
5	1.70939E-09	3.00000E-01	1.47633E-08	4.13448E-05	-7.70082E-03	
6	1.15872E-08	4.00000E-01	9.87778E-08	1.07644E-04	-6.84340E-03	
7	8.01263E-08	5.00000E-01	6.85391E-07	2.83066E-04	-5.98598E-03	
8	5.51763E-07	6.00000E-01	4.71636E-06	7.42807E-04	-5.12857E-03	
9	3.49087E-06	7.00000E-01	2.93911E-05	1.86839E-03	-4.27115E-03	
10	1.86721E-05	8.00000E-01	1.51812E-04	4.32112E-03	-3.41373E-03	
11	7.77797E-05	9.00000E-01	5.91077E-04	8.81928E-03	-2.55631E-03	
12	2.34196E-04	1.00000E+00	1.56417E-03	1.53035E-02	-1.69890E-03	
13	5.24028E-04	1.10000E+00	2.89831E-03	2.28917E-02	-8.41479E-04	
14	9.47303E-04	1.20000E+00	4.23275E-03	3.07783E-02	1.59389E-05	
15	1.48661E-03	1.30000E+00	5.39307E-03	3.85566E-02	8.73355E-04	
16	2.11728E-03	1.40000E+00	6.30672E-03	4.60139E-02	1.73077E-03	
17	2.81994E-03	1.50000E+00	7.02654E-03	5.31031E-02	2.58819E-03	
18	3.57640E-03	1.60000E+00	7.56467E-03	5.98030E-02	3.44561E-03	
19	4.36846E-03	1.70000E+00	7.92060E-03	6.60943E-02	4.30303E-03	
20	5.18973E-03	1.80000E+00	8.21268E-03	7.20398E-02	5.16044E-03	
21	6.02748E-03	1.90000E+00	8.37746E-03	7.76368E-02	6.01786E-03	
22	6.87742E-03	2.00000E+00	8.49943E-03	8.29302E-02	6.87528E-03	
23	7.73293E-03	2.10000E+00	8.55512E-03	8.79371E-02	7.73269E-03	
24	8.59011E-03	2.20000E+00	8.57181E-03	9.26829E-02	8.59011E-03	
25	9.44753E-03	2.30000E+00	8.57417E-03	9.71984E-02	9.44753E-03	
26	1.03005E-02	2.40000E+00	8.52945E-03	1.01491E-01	1.03005E-02	



---

**NOTE** The #REF notation in a cell indicates that a valid value could not be calculated by the formulator. This occurs when a formulator function needs multiple rows as arguments, when a calculated value is out of range, when a divide by zero is attempted, and so on.

In the GM column in [Figure 2-22](#), note the #REF notation in the first row. Each value in the GM column is a difference coefficient that is calculated as the ratio  $\Delta\text{DrainI} / \Delta\text{GateV}$ , where  $\Delta\text{DrainI}$  and  $\Delta\text{GateV}$  are differences between values in the present row and values in the previous row. Because, no previous row exists before the first row, a valid calculation is not possible for the first row. Hence, the formulator returns the #REF notation.

A column will contain multiple instances of #REF if the formulator function requires multiple prior cells for the calculation. For example, if the MAVG function is using five data points to calculate a moving average of a column containing five values, the first two and last two cells will contain #REF.

---

## Understanding and using the Data worksheet of a Sheet tab

The Data worksheet first appears when you open the Sheet tab (see [Figures 2-21](#) and [2-22](#)). The Data worksheet displays all the data that was last generated by the ITM or UTM for a particular site. The Data worksheet also contains the results of any formulator calculations that were performed on the last-generated data.

### Features of the Data worksheet are:

- Data is reported in Microsoft® Excel®-compatible format, each column containing the results for one test parameter or for a formulator calculation.

---

**NOTE** Some formulator calculations return only a single value.

---

The display of all columns for the test may span several pages horizontally. The display of data in a single column may span several pages vertically.

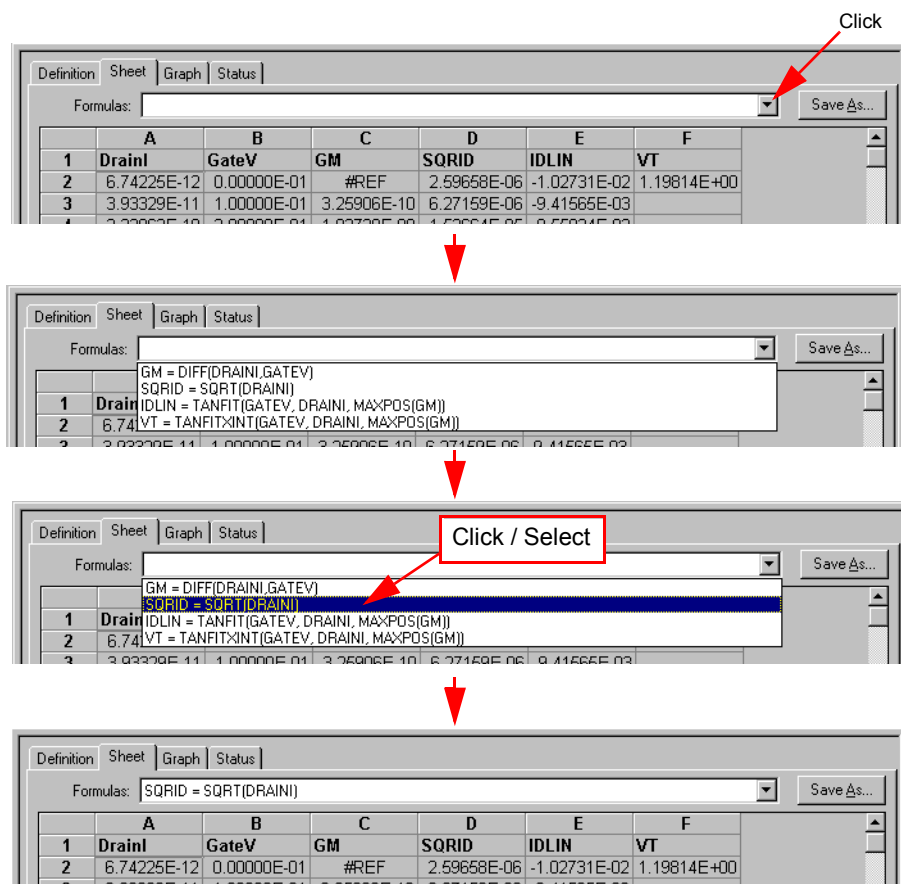
- Each column heading identifies the data below it:
  - The name of a test-results parameter (for example, current or voltage) that is assigned by KITE, by you (for an ITM only), or by the KULT programmer (for a UTM only). For ITM current and voltage naming, refer to the [Reference Manual, Understanding and configuring the Measuring Options area, page 6-128](#).
  - The name of a formulator results parameter.
- The data-source identifier, the formula box, and the save as button, each of which are discussed below.
- The contents of the Data worksheet are display-only. However, you can manipulate the contents of the Data worksheet after linking it to or pasting it in the Calc worksheet.

## Understanding the formula box of the Data worksheet

If a column in the Data worksheet contains the results of formulator calculations, you can locally display the formula (equation) that was used to obtain the results. Display the formula by selecting it from the formula box, as illustrated in [Figure 2-23](#). The steps in [Figure 2-23](#) display the formula that was used to obtain the SQRID results in [Figure 2-22](#).



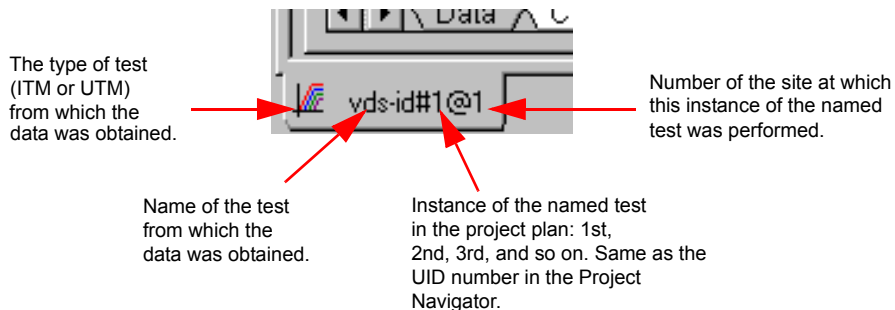
Figure 2-23  
Displaying a formulator equation using the formula box



### Understanding the data-source identifier

The ITM or UTM window tab at the bottom of all Sheet tab windows identifies the source of the data in the Sheet tab, as shown in [Figure 2-24](#).

Figure 2-24  
Data-source identifier



### Saving a worksheet

#### Saving a Sheet tab to the project plan

To save the displayed data to the project plan, do one of the following:



- Click **Save** in the **File** menu
- Click the single floppy-disk toolbar button
- Press **Ctrl+S** on the keyboard

### Saving the Sheet tab to an external spreadsheet file using the save as button

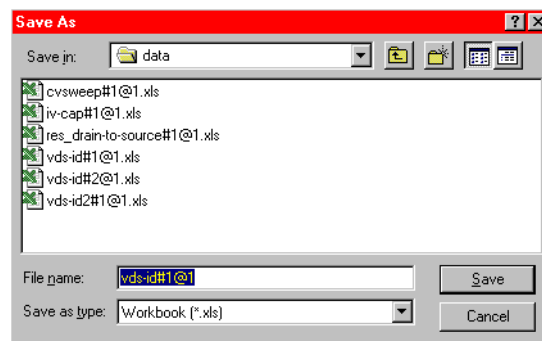
All data in the Sheet tab for a test is in Microsoft® Excel®-compatible format, with the .xls extension. In other words, the combined worksheets in the Sheet tab (including any Append1, Append 2, and so on worksheets) effectively comprise a workbook that can be used directly in an Microsoft® Excel®-compatible spreadsheet program.

**To save the contents of all Sheet tab worksheets to a designated folder simultaneously in a single .xls file:**

1. Click **Save As** in the upper right corner of any of the three worksheets. The Save As window displays, with workbook (\*.xls) as the default file type. See [Figure 2-25](#).

Figure 2-25

**Data Save As window, configured for workbook files**



2. In the **Save In** edit box of the save as window, select the location for the text file.
3. In the **File name** edit box of the save as window, Keithley Instruments recommends that you retain the default selection, which contains the data-source identifier (refer to [Understanding the data-source identifier](#)).
4. In the **Save as type** box, make no changes; retain the \*.xls type.
5. Click **Save**.

---

**NOTE** Do not attempt to use the save as button to save data to the project plan.

---

### Understanding Append worksheets

The following applies to the worksheets that are created by Append executions:

- The data generated for each Append execution of a test is located in an individual “Appendn” worksheet where “n” designates the nth Append execution. For example, the worksheets are labeled Append1, Append2, ... and so on.

---

**NOTE** You can specify the maximum number of Append executions and worksheets (the maximum value of n). After the maximum number of Append worksheets have been generated, the data from each Append execution replaces the data from the previous Append execution. For example, if the maximum value of n is 4, the data from the fifth Append execution replaces the data from the fourth Append execution. Refer also to [Append execution of tests, test sequences, and Project Plans](#) in Section 6 of the Reference Manual.

---

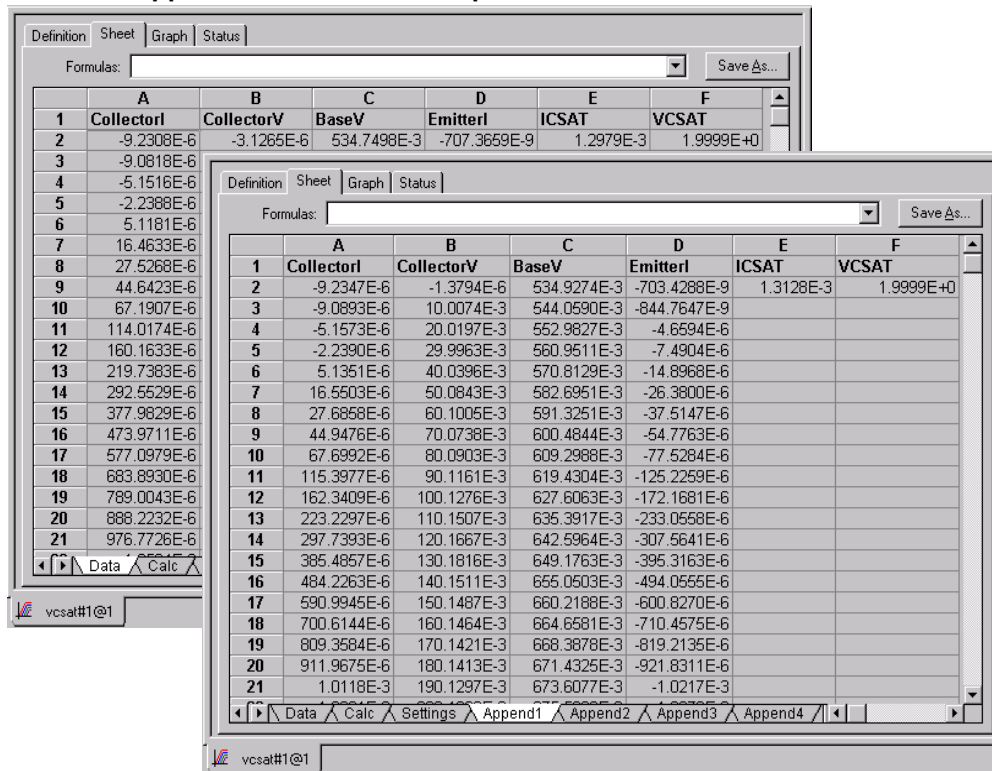


- Each Append worksheet is labeled with a separate tab to distinguish it from the Data worksheet for the test.
- Each Append worksheet contains the same columns and rows as the Data worksheet for the test.
- Each Append worksheet may be manipulated in the same way as the Data worksheet for the test.

See [Figure 2-26](#).

Figure 2-26

**Data and Append1 worksheets for a particular vcsat test**



**NOTE** To display hidden Append worksheet tabs, use the scroll buttons located at the left side of the tabs.

Figure 2-27

**Append worksheet tabs**



Append executions are not restricted to individual tests. An entire test sequence (device plan or subsite plan) or a project plan may be Append executed "n" times, resulting in "n" separate Append worksheets for each test in the sequence or project plan. Multi-site Append execution of a project plan results in multi-level sets of Append worksheets.



### Graphing the Append worksheet data

You can graph Append worksheet data in using the same procedure used to graph Data worksheet data. Refer to [Appending curves from multiple runs on a single graph](#) in Section 6 of the Reference Manual.

### Deleting Append worksheets

You can delete Append worksheets using the following three methods:

- **Clear Append Data method:** Involves the **Clear Append Data** toolbar button / menu item.
- **Run method:** Involves performing a **Run** execution.
- **Append Sets method:** Involves reducing the Project window **Append Sets** value.

The next subsection outlines advantages and disadvantages for each method.

#### Clear Append data method for deleting Append worksheets

Use the **Clear Append Data** function to permanently delete any or all Append worksheets for a selected test, test sequence, or project plan, either at one specific site or at all sites.

- **Advantages:**
  - Perhaps the easiest, most straightforward method.
  - Deletes Append worksheets without modifying the Data worksheet(s).
- **Disadvantages:**
  - Final Recovery from accidental deletion is not possible.

The Clear Append Data method is explained in the [Reference manual, Figure 6-228](#). If there is no Append data for an ITM or UTM, the Append list will be blank and the selection boxes for ITM / UTM Append data will be disabled.

## How to manage graphical test results in the Graph tab

### Opening a Graph tab

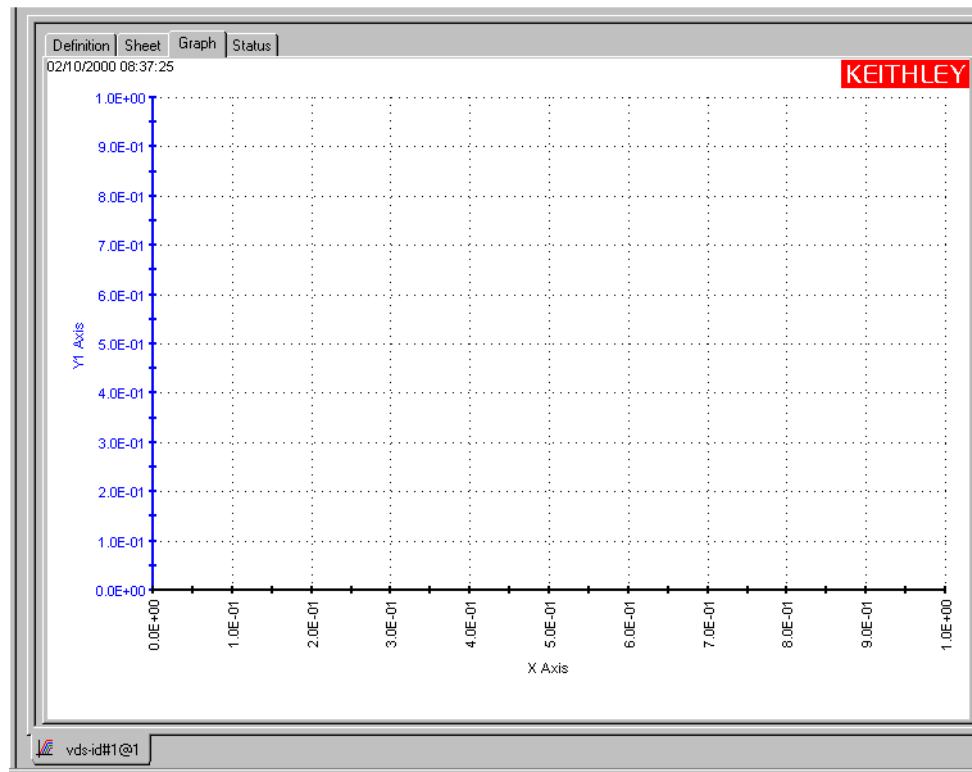
To open a Graph tab:

1. Open the ITM or UTM window for the selected test by double-clicking the test in the project navigator.
2. When the ITM or UTM window opens, click the displayed Graph tab. The Graph tab opens.

[Figure 2-28](#) displays an unconfigured graph for the `vds-id` ITM. The time and date at which the data was generated are displayed in the upper left corner. However, the axes are labeled and scaled generically, because no project data has yet been assigned to the axes.



Figure 2-28  
Example of an unconfigured Graph tab



The `vds-id` ITM is one of the ITMs that comes installed on your Model 4200-SCS with sample data, including a configured graph (Reference manual, Figure 6-6). The `vds-id` ITM has been used for illustration purposes through much of Section 6, including construction of the `u_build` project (Reference Manual, Building a completely new Project Plan, page 6-47). The definition tab for the `vds-id` ITM is shown in multiple places, including at the beginning of this section.

## Accessing the Graph tab windows

Several Graph tab windows control the properties of a graph.

You can access these windows in two ways:

- **Use the graph settings menu:** When defining a graph, you typically access all Graph tab windows using the graph settings menu.
- **Right-click on certain graph components:** When certain graph components are already displayed, you can open context-appropriate edit windows by right-clicking the components. Applicable graph components include titles, legends, comments, numerical coordinates, and values displayed through the data variables menu item.

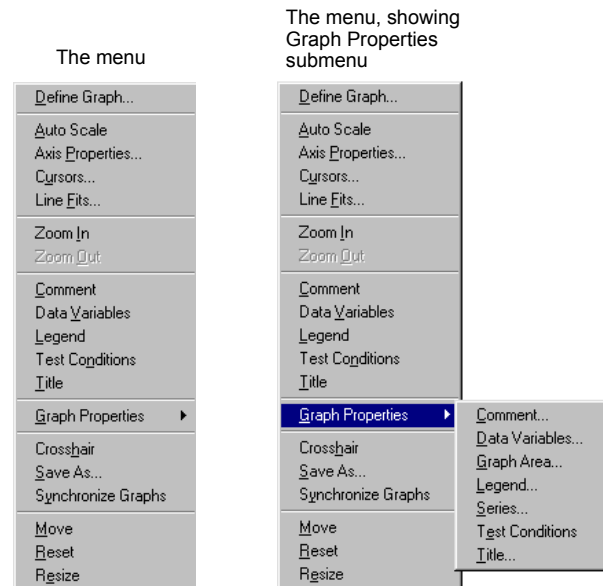
## Opening the graph settings menu

Open the graph settings menu by either of the following methods:

- **Menu access method I:** Right-click in any blank portion of the **Graph** tab (any place except on a **Graph** tab component). The **Graph Settings** menu appears as a pop-up menu. See Figure 2-29.



Figure 2-29  
Graph settings menu



- **Menu access method II:** In the **tools** menu of the KITE window, select graph settings. The menu that appears is identical to the pop-up menu shown in [Figure 2-29](#).

## Understanding the graph settings menu

Each item of the graph settings menu is summarized below:

- **Define graph:** Defines the parameters to be graphed and the axes on which these parameters are to be graphed. For more information, refer to [Defining data to be graphed on page 2-42](#).
- **Auto scale:** Automatically scales all axes at a single, chosen time. For more information, refer to the [Reference Manual, Automatically scaling the axes, page 6-217](#).
- **Axis properties:** Opens the Axes Properties window, which is the main access point for graph scaling and scale formatting. For more information, refer to the [Reference Manual, Defining the axis properties of the graph, page 6-213](#).
- **Cursors:** Opens the Cursors window, from which you can select and format cursors that display the precise numerical coordinates of specific points on the plot lines. For more information, refer to the [Reference Manual, Numerically displaying plot coordinates using cursors, page 6-229](#).
- **Line fits:** Allows you to fit lines directly to Graph tab plots. Up to two times. Select from the following types:
  - **Linear** (line through two data points)
  - **Regression** (regression line)
  - **Exponential**
  - **Logarithmic**
  - **Tangent**
- **Zoom in:** Allows you to enlarge and examine a small, selected part of the graph. For more information, refer to the [Reference Manual, Temporarily enlarging a selected area of the graph by zooming, page 6-279](#).
- **Zoom out:** Restores a graph to the original or previously zoomed size. For more information, refer to the [Reference Manual, Temporarily enlarging a selected area of the graph by zooming, page 6-279](#).



- **Comment:** Opens the Comment window, which allows you to add and format a comment. For more information, refer to the [Reference Manual, Adding a comment, page 6-273](#).
- **Data variables:** Opens the Data Variables window, from which you can configure the display of up to four data variables, along with the corresponding names. The data variables menu item also toggles the data-variable display. For more information about the Data Variables item, refer to the [Reference Manual, Numerically displaying extracted parameters and other data variables, page 6-257](#).
- **Legend:** Toggles the display of an automatically-created legend on and off. For more information about legends, refer to the [Reference Manual, Adding a legend, page 6-271](#).
- **Test Conditions:** Displays the primary test conditions used to obtain the data in the graph. For more information, refer to the [Reference Manual, Displaying test conditions, page 6-264](#).
- **Title:** Opens the Title window, which allows you to add and format a title. For more information, refer to the [Reference Manual, Adding a title, page 6-270](#).
- **Graph properties**
  - **Comment:** Opens the Comment window, which allows you to add and format a comment. Same function as comment in the main menu.
  - **Data variables:** Opens the Data Variables window, from which you can configure the display of up to four data variables, along with the corresponding names. Essentially, the same as Data Variables in the main menu, except that it allows you to open a Data Variables window without toggling the data-variables display.
  - **Graph area:** Opens the graph area menu, which allows you to change the graph foreground and background colors, toggle the time and date display, and make the graph 100% monochrome. For more information, refer to the [Reference Manual, Changing area properties of the graph, page 6-277](#).
  - **Legend:** Opens the legend properties window, which allows you to reformat the font, text or background color, or border of the legend. For more information, refer to the [Reference Manual, Adding a legend, page 6-271](#).
  - **Series:** Opens the Data Series properties window, from which you can define color, line pattern, plot symbol, and line width for each plot. For more information, refer to the [Reference Manual, Defining the plot properties of the graph: colors, line patterns, symbols, line widths, page 6-224](#).
  - **Test conditions:** Displays the primary test conditions used to obtain the data in the graph. For more information, refer to the [Reference Manual, Displaying test conditions, page 6-264](#).
  - **Title:** Opens the title window, which allows you to add and format a title. Same function as title in the main menu.
- **Crosshair:** Toggles the display of a pair of intersecting lines that can be positioned anywhere on the graph. For more information, refer to the [Reference Manual, Visually reading plot coordinates using cross hairs, page 6-242](#).
- **Save as:** Opens the save as window, which allows you to save a graph in bitmap (.bmp) format for use elsewhere, such as in a report. For more information, refer to the [Reference Manual, Saving a graph as a bitmap file, page 6-285](#).
- **Synchronize graphs:** For use when the presently open graph is one of several graphs for the same test (each graph representing the data for a different site). Selecting synchronize graphs automatically configures the graphs for all sites, identically, using the open graph as the master. For more information, refer to the [Reference Manual, Identically configuring the graphs resulting from one test executed at multiple sites, page 6-283](#).
- **Move:** Toggles between a normal cursor and a crossed-arrow cursor. Moving the crossed-arrow cursor moves the graph, allowing you to relocate it on the Graph tab. For more information, refer to the [Reference Manual, Changing the position of a graph, page 6-283](#).



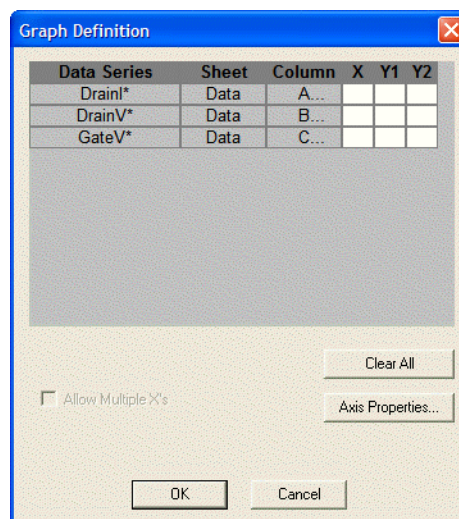
- **Reset:** Causes colors, graph size, and graph position to be restored to the defaults. For more information, refer to the [Reference Manual, Resetting certain graph properties to KITE defaults, page 6-285](#).
- **Resize:** Toggles between a normal cursor and a ruler cursor. Moving the ruler cursor expands or contracts the size of the graph. The new size is saved when the graph is saved. By contrast, selecting Zoom In affects only the view size, which cannot be saved. For more information, refer to the [Reference Manual, Changing the size of a graph, page 6-279](#).

## Defining data to be graphed

The Graph Definition window is used to define the data to be graphed. [Figure 2-30](#) shows the undefined Graph Definition window for a vds-id ITM.

Figure 2-30

**Graph definition window for a vds-id ITM (undefined)**



## Understanding table columns in the Graph Definition window

The table columns in the Graph Definition window are used as follows:

- **Data Series:** Lists the names (or other contents<sup>\*4</sup>) of every first-row cell of the data and Calc worksheets. If you have generated Append worksheets<sup>\*5</sup> for the test, the Data Series column also lists the names of every first-row cell in every Append worksheet. However, when multiple first-row cells name the same parameter (because multiple sets of data exist under that name) the following applies:
  - The name of the parameter is listed only once under Data Series, because it corresponds to a family of curves.
  - Asterisks (\*) appear next to all parameter names listed under Data Series.
- **Sheet:** Indicates whether the data comes from the Data worksheet, the Calc worksheet, or a specific Append worksheet.
- **Column:** Lists the parameter's Data, Calc, or Append worksheet column label (A, B, C, and so on).

4. KITE assumes that first-row cells contain variable names. However, a first-row Calc worksheet cell is allowed to contain a number, and KITE displays such a number under Data Series. Avoid placing numbers (or any unwanted plot parameter names) in the first row of a Calc worksheet.

5. For more information about generation and use of Append worksheets, refer to [Append execution of tests, test sequences, and Project Plans, Understanding and using Append worksheets of a Sheet tab, and Appending curves from multiple runs on a single graph](#) in Section 6 of the Reference Manual.



- **X, Y1, and Y2:** Are the axes of the graph, as follows:
  - X is the X axis.
  - Y1 is the Y axis on the left side of the graph.
  - Y2 is the Y axis on the right side of the graph.

---

**NOTE** *The scale and label of the Y2 axis are allowed to be different from the scale and label of the Y1 axis.*

---

The cells under the X, Y1, and Y2 may be selected and deselected by clicking the boxes.

- If you select a cell under X, the corresponding Data Series parameter is plotted on the X axis. KITE can plot multiple parameters on the X axes when the test does not define a family of curves (see the [Reference Manual, Allow Multiple X's, page 6-213](#)).
- Similarly, if you select a cell under Y1 or Y2, the corresponding Data Series parameter is plotted on the Y1 axis or the Y2 axis. KITE can plot multiple parameters on the Y1 and Y2 axes.

## Understanding buttons in the Graph Definition window

The buttons of the Graph Definition window are used as follows:

- **Clear all:** Click the **Clear All** button to clear all selections under columns X, Y1, and Y2.

---

**NOTE** *If you click the **Clear All** button by mistake, click the **Cancel** button to exit the Graph Definition window without making any changes.*

---

- **Axis properties:** A click of the axis properties button opens the axis properties window. You can also open the axis properties window by selecting axis properties in the graph settings menu. Before using the axis properties window, refer to the [Reference Manual, Defining the axis properties of the graph, page 6-213](#).

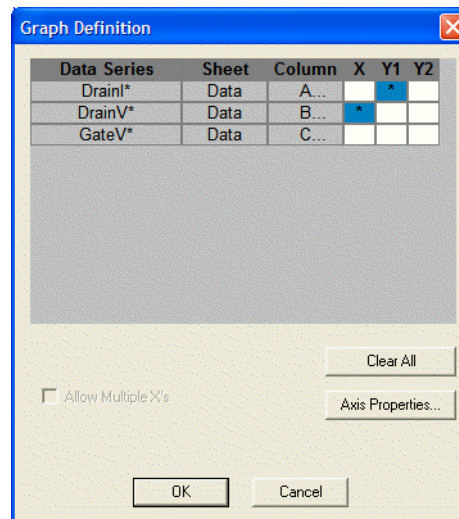
## Opening and using the Graph Definition window

To open and use the Graph Definition window:

1. In the Graph tab, display the graph settings menu by right-clicking the graph or by selecting **Tools > Graph Settings**.
2. In the graph settings menu, select **Define Graph**. The Graph Definition window opens.
3. Using the Graph Definition window, indicate which parameters are to be plotted and assign them to appropriate axes by selecting the appropriate X, Y1, and Y2 cells.

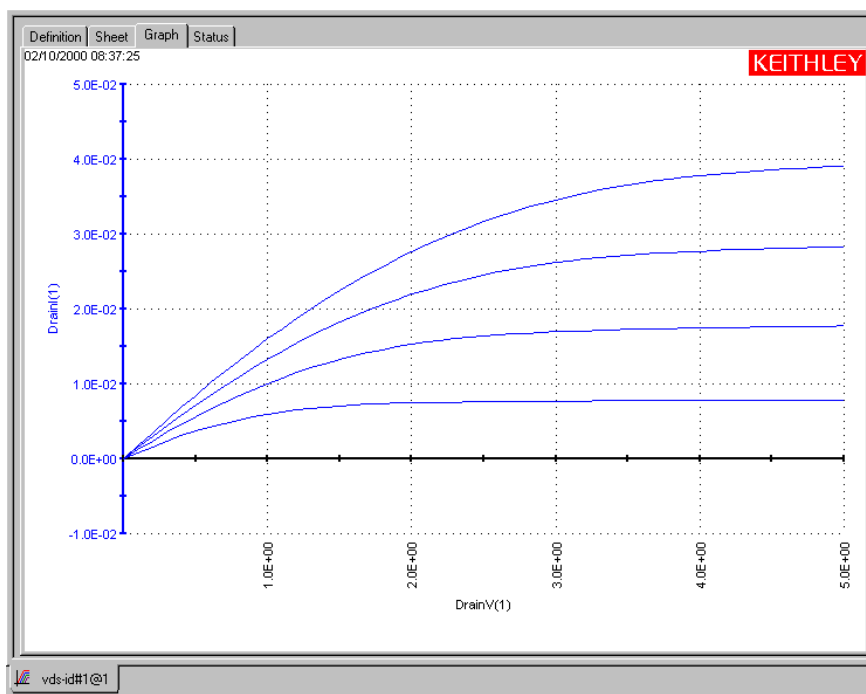


Figure 2-31  
Configured Graph Definition window for a vds-id ITM



- Click **OK**. The graph now displays plots of the selected parameters.  
In Figure 2-32, the vds-id graph now displays scaled axes and a series of four plots, based on the selections shown in Figure 2-31. The family of curves corresponds to four sets of data generated by drain-voltage sweeps at four different gate voltages.

Figure 2-32  
View of vds-id graph after Graph Definition window configuration



The axis labels shown in Figure 2-32 are not yet optimally named. KITE inserted the default data sheet column labels for sweep #1 of the Data Series. For more information about renaming the axis labels, refer to the [Reference Manual, Defining the axis properties of the graph](#), page 6-213.



## KITE library management

### Submitting devices, ITMs, and UTMs to libraries

If you create a customized device or test and wish to reuse it in more than one place in other project plans, you must first submit it to a device or test library.

#### Submitting devices to a library

You may submit a project plan device (an empty device plan) to any Device Library, if you submit it with a name that does not duplicate a device name that is already in the library.

##### To submit a device to a library:

1. In the project navigator, locate the subsite plan that contains the device plan you wish to submit. [Figure 2-33](#) highlights `subsite_b` of the `u_build` project plan (developed for illustration purposes during the example in the [Reference Manual, Building a completely new Project Plan, page 6-47](#). The `subsite_b` plan presently contains an added composite device plan to be submitted.

Figure 2-33

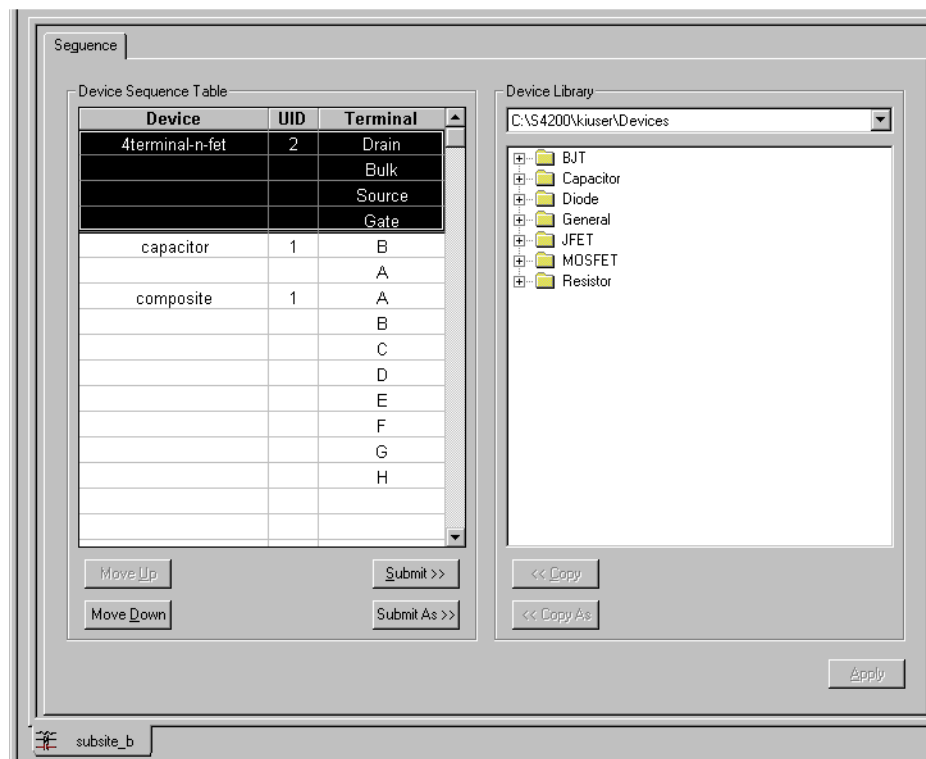
#### Subsite plan containing the device plan to be submitted

Project Tree	UID
u_build	0
InitializationSteps	0
subsite_a	0
4terminal-n-fet	1
vds-id	1
vds-id	2
4terminal-n-fet-2nd_in_subsite	1
vds-id2	1
subsite_b	0
4terminal-n-fet	2
res_drain-to-source	1
capacitor	1
charg_char	1
composite	1
TerminationSteps	0

2. Double-click the subsite plan that contains plans for the device that you wish to submit. The subsite plan window opens. See [Figure 2-34](#).



Figure 2-34  
Subsite plan window containing the device plan to be submitted



3. If you wish to submit the device plan to a Device Library directory other than the default Device Library directory,<sup>6</sup> select the alternate Device Library directory in the Device Library box of the subsite plan window.

**NOTE** Only the default Device Library directory is available in the Device Library box, unless other Device Library directories were previously added through the *directories* tab of the KITE Options window. Access the KITE Options window by selecting **Options** on the **Tools** menu. through the **Tools > Options** menu.

4. In the Device Library directory tree, select a destination folder that is appropriate for the device(s).
5. In the Device Sequence Table of the subsite plan window, select the device(s) to be submitted.

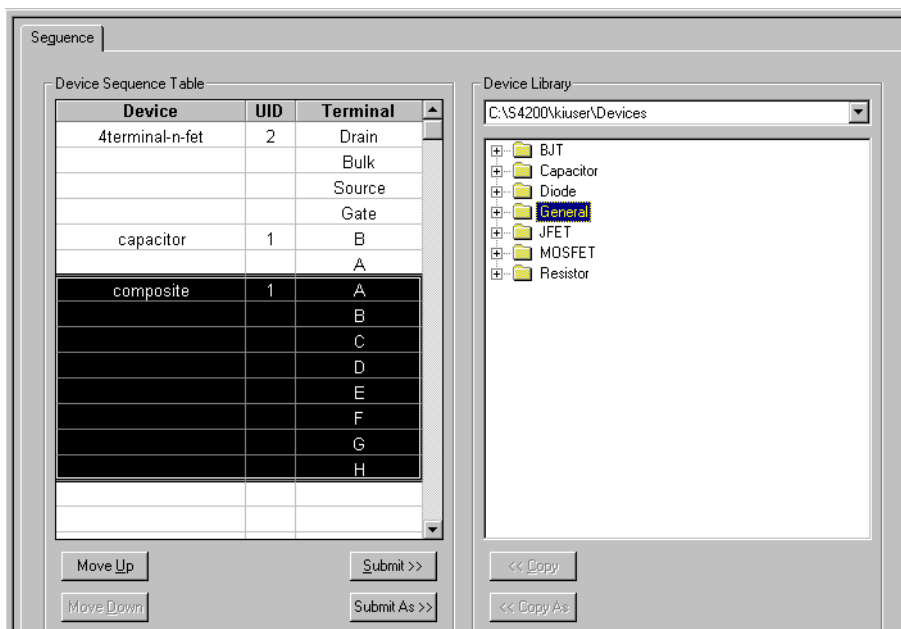
**NOTE** You may select and submit multiple device plans at the same time. To select a sequential group of device plans, hold down the **Shift** key while clicking the first and last device plan in the sequence. To select a group of individual device plans, hold down the **Ctrl** key while clicking the individual device plans.

Figure 2-35 shows the composite device selected in the Device Sequence Table, and the General destination folder selected in the Device Library.

6. For example, the C:\S4200\kiuser\Devices factory-default directory or another directory that was specified as the default using KCON, such as C:\S4200\YourName\Devices.



Figure 2-35  
Selected device and destination folder



6. Do one of the following:
  - If you wish to submit the selected device or devices with the original name(s), click the **Submit >>** button in the subsite plan window. The selected device or devices is submitted to the chosen folder.
  - Stop here. You have finished the device submission procedure.
  - If you wish to submit the selected device or devices under a different name or names, click the **Submit As >>** button in the subsite plan window. The Submit device dialog box opens, displaying the original name of the device (or, if you selected multiple devices, displaying the original name of one of the devices). See [Figure 2-36](#).

Figure 2-36  
Submit device dialog box



7. In the **As** edit box of the **Submit device** dialog box, type a name for the device.
8. Click **OK**. One of the following occurs:
  - If you selected only one device in the Device Sequence Table, the selected device is submitted to the chosen folder under the new name. Stop here; you have finished the device submission procedure.
  - If you selected multiple devices in the Device Sequence Table, the following occurs:
    - a. The device that you renamed in Step 7 is submitted to the chosen folder under the new name.
    - b. Then, another **Submit device** dialog box opens for another selected device.
9. Repeat Steps 7 and 8 until all of the selected devices have been submitted (until no more **Submit device** dialog boxes open).



## Submitting tests to a library

You may submit one or more ITMs or UTM's to any test library if you submit them with names that do not duplicate test names that are already in the library.

**NOTE** Before submitting any UTM to a library, make sure that it is configured. If you try to submit an unconfigured UTM, KITE displays the message shown in [Figure 2-37](#).

Figure 2-37

### Unconfigured UTM message

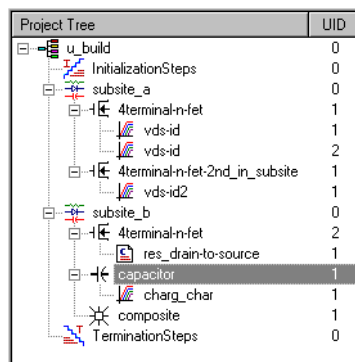


To submit the UTM's or ITM's (for simplicity, these will be referred to as “tests” for the rest of this topic):

1. In the project navigator, locate the device plan that contains the test or tests that you wish to submit. [Figure 2-38](#) shows an example device plan capacitor from an example `u_build` project plan used in the [Reference Manual, Building a completely new Project Plan, page 6-47](#)). The capacitor device plan contains the `charg_char` ITM to be submitted.

Figure 2-38

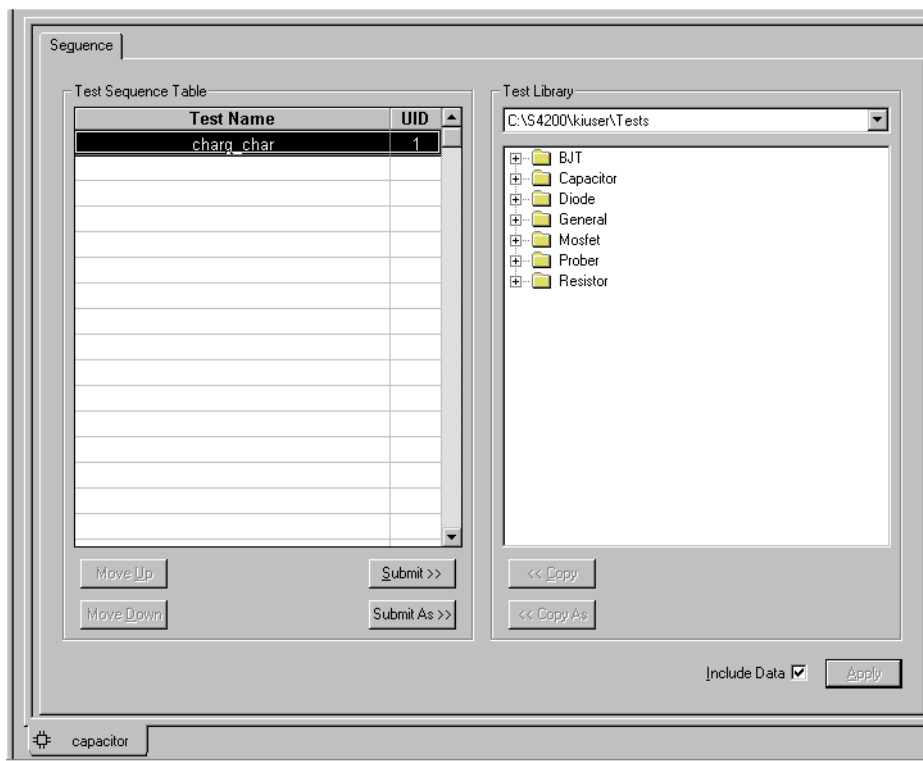
### Device plan containing an ITM to be submitted



2. Double-click the device plan that contains the test or tests that you wish to submit. The device plan window opens. See [Figure 2-39](#).



Figure 2-39  
Device plan window containing an ITM to be submitted



3. If you want to submit the tests to a test library directory other than the default test library directory,<sup>\*7</sup> select the alternate test library directory in the Test Library box in the device plan window.

**NOTE** Only the default test library directory is available in the Test Library box, unless other test library directories were previously added through the directories tab of the KITE Options window. The KITE Options window is accessed by selecting **Options** on the **Tools** menu.

4. In the Test Library directory tree, select a destination folder that is appropriate for the tests.
5. In the Test Sequence Table of the device plan window, select the test(s) to be submitted.

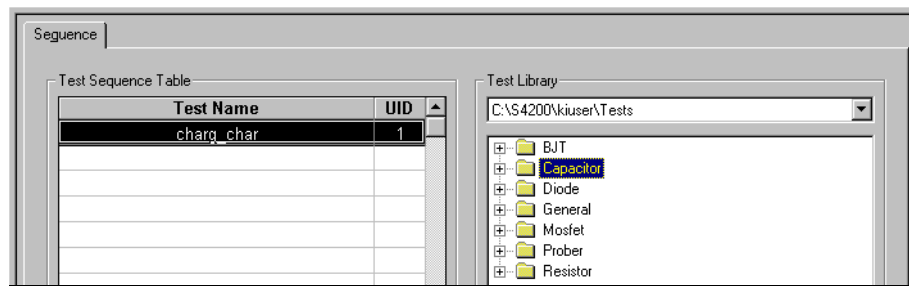
**NOTE** You may select and submit multiple ITMs and UTMs at the same time. To select a sequential group of ITMs and UTMs, hold down the **Shift** key while clicking the first and last ITM / UTM in the sequence. To select a group of individual ITMs and UTMs, hold down the **Ctrl** key while clicking the individual ITMs and UTMs.

Figure 2-40 shows the `charg_char` ITM selected in the Test Sequence Table, and the Capacitor destination folder selected in the Test Library.

7. For example, the C:\\$4200\kiuser\Tests factory-default directory or another directory that was specified as the default using KCON, such as C:\\$4200\YourName\Tests.

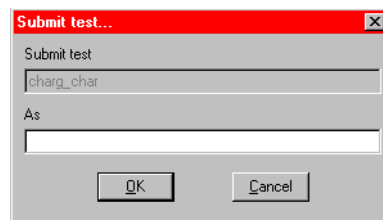


Figure 2-40  
Selected ITM and destination folder



6. Do one of the following:
  - If you wish to submit the selected test(s) with the original name(s), click the **Submit >>** button in the device plan window. The selected test(s) is submitted to the chosen folder. Stop here. You have finished the test submission procedure.
  - If you wish to submit the selected test(s) under a different name(s), click the **Submit As >>** button in the device plan window. The submit test dialog box opens, displaying the original name of the test (or, if you selected multiple tests, displaying the original name of one of the tests). See [Figure 2-41](#).

Figure 2-41  
Submit test dialog box



7. In the **As** edit box of the **Submit test** dialog box, type the submittal name for the test.
8. Click **OK**. One of the following occurs:
  - If you selected only one test in the **Test Sequence Table**, the selected test is submitted to the chosen folder under the new name. Stop here. You have finished the test submission procedure.
  - If you selected multiple tests in the **Test Sequence Table**, the following occurs:
    - The test that you renamed in Step 7 is submitted to the chosen folder under the new name.
    - Then, another **Submit test** dialog box opens for another selected test.
9. Repeat Steps 7 and 8 until all of the selected tests have been submitted (until no more **Submit test** dialog boxes open).



# Common Device Characterization Tests

In this section:

Topic	Page
<b>How to perform an I-V test on my device</b>	<b>3-4</b>
Default project overview	3-4
4- terminal n-MOSFET tests	3-5
Three terminal NPN BJT tests	3-6
Two wire resistor test	3-6
Diode tests	3-7
Connections	3-9
Leveraging the default project	3-9
Copying entire KITE project	3-10
Copying individual tests using the test library manager	3-10
Changing KITE startup behavior	3-11
<b>How to perform a C-V test on my device</b>	<b>3-13</b>
KITE ITM configuration	3-13
Definition tab	3-13
Terminal settings	3-13
Forcing functions and measure options	3-15
Selecting the forcing function	3-16
Setting the DC bias conditions	3-16
Setting the AC drive conditions	3-16
Measure settings	3-16
Test conditions	3-17
Advanced settings (terminal properties)	3-17
Status	3-18
Compensation	3-19
CVU ITM examples	3-19
CVU Voltage Bias	3-19
CVU Voltage Sweep	3-20
CVU Voltage List Sweep	3-21
CVU Frequency Sweep (bias)	3-22
CVU Frequency Sweep (step)	3-24
<b>How to perform a Pulsed I-V test on my device</b>	<b>3-26</b>
Introduction (PIV-A and PIV-Q)	3-26
What is Pulse IV	3-26
Why use Pulse IV	3-26
What PulseIV Packages are available for the Model 4200-SCS	3-27
Pulse IV for CMOS:Model 4200-PIV-A	3-27
What is the PIV-A PulseIV Package	3-27
Target applications and test projects for PIV-A	3-27
4200-PIV-A test connections	3-28
Supplied interconnect parts	3-28
Supplied tools	3-29
Model 8101- PIV test fixture	3-30
Prober Interconnect	3-31
DC Prober Interconnect	3-31
RF Prober Interconnect	3-32
PIV-A interconnect assembly procedure	3-33



<b>Using the PulseIV-Complete project for the first time</b>	<b>3-35</b>
Running AutocalScope	3-36
Running PulseIVCal	3-36
Running vds-id DC ITM	3-37
Running vds-id-pulse UTM	3-38
Running vds-id-pulse-vs-dc UTM	3-39
Running vgs-id DC ITM	3-40
Running vgs-id-pulse UTM	3-40
Running vgs-id-pulse-vs-dc UTM	3-41
Running scope-shot	3-41
Adjustable parameters in scopeshot_cal_pulseiv	3-42
Tips for using Pulse IV	3-43
Comparing DC and pulse results	3-43
<b>Pulse IV UTM descriptions</b>	<b>3-46</b>
cal_pulseiv	3-46
vdsid_pulseiv	3-48
Vdld_Pulse_DC_Family_pulseiv	3-50
vgsid_pulseiv	3-53
Vgld_DC_Pulse_pulseiv	3-55
scopeshot_cal_pulseiv	3-59
scopeshot_pulseiv	3-61
vdsid_pulseiv_demo	3-63
vgsid_pulseiv_demo	3-63
scopeshot_pulseiv_demo	3-63
<b>How to perform a Quiescent-point Pulsed I-V test (PIV-Q) on my device</b>	<b>3-64</b>
Q-Point Pulse IV – Model 4200-PIV-Q	3-64
What is the PIV-Q package	3-64
Target applications and test projects	3-64
<b>How to perform a reliability (stress-measure) tests on my device</b>	<b>3-66</b>
Connecting devices for stress / measure cycling	3-66
Overviewing the cycling-related tabs	3-67
Configuring subsite cycling	3-67
Understanding the Subsite Setup tab	3-67
Configuring the Subsite Setup tab	3-68
Step A: Enable cycling	3-68
Step B: Choose the mode of cycling	3-68
Step C1: Specify cycle timing; linear, log or list (stress / measure mode only)	3-68
Step C2: Specify number of cycles (cycle mode only)	3-70
Step D: Set periodic test intervals (stress / measure mode, log timing only)	3-70
Step E: Update and save the subsite setup configuration	3-71
Configuring device stress properties	3-71
Setting AC stress properties	3-73
<b>How to perform AC stress for wafer level reliability (WLR)</b>	<b>3-75</b>
Device Stress Properties configuration notes	3-78
Segment Stress / Measure Mode	3-81
Segment Arb™ stressing	3-82
Segment Stress / Measure Mode configuration	3-83
Configure stress counts	3-83
Configure Device Stress Properties	3-84
Executing subsite cycling	3-86
Multiple subsite cycling	3-86
Subsite cycling data sheets	3-86
Cycle Mode data sheet	3-86
Stress / Measure Mode data sheet	3-87
Settings window	3-88
Subsite cycling graphs	3-90
Cycle mode	3-90
Stress/measure mode	3-91
Configuration sequence for subsite cycling	3-92
<b>How to perform a flash memory test on my device</b>	<b>3-94</b>



Introduction .....	3-94
Theory of operation .....	3-94
Programming and erasing flash memory .....	3-94
Endurance testing .....	3-100
Disturb testing .....	3-100
Using a switch matrix .....	3-101
Pulse waveforms for NVM testing .....	3-102
Using Kpulse to create and export Segment Arb™ waveforms .....	3-103
Entering Segment Arb™ values into UTM array parameters .....	3-105
Flash connections .....	3-108
Direct connection to single DUT .....	3-112
Direct connection to array DUT for disturb testing .....	3-114
Switch matrix connection to array DUT .....	3-115
Flash projects .....	3-116
Flash-NAND tests .....	3-118
Flash-NOR tests .....	3-123
Flash-switch tests .....	3-123
Running any Flash Project for the first time .....	3-124
Running the Flash-NAND, Flash-NOR or Flash-Switch Project .....	3-125
Running the Program or Erase UTM .....	3-126
Running the Fast-Program-Erase UTM .....	3-128
Running the SetupDC UTM .....	3-129
Running the Vt-MaxGm ITM .....	3-129
Running the ConPin-Pulse or ConPin-DC UTM (Switch projects only) .....	3-129
FlashEndurance-NAND tests .....	3-130
FlashEndurance-NOR tests .....	3-136
FlashEndurance-switch tests .....	3-136
Running a FlashEndurance or FlashDisturb project .....	3-137
Running endurance or disturb looping .....	3-139
FlashDisturb tests .....	3-140
Explanation of flash UTM parameters .....	3-141
Error codes .....	3-143
Troubleshooting .....	3-144
No pulse output .....	3-144
Voltage levels do not match expected values .....	3-145
How to perform Charge Pumping .....	3-145
How to perform a Charge Trapping test .....	3-146
Slow single pulse charge trapping high K gate stack .....	3-146
Charge trapping procedure .....	3-147



## How to perform an I-V test on my device

**NOTE** It is assumed that the reader of this section already has a basic understanding of the Model 4200-SCS software environment and terminology. Please review [Model 4200-SCS Software Environment, page 2-3](#) before proceeding to Section 3.

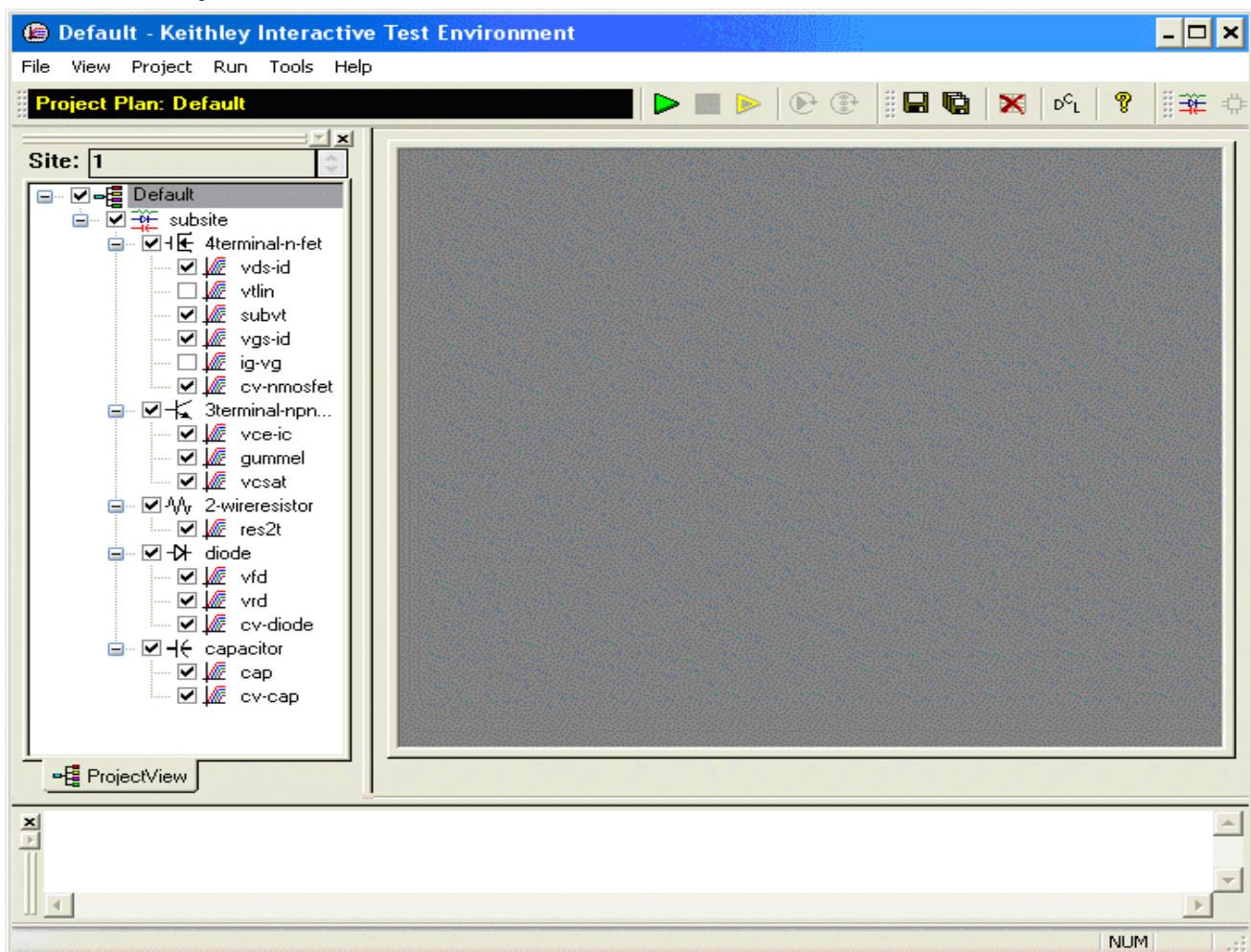
### Default project overview

The Keithley Interactive Test Environment (KITE) default project contains more than ten of the most common I-V tests a typical user might perform on a regular basis. These tests serve as examples and intended to be copied and modified to work for your own devices. These default tests cover 4-terminal FETs, three terminal BJTs, two terminal diodes, resistors, and capacitors.

The KITE project default should open automatically upon starting the KITE application. If it does not, the default project can be found at C:\S4200\kiuser\default\default.kpr.

Upon opening the default project in KITE, observe the following tests (see [Figure 3-1](#)):

Figure 3-1  
Project Plan: Default



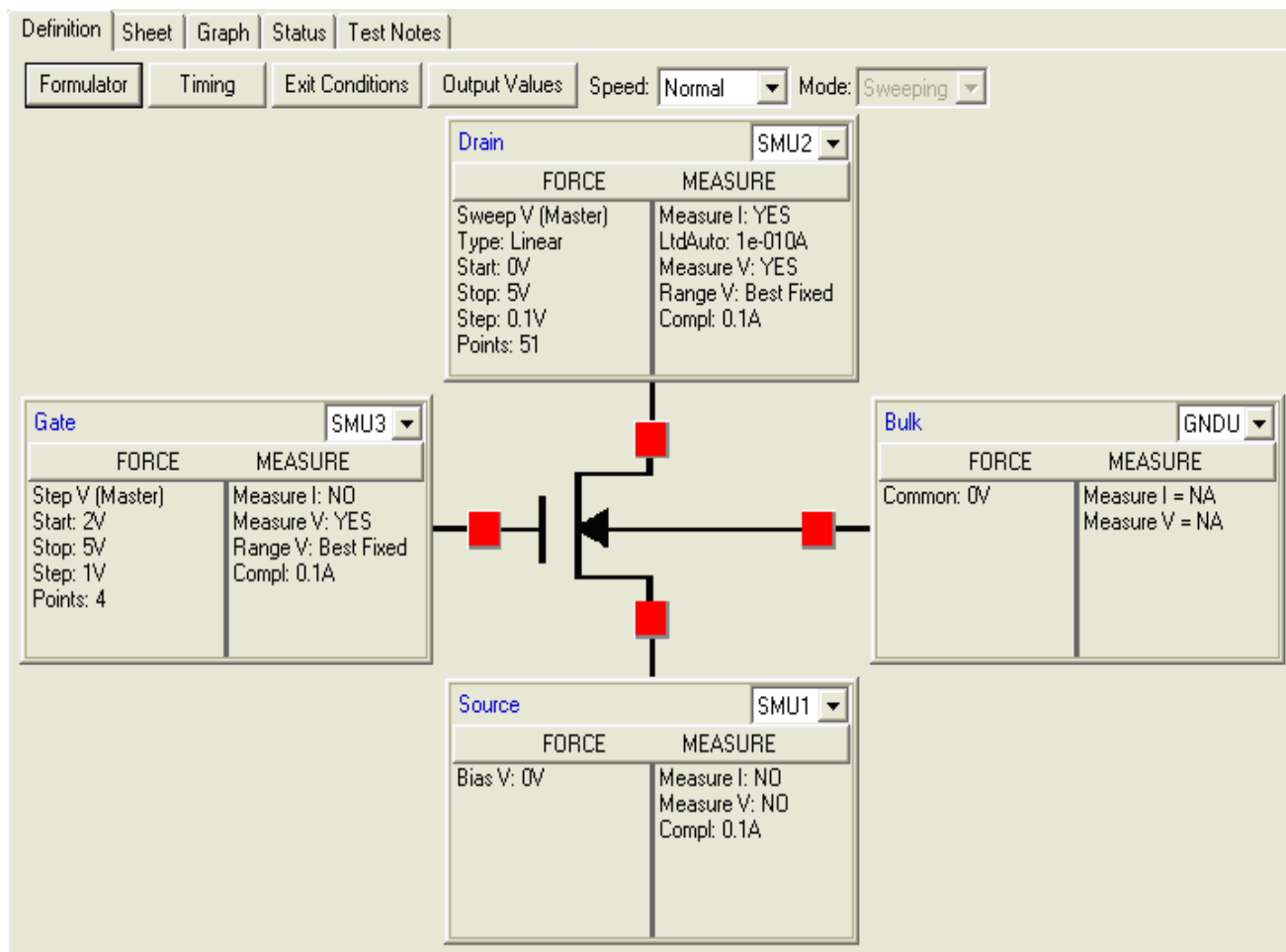


## 4-terminal n-MOSFET tests

By default, the following tests use three source-measure units (SMUs) and one ground unit (GNDU). It is also possible to use four SMUs, one for each device-under test (DUT) terminal (see [Figure 3-2](#)). For more information about the  $v_{ds}$ - $i_d$  Definition tab refer to the [Test definition, page 1-45](#).

Figure 3-2

### 4-terminal n-MOSFET tests



#### Description of 4-terminal n-MOSFET tests:

- $v_{ds}$ - $i_d$ : This test generates a family of I-V curves on a 4-terminal n-MOSFET and plots drain current versus drain voltage.
- $v_{tlin}$ : This test runs a typical linear curve fit to find the threshold voltage of a 4-terminal n-MOSFET and plots drain current vs. gate voltage.
- $subvt$ : This test runs an I-V sweep and calculates the sub-threshold voltage of a 4-terminal n-MOSFET and plots drain current vs. gate voltage.
- $v_{gs}$ - $i_d$ : This test runs an I-V sweep on the gate and calculates the threshold voltage using the Max GM method.
- $I_g$ - $v_g$ : This test runs a linear I-V sweep on the gate, plots gate voltage vs. gate current, and calculates the gate leakage current using formulator functions and a linear line fit.

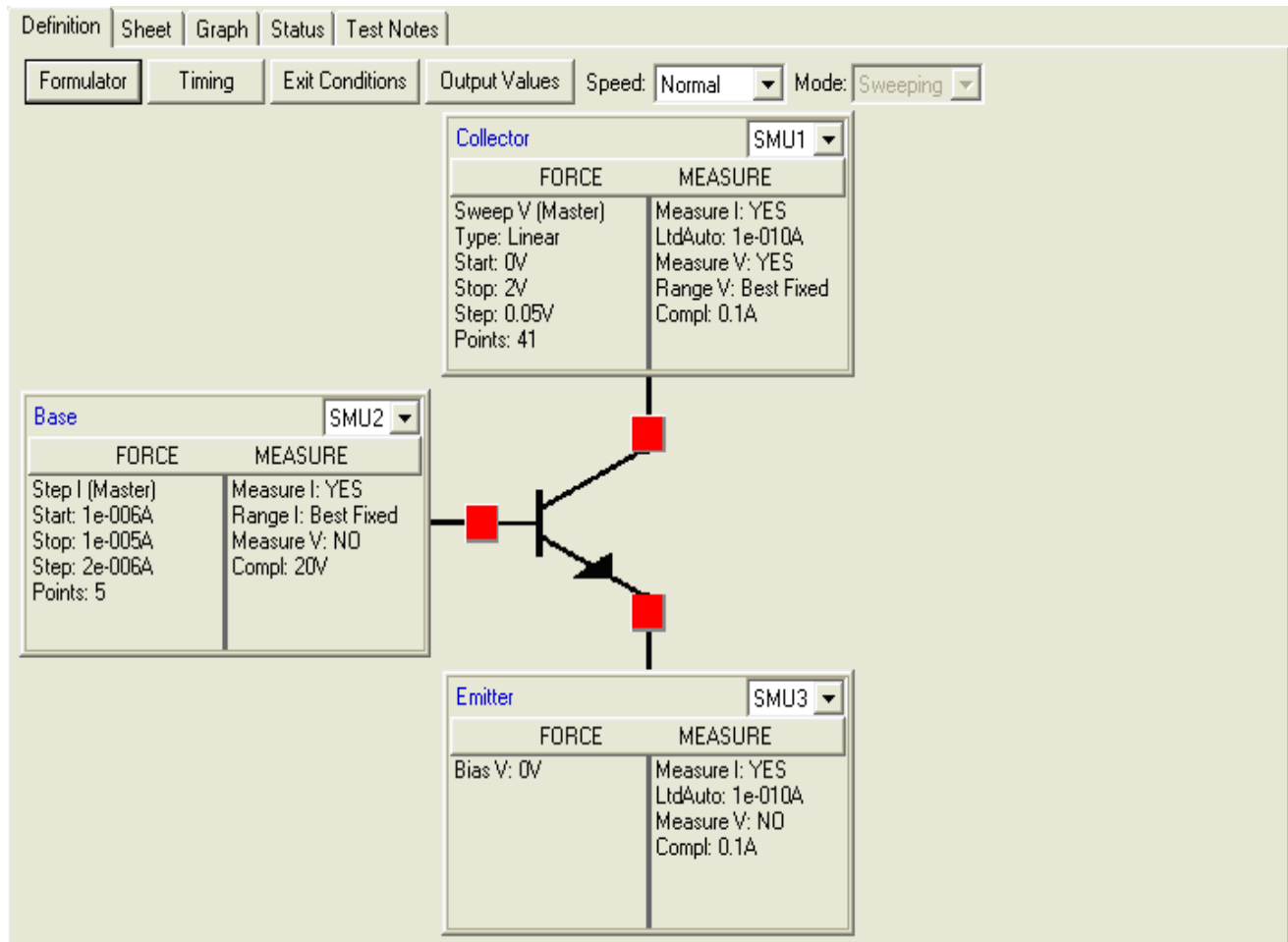


### Three terminal NPN BJT tests

The following tests require three SMUs (see [Figure 3-3](#)).

Figure 3-3

#### Three terminal NPN BJT tests



#### Description of three terminal NPN BJT tests:

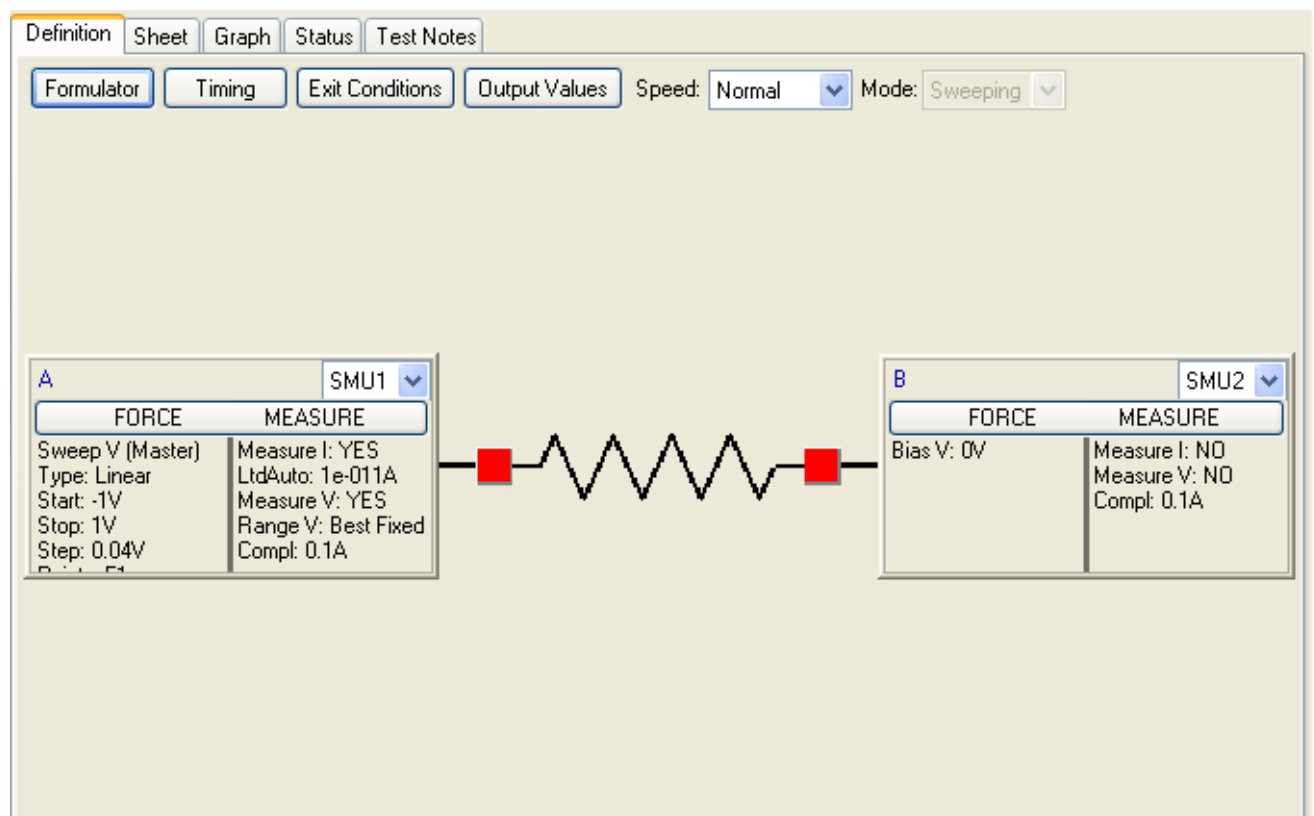
- vce-ic:** This test runs nested I-V sweeps to generate an n-p-n transistor collector family of curves. Collector current is plotted vs. collector voltage.
- gummel:** This test runs two voltage sweeps on the collector and base to produce a classic n-p-n transistor Gummel plot.
- vc-sat:** This test runs a voltage sweep on the collector, uses formulator functions to calculate ICSAT and VCSAT, and plots a collector I-V curve to show the n-p-n transistor saturation voltage.

### Two wire resistor test

By default, the following test uses two SMUs. It is also possible to use one SMU and the GNDU (see [Figure 3-4](#)).



Figure 3-4  
Two wire resistor test



#### Description of two wire resistor test:

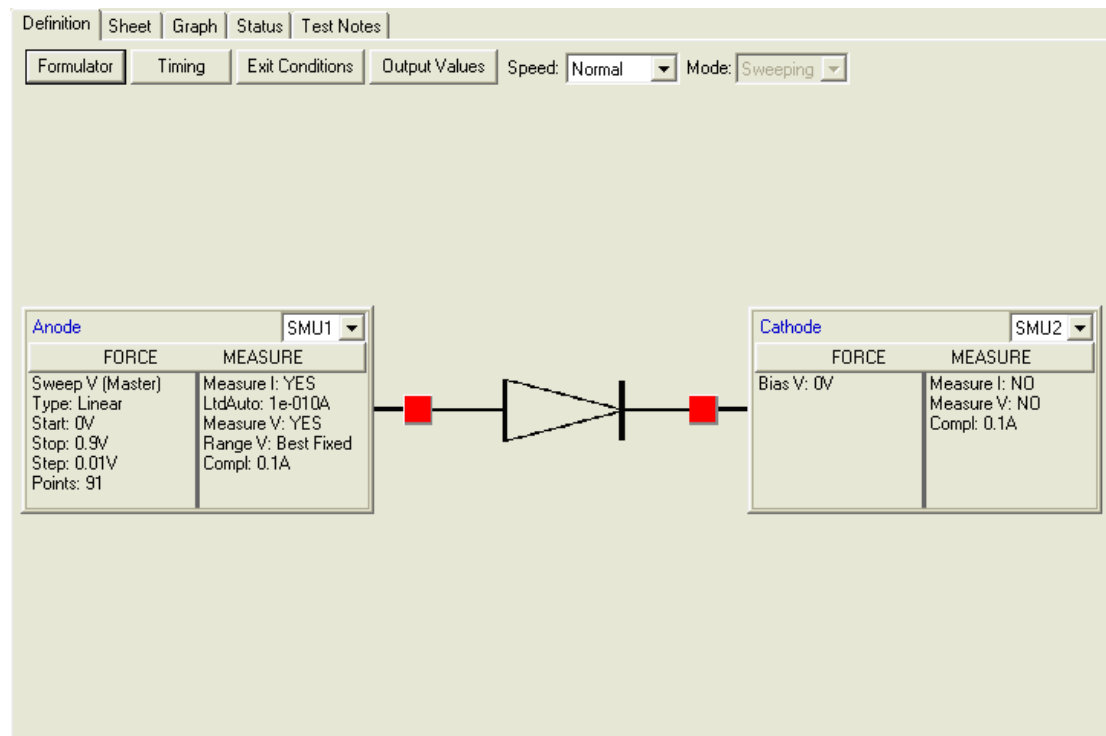
**res2t**: This test runs a simple negative-to-positive voltage sweep across a two-wire resistor and plots the current vs. voltage.

#### Diode tests

By default, these tests use two SMUs. It is also possible to use one SMU on the anode and the GNDU on the cathode (see [Figure 3-5](#)).



Figure 3-5  
Diode tests



#### Description of diode tests:

**vfd:** This test runs a linear forward I-V sweep through the anode, uses the formulator to calculate the exponential line fits, and plots anode current vs. anode voltage.

**vrđ:** This test runs a linear reverse I-V sweep through the anode, uses the formulator to calculate the exponential line fits, and plots anode current vs. anode voltage.

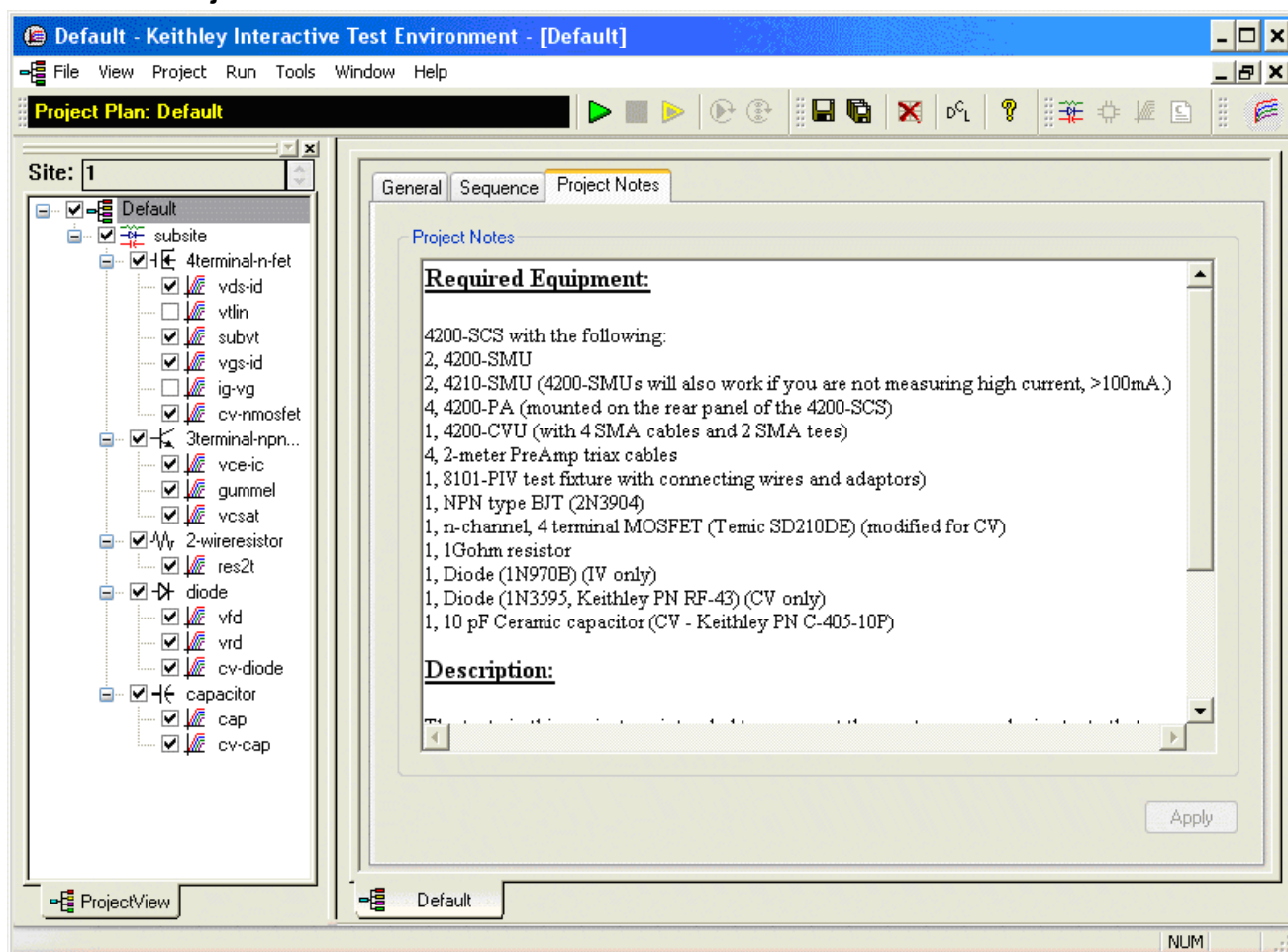
These tests serve as good examples on how to configure tests in the definition tab, how to use Formulator functions to perform common mathematical calculations and return them to the data sheet, and how to configure the graph to plot the data in a variety of ways.

All test parameters in the default project were written for standard discrete parts but can be easily modified for use with other discrete devices or devices on a semiconductor wafer.

To see exactly what discrete DUTs these tests were performed on, see the Default project notes by clicking on the default project tree node and then selecting the **Project Notes** tab (see [Figure 3-6](#)). Data sheets for the test DUTs used with the default project can be found on the Model 4200 Complete Reference webpage on the data sheets page.



Figure 3-6  
Project Notes tab



## Connections

The vast majority of I-V DUT connections are made by using mini triax cables (for SMU) or full triax-to-triax cables (for preamp), and connecting to the SMU/PA source and sense connectors on one end, and to your DUT on the other.

The SenseLO SMU connector is used only under special cases. For additional details and schematics on connecting DUTs to the [Connecting DUTs, page 1-40](#) or [Model 4200-SCS Reference Manual, Connections and Configuration, Section 4](#).

---

**NOTE** In general, the Model 4200-SCS utilizes BLACK triax cables for I-V testing, RED SMA cables for C-V testing, and WHITE SMA cables for pulse testing.

---

## Leveraging the default project

---

**CAUTION** It is strongly suggested that you do not modify the default project itself, but rather copy the entire project or individual test to another KITE project before making modifications.

---



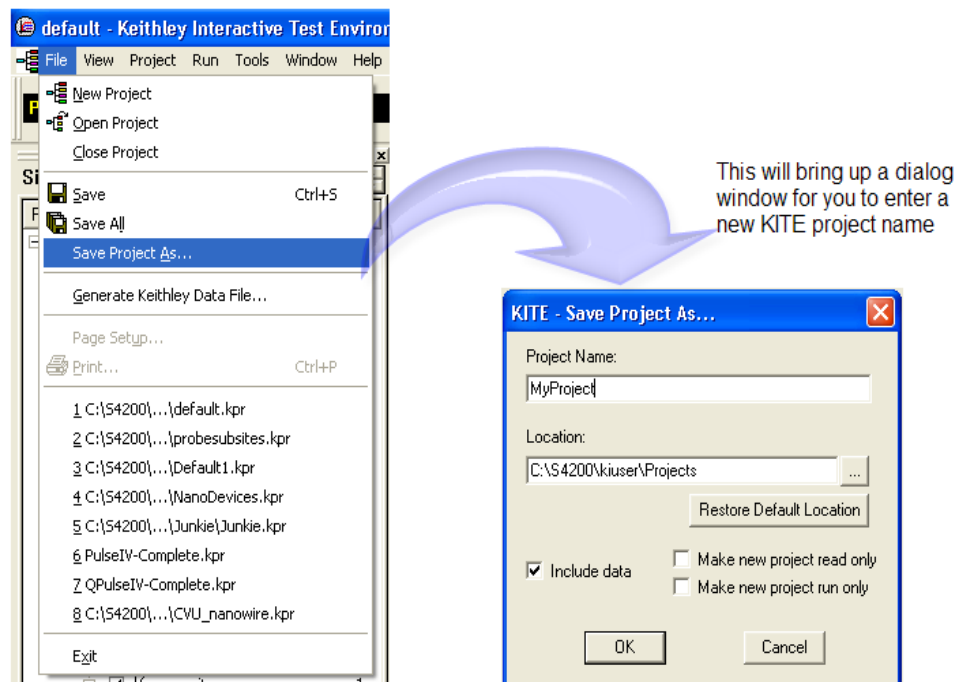
All modifications to any KITE Default project will be lost when upgrading the Model 4200-SCS software to a new version. All customer-created KITE projects and data will **NOT** be deleted or modified in any way when upgrading the Model 4200-SCS software or uninstalling KITE Interactive. See the Model 4200-SCS Release Notes (Installation Instructions) in the the Model 4200-SCS Complete Reference for details on all Default KITE projects.

## Copying entire KITE project

The entire Default project (or any KITE project for that matter) can be copied in its entirety in one easy step. After opening the default project, select the menu item **File, Save Project As...** (see [Figure 3-7](#)).

Figure 3-7

### Copying entire KITE project



After entering a new project name, click the **OK** button. This will copy the entire KITE project to another KITE project and will include all test definitions, formulas, graph settings, and data (by default).

Copying the default project first will preserve the original default project and allow you to add tests, delete tests, and modify tests without affecting the default project.

## Copying individual tests using the test library manager

Individual tests can be copied to test library and then copied into another project by performing the following steps (see [Figure 3-5](#)).

### To copy individual tests using the test library manager:

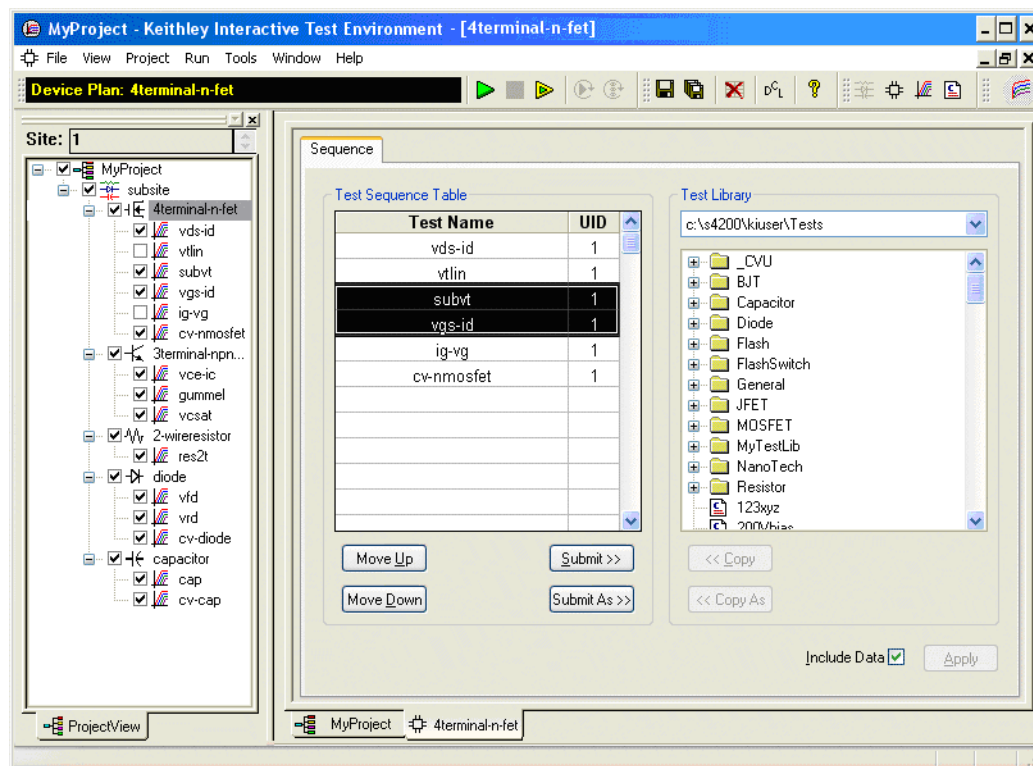
1. Double-click the device node that is the parent of the test you want to copy.
2. Select one or more tests from the Test Sequence Table (hold down the **Shift** or **Ctrl** key while clicking with the mouse to select multiple tests) (see [Figure 3-8](#)).
3. Click the **Submit** or **Submit As** buttons to submit the tests to a test library
4. Create a new KITE project or open a different KITE project
5. Double-click the device node you want to copy the test to.



6. Select one or more tests from the test library and copy or copy as the library tests to the KITE project.

See the [How to display and manage test results, page 2-24](#) or the [Reference Manual, Keithley Interactive Test Environment \(KITE\), page 6-1](#) for further details on managing KITE tests and projects.

Figure 3-8  
Selecting multiple tests



## Changing KITE startup behavior

To stop the Default project from opening automatically when starting KITE or to change the project that opens when starting KITE, perform the following steps (see figure [Figure 3-9](#)).

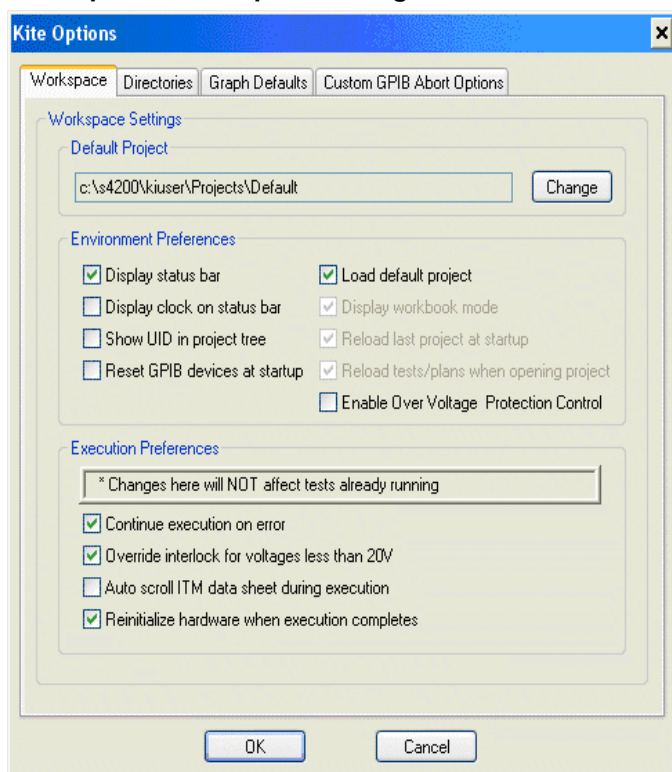
### To change KITE startup behavior:

1. Select the KITE **Tools... Options...** menu item.
2. To change which project opens when starting KITE, click the **Change** button and change the default KITE project.
3. To stop any default project from opening when starting KITE, uncheck the **Load Default Project** checkbox.

See the [Reference Manual, Customizing KITE, page 6-338](#) for further details on configuring KITE behavior.



Figure 3-9  
**KITE options workspace settings**





## How to perform a C-V test on my device

### KITE ITM configuration

KITE is the graphical user interface used to configure and run interactive test modules (ITMs). The workspace for KITE includes tabs for ITM configuration (Definition tab) and tabs for evaluating test results (Sheet, Graph, and Status tabs).

The following information explains how to configure an ITM that uses the Model 4200-CVU. The ITMs provided by Keithley Instruments are documented in [Reference manual, C-V project plans, page 15-28](#).

---

**NOTE** Details about KITE ITMs are provided in [Reference Manual, Keithley Interactive Test Environment \(KITE\), page 6-1](#). The KITE ITM information provided here is supplemental and pertains specifically to the Model 4200-CVU to configure ITMs.

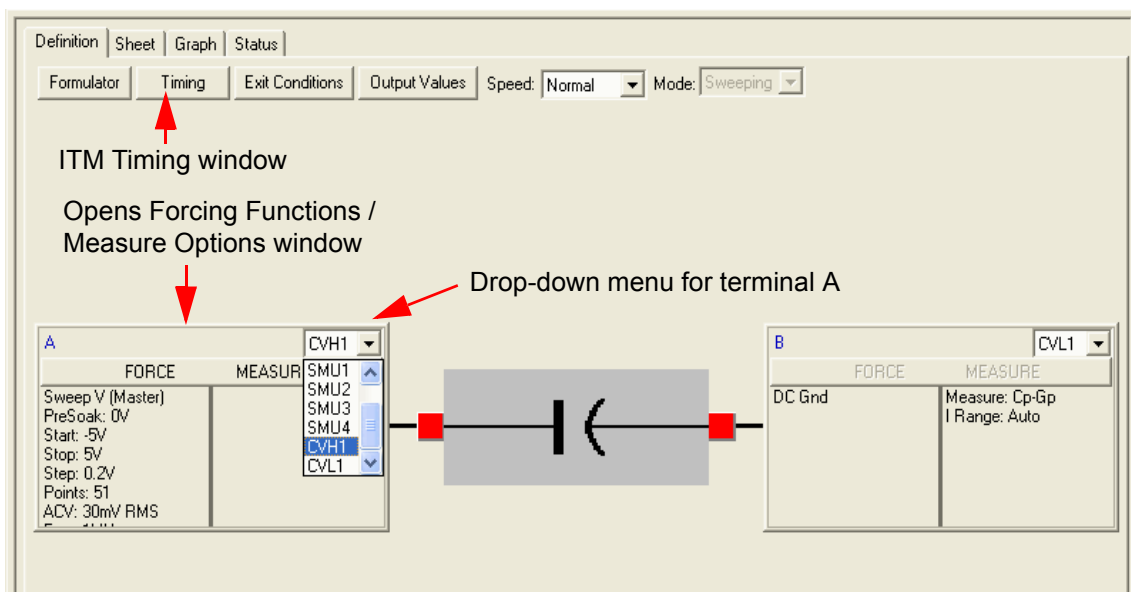
---

### Definition tab

#### Terminal settings

In the project navigator, double-click an **ITM** to open the Definition tab. [Figure 3-10](#) shows an example of a typical definition tab for a two-terminal device. When terminal A is set for CVH1, terminal B will be set for CVL1.

Figure 3-10  
Definition tab for an ITM



#### ITM timing

The ITM Timing window (see [Figure 3-11](#)) is used to set measurement speed and the test mode. The ITM Timing window is opened by clicking the **Timing** button located at the top of the definition tab (see [Figure 3-10](#)).



Figure 3-11

**ITM timing****A) Sweeping Mode selected**

The screenshot shows the 'ITM Timing' dialog box with the 'Sweeping Mode' tab selected. The 'Speed' section has 'Normal' selected. The 'Delay Factor' is 1 and the 'Filter Factor' is 1. The 'A/D Aperture Time' is set to 'Auto' and 'PLCs' is checked. The 'Sweep Delay' is 0.1 s and the 'Hold Time' is 2 s. The 'Interval' is 0 s, '#Samples' is 1, and 'Hold Time' is 0 s. A note states: 'NOTE: Remove all Sweeping/Stepping functions to allow Sampling Mode selection.' The 'SMU Power On Sequence' section has a list box and 'Move Up'/'Move Down' buttons. At the bottom, 'Timestamp Enabled' and 'Disable outputs at completion' are both checked. 'OK' and 'Cancel' buttons are at the bottom.

**B) Sampling Mode selected**

The screenshot shows the 'ITM Timing' dialog box with the 'Sampling Mode' tab selected. The 'Speed' section has 'Normal' selected. The 'Delay Factor' is 1 and the 'Filter Factor' is 1. The 'A/D Aperture Time' is set to 'Auto' and 'PLCs' is checked. The 'Interval' is 0.25 s, '#Samples' is 10, and 'Hold Time' is 1 s. The 'Sweep Delay' is 0 s and 'Hold Time' is 0 s. A note states: 'NOTE: Remove all Sweeping/Stepping functions to allow Sampling Mode selection.' The 'SMU Power On Sequence' section has a list box and 'Move Up'/'Move Down' buttons. At the bottom, 'Timestamp Enabled' and 'Disable outputs at completion' are both checked. 'OK' and 'Cancel' buttons are at the bottom.

**Speed**

There are four measurement speed settings:

- **Fast** Good choice for quick measurements where noise and settling time are not concerns.
- **Normal** Provides a good combination of speed and low noise, and is the best setting for most cases.
- **Quiet** Best choice when you need the lowest noise and most accurate measurements.
- **Custom** Allows you to fine-tune the timing parameters to meet a particular need. For details about this setting, refer to the [Reference manual, Timing window, page 6-134](#).

---

**NOTE** The above measurement speed selections can also be made from the speed drop-down menu located at the top of the Definition tab ([Figure 3-10](#)).

---

**Mode**

There are two test modes for the Model 4200-CVU: sweeping and sampling. The sweeping test mode applies to any ITM in which voltage or frequency varies with time. The sampling test mode applies to any ITM in which the forced voltage and frequency are static, with measurements made at timed intervals.

With the **Sweeping Mode** selected ([Figure 3-11A](#)), the sweep delay and hold time can be set. With the **Sampling Mode** selected ([Figure 3-11B](#)), the Interval, #samples, and hold time can be set. The setting ranges for these timing parameters are as follows:

- For the sweeping mode, **Sweep Delay** and **Hold Time** can be set from 0 to 999s. Use the drop-down menu to select **units** ( $\mu$ s, ms, or s).



- For the sampling mode, **Interval** and **Hold Time** can be set from 0 to 999s. Use the drop-down menu to select **units** ( $\mu$ s, ms, or s). The **# Samples** (measurements) can be set from 1 to 4096.

The [CVU ITM examples](#) provide details about the mode parameters.

---

**NOTE** The test mode (sweeping or sampling) can also be selected from the Mode drop-down menu located at the top of the definition tab ([Figure 3-10](#)).

---

### Timestamp and output disable

At the bottom of the ITM Timing window are controls for the timestamp and output disable:

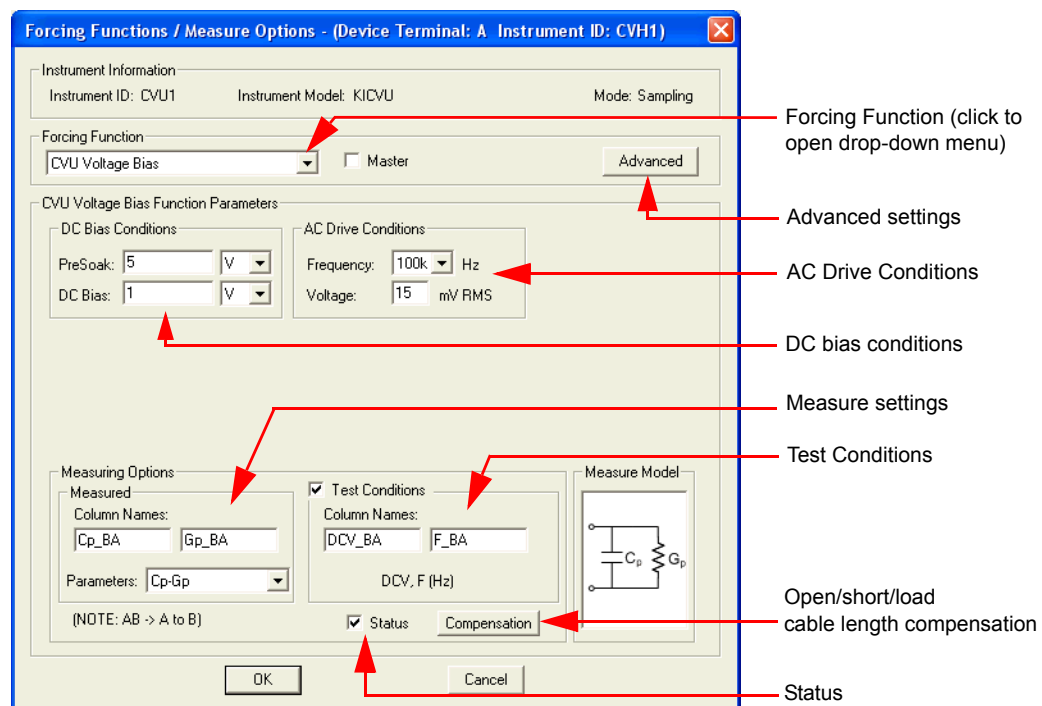
- With **Timestamp enabled** (checked), a timestamp for every measurement will be included in the Sheet tab.
- With **Disable outputs at completion** enabled (checked), the output will turn off (0 V) when the test is completed. When disabled, DC Bias voltage will remain at the last bias voltage level.

## Forcing functions and measure options

The Forcing Functions / Measure Options (FFMO) window is used to configure the force and measure options for the Model 4200-CVU. This window is opened by clicking the **FORCE MEASURE** bar for the CVH1 terminal in the definition tab (as shown in [Figure 3-10](#)). [Figure 3-12](#) shows an example of the FFMO window.

Figure 3-12

### Forcing function: CVU voltage bias (sampling mode)





## Selecting the forcing function

The forcing function options for the Model 4200-CVU are listed in [Table 3-1](#). After configuring the device terminals ([Terminal settings](#)), measurement speed ([Speed](#)), and the test mode ([Mode](#)), a forcing function can be selected from the **Forcing Function** drop-down menu (see [Figure 3-12](#)).

Table 3-1

### Forcing functions: DC bias settings and AC drive settings

Forcing Function	Test mode	DC bias condition		AC drive condition **		FFMO window example
CVU voltage bias	Sampling	PreSoak, DC bias	-30 V to 30 V	frequency	1 kHz to 10 MHz	<a href="#">Figure 3-12</a>
				voltage	10 mV to 100 mV	
CVU voltage sweep*	Sweeping	PreSoak, Start, Stop, Step	-30 V to 30 V	frequency	1 kHz to 10 MHz	<a href="#">Figure 3-17</a>
				voltage	10 mV to 100 mV	
CVU voltage list sweep	Sweeping	Data Points	1 to 4096	frequency	1 kHz to 10 MHz	<a href="#">Figure 3-19</a>
		Volts values, PreSoak	-30 V to 30 V	voltage	10 mV to 100 mV	
CVU frequency sweep (bias)	Sweeping	PreSoak, DC bias	-30 V to 30 V	start, stop frequency	1 kHz to 10 MHz	<a href="#">Figure 3-21</a>
				voltage	10 mV to 100 mV	
CVU frequency sweep (step)	Sweeping	PreSoak, Start, Stop, Step	-30 V to 30 V	start, stop frequency	1 kHz to 10 MHz	<a href="#">Figure 3-23</a>
				voltage	10 mV to 100 mV	

\* The CVU voltage sweep can be configured to perform a dual sweep (see [CVU Voltage Sweep](#)).

\*\* 4200-CVU lowest frequency is 10 kHz. 4210-CVU lowest frequency is 1 kHz.

## Setting the DC bias conditions

The DC bias conditions depend on which forcing function is presently selected. As shown in [Figure 3-12](#), settings include pre-soak voltage, DC bias, start voltage, stop voltage, step voltage, number of data points and volts values. [Table 3-1](#) lists the DC bias conditions (and setting range) for each forcing function.

## Setting the AC drive conditions

As shown in [Figure 3-12](#), the AC drive conditions include frequency (Hz) and voltage (mV RMS), and are summarized in [Table 3-1](#). Frequency can be set to the following values:

- 10 kHz to 90 kHz in 10 kHz steps
- 100 kHz to 900 kHz in 100 kHz steps
- 1 MHz to 10 MHz in 1 MHz steps

When performing a frequency sweep, the Model 4200-CVU will step through all the frequency points from start to stop. For example, if the start frequency is 800 kHz and the stop frequency is 3 MHz, the CVU will step through the following frequency points: 800 kHz, 900 kHz, 1 MHz, 2 MHz, 3 MHz.

## Measure settings

Referring to the Measure Settings area indicated in [Figure 3-12](#), use the drop-down menu for **Parameters** to select one of the following measurement options:

- **Z, Theta** Impedance and phase angle (degrees)
- **R+jX** Resistance and reactance
- **Cp-Gp** Parallel capacitance and conductance



- **Cs-Rs** Series capacitance and resistance
- **Cp-D** Parallel capacitance and dissipation factor
- **Cs-D** Series capacitance and dissipation factor

The **Column Names** appear in the data sheet (see the [Reference manual, Figure 15-175](#)). A default name (for example, Cs\_AB) can be changed by typing in a different name.

---

**CAUTION** Changing the measurement options will change the Column vector names, which may then cause any formulator functions to be erased. When using any of the Keithley Instruments-supplied tests or libraries, please leave the measurement option set to Cp-Gp.

---

### Test conditions

The DC V bias and drive frequency values used for the test will appear in the data sheet when **Test Conditions** (shown in [Figure 3-12](#)) is enabled (checked). The column names appear in the data sheet (see the [Reference manual, Figure 15-175](#)). A default name (for example, F\_AB) can be changed by typing in a different name.

### Advanced settings (terminal properties)

The AC drive voltage and DC bias voltage can be applied to either the CVH1 terminal or the CVL1 terminal. Clicking the **Advanced** button (shown in [Figure 3-12](#)) opens the Model 4200-CVU terminal properties window shown in [Figure 3-13](#).

Figure 3-13  
**CVU terminal properties window**

By default (reset), AC drive is applied to the CVH1 terminal and the current measurement is made at the CVL1 terminal. A drop-down menu sets the range for the current measurement (Auto, 1μA, 30μA, or 1mA). The terminal properties can be toggled by clicking one of the other AC radio buttons.

By default (reset), DC bias is also applied to the CVH1 terminal. DC bias can instead be applied to the CVHL terminal by clicking one of the other DC radio buttons.

[Figure 3-14](#) shows the four possible configurations for terminal properties.



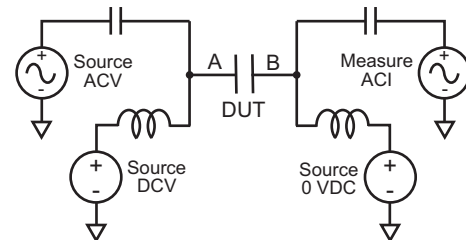
Figure 3-14

**Advanced settings for terminal properties****Configuration A (default)**

Use the following settings to source AC drive voltage and DC bias voltage to terminal A and measure AC current at terminal B:

	AC		DC	
	Source V	Measure I	Source V	Offset
CVH1 (A)	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
CVL1 (B)	<input type="radio"/>	Auto <input type="button" value="v"/>	<input type="radio"/>	0 V

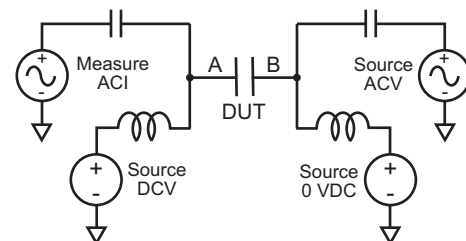
Simplified Test Circuit:

**Configuration B**

Use the following settings to source AC drive voltage to terminal B, source DC bias voltage to terminal A, and measure AC current at terminal A:

	AC		DC	
	Source V	Measure I	Source V	Offset
CVH1 (A)	<input type="radio"/>	Auto <input type="button" value="v"/>	<input checked="" type="radio"/>	<input type="radio"/>
CVL1 (B)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	0 V

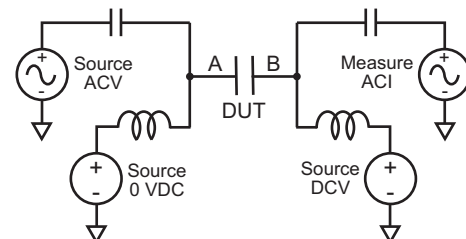
Simplified Test Circuit:

**Configuration C**

Use the following settings to source AC drive voltage to terminal A, source DC bias voltage to terminal B, and measure AC current at terminal B:

	AC		DC	
	Source V	Measure I	Source V	Offset
CVH1 (A)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	0 V
CVL1 (B)	<input type="radio"/>	Auto <input type="button" value="v"/>	<input checked="" type="radio"/>	<input type="radio"/>

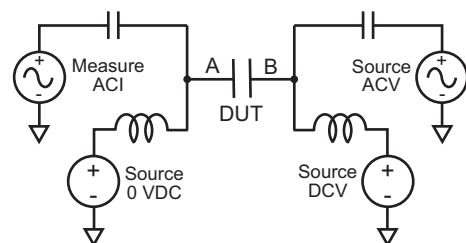
Simplified Test Circuit:

**Configuration D**

Use the following settings to source AC drive voltage and DC bias voltage to terminal B, and measure AC current at terminal A:

	AC		DC	
	Source V	Measure I	Source V	Offset
CVH1 (A)	<input type="radio"/>	Auto <input type="button" value="v"/>	<input type="radio"/>	0 V
CVL1 (B)	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Simplified Test Circuit:

**Status**

With Status enabled (as shown in [Figure 3-12](#)), the following errors will be reported in the sheet and graph tabs when a measurement fault occurs:



- ABB fails to lock
- AC voltage high overflow
- AC voltage low overflow
- AC current overflow

For more information refer to the [Reference manual, CVU measurement status, page 15-161](#).

## Compensation

After making connections for the test, connection compensation must be performed and enabled before running test. See the [Reference manual, Connection compensation, page 15-17](#) to perform connection compensation.

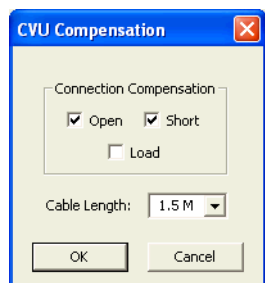
The CVU compensation window (shown in [Figure 3-15](#)) is used to enable (check) **Open**, **Short**, and **Load** compensation. The window is opened by clicking the **Compensation** button in the force-measure window.

**Cable length setting is to be made from the compensation window:**

- **0 M** = Disables compensation
- **1.5M** = 1.5 meters
- **3 M** = 3.0 meters
- **Custom** = Cable length coefficients measured by the user using the **Tools > Connection Compensation** dialog box.

Figure 3-15

### CVU compensation window



## CVU ITM examples

### CVU Voltage Bias

[Figure 3-12](#) shows an example of a FFMO window with **CVU Voltage Bias** selected as the forcing function to measure Cp-Gp. The Sampling test mode must be selected for this test (see [Figure 3-11](#)).

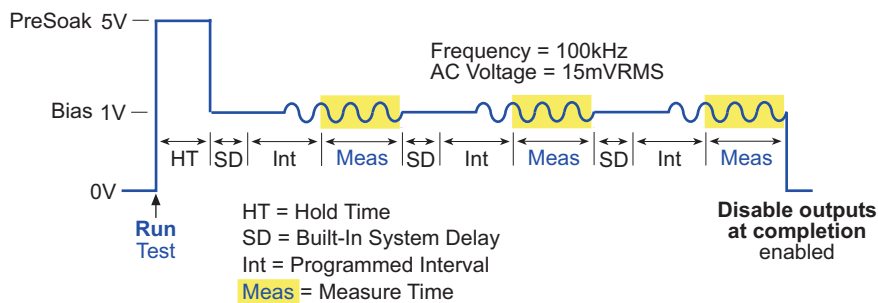
When this test is run (see [Figure 3-16](#)), the following force-measure sequence occurs:

1. The DC source goes to the PreSoak voltage of 5 V for the hold time period.
2. The DC source goes to the DC bias voltage of 1 V.
3. After the built-in system delay and Interval, the Model 4200-CVU performs a measurement. The AC test signal is applied just before the start of the measurement. AC drive is turned off after the measurement is completed.
4. Step 3 is repeated for every sample.

The number of samples (measurements), interval between each measurement, hold time and output disable are set from the [ITM timing](#) window for sampling.



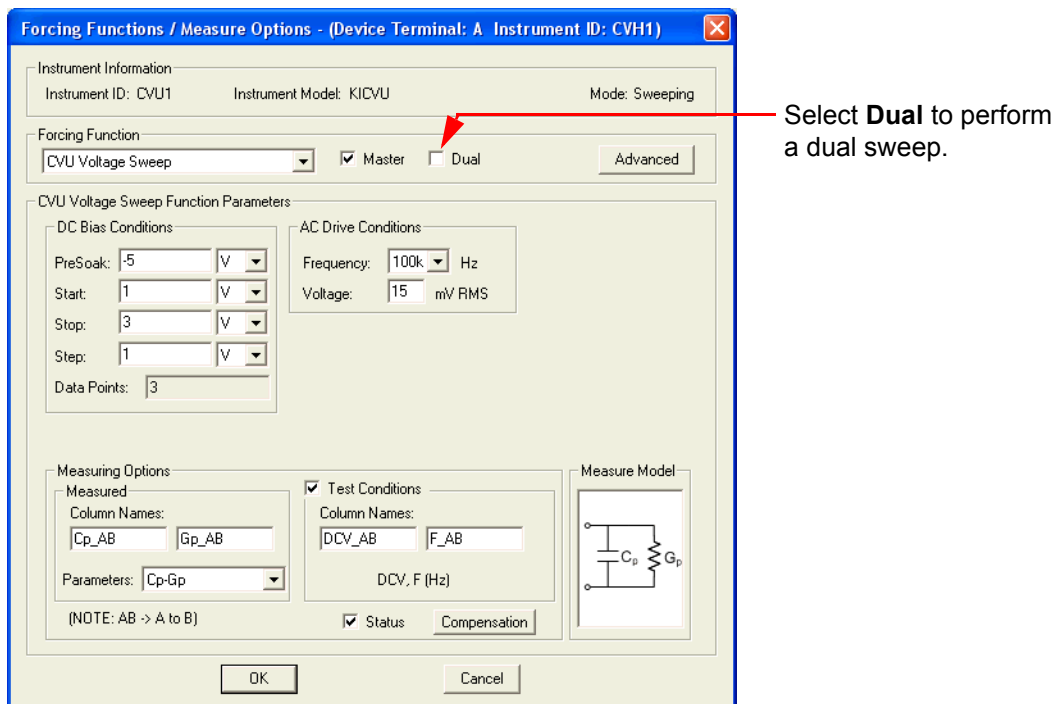
Figure 3-16  
**CVU Voltage Bias output**



## CVU Voltage Sweep

Figure 3-17 shows an example of a FFMO window with **CVU Voltage Sweep** selected as the forcing function to measure Cp-Gp. The **Sweeping** test mode must be selected for this test (see Figure 3-11).

Figure 3-17  
**Forcing Function: CVU Voltage Sweep**



When this test is run (see Figure 3-18), the following force-measure sequence occurs:

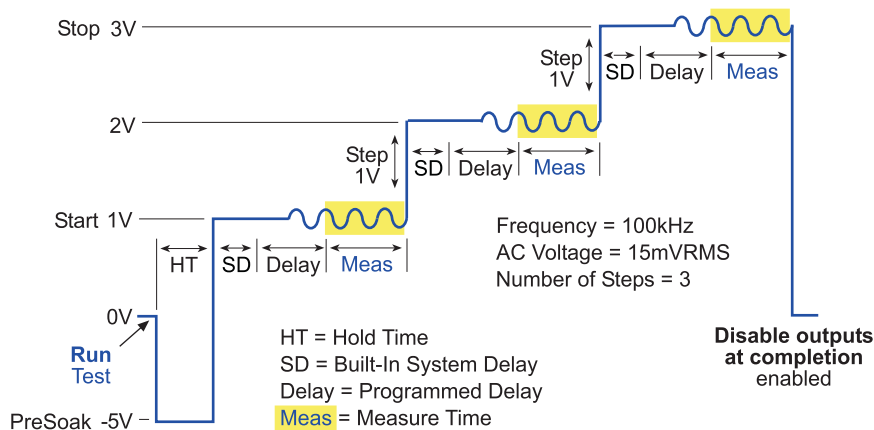
1. The DC source goes to the PreSoak voltage of -5 V for the hold time period.
2. The DC bias voltage goes to the first step of the sweep (1 V).
3. After the built-in system delay and programmed delay, the Model 4200-CVU will perform a measurement. The AC test signal is applied just before the start of the measurement. AC drive is turned off after the measurement is completed.
4. Steps 2 and 3 are repeated for the 2 V and 3 V DC bias voltage steps. The Sweep delay repeats at the beginning of each subsequent step.

The sweep delay, hold time and output disable are set from the [ITM timing](#) window for sweeping.



A dual CVU Voltage Sweep can be performed by selecting (checking) **Dual** in the Forcing Functions / Measure Options window. After the last (stop) step is measured, the sweep will continue in the reverse direction. For the force settings shown in Figure 3-17, the dual sweep will step as follows. 1 V, 2 V, 3 V, 3 V, 2 V, and 1 V. The number of measurements will double to six.

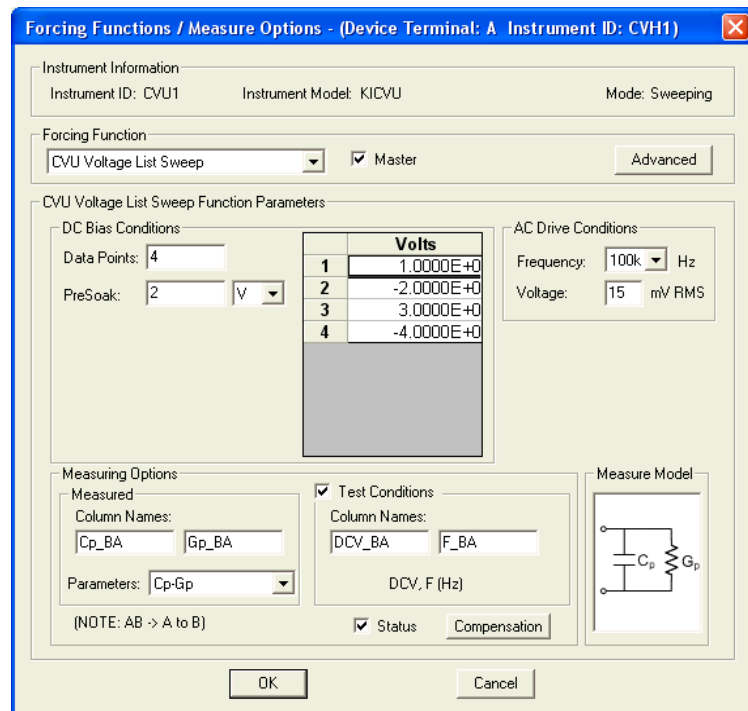
Figure 3-18  
**CVU Voltage Sweep output**



## CVU Voltage List Sweep

Figure 3-19 shows an example of a FFMO window with **CVU Voltage List Sweep** selected as the forcing function to measure Cp-Gp. The **Sweeping** test mode must be selected for this test (see Figure 3-11).

Figure 3-19  
**Forcing Function: CVU Voltage List Sweep**



When this test is run (see Figure 3-20), the following force-measure sequence occurs:

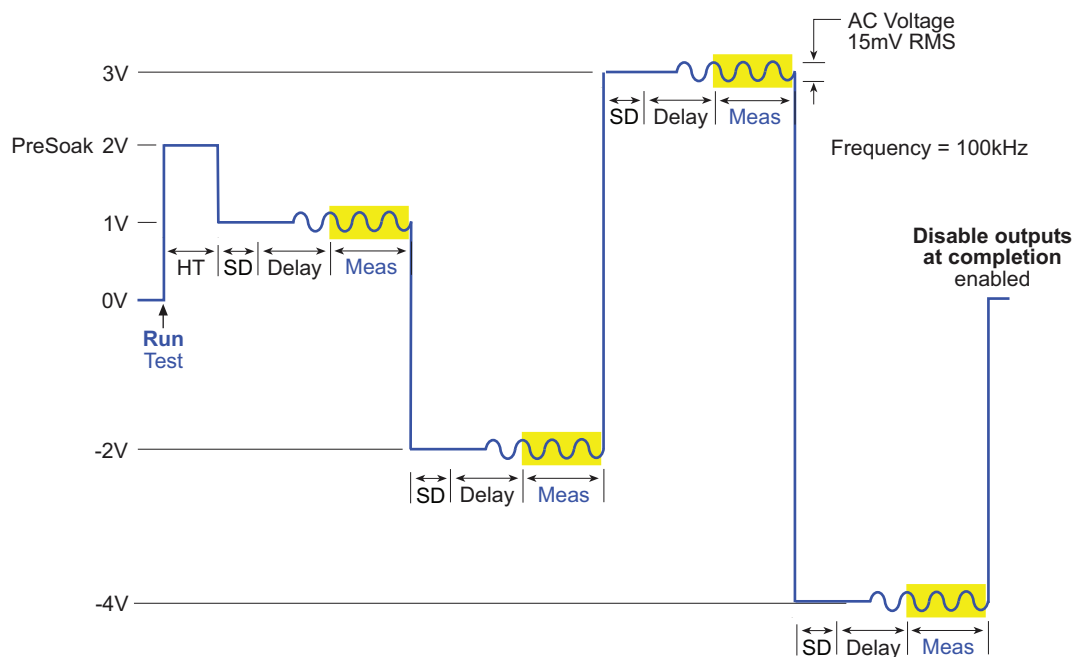


1. The DC source goes to the PreSoak voltage of 2 V for the hold time period.
2. The DC Bias goes to the first sweep point voltage (1 V).
3. After the built-in system delay and programmed delay, the Model 4200-CVU will perform a measurement. The AC test signal is applied just before the start of the measurement. AC drive is turned off after the measurement is completed.
4. Steps 2 and 3 are repeated for the -2 V, 3 V and -4 V DC bias voltages. The hold time delay repeats at the beginning of each subsequent step.

The sweep delay, hold time and output disable are set from the [ITM timing](#) window for sweeping.

Figure 3-20

### CVU Voltage List Sweep output

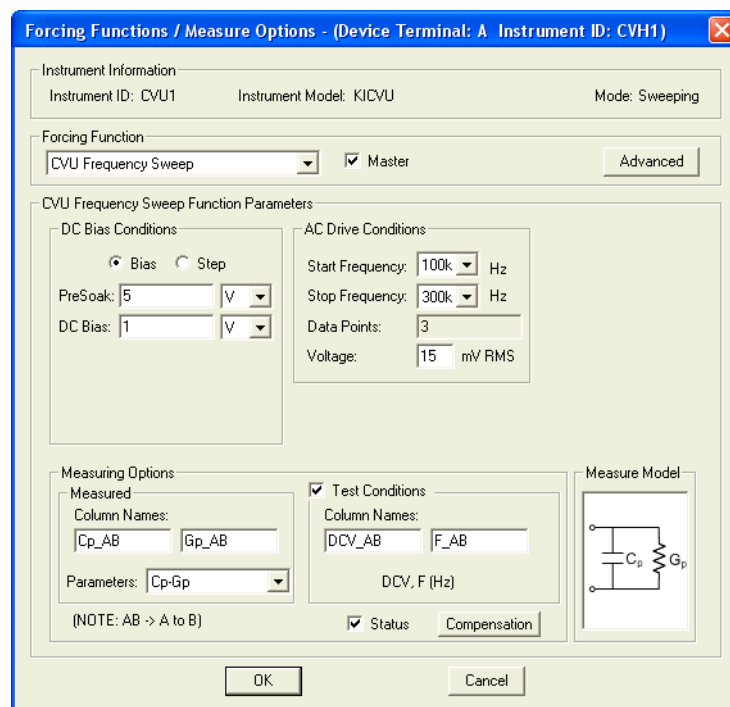


### CVU Frequency Sweep (bias)

[Figure 3-21](#) shows an example of a FFMO window with **CVU Frequency Sweep (bias)** selected as the forcing function to measure  $C_p$ - $G_p$ . The **Sweeping** test mode must be selected for this test (see [Figure 3-11](#)).



Figure 3-21

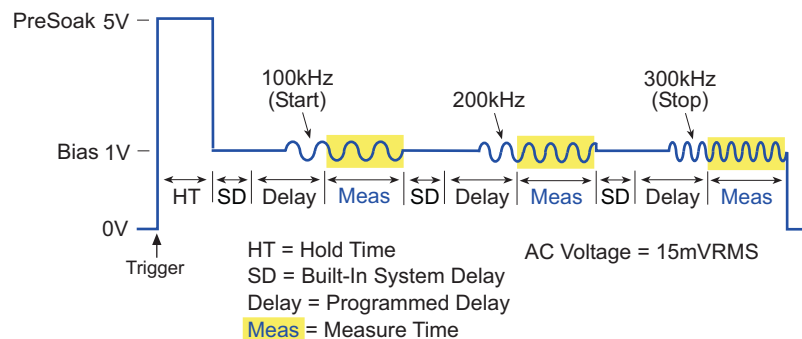
**Forcing Function: CVU Frequency Sweep (bias)**

When this test is run (see Figure 3-21), the following force-measure sequence occurs:

1. The DC source goes to the PreSoak voltage of 5 V for the hold time period.
2. The DC bias goes to 1 V for the system delay and programmed delay time periods.
3. The Model 4200-CVU performs a measurement for the first frequency point (100 kHz). The AC test signal is applied just before the start of the measurement. AC drive is turned off after the measurement is completed.
4. Step 3 is repeated for the other frequency points. The system delay and programmed delay are repeated for each subsequent measurement.

The sweep delay, hold time and output disable are set from the [ITM timing](#) window for sweeping.

Figure 3-22

**CVU Frequency Sweep (bias) output**



## CVU Frequency Sweep (step)

Figure 3-23 shows an example of a FFMO window with **CVU Frequency Sweep** (step) selected as the forcing function to measure Cp-Gp. The **Sweeping** test mode must be selected for this test (see Figure 3-11).

Figure 3-23  
Forcing Function: CVU Frequency Sweep (step)

The screenshot shows the 'Forcing Functions / Measure Options' dialog box for a device terminal A and instrument ID CVH1. The mode is set to 'Sweeping'. The 'Forcing Function' is 'CVU Frequency Sweep' and 'Master' is checked. The 'Advanced' button is visible. The 'CVU Frequency Sweep Function Parameters' section includes 'DC Bias Conditions' (PreSoak: -1 V, Start: 0 V, Stop: 2 V, Step: 1 V, Data Points: 3) and 'AC Drive Conditions' (Start Frequency: 100k Hz, Stop Frequency: 200k Hz, Data Points: 2, Voltage: 30 mV RMS). The 'Measuring Options' section shows 'Measured' as 'Cp\_AB' and 'Gp\_AB', 'Parameters' as 'Cp-Gp', and 'Test Conditions' as 'DCV\_AB' and 'F\_AB'. The 'Measure Model' section shows a circuit diagram of a capacitor Cp in parallel with a conductance Gp. The 'Status' and 'Compensation' buttons are at the bottom.

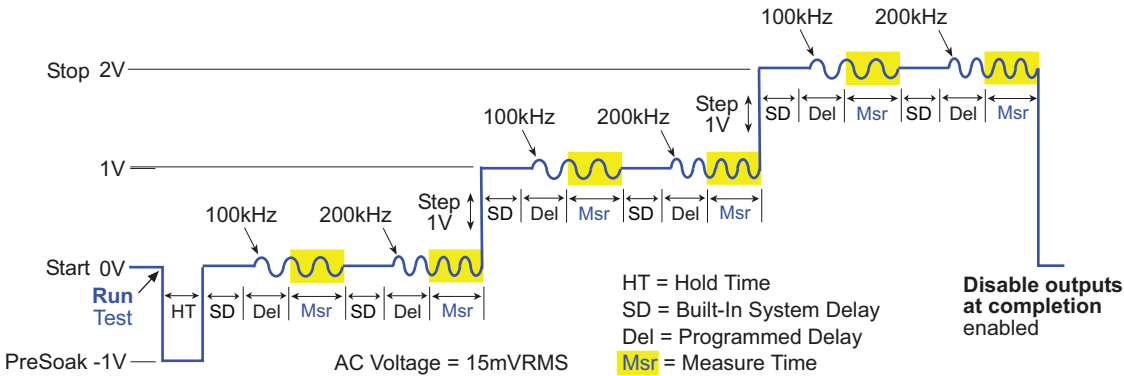
When this test is run (see Figure 3-24), the following force-measure sequence occurs:

1. The DC source goes to the PreSoak voltage of -1 V.
2. After the hold time, DC bias goes to 0V.
3. After the system delay and the programmed delay, the Model 4200-CVU performs a measurement for the 100 kHz frequency point. The AC signal is applied just before the start of the measurement.
4. After another system delay and programmed delay, a measurement is performed for the 200 kHz frequency point.
5. DC bias goes to 1 V.
6. Steps 3 and 4 are repeated.
7. DC bias goes to 2 V.
8. Steps 3 and 4 are repeated.

The sweep delay, hold time and output disable are set from the [ITM timing](#) window for sweeping.



Figure 3-24  
CVU Frequency Sweep (step) output





# How to perform a Pulsed I-V test on my device

## Introduction (PIV-A and PIV-Q)

Pulse IV is used in addition to DC IV test results to address two DUT behaviors: self heating (also called joule heating) and transient charging. For RF Transistors, especially those implemented with compound semiconductor materials, these two effects are called dispersion. The self heating and charging effects cause the DC and Pulse IV responses to differ.

Pulse IV addresses self heating by permitting the use of a low duty cycle, <0.1%, pulses to virtually eliminate heating within the DUT. Pulse IV addresses the charging effects by using pulse widths short enough so that charges cannot be sufficiently mobile within the pulse.

---

**NOTE** *The UTMs used for Pulse IV tests are described in the following paragraphs. These UTMs control all instrumentation for these applications. The pulse generator and scope cards can also be used as stand-alone instruments.*

*Reference Manual, Pulse Source-Measure Concepts, page 11-1 explains front panel operation and provides remote programming information for the pulse generator and scope. For remote programming, the pulse generator card uses LPTLib functions, while the scope card uses kscopeulib UTMs.*

*Reference Manual, Pulse Projects for Models 4200-PIV-A and 4200-PIV-Q, page 12-1 provides additional information about projects for the PIV-A and PIV-Q packages.*

*Reference Manual, Models 4220-PGU, 4225-PMU, and 4225-RPM, page 16-1 provides information on using the Models 4220-PGU and 4225-PMU to perform charge pumping tests.*

---

## What is Pulse IV

Pulse IV provides a user with the capability of running parametric curves on devices using pulsed rather than DC signals. A pulse source with a corresponding pulse measurement can be used in two general ways.

The first method is to provide DC-like parametric tests, where the measurement happens during the flat, settled part of the pulse. Typical tests are IV sweeps, such as a  $V_{ds}$ - $I_d$  family of curves or a  $V_{gs}$ - $I_d$  curve used for  $V_t$  extraction.

The second method is transient testing, where a single pulse waveform is used to investigate time varying parameter(s). An example of this second case would be using a single pulse waveform to investigate the  $I_d$  degradation versus time due to charge trapping or self-heating.

## Why use Pulse IV

Both methods of Pulse IV (PIV) testing listed above are used to overcome or study the effects of self heating (joule heating) and for time-domain studies, such as transient charge trapping in the DUT. The pulse and pulse IV testing is increasingly important in semiconductor research, device and process development.

This section will focus on the DC-like IV sweep capability of the PIV-A package, although other types of pulse testing are possible, such as charge pumping, single pulse charge trapping, AC stress, and non-volatile memory testing. Because charge pumping and floating gate memory testing use a pulse source with DC measure, these methods are not using pulse IV (pulse source with pulse measure) capabilities.



## What PulseIV Packages are available for the Model 4200-SCS

**PIV-A Package** – The Model 4200-PIV-A package provides pulse IV self heating for CMOS SOI for  $\leq 45\text{nm}$  technology node or any device that may benefit from low duty cycle pulsed IV testing to reduce the amount of power provided to the DUT during the test. The PIV-A package utilizes bias tees to permit both DC and pulse IV tests without re-cabling and pulses the DUT gate while DC biasing the DUT drain. See [Pulse IV for CMOS:Model 4200-PIV-A](#) for details about using the PIV-A package.

**PIV-Q Package** – The Model 4200-PIV-Q package provides higher power pulsing than the PIV-A package, while also permitting voltage pulsing from a non-zero bias, or quiescent point. The PIV-Q package provides voltage pulses to both the DUT gate and drain simultaneously. The PIV-Q package is appropriate for pulse IV testing of LDMOS and compound semiconductor FETs (HEMT, pHEMT) and other devices that require two channels of voltage pulsing, such as some HBTs. The PIV-Q package also provides DC tests without re-cabling. See [Q-Point Pulse IV – Model 4200-PIV-Q](#) for details about using the PIV-Q package.

## Pulse IV for CMOS:Model 4200-PIV-A

### What is the PIV-A PulseIV Package

The PIV-A package is an optional factory-installed kit to the Model 4200-SCS. The focus for the PIV-A package is testing lower power CMOS transistors that exhibit self-heating or charge trapping effects. Self-heating has been an issue for some higher power devices, but is emerging as a problem for lower power devices based on smaller dimensions and silicon-on-insulator (SOI) technology, where it is more difficult for the heat generated by the transistor to leave its immediate surroundings.

In addition to smaller dimensions, high k materials are being considered to greatly lower gate leakage current for future transistor technology. Unfortunately, these high k materials and related integration processes are not yet perfected and have both interface and bulk lattice imperfections that can cause charges to be trapped.

Both the charge trapping and self-heating effects can be largely avoided by using pulse IV instead of DC parametric testing.

To accomplish pulse IV testing of CMOS transistors, the PIV-A package consists of the following:

- Model 4205-PG2 Dual channel voltage pulse generator
- Model 4200-SCP2 Dual channel oscilloscope
- Pulse IV Interconnect Model 4205-Remote Bias Tees (RBTs) to combine both DC and pulse signals
- Pulse IV software – Projects and test routines for testing of CMOS transistors, including cable compensation and load-line algorithms to provide DC-like sweep results

### Target applications and test projects for PIV-A

The PIV-A package includes test projects that address the most common parametric transistor tests:  $V_{ds}\text{-id}$  and  $V_{gs}\text{-id}$ . These tests are provided in both DC and Pulse modes, allowing correlation between the two test methods, and have been configured for testing leading edge, lower-power CMOS devices. These tests, as well as initialization steps for scope auto-calibration and cable compensation, are included in a single Model 4200-SCS test project, Pulse-IV-Complete.

There is another Pulse IV test project, Demo-PulseIV. This demo project is a subset of PulseIVComplete and is intended for demonstrating the Pulse IV capabilities using a packaged demonstration DUT.

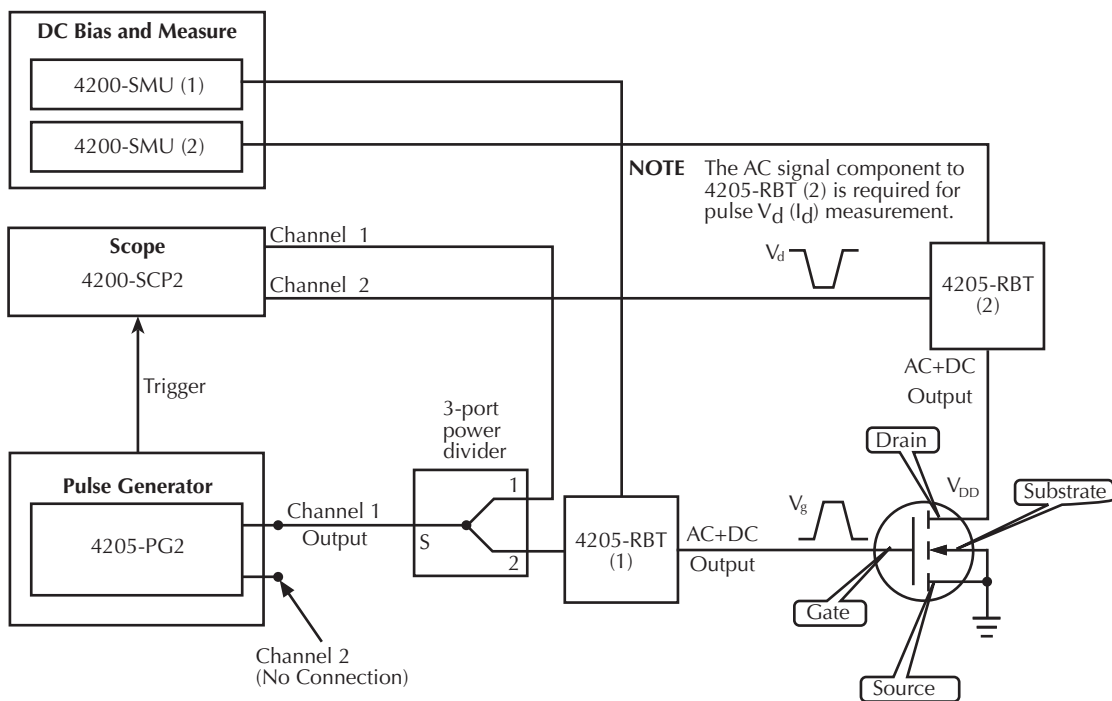


**NOTE** The user test modules (UTMs) used for Pulse IV tests are described in the following paragraphs. These UTMs control all instrumentation for these applications. The pulse generator and scope cards can also be used as stand-alone instruments. [Reference Manual, Pulse Source-Measure Concepts, page 11-1](#) explains front panel operation and provides remote programming information for individual control of the pulse generator and scope. For remote programming, the pulse generator card uses LPTLib functions, while the scope card uses kscopeulib UTMs.

## 4200-PIV-A test connections

The block diagram for PIV-A testing is shown in [Figure 3-25](#), and the hardware connections are shown in [Figure 3-26](#). A side view of the scope card is provided in [Figure 3-27](#) to show the adapters.

Figure 3-25  
Pulse IV—hardware setup block diagram



### Supplied interconnect parts

The interconnect parts listed in [Table 3-2](#) are supplied with the PIV-A package.



Table 3-2  
Supplied interconnect parts for Model 4200-PIV-A

Qty	Description	Comment
1	4.25 in / 10.8 cm white SMA cable	Interconnect for trigger
2	6 in / 15 cm white SMA cables	Interconnect between RBT and prober manipulator
2	13 in / 33 cm white SMA cables	Interconnect between RBT and prober manipulator (optional)
3	6.6 ft / 2 m white SMA cables	Interconnect between SCP2, pulse generator and RBTs
4	6.6 ft / 2 m black Triax cables	Interconnect between SMUs and RBTs
2	SMA female to BNC Male	Adapt SCP2 BNC channels to SMA
1	SMA female to SMB plug	Adapt SCP2 SMB trigger to SMA
1	Power Divider, Male/Female/Male	Connects to Gate side RBT AC IN connector

Supplied tools

The following tools are supplied with the Model 4200-SCS or PIV-A package:

- #1 Phillips screwdriver
- Torque wrench, 8 in / lb, with 5 / 16 in head installed

**NOTE** The various adapters, cables and hardware used for the pulse projects are shown in the [Reference manual, Figure 11-35](#).

Figure 3-26  
Pulse IV—hardware connections

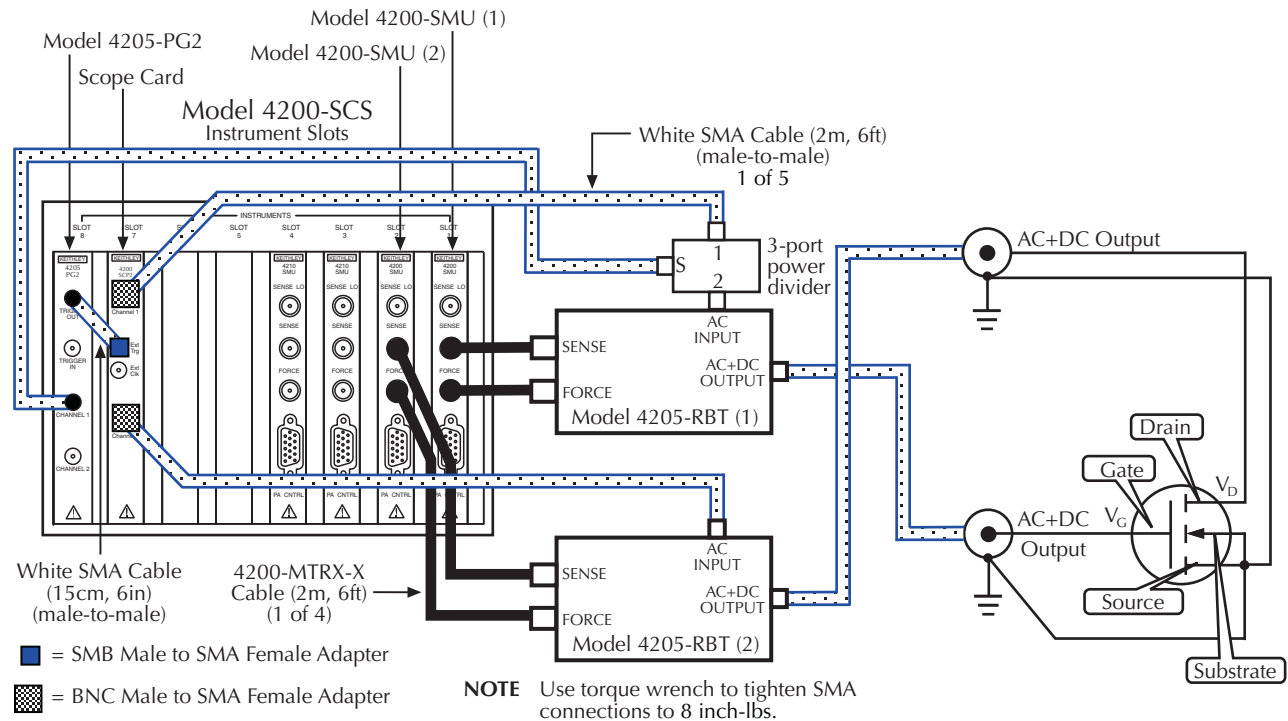
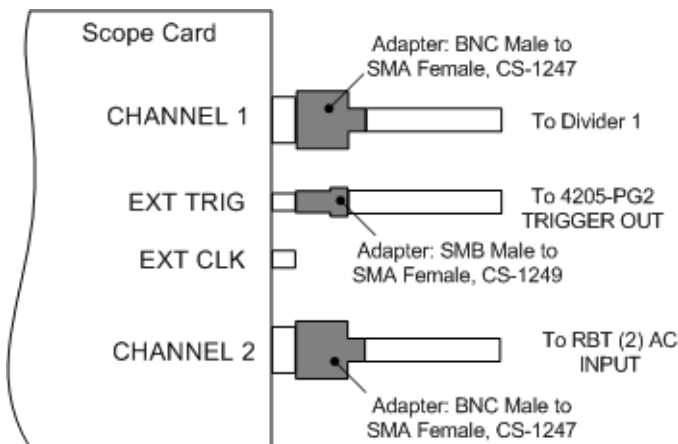




Figure 3-27  
Side view of scope card connections



### Model 8101- PIV test fixture

The Model 4200-PIV-A includes a test fixture and DUTs to verify proper PIV-A setup and operation, and is also useful for troubleshooting.

The test fixture, Model 8101-PIV (shown in [Figure 3-28](#)), has two electrically separate sockets to support testing three and four leaded devices. The lower socket, located near the latch, is for DC testing with SMUs and uses four Triax connectors. The upper, or back, socket, located near the hinge, is for Pulse IV or Pulse IV + DC testing and uses two SMA connectors.

The SMU socket has a triax connector for each of the four DUT pins. The pulse socket uses only two SMA connectors, where the DUT source and bulk connections are connected to ground (SMA coax shield) and are optimized for use with the Model 4200-PIV-A package.

The fixture may also be used with the Model 4200-PIV-Q package, but higher power testing, either DC or pulse IV, requires additional care to prevent damage to the included DUTs. The schematic of the 8101 test fixture is shown in [Figure 3-29](#).

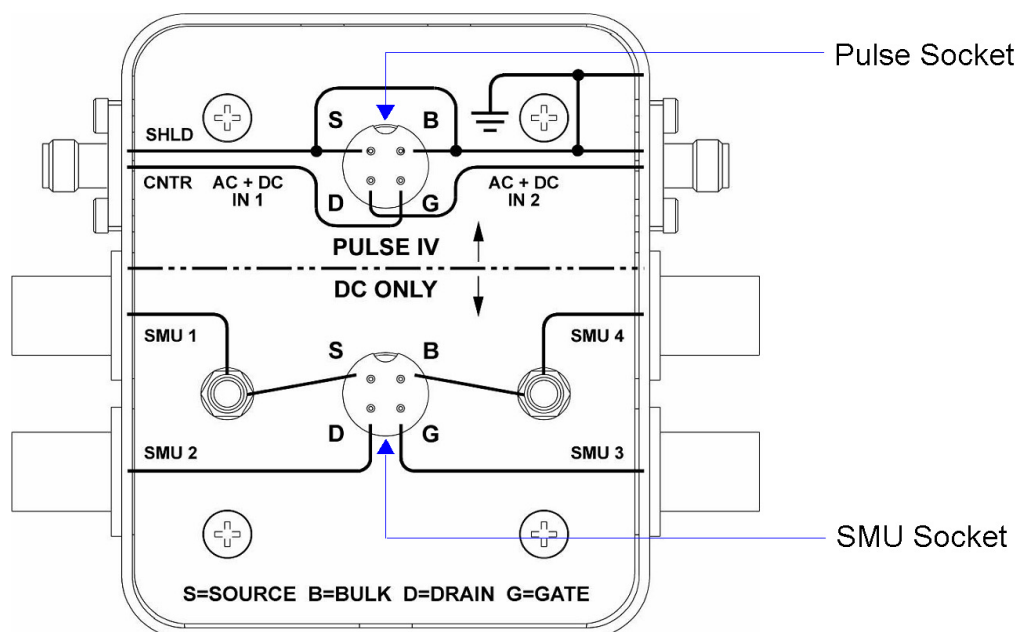
The tests included in both the PulseIV-Complete and PulseIV-Demo projects have parameter defaults that provide reasonable results with the included DUT (metal can TO-72, SD-210 nMOS FET).

Figure 3-28  
Model 8101-PIV test fixture





Figure 3-29  
**Model 8101-PIV schematic**



### Prober Interconnect

The PIV-A package provides both DC and Pulse capability to the DUT pins without re-cabling or switching. The key to this capability is the RBT, that uses passive electrical components to combine the low frequency DC signals with the high frequency pulse signals. For further information about the Model 4205-RBT refer to the [Reference Manual, Pulse Projects for Models 4200-PIV-A and 4200-PIV-Q, page 12-1](#).

The cabling from the RBT output is SMA, which will directly connect with RF probe manipulators using the DC probe adapter cables described below.

### DC Prober Interconnect

For DC structures, an adapter cable (Model 4200-PRB-C) is included to convert from the SMA to dual SSMC connections on DC manipulators. The adapter cable is shown in [Figure 3-30](#). Two 4200-PRB-C cables are included with the 4200-PIV-A package, which allows testing of either three or four terminal FETs.

Many DC probe manipulators are available with SSMC connections at the probe needle holder:

- Cascade DCM-2xx Kelvin DC probe manipulators
- Suss Microtec probe tips
- Signatone SCA-50 coaxial probes
- American Probe and Technologies:
  - 74CJ series coaxial probe holder
- Any probe interconnect with SSMC connectors near the probe tip

[Figure 3-31](#) shows the schematic diagram of the PRB-C adapter cable. The Ground Tap is to be connected to the Ground Tap of the second PRB-C adapter cable, as shown in [Figure 3-32](#).



Figure 3-30  
**PRB-C adapter cable – pulse SMA to SSMC Y**

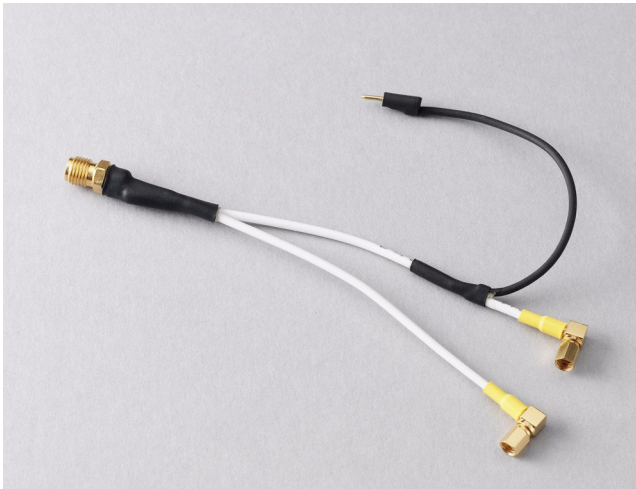
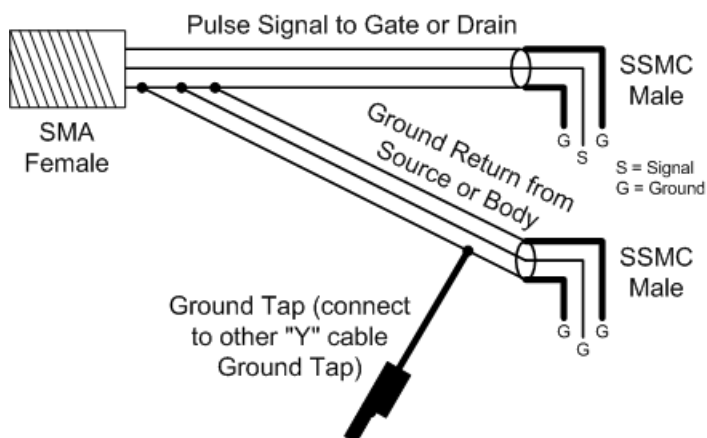


Figure 3-31  
**Schematic diagram of the PRB-C adapter cable**



These SMA to SSMC Y adapter cables are appropriate for on-wafer pulse IV testing of nominally DC structures. [Figure 3-32](#) shows Pulse IV connections from RBTs to DC probes for a DC layout DUT structure, using the PRB-C Y adapter cable.

These Y cables are not appropriate for higher frequency devices. The upper frequency limit is not specified, because the effect of actual device layout and probe configuration can have a significant impact. In general, any device that has an  $F_T$  much above 1GHz might oscillate when using a DC probe connection scheme and the PRB-C cables.

### RF Prober Interconnect

If the device has an RF layout (G-S-G), the Y adapter cables and DC probe manipulators will most likely be insufficient. In the case of RF G-S-G pad layout, do not use the Model 4200-PRB-C Y cables use the shorter SMA cables (6in/15cm) supplied with the PIV-A package to connect directly from the RBTs to the RF manipulators. The RBT with the power divider is connected to the Gate.

For additional information see the documentation included with the Model 4200-PRB-C (PA-928).



## PIV-A interconnect assembly procedure

1. Using the [Supplied interconnect parts on page 3-28](#) and [Supplied tools on page 3-29](#), refer to [Figure 3-25](#), [Figure 3-26](#) and [Figure 3-27](#) to configure the test setup for PIV testing. Use the supplied torque wrench for the SMA connections on the RBTs, power divider and manipulators. Use care when installing the cable to the scope card trigger SMB connector.
2. Perform one of the following procedures to connect the test system to the DUT:
  - For DC structures, prepare the probe connection by disconnecting all DC cables from the SSMC connectors on the needle holders. Continue setup of PIV-A by connecting a PRB-C cable to the 15cm (6in) SMA cable attached to each RBT. Refer to [Figure 3-32](#). Don't forget to connect the black shield jumpers to each other as shown in the middle of [Figure 3-32](#). Connecting shields together is necessary and very important, as it greatly reduces the inductance that is caused by the loop area of the interconnect.
  - For RF probes, connect the SMA cables from the RBTs to the RF probe manipulators, as shown in [Figure 3-33](#).
  - To use the supplied 8101-PIV test fixture (see [Figure 3-28](#)), connect the SMA cables from the RBTs to the 8101-PIV Test fixture as shown in [Figure 3-34](#). Install DUT as shown in [Figure 3-35](#).
3. Finish the setup by verifying connections and running a scope-shot test from the Pulse IV-Complete project.

Figure 3-32

### Pulse IV connections using PRB-C adapter cables

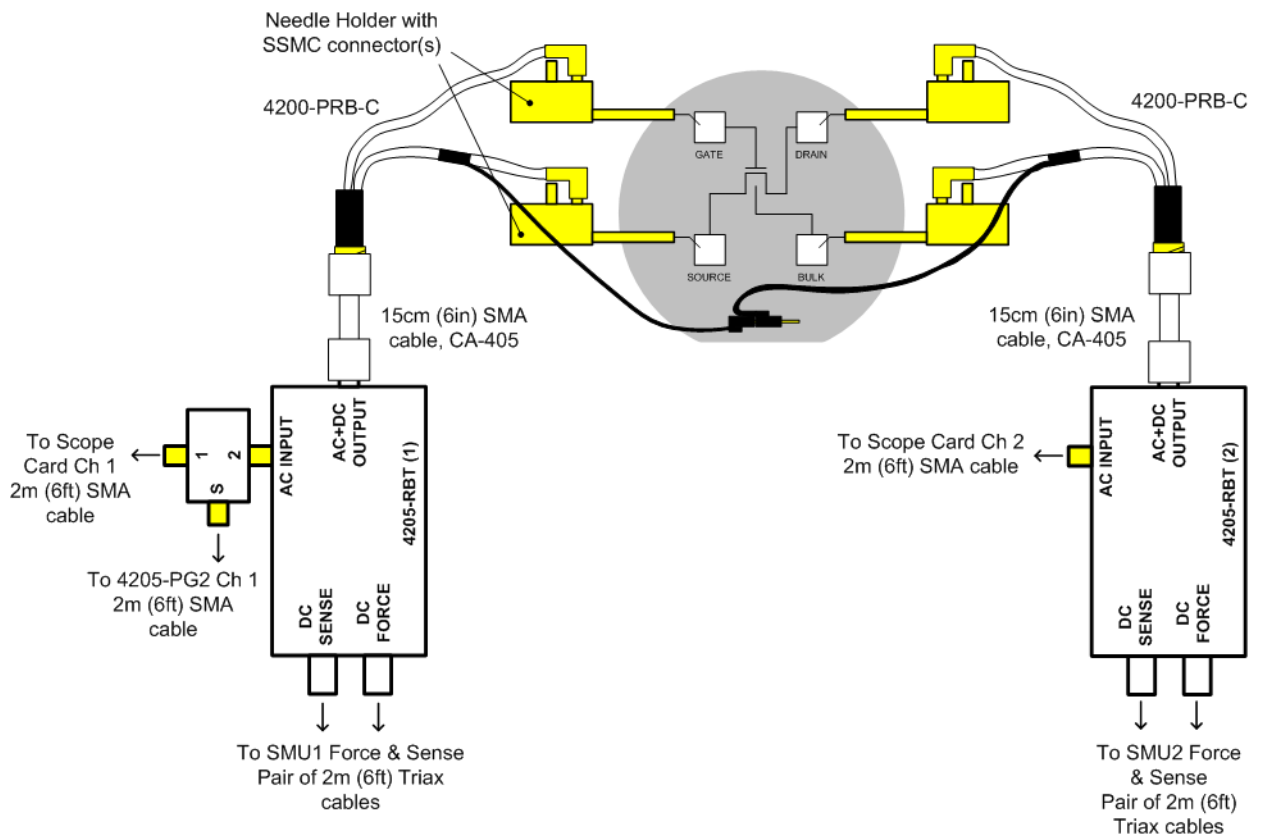




Figure 3-33  
Pulse IV connections using RF G-S-G probes

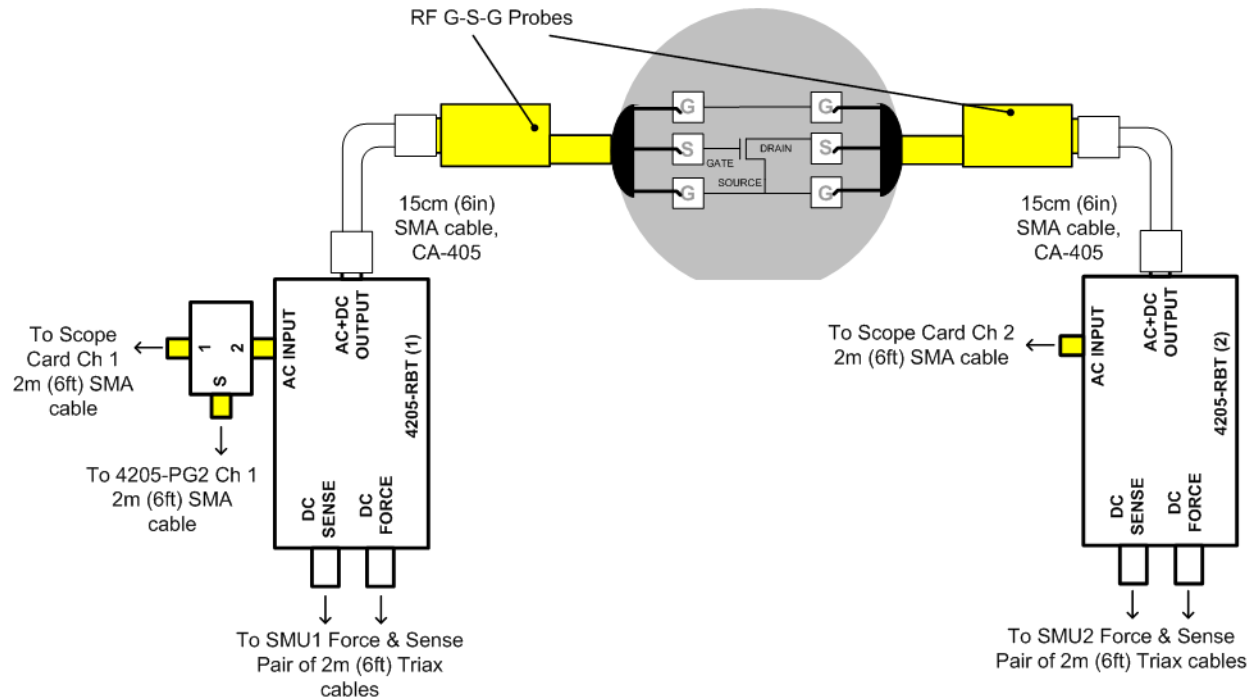


Figure 3-34  
Pulse IV connections using the 8101-PIV test fixture

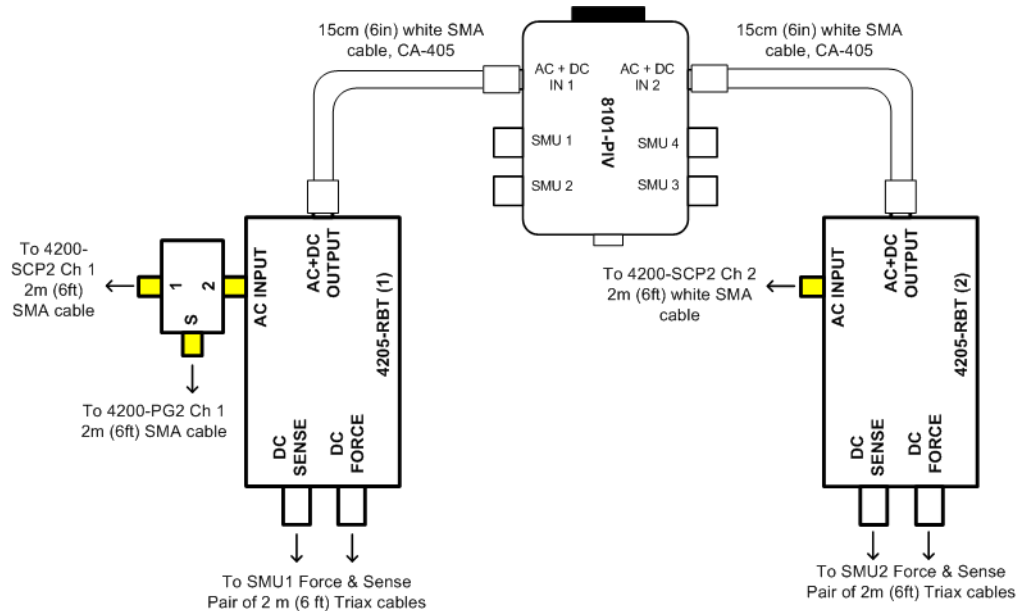
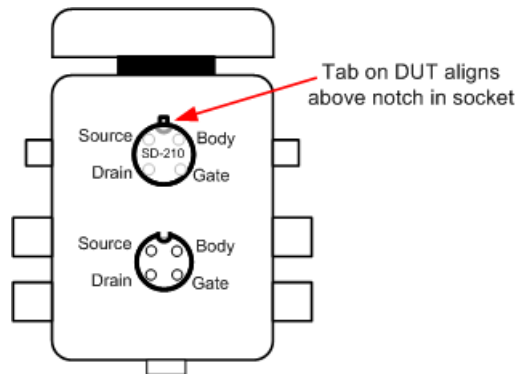




Figure 3-35  
DUT inserted in pulse socket of 8101-PIV test fixture



## Using the PulseIV-Complete project for the first time

### To use the PulseIV-Complete project:

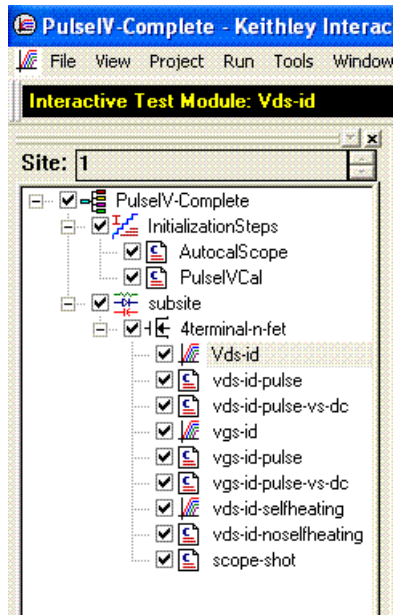
1. Connect PIV-A as explained above in [PIV-A interconnect assembly procedure on page 3-33](#).
2. If KITE is not running, start KITE by double-clicking the **KITE** icon on the Model 4200-SCS desktop.
3. Open the PulseIV-Complete project as follows:
  - a. Click **File** > Select **Open Project**.
  - b. If necessary, move up one level to display all the project folders, and double-click the `_Pulse` folder.
  - c. Double-click on the PulseIV-Complete folder.
  - d. Double-click `Pulse-IV-Complete.kpr` to open the project. [Figure 3-36](#) shows the project plan that is displayed on the left side of the KITE window.
4. Connect or touch-down on the chosen DUT.
5. Verify the setup as follows:
  - i. Follow the instructions for [Running scope-shot](#) to validate proper setup and operation of the PIV-A package. Ensure that both the gate and drain waveforms are visible and do not have any significant ringing or overshoot (see [Figure 3-45](#)).
  - ii. Try running vds-id-pulse ([Running vds-id-pulse UTM](#)) or vgs-id-pulse ([Running vgs-id-pulse UTM](#)) and look for a characteristic response.

If desired, DC IV tests may also be run ([Running vds-id DC ITM](#), [Running vgs-id DC ITM](#)). Once both the scope-shot and a pulse IV test have been verified, pulse system calibration can be performed.

6. Calibration: Perform the necessary pulse calibrations explained in [Running AutocalScope](#) and [Running PulseIVCal](#).
7. After successful pulse calibrations, the system is now ready to be used for pulse and DC characterization of transistor devices.



Figure 3-36  
Project plan for Pulse-IV Complete



## Running AutocalScope

AutocalScope should be run before any pulse calibration is performed. For best Pulse IV results, the AutocalScope should also be run before the first experiments of the day.

### To run AutocalScope:

1. The Model 4200-SCS should be turned on at least 30 minutes before performing any calibration or measurements.
2. Double-click **AutocalScope** in the project navigator (Figure 3-36).
3. Click the green **Run** button.
4. Follow the instructions given in the pop-up dialog box and disconnect all connections to the scope card.
5. The scope performs an autocal, that takes about one minute.
6. The test is complete when the Run button turns green. In the Sheet tab, `autoCalStatus=0` means that there were no errors.
7. Reconnect the cables to the scope card. Use care when installing the cable to the scope card trigger SMB connector.

## Running PulseIVCal

Verify proper setup by running a scope-shot. For on-wafer testing, have a through, or short, structure available, or ensure that sharing a pad for both the gate and drain probes provides a good connection. There are two steps to the calibration, open, and through/short.

### To run PulseIVCal:

1. If not already performed, run AutocalScope as explained above.
2. Double-click **PulseIVCal** in the project navigator (see Figure 3-36).
3. Click the green **Run** button to start the PulseIVCal.
4. Click **OK** on the first dialog box to continue the PulseIVCal (see Figure 3-37, left dialog box).



5. The second dialog box requests that the probe pins be raised from the wafer, breaking contact. Raise the probe pins or lower the wafer to create the Open condition. If using the 8101-PIV Test fixture, ensure that the pulse socket (near the fixture hinge), is empty.
6. Click **OK** on the Open dialog box (see [Figure 3-37](#), middle dialog box). The Open portion should take about one minute.
7. The third dialog box requests that the probe pins be connected to each other through another device. Lower the probes onto another device.
8. Click **OK** on the Through dialog box (see [Figure 3-37](#), right dialog box). The Through portion should take about one minute.
9. The test is complete when the Run button turns green. In the Sheet tab, `cal_pulseiv=0` implies that there were no errors.
10. The system is now ready to test regular devices.

Figure 3-37  
PulseIVCal dialog boxes

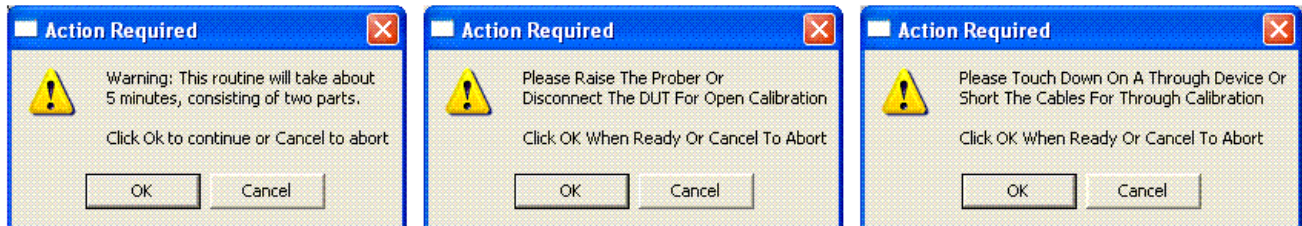
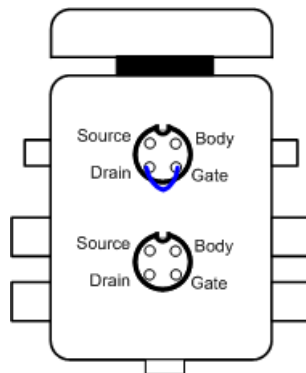


Figure 3-38  
8101-PIV shorted/through socket



## Running vds-id DC ITM

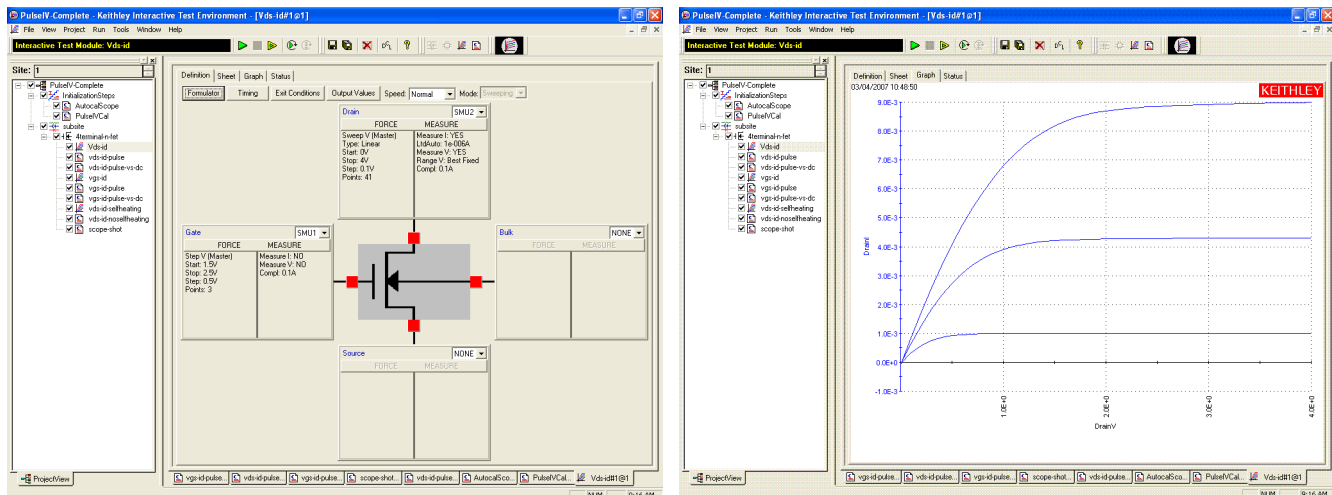
The default settings sweep the drain from 0-4 V in 100 mV steps while stepping through three gate voltages: 1.5 V, 2.0 V and 2.5 V (see [Figure 3-39](#)). When changing these settings, make note of the voltages and step size so that the same settings can be used in vds-id-pulse.

### To run vds-id DC ITM:

1. Double-click the **vds-id ITM** in the project navigator (see [Figure 3-36](#)).
2. Click the green **Run** button. Three vds-id curves will be generated and displayed on the graph.



Figure 3-39  
Default definition and typical graph for vds-id



## Running vds-id-pulse UTM

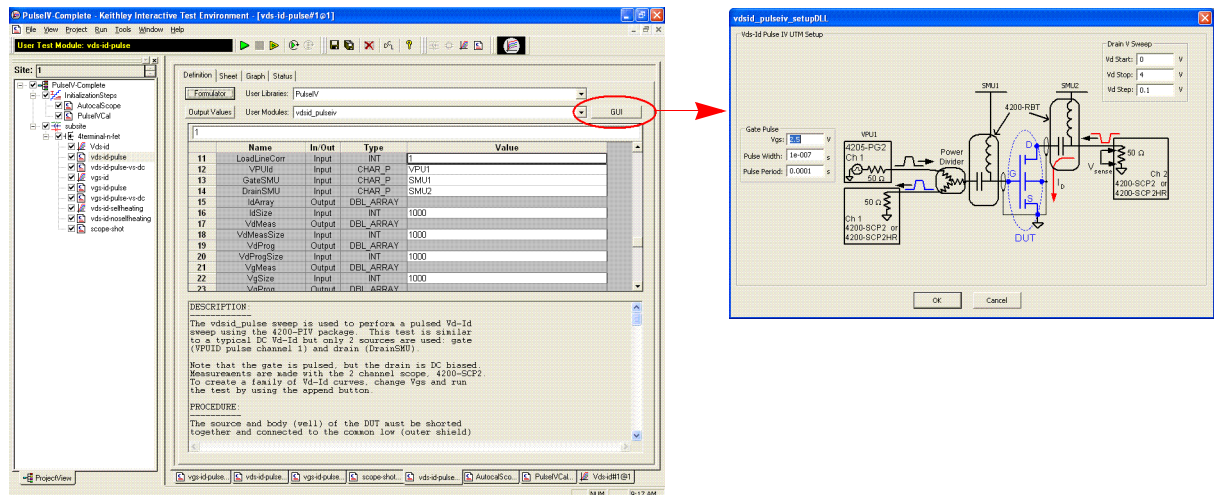
The default vds-id-pulse test uses (see [Figure 3-40](#)) the same drain voltage settings as the DC vds-id. The vds-id-pulse does not have the automatic step capability of the DC vds-id. There are two ways to generate a family of pulse IV curves. The easier way is to use the vds-id-pulse-vs-DC (see [Running vds-id-pulse-vs-dc UTM](#) below). If using the 8101-PIV test fixture, insert the metal can (SD-210) DUT as shown in [Figure 3-35](#).

To run the three gate voltages using single curve vds-id-pulse:

1. Ensure that the VdStart, VdStop, VdStep values match the values in the DC vds-id. To sweep from a high to a low voltage, enter voltages so that vdstop < vdstart and use a negative value for VdStep. If any values need to be modified, remember to press the **Enter** key after typing in the value.
2. Set vgs to the first voltage. The default is 1.5 V. Make sure to press the **Enter** key after typing in the value.
3. Click the green **Run** button.
4. After the test is finished, set vgs to the second voltage. The default is 2.0 V.
5. Click the yellow and green **Append** button.
6. After the test is finished, set vgs to the third voltage. The default is 2.5 V.
7. Click the yellow and green **Append** button.
8. To add or update the DC results on the pulse Graph, perform the procedure for [Comparing DC and pulse results](#).



Figure 3-40  
Default Definition tab and GUI For vds-id-pulse



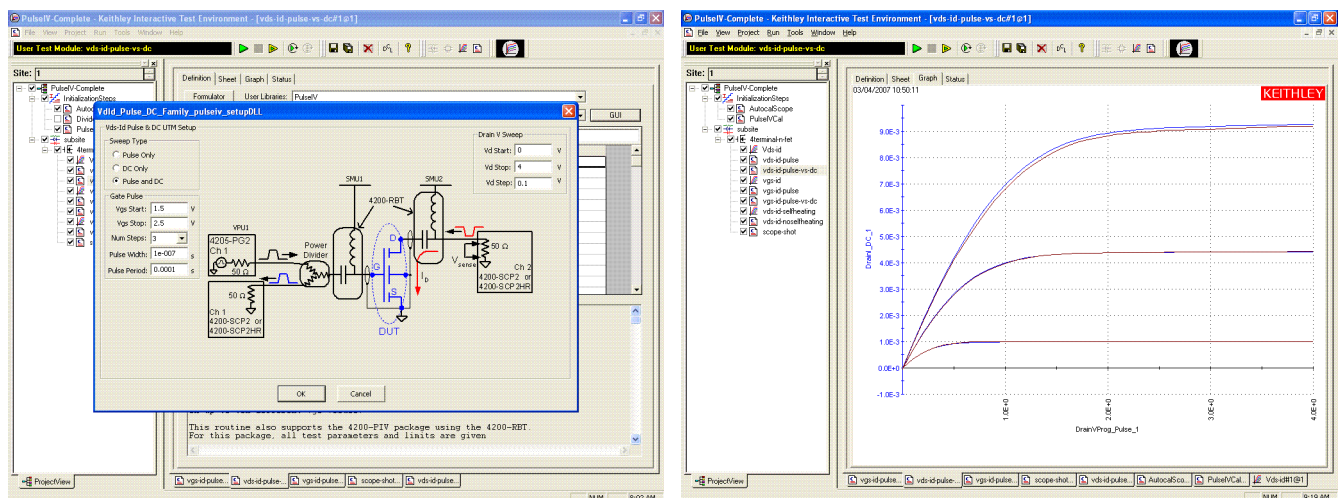
## Running vds-id-pulse-vs-dc UTM

The default settings are the same as the vds-id-pulse UTM, with the addition of the DC measurement parameters (see [Figure 3-41](#)):

To run vds-id-pulse-vs-dc UTM:

1. If measurement parameters (pulse average, NPLC, measure range) need to be set, use the definition table.
2. (Optional) If only source parameters need to be changed, use the UTM GUI by clicking the GUI button on the vds-id-pulse-vs-DC test. Modify the source parameters in the GUI, and click **OK** when finished.
3. Click the green **Run** button. For a test with three curves and 40 points per curve, the test should take about 1.5-2 minutes. During the test, neither the Graph tab or Sheet tab is updated.

Figure 3-41  
Default definition and typical graph for vds-id-pulse-vs-dc



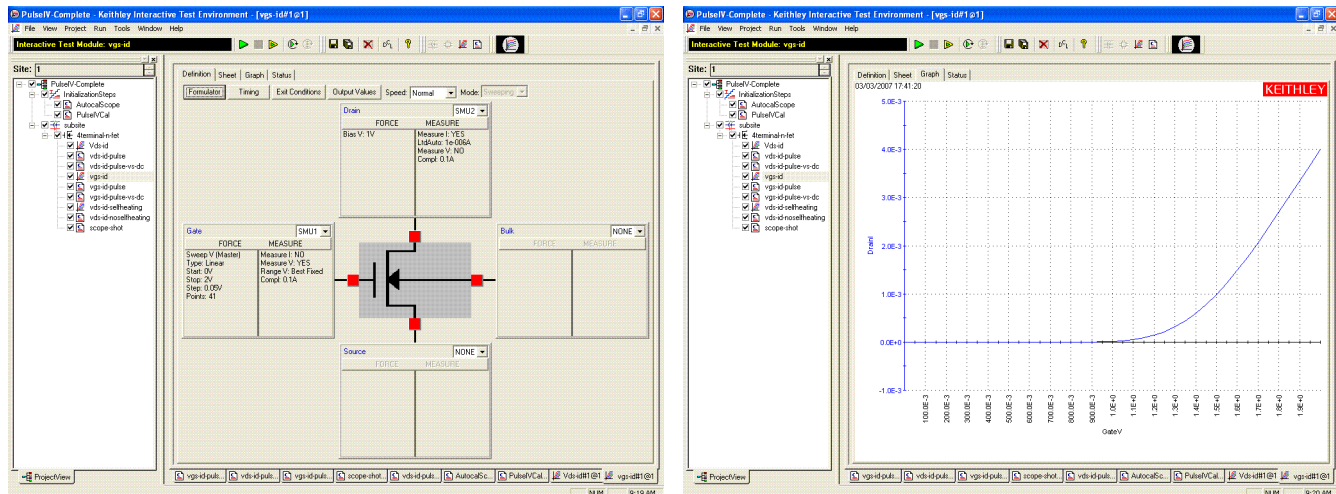


## Running vgs-id DC ITM

The default settings sweep the pulses on the gate from 0-2 V in 50 mV steps and set the drain voltage to 1 V (see [Figure 3-42](#)). When changing these settings, note the voltages and step size used so the same settings can be used in vgs-id-pulse.

1. Double-click **vgs-id ITM** in the project navigator.
2. Click the green **Run** button. The Vgs-id curve will be generated and displayed on the graph.

Figure 3-42  
Default definition and typical graph for vgs-id



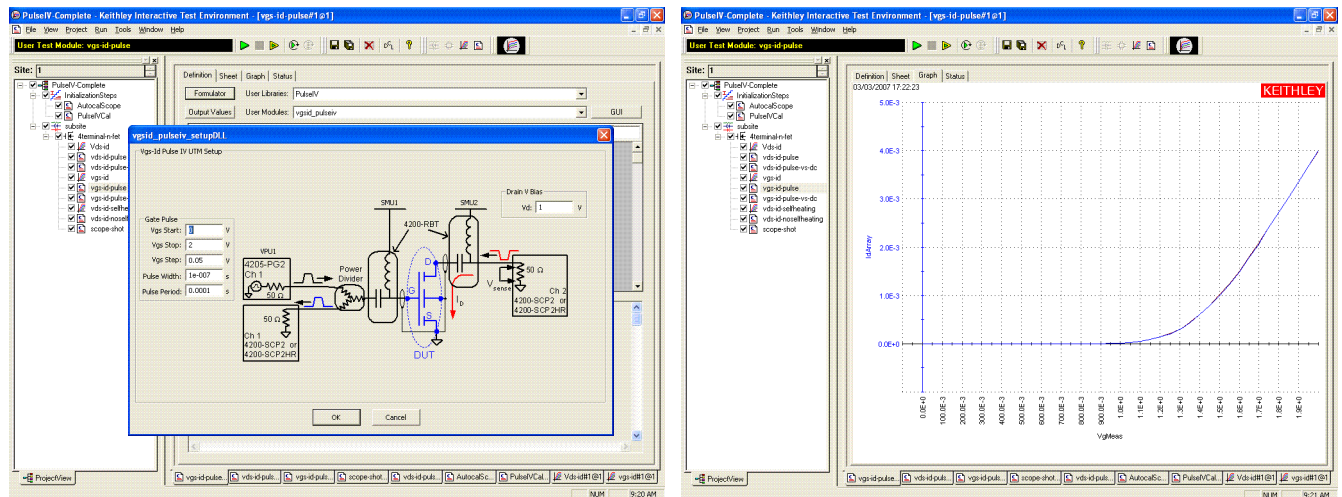
## Running vgs-id-pulse UTM

The default Vgs-id-pulse uses the same default settings as the DC Vgs-id (see [Figure 3-43](#)). If comparing Vgs-Id results for DC and Pulse IV, use this pulse-only routine and the [Comparing DC and pulse results](#) or use the single DC and Pulse UTM as described in [Running vgs-id-pulse-vs-dc UTM](#). Alternately, the source values may be entered using the UTM GUI:

1. Ensure that the Vds, VgStart, VgStop, VgStep values match the values in the DC Vgs-id. To sweep from a high to a low voltage, enter voltages so that VdStop < VdStart and use a negative value for VdStep. If any values need to be modified, remember to press the **Enter** key after typing in the value.
2. Click the green **Run** button.
3. To add or update the DC results on the pulse Graph, perform the procedure for [Comparing DC and pulse results](#).
4. To reduce noise, the smaller subthreshold currents of this test require a larger number of measurements to be averaged. For best results on smaller signals ( $I_d < 500\mu A$ ), use AverageNum = 0 to enable the adaptive filtering mode, where lower scope ranges will use a large AverageNum and higher ranges a lower AverageNum. If desired, a fixed number may be entered for AverageNum.



Figure 3-43  
Default definition and typical graph for vgs-id-pulse

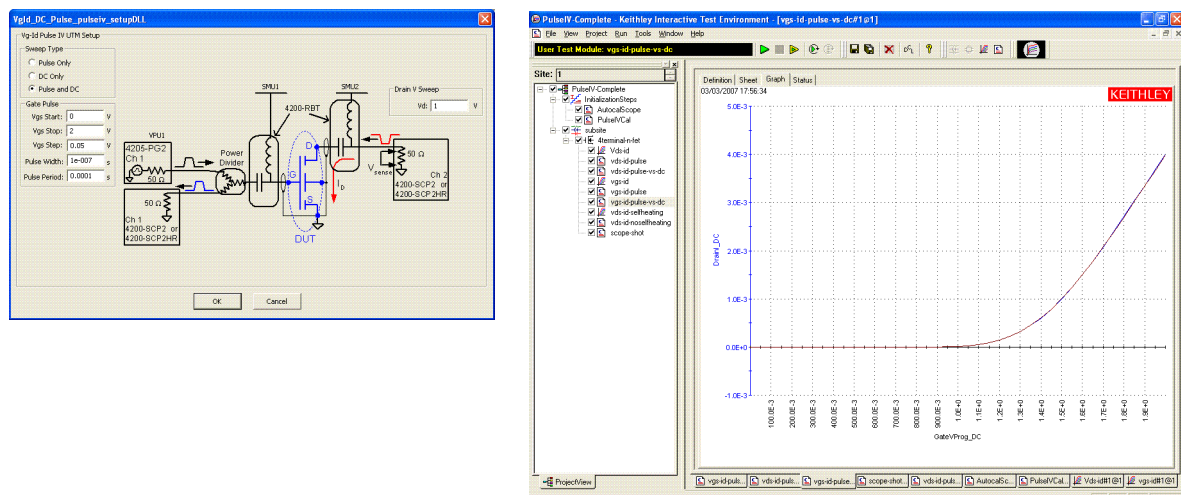


### Running vgs-id-pulse-vs-dc UTM

Instead of using the separate vgs-id ITM and vgs-id-pulse UTM to compare DC and pulse Vg-Id results, the vgs-id-pulse-vs-DC UTM combines both DC and pulse tests (see [Figure 3-44](#)):

1. If measurement parameters (pulse average, NPLC, measure range) need to be set, use the definition table.
2. (Optional) If only source parameters need to be changed, use the UTM GUI by clicking on the GUI button on the Vds-id-pulse-vs-DC test. Modify the source parameters and click **OK** when finished.
3. Click the green **Run** button. For a test with 40 points, the test should take about one minute. During the test, neither the graph tab or sheet tab is updated.

Figure 3-44  
Default GUI definition and typical graph for vgs-id-pulse-vs-dc



### Running scope-shot

The scope-shot test is used to verify proper connection and system setup. The waveform shown in [Figure 3-45](#) is a typical result; actual results should be similar. If waveform has significant ringing or overshoot, the pulse IV tests will not provide good results. Check the pulse interconnects to

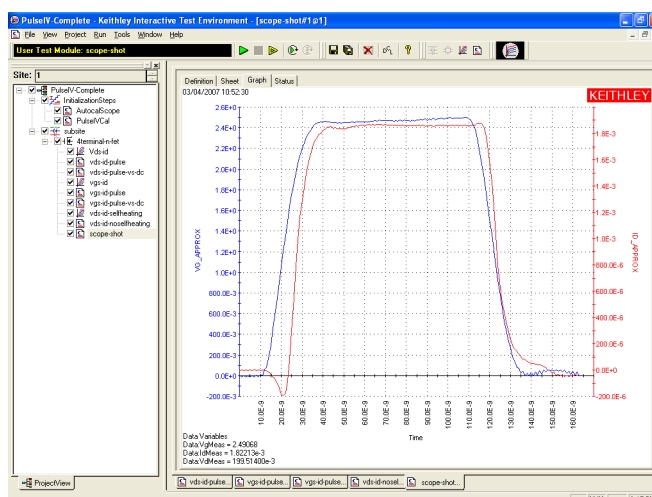


ensure proper cabling and ensure all connections are tight. If using the Model 4200- PRB-C cables (Y adapter cable for pulsing with DC interconnect and structures), ensure that the two ground lugs are connected together.

The left pulse curve (blue) is the pulse applied to the gate. The displayed waveform data has approximate calibration factors applied, but the calibrated measurement is given in the lower left portion of the Graph tab. Note that the Data Variables values in the lower left corner just display the Data values, not subsequent test runs (Appends) which are only graphically displayed. The right pulse curve (red) is the drain current, shown with approximate calibration factors applied, with the calibrated  $V_d$  and  $I_d$  measurements listed in the lower left corner of the graph.

The AverageNum value specifies the number of pulses that are averaged together to provide the data. The DUT will have more pulses applied than AverageNum due to other test factors, such as load line correction and measurement autoranging.

Figure 3-45  
Typical graphical result for scope-shot



### Adjustable parameters in scopeshot\_cal\_pulseiv

Vds	DC voltage for the drain
Vgs	Pulse voltage level for gate
PulseWidth	Vgs pulse width, full width half maximum (FWHM)
PulseAmplitude	Vgs, gate voltage pulse
PulsePeriod	Vgs pulse period. When using the Pulse IV setup with RBTs, use a PulsePeriod 1000 x PulseWidth, to keep the pulse duty cycle less than or equal to 0.1%. For most cases, it is best to use 200 E-6, that will allow an appropriate duty cycle across the range of supported 40-150 ns pulse widths.
GateRange	Scope card gate voltage range. Use 0 for autoranging, or a specific value for a fixed range. The scope range is centered around zero, so the 5 V range on the scope covers -2.5 V to +2.5 V. As an example, for a 3 V signal, use GateRange = 10 (-5 V to +5 V), not 5 (-2.5 to +2.5 V). Available ranges for the scope card: 0.05, 0.10, 0.25, 0.5, 1, 2, 5, 10 V.
DrainRange	Scope card drain voltage range. Use 0 for autoranging, or a specific value for a fixed range. To calculate an appropriate fixed range, use DrainRange = (Estimated $I_d$ ) x 50 x 2. See above for valid scope card ranges.



AverageNum	Number of pulses to average. For larger currents, $I_d > 500 \mu\text{A}$ , AverageNum = 10-25 is usually sufficient. For smaller $I_d$ , use 50-100. Larger values provide minimal additional improvement. All pulse IV tests have this setting, that controls how many pulses are used to return a result, not how many pulses are sent to the DUT.
LoadLineCorr	Turns drain side load line correction on or off. This is similar to the vds-id-pulse and vgs-id-pulse tests and is a routine to ensure that the desired $V_d$ is provided to the drain DUT terminal, regardless of the amount of $I_d$ flowing through the DUT.
VPUID	Pulse card identification string: VPUID = VPU1.
GateSMU	SMU for DC $V_g$ . Default is SMU1, but any other available SMU may be used.
DrainSMU	SMU for DC $V_d$ . Default is SMU2, but any other available SMU may be used.

### Tips for using Pulse IV

- Confirm connection: Use scope-shot as the first test after touching down on a device to confirm that there is proper connection to the DUT, before running PulseIVCal or any pulse tests.
- Always calibrate after any setup changes (new probe tips or manipulators, cable replacement).
- Proper pulse IV performance can be verified by testing a device that does not exhibit any selfheating or charge trapping effects. The 8101-PIV Test Fixture and SD-210 DUT provide good DC and pulse correlation for  $I_d < 10 \text{ mA}$  and  $V_d < 5 \text{ V}$ . For  $I_d < 1 \text{ mA}$ , set AverageNum = 0 to use the Adaptive Filtering. If a fixed number for AverageNum is desired, use AverageNum = 2000 for  $I_d < 500 \mu\text{A}$ .
- Pulse IV measurements have less resolution and sensitivity than typical DC results, so test parameters, such as averaging or smaller steps sizes, and post-test processing, such as curve fitting, may be required to obtain roughly equivalent results.

### Comparing DC and pulse results

There are two methods for comparing DC and Pulse IV results. The first method uses the UTMs that combine pulse and DC tests: Vds-id-pulse-vs-dc, vgs-id-pulse-vs-dc. The second method, described below, uses the data in the Sheet tab of KITE to compare any results across tests. This procedure explains how to copy the DC results into a pulse UTM to allow comparison between pulse and DC IV results in a single graph.

1. In the PulseIV-Complete project navigator, double-click the **vds-id ITM** in the project navigator.
2. Click the **Sheet** tab.
3. Choose the desired results worksheet. If there is only one set of curves, then the results are in the **Data** tab. If additional tests have been appended, choose the desired Append tab.
4. Highlight all of the data in the desired worksheet by clicking the **Entries Selection Cell** as shown in [Figure 3-46](#).
5. To copy the data, right-click the **Entries Selection Cell** and select **Copy** from the drop-down menu.
6. In the PulseIV-Complete project navigator, double-click **vds-id-pulse UTM**.
7. Click the **Sheet** tab.
8. Click the **Calc** worksheet tab.



9. Click cell A1.
10. To paste DC data into the pulse Calc worksheet, right-click cell **A1**, and select **Paste** from the drop-down menu. If there is previous data in the Calc worksheet, the paste operation will overwrite it. [Figure 3-47](#) shows the data that was pasted from the vds-id ITM.
11. All comparison DC and pulse data is now located in the same test. The graph needs to be defined to display the DC data located in the Calc worksheet.
12. For vds-id-pulse, click the **Graph** tab.
13. Right-click the graph and select the first option (**Define Graph**) from the drop-down menu. The Define Graph dialog box is shown in [Figure 3-48](#).
14. In Define Graph, click the cells in **Column Y1**, to add the appropriate DC curves. In this case, three vds-id curves have been added: Drain1(1), Drain1(2), Drain1(3). These Y1 cells are circled in red. All three of these Data Series are located in the Calc worksheet, as noted in the Sheet column. Also shown in [Figure 3-48](#) is graph with the three added curves.
15. To change graph colors or add data point shapes/patterns, move the cursor along the desired curve until the pointer appears. With the pointer displayed, right-click to get the Data Series Properties dialog box. Select a shape property to demarcate each data.
16. To verify pulse operation, use a DUT that does not exhibit any self-heating or transient charging effect. In a properly configured and calibrated system, the pulse IV results should correlate to the DC results within  $\pm 4\%$ , with many results less than or equal to 2%, when testing a device that does not exhibit heating or charging effects.

Figure 3-46  
Highlighting all entries in vds-id data sheet

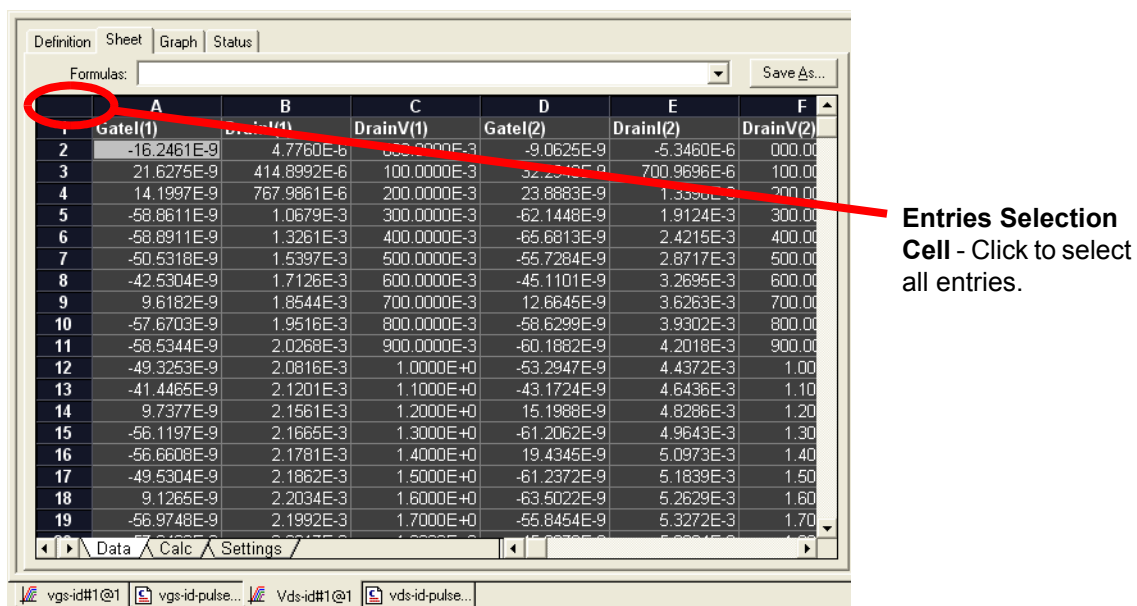




Figure 3-47

Data from vds-id pasted into vds-id-pulse calc sheet

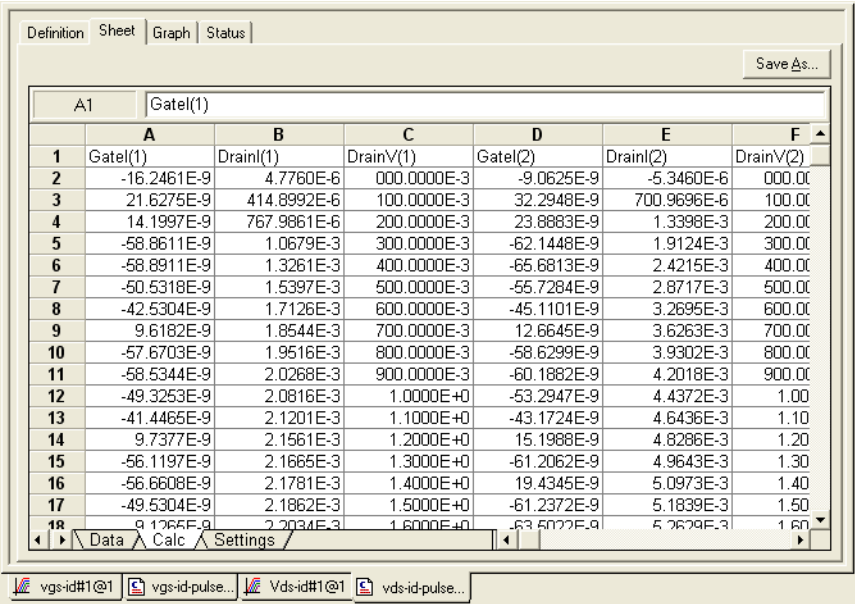
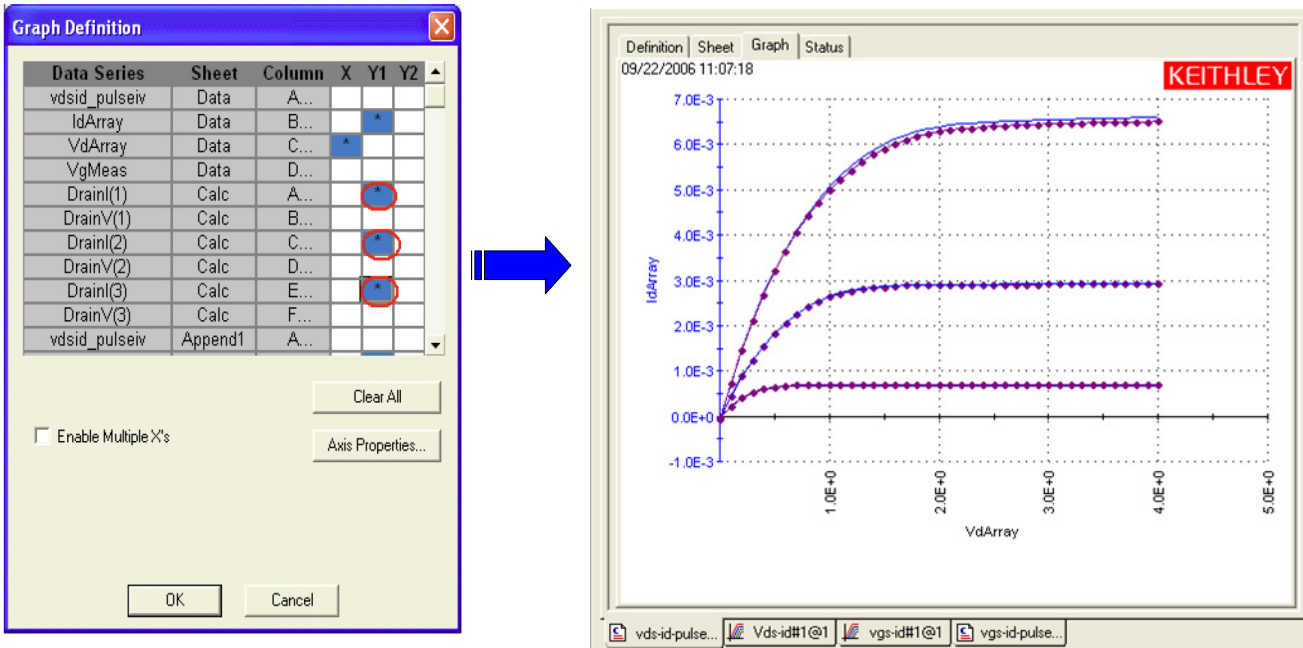


Figure 3-48

Graph Definition dialog box and resulting graph that shows the three added curves





## Pulse IV UTM descriptions

The pulse IV user library contains modules required to provide low duty cycle pulsed IV testing. The modules contained in the pulse IV user library are listed in [Table 3-3](#) with detailed information following the table.

Table 3-3

**Pulse IV UTMs**

User Module	Description
cal_pulseiv	Performs a cable compensation routine.
vdsid_pulseiv	Performs a pulsed Vd-Id sweep.
vdId_Pulse_DC_Family_pulseiv	Performs a Pulsed vs. DC Vd-Id sweep.
vgId_DC_DC_pulseiv	Performs a Pulsed vs. DC Vg-Id sweep.
vgsid_pulseiv	Performs a pulsed Vg-Ig sweep.
scopeshot_cal_pulseiv	Used to display a single Pulse IV scopeshot_pulseiv.
scopeshot_pulseiv	Displays a single Pulse IV scopeshot.
vdsid_pulseiv_demo	Performs a pulsed Vd-Id sweep, with simplified parameter list.
vgsid_pulseiv_demo	Performs a pulsed Vg-Id sweep, with simplified parameter list.

### cal\_pulseiv

**Description** The cal\_pulseiv module is used to perform a cable compensation routine for the 4200-PIV package. This routine permits the system to compensate for losses in the cabling from the 4200 to the connection to the DUT. Use this routine during initial system setup and whenever changes are made in any part of the interconnect (cables, 4200-RBTs, probe manipulators or pins). There are two main steps to this procedure:

- Open cal—the gate signal is measured while there is no connection to the DUT.
- Through cal—the drain signal is measured while making contact on a Through structure (or by shorting the two 4200-RBTs, AC+DC outputs with an appropriate cable, or adapter).

The factors generated by this routine are used during any testing where the 4200-RBTs are used (vdsid\_pulse, vgsid\_pulse). Make sure to set the appropriate values for the cal\_pulseiv parameters in [Table 3-4](#), [Table 3-5](#) and [Table 3-6](#) contain outputs and return values, respectively.

**Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the RBTs. The RBT connected to GateSMU (with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. For detailed connection information, refer to the [PIV-A interconnect assembly procedure on page 3-33](#).



Table 3-4  
Inputs for cal\_pulseiv

Input	Type	Description	Default
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with the 4200-PIV package.	VPUID
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SmUs in the system.	GateSMU
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SMUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.	DrainSMU
vRange	int	The pulse generator card voltage source range to be calibrated (V). Valid values are: 5, 20.	5
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 40 us to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%.	100 e-6
Vs_Size Vm1_size Vm2_size	int	Set to a value that is at least equal to the number of steps in the sweep and all three must be the same value.	100 100 100

Table 3-5  
Outputs for cal\_pulseiv

Output	Type	Description
Vs	double *	The pulse source value (V).
Vm1	double *	The measured voltage from channel 1 of the scope card.
Vm2	double *	The measured voltage from channel 2 of the scope card.

Note: These outputs are included for compatibility with older setups. They no longer return any information.

Table 3-6  
Return values for cal\_pulseiv

Value	Description
0	OK
-13001	Array Sizes Do Not Match
-13002	Arrays Not Large Enough For Data
-13003	Invalid Instruments
-13004	Unable To Malloc Memory
-13005	Unable To Find Delay Between Channels
-13006	Scope Measurement Error
-13007	Unable To Write To Calibration Files
-13008	Invalid Range
-13009	Invalid Calibration Type
-13010	Calibration Data Does Not Meet Correlation Specification
-13998	Calibration Constant Error
-13999	Divider Cal Error



## vdsid\_pulseiv

**Description** The vdsid\_pulse sweep is used to perform a pulsed Vd-Id sweep using the 4200-PIV package. This test is similar to a typical DC Vd-Id but only two sources are used: gate (VPUID pulse channel 1) and drain (DrainSMU). The gate is pulsed, but the drain is DC biased.

Measurements are made with the two channel scope card. To create a family of Vd-Id curves, change Vgs and run the test by using the append button. Make sure to set the appropriate values for the Vds-Id parameters (see [Table 3-7](#)). [Table 3-8](#) and [Table 3-8](#) contain outputs and return values, respectively.

**Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the 4200-RBT. The RBT connected to GateSMU (with the power divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. For detailed connection information, refer to the [PIV-A interconnect assembly procedure on page 3-33](#).

Table 3-7  
Inputs for vdsid\_pulseiv

Input	Type	Description
Vgs	double	The pulsed gate-source voltage bias, output by channel 1 of the pulse generator card (VPUID).
Vg_off	double	The DC bias applied by the GateSMU to put device in the OFF state. Normally set to 0 V for enhancement FETs (may be non-zero for depletion FETs).
VdStart	double	The starting sweep value for Vd, output by the DrainSMU (defined below).
VdStop	double	The final sweep value for Vd, output by the DrainSMU (defined below).
VdStep	double	The sweep step size for the Vd sweep, output by the DrainSMU (defined below).
PulseWidth	double	The Vgs pulse width (PW). The PW can be 40 ns to 150 ns (10 ns resolution). Pulses wider than 150 ns will begin to be attenuated by the capacitor in the 4200-RBT.
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 100μs to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. The period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150 μs.
AverageNum	int	The number of pulses to average at each step of the sweep. For best low signal performance, set AverageNum = 0 for Adaptive Filtering.
GateRange	double	The voltage measure range for the scope channel measuring the Gate. Use 0 for scope autoranging, or specify a voltage value for a fixed range. Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10.
DrainRange	double	The voltage measure range for the scope channel measuring the Drain. Use 0 for scope autoranging, or specify a voltage value for a fixed range, where $V = I * 50 \Omega$ . Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10.
LoadLineCorr	int	Determines whether to use load line correction to compensate for the voltage drop caused by the 50 Ω sense resistor used to measure the drain current (Id). 1 = load line correction active. 0 = no load line correction.



Table 3-7 (continued)  
**Inputs for vdsid\_pulseiv**

Input	Type	Description
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with the 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SmUs in the system.
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SMUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.
IdSize VdMeasSize VdProgSize VgSize VgProgSize	int	Set to a value that is at least equal to the number of steps in the sweep and all five must be the same value.

Table 3-8  
**Outputs for vdsid\_pulseiv**

Output	Type	Description
IdArray	double *	The measured drain current from channel 2 of the scope card. This current is determined by measuring the voltage drop across the scope card 50 $\Omega$ termination, giving $I_d = V_d / 50 \Omega$
VdMeas	double *	Array of measured drain voltage values.
VdProg	double *	Array of programmed drain voltage values.
VgMeas	double *	The measured gate voltage from channel 1 of the scope card.
VgProg	double *	Array of programmed gate voltage values.

Table 3-9  
**Return values for vdsid\_pulseiv**

Value	Description
0	OK
-1	Invalid value for Vgs
-2	Invalid value for VdStart
-3	Invalid value for VdStop
-4	Invalid value for VdStep
-5	Invalid value for PulseWidth
-6	Invalid value for PulsePeriod
-7	Invalid value for AverageNum
-8	Invalid value for LoadLineCorr
-9	Array sizes do not match
-10	Array sizes not large enough for sweep
-11	Invalid VPUID
-12	Invalid GateSMU
-13	Invalid DrainSMU
-14	Unable to initialize PIV solution



## VdId\_Pulse\_DC\_Family\_pulseiv

**Description** The VdId\_Pulse\_DC\_Family\_pulseiv sweep is used to perform a Pulsed vs DC Vd-Id sweep using the 4200-PIV-A package. This test is similar to a typical Vd-Id but only two sources are used: one for the DUT Gate and one for the DUT Drain. Pulsed Measurements are made with the 2-channel scope, 4200-SCP2. To create a family of curves, choose an appropriate start and stop value for Vgs, and a number of steps.

This routine can run the sweeps in three different ways: 1) DC only; 2) Pulse only; 3) Pulse and DC curves. This routine supports from one to 10 Vd-Id curves based on up to 10 different Vgs values.

This routine also supports the 4200-PIV-A package using the 4200-RBT. For this package, all test parameters and limits are given below, except the 4200-PIV-A with the 4200-RBT has a max pulse width of 150 ns, not the 250 ns of the 4205-RBT.

All voltage levels specified below assume a 50  $\Omega$  DUT load.

**Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the 4205-RBT. The RBT connected to GateSMU (the RBT with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. Use either G-S-G probes for RF structures, or use DC probes with the 4200-PRB-C adapter cables for DC structures.

Set the appropriate values for the Vds-Id parameters. Inputs, outputs and returned values are provided in [Table 3-10](#), [Table 3-11](#) and [Table 3-12](#).

Table 3-10

**Inputs for VdId\_Pulse\_DC\_Family\_pulseiv**

Input	Type	Description
VgStart	double	The starting step value for Vg. For DC only sweeps, VgStart must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT. For pulse and pulse and DC Sweeps, VgStart must be between -5 V to +5 V.
VgStop	double	The final step value for Vg. For DC only sweeps, VgStop must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT. For pulse and pulse and DC Sweeps, VgStop must be between -5 V to +5 V.
VgNumSteps	double	The number of steps for Vg (Max = 10).
Vg_off	double	The DC bias applied by the GateSMU to put device in the OFF state. Set to 0 V for enhancement FETs (may be non-zero for depletion FETs).
VdStart	double	The starting sweep value for Vd. For DC only sweeps, VdStart must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT.
VdStop	double	The final sweep value for Vd. For DC only sweeps, VdStop must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT.
VdStep	double	The number of steps for the Vd sweep (Max = 10000).



Table 3-10 (continued)  
Inputs for VdId\_Pulse\_DC\_Family\_pulseiv

Input	Type	Description
PulseWidth	double	The Vgs pulse width (PW). The PW can be 40 ns to 250 ns (10 ns resolution). Pulses wider than 250 ns will begin to be attenuated by the coupling capacitor in the Remote Bias Tee (4205-RBT).
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 100µs to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. The period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150µs.
AverageNum	int	The number of pulses to average at each step of the sweep. For best low signal performance, set AverageNum = 0 for Adaptive Filtering.
GateScpRange	double	The voltage measure range for the scope channel measuring the Gate. Use 0 for scope autoranging, or specify a voltage value for a fixed range. Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10. These ranges are Vpp. For example, the 0.5 range covers -250 to +250 mV.
DrainScpRange	double	The voltage measure range for the scope channel measuring the Drain. Use 0 for scope autoranging, or specify a voltage value for a fixed range, where $V = I * 50 \Omega$ . Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10. These ranges are Vpp. For example, the 0.5 range covers -250 to +250 mV.
GateSMURange	int	The current measurement range to be used for the SMU on the DUT Gate terminal. Values correspond to the table below. Limited Auto means that the value given is the minimum measurement range used, with automatic ranging for larger currents. 1 Full Auto 2 Limited Auto 10 pA 3 Limited Auto 100 pA 4 Limited Auto 1 nA 5 Limited Auto 10 nA 6 Limited Auto 100 nA 7 Limited Auto 1 µA 8 Limited Auto 10 µA 9 Limited Auto 100 µA 10 Limited Auto 1 mA 11 Limited Auto 10 mA 12 Limited Auto 100 mA
DrainSMURange	int	The current measurement range to be used for the SMU on the DUT Drain terminal. Values correspond to the table below. Limited Auto means that the value given is the minimum measurement range used, with automatic ranging for larger currents. 1 Full Auto 2 Limited Auto 10 pA 3 Limited Auto 100 pA 4 Limited Auto 1 nA 5 Limited Auto 10 nA 6 Limited Auto 100 nA 7 Limited Auto 1 µA 8 Limited Auto 10 µA 9 Limited Auto 100 µA 10 Limited Auto 1 mA 11 Limited Auto 10 mA 12 Limited Auto 100 mA



Table 3-10 (continued)  
**Inputs for VdId\_Pulse\_DC\_Family\_pulseiv**

Input	Type	Description
LoadLineCorr	int	Determines whether to use load line correction to compensate for the voltage drop caused by the 50 $\Omega$ sense resistor used to measure the drain current (Id). 1 = load line correction active. 0 = no load line correction.
GateCompliance	double	The SMU current compliance for the DUT Gate.
DrainCompliance	double	The SMU current compliance for the DUT Drain.
NPLC	double	The DC measurement integration time in NPLC (Number of Power Line cycles).
DCSourceDelay	double	Time, in seconds, between the DC source and measure for each sweep point.
DC_vs_Pulse	int	Determines whether to run a DC and Pulse test or a DC only test or a Pulse only test. 0 - Pulse Only, 1 - DC Only, 2 - DC and Pulse.
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with a 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SMUs in the system.
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SMUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.
DrainVMeas_DC_Size DrainVProg_DC_Size DrainI_DC_Size GateVMeas_DC_Size GateVProg_DC_Size DrainVMeas_Pulse_Size DrainVProg_Pulse_Size DrainI_Pulse_Size GateVMeas_Pulse_Size GateIProg_Pulse_Size	int	Sizes of the output arrays. All arrays should be the same size and need to be large enough to hold all sweep points.

Table 3-11  
**Outputs for VdId\_Pulse\_DC\_Family\_pulseiv**

Output	Type	Description
DrainVProg_DC DrainVProg_Pulse	double	Array of programmed drain voltage values.
DrainVMeas_DC DrainVMeas_Pulse	double	Array of measured drain voltage values.
DrainI_DC DrainI_Pulse	double	Array of measured drain currents.
GateVMeas_DC GateVMeas_Pulse	double	Array of measured gate voltages.
GateVProg_DC GateVProg_Pulse	double	Array of programmed gate voltages.



Table 3-12  
Return values for Vdid\_Pulse\_DC\_Family\_pulseiv

Value	Description
0	OK
-1	Invalid value for Vgs
-2	Invalid value for VdStart
-3	Invalid value for VdStop
-4	Invalid value for VdStep
-5	Invalid value for PulseWidth
-6	Invalid value for PulsePeriod
-7	Invalid value for AverageNum
-8	Invalid value for LoadLineCorr
-9	Array sizes do not match
-10	Array sizes not large enough for sweep
-11	Invalid VPUId
-12	Invalid GateSMU
-13	Invalid DrainSMU
-14	Unable to initialize PIV solution
-15	Invalid GateSMU Range
-16	Invalid DrainSMU Range

## vg sid\_pulseiv

- Description** The vgsid\_pulse sweep is used to perform a pulsed Vg-Ig sweep using the 4200-PIV package. This test is similar to a typical DC Vg-Id but only two sources are used: gate (VPUId pulse channel 1) and drain (DrainSMU). The gate is pulsed, but the drain is DC biased. Measurements are made with the 2 channel scope card. Set the appropriate values for the Vgs-Id parameters ([Table 3-13](#)). [Table 3-14](#) and [Table 3-15](#) contain outputs and return values, respectively.
- Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the 4200-RBT. The RBT connected to GateSMU (with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. For detailed connection information, refer to the [PIV-A interconnect assembly procedure on page 3-33](#).



Table 3-13  
Inputs for vgsid\_pulseiv

Input	Type	Description
Vds	double	The drain-source voltage, output by the DrainSMU (defined below).
Vg_off	double	The DC bias applied by the GateSMU to put device in the OFF state. Normally set to 0 V for enhancement FETs (may be non-zero for depletion FETs).
VgStart	double	The starting sweep value for Vg, output by channel 1 of the pulse generator card (VPUID).
VgStop	double	The final sweep value for Vg, output by channel 1 of the pulse generator card (VPUID).
VgStep	double	The sweep step size for the Vg sweep, output by channel 1 of the pulse generator card (VPUID).
PulseWidth	double	The Vgs pulse width (PW). The PW can be 40 ns to 150 ns (10 ns resolution). Pulses wider than 150 ns will begin to be attenuated by the capacitor in the 4200-RBT.
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 100 $\mu$ s to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. The period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150 $\mu$ s.
AverageNum	int	The number of pulses to average at each step of the sweep. For best low signal performance, set AverageNum = 0 for Adaptive Filtering.
GateRange	double	The voltage measure range for the scope channel measuring the Gate. Use 0 for scope autoranging, or specify a voltage value for a fixed range. Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10.
DrainRange	double	The voltage measure range for the scope channel measuring the Drain. Use 0 for scope autoranging, or specify a voltage value for a fixed range, where $V = I * 50 \Omega$ . Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10.
LoadLineCorr	int	Determines whether to use load line correction to compensate for the voltage drop caused by the 50 $\Omega$ sense resistor used to measure the drain current ( $I_d$ ). 1 = load line correction active. 0 = no load line correction.
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with the 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SmUs in the system.
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SmUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.
IdSize VgMeasSize VgProgSize VdMeasSize VdProgSize	int	Set to a value that is at least equal to the number of steps in the sweep and all five must be the same value.



Table 3-14  
Outputs for Vgsid\_pulseiv

Output	Type	Description
VgMeas	double	Array of measured gate voltage values.
VgProg	double	Array of programmed gate voltage values.
VdMeas	double	Array of measured drain currents.
VdProg	double	Array of programmed drain voltages.
IdArray	double	Array of measured drain current values.

Table 3-15  
Return values for vgsid\_pulseiv

Value	Description
0	OK
-1	Invalid value for Vds
-2	Invalid value for VgStart
-3	Invalid value for VgStop
-4	Invalid value for VgStep
-5	Invalid value for PulseWidth
-6	Invalid value for PulsePeriod
-7	Invalid value for AverageNum
-8	Invalid value for LoadLineCorr
-9	Array sizes do not match
-10	Array sizes not large enough for sweep
-11	Invalid VPUId
-12	Invalid GateSMU
-13	Invalid DrainSMU
-14	Unable to initialize PIV solution

## Vgld\_DC\_Pulse\_pulseiv

**Description** The Vgld\_DC\_Pulse\_pulseiv sweep is used to perform a Pulsed vs DC Vg-Id sweep using the 4200-PIV-A package. This test is similar to a typical Vg-Id but only two sources are used: one for the DUT Gate and one for the DUT Drain. Pulsed Measurements are made with the 2-channel scope, 4200-SCP2.

This routine can run the sweeps in three different ways: 1) DC only; 2) Pulse only; 3) Pulse and DC curves. This routine supports from one to 10 Vd-Id curves based on up to 10 different Vgs values.

This routine also supports the 4200-PIV-A package using the 4200-RBT. For this package, all test parameters and limits are given below, except the 4200-PIV-A with the 4200-RBT has a max pulse width of 150 ns, not the 250 ns of the 4205-RBT.

All voltage levels specified below assume a 50  $\Omega$  DUT load.



**Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the 4205-RBT. The RBT connected to GateSMU (the RBT with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. Use either G-S-G probes for RF structures, or use DC probes with the 4200-PRB-C adapter cables for DC structures.

Set the appropriate values for the Vds-Id parameters. Inputs, outputs and returned values are provided in [Table 3-16](#), [Table 3-17](#) and [Table 3-18](#).

Table 3-16

**Inputs for Vgid\_DC\_Pulse\_pulseiv**

Input	Type	Description
Vds	double	The voltage value for Vd. For DC only sweeps, Vds must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT. For pulse and pulse and DC Sweeps, Vds must be between -5 V to +5 V.
VgStart	double	The starting step value for Vg. For DC only sweeps, VgStart must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT. For pulse and pulse and DC Sweeps, VgStart must be between -5 V to +5 V.
VgStop	double	The final step value for Vg. For DC only sweeps, VgStop must be between -200 V to +200 V dependent on the type of SMU and the current requirements of the DUT. For pulse and pulse and DC Sweeps, VgStop must be between -5 V to +5 V.
VgStep	double	The sweep step size for the Vg sweep, output by channel 1 of the pulse card (VPUID).
Vg_off	double	The DC bias applied by the GateSMU to put device in the OFF state. Normally set to 0 V for enhancement FETs (may be non-zero for depletion FETs). This package does not support a similar capability for the drain. For full quiescent, or bias, point testing, review the 4200-PIV-Q specs.
PulseWidth	double	The Vgs pulse width (PW). The PW can be 40 ns to 250 ns (10 ns resolution). Pulses wider than 250 ns will begin to be attenuated by the coupling capacitor in the Remote Bias Tee (4205-RBT).
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 100 $\mu$ s to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. The period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150 $\mu$ s.
AverageNum	int	The number of pulses to average at each step of the sweep. For best low signal performance, set AverageNum = 0 for Adaptive Filtering.
GateScpRange	double	The voltage measure range for the scope channel measuring the Gate. Use 0 for scope autoranging, or specify a voltage value for a fixed range. Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10. These ranges are Vpp. For example, the 0.5 range covers -250 to +250 mV.
DrainScpRange	double	The voltage measure range for the scope channel measuring the Drain. Use 0 for scope autoranging, or specify a voltage value for a fixed range, where $V = I * 50 \Omega$ . Valid voltages are 0.050, 0.1, 0.2, 0.5, 1, 2, 5, 10. These ranges are Vpp. For example, the 0.5 range covers -250 to +250 mV.



Table 3-16 (continued)  
Inputs for Vgid\_DC\_Pulse\_pulseiv

Input	Type	Description
GateSMURange	int	The current measurement range to be used for the SMU on the DUT Gate terminal. Values correspond to the table below. Limited Auto means that the value given is the minimum measurement range used, with automatic ranging for larger currents. 1 Full Auto 2 Limited Auto 10 pA 3 Limited Auto 100 pA 4 Limited Auto 1 nA 5 Limited Auto 10 nA 6 Limited Auto 100 nA 7 Limited Auto 1 uA 8 Limited Auto 10 uA 9 Limited Auto 100 uA 10 Limited Auto 1 mA 11 Limited Auto 10 mA 12 Limited Auto 100 mA
DrainSMURange	int	The current measurement range to be used for the SMU on the DUT Drain terminal. Values correspond to the table below. Limited Auto means that the value given is the minimum measurement range used, with automatic ranging for larger currents. 1 Full Auto 2 Limited Auto 10 pA 3 Limited Auto 100 pA 4 Limited Auto 1 nA 5 Limited Auto 10 nA 6 Limited Auto 100 nA 7 Limited Auto 1 uA 8 Limited Auto 10 uA 9 Limited Auto 100 uA 10 Limited Auto 1 mA 11 Limited Auto 10 mA 12 Limited Auto 100 mA
LoadLineCorr	int	Determines whether to use load line correction to compensate for the voltage drop caused by the 50 $\Omega$ sense resistor used to measure the drain current (Id). 1 = load line correction active. 0 = no load line correction.
GateCompliance	double	The SMU current compliance for the DUT Gate.
DrainCompliance	double	The SMU current compliance for the DUT Drain.
NPLC	double	The DC measurement integration time in NPLC (Number of Power Line cycles).
DCSourceDelay	double	Time, in seconds, between the DC source and measure for each sweep point.
DC_vs_Pulse	int	Determines whether to run a DC and Pulse test or a DC only test or a Pulse only test. 0 - Pulse Only, 1 - DC Only, 2 - DC and Pulse.
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with a 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SMUs in the system.



Table 3-16 (continued)  
**Inputs for Vgid\_DC\_Pulse\_pulseiv**

Input	Type	Description
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SMUs in the system. This is the SMU that applies the DC bias to the DUT drain during the pulse or DC sweep.
DrainI_DC_X_Size DrainVMeas_DC_Size DrainVProg_DC_Size GateVMeas_DC_X GateVProg_DC_Size DrainISize_Pulse_Size DrainVMeas_Pulse_Size DrainVProg_Pulse_Size GateVSize_Pulse_Size GateVProg_Pulse_Size	int	These values *must* be set to a value that is at least equal to the number of steps in the sweep and all values must be the same.

Table 3-17  
**Outputs for Vgid\_DC\_Pulse\_pulseiv**

Output	Type	Description
DrainI_DC/Pulse	double	The measured drain current from channel 2 of the 4200-SCP2 or the DrainSMU. In the case of Pulse, this current is determined by measuring the voltage drop across the 4200-SCP2 50 $\Omega$ termination, giving $I_d = V_d / 50 \Omega$
DrainVMeas_DC/Pulse	double	The measured drain voltage from channel 2 of the 4200-SCP2 in the case of pulse and the measured voltage on the DrainSMU in the case of DC.
DrainV_Prog_DC/Pulse	double	The programmed drain voltage, either supplied by the pulse card or the SMU for the drain.
GateVMeas_DC/Pulse	double	The measure gate voltage from channel 1 of the 4200-SCP2 in the case of pulse and the measured voltage on the GateSMU in the case of DC.
GateVProg_DC/Pulse	double	The programmed gate voltage, either supplied by the pulse card or Gate SMU.

Table 3-18  
**Return values for Vgid\_DC\_Pulse\_pulseiv**

Value	Description
0	OK
-1	Invalid value for Vds
-2	Invalid value for VgStart
-3	Invalid value for VgStop
-4	Invalid value for VgStep
-5	Invalid value for PulseWidth
-6	Invalid value for PulsePeriod



Table 3-18 (continued)  
Return values for Vgid\_DC\_Pulse\_pulseiv

Value	Description
-7	Invalid value for AverageNum
-8	Invalid value for LoadLineCorr
-9	Array sizes do not match
-10	Array sizes not large enough for sweep
-11	Invalid VPUId
-12	Invalid GateSMU
-13	Invalid DrainSMU
-14	Unable to initialize PIV solution
-15	Invalid GateSMU Range
-16	Invalid DrainSMU Range

## scopeshot\_cal\_pulseiv

**Description** The scopeshot\_cal\_pulseiv routine is used to display a single Pulse IV scopeshot\_pulseiv. This routine is useful to understand the basic source and measure concepts behind the Pulse IV methods for pulse vds-id and vgs-id. Measurements are made with cable compensation values applied to them and load line compensation can be used if desired.

**Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the RBT. The RBT connected to GateSMU (with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain.

Set the appropriate values for the scopeshot\_cal\_pulseiv. Set the appropriate values for the Vds-Id parameters. Inputs, outputs and returned values are provided in [Table 3-19](#), [Table 3-20](#) and [Table 3-21](#).

Table 3-19  
Inputs for scopeshot\_cal\_pulseiv

Input	Type	Description
Vds	double	The DC drain bias, provided by the DrainSMU.
Vgs	double	The pulse gate voltage amplitude. This can be set from -5 to +5 V.
VgStart	double	The starting sweep value for Vg, output by channel 1 of the pulse generator card (VPUID).
PulseWidth	double	The Vgs pulse width (PW). The PW can be 40 ns to 300 ns (10 ns resolution).
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 100 us to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. This period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150 us.
AverageNum	int	The number of pulses to average at each step of the sweep. For best low signal performance, set AverageNum = 0 for Adaptive Filtering.



Table 3-19 (continued)

**Inputs for scopeshot\_cal\_pulseiv**

Input	Type	Description
GateRange	double	The voltage measure range for the scope channel measuring the Gate. Use 0 for scope autoranging, or specify a voltage value for a fixed range. Valid voltages are 0.25, 0.5, 1.25, 2.5, 5, 10, 25, 50. The range is a full range value (for example, 2.5 is -1.25 V to +1.25 V).
LoadLineCorr	int	Determines whether to use load line correction to compensate for the voltage drop caused by the DUT impedance on the Drain. When load line correction is on (1), the test will start by assuming a high impedance value for the device and will approach the correct bias and pulse values over a series of pulses, that ensures that the sourced pulses match the requested values. When load line correction is turned off, the specified voltages will be sourced (1 = Use Load Line, 0 = No Load Line).
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with the 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SmUs in the system.
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SmUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.
TimeSize GatePulseSize DrainPulseSize	int	These values must be set to a GatePulseSize value that is at least equal to the DrainPulseSize number of steps in the sweep and all three must be the same value.

Table 3-20

**Outputs for scopeshot\_cal\_pulseiv**

Output	Type	Description
Time	double *	Array of time values from the 4200-SCP2 scope (s).
GatePulse	double *	Array of gate pulse voltages from channel 1 of the 4200-SCP2 scope.
DrainPulse	double *	Array of drain voltages from channel 2 of the 4200-SCP2 scope.
VgMeas	double *	Measured Gate Voltage.
VdMeas	double *	Measured Drain Voltage.
IdMeas	double *	Measured Gate Current.

Table 3-21

**Return values for scopeshot\_cal\_pulseiv**

Value	Description
0	OK
-1	Invalid Gate Voltage (Max 5 V).
-2	Invalid Drain Voltage (Max 210 V).
-5	Invalid Pulse Width (Min 40 ns).
-6	Invalid Pulse Period (Min 40 ns).



Table 3-21 (continued)  
**Return values for scopeshot\_cal\_pulseiv**

Value	Description
-7	Invalid Average Num (must be between 1 and 100000).
-8	Invalid LoadLineCorr (must be between 0 and 1).
-9	Time, GatePulse, and Drain Pulse array sizes must be equal.
-11	Invalid VPU. Specified VPU is not in current system configuration.
-12	Invalid GateSMU. Specified SMU is not in current system configuration.
-13	Invalid DrainSMU. Specified SMU is not in current system configuration.
-14	PIV Initialization Failed.

## scopeshot\_pulseiv

- Description** The scopeshot\_pulseiv routine displays a single Pulse IV scopeshot. This routine is useful to understand the basic source and measure concepts behind the Pulse IV methods for pulse vds-id and vgs-id. The scope waveforms are retrieved and displayed for both channels (no measurements are made). Make sure to set the appropriate values for the scopeshot\_pulseiv (see [Table 3-22](#)). [Table 3-23](#) and [Table 3-24](#) contain outputs and return values, respectively.
- Connection** The source and body (well) of the DUT must be shorted together and connected to the common low (outer shield) of the SMA cables on the AC+DC output of the 4200-RBT. The RBT connected to GateSMU (with the Power Divider) should be connected to the gate. The RBT connected to DrainSMU should be connected to the drain. For detailed connection information, refer to the [PIV-A interconnect assembly procedure on page 3-33](#).



Table 3-22  
Inputs for scopeshot\_pulseiv

Input	Type	Description
RiseTime	double	The gate pulse transition rise time (s). This can be set from 10 e-9 to 300 e-9 in 10 e-9 (10 ns) steps. This value programs the full transition time (0–100%), not the 10–90% time.
FallTime	double	The gate pulse transition fall time (s). This can be set from 10 e-9 to 300 e-9 in 10 e-9 (10 ns) steps. This value programs the full transition time (0–100%), not the 10–90% time.
PulseWidth	double	The gate pulse width (PW). The PW can be 20 ns to 1us (10 ns resolution). Pulses wider than 150 ns will begin to be attenuated by the capacitor in the 4200-RBT.
PulseBase	double	The pulse gate base voltage. This can be set from -5 to +5 V, inclusive of amplitude.
PulseAmplitude	double	The pulse gate voltage amplitude. This can be set from -5 to +5 V, inclusive of base voltage.
GateLoad	double	The scope card channel 1 input impedance for the gate. Either 50 or 1E6. Use 50 for Pulse IV with RBTs.
GateRange	double	The scope card channel 1 Y scale voltage range for the gate measurement. Typical values are 1, 2, 5 V.
DrainLoad	double	The scope card channel 2 input impedance for the drain. Either 50 or 1E6. Use 50 for Pulse IV with RBTs.
DrainRange	double	The scope card channel 2 Y scale voltage range for the drain measurement. Typical values are 1, 2, 5 V.
PulsePeriod	double	The pulse period for the Vgs pulse. The period can be set from 40 ns to 1 s (10 ns resolution). The period must be set so that the Duty Cycle (DC) is no more than 0.1%. This period is most easily calculated by multiplying the largest desired pulse width (PW) by 1000. Example: PW = 150 ns, so Period = 150 us.
AverageNum	int	The number of waveforms to average.
GateBias	double	The DC gate bias, provided by the gateSMU.
DrainBias	double	The DC drain bias, provided by the drainSMU.
VPUID	char *	The instrument ID. This should be set to VPU1 for 4200 systems with the 4200-PIV package.
GateSMU	char *	The SMU used for the Gate. This can be SMU1 up to the maximum number of SmUs in the system.
DrainSMU	char *	The SMU used for the Drain. This can be SMU1 up to the maximum number of SMUs in the system. This is the SMU that applies the DC bias to the DUT drain during the sweep.
TimeSize Ch1OutSize Ch2OutSize	int	Set to a value that is at least equal to the number of steps in the sweep and all three must be the same value.

Table 3-23  
Outputs for scopeshot\_pulseiv

Output	Type	Description
Time	double *	Array of time values from the scope card (s).
Ch1Out	double *	Array of gate voltages from channel 1 of the scope card.
Ch2Out	double *	Array of drain voltages from channel 2 of the scope card.



Table 3-24  
Return values for scopeshot\_pulseiv

Value	Description
0	OK
-1	Invalid Pulse Width (Min 40 ns)
-2	Invalid Pulse Period (Min 40 ns)
-3	Invalid Average Num (1 - 1000)
-4	Array Sizes Do Not Match
-5	Invalid VPU. Specified VPU Is Not In Current System Configuration
-6	Invalid GateSMU. Specified SMU Is Not In Current System Configuration
-7	Invalid DrainSMU. Specified SMU Is Not In Current System Configuration
Negative numbers are errors—refer to LPT and PulseIV documentation for description.	

## vdsid\_pulseiv\_demo

(Also see [vdsid\\_pulseiv](#) on page 3-48)

## vgsid\_pulseiv\_demo

(Also see [vgsid\\_pulseiv](#) on page 3-53)

## scopeshot\_pulseiv\_demo

(Also see [scopeshot\\_pulseiv](#) on page 3-61)

These three UTMs are functionally identical but simpler than their respective routines listed earlier in this section of the manual. The difference being less-used parameters have been eliminated from the parameter list and hard-coded (for example, SMU channels, ranges, load line).



# How to perform a Quiescent-point Pulsed I-V test (PIV-Q) on my device

## Q-Point Pulse IV – Model 4200-PIV-Q

This application provides q-point pulse IV testing for higher power compound semiconductor or LDMOS RF transistors, or any device may benefit from low duty cycle pulse IV testing. This application uses the Model 4200-PIV-Q package, that differs from the Model 4200-PIV-A package by:

- 1) Pulsing both the gate and drain
- 2) Providing higher power to the DUT drain
- 3) Pulsing from non-zero bias point, or quiescent point, testing

The compound semiconductor transistors consist of materials made from the III-V groups in the periodic table of the elements, but other groups or elements may be used.

## What is the PIV-Q package

The PIV-Q package is an optional factory-installed kit to the Model 4200-SCS. The focus for the PIV package is testing RF FETs that exhibit self-heating or charge trapping effects (also called dispersion).

To accomplish pulse IV testing of LDMOS and compound semiconductor FETs, the PIV-Q package consists of the following:

- Model 4205-PG2 Dual channel voltage pulse generator (Qty: three).
- Model 4200-SCP2HR Dual channel oscilloscope.
- 4205-PCU Pulse Combiner Unit to create one higher power pulse channel for the DUT drain by combining the four pulse channels from two 4205-PG2 cards.
- Pulse IV Interconnect adapters and cabling.
- Pulse IV software - Projects and test routines for testing of RF FETs, including cable compensation and load-line algorithms to provide DC-like sweep results.

## Target applications and test projects

The PIV-Q package includes test projects that address the most common parametric transistor tests:

- $V_{ds}$ -id
- $V_{gs}$ -id

These tests are provided in both DC and Pulse modes, allowing correlation between the two test methods.

These tests, as well as initialization steps for scope auto-calibration and cable compensation, are included in a single Model 4200 test project, QPulse-IV-Complete.

There are two methods for performing DC IV sweeps.

- Use the SMUs to provide the DC source and measure:  $V_{d-ID-DC}$ ,  $V_{g-ID-DC}$ . Because SMU output impedance varies with the source and measure ranges, high gain or high frequency DUTs may be difficult to test because of oscillation
- Pulse IV instrumentation may also be used to provide DC-like IV sweeps. Using the pulse source and measure hardware provides a fixed output impedance, which can provide better DC test results on high gain or high frequency DUTs that are prone to oscillation.



Use the desired pulse test and set the duty cycle from the typical low values up to 90+% to mimic a DC test by choosing a longer period, up to one second, with appropriate pulse width and transitions.

---

**NOTE** *For further information on how to use Model 4200-PIV-Q, refer to Application Note PA-956 Rev. B, Model 4200-PIV-Q Pulse IV Measurements for Compound Semiconductor and LDMOS Transistors. This and all 4200 related notes can be found on the Model 4200-SCS Complete Reference. See [QPulseIV-Complete project](#) in Section 12 of the Reference Manual for details on using the PIV-Q package.*

---



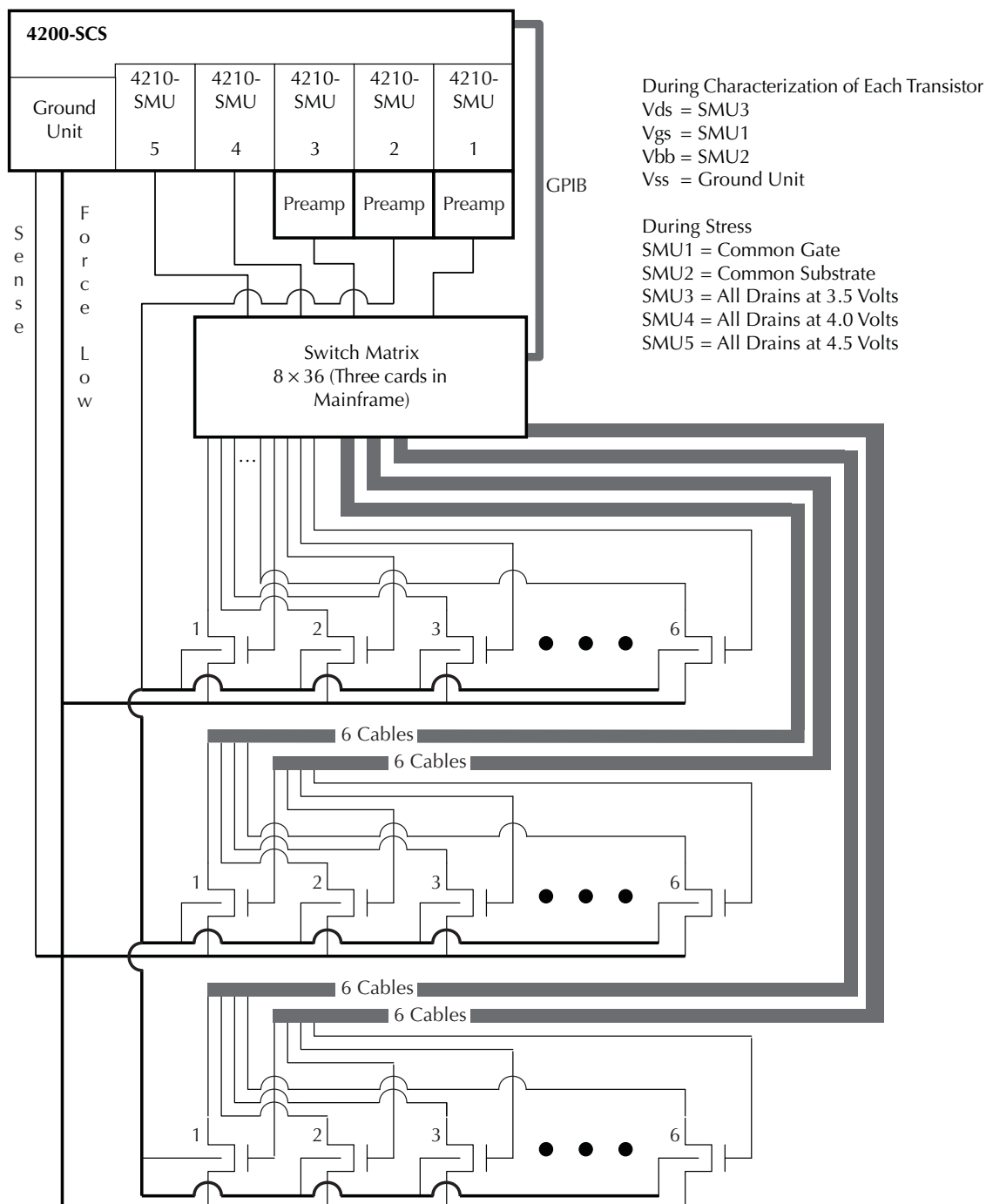
## How to perform a reliability (stress-measure) tests on my device

### Connecting devices for stress / measure cycling

Devices that are stress / measure cycled in parallel are connected through a switch matrix. [Figure 3-49](#) shows an example of such connections for an HCI evaluation.

Figure 3-49

#### Stress / measure wiring example

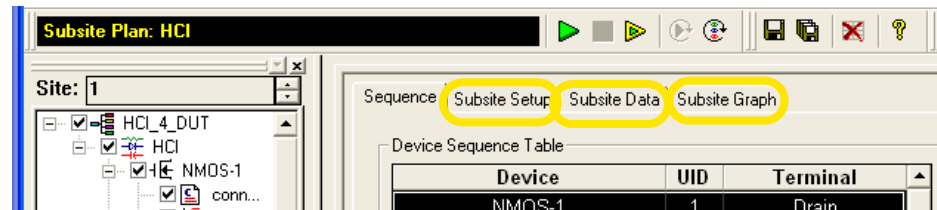




## Overviewing the cycling-related tabs

When you double-click the name of a subsite, KITE displays the tabs shown in [Figure 3-50](#). Use the Subsite Setup tab to configure cyclical tests, the Subsite Data tab to view test results numerically, and the Subsite Graph tab to view results graphically.

Figure 3-50  
General tab overview



## Configuring subsite cycling

### Understanding the Subsite Setup tab

Figure 3-51  
Subsite setup tab

Use to enable or disable cycling. If **Enable Cycles** is unchecked, KITE ignores the **Subsite Setup** tab settings and executes the Subsite Plan without cycling.

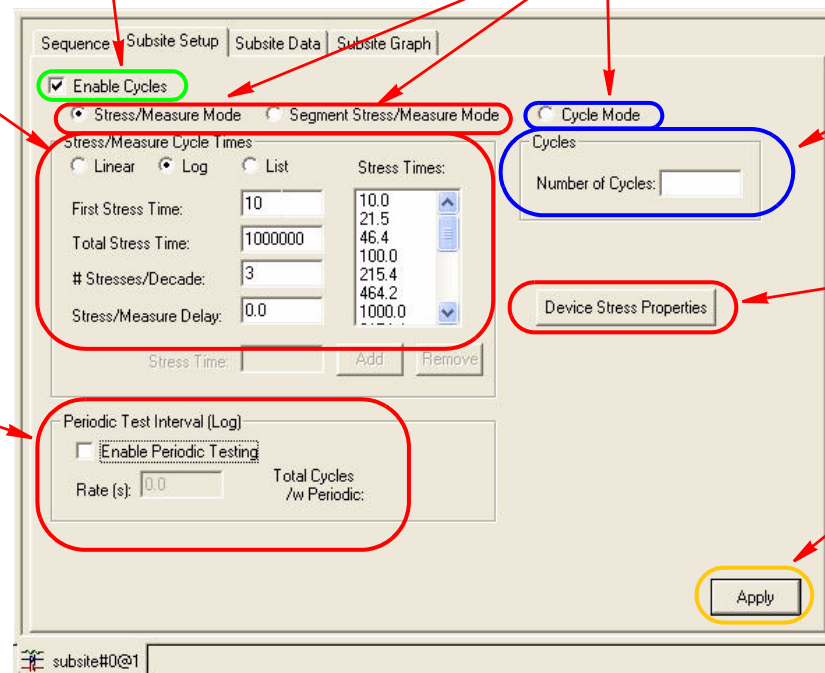
Use to choose one of the following cycling modes:

- Stress ↔ measure cycles: for example, HCI tests: by selecting a stress / measure mode.
- Only measure cycles: no stressing: by selecting **Cycle Mode**.

When **Stress/Measure Mode** is selected, use to set up cycle timing:

- **Linear**, if you want all stress times to be the same
- **Log**, if you want the stress time to increase logarithmically with each successive cycle.
- **Log**, if you want to specify each time.

When **Stress / Measure Mode** AND **Log** cycle times are selected, use to "ask" KITE to stop stressing and make measurements at the intervals specified in the **Rate(s)** field: *in addition to any intervals specified in the **Stress / Measure Cycle Times** area.* For more information, refer to [Step D: Set periodic test intervals \(stress / measure mode, log timing only\)](#) on page 3-70.



When **Cycle Mode** is selected, enter here the fixed number of times (cycles) that you want the subsite to execute (up to 128, maximum). No stressing is performed when in cycle mode.

When **Stress / Measure Mode** is selected, use to open the Device Stress Properties window, which is used to configure the subsite stressing parameters.

When a stress / measure mode is selected, use to apply and save the settings you have entered in the Setup tab. This button also applies and saves settings made in the Device Stress Properties window (see [Figure 3-59](#)).

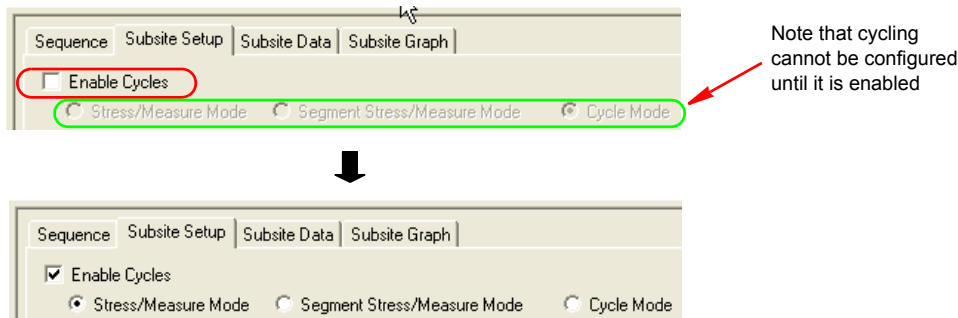


## Configuring the Subsite Setup tab

To set up cyclical testing, configure the Subsite Setup tab as shown in steps A through E below.

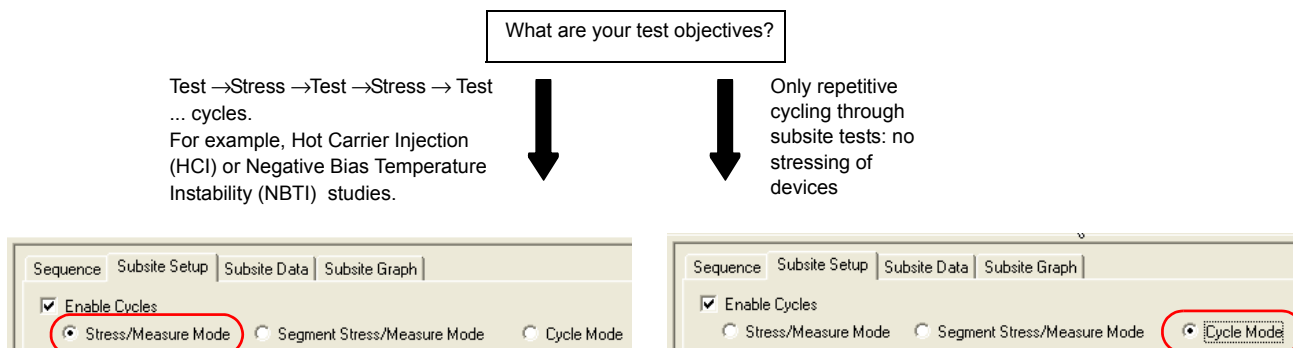
### Step A: Enable cycling

Figure 3-52  
Enabling cycling



### Step B: Choose the mode of cycling

Figure 3-53  
Specifying the mode of cycling (Stress / Measure Mode or Cycle Mode)



**NOTE** Only Cycle Mode is supported with Model 4225-RPMs present in the system. The Stress/Measure and Segment Stress/Measure modes are NOT supported if RPMs are present in the system. In order to run Stress/Measure and Segment Stress/Measure modes, all RPMs must be disconnected from the Model 4200-SCS and you must update the RPM configuration in KCON (see [Tools > Update DC Preamp and RPM Configuration](#) in Section 7).

### Step C1: Specify cycle timing; linear, log or list (stress / measure mode only)

[Figure 3-54](#) explains how to set timing for the linear and log modes, while [Figure 3-55](#) explains how to set timing for the list mode.



Figure 3-54  
Specifying timing (linear and log) for stress / measure mode

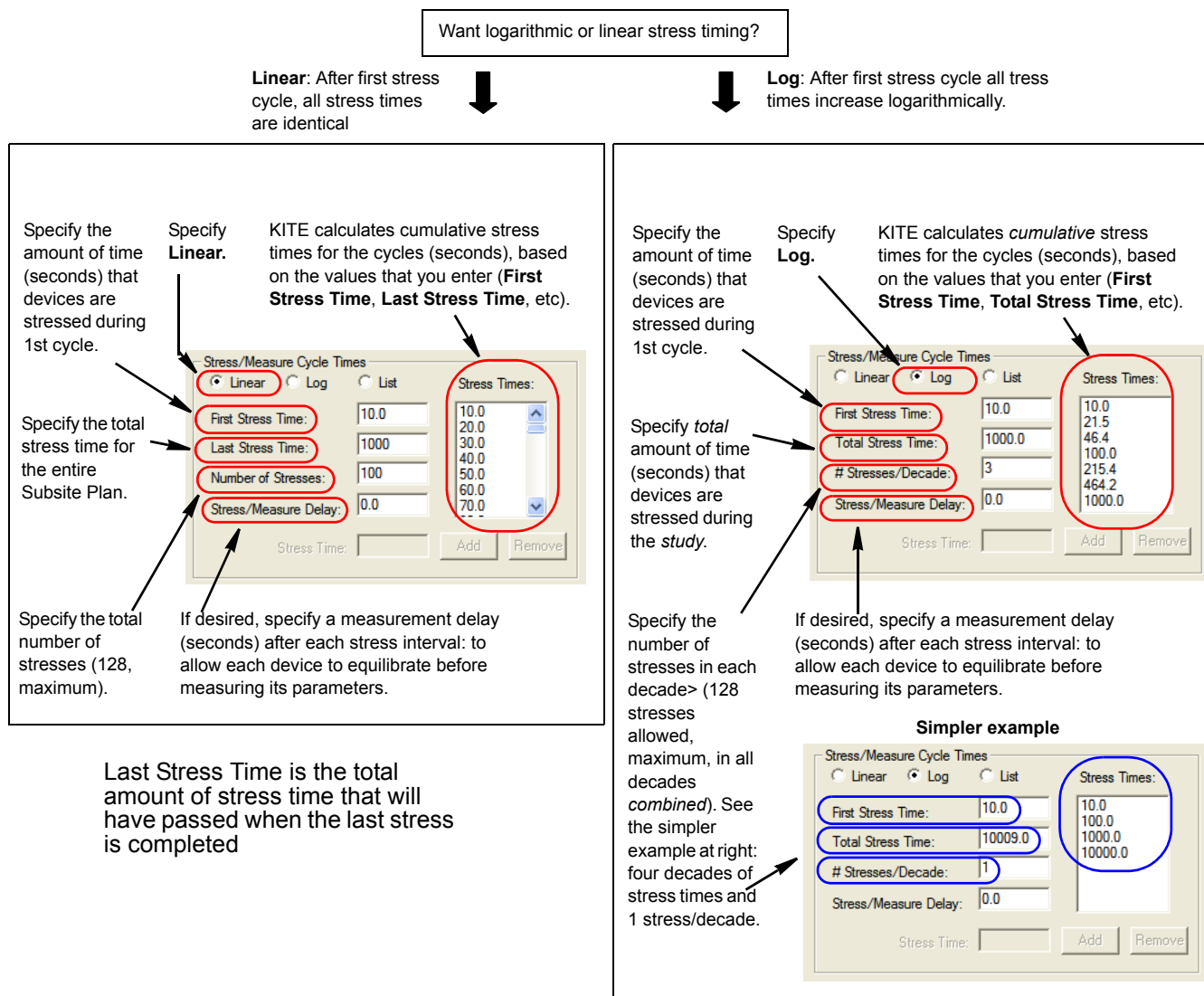
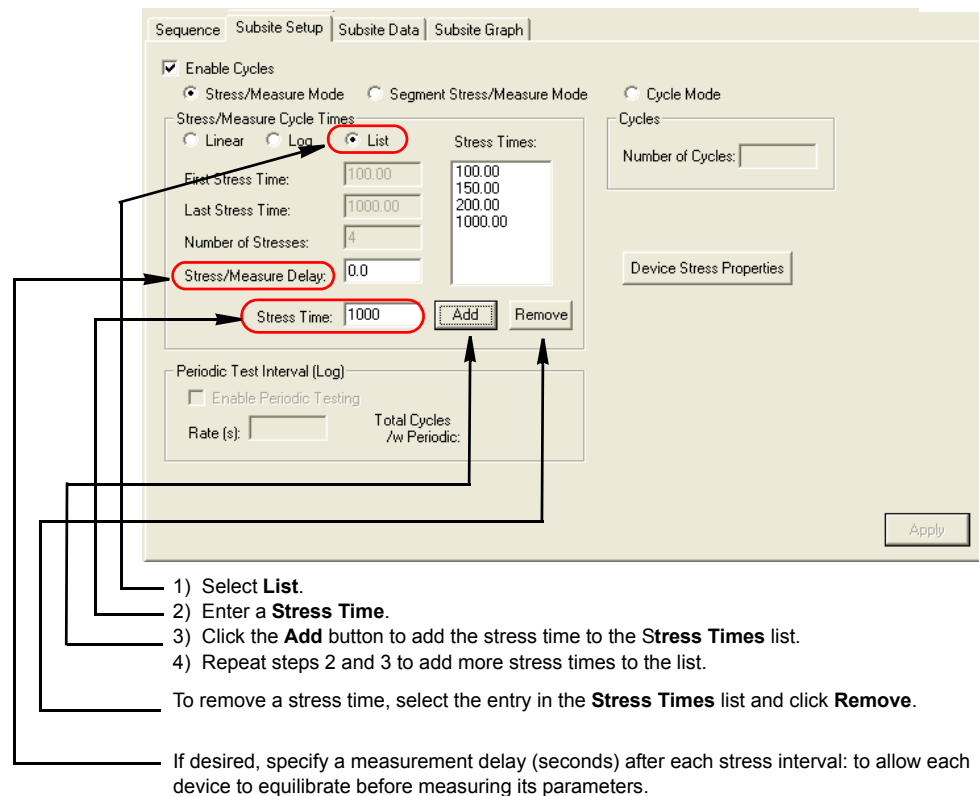


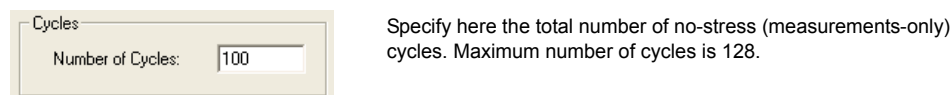


Figure 3-55  
Specifying timing (List)



### Step C2: Specify number of cycles (cycle mode only)

Figure 3-56  
Specifying the number of cycles



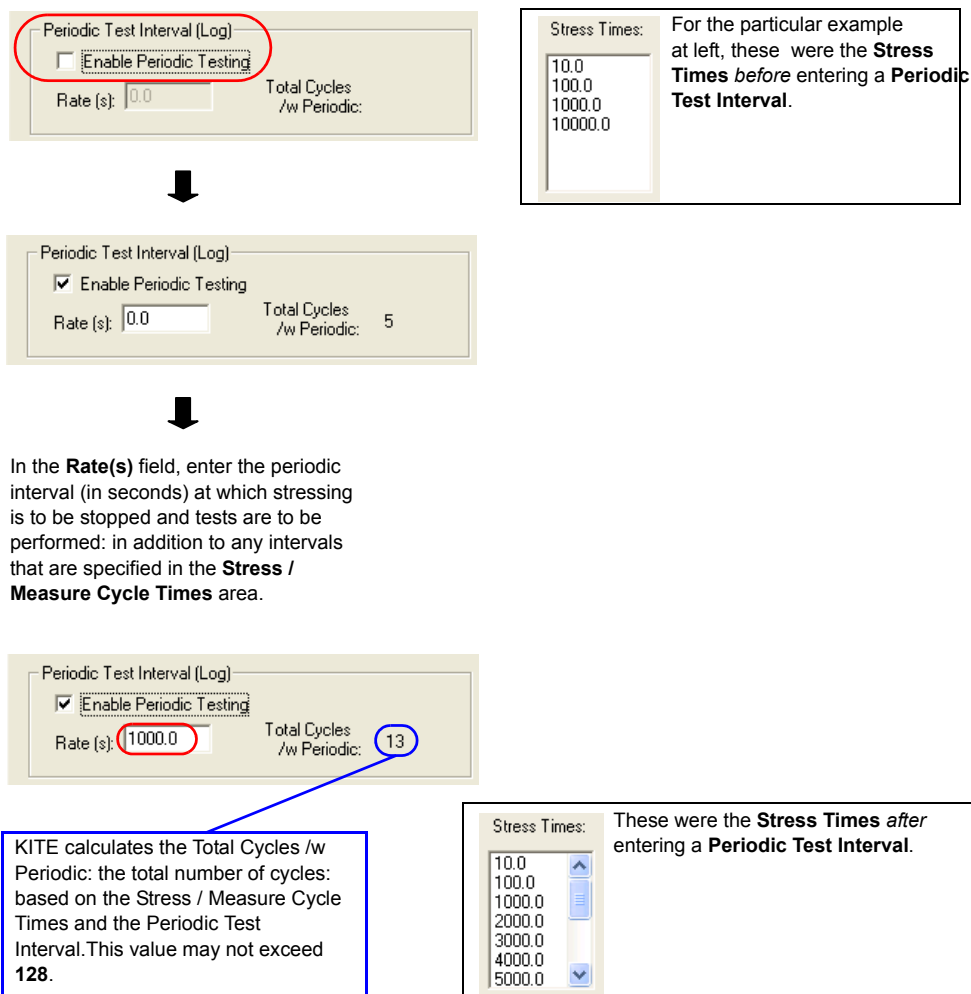
### Step D: Set periodic test intervals (stress / measure mode, log timing only)

In the **Periodic Test Interval (Log)** area you can specify uniform, periodic intervals at which to interrupt the stress to perform tests. These are in addition to the intervals specified under **Stress / Measure Cycle Times**. Specify these intervals as shown in [Figure 3-57](#).

**NOTE** You can use the **Periodic Test Interval (Log)** area only if you select **Log** in the **Stress / Measure Cycle Times** area (the **Periodic Test Interval (Log)** area is disabled if you select **Linear**).



Figure 3-57  
Setting periodic test intervals

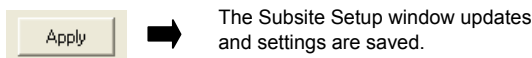


### Step E: Update and save the subsite setup configuration

Update the window and save your settings or setting changes by clicking the **Apply** button.

**NOTE** Subsite Setup tab calculations do not execute / update until you click **Apply** (see [Figure 3-58](#)).

Figure 3-58  
Saving the subsite setup configuration



## Configuring device stress properties

Properties for device stressing are set from the Device Stress Properties windows (see [Figure 3-59](#)). This window is opened by clicking the **Device Stress Properties** button on the Subsite Setup tab. The button is shown in [Figure 3-51](#).



The device stress properties window that opens will be for the first device in the subsite plan. There is a separate properties window for each device in the plan. The properties window for each device is displayed by clicking the **Next Device** or **Prev Device** button at the bottom of the window. The basic steps to configure each device are provided in [Figure 3-59](#). For details about configuration follow the illustration.

Figure 3-59

**Device Stress Properties: Setup steps for first device in Subsite Plan**

1) Select the wafer site number.  
See [Multi-site testing](#)

2) Select DC Voltage Stress, DC Current Stress or AC Voltage Stress and then enter the stress values (V or I) and limit values (I or V).  
See [DC voltage, DC current, or AC voltage stressing](#)

3) Assign connection pin numbers for this device. With "AC Voltage Stress" selected, click "VPU" checkbox if connected to the Model 4205-PG2.  
See [Device pin connections](#)

4) Degradation targets: Lists the tests and Output Values for this device. Targets can be enabled and the Target Values can be set (in % or Absolute Value).  
Click to enable or disable all Targets.  
See [Degradation targets](#)

5) Use the pull-down menus to control Stress Measurements. Options include:  
Do Not Measure  
First Stress Only  
Every Stress Cycle  
See [Stress measurements](#).

6) For a multi-device Subsite Plan, click **Next Device** to display the stress properties window for the next device.  
Clicking **Prev Device** selects the previous device.  
See [Device selection](#).

7) Repeat steps 2 through 6 for all devices in the Subsite Plan.

8) Repeat steps 1 through 7 to configure the Device Stress Properties for another site.

9) When active, enable to leave the outputs on after the end of a stress cycle.  
See [Leave stress conditions on](#).

10) Click to check SMU or VPU resources and connections for the active (displayed) site. Resource Status will indicate if enough SMUs and VPUs are present. Connections status will report any problems with matrix connections.  
See [Device pin connections](#).

11) When finished setting the stress properties for all devices and all sites, click **OK**.

12) In the Subsite Plan tab, (see [Figure 3-51](#)), click **Apply** to apply and save the settings made in the Device Stress Properties window.

This button appears only when **AC Voltage Stress** is selected. Click this button to open the window to make common settings for the Keithley pulse card. See [Setting AC stress properties](#)

See [Clear, copy, paste, and paste to all sites](#)

Click to cancel all changes and close window.

**NOTE** After setting the device stress properties for all devices and sites, steps 10 and 11 of the above procedure must be performed in order to apply and save the new settings. Failure to do so will cause the new settings to be lost.



---

**NOTE** *In the Device Stress Properties windows, the names for device terminals (for example, Drain, Gate, Source and Bulk) and the enabled fields for those terminals are set automatically by KITE. The terminal names correspond to the terminal names used by the ITMs for the device. When you double-click an ITM in the project navigator for the device, it will show the schematic of the device and the names of the terminals.*

---

## Setting AC stress properties

With the **AC voltage stress** type selected, DC and AC stress properties can be set. [Figure 3-60](#) shows an example for a device that is connected directly (no matrix) to channel 2 of the VPU (AC stress). The device is also connected to SMU1 for DC stress.

- **Device Pin / SMU connections:**

In this area of the window, the 2 pin assignment and the checked box for **VPU** indicates that the gate is connected to channel 2 of the VPU.

---

**NOTE** *If the gate is connected to Channel 1 instead of the VPU, the pin assignment must be set to 1.*

---

- The 1 pin assignment, and the unchecked box for VPU indicates that the drain is connected to SMU1. The 0 assignments for the source and bulk indicate no connection.

- **Stress conditions:**

In this area of the window, the AC stress voltage on the gate is set to 1 V. This is the high level for VPU pulse output.

- The DC stress on the drain is set to 1 VDC. Since there are no connections for the source and bulk, these stress voltages are set to 0 V.
- The **Gate Duty Cycle** is set to 50%. This means the high level (1 V) pulse will be applied for half (50%) of the pulse period, and the low level will be applied for the other half.

- **Stress measurements:**

This area of the window indicates that no stress measurements will be made. **I Gate Stress** selection is disabled.

---

**NOTE** *The VPU does not measure in this instance.*

---

- The stress measurement settings for the source and bulk are not relevant since there are no SMUs connected to the device.

---

**NOTE** *The AC stress settings support multiple pulse cards in the chassis.*

---

- **VPU Common Settings:**

The rest of the settings for the VPU are made from the **VPU Common Settings** window. This window (shown in [Figure 3-61](#)) is opened by clicking the **VPU Common Settings** button at the top of Device Stress Properties window.

- The pulse low values for channel 1 and channel 2 are set from this window.
- Rise time, fall time, frequency and the impedance of the load are also set from this window.



- These settings apply to both channels of the VPU.

**NOTE** If a switch matrix is used, please refer to [How to perform AC stress for wafer level reliability \(WLR\) on page 3-75](#) for recommended VPU parameter settings.

**NOTE** The bandwidth of the interconnect, including any switch matrix, will determine the fastest rise / fall transition transmitted with minimal over- or under-shoot. The impedance of the device terminal affects both the stress and low level voltages. An oscilloscope may be used to ensure that the rise / fall times and voltage levels match the desired test parameters.

Figure 3-60  
**AC stress properties settings**

Device Stress Properties - Subsite: subsite Device: 4terminal-n-fet Site #1

Stress Conditions

Active Site: 1

VPU Common Settings

Drain Stress: 1 V

Source Stress: 0 V

NW Stress: V

SE Stress: V

Drain Limit: 0.1 A

Source Limit: 0.1 A

NW Limit: A

SE Limit: A

Gate Stress: 1 V

Bulk Stress: 0 V

SW Stress: V

NE Stress: V

Gate Duty Cycle: 50 %

Bulk Limit: 0.1 A

SW Limit: A

NE Limit: A

Device Pin / SMU Connections

Drain Pin: 1

VPU

Northwest Pin: VPU

Gate Pin: 2

VPU

Southwest Pin: VPU

Source Pin: 0

VPU

Southeast Pin: VPU

Bulk Pin: 0

VPU

Northeast Pin: VPU

Stress Measurements

I Drain Stress: Do Not Measure

NW Stress: Do Not Measure

I Gate Stress: Do Not Measure

SW Stress: Do Not Measure

I Source Stress: Do Not Measure

SE Stress: Do Not Measure

I Bulk Stress: Do Not Measure

NE Stress: Do Not Measure

Parameter Properties/Degradation Targets

Tests

Output Values

% Abs

Target

Target Value

Stress Type

DC Voltage Stress

DC Current Stress

AC Voltage Stress

Figure 3-61  
**VPU common settings window**

VPU Common Settings

VPU Common Settings

Rise Time: 100e-9 s V Low Value (ch1): 0.00 V

Fall Time: 100e-9 s V Low Value (ch2): 0.00 V

Frequency: 1e+6 Hz

Load Impedance: 50 Ω

OK Cancel



## How to perform AC stress for wafer level reliability (WLR)

AC, or pulsed, stress is a useful addition to the typical stress-measure tests for investigating both semiconductor charge trapping and degradation behaviors. NBTI (negative bias temperature instability) and TDDB (time dependent dielectric breakdown) tests consist of stress / measure cycles.

The applied stress voltage is a DC signal, that is used because it maps more easily to device models. However, incorporating pulsed stress testing provides additional data that permits a better understanding of device performance in frequency-dependent circuits.

---

**NOTE** Key test parameters are contained in [Table 3-25](#).

---

1. Connect pulse generator to DUT during stress as shown in [Figure 3-62](#), [Figure 3-63](#), and [Figure 3-64](#).
2. The test pulse stresses the device for HCI, NBTI and TDDB test instead of DC bias by outputting a train of pulses for a period of time (stress time). Pulse characteristics are not changed during the stress-measure test.
3. The test then measures device characteristics using SMUs:  $V_{th}$ ,  $G_m$ , and so on.

Figure 3-62

**AC Pulse stress-measure—hardware setup block diagram**

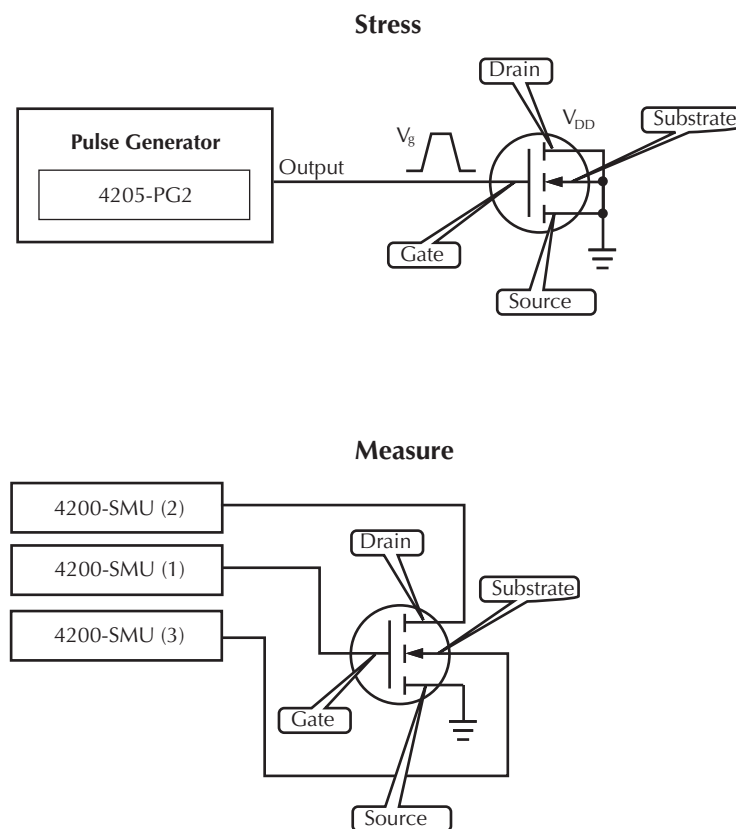




Figure 3-63  
**AC Pulse stress-measure—hardware matrix card simplified schematic**

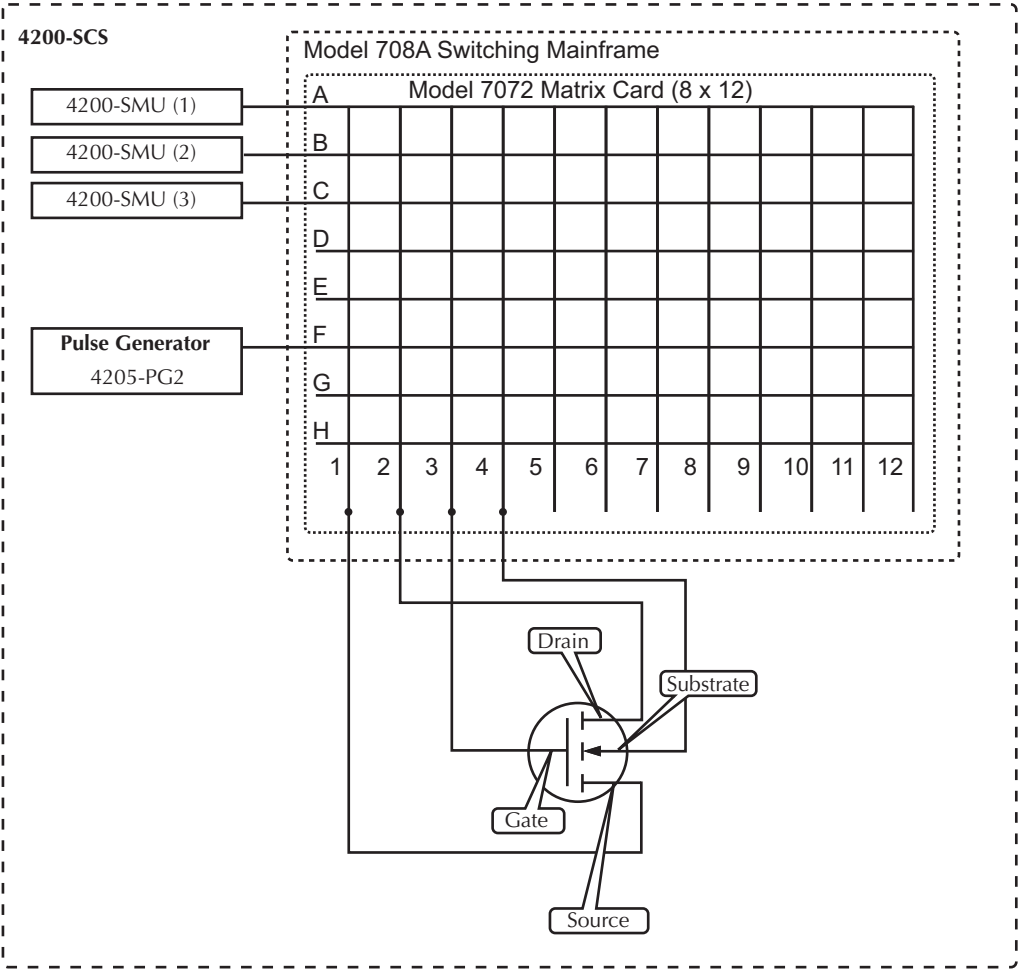




Figure 3-64  
AC Pulse stress-measure—hardware connections

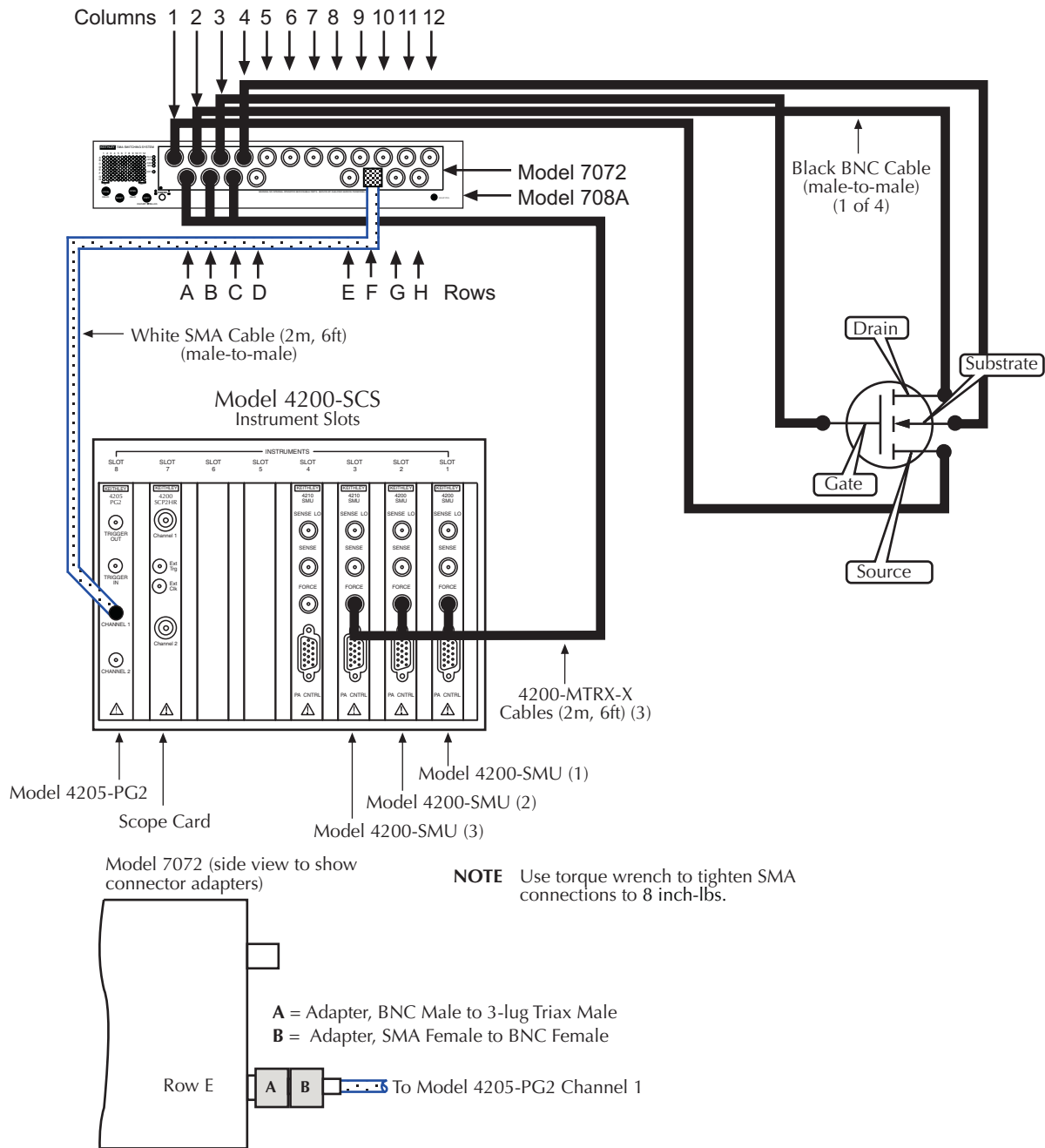


Table 3-25  
Key pulse generator parameters—AC stress for WLR

Parameters	Range / Specification
Rise / Fall time	Variable 100 ns–500μs
Pulse width	200 ns–1ms single pulse
Pulse amplitude	-5 to +5 V
Duty cycle	~50%
Base voltage	+/- 5 V



## Device Stress Properties configuration notes

The following information is supplemental to the procedure in [Figure 3-59](#) to configure Device Stress Properties:

### Multi-site testing

If your project is set up to run on more than one site you will need to set the Device Stress Properties for each site separately. This allows you to have different levels of stress on each site. After performing all the steps in [Figure 3-59](#) to configure the first site, repeat step 1 in [Figure 3-59](#) to select the next site. The COPY and PASTE buttons can be used to speed up the configuration process (see [Clear, copy, paste, and paste to all sites on page 3-80](#)).

### DC voltage, DC current, or AC voltage stressing

A device can be stressed with DC voltage or DC current using an SMU. When stressing with DC voltage, a current limit can be set, and when stressing with DC current, a voltage limit can be set. Limits are set to protect the device from damage.

One or two device terminals can also be stressed with AC voltage by each Keithley Instruments pulse card. Each pulse card has two output channels allowing two devices to be stressed by AC voltage.

---

**NOTE** *Current stressing: When setting the current stress level for each device in the subsite plan, keep in mind that a setting of zero (0) connects the device pin to the ground unit (0 V ground). In order to current stress a device, the current level must be set to a non-zero value.*

---

### Device pin connections

In the Device Stress Properties window (see [Figure 3-59](#)) there are input fields for device pin numbers. With DC Voltage Stress or DC Current Stress selected, the device pins are connected to an SMU or a matrix card. With AC Voltage Stress selected, checkboxes appear next to the device pin input fields which allow for the user to enable a VPU for each individual pin. If the pin is routed to an SMU, leave the corresponding VPU checkbox empty. If a pin is routed to a pulse card, click the corresponding checkbox to enter a checkmark.

**No matrix card system:** If a matrix card is not being used in the system, the pin number assignments for each device must match the actual physical connections to the SMUs. For example, if the drain of a device is connected to SMU2, the pin number assignment for Drain Pin in the Device Stress Properties window must be set to 2.

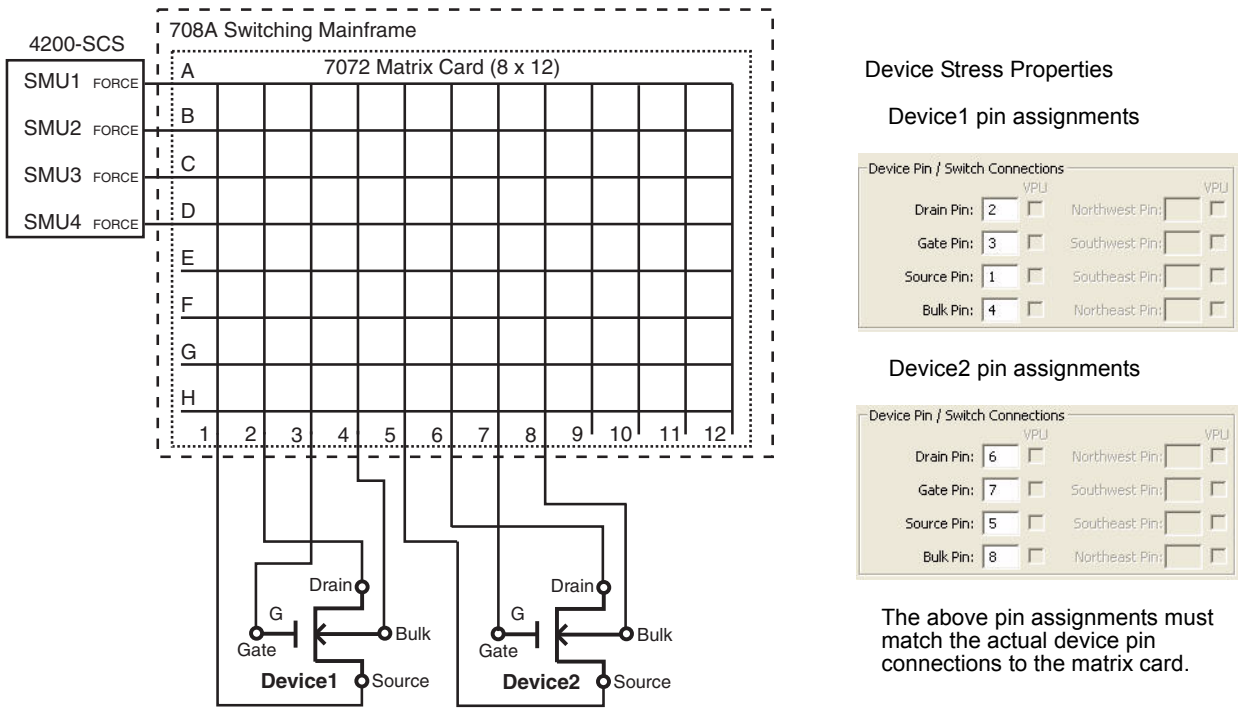
For a VPU with no matrix, assign value 1 to the device terminal that is connected to channel 1 of the VPU. If the device terminal is connected to VPU channel 2, assign it to value 2.

**Matrix card system:** For a system using a matrix card, the pin number assignments for each device must match the actual physical connections to the matrix card. [Figure 3-65](#) shows an example of how the Device Pin Connections properties must match the actual connections of the devices to the matrix card.

**SMUs:** If your voltage stress system is using a switch matrix, the Model 4200-SCS will try to maximize the amount of SMU sharing in order to allow parallel testing. It determines what pins can share SMUs in the following fashion. If pins from different devices have the same name (for example, Gate Pin, Drain Pin, etc). and the like named pins are assigned the same voltage stress, then when the stress is applied these pins will all be automatically connected to the same SMU through the switch matrix. That SMU will supply the voltage stress to all the pins simultaneously.



Figure 3-65  
Example of device pin connections to a matrix card



Degradation targets

Tests and Output Values for the device are listed in this area of the properties window. Post-stress Output Value readings are compared to the first cycle pre-stress readings. The % Change between the pre-stress and post-stress readings are listed in Subsite Data sheet.

An Output Value can be enabled (checked) as a Target and assigned a Target Value (in % or an absolute value). When all Targets for a device are reached, that device will not be tested for subsequent cycles. The Subsite Plan will stop when all enabled targets are reached or the last subsite cycle is completed.

Stress measurements

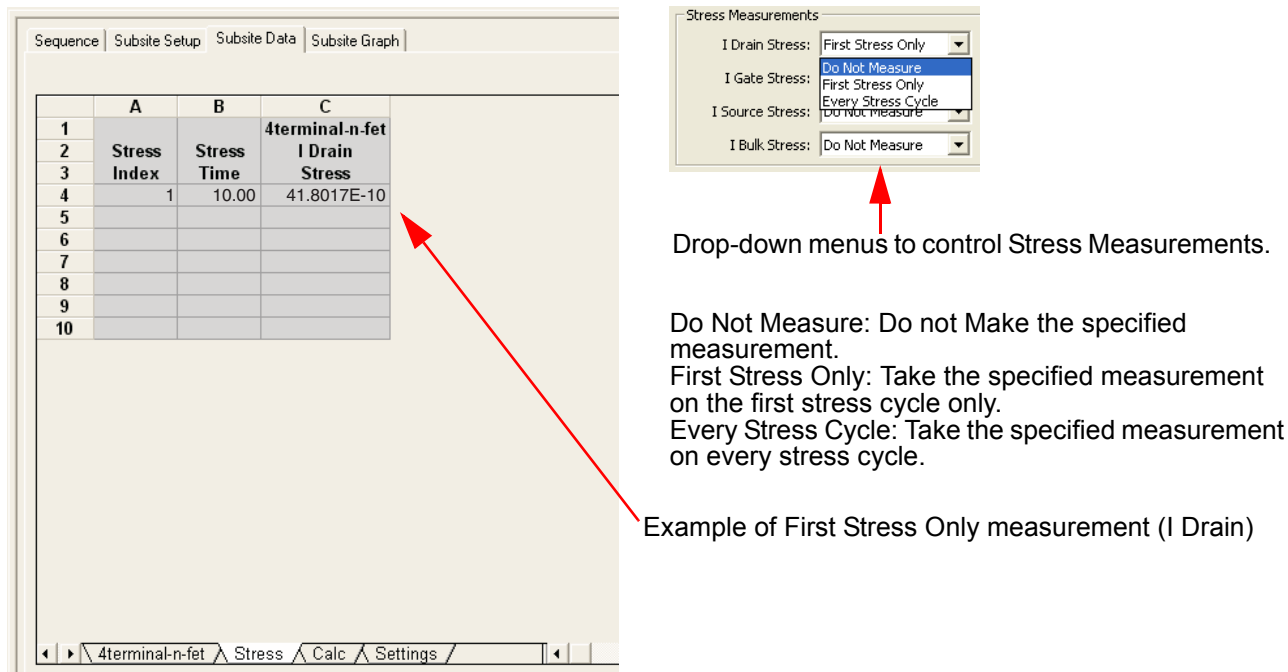
Stress measurements can be performed for each device. When the device is being stressed by DC voltage, the DC currents can be measured. If the device is being stressed by DC current, the DC voltages can be measured. Stress measurements are placed in the Subsite Data sheet for the **Stress** label.

When a device is being stressed by AC voltage (VPU), the current or voltage cannot be measured by the VPU. The VPU does not have measure capability.

Figure 3-66 shows an example of a single First Stress Only measurement for I Drain. If Every Stress Cycle was selected, then there would be a corresponding reading for every stress cycle.



Figure 3-66  
Example of “First Stress Only” measurement



### Device selection

The Device Stress Properties window corresponds to the selected device in the Subsite Plan. The individual properties window for the each device is selected using the **Next Device** or **Prev Device** buttons. If there is only one device in the Subsite Plan, these buttons will be disabled.

### Leave stress conditions on

**DC stressing only:** Enable the **Leave Stress Conditions On** button to leave the outputs of the SMUs on after the end of a stress cycle. This allows the stress to continue until the next test is performed in the project tree. You may want to keep stress on as long as possible so the DUT doesn't have time to relax before the tests are performed.

### Clear, copy, paste, and paste to all sites

**Clear:** Clicking the **Clear** button clears all stress properties data for the displayed device. It sets all voltage and current values to zero, sets device pin number assignments to zero, sets Stress Measurements to Do Not Measure, and disables all Targets (clears Target Values).

**Copy and Paste:** Copy and Paste allow properties settings for one device to be copied and pasted into the properties window for a different device. It can also be used to copy and paste settings into a different site.

### Use Copy and Paste as follows:

1. On the desired Device Stress Properties window, click **Copy** to copy the properties into the buffer.
2. If pasting to a different site, select the site as shown in step 1 in [Figure 3-59](#).
3. Use the **Next Device** or **Prev Device** button to display the properties window for the desired device.
4. Click **Paste** to overwrite the device properties with the properties stored in the buffer.



**Paste to All Sites:** After copying the properties for the desired Device Stress Properties window (as explained in step 1 above), click **Paste to All Sites** to overwrite the device properties for all available sites.

## Segment Stress / Measure Mode

---

**NOTE** *The following supplemental information explains stress testing using the Segment Stress / Measure Mode. This stress / measure mode is similar to the basic Stress / Measure Mode, but instead uses the Segment Arb™ pulse mode of a Keithley pulse card.*

---

Segment Stress / Measure Testing consists of two phases:

- During a measure phase, the SMUs perform DC measurements on the DUT.
- During a stress phase, the Keithley pulse card provide stress using Segment Arb™ waveforms, and the SMUs provide voltage bias and current limit. There are no measurements performed during the stress phase.

---

**NOTE** *Refer to [Segment Arb™ stressing on page 3-82](#) for details about using Segment Arb™ stressing to endurance test floating gate flash memory devices.*

---

[Figure 3-67](#) shows a typical stress / measure test system using a switch matrix to automate the stress and measure phases of the test:

- During a measure phase, the switch matrix connects the SMUs that will perform the DC measurements on the DUT. The Keithley pulse card is disconnected from the DUT during a measure phase.
- During a stress phase, the switch matrix connects the pulse generator to the DUT. It also connects SMUs that will be used for device pin grounding or biasing.

---

**NOTE** *The Model 708A Switching Mainframe and Model 7174A Matrix Card shown in [Figure 3-67](#) are added to the Model 4200-SCS system from KCON. See the [Reference manual, Appendix B, Using KCON to add a switch matrix to the system and configure its connections.](#)*

---



---

**NOTE** *When using segment stress / measure mode with multiple pulse cards installed in a Model 4200-SCS, trigger connections must be made as shown in [Figure 3-34](#). For further information, see [Section 11 Pulse Source-Measure Concepts](#) of the Reference Manual. Specifically, refer to [Trigger connections: Output synchronization](#) and [Multi-channel synchronization with the Segment Arb™ Mode](#).*

---

*the Reference Manual section 11 Pulse Output Configuration [make this a link please], specifically [Trigger connections: Output synchronization](#) [make link] and [Multi-channel synchronization with the Segment Arb™ Mode](#) [make link]*

---

**NOTE** *To effectively transmit the higher frequency components of the typical pulse (Segment Arb™ or Standard), a high bandwidth switch matrix should be used (for example, Keithley Instruments Model 7174A or 7173-50).*

---



Figure 3-67

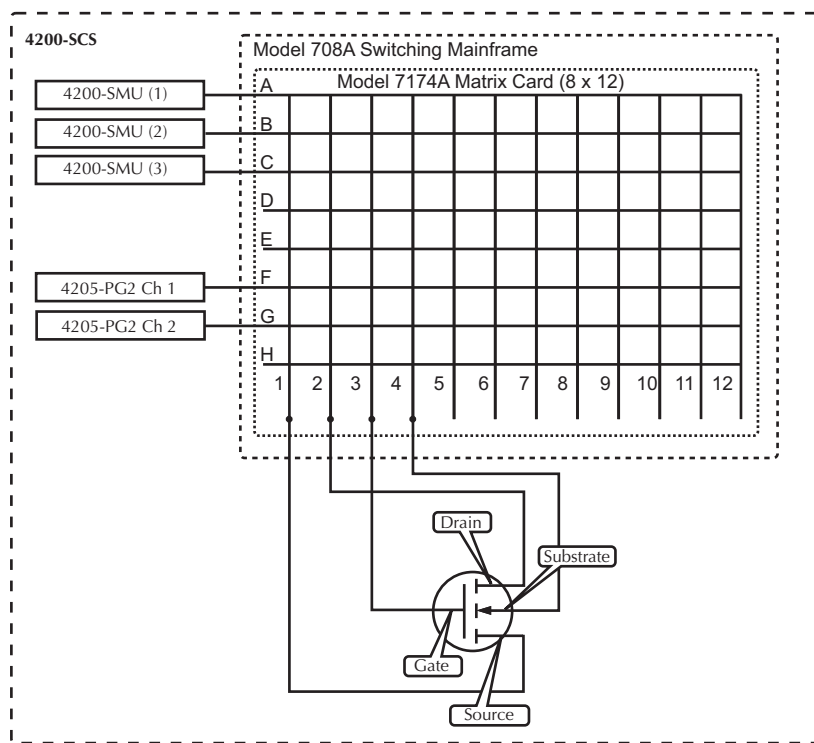
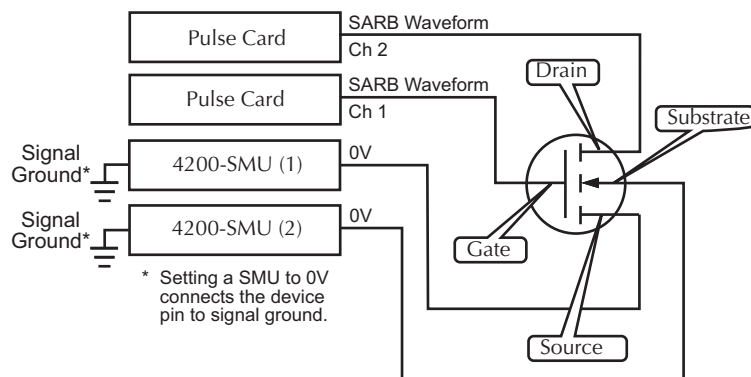
**Stress / measure test system****Segment Arb™ stressing**

Figure 3-68 shows an example of how a DUT can be stressed using Segment Arb™ waveforms. During a stress phase, the matrix shown in Figure 3-67 connects the channels of the Keithley pulse card to the drain and gate of the DUT. The pulse generator stresses the drain and gate by outputting Segment Arb™ waveforms.

Two Model 4200-SMUs (SMU1 and SMU2) are connected to the substrate and source terminals of the DUT, and are set to 0 V to effectively ground the terminals.

Figure 3-68

**Segment stressing: Stress phase example**



## Segment Stress / Measure Mode configuration

The Segment Stress / Measure Mode is configured from the subsite setup tab. After double-clicking the name of the subsite plan in the project navigator, select the subsite setup tab (see [Figure 3-69](#)).

For Segment Arb™ stressing, the waveform period is the fundamental unit of time for stressing. In the subsite setup tab, the term stress counts is used to specify the number of times the Segment Arb™ waveform will stress the device. For example, assume the stress count is three, and the waveform period is four seconds. For that stress cycle, the Segment Arb™ waveform will stress the device three times for a total stress time of twelve seconds.

### Configure stress counts

To configure the stress counts for the Segment / Stress Measure Mode: (see [Figure 3-69](#))

Figure 3-69

#### Segment Stress / Measure Mode: Subsite Setup

1. Select **Enable Cycles**.
2. Select the **Segment Stress / Measure Mode**.
3. Select and configure **Stress / Measure Cycle Times**:

**Linear cycle counts:** After setting the first and total stress counts, and the number of stresses, the linear **Stress Counts** will be automatically calculated and displayed when **Apply** is clicked (Step 5).

**Log cycle counts:** After setting the first and total stress counts, and the number of stresses per decade, the log **Stress Counts** will be automatically calculated and displayed when **Apply** is clicked (step 5) (see [Figure 3-70](#)).

**List cycle counts:** Cycle counts are added to the **Cycle Times** list by entering a count value into the **Stress Counts** field and clicking **Add**. A cycle count value can be removed by selecting it and clicking **Remove** (see [Figure 3-70](#))



If desired, specify a **Stress / Measure Delay**. This allows the device to settle after stressing before performing the DC measurements.

4. Periodic testing is not available for the Segment Stress / Measure Mode.
5. Clicking the **Apply** button updates the settings in the Subsite Setup tab. The button will be inactive if updating is not required.
6. Click **Device Stress Properties** to open the window to configure the Segment Arb™ waveform, SMU bias levels and matrix connections (see [Figure 3-71](#)). Proceed to [Configure Device Stress Properties on page 3-84](#) to continue the configuration process.

Figure 3-70

**Segment stress / measure mode: Log and list cycle counts**

Log cycle counts

List cycle counts

**Configure Device Stress Properties**

To configure the device stress properties for the Segment / Stress Measure Mode:  
(see [Figure 3-71](#))

Figure 3-71

**Segment Stress / Measure Mode: Device Stress Properties**

**General Settings**

NOTE: Set a SMU pin to -1 to indicate high impedance mode used when sharing a terminal with a PG2.

Active Site: 1

SMU1 Bias: 0 V SMU2 Bias: 0 V SMU3 Bias: 0 V SMU4 Bias: 0 V SMU5 Bias: 0 V

SMU1 Limit: 0.105 A SMU2 Limit: 0.105 A SMU3 Limit: 0.105 A SMU4 Limit: 0.105 A SMU5 Limit: 0.105 A

SMU1 Pins: 0 SMU2 Pins: 0 SMU3 Pins: 0 SMU4 Pins: 0 SMU5 Pins: 0

**Pulser Settings**

PG2-1 PG2-2 PG2-3

Channel 1 PG2-1 Channel 1 Pin: 0

Channel 2 PG2-1 Channel 2 Pin: 0

Channel 1 File: C:\S4200\user\K\Pulse\SarbFiles\Fish-NAND-Vg.ksf

Channel 2 File: C:\S4200\user\K\Pulse\SarbFiles\Fish-NAND-Vd.ksf

**Parameter Properties/Degradation Targets**

Tests	Output Values	% Abs	Target	Target Value
Vt-MaxGm#1	VT	<input checked="" type="radio"/> <input type="radio"/>	<input type="checkbox"/>	20.0

Device Name: FloatingGate

<< Prev Device Next Device >>

Clear Copy Paste Paste to All Sites OK Cancel

**1. Active Site selection:**

- When active, this field is used to select the wafer site number.



- If there is only one wafer site, this field will be inactive.
2. **PG2 selection:**
    - There is a Device Settings Properties window for each pulse generator card in the system.
    - Select the PG2 to be configured.
  3. **SMU Pins:**
    - **No switch matrix:**
      - With no switch matrix, the active SMU pin fields must be set to **0** (no connection) or **-1** (high impedance).
      - The **-1** setting puts the SMU in a high impedance mode, which is necessary if it shares a pin with a VPU.
    - **Switch matrix:**
      - With a switch matrix added to the system, the pin number settings determine signal routing for the SMUs through the matrix to the device pins.
  4. **PG2 Matrix connections:**
    - With a switch matrix added to the system, fields for PG2 Channel pins are active.
      - There is a PG2 pin connection setting each channel.
    - The pin number settings determine signal routing for the PG2 Channels through the matrix to the device pins.

---

**NOTE** A setting of **0** indicates no connection to the DUT (PG2-1 Channel not used).

---

5. **SMU settings:**
  - A DC bias voltage and current limit is set for each SMU being used in the stress test:
    - A bias setting of **0** V effectively grounds the terminal.
    - The fields for SMUs not installed in the system are inactive.
    - Only five SMU's are supported in Segment Stress / Measure Mode, SMU1 through SMU5.
6. **Segment Arb™ waveform file:**
  - Use to import the **.ksf** Segment Arb™ waveform file for each pulse generator channel.
  - An imported waveform will be shown in the previewer.
  - The **.ksf** waveform file can be created and saved (exported) in KPulse (see [How to Generate Basic Pulses, Section 5](#)).
7. **Degradation Targets:**
  - Lists the tests and Output Values for this device.
  - Targets can be enabled and the target values can be set (in % or Absolute Value).
  - The test sequence for the Segment Stress / Measure Mode, is the same as the test sequence for the basic [Stress/measure mode](#) described later in this section.

---

**NOTE** Output Values are imported into this target list from the ITM/UTMs in the device plan (see the [Reference manual, ITM Output Values, page 6-144](#), and [UTM Output Values, page 6-151](#)).

---

8. **Next Device and Prev Device buttons:**
  - For a multi-device Subsite Plan, click **Next Device** to display the stress properties window for the next device, and repeat steps 1 through 7.
  - Clicking **Prev Device** selects the previous device.
    - If there is only one device in the plan, these buttons will be inactive.
9. **Editing buttons:**



- Select the **Clear**, **Copy**, **Paste**, and **Paste to All Sites** buttons to perform editing operations for managing entries (see [Clear, copy, paste, and paste to all sites on page 3-80](#)).
10. **OK button:**
    - Click when finished setting the stress properties for all devices and all sites.
  11. In the Subsite Plan tab (see [Figure 3-69](#)), click the **Apply** button to apply and save the settings made in the Device Stress Properties window.

---

**CAUTION** After setting the device stress properties for all devices and sites, steps 11 of the above procedure must be performed in order to apply and save the new settings. Failure to do so will cause the new settings to be lost.

---

## Executing subsite cycling

With the Subsite Plan in the project navigator selected and enabled, subsite cycling is started by clicking the **Run Test/Plan** and **Cycle Subsites** button (see [Figure 3-72](#)):

- If the Cycle Mode is selected for subsite cycling, all cycles of the Subsite Plan will run.
- If using the Stress / Measure Mode is being used and there are enabled Targets, the Subsite Plan will terminate when all enabled Targets are reached.
- Otherwise, all cycles of the Subsite Plan will run.

Figure 3-72

**Starting subsite cycling**



Click **Run Test/Plan** and **Cycle Subsites** button to start subsite cycling.

### Multiple subsite cycling

If two or more subsites are configured for subsite cycling, they can all be run consecutively. When the first subsite is finished cycling, the next subsite will start automatically.

Multiple subsite cycling is started from the project level, rather than the subsite level. In the project navigator, select and enable the Project and then start cycling as shown in [Figure 3-72](#).

## Subsite cycling data sheets

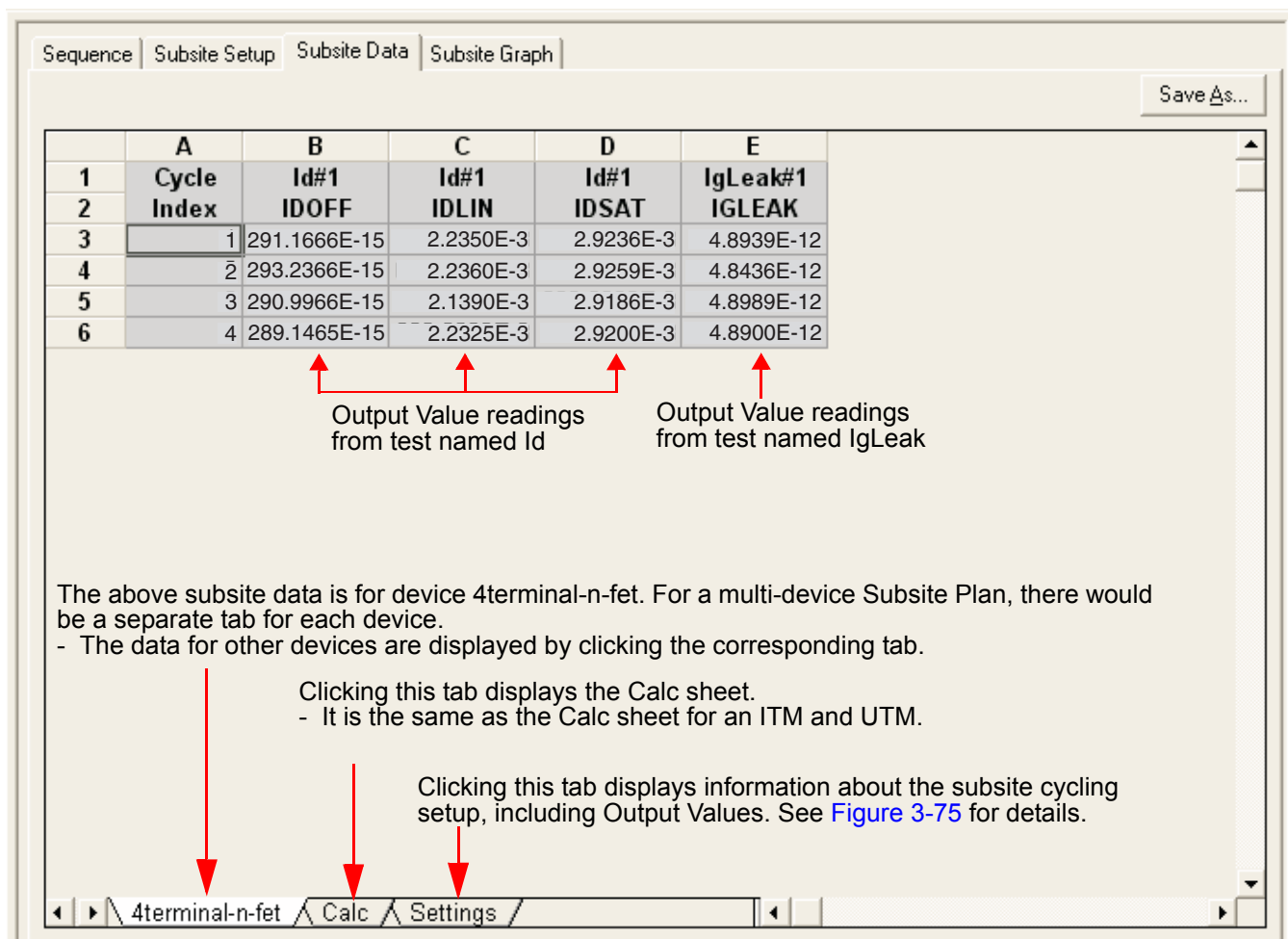
Spreadsheet Data for the Subsite Plan is acquired in the Subsite Data sheet. With the Subsite Plan opened in the workspace, the data sheet is displayed by clicking the Subsite Data tab.

### Cycle Mode data sheet

[Figure 3-73](#) shows an example data sheet for a Subsite Plan that has one device. Column A lists the cycles that were run. For the example in [Figure 3-73](#), four cycles were run. Columns B, C, D and E lists the readings for the four Output Values.



Figure 3-73  
Subsite Data sheet: Cycle Mode



### Stress / Measure Mode data sheet

Figure 3-74 shows an example data sheet for a Subsite Plan that has one device. Spreadsheet columns are explained as follows:

- Column A      Lists the cycles that were run. For the example in Figure 3-74, eight cycles were run.
- Column B      Lists the stress times (in seconds) for all cycles.

---

**NOTE**    *The stress for the first cycle is 0.0 seconds.*

---

This is the no-stress cycle for HCI testing.

- Column C      Output Values: Lists the measured readings for the first Output Value (IDOFF reading for the ID#1 test).
- Column D      Starting with Cycle 2, lists the % Change between each post-stress IDOFF reading and the pre-stress IDOFF reading (Cycle 1). The % Change for an Output Value is calculated as follows:

$$\% \text{ Change} = \text{ABS}[(\text{Post-Stress Rdg} - \text{Pre-Stress Rdg}) / \text{Pre-Stress Rdg} \times 100]$$



For the example in [Figure 3-74](#), the following shows how % Change IDOFF for Cycle 2 is calculated:

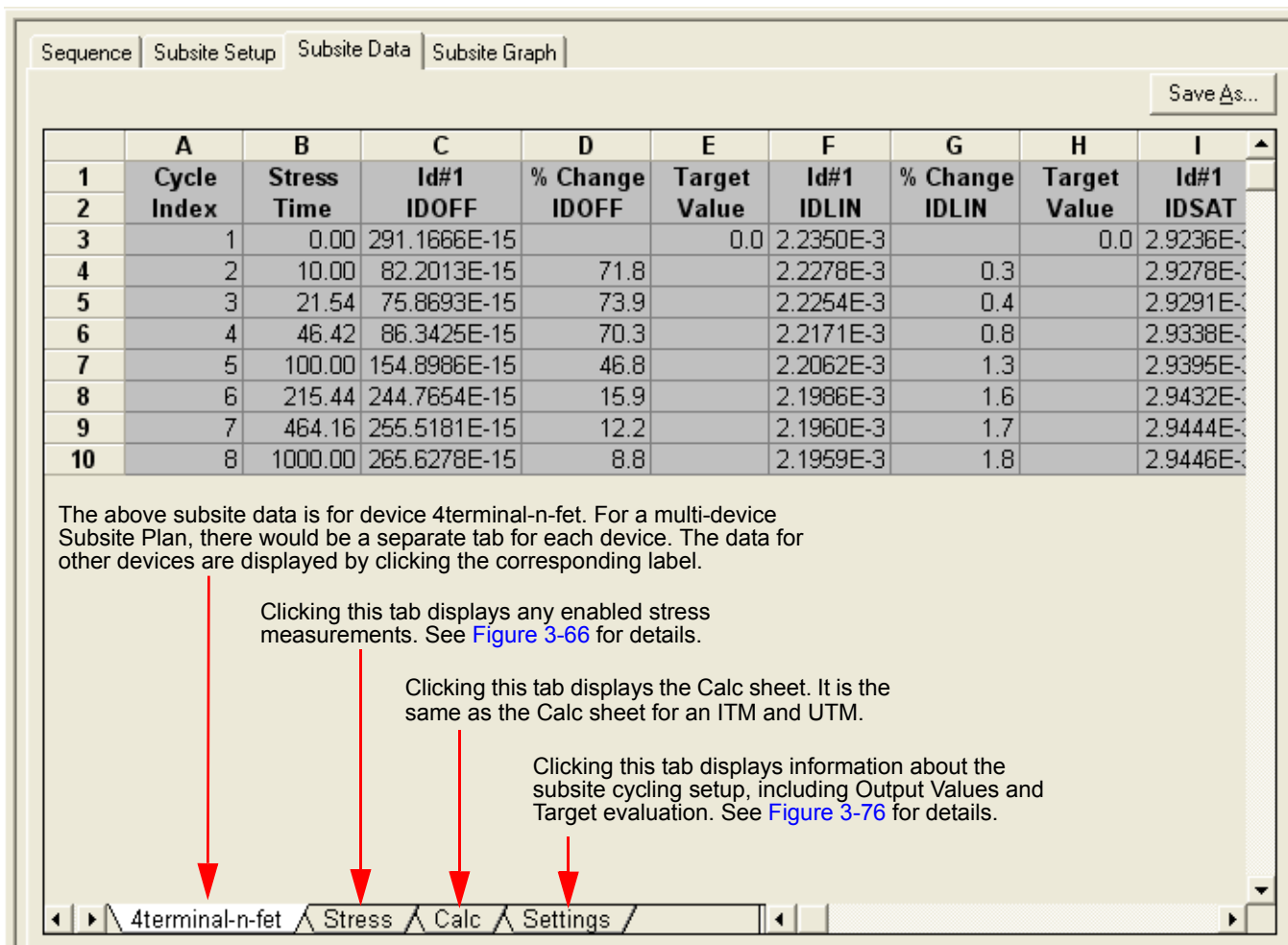
$$\begin{aligned}
 \% \text{ Change IDOFF} &= \text{ABS}[(82.2013\text{E-}15 - 291.1666\text{E-}15) / 291.1666\text{E-}15 \times 100] \\
 &= \text{ABS}[-208.9653\text{E-}15 / 291.1666\text{E-}15 \times 100] \\
 &= \text{ABS}[-0.7176 \times 100] \\
 &= 71.8
 \end{aligned}$$

Column E This is the Target Value that was assigned to the Output Value in the Device Stress Properties window (see step 4 in [Figure 3-59](#)). A Target Value of 0.0 indicates that the Target for IDDOF is disabled. A Target is reached when the % Change value equals or exceeds the Target Value.

Starting with Column F, every three columns provide readings for another Output Value, the % Change and the Target Value.

Figure 3-74

#### Subsite Data sheet: Stress / Measure Mode



#### Settings window

The Settings window displays information about the subsite cycling setup. The Settings window is displayed by clicking the **Settings** tab at the bottom of Subsite Setup tab (see [Figures 3-73 and 3-74](#)).



The Settings window for the Cycle Mode is shown in [Figure 3-75](#). It provides basic information on the subsite cycling setup and lists the Output Values for each device and test. The Settings window for the Stress / Measure Mode is shown in [Figure 3-76](#). It is similar to the Settings window for the Cycle Mode and includes information on Targets. For each enabled Target, the Target Value is listed. After subsite cycling, it also indicates if Targets have been reached.

Figure 3-75  
Subsite Data: Settings window for Cycle Mode

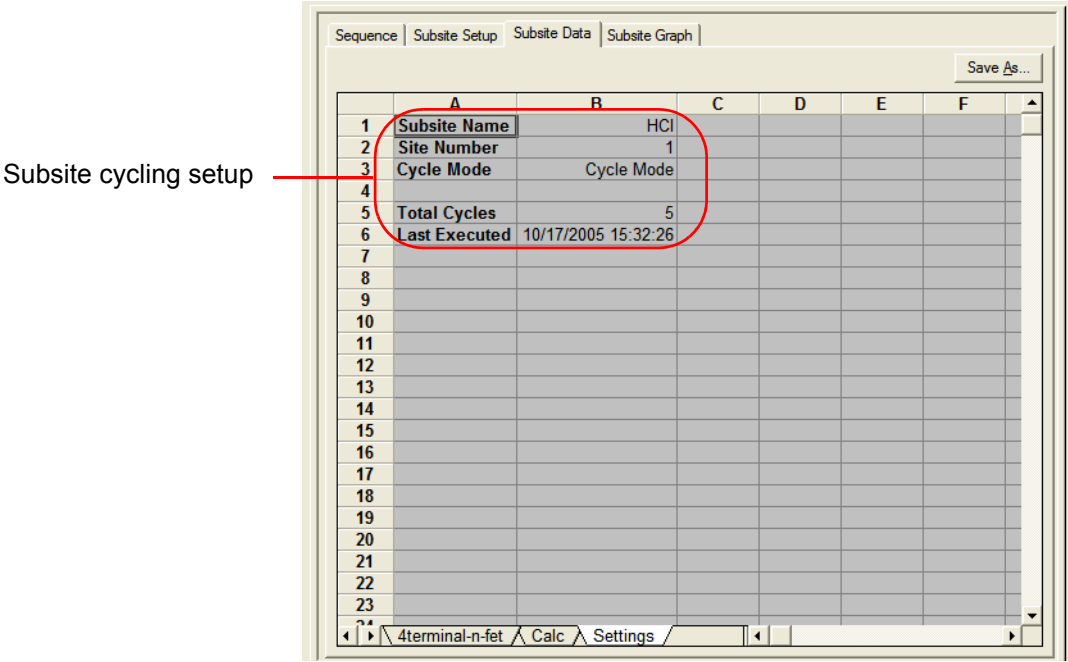




Figure 3-76

**Subsite Data: Settings window for Stress / Measure Mode**

Sequence | Subsite Setup | **Subsite Data** | Subsite Graph

Save As...

	A	B	C	D	E	F
1	Subsite Name	HCI				
2	Site Number	1				
3	Cycle Mode	Log Stress Mode				
4	First Stress Time	10.0				
5	Total Stress Time	10009.0				
6	# Stresses/Decade	1.0				
7	Stress/Measure Delay	0.0				
8	Stress Times	10.0	100.0	1000.0	10000.0	
9	Last Executed	10/17/2005 15:27:39				
10						
11	Device 1	4terminal-n-fet				
12	Test	Id#1	Id#1	Id#1	IgLeak#1	VtextLin#
13	Output Value	IDOFF	IDLIN	IDSAT	IGLEAK	VTEXTLIN
14	Enable Target	No	No	No	No	N
15	Target % Value	0.0	0.0	0.0	0.0	0.0
16	Target % Reached	No	No	No	No	N
17						
18						
19	Device Status:					
20	Device	4terminal-n-fet				
21	Status	OK				
22						
23	Stress Properties:					
24	Stress Type	DC Voltage Stress				
25	Number of Devices	1				
26						
27	Device Pin/SMU Connections:					
28	Device 1	Drain Pin (SMU)	Gate Pin (SMU)	Source Pin (SMU)	Bulk Pin (SMU)	
29	4terminal-n-fet	2	3	1	4	
30						
31						
32	Stress Measurements:					
33	Device 1	I Drain Stress	I Gate Stress	I Source Stress	I Bulk Stress	
34	4terminal-n-fet	Do Not Measure	Do Not Measure	Do Not Measure	Do Not Measure	
35						
36						
37	Stress Conditions:					
38	Device 1	V Drain Stress	I Drain Limit	V Gate Stress	I Gate Limit	V Source Stress
39	4terminal-n-fet	6.0000E+0	100.0000E-3	2.6000E+0	100.0000E-3	000.0000E-
40						

4terminal-n-fet | Stress | Calc | Settings

Subsite cycling  
setup

Output Values and  
Target information:  
Lists Output  
Values  
Identifies enabled  
Targets  
Lists the Target %  
Values  
Indicates if a  
Target was  
reached

Additional  
information  
provided for the  
stress / measure  
mode

## Subsite cycling graphs

Graphs for subsite cycling are located in the Subsite Graph tab of the Subsite Plan.

### Cycle mode

The graphs for the Cycle Mode plot Output Values versus the cycle index. Each data point in the graph represents an Output Value reading for each subsite cycle. [Figure 3-77](#) explains how to display the various graphs.

[Figure 3-77](#) shows the graph traces for test ID#1 for the NMOS-1 device. The three traces are for Output Values IDOFF, IDLIN and IDSAT.

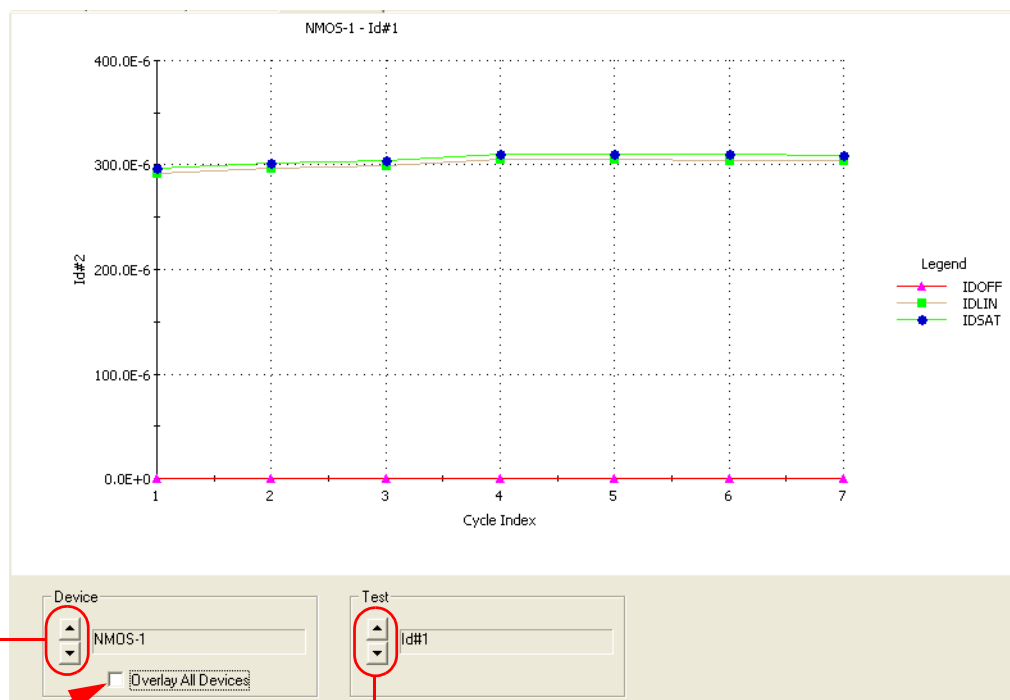


Figure 3-77  
Subsite graph tab: cycle mode

**NOTE**

For a single-device subsite plan, the Device select buttons and the checkbox to Overlay All Devices are disabled.

For a single-test subsite plan, the Test select buttons are disabled.



1) Use to select device.

Click (enter  $\sqrt{\phantom{x}}$ ) to display all the graph traces for all devices that were measured by the selected Test.

2) Use to select test.

### Stress/measure mode

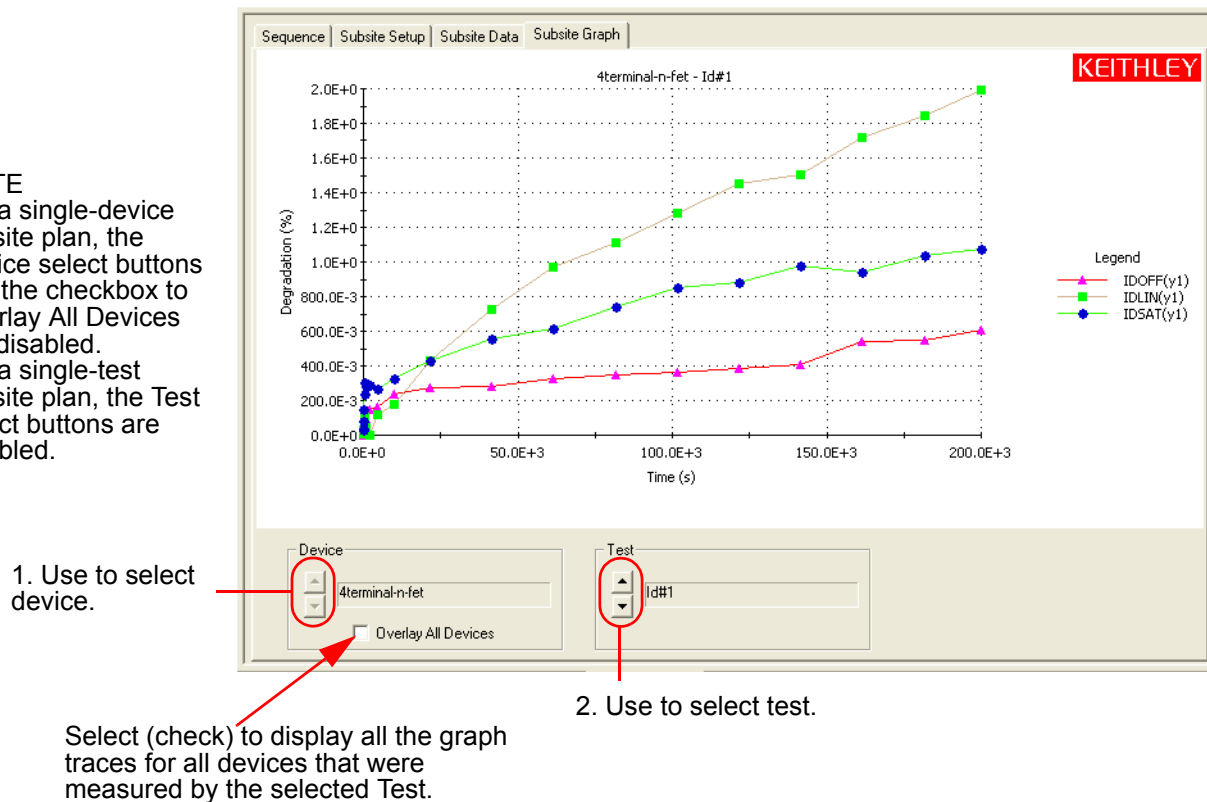
The graphs for the stress / measure mode plot degradation (in %) versus the stress times. Each data point in the graph represents the device degradation (% Change) for tests after each stress cycle (stress time). [Figure 3-78](#) explains how to show the graphs for a selected device test.

[Figure 3-78](#) shows the graph traces for test ID#1 for the 4terminal-n-fet device. The three traces are for Output Values IDOFF, IDLIN and IDSAT.



Figure 3-78  
Subsite Graph tab: Stress / Measure Mode

**NOTE**  
For a single-device subsite plan, the Device select buttons and the checkbox to Overlay All Devices are disabled. For a single-test subsite plan, the Test select buttons are disabled.



## Configuration sequence for subsite cycling

There are four project plans that use subsite cycling. These include HCI\_1\_DUT, HCI\_4\_DUT, NBTI\_1\_DUT, and EM\_const\_I. The process flow for these projects is shown in [Figure 3-79](#).

**NOTE** A new project plan for subsite cycling can be created or one of the four existing project plans can be modified as needed. For details, see the [Reference Manual, Building, modifying, and deleting a Project Plan, page 6-47](#).

When adding a device plan or test to a subsite cycling project, the following sequence must be followed:

1. Insert a device plan for the type of device to be tested. For example, if testing a 4-terminal, n-channel MOSFET, insert the 4terminal-n-fet device into the subsite plan.
2. Under the device plan, insert a new test (ITM or UTM) or copy a test from the test library and make the proper modifications.
3. Use the **Formulator** for the ITM or UTM to configure data calculations on test data.
  - The window to set the formulator is opened by clicking the **Formulator** button on the definition tab of the ITM or UTM.
    - For more information about how to use the **Formulator** refer to the [Reference Manual, Analyzing test data using the Formulator, page 6-288](#).
4. Select the **Output Values** to be exported to the subsite data sheet for inter-stressing monitoring.

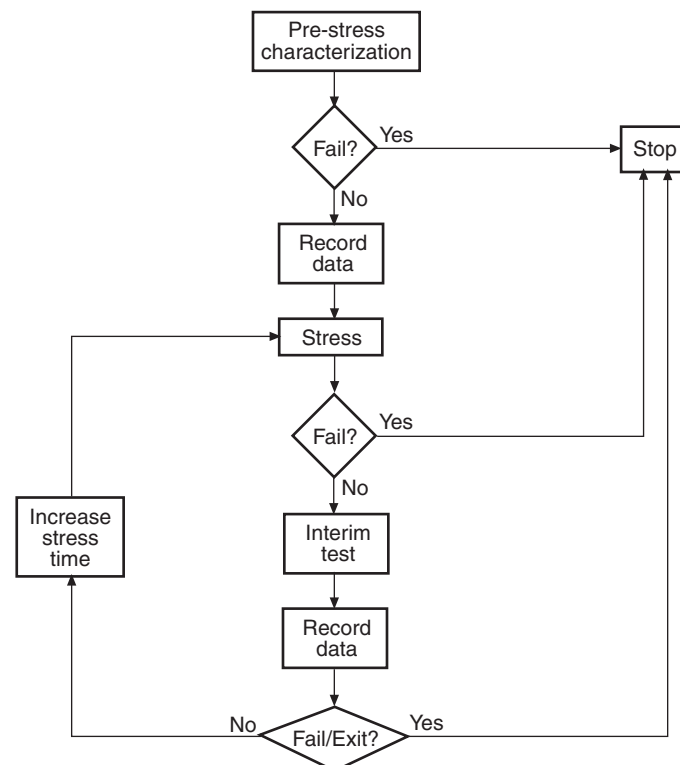


- The window to select output values is opened by clicking the **Output Values** button in the **Definition** tab for the ITM or UTM.
  - An Output Value is selected by clicking a checkbox to insert a ✓
    - For details, see the [Reference manual, ITM Output Values, page 6-144](#), and [UTM Output Values, page 6-151](#).
5. If desired, **Exit Conditions** for an ITM can be set. When an exit condition other than **None** is selected and the source for the ITM goes into compliance, the test, device plan, subsite plan, site, or project will terminate. **None** is the default exit condition.
    - The window to set the exit condition is opened by clicking the **Exit Conditions** button in the **Definition** tab of the ITM.
      - For details, see the [Reference Manual, ITM compliance exit conditions, page 6-144](#).
  6. Save the project plan by selecting **Save All** from the **File** menu (at the top of the KITE window). You can also save the project by clicking the **Save All** button on the toolbar.
  7. Repeat steps 2 through 6 for adding more tests for the same device.
  8. Repeat steps 1 through 7 for adding more devices to the subsite plan.
  9. Configure the subsite for subsite cycling (stress / measure mode: In the project navigator, double-click the subsite plan and select the Subsite Setup tab to configure subsite cycling.
    - a. Set the stress / measure mode cycle times for the subsite plan: The stress / measure mode and cycle times are set from the Subsite Setup tab.
    - b. Configure the stress properties and connection information for every device in the subsite plan: In the Subsite Setup tab, click the **Device Stress Properties** tab to open the properties window.

For detailed information about subsite cycling, see [Subsite cycling](#) in Section 6 of the reference manual.

Figure 3-79

**Process flow: HCI/NBTI/constant current EM**





# How to perform a flash memory test on my device

## Introduction

There are several projects included with the Model 4200-SCS FLASH package that facilitate testing of floating gate transistors (NOR, NAND), as well as other types of Non-volatile Memory (NVM). The package consists of two pulse cards (four pulse channels), projects described in this section, and all required interconnecting cables and adapters (see [Figure 3-89](#)).

Depending on the desired setup, at least two SMUs are required. To illustrate the flexibility of the Model 4200-SCS FLASH package [on page 3-94 Figure 3-84](#) and [Figure 3-85](#) depict a typical configuration using four SMUs.

This configuration permits independent source and measure for each terminal in a typical 4-terminal floating gate transistor.

---

**NOTE** The Model 4200-SCS FLASH package does not include a Model 4200-SCP2 (2-channel scope card). When using the Models 4205-PG2 or 4220-PGU, the scope card can be added for manual pulse height verification. Since the Model 4225-PMU has measure capability, a separate scope card is not necessary.

---



---

**NOTE** The [PMU-Flash-NAND project](#) (in [Section 16](#) of the Reference Manual) uses the Model 4225-PMU to test flash memory.

---

## Theory of operation

### Programming and erasing flash memory

A floating gate transistor is a modified field-effect transistor with an additional floating gate. The floating gate transistor is the basic storage structure for data in non-volatile memory. The floating gate (FG) stores charge, that represents data in memory (see [Figure 3-80](#)).

The control gate (CG) reads, programs, and erases the FG transistor. The presence of charge on the gate shifts the voltage threshold,  $V_T$ , to a higher voltage, as shown in [Figure 3-81](#).

Figure 3-80

**Cross section of a floating gate transistor in both the erased and programmed states**

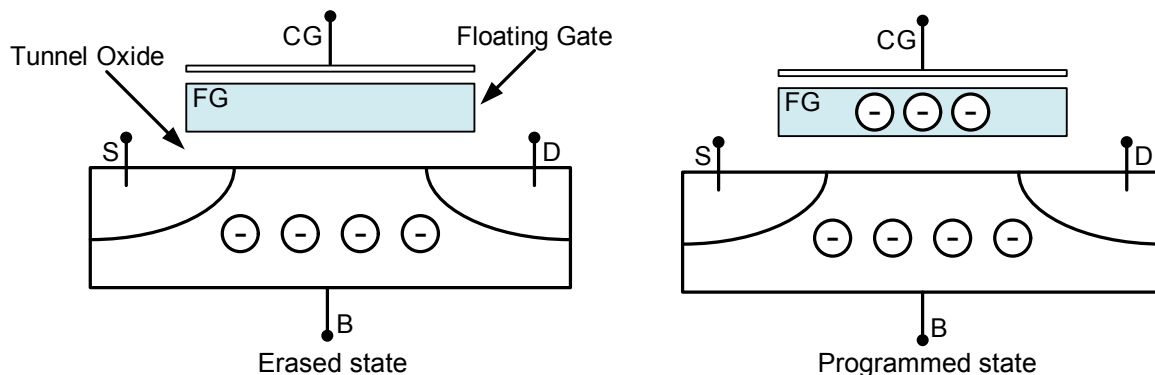
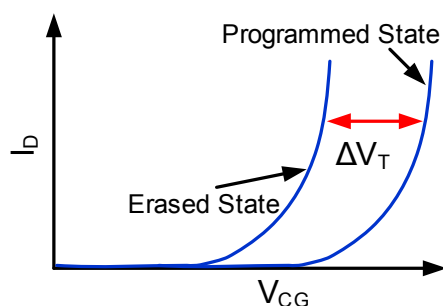




Figure 3-81

**Graph of shifted voltage threshold,  $V_T$ , due to stored charge on floating gate on a 1 bit (2 level) cell**



**The Flash transistors tests consist of two parts:**

1. Pulse waveforms to program or erase the DUT
2. DC measurements are taken to determine the state of the device

**This implies switching between two conditions:**

1. Pulse resources are connected to the DUT
2. Pulse resources are disconnected and the DC resources are connected to the DUT

**The pulses are used to move charge to or from the floating gate. There are two different methods to move charge:**

1. Tunneling
2. Hot electron injection (HEI)

The tunneling method is commonly known as Fowler-Nordheim (FN) tunneling, or quantum tunneling, and is a function of the electric potential across the tunneling oxide (see [Figure 3-82](#)). HEI is considered a damage mechanism in non-floating gate transistors, and is commonly called hot charge injection (HCI). HEI/HCI is a method that accelerates charges by applying a drain-source field, and then the charges are directed into the floating gate by a gate voltage.

[Figure 3-82](#) shows examples of tunneling to move charge to and from the FG.

- The electric field and the preferred direction of electron flow are indicated by the black arrows.
- The signal applied to each device terminal are indicated by the blue text and blue features.

---

**NOTE** Both the drain and source are not connected to any test instrumentation.

---

This condition may also be called floating or high impedance. [Figure 3-83](#) shows examples of moving charge using HEI. These conditions are only examples with approximate voltage values, and both pulse width and pulse height will vary depending on device structure and process details.

There are many other ways to provide similar electric fields and balance performance across a variety of parameters: program or erase speed, retention longevity, adjacent cell disturbance, endurance, and others.



Figure 3-82  
Fowler-Nordheim tunneling program and erase.

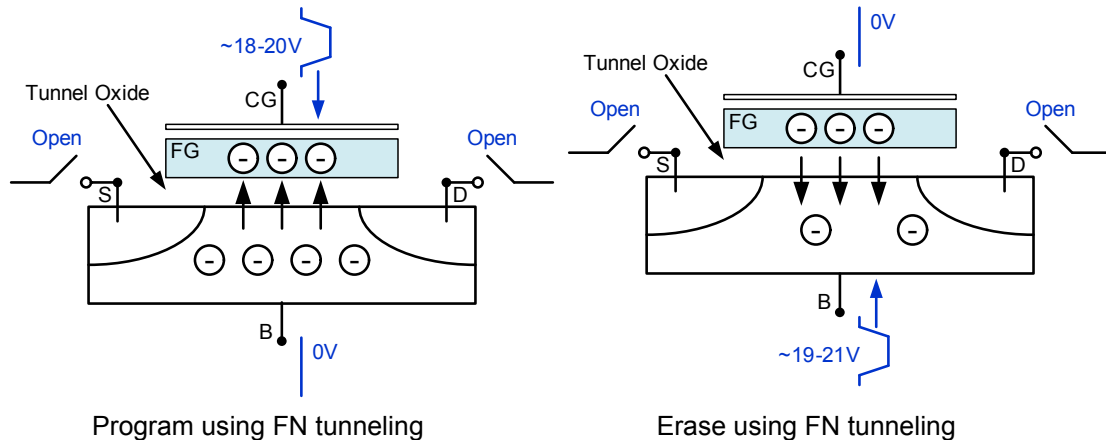
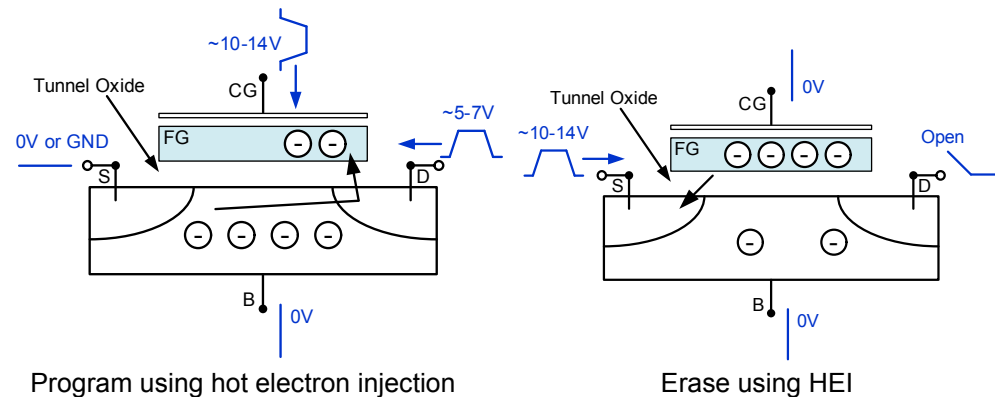


Figure 3-83  
Hot Electron Injection (HEI) program and erase.



The flash projects support two methods for performing the switching between the pulse and measure phases of the typical flash memory test.

The first is the typical method, using a switch matrix to route the pulse or DC signals to the DUT (see [Figure 3-84](#)). Using the switch matrix is more complicated, but provides flexibility for certain tests and test structures that use arrays. The second method utilizes the on-card isolation relays on both the SMUs and the Keithley pulse card to configure a simpler setup without the external switch matrix (see [Figure 3-85](#)).

Because both the SMUs and the Keithley pulse card have isolation relays located on the cards, it is possible to configure a simpler setup without the external switch matrix (see [Figure 3-85](#) and [Figure 3-89](#)). The advantage of the simpler setup is lower cost, while the switch matrix approach provides lower current measurement performance and flexibility necessary for testing arrays of test structures.

**NOTE** For the Model 4225-PMU, the [PMU-Flash-NAND project](#) uses the output relays of the instrument cards to switch the SMU and PMU outputs to the device terminals (see [Figure 16-92](#)).



To determine the state of the device, one would perform a  $V_g$ - $I_d$  sweep, then perform a calculation to find the voltage threshold,  $V_T$ . The shift in  $V_T$  represents a change in the amount of charge stored in the floating gate, that indicates the state of the cell, from fully programmed (1) to fully erased (0). The Model 4200-SCS FLASH package does not include the ability to measure the pulse waveform or pulse response.

Figure 3-84

**Block diagram of an example flash test setup using a switch matrix**

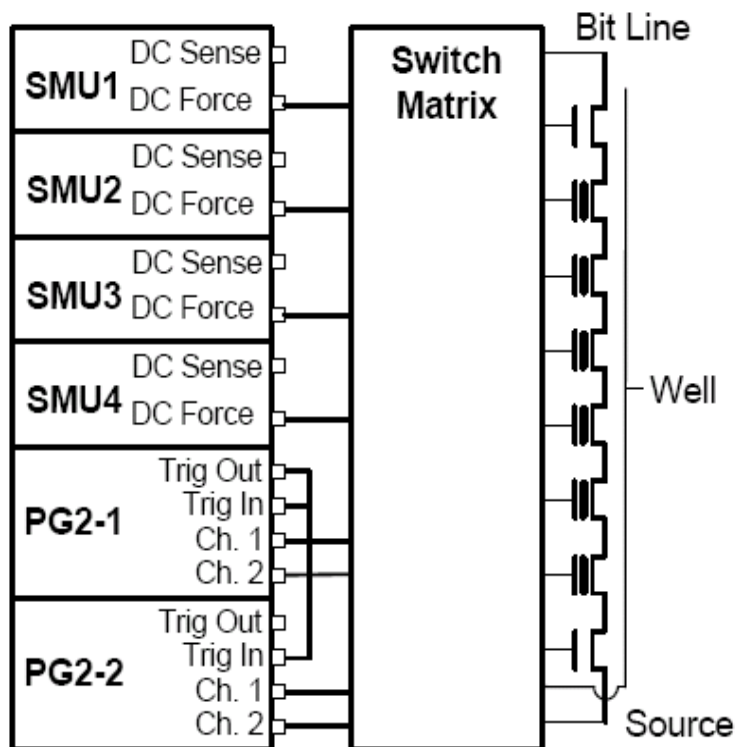
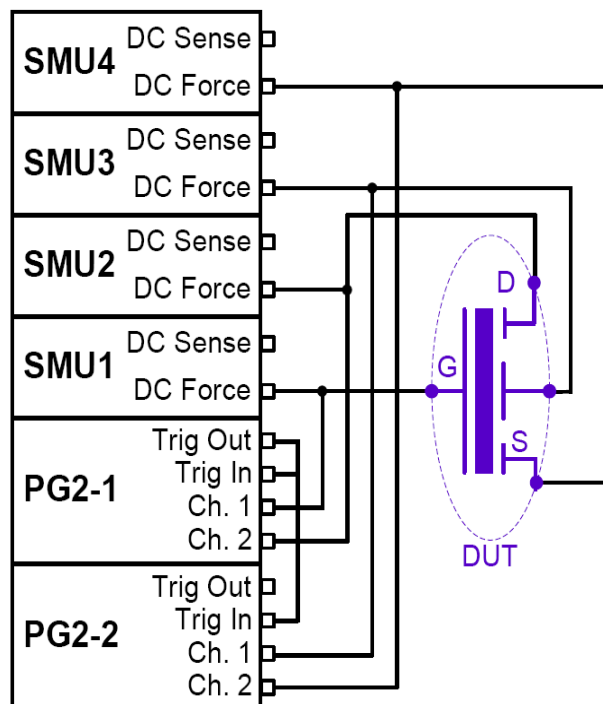




Figure 3-85

**Block diagram of a flash test setup without using a switch matrix (direct connect)**



The pulse waveforms are a program pulse (see [Figure 3-86](#)), an erase pulse (see [Figure 3-87](#)), or a waveform made up of both program and erase pulses (see [Figure 3-88](#)). All of these waveforms are implemented by using the Segment Arb™ capability. For more information about waveforms refer to the [Reference Manual, Pulse Source-Measure Concepts, page 11-1](#). There are many different methods and voltage levels for programming and erasing, so these are only examples.

Figure 3-86

**Program pulse waveforms for a floating gate DUT, with separate pulse waveforms for the DUT gate, drain, source, and bulk**

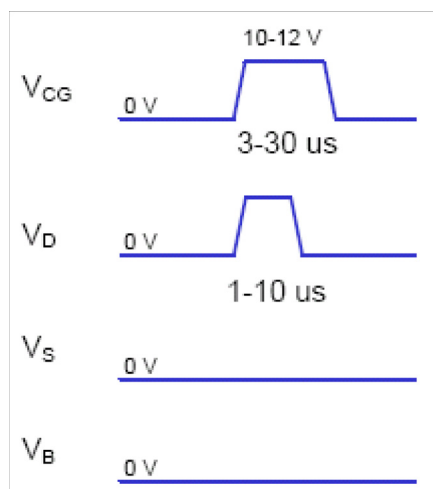




Figure 3-87

**Example erase pulse waveforms for a floating gate DUT, with separate pulse waveforms for the DUT gate, drain, source, and bulk**

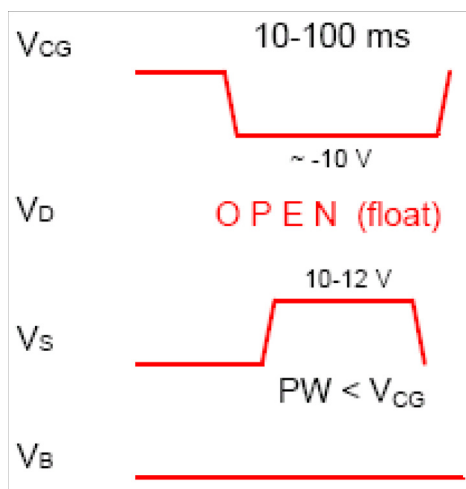
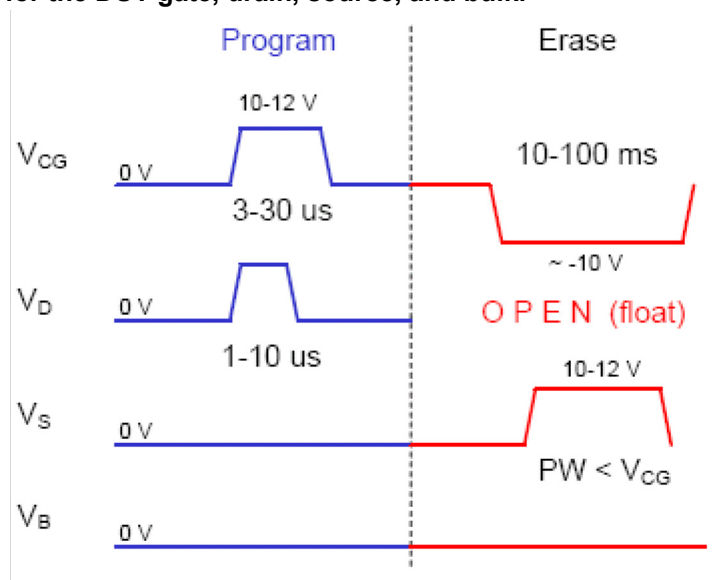


Figure 3-88

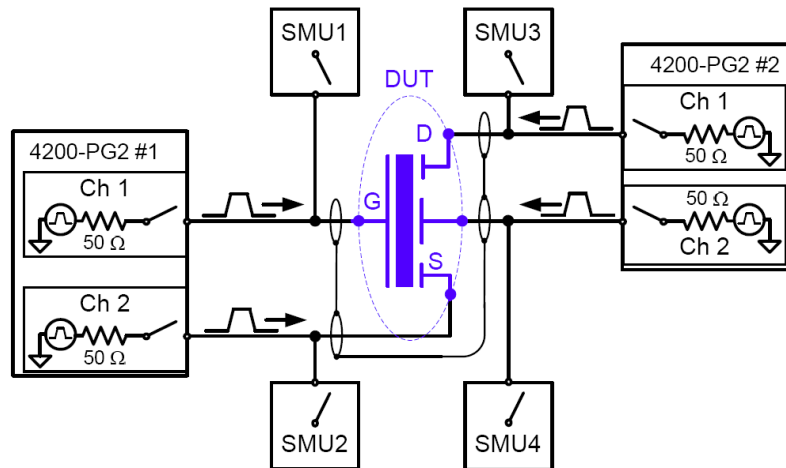
**Program + Erase pulse waveforms for a floating gate DUT, with separate pulse waveforms for the DUT gate, drain, source, and bulk.**



The block diagram for the Flash setup is shown in [Figure 3-89](#). Reconfiguring from the pulse stress to DC measure phases is done by activating the switches on the SMU and PG2 cards. During the pulse program / erase phase, the relays in the PG2 channels are closed and the relays in the SMUs are open. For the DC measure phase, the opposite is true.

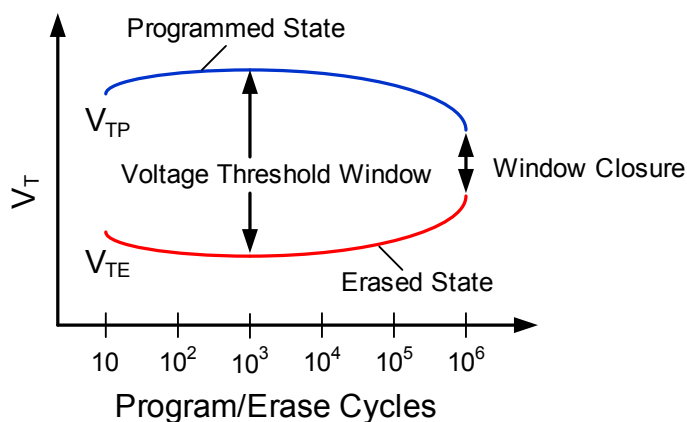


Figure 3-89

**Basic schematic of flash testing without a switch matrix****Endurance testing**

Endurance testing stresses the DUT with a number of Program+Erase waveform cycles, and then periodically measures both the voltage threshold in the programmed state  $V_{TP}$ , as well as the voltage threshold of the erased state,  $V_{TE}$ . The purpose of these tests is to determine the lifetime of the DUT, based on the number of Program+Erase cycles withstood by the device before a certain amount of shift, or degradation, in either the  $V_{TP}$  or  $V_{TE}$ , as shown in Figure 3-90. The endurance test is performed a set number of program and erase cycles (see Figure 3-88), while periodically measuring  $V_T$  for both the programmed and erased state. Figure 3-90 shows typical degradation on a NOR cell for both  $V_{TP}$  and  $V_{TE}$  as the number of applied program/erase cycles increases.

Figure 3-90

**Example results of  $V_T$  shift in an Endurance test on a NOR flash cell.****Disturb testing**

The purpose of the Disturb test is to pulse stress a device in an array test structure, then perform a measurement, such as  $V_T$ , on a device adjacent to the pulsed device.

The goal is to measure the amount of  $V_T$  shift in adjacent cells, either in the programmed or erased states, when a nearby device is pulsed with either a Program, Erase, or Program+Erase waveform.



The typical measurement is a  $V_T$  extraction based on a  $V_g$ - $I_d$  sweep, but any type of DC test may be configured. This test is similar to the endurance test, but the pulsing and measuring are performed on adjacent devices.

[Figure 3-91](#) shows an example configuration to pulse stress a device (Cell 2) and then test an adjacent device (Cell 1) in an array cell memory structure.

The solid-line blue circle indicates the cell to be pulse stressed, and the dotted-line red circles are the adjacent memory cells that will be disturbed by the stressing.

The stress / measure process is explained as follows.

**Initial test conditions** – SMU4 outputs a DC voltage to turn on the control devices for the array. This connects instrumentation at the top of array to the flash memory cells. SMU2 and SMU3 are set to output 0 V. This ensures that only the Cell 2 will be turned on during pulse stressing.

**Pulse stressing** – The output relay for SMU1 is opened, and the gate and drain of Cell 2 are pulse stressed by PG2 #1 (ch 1) and PG2 #2 (ch 1).

**Disturbed cell testing** – The outputs for the PG2s are turned off and their output relays are opened. SMU1 and SMU2 are then used to perform a DC  $V_g$ - $V_d$  sweep on Cell 1 to determine  $V_T$ .

### Using a switch matrix

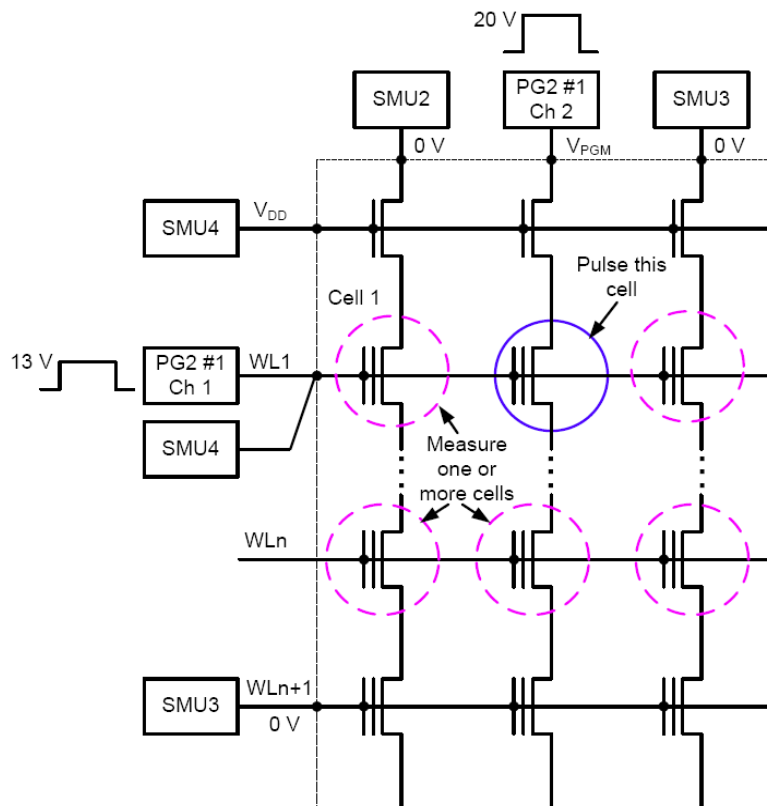
A limitation of the no-switch, direct connect test configuration shown in [Figure 3-91](#) is that only three devices can be measured. The test would have to be manually reconfigured or re-cabled to test other devices.

Without a switch matrix, the number of adjacent cells that can be measured is limited. Therefore, it is recommended that a switch matrix be used for disturb testing, as shown in [Figure 3-97](#).

Using a switch matrix allows the flexibility of routing pulse and DC signals without having to make connection changes. Also, this type of structure uses a multi-pin probe card, that provides an additional opportunity for mapping test resources to DUT pins. For example, a SMU can be shared across multiple device terminals where the required voltage is the same.



Figure 3-91  
Disturb testing – configuration to test a single device



### Pulse waveforms for NVM testing

A pulse card has several attributes that support NVM testing. To perform the multi-level pulse waveforms for the typical program / erase waveform (see [Figure 3-88](#)), each pulse card channel has the Segment Arb™ capability.

For more information about Segment Arb™ waveforms refer to the [Segment Arb™ waveforms](#). The ability to disconnect, or float, a particular device pin, within the Segment Arb™ waveform requires an inline solid state relay. This solid state relay is called the high endurance output relay (HEOR).

The pulse card output channels each have 50  $\Omega$  output impedance. When current flows through the pulse channel, there is a voltage drop across this 50  $\Omega$  resistor internal to the pulse card. This dictates that the voltage at the output may be different from what is expected based on the resistance of the DUT. This effect is called the Load Line Effect and is covered in more detail in [Reference Manual, DUT resistance determines pulse voltage across DUT, page 11-11](#).

- The gate of a flash or NVM device is high impedance.
- The voltage at the gate will be double of the programmed voltage.
- The voltage at the drain will be a function of the resistance of the drain-source, as mentioned above.

Adjusting the pulse level to match the desired drain voltage is performed iteratively with an oscilloscope to measure  $V_D$  during the pulse.

The projects in the Flash package use two methods to define the multi-level waveforms used in flash memory testing (for example, [Figure 3-88](#)). For endurance or disturb testing, that uses the subsite stress / measure looping feature of KITE, the Kpulse application is used to define each



unique voltage waveform. The details for using Kpulse are provided in [How to Generate Basic Pulses, page 5-1](#).

Use Kpulse to define and export each unique waveform. The following procedure details how to create and export a hypothetical program/erase waveform.

### Using Kpulse to create and export Segment Arb™ waveforms

---

**NOTE** *Each segment pulse waveform must have the same total time. The minimum programmed time for any segment is 20 ns (20 E-9), but actual output waveform performance is determined by the channel output capability.*

---

#### To use Kpulse to create and export Segment Arb™ waveforms:

1. Close KITE and KCON, if open.
2. Open Kpulse.
3. Load the Kpulse setup file Kpulse\_Flash\_Example\_01.kps. Click **File > Load Setup**, then double-click Kpulse\_Flash\_Example\_01.kps. If this file is not available, see [Table 3-26](#) and enter the values into the Segment Arb™ definition tables in Kpulse. Kpulse should look similar to [Figure 3-92](#). See [Segment Arb™ waveforms](#) in [Section 5](#) for details on using Kpulse.
  - a. The keyboard version of copy (Ctrl-c) and paste (Ctrl-v) can be used to copy the Segment Arb™ values between channels. This is useful for ensuring that each waveform has the same period (total waveform time).
  - b. To select cells for copying, first move the entry cell to row 1 and Start (V).
    - Hold down the Shift key while using the cursor arrow keys to highlight all the cells in the Segment Arb™ waveform.
    - Press **Ctrl-c** to copy.
  - c. Place the entry cell into an undefined channel (Row 1, Start (V) column) and press Ctrl-v to copy.
    - Use the cursor arrow keys to move around and edit the various cells as necessary.
4. The trigger = 1 values in the 1<sup>st</sup> and 5<sup>th</sup> segments. These are the first segments in the program and erase pulses in a typical two pulse program/erase waveform.
  - In the case of multi-card waveform output, the trigger is not used as a typical trigger, but as a synchronizing signal between pulse cards (see the [Reference Manual, Multi-channel synchronization with the Segment Arb™ Mode, page 11-32](#)).

---

**NOTE** *It is recommended to use trigger = 1 for the first segment of each pulse in a waveform.*

---

5. For each unique waveform, export each to a file following the steps given in the [Exporting Segment Arb™ waveform files, page 5-8](#).
6. The exported Segment Arb™ files cannot be imported back into Kpulse and are saved in the path C:\S4200\kiuser\KPulse\Sarbfiles by default.
7. These waveforms will be chosen in the Subsite Setup Device Stress Properties window used in the FlashEndurance projects as well as FlashDisturb. For example, the waveforms are chosen by clicking the browse ... button on the Device Stress Properties in [Figure 3-114](#). The use of these projects is described below.



Figure 3-92  
Kpulse Segment Arb™ pulse card settings

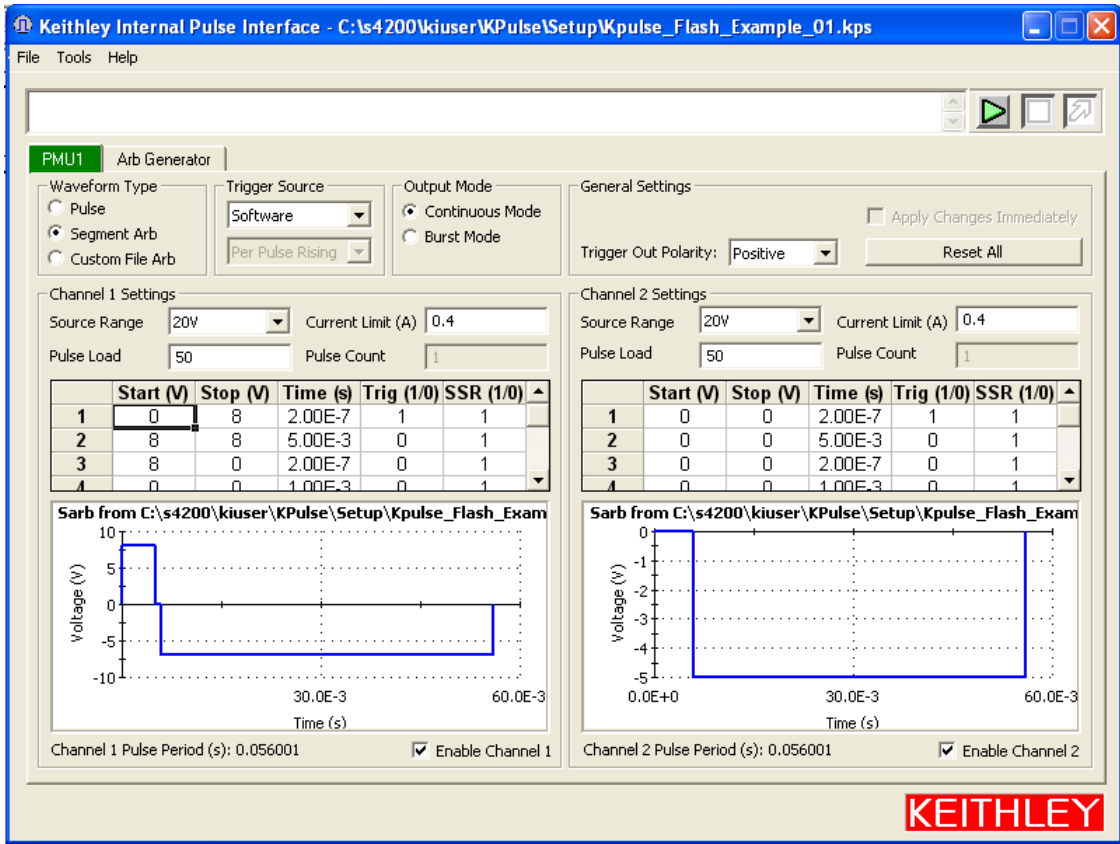


Table 3-26  
Segment Arb™ parameter values for example waveforms

PMU1 Channel 1					
Segment	Start V	Stop V	Time (s)	Trigger	HEOR*
1	0	8	20.00 E-8	1	1
2	8	8	50.00 E-4	0	1
3	8	0	20.00 E-8	0	1
4	0	0	10.00 E-4	0	1
5	0	-7	20.00 E-8	1	1
6	-7	-7	5.00 E-2	0	1
7	-7	0	20.00 E-8	0	1
8	0	0	20.00 E-8	0	1
PMU Channel 2					
Segment	Start V	Stop V	Time (s)	Trigger	HEOR*
1	0	0	20.00 E-8	1	1
2	0	0	50.00 E-4	0	1
3	0	0	20.00 E-8	0	1
4	0	0	10.00 E-4	0	1
5	0	-5	20.00 E-8	1	1
6	-5	-5	5.00 E-2	0	1
7	-5	0	20.00 E-8	0	1
8	0	0	20.00 E-8	0	1



---

**NOTE** HEOR – The solid state relay (SSR) on the output of each pulse channel, that provides a high impedance disconnect.

---

### Entering Segment Arb™ values into UTM array parameters

The second method for defining a Segment Arb™ waveform is by entering values into arrays for the UTM tests:

- Program
- Erase
- Fast-Program-Erase

These UTM-based Segment Arb™ waveforms have been partially pre-defined to reduce the number of parameters required. Figure 3-102 defines the parameters for the single pulse Program and Erase waveforms.

---

**NOTE** The sign of the PulseVoltages array determines whether the pulse is positive (usually for a Program pulse) or negative (usually for an Erase pulse).

---

Figure 3-103 defines the parameters for the dual pulse Program and Erase pulse waveform. Each parameter in these figures has a corresponding array, where each entry in the array represents a pulse channel used in the test.

---

**NOTE** The number of parameters and number of pulse channels in the test must be the same. The period of each pulse waveform must be the same.

---

This UTM method also includes the triggering settings to synchronize multiple PG2 cards, as described above in the Kpulse method, but they are built-in and do not require user modification:

1. Enter the number of pulse channels required for the test (NumPulseTerminals), up to the maximum number VPU channels in the Model 4200-SCS chassis (two channels per pulse card). Figure 3-93 shows four.
2. Enter the channel names for the number of channels specified above.
  - The names are VPU<sub>n</sub>CH<sub>m</sub>, where n is the number of the VPU card (numbered right to left when viewing the back of the Model 4200-SCS chassis) and m is the channel number (one or two), resulting in VPU1CH1,VPU1CH2.
  - There is a comma separator, but no spaces used.
3. Click each array entry and enter the pulse parameter values for each of the four channels.
  - There are five arrays (red arrows in Figure 3-93) for the five pulse parameters shown in Figure 3-102. See Figure 3-94 for two examples of array dialog box displayed after clicking the grey bar on the corresponding UTM parameter (see Figure 3-93, red arrows, one through five).

---

**NOTE** The number of parameters in each array must match the number specified for NumPulseTerminals. If the number of parameters is lower than a previous test, delete the values (**blank, not a zero**) in the unused cells.

---

- a. **PulseVoltages** – Pulse height in volts, assuming a 50 Ω device impedance. The maximum program voltage is 20 V, resulting in a nearly 40 V pulse on a gate or similar



- high impedance terminal. To open the solid state relay during the pulse (as shown for  $V_D$  in [Figure 3-88](#)), use -999.
- b. **PrePulseDelays** – Pre-pulse delay time in seconds. The minimum time is 20 ns (20 E-9), but actual output waveform performance is determined by the channel output capability. All timing delays are made the same across all channels.
  - c. **TransitionTimes** – Rise and fall times in seconds. The minimum time is 20 ns (20 E-9). All transition times are made the same across all channels.
  - d. **PulseWidths** – Pulse width time in seconds (FWHM – Full width half maximum, as shown in [Figure 3-102](#)). The minimum time is 20 ns (20 E-9). All pulse widths are made the same across all channels, but the total waveform time for each channel must be the same.
  - e. **PostPulseDelays** – Post-pulse delay time in seconds. The minimum time is 20 ns (20 E-9). All timing delays are made the same across all channels.
4. Enter the value for NumPulses.
    - For typical characterization, use `NumPulses = 1`. Setting a higher number is useful for testing multiple pulses before performing a SMU measurement
  5. Enter the number of SMUs being used as DC bias terminals, that is useful when DUT terminals require a DC bias to address a particular device.
    - For a direct connection to a single DUT, as shown in [Figure 3-95](#), enter 0. For [Figure 3-96](#), SMU 4 is used to DC bias Bit Line 1, so `NumSMUBiasTerminals = 1`.
    - Enter the SMU numbers, as a string.
      - a) For [Figure 3-95](#), leave `SMUBiasTerminals` blank.
      - b) For [Figure 3-96](#), use `SMUBiasTerminals = SMU4`.
    - If more than one bias terminal is required list all SMUs in ascending order separated by a comma, but no spaces.
  6. Enter the array of bias voltages for the SMUs listed in the previous two steps.
    - The number of values in the array must match the value of `NumSMUBiasTerminals`.
    - If a entry is not needed, delete the value and leave it blank, not 0.
  7. Enter the number of SMUs that are sharing a cable with a pulse channel into `NumSharedSMUs`.
    - Sharing means that one pulse and one SMU signal are combined to a single DUT terminal.
      - a) [Figure 3-95](#) shows four SMUs are paired with a pulse channel, with each SMU/pulse pair sharing a cable to a terminal.
      - b) [Figure 3-96](#) shows that three pairs of SMU/pulse channels are shared.
    - The SMA tees on each of the top three SMUs that incorporate both a pulse channel and a SMU signal into a single cable to a DUT terminal.
    - Supplying the shared SMU information allows the software to open the SMU relay during the pulse output, that is necessary to permit good pulse fidelity.
    - If a switch matrix is used in the configuration (see [Figure 3-97](#)), then use `NumSharedSMUs = 0`.
  8. Enter the SMU IDs for the SMU(s) sharing a cable with a pulse channel into `SharedSMUs`.
    - For the configuration in [Figure 3-96](#), `SharedSMUs = SMU1,SMU2,SMU3`.

---

**NOTE** *There are **NO** spaces allowed in the `SharedSMUs` string.*

---

Both of the Segment Arb™ definition methods are required and are test dependent. For all UTM, the UTM array approach is used.

For any stress / measure loop tests, such as endurance or disturb, use `Kpulse` to define and export the waveform files, then import waveforms into the Device Stress Properties.



When the same waveform is required in the stress / measure Device Stress Properties and in a UTM, the same waveform information must be manually entered using both methods.

In addition to the waveform definition, the interconnect between cards is necessary to provide synchronized multi-channel Segment Arb™ output.

The interconnection information below is for a typical two card (4 pulse channel) Model 4200-SCS FLASH configuration, using four Source Measure Units (SMUs).

In addition to the cabling there are corresponding parameters in the Segment Arb™ table that must be set. This is also covered in the [Reference Manual, Multi-channel synchronization with the Segment Arb™ Mode, page 11-32](#).

Figure 3-93  
Flash-NAND Project Definition Tab, including arrows for the 6 input arrays

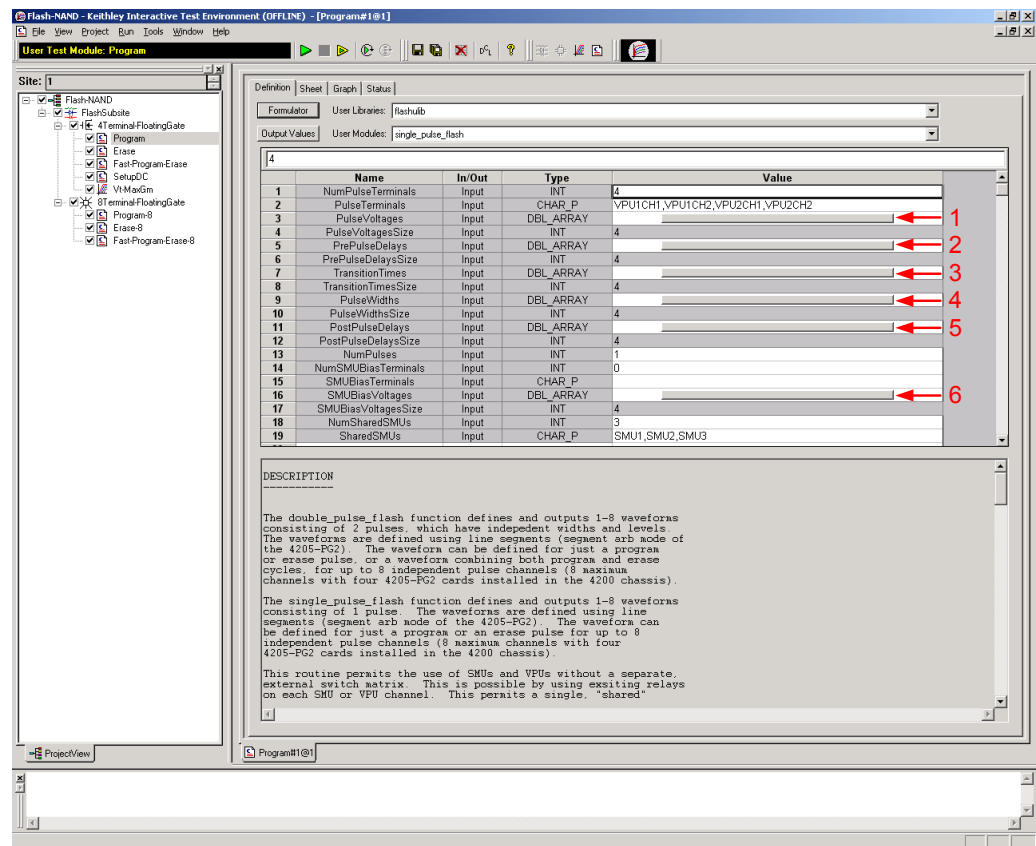
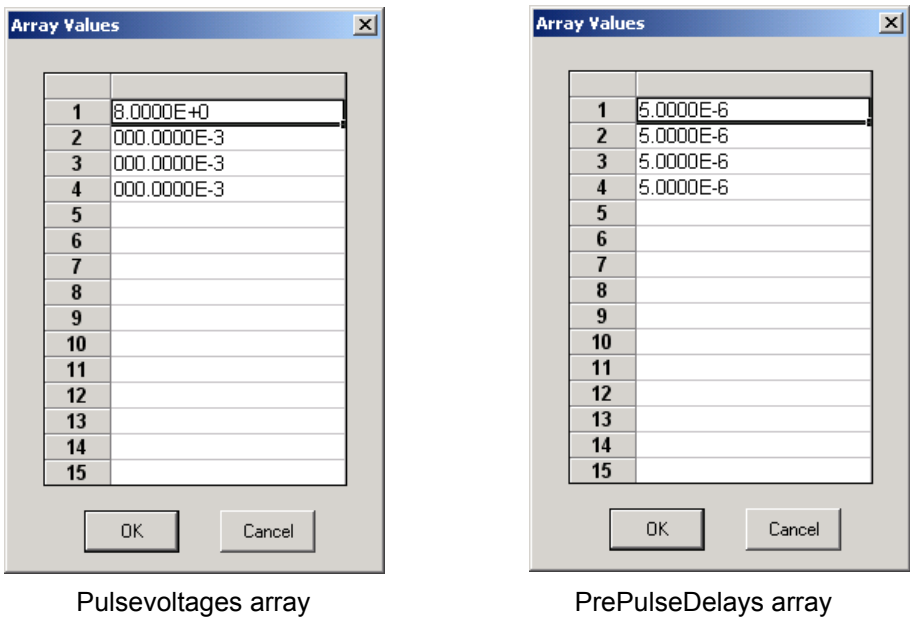




Figure 3-94  
Flash-NAND pulsevoltages array entry and prepulsedelays entry



Flash connections

The Flash package includes all the necessary cables and adapters required for the test connections. Also included is an 8 in / lb torque wrench for tightening the SMA connections.

The Model 4200-SCS Flash package has four channels of multi-level pulse capability. The number of SMUs is configurable. For a system without switching, it is best to have four SMUs, to match the number of pulse channels to connect to a three or four terminal DUT.

For a direct connect configuration, the minimum number of pulse channels is equal to the number of DUT terminals that need to be simultaneously pulsed, including terminals that must change from connected to disconnected, or open, states (see [Figure 3-82](#) and [Figure 3-88](#)), for either the program or erase condition.

The minimum number of SMUs is determined by the measurement tests and the number of DUT terminals.

**NOTE** The 4205-PG2 card is referred to as a VPU, voltage pulse unit, in the software.

[Figure 3-98](#) shows the items that are supplied with the Flash package.

Interconnect diagrams for flash testing are shown in [Figure 3-95](#), [Figure 3-96](#) and [Figure 3-97](#). [Figure 3-95](#) shows the connections for test configuration shown in [Figure 3-85](#) and [Figure 3-89](#), that is used for both initial program/erase investigation and endurance testing of a direct connect DUT.

This configuration does not require a switch matrix, and provides four channels of pulse and well as four SMUs, to permit full characterization of single (non-array) NVM DUT.



Figure 3-95

**Flash connections – program/erase and endurance testing using direct connection to a single, stand-alone 4-terminal device**

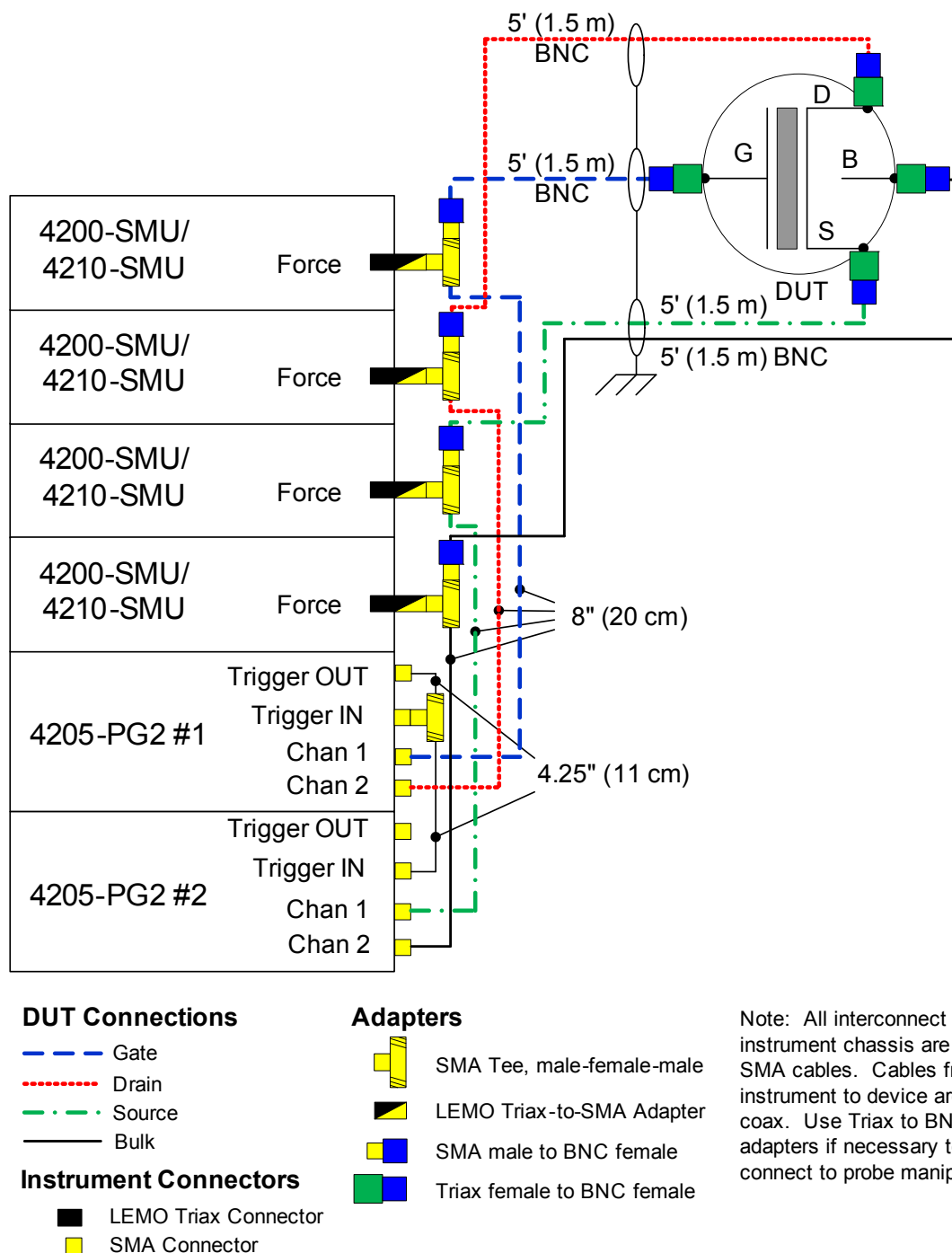


Figure 3-96 shows the connections for test similar to Figure 3-91, that is used for disturb testing. It is strongly recommended to use a switch matrix for testing array test structures, whether for endurance or disturb. However, it is possible to perform a limited test of an array structure without using a switch matrix, as one example is shown in Figure 3-96.

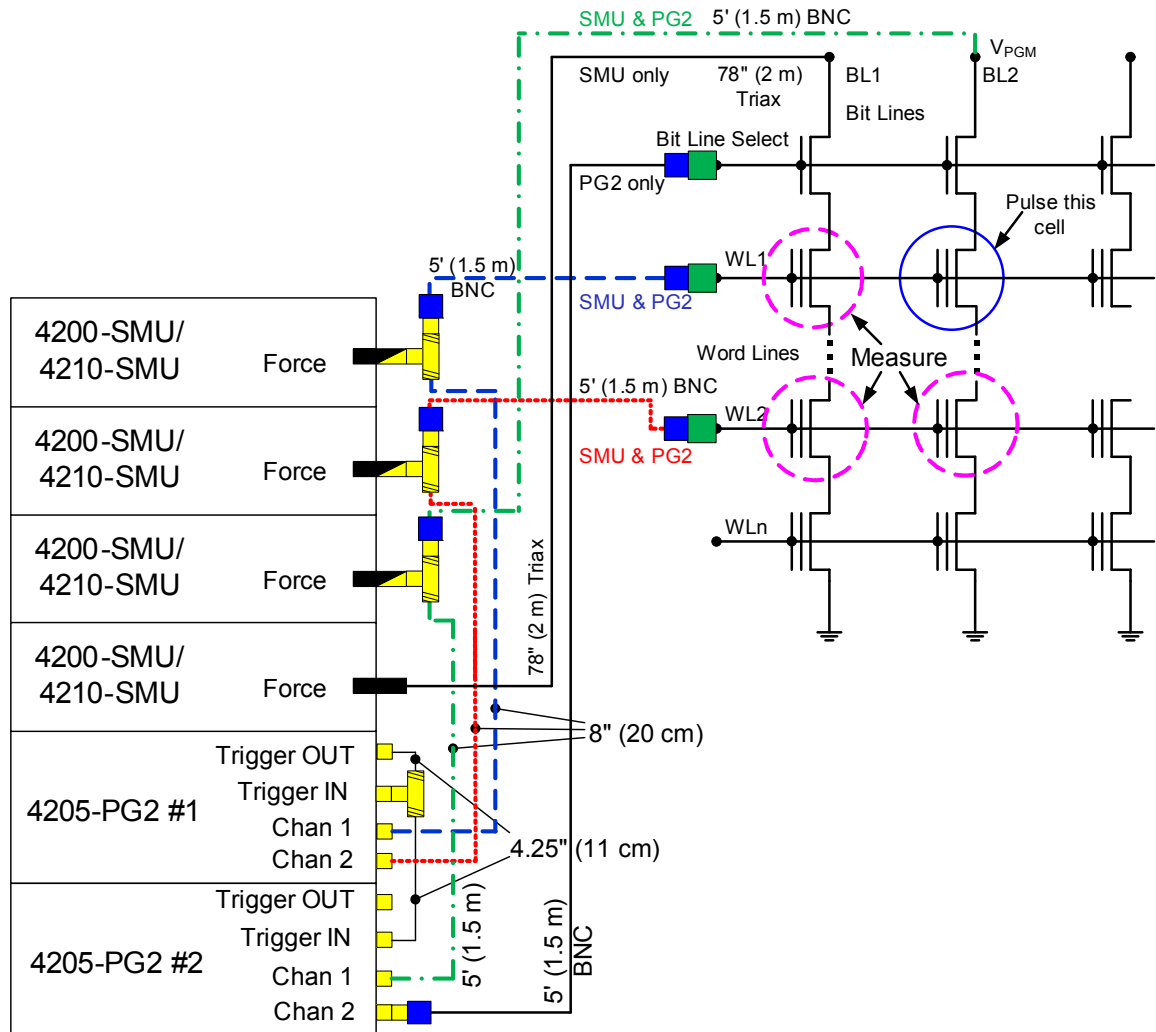
Figure 3-96 shows connection to an array test structure, where one of the four SMU+PG2 channels was split, to provide a total of five test signals to provide the minimum necessary



channels for the select pins (Bit Line Select, Bit Lines 1 and 2), to the pulse DUT (circled in blue), and the measure DUTs (circled in dashed purple).

Figure 3-96 allows for pulsing one DUT, while performing disturb measurements on the three DUTs labeled Measure. The preferred connection method for disturb testing, or any testing of an array DUT, is to add a switch matrix, as shown in Figure 3-97.

Figure 3-96

**Flash direct DUT connections – Disturb testing****DUT Connections**

- WL1 / Gate
- WL2 / Gate
- · - · - BL2 / Drain
- PG: BL Select / SMU: BL1

**Instrument Connectors**

- LEMO Triax Connector
- SMA Connector

**Adapters**

- SMA Tee, male-female-male
- LEMO Triax-to-SMA Adapter
- SMA male to BNC female
- Triax female to BNC female

Note: All interconnect on instrument chassis are white SMA cables. Cables from the instrument to device are BNC coax, except for the direct SMU4 connection, which is black Triax. Use Triax to BNC adapters if necessary to connect to probe manipulators.



Figure 3-97

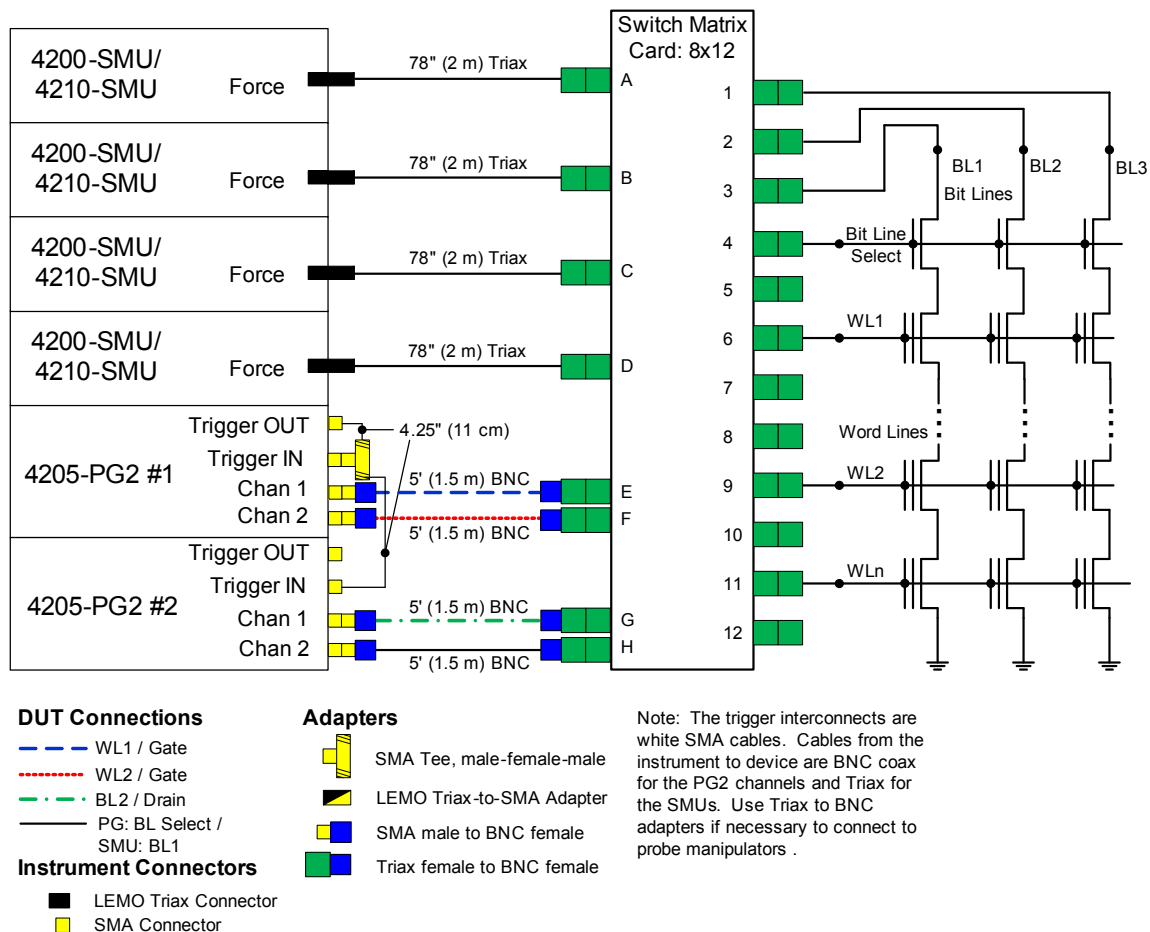
**Flash Switch connections – characterization, endurance or disturb testing**

Figure 3-98

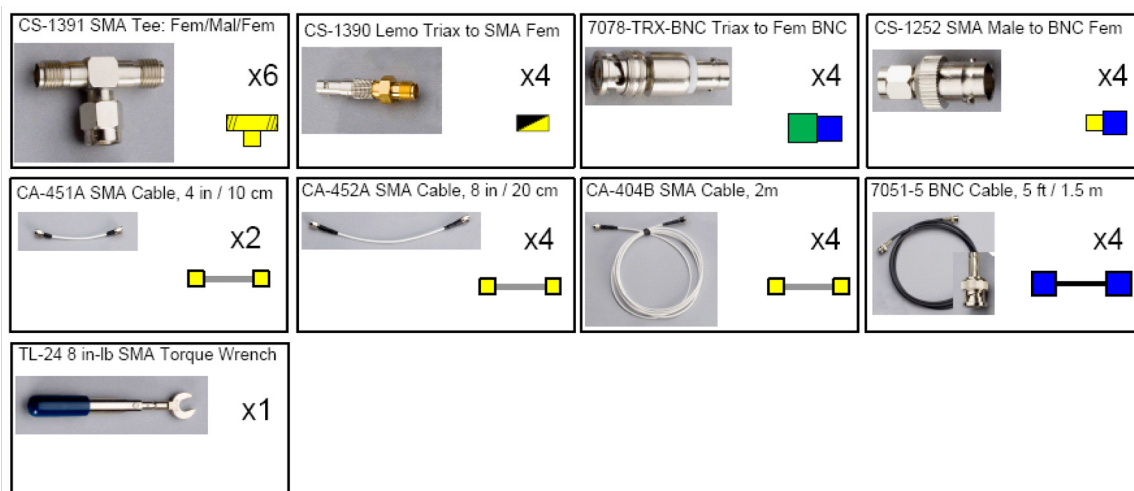
**Supplied items for Model 4200-SCS Flash package**



Table 3-27  
Interconnect parts for Model 4200-SCS FLASH package

Quantity	Description	Comment
6	SMA Tee, female – male – female	Trigger, combine SMU and PG2 channels
4	LEMO triax to SMA adapter	Adapt SMU Force output to SMA for signal interconnect
4	3 slot male triax to female BNC adapter	Convert BNC cabling to Triax for prober or switch matrix connection
4	SMA male to BNC female adapter	Adapt Tee to BNC for cabling from instrument to probe manipulators
2	4.25 in (10.8) cm white SMA cables	Interconnect for triggering
4	8 in (20.3 cm) white SMA cables	Interconnect between PG2 and SMU signals
4	6.6 ft (2 m) white SMA cables	
4	5 foot/ 1.5 m BNC cable	Connect to probe manipulators

Table 3-28  
Tools supplied with the Model 4200-SCS FLASH package

Quantity	Description
1	SMA Torque wrench, 8 in-pound, with 5/16 head installed

---

**NOTE** Use the supplied torque wrench to tighten each connection as it is assembled. Always connect and torque adapter/cable assemblies before attaching the assembly to the instrument cards. Pre-torquing eliminates any non-axial stress on the bulkhead connectors on the SMU or pulse cards, that could possibly cause damage to the cards installed in the Model 4200-SCS chassis, requiring repair.

---

To remove the LEMO triax-to-SMA adapter from a SMU, pull on the knurled silver portion of the connector to release the latches and permit the adapter to separate from the SMU connector.

---

**CAUTION** Failure to fully disengage the LEMO adapter latches may result in damage to the adapter and/or the SMU, requiring repair.

---

The connection instructions below assume a four channel Model 4200-SCS FLASH system, consisting of two 4205-PG2 cards (4 pulse channels) as well as four SMUs, either Model 4200-SCS SMU or 4210-SMU, with SMU preamps removed (see [Figure 3-83](#) or [Figure 3-84](#)).

To test on-wafer devices, there are various ways to connect the supplied SMA cables to the probe manipulators. For the direct connect method (see [Figure 3-95](#) and [Figure 3-96](#)) or switch method (see [Figure 3-97](#)), adapters convert the BNC cabling to the Triax connector compatible with many types of probe manipulators.

## Direct connection to single DUT

Cabling instructions for direct connect to single DUT are below. Refer to [Figure 3-95](#) for the following procedure.

---

**NOTE** In all of the following steps apply sufficient torque using the wrench.

---



These instructions are compatible with the following projects in the Projects\\_Memory folder:

- Flash-NAND
- Flash-NOR
- FlashDisturb-NAND
- FlashDisturb-NOR
- FlashEndurance-NAND
- FlashEndurance-NOR

1. Set up the Model 4200-SCS, referring to the [Getting Started, page 1-1](#), [Reference Manual, Installation, page 2-1](#), and [Connections and Configuration, page 4-1](#). For examples of cables, adapters, and connectors see [Reference manual, Figure 11-35](#).
2. Take one of the SMA Tees and connect the two shortest (4.25 inch or 10.8 cm) SMA cables to either end.
3. Connect this assembly to the right-most PG2 card, that is, the PG2 card in the lowest numbered slot.
  - First connect one of the SMA cables to TRIGGER OUT
  - Connect the SMA tee to TRIGGER IN.
4. Then connect the other SMA cable to TRIGGER IN on the second PG2 card. This second card is the card to the immediate left of the card in step 3.

---

**NOTE** *If the FLASH package consists of more than two PG2 cards, continue to connect the cable and Tees to the adjacent cards. Torque both connections using the wrench.*

*For a system consisting of four PG2 cards, there should be three SMA tees used to connect the triggering across the four cards.*

---

5. Take an SMA Tee and connect one SMA-to-BNC adapter to one of the female connectors.
6. Connect the assembly from step 5 to one Triax-to-SMA Adapter.
7. Connect one 8 inch (20.3 cm) SMA cable to the remaining SMA female connector.
8. Connect one 5 foot (1.5 m) black BNC cable to the BNC connection.
9. Perform steps 5-8 three more times.
10. Take one of the cable assemblies from step 9, connect the SMA to CHANNEL 2 of the PG2 in the left most slot (PG2 in the slot with the highest number).
11. Carefully insert the LEMO Triax connector from step 10 into the Force connector on the SMU in Slot 4.
12. Route BNC cable from SMU4 to the DUT terminal Bulk connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
13. Take one of the cable assemblies from step 9, connect the SMA to CHANNEL 1 of the PG2 in the left most slot (PG2 in the slot with the highest number).
14. Carefully insert the LEMO Triax connector from step 13 into the Force connector on the SMU in Slot 3.
15. Route BNC cable from SMU3 to the DUT terminal Source connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
16. Take one of the cable assemblies from step 9, connect the SMA to CHANNEL 2 of the PG2 in the right-most slot (PG2 in the slot with the lowest number).
17. Carefully insert the LEMO Triax connector from step 16 into the Force connector on the SMU in Slot 2.
18. Route BNC cable from SMU2 to the DUT terminal Drain connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.



19. Take one of the cable assemblies from step 9, connect the SMA to CHANNEL 1 of the PG2 in the right-most slot (PG2 in the slot with the lowest number).
20. Carefully insert the LEMO Triax connector from step 19 into the Force connector on the SMU in Slot 1.
21. Route BNC cable from SMU1 to the DUT terminal Gate connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.

## Direct connection to array DUT for disturb testing

Cabling instructions for direct connect to an array DUT are below. Refer to [Figure 3-96](#) for the following procedure. These instructions are compatible with the following projects in the Projects\\_Memory folder:

- FlashDisturb-NAND
- FlashDisturb-NOR
- FlashEndurance-NAND
- FlashEndurance-NOR

---

**NOTE** *In all of the following steps, when necessary torque both connections using the wrench.*

---

1. Set up the Model 4200-SCS, referring to the [Getting Started, page 1-1](#), [Reference Manual, Installation, page 2-1](#), and [Connections and Configuration, page 4-1](#).
2. Take one of the SMA Tees and connect the two shortest (4.25 inch or 10.8 cm) SMA cables to either end.
3. Connect this assembly to the right-most PG2 card, that is, the PG2 card in the lowest numbered slot. First connect one of the SMA cables to TRIGGER OUT and connect the SMA tee to TRIGGER IN.
4. Then connect the other SMA cable to TRIGGER IN on the second PG2 card. This second card is the card to the immediate left of the card in step 3.

---

**NOTE** *If the FLASH package consists of more than two PG2 cards, continue to connect the cable and Tees to the adjacent cards. For a system consisting of four PG2 cards, there should be three SMA tees used to connect the triggering across the four cards.*

---

5. Take one SMA-to-BNC adapter and connect one 5 foot (1.5 m) black BNC cable.
6. Take the cable from step 5 and connect the SMA adapter to CHANNEL 2 of the PG2 in the left most slot (PG2 in the slot with the highest number).
7. Route BNC cable from step 6 to the DUT array Bit Line Select connection. Use a Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
8. Take one black Lemo Triax to 3-slot Triax cable and insert the LEMO end into the Force connection on the left-most SMU in slot 4.
9. Route triax from SMU4 to the DUT array BL1 connection. Connect cable to probe manipulator.
10. Take an SMA Tee and connect one SMA-to-BNC adapter to one of the female connectors.
11. Connect the assembly from step 5 to one Triax-to-SMA Adapter.
12. Connect one 8 inch (20.3 cm) SMA cable to the remaining SMA female connector.
13. Connect one 5 foot (1.5 m) black BNC cable to the BNC connection.
14. Perform steps 10-13 two more times.



15. Take one of the cable assemblies from step 14, connect the SMA to CHANNEL 2 of the PG2 in the left most slot (PG2 in the slot with the highest number).
16. Carefully insert the LEMO Triax connector from step 15 into the Force connector on the SMU in Slot 4.
17. Route BNC cable from SMU4 to the DUT array WL2 terminal. Use a Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
18. Take one of the cable assemblies from step 14, connect the SMA to CHANNEL 1 of the PG2 in the left most slot (PG2 in the slot with the highest number).
19. Carefully insert the LEMO Triax connector from step 18 into the Force connector on the SMU in Slot 3.
20. Route BNC cable from SMU3 to the DUT array BL2 connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
21. Take one of the cable assemblies from step 14, connect the SMA to CHANNEL 2 of the PG2 in the right-most slot (PG2 in the slot with the lowest number).
22. Carefully insert the LEMO Triax connector from step 21 into the Force connector on the SMU in Slot 2.
23. Route BNC cable from SMU2 to the DUT array WL2 connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.
24. Take one of the cable assemblies from step 14, connect the SMA to CHANNEL 1 of the PG2 in the right-most slot (PG2 in the slot with the lowest number).
25. Carefully insert the LEMO Triax connector from step 24 into the Force connector on the SMU in Slot 1.
26. Route BNC cable from SMU1 to the DUT array WL1 connection. Connect Triax-to-BNC adapter, if necessary. Connect cable to probe manipulator.

## Switch matrix connection to array DUT

Cabling instructions for switch connect to an array DUT are below. Refer to [Figure 3-97](#) for the following procedure. These instructions are compatible with the following projects in the Projects\\_Memory folder:

- Flash-Switch
- FlashDisturb-Switch
- FlashEndurance-Switch

Unlike the direct connect methods described above, the use of a switch matrix permits the use of SMU preamp.

1. Set up the Model 4200-SCS, referring to the [Getting Started, page 1-1](#), [Reference Manual, Installation, page 2-1](#), and [Connections and Configuration, page 4-1](#).
2. Set up the 707 or 708 Switch matrix using the [Reference manual, Appendix B, Using KCON to add a switch matrix to the system](#). See [Figure 3-99](#) for a suggested configuration for the row-column connection, consistent with [Figure 3-97](#).
3. Take one of the SMA Tees and connect the two shortest (4.25 inch or 10.8 cm) SMA cables to either end.
4. Connect this assembly to the right-most PG2 card, that is, the PG2 card in the lowest numbered slot. First connect one of the SMA cables to TRIGGER OUT and connect the SMA tee to TRIGGER IN.
5. Then connect the other SMA cable to TRIGGER IN on the second PG2 card. This second card is the card to the immediate left of the card in step 4.



**NOTE** *If the FLASH package consists of more than two PG2 cards, continue to connect the cable and Tees to the adjacent cards. For a system consisting of four PG2 cards, there should be three SMA tees used to connect the triggering across the four cards.*

6. Take one SMA-to-BNC adapter and connect one 5 foot (1.5 m) black BNC cable.
7. Take the cable from step 5 and connect the SMA adapter to CHANNEL 2 of the PG2 in the left most slot (PG2 in the slot with the highest number).
8. Route BNC cable from step 7 to the switch matrix card Triax input, using a Triax-to-BNC adapter.
9. Repeat steps 6-8 for the other three PG2 pulse channels.
10. Take one black Lemo Triax to three-slot Triax cable and insert the LEMO end into the Force connection on the left-most SMU in slot four.
11. Route triax from SMU4 to the switch matrix card Triax input.
12. Repeat steps 10-11 for the remaining three SMUs.
13. Use triax cables to route the switch matrix outputs to the array DUT probe manipulators.

Figure 3-99

**KCON Row-Column Card Properties for Flash testing with four SMUs and four VPU pulse channels**

The screenshot shows a software window titled 'Properties' with a tab labeled 'Card Properties'. Inside, it specifies 'Model: Keithley 7174 Low Current Matrix Card' and 'Slot: 1'. Below this, there are two main sections: 'Rows' and 'Columns'. The 'Rows' section contains eight dropdown menus labeled A through H, with the following selections: A: SMU1 Force, B: SMU2 Force, C: SMU3 Force, D: SMU4 Force, E: VPU1 Channel 1, F: VPU1 Channel 2, G: VPU2 Channel 1, and H: VPU2 Channel 2. The 'Columns' section contains twelve dropdown menus labeled 1 through 12, each with the selection 'Pin X Force' (e.g., 1: Pin 1 Force, 2: Pin 2 Force, ..., 12: Pin 12 Force).

## Flash projects

There are three types of projects included with the Model 4200-SCS FLASH package:

- Initial Characterization
- Endurance
- Disturb



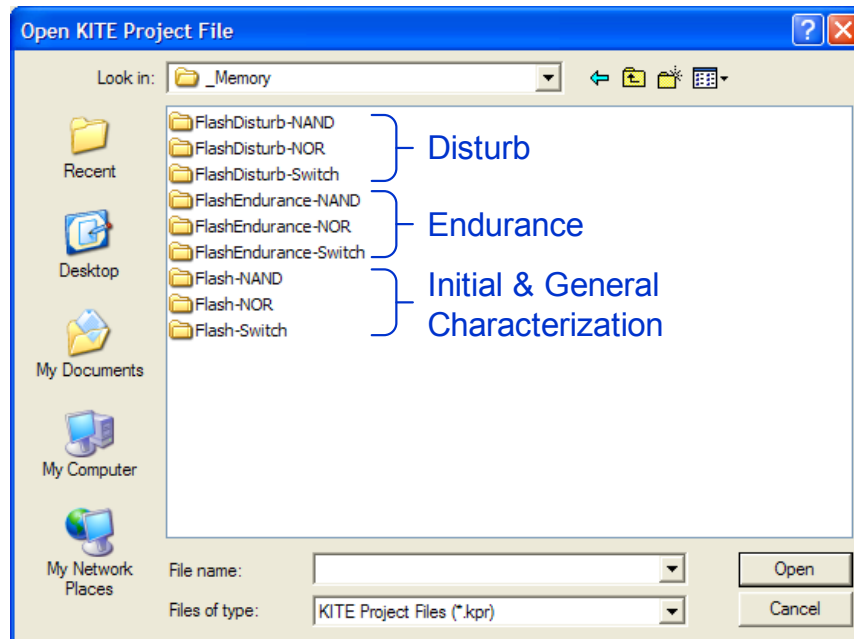
Each type of project has three different sets of defaults for common setups:

- NAND device (direct connect)
- NOR device (direct connect)
- Switch (using a Keithley 707A/708A switch matrix and compatible card(s)).

This results in the nine different projects in the Projects\\_Memory folder as shown in [Figure 3-100](#).

Figure 3-100

**Project listing \_Memory folder**



There are three similar projects that provides the ability to send  $n$  pulses to the DUT, then perform a  $V_T$  sweep.

- Flash-NAND project
- Flash-NOR project
- Flash-Switch project

The pulses can be either a single pulse program or erase waveform, or the combined program and erase waveform. [Figure 3-101](#) shows the program and erase tests for the Flash-NAND project. These tests allow investigation into program and erase state dependencies on pulse parameters. There are three different waveform types available:

- Program Erase
- Fast Program
- Erase

The Program waveform and Erase waveform output pulses with a single set of parameters for the pulse width, transition (0-100% rise/fall), and level (see [Figure 3-102](#)).

The Fast Program and Erase test waveform uses two pulses, that can have independent widths and levels (see [Figure 3-103](#)).

Each test permits programming the pulse width, level and transition (0-100% rise/fall) parameters, as well as the number of pulses.

For extended Program Erase cycling, use one of the FlashEndurance projects.



Instead of a voltage the disconnected or open state may also be chosen for any pulse segment. The open state is useful when tunneling for programming or erasing a floating gate transistor.

These projects support from one to eight pulse channels to support typical 4-terminal devices, as well as higher pin count devices or array test structures.

---

**NOTE** *The 8-terminal testing requires four Model 4205-PG2 cards and, for most tests, a compatible external switch matrix.*

---

The purpose of these projects is to initially characterize a floating gate transistor. For example, determine the appropriate pulse parameters for both the program and erase waveforms to reach a target  $V_{TE}$  and  $V_{TP}$ .

This can be done by setting the Program pulse height to the desired value, but setting the pulse width to a fraction of the expected pulse width ([Entering Segment Arb™ values into UTM array parameters](#)).

1. Set the NumPulses to one and uncheck the Erase and Fast-Program-Erase tests.
2. Run the Program, SetupDC and  $V_t$ -MaxGm tests, monitoring the shift in the  $V_T$  and noting the number of pulses required to reach the target  $V_{TP}$ .

Then the same approach can be used for the erase. If the DUT was initially in an unknown state, the determination of appropriate pulse parameters for the program and erase waveforms may be iterative.

The Fast-Program-Erase test may be used to confirm that the chosen pulse parameters are providing an acceptable erase, and the  $V_T$  after the Fast-Program-Erase is not shifting.

Once acceptable pulse parameters are determined, use Kpulse to define and export the waveforms for use in the Endurance and Disturb projects (see [Using Kpulse to create and export Segment Arb™ waveforms](#)).

The difference between the Flash-NAND and Flash-NOR are the typical pulse widths and levels specific to the DUT type. The Flash-switch is a generic example of the Flash testing described above, but adds support for an external Keithley switch matrix.

## Flash-NAND tests

Flash-NAND tests consist of the following tests:

- Program
- Erase
- Fast-Program-Erase
- SetupDC
- $V_t$ -MaxGm
- Program-8
- Erase-8
- Fast-Program-Erase-8

---

**NOTE** *The Flash-NAND project navigator is shown in [Figure 3-101](#).*

---



Figure 3-101  
Flash-NAND project

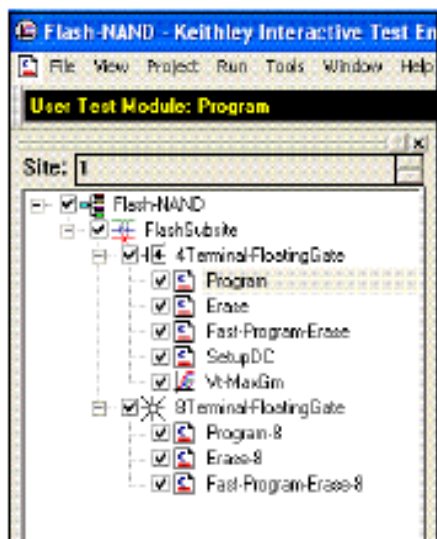


Figure 3-102  
Parameters for Program or Erase UTMs (using single\_pulse\_flash module)

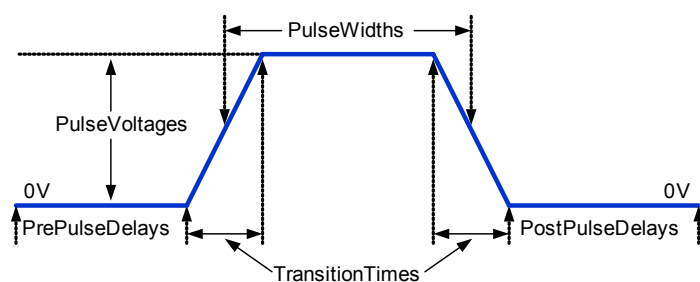
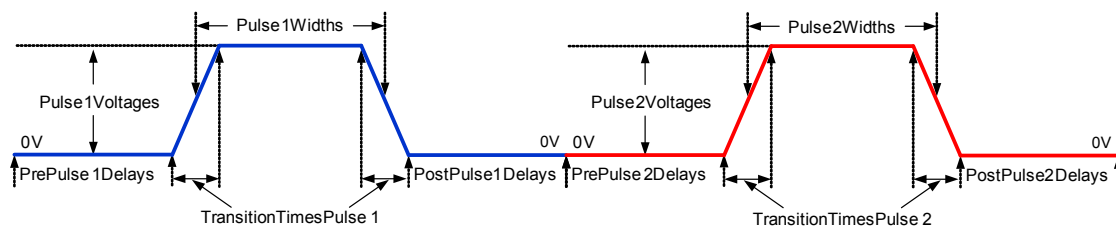


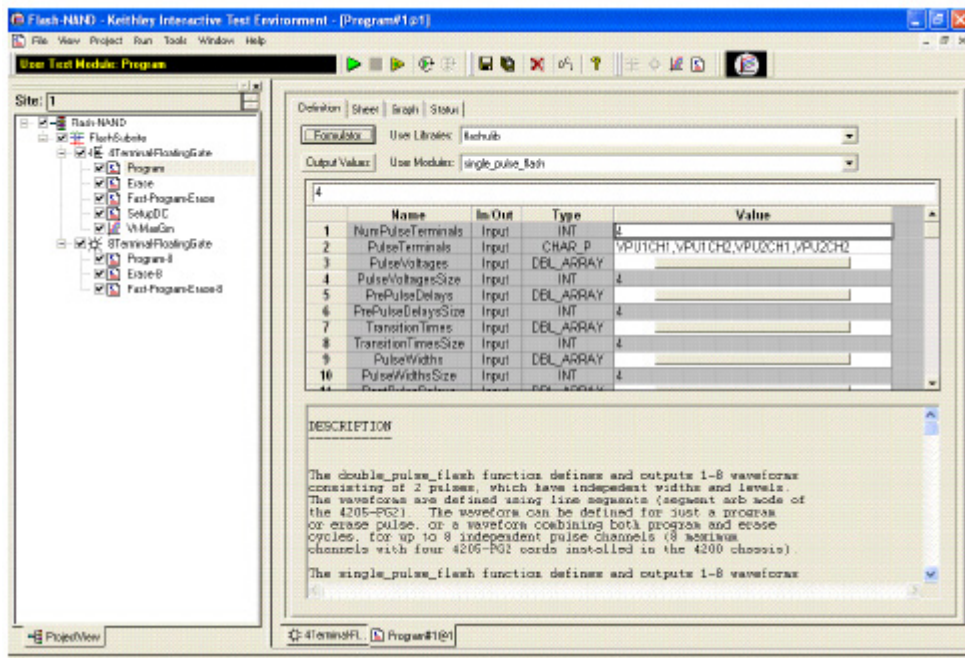
Figure 3-103  
Parameters for Fast Program-Erase pulse waveform (using double\_pulse\_flash module)



**Program test** – This test uses a partially pre-defined waveform, see [Figure 3-102](#), to program a flash memory device. The Definition tab for this test is shown in [Figure 3-104](#).

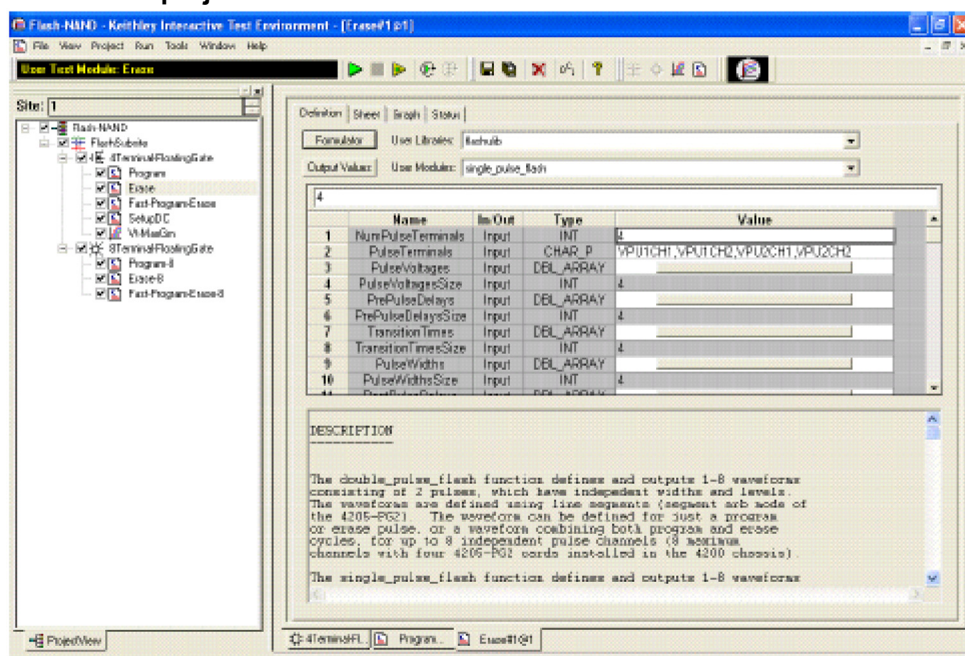


Figure 3-104  
Flash-NAND project – Program definition tab



**Erase test** – This test uses a partially pre-defined waveform, see [Figure 3-102](#), to erase a flash memory device. The Definition tab for this test is shown in [Figure 3-105](#).

Figure 3-105  
Flash-NAND project – Erase definition tab

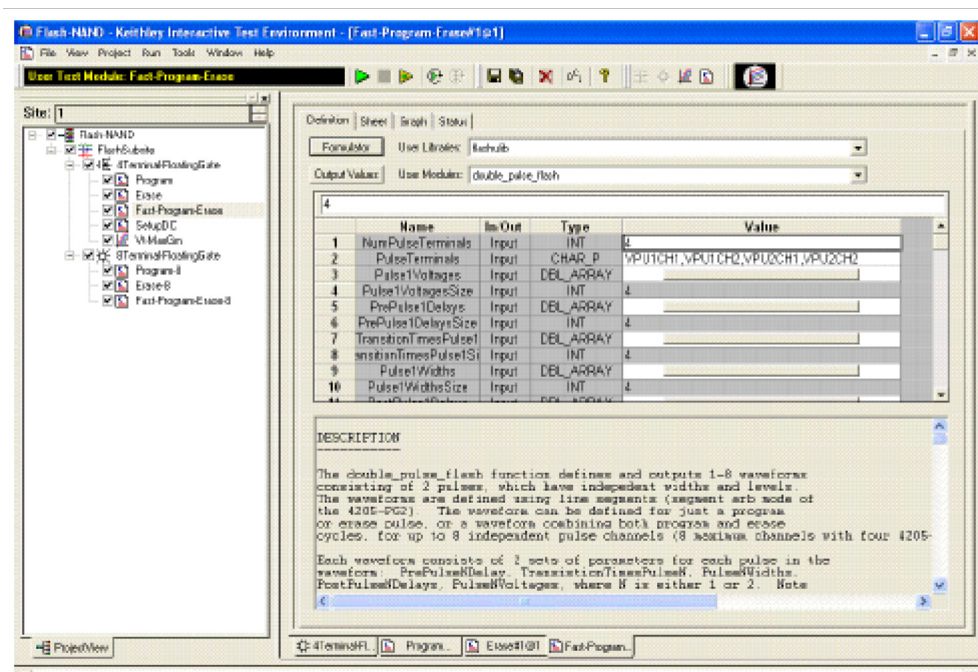




**Fast-Program-Erase test** – This test uses a partially pre-defined waveform, see [Figure 3-103](#), to program and erase a flash memory device. The Definition tab for this test is shown in [Figure 3-106](#).

Figure 3-106

#### Flash-NAND project – Fast-Program-Erase definition tab



**SetupDC test** – The Definition tab for this test is shown in [Figure 3-107](#). This test isolates the VPU outputs from the DUT, allowing the SMUs to perform a DC without signal interference from the pulse outputs. It does this by opening the HEOR for each VPU channel in the PulseTerminals list.

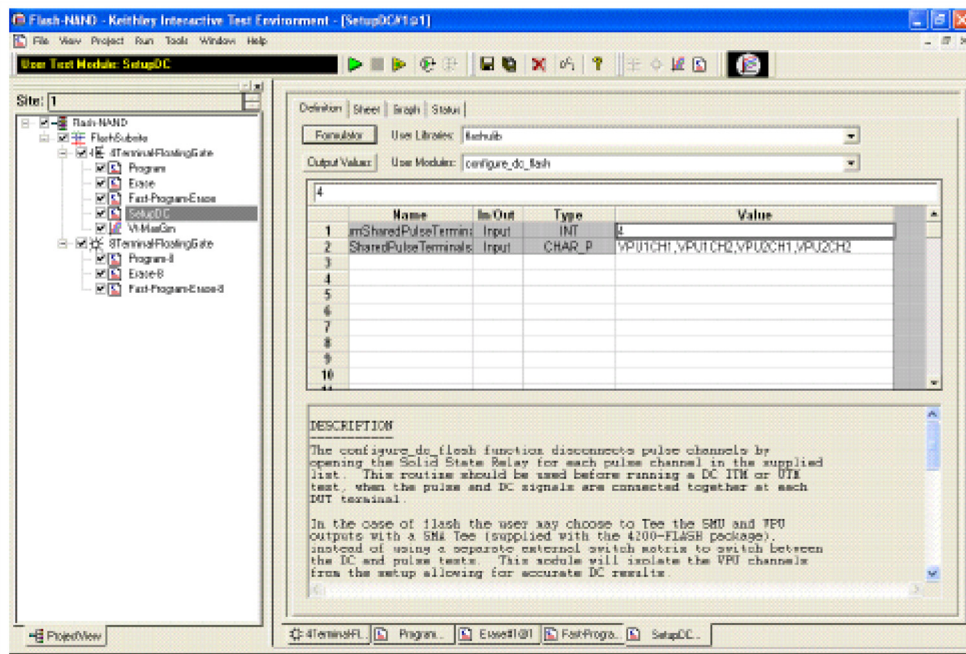
Disconnecting the VPU channels allows for accurate DC results. This isolation step is only necessary when using the direct connect method (see [Figure 3-95](#) and [Figure 3-96](#)), where the SMU and VPU signals are sharing a single connection to each device terminal (see [Figure 3-89](#)).

The same test step is called Open-VPU-Relay, and is optional for switch matrix configurations (see [Figure 3-97](#)), but is recommended to prevent accidental simultaneous connection of both SMU and PG2 channels to a single terminal.

The SetupDC test step is used in the configuration **without** a switch matrix and is required before any DC tests. When using a switch matrix, a ConPin test can replace the SetupDC test (see [Reference Manual, LPT functions, page 8-23](#)) to set the appropriate matrix connections prior to any DC tests.

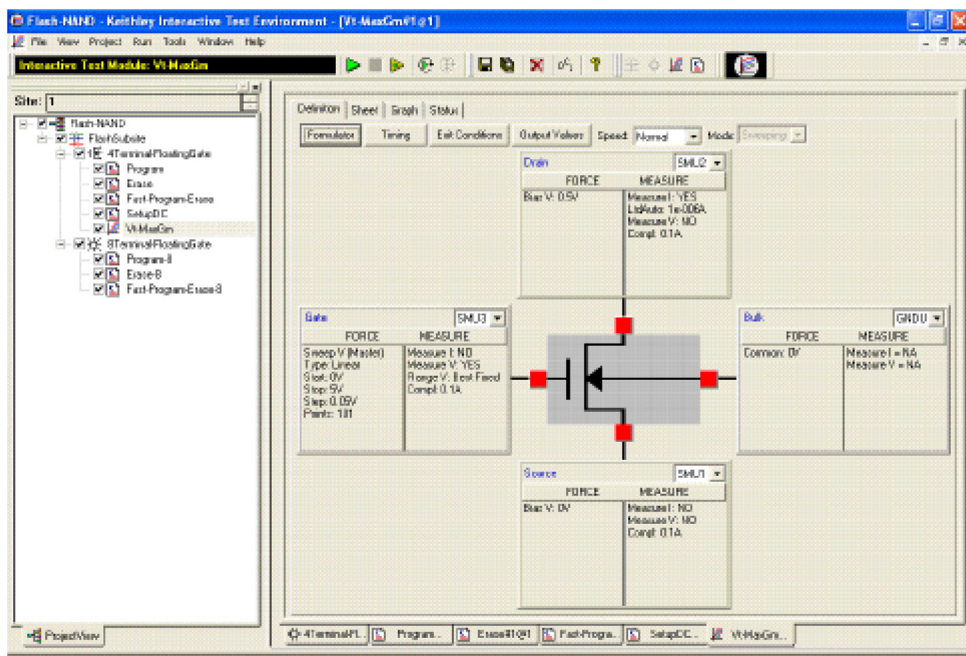


Figure 3-107  
Flash-NAND project – SetupDC definition tab



**Vt-MaxGm test** – This test is used to perform a DC voltage sweep on the gate of the DUT and measure the drain current at each sweep step. The default Definition tab for this test is shown in [Figure 3-108](#). SMU3 is configured to perform a 101 point sweep from 0 to 5 V in 50m V steps. SMU1 is configured to DC bias the drain at 0.5 V and measure current at each step of the sweep.

Figure 3-108  
Flash-NAND project – Vt-MaxGm definition tab



**Program-8 test** – This test uses Segment Arb™ waveforms to program an 8-terminal flash memory device.



**Erase-8 test** – This test uses Segment Arb™ waveforms to erase an 8-terminal flash memory device.

**Fast-Program-Erase-8 test** – This test uses Segment Arb™ waveforms to program and erase an 8-terminal flash memory device.

## Flash-NOR tests

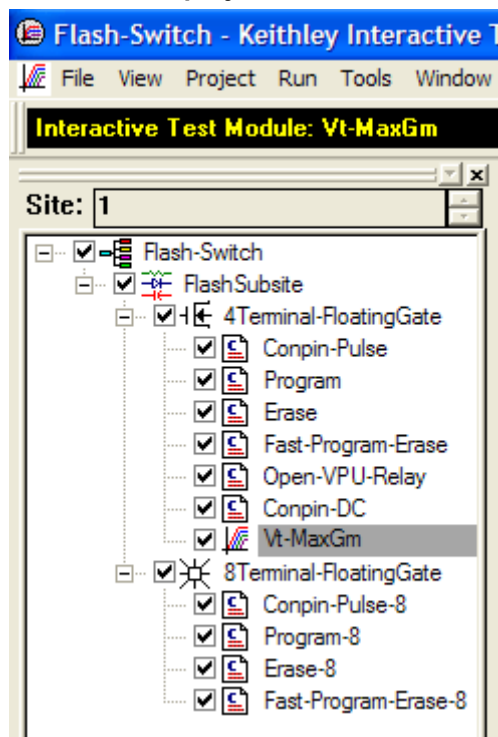
The Flash-NOR project has tests similar to the Flash-NAND project, with parameter defaults for NOR type floating gate DUTs.

## Flash-switch tests

The Flash-Switch project has similar tests to the Flash-NAND, with parameter defaults for using a switch matrix for more complex multi-DUT addressable test structures (see [Figure 3-84](#)). Also, SMU and 4205-PG2 pulse channels are connected to the matrix differently, eliminating the SMU+Pulse sharing of cables to the DUT. Using the switch means that ConPin tests (see [Figure 3-109](#)) are added after the Open-VPU-Relay tests in the direct-connect versions Flash-NAND (see [Figure 3-101](#)) and Flash-NOR.

Figure 3-109

### Flash-Switch project



**ConPin-Pulse or ConPin-DC test** – This test is used to connect pulse or SMUs to the DUT. [Figure 3-110](#) shows the definition tab for ConPin-Pulse. The parameters are typed into the UTM parameter table, with the Pin1, Pin2, etc determining where the instrument (SMU, VPU) signals connect. It is also possible to configure a single switch matrix card using the GUI. Click the GUI button shown in [Figure 3-110](#) to see the dialog in [Figure 3-111](#).



Figure 3-110  
ConPin-Pulse test Definition tab

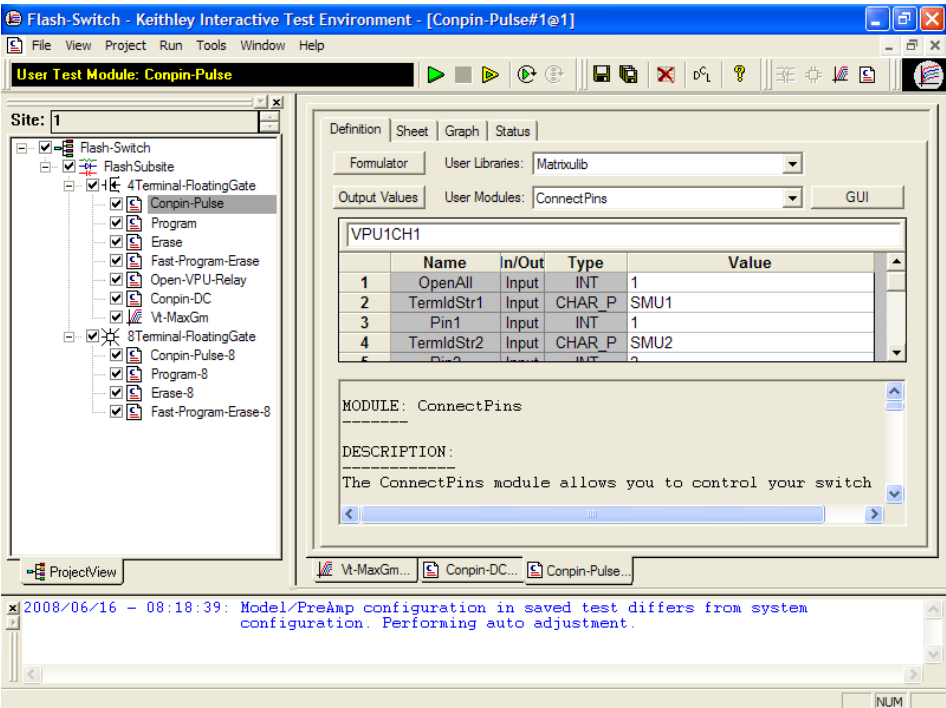
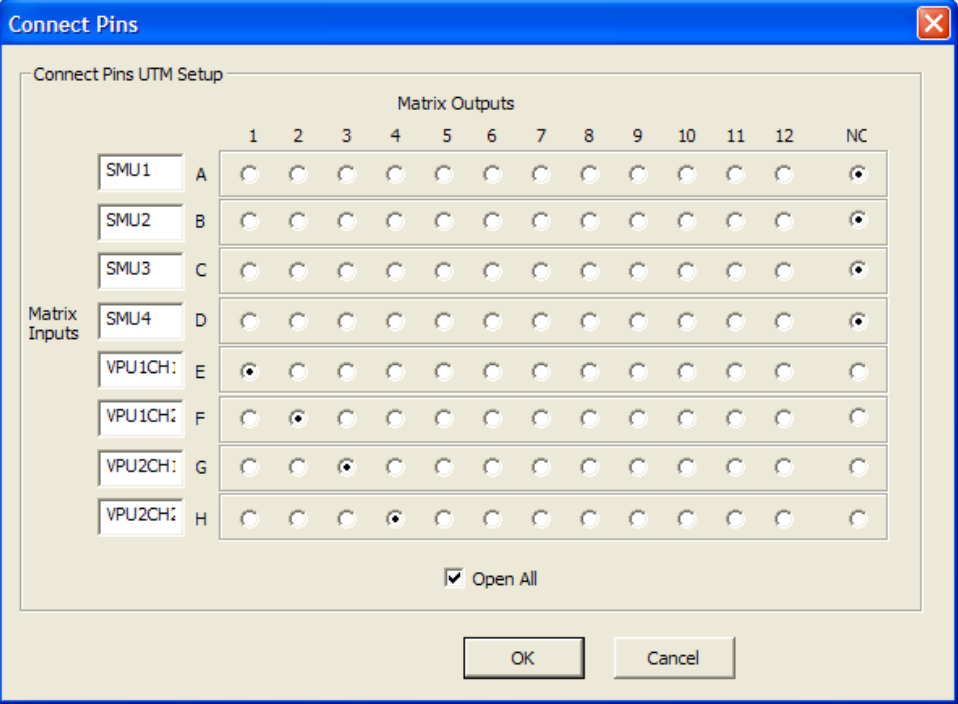


Figure 3-111  
ConPin-Pulse test GUI definition dialog



Running any Flash Project for the first time

- 1. Connect up the Model 4200-SCS FLASH package, using the Flash Connections instructions for one of the following configurations:



- a. Direct connection (no switch matrix) to single DUT: Use [Figure 3-95](#) and [Direct connection to single DUT](#).
- b. Direct connection (no switch matrix) to array DUT: [Figure 3-96](#) and [Direct connection to array DUT for disturb testing](#).
- c. Switch matrix connection to array or single DUT: Use [Figure 3-97](#) and [Switch matrix connection to array DUT](#).
2. See procedures below for using a specific Flash project.
  - a. Initial characterization using the [Running the Flash-NAND, Flash-NOR or Flash-Switch Project](#)
  - b. Endurance or Disturb testing using the [Running a FlashEndurance or FlashDisturb project](#) by using [FlashEndurance-NAND](#), [FlashEndurance-NOR](#), [FlashEndurance-Switch](#), [FlashDisturb-NAND](#), [FlashDisturb-NOR](#), or [FlashDisturb-Switch](#).

## Running the Flash-NAND, Flash-NOR or Flash-Switch Project

The Flash projects use a small number of tests and methods. This section will explain the tests and how to set parameter values.

These projects allow initial characterization of a device, including the determination of the pulse settings (pulse width, height, and transition time) that will provide a target programmed or erased  $V_T$ . After the appropriate pulse settings are determined, they can be used to perform Endurance or Disturb testing on the DUT.

1. If system connections have not been made, follow the instruction in [Running any Flash Project for the first time](#).
2. If KITE is not running, start KITE by double-clicking the KITE icon on the Model 4200-SCS desktop.
3. Open the appropriate KITE Flash project.
  - a. Within KITE, click **FILE > Open Project**. If the dialog window is not displaying the \_Memory folder, move up one or two levels to the display the Projects directory.
  - b. Double-click the \_Memory folder, then double-click the desired Flash test folder (Flash-NAND, Flash-NOR or Flash-Switch)
  - c. Double-click the Flash-NAND.kpr, Flash-NOR.kpr or Flash-Switch.kpr file to open the desired Model 4200-SCS Flash project. KITE should resemble [Figure 3-104](#) or [Figure 3-109](#) for Flash-Switch Project.
4. Touch-down or connect the DUT.
5. Verify setup and connection by running  $V_t$ -MaxGm test
  - a. Set appropriate voltages
  - b. Run the test by clicking the yellow and green triangle Append button.
  - c. Ensure that the  $V_G$ - $I_D$  and  $V_T$  results are reasonable.
6. Determine the appropriate pulse voltage levels.
  - a. Review the section [Pulse waveforms for NVM testing](#).
  - b. Recall that pulse voltage levels on the gate will double
    - i. For example, using PulseVoltage = 2 will result in  $V_G = 4$  V for a typical high-impedance (1 k  $\Omega$ ) terminal.
  - c. Use oscilloscope to determine appropriate PulseVoltage values for the desired  $V_G$  and  $V_D$ , making sure to use the 1 M  $\Omega$  input setting on the oscilloscope.
    - i. Drain: Connect the oscilloscope probe across the drain-source of the DUT.
    - ii. Use the Program and Erase UTMs to output pulses, while using the oscilloscope to measure the pulse height. Iterate by modifying the PulseVoltages to reach the target voltage.
7. Once the appropriate voltage levels are determined, the appropriate pulse width may be determined by iteratively outputting pulses while occasionally measuring the  $V_T$ .



- a. Start by using a pulse width that is shorter than the expected PW. For example, if 20  $\mu$ s is the expected PW, try using a 2  $\mu$ s PW.
- b. Enter the parameter values into the Program UTM, following the procedure in [Running the Program or Erase UTM](#). Initially, set NumPulses = 2, or another small number.
- c. Uncheck the Erase and Fast-Program-Erase tests.
- d. Enter the parameter values for the Vt-MaxGm test, following the procedure in [Running the Vt-MaxGm ITM](#).
- e. Run the test
  - i. Double-click 4Terminal-FloatingGate
  - ii. Press the Run button
- f. Check the graph on the Vt-MaxGm test. It will likely be too low on the first few runs, but note the total number of pulses sent to the DUT. Rerun test until the  $V_T$  has met the target value, and note the total pulse width to use to program the device, using either the Program or Fast-Program-Erase tests.
- g. Repeat the above step with the Erase test, feeding final results into the Erase and Fast-Program-Erase tests.
- h. Ensure that the erase parameters are fully erasing the DUT
  - i. Set the parameters in the Fast-Program-Erase test. Set NumPulses = 10.
  - ii. Uncheck the Program and Erase tests.
  - iii. Double-click 4Terminal-FloatingGate. Press run
  - iv. Note the  $V_T$ .
  - v. Change NumPulses = 100 or a larger number
  - vi. Double-click 4Terminal-FloatingGate. Press Append.
  - vii. Note the  $V_T$ . If the  $V_T$  value for the tests are similar, then the erase pulse is fully erasing the DUT.

## Running the Program or Erase UTM

These tests are located in all of the Flash projects. It outputs a number of pulses (Parameter = NumPulses) with a shape shown similar to [Figure 3-102](#), on a number of pulse channels (Parameters = NumPulseTerminals, PulseTerminals).

1. Enter the number of pulse waveforms required into NumPulseTerminals. This must be a minimum of one channel, up to a maximum of the number of channels available. For a Flash system with 2 4205-PG2 cards, there are four pulse channels available.
2. Enter which pulse channels will be used into PulseTerminals. This is a string of channels, in the form VPU1CH1,VPU1CH2,VPU2CH1,VPU2CH2. VPU1 is the 4205-PG2 in the lowest-numbered slot (right-most slot when looking at back of Model 4200-SCS chassis).
  - a) The characters are all capitalized and each channel is separated by a comma.
  - b) No spaces are allowed in the PulseTerminals string.
3. Enter the values in the five arrays that define the pulse shape, referring to [Figure 3-102](#) and the instructions in [Entering Segment Arb™ values into UTM array parameters](#). The number of non-blank entries in the array must match NumPulseTerminals, as shown in [Figure 3-104](#), NumPulseTerminals = 4, and the size each array (PulseVoltagesSize, PrePulseDelays, TransitionTimesSize, PulseWidthsSize, PostPulseDelaysSize) are also four.
  - a) PulseVoltages: Use a positive value for a waveform similar to [Figure 3-102](#). If a negative pulse is required, use a negative voltage value. To put a pulse channel into a disconnected, or high impedance, state, use -999.
  - b) PrePulseDelays, TransitionTimes, PulseWidths, PostPulseDelays: The minimum time is 20 E-9 (20 ns). Number 0 (zero) is not a valid input value. The maximum time is 1 s.
4. Enter the number of pulses into NumPulses. This parameter determines the number of pulses that will be output each time the test is run.



5. Enter the number of SMUs that are used as Bias Terminals into NumSMUBiasTerminals. An example of using an SMU as a bias terminal is shown in [Figure 3-96](#). The 4<sup>th</sup> SMU in [Figure 3-96](#) is a dedicated connection to a bit line on the array DUT. During a pulse test, such as Program or Erase, this SMU would output a DC voltage that would provide power to the drain terminal of the first column of the array.
6. Enter the SMU IDs for the SMU(s) used as a bias into SMUBiasTerminals. For the configuration in [Figure 3-96](#), SMUBiasTerminals = SMU4.
7. Enter the voltages in the array SMUBiasVoltages. These are the voltages for the SMUs listed in SMUBiasTerminals. The number of non-blank entries in the array must match NumSMUTerminals.
8. Enter the number of SMUs that are sharing a cable with a pulse channel into NumSharedSMUs. Sharing means that one pulse and one SMU signal are combined to a single DUT terminal. [Figure 3-96](#) shows that three pairs of SMU/pulse channels are shared. Note the SMA tees on each of the top three SMUs that incorporate both a pulse channel and a SMU signal into a single cable to a DUT terminal. Supplying the shared SMU information allows the software to open the SMU relay during the pulse output, that is necessary to permit good pulse fidelity. If a switch matrix is used in the configuration (see [Figure 3-97](#)), then use NumSharedSMUs = 0.
9. Enter the SMU IDs for the SMU(s) sharing a cable with a pulse channel into SharedSMUs. For the configuration in [Figure 3-96](#), SharedSMUs = SMU1,SMU2,SMU3. There are no spaces allowed in the SharedSMUs string.
10. Press the green triangle Run button to output the pulses.
11. Check the Data tab in the Sheet control. The single\_pulse\_flash value should be 0, indicating that there were no errors. No measurements are taken in this test, so there is no data to graph.
12. If single\_pulse\_flash is non-zero, pulses are not being output, or there are error messages in the Project Messages pane, see [Troubleshooting](#) section.

Table 3-29

**Parameter values for Program or Erase UTM for 4 or 2 channel configurations**

Parameter	Value for 4 channel test				Value for 2 channel test	
NumPulseTerminals	4				2	
PulseTerminals	VPU1CH1, VPU1CH2, VPU2CH1, VPU2CH2				VPU1CH1, VPU1CH2	
PulseVoltages	0	7	0	0	0	7
PrePulseDelays	1 E-6	1 E-6	1 E-6	1 E-6	1 E-6	1 E-6
TransitionTimes	3 E-7	3 E-7	3 E-7	3 E-7	3 E-7	3 E-7
PulseWidths	5 E-6	5 E-6	5 E-6	5 E-6	5 E-6	5 E-6
PostPulseDelays	2 E-6	2 E-6	2 E-6	2 E-6	2 E-6	2 E-6
NumPulses	1				1	
NumSMUBiasTerminals	0				0	
SMUBiasTerminals						
SMUBiasVoltages						
NumSharedSMUs	4				2	
SharedSMUs	SMU1,SMU2,SMU3,SMU4				SMU1,SMU2	

**NOTE** Channel count refers to the number of pulse+SMU channels with a direct connect setup. A setup with four channels of each pulse and SMU is in [Figure 3-95](#). All channels in group must have the same total time.



## Running the Fast-Program-Erase UTM

This test is used in all of the Flash projects. It outputs a number of pulse waveforms (Parameter = NumPulses) with a shape shown similar to [Figure 3-103](#), on a number of pulse channels (Parameters = NumPulseTerminals, PulseTerminals).

1. Enter the number of pulse waveforms required into NumPulseTerminals. This must be a minimum of one channel, up to a maximum of the number of channels available. For a typical Flash system with two 4205-PG2 cards, there are four pulse channels available.
2. Enter which pulse channels will be used into PulseTerminals. This is a string of channels, in the form VPU1CH1,VPU1CH2,VPU2CH1,VPU2CH2. VPU1 is the 4205-PG2 in the lowest-numbered slot (right-most slot when looking at back of Model 4200-SCS chassis). The characters are all capitalized and each channel is separated by a comma. No spaces are allowed in the PulseTerminal string.
3. Enter the values in the ten arrays that define the pulse shape, referring to [Figure 3-103](#) and the instructions in [Entering Segment Arb™ values into UTM array parameters](#). The number of non-blank entries in the array must match NumPulseTerminals, as shown in [Figure 3-104](#), NumPulseTerminals = 4, and the size each array (Pulse1VoltagesSize, PrePulse1Delays, TransitionTimesPulse1Size, Pulse1WidthsSize, PostPulse1DelaysSize, Pulse2VoltagesSize, PrePulse2Delays, TransitionTimesPulse2Size, Pulse2WidthsSize, PostPulse2DelaysSize) are also 4.
  - a. PulseVoltages: Use a positive value for a waveform similar to [Figure 3-103](#). If a negative pulse is required, use a negative voltage value. To put a pulse channel in to a disconnected, or high impedance, state, use -999.
  - b. PrePulseDelays, TransitionTimes, PulseWidths, PostPulseDelays: The minimum time is 20 E-9 (20 ns). Number 0 (zero) is not a valid input value. The maximum time is 1 s.
4. Enter the number of pulses into NumPulses. This parameter determines the number of program and erase pulse waveforms that will be output each time the test is run.
5. Enter the number of SMUs that are used as Bias Terminals into NumSMUBiasTerminals. An example of using an SMU as a bias terminal is shown in [Figure 3-96](#). The 4<sup>th</sup> SMU in [Figure 3-96](#) is a dedicated connection to a bit line on the array DUT. During a pulse test, such as Program or Erase, this SMU would output a DC voltage that would provide power to the drain terminal of the first column of the array.
6. Enter the SMU IDs for the SMU(s) used as a bias into SMUBiasTerminals. For the configuration in [Figure 3-96](#), SMUBiasTerminals = SMU4.
7. Enter the voltages in the array SMUBiasVoltages. These are the voltages for the SMUs listed in SMUBiasTerminals. The number of non-blank entries in the array must match NumSMUTerminals.
8. Enter the number of SMUs that are sharing a cable with a pulse channel into NumSharedSMUs. Sharing means that one pulse and one SMU signal are combined to a single DUT terminal. [Figure 3-96](#) shows that three pairs of SMU/pulse channels are shared.

---

**NOTE** *The SMA tees on each of the top three SMUs that incorporate both a pulse channel and a SMU signal into a single cable to a DUT terminal. Supplying the shared SMU information allows the software to open the SMU relay during the pulse output, that is necessary to permit good pulse fidelity. If a switch matrix is used in the configuration (see [Figure 3-97](#)), then use NumSharedSMUs = 0.*

---

9. Enter the SMU IDs for the SMU(s) sharing a cable with a pulse channel into SharedSMUs. For the configuration in [Figure 3-96](#), SharedSMUs = SMU1,SMU2,SMU3. There are no spaces allowed in the SharedSMUs string.
10. Press the green triangle Run button to output the pulses.



11. Check the Data tab in the Sheet control. The double\_pulse\_flash value should be 0, indicating that there were no errors. No measurements are taken in this test, so there is no data to graph.
12. If double\_pulse\_flash is non-zero, pulses are not being output, or there are error messages in the Project Messages pane, see [Troubleshooting](#) section.

## Running the SetupDC UTM

This UTM disconnects the PG2 channels by opening solid state relays. This is necessary when using the direct connect method (see [Figure 3-95](#) and [Figure 3-96](#)), to ensure that a subsequent SMU test is only connected to the DUT terminals:

1. Enter the number of shared terminals into SharedPulseTerminals. Sharing means that a single cable provides either a pulse or a SMU signal to a DUT terminal.
2. Enter the Pulse channel IDs for the VPU channels sharing a cable with a SMU into SharedPulseTerminals. For the configuration in [Figure 3-96](#), SharedPulseTerminals = VPU1CH1,VPU1CH2,VPU2CH1VPU2CH2. There are no spaces allowed in the string.
3. Check the Data tab on the Sheet tab, configure\_dc\_flash = 0. If the value is non-zero, or there are error messages in the Project Messages pane, see [Troubleshooting](#) section.

## Running the Vt-MaxGm ITM

This test performs a DC  $V_G$ - $I_D$  sweep on the DUT and extracts the threshold voltage ( $V_T$ ). The VT results for the first run are shown on the graph, in the lower left corner.  $V_T$  values for each test (run or append) is shown in each tab, in the right-most column headed VT. This test can be run after setting the device in any state, using the Program, Erase and/or Fast-Program-Erase UTMs. This test provides the  $V_T$ , but does not determine an appropriate, or target,  $V_T$ , that is usually provided by historical performance, a review of the device structure, or the device engineer.

1. Enter the voltage values for each SMU. Defaults have a voltage sweep on the gate, a fixed DC bias on the drain, and 0 V or a GNDU signal for the source and bulk.
2. Once the test is run, review the graph or sheet results.

## Running the ConPin-Pulse or ConPin-DC UTM (Switch projects only)

This test routes the desired pulse or SMU signals to the DUT by closing switches on a switch matrix card. See [Switch matrix connection to array DUT](#) for connection and switch matrix setup instructions. The UTM entries for TermIdStr1-TermIdStr8 Pin1-Pin8 define which instrument (SMU or VPU channel) get connected to which output pin.

1. Enter a value for OpenAll. The default value is 1, that opens all switches. The remaining parameters define which switches to close. If more than 8 closures are required for a test, use two ConPin tests, setting the second ConPin test OpenAll = 0, to ensure the first ConPin switch settings are not cleared. See [Figure 3-110](#) for the screenshot of the ConPin parameters.
2. Enter values for TermIdStr1 and Pin1. This first pair of parameters determines which instrument, either SMU or VPU channel, gets output. For a SMU to be output on the 1<sup>st</sup> output of the switch matrix, TermIdStr1 = SMU1 and Pin1 = 1. If no connection is desired set Pin1 = 0. Another way to set the connections is to use the GUI. To use the GUI, click the GUI button (see [Figure 3-110](#)) that displays the dialog shown in [Figure 3-111](#). Click the desired crosspoint closures. The Input strings shown in the GUI must match the labels supplied in the KCON setup for the switch matrix card (see [Figure 3-99](#)).
3. Continue to enter values for the remaining 7 pairs of TermIdStr and Pins.
4. Running the test will set the switch closures as specified.



There are three similar projects that stress the DUT with a number of Program+Erase waveforms, then periodically measures the  $V_T$  :

- FlashEndurance-NAND project
- FlashEndurance-NOR project
- FlashEndurance-switch project

The purpose of these projects is to determine the lifetime of the DUT, based on the number of Program+Erase cycles withstood by the device before a certain amount of shift, or degradation, in the  $V_T$  or other measurement. The waveforms may be unique for each pulse channel, and are defined in the separate Kpulse program and saved to files. For more information refer to [Reference Manual, KPulse \(for Keithley Pulse Generator Cards\), page 13-1](#) and [Using KPulse to create and export Segment Arb™ waveforms](#).

These files are specified for each pulse channel in the test. The number and interval for the pulse stresses are set, as well as the desired SMU measurements. The typical measurement is a  $V_T$  extraction based on a  $V_G$ - $I_D$  sweep, but any type of DC test may be configured.

These projects support from one to eight pulse channels to support typical 4-terminal devices, as well as higher pin count devices or array test structures. The 8-terminal testing requires four Model 4205-PG2 cards and, for most tests, a compatible external switch matrix.

The difference between the FlashEndurance-NAND and FlashEndurance-NOR are the difference in the typical pulse widths and levels specific to the DUT type. The FlashEndurance-Switch project is a generic example of the Flash testing described above, but adds support for an external Keithley switch matrix. Example results for the Endurance tests are shown in [Figure 3-113](#) and [Figure 3-90](#).

## FlashEndurance-NAND tests

FlashEndurance-NAND tests consist of the following test:

- Program
- SetupDC-Program
- $V_T$ -MaxGm-Program
- Erase
- SetupDC-Erase
- $V_T$ -MaxGm-Erase

The project navigator for FlashEndurance-NAND is shown in [Figure 3-112](#). Stressing for the FlashEndurance-NAND tests are configured from the Subsite Setup tab for the FlashEndurance subsite plan.

The default setup (shown in [Figure 3-113](#) and [Figure 3-114](#)) uses Segment Arb™ waveforms to perform log stressing that ranges from 1 to 100,000 counts.

The Segment Arb™ waveform files (Flash-NAND-Vg-ksf and Flash-NAND-Vd-ksf) used for stressing are loaded into the Device Stress Properties window shown in [Figure 3-114](#). The stress properties window is opened by clicking the **Device Stress Properties** button in [Figure 3-113](#). Example results for the Endurance tests are shown in Subsite Graph tab (see [Figure 3-115](#)).



Figure 3-112  
**FlashEndurance-NAND project plan**

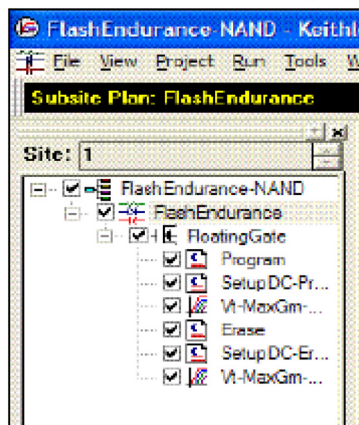


Figure 3-113  
**FlashEndurance-NAND project – Subsite Plan tab**

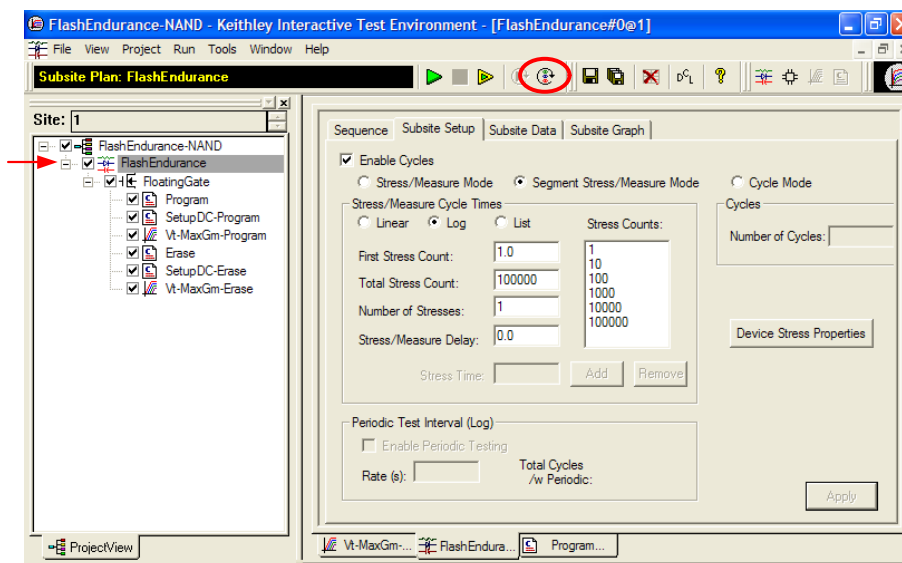




Figure 3-114  
FlashEndurance-NAND project – Device Stress Properties

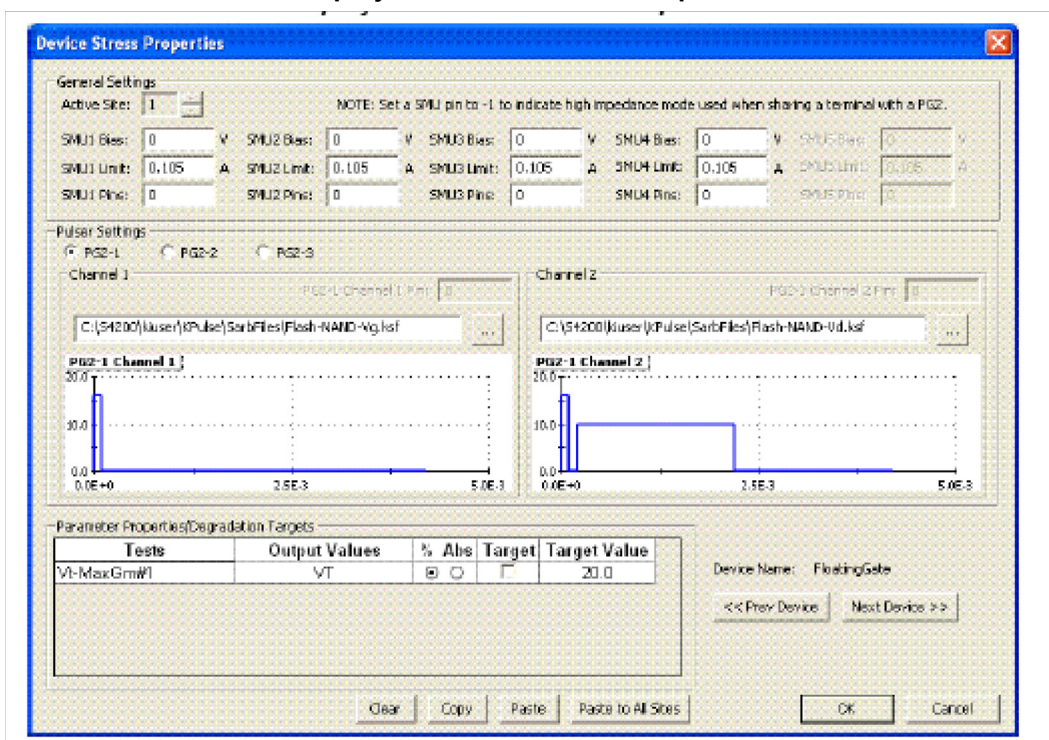
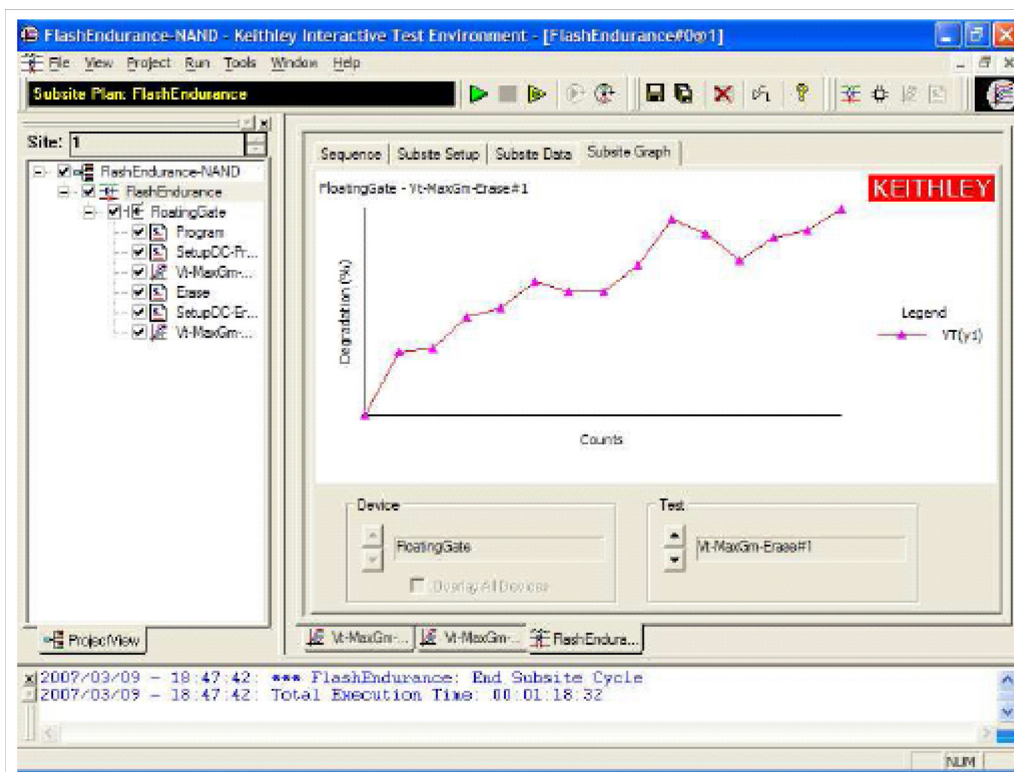


Figure 3-115  
FlashEndurance-NAND project – Subsite Graph tab

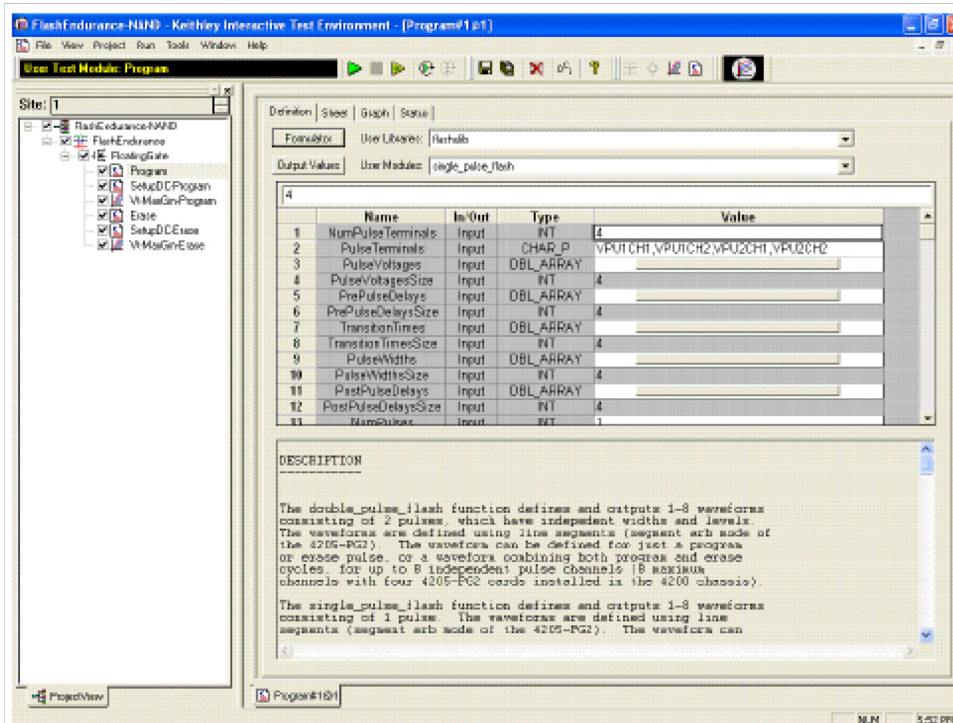




**Program test** – The Definition tab for this test is shown in [Figure 3-116](#). This test uses a partially predefined Segment Arb™ waveforms, see [Figure 3-102](#), to program a flash memory device and identical to the Program UTMs included in the other Flash projects.

Figure 3-116

#### FlashEndurance-NAND project – Program Definition tab



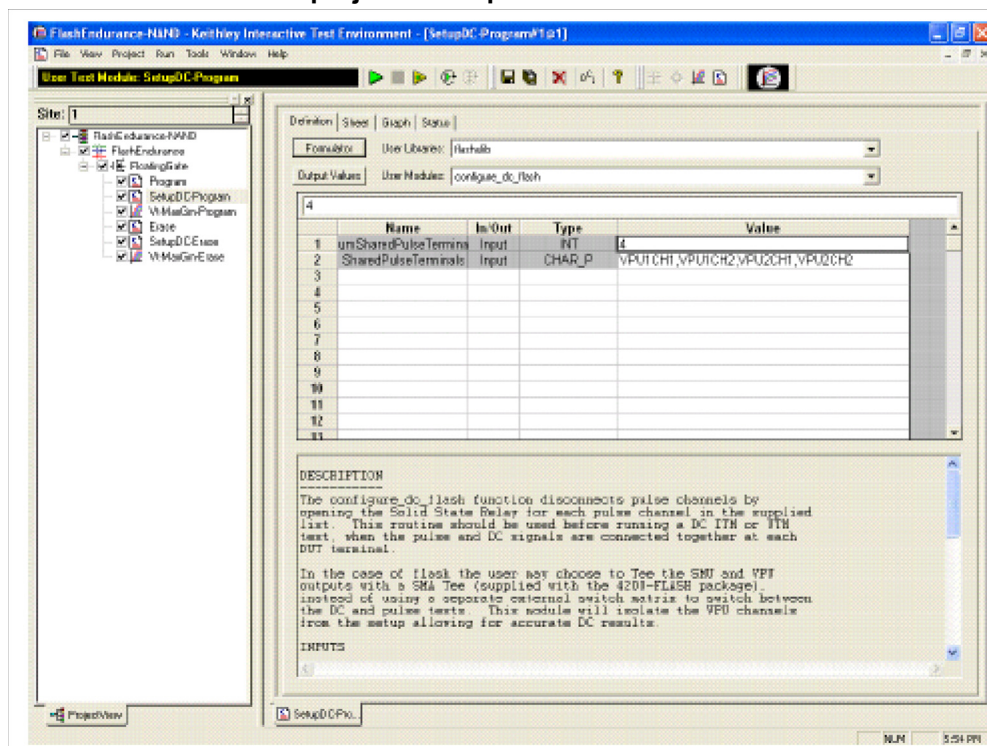
**SetupDC-Program test** – The Definition tab for this test is shown in [Figure 3-117](#). This test isolates the VPU outputs from the DUT. It does this by opening the HEOR for each VPU channel. Disconnecting the VPU channels allows for accurate DC results.

The SetupDC test is a UTM that should be used when using a directly wired DUT, without an external switch matrix. SetupDC disconnects the PG2 channels from the DUT to permit proper operation of any subsequent DC measurements.

When using a switch matrix, a ConPin test is used (see the [Reference Manual, LPT functions, page 8-23](#)) to set the appropriate matrix connections prior to any DC tests.



Figure 3-117  
FlashEndurance-NAND project – SetupDC Definition tab



**Vt-MaxGm-Program test** – This test is used to perform a DC voltage sweep on the gate of the DUT and measure the drain current at each sweep step. The default Definition tab for this test is shown in Figure 3-118. SMU3 is configured to perform a 101 point sweep from 0 to 5 V in 50 mV steps.

SMU1 is configured to DC bias the drain at 0.5 V and measure current at each step of the sweep. The results of the test are shown in the Graph tab (see Figure 3-119). The Vt-MaxGm tests may be replaced with another Vt or DC test. Or, additional DC tests may be added after this test.

Figure 3-118  
FlashEndurance-NAND project – Vt-MaxGm-Program Definition tab

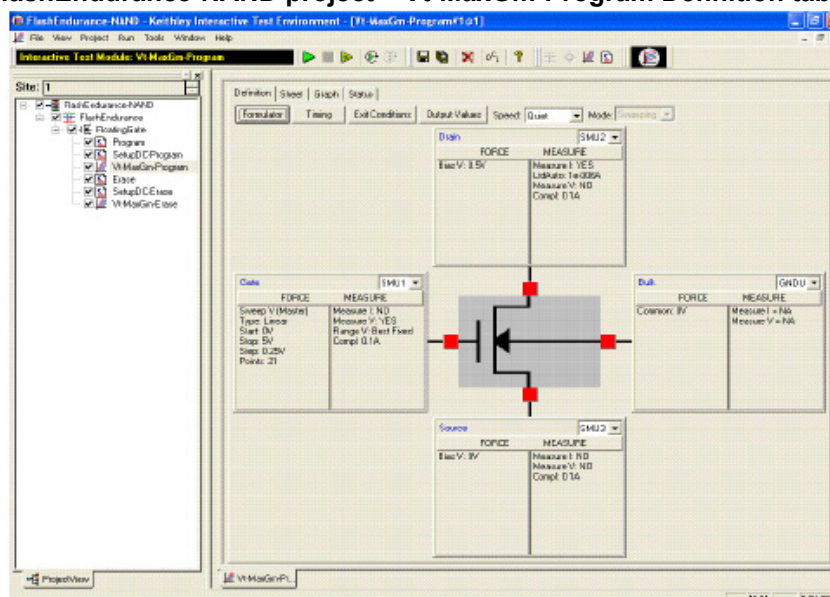
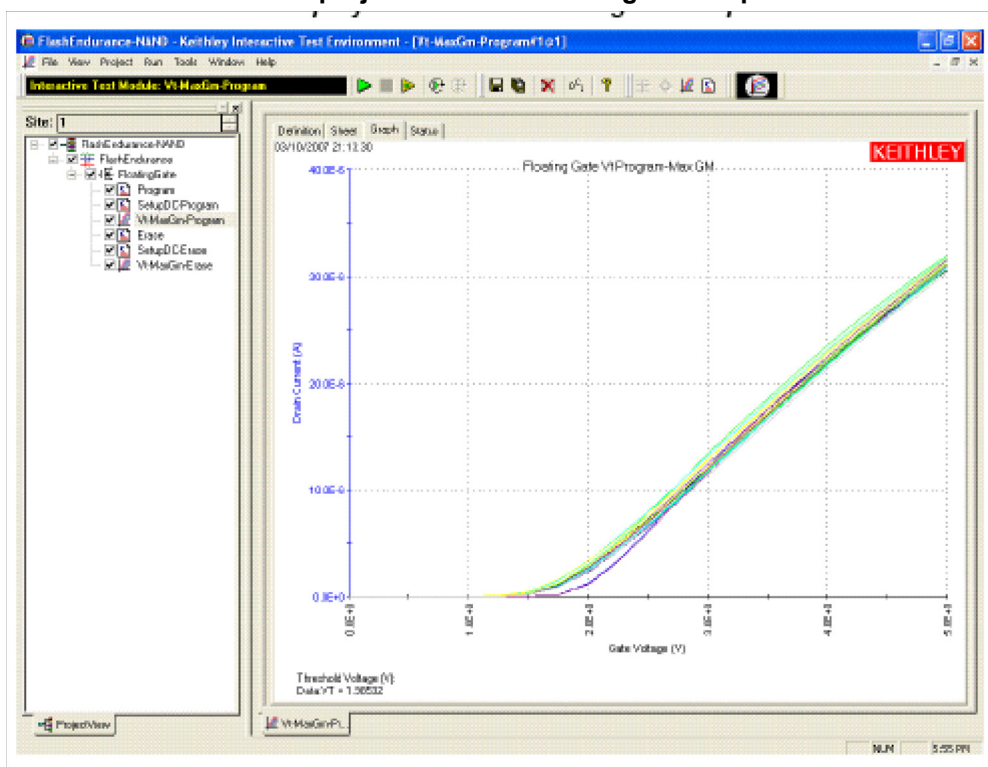




Figure 3-119  
FlashEndurance-NAND project – Vt-MaxGm-Program Graph tab

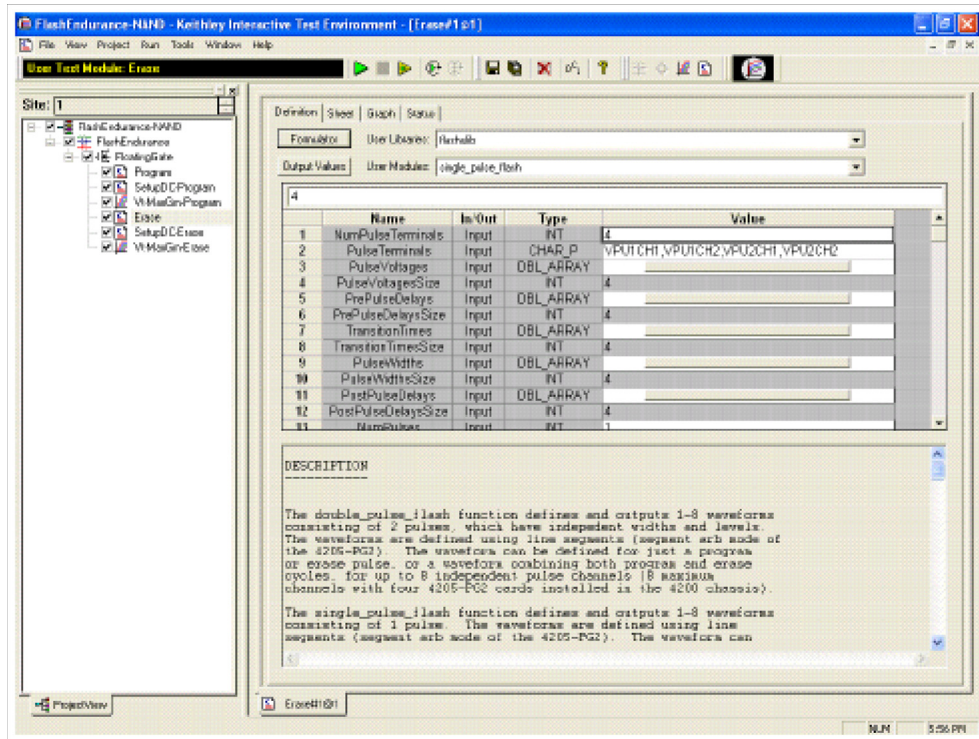




**Erase test** – This test uses Segment Arb™ waveforms to program a flash memory device. The default Definition tab for this test is shown in [Figure 3-120](#).

Figure 3-120

### FlashEndurance-NAND project – Erase Definition tab



**SetupDC-Erase test** – This test isolates the VPU outputs from the DUT. It does this by opening the HEOR for each VPU channel. Disconnecting the VPU channels allows for accurate DC results.

**Vt-MaxGm-Erase test** – This test is used to perform a DC voltage sweep on the gate of the DUT and measure the drain current at each sweep step. SMU3 is configured to perform a 101 point sweep from 0 to 5 V in 50 mV steps. SMU1 is configured to DC bias the drain at 0.5 V and measure current at each step of the sweep.

The Vt-MaxGm tests may be replaced with another Vt or DC test. Or, additional DC tests may be added after this test.

### FlashEndurance-NOR tests

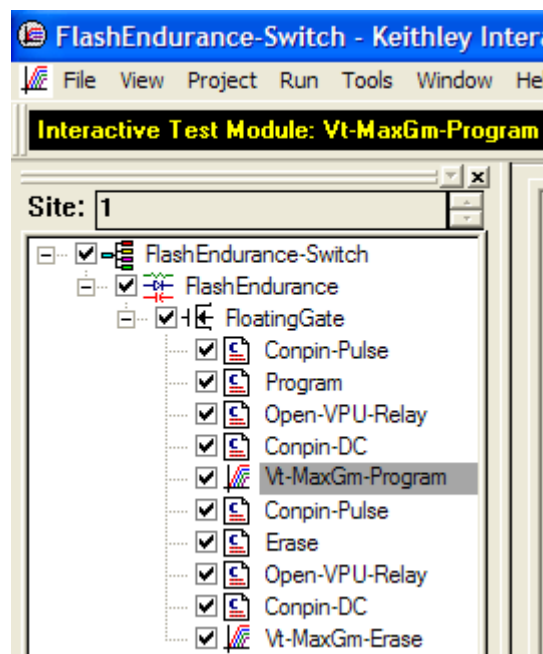
The FlashEndurance-NOR project has tests similar to the FlashEndurance-NAND project, with defaults for NOR type floating gate DUTs.

### FlashEndurance-switch tests

The FlashEndurance-switch project, [Figure 3-121](#), has similar tests to the FlashEndurance-NAND, with defaults for using a switch matrix for more complex multi-DUT addressable test structures (see [Figure 3-97](#)). Also note the additional test, Open-VPU-Relay, added prior to Conpin-DC tests. This step ensures that VPU channels will not be inadvertently connected to a device terminal when the SMU testing is performed.



Figure 3-121

**FlashEndurance-Switch project****Running a FlashEndurance or FlashDisturb project**

This section explains how to use the following Flash projects:

- FlashEndurance-NAND
- FlashEndurance-NOR
- FlashEndurance-Switch
- FlashDisturb-NAND
- FlashDisturb-NOR
- FlashDisturb-Switch

These Flash projects use a small number of tests and methods. This section will explain the tests and how to set parameter values for [Endurance testing](#) or [Disturb testing](#).

Before using any of these projects, determine the appropriate pulse voltages and widths by first using the appropriate project, using the procedures in [Running the Flash-NAND, Flash-NOR or Flash-Switch Project](#).

The Endurance and Disturb projects include everything from the corresponding Flash-NAND, Flash-NOR, or Flash-Switch projects. To use an Endurance or Disturb test, ensure that each test in the project navigator is functioning properly by following the procedures below. After setting up all of the tests, the information below will continue to explain the setup for the stress / measure looping that is the core of any endurance or disturb test.

1. If system connections have not been made, follow the instruction in [Running any Flash Project for the first time](#).
2. If KITE is not running, start KITE by double-clicking the KITE icon on the Model 4200-SCS desktop.
3. Open the appropriate KITE Flash project.
  - a. Within KITE, click FILE > Open Project. If the dialog window is not displaying the \_Memory folder, move up one or two levels to the display the Projects directory. Double-click the \_Memory folder, then double-click the desired Flash test folder (see list above



- this procedure), then double-click the appropriate \*.kpr file to open the desired Model 4200-SCS Flash project.
- b. KITE should resemble [Figure 3-116](#).
  4. Touch-down or connect the DUT.
  5. Verify setup and connection by running Vt-MaxGm test
    - a. Set appropriate voltages
    - b. Run the test by clicking the yellow and green triangle Append button.
    - c. Ensure that the  $V_{G-ID}$  and  $V_T$  results are reasonable.
  6. Determine the appropriate pulse voltage levels:
    - a. Review the section [Pulse waveforms for NVM testing on page 3-102](#).
    - b. Recall that pulse voltage levels on the gate will double:
      - i. For example, using PulseVoltage = 2 will result in  $V_G = 4$  V for a typical high-impedance (1 k  $\Omega$ ) terminal.
    - c. Use oscilloscope to determine appropriate PulseVoltage values for the desired  $V_G$  and  $V_D$ , making sure to use the 1 M  $\Omega$  input setting on the oscilloscope:
      - i. Drain: Connect the oscilloscope probe across the drain-source of the DUT.
      - ii. Use the Program and Erase UTMs to output pulses, while using the oscilloscope to measure the pulse height. Iterate by modifying the PulseVoltages to reach the target voltage.
  7. Once the appropriate voltage level is determined, the appropriate pulse width may be determined by iteratively outputting pulses while occasionally measuring the  $V_T$ .
    - a. Start by using a pulse width that is shorter than the expected PW. For example, if 20  $\mu$ s is the expected PW, try using a 2  $\mu$ s PW.
    - b. Enter the parameter values into the Program UTM, following the procedure in [Running the Program or Erase UTM](#). Initially, set NumPulses = 2, or another small number.
    - c. Uncheck the Erase and Fast-Program-Erase tests.
    - d. Enter the parameter values for the Vt-MaxGm test, following the procedure in [Running the Vt-MaxGm ITM](#).
    - e. Run the test:
      - i. Double-click 4Terminal-FloatingGate
      - ii. Click the **Run** button
      - iii. Check the graph on the Vt-MaxGm test. It will likely be too low on the first few runs, but note the total number of pulses sent to the DUT.
      - iv. Rerun test by following the previous Run the Test steps. Once the  $V_T$  has met the target value, note the total pulse width to use to program the device, using either the Program or Fast-Program-Erase tests.
    - f. Repeat above with the Erase test, feeding final results into the Erase and Fast-Program-Erase tests.
    - g. Ensure that the erase parameters are fully erasing the DUT
      - i. Set the parameters in the Fast-Program-Erase test. Set NumPulses = 10.
      - ii. Uncheck the Program and Erase tests.
      - iii. Double-click 4Terminal-FloatingGate. Press run.
      - iv. Note the  $V_T$ .
      - v. Change NumPulses = 100 or a larger number.
      - vi. Double-click 4Terminal-FloatingGate. Press Append.
      - vii. Note the  $V_T$ . If the  $V_T$  value for the tests are similar, then the erase pulse is fully erasing the DUT.

The following link to procedures for these UTM and ITM tests:

[Running the Program or Erase UTM](#)

[Running the Fast-Program-Erase UTM](#)

[Running the SetupDC UTM](#)



### Running the Vt-MaxGm ITM

### Running the ConPin-Pulse or ConPin-DC UTM (Switch projects only)

## Running endurance or disturb looping

The Endurance or Disturb testing is essentially a stress / measure test. The stress portion applied a number of pulse waveforms to the DUT, then periodically measurements are performed.

1. Double-click FlashSubsite (see red arrow in [Figure 3-113](#)), then click the Subsite Setup tab. The screen should resemble [Figure 3-113](#).
2. Ensure that the Segment Stress / Measure Mode is chosen.
3. Determine the stress intervals and how often the measurements are performed. Each entry in the Stress Counts box is the number of waveforms that will be output. After the listed number of waveform counts is output, measurements are performed. All checked boxes in the project navigator will be run after each stress interval.
  - a. Choose Linear, Log or List
  - b. Enter the First stress count, that must be at least 1.
  - c. Enter the Total Stress Count, that is the last stress interval output.
  - d. Number of stresses is the number of stress intervals.
    - i. For linear, the number of total stress intervals
    - ii. For log, the number of stress intervals per decade count of stress counts
  - e. Press the Apply button to see the updated Stress Counts and intervals.
4. Click the Device Stress Properties button, that will display something similar to [Figure 3-114](#).
5. The General Settings show the SMU settings during the stress portion of the test. These settings are necessary when using an array DUT structure, either direct connect (see [Figure 3-96](#)) or using a switch matrix (see [Figure 3-97](#)).
  - a. If a SMU bias is required, then set the voltage and current limit.
  - b. If using shared cabling, or SMU/VPU pairings, set all Pins entries = -1, to disconnect the SMUs during the stress, allowing the pulse signals to properly reach the DUT terminals.
  - c. If using a switch matrix, set the pin connection. If no connection is required, input 0.
6. Pulse generator settings configure the waveforms used during the stress. See [Using Kpulse to create and export Segment Arb™ waveforms](#) to create the desired multi-level pulse waveforms.
  - a. Click the ... button for each available pulse channel. Choose the desired waveform, previously created and exported, from the available list. Each channel must have an associated .ksf waveform and each waveform should have the same duration.
7. Parameter properties show which parameters are graphed in the Subsite graph, and if any test should end early. If a test should end after a certain VT shift, either an absolute voltage shift, or a % shift follow the below:
  - a. Choose % or Abs
  - b. Check the box
  - c. Enter the Target value.
8. Before running the test for the first time, it is recommended to try out the project on a scrap device:
  - a) Ensure that the project navigator is showing the FlashEndurance entry highlighted, as shown in [Figure 3-113](#).
  - b) Click the **Run Test/Subsite** button (in the red oval on [Figure 3-113](#)).
  - c) Ensure that the test cycles through each test in the project navigator, and that data is input into the Subsite Data tab.
9. Move to a fresh device and click the **Run Test/Subsite** icon.



10. If errors or unexpected operation occurs, see the [Error codes](#) and [Troubleshooting](#) sections.

The following three projects are similar, and use the shared stress-measure looping capability of the FlashEndurance projects:

- FlashDisturb-NAND project
- FlashDisturb-NOR project
- FlashDisturb-switch project

The purpose of the Disturb test is to pulse stress a device in an array test structure, then perform a measurement, such as  $V_T$ , on a device adjacent to the pulsed device.

The goal is to measure the amount of  $V_T$  shift in adjacent cells, either in the programmed or erased states, when a nearby device is pulsed with either a Program, Erase, or Program+Erase waveforms.

The typical measurement is a  $V_T$  extraction based on a  $V_g$ -Id sweep, but any type of DC test may be configured.

The difference between the FlashDisturb-NAND and FlashDisturb-NOR are the typical pulse widths and levels specific to the DUT type. The FlashDisturb-switch is a generic example of the Flash testing described above, but adds support for an external Keithley switch matrix.

## FlashDisturb tests

The FlashDisturb tests consists of the following tests:

- Program
- SetupDC-Program
- Vt-MaxGm-Program
- Erase
- SetupDC-Erase
- Vt-MaxGm-Erase

The six tests listed above are the same ones used for endurance testing (see [FlashEndurance-NAND tests](#) for details).

Stressing for the disturb tests are configured from the subsite setup tab for a disturb project subsite plan. The default subsite setup for FlashDisturb-NAND (shown in [Figure 3-122](#)) uses Segment Arb™ waveforms, defined and saved to file using Kpulse, to perform log stressing that ranges from 100,000 to 1,000,000 counts.

The Segment Arb™ waveform files (Flash-NAND-Vg.ksf and Flash-NAND-Vd.ksf) used for stressing are loaded into the device stress properties window shown in [Figure 3-123](#). The stress properties window is opened by clicking the **Device Stress Properties** button in [Figure 3-122](#).



Figure 3-122  
FlashDisturb-NAND project – subsite setup tab

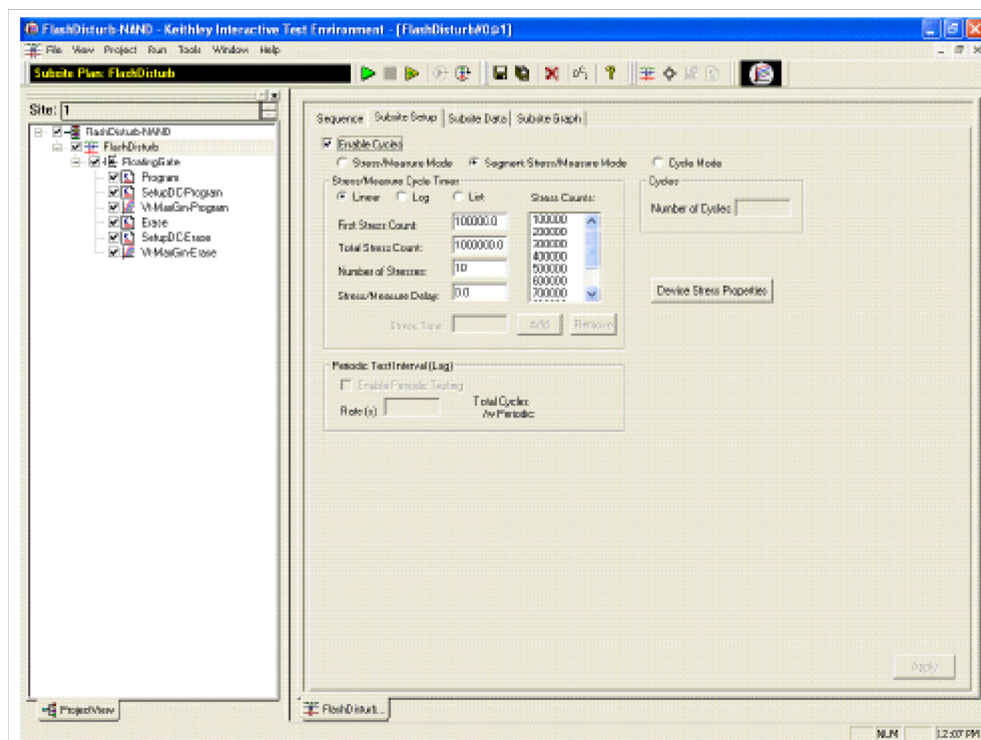
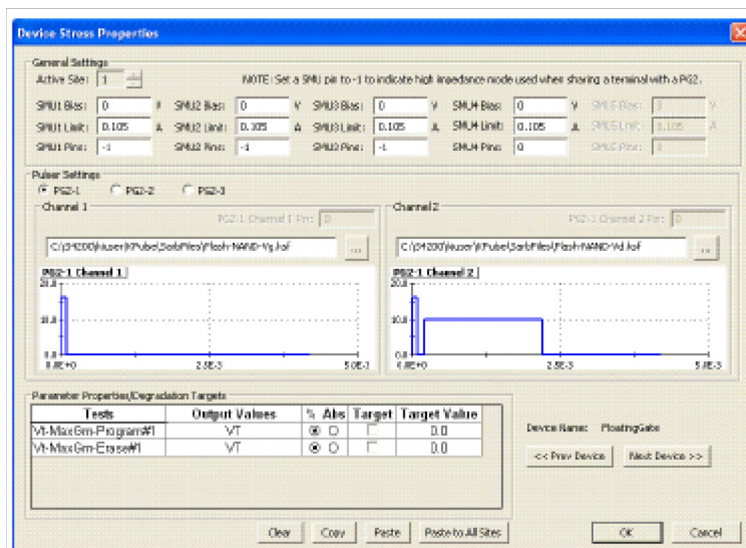


Figure 3-123  
FlashDisturb-NAND project – device stress properties



## Explanation of flash UTM parameters

NumPulseTerminals

(int) The number of pulse terminals, or pulse channels, to use for the test. The number of pulse terminals ranges from one to eight.

PulseTerminals

(char \*) A string representation of all the VPU channels being used in the test, matching the number given in NumPulse. For example if the setup is such that VPU1 Channel 1 and VPU2



	Channel 2 are being used, then PulseTerminals should look like this: VPU1CH1,VPU2CH2. There are no spaces in this list of channels.
Pulse1Voltages	(double) Array of voltage values for the pulse height (0 V referenced) of first pulse on each pulse channel. Valid values range from -20 V to +20 V. All voltage levels assume a 50 $\Omega$ load. In order to float a channel (disconnect pulse output from a DUT pin), using the Solid State Relay, use -999. Minimum time required for a SSR open or close is 100 $\mu$ s.
PrePulse1Delays	(double) Array of time values used as a delay before the first pulse is output. Valid values range from 20 ns to 1 s in 10 ns increments (s).
TransitionTimesPulse1	(double) The amount of time it will take the first pulse to rise/fall (0-100%/100-0%) from the BaseValue (0 V) to the given Pulse Voltage. If the pulse voltage level is from -5 to +5 V, then the valid transition times are from 20 ns to 33 ms in 10 ns increments, else if pulse voltage is within -20 to +20 V, then valid values range from 100 ns to 33 ms in 10 ns increments (s).
Pulse1Widths	(double) Array of values defining the pulse widths for the first pulse of each channel. Minimum values are 20 ns to 1 s. Pulse width is defined as FWHM, so it includes half of the fall time and half of the rise time (transition time), in seconds.
PostPulse1Delays	(double) Array of time values used as a delay after the first pulse is output (that is, time at the 0 V base voltage). Valid values are 20 ns to 1 s in 10 ns increments (s).
Pulse2Voltages	(double) Array of voltage values for the pulse height (0 V referenced) of second pulse on each pulse channel. Valid values range from -20 V to +20 V. All tests assume a 50 $\Omega$ load. In order to float a channel, or disconnect from a DUT pin, using the Solid State Relay, use -999. Minimum time required for a SSR open or close is 100 $\mu$ s.
PrePulse2Delays	(double) Array of time values used as a delay before the second pulse is output. This delay happens after the PostPulse1Delays. Valid values range from 20 ns to 1 s in 10 ns increments (s).
TransitionTimesPulse2	(double) The amount of time it will take the second pulse to rise/fall (0-100%/100-0%) from the BaseValue (0 V) to the given Pulse Voltage. If the pulse voltage level is from -5 to +5 V, then the valid transition times are from 20 ns to 33 ms in 10 ns increments, else if pulse voltage is within -20 to +20 V, then valid values range from 100 ns to 33 ms in 10 ns increments (s).
Pulse2Widths	(double) Array of values defining the pulse widths for the second pulse in each channel. Minimum values are 20 ns to 1s. Pulse width is defined as FWHM, so it includes half of the fall time and half of the rise time (transition time), in seconds.
PostPulse2Delays	(double) Array of time values used as a delay after the second pulse is output (that is, time at the 0 V base voltage). Valid values are 20 ns to 1 s in 10 ns increments (s).
NumPulses	(int) The number of pulses to output. Valid range: 1 to (2 <sup>31</sup> ) (about 4.2 billion).



NumSMUBiasTerminals	(int) The number of bias SMUs to include in the test. These are SMUs that are not connected in the SMU+Pulse sharing configuration, but additional SMUs that could be used for biasing word or bit lines.
SMUBiasTerminals	(char *) A string representation of all the SMU channels being used in the test. For example, if the setup is such that SMU1 and SMU2 are being used for a bias, then SMUBiasTerminals would be: SMU1,SMU2.
SMUBiasVoltages	(double) Array of SMU bias values used during the test. The values correspond to the number and order in the SMUBiasTerminals string.
NumSharedSMUs	(int) The number of SMUs sharing a connection to the device with a VPU. Sharing a connection means using a Tee to combine a SMU and VPU channel for a DUT terminal, instead of using an external switch matrix. This variable stores the number of shared SMU+Pulse instances.
SharedSMUs	(char *) A string representation of all the shared SMU channels being used in the test. This string is used to disconnect each SMU from the shared cable during pulse output. For example, if SMU1 is connected through a Tee to a pulse channel and SMU2 is also connected to another pulse channel, then the SharedSMUs string would be: SMU1, SMU2.
SharedPulseTerminals	(int) Number of Pulse channels that are paired with an SMU. This parameter is used in conjunction with SharedPulseTerminals. See <a href="#">Figure 3-89</a> and <a href="#">Figure 3-95</a> for examples of a SMU and VPU sharing a cable to a DUT terminal.
SharedPulseTerminals	(char *) A list of pulse channels that each share a cable with a SMU. The list for two channels on the lowest numbered VPU would be: VPU1CH1,VPUCH2. There are no spaces or quotation marks in the string. See <a href="#">Figure 3-89</a> and <a href="#">Figure 3-95</a> for examples of a SMU and VPU sharing a cable to a DUT terminal.
OpenAll	(int) Value for ConPin test that determines if all matrix switch points are opened, before the desired switch point closures. Using OpenAll = 1 essential resets the switch to an all open state, then the desired switches are closed.

## Error codes

0	No Errors
-16001	Invalid number of pulse terminals
-16002	PulseVoltagesSize has to match the number of pulse terminals
-16003	PrePulseDelaysSize has to match the number of pulse terminals
-16004	TransitionTimesSize has to match the number of pulse terminals
-16005	PulseWidthsSize has to match the number of pulse terminals
-16006	PostPulseDelaysSize has to match the number of pulse terminals
-16007	Invalid number of bias SMU terminals
-16008	Invalid name of shared pulse terminal(s)



-16009	Required bias SMU is not available in current configuration
-16010	Required shared SMU is not available in current configuration
-16011	Required VPU is not available in current configuration
-16012	PrePulseDelay value is out of valid range
-16013	PrePulseDelay value has to be in 10 ns increments
-16014	TransitionTime value is out of valid range
-16015	TransitionTime value has to be in 10 ns increments
-16016	Pulse level value is out of valid range
-16017	Pulse width value is out of valid range
-16018	PostPulseDelay value is out of valid range
-16019	PostPulseDelay value has to be in 10 ns increments

## Troubleshooting

Check the [Error codes](#) for additional information.

### No pulse output

If pulses are not being output, please check the following:

1. Ensure proper cabling. The trigger interconnections between the pulse cards must match the diagram shown in [Figure 3-95](#), [Figure 3-96](#), or [Figure 3-97](#). The TRIGGER OUT from the pulse card in the lowest numbered slot (right-most slot) must be cabled into TRIGGER IN of the same card, as well as the TRIGGER IN of all adjacent pulse cards.
2. All size values (PulseVoltagesSize, PrePulseDelaysSize, and so on) must match the value of NumPulseTerminals. As shown in [Figure 3-93](#), NumPulseTerminals = four and there are four entries in:
  - PulseTerminals
  - PulseVoltages
  - PrePulseDelays
  - TransitionTimes
  - PulseWidths
  - PostPulseDelays

This rule must be followed for any Program, Erase or Fast-Program-Erase UTM.

3. Ensure that all time-based pulse parameters are not zero. The minimum time interval is 20 ns (20 E-9). This applies to the parameters:
  - PrePulseDelays
  - TransitionTimes
  - PulseWidths
  - PostPulseDelays

This rule must be followed for any Program, Erase or Fast-Program-Erase UTM.

4. Ensure that all pulse channel waveforms have the same total time, or period.
  - a. To check for a single pulse Program or Erase UTM, add up the following for each channel:
    - PrePulseDelays
    - TransitionsTimes
    - PulseWidths



- PostPulseDelays

All channels should have the same total. If they do not have the same total time, make them the same by modifying the timing.

- b. To check for a double pulse Fast-Program-Erase UTM, add up the following for each channel:

- PrePulse1Delays
- TransitionsTimesPulse1
- Pulse1Widths
- PostPulse1Delays
- PrePulse2Delays
- TransitionsTimesPulse2
- Pulse2Widths
- PostPulse2Delays

All channels should have the same total. If they do not have the same total time, make them the same by modifying the timing.

### Voltage levels do not match expected values

If the voltage at the DUT terminal is not the expected level, please check the following:

1. The pulse channel is a 50  $\Omega$  output and expects a 50  $\Omega$  DUT terminal impedance. For a gate, or other high impedance ( $>1$  k $\Omega$ ) terminal, the voltage at the terminal will be twice (2x) the value specified. For example, setting PulseVoltage = 2 will result in a 4 V level at the DUT gate. See the [Reference manual, Load Line Effect Compensation: Coping with the Load Line Effect, page 11-15](#) for additional details about the effect of the DUT impedance on the pulse level.
2. If the DUT terminal is the drain, alternate manual methods are appropriate. The most common method for determining the pulse voltage level on the drain is to use an oscilloscope with the scope input impedance set to 1 M $\Omega$ .
  - a) Ensure that the gate voltage level meets the desired value before setting other voltage levels.
  - b) Modify the PulseVoltage until the level matched the desired level.

The drain voltage level is a function of the drain-source impedance, that is largely determined by the gate voltage.

## How to perform Charge Pumping

Charge Pumping is a useful technique for understanding gate stack behavior. Charge Pumping characterizes interface and charge-trapping phenomena. The change in the CP results can be used to determine the amount of degradation caused by typical reliability test methods, employing either DC or pulsed stress.

The Model 4200-SCS provides pre-configured tests to perform Charge Pumping. These tests are included in the KITE project plan for Charge Pumping.

The User Library for Charge Pumping is also provided in Section 16 of the Reference Manual (see [chargepumping user library](#)). You can use the chargepumping User Library and an appropriate User Module to create a new UTM (test) in a project plan.



## How to perform a Charge Trapping test

---

**NOTE** The chargetrapping project uses the 4205-PG2 pulse generator card and Model 4200-SCP2 scope card. The Model 4225-PMU can also be used for charge trapping (see [chargepumping user library](#) in Section 16 of the Reference Manual).

---

### Slow single pulse charge trapping high K gate stack

The key to using the single pulse method is to look at the charge trapping and de-trapping behavior within a single, well-configured gate pulse (see [Figure 3-124](#)). The gate pulse usually starts in a position that discharges the gate capacitor before the voltage ramp begins.

This is to clean up any residual charges that might be trapped in the gate. Then, during the rise time of the voltage ramp, the corresponding drain current response is captured, allowing a  $V_{gs}$ - $I_d$  curve to be formed.

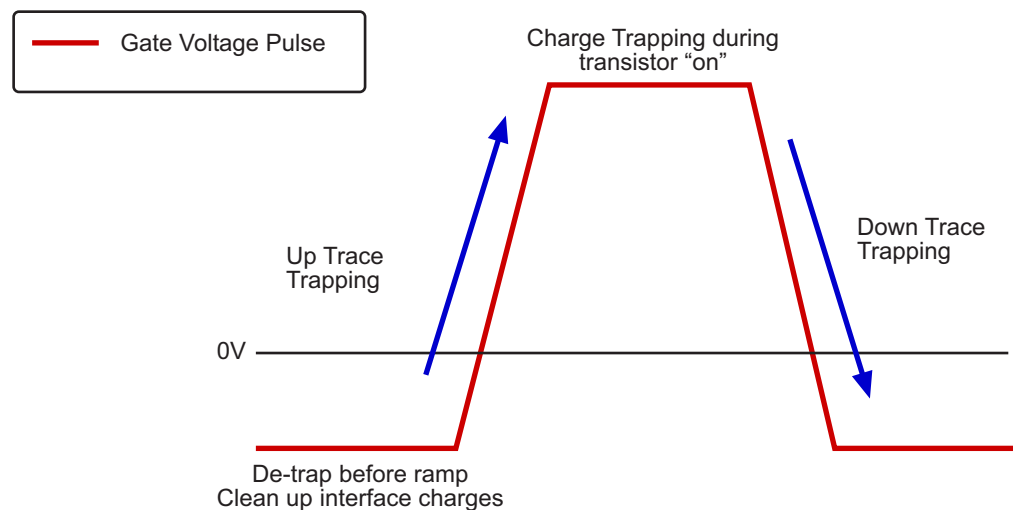
Slow single pulse refers to rise and fall transition times of 100 ns minimum, with a pulse width of at least 1  $\mu$ s.

These relatively slow pulse parameters mean that the RBT are not used and a simple splitter can be used for monitoring the drain current pulse.

For each measurement, a pulse is applied to the gate of the transistor while its drain is biased at a certain voltage. The change in drain current, resulting from the gate pulse, appears on the digital oscilloscope.

Figure 3-124

#### Trapping and de-trapping in a single gate voltage pulse




---

**NOTE** The Charge Trapping project provides two capabilities: Slow charge trapping on a device and the ability to perform relatively generic transient IV tests on a device. The time-based voltages for the gate signal need to be multiplied by 1.33 if the power divider is used as shown in [Figure 3-125](#).

---



Charge trapping procedure

1. Perform cable correction (open and through, if necessary), with calibration substrate. Open and through correction measurements are taken and inputted into correction algorithm to calculate cable losses.
2. Connect DUT (transistor) as shown in [Figure 3-125](#) and [Figure 3-126](#).
3. Input test parameters, refer to key parameters contained in [Table 3-30](#).
4. The UTM will pulse the gate with single pulse (for average >1 use a series of very low duty cycle pulses), bias drain with a PG2, capture drain current response on oscilloscope, then calculate corresponding drain current ( $V_{gs}$ - $I_d$ ) from the whole waveform.
5. To ensure a determinate number of pulses are applied to the DUT, the period must be set to >10 ms. Wider pulse widths require a longer period. If the period is too short, pulse(s) will not be measured and will cause the UTM to hang, requiring KITE to be manually halted.

Figure 3-125  
Slow single pulse—hardware setup block diagram

NOTE This configuration can handle pulse widths  $\geq 100$ ns which is too wide to use Remote Bias Tees.

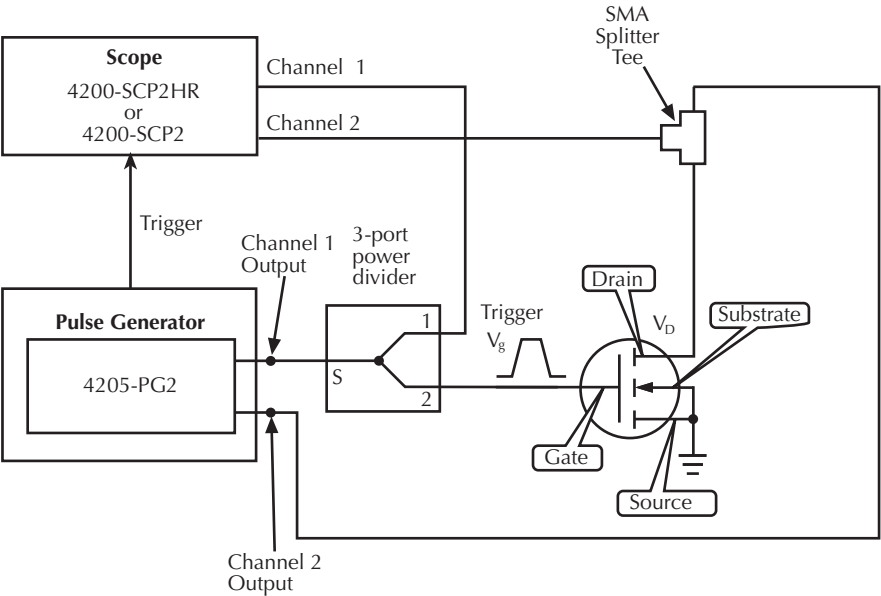


Table 3-30  
Key parameters—Slow Single Pulse Charge Trapping

Parameters	Range/Specification
Application	Pulse I-V like application
Rise / Fall time	Variable 10 ns–10 $\mu$ s
Pulse width	5 $\mu$ s–1 ms single pulse
Pulse amplitude	0-5 V
Base voltage	+/- 5 V
Load impedance	50 $\Omega$ or 1 M $\Omega$



Figure 3-126  
Slow single pulse—hardware connection

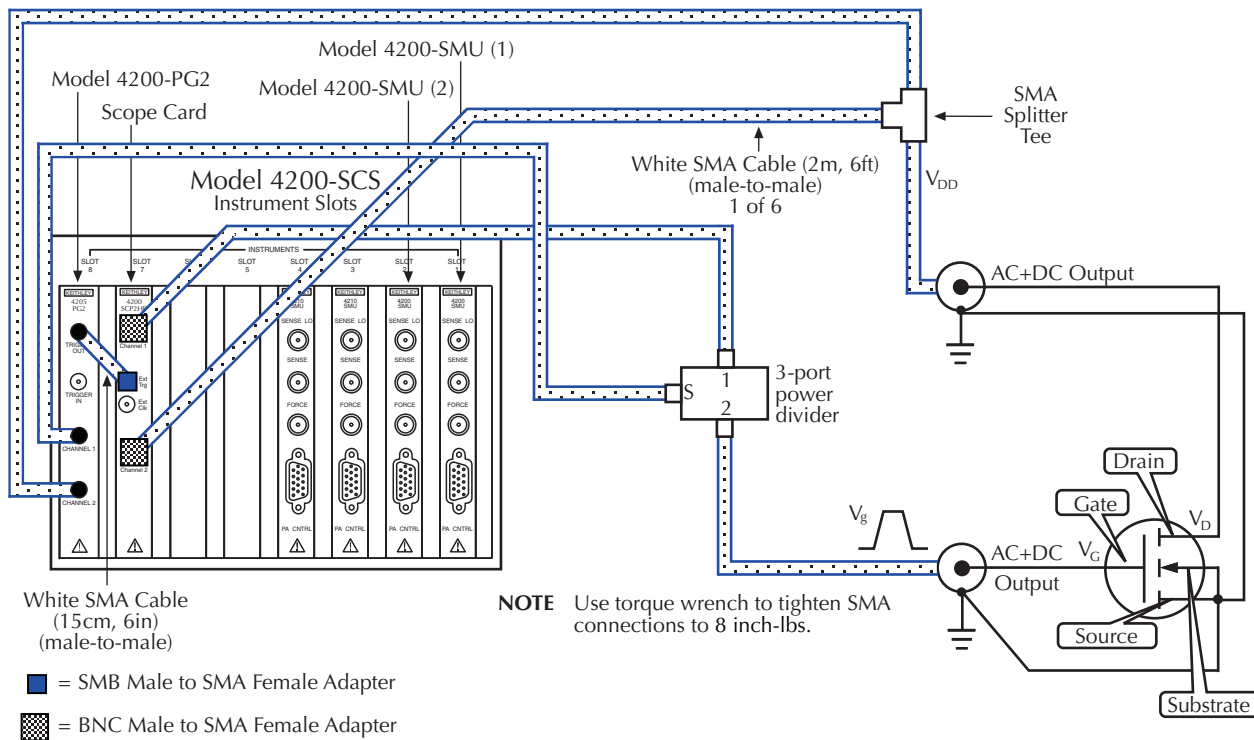


Figure 3-127  
Example slow single pulse waveform graph

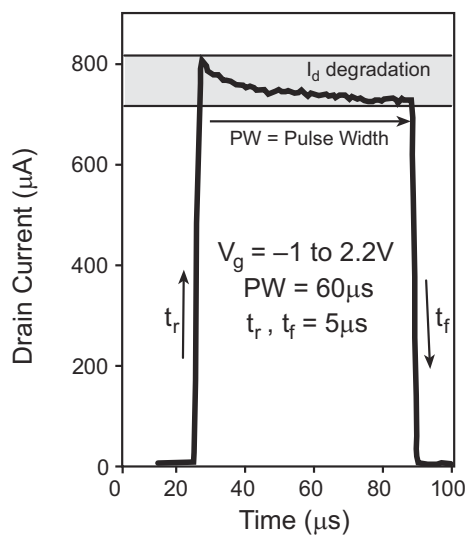
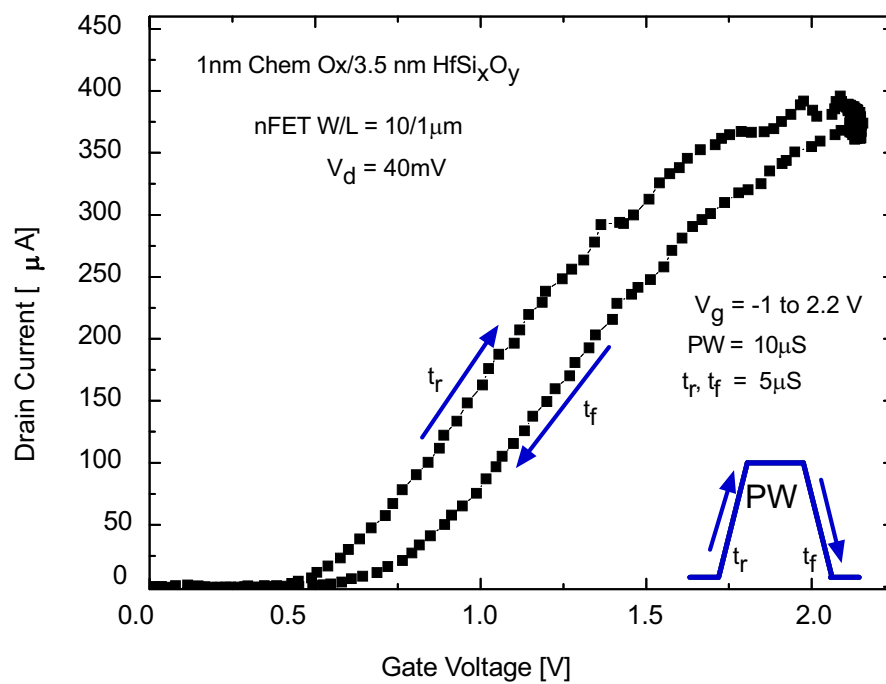




Figure 3-128

**Single slow pulse example data plot**







# How to Control Other Instruments with the Model 4200-SCS

In this section:

Topic	Page
How to control external equipment . . . . .	4-2
Controlling external equipment overview . . . . .	4-2
Keithley Configuration Utility (KCON) . . . . .	4-5
How to control a switch matrix . . . . .	4-7
KCON setup . . . . .	4-8
Open KITE and the ivswitch project . . . . .	4-11
Running test sequences . . . . .	4-11
The connect test . . . . .	4-13
How to control a probe station . . . . .	4-15
Prober control overview . . . . .	4-16
Test system connections . . . . .	4-17
KCON setup . . . . .	4-18
Probe station configuration . . . . .	4-20
Open the probesubsites project . . . . .	4-21
Open the project plan window . . . . .	4-21
Test descriptions . . . . .	4-22
Running the test sequence . . . . .	4-25
Test data . . . . .	4-26
Running individual plans or tests . . . . .	4-27
How to control an external pulse generator . . . . .	4-27
Test system connections . . . . .	4-28
KCON setup . . . . .	4-28
Open the ivpgswitch project . . . . .	4-31
Description of tests . . . . .	4-32
First connect test . . . . .	4-32
First id-vg test . . . . .	4-32
Second connect test . . . . .	4-33
pgu1-init test . . . . .	4-33
pgu1-setup test . . . . .	4-34
pgu-trigger test . . . . .	4-34
Third connect test . . . . .	4-34
Second id-vg test . . . . .	4-34
Running the test sequence . . . . .	4-35
Compare the test results . . . . .	4-35
Overlaying graphs . . . . .	4-36
How to control an external CV analyzer . . . . .	4-37
Connections . . . . .	4-37
KCON setup . . . . .	4-38
Create a new project . . . . .	4-39
Add a subsite plan . . . . .	4-40
Add a device plan . . . . .	4-41
Add a UTM . . . . .	4-42
Modifying the cvsweep UTM . . . . .	4-43
Executing the test . . . . .	4-44
What if my equipment is not listed in KCON . . . . .	4-44



## How to control external equipment

To complete the tutorials in this section, and obtain data that functionally correlates with the sample data and projects provided, you will need the following equipment:

- 1 - Keithley Model 4200-SCS User's Manual with a total of three SMUs (preamps not required)
- 1 - Keithley Model 590 CV Analyzer
- 1 - Hewlett Packard 8110A/81110A Pulse Generator
- 1 - Keithley Model 707 or 708 Switch Matrix
- 1 - Keithley Model 7072 or 7174 8×12 matrix card
- 1 - Keithley Model 8006 Component Test Fixture
- 1 - Probe station (manual or supported semi-automatic) and a wafer containing test devices (MOS capacitor, N-channel MOSFET, and NPN bi-polar transistor)
- 2 - Keithley Model 4801 BNC cables
- 1 - Keithley Model 7078-TRX-BNC adapter
- 1 - Keithley Model 8007-GND-3 cable
- 4 - Keithley Model 4200-MTRX-X cables (0 if using preamps)
- 8 - Keithley Model 4200-TRX-X cables (11 if using preamps)
- 2 - Keithley Model 7007 GPIB cables
- 1 - Keithley Model 236-ILC-3 safety interlock cable

### Controlling external equipment overview

The Keithley Instruments Model 4200 Semiconductor Characterization System (SCS) can control any external instrument or component connected to any of the following communication interfaces:

- IEEE-488 (GPIB) bus
- RS-232 (COM1) port
- Ethernet<sup>1</sup>

When an external instrument is added to the system configuration, it is grouped into one of the following categories:

- Switch matrix
- Capacitance meter
- Pulse generator
- Probe station or test fixture
- General-purpose test instrument

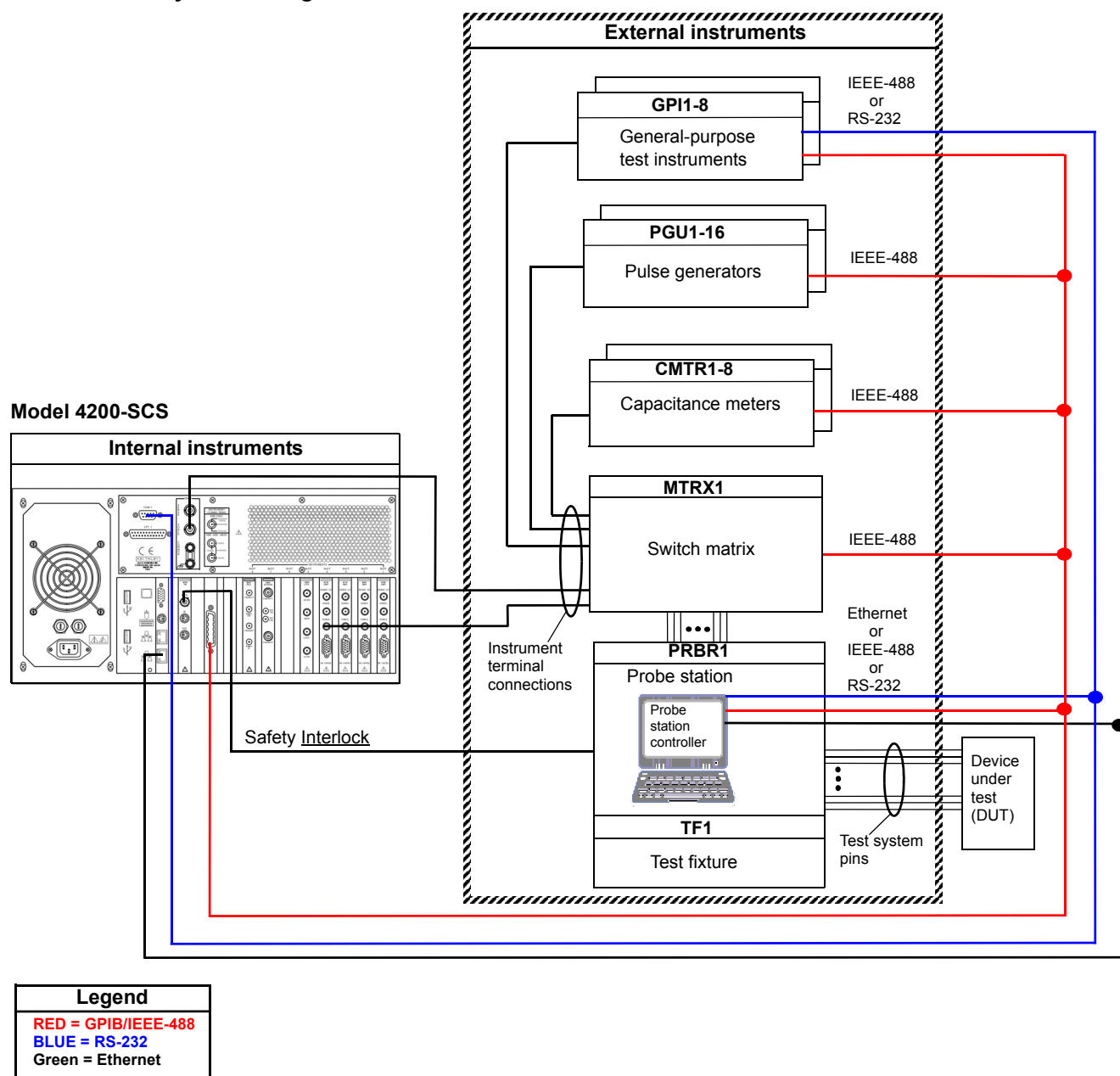
[Figure 4-1](#) shows the relationship between internal and external instrumentation and illustrates each instrument category. For more information about relationships between internal and external instrumentation, refer to the [Reference Manual Keithley CONFIGuration Utility \(KCON\)](#), [page 7-1](#).

---

1. See the [Reference Manual KXCI Ethernet client driver](#), [page 9-102](#) to learn how to enable ethernet control of external instruments.



Figure 4-1  
System configuration with external instruments



User modules are utilized to access these communication interfaces and control external equipment. User modules are stored in user libraries, that are created and maintained with the Keithley User Library Tool (KULT). For information about creating and maintaining user libraries refer to the [Advanced KULT features](#) in Section 8 of the Reference Manual.

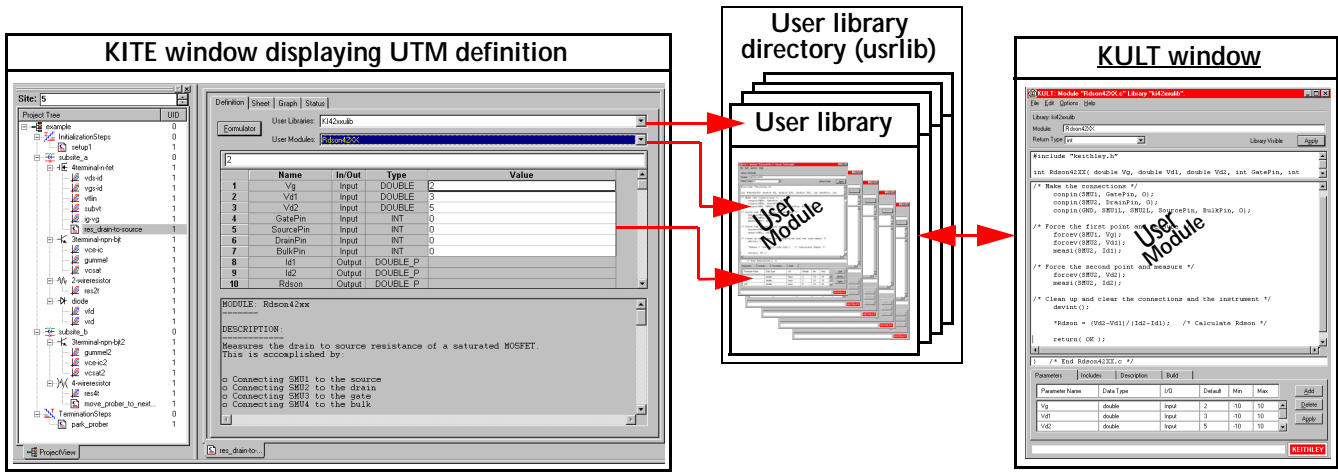
To execute a KULT user module in Keithley Interactive Test Environment (KITE) create a KITE user test module (UTM) and connect it to the user module. Once this user module is connected to the UTM, the following occurs each time KITE executes the UTM:

- KITE dynamically loads the user module and the appropriate user library.
- KITE passes the user-module parameters stored in the UTM to the user module.
- Data generated by the user module is returned to the UTM for interactive analysis.



Figure 4-2 illustrates the relationships between user libraries, user modules, UTMs, KITE, and KULT.

Figure 4-2 Relationships between KULT and KITE - user libraries, user modules, and UTMs



Keithley Instruments provides a number of standard user libraries to control external equipment used in semiconductor characterization applications. Standard libraries of user modules for the following equipment are provided in Table 4-1:

Table 4-1 Supported external equipment table

Category	Instrument	Keithley user library / additional information
Switch matrix	Keithley Model 707/707A Switching Matrix	matrixulib / Model 4200-SCS Reference Manual
Capacitance meter	Keithley Model 590 CV Analyzer	ki590ulib / Model 4200-SCS Reference Manual
	ki595ulib	Model 595 Quasistatic CV Meter Instruction Manual (document number 595-901-01)
	ki82ulib	Model 4200-SCS Reference manual
	Hewlett Packard Model 4284 LCR Meter	hp4284ulib / Model 4200-SCS Reference Manual
Pulse generator	Hewlett Packard Model 8110A Pulse Generator	hp8110ulib / Model 4200-SCS Reference Manual
	Keithley Model 340X Pulse Generator	KI340xulib / Model 4200-SCS Reference Manual
Probe station	Karl Suss Model PA-200 Semiautomatic probe station	prbgen / Model 4200-SCS Reference Manual
	Micromanipulator Model 8860 Semiautomatic probe station	prbgen / Model 4200-SCS Reference Manual
	Manual or Fake probe station	prbgen / Model 4200-SCS Reference Manual
Test fixture	Keithley Model 8006 Component Test Fixture	(not applicable)
	Keithley Model 8007 Semiconductor Test Fixture	(not applicable)
	Generic test fixture	(not applicable)
General-purpose test instrument	(Any IEEE-488 or RS-232 or controlled instrument or Ethernet equipment)	(created by user)



---

**NOTE** *Contact Keithley for the most current list of supported external equipment.*

---

## Keithley Configuration Utility (KCON)

You do not need to perform system configuration operations if you use only internal instruments — factory-installed Source-Measure Units (SMUs), preamps, and the ground unit (GNDU). The Model 4200-SCS automatically detects internal instruments and configures the system accordingly for local operation.

However, after adding supported external instruments, switch matrices, external General Purpose Instrument Bus (GPIB), probe stations, and so on. You must properly configure the system so that KITE and Keithley External Control Interface (KXCI) can utilize these resources. Also, if you need remote operation of the Model 4200-SCS, through KXCI, you must further configure the system.

Perform these configurations using the Keithley CONfiguration utility (KCON). [Figure 4-3](#) provides an overview of KCON features. For details about using KCON, refer to the [Reference Manual Keithley CONfiguration Utility \(KCON\), page 7-1](#).

To start KCON, double-click the KCON icon on the Windows desktop.

---

**NOTE** *If KCON is running, you cannot start KITE or KXCI. If KITE or KXCI is already running, you can start KCON in read-only mode and cannot save any system configuration changes that you make.*

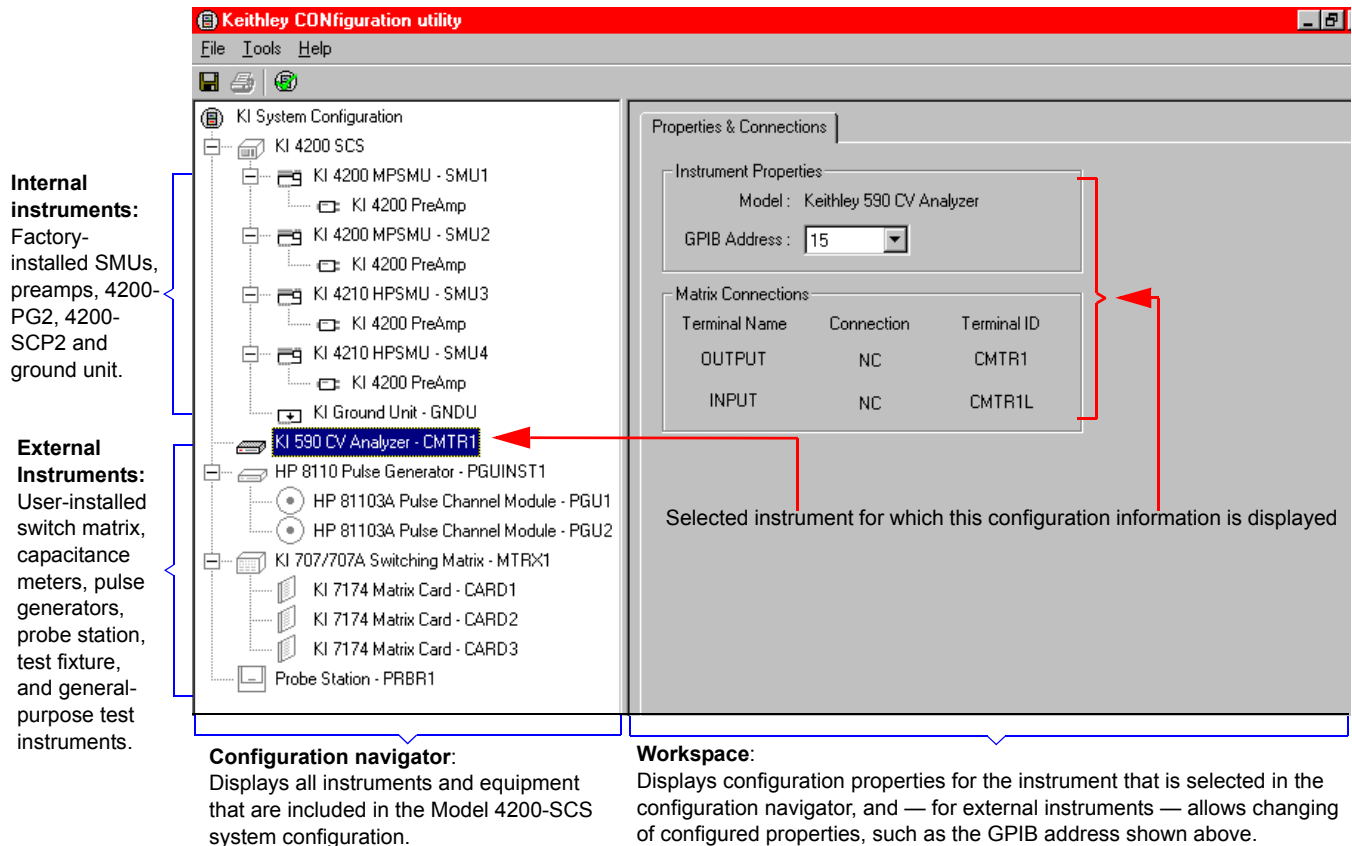
*If you select **KI System Configuration** in the KCON configuration navigator, the Workspace displays a summary of the entire system configuration.*

*If you select **KI Model 4200 SCS** in the configuration navigator, the workspace displays abbreviated system properties and SMU slot assignments, and allows you to perform the following:*

- *Specify the correct power line frequency (60 Hz or 50 Hz) for your installation.*
  - *Configure the system for remote control using KXCI.*
  - *Specify a particular user library to be the active user library, or the active user library will default to C:\S4200\kiuser\usrlib.*
-



Figure 4-3  
Keithley CONFIGuration Utility Overview

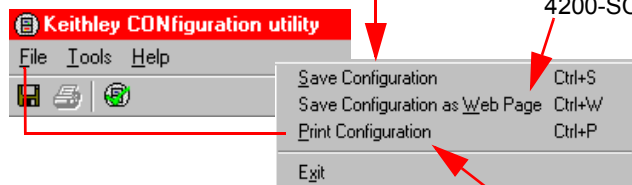


Figures 4-4, 4-5, and 4-6 describe the menu options of the KCON utility graphical user interface (GUI).

Figure 4-4  
KCON utility file menu

Saves the revised system configuration, making it the working configuration for KITE, KULT, or KXCI. If you do not save the changes, the configuration reverts to the last-saved configuration.

Saves the system configuration as an HTML file that can be viewed in a web browser. If you first select **KI System Configuration** in the configuration navigator, this menu item generates a web page that displays general Model 4200-SCS system information.



Closes the KCON program. If you revised but did not save the configuration, KCON reminds you.

Prints the general system configuration information — the information that displays in the KCON workspace when **KI System Configuration** is selected in the configuration navigator.



Figure 4-5  
KCON Utility Tools menu

Adds a supported external instrument that is selected by category in the first submenu and, where applicable, by model number in the second submenu. External instruments are controlled by KITE User Test Modules (UTMs). UTMs are in turn connected to KULT user modules — libraries that are included with the Model 4200-SCS.

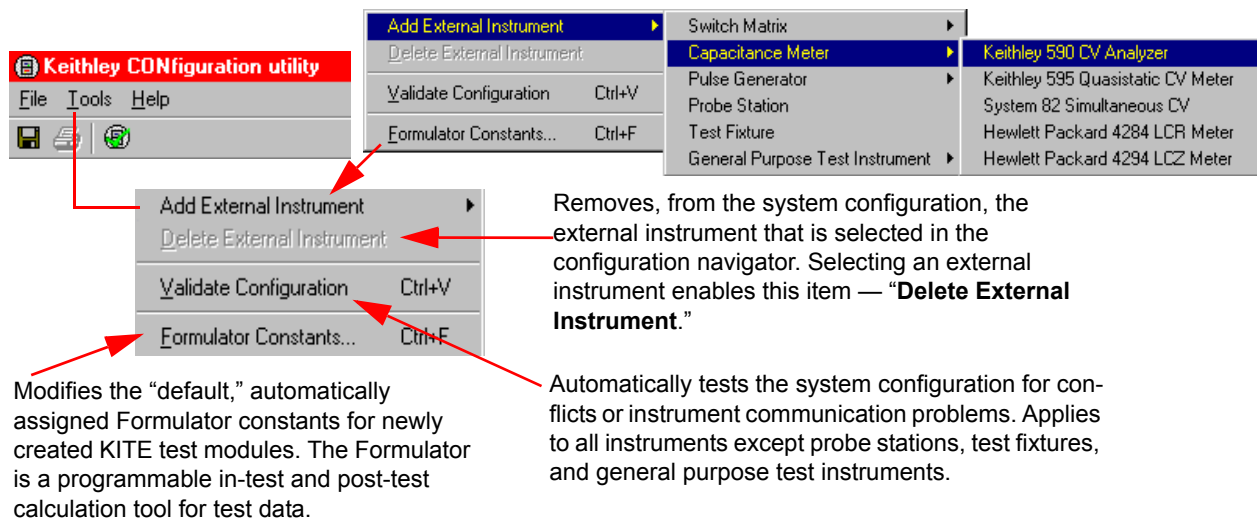
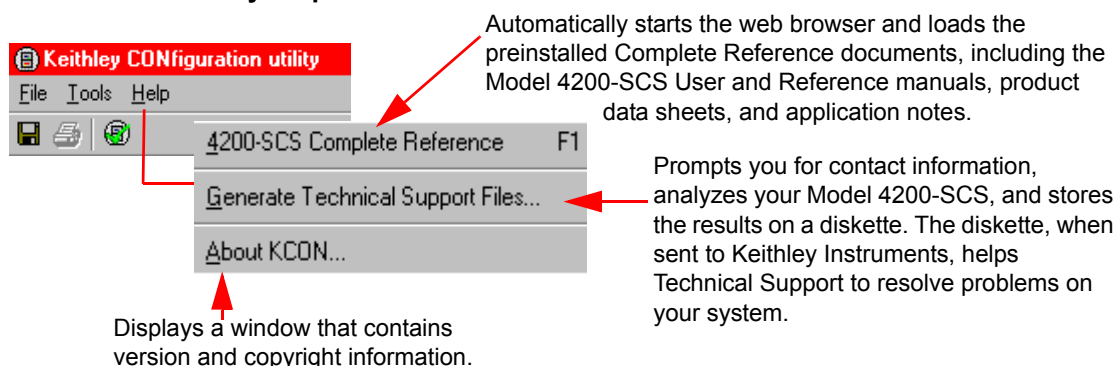


Figure 4-6  
KCON Utility Help menu



## How to control a switch matrix

This tutorial demonstrates how to use a switch matrix to connect any instrument terminal to any test system pin automatically. The `ivswitch` sample project will be used to illustrate this functionality. Before loading and running the `ivswitch` project, the Model 4200-SCS, switch matrix, and component test fixture must be connected as illustrated in [Figure 4-7](#).

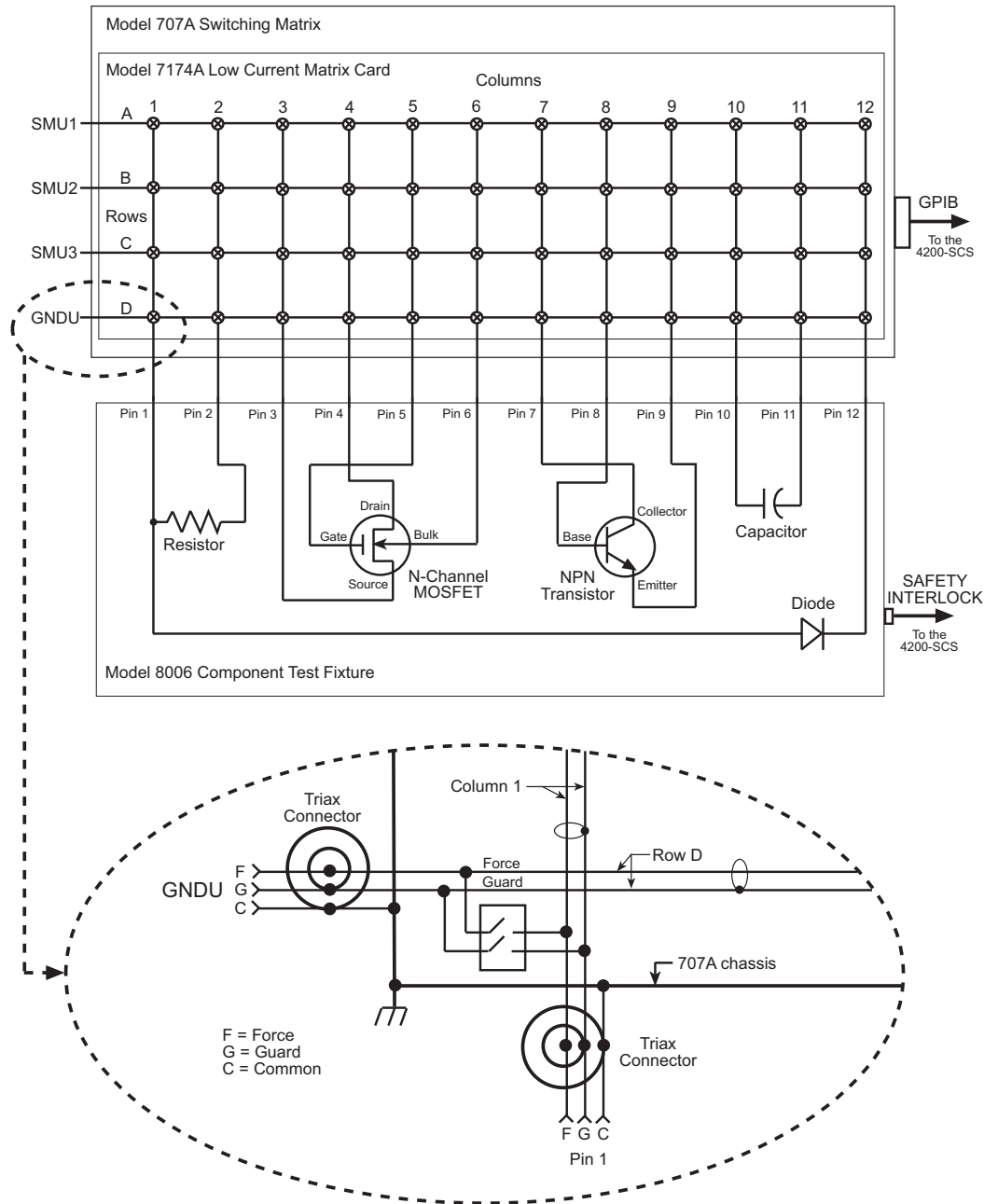
The switch matrix is controlled by the Model 4200-SCS using the GPIB bus. Use a Model 7007 GPIB cable to connect the Model 707 Switching Matrix to the Model 4200-SCS. For connection details, refer to the [Reference Manual KI 70X Switching Matrix Properties tab, page 7-29](#) and [Test connections for a switch matrix, page 15-12](#). The example below shows a Model 7174A matrix card installed in slot 1 of a Model 707A Switching Matrix. The row-column connection scheme is used for this tutorial.

A UTM is used to control the switch matrix. When a test sequence for a device is started, the UTM will close the appropriate matrix crosspoints to connect the specified instrument terminals to the



appropriate test system pins. For details about UTMs, refer to the Model 4200-SCS Reference Manual.

Figure 4-7  
Devices connected to 707A switching matrix



## KCON setup

After connecting the system as indicated in [Figure 4-7](#), run the KCON utility to add the switch matrix and test fixture to the system configuration. KCON is used to manage the configuration of all instrumentation controlled by the Model 4200-SCS software. Once the matrix and test fixture(s) have been added, and the instrument-to-matrix-to-pins connections have been defined, simply specify an instrument terminal and test system pin, and KITE will automatically connect the three



using the matrix. Changes to the system configuration will only be necessary when changes to instrument-to-matrix-to-pins wiring are made.

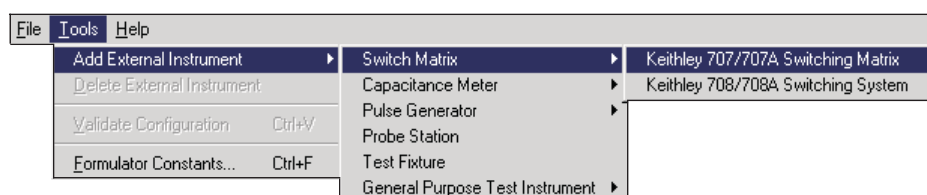
Follow the steps below to start KCON and modify the system configuration as described above. For additional information about KCON, refer to the [Reference Manual Keithley CONfiguration Utility \(KCON\), page 7-1](#). Similarly, for additional information about switch matrix configuration and usage, refer to the [Reference Manual, Appendix B, Using Switch Matrices](#).

#### To add a switch matrix to the system configuration:

1. On the desktop, double-click the **KCON** icon to open **KCON**.
2. Using the **Tools** menu, select **Add External Instrument > Switch Matrix**, and then select the desired switch matrix to the system configuration as indicated in [Figure 4-8](#).

Figure 4-8

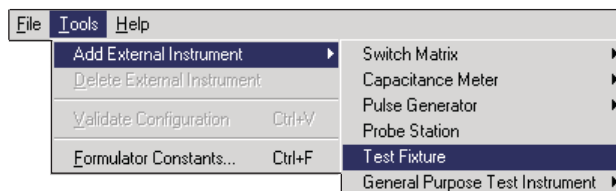
#### Add a switch matrix to the system configuration



3. Using the **Tools** menu, select **Add External Instrument**, and add a **Test Fixture** to the system configuration as indicated in [Figure 4-9](#).

Figure 4-9

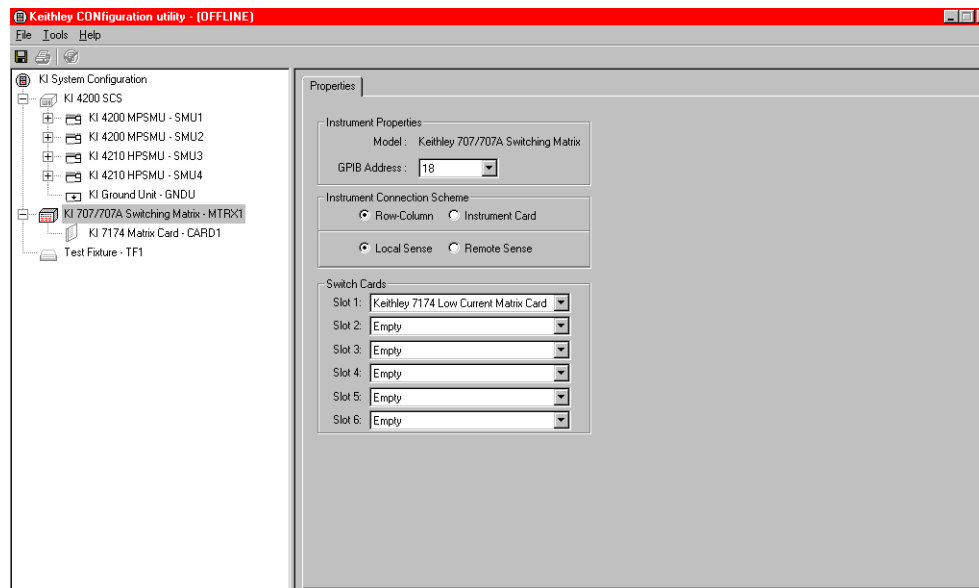
#### Add a test fixture to the system configuration



4. Select the **KI 707/707A Switching Matrix - MTRX1** item in the configuration navigator (tree control on left side of screen) and add a **Keithley 7174 Low Current Matrix Card** to **Slot 1** of the switch matrix. Add the switch card using the pull-down menu on the **Properties** tab. See [Figure 4-10](#).

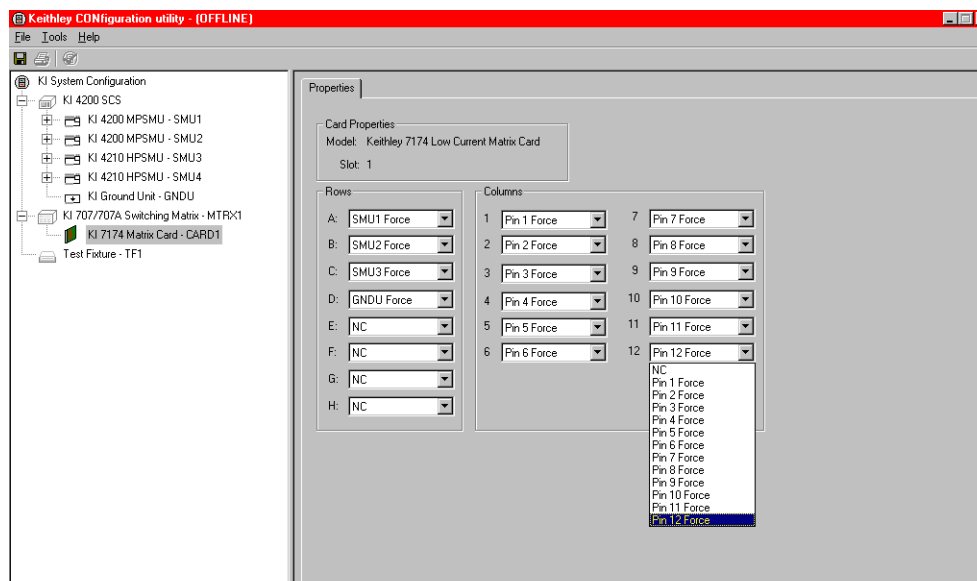


Figure 4-10  
Add a switch card to the system configuration



5. Select the **KI 7174 Matrix Card - CARD1** item in the configuration navigator. Connect the SMUs, GNDU, and test fixture pins as indicated in [Figure 4-7](#) using the pull-down menus on the **Properties** tab. See [Figure 4-11](#).

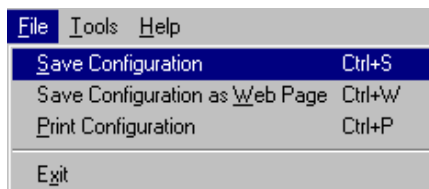
Figure 4-11  
Define the system connections



6. Select **File > Save** and save the system configuration and exit **KCON**. See [Figure 4-12](#).



Figure 4-12  
Save the system configuration

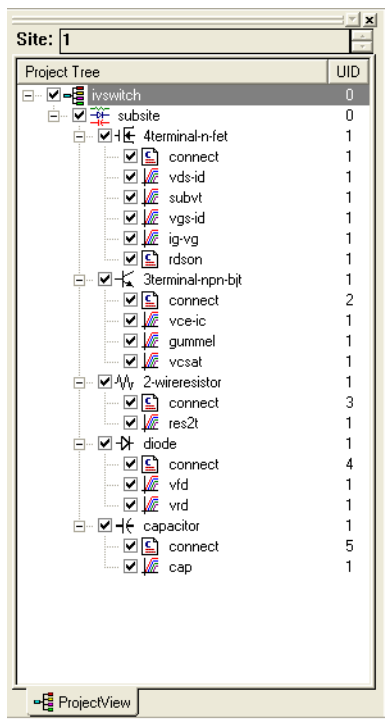


## Open KITE and the ivswitch project

To open KITE and the `ivswitch` project:

1. On the desktop, double-click the **KITE** icon to open KITE.
2. Open the `ivswitch` project from the **File** menu on the KITE toolbar (click **Open Project**).  
The project navigator for the `ivswitch` project is shown in [Figure 4-13](#).

Figure 4-13  
Project navigator - ivswitch project




## Running test sequences

**NOTE** For detailed information about test and sequence execution, refer to the [Run execution of individual tests and test sequences](#) in Section 6 of the Reference Manual.

The `ivswitch` project uses the same ITMs that are used in the default project. The primary difference between the two projects is that the `ivswitch` project uses connect UTMs to control the switch matrix. As shown in [Figure 4-13](#), there is a connect UTM at the beginning of each device test sequence.



A test sequence for a device is executed by selecting the device plan, and then clicking the green **Run** button .

When a device plan is started, the connect test closes the appropriate matrix crosspoints to connect the instruments to the appropriate device.

All devices may be tested by selecting the **Subsite Plan** and clicking the green **Run** button .

Figure 4-14 through Figure 4-18 show the signal paths that are automatically selected for the five devices.

Figure 4-14  
**Signal paths for 4terminal-n-fet tests**

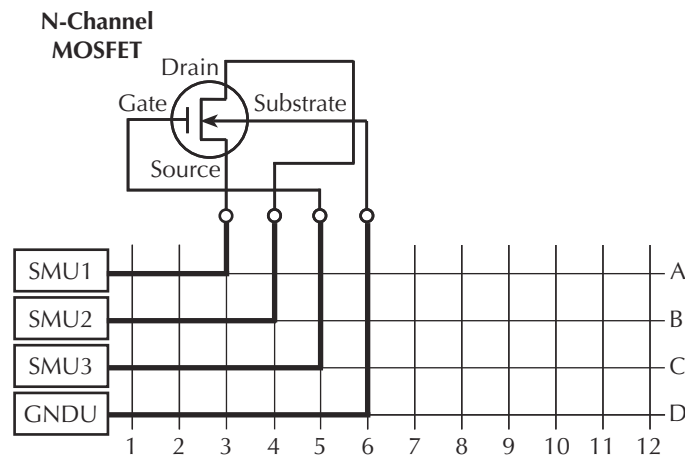


Figure 4-15  
**Signal paths for 3terminal-npn-bjt tests**

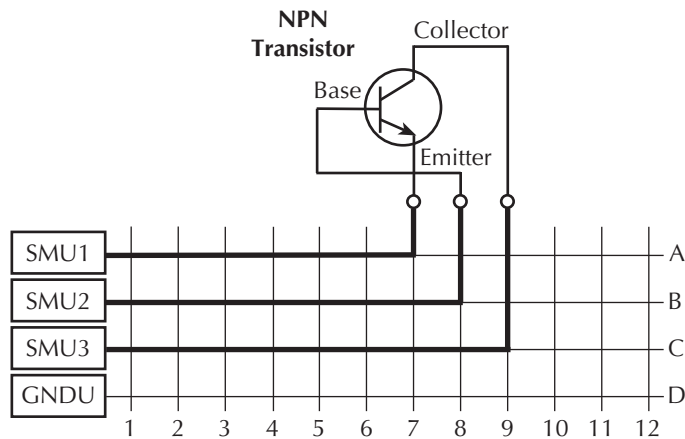




Figure 4-16  
Signal paths for 2-wire resistor tests

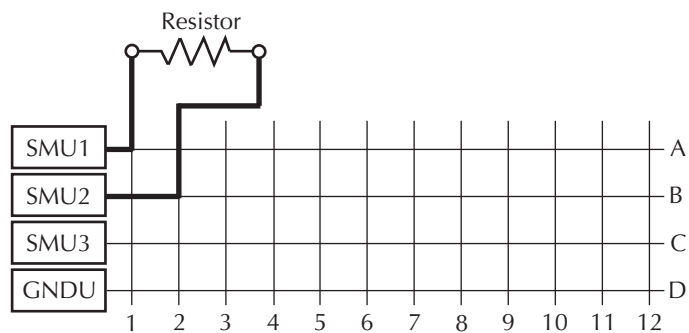


Figure 4-17  
Signal paths for diode tests

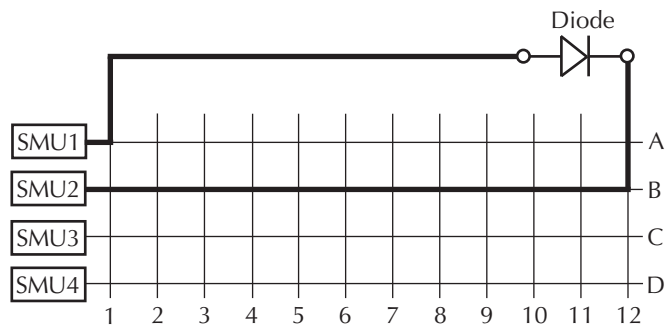
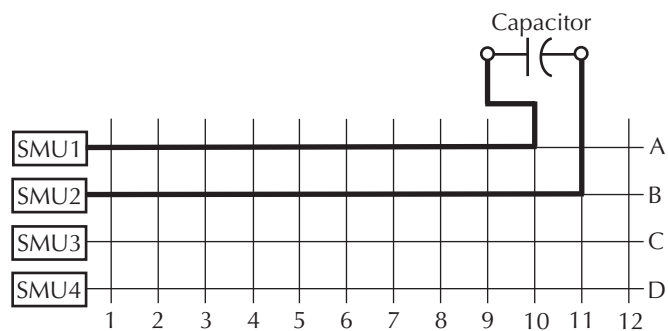


Figure 4-18  
Signal paths for capacitor test

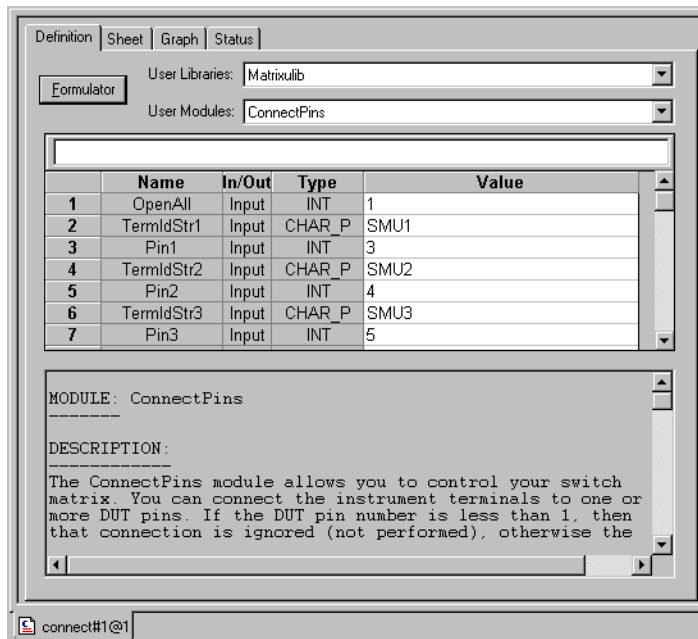


## The connect test

In the project navigator, double-click connect under the 4terminal-n-fet device to open the test. The test is shown in [Figure 4-19](#).



Figure 4-19  
The connect test



The connect test is a UTM. KITE supports two types of test modules: ITMs and UTMs. A UTM, like an ITM, has definition, sheet, graph, and status tabs. The operation of each tab, regardless of test module type, is identical except for the definition tab.

On the UTM **Definition** tab, you can connect the UTM to a user module located within a user library, and then set the module parameter values. This information is stored with the UTM when it is saved. When a UTM is executed, the parameters will be passed from the UTM to the user module and the user module will be executed. User libraries and user modules are created and managed using the Keithley User Library Tool (KULT). For more information about user libraries refer to the [Reference Manual, Keithley User Library Tool \(KULT\), Section 8](#).

In this example, the connect UTM is connected to the **ConnectPins** user module in the **Matrixulib** user library. **ConnectPins** has a total of 17 parameters. The first parameter, **OpenAll**, will cause **ConnectPins** to open all matrix crosspoints before closing any additional crosspoints.

---

**NOTE** *It is a good practice to open all the switch connections before making any new closures.*

---



---

**CAUTION** **Inadvertent switch closures may damage device-under test (DUT).**

---

The 16 additional parameters are comprised of eight terminal-pin pairs. As shown in [Figure 4-20](#), each specified terminal-pin-pair causes **ConnectPins** to make the desired matrix connection. Because the instrument-to-matrix-to-pin connectivity was defined using KCON, KITE is able to connect the specified instrument terminals to the appropriate tester pins automatically .



Figure 4-20

**Connect parameters for 4terminal-n-fet device**

	Name	In/Out	Type	Value	
1	OpenAll	Input	INT	1	← Opens all relays
2	TermIdStr1	Input	CHAR_P	SMU1	← Connects SMU1 to pin 3 of test fixture
3	Pin1	Input	INT	3	←
4	TermIdStr2	Input	CHAR_P	SMU2	← Connects SMU2 to pin 4 of test fixture
5	Pin2	Input	INT	4	←
6	TermIdStr3	Input	CHAR_P	SMU3	← Connects SMU3 to pin 5 of test fixture
7	Pin3	Input	INT	5	←
•					
•					
•					
16	TermIdStr8	Input	CHAR_P	GNDU	← Connects GNDU to pin 6 of test fixture
17	Pin8	Input	INT	6	←

**NOTE** If a pin parameter is < 1, the terminal-pin-pair is ignored and no matrix connections are made.

## How to control a probe station

This tutorial demonstrates how to control a probe station to test five identical sites (or die or reticles) on a sample wafer.

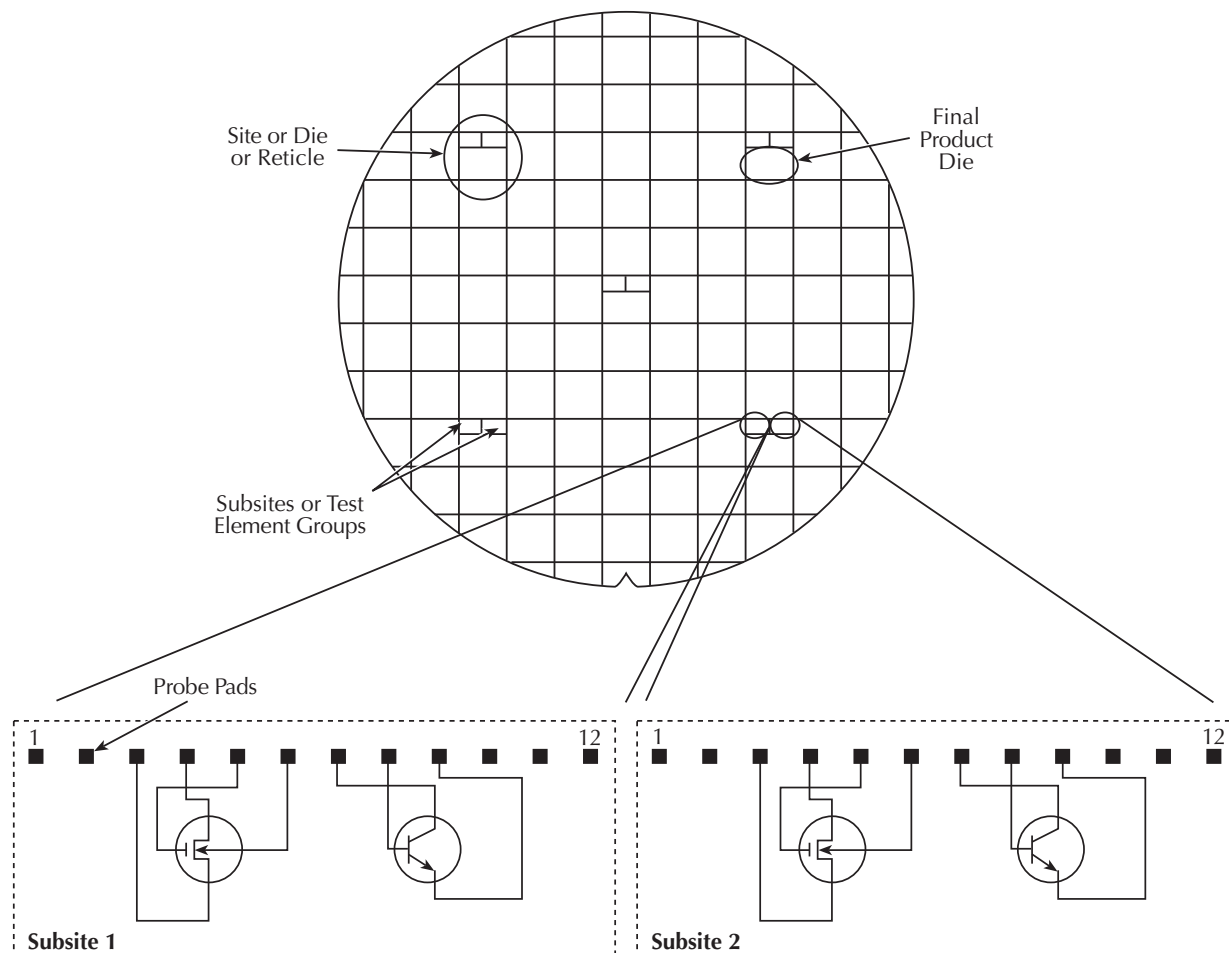
Each wafer site has two subsites (or test element groups). At each subsite there are two devices (or test elements) to be tested:

- 4-terminal N-channel MOSFET
- 3-terminal NPN transistor.

The subsites need not be identical, but for simplicity they are assumed to be the same. This is illustrated below in [Figure 4-21](#).



Figure 4-21  
Sample wafer organization



## Prober control overview

A probe station, like any other external instrument, is controlled by the Model 4200-SCS through user modules. Basic system connections are illustrated in [Figure 4-1](#). A library of user modules, called `prbgen`, is provided with the Model 4200-SCS to facilitate prober control. This generic prober user library, developed and maintained by Keithley Instruments, allows KITE to control all supported probers in the same manner. Therefore, KITE projects utilizing `prbgen` will work with any prober supported by Keithley Instruments. Refer to [Table 4-2](#) for the list of supported probers.

---

**NOTE** The information provided in this overview is a summary of the information provided in the [Reference Manual, Using a Probe Station, Appendix G](#).

---



Table 4-2  
Supported probers

Supported Probe Stations	Additional Information Model 4200-SCS Reference Manual
Suss MicroTec Model PA-200	<a href="#">Appendix H, Suss MicroTec PA-200 Prober</a>
Micromanipulator Model 8860	<a href="#">Appendix I, Micromanipulator 8860 Prober</a>
Manual (or Fake)	<a href="#">Appendix H, Suss MicroTec PA-200 Prober</a>

**NOTE** Contact Keithley Instruments for the most up to date list of supported probe stations.

Sophisticated prober control software, available from each supported prober vendor, provides access to the full feature set of each prober. In all cases, this prober control software provides the ability to define a list of wafer locations to be probed. The Model 4200-SCS relies on the prober controller, and associated software, to maintain this probe list. The `prbgen` user modules communicate with the prober controller, through the GPIB bus or COM1 port in most cases, to instruct it to step through the probe list. This technique of prober control is referred to as learn mode," because the prober control software is taught where each probe location is physically located. [Table 4-3](#) summarizes the user modules included in the `prbgen` prober control user library.

Table 4-3  
`prbgen` user modules

User Module	Description
PrInit	Initializes the prober driver and establishes the reference site (or die). All ITM or UTM data acquired by KITE will be tagged with [row, column] site coordinate information that is relative to the reference site.
PrChuck	Instructs the prober to move the probe station chuck up or down, making or breaking contact between the wafer and test system pins (probe needles).
PrSSMovNxt	Instructs the prober to move to the next subsite (or test element group) in the probe list.
PrMovNxt	Instructs the prober to move to the next site (or die) in the probe list.

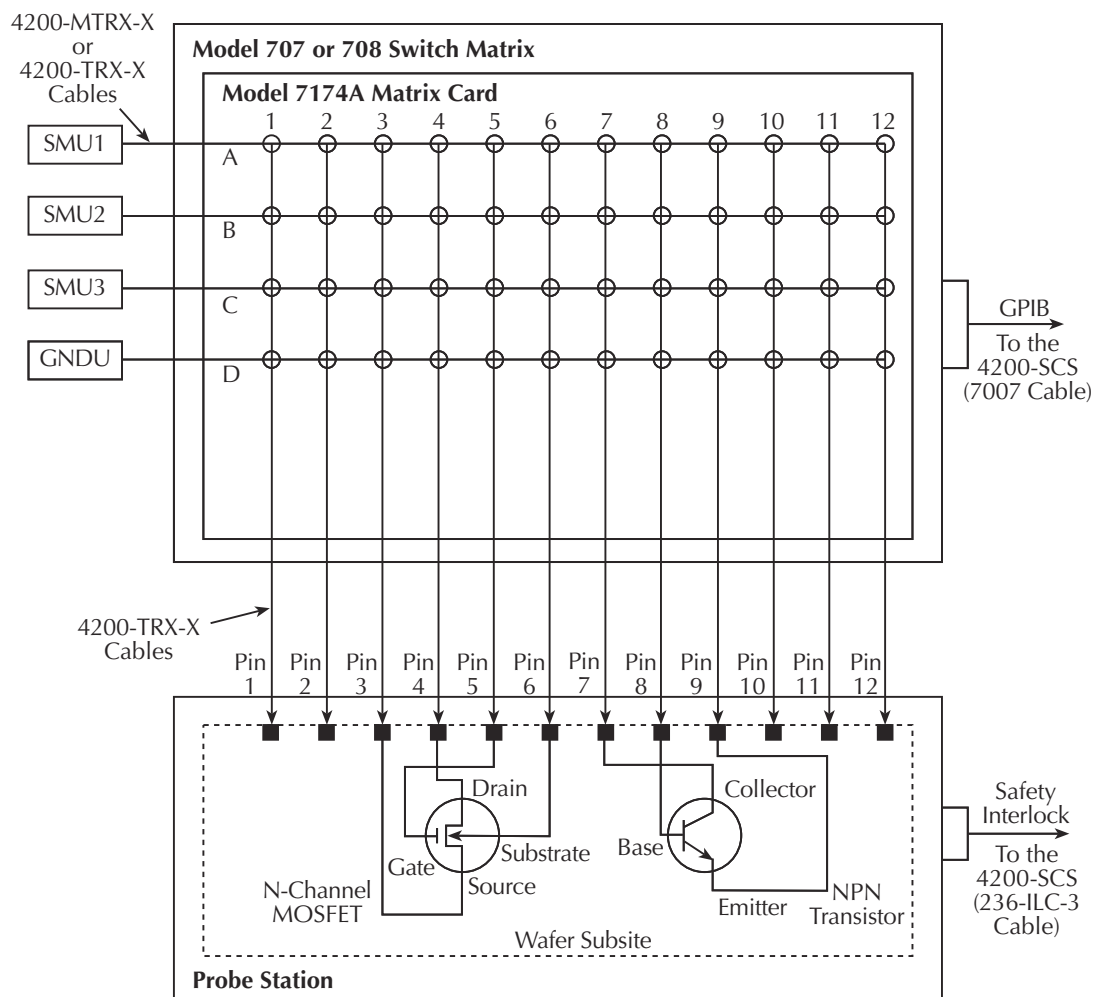
Before a KITE project that utilizes the `prbgen` user library can be executed, the probe list must be created using the appropriate prober control software. Helpful instructions for creating the probe list for each supported prober are included in the [Reference manual, Appendix H](#) and [Appendix I](#). Refer to [Table 4-2](#) for additional information.

## Test system connections

A typical test system for this tutorial is shown in [Figure 4-22](#). As shown, the Model 4200-SCS and probe station are connected to a Model 7174A matrix card. The matrix card is installed in the switch matrix, and the switch matrix and probe station are controlled through the GPIB bus. For connection details as well as information about the KCON utility, refer to the [Reference Manual Keithley CONFIGuration Utility \(KCON\)](#), page 7-1.



Figure 4-22  
**System configuration for the probesubsites project**



## KCON setup

For this tutorial, the following external equipment must be added to the system configuration:

- Switch matrix
- Matrix card
- Probe station

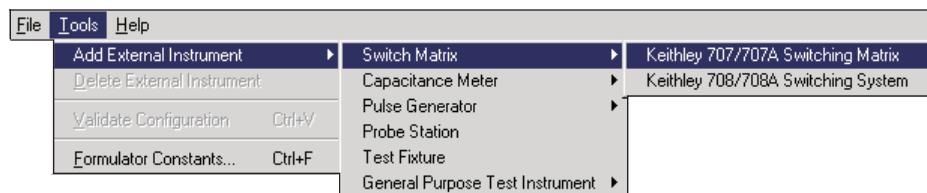
KCON is used to add external equipment to the test system. Below is a step by step procedure for adding the necessary equipment to the system configuration:

### To setup KCON:

1. Start KCON. Double-click the **KCON** icon on the desktop or use the **Start** menu, and select **Start > Programs > Keithley > KCON**.
2. Add the **Keithley Model 707/707A Switching Matrix** to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-23](#).

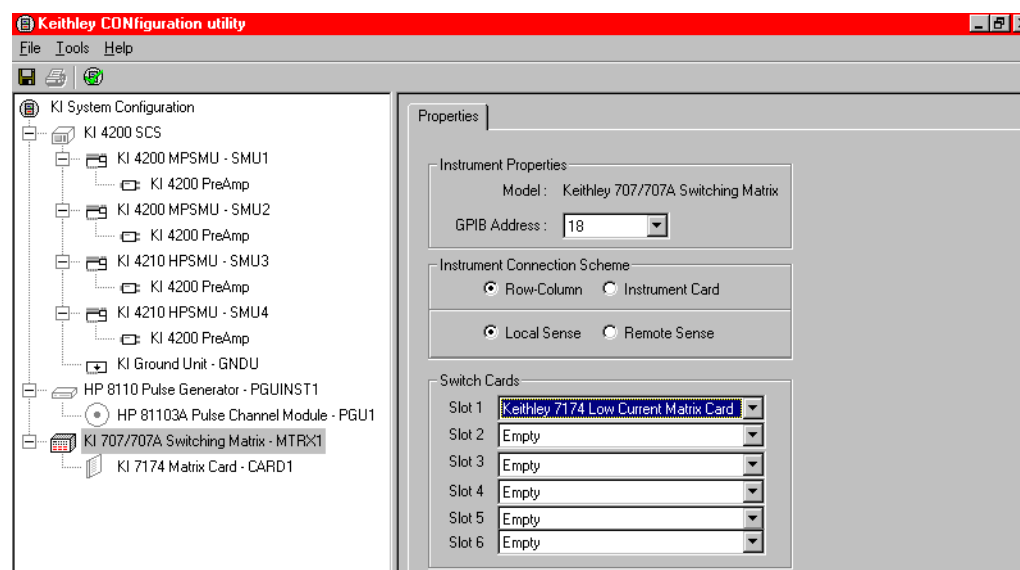


Figure 4-23  
Adding a switch matrix



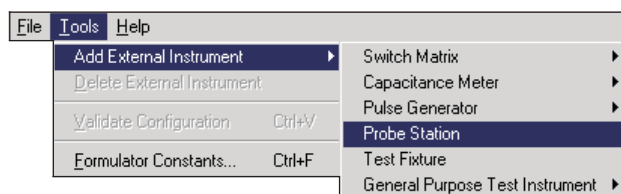
3. Set the GPIB Address for the switch matrix and add the Model 7174A matrix card in Slot 1, as illustrated in [Figure 4-24](#).

Figure 4-24  
Configuring the switch matrix



4. Add a manual probe station to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-25](#). If a test fixture is already part of the configuration, it must be removed before the probe station can be added. To remove any external component from the system configuration, select it in the configuration navigator, and press the **DELETE** key.

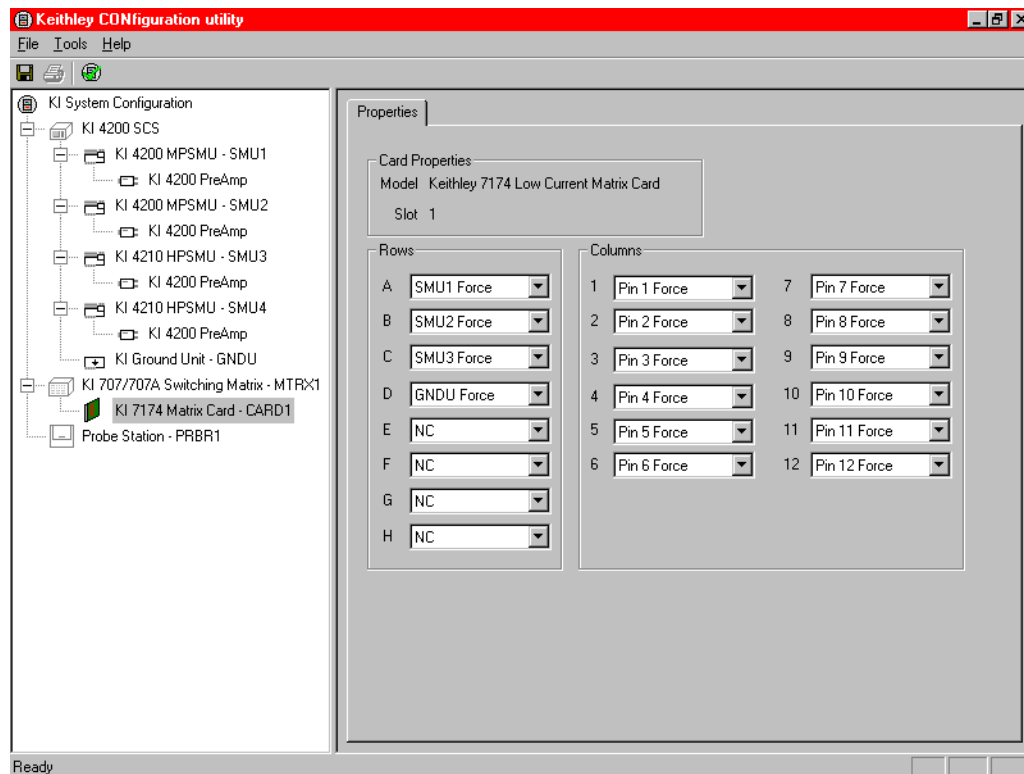
Figure 4-25  
Adding a probe station



5. Connect the instrument terminals and probe station pins to the switch matrix by selecting the **KI 7174 Matrix Card - CARD1** in the configuration navigator and configuring it as illustrated in [Figure 4-26](#). Detailed information about switch matrix configuration can be found in the [Reference Manual, Appendix B, Using Switch Matrices](#).

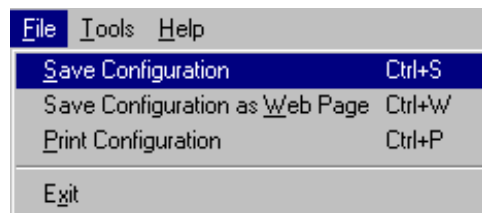


Figure 4-26  
Connecting the switch matrix



6. **Save** the configuration using the **KCON File** menu as illustrated in [Figure 4-27](#).

Figure 4-27  
Saving the system configuration



## Probe station configuration

Before KITE can begin controlling a probe station, the probe station must be properly configured.

**The probe station configuration includes:**

1. Making test system measurement and communication connections.
2. Creating a probe list using the appropriate prober control software.
3. Loading and aligning the wafer.

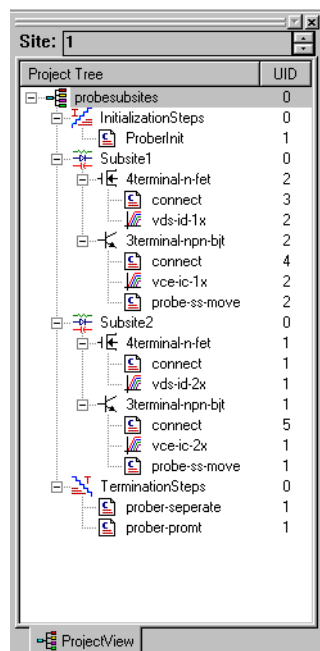
Helpful configuration instructions for each supported prober are included in the [Reference Manual, Appendix H, Probe station configuration](#). Refer to [Table 4-2](#) for additional information. This tutorial uses a manual probe station, however the probe station configuration is simple, because step 2 can be omitted. To configure a manual probe station, connect the test system measurement signals to the probe station as indicated in [Figure 4-22](#), and align the prober to the first subsite (test element group) in the test sequence.



## Open the probesubsites project

Open the probesubsites project from the **File** menu on the KITE toolbar (click **Open Project**). The project navigator for the probesubsites project is shown in [Figure 4-28](#).

Figure 4-28  
Project navigator



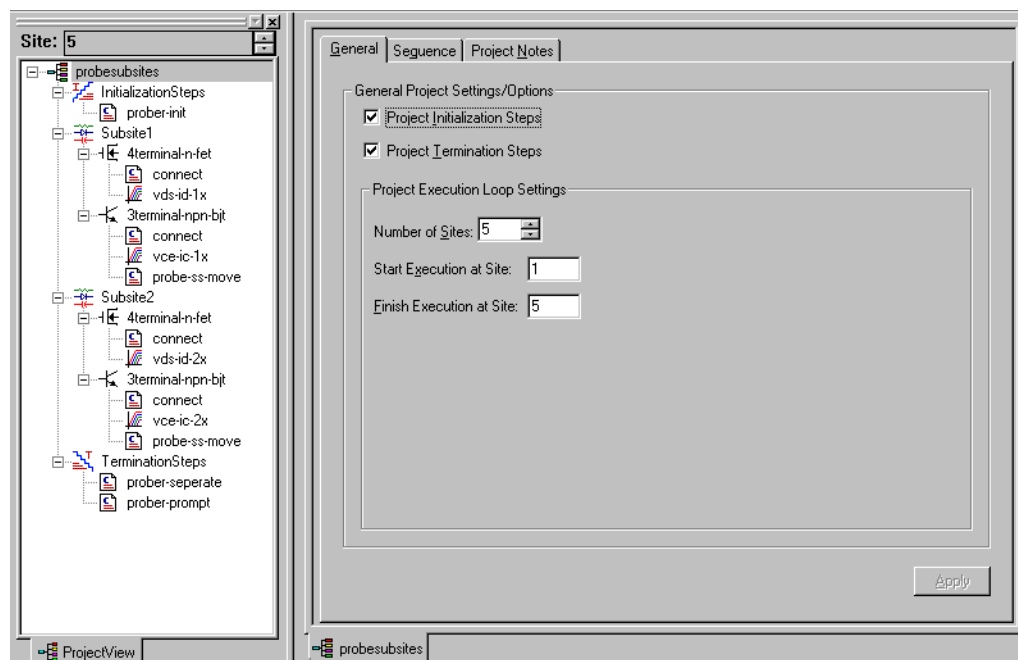
## Open the project plan window

In the project navigator, double-click **probesubsites** to open the project plan window. For this tutorial, five sites on a wafer are to be tested. As shown in [Figure 4-29](#), set up the project plan as follows and click the **Apply** button at the bottom right-hand corner of the window:

1. Enable ( ✓ ) Project Initialization Steps
2. Enable ( ✓ ) Project Termination Steps
3. Start Execution at Site: 1
4. Finish Execution at Site: 5



Figure 4-29  
Modified project plan settings



## Test descriptions

Test descriptions for the probesubsites project are provided in [Table 4-4](#). Tests can be opened in the workspace by double-clicking them in the project navigator.

---

**NOTE** The connect UTMs are used to control the switch matrix.

---



Table 4-4  
probesubsites test descriptions

probesubsites Project	Test Description
<b>InitializationSteps</b> prober-init	Initializes the prober driver (see <a href="#">Figure 4-30</a> ).
<b>Subsite1</b> 4terminal-n-fet connect  vds-id-1x 3terminal-npn-bjt connect vce-ic-1x probe-ss-move	Connects the SMUs to the probes for the N-channel MOSFET (see <a href="#">Figure 4-31</a> ). Generates a family of curves ( $I_D$ vs. $V_D$ ) for the MOSFET.  Connects the SMUs to the probes for the NPN transistor (see <a href="#">Figure 4-32</a> ). Generates a collector family of curves ( $I_C$ vs. $V_C$ ) for the transistor. Moves the prober to next subsite.
<b>Subsite2</b> 4terminal-n-fet connect  vds-id-2x 3terminal-npn-bjt connect vce-ic-2x probe-ss-move	Connects the SMUs to the probes for the N-channel MOSFET (see <a href="#">Figure 4-31</a> ). Generates a family of curves ( $I_D$ vs. $V_D$ ) for the MOSFET.  Connects the SMUs to the probes for the NPN transistor (see <a href="#">Figure 4-32</a> ). Generates a collector family of curves ( $I_C$ vs. $V_C$ ) for the transistor. Moves the prober to the first subsite of the next site.
<b>TerminationSteps</b> prober-separate prober-prompt	The following steps occur after all three sites are tested: Separates the prober pins from the wafer (see <a href="#">Figure 4-33</a> ). Displays a pop-up window indicating that testing is finished (see <a href="#">Figure 4-34</a> ).

Figure 4-30  
prober-init

	Name	In/Out	Type	Value
1	mode	Input	INT	6
2	x_die_size	Input	DOUBLE	2.200000e+001
3	y_die_size	Input	DOUBLE	2.200000e+001
4	x_start_position	Input	INT	0
5	y_start_position	Input	INT	0
6	units	Input	INT	1
7	subproptype	Input	INT	0

- Line 1:** Parameter value **6** selects the Learn control mode. Assumes that the probe list is maintained by the prober controller software.
- Lines 2 and 3:** These parameters (along with the units setting in Line 6) input a die size of 22 mm x 22 mm.
- Lines 4 and 5:** These parameters input the initial prober position as the 0, 0 coordinates.
- Line 6:** Parameter value 1 sets units for die size (lines 2 and 3) to metric.
- Line 7:** (not used)



Figure 4-31  
**Connect SMUs to N-channel MOSFET**

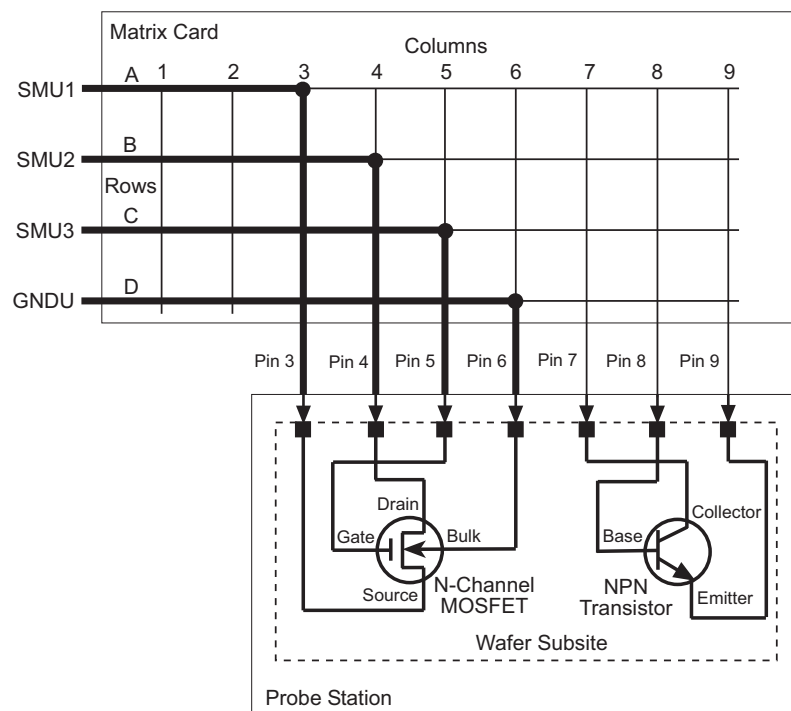


Figure 4-32  
**Connect SMUs to NPN transistor**

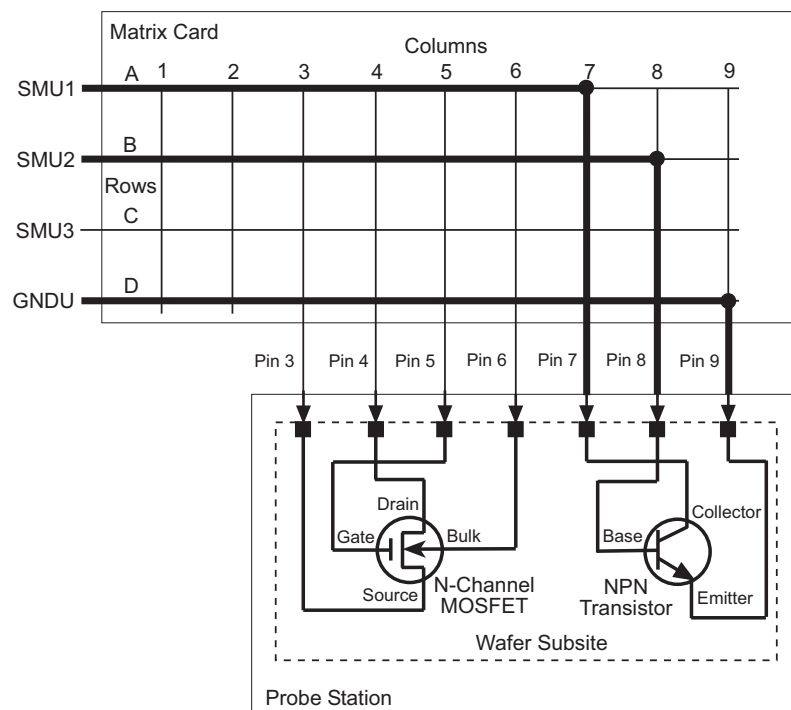




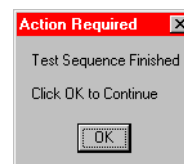
Figure 4-33  
**prober-separate**

	Name	In/Out	Type	Value
1	chuck_position	Input	INT	0
2				
3				
4				
5				
6				
7				

**Line 1:** Parameter value **0** separates the prober pins from the wafer.

Figure 4-34  
**prober-prompt test and dialog window**

	Name	In/Out	Type	Value
1	NumberOfMessages	Input	INT	3
2	Message1Text	Input	CHAR_P	Test Sequence Finished
3	Message2Text	Input	CHAR_P	Click OK to Continue
4	Message3Text	Input	CHAR_P	
5	Message4Text	Input	CHAR_P	
6				



A. Prober-prompt test window


B. Dialog box

**Line 1:** Parameter value **3** specifies three lines of text to be displayed.

**Lines 2 thorough 5:** Text messages to be displayed in dialog box when testing is finished (see B. Dialog box).

## Running the test sequence

To test the five wafer sites:

1. Manually align the prober to test Subsite 1 of Site 1. Make sure the prober pins are making contact with the wafer probe pads.
2. In the project navigator, click **probesubsites** in the KITE project navigator to select the project.
3. Click the green **Run** button  to execute the test sequence.

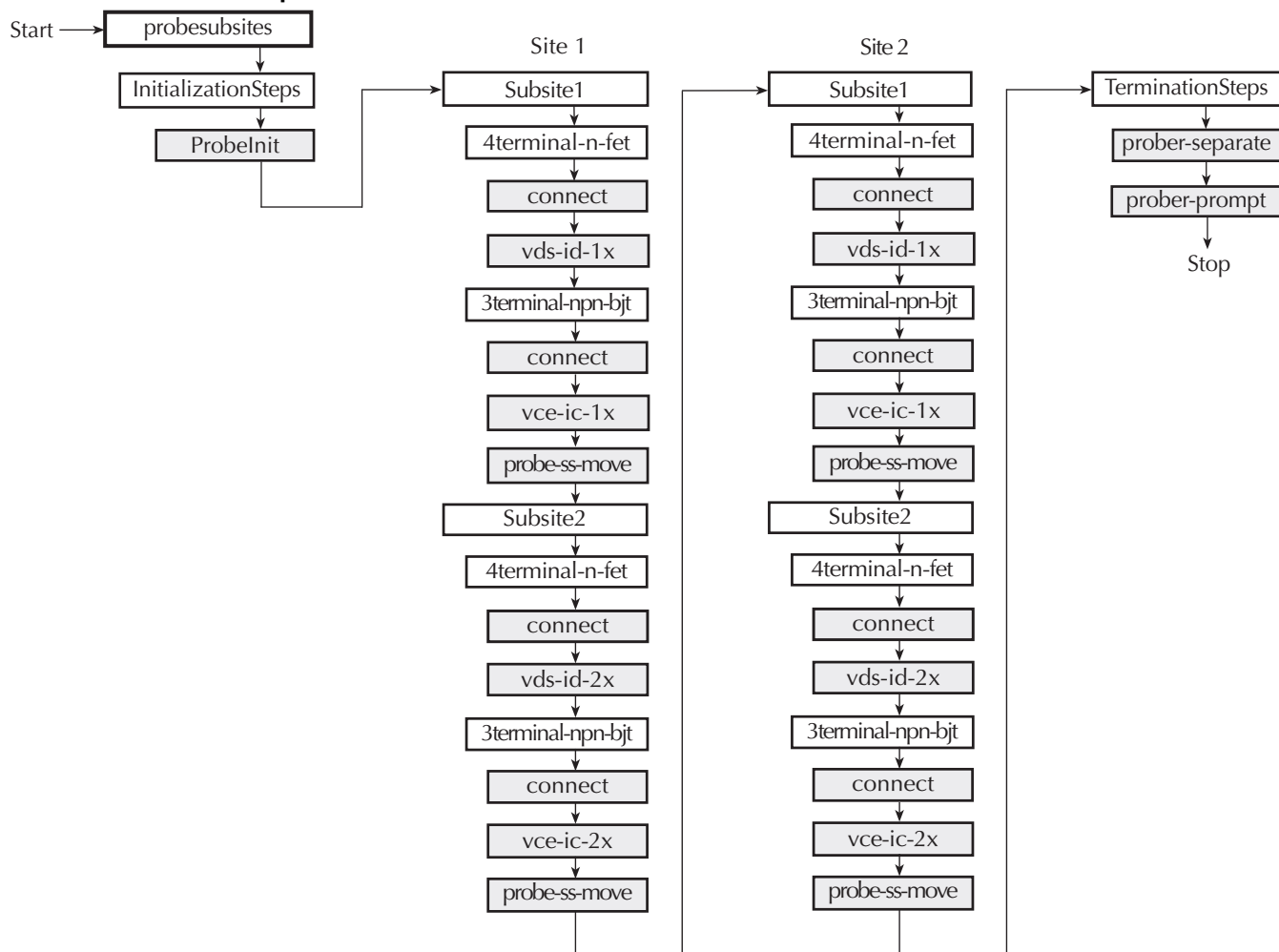
**NOTE** Because a manual probe station is being used, the prober will not actually move when the prober control UTM's are executed. However, a pop-up dialog box will appear, instructing you to move the probes to the next subsite in the test sequence.



The test sequence is shown in [Figure 4-35](#). After the probe is initialized by the `prober-init` command, the tests for subsite 1 and subsite 2 are performed at site 1. The last test for site 1 (`probe-ss-move`) moves the probe to site 2, where the subsite tests are repeated.

After all five sites are tested, the probe pins separate from the wafer (`prober-separate`), and a dialog box (`prober-prompt`) will alert you that the test sequence is finished (see [Figure 4-34B](#)). Click **OK** to continue.

Figure 4-35  
Test sequence



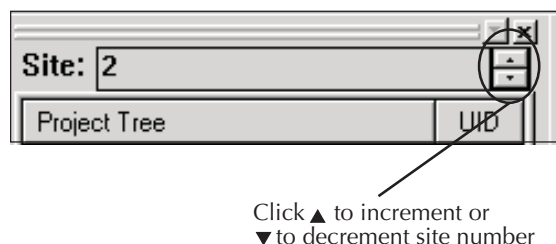
## Test data

Since five sites were tested, there will be five sets of test data: one for each site. Remember, a test is opened by double-clicking it in the project navigator. Test data is viewed by clicking the **Graph** or **Sheet** tab for the test.

When you double-click a test to open it, its test data corresponds to the site number displayed by the site navigator at the top of the project navigator. As shown in [Figure 4-36](#), click the up or down arrow to change the site number. For example, to view test data for Site 2, set the site navigator to Site 2 and double-click the desired test.

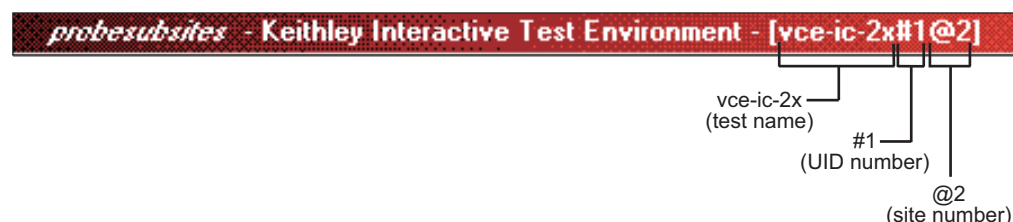


Figure 4-36  
Site Navigator



The title bar at the top of the KITE panel indicates which test is presently being displayed. In [Figure 4-37](#), test `vce-ic-2x` for Site 2 is displayed. The unique identifier (UID) distinguishes this test from any other test having the same name.


Figure 4-37  
KITE title bar



## Running individual plans or tests

You can run any subsite plan, device plan, or test in the project. The test sequence will stop after the plan or test is finished.

### To show how to run the 3terminal-npn-bjt device plan for subsite 2 of site 2:

1. Manually position the prober to test **Subsite 2** of **Site 2**. Make sure the prober pins are making contact with the subsite pads.
2. Set the site navigator to **Site 2**.
3. In the project navigator, click **3terminal-npn-bjt** for Subsite2 to select the device plan.
4. Click the green **Run** button  to start the test sequence.

## How to control an external pulse generator

This tutorial demonstrates how to control a pulse generator to stress a semiconductor device and analyze the effects of the stress. The applied stress is a burst of 3.5V pulses across the gate-substrate (bulk) terminals of an N-channel MOSFET.

### To run the basic test sequence:

1. Measure the transfer characteristics of the device before the stress.
2. Apply a stress burst of 3.5V pulses.
3. Measure the transfer characteristics of the device after the stress.

The **after-stress** characteristics can then be compared to the **before-stress** characteristics.

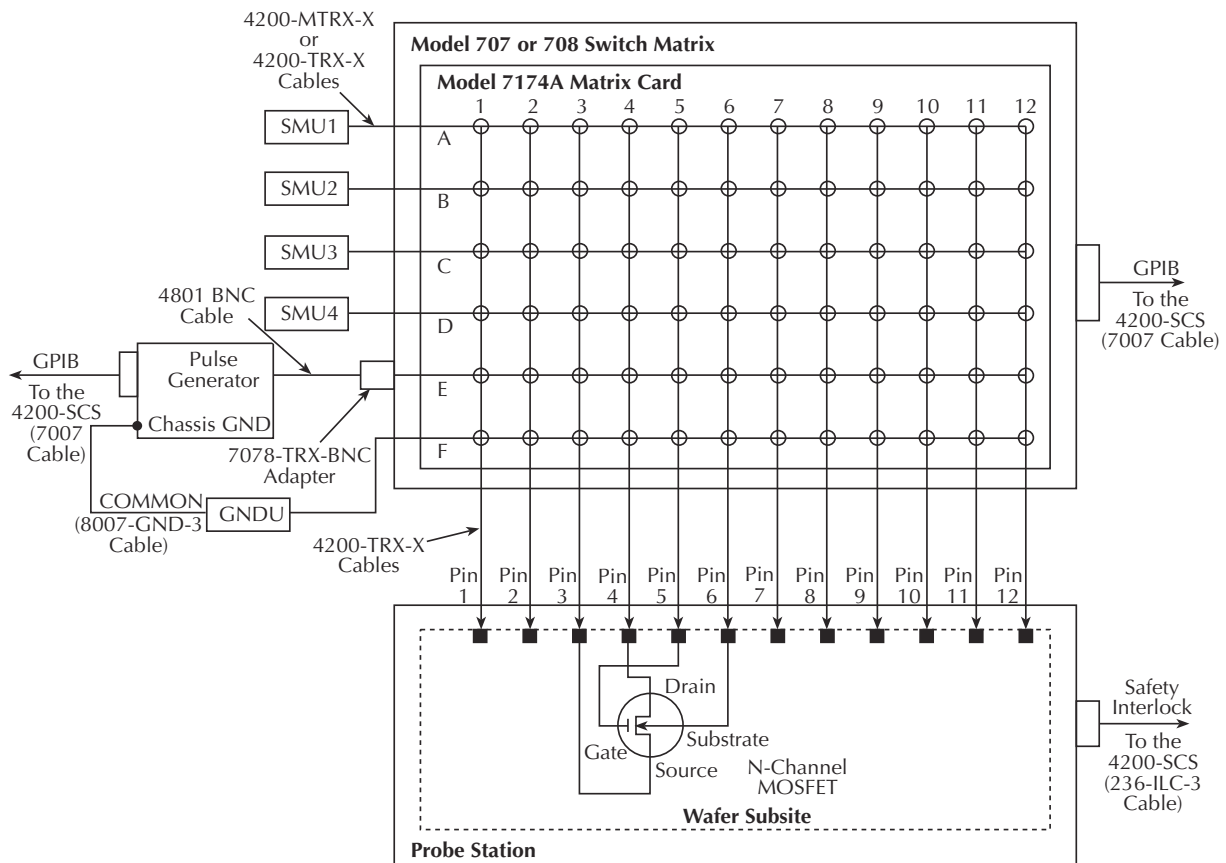


## Test system connections

A typical test system for this application is shown in [Figure 4-38](#). As shown, the Model 4200-SCS, HP Model 8110A/81110A pulse generator (PGU), and the DUT are connected to the Model 7174A Low Current Matrix Card. UTMs are used to control the switch matrix and the PGU. For details about SMU, GNDU, and matrix card connections, refer to the [Reference Manual, Appendix B, Using Switch Matrices](#).

The Model 7174A matrix card is installed in the Model 707/707A or Model 708/708A Switching Matrix. The switch matrix and PGU are controlled through the GPIB. Use the Model 7007 GPIB cables to connect the switch matrix and PGU to the Model 4200-SCS. For details about GPIB connections, refer to the [Reference Manual, Appendix B, GPIB connections](#).

Figure 4-38  
Test system for “ivpgswitch” project



## KCON setup

For this tutorial, a Hewlett Packard Model 8110A/81110A pulse generator, Keithley Instruments Model 707A Switching Matrix, Keithley Instruments Model 7174A Low Current Matrix Card, and a test fixture must be added to the system configuration.

KCON is used to add external equipment to the test system. Follow the steps below to add these components to the system configuration.

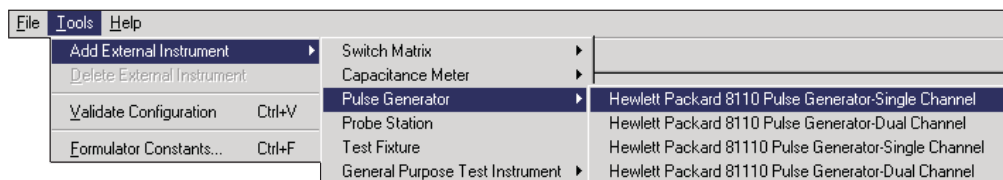
Detailed information about KCON can be found in the [Reference Manual Keithley CONFIGuration Utility \(KCON\)](#), page 7-1.

**To setup KCON:**



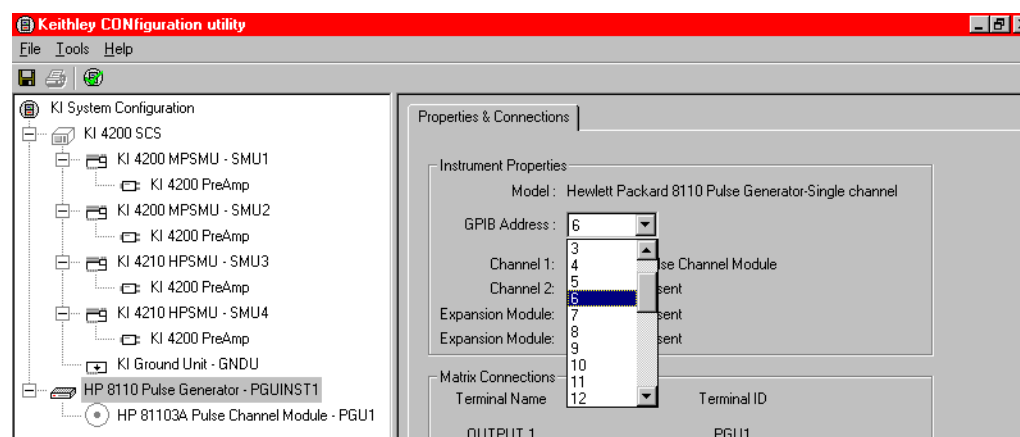
1. Start KCON. Double-click the **KCON** icon or use the **Start** menu, **Start > Programs > Keithley > KCON**.
2. Add the Hewlett Packard Model 8110A/81110A Pulse Generator to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-39](#).

Figure 4-39  
Adding a pulse generator



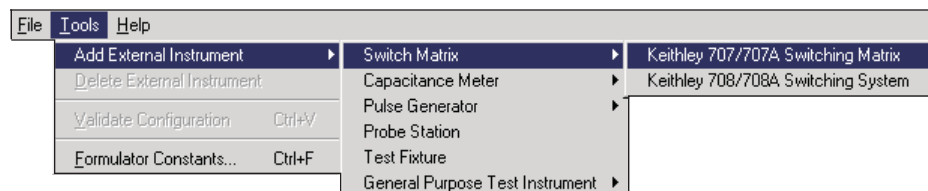
3. Set the GPIB Address for the pulse generator by selecting it in the configuration navigator and entering the appropriate GPIB Address on the Properties & Connections tab. This is illustrated in [Figure 4-40](#).

Figure 4-40  
Pulse generator configuration



4. Add the **Keithley Model 707/707A Switching Matrix** to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-41](#).

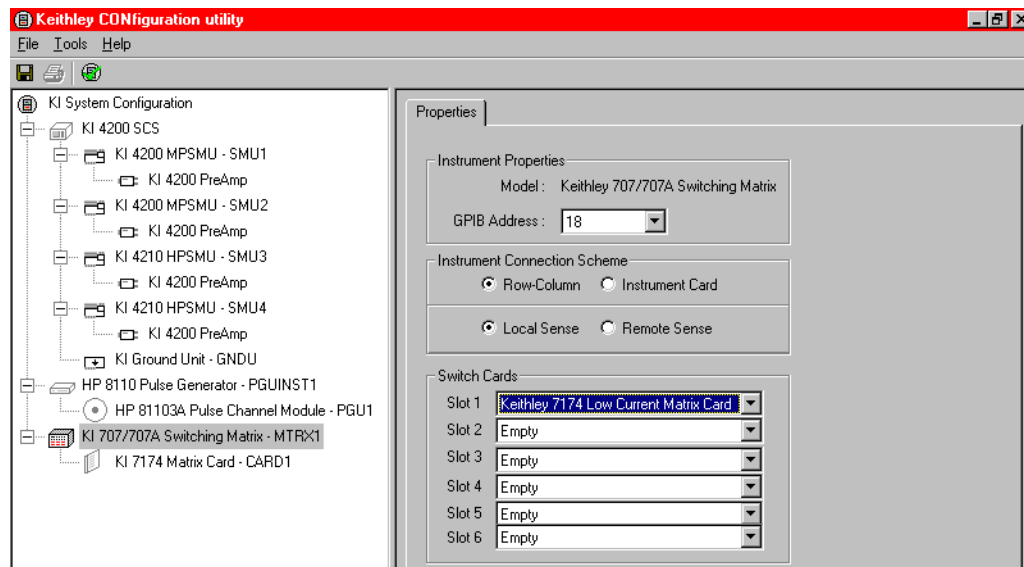
Figure 4-41  
Adding a switch matrix



5. Set the GPIB Address for the switch matrix and add the Model 7174A matrix card in Slot 1 as illustrated in [Figure 4-42](#).

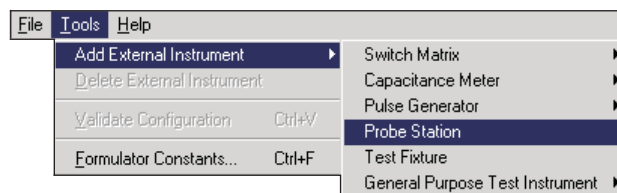


Figure 4-42  
Configuring the switch matrix



6. Add a manual probe station to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-43](#). If a test fixture is already part of the configuration, it must be removed before the probe station can be added. To remove any external component from the system configuration, select it in the configuration navigator and press the **DELETE** key.

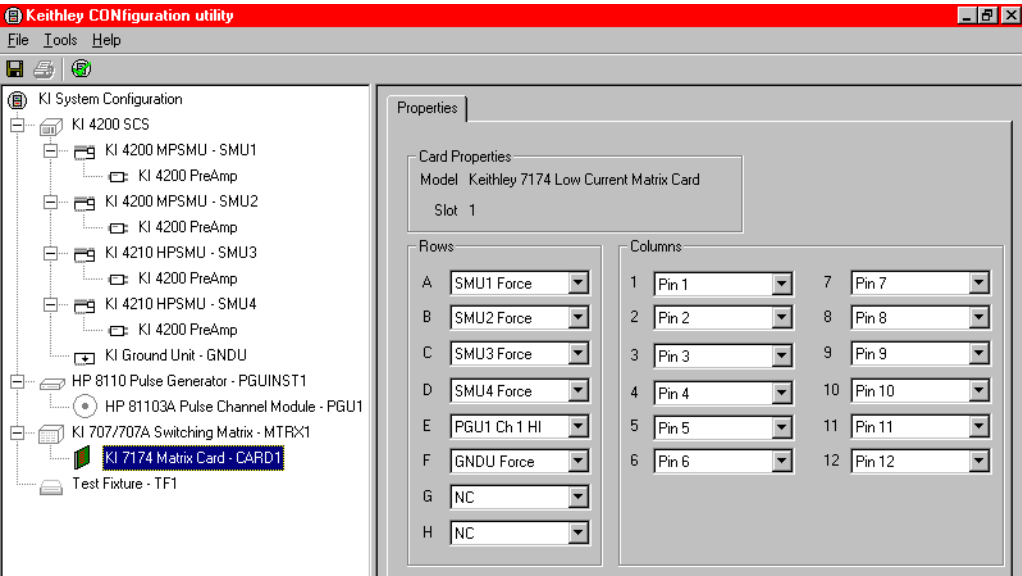
Figure 4-43  
Adding a probe station



7. Connect the instrument terminals and probe station pins to the switch matrix by selecting the **KI 7174 Matrix Card - CARD1** in the configuration navigator and configuring it as illustrated in [Figure 4-44](#). Detailed information about switch matrix configuration can be found in the [Reference Manual, Appendix B, Using Switch Matrices](#).

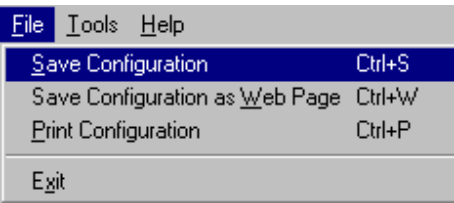


Figure 4-44  
Connecting the switch matrix



8. **Save** the configuration using the KCON **File** menu, as illustrated in [Figure 4-45](#).

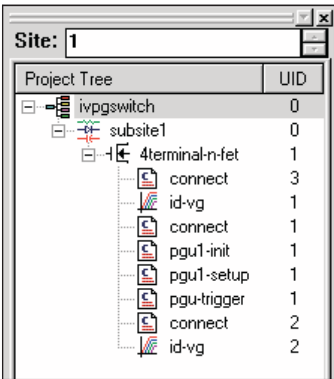
Figure 4-45  
Saving the system configuration



Open the ivpgswitch project

Open the `ivpgswitch` project from the **File** menu (select **Open Project**). The project navigator for the `ivpgswitch` project is shown in [Figure 4-46](#).

Figure 4-46  
Project navigator: `ivpgswitch` project





Description of tests

First connect test

The first test, connect, is a UTM that connects the device to the four SMUs. In the project navigator, double-click the first **connect** UTM to open it. [Figure 4-47](#) shows the parameters that connect the device to the SMUs.

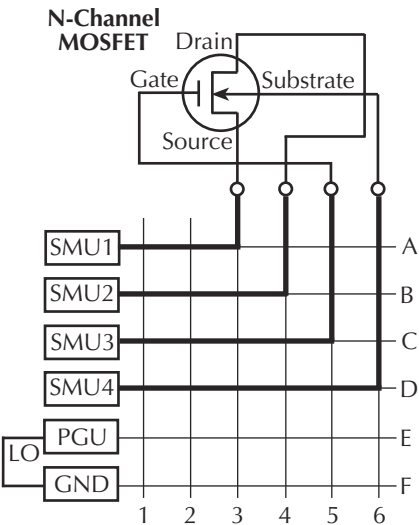
**NOTE** The first parameter (line 1) opens any relays that may have been closed by a previous test.

For the other parameter shown in [Figure 4-47](#), the device connects to the SMUs as shown in [Figure 4-48](#). For details about the connect UTM, refer to the [Reference Manual, Appendix B, Using Switch Matrices](#).

Figure 4-47  
First connect test: Connects the device to the SMUs

	Name	In/Out	Type	Value
1	OpenAll	Input	INT	1 ← Opens all relays
2	TermIdStr1	Input	CHAR_P	SMU1 ← Connects SMU1 to pin 3 of test fixture
3	Pin1	Input	INT	3
4	TermIdStr2	Input	CHAR_P	SMU2 ← Connects SMU2 to pin 4 of test fixture
5	Pin2	Input	INT	4
6	TermIdStr3	Input	CHAR_P	SMU3 ← Connects SMU3 to pin 5 of test fixture
7	Pin3	Input	INT	5
8	TermIdStr4	Input	CHAR_P	SMU4 ← Connects SMU4 to pin 6 of test fixture
9	Pin4	Input	INT	6

Figure 4-48  
Signal paths for the pre- and post-stress tests



First id-vg test

The **id-vg** ITM measures the transfer characteristics of the N-channel MOSFET. The  $I_D$  vs.  $V_G$  data points are graphed. The test also calculates and graphs transconductance. This is the **before-stress** characterization test.



Second connect test

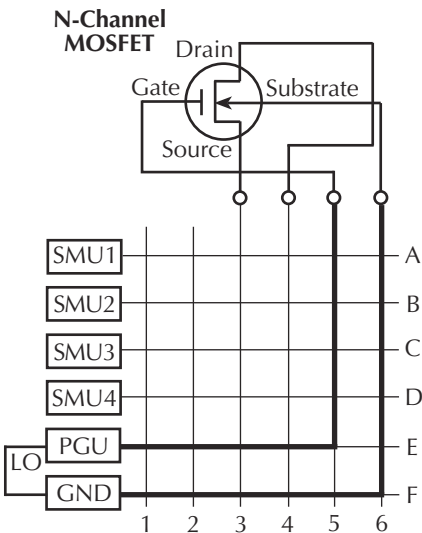
This connect UTM connects the device to the PGU and the GNDU. In the project navigator, double-click the second **connect** test to open it. [Figure 4-49](#) shows the parameters that connect the device to the PGU. Not shown is line 1 (OpenAll) that opens the relays closed by the previous connect test. Line 1 is shown in [Figure 4-47](#).

For the parameters shown in [Figure 4-49](#), the device connection pathways to the PGU and GNDU are shown in [Figure 4-50](#). Remember that if your physical matrix connections are different, you will have to change the connection parameters in the UTM to match them.

Figure 4-49  
Second connect test - connects the device to the PGU

	Name	In/Out	Type	Value
10	TermIdStr5	Input	CHAR_P	PGU ← Connects PGU to pin 5 of test fixture
11	Pin5	Input	INT	5
12	TermIdStr6	Input	CHAR_P	CMTR1
13	Pin6	Input	INT	0
14	TermIdStr7	Input	CHAR_P	CMTR1L
15	Pin7	Input	INT	0
16	TermIdStr8	Input	CHAR_P	GNDU ← Connects GNDU to pin 6 of test fixture
17	Pin8	Input	INT	6

Figure 4-50  
Signal paths to apply the pulse stress



pgu1-init test

In the project navigator, double-click **pgu1-init** to open the test. This one parameter test (see [Figure 4-51](#)) initializes the PGU. For example, it disables the output, resets errors, and sets triggering. More information about the initialized state is provided in the description area of the definition tab. For details about the UTMs for the pulse generator, refer to the [Reference Manual](#), [Appendix F, Pulse generator test example](#).

Figure 4-51  
PGU initialization

	Name	In/Out	Type	Value
1	InstIdStr	Input	CHAR_P	PGU1 ← Initializes HP 8110

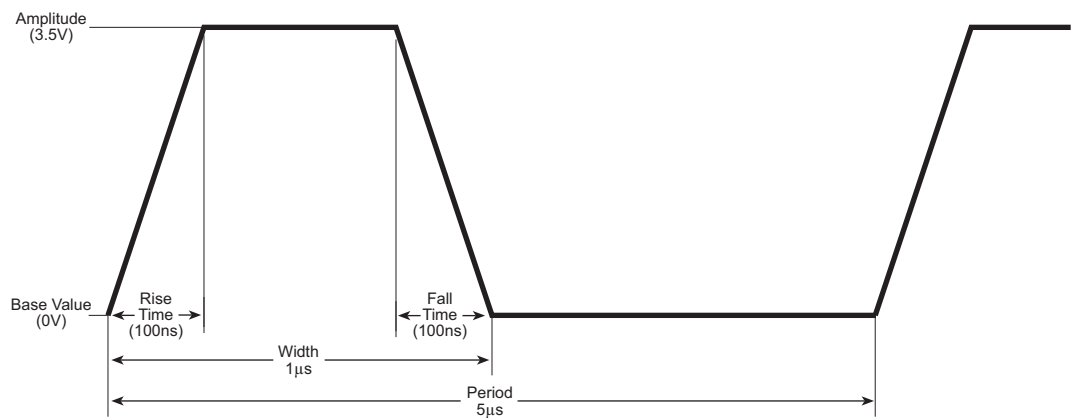


**pgu1-setup test**

In the project navigator, double-click **pgu1-setup** to open the test. The complete parameter listing for the test is shown in [Figure 4-52](#). These parameters to configure the PGU are explained in the description area of the definition tab.

[Figure 4-52](#) shows the pulse that is configured by this test.

Figure 4-52  
**PGU stress pulse specifications**



**NOTE** The pulse is not drawn to scale.

**pgu-trigger test**

In the project navigator, double-click **pgu-trigger** to open the test. The two-line parameter list for this test is shown in [Figure 4-53](#). This test triggers the PGU to output 60,000 pulses to the N-channel MOSFET.

Figure 4-53  
**pgu-trigger test: Trigger the burst of stress pulses**

	Name	In/Out	Type	Value
1	InstIdStr	Input	CHAR_P	PGU1
2	Count	Input	INT	60000 ← <i>Triggers burst of pulses</i>

**Third connect test**

This connect test is the same as the first connect test. It connects the device to the SMUs so that the transfer characteristics can be determined after applying the pulse stress (see [Figure 4-47](#) and [Figure 4-48](#)).

**Second id-vg test**

This *id-vg* test is the same as the first *id-vg* test. It measures the transfer characteristics of the N-channel MOSFET. This is the after-stress characterization test.



## Running the test sequence


To run the test sequence, select (click) the **4terminal-n-fet** device in the project navigator, and then click the green **Run** button . The test sequence is summarized in [Table 4-5](#).

Table 4-5

**Test sequence for ivpgswitch project**

	Test	Description
1	connect	Connects the MOSFET to the four SMUs.
2	id-vg	Measures the initial transfer characteristics of the MOSFET.
3	connect	Connects the MOSFET to the PGU.
4	pgul-init	Initializes the PGU.
5	pgul-setup	Configures the PGU output pulse.
6	pgu-trigger	Triggers the PGU to output a burst of pulses.
7	connect	Connects the MOSFET to the four SMUs.
8	id-vg	Measures the final transfer characteristics of the MOSFET.

## Compare the test results

A way to compare `id-vg` test results is to do a side-by-side visual inspection of the two graphs. In the project navigator, double-click the two **id-vg** tests to open them in the Workspace.

### To compare the test results:

1. **Close some UTMs:** To reduce clutter, you may want to remove any other tests (UTMs) from the Workspace. [Figure 4-54](#) shows the button to close a displayed test.
2. **Make room for the two graphs:**
  - a. Hide the project navigator to expand the size of the Workspace.
  - b. Reduce the size of the test documents.

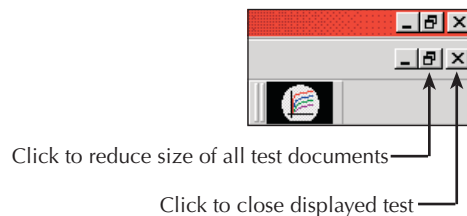
---

**NOTE** The close button (X) is located at the top right corner of the project navigator. [Figure 4-54](#) shows the button to reduce the size of the test documents in the workspace.

---

Figure 4-54

**Buttons to close or reduce size of test documents**

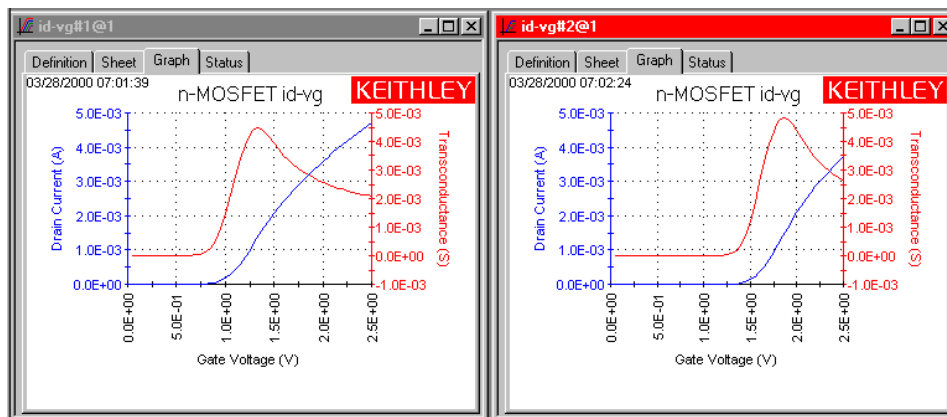


### To close or reduce size of test documents:

1. **Position tests side-by-side:** A test document is moved by clicking the title bar at the top of the document and dragging it to the desired location in the workspace.
2. **Display the graphs:** The graph for each test is displayed by clicking the **Graph** tab. [Figure 4-55](#) shows typical graphs for the two `id-vg` tests.



Figure 4-55  
id-vg graphs



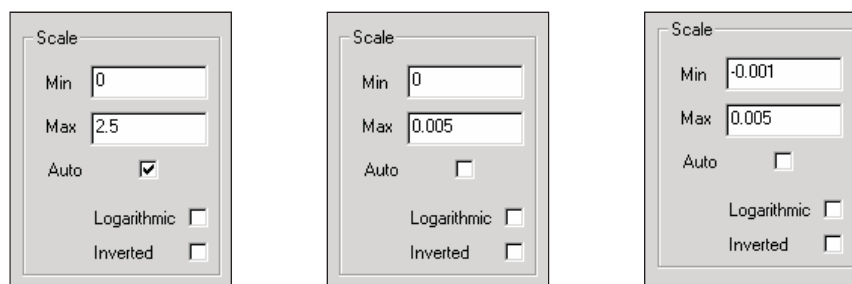
A. Before-stress graph

B. After-stress graph

3. **Scale settings:** To effectively compare the two graphs, they must both have the same scale settings. Figure 4-56 shows the scale settings for the graphs in Figure 4-55.
  - Scale settings for a graph are set by clicking the axis properties item in the graph menu.
  - A graph menu is displayed by placing the mouse pointer in an open area of the graph, and then right-clicking the mouse.

**NOTE** Remember that there is a separate graph menu (and axis properties window) for each graph.

Figure 4-56  
Graph scale settings



A. X-axis

B. Y-axis

C. Y2-axis

4. **Compare graphs:** Visually inspect the two graphs for differences caused by the stress. You can also click the **Sheet** tabs and compare the data collected for the two tests.

## Overlaying graphs

To compare the two graphs by laying the after-stress graph over the before-stress graph:

1. For the after-stress test, click the **Sheet** tab to display the data spreadsheet.
2. Select all five columns by clicking and dragging the mouse pointer from Column **A** through Column **E**. Press **Ctrl + C** to copy those columns.
3. For the before-stress test, click the **Sheet** tab, and then the **Calc** tab (located at the bottom).



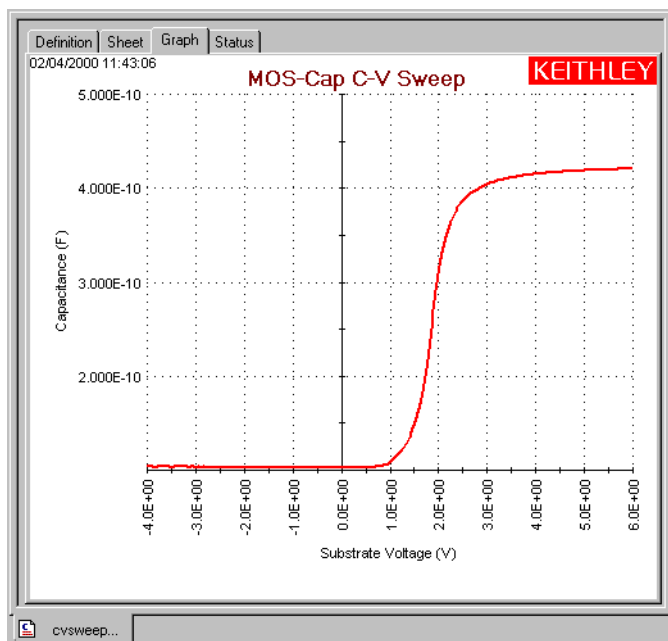
4. In the Calc spreadsheet, click cell **A1** to select it, and then press **Ctrl + V**. This pastes the copied columns into the Calc spreadsheet. This after-stress data is now available to be graphed.
5. In the Calc spreadsheet, rename the **DrainI** and **GM** columns to distinguish them as after-stress (AS) data. For example, change **DrainI** to **DrainI(AS)**, and change **GM** to **GM(AS)**.
6. Click the **Graph** tab for the present test (before-stress). In an open area of the graph, right-click the mouse to open the graph menu. In the graph menu, click **Define Graph** to open the graph definition window.
7. In the graph definition window, click the **Y1/DrainI(AS)** cell and the **Y2/GM(AS)** cell to select them, and click **Ok**. The graph will now show the overlaid data.
8. From the graph menu, use the **Legend** and **Graph Properties - Series** items to add a legend and to change the line properties of the graph, if desired.

## How to control an external CV analyzer

This tutorial demonstrates how to control a Keithley Model 590 CV Analyzer to acquire capacitance versus voltage (CV) data from a MOS capacitor. This tutorial also demonstrates how to create a new KITE project. The new project will contain one UTM that is connected to a standard CV user module supplied with each Model 4200-SCS.

The CV Analyzer will apply a linear staircase voltage sweep to a capacitor. A capacitance measurement will be performed on every voltage step of the sweep. [Figure 4-57](#) shows a typical CV curve generated by this test.

Figure 4-57  
Typical CV curve



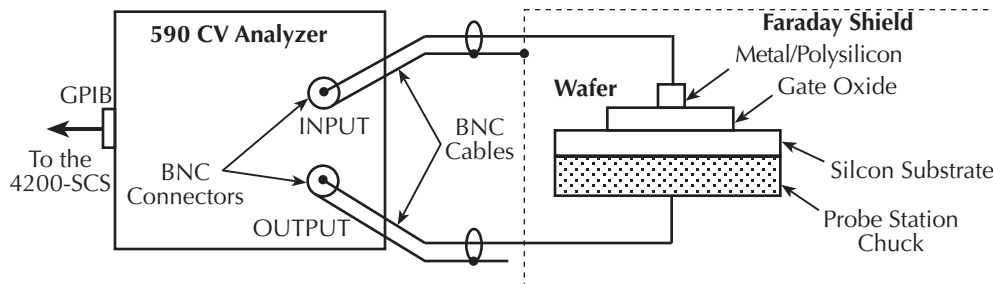
## Connections

Connection details for the Model 590 CV Analyzer are provided in the [Reference Manual, Appendix C, Using a Keithley Instruments Model 590 CV Analyzer](#). The INPUT and OUTPUT connectors of the Model 590 are connected to the capacitor using Model 4801 (RG-58) BNC



cables. The Model 590 is controlled by the Model 4200-SCS through the GPIB bus. Use a Model 7007 GPIB cable to connect the Model 590 to the Model 4200-SCS. [Figure 4-58](#) provides an illustration of these connections.

Figure 4-58

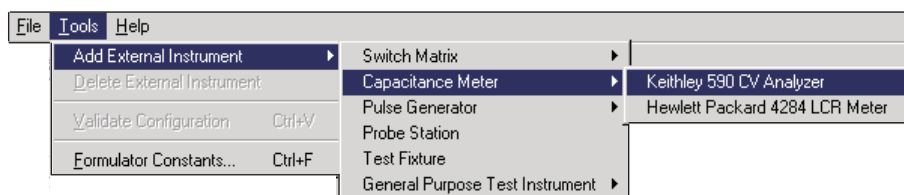
**Keithley Model 590 CV Analyzer DUT connections****KCON setup**

For this tutorial, the Model 590 CV Analyzer must be included in the Model 4200-SCS system configuration. KCON is used to add external equipment to the test system. For details about KCON, refer to the [Reference Manual Keithley CONFIGURATION Utility \(KCON\)](#), page 7-1.

**To add the Model 590 to the system configuration using KCON:**

1. Start KCON. Double-click the **KCON** icon or use the **Start** menu, **Start > Programs > Keithley > KCON**.
2. Add the Keithley Instruments Model 590 CV Analyzer to the system configuration using the **KCON Tools** menu as illustrated in [Figure 4-59](#).

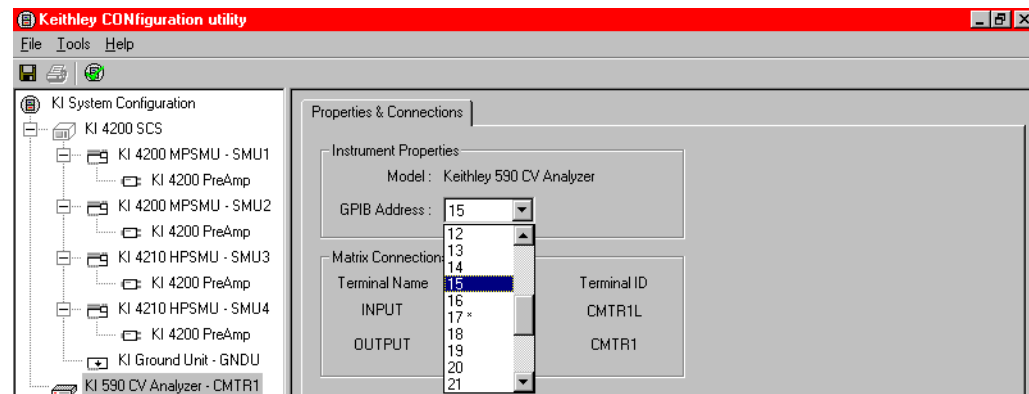
Figure 4-59

**Adding a Keithley 590 CV Analyzer to the system configuration**

3. Set the GPIB address for the Model 590 by selecting the **KI 590 CV Analyzer - CMTR1** in the configuration navigator and entering the appropriate GPIB address on the Properties & Connections tab. This is illustrated in [Figure 4-60](#).

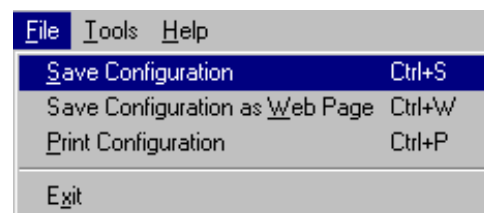


Figure 4-60  
Setting the Model 590 GPIB address



4. **Save** the configuration using the KCON **File** menu as illustrated in [Figure 4-61](#).

Figure 4-61  
Saving the system configuration

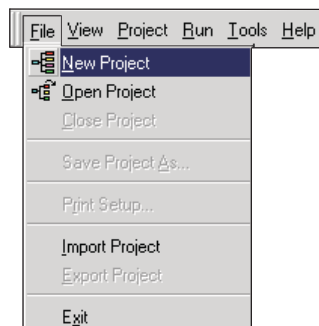


## Create a new project

To create a new project:

1. On the KITE toolbar, select **New Project** from the **File** menu (see [Figure 4-62](#)) to open the define new project window. The new project definition window is shown in [Figure 4-63A](#).

Figure 4-62  
New project menu selection



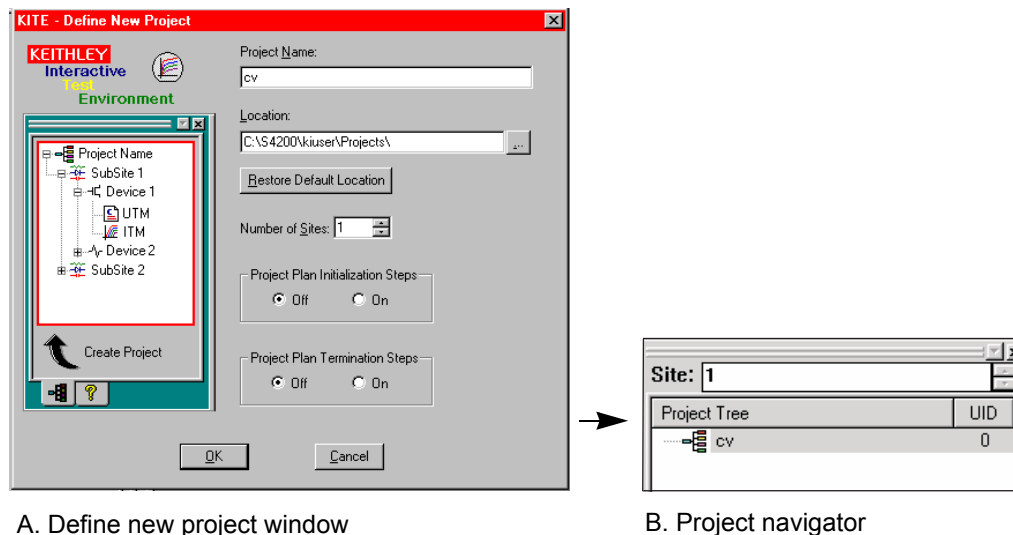
2. Type in the name of the project (**cv**) and define it as shown in [Figure 4-63A](#). The directory path shown in the location box is the default location where the factory defined projects are located. Make sure the specified number of sites is **1**, and initialization and termination steps are **Off**.



- With the project defined as shown in [Figure 4-63A](#), click the **Ok** button at the bottom of the window. The project name will appear in the project navigator as shown in [Figure 4-63B](#).

**NOTE** For details about creating a project, refer to the [Reference Manual Building, modifying, and deleting a Project Plan, page 6-47](#).

Figure 4-63

**Define new project**

A. Define new project window

B. Project navigator

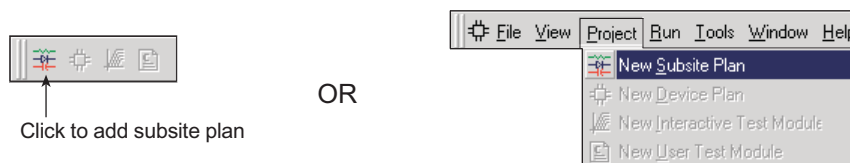
**Add a subsite plan**

A subsite, or test element group, is a collection of devices to be tested.

**To add a subsite plan:**

- Open the **Add New Subsite Plan to Project** window by clicking the **Add new Subsite Plan** button on the toolbar (see [Figure 4-64A](#)), or click the **New Subsite Plan** item on the project menu (see [Figure 4-64B](#)).

Figure 4-64

**Add a new subsite plan to a KITE project**

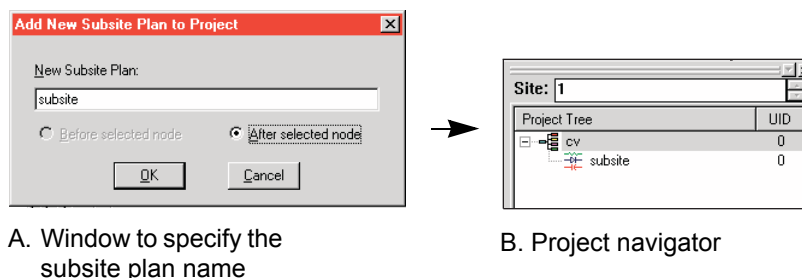
A. Add new Subsite Plan button

B. Project menu

- With the add new subsite plan to project window open (see [Figure 4-65A](#)), type in the name **subsite**, and click **Ok**. The subsite plan appears in the project navigator as shown in [Figure 4-65B](#).



Figure 4-65  
Add a new subsite plan

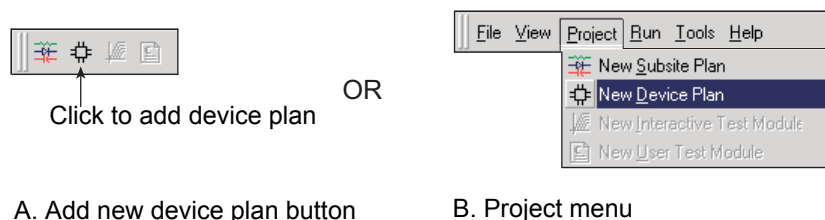


## Add a device plan

To add a device plan:

1. A device plan is a collection of tests to be performed on a particular device. **Open the Add New Device Plan to Project window** by clicking the **Add new Device Plan** button on the toolbar (see Figure 4-66A). It can also be opened by clicking the **New Device Plan** item on the **Project** menu (see Figure 4-66B).

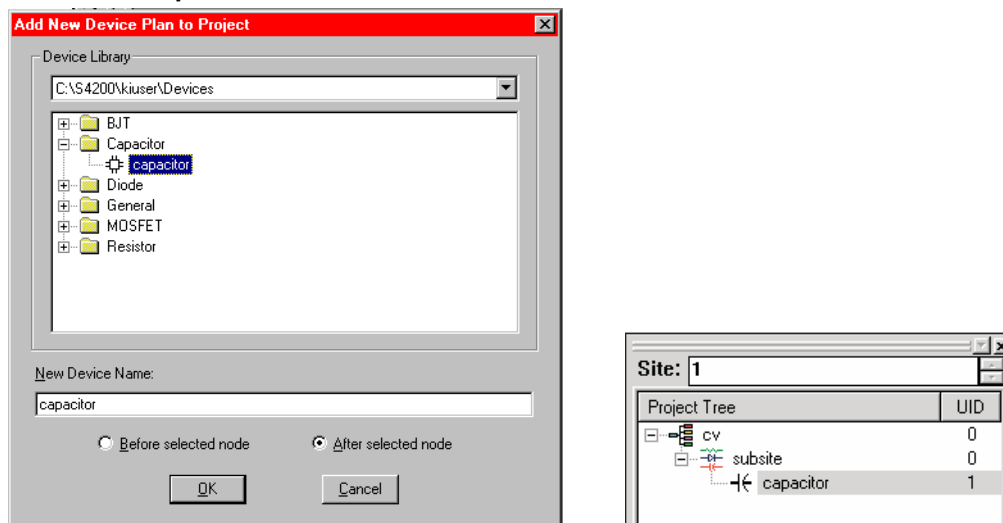
Figure 4-66  
Add a new device plan to a KITE project



2. In the window to add a device plan (Figure 4-67A), double-click the **Capacitor** folder to open it, and then click **Capacitor** to select that device plan name.
3. With the capacitor device plan selected, as shown in Figure 4-67A, click **Ok** at the bottom of the window. The device plan will appear in the project navigator as shown in Figure 4-67B.



Figure 4-67

**Add a device plan**

A. Window to specify device plan name

B. Project Navigator

**Add a UTM**

To add the **cvsweep** UTM to the new project by copying it from the default test library (C:\S4200\kiuser\tests):

1. In the project navigator, double-click the **Capacitor** device to open the device plan window.
2. On the sequence tab of the device plan window, use the **Test Library** pull-down menu to select the default test library as shown in [Figure 4-68](#).
3. Double-click the **Capacitor** folder to open it and display the available tests for that device. [Figure 4-69A](#) shows the Capacitor folder opened.
4. For the Capacitor folder, click **cvsweep** to select it. [Figure 4-69](#) shows cvsweep selected.
5. Click **Copy** to place the test in the test sequence table. [Figure 4-69](#) shows cvsweep copied into the test sequence table.
6. At the bottom of the device plan window, click **Apply** to copy the test into the project navigator. [Figure 4-69B](#) shows the cvsweep UTM added to the project.
7. If desired, the device plan window may be closed by pressing the close (X) button. The close (X) button is located on the right, above the device plan window.

Figure 4-68

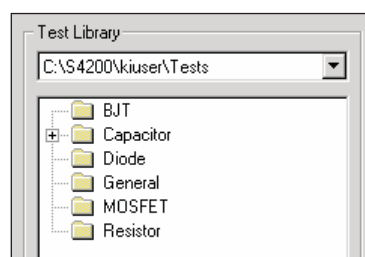
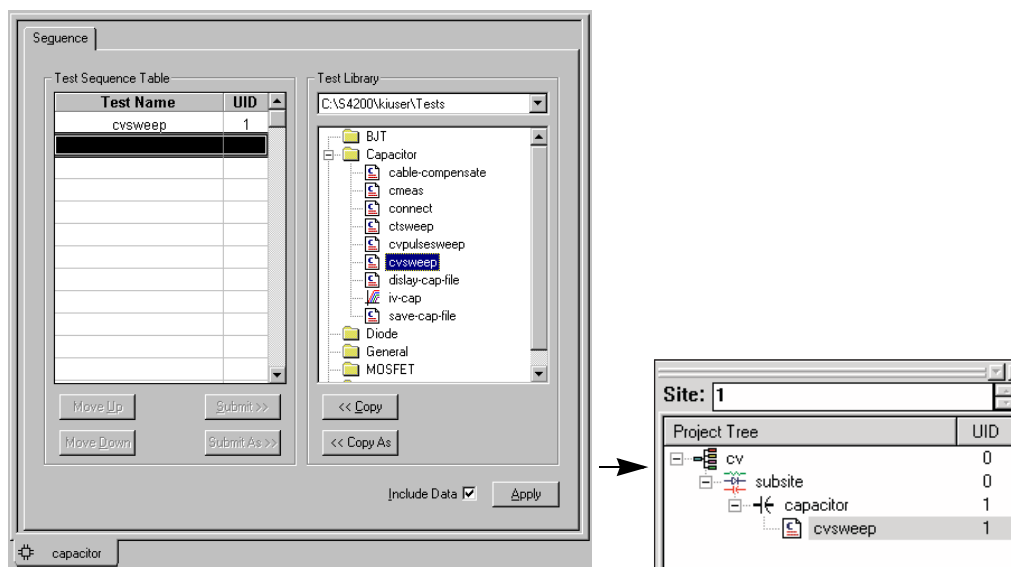
**Default test library folders**



Figure 4-69  
Add the **cvswEEP** UTM



A. Window to specify the test module name

B. Project navigator

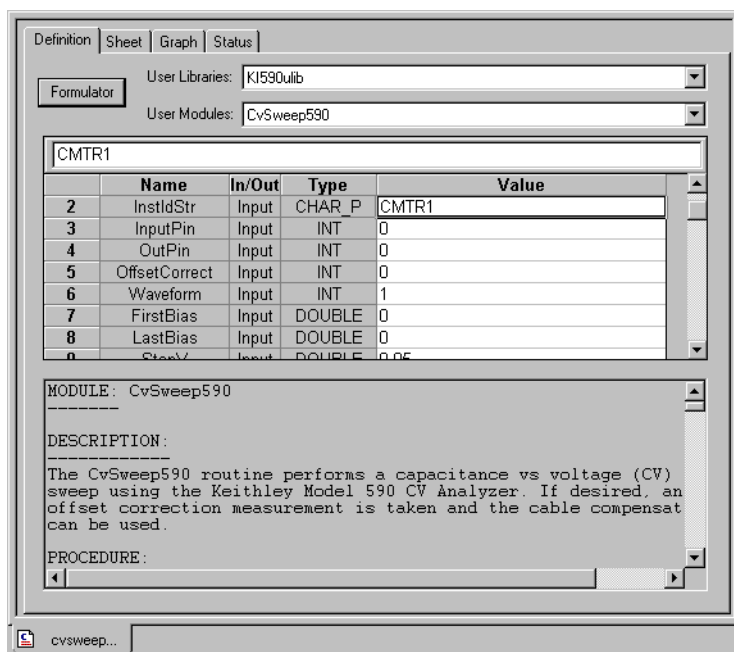
## Modifying the **cvswEEP** UTM

The default **cvswEEP** parameters will sweep the voltage from -4 V to +6 V. If these parameters are acceptable, proceed to [Executing the test on page 4-44](#).

To modify the parameters:

1. In the project navigator, double-click the **cvswEEP** UTM to open it. The window in [Figure 4-70](#) will be displayed:

Figure 4-70  
**cvswEEP** UTM





2. Click the **Definition** tab and make the desired parameter changes to the test.

---

**NOTE** For details about the cvsweep UTM, refer to the [Reference Manual, Appendix C, Open and execute cvsweep UTM](#).

---

## Executing the test

Since this new project has only one subsite plan and only one device plan, the test can be run from any level in the project navigator. To run the cvsweep test, click the green **Run** button. After the test is finished, use the sheet and graph tabs to view and analyze the results.

---

**NOTE** The Model 4200-SCS also supports the Keithley Instruments Model 595 Quasistatic C-V Meter and the Keithley Instruments Model 82-WIN Simultaneous C-V System.

For more information, refer to the [Reference Manual KI595 CV Analyzer Properties and Connections tab, page 7-24 and Appendix E, Using a Keithley Model 82 C-V System](#).

---

## What if my equipment is not listed in KCON

A typical test setup often involves several instruments, performing sourcing, measuring, or auxiliary functions, all connected to a common communications bus (typically GPIB) and controlled by a PC station. The Keithley Model 4200-SCS parameter analyzer eliminated the need for a dedicated PC. Its interactive test environment, KITE, allows users to use the Model 4200-SCS both as a parameter analyzer and an external instrument controller, making it a “command-and-control center” of the entire instrument rack. KITE software already supports pulse generators, switch matrices, and CV analyzers, through software control modules known as drivers. Occasionally, a user may need to control an instrument that is not supported by a standard Keithley Instruments driver library.

To learn more about creating external instrument drivers for the Model 4200-SCS refer to the Keithley Instruments Technotes No. 2661-1005 located on the Model 4200 Complete Reference.



---

## How to Generate Basic Pulses

In this section:

Topic	Page
Keithley Pulse Application (KPulse) .....	5-2
KPulse: Getting started .....	5-2
Starting KPulse .....	5-2
KPulse setup and help .....	5-3
Triggering .....	5-3
Standard pulse waveforms .....	5-4
Segment Arb™ waveforms .....	5-6
Exporting Segment Arb™ waveform files .....	5-8
Custom file arb waveforms (full-arb) .....	5-8
Waveform types .....	5-12



# Keithley Pulse Application (KPulse)

## KPulse: Getting started

Keithley Pulse Application (KPulse) is a graphical user interface (GUI) that is a non-programming alternative to configure and control the installed Keithley pulse cards. It is used for quick tests requiring minimal interaction with other Model 4200-SCS test resources.

The Kpulse application supports the source-only configuration of Models 4225-PMU, 4220-PGU, 4205-PG2, and 4200-PG2 2-channel pulse cards. The Model 4225-PMU is identified as PMU on the card tab, whereas all other card types are identified as VPU.

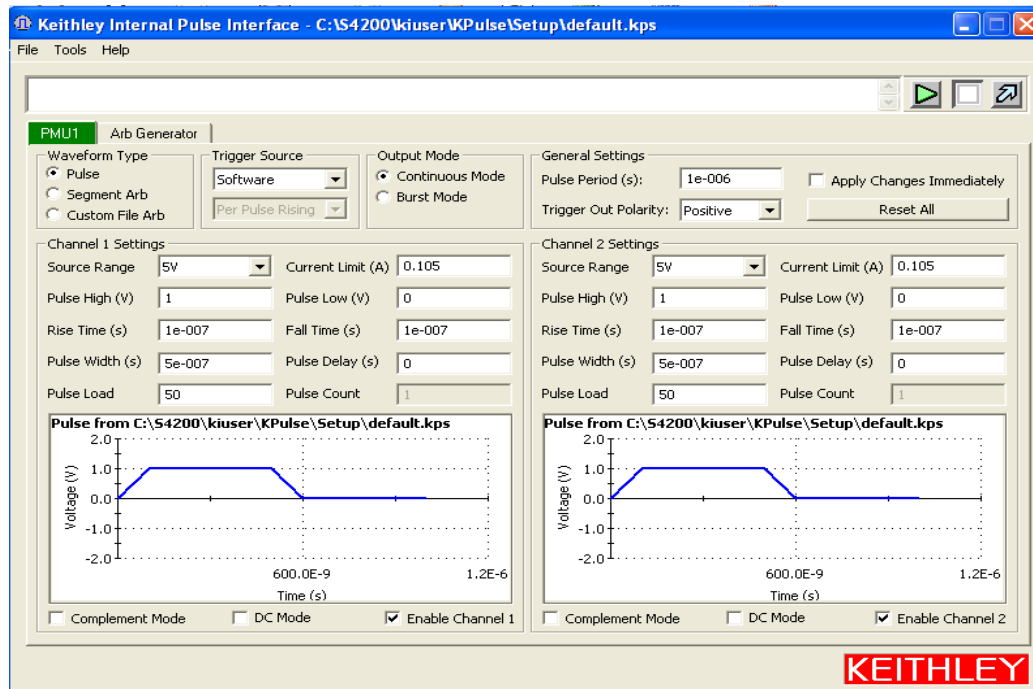
## Starting KPulse

The KPulse GUI (Figure 5-1) is opened by double-clicking the **KPulse** icon on the desktop. The GUI example shows one PMU installed in the system.

From the GUI, each pulse generator can be used to configure and control the following waveform types:

- **Standard pulse waveforms** and **Segment Arb™ waveforms**: Pulses are configured and run from the VPU or PMU tabs of KPulse. There is a tab for every Keithley pulse card installed in the Model 4200-SCS.
- **Custom file arb waveforms (full-arb)**: Pulses are configured and saved as a .kaf file using the Arb Generator tab of KPulse. A VPU or PMU tab can then be used to load the saved .kaf file into the pulse generator and run it.

Figure 5-1  
KPulse GUI



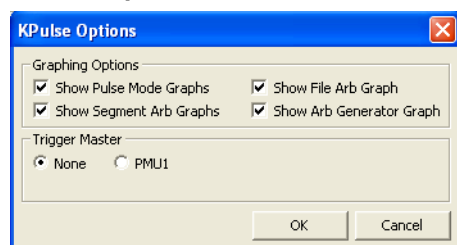


## KPulse setup and help

The following menus are located at the top-left corner of KPulse:

- File** Use this menu to load/save KPulse setups and exit KPulse. By default, setup files are saved at the following command path location: C:\S4200\kiuser\KPulse\Setup.
- Tools** From this menu, click **Options** to open the **KPulse Options** dialog box (see [Figure 5-2](#)):
- **Graphing Options:** The following pulse preview graphs can be disabled or enabled:
    - **Show Pulse Mode Graphs** When enabled, shows the Standard Pulse waveform previewers for each PG2 tab.
    - **Show Segment Arb™ Graphs** When enabled, shows the Segment Arb™ pulse waveform previewers for each PG2 tab.
    - **Show File Arb Graph** When enabled, shows the Custom File Arb pulse waveform previewer for each PG2 tab.
    - **Show Arb Generator Graph** When enabled, shows the Arb Generator pulse waveform previewer.
    - **Trigger Master** Select the PG2 that will serve as the trigger master, or select **None** if you are not using a trigger master (see the [Reference Manual, Triggering, page 5-3](#)).
- Help** Use this menu to access Model 4200-SCS Complete Reference information, and to open the About KPulse dialog box.

Figure 5-2  
KPulse options



## Triggering

With a Keithley pulse card selected as the trigger master, its Trigger Out can be used to start (trigger) itself or other PG2s in the system. For details about using a pulse card as the trigger master, see the [Reference Manual, Pulse output synchronization, page 11-29](#).

---

**NOTE** For the master pulse card, the polarity of the pulse trigger source (*pulse\_trig\_source*) and pulse trigger polarity (*pulse\_trig\_polarity*) function must be the same. If you are using a rising-edge trigger source, the pulse trigger polarity must be positive. If you are using a falling-edge trigger source, the pulse trigger polarity must be negative. This requirement applies to all three pulse modes (Standard Pulse, Segment Arb™, and FARB).

---



---

**NOTE** When triggering multiple pulse cards in a master/slave configuration, changing the master card's trigger output polarity will result in a transition in the trigger output levels that may be interpreted as a trigger pulse by the other cards.

---



## Standard pulse waveforms

Standard pulse waveforms are configured and controlled from the pulse card tabs in the GUI. [Figure 5-3](#) explains how to use the GUI for Standard Pulse output.

**Standard pulse waveform previewers:** KPulse provides a preview of configured standard pulse waveforms for each enabled channel. Each waveform previewer shows the high and low levels, and timing for the waveform.

In [Figure 5-3](#), the configuration shown in the waveform previewer for Channel 1 uses the default settings for KPulse (pulse high = 1 V and pulse low = 0 V). Channel 2 uses the same settings, but the complement mode is enabled. Pulse high goes to pulse low level (0 V) and pulse low goes to pulse high level (1 V).



Figure 5-3  
Standard pulse operation

- 1) Click the pulse card tab.
- 2) Select **Pulse** to configure the Standard pulse **Waveform Type**.
- 3) **Enable Channel 1** and/or **Enable Channel 2** – A channel must be enabled in order to preview its waveform and turn on its output.
- 4) Configure triggers for both channels of the pulse card:
  - **Trigger Source** – **Software**, **External** or **Internal Bus**: With **External** selected, select the trigger source: **Initial Falling**, **Initial Rising**, **Per Pulse Falling** or **Per Pulse Rising**.
  - **Output Mode** – Select the output trigger mode: **Continuous Mode** or **Burst Mode**.
- 5) Configure the **General Settings** for both channels of the pulse card:
  - Set the **Pulse Period** in seconds.
  - Set the **Trigger Polarity**: **Positive** or **Negative**.
  - Select **Apply Changes Immediately** to enable automatic update for pulse output. After outputs are turned on (step 9), pulse output updates immediately when settings are changed.  
OR  
Click the **Apply Settings** button to manually apply settings. This button is disabled when **Apply Changes Immediately** is enabled.
  - Clicking **Reset All** returns the pulse card to the Standard Pulse waveform type and its default settings. It also updates the previewer.
- 6) Configure the **Channel 1 Settings** and/or **Channel 2 Settings**. The Pulse Count field is active if the Burst Mode is the selected trigger mode.
- 7) *Optional* – DC Mode and Complement Mode:
  - With the **DC Mode** selected, the output will be fixed DC at the Pulse High level. Disabling DC Mode returns the output to the previously defined pulse.
  - Enable the **Complement Mode** to set pulse high to the low level, and pulse low to the high level.
- 8) To configure other installed pulse cards for Standard Pulse, click on the tab for the desired pulse card and repeat steps 1 through 7.
- 9) Turn on all enabled channels – Click the green triangle to turn on enabled channels for all pulse cards installed in the Model 4200-SCS. With the output on, the square box will turn red. Clicking the red box turns off the outputs.



## Segment Arb™ waveforms

Segment Arb™ waveforms are configured and controlled from the PG2 tabs in the GUI. [Figure 5-4](#) explains how to use the GUI for standard pulse output.

---

**NOTE** *Due to the Segment Arb™ engine overhead, there is an additional 10 ns interval added to the end of the last segment of a Segment Arb™ waveform. During this interval, the output voltage, solid-state relay control (high endurance output relay (HEOR)), and trigger output values remain the same as the final value reached in the last segment.*

---

### Start, stop, and time restrictions:

- The start level of the first segment and the stop level of the last segment must be the same. In [Figure 5-4](#), Segment 1 start and Segment 7 stop are both set for 0.0 V.
- The stop level for a segment must be the same as the start level for the next segment. In [Figure 5-4](#), the stop level for Segment 1 is 1.0 V, which is the same as start level for Segment 2 (no discontinuities are allowed).
- Time values are in 10 ns increments, with a minimum of 20 ns.

**Segment Arb™ pulse waveform previewers:** KPulse provides a preview of configured Segment Arb™ waveforms for each enabled channel. Each waveform previewer shows the segment levels and timing for the waveform.



Figure 5-4  
Segment Arb™ operation

1) Click the tab for the pulse card to be configured.

2) Select **Segment Arb** to configure the Segment-Arb pulse **Waveform Type**.

3) **Enable Channel 1** and/or **Enable Channel 2** – A channel must be enabled in order to preview its waveform and turn on its output.

4) Configure triggers for both channels of the pulse card:

- **Trigger Source** – **Software**, **External** or **Internal Bus**: With **External** selected, select the trigger source: **Initial Falling**, **Initial Rising**, **Per Pulse Falling** or **Per Pulse Rising**.
- **Output Mode** – Select the output trigger mode: **Continuous Mode** or **Burst Mode**.

5) Configure the **General Settings** for both channels of the pulse card:

- Set the **Trigger Polarity**: **Positive** or **Negative**.
- Select **Apply Changes Immediately** to automatically apply settings and update the previewer.  
OR  
Click the **Apply Settings** button to manually apply settings and update the previewer. This button is disabled when **Apply Changes Immediately** is enabled.
- Clicking **Reset All** returns the pulse card to the Standard Pulse waveform type and its default settings. It also updates the previewer.

6) Configure the **Channel 1 Settings** and/or **Channel 2 Settings**:

- Set the **Source Range** (volts), **Current Limit** (amps) and **Pulse Load** (ohms). If the trigger mode is set to **Burst Mode**, set the **Pulse Count**.
- In the table, enter the **Start** voltage, **Stop** voltage, **Time** (in seconds), TTL output **Trigger** level (0 = low, 1 = high) and the state of the **SSR** (Solid State Relay) (0 = open, 1 = closed).

7) To configure other installed pulse cards for Segment-Arb, repeat Steps 1 through 6.

8) Turn on all enabled channels – Click the green triangle to turn on enabled channels for all pulse installed in the Model 4200-SCS. With the output on, the square box will turn red. Clicking the red box turns off the outputs.

**NOTE** The output trigger levels are not shown in the waveform previewers.



## Exporting Segment Arb™ waveform files

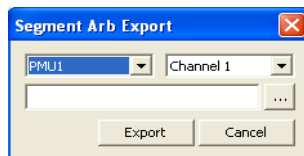
After configuring a Segment Arb™ waveforms in KPulse, it can be saved as a .ksf file. SARB .ksf files should be exported into the SarbFiles folder at the following command path:

C:\S4200\kiuser\KPulse\SarbFiles

### To export a Segment Arb™ waveform file:

1. At the top-left corner of KPulse, click **Tools**, and then click **Export Segment Arb™** to open the Segment Arb™ Export dialog box (see [Figure 5-5](#)).
2. In the Segment Arb™ Export dialog box, select the **PG2** and channel for the waveform to be exported.
3. In the Segment Arb™ Export dialog box, use the file navigation button (...) to locate the target folder, and type a name for the file. The .ksf extension will be added automatically.

Figure 5-5  
Segment Arb™ Export dialog box

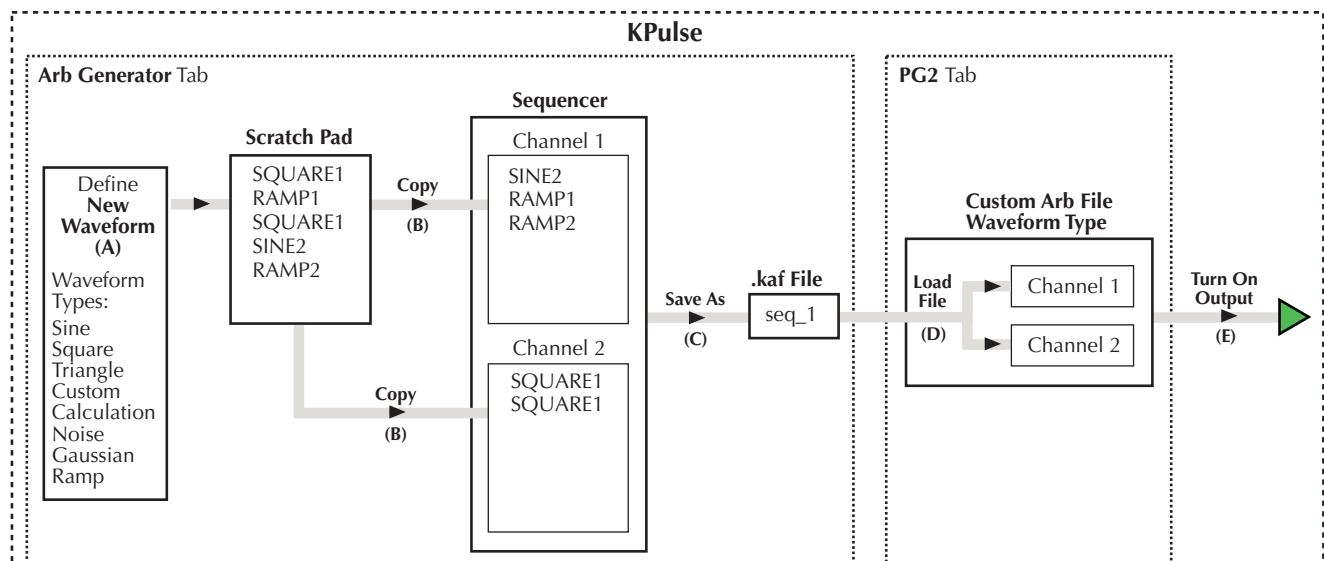


A saved seg\_arb.ksf waveform file can be imported back into the pulse generator card using the [seg\\_arb\\_file](#) function. For Segment Arb™ stress/measure testing, the .ksf file can be imported using the KITE Device Stress Properties dialog box shown in [Model 4200-SCS Reference Manual, Section 6, Figure 6-393](#). For details about Segment Arb™ stress/measure testing, see the [Reference Manual, Segment stress/measure mode, page 6-323](#).

## Custom file arb waveforms (full-arb)

[Figure 5-6](#) summarizes the basic processes to create a custom full-arb waveform file, to load the file into a pulse card, and to output the pulse waveforms.

Figure 5-6  
Basic process to create and output custom file Arb waveforms

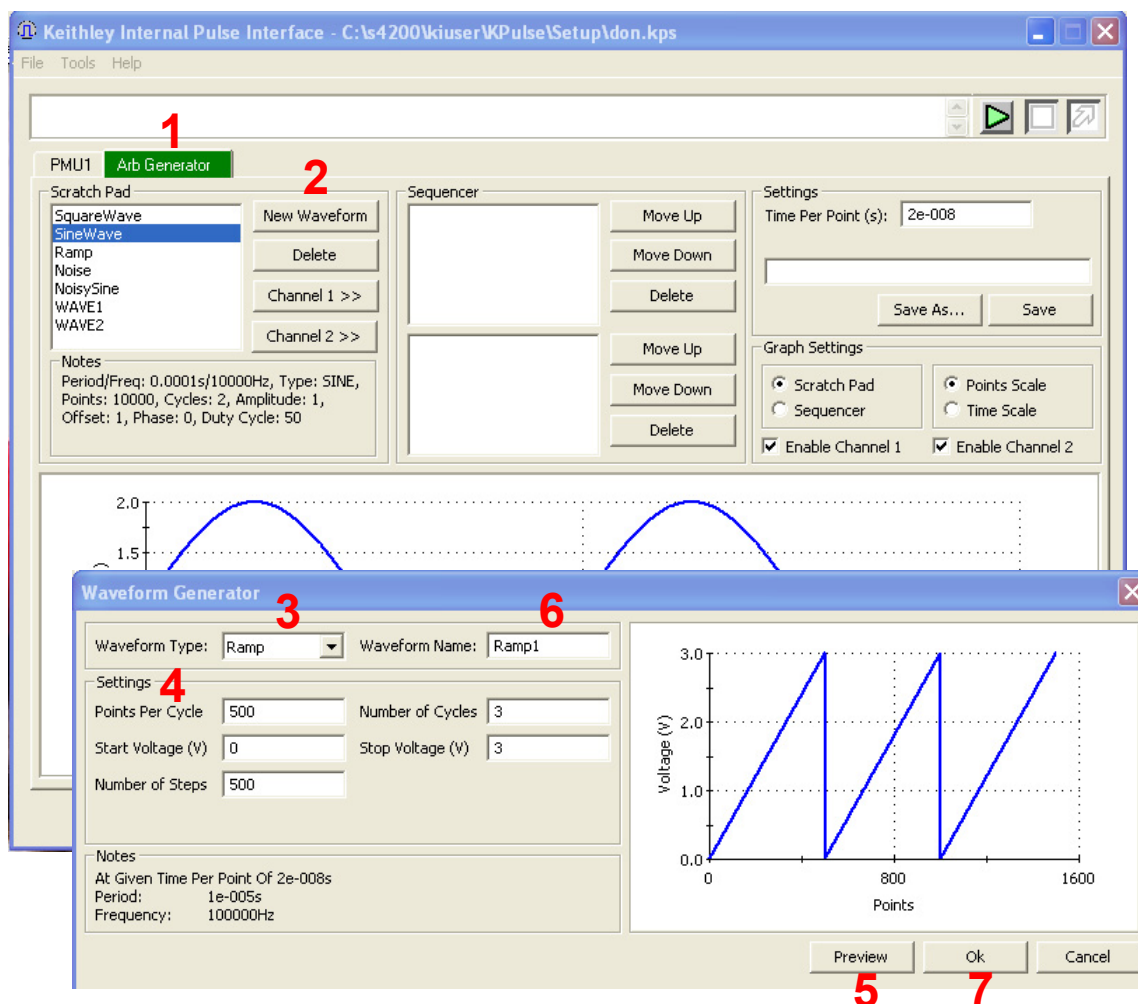




**To create custom file arb waveforms (full-arb):**

- A. Select and configure waveforms:
  - After selecting an available waveform type, configuring its settings, and naming it, the waveform is placed in the Scratch Pad.
  - Waveforms will remain in the Scratch Pad until they are deleted by the user.
  - Refer to the [Waveform types on page 5-12](#) for information about the waveform types available for custom file arb. Refer to [Figure 5-7](#) and the steps that follow for details.
- B. Copy the waveforms into the Sequencer for Channel 1, Channel 2, or both:
  - The order that two or more waveforms appear in a channel sequencer is the order that the waveforms will be output by that channel.
  - Refer to [Figure 5-8](#) and the steps that follow for details.
- C. Save the waveforms in the Sequencer as a .kaf file. See step 5 for [Figure 5-8](#).
- D. Load the .kaf waveform file into a pulse generator (using the appropriate PG2 tab): Refer to [Figure 5-9](#) and the steps that follow for details.
- E. Turn on the output for enabled channels. See step 7 for [Figure 5-9](#).

Figure 5-7

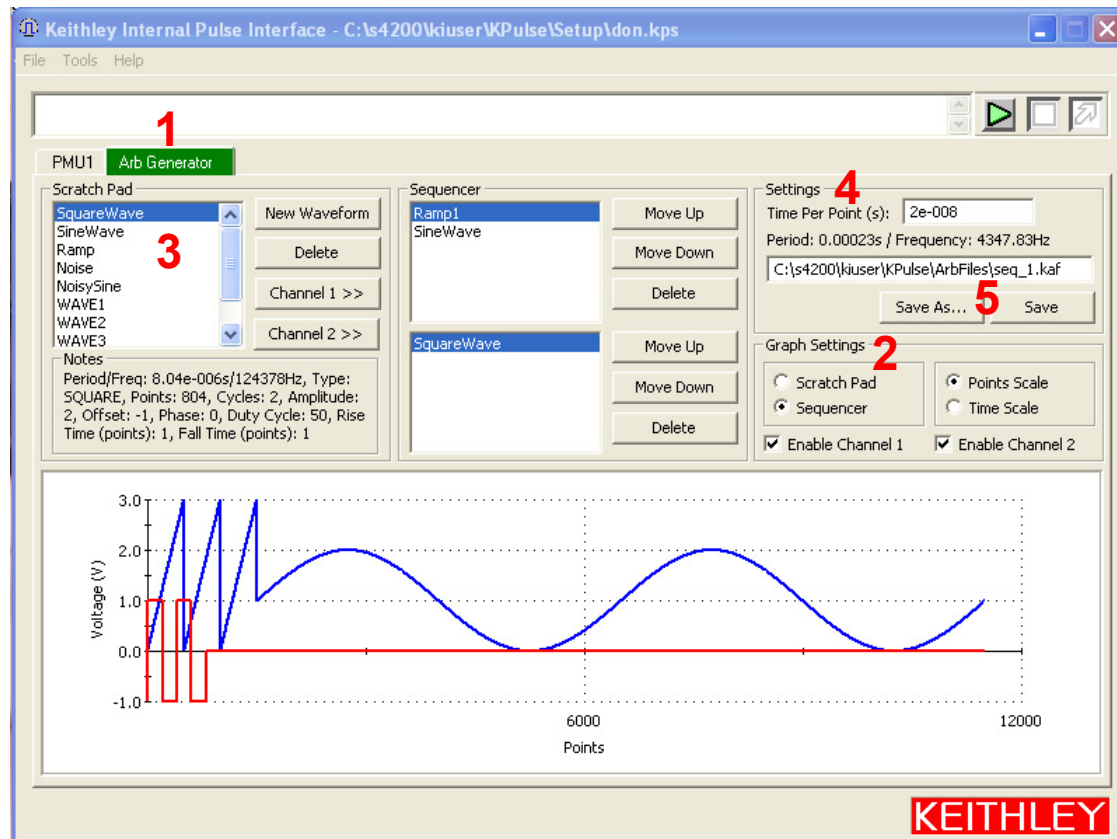
**Custom Arb file operation: Select and configure waveforms**

- 1) Click the **Arb Generator** tab.
- 2) Click **New Waveform** to open the Waveform Generator window.
- 3) Use the drop-down menu to select the **Waveform Type** to be created.



- 4) Configure the **Settings** for the selected waveform type.
- 5) Click **Preview** to update the preview of the waveform.
- 6) Type in a name for the waveform. You cannot use a name that is already used in the Scratch Pad.
- 7) Click **OK** to create the waveform. The new waveform will be added to the Scratch Pad. [Figure 5-8](#) shows the new waveform named "Ramp1" has been added to the Scratch Pad.
- 8) Repeat steps 2 through 7 to create another waveform in the Scratch Pad.

Figure 5-8

**Custom Arb file operation: Copy waveforms into Sequencer**

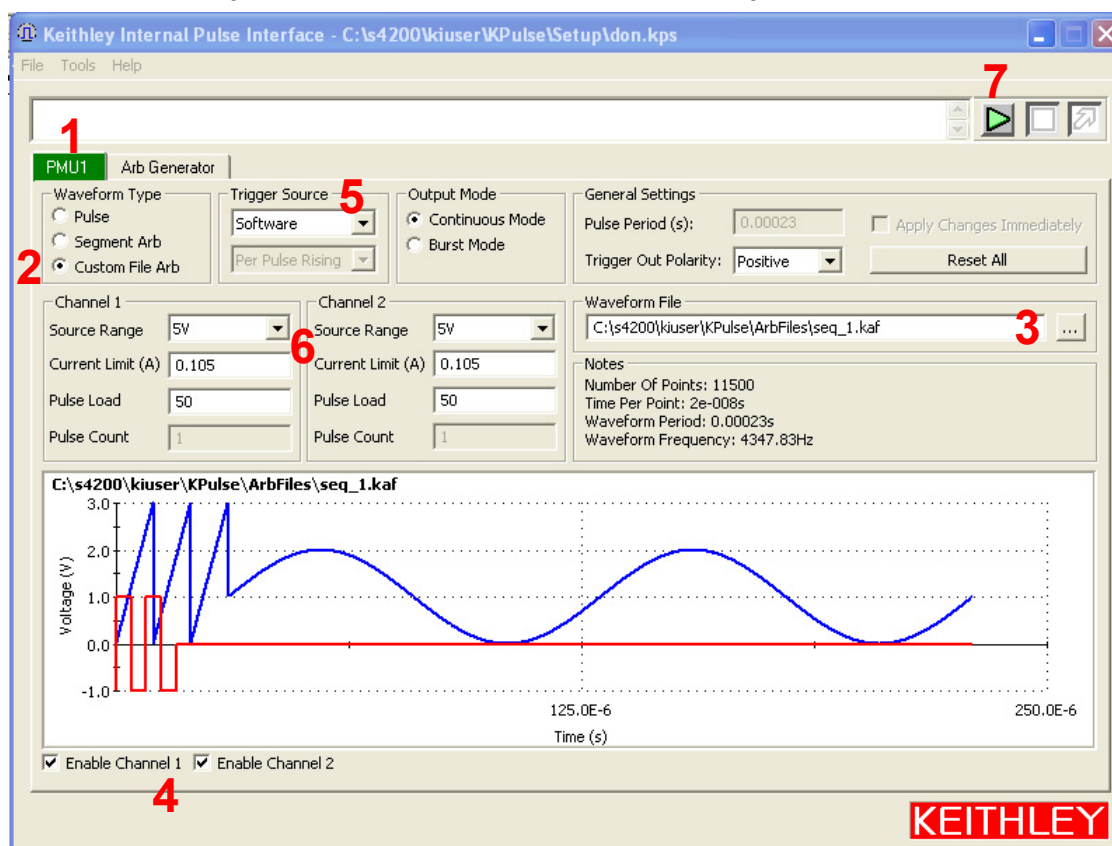
- 1) Click the **Arb Generator** tab.
- 2) Configure **Graph Settings** for the previewer:
  - Select **Scratchpad** or **Sequencer**:
    - Scratchpad previews the waveform that is selected in the Scratch Pad.
    - Sequencer previews enabled waveform sequences (see next bullet).
  - To preview the waveform(s) in the Sequencer, **Enable Channel 1** and/or **Enable Channel 2**.
  - Select the scale for the graph; **Points Scale** or **Time Scale**.
- 3) Copy Scratch Pad waveforms into the Sequencer:
  - a) In the Scratch Pad, click (select) a waveform to be copied into the Sequencer.
  - b) Click **Channel 1** to copy the selected waveform onto the Sequencer for Channel 1, and/or click **Channel 2** to copy the waveform into the Sequencer for Channel 2.
  - c) To copy another waveform into the Sequencer, repeat steps a and b.



- Changing the waveform sequence - The waveform sequence for each channel can be changed. After clicking (selecting) a waveform in the Sequencer, click **Move Up** or **Move Down**.
  - **Delete** buttons - After clicking (selecting) a waveform in the Scratch Pad or Sequencer, click the appropriate Delete button to remove it. Note that deleting a waveform from the Scratch Pad also removes it from the Sequencer.
- 4) Set the **Time per Point** (in seconds). This is the time interval between each point in the waveform(s).
  - 5) Save the waveform(s) as a Keithley Arb File (.kaf). By default, .kaf files are saved in a folder named "ArhFiles" at the following path: C: S4200\kiuser\KPulse\ArbFiles.
    - Use **Save As** to name the file and save it.
    - After any subsequent changes, click **Save** to overwrite the .kaf file.

Figure 5-9

**Custom Arb file operation: Load waveform and turn on output**



- 1) Click a pulse card tab.
- 2) Select **Custom File Arb**.
- 3) Click the **Waveform File** button and then select and load the the desired .kaf file.
- 4) **Enable Channel 1** and/or **Enable Channel 2** - The loaded .kaf file will consist of a waveform for one or both of the channels. If the .kaf file was saved with one or both channels enabled, the .kat tile will load into this tab with the same channels enabled. A channel must be enabled in order to preview and output its waveform. The waveform for Channel 1 is blue. and the waveform for Channel 2 is red.
- 5) Configure triggers for both chinnels of the pulse card:



- **Trigger Source** - **Software**, **External**, or **Internal Bus**. With **External** enabled, select the trigger source: **Initial Falling**, **Initial Rising**, **Per Pulse Falling**, or **Per Pulse Rising**.
  - **Output Mode** - Select the output trigger mode: **Continuous Mode** or **Burst Mode**.
- 6) Configure the **Channel 1 Settings** and/or **Channel 2 Settings**. the Pulse Count field is active if the Burst Mode is the selected trigger mode.
- Note: To configure other installed pulse cards for Custom File Arb, repeat Steps 1 through 6.
- 7) Turn on all enabled channels - Click the green triangle to turn on enabled channels for all installed pulse cards in the Model 4200-SCS. With the output on, the square box will turn red. Clicking the red box turns off the outputs of all pulse cards

## Waveform types

KPulse provides the following fundamental waveform types to use as the building blocks for custom file arb:

- [Sine waveform](#)
- [Square waveform](#)
- [Triangle waveform](#)
- [Custom waveform](#)
- [Calculation waveform](#)
- [Noise waveform](#)
- [Gaussian waveform](#)
- [Ramp waveform](#)
- [Sequences waveform](#)

As explained in [Figure 5-7](#), a waveform is created using the Waveform Generator. After selecting and configuring one of the above waveform types, the waveform is placed into the Scratch Pad.

---

**NOTE** The period for the waveforms is determined by the **Time Per Point** setting in the Arb Generator tab (step 4 in [Figure 5-8](#)).

---

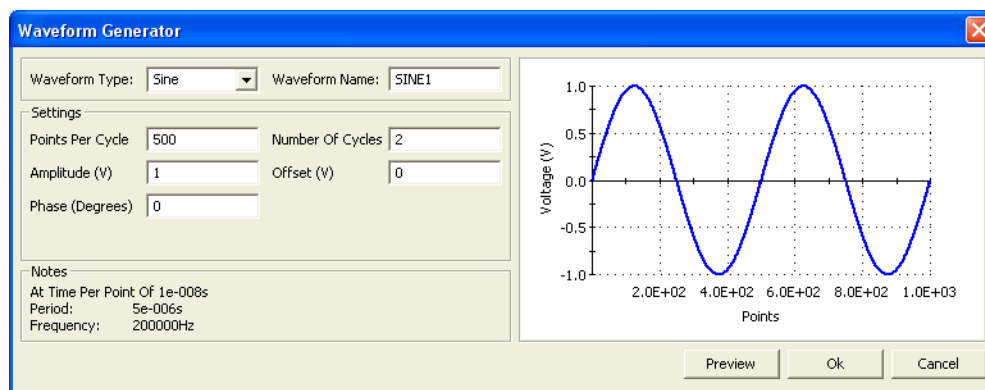
### Sine waveform

An example of a Sine waveform, using the default settings, is shown in [Figure 5-10](#). The waveform for this example is named SINE1, but can be any name that is not already used in the Scratch Pad.

After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.



Figure 5-10  
**Sine waveform (default settings)**

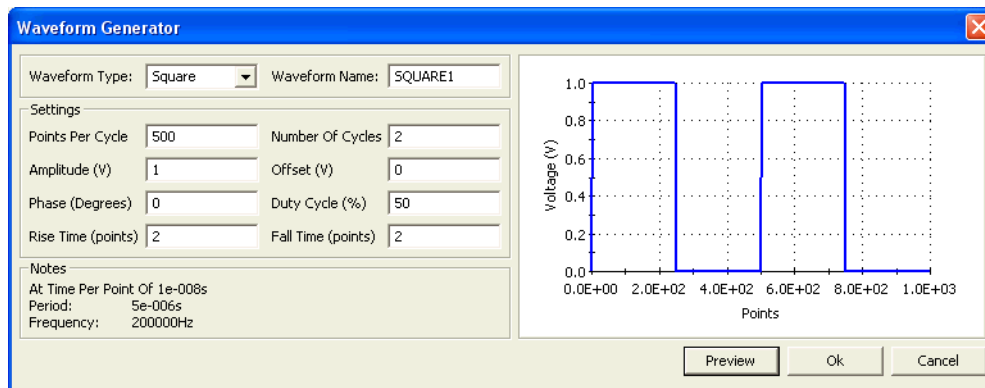


### Square waveform

An example of a square waveform, using the default settings, is shown in [Figure 5-11](#). The waveform for this example is named SQUARE1, but can be any name that is not already used in the Scratch Pad.

After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-11  
**Square waveform (default settings)**



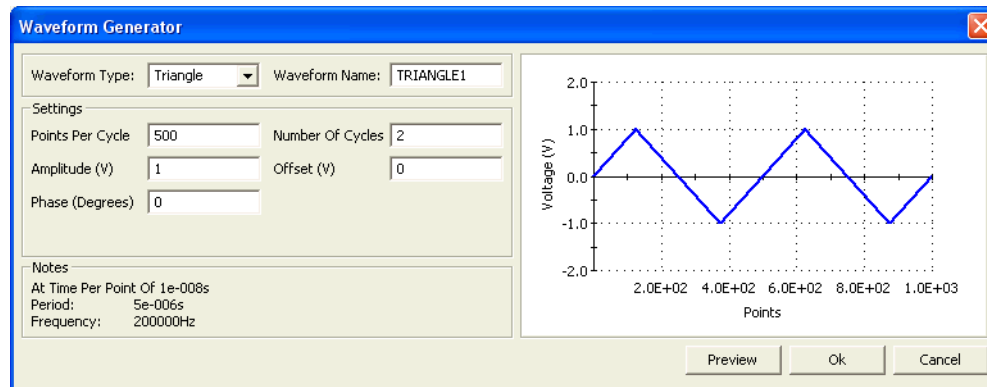
### Triangle waveform

An example of a triangle waveform, using the default settings, is shown in [Figure 5-12](#). The waveform for this example is named TRIANGLE1, but can be any name that is not already used in the Scratch Pad.

After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.



Figure 5-12  
Triangle waveform (default settings)



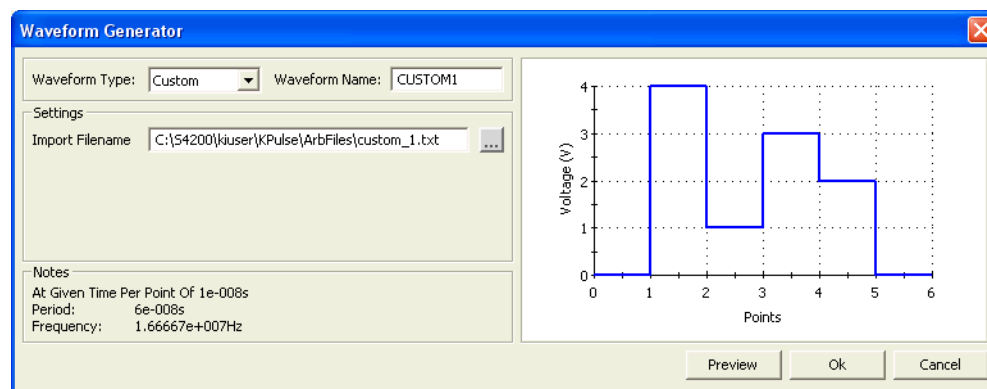
### Custom waveform

An example of a custom waveform is shown in Figure 5-13. The waveform for this example is named CUSTOM1, but can be any name that is not already used in the Scratch Pad.

The voltage values for the waveform are retrieved from an imported file (.txt or .csv). After creating a file (.txt or .csv) for the custom waveform, use **Import Filename** shown in Figure 5-13 to import the file into the Waveform Generator.

After importing the file, click **Preview** to show the waveform. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-13  
Custom waveform



### Creating a file (.txt or .csv) for custom waveform

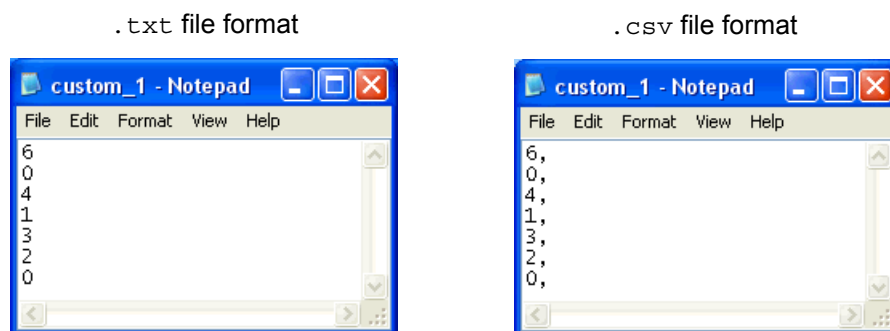
The waveform file is created using a text editor utility, such as Notepad.

#### To create the list of voltage points:

1. Open a text editor utility.
2. On the first line, type the number of voltage points in the waveform, and then type the list (one per line) of values for the waveform:
  - **.txt file format** As shown in Figure 5-14, commas are not used to separate values.
  - **.csv file format** As shown in Figure 5-14, commas are used to separate values. Only the first column of data is used for the waveform. Additional columns are ignored.



Figure 5-14  
Creating a .txt or .csv file for a custom waveform



3. The custom waveform in Figure 5-13 is a simple 6-point waveform made up of these voltage values: 0 V, 4 V, 1 V, 3 V, 2 V, 0 V. Those seven entries are shown in the text editors in Figure 5-14.

---

**NOTE** The time at each point is determined by the Time Per Point setting in the Arb Generator tab (Step 4 in Figure 5-8).

---

4. Save as a waveform file (.txt or .csv) in the ArbFiles folder at the following command path location:  
C:\S4200\kiuser\KPulse\ArbFiles

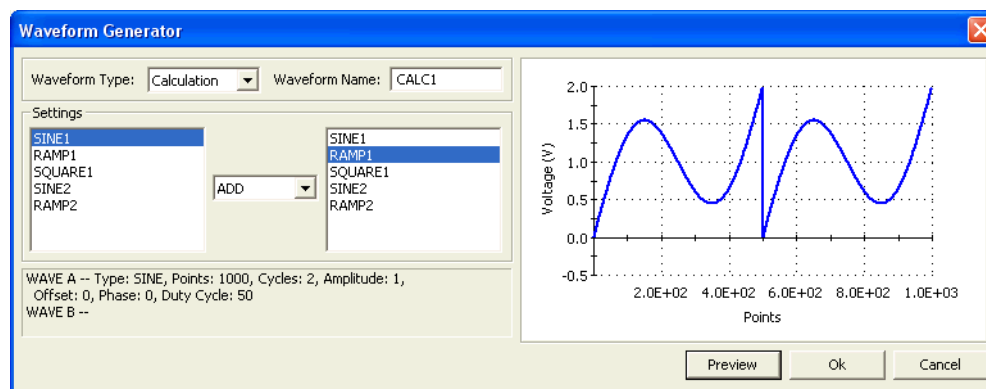
### Calculation waveform

An example of a calculation waveform is shown in Figure 5-15. The waveform for this example is named CALC1, but can be any name that is not already used in the Scratch Pad.

The calculation (add, subtract, multiple or divide) performs the selected math operation on two selected Scratch Pad waveforms. In Figure 5-15, SINE1 is added to RAMP1.

After selecting the two waveforms and the math operation, click **Preview** to display the result of the calculation. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-15  
Calculation waveform



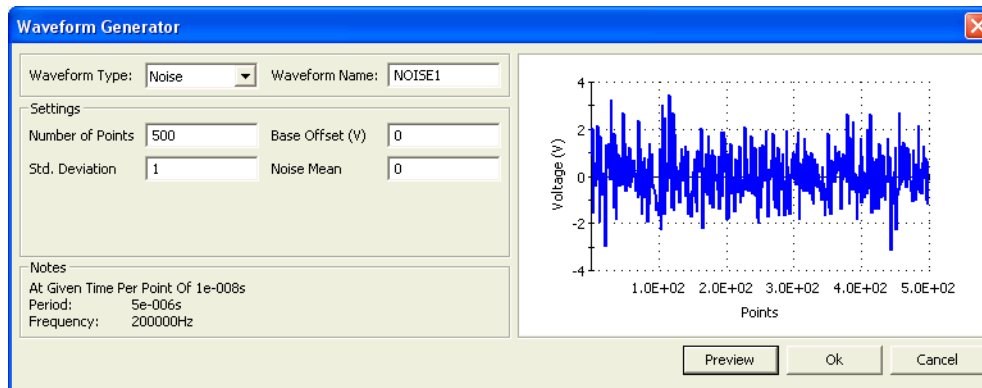
### Noise waveform

An example of a noise waveform, using the default settings, is shown in Figure 5-16. The waveform for this example is named NOISE1, but can be any name that is not already used in the Scratch Pad.



After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-16  
**Noise waveform (default settings)**

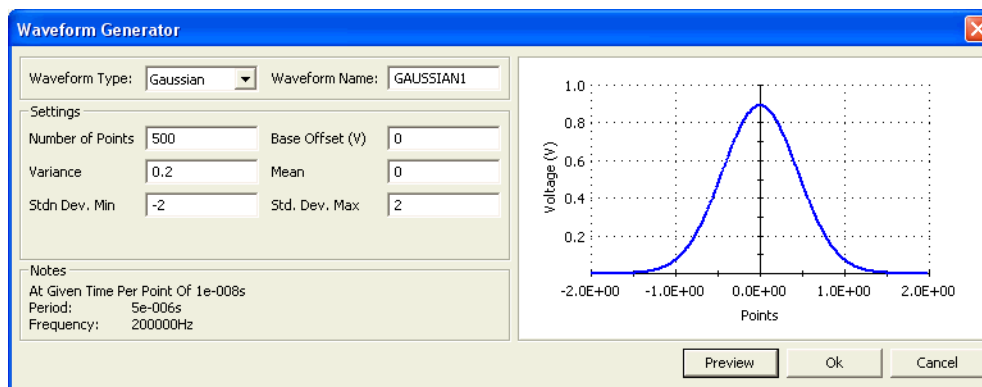


### Gaussian waveform

An example of a Gaussian waveform, using the default settings, is shown in [Figure 5-17](#). The waveform for this example is named GAUSSIAN1, but can be any name that is not already used in the Scratch Pad.

After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-17  
**Gaussian waveform (default settings)**



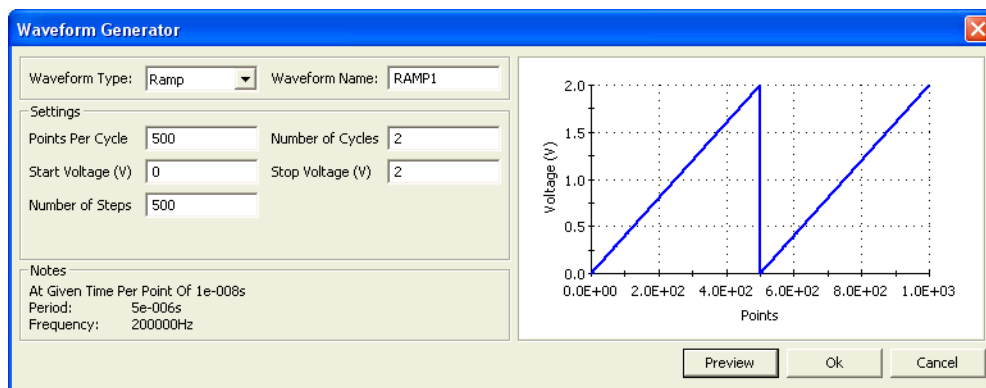
### Ramp waveform

An example of a ramp waveform, using the default settings, is shown in [Figure 5-18](#). The waveform for this example is named RAMP1, but can be any name that is not already used in the Scratch Pad.

After changing one or more settings, click **Preview** to display the waveform. Clicking **Ok** places the waveform in the Scratch Pad.



Figure 5-18  
**Ramp waveform (default settings)**



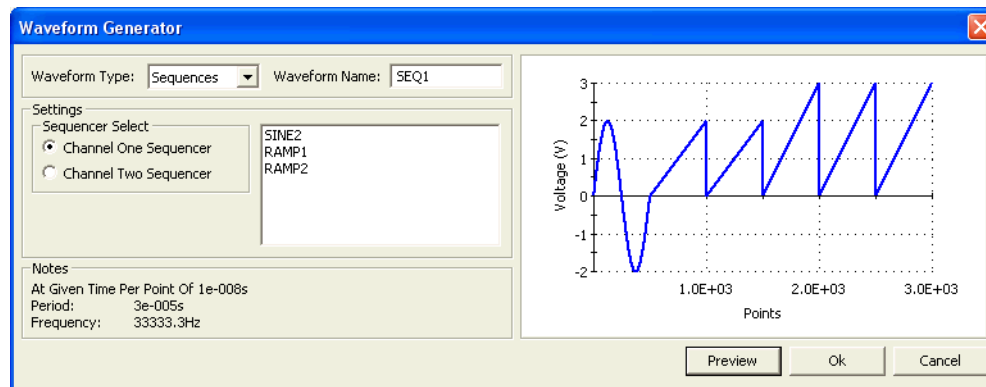
### Sequences waveform

An example of a sequences waveform is shown in [Figure 5-19](#). The waveform for this example is named SEQ1, but can be any name that is not already used in the Scratch Pad.

A sequence waveform consists of the waveforms that are present in the Channel 1 or Channel 2 Sequencer. [Figure 5-7](#) shows the Sequencer for the two channels.

After selecting either **Channel One Sequencer** or **Channel Two Sequencer**, click **Preview** to show the waveform. Clicking **Ok** places the waveform in the Scratch Pad.

Figure 5-19  
**Sequences waveform**









---

## How to Control the 4200-SCS Remotely from a PC

In this section:

Topic	Page
Remote control overview (GPIB, Ethernet) .....	6-2
GPIB standards .....	6-2
Communication connections .....	6-2
Setting remote control mode (GPIB versus Ethernet) .....	6-3
KXCI settings tab .....	6-4
KXCI control interface .....	6-4
Starting KXCI and the GPIB command interpreter .....	6-4



## Remote control overview (GPIB, Ethernet)

The Keithley External Control Interface (KXCI) allows you to use an external computer to remotely control the SMUs, pulse generator cards and scope card in the Model 4200-SCS over the general purpose instrument bus (GPIB) or Ethernet (as defined in KCON). When controlled by an external computer, the Model 4200-SCS functions like any other GPIB instrument.

The KXCI GPIB command sets for the SMUs and PG2s are similar to the command set used by the HP Model 4145B. This similarity allows many programs already developed for the HP Model 4145B run on the Model 4200-SCS.

### GPIB standards

The GPIB is the IEEE-488 instrumentation data bus with hardware and programming standards originally adopted by the Institute of Electrical and Electronic Engineers (IEEE) in 1975. The Model 4200 conforms to these standards:

- IEEE-488.1-1987
- IEEE-488.2-1992

### Communication connections

**GPIB connections:** To connect the Model 4200-SCS to the GPIB, use a cable equipped with standard IEEE-488 connectors as shown in [Figure 6-1](#). Either end of this cable mates to the IEEE-488 connector on the rear panel of the Model 4200-SCS ([Figure 6-2](#)). Connect the other end of the cable to the IEEE-488 connector on the computer.

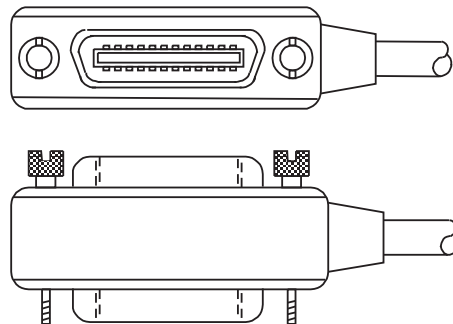
The connectors on the cable are stackable to allow GPIB connection to other instruments. However, to avoid damage, do not stack more than three connectors on any one unit.

---

**NOTE** To minimize interference caused by electromagnetic radiation, use shielded IEEE-488 cables such as the Keithley Instruments Models 7007-1 and 7007-2.

---

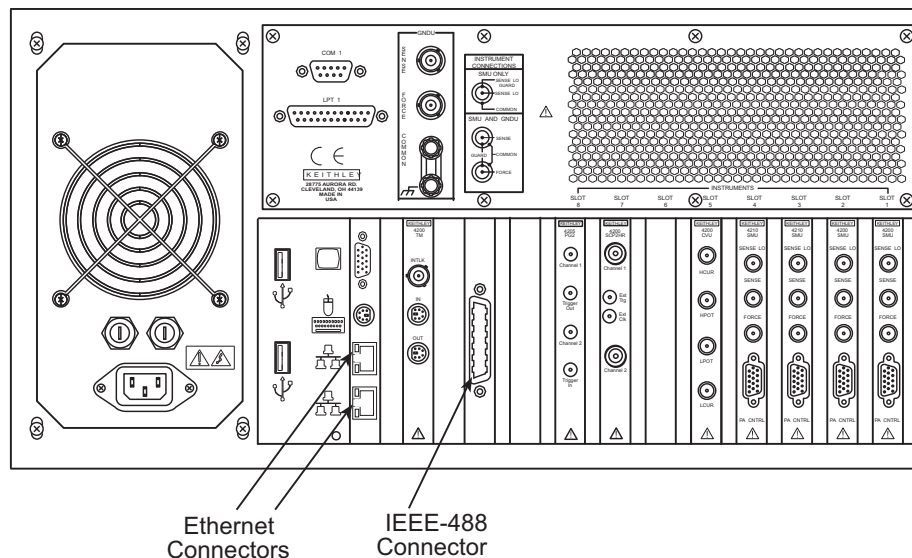
Figure 6-1  
**IEEE-488 Cable connectors**



**Ethernet connections:** Use a standard cable (CAT-5, RJ-45 terminated) to connect to the Model 4200-SCS (see [Figure 6-2](#)).



Figure 6-2  
**IEEE-488 and Ethernet connectors on Model 4200-SCS**  
**Model 4200-SCS**



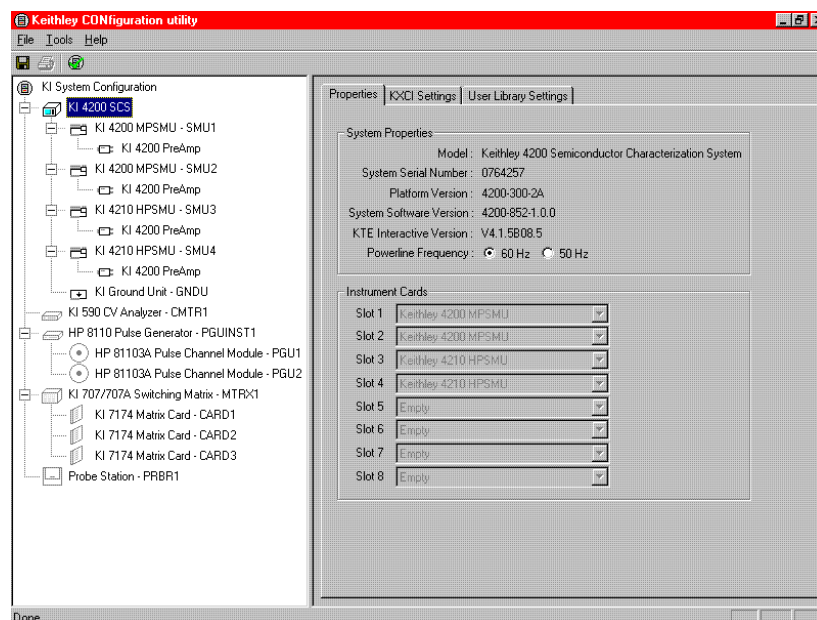
## Setting remote control mode (GPIB versus Ethernet)

Setting remote control mode on the 4200 is done through KCON on the KXCI settings tab. By default the 4200 is setup for GPIB remote control.

**Select the remote control mode on the Model 4200-SCS:**

1. Open KCON and click the **KI 4200-SCS** system node (see [Figure 6-3](#)).

Figure 6-3  
**KI 4200 SCS properties tab**





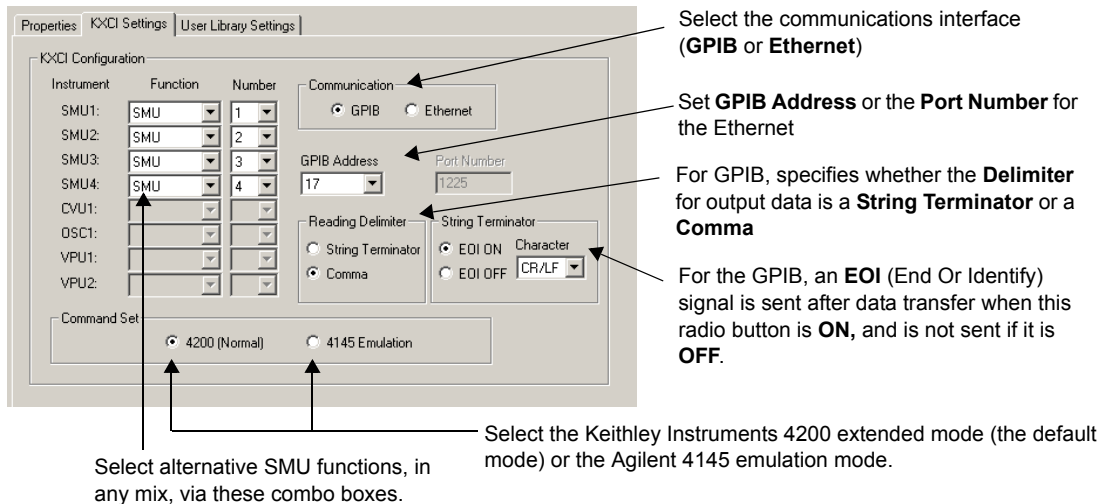
## KXCI settings tab

The **KXCI Settings** tab stores the KXCI settings. KXCI allows the Model 4200-SCS to behave as a GPIB slave, or Ethernet slave (see [Figure 6-4](#)).

- Click the KXCI settings tab.

Figure 6-4

### KI 4200 SCS KXCI settings tab



KXCI facilitates using an external computer to remotely control the Model 4200-SCS over the GPIB bus and Ethernet. In many cases, test programs developed for use with an Agilent 4145B run without modification when they are used with a Model 4200-SCS running KXCI. For detailed information regarding KXCI refer to the [Reference Manual, Keithley External Control Interface \(KXCI\), Section 9](#).

The **Function**, **Communication**, **GPIB Address/Port Number**, **Delimiter**, **EOI**, and **Command Set** settings on the **KXCI Settings** tab are each described in the [Reference Manual, KXCI Settings tab, page 7-15](#).

## KXCI control interface

### Starting KXCI and the GPIB command interpreter

Start the KXCI program by one of the following methods:

- Double-click the **KXCI** icon on the desktop.
- Click **Start > Programs > Keithley > KXCI**.

The KXCI user interface appears and the command interpreter starts (see [Figure 6-5](#)).

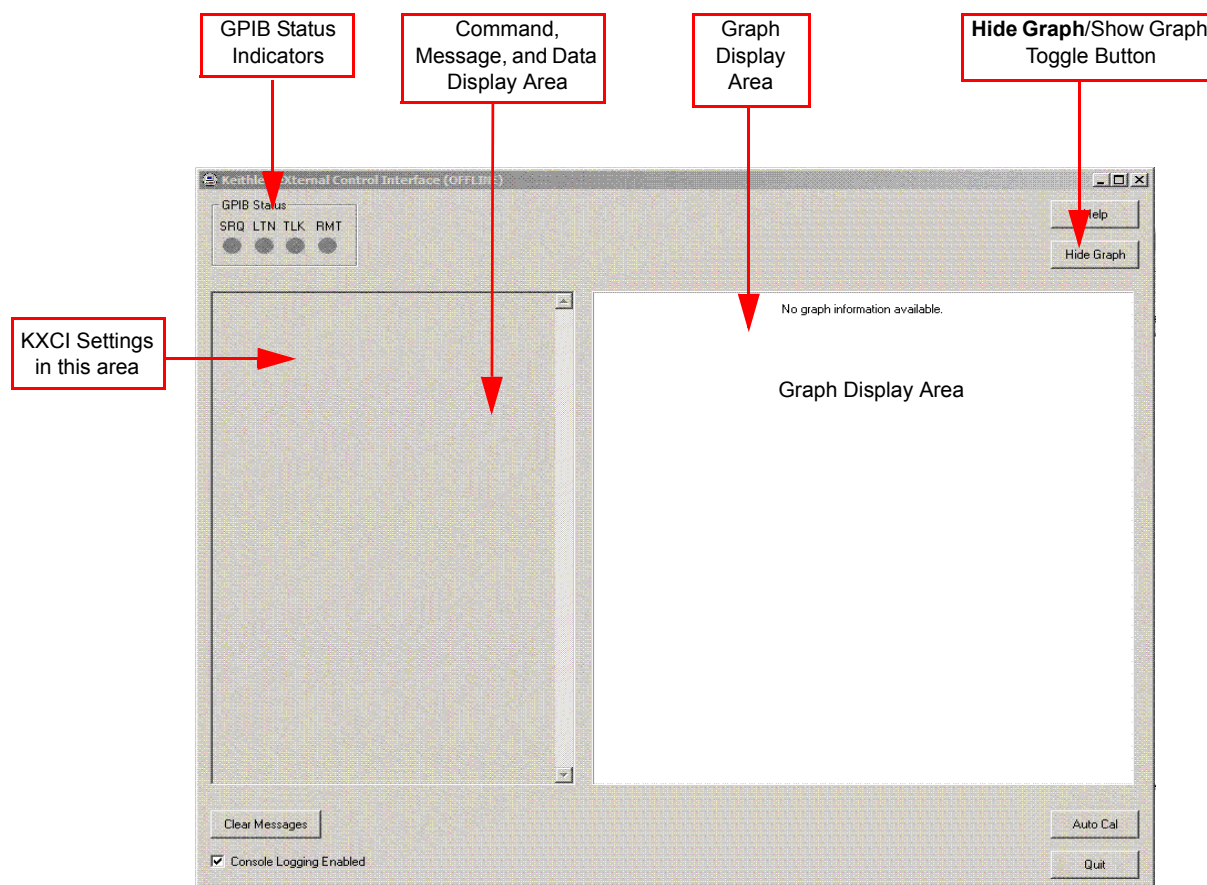
---

**NOTE** The **GPIB Status Indicators** only apply if GPIB communication is selected.

---



Figure 6-5  
KXCI user interface



The KXCI user interface and command interpreter controls the Model 4200-SCS over the GPIB or Ethernet, as defined in KCON. For detailed information regarding KXCI refer to the [Reference Manual, Keithley External Control Interface \(KXCI\), Section 9](#).

---

**NOTE** *Keithley Interactive Test Environment (KITE) and KXCI are mutually exclusive applications and cannot be run at the same time.*

**KITE:** *In this mode, the Model 4200-SCS is the controller and controls all internal and external instruments.*

**KXCI:** *In this mode, the Model 4200-SCS is a slave to a controlling PC over GPIB or Ethernet.*

---



---

**NOTE** *For further information about controlling the Model 4200-SCS through Ethernet, refer to [Reference Manual, Communication connections, page 9-4](#) and [Reference Manual, KXCI Ethernet client driver, page 9-102](#).*

---







# How to Create UTMs Using C Programming

In this section:

Topic	Page
<b>Keithley User Library Tool (KULT)</b> .....	<b>7-3</b>
Introduction .....	7-3
KULT window .....	7-3
Understanding the module identification area .....	7-4
Understanding the module parameter display area .....	7-5
Understanding the module code entry area .....	7-5
Understanding the terminating brace area .....	7-6
Understanding the tab areas .....	7-6
Parameters tab area .....	7-6
Parameter name field .....	7-7
Data type field .....	7-7
I/O field .....	7-7
Default, min, and max fields .....	7-8
Includes tab area .....	7-8
Description tab area .....	7-9
Build tab area .....	7-10
Understanding the status bar .....	7-10
Understanding the menus .....	7-10
File menu .....	7-11
Edit menu .....	7-12
Options menu .....	7-13
Help menu .....	7-14
<b>KULT Tutorials</b> .....	<b>7-14</b>
Tutorial 1: Creating a new user library and user module .....	7-15
Starting KULT .....	7-15
Naming a new user library .....	7-16
Naming a new user module .....	7-16
Entering the return type .....	7-17
Entering user module code .....	7-18
Entering parameters .....	7-18
Entering header files .....	7-20
Documenting the user module .....	7-20
Saving the user module .....	7-21
Compiling the user module .....	7-22
Finding code errors .....	7-23
Building the user library to include the new user module .....	7-24
Finding build errors .....	7-24
Checking the user module .....	7-24
Tutorial 2: Creating a user module that returns data arrays .....	7-28
Naming new user library and new VSweep user module .....	7-28
Entering the VSweep user-module return type .....	7-28
Entering the VSweep user-module code .....	7-28
Entering the VSweep user-module parameters .....	7-29
Entering the VSweep user-module header files .....	7-31
Documenting the VSweep user module .....	7-32
Saving the VSweep user module .....	7-32



<b>Compiling and building the VSweep user module .....</b>	<b>7-32</b>
<b>Checking the VSweep user module .....</b>	<b>7-32</b>
<b>Tutorial 3: Calling one user module from within another .....</b>	<b>7-34</b>
<b>Creating VSweepBeep user module copying existing user module ...</b>	<b>7-34</b>
<b>Calling independent user module from VSweepBeep user module ....</b>	<b>7-35</b>
<b>Specifying user library dependencies in VSweepBeep user module ..</b>	<b>7-37</b>
<b>Compiling and building the VSweepBeep user module .....</b>	<b>7-38</b>
<b>Checking the VSweepBeep user module .....</b>	<b>7-38</b>



# Keithley User Library Tool (KULT)

## Introduction

The Keithley User Library Tool (KULT) is a tool used to create and manage *user libraries*. A user library is a collection of one or more *user modules*. User modules are C programming language subroutines (or functions). User libraries are created to control instrumentation, analyze data, or perform any other system automation task programmatically. Once a user library has been successfully built using KULT, its user modules can be executed using the Keithley Interactive Test Environment (KITE) software tool.

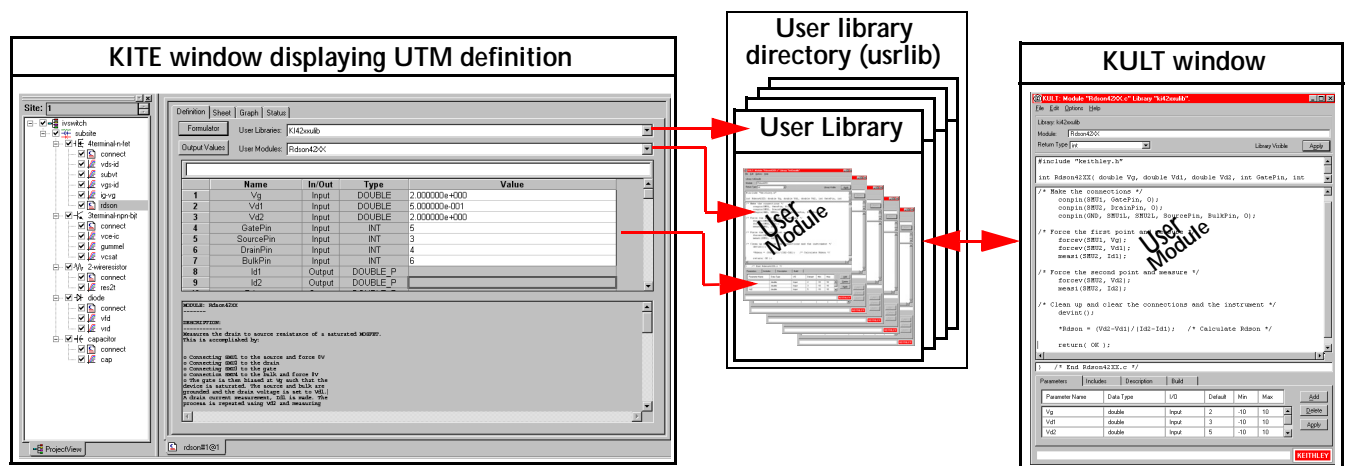
KULT provides a simple graphical user interface that helps even a novice programmer to effectively enter code, compile a user module, and link (build) a user library. KULT also provides user-library management features, including the copy module, copy library, delete module, and delete library menu commands. KULT manages user libraries in a structured manner. You can create your own user libraries to extend the capabilities of the Model 4200-SCS without requiring a software upgrade from Keithley.

To execute a KULT user module in KITE, you create a KITE user test module (UTM) and connect it to the user module. Once this user module is connected to the UTM, the following occurs each time KITE executes the UTM:

- KITE dynamically loads the user module and the appropriate user library directory (usrlib).
- KITE passes the user-module parameters (stored in the UTM) to the user module.
- Data generated by the user module is returned to the UTM for interactive analysis.

Figure 7-1 illustrates the relationships between user libraries, user modules, UTMs, KITE, and KULT.

Figure 7-1  
Relationships between KULT, KITE, user libraries, and UTMs

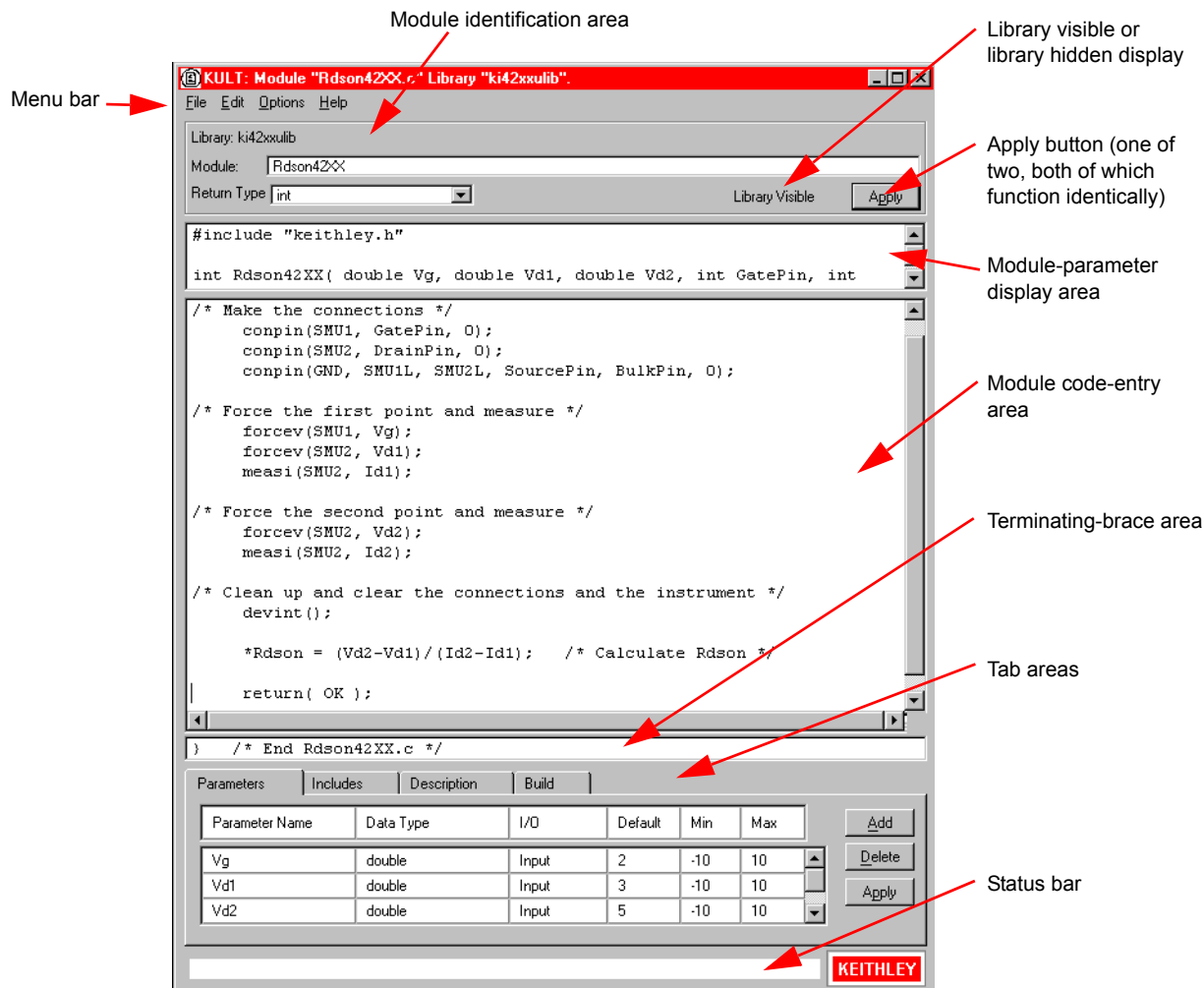


## KULT window

The KULT window is shown in Figure 7-2. It provides all the menus, controls, and user-entry areas needed to create/edit/view and build a user library, and to create/edit/view and compile a user module.



Figure 7-2  
KULT window



Each feature of the KULT window is discussed in the following subsections.

## Understanding the module identification area

The module identification area is located directly below the menu bar and defines the presently open user library and user module. The components of this area are used as follows:

- **Library Name:** Displays the name of the presently open (active) user library. To specify a user library, use the **File > New Library** or the **File > Open Library** command (see [File menu on page 7-11](#) for more information).
- **Module Name:** Displays the name of the presently open user module. To specify a user module, use the **File > New Module** or the **File > Open Module** command (see [File menu on page 7-11](#) for more information). Module Name also allows you to create a copy of the presently open user module in the same user library:
  1. Enter a new name in the **Module:** text box
  2. Click **Apply**.



---

**NOTE** *When naming a user module, remember to conform to case-sensitive C programming language naming conventions. Do not duplicate names of existing user modules or user libraries.*

---

- **Return type:** Defines the data type of all codes that are returned by `return(code)` statements in the user module. A scroll box, in which you select one of the following variable types:
  - **char:** Character data
  - **double:** Double precision data
  - **float:** Single precision, floating point data
  - **int:** Integer data
  - **long:** 32-bit integer data
  - **void:** No data returned

---

**NOTE** *When a UTM is executed by KITE, the value of the `return(code)` statement is displayed on the Data worksheet in the column labeled with the module name.*

---

**Library Visible / Library Hidden:** Displays whether or not the presently open user library is available to KITE. To change the hidden/visible status, check or uncheck the **Hide Library** option in the **Options** menu ([Figure 7-11](#)).

**Apply:** Used to update the presently open user module to reflect additions and changes. Also, if you change the **Module Name** name for the presently open user module, clicking the **Apply** button creates a copy of the module under the new name.

## Understanding the module parameter display area

The module parameter area is a display-only area that is located directly below the module identification area. In the module-parameter area, KULT displays the following:

- The C-language function prototype for the user module, reflecting the parameters that are specified in the Parameters tab area, and the `return(code)` data type.
- The `#include` and `#define` statements that are specified in the Includes tab areas.

## Understanding the module code entry area

The module code-entry area is located below the module-parameter area. The module code-entry area is the KULT window location where you enter, edit, or view the user-module C-code. Scroll bars located to the right and below the module-code entry area let you move through the code.

---

**NOTE** *Do Not enter the following C-code items in the module code-entry area (KULT enters these at special locations based on information in other places in the KULT window):*

- *#include and #define statements*
- *The function prototype*
- *The terminating brace*



*To control internal or external instrumentation, use functions from the Linear Parametric Test Library (LPTLib). For more information refer to the [Reference Manual, LPT Library Function Reference, page 8-22](#).*

## Understanding the terminating brace area

The terminating-brace area is a display-only area. KULT automatically enters and displays the terminating-brace for the user module code.

## Understanding the tab areas

Four tab areas are accessed by clicking one of the following tabs:

- Parameters
- Includes
- Description
- Build

Selections within the Parameters tab, Includes tab, and Description tab areas are facilitated using pop-up menus. The next four subsections describe the tab areas.

## Parameters tab area

The Parameters tab area is used to define and display the following for each parameter that is included in the user module call:

- Parameter name
- Parameter data type
- Input or output (I/O) data direction
- Default, min, and max values for the parameter

The Parameters tab area is located near the bottom of the KULT main screen. See the example in [Figure 7-3](#).

Figure 7-3

**Parameters tab area, example entries for the RDS on 42XX user module**

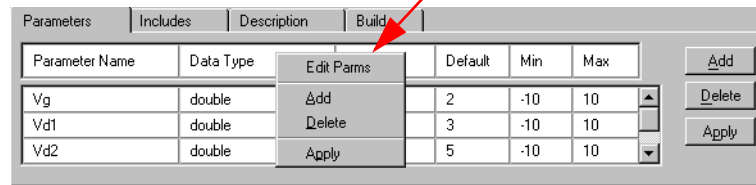
Parameter Name	Data Type	I/O	Default	Min	Max
Vg	double	Input	2	-10	10
Vd1	double	Input	3	-10	10
Vd2	double	Input	5	-10	10

Buttons: Add, Delete, Apply

A pop-up menu duplicates the Add, Delete, and Apply buttons at the right side of the Parameters tab area. Open the pop-up menu by right-clicking anywhere in the Parameters tab area (see [Figure 7-4](#)).



Figure 7-4  
Parameters tab area Add, Delete, Apply pop-up menu



#### To add, delete, or apply a parameter:

- To add a parameter, click **Add** and then enter the information as indicated in the field descriptions that follow.
- To delete a parameter, first click the parameter name or any of the adjacent fields; then click **Delete**.
- To apply changes made in the Parameters tab area, click **Apply**.

The next four subsections describe the six fields of the Parameters tab area.

#### Parameter name field

The parameter name field identifies the parameters that are passed to the user module. That is, these are the same parameters that would be specified in the user-module function prototype (which KULT constructs from the Parameters tab entries when you click **Apply**, and then displays in the module-parameter display area).

#### Data type field

The data type field specifies the parameter data type. Clicking on the arrow at the right of the data type field activates a pop-up menu, which lists the following data types:

- **char** Character data
- **char\*** Pointer to character data
- **float** Single precision, floating point data
- **float\*** Pointer to single precision, floating point data
- **double** Double precision data
- **double\*** Pointer to double precision point data
- **int** Integer data
- **int\*** Pointer to integer data
- **long** 32-bit integer data
- **long\*** Pointer to 32-bit integer data
- **F\_ARRAY\_T** Floating point array type
- **I\_ARRAY\_T** Integer array type
- **D\_ARRAY\_T** Double precision array type

#### I/O field

The I/O field defines whether the parameter is an input or output type. Clicking on the arrow to the right of the I/O field activates a pop-up menu that shows the Input and output selections.

---

**NOTE** The pop-up menu is displayed, and output can be selected, only for pointer data types (**char\***, **float\***, **double\***, and so on.) and array data types (**I\_ARRAY\_T**, **F\_ARRAY\_T**, and **D\_ARRAY\_T**).

---



## Default, min, and max fields

The **Default, Min, and Max** fields are used to specify the following:

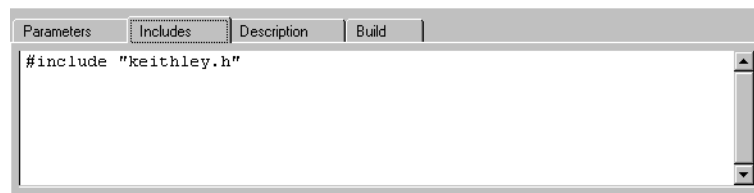
- **Defaultfield:** Specifies the default value for a non-array (only) input parameter.
- **Min field:** Specifies the minimum recommended value for a non-array (only) input parameter. When the user module is used in a KITE UTM, configuration of the UTM with a parameter value smaller than the min value causes KITE to display an out-of-range message (for a brief explanation of UTMs, refer to the [Reference Manual, User test module \(UTM\), page 6-45](#)).
- **Defaultfield:** Specifies the maximum recommended value for a non-array (only) input parameter. When the user module is used in a KITE UTM, configuration of the UTM with a parameter value larger than the max value causes KITE to display an out-of-range message.

## Includes tab area

The includes tab area lists the header files used within the user module. This area can be used to add `#include` and `#define` statements to the presently open user module. (see [Figure 7-5](#))

Figure 7-5

**Default Includes tab area**



By default, KULT automatically enters the `keithley.h` header file into the includes tab area. The `keithley.h` header file includes the following frequently used C-programming interfaces:

- `#include<stdio.h>`
- `#include<stdlib.h>`
- `#include<string.h>`
- `#include<math.h>`
- `#include"windows.h"`

In most cases, it is not necessary to add items to the includes tab area, because `keithley.h` provides access to the most common C functions. However, in some cases, both of the following may apply:

- You do not wish to include `keithley.h`
- You wish to include only the header files specifically needed by your user module, and all of the user modules on which it depends.

If so, you must minimally include the following header files and `#define` statements to properly compile and build user modules and user libraries:

```
#include "lptdef.h"
#include "lptdef_lowercase.h"
#include "kilogmsg_proto.h"
#include "ktemalloc.h"
#include "usrlib_proto.h"
#define PTextit _exit
```

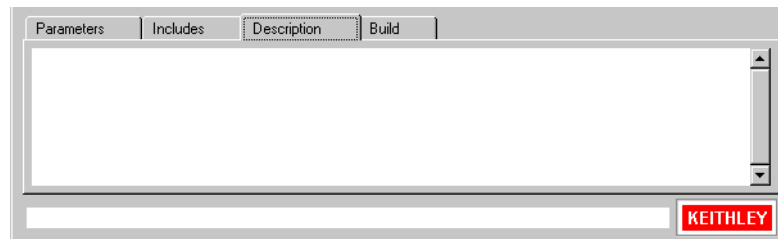


```
#define exit Unsupported Syntax
#define abort Unsupported Syntax
#define terminate Unsupported Syntax
```

## Description tab area

The description tab area, shown in [Figure 7-6](#), allows you to enter descriptive information for the presently open user module. Information entered in this area documents the module to the KITE user and is used to create KITE user library help.

Figure 7-6  
Description tab area




---

**CAUTION** Do not use C-code comment designators (*/\**, *\*/*, or *//*) in the Description tab area. When the user-module code is compiled, KULT also evaluates the text in this area. C-code comment designators in the Description tab area can be misinterpreted, causing errors.

---



---

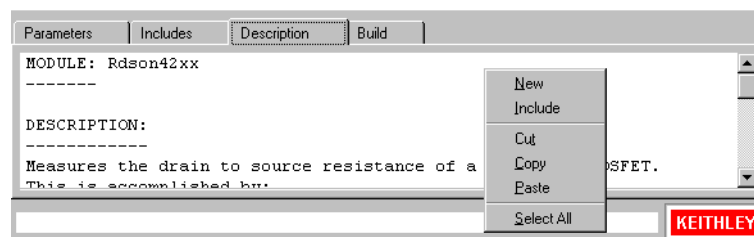
**NOTE** Do not place a period in the first column (the left-most position) of any line in the description tab area. Any text after a first-column period will not be displayed in the documentation area of a KITE UTM definition document.

---

### To enter a description:

1. Left-click in the **Description** tab area and enter the description.
2. Right-click in the **Description** tab area to open a pop-up edit menu. (see [Figure 7-7](#))

Figure 7-7  
Pop-up edit menu for the Description tab area



### The pop-up edit menu commands in the description tab area are used as follows:

- **New:** Deletes the present description from the description tab area, allowing you to enter a new description.
- **Include:** Imports any file that you specify, typically a text file, into the document tab area: only (to import a \*.c file into the module code-entry area. See [File](#)



[menu on page 7-11](#) information on the include command). Clicking **Include** displays the Include dialog box, in which you either browse and select a file or directly enter a file name and path. Clicking **Open** inserts the contents of the file at the cursor location.

- **Cut:** Removes highlighted text from the description tab area and copies it to the Windows® Clipboard®. The text on the clipboard can be restored to a new location(s), within or outside of KULT, using the paste function.
- **Copy:** Copies highlighted text from the description tab area to the Windows® Clipboard®. The text on the clipboard can be placed at a new location(s), within or outside of KULT, using the paste function.
- **Paste:** Places text from the Windows® Clipboard® at a selected location in the description tab area.
- **Select All:** Selects everything in the description tab area.

## Build tab area

The build tab area displays any error or warning messages that are generated during a code compilation or user-library build operation. When you click on a compilation-error message that is displayed in the build tab area, KULT highlights either the line of code where the error occurred or the next line, depending on how the compiler caught the error. KULT also highlights the error message. This facilitates error corrections.

If no errors are found, the build tab area displays one of the following:

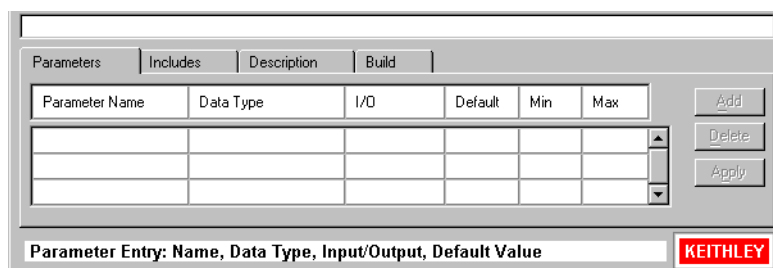
- After a compilation: **No Errors/Warnings Reported, Compilation was Successful.**
- After a build: **No Errors/Warnings Reported, Library Build was Successful.**

## Understanding the status bar

The status bar, located at the bottom of the KULT dialog box, displays a description of the KULT dialog box area at the cursor location. For example, if the cursor is in the parameter tab area, the status bar describes that area, as shown in [Figure 7-8](#).

Figure 7-8

**Example of description in status bar**



## Understanding the menus

This subsection describes the menus on the menu bar, which is located at the top of the KULT dialog box.

---

**NOTE** Most of the menu-bar menus apply to the part of the KULT dialog box that is above the tab areas. Refer to [Understanding the tab areas on page 7-6](#) earlier in this section for information about special pop-up menus in the tab areas.

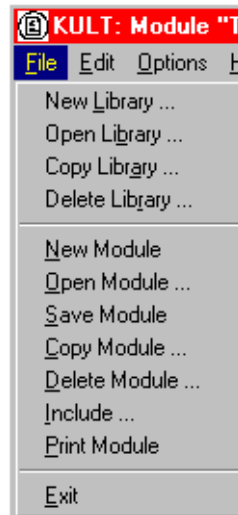
---



## File menu

The **File** menu is shown in [Figure 7-9](#).

Figure 7-9  
KULT File menu



### File menu commands:

- **New Library:** Creates a new user library. Clicking **New Library** displays the enter library dialog box, in which you can name a new user library. Clicking **Ok** initializes and opens the new user library in place of the presently open library.

---

**NOTE** By default, user libraries are stored in the `C:\S4200\kiuser\usrlib` directory. However, they can be stored on any accessible disk drive. For more information refer to the [Reference Manual, Changing the active user-library directory, page 8-7](#).

---

- **Open Library:** Opens an existing user library. Clicking **Open Library** displays the open library scroll list in which you can select an existing user library. Clicking **Ok** opens the selected library in place of the presently open library.
- **Copy Library:** Creates a copy of the currently open user library. Clicking **Copy Library** displays the enter library dialog box, in which you name the new user library into which to copy the presently open library. Thereafter, clicking **Ok** copies the presently open user library into the new library.
- **Delete Library:** Deletes an existing user library. Clicking **Delete Library** displays the delete library scroll list in which you can select the user library to be deleted. Thereafter, clicking **Ok** deletes the selected library and all of its contents.
- **New Module:** Creates a new user module. Clicking **New Module** clears module information in the KULT window and allows a new user-module name to be entered in the **Module** text box (the name must not duplicate the name of any existing user module or user library in the entire collection of user libraries). After entering the name, clicking **Apply** initializes the new user module.
- **Open Module:** Opens an existing user module. Clicking **Open Module** displays the open module scroll list in which you can select an existing user module. Clicking **Ok** opens the selected module in place of the currently open module.
- **Save Module:** Saves the presently open user module.



- **Copy Module:** Creates a copy of the currently open user module. Clicking **Copy Module** displays the select library scroll list; there you select the user library in which to copy the presently open user module. Then, clicking **Ok** displays the enter new module dialog box; there you must enter a unique user-module name that must not duplicate the name of any existing user module or user library (in the entire collection of user libraries). Clicking **Ok** in the dialog box copies the presently open module into the selected library, under the new name (the presently open module remains open).
- **Delete Module:** Deletes a user module from the presently open user library. Clicking **Delete Module** displays the KULT: Library [PresentlyOpenLibraryName] list box; there you select the module to be deleted. Clicking **Ok** deletes the selected module (the presently open module continues to be displayed, even if it is the module that you deleted). However, the executable user-library file, a dynamic link library (DLL), will still contain the deleted module until you rebuild the library.
- **Include:** Imports a \*.C file that you specify into the module code-entry area only (to insert a text or other file into the document tab area, refer to [Description tab area](#) and read about the include pop-up menu command). Clicking **Include** displays the include other file dialog box; there you either browse and select a file or directly enter a file name and path. Thereafter, clicking **Open** inserts the file at the cursor location.

---

**CAUTION** The **File > Include** command inserts *everything* from the specified file. If the specified file is the source file for a KULT user module <ModuleName.c>, everything that KULT saves into the user module (not only the C code) is imported. Hence, you must edit the entered text to remove all but the needed information. In particular, you must remove any comments of the form `/* USRLIB MODULE ____ */`, which KULT interprets in a special way when using the module.

Sometimes the following is more efficient than using the **File > Include** command: copying only the wanted code text from the source file, then pasting it into the module code-entry area.

---

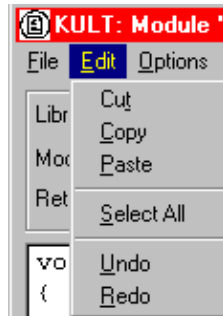
- **Print Module:** Prints a DOS text file containing all of the information for the presently open user module. The text file is arranged in the form that KULT uses internally.
- **Exit:** Exits KULT.

## Edit menu

The **Edit** menu contains typical Windows® editing commands. (see [Figure 7-10](#))



Figure 7-10  
KULT Edit menu



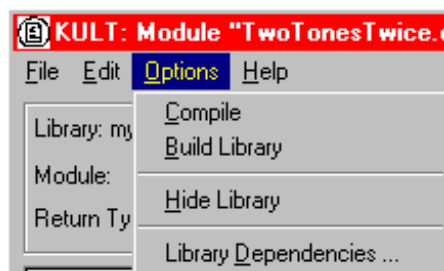
**Edit menu commands:**

- **Cut:** Removes highlighted text and copies it to the Windows® Clipboard®. The text on the clipboard can be restored to a new location(s), within or outside of KULT, using the paste function.
- **Copy:** Copies highlighted text to the Windows® Clipboard®. The text on the clipboard can be placed at a new location(s), within or outside of KULT, using the paste function.
- **Paste:** Places the text from the Windows® Clipboard® at a selected location.
- **Select All:** Selects everything in the module code-entry area.
- **Undo:** Allows you to reverse up to the last ten changes made in the module code-entry area.
- **Redo:** Allows you to reverse up to the last ten undo operations in the module code-entry area.

## Options menu

The KULT Options menu is shown in [Figure 7-11](#).

Figure 7-11  
KULT Options menu



**Options menu commands:**

- **Compile:** When clicked, compiles the source files for the presently open user module into object files and checks for errors in the module.
- **Build Library:** When clicked, adds the presently open user module (or updates changes) to the presently open user library. All of the modules in the presently open user library, and any libraries on which the presently open module depends, are linked together. A dynamic link library (DLL) is then created that is accessible using UTMs in KITE.



---

**NOTE** *Some Keithley Instruments-supplied user libraries contain dependencies. If for any reason you need to build or rebuild such libraries, be sure that you specify the dependencies in the window opened by **Options > Library Dependencies** (refer to descriptions below and to details in the [Reference Manual, Working with interdependent user modules and user libraries, page 8-13](#)). Otherwise, the Build Library function will fail. For example, `ki82ulib` depends on `ki590ulib` and `winulib`. You must specify these dependencies before rebuilding `ki82ulib`, (for example, after making changes).*

---

- **Hide Library:** When checked, causes the current user library to be unavailable to KITE. For example, use hide library if you want to designate that a user library is only to be called by another user library and is not to be connected to a UTM.
- **Library Dependencies:** When clicked, displays the Library Dependencies list box; there you must specify each user library that is called by, and must be linked to the presently open user library. All list-box selections toggle (**Do not** hold down the control or shift key to make multiple selections).

## Help menu

Help menu contains online help information about KULT:

- **Contents:** Allows access to the online KULT manual and other Model 4200-SCS reference information.
- **About KULT:** Displays the software version.

## KULT Tutorials

This section includes three tutorials. Each tutorial provides step-by-step instructions for accomplishing common tasks with KULT. The name of each tutorial is included below along with a summary of topics that are discussed:

- **Tutorial 1: Creating a new user library and new user module**
  - Naming a new user library
  - Naming a new user module
  - Entering a return type
  - Entering user module code
  - Entering parameters
  - Entering header files
  - Documenting the user module
  - Saving the user module
  - Compiling the user module
  - Finding code errors
  - Building the user library to include the new user module
  - Finding build errors
  - Checking the user module
- **Tutorial 2: Creating a user module that returns data arrays**
  - Naming a new user library and new `VSweep` user module



- Entering the VSweep user-module return type
  - Entering the VSweep user-module code
  - Entering the VSweep user-module parameters
  - Entering the VSweep user-module header files
  - Documenting the VSweep user module
  - Saving the VSweep user module
  - Compiling and building the VSweep user module
  - Checking the VSweep user module
- **Tutorial 3: Calling one user module from within another**
    - Creating the VSweepBeep user module by copying an existing user module
    - Calling an independent user module from the VSweepBeep user module
    - Specifying user library dependencies in the VSweepBeep user module
    - Compiling and building the VSweepBeep user module
    - Checking the VSweepBeep user module

## Tutorial 1: Creating a new user library and user module

KULT is a tool that facilitates the development of user libraries. Each user library is comprised of one or more user modules, and each user module is created using the C programming language.

This subsection contains a tutorial that is designed to show you how to create a new user library and new user module. A hands-on example is provided that illustrates how to create a user library containing a user module that simply activates the internal beeper of the Model 4200-SCS.

### Starting KULT

#### To start KULT:




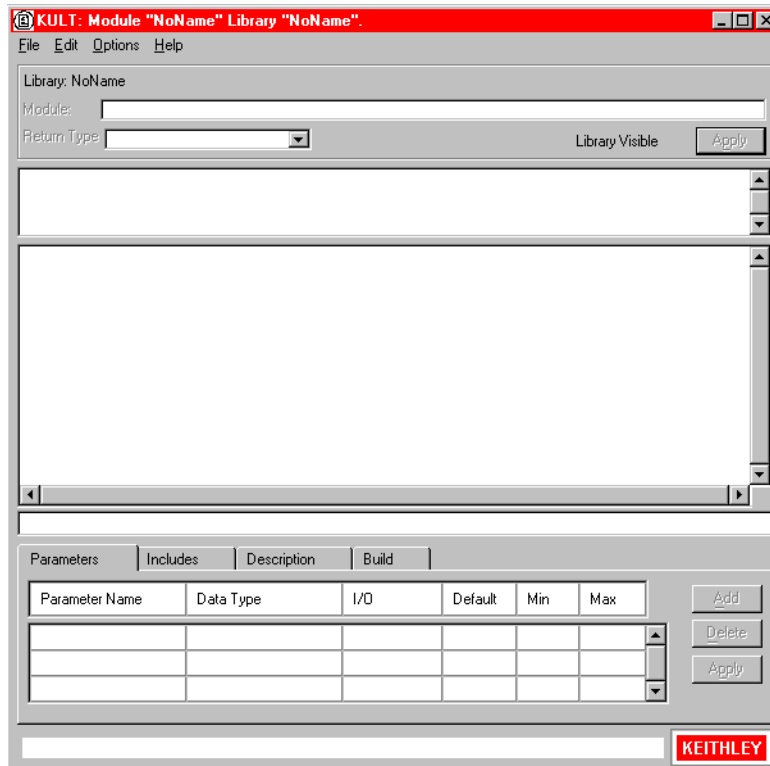
1. Start KULT by double-clicking the KULT icon  on the desktop or by clicking **KULT** in the Windows® **Start** menu (**Start > Programs > Keithley > KULT**).
2. A blank KULT window appears named **KULT: Module “NoName” Library “NoName.”** (see [Figure 7-12](#))
3. Continue with Naming a new user library.



Figure 7-12  
**Blank KULT window**



## Naming a new user library

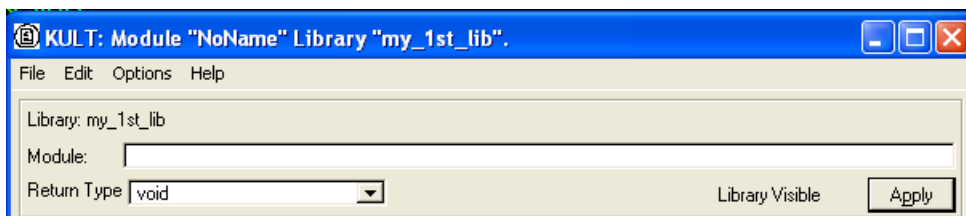
To name a new user library:

1. In the KULT file menu, click **New Library**.
2. In the Enter Library dialog box that appears, enter the new user library name. For this tutorial, enter `my_1st_lib` as the new user library name.

The dialog box name changes to KULT: Module **"NoName"** Library `my_1st_lib`, and the name next to library in the top left side of the dialog box is now `my_1st_lib`. (see [Figure 7-13](#))

3. Continue with Naming a new user module.

Figure 7-13  
**KULT window after naming user library**



## Naming a new user module

To name a new user module:

1. In the KULT file menu, click **New Module**.



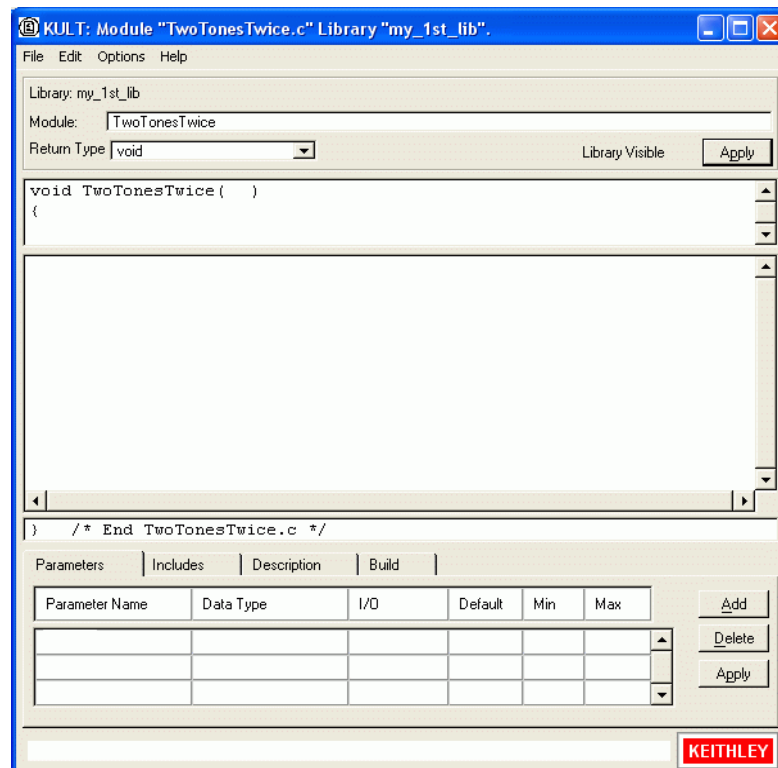
2. In the module text box at the top of the KULT window, enter the new user-module name. For this tutorial, enter `TwoTonesTwice` as the new user-module name.
3. Click **Apply**.

The KULT window changes as follows:

- The window's name changes to **KULT: Module "TwoTonesTwice" Library** `my_1st_lib`.
- You now see entries in the user-module parameters display area, in the terminating-brace display, and when you click the **Includes** tab, in the includes tab area. (see [Figure 7-14](#))

Figure 7-14

**KULT window after naming user module**



**NOTE** To view the entire module parameter display area, use the scroll arrows. For the `TwoTonesTwice` user module, the module parameter display area contains the following items:

- `#include "keithley.h"`
- `void TwoTonesTwice ( )`

4. Continue with Entering the return type.

### Entering the return type

If your user module will generate a return value, select the data type for the return value in the **Return Type** scroll box. However, the `TwoTonesTwice` user module will not produce a return value. Therefore, for the `TwoTonesTwice` module, retain the **void** default entry.

Continue with Entering user module code.



## Entering user module code

Enter the C code, referring to [Reference Manual, LPT Library Function Reference, page 8-22](#) for a complete list of supported I/O and SMU commands.

### To enter the C code:

1. Enter the new C code into the module-code entry area.  
For the `TwoTonesTwice` user module, enter the simple code listed below. The code deliberately contains a semicolon error to illustrate a KULT debug capability.

```
/* Beeps four times at two alternating user-settable frequencies. */
/* Makes use of Windows Beep (frequency, duration) function. */
/* Frequency of beep is long integer, in units of Hz. */
/* Duration of beep is long integer, in units of milliseconds. */
Beep(Freq1, 500); /* Beep at first frequency for 500ms */
Beep(Freq2, 500); /* Beep at second frequency */
Beep(Freq1, 500);
Beep(Freq2, 500) /* NOTE deliberately forget semicolon initially */
```

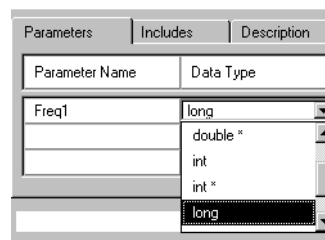
2. Continue with Entering parameters.

## Entering parameters

### To enter the required parameters for the code:

1. Click the **Parameters** tab (if the parameters tab area is not already displayed).
2. Click the **Add** button at the right side of the parameters tab area.
3. Under **Parameter Name**, enter the first parameter name (or accept the default). For the `TwoTonesTwice` user module, replace the default name with **Freq1**.
4. Under data type, enter the C data type for the first parameter. When you click the cell, a pop-up menu appears, displaying the allowed data types. (see [Figure 7-15](#))

Figure 7-15  
Data Type pop-up menu



For the `TwoTonesTwice` user module, select **long** under **Data Type**.

---

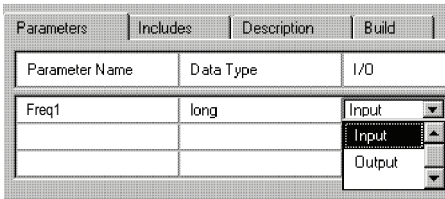
**NOTE** For an output parameter, only the following data types are acceptable: pointers (`char*`, `float*`, `double*`, and so on.) and arrays (`I_ARRAY_T`, `F_ARRAY_T`, or `D_ARRAY_T`).

---

5. Under I/O, specify whether the first parameter is an input or output parameter. If you specified a pointer or array data type under Data Type a scroll box appears when you click the **I/O** entry cell, displaying the Input and Output options. (see [Figure 7-16](#))



Figure 7-16  
I/O pop-up menu for pointers and arrays



If you do not specify a pointer or array data type under **Data Type**, you cannot change the default **Input** entry. For the `TwoTonesTwice` user module, the default **Input** entry is correct.

- 6. Under **Default**, **Minimum**, and **Maximum**, enter default, minimum, and maximum values for the parameter: to simplify and limit the choices to the user. For the `TwoTonesTwice` user module, enter **1000**, **800**, and **1200**, respectively.
- 7. Repeat steps 2 through 6 for all additional input and output parameters for the user module that you are creating. For the `TwoTonesTwice` module, add one more parameter, as shown in [Table 7-1](#).

Table 7-1  
**TwoTonesTwice entries for second line of Parameters tab area**

Parameter name	Data type	I/O	Default	Min	Max
Freq2	long	Input	400	300	500

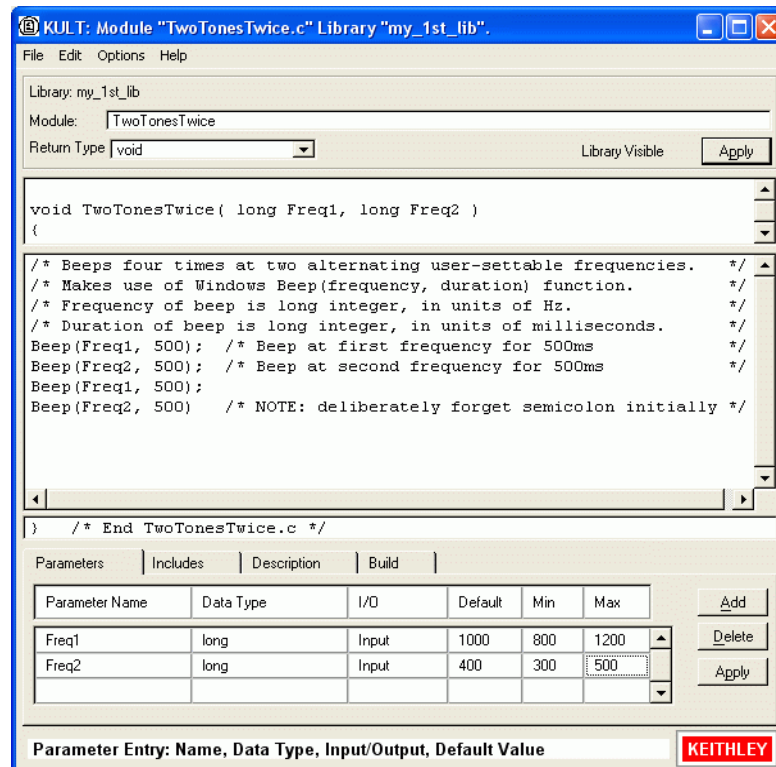
- 8. Click **Apply**.

**NOTE** The two **Apply** buttons, at the top and bottom of the dialog box, act identically.

In the module-parameter display area, the function prototype now includes the declared parameters. (see [Figure 7-17](#))



Figure 7-17  
**KULT window after entering and applying code and parameters**



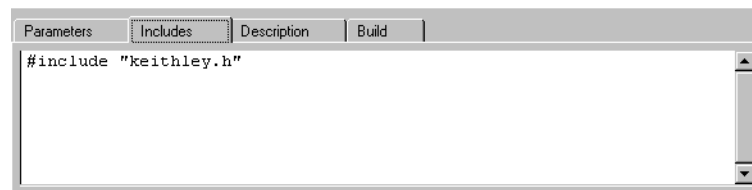
9. Continue with Entering header files.

## Entering header files

To enter the header files:

1. Click on the **Includes** tab at the bottom of the dialog box. The Includes tab area appears. (see [Figure 7-18](#))

Figure 7-18  
**Default Includes tab area**



2. Enter any additional header files that are needed by the user module. No additional header files are needed for the `TwoTonesTwice` user module or for any of the user libraries supplied by Keithley Instruments.
3. Click **Apply**.
4. Continue with Documenting the user module.

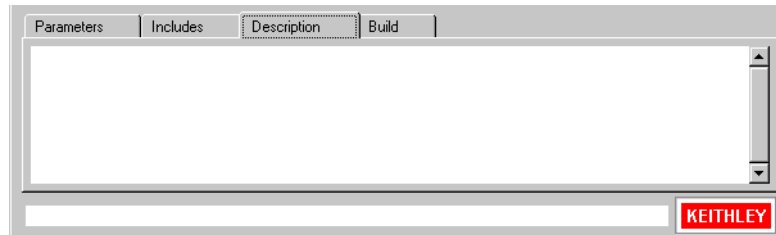
## Documenting the user module

To document the user module:



1. Click the **Description** tab at the bottom of the dialog box. The description tab area opens. (see [Figure 7-19](#))

Figure 7-19  
Description tab area



2. Enter any text needed to adequately document the user module to the KITE user, who does not see the comments that you include with the code.

---

**CAUTION** Do not use C-code comment designators (*/\*, \*/*, or *//*) in the Description tab area. When the user module is compiled, KULT also evaluates the Description text. C-code comment designators in the Description tab area can be misinterpreted, causing errors.

---



---

**NOTE** Do not place a period in the first column (the left-most position) of any line in the description tab area. Any text after a first-column period will not be displayed in the UTM description area.

---

For the `TwoTonesTwice` user module, enter the following information in the Description tab area:

**MODULE :**

`TwoTonesTwice`

**DESCRIPTION :**

Execution results in sounding of four beeps at two alternating user-settable frequencies. Each beeps sounds for 500ms.

**INPUTS :**

`Freq1` (double) is the frequency, in Hz, of the first and third beep.

`Freq2` (double) is the frequency, in Hz, of the second and fourth beep.

**OUTPUTS :**

None

**RETURN VALUES :**

None

3. Continue with Saving the user module.

### Saving the user module

Save the user module by clicking **Save Module** in the **File** menu.

Continue with Compiling the user module.



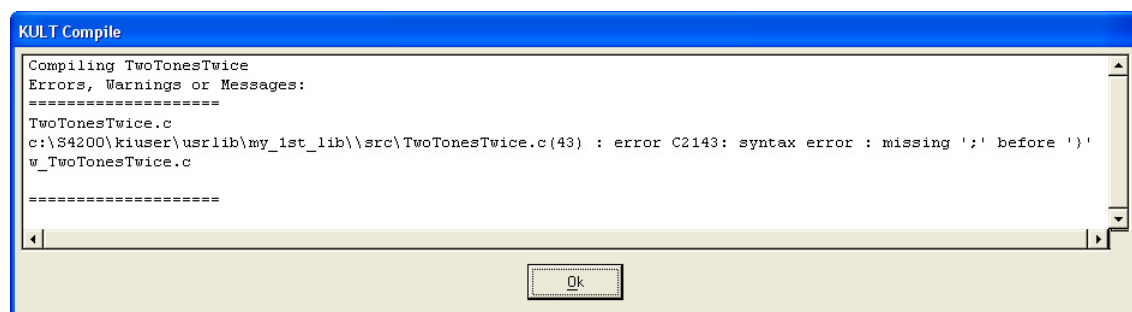
## Compiling the user module

### To compile the user module:

1. Click the **Build** tab at the bottom of the dialog box. The build tab area opens. After you compile a user module, the build tab area displays either a confirmation that the module compiled successfully or displays one or more compile-error messages.
2. In the **Options** menu, click **Compile**. The following occurs:
  - a. The user-module C source-code file is compiled.
  - b. The KULT Compile message box indicates the compilation progress and, if problems are encountered, displays error messages. For example, when you first compile the `TwoTonesTwice` user module (with a missing semicolon), you see the KULT Compile message box shown in [Figure 7-20](#).

Figure 7-20

### KULT Compile message box with error message



- c. When the KULT Compile message box closes (or, if there are error messages, when you click **Ok**), the build tab area displays either of the following:
  - If the compilation was successful, the following message appears: **No Errors/Warnings Reported, Compilation was Successful.**
  - If the compilation was unsuccessful, the error message(s), if any, that was displayed in the KULT Compile message box also displays in the **Build** tab area.

---

**NOTE** *True compilation errors (errors that prevent the user module from compiling) are displayed in red.*

*Warnings, which disclose suspect code that does not prevent compilation (such as an unused variable declaration) are displayed in blue.*

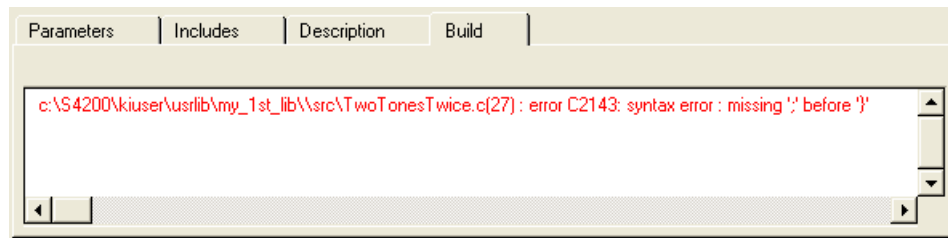
---

- For example, after you first compile the `TwoTonesTwice` user module (with the semicolon error) and click **Ok** in the KULT Compile message box, the build tab area appears as in [Figure 7-21](#).

Figure 7-21

### Compile error message in the Build tab area





3. Continue with Finding code errors.

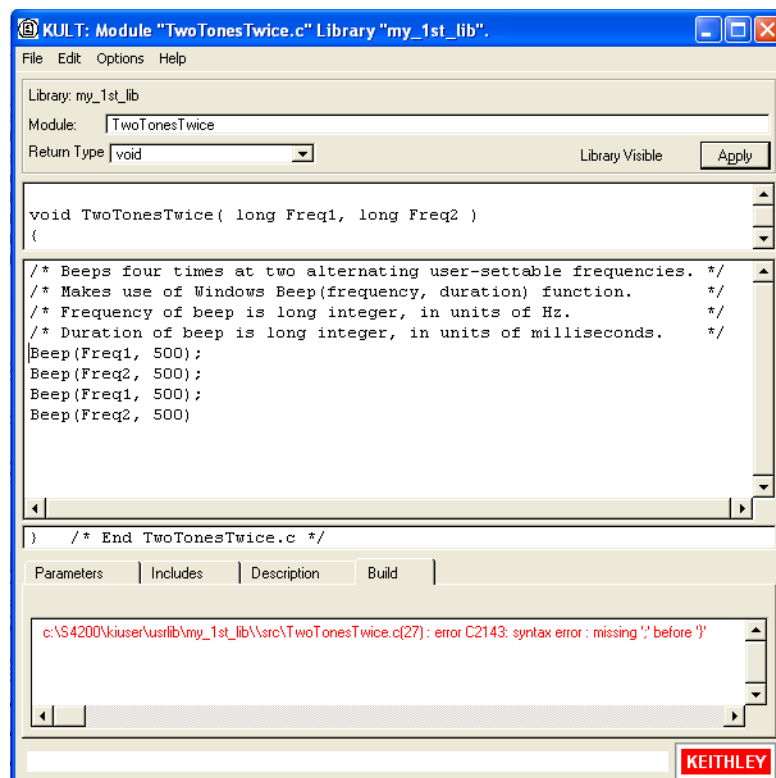
### Finding code errors

When you click on a compilation-error message that is displayed in the build tab area, KULT highlights either the line of code where the error occurred or the next line, depending on how the compiler caught the error. KULT also highlights the error message.

**To find code errors for the `TwoTonesTwice` user module:**

1. Click the error message. The last line of code (the line missing the semicolon) is highlighted, as shown in [Figure 7-22](#).

Figure 7-22  
Finding a code error

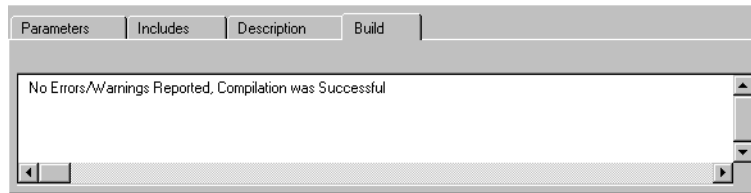


2. Add the missing semicolon at the end of the code [`Beep (Freq2, 500) ;`], and delete the comment about the missing semicolon.
3. Save the user module.
4. Compile the user module again.
  - The KULT Compile message box should now display no error messages and disappears automatically.



- The build tab area should display the successful-compilation message.  
(see [Figure 7-23](#))

Figure 7-23

**Successful-compilation message displayed in build tab area**

5. Continue with Building the user library to include the new user module.

**Building the user library to include the new user module**

After you have successfully compiled the user module, build the user library (or rebuild the user library) to include the module.

**To build the user library:**

1. Keep the Build tab area open.
2. In the **Options** menu, click **Build**. The following occurs:
  - a. The user library builds. All of the user modules in the presently open user library, and any libraries on which the presently open user module depends, are linked together. A DLL is then created that is accessible using UTMs in KITE.
  - b. The KULT Build Library message box (similar to the KULT Compile message box) indicates the build progress and, if linker problems are encountered, displays error messages (when you build the `TwoTonesTwice` user module, you should see no errors).
  - c. When the KULT Build Library message box closes (or, if there are error messages when you click **Ok**), the Build tab area displays either of the following:
    - If the compilation was successful, the following message appears: **No Errors/Warnings Reported, Library Build was Successful.**
    - If the compilation was unsuccessful, error messages, if any, that were displayed in the KULT Build Library message dialog box also display in the Build tab area (in red, only).

**Finding build errors**

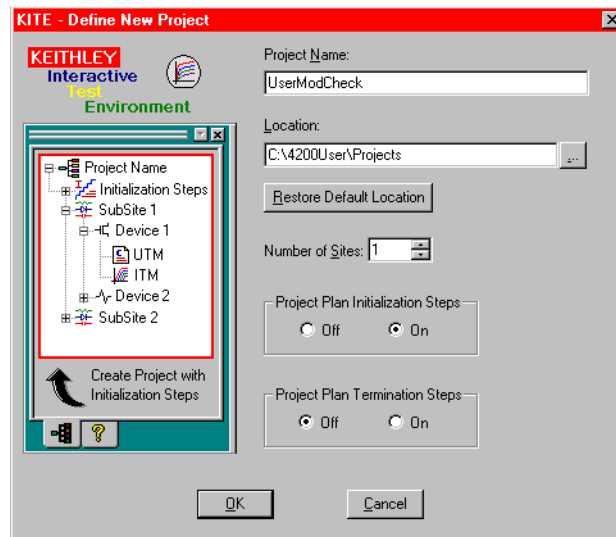
Find build errors using the information in the error message.

**Checking the user module****To check the user module by creating and executing a UTM in KITE:**

1. Create a simple KITE project to check user modules, as follows:
  - a. Start KITE by double-clicking the KITE icon on the desktop or by clicking **KITE** in the Windows® start menu (**Start > Programs > Keithley > KITE**).
  - b. In the KITE **File** menu, click **New Project**. The KITE - Define New Project dialog box appears.
  - c. In the KITE - Define New Project dialog box, do the following:
    - In the Project Name text box, enter a project name. A logical choice is **UserModCheck**. For this tutorial, enter **UserModCheck**.
    - Under Project Plan Initialization Steps, click **On**. (see [Figure 7-24](#))

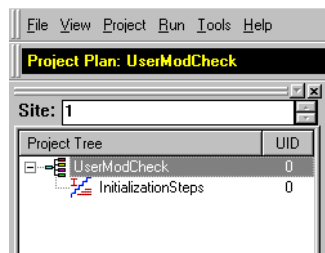


Figure 7-24  
Defining the UserModCheck project



- d. In the KITE - Define New Project dialog box, click **Ok**. The plan for the new project appears in the project navigator. (see [Figure 7-25](#))

Figure 7-25  
Initial UserModCheck project



2. Insert a new UTM in the project navigator. This will be used to execute the user module that you wish to check.

---

**NOTE** For simplicity, user modules that you create with KULT can be checked by creating Initialization Steps UTMs in the project navigator. UTMs can also be attached to specific devices on specific subsites. This tutorial uses the Initialization Steps approach.

---

Do the following in KITE:

- In the Check Project Navigator, single-click on **Initialization Steps**.
- Single-click on the **Add new UTM** icon, found at the top of the KITE dialog box, or select **Project > New UTM**.

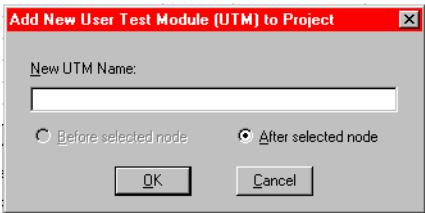


The Add New User Test Module (UTM) to Project dialog box appears. (see [Figure 7-26](#))



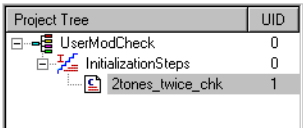
The Add New User Test Module (UTM) to Project dialog box appears. (see [Figure 7-26](#))

Figure 7-26  
Add New User Test Module (UTM) to Project dialog box



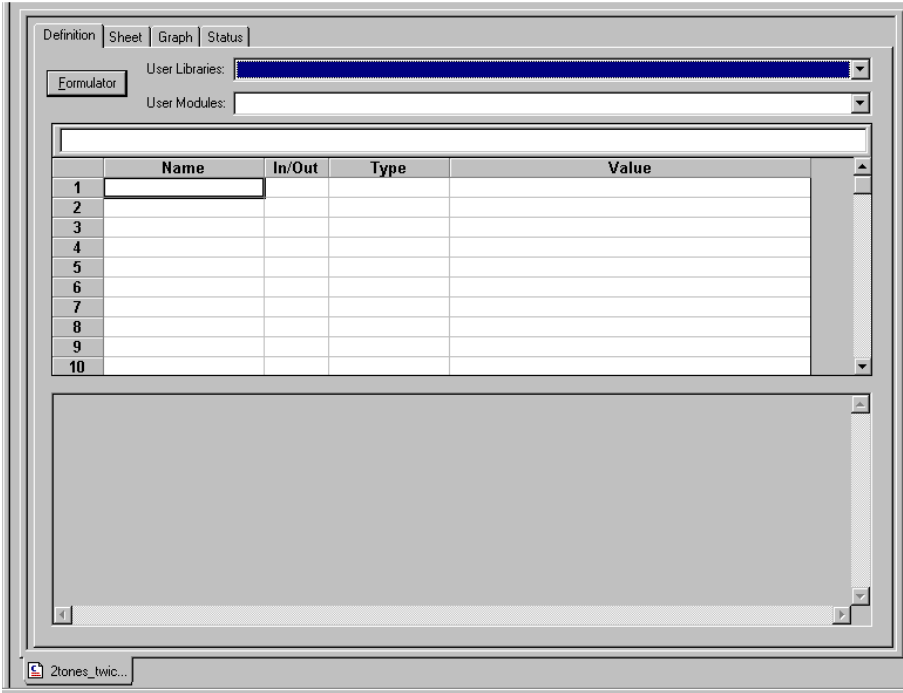
- c. In the Add New User Test Module (UTM) to Project dialog box, enter the name for the UTM. For the TwoTonesTwice user module, enter 2tones\_twice\_chk.
- d. Click **Ok**. The new UTM now appears under Initialization Steps in the project navigator. (see [Figure 7-27](#))

Figure 7-27  
New UTM inserted in the project plan



- 3. Configure the new UTM so that it executes the user module that you wish to check. Do the following:
  - a. In the project navigator, double-click the name of the new UTM (for the tutorial, double-click 2tones-twice-chk). A blank UTM Definition document appears. (see [Figure 7-28](#))

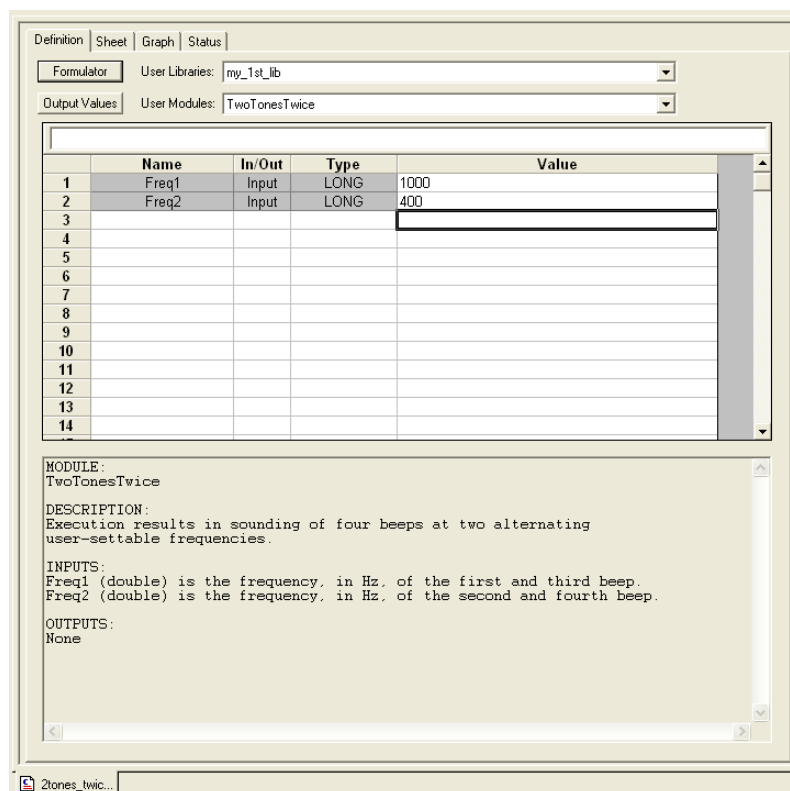
Figure 7-28  
Blank UTM Definition document








- b. In the User Libraries scroll box of the UTM Definition document, select the user library that contains the user module that you wish to test. For the tutorial, select `my_1st_lib`.
- c. In the User Modules scroll box of the UTM Definition document, select the user module that you wish to test (for the tutorial, select `TwoTonesTwice`). The UTM now displays the configuration parameters for the selected user module. (see [Figure 7-29](#))

Figure 7-29  
Configured UTM



- d. Accept the default parameters for now (you can experiment later after you establish that the user module executes correctly).
- e. Save the UTM and the project by clicking the **Save All** icon  at the top of the KITE screen, or by clicking **Save All** in the KITE **File** menu.
- f. Execute the UTM by clicking the green **Run test/plan** icon  at the top of the KITE screen, or by selecting **Run** in the KITE run menu.

The Run test/plan icon becomes gray, the test executes, and the square Abort test/plan icon illuminates red  for the duration of the test.

---

**NOTE** If you need to abort the test during execution, click the red **Abort Test/Plan** icon.

---

When you execute the `TwoTonesTwice` user module in the `2tones_twice_chk` UTM, you should hear a sequence of four tones, sounded at alternating frequencies.

4. If the user module that you created generates data, check the execution results in the UTM Data worksheet. To view the Data worksheet, click the **Sheet** tab at the top of the UTM Definition document.



---

**NOTE** For a UTM, the Data worksheet (and, if defined, the Graph document) always update at the conclusion of execution. Remember, you cannot view numerical and graphical results in real time.

---

The TwoTonesTwice user module, executing in the 2tones\_twice\_chk UTM, generates no data. For an example of numerical data, see the Tutorial 2 data under [Checking the VSweep user module](#) later in this section.

For more details on building a project, creating a UTM, and executing a UTM, refer to the following subsections in the [Reference Manual, Keithley Interactive Test Environment \(KITE\), page 6-1](#).

- [Building, modifying, and deleting a Project Plan](#)
- [Configuring the UTMs](#)
- [Executing Project Plans, Subsite Plans, Device Plans, and tests](#) in Section 6 of the Reference Manual

## Tutorial 2: Creating a user module that returns data arrays

This subsection provides a tutorial that is designed to help you to use array variables in KULT. It also illustrates the use of return types (or codes), and the use of two functions from the Keithley Linear Parametric Test Library (LPTLib).

Most of the basic steps that were detailed above are only abbreviated in this tutorial. Before doing the following tutorial, we suggest first completing [Tutorial 1: Creating a new user library and user module](#) explained earlier in this section.

### Naming new user library and new VSweep user module

To name new user library and new VSweep user module:

1. Start KULT by double-clicking the **KULT** icon on the desktop.
2. In the KULT file menu, click **New Library**.
3. In the Enter Library dialog box that appears, enter `my_2nd_lib` as the new user library name.
4. In the KULT file menu, click **New Module**.
5. In the **Module** text box at the top of the KULT dialog box, enter **VSweep** as the new module name.
6. Click **Apply**.
7. Continue with Entering the VSweep user-module return type.

### Entering the VSweep user-module return type

The VSweep user module generates an integer return value. Therefore, select **int** in the Return Type scroll box.

Continue with Entering the VSweep user-module code.

### Entering the VSweep user-module code

In the module code-entry area, enter the C code for the VSweep user module. The code is listed below (enter the code with KULT dialog box in full screen view).

```
/* VSweep module
-----
Sweeps through specified V range & measures I, using specified number of points.
Places forced voltage & measured current values (Vforce and Imeas) in output
arrays.
```



---

**NOTE** For  $n$  increments, specify  $n+1$  array size (for both NumIPoints and NumVPoints).

---

```

*/
double vstep, v; /* Declaration of module internal variables. */
int i;
if ( (Vstart == Vstop) ) /* Stops execution and returns -1 if */
    return( -1 ); /* sweep range is zero. */

if ( (NumIPoints != NumVPoints) ) /* Stops execution and returns -2 if */
    return( -2 ); /* V and I array sizes do not match. */

vstep = (Vstop-Vstart) / (NumVPoints -1); /* Calculates V-increment size. */

for(i=0, v = Vstart; i < NumIPoints; i++) /* Loops through specified number of */
    /* data points. */
{
    forcev(SMU1, v); /* LPTLib function forceX, which forces a V or I. */
    measi(SMU1, &Imeas[i]); /* LPTLib function measX, which measures a V or I. */
    /* Be sure to specify the *address* of the array. */

    Vforce[i] = v; /* Returns Vforce array for display in UTM Sheet. */

    v = v + vstep; /* Increments the forced voltage. */
}

return( 0 ); /* Returns zero if execution Ok.*/

```

Continue with Entering the VSweep user-module parameters.

## Entering the VSweep user-module parameters

To enter the required parameters for the code:

1. Click the **Parameters** tab (if the Parameters tab area is not already displayed).
2. Enter the information for the two voltage input parameters, as shown in [Table 7-2](#). Click the **Add** button before adding each new parameter.

Table 7-2

**VSweep entries for the two voltage input parameters**

Parameter name	Data type	I/O	Default	Min	Max
Vstart	double	Input	0	-200	200
Vstop	double	Input	5	-200	200

---

**NOTE** When executing the Vsweep user module in a UTM, the start and stop voltages (Vstart and Vstop) must differ. Otherwise, the first return statement in the code halts execution and returns an error number (-1). When a user module is executed using a KITE UTM, this return code is stored in the UTM Data worksheet. The return code is stored in a column that is labeled with the user-module name.

---



- Click **Add**, and on the third line, enter the measured-current parameter information shown in [Table 7-3](#).

Table 7-3

**Entries for the VSweep measured-current parameter**

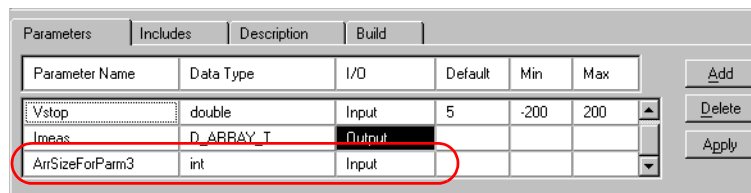
Parameter name	Data type	I/O	Default	Min	Max
Imeas	D_ARRAY_T	Output			

**NOTE** The following about the double-precision `D_ARRAY_T` array type, which applies also to the `I_ARRAY_T` and `F_ARRAY_T` array types:

- `D_ARRAY_T`, `I_ARRAY_T` and `F_ARRAY_T` are special array types that are unique to KULT.
- You cannot enter values in the Default, Min, and Max fields.
- On the scroll bar in the Parameters tab area, there is a space below the slider. This space indicates the existence of a hidden fourth line of incomplete parameter information: the array-size parameter specification (described in the next step).

- Scroll down to reveal line four of the Parameters tab area. Line four contains the KULT entered array-size parameter for the array that is specified on line three. (see [Figure 7-30](#))

Figure 7-30

**KULT-entered array-size parameters**

**NOTE** The following about the array size specification line:

- KULT enters initial information on this line automatically.
- The default Parameter Name entry is only a description of the required array-size parameter. You must replace it with an appropriate array-size parameter (for the VSweep user module, the correct entry is NumIPoints, as required per the user module code).
- The Data Type and I/O entries are correct as entered.
- You can enter a Default, Min, and Max array size.
- An array-size parameter line always appears after an array parameter line. You must always enter array-size parameters in the specified line.

- In the fourth line, under Parameter Name, in place of the designation ArrSizeForParm3, enter the parameter NumIPoints.
- On the fourth line, under Default, enter the number **11** for the default current-array size.



- Click **Add** and, on the 5th line, enter the forced-voltage parameter information shown in [Table 7-4](#).

Table 7-4

**Entries for the VSweep forced-voltage parameter**

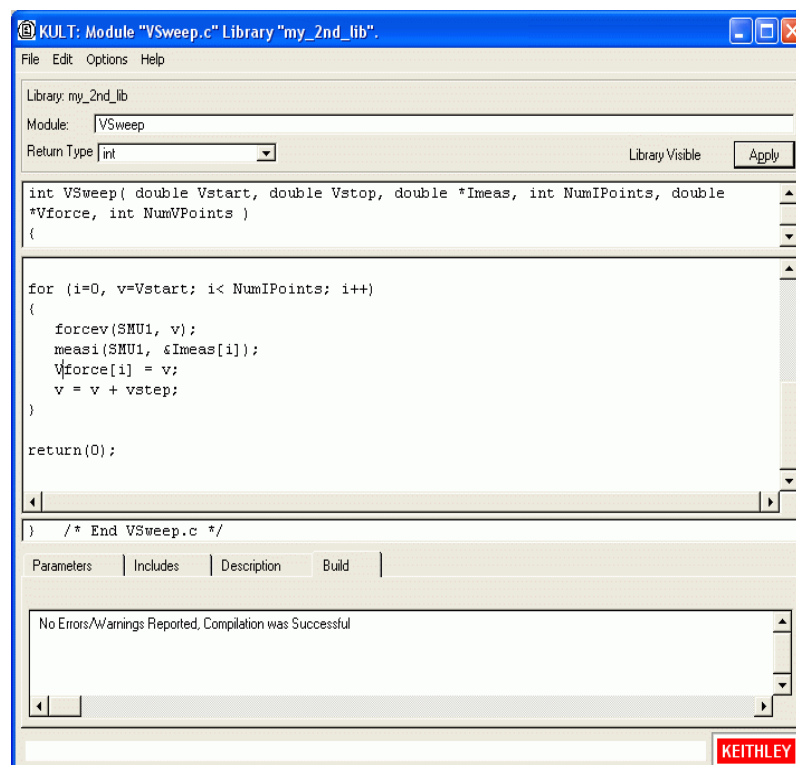
Parameter name	Data type	I/O	Default	Min	Max
Vforce	D_ARRAY_T	Output			

- In the sixth line, under **Parameter Name**, in place of the designation `ArrSizeForParm5`, enter `NumVPoints`.
- In the sixth line, under **Default**, enter the number **11** for the default voltage array size.

**NOTE** When executing the *VSweep* user module in a UTM, the current and voltage array sizes must match; *NumIPoints* must equal *NumVPoints*. Otherwise, the second return statement in the code halts execution and returns an error number (-2) in the *VSweep* column of the UTM Data worksheet.

- Click **Apply**. In the module-parameter display area, the function prototype now includes the declared parameters. (see [Figure 7-31](#))

Figure 7-31

**VSweep user-module dialog box after entering and applying code and parameters**

- Continue with Entering the *VSweep* user-module header files.

**Entering the VSweep user-module header files**

You do not need to enter any header files for the *VSweep* user module. The default `keithley.h` header file is sufficient. Continue with Documenting the *VSweep* user module.



## Documenting the VSweep user module

After clicking the Description tab, enter documentation for the user module, based on the comments provided in the code and other information about the module. Then continue with Saving the VSweep user module.

## Saving the VSweep user module

Save the user module by clicking **Save Module** in the **File** menu. Then continue with Compiling and building the VSweep user module.

## Compiling and building the VSweep user module

To compile the user module:

1. Click the **Build** tab at the bottom of the dialog box. The Build tab area opens.
2. In the Options menu, click **Compile**. The user module compiles. If the code and parameters were entered as specified, you should not see error messages.

---

**NOTE** If you do see error messages, check for typographical errors; then fix and recompile the user module. If necessary, review [Finding code errors on page 7-23](#) earlier in this section.

---

3. In the **Options** menu, click **Build**. The user library builds. You should not see error messages.
4. Continue with Checking the VSweep user module.

## Checking the VSweep user module

Check the user module by creating and executing a UTM in KITE, using the general procedure described in Tutorial 1 under [Checking the user module](#) earlier in this section. Before proceeding, observe the following guidelines:

1. Connect a 1 k  $\Omega$  resistor between the FORCE terminal of the ground unit (GNDU) and the FORCE terminal of SMU1.
2. Instead of creating a new project, reuse the UserModCheck project that you created in Tutorial 1. Add a new UTM called `v_sweep_chk`. You will subsequently use `v_sweep_chk` to execute the VSweep user module. (see [Figure 7-32](#))

Figure 7-32

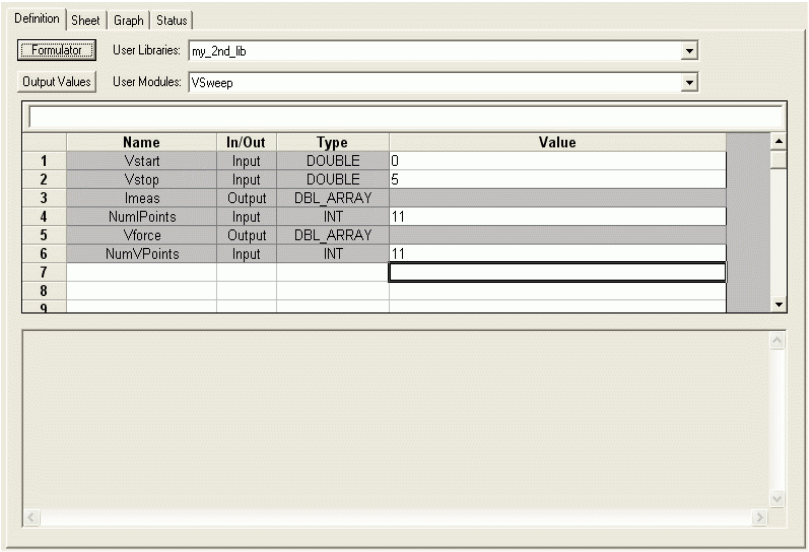
**New `v_sweep_chk` check UTM inserted in the project plan**

Project Tree	UID
UserModCheck	0
InitializationSteps	0
2tones_twice_chk	1
v_sweep_chk	1

3. Configure the `v_sweep_chk` UTM to execute the VSweep user module, which is found in the `my_2nd_lib` user library. [Figure 7-33](#) shows the UTM configured with the default parameter values.

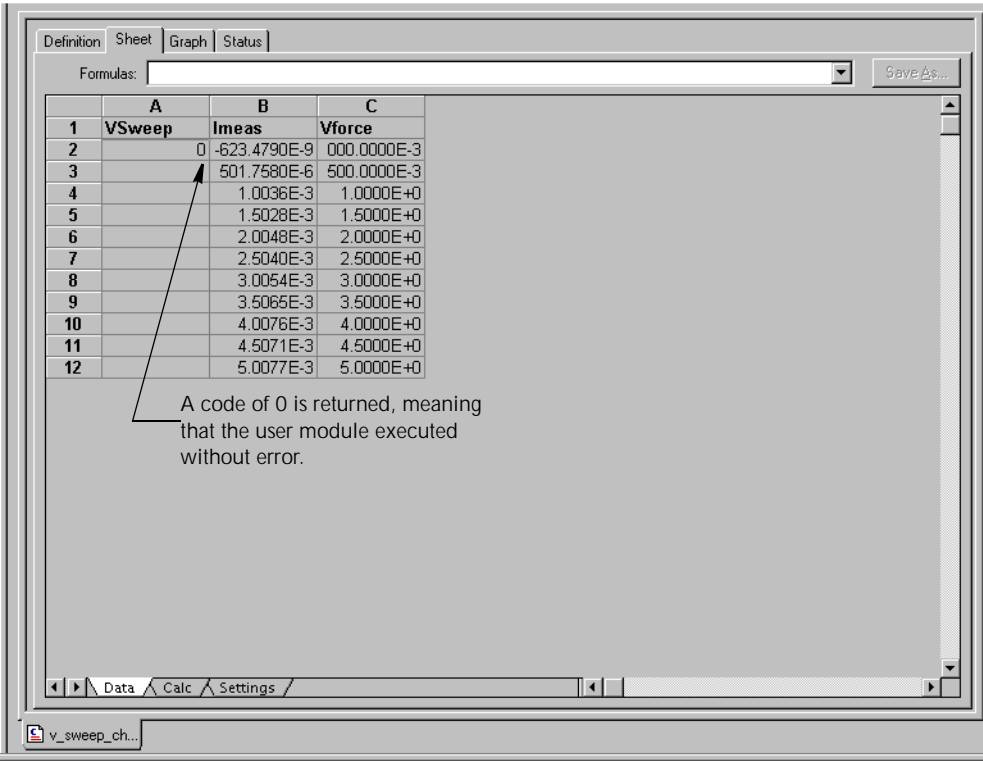


Figure 7-33  
Configure v\_sweep\_chk UTM



- 4. Execute the UTM as directed in Tutorial 1, using the default parameter values.
- 5. At the conclusion of execution, review the results in the Data worksheet. If you connected a 1 k  $\Omega$  resistor between SMU1 and GNDU, used the default UTM parameter values, and executed the UTM successfully, the results should appear similar to the results in [Figure 7-34](#). The current/voltage ratio for each row of results should be approximately 1 mA / V.

Figure 7-34  
Reviewing Data worksheet after executing a UTM





## Tutorial 3: Calling one user module from within another

KULT allows a user module to call other user modules. A called user module may be located within the same user library as the calling module or may be located in another user library. This subsection provides a brief tutorial that illustrates application of such dependencies. It also illustrates the **File > Copy Module** command.

In this tutorial, you create a new user module using two user modules that were created in the previous tutorials: [Tutorial 1: Creating a new user library and user module](#), and [Tutorial 2: Creating a user module that returns data arrays](#) earlier in this section:

- The VSweep user module, in the `my_2nd_lib` user library (a copy of which will be used as the dependent user library).
- The TwoTonesTwice user module, in the `my_1st_lib` user library, which is the independent user library that will be called by the VSweep user module.

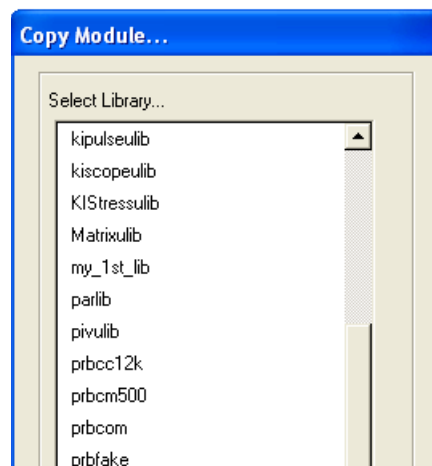
A copy of the VSweep user module, to be called VSweepBeep, calls the TwoTonesTwice user module to signal the end of execution.

### Creating VSweepBeep user module copying existing user module

To create the new user module:

1. Start KULT.
2. Open the VSweep user module as follows:
  - a. Open `my_2nd_lib` by:
    - i. Clicking **File > Open Library**
    - ii. Selecting `my_2nd_lib` from the list box that appears
    - iii. Clicking **Ok**.
  - b. Click **File > Open Module**, select `VSweep.c` from the list box that appears, and click **Ok**.
3. Copy `VSweep.c` to the new user module, VSweepBeep, as follows:
  - a. In the **File** menu, click **Copy Module**. The Copy Module list box appears. (see [Figure 7-35](#))

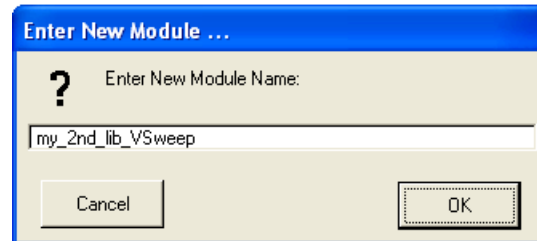
Figure 7-35  
Copy Module list box



- b. In the Copy Module list box, select `my_2nd_lib` (in this specific case, the user library for the copy is the same as the user library for the source) and click **Ok**. The Enter New Module dialog box appears. (see [Figure 7-36](#))



Figure 7-36  
Enter New Module dialog box



- c. In the Enter New Module dialog box, enter the name **VSweepBeep** instead of the default name, and click **Ok**.

---

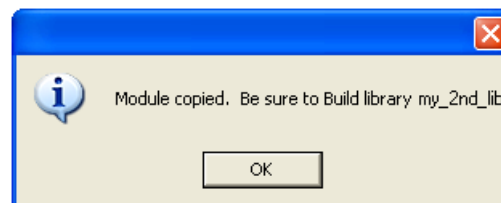
**NOTE** Each user module must have a unique name, regardless of the user library in which it resides. That is, in the presently active user library folder, there can only be one user module of a given name in its entire collection of libraries

More than one collection of user libraries can be maintained and accessed, each collection residing in a separate usrlib. However, only one usrlib can be active at a time. For more information, refer to the [Reference Manual, Managing user libraries, page 8-2](#).

---

KULT creates a copy of the user module under the new name and displays a message indicating the need to rebuild the user library. (see [Figure 7-37](#))

Figure 7-37  
Library build message box



You may skip the rebuild for now. Continue with the next step.

4. Open the new VSweepBeep user module.
  - a. Click **File > Open Module**.
  - b. Select **VSweepBeep.c** from the list box that appears. The KULT dialog box displays the VSweepBeep user module.
5. Continue with Calling independent user module from VSweepBeep user module.

---

**NOTE** You can also create a copy of the presently open user module in the same user library as follows:

1. Enter a new name in the User Module text box.
  2. Click *Apply*. Before using the user module, you must save and compile it, and then rebuild the user library.
- 

## Calling independent user module from VSweepBeep user module

To call the TwoTonesTwice user module at the end of the VSweepBeep user module:



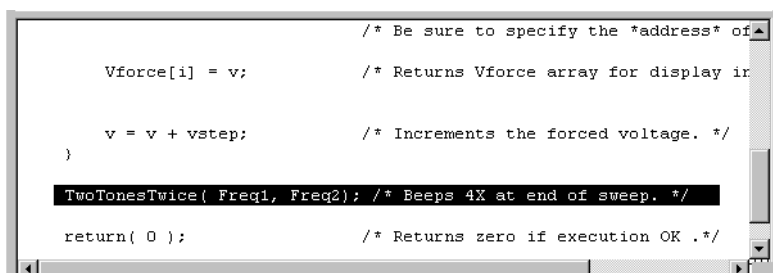
1. At the end of `VSweepBeep`, and just before the `return(0)` statement, add the following statement:

```
TwoTonesTwice(Freq1, Freq2); /* Beeps 4X at end of sweep. */
```

(see [Figure 7-38](#))

Figure 7-38

#### Calling `TwoTonesTwice` from `VSweepBeep`



2. In the Parameters tab area, add the **Freq1** and **Freq2** parameters, just as you did when you created the `TwoTonesTwice` user module, changing the Default, Min, and Max values if you prefer. (see [Table 7-5](#))

Table 7-5

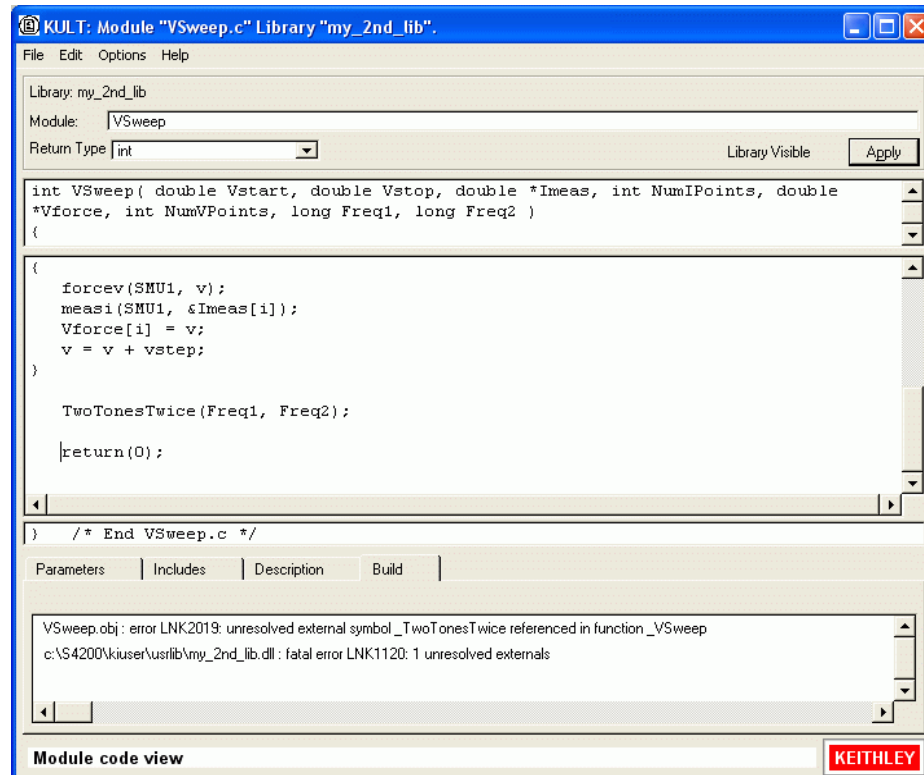
#### Parameter entries for the called user module, `TwoTonesTwice`

Parameter name	Data type	I/O	Default	Min	Max
Freq1	long	Input	1000	800	1200
Freq2	long	Input	400	300	500

3. Click **Apply**. The **Freq1** and **Freq2** parameters are added to the function prototype. (see [Figure 7-39](#))



Figure 7-39  
Completed VSweepBeep user module



4. Continue with Specifying user library dependencies in VSweepBeep user module.

### Specifying user library dependencies in VSweepBeep user module

Before compiling the presently open user module, you must specify all user libraries on which the user module depends (that is, the other user libraries containing user modules that are called).

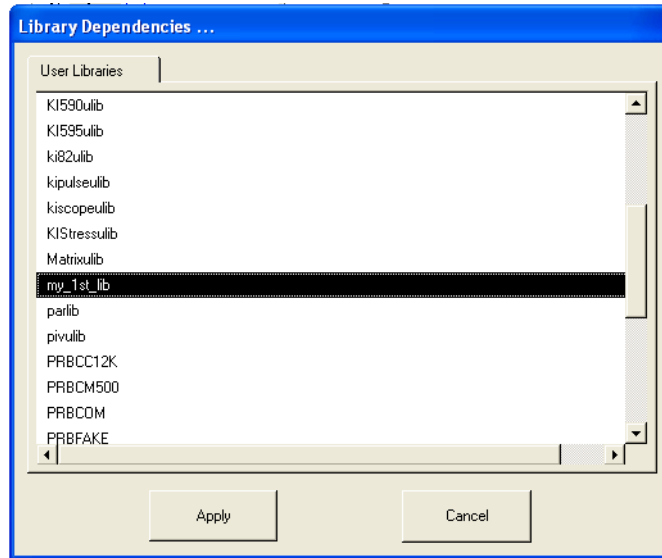
The VSweepBeep user module depends on the my\_1st\_lib user library.

#### To specify this dependency:

1. In the **Options** menu, click **Library Dependencies**. The Library Dependencies list box appears. (see [Figure 7-40](#))



Figure 7-40  
Library Dependencies list box



2. In general, in the Library Dependencies list box, select all user libraries on which the presently open user module depends (each selection toggles on and off). For the VSweepBeep module, only select `my_1st_lib`. (see [Figure 7-40](#))
3. Click **Apply**.
4. Continue with Compiling and building the VSweepBeep user module.

## Compiling and building the VSweepBeep user module

To compile and build the VSweepBeep user module:

1. Save the VSweepBeep user module.
2. Click the **Build** tab at the bottom of the dialog box. The Build tab area opens.
3. In the **Options** menu, click **Compile**. The user module compiles. If the code and parameters were entered as specified, you should not see error messages.

---

**NOTE** If you do see error messages, check for typographical errors; then fix and recompile the module. If necessary, review [Finding code errors on page 7-23](#) earlier in this section.

---

4. In the **Options** menu, click **Build**. The user library builds. You should not see error messages.
5. Continue with Checking the VSweepBeep user module.

## Checking the VSweepBeep user module

Check the user module just as you did in Tutorials 1 and 2, by creating and executing a UTM in KITE (refer to the general procedure described in Tutorial 1 under [Checking the user module on page 7-24](#)).

The text of the tutorial-specific guidelines below are almost identical to the text of the Tutorial 2 guidelines. Also, the data produced should be the same as the Tutorial 2 data. However, additionally, four beeps should sound at the end of execution, just as when you tested the `TwoTonesTwice` user module in Tutorial 1.

**Before proceeding, observe the following guidelines:**



1. Connect a 1 k  $\Omega$  resistor between the FORCE terminal of the GNDU and the FORCE terminal of SMU1.
2. Instead of creating a new project, reuse the UserModCheck project that you created in Tutorial 1. Add to this project a UTM called `v_sweep_bp_chk`.
3. Configure the `v_sweep_bp_chk` UTM to execute the `VSweepBeep` user module, which is found in the `my_2nd_lib` user library.
4. Execute the `v_sweep_bp_chk` UTM. Near the end of a successful execution, you should hear a sequence of four tones, sounded at alternating frequencies.
5. At the conclusion of execution, review the results in the Data worksheet (or the Graph document, if configured). If you connected a 1 k  $\Omega$  resistor between SMU1 and GNDU, used the default UTM parameter values, and executed the UTM successfully, your results should appear similar to the results in [Figure 7-34](#) at the end of Tutorial 2. The current/voltage ratio for each row of results should be approximately 1 mA / V.

To learn more about Advanced KULT features, refer to the [Reference Manual, Advanced KULT features, page 8-2](#).







## Symbols

"connect" parameters for "4terminal-n-fet" device .....	4-15
"cvsweep" UTM .....	4-43
"probesubsites" test descriptions .....	4-23

## Numerics

4- terminal n-MOSFET tests .....	3-5
4200-SCS Hardware Overview .....	1-11
59283	
Heading Level 3	
Average equivalent time points .....	1-38
72968	
Heading Level 4	
2.4.1.5 Project Plan .....	2-22

## A

Acquisition sample rate .....	1-37
Add	
New User Test Module (UTM) to Project dialog box .....	7-26
Add a Device Plan .....	4-42
Add a new Device Plan to a KITE project .....	4-41
Add a new Subsite Plan .....	4-41
Add a new Subsite Plan to a KITE project .....	4-40
Add a switch card to the system configuration .....	4-10
Add a switch matrix to the system configuration .....	4-9
Add a test fixture to the system configuration .....	4-9
Add the "cvsweep" UTM .....	4-43
Adding a Keithley 590 CV Analyzer to the system configuration .....	4-38
Adding a probe station .....	4-19, 4-30
Adding a pulse generator .....	4-29
Adding a switch matrix .....	4-19, 4-29
Advanced Connections .....	1-45

## B

Basic	
ground unit characteristics .....	1-40
SMU source-measure configuration .....	1-16
SMU/PreAmp source-measure configuration .....	1-20
Basic test execution .....	2-21
Blank	
UTM Definition document .....	7-26
Boot the system and log in .....	1-45
Build tab area .....	7-10
Building the user library to include the new user module .....	7-24
Buttons to close or reduce size of test documents .....	4-35

## C

Calculate functions .....	1-39
Calling TwoTonesTwice from VSweepBeep .....	7-36
Changing KITE startup behavior .....	3-11
Chassis ground .....	1-43
Checking the user module .....	7-24

Clear Append Data method for deleting Append worksheets .....	2-38
Communication connections .....	6-2
Compensation .....	3-19
Compiling the user module .....	7-22
Completed VSweepBeep user module .....	7-37
Configuration	
ITMs .....	2-17
Configuring the switch matrix .....	4-19, 4-30
Connect SMUs to N-channel MOSFET .....	4-24
Connect SMUs to NPN transistor .....	4-24
connect test .....	4-14
Connecting Devices Under Test (DUTs) .....	1-43
Connecting the switch matrix .....	4-20, 4-31
Connections .....	3-9
Contents of the Keithley Device file new-mosfet.kdv .....	2-28
Controlling a CV Analyzer .....	4-37
Add a Device Plan .....	4-41
Add a Subsite Plan .....	4-40
Add a UTM .....	4-42
Connections .....	4-37
Create a new project .....	4-39
KCON setup .....	4-38
Modifying the "cvsweep" UTM .....	4-43
Controlling a probe station	
KCON setup .....	4-18
Open the "probesubsites" project .....	4-21
Open the project plan window .....	4-21
Probe station configuration .....	4-20
Prober control overview .....	4-16
Running the test sequence .....	4-25
Test data .....	4-26
Test descriptions .....	4-22
Test system connections .....	4-17
Controlling a pulse generator	
Compare the test results .....	4-35
Overlaying graphs .....	4-36
Description of tests .....	4-32
"pgu1-init" test .....	4-33
"pgu1-setup" test .....	4-34
"pgu-trigger" test .....	4-34
First "connect" test .....	4-32
First "id-vg" test .....	4-32
Second "connect" test .....	4-33
Second "id-vg" test .....	4-34
Third "connect" test .....	4-34
KCON setup .....	4-28
Open the "ivpgswitch" project .....	4-31
Running the test sequence .....	4-35
Test system connections .....	4-28
Controlling a switch matrix	
"connect" test description .....	4-13
Running test sequences .....	4-11
Controlling external equipment overview .....	4-2
Copy Module list box .....	7-34
Copying entire KITE project .....	3-10
Copying individual tests using the test library manager .....	3-10



- CVU Frequency Sweep (bias) ..... 3-22
  - CVU Frequency Sweep (step) ..... 3-24
  - CVU Voltage Bias ..... 3-19
  - CVU Voltage List Sweep ..... 3-21
  - CVU Voltage Sweep ..... 3-20
- D**
- Data Save As window ..... 2-36
  - Data Type field ..... 7-7
  - Data Type pop-up menu ..... 7-18
  - Data worksheet for the VSweep user module, when  
used to evaluate a 1kW resistor ..... 7-33
  - Data worksheet of a Sheet tab containing both data  
and Formulator results ..... 2-33
  - Data worksheet of a Sheet tab containing data for  
multiple sweeps ..... 2-33
  - Data-source identifier ..... 2-35
  - DC Preamp ..... 1-20
  - Default project overview ..... 3-4
  - Default test library folders ..... 4-42
  - Default user directory ..... 2-25
  - Default, Min, and Max fields ..... 7-8
  - Define new project ..... 4-40
  - Define the system connections ..... 4-10
  - Defining the UserModCheck project ..... 7-25
  - Device Plan ..... 2-22
    - containing an ITM to be submitted ..... 2-48
    - window containing an ITM to be submitted ..... 2-49
  - Devices connected to 707A switching matrix ..... 4-8
  - Digital storage oscilloscope card ..... 1-35
  - Diode tests ..... 3-7
  - Displaying a Formulator equation using the Formula  
combo box ..... 2-35
  - Documenting the user module ..... 7-20
  - Dual Sweep ..... 2-15
- E**
- Edit menu ..... 7-12
  - EM process flow ..... 3-93
  - Enter New Module dialog box ..... 7-35
  - Entering header files ..... 7-20
  - Entering parameters ..... 7-18
  - Entering the return type ..... 7-17
  - Entering user module code ..... 7-18
  - Entries for the VSweep forced-voltage parameter .....  
7-31
  - Environmental Considerations ..... 1-4
  - Example
    - project ..... 2-23
  - Example of
    - description in status bar ..... 7-10
  - Executing an individual test ..... 2-22
- F**
- File menu ..... 7-11
  - Finding
    - build errors ..... 7-24
    - code errors ..... 7-23
  - Finding code errors ..... 7-23
  - First "connect" test - connects the device to the SMUs  
4-32
  - Forcing function
    - Configuration of ..... 2-17
    - Types summary ..... 2-13
- G**
- Front Panel ..... 1-11
  - Front panel ..... 1-11, 1-12, 2-4
  - Full Kelvin PreAmp/ground unit connections ..... 1-41
  - Full Kelvin SMU/ground unit connections ..... 1-41
- G**
- GPIB instrument connections ..... 1-9
  - GPIB standards ..... 6-2
  - Graph scale settings ..... 4-36
  - Graph Settings menu ..... 2-40
  - Graphing the Append worksheet data ..... 2-38
  - Ground unit ..... 1-39, 1-40
- H**
- HCI process flow ..... 3-93
  - Help menu ..... 7-14
  - How do I perform a charge pumping? ..... 3-146
  - How do I... Use the 4200-SCS to perform a Pulsed I-V  
test on my device? ..... 3-26
  - How to control a probe station ..... 4-15
  - How to control a switch matrix ..... 4-7
  - How to Control an external pulse generator ..... 4-27
  - How to Control external equipment ..... 4-2
  - How to Create your own ITMs ..... 2-13
  - How to Manage graphical test results in Graph Tab .....  
2-38
  - How to Run a Basic Test ..... 1-45
- I**
- I/O field ..... 7-7
  - I/O pop-up menu for pointers and arrays ..... 7-19
  - id-vg graphs ..... 4-36
  - Includes tab area ..... 7-8
  - Initial UserModCheck project ..... 7-25
  - Initialization and Termination Steps ..... 2-22
  - Input attenuation ..... 1-37
  - Input coupling ..... 1-37
  - Input filter ..... 1-37
  - Installation
    - Environmental requirements ..... 1-4
      - Operating environment ..... 1-4
      - Cleanliness ..... 1-5
      - Proper ventilation ..... 1-4
      - Temperature and humidity ..... 1-4
    - Shipping and storage environment ..... 1-4
  - Powering the 4200-SCS
    - Line power ..... 1-5
    - Line frequency setting ..... 1-6
    - Line fuses ..... 1-7
    - Line power connection ..... 1-6
    - Warm-up period ..... 1-7
  - SMU connections
    - Test fixtures ..... 1-43
  - Installation and System Connections ..... 1-3
  - Instrument panels ..... 1-11
  - Introduction ..... 1-14, 7-3
  - ITM (Interactive Test Module)
    - Definition tab ..... 2-11
  - ITMs and UTMs in the Project Navigator ..... 2-9
- K**
- KCON (Keithley Configuration Utility) ..... 4-5
  - KCON setup ..... 4-8



Keithley eXternal Control Interface (KXCI) .....	6-4
Keithley Interactive Test Environment (KITE)	
Displaying and analyzing project results	
Displaying and analyzing data using the	
Sheet tab .....	2-31
Saving the worksheet .....	2-35
Saving a Sheet tab to the project .....	2-35
Saving the Sheet tab to an external	
spreadsheet file using the Save As	
button .....	2-36
Understanding and using the Data	
worksheet of a Sheet tab .....	2-34
Understanding the data-source identifier	
2-35	
Managing KITE application files and test results	
Devices	
Creating and adding .....	2-27
files .....	2-27
libraries .....	2-25
library access selection .....	2-26
subdirectory .....	2-25
Projects subdirectory .....	2-28
Tests subdirectory .....	2-29
Overviewing KITE	
Defining a UTM .....	2-11
Defining an ITM .....	2-10
KITE interface .....	2-3
Project Navigator .....	2-5
Submitting devices, ITMs, and UTMs to libraries	
Submitting devices to a library .....	2-45
Submitting tests to a library .....	2-48
Understanding KITE	
Project components .....	2-7
Devices .....	2-7
Sites .....	2-7
Subsites .....	2-7
Tests .....	2-7
Project defined .....	2-6
Keithley Model 590 CV Analyzer DUT connections ..	
4-38	
Keithley User Library Tool (KULT) .....	7-3
KULT window .....	7-3
Understanding	
module code entry area .....	7-5
module identification area .....	7-4
module parameter display area .....	7-5
tab areas .....	7-6
terminating brace area .....	7-6
Tutorial 2	
Creating a user module that returns data	
arrays .....	7-28
Checking the VSweep user module .....	7-32
Compiling and building the VSweep user	
module .....	7-32
Documenting the VSweep user module ..	
7-32	
Entering the VSweep user module code	
7-28	
Entering the VSweep user module header	
files .....	7-31
Entering the VSweep user module	
parameters .....	7-29
Entering the VSweep user module return	
type .....	7-28
Naming a new user library and the new	
VSweep user module .....	7-28
Saving the VSweep user module ..	
7-32	
Tutorial 3	
Calling one user module from within	
another .....	7-34
Calling an independent user module from	
the VSweepBeep user module ..	7-35
Checking the VSweepBeep user module	
7-38	
Compiling and building the VSweepBeep	
user module .....	7-38
Creating the VSweepBeep user module	
by copying an existing user module ..	
7-34	
Specifying user library dependencies in	
the VSweepBeep user module ..	7-37
Keyboard connections .....	1-8
KI	
4200 SCS KXCI Settings tab .....	6-4
KITE (Keithley Interactive Test Environment)	
Configuring the Project ITMs	
Connections, matching physical and virtual	
2-17	
Forcing functions	
configuring for each device terminal ..	2-17
KITE interface overview .....	2-4
KITE ITM configuration .....	3-13
KITE Library management .....	2-45
KITE project folders .....	2-29
KITE project structure .....	2-3
KPulse	
Custom File Arb waveforms (Full-Arb) .....	5-8
Segment Arb waveforms .....	5-6
Setup and help .....	5-2
Standard Pulse waveforms .....	5-4
Starting KPulse .....	5-2
Triggering .....	5-3
Waveform types .....	5-11
KULT	
Compile message box with error message ..	7-22
entered array-size parameters .....	7-30
file menu .....	7-11
window after entering and applying code and	
parameters .....	7-20
window after naming user library .....	7-16
window after naming user module .....	7-17
window, Blank .....	7-16
KULT Tutorials .....	7-14
KULT window .....	7-4
KXCI	
console .....	6-5
KXCI Settings tab .....	6-3

## L

LAN connections .....	1-11
Leveraging the Default Project .....	3-9
Library build message box .....	7-35
Library Dependencies list box .....	7-38
Line power receptacle .....	1-6
Line-item descriptions for a .kdv file .....	2-28
Locate and Run the "vds-id" test module .....	1-48



**M**

Measure settings .....	3-16
Measurement Hardware	
SMU with Model 4200-PA overview	
Basic characteristics	
Voltage characteristics .....	1-19
Model 4200-CVU card .....	1-23
Force-measure timing .....	1-26
Measurement circuit .....	1-24
Measurement functions .....	1-24
Measurement overview .....	1-23
Model 4200-PA	
connectors .....	1-22
Model 8101- PIV test fixture .....	3-30
Model 8101-PIV test fixture .....	3-30
Models 4200-SMU and 4210-SMU	
connectors .....	1-17
current characteristics .....	1-14
voltage characteristics .....	1-15
Modified project plan settings .....	4-22

**N**

Naming a new user module .....	7-16
NBTI (negative bias temperature instability) .....	3-75
NBTI process flow .....	3-93
New Project menu selection .....	4-39
New UTM inserted in the project plan .....	7-26
New v_sweep_chk check UTM inserted in the project plan .....	7-32

**O**

Open KITE .....	1-46
Open KITE and the "ivswitch" project .....	4-11
Options menu .....	7-13

**P**

Parameter entries for the called user module,	
TwoTonesTwice .....	7-36
Parameter Name field .....	7-7
Parameters tab area .....	7-6
Add, Delete, Apply pop-up menu .....	7-7
with example entries for the RDSon42XX user module .....	7-6
PGU initialization .....	4-33
PGU stress pulse specifications .....	4-34
pgu-trigger test - trigger the burst of stress pulses .....	4-34
PIV-A test connections .....	3-28
Power Divider .....	1-33, 1-34
Powering up the 4200-SCS .....	1-5
prbgen user modules .....	4-17
Primary differences between an ITM and a UTM .....	2-9
Printer connections .....	1-10
Probe station connections .....	1-10
prober-init .....	4-23
prober-prompt test and dialog window .....	4-25
prober-separate .....	4-25
Programming and erasing flash memory .....	3-94
Project Navigator - "ivpgswitch" project .....	4-31
Project Navigator - "ivswitch" project .....	4-11
Project Navigator - probesubsites project .....	4-21
Project Navigator Checkboxes .....	2-21
Pulse generator card .....	1-27

Pulse generator configuration .....	4-29
Pulse IV for CMOS	
Model 4200-PIV-A .....	3-27
Pulse Mode (SMUs) .....	2-16
Pulse projects	
Power Divider .....	1-33, 1-34
RBT .....	1-33, 1-34
Pulse source-measure	
Full-Arb (FARB) .....	1-31
KPulse .....	1-32
Pulse generator settings .....	1-32
Scope card settings .....	1-36
Standard Pulse .....	1-28
Pulse source-measure UTMs	
PulseVulib .....	3-46
cal_pulseiv .....	3-46
scopeshot_pulseiv .....	3-59, 3-61
scopeshot_cal_pulseiv .....	3-59
scopeshot_pulseiv_demo .....	3-63
Vdid_Pulse_DC_Family_pulseiv .....	3-50
vdsid_pulseiv .....	3-48
vdsid_pulseiv_demo .....	3-63
Vgid_DC_Pulse_pulseiv .....	3-55
vgsid_pulseiv .....	3-53
vgsid_pulseiv_demo .....	3-63

**R**

RBT .....	1-33, 1-34
Rear Panel .....	1-12
Rear panel .....	1-13
Relationships between KULT and KITE and between user libraries, user modules, and UTMs .....	7-3
Remote Bias Tee (RBT) and 3-port Power Divider ...	1-32
Remote control overview (GPIO, Ethernet, remote desktop) .....	6-2
Run "vds-id" test .....	1-50
Running the test .....	2-23

**S**

Sample wafer organization .....	4-16
Save the system configuration .....	4-11
Saving the system configuration ....	4-20, 4-31, 4-39
Saving the user module .....	7-21
SCP2 (Oscilloscope) .....	1-35
Second "connect" test - connects the device to the PGU .....	4-33
Segment Arb .....	1-29
Segment Stress/Measure Mode .....	3-81
Selected device and destination folder .....	2-47
Selected ITM and destination folder .....	2-50
Selecting a test .....	2-22
Setting the AC drive conditions .....	3-16
Setting the DC bias conditions .....	3-16
Setting the Model 590 GPIB address .....	4-39
Signal paths for "2-wireresistor" tests .....	4-13
Signal paths for "3terminal-npn-bjt" tests .....	4-12
Signal paths for "4terminal-n-fet" tests .....	4-12
Signal paths for "capacitor" test .....	4-13
Signal paths for "diode" tests .....	4-13
Signal paths for the pre and post stress tests ....	4-32
Signal paths to apply the pulse stress .....	4-33
SMU terminals and connectors .....	1-16
SMU with Model 4200-PA	



voltage characteristics .....	1-19
SMU with Model 4200-PA current characteristics .....	1-19
Source-Measure Hardware	
Ground unit (GNDU) overview	
Basic characteristics .....	1-39
Basic circuit configurations .....	1-40
Ground unit connections .....	1-40
Ground unit DUT connections .....	1-40
Ground unit terminals and connectors .....	1-41
Chassis ground .....	1-42
COMMON terminal .....	1-42
FORCE terminal .....	1-42
SENSE terminal .....	1-42
Models 4200-SMU and 4210-SMU overview .....	1-14
Basic characteristics .....	1-14
Current characteristics .....	1-14
Voltage characteristics .....	1-15
Basic SMU circuit configuration .....	1-15
SMU terminals and connectors	
FORCE terminal .....	1-17
PA CNTRL connector .....	1-18
SENSE LO terminal .....	1-18
SENSE terminal .....	1-17
SMU with Model 4200-PA overview .....	1-18
Basic characteristics .....	1-18
Current characteristics .....	1-18
Basic SMU/PreAmp circuit configuration .....	1-19
PreAmp terminals and connectors .....	1-20
FORCE terminal .....	1-21
PreAmp CONTROL connector .....	1-22
SENSE terminal .....	1-22
Starting KULT .....	7-15
Starting KXCI and the GPIB command interpreter .....	6-4
Status .....	3-18
Stressing	
Segment Stress/Measure Mode .....	3-81
Submit device dialog box .....	2-47
Submit test dialog box .....	2-50
Submitting devices, ITMs, and UTMs to libraries .....	2-45
Subsite Cycling	
Segment Stress/Measure Mode .....	3-81
Subsite cycling	
Configuration .....	3-67
Configuration sequence .....	3-92
Cycle Mode	
Subsite Data sheet .....	3-87
Subsite Graph .....	3-91
Device connections .....	3-66
Mode selection .....	3-68
Running cycle subsite .....	3-86
Segment Stress/Measure Mode configuration .....	3-83
Stress/Measure Mode	
Configuring device stress properties .....	3-71
Subsite Data sheet .....	3-88
Subsite Graph .....	3-92
Subsite Setup tab .....	3-67
Timing setup .....	3-69, 3-70
Subsite Data sheet	
Cycle Mode .....	3-87
Stress/Measure Mode .....	3-87, 3-88
Subsite Graph tab	
Cycle Mode .....	3-91
Stress/Measure Mode .....	3-92
Subsite Plan .....	2-22
Subsite Plan containing the Device Plan to be submitted .....	2-45
Subsite Plan window containing the Device Plan to be submitted .....	2-46
Subsite Settings window	
Cycle Mode .....	3-89
Stress/Measure Mode .....	3-90
Subsite Setup tab .....	3-67
Supplied .....	3-29
Supplied interconnect parts .....	3-28
Supported external equipment table .....	4-4
Supported probers .....	4-17
Sweep mode (triggering) .....	1-37
System configuration for the "probesubsites" project .....	4-18
System configuration with external instruments .....	4-3
System Connections .....	1-7
System connections	
Connecting a LAN .....	1-10
Connecting a printer .....	1-10
Connecting a prober .....	1-9
Connecting GPIB instruments .....	1-9
Connecting the keyboard and mouse (optional) .....	1-7

## T

### Tab area

Compile error message in the Build tab area .....	7-22
Default Includes .....	7-8, 7-20
Description .....	7-21
Pop-up edit menu for the Description .....	7-9
Successful-compilation message displayed in Build tab area .....	7-24
Target applications and test projects for PIV-Q .....	3-27
TDDb (time dependent dielectric breakdown) .....	3-75
Test conditions .....	3-17
Test definition .....	1-49
Test library	
access selection .....	2-30
results folder .....	2-31
Test sequence .....	4-26
Test sequence for "ivpgswitch" project .....	4-35
Test system for "ivpgswitch" project .....	4-28
Testing with less than $\pm 20$ volts .....	1-43
Testing with more than $\pm 20$ volts .....	1-44
Tests (ITMs and UTMs) .....	2-21
Three terminal NPN BJT tests .....	3-6
Tutorial #1	
Creating a new user library and user module .....	7-15
Tutorial #2	
Creating a user module that returns data arrays .....	7-28
Two wire resistor test .....	3-6
Typical	
test fixture .....	1-44
Typical CV curve .....	4-37



**U**

Unconfigured Graph Definition window for the "vds-id"	
ITM .....	2-44
Unconfigured Graph Definition window for the vds-id	
ITM .....	2-42
Unconfigured UTM message .....	2-48
Understanding Append worksheets .....	2-36
Understanding KITE .....	2-3
Understanding the Formula combo box of the Data	
worksheet .....	2-34
Understanding the menus .....	7-10
Understanding the status bar .....	7-10
Unpacking and inspection	
Inspection for damage .....	1-3
Manual package .....	1-3
Repacking for shipment .....	1-3
Shipment contents .....	1-3
Unpacking the 4200-SCS .....	1-3
Usrlib subdirectory .....	2-31
UTM (User Test Modules)	
Configured v_sweep_chk .....	7-33
Definition tab .....	2-12
UTM Configured .....	7-27

**V**

Vds-id	
graph after configuring its Graph Definition	
window .....	2-44
View and Save the Graph Data .....	1-51
View and save the Sheet data .....	1-50
VSweep entries for the two voltage input parameters	
7-29	
VSweep user-module window after entering and	
applying code and parameters .....	7-31

**W**

What if my equipment is not listed in KCON .....	4-44
What is the PIV-A PulseIV Package .....	3-27



Specifications are subject to change without notice.  
All Keithley trademarks and trade names are the property of Keithley Instruments, Inc.  
All other trademarks and trade names are the property of their respective companies.

---



A G R E A T E R M E A S U R E O F C O N F I D E N C E

**Keithley Instruments, Inc.**

**Corporate Headquarters** • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY • [www.keithley.com](http://www.keithley.com)