

**KEITHLEY**

# Model 2800 RF Power Analyzer

## Reference Manual

# WARRANTY

Keithley Instruments, Inc. warrants this product to be free from defects in material and workmanship for a period of 1 year from date of shipment.

Keithley Instruments, Inc. warrants the following items for 90 days from the date of shipment: probes, cables, rechargeable batteries, diskettes, and documentation.

During the warranty period, we will, at our option, either repair or replace any product that proves to be defective.

To exercise this warranty, write or call your local Keithley representative, or contact Keithley headquarters in Cleveland, Ohio. You will be given prompt assistance and return instructions. Send the product, transportation prepaid, to the indicated service facility. Repairs will be made and the product returned, transportation prepaid. Repaired or replaced products are warranted for the balance of the original warranty period, or at least 90 days.

## LIMITATION OF WARRANTY

This warranty does not apply to defects resulting from product modification without Keithley's express written consent, or misuse of any product or part. This warranty also does not apply to fuses, software, non-rechargeable batteries, damage from battery leakage, or problems arising from normal wear or failure to follow instructions.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES.

NEITHER KEITHLEY INSTRUMENTS, INC. NOR ANY OF ITS EMPLOYEES SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF ITS INSTRUMENTS AND SOFTWARE EVEN IF KEITHLEY INSTRUMENTS, INC., HAS BEEN ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES. SUCH EXCLUDED DAMAGES SHALL INCLUDE, BUT ARE NOT LIMITED TO: COSTS OF REMOVAL AND INSTALLATION, LOSSES SUSTAINED AS THE RESULT OF INJURY TO ANY PERSON, OR DAMAGE TO PROPERTY.

**KEITHLEY**

A GREATER MEASURE OF CONFIDENCE

**Keithley Instruments, Inc.**

**Corporate Headquarters** • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY (534-8453) • [www.keithley.com](http://www.keithley.com)

**Belgium:** Sint-Pieters-Leeuw • 02-363 00 40 • Fax: 02-363 00 64 • [www.keithley.nl](http://www.keithley.nl)

**China:** Beijing • 8610-82251886 • Fax: 8610-82251892 • [www.keithley.com.cn](http://www.keithley.com.cn)

**Finland:** Helsinki • 09-5306-6560 • Fax: 09-5306-6565 • [www.keithley.com](http://www.keithley.com)

**France:** Saint-Aubin • 01-64 53 20 20 • Fax: 01-60 11 77 26 • [www.keithley.fr](http://www.keithley.fr)

**Germany:** Germering • 089/84 93 07-40 • Fax: 089/84 93 07-34 • [www.keithley.de](http://www.keithley.de)

**Great Britain:** Theale • 0118 929 7500 • Fax: 0118 929 7519 • [www.keithley.co.uk](http://www.keithley.co.uk)

**India:** Bangalore • 080 212 8027 • Fax: 080 212 8005 • [www.keithley.com](http://www.keithley.com)

**Italy:** Milano • 02-48 39 16 01 • Fax: 02- 48 39 16 28 • [www.keithley.it](http://www.keithley.it)

**Japan:** Tokyo • 81-3-5733-7555 • Fax: 81-3-5733-7556 • [www.keithley.jp](http://www.keithley.jp)

**Korea:** Seoul • 82-2-574-7778 • Fax: 82-2-574-7838 • [www.keithley.com](http://www.keithley.com)

**Netherlands:** Gorinchem • 0183-635333 • Fax: 0183-630821 • [www.keithley.nl](http://www.keithley.nl)

**Singapore:** Singapore • 65-6747-9077 • Fax: 65-6747-2991 • [www.keithley.com](http://www.keithley.com)

**Sweden:** Solna • 08-509 04 600 • Fax: 08-655 26 10 • [www.keithley.com](http://www.keithley.com)

**Taiwan:** Hsinchu • 886-3-572-9077 • Fax: 886-3-572-9031 • [www.keithley.com.tw](http://www.keithley.com.tw)

---

# Model 2800 RF Power Analyzer Reference Manual

©2003, Keithley Instruments, Inc.  
All rights reserved.  
Cleveland, Ohio, U.S.A.  
Second Printing, January 2004  
Document Number: 2800-901-01 Rev. B

---

# Manual Print History

The print history shown below lists the printing dates of all Revisions and Addenda created for this manual. The Revision Level letter increases alphabetically as the manual undergoes subsequent updates. Addenda, which are released between Revisions, contain important change information that the user should incorporate immediately into the manual. Addenda are numbered sequentially. When a new Revision is created, all Addenda associated with the previous Revision of the manual are incorporated into the new Revision of the manual. Each new Revision includes a revised copy of this print history page.

Revision A (Document Number 2800-901-01) ..... January 2003  
Revision B (Document Number 2800-901-01) ..... January 2004

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 manual for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product 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 manual. 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, and perform safe installations and repairs of products. Only properly trained service personnel may perform installation and service procedures.

Keithley 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 Manual.

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 1000 volts, **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, make sure 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 ca-

bles 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 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 indicates that the user should refer to the operating instructions located in the manual.

The  symbol on an instrument shows that it can source or measure 1000 volts or more, including the combined effect of normal and common mode voltages. Use standard safety precautions to avoid personal contact with these voltages.

The  symbol indicates a connection terminal to the equipment frame.

The **WARNING** heading in a manual 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 a manual 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., 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.

---

# Table of Contents

## 1 Remote (GPIB) Operation

|  |      |
|--|------|
| Differences between remote and local operation ..... | 1-2  |
| Front panel aspects of GPIB operation .....          | 1-2  |
| Error and status messages .....                      | 1-2  |
| GPIB status indicators .....                         | 1-2  |
| LOCAL key .....                                      | 1-3  |
| Selecting GPIB primary address and protocol .....    | 1-4  |
| GPIB address .....                                   | 1-4  |
| GPIB protocols .....                                 | 1-4  |
| GPIB connections .....                               | 1-4  |
| Programming commands .....                           | 1-7  |
| Programming syntax .....                             | 1-8  |
| Command words .....                                  | 1-8  |
| Query commands .....                                 | 1-10 |
| Case sensitivity .....                               | 1-10 |
| Long-form and short-form versions .....              | 1-10 |
| Short-form rules .....                               | 1-11 |
| Program messages .....                               | 1-11 |
| Response messages .....                              | 1-14 |
| Message exchange protocol .....                      | 1-15 |

## 2 GPIB Commands

|                                     |      |
|-------------------------------------|------|
| SCPI commands .....                 | 2-2  |
| CALCulate3 subsystem commands ..... | 2-2  |
| DISPlay subsystem commands .....    | 2-4  |
| FORMat subsystem commands .....     | 2-6  |
| INSTrument subsystem commands ..... | 2-14 |
| SENSE subsystem commands .....      | 2-15 |
| SOURce2 subsystem commands .....    | 2-31 |
| STATus subsystem commands .....     | 2-34 |
| SYSTem subsystem commands .....     | 2-37 |
| TRACe subsystem commands .....      | 2-42 |
| Trigger commands .....              | 2-44 |
| UNIT subsystem commands .....       | 2-47 |
| Signal-oriented commands .....      | 2-48 |
| Common commands .....               | 2-50 |
| General bus commands .....          | 2-55 |

---

### **3 Triggering, Status Structure, and Digital I/O**

|   |      |
|---|------|
| Triggering .....                            | 3-2  |
| Trigger model .....                         | 3-2  |
| External triggering .....                   | 3-5  |
| Trigger configuration .....                 | 3-8  |
| Status structure .....                      | 3-13 |
| Operational overview .....                  | 3-13 |
| Clearing registers and queues .....         | 3-15 |
| Programming and reading registers .....     | 3-16 |
| Status byte and service request (SRQ) ..... | 3-19 |
| Status register sets .....                  | 3-23 |
| Queues .....                                | 3-32 |
| Digital I/O .....                           | 3-35 |
| Digital input .....                         | 3-35 |
| Digital outputs .....                       | 3-36 |
| Setting digital output lines .....          | 3-40 |

### **4 Functionality Verification**

|  |      |
|--|------|
| Introduction .....                                 | 4-2  |
| Test requirements .....                            | 4-3  |
| Environmental conditions .....                     | 4-3  |
| Warm-up period .....                               | 4-3  |
| Line power .....                                   | 4-3  |
| Recommended test equipment .....                   | 4-4  |
| Calculating test limits .....                      | 4-5  |
| Example reading limit calculation .....            | 4-5  |
| Restoring factory defaults .....                   | 4-5  |
| Test summary and considerations .....              | 4-6  |
| Functionality test summary .....                   | 4-6  |
| Test considerations .....                          | 4-6  |
| Functionality tests .....                          | 4-6  |
| VSWR measurements .....                            | 4-6  |
| Noise floor .....                                  | 4-8  |
| Reference standard frequency accuracy .....        | 4-10 |
| Primary channel power measurement accuracy .....   | 4-10 |
| Adjacent channel power measurement accuracy .....  | 4-14 |
| Alternate channel power measurement accuracy ..... | 4-15 |



---

|          |  |      |
|----------|--|------|
| <b>5</b> | <b>Disassembly</b>                         |      |
|          | Introduction .....                         | 5-2  |
|          | Handling and cleaning .....                | 5-2  |
|          | Handling PC boards and modules .....       | 5-2  |
|          | Static sensitive devices .....             | 5-3  |
|          | Disassembly procedures .....               | 5-4  |
|          | Module locations .....                     | 5-4  |
|          | Case cover removal .....                   | 5-5  |
|          | RF DownConverter module removal .....      | 5-6  |
|          | Local oscillator amplifier removal .....   | 5-6  |
|          | Reference module removal .....             | 5-6  |
|          | Frequency synthesizer module removal ..... | 5-7  |
|          | DSP module removal .....                   | 5-7  |
|          | Dither module removal .....                | 5-7  |
|          | Digital board removal .....                | 5-8  |
|          | Removing power components .....            | 5-8  |
|          | Front panel disassembly .....              | 5-9  |
|          | Fan removal .....                          | 5-9  |
|          | Instrument reassembly .....                | 5-10 |
|          | Power module wire connections .....        | 5-10 |

**A Specifications**

**B Status and Error Messages**

**C GPIB Protocols**

|  |                                       |     |
|--|---------------------------------------|-----|
|  | Introduction .....                    | C-2 |
|  | Selecting the GPIB protocol .....     | C-2 |
|  | Remote operation .....                | C-2 |
|  | Protocol differences .....            | C-3 |
|  | Message exchange protocol (MEP) ..... | C-3 |
|  | Using SCPI-based programs .....       | C-3 |
|  | NRFD hold-off .....                   | C-4 |
|  | Trigger-on-talk .....                 | C-4 |
|  | Message available .....               | C-4 |
|  | General operation notes .....         | C-4 |

---

  
**D****GPIB overview**

|                                    |      |
|------------------------------------|------|
| Introduction .....                 | D-2  |
| Bus description .....              | D-2  |
| Bus lines .....                    | D-4  |
| Data lines .....                   | D-5  |
| Bus management lines .....         | D-5  |
| Handshake lines .....              | D-5  |
| Bus commands .....                 | D-6  |
| Uniline commands .....             | D-7  |
| Universal multiline commands ..... | D-8  |
| Addressed multiline commands ..... | D-8  |
| Address commands .....             | D-9  |
| Unaddress commands .....           | D-10 |
| Common commands .....              | D-10 |
| SCPI commands .....                | D-10 |
| Command codes .....                | D-11 |
| Typical command sequences .....    | D-13 |
| IEEE command groups .....          | D-14 |
| Interface function codes .....     | D-15 |

---

# List of Illustrations

|             |  |      |
|-------------|--|------|
| <b>1</b>    | <b>Remote (GPIB) Operation</b>   |      |
| Figure 1-1  | IEEE-488 connector .....   | 1-4  |
| Figure 1-2  | IEEE-488 connections .....   | 1-5  |
| Figure 1-3  | IEEE-488 connector location .....  | 1-6  |
| <b>2</b>    | <b>GPIB Commands</b>   |      |
| Figure 2-1  | ASCII data format (power function and buffer) .....                                  | 2-8  |
| Figure 2-2  | IEEE-754 data formats .....  | 2-9  |
| Figure 2-3  | Key-press codes .....  | 2-39 |
| <b>3</b>    | <b>Triggering, Status Structure, and Digital I/O</b>                                 |      |
| Figure 3-1  | Trigger model (remote operation) .....   | 3-3  |
| Figure 3-2  | EXTERNAL TRIGGER and METER COMPLETE connectors .....                                 | 3-5  |
| Figure 3-3  | Trigger input pulse specifications (External Trigger) .....                          | 3-6  |
| Figure 3-4  | Trigger output pulse specifications (Meter Complete) .....                           | 3-7  |
| Figure 3-5  | Typical external triggering connections .....  | 3-7  |
| Figure 3-6  | Model 2800 status register structure .....   | 3-14 |
| Figure 3-7  | 16-bit status register .....   | 3-17 |
| Figure 3-8  | Status byte and service request (SRQ) .....  | 3-19 |
| Figure 3-9  | Standard event status .....  | 3-24 |
| Figure 3-10 | Operation event status .....   | 3-26 |
| Figure 3-11 | Measurement event status .....   | 3-28 |
| Figure 3-12 | Questionable event status .....  | 3-29 |
| Figure 3-13 | Digital I/O port .....   | 3-35 |
| Figure 3-14 | Digital output port simplified schematic .....                                       | 3-36 |
| Figure 3-15 | Controlling externally powered relays .....  | 3-38 |
| Figure 3-16 | NAND gate control .....  | 3-39 |
| <b>4</b>    | <b>Functionality Verification</b>  |      |
| Figure 4-1  | Connections for VSWR tests .....   | 4-7  |
| Figure 4-2  | Connections for noise floor tests .....  | 4-9  |
| Figure 4-3  | Connections for reference standard frequency accuracy .....                          | 4-10 |
| Figure 4-4  | Connections for power splitter error measurements .....                              | 4-11 |
| Figure 4-5  | Connections for primary, adjacent, and alternate channel<br>power measurements ..... | 4-13 |
| <b>5</b>    | <b>Disassembly</b>   |      |
| Figure 5-1  | Top module locations .....   | 5-4  |
| Figure 5-2  | Bottom module locations .....  | 5-5  |

---

## **D**      **GPIB overview**

|            |                                   |                      |
|------------|-----------------------------------|----------------------|
| Figure D-1 | GPIB configuration .....          | <a href="#">D-3</a>  |
| Figure D-2 | IEEE-488 handshake sequence ..... | <a href="#">D-6</a>  |
| Figure D-3 | Command codes .....               | <a href="#">D-12</a> |

---

# List of Tables

|            |   |      |
|------------|---|------|
| <b>2</b>   | <b>GPIB Commands</b>  |      |
| Table 2-1  | CALCulate3 command summary .....                                  | 2-2  |
| Table 2-2  | DISPlay command summary .....                                     | 2-4  |
| Table 2-3  | FORMat command summary .....                                      | 2-6  |
| Table 2-4  | INSTRument command summary .....                                  | 2-14 |
| Table 2-5  | SENSe command summary .....                                       | 2-15 |
| Table 2-6  | SOURce2 command summary .....                                     | 2-31 |
| Table 2-7  | Output patterns for NRf parameter .....                           | 2-33 |
| Table 2-8  | STATus command summary .....                                      | 2-34 |
| Table 2-9  | SYSTem command summary .....                                      | 2-37 |
| Table 2-10 | Key-press codes .....   | 2-39 |
| Table 2-11 | TRACe command summary .....                                       | 2-42 |
| Table 2-12 | Trigger command summary .....                                     | 2-44 |
| Table 2-13 | UNIT command summary .....  | 2-47 |
| Table 2-14 | Signal-oriented measurement command summary .....                 | 2-48 |
| Table 2-15 | IEEE-488.2 common commands and queries .....                      | 2-50 |
| Table 2-16 | General bus commands and associated statements .....              | 2-55 |
| <b>3</b>   | <b>Triggering, Status Structure, and Digital I/O</b>              |      |
| Table 3-1  | SCPI commands — triggering .....                                  | 3-9  |
| Table 3-2  | Common and SCPI commands to reset registers and clear queues .... | 3-16 |
| Table 3-3  | Data format commands for reading status register .....            | 3-18 |
| Table 3-4  | Status Byte and Service Request Enable Register commands .....    | 3-22 |
| Table 3-5  | Condition register commands .....                                 | 3-30 |
| Table 3-6  | Event register commands .....                                     | 3-30 |
| Table 3-7  | Event enable registers commands .....                             | 3-31 |
| Table 3-8  | Error queue commands .....  | 3-34 |
| Table 3-9  | Output patterns for NRf parameter .....                           | 3-40 |
| Table 3-10 | SOURce2 command summary .....                                     | 3-41 |
| <b>4</b>   | <b>Functionality Verification</b>                                 |      |
| Table 4-1  | Recommended test equipment .....                                  | 4-4  |
| Table 4-2  | Noise floor test parameters .....                                 | 4-9  |
| Table 4-3  | Power splitter error summary .....                                | 4-11 |
| Table 4-4  | Primary channel power measurement test summary .....              | 4-13 |
| Table 4-5  | Upper adjacent channel power measurement test summary .....       | 4-15 |
| Table 4-6  | Upper alternate channel power measurement test summary .....      | 4-16 |
| <b>5</b>   | <b>Disassembly</b>  |      |
| Table 5-1  | Power module wire colors .....                                    | 5-10 |

---

## **B**            **Status and Error Messages**

|           |                                 |                     |
|-----------|---------------------------------|---------------------|
| Table B-1 | Status and error messages ..... | <a href="#">B-2</a> |
|-----------|---------------------------------|---------------------|

## **D**            **GPIB overview**

|           |  |                      |
|-----------|--|----------------------|
| Table D-1 | IEEE-488 bus command summary .....         | <a href="#">D-7</a>  |
| Table D-2 | Hexadecimal and decimal command codes..... | <a href="#">D-11</a> |
| Table D-3 | Typical addressed command sequence .....   | <a href="#">D-13</a> |
| Table D-4 | Typical addressed command sequence .....   | <a href="#">D-13</a> |
| Table D-5 | IEEE command groups .....                  | <a href="#">D-14</a> |
| Table D-6 | Model 2800 interface function codes.....   | <a href="#">D-15</a> |

# Remote (GPIB) Operation

---

## Section topics

Differences between remote and local operation

Front panel aspects of GPIB operation

Error and status messages

GPIB status indicators

LOCAL key

Selecting GPIB primary address and protocol

GPIB connections

Programming commands

Programming syntax

Command words

Query commands

Case sensitivity

Long-form and short-form versions

Short-form rules

Program messages

Response messages

Message exchange protocol

## **NOTE**

*A software driver to control the Model 2800 is available as a free download on the Keithley website ([www.keithley.com](http://www.keithley.com)). This driver is suitable for use with LabView, LabWindows CVI, Visual Basic, and C++. The driver is available in the download section of the 2800 product page on the website.*

# Differences between remote and local operation

Sending a GPIB command places the Model 2800 in the remote mode of operation. While in remote, all front panel control keys (except LOCAL) are disabled. Pressing the LOCAL key returns the instrument to the local mode.

When changing from local to remote operation, the following occurs:

- The Model 2800 stops taking readings and is placed in the idle state (see “[Triggering](#)” in [Section 3](#) for details).
- All menus are exited.
- All pending front panel operations are aborted.
- All other settings remain unaffected.
- Data in the sample buffer is lost (e.g., FETCh? will not return any data until a reading is taken while in remote).

When changing from remote to local operation, the following occurs:

- The Model 2800 goes from idle to the continuous measurement state (see “[Triggering](#)” in [Section 3](#) for details).
- All user-defined displays are disabled. The display is re-enabled if it was previously turned off (see “[DISPlay subsystem commands](#)” in [Section 2](#) for details).
- If the Immediate trigger source is selected, readings are continuously taken (see “[Triggering](#)” in [Section 3](#) for details).

## Front panel aspects of GPIB operation

### Error and status messages

See [Appendix B](#) for a list of error and status messages associated with IEEE-488 programming. The instrument can be programmed to generate an SRQ and command queries can be performed to check for specific error conditions.

### GPIB status indicators

The REM (remote), TALK (talk), LSTN (listen), and SRQ (service request) annunciators show the GPIB bus status. Each of these indicators is described below.



## REM

This indicator shows when the instrument is in the remote state. REM does not necessarily indicate the state of the bus REN line, as the instrument must be addressed to listen with REN true before the REM indicator turns on. When the instrument is in remote, all front panel keys (except for the **LOCAL** key) are locked out. When REM is turned off, the instrument is in the local state and front panel operation is restored.

**NOTE**      *If LLO is in effect, **LOCAL** will be locked out.*

## TALK

This indicator is on when the instrument is in the talker active state. Place the unit in the talk state by addressing it to talk with the correct MTA (My Talk Address) command. TALK is off when the unit is in the talker idle state. Place the unit in the talker idle state by sending an UNT (Untalk) command, addressing it to listen, or sending the IFC (Interface Clear) command.

## LSTN

This indicator is on when the Model 2800 is in the listener active state, which is activated by addressing the instrument to listen with the correct MLA (My Listen Address) command. LSTN is off when the unit is in the listener idle state. Place the unit in the listener idle state by sending UNL (Unlisten), addressing it to talk, or sending IFC (Interface Clear) command over the bus.

## SRQ

You can program the instrument to generate a service request (SRQ) when one or more errors or conditions occur. When this indicator is on, a service request has been generated. This indicator stays on until the serial poll byte is read or all the conditions that caused SRQ have been cleared. See [Section 3](#), “[Status structure](#),” for more information.

## LOCAL key

The LOCAL key cancels the remote state and restores local operation of the instrument.

Pressing the LOCAL key also turns off the REM indicator and returns the display to normal if a user-defined message was displayed.

If the LLO (Local Lockout) command is in effect, the LOCAL key is also inoperative.

# Selecting GPIB primary address and protocol

## GPIB address

In order to avoid communication conflicts over the bus, each device in the GPIB system must have a unique primary address. The Model 2800 is shipped from the factory with the primary address set to 28.

The set address is displayed on power-up. It can also be viewed and changed from the GPIB option of the Main Menu (see “GPIB address and protocol” on page 1-26 of the User’s Manual).

## GPIB protocols

GPIB operation can use one of two protocols: SCPI or 488.1. Both protocols support SCPI programming. Even though the 488.1 protocol is faster, it has more restrictions than the SCPI protocol. The Model 2800 is shipped from the factory with the SCPI protocol selected. The 488.1 protocol is typically used when increased speed is required.

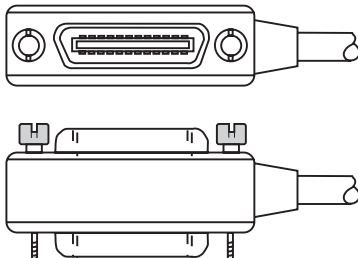
The selected protocol is displayed on power-up. It can also be viewed and changed from the GPIB option of the Main Menu (see “GPIB address and protocol” on page 1-26 of the User’s Manual).

**NOTE** Details on “*GPIB Protocols*” are provided in [Appendix C](#).

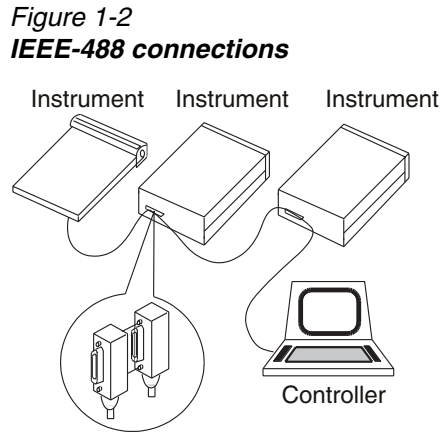
# GPIB connections

To connect the Model 2800 to the GPIB bus, use a cable equipped with standard IEEE-488 connectors as shown in [Figure 1-1](#).

*Figure 1-1*  
**IEEE-488 connector**



To allow many parallel connections to one instrument, stack the connectors as shown in Figure 1-2. Two screws are located on each connector to ensure that connections remain secure. Present standards call for metric threads, which are identified with dark-colored screws. Earlier versions have different screws, which are silver-colored. Do not use these types of connectors on the Model 2800; it is designed for metric threads.



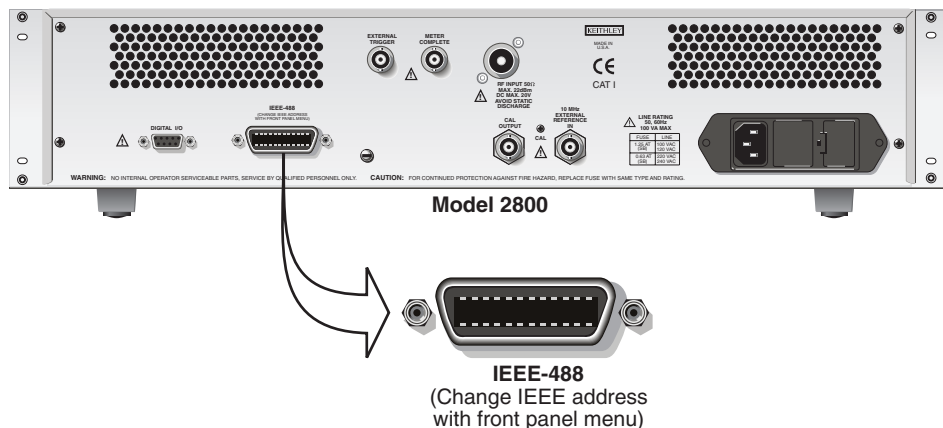
**Note:** To prevent physical damage, do not stack more than three connectors on any one unit.

**NOTE** To minimize interference caused by electromagnetic radiation, use only shielded IEEE-488 cables. Available shielded cables from Keithley are Models 7007-1 and 7007-2.

To connect the Model 2800 to the IEEE-488 bus, follow these steps:

1. Line up the cable connector with the connector located on the rear panel. [Figure 1-3](#) shows the location of the IEEE-488 connector.
2. Tighten the screws securely, making sure not to overtighten them.
3. Connect any additional connectors from other instruments as required for your application.
4. Make sure the other end of the cable is properly connected to the controller. Most controllers are equipped with an IEEE-488 style connector, but a few may require a different type of connecting cable. See your controller's instruction manual for information about properly connecting to the IEEE-488 bus.

**Figure 1-3**  
**IEEE-488 connector location**



**NOTE** You can only have 15 devices connected to an IEEE-488 bus, including the controller. The maximum cable length is either 20 meters or two meters multiplied by the number of devices, whichever is less. Not observing these limits may cause erratic bus operation.

# Programming commands

## SCPI commands

SCPI commands are used to control most instrument operations of the Model 2800. For details on these commands, see “[SCPI commands](#)” in [Section 2](#).

## Signal-oriented measurement commands

These commands are used to trigger measurements and return the readings. For details on these commands, see “[Signal-oriented commands](#)” in [Section 2](#).

## Common commands

Common commands are device commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard. For details on these commands, see “[Common commands](#)” in [Section 2](#).

## General bus commands

General bus commands are those commands, such as DCL, that have the same general meaning regardless of the instrument. For details on these commands, see “[General bus commands](#)” in [Section 2](#).

# Programming syntax

The following information covers syntax for both common commands and SCPI commands. For information not covered here, see the IEEE-488.2 and SCPI standards. See [Section 2](#) for details on “[SCPI commands](#)” and “[Common commands](#).”

## Command words

Program messages are made up of one or more command words.

### Commands and command parameters

Common commands and SCPI commands may or may not use a parameter. The following are some examples:

|                           |                           |
|---------------------------|---------------------------|
| *RCL <NRf>                | Parameter <NRf> required  |
| TRACe:FEED:CONTRol <name> | Parameter <name> required |
| SYSTem:PRESet             | No parameter used         |

**NOTE** *At least one space between the command word and the parameter is required.*

**Brackets [ ]** — Some command words are enclosed in brackets ([ ]). These brackets are used to denote an optional command word that does not need to be included in the program message. For example:

INITiate[:IMMEDIATE]

These brackets indicate that IMMEDIATE is implied (optional) and does not have to be used. Therefore, the above command can be sent in one of two ways:

INITiate

or

INITiate:IMMEDIATE

Notice that the optional command is to be used without the brackets.

**Parameter types** — The following are some of the more common parameter types:

**<b>** Boolean — Used to enable or disable an instrument operation. 0 or OFF disables the operation and 1 or ON enables the operation.  
Example:

INPut:PROTection:STATe ON      Enable input protection

**<name>** Name parameter — Select a parameter name from a listed group.  
Example:

<name> = DBM or DBC

UNIT:POWer DBM     Select dBm units

**<NRf>** Numeric representation format — This parameter is a number that can be expressed as an integer (e.g., 8), a real number (e.g., 23.6), or an exponent (2.3E6). Example:

SYSTem:KEY 12     Press **Menu** key from over the bus

**<n>** Numeric value — A numeric value parameter can consist of an NRf number or one of the following name parameters: DEFault, MINimum, MAXimum. When the DEFault parameter is used, the instrument is programmed to the \*RST default value. When the MINimum parameter is used, the instrument is programmed to the lowest allowable value. When the MAXimum parameter is used, the instrument is programmed to the largest allowable value.  
Examples:

TRIGger:COUNt 1                     Sets count to 1

TRIGger:COUNt DEFault             Sets count to 1

TRIGger:COUNt MINimum             Sets count to 1

TRIGger:COUNt MAXimum             Sets count to 2500

**<list>** List — Specify one or more numbers for a list. Example:

STATus:QUEue:ENABLE (-110:-222)     Enable errors -110 through -222

**Angle brackets < >** Angle brackets (< >) are used to denote a parameter type. Do not include the brackets in the program message. For example:

INPut:PROTection:STATe <b>

The <b> indicates a Boolean-type parameter is required. Therefore, to enable input protection, you must send the command with the ON or 1 parameter as follows:

INPut:PROTection:STATe ON

INPut:PROTection:STATe 1

## Query commands

This type of command requests (queries) the presently programmed status. It is identified by the question mark (?) at the end of the fundamental form of the command. Most commands have a query form:

TRIGger:COUNT?                      Queries the trigger count.

Most commands that require a numeric parameter (<n>) can also use the DEFault, MINimum, and MAXimum parameters for the query form. These query forms are used to determine the \*RST default value and the upper and lower limits for the fundamental command. Examples:

TRIGger:COUNT? DEFault              Queries the \*RST default value.

TRIGger:COUNT? MINimum              Queries the lowest allowable value.

TRIGger:COUNT? MAXimum              Queries the largest allowable value.

## Case sensitivity

Common commands and SCPI commands are not case sensitive. You can use upper or lower case and any case combination. Examples:

\*RST                      = \*rst  
 READ?                    = read?  
 SYSTem:PRESet = system:preset

**NOTE**                    *Using all upper case will result in slightly faster command response times.*

## Long-form and short-form versions

A SCPI command word can be sent in its long-form or short-form version. The command subsystem tables in [Section 2](#) provide the long-form version. However, the short-form version is indicated by upper case characters. Examples:

SYSTem:PRESet            long-form  
 SYST:PRES                short-form  
 SYSTem:PRES              long-form and short-form combination

Note that each command word must be in long-form or short-form, and not something in between. For example, SYSTe:PRESe is illegal and will generate an error. The command will not be executed.



## Short-form rules

Use the following rules to determine the short-form version of any SCPI command:

- If the length of the command word is four letters or less, no short form version exists. Example:  
:data? = :data?

These rules apply to command words that exceed four letters:

- If the fourth letter of the command word is a vowel (including “y”), delete it and all the letters after it. Example:  
immediate = imm
- If the fourth letter of the command word is a consonant, retain it but drop all the letters after it. Example:  
format = form
- If the command contains a question mark (; query) or a non-optional number included in the command word, you must include it in the short-form version. Example:  
:delay? = :del?
- Command words or characters that are enclosed in brackets ([ ]) are optional and need not be included in the program message.

**NOTE**      *For fastest response to commands, always use short forms.*

## Program messages

A program message is made up of one or more command words sent by the computer to the instrument. Each common command is a three letter acronym preceded by an asterisk (\*). SCPI commands are categorized in the STATus subsystem and are used to explain how command words are structured to formulate program messages.

| STATus        | Path (Root)           |
|---------------|-----------------------|
| :OPERation    | Path                  |
| :ENABle <NRf> | Command and parameter |
| :ENABle?      | Query command         |
| :PRESet       | Command               |

## Single command messages

The above command structure has three levels. The first level is made up of the root command (STATus) and serves as a path. The second level is made up of another path (:OPERation) and a command (:PRESet). The third path is made up of one command for the :OPERation path. The three commands in this structure can be executed by sending three separate program messages as follows:

```
stat:oper:enab <NRf>
stat:oper:enab?
stat:pres
```

In each of the above program messages, the path pointer starts at the root command (stat) and moves down the command levels until the command is executed.

## Multiple command messages

You can send multiple command messages in the same program message as long as they are separated by semicolons (;). The following is an example showing two commands in one program message:

```
stat:oper?; stat:oper:enab <NRf>
```

When the above is sent, the first command word is recognized as the root command (stat). When the next colon is detected, the path pointer moves down to the text command level and executes the command. When the path pointer sees the colon after the semicolon (;), it resets back to the root level and starts over.

Commands that are on the same command level can be executed without having to retype the entire command path. Example:

```
stat:oper:enab <NRf>; enab?
```

After the first command (:enab) is executed, the path pointer is at the third command level in the structure. Since :enab? is also on the third level, it can be typed in without repeating the entire path name. Notice that the leading colon for :enab? is not included in the program message. If a colon were included, the path pointer would reset to the root level and expect a root command. Since :enab? is not a root command, an error would occur.

## Command path rules

- Each new program message must begin with the root command, unless it is optional (e.g., [SENSe]). If the root is optional, treat a command word on the next level as the root. For fastest operation, do not send optional data.
- A colon (:) can be used at the beginning of a program message. However, using the colon slows down execution time. Example:  
stat:pres = :stat:pres
- When the path pointer detects a colon (:) it moves down to the next command level. An exception is when the path pointer detects a semicolon (;), which is used to separate commands within the program message (see next rule).
- When the path pointer detects a colon (:) that immediately follows a semicolon (;), it resets back to the root level.
- The path pointer can only move down. It cannot be moved up a level. Executing a command at a higher level requires that you start over at the root command.

## Using common and SCPI commands in the same message

Both common commands and SCPI commands can be used in the same message as long as they are separated by semicolons (;). A common command can be executed at any command level and will not affect the path pointer. Example:

```
stat:oper:enab <NRf>; *ESE <NRf>
```

## Program message terminator (PMT)

Each program message must be terminated with an LF (line feed), EOI (end or identify), or an LF+EOI. The bus will hang if your computer does not provide this termination. The following example shows how a command program message must be terminated:

```
inp:prot:stat on <PMT>
```

## Command execution rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and, of course, is not executed.
- Valid commands that precede an invalid command in a multiple command program message are executed.
- Valid commands that follow an invalid command in a multiple command program message are ignored.

## Response messages

A response message is the message sent by the instrument to the computer in response to a query command program message.

### Sending a response message

After sending a query command, the response message is placed in the Output Queue. When the Model 2800 is then addressed to talk, the response message is sent from the Output Queue to the computer.

### Multiple response messages

If you send more than one query command in the same program message, the multiple response messages for all the queries are sent to the computer when the Model 2800 is addressed to talk. The responses are sent in the order the query commands were sent and are separated by semicolons (;). Items within the same query are separated by commas (,). The following example shows the response message for a program message that contains four single item query commands:

```
0; 1; 1; 0
```

### Response message terminator (RMT)

Each response is terminated with an LF (line feed) and EOI (end or identify). The following example shows how a multiple response message is terminated:

```
0; 1; 1; 0 <RMT>
```

## Message exchange protocol

Two rules summarize the message exchange protocol:

**Rule 1.** You must always tell the Model 2800 what to send to the computer.

The following two steps must always be performed to send information from the instrument to the computer:

1. Send the appropriate query command(s) in a program message.
2. Address the Model 2800 to talk.

**Rule 2.** The complete response message must be received by the computer before another program message can be sent to the Model 2800.

# 2

# GPIB Commands

---

## Section topics

### SCPI commands:

|                                      |            |
|--------------------------------------|------------|
| <b>CALCulate3 subsystem commands</b> | Table 2-1  |
| <b>DISPlay subsystem commands</b>    | Table 2-2  |
| <b>FORMat subsystem commands</b>     | Table 2-3  |
| <b>INSTRument subsystem commands</b> | Table 2-4  |
| <b>SENSe subsystem commands</b>      | Table 2-5  |
| <b>SOURce2 subsystem commands</b>    | Table 2-6  |
| <b>STATus subsystem commands</b>     | Table 2-8  |
| <b>SYSTem subsystem commands</b>     | Table 2-9  |
| <b>TRACe subsystem commands</b>      | Table 2-11 |
| <b>Trigger commands</b>              | Table 2-12 |
| <b>UNIT subsystem commands</b>       | Table 2-13 |

|                                 |            |
|---------------------------------|------------|
| <b>Signal-oriented commands</b> | Table 2-14 |
|---------------------------------|------------|

|                        |            |
|------------------------|------------|
| <b>Common commands</b> | Table 2-15 |
|------------------------|------------|

|                             |            |
|-----------------------------|------------|
| <b>General bus commands</b> | Table 2-16 |
|-----------------------------|------------|

**NOTE** *A software driver to control the Model 2800 is available as a free download on the Keithley website ([www.keithley.com](http://www.keithley.com)). This driver is suitable for use with LabView, LabWindows CVI, Visual Basic, and C++. The driver is available in the download section of the 2800 product page on the website.*

# SCPI commands

## CALCulate3 subsystem commands

The commands listed in [Table 2-1](#) are used to select and read the result of a statistic on readings stored in the buffer.

Table 2-1

**CALCulate3 command summary**

| Command        | Description   | Default | Ref                    | SCPI |
|----------------|---|---------|------------------------|------|
| CALCulate3     |   |         |                        | ✓    |
| :FORMat <name> | Select statistic to be returned: MEAN, SDEViation, MAXimum, MINimum, or PKPK. | MEAN    | <a href="#">A calc</a> | ✓    |
| :FORMat?       | Query selected statistic.   |         |                        | ✓    |
| :DATA?         | Return result of selected statistic.  |         | <a href="#">B calc</a> | ✓    |

### **Reference**

*A calc*

### **Select buffer statistic:**

#### **CALCulate3:FORMat <name>**

|        |   |            |                                |
|--------|---|------------|--------------------------------|
| <name> | = | MEAN       | Mean (average) calculation     |
|        | = | SDEViation | Standard deviation calculation |
|        | = | MAXimum    | Highest reading in buffer      |
|        | = | MINimum    | Lowest reading in buffer       |
|        | = | PKPK       | MAXimum - MINimum              |

This command is used to select the desired statistic for readings stored in the buffer. For details on buffer statistics, see “Buffer statistics” on page 3-33 of the User’s Manual.

*B calc***Read result of statistic:****CALCulate3:DATA?**

This command is used to perform the selected statistic operation and read the result(s). The result(s) is always returned in ASCII format.

The statistic operation will be performed for every function specified as a data element for the buffer. The FORMat:ELEMents:TRACe command (see “[FORMat subsystem commands](#)”) is used to specify the functions for the statistic operations.

**Example** – Calculate mean of readings in buffer:

```
CALC3 : FORM MEAN      ' Select mean statistic.  
CALC3 : DATA?        ' Calculate and read the result of the mean statistic.
```



## DISPlay subsystem commands

The DISPlay subsystem commands allow the user to display custom text messages. The display can also be disabled (turned off) to increase speed. These commands are listed in [Table 2-2](#).

Table 2-2

**DISPlay command summary**

| Command      | Description                                | Default    | Ref    | SCPI |
|--------------|--|------------|--------|------|
| DISPlay      |  | (see Note) |        |      |
| [:WINDow[1]] | Messages for top display:                  |            |        | ✓    |
| :TEXT        | Path to control user text messages.        |            |        | ✓    |
| [:DATA] <a>  | ASCII message “a” (up to 20 characters).   | (none)     | A disp | ✓    |
| [:DATA]?     | Query text message.                        |            |        |      |
| :STATe <b>   | Enable or disable message mode.            | OFF        | B disp | ✓    |
| :STATe?      | Query text message state.                  |            |        | ✓    |
| :DATA?       | Read data on top display.                  |            | C disp | ✓    |
| :WINDow2     | Messages for bottom display:               |            |        | ✓    |
| :TEXT        | Path to control user text messages.        |            |        | ✓    |
| [:DATA] <a>  | ASCII message “a” (up to 32 characters).   | (none)     | A disp | ✓    |
| [:DATA]?     | Query text message.                        |            |        |      |
| :STATe <b>   | Enable or disable message mode.            | OFF        | B disp | ✓    |
| :STATe?      | Query text message state.                  |            |        | ✓    |
| :DATA?       | Read data on bottom display.               |            | C disp | ✓    |
| :ENABLE <b>  | Enable or disable the front panel display. | OFF        | D disp | ✓    |
| :ENABLE?     | Query state of the display.                |            |        | ✓    |

Note: \*RST and SYSTem:PRESet have no effect on commands in this subsystem. The listed defaults are power-on defaults.

### Reference

A disp

### Define text messages:

DISPlay[:WINDow[1]]:TEXT[:DATA] <a>

Top display

DISPlay:WINDow2:TEXT[:DATA] <a>

Bottom display

<a> = ASCII message

This command defines the text message for the top and bottom displays. A message can be as long as 20 characters for the top display and 32 characters for the bottom display. A space counts as a character. Excess message characters results in an error. The characters must be enclosed in either single quotes ( ' ') or double quotes ( " " ).

*B disp***Text message control:**

|   |                |
|---|----------------|
| <b>DISPlay[:WINDow[1]]:TEXT:STATe &lt;b&gt;</b> | Top display    |
| <b>DISPlay:WINDow2:TEXT:STATe &lt;b&gt;</b>     | Bottom display |
| <b>&lt;b&gt; = ON or OFF</b>                    |                |

This command enables and disables the text message mode. When enabled, a defined message is displayed. When disabled, the text message is removed from the display.

A user defined text message remains displayed only as long as the instrument is in remote. Taking the instrument out of remote (by pressing the LOCAL key or sending GTL) cancels the message and disables the text message mode.

*C disp***Read display messages**

|                                  |                |
|----------------------------------|----------------|
| <b>DISPlay[:WINDow[1]]:DATA?</b> | Top display    |
| <b>DISPlay:WINDow2:DATA?</b>     | Bottom display |

Use these commands to read what is currently being displayed on the top and bottom displays. After sending one of these commands and addressing the Model 2800 to talk, the displayed data (not including text message) will be sent to the computer.

*D disp***Display circuitry**

**DISPlay:ENABLE <b>**  
**<b> = ON or OFF**

This command is used to enable and disable the front panel display circuitry. When disabled, the instrument operates at a higher speed. While disabled, the display is frozen with the following message:

```
FRONT PANEL DISABLED
Press Local to resume
```

All front panel controls (except LOCAL) are disabled. Normal display operation can be resumed by using the ENABLE command to enable the display or by putting the Model 2800 into local mode (press LOCAL).

**Example** – Display text message “TESTING”:

|                                 |                             |
|---------------------------------|-----------------------------|
| <b>DISP:TEXT:DATA 'TESTING'</b> | ' Define text message.      |
| <b>DISP:TEXT:STAT ON</b>        | ' Enable text message mode. |

## FORMat subsystem commands

The FORMat subsystem commands are used to select the data format for transferring instrument readings over the bus. These commands are listed in [Table 2-3](#).

Table 2-3

### FORMat command summary

| Command                   | Description  | Default | Ref                    | SCPI |
|---------------------------|--|---------|------------------------|------|
| FORMat                    |  |         |                        |      |
| :SREGister <name>         | Select data format for reading status event registers: ASCii, HEXadecimal, OCTal, or BINary. | ASCii   | <a href="#">A form</a> | ✓    |
| :SREGister?               | Query format for reading status event registers.   |         |                        | ✓    |
| [:DATA] <type>[,<length>] | Select data format: ASCii, REAL,32 or SREal.   | ASCii   | <a href="#">B form</a> | ✓    |
| [:DATA]?                  | Query data format.   |         |                        | ✓    |
| :ELEMents                 | Specify data elements for read command responses:  |         | <a href="#">C form</a> |      |
| [:POWer] <item list>      | Power function: PRIMary, ADJL, ADJU, ALTL, ALTU, TIME, STATus, or ALL.                       | ALL     |                        |      |
| :PEAK <item list>         | PEAK function: POWer, FREQuency, TIME, STATus, or ALL.                                       | ALL     |                        |      |
| :DFREquency <item list>   | $\Delta$ F function: FREQuency, TIME, STATus, or ALL.  | ALL     |                        |      |
| :TRACe <item list>        | Buffer (TRACe:DATA?): PRIMary, ADJL, ADJU, ALTL, ALTU, TIME, STATus, or ALL.                 | ALL     |                        |      |
| :ELEMents?                | Query data elements for read command responses.  |         |                        |      |
| :BORDER <name>            | Specify byte order: NORMal or SWAPped.   | (Note)  | <a href="#">D form</a> | ✓    |
| :BORDER?                  | Query byte order.  |         |                        | ✓    |
| :SOURce2 <name>           | Specify SOURce2 data format: ASCii, BINary, HEXadecimal, or OCTal.                           | ASC     | <a href="#">E form</a> |      |
| :SOURce2?                 | Query SOURce2 data format.   |         |                        |      |

Note: Byte order – \*RST default is NORMal. SYSTem:PRESet default is SWAPped.

**Reference**

A form

**Data format for reading status registers****FORMat:SREGister** <name>

|        |   |             |                    |
|--------|---|-------------|--------------------|
| <name> | = | ASCIi       | Decimal format     |
|        | = | HEXadecimal | Hexadecimal format |
|        | = | OCTal       | Octal format       |
|        | = | BINary      | Binary format      |

This command is used to set the response message format for the query commands used to read the contents of status event registers. When a status register is queried, the response message is a value that indicates which bits in the register are set. For example, if bits B5, B4, B2, B1, and B0 of a register are set (110111), the following values will be returned for the selected data format:

|             |          |                     |
|-------------|----------|---------------------|
| ASCIi       | 55       | (decimal value)     |
| Hexadecimal | #H37     | (hexadecimal value) |
| OCTal       | #Q67     | (octal value)       |
| BINary      | #B110111 | (binary value)      |

**Example** – Select binary format for reading status registers:

```
FORM:SREG BIN
```

B form

**Data format for readings****FORMat[:DATA]** <type>[,<length>]

|                   |   |          |   |
|-------------------|---|----------|---|
| <type>[,<length>] | = | ASCIi    | ASCII format                            |
|                   | = | REAL, 32 | Binary IEEE-754 single precision format |
|                   | = | SREal    | Binary IEEE-754 single precision format |

**NOTE** <length> is not used for the ASCII or SREal parameters.

The response to READ?, FETCH?, MEASure?, or TRACe:DATA? over the GPIB can be returned in either the ASCII or binary format. All other queries are returned in ASCII, regardless of the selected format.

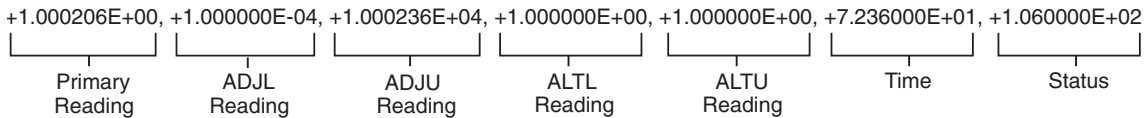
**NOTE** Regardless of which data format for output strings is selected, the instrument will only respond to input commands using the ASCII format.

### ASCII data format

The ASCII data format is in a direct readable form for the operator. Most programming languages easily convert ASCII mantissa and exponent to other formats. However, some speed is compromised to accommodate the conversion. Figure 2-1 shows an example ASCII string that includes all the data elements for the power function and the buffer (see “Data elements for readings”).

Figure 2-1 also shows the byte order of the data string. Data elements not specified by the :ELEMENTS command are simply not included in the string.

Figure 2-1  
ASCII data format (power function and buffer)



### IEEE-754 single precision format

REAL, 32 or SREal will select the binary IEEE-754 single precision data format. Figure 2-2 shows the normal byte order format for each data element. For example, if three data elements are selected, the data string for each reading conversion is made up of three 4-byte data blocks. Note that the data string for each reading conversion is preceded by a 2-byte header that is the binary equivalent of an ASCII # sign and 0. Figure 2-2 does not show the byte for the terminator that is attached to the end of each data string. Note that the byte order of the data string can be sent in reverse order (see Reference D).

During binary transfers, never un-talk the Model 2800 until after the data is read (input) to the computer. Also, to avoid erratic operation, the readings of the data string (and terminator) should be acquired in one piece. The header (#0) can be read separately before the rest of the string.

The number of bytes to be transferred can be calculated as follows:

$$\text{Bytes} = 2 + (\text{Rdgs} \times 4) + 1$$

where: 2 is the number of bytes for the header (#0).

Rdgs is the product of the number of selected data elements and trigger count.

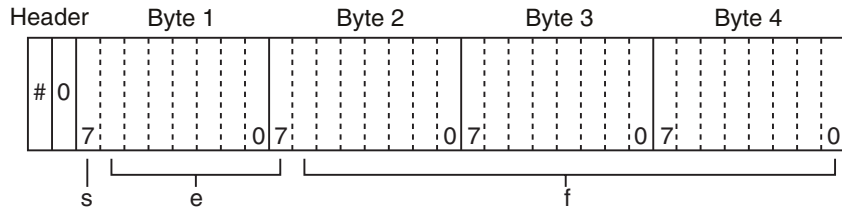
4 is the number of bytes for each reading.

1 is the byte for the terminator.

For example, assume the Model 2800 is configured to perform 10 measurements and send the 10 reading strings to the computer using the binary format.

$$\begin{aligned} \text{Bytes} &= 2 + (10 \times 4) + 1 \\ &= 43 \end{aligned}$$

**Figure 2-2**  
**IEEE-754 data formats**



- s = sign bit (0 = positive, 1 = negative)
- e = exponent bits (8)
- f = fraction bits (23)

Normal byte order shown. For swapped byte order, bytes sent in reverse order: Header, Byte 4, Byte 3, Byte 2, Byte 1.

The Header is only sent once for each measurement conversion.

**Example** – Select ASCII data format for readings:

FORM:DATA ASC

C form

## Data elements for readings

|   |                                  |
|---|----------------------------------|
| <b>FORMat:ELEMents[:POWER]</b> <item list>    | Elements for power readings      |
| <b>FORMat:ELEMents:PEAK</b> <item list>       | Elements for peak readings       |
| <b>FORMat:ELEMents:DFRequency</b> <item list> | Elements for $\Delta F$ readings |
| <b>FORMat:ELEMents:TRACe</b> <item list>      | Elements for buffer readings     |

Parameters for **:POWER** and **:TRACe**:

|              |                |                            |
|--------------|----------------|----------------------------|
| <item list>= | <b>PRIMary</b> | Primary reading            |
| =            | <b>ADJL</b>    | Adjacent (lower) reading*  |
| =            | <b>ADJU</b>    | Adjacent (upper) reading*  |
| =            | <b>ALTL</b>    | Alternate (lower) reading  |
| =            | <b>ALTL</b>    | Alternate (lower) reading  |
| =            | <b>TIME</b>    | Timestamp                  |
| =            | <b>STATus</b>  | Status word                |
| =            | <b>ALL</b>     | Include all above elements |

\*For AMPS measurements, use the ADJL element to return the carrier reading. Use the ADJU to return the upper sideband reading.

Parameters for **:PEAK**:

|              |                  |                                 |
|--------------|------------------|---------------------------------|
| <item list>= | <b>POWER</b>     | Peak power reading              |
| =            | <b>FREQuency</b> | Frequency of peak power reading |
| =            | <b>TIME</b>      | Timestamp                       |
| =            | <b>STATus</b>    | Status word                     |
| =            | <b>ALL</b>       | Include all above elements      |

Parameters for **:DFRequency**:

|              |                  |                            |
|--------------|------------------|----------------------------|
| <item list>= | <b>FREQuency</b> | $\Delta F$ reading         |
| =            | <b>TIME</b>      | Timestamp                  |
| =            | <b>STATus</b>    | Status word                |
| =            | <b>ALL</b>       | Include all above elements |

For **FORM:ELEM:POW**, **FORM:ELEM:PEAK**, and **FORM:ELEM:DFR**, the specified elements are included in the data string in response to **READ?**, **MEASure?**, or **FETCh?**.

For **FORM:ELEM:TRAC**, the specified elements are included in the data string in response to **TRACe:DATA?**.

**NOTE** Peak (**:PEAK**) and  $\Delta F$  (**:DFRequency**) readings cannot be stored in the buffer. Attempting to store Peak or  $\Delta F$  readings will generate error +831 "Invalid with PEAK/FREQ function active".

Each element in the item list must be separated by a comma (i.e. send “FORM:ELEM:POW PRIMary, ADJU, ADJL” to include the primary and adjacent reading elements in the data string). All the elements for the ASCII format for the power and buffer are shown in [Figure 2-1](#).

When a specified data element has invalid data associated with it, NAN (Not A Number) will be the response. NAN is returned as +9.91E37.

**Timestamp** – With timestamp selected, each group of readings are referenced to a point in time. The timestamp operates as a timer that starts at zero seconds when the instrument is turned on. The timestamp for each group of readings is referenced (in seconds) to the start time. After 4,294,967 seconds, the timer resets to zero and starts over.

For buffer readings (TRACe:DATA?), the timestamp can be referenced to the first reading stored in the buffer (absolute format) which is timestamped 0 seconds or can provide the time between each reading (delta format). Timestamp format is set using the TRACe:TSTamp:FORMat command (see “[TRACe command summary](#)”).

**Status word** – A status word is available to provide status information on the measured readings. The 16-bit status word is sent in decimal form and has to be converted by the user to the binary equivalent to determine the state of each bit in the word. For example if the status value is 24, the binary equivalent is 11000. Bits 3 and 4 are set.

The significance of each status bit is explained as follows:

|                        |  |
|------------------------|--|
| Bit 0 – OFLO           | Set to 1 when the measurement(s) was made with the Model 2800 in an A/D overload condition. Otherwise, cleared to 0.         |
| Bit 1 – FILT           | Set to 1 when the measurement(s) was made with the averaging filter enabled. Otherwise, cleared to 0.                        |
| Bit 2 – Ext Ref        | Set to 1 if an external 10MHz reference is connected, but cannot be locked in or is out of spec. Otherwise, cleared to 0.    |
| Bit 3 – Local Osc Lock | Set to 1 if the local oscillator was locked after a frequency change and before a reading is taken. Otherwise, cleared to 0. |
| Bit 4 – Input Prot     | Set to 1 when input protection is enabled. Otherwise, cleared to 0.  |
| Bit 5 – Cable Corr     | Set to 1 when cable correction is enabled. Otherwise, cleared to 0.  |
| Bit 6 – Temp Comp      | Set to 1 when temperature compensation is enabled. Otherwise, cleared to 0.  |



|                          |   |
|--------------------------|---|
| Bit 7 – RRC Filt         | Set to 1 when the root-raised cosine (RRC) filter is enabled with the WCDMA standard selected. Otherwise, cleared to 0. This bit has no effect on other standards |
| Bit 8 – FREQ Range Error | Set to 1 when a $\Delta F$ measurement is out of range. Otherwise, cleared to 0.  |
| Bit 9 – FREQ Power Error | Set to 1 if input power is too low when measuring $\Delta F$ . Otherwise, cleared to 0.   |
| Bits 10 through 15       | Not used.   |

**Example** – Specify data elements for peak power readings:

```
FORM:ELEM:PEAK POWER,FREQ,TIME ' Power, frequency, and timestamp.
```

D form

## Byte order

**FORMat:BORDER <name>**

```
<name> = NORMAL    Normal byte order
        = SWAPPed   Swapped byte order
```

This command is used to control the byte order for the IEEE-754 binary formats. For normal byte order, the data format for each element is sent as follows:

```
Byte 1 Byte 2 Byte 3 Byte 4 (Single precision)
```

For reverse byte order, the data format for each element is sent as follows:

```
Byte 4 Byte 3 Byte 2 Byte 1 (Single precision)
```

The “#0” Header is not affected by this command. The Header is always sent at the beginning of the data string for each measurement conversion.

The ASCII data format can only be sent in the normal byte order. The SWAPPed selection is simply ignored when the ASCII format is selected.

**NOTE** *The SWAPPed byte order must be used when transmitting binary data to any IBM PC compatible computer.*

**Example** – Select normal byte order for binary readings:

```
FORM[:DATA] SRE ' Select binary data format.
FORM:BORD NORM ' Select normal byte order.
```

*E form***Data format for SOURCe2 commands**

|                       |                      |   |
|-----------------------|----------------------|---|
| <b>FORMat:SOURce2</b> | <b>&lt;name&gt;</b>  | <b>Set data format for SOURce2 commands</b> |
|                       | <b>= ASCii</b>       | <b>Decimal format</b>                       |
|                       | <b>= HEXadecimal</b> | <b>Hexadecimal format</b>                   |
|                       | <b>= OCTal</b>       | <b>Octal format</b>                         |
|                       | <b>= BINary</b>      | <b>Binary format</b>                        |

This command is used to set the format for SOURce2 commands. SOURce2 commands are used to set the four digital output lines of the Model 2800. The commands to set the digital outputs are listed in [Table 2-11](#). Details on “[Digital outputs](#)” are provided in [Section 3](#).

**Example** – Configure data string for readings:

```
FORM:DATA ASC           ' Select ASCII data format.
FORM:ELEM:POW PRIM,STAT ' Select primary and status elements.
```

## INSTRument subsystem commands

This subsystem is used to select the measurement mode. The command to select measurement mode is listed in [Table 2-4](#).

Table 2-4

**INSTRument command summary**

| Command           | Description  | Default | Ref                    | SCPI |
|-------------------|--|---------|------------------------|------|
| INSTRument        |  |         |                        |      |
| [[:SElect] <name> | Select transmission standard: AMPS, CDMA1, CDMA2K, GSM, NADC, PHS, or WCDMA. | CDMA1   | <a href="#">A inst</a> | ✓    |
| [[:SElect]?       | Query transmission standard.   |         |                        | ✓    |

### Reference

*A inst*

### **Transmission standards**

|                             |                              |
|-----------------------------|------------------------------|
| INSTRument[[:SElect] <name> | Select transmission standard |
| <name> = AMPS               | AMPS standard                |
| = CDMA1                     | CDMAOne standard             |
| = CDMA2K                    | CDMA 2000 standard           |
| = GSM                       | GSM standard                 |
| = NADC                      | NADC standard                |
| = PHS                       | PHS standard                 |
| = WCDMA                     | WCDMA standard               |

This command is used to select a transmission standard for bandwidth power measurements. After selecting the transmission standard, select a standard type. The commands to select standard types are listed in [“Communication bands”](#).

Each standard retains its frequency settings. For example, after you set the offset frequency for the CDMAOne Cell standard, that settings will be reinstated each time CDMAOne Cell is selected.

**Example** – Select CDMAOne standard and cell band:

```
INST:SEL CDMA1
SENS:CDMA1:TYPE CELL
```

**NOTE** See [“Communication bands”](#) for details on selecting the cell band.

## SENSe subsystem commands

The SENSe subsystem commands provide control of power measurements for a selected standard, such as CDMA1(cellular, PCS), AMPS, CDMA2000, and WCDMA. Commands are also provided to measure peak power. These commands are listed in [Table 2-5](#).

**User's Manual reference** – Detailed procedures for power measurements are provided in [Section 3](#) and [Section 4](#) of the User's Manual.

Table 2-5

**SENSe command summary**

| Command          | Description  | Default                   | Ref    | SCPI |
|------------------|--|---------------------------|--------|------|
| [SENSe[1]]       |  |                           |        |      |
| :FUNCTION <name> | Select function: POWER, PEAK, or DFRequency.   | POWER                     | A sens | ✓    |
| :FUNCTION?       | Query selected function.   |                           |        | ✓    |
| [:PRIMary]       | Primary channel:   |                           | B sens |      |
| :BANDwidth <NRf> | Set primary BW for GSM: 30e3 or 400e3. <sup>4</sup><br>Set primary BW for PHS: 192e3 or 1e6. <sup>5</sup><br>Set primary BW for NADC: 30e3 or 100e3. | 400kHz<br>192kHz<br>30kHz |        |      |
| :BWIDth <NRf>    | Set primary BW for GSM: 30e3 or 400e3. <sup>4</sup><br>Set primary BW for PHS: 192e3 or 1e6. <sup>5</sup><br>Set primary BW for NADC: 30e3 or 100e3. | 400kHz<br>192kHz<br>30kHz |        |      |
| :BANDwidth?      | Query primary bandwidth.   | 1.23MHz <sup>3</sup>      |        |      |
| :BWIDth?         | Query primary bandwidth.   | 1.23MHz <sup>3</sup>      |        |      |
| :AVERage         | Configure averaging:   |                           | C sens |      |
| :COUNT <n>       | Specify averaging count: 1 to 100.   | 6                         |        | ✓    |
| :COUNT?          | Query averaging count.   |                           |        | ✓    |
| :TCONtrol <name> | Select averaging type: REPeat or MOVing.   | MOV                       |        | ✓    |
| :TCONtrol?       | Query averaging type.  |                           |        | ✓    |
| [:STATe] <b>     | Enable or disable filter.  | OFF                       |        | ✓    |
| [:STATe]?        | Query state of filter.   |                           |        | ✓    |
| :ADJacent        | Adjacent channel:  |                           |        |      |
| :BANDwidth?      | Query adjacent bandwidth.  | 30kHz <sup>3</sup>        | B sens |      |
| :BWIDth?         | Query adjacent bandwidth.  | 30kHz <sup>3</sup>        | B sens |      |
| :OFFSet <NRf>    | Set adjacent frequency offset. <sup>1</sup>  | 885.0kHz                  | D sens |      |
| :OFFSet?         | Query adjacent frequency offset.   |                           |        |      |

Table 2-5 (cont.)  
**SENSe command summary**

| Command            | Description  | Default              | Ref    | SCPI |
|--------------------|--|----------------------|--------|------|
| :ALTErnate         | Alternate channel:   |                      |        |      |
| :BANDwidth?        | Query alternate bandwidth.                                       | 30kHz <sup>3</sup>   | B sens |      |
| :BWIDth?           | Query alternate bandwidth.                                       | 30kHz <sup>3</sup>   | B sens |      |
| :OFFSet?           | Query alternate frequency offset.                                | 1.98MHz <sup>3</sup> | D sens |      |
| :FREQuency         | Set center frequency:  |                      | E sens |      |
| :CHANnel <NRf>     | Set as a frequency channel. <sup>2</sup>                         | 1                    |        |      |
| :CHANnel?          | Query channel number.  |                      |        |      |
| :CENTer <NRf>      | Set as a center frequency. <sup>2</sup>                          | 825.03MHz            |        | ✓    |
| :CENTer?           | Query center frequency.  | 825.03MHz            |        | ✓    |
| :POWer             | Mode, attenuator, and concurrent readings:                       |                      |        |      |
| :MODE <name>       | Select measurement mode: FIXed or LIST.                          | FIX                  | F sens |      |
| :MODE?             | Query measurement mode.  |                      |        |      |
| :ATTenuation <NRf> | Set power attenuator: 0 to 50dB.                                 | 26dB                 | G sens |      |
| :ACQuire           | Use optimum setting (auto attenuation).                          |                      |        |      |
| :ATTenuation?      | Query power attenuator.  |                      |        |      |
| :CONCurent <b>     | Enable/disable measurement of alternate and adjacent channels.   | ON                   | H sens |      |
| :CONCurent?        | Query state of concurrent measurements.                          |                      |        |      |
| :ACQuisition       | Set acquisition time:  |                      | I sens |      |
| :TIME <NRf>        | Set acquisition time:<br>1e-6 to 999.999e-3 (sec) <sup>6</sup> . | 0.01                 |        |      |
| :TIME?             | Query acquisition time.  |                      |        |      |
| :CORRection        | Cable compensation:  |                      | J sens |      |
| :CELL <NRf>        | Set high-band correction value: 0 to 99.99 (dB).                 | 0.00dB               |        |      |
| :CELL?             | Query high-band correction value.                                |                      |        |      |
| :PCS <NRf>         | Set low-band correction value: 0 to 99.99 (dB).                  | 0.00dB               |        |      |
| :PCS?              | Query low-band correction value.                                 |                      |        |      |
| [:STATe] <b>       | Enable or disable cable correction.                              | OFF                  |        |      |
| [:STATe]?          | Query state of cable compensation.                               |                      |        |      |
| :INPut             | Input:   |                      | K sens |      |
| :PROTection        | Protection:  |                      |        |      |
| :STATe <b>         | Enable or disable input protection.                              | OFF                  |        |      |
| :STATe?            | Query state of input protection.                                 |                      |        |      |

Table 2-5 (cont.)

**SENSe command summary**

| Command           | Description  | Default | Ref    | SCPI |
|-------------------|--|---------|--------|------|
| :TEMPerature      | Temperature compensation:                                |         | L sens |      |
| :OCOMpensated <b> | Enable/disable temperature compensation.                 | ON      |        |      |
| :ACQuire          | Acquire temperature and use to calculate readings.       |         |        |      |
| :OCOMpensated?    | Query state of temperature compensation.                 |         |        |      |
| :CDMA1            | CDMAOne standard:  |         | M sens |      |
| :TYPE <name>      | Specify type: CELL, PCS, or KPCS.                        | CELL    |        |      |
| :TYPE?            | Query type for CDMAOne.                                  |         |        |      |
| :NADC             | NADC standard:   |         | M sens |      |
| :TYPE <name>      | Specify type: CELL or PCS.                               | CELL    |        |      |
| :TYPE?            | Query type for NADC.                                     |         |        |      |
| :GSM              | GSM standard:  |         | M sens |      |
| :STAir? <NRf>     | Query sequential time slots.<br><NRf> = number of slots. |         | N sens |      |
| :TMASK            |  |         | N sens |      |
| :STATistics?      | Query time mask statistics of a GSM pulse.               |         |        |      |
| :TMASK?           | Query power vs. time of a GSM pulse.                     |         | N sens |      |
| :TYPE <name>      | Specify type: CELL, DCS, PCS, or US850.                  | CELL    |        |      |
| :TYPE?            | Query type for GSM.                                      |         |        |      |
| :CDMA2K           | CDMA 2000 standard:                                      |         | M sens |      |
| :TYPE <name>      | Specify type: BC0, BC1, BC4, BC6, BC8, or BC9.           | BC0     |        |      |
| :TYPE?            | Query type for CDMA 2000.                                |         |        |      |
| :SRATe <name>     | Set spreading rate: SR1 or SR3.                          | SR1     |        |      |
| :SRATe?           | Query spreading rate.                                    |         |        |      |
| :WCDMA            | WCDMA standard:  |         | M sens |      |
| :TYPE <name>      | Specify type: OB1, OB2, or OB3.                          | OB1     |        |      |
| :TYPE?            | Query type for WCDMA.                                    |         |        |      |
| :RRC <b>          | Enable or disable Root Raised Cosine filter.             | ON      |        |      |

Table 2-5 (cont.)

**SENSe command summary**

| Command             | Description  | Default | Ref                    | SCPI |
|---------------------|--|---------|------------------------|------|
| :LIST               | Sweep list:  |         | <a href="#">O sens</a> |      |
| :CONTRol <name>     | Select sweep list type: FREQuency or ATTenuation. <sup>7</sup> | FREQ    |                        |      |
| :FREQuency <list>   | Create frequency list: NRf,[NRf,...NRf].                       |         |                        | ✓    |
| :POINts?            | Query number of sweep points in list.                          |         |                        | ✓    |
| :APPend <list>      | Append frequency list: NRf,[NRf,...NRf].                       |         |                        |      |
| :FREQuency?         | Query frequency list.  |         |                        | ✓    |
| :ATTenuation <list> | Create attenuator list: NRf,[NRf,...NRf].                      |         |                        |      |
| :POINts?            | Query number of sweep points in list.                          |         |                        |      |
| :APPend <list>      | Append attenuator list: NRf,[NRf,...NRf].                      |         |                        |      |
| :ATTenuation?       | Query attenuator list.   |         |                        |      |

<sup>1</sup> The setting range for the adjacent channel offset depends on the selected standard. See the chart for “Adjacent Channel” in [Appendix A – Specifications](#).

<sup>2</sup> The setting ranges for center frequency and channel depend on the selected standard. See the chart for “Frequency Settings” in [Appendix A – Specifications](#).

<sup>3</sup> The listed default is the setting for the CDMAOne Cell standard.

<sup>4</sup> In order to set the GSM primary BW to 30kHz, concurrent measurements must be enabled (SENS:POW:CONC ON).

<sup>5</sup> In order to set the PHS primary BW to 1MHz, concurrent measurements must be disabled (SENS:POW:CONC OFF).

<sup>6</sup> 1e-6 to 999.999e-3 for GSM and PHS. 1e-5 to 999.99e-3 for CDMAOne, CDMA2000, and NDAC.

<sup>7</sup> FREQuency and ATTenuation can be used together for concurrent sweep. Parameters must be separated by a comma, for example: LIST:CONT FREQ, ATT

**Reference**

*A sens*

**Function**

|   |          |  |
|---|----------|--|
| <b>[SENSe[1]]:FUNction &lt;name&gt;</b> |          |  |
| <b>&lt;name&gt;</b>                     | <b>=</b> | <b>POWER</b> <b>Power function</b>                     |
|   | <b>=</b> | <b>PEAK</b> <b>Peak function</b>                       |
|   | <b>=</b> | <b>DFrequency</b> <b><math>\Delta</math>F function</b> |

This command is used to select the measurement function. There are three basic measurement functions for the Model 2800: power, peak, and  $\Delta$ F.

Power measurements are the bandwidth power measurements for the selected transmission standard. These include power measurements for primary, adjacent, and alternate channels.

The peak power function measures the five highest peak amplitudes within a specified frequency spectrum. The frequency of each peak is also measured.

Before selecting the  $\Delta$ F function, make sure the instrument is displaying a valid power reading for the desired standard. The instrument must not be in an overrange condition. The minimum amplitude that can be detected by the Model 2800 for a  $\Delta$ F measurement is -35dBm.

Example:

SENS : FUNC POW               ' Select power function.

*B sens*

**Bandwidths**

|   |                                   |
|---|-----------------------------------|
| <b>[SENSe[1]][:PRIMary]:BANDwidth &lt;NRf&gt;</b> | <b>Set primary channel BW</b>     |
| <b>[SENSe[1]][:PRIMary]:BANDwidth?</b>            | <b>Query primary channel BW</b>   |
| <b>[SENSe[1]]:ADJacent:~BANDwidth?</b>            | <b>Query adjacent channel BW</b>  |
| <b>[SENSe[1]]:ALTErnate:~BANDwidth?</b>           | <b>Query alternate channel BW</b> |
| <b>[SENSe[1]][:PRIMary]:B~IDth &lt;NRf&gt;</b>    | <b>Set primary channel BW</b>     |
| <b>[SENSe[1]][:PRIMary]:B~IDth?</b>               | <b>Query primary channel BW</b>   |
| <b>[SENSe[1]]:ADJacent:~B~IDth?</b>               | <b>Query adjacent channel BW</b>  |
| <b>[SENSe[1]]:ALTErnate:~B~IDth?</b>              | <b>Query alternate channel BW</b> |

**GSM primary bandwidth:** The primary channel bandwidth for GSM can be set to either 30kHz or 400kHz. The default BW setting is 400kHz. The 30kHz BW can only be selected if concurrent measurements are enabled (SENS:POW:CONC ON). Enabling concurrent measurements enables the ability to measure adjacent and alternate channels.



With concurrent measurements disabled (SENS:POW:CONC OFF), 400kHz is the only valid primary channel BW setting for GSM. Trying to select the 30kHz setting will result in error +833 (Meas BW fixed or invalid).

**PHS primary bandwidth:** The primary channel bandwidth for PHS can be set to either 192kHz or 1MHz. The default BW setting is 192kHz. The 1MHz BW can only be selected if concurrent measurements are disabled (SENS:POW:CONC OFF).

With concurrent measurements enabled (SENS:POW:CONC ON), 192kHz is the only valid primary channel BW setting for PHS. Trying to select the 1MHz setting will result in error +833 (Meas BW fixed or invalid).

**NOTES** See “[Concurrent measurements](#)” for details on enabling or disabling concurrent measurements.

Error +833 “Meas BW fixed or invalid” will result whenever you try to set a bandwidth that is fixed or not valid, as described above.

**NADC primary bandwidth:** The primary channel bandwidth for NADC can be set to either 30kHz or 100kHz. The default BW setting is 30kHz.

All other bandwidth settings are fixed and cannot be changed by the user. The above query commands are used to query the bandwidths for the primary, adjacent, and alternate channels of the currently selected transmission standard.

A :BANDwidth? command and its corresponding :BWIDth? command performs the same query operation. Use either one. For example, use :PRIMary:BANDwidth? or :PRIMary:BWIDth? to query primary channel bandwidth.

**Example 1** – Query bandwidths for CDMAOne Cell standard:

|                      |                                      |
|----------------------|--------------------------------------|
| INST:SEL CDMA1       | ' Select the CDMAOne standard.       |
| SENS:CDMA1:TYPE CELL | ' Select the Cell band.              |
| SENS:FUNC POW        | ' Select bandwidth power function.   |
| SENS:PRIM:BAND?      | ' Query primary bandwidth (1.23MHz). |
| SENS:ADJ:BAND?       | ' Query adjacent bandwidth (30kHz).  |
| SENS:ALT:BAND?       | ' Query alternate bandwidth (30kHz). |

**Example 2** – Set primary BW for GSM to 30kHz:

|                     |                                    |
|---------------------|------------------------------------|
| INST:SEL GSM        | ' Select the GSM standard.         |
| SENS:GSM:TYPE CELL  | ' Select the Cell band.            |
| SENS:FUNC POW       | ' Select bandwidth power function. |
| SENS:POW:CONC ON    | ' Select concurrent measurements.  |
| SENS:PRIM:BAND 30e3 | ' Set primary bandwidth to 30kHz.  |
| SENS:PRIM:BAND?     | ' Query primary bandwidth (30kHz). |

*C sens*

## Averaging

Averaging can be used to improve repeatability for CDMA measurements. The “Averaging” feature is explained on page 2-5 of User’s Manual.

**[SENSe[1]]:AVERAge:COUNT <n>**                      **Specify averaging count**  
 <n> = 1 to 100                      **Averaging count**

This command sets the averaging count. In general, reading repeatability is improved by increasing the count value. However, increasing the count value also slows down the reading rate. Typically, averaging count is set to a value that is a compromise between repeatability and speed.

**[SENSe[1]]:AVERAge:TCONtrol <name>**                      **Set averaging type**  
 <name> = REPeat                      **Repeat filter type**  
           = MOVing                      **Moving filter type**

This command sets the averaging type. For the repeat type, a stack is filled with a specified number of reading conversions (as set by :COUNT) and then averaged to yield a single reading. The stack is emptied, again filled with a conversion, and another reading is averaged.

For the moving filter, the stack is not emptied after the first reading is averaged. Instead, a new measurement conversion is placed in the stack and the oldest conversion is removed. The stack is then averaged to yield a reading. This process of adding a new conversion and removing the oldest conversion continues for each measurement.

**[SENSe[1]]:AVERAge[:STATe] <b>**                      **Control averaging**  
 <b> = ON or 1                      **Enable averaging**  
       = OFF or 0                      **Disable averaging**

This command is used to enable or disable averaging. When enabled, the averaging process is performed as configured by the :COUNT and :TCONfigure commands.

Keep in mind that using averaging with a count of one is effectively the same as disabling averaging.

**Example** – Configure averaging:

```
SENS:AVER:COUN 10      ' Set averaging count to 10.
SENS:AVER:TCON MOV     ' Select moving type.
SENS:AVER:STAT ON     ' Enable averaging.
```

*D sens*

## Offset frequencies

|                                  |                                |
|----------------------------------|--------------------------------|
| [SENSe[1]]:ADJacent:OFFSet <NRf> | Set adjacent channel offset    |
| [SENSe[1]]:ADJacent:OFFSet?      | Query adjacent channel offset  |
| [SENSe[1]]:ALTErnate:OFFSet?     | Query alternate channel offset |

<NRf> = The setting range for the adjacent channel offset depends on the selected standard. See the chart for “Adjacent Channel” in [Appendix A – Specifications](#).

The SENS:ADJ:OFF <NRf> command can be used change the adjacent channel offset. The alternate channel offset is fixed and cannot be changed. The alternate channel offset frequency depends on the selected transmission standard. The fixed alternate channel offsets settings are listed in Table 4-1 on page 4-3 in the User’s Manual.

**Examples** — Set frequency offsets for CDMAOne cell:

```
INST:SEL CDMA1        ' Select the CDMAOne standard.
SENS:CDMA1:TYPE CELL  ' Select Cell band.
SENS:ADJ:OFFS 900e3    ' Set adjacent offset 900kHz.
SENS:ALT:OFFS?        ' Query alternate offset (1.98MHz).
```

*E sens*

## Center frequency

|                                    |                          |
|------------------------------------|--------------------------|
| [SENSe[1]]:FREquency:CHANnel <NRf> | Set as frequency channel |
| [SENSe[1]]:FREquency:CENTer <NRf>  | Set as center frequency  |

<NRf> = The setting ranges for center frequency and channel depend on the selected standard. See the chart for “Frequency Settings” in [Appendix A – Specifications](#).

The center frequency is fundamental to all Model 2800 measurements. This is the initial reference point for measurements. Measurements are either performed for a bandwidth centered at this frequency (primary, peak, and  $\Delta F$  measurements) or an offset is specified to move the frequency pointer (adjacent and alternate measurements).

The center frequency can be specified as a channel or it can be set as a frequency. For all standards except WCDMA, the center frequency can be set anywhere within the range of band classes (BC0, BC1, BC4, BC6, BC8, BC9). The channel number will be dashed out on the front panel for center frequency not equivalent to a channel number.

#### Examples:

```
SENS:FREQ:CHAN 1           ' Set channel 1 as center frequency.
SENS:FREQ:CENT 825.03e6    ' Set center frequency to 825.03MHz.
```

*F sens*

## Measurement mode

```
[SENSe[1]]:POWer:MODE <name>      Measurement mode
<name> = FIXed      Fixed power measurement mode
        = LIST       Sweep list mode
```

With the FIXed mode selected, the instrument is in the normal measurement mode. That is, each measurement will reflect the current measurement configuration of the instrument. FIXed is the default setting.

With the sweep LIST mode selected, the instrument will perform a measurement sweep using a listed set of center frequency settings or attenuator settings. Details on “List mode” operation are provided on page 3-20 of the User’s Manual.

#### Example:

```
SENS:POW:MODE LIST        ' Enable (select) list mode.
SENS:POW:MODE FIX         ' Disable list mode (select fixed mode).
```

*G sens*

## Power attenuation

```
[SENSe[1]]:POWer:ATTenuation <NRf>      Manual attenuation
[SENSe[1]]:POWer:ATTenuation:ACQuire     Auto attenuation
<NRf> = 0 to 50      dB
```

In order to perform accurate power measurements, the input signal needs to be properly attenuated by the Model 2800. The attenuator setting depends on the set center frequency and the input power level.

Nominal attenuator settings for measuring power levels can be defined by the following formulas:

For the Cellular Bands (824 – 849MHz, 880 – 915MHz):  
**Attenuator Value (dB) = Expected Input Power + 14**

DCS,PCS Bands (1710 – 1785MHz, 1850 – 1980MHz)  
**Attenuator Value (dB) = Expected Input Power + 30**

When measuring primary power measurements only, one attenuator setting can be used over a maximum of a 30dB range.

The range over which power measurements can be made for a given attenuator setting is as follows:

For the Cellular Bands (824 – 849MHz, 880 – 915MHz):

**(Attenuation setting – 70) <Input Power (dBm) <(Attenuation – 14) [dBm]**

DCS,PCS Bands (1710 – 1785MHz, 1850 – 1980MHz):

**(Attenuation setting – 70) <Input Power (dBm) <(Attenuation –24) [dBm]**

**NOTE** See “Attenuation” in the Model 2800 User’s Manual for details.

**Auto attenuation** – For CDMA measurements, auto attenuation (SENS:POW:ATT:ACQ) can be used if a high quality CDMA source with very low intermodulation products is being used. If such a source is not being used, DO NOT use auto attenuation. The “Auto attenuation (CDMA only)” process is explained on page 2-16 of the User’s Manual.

**NOTE** *Attempting to use auto attenuation for the NADC, GSM, PHS, or AMPS standard will result in error +832 (Invalid with standard selected).*  
*Attempting to use auto attenuation with external trigger selected will result in error +834 (Invalid with EXTERNAL trigger).*

Example: Assumes a CDMA standard is already selected.

SENS:POWer:ATT 20            ' Set attenuator to 20 dB.

H sens

## Concurrent measurements

[SENSe[1]]:POWer:CONCurrenT <b>

<b> = ON or 1      Enable concurrent measurements  
 = OFF or 0      Disable concurrent measurements

By default, the Model 2800 measures the primary, adjacent, and alternate channels concurrently. However, when only the primary channel needs to be measured, the measurement of adjacent and alternate channels can be disabled. Disabling concurrent measurements speeds up the measurement process considerably.

**NOTE**      *With concurrent measurements disabled, the GSM primary channel bandwidth cannot be set to 30kHz. If the GSM primary bandwidth was previously set to 30kHz, it will reset to 400kHz when concurrent measurements are disabled. For details on setting GSM primary bandwidth, see “Bandwidths”.*

### Example:

SENS:POWer:CONC ON      ' Enable concurrent measurements.

I sens

## Acquisition time

[SENSe[1]]:ACQuisition:TIME <NRf>

<NRf> = 1e-6 to 999.999e-3      Acquisition time (in seconds)  
 for GSM and PHS

<NRf> = 1e-5 to 999.99e-3      Acquisition time (in seconds)  
 for CDMAOne, CDMA2000, NDAC

In general, acquisition time is the amplitude sampling time for a power measurement conversion. Reading repeatability is increased by increasing the acquisition time.

For NADC, GSM, and PHS power measurements, the acquisition time setting is critical and is limited by the length of the slot and frame. For a burst power measurement, the acquisition time must be set such that only the power within the slot is measured. For average burst power, the acquisition time must encompass an entire frame. Details on NADC, GSM, and PHS measurements are provided in Section 4, “Mobile Transmitter Measurements” of the User’s Manual.

**Example:**

```

INST:SEL NADC           ' Select the NADC standard.
SENS:NADC:TYPE CELL    ' Select Cell band.
SENS:ACQ:TIME 20e-3    ' Set acquisition time to 20ms.

```

*J sens***Cable compensation**

```

[SENSe[1]]:CORRection:CELL <NRf>    Low-band correction (<1GHz)
[SENSe[1]]:CORRection:PCS <NRf>     High-band correction (>1GHz)
<NRf> = 0 to 99.99    Correction value (dB)

```

Power loss in an RF input cable could be large enough to affect the integrity of your measurements. By using cable compensation, the power loss is factored into the calculation for the power measurement. This cancels the effect of the cable power loss. When using the level trigger source and when cable correction is enabled, the level trigger level will be detected on the cable corrected measured value.

In general, you must first determine the power loss in the signal path between the DUT and the Model 2800, and then enter the power loss value into the Model 2800. The procedure to perform “Cable compensation” is provided in the User’s Manual on page 3-5.

*K sens***Input protection**

```

[SENSe[1]]:INPut:PROTection:STATe <b>
<b> = ON or 1    Enable input protection
     = OFF or 0   Disable input protection

```

The Model 2800 has Input Protection that can be controlled by the user. When enabled, input protection serves two purposes: Overload protection and local oscillator emission attenuation. See page 2-7 of the User’s Manual for details on “Input protection”.

**NOTE** *Enabling input protection slows down measurement speed. It adds 500 $\mu$ s to the measurement time. Input protection must be disabled for the GSM standard since it measures high speed pulses (579 $\mu$ s),*

**Example:**

```

SENS:INP:PROT:STAT OFF ' Disable input protection.

```

L sens

## Temperature compensation

[SENSe[1]]:TEMPerature:OCOMpensated <b>

<b> = ON or 1      Enable temperature compensation  
 = OFF or 0      Disable temperature compensation

Temperature compensation compensates for ambient temperature drift which adversely affects measured readings. Details on “Temperature compensation” are provided on page 2-7 of the User’s Manual.

**NOTE**    *Disabling temperature compensation increases measurement speed. However, it should only be disabled for short periods of time and then re-enabled.*

### Examples:

SENS:TEMP:OCOM OFF    ' Disable temperature compensation.  
 SENS:TEMP:OCOM ON    ' Enable temperature compensation.

[SENSe[1]]:TEMPerature:OCOMpensated:ACQuire

This command forces (acquires) a temperature measurement while temperature compensation is disabled. The latest temperature reading is then used in the algorithm to calculate subsequent readings.

M sens

## Communication bands

The following commands are used to select the band for the transmission standards.

[SENSe[1]]:CDMA1:TYPE <name>

<name> = CELL      Select Cell band  
 = PCS      Select PCS band  
 = KPCS      Select KPCS band

[SENSe[1]]:NADC:TYPE <name>

<name> = CELL      Select Cell band  
 = PCS      Select PCS band

[SENSe[1]]:GSM:TYPE <name>

<name> = CELL      Select Cell band  
 = DCS      Select DCS band  
 = PCS      Select PCS band  
 = US850      Select US850 band



```
[SENSe[1]]:CDMA2K:TYPE <name>
[SENSe[1]]:CDMA2K:SRATe <name>
TYPE <name> = BC0      Select band class 0
              = BC1      Select band class 1
              = BC4      Select band class 4
              = BC6      Select band class 6
              = BC8      Select band class 8
              = BC9      Select band class 9
SRATe <name> = SR1      Select 1x spreading rate
              = SR3      Select 3x spreading rate
```

```
[SENSe[1]]:WCDMA:TYPE <name>
<name> = OB1      Select OB1 standard type
        = OB2      Select OB2 standard type
        = OB3      Select OB3 standard type
```

**Example:**

```
INST:SEL CDMA1           ' Select CDMAOne standard.
SENS:CDMA1:TYPE CELL    ' Select Cell band.
```

*N sens*

## Sweep list

The following commands are used to configure a sweep list for center frequency settings or attenuator settings. When the sweep is run, measurements using each sweep point will be performed. Details on “List mode” operation, including remote programming examples, are provided on page 3-20 of the User’s Manual.

**Select sweep list type:**

```
[SENSe[1]]:LIST:CONTROL <name>
<name> = FREQuency      Select frequency list type
        = ATTenuation    Select attenuator list type
        = FREQ, ATT      Concurrent frequency, attenuator list type
```

Use this command to select the type of list to create. With the **FREQ** list selected, sweep point values will be interpreted as center frequency settings. With the **ATT** list selected, sweep points values will be interpreted as attenuator settings. Examples:

```
SENS:LIST:CONT FREQ      ' Select frequency list type.
SENS:LIST:CONT ATT      ' Select attenuator list type.
SENS:LIST:CONT FREQ,ATT  ' Select concurrent list type.
```

**Create sweep list:**

|   |                                 |
|---|---------------------------------|
| <b>[SENSe[1]]:LIST:FREQuency &lt;list&gt;</b>   | <b>Create frequency list</b>    |
| <b>[SENSe[1]]:LIST:ATTenuation &lt;list&gt;</b> | <b>Create attenuator list</b>   |
| <b>&lt;list&gt; = NRf[NRf,NRf,...NRf]</b>       | <b>Specify list of settings</b> |

Use one of these commands to specify a frequency list or an attenuator list. For a frequency list, the sweep points are center frequency settings. For an attenuator list, the sweep points are attenuator settings.

Each sweep point in the list must be separated by a comma. The order of sweep points in the list determines the order of the sweep. Examples:

```
SENS:LIST:FREQ 825.03e6,825.06e6,825.09e6 ' Frequency list:
' 825.03MHz,
' 825.06MHz,
' 825.09MHz.
```

```
SENS:LIST:ATT 20,22,24,26 ' Attenuator list: 20dB, 22dB, 24dB,
' 26dB.
```

**Append sweep list:**

|  |                                 |
|--|---------------------------------|
| <b>[SENSe[1]]:LIST:FREQuency:APPend &lt;list&gt;</b>   | <b>Append frequency list</b>    |
| <b>[SENSe[1]]:LIST:ATTenuation:APPend &lt;list&gt;</b> | <b>Append attenuator list</b>   |
| <b>&lt;list&gt; = NRf[NRf,NRf,...NRf]</b>              | <b>Specify list of settings</b> |

Use one of these commands to append (add) sweep points to the list. Each appended sweep point is added to the end of the list. Examples:

```
SENS:LIST:FREQ:APP 825.12e6,825.15e6 ' Add two sweep points
' to frequency list:
' 825.12MHz,
' 825.15MHz.
```

```
SENS:LIST:ATT 28,30 ' Add two sweep points to attenuator
' list: 28dB, 30dB.
```

**Query number of sweep points in list:****[SENSe[1]]:LIST:FREQuency:POINts?****[SENSe[1]]:LIST:ATTenuation:POINts?**

Use one of these query commands to determine the number of sweep points in the list.

The trigger count determines the number of measurements to be performed. Therefore for a sweep, the trigger count is usually set to the number of sweep points in the list. For example, if the sweep list consists of five points, the trigger count should be set to five. See “[TRIGGer commands](#)” for details on the TRIGGer:COUNT command.

*O sens****GSM staircase and time analysis***

The following commands are used to perform additional analysis for GSM signals only, including GSM staircase and GSM power vs. time, and time mask statistics.

**NOTE** See “*GSM staircase and time analysis*”, page 3-26 of the *Model 2800 User’s Manual* for complete details.

**GSM staircase mode:****[SENSe[1]]:GSM:STAir? <NRf>** Query sequential GSM slots

When initiated with GSM:STAir? command, the Model 2800 will wait until the threshold of the user set level is crossed. After the threshold has been crossed, the Model 2800 will wait the desired delay time set by the user before taking measurements on the RF input. The measurement of the second slot will begin at exactly one GSM time slot after the delay. This process is repeated for the number of <NRf> slot steps that are programmed.

**GSM power vs. time:****[SENSe[1]]:GSM:TMASk?** Query power vs. time of a GSM pulse.

This query returns the power versus time function of a GSM pulse and requires that a synchronizing pulse be sent. The function will acquire 620 samples that are spaced 0.977 $\mu$ s apart. The function will return some noise before and after the pulse so that the user will be able to verify that the pulse complies with the GSM standard.

**GSM time mask statistics:****[SENSe[1]]:GSM:TMASk:STATistics? Query time mask statistics**

This query will return the 10%-90% rise time, 90%-10% fall time, maximum and minimum values of readings taken during the delay time, and maximum and minimum values of readings taken during the acquisition time. This function also requires that a synchronizing pulse be sent.

**SOURce2 subsystem commands**

The SOURce2 subsystem commands (Table 2-6) are used to set the logic level for the four output lines of the Digital I/O. These output lines are intended for the general purpose to control low current devices.

**NOTE** Details on using the “Digital I/O” are provided in Section 3.

Table 2-6

**SOURce2 command summary**

| Command                | Description  | Default |
|------------------------|--|---------|
| SOURce2                |  |         |
| :TTL                   | Digital I/O port:                                      |         |
| :INPut?                | Query digital input state: Hi (1) or Lo (0).           |         |
| :LEVel] <NDN> or <NRf> | Specify bit pattern for the four digital output lines. | 15      |
| [:LEVel]?              | Query output bit pattern.                              |         |

## Digital input

**SOURce2:TTL:INPut?**

**Query digital input state**

Pin 8 of the digital I/O port is the input line. When TTL high is present on this line, :INPut? returns a “1”. When TTL low is on this line, this query returns a “0”. See “[Digital input](#)” in [Section 3](#) for details on using the digital input port.

## Digital outputs

**SOURce2:TTL[:LEVel] <NRf>**

**Set digital output pattern**

|       |   |         |                    |                     |
|-------|---|---------|--------------------|---------------------|
| <NRf> | = | 0 to 15 | Decimal format     |                     |
| <NDN> | = | #Bxxxx  | Binary format      | xxxx = 0000 to 1111 |
|       | = | #Hx     | Hexadecimal format | x = 0 to F          |
|       | = | #Qx     | Octal format       | x = 0 to 17         |

The Digital I/O of the Model 2800 has four output lines that can be used to control external devices, such as logic circuits or low current relays. The output lines are powered by an internal +5V, 100mA supply and can be used to source or sink current. Details on using the “[Digital I/O](#)” are provided in [Section 3](#).

An output line can be set high (H) or low (L). When using the decimal format to set the bit pattern, use [Table 2-7](#) to determine the decimal value for the parameter. For example, to set output 4 high and the outputs 1, 2, and 3 low, use parameter value “8” as shown in the chart.

When using one of the non-decimal formats, convert the decimal value for the bit pattern to the equivalent binary, octal, or hexadecimal value. Be sure to use the appropriate header (#B, #Q, or #H) with the non-decimal number. For example, for decimal 8, the binary equivalent for the parameter is #B1000.

When using the query command to read the set bit pattern, the format for the returned value depends on the presently set data format for SOURce2 commands. The FORMat:SOURce2 command is used to select the data format (see “[Data format for SOURce2 commands](#)”).

*Table 2-7*  
**Output patterns for NRf parameter**

| Out 4 | Out 3 | Out 2 | Out 1 | NRf |
|-------|-------|-------|-------|-----|
| L     | L     | L     | L     | 0   |
| L     | L     | L     | H     | 1   |
| L     | L     | H     | L     | 2   |
| L     | L     | H     | H     | 3   |
| L     | H     | L     | L     | 4   |
| L     | H     | L     | H     | 5   |
| L     | H     | H     | L     | 6   |
| L     | H     | H     | H     | 7   |
| H     | L     | L     | L     | 8   |
| H     | L     | L     | H     | 9   |
| H     | L     | H     | L     | 10  |
| H     | L     | H     | H     | 11  |
| H     | H     | L     | L     | 12  |
| H     | H     | L     | H     | 13  |
| H     | H     | H     | L     | 14  |
| H     | H     | H     | H     | 15  |

**Example** – Set outputs 4 and 2 high and outputs 3 and 1 low:

SOUR2:TTL:LEV #B1010 ' Binary format.

## STATus subsystem commands

The STATus subsystem commands are used to control and read the status registers of the Model 2800. These commands are listed in [Table 2-8](#).

**NOTE** Details on “[Status structure](#)” are provided in [Section 3](#).

Table 2-8

**STATus command summary**

| Command                | Description                                     | Default  | Ref                    | SCPI |
|------------------------|---|----------|------------------------|------|
| STATus                 |   | (Note 1) |                        | ✓    |
| :MEASurement           | Control measurement event registers:            |          |                        |      |
| [:EVENT]?              | Read the event register.                        | (Note 2) | <a href="#">A stat</a> |      |
| :ENABLE <NDN> or <NRf> | Program the enable register.                    | (Note 3) | <a href="#">B stat</a> |      |
| :ENABLE?               | Read the enable register.                       |          |                        |      |
| :CONDition?            | Read the condition register.                    |          | <a href="#">C stat</a> |      |
| :OPERation             | Control operation status registers:             |          |                        | ✓    |
| [:EVENT]?              | Read the event register.                        | (Note 2) | <a href="#">A stat</a> | ✓    |
| :ENABLE <NDN> or <NRf> | Program the enable register.                    | (Note 3) | <a href="#">B stat</a> | ✓    |
| :ENABLE?               | Read the enable register.                       |          |                        | ✓    |
| :CONDition?            | Read the condition register.                    |          | <a href="#">C stat</a> | ✓    |
| :QUESTionable          | Control questionable status registers:          |          |                        | ✓    |
| [:EVENT]?              | Read the event register.                        | (Note 2) | <a href="#">A stat</a> | ✓    |
| :ENABLE <NDN> or <NRf> | Program the enable register.                    | (Note 3) | <a href="#">B stat</a> | ✓    |
| :ENABLE?               | Read the enable register.                       |          |                        | ✓    |
| :CONDition?            | Read the condition register.                    |          | <a href="#">C stat</a> | ✓    |
| :PRESet                | Return status registers to default states.      |          |                        | ✓    |
| :QUEue                 | Access error queue (error and status messages): |          | <a href="#">D stat</a> | ✓    |
| [:NEXT]?               | Read the oldest message.                        | (Note 4) |                        | ✓    |
| :ENABLE <list>         | Specify messages to be placed in queue.         | (Note 5) |                        | ✓    |
| :ENABLE?               | Read the enabled messages.                      |          |                        | ✓    |
| :DISable <list>        | Specify messages not to be placed in queue.     | (Note 5) |                        |      |
| :DISable?              | Read the disabled messages.                     |          |                        |      |
| :CLEar                 | Clears all messages from Error Queue.           |          |                        |      |

1. Commands in this subsystem are not affected by \*RST and SYSTem:PRESet. The effects of cycling power, \*CLS, and STATus:PRESet are explained by the following notes.
3. Event Registers: Power-up and \*CLS clears all bits of the registers. STATus:PRESet has no effect.
4. Enable Registers: Power-up and STATus:PRESet clears all bits of the registers. \*CLS has no effect.
5. Error Queue: Power-up and \*CLS clears the Error Queue. STATus:PRESet has no effect.
6. Enable/Disable Error Queue Messages: Power-up clears list of messages.\*CLS and STATus:PRESet has no effect.

**Parameters:**

|                    |          |                   |                           |                        |
|--------------------|----------|-------------------|---------------------------|------------------------|
| <b>&lt;NRf&gt;</b> | <b>=</b> | <b>0 to 32767</b> | <b>Decimal format</b>     |                        |
| <b>&lt;NDN&gt;</b> | <b>=</b> | <b>#Bxx...x</b>   | <b>Binary format</b>      | <b>each x = 0 or 1</b> |
|                    | <b>=</b> | <b>#Hx</b>        | <b>Hexadecimal format</b> | <b>x = 0 to 7FFF</b>   |
|                    | <b>=</b> | <b>#Qx</b>        | <b>Octal format</b>       | <b>x = 0 to 77777</b>  |

The parameter values to set the individual bits of the enable registers can be sent using the decimal format (NRf) or a non-decimal format (NDN). Details on “[Programming enable registers](#)” are provided in [Section 3](#).

When reading a status register, the presently set data format determines the type of value (decimal or non-decimal) that is returned. The “[FORMat:SREGister <name>](#)” command is used to set the data format. Details on “[Reading registers](#)” are provided in [Section 3](#).

**Reference***A stat***Register sets**

Details on the “[Status register sets](#)” are provided in [Section 3](#).

*Read event registers:*

|                                     |   |
|-------------------------------------|---|
| <b>STATus:MEASurement[:EVENT]?</b>  | <b>Read measurement event register</b>  |
| <b>STATus:OPERation[:EVENT]?</b>    | <b>Read operation event register</b>    |
| <b>STATus:QUESTionable[:EVENT]?</b> | <b>Read questionable event register</b> |

Use these commands to read the measurement, operation, and questionable event registers.

*B stat**Program enable registers:*

|  |                                     |
|--|-------------------------------------|
| <b>STATus:MEASurement:ENABLE &lt;NDN&gt; or &lt;NRf&gt;</b>  | <b>Measurement enable register</b>  |
| <b>STATus:OPERation:ENABLE &lt;NDN&gt; or &lt;NRf&gt;</b>    | <b>Operation enable register</b>    |
| <b>STATus:QUESTionable:ENABLE &lt;NDN&gt; or &lt;NRf&gt;</b> | <b>Questionable enable register</b> |

Use these commands to program the measurement, operation, and questionable enable registers. Use the query form for these commands (:ENABLE?) are used to read the programmed values.

**Example** – Enable bit B9 (buffer full) of the measurement event register:

```
STAT:MEAS:ENAB 512
```



*C stat**Read condition registers:*

|                                       |   |
|---------------------------------------|---|
| <b>STATus:MEASurement:CONDition?</b>  | <b>Read measurement condition register</b>  |
| <b>STATus:OPERation:CONDition?</b>    | <b>Read operation condition register</b>    |
| <b>STATus:QUESTionable:CONDition?</b> | <b>Read questionable condition register</b> |

Use these commands to program the measurement, operation, and questionable condition registers.

*D stat***Error queue**

The Model 2800 uses an error queue for error and status messages. When an error or status event occurs, it is placed in the queue. Details on “Queues” are provided in [Section 3](#).

|                             |                         |
|-----------------------------|-------------------------|
| <b>STATus:QUEue[:NEXT]?</b> | <b>Read error queue</b> |
|-----------------------------|-------------------------|

Use this command to read the error queue. It returns and then clears the oldest message in queue. Error and status messages are listed in [Appendix B](#).

|  |                                |
|--|--------------------------------|
| <b>STATus:QUEue:ENABle &lt;list&gt;</b>  | <b>Enable listed messages</b>  |
| <b>STATus:QUEue:DISABle &lt;list&gt;</b> | <b>Disable listed messages</b> |
| <b>&lt;list&gt; = List of codes</b>      |                                |

The following examples show various forms for using the <list> parameter.

|                              |   |
|------------------------------|---|
| <b>&lt;list&gt; = (-110)</b> | Single message                                      |
| <b>= (-110:-222)</b>         | Range of messages (-110 through -222)               |
| <b>= (-110:-222, -220)</b>   | Range entry and single entry (separated by a comma) |

When you enable messages, messages not specified in the list are disabled and will not be placed in the queue. When you disable messages, each listed message is removed from the enabled list.

**Example** – Specify list of enable messages:

```
STAT:QUE:ENAB (-100:-300) ' Enable messages -100 through -300.
```

|                           |                          |
|---------------------------|--------------------------|
| <b>STATus:QUEue:CLEAr</b> | <b>Clear error queue</b> |
|---------------------------|--------------------------|

Use this command to clear all messages from the error queue.

## SYSTem subsystem commands

The SYSTem subsystem commands are used to control miscellaneous instrument operations of the Model 2800. These commands are listed in [Table 2-9](#).

Table 2-9

**SYSTem command summary**

| Command         | Description  | Default  | Ref    | SCPI |
|-----------------|--|----------|--------|------|
| SYSTem          |  |          |        |      |
| :PRESet         | Return to :SYST:PRES defaults.                                 |          | A syst | ✓    |
| :POSetup <name> | Select power-on setup: (RST, PRESet, SAV0, SAV1, or SAV2).     |          | B syst |      |
| :POSetup?       | Query power-on setup.  |          |        |      |
| :KEY <n>        | Simulate key-press (1 to 31: See <a href="#">Figure 2-3</a> ). |          | C syst | ✓    |
| :KEY?           | Query the last “pressed” key.                                  |          |        | ✓    |
| :ERRor?         | Query (read) Error Queue.                                      | (Note 1) | D syst | ✓    |
| [:NEXT]?        | Return and clear the latest error (code and message).          |          |        |      |
| :ALL?           | Return and clear all errors (code and message).                |          |        |      |
| :COUNT?         | Return number of errors in queue.                              |          |        |      |
| :CODE           | Return error code number only:                                 |          |        |      |
| [:NEXT]?        | Return and clear oldest error (code only).                     |          |        |      |
| :ALL?           | Return and clear all errors (codes only).                      |          |        |      |
| :CLEar          | Clears codes and messages from Error Queue.                    |          |        | ✓    |
| :MEP            | Select the GPIB protocol.                                      | (Note 2) | E syst |      |
| [:STATE] <b>    | Select SCPI protocol (1) or 488.1 protocol (0).                |          |        |      |
| [:STATE]?       | Query selected protocol.                                       |          |        |      |
| :VERSion?       | Query rev level of SCPI standard.                              |          |        | ✓    |
| :TIME           | Timestamp:   |          | F syst |      |
| :RESet          | Reset system timestamp to zero seconds.                        |          |        |      |
| :TIME?          | Query timestamp value.   |          |        |      |

## Notes:

1. Power-up and \*CLS clears the error queue. \*RST, SYSTem:PRESet, and STATus:PRESet has no effect on the error queue.
2. \*RST and SYSTem:PRESet have no effect on protocol. At the factory, the SCPI protocol is selected.

**Reference**

A syst

**System presets****SYSTem:PRESet**                      **Return 2800 to factory defaults**

Selects IMMEDIATE trigger mode and returns the instrument to the factory defaults. Default settings are listed in the SCPI tables under the “Default” column heading. Details on GPIB “[Triggering](#)” are provided in [Section 3](#).

B syst

**Power-on setup****SYSTem:POSet <name>**                      **Select power-on setup**

|        |   |        |                             |
|--------|---|--------|-----------------------------|
| <name> | = | RST    | *RST defaults               |
|        | = | PRESet | SYSTem:PRESet defaults      |
|        | = | SAVx   | User setup (x = 0, 1, or 2) |

This command is used to select the power-on defaults. The instrument can be set to power on to one of the factory default setups (\*RST or SYSTem:PRESet) or to a user-setup. Details on “Defaults & user setups” are provided on page 1-30 of the User’s Manual.

C syst

**Key-press codes****SYSTem:KEY <NRf>**                      **Simulate front panel key-press**

&lt;n&gt; = 1 to 31

**Parameters** The key codes are listed in [Table 2-10](#) and shown in [Figure 2-3](#). There are two sets of key codes listed in the table. The set of codes on the left side of the table are presented with the codes listed in numerical order. The set of codes on the right side present the keys as they are laid out on the front panel. The order of keys is from top to bottom and left to right.

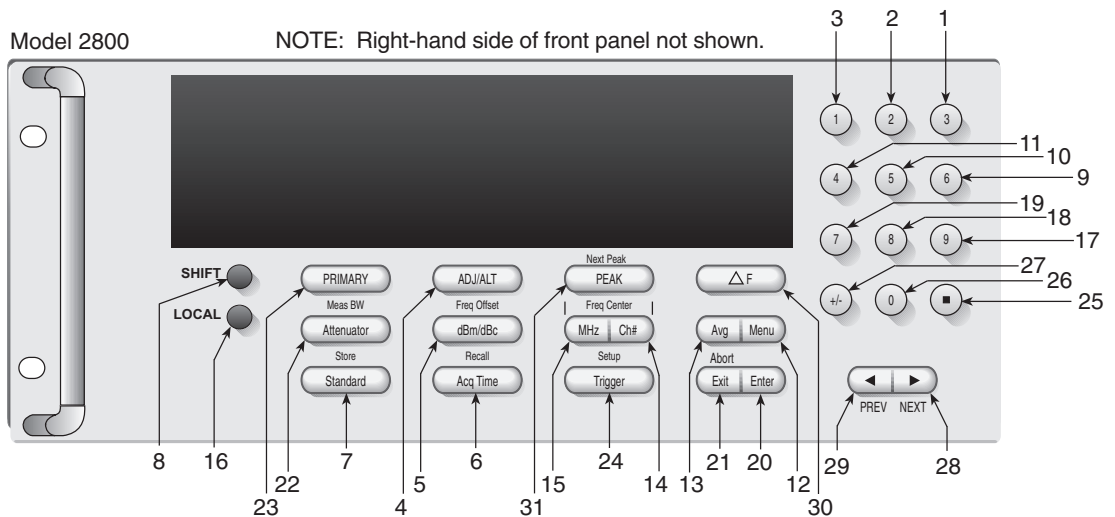
This command is used to simulate front panel key presses. For example, to select the power measurement function, send the following command to simulate pressing the “PRIMARY” key: SYSTem:KEY 23.

The queue for the :KEY? query command can only hold one key-press. When :KEY? is sent and Model 2800 is addressed to talk, the key-press code number for the last key “pressed” is sent to the computer. If no key has been “pressed,” SYST:KEY? returns 63.

Table 2-10  
Key-press codes

| Numerical Order |     |                |     | Top to Bottom/Left to Right |     |              |     |
|-----------------|-----|----------------|-----|-----------------------------|-----|--------------|-----|
| Key             | <n> | Key            | <n> | Key                         | <n> | Key          | <n> |
| 3 key           | 1   | 9 key          | 17  | 1 key                       | 3   | dBm/dBc key  | 5   |
| 2 key           | 2   | 8 key          | 18  | 2 key                       | 2   | MHz key      | 15  |
| 1 key           | 3   | 7 key          | 19  | 3 key                       | 1   | Ch# key      | 14  |
| ADJ/ALT key     | 4   | Enter key      | 20  | 4 key                       | 11  | Avg key      | 13  |
| dBm/dBc key     | 5   | Exit key       | 21  | 5 key                       | 10  | Menu key     | 12  |
| Acq Time key    | 6   | Attenuator key | 22  | 6 key                       | 9   | ± key        | 27  |
| Standard key    | 7   | PRIMARY key    | 23  | SHIFT key                   | 8   | 0 key        | 26  |
| SHIFT key       | 8   | Trigger key    | 24  | PRIMARY key                 | 23  | ■ key        | 25  |
| 6 key           | 9   | ■ key          | 25  | ADJ/ALT key                 | 4   | Standard key | 7   |
| 5 key           | 10  | 0 key          | 26  | PEAK key                    | 31  | Acq Time key | 6   |
| 4 key           | 11  | ± key          | 27  | ΔF                          | 30  | Trigger key  | 24  |
| Menu key        | 12  | NEXT key       | 28  | 7 key                       | 19  | Exit key     | 21  |
| Avg key         | 13  | PREV key       | 29  | 8 key                       | 18  | Enter key    | 20  |
| Ch# key         | 14  | ΔF             | 30  | 9 key                       | 17  | PREV key     | 29  |
| MHz key         | 15  | PEAK key       | 31  | LOCAL key                   | 16  | NEXT key     | 28  |
| LOCAL key       | 16  |                |     | Attenuator key              | 22  |              |     |

Figure 2-3  
Key-press codes



D syst

## Error queue

**SYSTem:ERRor[:NEXT]?**      Read oldest error (code and message)

As error and status messages occur, they are placed in the Error Queue. The Error Queue is a first-in first-out (FIFO) register that can hold up to 10 messages. After sending this command and addressing the Model 2800 to talk, the oldest code and message is sent to the computer and is then removed from the queue.

**NOTE**      The *STATus:QUEue[:NEXT]?* command (see [Table 2-8](#)) performs the same function as *SYSTem:ERRor[:NEXT]?*.

**SYSTem:ERRor:ALL?**      Read all errors (codes and messages)

This query command is similar to the *[:NEXT]?* command except that all messages in the Error Queue are sent to the computer when the Model 2800 is addressed to talk. All codes and messages are removed from the queue.

**SYSTem:ERRor:COUNT?**      Read the number of errors

After sending this command and addressing the Model 2800 to talk, a decimal number will be sent to the computer. That is the number of errors in the Error Queue.

**SYSTem:ERRor:CODE[:NEXT]?**      Read oldest error (code only)

This command is identical to the *:ERRor[:NEXT]?* command, except only the code is returned. The message itself is not returned. The error is cleared from the queue.

**SYSTem:ERRor:CODE:ALL?**      Read all errors (codes only)

This query command is identical to the *:ERRor:ALL?* command, except only the codes are returned. The actual messages are not returned. All errors are cleared from the queue.

**SYSTem:CLEar**      Clear error queue

This command clears all codes and messages from the error queue.

*E syst*

## **GPIB protocol**

**SYSTem:MEP[:STATe] <b>**      **Select GPIB protocol**  
**<b> = 1 or ON**            **Selects SCPI protocol**  
          **= 0 or OFF**        **Selects 488.1 protocol**

This command is used to select the GPIB protocol for the Model 2800:

SYSTem:MEP 1      ' Select SCPI protocol  
SYSTem:MEP 0      ' Select 488.1 protocol

At the factory, the SCPI protocol is selected. In general, the 488.1 protocol is faster, but has limitations associated with its use. See [Appendix C](#) for details on GPIB protocol.

*F syst*

## **Timestamp**

**SYSTem:TIME:RESet**            **Reset timestamp**  
**SYSTem:TIME?**                **Query timestamp value**

The Model 2800 can use a system timestamp for returned readings. The timer for the timestamp starts at zero seconds when the instrument is turned on.

Details on the system timestamp are provided in [“FORMat subsystem commands,” page 2-6](#).

## TRACe subsystem commands

The TRACe subsystem commands (Table 2-11) are used to store up to 2500 readings in the TRACe buffer for later recall. Statistics on buffer readings can also be performed. Use the “CALCulate3 subsystem commands” to select and read the result of a statistic.

Table 2-11

**TRACe command summary**

| Command         | Description                               | Default* | Ref    | SCPI |
|-----------------|---|----------|--------|------|
| TRACe           |   |          |        |      |
| :POINTs <n>     | Specify size of buffer: 1 to 2500.        | 100      | A trac | ✓    |
| :ACTual?        | Returns number of stored readings.        |          |        |      |
| :POINTs?        | Query buffer size.                        |          |        | ✓    |
| :TSTamp         | Timestamp type:                           |          | B trac |      |
| :FORMat <name>  | Select timestamp type: ABSolute or DELTa. | DELT     |        |      |
| :FORMat?        | Query timestamp type.                     |          |        |      |
| :FEED           | Control buffer:                           |          | C trac |      |
| :CONTrol <name> | Enable (NEXT) or disable (NEVer) buffer.  | NEVer    |        | ✓    |
| :CONTrol?       | Query state of buffer control.            |          |        | ✓    |
| :DATA?          | Returns readings stored in buffer.        |          | C trac | ✓    |
| :CLEar          | Clears the buffer.                        |          | C trac |      |

\*SYSTEM:PRESet and \*RST have an effect on these commands. The listed defaults are set at the factory.

### Reference

A trac

### **Buffer size**

**TRACe:POINTs <n>**                      **Set buffer size**  
 <n> = 1 to 2500                      **Number of readings to store**

This command is used to specify the number of readings (data arrays) to store in the buffer. A buffer size of zero is not valid (error -222).

The query form for this command (TRACe:POINTs?) returns the programmed number of readings to be stored. The next command query returns the actual number of readings stored in the buffer.

**TRACe:POINTs:ACTual?**                      **Query number of stored readings**

This command query returns the actual number of readings stored in the buffer.

B trac

## Timestamp

**TRACe:TSTamp:FORMat <name>**      **Select timestamp format**

<name> = **ABSolute**      **Referenced to zero seconds**  
           = **DELTA**      **Time difference between each reading**

Use this command to select timestamp format. The timestamp can be reference to 0.000 seconds (ABSolute) or it can indicate the time difference between each stored reading (DELTA).

C trac

## Buffer control

**TRACe:FEED:CONTRol <name>**      **Enable or disable the buffer**

<name> = **NEXT**      **Enable buffer**  
           = **NEVer**      **Disable buffer**

Use this command to enable or disable the buffer. When enabled, a reading will be stored into the buffer every time the instrument is triggered. If in the continuous measurement mode, storage will start as soon as the buffer is enabled.

The storage process can be stopped by selecting NEVer.

**TRACe:DATA?**      **Read buffer**

Use this command to return the readings stored in the buffer. The data elements that are returned with each reading are specified by the FORMat:ELEMents:TRACe command (see “[Data elements for readings](#)” for details).

**TRACe:CLEAr**      **Clear buffer**

Use this command to clear the buffer of all readings. Note that turning off the instrument clears the buffer.

**Example** – Store 100 readings in the buffer:

```
TRAC:CLE                    | Clear buffer.
TRAC:POIN 100              | Set buffer size to 100.
TRAC:TST:FORM ABS         | Select absolute timestamp.
TRAC:FEED:CONT NEXT      | Enable buffer.
TRIG:COUN 100             | Set trigger count to 100.
INIT                        | Start measure/storage process.
TRAC:DATA?                | Read buffer.
```



## Trigger commands

The trigger commands (Table 2-12) control the number of measurements to perform and when to take them. For remote operation, a trigger will initiate one or more measurements. Details on “Triggering” are provided in Section 3.

Table 2-12

**Trigger command summary**

| Command        | Description  | Default | Ref    | SCPI |
|----------------|--|---------|--------|------|
| ABORt          | Reset trigger system.  |         | A trig | ✓    |
| INITiate       | Initiation:  |         | B trig | ✓    |
| [:IMMEDIATE]   | Initiate one trigger cycle.                                      |         |        | ✓    |
| TRIGger        | Triggering:  |         | C trig |      |
| [:SEQUENCE[1]] |  |         |        |      |
| :SOURce <name> | Select control source: IMMEDIATE, EXTERNAL, BUS, or LEVEL.       | IMM     |        | ✓    |
| :SOURce?       | Query control source.  |         |        | ✓    |
| :COUNT <n>     | Set trigger count: 1 to 2500 or INFINITE.                        | 1       |        | ✓    |
| :COUNT?        | Query trigger count.   |         |        | ✓    |
| :DELAY <n>     | Set delay: 0 to 999.999ms.                                       | 0.00001 |        | ✓    |
| :DELAY?        | Query trigger delay.   |         |        | ✓    |
| :LEVEL <n>     | Set trigger level for LEVEL control source. -99.99 to +20 (dBm). | 0       |        |      |
| :LEVEL?        | Query trigger level.   |         |        |      |
| :TIMEOUT <n>   | Set the trigger timeout: 1 to 300 (seconds).                     | 5       |        |      |
| :TIMEOUT?      | Query the trigger timeout.                                       |         |        |      |

### Reference

#### A trig

#### ABORt

Resets the Model 2800 trigger system and places it in the idle state.

#### B trig

#### INITiate[:IMMEDIATE]

#### Immediate initiation

An initiate command takes the instrument out of idle to perform one or more measurements. The instrument will return to the idle state after performing the programmed number of measurements, as set by the trigger count.

**NOTE** Whenever the instrument is operating within the trigger model, sending an initiate command causes an error and will be ignored.

C trig

## TRIGGer commands

The TRIGger commands are used to select the control source, set trigger count, set trigger delay, and set the trigger level for level control source.

Details on these trigger commands are covered in “[Trigger model operation](#)” in [Section 3](#). The commands and parameters are listed as follows:

|  |                              |
|--|------------------------------|
| <b>TRIGger[:SEquence[1]]:SOURce &lt;name&gt;</b> | <b>Select control source</b> |
| <name> = IMMEDIATE                               | Immediate control source     |
| = BUS  | Bus control source           |
| = EXTERNAL                                       | External control source      |
| = LEVEL  | Level control source         |

Use this command to select the control source. With Immediate selected, event detection is satisfied immediately. With Bus selected, event detection is satisfied when a bus trigger (\*TRG or GET) is received.

With External selected, event detection is satisfied when an external trigger is received. With Level selected, event detection is satisfied when the programmed input power level is detected. The TRIG:SEQ:LEV command is used to set the trigger level. While waiting for detection while in External or Level, no further commands will be executed until the unit returns to idle.

|  |                          |
|--|--------------------------|
| <b>TRIGger[:SEquence[1]]:COUNT &lt;n&gt;</b> | <b>Set trigger count</b> |
| <n> = 1 to 2500                              | Trigger count            |

Use this command to set the trigger count. The trigger count determines the number of measurements to be performed during a trigger cycle.

|  |                            |
|--|----------------------------|
| <b>TRIGger[:SEquence[1]]:DELay &lt;n&gt;</b> | <b>Set trigger delay</b>   |
| <n> = 0 to 999.999ms                         | Trigger delay (in seconds) |

Use this command to set the trigger delay. It is typically used to allow the signal to settle before performing the measurement.

**TRIGger[:SEquence[1]]:TIMEout <n>**                      **Set trigger timeout**  
**<n> = 1 to 300**                      **Trigger timeout (in seconds)**

This command controls the timeout period for the Level and External trigger control sources. With the Level trigger source selected, a measurement is triggered when the specified input level is detected. With the External trigger source selected, a measurement is triggered when an input trigger from an external device is received.

After the instrument is taken out of idle (e.g., INIT, READ? or MEAS? sent), it waits for the Level or External trigger to occur. However, if the specified trigger timeout period expires before the trigger occurs, a timeout error (+703, Trigger Timeout Error) occurs and the Model 2800 returns to the idle state.

**TRIGger[:SEquence[1]]:LEVel <n>**                      **Set trigger level**  
**<n> = -99.99 to +20**                      **Set trigger level (dBm)**

With the Level control source selected, the level determines the input power level that will trigger a measurement. If the set level is not detected, a measurement will not occur.

## UNIT subsystem commands

The unit subsystem is used to select the units designator for power readings. The command to set units is listed in [Table 2-13](#).

Table 2-13

### UNIT command summary

| Command       | Description  | Default | Ref | SCPI |
|---------------|--|---------|-----|------|
| UNIT          |  |         |     |      |
| :POWer <name> | Set units for adjacent and alternate channel readings. Power readings: DBM or DBC. | DBM     | —   | ✓    |
| :POWer?       | Query units.   |         |     | ✓    |

### Reference

**UNIT:POWer <name>**                      **Select units for adjacent and alternate channels.**

<name> = **DBM**              **dBm power units**

          = **DBC**              **dBc power units**

Use this command to select the units for adjacent and alternate channel power readings. For dBm units, power is referenced to 1mW. For dBc, the reading result is the difference between the Primary and Alternate/Adjacent readings (in dB).

Regardless of the unit setting, Primary measurements will only be displayed in dBm units. For Alternate and Adjacent measurements, the unit can be dBm or dBc.

## Signal-oriented commands

The signal oriented measurement commands are used to acquire readings. You can use these high-level instructions to control the measurement process. These commands are summarized in [Table 2-14](#).

Table 2-14

### Signal-oriented measurement command summary

| Command  | Description   | Ref                   |
|----------|---|-----------------------|
| FETCh?   | Returns the latest reading(s).                          | <a href="#">A sig</a> |
| READ?    | Triggers one or more readings and returns the readings. | <a href="#">B sig</a> |
| MEASure? | Triggers one or more readings and returns the readings. | <a href="#">C sig</a> |

**NOTE** For details on using the above signal-oriented measurement commands, see the “Summary of read commands” on page 1-27 of the User’s Manual. It also explains restrictions associated with these commands.

### Reference

*A sig*

#### **FETCh?** Return latest reading(s)

This query command requests the latest post-processed readings stored in the sample buffer (see “[Remote \(GPIB\) Operation](#)” in [Section 1](#)). After sending this command and addressing the Model 2800 to talk, the readings are sent to the computer. This command does not affect the instrument setup.

This command does not trigger measurements. It simply requests the last available readings. Note that this command can repeatedly return the same readings. Until there are new readings, this command continues to return the old readings. For example, assume that the Model 2800 performed 20 measurements. The FETCh? command will request the readings for those 20 measurements. If FETCh? is sent while performing measurements, it will not be executed until the Model 2800 goes back into idle.

The readings that are acquired depend on which data elements are selected (see “[Data elements for readings](#)” for details). Functions not measured are assigned the NAN (not a number) value of +9.91e37.

For example, assume that Primary and Adjacent readings are selected as data elements and the instrument performs one measurement cycle. The reading string returned by FETCh? will include the Primary and Adjacent readings. The readings for the Alternate channel will be NANs since they are not measured.

*B sig***READ?** **Trigger and return latest reading(s)**

This command is used to trigger and acquire readings. The number of readings depends on the trigger count. If configured for 20 measurements (trigger count = 20), then 20 sets of readings will be acquired after the Model 2800 returns to the idle state.

When this command is sent, the following commands execute in the order that they are presented:

- INITiate
- FETCh?

The INITiate command starts operation by taking the instrument out of idle. After all measurements are completed, the Model 2800 goes back into idle at which time the FETCh? command is executed. The readings are sent to the computer and displayed when the Model 2800 is addressed to talk.

**NOTE** *The READ? command cannot be used with the BUS control source selected. Details on “Triggering” are provided in [Section 3](#).*

*C sig***MEASure?** **Trigger and return latest reading(s)**

This command is also used to trigger and acquire readings. It operates the same as the READ? command.

**NOTE** *The READ? command cannot be used with the BUS control source selected. Details on “Triggering” are provided in [Section 3](#).*

## Common commands

Common commands (listed in [Table 2-15](#)) are device commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard.

*Table 2-15*  
**IEEE-488.2 common commands and queries**

| Command             | Name                          | Description  | Ref                   |
|---------------------|-------------------------------|--|-----------------------|
| *CLS                | Clear status                  | Clears all event registers and Error Queue.  | <a href="#">A com</a> |
| *ESE <NRf> or <NDN> | Standard Event Enable command | Program the Standard Event Enable Register.  | <a href="#">A com</a> |
| *ESE?               | Event enable query            | Read the Standard Event Enable Register.   | <a href="#">A com</a> |
| *ESR?               | Event status register query   | Read and clear the Standard Event Enable Register.   | <a href="#">A com</a> |
| *IDN?               | Identification query          | Returns the manufacturer, model number, serial number, and firmware revision levels of the unit.             | <a href="#">B com</a> |
| *OPC                | Operation complete            | Set the Operation Complete bit in the Standard Event Register after all pending commands have been executed. | <a href="#">B com</a> |
| *OPC?               | Operation complete query      | Places an ASCII "1" into the Output Queue when all pending selected device operations have been completed.   | <a href="#">B com</a> |
| *RCL <NRf>          | Recall                        | Returns the 2800 to the user-saved setup.  | <a href="#">C com</a> |
| *RST                | Reset                         | Returns the 2800 to the *RST default conditions.   | <a href="#">D com</a> |
| *SAV <NRf>          | Save                          | Saves the present setup as the user-saved setup.   | <a href="#">C com</a> |
| *SRE <NRf> or <NDN> | Service request enable        | Programs the Service Request Enable Register.  | <a href="#">A com</a> |
| *SRE?               | Service request enable query  | Reads the Service Request Enable Register.   | <a href="#">A com</a> |
| *STB?               | Status byte query             | Reads the Status Byte Register.  | <a href="#">A com</a> |
| *TRG                | Trigger                       | Sends a bus trigger to the Model 2800.   | <a href="#">E com</a> |
| *TST?               | Self-test query               | Performs a checksum test on ROM and returns the result.  | <a href="#">F com</a> |
| *WAI                | Wait-to-continue              | Wait until all previous commands are executed.   | <a href="#">G com</a> |

**Reference***A com***Status structure**

\*CLS, \*ESE, \*ESE?, \*ESR?, \*SRE, \*SRE?, and \*STB? are common associated with the status structure. For details, see “[Status structure](#)” in [Section 3](#).

**Parameters for \*ESE AND \*SRE commands:**

|       |   |          |                    |                   |
|-------|---|----------|--------------------|-------------------|
| <NRf> | = | 0 to 255 | Decimal format     |                   |
| <NDN> | = | #Bx      | Binary format      | x = 0 to 11111111 |
|       | = | #Hx      | Hexadecimal format | x = 0 to FF       |
|       | = | #Qx      | Octal format       | x = 0 to 377      |

The parameter values to set the individual bits of the enable registers can be set using the decimal format (NRf) or a non-decimal format (NDN). Details on “[Programming enable registers](#)” are provided in [Section 3](#).

When reading a status register, the presently set data format determines the type of value (decimal or non-decimal) that is returned. The “[FORMat:SREGister <name>](#)” command is used to set the data format. Details on “[Programming and reading registers](#)” are provided in [Section 3](#).

*B com***\*IDN? — identification query      Reads identification code**

The identification code includes the manufacturer, model number, serial number, and firmware revision levels and is sent in the following format:

```
KEITHLEY INSTRUMENTS INC., MODEL 2800, nnnnnnn, xxx Oct 2 2002
16:59:04 /yyy /zzz
```

Where:

- nnnnnnn is the serial number.
- xxx is the firmware revision level.
- Oct 2 2002 16:59:04 is the date and time (24hr format, hr:min:sec) for the firmware release.
- yyy is the display board revision.
- zzz is the DSP firmware revision.



|   |                                     |
|---|-------------------------------------|
| <b>*OPC — operation complete</b>        | <b>Sets OPC bit</b>                 |
| <b>*OPC? — operation complete query</b> | <b>Places a “1” in output queue</b> |

When \*OPC is sent, the OPC bit in the Standard Event Register will set after all pending command operations are complete. When \*OPC? is sent, an ASCII “1” is placed in the Output Queue after all pending command operations are complete.

Typically, either one of these commands is sent after the INITiate command. The INITiate command is used to take the instrument out of idle in order to perform measurements. While operating within the trigger model layers, all sent commands (except DCL, SDC, IFC, SYSTem:PRESet, \*RST, \*RCL, \*TRG, GET, and ABORt) will not execute.

After all programmed operations are completed, the instrument returns to the idle state at which time all pending commands (including \*OPC and/or \*OPC?) are executed. After the last pending command is executed, the OPC bit and/or an ASCII “1” is placed in the Output Queue.

#### **\*OPC programming example**

The following command sequence will perform 10 measurements. After the measurements are completed (in approximately 10ms), an ASCII “1” will be placed in the Output Queue and displayed on the computer CRT. Note that additional codes must be added to query the instrument for the presence of the ASCII “1” in the Output Queue.

```
*RST           ' Return 2800 to *RST defaults.
TRIG:DEL 1e-3  ' Set trigger delay for 1ms.
TRIG:COUN 10   ' Program for 10 measurements and stop.
INIT          ' Start measurement process.
*OPC?         ' Send *OPC? to query Output Queue.
```

*C com*

|                                  |   |
|----------------------------------|---|
| <b>*SAV &lt;NRf&gt; — save</b>   | <b>Save present setup in memory</b>     |
| <b>*RCL &lt;NRf&gt; — recall</b> | <b>Return to setup stored in memory</b> |
| <b>&lt;NRf&gt; = 0</b>           | <b>Memory location 0</b>                |
| <b>= 1</b>                       | <b>Memory location 1</b>                |
| <b>= 2</b>                       | <b>Memory location 2</b>                |

Use the \*SAV command to save the present instrument setup configuration in memory for later recall. Any control affected by \*RST can be saved by the \*SAV command. The \*RCL command is used to restore the instrument to the saved setup configuration. Three setup configurations can be saved and recalled.

The Model 2800 ships from the factory with SYSTem:PRESet defaults loaded into the available setup memory. If a recall error occurs, the setup memory defaults to the SYSTem:PRESet values.

### **\*SAV, \*RCL programming example**

The following summarizes the basic command sequence for saving and recalling a setup. The present setup is stored in memory location 2, GPIB defaults are restored, and the memory location 2 setup is recalled.

```
*SAV 2          ' Save present setup in memory location 2.
*RST            ' Restore GPIB defaults.
*RCL 2          ' Recall location 2 setup.
```

### *D com*

#### **\*RST — reset**                      **Return Model 2800 to factory defaults**

When the \*RST command is sent, the Model 2800 performs the following operations:

- Returns the Model 2800 to the factory default conditions (see “Default” column of SCPI tables).
- Places the instrument in a “one-shot” trigger mode.
- Cancels all pending commands.
- Cancels response to any previously received \*OPC and \*OPC? commands.

### *E com*

#### **\*TRG — trigger**                      **Send bus trigger to Model 2800**

Use the \*TRG command to issue a GPIB trigger to the Model 2800. It has the same effect as a group execute trigger (GET).

Use the \*TRG command as an event to control operation. The Model 2800 reacts to this trigger if BUS is the programmed control source. Details on “[Triggering](#)” are covered in [Section 3](#).

### **\*TRG programming example**

The following command sequence configures the Model 2800 to be controlled by bus triggers. The last command, which sends a bus trigger, triggers one measurement. Each subsequent bus trigger will also trigger a single measurement.

**NOTE** With *TRIG:SOURce BUS* selected, do not send any commands (except *\*TRG*, *GET*, *DCL*, *SDC*, *IFC*, and *ABORT*) while performing measurements. Otherwise, erratic operation will occur.

|                      |                                  |
|----------------------|----------------------------------|
| <i>*RST</i>          | ' Restore GPIB defaults.         |
| <i>TRIG:SOUR BUS</i> | ' Select the BUS control source. |
| <i>INIT</i>          | ' Take 2800 out of idle.         |
| <i>*TRG</i>          | ' Trigger one measurement.       |

### F com

#### **\*TST? — self-test query**                      **Run self test and read result**

Use this query command to perform a checksum test on ROM. The command places the coded result (0 or 1) in the Output Queue. When the Model 2800 is addressed to talk, the coded result is sent from the Output Queue to the computer.

A returned value of zero (0) indicates that the test passed and a value of one (1) indicates that the test failed.

### G com

#### **\*WAI — wait-to-continue**                      **Wait until previous commands are completed**

Effectively, the *\*WAI* command is a No-Op (no operation) for the Model 2800 and thus, does not need to be used.

Two types of device commands exist:

- Sequential commands — A command whose operations are allowed to finish before the next command is executed.
- Overlapped commands — A command that allows the execution of subsequent commands while device operations of the Overlapped command are still in progress.

The *\*WAI* command is used to suspend the execution of subsequent commands until the device operations of all previous Overlapped commands are finished. The *\*WAI* command is not needed for Sequential commands.

## General bus commands

General commands are those commands, such as DCL, that have the same general meaning regardless of the instrument. [Table 2-16](#) lists the general bus commands.

Table 2-16

### **General bus commands and associated statements**

| Command  | Effect on Model 2800  | Ref                   |
|----------|---|-----------------------|
| REN      | Goes into effect when next addressed to listen.             | <a href="#">A gen</a> |
| IFC      | Reset interface. Goes into talker and listener idle states. | <a href="#">B gen</a> |
| LLO      | LOCAL key locked out.                                       | <a href="#">C gen</a> |
| GTL      | Cancel remote: Restore Model 2800 front panel operation.    | <a href="#">D gen</a> |
| DCL      | Returns all devices to known conditions.                    | <a href="#">E gen</a> |
| SDC      | Returns Model 2800 to known conditions.                     | <a href="#">F gen</a> |
| GET      | Initiates a trigger.  | <a href="#">G gen</a> |
| SPE, SPD | Serial polls the Model 2800.                                | <a href="#">H gen</a> |

### **Reference**

#### *A gen*

#### **REN (remote enable)**

The remote enable command is sent to the Model 2800 by the controller to set up the instrument for remote operation. Generally, the instrument should be placed in the remote mode before you attempt to program it over the bus. Setting REN true does not place the instrument in the remote state. You must address the instrument to listen after setting REN true before it goes into remote.

The Model 2800 must be in remote in order to use the following commands to trigger and acquire readings:

- :INITiate and then :FETCh?
- :READ?
- :MEASure?

Note that all front panel controls except for LOCAL (and POWER) are inoperative while the instrument is in remote. You can restore normal front panel operation by pressing the LOCAL key.

*B gen***IFC (interface clear)**

The IFC command is sent by the controller to place the Model 2800 in the local, talker, listener idle states. The unit responds to the IFC command by canceling front panel TALK or LSTN lights, if the instrument was previously placed in one of these states.

Note that this command does not affect the status of the instrument. Settings, data, and event registers are not changed.

To send the IFC command, the controller need only set the IFC line true for a minimum of 100 $\mu$ s.

*C gen***LLO (local lockout)**

Use the LLO command to prevent local operation of the instrument. After the unit receives LLO, all of its front panel controls are inoperative. In this state, pressing LOCAL will not restore control to the front panel. The GTL command restores control to the front panel. Cycling power will also cancel local lockout.

*D gen***GTL (go to local)**

Use the GTL command to put a remote-mode instrument into local mode. The GTL command also restores front panel key operation.

*E gen***DCL (device clear)**

Use the DCL command to clear the GPIB interface and return it to a known state. Note that the DCL command is not an addressed command, so all instruments equipped to implement DCL will do so simultaneously.

When the Model 2800 receives a DCL command, it clears the Input Buffer and Output Queue, cancels deferred commands, and clears any command that prevents the processing of any other device command. A DCL does not affect instrument settings and stored data.

*F gen***SDC (selective device clear)**

The SDC command is an addressed command that performs essentially the same function as the DCL command. However, since each device must be individually addressed, the SDC command provides a method to clear only selected instruments instead of clearing all instruments simultaneously, as is the case with DCL.

*G gen***GET (group execute trigger)**

GET is a GPIB trigger that is used as trigger event to control operation. The Model 2800 reacts to this trigger if it is the programmed control source. The following command selects the GPIB arm control source:

TRIGGer:SOURce BUS

**NOTE** *With TRIG:SOUR BUS selected and an initiate command sent, do not send any commands (except GET, DCL, SDC, IFC, \*TRG, and ABORt) while performing measurement. If you do, erratic operation will occur.*

*H gen***SPE, SPD (serial polling)**

Use the serial polling sequence to obtain the Model 2800 serial poll byte. The serial poll byte contains important information about internal functions (see “[Status structure](#)” in [Section 3](#)). Generally, the serial polling sequence is used by the controller to determine which of several instruments has requested service with the SRQ line. However, the serial polling sequence may be performed at any time to obtain the status byte from the Model 2800.

# 3

# Triggering, Status Structure, and Digital I/O

---

## Section topics

### Triggering

#### Trigger model

Idle and initiate

Trigger model operation

#### External triggering

External trigger input

Meter Complete output

Connections

Selecting external triggering

#### Trigger configuration

Front panel operation

Triggering commands

### Status structure

#### Operational overview

#### Clearing registers and queues

#### Programming and reading registers

Programming enable registers

Reading registers

#### Status byte and service request (SRQ)

Status Byte Register

Service Request Enable Register

Serial polling and SRQ

Status byte and service request  
commands

Serial poll programming example

### Status structure (cont.)

#### Status register sets

Register bit descriptions

Standard event register

Operation event register

Measurement event register

Questionable event register

Condition registers

Event registers

Event enable registers

#### Queues

Output queue

Error queue

### Digital I/O

#### Digital input

External trigger input

#### Digital outputs

Sink mode — controlling external devices

Source mode (logic control)

#### Setting digital output lines

# Triggering

In general, there are two basic trigger modes for the Model 2800: continuous and one-shot. When using front panel operation, the instrument is in the continuous trigger mode. That is, measurements are performed continuously.

For remote operation, the instrument is in a one-shot trigger mode. Initially, the instrument will be in the idle state (no measurements). When the Model 2800 is triggered, one or more measurements will occur. With a finite trigger count specified, the instrument will return to the idle state when finished performing measurements.

## Trigger model

The following paragraphs describe how the Model 2800 operates for remote operation. The flow chart in [Figure 3-1](#) summarizes operation over the bus. The flow chart is called the trigger model because operation is controlled by SCPI commands from the Trigger subsystem. Key SCPI commands are included in the trigger model.

### Idle and initiate

The instrument is considered to be in the idle state whenever operation is at the top of the trigger model. As shown in [Figure 3-1](#), initiation needs to be satisfied to take the instrument out of idle. While in the idle state, the instrument cannot perform any measure operations.

The following commands will return operation to the top of the trigger model (idle) at the START point of the trigger model:

- ABORt
- SDC, DCL, or IFC
- \*RCLx (x =0, 1, or 2)
- SYSTem:PRESet or \*RST

While in the idle state, the following commands satisfy initiation to take the instrument out of idle:

- INITiate
- READ?
- MEAS?

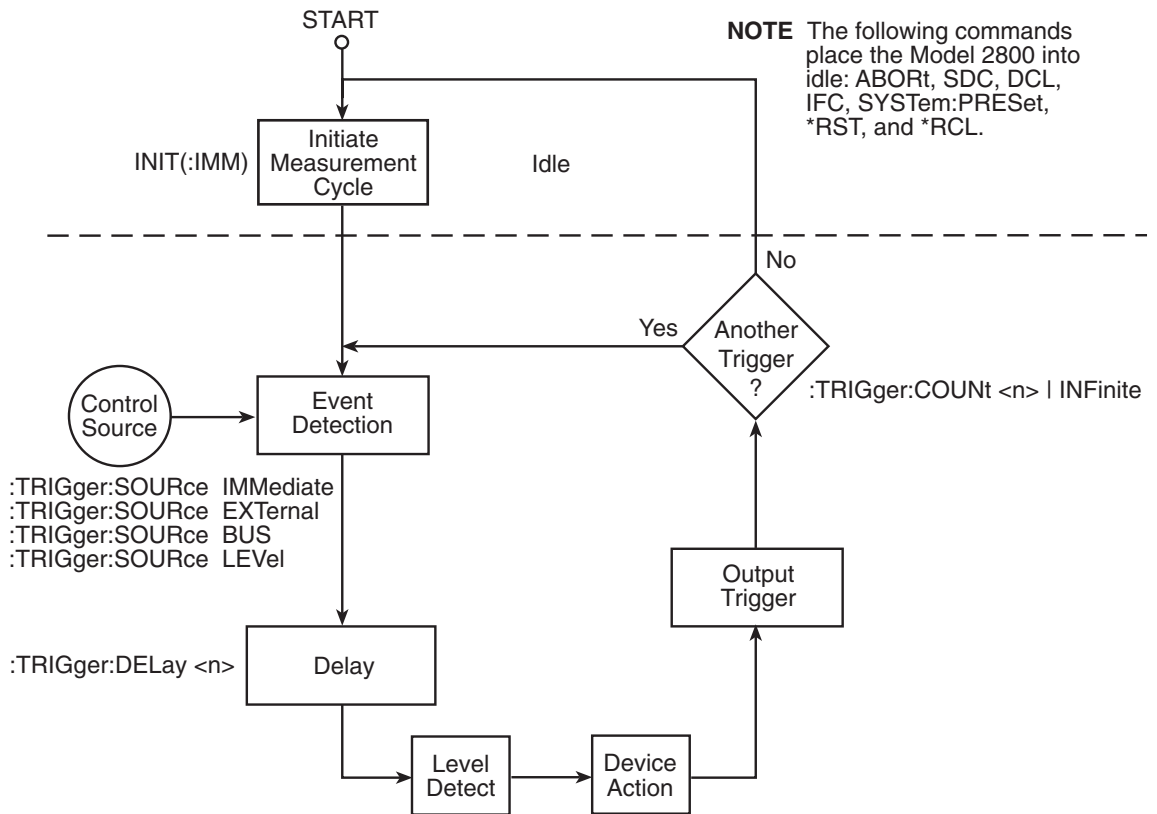
After sending INIT, READ?, or MEAS?, any further commands (except for \*TRG, GET, SDC, DCL, \*RST, and SYST:PRES) will not be processed until the unit returns to idle.



After sending INIT, READ?, or MEAS? with the LEVEL or EXTERNAL trigger source selected, no further commands will be executed until the unit returns to idle.

**NOTE** While in remote, pressing the LOCAL key restores continuous front panel operation.

Figure 3-1  
Trigger model (remote operation)



## Trigger model operation

Once the instrument is taken out of idle, operation proceeds through the trigger model down to the device action. In general, the device action performs a measurement.

**Control Source** — As shown in [Figure 3-1](#), a control source is used to hold up operation until the programmed event occurs. The control source options are as follows:

- **IMMediate** — Event detection is immediately satisfied allowing operation to continue.
- **EXTernal** — Event detection occurs when an input trigger is received (typically via the External Trigger connector). While waiting for the trigger, no further commands will be executed until the unit returns to idle.
- **BUS** — Event detection occurs when a bus trigger (\*TRG or GET) is received. With the Model 2800 in local, pressing the **Trigger** key also triggers the instrument.
- **LEVel** — Event detection occurs when a specified input power level is detected. While waiting for the trigger level, no further commands will be executed until the unit returns to idle.

**NOTE** *Should the unit lock up while waiting for a Level trigger, press **SHIFT** and then **Abort** to restore control of the front panel keys.*

**Delay** — A programmable delay is available after event detection. A delay is typically used to allow an external source to settle before performing a measurement. The delay can be set from 0 to 999.999ms.

**Level Detection** — When using the Level control source, the operation holds up until the programmed level at the input is detected. Level can be set from -99.99 to +20dBm.

**Device Action** — The Device Action performs the averaging process (if enabled) and yields the final measured reading. With averaging enabled, the instrument measures the specified number of samples and then uses an averaging algorithm to yield a single averaged reading. With averaging disabled, the Model 2800 performs one measurement.

**Output Trigger** — After the device action, the Model 2800 generates an output trigger. The output trigger is applied to the Meter Complete connector on the rear panel. It can be used to trigger an external instrument to perform an operation.

**NOTE** The Model 2800 generates an output trigger in EXTERNAL trigger mode only.

**Trigger Counter** — A programmable counter is used to repeat operations within the trigger model. The counter can be set from 1 to 2500. For example, assume the IMMEDIATE control source is selected. To acquire 10 readings, you can set the trigger count to 10 and send the READ? command. After the 10 readings are acquired, they will be sent to the computer when the Model 2800 is addressed to talk. The trigger count can instead be set to INFINITE. The instrument remains in the trigger model to continuously perform measurements.

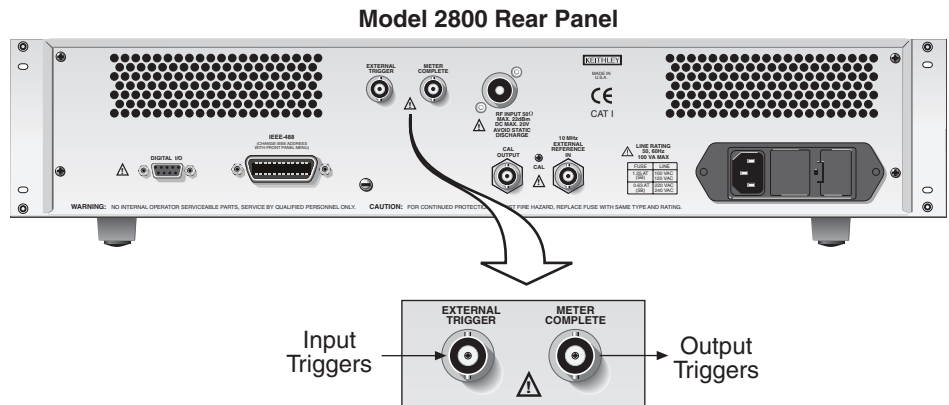
## External triggering

An external device or circuit can be used to trigger one or more measurements by the Model 2800. With external triggering selected, the Model 2800 will be triggered when a falling-edge TTL-compatible pulse is received via the EXTERNAL TRIGGER connector.

With external triggering selected, the Model 2800 will output a TTL pulse to its METER COMPLETE connector after every measurement. This output pulse can be used to trigger an external device or circuit to perform a particular operation.

Figure 3-2 shows the rear panel BNC connectors that are used for external triggering.

Figure 3-2  
**EXTERNAL TRIGGER and METER COMPLETE connectors**

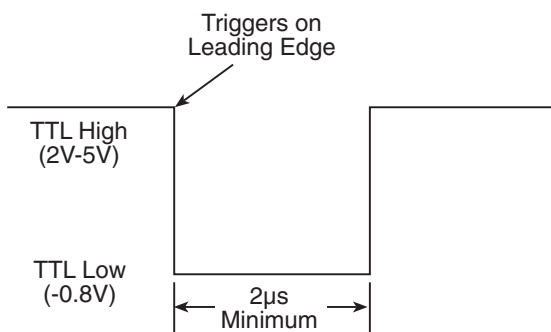


## External trigger input

The EXTERNAL TRIGGER input requires a falling-edge, TTL-compatible pulse with the specifications shown in [Figure 3-3](#). In general, external triggers can be used to control measure operations. For the Model 2800 to respond to external triggers, the trigger model must be configured for it.

Figure 3-3

### Trigger input pulse specifications (External Trigger)



## Meter Complete output

The Meter Complete output provides a TTL-compatible output pulse that can be used to trigger other instruments. Meter complete is only active when the External Trigger source is selected.

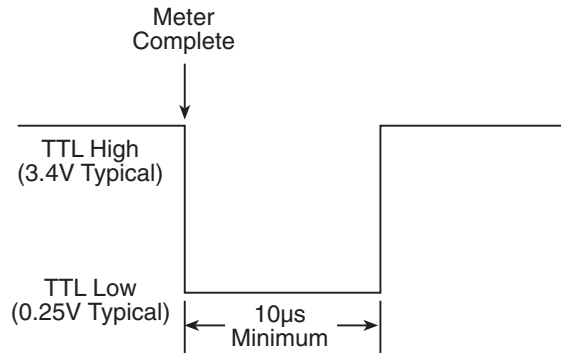
The specifications for this trigger pulse are shown in [Figure 3-4](#). Typically, you would want the Model 2800 to output a trigger after the settling time of each measurement.

## Repeat averaging

For external triggering, the instrument must receive N number of triggers to yield a single averaged reading (where N is the averaging count). For example, assume the Model 2800 is configured to measure the primary channel only, and the averaging count (N) is set to 10. Each received trigger will cause a measurement to occur and temporarily store it in an averaging stack (buffer). After the stack fills (10 readings stored), they are averaged to yield a single displayed averaged reading.

Modify the above example so that the primary channel, upper adjacent channel, lower adjacent channel, upper alternate channel and lower alternate channel are measured. The process is the same except that an averaged reading is yielded for every channel.

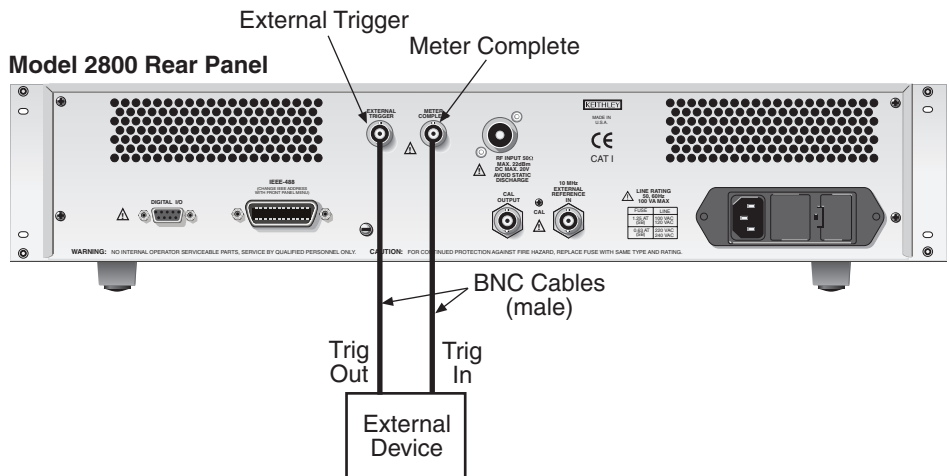
**Figure 3-4**  
**Trigger output pulse specifications (Meter Complete)**



### Connections

The rear panel connectors for EXTERNAL TRIGGER and METER COMPLETE are female BNC connectors. Use standard BNC cables terminated with male BNC connectors to make connections to the Model 2800. Figure 3-5 shows typical external triggering connections to an external device.

**Figure 3-5**  
**Typical external triggering connections**



## Selecting external triggering

From the front panel, external triggering is selected from the trigger setup menu which is accessed by pressing **SHIFT** and then **SETUP**. From the **SOURCE** menu item, select the **EXTERNAL** control source. See “[Trigger configuration](#)” for details on using the trigger setup menu.

For remote programming, the Model 2800 is placed into the external trigger mode by selecting the EXTERNAL trigger source with the following command:

```
TRIG:SEQ:SOUR EXT
```

All triggering commands are summarized in [Table 3-1](#).

**NOTE** *In order to use the front panel **TRIGGER** key, the instrument must first be taken out of remote by pressing the **LOCAL** key or sending the GTL command (see “[General bus commands](#)” in [Section 2](#)).*

## Trigger configuration

### Front panel operation

To configure triggering, press **SHIFT** and then **SETUP** to access the trigger menu.

1. **Source** – Select the trigger source:
  - a. Use the **PREV** or **NEXT** key to place the blinking cursor on **SOURCE** and press **Enter**.
  - b. Use the **PREV** or **NEXT** key to place the blinking cursor on the desired source (**EXTERNAL**, **IMMEDIATE**, **BUS**, or **LEVEL**) and press **Enter**.  
If you selected **LEVEL**, you will be prompted to enter the trigger level (-99.99 to +20.00dBm). Use the numerical keypad and the **PREV/NEXT** keys to key in the desired trigger level and press **Enter**.
2. **Delay** – Enter the trigger delay time:
  - a. Press the **NEXT** key to place the blinking cursor on **DELAY** and press **Enter** to display the present trigger delay (in milliseconds). The maximum delay that can be set depends on the selected transmission standard. See the chart for “[Measurement Parameters \(Trigger Delay Range\)](#)” in [Appendix A – Specifications](#).
  - b. Use the numerical keypad and the **PREV/NEXT** keys to key in the desired trigger delay and press **Enter**.

3. **Count** – Set the trigger count:
  - a. Use the **PREV** or **NEXT** key to place the blinking cursor on **COUNT** and press **Enter**.
  - b. Use the **PREV** or **NEXT** key to place the blinking cursor on **FINITE** or **INFINITE** and press **Enter**.  
 If you selected **FINITE**, you will be prompted to enter the trigger count value (1 to 2500). Use the numerical keypad and the **PREV/NEXT** keys to key in the desired trigger count and press **Enter**.
4. Press **Exit** to return to the measure display state.

### Triggering commands

Commands for triggering are summarized in [Table 3-1](#).

Table 3-1  
**SCPI commands — triggering**

| Commands                  | Description   | Default | Ref    |
|---------------------------|---|---------|--------|
| ABORt                     | Reset trigger system.   |         | A trig |
| INITiate<br>[:IMMediate]  | Initiation:<br>Initiate trigger cycle.  |         | B trig |
| FEtCh?                    | Request the last reading(s).  |         | C trig |
| READ?                     | Trigger measurement(s) and request reading(s).  |         | C trig |
| MEASure?                  | Trigger measurement(s) and request reading(s).  |         | C trig |
| TRIGger<br>[:SEQuence[1]] |   |         | D trig |
| :SOURce <name>            | Select control source: IMMEDIATE, EXTERNAL, BUS, or LEVEL.                                    | IMM     |        |
| :COUNT <n>                | Set trigger count: 1 to 2500 or INFINITE.   | 1       |        |
| :DELay <n>                | Set delay: 0 to 999.999ms.  | 10µs    |        |
| :LEVel <n>                | Set trigger level to start measurement: -99.99 to +20 (dBm).                                  | 0       |        |
| :TIMEout <n>              | Set trigger timeout: 1 to 300 (seconds).  | 5       |        |
| *RST                      | Restore factory defaults (see “Default” column of this table). Places 2800 in the idle state. |         | E trig |
| SYSTem:PRESet             | Restore factory defaults (see “Default” column of this table). Places 2800 in idle state.     |         | E trig |

**Reference***A trig***ABORt**

The ABORt command resets the Model 2800 trigger system and places the instrument in the idle state. The Model 2800 will stay in the idle state until an INITiate command is received.

*B trig***INITiate[:IMMediate]                      Immediate initiation**

The INITiate command takes the instrument out of idle to perform one or more measurements. For a finite trigger count, the instrument will return to the idle state after performing the programmed number of measurements.

**NOTE**    *Whenever the instrument is operating within the trigger model, sending the INIT command causes an error and will be ignored.*

*C trig***FETCh?, READ?, and MEASure?**

FETCh? returns the last measured reading, It does not trigger a measurement. It simply returns the last reading.

The MEAS? or READ? will trigger and return one or more readings. Both of these commands are identical.

**NOTE**    *The “Summary of read commands”, on page 1-28 of the User’s Manual, explains how to use these commands to trigger and return readings. It also explains restrictions associated with these commands.*

*D trig***TRIGGer commands:**

The TRIGGer commands are used to select the control source, set trigger count, set trigger delay, and set the trigger level for the level control source.

Details on these trigger commands are covered in “[Trigger model operation](#)”. The commands and parameters are listed as follows:

| <b>TRIGGer[:SEquence[1]]:SOURce &lt;name&gt;</b> | <b>Select control source</b>    |
|--|---------------------------------|
| <b>&lt;name&gt; = IMMEDIATE</b>                  | <b>Immediate control source</b> |
| <b>= BUS</b>                                     | <b>Bus control source</b>       |
| <b>= EXTERNAL</b>                                | <b>External control source</b>  |
| <b>= LEVEL</b>                                   | <b>Level control source</b>     |



Use this command to select the control source. With Immediate selected, event detection is satisfied immediately. With Bus selected, event detection is satisfied when a bus trigger (\*TRG or GET) is received. Pressing the **TRIGGER** key also satisfies event detection (see “[External triggering](#)” for details).

**NOTE** *With the Bus control source selected, READ? and MEAS? cannot be used.*

With External selected, event detection is satisfied when an external trigger is received via the External Trigger connector.

With Level selected, event detection is satisfied when the programmed input power level is detected. The TRIG:SEQ:LEV command is used to set the trigger level.

**TRIGger[:SEQuence[1]]:COUNT <n>** **Set trigger count**  
 <n> = 1 to 2500 **Finite trigger count**  
       = INFinite **Infinite trigger count**

Use this command to set the trigger count. For a finite trigger count, the specified number of measurements are performed. When finished, the instrument returns to the idle state.

For the infinite trigger count, the instrument remains within the trigger model to continuously perform measurements.

**TRIGger[:SEQuence[1]]:DELay <n>** **Set trigger delay**  
 <n> = 0 to 999.999ms **Trigger delay (in seconds)**

Use this command to set the trigger delay before each measurement. It is typically used to allow the signal to settle before performing the measurement.

**TRIGger[:SEQuence[1]]:TIMEout <n>**                      **Set trigger timeout**  
 <n> = 1 to 300                      **Trigger timeout (in seconds)**

This command controls the timeout period for the level and external trigger control sources. With the level trigger source selected, a measurement is triggered when the specified input level is detected. With the external trigger source selected, a measurement is triggered when an input trigger from an external device is received.

**TRIGger[:SEQuence[1]]:LEVel <n>**                      **Set trigger level**  
 <n> = -99.99 to +20                      **Set trigger level (dBm)**

With the Level control source selected, the level determines the input power level that will trigger a measurement. If the set level is not detected, a measurement will not occur.

*E trig*

### ***Reset commands:***

**\*RST**  
**SYSTem:PRESet**

Both of these commands return the instrument to its factory defaults and returns the instrument to the idle state. The defaults are listed in the “Default” column of the SCPI tables.

The only difference between \*RST and SYST:PRESet is that the byte order for binary data transfer is reversed:

For \*RST, the default is NORMal (FORM:BORD NORM).

For SYST:PRESet, the default is SWAPPed (FORM:BORD SWAP).

Details on “[Byte order](#)” for binary data transfer is provided in [Section 2](#).

# Status structure

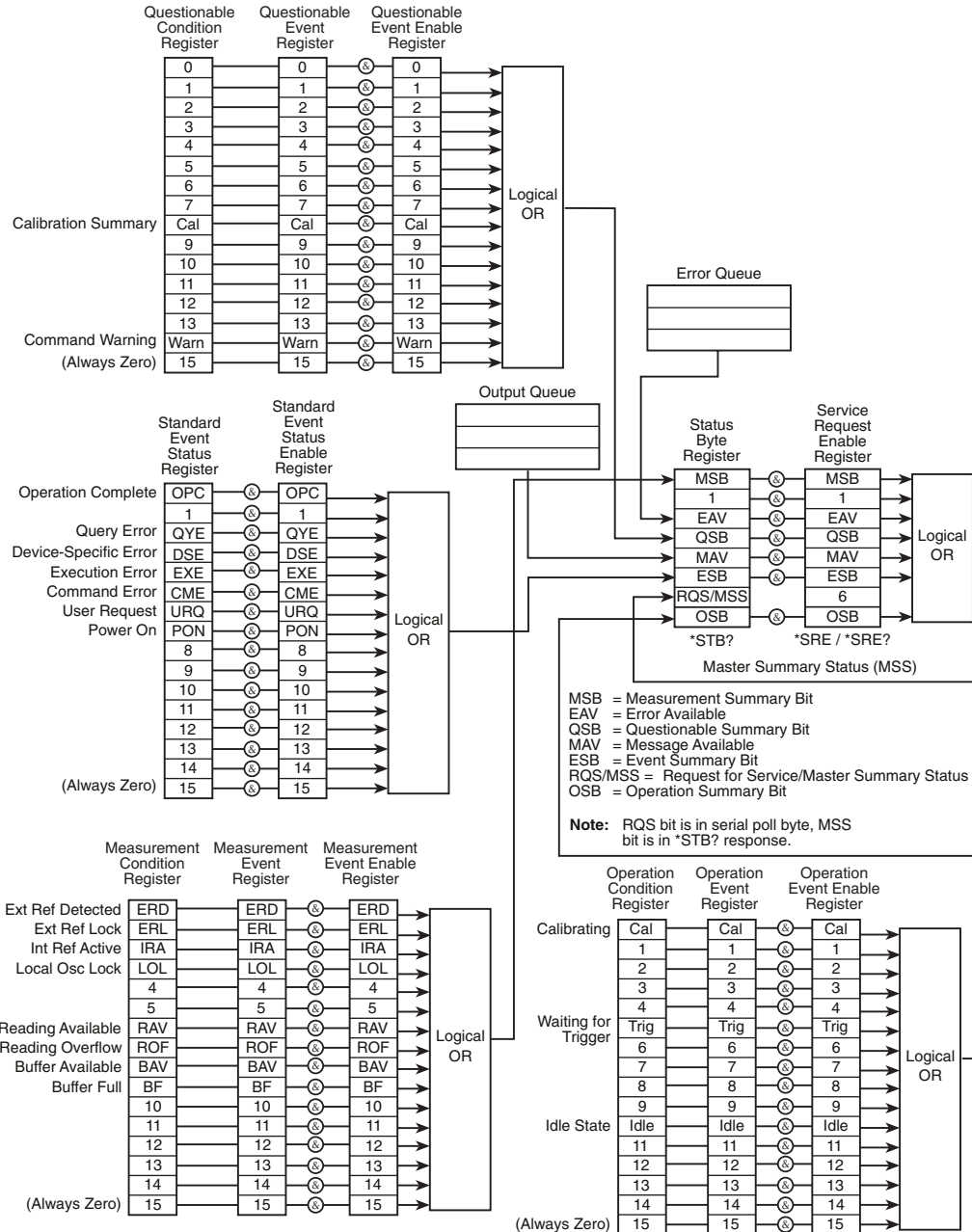
## Operational overview

The Model 2800 provides a series of status registers and queues allowing the operator to monitor and manipulate the various instrument events. The status structure is shown in [Figure 3-6](#). The heart of the status structure is the Status Byte Register. This register can be read by the user's test program to determine if a service request (SRQ) has occurred and what event caused it.

## Status byte and SRQ

The Status Byte Register receives the summary bits of four status register sets and two queues. The register sets and queues monitor the various instrument events. When an enabled event occurs, it sets a summary bit in the Status Byte Register. When a summary bit of the Status Byte is set and its corresponding enable bit is set (as programmed by the user), the RQS/MSS bit will set to indicate that an SRQ has occurred.

**Figure 3-6**  
**Model 2800 status register structure**



## Status register sets

A typical status register set is made up of a condition register, an event register, and an event enable register. A condition register is a read-only register that constantly updates to reflect the present operating conditions of the instrument.

When an event occurs, the appropriate event register bit sets to 1. The bit remains latched to 1 until the register is reset. When an event register bit is set and its corresponding enable bit is set (as programmed by the user), the output (summary) of the register will set to 1, which in turn sets the summary bit of the Status Byte Register.

## Queues

The Model 2800 uses an Output Queue and an Error Queue. The response messages to query commands are placed in the Output Queue. As various programming errors and status messages occur, they are placed in the Error Queue. When a queue contains data, it sets the appropriate summary bit of the Status Byte Register.

**NOTE** *While operating within the trigger model (not in idle), you cannot send a status model command (e.g., \*ESE?). These commands can only be executed when the Model 2800 is in idle.*

## Clearing registers and queues

When the Model 2800 is turned on, the bits of all registers in the status structure are cleared (reset to 0) and the two queues are empty. Commands to reset the event registers, event enable registers, and the Error Queue are listed in [Table 3-2](#). In addition to these commands, any enable register can be reset by sending the 0 parameter value with the individual command to program the register.

**NOTE** *SYSTEM:PRESet and \*RST have no effect on status structure registers and queues.*

Table 3-2

**Common and SCPI commands to reset registers and clear queues**

| Commands                      | Description   | Ref.   |
|-------------------------------|---|--------|
| To Reset Registers:<br>*CLS   | Reset all bits of the following event registers to 0:<br>Standard Event Register<br>Operation Event Register<br>Measurement Event Register<br>Questionable Event Register | Note 1 |
| STATus:PRESet                 | Reset all bits of the following enable registers to 0:<br>Operation Event Enable Register<br>Measurement Event Enable Register<br>Questionable Event Enable Register      | Note 1 |
| To Clear Error Queue:<br>*CLS | Clear all messages from Error Queue   | Note 2 |
| STATus:QUEue:CLEar            | Clear messages from Error Queue   | Note 3 |
| SYSTem:ERRor:CLEar            | Clear messages from Error Queue   | Note 3 |

1. The Standard Event Enable Register is not reset by STATus:PRESet or \*CLS. Send the 0 parameter value with \*ESE to reset all bits of that enable register to 0 (see Status byte and service request commands).
2. STATus:PRESet has no effect on the Error Queue.
3. Use either of the two clear commands to clear the Error Queue.

## Programming and reading registers

### Programming enable registers

The only registers that can be programmed by the user are the enable registers. All other registers in the status structure are read-only registers. The following explains how to ascertain the parameter values for the various commands used to program enable registers. The actual commands are covered later in this section (see [Table 3-4](#) and [Table 3-6](#)).

A command to program an event enable register is sent with a parameter value that determines the desired state (0 or 1) of each bit in the appropriate register. An enable register can be programmed using any of the following data formats for the parameter value: binary, decimal, hexadecimal, or octal.

The bit positions of the register (see [Figure 3-7](#)) indicate the binary parameter value. For example, if you wish to set bits B4, B3 and B1, the binary value would be 11010 (where B4=1, B3=1, B1=1 and all other bits are 0). When you use one of the other formats, convert the binary number to its decimal, hexadecimal, or octal equivalent:

Binary 11010 = Decimal 26 = Hexadecimal 1A = Octal 32

Note that [Figure 3-7](#) includes the decimal weight for each register bit. To set bits B4, B3, and B1, the decimal parameter value would be the sum of the decimal weights for those bits (16+8+2 = 26).

*Figure 3-7*  
**16-bit status register**

*A) Bits 0 through 7*

|              |                   |                   |                   |                   |                   |                   |                   |                   |
|--------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Bit Position | B7                | B6                | B5                | B4                | B3                | B2                | B1                | B0                |
| Binary Value | 0/1               | 0/1               | 0/1               | 0/1               | 0/1               | 0/1               | 0/1               | 0/1               |
| Decimal      | 128               | 64                | 32                | 16                | 8                 | 4                 | 2                 | 1                 |
| Weights      | (2 <sup>7</sup> ) | (2 <sup>6</sup> ) | (2 <sup>5</sup> ) | (2 <sup>4</sup> ) | (2 <sup>3</sup> ) | (2 <sup>2</sup> ) | (2 <sup>1</sup> ) | (2 <sup>0</sup> ) |

*B) Bits 8 through 15*

|              |                    |                    |                    |                    |                    |                    |                   |                   |
|--------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-------------------|-------------------|
| Bit Position | B15                | B14                | B13                | B12                | B11                | B10                | B9                | B8                |
| Binary Value | 0/1                | 0/1                | 0/1                | 0/1                | 0/1                | 0/1                | 0/1               | 0/1               |
| Decimal      | 32768              | 16384              | 8192               | 4096               | 2048               | 1024               | 512               | 256               |
| Weights      | (2 <sup>15</sup> ) | (2 <sup>14</sup> ) | (2 <sup>13</sup> ) | (2 <sup>12</sup> ) | (2 <sup>11</sup> ) | (2 <sup>10</sup> ) | (2 <sup>9</sup> ) | (2 <sup>8</sup> ) |

The <NDN> (non-decimal numeric) parameter type is used to send non-decimal values. These values require a header (#B, #H, or #Q) to identify the data format being sent. The letter in the header can be upper or lower case. The <NRf> (numeric representation format) parameter type is used to send decimal values and does not use a header. The following examples show the proper parameter syntax for setting Bits B5, B3, and B2:

#b101100      Binary format (<NDN> parameter type);  
 #h2C            Hexadecimal format (<NDN> parameter type)  
 #q54            Octal format (<NDN> parameter type)  
 44              Decimal format (<NRf> parameter type)

Valid characters for the non-decimal parameter values are shown as follows:

| <NDN> format | Valid characters            |
|--------------|-----------------------------|
| Binary       | 1s and 0s                   |
| Hexadecimal  | 0 through 9 and A through F |
| Octal        | 0 through 7                 |

## Reading registers

Any register in the status structure can be read by using the appropriate query (?) command. The following explains how to interpret the returned value (response message). The actual query commands are covered later in this section (see [Table 3-4](#), [Table 3-5](#), and [Table 3-6](#)).

The response message will be a value that indicates which bits in the register are set. That value (if not already binary) will have to be converted to its binary equivalent. For example, for a binary value of 100101, bits B5, B2, and B0 are set.

The returned value can be in the binary, decimal, hexadecimal, or octal format. The FORMat:SREGister command is used to select the data format for the returned value (see [Table 3-3](#)).

For non-decimal formats, one of the following headers will accompany the returned value to indicate which format is selected:

#B = Header for binary values

#H = Header for hexadecimal values

#Q = Header for octal values

Table 3-3

### Data format commands for reading status register

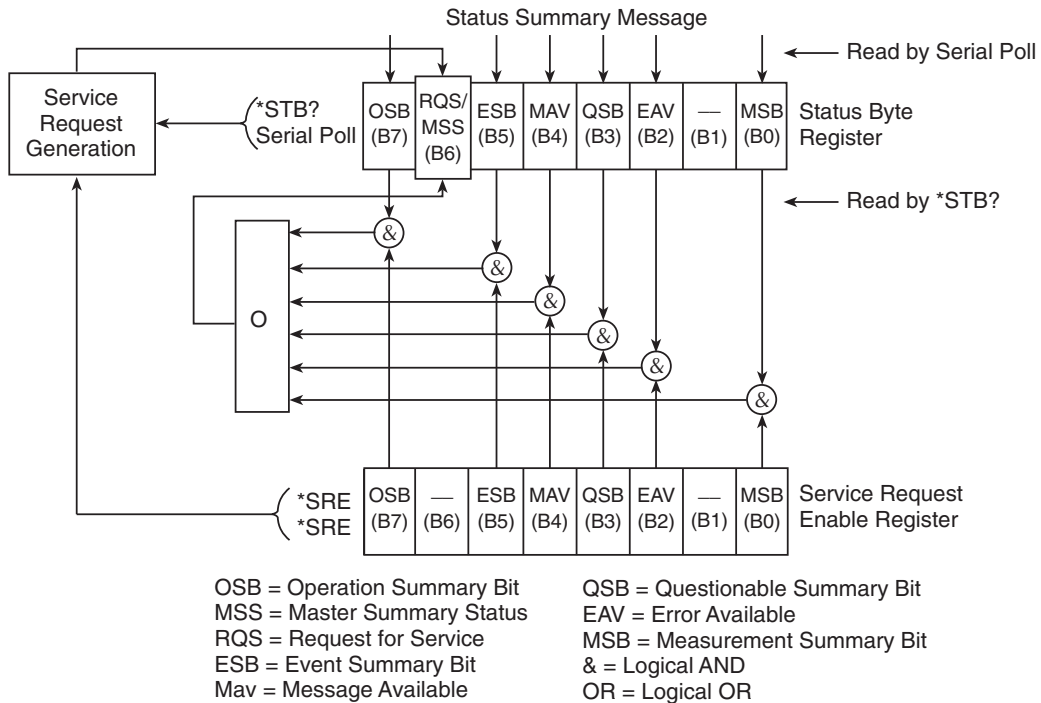
| Command                  | Description  | Default |
|--------------------------|--|---------|
| :FORMat:SREGister <name> | Select data format for reading status registers:<br><name> = ASCii      Decimal format<br>HEXadecimal      Hexadecimal format<br>OCTal              Octal format<br>BINary             Binary format | ASCii   |



## Status byte and service request (SRQ)

Service request is controlled by two 8-bit registers; the Status Byte Register and the Service Request Enable Register. Figure 3-8 shows the structure of these registers.

Figure 3-8  
Status byte and service request (SRQ)



### Status Byte Register

The summary messages from the status registers and queues are used to set or clear the appropriate bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. These summary bits do not latch and their states (0 or 1) are solely dependent on the summary messages (0 or 1). For example, if the Standard Event Register is read, its register will clear. As a result, its summary message will reset to 0, which in turn will reset the ESB bit in the Status Byte Register.

The bits of the Status Byte Register are described as follows:

- **Bit B0, Measurement Summary Bit (MSB)** — Set summary bit indicates that an enabled measurement event has occurred.
- **Bit B1** — Not used.
- **Bit B2, Error Available (EAV)** — Set summary bit indicates that an error or status message is present in the Error Queue.
- **Bit B3, Questionable Summary Bit (QSB)** — Set summary bit indicates that an enabled questionable event has occurred.
- **Bit B4, Message Available (MAV)** — Set summary bit indicates that a response message is present in the Output Queue.
- **Bit B5, Event Summary Bit (ESB)** — Set summary bit indicates that an enabled standard event has occurred.
- **Bit B6, Request Service (RQS)/Master Summary Status (MSS)** — Set bit indicates that an enabled summary bit of the Status Byte Register is set.
- **Bit B7, Operation Summary Bit (OSB)** — Set summary bit indicates that an enabled operation event has occurred.

Depending on how it is used, Bit B6 of the Status Byte Register is either the Request for Service (RQS) bit or the Master Summary Status (MSS) bit:

- When using the serial poll sequence of the Model 2800 to obtain the status byte (a.k.a. serial poll byte), B6 is the RQS bit. See Serial polling and SRQ for details on using the serial poll sequence.
- When using the \*STB? command (see [Table 3-4](#)) to read the status byte, B6 is the MSS bit.

## Service Request Enable Register

The generation of a service request is controlled by the Service Request Enable Register. This register is programmed by you and is used to enable or disable the setting of bit B6 (RQS/MSS) by the Status Summary Message bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. As shown in [Figure 3-8](#), the summary bits are logically ANDed (&) with the corresponding enable bits of the Service Request Enable Register. When a set (1) summary bit is ANDed with an enabled (1) bit of the enable register, the logic “1” output is applied to the input of the OR gate and, therefore, sets the MSS/RQS bit in the Status Byte Register.

The individual bits of the Service Request Enable Register can be set or cleared by using the \*SRE common command. To read the Service Request Enable Register, use the \*SRE? query command. The Service Request Enable Register clears when power is cycled or a parameter value of 0 is sent with the \*SRE command (i.e., \*SRE 0). The commands to program and read the SRQ Enable Register are listed in [Table 3-4](#).

## Serial polling and SRQ

Any enabled event summary bit that goes from 0 to 1 will set bit B6 and generate an SRQ (service request). In your test program, you can periodically read the Status Byte to check if an SRQ has occurred and what caused it. If an SRQ occurs, the program can, for example, branch to an appropriate subroutine that will service the request.

Typically, SRQs are managed by the serial poll sequence of the Model 2800. If an SRQ does not occur, bit B6 (RQS) of the Status Byte Register will remain cleared and the program will simply proceed normally after the serial poll is performed. If an SRQ does occur, bit B6 of the Status Byte Register will set and the program can branch to a service subroutine when the SRQ is detected by the serial poll.

The serial poll automatically resets RQS of the Status Byte Register. This allows subsequent serial polls to monitor bit B6 for an SRQ occurrence generated by other event types. After a serial poll, the same event can cause another SRQ, even if the event register that caused the first SRQ has not been cleared.

The serial poll does not clear MSS. The MSS bit stays set until all Status Byte summary bits are reset.

## SPE, SPD (serial polling)

The SPE, SPD General Bus Command sequence is used to serial poll the Model 2800. Serial polling obtains the serial poll byte (status byte). Typically, serial polling is used by the controller to determine which of several instruments has requested service with the SRQ line.

## Status byte and service request commands

The commands to program and read the Status Byte Register and Service Request Enable Register are listed in [Table 3-4](#). For details on programming and reading registers, see “[Programming enable registers](#)” and “[Reading registers](#)”.

**NOTE**      *To reset the bits of the Service Request Enable Register to 0, use 0 as the parameter value for the \*SRE command (i.e. \*SRE 0).*

Table 3-4

**Status Byte and Service Request Enable Register commands**

| Command             | Description  | Default |
|---------------------|--|---------|
| *STB?               | Read Status Byte Register.   | (Note)  |
| *SRE <NDN> or <NRf> | Program the Service Request Enable Register:<br><NDN> = #Bxx...x Binary format (each x = 1 or 0)<br>= #Hx Hexadecimal format (x = 0 to FF)<br>= #Qx Octal format (x = 0 to 377)<br><NRf> = 0 to 255 Decimal format |         |
| *SRE?               | Read the Service Request Enable Register   |         |

Note: \*CLS and STATus:PRESet have no effect on the Service Request Enable Register.

**Programming example - set MSS (B6) when error occurs**

The second command in the following sequence enables EAV (error available). When an invalid command is sent (line 4), bits B2 (EAV) and B6 (MSS) of the Status Byte Register set to 1. The last command reads the Status Byte Register using the binary format (which directly indicates which bits are set).

The command to select format (FORMat:SREGister) is documented in [Table 3-3](#). To determine the exact nature of the error, you will have to read the Error Queue (see [“Queues”](#)).

```
*CLS           | Clear Error Queue.
*SRE 4         | Enable EAV.
FORM:SREG BIN | Select binary format.
*XYZ           | Generate error.
*STB?         | Read Status Byte Register.
```

**Serial poll programming example**

This example is written specifically for the KPCI-488.2 GPIB card and QuickBasic/VisualBasic with the appropriate IEEE libraries. Other types of cards and/or languages may have different function calls that are equivalent to the initialize(), transmit(), send(), srq, and spoll() calls used below.

## SRQ when buffer fills with 2000 readings

The following program will store 2000 readings in the buffer. When the buffer fills, a message will be displayed on the computer to indicate that the buffer is full.

```
' $INCLUDE: 'ieeeqb.bi'

CLS                               ' Clear PC output screen.
CONST addr = 28                   ' Set instrument address.
'
' Init GPIB.
'
CALL initialize(21, 0)
CALL transmit("unt unl listen " + STR$(addr) + " sdc unl", status%) ' Send Device Clear.

CALL send(addr, "*rst", status%)  ' Restore *rst defaults.
CALL send(addr, "trac:cle", status%) ' Clear buffer.
CALL send(addr, "trig:coun inf", status%) ' Infinite trigger count.
CALL send(addr, "trac:poin 2000", status%) ' Set buffer size to 2000.
CALL send(addr, "trac:feed:cont next", status%) ' Enable buffer.
CALL send(addr, "stat:pres", status%) ' Reset measure enable bits.
CALL send(addr, "*cls", status%)  ' Clear all event registers.
CALL send(addr, "stat:meas:enab 512", status%) ' Clear Enable buffer
                                           ' bit: B9.

CALL send(addr, "*ese 0", status%)  ' Disable standard events.
CALL send(addr, "*sre 1", status%)  ' Enable measurement events.
CALL send(addr, "init", status%)    ' Start measure/store process.

WaitSRQ:
  WHILE srq = 0: WEND              ' Wait for GPIB SRQ line to go true.
  CALL spoll(addr, poll%, status%) ' Clear rqs/mss bit in status byte
                                   ' register.
  CALL send(addr, "*cls", status%) ' Clear all event registers.
  PRINT "BUFFER FULL"              ' Display buffer full message.
END
```

## Status register sets

As shown in [Figure 3-6](#), there are four status register sets in the status structure of the Model 2800: Standard Event Status, Operation Event Status, Measurement Event Status, and Questionable Event Status.

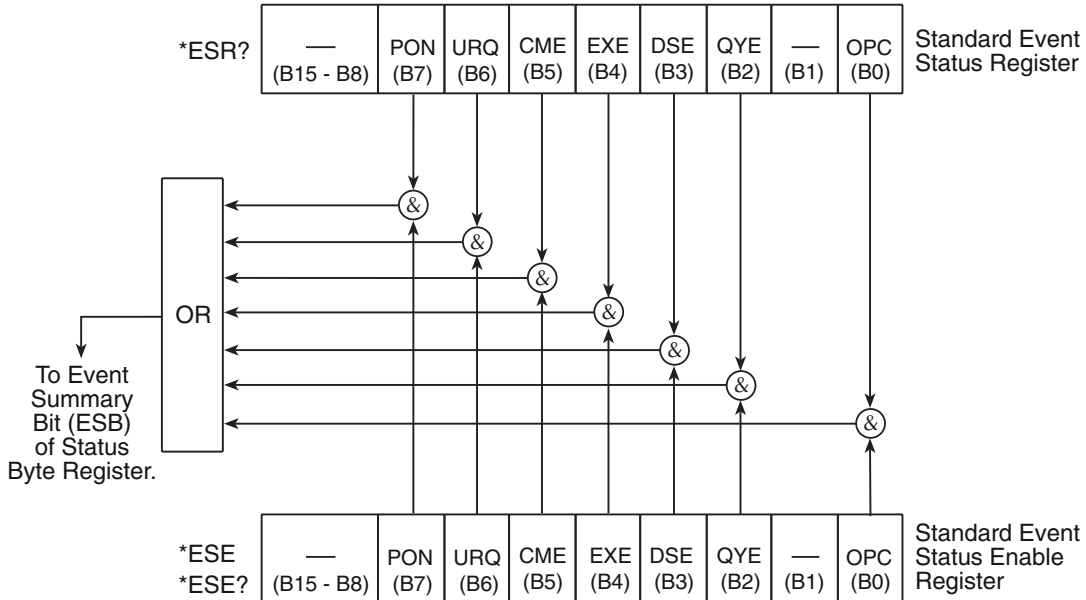
## Register bit descriptions

### Standard event register

The used bits of the Standard Event Register (Figure 3-9) are described as follows:

- **Bit B0, Operation Complete (OPC)** — Set bit indicates that all pending selected device operations are completed and the Model 2800 is ready to accept new commands. This bit only sets in response to the \*OPC? query command. See Section 2 (“Common commands”) for details on \*OPC and \*OPC?.
- **Bit B1** — Not used.
- **Bit B2, Query Error (QYE)** — Set bit indicates that you attempted to read data from an empty Output Queue.

Figure 3-9  
Standard event status



PON = Power On  
 URQ = User Request  
 CME = Command Error  
 EXE = Execution Error  
 DSE = Device-Specific Error  
 QYE = Query Error  
 OPC = Operation Complete  
 & = Logical AND  
 OR = Logical OR

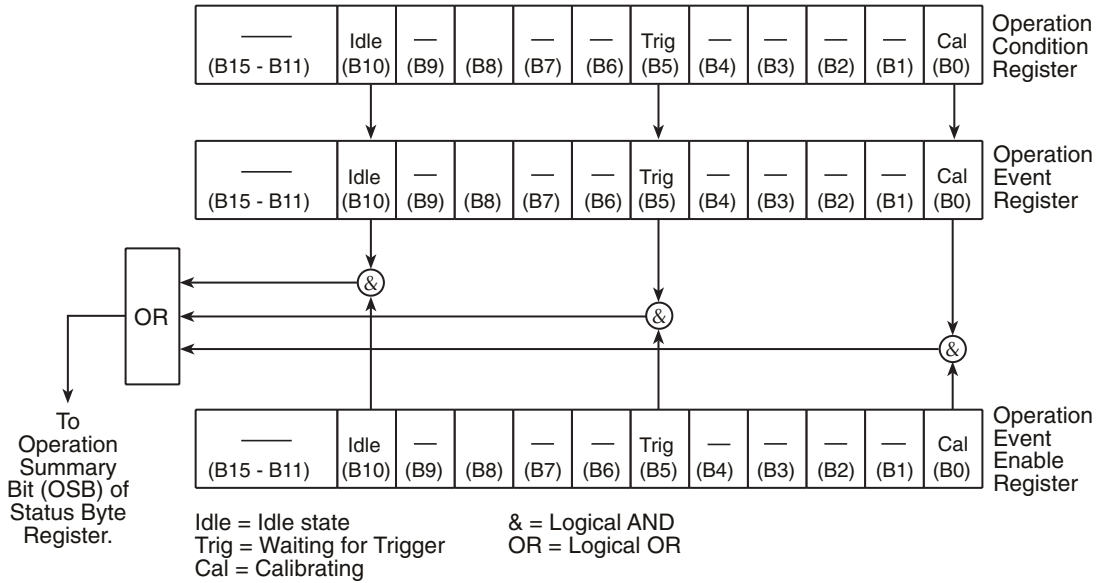
- **Bit B3, Device-Specific Error (DSE)** — Set bit indicates that an instrument operation did not execute properly due to some internal condition.
- **Bit B4, Execution Error (EXE)** — Set bit indicates that the Model 2800 detected an error while trying to execute a command. Places error message -200 in the error queue.
- **Bit B5, Command Error (CME)** — Set bit indicates that a command error has occurred. Places error message -100 in the error queue. Command errors include:
  - **IEEE-488.2 syntax error** – Model 2800 received a message that does not follow the defined syntax of the IEEE-488.2 standard.
  - **Semantic error** — Model 2800 received a command that was misspelled or received an optional IEEE-488.2 command that is not implemented.
  - The instrument received a Group Execute Trigger (GET) inside a program message.
- **Bit B6, User Request (URQ)** — Set bit indicates that the LOCAL key on the Model 2800 front panel was pressed.
- **Bit B7, Power On (PON)** — Set bit indicates that the Model 2800 has been turned off and turned back on since the last time this register has been read.

### Operation event register

The bits of the Operation Event Register ([Figure 3-10](#)) are described as follows:

- **Bits B0, Calibrating (Cal)** — Set bit indicates that the Model 2800 is calibrating.
- **Bits B1 through B4** — Not used.
- **Bit B5, Waiting for Trigger (Trig)** — Set bit indicates that the Model 2800 is in the trigger layer waiting for a trigger event to occur.
- **Bits B6 through B9** — Not used.
- **Bit B10, Idle State (Idle)** — Set bit indicates the Model 2800 is in the idle state.
- **Bits B11 through B15** — Not used.

Figure 3-10  
**Operation event status**



Section topics

Section topics

Section topics

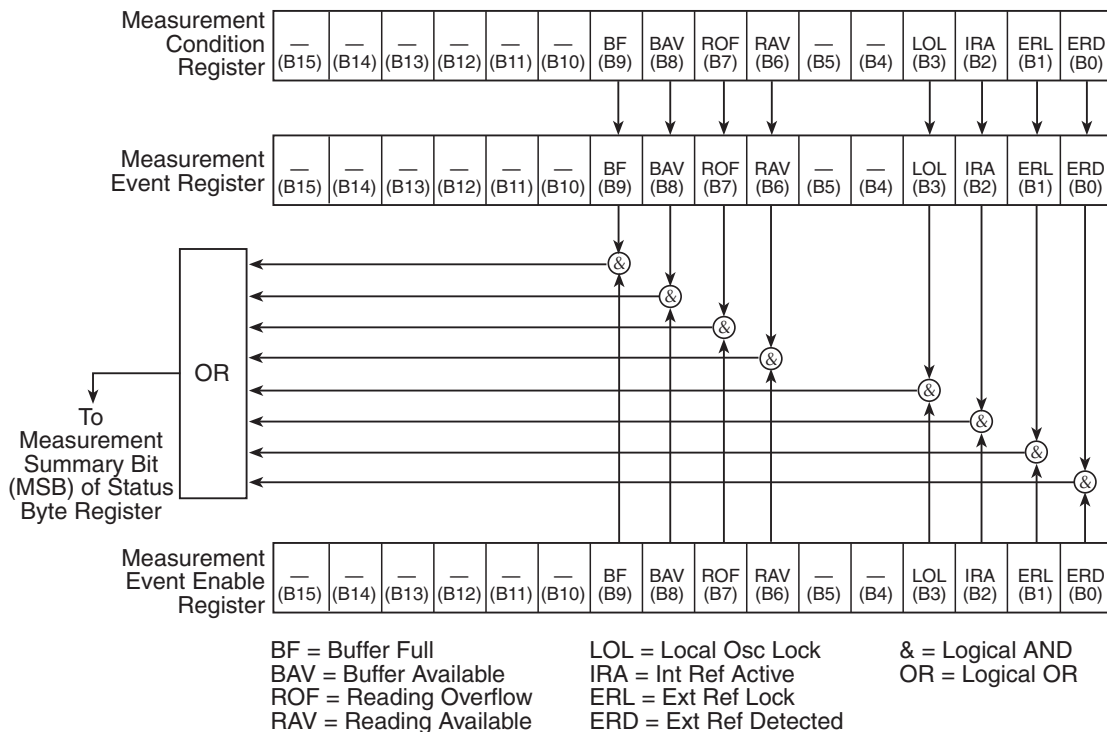


## Measurement event register

The used bits of the Measurement Event Register (Figure 3-11) are described as follows:

- **Bit B0, External Reference Detected (ERD)** — Set bit indicates that an external reference is connected and has been detected. Places status message +115 in the status queue. See next bit to determine if the external reference is locked in.
- **Bit B1, External Reference Lock (ERL)** — Set bit indicates that the external reference has been locked in by the Model 2800. Places status message +116 in the status queue.
- **Bit B2, Internal Reference Active (IRA)** — Set bit indicates that an external reference is NOT connected and the internal reference is being used (active). Places status message +117 in the status queue.
- **Bit B3, Local Oscillator Locked (LOL)** — Set bit indicates that the local oscillator is settled and locked-in (ready for measurements). Places status message +118 in the status queue.
- **Bits B4 and B5** — Not used.
- **Bit B6, Reading Available (RAV)** — Set bit indicates that a reading was taken and processed. Places status message +106 in the status queue.
- **Bit B7, Reading Overflow (ROF)** — Set bit indicates that the reading exceeds the measurement range of the instrument. Places status message +107 in the status queue.
- **Bit B8, Buffer Available (BAV)** — Set bit indicates that there are at least two readings in the buffer. Places status message +108 in the status queue.
- **Bit B9, Buffer Full (BF)** — Set bit indicates that the trace buffer is full. Places status message +109 in the status queue.
- **Bits B10 through B15** — Not used.

Figure 3-11  
Measurement event status



Section topics

Section topics

Section topics

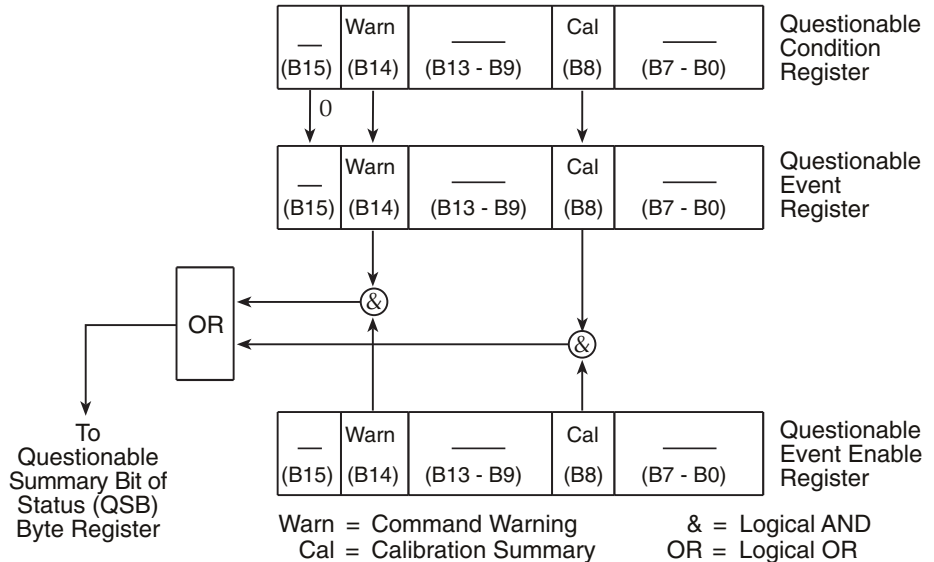
### Questionable event register

The used bits of the Questionable Event Register (Figure 3-12) are described as follows:

- **Bits B0 through B7** — Not used.
- **Bit B8, Calibration Summary (Cal)** — Set bit indicates that an invalid calibration constant was detected during the power-up sequence. The instrument will instead use a default calibration constant. This error will clear after successful calibration of the instrument.
- **Bits B9 through B13** — Not used.
- **Bit B14, Command Warning (Warn)** — Set bit indicates that a Signal Oriented Measurement Command parameter has been ignored. Places status message +414 in the status queue.
- **Bit B15** — Not used.

**NOTE** Whenever a questionable event occurs, the ERR annunciator will turn on. The annunciator will turn off when the questionable event clears.

Figure 3-12  
**Questionable event status**



## Condition registers

As [Figure 3-6](#) shows, each status register set (except the Standard Event Register set) has a condition register. A condition register is a real-time, read-only register that constantly updates to reflect the present operating conditions of the instrument. For example, while the Model 2800 is in the idle state, bit B10 (Idle) of the Operation Condition Register will be set. When the instrument is taken out of idle, bit B10 clears.

The commands to read the condition registers are listed in [Table 3-5](#). For details on reading registers, see “[Reading registers](#)”.

Table 3-5

**Condition register commands**

| Command                        | Description                           |
|--------------------------------|---------------------------------------|
| STATus:OPERation:CONDition?    | Read Operation Condition Register.    |
| STATus:MEASurement:CONDition?  | Read Measurement Condition Register.  |
| STATus:QUESTionable:CONDition? | Read Questionable Condition Register. |

## Event registers

As [Figure 3-6](#) shows, each status register set has an event register. When an event occurs, the appropriate event register bit sets to 1. The bit remains latched to 1 until the register is reset. Reading an event register clears the bits of that register. \*CLS resets all four event registers.

The commands to read the event registers are listed in [Table 3-6](#). For details on reading registers, see “[Reading registers](#)”.

Table 3-6

**Event register commands**

| Command                       | Description                          |
|-------------------------------|--------------------------------------|
| *ESR?                         | Read Standard Event Status Register. |
| STATus:OPERation:[:EVENT]?    | Read Operation Event Register.       |
| STATus:MEASurement:[:EVENT]?  | Read Measurement Event Register.     |
| STATus:QUESTionable:[:EVENT]? | Read Questionable Event Register.    |

Note: Power-up and \*CLS resets all bits of all event registers to 0. STATus:PRESet has no effect.

### Event enable registers

As Figure 3-6 shows, each status register set has an enable register. Each event register bit is logically ANDed (&) to a corresponding enable bit of an enable register. Therefore, when an event bit is set and the corresponding enable bit is set (as programmed by the user), the output (summary) of the register will set to 1, which in turn sets the summary bit of the Status Byte Register.

The commands to program and read the event enable registers are listed in Table 3-7. For details on programming and reading registers, see “Programming enable registers” and “Reading registers”.

**NOTE** *The bits of any enable register can be reset to 0 by sending the 0 parameter value with the appropriate enable command (i.e. STATus:OPERation:ENABle 0).*

Table 3-7  
**Event enable registers commands**

| Command                | Description   | Default |
|------------------------|---|---------|
| *ESE <NDN> or <NRf>    | Program Standard Event Enable Register. (See “Parameters”.) | (Note)  |
| *ESE?                  | Read Standard Event Enable Register.                        |         |
| STATus                 | STATus Subsystem:   |         |
| :OPERation             | Operation Event Enable Register:                            |         |
| :ENABle <NDN> or <NRf> | Program enable register. (See “Parameters”.)                |         |
| :ENABle?               | Read enable register.                                       |         |
| :MEASurement           | Measurement Event Enable Register:                          |         |
| :ENABle <NDN> or <NRf> | Program enable register. (See “Parameters”.)                |         |
| :ENABle?               | Read enable register.                                       |         |
| :QUESTionable          | Questionable Event Enable Register:                         |         |
| :ENABle <NDN> or <NRf> | Program enable register. (See “Parameters”.)                |         |
| :ENABle?               | Read Measurement Event Enable Register:                     |         |

Parameters:

- <NDN>= #Bxx...x    Binary format (each x = 1 or 0)
- = #Hx        Hexadecimal format (x = 0 to 7FFF)
- = #Qx        Octal format (x = 0 to 77777)
- <NRf>= 0 to 32767    Decimal format

Note: Power-up and STATus:PRESet resets all bits of all enable registers to 0. \*CLS has no effect.

### Programming example — program and read register set

The following command sequence programs and reads the measurement register set. Registers are read using the binary format (which directly indicates which bits are set). The command to select format (FORMat:SREGister) is listed in [Table 3-3](#).

```
FORM:SREG BIN          ' Select binary format to read registers.
STAT:MEAS:ENAB 512     ' Enable BFL (buffer full).
STAT:MEAS:COND?       ' Read Measurement Condition Register.
STAT:MEAS?            ' Read Measurement Event Register.
```

## Queues

The Model 2800 uses two queues, which are first-in first-out (FIFO) registers:

- Output Queue — Used to hold reading and response messages.
- Error Queue — Used to hold error and status messages.

The Model 2800 status model ([Figure 3-6](#)) shows how the two queues are structured with the other registers.

### Output queue

The Output Queue holds data that pertains to the normal operation of the instrument. For example, when a query command is sent, the response message is placed in the Output Queue.

When data is placed in the Output Queue, the Message Available (MAV) bit in the Status Byte Register sets. A data message is cleared from the Output Queue when it is read. The Output Queue is considered cleared when it is empty. An empty Output Queue clears the MAV bit in the Status Byte Register.

A message is read from the Output Queue by addressing the Model 2800 to talk after the appropriate query is sent.

### Error queue

The Error Queue holds error and status messages. When an error or status event occurs, a message that defines the error/status is placed in the Error Queue.

When a message is placed in the Error Queue, the Error Available (EAV) bit in the Status Byte Register is set. An error/status message is cleared from the Error Queue when it is read. The Error Queue is considered cleared when it is empty. An empty Error Queue clears the EAV bit in the Status Byte Register.

The Error Queue holds up to 10 error/status messages. The commands to read the Error Queue are listed in [Table 3-8](#). When you read a message in the Error Queue, the “oldest” message is read and then removed from the queue. If the queue becomes full, the message “350, ‘Queue Overflow’” will occupy the last memory location. On power-up, the Error Queue is empty. When empty, the message “0, No Error” is placed in the queue.

Messages in the Error Queue are preceded by a code number. Negative (-) numbers are used for SCPI-defined messages and positive (+) numbers are used for Keithley-defined messages. The messages are listed in [Appendix B](#).

On power-up, all error messages are enabled and will go into the Error Queue as they occur. Status messages are not enabled and will not go into the queue. As listed in [Table 3-8](#), there are commands to enable and/or disable messages. For these commands, the <list> parameter is used to specify which messages to enable or disable. The messages are specified by their codes. The following examples show various forms for using the <list> parameter.

|        |                     |   |
|--------|---------------------|---|
| <list> | = (-110)            | Single message                                      |
|        | = (-110:-222)       | Range of messages (-110 through -222)               |
|        | = (-110:-222, -220) | Range entry and single entry (separated by a comma) |

When you enable messages, messages not specified in the list are disabled. When you disable messages, each listed message is removed from the enabled list.

**NOTE**      *To prevent all messages from entering the Error Queue, send the null list parameter as follows: STATus:QUEue:ENABle ()*

Table 3-8  
**Error queue commands**

| Command  | Description  | Default                                  |
|--|--|--|
| STATus<br>:QUEue<br>[:NEXT]?<br>:ENABle <list><br>:ENABle?<br>:DISABle <list><br>:DISABle?<br>:CLEar | STATus Subsystem:<br>Read Error Queue:<br>Read and clear oldest error/status (code and message).<br>Specify error and status messages for Error Queue.<br>Read the enabled messages.<br>Specify messages not to be placed in queue.<br>Read the disabled messages.<br>Clear messages from Error Queue.   | (Note 1)<br><br>(Note 2)<br><br>(Note 2) |
| SYSTem<br>:ERRor<br>[:NEXT]?<br>:ALL?<br>:COUNT?<br>:CODE<br>[:NEXT]?<br>:ALL?<br>:CLEar             | SYSTem Subsystem:<br>Read Error Queue:<br>Read and clear oldest error/status (code and message).<br>Read and clear all errors/status (code and message).<br>Read the number of messages in queue.<br>Code numbers only:<br>Read and clear oldest error/status (code only).<br>Read and clear all errors/status (codes only).<br>Clear messages from Error Queue. | (Note 1)                                 |

## Notes:

1. Power-up and \*CLS empties the Error Queue. STATus:PRESet has no effect.
2. Power-up enables error messages and disables status messages. \*CLS and STATus:PRESet have no effect.

### Programming example — read error queue

The following command reads the error queue:

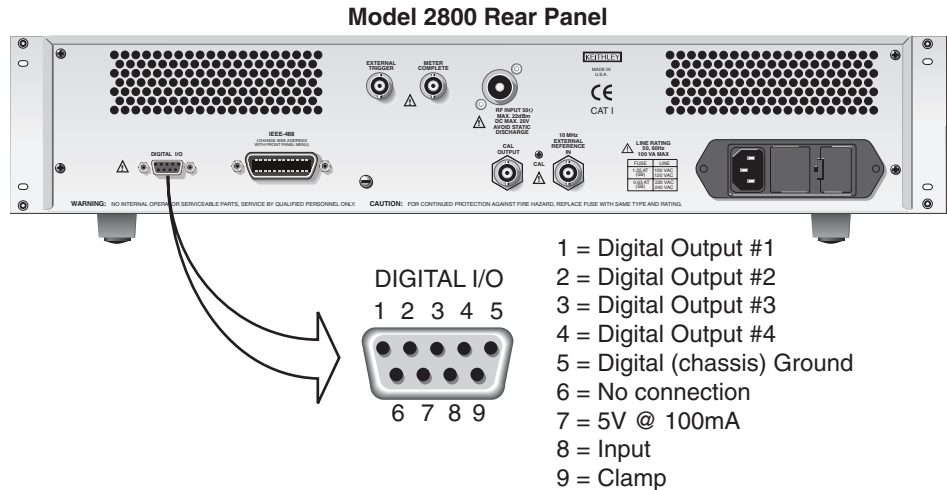
```
STAT:QUE?
```



# Digital I/O

Model 2800's Digital I/O port is accessed at a male DB-9 connector located on the rear panel. The connector location and pin designations are shown in [Figure 3-13](#).

Figure 3-13  
Digital I/O port



## Digital input

Pin 8 of the Digital I/O is the digital input line. A TTL high or TTL low can be applied to this line. The state of the input (high or low) can be determined from the front panel or over the bus.

### Front panel operation

1. Press **Menu** to access the main menu.
2. Use the **NEXT** or **PREV** key to place the blinking cursor on **GENERAL** and press **Enter**.
3. Place the cursor on **DIGITAL** and press **Enter**.
4. Place the cursor on **INPUT** and press **Enter** to display the state (**HI** or **LO**) of the digital input.
5. Use the **Exit** key to back out of the menu structure.

## Remote programming

Use the following query to read the state of the digital input:

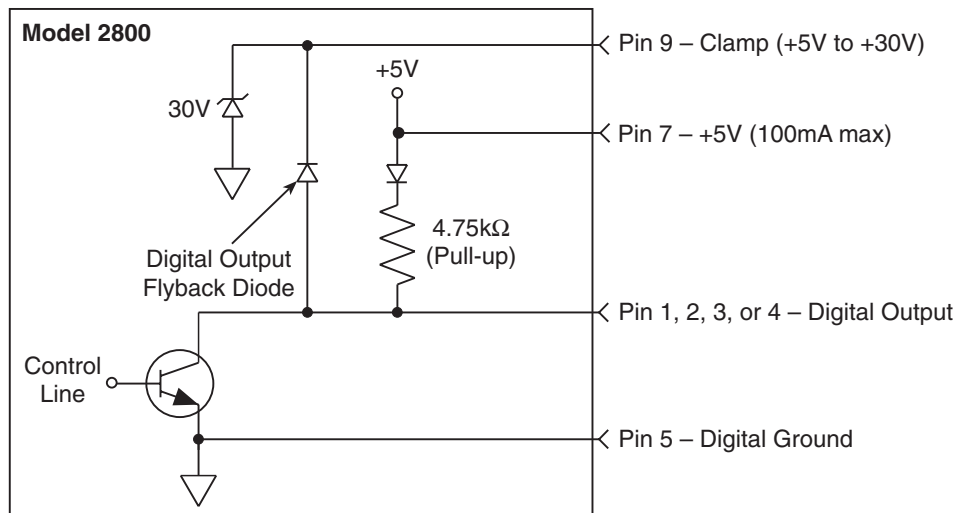
```
SOURce2:TTL:INPut?
```

If a high TTL level is on the input, this command will return a “1”. If a low TTL level is present, a “0” will be returned.

## Digital outputs

The digital I/O port has four digital outputs. Each digital output can be used as a sink to control devices (e.g., relays) or as a source to provide input to external logic (TTL or CMOS) circuitry. The simplified schematic for the digital outputs are shown in Figure 3-14. Note that this illustration shows the schematic for one digital output. All four digital output circuits are identical.

Figure 3-14  
**Digital output port simplified schematic**



## Sink mode — controlling external devices

Each output can be operated from an external supply (voltage range from +5V to +30V applied through the external device being driven). The high current sink capacity of the output driver allows direct control of relays, solenoids, and lamps (no additional circuitry needed).

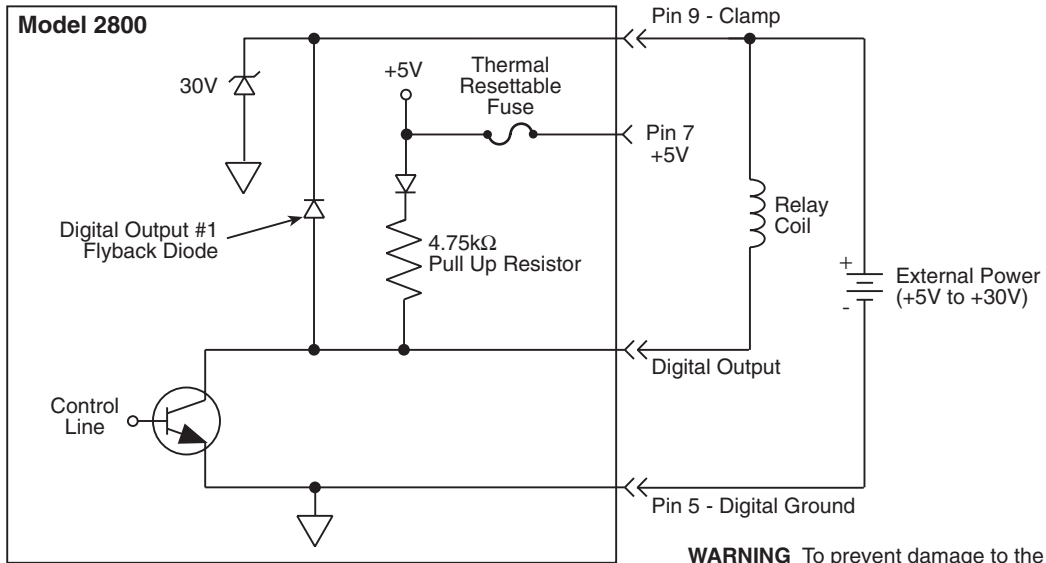
As shown in [Figure 3-14](#), each of the digital, open-collector outputs includes a built-in pull up resistor connected to +5V. The output transistor is capable of sinking 250mA from voltages up to +30V. Each output channel contains a fly-back diode for protection when switching inductive loads (such as a low power solenoid or relay coils). To use these fly-back diodes, connect the external supply voltage to pin 9 of the digital I/O port. Make sure the external supply voltage is between +5V and +30V and the current required by the device does not exceed 250mA.

**CAUTION** On pin 9, do not exceed +30V. For the output lines, do not exceed the maximum sink current. The maximum sink current for an output line is 250mA. Exceeding these limits may cause damage to the instrument that is not covered by the warranty.

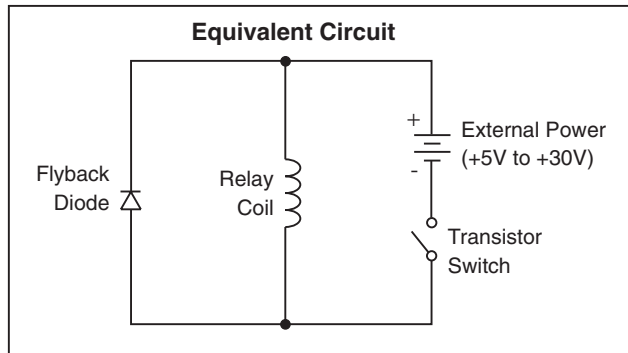
An externally powered relay connected to the digital output port is shown in [Figure 3-15](#). Other externally powered devices can be similarly connected by replacing the relay with the device. When the output line is pulled low (0V), the output transistor sinks current through the external device. In the high state, the output transistor is off (transistor switch open). This interrupts current flow through the external device.

**Internal +5V supply** – The +5V supply of the Digital I/O (pin 7) can be used to drive low power devices. The maximum current that can be delivered to a DUT load is 100mA. Do not exceed this limit. In [Figure 3-15](#), the internal 5V supply can be used instead of the External Power source by connecting the high side of the coil to pin 7 of the Digital I/O. Make sure both the coil and External Power source are disconnected from pin 9.

Figure 3-15  
Controlling externally powered relays



**WARNING** To prevent damage to the Model 2800, do not exceed +30V and make sure the clamp is properly connected, as shown.



## Source mode (logic control)

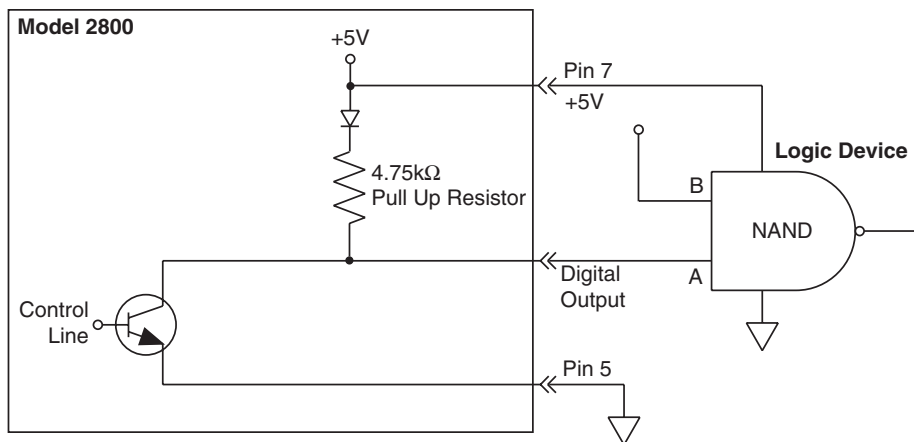
The digital outputs can be used as logic inputs to active TTL, low-power TTL, or CMOS inputs. For this mode of operation, the output lines can source up to 200 $\mu$ A.

**CAUTION** Each output line can source up to 200 $\mu$ A. Exceeding 200 $\mu$ A may cause damage to Model 2800 that is not covered by the warranty.

Figure 3-16 shows how to connect a logic device to one of the output lines. When the output line is pulled high, the transistor will turn off (transistor switch open) to provide a reliable logic high output (>3.75V). When the output line goes low, the transistor turns on (transistor switch closed) to route current to digital ground. As a result, a low logic output (0V) is provided at the output.

If the second input (B) of the NAND gate is connected to another output line of the port, the output of the NAND gate will go to logic 0 when both digital outputs are set high.

Figure 3-16  
NAND gate control



## Setting digital output lines

### Front panel operation

An output line can be set high (H) or low (L). From the front panel, the decimal equivalent of the bit pattern is set. Use [Table 3-9](#) to determine the decimal value for the bit pattern. For example, to set output 4 high and the outputs 1, 2, and 3 low, use decimal value “8” as shown in the [Table 3-9](#).

1. Press **Menu** to access the main menu.
2. Use the **NEXT** or **PREV** key to place the blinking cursor on **GENERAL** and press **Enter**.
3. Place the cursor on **DIGITAL** and press **Enter**.
4. Place the cursor on **OUTPUT** and press **Enter**.
5. Key in the desired digital output pattern (in decimal) and press **Enter**.
6. Use the **Exit** key to back out of the menu structure.

*Table 3-9*

**Output patterns for NRf parameter**

| Out 4 | Out 3 | Out 2 | Out 1 | Decimal |
|-------|-------|-------|-------|---------|
| L     | L     | L     | L     | 0       |
| L     | L     | L     | H     | 1       |
| L     | L     | H     | L     | 2       |
| L     | L     | H     | H     | 3       |
| L     | H     | L     | L     | 4       |
| L     | H     | L     | H     | 5       |
| L     | H     | H     | L     | 6       |
| L     | H     | H     | H     | 7       |
| H     | L     | L     | L     | 8       |
| H     | L     | L     | H     | 9       |
| H     | L     | H     | L     | 10      |
| H     | L     | H     | H     | 11      |
| H     | H     | L     | L     | 12      |
| H     | H     | L     | H     | 13      |
| H     | H     | H     | L     | 14      |
| H     | H     | H     | H     | 15      |

## Remote programming

The SOURce2 subsystem commands are used to set the logic level for the four output lines of the Digital I/O. These commands are listed in [Table 2-6](#) in [Section 2](#).

An output line can be set high (H) or low (L). When using the decimal format to set the bit pattern, use [Table 3-9](#) to determine the decimal value for the parameter. For example, to set output 4 high and the outputs 1, 2, and 3 low, use parameter value “8” as shown in the [Table 3-9](#).

When using one of the non-decimal formats, convert the decimal value for the bit pattern to the equivalent binary, octal or hexadecimal value. Be sure to use the appropriate header (#B, #Q, or #H) with the non-decimal number. For example, for decimal 8 the binary equivalent for the parameter is #B1000.

When using the query command to read the set bit pattern, the format for the returned value depends on the presently set data format for SOURce2 commands. The FORMat:SOURce2 command is used to select the data format. Details to select “[Data format for SOURce2 commands](#)” are provided in [Section 2](#).

**Example** – Set outputs 4 and 2 high and outputs 3 and 1 low:

```
SOUR2:TTL:LEV #B1010 ' Binary format.
```

*Table 3-10*  
**SOURce2 command summary**

| Command                 | Description   | Default |
|-------------------------|---|---------|
| SOURce2                 |   |         |
| :TTL                    |   |         |
| [:LEVel] <NDN> or <NRf> | Specify bit pattern for the four digital output lines | 15      |
| [:LEVel]?               | Query set bit pattern.                                |         |
| :INPut?                 | Query digital input state: Hi (1) or Lo (0).          |         |

**Parameters:**

- <NRf> = 0 to 15      **Decimal format**
- <NDN> = #Bxxx      **Binary format**      xxx = 0000 to 1111
- = #Hx        **Hexadecimal format**    x = 0 to F
- = #Qx        **Octal format**            x = 0 to 17

# 4

# Functionality Verification

---

## Section topics

### Introduction

### Test requirements

- Environmental conditions

- Warm-up period

- Line power

### Recommended test equipment

### Calculating test limits

- Example reading limit calculation

### Restoring factory defaults

### Test summary and considerations

- Functionality test summary

- Test considerations

### Functionality tests

- VSWR measurements

- Noise floor

- Reference standard frequency accuracy

- Primary channel power measurement accuracy

- Adjacent channel power measurement accuracy

- Alternate channel power measurement accuracy



# Introduction

**NOTE** *In order to verify that the Model 2800 is within the limits stated in the instrument's one-year accuracy specifications, Keithley verifies approximately 15,000 test points during routine calibration using a specially designed automated platform. It is recommended that the instrument be returned to Keithley annually for performance verification and calibration.*

**WARNING** **The information in this section is intended only for qualified service personnel. Do not attempt these procedures unless you are qualified to do so.**

Information is provided in this section to offer recommendations to the owner of the Model 2800 to perform test procedures to the extent that is required by their quality system in the event that they have elected not to return the instrument to Keithley for routine verification/calibration. Any verification procedure for this instrument needs to account for measurement uncertainties that arise from connector and component mismatch.

Before proceeding, the laboratory should characterize the components used with the testing process in order to correct for mismatch and to estimate measurement uncertainty appropriately.

**NOTE** *The test procedures in this section provide a relatively low-cost method of verifying instrument performance. The techniques used for these tests should be adequate for most users. However, measurement uncertainty may not be sufficient for those with more stringent requirements. If necessary, contact your Keithley representative or the factory for information on more comprehensive procedures to verify Model 2800 performance.*

# Test requirements

Be sure that you perform the functionality tests:

- Under the proper environmental conditions.
- After the specified warm-up period.
- Using the correct line voltage.
- Using the proper test equipment.
- Using the specified reading limits.

## Environmental conditions

Conduct the procedures in a test environment that has:

- An ambient temperature of 18° to 28°C (65° to 82°F).
- A relative humidity of less than 70% unless otherwise noted.

## Warm-up period

Allow the Model 2800 to warm up for at least one hour before conducting the procedures.

If the instrument has been subjected to temperature extremes (those outside the ranges stated above), allow additional time for the instrument's internal temperature to stabilize. Allow one extra hour to stabilize a unit that is 10°C (18°F) outside the specified temperature range.

Also, allow the test equipment to warm up for the minimum time specified by the manufacturer.

## Line power

The Model 2800 requires a line voltage of 100V to 120V or 220V to 240V and a line frequency of 50Hz or 60Hz. Note that the line voltage must be manually set to either 100V/120V or 220V/240V. For details, see "Power-up" in Section 1 of the User's Manual.

## Recommended test equipment

Table 4-1 summarizes recommended test equipment.

Table 4-1

### Recommended test equipment

| Description  | Manufacturer/model<br>(key specifications*)                 | Test(s)   |
|--|---|---|
| Power Meter/<br>CW Power Sensor                                  | Agilent E4418B/4412A  | Primary channel power accuracy<br>Adjacent channel power accuracy<br>Alternate channel power accuracy |
| Signal Generator   | Rohde and Schwarz SML 03                                    | Primary channel power accuracy<br>Adjacent channel power accuracy<br>Alternate channel power accuracy |
| Power Splitter   | Agilent 11667A  | Primary channel power accuracy<br>Adjacent channel power accuracy<br>Alternate channel power accuracy |
| Type N 50Ω Terminator  | Agilent 909A OPT 12<br>(dc to 2GHz)                         | Power splitter correction   |
| Frequency Counter  | Agilent 53131   | Reference frequency accuracy  |
| S-Parameter Vector<br>Network Analyzer<br>Type N Calibration Kit | Agilent 8714ES<br>Agilent 85032B<br>(Open, short, 50Ω load) | VSWR  |
| Torque Wrench for<br>Type N Connectors                           | 12in-lb. maximum torque                                     | All except reference frequency<br>accuracy.   |

\* Frequency specifications show only the range necessary for Model 2800 measurements. Accuracy specifications shown are for recommended test equipment across that frequency range where applicable.

## Calculating test limits

The test limits stated in this section have been calculated using only the Model 2800 one-year accuracy specifications and they do not include test equipment uncertainty. If a particular measurement falls slightly outside the allowable range, recalculate the new limits based on both Model 2800 specifications and pertinent calibration equipment specifications.

### Example reading limit calculation

The following is an example of how reading limits have been calculated. Assume you are testing the CDMAOne cell standard primary power measurement function at -10dBm. Using  $\pm 0.5$ dB accuracy, the calculated limits are:

Reading limits = -10dBm  $\pm 0.5$ dB

Reading limits = -10.5dBm to -9.5dBm

## Restoring factory defaults

Before performing the procedures, restore the instrument to its factory defaults as follows:

1. Press **SHIFT** and then **Setup**.
2. Select **RESET**, then restore the factory default conditions by pressing **Enter**.

# Test summary and considerations

## Functionality test summary

Model 2800 test procedures include:

- Noise floor
- Reference frequency accuracy
- VSWR
- Primary channel power measurement accuracy
- Adjacent channel power measurement accuracy
- Alternate channel power measurement accuracy

## Test considerations

When performing the procedures:

- Be sure to restore factory defaults as outlined above.
- Make sure that the test equipment is properly warmed up and connected to the correct Model 2800 connector.
- Make sure the test equipment and Model 2800 settings are correct.

# Functionality tests

## VSWR measurements

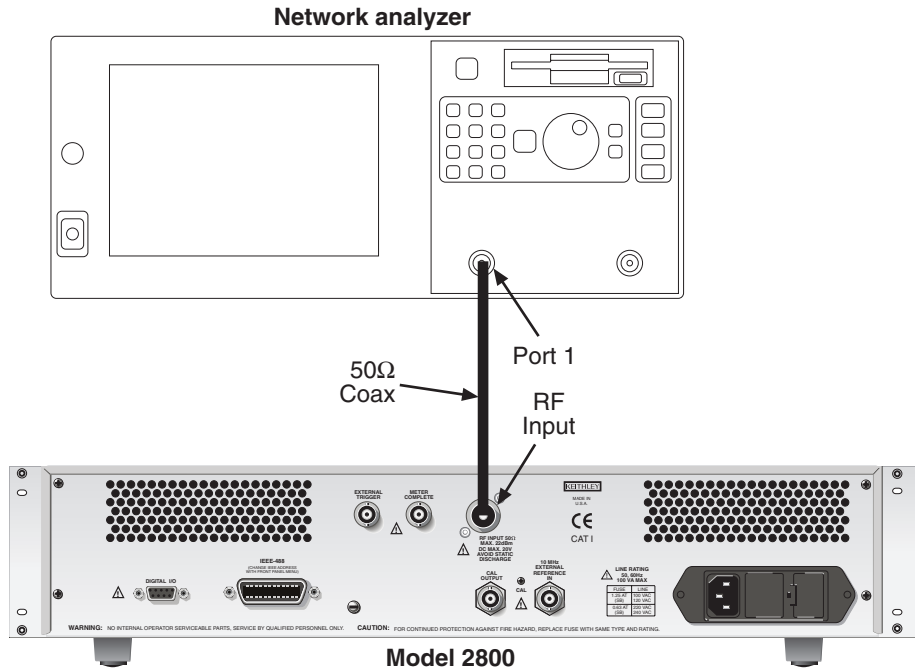
**NOTE** *If VSWR tests fail, most likely the accuracy tests for power measurement accuracy tests will also fail.*

**NOTE** *Verify the VSWR reading limits in the following procedure with the most recent product specifications (see [Appendix A](#)).*

1. Turn on the network analyzer and allow the unit to warm up for one hour.
2. Set up and calibrate the network analyzer to measure 1-port S-parameters (i.e., either S11 or S22). Set up a frequency sweep from 824 to 915MHz. Set the port power level to -20dBm. Calibration should be done using the short, open, and 50Ω load elements from the kit.
3. Set the network analyzer to the SWR display format.

4. Connect the network analyzer PORT 1 to the Model 2800 INPUT jack (Figure 4-1). Tighten connectors to the recommended specification using a torque wrench.
5. Restore Model 2800 factory defaults as previously explained and set to Band Class 9, channel 699.
6. Using the **Attenuator** key, set the attenuation value to 0dB.
7. Set the network analyzer for a test frequency of 825MHz with a 0dBm output amplitude.
8. Verify that the VSWR reading is <1.3:1.
9. Repeat steps 6 through 8 for attenuator settings at 2dB increments up to the maximum attenuation of 50dB. VSWR readings should be as follows:
  - <1.3 for settings  $\leq 4$ dB
  - <1.2 for settings >4dB
10. Repeat steps 2 through 9 using the analyzer to sweep from 1710 to 1980MHz and select Band Class 8, channel 0.

Figure 4-1  
Connections for VSWR tests



## Noise floor

When testing the primary channel noise floor for CDMA2000, it is only necessary to test using the 1x bandwidth (1.23MHz). The noise floor limits for CDMA2000 are shown in [Table 4-2](#). The equivalent noise floor limits for standards with measurement bandwidths other than 1.23MHz can then be calculated using the following equation:

### Noise floor calculation

NoiseFloor in BW2 = NDBW1 + 10 log<sub>10</sub> (BW2),

Where NDBW1 equals the noise density in dBm/Hz for CDMA2000 1x (listed by center frequency in [Table 4-2](#)) and BW2 is the 'new' measurement bandwidth that the user wishes to convert the CDMA2000 1X noise floor to. Note that this equation applies for measurements in any bandwidth, including adjacent/alternate measurements.

**Example** – Calculate the CDMAOne Cell adjacent channel noise floor. The nominal center frequency is 825MHz and the adjacent channel bandwidth (BW2 in above equation) is 30kHz. The Noise Density (NDBW1) for 825MHz is -138dBm.

$$\begin{aligned}\text{Noise Floor in BW2} &= \text{NDBW1} + 10 \log_{10} (\text{BW2}) \\ &= -138\text{dBm/Hz} + 10 \log_{10} (30\text{kHz}) \\ &= -93\text{dBm}\end{aligned}$$

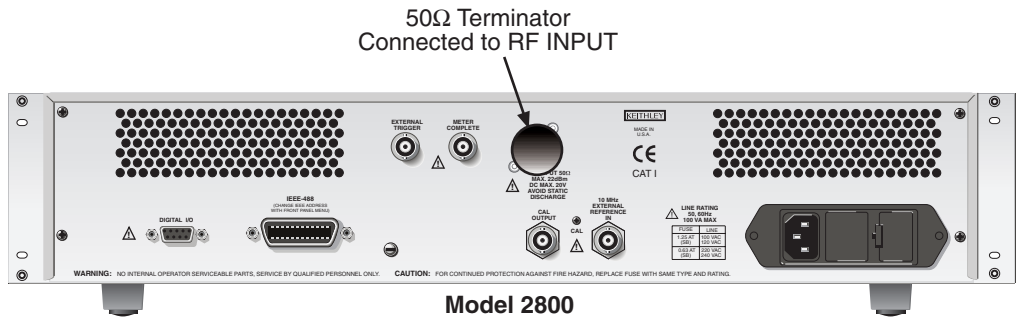
### Test procedure

The following procedure applies specifically to the CDMA2000 transmission standard. The noise floor limits are listed in [Table 4-2](#). To test other standards or bandwidths, make sure to alter the procedure appropriately and use the “[Noise floor calculation](#)” to determine the limits.

1. Connect the type N 50Ω terminator to the Model 2800 RF INPUT jack ([Figure 4-2](#)). Tighten the terminator to the recommended specification using a torque wrench.
2. Restore Model 2800 factory defaults as outlined above.
3. Using the **Attenuator** key, set the attenuation value to 0dB.
4. Press the **Standard** key to display the standard menu.
5. Using the **PREV** and **NEXT** keys, select **CDMA2K** then press **Enter**.

6. Based on the center frequency of interest, select the appropriate 1x band class and press **Enter**.
7. Press the **Avg** key, then enter 100 as the number of readings to average.
8. Press the **Acq Time** key, then enter an acquisition time of 10ms.
9. Using the **Freq Center** key, select channel 1.
10. Press the **PRIMARY** key to display primary channel power.
11. Verify that the Model 2800 primary channel power reading is no higher than the value listed in [Table 4-2](#) for the presently selected band class.
12. Repeat steps 4 through 11 to test noise floor for another center frequency and associated CDMA2000 1x band class.

**Figure 4-2**  
**Connections for noise floor tests**



**Table 4-2**  
**Noise floor test parameters**

| CDMA2000          | Frequency Range <sup>1</sup> (MHz) | Noise floor <sup>2</sup> @ 1.23MHz BW | dBm/Hz    |
|-------------------|------------------------------------|---------------------------------------|-----------|
| Band Class 0 (1x) | 824.04 to 848.97                   | ≤-77.5dBm                             | -137.5dBm |
| Band Class 1 (1x) | 1850.00 to 1909.95                 | ≤-80.0dBm                             | -140.0dBm |
| Band Class 4 (1x) | 1750.00 to 1779.95                 | ≤-78.0dBm                             | -138.0dBm |
| Band Class 6 (1x) | 1920.00 to 1979.95                 | ≤-73.6dBm                             | -133.6dBm |
| Band Class 8 (1x) | 1710.00 to 1784.95                 | ≤-78.0dBm                             | -138.0dBm |
| Band Class 9 (1x) | 880.00 to 914.95                   | ≤-75.0dBm                             | -135.0dBm |

<sup>1</sup>Center frequency of the primary channel depends on the selected channel.

<sup>2</sup>Noise floor measured for the primary channel.

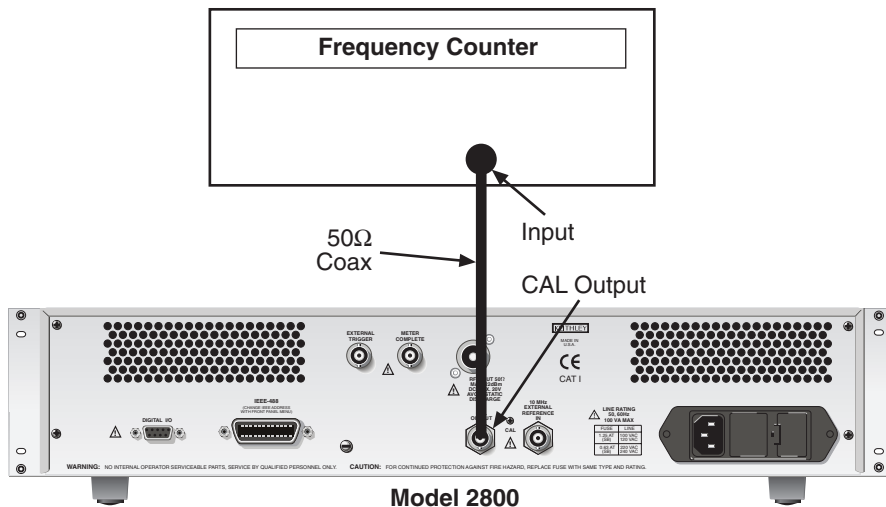


## Reference standard frequency accuracy

1. Connect the frequency counter to the Model 2800 CAL output jack (Figure 4-3).
2. Disconnect any equipment and cables from the EXTERNAL REFERENCE IN jack.
3. Restore Model 2800 factory defaults as outlined above.
4. Set the frequency counter to measure a nominal 10MHz frequency.
5. Verify that the frequency counter reading is 10MHz  $\pm$ 10Hz.

Figure 4-3

### Connections for reference standard frequency accuracy



## Primary channel power measurement accuracy

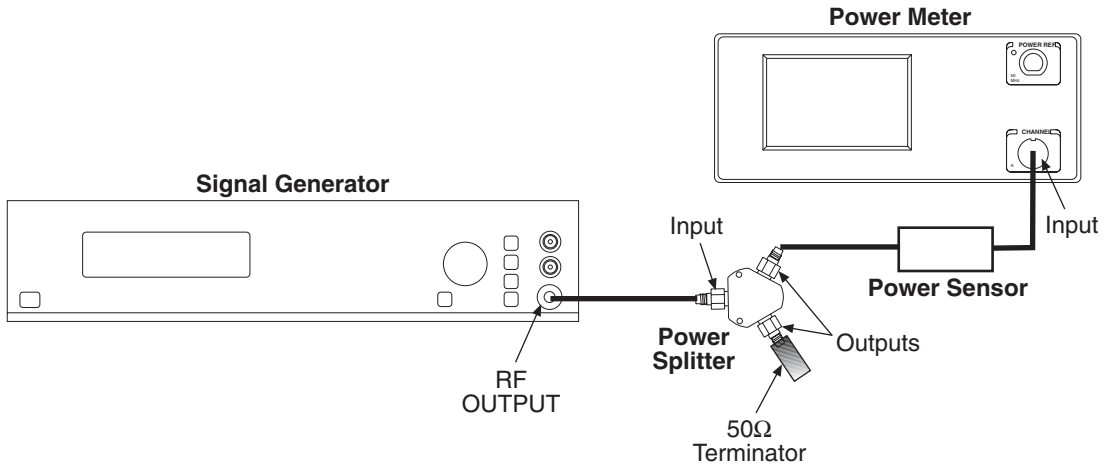
### Measuring power splitter correction factor

**NOTE** You must measure power splitter output port imbalance to determine a correction factor using the following procedure before measuring primary channel power.

1. Connect the type N 50Ω terminator, power meter, power sensor, and signal generator to the power splitter (Figure 4-4). Tighten all connectors to the recommended specification using a torque wrench.
2. Set the signal generator output to -10dBm continuous wave output as measured on the power meter.

3. Set the signal generator output frequency to 825.03MHz.
4. Note and record the power meter reading.
5. Reverse the terminator and power meter connections to the power splitter outputs so that opposite splitter output connections are used.
6. Note and record the power meter reading with reversed splitter connections.
7. Calculate the power splitter error by subtracting the reading obtained in step 6 from the one taken in step 4. Record this calculation as a positive or negative value in dB as the divider correction factor in the space provided in [Table 4-3](#).
8. Repeat steps 3 through 7 for all frequencies listed in [Table 4-3](#).

**Figure 4-4**  
**Connections for power splitter error measurements**



**Table 4-3**  
**Power splitter error summary**

| Frequency | Power meter reading <sup>1</sup><br>(normal connections) | Power meter reading <sup>1</sup><br>(reversed connections) | Power splitter<br>correction factor <sup>1</sup> |
|-----------|--|--|--|
| 825.03MHz | _____ dBm  | _____ dBm  | _____ dB   |
| 880MHz    | _____ dBm  | _____ dBm  | _____ dB   |
| 1710MHz   | _____ dBm  | _____ dBm  | _____ dB   |
| 1850MHz   | _____ dBm  | _____ dBm  | _____ dB   |
| 1920MHz   | _____ dBm  | _____ dBm  | _____ dB   |

<sup>1</sup>Calculated from power meter readings with normal and reversed connections. See procedure.

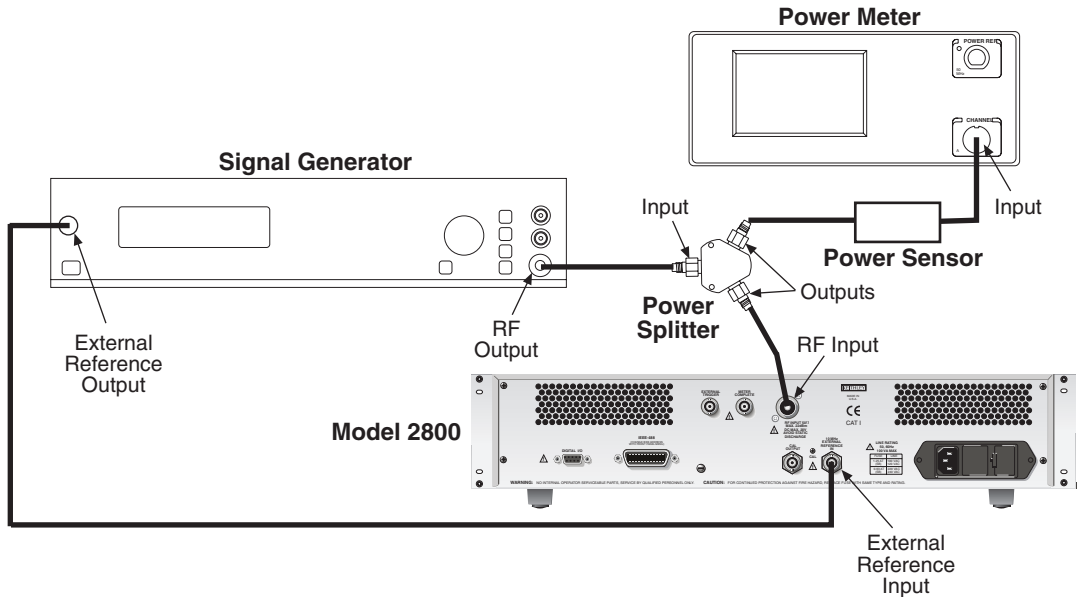
## Power meter calibration

Make sure the power meter is calibrated before using it.

### Primary channel power measurements

1. Connect the Model 2800, signal generator, power meter, and power sensor to the power splitter (see [Figure 4-5](#)). As shown, make sure the external reference is connected. Tighten all connectors to the recommended specification using a torque wrench.
2. Restore Model 2800 factory defaults as outlined above.
3. Using the **Attenuator** key, set the attenuation value to 26dB.
4. Press the **Standard** key to display the standard menu.
5. Using the **PREV** and **NEXT** keys, select **CDMA2K** then press **Enter**.
6. Select the desired band class, then press **Enter**.
7. Press the **Avg** key, then enter 10 as the number of readings to average.
8. Press the **Acq Time** key, then enter an acquisition time of 10ms.
9. Press the **PRIMARY** key to display primary channel power.
10. Set the signal generator for a 0dBm CW (continuous wave) output as measured on the power meter at a frequency of 825.03MHz.
11. Note and record the power meter reading, then apply the divider correction factor calculated previously. Verify that the corrected power reading is within the limits for the present standard and frequency listed in [Table 4-4](#).
12. Repeat steps 4 through 11 for all band classes and frequencies listed in [Table 4-4](#).

**Figure 4-5**  
**Connections for primary, adjacent, and alternate channel power measurements**



**Table 4-4**  
**Primary channel power measurement test summary**

| CDMA2000 (1x) | Signal generator output frequency <sup>1</sup> | Corrected power meter reading <sup>2</sup> | Model 2800 reading limits <sup>3</sup> |
|---------------|--|--|--|
| Band Class 0  | 825.03MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 1  | 1850.05MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 4  | 1750.05MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 6  | 1920.05MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 8  | 1710.05MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 9  | 880.05MHz                                      | _____ dBm                                  | _____ dBm                              |

<sup>1</sup>Assumes factory default frequency for each band class.

<sup>2</sup>Calculated from power splitter correction factor.

<sup>3</sup>User needs to calculate this value based on the test equipment uncertainty.

## Adjacent channel power measurement accuracy

**NOTE** *You must measure power splitter output port imbalance to determine a correction factor using the procedure above before measuring adjacent channel power.*

1. Connect the Model 2800, signal generator, power meter, and power sensor to the power splitter (Figure 4-5). As shown, make sure the external reference is connected. Tighten all connectors to the recommended specification using a torque wrench.

**NOTE** *This procedure checks the upper adjacent channel. To check the lower adjacent channel, change the “Signal generator output frequency” settings in Table 4-5 to test the lower adjacent channel and repeat the following steps. For the lower adjacent channel, the signal generator output is calculated as follows:*

*Output frequency = Center frequency - Adjacent offset frequency*

2. Restore Model 2800 factory defaults as outlined above.
3. Using the **Attenuator** key, set the attenuation value to 26dB.
4. Press the **Standard** key to display the standard menu.
5. Using the **PREV** and **NEXT** keys, select **CDMA2K** then press **Enter**.
6. Select the desired band class, then press **Enter**.
7. Using the **MHz** key, set Model 2800 to the appropriate center frequency as shown in Table 4-5.
8. Press the **Avg** key, then enter 10 as the number of readings to average.
9. Press the **Acq Time** key, then enter an acquisition time of 10ms.
10. Press the **ADJ/ALT** key until the front panel displays adjacent channel power.
11. Set the signal generator to the appropriate signal generator output frequency as shown in Table 4-5.

12. Note and record the power meter reading for the upper adjacent channel, then apply the divider correction factor calculated previously. Verify that the corrected power reading is within the limits for the present standard and frequency listed in [Table 4-5](#).
13. Repeat steps 4 through 12 for all band classes and frequencies listed in [Table 4-5](#).

Table 4-5

**Upper adjacent channel power measurement test summary**

| CDMA2000 (1x) | 2800 center frequency | Signal generator output frequency <sup>1</sup> | Corrected power meter reading <sup>2</sup> | Model 2800 reading limits <sup>3</sup> |
|---------------|-----------------------|--|--|--|
| Band Class 0  | 825.03MHz             | 825.915MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 1  | 1850.05MHz            | 1851.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 4  | 1750.05MHz            | 1751.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 6  | 1920.05MHz            | 1921.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 8  | 1710.05MHz            | 1711.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 9  | 880.05MHz             | 880.935MHz                                     | _____ dBm                                  | _____ dBm                              |

<sup>1</sup> Output frequency = Center frequency + Adjacent offset frequency. Assumes factory defaults for center frequencies and adjacent offsets.

<sup>2</sup> Calculated from power splitter correction factor.

<sup>3</sup> User needs to calculate this value based on the test equipment uncertainty.

## Alternate channel power measurement accuracy

**NOTE** *You must measure power splitter output port imbalance to determine a correction factor using the procedure above before measuring alternate channel power.*

1. Connect the Model 2800, signal generator, power meter, and power sensor to the power splitter ([Figure 4-5](#)). As shown, make sure the external reference is connected. Tighten all connectors to the recommended specification using a torque wrench.

**NOTE** *This procedure checks the upper alternate channel. To check the lower alternate channel, change the “Signal generator output frequency” settings in [Table 4-6](#) to test the lower alternate channel, and repeat the following steps. For the lower alternate channel, the signal generator output is calculated as follows:*

$$\text{Output frequency} = \text{Center frequency} - \text{Alternate offset frequency}$$

2. Restore Model 2800 factory defaults as outlined above.

3. Using the **Attenuator** key, set the attenuation value to 26dB.
4. Press the **Standard** key to display the standard menu.
5. Using the **PREV** and **NEXT** keys, select **CDMA2K** then press **Enter**.
6. Select the desired band class, then press **Enter**.
7. Using the **MHz** key, set Model 2800 to the appropriate center frequency as shown in [Table 4-6](#).
8. Press the **Avg** key, then enter 10 as the number of readings to average.
9. Press the **Acq Time** key, then enter an acquisition time of 10ms.
10. Press the **ADJ/ALT** key to until the front panel displays alternate channel power.
11. Set the signal generator for a 0dBm continuous wave output using the appropriate signal generator output frequency as shown in [Table 4-6](#).
12. Note and record the power meter reading for the upper alternate channel, then apply the divider correction factor calculated previously. Verify that the corrected power reading is within the limits for the present standard and frequency listed in [Table 4-6](#).
13. Repeat steps 4 through 12 for all band classes and frequencies listed in [Table 4-6](#).

*Table 4-6  
Upper alternate channel power measurement test summary*

| CDMA2000 (1x) | 2800 center frequency | Signal generator output frequency <sup>1</sup> | Corrected power meter reading <sup>2</sup> | Model 2800 reading limits <sup>3</sup> |
|---------------|-----------------------|--|--|--|
| Band Class 0  | 825.03MHz             | 825.915MHz                                     | _____ dBm                                  | _____ dBm                              |
| Band Class 1  | 1850.05MHz            | 1851.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 4  | 1750.05MHz            | 1751.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 6  | 1920.05MHz            | 1921.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 8  | 1710.05MHz            | 1711.3MHz                                      | _____ dBm                                  | _____ dBm                              |
| Band Class 9  | 880.05MHz             | 880.935MHz                                     | _____ dBm                                  | _____ dBm                              |

<sup>1</sup> Output frequency = Center frequency - Alternate offset frequency. Assumes factory defaults for center frequencies and alternate offsets.

<sup>2</sup> Calculated from power splitter correction factor.

<sup>3</sup> User needs to calculate this value based on the test equipment uncertainty.

# 5 Disassembly

---

## Section topics

### Handling and cleaning

- Handling PC boards and modules
- Static sensitive devices

### Disassembly procedures

#### Disassembly procedures

- Module locations
- Case cover removal
- RF DownConverter module removal
- Local oscillator amplifier removal
- Reference module removal
- Frequency synthesizer module removal
- DSP module removal
- Digital board removal
- Removing power components
  - Power supply board removal
  - Power transformer removal
  - Power module removal

- Front panel disassembly
- Fan removal

### Instrument reassembly

- Power module wire connections



# Introduction

**NOTE** *Any disassembly requires re-calibration of the Model 2800. The unit must be returned to Keithley for calibration. Keithley contact information is found at the front of this manual.*

This section explains how to handle, clean, and disassemble the Model 2800. Disassembly drawings are located at the end of this section.

**WARNING** **The information in this section is intended only for qualified service personnel. Do not attempt to replace parts or otherwise service the unit unless you are qualified to do so.**

## Handling and cleaning

To avoid contaminating PC board traces with body oil or other foreign matter, avoid touching the PC board traces while you are removing modules or boards. Some circuit board areas have high-impedance devices or sensitive circuitry where contamination could cause degraded performance.

### Handling PC boards and modules

Observe the following precautions when handling PC boards and modules:

- Wear cotton gloves.
- Only handle PC boards and modules by the edges and shields.
- Do not touch any board traces or components not associated with repair.
- Do not touch areas adjacent to electrical contacts.
- Use dry nitrogen gas to clean dust off PC boards.

## Static sensitive devices

CMOS devices operate at very high impedance levels. Therefore, any static that builds up on you or your clothing may be sufficient to destroy these devices if they are not handled properly. Use the following precautions to avoid damaging them:

**CAUTION** Many CMOS devices are installed in the Model 2800. Handle all semiconductor devices as being static sensitive.

- Transport and handle ICs only in containers specially designed to prevent static build-up. Typically, you will receive these parts in anti-static containers made of plastic or foam. Keep these devices in their original containers until ready for installation.
- Remove the devices from their protective containers only at a properly grounded work station. Also, ground yourself with a suitable wrist strap.
- Handle the devices only by the body; do not touch the pins.
- Ground any printed circuit board, into which a semiconductor device is to be inserted, to the bench or table.
- Use only anti-static type desoldering tools.
- Use only grounded-tip soldering irons.
- Once the device is installed in the PC board, it is normally adequately protected and you can handle the boards normally.

# Disassembly procedures

**WARNING** The information in this section is intended only for qualified service personnel. Do not attempt to replace parts or otherwise service the unit unless you are qualified to do so.

**WARNING** Before attempting to service the equipment, disconnect the line cord and all cables attached to the instrument.

## Module locations

Figure 5-1 shows top module locations, while Figure 5-2 shows bottom module locations. Top modules must be removed to access the bottom modules as outlined in the disassembly procedures.

*Figure 5-1*  
**Top module locations**

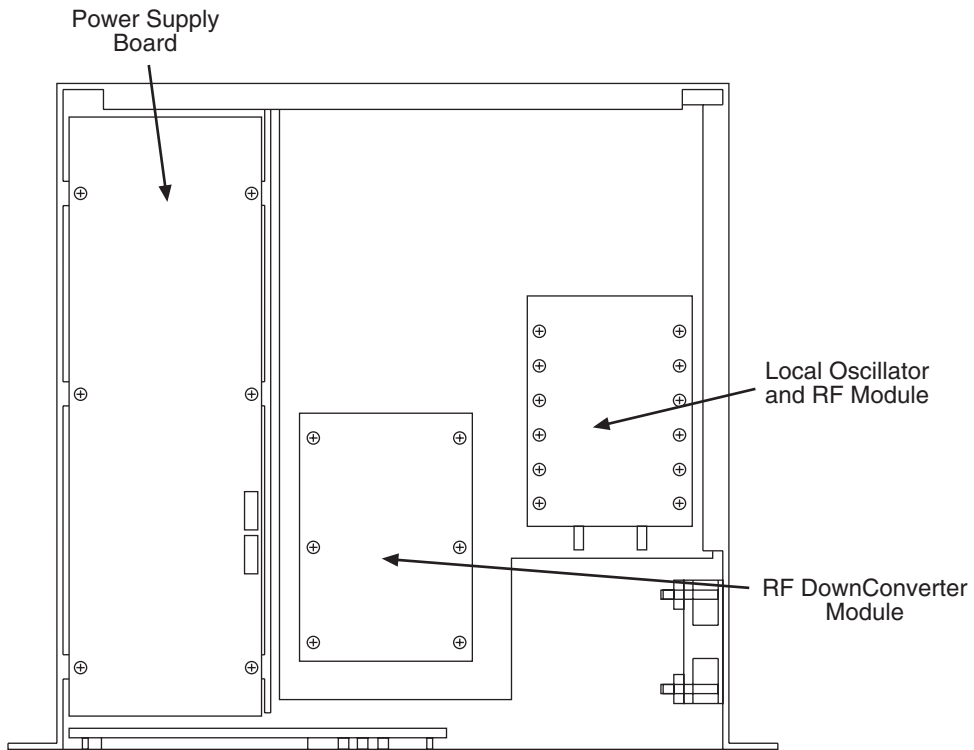
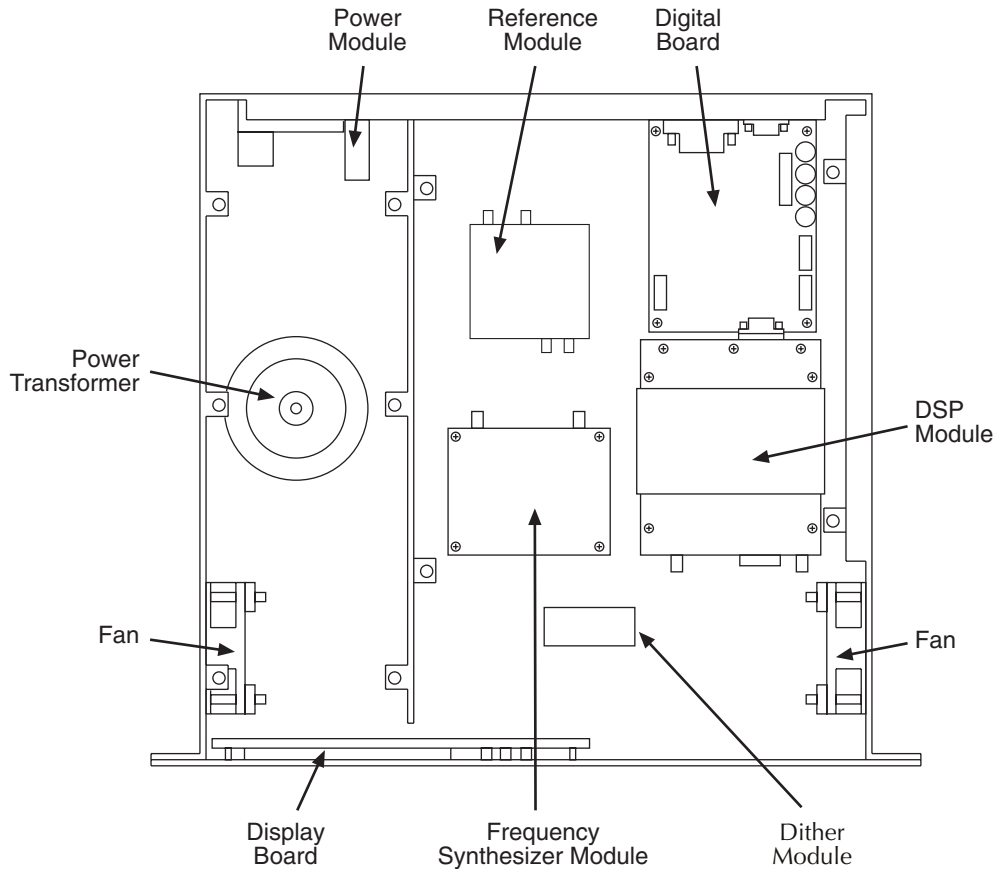


Figure 5-2  
Bottom module locations



## Case cover removal

Follow the steps below to remove the case cover to gain access to internal parts.

**WARNING** Before removing the case cover, disconnect the line cord and any cables from the instrument.

1. Remove the four screws (two on each side) that secure the case to the chassis.
2. To remove the case, grasp the case at the sides and carefully slide the cover back about an inch, then lift the case cover off the chassis.

## RF DownConverter module removal

The RF DownConverter module is located at the center of the chassis on top (Figure 5-1). Perform the following steps to remove the module. This procedure assumes that the case cover is already removed.

1. Note and record the location of all coax and control connectors, then disconnect them from the module.
2. Remove the screws that secure the module to the chassis.
3. Unscrew the coax attenuator connector that connects the module circuit board to the short flexible coax cable while slowly pulling the module away from the rear of the chassis.
4. Once the coax connector is clear, remove the module from the chassis.

## Local oscillator amplifier removal

The local oscillator amplifier is located at the right of the chassis on top (Figure 5-1). Perform the following steps to remove the module. This procedure assumes that the case cover is already removed.

1. Note and record the location of all coax and control connectors, then disconnect them from the module.
2. Remove the screws that hold the top cover of the module to the bottom cover.
3. Remove the three SMA hex nuts and two jack screws.
4. The top cover is now free to be removed.
5. Remove the four screws that hold the PCB to the bottom cover, and remove the PCB.

## Reference module removal

The reference module is located at the bottom center of the chassis near the rear panel (Figure 5-2). Perform the following steps to remove the module. This procedure assumes that the RF DownConverter, local oscillator, and RF modules are already removed.

1. Remove the screws that secure the top shelf to the chassis, then remove the shelf.
2. Note and record the location of all cables, then disconnect them from the module.
3. Remove the screws that secure the module to the chassis, which is accessible from under the unit.
4. Remove the module from the chassis.

## Frequency synthesizer module removal

The frequency synthesizer module is located at the bottom center of the chassis near the front (Figure 5-2). Perform the following steps to remove the module. This procedure assumes that the RF DownConverter and local oscillator amplifier modules are already removed.

1. If you have not already done so, remove the screws that secure the top shelf to the chassis. Remove the shelf.
2. Note and record the location of all connectors, then disconnect them from the module.
3. Remove the top screws that secure the module to the chassis.
4. Remove the module from the chassis by pulling it away from the standoffs that attach to the bottom of the chassis.

## DSP module removal

The DSP module is located at the bottom right of the chassis near the front (Figure 5-2). Perform the following steps to remove the module. This procedure assumes that the RF DownConverter and local oscillator amplifier modules are already removed.

1. If you have not already done so, remove the screws that secure the top shelf to the chassis. Remove the shelf.
2. Note and record the location of all connectors, then disconnect them from the module.
3. Remove the nuts that secure the shelf bracket to the chassis, then remove the bracket.
4. Pull the DSP module away from the digital board. You should feel a small amount of resistance due to the connection of the 9-pin DSUB connector on the DSP board to the mating.
5. The DSP module should now lift freely out of the chassis.

## Dither module removal

The dither module is located near the DSP module (Figure 5-2). Perform the following steps to remove the module.

1. Remove all cables connected to the dither module.
2. Pull firmly on the module until it comes free.

## Digital board removal

The digital board is located at the bottom right of the chassis near the back (Figure 5-2). Perform the following steps to remove the module. This procedure assumes that the RF DownConverter, local oscillator, RF module, and DSP module are already removed.

1. Note and record the location of all connectors, then disconnect them from the board.
2. Remove the jack screws that secure the Digital I/O and IEEE-488 connectors to the rear of the chassis.
3. Remove the screws that secure the board to the chassis.
4. Remove the board from the chassis by pulling it free of the rear of the chassis.

## Removing power components

Use the following procedures to remove the power supply board, power transformer, and power module. This requires that the case cover be removed as previously explained.

### Power supply board removal

The power supply board is located at the left side of the chassis on top (Figure 5-2). Perform the following steps to remove the board:

1. Remove the screws that secure the power supply board to the chassis, then remove the board.
2. Carefully hinge the power supply board up to the right until you can access the connectors underneath.
3. Note and record the location of all ribbon and power connectors, then disconnect them from the board.
4. Remove the power supply board from the chassis by pulling it free of the chassis.

## Power transformer removal

Perform the following steps to remove the toroidal power transformer (Figure 5-2):

1. Disconnect the wires going to the rear panel power module.
2. Remove the screw that secures the transformer to the bottom of the chassis.
3. Remove the power transformer from the chassis.

## Power module removal

Perform the following steps to remove the power module (Figure 5-2):

1. Disconnect the wires going to the power transformer if you have not already done so.
2. Disconnect the power module's ground wire. This green and yellow wire connects to a threaded stud on the chassis with a keq nut.
3. Remove the two screws that secure the power module to the rear of the chassis.
4. Push the module out of the access hole.

## Front panel disassembly

Use the following procedures to remove the display board (Figure 5-2) and/or the pushbutton switch pad:

1. Unplug the display board ribbon cable if you have not already done so.
2. Remove the front panel assembly. Remove the four screws (two on each side) that secure the front panel to the chassis, then pull the front panel assembly forward until it separates from the chassis.
3. Remove the four screws that secure the display board to the front panel. Pull the display board from the front panel.
4. Remove the switch pad by pulling it from the front panel.

## Fan removal

Use the following procedures to remove the two cooling fans (Figure 5-2):

1. Unplug the fan power cables if you have not already done so.
2. Remove the screws and nuts that secure each fan to the side and bottom of the chassis.
3. Remove the fan by pulling it free of the chassis.

**NOTE**      *The air flow direction for the fan is from outside to inside.*



## Instrument reassembly

Reassemble the instrument by reversing the previous disassembly procedures. Make sure that all parts are properly seated and secured, and that all connections are properly made.

Torque wrench – To tighten SMA connectors, use an SMA torque wrench set at 10 in-lbs.

**WARNING** To ensure continued protection against electrical shock, verify that power line ground (green and yellow wire attached to the power module) is connected to the chassis. Also, be sure to install all four case screws to assure a good case-to-chassis ground connection.

## Power module wire connections

Use the information in [Table 5-1](#) to connect power module wires.

*Table 5-1*  
**Power module wire colors**

| Location<br>(viewed from rear) | Wire color  |
|--------------------------------|-------------|
| Left top to right top          | Blue jumper |
| Center top                     | Blue        |
| Left middle                    | Orange      |
| Center middle                  | White       |
| Left bottom                    | Black       |
| Center bottom                  | Yellow      |
| Right bottom                   | Brown       |

# A Specifications

---

# Model 2800 RF Power Analyzer Specifications

FREQUENCY RANGE: 824 - 849MHz, 880 - 915MHz, 1710 - 1785MHz, 1850 - 1980MHz

## FREQUENCY SETTINGS:

| Cellular Standard                                    | Channel number, N                                      | Center Frequency, MHz  | Frequency Band, MHz     |
|--|--|------------------------|-------------------------|
| cdmaOne Cellular Band                                | 1 - 799  | 825.000 + 0.030*N      | 824.01MHz - 848.97MHz   |
|  | 990 - 1023   | 825.000+0.030*(N-1023) |                         |
| cdmaOne PCS Band                                     | 0 - 1199   | 1850.000 + 0.050*N     | 1850.00MHz - 1909.95MHz |
| North American Digital Cellular (NADC) Cellular Band | 1 - 799  | 825.000+ 0.030*N       | 824.01MHz - 848.97MHz   |
|  | 990 - 1023   | 825.000+0.030*(N-1023) |                         |
| North American Digital Cellular (NADC) PCS Band      | 1 - 1999   | 1849.980+0.030*N       | 1850.01MHz - 1909.95MHz |
| AMPS   | 1 - 799  | 825.000+0.030*N        | 824.01MHz - 848.97MHz   |
|  | 990 - 1023   | 825.000+0.030*(N-1023) |                         |
| GSM Cellular Band (GSM 850)                          | 128 - 251  | 824.2 + 0.2*(N-128)    | 824.2MHz - 848.8MHz     |
| GSM Cellular Band (P-GSM 900, E-GSM 900)             | 0 - 124  | 890.0 + 0.2*N          | 880.2MHz - 914.8MHz     |
|  | 975 - 1023   | 890.0 + 0.2*(N-1024)   |                         |
| DCS 1800 Band  | 512 - 885  | 1710.2 + 0.2*(N-512)   | 1710.2MHz - 1784.8MHz   |
| GSM PCS 1900 Band                                    | 512 - 810  | 1850.2 + 0.2*(N-512)   | 1850.2MHz - 1909.8MHz   |
| cdma2000 Band Class 0                                | 1 - 799  | 825.000+0.030*N        | 824.04MHz - 848.97MHz   |
|  | 991 - 1023   | 825.000+0.030*(N-1023) |                         |
| cdma2000 Band Class 1                                | 0 - 1199   | 1850.000+0.050*N       | 1850.00MHz - 1909.95MHz |
| cdma2000 Band Class 4                                | 0 - 599  | 1750.000+0.050*N       | 1750.00MHz - 1779.95MHz |
| cdma2000 Band Class 6                                | 0 - 1199   | 1920.000+0.050*N       | 1920.00MHz - 1979.95MHz |
| cdma2000 Band Class 8                                | 0 - 1499   | 1710.000 +0.050*N      | 1710.00MHz - 1784.95MHz |
| cdma2000 Band Class 9                                | 0 - 699  | 880.000 + 0.050*N      | 880.00MHz - 914.95MHz   |
| WCDMA Operating Band 1                               | 9612 - 9888  | 0.2*N                  | 1922.4MHz - 1977.6MHz   |
| WCDMA Operating Band 2                               | 9262 - 9538  | 0.2*N                  | 1852.4MHz - 1907.6MHz   |
|  | 12, 37, 62, 87, 112, 137, 162, 187, 212, 237, 262, 287 | 1850.1+0.2*N           |                         |
| WCDMA Operating Band 3                               | 8562 - 8913  | 0.2*N                  | 1712.4MHz - 1782.6MHz   |
| Korean PCS   | 0 - 599  | 1750.000+0.050*N       | 1750.00MHz - 1779.95MHz |
| PHS  | 1 - 82   | 1895.15+0.3*(N-1)      | 1884.65MHz - 1919.45MHz |
|  | 221 - 255  | 1884.65+0.3*(N-221)    |                         |

### INTERNAL REFERENCE OSCILLATOR REFERENCE:

Aging per year: 1ppm  
 Temperature Drift: (5°C to 40°C): 0.5ppm

### EXTERNAL FREQUENCY REFERENCE INPUT:

Frequency: 10MHz ± 5ppm  
 Power: ≥2dBm  
 Input Impedance: 50Ω, nominal

### INPUT:

Connector: Type N Female  
 Impedance: 50Ω  
 Maximum Overload Value: 22dBm, continuous, 20V<sub>DC</sub>  
 VSWR: < 1.3:1

# Model 2800 RF Power Analyzer Specifications

## PRIMARY CHANNEL: MEASUREMENT RANGES

|   | CDMAOne   | NADC                                      | AMPS                  | GSM   | CDMA2000 1X  | CDMA2000 3X <sup>2</sup>  | WCDMA <sup>2</sup>   | PHS                |
|---|---|---|-----------------------|---|--|---|--|--------------------|
| <b>Measurement Bandwidth<sup>9</sup></b>                                  | 1.23MHz   | 100kHz                                    | 30kHz                 | 400kHz  | 1.23MHz  | 3.69MHz   | 3.84MHz  | 192kHz/1MHz        |
| <b>Accuracy (23°C ± 5°C)<sup>1</sup>:</b><br>20dBm to -50dBm              | ±0.5dB  | ±0.5dB                                    | ±0.5dB                | ±0.5dB  | All Bands ± 0.5dB  | All Bands ± 0.5dB   | All Bands ± 0.5dB  | ±0.5dB             |
| -50.01dBm to -60dBm   | Cellular:<br>± 0.73dB<br>PCS:<br>± 0.73dB<br>Korean:<br>± 0.73dB<br>PCS | Cellular:<br>± 0.73dB<br>PCS:<br>± 0.6dB  | Cellular:<br>± 0.73dB | Cellular:<br>± 0.73dB<br>DCS:<br>± 0.6dB<br>PCS:<br>± 0.6dB   | All Bands ± 0.73dB   | Band Class 0:<br>± 0.6dB<br>Band Class 1:<br>± 0.6dB<br>Band Class 4,8:<br>± 0.6dB<br>Band Class 6:<br>± 0.72dB<br>Band Class 9:<br>± 0.6dB | Operating Band 1:<br>± 0.73dB<br>Operating Band 2:<br>± 0.6dB<br>Operating Band 3:<br>± 0.6B | ±0.7dB             |
| -60.01dBm to -70dBm <sup>2</sup>  | Cellular:<br>± 1.07dB<br>PCS:<br>± 0.94dB<br>Korean PCS:<br>± 1.02dB    | Cellular:<br>± 0.94dB<br>PCS:<br>± 0.94dB | Cellular:<br>± 0.94dB | Cellular:<br>± 0.94dB<br>DCS:<br>± 0.94dB<br>PCS:<br>± 0.94dB | Band Class 0:<br>± 1.07dB<br>Band Class 1:<br>± 0.94dB<br>Band Class 4,8:<br>± 1.02dB<br>Band Class 9:<br>± 1.65dB               |   |  | ±0.9dB             |
| <b>Repeatability</b> Mod. <sup>3</sup><br>CW <sup>12</sup>                | ± 0.1dB<br>±0.05dB  | ± 0.1dB<br>±0.05dB                        | ± 0.1dB<br>±0.05dB    | ± 0.1dB<br>±0.05dB  | ± 0.1dB<br>±0.05dB   | ± 0.1dB<br>±0.05dB  | ± 0.1dB<br>±0.05dB   | ± 0.1dB<br>±0.05dB |
| <b>Noise Floor<sup>19</sup>:</b><br>Cellular Band<br>DCS Band<br>PCS Band | ≤-77.5dBm<br>≤-80.0dBm  | ≤-88.6dBm<br>≤-91.0dBm                    | ≤-89.0dBm             | ≤-82.4dBm<br>≤-82.9dBm<br>≤-84.9dBm                           | Band Class 0 ≤-77.5dBm<br>Band Class 1 ≤-80.0dBm<br>Band Class 4,8 ≤-78.0dBm<br>Band Class 6 ≤-73.6dBm<br>Band Class 9 ≤-75.0dBm | Band Class 0 ≤-72.7dBm<br>Band Class 1 ≤-75.2dBm<br>Band Class 4,8 ≤-73.2dBm<br>Band Class 6 ≤-68.8dBm<br>Band Class 9 ≤-70.2dBm            | Operating Band 1 ≤-68.7dBm<br>Operating Band 2 ≤-75.1dBm<br>Operating Band 3 ≤-73.1dBm       | ≤-82.0dBm          |

## ADJACENT CHANNEL: (PRIMARY CHANNEL INPUT SIGNAL IN RANGE 5dBm TO 20dBm)

|   | CDMAOne                                   | NADC                  | GSM                   | CDMA2000 1X  | CDMA2000 3X  | WCDMA                 | PHS                   |
|---|---|-----------------------|-----------------------|--|--|-----------------------|-----------------------|
| <b>Measurement Bandwidth<sup>9</sup></b>                                | 30kHz                                     | 25kHz                 | 30kHz                 | 30kHz  | 30kHz  | 3.84MHz               | 192kHz                |
| <b>Offset from Center Frequency:</b><br>Cellular<br>DCS, PCS            | ± 885kHz, ± 900kHz<br>± 900kHz, ± 1250kHz | ± 30kHz               | ± 200kHz              | ± 885kHz, ± 900kHz                                     | ± 2.65MHz  | ± 5MHz                | 600kHz                |
| <b>Range at Specified Accuracy<sup>15</sup></b>                         | >55dBc                                    | >36dBc                | >40dBc                | >55dBc   | >55dBc   | >33dBc                | <-42dBm               |
| <b>Accuracy (23°C ± 5°C)<sup>1</sup></b><br>Relative to primary channel | ± 1.0dB                                   | ± 1.0dB               | ± 1.0dB               | Band Classes 0,1,4,8,9: ± 1.0dB<br>Band Class 6: 1.5dB | Band Classes 0,1,4,8,9: ± 1.0dB<br>Band Class 6: 1.5dB | ± 2.0dB <sup>20</sup> | ± 1.0dB               |
| <b>Repeatability<sup>21</sup>:</b> CW                                   | ± 0.5dB <sup>4</sup>                      | ± 0.5dB <sup>11</sup> | ± 0.5dB <sup>11</sup> | ± 0.5dB <sup>4</sup>                                   | ± 0.5dB <sup>4</sup>                                   | ± 1.0dB <sup>20</sup> | ± 0.5dB <sup>11</sup> |

## ALTERNATE CHANNEL: (PRIMARY CHANNEL INPUT SIGNAL IN RANGE 5dBm TO 20dBm)

|   | CDMAOne                | NADC                  | GSM                   | CDMA2000 1X  | CDMA2000 3X  | WCDMA                 | PHS                   |
|---|------------------------|-----------------------|-----------------------|--|--|-----------------------|-----------------------|
| <b>Measurement Bandwidth<sup>9</sup></b>                                | 30kHz                  | 25kHz                 | 30kHz                 | 30kHz  | 30kHz  | 3.84MHz               | 192kHz                |
| <b>Offset from Center Frequency:</b><br>Cellular<br>DCS, PCS            | ± 1.98MHz<br>± 1.98MHz | ± 60kHz<br>± 60kHz    | ± 400kHz<br>± 400kHz  | ± 1.98MHz<br>± 1.98MHz                                 | ± 3.10MHz  | ± 10MHz               | 900kHz                |
| <b>Range at Specified Accuracy<sup>15</sup></b>                         | >55dBc                 | >48dBc                | >60dBc                | >55dBc   | >55dBc   | >43dBc                | <-42dBm               |
| <b>Accuracy (23°C ± 5°C)<sup>1</sup></b><br>Relative to primary channel | ± 1.0dB                | ± 1.0dB               | ± 1.0dB               | Band Classes 0,1,4,8,9: ± 1.0dB<br>Band Class 6: 1.5dB | Band Classes 0,1,4,8,9: ± 1.0dB<br>Band Class 6: 1.5dB | ± 2.0dB <sup>20</sup> | ± 1.0dB               |
| <b>Repeatability<sup>21</sup>:</b> CW                                   | ± 0.5dB <sup>4</sup>   | ± 0.5dB <sup>11</sup> | ± 0.5dB <sup>11</sup> | ± 0.5dB <sup>4</sup>                                   | ± 0.5dB <sup>4</sup>                                   | ± 1.0dB <sup>20</sup> | ± 0.5dB <sup>11</sup> |

# Model 2800 RF Power Analyzer Specifications

## OTHER MEASUREMENTS

**UPPER SIDEBAND POWER (AMPS STANDARD ONLY):** Measurement of Power Relative to Carrier @ 10kHz offset

**Frequency Range:** 824 - 849MHz

**Carrier Measurement Bandwidth:** 1kHz

**Level:** 20dBm to -40dBm

**Accuracy:** ± 0.5dB

**Dynamic Range:** 28dB

**FREQUENCY<sup>16</sup>: RANGE:** 824 - 849MHz, 880 - 915MHz, 1710 - 1785MHz, 1850 - 1980MHz

**Resolution:** 1Hz

**Displayed Value:** Difference between measured frequency and entered center frequency

**Measurement Window:** ± 700kHz, nominal, from entered channel number or center frequency

**Level:** 20dBm to -25dBm

**Accuracy:** ±1Hz (with external reference)

**PEAK FUNCTION:** Computes power level and frequency of 5 highest power components in primary channel power spectrum

**Range:** 824 - 849MHz, 880 - 915MHz, 1710 - 1785MHz, 1850 - 1980MHz

**Frequency Resolution:** 5kHz

**Displayed Values:** Power in dBm, Frequency in MHz

**Measurement Window:** ±1.82MHz from entered channel number or center frequency

**Level:** 20dBm to -25dBm

**Level Accuracy:** ±0.5dBm<sup>18</sup>

**GSM BURST POWER WAVEFORM ACQUISITION:** Acquires 577µs power burst

**Parameters computed:** rise time, maximum and minimum power levels of rising transient edge, maximum and minimum power levels of information transfer portion of burst

**Sample time resolution:** 1µs

**Power Accuracy:** ±0.5dB

**Number of waveform samples:** 620

### TRIGGER METHODS:

|                      | Latency |
|----------------------|---------|
| Level                | 3µs     |
| External Trigger     | 100µs   |
| IEEE-488 Bus Command | 2.5ms   |

## MEASUREMENT PARAMETERS

|                             | CDMAOne          | NADC             | AMPS | GSM             | CDMA2000 1X      | CDMA2000 3X      | WCDMA         | PHS             |
|-----------------------------|------------------|------------------|------|-----------------|------------------|------------------|---------------|-----------------|
| <b>Trigger Delay:</b> Range | 0 - 999.999ms    | 0 - 12.790ms     |      | 0 - 3.990ms     | 0 - 999.999ms    | 0 - 999.999ms    | 0 - 999.999ms | 0 - 3.990ms     |
| Resolution                  | 1µs              | 1µs              |      | 1µs             | 1µs              | 1µs              | 1µs           | 1µs             |
| <b>Acquisition Time:</b>    |                  |                  |      |                 |                  |                  |               |                 |
| Primary Channel             | 200µs            | 10µs - 999.999ms | 20ms | 1µs - 999.999ms | 200µs            | 200µs            | 200µs         | 1µs - 999.999ms |
| Adjacent/Alternate Channel  | 10µs - 999.999ms | 10µs - 999.999ms |      | 1µs - 999.999ms | 10µs - 999.999ms | 10µs - 999.999ms | 200µs         | 1µs - 999.999ms |
| Resolution                  | 10µs             | 10µs             |      | 1µs             | 10µs             | 10µs             |               | 1µs             |
| <b>Number of Averages:</b>  | 1 - 100          | 1 - 100          |      | 1 - 100         | 1 - 100          | 1 - 100          | 1 - 100       | 1 - 100         |

## MEASUREMENT TIME (TYPICAL)<sup>7</sup>:

|   | CDMAOne | NADC <sup>11</sup>    | AMPS  | GSM <sup>11</sup>     | CDMA2000 1X | CDMA2000 3X | WCDMA               | PHS                   |
|---|---------|-----------------------|-------|-----------------------|-------------|-------------|---------------------|-----------------------|
| Primary Channel Power Measurement <sup>5</sup>  | 6ms     | 11ms <sup>13</sup>    | 30ms  | 4ms <sup>13</sup>     | 6ms         | 7ms         | 7ms                 | 4ms <sup>13</sup>     |
| Primary Channel Power Measurement, Two Adjacent Channel Power Measurements, and Two Alternate Channel Power Measurements <sup>5,6</sup> | 26ms    | 16ms <sup>10,13</sup> | N/A   | 10ms <sup>10,13</sup> | 26ms        | 26ms        | 75ms <sup>17</sup>  | 10ms <sup>10,13</sup> |
| Time to complete 10 different power measurements at a single frequency <sup>5,8,14</sup>  | 80ms    | 141ms <sup>13</sup>   | 330ms | 81ms <sup>13</sup>    | 80ms        | 88ms        | 94ms                | 81ms <sup>13</sup>    |
| Time to complete 10 power measurements of a single power level at different frequencies <sup>5,14</sup>                                 | 190ms   | 371ms <sup>13</sup>   | 575ms | 325ms <sup>13</sup>   | 190ms       | 195ms       | 225ms <sup>17</sup> | 325ms <sup>13</sup>   |

# Model 2800 RF Power Analyzer Specifications

## GENERAL

**PROGRAMMABILITY:** IEEE-488.2 (SCPI – 1995.0), 3 user-definable power-up states plus factory default.

**MEMORY BUFFER: 2500 Sets Of 5 Readings** – primary channel power, upper and lower adjacent channel power, upper and lower alternate channel power, with time stamp, peak reading, average reading, and standard deviation

**DIGITAL INTERFACE:**

**Digital I/O:** 1 – digital input, 4 – digital outputs with 250mA sink capability, maximum clamp voltage: 30V<sub>DC</sub>, 5V@100mA DC Source

**REAR CONNECTIONS:** RF Input – Type N connector, External Trigger, Meter Complete, External Reference In, Cal Output – BNC connector, Digital I/O – DB9 connector, IEEE-488 – 24-Pin EMI-shielded receptacle, Power – Power Switch/Line Entry Module with DPDT switch, 2 fuses, and IEC 320 plug

**POWER SUPPLY:** 100V/120V/220V/240V

**LINE FREQUENCY:** 50Hz to 60Hz

**POWER CONSUMPTION:** 100VA

**ENVIRONMENT:**

**Operating:** 5° to 40°C, 70%R.H., non-condensing, up to 35° C

**Storage:** 0° to 50°C

**WARRANTY:** 1year

**SAFETY:** Complies with European Union Directive 73/23/EEC, EN61010-1

**EMC:** Complies with European Union Directive 89/336/EEC, EN61326-1

**VIBRATION:** MIL-PRF-28800F Class 3 Random

**WARM-UP:** 1-hour to rated accuracy during normal operation. If the instrument has been stored in an unpowered state for more than 2 weeks, allow 2 hours for warm-up.

**DIMENSIONS:**

**Bench configuration (with handle and feet):** 104mm high x 485mm wide x 478mm deep (4.125in x 19in x 18.75in)

**Rack Mounting:** 89mm high x 485mm wide x 478mm deep (3.5in x 19in x 18.75in)

**WEIGHT:**

**Net Weight:** 13.14 kg (28.9 lbs)

**Shipping Weight:** 14.5kg (32lbs)

**ACCESSORIES SUPPLIED:** Hardcopy User's manual, RF Product Information CD-ROM, rack mount kit, bench assembly kit.

**Notes:**

1. Based on measurements of NIST traceable CW signals, and locked to the source reference. Exclusive of input mismatch. Derate by  $\pm 0.05\text{dB}/^\circ\text{C}$  beyond  $23^\circ\text{C} \pm 5^\circ\text{C}$ .
2. Accuracy values are based on an average of 6 measurements for CDMA2000 3X and WCDMA measurements between  $-50.01\text{dBm}$  and  $-60\text{dBm}$ , and for all standards between  $-60.01\text{dBm}$  and  $-70\text{dBm}$ .
3. Defined as 2 standard deviations of 100 consecutive readings while measuring a modulated signal. The following conditions apply: default acquisition times and signal/noise  $> 30\text{dB}$ .
4. Measurement acquisition time: 10ms.
5. Range:  $20\text{dBm}$  to  $-60\text{dBm}$  (except CDMA2000 3X and WCDMA which are for  $20\text{dBm}$  to  $-50\text{dBm}$ ), averaging off, display off, input protection off, temperature compensation off, binary transfer, and 488.1 protocol. For measurements of only the primary channel, the adjacent channel measurements and the alternate channel measurements are disabled.
6. Adjacent channel power and alternate channel power acquisition time: 10ms.
7. Times are defined at specified repeatability and include IEEE-488 transfer times. CDMA2000 3X and WCDMA repeatability specifications are excluded.
8. Includes time required to make two attenuator changes.
9. Bandwidths are designed to conform to definitions in cellular standards.
10. Measurement times include 3 measurements: primary channel power measurements and upper and lower adjacent channel power measurements.
11. Acquisition time is one time slot: NADC = 6.0ms, GSM, PHS = 500 $\mu\text{s}$ . Level trigger is used.
12. Defined as 2 standard deviations of 100 consecutive measurements spaced evenly over an 8 hour period with ambient temperature of  $23 \pm 1^\circ\text{C}$ . The following conditions apply: default acquisition times and signal/noise  $> 30\text{dB}$ .
13. Pulse measurement times are defined from the time the rising edge of each power pulse is detected to the completion of the data transfer.
14. Times are exclusive of DUT settling times.
15. The range is defined as the maximum detectable difference between the primary channel power and the adjacent (or alternate) channel power for the specified accuracy and specified measurement bandwidth.
16. Defined for measurements on un-modulated carrier waveforms, cdmaOne, and cdma2000 1X modulated waveforms.
17. Times are based on measurements in Operating Band One.
18. Defined for signals whose carrier is at the programmed center frequency or offset by an integer multiple of 5kHz.
19. Defined with 0dB programmed attenuation,  $23^\circ\text{C} \pm 5^\circ\text{C}$ , input terminated with 50 $\Omega$ , and 100 readings averaged.
20. Based on a 200 $\mu\text{s}$  acquisition time and 6 measurements averaged.
21. Defined as 2 standard deviations of 100 consecutive measurements spaced evenly over an 8-hour period with ambient temperature of  $23 \pm 1^\circ\text{C}$ . The following conditions apply: default acquisition times and signal/noise  $> 6\text{dB}$ .

**B**

# Status and Error Messages

---

*Table B-1*  
**Status and error messages**

| <b>Number</b> | <b>Description</b>                          | <b>Event</b> |
|---------------|---|--------------|
| -440          | Query untermiated after indefinite response | EE           |
| -430          | Query deadlocked                            | EE           |
| -420          | Query untermiated                           | EE           |
| -410          | Query interrupted                           | EE           |
| -363          | Input buffer overrun                        | SYS          |
| -362          | Framing error in program message            | EE           |
| -361          | Parity error in program message             | EE           |
| -360          | Communications error                        | EE           |
| -350          | Queue overflow                              | SYS          |
| -330          | Self-test failed                            | EE           |
| -314          | Save/recall memory lost                     | EE           |
| -315          | Configuration memory lost                   | EE           |
| -285          | Program syntax error                        | EE           |
| -284          | Program currently running                   | EE           |
| -282          | Illegal program name                        | EE           |
| -281          | Cannot create program                       | EE           |
| -260          | Expression error                            | EE           |
| -241          | Hardware missing                            | EE           |
| -230          | Data corrupt or stale                       | EE           |
| -225          | Out of memory                               | EE           |
| -224          | Illegal parameter value                     | EE           |
| -223          | Too much data                               | EE           |
| -222          | Parameter data out of range                 | EE           |
| -221          | Settings conflict                           | EE           |
| -220          | Parameter error                             | EE           |
| -215          | Arm deadlock                                | EE           |
| -214          | Trigger deadlock                            | EE           |
| -213          | Init ignored                                | EE           |
| -212          | Arm ignored                                 | EE           |
| -211          | Trigger ignored                             | EE           |
| -210          | Trigger error                               | EE           |
| -203          | Command protected                           | EE           |
| -202          | Settings lost due to rtl                    | EE           |
| -201          | Invalid while in local                      | EE           |
| -200          | Execution error                             | EE           |
| -178          | Expression data not allowed                 | EE           |
| -171          | Invalid expression                          | EE           |
| -170          | Expression error                            | EE           |
| -168          | Block data not allowed                      | EE           |
| -161          | Invalid block data                          | EE           |



Table B-1 (cont.)  
**Status and error messages**

| <b>Number</b> | <b>Description</b>          | <b>Event</b> |
|---------------|-----------------------------|--------------|
| -160          | Block data error            | EE           |
| -158          | String data not allowed     | EE           |
| -154          | String too long             | EE           |
| -151          | Invalid string data         | EE           |
| -150          | String data error           | EE           |
| -148          | Character data not allowed  | EE           |
| -144          | Character data too long     | EE           |
| -141          | Invalid character data      | EE           |
| -140          | Character data error        | EE           |
| -128          | Numeric data not allowed    | EE           |
| -124          | Too many digits             | EE           |
| -123          | Exponent too large          | EE           |
| -121          | Invalid character in number | EE           |
| -120          | Numeric data error          | EE           |
| -114          | Header suffix out of range  | EE           |
| -113          | Undefined header            | EE           |
| -112          | Program mnemonic too long   | EE           |
| -111          | Header separator error      | EE           |
| -110          | Command header error        | EE           |
| -109          | Missing parameter           | EE           |
| -108          | Parameter not allowed       | EE           |
| -105          | GET not allowed             | EE           |
| -104          | Data type error             | EE           |
| -103          | Invalid separator           | EE           |
| -102          | Syntax error                | EE           |
| -101          | Invalid character           | EE           |
| -100          | Command error               | EE           |
| +000          | No error                    | SE           |

Table B-1 (cont.)  
**Status and error messages**

| Number | Description                            | Event |
|--------|--|-------|
| +106   | Reading available                      | SE    |
| +107   | Reading overflow                       | SE    |
| +108   | Buffer available                       | SE    |
| +109   | Buffer full                            | SE    |
| +115   | External reference detected            | SE    |
| +116   | External reference locked              | SE    |
| +117   | Internal reference active              | SE    |
| +118   | Local oscillator locked                | SE    |
| +200   | Operation complete                     | SE    |
| +305   | Waiting in trigger layer               | SE    |
| +310   | Re-entering the idle layer             | SE    |
| +408   | Questionable calibration               | SE    |
| +414   | Command Warning                        | SE    |
| +500   | Date of calibration not set            | EE    |
| +501   | Next date of calibration not set       | EE    |
| +502   | Calibration data invalid               | EE    |
| +509   | Not permitted with cal locked          | EE    |
| +510   | Not permitted with cal un-locked       | EE    |
| +601   | Reading buffer data lost               | EE    |
| +602   | GPIB address lost                      | EE    |
| +603   | Power-on state lost                    | EE    |
| +605   | Calibration dates lost                 | EE    |
| +606   | GPIB communication language lost       | EE    |
| +700   | Invalid system communication           | EE    |
| +703   | Trigger timeout error                  | EE    |
| +800   | Illegal with storage active            | EE    |
| +830   | Invalid with INFinite TRIG:COUNT       | EE    |
| +831   | Invalid with PEAK/FREQ function active | EE    |
| +832   | Invalid with standard selected.        | EE    |
| +833   | Meas BW is fixed or invalid            | EE    |
| +834   | Invalid with EXTERNAL trigger          | EE    |
| +900   | Internal system error                  | EE    |

EE = error event

SE = status event

SYS = system error event

# C GPIB Protocols

---

## Introduction

The Model 2800 supports two GPIB protocols: SCPI and 488.1. The 488.1 protocol is included to significantly increase speed over the GPIB.

When using the 488.1 protocol, throughput is enhanced up to 10 times for data sent to the Model 2800 (command messages) and up to 20 times for data returned by the Model 2800 (response messages).

**NOTE**      *With the 488.1 protocol selected, you will still use SCPI commands to program the Model 2800. Operation differences between the two protocols are discussed in this appendix.*

## Selecting the GPIB protocol

Perform the following steps to check/change the GPIB protocol:

When switching between the SCPI protocol and 488.1 protocol, the GPIB protocol setting is saved in EEPROM and the unit will power up with that selected protocol.

1. Press **Menu** to display the main menu.
2. Use the **NEXT** or **PREV** key to place the blinking cursor on **GPIB** and press **Enter** to display the present address. Press **Enter** again to display the protocol.
3. Place the cursor on the desired protocol (488.1 or SCPI) and press **Enter**.
4. Use the **Exit** key to back out of the menu structure.

## Remote operation

Use the following command to select GPIB protocol:

```
SYSTem:MEP[:STATe] 1 ' SCPI protocol.  
SYSTem:MEP[:STATe] 0 ' 488.1 protocol.
```

When the 488.1 protocol is selected, the message exchange protocol (MEP) disables. You can use the following query to request the state of MEP and determine which protocol is enabled:

```
SYSTem:MEP[:STATe] ?
```

If a "1" is returned, MEP is enabled and the SCPI protocol is selected. A "0" indicates that MEP is disabled and the 488.1 protocol is enabled. To summarize:

1 = SCPI protocol  
0 = 488.1 protocol

# Protocol differences

The following information covers the differences between the 488.1 protocol and the SCPI protocol.

## Message exchange protocol (MEP)

When the 488.1 protocol is selected, the MEP is disabled to speed up GPIB operation.

The following guidelines/limitations must be followed when using the 488.1 protocol:

- If a query is sent, it must be the only command on the line (this limitation also means no multiple queries can be sent). Otherwise, full SCPI command syntax is still supported including long-form commands, short form commands, multiple commands, and MIN/MAX/DEF parameter definitions.

For example, the following command strings are invalid:

```
:POW:CONC ON;*OPC?
```

```
:POW:ATT?;;READ?
```

```
:READ?;;READ?
```

The following command strings are valid:

```
:sens:aver on;;freq:chan 1
```

```
:ADJ:BAND?
```

```
:READ?
```

- When a query is sent, either the data must be read back or a Device Clear (DCL) or Interface Clear (IFC) must be performed to reset the query.
- When sending a command or query, do not attempt to read data from the Model 2800 until the terminator has been sent (usually Line Feed with EOI). Otherwise, a DCL or IFC must be sent to reset the input parser.
- When receiving data, all data, up to and including the terminator (LF with EOI), must be accepted. Otherwise, a DCL or IFC must be sent to reset the output task.
- Empty command strings (terminator only) should not be sent.

## Using SCPI-based programs

In general, an existing 488.1 based program will run properly and faster than the SCPI protocol as long as it meets the above guidelines and limitations. SCPI commands will work properly with either the SCPI or 488.1 protocol selected.

## NRFD hold-off

\*OPC, \*OPC?, and \*WAI are still functional but are not needed for the 488.1 protocol. When sending commands, the GPIB is automatically held off when it detects a terminator. The hold-off is released when all the commands have finished executing or if there is some parser or command error. An exception is an initiate command, which releases the hold-off immediately and does not wait for all of the readings to be acquired. This immediate release of bus hold-off is done to support GET, SDC, IFC, \*TRG, \*RCL, \*RST, SYSTem:PRESet, and ABORt during data acquisition.

## Trigger-on-talk

Trigger-on-talk functionality has been added for the 488.1 protocol. If a query has not been received by the instrument, the Model 2800 will automatically assume a READ? command has been sent when it is addressed to talk. This technique increases GPIB speed by decreasing the transmission and parser times for the command.

Trigger-on-talk is extremely useful in the single-shot reading mode (\*RST default) and is the main reason for an up to 30% speed improvement over the SCPI protocol.

## Message available

The MAV (message available) bit in the Serial Poll byte will be sent when the query is finished being processed, not when there is data available in the output buffer (as with the SCPI protocol). For the 488.1 protocol, output data will not be formatted until the first request for data is received. This delay may cause unexpected time-outs when using SRQ on MAV for queries that take a long time to execute.

## General operation notes

- The TALK, LSTN, and SRQ annunciators are not functional in the 488.1 protocol. This speeds up data through-put greatly. The REM annunciator still operates since it is critical to fundamental GPIB operation.
- If the unit is in REMote, the GTL command may not put the Model 2800 into the local mode. Only the front panel LOCAL key is guaranteed to operate, if not in local lockout (LLO). GTL will still disable LLO.
- IEEE-488 bus commands and features (GET, IFC, SDC, DCL, LLO, Serial Poll, and SRQ) are still fully supported.
- Multiple TALKs on the same query are supported as in the SCPI protocol. This feature is useful when reading back long ASCII strings.

**D**

# GPIB overview

---

## Introduction

The IEEE-488 bus is a communication system between two or more electronic devices. A device can be either an instrument or a computer. When a computer is used on the bus, it serves as a supervisor of the communication exchange between all the devices and is known as the controller. Supervision by the controller consists of determining which device will talk and which device will listen. As a talker, a device will output information and as a listener, a device will receive information. To simplify the task of keeping track of the devices, a unique address number is assigned to each.

On the bus, only one device can talk at a time and is addressed to talk by the controller. The device that is talking is known as the active talker. The devices that need to listen to the talker are addressed to listen by the controller. Each listener is then referred to as an active listener. Devices that do not need to listen are instructed to unlisten. The reason for the unlisten instruction is to optimize the speed of bus information transfer since the task of listening takes up bus time.

Through the use of control lines, a handshake sequence takes place in the transfer process of information from a talker to a listener. This handshake sequence helps ensure the credibility of the information transfer. The basic handshake sequence between an active controller (talker) and a listener is as follows:

1. The listener indicates that it is ready to listen.
2. The talker places the byte of data on the bus and indicates that the data is available to the listener.
3. The listener, aware that the data is available, accepts the data and then indicates that the data has been accepted.
4. The talker, aware that the data has been accepted, stops sending data and indicates that data is not being sent.
5. The listener, aware that there is no data on the bus, indicates that it is ready for the next byte of data.

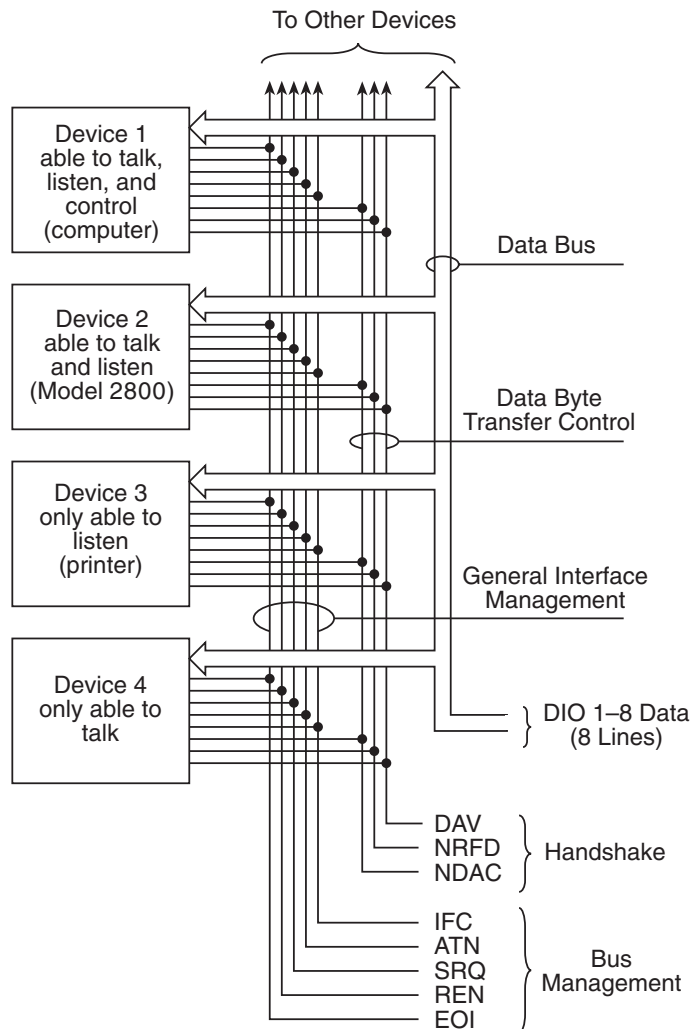
## Bus description

The IEEE-488 bus, which is also referred to as the GPIB (General Purpose Interface Bus), was designed as a parallel transfer medium to optimize data transfer without using an excessive number of bus lines. In keeping with this goal, the bus has only eight data lines that are used for both data and with most commands. Five bus management lines and three handshake lines round out the complement of bus signal lines.



A typical setup for controlled operation is shown in [Figure D-1](#). Generally, a system will contain one controller and a number of other instruments to which the commands are given. Device operation is categorized into three operators: controller, talker, and listener. The controller controls the instruments on the bus. The talker sends data while a listener receives data. Depending on the type of instrument, any particular device can be a talker only, a listener only, or both a talker and listener.

**Figure D-1**  
**GPIB configuration**



There are two categories of controllers: system controller and basic controller. Both are able to control other instruments, but only the system controller has the absolute authority in the system. In a system with more than one controller, only one controller may be active at any given time. Certain protocol is used to pass control from one controller to another.

The IEEE-488 bus is limited to 15 devices, including the controller. Thus, any number of talkers and listeners up to that limit may be present on the bus at one time. Although several devices may be commanded to listen simultaneously, the bus can have only one active talker or communications would be scrambled.

A device is placed in the talk or listen state by sending an appropriate talk or listen command. These talk and listen commands are derived from an instrument's primary address. The primary address may have any value between 0 and 31, and is generally set by rear panel DIP switches or programmed in from the front panel of the instrument. The actual listen address value sent out over the bus is obtained by OR'ing the primary address with #h20. For example, if the primary address is #h1C, the actual listen address is #h3C (#h3C = #h1C + #h20). In a similar manner, the talk address is obtained by OR'ing the primary address with #h40. With the present example, the talk address derived from a primary address of #h1C would be #h5C (#h5C = #h1C + #h40).

The IEEE-488 standards also include another addressing mode called secondary addressing. Secondary addresses lie in the range of #h60-#h7F. Note, however, that many devices, including the Model 2800, do not use secondary addressing.

Once a device is addressed to talk or listen, the appropriate bus transactions take place. For example, if the instrument is addressed to talk, it places its data string on the bus one byte at a time. The controller reads the information and the appropriate software can be used to direct the information to the desired location.

## Bus lines

The signal lines on the IEEE-488 bus are grouped into three different categories: data lines, management lines, and handshake lines. The data lines handle bus data and commands, while the management and handshake lines ensure that proper data transfer and operation takes place. Each bus line is active low, with approximately zero volts representing a logic 1 (true). The following paragraphs describe the operation of these lines.

## Data lines

The IEEE-488 bus uses eight data lines that transfer data one byte at a time. DIO1 (Data Input/Output) through DIO8 (Data Input/Output) are the eight data lines used to transmit both data and multiline commands and are bi-directional. The data lines operate with low true logic.

## Bus management lines

The five bus management lines help to ensure proper interface control and management. These lines are used to send the uniline commands.

**ATN (Attention)** — The ATN state determines how information on the data bus is to be interpreted.

**IFC (Interface Clear)** — The IFC line controls clearing of instruments from the bus.

**REN (Remote Enable)** — The REN line is used to place the instrument on the bus in the remote mode.

**EOI (End or Identify)** — The EOI line is used to mark the end of a multi-byte data transfer sequence.

**SRQ (Service Request)** — The SRQ line is used by devices when they require service from the controller.

## Handshake lines

The bus handshake lines operate in an interlocked sequence. This method ensures reliable data transmission regardless of the transfer rate. Generally, data transfer will occur at a rate determined by the slowest active device on the bus.

One of the three handshake lines is controlled by the source (the talker sending information), while the remaining two lines are controlled by accepting devices (the listener or listeners receiving the information). The three handshake lines are:

**DAV (DATA VALID)** — The source controls the state of the DAV line to indicate to any listening devices whether or not data bus information is valid.

**NRFD (Not Ready For Data)** — The acceptor controls the state of NRFD. It is used to signal to the transmitting device to hold off the byte transfer sequence until the accepting device is ready.

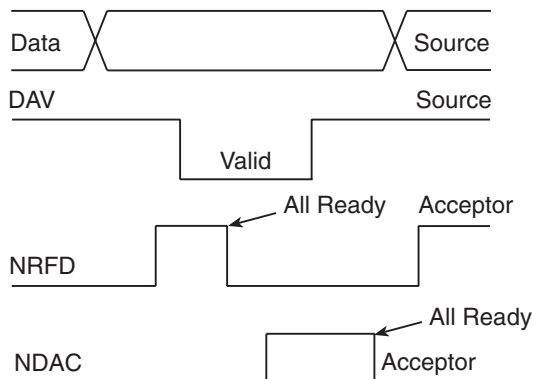
**NDAC (Not Data Accepted)** — NDAC is also controlled by the accepting device. The state of NDAC tells the source whether or not the device has accepted the data byte.

The complete handshake sequence for one data byte is shown in [Figure D-2](#). Once data is placed on the data lines, the source checks to see that NRFD is high, indicating that all active devices are ready. At the same time, NDAC should be low from the previous byte transfer. If these conditions are not met, the source must wait until NDAC and NRFD have the correct status. If the source is a controller, NRFD and NDAC must be stable for at least 100ns after ATN is set true. Because of the possibility of a bus hang up, many controllers have time-out routines that display messages in case the transfer sequence stops for any reason.

Once all NDAC and NRFD are properly set, the source sets DAV low, indicating to accepting devices that the byte on the data lines is now valid. NRFD will then go low and NDAC will go high once all devices have accepted the data. Each device will release NDAC at its own rate, but NDAC will not be released to go high until all devices have accepted the data byte.

The previous sequence is used to transfer both data, talk, and listen addresses, as well as multiline commands. The state of the ATN line determines whether the data bus contains data, addresses, or commands as described in the following paragraphs.

*Figure D-2*  
**IEEE-488 handshake sequence**



## Bus commands

The instrument may be given a number of special bus commands through the IEEE-488 interface. The following paragraphs briefly describe the purpose of the bus commands which are grouped into the following four categories.

1. **Uniline commands** — Sent by setting the associated bus lines true. For example, to assert REN (Remote Enable), the REN line would be set low (true).
2. **Multiline commands** — General bus commands which are sent over the data lines with the ATN line true (low).
3. **Common commands** — Commands that are common to all devices on the bus; sent with ATN high (false).
4. **SCPI commands** — Commands that are particular to each device on the bus; sent with ATN (false).

These bus commands and their general purpose are summarized in [Table D-1](#).

Table D-1

**IEEE-488 bus command summary**

| Command type           | Command                      | State of ATN line | Comments  |
|------------------------|------------------------------|-------------------|---|
| Uniline                | REN (Remote Enable)          | X                 | Set up devices for remote operation.                              |
|                        | EOI                          | X                 | Marks end of transmission.  |
|                        | IFC (Interface Clear)        | X                 | Clears interface.   |
|                        | ATN (Attention)              | Low               | Defines data bus contents.  |
|                        | SRQ                          | X                 | Controlled by external device.                                    |
| Multiline<br>Universal | LLO (Local Lockout)          | Low               | Locks out local operation.  |
|                        | DCL (Device Clear)           | Low               | Returns device to default conditions.                             |
|                        | SPE (Serial Enable)          | Low               | Enables serial polling.   |
|                        | SPD (Serial Poll Disable)    | Low               | Disables serial polling.  |
| Addressed              | SDC (Selective Device Clear) | Low               | Returns unit to default conditions.                               |
|                        | GTL (Go To Local)            | Low               | Returns device to local.  |
| Unaddressed            | UNL (Unlisten)               | Low               | Removes all listeners from the bus.                               |
|                        | UNT (Untalk)                 | Low               | Removes any talkers from the bus.                                 |
| Common                 | —                            | High              | Programs IEEE-488.2 compatible instruments for common operations. |
| SCPI                   | —                            | High              | Programs SCPI compatible instruments for particular operations.   |

## Uniline commands

ATN, IFC, and REN are asserted only by the controller. SRQ is asserted by an external device. EOI may be asserted either by the controller or other devices

depending on the direction of data transfer. The following is a description of each command. Each command is sent by setting the corresponding bus line true.

**REN (Remote Enable)** — REN is sent to set up instruments on the bus for remote operation. When REN is true, devices will be removed from the local mode. Depending on device configuration, all front panel controls except the LOCAL button (if the device is so equipped) may be locked out when REN is true. Generally, REN should be sent before attempting to program instruments over the bus.

**EOI (End or Identify)** — EOI is used to positively identify the last byte in a multi-byte transfer sequence, thus allowing data words of various lengths to be transmitted easily.

**IFC (Interface Clear)** — IFC is used to clear the interface and return all devices to the talker and listener idle states.

**ATN (Attention)** — The controller sends ATN while transmitting addresses or multiline commands.

**SRQ (Service Request)** — SRQ is asserted by a device when it requires service from a controller.

## Universal multiline commands

Universal commands are those multiline commands that require no addressing. All devices equipped to implement such commands will do so simultaneously when the commands are transmitted. As with all multiline commands, these commands are transmitted with ATN true.

**LLO (Local Lockout)** — LLO is sent to the instrument to lock out the LOCAL key and all their front panel controls.

**DCL (Device Clear)** — DCL is used to return instruments to some default state. Instruments usually return to their power-up conditions.

**SPE (Serial Poll Enable)** — SPE is the first step in the serial polling sequence which is used to determine which device has requested service.

**SPD (Serial Poll Disable)** — SPD is used by the controller to remove all devices on the bus from the serial poll mode and is generally the last command in the serial polling sequence.

## Addressed multiline commands

Addressed commands are multiline commands that must be preceded by the device listen address before that instrument will respond to the command in ques-

tion. Note that only the addressed device will respond to these commands. Both the commands and the address preceding it are sent with ATN true.

**SDC (Selective Device Clear)** — The SDC command performs essentially the same function as the DCL command except that only the addressed device responds. Generally, instruments return to their power-up default conditions when responding to the SDC command.

**GTL (Go To Local)** — The GTL command is used to remove instruments from the remote mode. With some instruments, GTL also unlocks front panel controls if they were previously locked out with the LLO command.

**GET (Group Execute Trigger)** — The GET command is used to trigger devices to perform a specific action that depends on device configuration (for example, take a reading). Although GET is an addressed command, many devices respond to GET without addressing.

## Address commands

Addressed commands include two primary command groups and a secondary address group. ATN is true when these commands are asserted. The commands include:

**LAG (Listen Address Group)** — These listen commands are derived from an instrument's primary address and are used to address devices to listen. The actual command byte is obtained by OR'ing the primary address with #h20.

**TAG (Talk Address Group)** — The talk commands are derived from the primary address by OR'ing the address with #h40. Talk commands are used to address devices to talk.

**SCG (Secondary Command Group)** — Commands in this group provide additional addressing capabilities. Many devices (including the Model 2800) do not use these commands.

## Unaddress commands

The two unaddress commands are used by the controller to remove any talkers or listeners from the bus. ATN is true when these commands are asserted.

**UNL (Unlisten)** — Listeners are placed in the listener idle state by the UNL command.

**UNT (Untalk)** — Any previously commanded talkers will be placed in the talker idle state by the UNT command.

## Common commands

Common commands are commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard.

Generally, these commands are sent as one or more ASCII characters that tell the device to perform a common operation, such as reset. The IEEE-488 bus treats these commands as data in that ATN is false when the commands are transmitted.

## SCPI commands

SCPI commands are commands that are particular to each device on the bus. These commands are designated by the instrument manufacturer and are based on the instrument model defined by the Standard Commands for Programmable Instruments (SCPI) Consortium's SCPI standard.

Generally, these commands are sent as one or more ASCII characters that tell the device to perform a particular operation, such as setting a range or closing a relay. The IEEE-488 bus treats these commands as data in that ATN is false when the commands are transmitted.



## Command codes

Command codes for the various commands that use the data lines are summarized in [Figure D-3](#). Hexadecimal and the decimal values for the various commands are listed in [Table D-2](#).

*Table D-2*  
**Hexadecimal and decimal command codes**

| <b>Command</b> | <b>Hex value</b> | <b>Decimal value</b> |
|----------------|------------------|----------------------|
| GTL            | 01               | 1                    |
| SDC            | 04               | 4                    |
| GET            | 08               | 8                    |
| LLO            | 11               | 17                   |
| DCL            | 14               | 20                   |
| SPE            | 18               | 24                   |
| SPD            | 19               | 25                   |
| LAG            | 20-3F            | 32-63                |
| TAG            | 40-5F            | 64-95                |
| SCG            | 60-7F            | 96-127               |
| UNL            | 3F               | 63                   |
| UNT            | 5F               | 95                   |



## Typical command sequences

For the various multiline commands, a specific bus sequence must take place to properly send the command. In particular, the correct listen address must be sent to the instrument before it will respond to addressed commands. [Table D-3](#) lists a typical bus sequence for sending the addressed multiline commands. In this instance, the SDC command is being sent to the instrument. UNL is generally sent as part of the sequence to ensure that no other active listeners are present. Note that ATN is true for both the listen command and the SDC command byte itself.

*Table D-3*

### ***Typical addressed command sequence***

| Step | Command | ATN state    | Data bus |     |         |
|------|---------|--------------|----------|-----|---------|
|      |         |              | ASCII    | Hex | Decimal |
| 1    | UNL     | Set low      | ?        | 3F  | 63      |
| 2    | LAG*    | Stays low    | <        | 3C  | 60      |
| 3    | SDC     | Stays low    | EOT      | 04  | 4       |
| 4    |         | Returns high |          |     |         |

\*Assumes primary address = 28.

[Table D-4](#) gives a typical common command sequence. In this instance, ATN is true while the instrument is being addressed, but it is set high while sending the common command string.

*Table D-4*

### ***Typical addressed command sequence***

| Step | Command | ATN state  | Data bus |     |         |
|------|---------|------------|----------|-----|---------|
|      |         |            | ASCII    | Hex | Decimal |
| 1    | UNL     | Set low    | ?        | 3F  | 63      |
| 2    | LAG*    | Stays low  | <        | 3C  | 60      |
| 3    | Data    | Set high   | *        | 2A  | 42      |
| 4    | Data    | Stays high | R        | 52  | 82      |
| 5    | Data    | Stays high | S        | 53  | 83      |
| 6    | Data    | Stays high | T        | 54  | 84      |

\*Assumes primary address = 28.

## IEEE command groups

Command groups supported by the Model 2800 are listed in [Table D-5](#). Common commands and SCPI commands are not included in this list.

*Table D-5*

### **IEEE command groups**

|                                |  |
|--------------------------------|--|
| <b>HANDSHAKE COMMAND GROUP</b> |  |
|                                | NDAC = NOT DATA ACCEPTED<br>NRFD = NOT READY FOR DATA<br>DAV = DATA VALID  |
| <b>UNIVERSAL COMMAND GROUP</b> |  |
|                                | ATN = ATTENTION<br>DCL = DEVICE CLEAR<br>IFC = INTERFACE CLEAR<br>REN = REMOTE ENABLE<br>SPD = SERIAL POLL DISABLE<br>SPE = SERIAL POLL ENABLE |
| <b>ADDRESS COMMAND GROUP</b>   |  |
| LISTEN                         | LAG = LISTEN ADDRESS GROUP<br>MLA = MY LISTEN ADDRESS<br>UNL = UNLISTEN  |
| TALK                           | TAG = TALK ADDRESS GROUP<br>MTA = MY TALK ADDRESS<br>UNT = UNTALK<br>OTA = OTHER TALK ADDRESS  |
| <b>ADDRESSED COMMAND GROUP</b> |  |
|                                | ACG = ADDRESSED COMMAND GROUP<br>GTL = GO TO LOCAL<br>SDC = SELECTIVE DEVICE CLEAR   |
| <b>STATUS COMMAND GROUP</b>    |  |
|                                | RQS = REQUEST SERVICE<br>SRQ = SERIAL POLL REQUEST<br>STB = STATUS BYTE<br>EOI = END OR IDENTIFY   |

## Interface function codes

The interface function codes, which are part of the IEEE-488 standards, define an instrument's ability to support various interface functions and should not be confused with programming commands found elsewhere in this manual. The interface function codes for the Model 2800 are listed in [Table D-6](#).

Table D-6

**Model 2800 interface function codes**

| Code | Interface function   |
|------|--|
| SH1  | Source Handshake capability.   |
| AH1  | Acceptor Handshake capability.   |
| T5   | Talker (basic talker, talk-only, serial poll, unaddressed to talk on LAG). |
| L4   | Listener (basic listener, unaddressed to listen on TAG).                   |
| SR1  | Service Request capability.  |
| RL1  | Remote/Local capability.   |
| PP0  | No Parallel Poll capability.   |
| DC1  | Device Clear capability.   |
| DT1  | Device Trigger capability.   |
| C0   | No Controller capability.  |
| E1   | Open collector bus drivers.  |
| TE0  | No Extended Talker capability.   |
| LE0  | No Extended Listener capability.   |

The codes define Model 2800 capabilities as follows:

**SH (Source Handshake Function)** — SH1 defines the ability of the instrument to initiate the transfer of message/data over the data bus.

**AH (Acceptor Handshake Function)** — AH1 defines the ability of the instrument to guarantee proper reception of message/data transmitted over the data bus.

**T (Talker Function)** — The ability of the instrument to send data over the bus to other devices is provided by the T function. Instrument talker capabilities (T5) exist only after the instrument has been addressed to talk.

**L (Listener Function)** — The ability for the instrument to receive device-dependent data over the bus from other devices is provided by the L function. Listener capabilities (L4) of the instrument exist only after it has been addressed to listen.

**SR (Service Request Function)** — SR1 defines the ability of the instrument to request service from the controller.

**RL (Remote-Local Function)** — RL1 defines the ability of the instrument to be placed in the remote or local modes.

**PP (Parallel Poll Function)** — The instrument does not have parallel polling capabilities (PP0).

**DC (Device Clear Function)** — DC1 defines the ability of the instrument to be cleared (initialized).

**DT (Device Trigger Function)** — DT1 defines the ability of the Model 2800 to have readings triggered.

**C (Controller Function)** — The instrument does not have controller capabilities (C0).

**TE (Extended Talker Function)** — The instrument does not have extended talker capabilities (TE0).

**LE (Extended Listener Function)** — The instrument does not have extended listener capabilities (LE0).

**E (Bus Driver Type)** — The instrument has open-collector bus drivers (E1).

---

# Index

## A

ASCII data format [2-8](#)

## B

Bus commands [D-6](#)  
  Address [D-9](#)  
  Addressed multiline [D-8](#)  
  Codes [D-11](#)  
  Common [D-10](#)  
  IEEE command groups [D-14](#)  
  SCPI [D-10](#)  
  Typical command sequences [D-13](#)  
  Unaddress [D-10](#)  
  Uniline [D-7](#)  
  Universal multiline [D-8](#)  
Bus description [D-2](#)  
Bus lines [D-4](#)  
Bus management lines [D-5](#)

## C

CALCulate3 commands [2-2](#)  
Case cover removal [5-5](#)  
Common commands [2-50](#)  
  \*CLS - clear status [2-51](#)  
  \*ESE - standard event enable [2-51](#)  
  \*ESE? - standard event enable query [2-51](#)  
  \*ESR? - event status register query [2-51](#)  
  \*IDN? - identification query [2-51](#)  
  \*OPC - operation complete [2-52](#)  
  \*OPC? - operation completer query [2-52](#)  
  \*RCL - recall [2-52](#)  
  \*RST - reset [2-53](#)  
  \*SAV - save [2-52](#)  
  \*SRE - service request enable query [2-51](#)  
  \*SRE? - service request enable [2-51](#)  
  \*STB? - status byte query [2-51](#)  
  \*TRG - trigger [2-53](#)  
  \*TST? - self-test query [2-54](#)  
  \*WAI - wait-to-continue [2-54](#)  
SPE, SPD (serial polling) [2-57](#)

## D

Data format  
  ASCII [2-8](#)  
  IEEE-754 single precision [2-8](#)  
Data lines [D-5](#)  
Digital board removal [5-8](#)  
Digital I/O [3-35](#)  
Digital inputs [3-35](#)  
Digital outputs [3-36](#)  
  Commands [3-41](#)  
  Controlling relays [3-38](#)  
  Logic control [3-39](#)  
  Setting output lines [3-40](#)  
  Sink mode, controlling external devices [3-37](#)  
  Source mode [3-39](#)  
Disassembly  
  Case cover removal [5-5](#)  
  Digital board removal [5-8](#)  
  DSP module removal [5-7](#)  
  Fan removal [5-9](#)  
  Frequency synthesizer module removal [5-7](#)  
  Front panel disassembly [5-9](#)  
  Handling and cleaning [5-2](#)  
  Local oscillator amplifier removal [5-6](#)  
  Module Locations  
    Bottom module [5-5](#)  
  Module locations [5-4](#)  
    Top module [5-4](#)  
  Power module removal [5-9](#)  
  Power module wire connections (color ID) [5-10](#)  
  Power supply board removal [5-8](#)  
  Power transformer removal [5-9](#)  
  Reference module removal [5-6](#)  
  RF down converter module removal [5-6](#)  
  Static sensitive devices [5-3](#)  
DISPLAY commands [2-4](#)  
DSP module removal [5-7](#)

## E

Error and status messages [1-2](#)  
Error messages [B-2](#)  
Error queue [3-32](#)  
External trigger input [3-6](#)  
External triggering [3-5](#)  
  Connectors [3-5](#)  
  External trigger input [3-6](#)  
  Input pulse specifications [3-6](#)  
  Meter complete output [3-6](#)  
  Output pulse specifications [3-7](#)

## F

- Fan removal [5-9](#)
- FETCH? [2-48](#)
- FORMat commands [2-6](#)
- Frequency synthesizer module removal [5-7](#)
- Front panel disassembly [5-9](#)
- Functionality verification
  - Calculating limits [4-5](#)
  - Connections
    - Noise floor [4-9](#)
    - Power splitter correction factor [4-11](#)
    - Primary, adjacent and alternate channels power [4-13](#)
    - Reference standard frequency [4-10](#)
    - VSWR [4-7](#)
  - Environmental conditions [4-3](#)
  - Line power setting [4-3](#)
  - Noise floor test parameters [4-9](#)
  - Recommended test equipment [4-4](#)
  - Test considerations [4-6](#)
  - Test requirements [4-3](#)
  - Tests
    - Adjacent channel power [4-14](#)
    - Alternate channel power [4-15](#)
    - Noise floor [4-8](#)
    - Primary channel power [4-10](#)
    - Reference standard frequency [4-10](#)
    - VSWR [4-6](#)
  - Warm-up period [4-3](#)

## G

- General bus commands [2-55](#)
  - DCL (device clear) [2-56](#)
  - GET (group execute trigger) [2-57](#)
  - GTL (go to local) [2-56](#)
  - IFC (interface clear) [2-56](#)
  - LLO (local lockout) [2-56](#)
  - REN (remote enable) [2-55](#)
  - SDC (selective device clear) [2-57](#)
- GPIOB
  - Address [1-4](#)
  - Connections [1-4](#), [1-5](#)
  - Connector location [1-6](#)
  - Protocol [1-4](#)
- GPIO configuration [D-3](#)
- GPIO handshake sequence [D-6](#)

- GPIOB protocol [C-1](#)
  - Differences [C-3](#)
  - Message exchange [C-3](#)
  - NRFD hold-off [C-4](#)
  - Selecting [C-2](#)
- GPIOB status indicators
  - LSTN [1-3](#)
  - REM [1-3](#)
  - SRQ [1-3](#)
  - TALK [1-3](#)

## H

- Handling and cleaning [5-2](#)
- Handshake lines [D-5](#)

## I

- Idle [3-2](#)
- IEEE-754 data format [2-9](#)
- Initiate [3-2](#)
- INSTRument commands [2-14](#)
- Interface function codes [D-15](#)

## L

- LOCAL key [1-3](#)
- Local oscillator amplifier removal [5-6](#)

## M

- MEASure? [2-49](#)
- Measurement event register [3-28](#)
- Message exchange protocol [1-15](#)
- Message exchange protocol (MEP) [C-3](#)
- Messages
  - Response [1-14](#)
- Meter complete output [3-6](#)

## O

- Operation event register [3-26](#)
- Output queue [3-32](#)

## P

- Power module removal [5-9](#)
- Power module wire colors [5-10](#)
- Power supply board removal [5-8](#)
- Power transformer removal [5-9](#)
- Programming examples
  - Command error detection [3-22](#)
  - Digital outputs, set [3-41](#)
  - Error queue, read [3-34](#)
  - Registers, program and read [3-32](#)
  - Serial poll (SRQ) [3-22](#)



---

- Programming syntax
  - Case sensitivity [1-10](#)
  - Command words [1-8](#)
  - Long-form and short-form versions [1-10](#)
  - Message exchange protocol [1-15](#)
  - Program messages [1-11](#)
  - Query commands [1-10](#)
  - Response messages [1-14](#)
  - Short-form rules [1-11](#)

## Q

- Questionable event register [3-29](#)
- Queues [3-15](#), [3-32](#)
  - Error queue [3-32](#)
  - Output queue [3-32](#)

## R

- READ? [2-49](#)
- Reference module removal [5-6](#)
- Registers
  - Condition [3-30](#)
  - Event enable [3-31](#)
  - Measurement event [3-27](#)
  - Operation event [3-25](#)
  - Questionable event [3-29](#)
  - Standard event [3-24](#)
- Response messages
  - Multiple [1-14](#)
  - Sending [1-14](#)
  - Terminator (RMT) [1-14](#)
- RF down converter module removal [5-6](#)

## S

- SCPI commands
  - CALCulate3 commands [2-2](#)
  - DISPlay commands [2-4](#)
  - FORMat commands [2-6](#)
  - INSTrument commands [2-14](#)
  - SENSe commands [2-15](#)
  - SOURce2 commands [2-31](#)
  - STATus commands [2-34](#)
  - SYSTem commands [2-42](#)
  - TRACe commands [2-42](#)
  - Trigger commands [2-44](#)
  - UNIT commands [2-47](#)
- SENSe commands [2-15](#)
- Serial polling [3-21](#)
- Service Request Enable Register [3-20](#)
- Signal-oriented commands [2-48](#)
- SOURce2 commands [2-31](#), [3-41](#)
- Specifications [A-1](#)
- SRQ [3-21](#)

- Standard event register [3-24](#)
- Static sensitive devices [5-3](#)
- Status Byte Register [3-19](#)
- STATus commands [2-34](#)
- Status messages [B-2](#)
- Status structure [3-13](#), [3-14](#)
  - Clearing registers and queues [3-15](#)
  - Commands
    - Clear queues [3-16](#)
    - Condition register [3-30](#)
    - Data format [3-18](#)
    - Error queue [3-34](#)
    - Event enable [3-31](#)
    - Event register [3-30](#)
    - Reset registers [3-16](#)
    - Service Request Enable [3-22](#)
    - Status Byte [3-22](#)
  - Event enable registers [3-31](#)
  - Event registers [3-30](#)
  - Measurement event register [3-27](#), [3-28](#)
  - Operation event register [3-25](#), [3-26](#)
  - Programming enable registers [3-16](#)
  - Questionable event register [3-29](#)
  - Queues [3-15](#), [3-32](#)
  - Reading registers [3-18](#)
  - Register sets [3-15](#)
  - Serial polling [3-21](#)
  - Service request (SRQ) [3-19](#)
  - Service Request Enable Register [3-20](#)
  - SPE, SPD (serial polling) [3-21](#)
  - SRQ [3-13](#)
  - Standard event register [3-24](#)
  - Status byte [3-13](#), [3-19](#)
  - Status Byte Register [3-19](#)
- SYSTem commands [2-37](#)

## T

- TRACe commands [2-42](#)
- Trigger commands [2-44](#)
- Trigger model [3-2](#)
  - Idle and initiate [3-2](#)
  - Operation [3-4](#)
- Triggering [3-2](#)
  - Commands [3-9](#)

## U

- UNIT commands [2-47](#)



# Service Form

**Model No.** \_\_\_\_\_ **Serial No.** \_\_\_\_\_ **Date** \_\_\_\_\_

**Name and Telephone No.** \_\_\_\_\_

**Company** \_\_\_\_\_

List all control settings, describe problem and check boxes that apply to problem. \_\_\_\_\_

Intermittent                       Analog output follows display                       Particular range or function bad; specify \_\_\_\_\_

IEEE failure                       Obvious problem on power-up                       Batteries and fuses are OK

Front panel operational     All ranges or functions are bad                       Checked all cables

Display or output (check one)

Drifts                                       Unable to zero                                       Unstable

Overload                                       Will not read applied input

Calibration only                       Certificate of calibration required                       Data required

(attach any additional sheets as necessary)

Show a block diagram of your measurement including all instruments connected (whether power is turned on or not). Also, describe signal source.

Where is the measurement being performed? (factory, controlled laboratory, out-of-doors, etc.) \_\_\_\_\_

What power line voltage is used? \_\_\_\_\_ Ambient temperature? \_\_\_\_\_ °F

Relative humidity? \_\_\_\_\_ Other? \_\_\_\_\_

Any additional information. (If special modifications have been made by the user, please describe.)

Be sure to include your name and phone number on this service form.

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.

**KEITHLEY**

**Keithley Instruments, Inc.**

28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168

**1-888-KEITHLEY (534-8453) • [www.keithley.com](http://www.keithley.com)**

**Sales Offices:**

**BELGIUM:** Bergensesteenweg 709 • B-1600 Sint-Pieters-Leeuw • 02-363 00 40 • Fax: 02/363 00 64

**CHINA:** Yuan Chen Xin Building, Room 705 • 12 Yumin Road, Dewai, Madian • Beijing 100029 • 8610-8225-1886 • Fax: 8610-8225-1892

**FINLAND:** Tietäjantie 2 • 02130 Espoo • Phone: 09-54 75 08 10 • Fax: 09-25 10 51 00

**FRANCE:** 3, allée des Garays • 91127 Palaiseau Cédex • 01-64 53 20 20 • Fax: 01-60 11 77 26

**GERMANY:** Landsberger Strasse 65 • 82110 Germering • 089/84 93 07-40 • Fax: 089/84 93 07-34

**GREAT BRITAIN:** Unit 2 Commerce Park, Brunel Road • Theale • Berkshire RG7 4AB • 0118 929 7500 • Fax: 0118 929 7519

**INDIA:** 1/5 Eagles Street • Langford Town • Bangalore 560 025 • 080 212 8027 • Fax: 080 212 8005

**ITALY:** Viale San Gimignano, 38 • 20146 Milano • 02-48 39 16 01 • Fax: 02-48 30 22 74

**JAPAN:** New Pier Takeshiba North Tower 13F • 11-1, Kaigan 1-chome • Minato-ku, Tokyo 105-0022 • 81-3-5733-7555 • Fax: 81-3-5733-7556

**KOREA:** 2FL., URI Building • 2-14 Yangjae-Dong • Seocho-Gu, Seoul 137-888 • 82-2-574-7778 • Fax: 82-2-574-7838

**NETHERLANDS:** Postbus 559 • 4200 AN Gorinchem • 0183-635333 • Fax: 0183-630821

**SWEDEN:** c/o Regus Business Centre • Frosundaviks Allé 15, 4tr • 169 70 Solna • 08-509 04 600 • Fax: 08-655 26 10

**TAIWAN:** 13F-3. No. 6, Lane 99 Pu-Ding Road • Hsinchu, Taiwan, R.O.C. • 886-3-572-9077 • Fax: 886-3-572-9031