# $I_{DDQ}$ Testing and Standby Current Testing with the Model 2400 SourceMeter® Instrument

## Introduction

Manufacturers of CMOS integrated circuits and battery-powered electronic products need to measure the quiescent (or "standby") power supply current to verify quality during production testing. The process of measuring the leakage currents of CMOS integrated circuits or finished products that contain CMOS ICs is known as "$I_{DDQ}$" testing. This test requires measuring the current of the $V_{DD}$ power supply while the IC is in the quiescent state. It is done to check for shorted gate oxide and other IC defects that may cause a failure over time. Similarly, the power supply current of battery-powered products that contain bipolar transistors or other ICs can be measured while these ICs are in a quiescent mode. These types of products include portable battery-powered consumer electronics, such as cellular phones, pagers, and notebook computers, as well as implantable medical devices, such as pacemakers and defibrillators. The goal of these tests is to ensure the products satisfy the consumer demand for longer operating periods from a given level of battery charge as well as operational quality. Testing must be performed as quickly as possible to ensure acceptable throughput, but also thoroughly to assure product quality.

When choosing a measurement instrument to perform these tests, two of the most important considerations are speed and accuracy. However, when measuring small currents, a trade-off sometimes must be made between speed and accuracy, so custom hardware is often designed to perform these tests. Custom hardware may require lengthy design time and is not always easy to maintain, whereas commercially-available test systems are typically easy to use, readily available, and economical in terms of rack space. This application note describes how to configure a quiescent current measurement system using the Model 2400 SourceMeter® instrument to perform the functions of sourcing voltage and measuring current. An example program for the Model 2400 is provided that can perform roughly 2500 $I_{DDQ}$ measurements per second. Also, a program to test the current drain of a cellular phone is available via Keithley's web site.

## $I_{DDQ}$ Testing of CMOS ICs

### Test Description

This test involves measuring the current of the $V_{DD}$ power supply of a CMOS IC when the inputs are at either $V_{DD}$ or $V_{SS}$ and the outputs are not connected. *Figure 1* is a diagram of a test set-up for a single CMOS inverter. In this example, the Model 2400 is used to source the supply voltage ($V_{DD}$) and to measure the resulting quiescent current. ($I_{SSQ}$ testing can be performed in a similar manner by measuring the $V_{SS}$ power supply current.)
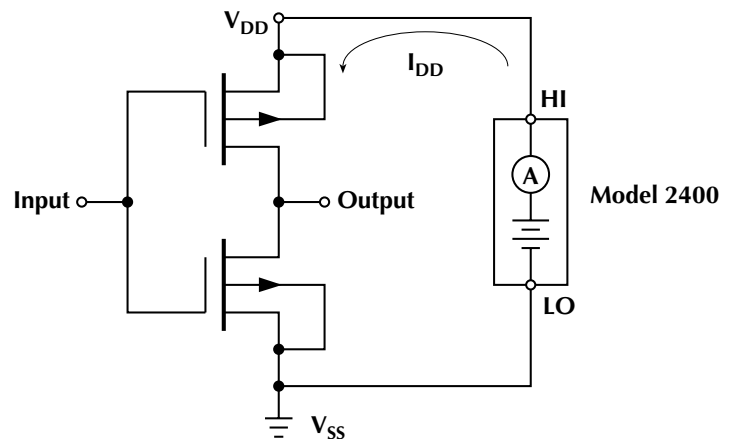


*Figure 1. Measuring the Quiescent Current of a Single CMOS Inverter*

While this example shows an IC with a single gate, many ICs have thousands of gates. Usually, a predetermined series of test vectors (i.e., a pattern of logical ones and zeros applied to the inputs) is used to reduce the number of quiescent current measurements that must be made to ensure that all the gates are toggled or the desired IC logic states are tested.

To make the measurements, a test vector is applied to the inputs of the IC; after a specified settling time, the resulting current is measured. To keep the IC in a functional mode, a constant voltage is applied to the $V_{DD}$ pin(s) of the IC throughout the test. After a measurement is taken, it is compared to a preset threshold to determine if the part passes or fails. This threshold is often set at the $\mu A$ or $nA$ level and is usually determined by a statistical analysis of $I_{DDQ}$ for several good ICs. A good CMOS has high current only when switching. Depending on the type of defect involved, the $I_{DDQ}$ of a bad IC will be much higher.

## Test System Configuration

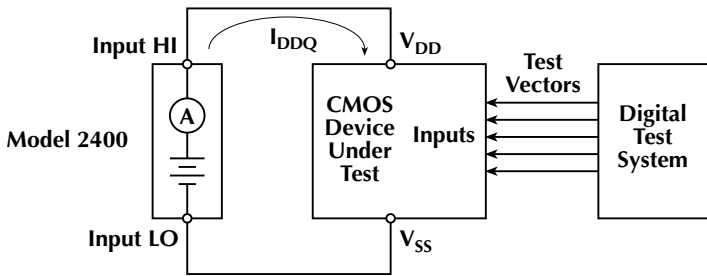*Figure 2* is a 2400-based system for testing the $I_{DDQ}$ for CMOS ICs.

As *Figure 2* shows, the HI and LO terminals of the 2400 are connected to the $V_{DD}$ and $V_{SS}$ terminals of the CMOS IC. The 2400 supplies a constant DC voltage to the IC throughout the test. The inputs of the IC are connected to a digital test system that ensures all the gates are toggled or the desired logic states are achieved.

To maximize speed, the 2400 is triggered externally by the digital test system after a vector has been sent to the CMOS IC. After the $I_{DDQ}$ value is compared to the threshold level, the 2400 returns a trigger to the digital test system, which generates another test vector. This process is repeated until all test vectors are generated or the IC fails the test.

***Example 1.*** Programming the Model 2400 to perform an $I_{DDQ}$ test rapidly.

***Application:*** In this example, 2000 test vectors will be sent to a CMOS IC that must be powered with a constant 2V. A quiescent current of 1µA or less will be considered an acceptable current. The actual measurements need not be retained. To ensure acceptable production throughput, measurements must be made as quickly as possible, preferably in one second or less. The measurement instrument must send a pass or fail indication to the digital test system.

***Solution:*** To meet the time limitation, the 2400 must be programmed to run as fast as possible, which requires disabling features that slow its operation. The programmable source and trigger delays must be set to 0 seconds. Autoranging, auto zeroing, the front panel display, filters, and measurement functions must be turned off. To reduce test time further, the 2400 can be set up as a programmable comparator, rather than taking actual measurements.

Next, a pass/fail limit is established. Via the digital I/O port, the 2400 will output a specified bit pattern to the digital test system if it goes into compliance (in other words, if the 2400's output current reaches the programmed compliance or limit, which, in this case, occurs at 1µA). If the IC passes (i.e., the 2400 does not go into compliance), a different bit pattern is sent. The

2400 only detects whether or not it is in compliance and does not take any readings. The 2400 is programmed to generate a service request (SRQ) if it goes into compliance or when the specified number of test vectors (in this case, 2000) is completed. On compliance, a fail bit pattern is sent to the digital tester and the program aborts.

The following program performs the test. As written, it enables the 2400 to perform the $I_{DDQ}$ test at a sample rate of about 2.5kHz. The timing may vary, depending on other instrumentation in the system.

```
'$INCLUDE: 'ieeeqb.bi'
' program name is iddq1.bas
' This program uses Microsoft QuickBasic 4.5 and the KPC-488.2AT IEEE
' Interface Card
CALL initialize(21, 0)
CALL transmit("UNT UNL MTA LISTEN 24 SDC UNL UNT", gpib.status%)
CALL send(24, "*RST", gpib.status%)
A$ = SPACE$(80)
REM*****global paramters
CALL send(24, ":SYST:AZERO 0", gpib.status%) 'auto zero off
CALL send(24, ":SYST:RSEN 0", gpib.status%) 'local sense
CALL send(24, ":DISP:ENABLE OFF", gpib.status%) 'turn display off
REM******set source
CALL send(24, ":SOURCE:FUNC:MODE VOLT", gpib.status%)
CALL send(24, ":SOURCE:VOLT:RANG 10", gpib.status%)
CALL send(24, ":SOURCE:VOLT 2", gpib.status%)
CALL send(24, ":SOURCE:VOLT:MODE FIXED", gpib.status%)
CALL send(24, ":SOURCE:DELAY 0", gpib.status%)
REM****set compliance
CALL send(24, ":SENS:FUNC:OFF:ALL", gpib.status%)
CALL send(24, ":CURR:RANG 1E-6;NPLC 0.01;PROT 1E-6", gpib.status%)
REM*******set up limit structure
CALL send(24, ":CALC2:LIM:COMP:FAIL IN", gpib.status%)
CALL send(24, ":CALC2:LIM:COMP:SOURce2 13", gpib.status%)
CALL send(24, ":CALC2:LIM:STATE ON", gpib.status%)
REM*****set trigger
CALL send(24, "ARM:COUNT 1", gpib.status%)
CALL send(24, "TRIG:COUN 2000", gpib.status%)
CALL send(24, "TRIG:SOURCE TLINK; ILIN 2; OLIN 1;OUTP SENS",
          gpib.status%)
CALL send(24, "TRIG:DELAY 0", gpib.status%)
REM*****set up SRQ on compliance fail or test done (idle)
CALL send(24, "STATUS:MEAS:ENABLE 1", gpib.status%)
CALL send(24, "STATUS:OPER:ENABLE 1024", gpib.status%)
CALL send(24, "*SRE 129", gpib.status%) 'MSB (1) and OSB (128)
CALL send(24, "OUTP ON;*OPC?", gpib.status%)
CALL enter(A$, I%, 24, gpib.status%)
IF gpib.status% THEN STOP
CALL send(24, "INIT", gpib.status%)
DO: LOOP UNTIL srq% <> 0
' Abort test if in compliance
```

```
CALL send(24, "ABORT", gpib.status%)
' Clear SRQ event
CALL spoll(24, poll%, gpib.status%)
CALL send(24, "STAT:MEAS?;OPER?", gpib.status%)
CALL enter(A$, l%, 24, gpib.status%)
CALL send(24, ":OUTP OFF", gpib.status%)
END
```

# Standby Current Testing

## Test Description

This test involves measuring the current of a battery-powered product while it is in a standby state. A typical test set-up is shown in *Figure 3*.
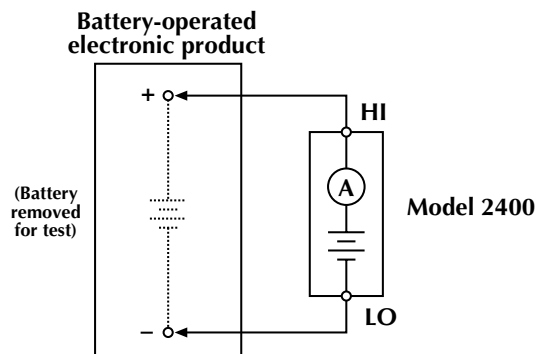


*Figure 3. Testing the Off Drain Current of a Battery-Powered Electronic Product*

In this example, the HI and LO terminals of the 2400 are connected to the HI and LO battery terminals of the product to be tested. The 2400's voltage source simulates the internal battery of the product and its ammeter measures the current while the part is in a standby mode. This measurement is compared to a pre-specified limit to determine if the product passes or fails. An acceptable current is often in the mA or μA range, but can vary greatly depending on the application.

## Test System Configuration

*Figure 4* is a block diagram of a 2400-based standby current production test system.

The battery-powered product is placed in a test fixture connected to the 2400. When triggered, the 2400 outputs a voltage and measures the resulting current. As *Figure 4* illustrates, the Model 2400 has both an IEEE-488 port and a digital I/O port. The digital I/O port can send signals directly to and receive signals from an automatic parts handler. The measured current of the DUT is compared to limit values that have been preset in the 2400. A TTL level signal generated from the 2400's

digital I/O port indicates whether the device passed or failed the tests. Based on the received signal, the automated handler will route the device to the appropriate bin or other location.
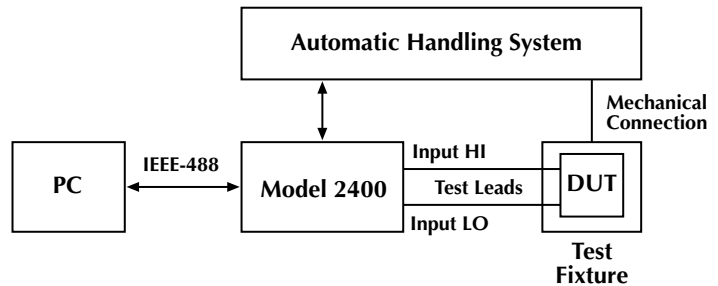


*Figure 4. 2400-Based Production Test System to Measure Current of Battery Powered Electronic Devices*

*Example 2.* Testing the Current of a Cellular Phone

In this application, the 2400 takes the place of the cellular phone's rechargeable battery and measures the current while the phone is in the "talk," "standby," and "off" modes. The test voltage remains the same for all three measurements. For this test, a cellular phone fails if its measured current exceeds the threshold levels for the given mode:

| Mode | Threshold Current | Bit Patterns | |
|------|------|------|------|
| | | Pass | Fail |
| **Talk** | 400mA | 0001 | 0010 |
| **Standby** | 10mA | 0001 | 0110 |
| **Off** | 100μA | 0001 | 1000 |

Each threshold current has its corresponding pass and fail bit patterns. The 2400 has four digital output lines that can be used to set up bit patterns to produce TTL level signals for controlling external equipment, such as an automated parts handler. The most efficient way to execute this test is to use the 2400's Source Memory, which makes it possible to store up to 100 complete test set-ups that can be initiated over the IEEE bus with a single command. When these tests are performed, each threshold current level and corresponding bit pattern is configured and stored in a separate Source Memory location. The Source Memory can be programmed to run the entire test or stop after a failure. Setting up the Source Memory is illustrated in the IDDQ2.bas example program.

## IDDQ2.bas Example Program

This example program has been developed to test the current of a cellular phone while in the "talk," "standby," and "off" modes. The program instructs the 2400 to output a constant 4.5 volts and then triggers it to test for current compliance levels of 400mA, 10mA and 100μA. The pass/fail bit patterns are the same as those in the previous example. At the end of the test, the three current measurements are displayed on the computer screen.

To obtain a digital file of the Example Program, access Keithley's World Wide Web site (http://www.keithley.com). NOTE: The test program may require some modifications, depending on the type of parts handler used and its specific timing requirements, the manufacturer of the IEEE-488 card, the type of computer controller used, and voltage and current limits.

## Equipment List

The following equipment is required to assemble the cellular phone test system and run the IDDQ2.bas example program:

1. Keithley Model 2400 SourceMeter Instrument.
2. PC with Keithley Model KPC-488.2 Interface Card.
3. Automatic Handler System with test fixture.
4. Keithley Model 7007-2 IEEE-488 Interface Cable.
5. Custom DB-9 digital I/O handler interface cable (to interface the 2400 to the handler system).
6. Test leads to connect the 2400 to the test fixture.

## Alternative Solutions

The 2400 current meter can measure current from 1A full scale to 1µA full scale with 10ppm resolution on any range. If greater current resolution is desired, an instrument with a more sensitive ammeter, such as the Model 236 or 238 Source-Measure Unit, must be used. However, due to the greater sensitivity of these instruments, the test will run more slowly than when the 2400 is used. If currents greater than 1A must be measured the Model 2420 3A SourceMeter instrument or the Model 2440 5A SourceMeter Instrument can be used.

Another alternative solution is to use a readback power supply for both sourcing voltage and measuring current. However, the readback resolution of most programmable sources is in the mA range, so they cannot be used for low current tests. In addition, these supplies often have sense lines with relatively low input impedance and correspondingly high current (µA range), which can affect measurement accuracy.

## For Further Reading

D. Leslie, "QTAG: The Evolution of a Standard Monitor: A Progress Report," *Evaluation Engineering*, pp 26-32, Oct. 1995.

M.J. Riezenman, "Technology 1996: Test & Measurement," *IEEE Spectrum*, pp. 65-69, Jan. 1996.

J.M. Soden and C.F. Hawkins, "$I_{DDQ}$ Testing and Defect Classes — A Tutorial," in *Proc. of Custom Integrated Circuits Conf*, 1995, pp. 633-642.

## Acknowledgments

The author gratefully acknowledges the advice and assistance of Jerry M. Soden of Sandia National Laboratories, Albuquerque, New Mexico.

**KEITHLEY**