

用2600系列数字源表进行 I_{DDQ} 测试和待机电流测试

引言

CMOS集成电路（CMOS IC）和电池供电产品的制造商需要测量静态（或“待机”）电源电流用于验证生产测试质量。CMOS IC或其中含有CMOS IC成品的漏电电流测量过程被称为 I_{DDQ} 测试。此测试要求在IC处于静态条件下测量 V_{DD} 电源电流。测试的目的是检查栅氧化层短路及可能随时间导致IC失效的其它缺陷。同样地，带有双极晶体管的电池供电产品或其它IC的电源电流也可以在静态模式下测量。这些产品类型包括便携式电池供电的消费性电子产品，例如移动电话、寻呼机和笔记本电脑，以及可植入的医疗设备，例如心脏起搏器和除颤器。这些测试的目标是在给定电池充电电平和在工作质量条件下确保产品满足消费者对更长工作时间的要求。测试必须尽快完成以确保吞吐量符合要求，而且测试必须全面以确保产品质量。

在选择执行这些测试的测量仪器时，两项最重要的考虑是速度和准确度。但是，在测量小电流时，有时必须在速度和准确度两者间做出取舍，所以通常需要用定制硬件完成这些测试。定制硬件的设计时间可能很漫长而且一般不容易维护，而商购的测试系统通常易于使用、容易买到而且节省机架空间。

另一篇吉时利应用笔记（#804）介绍了如何用2400系列源表执行 I_{DDQ} 测试和静态电流测量。本应用笔记介绍了如何用吉时利最新的2600系列源表实现上述测试。这种新一代源表的产品系列包括单通道2601和双通道2602。由于具有内建测试脚本处理器（TSP™）和新的仪器间通信接口（TSP-Link™），2600系列仪器比前几代源表的功能更强大和更灵活。吉时利提供了每秒能执行约2500次 I_{DDQ} 测量的测试脚本实例。此脚本电子版适合于2600系列的两个型号，而且可以从吉时利网站（www.keithley.com）下载。测试移动电话漏电流的测试脚本也可以在吉时利网站下载。

CMOS IC的 I_{DDQ} 测试

测试介绍

此测试涉及在输入端处于 V_{DD} 或 V_{SS} 并且输出端未连接时，测量CMOS IC的 V_{DD} 电源耗电流。图1是一个CMOS反相器的测试设置框图。在这个例子中，2601/2602用于源电压（ V_{DD} ）并测量产生的静态电流。

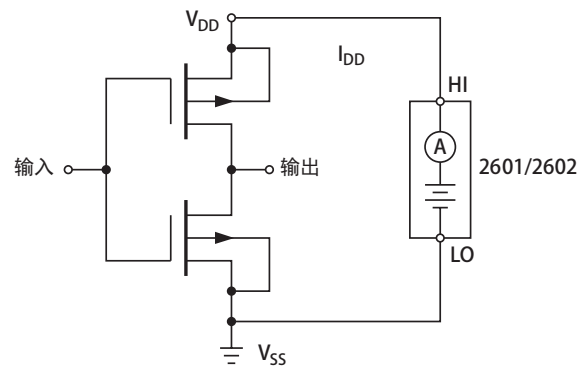


图1. 测量一个CMOS反相器的静态电流

虽然这个例子示出的IC只有一个栅极，但是许多IC具有数以千计的栅极。因而，通常要使用预定的一系列测试向量（即施加至输入端的逻辑1和0的模式）减少静态电流的测量次数，并且必须保证全部栅极都被切换或者所需IC逻辑状态都经过测试。

在整个测试中对IC的 V_{DD} 引脚施加恒定电压使IC保持在工作状态。一个好的CMOS器件仅在开关时从 V_{DD} 电源消耗大电流；在静态条件下的耗电极低。与涉及的缺陷类型有关，一颗有瑕疵IC的 I_{DDQ} 将会高得多。测量时，将测试向量施加至IC输入端，然后经过规定的建立时间后，测量产生的电流。完成测量后，将测量电流与预设阈值相比较以确定器件通过还是失效。此阈值通常设定为微安或纳安级而且通常由多颗完好IC的 I_{DDQ} 统计分析确定。随着器件变得越来越复杂， I_{DDQ} 测试不能总是用简单阈值测试执行。在某些情况下，必须对被测器件（DUT）进行 I_{DDQ} 数据统计分析以便可靠地确定通过/失效状态。2600系列源表非常适于这两种测试方案。

测试系统配置

图2是面向CMOS IC，基于2600系列的 I_{DDQ} 测试系统。

如图2所示，260X的HI和LO端子连至CMOS IC的 V_{DD} 和 V_{SS} 端子。在整个测试过程中，260X为IC提供恒定的直流电压。IC的输入端连至“数字测试系统”，这确保切换了全部栅极或实现了要求的逻辑状态。假设此测试系统还控制着机械位置、DUT探测和处置好/坏器件。

对260X的控制可以像标准可编程仪器那样，通过IEEE-488总线（GPIB）或者RS-232发送独立指令实现。260X上的这两种通信接口都是标准的。但是，为实现最大吞吐量，可以将完整的测试脚本下载至仪器的测试脚本处理器，然后执行几乎独立于PC主机（系统控制器）的全部测试。当260X通过GPIB连接至主控制器时，它实际上能通过其RS-232端口控制另一台仪器。因此，在适当情况下，260X能发送ASCII命令字符串至数字控制系统并从数字控制系统接收数据。

为进一步提升速度，外部硬件触发器用于同步 I_{DDQ} 测量和测试向量的使用。260X配备的14条数字输入/输出线路能用于数字控制（在此例中为通过/失效状态）或用作输入或输出触发线路。当向量被发送至CMOS IC时，数字测试系统触发260X。当 I_{DDQ} 值被评估完后，260X返回一个触发信号至数字测试系统，数字测试系统生成另一个测试向量。此过程重复进行直至产生全部测试向量或者IC未能通过测试。测试完成后，260X向其数字I/O（DIO）端口写入预先确定的位模式用于向数字测试系统指示器件的通过/失效状态。

当 I_{DDQ} 测试的通过/失效状态仅通过源电流与阈值电平的比较来确定时，260X至少有两种方法完成这种测量和检查。如果需要 I_{DDQ} 的实际值，那么260X能测量电流并将测量值与阈值进行比较。如果电流超出阈值电平，那

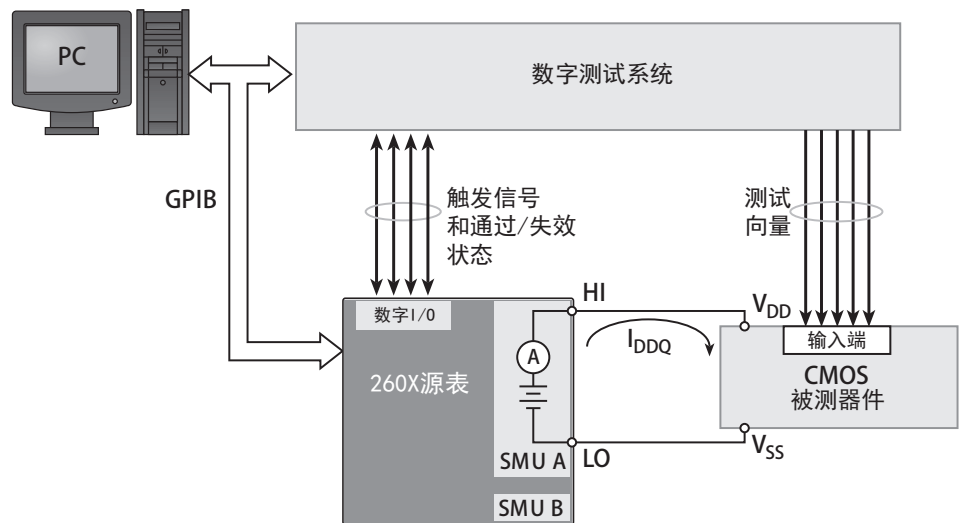


图2. I_{DDQ} 测试系统配置

么测试失败；否则，测试通过。260X能按要求将任意或全部测量值以及通过/失效状态返回至PC主机。如果无需 I_{DDQ} 的实际值，那么260X能配置为数字比较器以实现更高的测试吞吐量。将260X的电流箝位极限设为阈值。施加测试向量并确定260X的箝位状态。如果耗电电流试图超出此极限，那么260X将“进入箝位”并将电流箝在此极限。当发生此情况时， I_{DDQ} 测试失败。如果电流未超出此阈值，那么设备将不会进入箝位状态并且测试通过。由于无需用测量确定仪器的箝位状态，所以后面的方法通常比前面的快。

如前所述，复杂器件的 I_{DDQ} 测试不能总用简单阈值测试来实现。在某些情况下，必须对被测器件（DUT）进行 I_{DDQ} 数据统计分析以便可靠地确定通过/失效状态。在从260X获取全部测试数据后，PC主机就能进行数据分析。但数据传输的过程相当慢，会显著影响测试吞吐量。如果不需要保存读数，那么数据传输的代价太大了。用于设置测试脚本处理器的测试脚本语言包括数学库及其它功能，能在仪器中进行大量分析，从而无需传输全部数据。260X的深存储器进一步提供了便利。每条SMU通道的两个非易失缓冲器能保存多达100,000个读数。易失存储器能用于更多数据的存储。

远距离操作和测试脚本处理器的使用

在进入实际测试实例之前，我们先讨论用于此测试应用的新仪器特性。2600系列源表具有强大的嵌入式计算机或测试脚本处理器，因而能实现在堆架式仪器中从未见过的功能。可以将完整的测试程序（脚本）下载至TSP。与其它常见的编程语言一样，一个精心设计的脚本创建了能用测试程序或其它函数调用的可重用函数或子程序。可以将参数传至这些函数。在这里示出的 I_{DDQ} 测试实例中，创建一个函数执行 I_{DDQ} 测试，检查数据并返回测试结果。而且，此函数能处理全部触发同步以及介于数字测试系统和260X之间的其它数字I/O。此函数能通过系统控制PC调用或通过TSP的其它测试函数调用。创建此函数的脚本须用GPIB或RS-232下载至260X。当脚本首次下载后，将被保存在易失存储器中；但如果需要，也可以保存至非易失存储器。必须运行脚本才能创建函数。此函数总是在易失存储器中，这意味着在每一个电源周期都必须重新创建函数。创建的脚本可以在任何时候显示地运行或者设定为上电时自动运行。当系统控制PC调用此函数时，260X将对指定数量的器件执行完整的测试序列，无需系统控制器更多地干预，因此节省了通讯时间并提高了系统吞吐量。

可以使用任何文本编辑器创建脚本。但是，吉时利提供的一款称作测试脚本生成器的免费应用程序能创建、调试和组织脚本。测试脚本生成器可以将脚本下载至仪器的易失存储器或保存到仪器的非易失存储器中。而且，测试脚本生成器能运行脚本。脚本还可以用其它语言（例如Visual Basic®、Visual C/C++®或LabVIEW®）创建的应用程序装载和运行。脚本保存在存储器以后，还可以从前面板运行。

2602能完成全部的测量、计算和检查，因而无需将数据发送至主计算机（系统控制器）处理。但如需保存记录或为了其它目的，也可将数据发送至主计算机。如脚本实例所示，“print（打印）”语句用于将数据发回主计算机。打印语句列出的数据放在仪器的输出队列以供主机读取。

应用举例1：I_{DDQ}测试

在此例中，将2000个测试向量发送至必须用恒定2V加电的CMOS IC。此例将使用一个阈值电平，而且1 μA及以下的静态电流将被认为是合格电流。无需保存实际的测量值，所以将只检查260X的箝位状态。为确保生产吞吐量符合要求，必须尽快完成测量，最好在1秒以内。测量仪器必须发送一个通过/失效指示至数字测试系统。

方案：开发TSP脚本用于创建“IddqTest(smu, ndevices, nvectors)函数”。此脚本的一部分被列在下面。完整的脚本可以在吉时利网站（www.keithley.com）下载。此脚本可以用测试脚本生成器查看、编辑、装载和运行。由于无法获得实际的“数字测试系统”，因而使用I_{DDQ}测量与测试向量应用同步的触发线路进行环绕测试以评估函数的速度性能。260X的输入触发线路连至其输出触发线路。在此条件下，260X能以每秒约2500个向量的速率执行I_{DDQ}测试。

函数定义如下。除了这里示出的以外，实际脚本包括更多注释。注释用双横线（--）标识。函数中声明了几个局部变量。所有变量都是全局的，除非显式声明为局部变量。此函数在执行实际测试前要进行一些仪器初始化设置。此设置包括设定2V源电平和1 μA箝位极限，选择电压感测模式以及配置触发线路。例子中使用了本地（2线）电压感测。可以直接将其改为远端（4线）感测。

```
function IddqTest(smu, ndevices, nvectors)
-- 传递参数:
-- 源测量单元是用于测试的SMU (A或B)
-- ndevices是待测IC的数量
-- nvectors是Iddq测试序列用到的向量数量

-- 默认smua, 如果未指定SMU.
if smu == nil then smu = smua end

-- ***** 声明并初始化临时变量 *****

-- 保存“等待触发信号”布尔状态的变量
local l_sot_received
local l_trig_received

-- 放弃测试标记（布尔型）
local l_abort_test

-- 保存定时信息的变量
local l_start_time, l_stop_time, l_elapsed_time

-- 计数器变量
local l_i, l_nvectors_remaining

-- 用于模拟器件机械手分级的表格
local l_bins = {0,0} -- 全部分级初始设置为0

-- ***** 执行260X的初始化设置 *****

smu.reset() -- 将SMU重置为缺省设置
smu.source.func = smu.OUTPUT_DCVOLTS -- 源DCV
smu.source.rangev = 2 -- 将自动选择6V量程
smu.source.levelv = 2 -- 源2V
smu.source.limiti = 1E-6 -- 电流箝位设为1uA
smu.sense = smu.SENSE_LOCAL -- 用smu.SENSE_REMOTE进行4线感测
```

```

-- 配置数字I/O端口

digio.writeprotect = 0          -- 全部比特撤销保护
digio.writeport(30)           -- 将2、3、4和5比特/行设置为高
digio.writeprotect = 30       -- 触发线路2、3、4和5写保护

-- 配置触发线路2（输入SOT）
digio.trigger[2].mode = digio.TRIG_FALLING    -- 检测下降沿
digio.trigger[2].clear()                     -- 清除“锁存”触发信号

-- 配置触发线路3（输出EOT）
digio.trigger[3].mode = digio.TRIG_FALLING    -- 输出TTL低电平脉冲
digio.trigger[3].pulsewidth = 10E-6          -- 确保最小脉冲

-- 配置触发线路4（在设置了测试向量时输入TRIG）
digio.trigger[4].mode = digio.TRIG_FALLING    -- 检测下降沿
digio.trigger[4].clear()                     -- 清除“锁存”触发信号

-- 配置触发线路5（在完成IDDQ测量后输出TRIG）
digio.trigger[5].mode = digio.TRIG_FALLING    -- 输出TTL低电平脉冲
digio.trigger[5].pulsewidth = 10E-6          -- 确保最小脉冲

-- 清除错误队列
errorqueue.clear()

-- ***** 执行测试 *****

-- 在前面板显示一些状态信息
display.clear()
display.setCursor (1,1)
display.setText(" Test In Progress")
display.setCursor (2,1)
display.setText(" Testing " .. tostring(ndevices) .. " Parts")
timer.reset()
l_start_time = timer.measure.t()

for l_i = 1, ndevices do

    -- 等待SOT指示器件准备好进行测试；10ms后超时
    l_sot_received = digio.trigger[2].wait(0.01)

    -- 打开SMU输出；等到测试完成
    smu.source.output = smu.OUTPUT_ON

    -- 初始化中断标记和向量计数器
    l_abort_test = false
    l_nvectors_remaining = nvectors

    digio.trigger[4].clear()    -- 清除全部“锁存”触发信号

    -- 重复测试直至不再有测试向量或测试失败
    while (l_nvectors_remaining > 0) and not (l_abort_test) do

        -- 等待来自数字测试系统的触发信号；10ms后超时
        l_trig_received = digio.trigger[4].wait(10E-3)

        -- 如需建立时间，delay(0.0005)在这里插入延时

        -- 检查箝位状态；返回布尔值真或假
        l_incompliance = smu.source.compliance

        -- 如果源在箝位状态，那么器件损坏并且中断测试
        if l_incompliance then

            digio.writeport(32) --写失效模式至DIO
            l_bins[2] = l_bins[2]+1    -- “分级”器件
            l_abort_test = true -- 将中断测试标记置为真并退出循环

        else    -- 否则，器件仍然是好的，所以继续测试

            digio.trigger[5].assert()    -- 输出触发信号至“数字测试系统”
            l_nvectors_remaining = l_nvectors_remaining - 1    -- 新计数

        end --if语句
    end
end

```



```

end --while语句
-- 如果测试不中断，那么器件是好的
if not (l_abort_test) then

    digio.writeport(64)          -- 写通过模式至DIO
    l_bins[1] = l_bins[1]+1     -- “分级”器件

end --if语句

--关闭SMU输出
smu.source.output = smu.OUTPUT_OFF

-- 输出EOT触发信号
digio.trigger[3].assert()

-- 清除分级码（将全部撤销保护的比特置为零）
delay(0.0001)                  -- 清除分级码之前延迟的单位为秒
digio.writeport(0)

end --for语句

l_stop_time = timer.measure.t()
l_elapsed_time = l_stop_time - l_start_time

-- 在260X前面板显示吞吐率和最终分级结果
display.clear()
display.setCursor (1,1)
display.setText("Parts per sec = "..tostring(ndevices / l_elapsed_time))
display.setCursor(2,1)
display.setText("Bin Count: Good= "..l_bins[1].." Bad= "..l_bins[2])

-- 将速度与分级结果写至输出队列以供PC主机读取
print("Elapsed time = "..l_elapsed_time.." sec")
print("Parts per sec = "..tostring(ndevices / l_elapsed_time))
print("Bin Count: Good Parts: "..l_bins[1].." Bad Parts: "..l_bins[2])

```

end --IddqTest函数

用测试脚本生成器或者其它应用执行示例的IDDQ测试脚本仅创建了函数，不执行任何测试。执行IDDQ测试需要调用IddqTest()函数。例如，为了用2000个测试向量测试100个器件，系统控制器必须发送指令“IddqTest(smua, 100, 2000)”。作为对函数调用的响应，260X等待外部数字测试系统对每个DUT的开始测试(SOT)触发信号。260X收到SOT信号后，打开SMU输出并等待数字测试系统的触发信号。数字测试系统向IC输入端施加测试向量，然后发送一个触发信号至260X。260X接到触发信号后，等待一个预定的建立时间，然后检查其箝位状态。如果不在箝位状态，那么这项IDDQ测试通过，而且260X输出一个触发信号至数字测试系统并进入循环等着下一个施加的测试向量。此过程持续进行直至完成全部测试向量或者260X进入箝位状态。如果成功通过全部测试向量，260X就向其DIO端口写入十进制数64表明通过了数字测试系统。表元素“l_bins[1]”加1用于模拟器件的分级。然后，260X输出测试结束(EOT)触发信号至数字测试系统，表明已完成DUT的IDDQ测试序列。

如果260X进入箝位状态，那么IDDQ测试失败。在单项测试失败时，260X采用“立即”分级方案，这意味着260X会立即向数字测试系统写一个失效位模式（十进制数32），中断

剩余的测试序列，然后输出EOT触发信号至数字测试系统。在输出了失效模式后，表元素“l_bins[2]”加1用于模拟分级过程。在260X通过写十进制数0至DIO端口来发送EOT触发信号后，260X等待一个预定的时间间隔就会清除通过/失效位模式。如果有更多的DUT要测试，260X进入循环并等待下一个SOT触发信号。当完成全部测试后，在260X前面板会显示吞吐率和分级结果；吞吐率和分级结果也会打印至输出队列以供系统控制器读取。

待机电流测试

测试介绍

此测试包括测量待机状态下电池供电产品的电流。典型的测试设置如图3所示。

在这个例子中，260X的HI和LO端子连至待测产品的电池HI和LO端子。260X的电压源模拟产品的内部电池，260X的安培表测量部件在待机模式下的电流。比较测量结果与规定的极限并判断该产品是通过还是失效。合格的电流通常在毫安或微安范围，但根据应用不同会有很大变化。

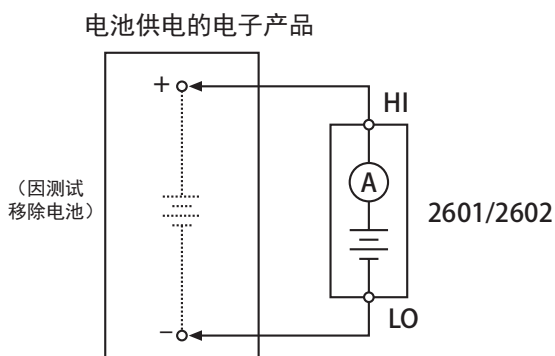


图3. 测试电池供电产品的待机漏电流

测试系统配置

图4是基于260X的待机电流生产测试系统框图。电池供电产品放在连至260X的测试夹具中。触发后，260X输出一个电压并测量产生的电流。如图4所示，260X具有IEEE-488和RS-232通信端口，以及数字I/O端口。当260X通过GPIB连至PC主机时，260X就能通过RS-232与机械手直接通信。而且，数字I/O端口能直接发送信号至机械手，也能从机械手接收信号。将DUT的测量电流与260X中预设的极值比较。260X数字I/O端口产生的TTL信号电平用于指示器件是通过测试还是测试失败。根据接收到的信号，自动机械手将器件连至适合的分级或其它位置。

应用举例2：测试移动电话的耗电电流

在这个应用中，260X代替了移动电话的可充电电池并测量电话在“通话”，“待机”和“关机”模式下的电流。这三项测量都使用4.5V测试电压。对于此测试而言，如果测量电流值超出给定模式下的阈值电平那么移动电话未通过测试。每个阈值电流都有对应的通过和失效位模式：

模式	阈值电流	通过位模式	失效位模式
通话	400mA	0001	0010
待机	10mA	0001	0100
关机	100μA	0001	1000

方案：开发一个TSP脚本，用于创建测试手机耗电电流的函数。除了源电流是实际测量和检查的而不是仅将箝位值用作阈值极限以外，此脚本与I_{DDQ}测试脚本很类似。完整的脚本可以在吉时利网站（www.keithley.com）下载。此脚本可以用测试脚本生成器查看、编辑、装载和运行。

设备列表

要求使用下列设备构建所述二极管生产测试系统和运行吉时利的示例脚本：

1. 吉时利2601或2602源表
2. “数字测试系统”和/或器件机械手与测试夹具
3. IEEE-488（GPIB）接口卡（KUSB-488、KPCI-488或者等效装备）
4. 吉时利7007 IEEE-488接口电缆
5. 定制的DB-25数字I/O机械手接口电缆，用于将仪器连接至数字测试系统和/或机械手
6. 测试线，连接测量仪器与测试夹具

替代方案

260X电流表能测量从3A至100nA的全量程电流，其中每个量程的分辨率为10ppm。如果需要更高的电流分辨率，必须使用带有更高灵敏度安培表的仪器。例如6430型亚飞安远程源表和236、237和238型源测量单元。但是，由于这些仪器的灵敏度较高，其测试速度将比使用260X的速度慢。如果必须测量大于3A的电流，可以使用2440型5A源表。

另一个方案是使用回读电源用于源电压和测量电流。但是，大多数可编程源的回读分辨率为毫安范围，所以不能用于低电流测试。此外，这些电源常带有极低输入阻抗和相对高电流（微安量程）的感测线，这会严重影响测量准确度。

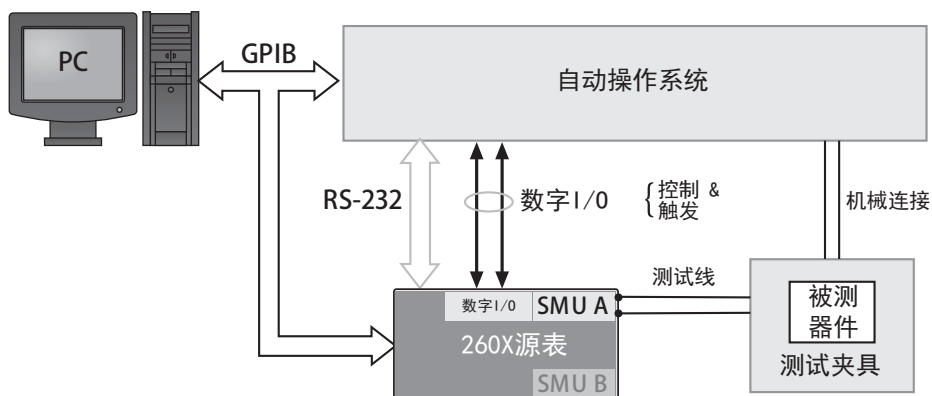


图4. 基于260X的待机电流生产测试系统框图

测试系统安全性

许多电气测试系统或测量仪器都能测量或输出危险的电压和功率电平。而且，在单错误条件下（例如，设置错误或仪器故障），甚至在系统提示当前无危险的情况下，也可能输出危险电平。因此在任何时候都要保护操作人员远离这些高电压和高功率电平等危险情况至关重要。保护方法包括：

- 设计测试装置，防止操作人员接触任何危险电路。
- 确保被测器件完全封闭，保护操作人员免受飞溅碎片的伤害。例如，在施加过高电压或功率时，电容器和半导体器件都有可能爆炸。
- 操作人员可能触碰到的全部电气连接都必须双重绝缘。双重绝缘可以在其中一个绝缘层失效后仍然确保操作人员的安全。
- 当测试装置盖打开时，使用可靠性高的失效-安全互锁开关断开电源。
- 尽可能使用自动机械手，让操作人员无需接触测试夹具内部，也无需打开防护。
- 为所有系统用户提供正确的培训，让他们了解各种潜在危险并知道如何保护自己不受伤害。

测试系统设计工程师、集成工程师和安装工程师的职责是确保对操作人员和维护人员的保护到位而且有效。


延伸阅读

D. Leslie, “QTAG: The Evolution of a Standard Monitor: A Progress Report (QTAG: 标准监视器的演进: 发展报告),” Evaluation Engineering, pp 26-32, Oct. 1995.

M. J. Riezenman, “Technology 1996: Test & Measurement (1996年技术: 测试与测量),” IEEE Spectrum, pp. 65-69, Jan. 1996.

S. S. Sabade and D. M. Walker, “IDDX-based Test Methods: A Survey (基于IDDX的测试方法: 调查),” ACM Transactions on Design Automation of Electronic Systems, Vol. 9, No. 2, pp 159-198, April 2003.

J. M. Soden and C. F. Hawkins, “IDDQ Testing and Defect Classes — A Tutorial (IDDQ测试与缺陷种类——专题指南),” in Proc. of Custom Integrated Circuits Conf, 1995, pp. 633-642.

 **北京海洋兴业科技股份有限公司** (证券代码: 839145)

北京市西三旗东黄平路19号龙旗广场4号楼(E座)906室

电话: 010-62176775 62178811 62176785

企业QQ: 800057747 维修QQ: 508005118

企业官网: www.hyxyyq.com

邮编: 100096

传真: 010-62176619

邮箱: market@oitek.com.cn

购线网: www.gooxian.net



扫描二维码关注我们
查找微信公众号: 海洋仪器