



テスト時間を削減する電源の使用法 – 10のヒント

ご注意

2002年6月13日より、製品のオプション構成が変更されています。
カタログの記載と異なりますので、ご発注の前にご確認をお願いします。



Agilent Technologies



DUTの待ち時間なくす

テスト・プログラムの多くが、ほとんどの時間を持つことに費やしています。しかし通常は、大量のデータ転送をしている場合を除いて、責められるべきはコンピュータ、I/O、測定器ではありません。コンピュータ・ハードウェアのアップグレードや、より高速な測定器を求める前に、実行シーケンスを注意深く見直しましょう。

改善の第1段階

まず、次のテストのためにDUTを必要な状態にするテストを見つけます。例えば、テスト開始のためにDUTをオフにする必要がある場合は、DUTをオフにして終るようなテストを行います。DUTのウォームアップが必要なテストの場合は、それをシーケンスの後のはうに置いて、システム・タイマを使用してDUTを十分な時間オンにします。これらは常に可能ではありませんが、大きな改善になります(また、よく練られたテスト・プランなら、すでにこのレベルの最適化は通常実施済みとなっています)。

改善の第2段階

最適化の次のレベルは、個別のテスト・ベースで行います。次のような典型的なシーケンスを考えます。

- DUTに負荷をかけ、またはそのプログラムされたステートにセットアップし、DUT出力が安定するのを待つ。

- 測定器を動作させるためにリレーを接続し、リレーが閉じるのを待つ。
- 測定器をセットアップして、セットアップの完了を待つ。
- 測定を開始して、測定の完了を待つ。
- リレーを切り離す。
- 電源をオフにする。
- DUT出力が安定するのを待つ。

一般にどのステップも、動作の完了を待っています。さらに、大部分のDUTは、パワーを印加したり、負荷条件が変化した後は、安定するまでの時間が必要です。プログラミングと待ちステージを分離すると、一方の測定を待つ間に他方をプログラムといったテストの改良が可能です。

- DUTに負荷をかける。
- 測定器を動作させるためにリレーを接続する。
- 測定器をセットアップする。
- 上記の動作が完了するのを待つ。
 - リレーが閉じる
 - 測定器の安定
 - DUT出力の安定
- 測定を開始する。
- 測定の完了を待つ。
- リレーを切り離す。
- 電源をオフにする。
- DUT出力が安定するのを待つ。

待ち時間を重ねることで、全体の遅延を最小限に抑えられます。DUTが安定するのを待つ間に、テスト・プログラムはリレーのプログラムと測定器のセットアップを行ないます。

待ち時間を重ねるには、共通のタイマまたはグローバル・タイマを使用します。測定器やDUTをセットアップする各プログラム・ルーチンは、それらの動作にかかる時間をグローバル・タイマに知らせます。これにより、どの動作が最長の時間を要するかが分かります。その後、測定や他のテストが前のコマンドの終了を待つ必要がある場合は、单一のウエイト関数を呼び出し、グローバル・タイマの完了を待って、続行します。

- DUTに負荷をかける。
- 測定器を動作させるためにリレーを接続する。
- 測定器をセットアップする。
- グローバル・タイマの経過を待つ。
- 測定を開始する。
- グローバル・タイマの経過を待つ。
- リレーを切り離す。
- 電源をオフにする。

この方法により、テストは測定器セットアップのために絶対に必要な時間以上待つ必要がなくなり、プログラミングも簡単になります。



複数の電源とGPIB動作の オーバラップによるテスト 時間の短縮

測定スピードが重要な場合は、1台のマルチ出力電源の代りに、複数台の単出力電源の使用を考えます。複数台の電源を使用することで、GPIB動作をオーバラップさせて、マルチ出力電源でのシーケンシャルなコマンド処理による遅延を防ぐことができます。マルチ出力電源では、それぞれの出力に送られるコマンドは、一度に1出力ずつ、シーケンシャルに処理されます。これに対して複数台の電源のセットアップでは、ある電源がコマンドを受けている間に他の電源はコマンドを処理できます。

この方法は、電源への問合せを行う場合に最もメリットがあります。マルチ出力電源の場合は、ある出力に測定コマンドを送って応答を得るプロセスを、次の出力に問合せする前に終らせる必要があります。一度に1つずつ測定しなければならないため、このシーケンスのような問合せでは2測定サイクルが必要です。

```
OUTPUT Dev1;"VOUT1?"  
ENTER Dev1;"Volt1  
OUTPUT Dev1;"VOUT2?"  
ENTER Dev1;"Volt2
```

これに対し複数台の電源の場合は、最初に全電源に対しコマンドを送って測定を開始してから、応答を受けることができます。測定がオーバラップするため、この問合せシーケンスは1測定サイクルしか必要ありません。

```
OUTPUT Dev1;"MEAS:VOLT"  
OUTPUT Dev2;"MEAS:VOLT?"  
ENTER Dev1;"Volt1  
ENTER Dev2;"Volt2
```

VISAソフトウェア・ドライバを使用している場合は、問合せを実行するためにはviQueryf()が便利な関数です。しかし、この関数はオーバラップした動作ができません。オーバラップした問合せを行うには、viPrintf()およびviScanf()を使用して問合せを個々のステップに分解します。例えば、以下のようになります。

```
viPrintf(viDev1, "MEAS:VOLT?\n");  
viPrintf(viDev2, "MEAS:VOLT?\n");  
viScanf(viDev1, "%lf",&Volt1);  
viScanf(viDev2, "%lf",&Volt2);
```

個々のセットアップや問合せについての時間短縮は大したことがないかも知れませんが、複雑なくり返しテストでは、累積された時間短縮によりシステム・スループット全体に対して大きな効果が得られます。

INSIGHT³

電源および負荷の内蔵測定機能の利用

多くの電源や電子負荷に内蔵されている測定機能を利用することで、自動テストの時間を短縮すると同時に、複雑さを緩和できます。電源では、これらの機能により出力電圧および電流を測定できます。電子負荷では、負荷入力電圧および電流を測定できます。

その良い例が、4出力を備えたDC-DCコンバータのテストです。このデバイスを完全にテストするには、コンバータへの入力電圧と、4つの出力を測定する必要があります。電圧測定用に1台のDMMがある場合は、各測

定をシーケンス化するにはマルチプレクサが必要です(図1)。このセットアップでは、複雑さに加えて、テスト・プログラムは、マルチプレクサのスイッチングと各測定のセッティングを待つ必要があります。

これに対して、コンバータをテストするために、DC電源と負荷を用いることができます(図2)。これらはすでにDUTに接続されており、スイッチング遅延もないために、セットアップとテストの両方のステージがはるかに高速化されます。ここで、リモート・センシングの利用も考えられま

す。これは必須ではありませんが、負荷やDC電源ではなくDUTでのレギュレーションと測定が提供されるため、一般に良い方法といえます。

DMMとマルチプレクサが不要なため時間とラック・スペースの削減はもとより、スイッチングが不要なため、高速なテスト、高い信頼性、シンプルな構成というメリットが得られます。電流測定の場合も同じ方法を利用でき、電流シャントの必要がありません。

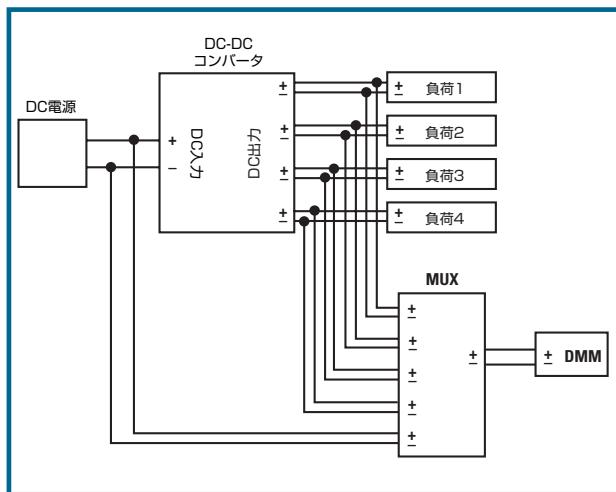


図1. 1台のDMMにより4出力DC-DCコンバータをテストする場合、複雑なマルチプレクシングが必要で、大きな遅延も発生します。

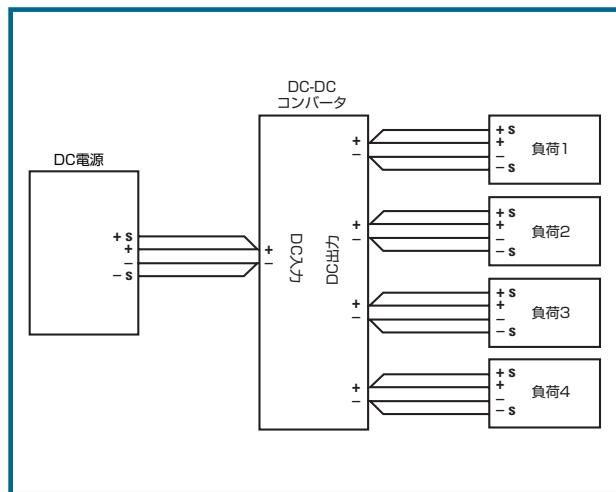


図2. DC電源および電子負荷の内蔵測定機能を使えば、DMMおよびMUXが不要となり、テスト・スピードも大幅に向上します。

INSIGHT⁴

ワンボックス電源を使用したATEシステムの統合、プログラミング、保守時間の削減

今日、システム電源、AC電源、電子負荷の多くのが、電圧／電流プログラミング機能、ステータス・リードバック機能、サービス要求割り込み機能を単一のパッケージ内に内蔵しています(図1)。ATEシステムのために、個々のコンポーネントから電源サブシステムを組み立てて、その性能を心配する必要がありません。ワンボックス電源の仕様を見れば、どのような性能が期待できるかが正確にわかります。

利点

ワンボックス・ソリューションは、複雑さを緩和し信頼性を高めるさまざまな機能を提供します。

- **完全に仕様化された性能。** GPIB入力から電源出力まで、測定器全体が仕様化されています。
- **システム・コストの削減。** すべてがワンボックス内にあるため、購入すべき項目のリストや、それら

をラックに詰め込む必要もありません。外部のケーブリングやユニット間の相互接続も不要のため、信頼性が大幅に高まり、統合が容易になります。

- **使いやすさ。** フロントパネルのコントロールにより、システム開発がスピードアップします。出力は、ボルト、アンペア、文書化プログラミング・コマンドでプログラム可能です。また、ステータス・リードバックや電子校正などの拡張機能により、開発が加速され、保守作業も軽減できます。
- **アプリケーションの保護。** 危険な状況が起れば、過電流／過電圧保護が出力をシャットダウンし、割り込み要求を送出します。また、新しい測定器の多くが装備する外部制御ポートにより、外部イベントに対応した緊急システム・シャットダウンの組み込みも簡単です。

高度な機能

多くの最新の電源製品は、測定機能も内蔵しています。

- **補助DVM機能。** 最近の電源製品は測定機能を内蔵するものがあるので、追加測定器が少なくなったり、無くて済む場合があります。
- **パワー・アナライザ測定機能。** 内蔵のパワー・アナライザにより、AC電源で生じたライン障害が被試験装置に及ぼす影響をモニタできます。さらにAC電源には、パワー保護装置の出力に接続できる補助パワー・アナライザ入力を持つものもあります。

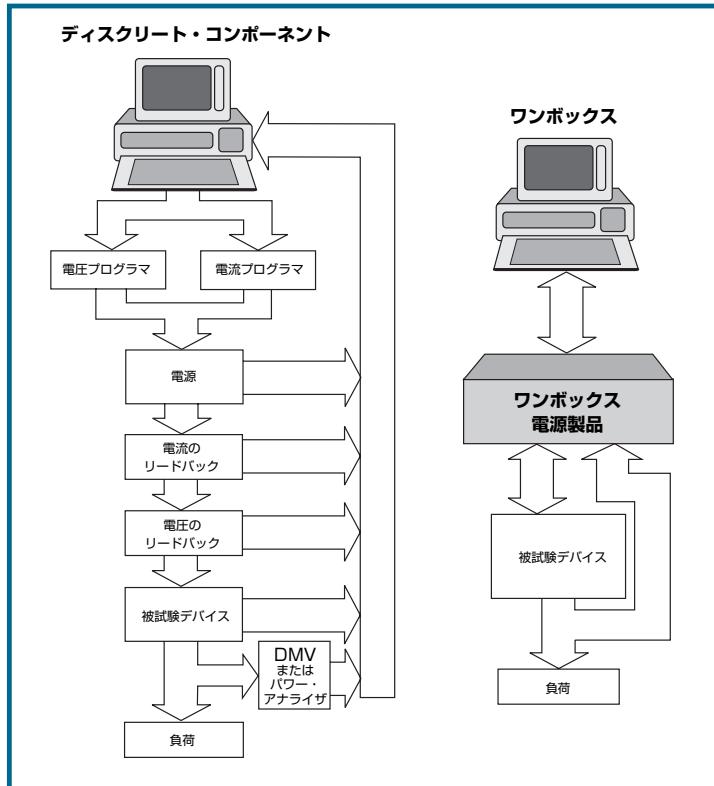


図1. 電源製品のワンボックス・ソリューションは、自動テストに必要なコンポーネントの多くを1台の装置に内蔵しています。



INSIGHT^s

ディジタル信号処理による測定器の性能と価値の増大

低コストの浮動小数点ディジタル信号処理(DSP)が、多くのACおよびDC電源の性能、機能、価値を増大させました。このような機能の増大が、テスト・スループットや開発時間の大幅な改善にもつながっています。

アナログの限界を克服

DSPは、測定器の全体的な性能が最終的に個々のコンポーネントによって制限されるという、従来のアナログ設計の限界を克服するという点で、大きな前進をもたらしました。最良の設計でも、半導体プロセスの限界、ノイズ、温度変化、必要とする信号の劣化、その他の要素の影響を受けることがあります。このような不要な影響の補正は、非常に困難で、場合によっては不可能でした。

今日の測定器の設計では、処理チェーンのできるだけ早い段階で信号をデジタル化することで、これらの障害を乗り越えようとしています。デジタル化した後は、誤差源の制御がより容易なディジタル・ドメインですべての処理を行います。

電子計測の新たな可能性を創造

ディジタル・ドメインでの基本パラメータの測定確度が向上するに伴い、データの処理に関して事実上無限の

可能性が開かれます。例えば、DSPベースの電源は、一般に出力電圧／電流のデジタル化を行うだけでなく、デジタル・オシロスコープと同様の方法でタイム・ドメイン波形を捕捉します。瞬時パワーからFFT、カスタム・フィルタ測定にいたるまでを、アナログ設計で必要な物理回路ではなく、アルゴリズムによって実現できます。

インテリジェント測定器が処理作業をホスト・コンピュータから譲り受けることにより、自動テストのスピードアップも可能となります。

より良い情報でより良い設計

テスト・デバイスの電力消費に関する情報は、ヒューズの選択、負荷の予測、その他の重要な設計判断に役立ちます。スペクトラム情報により、AC電源ラインの高調波放射の振舞いを理解できるだけでなく、そのような放射を規制する規格への適合テストも可能になります。携帯電話などの繰り返しレートの低いパルス負荷デバイスの電力消費プロファイルは、電池寿命についての貴重な判断情報となります。生データを超えた処理済みの情報を提供できるので、DSPは設計および製造のテスト・サイクルを短縮することができます。



INSIGHT⁶

より正確な電源の使用

電源の確度(出力レベル・プログラミングとリードバックの両方の確度)が向上することのメリットは、正確さ、分解能を考えれば明らかです。しかし、電源の確度向上は、スピードも大きく向上させます。

電源が正確さや安定性に欠けると、常にDMMを使用して出力レベルを検証したり、プログラム・ループを使って電圧を必要な値またはその近似値に保つ必要があります。温度変化、突然の負荷変動、不十分な分解能、これらは問題の原因のごく一部にしかすぎません。より正確な電源を使用

すると、複雑さ、出費、遅延を避けることができます。

あるシステムを構築するのに、正確でない電源や多くのシャント、マルチプレクサ、DMMなどのコンポーネントを使わなければならぬとき、状況はより困難なものとなります。この点で、ヒント4で紹介したワンボックス電源システムが役立ちます。システム内のディスクリート・コンポーネントが少なくなるため、信頼性が高まり、エラーや不安定性が減少します。



大量のデータ転送に バイナリ・モードを使用

測定器との間で、測定波形などの大量のデータを転送する場合、その測定器がバイナリ転送をサポートしているかを確認してください。測定器が波形キャプチャ機能を持っている場合は、ASCIIモードに加えてバイナリ・モードを提供していることがあります。データのみを転送する場合には、ASCIIの英数字機能は必要ありません。バイナリ転送はバイト数が少なくて済むため、時間も半分かそれ以下で済みます。

バイナリ・モードの欠点は、正しい転送を行うために、追加のデータ処理作業(バイト順など)が発生することです。(また、ASCIIモードを意図している場合は、うっかりバイナリ・モードに設定しないように注意する必要があります。)

バイナリ転送では、VISA I/Oライブラリがユーザに代わって処理の多くを行います。次の例は、ACおよびDC電源のバイナリ・モードの使用法を示しています。このプログラムはCで書かれていますが、コンセプトはどのような言語でも同じです。(VISAライブラリをVisual Basic®、Agilent VEE、LabVIEW®、その他のプログラム環境から呼び出します。)

```
/*
* このプログラムはAgilent AC電源6811B、6812B、6813B、6841A、6842A、6843A、または
* Agilent DC電源66309B、66309D、66311A、66311B、66311D、66312A、66332Aの
* いずれかを使用した、バイナリ・モード・データ転送の例を示します。
*
* 測定器との通信に、VISA GPIBライブラリを使用します。実行ファイルの作成時に、
* 本プログラムをvisa32.libにリンクします。
*
* 結果は標準出力に書き出され、通常はファイルヘリダイレクトする必要があります。
***** */

#include <stdio.h>
#include "visa.h"

#define MAX_POINTS 4096      /* 最大の配列は4096ポイント */
#define PWR_SRC_ADDR "GPIB0::5::INSTR" /* GPIBとのインターフェース、アドレス5 */
#define TIMEOUT 10000        /* 10秒 */

void main()
{
    ViSession defaultRM, PwrSrc;
    float aVolts[MAX_POINTS];
    long lPoints, i;

    /* デフォルトのリソース・マネージャ (各VISAプログラムで必要) と
     * 電源 (AC電源またはDC電源) に対して、
     * セッションをオープン
     */
    viOpenDefaultRM(&defaultRM);
    viOpen(defaultRM, PWR_SRC_ADDR, VI_NULL, VI_NULL, &PwrSrc);

    /* 大きな配列の転送に十分な時間のタイムアウトを設定
     */
    viSetAttribute(PwrSrc, VI_ATTR_TMO_VALUE, TIMEOUT);

    /* 電源のデータ収集サブシステムをアームしてそれをトリガ
     */
    viPrintf(PwrSrc, "initiate:name acquire\n");
    viPrintf(PwrSrc, "trigger:acquire:immediate\n");

    /* 配列の転送にバイナリ・フォーマットを選択
     */
    viPrintf(PwrSrc, "format real\n");

    /* 電圧データの配列を電源に問合せ
     */
    viPrintf(PwrSrc, "fetch:array:voltage?\n");

    /* データをフェッチ。##zbフォーマットは以下を実行:
     * - 使用するプロセッサ (Intel、Motorolaなど) に対してバイトを適切に調整して、
     *   バイナリ・データを配列aVoltsに入れる。
     * - 読み取りポイントの最大数としてlPointsの初期値を使用。
     * - 読み取りポイントの実際の数をlPointsに入れる。
     * - ##tフォーマットは、次の問合せと干渉しないように、
     *   VISAライブラリの入力バッファにバイトが残らないようにする。
     */
    lPoints = MAX_POINTS;
    viScanf(PwrSrc, "%#zb%*t", &lPoints, aVolts);

    /* データ配列を標準出力に送出
     */
    printf("%ld points\n\n", lPoints);
    for (i=0; i<lPoints; ++i)
        printf("%7.2f\n", aVolts[i]);

    /* VISAセッションをクローズ */
    viClose(PwrSrc);
    viClose(defaultRM);
}
```



高度なステータス・レポート機能の活用

今日のシステム電源／負荷のような機能が与えられれば、測定器内部で何が起きているか、入力信号や他の要素の変化に応じてどのような動作がとられたかについて、従来よりも多くを知る必要があります。

例えば、一般的なモニタリング・タスクとして、電源が定電流(CC)モードにいつ移るかのモニタリングがあります。このモードでは、電源は負荷変動に応じて、指定した電流レベルを維持するために電圧を調整します。これは、例えば論理デバイスがフェイルして、被試験デバイスから急に高い電流が流れた場合などに生じる可能性があります。このような状況への対応を誤れば、負荷に過大な損傷を与えることや、危険な状態が発生したりします。

これに対しては、GPIBを通して、CCビットの変化を確認することにより、電源のステータスを読み続けることが可能です。しかしこれは、コンピュータにとって遅く、時間を浪費する作業です。

より高速な方法は、CCモードになるとシリアル・ポール・レジスタにビ

ットをセットするように、電源を設定する方法です。シリアル・ポールの実行は、より速いGPIB動作であり、各チェックにおいてプロセスが消費する時間が少なくて済みます。また、プログラミング環境がGPIBサービス要求(SRQ)による割り込みをサポートしている場合は、ポーリング・プロセスを省いて、CCモードに入れば電源がSRQを発生させるように設定できます。

ステータス・レジスタのセットアップは測定器により異なりますが、非常に強力なモニタリング／レスポンス機能を提供するように、時間とともに発展してきました。例えば、多くのAgilent Technologiesの電源では、標準イベント(CV、CCなど通常の動作状態)および問題イベントとしての動作モード／状態を示すレジスタを提供しています。問題イベントには過電圧、過電流、過熱状態、および非レギュレーション(電源が定電流モードでも定電圧モードでない)への遷移が含まれています。これらの状態のすべてが、ステータス・レジスタにビットをセットし、またSRQの発生に使用できます。

リスト・モードと バックプレーン・トリガの使用

リスト・モードという機能を持つ電源では、完全な測定器セットアップを保存して、単一のコマンドで(長い個々の構成ステップの連続でなく)、セットアップを呼び出すことができます。テスト・シーケンスが複雑なほど、リスト・モードによる時間の節約も大きくなります。またリスト・モードは、同時にさまざまな電圧レベルが必要な、マイクロプロセッサなどのアプリケーションに対して便利です。

モジュラ型電源システムでは、リスト・モードとともにバックプレーンを介したトリガを使用すると、さらに高度な機能と柔軟性が得られます。例えば、さまざまなモジュールに対して、それぞれが特定の電圧レベルを出力するように予め構成し、1つのトリガ・コマンドで、それらを異なる時間にオンにできます。図1は、そのようなセットアップのブロック図を示しています。図2は、その結果の出力曲線です。

リスト・モードとトリガリングは、テスト・システムのコンピュータによる制御タスクを削減することにより、さらに別の方法でも役に立ちます。プログラム・シーケンスをスタートさせる单一トリガ・コマンドとともに、バックプレーンを介したモジュール間のトリガリングを使用して、次のモジュールのリスト・モードのセットアップを開始できます。

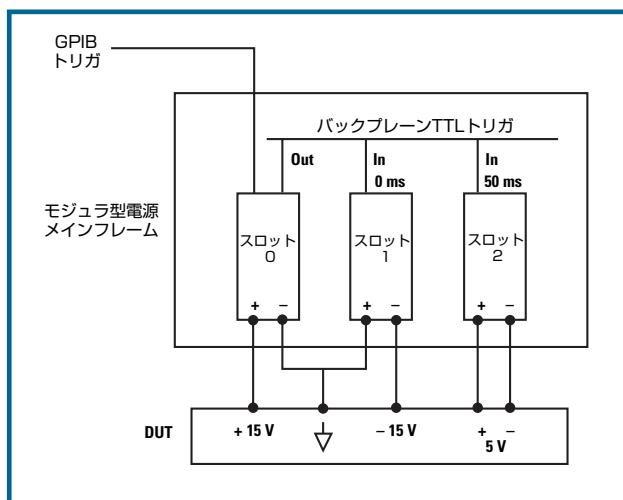


図1. この3モジュールのセットアップは、バックプレーン・トリガを使用して、異なる時間に出力を開始します。スロット0と1は同時にパワーを供給し、スロット2は50ms後に供給を開始します。

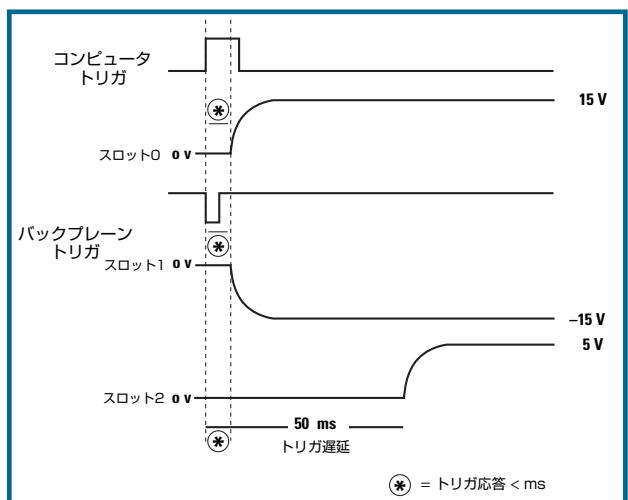


図2. 図1のセットアップによるパワー曲線



ダウンプログラミングを備えた電源による高速レベル変更

ダウンプログラミングとして知られる機能を備えた電源を使用すると、特に複数の電圧レベルの設定が必要な場合に、テスト時間を大幅に短縮できます。ダウンプログラミングを使用しないと、出力電圧レベルを下げる際に、電源の出力フィルタ内のキャパシタ（または負荷容量）の放電に数秒から数分かかることがあります（負荷が軽いほど時間がかかる）。

ダウンプログラミングは能動回路を使用して、ほとんどの場合、数ミリ秒で出力を新たなレベルまで下げます。この回路は、（手動またはプログラム）で設定した電圧レベルが現在の出力レベルよりも低ければ自動的に作動します。ほとんどの電源でダウンプログラミングのレベルは固定ですが、プログラマブル・ダウンプログラミングを備えた電源もあります。

時間が重要なテストでは、ダウンプログラミングの遅延に注意してください。プログラミング・アップは一般にプログラミング・ダウンよりも高速なため、複数のテストをシーケ

ンス化する場合、連続した各テストが同じ電圧レベル、または前よりも後の方が高い電圧レベルとなるようにします。

ダウンプログラミングは、バッテリの放電にも便利です。バッテリを充電した後で出力電圧を下げたとき（または一部の電源で“OUTPUT OFF”コマンドにより出力をオフにしたとき）、ダウンプログラミングがオンとなって電源を電子負荷のように動作させて、バッテリを放電します。これは必要なときには非常に便利な機能ですが、放電させたくないバッテリをうっかり放電させてしまうこともあります。放電しないように、電源の出力レベルを下げる前にバッテリを切斷してください。

注意: バッテリの充電／放電時に、電源のAC電源をオフにしないでください。電源の過電圧保護回路が作動してバッテリをショートし、このため電源が損傷したり、バッテリが過熱することがあります。

アジレント・テクノロジー株式会社
本社 〒192-8510 東京都八王子市高倉町9-1

計測
お客様窓口

受付時間 9:00~19:00
(土・日・祭日を除く)
※FAXは24時間受付け

TEL **0120-421-345**
(0426-56-7832)
FAX **0120-421-678**
(0426-56-7840)
E-mail: contact_japan@agilent.com

電子計測ホームページ
<http://www.agilent.co.jp/find/tm>

●記載事項は変更になる場合があります。
ご発注の際はご確認ください。

Copyright 2002
アジレント・テクノロジー株式会社

Agilent Technologiesの電源製品、
システム、アプリケーションについての
詳細は、当社のエンジニアにお問合せ
いただか、または当社のWebサイト --
www.agilent-tech.com/find/insight10
-- を参照してください。
このWebサイトでは詳細な製品情報、
ドライバやソフトウェアのための
ダウンロード・リンク、
アプリケーション・ノート、技術論文、
FAQへの回答などを提供しています。

電子計測UPDATE

www.agilent.com/find/emailupdates-Japan

無料の電子メール情報

Agilentからの最新情報を記載した電子メールを無料でお送りします。ご購読いただいた方には、選択された分野に関する最新情報を定期的にお届けします。対象となる分野は、サポート、製品とサービス、アプリケーション、プロモーション、イベント、その他です。購読の中止や選択分野の変更も簡単にできます。

購読申込みはこちらから：<http://www.agilent.com/find/emailupdates-Japan>

Agilentは皆様のプライバシーを尊重し、保護することをお約束します。皆様に対する当社のお約束の内容は、<http://www.agilent.com/go/privacy>にある当社のプライバシー・ステートメントに記載されています。Agilentのプライバシー方針に関するご質問はprivacy_advocate@agilent.comまでお寄せください。



Agilent Technologies

June 6, 2002
5968-6359J
0000-00DEP