

2602型システムソースメータを使用した スケーラブル、マルチピン、マルチ ファンクションIC試験システムの作成

はじめに

携帯電話、PDA、デジタルカメラ、ポータブルオーディオシステムなどの家庭用電化製品は、年々小型化、高速化、低価格化が進み、新製品の市場投入までの時間が以前と比較して短くなってきています。これに追随するため、半導体、パッシブ/アクティブ部品業界では、統合性の高い、より複雑な開発プロセスが必要です。このため、より小さいスペースにより多くの回路を実装することになり、アナログ、デジタル、さらにはRF回路がシステムオンチップ(SoC)にすべて実装されます。同様に、ディスクリート部品製造メーカーも、集積度をあげるため複数の部品を1つのチップに実装しています。

一般的に、被測定デバイス(DUT)が複雑になるにつれて、より大量で、複雑な試験が要求されます。全体の製造コスト削減の一環として試験コスト(COT)削減のプレッシャーに常にさらされている電気電子部品製造メーカーにとって、これは大きな問題となり得ます。試験スループットはCOT削減に非常に重要ですが、これは話の一部分にすぎません。一般的に、COTの削減は、より多くのデバイスを短時間で、より小さいフロアスペースで実行することを意味します。また、より効率的に試験装置を利用、再利用することも意味します。

これらの要求事項は、テストシステム設計のCOTを削減とどのような関係があるのでしょうか？

- ピン数が多く、複雑な試験では、試験システムに割り当てられたスペースにより多くの印加測定チャンネルを配置する必要があり、システムの密度を上げることが非常に重要である。
- 新しい部品開発ペースにより効率的な利用/再利用が非現実的であり、不可能な場合には、新しい部品ごとに特化したシステムを設計している。マルチピンデバイスの試験を行い、最小限のシステム再設計で新しいデバイスを処理するためには、柔軟性と拡張性が高く、再構成可能なシステムが必要である。
- カスタム装置ではなく、可能なかぎり市販の試験装置を使用することによって、再利用性が最大限に高まる。
- 平行試験機能を備えた多チャンネル試験システムは、複雑なICやSOCの異なるセクションを同時に試験できるため、スループットが劇的に高い。
- 一連の試験シーケンスを測定器のメモリにダウンロードし、システムコントローラを他の機能に解放できる測定器によりスループットを向上することも可。最近まで、多チャンネル印加測定アプリケーションに最適なソリューションはほとんどなく、試験技術者は選択の余地がなかった。

最近まで、多チャンネル印加測定アプリケーションに最適なソリューションはほとんどなく、試験技術者は次からの選択を余儀なくされてきました。

- 比較的単純なシステムは、PC制御による測定器ベースのシステムで、低速である。
- 高速なシステムは、カスタムトリガ、デジタルI/O、通信コントローラを使用した、複雑な測定器ベースのシステムである。
- 完全に統合された高速なメインフレームベースのシステムは、コストパフォーマンスとスペース効率が悪い。

ケースレーの2600シリーズシステムソースメータは、新たな選択肢で、ラック&スタック測定器の拡張性と柔軟性と、メインフレームベースの統合性と高スループットを兼ね備えています。2600シリーズは、従来モデルの2400シリーズソースメータの機能にさらに磨きをかけています。2400シリーズソースメータは高速、ノンストップでの生産環境での信頼性の高い動作を目的として設計されており、今もなお、多くの試験アプリケーションでの理想的なソリューションですが、多チャンネル印加測定システムの構築に関しては限界があります。本アプリケーションノートでは、2600シリーズによって複雑なマルチピン、マルチファンクションIC試験における数多くの要求事項に対応する方法についてデジタル-アナログコンバータ(DAC)試験の実例をあげて解説します。

内蔵Test Script ProcessorとTSP-Link

図1は、2602型システムソースメータを使用して多チャンネル生産試験システムを実装する方法を示しています。

ケースレーの新しい高速SMU(ソースメジャーユニット)設計、内蔵Test Script Processor (TSP™)、TSP-Linkバスを採用した2600シリーズは、最も多いスループット要求を満たすために必要なスピードを実現します。Test Script Processorは、スケーラブルな内蔵コンピュータです。TSP-Linkは、ユニット間通信とトリガ同期のバスで、基本的には、外部バックプレーンです。TSP-Linkにより、システムインテグレートは複数台の2600シリーズソースメータを接続し、同じシャーシに格納されているかのようにプログラミングできます。プログラミング・インタフェースが簡単であるため、強力な高速な多チャンネル試験のプログラムをすぐに作成できます。制約となるメインフレームがないため、最大64ノードの統合試験システムをシームレスに作成できます。この例では、2602型システムソースメータが各ノードで、システムは128SMUチャンネルまで拡張できます。

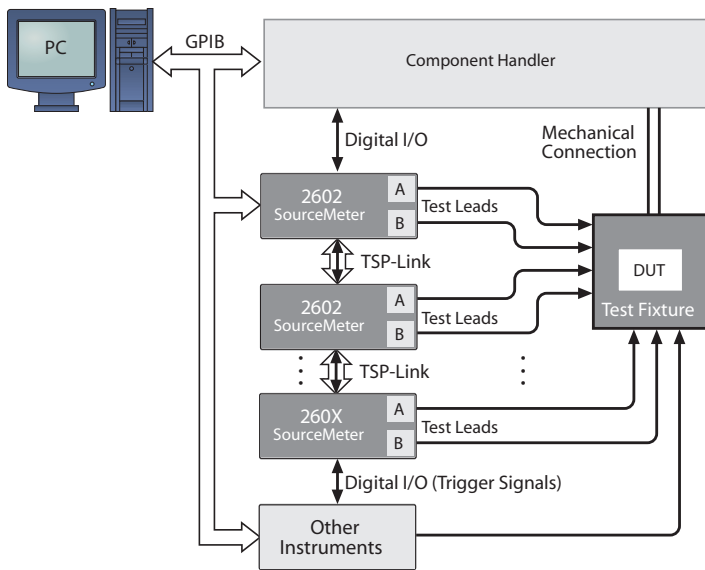


図1. 2600型システムソースメータを使用した多チャンネル試験システムの実装

2U、ハーフラックの筐体に2チャンネルが集約されているため、2602型は市販されているSMUベースシステムのなかで最高レベルのチャンネル密度を実現し、増えつつけるピン数に対応し、ラックスペースを節約します。これらの機能は、最適なシステムの開発をサポートし、チャンネルコストを劇的に削減し、試験システム全体のコストパフォーマンスを高めます。

試験装置とシステムコントローラとの間の通信は、スループットの大きなボトルネックとなりえます。プローバやハンドラが機械動作を行っている間に、コマンドやデータの転送を行うことによって、貴重な試験時間を節約するのが一般的な方法です。しかし、試験システムが複雑さを増すにつれて、コントローラと測定器の間でより多くのやりとりが必要となり、従来の方法では実現が難しい場合があります。

システム内の1台の測定器に、試験プログラム全体をダウンロードし、その測定器が自身の動作と他の測定器の動作を、高速インタフェースを介して制御することを想像してみてください。これがまさに、Test Script ProcessorとTSP-Linkが2600シリーズで実現できることです(図1を参照)。TSPスクリプト(プログラム)は、「マスター」ユニットの揮発性メモリまたは不揮発性メモリのどちらにもダウンロードできます。「マスター」ユニットは、自身の動作と、接続されているその他のすべての「スレーブ」ユニットの動作を制御し、1つのシステムとして動作します。この機能により、ラック内の他の測定器とのインタフェースとなるシステムコントローラが不要になり、システム全体のスループットが向上します。さらに、2600シリーズは十分な容量のメモリを備え、ホストコントローラが2600ベースシステムとやりとりする頻度を削減しています。ソースメータの使用状況によって、メモリの割り当ては変わりますが、プログラムメモリには50,000行以上のコードを格納できることが保証されています。各SMUは2つの不揮発性データバッファを備え、選択したデータオプションによって50,000から100,000個の読み取り値を格納できます。また、アプリケーションに応じて、揮発性メモリをデータストレージとして使用することも可能です。

図1は、マスターユニットとスレーブユニットがTSP-Linkで接続されています。高速な、シリアル通信バスとハードウェアトリガラインが測定器間のリアルタイム制御とデータ交換インタフェースを接続しています。TSP-Linkはピアツーピアインタフェースであり、測定器とホストコンピュータとの間ではなく、測定器間だけの通信を行います。USBやFireWireなどの一般的なホストインタフェースの中にはTSP-Linkバスよりも高いローバンド幅を備えたものがありますが、これらはホストコンピュータとやりとりするために高度なデータ転送プロトコルとその他の互換機能を必要とします。このため、測定器制御では一般的な短いメッセージでは、高いオーバーヘッドが発生しがちです。その結果、少量の情報を送信してやりとりする場合、これらのホストインタフェースはGPIB同様のスピードしか実現できません。TSP-Linkは、測定器間の通信に限定されているため、より単純なメッセージ交換プロトコルで動作可能であり、オーバーヘッドを低減し、従来のGPIBと比較して10倍近い速度を実現しています。

一般的なIC部品の試験

図2は、一般的なIC部品を示しています。部品によっては、アナログ、および入/出力の任意の組み合わせがあり、また1個以上のパワー入力を含む場合があります。図2は、入力と出力で必要となる適用する信号と測定を示しています。測定の中には、カウンタ/タイマ、ゲイン/位相メータ、インピーダンスアナライザ、LCZメータ、スペクトラムアナライザ、またはRFパワーメータなどの測定器が必要なものがありますが、多くの印加/測定要求は、ソースメータで簡単に、満足のいく結果が得られます。

ソースメータは、ユニークなソリューションですが、多くの試験エンジニアにはまだ知られていません。ソースメータの利点は、DCのI-V測定の実行機能で、試料(DUT)に電圧を印加して流れる電流を測定、または試料に電流を印加して電圧降下を測定します。どちらの場合も、ローカル(2線式)およびリモート(4線式)電圧センサが利用できます。ソースメータは高い柔軟性を備えているため、スタンドアロン型の精密定電圧源や定電流源、電圧計や電流計と同じぐらい簡単に使用できます。印加はプログラマブルな4象限動作が可能で、バイポーラの電圧と電流を印加でき、またパワーの印加とシンクの両方が行えます。多くの従来型の電源とは異なり、ソースメータは印加電流や印加電圧の実際の値を測定できるため、プログラミングされた値のみに依存する場合と比べて高確度です。コンプライアンス回路が、電圧源によって試料に流れる電流を制限する、または電流源による試料への電圧降下を制限することによって、被測定デバイスを保護します。精密な印加機能とDMMの測定機能が1台に集約されているため、ケーススレーではこの測定器を「ソースメータ」と命名しましたが、実際の電圧および電流の測定機能は同等分解能のDMMと比較して多くの点で優れています。大半のDMMでは高電圧レンジの入力インピーダンスが10MΩに低下しますが、ソースメータの電圧測定機能の入力インピーダンスは、多くのすべての電圧測定レンジにおいて10GΩ以上です。これは、フルダイナミックレンジにおいて電圧計によるロードが最小限に抑えられることを意味します。相応して、電流測定は、DMMで一般的に採用されているシャント電流計ではなく、フィードバック電流計を使用して行われます。フィードバック電流計は、シャント電流計よりも電圧負荷が非常に低く、理想的なゼロインピーダンス電流計であることを意味します。さらに、ソースメータは、DMMと比較してより広い電流測定ダイナミックレンジを利用できます。

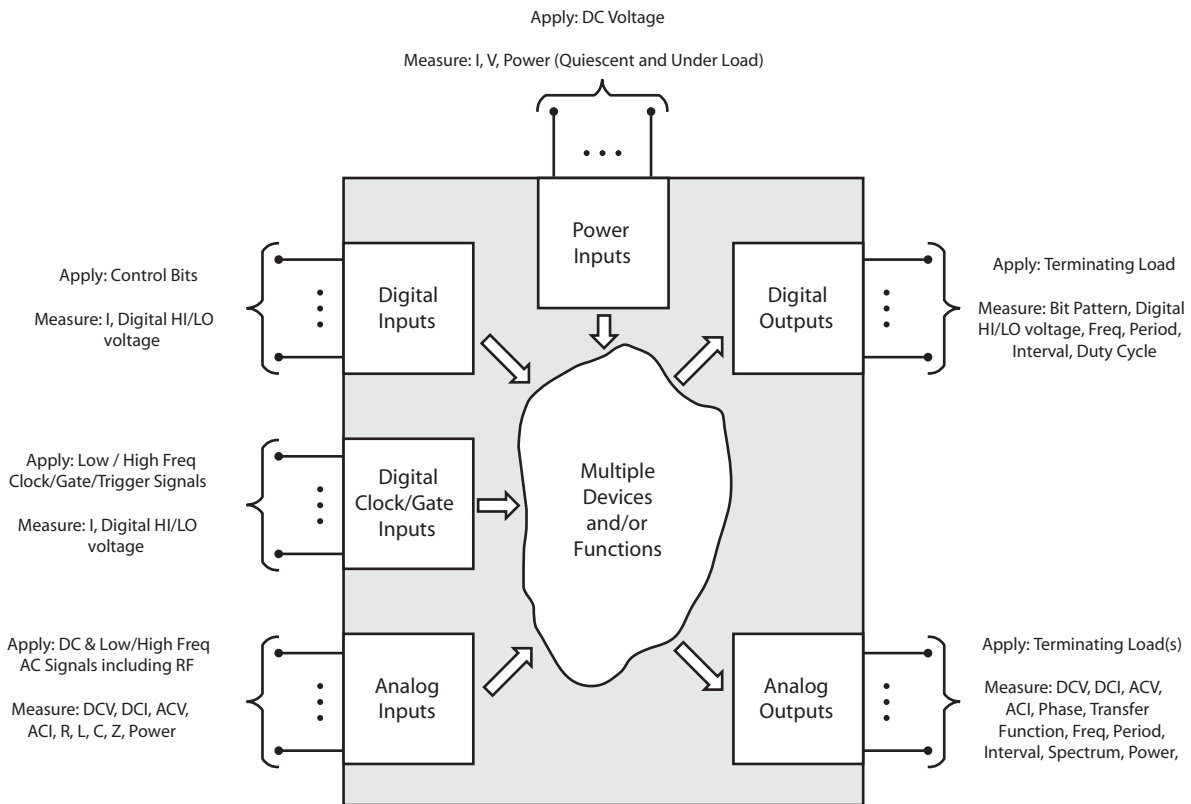


図2. 複雑なICは複数のアナログ/デジタルの入出力が必要です。試験では、複数のスティミュラス信号を使用した複数の測定が必要です。

また、ソースメータは、重要なスイープ、パルス機能を備えています。このため、AC測定と認識されている多くの測定を、パルスまたはスイープDCによって実行でき、幅広い範囲の試験ニーズにこたえるユニバーサルアナログI/Oピンを備えています。弊社が用意した一般的なICの例では、これはつまり、今日のミックスドシグナルデバイスの試験を完全に行うためには追加の装置が必要となりますが、1台または複数台のソースメータを使用すれば他に必要なものはほとんどありません。

デジタル-アナログコンバータ試験の例

多チャンネル、マルチファンクション試験アプリケーションにおける2602型のパワーと柔軟性を説明するため、8ビット乗算デジタル-アナログコンバータ(DAC)を例にします。図3は、このアプリケーションの試験セットアップを示しています。TSP-Linkによって接続されているそれぞれの測定器には、 GPIBアドレスと同様に、ユニークなノード番号が割り当てられます。DACの例では、2つのノード(1および2)が使用されており、各ノードが2つのSMU(AおよびB)を備えています。ノード1は「マスター」ノード、ノード2は「スレーブ」ノードです。コントローラは、 GPIB(またはRS-232)経由で、コマンドを送信するか、またはマスターユニットに完全なテストスクリプトを送信し、マスターユニットがTSP-Linkを介してコマンドをスレーブに送信します。DACにはDCパワーと基準電圧が必要です。ノード2 SMU Aは、+15VDCをDACに電力供給します。すべてのデジタル入力がLowに設定され、再びデジタル入力がHighに設定されたときに、実際に供給された電圧と、DACに流れる電流が測定されます。ノード2 SMU Bは V_{REF} として+10VDCを供給します。すべてのデジタル入力がHighに設定されたときに、この端子の入力抵抗が測定されます。

DACには2つの補完電流出力があります。通常は終端抵抗をこの出力に接続し、測定可能な電圧値に電流を変換しますが、ソースメータは、電流を直接測定できます。これらの出力で測定されるレベルは、デジタル入力の値に依存する係数でスケールした V_{REF} に比例します。

デジタル入力がすべてHighの場合、 I_{OUT1} は最大値、 I_{OUT2} は最小値です。逆に、デジタル入力がすべてLowの場合、 I_{OUT1} は最小値、 I_{OUT2} は最大値です。出力は、デジタル入力の関数として測定され、そのリニアリティが評価されます。

DACのリニアリティ測定は、通常、定常状態で行われます。入力コードはコンバータのデジタル入力に適用され、必要に応じて、コンバータは、コンバータ内を通過するデータによってストローブされます。この例では、DACはストローブを必要としていません。ノード1 デジタル I/O (ビット1から8)は、DACにデジタル入力を供給します。ノード1 DIOの残りのビットと、ノード2DIOの14ビットすべてが、トリガリングまたはデジタル制御に使用できます。図3に示されているとおり、ノード1デジタルポートは、部品ハンドラと通信して、ビニング動作を制御します。入力コードが適用されると、DACの出力は定常化が可能になり、複数回測定を行ってノイズを除去した後、平均出力電流または平均出力電圧が記録されます。適切なセトリグ時間の後、ノード1 SMU Aが I_{OUT1} 、ノード1 SMU Bが I_{OUT2} を測定します。ソースメータ内蔵のデジタルフィルタが、各出力読み値のについて、5つの測定を自動的に平均します。

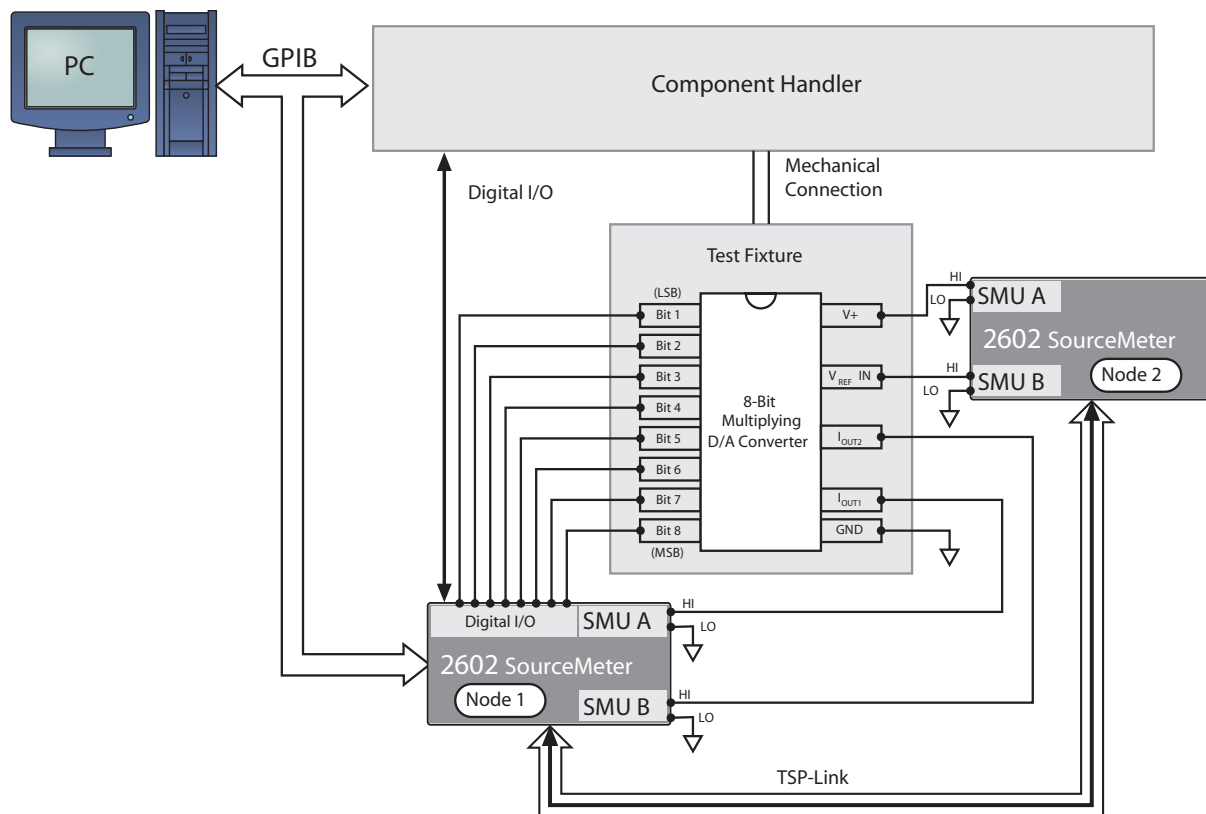


図 3. 8ビット乗算デジタルアナログコンバータ試験用の、2台の2602型のセットアップ

評価する線形パラメータがいくつかあります。

最小有効ビット(LSB)サイズは、入力を1ずつ増分した場合、それに対応する出力変化量です。

$$LSB = \frac{FSR}{2^n - 1}$$

FSRは、コンバータのフルスケールレンジ、nはビット数(分解能)です。

オフセットエラーは、デジタル入力が0の場合の、公称オフセットポイントと実際のオフセットポイントとの差です。公称オフセットが0であれば、デジタル入力が0の場合、オフセットは単純に測定された電流または電力の出力になります。このエラーはすべてのコードに対して同量の影響を及ぼします。このエラーは、LSB単位、またはフルスケールレンジのパーセントで表します。INL測定の一部として算出されるインタセプトの値と等価です。

ゲインエラーは、デジタル入力がフルスケールで、オフセットエラーが0に修正されたときの、伝達関数上の公称ゲインポイントと実際のゲインポイントとの差です。このエラーは、実際の伝達関数と理想の伝達関数のスロープ差を表しており、各ステップにおける同一パーセンテージエラーに相応します。このエラーは、LSB単位、またはフルスケールレンジのパーセントで表します。

積分非直線性 (INL)は、実際のDAC伝達関数の、直線からの偏差を示します。DACの場合、この偏差は各ステップで測定されます。直線は、実際の伝達関数の偏差を最小化するように決定されるベストフィット直線、または伝達関数の終端を結ぶ線のいずれかとなります。INLは、LSB単位またはフルスケールレンジのパーセントで表します。

LSB単位のINLは以下の等式で求められます。

$$INL = \frac{I_{CODE} - [(SLOPE * CODE) + INTERCEPT]}{LSB}$$

CODEは適用されるデジタル入力、 I_{CODE} はCODEで測定される出力電流(または電圧)、SLOPEは基準直線のスロープ、INTERCEPTは基準直線の”Y切片”です。INLの結果は、LSB単位で表されます。ベストフィット基準直線を使用した場合、スロープとインタセプトは線形回帰によって計算されます。終端ベースの基準直線を使用した場合、スロープはLSBと同様に($FSR = I_{MAXCODE} - I_{MINCODE}$)算出され、LSBと同じ数値を持ちます。この場合のインタセプトはデジタル入力が0の場合に測定される、コンバータのオフセット電流(または電圧)です(すなわちCODE =0)。

微分非直線性 (DNL)は、実際のステップ高と、1LSBの理想値の差です。ステップ高がちょうど1LSBの場合、DNLエラーは0になります。DNLのターゲット値(≤1LSB)であれば、DACは単調です。これは、データが全く失われていないことを意味します。出力はデジタル入力に応じて絶えず変化しているため、デジタル増分に応じて増加し、デジタル減分に応じて減少するためです。DNLは、LSB単位またはフルスケールレンジのパーセントで表します。

LSB単位のDNLは以下の等式で求められます。

$$DNL = \frac{(I_{CODE} - I_{CODE-1})}{LSB} - 1$$

I_{CODE} はCODEで測定される出力電流(または電圧)、 I_{CODE-1} は前のコードで測定された電流(または電圧)です。

DAC 試験の例では、オフセット、ゲインエラー、INLとDNLの両方が、測定完了後の両出力に対して決まります。 I_{OUT1} は、INL評価がベストフィット基準直線を使用して行われ、 I_{OUT2} は終端基準直線を使用して行われます。¹

テストスクリプト

従来、試験エンジニアは、PCまたはコントローラに常駐させる試験プログラムを作成していました。このようなプログラムは、テスト・エグゼクティブ、一連の関数プロシージャとその他のルーチンで構成されています。

エグゼクティブは、各種の関数やサブルーチンを正しい順番で呼び出し、試験フローを制御します。関数とサブルーチンは、テストシステム内の測定器にコマンドを送信して、測定器を設定したり、測定を開始したりします。

```
node[1].smua.source.func = node[1].smua.OUTPUT_DCVOLTS
node[1].smua.source.levelv = 0
```

スクリプト言語ではエイリアスを使用できるため、コードの判読性、実行スピードを向上できます。DAC試験の例では、次のエイリアスが定義されています。

```
MASTER = node[1]           --Alias indicating control is via Node 1
SLAVE = node[2]            --Node 2 is controlled by MASTER via TSP-Link
IOUT1 = MASTER.smua        --Alias for SMU measuring current output #1
                             --IOUT1 is equivalent to node[1].smua
IOUT2 = MASTER.smub        --Alias for SMU measuring current output #2
                             --IOUT2 is equivalent to node[1].smub
DIO = MASTER.digio         --Alias for Digital I/O of 2602 #1
                             --DIO is equivalent to node[1].digio
VPLUS = SLAVE.smua         --Alias for SMU supplying V+ and measuring current draw
                             --VPLUS is equivalent to node[2].smua
VREF = SLAVE.smub          --Alias for SMU supplying reference voltage (Vref)
                             --VREF is equivalent to node[2].smub
```

エイリアスは、サンプルスクリプト全体で使用されています。定義されたエイリアスを使用するには、コマンドを次のように書き換えます。

```
IOUT1.source.func = IOUT1.OUTPUT_DCVOLTS
IOUT1.source.levelv = 0
```

そして、測定器から、データやステータス情報を取得し、データを処理、評価して、被測定デバイスの可否を判定し、データを保管します。一般的に、それぞれのDUTが試験されるたびに、コントローラはコマンドシーケンスを測定器に繰り返し送信し、評価対象のデータを継続して取得します。コントローラと測定器との間のこれらの通信はすべて、試験速度の著しい低下の原因となります。Test Script Processorは、制御プログラムの大部分を、ソースメータの揮発性または不揮発性のメモリにダウンロードできます。TSPにダウンロードされたプログラムは、スクリプトと呼ばれます。スクリプトは、複数試験を実行する長い1つのプログラムとして記述することもできますが、PCの制御プログラムのように関数を作成して呼び出すスクリプトを記述できます。いったん関数を作成すれば、Test Script Processorに常駐しているスクリプトや他の関数、またはコントローラ内のテストエグゼクティブから呼び出せます。関数にはパラメータを渡せるので、ホストコントローラに、入力信号レベルや検査リミットなどのDUT特有の試験パラメータを渡すことができます。

DAC試験のスクリプトの詳細資料が、ケースレーのWebサイト www.keithley.com からダウンロードできます。このスクリプトは完全版で、図3のように構成した2台の2602型ソースメータで使用します。下のコードは、2600シリーズの新しいスクリプト言語の概要をつかんでいただくためにDAC試験スクリプトから抜粋しました。連結ダッシュ(-)は、コメントを示しています。

では、一般的なコマンドをいくつかみていきましょう。

¹ このアプリケーション内での定義は、以下のドキュメントから抜粋しました。

- “Optimizing Setup Conditions for High Accuracy Measurements of the HI5741,” Intersil Corporation, Milpitas, CA, Application Note 9619, May 1996.
- “Digital-Analog Converters Are a ‘Bit’ Analog,” Maxim Integrated Products, Inc., Sunnyvale, CA, Application Note 1055, April 16, 2002.
- “Understanding Data Converters,” Texas Instruments, Inc., Dallas, TX, Application Report SLAA013, 1995.

一般的に、スクリプト言語では、変数を明確に宣言する必要はありません。変数に割り当てられた値に応じて、暗黙的に宣言され、型が決まります。ただし、テーブル(すなわち、配列)は例外で、明確に指定する必要があります。すべての変数は、ローカル変数として明示的に宣言しない限り、グローバルです。次のコードでは、定数が使用されています。

```
Vref = 10          --Use +10VDC reference voltage
IoutMax = 0.002   --Max expected current output
Nplc = 0.001     --Integration time for SMU A-to-D converters (in terms of power line cycles)
Nbits = 8        --Number of DAC control bits (digital inputs)
Ncodes = 2^Nbits --Number of possible control codes
MaxCode = Ncodes - 1 --Decimal equivalent of full-scale code (255 for 8-bit DAC)
Lsb = Vref / MaxCode --Nominal value of least significant bit
```

一般的には、実際の試験シーケンスを開始する前に、測定器の初期設定を行います。この例では、初期設定で印加機能とレンジ、測定機能とレンジ、電圧センスモードなどを行っています。4つのSMUすべてが同様に構成されます。ノード1 SMUAのセットアップコマンドの一部を次に示します。

```
MASTER.reset()          --Reset all Node 1 logical instruments to default settings
IOUT1.sense = IOUT1.SENSE_REMOTE --Use REMOTE (4-wire) voltage sensing
IOUT1.source.func = IOUT1.OUTPUT_DCVOLTS --Configure SMU to source DCV
IOUT1.source.rangev = 0 --Set voltage source ranges;
                          --2602 picks appropriate range based on programmed value
IOUT1.source.levelv = 0 --To measure current, source zero volts on lowest range
IOUT1.source.limits = 1.2 * IoutMax --Set current compliance limit (20% over max)
IOUT1.measure.nplc = Nplc --Set integration times for all measurements
IOUT1.measure.autozero = IOUT1.AUTOZERO_AUTO --Autozero for max accuracy;
IOUT1.measure.rangei = IoutMax --Set up current measurement range; Measurement
                                --range for source function fixed at source range val
IOUT1.measure.filter.type = IOUT1.FILTER_REPEAT_AVG --Use REPEAT filter
IOUT1.measure.filter.count = 5 --Reading will be average of 5 consecutive measurements
IOUT1.measure.filter.enable = IOUT1.FILTER_ON --Enable Node 1 SMU A digital filter
                                --Set measurement parameters the 2602s will display
                                --(if display is enabled)
                                --Displays can be disabled to improve test speed
MASTER.display.screen = MASTER.display.SMUA_SMUB --Digital port isn't affected by reset so user
                                                    --must set desired initial state
DIO.writeport(0) --Set all digital control bits to zero
DIO.writeprotect(16128) --Write protect bits 9 through 14, which are reserved for
                          --component handler control in this example.
```

初期設定が完了すると、DAC試験が実行されます。ここでは、I_{OUT1}端子で行われるINL測定とDNL測定のみをご紹介します。その他の試験については、完全版のサンプルスクリプトを参照ください。

注意: ソースメータは、常に、内部ソースで電流を測定していると認識します。この例では、正電流電流が端子から流れ、負電流が端子に流れます。この結果、ノード1のSMUA、Bなど、「電流測定のみ」モードで動作するソースメータは、通常の電流計で想定される場合とは逆の極性で測定します。回路からソースメータに流れる正電流は負電流として測定されます。逆も同様です。

```
IOUT1.source.output = IOUT1.OUTPUT_ON --Turn ON SMU outputs
iout1 = {} --Declare table to hold measurements at output IOUT1; table index begins with 1

for j = 0, MaxCode do --j is the code applied to the digital inputs
    DIO.writeport(j) --Apply digital inputs
    delay(0.001) --Allow 1ms settling time
    iout1[j+1] = -IOUT1.measure.i() --Minus sign corrects for polarity of measurements
end --for

IOUT1.source.output = IOUT1.OUTPUT_OFF --Turn OFF outputs of all SMUs
```

測定が完了すると、ノード1のTest Script Processorはデータの計算と合否判定を行います。スクリプト言語には拡張演算ライブラリが用意されています。TSPで複雑な演算を行えるため、ホストコントローラにデータを送信して処理する必要はありません。完全版のサンプルスクリプトでは、2602型で線形回帰演算を実行する方法を紹介しています。

```
--Compute maximum integral nonlinearity(INL)
--Check for monotonicity; Compute maximum differential nonlinearity(DNL)
--Slope_bf and intercept_bf are the slope and intercept of the best-fit straight line
inlmax_iout1 = 0
dnlmax_iout1 = 0
mono_iout1 = "Monotonic"

for j = 0, MaxCode do
    inl_iout1 = math.abs(iout1[j+1]-(slope_bf * j + intercept_bf)) --Calcs for IOUT1
    if inl_iout1 > inlmax_iout1 then
```

```

    inlmax_iout1 = inl_iout1
end --if

if j > 0 then
  --Test for monotonicity
  diff_iout1 = iout1[j] - iout1[j-1]
  if diff_iout1 < 0 then
    mono_iout1 = "NON-Monotonic"
  end --if
  --Compute dnl and test for max.
  dnl_iout1 = math.abs(diff_iout1 - Lsb)
  if dnl_iout1 > dnlmax_iout1 then
    dnlmax_iout1 = dnl_iout1
  end --if
end --if
end --for

inl_iout1_lsb = inlmax_iout1 / Lsb      --Express INL and DNL in terms of nominal LSB
dnl_iout1_lsb = dnlmax_iout1 / Lsb

```

さまざまなDACパラメータが算出されると、TSPは値を検査し、パーツの合否状況を判定します。TSPは、デジタルビットパターンをノード1のDIOポートに書き出すことによって、適切なビニングコマンドを部品ハンドラに送信します。

```

if PartStatus ="GOOD" then
  DIO.writeport(GoodBitPattern) --Send "good part" bit pattern to component handler
else
  DIO.writeport(BadBitPattern)  --Send "bad part" bit pattern to component handler
end --if

```

すべての試験データはTSPによって処理、評価されるため、データをホストコントローラに送信する必要はありません。しかし、SPC用にデータ取得が必要な場合や、他のデータロギング、レコード保管の要求を満たす必要がある場合は、データを簡単にコントローラに送信できます。“print”関数は指定されたパラメータを、2602型の出力キューに書き出します。ホストコントローラは、その出力キューにデータをアップロードできます。データや試験結果を、測定器のフロントパネルに表示することができます。データのフォーマットには、標準の“C”言語のようなフォーマット文字列を使用できます。

```

--Send the monotonicity results and max INL and DNL values measured at IOUT1
print(string.format("%s, %1.2f, %1.2f", mono_iout1, dnl_iout1_lsb, inl_iout1_lsb))
--Display INL & DNL on front panel displays
MASTER.display.clear()
MASTER.display.setcursor(1,1,0)
MASTER.display.settext(string.format("INL= %1.2f LSBs", inl_iout1_lsb))
MASTER.display.setcursor(2,1,0)
MASTER.display.settext(string.format("DNL= %1.2f LSBs, dnl_iout1_lsb))

```

スクリプトのダウンロード／実行、データの取得

これまでスクリプトの概要をみてきましたが、ここでは、スクリプトを作成、ダウンロード、実行する方法について簡単に説明します。

DACの例では、TSP-Linkで相互接続した2台のソースメータを使用します。最初にしなければならないのは、TSP-Linkのセットアップです。TSP-Linkケーブルとして、Ethernetクロスオーバーケーブルを使用できます。バンド幅機能に優れているためCat 6ケーブルを推奨しますがCat 5eとCat 6のどちらのケーブルでも機能します。前述のとおり、TSP-Linkで接続された各測定器には、 GPIBアドレスと同様に、ユニークなノード番号が割り当てられます。ノード番号(1から64)は、フロントパネルから、またはリモート制御を介して、設定できます。DAC試験の例を実行するには、マスターノード番号を1、スレーブノード番号を2に設定してください(マスターは、ホストコントローラに接続しているソースメータです)。TSP-Linkを介した通信を行うには、バスを初期化またはリセットしておく必要があります。この作業も、フロントパネルから、またはリモート制御を介して行えます。TSP-Linkをリモートで設定するには、以下のようにします。

- TSP-Linkのステータスをチェックするには、“print(tsplink.state)”を送信します。応答は“online”または“offline”のいずれかになります。
- TSP-Linkがオフラインの場合は、“tsplink.reset()”を送信してリセット(オンライン化)します。
- ホストコントローラまたは“マスター” TSPからTSP-Linkをリセットできます。

ホストコントローラは個々のコマンドを2600シリーズに送信し、一般的なプログラマブル装置とやりとりできますが、2600シリーズの特長は、完全なテストスクリプトをロードして実行する機能です。明らかに、スクリプトは単なる測定器コマンドのリストにとどまらず、一般的なプログラム言語を使用して作成した場合と同様な完全なプログラムとして作成できます。スクリプトは任意のテキストエディタを使用して作成できますが、ケースレーではスクリプトの作成、管理、デバッグ環境としてTest Script Builderというアプリケーションを提供しています。このソフトを使用すれば、スクリプトを揮発性または不揮発性メモリに簡単にダウンロードし、実行できます。Visual Basic、Visual

C/C++、LabVIEWなど他の言語で作成されたスクリプトも、同様にダウンロード、実行できます。いったんスクリプトが測定器内蔵のメモリに格納されれば、フロントパネルからも実行できます。DAC試験のサンプルスクリプトをロードして実行する場合、DAC試験に必要なすべての関数が作成されます。作成される関数の1つであるTestDACは、TSP常駐のテストエクゼクティブとして機能し、DAC試験を実行する他のさまざまな関数を呼び出します。ホストコントローラがマスタユニットに“TestDAC()”を送信すると、テスト全体が実行されます。

2602型を使用すれば、処理のたびにデータをホストコントローラに送信する必要がなくなります。単純な“print”コマンドを使用すれば、必要に応じてデータを取得できます。出力キューからデータを取得するため、それぞれのPrintコマンドに対応するホストプログラムの“enter”構文を記述する必要があります。

装置リスト

DAC試験システムを構築し、ケースレーのWebサイトwww.keithley.comからダウンロード可能なサンプルテストスクリプトを実行するには、以下の装置が必要です。

1. ケースレー2602型2チャンネルシステムソースメータ 2台
2. PC、およびKPCI-488 GPIBインタフェースカード、またはKUSB-488 USB-GPIBアダプタ
3. ケースレー7007 IEEE-488インタフェースケーブル
4. CAT 5e または CAT 6 ENET ケーブル (TSP-Linkを介した、ソースメータ2台の接続用)
5. AD7523JN 8ビット乗算DAC (サンプルスクリプトはこのDACモデルを使用して検証)
6. テストフィクスチャ付部品ハンドラ (生産試験システムの一部として必要。スクリプトを実行するだけであれば不要。)
7. カスタムDB-25デジタルI/Oハンドラ・インタフェースケーブル (ソースメータとハンドラのインタフェース用。生産試験システムの一部として必要。スクリプトを実行するだけであれば不要。)
8. テストリード(ソースメータとテストフィクスチャの接続用)

テストシステムの安全性

多くの電気試験システムは、危険電圧と電力レベルの印加または測定機能を備えています。また、単一の故障状態(プログラミングエラーや測定器の不具合など)では、ハザードが存在しないとシステムに示されている場合でもハザードレベルの出力が可能です。これらの電圧や電力レベルは高いため、常にハザードからオペレーターを保護する必要があります。保護の方法には、以下があります。

- オペレーターが危険な回路に接触しないようにテストフィクスチャを設計する。
- 破片の飛散からオペレーターを保護するため、被測定デバイスを確実に囲む。たとえば、コンデンサや半導体に過電圧、過電力を適用すると、爆発する可能性がある。
- オペレーターが接触できる電気接続をすべて2重に絶縁する。2重に絶縁することで、1個の絶縁レイヤが破損した場合でもオペレーターは保護される。
- テストフィクスチャのカバーが開いた場合は、信頼性の高いフェールセーフ・インターロック・スイッチが電源を切るようにする。
- オペレーターが、テストフィクスチャの内部にアクセスしたりガードをオープンしたりする必要がない場所で自動ハンドラを使用する。
- システムのすべてのユーザーにトレーニングを実施し、潜在的な危険性、および事故から身を守る方法について理解させる。

試験システム的设计者、インテグレーター、敷設者は、かならずオペレーター、保守管理者の保護対策を確実に実施しなければなりません。

本資料の内容は予告なく変更されることがあります。

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

ケースレーインスツルメンツ株式会社 本社：105-0022 東京都港区海岸1-11-1 ニューピア竹芝ノースタワー13F •03-5733-7555 • Fax: 03-5733-7556
大阪：540-6107 大阪市中央区城見2-1-61 MIDIタワー7F •06-6946-7790 • Fax:06-6946-7791
Web: www.keithley.jp Email: info.jp@keithley.com