Companion

設計者の手引き：設計検証をより簡単に

Tektronix

Enabling Innovation

目　次

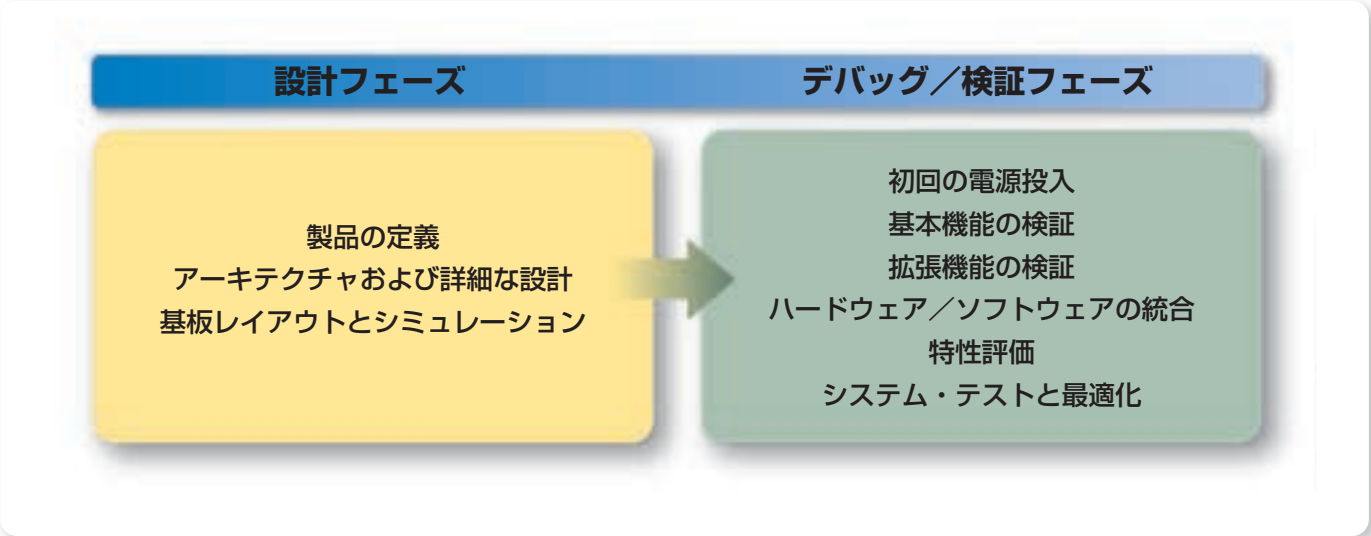


図1：*設計プロセスの概要*

1 はじめに

この入門書は、開発期間の短縮、コスト制約の要求に直面する設計現場にあって、最新のデジタル・システムのデバッグ／検証をよりシンプルにする必要に迫られているエンジニアの方を対象に書かれています。

設計ツールおよびEDA（Electronic Design Automation）ソフトウェアの進歩により、より複雑な設計を、より短期間に実行できるようになりました。デバッグ／検証ステージにおいても、厳しいスケジュールに対応するだけのプロダクティビティ（生産性）が要求されています。この入門書では、設計の検証をより高い生産性で実行し、問題が発生した場合のデバッグをより効率的に行うための手法を中心に説明しています。新規にマイクロプロセッサベースの組込みシステムを開発する場合の製品コンセプトの決定から設計、検証、製品完成までの工程を例に説明します。

1.1 設計プロセスの概要

図1は、システムを市場に投入するまでの、ハイ・レベルなプロセスを示しています。設計フェーズでは、コンセプトを検討、立案し、代替案も検討した後、最終案を決定します。デバッグ／検証フェーズでは、設計の正確さを検証し、機能、信頼性の両面で問題が発見された場合はこれを修正し、設計したものが確実に製造できることを確認します。

設計フェーズ

設計フェーズにはいくつかのステップがあり、各ステップで異なる作業と、対応する適切なツールが必要になります。アーキテクチャの決定にはシステム・シミュレーション・ツール、詳細な機械設計では熱解析ツールや高性能なモデリング・プログラムが、デザインではRTL（Register Transfer Language）コード、基板設計では配線用ソフトウェアが必要になります。

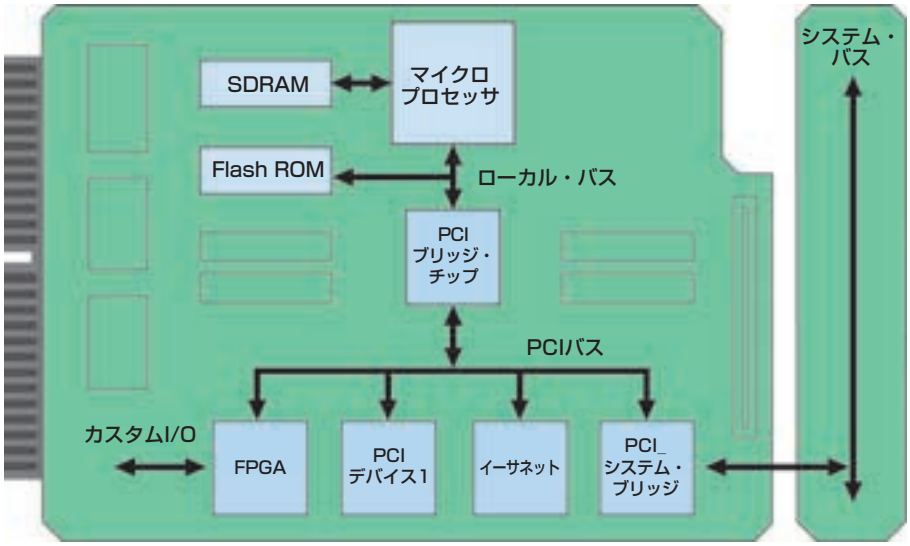


図2：*組込みシステムの設計例*

設計コンセプトが検討、決定されたら、ステップごとに詳細なプランニングで進めるか、または初期設定のままで行うかを決定します。これは、どちらがトータル的に検証に要する時間を短縮できるかによります。各ステップにおける決定が、設計検証、デバッグにどのように影響を及ぼすことになるかを理解することは非常に重要です。正しく理解することで、開発（早期に問題点を見つけ、手直しを最小にすること）に要する時間を短縮し、最適なタイミングで新製品を市場に投入することが可能になります。

デバッグ／検証フェーズ

エンジニアにとって最もわくわくし、報われる時間は、実験室で新しい基板に初めて電源を入れるときでしょう。また徐々にシステムとして組みあがってくるまでの数日間ではないでしょうか。プロダクティビティを低下させるような落とし穴に落ちることなく、デバッグ・フェーズ期間を期待しながら、またプロダクティビティを高めながら進めるためには、慎重なプラニングが必要になります。起こりそうな問題に注意しながらテスト手順をドキュメントにし、最新の測定ツールを理解しておくことで、わくわく感がフラストレーションにならずに済みます。

1.2 設計プロジェクトの概要

今回の設計例は、通信インフラストラクチャ機器、プリンタ、ビデオ機器などの複雑なシステムで多く見られる、一般的なプロセッサ・ベースの組込みシステムです。図2に示す設計例では、プロセッサとメモリ、プロセッサと周辺機器を接続するための内部コミュニケーション・バス、および周辺機器のI/Oデバイスで構成されています。コストに対する要求を満たすため、システムではSDRAM、PCI、イーサネット・インタフェースなど、十分にテストされ、広く利用されている技術を採用します。

高度に統合されたプロセッサは、166MHzで動作するSDRAMをサポートするメモリ・コントローラを内蔵することによってコストを下げることができます。この速度は、必要なメモリ帯域に対して十分ですが、シグナル・インテグリティに関する問題が発生する可能性があります。

PCIは、基板上のチップ間のバスとして使用します。新しいPCI Express規格による追加的な性能（および上がるであろうコスト）は必要ありません。要求される性能には、66MHz、64ビットのPCI実装が必要になります。このローカルPCIバスは、専用のシステム・バスとは分離され、FPGAによってブリッジされます。

イーサネット・ポートは、エンド・カストマとのコミュニケーション・パスとして、またデバッグ・フェーズにおけるデバッグ・ポートとして使用します。最新のギガビット・イーサネットよりも、コスト効率の良い10/100ビット・イーサネットを実装する予定です。

この設計では最新の技術は採用しませんが、新しい基板の各機能をすべて検証し、バグを見つけて修正し、設計の特性評価も実行する必要があります。このためには、柔軟性があって使いやすいリアルタイム・オシロスコープ、ロジック・アナライザが必要であり、システムを詳細に観測、解析できるよう、この2つの計測器を効果的に統合するための機能も必要になります。

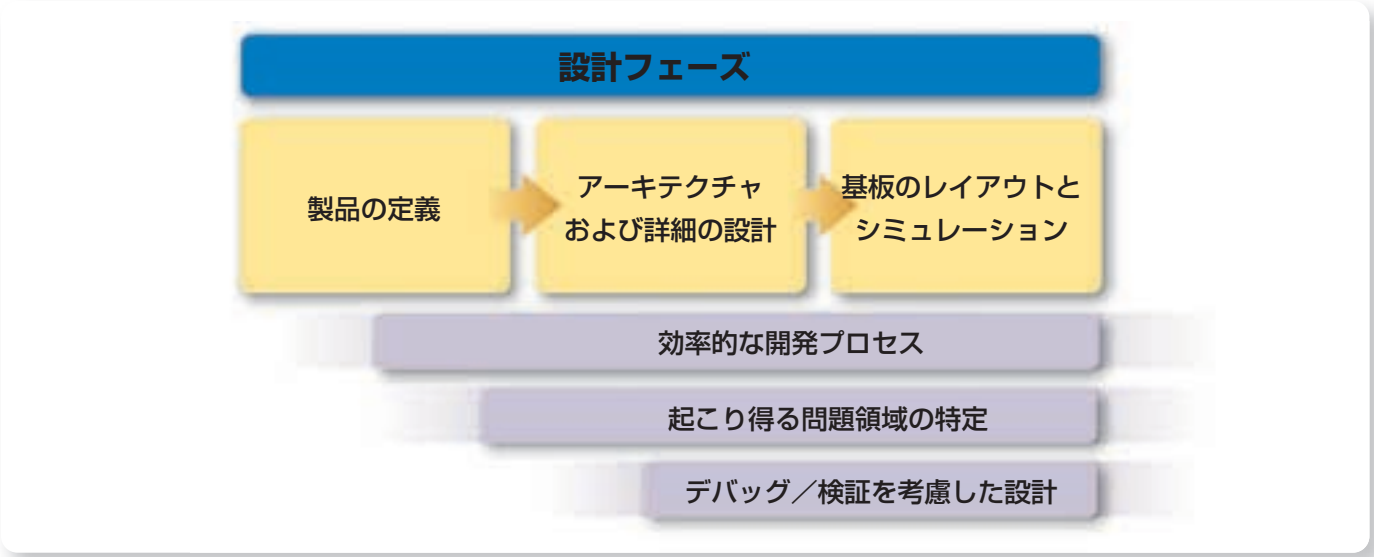


図3：*設計フェーズの概要*

2 設計フェーズ

設計フェーズでは、図3の左から右に示すように設計概要から順番に進めていきます。各ステップでは、アーキテクチャ・ブロックの定義、各パートの詳細設計、回路基板のレイアウトなどの詳細な設計が追加されます。

各ステップでは、全体のプロダクティビティに影響を及ぼすような判断がなされます。信号へのアクセス方法、ソフトウェア・デバッグ・ツールとの接続など、はっきりしているものもありますが、テクノロジの選択、コンポーネントの選択、機械的なパッケージング・コンセプトなど、はっきりしないものもあります。最高のプロダクティビティを得るには、以下の項目が重要になります。

- 効率的な開発プロセスの導入
- 起こり得る問題の特定
- デバッグ／検証を考慮した設計

2.1 効率的な開発プロセス

どの製品開発チームも、ドキュメント化された開発プロセスが求められます。このプロセスでは、顧客の満足が確実に得られるよう、必要なチェック項目、予防手段を定義しています。顧客の要求を取込み、検証するメカニズムを定義する必要があります。全体を通して製品コストや開発コストをレビューし、望まれる利益が得られなければなりません。アーキテクチャのレビュー、設計のレビュー、コードのレビューのためには、最高の技能を集結させる必要があります。

開発プロセスに忠実に従おうとすると、当初はオーバヘッドが増加することでプロダクティビティが低下するように感じることがあります。しかし実際には、明確に定義された（しかし、柔軟性のある）開発プロセスにより、アーキテクチャの明確な定義、また基板の改変やオペレーション・システムを作成するのに必要なソフトウェア・リリース回数が削減することなどにより、プロダクティビティが大幅に向上します。

2.2 起こり得る問題の推測

起こり得る問題を推測するのは難しいことです。過去の経験より多くのことを学び、設計を重ねるたびに見識は広がります。しかし、これが初めての設計の場合はどうでしょう。あるいは、シグナル・インテグリティの問題を考慮する必要のある高速なクロック周波数、エッジ・レートで設計を依頼された場合はどうでしょう。起こり得る問題には、大きく分けて、機能的な問題と波形品質に関する問題の2種類があります。

機能的な問題

機能的な問題の原因としては、購入したコンポーネントの動作に対する誤った理解、FPGAで実装されているRTLレベルでのエラー、あるいはハードウェア／ソフトウェア間の相互動作に関する不具合などがあります。

効果的な開発プロセスを持っているチームは、設計レビューや基板レベルでのシミュレーションにおいて機能的な問題を十分に把握することができます。設計図上での検証は、電源の配分、クロックの配分、また双方向バッファのコントロール信号の反転ロジック・レベルなどの一般的な問題を調べるには適しています。しかし、設計図上だけでは発見することが難しい問題も数多くありますので、あまり効率的ではありません。

ゲートの数が100万個を超えるような場合、製品化までの時間が短縮できることなどから、FPGAが数多くの機能を実装するような最新のシステムで広く使用されています。設計ツールを使用することで、抽象度の高い概念設計や、複雑な設計をすばやく合成して、配置配線を短時間で実行することが可能になります。これに対し、テスト・ベンチを設計したり、スティミュラス・モデルを書いたり、テスト・ケースを管理したりすることには、プロダクティビティに限界があります。現実には、設計フェーズで発見された問題点やバグは、簡単にまた安価に修正することができます。問題を早期に発見できればデバッグ時間は短縮でき、開発コストも低減できます。しかし、シミュレーションには限界があります。例えば、クロックの境界で同期をとる信号に関する問題は検出することが難しく、テスト・ケースにおいてはしばしば完全ではなく、複雑なハードウェア／ソフトウェアの相互関係ではモデル化することも困難です。

シグナル・インテグリティに関する問題

シグナル・インテグリティに関する問題は、ノイズ、歪み、異常波形など、アナログ領域で信号を劣化させることから起こります。信号パス設計、インピーダンスや負荷、伝送ライン効果などの変動する要素によってもシグナル・インテグリティは影響を受けます。また、回路基板の電源供給によっても影響を受けます。これらの問題を最小にし、問題が起こった場合にこれを解決するのは設計者の責務です。

信号を劣化させる大きな要因には以下の2つが考えられます：

- デジタル問題 ── おもにタイミングに関連します。バスの競合、セットアップ／ホールド時間違反、メタステーブル、レース・コンディションなどが、バスまたはデバイス出力における異常信号の原因となります。
- アナログ問題 ── 小振幅信号、低速または高速の遷移時間、グリッチ、オーバシュート、クロストーク、ノイズ。これらの現象は、主に回路基板の設計または信号終端の問題により起こりますが、他の原因も考えられます。

デジタルとアナログ間のシグナル・インテグリティ問題には、相互作用および相互依存関係が作用するものです。例えば、ゲート入力における立上り時間の遅延は出力パルスを遅らせ、その結果として、デジタル環境においてバス競合を起こします。シグナル・インテグリティ検証とトラブルシューティングの完全なソリューションには、デジタルとアナログの測定ツールが必要になります。

経験の豊富なエンジニアは、波形品質の維持は設計プロセスにおける継続的なチェックの成果として実現できることを認識しています。設計プロセスが進むにつれてシグナル・インテグリティに関する問題は複雑になり、調べることはより難しくなります。最初の試作基板で気付かなかったわずかな異常であっても、その基板が他のシステムと組合わされたときに、システム全体をクラッシュさせることがあります。

このような現実において、どこからシグナル・インテグリティ検証を始めたらよいでしょうか。設計フェーズにおいては、本当に早期からシグナル・インテグリティを考慮する必要があります。組込みシステム設計では、共通の項目がいくつかあります。まず、クロック信号の適切な供給は重要です。クロックがどのように生成され、基板においてどのように供給されるかは、EMI（電磁妨害）からタイミング要求に対するマージン（または不足）に影響を及ぼします。アーキテクチャ定義の早い段階における決定は影響を及ぼします。コンポーネントの選択も影響を及ぼします。スキューを排除して基板全体にクロックを供給するために、一般的なバッファ・デバイスを使いますか、それともPLLがビルトインされた特別なICを使いますか。

システム設計で慎重なプラニングが必要になるもう一つの要素が、電源供給です。これは、各地域により規制される電圧への変換、アナログ部においてきれいな電源かどうか、基板構成など、電源設計のすべての要素が含まれます。

2.3 デバッグ／検証を考慮に入れた設計

設計において直面すると思われる問題の種類を理解して、検証およびテスト・プランを進めます。以下のことを考慮に入れて計画することで、起こり得る障害を排除することができます。

── テストする機能とその実行方法を確認する

── 検証するインタフェースと信号を確認する

── 測定項目を確認する

この計画は設計フェーズで立案すべきです。設計フェーズでデバッグや検証を考慮せず、効果的なデバッグが制限されることは避けるべきです。今日の複雑な設計では、トラブルシュートやデバッグの必要性を無視することはできません。

デバッグを簡単に行うためには、どのような設計をすればよいでしょうか。理想的には、ロジック・アナライザとオシロスコープで簡単にプロービングできるようなプロービング・ポイントを設けることです。しかし、どこにでもプロービング・ポイントを設けることができればよいのですが、現実には難しい問題です。基板のスペースで制約を受けながら、デバッグのための設計も行う必要があります。限られたスペースで必要とされる機能を盛り込み、さらにテスト・ポイントを追加しなければなりません。どのように解決すればよいのでしょうか。

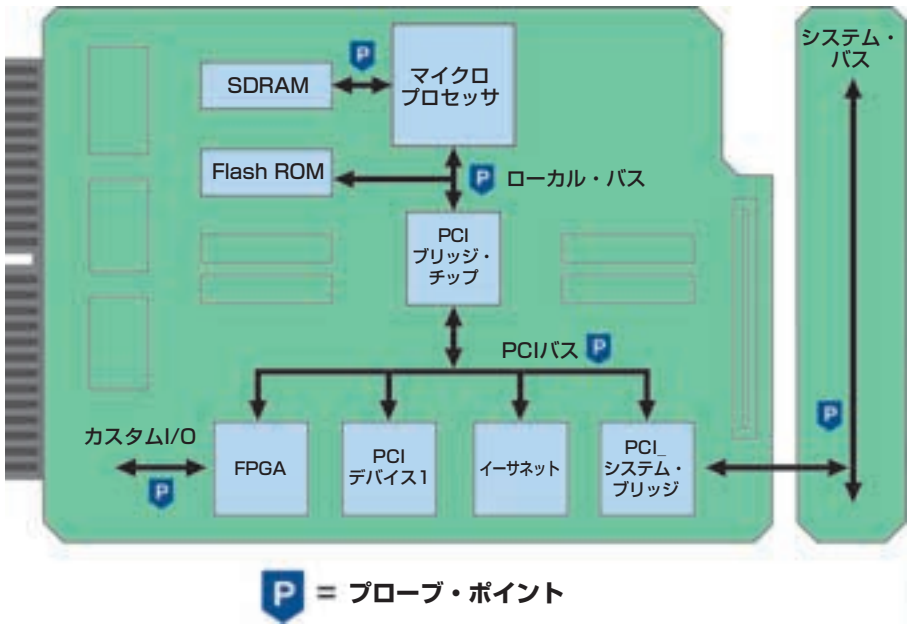


図4：*ロジック・アナライザのプローブ・ポイント*

2.3.1 設計例

ここで示した設計を例に、まずいくつかの質問から始めます。

- マイクロプロセッサの可視性はどの程度重要でしょうか。ハードウェアの検証を中心に考えるべきでしょうか。ソフトウェアのデバッグも重要ですか。
- どの内部バスを観測しますか。オシロスコープのテスト・ポイントで観測しますか、ロジック・アナライザのテスト・ポイントでも観測しますか。
- どの設計マージンが重要ですか。それをどのように検証しますか。温度などの環境要素は考慮する必要がありますか。
- 信号が測定できない場合はどうしますか。こうなった場合、スケジュールに影響しますか。基板を作り直さなければなりませんか。
- テスト・ポイントやプローブは、信号にどのような影響がありますか。過度の容量負荷によって、期待通りに動作しないことはありませんか。

まず始めに、図4に示すように、ロジック・アナライザですべてのバスにアクセスしてみます。

ローカル・バス ― ローカル・バスにアクセスすることで、ブートに関する問題をモニタ、デバッグできます。ここで初めてハードウェアとソフトウェアが同時に動作することになります。

SDRAMインタフェース ― システム・データ構造とシステム・コードはSDRAMに保存されます。ソフトウェアの実行をリアルタイムにトレースできる可視性が提供されます。これにより、ソフトウェアの性能解析とハードウェア／ソフトウェアの相互作用のデバッグが可能になります。

PCIバス ― 性能のゴールとは、PCIの周波数帯域が確実に利用可能かということになります。ローカルのPCIバスに簡単にアクセスできることは、スループットに関する問題を解決するだけでなく、システム内の2つのFPGAの機能デバッグも可能になります。

図5：*汎用ロジック・アナライザ・プローブ*

図6：*高実装密度ロジック・アナライザ・プローブ*

カスタムI/Oインタフェース ── 一つのFPGAが機能を実装することで、競合他社と差別化することができます。シミュレーションだけでは、このFPGAにおける機能およびタイミング関係の問題を捉えることはできません。

システム・バス ── この独自バスは、システムでキーとなります。すべてのシステム・バスを取込むことができると、システムに関する問題は容易に解決できます。

どのようにしてロジック・アナライザをこのポイントに接続しますか。

2つのオプションがあります。一つは、汎用プローブ（図5を参照）を使用することです。比較的少ない信号の観測には便利ですが、PCIなどのバスで数多くのプローブを接続する場合は、接続自体の手間と信号品質の劣化を考慮する必要があります。

もう一つのオプションは、コネクタやアクセス・ポイントを直接基板に追加することです。これにより、汎用プローブの接続自体の手間と信号品質の劣化を防ぐことはできます。しかし、貴重な基板スペースを使うことになります。図6は、代表的な高実装密度プローブと基板への実装の様子を示しています。

図7：*ロジック・アナライザ用マイクロプロセッサ・アダプタ（写真提供：Ironwood Electronics社）*

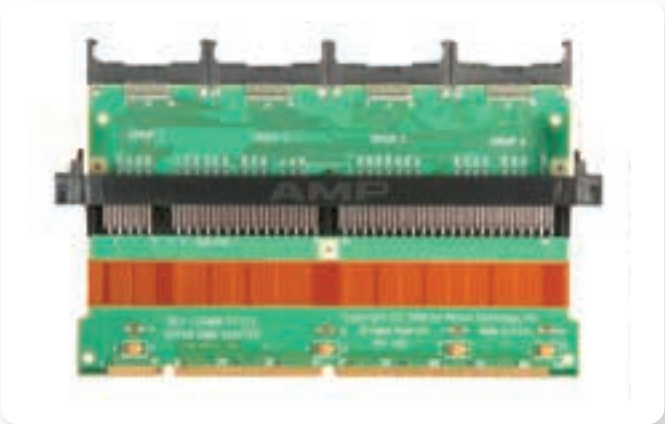


図8：*ロジック・アナライザ・プロービング用DIMMインターポーザ（写真提供：Nexus Technologies社）*

2.3.2 プロービングの決定

ここで使用するインタフェースのバス幅を考えると、高実装密度ロジック・アナライザ・プローブの使用が良いように思われます。ここで、基板には十分なスペースがないことがわかりました。しかし、アクセス・ポイントの可能性を排除する前に、すべての代替案を検討する必要があります。基板のスペースを犠牲にすることなくプロービングする方法があるかもしれません。

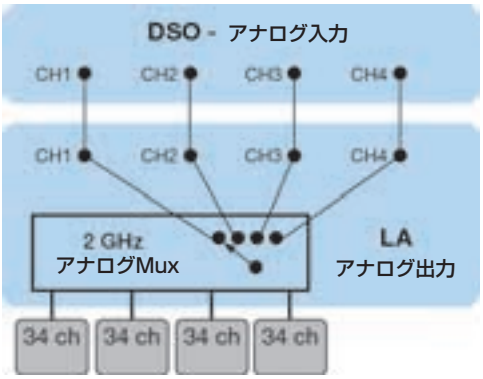
例えば、マイクロプロセッサにプロービングする場合プローブ・アダプタを使うことができます。通常、このタイプのアダプタは、基板上のマイクロプロセッサのある場所に半田付けされます。マイクロプロセッサは高速なソケット上に取り付けられ、ロジック・アナライザのテスト・ポイントは基板の周辺に追加されます。代表的なアダプタを図7に示します。アダプタを使用することで基板のスペースを節約でき、重要なローカル・バスとSDRAM信号にアクセスできます。残念ながら、今回使用するマイクロプロセッサに合うアダプタはありませんので、ローカル・バスへのアクセスには高実装密度ロジック・アナライザ・プローブを使用します。

SDRAMインタフェースのアクセスには、2つのオプションがあります。一つは、DIMMソケットを考えている場合はインターポーザの使用が考えられます。代表的なインターポーザを図8に示します。

DIMMソケットが利用できない場合は、アクセス・ポイントは基板設計に含めて考える必要があります。今回は一つのDIMMソケットを使用しますので、インターポーザ基板を使ってSDRAMにアクセスします。

図9：*ロジック・アナライザ・プロービング用PCIインターポーザ（写真提供：Nexus Technologies社）*

同様のオプションがPCIバスにも当てはまります。代表的なPCIインターポーザを図9に示します。今回のPCIバスは組込みバスであり、PCIスロットはありませんので、PCIインターポーザは使用できません。内部のPCIバスを確認するためには、テスト・ポイントを基板に設計する必要があります。



- シングルポイントによるデジタルとアナログのプロービング
- すべてのチャンネルで2GHzのアナログ帯域
- 136のロジック・アナライザ入力チャンネルから、任意の4チャンネルを4つのアナログ出力BNCにマルチプレクス
- アナログのプローブ出力は常にライブ波形

図10：*iCapture™によるマルチプレクス*

各バス、インタフェースは、順次このように検討します。検討時には、先に説明したような検討項目を自問自答してみます。検討した結果、この基板の例ではデバッグ・プランを表1のようにまとめます。

2.3.3 プロービングのまとめ

必要とされる十分なアクセス・ポイントを設けられないこともあり、このような場合は最も影響が少なくなるようにアクセス・ポイントを減らす必要があります。幸いなことに、この例では、上記に示したプランで使用するのに十分な基板スペースが確保できています。

オシロスコープによるアクセスはどうでしょうか。ここでロジック・アナライザによりアクセスされるインタフェースには含まれないものの、アクセスが必要になる信号があるかもしれません。

考慮しなければならない信号としては、電源、クロック、リセット信号などがあります。オシロスコープのプローブのためには、アクセスしやすい、適切な数のグランド・ポイントが必要になります。

バス	アクセス方法
ローカル・バス	設計に組込まれた高実装密度ロジック・アナライザ・プローブ・ポイント
SDRAM	DIMMインターポーザ・カード
内部PCIバス	設計に組込まれた高実装密度ロジック・アナライザ・プローブ・ポイント
FPGA I/O	設計に組込まれた高実装密度ロジック・アナライザ・プローブ・ポイント
システム・バス	カスタム・インターポーザ・カード

表1：*ロジック・アナライザのためのアクセス・ポイント・プラン*

オシロスコープのプローブの代わりに、デジタルとアナログの両方の情報をプロービングできるロジック・アナライザもあります。アナログ信号を選択して、ロジック・アナライザからオシロスコープに送ることができます。基本的な概念を図10に示します。

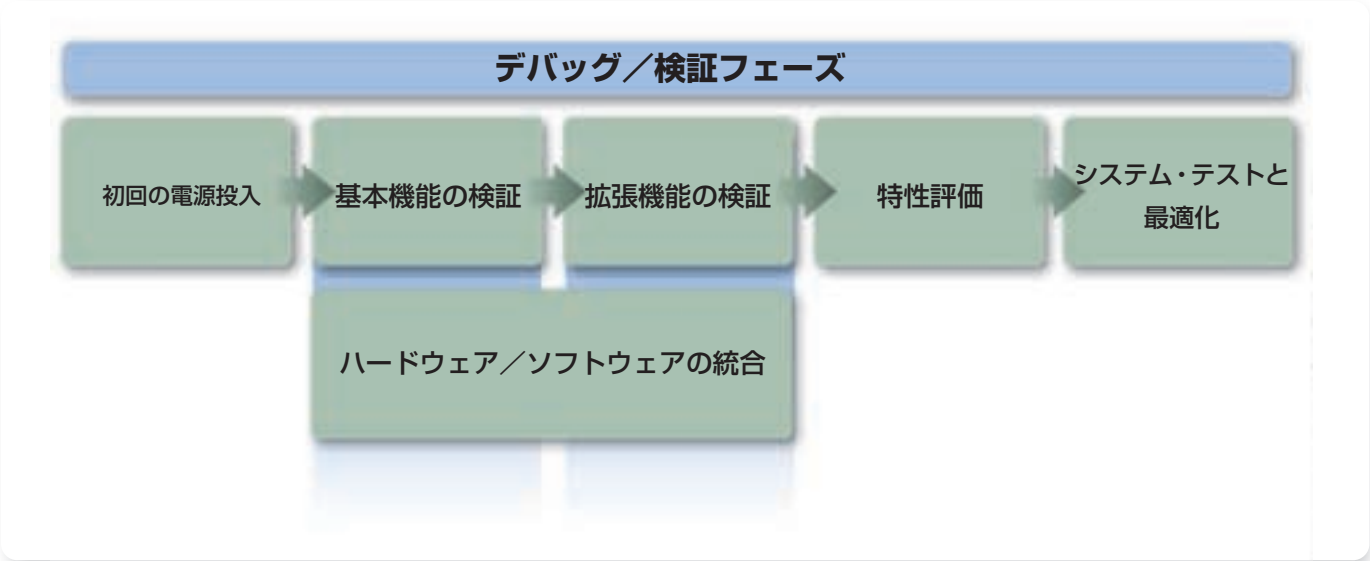


図11：*デバッグ／検証の概要*

2.4 まとめ

この章では、効果的なデバッグと検証を製品開発サイクルの設計フェーズから始めることを説明しました。基板が組みあがるまでデバッグと検証を行わない場合は、デバッグと検証が不可能ではないにしても、非常に難しくなることがあります。

効果的な設計プロセスは、デバッグと検証時間を最小にするための第一ステップです。ドキュメント化され、実行される設計プロセスの最終的な目的は、できるだけ早期に起こり得るエラーを数多く検出、修正することであり、後に続くデバッグ／検証フェーズでこれを実行しないことにあります。

どこで問題が発生するかを理解することが次に重要となります。機能的なバグとシグナル・インテグリティ問題が特に重要になります。起こり得る問題のためには、クロックの供給、リセットおよび電源は十分に検証する必要があります。高速回路は慎重に配置、配線することで、シグナル・インテグリティ問題を最小にすることができます。

最後に、適切なテスト・アクセス・ポイントを設けることで、デバッグと検証が容易になります。設計フェーズでデバッグ／検証のための設計を行うことは、ロジックが正しく機能することと同様に重要です。

3 デバッグ／検証フェーズ

十分な作業と慎重なプラニングの後、基板が届きます。これからデバッグと検証が始まります。デバッグと検証には、図11に示すように6つの基本的なステップがあります。製品が出荷される前に、各ステップを実行しなければなりません。それでは、個々のステップを見ていきましょう。

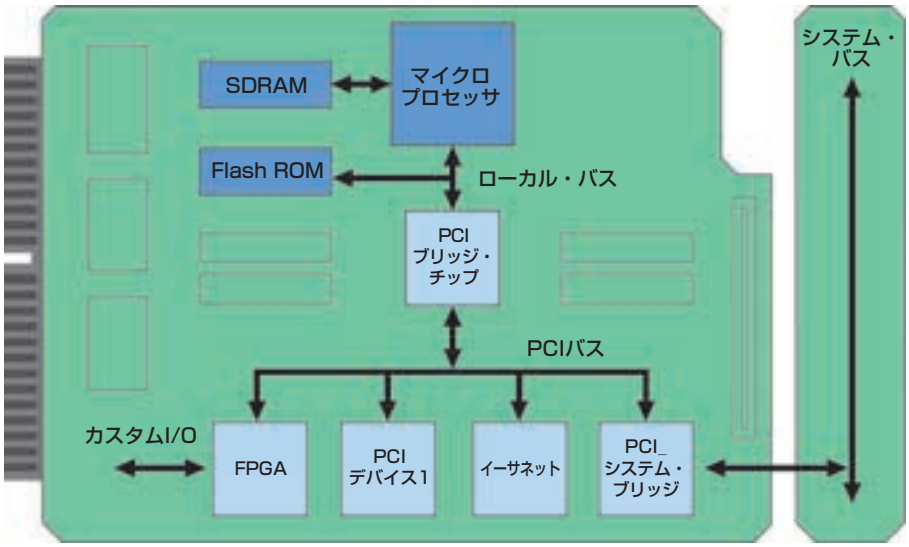


図12：*マイクロプロセッサのための基本機能検証*

3.1 初回の電源投入

発煙テストは最も簡単なステップです。電源とグランドがショートしていないことを確認した後、基板の電源をオンします。発煙があるようでは、当然のことながら不良です。

3.2 基本機能の検証

実質的な最初のステップが基本機能の検証です。このステップでは、設計のコアとなる部分を実行します。マイクロプロセッサは動作しているか、電源をオンにすると期待通りの動作が実行されているか、クロックは動作しているか、リセットは正しく機能するか、電源は基板全体に正しく供給されているかなどをチェックします。これらの基本機能を検証したら、マイクロプロセッサが正しくブートするかを確かめる必要があります。意図的にシステムの残りの部分は無視し、図12に示す設計のコア部分に集中します。

うまく動作しない原因は？数多く考えられます。起こり得る機能のエラーやシグナル・インテグリティ問題に対処するだけでなく、基板の製造エラーにも対処する必要があります。

この段階で効果的に問題をデバッグするには、互いの信号の関係を正確に観測する必要があります。この段階では詳細な解析は必要ありませんが、信号の動作を視覚化できることが必要になります。設計ノートやホワイトボードに書かれたシーケンスと合っているかどうか、システム・リセットが解除されているボルト値を確認する必要があります。電源シーケンスを確認し、マイクロプロセッサのブート・シーケンスとローカル・バス・サイクルを視覚化する必要もあります。そのためには、リアルタイム・オシロスコープとロジック・アナライザが必要になります。オシロスコープとロジック・アナライザを使用することで、設計結果を確実に確認することができます。

まずは、リアルタイム・デジタル・ストレージ・オシロスコープ（DSO）またはデジタル・フォスファ・オシロスコープ（DPO）を使用することになります。どちらのオシロスコープも必要な性能（周波数帯域、サンプル・レートなど）を満たしているだけでなく、豊富なトリガ機能やプロービング・オプションも提供しています。これらのリアルタイム計測器では、電源のノイズから高速信号まで、テスト・ポイントに簡単に接続でき、確実に信号を取込むことが最も重要です。

DSOは、高速エッジまたは狭いパルス幅をもった、低速または高速の繰り返しレートを持つ信号に適しています。DSOはまた、電源シーケンスやリセット動作などの単発イベントや過渡現象の測定にも最適です。この設計プロジェクトでは、TDS6000シリーズが最適なソリューションです。

デジタル回路のトラブルシュートと間欠的に発生する信号の検出には、DPOが最適です。DPOの優れた波形取込みレートにより、どのオシロスコープよりも高速に、次々に取込まれる情報をすばやく重ね書きし、発生頻度をカラーではっきりと表示することができます。この例では、デジタル・システムの設計アプリケーションの要求に対して、TDS5000Bシリーズが優れたソリューションとなります。

オシロスコープによる高速信号測定では、プローブの選択が重要になります。基板から正確な信号が取込めないと、どのような優秀なエンジニアであっても問題を効果的にデバッグすることはできません。設計フェーズで検討したデバッグ・プランを考慮した設計では、正確な信号取込みをするための鍵となるのは明白ですが、適切なプローブを使用することも重要になります。プローブの周波数帯域は、オシロスコープの帯域と同じであることが望ましく、また、信号に与える負荷が最小になるようなプローブを選択します。理想的には、プローブを含む測定システムの周波数帯域は、測定する信号の少なくとも 3 倍の周波数帯域になるようにします。次のステップでは、電源を検証します。

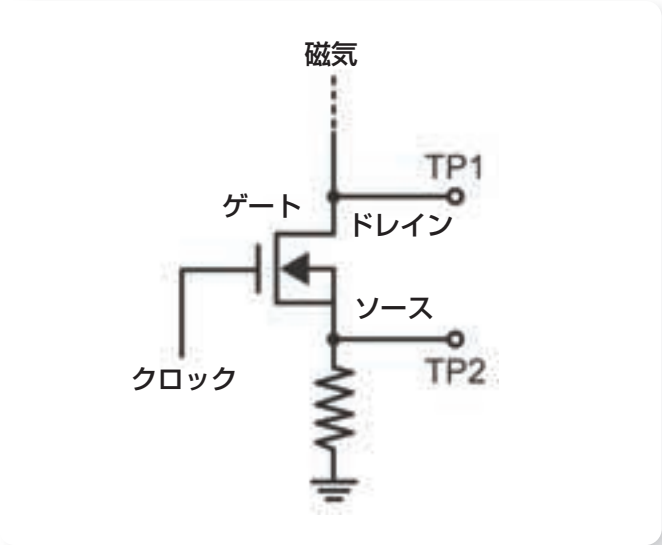


図13：*スイッチング電源（SMPS）*

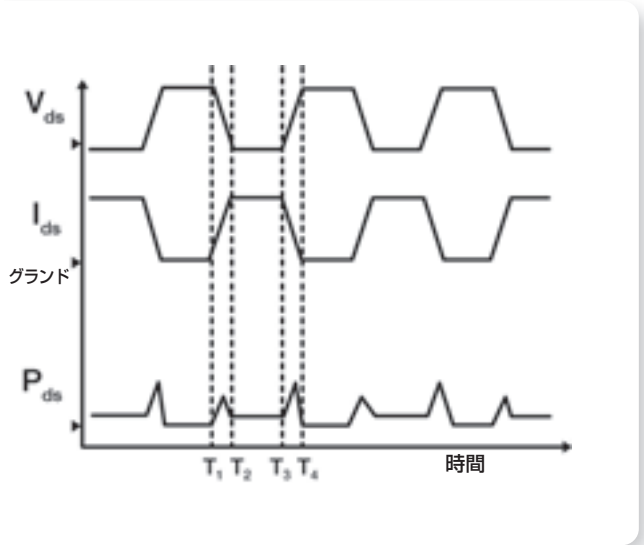


図14：*スイッチング・デバイスの動作*

3.2.1 電源解析

この設計例で使用されている電源は、負荷変動に効率的に対応できるスイッチング電源です。電源信号の経路は、パッシブ・コンポーネント、アクティブ・コンポーネント、磁気コンポーネントから成ります。図13は、スイッチング電源の簡略図であり、電力変換部をアクティブ要素、パッシブ要素および磁気要素で示しています。

スイッチング電源技術は、MOSFET（Metal Oxide Semiconductor Field Effect Transistors）やIGBT (Insulated Gated Bipolar Transistors) などのパワー・スイッチング・デバイスに基づいています。これらのデバイスは、高速なスイッチング時間と不規則な高電圧スパイクに耐える性能を持っています。また、オン／オフ時の低電力損失、発熱が少なく高効率など、優れた性能も持っています。スイッチング電源の全体としての性能は、多くの場合、スイッチング・デバイスで決定されます。

TDS6000シリーズを使用して、スイッチング電源の2つの重要な要素のスイッチング損失と電源リップルを定量化してみましょう。

3.2.1.1 電源のスイッチング損失

電源のスイッチング損失は、電源の効率を左右しますので、スイッチング損失測定は非常に重要で、迅速に実行できることが求められます。スイッチング損失の測定では、電源が安定して動作している状態と、ダイナミックに負荷が変動している状態の両方において解析する必要があります。

スイッチング・デバイスにおける電力損失を測定するため、まず図14に示すスイッチング・デバイスの信号を見てみます。

デバイスがオフのときはスイッチング・デバイス間の電圧は高くなり、導通状態（オン状態）では低く（V saturation）なります。デバイスがオフの状態では、電流は流れません。しかし、導通状態では流れる電流値は最大になります。電力波形を観測すると、瞬時最大電力損失はトランジション時に発生します。スイッチング・デバイスがオフからオンに遷移する期間における電力損失を、ターン・オン（T_{on}）損失と呼びます。オンからオフに遷移する期間における電力損失を、ターン・オフ（T_{off}）損失と呼びます。

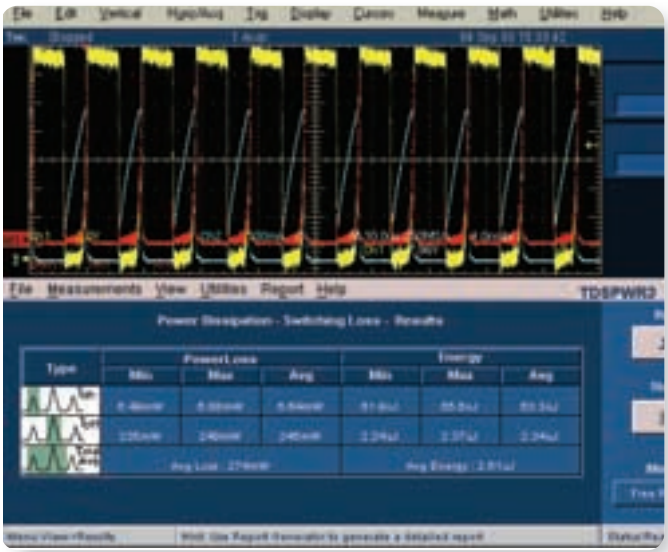


図15：*スイッチング損失測定例*

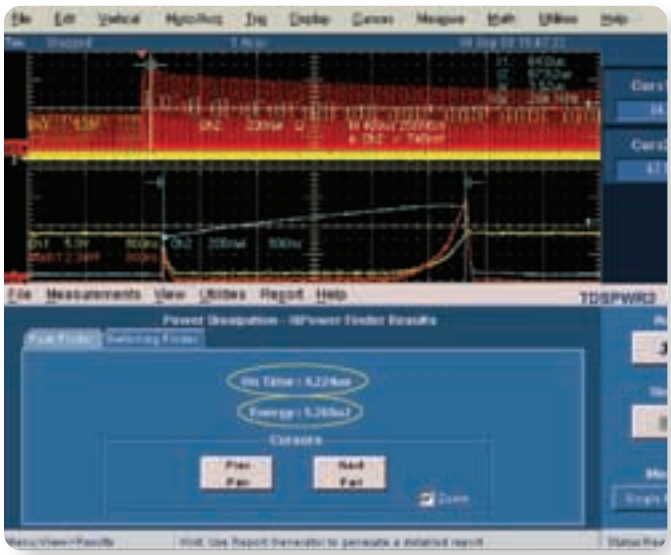


図16：*HiPower Finderによる測定例*

導通期間における電力損失は、定常損失と呼びます。整数回数のサイクルにおける電力損失は、トータル平均電力損失と呼びます。上の図では、T1からT2における電力損失がターン・オン（T_{on}）損失、T3からT4における電力損失がターン・オフ（T_{off}）損失、T1からT4における損失がサイクル損失になります。

TDSPWRアプリケーション・ソフトウェアを組込んだTDS6000シリーズでは、ボタンを一回押すだけで、この測定を自動的に実行することができます。

図15に示す測定から、設計を最適化するための情報が得られます。例えば、T_{on}損失とT_{off}損失からは、トータル電力損失が低減できるかがわかります。トータル平均電力損失値からはヒート・シンクの最適設計が、また、突出した電力損失がないか観測することで、電源の信頼性が解析できます。

多くのシステムでは、電源は、時間による負荷変動に対応して電源を供給することが必要になります。負荷が変化すると、スイッチング・デバイスにおけるスイッチング損失は変化します。瞬時電力損失が規定値内に入っていることを確認するためには、イベントを取込み、その電力損失を解析する必要があります。TDSPWRソフトウェアのHiPower　Finder機能ではこの測定を自動化し、必要な情報を図16のように表示します。

3.2.1.2 電源リップル

最後に、電源のリップルを解析します。リップルは、出力電圧の不要な周波数成分です。リップルに関しても、ボタンを押すだけで図17に示すような結果が得られます。

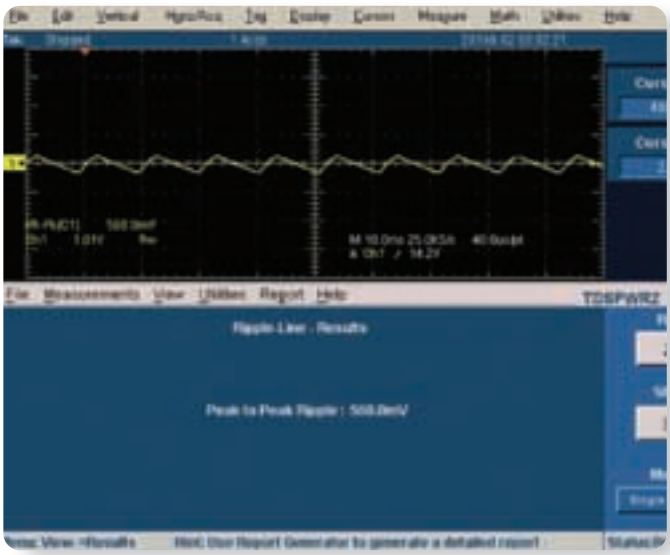


図17：*電源のリップル測定*

3.2.2 基本機能の検証

DSO、DPOは基本機能検証において大いに役立ちますが、十分でない場合もあります。どのような場合に十分とは言えないのでしょうか。

3.2.2.1 マイクロプロセッサのリセット・デバッグ

最も一般的な状況としては、4本以上の信号のタイミング関係を観測する場合があります。例えば、マイクロプロセッサがリセット状態から開放された後、5～7本のコントロール信号、さらにアドレスとデータ・ラインをモニタする必要があります。8ビットのコントローラでは21本の信号に、32ビットのマイクロプロセッサでは75本の信号になります。ポータブル・タイプのTLAシリーズ・ロジック・アナライザは、34、68、102、136チャンネルの入力取込みが可能であり、この種のデバッグに最適です。さらにチャンネル数が必要な場合は、TLA7000シリーズ・ロジック・アナライザ・モジュールが用意されています。

各信号のタイミング関係は、ロジック・アナライザの非同期クロック・モードを使用して観測します。非同期クロック・モードでは、ロジック・アナライザの内部クロック信号によって被測定システムからデータを取込みます。すべてのサンプル・データは、一定の時間間隔で取込まれます。非同期クロック・モードによる取込みは、タイミング・アクイジションとも呼ばれます。

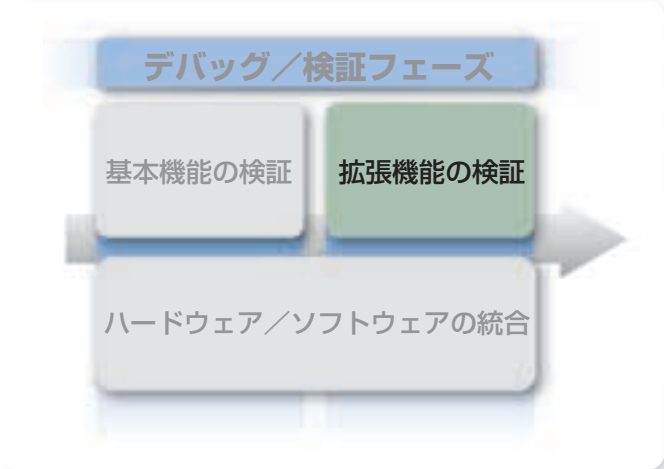
3.2.2.2 マイクロプロセッサのブート・デバッグ

オシロスコープでは十分でないもう一つの状況は、マイクロプロセッサやバスの動作をモニタすることです。例えば、マイクロプロセッサのブート動作を検証、デバッグするには、ロジック・アナライザを使用してマイクロプロセッサとフラッシュROM間のインタフェースをモニタし、ブート・コード動作を表示します。これにより、75もの信号を波形データとしてだけではなく、データを抽象化してシステムの動作として確認できます。

マイクロプロセッサのブート・シーケンスを観測するには、ロジック・アナライザの同期クロック・モードを使用します。同期クロック・モードでは、ロジック・アナライザはデータを取込む際のクロックとして、被測定回路のクロックを使用します。クロック信号は、固定の周波数であったり、非常に不安定であったりします。同期クロック・モードによる信号取込みは、しばしばステート・アクイジションとも呼ばれます。

3.2.3 まとめ

基本機能の検証フェーズでは、設計における重要な機能を検証します。信頼性の高い機能動作が得られないと、その他の機能やインタフェースは検証できません。基本機能の検証では、電源、リセット、クロック、重要なコントロール信号などを検証します。この作業には、オシロスコープとロジック・アナライザが適しています。



拡張機能の検証

3.3 拡張機能の検証

最小限の、しかし安定したシステムが動作したならば、別のブロックを検証する必要があります。理想的には1度に、1ブロック、1機能、1インタフェースごとに検証します。一つが終わったら、次に進みます。設計のコア部分の次に検証するのが、ローカルPCIバス・ブリッジ・チップまでのローカル・バスです。この検証では、ドライバがブリッジ・チップを正しく初期化し、プログラムしているか、これにより適切なバス・サイクルが生成されているか、タイミング・パラメータは合っているかなどを確認します。

3.3.1 統合ツールの使用

設計をスケジュール通りに進めるためには、問題の本当の原因を迅速に突き止める必要があります。アナログなのか、デジタルなのか。このためにはアナログ、デジタル両方の領域で利用できるツールやトラブルシュートの手法が必要になります。これには、すでに使用している測定機器、つまりDPO、DSOなどのオシロスコープとロジック・アナライザの組合せが適しています。すでにご説明したように、DSOは、グリッチ、信号の歪み、遷移時間、クリティカルなセットアップ／ホールド時間など、個々のイベント測定に最適なツールです。ロジック・アナライザは、システム内のロジック信号を、タイミング情報とともに基本形式であるバイナリ値として取込みます。アナログとデジタルという、2つの領域間の相互関係を取込むことは、効果的なトラブルシュートにおいて非常に重要です。

図18：*iLink® ツールセット*

当社TLAシリーズ・ロジック・アナライザやTDSシリーズDPOなどの最新の計測ソリューションには、2つの計測器を統合し、トリガや時間相関をとった表示機能が装備されています。図18に示すiLink®ツールセットは、ロジック・アナライザとオシロスコープを統合させることができます。この章では、これらの計測器を組合せ、詳細な設計問題に取り組む方法について説明します。

iLink®ツールセット： 2つの強力な計測器を統合

ロジック・アナライザとオシロスコープは、デジタル分野でのトラブルシュートのためのツールとして長年使われていますが、すべてのエンジニアがこの2つの計測器を統合した場合の優れた機能を使いこなしているわけではありません。

ロジック・アナライザを使用することで、デジタル・ストリームにおける回路の問題点にトリガし、関連するイベントを取込むことにより、デバッグや検証を迅速に実行できます。オシロスコープを使用することで、デジタル・タイミング・ダイアグラムから生のアナログ波形を表示し、シグナル・インテグリティ問題を明らかにします。

当社のロジック・アナライザには、業界で唯一のロジック・アナライザ／オシロスコープの統合パッケージであるiLink®ツールセットを提供しているものがあります。iLink®ツールセットは、当社TLAロジック・アナライザといくつかのTDSシリーズ・オシロスコープを統合します。

iLink®ツールセットにより、時間相関のとれたデジタル信号とアナログ信号が、ロジック・アナライザに表示されます。ロジック・アナライザはデジタル形式で信号を取込み、表示する一方、接続されたTDSシリーズ・オシロスコープは、同じ信号をアナログ形式で取込み、ロジック・アナライザのディスプレイに表示します。これらが同時に観測できることにより、例えばデジタル領域のタイミング問題がアナログ領域でどのようにグリッチとなって発生するか確認することができます。

iLink®ツールセットは、迅速な問題解決、トラブルシュートのために開発された統合パッケージです。

- iCapture™では、ロジック・アナライザの1本のプロービングにより、デジタル信号とアナログ信号を同時に取込みます。
- iView表示機能により、ロジック・アナライザとオシロスコープによる測定を、時間相関をとってロジック・アナライザに表示することができます。
- iVerify®解析機能により、オシロスコープで生成されたアイ・ダイアグラム・データを利用し、マルチ・チャネル・バス解析と検証を行います。

前述したように、基板製作時に発生する代表的な問題としてオープンや短絡があります。設計をスケジュール通りに進めるためには、基板のオープンまたは短絡の箇所を早急に突き止める必要があります。残念なことに、ローカルPCIバスをテストする段階で基板製作時での問題が発見されました。

このような問題を突き止める一般的なテクニックとしては、メモリまたはデバッグ・スクリプトの使用があります。この簡単なソフトウェア・ルーチンでは、あらかじめ定義されたデータ・パターンをアドレス・レンジまたは一つのアドレスに書き込み、読み出します。この一例はビット“1”を交互に書き込むテストです。つまり、0x55と0xAAのパターンをアドレスに交互に書き込むことで、すべてのシングル・データ・ビットを0と1で切り替えます。この書き込みを繰り返し実行し、各データ・ビットをオシロスコープでプロービングしてすべてのデータ・ビットがトグルし、1、0に固定していないか、またトライ・ステート状態になっていないかを検証します。

このテストは一見簡単そうに見えますが、実際には難しい点もあります。一つのデータ・ビットをモニタする手順を、以下に示します。

1. 回路図からデータ・ビットを見つけ、プロービングできるポイントを調べます。
2. 基板の配置図を確認し、プロービングしやすいポイントを見つけます。ICのピンにプロービングしなければなりませんか。ビア経由ですか。あらたにテスト・ポイントを設けなければなりませんか。この例では、PCI_AD0（PCIバスのアドレス／データ・バスのビット0）をU34の18ピンでプロービングします。
3. 目視または基板配置図からU34を見つけます。
4. U34から18ピンを探し出し、オシロスコープのプローブを接続します。これは簡単なことではありません。微細なピン・ピッチとBGAが問題を複雑にしています。

この3分間の手順をあと31回繰り返すと90分かかり、一つの32ビット・バスをチェックできます。もっと簡単で生産性の高い方法があるはずです。

デバッグ・プランでは、PCIバス専用のテスト・ポイントが設計されていました。設計では、図19に示すようなP6960型高実装密度ロジック・アナライザ・プローブのためのパッドが含まれています。P6960型では高価なオンボード・コネクタは不要なD-Maxコネクタレス・プロービング技術が採用されています。

図19：*D-Maxコネクタレス・プローブ*

このプローブは、選択されたアナログ信号を、TLAシリーズiCapture™マルチプレクサにより、ロジック・アナライザに接続されたオシロスコープに送るように設計されています。iCapture™マルチプレクサにより、図10のようにロジック・アナライザの1本のプロービングにより、デジタル信号とアナログ信号を同時に取込みます。

オシロスコープを使用して32本の異なった信号をプロービングする代わりに、P6960型プローブ1本を基板に接続し、マウス・クリックするだけで、32ビットが正しくトグルしていることを検証できます。手順は簡単です。

1. 34チャンネルD-Maxコネクタレス・プローブを、図20のように基板に接続します。
2. ロジック・アナライザ・モジュールのCH1アナログ出力を、オシロスコープの未使用チャンネルに接続します。
3. TLAアプリケーションのAnalog Feedsダイアログ・ボックスにおいて、測定する信号をアナログ出力から選択します。この例では、PCI_AD31：0をCH1出力に割り当てます。Analog Feed Cycling チェックボックスでCH1を選択し、32チャンネルの各信号が簡単にオシロスコープに送られます。
4. 右矢印ボタンをクリックし、次の信号をオシロスコープに送ります。信号がトグルしていることをオシロスコープで確認します。
5. 手順4を繰り返し、すべての信号をチェックします。

90分かかった32ビット・バスのチェックが、32回のマウス・クリック、わずか90秒の作業に軽減されました。この方法により、PCI_AD7とPCI_AD13が半田ブリッジにより短絡していることをすばやく確認できました。

図20：*基板に接続されたD-Maxコネクタレス・プローブ*

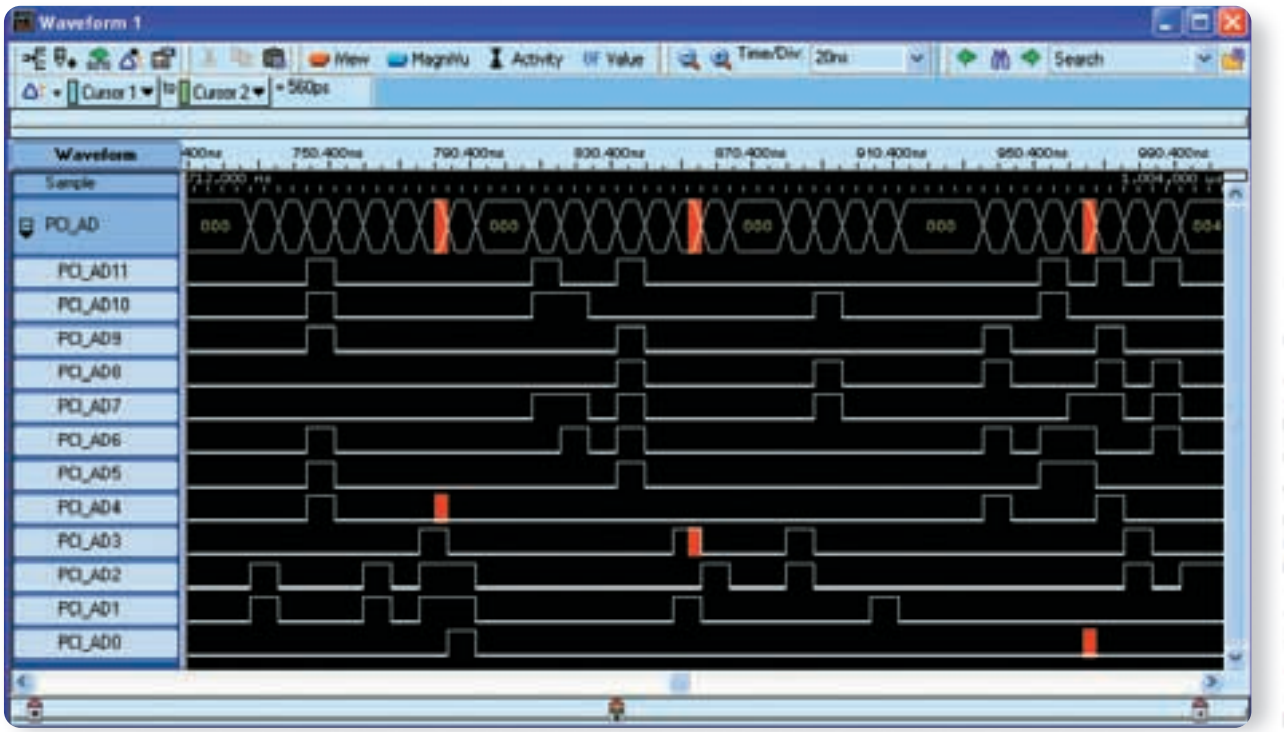


図21：*ロジック・アナライザによりグリッチでトリガし、各グリッチの位置にフラグを付けて表示した例*

他の機能ブロックのデバッグに進むときになって、別なデバッグ問題が発生しました。この例では、データは当初、シミュレーション結果どおりにローカル・バスからPCIバスに確実に転送されていました。しかし、いくつかのトランザクションの後、ローカルPCIバスでエラーが発生しました。PCI_ADバス（PCIアドレス／データ・バス）で0x0008となるべき値が0x0000となっており、一回だけではなく繰り返されています。完全に止まってしまったり、配線が間違っているのではなく、同じエラーが連続的に発生しているようです。不安定な問題により間欠的なイベントが発生し、正しいデータ・ビットと判断されてしまい、16進の値が別な値になっています。繰り返し性の問題であることから、試作基板のレイアウトまたは組み立てのエラーによって発生するグリッチによるものと考えられます。

PCI_ADバス・エラーの原因はどこにあるのでしょうか。

ロジック・アナライザのグリッチ・トリガと表示モードにより、エラーを検出することができます。このモードではグリッチを検出してトリガし、各グリッジの位置をタイミング表示でフラグを付けて表示します。ロジック・アナライザでは、グリッチは、サンプル・ポイント間で発生する2つ以上のトランジションと定義しています。TLAシリーズ・ロジック・アナライザによる測定結果の例を、図21に示します。

図21には、2種類の波形が表示されています。スクリーン上部には、ワード値に対応したPCI_ADバス波形として表示されています。バス波形は、多くの個々の信号をステートとしてわかりやすく表示しますので、トラブルシュートの時間を短縮することができます。バス波形の下には、個々の信号に展開されて表示され、各グリッチの位置にはフラグが付いて表示されます。

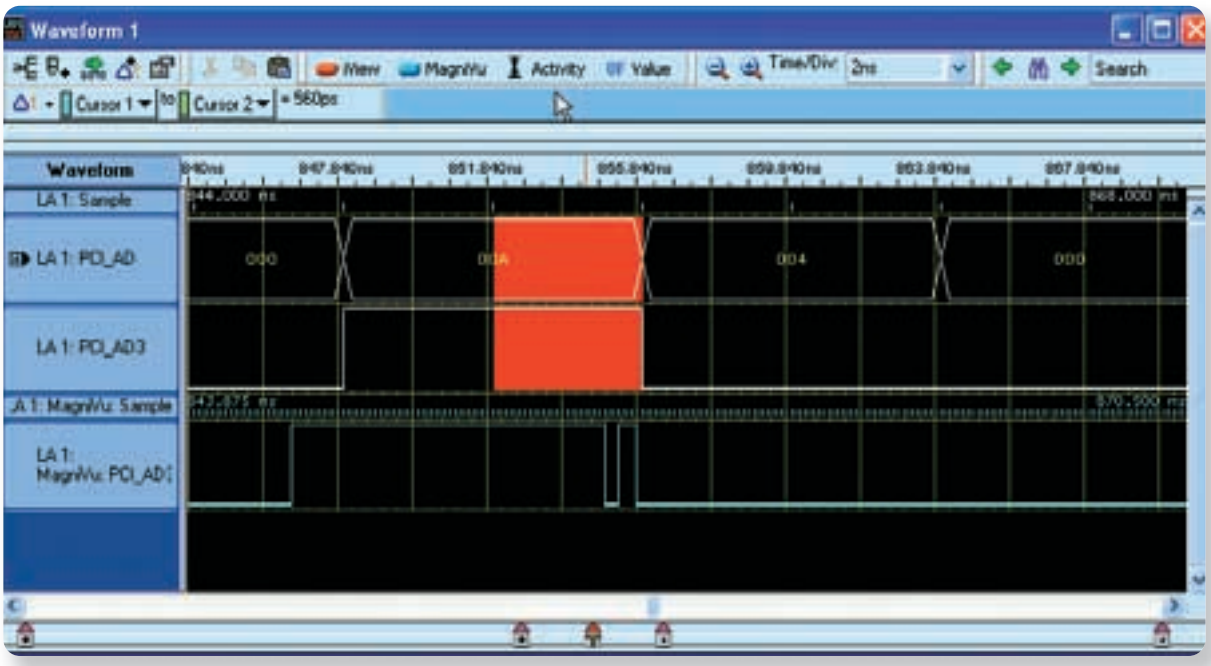


図22：*ロジック・アナライザのMagniVu®アクイジションにより、PCI_AD3信号のエラーが観測できます。*

図21では、ロジック・アナライザのサンプル・クロック間隔は4.00nsです。この例では、TLAシリーズ・ロジック・アナライザのMagniVu®アクイジションにより、トリガ・ポイント前後を125ps間隔で取込み、任意の信号を高分解能で別途表示します。この機能では、高解像度データは、メイン・タイミング・データと同時に、同じプローブで取込まれます。

図22では、MagniVu®アクイジションによる波形が追加表示されています。ここでは、125psのMagniVu®アクイジションのクロック信号と、PCI_AD3信号の詳細がバス形式で表示されています。信号は、サイクル後半にトランジションが表されています。正しくないバス値が生成されていることがわかっていますので、このトランジションがエラーの原因と思われます。しかし、どのような要因でただしくないトランジションが起きているのでしょうか。

デジタル信号の異常は、アナログ信号のシグナル・インテグリティ問題から発生することがあります。iLink®ツールセット（図18）では、デジタル信号に問題を引き起こすアナログ特性を明らかにすることができます。モジュラ・タイプのTLAシリーズによるiCapture™マルチプレクサでは、ロジック・アナライザ内部のアナログ・マルチプレクサを経由して、D-Maxコネクタレス・プローブから任意の4つの信号を、接続されたTDSシリーズ・オシロスコープに送ります。アナログとデジタルの両方の信号が送られますので、ダブル・プロービングの必要がなく、プロービングによる信号への負荷も軽減できます。iLink®ツールセットのもう一つの機能であるiViewでは、デジタル波形とアナログ波形を、時間相関をとってロジック・アナライザに表示することができます。

図23：*iView機能により、PCI_AD3に潜むデジタル・エラーのアナログ特性が観測できます。*

図23では、PCI_AD3のアナログ信号とデジタルによる表示を、TLAシリーズで並べて表示しています。これから、次のようなことがわかります。デジタル・グリッチが発生した瞬間、アナログ信号の振幅はロジック・スレッショルドの領域で減少します。明らかにスレッショルド電位を割り込み、一瞬、ローまたはロジック0レベルになります。次に、スレッショルド・レベルを十分に越えるだけの電圧になり、ロジック0にスイッチングする前の、ハイまたはロジック1の電圧レベルに戻っています。

このアナログ特性が、バスのグリッチと16進出力におけるエラーの原因となっています。このような不安定な動きは、すべての立下りエッジに起きているものではなく、多くのパルスは問題ありません。当然のことですが、有効となるエッジがどこで発生すべきか、すなわち、このバス・サイクルの波形の不安定な部分の前なのか、後なのかについて、設計モデルをレビューする必要があります。

経験豊富なエンジニアであれば、この歪んだ波形から原因を認識することができます。このような減衰したロジック・レベルは、通常、適切に終端されていない伝送ラインの反射によって発生します。この設計例では高速エッジを持つ信号の受信端に、終端抵抗がありませんでした。その結果、立上りおよび立下りエッジにおいて不規則な減衰が発生しました。

このように、ロジック・アナライザとオシロスコープを組合わせたトラブルシュートにおいて、iLink®ツールセットが提供する以下の4種類の表示、解析機能によって、ハイレベルな観測から個々の信号の詳細観測が可能になります。

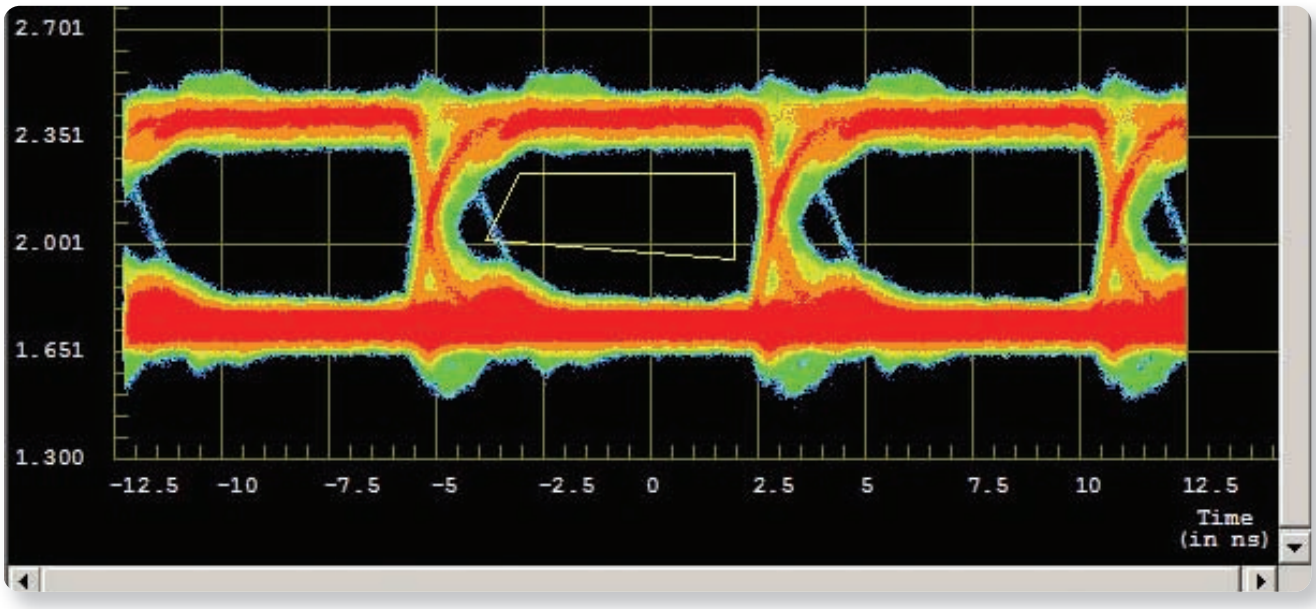


図24：*iVerify®解析機能では、アイ・ダイアグラム・フォーマット表示により、複数の信号とエッジ変化が同時に観測できます。*

- バス波形表示では、バス上で発生する問題を一目で確認できます。
- タイミング波形表示では、どの信号ラインが問題なのかを確認できます。
- MagniVu®タイミング波形では、エラー箇所を時間的に高解像度で特定します。
- アナログ波形は、iCapture™マルチプレクサによって接続されたDSOとiView機能によって表示され、信号のアナログ特性が観測できますので、原因を究明することができます。

3.3.2 シグナル・インテグリティ問題を迅速に検出する方法について

ロジック・アナライザに装備されているiLink®ツールセットには、iVerify®という、トラブルシュートに便利なもう一つのツールが用意されおり、ロジック・アナライザ上で複数のチャンネルのアイ・ダイアグラムを表示して解析できます。

アイ・ダイアグラムでは、クロックに同期したバス信号のデータ有効ウィンドウと一般的なシグナル・インテグリティを視覚的に観測することができます。最近の多くのバス、特にシリアル・バスにおいて、コンプライアンス・テストで必要とされるツールですが、任意の信号ラインもアイ・ダイアグラムとして観測することができます。

iVerify®解析により、立上り／立下りパルス両方のリーディング・エッジとトレーリング・エッジを含む複数のアイ・ダイアグラムを一度に観測して、トラブルシュートを迅速に実行できます。

iVerify®解析機能の例を、図24に示します。この例では、ADバスのすべての32チャンネルの信号が重ね書きされています。複数の信号を一度に観測できることの利点は明らかであり、32ビット・バス、64ビット・バスにおいても便利な機能です。D-Maxコネクタレス・プローブに接続されたグループの信号であればどの信号でも選択することができます。

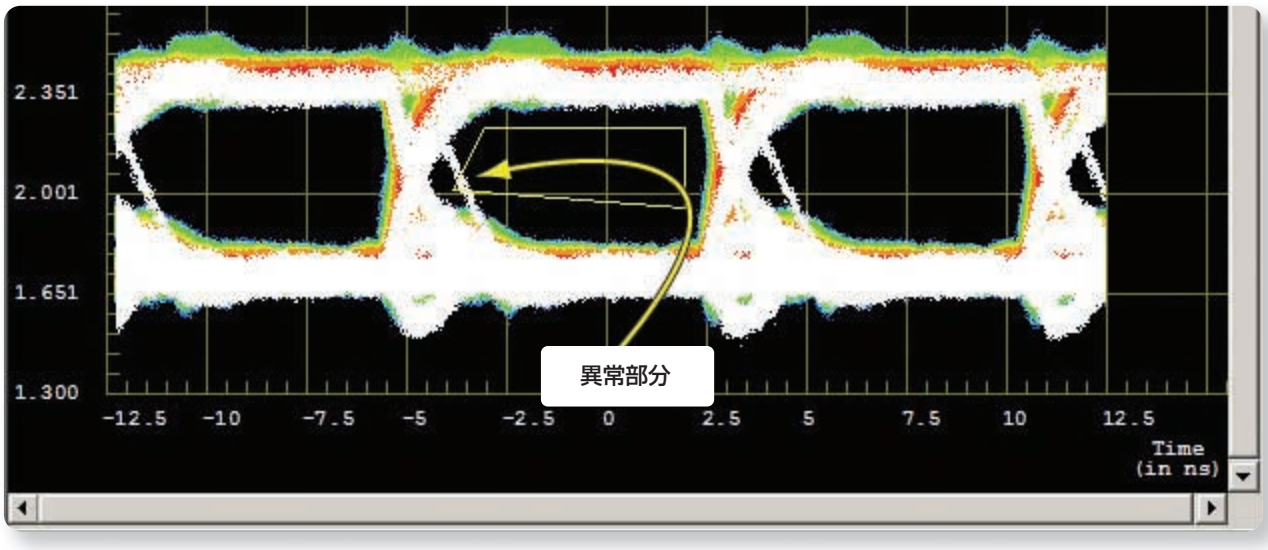


図25：*iVerify®解析ツールにより、異常波形をハイライトして、簡単に観測することができます。*

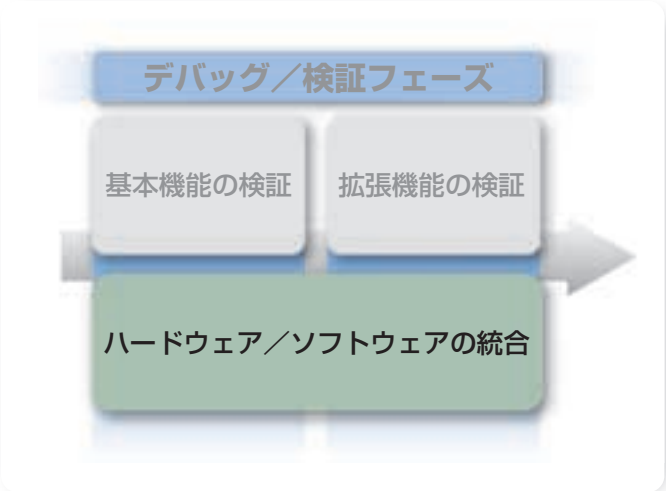
アイ・ダイアグラムでは、すべてのロジック・トランジションを一度に観測することができますので、信号の状態をすばやく評価することができます。遅い立上り時間、過渡時間、減衰レベルなどのアナログ問題が確認できます。エンジニアによっては、まずアイ・ダイアグラム全体を観測し、次に個々の異常に関して調べる人もいます。

図24のアイ・ダイアグラムでは、信号に異常が確認できます。薄い青色のラインは、間欠的に発生していることを意味します。しかし、信号の中で、最低一つは通常の範囲を外れていることがわかります。マスク機能により、どの信号が問題になっているのかが確認できます。設定したマスク領域外に信号が出た場合は、その信号はハイライトされ、イメージの最も前面に表示されます。図25の例では、異常信号が最前面に白い色でハイライト表示されています。

この例では、PCI_AD3信号のエッジに異常があることがわかりました。問題の原因はクロストークであり、回路基板の隣接した配線パターンからの信号によるものでした。

3.3.3 まとめ

多くの場合、機能検証フェーズではいくつかの問題は検出できません。これは、機能問題またはシグナル・インテグリティに関する問題です。ロジック・アナライザは、デジタル機能のテストで最初に使用する計測器です。しかし、デジタルに関する問題は、ここで取り上げた不適切な終端やクロストークなどによるエッジ不良を含むアナログ信号に起因することがあります。ロジック・アナライザとオシロスコープを組合せ、一つの画面で時間相関をとったデジタル信号とアナログ信号を評価することで、どちらの領域の問題も簡単に解決できます。



ハードウェア／ソフトウェアの統合

3.4 ハードウェア／ソフトウェアの統合

ハードウェア／ソフトウェアの統合は、実際には基本機能の検証フェーズから始まり、拡張機能の検証フェーズまで続きます。ハードウェア／ソフトウェアの統合に関する問題はデバッグが難しく、システムとしてハードウェアとソフトウェアの両方が調べられるツールが必要になります。ロジック・アナライザは、特定の信号とプロセッサの命令実行フローを関連付けて観測することができます。これにより、チームのハードウェア担当もソフトウェア担当も、なぜ特定のインストラクションがメモリ・エラーを起こすか、また、なぜソフトウェアが予期しないブランチを実行するかが理解できます。

3.4.1 マイクロプロセッサのブート・コードのデバッグ

組込みシステムのマイクロプロセッサをブートすることは難しい作業です。未知のハードウェアが、未知のソフトウェアと初めて組合わされます。シーケンスがクラッシュすることがあります。組込みシステムは、PCなどの非組込みシステムとは異なり、誤った動作によってターゲット・システムがクラッシュすることに対して通常は何の保護もしていません。強固なオペレーティング・システムは、通常、間違った動作を特定するための各種のメカニズムを備えています。しかし、組込システムはそうではありません。したがって、ブート・コードがクラッシュするような場合、システム全体がダウンし、問題をデバッグするための有効な情報も消失してしまいます。

この例の試作品では、マイクロプロセッサのブートでは問題は発生しませんでした。ときどきですが、システムが予期せずにクラッシュします。クラッシュの根本原因はわかりませんでしたので、トラブルのメカニズムにトリガすることはできません。クラッシュをデバッグする場合、トラブルの現象にトリガするか、あるいは起こるべき何かが起きていないことに対してトリガすることが有効です。この例では、ローカル・バス信号の一つの欠落にトリガしました。このストローブ信号が十分にトグルしないとき、マイクロプロセッサは期待通りに機能しませんでした。このため、監視用にウオッチドッグまたはハートビート・パルスをシステムに組込んでみました。ハートビートがパルスを刻んでいる間は、システムは正常に動作しました。ハートビート・パルスが止まると、エラーが発生することがわかりました。

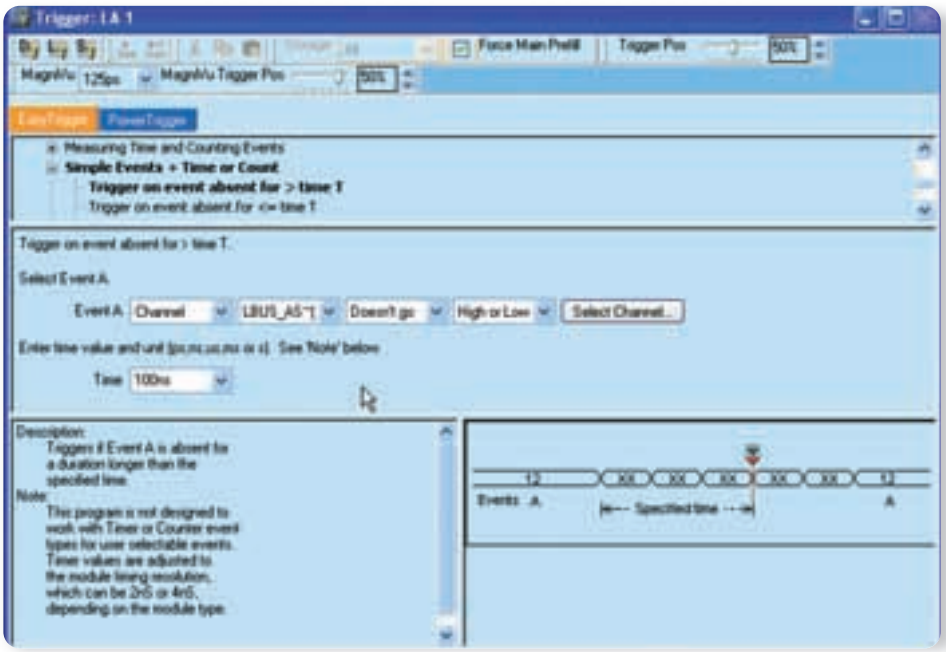


図26：*ロジック・アナライザのタイムアウト・トリガ例*

幸いなことに、何も起きないことに対してロジック・アナライザでトリガするよう設定し、システムの状態を詳細に表示させるのは容易なことでした。動作の欠落に対してトリガすることを、タイムアウト・トリガと呼びます。特定のラインまたはラインのグループに対して、予期される一定の時間ロジック信号が変化せず何も起きない場合にトリガするよう、ロジック・アナライザを設定します。図26では、TLAシリーズ・ロジック・アナライザのEasyTriggerメニューでタイムアウト・トリガを設定した例を示しています。

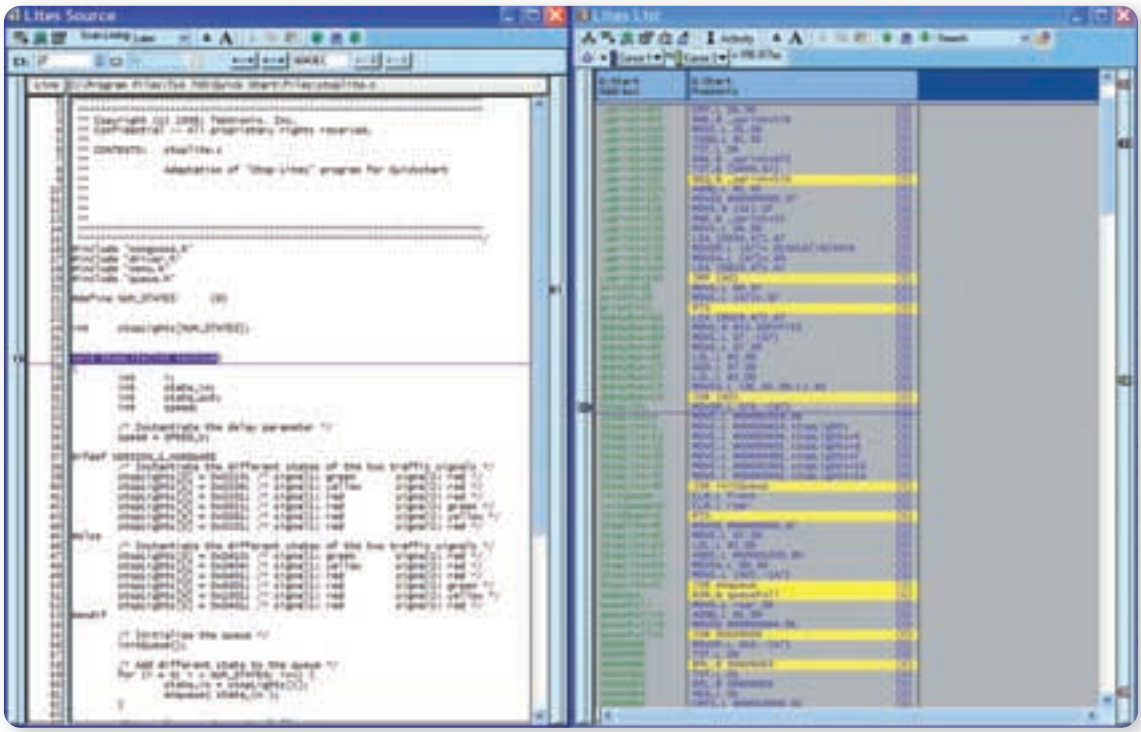


図27：*ロジック・アナライザのソースコード・ウィンドウ例*

3.4.1.1 ロジック・アナライザのソースコード・ウィンドウの使用

トラブルの現象が捉えられたならば、ロジック・アナライザのソースコード・ウィンドウを使い、図27に示すように、取込んだデータとソース・コードを関連付けて表示します。

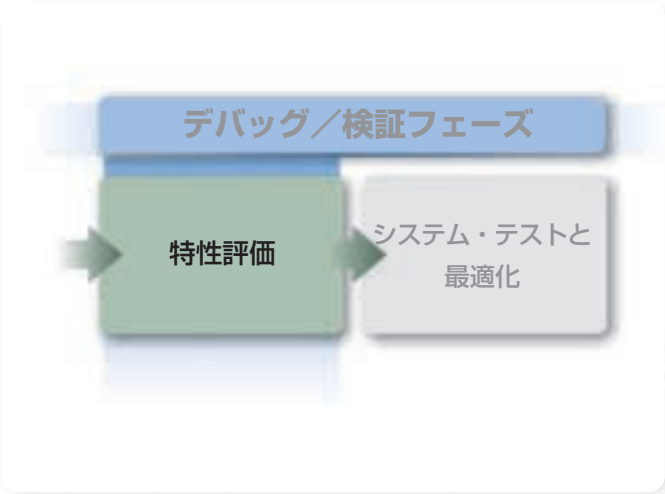
最初のインストラクションの取込みから始まり、予期しないインタラプトを処理するためにブランチするまではプロセッサが正しく機能することを確認しました。これにより、まだ初期化されていないメモリの場所にブランチするようコードすることになっていました。しばらくしてプロセッサは停止し、クラッシュしました。

ブートアップ時にインタラプトをマスクするよう、簡単なハードウェア変更を行ったところ、この問題は解決しました。さらに、ブート・シーケンスにおいて、できる限り早く、ソフトウェアでインタラプトをマスクするようアレンジし直しました。

3.4.2 まとめ

ハードウェアとソフトウェアの統合で発生する数多くの問題は、その原因を正確に突き止めるのが困難な場合があります。ハードウェアの問題なのか、それともソフトウェアの問題なのか。ハードウェアとソフトウェアの動作を同時に取込めるロジック・アナライザは、ハードウェア／ソフトウェアの統合問題の解決に最適なツールです。

すべての機能の検証が終わったら、設計は特性評価のフェーズに移ります。



特性評価

3.5 特性評価

いよいよ基板は、性能マージン、上限、耐性などの特性を評価することになります。これらの情報により、製造に値するだけの十分な信頼性があることを確認します。

多くの場合、これらのパラメータは、いろいろな温度、湿度、高度、振動などの条件化でテストされます。消費電力は検討され、多くの製品ではバッテリ寿命も重要となります。EMIに対するデバイスの耐性と同時に、EMI放射についても評価し、ドキュメント化する必要があります。

これまでの設計プロセスでも使用した、リアルタイム・オシロスコープ、広帯域プローブ、専用の測定／解析ソフトウェアは、最終製品の限界値を知るためにも使用されます。この段階では、負荷、EMIなどの外的要因による問題がない限り、シグナル・インテグリティは問題になりません。

特性評価には、詳細なストレス・テストも含まれます。このプロセスでは、デバイスには供給電圧の変動が加えられ、あまり理想的とは言えない環境下で正しく機能するかを評価します。また、故意に欠陥のある信号を入力する方法もあります。欠陥のある信号としては、限界に近い振幅、遅い立上り時間、オーバシュートなどの異常が挙げられます。当社AWGシリーズ任意波形ゼネレータ（AWG）は、これらの信号を生成するのに最適なツールです。

3.5.1 設計エンジニア向けの仕様とエンド・ユーザ向けの仕様

ほとんどの電子製品には、2つの仕様レベルがあります。

- エンド・ユーザ向けに保証された公開仕様：製品マニュアルやブローシャなどに記載されます。
- 製品マニュアルやブローシャに記載されない（非公開）規格仕様：設計プロセスをサポートするための仕様であり、記載（公開）された仕様に対するガードバンドが設けられています。ガードバンドとは、量産化される通常製品が記載（公開）された仕様に収まるよう設けられたマージンです。

両方の仕様は、開発プロジェクトを通したガイドラインとして機能するエンジニアリング仕様としてまとめられます。最終的な測定で、公開、非公開の両方の仕様を満たすことを確認します。関係するデバイスによって異なることはありますが、測定するパラメータとしては、セットアップ／ホールド時間などのタイミング耐性、パルス振幅と立上り時間、クロック周波数の安定度、ノイズ特性、ジッタ、インピーダンス範囲などがあります。すべての測定は、本書ですでに説明したように、計測器の測定能力範囲に十分に入っている必要があります。

3.5.2 セットアップ／ホールド時間テスト

製造におけるライン停止、フィールドでの信頼性問題における最も大きな原因の一つに、不適切なタイミング・マージンが挙げられます。セットアップ／ホールド時間は、最も重要な同期タイミング・パラメータです。これを確かめることで、設計が適切なセットアップ／ホールド・マージンを持っているか確認できます。しかし、設計においてセットアップ／ホールド時間のマージンを測定、ドキュメント化することは、非常に手間のかかる作業です。

オシロスコープを使用する場合、セットアップ／ホールド時間を測定するには2種類の方法があります。まず、クロックと一本のデータをプロービングします。オシロスコープは、セットアップ時間違反またはホールド時間違反にトリガ設定します。一度に一つの信号しかテストできないというのが欠点です。

オシロスコープによるもう一つの方法は、クロック信号と最大3本のデータ・ラインを取込み、タイミング違反を発見するアプリケーションを準備し、データを送った後処理します。このアプローチでは、わずかにテスト時間が短縮できますが、貴重な開発時間をアプリケーションのプログラミングに費やす必要があります。

ロジック・アナライザは、数多くの信号のセットアップ／ホールド時間のマージンを一度に検証するのに理想的なツールです。TLAシリーズ・ロジック・アナライザは、すべての信号に対して一度に、ユーザ定義したセットアップ／ホールド時間違反でトリガし、表示することで、セットアップ／ホールド時間違反を自動的に検出します。内部PCIバスを例に見てみましょう。

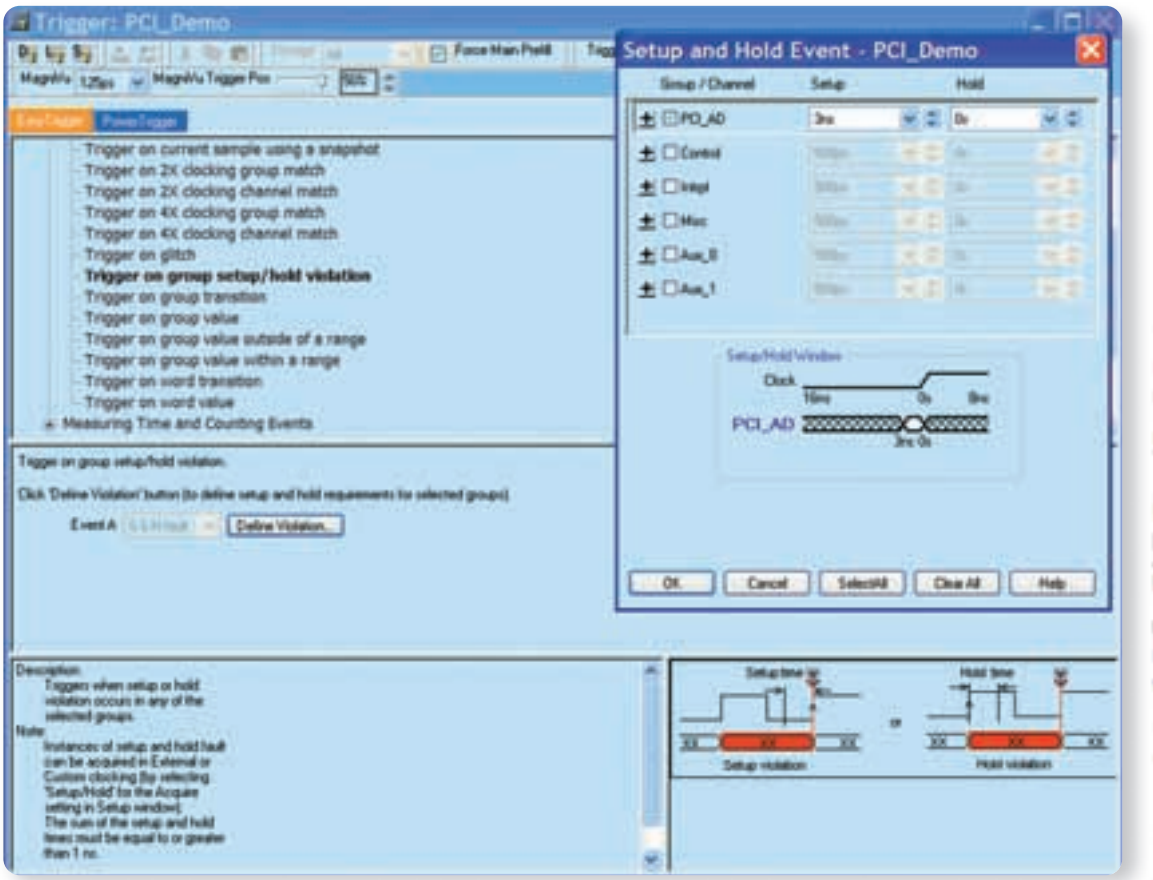


図28：*セットアップ／ホールド時間違反トリガ*

PCI_ADラインに必要なセットアップ時間は3ns、ホールド時間は0sです。ロジック・アナライザの高実装密度プローブにより、PCI_AD信号には簡単に接続できます。ロジック・アナライザのトリガ設定と、検出されハイライトされたセットアップ／ホールド時間違反を、図28に示します。夕方、ロジック・アナライザのRun/Stopボタンを押して帰宅しました。夜間中、ロジック・アナライザは全てのバス・ラインで違反がないかPCI_ADバスをモニタします。違反が発生すると、ロジック・アナライザはトリガし、問題を捕捉します。翌朝オフィスに来てロジック・アナライザを見ると、テスト・ステータスが表示されています。ロジック・アナライザがトリガしていたならば、違反が発生していたことを意味します。トリガがない場合は、セットアップ／ホールド時間違反はなかったことを意味します。従来のオシロスコープよりも、より簡単なアプローチです。

3.5.3 まとめ

特性評価のフェーズで必要となる作業の多くは、多くの繰り返し測定が必要になります。例えば、セットアップとホールド時間のマージンは、いくつかの異なる動作温度で検証する必要があります。優れたテスト・ツールを使うことで、作業が大幅に簡単になり、測定に必要な時間を短縮できます。

システム・テストと最適化

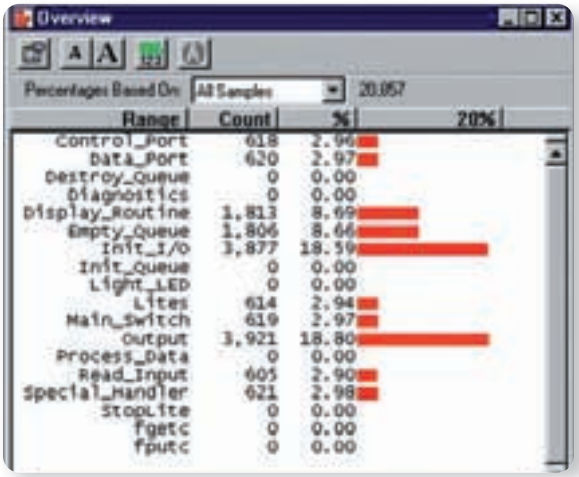


図29：レンジ・オーバビュー

3.6 システム・テストと最適化

最新の組込みソフトウェア・アプリケーションは、ますます大きく、複雑になり、全体の流れ、ソフトウェアの実行時間を把握するのは難しくなっています。組込みソフトウェアの開発エンジニアは、コードが正しく機能するようにしましたが、性能面では満足出来ないケースが増えてきました。性能面で満足させるためには、プロジェクトの最終段階において性能の最適化をはかる必要があります。よく引用される経験則では、「20%のコードの実行に80%の時間を使う」と言われていますが、「20%のコード」とはどの部分でしょうか。このようなチューニングは、製品のスループット要求を満たすために必要です。

ロジック・アナライザは、数百ものソフトウェア・モジュールのどれがプロセッサの実行時間の大部分を消費しているかを示す概要を提供することができます。PA（Performance Analysis、パフォーマンス解析）では、ソフトウェアのどの部分がどの程度時間を使っているかを把握することができます。この情報により、最適化が必要な場合、性能向上の面で最も効果の高いルーチンをすばやく確認することができます。

図29に示すレンジ・オーバビューでは、取込んだデータをユーザ定義のレンジに切り分け、ヒストグラム・フォーマットで各レンジのヒット数としてグルーピングし表示します。これは、どのソフトウェア・モジュールが最も実行時間を使っているかを知るのに適しています。

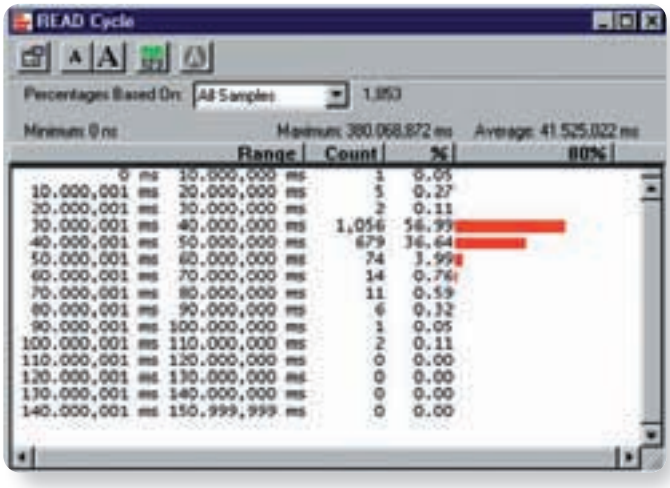


図30：*シングル・イベント*

割り込み処理／例外ハンドラなど、処理時間が重要となるルーチンは、設計目標を満たしているか検証しておく必要があります。TLAシリーズ・ロジック・アナライザのシングル・イベントは、ロジック・アナライザのタイマとカウンタを使用し、一つのルーチンの実行時間／カウントの範囲を表示する、パフォーマンス解析測定モードです。図30はその結果ですが、一つのイベントが複数回実行される場合の最小、最大、平均時間を示しています。

ソフトウェアのパフォーマンス解析の他に、どのPCIエージェントが帯域を消費しているか確認する必要があります。レンジ・オーバビュー機能があれば、簡単に実行できます。必要なことは、各エージェントのI/Oとメモリ空間に合うようにレンジを定義するだけです。

3.6.1 まとめ

システム・テストのフェーズでは、設計がすべての動作パラメータを満足しているかを検証します。応答時間は十分に早いですか。すべてのデータを効果的に管理するだけの十分なシステム帯域がありますか。TLAシリーズ・ロジック・アナライザのパフォーマンス解析ツールは、システムのボトルネックを検出し、ハードウェアとソフトウェアの両方を最適化する手助けになります。

4 まとめと結論

この組込みシステムの開発プロセスにおいて、重要な測定作業が簡単になるような領域に焦点を当てて説明しました。

4.1 設計フェーズ

設計フェーズにおける決定事項は、機能検証、機能／シグナル・インテグリティ問題の検出と解決、システム・チューニングに大きく影響します。優れた生産性は、設計フェーズにおいて効果的な開発プロセスを守り、起こりそうな問題を考慮し、デバッグを考慮した設計を行うことによって実現されることを説明しました。

4.2 デバッグ／検証フェーズ

初回の電源投入は、デバッグ／検証プロセスの6つのプロセスの中では最も理解しやすいプロセスです。実質的な最初のステップですが、基本的な機能検証であり、システム実行のコアとなります。

問題をデバッグする際に最初に使用するのが、リアルタイムDSOまたはDPOです。これまで見てきたように、これらのリアルタイム計測器を使用することで、電源ノイズから高速信号まで簡単にプロービングでき、確実に波形を取込むことが可能になります。ロジック・アナライザは、4つ以上の信号の関係やソフトウェアの動作を観測するのに最適です。

機能ブロックの検証、デバッグが進むにつれ、基板製造レベルでの問題やシグナル・インテグリティに関する問題が発生する確率が増加していきます。TLAシリーズ・ロジック・アナライザのiCapture™マルチプレクサ機能により、基板製造レベルでの問題を検出し、グリッチにトリガして識別表示することができ、シグナル・インテグリティ問題の検出がすみやかに実行できることを確認しました。

複雑なハードウェア／ソフトウェア統合問題の解決には、非常に時間がかかります。それはハードウェアの問題なのか、ソフトウェアの問題なのか、これは常に問題となることです。ロジック・アナライザにより、ハードウェアのイベントとソフトウェアの実行を関連付けて観測することで、ハードウェア／ソフトウェアの統合問題を解決することを確認しました。

また、TLAシリーズ・ロジック・アナライザのセットアップ／ホールド時間違反トリガやパフォーマンス解析などの優れた機能により、特性評価フェーズのみならず、システム・テストや最適化フェーズでも迅速な検証が可能なことを確認しました。

4.3 結論

この入門書では、開発サイクルの初期の段階からデバッグや検証を考慮する必要性について、例を挙げて説明しました。これが適切にプラニングされていない場合、グリッチ、異常波形、進行を妨げるような不具合などを発見することができなくなり、次のステップに進むことができなくなります。高性能のオシロスコープ、ロジック・アナライザ、信号ゼネレータと自動取込／解析ツールを組合せることで、現在のデバッグに関する難しい問題もすばやく解決することができます。

Tektronix お問い合わせ先:

東南アジア諸国／オーストラリア (65) 6356 3900
オーストリア +41 52 675 3777
バルカン半島／イスラエル／アフリカ南部諸国およびISE諸国 +41 52 675 3777
ベルギー 07 81 60166
ブラジルおよび南米 55 (11) 3741-8360
カナダ 1 (800) 661-5625
中東ヨーロッパ／ウクライナおよびバルト海諸国 +41 52 675 3777
中央ヨーロッパおよびギリシャ +41 52 675 3777
デンマーク +45 80 88 1401
フィンランド +41 52 675 3777
フランス +33 (0) 1 69 86 81 81
ドイツ +49 (221) 94 77 400
香港 (852) 2585-6688
インド (91) 80-22275577
イタリア +39 (02) 25086 1
日本 81 (3) 6714-3010
ルクセンブルグ +44 (0) 1344 392400
メキシコ、中米およびカリブ海諸国 52 (55) 5424700
中東アジア／北アフリカ +41 52 675 3777
オランダ 090 02 021797
ノルウェー 800 16098
中華人民共和国 86 (10) 6235 1230
ポーランド +41 52 675 3777
ポルトガル 80 08 12370
大韓民国 82 (2) 528-5299
ロシアおよびCIS諸国 +7 (495) 7484900
南アフリカ +27 11 254 8360
スペイン (+34) 901 988 054
スウェーデン 020 08 80371
スイス +41 52 675 3777
台湾 886 (2) 2722-9622
イギリスおよびアイルランド +44 (0) 1344 392400
アメリカ 1 (800) 426-2200
その他の地域からのお問い合わせ 1 (503) 627-7111

Updated 12 May 2006

詳細について

当社は、最先端テクノロジに携わるエンジニアのために、資料を用意しています。当社ホームページ(**www.tektronix.co.jp** または**www.tektronix.com**)をご参照ください。

05/06 FLG/WOW 52Z-18668-1

Tektronix
Enabling Innovation

日本テクトロニクス株式会社
東京都港区港南2-15-2 品川インターシティ B棟6階 〒108-6106
製品についてのご質問・ご相談は、お客様コールセンターまでお問い合わせください。

TEL 03-6714-3010 FAX 0120-046-011
電話受付時間／9:00～12:00・13:00～18:00 月曜～金曜（祝日は除く）

当社ホームページをご覧ください。www.tektronix.co.jp
お客様コールセンター ccc.jp@tektronix.com