

Xcell journal

ISSUE 54 2005

THE AUTHORITATIVE JOURNAL FOR PROGRAMABLE LOGIC USERS

デザインにおける挑戦

危険をどのように回避し
成功を勝ち取るか

PERFORMANCE

設計・分析ツール
PlanAhead の利用効果

COST

ザイリンクス CPLD を使った
QWERTY キーボードの実装

POWER

パフォーマンスか消費電力か:
その最適解を得る

CONNECTIVITY

シグナル インテグリティの
実現

DEBUGGING

ミックスド シグナル
オシロスコープを用いた
FPGA の高速デバッグ

 XILINX®

VIEWPOINT

ハードウェアがソフトウェアに出会うとき..... 1

SYSTEM PERFORMANCE

より高速・よりフレキシブルなエンベデッド システム..... 5

高性能 DSP デジタル信号処理システムを実現する
制御回路の設計 9

Virtex-4 による記録のパフォーマンスの達成 14

設計・分析ツール PlanAhead の利用効果 18

最適化された FIFO ソリューションの設計..... 22

TOTAL COST

ザイリンクス CPLD を使った QWERTY キーパッドの実装... 25

ザイリンクス & マイクロソフト社が推進する高性能
テレマティクス システム 29

POWER MANAGEMENT

消費電力を低減させる設計テクニック..... 34

パフォーマンスが消費電力か：その最適解を得る..... 37

ハンドヘルド アプリケーションへの CPLD の適用..... 43

CONNECTIVITY

FPGA デザインにおけるメモリ問題の早期発見法 47

Spartan-3E FPGA を Intel StrataFlash
メモリに接続する..... 51

Virtex-4 でソース同期を実現するツールキットの機能解説... 56

システム パケット インターフェイス間のブリッジング... 60

シグナル インテグリティの実現..... 64

DEBUGGING

ハードウェアとソフトウェアの協調検証..... 69

Synplify Pro によるタイミング クロージャ..... 74

ミックスド シグナル オシロスコープを用いた
FPGA の高速デバッグ 78

GENERAL

鉛フリー & グリーンプロダクトへの挑戦 82

知識はパフォーマンスを創出する：ザイリンクス教育サービス 84

SUCCESS STORY from JAPAN

低インピーダンスと大容量を実現する注目のデバイス
「ブロードライザ」..... 90

ザイリンクス イベント カレンダー 93

LETTER FROM EDITOR

人生はチャレンジ：チャンスと見るか、災難と見るか... 94

広告索引

株式会社ミッシュインターナショナル 13

株式会社アイダックス 68

株式会社コンピューテックス 73

株式会社ソリトンシステムズ 77

有限会社ヒューマンデータ 81

Xcell Journalのご送付先住所等の変更は：
<http://www.xilinx.co.jp/xcell/henko/>
Xcell Journalの新規定期購読のお申込みは：
<http://www.xilinx.co.jp/xcell/toroku/>

SYSTEM PERFORMANCE

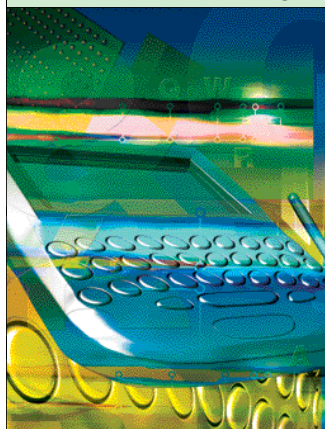


設計・分析ツール PlanAhead の 利用効果

設計コストの削減とパフォーマンス
向上に貢献してきた PlanAhead
ソフトウェア

18

TOTAL COST

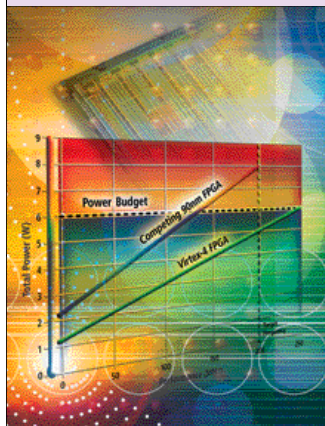


ザイリンクス CPLD を使った QWERTY キーパッドの実装

典型的な携帯端末用
DTMF キーパッドを
QWERTY キーパッドに変える

25

POWER MANAGEMENT

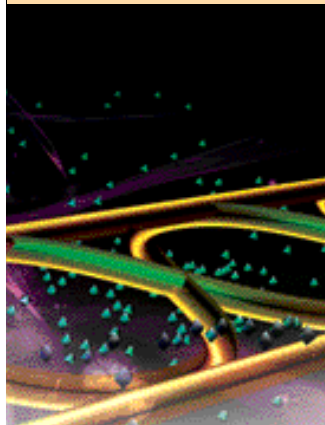


パフォーマンスが 消費電力か： その最適解を得る

90 nm 特有の変曲点を初めて
克服したザイリンクスの
取り組みと成果

37

SYSTEM PERFORMANCE



シグナル インテグリティ の実現

ノイズやジッタの問題を
いかに克服するか

64



When Hardware Met Software

ハードウェアがソフトウェアに出会うとき

ハードウェア/ソフトウェア統合の広大な溝に橋をかける

Anthony Townsend
Xilinx Design Services Engineering Manager
Xilinx, Inc.
tony.townsend@xilinx.com



重要なのは、共通点でなく相違点です。
今まで、デザイン関連企業全体としては、
ハードウェアのシミュレータとエミュレータを開発することに、
より多くの時間とリソースを費やしてきました。

ここで述べることは、誰にでも経験があるか、あるいは近いうちに経験することになるものでしょう。

インテグレーションの苦悩

まるで永遠に続くかに思えた仕様変更とあまりにも短い開発期間によって、ほとんど人間らしい生活をしてこなかった 2 か月間が過ぎ、デザインすべてのブロックのシミュレーションとシステムレベルのシミュレーションがようやく終了しました。本来なら喜ぶべきところでしょうが、そんな気分にはなれません。というのも、どんなプロジェクトであれ、最も困難で、ストレスの溜まる苦難の段階がここから始まるからです。その段階とは、ハードウェアとソフトウェアのインテグレーションです。

ハードウェアとソフトウェアのインテグレーションは、どこでも頭痛の種です。他社の知り合いに尋ねてみるといいでしょう。口うるさいマネージャがどこにでもいるように、ハードウェアとソフトウェアのインテグレーションをめぐる苦労はどこでも存在します。書店へ行って専門書のコーナーを覗いてみてください。高速化やリアルタイム化、高性能、低電力、テストを目的としたデザインの本来ならいくらでも見つかりますが、ハードウェアとソフトウェアのインテグレーションに関する本は皆無でしょう。

この種のインテグレーションに効く特効薬は誰も発見しておらず、サイリクスさえまだ見つけていません。本稿では、インテグレーションの苦悩を完全に取り払うことはできないまでも、痛みを和らげるいくつかの方法をご紹介します。

ハードウェア開発チームの言い分

図 1 のデザインフローは、ハードウェア設計における構想から完成までの一般的な流れを示しています。フィードバックパスがあるものの、このフローの大部分はスムーズに進めることができ、克服できない問題に遭遇することはまずないでしょう。デザインのコーディングが終了しシミュレーションをパスするころには、既にボードが出来上がっているか、完成間近になっているはずで

す。ボード上で最初のスモークテストを実施したら、いよいよデザインの実装に着手します。まず、ホストマシンがボードと通信できることを確かめるため、基本的なアクセステストを行います。次に、メモリへのインターフェイス、チップ間のインターフェイスと動作、ホストプロセッサへのインターフェイスをテストするため、基本的なソフトウェアテストコードを書きま

す。また、メモリマップ、割り込み、ステータス、制御レジスタをテストするコードを書いて実行し、タイミングを検証します。後は、テスト済みのボードをソフトウェア開発チームに渡すだけです。

ソフトウェア開発チームがボードを動かします。ところが、ハードウェアが動きません。まったく動かないのですから、せっかくテストコードを書いてデザインを検証しても無駄だったわけです。データは破損し、割り込みが正常に機能してもクリアしません。システムに電源を入れてすぐにクラッシュしなかったのが幸いだったほどのです。

ハードウェア開発者はソフトウェア開発者に、いったい何をやっているのかと問い詰めるでしょう。テストコードは普通に走るし、ハードウェアに問題がないことは確認済みだと説明します。ハードウェア開発者は、ソフトウェア開発者はこの 6 か月間何をしていたのか、もしかしたら基本的なソフトウェアさえまともに機能していないのではないかと、不安に思うことでしょう。

ソフトウェア開発チームの言い分

図 1 のデザインフローはまた、ソフトウェア設計における構想から完成までの一般的な流れも示しています。フィードバックパスがあるものの、このフローの大部分はスムーズに進み、克服できない問題に遭遇することはほとんどないでしょう。プロジェクトの立ち上げ当初はハードウェアプラットフォームがなかったため、ソフトウェア開発者は、コードを開発しテストするためのホストエミュレーション環境を構築しました。

最初に、OS の基本的なテストから始め、関数呼び出し、割り込み、GUI をテストします。パフォーマンスを考えてコードを最適化し、演算パスを検証します。また、アルゴリズムをチェックして、バグが含まれていることの多いコーナーケースを検証します。ボードを入手するまでにソフトウェアが完成するよう、できる限りのことを行いますが、結局「ハードウェアが届くのは予定より 6 週間遅れる」と言われてしまいます。それでもプロジェクトの締め切りは延期されません。そして 6 週間後、ようやくハードウェアが届きます。

ボードに同梱されてこなかった電源とケーブルを探して、あちらこちらと連絡を取ります。やっとそろったところでボードに電源を入れます。ところが、システムはまるで死んでいるかのように動きません。ここから、デバッグしながらシステムの隅々まで見て問題点を突き止めるという大変な重労働が始まります。データはリトルエンディアン (Little Endian) のフ

フォーマットで送ると事前に合意していたにもかかわらず、ハードウェアがビッグ エンディアン (Big Endian) になっていることが発覚します。

ソフトウェア開発者はハードウェア開発者に、いったい何をやっているのかと問い詰めます。ハードウェア開発者は間髪を入れず、「テスト コードはきちんと走ったし、ハードウェアに問題はない」と反論します。しかしながら、ソフトウェア開発者は、ハードウェア開発者はこの 6 カ月間何をしていたのか、もしかしたら基本的なハードウェアさえまともに機能してないのではないかと、不安に思うことでしょう。

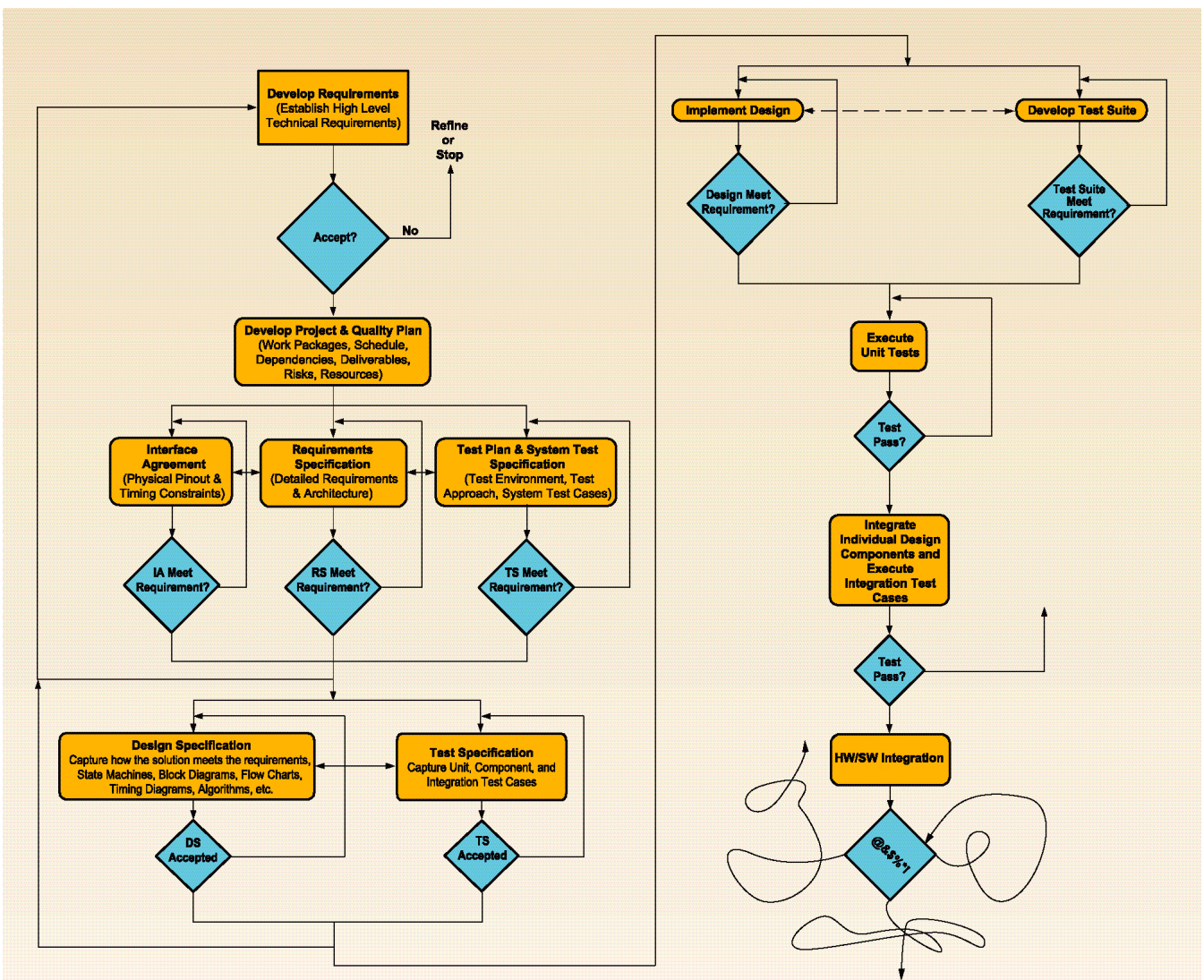
共通点ではなく相違点

以上は、やや極端な話ですが、過去または現在の開発プロジェクトにおいて読者のみなさんが実際に経験したことではないでしょうか。それにもかかわらず、このトピックについて書かれた本があまり見当たらないのは、現状にかなり矛盾があるからです。

図 1 がハードウェアとソフトウェアの両方に当てはまるとしたら、デザイン フローは非常に似ており、入れ替えてもかまわないくらいです。このことはインテグレーションの助けにならないでしょうか？ また、これを踏まえて、ハードウェアとソフトウェアのやり取りを記述する「ハードウェア/ソフトウェア インターフェイス仕様書」(呼び方は会社によって異なるでしょうが)を用意するのはどうでしょうか？ これは、インテグレーションの際に起きる問題を回避することに役立たないでしょうか？

ここで重要なのは、共通点でなく相違点です。今まで、デザイン関連企業全体としては、ハードウェアのシミュレータとエミュレータを開発することに、より多くの時間とリソースを費やしてきました。これは、米軍が VHDL (VHSIC Hardware Description Language) を採用したことをきっかけに本格的に始まり、その後 Verilog、最近では C-to-gates テクノロジーへ広がっています。そしてシミュレーション技術は、信頼性、精度とも非常に高いレベルに到達しています。今やほとんどのベンダが HDL モデルを提供しており、このモデルを使えば個々のデバイスでなくハードウェア システム全体をシミュレ

図 1 FPGAベースのシステム



ートできます。半面、設計エンジニアたちは、システムの他の部分を往々にして無視してきたといえるのではないのでしょうか。

ターゲットとするプラットフォームなしでソフトウェアをテストする場合、設計エンジニアはホストにそのシステムのモデルを構築する必要があります。ソフトウェア開発者はこのモデルを使ってシステム ソフトウェアを開発し、テストします。多くの場合、このモデルはデザインされるソフトウェアよりも複雑ですが、システムのあらゆる面を正確にモデル化できるわけではありません。まして、個々のコンポーネントにいたってはなおさらです。

ハードウェア設計者の場合はハードウェア モデルを利用できますが、ソフトウェア設計者向けに同様のモデルを提供している企業はめったにありません。協調シミュレーションは、2、3の単純な命令をシミュレートするだけでも時間がかかり、ルーチンやオペレーション全体をシミュレートするのは不可能に近いので、ソフトウェア、ハードウェア双方の設計者にとってほとんど利用価値がないのが実状です。

デザイン フローは基本的に同じように見えるかもしれませんが、これを実際に実行する段になるとまったく異なります。インテグレーションの段階で起こる問題の大部分は、こうした違いと、その違いを理解していないことが原因です。ハードウェア開発チームは、どんなに単純な機能であっても、ソフトウェア開発チームが確実に検証を行うには現物のハードウェアが不可欠だということを正しく理解しているとはいえないのです。

一方、ハードウェア開発チームは技術を過信しすぎる面があります。テスト ベンチを書く人とデザインする人は別々にすべきだというのは、多くのハードウェア設計者が口にするところです。ところが、ハードウェア開発者はボードをデバッグするための独自のテスト コードを書き、その際、ロジックをデザインするときと同じ誤りを犯しがちです。そして、本当は検証できていないのににもかかわらず、検証できたと思い込んでしまうことがあります。

技術ではなくコミュニケーション

インテグレーションの苦痛をできる限り抑えるには、技術ではなく、コミュニケーションと相互理解、相手に対する思いやりを重視すべきです。相手チームの作業内容や、遵守事項、使用しているプロセスとテクニックを理解することで、インテグレーションに向けてよりよい準備を整えておくことができます。コミュニケーションを図れば、インテグレーションの時間と労力を最小限に抑えられます。それには、相互の努力が最初は必要になります。

ハードウェア開発チームがボードを検証するのに使うテストコードを、あらかじめソフトウェア開発チームに書いてもらうのも 1 つのアイデアです。通常、この工程はスケジュールに組み込まれませんが、いくつかの点で理にかなっています。まず、テスト コードをソフトウェア開発チームに書かせることで、仕様がそれぞれのチームにより異なって解釈される危険を最小限に抑えられます。問題を初期段階で解決できるわけです。第 2 に、ビッグ エンディアンとリトル エンディアンなど、後々スケジュール

ルに大幅な遅延を招きかねないたくさんの細かな問題点を見つけ、影響をできるだけ小さくできます。これに加えて、ソフトウェア テスト プランを作り、ハードウェア開発チームが何のソフトウェアをどういう順序でテストするのかわかるようにすれば、インテグレーションの苦労はかなり緩和されるでしょう。

2 番目のアイデアは、ハードウェア開発チームが早い段階から頻繁にデザインをドキュメント化することです。ソフトウェア開発チームはターゲット ハードウェアが届くまで正規のテストに着手できないため、ハードウェア開発チームから提供されたドキュメントに頼るしかありません。ソフトウェア開発チームが完全なドキュメントを受け取り、そのドキュメントが常に最新版であることが重要です。また、ハードウェア開発チームはデザインを変更する前に、ソフトウェア開発チームに相談すべきでしょう。ハードウェア開発チームから見れば些細な変更であっても、特にプロジェクトの後半になれば、ソフトウェアを大々的に変更しなければならなくなることがあるのです。

最後に、ソフトウェア開発チームに安定したターゲット プラットフォームを提供することです。これは、完全でバグのないプラットフォームという意味ではなく、首尾一貫している、よく知られたプラットフォームという意味です。そしてハードウェア開発チームが、ボード/ロジックに関して明らかになっている問題点のすべてをソフトウェア開発チームに伝達することが重要となります。これにより、ソフトウェア開発チームが、既に明らかになっている問題点を追求する無駄な時間を費やすことを防げます。

お互いの幸せのために

仕様は変わるものですし、またスケジュールは変更されることがあります。上司から厳しい注文を突き付けられることもあるでしょう。いかなるプロジェクトであれ、技術面、運営面で多くの困難が伴いますが、ハードウェアとソフトウェアのインテグレーションほど、プロジェクトを危機にさらしたり、デザイン エンジニアに多大なストレスを与えたりするプロセスはありません。したがって、最初からそうした事態を想定したうえでプラン、デザインすることが大切なのです。

ハードウェアとソフトウェアのスムーズなインテグレーションには、特効薬などありません。ですが、なにかしらプランを立て、相手チームが何をしているのか、何をしようとしているのか、理解することはできるはずです。理解したうえで、ハードウェアあるいはソフトウェア開発チームと実際に話すことで、インテグレーションはかなり楽になります。最初の小さな犠牲が、最後の大きな成果を生むのです。

ソフトウェア開発チームがハードウェア開発チームのためにテスト コードを書くというのは予想外だったかもしれませんが、両チームが問題点を早期に検出し、修正できる方法はあるのです。後は、無理のないインテグレーションやテスト プラン、デザイン上の想定外の特徴を含む詳細にドキュメント化されたハードウェア、そして安定したハードウェア プラットフォームがあれば、インテグレーションの苦労はかなり緩和されるはずです。



SYSTEM PERFORMANCE

Faster and More Flexible Embedded Systems

より高速・よりフレキシブルな エンベデッド システム

プログラマブル プラットフォーム FPGA と
インテリジェント ツールが高性能処理ソリューションを創出する

Anthony Townsend
Product Marketing Manager,
Xilinx Embedded Solutions Marketing
Xilinx, Inc.
jay.gould@xilinx.com

 Platform Studio™

リアルタイム コンピューティングの仕様を決める際は、新しいシステムが過去のデザインより高速かつ柔軟であることが条件です。変化し続ける業界標準に対応するには、将来のエンベデッド システム デザインに新たなレベルのカスタマイズ性を実現する必要があり、また高性能というニーズに対しては従来のプロセッシング デザインに新たな課題が生じます。

過度にカスタマイズしたデザインによって制約を受けるのは避けたいですし、パフォーマンスを改善する目的でシステム クロックを高速にトグルし続けるのも不可能です。より高速かつ柔軟なエンベデッド プロセッシング システムを開発するよりよい方法が他にあるはずで

す。プラットフォーム FPGA は、多数の先進デザインをサポートするプログラマブル SoC であり、オンボード メモリ、DSP デジタル信号処理能力、エンベデッド プロセッシング、ハードウェア アクセラレーションを実現したコプロセッシングを組み込んでいます。この最新デバイスは再プログラミングとフィールド アップグレードが可能で、製品のライフサイクルを通じて、フィールドに展開後でも、バグ修正、機能の拡張、パフォーマンスの最適化、そして新たな業界標準のサポートが可能になります。

プログラマブル SoC デバイスにはこれらパワフルな機能が装備されていることから、あとは適切なツールを使ってこの内蔵されたパフォーマンスを存分に活用するだけです。

インテリジェント ツール

最近の業界調査によると、デザイン エンジニアは最終製品の開発にあたり、実際のデバイスやオペレーティング システムより、むしろインテリジェント ツールを重視する傾向があるよう

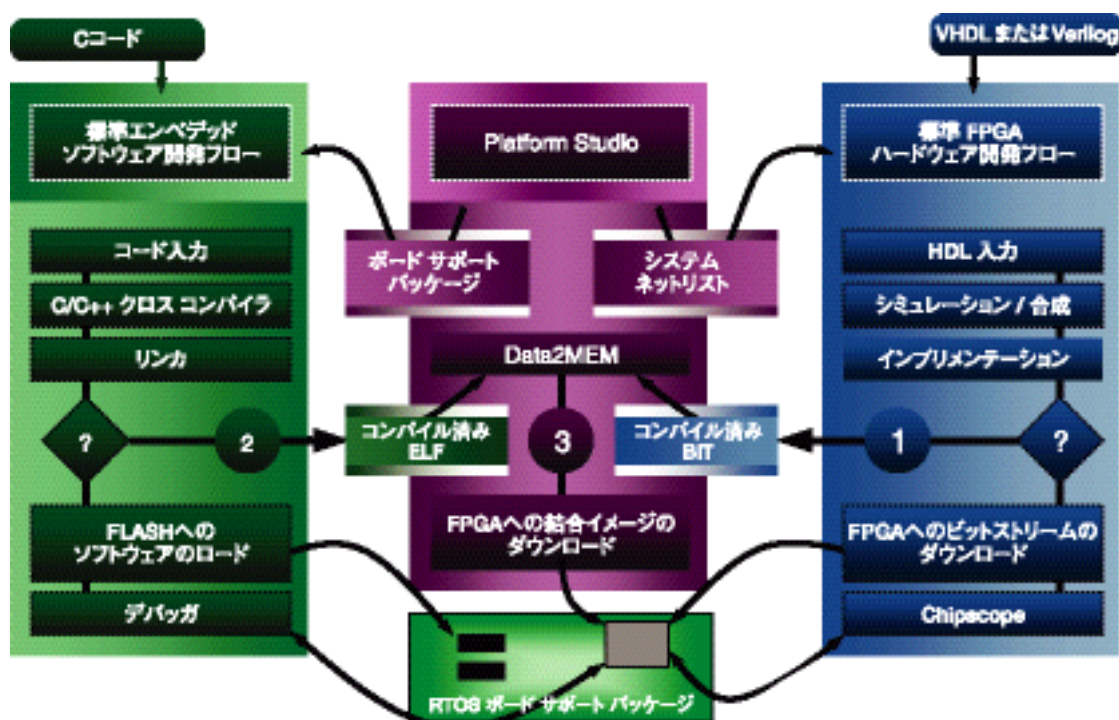
です。この傾向が確かなら、次のエンベデッド システムをデザインする前に、まず適切なツール スイートを選択することが、製品スケジュールとプロジェクト全体の成功に不可欠ということになります。

今日の開発環境には、あらゆるシステム オプションを理解し、複数の種類のプロセッサ コア、およびコプロセッシングと IP の作成/カスタマイズをサポートするプラットフォームを意識したツールが欠かせません。そしてそのデザイン ウィザードと自動モジュール生成機能によって、エラーの削減と開発プロセスの合理化が可能となり、さらにハードウェアとソフトウェアのデバッグの統合により、いち早くバグを見つけて修正できます。インテリジェント ツールを賢く選ぶことで、開発作業を迅速化し、最適なパフォーマンスを得られるのです。

標準的なフローとイノベーション

ザイリンクスは、プラットフォーム FPGA の開発を念頭に置いて Xilinx Platform Studio (XPS) ツール スイートを開発し、ハードウェア (FPGA 用の HDL/ネットリスト) とエンベデッドソフトウェア デザイン (コア エンジン処理するための C/ELF コード) の両方に対して、従来のフローをサポートします (図 1)。XPS は、あらゆるプログラマブル プロセッシング ソリューションをサポートする統一された開発ツール スイートを提供するだけでなく、エンベデッド開発を高速化する新機能を採用したことで IEC (International Engineering Consortium) の DesignVision Innovation Award を受賞しました。XPS のようなプラットフォームを意識したツールを使うことにより、デザイン ウィザードに従ってリアルタイムのハードウェア/ソフト

図 1 XPS デザインフロー



ウェア システムを素早く開発することができます。

デザイン ウィザードは基本システムの開発プロセスにて、最初の選択/仮定ではサポートされていないデザイン オプションをマスクで隠すことにより、エラーを削減します。たとえば、XPS は PowerPC™ ハードプロセッサと MicroBlaze™ ソフトプロセッサのコア デザインを両方サポートしますが、どちらか一方を選択すると、もう片方のオプションは自動的に除外されます。

IP のインポート、作成、カスタマイズは別のデザイン ウィザードにより合理化されており、一度使った IP を同じデザインの別の場所、もしくは将来的に別のデザインで簡単に再利用できるように、IP をサポートしています。

さらに、XPS はエラーの原因になりやすい退屈な手作業によるデザイン ステップに代わり、多彩な自動ジェネレータを用意することで革新的でスピーディな開発プロセスを実現しています。XPS は、プラットフォーム FPGA のシリコン特性とオプションを認識するため、選択したペリフェラルのソフトウェアドライバを生成したり、ボード オプション用のサンプル テスト コードを生成したり、また Wind River Systems 社の VxWorks やエンベデッド Linux など広く使用されている RTOS/eOS (リアルタイム オペレーティング システム/エンベデッド オペレーティング システム) 用に BSP (ボード サポート パッケージ) を作成したりすることも自動的に行えます。

また、XPS は C コードを FPGA ビットストリームにマージするユニークなユーティリティ (Data2Mem) を備えており、FPGA の配置・配線ツールを繰り返し実行するという無駄な時間を費やすことなく、ソフトウェアをリアルタイムで開発、デバッグすることができます。

加えて、FPGA ダウンロード、FPGA デバッグ、C コードダウンロード、およびソフトウェア デバッグ機能を単一のプローブを通して実行できる、統一された JTAG コネクション メソッドにより、効率を大幅に高めています。他の従来型メソッドは、複数のプローブと、異なるステップ間でのハードウェア コネクションの切り替えを必要とします。

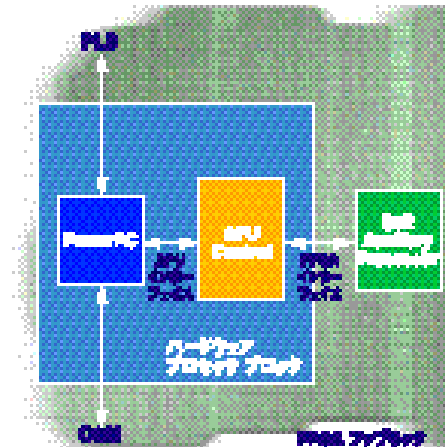
実際、XPS はハードウェアとソフトウェアのデバッグをユニークに統合することで、お互いがクロストリガできるようにしています。このようにシステムに新たな可視性を実現したことにより、欠陥がハードウェアとソフトウェアのどちらにあるかにかかわらず、エンベデッド設計チームはバグをいち早く見つけて修正できます。

コプロセッシングによる高速化

ほとんどのシステム要件を満たす、柔軟なプロセッサ ベースのプラットフォームがあるとします。必要なパフォーマンスを達成するため、コアのクロック周期をどれだけ高速にできるでしょうか？

ただし、プロセッサのクロックを高速化しただけではパフォーマンスの問題をすべて解消できないことは、既にご存じのことだと思います。ディスクリット プロセッサと放熱という物理的制約は別にして、クロックをいくら高速化しても、コアがすべ

図 2 Virtex-4 FX APU のブロック図



てのリアルタイム イベントに応答し、アプリケーションを稼働させ、そのサービスを完了できるとは限りません。そこで現在、優先度の低いタスクを主制御プロセッサから分離してオフロードし、メイン ユニットがリアルタイムで応答できるようにする「マルチプロセッサ」ソリューションが増えつつあります。

プログラマブル プラットフォームには、独自アプリケーションをカスタマイズできる市販デバイスを使って、この問題に対処する方法がいくつかあります。ハード、ソフト両方のプロセッサ コアをサポートする プラットフォーム FPGA が提供するソリューションは、優先度の高いタスクを内蔵のハード プロセッサに専念させ、優先度の低いタスクをソフト プロセッサ コアのインスタンス化にオフロードする方法です。既にエンベデッド PowerPC エンジンを走らせている プラットフォーム FPGA デバイスに、1 つ以上の MicroBlaze ソフト プロセッサを追加するオプションがあります。これをサポートするデバイスとしては、PowerPC 内蔵の Virtex™-II PRO FPGA や最新の Virtex-4 FX ファミリのデバイスがあります。これらデバイスの PowerPC コアは、マクロとして挿入され、シリコン内の FPGA ハードウェア リソースから構築した MicroBlaze IP コアによって補完できます。

この他有望な代替アプローチとして、「コプロセッシング」という概念をインプリメントすることが挙げられます。この場合、インテリジェント ツールを使って、エンベデッド PowerPC コアと高性能 FPGA ファブリックを直接接続し、高性能 FPGA ファブリックのハードウェア アクセラレータ機能が PowerPC の拡張機能として動作できるようにします。図 2 に示すとおり、演算が非常に複雑なアプリケーションをメイン CPU からオフロードすることにより、全体のシステム パフォーマンスを改善できます。

FPGA のハードウェア ファブリックは、その性質上、並列構造となっており、これを使えばクロッキング方式より数十倍も速くシステム機能を高速化することができます。この図 2 の例では、PowerPC コアは、APU (補助プロセッサ ユニット) によって補完されており、APU はデータ プロセッシング

グ、浮動小数点演算、ビデオ プロセッシングといったアプリケーションを扱える並列ソフト プロセッサと連結しています。こうした直接接続は、他のマルチコア プロセッサとアービトラード バス ソリューションと比べ、高バンド幅や低遅延ソリューションといった並列の利点を提供します。

パフォーマンス解析

しばしば設計のどこでパフォーマンスが劣化しているのか、原因を突き止める必要性が生じるでしょう。エンベデッド ソフトウェアの場合、コードの実行はデザイナーに「見えない」のが一般的ですので、そのデバッグと解析には多少手間がかかります。紙の上では設計が仕様を満たしているように見えても、非同期割り込みなど実際リアルタイム ハードウェアで走らせると、パフォーマンスの要件を満たさないことが少なくありません。その際、インテリジェント ツールを使うことにより動作デバイスの内側を見ることができ、ブラック ボックスの外であれこれ推測する必要がありません。

Xilinx Platform Studio のバージョン 7.1 は、ソフトウェアが実際にどう走っているかや、どこでパフォーマンスが劣化しているかを詳しく把握できる、一連のパフォーマンス 解析ツールを備えています (図 3)。実行時間を最も占めるのはどのソフトウェア関数か、どの関数が他の関数を呼び出すのか、また何回呼び出すのかを知ることで、エンベデッド設計が実際にどのように走っているか詳細に理解することができます。実行に時間がかかる関数や、他のルーチンによって何度も呼び出される関数は、コプロセッシングの拡張機能として並列ハードウェアに移すことにより、高速化できる主な候補となり得ます。

図 3 は、ソフトウェアの実行状況を追跡して、ツール上に詳しく表示することにより、効率の改善ができそうな部分を素早く

簡単に判別できることも示しています。仮定に基づいて何度も試行錯誤するのは時間がかかり、その分さほどパフォーマンスが改善されないものですが、これならそうした無駄を省けます。C コードや関数全体をインライン化すれば、ローカルなレベルのスピードアップは実現されますが、時間がかかるルーチンを高性能な FPGA ハードウェアに移してしまえば、10 倍の改善になることもあるのです。コードの実行を個別の関数名ごとにインテリジェントに表示することで、どのソフトウェア ルーチンを調整すべきかが正確にわかり、システム パフォーマンスの改善に大きな見返りが期待できます。

結論

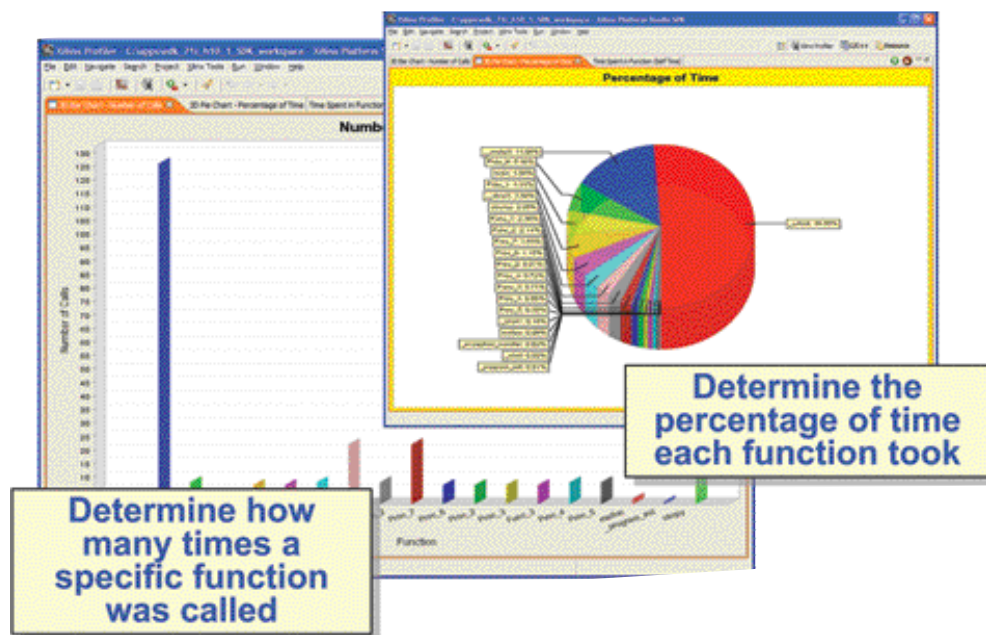
インテリジェントなプラットフォームを意識したツールは、エンベデッド ソフトウェア コードの非効率的な部分を識別するのに役立ち、パフォーマンスを最適化できます。どのソフトウェア関数を合理化する必要があるかを知ること、ハードウェア/ソフトウェアの分割を進め、プログラマブル FPGA ファブリック内のより多くのモジュールを高速化することが可能です。

パラレル FPGA ハードウェア リソースの高性能な特性、また Virtex-4 FX APU に代表される使いやすいプログラマブル コプロセッシング テクノロジーの登場により、これまで以上に高速かつ柔軟なエンベデッド プロセッシング システムを開発できるのです。ザイリンクスのツールとプログラマブル エンベデッド プラットフォーム FPGA を利用することで、リアルタイムの開発者はパフォーマンスの大幅な改善を達成できるのです。

Platform Studio ツールの詳細は、<http://www.xilinx.co.jp/edk/> をご覧ください。また、ザイリンクスの全エンベデッド プロセッシング ソリューションについては、<http://www.xilinx.co.jp/processor/> をご覧ください。



図 3 XPS によるパフォーマンス解析表示





Designing Control Circuits for High-Performance DSP Systems

高性能 DSP デジタル 信号処理システムを 実現する制御回路の設計

作業効率を向上させる基本テクニック

Narinder Lall

Sr. DSP Marketing Manager
Xilinx, Inc
narinder.lall@xilinx.com

Brad Taylor

SystemGenerator Applications Manager
Xilinx, Inc.
brad.taylor@xilinx.com

FPGA は、ASIC の代替品として利用されたり、DSP プロセッサの信号処理のパフォーマンス改善として利用されるなど、高性能な信号処理機能をインプリメントするためのエンジンとして飛躍的な進歩を遂げてきました。FPGA を信号処理用途として使う方法は数多くの書籍や記事で取り上げられていますが、こうしたシステムに制御回路を組み込む方法についてはあ

まり紹介されていません。

FPGA ベースの DSP システムに制御回路をインプリメントする場合、次の 2 つの点を決定する必要があります。

- ・制御回路をハードウェアにインプリメントするのか、それともソフトウェア アルゴリズムとして開発するのか？
- ・制御回路をできるだけ効率的に、かつ少ない労力で開発するには、どのようなビルディング ブロックが利用できるか？

ソフトウェアか、ハードウェアか？

設計の初期段階で、ハードウェアにインプリメントするアルゴリズムなのか、ソフトウェアにインプリメントするアルゴリズムなのか決定します。ソフトウェアにインプリメントするアルゴリズムは、ソフト マイクロプロセッサ（ザイリンクス PicoBlaze™ および MicroBlaze™ プロセッサ）、もしくは

表 1 ハードウェアとソフトウェア間のトレードオフの関係

	デジジョン用の クロック数	アルゴリズム の複雑さ	RTOS	浮動小数点 演算	ホストとの 通信
ハードウェア ベースの制御	1~10	シンプル	無	無	困難
ソフトウェア ベースの制御	100~100000	複雑	有	時々有	容易



ハード エンベデッド マイクロプロセッサ (PowerPC™405) を使用します。ハードウェアとソフトウェアのどちらのアプローチを採るかの判断基準は、表 1 のとおりです。

アプローチの選択にあたっては、さまざまな特性を考慮する必要があります。

• アルゴリズムの複雑さ

C コードの行数をそれほど必要としない単純なアルゴリズムなら、ソフトウェアとハードウェアのどちらでも簡単にインプリメントできます。何行の C コードが 1 スライスに相当するかを測る絶対的な尺度は存在しませんが、およその目安として、1 行で 1 ～ 10 のスライスに相当すると考えればいいでしょう。アルゴリズムが複雑になればなるほど、ハードウェアにアルゴリズムをインプリメントしてテストするのは難しくなります。何行もの C コードで書かれた複雑なアルゴリズムは、マイクロプロセッサにインプリメントするほうが簡単であり、実際ほとんどの設計者はこの方法を採用しています。

• RTOS の必要性

制御アルゴリズムに RTOS が不可欠な場合には、Virtex™-II PRO や Virtex™-4 FX FPGA上のハード エンベデッド PowerPC、もしくは外部マイクロプロセッサを利用するソフトウェア アプローチのほうが適しています。現在、Wind River 社と MontaVista 社は、これらマイクロプロセッサに対する RTOS を提供しています。

• ホストとの通信

ホスト プロセッサとの通信は、なんらかのバス アーキテクチャを必要とすることが多々あります。この場合、OPB などのバス アーキテクチャをサポートする MicroBlaze プロセッサ

や PowerPC プロセッサといったマイクロプロセッサが理想的です。ステート マシンを使ってハードウェア ベースでホストと通信することもできますが、多少複雑になります。

• デシジョンのスピード

デシジョンのスピードを「デシジョンあたりのクロック数」とする場合、スピードが必要なデシジョンについてはハードウェア回路が望ましいでしょう。数百、数千のクロック数でのデシジョンで問題ない場合、この程度のパフォーマンスであればソフトウェア ベースのアルゴリズムで十分対応できます。

• 浮動小数点演算の必要性

浮動小数点演算は制御機能とはあまり関係ありませんが、システムが制御のために採用するケースがないわけではありません。フィルタ係数の計算はその一例です。逆行列の計算が必要なソナー システムでは、開発のしやすさから、浮動小数点制御が好まれています。浮動小数点制御は、制御の精度が高く、アルゴリズムが固定小数点で利用できない場合にも使われます。

ハードウェアかソフトウェアかを決定したら、ビルディングブロックの利用を検討していきます。制御タスクの種類によって最適なブロックは異なります。

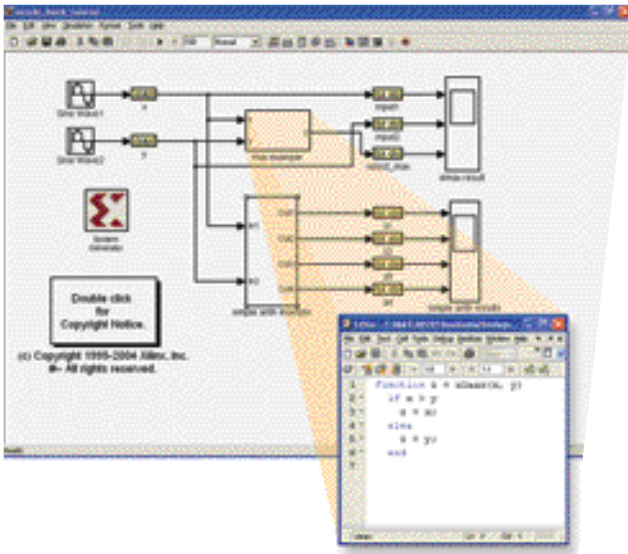
制御タスクの種類とそれぞれに適した回路

制御タスクにはたくさんの種類があります。ここでは、信号処理システムにありがちな次のような問題を取り上げます。

- ハードウェア ベース
 - データドリブンのマルチプレクシング

表 2 制御における問題とSystemGenerator で利用可能なツールキット

ツールキット	制御における問題				
	シーケンシング	ステートマシン	データドリブンのマルチプレクシング	サンプルレートの制御	低レート制御アルゴリズム
パターン生成コンポーネント (ROM、エクスプレッション、コンパレータ、カウンタ、ディレイ)	X				
M-コード		X	X		
コンパレータ			X		
FIFO/クロック イネーブル/アップダウン コンバージョン				X	
PicoBlaze					X
MicroBlaze/PowerPC 405					X

図 1 M-コードを使用するデータドリブンマルチプレクシングの例


- 有限状態 マシン (FSM) のインプリメント
- サンプル レートの制御
- シーケンシング - パターン生成

・ソフトウェア ベース

- 低レート制御アルゴリズムのインプリメント
- 物理レイヤ データパス (MAC レイヤ) の非常に複雑な制御 (本稿の対象外)

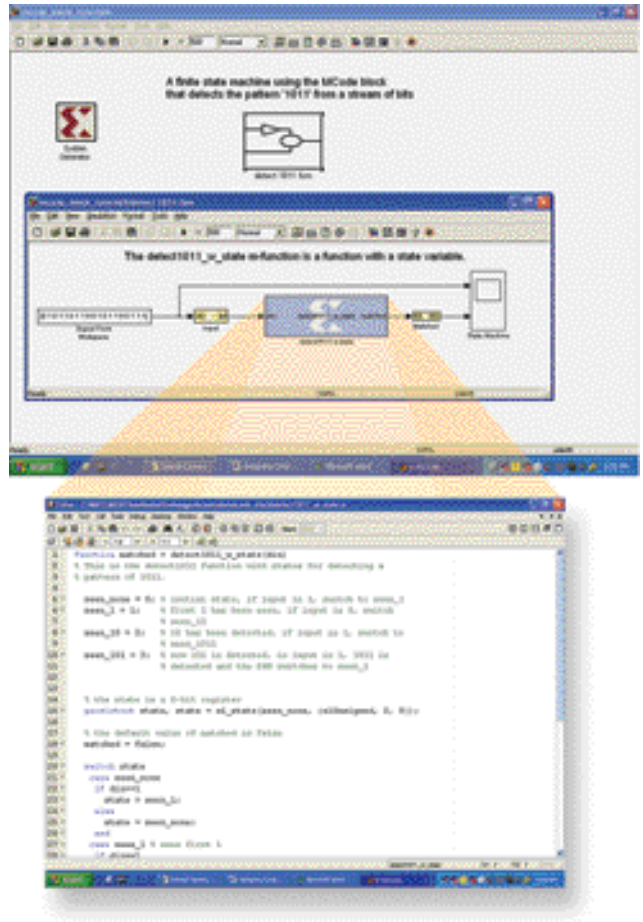
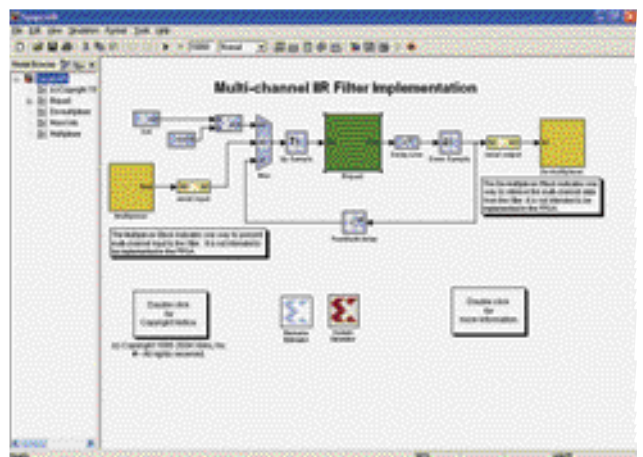
表 2 に、ツールと代表的な制御タスクの種類を紹介します。この表に記載の内容は、あくまでも 1 つの選択肢としてお考えください。DSP システムを FPGA にモデル化して設計するツールとしては、ザイリンクス SystemGenerator for DSP があります。ここでは、SystemGenerator に無償で同梱されている制御回路の使い方に関するサンプル デモを使用しながらいくつかの例を紹介していきます。

タスク 1: データドリブンのマルチプレクシング

現在の状態 (ステート) を監視する必要のない制御タスクの例として、データを監視しそのデータにテストを実施するデータドリブンのマルチプレクシングがあります。制御回路の出力はこれらテストの結果で決まります。図 1 に、データドリブンマルチプレクシングの例を示します。この図では、関数は xlmax という単純な MATLAB 関数により求められます。 $x > y$ でない限りは入力 y が選択され、 $x > y$ の場合は入力 x が選択されます。

タスク 2: FSM のインプリメント

入力からの現在のストアされた状態に基づいて決定しなければならない場合は、有限状態 マシンが使われます。ハードウェア ベースの高性能な DSP システムについては、1クロ

図 2 SystemGenerator におけるFSMのインプリメント例

図 3 サンプルレートの制御


ックごとに状態監視を実行する回路も珍しくありません。

FSM にはたくさんのインプリメント方法がありますが、SystemGenerator のデザイン内に FSM をインプリメントする最も一般的だと考えられる方法は、アルゴリズム開発者に評価の高い M-コード CASE 文を使う方法と、ハードウェアエンジニアに支持されている HDL を書く方法でしょう。HDL はブラックボックスを使って SystemGenerator のデザインに簡

単に取り込むことができ、必要に応じて ModelSim を使って協調シミュレーションが行えます。

図 2 のとおり、detect1011_w_state ファイルの MATLAB スクリプトとリンクしている M-コード ブロックを使うことにより、FSM を SystemGenerator にいとも簡単にインプリメントすることができます。このスクリプトの目的は、MATLAB のワークスペースから渡された信号から 1011 パターンを検出することです。

タスク 3 : サンプル レートの制御

高性能な DSP システムでは、サンプルがシステムまたはその一部に、FPGA クロックとは異なるレートで入力されることが多々あります。この場合、設計者はサンプル レートの制御を行う手法を習得する必要があります。サンプル レート制御回路のデザインを容易にする SystemGenerator 内のソリューションとして、アップ/ダウン サンプリング、クロック ドメイン、FIFO、クロック イネーブルがあります。

図 3 は、1 つの時分割二次型構造 (biquad) による複数の IIR フィルタをインプリメントする方法を示しています。具体的には、1 つのハードウェア biquad で構成された「フォルディング」したアーキテクチャを使い、それぞれ 4 つのカスケード biquad からなる 15 個の異なる IIR フィルタが実現されています。ハードウェア フォルディングとは、多数のアルゴリズムオペレーションを 1 つの機能単位 (加算器、乗算器) に時分割多重化するための手法です。オーディオや制御など、低サンプル レートのアプリケーションでは、ハードウェア リソースを時分割利用することで必要なシリコン面積を大幅に削減できます。

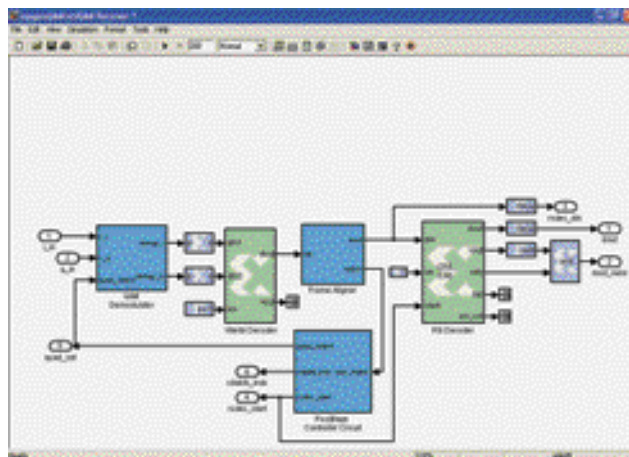
このデザインには、シリアル データとフィードバック パスの間でいずれかを選択する、2 入力マルチプレクサに送り込む制限付カウンタや、biquad を通してデータ レートを制御するアップサンプルおよびダウンサンプル ブロックなど、多くの制御回路が使われています。

タスク 4 : シーケンシング (パターン生成)

通常、シーケンシングの問題は、周期的な制御パターンの必要性から生じます。これは予測可能ではありますが、かならずしも現在のストアされた状態には依存しません。シーケンシングの一般的なソリューションは、1つのシンプルなパターン ジェネレータを使用することです。パターン ジェネレータは、カウンタ、コンパレータ、遅延素子、ROM といったビルディング ブロック、もしくは SystemGenerator ブロック セット内のロジック表現ブロックを使うことで構築できます。この手法の長所はシンプルさにありますが、多くの設計者はいまだに複雑で不要なステート マシンを使う傾向にあります。

図 3 の biquad ブロックの中に、パターン ジェネレータの例が示されています。biquad ブロック内のアドレス ジェネレータ (図にはない) は、RAM と ROM の全アドレス、およびフォルディングされている biquad モジュール内のシングルポート RAM に対する書き込みイネーブル信号を生成します。

図 4 PicoBlaze マイクロプロセッサを使って構築した QAM パケット検出回路



タスク 5 : 低レート制御アルゴリズム

ザイリンクスのマイクロプロセッサを使って低レートのアルゴリズムをインプリメントすることが、ますます一般化しつつあります。現在主流となっている 3 種類のプロセッサ、PicoBlaze 8 ビット プロセッサ、MicroBlaze 32 ビット プロセッサ、そしてエンベデッド IBM PowerPC 405 32 ビット プロセッサのいずれかを使うことで、手元のタスクに応じてスケールリングすることができます。オンチップ プロセッサを必要とする一般的なタスクとしては、フィルタ係数の計算、タスクのスケジューリング、パケットの検出 (例: FEC レシーバ)、および RTOS インプリメンテーションがあります。

図 4 は、PicoBlaze マイクロプロセッサを使って構築したシンプルな制御回路です。この例は、QAM 入力ソースに適応チャネル イコライゼーションとキャリア リカバリを実行する 16 QAM 復調器の受信パスを形成します。トランスミッタにより付加された同期マーカ (ASM) は、連結 FEC が適用される前に復調データから除去されます。PicoBlaze マイクロコントローラは RS デコーダを制御し、受信パケットのフレーム アラインメントを維持するとともに、デマッピング時の QAM-16 象限基準を周期的に調整します。

結論

高性能な FPGA ベースの DSP システムを対象に制御回路をインプリメントするときは、SystemGenerator に同梱されているたくさんのビルディング ブロックを利用することで作業を簡略化できます。表 1 と表 2 の、考慮すべき判断基準と、可能な制御ソリューションのリストを参考にしてください。

ザイリンクスの SystemGenerator ツールは、ここに紹介した以外にもたくさんのデザインを同梱し、995 ドルで提供しています。http://www.xilinx.co.jp/SystemGenerator_dsp/ でダウンロード可能な評価版は、60 日間無償でお試いただけます。



Achieve Breakthrough Performance in Your System

Virtex-4 による記録的パフォーマンスの達成

低消費電力と卓越したシグナル インテグリティを実現しながら
最高のパフォーマンスを提供

Adrian Cosoroaba
Marketing Manager
Xilinx, Inc.
adrian.cosoroaba@xilinx.com



今日のシステム パフォーマンスは、FPGA のクロック レートだけで語ることはできません。システムによってそれぞれ要件が異なり、達成可能な最大性能はロジック ファブリックの性能、I/O バンド幅、エンベデッド プロセッシング、DSP デジタル信号処理の性能など、さまざまな要因で決まるからです。また、これら要件は、消費電力の制限やシグナル インテグリティ、コストの制約を受けることがあります。

ザイリンクスは、これらの要件に応えるとともに、システムの目標性能を今まで以上に容易に満たすことができるよう、数百の顧客企業から意見や要望を聞いたうえで Virtex™-4 FPGA ファミリーを開発しました。本稿では、Virtex-4 FPGA が提供する新機能が、いかにシステム パフォーマンスの多彩な要件を満たすのに役立つかをご紹介します。

システム設計の問題点

プロセスの寸法が縮小されるたびに新世代のデバイスが登場し、半導体ベンダはより高速なクロック レートの提供が可能になります。しかし、今日のシステム パフォーマンスの問題点は、従来のグルー ロジックやクロックレートの最大化という範囲を超えています。たとえば、PC の世界では、システム パフォーマンスの真のボトルネックはクロック周波数ではなく、システムの他のブロックがどれだけ必要な周波数で連携できるかにあります。

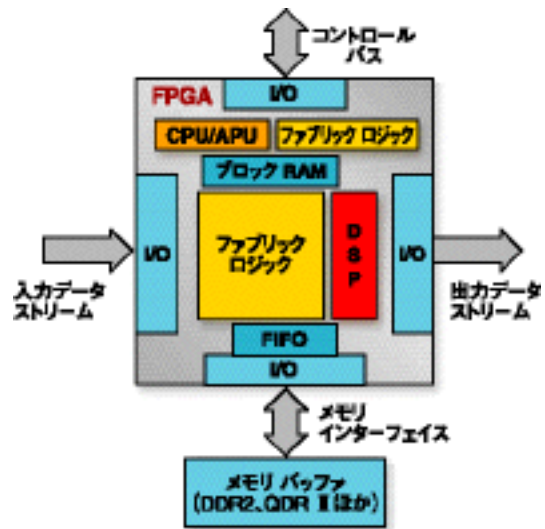
こうした問題点について、高性能な FPGA を採用したアプリケーションの視点からみていきましょう。ビデオ ストリーミング プロセッシングやパケット データ プロセッシング、ストレージ システム、無線基地局など、多くのアプリケーションは次のような類似の機能を採用しているようです。

- ・着信および発信データ ストリーム
- ・複数の接続規格のブリッジング
- ・算術演算と DSP（信号調整とデータ プロセッシング）
- ・外部メモリとのインターフェイス
- ・ステート マシン
- ・データ バッファリング
- ・エンベデッド プロセッシング（図 1）

これら機能を容易に開発できるようにするため、Virtex-4 FPGA には、エンベデッドでありながらパラメータ化が可能なハード IP として、ビルディング ブロックが用意されています。DSP スライス、エンベデッド CPU、専用 I/O 回路、オンチップ RAM（ブロック RAM、FIFO）といった複雑な機能を統合することで、1 個の FPGA デバイス内にプログラマブル システムを構築することが可能になります。

システムの要件を満たすには、I/O バンド幅、プログラマブル ロジック、オンチップ RAM、DSP、エンベデッド プロセッシングを正しく組み合わせる必要があります。Virtex-4 FPGA では、機能を理想的に組み合わせることができるよう、17 のデバイスからなる LX、SX、FX の 3 種類のプラットフォームを用意しました。

図 1 FPGA ベースのシステム



Virtex-4 FPGA は高度なロジック ファブリック機能だけでなく、カスタマイズされた XtremeDSP™ MAC とエンベデッド PowerPC™ プロセッサも提供し、目標とするデザイン性能を達成できるよう十分な能力を持たせています。

パフォーマンスを追求するうえで、I/O のバンド幅が足かせになることが少なくありません。I/O のボトルネックを解消するため、Virtex-4 FPGA は、1 Gbps ChipSync™ ソース同期回路と、622 Mbps ~ 10.3125 Gbps のシリアルトランシーバを内蔵し、バンド幅の目標達成に貢献します。

システム パフォーマンスのカテゴリ

パフォーマンスと Virtex-4 FPGA のさまざまな側面について、ロジック ファブリック、エンベデッド プロセッシング、DSP、オンチップ RAM、高速シリアル、I/O メモリ バンド幅、I/O LVDS バンド幅という 8 つの主なパフォーマンス カテゴリからみていくことにしましょう。図 2 は、それぞれのカテゴリにおけるザイリンクスと競合ベンダの 90 nm FPGA 製品との比較です。

ロジック ファブリックの性能

ザイリンクスは、Virtex-4 デバイスの開発に先進の 90 nm テクノロジーを用いることで、以前から高速であったプログラマブル ロジック ファブリックの性能をさらに高めています。柔軟なルックアップ テーブル（LUT）アーキテクチャ（任意の LUT を 16 ビット RAM や 16 ビット シフト レジスタに変換可）、高速キャリア チェーン、演算ブロックにより、さらなるパフォーマンス ゲインを実現しています。

ロジック性能を決める重要な柱である 500 MHz のグローバル クロック構造は、スキューやジッタ、デューティ サイクルのひずみを減らすため、フル差動としてあります。また、Virtex-4 FPGA には階層型のクロック構造（グローバル クロックとリージョナル クロック）およびクロック管理回路もあります。実物の

デザインを複数使い、ロジック ファブリックの性能を評価したところ、同じく 90 nm を採用する競合他社製品と比較して 70 % も優れていることがわかりました。また、Virtex-4 の性能はこれらデザイン全体の平均で見ても 15 % 優位に立っています。Virtex-4 デバイスでデザインした場合、スピード グレード面での優位性を効果的に達成できることになります。

エンベデッド プロセッシング

Virtex-4 FX プラットフォーム FPGA は、最大で 2 個の PowerPC 405 コアを提供し、それぞれのコアは 450 MHz で 702 DMIPS の性能を実現しながら、消費電力はわずか 0.45 mW/MHz に抑えられています。これは最高性能のソフト マイクロプロセッサ コアの 3 倍の性能に匹敵します。

さらに、新しく提供された補助プロセッサ ユニット (APU) コントローラにより、カスタム コプロセッサとハードウェア アクセラレータが統合されるため、さらなる性能の向上が容易に達成できます。APU コントローラは、FPGA にインプリメントされているコプロセッサ モジュールをエンベデッド PowerPC プロセッサに接続するための低遅延パスを提供します。これらユーザー定義のコンフィギュラブルなハードウェア アクセラレータ機能は、PowerPC 405 への拡張機能として動作し、複雑な計算を必要とするタスクから CPU を開放します。たとえば、浮動小数点の計算をハードウェアにインプリメントすると、ソフトウェア エミュレーションと比べて 20 倍の性能改善になります。イーサネットの接続は、PowerPC プロセッサの近くにインプリメントされている 10/100/1000 Mbps のトライモード イーサネット MAC で行います。

DSP の性能

XtremeDSP™ スライスはユーザー コンフィギュラブルな融通

性に富むブロックであり、従来のインプリメンテーションと比べて 2 倍の DSP 性能を実現しながら、消費電力は 7 分の 1 以下です。各スライスには専用の 18 × 18 ビットの 2 つの補数符号付き乗算器と、フィードバック パス付きの 3 入力加算器/減算器/アキュムレータがあります。512 個の Xtreme DSP スライスを 500 MHz で動作させた場合、Virtex-4 1 個で 256 GigaMAC/秒 (18 × 18 GMAC) の性能を発揮するのです。

XtremeDSP スライスをコンフィギュレーションすることで、ロジック ファブリック リソースをまったく消費することなく、乗算器、カウンタ、乗算アキュムレータなど、多数の機能をインプリメントできます。ファブリックの配線による遅延を招かずに複雑なシストリック機能をインプリメントできるので、大きなパフォーマンス ゲインを得られます。たとえば、32 タップの FIR インプリメンテーションでは、Virtex-4 の性能は競合他社デバイスを 40 % 上回ります。

オンチップ メモリの性能

Virtex-4 ファミリーは、旧世代で実証済みのオンチップ メモリ、18 Kb デュアルポート ブロック RAM のサイズと基本構造を継承していますが、データ出力パイプライン レジスタを追加することで最大 500 MHz まで高速化しています。2 つの各ポートの幅はそれぞれ個別に制御でき、書き込みモードでは、以前に格納したデータと新規データのどちらを自動的に読み出すか選ぶことができます。2 つの隣接するブロック RAM を組み合わせれば、スピードを損なわずに 32 K × 1 RAM を形成でき、また追加のロジックを使わなくてもハミング符号で自動的にエラー訂正を行う、深さ 512、幅 64 の RAM を形成できます。

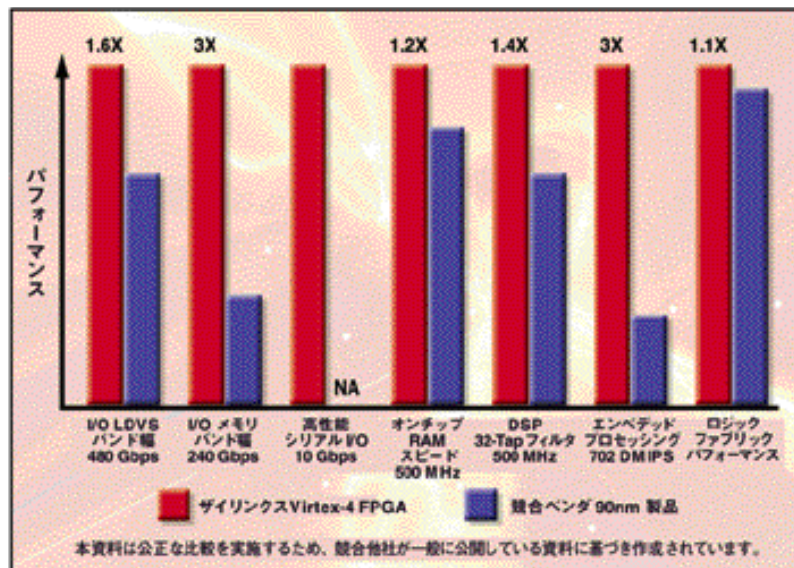
各ブロック RAM にはそれぞれ独自の FIFO コントローラがあります。このコントローラは、Virtex-4 FPGA のユニークな特徴で、追加のロジック リソースなしで 500 MHz の機能性を提供します。ブロック RAM の性能は、競合他社のデバイスと比較して少なくとも 20 % 上回ります。

しかし、FPGA 内部のブロックを高速に走らせるだけでは十分とはいえません。システム パフォーマンスを最大限引き出すには、FPGA とシステム内の他のコンポーネント間で効率的にやり取りできることが条件です。その点、Virtex-4 FPGA はチップ間、ボード間、そしてボックス間で可能な限り高いバンド幅を達成できる柔軟性を備えています。

高速シリアル I/O

デザインがより高速なインターフェイス スピードに移行するのに伴い、シリアル インターコネクトは消費電力とボード スペースを節約し、ま

図 2 Virtex-4と競合他社製品のパフォーマンス比較



たデザインの簡素化とコスト削減にもつながります。Virtex-4 RocketIO™ MGT のスピード性能は 622 Mbps ~ 10.3125 Gbps と、デバイスとしては最も幅広いレンジに属します。トランシーバはフル プログラマブルであるため、さまざまなスピードとシリアル規格に対応します。PCI Express、シリアル ATA、ファイバチャネル、ギガビット イーサネット、オーロラといった規格には、リンクレイヤ IP が利用できます。

メモリ I/O のバンド幅

今日、大多数のシステムは、一時的なデータ格納用として FPGA 外部のデータ バッファを必要とします。このバッファのバンド幅は、全体的な性能を決定する大事な要素になることがあります。

DDR2 SDRAM、QDR II SRAM、RLDRAM II などのメモリ インターフェイスはソース同期で、ピンあたりのデータ レートは 533 Mbps を超えます。メモリ バンド幅はピンあたりのデータ レートだけでなく、バスの幅によっても決まります。すべての I/O に内蔵されている ChipSync 回路は、物理レイヤ インターフェイスを簡素化し、他のプログラマブル ソリューションより 3 倍広いバスをインプリメントできるため、バンド幅は最大で 260 Gbps になります。

データ キャプチャの信頼性を高めるため、ChipSync 回路には遅延エレメントも組み込まれており、75 ps 単位で調整することが可能で、クロック信号とデータ信号の間で正しくアライメントを取ることができます。設計時でなく実行時にタイミングを調整できるため、デザインのマージンを格段に改善できるのです。また、ザイリンクスは、メモリ インターフェイスのインプリメンテーションをさらに短縮化できるよう、ハードウェア検証済みのリファレンス デザイン、開発システム、ソフトウェア ツールを提供しています。

LVDS I/O バンド幅

ChipSync テクノロジーを用いることにより、エンベデッド SERDES ブロックがパラレル 信号をシリアル化/シリアル 信号をデシリアル化してデータ レートを FPGA 内部の回路速度にマッチさせるため、差動パラレルバス インターフェイスのデザインを簡易化できます。加えて、デザイン マージンの向上を目的にビットやチャネルごとのデスクューを行えるため、SPI-4.2、XSBI、SFI-4、RapidIO などのインターフェイスを容易にデザインできます。Virtex-4 FPGA はすべての I/O に ChipSync テクノロジーを採用し、最も柔軟性に富む I/O ソリューションとなっています。これにより、より広い 1 Gbps の LVDS バスを実現でき、最大バンド幅が 480 Gbps になるため、競合他社製品を 60 % も凌駕しています。

その他パフォーマンスの課題

FPGA で目標のシステム パフォーマンスを達成したくて

も、シグナル インテグリティやコスト、消費電力がネックになって思うようにいかないことがあります。

革新的な Application Specific Modular Block (ASMBL) アーキテクチャにより、I/O、クロック、電源、グラウンドの各ピンを、チップの周囲だけでなくシリコン チップの任意の場所に配置することが可能です。これにより、I/O とアレイの依存関係、電源とグラウンドの配線、ハード IP スケーリングにかかわる問題が軽減されます。

さらに、Virtex-4 FPGA のパッケージング テクノロジー、Sparse Chevron (スペース シェブロン) により、電源とグラウンド ピンをパッケージ全体に均一に配置しています。設計者にとってのメリットは、シグナル インテグリティの改善です。ハワード ジョンソン博士が実証したように、Virtex-4 デバイスは競合他社製品と比較して同時スイッチング出力 (SSO) ノイズとクロストークが 7 分の 1 に低減されるのです。

プログラマブル ロジック、DSP スライス、ブロック RAM、I/O コラム、MGT、クロッキング、および PowerPC エンベデッド コアをコラム ベースでインプリメントした ASMBL アーキテクチャには、より柔軟なリソース割り当てという、もう 1 つの大きなメリットがあります。ザイリンクスがロジック リソースに最適化した LX プラットフォーム、DSP に最適化した SX プラットフォーム、エンベデッド プロセッシングと高速シリアル アプリケーションに最適化した FX プラットフォームの 3 種類の Virtex-4 プラットフォームを提供できるのは、まさにその恩恵といえます。

目標性能を実現するうえでもう 1 つの障害となるのが、デバイスの許容電力です。消費電力はクロック レートにしたがって増えるため、たとえ選択したデバイスが高性能であっても、目標性能の周波数で許容電力を超えてしまうことが考えられるのです。消費電力の低いデバイスを選択すれば、許容電力以内に収めながら目標性能を達成できるうえ、電源が小さくなり冷却の要件が緩和されることでシステム コストの低減、ひいては信頼性の向上にもつながります。

Virtex-4 FPGA は、スタティック電力を大幅に削減する特別なトリプル酸化膜の 90 nm テクノロジーを採用しています。また、エンベデッド IP などの一般的に使われている機能をインプリメントすることで、旧世代や競合他社製デバイスと比較してダイナミック電力をいっそう削減しています。Virtex-4 FPGA を測定、解析し、競合他社のツールやシリコンと比較したところ、競合他社の 90 nm FPGA より消費電力が 1 ~ 5 W 低いことがわかりました。

結論

Virtex-4 FPGA は、内蔵されている革新的なシリコン機能、広範なエンベデッド IP、トリプル酸化膜の 90 nm テクノロジー、そしてユニークなパッケージングを採用しており、設計者は最小のコストで画期的な性能を実現できます。Virtex-4 FPGA に関する詳細は、<http://www.xilinx.co.jp/virtex4/> をご覧ください。





The PlanAhead Experience

設計・分析ツール PlanAhead の利用効果

設計コストの削減とパフォーマンス向上に貢献してきた
PlanAhead ソフトウェア

Chris Zeh
Product Applications Engineer
Xilinx, Inc.
chris.zeh@xilinx.com

ザイリックスの設計・解析ツールである PlanAhead™ を使うことにより、パフォーマンスを改善し、開発期間の短縮と設計コストの削減に成功しているユーザーがますます増えています。こうしたユーザーから、FPGA のデザイン フローにおいて完全なパラダイム シフトが起こったという声を耳にしました。

私はこの数週間、ザイリックスのアプリケーション エンジニア数人と、PlanAhead ソフトウェアの使用感について話してきました。そこでわかったことは、PlanAhead 設計ツールが大きな力を発揮するのは、主に 3 つの障害に直面したときだということです。その 3 つとは、デザインがタイミングを満たせないとき、デザインがターゲット デバイスに収まらないとき、そして配置・配線の実行時間が長すぎる時です。また、配置・配線の結果を見る目的で PlanAhead ソフトウェアを使っているエンジニアが多いこともわかりました。

本稿では、私を含めてアプリケーション エンジニアが使用しているプロセスを紹介するとともに、ユーザーが PlanAhead ソ

フトウェアを使うことにどんなメリットを感じているのか、その統計を紹介します。ユーザーが独自にフロアプランを作成する場合、やり方がまずいと、タイミングの悪化や大型デバイスの使用を余儀なくされたり、配置・配線の実行時間が延びたりする恐れがあります。本稿では、目標とするパフォーマンスの達成を支援する PlanAhead の機能に関して、そのコンセプトを紹介します。

■ タイミングを満たすことができない場合

私たちアプリケーション エンジニアは、タイミングを満たすのが難しいデザインを受け取ると、まずフロアプランニングの制約なしでそのデザインを実行します。そのためには、既存のエリア グループと大型コンポーネントの物理的制約を取り払い、なおかつピンの配置をそのまま残す必要があります。配置・配線時間が非常に長い場合には、クリティカル パスを早めに見積もるため、PlanAhead ソフトウェア内のスタティック タイミング ツール、TimeAhead を実行します。TimeAhead による解析はまた、レジスタを追加してクリティカル パスをパイプライン化するなど RTL の修正が必要なデザインのエリアを指定するうえでも便利です。

可能な場合は、デザインに配置・配線を実行し、PlanAhead の環境内で配置とタイミングの結果を確認します。配置・配線したデザインのタイミング結果を見れば、実際の遅延に基づくクリティカルパスがわかります（図 1）。デバイスビューには、配置後のデザインと、TRCE レポートから報告されたタイミングパスが表示されます。

次に、タイミング レポートからデザインのクリティカルパスに基づいて PBlock またはエリア グループを作成します。この PBlock はフローティングでも、長方形など任意の形状としてでも定義することができます。PBlock にはデザイン内の任意のロケーションからロジックを入れることができ、RTL 論理階層に限定されません。インポートした配置・配線デザインを解析し、PBlock を構成する際、既に配置されたエレメントを PBlock を構成する起点とします。

PBlock は、各種モジュールの接続性に基づいてデザイン フローの方向を決めることができます。配置・配線したデザインに対して、I/O からモジュール、またモジュール間の接続性がネット バンドルを通して示されます。ここで、通常はデバイスへの搭載、または除去に関連するクリティカルパスをカバーするための PBlock を作成します。

回路図ビューには、デザインの接続性も表示されます（図 2）。PBlock は、デバイス、回路図、もしくはネットリストの各ビューに作成されます。クリティカルパスと、既存のフロアプランニングに関連するその他のタイミングパスは、デバイスと回路図の各ビューでハイライト表示できます。

通常、回路図ビューは、クリティカル モジュールの実際の配置を考慮することなく、それらモジュールとクリティカルパスを調べるために使われます。デバイスビューでは、既存の PBlock 周辺の過密度と、PBlock のリソース使用率が示されます。PBlock 内でタイミングがクリティカルな場合（リソース使用率が高い場合）、インターコネクットの長さや遅延を短縮するためロジックを高密度化することで、PBlock のサイズを変更し、過密なロジックにタイミングを満たせるだけの余裕を与えます。また、過密を緩和するため PBlock を移動したり、密集しているロジックを 1 つの PBlock にマージしたりすることも可能です。

PBlock を作成、配置し、その後フロアプランニングの制約をかけて配置・配線を実行します。また、ブロック RAM、DSP48、DCM などの大型コンポーネントとクリティカルロジックのロケーションをロックします。次にデザインに配置・配線を実行し、タイミングレポートとデザインの配置を検討し制約を修正して、このプロセスを繰り返します（図 2）。さらに、タイミングクリティカルなパスについては、2、3 のコンポーネントをグループ化できるよう、PBlock の中に別の小さな「子」PBlock を作成することもできます。

配置・配線は、デザイン全体だけでなく、個々の PBlock でも実行されます。こうすることで、モジュールや PBlock 内でデザインの残りの部分に影響を与えることなくタイミングを満たせるかどうか判断できます。特定の PBlock でタイミングを満たすことができない場合、ソースコードに戻って書き直します。

PlanAhead ソフトウェアは、任意のモジュールに対して、

図 1 タイミング解析とサンプルデザインの配置ビュー

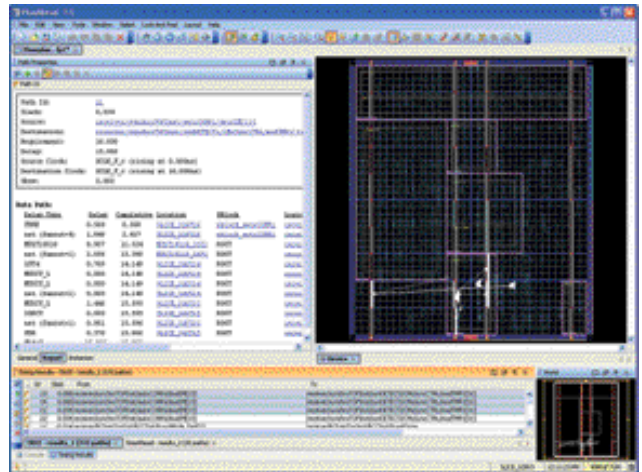
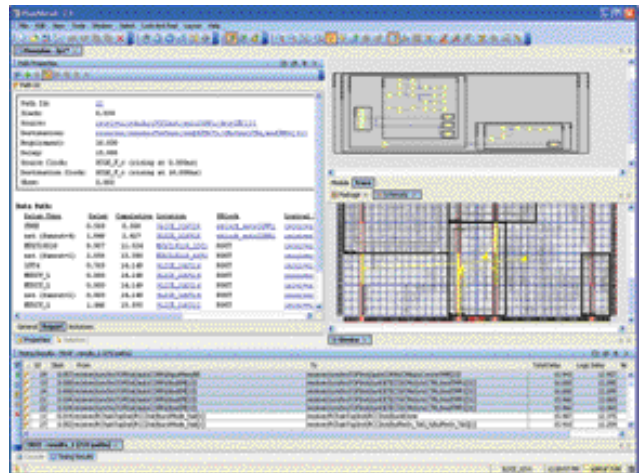


図 2 回路図ビューにおけるデザイン接続性、より大きなコンポーネントを配置したときのタイミング解析



アップデートしたネットリストをインポートし、最初に作成したオリジナルの PBlock を維持できます。タイミングの要件が満たされた後、その PBlock の全エレメントにロケーションの制約を設定します。これにより、デザインの残りの部分を配置・配線するときにもタイミングが保持されます。

私たちが目にするタイミングの失敗は、そのほとんどがセットアップ（時間）違反によるものです。中にはホールド（時間）違反のこともあり、その場合は修正に手間がかかります。ホールド違反が起こる最大の原因は、クリティカルなクロッキングコンポーネントの配置にあります。DCM とそれに対応するグローバルバッファがデバイスの反対側に配置されているとき、膨大な数のクロックスキューとホールド違反が起こるのです。

PlanAhead ソフトウェアは、この種の問題をデザインフローの早い段階で識別できるよう、堅牢な DRC チェックセットを備えています。これらコンポーネントの配置を見るには回路図ビューとデバイスビューを使います。その後、DCM またはグローバルバッファのいずれかの配置を動かし、これらコンポーネントがお互いに近くなるようにします。

デザインに不適合がある場合

デザインがターゲット デバイスにうまく適合しないときは、フロアプランニングの制約を外した状態で実行します。通常は PBlock のオーバーラップが原因であるため、オーバーラップする PBlock を除去します。前のセクションで述べたように、配置・配線ツールを楽に使えるよう、大型コンポーネントは手作業で配置します。

デザインにとってタイミングがあまり重要でない場合、そのデザインの主要な階層をブロック化して PBlock を作成します。これらの PBlock は、モジュールとネット バンドルの接続性に基づいて、デザイン フローの方向を決めるために使われます。バンドルは、そのバンドルのサイズに基づく接続性の量を示します。PBlock は、ネット バンドルと PBlock の統計に基づき、PBlock を駆動するコンポーネントの近くに配置します。また、PBlock にデザイン全体を配置し、デザインのこの部分をできるだけ集積して詰め込むよう MAP に指示するため、PBlock に圧縮属性を使います。この属性は、デバイスの残りのスペースに他のロジックを入れる余裕を増やす半面、タイミングに悪影響を与えます。

さらに、デザインの個々の PBlock に配置・配線を実行します。これにより、PBlock に配置したロジックを、定義済みの PBlock サイズ内に配置・配線し、最終的に指定したデバイスに収めることができます。PBlock のサイズは、MAP が失敗するまで手作業で圧縮できます。MAP が失敗したら、既に圧縮が成功している 1 つ前の PBlock サイズに戻します。

PBlock の統計では、使用率が 100 % を超過していると表示されるかもしれませんが、実際に判断するのは MAP です。他にもタイミング クリティカルでない PBlock が存在する場合はこのプロセスを繰り返し、圧縮のための属性を使用してそれら PBlock を可能な限り高密度に集積していきます。

この時点で、すべての PBlock とデザイン全体に配置・配線を実行します。それでも適合しない場合は、より大型のデバイスを選ぶか、ソース コードに戻って書き直します。ほとんどの場合、PBlock 単位で配置・配線を実行し、アップデートしたモジュールを PlanAhead 設計ツールにインポートします。このプロセスは、既存の PBlock サイズと物理的なロケーションの制約を保持します。デザインがターゲット デバイスに適合したら、デザイン全体に対して物理的なロケーションをロックします。

配置・配線時間が長すぎる場合

アプリケーション エンジニアが受け取る大多数のデザインで最もクリティカルな問題は、前述の 2 つです。配置・配線の実行時間は、これらの問題が解決されれば短縮されるため、ツールが目標とするパフォーマンスの達成に長い実行時間を必要とすることはありません。前述した問題が特に発生しなければ、タイミング レポートと配置・配線したデザインを見て、主要モジュールの PBlock を作成します。

図 3 TimeAhead でタイミング解析をしたデザインにおけるデバイス、パッケージ、回路図の各ビュー

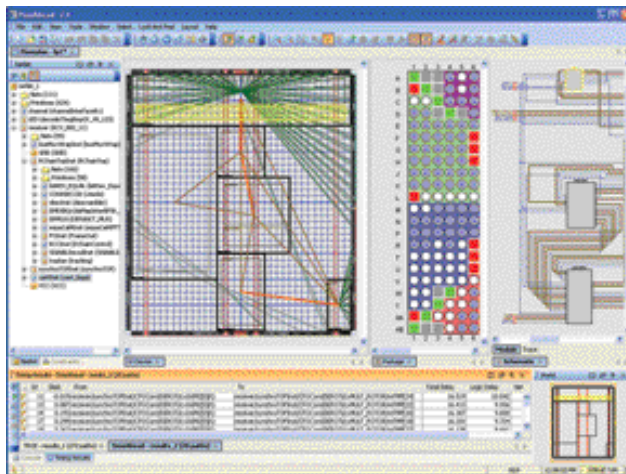


表 1 ユーザーにおけるPlanAhead デザインツールの使用前・使用後の状況

	PlanAhead 使用前	PlanAhead 使用后
ユーザー A	90 % の LUT 使用率 タイミング エラー	82 % の LUT 使用率 2 つの回路において タイミング エラーを解決
ユーザー B	タイミング エラー 要求：250 MHz 実際：235 MHz	タイミング エラーを 完全に克服
ユーザー C	タイミング エラー 長い配置・ 配線時間	タイミング エラーを解決 配置・配線時間を 85 % 短縮
ユーザー D	タイミング エラー デザイン サイズに 対してデザインが 大きすぎる	タイミング エラーを解決 リソース使用率を 11 % 削減し、デザインをデバ イスに収めることに成功
ユーザー E	タイミング エラー	タイミング スコアを 98 % 削減 タイミング エラーを 97 % 削減
ユーザー F	タイミング エラー 長い配置・ 配線時間	タイミング エラーを 90 % 削減 配置・配線の実行時間 を 30 % 短縮

配置・配線の実行時間を短縮する鍵は、デザイン全体に対する物理的ロケーションの制約です。デザインに対する正しい配置制約を見つけるためには、各 PBlock に配置・配線を実行し、おのこのタイミング制約に従ってそのサイズを決めます。これは、デザイン内の論理階層に基づきインクリメンタルなアップデート アプローチを使うときにも可能です。PlanAhead ソフトウェアは、反復使用されるモジュール ネットリストを使ってデザイン内の任意のモジュールを選択的にアップデートできます。これにより、インクリメンタルな合成手法を利用できることになります。モジュール と I/O パッド間、またモジュール間そのもののコネクティビティを見ることで、PBlock の配置を決定できるのです。タイミング クリティカルでない PBlock は、手作業で圧縮して、より大型のコンポーネントを配置します。

その後、デザインに配置・配線を実行し、タイミング レポートとデザインの配置を見ながら制約を変更することで、このプロセスを繰り返していきます。こうした問題がいくつも複合している場合は、複数のアプローチを組み合わせ、各テクニックを繰り返すことで、問題を解決していきます。

デザインの表示

多くのアプリケーション エンジニアは、配置・配線を実行する後に、デザインを視覚的に見るために PlanAhead 設計ツールを

使います。このツールを使用すると、タイミング パスとデザインの配置を画面上で見ることができます。回路図、デバイス、パッケージの各ビューにより、デザインを多角的な視点で理解することができます (図 3)。デバイスビューでは、SLICE/CLB と I/O バンク間の相関関係がわかります。タイミング パスもわかりやすく表示され、修正措置を講じることができます。

アプリケーション エンジニアとの会話を通じて、大多数のユーザーが設計作業とその結果に満足していることがわかりました。表 1 は、PlanAhead ソフトウェアを使う前と後で、実際にユーザーの状況がどう変化したかを示しています。

結論

本稿では、ザイリンクスのユーザーとアプリケーション エンジニアが、主に 3 つの問題を克服するために PlanAhead 設計ツールをどう使用しているかを取り上げました。ここで紹介した考え方は、パフォーマンスの向上を達成するうえで役立つものだと思信しています。最近、あるユーザーより、「PlanAhead ソフトウェアを使うことにより、検証とプロトタイプを迅速化でき、繰り返しの回数も減った」という声を耳にしましたが、おそらく皆さんも同意見ではないでしょうか。

PlanAhead 設計ツールの詳細は、<http://www.xilinx.co.jp/planahead/>をご覧ください。



ザイリンクス トレーニング スケジュール

ザイリンクスでは、大規模、高速 FPGA を対象にした FPGA 設計のための各種トレーニングを各地で開催しております。是非ご利用ください。

コース名	日 程	開 催 地
ISE デザイン入力	11月 8日(火)	ザイリンクス 東京会場
	12月 1日(木)	ザイリンクス 東京会場
FPGA 設計導入	11月 9日(水)	ザイリンクス 東京会場
	12月 2日(金)	ザイリンクス 東京会場
FPGA 設計実践	10月 19日(水)～20日(木)	新光商事 東京会場
	27日(木)～28日(金)	メメック ジャパン 東京会場
	11月 21日(月)～22日(火)	新光商事 大阪会場
	12月 6日(火)～7日(水)	ザイリンクス 東京会場
Virtex-4 デザイン	12月 21日(水)～22日(木)	メメック ジャパン 東京会場
	10月 20日(木)～21日(金)	ザイリンクス 東京会場
	11月 17日(木)～18日(金)	ザイリンクス 東京会場
	12月 15日(木)～16日(金)	ザイリンクス 東京会場
MGT シリアル I/O デザイン	11月 24日(木)～25日(金)	ザイリンクス 東京会場
アドバンスド FPGA 設計	11月 10日(木)～11日(金)	ザイリンクス 東京会場
	14日(月)～15日(火)	新光商事 大阪会場
シグナルインテグリティと基板設計の基礎	12月 13日(火)～14日(水)	ザイリンクス 東京会場
エンベデッドシステム開発	11月 20日(木)～21日(金)	ザイリンクス 東京会場
DSP デザインフロー	11月 14日(月)～16日(水)	ザイリンクス 東京会場

* すべてのトレーニングは、ザイリンクス認定インストラクターによるオフィシャルトレーニングです。

* 日程および会場は、都合により変更となる場合もございます。最新情報はザイリンクストレーニングWebサイトをご覧ください。

▼▼▼ 詳細とご登録はこちらから ▼▼▼

http://www.xilinx.co.jp/education_schedule/

Never Design Another FIFO

最適化された FIFO ソリューションの設計

FIFO Generator IP core はあらゆるコンフィギュレーションに最適化された FIFO ソリューションを提供

Tom Fischhaber

Staff Design Engineer, IP Solutions Division
Xilinx, Inc.
tom.fischhaber@xilinx.com

James Ogden

Design Engineer, IP Solutions Division
Xilinx, Inc.
james.ogden@xilinx.com

デジタル設計の世界では、非常に手のかかるデータ操作やバッファリングに必要な構造体として、First-In First-Out メモリ キュー (FIFO) がいたるところで使われています。ではクロック ドメインの交差(クロッシング)はすべて正しくタイミングされ、同期化されているのでしょうか? 16 ビットのデータパスを 64 ビットに変換するにはどうすればいいのでしょうか? FIFO デザインにおけるこれらの問題は難しく、実装に時間がかかるうえ、エラーを招きやすいものです。サイリンクスの FIFO Generator コアは、こうした問題を解消し、コンフィギュレーション可能なグラフィカル ユーザー インターフェイス (GUI) を通して各種の複雑な FIFO デザインを提供しているため、設計者はシステムの要件に専念できます。

アプリケーション ノートからリファレンス デザイン、IP コアにいたるまで、サイリンクスは FIFO の開発に長い歴史を持っています。FIFO Generator により、想像し得る限りほとんどの FIFO コンフィギュレーションが、フルに最適化、プリエンジニアリングされたソリューションとして、Core Generator™ を通して提供されます。FIFO Generator は、ブロック RAM や分散 RAM、シフト レジスタ、Virtex™-4 ビルトイン FIFO など、多彩なメモリ タイプをサポートします。また、コアは 1 個の共通クロック、もしくは 2 個の独立ク

ロックのいずれかを持つ書き込みおよび読み出しインターフェイスをサポートします。さらに、その他のオプションも GUI を通して簡単にカスタマイズできます。本稿では、FIFO Generator ソリューションの利点と、このソリューションがユーザーのニーズに適合する FIFO を短期間で開発するうえでどう役立つのかにスポットを当てて紹介します。

共通および独立クロックドメイン

FIFO Generator は、書き込みと読み出し操作に対して、1 個の共通クロックと 2 個の独立クロックの両方で FIFO をサポートします。共通クロック コンフィギュレーションは、さまざまなステータス フラグをサポートする小型で高速、待ち時間の短い FIFO を提供するため、1 クロックによるデータ バッファリング アプリケーションに最適です。

独立クロック コンフィギュレーションはさらに利便性に優れ、極めてエラーを招きやすい FIFO デザインをボタン 1 つで解決します。FIFO Generator はクロックドメイン間の同期化を処理し、位相と周波数に条件を課すことはありません。FIFO Generator は独立したクロック デザインの複雑さを解消するだけでなく、あらゆるステータス フラグを含むさまざまな追加機能を備えているため、アプリケーションに応じて FIFO Generator をカスタマイズできます。

図 1 は、FIFO Generator GUI の最初のページで、さまざまなメモリ タイプを使って 1 個の共通クロック、もしくは 2 個の独立クロックを使用する FIFO をどのように構成するかが理解できます。図 1 は、FIFO Generator v2.1 リリースにおいて、各コンフィギュレーションでサポートされている主な機能も示されています。このリリースは、IP Update 1 をインストール済みの ISE™ 7.1i ソフトウェアに無償で提供されています。

Virtex-4 にビルトインされている FIFO サポート

Virtex-4 アーキテクチャには、あらゆるオンチップ ブロック RAM を備える ビルトイン FIFO コントローラがあります (Xcell ジャーナル 52 号の記事「容易になった FIFO」を参照)。FIFO Generator は、この新しいビルトイン FIFO を使うことで、高性能な独立クロック FIFO に容易にアクセスできるようにし、消費電力を大幅に低減しながら FPGA ファブリック リソースを節約します。

FIFO Generator は、任意の幅と深さを持つ FIFO と、追加のステータス フラグを提供することで、ビルトイン FIFO の機能を拡張しています。複数のエンベデッド FIFO の連結とステータス フラグに対する追加ロジックは自動的に処理されるため、最小限の FPGA リソースで非常に高性能なデザインをサポートできるのです。図 2 に、ビルトイン FIFO デザインの機能図を示します。

非対称アスペクト比/FWFT

データ幅変換を必要とするアプリケーションは、異なる周波数で動作する際にほとんどの場合 2 個のクロックを必要とします。周波数に関連性がある場合、このタスクは容易に実装できます。しかし、2 つのクロック レートになんの関係もない場合は、かなり難しくなります。FIFO Generator は、2 つのクロックドメイン間にどんな関係があっても、格納データの幅を自動的に変換します。図 3 は、8 ビットの書き込みインターフェイスと 2 ビットの読み出しインターフェイスを持つ FIFO がどのように動作するかを示しており、この場合は 4 : 1 の書き込み/読み出しアスペクト比になります。

異なるバス幅と独立クロックを持つ FIFO はデザインが複雑ですが、FIFO Generator を使うことにより、他の FIFO コンフィギュレーションを作成、使用すると同じくらい簡単になります。また、FIFO Generator は、この機能以外にもフルセットのステ

図 1 FIFO Generator の GUI

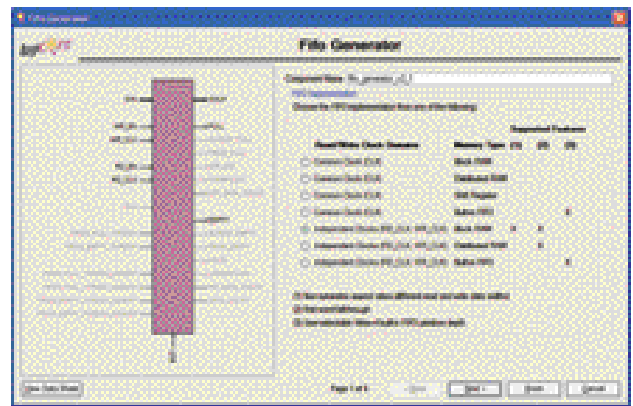
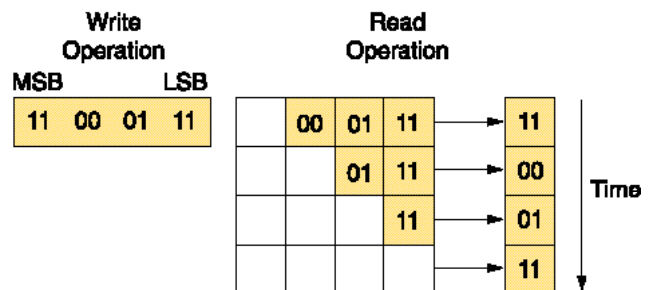


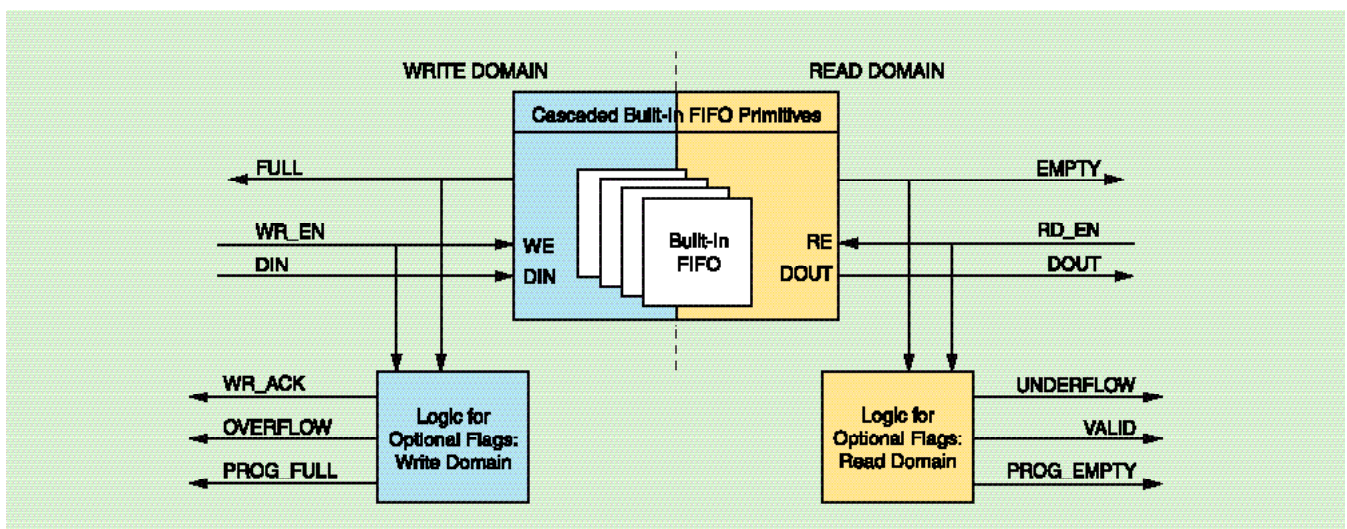
図 3 非対称的な書き込み/読み出しインターフェイスを持つ FIFO のアスペクト比(4 : 1)



ータス フラグや First Word Fall Through (FWFT) など、豊富な機能を備えています。

FWFT は、読み出し操作を発行することなく FIFO から次のワードを先に見る機能を提供するものです。FIFO にデータがあれば、最初のワードが FIFO を通過し、出力バスに自動的に出力されます。FWFT は、短い待ち時間でデータにアクセスする必要があるアプリケーションと、読み出されたデータの内容に応じてスロットルを必要とするアプリケーションに便利で

図 2 FIFO Generator によって拡張されたビルトイン FIFO デザイン機能図





す。FWFT のサポートは、FIFO Generator v2.1 リリースより新たに提供しています。

メモリ タイプ

Virtex-4 デバイスのビルトイン FIFO に加えて、FIFO Generator はブロック RAM、分散 RAM、シフト レジスタを含む各種メモリ タイプをサポートします。FIFO Generator は選択したメモリ タイプの幅と深さを組み合わせることで、常に最適化したソリューションを生成します。表 1 に、各メモリタイプの利点をまとめます。

結論

ザイリンクスの FIFO Generator コアは、複雑な機能をボタ

ン 1 つで扱えるオール イン ワンの FIFO ソリューションです。従来の FIFO 機能に加え、Virtex-4 FPGA に対するビルトイン FIFO のサポート、非対称アスペクト比、FWFT などの特殊機能も用意されています。このコアがシステム設計者に与えるのは安心感です。FIFO デザインはザイリンクスに任せ、ユーザーは手がけているデザイン課題の解決に専念できるのです。

ザイリンクスでは、今後も Xilinx FPGA ファミリー向けに FIFO Generator を拡張し、ユーザーが次世代のデザイン要件を満たすための高い価値を持つ機能を追加し続けていく所存です。

FIFO Generator の詳細は、[http://www.xilinx.co.jp/xlnx/xebiz/designResources/ip_product_details.jsp?key=FIFO Generator/](http://www.xilinx.co.jp/xlnx/xebiz/designResources/ip_product_details.jsp?key=FIFO%20Generator/) をご覧ください。Virtex-4 ビルトイン FIFO とその利点については、http://www.xilinx.co.jp/xcell/xl52/jp52xcell_13.pdf をご覧ください。



表 1 各メモリタイプのサポート状況と利点

	独立クロック	共通クロック	バッファリング が小さい アプリケーション	バッファリングが 中ぐらいから大きい アプリケーション	高パフォーマンス	最小の ファブリック リソース
ビルトイン FIFO	●	●		●	●	●
ブロック RAM	●	●		●	●	●
シフトレジスタ		●	●		●	
分散 RAM	●		●		●	

ザイリンクス ウェブセミナー

プログラマブル ロジック ソリューションのリーダーであるザイリンクスの世界最高速FPGAのさまざまな機能と活用方法をご紹介します。コストを抑え、最大のパフォーマンスを実現するための最新情報を手に入れてください。

好評配信中 >>>

- Virtex™- 4 シグナル インテグリティ
- Virtex™- 4 ローパワー・アドバンテージ
- Virtex™- 4 メモリインターフェイス・アドバンテージ

セミナー内容の詳細／ご視聴は今すぐこちらから>>> <http://www.xilinx.co.jp/webseminar/>





Using a CPLD to Implement a QWERTY Keypad

ザイリンクス CPLD を 使った QWERTY キーパッドの実装

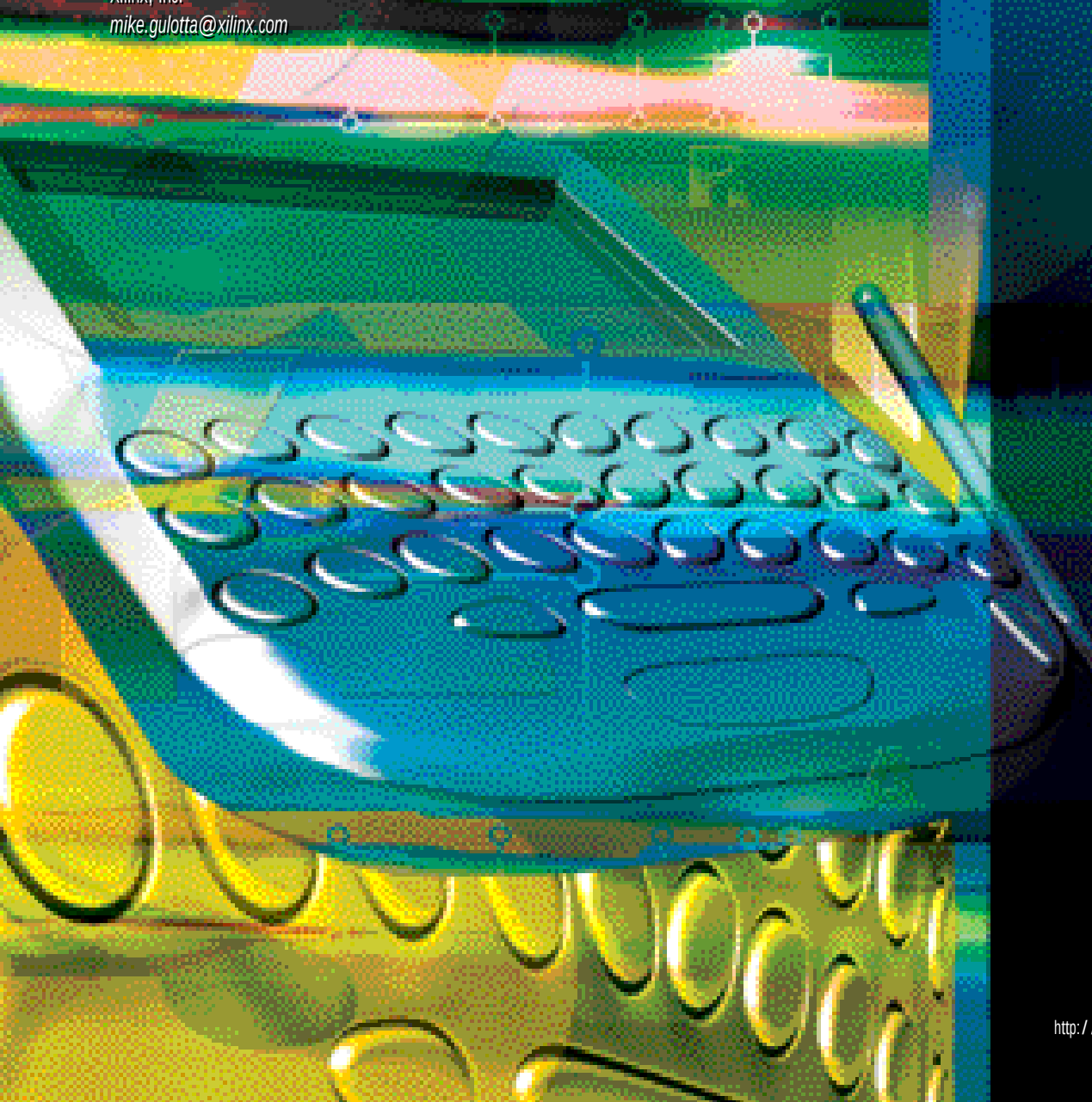
典型的な携帯端末用 DTMF キーパッドを
QWERTY キーパッドに変える

Mike Gulotta

Xilinx FAE

Xilinx, Inc.

mike.gulotta@xilinx.com



携帯電話をはじめとする携帯端末に次々と機能が追加されていくなか、デザインの選択肢は常に進化しています。テキストメッセージングや Web ブラウジングといったポピュラーな機能は、より多くのデータ入力を必要としますが、従来のデュアルトーン マルチ周波数 (DTMF:「0~9」「#」「*」などのキーで構成) キーパッドだと使い勝手がよくありません。このタイプのキーパッドでは、マルチタップ方式でデータを入力しなければならず、効率が悪いうえエラーを招きやすいのです。

テキストを入力しやすくする 1 つの方法は、QWERTY キーパッド (図 1) を使うことです。DTMF のハンドセットだと通常 12 個のキーですが、QWERTY には 40 個以上のキーがあります。ただ、キーが多いぶんハンドセットが大型化し、電子部品の数が増えるというデメリットがあります。

テキスト メッセージを書くユーザーにとっては、文字数制限に目をつぶっても QWERTY キーパッドのほうが便利でしょう。テキストの入力がはるかに楽ですし、メッセージやデータの入力に親指を 2 本使えるからです。最近、携帯電話メーカ数社から、テキスト ユーザー向けに QWERTY キーパッド搭載のハンドセットが発表されています。

データ入力のキーパッドをデザインする方法はいろいろありますが、これといった規格は存在しません。本稿では、従来の DTMF タイプのキーパッドに新たにキーを追加するというデザイン上の課題について、1 つの実現可能なソリューションを検証します。

図 1 QWERTY キーパッドの配列 (モトローラ社のモデル A630)



QWERTY のビルディング ブロック

本稿のソリューションには、ザイリンクス CoolRunner™-II CPLD を使用します。CoolRunner-II CPLD は、低消費電力、小型のパッケージ オプション、低コストという特徴により、このアプリケーションに理想的です。

DTMF から QWERTY キーパッドに移行する場合、より多く

図 2 8 個の GPIO を使うプロセッサに接続された簡単な 4 × 4 のキーパッド

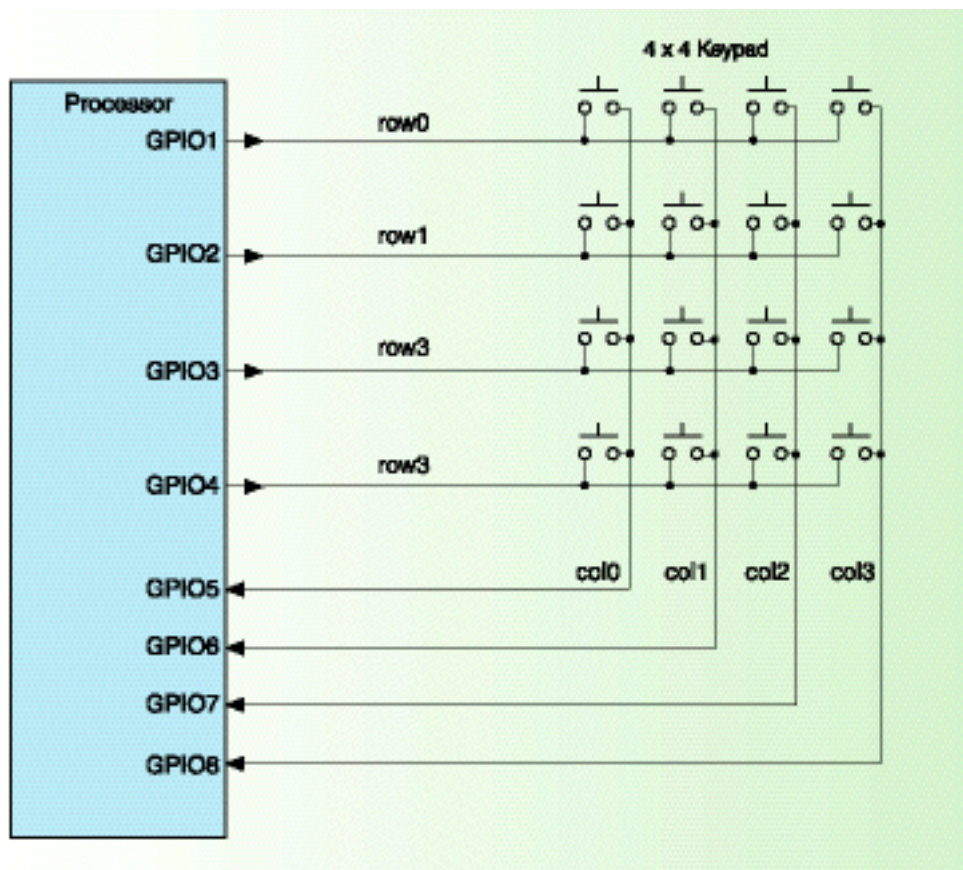
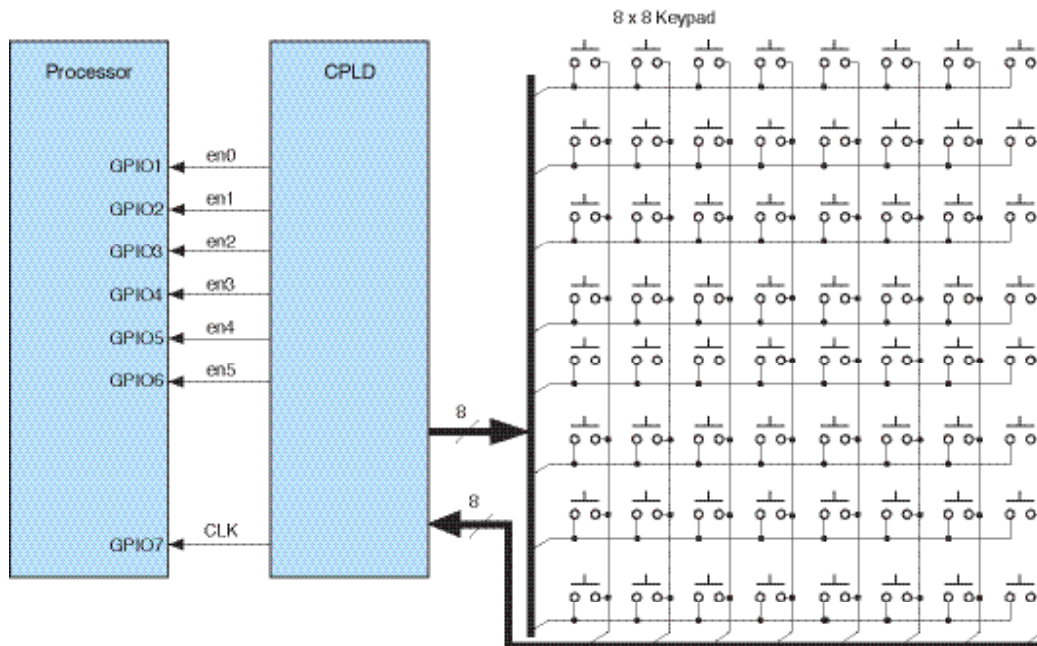


図 3 CoolRunner-II CPLDによるGPIO の拡張



のキーが必要となるため、汎用 I/O (GPIO) の数も増えます。たとえば、DTMF キーパッドには 4 行、3 列しかありませんが、QWERTY キーパッドは 8 行、8 列になるケースもあります。しかし、キーパッドのサイズはエンド システムの要件に応じて異なります。

一般に、キーパッドの行と列へのインターフェイスにはプロセッサか DSP が使われます (図 2)。プロセッサは、行をスキャンし列を監視することで、ロジックの変化をモニタします。ロジックの変化は、ユーザーがボタンを 1 個押したことを意味します。プロセッサは、どの行がスキャンされ、どの列が状態を変えたかを知ることで、押されたボタンを推定することができます。

I/O の拡張

QWERTY キーパッドのように、比較的多くの I/O を必要とするキーパッドをデザインする場合、手持ちのプロセッサでは GPIO が足りないことがあります。この場合、考えられるソリューションは、プロセッサが必要とする I/O の数を減らすため、I/O エクスパンダとして CPLD を使用することです。

図 3 は、プロセッサとキーパッドの間にある CPLD が、一方はキーパッドの行と列、もう一方はプロセッサの GPIO にインターフェイスしている状態です。このように、CPLD を使うと、8 × 8 のキーパッドに必要なプロセッサの GPIO ポートは、4 × 4 のキーパッドの場合と同数に抑えることができます (実際には 1 個少ない)。CPLD を使わなければ、このプロセッサには 7 個でなく 16 個の GPIO ポートが必要になるのです。

スキャンとエンコード

CPLD は、プロセッサが必要とする GPIO の数を減らすだけ

でなく、状態の変化を知るため行をスキャンし列を監視するといった、いくつかのプロセッサ機能をオフロードできます。ユーザーがどれかキーを押すと、CPLD はスキャンを中止して、ただちにエンコード ワードを生成し、そのワードをプロセッサに送ります。プロセッサは、このエンコード ワードを通して、どのキーが押されたかを知ります。押されたボタンをプロセッサに伝えるのにエンコード ワードが使われるため、プロセッサが必要とする I/O の数が少なくてすむわけです。

図 3 に示す例では、エンコード ワードを表現するために 6 ビットが使われています。6 ビットで 26、つまりそれぞれ異なるキーを表す 64 個の値になります。ただし、そのうちの 1 個は、何もキーが押されていない状態を表す値です。したがって、GPIO を追加しない限り、この例では 63 個のキーしか表現できないことになります。

キーパッドのスキャンは CPLD によって行われるため、プロセッサがスキャンする必要はありませんが、GPIO 上の変化を監視する必要があります。どのキーが押されたかの情報は 6 ビット ワードでエンコードされるため、単にその情報を推定する必要がないというだけです。

スイッチ デバウンスの除去も必要です。デバウンスは、CPLD とプロセッサ、どちらのデバイスにリソースがあるかに応じて、いずれかにデザインできます。プロセッサで行うほうが、CPLD のサイズとコストを最小限に抑えることができます。

このデザイン例を要約すると、まず CPLD がキーパッドをスキャンして押されたキーを知り、そのエンコード ワードを計算してプロセッサに送り、プロセッサはそれを読み込んで解釈します。この機能はプロセッサにスキャンの負荷をかけないだけでなく、GPIO を拡張する効果もあります。

このデザインは、CoolRunner-II 32 マクロセル デバイスに非常に適しており、利用率は最大 75 % で、残りの 25 % は他の

機能に利用することができます。他にも、消費電力を削減し、CoolRunner-II の省電力機能を活用できるデザイン アイデアがあります。

CPLD のデザイン詳細

キーパッドの行をスキャンするため、バレル シフト レジスタはすべて 1 で初期化されています。ただし、1 個のビットだけはあらかじめ 0 (ゼロ) にプリセットされています。シフトレジスタの各ビットは、キーパッドの行に接続されている CPLD の出力ピンを駆動します。シフトレジスタがクロッキングされるたびに、ゼロ (0) ビットがバレル シフト内をシフトし、一度に 1 つずつ Low に駆動することで行をスキャンしていきます。キーパッドの列は CPLD への入力となりますが、各入力は内部のプルアップ抵抗で High にプルアップされています。

キーが押されていないときは、CPLD へのすべての列入力は受動的にロジック High に引き上げられます。列の入力のすべては AND 回路を介して出力されるため、出力のロジック 1 は、キーが押されていないことを意味します。

AND の出力はシフト レジスタへのイネーブル信号として使われます。キーが押されると、行と列の接続が確立され、キーが押されている列がそのキーに関連する行により Low に駆動されます。キーが押されると AND の出力は Low になり、シフト レジスタはディスエーブルされます。

この時点で、シフト レジスタは押されているキーの行を Low に駆動しており、そのキーの列も Low です。この情報を関連付けるため、行のビット(シフト レジスタの出力)用に 1 つ、また列の入力用に 1 つという、2 つのエンコーダが使われます。2 つのエンコーダの出力をグループ化してエンコード ワードを形成し、それをプロセッサに送るわけです。図 4 に、この動作のブロック図を示します。

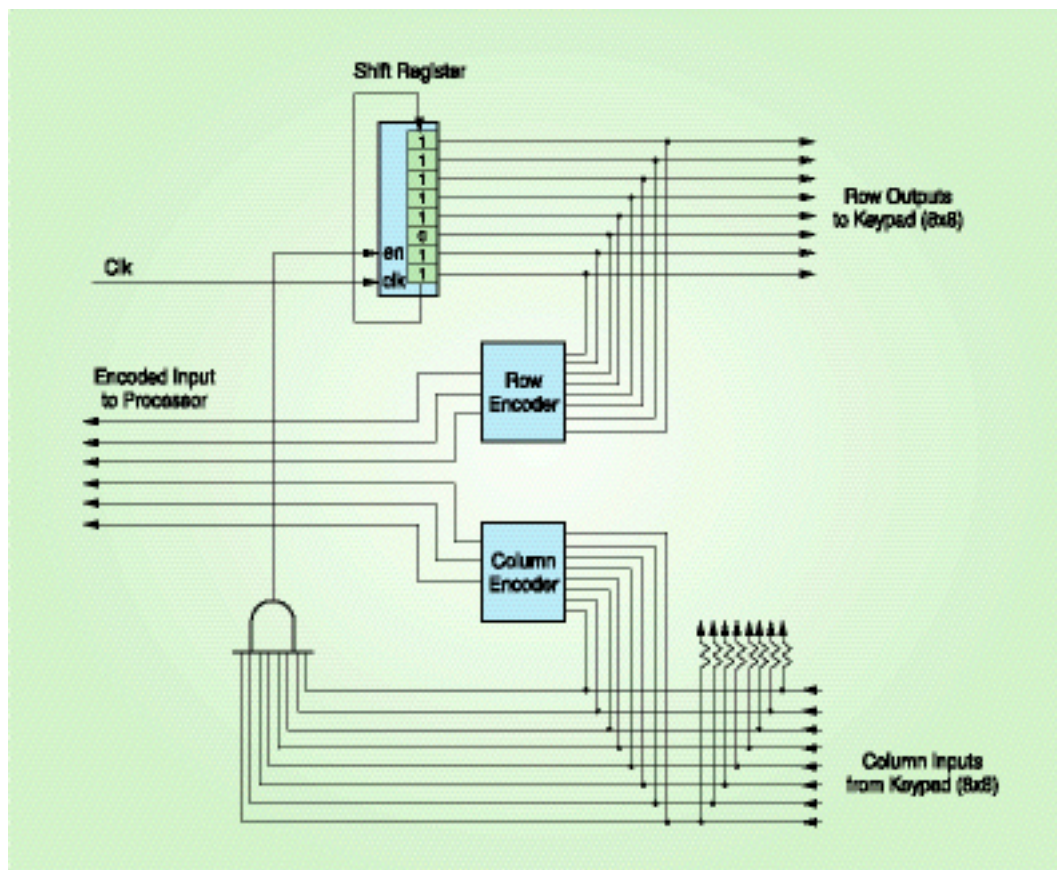
結論

ザイリンクス CoolRunner-II CPLD を使うことで、デザインの柔軟性と低消費電力が得られます。I/O 拡張の他に、CPLD には電圧変換、I/O 規格の変換、入力ヒステリシスといった他の「グルー」機能を吸収することができます。

CPLD は、プログラマブルで、同じデバイスを異なるキーパッドや異なる製品に使用できるため、生産量の増加とコスト削減につながります。また、プログラムの変更を行えることから設計の変更を後で加えることが可能で、リスクの低減にもなります。

このアプリケーションの詳細は、ザイリンクスのアプリケーション ノート「Implementing Keypad Scanners with CoolRunner-II」(<http://www.xilinx.co.jp/bvdocs/appnotes/xapp512.pdf>)をご覧ください。また、ザイリンクスの CPLD については、<http://www.xilinx.co.jp/cpld/> をご覧ください。

図 4 キーパッドの入出力との関係を示すブロック図

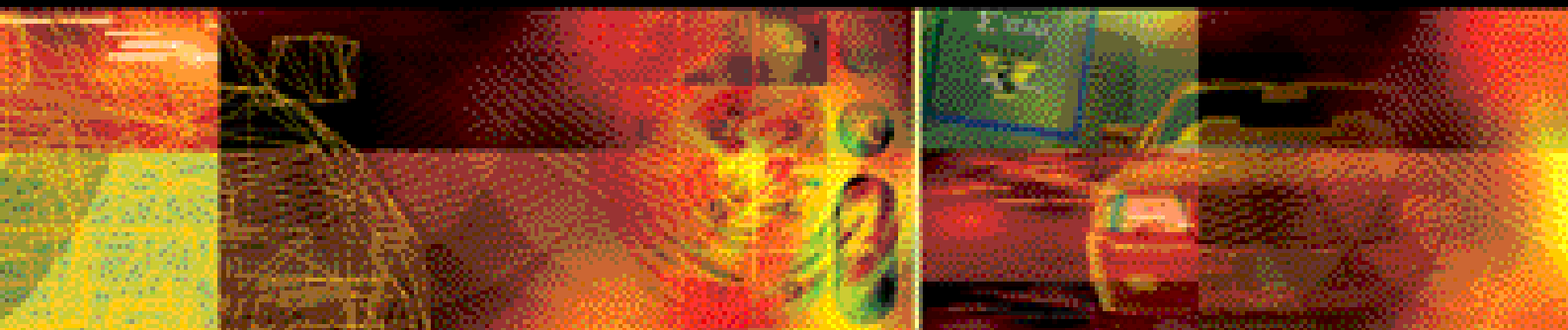




Smart Telematics Systems from Xilinx and Microsoft Corp.'s Automotive Business Unit

ザイリンクス & マイクロソフト社が推進する 高性能テレマティクス システム

デザインの柔軟性とパフォーマンスにより採用された
Spartan-3 FPGA



Rodney Stewart
System Architect, Automotive
Xilinx, Inc.
rodney.stewart@xilinx.com

David Vornholt
Strategic Relationships Manager
Xilinx, Inc.
david.vornholt@xilinx.com

米運輸省の調査によると、世界中の人々が自動車で移動する時間は週に 5 億時間を超えるそうです。これだけ長い時間を過ごすのですから、車内で娯楽を楽しんだり、家族や恋人と電話で話したり、ふだん仕事場でやる作業を車内でもできればと思うのは当然です。

その中でも、今最も求められているのが車内での通信手段です。これは携帯電話の利用を考えればわかります。大渋滞に巻き込まれたり、道を間違えたり、燃料切れになったりすると、目的の時間に遅刻してしまいます。

では、ドライバが安全運転しながら外と通信し、目的地に時間どおりに到着するにはどうすればいいのでしょうか？ 賢い方

法は、音声コマンドで起動する通信/制御装置とインターネット接続を組み合わせることです。これを実現したのが、さまざまなモバイル デバイスをシームレスに統合し、インターネットとワイヤレス ネットワークを通して情報を配信するハブ、マイクロソフト社のテレマティクス プラットフォームです。

マイクロソフトのテレマティクス プラットフォームには、次のような機能があります。

- 最先端の高品質な音声認識/合成技術
- 渋滞回避、最新ニュースの取得、MSN Auto を通して最寄りの最安値の給油所検索（現在は米国内のみ）など、オンデマンドの Web サービス
- GPS を利用し、目的地や矢印での方向案内を設定するなど、ナビゲーションのカスタマイズ
- PDA/携帯電話と Bluetooth テクノロジーの統合。Bluetooth は、携帯電話と PDA を車載の電子システムにワイヤレス接続し、ドライバは車内のオーディオ システムを通して、声で電話の発信、受信や会議のスケジュール確認、重要データへのアクセスを実現
- 故障や保守の警告など、自動車の「健康状態」をチェックする遠隔診断。自動車の寿命を通してエンジン性能の改善に貢献

マイクロソフト社のオートモーティブ ビジネス ユニットとザイリンクスは、世界中のドライバに向けてよりシンプルで信頼性に優れ、手ごろな価格のソリューションの開発を促進するため、これらのメリットを低コストで提供するリファレンス プラットフォームを構築しました。

柔軟かつスケーラブルなプラットフォーム

従来、オートモーティブ向け電子機器の設計にあたっては、自動車メーカーの要望をもとに、その自動車にしか利用できない極めて限定的で柔軟性のないソリューションを開発していました。テレマティクスとインフォテインメントにより、今日の自動車業界は「コネクテッド カー」向けにいかにして製品やシステムをデザインするか、再考を迫られています。

テレマティクスのようなアプリケーションにおいて自動車にコンシューマの世界が融合されたことで、これまで開発スピードが遅く、保守的でコスト優先だった自動車業界は、「コンシューマ向けの開発」という新たな考え方を迫られました。消費者は常に次の新製品を期待するため、この新たなニーズに応える

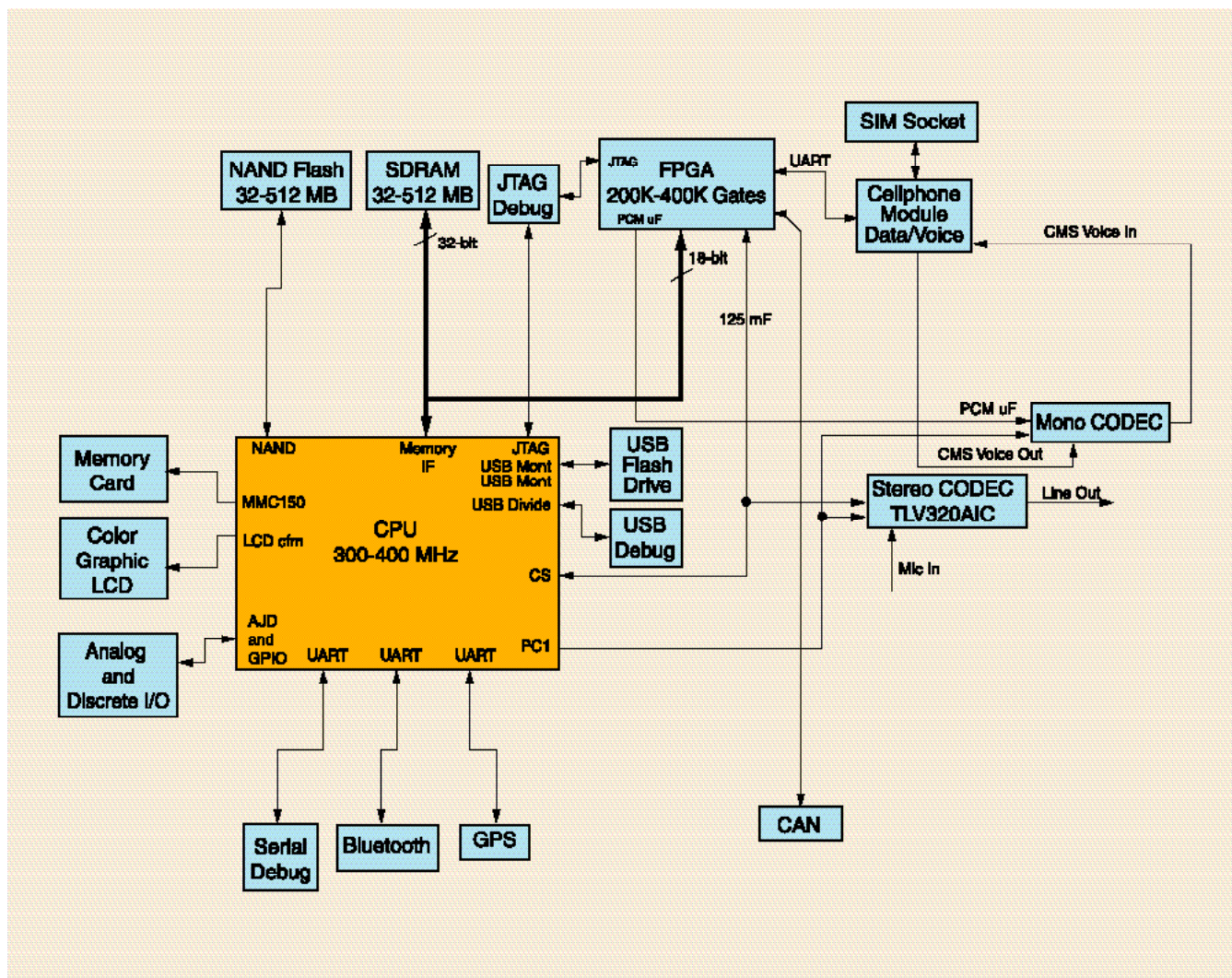
には、急速な技術革新が必要です。

そのためには、現在のアプリケーションだけでなく、将来の未知の機能にも対応し得る柔軟なアーキテクチャを開発し、設計手法を根本から変える必要があります。つまり、自動車向けの一般的な電子機器を設計するときのような、数年がかりの開発・検証サイクルとは矛盾するのです。今プラットフォームを開発して 2、3 年後に発表するクルマに搭載する場合、製品開発段階はもとより、発売後も予期せぬ変化に対応できるよう、十分なシステム リソースを用意しておくことが大切です。

どのプラットフォームにも言えることですが、基本的なシステムから高性能でハイエンドなテレマティクス システムにいたるまで、アーキテクチャの採用の鍵は、その柔軟性と拡張性にあります。このことを念頭に置き、マイクロソフトはカスタマイズ可能でスケーラブルな、真に自動車業界の標準となり得るテレマティクス プラットフォームを開発しました。

このプラットフォームは、ARM 9 ベースのマイクロコントローラを採用し、32 MB Flash RAM/32 MB DRAM 以上のメモリをサポートするとともに、GPS、Bluetooth と GSM 電話モジュールを搭載しています。車外との接続には、CAN ネット

図 1 マイクロソフト社のテレマティクスプラットフォーム ハードウェア アーキテクチャ



トワーク インターフェイスと、LED ドライバやボタン入力などのファンクション用にプロテクト付きのアナログおよびデジタル I/O を備えています。図 1 に、プラットフォームの基本的なアーキテクチャを示します。

マイクロソフトは、柔軟性と高度な統合性を兼ね備える FPGA テクノロジーの優位性を採用しました。このプラットフォームには、GSM 電話インターフェイス、車載インターフェイス (CAN コントローラと K ライン)、先進の音声信号コンディショニング/ルーティングなど、それぞれ別々の目的を実現するため、Spartan™-3 XC3S400 FPGA が採用されました (図 2)。

FPGA が提供する高度な統合性には、1 個のデバイスに複数のバス、インターフェイス、クロックを収容できるため、EMI (電磁干渉) を伴うデザインをより管理しやすくなるというメリットがあります。さらに、コンポーネントとボード スペースの低減は、自動車設計における重要な要素である生産コストの削減と製造品質の向上につながります。

マイクロソフトは、自動車開発の特性と、多彩な車載インターフェイスの存在を理解したうえで、システムの基本的なアーキテクチャと性能に影響を及ぼすことなくバックエンドの車載

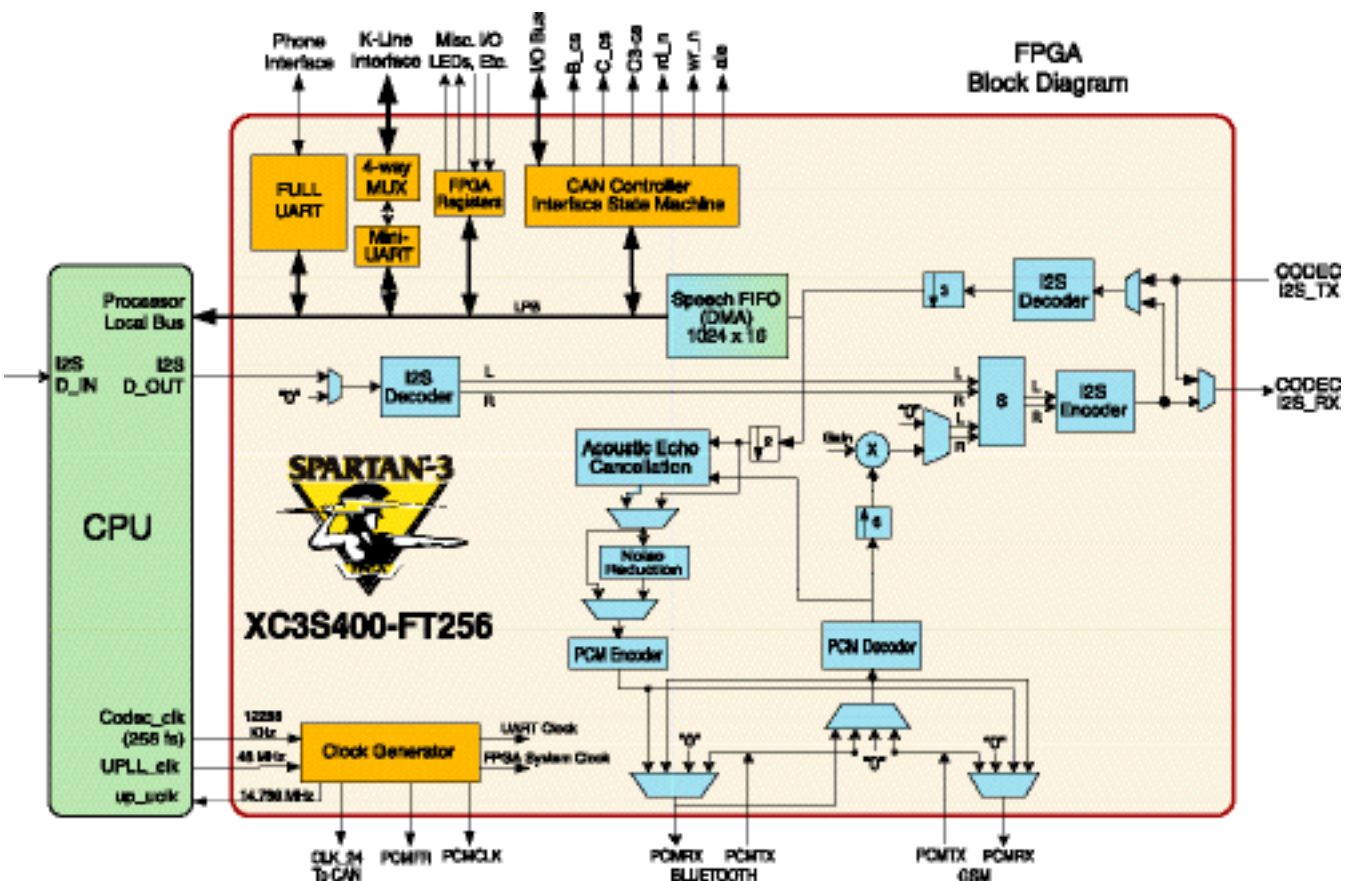
インターフェイスを素早く変更できるよう、あえて柔軟なソリューションをデザインしたのです。たとえば、将来的にはこの FPGA ソリューションを、MOST や IDB-1394 といったオートモーティブ バスや、あるいは別のデジタル車載ネットワークなどのエンド アプリケーションのニーズに合わせて適応できるようになるでしょう。

音声認識システム

マイクロソフトのテレマティクス プラットフォームの核となるのが、音声認識 (VR) システムです。ほとんどの VR システムでは、音声信号パスがアナログ バイアス/フィルタリングやデジタル化、デジタル フィルタリングの処理が施され、信号が最終的に音声処理のために VR エンジンに送られます。

このパスには、エレクトロニック プラットフォームのボード上と、電子装置に入る前の車内環境の両方において、システムに無用のノイズが入り込む機会が複数あります。マイクの位置と種類がそのアプリケーションと環境に正しく適合しているか、製品開発者と自動車メーカーの双方が確認する必要があるの

図 2 ザイリンクス Spartan-3 FPGA によるデザイン



です。

VR エンジンがクリーンかつ安定した音声信号を受信するのが理想ですが、自動車はその性格上動くものであるため、許容できるレベルの音声認識を実装するのはそう簡単なことではありません。言語やアクセント、性別などにより、音声認識はもともと難しいうえ、走行速度や窓の位置（開閉）、ロード ノイズ、天候状況（雨や風）といった要因がいつその妨げになります。こうした追加要因から、信号が VR エンジンに送られる前に、非常に適応性の高いデジタル フィルタリング アルゴリズムを使ってプリ コンディショニングすることの重要性が高まりました。

マイクロソフトは、この信号プリ コンディショニングをハードウェアに実装し、ザイリンクスのパラレル DSP プロセッシングを使用することにしました。104 個のエンベデッド 18 ビット乗算器を搭載する Spartan-3 FPGA は、低コストなデバイスに MAC エンジン、分散型算術 FIR フィルタ、フル パラレル FIR フィルタといったコンパクトな DSP 構造を実装するのに理想的です。

また、マイクロソフトはプロセッサに多大な負荷がかかるソフトウェア フィルタリングをハードウェアに実装しました。もちろん、こうしたプリプロセスは専用の DSP チップなど、ASSP でも可能です。しかし、その場合、プラットフォームの他の部分における高度な統合の利点が失われてしまいます。

テレマティクスと VR を組み合わせることで、ユーザーと環境の特徴（言語：英語、アクセント：スコットランド訛り、性別：女性）に応じて適応、アップグレード可能な VR エンジンと DSP フィルタを実装できるのです。

特にインフォテインメントのような自動車向け製品を設計する際には、将来いつでもアップグレードできるよう、バンド幅に十分余裕を持たせることが重要であり、これは FPGA にも適用されます。自動車の OEM メーカーにとって、将来のプラットフォームには、柔軟かつスケーラブルなファームウェアを実現できるアーキテクチャがますます必要になってくるのです。

今のところマイクロソフトのプラットフォームにはまだ実装されていませんが、システム コプロセッサとして働くソフト プロセッサは容易に追加できます。マイクロソフトのデザインで主プロセッサから DSP プロセッシングがオフロードされたように、主システム プロセッサから処理の負荷を一部取り除くため、ザイリンクスの 32 ビット MicroBlaze™ ソフト プロセッサや 8 ビット PicoBlaze™ マイクロコントローラなどのエンベデッド プロセッサを使用することもできます。

オートモーティブ アプリケーション向け FPGA

近年、自動車の電子機器は、従来の車体制御やエンジン マネジメントだけでなく、ドライバ支援システムやテレマティクス アプリケーションという新しい分野でも飛躍的な成長を遂げてきました。IEEE が最近公表した調査によると、自動車の電子機器は年間 16 % の割合で増えており、2005 年には中型車のコストのうち電子機器が占める割合は 25 % になると予想され

ています。

テレマティクス システムは、短い Time-to-Market（市場投入までの期間）と Time-in-Market（市場で販売される期間）、規格とプロトコルの頻繁な移り変わりという、まさにコンシューマ向け製品と同じような特徴を示しています。こうした観点から見ると、エンジニアのデザイン アプローチはもちろんのこと、デザインを迅速に作成、再利用し、将来的なアップグレードにもすぐ対応できるようにするため、ハードウェアの選択は大きく変わろうとしています。

現在、FPGA テクノロジはこれらの要件を解決できます。ザイリンクスは、今後も Xilinx Automotive (XA) ファミリーを通して、テレマティクスがカー インフォテインメントといったアプリケーションをサポートし続けていきます。同ファミリーの特徴は次のとおりです。

- ・ 温度範囲の拡張 — 最高 125 °C
- ・ 自動車業界の生産部品承認プロセス (PPAP: Production Part Approval Process) をフル サポート
- ・ 業界公認の AEC-Q100 デバイス認定フロー
- ・ 自動車業界の国際的な品質管理規格 ISO TS 16949 と、RoHS 指令に準拠するため鉛フリー パッケージに対応

ザイリンクスの Spartan ファミリー FPGA をベースとするこれらのデバイスは、ロジックあたりの低コスト（システム ゲート）や I/O あたりの低コスト、さらにシングル デバイス上の複数の I/O 規格や高速 DSP 用のエンベデッド乗算器などの先進機能が要求されるデジタル デザインに理想的です。

結論

マイクロソフト オートモーティブ ビジネス ユニットやザイリンクス オートモーティブといった協業各社の努力により、マイクロソフトのテレマティクス プラットフォームのビジョンが今まさに現実になろうとしています。現在、大手自動車メーカー各社は、次の特徴を持つプラットフォームに向けて、着々と主要テクノロジーを採用しているところです。

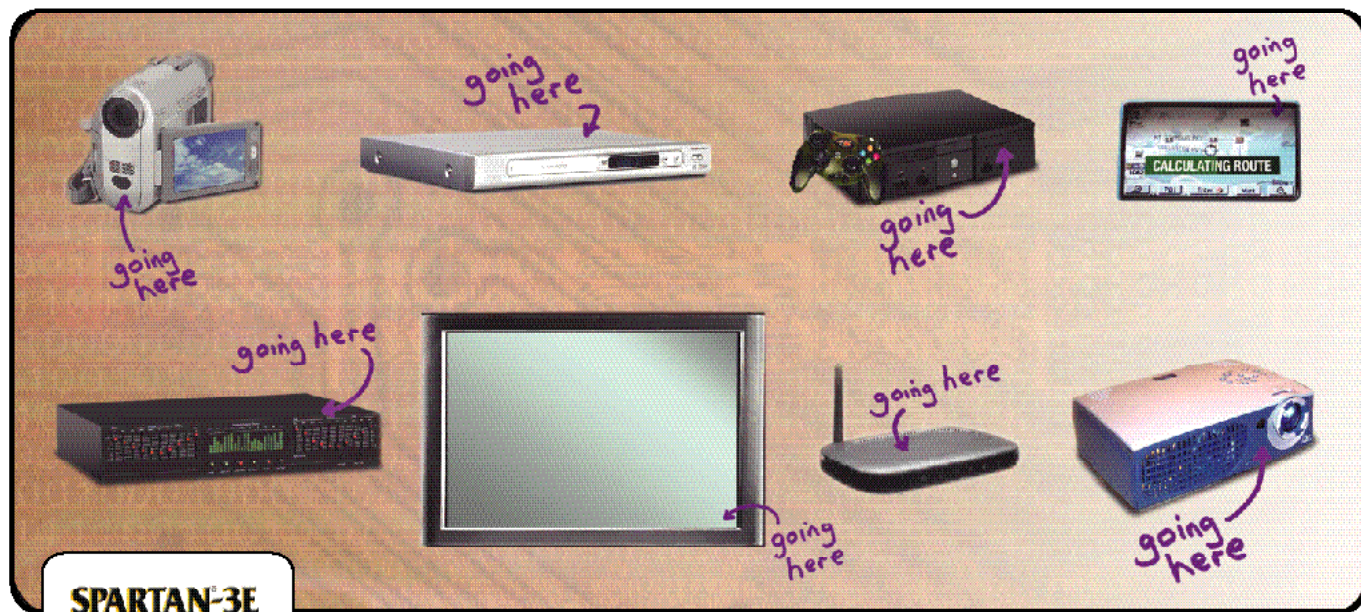
- ・ 高価値で手ごろな価格のテレマティクス ソリューション
- ・ ワイヤレス ネットワークを介した信頼性の高いコネクティビティ
- ・ 高品質な音声認識
- ・ アプリケーション開発者のためのサポート範囲の広い OS
- ・ 低コストのハードウェア

開発者たちは、これらのプラットフォームを使ってエンドユーザーにさらに価値の高いものを提供するため、継続的に投資し続けていくことになるのです。

詳細は、<http://www.microsoft.com/automotive/windowsautomotive/about.mspx/> と <http://www.xilinx.co.jp/automotive/> をご覧ください。



Spartan-3Eは、 これらの量産向けアプリケーションに 採用が決定されています。



SPARTAN-3E



デジタルコンシューマ アプリケーションに最適!

高性能90nmプロセステクノロジーを採用した最新のSpartan-3Eデバイスは、ロジック重視のデザインに最適化された製品で、特にデジタルコンシューマ アプリケーションに最適な機能を備えています。従来比30%以上安価になったSpartan-3E FPGAは、ASICを置き換えるだけでなく、柔軟でかつタイムトゥマーケットを実現できます。

業界初の10万システムゲートFPGAがわずか2ドル*で手に入ります

Spartan-3E FPGAは、10万システムゲートの柔軟なアーキテクチャに加え、低コスト、デジタル信号処理を実現可能な高性能エンベデッド乗算器、72KビットのブロックRAM、柔軟なクロックシステムを構築することのできるDCM、デジタルクロックマネージャ機能に加え、mini-LVDS、RSDSインターフェースを含むさまざまなI/O標準をサポートしています。また、Spartan-3Eファミリでは、最大160万システムゲートまでの高集積度をサポート。これらは、既の実証済みの90nmプロセスを使用しており、これら業界最先端クラスの製造技術とコスト効率に優れたアーキテクチャの融合で今までにないプライスポイントと最高の価値をお届けします。



すぐに使いこなせる
ソフトウェア

→ **MAKE IT YOUR ASIC**

*XC3S100Eの2006年7月以降、50万個での価格です。
また、出荷時期および納期は変更される場合があります。



The Programmable Logic Company™

www.xilinx.co.jp/spartan3e



Design Techniques to Reduce Power Consumption 消費電力を低減させる 設計テクニック

複数のデザイン テクニックと ISE 電力解析ツールを使った消費電力の制御

Arthur Yang
Sr. Product Applications Engineer
Xilinx, Inc.
arthur.yang@xilinx.com

FPGA は、世代交代のたびにますます高速化、高密度化、大型化しています。それに伴って消費電力が増えないようにするにはどうすればいいのでしょうか？ どのデバイスを選ぶかというわかりやすい意思決定から、使用する周波数に基づいてステート マシンの値を選ぶという細かな判断にいたるまで、システムの消費電力に影響を与える設計上の選択肢は多数あります。

本稿で論じるデザイン テクニックがなぜ消費電力の節約になるのかを理解するため、消費電力とは何かを、ここで簡単に触れておきます。

消費電力は、ダイナミック消費電力とスタティック消費電力の大きく 2 つに分けることができます。ダイナミック消費電力は、デバイス内で容量負荷を充放電するために必要な電力であり、周波数、電圧、負荷に大きく依存します。これら 3 種類の変数は、それぞれ設計者が決めることができます。ダイナミック消費電力は、次の式で計算されます。

$$\text{ダイナミック消費電力} = \text{キャパシタンス} \times \text{電圧}^2 \times \text{周波数}$$

スタティック消費電力は、デバイス内のすべてのトランジスタのリーク電流（ソース/ドレイン間とゲートのリーク。静止電流と総称することもある）によって発生する電力と、その他の所定電力を合計したものです。リーク電流はジャンクション温度とトランジスタのサイズに大きく左右されます。詳細については、ザイリンクスのホワイトペーパー「Static Power and the Importance of Realistic Junction Temperature Analysis」(<http://www.xilinx.co.jp/bvdocs/whitepapers/wp221.pdf>) をご覧ください。

所要定電力には、プルアップ抵抗など、終端処理による電流

リークが含まれます。リークに対してできることは限られていますが、定電力は抑制できます。

消費電力については早めに考えておく

消費電力の決定が最も有効なのは、デザインの初期段階です。クロックに BUFGMUX を挿入するのに比べれば、最初に部品を決めておくことのほうが消費電力にはるかに大きな意味があるのです。設計の際、消費電力について考えるのに時期尚早ということはありません。

デザインに最適な部品を選ぶ

静止電力は部品によって異なります。一般に、デバイスのプロセス技術が微細なほど、リーク電流は高くなります。しかし、プロセス技術はどれも同じように作られているわけではありません。

たとえば、一口に 90 nm FPGA テクノロジーと言っても、Virtex™-4 のデバイスと他社のテクノロジーでは静止電力に大きな開きがあります。詳細は、ザイリンクスのホワイトペーパー「Power vs. Performance: The 90 nm Inflection Point」([http://www.xilinx.co.jp/bvdocs/whitepapers/wp223](http://www.xilinx.co.jp/bvdocs/whitepapers/wp223.pdf)) pdf をご覧ください。

プロセス技術の微細化に伴い静止電力が増すとはいえ、プロセスが微細なほど電圧と容量は低下するため、ダイナミック電力も低くなります。ですから、デザインによって、スタンバイ（静止）電力とダイナミック電力のどちらに重きを置くか考える必要があります。

ザイリンクスのデバイスはどれも、汎用のスライス ロジックセルに加え、専用のロジックを備えています。これらの専用のロジックには、ブロック RAM、18 X 18 乗算器、DSP 48 ブロック、SRL16 などがあります。スライス ベースのロジックではなく、かならず等価の専用のロジックを使用して

クロックの管理

デザイン内で電力を消費する信号のうち、クロックは最も消費電力の多い部品です。クロック周波数は 100 MHz かもしれませんが、このクロックから派生する信号の動作周波数は主クロック周波数の数分の 1 ということが珍しくありません (一般に 12 % ~ 15 %)。加えて、クロックはファンアウトも高いため、低電力化にはクロックをしっかりと検討する必要があります。

デザインの一部を非アクティブ状態にできるなら、クロックツリーがトグルするのを避けるには、クロック イネーブルでなく BUFGMUX を使うことを考えるとよいでしょう。クロック イネーブルはレジスタが不必要にトグルするのを回避しますが、それでもクロック ツリーはトグルし、電力を消費します。とはいえ、何もしないよりはクロック イネーブルを使うほうがまだましです。

使用する象限をできるだけ少なくするには、クロックを分離してください。未使用のクロック ツリーの象限はトグルしないため、クロック ネットにかかる負荷が軽減されます。慎重なフロアプランニングにより、実際のデザインに影響を与えることなくこの目標を達成できるはずです。

パワー解析ツール

ザイリンクスは、Web Power Tool というプリ インプリメンテーション ツールと、XPower というポスト インプリメンテーション ツールの 2 種類の消費電力見積もりツールを提供しています。

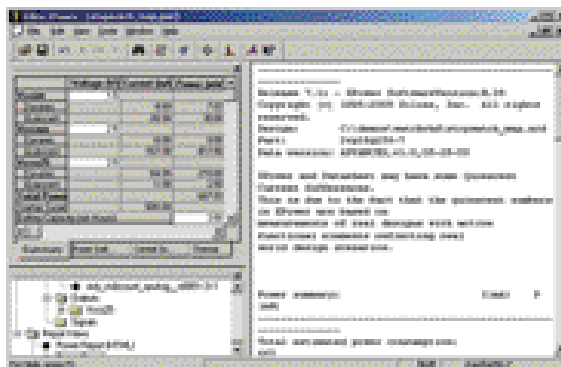
Web Power Tool (<http://www.xilinx.co.jp/powe/>)は、ロジック使用量の概算に基づいて消費電力を見積もるツールです。これを使うことにより、実際のデザイン ファイルがなくても、デザイン ユーティリゼーションの見積もりだけで消費電力を評価できます。

XPower は、実際のデバイス使用量を解析するポスト インプリメンテーション ツールであり、実際のポスト フィット シミュレーション データ (VCD ファイル形式) と併用することで、正確な消費電力データを得られます。XPower を使うと、シリコンにまったく触れずに、設計変更が消費電力全体に及ぼす影響を解析できます。

Web ベースのパワー解析ツール

Web ベースの電力見積もりは、デザイン フローの初期段階にデバイスの消費電力を把握するための最も手っ取り早い方法です。このツールは四半期ごとにリリースされ、常に最新の情報を提供しています。インストールやダウンロードは必要なく、Web ブラウザさえあれば利用できます。デザイン パラメータを指定し、デザインの設定値を保存してロードすることができるため、繰り返し使用する際にデザイン パラメータを再入力する必要がありません。デザイン ビヘイビアの見積もりとターゲット デバイスさえあればすぐに始められるのです。

図 2 XPower 画面



XPower – 個別デザインの統合型消費電力解析ツール

Xilinx ISE™ のあらゆるデザイン ツールに無償で提供されている XPower を使うと、デザイン ベースで消費電力の詳細な見積もりが得られます。XPower は、マッピング後、もしくは配置・配線後のデザインに基づいてデバイスの消費電力を見積もります。消費電力の見積もりにあたっては、既に成熟し量産中の FPGA および CPLD の場合、10 % 以下のデザインスイートの平均エラー率で計算します。また、デザイン ファイルとデバイス データを考慮し、デバイスの消費電力見積もりを、個別のデザイン情報にカスタマイズした高い精度でレポートします。XPower は ISE ソフトウェアに直接統合されており、ネットの消費電力を階層構造で詳しく表示するとともに、詳細なサマリ レポートや、初めて使うユーザーが簡単に利用できるよう消費電力ウィザードも備えています。XPower はシミュレートしたデザイン アクティビティ データ を取り込み、GUI とバッチ モードの両方で実行されます (図 2)。

XPower は、デザイン内の各ネットおよびロジック エLEMENT を考慮します。ISE デザイン ファイルは正確なリソース使用量を提供し、XPower は配線情報をキャラクタライズされたキャパシタンス データと相互参照します。その後、物理リソースがキャラクタライズされてキャパシタンスが計算されます。デザインのキャラクタライズは、より新しいデバイスに対しても最も正確な結果を提供できるよう、継続的に行われています。XPower は、ネット トグル レートと出力負荷を使用し、さらに消費電力とジャンクション温度を計算します。ネットの消費電力のデータを個別に表示することもできます。

結論

より安価でシンプルな熱管理や、最先端 FPGA の所要電力増大に伴う電源ニーズが高まったことから、低電力化を目指したデザイン コンセプトがますます重視されるようになりました。ザイリンクスの最新デバイス、Virtex-4 FPGA は、スタティク消費電力を予測以下に抑え 90 nm の高性能を実現しています。ザイリンクスの消費電力見積もりツールを使い、低電力設計を心がけることで、消費電力の目標を今まで以上に容易に達成できることでしょう。



Performance vs. Power: Getting the Best of Both Worlds

パフォーマンスが 消費電力が： その最適解を得る

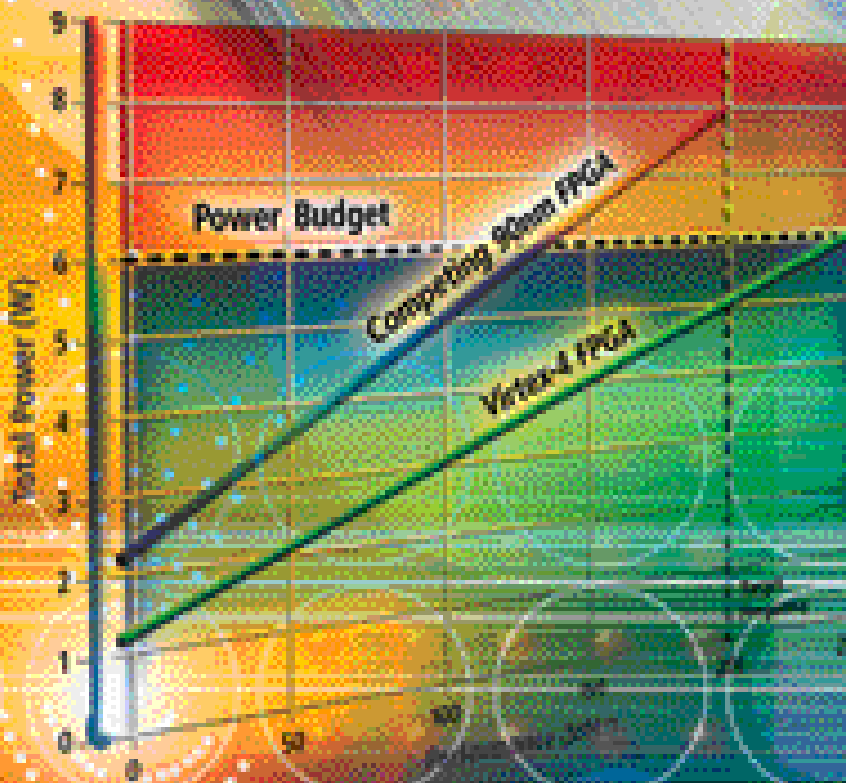
90 nm 特有の変曲点を初めて克服した
ザイリンクスの取り組みと成果

Anil Telikepalli

Marketing Manager, Virtex Solutions

Xilinx, Inc.

anil.telikepalli@xilinx.com



高性能な 90 nm FPGA のうち、どれが最小の消費電力なのか、白熱した議論が交わされています。つまり半導体業界は、性能と電力/熱バジェットが競合する 90 nm プロセスの臨界変曲点を通過したのです。ユーザーはできるだけ高い性能を求めますが、FPGA の選択にあたっては、どのデバイスが最も低い消費電力なのかで判断するのが最近の傾向です。

過度な消費電力は、さまざまな点からコスト高になります。ヒートシンクからファン、先進の熱交換器にいたるまで特別なデザインを必要とするだけでなく、動作についても慎重に考慮する必要があります。また、より大容量の電源装置を使うコストも考えなくてはなりません。

おそらく最も重大な問題は、過度な電力が信頼性に及ぼす影響でしょう。ジャンクション温度が上がるにつれて、トランジスタの消費電力が増えるため、デバイスの温度はいっそう上昇します。この現象を放っておくと、熱暴走を起こしかねません。常時稼働し続けるシステムでジャンクション温度が 85 ~ 100 °C を超える場合、デバイスの信頼性が損なわれます。

幸いにも、ザイリンクス社は 3 年以上前、Spartan™-3 FPGA (90 nm プロセス採用の初の ザイリンクス FPGA ファミリ)を開発していた初期段階に、この 90 nm 特有の変曲点を示す最初の兆候に遭遇しました。ザイリンクスは、90 nm 特有の電力の問題を解決するため、ただちに新しい手法の開発に着手しました。その甲斐あって、2004 年 9 月にさらに高性能な Virtex™-4 ファミリを発表し、ようやくこの新しいファミリが 90 nm FPGA の分野で最高の性能と最小の消費電力を同時に達成できたという確信を得ました。

FPGA における消費電力の低減

消費電力は、大きくスタティック消費電力とダイナミック消費

電力の 2 つに分けられ、それぞれが特有の問題を招きます。90 nm FPGA の場合、最も問題となるのはスタティック消費電力のほうです。

スタティック消費電力

スタティック消費電力は、デバイスが何も機能を実行してない状態でも消費されるスタンバイ電力です。原因は FPGA 内部のトランジスタを流れるリーク電流にあります。プロセスが新しくなるたびにトランジスタは小型化され、それに伴ってリーク電流は増えていきます。この原理こそ、90 nm プロセスが大きな変曲点となる主な理由の 1 つです。

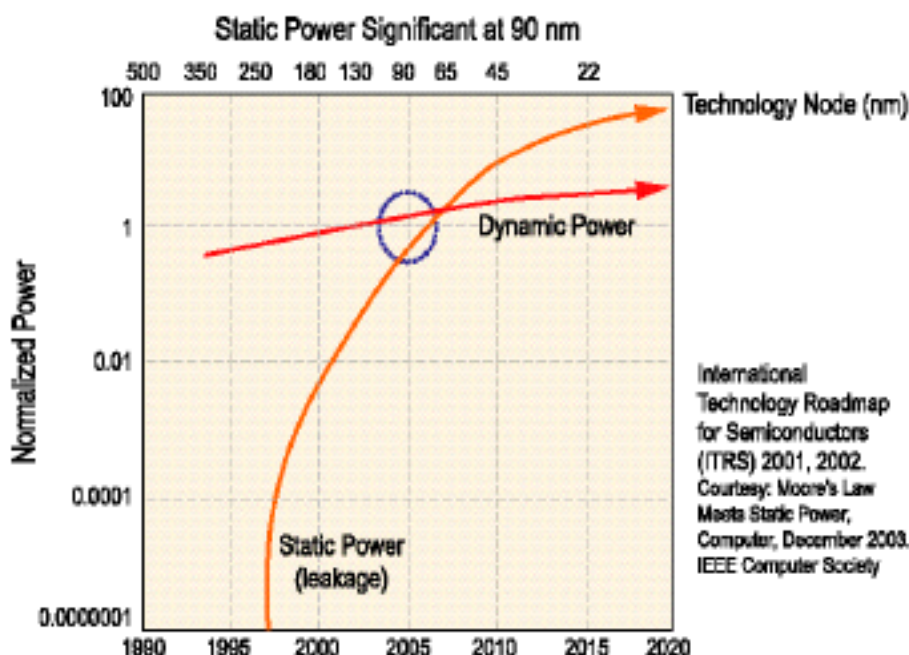
また、FPGA の総消費電力においても、初めてスタティック消費電力がダイナミック消費電力を超える最大の要因になるうとしています。プロセスが微細化すれば、コア電圧は低下し、寄生キャパシタンスも低下します。この結果、プロセスが新しくなるにつれて周波数は高くなるものの、ダイナミック消費電力の増加率は落ちます。対照的に、0.25 μm (注: 0.25 μm = 250 nm で、s は複数形を意味する s です)以下のスタティック消費電力は、新しいプロセスに移行するたびに指数的に増加してきました。

ここで、FPGA にとって変曲点が極めてクリティカルな要因になるわけですが、ザイリンクスはこの分野でも競合他社を大きくリードしています。トランジスタが小さいほどスピードは高速ですが、リーク電流は増えます。ゲートの酸化膜を厚くすればリーク電流は減りますが、性能は悪くなります。しかし、ASIC や ASSP、マイクロプロセッサとは異なり、ザイリンクスの FPGA ではすべてのトランジスタが最高速度でスイッチングする必要はありません。プログラマブル ロジックに使われるコンフィギュレーション メモリセルは膨大な数のトランジ

スタから構成され、プログラマブル インターコネクト(相互接続)配線の実装にはパストランジスタが使われます。コンフィギュレーション メモリセルは高速である必要はなく、プログラマブル インターコネクトトランジスタが高速でなければならないのはソース/ドレイン間だけであり、ゲートの制御下ではその必要はありません。ザイリンクスは、これらの要因を考慮して、性能を犠牲にすることなくリーク電流を低減するように、ゲート酸化膜の厚さを選択的に増やすことに成功しました。

Virtex-4 FPGA は、スタティック消費電力の問題を解決するため、トリプル酸化膜技術という新しいプロセス アプローチ

図 1 スタティック消費電力とダイナミック消費電力が交差する90 nmの変曲点



チを採用しています。この第 3 のゲート酸化膜は非常に薄いものの、これを用いたトランジスタのリーク電流は、Virtex-II Pro FPGA や Virtex-4 FPGA の他の部分に使われている標準的な薄膜酸化膜トランジスタよりはるかに低くなっています。

さらに、ザイリンクスは、I/O、コンフィギュレーション メモリ、インターコネクト パストランジスタ、ロジックおよびインターコネクト バッファにおけるリーク電流と性能のバランスをとるため、 V_T を含む他の多くのトランジスタ パラメータを最適化しました。図 2 に示すとおり、Virtex-4 FPGA のスタティック消費電力は、旧型の 130 nm Virtex-II Pro FPGA より 50 % も低減しています。より微細なプロセス ノードに移行してスタティック消費電力が低減したのは、FPGA の歴史上初めてのことでしょう。

ダイナミック消費電力

FPGA のダイナミック消費電力は、コア電圧 (V)、周波数 (f)、寄生キャパシタンス (C) という 3 要素からなります。加えて、ダイナミック消費電力はデータ トグル レート (k) に比例します。幸いなことに、コア電圧とキャパシタンスはプロセス ノードが新しくなるたびに減少するため、ダイナミック消費電力は低下します。逆に、デザインの動作周波数が高まるとダイナミック消費電力も増大します。ダイナミック消費電力は、次の式で求められます。

$$P = k \times C \times V^2 \times f$$

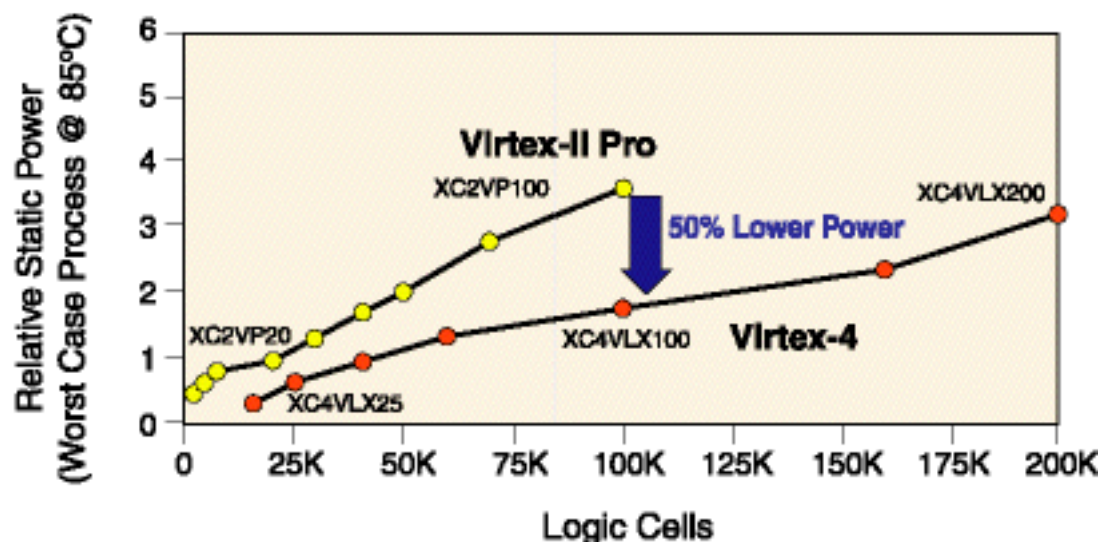
FPGA でダイナミック消費電力を低減させる主な方法の 1 つに、デザインにおけるエンベデッド機能の使い方があります。エンベデッド機能は、コンフィギュレーション可能なロジックブロックやプログラマブルなインターコネクトとしてではなく、ハードワイヤ機能として実装することにより、スタティック消

費電力、ダイナミック消費電力とも低く抑えることができます。ハード ロジックが使うトランジスタの数は、プログラマブルロジックよりはるかに少ないからです。さらに、ハードワイヤ化されたエンベデッド機能にはプログラマブル インターコネクトトランジスタが存在しないことも、ダイナミック消費電力のいっそうの低減につながります。

これらのハード IP コアは、占有スペースがずっと小さいうえ、はるかに高いパフォーマンスを提供し、同じ機能をソフト IP バージョンで実装する場合と比べて消費電力が 80 ~ 95 % も低減されます。また、ハード IP コアをプログラマブルかつパラメータ化可能にすることで、FPGA 本来の柔軟性を損なうことはありません。ザイリンクスが Virtex-4 FPGA にハード IP コアとして提供する機能は次のとおりです。

- 450 MHz PowerPC™ プロセッサ：
すべてのマイクロプロセッサとエンベデッド プロセッシング アプリケーションに対応するハードウェア アクセラレーション用の APU (補助プロセッシング ユニット) インターフェイスを搭載
- 500 MHz XtremeDSP™ スライス：
単純な算術演算機能やフィルタ機能から、複雑な高性能 DSP 機能をサポート
- 500 MHz デジタル クロック マネージャ (DCM) とフェーズ マッチ クロック ディバイダ (PMCD)：
クロック合成、クロック管理、フェーズの一致をサポート
- ChipSync™ ブロック：
メモリ、ネットワーキング、テレコム アプリケーションにおけるソース同期インターフェイスを簡素化するため、すべての I/O にビルトインされた SERDES とデータ アライメ

図 2 トリプル酸化膜技術の採用がトレンドを反転：Virtex-4 デバイスは、旧型の 130 nm 技術製品よりも低いスタティック消費電力を実現



ント機能を搭載

- RocketIO™ トランシーバ(622 Mbps ~ 10.3125 Gbps) :
物理コーディング サブレイヤ (PCS) と物理メディア アタッチメント(PMA) を内蔵
- トライモード イーサネット MAC (10/100/1000 Mbps) :
RocketIO トランシーバと直接インターフェイスが可能
- 分散型 RAM と 18 Kb ブロック RAM 内蔵の Smart RAM メモリ :
各ブロック RAM に、RAM を FIFO に変換するためのビルトイン FIFO ロジックと、エラー訂正コード (ECC) 回路を搭載

このような一般的に使われるブロックをハード IP に移すことの利点に加えて、ザイリンクスの Advanced Silicon Modular Block (ASMBL) アーキテクチャが Virtex-4 の消費電力に与える利点も見逃せません。3 種類の Virtex-4 プラットフォーム (LX、FX、SX) は、それぞれロジック、エンベデッド プロセッシング、シグナル プロセッシングという特定のアプリケーションドメインの要件を満たすため、ロジックセル、メモリ、I/O、DSP、プロセッサの標準比率は、そのドメイン向けに最適化されてきました。この結果、Virtex-4 デバイスは、ドメインごとに最適化した消費電力を提供する初の FPGA となっているのです。

エンドマーケットの電力要件

トリプル酸化膜技術でスタティック消費電力を、エンベデッド ハード IP を使ってダイナミック消費電力を大幅削減したのはよいとして、それがユーザーのデザインにどういう意味を持つのかとお考えかもしれません。物事の本質を理解するには、

最も単純な例が一番役立つことがあります。他のエンベデッド IP をいっさい考慮に入れずに、Virtex-4 デバイスとこれと同等の集積度を持つ競合他社のデバイスに、同じ数の汎用ロジックとメモリを使用したところ、Virtex-4 FPGA は消費電力が 1~5 W も節約されました(図 3)。では、これは実際のアプリケーションで、電力とコストの節約にどう結びつくのでしょうか？

電力バジェット

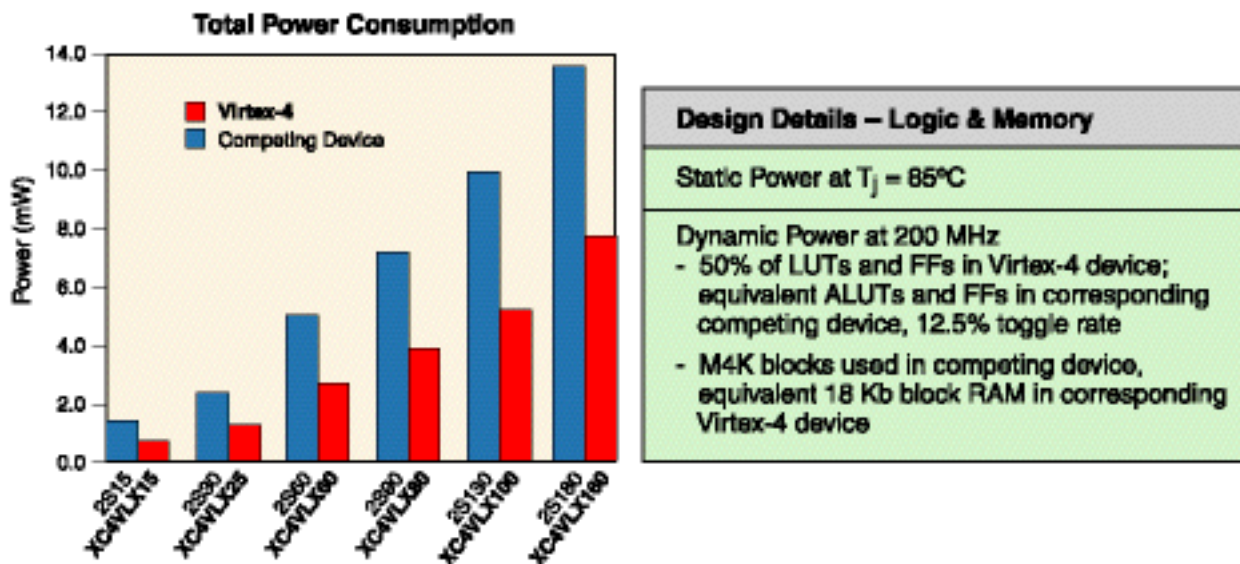
ほとんどの製品は、規格やコスト目標、信頼性の要件により、かならず電力バジェットが存在します。消費電力と温度には相関関係があるため、動作温度の目標を達成することも同じく重要です。システム設計者は、各ボードとそのボードに使われているデバイスに対して、システム レベルで一定の電力バジェットを持っています。また、有線/無線ネットワーク、ストレージ/サーバー、自動車、航空宇宙/防衛など、高性能な FPGA を使用している業界も厳しい電力バジェットに従います。ここで、電力バジェットが極めて重要な意味を持つ、いくつかのアプリケーションを取り上げていきます。

有線ネットワーク : メトロ アグリゲーション

メトロ アグリゲーションとは、メトロポリタン エリア ネットワーク (MAN) 内のセントラル オフィス (CO) にアクセス回線を集中化することです。各 CO 内の装置は常時稼働する必要があるため、運用コスト、電源および空調装置の有効容量に多大な負荷がかかります。手段はどうであれ、装置のベンダがシステムの総消費電力を削減できれば、サービス プロバイダは大きなメリットを享受できるのです。

通常、メトロ アグリゲーション機器のラック内のカードに対する電力バジェットは、平均して 20 ~ 30 W です。これら

図 3 Virtex-4 のデザインによって FPGA あたり 1 ~ 5 W を削減



製品の受容性、信頼性、利益性は、性能と同等、あるいはそれ以上に電力効率に依存するのです

ードに使われている FPGA はそれぞれ 4 ~ 5 W を消費し、多くのデザインは複数の FPGA を使用します。

たとえば、マルチ サービスを提供するプラットフォームのラインカードと FPGA の電力バジェットは次のようになります。

- 12 ポート DS3 カード：30 W ; FPGA = 4 ~ 5 W
- 4 ポート OC-12 カード：28 W ; FPGA = 4 ~ 5 W
- 12 ポート 10/100 Base-T カード：50 W ;
FPGA = 4 ~ 5 W
- 32 ポート T1/E1 カード：9 W ; FPGA = 2 ~ 3 W

これらのアプリケーションに Virtex-4 FPGA を使うことで、サービス プロバイダの運用コストは劇的に削減されます。Virtex-4 FPGA 1 個あたり、競合他社の 90 nm FPGA と比べて 1 ~ 5 W の電力節約になるのです。

有線ネットワーク：メトロ アクセス

CO に展開するメトロ アグリゲーション装置とは異なり、メトロ アクセス装置はネットワークのエッジにあります。この装置は、エア フローが制限され、空調設備がほとんど存在しない屋外に設置されます。システム例としては、パッシブ オプティカル ネットワーク(PON)、デジタル ループ キャリア(DLC)、ケーブル モデム終端システム(CMTS)があります。これらのシステムは 85 °C を超える温度で常時稼働し、ジャンクション温度は 100 °C に達します。温度の上昇に伴い、トランジスタのリーク電流、ひいてはスタティック消費電力も増大します。したがって、この分野の装置ベンダは、信頼性を確保するため、電力バジェットに厳しい制約を受けます(10 ~ 12 W/カード、4 ~ 8 W/FPGA)。

こうしたアプリケーションは消費電力に非常に敏感で、デザインが実際に使い物になるかどうか、わずか 0.5 W の節約にかかっていることさえあります。Virtex-4 デバイスは FPGA あたり 1 ~ 5 W を節約するため、装置ベンダとサービス プロバイダの双方にとって多大な利点があります。

無線基地局

開設コストが安く手軽に使えることから、携帯電話の市場は固定電話ネットワークを上回る成長を遂げてきました。この分野でも、サービス プロバイダは基地局が設置されている屋外環境からくる信頼性の問題の軽減、そして運用コストの低減という両方の面から、Virtex-4 FPGA の低消費電力にいに価値があるか、実際に数字で測ることができます。

35,000 台の無線基地局を結ぶ典型的なネットワークを敷

設したサービス プロバイダの場合、電気料金だけで年間 100 万ドル以上の節約になります。次の電力バジェットを考えてください。

- 16 ライン カード/基地局：1 FPGA/ライン カード
- 電力バジェット/ライン カード = 20 W
- FPGA 電力バジェット = 6 W

かなり控えめな見積もりですが、Virtex-4 FPGA を使って 2 W の消費電力を節約したと仮定した場合、サービス プロバイダは基地局当たり 32 W の電力を節約でき、ネットワーク全体で 1.12 MW の節約になります。KWh あたりの電気料金が 10 セントだとすると、ネットワーク内の 35,000 台の基地局全体で年間約 100 万ドルの節約になるのです。

基地局あたり 32 W を削減した場合、冷却装置のコスト、バッテリー バックアップのコスト、電源および電力管理コストに対する資本支出が削減されるわけですから、サービス プロバイダの収支決算も改善されるのです。

結論

いかにして最小コストで最大性能を実現するかが、FPGA の技術革新における最大の目玉になりました。今日、ユーザーは消費電力を最小限に抑えたいと望んでいます。電力の節約は、技術的成本や財政的成本を問わず、あらゆるバジェットを削減します。製品の受容性、信頼性、利益性は、性能と同等、あるいはそれ以上に電力効率に依存するのです。Virtex-4 FPGA は、堅牢で高性能な機能に加えて、消費電力の点でも大変優れています。

とはいえ、FPGA 市場における競争は、90 nm デバイスで幕を閉じるわけではありません。今後、65 nm ノード、そしてさらに微細なノードに移行していけば、新たに面白いダイナミクスが起こるでしょう。幸いにも、ザイリンクスのトリプル酸化膜技術は、新しいプロセスへの移行に伴いスムーズにスケールリングできるというメリットがあります。

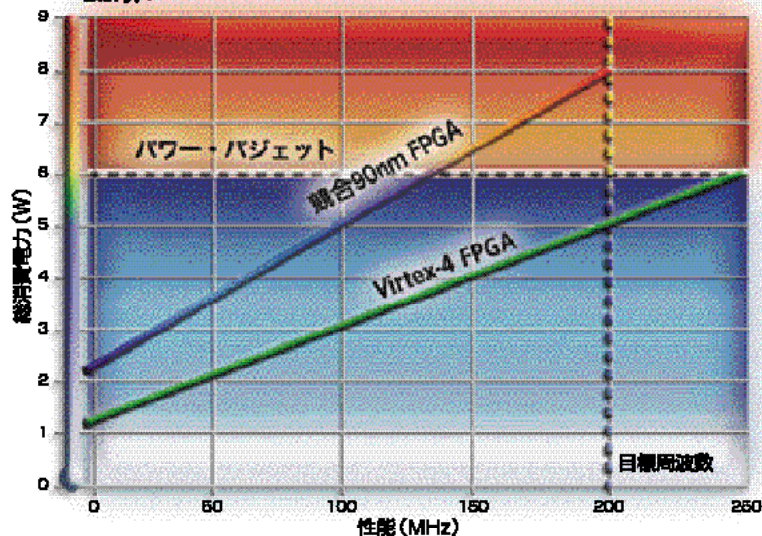
必要に応じてハード IP を埋め込むことの価値は、業界全体が認めるところです。ザイリンクスは、ソリューション全体の柔軟性を高め、さらなる性能向上と消費電力の低減を目指して、プログラマブル ロジックに適正量のプログラマブル エンベデッド IP を組み込みました。近い将来、ユーザーは、性能と消費電力のバランスが最も秀でているプラットフォーム FPGA だけを採用することでしょう。

電力バジェット、トレーニング、チュートリアル、ホワイトペーパー、消費電力解析/最適化ツールの詳細は、<http://www.xilinx.co.jp/virtex4/lowpower/> をご覧ください。



低消費電力のXILINX FPGAで、 あなたのシステムをいつもクールに

設計例：



設計例： LX200 vs. 2580, 目標周波数=200 MHz, フーストケース・プロセス
25K LUT, 80Kフリップフロップ, 1MbitオンチップRAM, 64 DSPブロック, 128 2.5V IO
サイリックス ツール v4.0および統合ツールv2.1にて
高集積度デバイスでは最大で7ワットの消費電力削減が可能



FPGA1個あたり1~5ワットも
消費電力が削減できるのは
Virtex-4だけ

現実的な動作温度 ($T_j=85^{\circ}\text{C}$) での仕様をご自身でお確かめください。
トータルな消費電力削減に関して、Virtex-4に優るFPGAはありません。

- スタティク消費電力を73%削減
- ダイナミック消費電力を最大86%削減

トリプル酸化膜テクノロジーとエンベデッドIP

90nmテクノロジー・ノードにおけるシステム・レベル設計者が直面する次の問題は消費電力です。デバイスによっては、漏れ電流、スタティク消費電力の急増、さらには熱暴走を引き起こしてしまうもあり、このため、Virtex-4 FPGAはトリプル酸化膜テクノロジー、エンベデッドIP、省電力設定回路を使用して設計されました。これにより予定消費電力内で目標とする性能が達成できます。

www.xilinx.co.jp/virtex4/lowpower

最少のシステムコストで、最高のシステム・パフォーマンスを。



The Programmable Logic Company™



Merging CPLD Features into Handheld Applications

ハンドヘルド アプリケーションへの CPLD の適用

ザイリンクス CPLDを量産・低消費電力の
民生用アプリケーションに採用

Steve Prokosch
High Volume Marketing Manager
Xilinx, Inc.
steve.prokosch@xilinx.com



CPLD は、電圧変換、I/O 規格の変換、HSTL および SSTL メモリ インターフェイス、クロッキング機能、より高性能なフリップフロップなど、一般的な機能をいっそう統合化しています。

ザイリンクスの CPLD は、長年にわたり、制御ロジックやステート マシン、シンプルなシステム インテグレーションに最適なソリューションでしたが、消費電力が高いという難点から消費者向けの携帯機器には使われませんでした。1990 年代後期、CPLD は最初の CoolRunner™ 製品ファミリーにより低消費電力の分野に足を踏み入れました。しかし、多くの量産アプリケーションには簡単に手の届く価格ではありませんでした。今日、ムーアの法則による着実な進化を経て、その価格はディスプレイ ロジック デバイスと十分競合できるレベルまで下がっています。多彩なロジック ファンクションをシングル パッケージに簡単にインプリメントできるのです。さらに、ボード スペースを節約し、リプログラムablの利点を活かして Time-to-Market の期間を短縮することができます。

設計者の中には、CPLD は単なるロジックとフリップフロップにすぎないと、いまだに信じている人がいるかもしれませんが、今日の最新の CPLD では、1 つのデバイスからより多くの機能を実現できるのです。たとえば、レガシー製品とのミスマッチを解消するためシステムを統合する必要がありますが、毎年新しいディスプレイ ロジック デバイスが発表されています。CPLD は、複数の I/O バンクを低コストで提供することで、このニッチ市場を埋める役割を果たします。

また、CPLD は、電圧変換、I/O 規格の変換、HSTL および SSTL メモリ インターフェイス、クロッキング機能、より高性能なフリップフロップなど、一般的な機能をさらに統合しています。

これだけでありません。CoolRunner-II CPLD には、デザインのリードバックやコピーを防止する精巧なスキームが組み込まれています。スパイ行為を働こうにも、カプセル化されたデザイン ファイルを開くのに膨大な作業が必要となり、それにかかるコストと労力を考えるととても見合わないでしょう。

CoolRunner-II CPLD は低消費電力機能を備えており、デザイン全体の消費電力をダイナミックに低減させることができます。低消費電力機能には、入力のゲート化、クロック周波数

のスケールリング(分周と逡倍)、入力ヒステリシスがあります。これらの機能を使うことで、競合他社との差別化を実現しながら、低消費電力化や低コスト化を図り、いち早い市場投入を可能にします。

サクセス ストーリ

台湾の HTC 社は、OEM/ODM の顧客向けにモバイルコンピューティング/通信ソリューションを設計、製造する専門メーカー

です。消費者向け製品の設計を得意とする同社は、今日、ワイヤレス ハンドセット市場にも事業拡大しています。図 1 は、HTC 社の GSM/GPRS Magician というハンドセット プラットフォームで、カメラ、タッチ スクリーン、SD/MMC メモリ拡張スロット、マイク、オーディオジャック、同期通信用のミニ USB コネクタを内蔵しています。

同社の製品には、スマートフォン、スマートミュージック フォン、PDA フォン、コンパクト PDA があります。マイクロソフト社が Windows CE デザインのプラットフォーム開発パートナーとして HTC を選んだのは、こうした強い特化性が評価されたためです。

THT Business Research 社によると、HTC はポケット PC をベースとする PDA の世界最大メーカーであり、OEM 供給全体の 48 % を占めます。HTC は G3 ハンドセットに巨大な市場を見込み、2001 年第 2 四半期に製造プロセスを開始しました。現在、同社は Windows CE ベースの PDA フォンの 90 % 以上を製造しており、年間

350 万台以上を出荷し続けています。

HTC は、常に高いコスト競争力を維持するとともに、他のハンドセット メーカーのように需要が出るまでただ待つのではなく、積極的に新機能を追加していくことで、一部の大手有名メーカーから市場シェアを奪ってきました。各種市場調査によると、ローエンドのハンドセットは生産台数が減少しているものの、多機能型のハンドセットは今も伸び続けています(表 1 を参照)。2006 年までには、ハンドセットの新規販売台数のう



図 1 台湾 HTC 社の携帯端末 Magician

表 1 全世界の携帯電話のタイプ別出荷台数(2001～2008年) 出典：IDC 単位：%

	2001	2002	2003	2004	2005	2006	2007	2008
Entry Level	97.7	95.4	84.2	68.6	52.5	40.1	29.3	20.8
Midrange Feature	1.9	3.1	10.6	21.1	28.0	31.4	32.2	34.3
High End	0.1	0.6	3.4	7.3	12.6	19.0	26.4	30.3
Converged Mobile Device	0.4	0.8	1.8	3.0	7.0	9.5	12.0	14.6

ちエントリーレベルのカテゴリに属するものは40%程度になり、その後は毎年10%ずつ減少していくと予想されています。こうした市場の激変に備え、ハンドセットメーカーは消費者トレンドに乗り遅れないよう、早急に製品の仕様変更に取り組んでいく必要があるのです。

また、Windows CE搭載のハンドセットが大躍進を遂げています。調査会社ガートナー社が2005年4月に発表したレポートによると、Windows CEベースのPDAが出荷台数トップとなっています(表2)。

HTCは3年前、市場における自社製品の競争力を高めるため、代替ロジックデバイスの研究調査に着手しました。評価の基準は、価格、機能、使いやすさ、消費電力でした。また、インテグレーションの問題と、市場動向に沿って新製品を開発し続けるにはどうすればよいかも検討しました。

ディスプレイやタッチスクリーン、メモリ、ワイヤレス通信の技術革新に乗り遅れないよう、HTCは柔軟かつ多機能なソリューションを必要としていたのです。こうした製品は6～8カ月単位で移り変わるため、最もコストの安い最新コンポーネントを採用するために再設計を余儀なくされることが少なくありません。HTCは、サポート面と機能対コスト競争力の両面から複数のベンダを候補に挙げました。また、価格設定やロードマップ、技術支援の面で信頼できそうなベンダに特に注意を払いながら、パートナーシップについても検討しました。

当然の選択

HTCは、サプライヤとコンポーネントを多面的に評価した結果、低消費電力CPLD製品の最適なベンダとしてザイリンクスを選びました。価格だけでなく、製品や納入実績、品質、特別なニーズへの対応における業界の評判を聞き、競合他社と比較した結果、ザイリンクスが評価されました。HTCによると、一番重視した基準はデバイスの機能でした。基本消費電力が電力バジェット以上だったことに加え、低消費電力の拡張機能も部品の選定に大きな役割を果たしました。HTCのピーター・チョウ社長は次のように述べています。「ザイリンクスの製品には高性能な機能が満載されていました。最先端のテクノロジー、完全にプログラマブルなシステムデザイン、フルサポートのテクニカルサービスは、HTCの成功に不可欠だったのです」。

他のサプライヤと比べて、消費電力は基本的には同じでした。

表 2 全世界のOS別PDAの出荷台数

Operating System	Units Shipped	Market Share
Windows CE	1,759,497	45.5%
Palm OS	1,301,740	33.7%
RIM	698,000	18.1%
Linux	24,300	0.6%
Others	82,000	2.1%

決定要因となったのは、他の競合製品にはない低消費電力機能を組み込むことができる点でした。低消費電力化の機能が統合されていることで、HTCの設計者はバッテリーの寿命延長と電話機能の拡充に向け独自の方法を採用することができました。また、クロック機能を使うことで、外部オシレータを除去し、コスト削減を実現しました。これらのクロッキング機能は、総消費電力を低減し、バッテリーの寿命を延ばすうえでも役立ちました。設計者は、この機能1つだけで、追加の電力を使うことなくボードスペースを削減したのです。

バッテリーの寿命延長に貢献したもう1つの低消費電力機能は、入力ゲートの静電化です。この手法は他の製品でも使われていますが、ザイリンクスはこれをいち早くCPLDに採用しました。この機能は、回路をスタンバイ、つまり静止状態にパワーダウンさせることで、デザイン全体のダイナミック総消費電力を低減させるのに有効でした。一部の時間しか使われない回路をオフにすることで、同様の機能を搭載する競合製品よりバッテリー寿命をはるかに延長できるのです。

HTCのデザインには、デバイスのインテグレーションも大きな利点でした。特定の回路に対する電圧レベルをシフトするために1個のデバイスを使うことで、ディスクリートデバイスは不要になりました。小型のチップスケールパッケージは、ディスクリートデバイスと比べてボードスペースの節約になりました。また、高集積化の目標に直面したときには、各種機能を1つのリプログラマブルデバイスに一体化することでレイアウトの問題を解消できたそうです。複数の信号パスを1つのPCBレイヤに高密度化することで、レイアウトがはるかに容易になりました。

低消費電力 CPLD は HTC に高く評価され、
同社は CoolRunner-II 製品が提供する機能を活用しています。

WebPACK による設計期間の短縮

HTC のように特殊な機能を利用するには、できるだけ余計な労力をかけずにそれらをインプリメントできることが必要です。クロック周波数のスケーリングや、信号のゲート化、ヒステリシスといった低消費電力機能は使いやすく、ISE™ Web PACK™ ソフトウェアにサンプル コードが同梱されています。このため、HTC はクリティカル タイミングと消費電力の目標がすべて達成されるという確信を持って、より短期間でデザインを変更できたのです。

ザイリンクスの WebPACK ソフトウェアの一番の利点は、これら機能が無償で使えることです。VHDL や Verilog などのハイレベル言語でのデザインが可能で、ファンクションとタイミングの検証ツールもスムーズに実行することができます。そのうえ、XPower 消費電力解析ツールを使えば、動作の各ポイントでどれくらいの電力が消費されるか、高い信頼性で見積もることができます。この使いやすいソフトウェア ツールは、HTC の設計者に、シミュレーションを通して得られる結果は

実際のデザインで測定した結果と同じだという自信を与えました。HTC は、何を求めるかを事前に正確に知っておくことで、デザインにそのほとんどを取り入れることができました。

結論

低消費電力 CPLD は HTC に高く評価され、同社は CoolRunner-II 製品が提供する機能を活用して、ミッドレンジからハイエンドのハンドセット市場で競合他社より大きなシェアを獲得しています。ユーザー数が増え、機能もさらに充実する中、HTC は新興市場でトップの座を保ち、成長を維持しています。ザイリンクスのシリコン製品とソフトウェア ツールを使うことで、HTC は今後も革新的なソリューションをデザインし続け、スケジュールどおりに市場に投入していくことでしょう。

サイリンクスの CPLD ファミリの詳細は、<http://www.xilinx.co.jp/cpld/index.htm> をご覧ください。HTC のハンドセット製品については、<http://www.htc.com.tw/> をご覧ください。



顧客の経営者や役員は、ERPシステムを使いこなしていません。それはERPの導入以上に重要なポイントだといえます。サイバーナクスやL&L Groupの「ERP Implementation 2014」は、過去3年間のERP導入率、成功と失敗の事例をまとめた。そしてユーザーがERPシステムを成功させるためのポイントも示されています。

आचार्य श्री अरवि प्रसाद झा, एमए-बि०ए

[illegible]

NAME	AGE
JOHN DOE	25
JANE SMITH	30
BOB JONES	22
ALICE BROWN	28
CHARLIE WHITE	35
DAVID GREEN	20
EVE BLACK	27
FRANK GRAY	32
GRACE HILL	24
HELEN WATSON	29
IRVING KING	31
JACK LEE	26
JILL SCOTT	23
JOHN TAYLOR	33
JANE ANDERSON	21
BOB HARRIS	28
ALICE MARTIN	34
CHARLIE THOMAS	25
DAVID ROSS	30
EVE KING	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32
HELEN FOSTER	24
IRVING PETERSON	29
JACK COLE	31
JILL PERKINS	26
JOHN LONG	33
JANE HAYES	21
BOB STEVENSON	28
ALICE TOLSON	34
CHARLIE SIMS	25
DAVID STEVENSON	30
EVE HARRIS	22
FRANK WOOD	27
GRACE BAKER	32

Case Study - MPT-Crutch

DuoRunner はデザインソフトの性能を
いかに向上させるかというテーマでサ
ブティンが議論されています。開発者
ボード、プログラマーグループ、
デザイナーズ、ソフトウェアインテグ
レーターすべてが意見を述べます。

今までの中核企業は、経営者やディレクターが個別に現場に立寄り、会って話をし、聞いて、指導し、調整し、やり直してあげていくことで、また、この「*the way of the zen*」のやり方で、ディレクターとして、*the way of the zen*で二重の意味で仕事をこなす。この「*the way of the zen*」が、すべてディレクターやデザイナーが、



How to Detect Potential Memory Problems Early in FPGA Designs

FPGA デザインにおけるメモリ問題の早期発見法

Micron Technology 社が実践する FPGA メモリのための最新システム互換テスト

Larry French
FAE Manager
Micron Semiconductor Products, Inc.
lfrench@micron.com

設計者のみなさんは、ボードのシミュレーションやプロトタイプ構築・テストに膨大な時間を費やしていることと思います。プロトタイプで実施するテストは、製造段階、もしくは使用現場で起こりうる問題を効果的に検出してくれるため非常に重要です。

FPGA システムに組み込まれている DRAM やその他のメモリは、FPGA 単体とは異なるテスト手法を必要とします。メモリのデザイン、テスト、検証の各ツールを正しく選択することにより、設計期間が短縮され、潜在的な問題を検出できる確率も高まります。本稿では、メモリを使用するザイリンクス FPGA デザインを徹底的にデバッグするためのベスト プラクティスについて論じていきます。

メモリのデザイン、テスト、検証の各ツール

デザインをシミュレーション、デバッグするためのツールは多数あります。表 1 は、メモリのデザインに使用する 5 つの基本ツールのリストです。ここでは熱シミュレーション ツールを

除いて、デザインの機能性と堅牢性を検証するためのツールに絞っています。表 2 は、各ツールがどんな場合に使用すると最も効果的かを示しています。

本稿では、表 2 で示す製品開発の 5 つのフェーズを取り上げていきます。

- ・フェーズ 1 : デザイン (ハードウェアなし、シミュレーションのみ)
- ・フェーズ 2 : アルファ プロトタイプ (製造前にデザインとハードウェアを変更する可能性あり)
- ・フェーズ 3 : ベータ プロトタイプ (ほぼ製造品に近いシステム)
- ・フェーズ 4 : 製造中
- ・フェーズ 5 : 製造後 (メモリ アップグレードや現場での交換)

シグナル インテグリティ テストの必要性

シグナル インテグリティ (SI) は万能薬ではないので、適切に使用すべきです。SI を使いすぎる傾向がありますが、過度な使用は控えるべきでしょう。最初期またはアルファ プロトタイプでの SI は、システムに次のようなメモリ障害が多数発生しないかを検証するための重要ツールと考えるべきです。

表 1 メモリデザイン、テスト、検証のためのツール類

ツール	例
電氣的シミュレーション	SPICE または IBIS
ビヘイビアシミュレーション	Verilog または VHDL
シグナルインテグリティ	オシロスコープとプローブ (より正確なシグナルを 捕らえるには 混在モードで使用)
マージンテスト	さまざまな温度や 電圧下での ガードバンドテストや マージンテスト
互換性テスト	ソフトウェアの機能テスト または システムのリブートテスト

- ・リンギングとオーバーシュート/アンダーシュート
- ・タイミング違反
 - セットアップとホールド タイム
 - スルー レート(駆動の弱い、もしくは強い信号)
 - セットアップ/ホールド タイム(データ、クロック、コントロール)
 - クロック デューティ サイクルと差動クロックの交差 (CK/CK#)
 - バスの競合

一方、ボードの信号に変更がない限り、ベータ プロトタイプ
のフェーズでの SI は重用ではありません。各信号はアルファ

プロトタイプで検証されるからです。しかし、信号が変更され
た場合でも、SI を使い、変更されたネットに SI の問題がないか
の確認はできます。製造段階での SI テストを実施する必要性は、
絶対ではありませんが、めったにないと思っていいでしょう。

一般に SI がテストに過剰に使われる理由は、エンジニアとい
うのはオシロスコープを見たり、システムがテストされたこと
を証明するドキュメントとしてキャプチャや写真を使ったりす
ることに安心感を覚えるからです(図 1)。しかし、Micron
Technology 社は、過去の幅広い経験を通して、障害を突き
止めるにはもっと有効なツールがあることを知っています。事
実、今までの経験上、SI ですべての種類のシステム障害を検
出するのは不可能です。

SI テストの限界

SI テストには、根本的な限界が多数あります。中でも筆頭
に挙げられるのが、メモリ業界がファインピッチ ボールグリッ
ド アレイ (FBGA) のパッケージに移行したことです。このパ
ッケージの下ではプローブする方法がないため、貴重なボード
スペースにプローブ ピンの配置場所を確保しない限り、SI
はほとんど不可能です。

Micron Technology 社は、メモリ クオリフィケーション
テストの実施中、SI ラボで数十万のスコープ撮影を行いました。
このデータから、システムの問題はマージンおよび互換性
テストを使うことにより最も簡単に発見できるという結論に達
しました。SI はアルファ プロトタイプ フェーズでは有用で
すが、ベータ プロトタイプと製造フェーズでは上記のテストを
使うべきです。

SI テストの結果、他にも次のことがわかりました。

- ・ SI は、メモリやシステム レベルの診断で識別されなかった問
題を 1 つも発見できませんでした。つまり、SI は他のテ
ストと同じ障害を発見しただけで、機能的にはマージン テ
ストとソフトウェア テストの重複にすぎません。

表 2 各デザインフェーズで使用するメモリ機能検証ツール

ツール	デザイン	アルファ プロトサイズ	ベータ プロトサイズ	製造中	製造後
電氣的シミュレーション	必要不可欠	非常に有用	一部有用	ほとんど不要	無 用
ビヘイビア シミュレーション	必要不可欠	非常に有用	一部有用	ほとんど不要	無 用
シグナル インテグリティ	不 可	クリティカル	一部有用	ほとんど不要	無 用
マージン テスト	不 可	必要不可欠	必要不可欠	必要不可欠	必要不可欠
互換性	不 可	有 用	必要不可欠	必要不可欠	必要不可欠

- SI は時間がかかります。64 ビットもしくは 72 ビットのデータバスをプローブしてスコープ撮影をすると、相当な時間を覚悟しなければなりません。
- SI は高価な機器を使います。正確なスコープ撮影を収集するには、高額なオシロスコープとプローブが必要になります。
- SI は貴重なエンジニアリング リソースを費やします。スコープ撮影を評価するには、高レベルなエンジニアリング アナリシスが必要です。
- SI ではすべてのエラーを発見できません。SI で検出できないエラーは、マージンおよび互換性テストで発見できます。

FPGA とメモリの問題を発見するのに最適なテストは、マージン テストおよび互換性テストです。

マージン テスト

極端な温度や電圧下でシステムがどう動作するかを評価するには、マージン テストを使います。スルーレート、ドライブ能力、アクセス時間など、温度や電圧に応じて変化するシステム パラメータは多数あります。室温でシステムを検証するだけでは不十分です。Micron Technology 社は、マージン テストによって SI で検出できないシステム障害を検出できるという、もう 1 つの利点があることに気づきました。

半導体業界において、マージン テストのベスト プラクティスは 4 コーナー テストです。マージン テストの途中で発生し得る障害のほとんどは、次のいずれかです。

- コーナー 1：高電圧、高温
- コーナー 2：高電圧、低温
- コーナー 3：低電圧、高温
- コーナー 4：低電圧、低温

注意すべき点は、アルファ プロトタイプ段階ではデザインはまだ変わり続けているため、マージンはベータ プロトタイプで改善されるということです。アルファ プロトタイプでは、マージン テストを実施してもあまり意味がないということです。広範なマージン テストを行うのは、システムがほぼ製造品に近い段階に達してからが有効です。

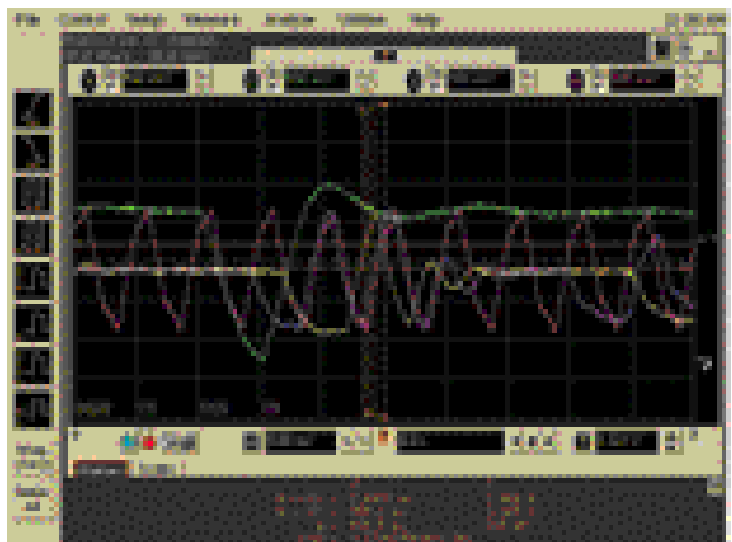
互換性テスト

互換性テストとは、システムで走らせるソフトウェアのテストです。たとえば、BIOS、システムの OS、エンド ユーザーソフトウェア、エンベデッド ソフトウェア、テスト プログラムなどです。パーソナル コンピュータ (PC) は極めてプログラマブルであるため、多種多様なソフトウェア テストを実施する必要があります。

FPGA がプロセッサのような役割を果たすエンベデッド システムでは、互換性テストにも膨大な数のテストが必要になることがあります。DRAM が FIFO やバッファなど特別な目的を持つ他のエンベデッド アプリケーションの場合、定義上ソフトウェア テストの対象は最終的なアプリケーションに限定されます。こうした種類のシステムでは、システム レベルの問題や障害を検出するには、マージン テストに加えて徹底的な互換性テストを行うのがベストです。

ザイリンクス FPGA のプログラマブルな性格から、なにか特別な FPGA メモリ テスト プログラムを想像するかもしれませんが、DRAM インターフェイスを検証するため、メモリとの間で多数のテスト ベクタ (チェッカーボード、反転) を実行する目的でのみ使われます。このテスト プログラムは、メモリ障害をまったく識別できないことがある標準的なエンベデッド プログラムとは対照的に、ビット、アドレス、行のエラー検出するために簡単に書くことができます。このプログラムはマージン テスト中にも実行できます。メモリ インターフェイスが極め

図 1 オシロスコープ画面に表示された典型的なシグナル インテグリティの様子



マージン テストおよび互換性テストは、SI などの従来メソッドよりシステム内の多くのマージナリティと問題を識別できます。

て限定的なオペレーション セットを実行するエンベデッド アプリケーションには特に有用でしょう。また、この種のテストは最終製品の広範な SI テストより価値があります。

無視してはならないテスト

この他重要なテストとして、以下が挙げられます。これらのテストを無視すると、製造段階と使用現場で検出しにくい細かな問題が断続的に発生する恐れがあります。

電源投入サイクル

メモリのテスト プランには、製造段階や使用現場で問題を招く恐れのある、必ず含めるべきものがあります。まず、電源投入サイクルのテストです。電源の投入時には、電圧の上昇や JEDEC 規格の DRAM 初期化シーケンスなど、たくさんの固有なイベントが起こります。PC をテストする際は、断続的に起こる電源投入時の問題を捉えるため、電源投入サイクルを実施するというのが業界のベスト プラクティスとされています。

電源投入サイクルには、コールド ブート サイクルとウォーム ブート サイクルの 2 種類のサイクルがあります。コールド ブートとは、システムが稼働しておらず室温状態からブートすることであり、ウォーム ブートは、システムがしばらく稼働していて、内部温度が安定している状態からブートすることです。温度に左右される問題を判別するため、両方のテストを実施する必要があります。

ロジック アナライザやミックスモード アナリシスは有用か？

表 1 にロジック アナライザが掲載されていないことにお気づきでしょう。通常、デバッグ ラボはこのツールをデザイン/デバッグ プロセスの必須部分として採用するものですが、本稿ではロジック アナライザについて特に触れていません。コストと時間を考えると、システムの障害や問題を検出する目的でこのツールを真っ先に使うことはまずないからです。とはいえ、ロジック アナライザは、問題が判明した後、その問題から根本原因を知るうえで重要なツールです。シグナル インテグリティと同様、ロジック アナライザも問題の検出後に使用します。

セルフ リフレッシュ テスト

DRAM セルでは電流リークが発生するため、正しく動作するよう頻繁にリフレッシュする必要があります。セルフ リフレッシュは、メモリが長時間使われていないときにシステム電力を節約するための非常に便利な方法です。ここで重要なのは、メモリ コントローラがセルフ リフレッシュを開始、終了する際、規格に準拠した適切なコマンドを実行することです。そうでない場合は、データが消失する恐れがあります。

電源投入サイクルと同様、セルフ リフレッシュ サイクルは有用な互換性テストです。セルフ リフレッシュの開始、終了に関わる問題が断続的に発生する場合は、サイクルを何度か繰り返すことで検出できるはずです。セルフ リフレッシュを使用しないアプリケーションの場合は、このテストは完全にスキップしてください。

継続的な品質管理

最後に、継続的な品質管理に対するテスト手法を考えましょう。システムが製造段階に入った後、メモリ デバイスをクオリファイするためどんなテストを行うかを十分に検討する必要があります。このようなテストを頻繁に実施することにより、十分な量のコンポーネントを確保し、製造が中断されないようにすることが大切です。

製造中は、システムは安定しており変わることはありません。経験上、継続的な品質管理に重要なテストは、マージンおよび互換性テストであることがわかっています。システムは安定していますので、SI テストはまったくと言っていいほど無意味です。

結論

本稿の目的は、FPGA とメモリ インターフェイスのテストおよび検証方法について、設計者のみなさんに再考を促すことです。効率のよいテスト プラクティスを使うことにより、メモリの品質管理にかかるエンジニアリング時間をすぐにでも短縮できます。さらに、マージン テストおよび互換性テストを正しく行うことで、SI などの従来メソッドよりシステム内の多くのマージナリティと問題を識別できるのです。1 つですべての問題を解消してくれるテスト手法は存在しないため、それぞれのデザインにとって一番有効なテスト手法を見つける必要があります。

メモリ テストに関する詳細は、Micron Technology 社の季刊誌「DesignLine」の最新記事「Understanding the Value of Signal Integrity」(<http://www.micron.com/>)をご覧ください。





Connecting Intel StrataFlash Memory to
Spartan-3E FPGAs

Spartan-3E FPGA を Intel StrataFlash メモリに接続する

特別な知識・技術が不要で、低コスト・高集積度な
インテル社のフラッシュメモリとの連携

Ying Sue
Senior Technical Marketing Engineer,
Flash Products Group
Intel
ying.sue@intel.com

ザイリンクスの Spartan™-3E FPGA ファミリは、集積度 10 万～160 万システム ゲート規模で量産向け、低コストが要求される民生用電子機器に最適です。旧世代の Spartan デバイス製品より性能、コスト パフォーマンスともに向上しており、標準的なパラレル NOR フラッシュ メモリへのグルーレスな (グルー ロジックを使わない) インターフェイスを可能にする新しいコンフィギュレーション モードを備えています。コンフィギュレーション終了後は、コンフィギュレーション ピンのほぼすべてをユーザー I/O として使用できます。

バイト幅ペリフェラル インターフェイス (BPI) パラレル フラッシュ モードと呼ばれるこのコンフィギュレーション モードでは、低コストで高集積な Intel StrataFlash 3V Memory (J3)、通称 J3 Memory を利用できます。J3 Memory は、同じスペースに2倍のビットを提供するマルチレベル セル技術採用の Intel ETOX プロセス技術を使用しています。J3 Memory には柔軟性を考慮して多彩なパッケージと集積度が用意されており、Spartan-3E FPGA にグルーレスに接続することで以下のすべてを格納できます。

- 1 個以上の FPGA に対するビットストリーム
- FPGA 内のソフト CPU コアに対するブート コード、パラメータ、データ
- Spartan-3E デバイスのマルチブート機能を利用する同一 FPGA に対する複数のビットストリーム

図 1 は、マルチブートに対するデュアル ビットストリーム

と、コード/データ ストレージが、J3 Memory デバイスでどのように共存できるかを示しています。

デザイン ノート

図 2 は、3.3 V の環境における 2 個の Spartan-3E FPGA と 1 個の J3 Memory フラッシュ デバイス間の接続を示しています。ここでは、電源投入シーケンス、リセット、ホットスワップ、フラッシュ メモリの内容保護、x8/x16 モードのトグルなど、デザイン上の要点を詳しく解説します。

電源投入シーケンス

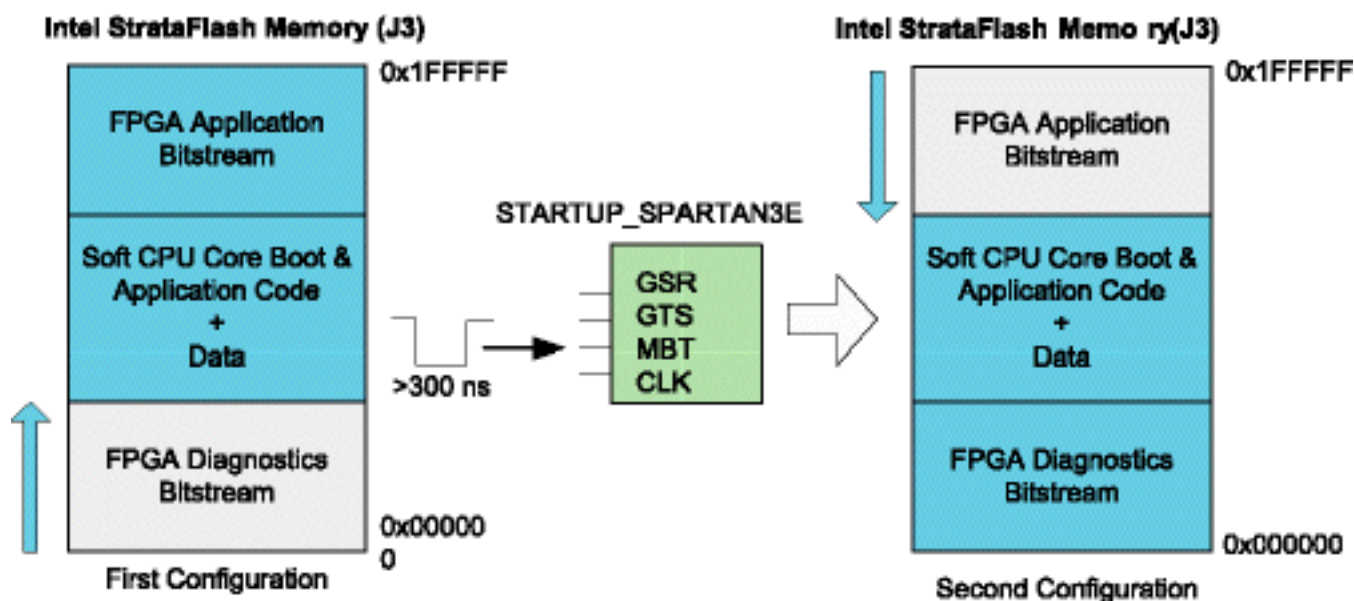
3.3 V のアプリケーションで Spartan-3E デバイスと J3 Memory をサポートするには、3 つの電源電圧が必要です。

- 3.3 V : J3 Memory の VCC と VCCQ 電源端子、および FPGA の VCCO_1 と VCCO_2 電源端子に接続
- 2.5 V : FPGA の VCCAUX 電源端子に接続
- 1.2 V : FPGA の VCCINT 電源端子に接続

FPGA の VCCO_0 と VCCO_3 電源端子は、アプリケーションに応じて 3.3 V、2.5 V、1.8 V、1.5 V、もしくは 1.2 V にできます。

Spartan-3E FPGA のデータシートには、フラッシュ メモリの準備が整う前に FPGA がフラッシュ デバイスから読み出

図 1 フラッシュデバイス(B4980)における複数のビットストリームとコード/データストレージ



この場合のザイリンクス Spartan-3E の使用に関する詳細は、ザイリンクスのデータシート DS312-2 を参照してください。

すというシリアル フラッシュ モード (SPI モード) における電源投入時の注意点が記載されています。一般的なフラッシュ デバイスは、電圧が名目値 (J3 Memory の場合は 2.7 V) に到達した後に内部の初期化を完了するまでに若干時間がかかるため (J3 Memory の場合で 60 μ s)、パラレル フラッシュ デバイスでもこれを考慮する必要があります。

2.5 V と 1.2 V が有効で、3.3 V が 0.4 V~1.0 V (J3 Memory など、フラッシュ デバイスの最小動作電圧未満の電圧レベル) に達する場合、FPGA はフラッシュ デバイスの準備ができる前にプログラミング シーケンスを開始します。この場合、次のいずれかのシナリオが起り得ます。

- ・ 2.5 V と 1.2 V が最小必要電圧に達する前に 3.3V が有効になります (2.7 V に到達)。FPGA がプログラミング シーケンスを開始する際、J3 Memory などのフラッシュ デバイスはいつでも読み出す準備ができているため、この場合問題が発生することはありません。

- ・ 2.5 V と 1.2 V の後で 3.3 V が有効になります。これを回避するには、一般に、3.3 V 電圧モニタを使用して、3.3 V 電源が最小動作スレッシュホールドに到達した後、PROG_B または INIT_B ピンを少なくとも 1 ms の間 Low にホールドします。

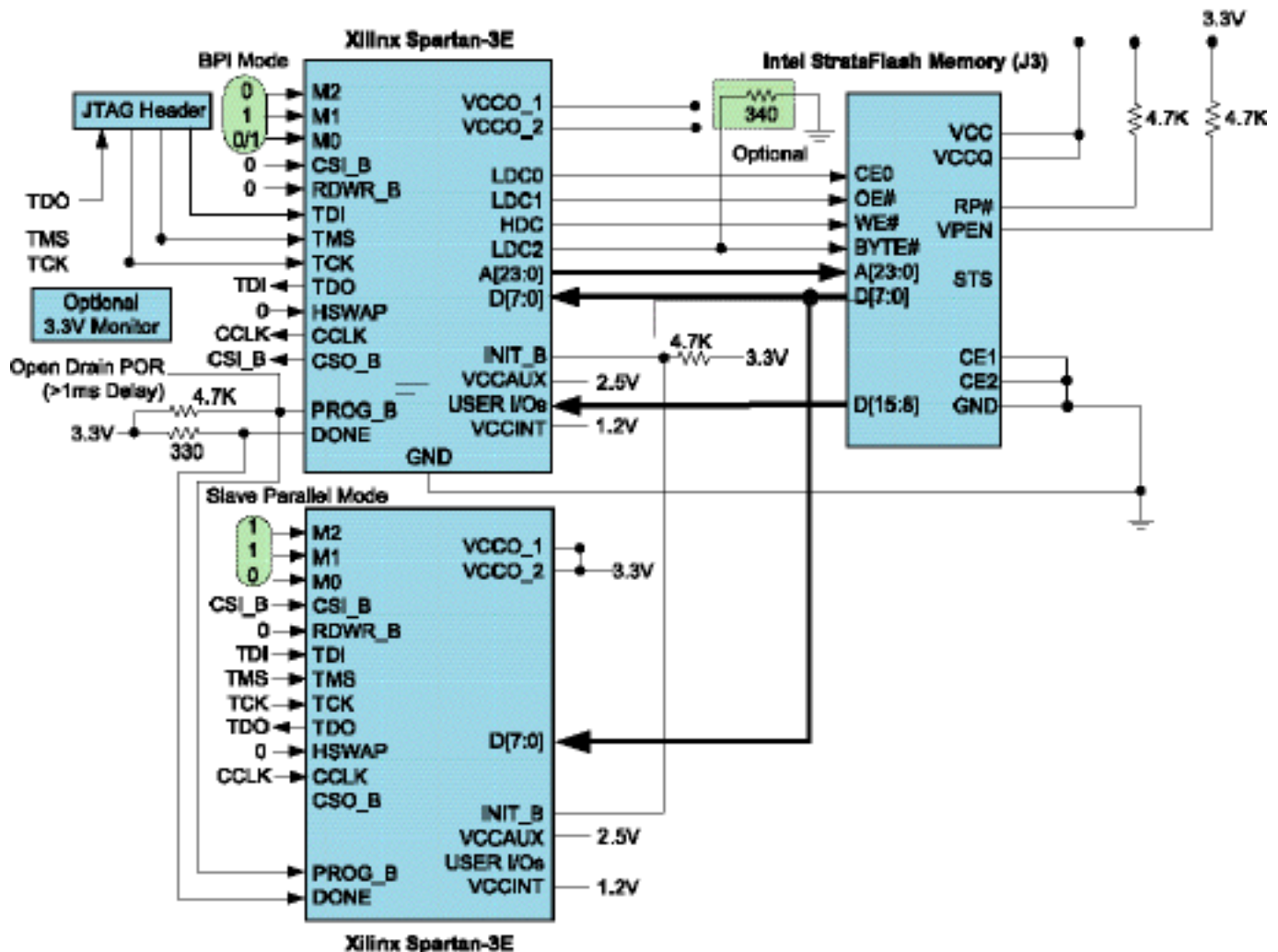
リセット

J3 Memory では、RP# ピンはリセット入力です。リセットでは、内部のフラッシュ回路はディスエーブルされ、出力はハイインピーダンス状態に置かれます。RP# ピンは、最小 Low パルス (35 μ s) が印加されると J3 Memory などのフラッシュ デバイスを非同期ページ モード (リード アレイ) に置きます。

RP# ピンには、次の接続オプションがあります。

- RP# を 3.3 V に接続：

図 1 フラッシュデバイス(B4980)における複数のビットストリームとコード/データストレージ



このように接続した場合、J3 Memory などのフラッシュデバイスは、電源がサイクルされるまでリセットされません。この接続は、FPGA が電源投入時のみプログラミングされ、電源サイクルなしでは再プログラミングされないアプリケーションに適しています。また、この接続は、PROG_B ピンがトグルされている間にフラッシュ デバイスが非リード アレイモードになるリスクが極めて小さいことがわかっている場合にも利用できます。

• RP# を System Reset に接続：

この場合、フラッシュ デバイスがリセットされる前に FPGA が読み出しを始めないよう、RP# が High になる前に PROG_B ピンが Low から High に駆動されないことを確認する必要があります。J3 Memory の場合、一般的な動作電圧では、この時間は RP# が High になってから 150 ns ~ 210 ns です。

• RP# と PROG_B を一緒に接続：

FPGA の再プログラムが発行されると、この接続はフラッシュ デバイスをリード アレイ モードに自動的にリセットします。電圧モニタの出力を使って両方の入力を駆動できますが、これは次の条件に限られます。

- J3 Memory などのフラッシュ デバイスに対する最小リセット パルス幅が満たされている場合。FPGA 上の PROG_B ピンはわずか 0.3 μ s の Low パルスだけでよいのですが、J3 Memory は 35 μ s を必要とします。
- FPGA 上の PROG_B ピンが High に移行してから INIT_B が High に移行するまでの TPL 遅延が、一般的な範囲である 150 ns ~ 210 ns (Intel StrataFlash Memory [J3] データシートの R5 パラメータ)を超える場合。ザイリンクスのデータシートによると、TPL 最小遅延は約 2 ms です。

HSWAP

たとえば、フラッシュ メモリが誤って非リード アレイ モードに置かれたりした場合に、FPGA の電源投入時にフラッシュ メモリに誤ってアクセスするのを防ぐため、HSWAP ピンは 0 に設定できます。0 に設定することで、LDC0 (CE#)、LDC1 (OE#)、HDC (WE#) を High にプルアップする内部のプルアップ抵抗をイネーブルできます。LDC2 ピンには特別な配慮が必要です。HSWAP ピン が High にプルアップされることを必要とするアプリケーションの場合、LDC0、LDC1、HDC の出力に外部のプルアップ抵抗が必要になります。LDC2 (BYTE#)ピンには 4.7 K のプルダウン抵抗が必要です。

x8 モデル専用の動作

BPI モード (M[2:0] = 0b010 または 0b011) では、

FPGA は x8 モードで起動し、コンフィギュレーション中、フラッシュ デバイスの BYTE# ピンに接続されている LDC2 ピンを Low に駆動します。アプリケーションが x16 モードを必要としない場合は、J3 Memory の BYTE# ピンを Low に固定し、LDC2 出力を J3 Memory に接続しないでください。

x8 と x16 モード間でトグル

LDC2 ピンがフラッシュ デバイスの BYTE# ピンに接続されている場合、FPGA はコンフィギュレーション終了後にフラッシュ デバイスを x8 モードから x16 モードに切り替えるため、LDC2 ピンを High で駆動できます。

バイト (x8) モードとワード (x16) モードをトグルするときは、最下位アドレス ロケーションとモード切り替えの遅延を考慮する必要があります。J3 Memory デバイスでは、A0 アドレス ラインがバイト ロケーションを選択します。

x8 と x16 モード間の切り替え待ち時間は、LDC2 ピンがロジック ステートを変え、有効データがフラッシュ デバイスから出力できるようになるまで、1000 ns (Intel データシートでは R12 tFLQV/FHQV パラメータ) です。HSWAP は、次の 2 通りのシナリオにおいて、電源投入時に LDC2 をトライステートにする機能を制御するため、HSWAP の設定と共にこの待ち時間も考慮に入れる必要があります。

• HSWAP が High に固定されている (内部プルアップ抵抗がディスエーブルされている) 場合：

LDC2 に 4.7 K プルダウン抵抗を接続します。このピンは電源投入時に Low にプルダウンされ、コンフィギュレーション中は Low のままです。J3 Memory は、FPGA がコンフィギュレーション ビットストリームをロードし始めた時点では x8 モードにあります。LDC1 と LDC0 は、フラッシュ デバイスが誤って選択されたり、電源投入時に非リード アレイ モードに入ったりしないようプルアップ抵抗を持っている必要があります。

• HSWAP が Low に固定されている場合：

このシナリオでは、LDC2 ピンが電源投入時に内部プルアップ抵抗を通して High にプルアップされ、その後 LDC0 (CE#)、LDC1 (OE#) と同時に Low に駆動されます。FPGA は、フラッシュ デバイスがモードを切り替えるのに必要な 1000 ns 以内にコンフィギュレーション データを読み込みます。これには次の 3 つの対策があります。

- 第 1 の対策は、図 1 に示すとおり、内部プルアップ抵抗に打ち勝ち、FPGA がコンフィギュレーション データを読み込み始める際のフラッシュ デバイスが x8 モードであるよう、LDC2 出力に 340 Ω のプルダウン抵抗を必要とします。340 Ω という値は、Spartan-3 のデータに基づくものであり、FPGA の内部プルアップ抵抗に関してより多くの特性データが利用できるように

なれば、さらに大きくすることも可能です。この際注意する点は、アプリケーションが x16 モードを使用するために LDC2 ピンを High に切り替える必要がある場合、強力なプルダウン抵抗に打ち勝つために大きな出力バッファが必要だという点です。

- 第 2 の対策は、FPGA はコンフィギュレーションを開始する前に初期化シーケンスを必要とするという事実を利用するものです。フラッシュが x8 から x16 モードに切り替わる時、バス上に無効なデータが存在するため、フラッシュ デバイスがモードの切り替えを終了するまで FPGA がこのシーケンスを見るのを妨げます。FPGA デバイスが開始シーケンスを見ないと、正しいシーケンスを見るまでアドレスをインクリメントし続け、残りの有効アドレスを通り、アドレス 0 に戻ります。この結果、ビットストリームのロード時間が長くなることがあります（ビットストリームは最終的にはロードされます）。
- 第 3 の対策は、FPGA ビットストリームに 16 バイトの 0xFF ダミー データをプリフィクスします。このプリフィクスを使うと、フラッシュがモード切り替えを完了した時点、つまりバイト 17 のアドレスに達したとき、FPGA は開始シーケンスを見つけることになります。この場合、ザイリンクスがビットストリーム生成コードを変更する必要があり、ビットストリームのサイズが若干大きくなります。この対策が利用できるかどうかは、ザイリンクスにお問い合わせください。

デザイン上の他の留意点

- アドレス指定：
Spartan-3E FPGA は最大 256 Mb のフラッシュ メモリをアドレス指定できます。複数のフラッシュ デバイスがある場合、低集積フラッシュ デバイスを高集積フラッシュ デバイスでスタッキングすることが可能です。この場合、低集積フラッシュ デバイス上の未使用のアドレス ピンは、FPGA 上の対応するアドレス ピンに接続しておいても問題ありません。これら未使用のピンは、フラッシュ デバイス上では無接続扱いとなります。
- ConfigRate の設定：
J3 Memory への最初のアクセス時間は、集積度に応じて 110 ns ~150 ns です。FPGA の最大 CCLK コンフィギュレーション レート (ConfigRate) を適切に設定してください。
- フラッシュ メモリの内容の保護：
J3 Memoryでは、フラッシュ メモリの内容を保護するため VPEN 入力を使用できます。VPEN 入力が Vpenlk (2.2 V) 以下に駆動されると、フラッシュ メモリの内容を変更で

きなくなります。使用しない場合、この入力は 3.3 V に固定できます。フラッシュ メモリの内容変更を許可/禁止するため、FPGA 上のピンに接続することも可能です。

- 電源のデカップリング：
J3 Memory などのフラッシュ デバイスがイネーブルされている場合、多くの内部条件が変わります。回路に電圧が加えられ、チャージ ポンプに電源が供給され、内部電圧ノードが急激に上がります。このような内部活動は、一過性の信号を生み出します。これらの影響を最小限に抑えるため、各 VCC/VSS および VCCQ 信号の間に 0.1 uF のセラミック キャパシタが必要になります。キャパシタはデバイス接続部のできるだけ近くに配置する必要があります。
- FPGA コンフィギュレーション ピンの再使用：
J3 Memory などのフラッシュ デバイスを駆動するピンのほとんどは、汎用 I/O ピンとして使用できます。ただし、次のピンは再使用しないでください。
 - LDC0：フラッシュ チップセット イネーブル
 - LDC2：フラッシュ バイト/ワード モード制御
- Execute-In-Place (XIP)：
インテル社は、XIP をサポートする一連のソフトウェアを提供しています。XIP とは、外部 RAM を使用せずに消費電力を削減するため、J3 Memory などのフラッシュ デバイスのコードを直接実行することです。ソフト CPU コアに対するソフトウェア デザインでは、こうした補助機能を使用できます。CPU コードとデータに対して XIP 使用モデルが展開されている場合、VPEN 入力を High にプルアップする必要があります。また、フラッシュのコンフィギュレーションコードとブートコードを含むメモリ ブロックは、偶発的なプログラミングや消去を防ぐため、ソフトウェア コマンドを使って個別にロックできます。

結論

Spartan-3E デバイスに BPI コンフィギュレーション モードを追加することで、FPGA ビットストリームとブート/アプリケーション コードを標準的な NOR フラッシュ メモリに統合することが可能です。

この結果、Spartan-3E FPGA と Intel StrataFlash J3 NOR メモリの低コスト、幅広い集積度という特徴を活かし、旧世代の Spartan FPGA では達成できなかった、より広範な量産アプリケーションに優れたコスト効果で採用していただけます。

J3 Memory の詳細は、<http://www.intel.com/design/flcomp/prodbref/298044.htm> をご覧ください。ザイリンクスの Spartan-3E FPGA ファミリのデータシートは、<http://direct.xilinx.co.jp/bvdocs/publications/ds312.pdf> をご覧ください。





Virtex-4 Source-Synchronous Interfaces Tool Kit

Virtex-4 で ソース同期を実現する ツールキットの機能解説

Virtex-4 LX FPGAファミリ ベースのアプリケーションの
デザインと検証に完璧な開発プラットフォームを提供する
Virtex-4 ML450 ソース同期インターフェイス ツールキット

David Naylor
Sr. Product Applications Engineer
Xilinx, Inc.
david.naylor@xilinx.com

今日、テレコムおよびネットワーキング システムの多くは、低電圧差動信号処理 (LVDS) やその他の差動 I/O 規格に準拠する高バンド幅のインターフェイスを使用しています。差動 I/O 規格は、システム性能とシグナル インテグリティを改善することで、システムの設計を簡素化します。

最先端のシステム デザインを可能にしているのが、SPI-4.2、SFI、RapidIO、HyperTransport といった I/O ベースのプロトコルです。これらテクノロジーの利点を活かすには、デバイスがお互いに連携し、かつ規格に準拠するよう、さまざまな課題を解消していく必要があります。

ザイリンクスは、主要なシステム インターフェイス プロトコルについて、Virtex™-4 開発ボード、規格に準拠する知的財産 (IP) コア、そして無償のリファレンス デザインを提供しています。したがって、設計者は相互運用性と規格の準拠に頭を悩ませることなく、ユーザーのアプリケーション デザインに専念できるのです。

Virtex-4 ソース同期インターフェイス ツールキットを使うと、ネットワーキングからテレコム、サーバ、コンピューティングシステムにいたるまで、かつてないほど迅速かつ容易にデザイ

ンできます。

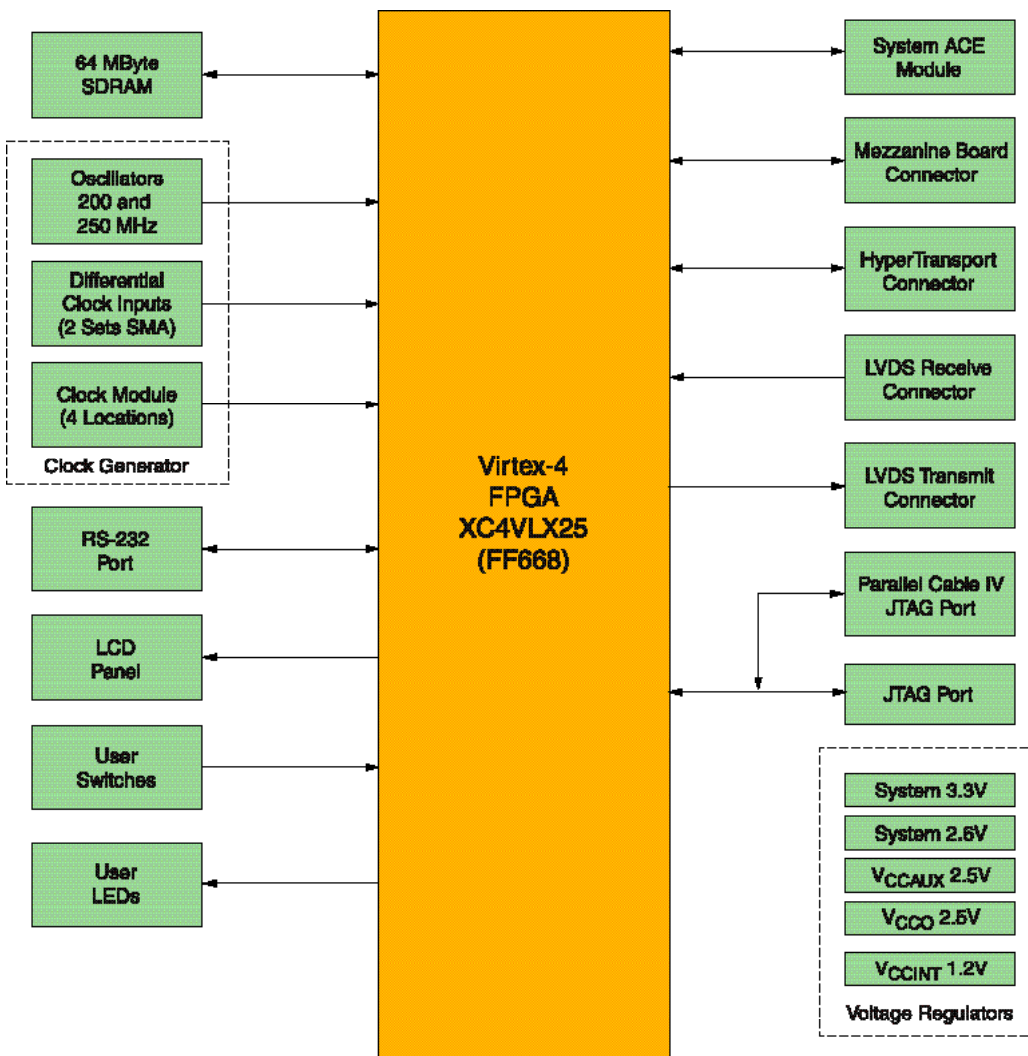
Virtex-4 ML450 ソース同期インターフェイス ツールキットには、次のものが同梱されています。

- Virtex-4 ML450 開発ボード (XC4VLX25FF668 搭載)
- 5 V/6.5 AC/DC 電源
- 国別の電源コード
- RS232 シリアル ケーブル、DB9-F - DB9-F
- 4 枚のクロック モジュールドータ ボード
- LVDS テスト用の「ブルーリボン」ループバックケーブル 2 組
- ドキュメントとリファレンス デザインを収録した CD-ROM

Virtex-4 ML450 開発ボード

Virtex-4 ML450 ソース同期インターフェイス ツールキット のメイン コンポーネントは、ML450 開発ボードです (図 1)。Virtex-4 XC4VLX25FF668 が高速コネクタに接続されているこのボードは、複数の一般的なプロトコルを使っ

図 1 Virtex-4 ML450 ネットワーキング インターフェイス開発ボード



て高速なインターフェイス デザインを開発できます。

ML450 開発ボードは、FPGA にいくつかの接続インターフェイスを提供するシンプルなボードです。このボードは、電圧レギュレータ、シリアル RS232 コネクタ、小型のグラフィックス LCD、数個のユーザー プッシュボタンとLED、DDR-1 SDRAM を含む基本的なサポート回路で構成されています。これらシンプルなペリフェラルは、PC と FPGA との通信を実現するとともに、基本的な入力/出力インジケータを提供します。コンフィギュレーションは JTAG コネクタを介してイネーブルされますが、ビットストリームのストレージとロードに System ACE™ CF (コンパクト フラッシュ) を使うこともできます。

ボードは、Virtex-4 FPGA の高速な I/O 機能を実際に利用できるようにデザインされています。80 の差動チャンネルを Samtec 社の 4 つのコネクタにピンアウトし、40 ペアのそれぞれを FPGA の片側にある 2 つのコネクタに配線してあるため、「送信」と「受信」インターフェイスを指定できます。ツールキットには、「ミニ同軸」フラット ケーブルが同梱されており、送信コネクタと受信コネクタの間で高速データのループバックが可能です。

これらの差動信号に加えて、HyperTransport Consortium DUT (Device Under Test) 準拠のコネクタに別途 32 ペアが配線されているため、HyperTransport に準拠する他のボードへのインターフェイスを開発することもできます。

ML450 評価プラットフォームは、SFI-4 を含む広範な通信規格をサポートしています。図 2 は、ML450 ボード上で動作している SFI-4 デモのユーザー インターフェイスです。このユーザー インターフェイスには、16 個の LVDS トランスミッタから受信したデータの整合性を測定する、ビット エラー レート テスタ (BERT) があります。複数の擬似ランダム ビットシーケンスからどれか 1 つを選んでデータをシミュレートできるうえ、ユーザー インターフェイス上のエラー カウンタが、伝送中に起こるすべてのビット エラーをその都度カウントしていきます。また、BERT はどのチャンネルが受信エラーを起こしたかを追跡しているため、マルチ チャンネル デザインでのトラブルシューティングを容易に行うことができます。

ML450 プラットフォームは、最大 700 MHz の 16 チャンネル シングルデータ レート (SDR) (図 3) と、最大 500 MHz の 16 チャンネル ダブル データ レート (DDR) (図 4) の両方のデザインをサポートします。16 チャンネルの伝送媒体は、Samtec 社の 2 個のコネクタ、12 インチのリボン ケーブル、10 インチの FR4 で構成されます。

ML450 は、FPGA の電源ピンが名目値の $\pm 5\%$ となるよう命令する機能も備えています。この機能は、デザインにストレスをかけてマージナル ビヘイビアを識別するための便利な方法です。電圧はユーザー インターフェイスのスライダ ボタンを左右に動かすだけで制御できます。図 2 は、VccAux を $+5\%$ に上げ、他のサブライ電圧を名目値のままにした状態を示しています。

ボードに差し込む小型のドータカードは、SFI-4 デザインに使われる高速クロックを提供します。このドータカードは、200

～ 700 MHz のプログラマブルな LVDS クロックを生成するため、面倒なベンチトップ パルス ジェネレータは不要です。

ボードの特徴

ML450 開発ボードには次のものが含まれています。

- XC4VLX25FF668
- 8 つのクロック ソース
 - 200MHz と 250MHz のオンボード オシレータ
 - SMA 差動クロック入力コネクタ 2 組
 - Samtec 社のクロック モジュール コネクタ 4 個
- 64 × 128 ピクセルの LCD 1 個
- DB9-M RS232 ポート 1 個
- 最大 8 つの FPGA コンフィギュレーション イメージ ファイルを格納/ダウンロードできる System ACE CF コンフィギュレーション コントローラ
- Samtec 社の LVDS コネクタ 4 個
(合計 40 の入力チャンネルと 40 の出力チャンネル)
- HyperTransport コネクタ 1 個
(HyperTransport Consortium DUT コネクタ準拠)
- $\pm 5\%$ の出力マージン テスト機能を持つオンボード 電圧レギュレータ

クロック生成

ML450 開発ボードのクロック生成部分は、Virtex-4 FPGA が必要とするすべてのクロックを提供します。クロック ソースは 8 つあります。

- Epson EG2121CA 2.5 V 250 MHz 差動低電圧ポジティブ エミッタ結合ロジック (LVPECL) オシレータ
- Epson EG2121CA 2.5 V 200 MHz 差動 LVPECL オシレータ
- 差動 SMA クロック入力 2 個
- Samtec 社のユーザー クロック ソケット 4 個

差動 SMA クロック入力が FPGA のグローバル クロック入力に接続されており、上下の半分ずつにアクセスします。200 MHz のオンボード オシレータは I/O の遅延を調整し、HyperTransport IP で使えるよう 250 MHz のオンボード オシレータも用意されています。ツールキットに含まれている 4 個のクロック モジュールは次のとおりです。

- タイプ A: ダイレクト バランス方式の差動 SMA 入力
- タイプ B: Epson EG2121CA 2.5 V 400 MHz 差動 LVPECL
- タイプ C: ICS プログラマブル、200 ～ 700 MHz
- タイプ D: LVDS に結合され、バランスを取っていないシングルエンドの変圧器

クロック モジュールのドータボードの出力は、すべてドータボード上でLVDSに変換されることに注意してください。

SDRAM メモリ

ML450 開発ボードは、64 MB の DDR-1 SDRAM メモリ (Micron Semiconductor 社の MT46V32M16N-5B) を提供しています。

液晶ディスプレイ

ML450 開発ボードは、(注：必要なければ取り除いてください。以下同様) 64 × 128 LCD パネルに 8 ビット インターフェイスを提供しています (DisplayTech 社の Q64128E-FC-BC-3LP、64 × 128)。

RS232 ポート

ML450 開発ボードは、シンプルな RS232 ポートに DB9-M 接続を提供しています。このボードは RD、TD、RTS、CTS 信号の駆動に、Maxim 社の MAX3316 デバイスを使用します。シリアル通信をイネーブルするには、FPGA 内に UART コアを用意する必要があります。

System ACE インターフェイス

ML450 開発ボードは、JTAG コンフィギュレーション コネクタに加え、Virtex-4 をコンフィギュレーションするための System ACE インターフェイスを提供しています。このインターフェイスにより、ソフトウェア設計者は FPGA 内のソフト プロセッサ IP に対して、リムーバブルな CF カードからコードを実行できます。

LVDS コネクタ

ML450 開発ボードは、40 チャンネルの送信 LVDS 信号と 40 チャンネルの受信 LVDS 信号を提供しています。これらの信号は、送信用として 2 個の Samtec 社の QSE-DP コネクタ、また受信用として別途 2 個のコネクタに分配されます。

HyperTransport コネクタ

ML450 開発ボードは、16 チャンネルの送信/受信データに加え、Samtec社の QSE HyperTransport コネクタに多種多様な制御信号を提供しています。

結論

Virtex-4 ソース同期インターフェイス ツールキットを使うことにより、ネットワーク、テレコム、サーバー、コンピュー

ティング システムなどを、これまで以上に迅速かつ容易にデザインできます。

デモ ボードとキットの詳細は、<http://www.xilinx.co.jp/ml450/> をご覧ください。



図 2 BERT のユーザーインターフェイス



図 3 700 MHz SDR LVDS アイダイアグラム

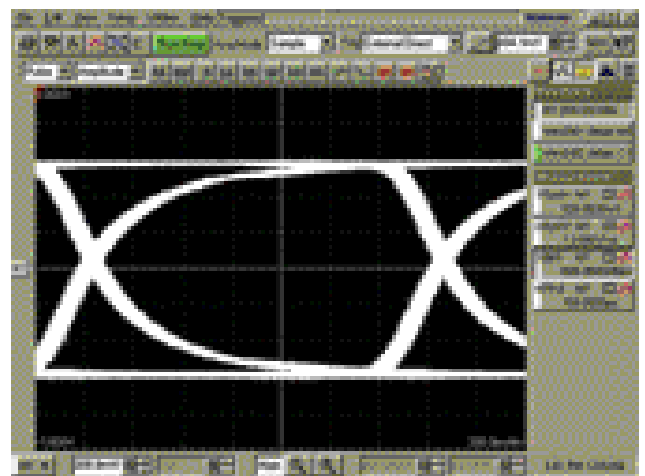
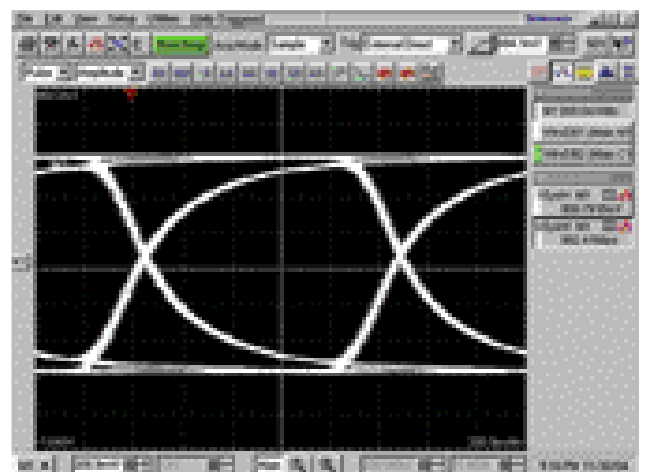


図 4 500 MHz DDR LVDS アイダイアグラム





Bridging System Packet Interfaces

システム パケット インターフェイス間の ブリッジング

知的資産とハードウェア プラットフォームの活用により
短期接続が可能になる

Mark McLaughlin

Design Engineer, IP Solutions Division

Xilinx, Inc

mark.mclaughlin@xilinx.com

Tom Fischhaber

Staff Design Engineer, IP Solutions Division

Xilinx, Inc

tom.fischhaber@xilinx.com

Jeremy Goolsby

Staff Design Engineer, IP Solutions Division

Xilinx, Inc

jeremy.goolsby@xilinx.com

この 2、3 年の間に、光インターネットワーキング フォーラム (OIF) のシステム パケット インターフェイス (SPI)、2.5 Gb/OC-48 (SPI-3) と 10 Gb/OC-192 (SPI-4.2) が、あらゆる主要フレーム ASSP のデファクト スタンダードになりました。SPI インターフェイスは、インテル社の IXP2800 や IXP2400 といった次世代プロセッサを含むネットワーク プロセッサの分野にも普及してきました。これらのインターフェイス規格は範囲は広いものではありませんが、Time-to-Market が最優先される業界で完全に準拠するソリューションを提供しなければならないシステム設計者にとって、大きなデザイン課題でもあります。しかし、ザイリンクスの Virtex™-4 アーキテクチャを用いることで、これらマルチギガビットのシステム パケット インターフェイス アプリケーションをインプリメントすることが可能になります。

Virtex-4 デバイスと、ザイリンクスの種々の IP ソリューションを併用することで、システム設計者は次世代システムをかつてないスピードで構築できます。こうした製品は複数のプロトコルのブリッジングを必要とすることが多く、まさに FPGA が最適なアプリケーションです。

図 1 に、一般的なブリッジング アプリケーションの 2 つの例を示します。最初の FPGA は、既存のフレームを活かしつつ、一般的によく使われている Intel IXP2800 ネットワーク プロセッサのサポートを可能にすることで、4 つの SPI-3 (PL3) インターフェイスを 1 つの SPI-4.2 インターフェイスにブリッジングします。2 番目の FPGA は、Virtex-4 エンベデッド マルチギガビットトランシーバを使って SPI-4.2 インターフェイスをバックプレーンにブリッジングします。Virtex-4 FX ファミリーは、PCI Express、XAUI、Aurora など、幅広いバックプレーン アプリケーションをサポートします。

ザイリンクスは、ブリッジング アプリケーションを容易に開発できるよう、多数のアプリケーション ノートとリファレンス デザインを提供しています。ザイリンクスとパートナー各社の開発ボードを使用することで、これらのハードウェア デザインを数週間や数カ月ではなく、数時間から数日程度で完了できます。ザイリンクスは以下のリファレンス デザインを提供しており、<http://www.xilinx.co.jp/support/library.htm> の「アプリケーション ノート」より入手可能です。

- ギガビット システム リファレンス デザイン (XAPP 536)
- GFP-F 対応のオプションを備えた SPI-4.2 へのギガビット イーサネット集合体 (XAPP 695)
- メッシュ ファブリック リファレンス デザイン (XAPP 698)
- ギガビット イーサネットから Aurora へのブリッジ (XAPP 777)
- SPI-4.2 からクワッド SPI-3 へのブリッジ (XAPP 525)

本稿では、SPI-4.2 からクワッド SPI-3 へのブリッジの利点と有用性についてご紹介し、4 つの SPI-3 コアを 1 つの SPI-4.2 コアにブリッジする方法を解説します。このソリュー

ーションは、チャンネル化したバッファリング、アービトレーション、およびシステム パケット インターフェイス プロトコルを使うフロー制御をインプリメントします。このロジックを必要とするシステム設計者なら、誰でもこのデザイン例をうまく活用できるはずです。

SPI-4.2 から クワッド SPI-3 へのブリッジ

図 2 に、SPI-4.2 からクワッド SPI-3 へのブリッジ デザインの概要を示します。このデザインは SPI-3 と SPI-4.2 の間で電氣的インターフェイスを変換するだけでなく、データ バッファリングとデータ幅変換、アービトレーション、フロー制御の管理も行います。

SPI-3 から SPI-4.2 インターフェイスへのデータフローについては、ブリッジは複数のデータ ストリーム (最大で 4 つの異なる SPI-3 インターフェイス) を蓄積し、SPI-4.2 データの 1 つの出力ストリームを生成します。各チャンネルにはプログラマブルなデータが格納され、SPI-4.2 インターフェイスからのフロー制御が各チャンネル上で転送するデータ量を決定します。シンプルなラウンドロビン アービトレーションがインプリメントされていますが、システムの要件に応じて、より複雑なアルゴリズムを提供するよう容易に拡張できます。

SPI-4.2 から SPI-3 の方向では、ブリッジは 1 つの SPI-4.2 インターフェイスを個々のチャンネルにデマルチプレクスし、4 つの別々の SPI-3 出力ストリームを生成します。各チャンネルにはプログラマブルなデータが格納され、SPI-3 インターフェイスからのフロー制御が各チャンネル上で転送するデータ量を決定します。

リファレンス デザインには、4 つの SPI-3 コアと SPI-4.2 コアの間にブリッジ ロジックをインプリメントするために必要なすべてのデザイン ファイル (VHDL と Verilog の両方) が同梱されています。また、リファレンス デザインはユーザーの要件に基づいてデザインをカスタマイズできるよう、各種パラメータをサポートしています。

- フロー制御情報を決定するための FIFO しきい値
- 追加のデバイス/パッケージ情報
- SPI-4.2 コアのスタティックおよびダイナミック アラインメント コンフィギュレーション
- コンフィギャブルな FIFO の深さ

これらのデザイン ファイルとドキュメントを利用することで、SPI-4.2 からクワッド SPI-3 へのブリッジング ニーズを満たす、完全なソリューションを実現できます。このブリッジの詳細な説明と、サポートするデザイン ファイルについては、前述のアプリケーション ノート XAPP 525 をご覧ください。

このリファレンス デザインは、SPI-3 から SPI-4.2 のアプリケーション以外でブリッジング アプリケーションを作成するシステム設計者にとっても非常に有効です。このブリッジによってインプリメントされる機能としては、チャンネルごとの

データ バッファリング、アービトレーション、フロー制御などがあります。チャンネル化した FIFO は、非対称アスペクト比と First-Word Fall-Through (FWFT) を含む各種の機能をサポートするザイリンクスの FIFO Generator により作成されます。非対称アスペクト比は異なるデータ幅を容易に変換でき、また FWFT は読み出し動作を発行しなくても FIFO から次に利用可能なワードを先読みできます。FIFO Generator の詳細は、本号の「最適化された FIFO ソリューションの設計法 (Never Design Another FIFO)」をご覧ください。

IP コアのソリューション

SPI-3 と SPI-4.2 コアは、OIF の仕様を満たすため、高速な I/O 動作とタイトなタイミングを必要とします。これらのコアは、Virtex-4 ChipSync™ テクノロジーを通して OIF の要件を容易に満たし、エンベデッド I/O SERDES とダイナミック フェーズ アラインメントを使用できるようにします。これにより、SPI ユーザーにとってデザインの複雑さが大幅に緩和され、SPI インターフェイスではなくシステムの要件に専念できるようになります。

SPI ソリューションは標準の IP リリース形式で CORE Generator を通して提供され、即座にシミュレーションできる機能を実装で提供します。また、ハードウェア評価ライセンスを取得すると、SPI コアをハードウェアにダウンロードしてシステムをフルに評価できます。コアは 2 時間ほどで時間切れとなり、その間にアプリケーションでハードウェアを完全に評価できます。

SPI IP スイートには、OC-192 アプリケーション用として SPI-4.2、また OC-48 アプリケーション用として SPI-3 Link、SPI-3 PHY、SPI-4.2 Lite が同梱されています。SPI-4.2 と SPI-3 IP は、これら複雑なインターフェイス プロトコルに対応するシンプルな既成のソリューションを提供します。

SPI-4.2 および SPI-4.2 Lite コア

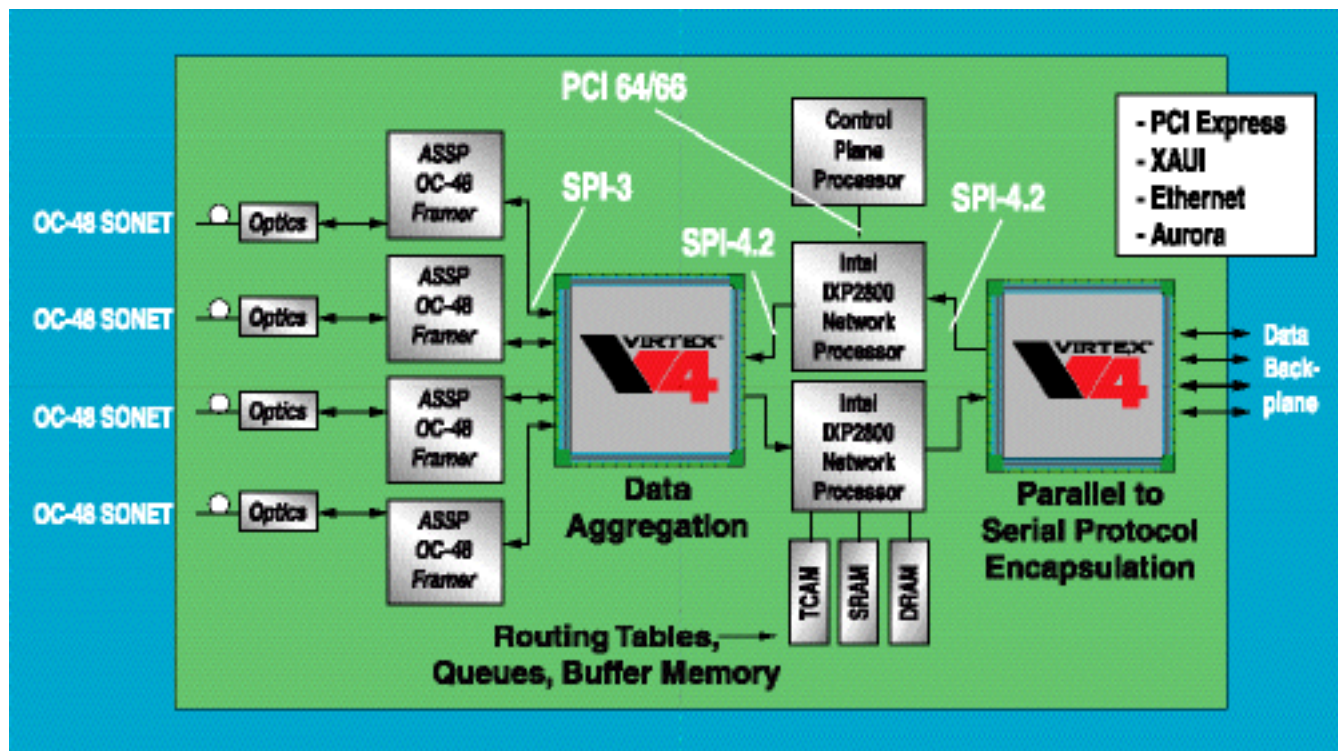
ザイリンクスの SPI-4.2 IP は、完全に検証済みのプラグイン SPI-4.2 インターフェイス ソリューションです。このコアは、ダイナミック フェーズ アラインメントを使って 1 Gbps を超えるデータ レートで動作する機能など、コアを特定のデザイン条件に合わせて作り変えるためのコンフィギュレーション オプションを多数備えています。

ザイリンクスは、SPI-4.2 フルレート コアに加えて、SPI-4.2 Lite コアも提供しています。SPI-4.2 Lite IP は、SPI-4.2 インターフェイスの効率を、それより遅い OC-48 アプリケーション用に流用した IP コアです。OC-96 の最大周波数 (5 Gbps) で動作する点を除けば SPI-4.2 OIF 仕様に完全に準拠しており、フル レート SPI-4.2 コアのリソースの 50 % も必要としません。

SPI-3 PHY および SPI-3 Link コア

ザイリンクスの SPI-3 PHY および SPI-3 Link コアは、OC-48 アプリケーションに最適なソリューションを提供し、コアをアプリケーション ニーズに合わせてカスタマイズするための

図 1 Virtex-4 FPGA を使った 2 つのブリッジングアプリケーション例



各種オプションをサポートします。SPI-3 Link および SPI-3 PHY のコアは 32 ビット インターフェイスだけでなく、さまざまなフレームワーク プロセッサにインターフェイスできるように 8 ビットおよび 16 ビットのインターフェイスもサポートします。かつて固定ポイントのソリューションとして提供されていた SPI-3 PHY と SPI-3 Link コアは、2005 年第 3 四半期に ISE™ 7.1i IP アップデート 3 でリリース予定です。

結論

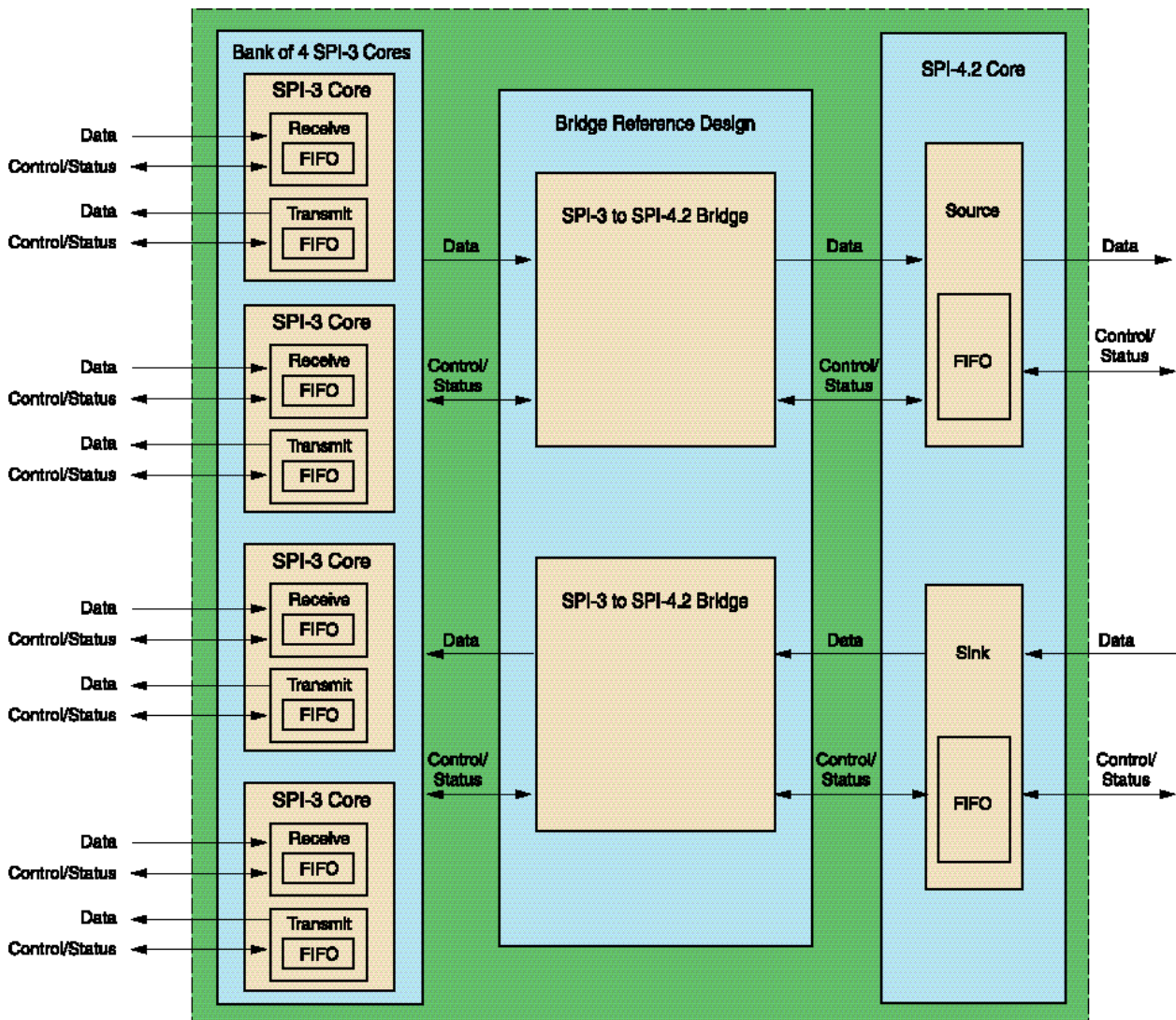
ザイリンクスの IP とリファレンス デザインは、プロトコル間にカスタムのブリッジング ソリューションをインプリメントするための卓越したソリューションです。SPI-4.2 からクワッド SPI-3 へのブリッジング アプリケーションや、チャンネル化したバッファリング、アービトレーション、およびシステム パケット インターフェイス プロトコルを使うフロー制御を必要とするシステムに対して、XAPP525 ブリッジ デザインをそのまま流用できます。さらに、SPI コアはこれらの複雑なインタ

ーフェイス標準に完全に準拠するドロップイン ソリューションを提供します。ザイリンクスの SPI ソリューションは、デザイン課題の負荷を軽減するため、ユーザーは Time-to-Market をより実現することができます。低コストであらゆる機能を完備した Virtex-4 ファミリーと、各種の SPI IP およびマルチギガビット ブリッジ デザインは、まさに最高のコンビネーションになります。

本稿では SPI IP に関してのみご紹介しましたが、ザイリンクスはあらゆる接続性およびブリッジング ニーズを満たす、豊富な IP ソリューションを用意しています。詳細は、http://www.xilinx.co.jp/products/design_resources/conn_central/index.htm をご覧ください。

SPI-4.2 コアと、Virtex-4 アーキテクチャにおけるその利点については、Xcell Journal 52 号に掲載されている http://www.xilinx.co.jp/xcell/xl52/jp52xcell_05.pdf をご覧ください。また、SPI-4.2 と SPI-3 IP の各ソリューションについては、http://www.xilinx.co.jp/ipcenter/posphyl4/spi42_core.htm をご覧ください。

図 2 SPI-4.2 からクワッド SPI-3 へのブリッジデザインのブロックダイアグラム





Managing Signal Integrity

シグナル インテグリティの 実現

ノイズやジッタの問題をいかに克服するか

Steve Sharp

Sr. Marketing Manager

Xilinx, Inc

steve.sharp@xilinx.com

ゲームの言い分

Panch Chandrasekaran

Connectivity Marketing Manager

Xilinx, Inc

panch.chandrasekaran@xilinx.com

シグナル インテグリティの問題は、混雑が激しく騒々しい展示会場で人と会話をすることに似ています。ホール片隅の防音壁に囲まれた静かな場所にいて、近くにあまり人がいなければ、騒音で話が聞こえないということはありません。しかし、周りに数百人の来場者がいて、近くの展示ブースからマイクの声が響き、音を遮断もしくは吸収する壁が何もない展示フロアの中央では、ほとんど会話にならないでしょう。

ロジック デザインの古き良き時代には、シグナル インテグリティについて考えることはまずありませんでした。5 V の電源、ボード上に実際に配線されているリード線付きの DIP パッケージ、そして当時としては高速な 5 MHz のマイクロプロセッサというのが一般的でした。

ボードのレイアウトに少し気を配って各チップの隣にセラミック製のバイパス キャパシタを配置しておけば、信号について悩むことはなかったのです。1 は 1、0 は 0 のままでした。た

とえ信号に 100 mV のノイズがかかっても、ロジック レベルを変えるほどの力はありませんでした。

今日、設計者はビット レートやエッジ レート、クロック スピードの高速化というデザイン要件と、動作電圧の低減、パッケージ サイズとボール ピッチの微細化、そして狭くなる一方のボード スペースにさらに多くのコンポーネントを詰め込むという技術的進歩の狭間で四苦八苦しています。

今日のシグナル インテグリティ

現在のソース同期インターフェイスを見てみましょう。DDR と QDR メモリ インターフェイスは急激に高速化しており、DDR2 のスピードは 500 Mbps を超えるほどです。ビット レートも高速化し、バス幅はますます広がっています。また、ビット レートの高速化に伴いエッジ レートも高速化し、今ではわずか 2 ~ 300 ピコ秒です。

高速化するのはいよとして、ここでいくつかの問題に対処する必要があります。スピードが遅かったころはまったく問題にならなかった寄生インダクタンスと寄生キャパシタンスが、突如として非常に重要な問題になってきます。その結果として起こるノイズは大きな懸念材料です。今日の FPGA では、一般に数百の I/O が切り替わり、高レベルな同時スイッチング出力ノイズ (SSN) が発生します。SSN はシステムに多くの影響を及ぼし、特にジッタはタイミング マージンを減少させたり、システム障害を招いたりすることさえあります。

シグナル インテグリティを無視して、システムがデザインどおりに動作するはずだと楽観視するのは禁物です。後になって、システムを正常に動作させるためにクロック レートを下げたり、シグナル インテグリティの問題を解決するためにボードをゼロから再設計せざるを得ない状況に直面する可能性があるからです。

シグナル インテグリティの現状

良好なシグナル インテグリティとは、ロジック信号における不要なノイズを制御することです。ノイズは、主に次の 2 つに分けられます。

- ・レベル関連のノイズは、信号のロジック レベルに影響を与えます。ノイズが大きすぎると、信号がしきい値を超えて適正なロジックステートから不適正なステートになり、他のロジックに伝播していくことがあります。
- ・時間関連のノイズ、つまりジッタは、信号遷移のポジションに影響を与え、データ サンプリングのセットアップ/ホールド ウィンドウ違反を招いて、その結果不正なデータがサンプリングされてシステム全体に伝播していきます。

レベル ノイズとジッタを併発すると、電圧と時間の両方で信号マージンが減少し、良好なデータを利用できる「アイ」が縮小されることになります(図 1)。

ノイズの制御

シグナル インテグリティを確保するには、しっかりしたパッケージが不可欠です。システムにはノイズの発生源がたくさんあります。

ノイズの発生源がボードにある場合、時間と手間のかかるデバッグ プロセスを経てどうにか問題の箇所を見つけて解決できる可能性はあります。問題がパッケージにある場合は、デザ

インやベンダ、部品を変える以外、選択肢はありません。その場合、かなりの時間が無駄になり、製品がもたらすはずの収益が大きく損なわれます。これを回避するためにも、デザインのしっかりした低インダクタンスのパッケージを使うことが重要なのです。

スピードがまだ遅かった時代には、短い信号パスが信号特性を変えることはありませんでした。しかし、たとえビットの立ち上がり時間が数ナノ秒であっても立ち上がり時間が数百ピコ

秒になる今日、信号の周波数成分はギガヘルツに達し、パッケージ トレース(配線)のようなごく短い信号パスさえ信号に影響を及ぼします。

信号ラインには、かならず戻り電流のためのリターンパスがあります。シングルエンド信号の場合、リターンパスは通常、GND または VCC リファレンス プレーンです。50 Ω のラインを維持するため、リターンパスは信号のすぐ近くに配置する必要があります。

PCB トレースはそれほど心配ないものの、ビアには細心の注意を払う必要があります。大規模 FPGA の場合、ブレイクアウト領域(パッケージ ボールから PCB までのエリア)は信号ビアの密集部分であるため、極めて重要です。

一般に、SSN は「グラウンド バウンス」として観察され、2 つの異なる現象によって引き起こされます。

まず、ビア領域のクロストークに起因するノイズはループ インダクタンスに左右され、ループ インダクタンスはグラウンド/電源リファレンス

ピンの配置場所から信号ピンまでの近さで決まります。リファレンス ピンから遠い信号ピンほどノイズの影響を受けやすくなります。

この問題は、領域内の多数の I/O が同時に切り替わる時にいっそう悪化します。したがって、適切なピンアウト アーキテクチャ、つまりパッケージ内にグラウンド/電源と信号ピンを正しく分布させることが極めて重要なのです。

2 番目に、許容範囲内でシグナル インテグリティを維持するには、FPGA にクリーンな電源を維持することが大切です

図 1 アイダイアグラムに悪影響を与えるノイズとジッタ

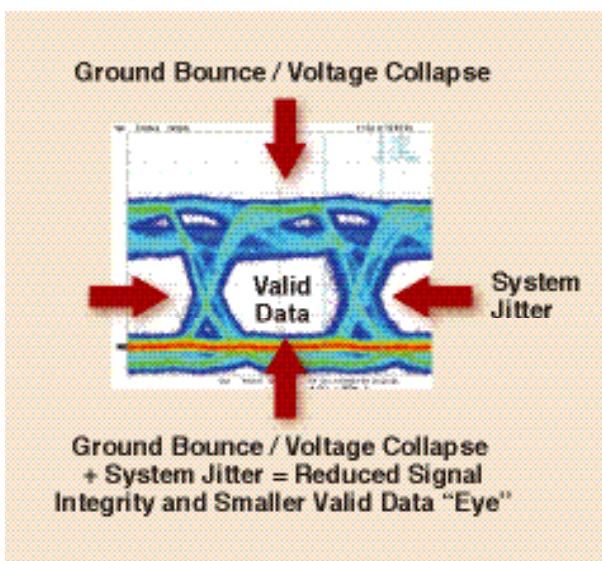
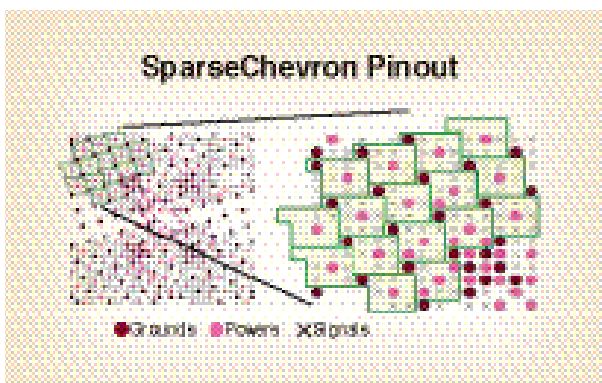


図 2 最適なピン配列を実現する SparseChevron ピンアウト



す。VCC の値が 1.2 V まで下がれば、ノイズ マージンは低減します。

さらに、パワー レール (電源配線) にノイズがあると、出力にジッタが起こり、利用可能なタイミング マージンが縮小します。ノイズはパッケージのインダクタンスと、同時に切り替わる I/O 数に依存するため、最適な信号の取り扱いには低インダクタンスのパッケージが不可欠です。

SSN の課題に対処

SSN の課題に対処しているパッケージの 1 つに、ザイリンクスの Virtex™-4 FPGA パッケージがあります。このパッケージは、LVDS などの差動インターフェイスよりノイズの影響を受けやすい、より高速なシングルエンドのインターフェイスで優れたノイズ パフォーマンスを発揮します。

Virtex™-4 FPGA パッケージのピンアウト アーキテクチャは、総ノイズの約 80 % をカットしています。また、SparseChevron (スパースシェブロン) ピンアウトと呼ばれる、信号、グラウンド、電源ピンを一定間隔に配列したタイル パターンを通して、最適なピン配列を達成しています (図 2)。

このパッケージの信号、グラウンド、電源比は 8 : 1 : 1 です。電源とグラウンドは戻り電流のパスとしても同等に有効ですので、このパッケージの実際の信号対リターン比は 4 : 1 となります。また、ピンの分布は、戻り電流ループを最小限に抑えるため、すべての信号ピンがリターン ピンに隣接するようになっています。

さらに、パッケージのすべてのエリアにある豊富なリターンパスは、戻り電流に対する低インピーダンス パスを提供しま

す。また、ピンアウトは aggressor (侵略してくる駒) からのノイズをより小さな領域に閉じ込め、aggressor の影響が距離に応じて急減するようになっています。クロストークのノイズは蓄積しますので、これにより総 SSN が低減します。

信号終端の簡易化

オンチップの終端処理 (アクティブ終端) は、外部コンポーネントを除去し、終端をドライバやレシーバなど、もっとも問題になりやすい場所の近くに置きます。

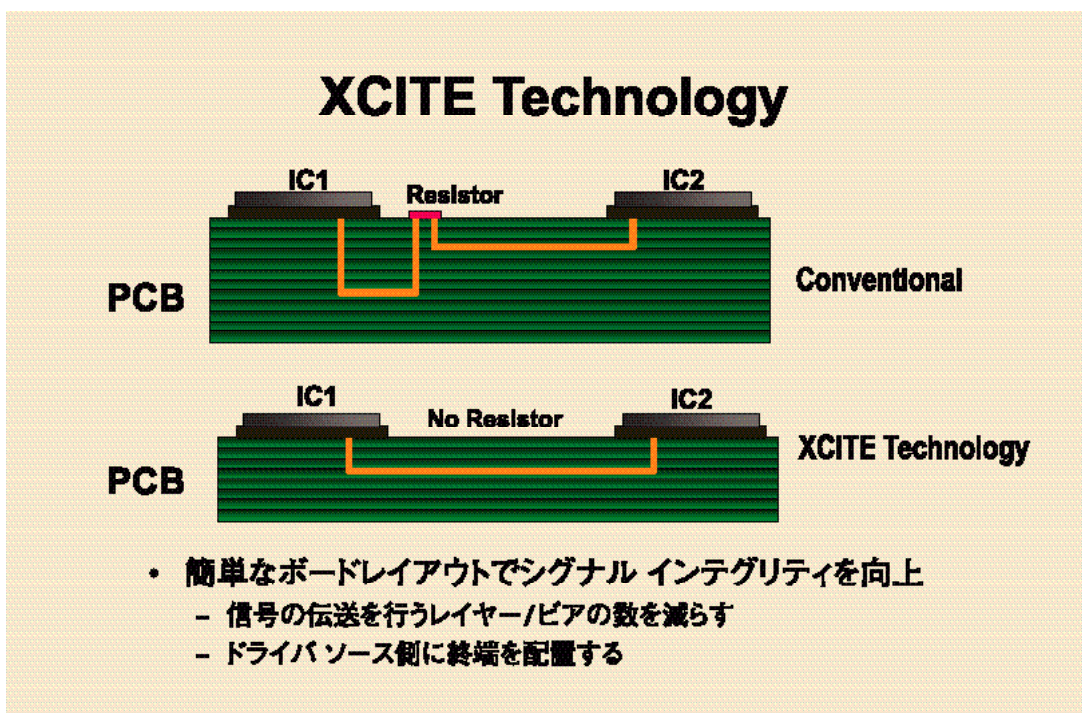
理想的な 50 Ω のライン インピーダンスを維持するため、デザインの際は終端抵抗を各信号に配置するのが一般的です。数百の I/O 信号ともなれば、外部終端抵抗も何百という数になります。ボードに抵抗を配置し、電源とグラウンド プレーンに接続するのは、物理的にけっして容易ではありません。

Virtex FPGA に使われているオンチップのアクティブ I/O 終端、Xilinx Controlled Impedance Technology (XCITE) は、信号の終端処理に伴う多くの問題を解決します。

XCITE は、シングルおよび差動終端に、パラレルとシリアル両方の等価オプションを提供します。インピーダンスは内部リファレンス電圧により制御され、すべての I/O ピンで利用できます。

このアクティブ終端は温度と電圧を自動補正し、終端をそれが属するバッファ回路内に入れ、数百のディスクリート抵抗を取り除くことでボード スペースとコストを節約します。図 3 は、従来の終端技術とザイリンクスの XCITE DCI 終端技術を両方使い、ボード レイアウトと信号トレース パスを単純化したものを示しています。

図 3 PCB の配線を単純化しコストを低減するザイリンクスの XCITE DCI 終端技術



電源プレーンのインテグリティ

電源およびグランド プレーンは、FPGA デザインでシグナル インテグリティを維持するために重要です。対象とする周波数範囲全域にわたり特性インピーダンス (Z_0) を維持するには、シングルエンド信号に対するリファレンス プレーンが非常に低いインピーダンスでなくてはなりません。

そうでないと、インピーダンスが不連続になり、反射によるジッタを招きます。加えて、電源およびグランド プレーンにノイズが多いと、チップ上の回路パフォーマンスに影響を与え、別のジッタを発生させます。インピーダンスを最小限に抑えるため、連続的な電源およびグランド プレーンを備えたパッケージをデザインすることが重要です。

通常、PCB の設計者はノイズを除去してクリーンな電源を維持するため、デカップリング キャパシタを使います。高周波ノイズを低減させるには、デカップリング キャパシタをノイズの発生源近くに配置します。最先端の ASIC と FPGA は、電源ノイズをクリーニングしやすくするため、非常に低インダクタンスなデカップリング キャパシタを搭載しています。

シグナル インテグリティの問題を補正

システムのシグナル インテグリティを改善すれば、FPGA の I/O ピンに到達する高周波信号のデータ有効ウィンドウ (アイ) が向上します。とはいえ、これだけでは問題の半分しか解決したことになりません。図 4 の 533 Mbps DDR2 SDRAM の例に示すように、いかに卓越したデザインであってもデータ有効ウィンドウは縮小します。入力回路は、縮小しているデータ有効ウィンドウの中央にクロックをセンタリングすることでデータをキャプチャする機能を必要とします。

Virtex-4 FPGA は、すべての I/O ブロックに特殊な ChipSync™ テクノロジーを採用することで、データを容易に、かつ高い信頼性でキャプチャすることが可能です。ChipSync は、データを FPGA クロックにセンタリングするのに必要なタップ遅延を生成する IDELAY という高精度な遅延を備えています。I/O ブロックに含まれているメモリ ストロブ エッジ検出口ジックは、この高精度遅延を使ってメモリ ストロブのエ

ッジを検出し、そこからパルス センターを計算します。1 番目から 2 番目までのエッジ間でカウントされた遅延タップの数によりデータを遅延させることで、データ ウィンドウ の中央を FPGA クロック出力のエッジにそろえるのです。この高精度遅延ブロックにより生成されるタップ遅延を使えば、データとクロックを 75 ps の精度以内にそろえることができます。

また、ChipSync テクノロジーは、パラレル インターフェイスをシリアル化、デシリアル化することでデータ レートを内部 FPGA 回路のスピードにマッチさせるエンベデッド SERDES ブロックを搭載しており、差動パラレルバス インターフェイスのデザインを簡易化します。さらに、このテクノロジーはデザイン マージンを高めるためビット単位とチャンネル単位のデスクュー (スキュー除去機能) を提供し、SPI-4.2、XSBI、SFI-4、RapidIO といったインターフェイスのデザインを簡易化します。

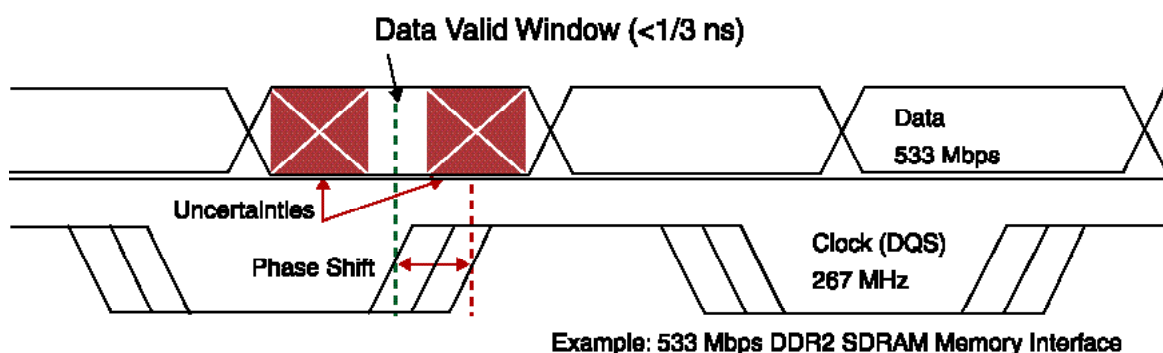
結論

シグナル インテグリティは今日の高速デザインにおける重要な課題であり、今後、より小さなボード スペースにさらに多くの高速信号が詰め込まれ、パッケージが高集積化し、ボール間隔が縮小するに従って、ますます重要になっています。

シグナル インテグリティの問題は電圧と時間の両方に影響を与え、システム内の有効データ ウィンドウを縮小します。これらの問題が大きくなると、システムがまったく動作しなかったり、信頼性が極度に劣化したりして、長い時間と莫大なコストをかけてシステムを再設計しなければならないとなります。

高速システムで信号ノイズを完全に取り払うのは不可能かもしれませんが、いくつかの鍵となる領域に注意を払うことにより、ノイズを最小限に抑えたりタイミングを調節したりして、システム性能の劣化を避けることができます。たとえば、しっかりエンジニアリングされた低インダクタンスのパッケージ、電源デカップリングが組み込まれているデバイス、また必要に応じてアクティブな信号終端を使い、データ有効ウィンドウとクロックの関係性を調節する機能を備えたデバイスを選ぶことで、かなり緩和されるのです。詳細は、<http://www.xilinx.co.jp/signalintegrity/> をご覧ください。

図 4 Virtex-4 FPGAを使った2つのブリッジング アプリケーション例



Hardware / Software Co-Verification

ハードウェアとソフトウェアの協調検証

Mentor Graphics 社の Seamless FPGA による
ソフトウェアおよびハードウェアの可視化と、
プロセスにおけるデザインの反復ループの高速検証を実現

Ross Nelson
Seamless FPGA Product Manager
Mentor Graphics Corporation
ross_nelson@mentor.com

バグ検査を行い、エンベデッド プロセッサをベースとするデザインの HDL を少しだけ手直したという経験は誰にでもあるのではないのでしょうか。いよいよ次は、合成、配置・配線に進むという段階まで来て、突然、結果がわかるのは明日になると気づきます。

大規模の FPGA にはシステム全体を詰め込むことができますが、こうした複雑なシステムを限られた可視性とわずか 1 日のターンアラウンド時間でデバッグしようとする、数週間もの貴重な時間を無駄に費やす危険性があります。

ハードウェアとソフトウェアの協調検証は、長年、複雑な ASIC デザインに使用され多大な成果を挙げてきました。そして現在、FPGA の設計者を対象とする Mentor Graphics 社の Seamless FPGA は、ロジック シミュレータとソフトウェア デバッガの両方のデバッグ生産性を高めます。Seamless FPGA を使用した協調検証は、デザインの反復ループから合成と配置・配線を除外できるうえ、ロジック シミュレーションの 1,000 倍高速に検証することができます。

デザイン反復ループの短縮

既にさまざまな開発ボードが用意されていることから、多く

の FPGA 設計者はそれらボードを反復性の高いデザイン ループに採用しています。しかしながら、開発ボードではデザインを反復するごとに多大なオーバーヘッドが生じます。このオーバーヘッドは、まずはロジック合成、その後配置・配線で発生します。これら時間のかかるステップは、最終デザインを設計するためには必要ですが、検証プラットフォームとしてシミュレーションにターゲットを絞ることで、非常に反復の多いデザイン デバッグ ループからは除外できます。

検証エンジンとしてシミュレーションを使う場合、HDL の編集から検証までの過程で唯一オーバーヘッドになるのは、HDL の比較的迅速なコンパイルのみです。あとは、前回の FPGA デザインで配置・配線を何回実行したか、また、配置・配線を実行するたびに PC に時間がかかったかを見ることにより、次のエンベデッド FPGA で節約できる時間を容易に計算できることになります。

シミュレーションの実行時間が、開発ボードの実際のスピードより遅いことは事実です。Seamless FPGA は、いくつかの革新的な手法により、エンベデッド ソフトウェアのシミュレーションを劇的に高速化します。一般的なシステムの場合、スピードアップは数桁にも及びます。

ハードウェアとソフトウェアの可視性を改善

FPGA デザインをデバッグするためには、そのデザインについて完全かつ明確な可視性が不可欠です。ハードウェアで何が起っていて、ソフトウェアが何を実行しているのかを知る

必要があります。レジスタを変更したり、信号を強制的に別の状態に変更できる必要があります。ときには時間を止めて詳しく調べなければならないこともあります。可視性が高いほど、問題の発見からバグの解決までの時間も短くなるわけです。

ハードウェアの可視性

FPGA 内部のプロープはもとより、FPGA のピン上でのプロープは一苦労です。ザイリンクスの ChipScope™ Pro アナライザを利用するとかなり楽にはなりますが、ロジック シミュレータならすべての信号を見ることができるうえ、それぞれの値を変更することも可能です。ソース HDL からコードを順番に見ていき、変数を確認したり時間を止めたりできます。詳細かつ迅速で議論の余地のない可視性という意味では、ロジック シミュレーションがベストです。

ソフトウェアの可視性

ロジック シミュレーションでは、ソフトウェアの可視性は議論の余地が残るアイテムの 1 つです。完全に機能するプロセッサ モデルを走らせればソフトウェアを実行できますが、波形だけ見てもそのプロセッサの R3 に何があるかまではわかりません。

協調検証は、ソフトウェア デバッガにリンクされている高度なプロセッサ モデルを提供します。Mentor Graphics 社の

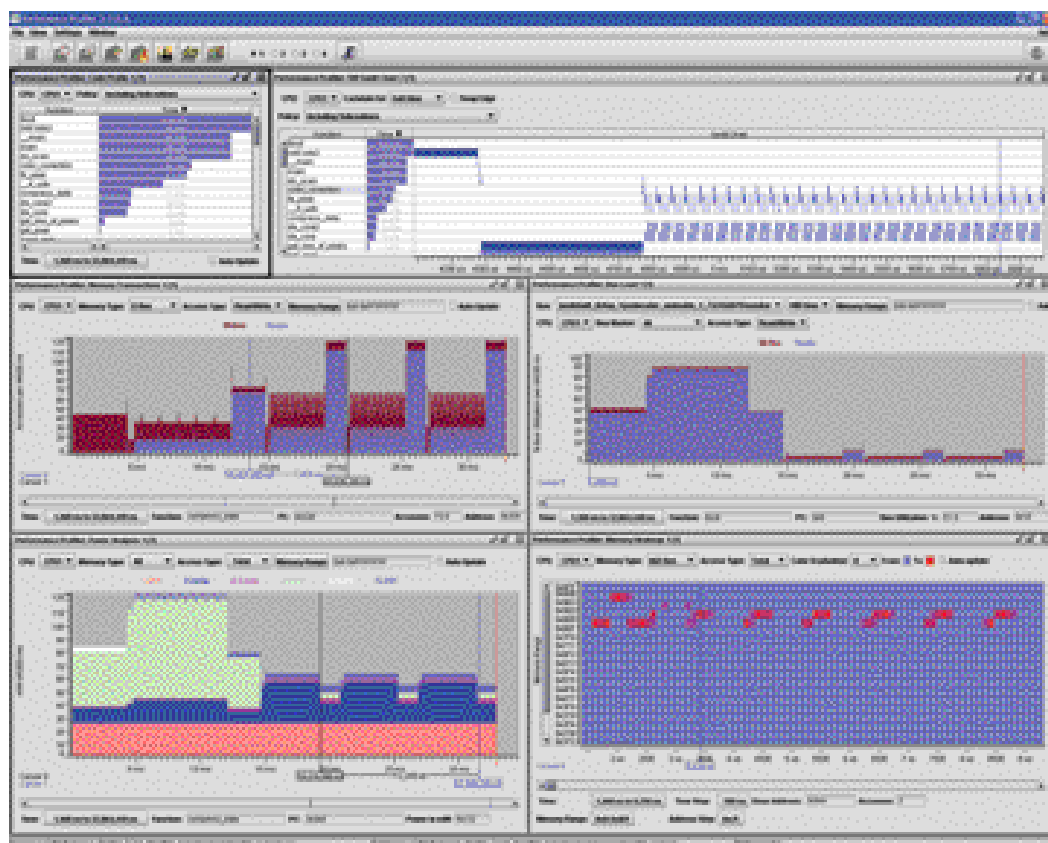
XRAY デバッガでは、レジスタからメモリ、スタック、変数にいたるまで、すべてを表示、変更できます。また、XRAY はシンボリック デバッグ用にソース コード ビューも提供します。ソース レベルやアセンブリ レベルでコードを 1 ステップずつ進めて、ブレークポイントを使って実行を停止したり強力なマクロを実行したりできます。

Accelerated Technology 社の Nucleus リアルタイムオペレーティング システム (RTOS) を使用する場合は、タスク、メールボックス、キュー、パイプ、信号、イベント、セマフォ、メモリ プールのステータスを見ることができます。

単独のロジック シミュレーションより高速

ロジック シミュレーションで、標準的なプロセッサ モデルに膨大なソフトウェアを実行させるのは得策ではありません。これは、実行時間が長すぎるためです。とはいえ、実際のところ、ソフトウェアを走らせるのが検証方法として非常に有効なことも事実です。診断、デバイスドライバ、ボード サポート パッケージ (BSP) のコードを実行し、RTOS をブートし、低水準のアプリケーション コードを実行することのメリットは計り知れません。実際にソフトウェアが使うのと同じ条件でハードウェアを検証するわけですから、効果的であることは明らかです。同様に、ソフトウェアについても、ボードが実際に構築される前に、シミュレーションにボード レベルの外部コンポーネントを含めた形で実際のデザインを対象にテストできるのです。

図 1 Seamless FPGA のシステム プロファイラ画面



これまで、実際にシステムをブートして何か処理を行うために、いかにして十分なソフトウェアを実行するかが課題となってきました。協調検証は、シミュレーション時間の大半がプロセッサからメモリまでの同じバスを何度も検査することに費やされるという単純な事実を活かすことで、実行時間を短縮化するのは、あまりに負担が大きすぎます。

ロジックシミュレーションでハードウェアのペリフェラルにアクセスすると必ずバスサイクルが発生しますが、一般に命令のフェッチとスタック操作は実行時間を高速化するためにオフロードしてかまいません。Seamless FPGA は、ロジックシミュレータでどのバスサイクルが実行され、どれが実行されないかを指定できるようにすることで、パフォーマンスをトレードオフできます。しかも、バスサイクルの指定はシミュレーション中いつでも変更できます。まずサイクルアキュレートなフルビヘイビアでリセットを実行し、その後命令フェッチとスタックアクセスをオフにしてRTOSをブートできます。

ロジックシミュレータを通してメモリにアクセスするには、いくつかのハードウェアクロックサイクルが必要です。各クロックサイクルは、FPGAの中にある膨大な量の他のロジックを

すべて通過するため、ロジックシミュレータにかなりの負担がかかります。ロジックシミュレータでバスサイクルを実行する代わりに、「バックドア」を使ってメモリの内容に直接アクセスすれば、アクセススピードが数桁高速になります。

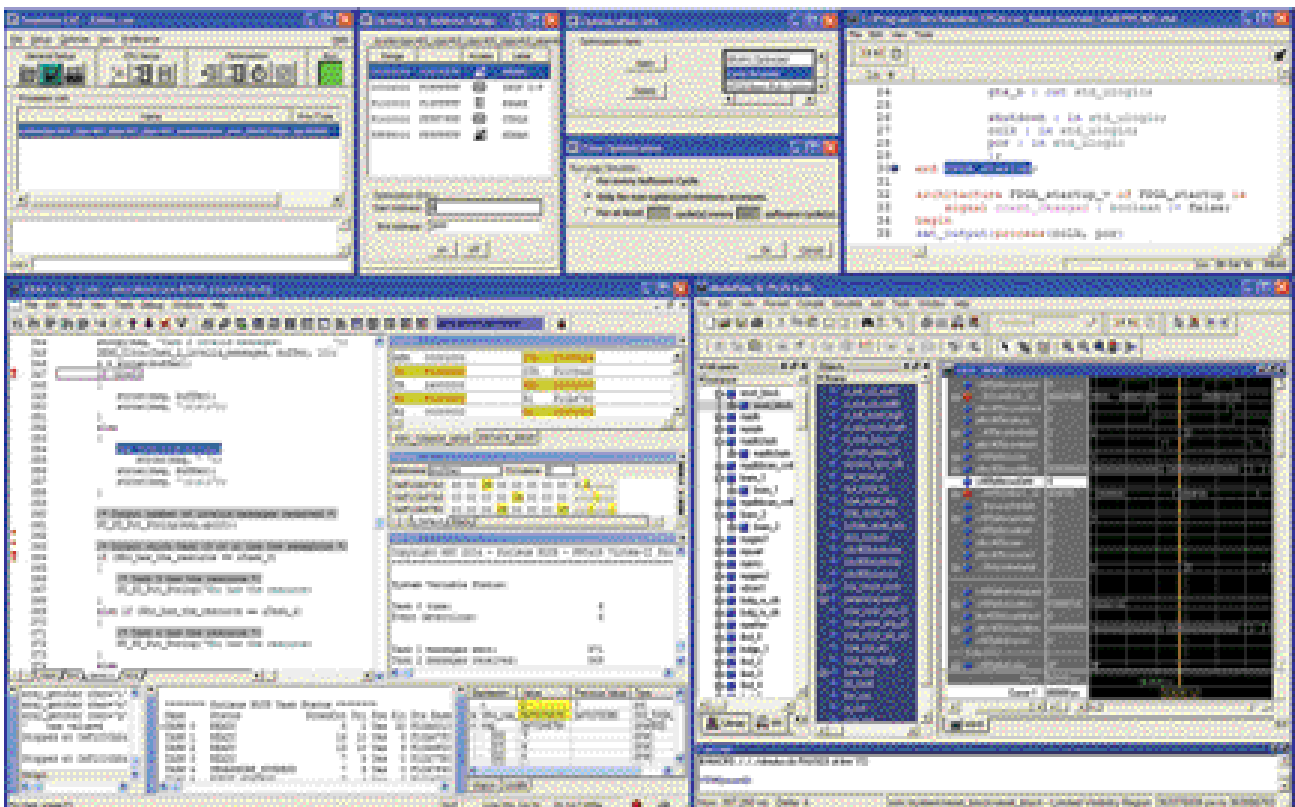
スピードアップの効果は顕著に表れます。たとえば、次のデータはザイリンクスのVirtexTM-II Pro FPGA上でNucleusを走らせているPowerPCTMを使用した典型的なデザインコンフィギュレーションから得たデータです。ロジックシミュレーション単独でNucleus RTOSをブートすると、12時間13分かかります。しかし、前述のテクニックで同じタスクを行えば、わずか6秒で完了し、7,330倍のスピードアップになるのです。

Seamless FPGAは、このテクニックを使うことで、バックドアを通してザイリンクスのブロックRAMメモリモデルやその他任意のメモリデバイスにメモリコンテンツのコヒーレントビューを維持します。したがって、DMAコントローラがメモリに、プロセッサによって後で実行される処理を落とし込んだ場合でも、すべて正常に動作します。また、プロセッサが大きなデータパケットを生成し、ハードウェアにDMAを使ってそれを伝送するよう命令したときでも、データの矛盾が生じることはありません。

プロセッサバスのボトルネックの識別

FPGAプラットフォームの性能は、デザインのメモリ構造に大きく左右されます。ブロックRAMや外部メモリでなく

図2 ザイリンクスのVirtexTM-II Pro PowerPCプロセッサ上でNucleus RTOSを走らせているSeamless FPGAでロジックModelSimとXRAYとModelSimを呼び出した画面



ソフトウェアは単独で走るわけではなく、ハードウェアと連携するため、ハードとソフトのインターフェイスをしっかり作り込むには、複数の設計チームが複数の専門分野で協力し合う必要があります。

キャッシュに入れるべきデータは何でしょうか？ どこにボトルネックがあるのでしょうか？ 他のバス マスタがより大きなプロセッサ パワーを要求しているのでしょうか？ こうした疑問を持つことは重要ですが、ハードウェア/ソフトウェア アプリケーションから実際のデータを取得しない限り、答えを導くのは困難です。

Seamless FPGA は、シミュレーションから性能データを収集してシステム プロファイラ(図 1)にグラフィカルに表示し、設計者に次のような情報を与えます。

- どのファンクションが CPU 時間をもっとも多く消費するか
- アクティビティの予想外の沈黙化、もしくはバースト
- キャッシュの効率とメモリのホット スポット
- ファンクション レベルでのコード実行と実行時間
- バス使用率とバス マスタの競合

使いやすさとインテグレーション

Seamless FPGA は使いやすく、セットアップも簡単です。Seamless FPGA は、設計者が Xilinx Platform Studio (XPS) に既に入力済みのナレッジを使い、デザインを協調検証するようそれ自体を自動的に構成します。Seamless FPGA は、ModelSim と同等の機能とユーザー インターフェイスを提供します。XRAY ソフトウェア デバッガは、step、step over、run など、多くの操作に ModelSim と同じメニュー アイコンを使います。

Seamless FPGA をセットアップするには、Xilinx Platform Studio から [File] → [Import] を選択し、XPS プロジェクト名を指定します。このインポート プロセスはすべてのセットアップ プロセスを行い、1 分ほどで ModelSim および XRAY デバッガを呼び出します。

デザインにザイリンクスのプロセッサが 2 つ以上ある場合、各プロセッサに 1 つずつ、追加のソフトウェア デバッガ ウィンドウが表示されます。

ModelSim と XRAY が呼び出されたら(図 2)、いつでもデザインを検証できます。ModelSim で、必要なスティミュラス コマンド(通常は reset と clock、およびデザイン特有のスティミュラス)を入力し、[run] をクリックします。XRAY では、[go] または [step] をクリックするとエンベデッド コードを検証していきます。デフォルトの場合、すべてのバス サイクルがハードウェア シミュレーションに向けられます。

ソフトウェアの実行速度を上げるため、3 つのアイコン選択

肢が用意されます。これらのアイコンは、ロジック シミュレータにすべてのバス サイクルを実行するよう要求せずに、Seamless FPGA にバックドアを通してメモリコンテンツにアクセスするよう指示することでソフトウェアの実行を高速化するため、「最適化(Optimizations)」というラベルが付けられています。最初のボタンは、すべての命令フェッチ サイクルに、バックドアを使うよう指示します。2 番目のボタンでは、バックドアを使うアドレス レンジの数を指定できます。アクセスがバックドアを使う場合、ソフトウェアと共にロック ステップでそのままロジック シミュレーションを続行するか、もしくはその条件を除外するか、どちらか一方を選ぶことができます。

最適化の設定は、シミュレーション セッションを実行中にいつでもすぐに変更できます。したがって、ソフトウェア内の特定ポイントまで素早く実行し、その後すべてのバス サイクルをイネーブルして詳細でサイクル アキュレートな検証を行うことも可能です。

結論

エンベデッド プロセッサを採用する大規模な FPGA デザインでは、デザインを 2、3 週間で完了するのはまず不可能です。このようなデザインは非常に複雑で、有効なシステムをスケジュールどおりに完成するにはそれに比例して複雑なバグを追跡、解決していく必要があります。

FPGA に使用するソフトウェアの内容によってシステムのコストが削減され、システム構成の自由度と機能性が増します。しかし、ソフトウェアは単独で走るわけではなく、ハードウェアと連携するため、ハードとソフトのインターフェイスをしっかり作り込むには、複数の設計チームが複数の専門分野で協力し合う必要があります。

Seamless FPGA は、生産性の高いハードウェアおよびソフトウェア デバッグ環境によりハードウェアとソフトウェアのギャップを橋渡しし、バグとパフォーマンスのボトルネックを効率的に発見することができます。いったんバグを修正したら、PC が何時間も連続で配置・配線を調べるのをじっと待つ必要なく、すぐに修正を有効にして検証できるのです。

Seamless FPGA の 30 日間無料で使える評価版は、<http://www.seamlessfpga.com/> からダウンロードが可能です。このサイトに用意されているデザイン例と Quick Start Guide を見れば、すぐに使いこなせます。詳細は、seamless_fpga@mentor.com まで電子メールでお問い合わせください。





Timing Closure with Synplify Pro Software

Synplify Pro による タイミング クロージャ

タイミング クロージャをスムーズに行うための
ツール活用のポイント

Steven Elzinga
Senior Product Applications Engineer
Xilinx, Inc
steven.elzinga@xilinx.com

図 1 Synplify Pro ソフトウェアにおけるピリオド制約

	Enabled	Clock	Frequency (MHz)	Period (ns)	Clock Group	Rise At (ns)	Fall At (ns)	Duty Cycle (%)	Route (ns)
1	☒	CLK	218	4.587	default_clkgroup_0			50	
2	☒	CLK							
3	☒								
4	☒								
5	☒								
6	☒								
7	☒								

Synplicity 社の Synplify Pro ソフトウェアは、ザイリンクスの Spartan™-3 リソースを最大限活かすパワフルな合成ツールです。クリティカル パスを識別するために ザイリンクス ISE™ ツールを使用している場合、Synplify の制約入力機能を使ってタイミング クロージャを容易に行うことができます。

また、Synplify はデザインのタイミング クロージャに便利な多数の合成のための命令を備えています。Synplify Pro ツールの各種機能を利用することにより、Spartan-3 デバイスの性能を最大限引き出すことができます。

セットアップ

本稿では、Spartan-3 XC3S50TQ144-4 をターゲットに、Synplify Pro ソフトウェア(バージョン 7.7.1)とザイリンクス ISE ソフトウェア(バージョン 6.3i)を使用しテストケースについて紹介します。最初の合成/インプリメンテーション実行と Synplify のさまざまな合成オプション、およびその制約エディタ SCOPE には、デフォルト設定をそのまま使いました。

ISE ソフトウェアの中で SCOPE にアクセスすることはできませんので、Synplify と ISE プロジェクトを別々に作成しました。Synplify Pro プロジェクトと ISE プロジェクトの両方に同じプロジェクト ディレクトリを使うことで、新規に作成した EDIF ファイルと NCF ファイルをすぐに ISE ツールで利用できるようになります。UCF と同じシンタックスを持つ NCF は、Synplify ツールを通して入力した制約から自動的に作成されます。

NCF が確実に使用されるよう、[Implementation Results] タブで Synplify Pro インプリメンテーション オプション「Write Vendor Constraint File (ベンダ制約ファイルを記述)」を選択します。ISE ソフトウェアが NCF ファイルを自動的に使うようにするため、NCF ファイルの名前を EDIF ファイル名と同じにする必要があります。

デフォルト設定の変更

Synplify ツールをデフォルト設定のままにしてデザインを実行したところ、予測周波数 163 MHz、インプリメンテーション後の実際の周波数 115 MHz という結果でした。デザインは Synplify Pro ソフトウェアにより 191 MHz に自動制約されていたため、NCF では 191 MHz ピリオド制約という結果になりました。このテスト ケースでは恣意的にデザイン目

標を 190 MHz に設定しましたので、75 MHz 改善する必要があります。

このデザインは、有限ステート マシンとして解釈できる case ステートメントを持っています。ステート マシンに対して異なるエンコード方式を探索し、最善のインプリメンテーションを決定する FSM Explorer オプションがオンに設定されています。合成ツールは、リタイミング オプションにより非同期ロジックを通してレジスタを移動し、レジスタ間の遅延分布をより均等にします。これら両方の合成オプションを選択したところ、デザインの合成予測スピードが 217.7 MHz まで上がりました。

ほとんどのデザインでは、過度な制約はかえってマイナスの結果を招きます。Synplify Pro ツールは、合成ピリオド制約を指定しないと、FSM Explorer とリタイミング オプションの選択後、デザインを 267 MHz に自動制約しました。この制約は予測していた結果よりはるかに大きかったため、SCOPE に 218 MHz の制約を使用しました(図 1)。

SDC ファイルが作成された後、このファイルをプロジェクトに追加しました。制約が追加されたことで、Synplify Pro ソフトウェアのレポートは 219.2 MHz のスピードになりました。また、制約を 220 MHz にしても、予測される結果は 219.2 MHz とまったく同じでした。

解析

ISE ソフトウェアのデフォルト設定では、164 MHz の結果を

図 2 「s_」の接続ディベティ

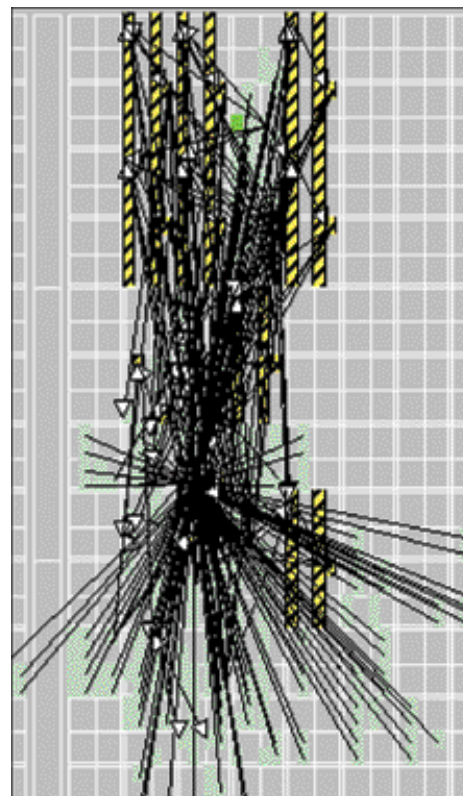


図 3 エリアグループ制約

Timing constraint: TS_CLK - PERIOD TIMEGRP "CLK" 4.646 ns HIGH 80.000000 % ;			
1669 items analyzed, 43 timing errors detected. (43 setup errors, 0 hold errors)			
Minimum period is 6.000ns.			
Slack: -1.455ns (requirement ≥ (data path + clock path skew + uncertainty))			
:			
:			
Location	Delay type	Delay(ns)	Physical Resource Logical Resource(s)
SLICE_X21Y27.YQ	Tcko	0.720	s_SYND2_inv[1] s_inv_y_259_0_dreg[1]
SLICE_X15Y15.G3	net (fanout=5)	1.731	s_SYND2_inv[1]
SLICE_X15Y18.Y	Tilo	0.550	N_423 s_mult_z_1_0[1]
SLICE_X15Y15.F4	net (fanout=1)	0.015	s_mult_z_1_0[1]/O
SLICE_X15Y18.X	Tilo	0.550	N_423 gO_x2_1_0
SLICE_X12Y16.F2	net (fanout=1)	0.552	N_423
SLICE_X12Y18.X	Tilo	0.608	s_DATA_OUT_ret_72 gO_x2_0_2
SLICE_X12Y13.BY	net (fanout=1)	1.014	s_mult_out[1]
SLICE_X12Y13.CLK	Tdick	0.260	s_LOC2_ret s_LOC2_ret
Total		6.000ns (2.688ns logic, 3.312ns route)	

達成しました。この結果は、Synplify ソフトウェアが NCF に渡したピリオド制約に基づいています。PAR エフォート レベルを High に設定すると、結果は 166.7 MHz になります。タイミング アナライザからのレポートを見ると、「s_」というインスタンスを通して共通のクリティカル パスが見えます。

フロアプランナ

ザイリンクス フロアプランナを使用すると、「s_」のコネクティビティを決めることができます。図 2 では、「s_」が選択されています(黄色)。黒い線は、デザインの他の部分に対するそのコネクティビティを表しています。

このインスタンスは一面に拡散されているため、エリア グループ制約が必要です。そこで、UCF にエリア グループ制約を入力する代わりに、SCOPE 制約を入力しました。この SCOPE 制約により、Synplify ツールはロジックの物理的配置に基づいてタイミングの決定を下すことができます。事実、合成ツールをバイパスして UCF ファイルに制約を直接入力したところ、より悪い結果になりました。

図 4 テストの解析結果。1,669 アイテム中、43 のタイミングエラーを検出

Enabled	Object Type	Object	Attribute	Value
☒	Instance	s_	xc_area_group	slice_x10y8:slice_x16y28
☒				
☒				
☒				
☒				

エリア グループ

[Attributes] タブを選択すると、SCOPE を通してエリア グループ制約を入力できます。SCOPE 内の各セルにはプルダウン メニューがあり、正しい値だけが選択できるようになっています。図 3 に示すとおり、「s_」インスタンスが見つかり、xc_area_group 制約が選択されています。

何度か繰り返したところ、ISE ソフトウェアでうまく機能する良好なエリア グループ制約が見つかり、最小ピリオド 187 MHz を達成しました。さらにこの反復プロセスを続け、異なるインスタンスにエリア グループを試したり、異なる PAR コスト テーブルを試したりしました(PAR コスト テーブルは、配置・配線プロセスに対して異なる開始点を与えます)。

最終的に「s_」内の 2 つのインスタンスにエリア グループ制約を配置し、PAR コスト テーブルの設定値として 8 を使用した結果、最後に最小ピリオド 189 MHz が得られ、当初設定した目標の 190 MHz に近づき、制約を入力しないデザイン スピード 115 MHz より大幅に改善されました。

最後に、NCF ピリオド制約として 220 MHz を使用しました。実際の制約である 190 MHz を使うと、インプリメンテーション後のデザイン性能は 189 MHz から 180 MHz に低下しました。これは PAR の正常なビヘイビアではありません。PAR で過度に制約すると合成で過度に制約するのと同じくマイナスの結果を招くため、PAR での過度なピリオド制約がプラスの影響を与えるデザインは非常にまれなケースといえます。

結論

本稿のテストケースでは、デザインがコンパクトだったため、最小の Spartan-3 デバイスに容易に収まりました。しかし、同じメソッドを使うことにより、Synplify Pro の制約とスイッチを使用してそれらの制約を ISE ソフトウェアに渡すことで、これよりずっと大きなデザインでもタイミングを満たすことができるのです。

Synplify Pro ソフトウェア、SCOPE、ザイリンクス フロアプランナでスイッチを使うことで、本テストケースのデザインは Spartan-3 デザインに設定した目標と、最適なタイミング クロージャを達成したのです。

詳細については、<http://www.synplcity.com/products/synplifypro/index.html> と <http://www.xilinx.co.jp/spartan3/> をご覧ください。



Speed FPGA Debug with Mixed-Signal Oscilloscopes

ミックスド シグナル オシロスコープを用いた FPGA の高速デバッグ

Agilent Technologies 社の MSO とザイリンクスの
ChipScope Pro アナライザによる内部測定

Joel Woodward
Senior Product Manager
Agilent Technologies
joel_woodward@agilent.com

長年、デジタル設計者はデバッグをオシロスコープに頼ってきました。FPGA がデジタル設計の主役となった今、プログラマブル ロジックを含むシステムを高速にデバッグする必要性がかつてなく強まっています。しかし、従来のオシロスコープ技術では、FPGA のファンクション デバッグに対応しきれなくなっています。そこで、FPGA を使ったシステムを開発する場合、ミックスド シグナル オシロスコープ (MSO) という新しいタイプのオシロスコープが重要な機能を提供します。

従来のオシロスコープと同様、MSO はシグナル インテグリティ、ジッタ、信号特性をパラメータ測定するための豊富な機能を備えています。MSO には、アナログ入力チャンネルが 2 つと 4 つのバージョンがあり、どちらかを選ぶことができます。また、300 MHz ~ 1 GHz まで幅広いバンド幅があり、これらの機能は信号のパラメータをチェックするために有用です。たとえば、ザイリンクスの FPGA Editor を使って I/O 規格と駆動強度を変更し、その後 MSO のスコープ チャンネルを使用して実際の I/O 特性を測定できます。

MSO と従来の DSO (デジタル ストレージ オシロスコープ) の大きな違いは、MSO には 16 のデジタル非同期サンプリング チャンネルが追加されていることです。また、デジタル チャンネルのサンプリング速度を選ぶことも可能です。デジタル チャンネルはアナログ チャンネルのメモリ ストレージから独立した大容量のメモリ ストレージを提供します。デジタル チャンネルの機能をさまざまな形で活用すれば、FPGA を使ったシステムを開発するうえで非常に役立ちます。

バスのトリガと表示

従来のオシロスコープはデジタル トリガ機能を備え、アナログ チャンネル上の各種パターンでトリガできます。4 チャンネルのオシロスコープでは、4 つの信号幅を 1 つのパターンでトリガできます。

通常、デバッグするときは、トリガ条件として特定のイベント (事象) を使ってバスを測定します。MSO のデジタル チャンネルを使用すると、16 もの信号を 1 つのデジタル パターンでトリガできます。この機能は、ステート マシンやエンベデッド マイクロコントローラ、データ バスを測定するとき大いに役立ちます。さらに、図 1 に示すように、トリガ幅を最大 20 の信号まで拡張すれば、4 つのアナログ チャンネルすべてでトリガをかけて測定値をキャプチャできます。

アナログ測定とデジタル測定の相関関係

デジタル チャンネルは厳密なデジタル測定の目的に使うことができますが、たとえばデジタル バスでトリガし、このトリガ条件でスコープ測定を行うなど、本来ファンクションとパラメータの両方に関わる問題を調べるための手段として使うのが最も効果的です。

Agilent Technologies 社のある設計チームは、開発中のエンベデッド製品に時折ソフトウェアの不具合を経験しました。この異常は週に 1 回程度、ごくまれにしか起こりませんでした。そこでソフトウェア チームは診断ソフトを開発し、この異常をもっと頻繁に起こすようにしました。診断ソフトを使用した結果、問題は FPGA に組み込まれている PCI バス上の読み出しサイクル中に起こることが判明しました。

設計チームは、PCI ステータス信号をピンから取り出し、

これら信号に MSO のデジタル チャンネルを接続しました。そして、ただちに PCI 読み出しサイクルに MSO デジタルトリガをセットアップし、MSO デジタル チャンネルが PCI バス読み出しサイクルを記録したらそれを取得（アクイジション）するよう、MSO スコープ チャンネルを設定しました。

このチームは、特定のバス サイクルでトリガする機能を使用したことにより問題をすぐに解決できました。あるクロックの立ち上がり時間が遅すぎたために、読み出しサイクルに影響を与えていたのです。さっそく、デザインに変更を加え、FPGA に新しいコンフィギュレーション ファイルをダウンロードしました。こうして再プログラム可能な FPGA と MSO の測定結果を使うことで問題を修正し、予定通りに製品を出荷できました。

広範な内部可視性

内部信号にアクセスするには、一般に信号をピンまで引き出してオシロスコープでプローブできるよう、ルートアウト（配線引き出し）の手法を使います。従来のオシロスコープだと、一度に 2 つまたは 4 つの信号にしかアクセスできません。しかし、ほとんどの問題ではもっと多くの信号を同時に見る必要があるため、2 つか 4 つという狭い可視性ではデバッグが困難です。新しい信号にアクセスするには、デザインを変更し、再合成し、新たに配置・配線を実行することで、オシロスコープが信号にアクセスできるようにする必要があります。このプロセスには何時間もかかります。

MSO のデジタル チャンネルを使うことにより、一度に 16 の FPGA 内部の信号を見ることができます。MSO のデジタル チャンネルの威力は、サイリンクスの ChipScope™ Pro ツールや Agilent Technologies 社の FPGA Dynamic Probe などのオンチップ技術と併用すればさらに強化されます。

設計チームは、ChipScope Pro アナライザを使うことで、FPGA デザインに Agilent Technologies 社のデバッグ コアを組み込むことができます。ATC2 と呼ばれるこのデバッグ コアにより、信号をピンに容易に配線できるうえ、MSO の高速セットアップが可能になり、また新しいグループの内部信号をすぐに測定できます。この機能は、16 のデジタル チャンネルの到達距離を FPGA デザイン内部まで拡張します。

MSO のデジタル チャンネルが FPGA のデバッグにいかにも有効かを理解するため、ここで簡単な通信システムの例をみてみましょう。あるステート マシンが、トランザクション ID と一緒に 16 ビット データのパケットを送出するプロセスを駆動するとします。パラレル データがシリアルライズされ、シリアル チャンネルで送られたあとデシリアルライズされ、モニタに入れます。モニタにある 2 番目のステート マシンは、パケット受信プロセスを駆動し、パケットを外部メモリに格納できるよう、データとトランザクション ID を分離します。また、このステート マシンは承認 ID (Acknowledge ID) を生成し、データが受信されたことを伝えるため送信側に送り返されます。このデザインは元々、1 つのデバッグ ポートに対して 16 ピンを割り当てていました。

ChipScope Pro コア インサータを使うと、ATC2 コアをパラメータ化することができます。コア インサータの利点として、コアの挿入を合成の後、配置・配線前に行うため、元の HDL デザインを変更する必要がありません。コア インサータを使って、アクティブな信号バンクとしてグループ化する内部信号を指定するだけでよいのです。配置・配線は元のユーザー制約 ファイルを使うため、追加の作業は必要ありません。

また、コア インサータは指定した信号の名前とグループを含む小さなファイルを生成します。このファイルは「.cdc」と呼ばれ、MSO で動作する FPGA Dynamic Probe アプリケーションにより読み込まれます。MSO に送る信号グループを変更すると、MSO は自動的にこの信号命名ファイルを使ってディスプレイ上の信号名を正しくアップデートします。新しい信号をピンに引き出すのに何時間もかかるルートアウト方式とは対照的に、FPGA Dynamic Probe を使えば内部信号の新しいグループにわずか 1 秒程度でアクセスできるのです。

高速にデバッグするためには、デザインの 4 つのセクションを見る必要があります。そこで設計者は、図 2 に示すように、16 ピンのデバッグ ポートを介して一度に 16 の信号、合計で 64 の信号にアクセスするため、4 つの信号バンクを備えるコアを作成しました。ATC2 コアは、最高 64 の信号バンクを持つようパラメータ化が可能で、MSO の 16 のデジタル チャンネルを使って 1,024 の内部信号にアクセスできます。

タイミング コア

ATC2 コアは、タイミング (非同期) もしくはステート (同期) コアのいずれかとしてコンフィギュレーションできます。MSO ではどちらのコアもサポートされます。コア インサータは、合成の後、配置・配線の前にデザインにコアを挿入します。タイミング コアを指定した場合、配置・配線ツールはプローブされている信号と出力ピンの間にフリップ フロップを配

図 1 Agilent Technologies 社のミックスド シグナル オシロスコープは、4 つのアナログ チャンネルに加え、16 のデジタル チャンネルを提供します。最大 20 の信号までトリガして表示できます。デジタル チャンネル幅は、サイリンクス FPGA の内部信号を測定するのに最適です。



置せず、信号からピンまでの測定用配線は False パスとして扱われます。これにより、配置・配線ツールは、特定の信号をピンに配線する際のスピードに関する制約を無視できるのです。

タイミング コアには JTAG コントローラが含まれていますが、このコントローラは新しい信号バンクの選択など、少ない情報交換だけに使われるため、一般に実行速度は非常に低速です(< 5 MHz)。タイミング コアを使うと、複数のクロックドメインにわたって信号を見たり、1 クロック サイクル以下の異常動作を見たりできるため非常に便利です。タイミング コアは、信号パスの間にスキューが存在することが欠点です。

ステート コア

従来のオシロスコープと MSO は、共に非同期でデータの

図 2 ChipScope Pro を使って挿入された ATC2 コアにより、各種内部信号と計測用ピンを瞬時に切り換え可能。各 ATC2 コアは、1つのシグナルバンクにも 64のシグナルバンクにもパラメタライズが可能です。

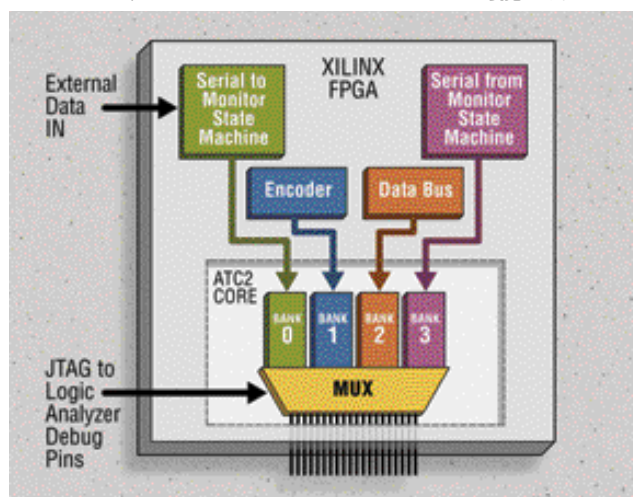
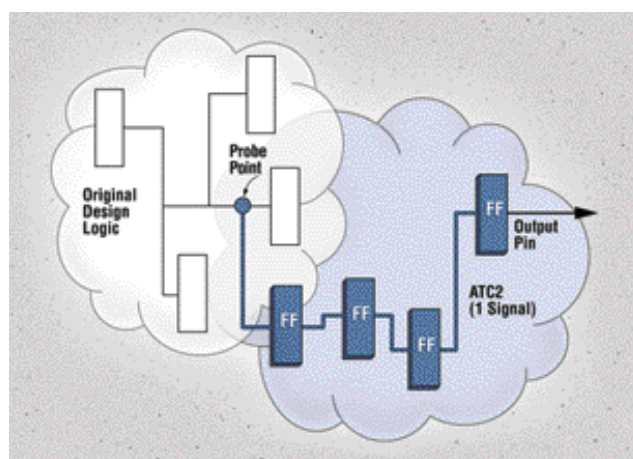


図 3 FPGA ダイナミックプローブの使用により、信号をデバッグ ポートへ送出する際に 4 段階のフリップフロップが可能になるパイプライン アーキテクチャ。この自動化のアプローチにより、配置・配線をタイミング要求に合わせて柔軟に設計できます。ルータは、ATC2 コアをチップ全体に動作させるよう単独でタイミングを取ります。



取得が可能です。サンプルは、スコープ内の調節可能なクロック リファレンスを使って格納されます。このため、計測器はクロック サイクル間の無効な遷移をキャプチャすることから、同期イベントを正確にキャプチャして解読するのは困難です。

1 つのクロックドメインにある同期情報をキャプチャするうえでより有効な方法は、ATC2 コアをステート コアとしてパラメータ化することです。ステート コアはパイプライン アーキテクチャを採用しているため、デザインのタイミングに与える影響は最小限です。プローブされている信号と出力パッドの間には合計 4 つのフリップ フロップが配置されています(図 3)。デザイン ツールは、最初のフリップ フロップをプローブされる信号のできるだけ近くに配置します。追加の 3 段階のパイプライン化により、信号は出力パッドに到達する前に 3 つのクロック サイクルの余裕ができます。ATC2 コアのパイプライン アーキテクチャにより、配置・配線ツールがそのデザインの当初のタイミング目標を満たせる確率がずっと高まるのです。

コアは同期型であるため、配置・配線ツールは信号パス間のスキューを除去します。ステート コアは、1 つのタイムドメインで働くことが欠点です。ステート コアの手法を使う場合は、複数のステート コアを挿入することで、いくつものクロックドメインにわたって測定できます。MSO は、1 つの FPGA 内、もしくは 1 つのスキャン チェーン上の複数の FPGA に分散されている複数の ATC2 コアに、一度に 1 つずつアクセスできます。

MSO のデジタル チャネルは完全に非同期のアクイジションを提供します。FPGA をデバッグする際には、最初のアクイジションが非同期で行われても、MSO に同期の測定を表示させるための方法があります。ATC2 ステート コアは、クロックに同期してクロック信号および信号ステートを出力します。MSO のデジタル チャネルは、このあらかじめフォーマットされたステート情報を取得します。その後 MSO は、クロックとして 1 つの信号を指定できるステート表示機能を使ってこの測定値を後処理します。MSO は有効なステート間でのすべての遷移以外はフィルタアウト(排除)します。これにより、FPGA 内部で同期測定を行うことができるのです。

結論

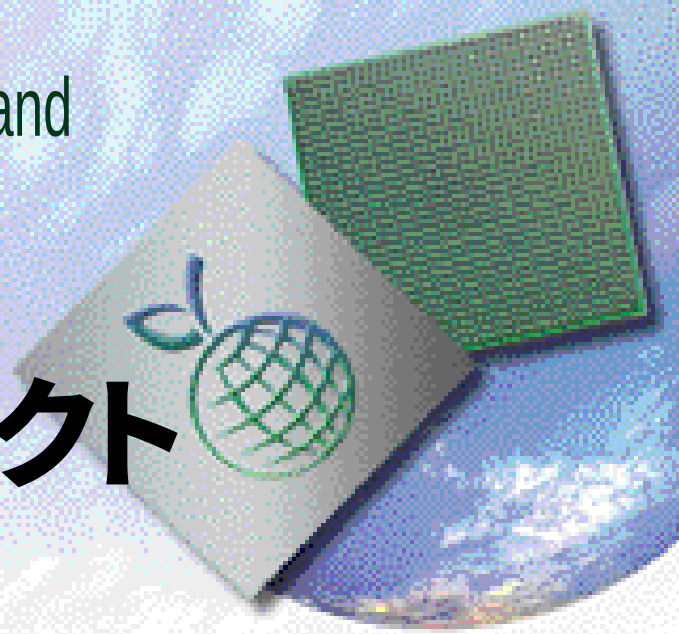
FPGA テクノロジーの再プログラム可能な特徴により、実際のデバッグを高速で反復する際、シミュレーションに大きなメリットを与えます。FPGA がいっそう複雑化するのに伴い、内部を効率的に見ることがますます重要になっています。ミックスド シグナル オシロスコープは、FPGA を使用するシステム設計のニーズにマッチする、ユニークな測定機能を備えています。MSO のデジタル測定機能を活用するのに役立つアプリケーションは、開発期間の短縮とより高品質なデザインを実現するための触媒となるのです。

Agilent Technologies 社の MSO 向け FPGA Dynamic Probe についての詳細は、<http://www.agilent.com/find/msoFPGA/> をご覧ください。



Designer Challenges for Pb-Free and Green Products

鉛フリー & グリーンプロダクトへの挑戦



ザイリンクスの鉛フリー製品を使い環境ソリューションへ容易に移行

Paula Ungs
Sr. Marketing Manager
Xilinx, Inc
paula.ungs@xilinx.com

環境面での安全性はもとより、世界中で採択されている環境関連法や OEM の要件を満たす製品をいかに設計するかが大きな課題となっています。こうした状況下で成功するためには、電子機器からの有害物質除去を求める国際的な法案とその発効日を知っておく必要があります。実務レベルでは、鉛 (Pb) フリーのデバイスがボードのデザインにどう影響するかを理解する必要があります。ザイリンクスはこれらの課題を理解しており、環境に優しい鉛フリーの製品へ容易に移行するための手法を提供しています。

世界的な環境要請

世界中に出荷できる製品を開発するには、電子機器から鉛やその他の有害物質を除去するよう求める世界各国の法案とその発効日を遵守する必要があります。図 1 に示すとおり、中でも最も注目されているのが、欧州連合 (EU) の特定有害物質規制法案 (RoHS) です。RoHS により、2006 年 7 月 1 日以降、鉛、水銀、カドミウム、六価クロム、および 2 種類の難燃剤 (PBB と PBDE) に対する同法案の規制に準拠しない電子機器は、EU 域内に輸出できなくなります。

日本は RoHS を受け入れ、リサイクリングや再利用を促す法律により独自の環境イニシアティブを立ち上げています。中国では最近、EU と同じ 2006 年 7 月 1 日までに、指定物質の規制ではなく、全面禁止という RoHS よりさらに厳しい法案を採択する予定を明らかにしました。また、米国内の各州は、上記物質に加えて他の有害物質も含めて禁止することを検討しています。

鉛フリーのボード デザイン

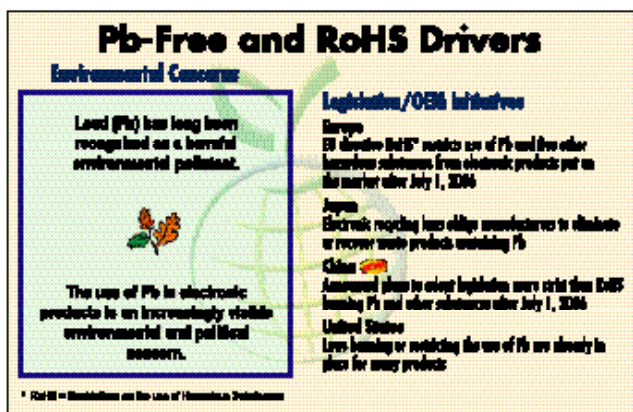
鉛フリーのソリューションはボード デザインにどのような影響を与えるのでしょうか？ 一般に、鉛が電子機器に占める割合は、半田 (75 %)、ボード (20 %)、コンポーネント (5 %) となっていますが、これがソリューション全体にどういう意味を持つかを理解する必要があります。半導体業界の「グリーン (環境に優しい)」コンポーネントには新しい材料が使われ、半田の材料とプロセスを変更する必要があるため、当然ボードの選択にも影響を与えます。しかし、ほとんどの鉛フリー パッケージは標準のパッケージと同じ形状、適合性、ファンクションを持っているため、コンポーネント サイドから見れば特別なボード レイアウトは不要です。

PQG や TQG など、リードに光沢のない Sn (錫) 仕上げを施した鉛フリーのリード フレーム パッケージは、従来の SnPb (錫鉛合金) 製造プロセスに下位互換であるため、考慮する必要がありません。しかし、この業界の多くの鉛フリー BGA パッケージ ソリューションは、従来の製造プロセスに下位互換ではない SnAgCu (錫、銀、銅の合金) 半田ボールを使用しています。SnAgCu は、標準パッケージより高いピークのリフロー温度を必要とします。このような状況では、設計者が同じボードに鉛フリーと標準コンポーネントを混在させるのは困難です。このため、ボードを鉛フリーにするかどうか事前に決めておく必要があるのです。

グリーン製品のための BOM (部材)

鉛フリーのソリューションをデザインする場合、BOM の作成が難しくなります。鉛フリー デバイスは標準デバイスとは製造プロセスが異なるため、両方を同じボードに併用できるとは限りません。したがって、ボード デザインに含めるとき

図 1 世界各国に法制化を促す環境保護の動き



は、そのデバイスが鉛フリーなのか標準デバイスなのかを知っておく必要があります。ここで注意すべきは、一部のベンダは単に標準デバイスを鉛フリーに「転換」しているだけだということです。その場合、部品番号が変わらないため、部品のタイプを容易に判別することすらできません。

ところで、BOM のコストはどうでしょうか？ 多くの設計者は、厳しい市場で競争していけるだけの費用対効果の高いソリューションを開発しようと奮闘しています。一方、多くの部品メーカは、鉛フリーの製品をコストが増加した分、高めに価格設定しています。従来も苦しかった開発環境が、予想外の高コストによってさらに圧迫されているのです。

可用性と信頼性

設計者なら誰でも、ボードにデザインした部品に高い信頼性と、いつでも入手できるという確信を求めます。ボードのプロトタイプを作成したり、量産品に組み込んだりする場合、選択した部品をすぐに入手できるという安心感を求めます。部品メーカは環境に優しい製品の市販化を急いでいるものの、その時期はまちまちのため、各部品を同時に入手できるかを予測するのは困難です。

また、鉛フリーの部品が高品質で信頼性の高いデバイスであるという確証も必要です。部材を変更するわけですから、新しいデバイスの品質が完全に認定されていることと、パッケージがより高いリフロー温度に耐えられることが不可欠です。

図 2 ザイリンクスの鉛フリー製品には、製造番号に「G」が追加されている



表 1 ザイリンクスの鉛フリー パッケージの情報

Package Type	Pb-Free Materials	RoHS Compliant?	Enhanced for Higher Temps?	Change to Form, Fit, Function?	Compatible w/SnPb Solder?
Lead Frame	100% Matte Tin Plating on Leads	Yes	Yes	No	Yes
BGA and Flip-Chip	Tin-Silver-Copper (Sn4Ag 0.5 Cu) Solder Balls	Yes	Yes	No	Not Recommended

鉛フリーへの移行

環境要請に対応するソリューションを容易に開発できるよう、ザイリンクスはあらゆる法案の施行に先駆けて RoHS 準拠の製品を提供する意向です。ザイリンクスは、電子業界に影響を与えるかもしれない新しい環境保護法を事前に把握するため、業界コンソーシアムに積極的に参加しています。ザイリンクスのグリーン コンポーネントは、世界中のあらゆる法律に準拠する予定です。

また、ザイリンクスの鉛フリー パッケージは、標準パッケージと同じ形状、適合性、ファンクションを持ちます。ザイリンクスはサプライヤと共同で、鉛フリー コンポーネントの部材を選択しました。これらの部材は業界で幅広く利用されています。ザイリンクスの鉛フリー パッケージは、リード フレームパッケージには光沢のない錫仕上げ、BGA とフリップチップパッケージには SnAgCu 半田ボールを採用しています。ザイリンクスの鉛フリー パッケージはすべて RoHS に準拠しており、そのほとんどは標準品より追加コストがかかることはありません(表 1)。

鉛フリーの製品を容易に識別できるよう、ザイリンクスは特殊な部品番号を設定しています(図 2)。この番号で鉛フリーの製品をすぐに指定、識別でき、時間と手間を節約します。さらに、ザイリンクスは「デュアルライン」戦略を採用し、ユーザー自身のペースで移行できるよう、標準製品と鉛フリー製品を並行して用意しています。

結論

ザイリンクスの鉛フリー製品でデザインする場合でも、すべての製品の品質は認定済みで、かつ標準品と同じ高い信頼性を備えているため、安心してお使いいただけます。また、新しい鉛フリーの部材に伴う高いリフロー温度の熱ストレスに耐えられるよう、パッケージの部材は強化されています。

品質と信頼性についての詳細は、ザイリンクスの Web サイト <http://www.xilinx.co.jp/quality/> にある「デバイス信頼性レポート (Device Reliability Report)」をご覧ください。また、ザイリンクスの鉛フリー製品については、<http://www.xilinx.co.jp/pbfree/> をご覧ください。

Xilinx Education Services:
Knowledge Creates Performance

知識はパフォーマンス を創出する： ザイリンクス教育サービス

FPGA デザイン コースは、必要な戦略とパフォーマンス向上
のための技術を提供する

Rhett Whatcott
Senior Engineer / Course Developer
Xilinx, Inc.
rhett.whatcott@xilinx.com

パフォーマンスの要件を満たし、かつ期日までに予算内でデザインを完成させたいというのは、すべての設計者に共通する究極の願いです。ザイリンクスの教育コースは、プロジェクトをスケジュールどおり予算内で実施するために必要な知識を提供します。

ザイリンクスの教育サービスは、デザインの専門分野ごとに6つのカリキュラム(言語、FPGA デザイン、PCI デザイン、DSP デザイン、高速デザイン、エンベデッド デザイン)を用意しています。本稿では、その中のFPGA デザインに関して、基礎知識の習得から高性能なデザイン目標達成まで、各コースの具体的な内容を紹介します。

たとえば、FPGA シリーズのコースでは、タイミングクロージャを達成するための戦略をステップバイステップで学んでいきます。この戦略には、性能改善に加え、配線の減少による低電力化という利点もあります。また、すべてのロジックコースでは、ザイリンクスのISE™ ソフトウェア、コア、アーキテクチャ上の特徴を活かすためのベスト プラクティスやヒント、テクニックを学習します。以下のコース概要では、受講者が取り組んでいく、いくつかのデザイン課題を説明します。

FPGA 設計導入コース

FPGA 設計導入では、ハードウェアとソフトウェアの基本的な特徴や、コーディングのヒント、デザイン パフォーマンス、信頼性について学習します。

再インプリメンテーションした場合、回路内のファンクションは変わるのでしょ

うか？ シンプルな非同期デザインと、それに相当する同期デザイン例を通して、回路の信頼性に関する落とし穴を避け、信頼性の高い同期回路に置き換える方法について説明します。この同期デザインテクニックによって、信頼性と安定した構造が得られると同時に、パフォーマンス改善のための基盤を整備することができます。

また、FPGA に対する初期のピン制約を作成するのに苦労している技術者には、ピン制約の作成に PACE (ピン配置とエリア制約エディタ) 使うことによって、パフォーマンスにボトルネックを招かずに FPGA ファブリックを活用できることを説明します。

複雑なコアが、あらかじめザイリンクスの FPGA 用に作成、最適化されていたらどんなに便利でしょう。このコースでは、ザイリンクスの Architecture Wizard を使い、インスタレーションするためのコアを作成します。

このコースで学ぶ内容は次のとおりです。

- ・ザイリンクスの ISE Project Navigator を使用して FPGA デザインをインプリメント
- ・該当するレポートを読み、デザイン目標が達成されたかどうかを判断
- ・PACE でピン ロケーションを割り当てパフォーマンスを向上
- ・Architecture Wizard ソフトウェアで DCM をコンフィギュレーション
- ・制約エディタを使って、グローバル タイミング制約を作成してパフォーマンスを向上
- ・ソフトウェア オプションを指定してパフォーマンスを向上
- ・同期デザイン テクニックを使い、信頼性の高いデザインを作成してパフォーマンスを改善

FPGA 設計導入コースでは、ザイリンクス FPGA を使い、信頼性の高いデザインを作成し、領域を改善して性能を高めるための基礎知識を習得します。このコースは、ロジック/デジタル デザインの基本知識と、VHDL もしくは Verilog についての基本知識を持っていることが前提条件です。

FPGA 設計実践コース

パフォーマンス目標をなかなか満たせず苦労していないでしょうか？ ソフトウェア オプションを変更しても好転しない、あるいはさらに悪化するということはないでしょうか？ FPGA 設計実践コースでは、シンプルなタイミング クロージャのフロー (図 1) と、どのオプションを使えば何が期待できるかを学びます。このコースで学習するテクニックを使えば、デザイン パフォーマンスを大幅に改善できるようになります。

では、入力信号の同期化で苦労していないでしょうか？ 1つのクロック ドメインから別のドメインにデータを交差させる必要はないでしょうか？ このコースでは、同期デザイン テクニックを説明し、パイプライン化の方法、ファンアウトのタグ信号の改善方法、同期化回路を使う方法について解説します。

図 1 タイミング クロージャのフロー チャート

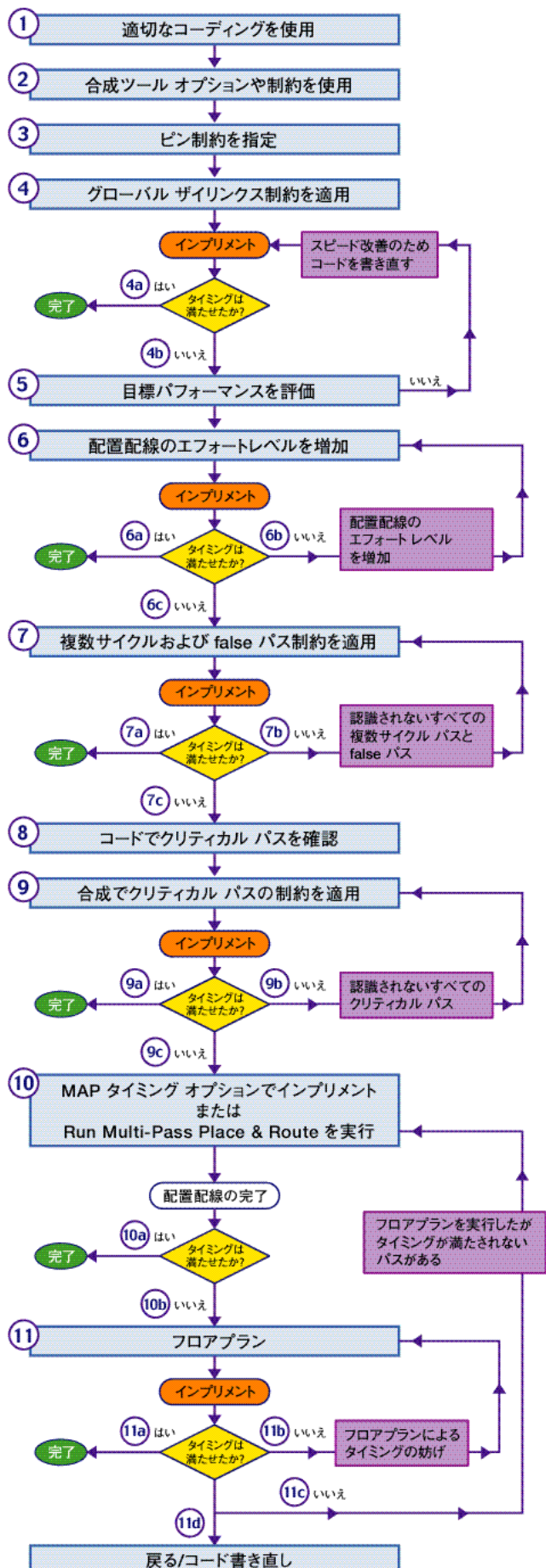
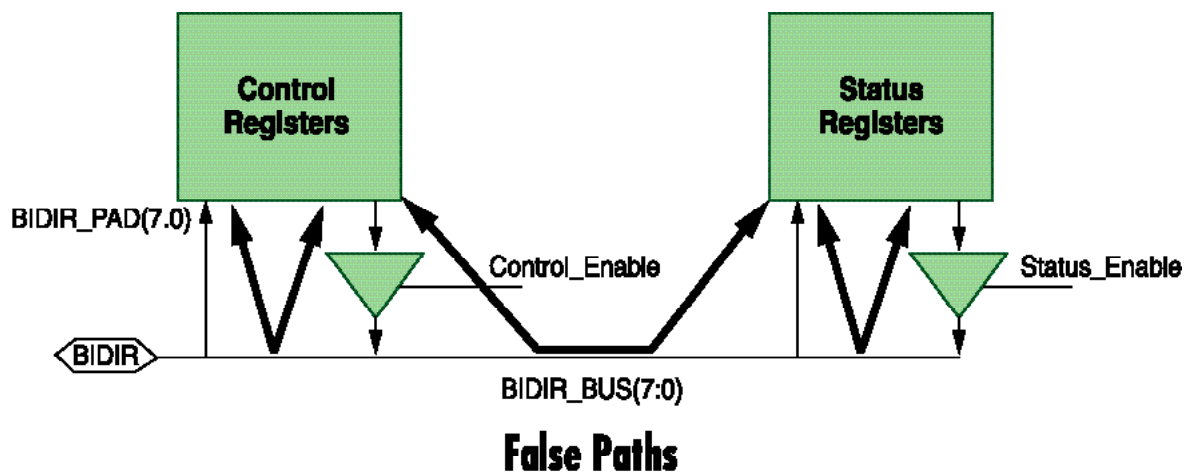


図 2 双方向パッドを経由するFalse パス



また、合成ソフトウェアの設定が目的のタスクを達成できないことはないでしょうか？ 各オプションの役割をきちんと理解しているでしょうか？ 合成ソフトウェアのオプションを活用することにより演習デザインのパフォーマンスを 20 % 改善できることを、このコースで説明します。

設計中のデザインに複数サイクル パスや False パスはないでしょうか？ 複数サイクル パスと False パスの制約をデザインに指定することにより(図 2)、デザイン パフォーマンスを高速化できることを説明します。

タイミング制約の失敗はどのように発見し、修正するにはどうするのでしょうか？ このコースでは、演習デザインのパフォーマンスを、ザイリンクスの Timing Analyzer が提供する情報を有効に使うことによって大幅に改善する方法を解説します。

結果を改善するため、無数のインプリメンテーション オプションをどのように使っているのでしょうか？ 目的とするパフォーマンス改善率は、2 %、10 %、あるいはそれ以上でしょうか？ それぞれの状況に応じて、どのインプリメンテーション オプションが最も有用かを判別していきます。

このコースで学ぶ内容は次のとおりです。

- Virtex™-II ベースのデバイス リソースを効率的に利用するための HDL コードを書く
- CORE Generator™ ソフトウェアで、カスタマイズ/最適化したコアを作成する
- タイミング エラーを突き止めるため Timing Analyzer を使い、パフォーマンス改善の戦略を立てる
- タイミング クロージャを達成するため、デザイン/コーディング テクニックとソフトウェア オプションを使う
- ザイリンクスの Constraints Editor で、グローバルおよびパス指定のタイミング制約を使ってデザインを正しく完全に制約する
- 最適なソフトウェア オプションを使うことでデザイン パフォーマンスを改善し、ソフトウェア ランタイムを短縮する

パフォーマンスを改善し、ソフトウェア ランタイムを短縮する

ここまでの 2 コースでは、差し迫ったデザイン ニーズに対処するための知識を習得していきます。より遅いスピード グレードのデバイスを使ったり、デザインをより小さなデバイスに収めたりするテクニックを学習するため、結果的にコストの節約にもつながります。FPGA 設計実践で学ぶツールとデザイン メソッドをマスターすることにより、デザインをより迅速に作成できるため、パフォーマンスの改善や開発時間の短縮、ひいてはコストの節約につながるのです。

このコースは、FPGA 設計導入コースを終了し、VHDL や Verilog の基本知識があることを前提としています。

Virtex-4 デザイン コース

FPGA 設計導入と FPGA 設計実践のコースを終了すると、次は最新かつ最も集積度の高いデバイス、Virtex-4 FPGA のデザインについて学びます。Virtex-4 ファミリのデザイン コースでは、可能な限りの最大パフォーマンス（内部周波数 500 MHz）を引き出し、また消費電力を削減するための方法を習得します。このコースは Virtex-4 FPGA ファブリックの多彩な新機能や拡張機能を取り上げ、主に演習に重点を置きます。

最新の DSP48 リソースを数値演算にどのように利用するのか、また、DSP アプリケーションにはどう使うのでしょうか？ 演習では、500 MHz の動作を必要とする DSP および数値演算アプリケーションに DSP48 を使用します。このリソースは、6：1 マルチプレクサなどのロジック リソースをインプリメントするためにも使用することが可能です。Virtex-4 ファミリのデザイン コースでは、このダイナミックで高性能、低消費電力のリソースを活用するために必要な情報を学習します。

10 個のグローバル クロックと 6 個のリージョナル クロックを備えたクロックをデザインするにはどうしたらいいのでしょうか？ 演習では、Virtex-4 デバイスで複雑なクロックをデ

ザインするため、最新の Xesium クロッキング リソース (図 3) を使用します。具体的には、拡張 DCM、新しいフェーズ マッチド クロック ディバイダ (PMCD)、拡張グローバル クロック バッファ (BUFGCTRL)、最新のリージョナル クロッキング リソース (BUFIO と BUFR) を利用します。

ブロック RAM のパフォーマンス改善や専用 FIFO リソースについて耳にしたことはありますか？ この演習では、新しいオプションの出力レジスタを採用する 500 MHz のブロック RAM コアを作成します。また、新しい専用の FIFO16 リソースを利用するコアも作成します。

ソース同期インターフェイスをデザインする必要はないでしょうか？ 新しい Virtex-4 IOB タイルには、ISERDES および OSERDES リソースが含まれています。このコースでは、独自のソース同期インターフェイスを作成するため、新しい Xesium クロッキング リソースと ISERDES/OSERDES リソースを使う方法を学習します。また、ChipSync ウィザードと、ソース同期インターフェイスを作成するためのメモリ インターフェイス ジェネレータ (MIG) についても学びます。

このコースで学ぶ内容は次のとおりです。

- Xesium クロッキング ソースを使用する
- DCM の周波数合成出力 (CLKFX) とデジタル ファイン フェーズ シフト (DPS) を動的にリコンフィギュレーションする
- 新しい PMCD を使い、位置が一致した分周クロックを作成する

- ChipSync ウィザードまたはメモリ インターフェイス ジェネレータ (MIG) を使い、カスタマイズしたソース同期インターフェイス コアを作成する
- 新しい Virtex-4 ブロック RAM と FIFO16 を使い、メモリ リソースのパフォーマンスを高める
- DSP48 リソースを使い、数値演算回路と DSP 回路のパフォーマンスを高め、消費電力を削減する

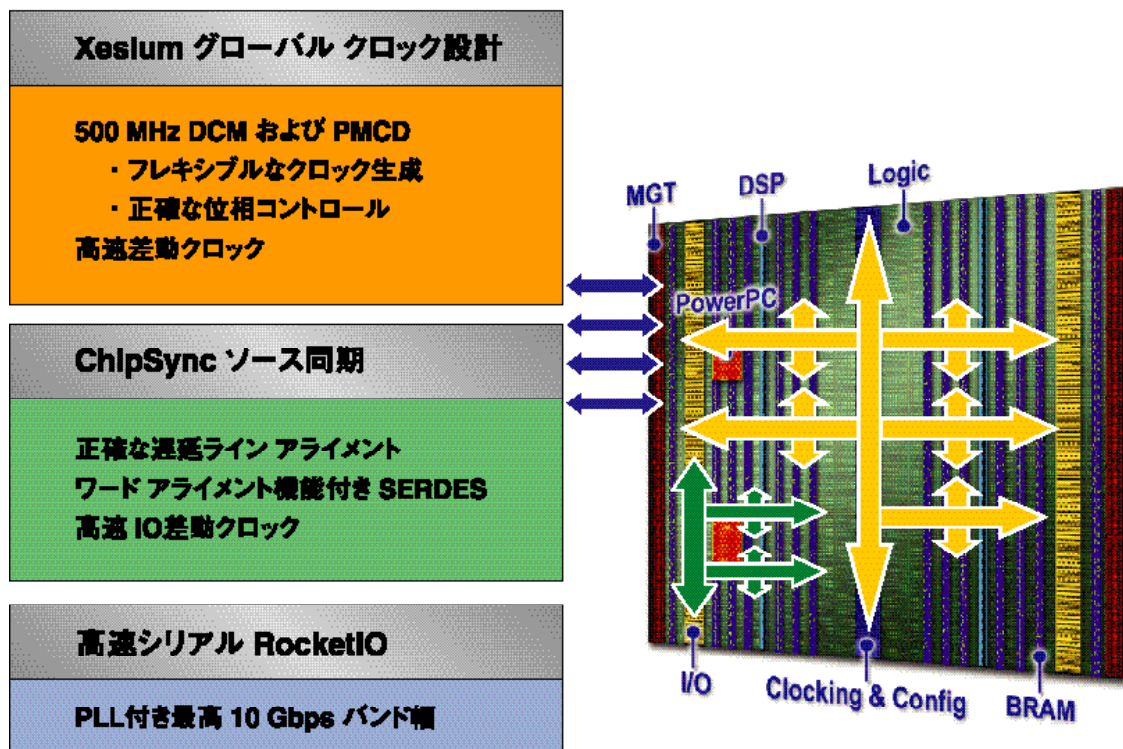
このコースを受講すると、Virtex-4 FPGA ファミリのパフォーマンス改善/低電力化機能をフルに活かすことができますようになります。このコースを受講するには、VHDL または Verilog について中級レベルの知識を習得していることに加え、FPGA 設計導入と FPGA 設計実践の各コースを終了していることが前提条件です。

アドバンスド FPGA 設計コース

アドバンスド FPGA 設計コースでは、ここまでの 3 コースの知識を基に、ISE ツールとザイリンクスのハードウェアの最も高度なファンクションに挑戦します。このコースでは、よりよい結果の達成やより迅速な変更、また、その結果を維持するためのさまざまなテーマについて解説します。受講時間の半分は演習にあて、体験を通して知識とスキルを高めていきます。

デザインに多少の変更を加えるたびに、そのデザインを完全にインプリメントし直す必要が生じて、イライラすることはな

図 3 Xesiumクロッキングリソース



いでしょうか？ 合成とインプリメンテーションに時間がかかりすぎることはないでしょうか？ タイミングの結果が変わることはないでしょうか？ 演習では、インクリメンタルなデザイン変更でインクリメンタル デザイン テクニック(図 4)を使用し、それまでのタイミング結果を維持します。これにより、反復的な合成とインプリメンテーションのランタイムを 50 % 短縮できます。

インプリメンテーション レイアウトを作成する準備はできていますでしょうか？ 演習では、デザイン パフォーマンスを高めるため、フロアプランナあるいは PACE を使ってデザイン レイアウトを作成します。また、これらのツールはインクリメンタルなデザイン テクニックを採用するデザインにも使われます。

8 個以上のクロックを使うデザインはないでしょうか？ Virtex-II ベースのデバイスにはたくさんのクロック機能がありますが、それらの影響についても考える必要があります。演習では、用意されている全機能を利用し、Virtex-II デバイスに複雑なマルチクロックをデザインする方法について、ステップ バイ ステップでわかりやすく解説します。Virtex-4 のクロック リソースについても説明します。

デザイン内の特定パスでタイミングを満たすのに苦労していないでしょうか？ このコースでは、問題となっているタイミング パスに対し、いつどのように相対配置マクロ (RPM) を作成すればよいかを理解できるようになります。RPM によ

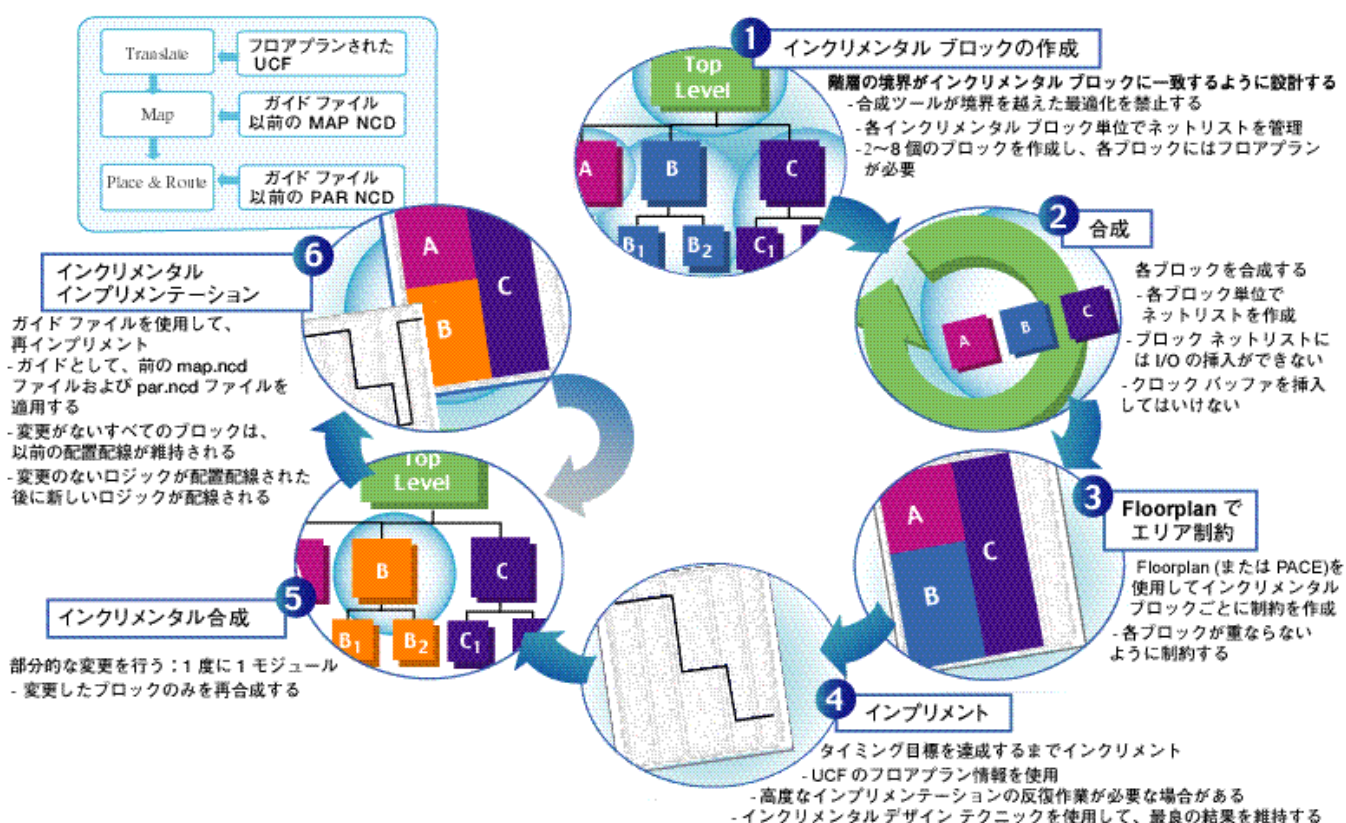
り、含まれている各エレメントに対して予測可能なパフォーマンス結果が得られます。演習では、RPM を使用してクリティカル パスに対するパフォーマンス目標を満たせることを確認します。

GUI よりスクリプトを使いたいとは思いませんか？ このコースで紹介するテクニックや戦略、オプションを使うことにより、パフォーマンス改善に最も有効なスイッチの使い方を習得できます。また、スクリプト作成の演習では、各種の設定とオプションを使ってデザイン パフォーマンスを大幅に改善します。

ここでは次のことを学びます。

- ・パフォーマンスを高めるためタイミング制約と配置制約を作成、編集する
- ・RPM を作成し、クリティカル パスのパフォーマンスを改善すると共に、複雑なファンクションに予想可能なタイミング結果を達成する
- ・Virtex-II と Spartan™-3 FPGA に対して効率的にクロックをデザインする
- ・設計サイクルを短縮しパフォーマンス結果を維持するため、インクリメンタル デザイン テクニックを使う
- ・より効率的なインサーキット テストのため、FPGA Editor でインプリメントしたデザインを簡単に手直しする

図 4 インクリメンタルなデザイン テクニック/デザインフロー



- ・パフォーマンスを高め、エリアを削減するため、スクリプトとソフトウェア オプションを使用する
- ・最適なパフォーマンスを達成するため体系的なタイミングクロージャ フローを使う

このコースの目標は、経験豊富な設計者に、画期的なパフォーマンスを得るためのテクニック、オプション、戦略を提供することです。パフォーマンスの目標を達成したら、1つのイタレーションから次のものに移行した場合でもそのパフォーマンスを保つ方法をお教えします。

ユーザーの設計目標は、私たちの教育目標でもあります。このクラスは、特定基準を満たすと認定された豊富な経験を持つインストラクタやデザイン エンジニアによって実施されます。講師は参加者一人ひとりが学習目的を達成できるよう、教育内容とディスカッションを受講者のニーズに合わせて調整します。このコースは、FPGA 設計導入と FPGA 設計実践の各コースを終了していることを前提とします。Verilog または VHDL について中級程度の知識があることと、サイリンクス FPGA でのデザイン経験が少なくとも 6 カ月あることが条

件です。

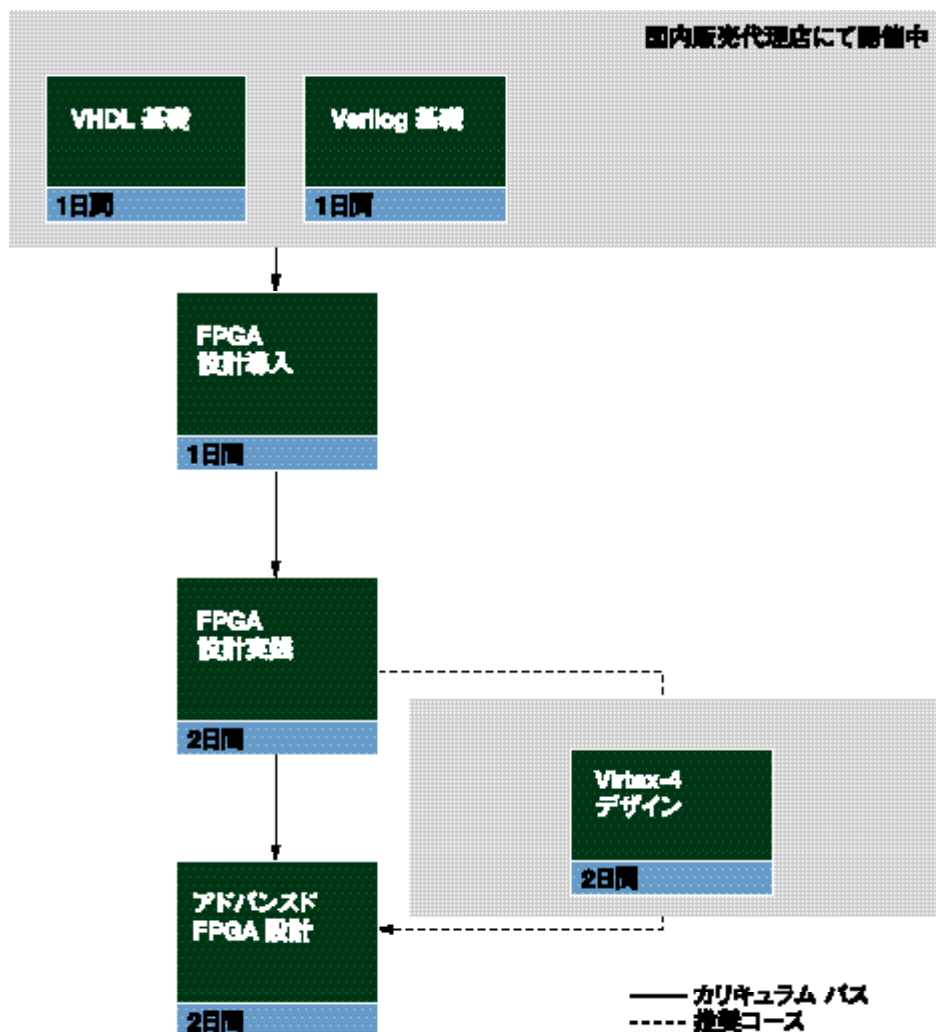
結論

教育サービスの目標は、ユーザーができるだけ短期間でデザイン目標を達成できるよう支援することです。サイリンクスは、ユーザーの複雑なデザインや目標に応えられる、業界で最も包括的なコースを提供します。ソフトウェアやハードウェアの戦略的な正しい使い方をすぐに習得しなくても、数日、あるいは数週間かけながらコツコツと目標達成に向けて努力することも可能ですが、最初からプロのやり方を学ぶことにより、時間とコストを節約することができます。

世界中で提供されているトレーニング コースの一覧や、自分に必要なコースが何かを知るためのスキル チェックについては、<http://www.xilinx.co.jp/education/> をご覧いただくか、Web サイトの「製品とサービス」より「トレーニング」リンクをクリックしてください(図 5 は FPGA デザインのカリキュラムです)。サイリンクスのデバイスを使用する際の助けができればと願っています。



図 5 FPGA デザインコースのカリキュラム パス





低インピーダンスと大容量を実現する注目のデバイス「プロードライザ」

FPGA の高性能化・多機能化・大規模化に対応する
デカップリング回路に最適なデバイス

NECトーキン株式会社
エネルギーデバイス事業本部
ソリューション技術部
坪田一成

はじめに

近年、半導体の 1 チップ化を背景に、FPGA の高性能化・多機能化・大規模化が急速に進んでいます。中には、ザイリンクスの Virtex-4 ファミリのよう PowerPC プロセッサを搭載する製品も登場しています。そして、FPGA に搭載されるトランジスタ数は増加の一途をたどり、そこで使われる電流も増加の傾向にあります。このため、FPGA の低消費電力化が大きな課題となっており、90 nm ルールで製造される FPGA の動作電圧は、1 V 程度まで低電圧化が進んでいます。

このような動向の中で、FPGA の電圧を安定化する電源への要求も非常に厳しくなっており、最近では、POL 電源が推奨されています。さらに、POL と FPGA の間を取り持つデカップリング回路に対する要求も高度になっており、デカップリング回路から FPGA の所要電流を供給するためには、デカッ

プリング回路自体が低インピーダンスであることが要求されています。

NECトーキンではこうした時代の要請に応えるため、デカップリング回路に最適なデバイスとして、広い周波数帯域にわたって極めて低いインピーダンス特性を実現するデバイス「プロードライザ」を開発し、発表しました。

現在の製品ラインナップ

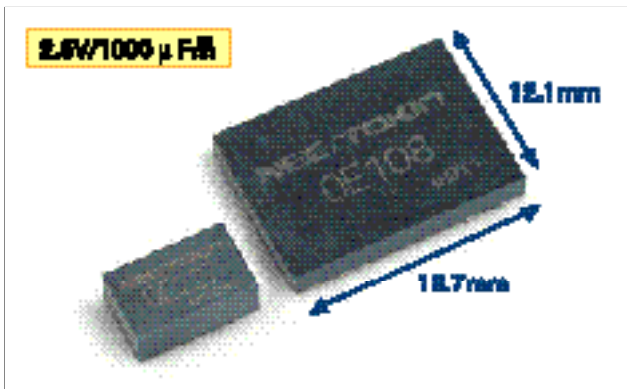
表 1 は、プロードライザの製品ラインナップです。写真 1 に、プロードライザの外観を示しました。NECトーキンではこのように、最新の半導体における低電圧化とトランジスタ数の増大化による大電流化に対応するため、低インピーダンス化と大容量化を積極的に展開しています。D ケース(8.5 mm × 5.3 mm × 3.5 mm)形状の 33 μ F 品と 100 μ F 品が、FPGA の

表 1 プロードライザの製品ラインナップ

型 番	形 状 (mm)	定格電圧 (V)	静電容量 (μ F)	ESL1 (pH) ※1
PFAD350E336M (※2)	8.5× 5.3×3.5	2.5	33	10
PFAD350E107M (※2)	8.5× 5.3×3.5	2.5	100	10
PFAD350E227M (※2)	8.5× 5.3×3.5	2.5	220	10
PFAF250E257M	16.7×12.1×2.5	2.5	250	1
PFAF250E108M	16.7×12.1×2.5	2.5	1000	1

※1 ESL1 : S21より相当するインダクタンス値に換算した代表値 ※2 開発中

写真 1 ブロードライザの外観



デカップリング回路用を想定したブロードライザです。F ケース形状の 1000 μ F 品は、ノート PC やサーバーなどの CPU のデカップリング回路で使用されることを想定したブロードライザです。いずれも製品の実装性を大幅に向上することを可能にしています。

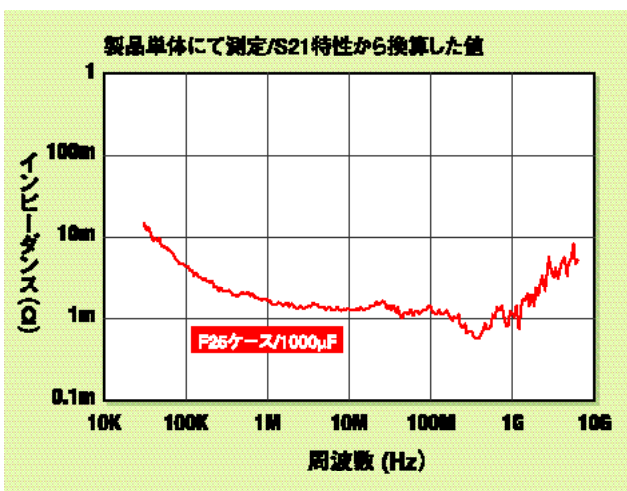
ブロードライザの特性

ブロードライザは、電極の部分を伝送線路構造としたことにより、周波数に依存しない一定の特性インピーダンスを有する構造となっています。さらに、大容量かつ低抵抗の材料を採用したことにより、特性インピーダンスを数 $m\Omega$ の大きさに抑えることが可能になり、広い周波数帯域にわたって極めて低いインピーダンス特性を実現しています。

図 1 に、1000 μ F のインピーダンス特性を示しました。

従来のコンデンサにおけるインピーダンス特性は、図 2 のように、低インピーダンスである周波数領域が自己共振周波数の近傍に限定されます。そのため、広い周波数帯域で低インピーダンスなデカップリング回路を実現するには、自己共振周波数の異なるコンデンサを多数組み合わせる必要がありました。

図 1 ブロードライザのインピーダンス特性



これに対し、ブロードライザのインピーダンス特性は、図 1 にあるように 1 個で広い周波数帯域をカバーし低インピーダンスの特性を実現するため、従来、複数のコンデンサの組み合わせにより実現していたデカップリング回路をブロードライザのみで実現でき、大幅な設計の簡略化と部品点数の削減、および実装基板の小型化が可能になりました。

ノート PC におけるデカップリング回路への応用

現在、ノート PC で汎用的に使用されている CPU のデカップリング回路にブロードライザを適用し、部品点数の削減効果を確認しました。使用した CPU は、クロック周波数 1.6 GHz、コア電圧 0.844 ~ 1.356 V、最大消費電流 3.2 A というスペックです。

この CPU のコア電圧に対するデカップリング回路を従来のコンデンサで構成すると、一例ですが、チップタイプのアルミ固体電解コンデンサ 220 μ F が 4 個、チップタイプの積層セラミックコンデンサ 10 μ F が 35 個の計 39 個ものコンデンサが必要になります。

写真 2 は、これらのコンデンサを並べた従来型のデカップリング回路と、コンデンサの代わりにブロードライザ 1000 μ F 1 個で置き換えたものです。ブロードライザ 1 個でコア電圧のデカップリング回路を構成したノート PC は問題なく動作しており、39 個のコンデンサを 1 個のブロードライザで置き換えられることが確認できました。

FPGA の応用例

最近の FPGA は、10 Gb/s のシリアルトランシーバを搭載しているため、高周波のノイズによるシグナル インテグリティの劣化が大きな問題になっています。

このような高周波ノイズに対しブロードライザがどのような効果をもたらすか、シミュレーションで検証した例があります。対象としたシステムは、大規模な FPGA を想定し、FPGA

図 2 コンデンサのインピーダンス特性

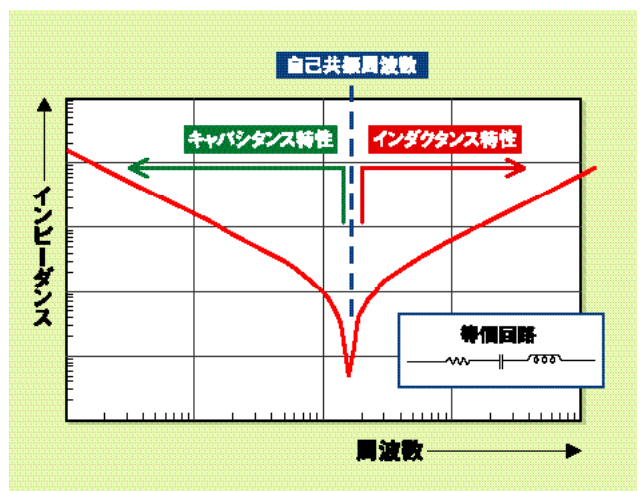
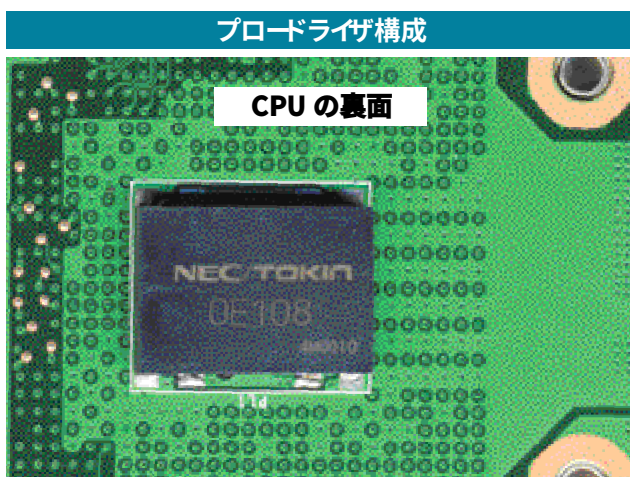


写真 2 デカップリング部の外観

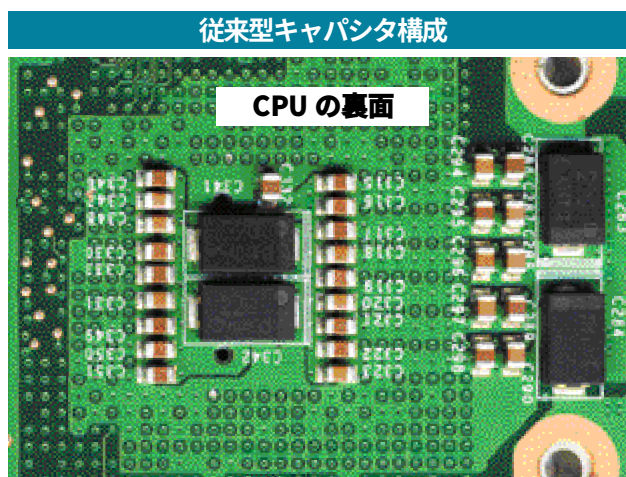
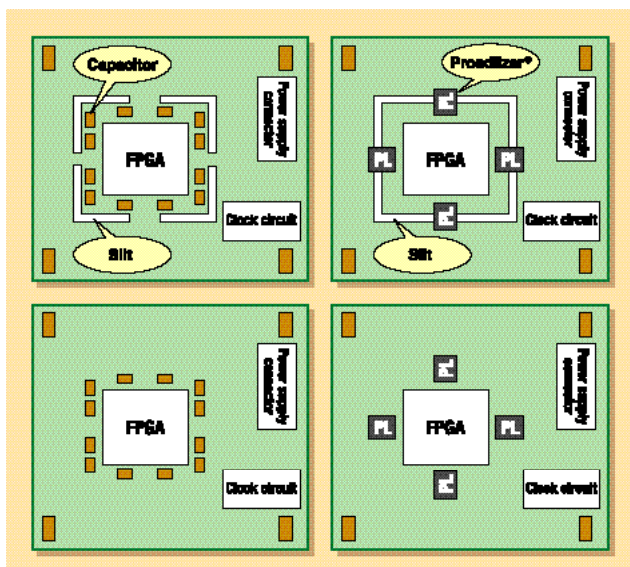


ブロードライザ 1個

と電源間のスリットによって電源の Vcc 層を完全に分離し、電源の供給をブロードライザを通して供給するようにした設計です。基板上では、ブロードライザの代わりに通常のコンデンサを配置したものとブロードライザを配置したものの2通りとし、それぞれ、スリットを入れたものと入れないものを設計しました。

通常のコンデンサの場合、スリットで完全に分離すると電源が供給できないので、ブロードライザを搭載する位置と同じ位置のスリットを除いて Vcc 層を接続しています。図 3 が、基板上の部品とスリットの位置のイメージです。そして、この基板を SPEED 2000 というシミュレータでシミュレーションした結果が図 4 になります。図 4 から、スリットとブロードライザの組合せによる EMI への効果が明らかでしょう。今後は、ザイリンクスの FPGA を搭載したボードで効果を検証する予定です。

図 3 FPGA 応用例



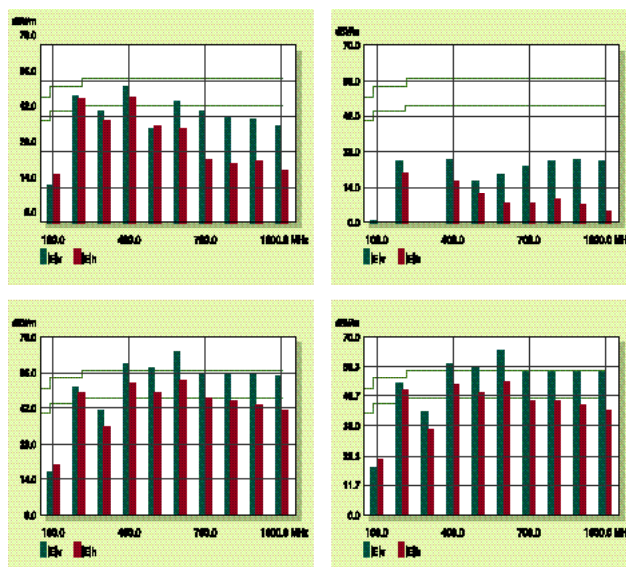
コンデンサ 39個

NECトーキンでは最近、コンデンサのような受動部品を 1 つにできるという期待を込めて、ブロードライザのようなデバイスを総称して「IPC: Integrated Passive Component」と呼び始めています。そして「IPC」という言葉が、LSI や CPU のように、広く一般に理解していただけるよう「IPC」の商品群を拡大したいと考えています。

CORPORATE PROFILE

設立：1938 (昭和 13) 年 4 月 8 日
本店：宮城県仙台市
資本金：129 億 9,021 万 2,211 円
売上高：1,194 億円 (2004 年度)
従業員数：単独／1,567 名、
連結／国内 1,553 名、海外 1 万 6,013 名
<http://www.nec-tokin.com/>

図 4 シミュレーションの結果：EMI



ザイリンクス イベント カレンダー

ザイリンクスは、年間を通じて多数のトレードショーやイベントに参加しています。これらのイベントは、ザイリンクスのシリコンやソフトウェアの専門家がお客様からの質問にお答えしたり、最新製品やザイリンクスのカスタマのサクセスストーリーをご紹介する機会です。

10~12月

ザイリンクス ウェブ セミナ



◇好評配信中

第1回 Virtex™- 4 シグナル インテグリティ

高速インターフェイスをご使用になるエンジニアおよびシグナル インテグリティでお困りの方を対象に、その理論と Virtex-4 のシグナル インテグリティ特性を競合デバイスとの実測比較データを使いながらご紹介いたします。

第2回 Virtex™- 4 ローパワー・アドバンテージ

システム設計時における消費電力の低減の重要性と、Virtex-4 マルチプラットフォーム FPGA の低消費電力の優位性とテクノロジーについてご紹介いたします。また、競合デバイスとの比較により具体的に何が優れているかを実測データを使用して解説いたします。

第3回 Virtex™- 4 メモリインターフェイス・アドバンテージ

パワフルな高速メモリインターフェイスの構築を容易にする Virtex-4 は、すべてのプラットフォームにコンフィギャブルなハイパフォーマンス SelectIO™ テクノロジーが搭載され、多様な I/O 規格に対応しています。本セミナーでは、DDR2 を中心に Virtex-4 を用いたメモリ インターフェイス設計の利点や関連する開発ボード、リファレンス デザインをご紹介します。

◇所要時間：各回約25分

◇URL：<http://www.xilinx.co.jp/webseminar/>

11月 1日 ~ 2日 NE Embedded Processor Symposium 2005

ザイリンクスが講演と展示を行います。

◇主 催：日経エレクトロニクス誌

◇会 場：品川 御殿山ヒルズ

◇ザイリンクス講演：
XILINX組込みCPUを採用したSoftware Accelerationの実現

◇講演者：Tim Erjavec (プログラマブルシステム マーケティング、シニア マネージャ)

◇URL：<http://techon.nikkeibp.co.jp/seminar/eps05/>

11月16日 ~ 17日 Embedded Technology 2005

ザイリンクスが講演と展示を行います。

◇主 催：社団法人日本システムハウス協会

◇会 場：パシフィコ横浜

◇URL：<http://www.jasa.or.jp/et/>

12月 6日 ~ 7日 MATLAB EXPO 2005

ザイリンクスが講演と展示を行います。

◇主 催：サイバネットシステム株式会社

◇会 場：東京プリンスホテル パークタワー

◇URL：<http://www.cybernet.co.jp/matlab/>

12月14日 ~ 16日 TRONSHOW 2006

ザイリンクスが展示を行います。

◇主 催：T-Engineフォーラム/社団法人トロン協会

◇会 場：東京国際フォーラム

◇URL：<http://www.tron.org/show.html>

ザイリンクス販売代理店イベント

日 程	イベント内容	URL	開催地
10月28日	メメック ジャパン株式会社主催「FPGA セミナー」	http://www.memec.co.jp/	福 岡
11月 2日	FPGA コンソーシアム主催「5 都市 FPGA カンファレンス」 スポンサーとして以下の代理店が出展 ・東京エレクトロニクス株式会社 ・メメック ジャパン株式会社	http://www.fpga.or.jp/5city.html	札 幌
11月 9日	FPGA コンソーシアム主催「5 都市 FPGA カンファレンス」 スポンサーとして以下の代理店が出展 ・東京エレクトロニクス株式会社 ・メメック ジャパン株式会社	http://www.fpga.or.jp/5city.html	名古屋
11月16日 ~18日	社団法人日本システムハウス協会主催「Embedded Technology 2005」 ・東京エレクトロニクス株式会社がザイリンクスと協同出展 ・メメック ジャパン株式会社がDAFSエリアにて出展	http://www.jasa.or.jp/et/	東 京
11月25日	FPGA コンソーシアム主催「5 都市 FPGA カンファレンス」 スポンサーとして以下の代理店が出展 ・東京エレクトロニクス株式会社 ・メメック ジャパン株式会社	http://www.fpga.or.jp/5city.html	大 阪
11月29日	メメック ジャパン株式会社主催「FPGA セットアップセミナー」	http://www.memec.co.jp/	福 岡
12月 9日	FPGA コンソーシアム主催「5 都市 FPGA カンファレンス」 スポンサーとして以下の代理店が出展 ・東京エレクトロニクス株式会社 ・メメック ジャパン株式会社	http://www.fpga.or.jp/5city.html	福 岡
12月14日 ~16日	TRONSHOW 2006 新光商事株式会社がザイリンクスと協同出展	http://www.tron.org/show.html	東 京



編集長 Carlis Collins
editor@xilinx.com

編集 Forrest Couch
Forrest.couch@xilinx.com

編集アシスタント Charmaine Cooper Hussin

Xcell オンライン
編集担当 Tom Pyles
Tom.pyles@xilinx.com

アートディレクション Scott Blair



Xcell 54 号

2005 年 10 月 17 日発行

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400

ザイリンクス株式会社
〒163-1118
東京都新宿区西新宿 6-22-1
新宿スクエアタワー 18 階

© 2005 Xilinx, Inc. All Right Reserved.

XILINX や、Xcell のロゴ、その他本書に記載の商標は、米国 Xilinx, Inc. の登録商標です。PowerPC は、米国またはその他の国における IBM 社の商標です。ほかすべての名前は、各社の登録商標または商標です。

本書は、米国 Xilinx, Inc. が発行する英文季刊誌を、ザイリンクス株式会社が日本語に翻訳して発行したものです。

米国 Xilinx, Inc. および Xilinx, Inc. 株式会社は、本書に記載されたデータの使用に起因する第三者の特許権、他の権利、損害における一切の責任を負いません。

本書の一部または全部の無断転載、複写は、著作権法に基づき固く禁じます。

Printed in Japan

人生はチャレンジ： チャンスと見るか、 災難と見るか

「人生はチャレンジの連続だ」という言葉を、よく耳にします。しかし、自分自身を知り、経験を通してさらに成長するため、本気でチャレンジしたことはどれだけあるでしょうか？

皆様の中には、チャレンジとは成功と幸福までの道のりで突然出くわす、できれば避けたい障害物のようなものだとお考えの方もいるでしょう。しかし、本当にそうでしょうか？

私の個人的な経験では、チャレンジとは次の段階に移行するための大きなチャンスです。私生活でも仕事でも、今まで気づけなかった盲点を明らかにし、改善の動機を与え、どこを変えればより成長していけるのかを教えてください。

ザイリンクスは、今日のシステム設計者が直面しているデザイン チャレンジをしっかりと認識し、市場ニーズに応えるには何をどう変えるべきか常に模索しています。また、私たち自身にも盲点となっている改善点がないか、お客様の要望にも常に耳を傾けています。

本誌 Xcell Journal 54 号では、デザイン チャレンジにスポットを当て、システム パフォーマンス、総コスト、消費電力、コネクティビティ、デバッグに関する記事をお届けします。

ここで、いくつかご紹介しましょう。「Virtex-4 による記録的パフォーマンスの達成 (Achieve Breakthrough Performance in Your System)」(P.14) では、Virtex™-4 FPGA がいかにして消費電力の低減と卓越したシグナル インテグリティを達成しながら、記録的なシステム パフォーマンスの達成を実現したかを説明しています。

総コストの削減については、低コストのシステム インプリメンテーションの実例として、CPLD を使用して QWERTY キーボードをインプリメントする方法や、先進のテレマティクスでセーフティドライブとコネクティビティを同時に実現する方法を紹介しています。

消費電力に関する記事「パフォーマンスか消費電力か：その最適解を得る (Performance vs. Power : Getting the Best of Both Worlds)」(P.37) では、過度な電力が多くの点でコスト高になることを解説しています。過度な電力は、ヒート シンクからファン、高性能な熱交換器にいたるまで、デザインおよび動作上の特別な配慮を必要とします。より大型の電源を採用するコストも考えなくてはなりません。

ザイリンクスの IP Solutions Division に所属する数名のエンジニアが共同執筆したコネクティビティの記事「システム パケット インターフェイス間のブリッジング (Bridging System Packet Interfaces)」(P.60) では、ザイリンクスの IP およびリファレンス デザインがプロトコル間にカスタム ブリッジング ソリューションをインプリメントするための卓越したソリューションを紹介し、この記事は、SPI-4.2 - クワッド SPI-3 ブリッジの利点と使い方、つまり 4 つの SPI-3 コアを 1 つの SPI-4.2 コアにブリッジングする方法を解説しています。

そして最後に、デバッグに関する記事「ハードウェアとソフトウェアの協調検証 (Hardware/Software Co-Verification)」(P.69) では、ソフトウェアとハードウェアについて完全な可視性を得て、その過程でより高速なデザイン反復ループを達成する方法を論じています。

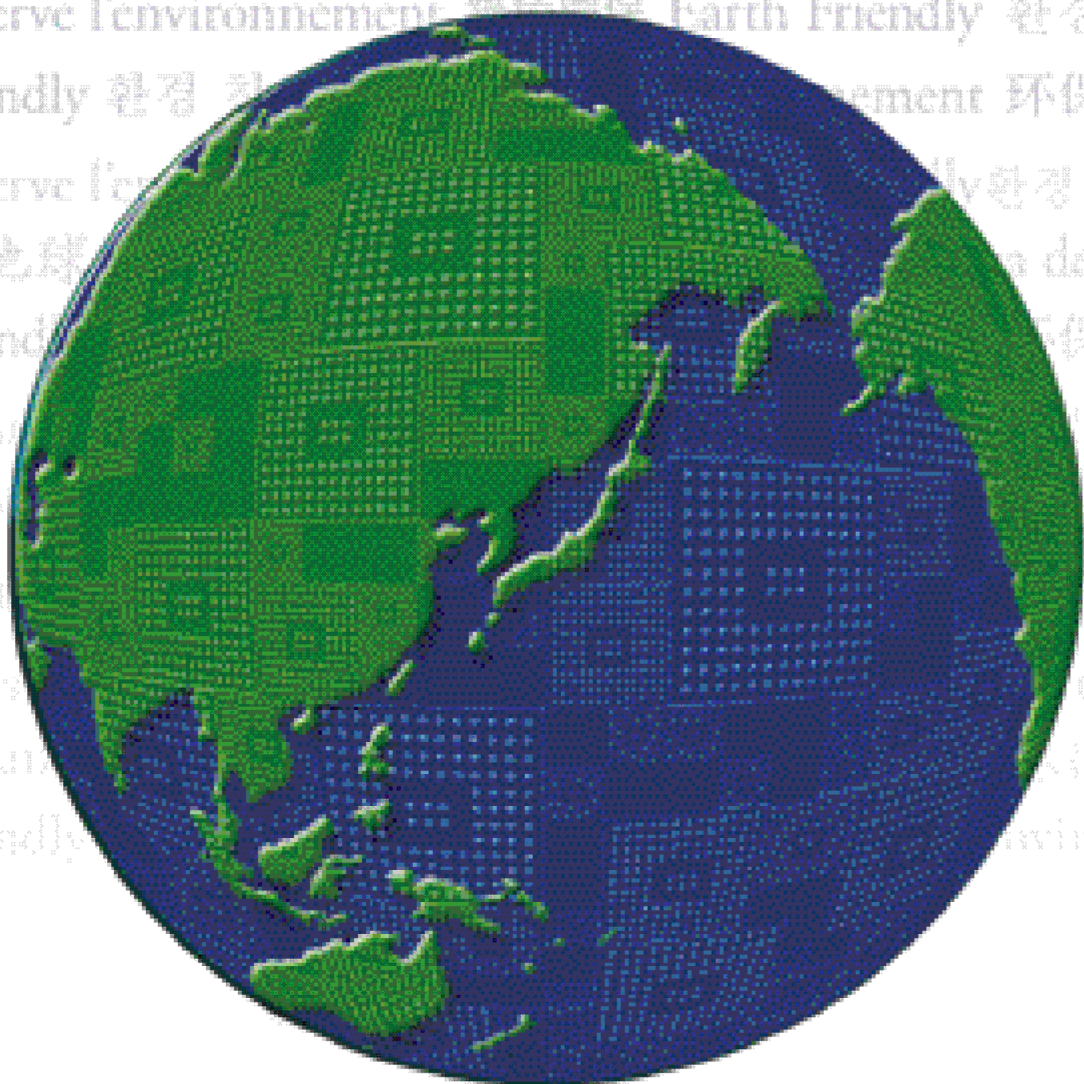
人生はチャレンジの連続ですが、大切なのはそれをチャンスととらえることです。デザイン チャレンジに関する今号の記事がお役に立てば幸いです。



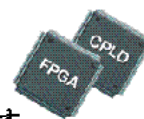
Forrest Couch

編集
Forrest Couch

Earth Friendly



ザイリンクスは鉛フリーソリューションで、RoHS準拠の広範なデバイス、業界最先端、高性能パッケージを提供しております。すでに環境に配慮したデバイス580万個以上の出荷実績*があります。



鉛フリー
RoHS準拠

ザイリンクスは「リサイクル」、「ゴミ削減」、「エネルギー管理」活動を通してグローバルに地球環境をサポートいたします。

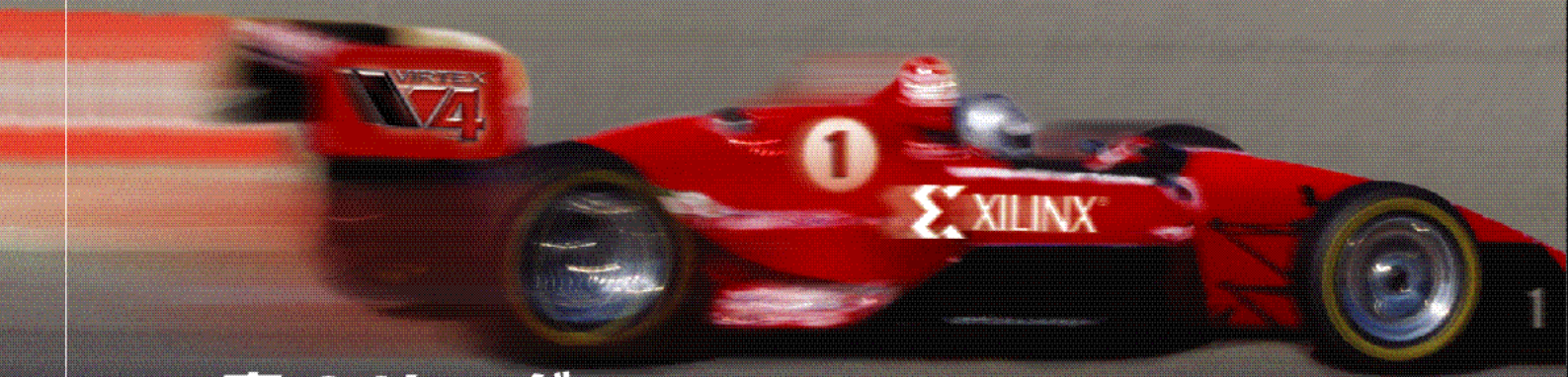
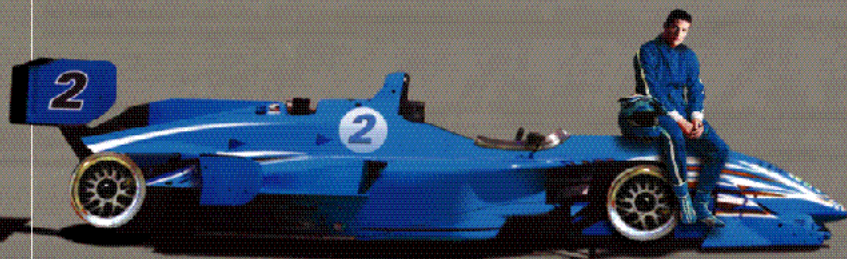
詳しくは www.xilinx.co.jp/pbfree/ をご覧ください。

*2005年6月末現在

XILINX
The Programmable Logic Company™

www.xilinx.co.jp

※RoHS(有害物質使用制限)
※記載されている会社名、製品名は各社の登録商標または商標です。



真のリーダ 真の価値 真の性能



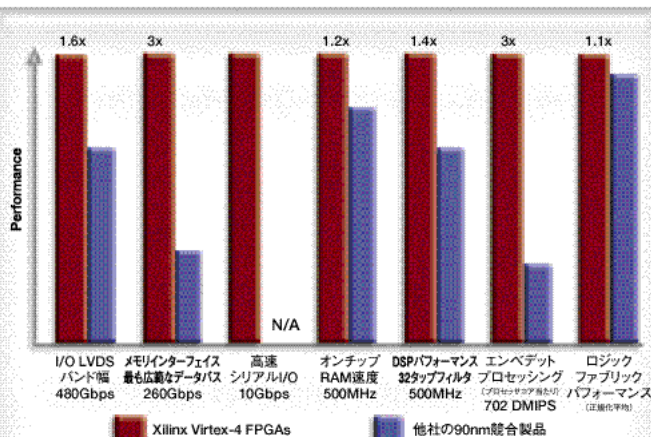
ロジックに
最適



DSPに
最適



プロセッシング/
コネクティビティに最適



(当社比較)



Virtex-4 FPGAは、どの競合90nm FPGAと比較しても3倍のパフォーマンスと集積度、1/10の消費電力、最高の「シグナル+パワー」インテグリティをお届けします。

Virtex-4™プラットフォームFPGAは、システムのあらゆる局面で優れた性能を発揮します。他のどんなFPGAも、I/Oバンド幅、オンチップRAM速度、DSPおよびエンベデッド プロセッシング、さらにロジック・ファブリックの性能すべてにおいてVirtex-4の敵ではありません。的確な機能すべてを装備していることが、最少コストで飛躍的な性能を達成する唯一の方法です。

世界最高最速クラスと低消費電力を両立させたFPGA!

Virtex-4 FPGAは、他の90nm FPGAに比べ1/10以下の電力消費によりインラッシュ消費電力(電源投入時にかかる消費電力)で最高94%、スタティック消費電力で最高78%、ダイナミック消費電力で86%もの低消費電力化を達成することにより、設計者の方々に業界最高レベルの性能を維持しながらこの低消費電力のメリットを十分に活かすことが可能となりました。



すぐに使いこなせる
ソフトウェア

www.xilinx.co.jp/virtex4/

最少のシステムコストで、最高のシステム・パフォーマンスを。



The Programmable Logic Company™

ザイリンクス株式会社 本社 〒163-1118 東京都新宿区西新宿6-22-1 新宿スクエアタワー18階
大阪営業所 〒532-0003 大阪市淀川区宮原3-4-30 ニッセイ新大阪ビル13階

製品のお問い合わせは下記の販売代理店へどうぞ

■新光商事(株)	東京 TEL (03) 5721-2062 (直)	大阪 TEL (06) 6303-3826 (代)
■菱洋エレクトロ(株)	東京 TEL (03) 3546-5011 (直)	大阪 TEL (06) 6301-1221 (代)
■メック ジャパン(株)	東京 TEL (03) 5978-8201 (直)	関西 TEL (06) 6391-1874 (代)
■東京エレクトロデバイス(株)	本社 TEL (045) 474-7089 (直)	大阪 TEL (06) 6399-1523 (直)

Xilinx, Virtexは、米国Xilinx, Inc.の登録商標または商標です。
他の全ての名称は各社の登録商標または商標です。