

# Xcell journal

SOLUTIONS FOR A PROGRAMMABLE WORLD

## FPGA ベースの技術革新により 多彩な広がりを見せる放送機器

### INSIDE

従来の常識を覆すエレクトロニクス製品のソフトウェア プログラム可能性

ザイリンクス FPGA のダイナミック パーシャル リコンフィギュレーション (DPR)

FPGA ベースのイノベーションで多彩な広がりを見せる放送分野



## Xcell journal

発行人 Mike Santarini  
mike.santarini@xilinx.com  
+1-408-626-5981

編集 Jacqueline Damian

アートディレクター Scott Blair

デザイン/制作 Teie, Gelwicks & Associates

日本語版統括 秋山 一雄  
kazuo.akiyama@xilinx.com

制作進行 竹腰 美優紀  
miyuki.takegoshi@xilinx.com

日本語版 制作・  
広告・印刷 有限会社エイ・シー・シー



japan.xilinx.com/xcell/

Xcell Journal 日本語版 69・70 合併号

2010年5月31日発行

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

**ザイリンクス株式会社**  
〒141-0032  
東京都品川区大崎 1-2-2  
アートヴィレッジ大崎セントラルタワー 4F

© 2010 Xilinx, Inc. All Right Reserved.

XILINX や、Xcell のロゴ、その他本書に記載の商標は、米国およびその他の各国の Xilinx 社の登録商標です。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. およびザイリンクス株式会社は、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

# 顧客との共同作業、チームワークが 次世代 FPGA 開発の鍵

## 新しいアーキテクチャの開発に注がれる数年がかりのハードワークと共同作業 そのアーキテクチャが登場する頃には次の開発が進行中

ザイリンクスに入社して 1 年半が過ぎましたが、この間、次世代 FPGA のコンセプトの定義とデザインにザイリンクスとそのパートナーが多くの労力を費やし、製品化する上で顧客の皆様が重要な役割を果たしてきたことを度々目の当たりにしてきました。ムーアの法則に従い 1 つの FPGA を開発するだけでも多数の工程が必要となりますが、ザイリンクスは 2009 年、2 つの FPGA を同時に開発しただけでなく、顧客のデザイン フローを自動化するターゲット デザイン プラットフォームも発表しました。

最新の FPGA 製品が完成する数年も前から、ザイリンクス製品開発チームが顧客や見込み客、そして場合によってはかつての顧客を訪れ、彼らの次世代システムに求める要件、搭載予定の機能、そして元顧客の場合は別のアプローチやデバイスに切り替えた理由、などの点についての情報収集を行います。その後、ザイリンクスの FAE や営業担当が日々収集しているフィードバックも加味し、顧客からの調査結果を詳細に検討します。

こうしてさまざまな市場の顧客に対する調査から得られた情報は、次世代 FPGA に組み込む機能、提供するソフト IP (または提携する IP パートナー)、そしてデザイン ツールに必要な機能などを決定する重要な要素となります。これは非常に地道な作業ではありますが、FPGA がビジネス的に成功を収めるには、幅広い顧客のニーズに応える機能、または急成長中の市場の顧客ニーズに応える機能、あるいはその両方を FPGA のアーキテクチャに組み込む必要があり、重要な過程であることは間違いありません。

顧客への調査と並行し、財務チームは個々のバーティカル市場や経済全体のデータについて、製造チームはファウンダリとプロセス ノードについてそれぞれ詳細に分析します。次に、これらの調査結果を総合的に分析し、開発の専門家とシリコン アーキテクトがアーキテクチャ、関連ツール、そして IP の要件の各仕様を作成する作業に着手します。アーキテクチャの仕様を決定するプロセスで大部分を占めるのは、ザイリンクスのエンジニアが考案した多数の新規特許技術から、アーキテクチャの仕様に含めるものを指定する作業です。これら特許技術は非常に優れたものであり、実際には「仕様に含めない技術」の決定に本当の難しさがあります。

事実、ある回路やアーキテクチャがハードウェア エンジニアリングや学術的な観点から魅力的に見えても、それをソフトウェアで活かすことができなかつたり、新しい回路がユーザーの多くにとって複雑であったり、それを含めることによってリリース時期が大幅に遅れるようであれば、それらを含めない方がビジネスとしては正解であるということがあります。

ザイリンクスの顧客諮問委員会やフィールド諮問委員会による審査を含め、仕様の見直しを複数回経た後、全体的なアーキテクチャ仕様が決定すると、チップ アーキテクトと設計者による設計作業が始まります。これと同時に、ツール開発チーム、IP チーム、開発ボード チームも始動します。まず、ツール開発チームは新しいアーキテクチャのサイズや FPGA に組み込まれる新しい回路のサポートに必要なツールを評価します。IP チームは従来の IP の移行と新規 IP の開発 (または IP パートナーとの提携) に着手し、開発ボード チームは基本ボードと特定市場向けボードの仕様を確定する作業

---

に入ります。ターゲット デザイン プラットフォーム (TDP) が登場したことにより、これらチーム間の連携をこれまで以上に強化し、ドメイン特化プラットフォームやマーケット特化プラットフォーム（ドーター カード開発ボード、IP、リファレンス デザイン、およびその他の技術資料）の開発が進められています。この TDP を活用することで、単純な設計作業が自動化され、付加価値を追加し、製品の差別化を図るための設計に専念していただけるようになります。

ファウンダリからファースト シリコンが届くと、ザイリンクスの専門エンジニアが詳細な特性評価とテストを行います。ザイリンクスの品質保証 (QA) チームはファウンダリと緊密に連絡を取り合い、100 万分の 1 の不具合率をもチェックする厳密な検査を行います。これと並行してザイリンクスの FAE、営業担当、販売代理店を対象に、新しいデバイス、ツール、IP の機能と利点についての研修が実施されます。この段階で設計チームの役割が終わりというわけではありません。シリコンの細かな欠陥を修正し、次は車載および航空宇宙 / 高信頼性 (A&D) 産業向けのデバイスを開発します。そして、特別な回路を追加したり、場合によっては新しい半導体材料を追加し、シリコンの信頼性をさらに高めます。最後に、これらデバイスに対して厳しい範囲でのテストを実施します。たとえば、車載および A&D 向けの広い温度範囲で信頼性テストが行われます。また、デバイスに荷電粒子ビーム（中性子など）を照射して耐障害性とエラー訂正ソフトウェアの機能をテストする徹底的な SEU (Single Event Upset) テストも行われます。

各製品ファミリのテスト完了後、最終工程として私も所属するマーケティング部門が記者発表資料、販促資料、技術資料を準備し、ウェブサイトを更新します。こうして、シリコン、ボード、ツール、関連 IP を含む新製品が市場に投入されます。そして、この時点ですでに次世代 FPGA に関連する顧客への調査や仕様の確定作業が進行しているのです。

これまで長年、業界誌の編集者として半導体産業に携わってきましたが、顧客の意見を取り入れて次世代デバイスの仕様を決定する作業を含め、新製品を市場に投入するまでに必要とされる労力と共同作業がどれほどであるか十分に理解していませんでした。Virtex<sup>®</sup>-6 および Spartan<sup>®</sup>-6 FPGA ファミリ、そしてターゲット デザイン プラットフォームという 3 つのリリースを同時に成し遂げたことから、FPGA を発明したザイリンクスが常に完成度の高いデバイスの開発に取り組み、多数の画期的製品を生み出せるようなゆみない努力を続けていることがお分かりいただけると思います。これらの成果は、顧客の皆様からの貴重なご意見やご指導なしには決してなしえなかったでしょう。この場を借りて皆様にお礼を申し上げるとともに、この成果を皆様がどのような形で活用してくださるのを楽しみにしています。



**Mike Santarini**  
発行人



## すぐに使えるFPGAボードで、設計や試作コストが下げられると思います。

平日16時までのご注文で即日発送にて最短翌日にお届けいたします。BGAパッケージのFPGAなど、大きな基板で6層や8層基板を製作するよりも、ヒューマンデータのFPGAボードを採用し、安価な4層や両面の基板でお仕事が進められます。

同じボードサイズで様々なFPGAやCPLDが選択できます。今後の新しいFPGAへもすぐに置き換えが可能です。当社は1997年からFPGA評価ボードの販売を行っており、豊富な納入実績がございます。当社のウェブサイトでは、回路図やマニュアル、パターン図などすべて公開しております。ご購入前に充分ご評価していただけます。

- FPGAの動作に必要な最低限の機能を搭載。単一電源ですぐに活用できます
- ACM/XCMシリーズはそれぞれ外形やコネクタ位置が同一で置き換えが可能です
- 豊富なラインナップで100種類以上の製品をご用意しています
- 回路図、マニュアルは購入前でも自由に参照できます
- 豊富な納入実績で安心してお使いいただけます
- 基本的に即納体制で最短翌日からご活用いただけます
- スピードグレード変更などのカスタマイズもご相談ください

### Virtex-5使用製品例



FPGA Virtex-5 搭載  
USB-FPGA ボード  
**EDX-006**

クレジット  
カードサイズ

MRAM  
搭載

USB  
Config

USB  
Comm

¥99,000(税込103,950)



- FPGA XC5VLX30-1FFG676C 搭載
- USB経由でFPGAコンフィギュレーション
- USBポートによりPCと通信が可能、FT232RLのSYNC FIFOモード対応
- 設計ミスによるFPGAの保護のため全I/Oに保護抵抗を挿入
- 電源：外部DC5V(USBケーブルのみでは動作致しません)
- 大容量電源回路搭載
- オンボードクロック 50MHz
- 7ピンJTAGコネクタ
- ユーザI/O 100本
- USBラインのESD保護およびサージ保護
- コンフィギュレーション用リセット機能
- 6層基板を採用

Virtex-5 LXT FFG665 搭載  
ブレッドボード  
**XCM-107 シリーズ**

ハーフ  
カードサイズ

Rocket  
I/O

¥118,000(税込123,900)~



- XC5VLX30T-1FFG665CまたはXC5VLX50T-1FFG665Cを搭載
- 豊富なI/Oを外部引き出し(128本)
- 大容量SPIフラッシュROMを搭載(iMPactでISP可能)
- オンボードクロック 30MHz、50MHz(LVTTL)
- FPGAへのコンフィギュレーションとROMへのISPが可能な7ピンJTAGコネクタ装備
- JTAG Buffer回路で、安定したダウンロードを実現
- 3.3V単一電源動作 1.0V、1.2V、2.5Vをボード内で生成
- VCCOを1系統分離使用可能(製造時にテストされません)
- コンフィギュレーション用リセット機能
- 6層基板を採用

Virtex-5 FFG676 搭載  
ブレッドボード  
**XCM-011 シリーズ**

クレジット  
カードサイズ

FRAM  
搭載

SDRAM  
搭載

¥120,000(税込126,000)~



- XC5VLXの676ピンBGAチップ搭載
- 豊富なI/Oを外部引き出し(100pin)
- 大容量SPIフラッシュROMを搭載(iMPactでISP可能)
- クロック48MHz、外部入力可能
- FPGAへのコンフィギュレーションとROMへのISPが可能な7ピンJTAGコネクタ装備
- JTAG Buffer回路で、安定したダウンロードを実現
- デバッグに便利な8つのテストポイント
- 汎用LED 2個
- 汎用スイッチ 2個
- 3.3V単一電源 1.0V、2.5Vをボード内で生成
- VCCO分離可能(製造時にテストされません)
- コンフィギュレーション用リセット回路
- 6層基板を採用

### Extended Spartan-3A (Spartan-3A, Spartan-3AN, Spartan-3A DSP) 使用製品例



デバイス名	ゲート数	パッケージ	Logic Cell	Maximum Distributed RAM Bits	Total Block RAM Bits	DCM	DSP	ボード I/O 数	製品型番	特長
XC3S400A-4FTG256C (Spartan-3A)	400 K	FTG256	8 K	56 K	360 K	4	-	56	XCM-305-400A	MRAM
XC3S700A-4FTG256C (Spartan-3A)	700 K	FTG256	13 K	92 K	360 K	8	-	56	XCM-305-700A	MRAM
XC3S700A-4FGG484C (Spartan-3A)	700 K	FGG484	13 K	92 K	360 K	8	-	100	XCM-015-700A	FRAM, DDR2
XC3S400A-4FTG256C (Spartan-3A)	400 K	FTG256	8 K	56 K	360 K	4	-	100	XCM-014-400A	FRAM
XC3S700A-4FTG256C (Spartan-3A)	700 K	FTG256	13 K	92 K	360 K	8	-	100	XCM-014-700A	FRAM
XC3S1400A-4FTG256C (Spartan-3A)	1400 K	FTG256	25 K	176 K	576 K	8	-	100	XCM-014-1400A	FRAM

デバイス名	ゲート数	パッケージ	Logic Cell	Maximum Distributed RAM Bits	Total Block RAM Bits	DCM	DSP	ボード I/O 数	製品型番	特長
XC3SD1800A-4FGG676C (Spartan-3A DSP)	1800 K	FGG676	37 K	260 K	1512 K	8	84	100	XCM-016-1800A	FRAM, SDRAM
XC3SD3400A-4FGG676C (Spartan-3A DSP)	3400 K	FGG676	54 K	373 K	2268 K	8	126	100	XCM-016-3400A	FRAM, SDRAM
XC3S200A-4VQG100C (Spartan-3A)	200 K	VQG144	4 K	28 K	288 K	4	-	48	XCM-304-200A	
XC3S700AN-4FGG484C (Spartan-3AN)	700 K	FGG484	13 K	92 K	360 K	8	-	128	XCM-108-700AN	
XC3S50AN-4TQG144C (Spartan-3AN)	50 K	TQG144	2 K	11 K	54 K	2	-	56	XCM-303-50AN	
XC3S200AN-4FTG256 (Spartan-3AN)	200 K	FTG256	4 K	28 K	288 K	4	-	100	EDX-005	USB Config, USB Comm

### アクセサリ

#### Digilent社 USB対応ダウンロードケーブル Platform Cable USB 互換品 **XUP-USB-JTAG-H**

ヒューマンデータ仕様  
ヒューマンデータ仕様とは、下記  
※印の付属品が添付されているもの  
になります。

¥18,500(税込19,425)



ダウンロードケーブル本体	XUP USB-JTAG (Digilent 製)
フライングワイヤKIT	ばら線のJTAG接続用ハーネス
ZKB-031KIT (同等品)※	2mmピッチ14pinコネクタ / 2.54mmピッチ7ピンコネクタ変換キット
ZKB-009KIT (同等品)※	2mmピッチ14pinコネクタ / 2.54mmピッチ10ピンコネクタ変換キット

製品にマニュアルなどの資料の付属はございません。使用方法はHW-USBと変わりません。

#### XCM/ACM-2シリーズ対応 電源付きユニバーサル基板 **ZKB-051**



ACM/XCM-2シリーズを搭載できる、電源付きユニバーサル基板です。

- ACM/XCM-2シリーズの全ピンを引き出し
- 3.3V DCDCコンバータ(10A)
- 電源スイッチ
- 電源LED
- GND用テストポイント 4個実装可能
- 基板サイズ 114 × 155 mm
- 4層基板を採用



¥17,000(税込17,850)

#### XCM/ACM-0シリーズ対応 ユニバーサル基板 **ZKB-054/ZKB-054KIT**



ACM/XCM-0シリーズに対応した3.3V用電源回路付きのユニバーサル基板です。完成品と組み立てキットがございます。

- ZKB-054(完成品)仕様
- 基板寸法: 115 X 155 mm
  - 厚み: 1.6mm レジスト、シルク付きスルーホール基板
  - Bellnix製DCDCコンバータ BST04-0.7S10PCM 実装
  - ACM/XCM-0シリーズ対応66ピンコネクタ2個実装
  - 電源スイッチ
  - 電源表示LED
  - 2.1ミリ用DCジャック



¥11,500(税込12,075)~

※その他FPGA Boardやアクセサリを100種類以上ラインナップしています。詳しくはウェブをご覧ください。

## VIEWPOINTS

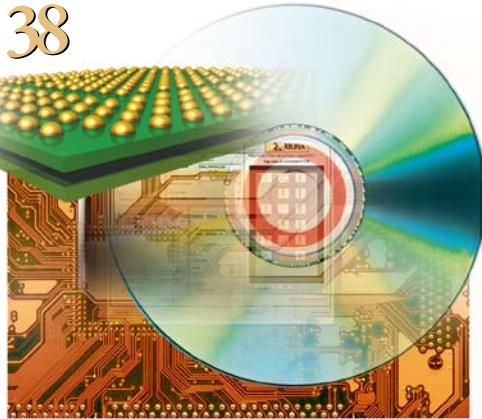
### Letter from the Publisher

顧客との共同作業、チームワークが  
次世代 FPGA 開発の鍵 ... 表 2

### Xpert Opinion

従来の常識を覆すエレクトロニクス製品の  
ソフトウェア プログラマビリティ ... 38

38



## Cover Story

FPGA ベースのイノベーションで  
多彩な広がりを見せる放送分野 ... 4

4



## XCELLENCE BY DESIGN APPLICATION FEATURES

### Xcellence in Automotive & ISM

オンザフライでのシステム変更を可能にする  
ザイリンクス FPGA のダイナミック パーシャル  
リコンフィギュレーション (DPR) ... 10



## THE XILINX XPERIENCE FEATURES

### Xperts Corner

リオーダー方式のコントローラによる  
DDR SDRAM の効率化 ... 18

### Xperts Corner

FPGA のタイミング クロージャ ... 22

### Xplanation: FPGA 101

シンプル MicroBlaze マイクロ コントローラ  
(SMM) のコンセプト ... 26

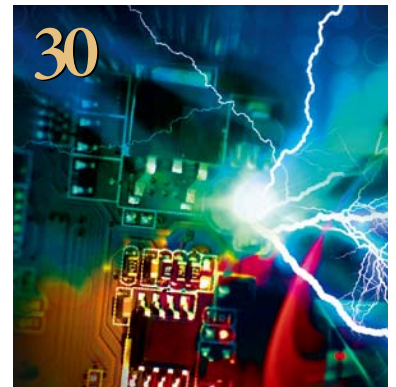
### ASK FAE-X

ハードウェア アクセラレーションによる  
MicroBlaze 処理の高速化 ... 30

26



30



## XTRA READING

Tools of Xcellence ... 42

### 広告索引

有限会社ヒューマンデータ ... 2  
株式会社アイダックス ... 17  
株式会社ミッシュインターナショナル ... 36  
マイクレル・セミコンダクタ・ジャパン株式会社 ... 37  
アルデック・ジャパン株式会社 ... 41  
株式会社沖情報システムズ ... 44

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



From Broadcast to 'Broader-cast'  
with FPGA-Based Innovations

# FPGA ベースのイノベーションで 多彩な広がりを見せる放送分野

ザイリンクスのデバイスが HD、3D に続くテレビのトレンドを牽引



Mike Santarini  
 Publisher, Xcell Journal  
 Xilinx, Inc.  
[mike.santarini@xilinx.com](mailto:mike.santarini@xilinx.com)

この 10 年間、エレクトロニクス業界ではフラットパネル テレビやその関連分野でめまぐるしいほどの技術が登場し、CRT やリアプロジェクション テレビ、アナログ (SD) 放送は過去のものとなりました。現在も、多数の企業から新しいテレビ技術が次々と展開されていますが、こうした急速な技術の進歩が消費者に真の価値をもたらすには、リビング ルーム、映画館、そして近い将来、携帯型機器や自動車などにも放送局や映画スタジオから最新かつ高度なコンテンツを配信できるような、放送に関連する機器の高度化が必須です。

放送および映画機器メーカーは、超ハイエンドのビデオ カメラを始めとして、スタジオでの編集や信号の送信に使用する機器に、以前からザイリンクスの FPGA をロジック IC として使用してきました。現在、この市場では 1080p 放送の普及を視野に入れた FPGA ベースの機器が増加しています。また、FPGA ベースの 3D テレビ機器の開発も既に始まっており、映画「アバター」のような 3D 品質の映像が小型スクリーンで楽しめるような時代も到来するでしょう。放送機器市場でも最大のセグメントであるテレビ放送市場を観察してみると、高度化するコンテンツの配信に多数の FPGA が放送機器全体で活用されていることが分かります。

### 放送の基礎知識：現場から画面までの流れ

放送は多数の人が毎日視聴しているにもかかわらず、その複雑な制作過程は知られていない、と指摘するのは、ザイリンクスの Broadcast Segment Marketing Group 担当マーケティング マネージャ、Ben Runyan です。確かに、放送の舞台裏では複雑な技術が数多く駆使され、多数の人が制作に携わっています。

放送および映画機器市場への注目はさほ

ど高くありませんが、高価な機材が少量しか流通しないという点で、競争の厳しい市場です。機器メーカーは、テレビ信号を家庭の最新式デジタル テレビまで届けたり、IMAX シアターや映画館のスクリーンに映画を映し出すために必要となるさまざまな機器を開発しています。Runyan によると、映画産業の方が新しい放送技術の導入に積極的（囲み記事を参照）ですが、テレビ業界も映像品質に対する要求の高まりに伴い、新技術の採用ペースが加速しているといえます。「ローカル テレビ局やネットワークのキー局を見学してみると分かりませんが、テレビ局にはさまざまな機材があり、天井や壁のいたる所にケーブルが張り巡らされています。テレビ番組の放送に費やされている労力と技術には、驚かされるばかりです」(Runyan)。

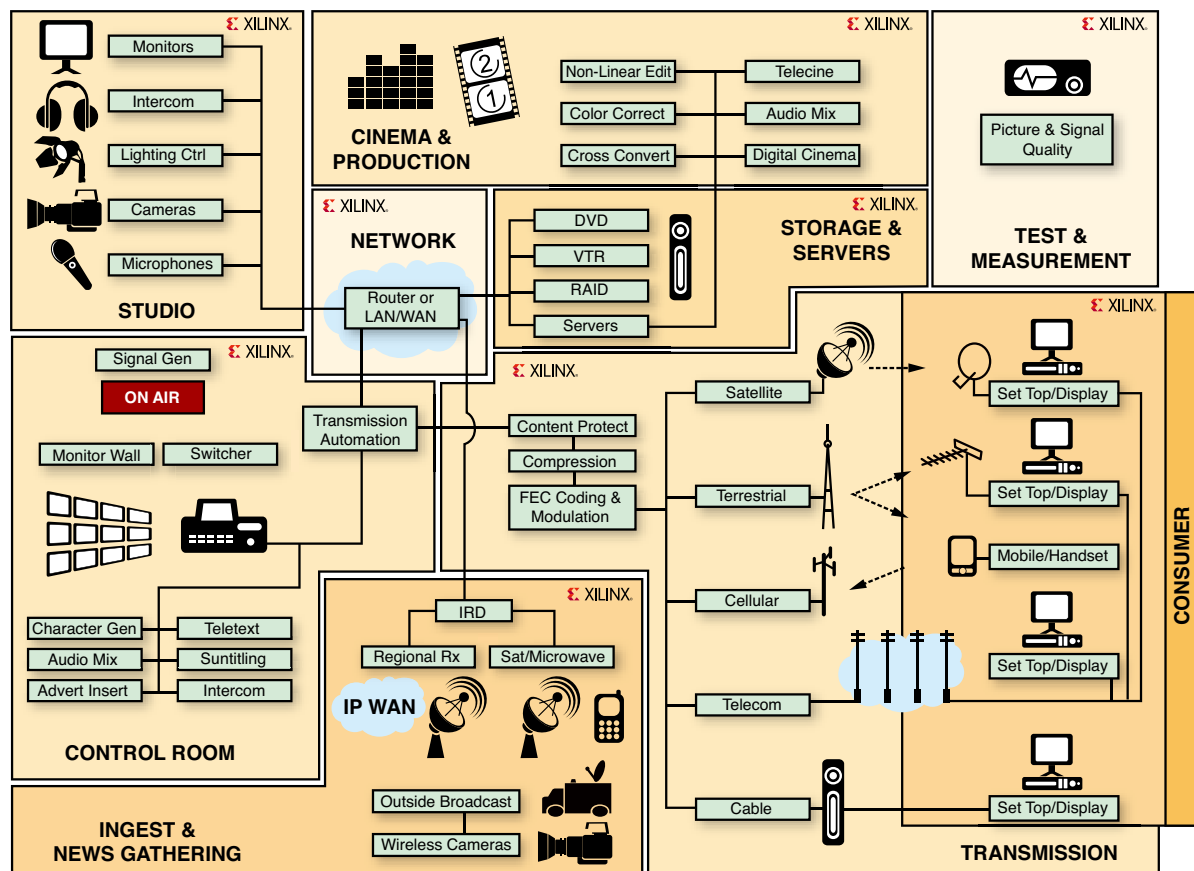
たとえばニュース中継の場合、マイクを持ったレポーターが高価な最新式デジタルビデオ カメラに向かって話します。カメラからは、非圧縮の音声 / ビデオ信号が現場の中継車に有線または無線で送信されます。通常、中継車にはポータブル型のビデオ表示 / 編集機器と無線装置が積まれています。中継車の機材は、信号を瞬時に圧縮し、ライブ映像や関連する録画コンテンツを直接送信塔に送信したり、衛星経由で放送局のスタジオに送信します。

スタジオでも複雑な機器が使用されており、現場から放送信号を受信するとコンテンツをストレージ サーバーに記録し、信号をコントロール ルームのミキシング / 編集機器やマルチスクリーン モニタに送信します。このコンテンツは、番組制作技術者やディレクタ、プロデューサの手によって外部からのほかの中継映像やスタジオ内のカメラ出力（ニュース番組の司会者の映像など）とミキシングされます。さらに、字幕やタイトル文字を追加し、イベントの順序を調整して最終的なマスタ プログラムが完成します。

次に、このマスタ プログラムはスタジオの送出機器に送られ、そこでケーブル、衛星、地上波、有線放送 (IP テレビ)、ワイヤレス放送（携帯電話向け）などの各種伝送媒体に応じた複数のコーデック規格を用いて圧縮されます。これらの圧縮された信



図 1 - 放送技術において活躍するザイリンクスの FPGA



号は、有線または無線の搬送ネットワークを利用してケーブル テレビや衛星放送のセットトップ ボックス (STB)、デジタルテレビ、PC、携帯型機器などに配信されます。そして、家庭にある受信機器では、信号の圧縮が解除され、コンテンツが表示されます。

一連の放送業務（特に生中継の放送）を支えるには、膨大な量の広帯域光ファイバおよび衛星設備、そして高性能ビデオ編集機器と大量のストレージ機器が必要であると Runyan は述べています。また、24 時間 365 日、1 バイトの欠落もなくビデオデータを処理し続ける必要があるため、きわめて高い信頼性も求められます。このように、高いスループット、柔軟性、信頼性の要件すべてを満足させるソリューションとして、多くの放送機器メーカーが FPGA を採用しています（図 1 参照）。

「現場のカメラ、中継車の機材、スタジオの編集機器（スイッチャー、ルータ、マ

ルチビューアなど）、ポスト エディット機器、大規模なビデオ サーバ ストレージファーム、圧縮および送信機器など、あらゆる放送機器でザイリンクスの FPGA が重要な役割を果たしています（図 2 参照）。事実、放送業界では以前から ASIC や ASSP に代わって FPGA が使用されるようになり、現在ではロジック IC の大半を占めています。高度なシステムの流通量は限られているため ASIC では開発コストがネックとなり、ASSP のハードウェア パフォーマンスではこの市場のニーズに対応できません」（Runyan）。1080p、3D、そして今後数年の内にモバイル放送へと市場が移行していく中で、テレビ放送機器はさらに高い仕様が要求され、ハードウェアのパフォーマンス、信頼性、そして特に柔軟性がより強く求められるようになります。このため、FPGA の活躍の場は増えていくだろうと Runyan は述べています。さらにザイリンクスは、最新

の Virtex®-6 および Spartan®-6 FPGA ファミリーを対象にしたターゲット デザイン プラットフォームを提供し、機器の短時間での開発、生産性の向上を支援しています。特に、近日リリースが予定されている Virtex-6 FPGA ブロードキャスト コネクティビティ キットは、この市場の機器メーカーに大きなメリットをもたらすことになるでしょう。「ターゲット デザイン プラットフォームにより、顧客の皆様はシステムを新規に開発する必要がなくなり、製品の差別化に時間を費やすことができるようになります。ザイリンクスはカスタマイズした放送機器専用のプラットフォームを幅広く提供し、高度なビデオ / オーディオ機能とネットワーク接続、HD ビデオのリアルタイム処理、プロ品質のマルチチャネル エンコード / デコード機能、高速デジタル信号処理による伝送および変調機能などが要求される機器の開発をサポートしています」（Runyan）。



## テレビ放送最大のトレンド、1080p

現在、放送業界が急速に前進している理由は、過去 25 年間にわたるさまざまな規格争いの決着がつき、デジタル テレビ時代の公平な競争の土壌が整ったことによると指摘するのは、DIS Consulting Corporation (Strategy Analytics と提携関係にある放送アナリスト会社) の CEO 兼チーフ アナリスト、Douglas I. Sheer 氏で、放送業界 40 年のベテランです。「この結果、企業は新しいタイプの音声および映像放送が制作できるようになり、それを配信する手段も増加しています。特にめざましいのが、伝統的な放送と IT の融合です。

現在、多くの人々が PC を利用してテレビ放送を観ていますが、今後は携帯型機器での視聴も増加するでしょう。こうした視聴形態の広がりを全米放送事業者連盟 (NAB) は「ブロードキャスト (Broadcast)」と呼んでいます」(Sheer 氏)。

テレビ放送業界の企業の多くが 1080p、

フル HD、ワイヤレス ビデオ放送をサポートした 3Gbps テクノロジーに対応するため、スタジオの新規建設や改修を進めています。

業界が HD 放送のロードマップを策定中だった約 8 年前、ESPN など一部のテレビ ネットワークは数百万ドルを投じて送信機器の導入やフル デジタルの 720p HDTV スタジオの建設を敢行しました。しかし当時は、既存の機器をなるべく長く使用し続けようというネットワークの方が多数派でした。

その後、政府の施策もあってまず北米で、そして最近では世界中でデジタル放送への移行が進行しているとザイリンクスの Runyan は述べています。米国では、720p への移行に消極的だった企業の多くが 720p をスキップし、1080p の HD 放送に対応したスタジオの建設やアップグレードに着手しています。

「1080p への移行は決して容易なことではありません。それは、いち早く 720p を導入した企業も同じことです。720p プログレッシブ スキャン フォーマットで使用

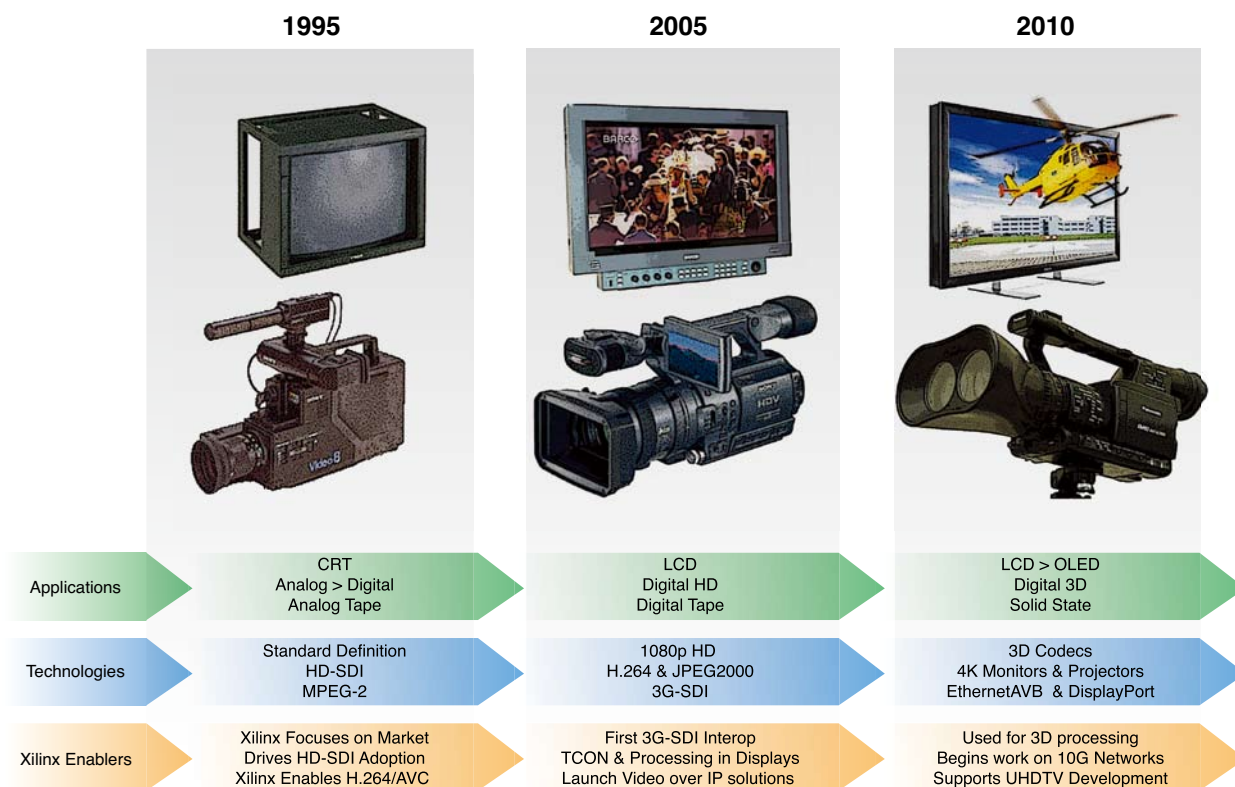
する HD シリアル デジタル インターフェイスのデータ レートは約 1.5Gbps ですが、1080p のデータ レートは 60fps の場合で約 3Gbps です。これは、ビットレート、帯域幅ともに (720p の) 2 倍であり、2 倍のデータ量を扱うことになります」(Runyan)。

また、HD という用語を曖昧に使用している人も多く、「店頭で比較しても解像度の違いは明確ではないと思いますが、放送機器の複雑さという点言えば、720p と 1080p はまったく異なります」(Runyan)。

この 10 年間で放送業界の企業の多くがデジタル伝送機器を購入、設置して HD への第一歩を大きく踏み出しましたが、このコストは莫大なものになったとアナリストの Sheer 氏は指摘しています。そして現在、低価格化もあって一般家庭で 1080p HD テレビへの買い換えが進んだ結果、企業も 1080p 対応の 3Gbps 機器の購入を開始しているところです。「今までは需要の浮き沈みに対して傍観していました。

消費者向けの 1080p テレビ市場は金融

図 2 - ターゲット デザイン プラットフォームで、放送技術の革新を支えるテクノロジーを提供



危機の前から成長していましたが、金融危機後いったん下降線をたどり、現在では景気の回復につれて再び上昇しています。この結果、1080p への消費者需要は本物であると判断した放送事業者が、1080p 対応機器の購入を開始したというのが実情です」(Sheer 氏)。

1080p で求められるデータ レートに対応するには、放送スタジオだけでなくテレビ放送で使用するすべての機器を広帯域化すると共に、新しい機能の追加やコーデック規格の変更などに対処できるようにハードウェア プログラマビリティの確保も必要になると Runyan は述べています。

「多くの企業にとって、1080p への移行には巨額の設備投資が必要となります。同じ機器を 20 年間継続して使用している企業も新しい機器への切り替えが必須となっています。中には、今まで使用していた機器が生産終了になったため新しい機器への移行を余儀なくされるケースもあります」(Runyan)。

たとえば、スタジオにはビデオ信号の品質を詳細にチェックする専門家がいます。以前は、超ハイエンドの CRT テレビを使用してチェックしていましたが、既に生産が終了しているため、新しいフラットパネル テレビに移行せざるを得なくなっています。「LCD、プラズマ、OLED などさまざまなフォーマットのモニタが多くブランドから発売されていますが、映像のアーチファクト、カラー、シェードなどを細かく識別できる専門家の間でも、スタジオ用として最適なモニタの選定には大きな議論があります」(Runyan)。

1080p に移行すると、放送機器で使用するコーデック規格は H.264/MPEG-4 AVC となりますが、MPEG-2 や従来からの規格もサポートする必要があると Runyan はさらに述べています。この H.264/MPEG-4 AVC は、ブルーレイや超高精細ストリーミング ビデオで使用される規格です。最近登場している多くのビデオ規格同様、H.264/AVC もマルチパート仕様となっています。この規格は完全に定義付けされていますが、SVC (Scalable Video Codec) や 3D 放送用

の MVC (Multi-Viewer Codec) といった派生仕様が次々と策定されています。今後、放送局で MPEG-4 の利用が拡大し、高速かつ高信頼性の HD および 3D 放送を実現していく上で必要な機能が明確になってくれば、これらの仕様も確立されていくでしょう。

「コーデックの仕様は策定段階です。機器メーカーは未だビデオの品質向上に取り組んでおり、今後、コーデックの一部が変更される可能性もあります。FPGA の採用がこの業界で最も有力な選択肢と位置づけられている理由はここにあります。FPGA ベースのシステムであれば、ニーズの変化や規格の発展に合わせて新しい機能を追加できます。コーデックに変更があった場合は、1 つの機器だけでなく、多くの機器に変更が必要となります。ザイリンクスと IP パートナが提供する幅広い種類のコーデック IP を利用すれば、機器のスタジオ導入後、迅速に新しいコーデック機能が追加できるのです」(Runyan)。

## 先行するデジタル シネマ

テレビ放送はようやく 1080p や 3D への移行を開始したところですが、このトレンドを先取りしているのがデジタル シネマです。デジタル シネマの基本解像度は 2K ピクセル (2,048 x 1,080 ピクセル) で、1080p HD テレビの 2 倍のビット レートが必要です。デジタル シネマ用のカメラはさらに解像度が高く、4,520 x 2,540 ピクセル (4K x 2K) に達しています。

そして、テレビにもいずれそのような時代が訪れるでしょう。

ザイリンクスの Broadcast Segment Marketing Group でマーケティング担当マネージャを務める Ben Runyan は次のように述べています。「映像技術の点では、映画の分野が全体的に先行しています。「アバター」のような大作映画では、巨額の制作費を投入して最先端機器が使用されていますが、これらの多くはまだ初期のカスタム機器です。最終的に、これらの技術はテレビ放送市場にも導入されますが、周辺の

インフラストラクチャを整備する必要があるため、導入には時間を要します。」

また、使用される信号のサイズが非常に大きいため、映画をネイティブの 3D フォーマットで撮影し、特殊効果とシームレスにミックスできる迫力あるライブ イメージを制作するには、超ハイエンドのカメラなど、映画撮影機器や編集機器などあらゆる機材が大容量の映像信号を高速に処理する必要があります。

高度な機器が必要となるのは映画スタジオだけではなくありません。上映館にも最新機器の導入が必要です。Runyan によると、最先端の映画館では映画スタジオから衛星経由でライブ コンテンツを直接受信できる機器が導入されているといいます。また、ライブ イベントやスポーツの生中継を上映できる映画館もあります。

「これにより、映画館は最大の収益源である店内販売に注力できるようになります。また、映画スタジオにも多くのメリットがあります」(Runyan)。

従来、映画館は映画スタジオからフィルム、テープ、ディスクなどの媒体で映像を受け取り、これをハイエンドのプロジェクタで上映していました。

ところが、映画スタジオは違法コピーによって毎年巨額の損失を被っています。こうした知的財産侵害の対策として、映画スタジオは現在、映画館のプロジェクタに映像を直接ダウンロードしたり、あるいはストリーミング配信するようになっています。これにより配給コストが削減されるだけでなく、最新の映画を即座に上映できる、映画タイトルの正確なコピーを入手できる、コンテンツの違法コピーを防止できる、などのメリットがもたらされます。これを受け、高性能のセキュリティ、暗号化、コピー プロテクトなどの開発も進んでいます。

このようなトレンドに対応するには、映画のダウンロードから上映までに対応した高度な機器を映画館に設置する必要があります。

1080p は、3Gbps 対応機器によって実現する数多くのテクノロジーの 1 つに過ぎないと Sheer 氏は述べています。



「1080p テクノロジーの需要のピーク時期、そして最も普及すると考えられるフォーマットはまだ明確になっていません」(Sheer 氏)。3Gbps 対応機器であれば、携帯型機器や自動車内にも放送の配信が可能となります。

「現在のモバイル放送の品質はそれほどよくありませんが、データ レートの向上に伴い改善されるでしょう」(Sheer 氏)。携帯型機器がより高度な処理性能とネットワーク機能を備えるようになれば、ノートブック PC やネットブックを用いて番組の視聴を希望する消費者ニーズに合わせてワイヤレス放送も活発になると考えられます。

## 次の巨大市場、3D テレビ

最近では、1080p の HD テレビ放送用機器だけでなく、家庭向けの鮮明な 3D テレビ放送を実現する機器もメーカー各社から登場しています。3D 放送のオンエア自体は数年も前から可能であったものの、映像の鮮明度が不十分であったと Sheer 氏と Runyan は揃って指摘しています。しかし近いうちに、最新の映画で使用されているような新世代の高精細 3D を家庭でも楽しむことが可能となります。今年、ラスベガスで開催された CES (Consumer Electronics Show) では、メーカー数社が 3D テレビを発表しました。現状では、専用メガネが必要なものと不要なものがあります。

3D テレビへの移行により、「放送事業者はより高付加価値のサービスを加入者に提供できるようになります。ケーブル テレビや衛星放送でデジタル HD または 3D の番組を放映したり、オンデマンド配信が可能となります。さらに、映画だけでなく、サッカーや野球の試合のようなライブ イベントも 3D で視聴できるようになります」(Runya)。イギリスでは既にサッカー中継の 3D 放送が試験的に開始されており、2010 年ワールドカップではさらに規模が拡大される予定です (<http://www.televisionbroadcast.com/article/93870> 参照)。

1080p のスタジオ機器を導入済みの企業にとって、3D 放送への移行は比較容

易とはいえ、大がかりなアップグレードになると Runyan は述べています。3D テレビ放送では複数のカメラで撮影した映像を使用するため、広い帯域幅が要求されます。そのため、業界全体で新しい H.264 コーデックと MVC を組み合わせて採用していく必要があります。つまり、スタジオ機器にはこのような機能と高いデータ スループットへの対応が求められます。

「スタジオ機器は将来的に多少複雑にならざるを得ません。少なくとも、2D 放送と 3D 放送の 2 つの信号を同時に処理できる機能が必要となります。放送形態としては、放送局でポスト エディット機器を使用して通常の 2D 番組を 3D に変換する方法もありますが、より本格的な方法では、3D カメラでコンテンツを撮影し、そこから 2D HD バージョンを作成しています。ジェームズ・キャメロン監督の話題作「アバター」もこの方法で制作されたのです」(Runyan)。

ただし、3D 技術はまだ市場において十分な実績がありません。専用メガネ不要の 3D テレビで本当に鮮明な映像が得られるのか、専用メガネが必要な機種では毎日メガネをかけることに抵抗はないのか、といった疑問が多く聞かえてきます。

3D テレビの長所と短所が市場で明らかになれば、3D テレビは急速に普及されていくと Sheer 氏は考えています。「映画館で厚紙のメガネをかけて映画を観た経験は誰にもあると思いますが、最新のステレオ 3D 技術は格段に改善されています。この技術はハリウッドでも支持が高く、スポーツの生中継でも利用されるようになっていきます。最も重要なのは、この技術が HD のインフラを利用できるという点でしょう。事実、HD の実現には四半世紀かかりましたが、3D は既に HD のインフラが整備されているため、それよりはるかに短期間で普及するはずです」(Sheer 氏)。

Sheer 氏は、3D テレビ市場はまず 3D ブルーレイ DVD プレーヤーから立ち上がると考えています。これにより、映画業界からリリースされている多くのステレオ 3D 映画が再生可能となります。それに続いて、家庭への 3D テレビ放送が開始されることになるでしょう。「スポーツ中継が

3D になれば、非常に大きな魅力があります」(Sheer 氏)。

## 1080p と 3D テレビの後

1080p の HD 放送が世界中で普及し、3D テクノロジーも映画の世界からメインストリームのテレビへと急速に拡大している現状を見ると、今後、放送技術の進歩は緩やかになるだろうという見方もあります。しかしそうならないことは明らかです。

Runyan によると、現在の HD テレビの 16 倍に相当する 7,680 x 4,320 ピクセルという驚異的な解像度の「超高精細テレビカメラ」を試作している企業もあるといいます。このカメラに必要とされるビット レートは非常に大きいので、現在は 1 日に 20 分の映像しか撮影できません。しかしエレクトロニクスの進歩とコーデックの改良により、超高精細テレビの品質に対応するビデオが遠くない時期に普及していくのは間違いありません。また、これらの発展により、別の可能性も出てきます。

「超高精細テレビに必要なビット レートが実現すれば、もはや現在のテレビや映画とは異なる形態のものが登場する可能性があります。たとえば、これほど大容量のビット ストリームなら本物のホログラフィ投影もサポート可能となるでしょう。ホログラフィであれば、スポーツの生中継もあらゆる角度から立体で楽しむことができます。その際、ズームイン時のレートは最低 1080p で、現在市販されている最も高精細なテレビと同一です」(Runyan)。

これが実現すると、テレビで放送を視聴する習慣はなくなり、新しいタイプのホログラフィ プロジェクタが使用される可能性が出てきます。そして、ここでもザイリンクスの FPGA が間違いなく使用されることでしょう。

ザイリンクスのブロードキャスト市場向けターゲット デザイン プラットフォームの詳細は、<http://japan.xilinx.com/broadcast> を参照してください。Virtex-6 FPGA ブロードキャスト コネクティビティ キットの詳細は、<http://japan.xilinx.com/v6bck> を参照してください。●●●

## Driver Assistance System Rides FPGA Reconfigurability

# オンザフライでのシステム変更を可能にする ザイリンクス FPGA のダイナミック パーシャルリコンフィギュレーション (DPR)

状況適応タイプのハードウェアを効果的に使用したビデオベースの  
運転支援アプリケーション

Christopher Claus  
PhD Candidate  
Technical University of Munich, Germany  
[Christopher.Claus@tum.de](mailto:Christopher.Claus@tum.de)

Florian Altenried  
Master's Candidate  
Technical University of Munich, Germany

Walter Stechele  
Professor  
Technical University of Munich, Germany  
[Walter.Stechele@tum.de](mailto:Walter.Stechele@tum.de)

ビデオベースの運転支援システムでは、さまざまな状況に応じて複雑なアルゴリズムをリアルタイムで処理する必要があります。しかし一般的な車載環境で利用できるハードウェアの性能を考えると、このようなシステムをソフトウェアで実装しても十分なリアルタイム性を得ることができないため、ハードウェア アクセラレーションが必要となります。

ASIC や規格品の ASSP といった専用ハードウェア回路では高度なリアルタイム処理が可能です。柔軟性に問題はありません。しかも、ASIC は近年、設計期間と開



発コストが増大していることに加え、運転支援用のビデオ アルゴリズムは標準化されておらず、設計変更が頻繁に発生するため、ASIC は適しているとは言えません。

その点、プログラミング性を備えた FPGA なら、市場投入までの期間短縮、コスト削減、製品差別化などの点で非常に有利です。ただ、同じプロセス テクノロジーの場合 ASIC と FPGA では、実現できる論理機能の量に大きな差があります。

ミュンヘン工科大学の集積システム研究所 (Institute for Integrated Systems) で私はほかの者とともに、エンベデッド システムのマルチプロセッサ SoC (MPSoC) アーキテクチャの研究に携わっています。また、この他にも、NoC (Network-on-Chip) やメモリ階層など、MPSoC の初期設計空間の探索を目的とした高レベルのシミュレーション手法についても研究を行っています ([www.lis.ei.tum.de](http://www.lis.ei.tum.de))。これら研究の一環として、高いリアルタイム処理性能とあらゆる状況の変化への適応性を備えたビジョンベースの運転支援システム向けに、リコンフィギュレーション可能な FPGA ベースの SoC アーキテクチャの開発に着手しました。

## アーキテクチャのコンセプト

今回開発した AutoVision アーキテクチャは、実行時に構成が変更できるハードウェア アクセラレータ エンジンを使用して未来型のビデオベースの運転支援システムを構築しようというものです。AutoVision システムでは、実行時に変更されるのはデバイスのごく一部のみです。今回の研究の柱となっているのは、ザイリンクスの FPGA を使用しているという点です。この製品にはダイナミック パーシャル リコンフィギュレーション (DPR) という画期的な機能が搭載されており、ザイリンクスの ISE®、EDK、PlanAhead™ などのエンベデッド デザイン ツールも用意されています。なお、アーキテクチャの詳細な説明に移る前に、今回のアプローチはザイリンクスから広範な支援を受けつつ、大学での独立した研究として実施したことをお

断りしておきます。

ビデオベースの運転支援システムでは、高いリアルタイム性能を得るために高度な柔軟性と本質的な並列性が必要となるため、動的に構成を変更できるハードウェアが最善の選択肢となります。ビデオ処理アルゴリズムは、高レベルのアプリケーションコードと低レベルのピクセル処理の 2 つに大別できます。高レベルのアプリケーションコードには非常に高い柔軟性が求められるため、PowerPC® や MicroBlaze™ などのエンベデッド CPU を使用したインプリメンテーションが適しています。一方、ピクセル処理では多数のピクセルに対して同じ処理の並列実行が必要となるため、ハードウェア アクセラレーションの使用が適しています。

このシステムでは、さまざまな運転状況に対処するために数種類のビデオ処理アルゴリズムを使用する必要があります。アルゴリズムの種類が異なると、ピクセル処理に使用するハードウェア アクセラレータ エンジンも異なるため、適切なアクセラレータをシステムの動作時に AutoVision チップにロードするようにします。その際、使用しないアクセラレータは FPGA から除いてチップの占有面積を抑えます。

DPR は一部の商用 Virtex® ファミリーに搭載された機能で、今回の研究プロジェクトではその中から Virtex-5 を選択しました。また、この機能は、日中 / 夜間運転、前進 / バック、運転速度の違い (都心部 / 幹線道路)、天候 (雨、霧、雪など) といった条件下で大きな効果を発揮します。

## 論理リソースの課題

リコンフィギュレーション可能なシステムを構築するには、リコンフィギュレーション可能なインターフェイス、ICAP (Internal Configuration Access Port) コントローラ、ビットストリーム転送用の高速インターコネクトという 3 つの論理リソースを追加する必要があります。これらリソースの増加分は、デザイン プロセスの後工程で DPR によって補われます。

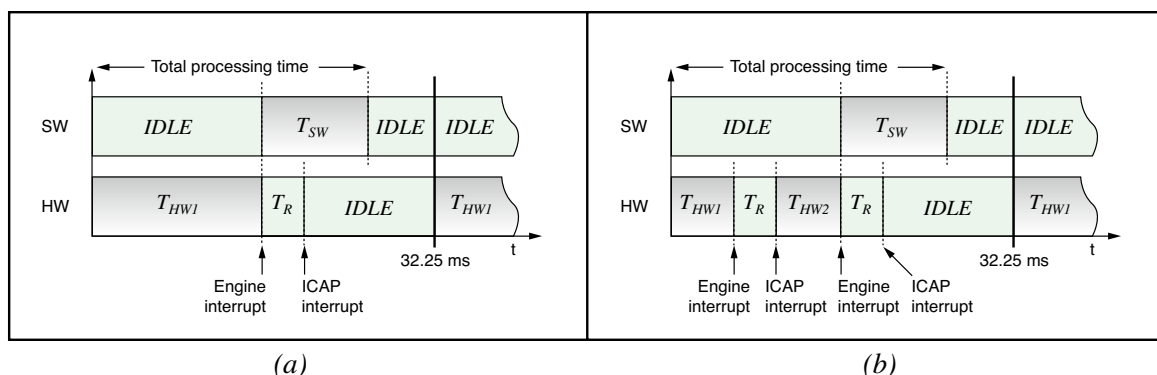
ICAP はコンフィギュレーション メモリに対してリードおよびライト アクセスを実行するためのもので、オンチップのセルフ リコンフィギュレーションに使用されます。リコンフィギュレーション可能なインターフェイスはバス マクロで構成されており、これによってリコンフィギュレーションの前、途中、後でリコンフィギュレーション可能な部分とそうでない部分との安全な接続が保証されます。現在はハードワイヤのマクロを使用していますが、今後、ザイリンクスの開発ツールでユーザーの介入なしに安全な接続が可能になれば、これらのマクロは不要となります。

第一に重要なのは、システムをリコンフィギュレーション可能にするためのデザイン変更を最小限にすることです。ICAP コントローラやその他の DPR 関連ロジックでデバイスのブロック RAM が大量に (20 以上) 占有されると、DPR の利点はなくなってしまいます。したがって、パーシャル ビットストリームをオンチップの BRAM に格納するという選択肢は不可能です。

同時に、リコンフィギュレーションの時間を最短にすることも必要です。絶対的な安全性が求められる状況でビデオベースの運転支援システムを使用する場合、DPR が原因でビデオ フレームを完全にキャプチャできないことがあってはなりません。ビットストリーム データから ICAP コントローラへの高速転送を確実にするため、ICAP コントローラを Native Port Interface 経由で直接 MPMC (Multi-Port Memory コントローラ) に接続しました。リコンフィギュレーション可能なインターフェイスとしてのバス マクロ、セルフ リコンフィギュレーションのための ICAP コントローラ、そして MPMC のポートを追加することは、いずれも DPR のオーバーヘッドであるといえます。なお、リコンフィギュレーションのプロセスは、リコンフィギュレーションの回数と実行タイミングに応じ、InterVFR (ビデオ フレーム間リコンフィギュレーション) と IntraVFR (ビデオ フレーム内リコンフィギュレーション) の 2 つに分類されます。

InterVFR プロシージャとは、連続す

図 1 - リコンフィギュレーションの回数と実行タイミングに応じて 2 つに分類されるリコンフィギュレーションのプロセス: (a) InterVFR (ビデオ フレーム間リコンフィギュレーション) と (b) IntraVFR (ビデオ フレーム内リコンフィギュレーション)



る 2 つのビデオ フレーム間でリコンフィギュレーション可能なモジュールを切り替える方式です。図 1(a) では、最初のフレームのエンジンによるイメージ処理時間を「THW1」、2 番目のフレームのエンジンによるイメージ処理時間を「THW2」と表記しています。「TR」は、モジュールを切り替える際のリコンフィギュレーション時間で、これにはリコンフィギュレーション可能な部分をブランク ビットストリームでクリアする時間と、別のパーシャル ビットストリームで新しいモジュールをロードする時間が含まれます。 $THW1 + TR + TSW < 32.25\text{ms}$  (31fps の場合) であれば、1 つのイメージをエンジンで処理した後にデバイスのリコンフィギュレーションを実行しても THW2 の開始に間に合うため、InterVFR が可能です。

一方、IntraVFR では 1 つのビデオ フレーム内で複数のリコンフィギュレーション可能なモジュールを切り替えます。たとえば、FPGA ベースのオプティカル フロー計算 [1] などこの方式が使用されます。この計算では 2 つのエンジンを使用しますが、逐次的に実行する必要があるためパイプライン実行はできません。最初のエンジンによるイメージ処理時間を「THW1」、2 番目のエンジンによる処理時間を「THW2」と表記しています。両エンジンのパーシャル ビットストリームのサイズは同一とします。したがって、両エンジンのリコンフィギュレーション時間も同

一となり、これを図 1(b) では「TR」(ブランクおよびモジュールのビットストリーム)と表記しています。 $THW1 + THW2 + 2 \cdot TR < 32.25\text{ms}$  (31fps の場合) であれば、IntraVFR が可能です。

ベスト ケースのリコンフィギュレーション時間 TR は、中間結果を CPU で処理する時間 TSW と完全にオーバーラップします。図 1(a) でも TR と TSW は完全にオーバーラップしているため、InterVFR では TR の時間を特別に確保する必要はありません。また、2 つの逐次実行エンジンを切り替える IntraVFR でも、2 回目のリコンフィギュレーション時間 TR については同様のことが言えます。つまり、2 つの TR のうち、1 つのビデオ フレームの処理時間に影響するのは、最初の TR のみです。InterVFR または IntraVFR システムをインプリメントする前に、設計準備が必要となります。

私達は、アルゴリズムのソフトウェア (SW) プロトタイプを作成後、コードをプロファイリングして特に高いパフォーマンスが要求される部分を特定しました。このような部分は、ハードウェア (HW) で実行するように変更する必要があります。ピクセル処理用のハードウェア アクセラレータは、構築ブロック形式の一連のモジュラー型ピクセル処理を利用してインプリメントしました。つまり、モジュラー型の Auto-Vision コンセプトではデータ入力、中間データ、データ出力のパスを再利用できる

ため、単一のピクセルとその周辺に対する処理のみを新規にインプリメントすればよいことになります。

ハードウェアとソフトウェア間のインターフェイスが定義されているため、エンベデッド CPU 上で実行する高レベル

のアプリケーション コードと並行してハードウェア アクセラレータのインプリメンテーションが実行できます (HW/SW 協調設計)。

HW アクセラレータのシミュレーションとテスト後、必要となる IP コアとインターコネクトすべてを含むリコンフィギュレーション不可のシステムを作成しました。このシステムは、パーシャル ビットストリームの作成前にシステム全体が正しく動作することを検証するために使用します。正常に動作するようであれば、PlanAhead ツールを使用してフロアプランニングを行い、デザイン ルール チェック (DRC) を実行します。

最後に、リコンフィギュレーション ツール (MAP と配置配線の変更) の早期評価版を使用してパーシャル ビットストリームを生成します。このパーシャル ビットストリームは、開発ボード上の CompactFlash カードに格納します。そして、システム初期化の時点で CompactFlash から DDR または DDR2 SDRAM のメインメモリ (ビデオ フレーム バッファとして使用) にビットストリームをコピーすることで、このデータへ高速でアクセスできるようにします。リコンフィギュレーションプロセスがトリガされると、約 20 サイクル後にパーシャル ビットストリームの最初の 4 バイトが ICAP 入力に現れます。

ハードウェア アクセラレータ エンジンは、リコンフィギュレーション前にジョブ



を完了しておく必要があるため、リコンフィギュレーション スケジュールを明確に決定しておくことが重要です。AutoVision システムでは、IP コアの通信はすべて割り込みを用いて行います。ビデオ フレームがメイン メモリに完全に転送されると、ビデオ入力からの割り込み信号によって PowerPC や MicroBlaze などのエンベデッド CPU に通知されます。

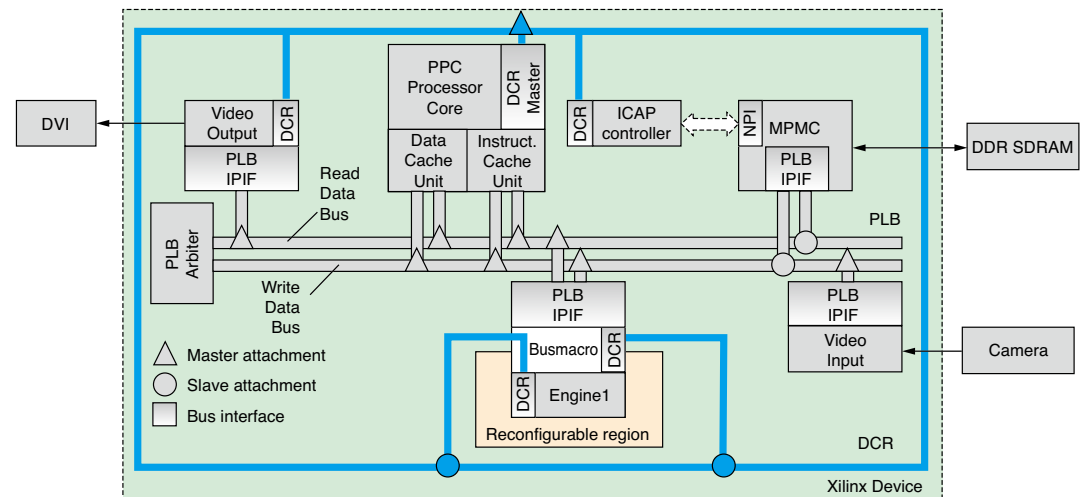
割り込み通知の受領後、CPU は即座に HW

アクセラレータに開始信号を送信します。このとき、ピクセル データおよび CPU への割り込みのフェッチ先メモリ アドレスも同時に送信されます。アクセラレータ エンジンの処理中にリコンフィギュレーション トリガを受信すると、システムはエンジン割り込みを受信した直後にリコンフィギュレーションを実行します。リコンフィギュレーション トリガの受信がなかった場合（またはリコンフィギュレーション処理が開始した直後の場合）は、エンベデッド CPU が高レベルのアプリケーション コードを実行します。

リコンフィギュレーションを開始する前に、CPU はバス マクロとリコンフィギュレーション可能な部分をシステムのほかの部分から切り離すためにコマンドを送信します。ICAP コントローラはダイレクト メモリ アクセス (DMA) とバースト転送を使用し、CPU を経由せずに転送を開始できます。CPU からは、パシシャル ビットストリームが格納されているメモリ アドレスと ICAP コントローラへのバースト転送の回数のみが送信され、リコンフィギュレーション プロセスが開始します。

高レベルのアプリケーション コードは並列に実行されるため、リコンフィギュレーション スケジュールに専用の時間枠を割り当てる必要はありません。新しいエ

図 2 - AutoVision システムのブロック図： ICAP コントローラは Native Port Interface を経由して MPMC に接続

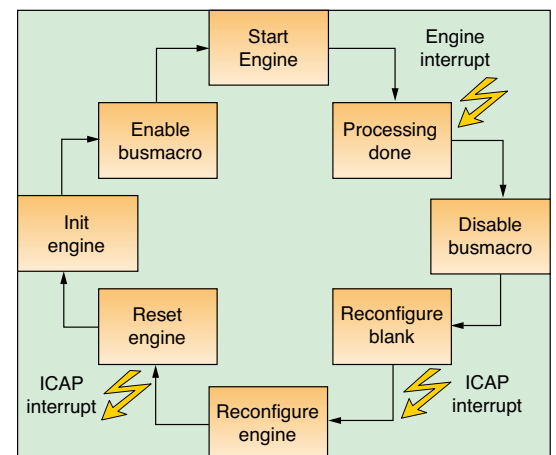


ンジンをロードする前に、ブランク ビットストリームを使用してパシシャルリコンフィギュレーション可能リージョン (PRR) をクリアします。そして、モジュールが消去されたことを ICAP コントローラから CPU に割り込みで通知すると、最後の処理に移ります。

リコンフィギュレーションの最後の処理では、ICAP コントローラが新しい HW アクセラレータを含むパシシャル ビットストリームをフェッチし、デバイスの PRR に対してリコンフィギュレーションを実行します。IntraVFR の場合は 1 つのフレーム内で 2 つのモジュールを切り替え、それぞれにブランク ビットストリームを使用して PRR をクリアするため、ビデオ フレーム 1 つ当たり合計 4 回のリコンフィギュレーションが実行されることになります。

リコンフィギュレーション実行中は、すべてのレジスタを定義された状態にするために PRR はリセット状態に保持されます。リコンフィギュレーションが完了する

図 3 - 明確に定義されたリコンフィギュレーションの実行タイミング



とシステムは即座にリセットを無効にし、エンジンのコンフィギュレーション レジスタを初期化します。最後に、新しいエンジンが動作を開始します。これら一連の手順を図 3 に示します。

リコンフィギュレーション時間を最短にするため、私達は ICAP にデータを入力する際の周波数をどこまで引き上げることができるかを調査しました。Virtex-5 ML507 FPGA では、コンフィギュレーション データを ICAP にブッシュする

周波数を 300MHz に設定しています。ICAP の入力幅は 4 バイトで、ザイリンクスの仕様では ICAP の最大周波数は 100MHz となっています。今回は、リコンフィギュレーション時間を最短にするために ICAP のオーバークロックが必要となりました。

ここでの目標は、ICAP にクロック サイクル当たり 4 バイトのデータを供給することです。現在の ICAP コントローラは動作周波数が 300MHz であるため、メイン メモリから ICAP コントローラの Virtex-5 FIFO プリミティブに十分なビットストリーム データを供給するには対策が必要となります。そこで、制御 FSM (有限状態マシン) を使用し、ICAP コントローラを Native Port Interface 経由で MPMC に接続する方法で、常に十分なデータを FIFO に供給できるようにしました。

MPMC にポートを 1 つ追加するには、データの一時バッファとして追加 FIFO が 1 つ必要になります。仕様外の動作周波数でもリコンフィギュレーションが正確に実行されることを確認するため、リコンフィギュレーションと並行してオンラインで検証を行うようにしています。ビットストリーム データが ICAP にプッシュされている間、システムは並行して巡回冗長チェック (CRC) を実行し、ビットストリームの整合性を確認します。また、ICAP は

リコンフィギュレーション プロセスの成功または失敗を示すワードを出力します。ICAP の出力と CRC IP から得られた検証情報をマージすることにより、リコンフィギュレーション プロセスに関する情報が実行時に生成可能となっています。700 万回以上リコンフィギュレーションを実行し

ましたが、エラーは一度も検出されませんでした。

## デモ アプリケーション

私達は、InterVFR の実証実験システムをザイリンクスの Virtex-II Pro デバイス

図 4 - 異なる環境での AutoVision システムの出力：晴れた日中での特徴となる点の検出と動き領域抽出 (左上)、トンネル入り口でのコントラスト強調 (右上)、トンネル内でのテールライト検出 (左下)、オプティカル フローによる都心部での動き検出 (右下)。

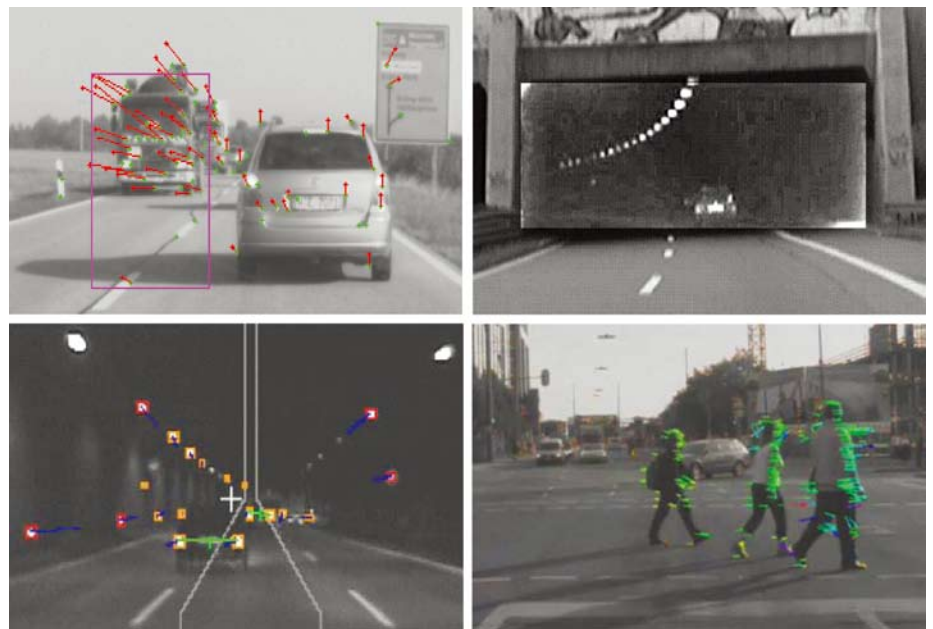
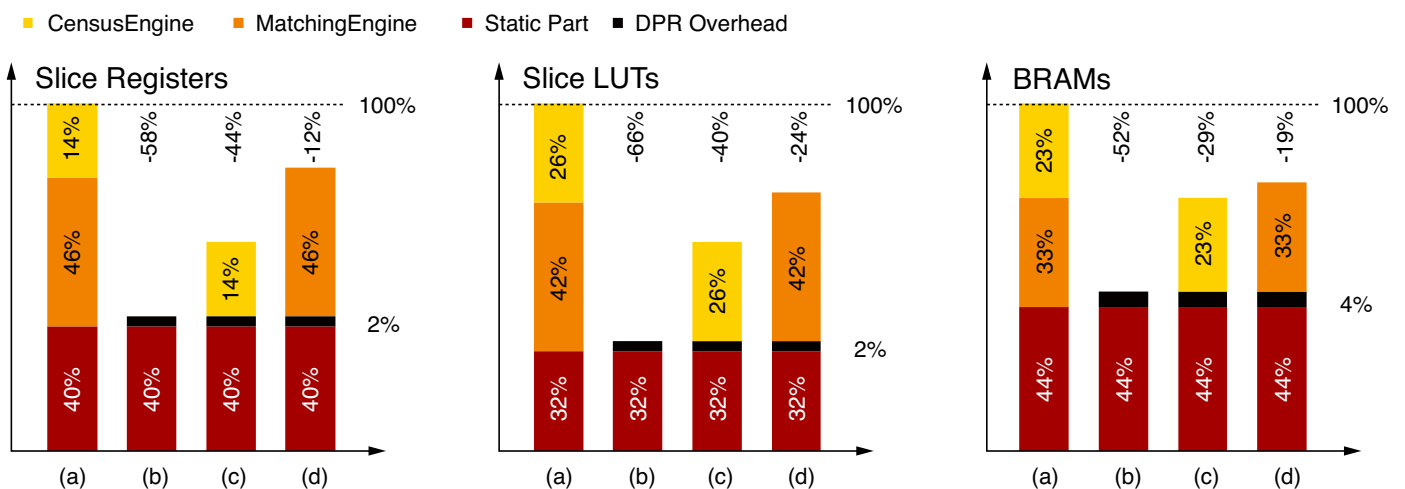


図 5 - リコンフィギュレーション可能なシステム (ワースト ケース) と不可能なシステムを比較した場合のリソース削減率



を搭載した XUPV2P ボードに実装しました。このデモでは、次のシナリオを想定しました。

まず、晴れた日中に幹線道路を運転します。ここでは、システムがほかの車の輪郭上の点を検出します。次に、車がトンネル入り口に近づきます。この状況では、ほかの車の輪郭上の点を検出するよりも、暗いトンネル入り口を注目すべき領域として識別し、この領域内でコントラストを強調した方が理にかなっていることとなります。トンネル内では、車のテールライトが重要な点となるため、トンネル内の照明と区別する処理が必要となります。このシナリオの詳細は、AutoVision システムを説明した 2007 年の記事 [2] を参照してください。

IntraVFR に関しては、現在、ザイリンクス ML507 開発ボードにプロトタイプを実装して実証実験を行っています。オプティカル フローの HW アクセラレータの詳細は、2009 年に発表した論文 [1] を参照してください。ここで紹介するオプティカル フローは、歩行者などの移動物体を検出することを目的としていますが、これは参考例として示したものに過ぎません。図 4 に、AutoVision システムの出力例を示します。

オプティカル フローの HW アクセラレータは、2 種類のエンジンで構成されています。1 つは CensusEngine で、イメージ内のすべてのピクセルとその周囲 15x15 ピクセルの領域を、この領域内の輝度分布を表すシグネチャに変換します。CensusEngine が出力するイメージは入力イメージと同一サイズですが、グレースケールのピクセル値の代わりにシグネチャ情報が含まれ、一般にセンサス イメージと呼ばれます。

このシステムでは、連続する 2 つのセンサス イメージをもう 1 つの HW アクセラレータである MatchingEngine にロードします。このエンジンは、連続する 2 つのイメージの特徴を比較し、対応する特徴の有無を判断します。対応する特徴がある場合は、一定のサーチ範囲（現在は 15x15 ピクセル）内でピクセルの動き

が判定されます。そして動きがあると判定されると、出力イメージに動作ベクタが付加されます。これら 2 つの HW アクセラレータは逐次的に実行されるため、IntraVFR を利用して 1 つのビデオ フレーム内で切り替えることができます。

最後に、InterVFR と IntraVFR はどちらもハードウェアのデモとしてインプリメントできることを述べておきます。IntraVFR のデモ機では、フレーム レート 31fps の場合、1 秒当たりのリコンフィギュレーション回数 (rps) は合計で  $(31 \times 4 \Rightarrow) 124$  となります。InterVFR と IntraVFR のデモは 1 つに統合できます。

## 結果とリソース

図 5 に、IntraVFR システムのリソース使用量を示します。ここでは、(a) リコンフィギュレーション不可のシステム、(b) ブランク モジュールのロードおよびリコンフィギュレーション可能なシステム、(c) CensusEngine のロードおよびリコンフィギュレーション可能なシステム、(d) MatchingEngine のロードおよびリコンフィギュレーション可能なシステムの 4 つについて、レジスタ、LUT、BRAM の使用量を比較しています。いずれの場合も、両エンジンを静的にインプリメントした (a) のシステムのリソース使用量を 100% としています。

この図からも分かるように、IntraVFR では、レジスタが 12%、LUT が 24%、BRAM が 19%、それぞれ使用量が低減されています。これは大きな低減率ではないように思えますが、切り替えるモジュールの数が 2 つというのはワースト ケースであることに注意してください。今回の AutoVision プロジェクト（輪郭上の点検出、夜間やトンネル入り口でのコントラスト強調、テールライト/ヘッドライト検出）のように、ハードウェア アクセラレータの数が 3 つ、4 つ、5 つ、またはそれ以上になると、論理リソースの使用量は大幅に削減される可能性が出てきます。

また、さまざまな天候状況に応じてビデオ処理を実行するには、相互排他的状

況に対応した新しい HW アクセラレータが必要となります。ICAP には動作周波数 300MHz でデータを供給しているため、ビットストリーム データのスループットは 1.2GB/s に達しています。このスループットで 1.1MB のパーシャル ビットストリーム (CensusEngine と MatchingEngine のパーシャル ビットストリームのサイズ) を使用した場合、デバイスのリコンフィギュレーションには  $1.1\text{MB} / 1.2\text{GB/s} = 0.92\text{ms}$  要します。

今後は、リコンフィギュレーションを実行した場合の消費電力がリコンフィギュレーション不可のシステムと比較して増減する量や、計測調査を継続していく予定です。1 時間に行うリコンフィギュレーションの回数が少なければ、DPR が全体的な消費電力に与える影響はわずかです。しかし 124rps というレートでリコンフィギュレーションを行った場合は、消費電力への影響は無視できません。リコンフィギュレーションを行わない場合は、リコンフィギュレーションに関連するレジスタへの電力供給をクロック ゲーティングで遮断することにより、全体的な消費電力の削減を図ることが理想です。

このほか、DPR 動作の信頼性を確認するために継続的な検証も必要です。DPR を車載環境で使用可能とするためには、数多くの検証が必要ですが、この手法は車載環境、特にビデオベースの運転支援システムに最大限の柔軟性をもたらし、全体的なコストも削減できる可能性を持っています。●●●

## 参考資料

- [1] C. Claus, A. Laika, L. Jia and W. Stechele, "High-performance FPGA-based optical flow calculation using the census transformation," in Proceedings of IV'09, Xi'an, China, 2009.
- [2] C. Claus, W. Stechele and A. Herkersdorf, "AutoVision: a run-time reconfigurable MPSoC architecture for future driver assistance systems," Information Technology Journal, vol. 49, no. 3, pp. 181–187, 2007.



# ザイリンクス トレーニング スケジュール '10年5月～'10年7月

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	受講料	5月	6月	7月
<b>FPGA デザイン</b>				
ISE デザイン ツール フロー	無償	*各販売代理店にて実施中 (詳細は下記より各社 Web サイト参照)		
FPGA 設計導入	無償	*各販売代理店にて実施中 (詳細は下記より各社 Web サイト参照)		
FPGA 設計実践	70,000円 (税別)	20日(木)～21日(金)	—	27日(火)～28日(水)
アドバンスド FPGA 設計	70,000円 (税別)	20日(木)～21日(金)	—	13日(火)～14日(水)
Virtex-6 ファミリ デザイン	無償 キャンペーン中	25日(火)～26日(水) 27日(木)～28日(金)	—	6日(火)～7日(水)
Spartan-6 ファミリ デザイン	無償キャンペーン中	20日(木)～21日(金)	—	20日(火)～21日(水)
PlanAhead を活用したタイミング クロージャ	70,000円 (税別)	13日(木)～14日(金)	3日(木)～4日(金)	1日(木)～2日(金) 29日(木)～30日(金)
ChipScope Pro を活用したデバッグング <span>7月新規開設!</span>	70,000円 (税別)	—	—	8日(木)～9日(金)(*予定)
<b>DSP デザイン</b>				
System Generator for DSPを 使用したDSPデザイン	70,000円 (税別)	25日(火)～26日(水)	22日(火)～23日(水)	20日(火)～21日(水)
<b>エンベデッド デザイン</b>				
エンベデッド システム開発	70,000円 (税別)	—	8日(火)～9日(水)	6日(火)～7日(水)
アドバンスド エンベデッド ソフトウェア開発	70,000円 (税別)	27日(木)～28日(金)	17日(木)～18日(金)	15日(木)～16日(金)
<b>コネクティビティ デザイン</b>				
MGTシリアル I/O デザイン	70,000円 (税別)	—	17日(木)～18日(金)	21日(水)～22日(木)
LogiCORE PCI Express システム開発 <span>6月新規開設!</span>	70,000円 (税別)	—	24日(木)～25日(金)	22日(木)～23日(金)

\*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://japan.xilinx.com/support/education-home.htm>

## ザイリンクス販売代理店オリジナル トレーニング

販売代理店各社のオリジナル トレーニングの内容およびスケジュールは、各社の Web サイトをご覧ください。

東京エレクトロン デバイス	<a href="http://ppg.teldevice.co.jp/">http://ppg.teldevice.co.jp/</a>
アヴネット ジャパン	<a href="http://www.avnet.co.jp/services/Training/index.asp">http://www.avnet.co.jp/services/Training/index.asp</a>
新光商事	<a href="https://xilinx.shinko-sj.co.jp/training/index.html">https://xilinx.shinko-sj.co.jp/training/index.html</a>
PALTEK	<a href="http://www.paltek.co.jp/seminar/index.htm">http://www.paltek.co.jp/seminar/index.htm</a>

# サンプリングが更に高速！

RF信号をダイレクトに記録・再生

PCDAQ: Data Acquisition System  
**iDAQS**

**高速DACモジュール：DA2500**  
2500MSPS, 14bits, 2ch  
XILINX, Virtex-5 FPGA



**高速ADCモジュール：AD14400**  
400MSPS, 14bits, 2ch  
XILINX, Virtex-5 FPGA



**データ収集再生装置：PCDAQ-Mini**  
800MB/sec 記録・再生速度  
1.1TB RAID (標準)

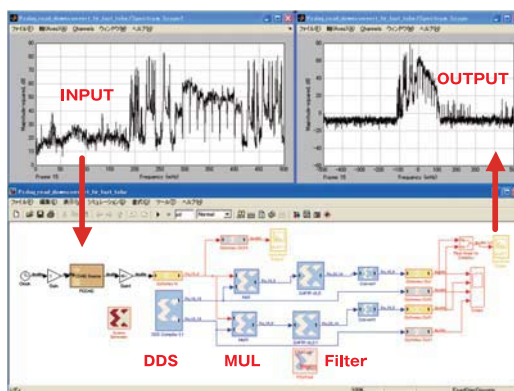


## データ収集再生装置：PCDAQシリーズ

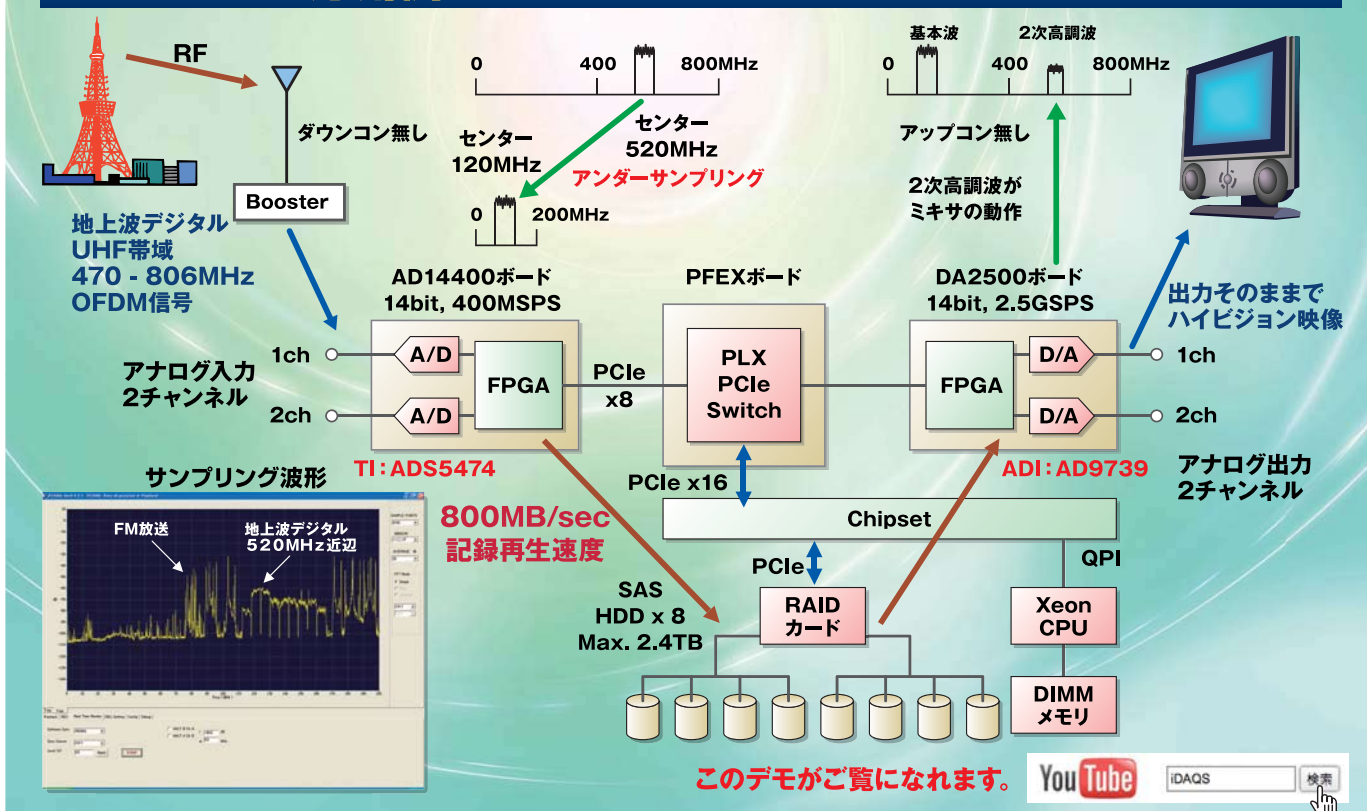
- 特徴：
- AD: 400MSPS, DA: 2500MSPSの高速サンプリング
  - RAIDディスクに800MB/secの長時間記録再生
  - FM放送、地上波デジタルのRF信号を記録再生  
ダウン/アップコンバータのアナログ変換無しで映る
  - SysGenが簡単動作するツールを提供

用途：デジタル信号処理の研究・開発  
地デジ、GPS、VICS、レーダ、MIMO、コグニティブ無線

**FPGA設計ツール：iBB**  
PCIe-DMA IP for System Generator  
Interface for MATLAB/Simulink



## 応用例：UHF ダイレクトサンプリング



株式会社 **アイダックス**

〒167-0042 東京都杉並区西荻北 3-13-5 M&Iビル 2F TEL 03-6762-1210 FAX 03-6762-1211

[www.idaqs.jp](http://www.idaqs.jp)



# Improving DDR SDRAM Efficiency with a Reordering Controller

## リオーダー方式のコントローラによる DDR SDRAM の効率化

実際のワークロードで抜群の持続転送レートを実現する Virtex-6  
メモリ コントローラ

Leith Johnson  
Managing Member  
Nokhu Systems LLC  
[leith@nokhu.com](mailto:leith@nokhu.com)

通常、高速 DDR (Double-Data-Rate) SDRAM のメーカー公称値はピーク データ転送レートを表しています。たとえばこの値が DDR3-1600 の場合、デバイスは最大で毎秒 1,600M 回の転送が行えることをメーカーが仕様として規定していることになります。

もちろん、この転送レートは実現可能な値ではありますが、実際のワークロードでこのレートが持続されません。これは、行アドレスの競合、データバスのターンアラウンド ペナルティ、ライト リカバリなどにより、実際の転送レートがピーク転送レートを下回るためです。

しかも、これらの要因による転送レート低下の影響は、高速化が図られた新しい世代の SDRAM ほど深刻になっています。このため、単純なインオーダー（並べ替えなし）のスケジューリング アルゴリズムを採用したメモリ コントローラでは、持続スループットが公称値のピーク転送レートをはるかに下回る場合が多く見られます。これに対し、より高度なリオーダー（並べ替え）方式でのスケ

ジューリングでは、転送レート低下の要因を克服し、実際のアプリケーションで高い持続転送レートが実現できます。

ザイリンクスは Virtex®-6 で初めて FPGA に最適化したリオーダー方式の DDR SDRAM メモリ コントローラを採用しました。この新しいコントローラによって、Virtex-6® ユーザーは容量、性能、電力効率が改善された最新世代の DDR SDRAM テクノロジを存分に活用できます。一見すると、DDR SDRAM コンポーネントは単純なリード/ライト メモリに過ぎませんが、最新の DDR SDRAM は非常に複雑なデバイスとなっています。DDR SDRAM コントローラは、多数のタイミング要件を満たしながらアドレス、コマンド、データのシーケンスをきわめて正確に生成しなければならず、高い性能を実現するには、最小許容タイミングでコマンドをパイプライン処理する必要があります。

## DDR SDRAM の詳細

では、メモリの性能が低下する仕組み、そしてピーク転送レートが高くなるほどその影響が深刻になる原因を考えていきましょう。

図 1 に示したように、基本的な DDR SDRAM アクセスでは、メモリ コントローラがメモリに対してアクティベート コマンドと行アドレスを送信し、RAS-to-CAS レイテンシ (行アドレス ストローブから列アドレス ストローブまでのクロック サイ

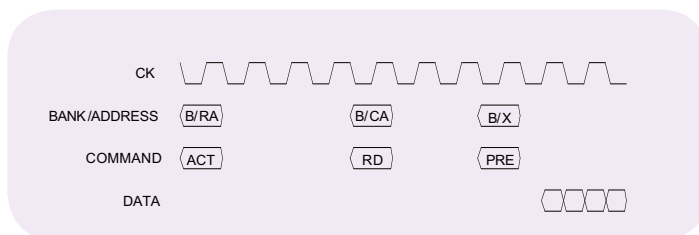
クル数として定義される) の後に列アドレスとリードまたはライト コマンドを送信します。さらに CAS レイテンシの後、リードの場合はデータをサンプ

リングし、ライトの場合はデータを供給して要求が完了します。データ転送が完了すると、コントローラはプリチャージ コマンドを発行してアクティブな行をクローズします。プリチャージ後に一定期間が経過すると、コントローラは再びアクティベート コマンドが発行できます。

その後、コントローラはプリチャージ-アクティベートのシーケンスなしに複数のリードまたはライト コマンドを連続して発行できます。これは、一般に「高速ページモード (fast page mode) アクセス」と呼ばれます。

高速ページモードは、多くの時間と電力を消費するアクティベート-プリチャージのシーケンスがないため非常に高効率になります。このモードを利用できるのは同じ行アドレスへのアクセスが連続する場合のみです。ワークロードの中に異なる行アドレスへのアクセスが混在する場合は、行アドレスが変わる部分でアクティベート-プリチャージのシーケンスを挿入しなければなりません。この場合、持続スループッ

図 1 - 基本的な DDR DRAM のリード サイクル



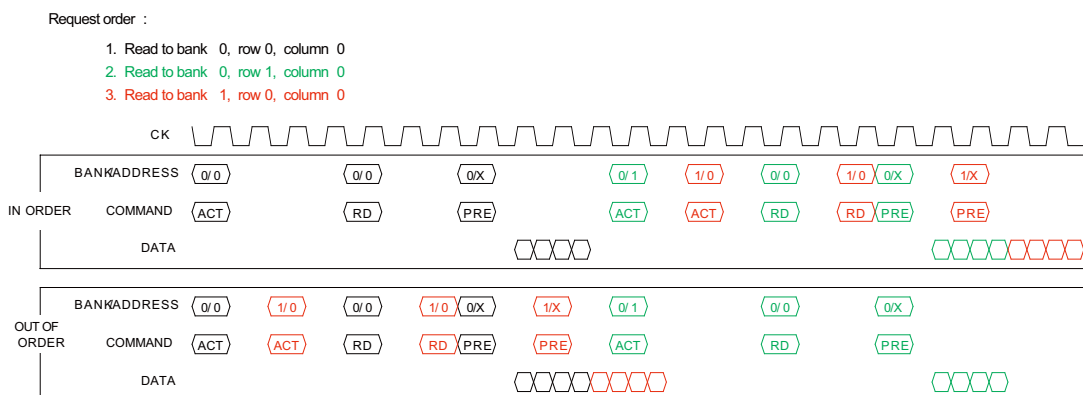
トはピーク転送レートよりもはるかに低くなります。

DDR SDRAM はいくつかの「バンク」で構成されています。各バンクは同一サイズで、互いにほぼ独立しています。DDR3 DRAM には 8 つのバンクがあり、異なるバンクへのアクセスはオーバーラップさせることができます。たとえば、あるワークロードでバンク 1 へのリード直後にバンク 2 へのリードが発生する場合、メモリ コントローラはまずバンク 1 にアクティベート コマンドと行アドレスを送信し、次にバンク 2 にアクティベート コマンドと行アドレスを送信できます。そして RAS-to-CAS レイテンシの後、コントローラはバンク 1 に列アドレスとリード コマンドを送信し、次にバンク 2 に列アドレスとリード コマンドを送信します。そして CAS レイテンシの後、これら 2 つのシームレスなデータ バースト転送を開始します。このような方法をローリング バンクモードと呼びます。シーケンシャルにバンクを切り替えてアクセスするようなワーク

ロードでは、このような方法を拡張してピーク転送レートを持続することも可能です。

高い DRAM 性能を維持するには、ワークロードが上述の高速ページモードかローリングバンクモードのどちらかになるように操作すればよいことになりますが、現実にはほとんど不可能です。一般的に、プロセッサが

図 2 - 並べ替えによるページ競合の解消





ら生じるワークロードは疑似乱数であり、このような操作は容易ではありません。

たとえば、同じバンク内の異なる行アドレスに連続してアクセスし（行アドレスの競合）、その直後に別のバンクにアクセスするようなワークロードの場合、単純なインオーダー方式のメモリコントローラではこれら3つのアクセスがすべて逐次的にスケジューリングされるため、2番目のアクセスのプリチャージ・アクティベートによって効率が大幅に低下します。

これに対し、リオーダー方式のメモリコントローラでは1番目のアクセスの有効化コマンドを送信後、3番目のアクセスのアクティベートコマンドを送信し、これらのアクセスをオーバーラップできるため、効率が向上します。

図2に、この概略を示します。上のパターンは3つの要求を順次実行したもので、下のパターンは3番目の要求を2番目の要求よりも先に実行したものです。この図から分かるように、シーケンスを並べ替えた方がインオーダーの場合よりも早く処理が完了します。

リオーダー方式のメモリコントローラは、行アドレスの競合による転送レートの低下を大幅に改善できるだけでなく、ライトリカバリおよびバスのターンアラウンドによる転送レート低下にも対応します。ライトサイクルの終了時、DRAM内部ではデータを実際にアレイに書き込む作業が行われており、ビジー状態となります。ライトアクセスが連続する場合はピークレートで処理できますが、ライトアクセスの後にリードアクセスを実行する場合は、DRAMアレイ内で書き込みが完了するまで待期する必要があります。この待期時間が、ライトリカバリ仕様となります。

DDR SDRAMは双方向の共有型バス構造をしています。このバスの

一方の端にはメモリコントローラ、もう一方の端には通常2～4つのDIMM (Dual In-line Memory Module) が接続されます。このバスの電氣的な長さは約5インチです。つまり、バス伝搬時間は約1ナノ秒となります。バスドライバが駆動を停止するとバスには必ずグリッチが伝搬し、このグリッチが安定するには、バス伝搬時間の2倍の時間が必要となります。

DDR SDRAMプロトコルには、DDR SDRAMの1クロックサイクルまたは2UI（ユニットインターバル）の長さのデータプリアンプルが組み込まれています。プリアンプルの間、データは転送されません。2つの連続するアクセスが1つのドライバによって駆動される場合、プリアンプルは省略できます。実際、プリアンプルを省略しなければピーク転送レートは達成できません。このライトリカバリとデータバスの問題は性能に大きな影響を与え、単純なインオーダー方式のメモリコントローラの場合、性能はワークロードによって大きく変化します。リード/ライトが交互に現れるアクセスパターンのワークロードの場合、性能はピーク転送レートの値を大幅に下回ります。

これに対し、リオーダー方式のメモリコントローラにはバスドライバごとにアクセスをグループ化する機能があります。単純なシングルランクコンフィギュレーションの場合、メモリコントローラはリードと

ライトをそれぞれグループ化してバーストモードで発行します。また、マルチランクコンフィギュレーションでは、各ランクのリードをグループ化してバーストで発行できるようにアクセスを整理します。

図3に、要求を並べ替えてバスターンアラウンドとライトリカバリのペナルティを解消した場合の効果を示します。この例では、バンク0へのシーケンシャルリードとバンク1へのシーケンシャルライトがインターリーブされています。2つのDMAマシンがメモリのリードとライトを行っている場合、このようなワークロードがよく発生します。

インオーダー方式の場合には、リードからライト、ライトからリードへと切り替わるたびにバス帯域幅が大幅に低下します。一方、リオーダー方式ではこれらのペナルティが発生しないため、スループットが格段に向上します。

トランザクションのリオーダーには、データの整合性と供給に関する問題が常に伴います。同じアドレスに対してリードの後にライトを実行すべきところで、ライトを先に実行してしまうとメモリ内容の整合性が損なわれるため、このような並べ替えは避ける必要があります。また、リードアクセスを先延ばしにするとデータの供給が滞るため、これも避けなければなりません。

さらに、一部のアプリケーションによってはメモリコントローラから返されるデー

図3 - 並べ替えによるライトリカバリおよびバスターンアラウンドのペナルティ解消

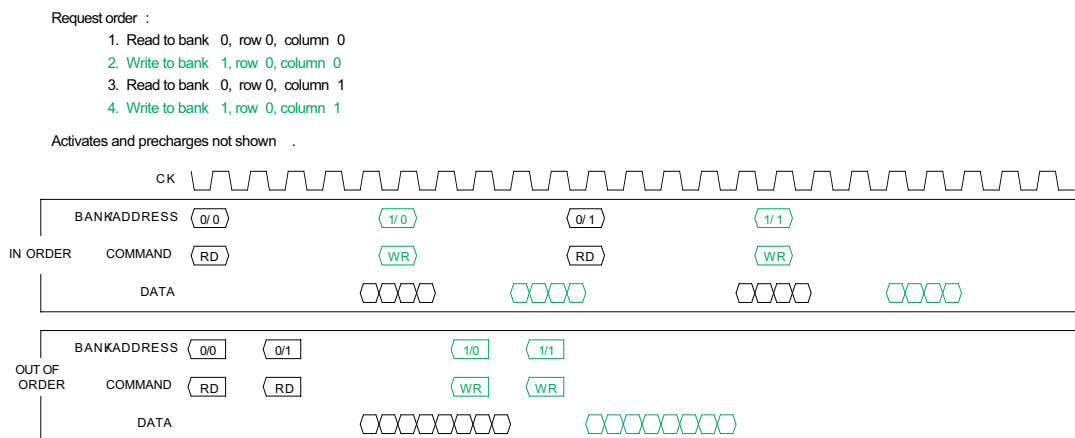




表 1 - DDR SDRAM の転送レートが大幅に上昇する一方で、基本的なデバイス特性はそれほど変化していない

	DDR1-6		DDR2-25		DDR3-15E	
ピーク転送レート (百万回 / 秒)	333		667		1333	
ユニット インターバル (ナノ秒)	3		1.5		0.75	
	ns	UIs	ns	UIs	ns	UIs
RAS サイクル	60	20	55	37	49.5	66
ライト リカバリ	15	5	15	10	15	20
2x tBUS_PROP	2.0	0.67	2.0	1.3	2.0	2.6

表 2 - 2 つのサンプル ワークロードにおいて、リオーダーを有効にするとメモリコントローラの効率が向上

	インオーダー	リオーダー
CPU-DMA ストリーム	36%	76%
リード/ライト交互	25%	63%

タの順序が変更されていることが問題となる場合があります。この問題は、リクエストの発行順にデータの順を戻すためのバッファをメモリコントローラに装備することで容易に修正できます。

一部のコーナーケースでは、リオーダーによって一部のリードアクセスのレイテンシが大きくなる場合もあります。しかし、リオーダーによって効率が向上するため、メモリコントローラの占有率が低下し、最終的に平均リードレイテンシは改善します。

表 1 に、各世代の DDR SDRAM がどのように高速化しているかをまとめます。ここでは、DDR1、DDR2、DDR3 の注目すべき仕様値を記載しています。ここに記載したスピードグレードは、ミッドレンジの量産品を反映したものです。

転送レートの数値は大幅に上昇していますが、RAS サイクルなどの基本的なデバイス特性はそれほど急激に向上しておらず、ライトリカバリに関してはまったく変化がありません。したがって、UI に対して正規化した相対的なペナルティは、世代が進むごとに急激に増大します。

DDR SDRAM のその他のパラメータも、スケーリングに関して同様の問題を抱えていますが、ここでは上述の 3 つの例を参考までに記載しています。

## インオーダーモードとリオーダーモード

では、実際に 2 つのサンプルワークロードを例に挙げ、新しい Virtex-6 DDR SDRAM メモリコントローラの効果を見てみましょう。このデバイスはインオーダーモードとリオーダーモードの両方に対応しているため、並べ替えの効果を容易に確認できます。

最初の例は、2 つのマスタをモデル化したものです。1 つは、CPU が行列を使用するコードシーケンスをあるステップ幅で実行しています。行列計算のステップ幅は、1 つのバンクに対して要求ごとに行アドレスを増加させるというメモリリクエストパターンを発生させています。つまり、1 つの DRAM バンク内の異なる行アドレスに対してリードアクセスが連続して発生するという最も効率の悪いパターンとなります。もう 1 つのマスタは DMA エンジンで、CPU とは別の DRAM バンクに対してシンプルなシーケンシャルリードアクセスを発生させます。

インオーダーとリオーダーの場合では、実際に実行される全体的な要求のストリームが異なります。CPU ストリームの実行

時間は、DRAM の行サイクルタイムの制約を受ける一方、DMA ワークロードは独立して処理を進めることができます。インオーダーの場合は、DMA ストリームは CPU ストリームとともに逐次的にスケジューリングされます。リオーダーの場合、インオーダーでは有効に利用されなかった DQ バスサイクルに DMA ストリームが挿入されます。CPU ストリームのスループットは DRAM の行サイクルタイムの制約を受けるため、あまり大きな改善はありません。しかし、リオーダーを有効にすると DMA ストリームの実行レートが高速化します。

2 番目の例は、1 つの DMA エンジンがシーケンシャルリードストリームを発行し、もう 1 つの DMA エンジンがシーケンシャルライトストリームを発行するというパターンをモデル化したものです。メモリコントローラは、リードリクエストとライトリクエストを交互に受け取るものとします。その他の競合を避けるためにリードはすべてバンク 0、行 0 をターゲットとし、ライトはすべてバンク 1、行 0 をターゲットとします。

これら 2 種類のワークロードを、いずれも最初はリオーダーを無効にして実行し、次にリオーダーを有効にして実行しました。メモリコントローラのパラメータはデフォルトのままとしています。例外としてリフレッシュ、ZQ、およびその他の DRAM メンテナンス機能は無効にしました。

表 2 は、これら 2 つのサンプルワークロードが定常動作となったところで一定時間にわたって測定した効率を比較したものです。効率は、ペイロードを含む UI の数を測定期間全体の UI の数で割って求めました。リオーダーを有効にすると、メモリコントローラの効率は飛躍的に向上します。

今後、DDR SDRAM が高速化すると基本的なデバイスパラメータとの差はますます大きくなるでしょう。最新の DDR SDRAM 世代の性能を最大限に引き出すには、今後のメモリコントローラに実装するスケジューリングアルゴリズムにも継続的な改良が求められます。

## How to Achieve Timing Closure on FPGAs

## FPGA のタイミングクロージャ

デザインの品質を最大限に高め、フィールドでの問題発生を不安を解消

```

nibble_proc : PROCESS (sys_clk_bufg)
BEGIN
    if rising_edge(sys_clk_bufg) then
        sys_reset = '1') then
            nibble_data_in <= "0000";
        elsif (divide_by_4 = '1') then
            nibble_data_in
            nibble_data_out <= nibble_data_in;
        end if;
    end if;
end process;

```

Nelson Lau  
Lead Hardware Engineer  
Spirent Communications  
nelson.lau@spirent.com

シミュレータでは問題なく動作したコードに、出荷後、再現性のないエラーが発生した経験はありませんか。あるいは、問題のなかったコードが、新しいバージョンのツールチェーンでコンパイルすると動作しなくなったということはないでしょうか。テストベンチを見直してテストカバレッジが100%であることを確認し、すべてのテストをエラーなく完了したのに、問題が発生するという事は意外に多いものです。

設計者は、コーディングとシミュレーションを重視し、FPGAのシリコン内部の動作にはそれほど注意を払わないことがあります。その結果、ロジック自体にエラーはないにもかかわらず、ロジック合成やタイミングの問題によってロジックの問題が発生するケースが見受けられます。

しかし、正しい手順で設計さえすれば、信頼性が高く予測可能なロジックを作成するFPGAコードの記述は困難なことではありません。

FPGAデザインでは、ロジック合成とそれに関連するタイミングクロージャはコンパイル時に実行されます。このコンパイルプロセスは、I/Oセル構造、非同期ロジック、タイミング制約など多くの要素から影響を受けるため、一連のツール処理を実行するごとに結果が異なることがあります。ここでは、こうしたばらつきをなくし、より短時間で確実にタイミングクロージャを達成する方法について詳しく解説します。

```

if rising_edge(sys_clk) then
    reset_1 <= reset;
    reset_2 <= reset_1 and reset;
    sys_reset <= reset_2 and reset_1 and reset;
end if;

```

```

if rising_edge(sys_clk) then
    if (sys_reset = '1') then
        data_in <= '0';
    else
        data_in <= sda;
    end if;
end if;

```



## I/O セル構造

一般に、FPGA の I/O ピンは自由にカスタマイズできるようになっています。このカスタマイズは、タイミング、駆動電流、終端など多くの要素に影響を与えます。したがって、I/O セル構造を明確に定義されていないと、ツールはデフォルト設定で処理を実行し、期待した結果が得られないことがあります。次に示す VHDL コードは、「sda: inout std\_logic;」宣言を使用して sda という名前の双方向 I/O バッファを作成するものです。

```
tri_state_proc : PROCESS (sys_clk)
BEGIN
    if rising_edge(sys_clk) then
        if (enable_in = '1') then
            sda <= data_in;
        else
            data_out <= sda;
            sda <= 'Z';
        end if;
    end if;
END PROCESS tri_state_proc;
```

合成ツールから見ると、このコードブロックには双方向バッファの実装方法について明確な指示がありません。このため、ツールの判断によってインプリメントされることになります。

実装方法としてまず考えられるのは、FPGA の I/O リングにある双方向バッファを使用するというもので、これは設計意図を正しく反映した方法です。2 つ目は、トライステートの出力バッファと入力バッファの両方をルックアップテーブル (LUT) ロジックに実装する方法です。そしてもう 1 つの可能性

として、I/O リングのトライステート出力バッファと LUT の入力バッファを使用する方法があります。おそらく、合成ツールの多くがこの方法で実装するでしょう。これら 3 つの方法で実装されるロジックはいずれも有効です。しかし、2 番目と 3 番目の方法では I/O ピンと LUT 間を信号が移動することによって配線遅延が増大します。また、これら 2 つの方法では、タイミングクロージャを達成するために必要なタイミング制約も多くなります。図 1 の FPGA Editor の画面を見ると、双方向 I/O の一部が I/O バッファの外に実装されていることが分かります。

このように、コードのクリティカルな部分については、実装方法を合成ツールの判断に任せるのではなく、定義する必要があります。合成後のロジックが意図したとおりであったとしても、合成ツールのバージョンによって異なる結果になる可能性があります。I/O ロジックなど、重要なロジックは設計者が明確に定義しておく必要があります。次に示す VHDL コードでは、ザ

イリンクスの IOBUF プリミティブを使用して I/O バッファを暗的に定義しています。また、バッファの電気的特性もすべて明確に定義されていることに注目してください。

```
sda_buff: IOBUF
```

```
generic map (IOSTANDARD => "LVCMOS25",
             IFD_DELAY_VALUE => "0", DRIVE => 12,
             SLEW => "SLOW")
```

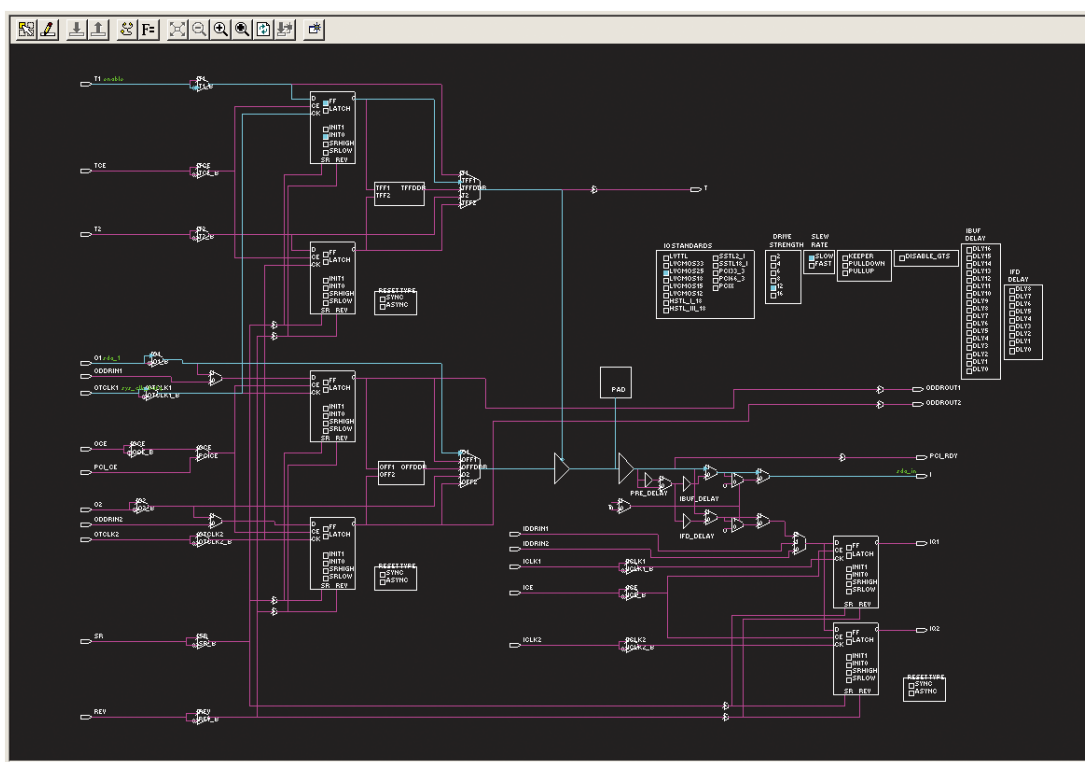
```
port map(o=> data_out, io=> sda,
         i=> data_in, t=> enable_in);
```

図 2 に示した FPGA Editor の画面からも分かるように、この双方向 I/O は完全に I/O バッファ内に実装されています。

## 非同期ロジックの落とし穴

非同期コードが含まれる場合、ロジックに制約を指定したり、シミュレーションやデバッグを実行することが難しくな

図 1 - FPGA Editor の画面：双方向 I/O の一部が I/O バッファの外に実装されています。





ります。通常、非同期ロジックのエラーは断続的にしか発生せず、再現性がほとんどありません。また、非同期ロジックに起因するエラーを検出できるようなテストベンチは、作成自体が不可能です。

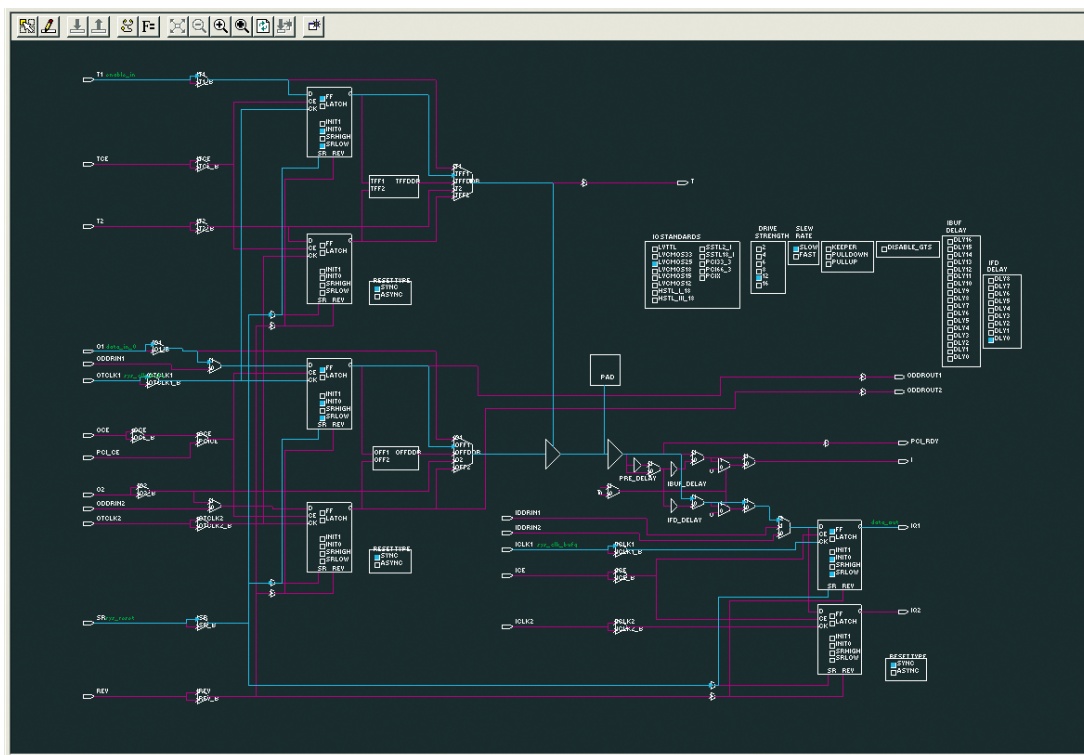
非同期ロジックの特定は一般に考えられているよりも難しく、検出できないことがしばしばあります。このため、設計者はデザインに紛れ込みやすい非同期ロジックのパターンをいくつも把握しておく必要があります。クロックに同期したロジックには必ず最小セットアップ/ホールドタイムが必要です。フリップフロップのリセット入力も例外ではありません。

次に示すコードは、非同期リセットを使用しています。これでは、フリップフロップのセットアップ/ホールドタイムの要件を満たすようにタイミング制約を与えることができません。

```
data_proc : PROCESS (sys_clk,reset)
BEGIN
    if (reset = '1') then
        data_in <= '0';
    elsif rising_edge(sys_clk) then
        data_in <= serial_in;
    end if;
END PROCESS data_proc;
```

次に示すコードは、同期リセットを使用しています。しかし、ほとんどのシステムでは、リセット信号はプッシュボタンスイッチなど、システムクロックとは関係のない信号です。リセットは長時間にわたってアサートまたはディアサートされ、ほとんどが静的な状態ですが、それでもレベルに変化があります。システムクロックの立ち上がりエッジに対してリセットがディアサートされると、フリップフロップのセット

図2 - VHDL コードを変更し、I/O ロジックと重要なロジックを明確に定義すると双方向 I/O は完全に I/O バッファ内に実装されます。



アップ タイム要件に違反することがありますが、これを制約する方法はありません。

```
data_proc : PROCESS (sys_clk)
BEGIN
    if rising_edge(sys_clk) then
        if (reset = '1') then
            data_in <= '0';
        else
            data_in <= serial_in;
        end if;
    end if;
END PROCESS data_proc;
```

このように、非同期信号は同期ロジックに直接入力できないという点さえ理解しておけば、問題は容易に解決できます。次に示すコードは、システムクロック sys\_clk に同期したリセット sys\_reset を作成します。非同期ロジックをサンプリングすると、メタスタビリティの問題が発生することがあります。この問題には、ラダー（はしご）型のサンプルを使用し、前段階との論理積をとることによって発生率を低くすることで対応できます。

```
data_proc : PROCESS (sys_clk)
BEGIN
    if rising_edge(sys_clk) then
        reset_1 <= reset;
        reset_2 <= reset_1 and reset;
        sys_reset <= reset_2 and reset_1
        and reset;
    end if;

    if rising_edge(sys_clk) then
        if (sys_reset = '1') then
            data_in <= '0';
        else
            data_in <= serial_in;
        end if;
    end if;
END PROCESS data_proc;
```

これで、すべてのロジックが同期ロジックになったとしましょう。しかし、ロジックとシステムクロックの同期は簡単に失われてしまうため、油断はできません。ここで注意すべきことは、システムクロックのローカル配線リソースの使用をツールチェーンに頼らないことです。この部分をツールに委ねると、ロジックに制約が適用

できなくなります。繰り返しますが、クリティカルなロジックはすべて明確に定義しておく必要があります。次に示す VHDL コードは、ザイリンクスの BUFG プリミティブを使用して sys\_clk を低スキューのネットを駆動する高ファンアウトの専用バッファに強制的に割り当てています。

```
gclk1: BUFG port map (I => sys_clk,0
=> sys_clk_bufg);
```

```
data_proc : PROCESS (sys_clk_bufg)
BEGIN
    if rising_edge(sys_clk_bufg) then
        reset_1 <= reset;
        reset_2 <= reset_1 and reset;
        sys_reset <= reset_2 and reset_1
        and reset;
    end if;

    if rising_edge(sys_clk_bufg) then
        if (sys_reset = '1') then
            data_in <= '0';
        else
            data_in <= serial_in;
        end if;
    end if;
END PROCESS data_proc;
```

デザインによっては、デシリアライズしたデータの処理に 1 つのマスタ クロックを分割する場合があります。例として、次の VHDL コードではシステム クロックの 1/4 のレートでデータをキャプチャする nibble\_proc というプロセスを示しています。

```
data_proc : PROCESS (sys_clk_bufg)
BEGIN
    if rising_edge(sys_clk_bufg) then
        reset_1 <= reset;
        reset_2 <= reset_1 and reset;
        sys_reset <= reset_2 and reset_1
        and reset;
    end if;

    if rising_edge(sys_clk_bufg) then
        if (sys_reset = '1') then
            two_bit_counter <= "00";
            divide_by_4 <= '0';
            nibble_wide_data <= "0000";
        else
            two_bit_counter
            <= two_bit_counter + 1;
            divide_by_4 <= two_bit_counter(0) and
            two_bit_counter(1);
```

```
nibble_wide_data(0)
<= serial_in;
nibble_wide_data(1)
<= nibble_wide_data(0);
nibble_wide_data(2)
<= nibble_wide_data(1);
nibble_wide_data(3)
<= nibble_wide_data(2);
    end if;
end if;
END PROCESS data_proc;

nibble_proc : PROCESS (divide_by_4)
BEGIN
    if rising_edge(divide_by_4) then
        if (sys_reset = '1') then
            nibble_data_in <= "0000";
        else
            nibble_data_in
            <= nibble_wide_data;
        end if;
    end if;
END PROCESS nibble_proc;
```

これですべてのロジックが同期したように見えますが、nibble\_proc はクロック ドメイン sys\_clk\_bufg からの nibble\_wide\_data をサンプリングする際に積項 divide\_by\_4 を使用していません。配線遅延があるため、divide\_by\_4 と sys\_clk\_bufg の間にははっきりと定義された位相関係がありません。divide\_by\_4 を BUFG に移動しても、やはり配線遅延が生じるため問題の解決にはなりません。これを解決するには、次に示すように nibble\_proc を sys\_clk\_bufg ドメインに維持したまま、divide\_by\_4 を修飾子として使用します。

```
nibble_proc : PROCESS (sys_clk_bufg)
BEGIN
    if rising_edge(sys_clk_bufg) then
        if (sys_reset = '1') then
            nibble_data_in <= "0000";
        elsif (divide_by_4 = '1') then
            nibble_data_in
            <= nibble_wide_data;
        end if;
    end if;
END PROCESS nibble_proc;
```

## タイミング制約の重要性

ロジックを正しく動作させるには、タイ

ミングを適切に制約する必要があります。すべてのコードを 100% 同期させ、すべての I/O にレジスタを付けるように注意しておけば、タイミング制約を指定することによってタイミング クロージャは大幅に簡略化されます。前述のコードをシステム クロック 100MHz で使用した場合、タイミング制約ファイルは次のようにわずか 4 行で記述できます。

```
NET sys_clk_bufg TNM_NET =
sys_clk_bufg;
TIMESPEC TS_sys_clk_bufg = PERIOD
sys_clk_bufg 10 ns HIGH 50%;

OFFSET = IN 6 ns BEFORE sys_clk;

OFFSET = OUT 6 ns AFTER sys_clk;
```

なお、ザイリンクス FPGA の I/O レジスタ付きロジックではセットアップ/ホールド タイムはほぼ一定であり、同一パッケージ内で大きく変わることはありません。それでも、デザインがシステム パラメータを確実に満たすようにするために、主に検証段階として制約を適用します。

## 3 つの簡単な手順

これまで見てきたように、信頼性の高いコードは次の 3 つの手順に従うだけで簡単に実装できます。

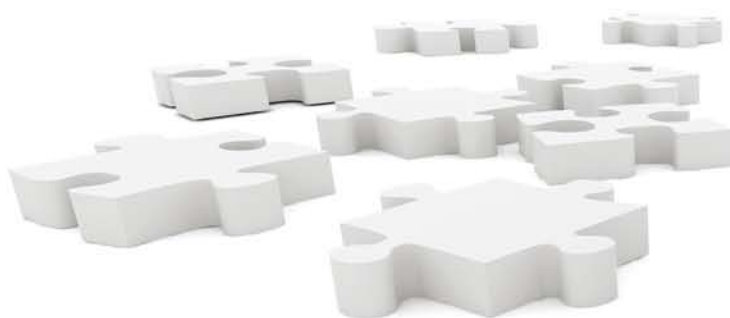
- 設計目的を合成ツールに推測させない。ザイリンクスのプリミティブを使用して I/O ピンとクリティカルなロジックのすべてを明確に定義する。I/O ピンの電気的特性も必ず定義する。
- ロジックは 100% 同期ロジックとし、すべてのロジックがマスタ クロック ドメインを基準に動作するようにする。
- タイミング制約を指定してタイミング クロージャを達成する。

これら 3 つの手順に従うだけで、合成とタイミングによるばらつきがなくなり、コードの動作に高い信頼性がもたらされます。🌈

## Easy Route to Simple MicroBlaze Microcontroller

# シンプル MicroBlaze マイクロ コントローラ (SMM) のコンセプト

FPGA デザインに容易に追加できる、オプション設定済みの  
コントローラで、特別なツールや複雑なスクリプトも不要

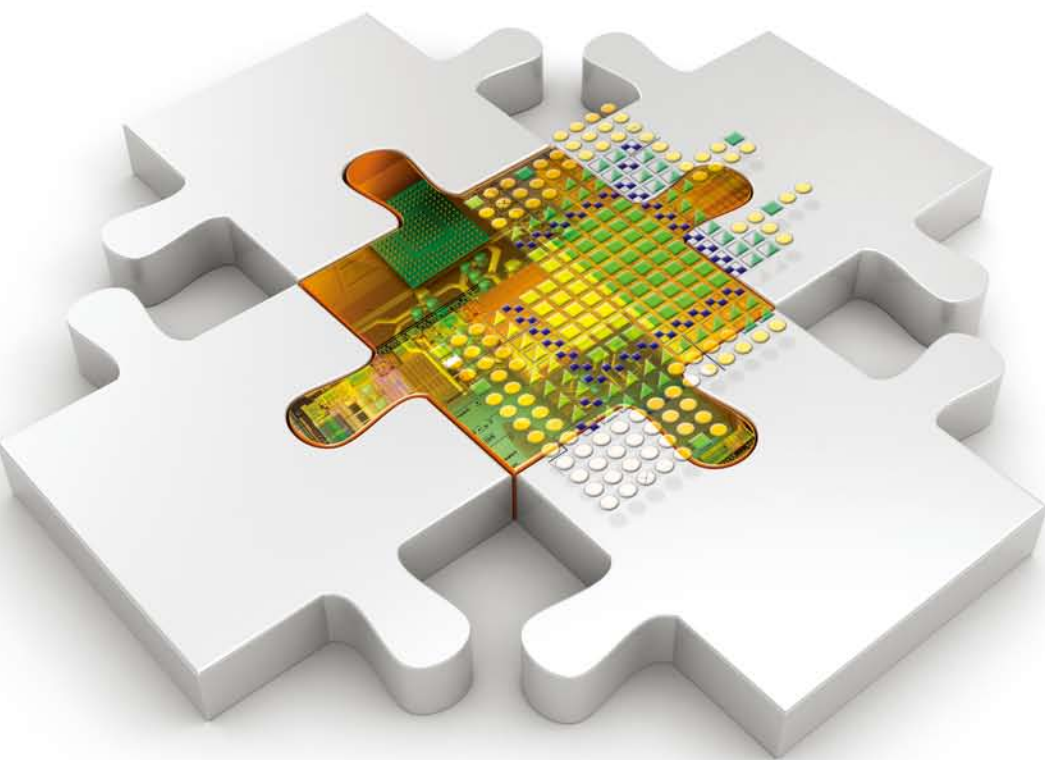


Christophe Charpentier  
Processor Specialist FAE  
Xilinx, Inc.  
[christophe.charpentier@xilinx.com](mailto:christophe.charpentier@xilinx.com)

エンベデッド マイクロコントローラは、単純なものから複雑なものまで幅広いアプリケーションで使用されています。ザイリックスは 2000 年以來、ハード (PowerPC® 405、PowerPC 440) およびファブリックベースの (MicroBlaze™) エンベデッド マイクロプロセッサを提供してきました。MicroBlaze は、シンプルな汎用アプリケーションだけでなくオペレーティング システムも実行できるなど、複雑なアプリケーションに対応できる点が大きな長所です。

MicroBlaze ソフト プロセッサはザイリックスの現行アーキテクチャすべてにインプリメント可能なため、ファミリ間の移行も容易で、抜群の柔軟性があります。ただし、MicroBlaze は 70 以上のパラメータが設定可能で、高度なエンベデッド ツールを使用するため、シンプルなマイクロコントローラで十分なアプリケーションの場合はシステム設計が面倒となることもあります。

しかし、ここで紹介する方法を活用すれば、オプション設定済みの SMM を構築し、どのような FPGA デザインにも簡単に追加できます。このコントローラは HDL に直接インスタンスシートされ、特別なスクリプトや複雑なフローを使用する必要もな





く標準の FPGA デザイン フローですぐに使用できます。これは 2 つのハードウェア インプリメンテーション ファイルと 1 つのソフトウェア定義ファイル、この 3 つのファイルが揃えば使用でき、この方法であれば、エンジニアは新しい知識を必要とせず FPGA エンベデッド デザインの開発が可能です。

ISE® のバージョン 11.1 以降のツールには、MicroBlaze のソフトウェア開発向けにスタンドアロンのソフトウェア開発キット (SDK) が付属され、フル機能のエンベデッド開発キット (EDK) がなくても C および C++ アプリケーションの作成とデバッグが行えるようになりました。

MicroBlaze には UART とデバッグの 2 つのオプションがあり、これらのオプションの有無によって異なる 4 つの構成で提供されています。表 1 に、各 FPGA ファミリーにおけるマイクロコントローラのサイズをオプション別に示します。これ以外に、Virtex® デバイスの場合は 2 つのブロック RAM、Spartan® デバイスの場合は 4 つのブロック RAM を使用します。アプリケーション コードのデバッグが完了してデバッグ オプションなしの構成に切り替えると、コントローラのサイズを抑えることができます。たとえば、Spartan-6 マイクロコントローラの場合、最小構成であれば必要なスライス数は 220 で済みます。

## マイクロコントローラの概要

SMM は、32 ビットの MicroBlaze プロセッサ、8KB の RAM/ROM、32 ビットのユーザー インターフェイスによる 64KB アドレス空間、割り込みサポート、UART (オプション)、JTAG デバッグ インターフェイス (オプション) で構成されています。図 1 に、システムのブロック図を示します。

クロック入力 (CLK) の周波数には下限がなく、上限はインプリメンテーションツールの対応範囲となります。アクティ

表 1 - シンプル MicroBlaze マイクロコントローラ (SMM) のサイズ : FPGA ファミリー別

Configuration	Spartan-3 Family			Virtex-5 Family			Spartan-6 Family			Virtex-6 Family		
	LUTs	FFs	Slices	LUTs	FFs	Slices	LUTs	FFs	Slices	LUTs	FFs	Slices
SMM + UART + Debug	1770	1240	1120	1020	1230	460	1350	1200	470	1350	1200	520
SMM + UART	1510	900	1090	830	880	360	1060	860	320	1050	870	350
SMM + Debug	1390	790	820	820	780	330	990	750	410	990	750	430
SMM	1130	450	630	630	440	220	690	440	220	690	440	250

ブ High のリセット入力 (RESET) は内部で入力クロックと同期しています。割り込みサポートは割り込み入力信号 (INTERUPT) によって実現しており、マイクロコントローラが割り込み処理を実行すると INTERRUPT\_ACK がそれを認識して応答します。シンプルなアドレス マッピングを利用したユーザー インターフェイスもクロックに同期しており、ユーザーによるカスタマイズが可能です。図 2 に、ユーザー インターフェイスのタイミング図を示します。バイトおよびハーフワードのトランザクションには、バイト イネーブル (BE) を使用します。

ソフトウェアでマップした 16 ビット幅のアドレス バスをデコードし、さまざまなカスタム インターフェイスやペリフェラルをマイクロコントローラに接続できます。読み出しデータは、チップ セレクト (CS) をアサートしてから 2 クロック サイクル後にサンプリングされます。

このマイクロコントローラには、オプションでシリアル 16450 UART を有効にした構成もありま

す。ボー レートはソフトウェアでプログラムされるため UART をクロック入力から独立して動作させることができます。デバッグ オプションは FPGA の内部リソースを使用し、FPGA の JTAG インター

フェイスに直接接続します。これにより、アプリケーションは、通常の FPGA ダウンロード ケーブルを用いてデバッグ可能となります。

## FPGA デザイン フロー

FPGA デザイン フローは、図 3 に示すように ISE を使用した一般的な FPGA インプリメンテーション フローと同様です。マイクロコントローラは、FPGA デザインの任意の階層に Verilog または VHDL でインスタンス化できます。ハードウェアに関連するファイルはマイクロコントローラのネットリスト (smm.ngc) とブロック RAM メモリのマップ ファイル (smm.bmm) の 2 つで、インプリメンテーションはこれらのファイルのみで完了します。

新しいツールを習得したり、複雑なスクリプトを用いたフローを使用する必要もなく、非常に容易に FPGA エンベデッド デザインを開発できます。マイクロコントローラのオプション構成を変更する場合

図 1 - MicroBlaze プロセッサ、メモリ、各種インターフェイスで構成される SMM

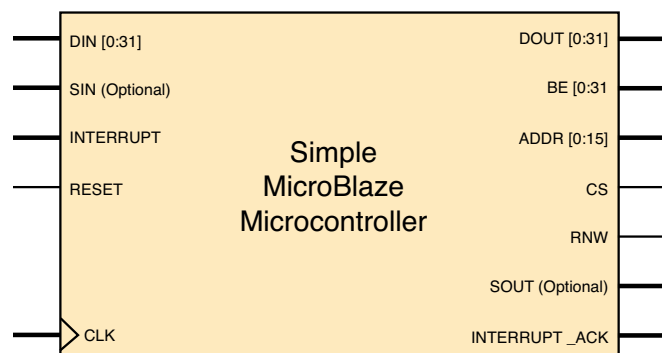
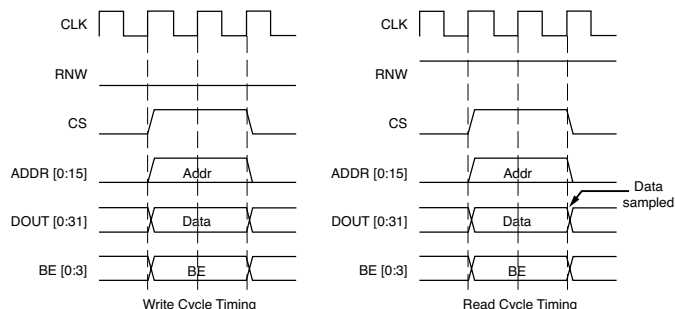


図 2 - シンプルなアドレス マッピングを利用した、クロック同期式のユーザー インターフェイス



も、ネットリスト ファイルを置き替えて FPGA を再インプリメントするのみです。インプリメンテーション ツールを実行すると、マイクロコントローラが使用するブロック RAM の物理的位置を示したファイル (smm\_bd.bmm) が作成されます。

## ソフトウェア アプリケーションのデザイン フロー

マイクロコントローラ アプリケーションの開発を始めるために必要な情報すべてが 1 つのソフトウェア記述ファイル (smm.xml) に含まれています。アプリケーションの開発は、FPGA デザイン フローと切り離して開始でき、FPGA デザインのインプリメンテーション前でも可能です。

ISE 11.1 以降では、SDK がスタンドアロンのオプションとして提供されています。この SDK には、ソフトウェア アプリケーションの開発に必要なツール、ドライバ、ライブラリ、ユーティリティすべてが含まれています。

図 4 に、ソフトウェア定義ファイルから始まる一般的な SDK の開発フローを示します。マイクロコントローラのアドレス空間には、8KB の RAM、ユーザー インターフェイス、そして UART レジスタ空間 (UART オプションを選択した場合) が含まれます。

ソフトウェア アプリケーションは、C または C++ で作成したものをマイクロコントローラのブロック RAM に格納できます。RAM 空間には、FPGA ビットストリーム内のアプリケーションをプリロードできる

ため、マイクロコントローラの ROM としての役割も果たします。FPGA のコンフィギュレーションが完了後、マイクロコントローラのリセットをディassertすると、アプリケーションが内部 RAM/ROM 空間から起動します。このマイクロコントローラは完全に自己完結型です。

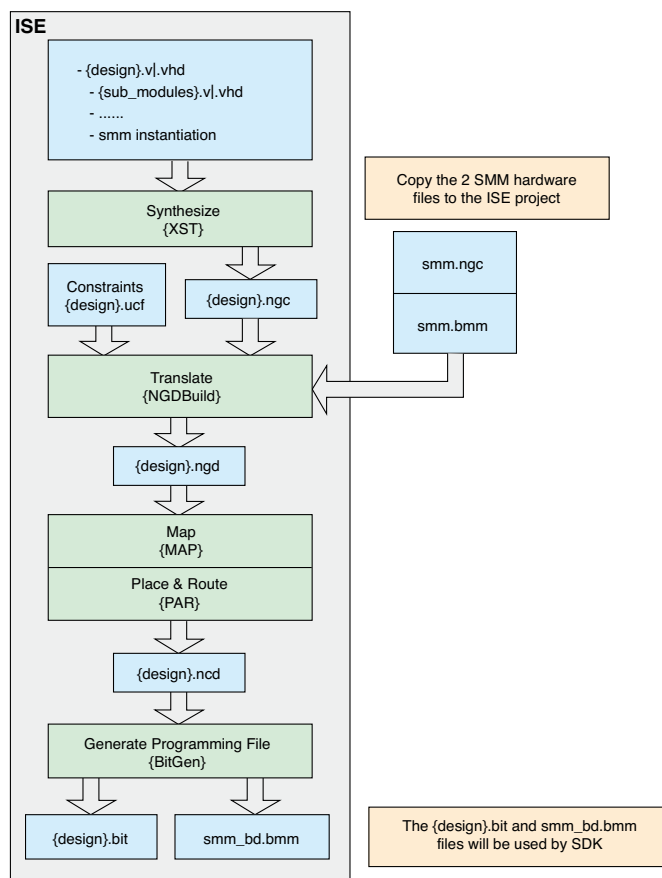
### デバッグ オプション

デバッグ オプションを使用すると、アプリケーションをソース レベルで完全にデバッグでき、メモリ、レジスタ、変数の内容が確認できます。デバッグを容易にするためにアプリケーション内にブレークポイントを設定することも可能です。デバッグには、FPGA のコンフィギュレーションと同じケーブルが使用できます。また、デバッグの完了後、デバッグ オプションのない構成に切り替えてコントローラのサイズを抑えることができます。

## デザイン例

ここで、ザイリンクスの 2 種類の開発ボードに対応した LCD コントローラのリファレンス デザインを紹介しましょう。このリファレンス デザインには、SMM のさまざまな機能が活用されています。LCD

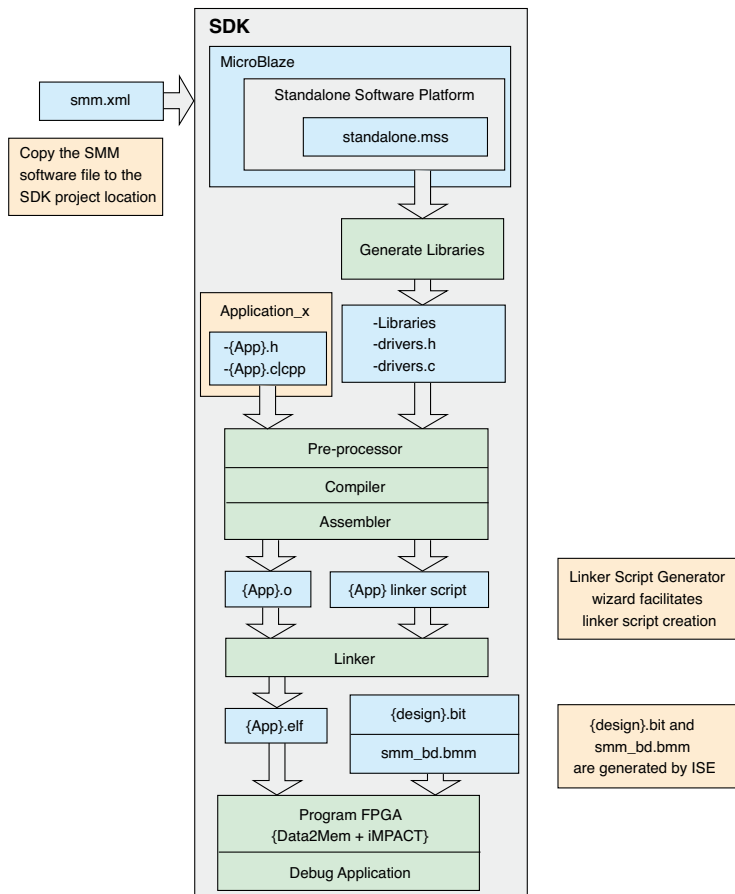
図 3 - FPGA デザイン フローは標準的な ISE FPGA インプリメンテーションフローと同様 (新しいツールやスクリプトは不要)



コントローラは、低速で簡単なハードウェア インターフェイス、長い初期化シーケンス、多数のキャラクタ コードなどの特徴から、小さなマイクロコントローラのインプリメンテーションに適しています。このデザインは、HDL と C コードの組み合わせによって、ボード上の LCD スクリーンにメッセージを出力します。ハードウェア インターフェイスは HDL で処理し、LCD スクリーンの初期化と制御はソフトウェアで行います。

LCD モジュールのタイミングは低速ですが、それと同時に命令またはデータ間に長い遅延が必要となります。たとえばディスプレイをクリアする命令の場合、次の命令またはデータを発行するまでに 1.52 ミリ秒の遅延が必要です。その他の命令では、40  $\mu$ s の遅延が必要なものや 1  $\mu$ s の遅延が必要なものなどがあります。

図 4 - ソフトウェア定義ファイルから始まる SDK 開発フロー



この遅延は C コードの while ループを用いて対処できますが、この方法は精度が非常に低く、コンパイラ最適化の影響を大きく受けます。そこで、FPGA にソフトウェアロード可能な 32 ビット カウンタを用意し、あらかじめ設定した遅延に達するとコントローラに割り込みをトリガするようにします。MicroBlaze がアドレス 0x10 に書き込みを実行すると、ユーザーインターフェイスのデータバスにあるデータに基づいてタイマが開始します。その後、MicroBlaze は割り込みを待機し、割り込みがあると実行を再開します。

MicroBlaze がユーザーインターフェイスのアドレス 0x0 に書き込みを実行すると、LCD コントローラハードウェアインターフェイスがトリガされ、このときのタイミングは HDL で処理されます。ユーザーインターフェイスデータバスは、命令またはデータ値を取り込みます。プッ

シュボタン入力は、ユーザーインターフェイスのアドレス 0x20 に接続されます。

この FPGA デザインは、トップレベルモジュール、LCD ハードウェア タイミングモジュール、ソフトウェアでアドレス指定可能なプログラマブル タイマで構成されています。トップレベル ファイルには、66MHz で動作するインスタンス化された SMM が含まれます。

C アプリケーションは 1 つのファイルに含まれています。このコードは、MicroBlaze の割り込みイネーブル、LCD スクリーンの初期化、異なる遅延の制御、LCD への出力 (2 行のキャラクタ)、プッシュボタンからの入力待ち、スクリーンのクリア、新しいメッセージの出力という連の処理を実行します。

Virtex-6 ML605 ボードの場合、このデザインは FPGA リソースの使用率が 1% 未満と少なく、デザインに効率よく

パーティションを設定できます。HDL のみでインプリメントとデバッグを行うよりも、C コードをコンパイルし、デバッグする方が大幅に実行時間が短縮されます。

## マイクロコントローラのカスタマイズ

このマイクロコントローラは MicroBlaze がベースとなっているため、設計者はさまざまな標準ペリフェラルやオプションを利用してエンベデッドシステムをカスタマイズできます。たとえば、異なる FPGA アーキテクチャに対応させたり、メインメモリを増やしたり、浮動小数点ユニットや標準の SPI、あるいは I2C ペリフェラルの追加も可能です。

マイクロコントローラシステムのカスタマイズには、EDK が必要です。EDK には多数の異なる構成がエンベデッドプロジェクトとして含まれており、これらはユーザーの要件に合わせて修正できます。たとえばメモリ容量を標準の 8KB から 16KB にしたい場合、EDK プロジェクトを開いて MicroBlaze の RAM 空間を修正し、新しいネットリスト、ブロック RAM メモリのマップ ファイル、ソフトウェア記述ファイルを作成します。そして、これらの新しいファイルを ISE と SDK のプロジェクトに追加します。

SMM がすべてのエンベデッドデザインに対応できるわけではありませんが、同一のマイクロコントローラを活用して効率よく制御機能を実現したいというニーズに応えることは可能です。また、EDK デザインを共有し、配布したいと考えているチームへコンセプトを提供することにもなります。このマイクロコントローラは、エンベデッドデザインのサイズにかかわらず、3 つのファイルを使用するだけでインプリメンテーションがすべて完了します。

このコントローラのコンセプトについての詳細は、以下のリンク先でザイリンクスのアプリケーション ノート XAPP1141 を参照してください。 [http://www.japan.com/support/documentation/application\\_notes/xapp1141.pdf](http://www.japan.com/support/documentation/application_notes/xapp1141.pdf)



Supercharge MicroBlaze Processing with Hardware Acceleration

# ハードウェア アクセラレーションによる MicroBlaze 処理の高速化

MicroBlaze コア周辺のデザインを最適化することによって、標準的なコントローラや DSP を上回る性能の強力なシステムを実現

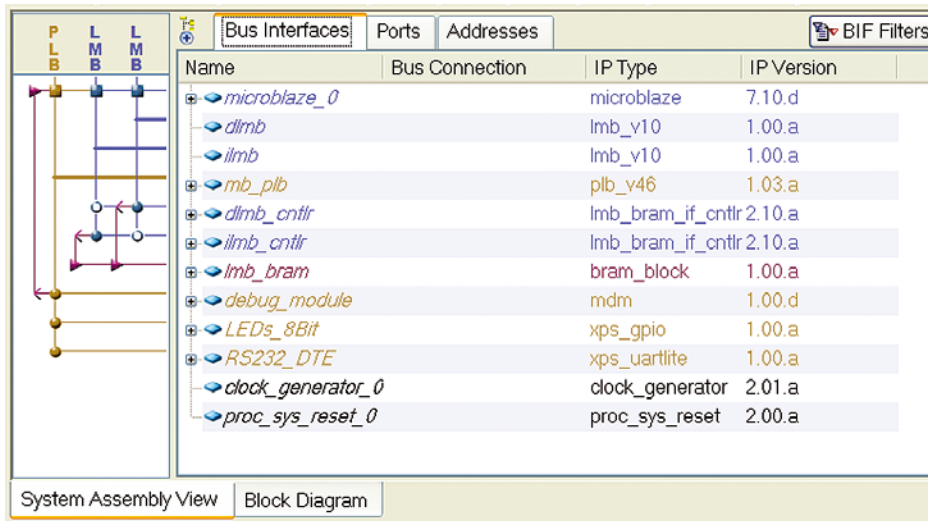


Karsten Trott  
Field Application Engineer  
Xilinx GmbH (Munich, Germany)  
[karsten.trott@xilinx.com](mailto:karsten.trott@xilinx.com)

アルゴリズムには、純粋なハードウェアに変換してプロセッサを高速化できるものが数多くあります。代表的なものを挙げると、標準偏差アルゴリズムは一定の時間内で最大値と最小値を求めるアルゴリズムで、フィルタや FFT は簡単にハードウェアに変換できるアルゴリズムです。さらに、ビット反転のように特殊なアルゴリズムも、適切なハードウェア アクセラレータを使用してハードウェアとして実装できます。

ザイリンクスの組み込みシステムは非常に柔軟であるため、ハードウェア アクセラレータの FPGA ベースのソリューションへの組み込みが容易です。FPGA の強力なコンピューティング性能を活かせば、標準的なプロセッサ、コントローラ、および DSP の性能を上回るシステムが容易に構築できるのです。

図 1 - 組み込みメモリのみを使用した一般的な MicroBlaze デザイン



ザイリンクスがエンベット開発キット (EDK) で提供している 2 種類の 32 ビット コアの 1 つ、MicroBlaze™ プロセッサはハードウェア アクセラレーションのインプリメントに適したプロセッサです。図 1 に典型的な MicroBlaze デザインを示します。コア自体に 32 ビット乗算器が含まれますが、浮動小数点演算装置 (FPU) やパラレル シフタ、専用ハードウェア アクセラレータは含まれません。Spartan® FPGA デバイスの場合、デフォルトのシステムには実装面積を最適化したバージョンの MicroBlaze (3 段パイプライン) が含まれていますが、性能評価を開始する際は、動作速度を最適化したバージョンの MicroBlaze (5 段パイプライン) が一般的に使用されます。これは小規模でシンプルなプロセッサですが、拡張も容易です。

本稿では、このプロセッサ デザインに対して実際にあった要望の例を 2 つ挙げ、ハードウェア アクセラレーションによる MicroBlaze の絶大な効果について検証していきます。ここでは、特に Spartan デバイスを使用して FPGA ソリューションが標準のコントローラ コアを上回るコストパフォーマンスを発揮することを実証していきます。ただし、同じ手法は Virtex® FPGA にも通用します。

## ビット反転アルゴリズムのインプリメンテーション

最初のサンプル アプリケーションでは、MicroBlaze プロセッサの動作速度を 50MHz とします。この動作速度は、デバイスが Spartan-3 や Spartan-6 であれば容易に得られます。ローカル メモリバス (命令およびデータ LMB) やプロセッサローカル バス (PLB) などの内部バスも同様に 50MHz で動作します。説明を簡単にするために、外部 DDR メモリは接続されていないものとします。

この CPU を使用してビット反転アルゴリズムをインプリメントしたいと仮定します。MicroBlaze にはこの機能を直接実行するハードウェアは含まれていません。また、このビット反転処理は毎秒 20,000 回実行する必要があるものとします。

この問題を解決するために、ユーザーの多くはまず純粋なソフトウェア インプリメンテーションを検討するでしょう。それは、この方法が最も容易に目的の機能を得ることができると考えられるためです。しかも、十分な性能のものであれば、変更を加えることなく使用できるという利点もあります。

そこで、まず簡単なソフトウェア アルゴリズムとしてインプリメントした、小規模

でシンプルなソリューションを用意しました。これは短いコードにまとまっており、理解しやすいのですが、効率はよくありません。

```
unsigned int v=value;
unsigned int r = v;
int s = sizeof(v) * CHAR_BIT - 1;
for (v >>= 1; v; v >>= 1)
{
    r <<= 1;
    r |= v & 1;
    s--;
}
r <<= s;
return r;
```

このインプリメンテーションは動作自体に問題ありませんが、動作速度を最適化した MicroBlaze (5 段パイプライン) 上で 32 ビットのシングル ワードに対してアルゴリズムを実行したところ、220 サイクルもかかりました。動作速度 50MHz の MicroBlaze 上で 20,000 回の演算を実行するのに要した時間は約 88 ミリ秒です。

次に、やや異なるアプローチでこのアルゴリズムの最適化を試みましたが、これもまだ純粋なソフトウェア ソリューションとしてインプリメントされています。

```
unsigned int v=value;
unsigned int r=0;
if (v & 0x00000001) r |= 0x80000000;
if (v & 0x00000002) r |= 0x40000000;
if (v & 0x00000004) r |= 0x20000000;
if (v & 0x00000008) r |= 0x10000000;

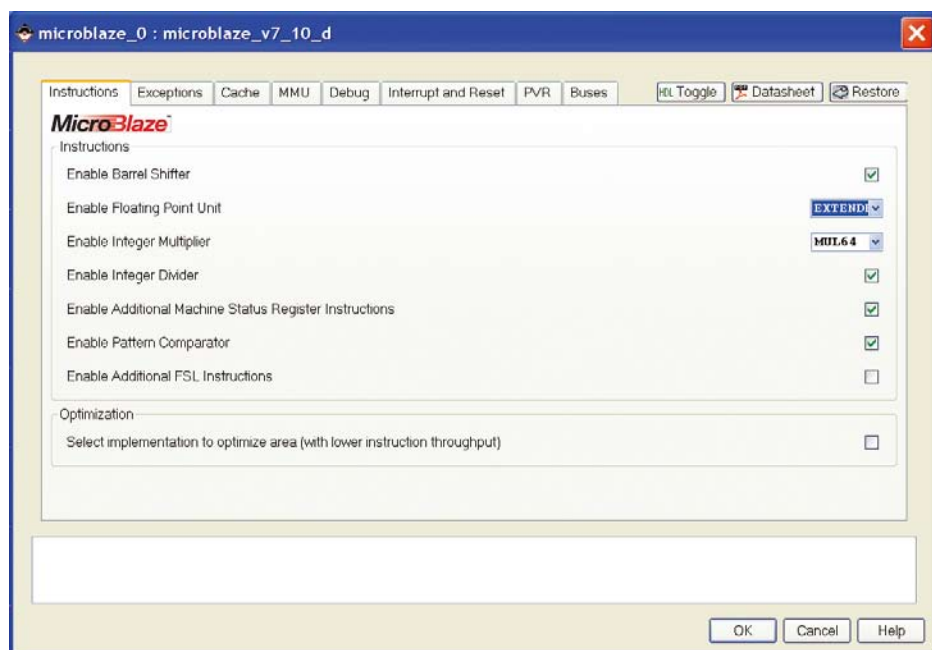
if (v & 0x00000010) r |= 0x08000000;
if (v & 0x00000020) r |= 0x04000000;
if (v & 0x00000040) r |= 0x02000000;
if (v & 0x00000080) r |= 0x01000000;

if (v & 0x00000100) r |= 0x00800000;
if (v & 0x00000200) r |= 0x00400000;
if (v & 0x00000400) r |= 0x00200000;
if (v & 0x00000800) r |= 0x00100000;

if (v & 0x00001000) r |= 0x00080000;
```



図 2 - MicroBlaze のプロパティでバレル シフタを有効に設定



```
if (v & 0x00002000) r |= 0x00040000;
if (v & 0x00004000) r |= 0x00020000;
if (v & 0x00008000) r |= 0x00010000;
```

```
if (v & 0x00010000) r |= 0x00008000;
if (v & 0x00020000) r |= 0x00004000;
if (v & 0x00040000) r |= 0x00002000;
if (v & 0x00080000) r |= 0x00001000;
```

```
if (v & 0x00100000) r |= 0x00000800;
if (v & 0x00200000) r |= 0x00000400;
if (v & 0x00400000) r |= 0x00000200;
if (v & 0x00800000) r |= 0x00000100;
```

```
if (v & 0x01000000) r |= 0x00000080;
if (v & 0x02000000) r |= 0x00000040;
if (v & 0x04000000) r |= 0x00000020;
if (v & 0x08000000) r |= 0x00000010;
```

```
if (v & 0x10000000) r |= 0x00000008;
if (v & 0x20000000) r |= 0x00000004;
if (v & 0x40000000) r |= 0x00000002;
if (v & 0x80000000) r |= 0x00000001;
return r;
```

この方法では、コードはずいぶん長くなっていますが、効率は改善されています。

標準の MicroBlaze で 32 ビットのシングルワードに対してこのコードを実行すると、130 サイクルかかりました。先ほどに比べると大きな改善ですが、まだ時間がかかりすぎています。20,000 回の演算に要する時間は約 52 ミリ秒です。

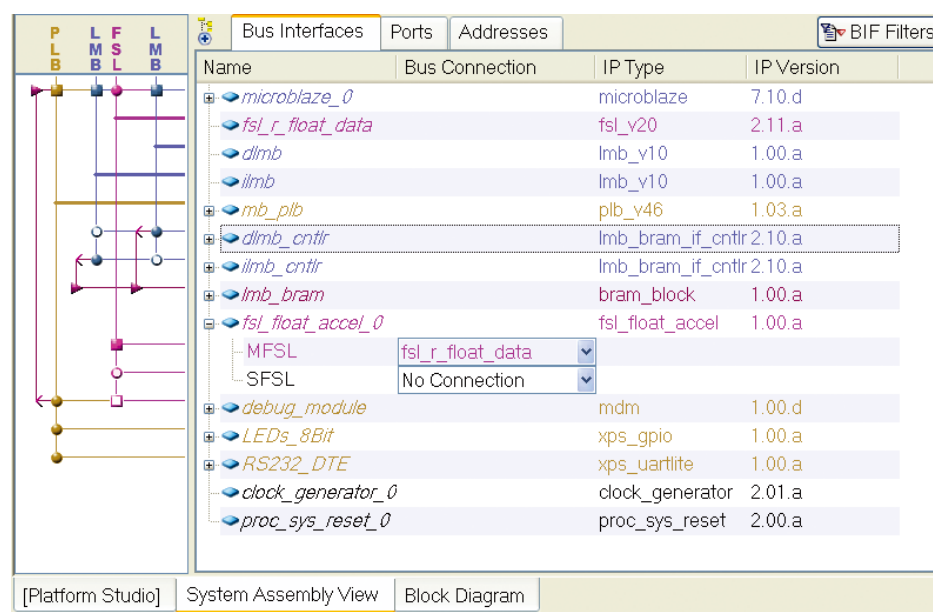
さらに、ユーザーはインターネットで検索してよりよいアルゴリズムを見つけ、次のようなコードを作成しました。

```
unsigned x = value;
unsigned r;
x = (((x & 0xaaaaaaaa) >> 1) | ((x & 0x55555555) << 1));
x = (((x & 0xcccccccc) >> 2) | ((x & 0x33333333) << 2));
x = (((x & 0xf0f0f0f0) >> 4) | ((x & 0x0f0f0f0f) << 4));
x = (((x & 0xff00ff00) >> 8) | ((x & 0x00ff00ff) << 8));
r = ((x >> 16) | (x << 16));
return r;
```

このコードは見た目にも効率がよく、サイズもコンパクトです。しかも分岐を使用していないため、パイプライン実行が乱れることもありません。このコードをコアシステムで実行すると、29 サイクルで処理が完了しました。

しかし、このアルゴリズムには 1、2、4、8、16 ビットのシフト演算が使用されています。そこで、MicroBlaze のプロパティウィンドウでバレル シフタを有効に設定します (図 2 参照)。バレル シフタを利用す

図 3 - FSL バスを使用して FSL ハードウェアを接続





ると、シフト演算の長さにかかわらずシフト命令を 1 サイクルで実行できます。この結果、MicroBlaze 上での純粋なソフトウェア アルゴリズムの実行速度がやや向上します。

MicroBlaze ハードウェアでバレル シフタを有効にすると、アルゴリズムの処理にかかるサイクル数は 22 に短縮されます。これは、最初に例としてあげたソフトウェア アルゴリズムに比べると大きな改善です。20,000 回実行しても約 8.8 ミリ秒しかかかっていません。これは最初のアルゴリズムから 10 倍の高速化ですが、それでもユーザーの要望を満たすには不十分です。

以上のことを踏まえると、純粋なソフトウェア ソリューションではどれだけ MicroBlaze コアのパラメータを調整してもこれ以上の高速化を望めないことがわかります。そこで、ハードウェア アクセラレータを最大限に活用した新しいアプローチによる純粋なハードウェア ソリューションの出番となります。

ハードウェア ソリューションでは、基本的な演算は MicroBlaze の Fast Simplex Link (FSL) にきわめてシンプルなコアを接続するだけで高速化できます。標準の FSL インプリメンテーションは、同期または非同期 FIFO を備えた FSL バスを使用し、MicroBlaze コアから FSL ハードウェア アクセラレータ IP (Intellectual Property) ヘデータを転送します。FIFO を備えた FSL バスにより、両側からのアクセスが分離されます。

FIFO を含む FSL バスを使用した標準インプリメンテーションの場合、実行時間は 4 サイクルで、MicroBlaze から FSL バスへのデータを FIFO に書き込み、FIFO から FSL IP へのデータ転送、FSL IP から FSL バスの FIFO への結果転送、FSL バスから MicroBlaze への結果の読み出しにそれぞれ 1 サイクルかかります。

図 3 に示したように、MicroBlaze から FSL バスへの接続と FSL バスから FSL IP への接続は、EDK のグラフィカルビューで容易に作成できます。ただし、これにはまだ改善の余地があります。アルゴリズムのレイテンシは実行時間を大きく左

右するため、なるべく小さくする必要がありますが、2 つの FSL バスを使用したインプリメンテーションでは、4 クロック サイクルが必要となります。そこで、FSL バスによる接続ではなく MicroBlaze とハードウェア アクセラレータを直接リンクすると、レイテンシを半分の 2 クロック サイクルまで低減させることができます。この構成の場合、データを FSL ハードウェア アクセラレータ IP に書き込むのに 1 サイクル、そして結果を読み出すのに 1 サイクルしかかかりません。

直接リンクを使用する際にはいくつか注意すべき事項があります。まず、コプロセッサ IP に入力を格納し、結果をレジスタード方式で出力する必要があります。FSL バスを使用する場合は、FIFO がこの役割を果たしていました。

また、直接リンクの場合、MicroBlaze と FSL IP のクロック速度が異なると扱いが困難になります。競合を避けるために、これら 2 つを同じ速度で動作させた方がよいでしょう。

では、FSL バスを介在させずに MicroBlaze と FSL IP を直接接続するにはどうすればよいでしょうか。答えは非常に簡単です。まず MicroBlaze のデータラインとハードウェア アクセラレータを接続し、次に必要に応じてハンドシェイク信号を接続します。

このビット反転 IP の例では、書き込み信号のみが必要です。この IP は非常に高速であるため MicroBlaze からの要求には十分に応答できます。

IP 自体は非常にシンプルです。次に、VHDL コードの抜粋を示します。

```
architecture behavioral of
    fsl_bitrev is
        -- data value sent by micro-
        blaze:
            signal data_value :
                std_logic_vector(0 to 31) := (others=>'0');

begin
    -- bitreversed value to write
```

```
back:
    FSL_M_Data <= data_value;
process(FSL_Clk)
begin
    if rising_edge(FSL_CLK) then
        if (FSL_S_Exists = '1') then
            -- create the bitreversed
data:
            data_value(0) <= FSL_S_Data(31);
            data_value(1) <= FSL_S_Data(30);
            data_value(2) <= FSL_S_Data(29);
            ...
            data_value(30) <= FSL_S_Data(1);
            data_value(31) <= FSL_S_Data(0);
        end if;
    end if;
end process;
end architecture behavioral;
```

この IP を FSL バスを介在させずに接続するには、プロジェクトの MHS ファイルを次のように変更する必要があります。

```
BEGIN microblaze
...
PARAMETER C_FSL_LINKS = 1
...
PORT FSL0_S_EXISTS = net_vcc
PORT FSL0_S_DATA = FSL0_S_DATA
PORT FSL0_M_DATA = FSL0_M_DATA
PORT FSL0_M_WRITE = FSL0_M_EXISTS
PORT FSL0_M_Full = net_gnd
END

BEGIN fsl_bitrev
PARAMETER INSTANCE = fsl_bitrev_0
PARAMETER HW_VER = 1.00.a
PORT FSL_S_DATA = FSL0_M_DATA
PORT FSL_S_EXISTS = FSL0_M_EXISTS
PORT FSL_M_Data = FSL0_S_DATA
PORT FSL_M_Full = net_gnd
PORT FSL_Clk = clk_50_0000MHz
END
```

これにより、処理速度は大幅に向上します。このハードウェア コアは、1 サイクルでデータを IP に書き込み、1 サイクルで読み出すことができるため、わずか 2 サイクルでビット反転演算を実行できま

す。ビット反転演算を 20,000 回実行しても 0.8 ミリ秒しかかかりません。

最初のアルゴリズムはこの 110 倍の時間がかかっていたことを考えると、これは非常に大きな高速化といえます。言い換えると、最初のアルゴリズムでこのアプリケーションを同じ時間で実行するには、動作速度約 5.5GHz のプロセッサが必要になります。いずれにせよ、最初に例として挙げたアルゴリズムとの差は歴然です。

最後に例として挙げた、改良を加えたソフトウェア アルゴリズムと比較しても、このシステムの方が 11 倍高速です。つまり、このアルゴリズムを純粋なソフトウェアで実行した場合、動作速度 550MHz の CPU が必要になります（しかも CPU にバレル シフタがあることが前提です）。

ここに示した例は、CPU にビット反転アドレス指定機能がない場合に有効です。この機能は多くの DSP に搭載されていますが、マイクロコントローラで搭載しているものはほとんどありません。しかし、この機能を追加できれば、アルゴリズムの処理は飛躍的に高速化します。

これまで見たように、ごくわずかな変更でも大きな効果があり、最終的には、コード サイズをわずか 2 ワードにまで縮小できました。もちろん、このインプリメンテーションではハードウェアにいくつかのスライスを追加する必要があります。しかし、これによって標準的なマイクロコントローラをしのぐ性能が得られることを考えると、十分に意味のあるトレードオフであると言えます。

## 浮動小数点演算の高速化

もう 1 つの例として、MicroBlaze のアルゴリズム高速化の効果を見てみましょう。あるユーザーから、MicroBlaze システム上での浮動小数点演算速度が非常に遅いという報告がありました。このアルゴリズムは、シンプルなループを使用していくつかの結果を一度に生成します。

```
for (i=0;i<512;i++) {
    f_sum      += farr[i];
```

```
    f_sum_prod += farr[i] * farr[i];
    f_sum_tprod += farr[i] * farr[i] *
farr[i];
    f_sqrt      +      =
sqrt(farr[i]);
    if (min_f > farr[i]) { min_f =
farr[i]; }
```

```
    if (max_f < farr[i]) { max_f =
farr[i]; }
```

値はすべて単精度浮動小数点型です。まず頭に浮かんだのは、浮動小数点演算装置が有効になっているかどうかという最も基本的な疑問でした。プロジェクトの設定

図 4 - 拡張 FPU サポートの有効化

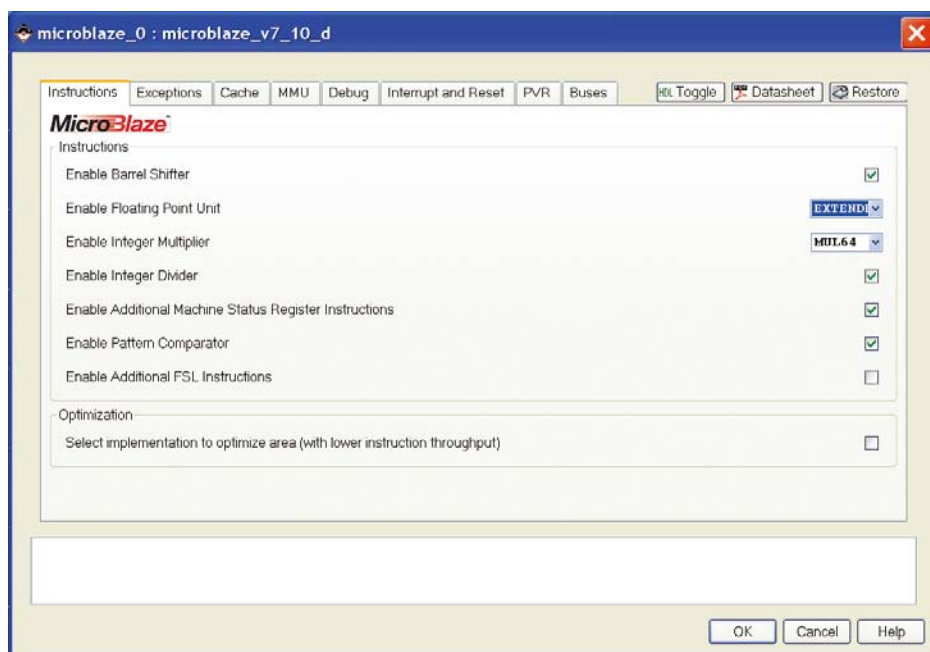
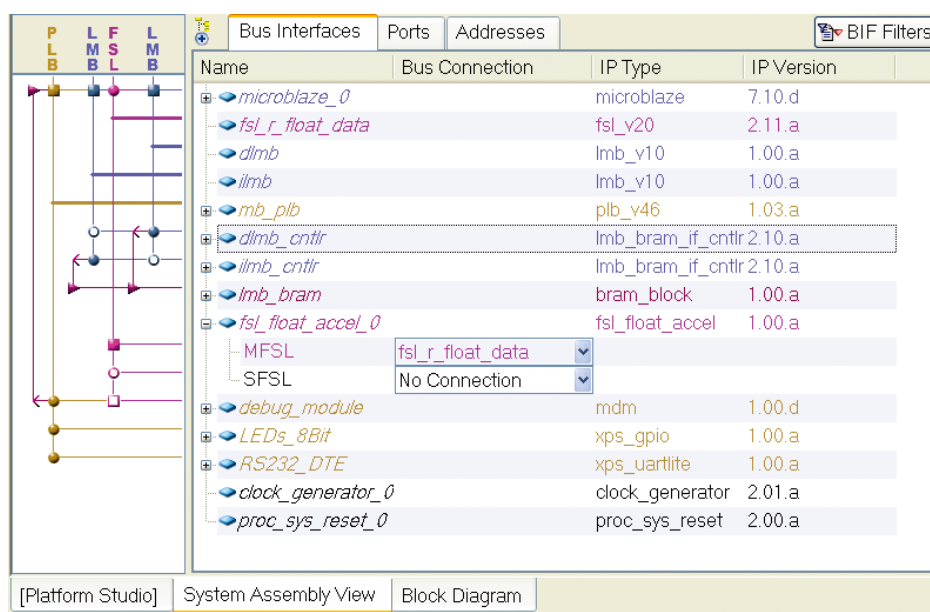


図 5 - 並列の浮動小数点演算アクセラレータを使用したシステム



を確認してみると、FPU は無効になったままでした。これでは、ごくわずかな数値の計算にも長時間かかります。FPU は、MicroBlaze のプロパティ設定で有効にできます (図 4 参照)。

FPU サポートは 2 種類あります。ここでは、`sqrt()` 演算にも対応できるように拡張 FPU を選択しました。この FPU を使用して、50MHz の MicroBlaze 上で 512 個の値に対してループを実行すると、1,108,685 サイクルかかります。生成されたアセンブラコードを見てみると、平方根の計算に算術ライブラリ関数を使用されていることが分かりました。この関数で次のように定義されています。

```
double sqrt(double);
```

しかし、`sqrt()` 関数は浮動小数点型の値の処理にしか使用されていませんでした。そこで、この関数に代わる新しい関数を MicroBlaze FPU によって次のように定義し、問題を回避することにしました。

```
float sqrtf(float);
```

`f_sqrt += sqrt(farr[i]);` の行を `f_sqrt += sqrtf(farr[i]);` に書き換えると、MicroBlaze に内蔵された FPU 内部平方根機能が使用されるようになります。これで、コードの実行時間は 35,336 サイクルにまで短縮されました。最初のインプリメンテーションは FPU をまったく使用していませんでしたが、わずかな変更を加えるだけで 31 倍もの高速化を果たしたことになります。最初のインプリメンテーションで同じ動作速度を達成するには、約 1.5GHz の CPU が必要となります。

しかしこのユーザーはこの結果では満足していなかったため、さらに別の方法で高速化を検討することにしました。この場合、アルゴリズムを浮動小数点演算から固定小数点演算へと変更することは得策ではありません。そこで、新しい専用のハードウェアアクセラレータ (FSL IP) を開発してループ処理速度の向上を図ることにしました。

この新しい FSL IP は CORE Generator™

モジュールの `floating-point_v4_0` を使用し、ADD (4)、MUL (2)、GREATER (1)、LESS (1)、SQRT (1) の演算に合計 9 つのインスタンスを作成します。これらはすべてインスタンス化され、同じ入力データに対して完全に並列に動作します (図 5 参照)。

FSL IP 内でインスタンスを作成する際はある程度の内部レイテンシが発生しますが、スループットは 1 です。このため、アクセラレータ内部のコントローラハードウェアに複数のスライスを追加する必要がありましたが、これにより、クロックサイクルごとに新しいデータをコプロセッサに供給できるようになりました。

処理ループの最後の部分のみ、結果を取得するまでに数サイクル余分に必要となります。

MicroBlaze と FSL IP を直接リンクで接続したため、FIFO は必要ありませんでした。転送されるデータはすべて IP 内部のバッファに格納され、その直後に処理されます。

FSL IP から MicroBlaze へ戻るリンクは、FSL バスを使用して作成しました。これは、複数の結果を MicroBlaze へ戻す必要があるためです。また、この処理は IP 内部で行った方がシンプルになるため、FSL バスを使用した方が簡単に実現できます。一部の CoreGen モジュールのレイテンシによって実行時間が長くなっていますが、これは関数 `getfsl()` の実行によって完全に補われます。また、MicroBlaze はすべての結果が FSL バスの FIFO に格納されるまで待機状態となり、データレートが 1 で必要なスループットが達成できていれば、問題にはなりません。

FSL バスによる遅延はそれほど大きなペナルティにはなりません (数サイクル程度)。この FSL IP を使用した C コードは次のようになります。

```
for (i=0;i<512;i++) {
    putfsl(farr[i],fsl0_id);
}

// get the min,max values:
getfsl(min_f,fsl0_id);
getfsl(max_f,fsl0_id);

// get the sum and products:
getfsl(f_sum,fsl0_id);
```

```
getfsl(f_sum_prod,fsl0_id);
getfsl(f_sum_tprod,fsl0_id);
getfsl(f_sqrt,fsl0_id);
```

最後に例としてあげたこのアルゴリズムのインプリメンテーションは、約 4,630 サイクルしかかかりませんでした。しかも、完全な浮動小数点のインプリメンテーションです。

ハードウェアアクセラレータのインプリメント用にわずかなスライスを追加しただけで、すべての結果をハードウェアで並列に計算できるようになりました。この結果、拡張 FPU をサポートしたインプリメンテーションに比べ、約 7.6 倍の高速化が実現しました。仮に 50MHz のプロセッサを標準のプロセッサで置き換えた場合、約 380MHz の CPU が必要になります (浮動小数点の `sqrt` 関数をハードウェアで実行できる CPU であることが前提です)。

また、`sqrt()` 関数を使用した最初の例との差はさらに顕著で、全体的に約 239 倍の高速化を実現しています。このインプリメンテーションで同じ動作速度を達成するには、約 12GHz の浮動小数点演算プロセッサが必要となります。

これらの例からも分かるように、わずかな変更を加えるだけでアルゴリズムの処理が飛躍的に高速化することがあります。適切にインプリメントすれば、50MHz の MicroBlaze システムでも高性能 DSP に匹敵する処理能力が得られます。

まず、実行に時間がかかりすぎているコアアルゴリズムを特定します。次に、ソフトウェアに簡単な変更を加えたり、ハードウェアを使用したり、あるいはハードウェアアクセラレータを使用し、より複雑な変更を加えて高速化を図ります。これらによって、プロセッサシステムの性能は必ず標準的なコントローラを上回るようになります。🌈

Karsten Trott はドイツ、ミュンヘン在住のザイリンクス FAE です。アナログチップデザインの博士号課程を修了しており、チップデザインと合成に深い造詣があります。

連絡先: Karsten.Trott@xilinx.com



# 無線信号処理 & 記録システム

## DTA-RFR16

ソフトウェア無線機、MIMO、コグニティブ無線等の評価・解析に最適 !!



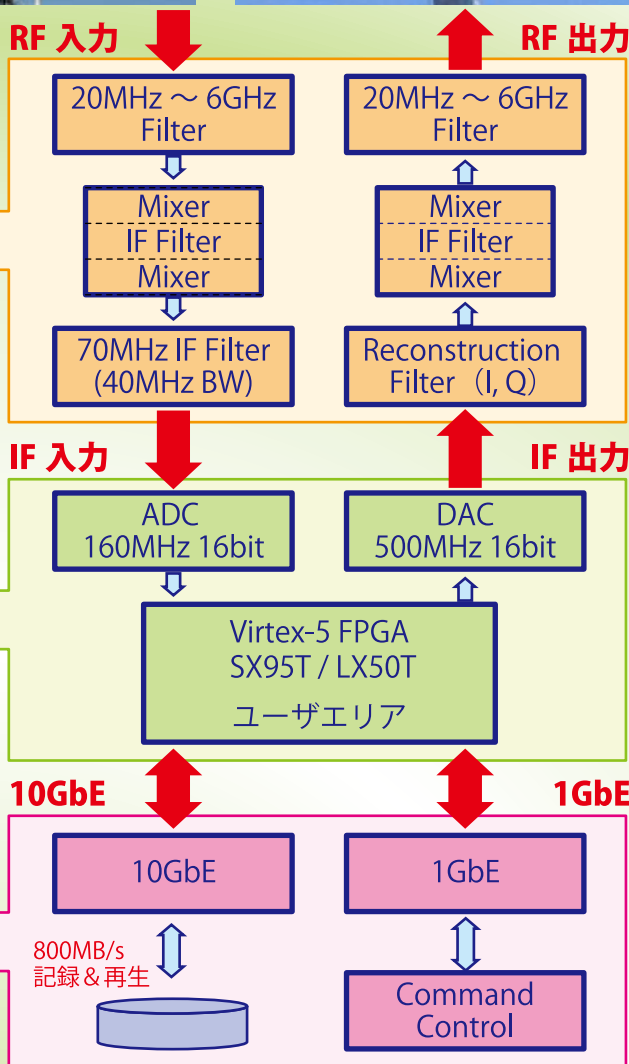
DTA-3200 : 16ch RF チューナ



DTA-2300 : 16ch IF 信号処理



DTA-5000L : 800MB/s データレコーダ



1. 19 インチラックマウントユニットでのコンプリート製品 !
2. RF チューナからデータレコーダまで、性能・機能別に豊富なラインナップ !
3. 各種スペックや Ch 数カスタマイズ可能 !

(お問い合わせは)  
sales@mish.co.jp  
<http://www.mish.co.jp/>



MISH  
INTERNATIONAL

株式会社ミツシュインターナショナル  
TEL 042-538-7650 FAX 042-534-1610  
〒190-0004 東京都立川市柏町4-56-1

# マイケレルのClockWorks™ファミリ

## 低消費電流で、超低Jitterを実現した Clock Synthesizer

**高速・大規模システムで問題となるClockのJitterと低消費電力を高いレベルで両立**

ますます高速化・大規模化するシステムにおいて、システムの安定動作のカギを握るのがクロック品質。Clock Synthesizerは、一般的に消費電流と低Jitterとはトレードオフの関係にあり、どちらかの特性を良くすると、もう一方の特性が犠牲になります。しかし、マイケレルのClockWorksファミリでは、Rotary Wave Technology<sup>注1</sup>というこれまでにない独自の技術を活用することで、この相反する要求を高いレベルで両立させました。

注1: Rotary Wave TechnologyはMultigig社の登録商標です



### RMS Phase Jitter ( Phase Noise ) 320fs以下

RMS Phase Jitterは、現行のClock Works製品群において、SM320001が320fs (typ)を、SM840021が250fs (typ)を実現。また、現在開発中の製品では、100fsを下回ります。

### 消費電流 77mA (typ)

消費電流は、各製品とも77mA (typ)と、従来の低Jitter Clock Synthesizerより30%以上低減。

### 特長

- 低RMS Phase Jitter
- 低消費電流
- 低PSRR
- 動作周波数 : ~850MHz
- 出力I/F: LVCMOS, (LVPECL, LVDS, HCSL 開発中)
- パッケージ: TSSOP, (MLF 開発予定)
- 動作温度: -40C~+85C

### Clock Works製品ファミリ

型番	XTAL 周波数 (MHz)	電源電圧 (V)	出力周波数 (MHz)	RMS Phase Jitter (typ)	消費電流 (typ)	Output I/F	パッケージ
SM840051	20.1416	2.5/3.3V	161.133 / 80.566	300fs at	77mA at 3.3V	LVCMOS	8-Pin TSSOP
	19.5313		156.25 / 78.125	156.25MHz			
SM840021	25	2.5/3.3V	125	250fs	77mA at 3.3V	LVCMOS	8-Pin TSSOP
SM840001	26.5625	2.5/3.3V	106.25 / 212.5	320fs at 212.5MHz	77mA at 3.3V	LVCMOS	8-Pin TSSOP

### 販売代理店

伯東株式会社  
〒160-8910

東京都新宿区新宿1-1-13  
TEL: 03-3355-7629

ユニ電子株式会社  
〒140-0001

東京都品川区北品川3-6-17  
TEL: 03-6811-7478

フューチャーエレクトロニクス株式会社  
〒220-8124

神奈川県横浜市西区みなとみらい2-2-1  
横浜ランドマークタワー24階  
TEL: 045-224-2155

株式会社リョーサン  
〒101-0031

東京都千代田区東神田2-3-5  
TEL: 03-3862-2635

詳しくは弊社ホームページをご覧ください。

<http://www.micrel.jp>

マイケレル・セミコンダクタ・ジャパン株式会社  
〒220-6014 神奈川県横浜市西区みなとみらい2-3-1  
クイーンズタワーA14階  
TEL: 045-224-6616 FAX: 045-224-6716

# Why Software Programmability of Electronics is a Game Changer

## 従来の常識を覆すエレクトロニクス製品のソフトウェア プログラマビリティ

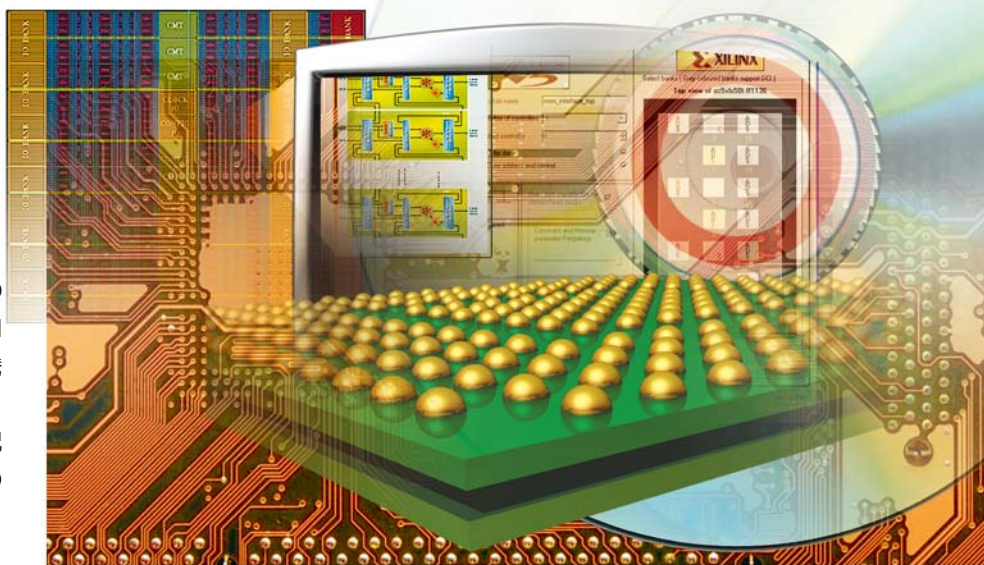
エンベデッド マイクロプロセッサを中心に、特定のタスク専用のハードウェア アクセラレータを複数配置した未来のアーキテクチャ



Jacques Benkoski  
USVP Venture Partner;  
Executive Chairman, Synfora  
[jacques.benkoski@synfora.com](mailto:jacques.benkoski@synfora.com)

半導体業界は現在、デザイン コストの上昇、製品化に必要な IP (Intellectual Property) の増加、マスク コストの高騰などの要因が重なり、ごく一部の量産チップ以外は採算がとれないという深刻な状況にあるというのが一般的な見方です。このため、今後はマイクロプロセッサとごくわずかな民生向け SoC のみが生き残り、FPGA の採用はコストや消費電力の要件がそれほど厳しくないアプリケーション向けの少量生産に限られると予想されています。

しかし実際には、IC のソフトウェア プログラマビリティの向上が求められ、こうした既成概念は根底から覆ることになります。最近では、ハードウェア拡張機能を自動で生成してソフトウェアの記述をマップするという機能を備えたツールがコンパイラの世界から登場しつつあります。これにより、エレクトロニクス製品はソフトウェアをプログラムしておけば自動的にコンパイルが行われるようになり、RTL で表現するアセンブリ言語に相当する過程は不要になります。



この結果、エンベデッド マイクロプロセッサを中心に据え、特定のタスクの処理に適したハードウェア アクセラレータを複数配置したアーキテクチャの IC が実現します。これは専用の ASSP や ASIC の話ですが、エンベデッド コアを搭載した最近の FPGA にも十分に当てはまります。

### マルチコアの問題点

同じタスクを実行するなら、汎用プロセッサ アーキテクチャ上でソフトウェアを実行するよりも専用ハードウェアを使用した方がコスト、消費電力、性能の面で数値

的に 2 桁は有利であることが知られています。現在では、モバイル アプリケーション以外でも低消費電力化ソリューションへの圧力が拡大しています。たとえば、エンベデッド コアを使用して 5Gbps のストリームを処理し、ビデオのリアルタイムトランスコードを行い、ビームシェーピングアンテナ アレイを利用しようとする、現実離れした動作速度が必要になります。

この問題は、特定のアプリケーションに専用のコアを割り当てるマルチコアによって解決するという意見もありますが、ホモジニアス アーキテクチャの単純なマルチスレッディングを別にすれば、マルチコア



プログラミングは非常に敷居が高いという問題点があります。特性の異なるヘテロジニアスな処理エンジンが、十分に特性評価されていないバスや NoC (Network-on-chip) を経由して通信するという状況はあまりにも複雑であり、容易に理解できるものではありません。しかも、少なくとも相対的にはソフトウェア エンジニア数が増え続け、ハードウェア エンジニア数は減り続けています。ところが企業の多くは、ソフトウェアを価値ある差別化要因という本来の姿ではなく、プラットフォームの必須コンポーネントという位置づけでしか提供できていません。

## 分散型コンピューティングエンジンとしての FPGA

FPGA の世界でも大きな変化が見られます。FPGA はもはや単なるグルー ロジックとしての用途にとどまらず、消費電力、コスト、性能の要件が厳しい多くのアプリケーションで魅力的な代替インプリメンテーションとして台頭しています。FPGA による代替の動きは、民生機器およびモバイル アプリケーションにも広がっています。さらに、演算要素とメモリを備えた通常のアーキテクチャの分散型コンピューティング ファブリックとしても FPGA が利用されています。この結果、FPGA はノイマン型デジタル プロセッサよりも性能、コスト、消費電力のバランスにはるかに優れた擬似シストリック アレイとして注目を集めるようになっていきます。

マイクロプロセッサまたは DSP プロセッサ ベースのデザインと分散型コンピューティング エンジンとしての FPGA の違いを理解することは容易ではありません。そこで、メモリからデータをフェッチしてプロセッサ内部のデータバスに取り込んだ後、データをメモリに書き戻すまでの処理に必要なハードウェア リソースの量を考察してみます。ノイマン型プロセッサアーキテクチャにおいて最大限の性能を発揮するには、分岐予測や投機的実行などのハードウェア リソースを追加する必要がありますが、これらはいずれもシリコンの実

装面積と消費電力の増大を招きます。

では、これらの反復ループを時間領域からアンロールし、FPGA に用意された分散型コンピューティングおよびメモリ リソースの二次元アレイにマップしたらどうでしょうか。データはアレイの片方からもう片方へと流れ、効率的に処理されます。本質的に、プロセッサでソフトウェアを実行した場合、ハードウェアにマップしたものよりも常に効率が悪くなるのはこのためです。ただしこれまでは、FPGA のハードウェア インプリメンテーションはアセンブリコーディングに相当する過程（すなわち、RTL レベルのデザイン）を経なければならず、コンパイラと高級言語を利用できるプロセッサとの間には大きなへだたりがありました。

その結果として、IC のソフトウェア プログラマビリティを可能にすることが強く求められています。ここで言うプログラマビリティとは、当初シリコン コンパイラの先駆者たちが思い描いていたようなものではなく、VLIW (Very Long Instruction Word) コンパイラの世界とハードウェア アクセラレータの世界から発展してきたものです。

以前より、マイクロプロセッサの設計者の間では、一般的なソフトウェア ルーチンや処理時間の長いソフトウェア ルーチンは専用の命令を追加してハードウェアで処理することが理にかなっていると認識されていました。1980 年代に使用されていた Intel 社の 8087 数値演算コプロセッサは、後に洗練されたマルチメディア拡張命令 (MMX) となって IA (Intel Architecture) に追加されました。最近では、ARM コアがコピキタスとなったことを受け、設計者たちはエンベデッド マイクロプロセッサでは処理が困難な冗長なタスクを同じような方法でオフロードしようとするようになりました。そこから現在へ至るまでの流れは、これまでの常識を覆すものであり、少なくとも結果論としてはごく自然な流れであったといえます。

エンベデッド マイクロプロセッサを中心に据え、付随的なハードウェア アクセラレータを配置した新しいアーキテクチャ

は、差別化要因としてのソフトウェアの能力を完全に引き出し、これらをハードウェアにマップすることによって従来の IC (SoC と FPGA の両方) が直面していた難題を根本から変革します。また、このアーキテクチャではハードウェア アクセラレータは高速化されたサブルーチンと見なされるため、ヘテロジニアス マルチコア プログラミングの問題も格段に扱いやすくなります。つまり、労力をかけずにソフトウェアの差別化をすべてハードウェアにマップできるようになるのです。半導体およびシステム企業各社の差別化された独自の IP を数多く使用して構築したシステム ソフトウェアの動作も、洗練された方法でキャプチャできます。しかも、これらは低消費電力で高性能の差別化されたハードウェア パッケージとして提供されるため、メーカーは自社 IP への対価を十分に受け取ることができます。

この結果、新製品の開発サイクルが短縮され、さらに高性能の家電機器、豊富な種類のモバイル インターネット デバイス、革新的なゲーム プラットフォーム、車載インフォテインメントなど、多くの市場に向けてターゲットを絞り込んだ複雑なソリューションを投入できるようになります。

## 無意味な競争を避ける

これらのコンパイラ ツールが FPGA の世界に与える影響はさらに絶大です。というのも、同じパラダイムを利用してソフトウェアをあらかじめ設計された分散型コンピューティング プラットフォームにクリーンにマップできるためです。今や、FPGA は DSP やその他の専用マシンと比べても、消費電力、コスト、性能のバランスに優れ、かつ遜色のないクリーンなプログラミング モデルを備えているため、これらを脅かす存在となりつつあります。しかし、これら 2 つの世界をいたずらに競合させる必要はありません。なぜなら、多くのアプリケーションは規格品の SoC に特定タスク用のハードウェア アクセラレータとしてコンパニオン FPGA を組み合わせて処理するのが理想的であるためです。

こうして、単にプロセッサ上でソフトウェアを実行するものから、プロセッサに SoC のハードウェア アクセラレータを組み合わせたもの、SoC/FPGA の協調処理ソリューション、コアとハードウェア アクセラレータを組み込んだ FPGA、そしてコンピューティング マシンとしての純粋な FPGA まで、あらゆる種類のソリューションが揃うことになります。これらすべての根底にある共通の要素は、プログラミングパラダイムが利用できる点にあります。

今年発表されたレポート『DSP Silicon

Strategies』によると、SoC に組み込まれた DSP の処理能力は現在既に個別 DSP のそれを上回っています。一方、ITRS も組み込みアクセラレータの数が今後爆発的に増加すると予測しています。現時点では、これらが「ソフトウェア プログラマブル IC」と呼ばれるようになるのかどうかは定かではありません。しかし一歩下がって全体像を見てみれば、このトレンドは明白かつ回避できないものであり、従来の常識を覆すものであることが分かります。🌈

Jacques Benkoski 氏は現在、U.S. Venture Partners のポートフォリオ企業数社の業務に協力し、Synfora の取締役会長も務めています。USVP 入社以前は Monterey Design Systems の CEO として活躍。それ以前は、Epic Design Technology、Synopsys、STMicroelectronics、IMEC、IBM で技術職、管理職を歴任しました。テクニオン イスラエル工科大学にてコンピュータ エンジニアリングの学士課程を修了。カーネギー メロン大学にてコンピュータ エンジニアリングの修士課程および博士課程を修了。

# GET ON TARGET



## パートナーの皆様御社の製品・サービスを Xcell journal 誌上で PR してみませんか？

Xcell Journal は、プログラマブル ロジック ユーザーへ、サイリンクス製品/ツールの最新情報をはじめ、システム/アプリケーションの解説、サービス/サポート情報、サードパーティ各社のツール情報などをお届けしています。

現在では日本各地の 10,000 名を超える幅広い分野のエンジニアの皆様が愛読しており、サイリンクスが主催・参加するイベントでも広く配布しています。

貴社製品/ソリューションのプロモーションに非常に効果的なメディアです。

### 広告掲載に関するお問い合わせ先

Xcell Journal 日本語版への広告出向に関するお問い合わせは  
e-mail にてご連絡下さい。

**有限会社 エイ・シー・シー**

**sohyama@jcom.home.ne.jp**



# Xilinx Designs Made Easy



## Active-HDL Designer Edition

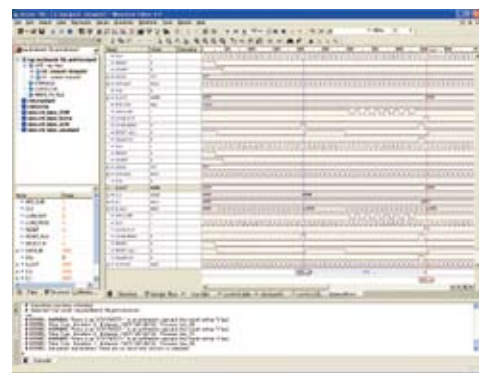
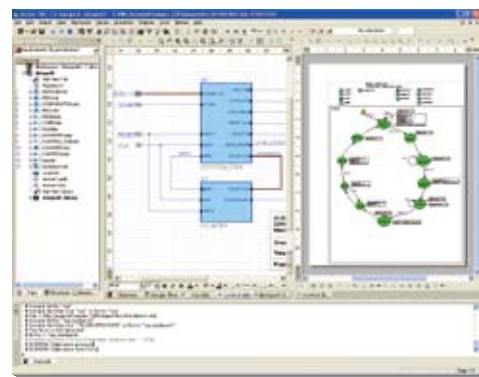
Active-HDL Designer Editionは、Verlog/VHDL混在シミュレーション、グラフィカルデザインエントリ機能を搭載し、大規模かつ複雑化するXilinx社FPGAの設計生産性を大幅に向上します。

### 特長

- **高速シミュレーションエンジン**
  - VHDL/Verilog/SystemVerilog 混在
  - 実行行数制限なし
  - FPGAベンダ提供シミュレータの2倍のパフォーマンス
- **グラフィカル・デザイン・エントリ**
  - ブロックダイアグラム、ステートマシンエディタ
- **Xilinx 社コンパイル済ライブラリの提供**
- **SecureIP サポート**
- **EDK サポート**
- **価格**
  - ¥198,000 (ノードロック年間ライセンス)
  - ¥247,500 (フローティング年間ライセンス)

詳細情報はこちらから

<http://www.aldec.co.jp/DesignerEdition/>



### アルデック・ジャパン株式会社

東京都新宿区新宿1-34-15 新宿エステートビル9F  
TEL: 03-5312-1791 FAX: 03-5312-1795 Email: sales-jp@aldec.com  
<http://www.aldec.co.jp>





# Tools of Xcellence

ザイリンクス パートナー ニュースおよび最新製品情報

Mike Santarini  
Publisher, Xcell Journal  
Xilinx, Inc.  
mike.santarini@xilinx.com

## メインストリームのユーザー層をターゲットにした Aldec 社の混在言語対応シミュレータ

Aldec Inc. (米国 Nevada 州 Henderson) 社の混在言語対応 Active-HDL ツールに、低価格版の Designer Edition (価格 \$1,995) が登場しました。この製品は、FPGA ベンダから提供される RTL シミュレータの 2 倍の速度でシミュレーションを実行します。

Aldec 社マーケティング部門のバイスプレジデントである Dave Rinehart 氏によると、Active-HDL Designer Edition は、FPGA ベンダが提供するシングル言語対応シミュレータ (一般的に \$1,000 未満) とサードパーティのツールベンダが提供する混在言語対応シミュレータ (シングルノードライセンスで一般に \$6,000 以上) の中間の価格帯をターゲットとしています。

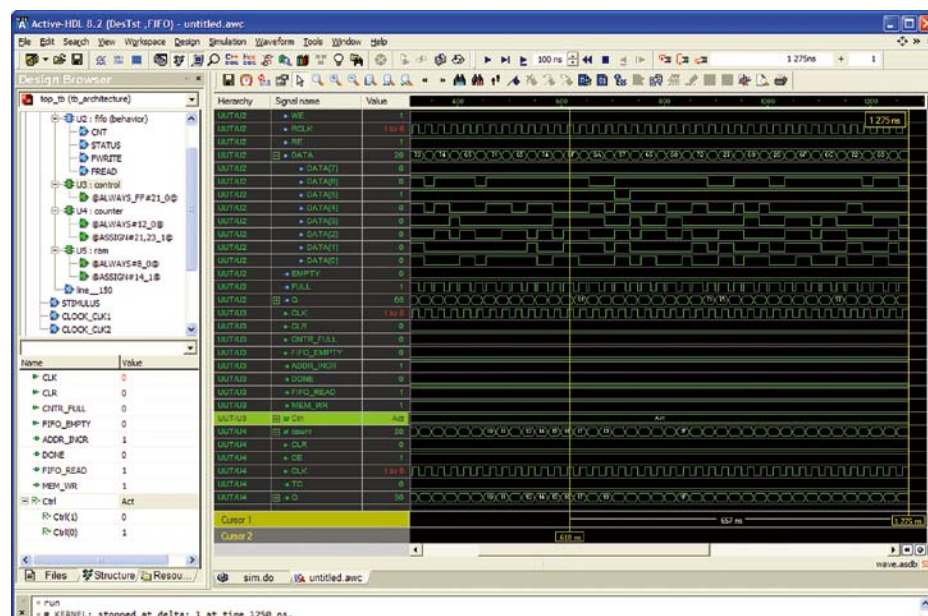
最近では、Verilog と VHDL の IP ブロックを混在させた FPGA デザインが増加しています。また、ASIC 開発の経験が長い設計者なら、高級言語の SystemVerilog や SystemC のブロックを開発、使用することも多いでしょう。

さまざまな言語変換ツールが市販されていますが、設計者にとっては一般的な

言語を混在したままネイティブに扱えるシミュレータの方が便利です。Aldec 社の Active-HDL Designer Edition では、VHDL、Verilog、SystemVerilog のデ

ザイン ファイルを混在させたシミュレーションが可能です。さらに、同ツールは VHDL と Verilog の暗号化 IP やザイリンクスの SecureIP にも対応しており、

Aldec 社の Active-HDL ツール Designer Edition は、FPGA ベンダが提供するシングル言語対応シミュレータと、サードパーティのツールベンダが提供する高価な混在言語対応シミュレータの中間の価格帯をターゲットにした製品です。



FPGA デザインのサイズに対する制約もありません。

Designer Edition は、Aldec 社の Active-HDL シリーズの中で機能制限版と位置付けられています。機能版は Designer Edition よりもシミュレーション速度が数倍も速く、SystemC のサポート、コード カバレッジ、デザイン ルール チェック (DRC)、DSP モデリング、検証などの機能も備えています。Rinehart 氏による

と、Active-HDL Designer Edition はパフォーマンスに制約を加えていますが、前述のようなハイエンド機能を省くことで、メインストリームの設計や検証エンジニアのニーズに特化した製品に仕上がっているといえます。使用頻度の少ない機能を削ることは、Aldec 社にとってサポート スタッフの負担を緩和し、時間とコストを削減するという意味もあります。

また、Designer Edition で Active-HDL

ツールの魅力に触れた設計者の多くが、より完全な機能を求めて完全版にアップグレードするのではないかと Rinehart 氏は考えています。

ノード ロック ライセンスで \$1,995、フローティング ライセンスで \$2,495 であれば、まずは Designer Edition を試してみる価値があるでしょう。

詳細は、[www.aldec.com/Designer Edition](http://www.aldec.com/DesignerEdition) をご覧ください。

## OKI 情報システムズ社の iDMAC for Gen2 国内初となる Virtex-6 用 PCI Express Gen2 x8 レーン対応 DMAC

株式会社沖情報システムズ（群馬県高崎市、以下 OKI 情報システムズ）は、業界最先端の大容量、高速データ転送を可能にした Virtex®-6 向け PCI Express® Gen2、8 レーン対応の国内初の製品化となった iDMAC® for Gen2 を開発、提供を開始しました。これにより、大容量、高速データ転送を必要とする産業用プリンタ、MRI や CT などの医療機器、半導体テストなどのアプリケーション向けに、PCI Express 対応製品の量産に向けた開発体制が整うこととなります。

OKI 情報システムズは、ザイリンクスのアライアンス パートナーとして、ザイリンクス LogiCORE™ エンドポイントとの組み合わせにより大容量データの高速転送を実現する iDMAC (Intelligent DMA Controller) を開発、販売を行っており、多くの納入実績を持っています。新製品の iDMAC for Gen2 は、これまでの iDMAC シリーズの実績をベースに、Virtex-5 世代からの技術継承により開発された最新製品ラインナップで、高信頼性を確保しながら前世代シリーズの 2 倍の性能を実現しています。これを利用することで、PCI Express 規格およびザイリンクスの PCI Express Endpoint Block Plus を意識せずに PCI Express デザインが可能になります。また、デザインに必要な検証済みの周辺リファレンス回路や検証環境

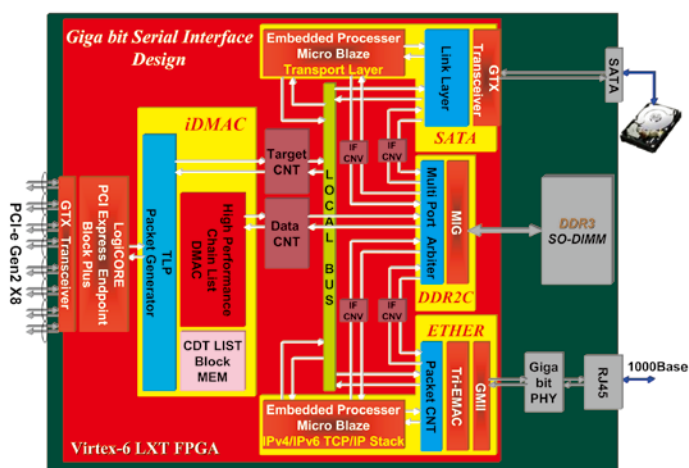
をライセンスに同包しており、ライセンス購入後すぐに設計が開始可能で、iDMAC 製品の開発技術者によるサポートの提供も行います。さらに、iDMAC による PCI Express の転送能力性能を確保するために、FPGA 1 チップでのギガビット イーサネット TCP/IP フルスタック転送や、シリアル ATA インターフェースを行うためのギガビット シリアル インターフェース IP コアの品揃えも拡充しています。

OKI 情報システムズの新聞英文 常務取締役によると、同社はザイリンクス アライアンス プログラムのメンバとなった 2006 年より、多くの日本のユーザーに iDMAC ソリューションを提供してきました。今回、日本で初めて Virtex-6 向けの PCI Express Gen2 x8 レーン対応の新製品 iDMACfor Gen2 を開発できたことにより、FPGA 業界トップであるザイリンクスが提供する最先端 40nm プロセス

FPGA、Virtex-6、使いやすい開発環境のザイリンクス ISE® Design Suite、充実したサポート体制とともに、さらに多くの日本のユーザーを支援します。

iDMAC for Gen2 は、2009 年 11 月 2 日から出荷を開始しており、既に大手メーカー数社で、採用前提での性能評価を実施されています。基本システムによるシングル ライセンスは 250 万円から提供しており、従来製品に比べ性能は 2 倍で価格はほぼ据え置きとなっています。iDMAC for Gen2 製品に関する詳細情報は <http://www.okijoho.co.jp/service/hard/prot/idmac.shtml> で公開しています。

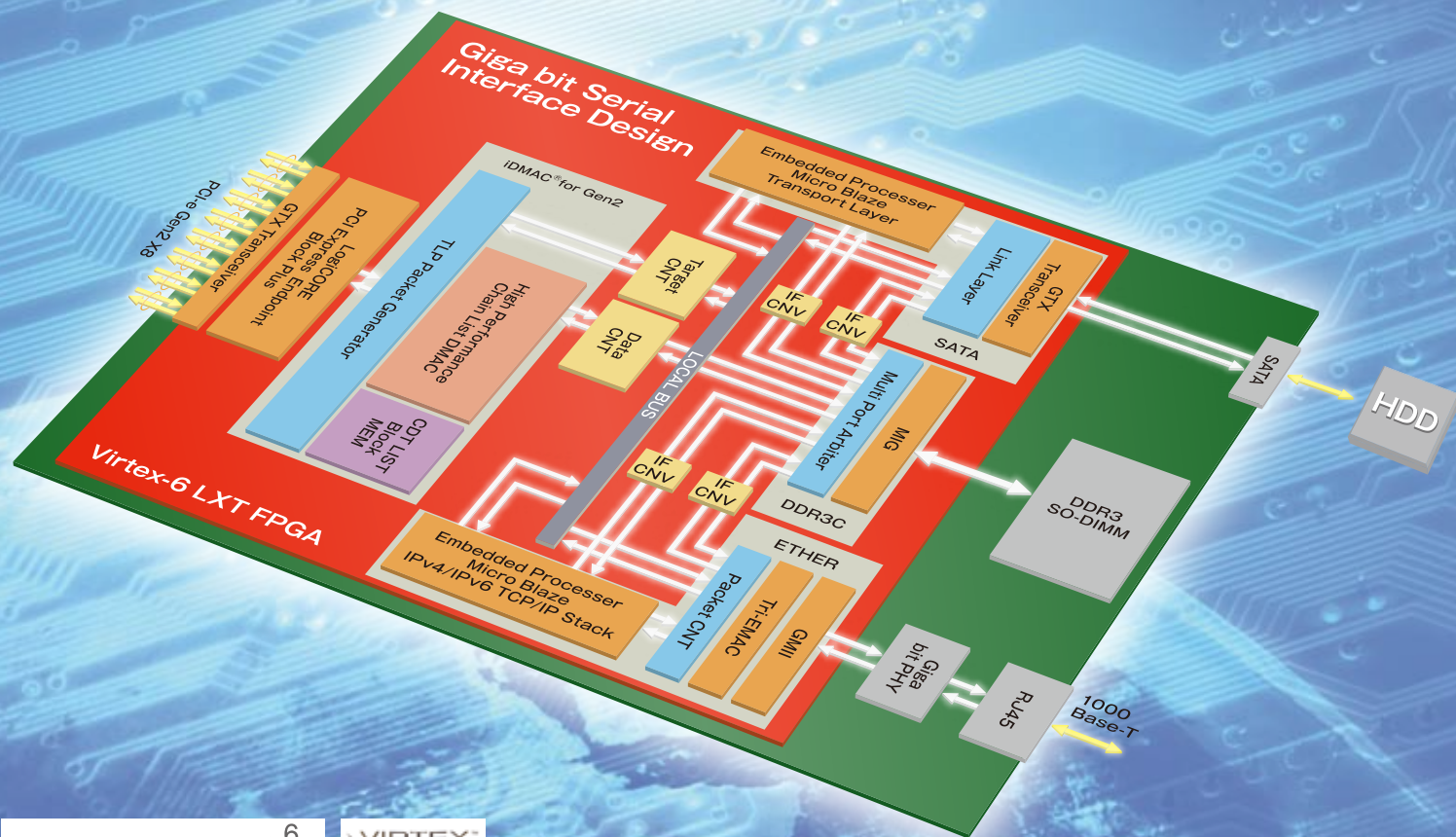
**iDMAC for Gen2 ソリューションにより、ザイリンクス Virtex-6 による PCI Express Gen2 x8 デザインを容易に設計できます。**





# PCI Express® 用 iDMAC® for Gen2 ソリューション

ザイリンクス社製Virtex®-6による  
PCI Express® Gen2 x8デザインを容易に実現可能



## 特長

- ザイリンクス社とのアライアンスプログラムにより実現
- PCI Express® 用ザイリンクスLogiCORE™ Endpoint COREと  
OKI情報製 iDMAC®ソリューションを組み合わせることで、PCI Express®の転送能力を最大限に向上可能
- PCI Express®デザイン構築の為に主要機能をプラットフォーム化してあるため開発期間を大幅に短縮可能
- ユーザ回路部のみの変更(開発)でPCI Express®を使用した各種のアプリケーションに適用可能

Device	PCIe Spec	DMAライト	DMAリード
Vertex-6	Gen2 8Lane	2049(MByte/S)	2402(MByte/S)
	Gen2 4Lane	1298(MByte/S)	1471(MByte/S)
Vertex-5	Gen1 8Lane	1234(MByte/S)	1467(MByte/S)
	Gen1 4Lane	695(MByte/S)	782(MByte/S)

DMAライト: メインメモリ→ローカルメモリ DMAリード: ローカルメモリ→メインメモリ  
※ペイロード部(有効データ)のDMA転送レートを上位PC上の専用アプリケーションで測定。





ニーズに合わせたプログラムを各種取り揃えて好評配信中!!  XILINX®

## FPGA 入門編!

FPGA をこれから始める方に FPGA の全体概要を解説した入門編と、ものづくりにチャレンジする経営者、技術管理者の方へなぜ今 FPGA / CPLD なのかをご説明します。

▶ 30 分で判る! FPGA 入門

▶ 15分で判る! FPGA 採用理由

## FPGA 活用編!

ザイリンクス FPGA を使った最先端デザインの設計手法や、さまざまなアプリケーション設計に求められるデザインチャレンジに対するソリューションを紹介・解説します。

▶ コンシューマ製品の差別化を実現する  
ザイリンクス ソリューション

**NEW!**

▶ Virtex®-6、Spartan®-6 FPGA での  
低消費電力デザインの実現

▶ メモリセントリックな  
システム構築技法

▶ ザイリンクス  
XtremeDSP™ ソリューション

▶ 高速メモリ インターフェイス  
ソリューションと設計手法

## 開発ツール編!

プログラマブルデバイスである FPGA の設計には開発ツールがキーになります。ザイリンクスが提供するユーザーフレンドリーな開発ツールの特徴や使い方、先端設計メソッドについて解説します。

▶ 製品の差別化を実現する開発ツール: ISE Design Suite  
- ターゲット デザイン プラットフォームのための手法 -

▶ PlanAhead で I/O ピン プラン  
ニング

▶ PlanAhead™8.1 階層デザイン  
とその解析ツール

## FPGA / CPLD 概要編!

FPGA の世界トップシェアを誇るザイリンクスが提案するソリューションや、ザイリンクスの最先端 FPGA の詳細を解説します。

▶ ターゲット デザイン プラットフォームで生産性を向上  
- Virtex-6 & Spartan-6 FPGA -

▶ Virtex-5 ファミリの紹介と Virtex-5 LX 概要

▶ ザイリンクス CPLD 応用事例

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



# ACCELERATE CHANGE. ADVANCE INNOVATION.

イノベーション実現へのパワーはあなたの手中にあります。イノベーションの実現に全エネルギーを集中させて、システムの差別化を図ってください。ザイリンクスのターゲットデザイン プラットフォームは、最先端FPGA製品と各種デザイン ツール、IPコア、開発ボード、リファレンス デザインなど革新的なデザインに必要なすべてを統合して、リスクフリーな開発環境として提供します。詳しくは <http://japan.xilinx.com>

## ザイリンクス株式会社

製品のお問い合わせは下記の販売代理店へどうぞ

■東京エレクトロン デバイス(株) TEL (045) 474-7089 x2web@teldevice.co.jp  
■(株)PALTEK TEL (045) 477-2005 info\_pal@paltek.co.jp  
■新光商事(株) TEL (03) 6361-8087 X-Pro@shinko-sj.co.jp

■アヴネット ジャパン(株) TEL (03) 5978-8201 EVAL-KITS-JP@avnet.com  
■富士通エレクトロニクス(株) TEL (045) 415-5825 fei-xinq@cs.jp.fujitsu.com

©2010 Xilinx, Inc. All rights reserved. ザイリンクスの名称およびロゴは米国およびその他の各国のザイリンクス社の登録商標または商標です。その他すべての名称は、それぞれの所有者に帰属します。