Xcell journal

77・78 合併号 2012 AUTUMN

SOLUTIONS FOR A PROGRAMMABLE WORLD

ザイリンクス、SSI テクノロジを採用した世界最高容量の FPGA を出荷開始

7 シリーズ ターゲット デザインプラットフォームがマーケットに登場

AUTOSAR および ISO 26262 規格に対応した ECU を FPGA で設計

柔軟なソフトウェア無線 (SDR) プラットフォームを実現する　Zynq-7000 EPP

デジタル信号処理の課題を FPGA で解決
ページ 39

XILINX®

japan.xilinx.com/xcell/

www.aldec.com/jp

Your Global Xilinx® Verification Solutions Provider

- VHDL, Verilog, SystemVerilog, SystemC
- OVM/UVM, VMM, OS-VMM
- コードおよびファンクション・カバレッジ
- DSP コ・シミュレーション（MATLAB® & Simulink®）
- エミュレーション（ハードウェア・アシステッド検証）
- アサーション・ベース検証＆デザインルール・チェック

アルデック・ジャパン株式会社
〒160-0022 東京都新宿区
新宿2-1-9 ステラ新宿 7F

Phone: 03-5312-1791
Fax: 03-5312-1795
Email: sales-jp@aldec.com
www.aldec.com/jp

Europe
Mercia House
51 The Green, South Bar
Banbury, OX16 9AB
United Kingdom

Phone: +44.1295. 20.1240
Email: sales-eu@aldec.com

Israel
1 Haofe Street
Kadima 60920
Israel

Phone: +972.072.2288316
Email: sales-il@aldec.com

China
Suite 2004, BaoAn Building
#800 DongFang Road
PuDong District
Shanghai City, 200122, P.R. China

Phone: +86.21.6875.2030
Email: info@aldec.com.cn

India
#2145, 17th Main
2nd Cross, HAL 2nd Stage
Indiranagar
Bangalore, 560008, India

Phone: +91.80.3255.1030
Email: sales-in@aldec.com

Taiwan
No. 37, Section 2
Liujia 5th Road
Hsinchu County 302
Zhubei City, Taiwan

Phone: +886.3.6587712
Email: sales-tw@aldec.com

Headquarters-US
2260 Corporate Circle
Henderson, NV 89074
USA

Phone: +1.702.990.4400
Email: sales@aldec.com

© 2012 Aldec と Xilinx は Aldec, Inc. および Xilinx, Inc. の登録商標です。その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

FPGA / CPLD 評価ボード

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。

- 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

Spartan-6使用製品例

Spartan-6 FGG484 搭載 FPGA ボード
XCM-110/110Z シリーズ

ハーフカードサイズ / MRAM 搭載 / DDR2 SDRAM 搭載

XC6SLX45-2FGG484C、XC6SLX75-2FGG484C、XC6SLX100-2FGG484Cまたは XC6SLX150-2FGG484Cを搭載

RoHS 指令対応品

¥44,000(税込46,200)～

Spartan-6 LXT FGG484 搭載 FPGA ボード
XCM-020 シリーズ

クレジットカードサイズ / Rocket I/O / DDR2 SDRAM 搭載 / MRAM 搭載

XC6SLX45T-2FGG484C、XC6SLX75T-2FGG484C XC6SLX100T-2FGG484CまたはXC6SLX150T-2FGG484Cを搭載

RoHS 指令対応品

¥66,000(税込69,300)～

Spartan-6 FGG484 搭載 FPGA ボード
XCM-018/018Z シリーズ

クレジットカードサイズ / MRAM 搭載 / DDR2 SDRAM 搭載

XC6SLX45-2FGG484C、XC6SLX75-2FGG484C、XC6SLX100-2FGG484CまたはXC6SLX150-2FGG484Cを搭載

RoHS 指令対応品

¥40,000(税込42,000)～

Spartan-6 FGG484 搭載 FPGA ボード
XCM-019 シリーズ

クレジットカードサイズ

XC6SLX45-2FGG484CまたはXC6SLX75-2FGG484Cを搭載

RoHS 指令対応品

¥39,000(税込40,950)～

Virtex-5使用製品例

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

XC5VLX30T-1FFG665CまたはXC5VLX50T-1FFG665Cを搭載

RoHS 指令対応品

¥118,000(税込123,900)～

Virtex-5 FFG676 搭載ブレッドボード
XCM-109 シリーズ

ハーフカードサイズ / SDRAM 搭載

XC5VLX30-1FFG676C、XC5VLX50-1FFG676C、XC5VLX85-1FFG676CまたはXC5VLX110-1FFG676Cを搭載

RoHS 指令対応品

¥77,000(税込80,850)～

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

XC5VLXの676ピンBGAチップ搭載

RoHS 指令対応品

¥108,000(税込113,400)～

FPGA Virtex-5 搭載 USB-FPGA ボード
EDX-006

クレジットカードサイズ / USB Config / USB Comm / MRAM 搭載

XC5VLX30-1FFG676C搭載

RoHS 指令対応品

¥99,000(税込103,950)

Spartan-3A/3AN使用製品例

Spartan-3A VQG100 搭載ブレッドボード
XCM-304

セミカードサイズ

XC3S200A-4VQG100Cを搭載

RoHS 指令対応品

¥19,000(税込19,950)

Spartan-3A DSP FGG676 搭載ブレッドボード
XCM-016 シリーズ

クレジットカードサイズ / FRAM 搭載 / SDRAM 搭載

XC3SD1800A-4FGG676CまたはXC3SD3400A-4FGG676Cを搭載

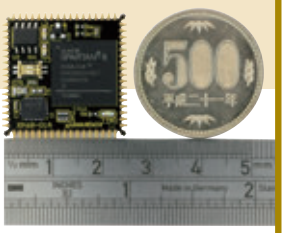
RoHS 指令対応品

¥49,000(税込51,450)～

PLCC68シリーズ

ICソケットに実装できるFPGA/CPLDモジュール

PLCC68シリーズは、ICソケットに搭載できるように設計された、FPGA・CPLDモジュールです。
市販のICソケットに実装できるので、ユニバーサル基板でのFPGAやCPLDの利用に便利です。
また、シリーズ別にピン割付が共通で、FPGA/CPLDのベンダーやタイプによらず電源やI/Oが共通となっています。
単電源動作も、IO電源を2系統分離して使用することも可能となっています。



- 50本のI/O(内4本はクロック兼用を優先的に割付)
- 3.3V単一電源　FPGA/CPLDが必要な補助電源を内蔵
- ICソケットに実装可能な基板サイズ25.3 × 25.3 [mm]
- VIOA、VIOB分離供給可能 (FPGA/CPLDのスペックによる) 最大3.3V
- JTAG信号
- 全シリーズ共通ピン割付
- 全シリーズRoHS指令対応品

twitter Follow me on Twitter

PLCC 68PIN Spartan-6 FPGA モジュール
XP68-01 シリーズ

- XC6SLX16-2CSG225Cを搭載
- 50本のI/Oを外部引き出し
- 3.3V単一電源動作
- オンボードクロック搭載(50MHz)
- コンフィギュレーションROM搭載
- 6層基板を採用
- ICソケットに実装可能

RoHS指令対応品

¥12,800(税込13,440)

PLCC 68ピン Spartan-3AN FPGA モジュール
XP68-02 シリーズ

- XC3S200AN-4FTG256Cを搭載
- 50本のI/Oを外部引き出し
- 3.3V単一電源動作
- オンボードクロック搭載(50MHz)
- コンフィギュレーションROMはFPGAに内蔵
- 6層基板を採用
- ICソケットに実装可能

RoHS指令対応品

¥12,000(税込12,600)

PLCC 68ピン Spartan-6 FPGA モジュール
XP68-03 シリーズ

- XC6SLX45-2CSG324Cを搭載
- 50本のI/Oを外部引き出し
- 3.3V単一電源動作
- オンボードクロック搭載(50MHz)
- コンフィギュレーションROM搭載
- 6層基板を採用
- ICソケットに実装可能

RoHS指令対応品

¥19,500(税込20,475)

※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。詳しくはウエブをご覧ください。

HUMANDATA 有限会社ヒューマンデータ
大阪府茨木市中穂積1-2-10 ジブラルタ生命茨木ビル 〒567-0034
072-620-2002 072-620-2003 s2@hdl.co.jp
URL www.hdl.co.jp

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 的場 千恵
chie.matoba@xilinx.com

進行 周藤 智子
tomoko.suto@xilinx.com

日本語版 制作・広告 有限会社エイ・シー・シー

Xcell PUBLICATIONS

japan.xilinx.com/xcell/

Xcell Journal 日本語版 77・78 合併号

2012 年 8 月 27 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

© 2012 Xilinx, Inc. All Right Reserved.

XILINX や、Xcell のロゴ、その他本書に記載の商標は、米国およびその他各国の Xilinx 社の登録商標です。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。

200 万ロジック セルの FPGA が遂に登場

筆者が IC デザイン メソドロジと半導体ビジネスについての記事を執筆し始めた 1995 年当時、最大容量の FPGA といえばザイリンクスの XC5215 でした。このデバイスは 1,936 ロジック セル、ASIC ゲート換算で 10k とほぼ同等という当時としては圧倒的な容量を実現していました。このように XC5215 は非常に画期的なデバイスであったにもかかわらず、最終製品で使用される場面は限られていました。当時の FPGA といえば 2 つの異なるデバイスを接続するためのグルー ロジックとして使用するか、滅多に必要とされないシステム機能を格納するというのが一般的な用途だったのです。

その頃の ASIC はまだ比較的設計が容易で、製造コストも手頃だったため、最終製品にカスタム機能を搭載する企業は ASIC を自社で製造するか外部に生産委託するのが常でした。しかし、こうした状況はその後一変します。それから 1 年もしないうちに、ASIC 業界は最先端の 0.5 ミクロン プロセスの導入によってもたらされた 10 万ゲートという途方もなく豊富なリソースをいかにして埋めるかという問題に直面したのでした。

結局、多くの議論を経てロジック チップ業界は、これらのゲートを IP ブロックとデザイン再利用で埋めてしまおうという結論に至りました。その後 15 年間にわたってシリコン プロセスはさらに微細化を続け、そのたびに業界はタイミング クロージャ、DFM (Design for Manufacturability : 製造性考慮設計)、ゲート リーク電流など次から次へと新しい課題への対処を迫られました。しかしこの業界のエンジニアたちの創意工夫にはいつも驚嘆の念を抱かずにはいられません。こうした問題に対しても、時間の経過と共に必ず見事な解決策が講じられてきました。

ASIC の製造に関する技術的な課題が解決されたとはいえ、IC 製造にかかるコストはまた別の問題です。現在、マスク コストは数百万ドルにも達しており、これにツールのコストや ASIC の複雑さなどを加味すると、ASIC で対応できるデザインの数というのは全体的に減少の一途をたどっています。最新のプロセス テクノロジを使用して ASIC を製造した場合、数百万台の出荷数が見込めなければ製造コストの回収は期待できません。

幸い、ASIC 以外の分野でもすばらしいエンジニアたちの活躍があります。ザイリンクスのエンジニアたちは、FPGA を単なる ASIC の置き換えにとどまらず、より魅力的なデバイスとするための努力を続けています。今号の「ザイリンクス、SSI テクノロジを採用した世界最高容量の FPGA を出荷開始」では、こうしたエンジニアたちの努力の賜物としてザイリンクスが出荷を開始した Virtex®-7 2000T FPGA について取り上げています。2,000 万ゲート ASIC に相当する 200 万ロジック セルを集積したこの新しいデバイスは業界で最も高容量な FPGA であるだけでなく、エンジニアリングと経済性の両面で ASIC をはるかにしのぐ利点を備えた選択肢として、これからますます多くのアプリケーションで採用されていくでしょう。また、このチップは「スタックド ダイ技術の SSI で 28Gbps I/O を集積し、新時代を迎えるザイリンクスの FPGA」(Xcell Journal 73 & 74 合併号) で筆者が取り上げたザイリンクスのスタックド シリコン インターコネクト (SSI) テクノロジを採用した初の商用デバイスでもあります。

SSI テクノロジとは 1 つのシリコン インターポーザー上に複数のシリコン スライス (すなわちダイ) を並べ、ちょうどプリント基板 (PCB) のトレースを使用してディスクリート IC を接続するようにスライスどうしをインターポーザー内部のトレースで接続するという技術です。しかも、ザイリンクスのツールを使用すればこれらは複数のチップとしてではなく 1 つの巨大なチップとして認識されます。このデバイスの開発に成功したことで、ザイリンクスのエンジニアは半導体業界における本格的な 3D IC 時代の到来に向けて非常に大きな第一歩を踏み出したことになります。

これまで SSI テクノロジの開発に携わってきたザイリンクスのスタッフにとって、開発の 5 年間は確かに困難の連続だったでしょうが、今こうしてデバイスがお客さまの最終製品に組み込まれるようになって大きな達成感に満たされているに違いありません。今度は、ユーザーの皆さまがこのデバイスを使用して画期的な新製品を開発し、新しい市場を切り開き、これまで以上の成功を収めることを願うばかりです。

Mike Santarini
発行人

XILINX® ウェブセミナー Web Seminar

All Programmable FPGA および SoC、3D IC の世界的なリーディング プロバイダーの
ザイリンクスが提供するプログラマブル ロジックからプログラマブル システム
インテグレーション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。

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!!

New!! 新セミナー登場

7 シリーズ ターゲット デザイン プラットフォーム
ザイリンクス シリアル デザイン ソリューション

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ご紹介・解説します。

- Zynq-7000 EPP アーキテクチャとエコシステム
- 7シリーズ FPGAで消費電力を大幅に削減
- ISE12を使用したパーシャル リコンフィギュレーションでシステムのコストと消費電力を最適化
- ザイリンクスで創造するコンシューマ製品の新しい価値

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、先端設計メソドロジについて解説します。

- 次世代FPGA設計手法セミナー PlanAhead デザイン解析ツール ～ 第1部、第2部、第3部、デモ ～
- AMBA AXI4 テクニカル セミナー
- 製品の差別化を実現する開発ツール:ISE Design Suite

FPGA/CPLD/EPP概要編

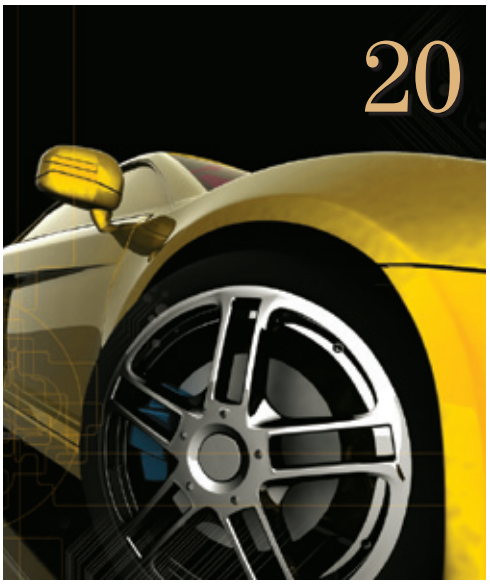
FPGAの世界トップシェアを誇るザイリンクスが提案するソリューションや、ザイリンクスの最先端FPGAの詳細を解説します。

- Zynq-7000 EPP アーキテクチャとエコシステム
- 28nm ザイリンクス 7シリーズ FPGA のアジャイル ミックスド シグナル テクノロジ

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/**

VIEWPOINTS

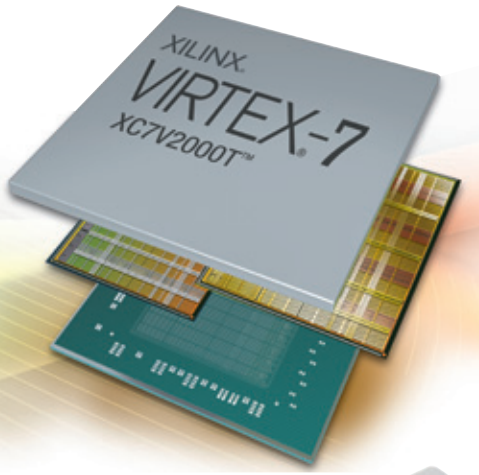
Letter From the Publisher
200 万ロジック セルの FPGA が遂に登場… **2**



Cover Story

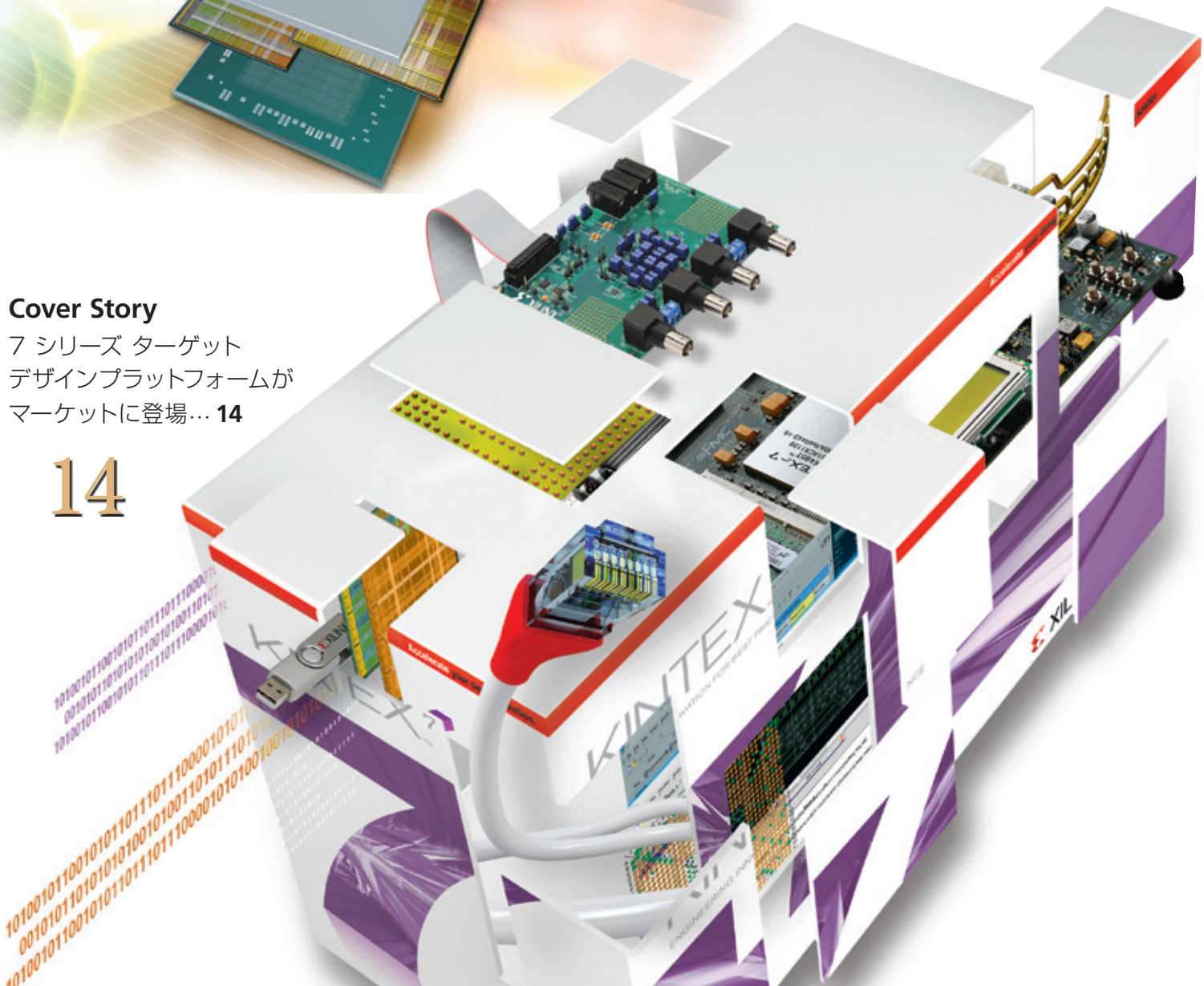
ザイリンクス、SSI テクノロジを採用した 世界最高容量の FPGA を出荷開始

8



Cover Story
7 シリーズ ターゲット デザインプラットフォームが マーケットに登場… **14**

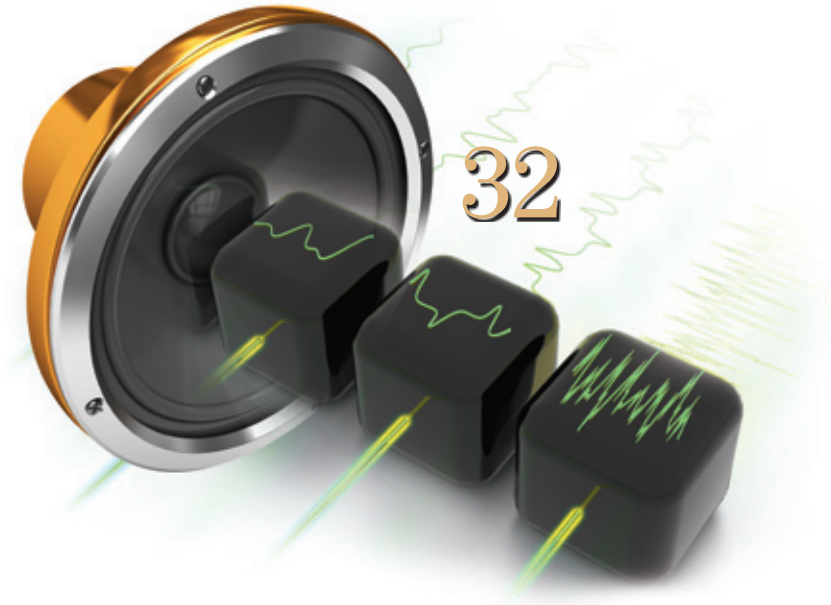
14



XCELLENCE BY DESIGN APPLICATION FEATURES

Xcellence in Automotive Applications
AUTOSAR および ISO 26262 機能安全規格に対応した ECU を FPGA で設計… **20**

Xcellence in Aerospace & Defense
柔軟なソフトウェア無線 (SDR) プラットフォームを実現する Zynq-7000 EPP… **32**



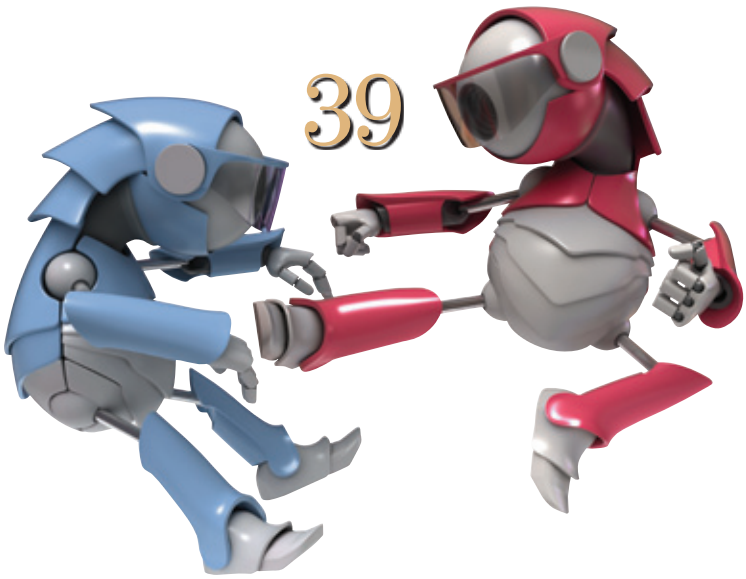
THE XILINX XPERIENCE FEATURES

Xplanation: FPGA101
デジタル信号処理の課題を FPGA で解決… **39**

Xplanation: FPGA101
組み込みシステムにおける浮動小数点演算の活用のヒント… **45**

Xperts Corner
オートモーティブ アプリケーションをにらんだデュアル MicroBlaze Xilkernel システム… **49**

ASK FAE-X
FPGA の消費電力最適化：正攻法的な取り組み… **54**



広告索引

アルデック・ジャパン株式会社 …… 表 2
有限会社ヒューマンデータ…**1**
東京エレクトロン デバイス株式会社 …**6**
株式会社 PALTEK…**7**

Xcell Journal の新規定期購読のお申込みは：
http://japan.xilinx.com/xcell/toroku/

2010 APEX® AWARDS FOR PUBLICATION EXCELLENCE
Excellence in Magazine & Journal Writing
2010, 2011

2011 APEX® AWARDS FOR PUBLICATION EXCELLENCE
Excellence in Magazine & Journal Design and Layout
2010, 2011

Kintex™-7 FPGA ディスプレイキット

Kintex-7 FPGA 評価プラットフォーム ACDC1.0 をベースボードにした 次世代ディスプレイ向けターゲット・デザイン・プラットフォーム

XILINX KINTEX 7™

inrevium
A Revolutionary Innovation Platform

特徴

- Kintex-7 FPGA ディスプレイキットは、ザイリンクス社28nm FPGA Kintex-7シリーズ325Tデバイス、および高速オンボードDDR3 SDRAMを搭載することにより、高速かつ大規模な評価環境を低コストで実現。
- HDMI1.4a、V-by-One® HS、LVDSに対応した業界標準FPGAメザニンカード（FMC）を同梱し、それぞれのカードに対応したリファレンスデザインを付属。

概要

- Kintex-7 FPGA ディスプレイキットは、Pro Edition（TB-7K-ACDC-PRO-TV）とFoundation Edition（TB-7K-ACDC-FND）を用意

TB-7K-ACDC-PRO-TV

TB-7K-ACDC-FND

- **ACDC1.0ベースボード**（TB-7K-325T-IMG）
  - 搭載FPGA：XC7K325T-FFG900
  - 搭載メモリ：DDR3 SDRAM（1866Mbps）2Gbit ×4
  - コンフィギュレーション：QUAD SPI Flash（128Mbit）×1
  - FMCインタフェース(*)：HPC（High Pin Count）×2
    LPC（Low Pin Count）×2
  - 高速ギガビットトランシーバ：5.4Gbps（GTX）×8ch ×2（HPC）
- **FMCカード**
  - TB-7K-ACDC-PRO-TV：HDMI1.4a（Tx/Rx）カード、V-by-One® HSカード、LVDSカード
  - TB-7K-ACDC-FND：HDMI1.4a（Tx/Rx）カード

リファレンスデザイン付属

- HDMIフレームバッファデザイン
- V-by-One® HSデザイン
- LVDSデザイン
- EDKベースシステムビルダデザイン

東京エレクトロン デバイス株式会社
PLD事業部
〒221-0056 神奈川県横浜市神奈川区金港町1番地4 横浜イーストスクエア
TEL：045-443-4016　FAX：045-443-4058
http://ppg.teldevice.co.jp/

XILINX ALLIANCE PROGRAM PREMIER MEMBER

東京エレクトロンデバイスでは自社の持つ情報・技術・サービスを「inrevium（インレビアム）」として商品化しております

PALTEK BOARD SELECTION

PALTEK Board 検索

Virtex-7 2000T 搭載 ASICプロトタイプボード

Virtex-7 2000T 1個、2個、4個、9個

XILINX VIRTEX 7

S2C Leading Innovation to Chip

V7 TAI Logic Module Resource Table Nine					
Product Type	Nine	Quad	Dual	SingleA	SingleB
Target Devices	9*7V2000T	4*7V2000T	2*7V2000T	1*7V2000T	1*7V2000T
ASIC Logic Gates (Max)	180M	80M	40M	20M	20M
FPGA Memory	400Mbits	180Mbits	90Mbits	45Mbits	45Mbits
DDR3 SO-DIMM slot	2	2	1	1	N/A
DDR2 SO-DIMM slot	N/A	2	1	N/A	1
External User I/O (Samtec)	2,400	1,440	1,200	840	840
Inter-FPGA Nets	See Diagram	See Diagram	530	N/A	N/A
Total SerDes Transceivers	32 GTX	32 GTX	32 GTX	16 GTX	16 GTX
PCI Express	x8 Gen2	Use VM*	Use VM*	Use VM*	Use VM*

*Use S2C TAI verification Modules for PCIe applications

FPGAボードはPALTEKへ

DIGILENT BEYOND THEORY

FPGA初心者に人気
Spartan-6
FPGA 評価ボード

価格:¥38,700-（税抜）※輸入運賃含む

アカデミック価格有

ウェブにて最新情報掲載中

http://www.paltek.co.jp/board/index.htm

FPGA 設計品質向上セミナー

FPGA設計、検証戦略の立案とシミュレーションと実機検証の分担

初開催

日時:2012年10月12日(金)10:00～18:00
場所:PALTEK新横浜本社

こんな経験ありませんか?

- 初期動作確認に苦労して、全体の設計スケジュールを大幅に変更せざるを得なかった
- 実機検証に膨大な時間が掛かっている
- 多大な検証時間を費やしたにも関わらず、市場出荷後に不具合が発生した

歴史的に実機評価が重視されるケースが多かったFPGA検証ですが、FPGAの進化、高機能化にともなって、従来の手法に限界が出てきております。FPGAの設計品質の向上がFPGA開発を成功に導く鍵に!

CME CMエンジニアリング株式会社　Mentor Graphics

プログラム概要

- 実機評価とシミュレーションの得手・不得手に関して
- 実機評価とシミュレーションの適用範囲
- 最新のASIC検証手法にみるシミュレーション活用方法とFPGAへの適用に関して
- 【実例】CPU、高速インタフェースを搭載したFPGAの検証手法について

http://www.paltek.co.jp/seminar/extra/cme/fpga.htm

PALTEK セミナー 検索

FPGA 向け RTL 設計 スタイルガイドセミナー

FPGA設計に特化したRTL設計ノウハウとRTL設計スタイルガイドを解説

好評につき随時開催

日時:2012年9月28日(金)10:00～18:00
場所:PALTEK新横浜本社
参加対象者:
・RTL設計における記述スタイルを身につけたいFPGA設計者
・更なるステップアップを求める初中級者

プログラム概要

- FPGA設計でのLVDS、PLL周辺回路作成ノウハウ
- 外部からの初期リセット入力とFPGA初期化
- 非同期クロックドメイン間の転送ノウハウ
- RTL記述ノウハウ
- 演算の記述方法
- テストベンチ記述の基本
- 期待値照合とランダム、アサーション

hdLab

http://www.paltek.co.jp/seminar/extra/fpga/1216FPGA_RTL.htm

株式会社PALTEK

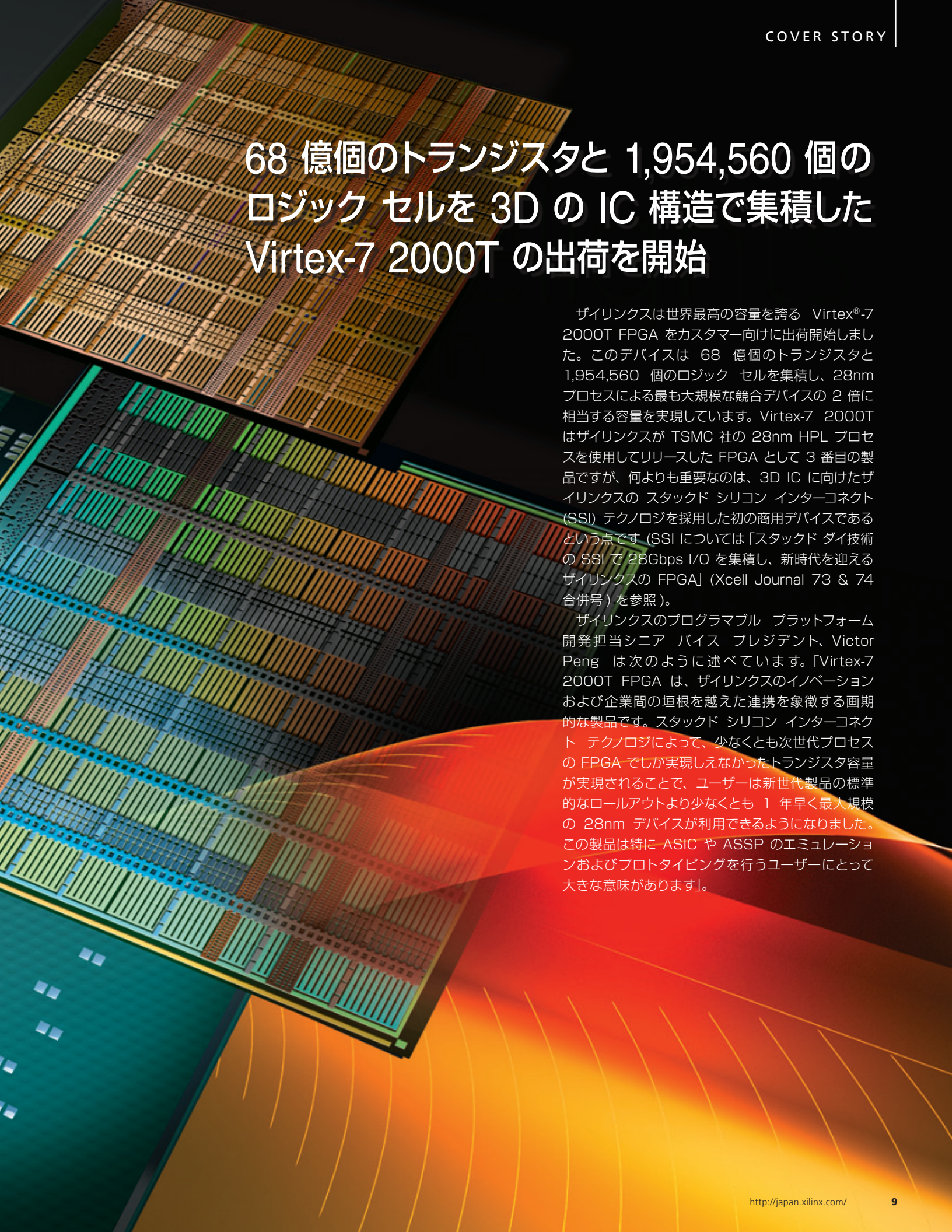
■新横浜本社　横浜市港北区新横浜2-3-12　TEL.045-477-2005
URL　http://www.paltek.co.jp/　E-mail　info_pal@paltek.co.jp
■西日本支社　大阪府吹田市江坂町1-14-33 TCSビル5F　TEL.06-6384-2281

SOLUTION SUPPLIER PALTEK

Xilinx Ships World's Highest-Capacity FPGA Using SSI Technology

ザイリンクス、SSI テクノロジを採用した世界最高容量の FPGA を出荷開始

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



68 億個のトランジスタと 1,954,560 個のロジック セルを 3D の IC 構造で集積した Virtex-7 2000T の出荷を開始

ザイリンクスは世界最高の容量を誇る Virtex®-7 2000T FPGA をカスタマー向けに出荷開始しました。このデバイスは 68 億個のトランジスタと 1,954,560 個のロジック セルを集積し、28nm プロセスによる最も大規模な競合デバイスの 2 倍に相当する容量を実現しています。Virtex-7 2000T はザイリンクスが TSMC 社の 28nm HPL プロセスを使用してリリースした FPGA として 3 番目の製品ですが、何よりも重要なのは、3D IC に向けたザイリンクスの スタックド シリコン インターコネクト (SSI) テクノロジを採用した初の商用デバイスであるという点です (SSI については「スタックド ダイ技術の SSI で 28Gbps I/O を集積し、新時代を迎えるザイリンクスの FPGA」(Xcell Journal 73 & 74 合併号) を参照)。

ザイリンクスのプログラマブル プラットフォーム開発担当シニア バイス プレジデント、Victor Peng は次のように述べています。「Virtex-7 2000T FPGA は、ザイリンクスのイノベーションおよび企業間の垣根を越えた連携を象徴する画期的な製品です。スタックド シリコン インターコネクト テクノロジによって、少なくとも次世代プロセスの FPGA でしか実現しえなかったトランジスタ容量が実現されることで、ユーザーは新世代製品の標準的なロールアウトより少なくとも 1 年早く最大規模の 28nm デバイスが利用できるようになりました。この製品は特に ASIC や ASSP のエミュレーションおよびプロトタイピングを行うユーザーにとって大きな意味があります」。

FPGA ベンダーはこれまで、新しいシリコン プロセスへの移行に伴ってトランジスタ数が 22 カ月ごとに 2 倍になるというムーアの法則を活用し、最新のプロセス テクノロジで新しいアーキテクチャをインプリメントしてきました。事実、この 20 年間、FPGA ベンダーは前世代の FPGA と比較して 2 倍の容量を持つデバイスをムーアの法則のペースに従って着実に提供してきました。

ザイリンクスは Virtex-7 2000T をはじめとする一部の Virtex-7 ファミリ デバイスでムーアの法則を超える (More than Moore) SSI テクノロジを採用しました。このテクノロジは、1 つのパッシブ シリコン インターポーザー上に複数のシリコン スライス (アクティブ ダイ) を横に並べ、インターポーザー内部のメタル インターコネクトを使用して接続するという技術です。これはちょうど、プリント基板のメタル インターコネクトを使用して異なる IC を接続するのとよく似ています (図 1)。この方法で、ザイリンクスはムーアの法則のペースを超えるデバイスの製造に成功しました。Virtex-7 2000T FPGA は、28nm プロセスを採用した最も近い競合デバイスの 2 倍、そしてザイリンクスの最も大規模な Virtex-6 FPGA と比べて 2.5 倍の容量があります。

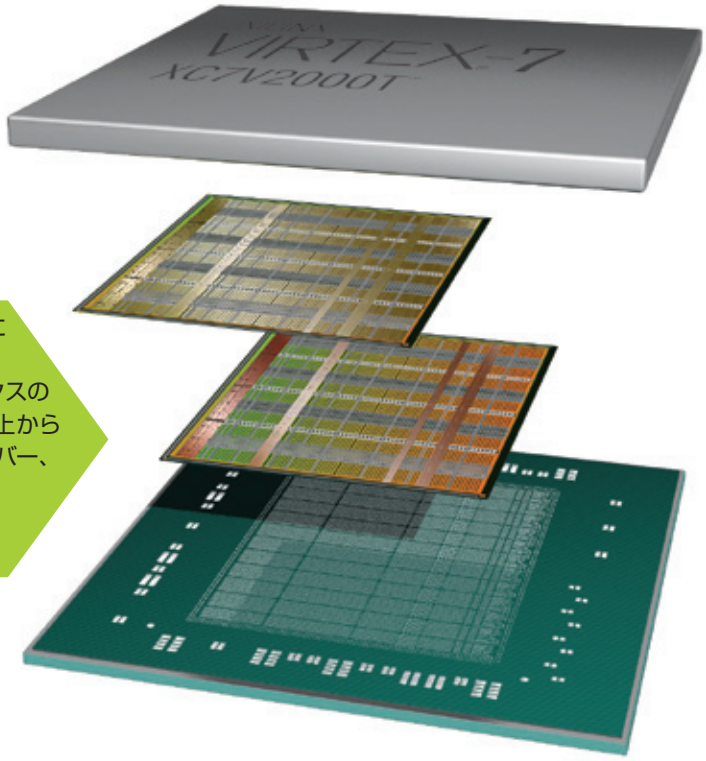
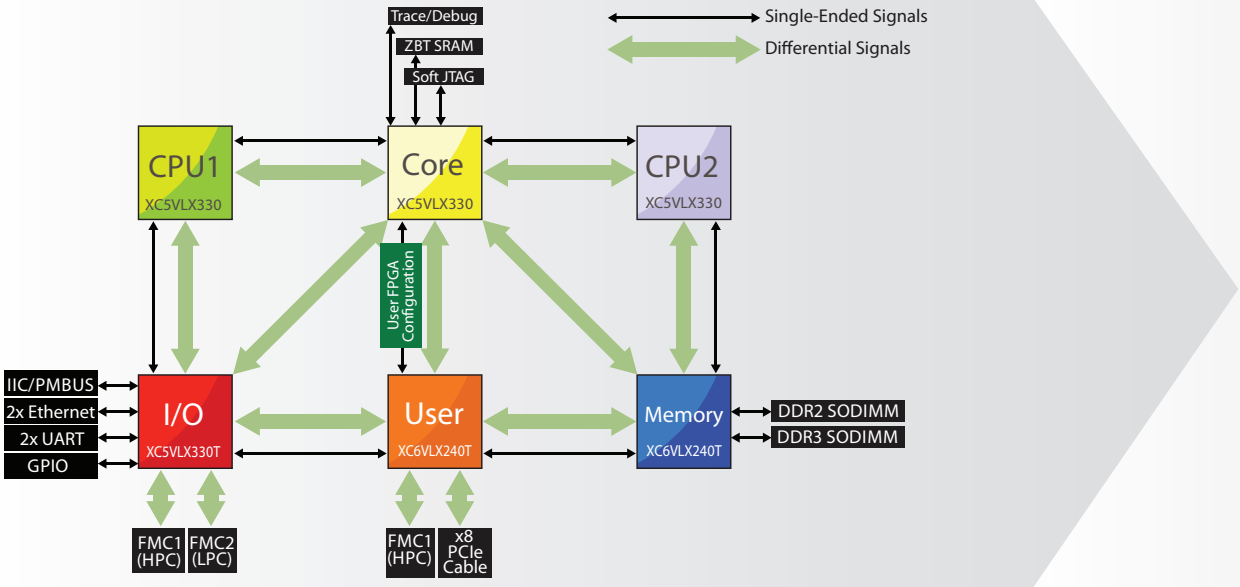


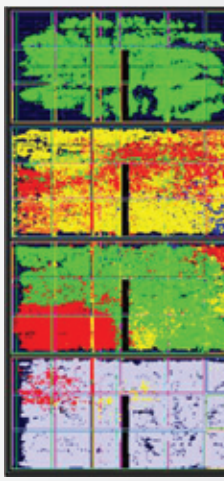
図 1 – 3D IC 時代に向けて大きな一歩を踏み出したザイリンクスの Virtex-7 2000T。上から順に、パッケージ カバー、スライス、シリコン インターポーザー、パッケージ基板。

しかし、ザイリンクスの Virtex-7 FPGA プロダクト ライン マネージャー、Panch Chandrasekaran が指摘するように、このアーキテクチャが特に優れているのは、2000T が 4 つのダイで構成されているにもかかわらず従来の FPGA と同じ使用モデルを維持しているという点です。

ザイリンクスのツール フローとメソドロジ

図 2 – Virtex-7 2000T は非常に容量が大きく、最大 2,000 万 ASIC ゲートの複雑なデザインにも対応でき、NRE コストの大幅な削減も可能。左側の図は、Virtex-7 2000T への実際のデザインのレイアウト例。





を利用すれば、このデバイスを 1 つのきわめて大規模な FPGA としてプログラムできます。

1,954,560 個のロジック セル以外にも、Virtex-7 2000T には合計 305,400 個の CLB (Configurable Logic Block) スライスと最大 21,550k ビットの分散 RAM が集積されています。また、2,160 個の DSP スライス、46,512Kb のブロック RAM、24 個の CMT (Clock Management Tile)、4 個の PCIe® ブロック、36 個の GTX トランシーバー (各 12.5Gbits/s) を備え、24 個の I/O バンクと合計 1,200 個のユーザー I/O があります。

今回リリースされた Virtex-7 2000T はザイリンクスにとって画期的な製品であるだけでなく、半導体業界にとっても 3D IC 時代への大きな前進という意義があります。一方でその真価は、このデバイスが最高容量のデバイスを待ち望むユーザーに対して革新技術への扉を開き、新しいデザイン機能を提供する点にあるとして、Chandrasekaran は次のように述べています。「ソフトウェア開発にシリコン エミュレーションを導入して製品開発期間を短縮したい、複数のチップを 1 つのデバイスに集約したい、あるいは ASIC をデザインに利用できないといった問題や要望を抱える設計チームは、この画期的なテクノロジによって大きな恩恵を受けることになるでしょう。SSI テクノロジを使用してインプリメントしたザイリンクスのデバイスであれば、設計者は次世代の容量をすぐに利用できるようになります」。

ASIC/IP のエミュレーションとプロトタイピング

Gary Smith EDA 社の設計ツール アナリストで ASIC の専門家である Gary Smith 氏によると、現在の平均的なハイエンド ASIC/ASSP デザインは 4 億 2,000 万ゲートに達しており、「11 億に達したものもあると聞いている」といいます。これほどゲート数が増えた結果、ASIC 設計チームの 90% 以上が商用のエミュレーション システムや内製の ASIC プロトタイピング ボードなど、何らかのハードウェアベースの検証システムを使用しています。

FPGA ベンダーが最高容量のデバイスを発表すると、真っ先にそのユーザーとなるのが商用エミュレーション システム ベンダーやデザインのプロトタイピングを担当する設計チームです。商用エミュレーション システム ベンダーは常に最高容量の FPGA の登場を待ち望んでいます。「More than Moore のアプローチで実現した Virtex-7 2000T の大容量を手にすることは、この業界において特に大きな意味を持ちます。彼らは次世代の容量を持つシステムをすぐに提供でき、そのユーザーは設計期間を短縮して画期的な新製品をいち早く市場に投入できるようになるのです」(Chandrasekaran)。

これら商用エミュレーション システムは、エミュレーションの対象となる ASIC や IP、システムの規模にもよりますが、ほとんどが複数のボードとラック数台分の FPGA を使用しています。ユーザーはこのようなエミュレーション システムを使用して短期間で機能検証を完了させ、ハードウェアとしてのデザインをソフトウェア チームに提供することによりソフトウェア開発を早期にスタートさせ、ファウンドリから ASIC の実チップが届く頃にはほとんど開発を完了させることができます。これはすべて Time to Market の短縮を目標としています。

商用エミュレーション システムの典型的な使用モデルとは次のとおりです。まず、ユーザーが ASIC または IP を設計し、従来の EDA 検証ソフトウェアを使用して機能検証を実行します。十分な検証後、商用エミュレーターを用いて RTL (Register Transfer-Level) レベルでデザインをインプリメントし、さらにデザインを検証します。通常、各エミュレーター ベンダーからはザイリンクスの設計ソフトウェアと組み合わせて動作する専用のソフトウェアが提供されており、これを使用して RTL を合成し、エミュレーターで最適な形で FPGA に分散して配置されるように ASIC デザインをブロックに分割することができます。次に、エミュレーター上でデザインを動作させ、エミュレーター ベンダーのソフトウェアで提供されるインターフェイスを利用してワークステーションまたは PC 上で動作する EDA 検証ツールからテストを実行します。

エミュレーター ベンダーは、廉価版のエミュレーターとして「レプリケート」または「プロトタイピング システム」と呼ばれるシステムも提供しています。これらは単に ASIC

の機能をエミュレートするためのものです。現在は多くの企業がこういったシステムをソフトウェア開発チームに支給して、デザイン上で動作するドライバー、ファームウェア、アプリケーションの早期開発に役立てています。

大規模な FPGA を使用すれば、エミュレーション ベンダーはより大容量のエミュレーション システムを構築できるだけでなく、使用する FPGA の個数を減らして消費電力と部材コストを抑えた中～低容量のシステムを構築して、デザインを動作させる全体的なクロック速度を引き上げることもできるとした上で、Chandrasekaran は次のように述べています。「Virtex-7 2000T は容量が非常に大きく、ベンダーは 1 つの FPGA チップでエミュレーターを構築することさえ可能になります。デザイン全体を少ないチップ、あるいはワンチップで動作させると、システムの全体的なパフォーマンスが向上します」。

また、100 万ドルを超える規格品の商用エミュレーション システムには手が出せないというデザインにも Virtex-7 2000T は理想的です。「多くの設計チームが ASIC あるいはシステム全体の機能のプロトタイピングやエミュレーション用にカスタムボードを内製し、ソフトウェアの早期開発を図っています。エミュレーション システムを利用して IC を開発する設計チームでも、ソフトウェア開発チーム向けに FPGA バージョンを内製しているケースもあります」(Chandrasekaran)。また、このデバイスは IP ベンダーが新しい IP ブロックの開発に使用できるほか、コアの機能を見込み客にデモンストレーションする目的でも使用できるなど、IP ベンダーにも大きな魅力があると Chandrasekaran は付け加えています。

システム アーキテクチャの集約と消費電力の削減

新しい Virtex-7 2000T は ASIC や IP のエミュレーションとプロトタイピングに役立つだけでなく、システムの高性能 / 高機能化と省電力化を両立させたいと考えるシステム アーキテクトにも大きな魅力があります。

「市場には複数の FPGA を使用した最終製品が数多く存在します。Virtex-7 2000T を使用すれば、複数の FPGA の機能を 1 つの FPGA に統合できます。このシステムの統合によりすべての機能がワンチップに集積されるため、パフォーマンスが向上します。しかもボード上の複数の IC を I/O インターフェイスで接続する必要がなくなるため、消費電力も削減されます。I/O インターフェイスの消費電力は I/O の数に比例します。つまり、デザインの性能が高いほど、そしてシステムに含まれる IC の数が多いほど消費電力は大きくなります」(Chandrasekaran)。

しかもシステム機能を複数の IC に分割する作業は複雑で、開発期間の長期化とテスト費用の増大を招きます。システム内の複数のデバイスを 1 つに集約できればこのような機能分割に伴う課題を解消し、検証やテストにかかるコストを削減できます。「競合 FPGA の 2 倍以上の容量を持つ Virtex-7 2000T を利用すれば、これまで以上にシステムの統合が進み、消費電力をマルチチップ ソリューションの 1/4 に抑えることも可能です。また、I/O のボトルネック解消によるシステム性能の向上、あるいは不必要なデザイン分割をなくすことによるシステムの複雑さ解消といった利点も得られます。大幅に削減されるボード面積を利用してほかの機能を追加するか、製品のサイズを小型化するかをアーキテクトが選択できます」(Chandrasekaran)。

7 シリーズのほかのデバイス同様、ザイリンクスは Virtex-7 2000T を TSMC 社の FPGA 専用 28nm HPL (High-Performance、Low Power) プロセス テクノロジを使用してインプリメントしました (詳細は、「7 シリーズ FPGA で消費電力を半減」(Xcell Journal 75 & 76 合併号) を参照)。HPL プロセスを採用した結果、Virtex-7 2000T のトランジスタは 28nm 高性能プロセスを使用してインプリメントした競合デバイスよりもリーク電流が低く抑えられていると Chandrasekaran は述べています。つまり、Virtex-7 2000T の消費電力はほぼ半分の容量の他社製デバイスと同レベルということになります。

ASIC の置き換え

これも非常に重要な点ですが、Virtex-7 2000T は、28nm プロセス ノードでの ASIC/ASSP 開発のコストとリスクを受け入れることが難しい多くのデザインにとっても魅力的な選択肢となります。シリコン プロセスの進歩に伴い、設計と製造のコストは急速に増大しています。28nm ノードでは、ASIC/ASSP の NRE (Non-Recurring Engineering : 初期開発費用) コストは 5,000 万ドルを超え、ASIC リスピンの可能性はほぼ 50% にまで達しています。設計サイクルでミスを 1 つ見逃してしまうだけで収益性は大きなダメージを受け、こうしたミスがいくつも重なるとプロジェクトのキャンセルや市場機会の損失、さらには会社自体の存亡の危機すら招く恐れがあります。

Virtex-7 2000T は 1,000 万～ 2,000 万ゲート相当の ASIC を置き換えることができ、ASIC につきものの NRE コストも発生しません。「1 つの小さなミスがマスクの作り直しという壊滅的な結果をもたらすことを懸念する必要がなくなり、設計者は設計そのものに集中できるようになります。しかも、Virtex-7 2000T は再プログラム可能なため、ミスがあってもプログラムし直すだけで済みます」(Chandrasekaran)。

これまでのメソドロジを継承

Virtex-7 2000T は非常に大規模なデバイスですが、そのプログラミング方法はこれまでとそれほど大きく変わっていません。

「数年前から、ザイリンクスは超大容量のデバイスを念頭に設計ツールの最適化を進めてきました。現在では、分割とフロアプランを効率よく行い、消費電力とパフォーマンスを最適化できるようになっています」(Chandrasekaran)。大規模な FPGA はほぼ例外なくと言っていいほど何らかの分割が必要になるのが一般的で、タイミングのクリティカルなファンクションを互いに近接して配置する必要があると Chandrasekaran は説明しています。Virtex-7 2000T に大規模なデザインをインプリメントする場合、ザイリンクスのツールはタイミングとパフォーマンスが最適となるようなフロアプランと分割をサポートします。

ザイリンクスの設計ツールは最新版で Virtex-7 2000T をサポートしており、ユーザーはすぐにでも Virtex-7 2000T を使用した設計を開始できます (Chandrasekaran)。ザイリンクスは来年にもモノリシックおよび SSI 構成の新しい Virtex-7 FPGA のリリースを予定しています。

Virtex-7 2000T の詳細とデモについては、japan.xilinx.com/7 をご覧ください。

比類なき生産性

次の10年の「All Programmable」デバイスに対応する Vivado™ Design Suit

技術革新をサポートするため

主要な設計の障壁を解消します

Vivadoツールはプログラマブル ロジックや I/O設計にかかる時間短縮を加速させるだけでなく、3Dスタックド シリコン インターコネクト テクノロジや ARM® プロセッシング システム、アナログ ミックスド シグナル (AMS)、および多くの IPコアが組み込まれたデバイスへのプログラマブル システムのインテグレーションとインプリメンテーションを短時間で実現します。

比類なき生産性の始まりです

http://japan.xilinx.com/designtools

XILINX®
ALL PROGRAMMABLE

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)443-4016 x2web@teldevice.co.jp ■アヴネット ジャパン(株) TEL(03)5792-8210 EVAL-KITS-JP@avnet.com ■(株)PALTEK TEL(045)477-2005 info_pal@paltek.co.jp
■富士通エレクトロニクス(株) TEL(045)415-5825 fei-xinq@cs.jp.fujitsu.com ■新光商事(株) TEL(03)6361-8086 X-Pro@shinko-sj.co.jp

©Copyright 2012 Xilinx, Inc. All rights reserved. ザイリンクスの名称およびロゴ、Vivado、その他記載のブランド名は米国およびその他の国におけるザイリンクス社の登録商標または商標です。ARM は、EU およびその他の国における ARM Limited の登録商標です。他のすべての商標はそれぞれの所有者の財産です。

Charge to Market with Xilinx 7 Series Targeted Design Platforms

7 シリーズ ターゲット デザイン プラットフォームがマーケットに登場

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

これまでの開発キットとは一味違う ザイリンクスの Kintex-7/Virtex-7 FPGA キットで FPGA デザインをスムーズに

ザイリンクスは 28nm プロセス採用の 7 シリーズ FPGA に対応するターゲット デザイン プラットフォーム (TDP) の第一陣として 3 種類の製品を発表しました。これら TDP を利用することで、ユーザーはいずれの FPGA が自社デザインに最適となるかを評価し、すぐに設計を開始できるようになります。Virtex®-7 FPGA および Kintex™-7 FPGA のベース キットには、アジャイル ミックスド シグナル (AMS) ドーターカードと業界標準の FMC コネクタが採用されています。この 2 つのベース キットに加え、FPGA を使用して複数の DSP 処理をより高速に実行したいと考えるユーザー向けに、Avnet 社から Xilinx Kintex-7 FPGA DSP キットがリリースされました。

これら 3 種類の新しいキットは、FPGA システム開発の目的でザイリンクスが展開を開始したプラットフォームの第 2 世代となるもので、その最初の製品としてリリースされたものです。ザイリンクスは 2009 年に 6 シリーズのデバイス群をサポートする形でターゲット デザイン プラットフォーム (TDP) 戦略を発表しました (Xcell Journal 日本語版 67 & 68 合併号の「ターゲット デザイン プラットフォームで技術革新を加速」を参照)。ザイリンクスのプラットフォーム戦略は、4 層からなるピラミッドの形で表現されます。4 つの層とは、ベース プラットフォーム、ドメイン プラットフォーム、マーケット プラットフォーム、そしてその頂上に来るのがユーザー デザインです (図 1)。この TDP 戦略は、ザイリンクスとザイリンクス エコシステムから提供される構築済みのソリューションを、自らの FPGA デザイン プロジェクトに簡単に加えてもらうことが狙いです。TDP をデザインの基礎に置くことで、ユーザーは製品を差別化する機能の設計に集中することができます。

今回リリースされた Virtex-7 FPGA VC707 評価キットおよび Kintex-7 FPGA KC705 評価キットは、ザイリンクスの TDP 階層の中のベース プラットフォームに位置づけられるもので、ザイリンクスのデバイス ファミリを評価することが主な目的で使用できるものです。これらのベース TDP に加えて、たとえば今回 Avnet 社からリリースされたドメイン別のキットを使用することで、ユーザーは DSP、エンベデッド、コネクティビティといったドメインにデザインを特化させる作業が容易になります。さらに設計時間を短縮するために、航空宇宙 / 防衛、オートモーティブ、民生機器、無線 / 有線通信、商業放送、工業 / 化学 / 医療などの産業分野の各マーケットに特化した開発プラットフォームも提供しています。7 シリーズ向けのこれらのマーケット特化キットの各バージョンも間もなく発表の予定です。

ザイリンクス TDP にはいずれも、FPGA を設計するための開発ボードやソフトウェアに加え、ザイリンクスやサードパーティ製の IP コア、リファレンス デザインおよびドキュメント、自社開発または提携企業開発によるさまざまな FPGA メザニン カード (FMC) ドーターカードが含まれています。これらのリソースが全体として強力な力を発揮し、ユーザーのデザインをシステム レベルで動作させるまでの時間がこれまで以上に短縮されます。

ザイリンクスは、これら 3 階層すべてにおいて 6 シリーズ デバイス向けの TDP を既に提供していますが、これからは、7 シリーズ デバイス用 TDP の展開開始となります。VC707、KC705、Kintex DSP の各キットは、今後数カ月で続々と登場する 7 シリーズ用キットの先陣を切る製品となるものです。

ザイリンクスでシニア戦略マーケティング

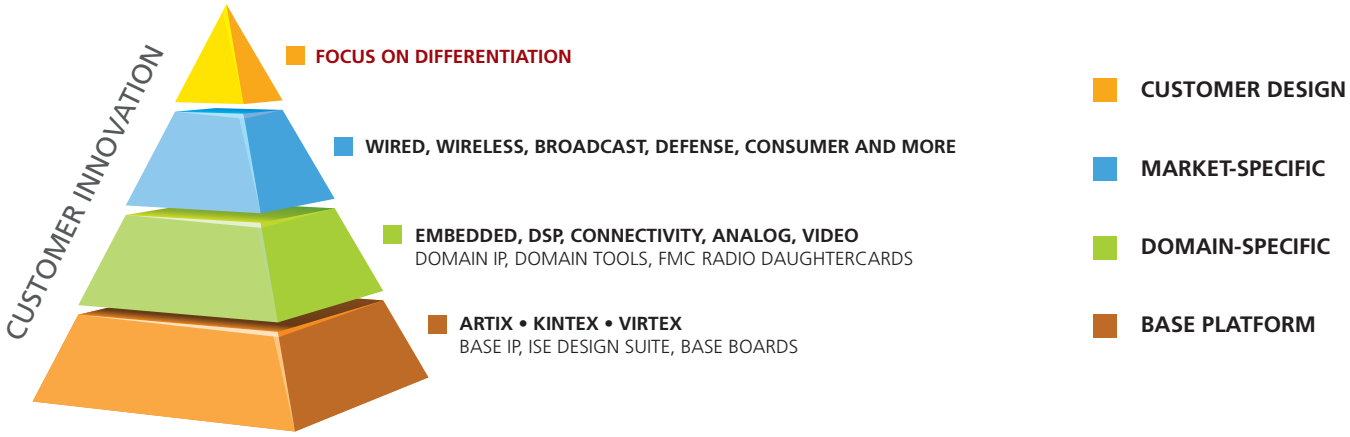


図 1 - 製品の差別化や開発期間の短縮が図れる、ザイリンクスのターゲット デザイン プラットフォームのコンセプト

マネージャーを務める Mark Moran は、TDP は従来の開発キットとは別物であると強調しています。プログラマブル チップ ベンダーならどこでもキットを提供しており、一般に開発ボードや電源コード、ソフトウェア、時にはコネクタなどが含まれています。

「通常のキットでは、まずソフトウェアを PC にダウンロードしてボードを接続し、電源を入れたのち、LED の点灯によって動作しているのがわかるという程度でした。さらに詳しく知りたい場合は、ベンダーのウェブサイトにアクセスして使用しているチップとボードに対応するドキュメントや IP を探さなくてはなりません。基本的な構築ブロックに関する情報が見つかるだけでも運が良いほうです。見つかったとしても、アップデートされていない古い資料という可能性も少なくありません。ベンダーの中には、単に動作させるのに必要な、ありふれた IP ブロックの購入を要求するところもあります。このようなベンダーからキットを買うことは、タイヤのない車を購入するようなものです。これでは運転席に座ってエンジンを掛けることができても、ドライブに行くことはできません」(Moran)。

さらに、似たような目的のキットを多種多様な形で用意しているベンダーも少なくありません。ユーザーにとっては要件に適したキットを見きわめるのが難しいばかりか、ベンダーにとっても質の高いサポートを提供するのが難しくなっていると、Moran は指摘します。「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、通常のキットと比較してより多くのものを提供します。いわばタイヤのない車ではなく、レース用のタイヤを装着し、ガソリンも満タン、チューニングも万全、すぐに高速で走り出せるフェラーリのようなキットです」(Moran)。

ザイリンクスでターゲット デザイン プラットフォームのシニア プロダクト マーケティング マネージャーを務める Evan Leal は、ターゲット デザイン プラットフォーム (TDP) として提供されるキットには、ボード、コネクタ、設計ツールに加えて、多彩なリファレンス デザインや無償の IP コア、場合によっては FMC カードも集約していると説明します。さらに、IP コア ライブラリと、ザイリンクス製およびサードパーティ製のプラグアンドプレイ FMC ドーターカードもすぐに利用できます。Moran は、「今回発表した TDP は当社の第 2 世代目に該当することから、これらの新しいキットをサポートする非常に充実したエコシステムが既に確立されていることを意味します。さらにこの世代のデバイスでは、ユーザーの作業開始をより一層容易かつ迅速にしています」と述べています。ザイリンクス アライアンス プログラム (http://japan.xilinx.com/alliance/index.htm) には、ボード、デザイン サービス、IP コア、ツールなどのベンダーが数百社加盟しており、ザイリンクスのターゲット デザイン プラットフォームを活かす豊富なソリューションやアプリケーションを提供しています。この度の Kintex-7 FPGA および Virtex-7 FPGA 用の両キットの発売にあたっては、4DSP、Analog Devices、Avnet、Northwest Logic、The Mathworks、Texas Instruments、および Xylon の各社に密接な協力を仰ぎました。

プラグアンドプレイ FMC カード

ザイリンクスの TDP によるアプローチの重要な特長のひとつは、Virtex-6 および Spartan®- 6 FPGA 用のキットを始めとして、すべてのキットに業界標準の FMC コネクタを採用したことです。FMC コネクタの採用によって、IP コア ベンダー、関連チップ メーカー、デザイン サービス企業が提供するドーターカードを使用してボードをすばやくザイリンクスのキットと接続し、デザインをシステム レベル環境で検証できるようになります。

Moran によると、2009 年に 6 シリーズで始まった TDP のエコシステムにおいては、現在既に 30 社以上のパートナー企業から提供される 100 以上の FMC カードが存在しますが、ここ数カ月間にザイリンクスが行ったプラグフェスト (接続試験会) では、多様な用途をカバーする 30 種以上の FMC を対象に検証を行った結果、新しい 7 シリーズ デバイスのプラットフォームとの互換性について確認が取れています。ザイリンクスは、今後数カ月の間にさらに多くの FMC カードが市場に登場すると予想しています。

7 シリーズ デバイスは、TDP 戦略そのものにも本質的な向上効果をもたらしました。Moran はその例として、昨年発売された 7 シリーズ デバイスに ARM® AMBA® AXI4 インターフェイスを導入した点を挙げています。AXI4 により、ザイリンクス製のものだけでなく、サードパーティ製およびユーザー独自の IP コアと、ザイリンクスの 7 シリーズ デバイスとの統合がより簡単になりました。また、ユーザー企業もザイリン

図 2 - Virtex-7 FPGA が採用されているシステムで求められる画期的な性能、容量、電力効率を必要とする設計者を対象にした、Virtex-7 FPGA VC707 評価キット

クス IP コア エコシステムの企業も、1 種類の IP コア インターコネクト構造にのみ集中できるため、IP コア製品の改良や新製品の開発も短縮が図れます。

新世代 TDP の別の大きなメリットとして、ザイリンクスが Virtex-7、Kintex-7、Artix™-7 に加え、Zynq™ -7000 エクステンシブル プロセッシング プラットフォームを含めたすべての 7 シリーズ デバイスをスケーラブル プログラマブル ロジック アーキテクチャで実装したという点が挙げられます。Moran によると、この統一アーキテクチャのおかげで、デザインの要件の進捗に従って複数の FPGA 間で行う IP コアの実装とデザインの移行が容易になります。これによってエンドユーザーは、デザインの再利用と評価が簡単に無駄なくできるようになります。

7 シリーズ デバイスのもうひとつの大きな特長として、Kintex-7 デバイスにもアジャイル ミックスド シグナル ブロックが搭載された点を Moran は付け加えます。これによってユーザーはデザイン内の内部ブロックをモニタリングおよびテストできるほか、ADC などのアナログ機能を別の外付け回路としてではなく FPGA 上に直接実装することができるようになります。結果、パフォーマンスの向上、ボード スペースの節約、部品コストの削減にもつながります。さらに、デバイスの消費電力のモニタリングも可能になり、この点でも他社製品に対するザイリンクス 7 シリーズの優位性が確認できると Moran は述べています (Xcell Journal 75 & 76 合併号の「7 シリーズ FPGA で消費電力を半減」を参照)。

「AMS ブロックは 7 シリーズの前までは Virtex デバイスにしかなかった機能ですが、7 シリーズでは Kintex デバイスにもこのブロックを搭載しました。AMS ブロックの恩恵を受けるユーザーが一気に広がることは間違いありません」(Moran)。

今回発表した 7 シリーズ向けの最初の TDP である 3 種類のキットには、ユーザーの作業開始とデザインを迅速化する機能が満載されています (Moran)。各キットで提供される多彩な機能の詳細は japan.xilinx.com をご覧ください。Moran および Leal が考える各キットの概要を以降のセクションにまとめておきましょう。

Virtex-7 FPGA VC707 評価キット

ザイリンクスの Virtex-7 FPGA VC707 評価キットは、Virtex-7 FPGA が採用されている多くのシステムで求められる、画期的な性能、容量、そして電力効率を必要とする設計者にとっては基本の TDP となる製品であると Moran は解説します。対象とするアプリケーションとしては、有線 / 無線通信、航空宇宙 / 防衛、医療、放送といった市場で使用される高度なシステムが挙げられます (図 2)。

このキットは Virtex-7 FPGA デバイスの中でも中規模な製品に該当する VX485T-2 FPGA に特化したものです。

ソフトウェアに関しては、Virtex-7 VX485T-2 FPGA にデバイス ロックされた ISE® Design Suite Logic Edition が同梱されています。シリアル コネクティビティとしては、ギガビット イーサネット (GMII、RGMII、SGMII)、SFP/SFP+ トランシーバー コネクタ、4 個の SMA コネクタ付き GTX ポート (TX と RX)、そして PCI Express® x8 エッジ コネクタが用意されています。

図 3 - 低消費電力かつ高速の通信および処理が求められる、幅広いアプリケーションを対象にした Kintex-7 FPGA KC705 評価キット

パラレル コネクティビティとしては 2 個の FMC-HPC コネクタ (8 個の GTX トランシーバー、160 本のシングルエンドまたはユーザー定義可能な 80 本の差動信号) が用意されています。メモリは、1,600Mbps の SODIMM DDR3 メモリ、PCIe® コンフィギュレーション用の 1Gb (128MB) BPI フラッシュ、SD カード インターフェイス、および 8kB IIC EEPROM が搭載されています。

その他のコネクティビティとして、HDMI ビデオ出力、UART-to-USB ブリッジ、2×16 LCD ヘッダー、8 個の LED、IIC およびアナログ ミックスド シグナル ポートなどが装備されています。さらに、AMS ブロックを速やかに評価できるように、AMS 評価カードも同梱されています (Leal)。

キットにはリファレンス デザインやデモ、およびドキュメント一式が同梱され、ユーザーはすぐに作業を開始できます。デザイン サンプルとデモには、ボード診断デモ、ChipScope™ Pro シリアル I/O ツール キット、IBERT トランシーバー テスト デザイン、マルチブート リファレンス デザイン、PCI Express Gen 2 (x8) テスト デザイン、および DDR3 メモリ インターフェイス デザインが用意されています。

ドキュメントに関しては、入門ガイド、ハードウェア ユーザーガイド、リファレンス デザインおよびサンプル デザインのユーザーガイドが提供されています。また、ベスト プラクティスに基づいてボードのレイアウトや開発時間を短縮するのに必要な情報を収めた回路図と UCF ファイルも添付されています。

デバイス コンフィギュレーション、クロッキング、制御、消費電力など、このボードに関する詳細は、Virtex-7 FPGA VC707 評価キットの専用ページ (http://japan.xilinx.com/products/boards-and-kits/EK-V7-VC707-G.htm) をご覧ください。

Kintex-7 FPGA KC705 評価キット

Kintex-7 ファミリ向けのベース TDP は、XC7K325T-FF900-2 FPGA を搭載した Kintex-7 FPGA KC705 評価キットです。Moran によると、このキットは、無線 / ベースバンド、レーダー、Edge QAM、トリプルレート SDI に代表される低消費電力かつ高速の通信および処理が求められるシステムにおいて、開発期間の短縮を実現する最大の柔軟性を備えた点が特長です (図 3)。

RoHS 準拠の Kintex-7 FPGA KC705 評価キットは、インパクトの大きな優れた機能やコスト削減につながる機能を数多く備えています。このキットには Kintex-7 XC7K325T FPGA にデバイス ロックされた ISE Design Suite : Logic Edition が同梱されています。

シリアル インターフェイスとしては、ギガビット イーサネット、SFP/SFP+ トランシーバー コネクタ、4 個の SMA コネクタ付き GTX ポート (TX、RX)、UART-to-USB ブリッジ、および PCI Express x8 エッジ コネクタが用意されています。

パラレル インターフェイスとしては、FMC-HPC コネクタ (4 個の GTX トランシーバー、116 本のシングルエンドまたはユーザー定義可能な 58 本の差動信号 (34 本の LA および 24 本の HA)) および FMC-LPC コネクタ (1 個の GTX トランシーバー、68 本のシングルエンドまたはユーザー定義可能な 34 本の差動信号) がそれぞれ 1 個ずつ搭載されています。

その他のコネクティビティとして、HDMI ビデオ出力、2×16 LCD ディスプレイ コネクタ、8 個の LED、IIC、LCD ヘッダー、およびアナログ ミックスド シグナル ポートが装備されています。

図 4 - 複数の DSP 処理を FPGA を使用してより高速に実行したいと考えるユーザーを対象にした、Avnet 社の Kintex-7 FPGA DSP キット

メモリは、1,600Mbps の SODIMM DDR3 メモリ、PCIe コンフィギュレーション用の 1Gb (128MB) BPI フラッシュ、SDIO-SD カード インターフェイス、16MB クワッド SPI フラッシュ、および 8kB IIC EEPROM が搭載されています。

キットには、ボード診断デモ、ChipScope Pro シリアル I/O ツールキット、IBERT トランシーバー テスト デザイン、マルチブート リファレンス デザイン、PCI Express Gen 2 (x8) テスト デザイン、および DDR3 メモリ インターフェイス デザインが含まれています。「この評価キットで提供される PCIe x4 Gen 2 および DDR3 FPGA に特化したターゲット デザインを使用すると、FPGA 設計者によって多く使用される機能を短時間で統合できるようになります」(Leal)。

ドキュメントに関しては、入門ガイド、ハードウェア ユーザーガイド、リファレンス デザインおよびサンプル デザインのユーザーガイドのほかに、回路図や UCF ファイルも提供されています。また、AMS ブロックを速やかに評価できるように、VC707 と同様に AMS 評価カードも同梱されています (Leal)。

キットにはさらに、数多くのケーブル / アダプター類 (汎用の 12V 電源アダプタ、2 本の USB ケーブル、1 本のイーサネット ケーブル、1 個の HDMI-to-DVI アダプター) も同梱されています。

デバイス コンフィギュレーション、クロッキング、制御、消費電力など、このボードに関する詳細は、Kintex-7 FPGA KC705 評価キットの専用ページ (http://japan.xilinx.com/products/boards-and-kits/EK-K7-KC705-G.htm) をご覧ください。

Kintex-7 FPGA DSP キット

以上のベース プラットフォーム 2 製品に加えて、Avnet 社から Xilinx Kintex-7 FPGA DSP キット ターゲット デザイン プラットフォームがリリースされました。これは複数の DSP 処理を FPGA を使用してより高速に実行したいと考えるユーザーを対象としています。Kintex-7 KX325T FPGA を中心として構築されたこの TDP には、アナログ信号とのインターフェイスとして高速アナログ FMC が含まれています (図 4)。

搭載されている高速アナログ ドーターカードは 4DSP 社の FMC150 です。このカードには、内部クロックでも外部供給のサンプル クロックでも動作可能な、2 系統の 14 ビット 250MSPS A/D チャネルと、2 系統の 16 ビット 800MSPS D/A チャネルが搭載されています。

Kintex-7 DSP キットはターゲット DSP リファレンス デザインと共に出荷され、デジタル アップ コンバージョン (DUC) とデジタル ダウン コンバージョン (DDC) の付いた高速アナログ インターフェイスが含まれています。エンドユーザーはこれらをデザインに取り込むことで、開発期間の短縮が週単位で図れます。

デバイス コンフィギュレーション、クロッキング、制御、消費電力など、このボードに関する詳細は、Kintex-7 FPGA DSP キットの専用ページ (http://japan.xilinx.com/products/boards-and-kits/AES-K7DSP-325T-G.htm) をご覧ください。

ザイリンクスは今後数カ月のうちに、7 シリーズ デバイスをサポートする新しいボードをリリースする予定です。現時点でリリースされているボードの詳細は http://japan.xilinx.com/products/boards-and-kits/index.htm をご覧ください。FMC ドーターカードの詳細は http://japan.xilinx.com/products/boards_kits/fmc.htm をご覧ください。

FPGA-based Automotive ECU Design Addresses AUTOSAR and ISO 26262 Standards

AUTOSAR および ISO 26262 規格に対応したオートモーティブ向け ECU を FPGA で設計

リコンフィギュレーション可能なハードウェア テクノロジを活用した車載機能のフレキシブルな統合

Francisco Fons
PhD Candidate
University Rovira i Virgili
Tarragona, Spain
francisco.fons@estudiants.urv.cat

Mariano Fons
PhD Candidate
University Rovira i Virgili
Tarragona, Spain
mariano.fons@estudiants.urv.cat

自動車メーカーは、運転を快適にし、安全性を高め、そして当然ながら競合他社に勝とうと、車内の至るところに採用されている電子制御ユニット（ECU）にさらに多くの先進機能を搭載しようとしています。このような背景の中、AUTOSAR（Automotive Open System Architecture）イニシアチブや国際的な機能安全規格の ISO 26262 が、オートモーティブ ECU デザインの技術とアーキテクチャの基盤として急速に広まりつつあります。

新型モデルに求められるさまざまな機能要件を満足するべく、自動車エレクトロニクスの密度は増大の一途をたどっていますが、これに歩調を合わせるかのように、FPGA ベンダーが提供するデバイスは、アプリケーション全体を集積できるだけの大容量化が進む一方で、旧世代のデバイスに比べて低消費電力かつ低コストが図られています。このようなトレンドは、オートモーティブ業界でリコンフィギュレーション可能なコンピューティング テクノロジの利用が今後進むであろうことを示唆しています。

筆者らは、AUTOSAR および ISO 26262 規格の両方に準拠した ECU を、MCU を使用したプラットフォームではなく、プログラマブル FPGA を利用して設計する新たな手法を考案しました。ここでは、並列性、カスタマイズ性、柔軟性、冗長性、および汎用性など、リコンフィギュレーション可能なハードウェアが備える特徴を最大限に生かすことに主眼を置きました。概念的な設計が完了した際はプロトタイプをインプリメントしたいと考えています。今回の狙いを実現するには、ハードウェアの ARM® デュアルコア Cortex™-A9 MPCore プロセッサと、パーシャル リコンフィギュレーション機能を有する 28nm のザイリンクス 7 シリーズ プログラマブル ロジック デバイスを組み合わせた、ザイリンクス Zynq™-7000 エクステンシブル プロセッシング プラットフォームが最適なソリューション候補の 1 つになると考えています。この FPGA プラットフォームは、要件に合致するだけでなく、車載ネットワークとして一般に使用される CAN やイーサネットなどの通信コントローラーをオンチップでサポートしています。

新たなアプリケーション

自動車の計算処理能力は複数の ECU に分散して存在しており、それぞれが通信ネットワークで接続されています。今後数年の間に、新たな自動車アプリケーションの登場を背景に、こうした計算処理能力はさらに向上すると見込まれています。新しいアプリケーションとしては、セーフティ機能やドライバー アシスタンス機能、自動車対自動車の通信機能、快適性や制御性を高める機能、エンタテインメント、広範な領域にまたがるハイブリッド電気テクノロジなどが挙げられます。自動車におけるエレクトロニクスの役割がさらに高まることは特に驚くべきことではありません。アナリストらは、オートモーティブ アプリケーション分野での半導体市場の年平均成長率（CAGR）は、今後 5 年間で 8% に達すると予測しています。半導体デバイスの中で急速な成長が見込まれるセグメントが、マイクロコントローラー（MCU）と、FPGA に代表されるプログラマブル ロジック デバイスです。

車載機能が豊富かつ高度になる一方で、自動車メーカーは、複雑化するこのようなシステムの設計および管理をより効率的に進める方法を模索してきました。結論から先に述べると、現在のオートモーティブ ECU の開発において、AUTOSAR および ISO 26262 の 2 つの規格が、ハードウェアおよびソフトウェアのアーキテクチャ設計、開発、および実装方法に大きな影響を与えています（囲み記事を参照）。

車両内に分散する ECU の標準システム ソフトウェア アーキテクチャを定義することを目的に、2003 年に自動車メーカー各社によって AUTOSAR が設立されました。一方の ISO 26262 は機能安全に焦点を当てており、基本的には故障の回避や検出を行い、故障の影響を緩和すると共に、確立されたシステムの安全目標を保護することを目的としています。ドライバー アシスタンスや動的制御など安全性に影響を及ぼす新しい機能が登場し、機能安全は自動車開発においてますます重要な課題になっています。なお ISO 26262 規格は、ソフトウェアおよびハードウェアの安全な開発を対象に 2011 年に承認されました。

ECU 全体の設計および開発フローは、これらの業界規格に拘束されることになり、つまりは系統立てた手順が必要です。筆者らのプロジェクトでは、システム アーキテクチャの最適化が図れるリコンフィギュレーション可能なハードウェア テクノロジを利用して、コスト効率の高い組み込みコンピューティング プラットフォームの設計を目標に掲げました。

システム アーキテクチャ

AUTOSAR および ISO 26262 規格は、マイクロコントローラー ユニットを使用したコンピューティング プラットフォームの実現を目指して、主にソフトウェア開発の観点から誕生しました。ここで、ハードウェア / ソフトウェアの協調設計やリコンフィギュレーション可能なコンピューティング手法の導入は、この分野にさまざまなメリットをもたらします。オートモーティブ ECU のハードウェア プラットフォームには標準的な MCU が一般に使用されますが、最近の FPGA はコストも下がると共に、一部の製品はハードコア プロセッサを内蔵できるなど、ECU の分野においても大規模な実装に適した信頼に足るソリューションとして認知され始めています。さらにオートモーティブ分野では新たな機能の統合が継続して進んでおり、このようなトレンドを背景に、並列コンピューティング アーキテクチャの必要性が高まっています。これは特にインフォテインメント分野で顕著となっており、高速なデジタル信号処理が可能な FPGA が普及していくと考えられます。こうした状況を踏まえて、ザイリンクスを始めとするプログラマブル ロジック ベンダーや MathWorks 社のような EDA ツールベンダーは、オートモーティブ市場に焦点を当てた取り組みを強化しています。

リコンフィギュレーション可能なハードウェアのメリットをオートモーティブ アプリケーションで最大限に活用することを目的として、筆者らは、自動車コンピューティング ネットワーク内に存在し、かつ、エンドユーザー機能のインプリメンテーションを担う、重要な ECU の 1 つであるボディ コントロール モジュールを対象とした事例を通じて、考案したテクノロジを解説します。この ECU はボディ ドメイン コントローラーとも呼ばれ、ワイパーやウォッシャーの作動、ライトの点灯、パワー ウィンドウの作動、エンジンの始動と停止、ドアミラーの調整、さらにはドアロックなど、車体のエレクトロニクス機能の統合制御を担っています。ここで筆者らは、セーフティ クリティカルな機能を備えた AUTOSAR 準拠の ECU システムを、FPGA プラットフォーム上に構築することを目標に掲げました。

実世界でのシナリオ

自動車メーカーが複雑化する車両機能をコスト効率の高い方法で管理したいと考えるのであれば、AUTOSAR が推進する ECU システム アーキテクチャの標準化を避けて通ることはできません。AUTOSAR によって、複数の ECU に分散する諸機能の上位レベルでの統合や、ソフトウェア部品の再利用が実現されます。AUTOSAR の最大のゴールは、ソフトウェアからハードウェアを切り離した統一的な ECU アーキテクチャを定義することにあります。すなわち、ハードウェアに依存しないインターフェイスを定義して、ソフトウェアの再利用を促そうという考えです。言い換えると、AUTOSAR 規格に則って記述されたソフトウェア部品は、AUTOSAR 準拠のランタイム環境に適切に統合された場合に、いかなるベンダーのマイクロコントローラー上でも動作することを意味しています。

以上の特徴によって自動車メーカーには多くの自由度がもたらされます。異なるサプライヤーによって開発された同じソフトウェア モジュールの同等バージョンを、プラグアン

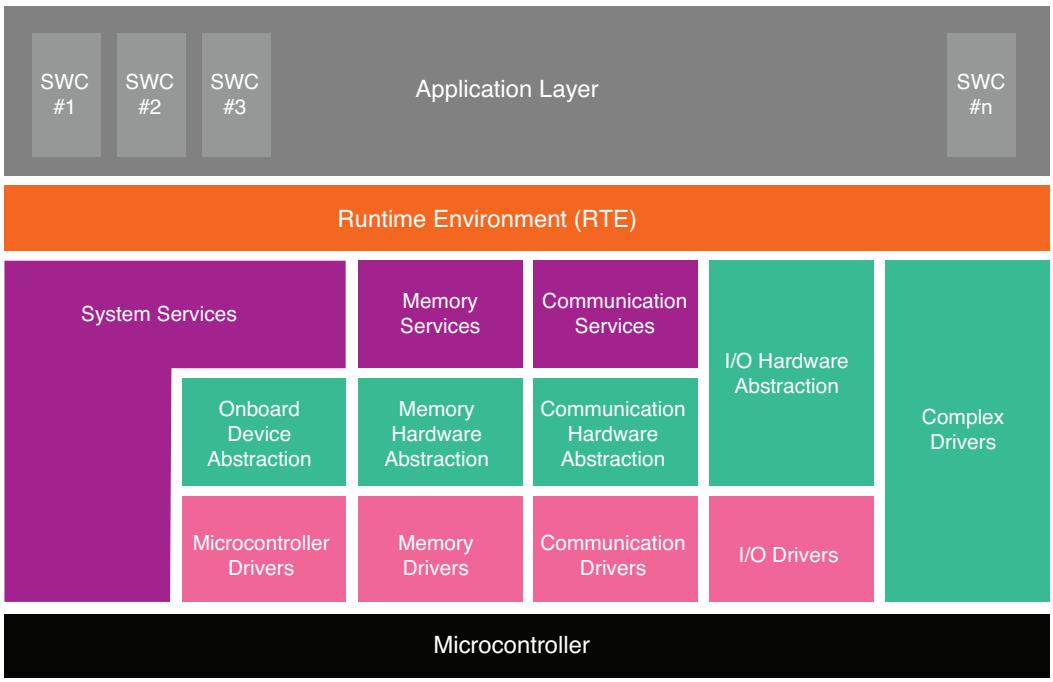


図 1 - MCU 層からアプリケーション層に至る、AUTOSAR のレイヤー モデル

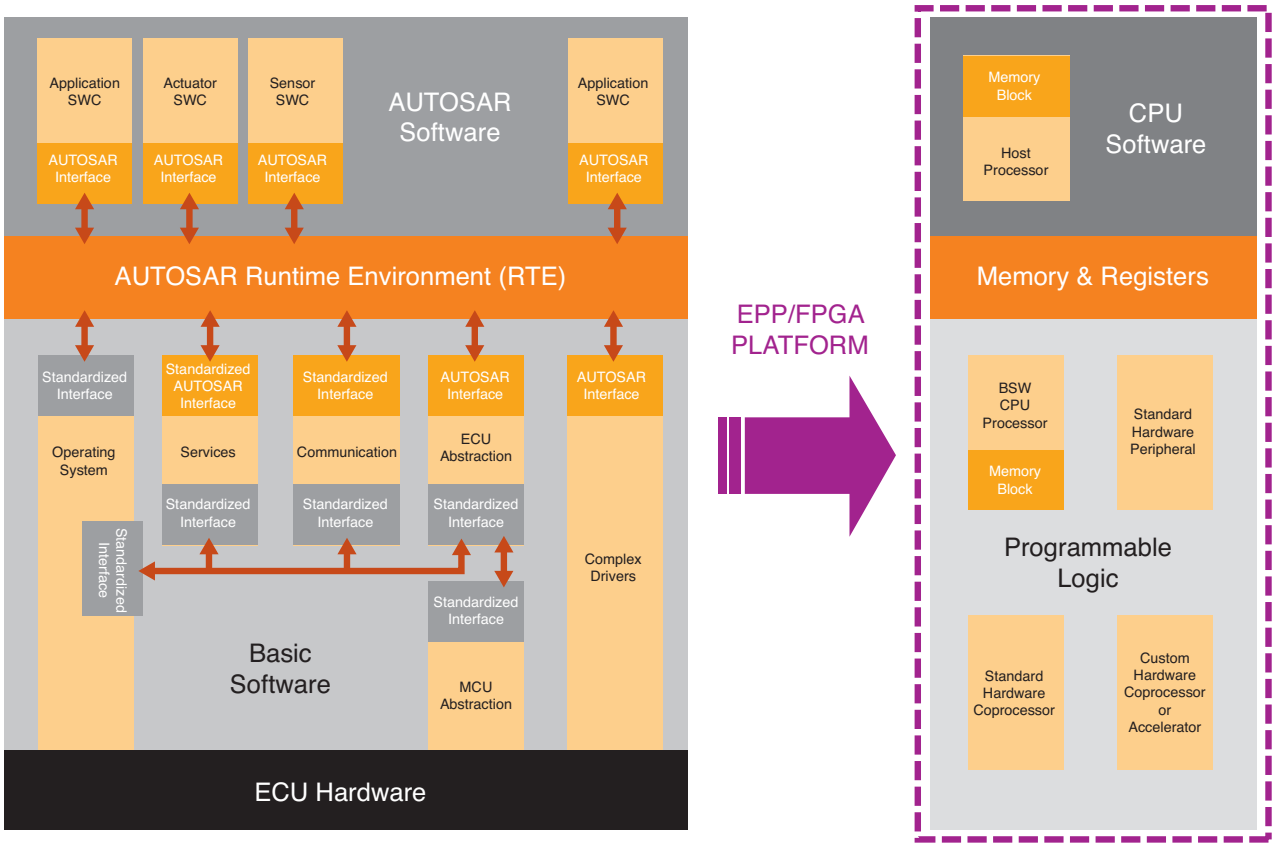


図 2 - AUTOSAR の ECU アーキテクチャを FPGA プラットフォームに適用

ドプレイのおかげで、車両内のほかの機能に影響を与えることなく、車両プラットフォーム上で透過的に入れ替えることができます。結果としてハードウェアとソフトウェアはそれぞれが高度に独立したものとなります。こういった分離は標準ソフトウェア API を介して接続される抽象化層によって実現されます。図 1 に AUTOSAR が定義する機能層を示します。

図 1 で最下層の黒い部分は MCU 自身によって構成されるハードウェア層または物理層で、すなわち、CPU と、その CPU に接続される標準ペリフェラルを表しています。マイクロコントローラーの上側にある基本ソフトウェア (BSW) は、ピンク色のマイクロコントローラー抽象化層 (MCAL)、緑色の ECU 抽象化層 (ECUAL) と複合ドライバー、および紫色のサービス層 (SRV) の 3 種類の層で構成されます。これらはいくつかの列あるいはスタックに分類されます (メモリ、通信、入力 / 出力など)。

ハードウェア部品に最も近い層がマイクロコントローラー抽象化層です。名前が示すとおり、この層によって MCU が抽象化されます。この目的は、マイクロコントローラー内の ハードウェア ペリフェラルを扱うハードウェアに依存しない API を実現することです。次に ECU 抽象化層によって、MCU と接続される ECU ボード上のその他のスマート デバイスが抽象化されます (たとえば、システム電圧レギュレータ、スマート スイッチング コントローラー、コンフィギュレーション可能なコミュニケーション トランシーバーなど)。続く 3 番目の層はサービス層です。サービス層はハードウェアにほぼ依存せず、必要な複数のバックグラウンドサービスを扱うのがその役割です。たとえば、ネットワーク サービス、不揮発性メモリの処理、システム ウォッチドッグの管理などが該当します。AUTOSAR ではこれら 3 種類の層を用いて基本的な一連のソフトウェア機能が定義され、オートモーティブ ECU に対して抽象化した上位レベル機能のすべてが特定のハードウェア プラットフォーム下で実現します。

4 番目のランタイム環境 (RTE) はアプリケーション ソフトウェアにコミュニケーション サービスを提供する層です。この層は、BSW の上位層と、アプリケーション層 (APP) の両方からアクセス可能な、一連の信号 (センダー / レシーバー ポート) およびサービス (クライアント / サーバー ポート) から構成されています。RTE は、アプリケーションを基本ソフトウェアから抽象化すると共に、ソフトウェア スタック型アーキテクチャの境界を決定し、汎用かつ交換可能なソフトウェア コード (APP) と個別かつハードウェア依存のコード (BSW) とを分離する役割を担います。言い換えると RTE は、ソフトウェア アプリケーションをハードウェア プラットフォームから分離する仕組みで、結果的に RTE よりも上位で動作するすべてのソフトウェア モジュールはプラットフォームに依存しなくなります。

RTE よりも上位ではソフトウェア アーキテクチャが変わり、レイヤー構造ではなく、アプリケーション層を介したコンポーネント形式になります。この機能はソフトウェア コンポーネント (SWC) のカプセル化が主な目的です。したがって、異なる車両プラットフォームの複数の ECU 間で機能のスケーラビリティと可搬性を実現するには、

AUTOSAR ソフトウェア コンポーネントのインターフェイスの標準化が鍵になります。AUTOSAR 規格によって、複合ドライバーを除くこれら 3 つのモジュールの API と機能が明確に規定されます。SWC は、ランタイム環境を通してのみ、ほかのモジュール (ECU 内および ECU 間) と通信を行います。

ECU に統合される機能が増えるにつれて、シングルコア MCU またはマルチコア分散される諸機能を APP 層を介して処理するために、ECU プラットフォームには CPU またはホスト プロセッサが必要です。MCU 層と基本ソフトウェア層の一部は、プログラマブル ロジック ファブリック内に合成してもかまいません。つまり、CPU に接続される標準ペリフェラルに加えて、カスタム ペリフェラルやコプロセッサもハードウェア内に共存可能であり、ソフトウェアを用いてすべてまたは一部を管理します。

MCU ベースの AUTOSAR ECU アーキテクチャを、層ごとのシステム パーティショニングを維持しながら、エクステンシブル プロセッシング プラットフォーム (EPP) または FPGA デバイスに移植する案を図 2 に示します。RTE 層の下には、オペレーティング システム (OS)、メモリ スタック、コミュニケーション スタック、I/O スタックなどが並びます。RTE 層の上にはアプリケーションがインプリメントされると共に、

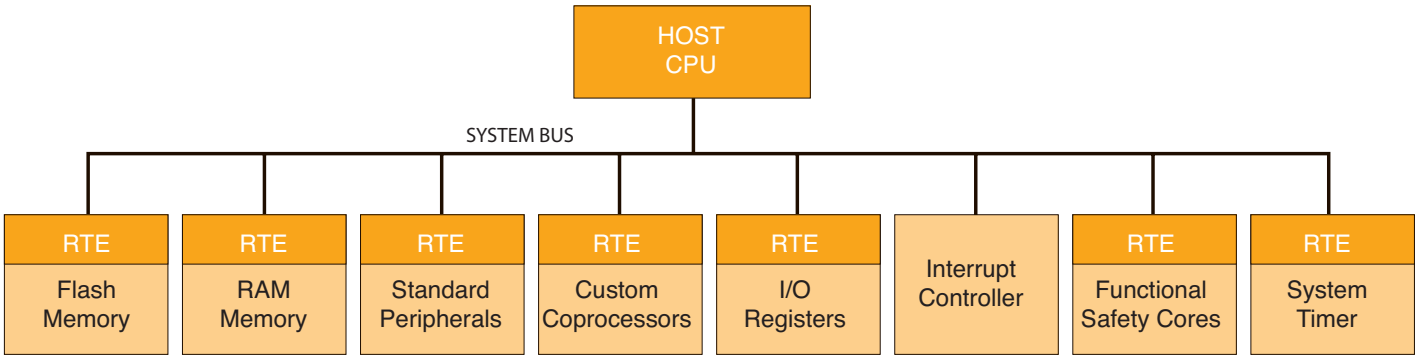


図 3 - FPGA に実装したオートモーティブ ECU のブロック図

MCU の置き換えとして、FPGA は最適な選択肢の 1 つに挙がります。プログラマブル ロジックを使用して ECU をインプリメントしたときに得られるメリットについては、AUTOSAR の各層の概要がヒントを握っています。ここからは、専用のスタティック ハードウェア (フラッシュ ベースまたは SRAM ベースの FPGA テクノロジ) を使用した筆者らのソリューションのインプリメンテーション方法と、続いて、この方法をランタイム リコンフィギュレーション可能なハードウェア インプリメンテーション (SRAM ベースのパーシャル リコンフィギュレーション可能 FPGA) へと拡張した場合について考察していきましょう。

FPGA ベースのスタティック ハードウェアを用いた ECU デザイン

CPU、メモリ、およびプログラマブル ロジックで構成される一般の組み込みシステムは、AUTOSAR のアーキテクチャに適合します。アプリケーションの管理、および複数のソフトウェア コンポーネントに

専用のコプロセッサまたはコア プロセッサには、元々のハードウェアとは独立して追加機能を自由にインプリメントできるため、その搭載は機能安全の視点からも適切であり、システムのデザインに高い柔軟性と必要に応じた冗長性をもたらします。また、中間の RTE 層は、FPGA に分散する RAM ブロック内、またはデバイスのロジック セルに埋め込まれているフリップフロップ内、あるいは外部メモリ内に合成することができます。さらに、RTE 信号インターフェイスを、シングルポート メモリを介して読み出しおよび書き込みの両動作を許可するように簡単に設計できます。または、干渉に対する保護手段として、AUTOSAR が定義するセンダー ソフトウェア ポートおよびレシーバー ソフトウェア ポートの相手方となるように、2 個の独立した読み出しポートと書き込みポートを備える単一のデュアルポート メモリで読み出しまたは書き込みのいずれかのトランザクションにアーキテクチャを制限するように設計することも難しくありません。

AUTOSAR インターフェイスを介して RTE と通信するソフトウェア コンポーネントが配置されます。

AUTOSAR アーキテクチャはそもそも複雑で、その実現には、強力な組み込みコンピューティング プラットフォームが必要です。最近の ECU は MCU プラットフォーム上の 32 ビット シングル コア プロセッサに構築されることが一般的ですが、求められるすべての演算能力を提供するにはシングル コアでは不十分になりつつあります。とはいえマルチコア CPU を使用したとしても、アービトレーション制御を用いたマルチプロセッサ バスを介して共有のプログラム / データ メモリにアクセスした場合に性能低下を招く可能性もあり、性能を引き出すにはときにはきわめて複雑なソリューションが必要になります。

代わりに、筆者らは、プログラマブル ロジックとホスト CPU 役のシングルコア プロセッサを中核に、高度なペリフェラル、コア プロセッサ、あるいはスレーブ プロセッサを周囲に配置する新たな構成を考案

しました。これらのコンピューティング ユニットはすべて、FPGA 内の専用 RAM ブロックから専用コードを読み出して動作するザイリンクスの PicoBlaze™ プロセッサまたは MicroBlaze™ プロセッサのような新しいソフトコア プロセッサ (専用プログラム メモリで動作するソフトコア プロセッサとは独立) として、あるいは、オーダーメイドのハードウェア アクセラレータとして、FPGA 上にインスタンシエートできます。いずれも単一の CPU と、CPU の一部のタスクをオフロードする高性能なペリフェラルで構成された単純なトポロジであり、システムの複雑性は高くありません。このような構成によって、ホスト CPU が APP 層全体をソフトウェアで管理しながら、専用ペリフェラルは互いに独立して並列かつ自律的に BSW 層を処理することができます。さらに、ペリフェラルの割り込み頻度を減らし、割り込みサービス ルーチンによる CPU の介在を抑えてホスト CPU のソフトウェア実行がより一定となるように、これらペリフェラルを設計する方法も採れます。図 3 に、システムのブロック図、および機能ユニットに分割して FPGA デバイス上に合成したコンポーネントを示します。

この方法は、マルチプロセッサ プラットフォームと同等のシステム性能が得られる一方で、(ソフトウェア開発およびメンテナンスの観点で) 複雑性はシングルコア プロセッサと同等レベルです。その代わりに、強力な専用ハードウェアや、ホスト プロセッサとは並列に動作する自律的な専用コプロセッサが必要です。

概念的にアイデアを単純化するために、このシステムのアーキテクチャを、RTE インターフェイスを基準に上位層と下位層の 2 つに分割します。上位層は AUTOSAR のアプリケーション層に相当し、車両のエンドユーザー機能を管理するソフトウェア コンポーネント群で構成されます。下位層はハードウェアと RTE リンクまでの基本ソフトウェアで構成されます。アプリケーション層は相対値として車両内の上位レベル機能の約 90% を担当し、RTE よりも上にあるこれらのソース コードはすべて再利用可能です。

一方の下位層は、柔軟性と汎用性を上位層に与える諸機能で構成されています。つまり、下位層は、再利用可能な各機能を特定ハードウェア プラットフォーム下で使用できるように、それぞれの機能のカスタマイズを担当します。この構成によって、上位層は基本的に、車両荷重、センサー、アクチュエーターなどの制御アルゴリズムを有限ステート マシン (FSM) にインプリメントした一連のソフトウェア機能として捉えることができます。これらのアルゴリズムは、OS が制御するソフトウェア タスク内でスケジューリングされ、CPU によって循環的に実行されます。

上位層に適した抽象化を行うために、下位層も、A/D コンバーター、PWM コントローラー、タイマー、メモリ コントローラーなど、CPU に接続されるすべての標準ペリフェラルのドライバーをインプリメントする役割を担います。リアルタイム処理が求められるイベントの管理も下位層が担当します。この点に関してプログラマブル ロジックにはいくつかのメリットがあります。一般にハードウェアに起因する外部イベントを用いるのではなく、RTE 信号を周期的に読み出しまたは書き込んで FSM を適宜動作させ、ソフトウェア機能の単純なシーケンスとしてアプリケーションを処理できるようなホスト CPU を実現しようというのがこの狙いです。下位層は、これらハードウェア イベントを隠蔽および管理し、前処理を行って、具体的なタスク スケジューリングに従って RTE の特定信号を更新するか、リアルタイムでなんらかの動作を実行することになるでしょう。

システム CPU に接続する専用ハードウェア コントローラーの場合、そのようなコントローラーが自律的に動作できるのであれば、必要な共有リソースは最小限に抑えられます。OS の視点から見ると、アービトレーション、レイテンシ、リトライ手段などが必要ないため、こういった構成はシステムの複雑性を緩和する働きがあります。

その他のメリットとしては、ハードウェアの方がソフトウェアよりも並列性に対して親和性が高いことから、マルチ スレッディングによってソフトウェアで実行される一部の機能を、専用ハードウェアであればより簡単にインプリメントできる点が挙げられます。より自由度の高いハードウェアを使用して、アルゴリズム内の計算集約型の部分をハードワイヤし、並列構造またはパイプライン構造のハードウェアとしてインプリメントすれば、ノイマン型の逐次実行のソフトウェア処理に比べて実行時間の短縮が図れます。

MCU 内および BSW 層内に合成されるペリフェラルやハードウェア コプロセッサに充分なレベルのインテリジェンスを与えることで、ECU ソフトウェアの複雑性が軽減され、CPU 処理が解放されます。

ECU プラットフォームが複雑化するのと並行して、システムが必要とする I/O 信号線の本数も増加しています。この点に関しては、ユーザー ピンが一般に多い FPGA は、マイクロコントローラーに比べて明らかなメリットがあります。MCU ベースの ECU では、デジタル シフト レジスタやアナログ マルチプレクサーなどのパラレル シリアル変換デバイスを外付けして、MCU の入出力を拡張しなければなりません。FPGA ではこのような周辺部品は一切不要であり、部品点数の削減とプリント基板の省スペース化が図れます。

最先端の FPGA デバイスには A/D コンバーターが既に内蔵されています。ECU の多くは必要な機能の一部を実現するためにアナログ信号 (たとえばバッテリ電圧) を使用するため、FPGA のこの機能はオートモーティブのデザインにおいてとても有益です。プログラマブル ロジック デバイス内に A/D コンバーターが統合されたことで、FPGA の新たな応用が見込まれます。

MCU と同様に FPGA はリモート更新機能を備えています。ただし、この場合、FPGA にダウンロードされるビット ストリームは、ソフトウェア コードだけではなく、ハードウェア回路も対象になるという点が重要です。つまり、製品が量産に入った段階でも、システム アップデートまたはアップグレードによってハードウェア回路を変更することも不可能ではありません。製品出荷後にバグ修正 (この場合はハードウェアとソフトウェアの両方) も可能なことから、このような柔軟性は自動車業界からも歓迎されています。

ISO 26262 で認められる安全関連の機能を組み込んだ ECU を構成するハードウェアとソフトウェアは、そのカテゴリに基づいて、なんらかの保護機能を備えなければなりません。ソフトウェアの視点からは、外部の干渉を受けないことを確実にする必要があります。すなわち、ECU で実行される非安全関連コードが、同じ ECU 内で実行される安全関連として分類されたコードの動作を損なうことがあってはなりません。同じプロセッサ上で安全関連機能と非安全関連機能を実行するには、こういった適切な分離が不可欠です。多くの場合、この要件は、MCU よりもプログラマブル ロジッ

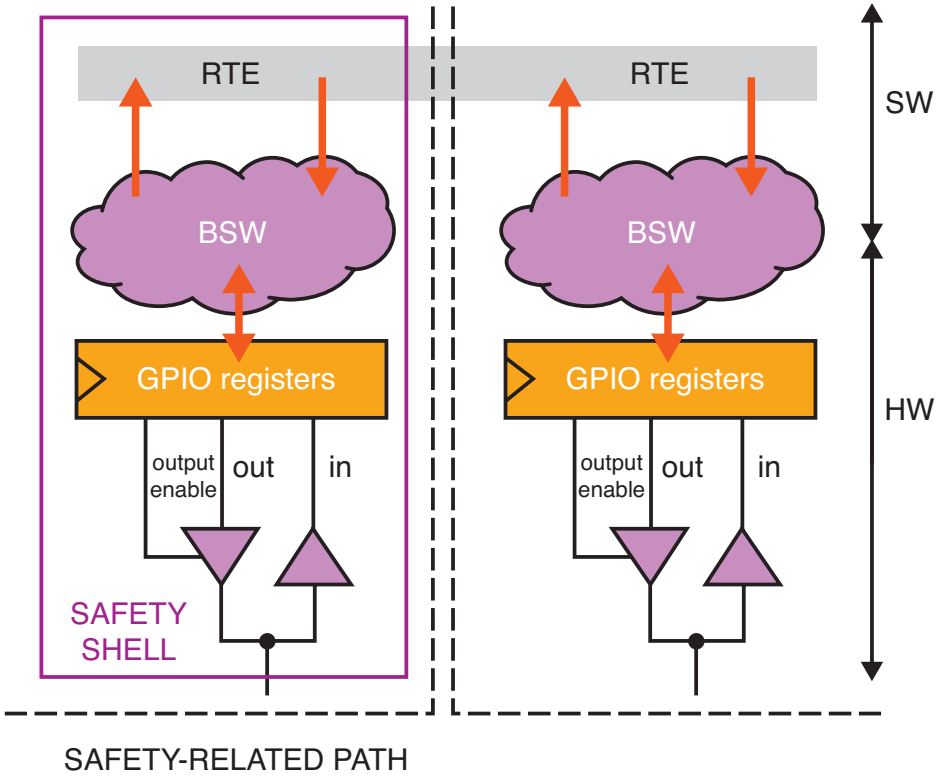


図 4 - 安全関連ポートから非安全関連ポートを分離して影響を排除した
セーフティ シェル アーキテクチャでのハードウェア / ソフトウェアの協調設計

クの方が柔軟に対応できます。

機能安全を目的としたメモリ保護ストラテジでは、権限が与えられた安全ソフトウェア コンポーネントのみが安全関連信号へ書き込みアクセスできるようにする必要があります。MCU デバイスにおいてメモリ パーティショニングは、ソフトウェア アプリケーションを互いに分離してそれぞれのデータ破壊を防ぐための、障害を隔離する手法の 1 つです。プログラマブル ロジックを使用すれば自己保護メカニズムをより効率的にインプリメントできると考えられます。専用の単一デュアルポート メモリを介して、データを書き込みポートに書き込み、読み出しポートから読み出されるようにして、安全信号に関連する RTE バッファーを管理できます。この方法では、ソフトウェア側からそれらの信号の書き込みまたは読み出しに個別の制限を与える専用ハードウェア コントローラーをインプリメントすることもできます。レジスタを使用することでも同じ手法が実現できます。

ECU システムに安全関連機能などの専用ハードウェア ソリューションを導入する場合、FPGA を使用すれば大きなメリットが得られます。たとえば I/O ピンと GPIO コントローラーについては、安全機能に関連するピンは ECU 内部の安全コンポーネントのみが排他的にアクセスできるようにし、デバイスのほかのピンとは分けて配置し、専用の I/O ポートとしてグループ化しておくことができます。デザインによって安全関連ピンと非安全関連ピンをシステム内で分離して干渉を確実に防止できる、優れた方法といえるでしょう。安全関連ピンの状態は安全関連コードでのみ管理され、非安全関連ピンへのアクセスによって損なわれることはありません。これを図 4 に示します。

さらに、GPIO が含まれるアプリケーションやソフトウェア コンポーネントの要件に合わせて各 GPIO ポートのサイズを決定できるため、複数のアプリケーション間で共有される物理リソースに GPIO ポートを変換するという MCU ポートでは必要となる手順を省略できます。このように FPGA では、システム メモリ マップ内の特定レジスタに割り当てられた特定のポートが、個々の SWC で管理される各アプリケーション (たとえば、パワー ウィンドウ、ワイパー、ドア ミラーなど) に与えられます。MCU プラットフォームでは、ポートは固定幅 (一般に、8 ビット、16 ビット、または 32 ビット) で、ビット単位ではなくワード単位でアドレス指定されるため、こういった割り当てはできません。上述の制御レジスタは、プログラムの実行に伴って複数の SWC がアクセスする共有リソースとして使用されます。

筆者らは、GPIO コントローラーに用いたのと同じストラテジをほかの標準ペリフェラルにも拡張して使用しました。AUTOSAR が SWC を使用して上位層で推進する機能のパーティショニングおよび分離を、プログラマブル ハードウェアによって下位層のリソースにも拡張しようという考えです。このようなテクニックは、

標準の MCU デバイスを使用した固定的なハードウェア ソリューションでは実現できません。

先に述べた MCU の標準ペリフェラルに対して行ったのと同様の分離ストラテジは、安全機能のすべてのチャネルまたはデータパスにも適用可能です。このような分離は高位に分類される安全機能 (ISO 26262 の ASIL レベル (囲み記事を参照) による分類) を実装する方法として特に有効です。ここでは、重複して実行されるようにこれらの各パーツを低位 ASIL レベルの冗長なパーティションに分解し、新たな ASIL レベルに則って実装されるようにします。冗長性を活用したこのような安全ストラテジはプログラマブル ロジックを選択する理由の 1 つであり、これにより、同じデバイス内にそれぞれ独立した同一の処理エンジンを複数インスタンシエートすることが可能になります。さらに、抽象化ソフトウェアによる方法に比べて、特に干渉の排除といった機能などは、アーキテクチャ的なアプローチ (ハードウェア) を使用する方が、特定の ASIL レベルへより明確かつ簡単に対応できます。というのも、スタック オーバーフローや C プログラミング言語でのデータ ポインターの不正な扱いといった設計上の欠陥は、システムに予期せぬ安全上の問題を引き起こす可能性があるからです。

機能安全にプログラマブル ロジックが持つ柔軟性を活用すると、デザイン上のメリットがもたらされてトリプル モジュラー冗長性 (TMR) 方式をインプリメントできるようになります。TMR は、航空宇宙アプリケーションにおいて、シングル イベント アップセット (SEU) を低減する対策として知られています。TMR は、3 系統の同一ロジック回路それぞれに同じ処理を並列実行させて、各出力を多数決回路で比較し、結果を得る仕組みです。このような機能をインプリメントするにはハードウェアが効率的です。

また、コストと消費電力が課題となるマーケットを対象にする場合は、ザイリンクス Zynq-7000 EPP のようなプログラマブル ロジック デバイスを使用すると、システム全体の消費電力を削減する機能を利用できます。このような機能の中には、MCU デバイスから継承したものもあります。処理システムのパワーオン オンリー モード、スリープ モード、ペリフェラルに依存しないクロック ドメインなどの機能によって、デバイスのアイドル期間中のダイナミック消費電力を大幅に抑えることができます。

プログラマブル ロジック デバイスにハードコア プロセッサを搭載した場合は、通常 MCU ベースのプラットフォームで開発する場合と同じく、まずシステム全体の機能をソフトウェアとして開発し、デザインにハードウェアを順次追加して、その一部をプログラマブル ロジック リソースに移植するといった設計手法が取られます。こういった手法には、純粋にソフトウェアでインプリメントする方法に比べて、ソリューションを複数のバージョンに分けて構築したり、カスタム ハードウェア内に機能の一部を統合できるなどのメリットがあります。

ランタイム リコンフィギュレーション可能なハードウェア上での ECU デザイン

プログラマブル ロジックを用いてスタティック ハードウェアおよびソフトウェアで ECU をインプリメントするメリットを一通り説明してきました。次に、ランタイム パーシャル リコンフィギュレーション (PR) 機能を備えた SRAM ベースの FPGA を使用したデザインに移ります。PR テクノロジによってオートモーティブ設計者にはさらなるメリットがもたらされます。

大きなメリットの 1 つとして、起動時にコンフィギュレーションする必要のない PR 領域が FPGA 内に存在すると、ECU のウェークアップ時やパワーアップ時などに要するシステムの起動時間を短縮できます。有益なパーシャル リコンフィギュレーションをサポートしていない FPGA は、パワーアップ時にすべての FPGA リソースをコンフィギュレーションしなければなりません。一方で、ランタイム リコンフィギュレーション可能な FPGA であれば、パーシャル ビットストリームをダウンロードして一部のみをリコンフィギュレーションすることが可能です。

既存の最先端の大容量 FPGA デバイスは、パワーアップ時にビットストリームすべてをダウンロードしなければならないため、コンフィギュレーション時間が非常に長くかかります。ランタイム パーシャル リコンフィギュレーションを使用すれば、このようなコンフィギュレーションに要するレイテンシを大幅に短縮できます。たとえば、パワーアップ時に最小限のサブシステム (ブートローダーとシステム アプリケーションがすぐに必要とする部分) のみをコンフィギュレーションし、システムの残りの部分は初期化が必要になるまでアイドルのまま放置しておくような使い方も可能です。パワーアップまたはウェークアップに対してシステムが速やかに応答しなければならないときに、スタートアップ時の動作を 2 つのフェーズに分割することで、初期化処理を短縮しようという狙いです。そのような制御を行うには、システム アーキテクチャを、スタティック領域と、1 つまたは複数のリコンフィギュレーション可能領域 (PRR) とに分割します。スタティック領域には、一般にホスト CPU に代表されるスタートアップ処理を担うシステム部分と、リコンフィギュレーション エンジン、およびビットストリーム リポジトリへのデータ リンクなどを統合しておきます。残りの領域のロジックは、アプリケーションが必要としたときにダウンロードされる、個々のパーシャル ビットストリームによって記述します。

さらに、PRR 領域を無効化すると、使用しない領域の大きさに比例してデバイスの消費電力を節減できます。ECU は自動車のバッテリで動作するため、省電力モードは特に重要です。そのため、オートモーティブ用 MCU には、車両が動作していないとき (つまり、スリープ モードのとき) に ECU の消費電力をできるだけ抑えるローパワー モードが装備されています。同じような考え方で、必要のないときに FPGA の一部に空のビットストリームを与えて無効化すると、ロジックの動作が少なくなり、結果としてダイナミック消費電力が抑えられます。

また、航空宇宙アプリケーションにおけるランタイム リコンフィギュレーション可能なロジックの使用を、オートモーティブ分野に適用する方法もあります。SEU あるいは電磁妨害 (EMI) に起因して SRAM リソースに障害が発生したときでも、コンフィギュレーションの一掃によってシステムを復帰させようというテクニックです。ハードウェア ペリフェラルを周期的にリコンフィギュレーションすることで、異常動作が発生した場合でも、システムの自己修復が保証されます。しかも、異常状態のまま動作する時間は一掃する周期よりも長くなることはありません。この着想は、たとえば MCU ペリフェラルを周期的にリコンフィギュレーションす

るなど、干渉に対する保護手段としてソフトウェアにも適用されています。

ランタイム パーシャル リコンフィギュレーション テクノロジの柔軟性によってもたらされるもう 1 つのメリットとして、FPGA リソースの特定箇所に恒久的または修復不能な回路障害が発生し、ロジックセルまたは RAM ブロックが影響を受けたときに、機能の再配置を行って障害を回避できるという点が挙げられます。ハードウェアまたはソフトウェアの障害を検出したとき、必要な機能を自動的に同じ ECU にあるプログラマブル ロジック デバイスの別の部分に再配置するか、または、ほかの ECU に再配置することが可能かもしれません。しかし、実現可能な考え方ではあるものの、このような機能は自動化ツールでは今のところサポートされていません。

ランタイム リコンフィギュレーション可能なコンピューティング テクノロジをオートモーティブ分野に適用したときの最大のメリットは、動作中の共有ハードウェア リソース上で複数の機能を時分割で実行できる点です。双方のアプリケーションを互いに排他的に動作させる場合に (たとえば、車両が前方に走行しているときは車線逸脱警報機能を動作させる一方で、バックしているときはリアカメラ ビューまたは駐車支援アプリケーションを動作させる)、ECU のコンピューティング リソースで処理する機能アプリケーションを時分割で動作させるという発想は、組み込みシステムのコスト低減と複雑さ緩和に加えて ECU の小型化や軽量化にもつながると考えられます。

このアイデアの仕組みは、環境あるいは外部条件の変化に対応して特定のアルゴリズムを自己適応するのと同じです。たとえば、エンジン制御アルゴリズムをパーシャル リコンフィギュレーションによって自己適応可能にしておけば、温度条件やバッテリ電圧にかかわらず、当該ハードウェア ブロッ

2 つの重要な規格

オートモーティブ業界では、組み込みシステムの複雑性をソフトウェアおよびハードウェア アーキテクチャの適切な組み合わせによって解決する AUTOSAR と、機能安全を規定する ISO 26262 (策定中) の 2 種類の重要な規格を念頭に置いて、カーエレクトロニクスの設計を進めなければなりません。ISO 26262 の適用や AUTOSAR の導入によって、ランタイムでのハードウェア障害、アプリケーション実行でのタイミングや論理順序の異常、およびデータ破壊などの安全上の問題の検出と対処などが、技術的な課題として浮上します。

AUTOSAR の概要

近年では、自動車に採用されている機械システムや油圧システムは、電子部品へと置き換えが進んでいます。制御機能、モニタリング機能、診断機能などがソフトウェアとして追加されるにつれて、このような置き換えはさらに進むでしょう。実際に機械的ソリューションあるいは油圧ソリューションのみを使用して設計したのでは実現が難しく高コストとなる新機能も、エレクトロニクス テクノロジであれば簡単に実現できます。この場合、エレクトロニクス部品は、不正な状態や障害を防止するために、厳しい安全要件に適合しなければなりません。

ソフトウェア関連製品のエラーはこれまでは比較的まれでしたが、量産製品の一部としてソフトウェアの使用が増大した結果システムはより複雑になり、製品開発期間の短縮もあいまって、今後は製品での障害につながる可能性もあります。オートモーティブ業界はこのような課題に対して業界団体を結成し、アプリケーションの動作を保証すると共に、安全で信頼性の高いソフトウェアの開発を保証する規格の制定に努めてきました。

たとえば MISRA (Motor Industry Software Reliability Association) は、フォード、ジャガー ランドローバーなどの自動車メーカー各社を始め、部品サプライヤーや設計会社も交えたコンソーシアムです。MISRA はソフトウェアのプログラミングに関して一連のルールを規定し、自動車やほかの組み込みシステムにおける安全関連のエレクトロニクス システムの開発にベスト プラクティスの適用を図ることを目的としています。

一方、AUTOSAR (Automotive Open System Architecture) は、自動車メーカーとサプライヤーに加えて、エレクトロニクス業界、半導体業界、およびソフトウェア業界のさまざまな企業によって結成された業界団体です。複数の主要な課題を対象に、オートモーティブの電気 / 電子 (E/E) アーキテクチャに関してオープンな業界標準規格を開発しています。対象としている課題には、高機能化に伴い複雑性を増すオートモーティブ電気 / 電子 (E/E) システムの管理、製品の変更やアップグレードを実現する柔軟性の向上、製品ライン内または製品ライン間でのソリューション スケーラビリティの向上、E/E システムの品質と信頼性の改善、および設計初期段階での設計ミスの検出などが挙げられます。

AUTOSAR イニシアチブが試みる挑戦は、増大するソフトウェアおよびエレクトロニクス テクノロジを、サプライヤーの巨大なエコシステムの垣根を越えて統合しようという、きわめて難しい問題です。ソフトウェアやハードウェアの選択肢の交換あるいはアップデートを単純化することによって、自動車内の複雑化する E/E システムを信頼性の高い方法で管理する基盤を作り、品質を犠牲にすることなくコスト効率の向上を図ろうというのが、AUTOSAR のアプローチです。

2003 年に設立された AUTOSAR イニシアチブは、以前の OSEK/VDX コンソーシアム (10 年前に設立され、ドイツとフランスの一部のメーカーが主導) の自然な発展形ともいえますが、より意欲的な目標が定められていると共に、世界中の大半の自動車 OEM に広がっています。

AUTOSAR のコアパートナーには、BWM グループ、ボッシュ、コンチネンタル、ダイムラー、フォード、ゼネラルモーターズ、PSA プジョー シトロエン、トヨタ、およびフォルクスワーゲン グループが名を連ねています。これらの企業に加えて、パートナー

クは常に最適な状態で動作するでしょう。これと同様の考え方は、たとえばセキュリティ レベルをパラメーターによってランタイムで適合できる暗号化コントローラーの設計など、通信システムにも適用できます。あるいは、ノイズの多い環境で、通信チャネル上のデータ伝送エラーを検出して訂正すると共に、検知した SN 比に基づいてハードウェア アーキテクチャを動的に適応させる、ECC エンコーダー / デコーダー IP などが挙げられます。

ザイリンクス Virtex®-4 FPGA を用い、1 つのスタティック領域と 1 つの PR 領域で構成した ECU システムの例を図 5 に示します。スタティック領域にはソフトコアの MicroBlaze プロセッサと ICAP ベースのリコンフィギュレーション コントローラーをインプリメントし、一方で PRR は、複数の機能タスクあるいはアプリケーションが時間ごとに入れ換わる共有リソースの役割を担います。

このような概念をさらに突き詰めると、生産ラインでコンフィギュレーションを行い、車両内で特定の機能向けにカスタマイズする、万能のオートモーティブ ECU プラットフォームの構想が生まれます。このアイデアは、リコンフィギュレーション可能なハードウェアを使用すれば技術的に実現は不可能でなく、量産工場でのロジスティクスの簡略化と在庫の抑制が図れると考えられます。その理由は、量産ラインで組み立てられるモジュールはハードウェアの観点からは統一されたプラットフォーム デザインまたは製品アーキテクチャ (フレキシブル ハードウェアを採用) で構築されていて、すべての車両に同じモジュールを水平展開できるからです。ダウンロードされるビットストリームによってのみ、それぞれの ECU に個別の機能が与えられます。

高機能型 ECU

オートモーティブ分野における革新技術

シップの成功に向けて 160 社余りが重要な役割を担っています。具体的には「標準化で協調し、実装で競争する (cooperate on standards, compete on implementation)」をスローガンに、各自動車メーカーとサプライヤーは、オートモーティブ E/E デザインのブレークスルーを実現するオープンかつ標準化されたシステム アーキテクチャの定義を共同で進めています (http://www.autosar.org)。

機能安全

International Electrotechnical Commission (国際電気標準会議) による IEC 61508 は、さまざまな分野の安全関連システムのアプリケーションを対象に、電気、電子、およびプログラマブル電子システムと部品の機能安全を定めた一般規格として、2004 年に発効され世界的に採用されています。

安全に関するさまざまな規格の中で、オートモーティブ分野の機能安全を対象としているのが、International Organization for Standardization (国際標準化機構) が定めた ISO 26262 です。本稿執筆時点においてこの規格はまだ策定中ですが、2012 年中には発効されると見込まれています。ISO 26262 は、オートモーティブ業界において、安全製品の設計活動すべてを対象かつ促進することを目的としています。構想段階、製品開発、量産、および実使用下での安全が対象です。

E/E システムの異常動作から生じる危険要因による許容不能なリスクをなくすことを目指した機能安全は、オートモーティブの設計において重要な要件の 1 つです。最近公開されたこの規格は、オートモーティブ業界の事例に即しながら、致命的な障害の防止を目的として、許容可能なリスクを規定しています。その中で「リスク」とは、危害または損傷が発生する可能性とその重大度を組み合わせたものとして定義されています。この規格は設計者に、エンジニアリング開発段階で考えられるすべての危険およびリスクを前もって評価し、適切な手段を用いてそれらを最小限に抑えることを求めています。ISO 26262 にはこれらのリスクを回避するために要件とプロセスを定めたガイドラインが用意されています。

ISO 26262 の考え方に従えば、車両内の機能は安全関連と非安全関連に分類されます。安全関連機能とは、異常が起こった場合にドライバーにリスクが及ぶ機能すべてを指します。安全関連と見なされるそれらの機能に対して、この規格は、考えられるリスク レベルを複数定めています。つまり、具体的な安全目標を維持するという観点で見たときに、一部の機能はほかの機能よりもより重要であることを意味します。

ISO 26262 は、発生しうる事故の重大度、特定の運転条件に遭遇する確率、および外部手段によるリスクの軽減を指標化した ASIL (Automotive-Safety Integrity Levels) を定義しています。ASIL レベルには D、C、B、A の 4 つのレベルがあり、D が最も厳しく、A が最も緩やかです。それぞれの ASIL には、許容不能なリスクを許容可能な残存リスクへと抑えるために、自動車メーカーとサプライヤーが満たさなければならない要件と推奨事項が定められています。

1 つの例を挙げます。自動車の走行中にステアリング コラムがロックされた場合、ドライバーは操舵ができなくなり、事故を引き起こす可能性が生じます。このようなリスクを許容可能なレベルに抑えるには、目標とする安全性を ISO 26262 規格と ASIL レベルに照らして、特定の安全設計基準に則ったステアリング コラムの制御機能を実装しなければなりません。

それぞれの安全目標レベルに応じて、ハードウェア設計者とソフトウェア開発者の双方は、関連する機能の実装において具体的な安全手段を考慮する必要があります。高位の ASIL レベル (D または C) に対しては、充分な非依存要素によって潜在的に低位の ASIL レベルが実現されるように、安全要件を冗長性を持った安全要件に分解する設計手法が一般的です。つまり、元々の安全要件を複数のプロセッサ (一般に MCU) に冗長的にインプリメントし、冗長系の配備によって致命的な障害の確率を最小化しようという考え方です。

最終的に自動車メーカーとサプライヤーは、自社の E/E システムが業界固有の規則で要求される機能安全および信頼性を提供していることを、認定機関に対して証明する必要があります (http://www.iso.org/iso)。

の大半は現在エレクトロニクスによってもたらされており、その割合は今後もさらに増加すると予想されています。これからの自動車は、自律走行、車両間通信、エンタテインメント、高度なセーフティ機能など、きわめて先進的なソフトウェアとハードウェア テクノロジに基づいて構成されるでしょう。ただし、大量生産の中で競争している自動車サプライヤーにとって、自動車は、処理の並列性を高め、PCB の簡素化を図ると同時に、性能とコストのバランスが取れたソリューションを得る手段として、ECU のデザインにリコンフィギュレーション可能ハードウェアを適用する可能性を示唆しています。

ここに示したアプローチは、まだ構想の初期段階ですが、オートモーティブ向け ECU システムのハードウェア / ソフトウェアの協調テムを実現する実績のある設計手法やツールが市場に既に存在しているほか、DO-254 などの航空セクターの部品やシステムの設計を統括する規格も揃い、FPGA デバイスも既に活用されています。

協調設計を原動力としたパラダイム変化

これまでで述べてきた構想は、ソフトウェアでのみ構築するソリューションを別の構

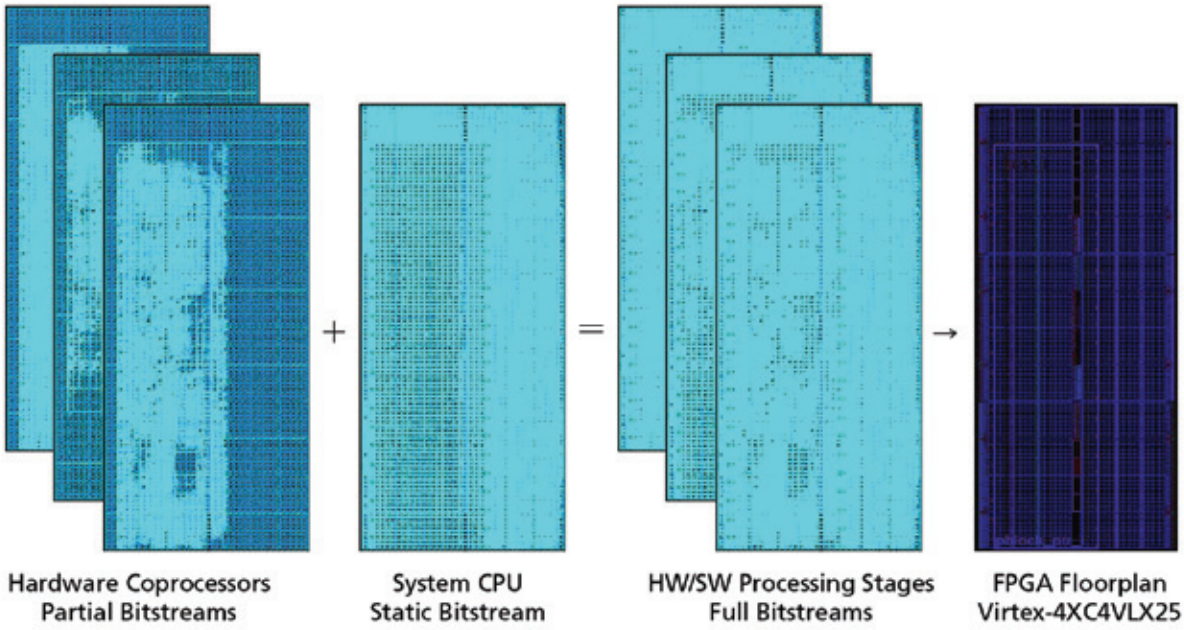


図 5 - 1 つの PR 領域と 1 つのスタティック領域で構成されるランタイム リコンフィギュレーション可能 FPGA に構築した、オートモーティブ ECU アプリケーション向けの低密度で一時的なパーティショニング

の組み込みシステムのコスト管理はきわめて重要です。そのため最近では、個々の ECU が分担する機能を見直し、1 台の自動車に搭載する ECU 数を削減しようという動きも見られるようになりました。これに対応するのが、より強力なコンピューティング プラットフォームです。

業界では、高機能型 ECU をドメイン コントローラーとして作成する取り組みが主流となってきています。すなわち、車両全体の視点でシステムの複雑性を緩和することを目的とし、複数のコア プロセッサおよびマイクロコントローラーを 1 枚のボードに搭載し、ほかのリソースやバス接続を共有させます。このようなトレンド設計を実行するために、AUTOSAR と ISO 26262 の両方をランタイム リコンフィギュレーション可能ハードウェア上に融合するきっかけになるものと考えられます。現時点の AUTOSAR はリコンフィギュレーション可能なハードウェアを対象としていませんが、筆者らはその可能性は将来は現実のものになると確信しています。ランタイム リコンフィギュレーション可能な SRAM ベースの FPGA は、環境条件が厳しく SEU の影響を受けやすい航空宇宙アプリケーションで既に採用されていますが、これまでの歴史の中で、オートモーティブ分野は航空宇宙分野が導入したテクノロジに追従してきたのも事実です。また、FPGA ベースでの安全関連システム成に置き換えることで、オートモーティブ分野におけるコンピューティングの枠組みを変えるものです。これを推進するのが、ECU を対象としたハードウェア / ソフトウェアの協調設計とリコンフィギュレーション可能なコンピューティング テクノロジです。フォン ノイマン型 MCU での純粋にソフトウェア ベースのアプローチをとる ECU では、性能、複雑性、安全性の理由から対応が困難と考えられます。プログラマブル ロジック デバイスのコストの継続的な低下と、オートモーティブ ECU の性能に対する要求の高まりにより、このようなパラダイムの変化はそれほど遠くない将来に現実となるに違いありません。

組み込みプロセッサの新しいカタチ、ZYNQをご紹介します。プロセッサはついにFPGAと統合され、これまで私たちがエクステンシブル プロセッシング プラットフォームと呼んでいた製品はZynq™と名づけられました。おなじみのプログラム手法がさらに直観的になり、お客様の要求を完全にカスタマイズでき、すばやく実装し、いち早く市場に投入できます。あなたがソフトウェアエンジニアで、ARM® Cortex™をご存知ならば、すでにZynqを知っているといえます。あなたがザイリンクスをご存知であれば、これが期待できるイノベーションであるとおわかりになると思います。
詳細は、http://japan.xilinx.com をご覧ください。

XILINX
ZYNQ™

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)443-4016 x2web@teldevice.co.jp ■アヴネット ジャパン(株) TEL(03)5792-8210 EVAL-KITS-JP@avnet.com ■(株)PALTEK TEL(045)477-2005 info_pal@paltek.co.jp
■富士通エレクトロニクス(株) TEL(045)415-5825 fei-xinq@cs.jp.fujitsu.com ■新光商事(株) TEL(03)6361-8086 X-Pro@shinko-sj.co.jp

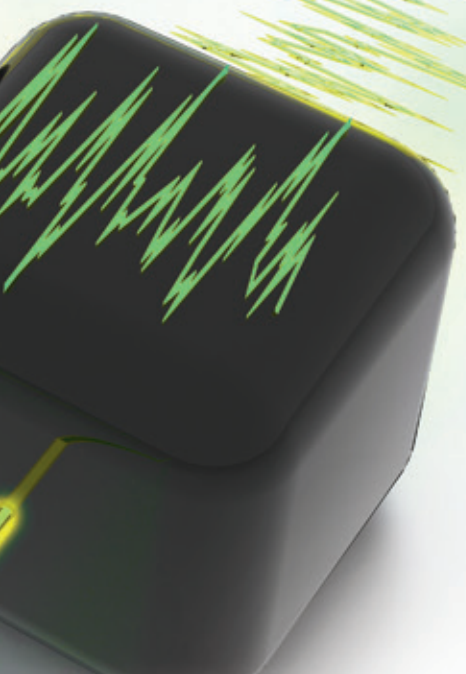
©Copyright 2012 Xilinx, Inc. All rights reserved. ザイリンクスの名称およびロゴ、Artix、Kintex、Virtex、ISEは米国およびその他各国のザイリンクス社の登録商標および商標です。

Zynq-7000 EPP Makes Flexible Software-Defined Radio Platform

柔軟なソフトウェア無線プラットフォームを実現する Zynq-7000 EPP

ザイリンクス FPGA に ARM プロセッサと数多くの波形処理に必要なロジックをすべて統合

Anita Schreiber
Senior Staff Applications Engineer
Defense Applications
and Systems Architecture
Xilinx, Inc.
Albuquerque, N.M.
Anita.Schreiber@xilinx.com



現在の戦術 / 商用ソフトウェア無線 (SDR) には、これまで使用されてきた多くの波形を含め、新しいワイドバンド波形およびブロードバンド波形をサポートできる高い柔軟性と強力な処理性能が要求されます。強力なセキュリティ機能を備えたザイリンクス Zynq™-7000 エクステンシブル プロセッシング プラットフォーム (EPP) は、ARM® テクノロジベースの高性能なプロセッシング システム (PS) に加え、パーシャル リコンフィギュレーションとザイリンクスのアイソレーション デザイン フロー (IDF) をサポートした大容量のプログラマブル ロジック (PL) ユニットを 1 つのデバイスに統合し、これら SDR アプリケーションに対応する理想的なソリューションとなります。

この記事では、Zynq-7000 EPP をベースにした SDR プロジェクトについて説明します。特に、PL ユニットの機能であるパーシャル リコンフィギュレーションと IDF を利用して、さまざまな波形への対応、そして部品点数および消費電力の削減を達成する方法について詳しく解説します。今回開発したサンプル デザインは、Zynq-7000 EPP を利用して 5 つのデバイスを 1 つに集約するなど部品点数の削減を図ると共に、今後の波形も同じハードウェア プラットフォームでサポートできる高い柔軟性を実現しています。また、電力低減機能も備える Zynq-7000 デバイスを SDR システムの中心に据えて使用すると、システムのサイズ、重量、電力、コスト (SWAP-C) を格段に改善し、柔軟なプログラマブル プラットフォームを実現できます。

図 1 は、戦術 SDR のサンプル ブロック図を示したものです。これを見るとわかるように、無線の平文ブロック (この中には機密情報が含まれることもあるため、レッド (red) サイドとも呼ばれる) には汎用プロセッサ (GPP) とレッド FPGA が含まれます。次に、平文の情報は暗号化され、暗号文に変換されます。この暗号文を処理するのがブラック (black) サイドで、この中にはブラック FPGA と GPP、そして波形処理用のモデム FPGA が含まれます。情報セキュリティを確保するため、このデザインでは無線の平文ブロックと暗号文ブロックが分離されており、機密情報などの重要なデータがシステムから平文の形で外部に漏えいするのを防いでいます。

このように、通常のインプリメンテーションでは 3 種類の FPGA と 2 つの GPP が使用されるため、これらの機能を実現するには 5 つのデバイスが必要です。モデム FPGA については、サポートする無線波形をすべて処理できるように設計する必要があります。複数の機能のうち 1 つの機能しか一度に必要ない場合でも、多くの場合モデムはそれらすべてを利用できる状態にあることが求められます。たとえば、統合戦術無線システムで使用される SRW (Soldier Radio Waveform) を処理するには約 120K のロジック セル、8M ビットの RAM、800 個の DSP スライスが必要で、MUOS (Mobile User Objective System) 波形の処理には 200K 以上のロジック セル、10M ビットの RAM、900 の DSP スライスが必要なため、モデム FPGA のサイズは非常に大きくなります。暗号化ブロックのレッド FPGA も、すべての関連波形に対応する暗号化アルゴリズムを格納しようとすると大規模なものとなります。このように必要なデバイスの数やデバイス間の I/O 信号処理量が多く (消費電力が多い)、デバイスのロジック密度が高い (静止電流が多い) 場合、SWAP-C の面で改善の余地が多く残されたソリューションとなってしまいます。

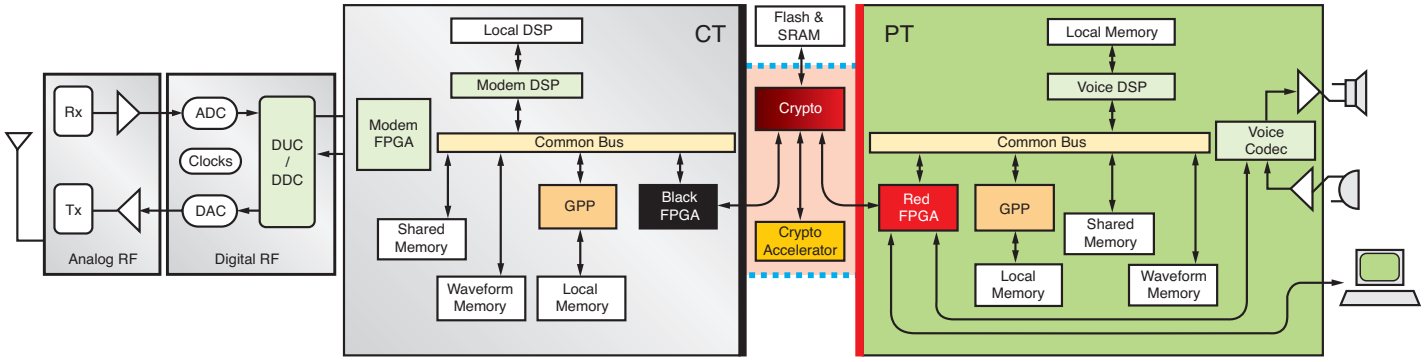


図 1 – レッド FPGA (機密情報用) とブラック FPGA を含むサンプル SDR のブロック図

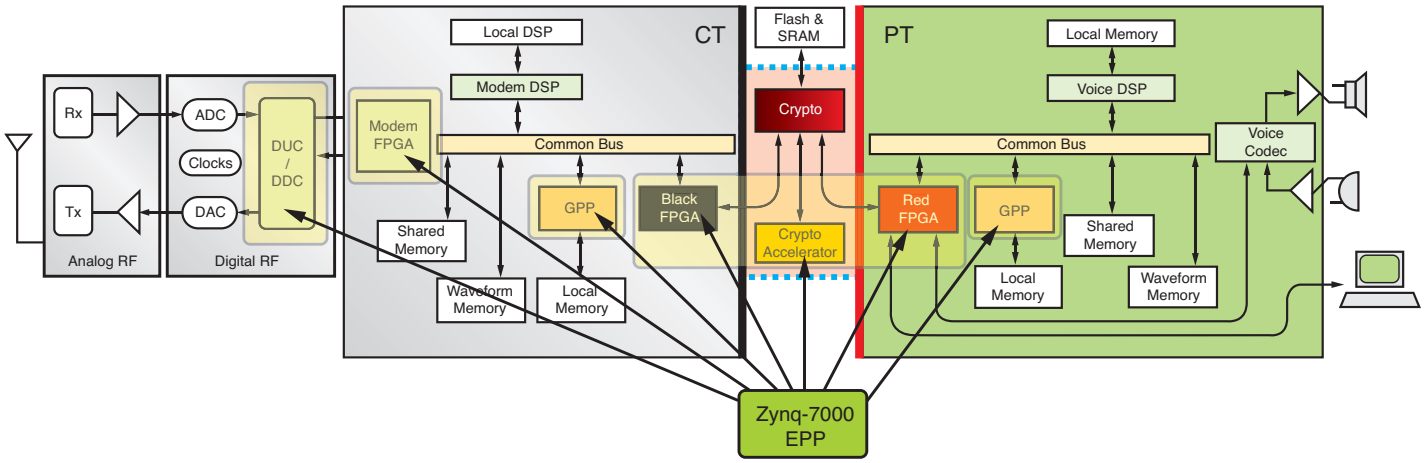


図 2 – 複数の FPGA を 1 つの Zynq-7000 EPP に統合したシンプルな戦術無線用 SDR

ザイリンクス Zynq-7000 エクステンシブル プロセッシング プラットフォーム (EPP) を使用すれば、モデム FPGA、ブラック FPGA、レッド FPGA、レッド GPP、ブラック GPP をすべて 1 つのデバイスに集約できます (図 2)。このように多くのデバイスを統合できる Zynq-7000 EPP は、SWAP-C を抑えつつ柔軟でセキュアな SDR ソリューションを実現します。

ザイリンクス Zynq-7000 EPP は、業界標準の ARM プロセッサをベースにしたプロセッシング システム (PS) とザイリンクスの 28nm HPL (High-Performance, Low Power) プロセスで製造したプログラマブル ロジック (PL) を 1 つのデバイスに統合したものです。PS には 2 つの ARM Cortex™-A9 プロセッサ、L2 キャッシュとオンチップ メモリ、さまざまなペリフェラルが用意され、汎用処理と波形処理を強力にサポートします。PL には豊富なロジック セル、コンフィギャラブル メモリ (デュアルポート RAM、FIFO、シフト レジスタ BRAM)、および DSP 用のハードウェア乗算器があり、必要なファンクションを高速に並列処理できます。

プロセッシング システム

プロセッシング システム (PS) には 2 つの ARM A9 コアが搭載されており、それぞれが 32KB の命令 / データ L1 キャッシュと、512KB の共有 L2 キャッシュを備えています。また、両方のプロセッサから低レイテンシでアクセスできる 256KB のオンチップ メモリや、キャッシュ コヒーレンシを維持するためのスヌープ制御ユニットも用意されています。PL 側のロジックと ARM A9 プロセッサはメモリを共有できるため、これらの間で細かな連携が可能です。また、アクセラレータ コヒーレンシ ポート (ACP) によって、PL 内のロジックから L2 キャッシュとオンチップ メモリへのアクセスが実現し、プロセッサと PL 間で整合が取れた情報の共有が可能になります。オンチップ メモリには PL から直接アクセスすることもできます。また、業界標準のペリフェラルやメモリ インターフェイスも多数用意されており、汎用処理や波形処理に対応しています (図 3)。

安全性に優れた処理

現在、米国政府が商用規格品を組み合わせて安全性に優れた製品を構築するという動きにある中、セキュアな処理機能を持つ Zynq-7000 EPP でレッド サイドとブラック サイドの GPP を両方置き換えることもアプリケーションによっては十分現実的となってきます。Zynq-7000 EPP にはブート プロセスとランタイム環境の両方をセキュリティ保護する機能があります。

一般的な ASSP とは異なり、Zynq-7000 には PL 内部のハードウェア AES 復号化エンジンと SHA-256 ベースの認証エンジン (HMAC) を使用し、暗号化したデザインを安全にコンフィギュレーションできるマスター セキュア ブート モードがあります。図 4 に、セキュアなブート シーケンスを示します。

マスター ARM A9 プロセッサはまずオンチップ ROM から起動します。次に、ブートストラップ ピンで指定された外部ブート デバイスから PS ブート イメージを読み込みます。それと同時に、A9 はデバイス コンフィギュレーション ブロックをコンフィギュレーションし、PS のマスター イメージを AES 復号化エンジンおよび HMAC 認証エンジンへプッシュします。コンフィギュレーション ロジックは、復号および認証処理されたイメージを内部でバッファリングせず、ただちにループバックして PS 内のオンチップ メモリに格納します。マスター A9 はコンフィギュレーション ロジックからの最終認証ステータスをポーリングし、イメージが正しく認証されたことを確認します。認証に失敗した場合、マスター A9 はシステム セキュア リセットをトリガーします。PS イメージの読み込みに成功したら、ファースト ステージ ブートローダーに制御が移ります。次に、ユーザー アプリケーションによって異なりますが、ブートローダーが処理を開始し、PL をコンフィギュレーションしてさらにソフトウェアをロードするか、または外部ソースからのさらなる命令を待つ場合もあります。

安全性に優れたランタイム環境

Zynq-7000 EPP はデバイス全体で ARM 社の TrustZone テクノロジを採用しており、セキュアな実行環境をサポートしています。TrustZone は、隔離されたメモリ領域でソフトウェア タスクを実行することによってコンテンツを未承認の読み出しや書き込みから保護する機能です。ほかのタスクからのアクセスを排除することで、信頼性の高い処理環境が構築、維持されます。

TrustZone は、ARM A9 プロセッサをはじめ PS 内のすべての要素に組み込まれ

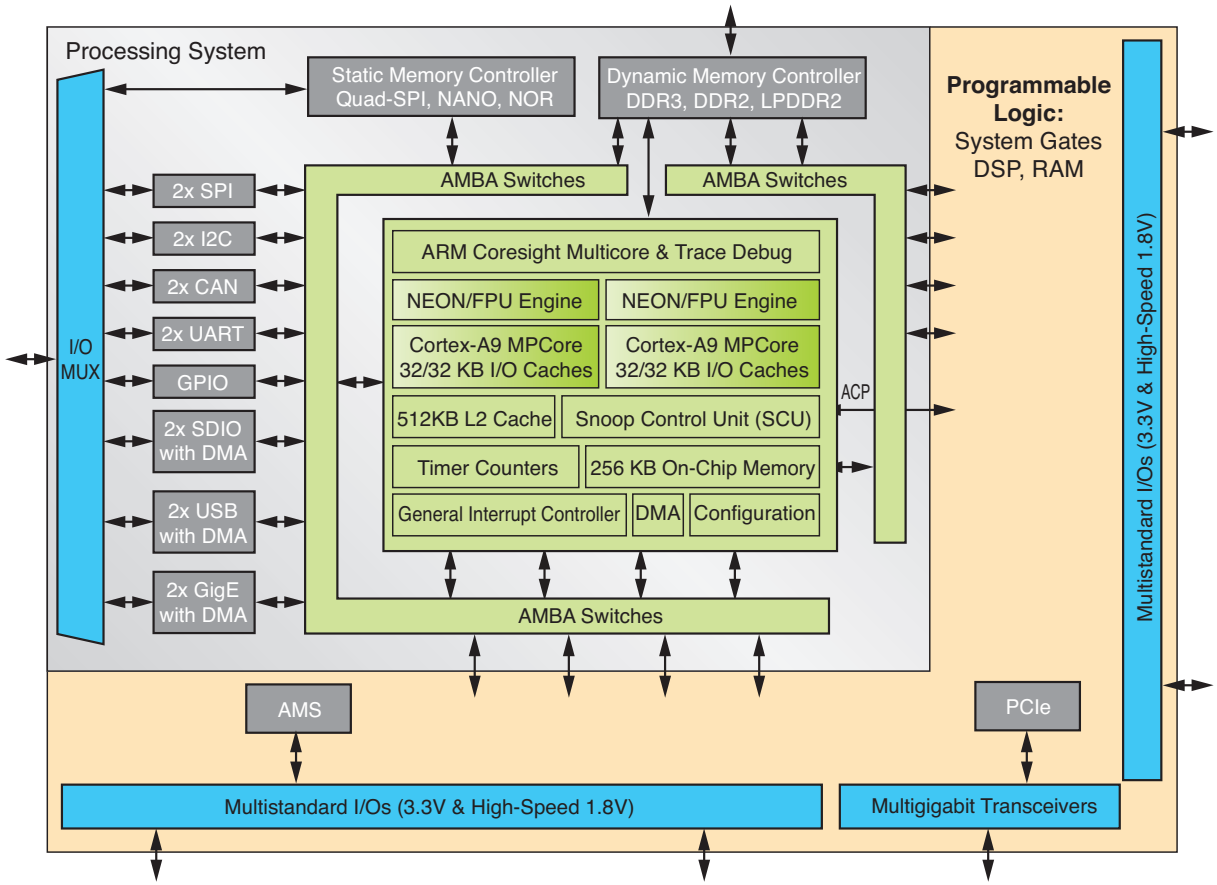


図 3 – 業界標準のペリフェラルとメモリ インターフェイスを数多くサポートした Zynq-7000 EPP のブロック図

ています。

さらに、AMBA® TrustZone 信号は PL にも拡張されており、PL 内部に信頼性の確保されたマスターおよびスレーブ デバイスを作成できます。完全かつセキュアな PL コンフィギュレーションがサポートされているため、ユーザーが作成したこれらのマスター / スレーブ デバイスは、PS 内のハード マクロ ブロックと同じように完全に保護されます。Zynq-7000 EPP と TrustZone ソフトウェアを組み合わせることにより、暗号鍵、機密データ、暗号情報を扱うことのできる安全性の高いシステムを実現できます。

プログラマブル ロジック

Zynq-7000 EPP のプログラマブル ロジック (PL) は ザイリンクスの 28nm HPL (High-Performance、Low Power) プロセス テクノロジで製造されています。Zynq-7000 ファミリにはロジック セル数 28,000 から 235,000 までの 4 種類の製品がラインナップされています。ザイリンクスの 7 シリーズのほかの製品ファミリ同様、Zynq-7000 EPP の PL はコンフィギャラブルなブロック RAM、プログラム可能な DSP 機能、ハード マクロの Gen2 PCIe® (大規模なデバイスのみ)、アジャイル ミックスド シグナル (AMS) テクノロジを備えています。AMS テクノロジは、1Msps の 12 ビット ADC (2 個)、独立したトラック アンド ホールド アンプ (2 個)、基準電圧と温度用オンチップ センサー、外部アナログ入力チャネルで構成されます。

図 2 に示すように、デジタル RF、そしてモデム FPGA、ブラック FPGA およびレッド FPGA のロジックはすべて Zynq-7000 EPP の PL に集約できます。これを可能にしているのは単に PL のファブリック容量が大きいからではなく、パーシャル リコンフィギュレーションとザイリンクス アイソレーション デザイン フロー (IDF) という 2 つの PL の機能を利用していることに理由があります。

パーシャル リコンフィギュレーション

Zynq-7000 EPP の PL に用意されているパーシャル リコンフィギュレーション機能は、動作中のデザインに対して PL の一部のコンフィギュレーションを変更し、別の機能を実行できるようにするもので、通常の FPGA テクノロジよりもさらに高い柔軟性を実現します。完全なコンフィギュレーション ファイルを使用して PL 全体のコンフィギュレーションが完了した後、パーシャル コンフィギュレーション ファイルをダウンロードしてリコンフィギュレーション可能な領域を書き換えることができます。

PL はスタティック ロジックとリコンフィギュレーション可能なロジックで構成されます。パーシャル コンフィギュレーション ファイルをロードしてもスタティック ロジックはまったく影響を受けず動作を継続します。リコンフィギュレーション可能なロジックはパーシャル リコンフィギュレーション ファイルの内容で置き換えられます。パーシャル リコンフィギュレーション ファイルは、リコンフィギュレーション対象外のロジックで実行中のアプリケーション動作に一切影響を与えずダウンロードできます。パーシャル リコンフィギュレーションを利用すると、1 つの Zynq-7000 EPP デバイス上で複数のハードウェア機能を動的に時分割で処理することができます。時間的に独立した機能を特定できたら、これらを隔離してリコンフィギュレーション可能なモジュールとしてインプリメントし、必要に応じて入れ替えて使用するようにします。このようにすれば、デザインに要するすべての機能を同時に PL に格納する必要はなく、Zynq-7000 EPP の PL はそのときどきの処理に必要な機能のみを格納できる容量があれば十分です。ソフトウェア無線 (SDR) は時間的に独立した機能を数多く備えるため、パーシャル リコンフィギュレーションのメリットは大きく、柔軟性とリソース使用率が格段に向上するほか、スタティック消費電力も削減されます。

また、Zynq-7000 EPP のアーキテクチャでは PS で実行中のソフトウェアがパーシャル リコンフィギュレーションを利用して PL の一部を書き換えることができるため、パーシャル リコンフィギュレーションの柔軟性はさらに向上します。セキュアなデザインでは、PS と PL のイメージは最初に認証されて信頼性が確立しているため、PS または PL の制御の下で読み込まれるパーシャル リコンフィギュレーション ファイルは暗号化されていなくても問題ありません。したがって、波形の種類に合わせてソフトウェア処理アルゴリズムを入れ替えるだけでなく、そのアルゴリズムをインプリメントする PL も適宜入れ替えることができます。

アイソレーション デザイン フロー (IDF)

アイソレーション デザイン フロー (IDF) を利用すると、レッドとブラックの処理ロジックを同一の FPGA デバイス上に配置でき、SDR の暗号ブロックの設計者は PL の機能を最大限に活かすことができます。IDF は、レッド ファンクションとブラック ファンクションを別々の FPGA ではなく 1 つのデバイス上で動作できるようにザイリンクスが開発したものです。このようなワンチップ アプリケーションの例としては、冗長構成の Type-I エンクリプター、常駐型のレッドおよびブラック データ、そして複数のセキュリティ レベルで動作する機能などがあります。

IDF では、未使用のデバイス コンポーネントをフェンスとして使用することにより、物理的に隔離された複数の独立ファンクションを 1 つの FPGA デバイスにインプリメントできます。このフェンスがそれぞれの独立したファンクションを分離し、デバイス内に隔離された領域を作成します。このフローでは、早期フロアプランニング、モジュラー型デザイン、モジュラー型合成を利用すると共に、各種ガイドラインや注意事項に従うことにより、目的のファンクションどうしが確実に隔離されるようにします。デザインのインプリメンテーション後、ザイリンクスのアイソレーション検証ツール (IVT) を使用してモジュールとフェンスを可視化し、アイソレーションに関するデザイン ルールを正しくインプリメントできたか検証することができます。

各波形処理ブロックをリコンフィギュレーション可能なロジックとして定義しておけば、Zynq-7000 EPP 内の PL は波形処理ブロックを一度に 1 つ格納できるだけの大きさがあればよいことになります。IDF を利用すれば、暗号化と復号 (レッドおよびブラック) 処理ロジックを各波形処理ブロック内に共存させることができます。パーシャル リコンフィギュレーションを用いれば、異なる波形処理ブロックをソフトウェア制御によって Zynq-7000 EPP の PL で動的に切り替えることができます (図 5)。

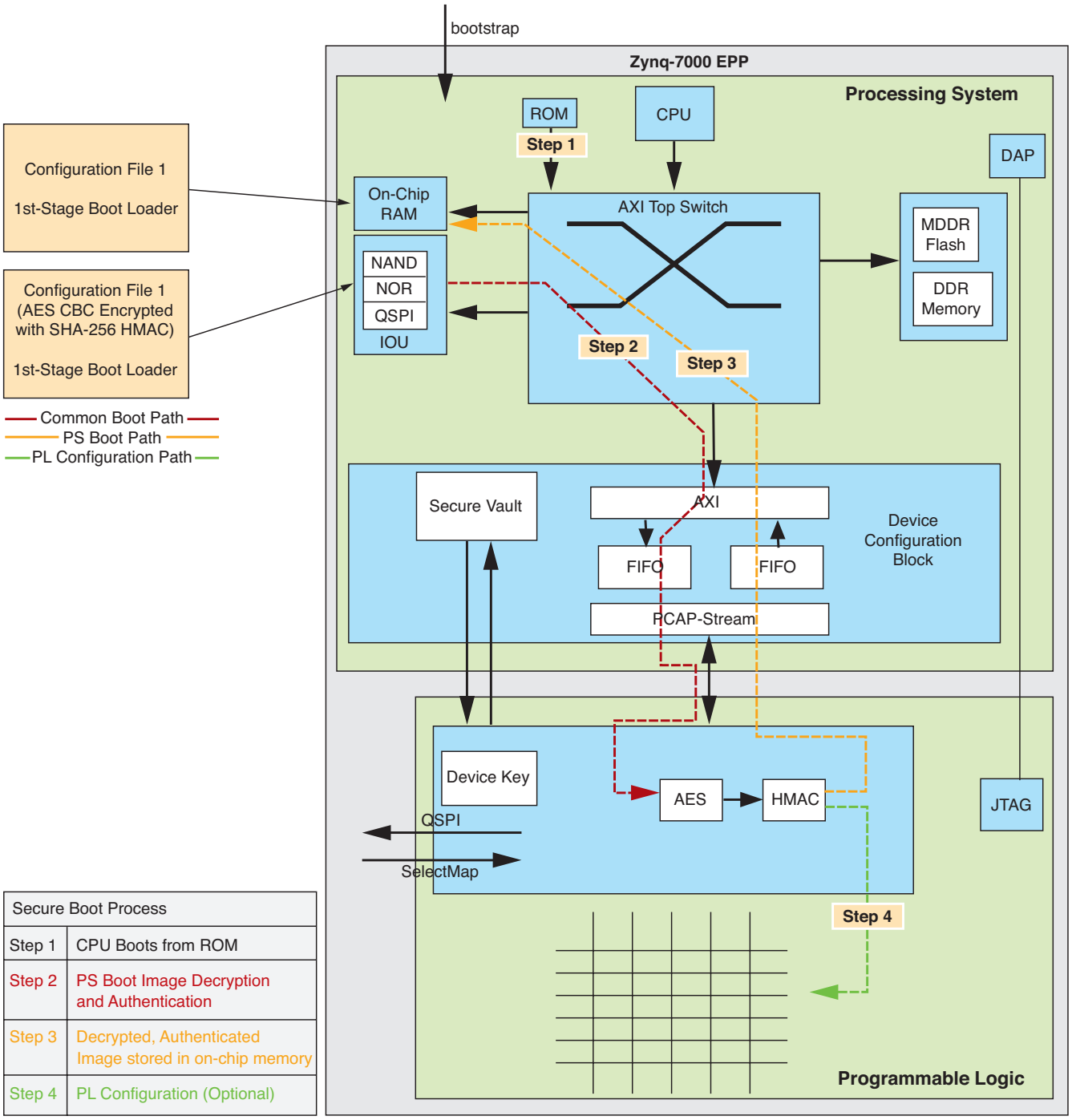
個々のアプリケーションのニーズに基づいて新しいソフトウェアと波形処理ブロックを開発し、Zynq-7000 EPP の PS と PL にリコンフィギュレーション可能なロジック

として動的に入れ替えできるようにしておけば、将来的に新しい波形にも同じ SDR で対応でき、SDR の全体的な製品寿命が長期化します。

省電力機能

Zynq-7000 EPP は、PS のみのパワーオン モード、スリープ モード、ペリフェラルごとに独立したクロック ドメインなど、処理システム全体の消費電力を削減するための機能を数多くサポートしています。これらの機能を使用することで、アイドル状態におけるデバイスのダイナミック消費電力を大幅に抑えることができます。

Zynq-7000 EPP の PS と PL はそれぞれ電源レールが独立しており、電源ピンもそれぞれ専用となっています。Zynq-7000 EPP は PS のみのパワーオン モードはサポートしていますが PL のみのパワーオン モードはサポートしていません。PS 内のユーザー ソフトウェアは PL



Secure Boot Process	
Step 1	CPU Boots from ROM
Step 2	PS Boot Image Decryption and Authentication
Step 3	Decrypted, Authenticated Image stored in on-chip memory
Step 4	PL Configuration (Optional)

図 4 – デバイスのセキュリティを制御する Zynq-7000 EPP のセキュア ブート プロセス

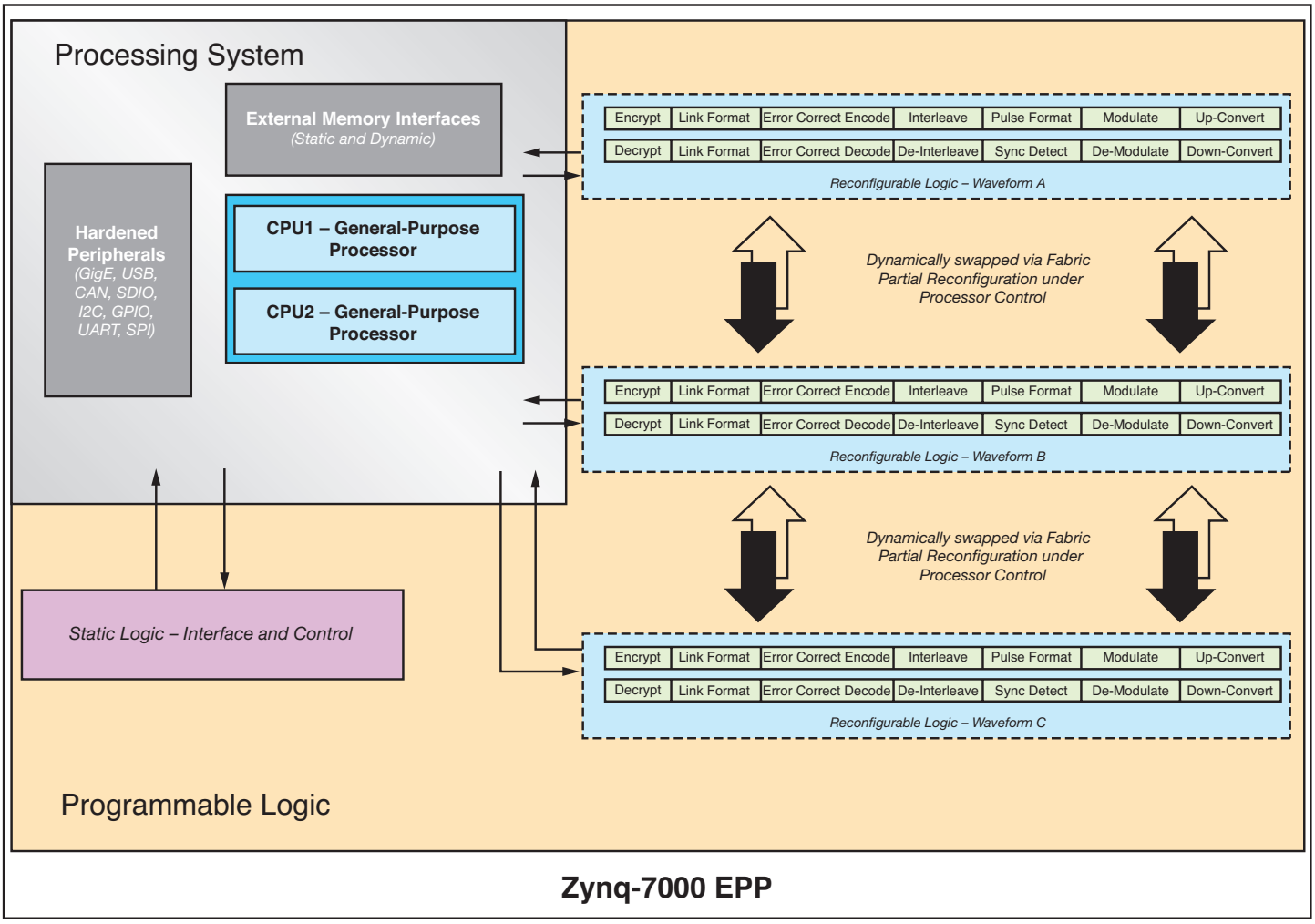


図 5 - Zynq-7000 EPP でパーシャル リコンフィギュレーションとアイソレーション デザイン フロー (IDF) を利用し、複数の波形をサポート

ファブリックの電源を随時オン / オフしてスタティック消費電力を削減することができます。PS の電源をオフにすると、PL は PS がリセット状態を終了するまで永続的にリセット状態に保たれます。

スリープ モード（割り込みまたは例外により復帰）では、ARM A9 プロセッサのうち 1 つのみが約 10MHz で動作し、インターコネクトはシャットダウンされ、DDR はセルフ リフレッシュ モードとなり、すべての DDR 終端はオフに切り替わり、そして選択したウェイクアップ デバイス（CAN、Ethernet、GPIO）を除く PS ペリフェラルはすべてクロック ゲーティング モードとなります。スリープ モードにおけるダイナミック消費電力は、ウェイクアップ割り込みを監視する CPU 回路の一部、スヌープ制御ユニット、そしてウェイクアップ ペリフェラル デバイスから発生するもののみです。さらに、スリープ モード時に PL の電源をシャットダウンするとスタティック消費電力も削減できます。Zynq-7000 EPP は多くのクロック ドメインをサポートしており、それぞれに独立したクロック ゲーティング制御があります。未使用のクロック ドメインをシャットダウンすれば、ダイナミック消費電力を削減できます。PS 内の各ペリフェラル（GPIO はサイズが小さく低速なので除く）は独立したクロック ドメインに属しており、それぞれにクロック ゲーティング制御が可能です。このため、システム レベルの制御レジスタを使用してソフトウェアで各ペリフェラルの電源をオフにできます。

安全性に優れたインテグレーション

セキュアなインテグレーションという伝統を受け継いだザイリンクス Zynq-7000 EPP は、これまで 5 つ以上のデバイスを使用して実現していた戦術 / 商用 SDR の処理性能とロジック機能をワンチップに集約します。このデバイスは、PL 内のロジック機能の一部またはすべてを動的にリコンフィギュレーションすることにより、同じロジック リソースを再利用して複数の波形をサポートできます。このため、将来必要になる波形も含めた数多くの波形を柔軟にサポートできます。

複数のデバイスを 1 つの Zynq-7000 EPP に集約することで、SDR 全体のサイズ、重量、電力、コストが大幅に削減されます。しかも、Zynq-7000 EPP にはスタティックおよびダイナミック消費電力を削減する機能も数多く用意されており、柔軟で安全性に優れた戦術 / 商用 SDR に理想的なソリューションとなります。

Using FPGAs to Solve Tough DSP Design Challenges

デジタル信号処理の課題を FPGA で解決

Reg Zatrepalek
DSP/FPGA Design Specialist
Hardent Inc.
rzatrepalek@hardent.com

DSP および FPGA テクノロジの概要と、両アーキテクチャを FIR フィルター アプリケーションに適用した場合の具体的な比較

アナログ信号を高速に計測し、フィルター処理し、ときには圧縮する DSP は、電子回路システムの中で重要な役割を担っています。このような処理を通じて、デジタル世界と、現実世界でもあるアナログ世界をつないでいるともいえますが、電子回路システムがより複雑になり、複数のアナログ信号源にも対応しなければならなくなった現在、エンジニアは難しい選択を迫られています。すなわち、複数の DSP を使用して互いの機能を同期させて処理するのが適切か、それとも高性能な DSP を単体で使用して高機能なソフトウェアで複数の機能を処理するのが適切か、という選択です。

最近のシステムは 1 つの DSP によるインプリメンテーションでは処理能力が不足するほど複雑化している場合が少なくありません。では複数の DSP を使用すればいいかというと、コスト、複雑さ、および消費電力の面で課題が生じ、システム アーキテクトにはなかなか受け入れてもらえないのが実情です。

こういった中で、高性能なデジタル信号処理を必要とするシステムに最適な選択肢として FPGA が検討されるようになっています。実際に、FPGA テクノロジは多くの場合、信号処理を対象とした高度な課題に対して、スタンドアロンのデジタル シグナル プロセッサよりもはるかに簡単にソリューションを実現できます。その理由を理解するために DSP の誕生と発展の過程を見ていきましょう。

信号処理専用のマイクロプロセッサ

従来の DSP アーキテクチャでは、ここ 20 年ほど、性能向上に対する要求に応えることが困難になってきています。ビデオ システムは高解像度化や 3D 化に向けて大きく進歩し、通信システムにおいてはさらなる広帯域を目指してテクノロジの限界が押し広げられていく中で、設計者は代替となるインプリメンテーション ストラテジを必要としています。デジタル信号処理アルゴリズムのインプリメンテーションに使用するハードウェアは、マイクロプロセッサ、ロジック、およびメモリの、3 つの基本的なデバイスに分類されることがあります。一部のデザインでは、アナログ - デジタル (A/D) 変換やデジタル - アナログ (D/A) 変換、さらには高速デジタル インターフェイス向けにハードウェアを追加する必要があります。

従来の DSP は信号処理に特化したマイクロプロセッサとして開発されています。このようなプロセッサはアルゴリズム中心の処理には適していますが、性能はクロック レート、および内部アーキテクチャの逐次処理という性質によって制限されます。このため、入力データ サンプルに対して一秒当たりに実行できる最大処理数は限られていました。一般に演算論理ユニット (ALU) の演算処理には命令あたり 3 ～ 4 クロック必要です。アーキテクチャをマルチコア化すれば性能は向上するかもしれませんが、こういった制約が解消されることはありません。したがって、従来の信号プロセッサを用いるデザインにはアルゴリズムのインプリメンテーションにアーキテクチャの構成要素を繰り返し利用する必要があります。加算、乗算、減算などの基本的な演算処理は、内部と外部のいずれからデータが与えられるかにかかわらず、すべて ALU を経由しなければなりません。

最新の高性能アプリケーションの多くに対応しようとしても、従来の DSP ではシステム要件を満たすことができません。これまで、デバイス内に複数の ALU を設けたり、ボード上に複数の DSP を設けるなど、さまざまな解決策が提案されてきましたが、コストの大幅な上昇を招いたり、単に問題をほかの部分にシフトさせただけでした。たとえば、複数のデバイスを使用して性能の向上を図ろうとすると、必要なデバイス数は急激に増加します。性能を 2 倍にすることが目的であれば 2 個のデバイスが必要となり、さらにその 2 倍の性能が必要であればデバイスを 4 個用いる必要があります。また、プログラマーの焦点も信号処理から、複数のプロセッサおよびコアにまたがってタスク スケジューリングを制御することへとシフトすることが多々あります。これにより、当面の問題であるデジタル信号処理ではなく、システム オーバーヘッドとなる数多くの追加コードへの対応を迫られます。

複雑化の一途を辿る DSP のインプリメンテーションに対して、FPGA テクノロジによって解決しようという動きが現れました。FPGA は当初、より高レベルな統合性、パフォーマンス、および柔軟性を実現するために、ディスクリート メモリやディスクリート ロジックを集積する手段の 1 つとして開発されました。現在 FPGA テクノロジは、ほぼすべての高性能システムにおいて欠かせない構成要素となっています。従来の DSP とは異なり、コンフィギャラブル ロジック ブロック (CLB)、メモリ、DSP スライスなどを一定に配置した大規模な並列構造が特長である FPGA は、VHDL や Verilog のような上位レベル言語か、ブロック図レベルでは System Generator を使用してプログラミングします。数多くの専用機能や IP コアが提供されており、FPGA に最適化された形で直接インプリメントできます。

FPGA にはシステム要件に合わせてデジタル信号処理回路をインプリメントできるというメリットがあります。つまり、マルチチャネル システムあるいは高速システムに対しては高度な並列処理を採用して性能を最大限に高められる一方で、低レート システムであれば逐次処理としてインプリメントできます。つまり設計者は、デバイスの逐次処理の制限に合わせてデザインを妥協するのではなく、アルゴリズムとシステム要件に合わせてインプリメンテーションを調整できるのです。さらに、超高速 I/O を使用すれば、データ キャプチャに始まり処理チェーンを経て最終出力に至るまでのデータ フロー性能を高めることができ、コストや性能のボトルネックを抑えられます。

例として、従来の DSP および FPGA アーキテクチャを使用してそれぞれ FIR フィルターをインプリメントし、両ソリューションの強みと弱みを明らかにしていきましょう。

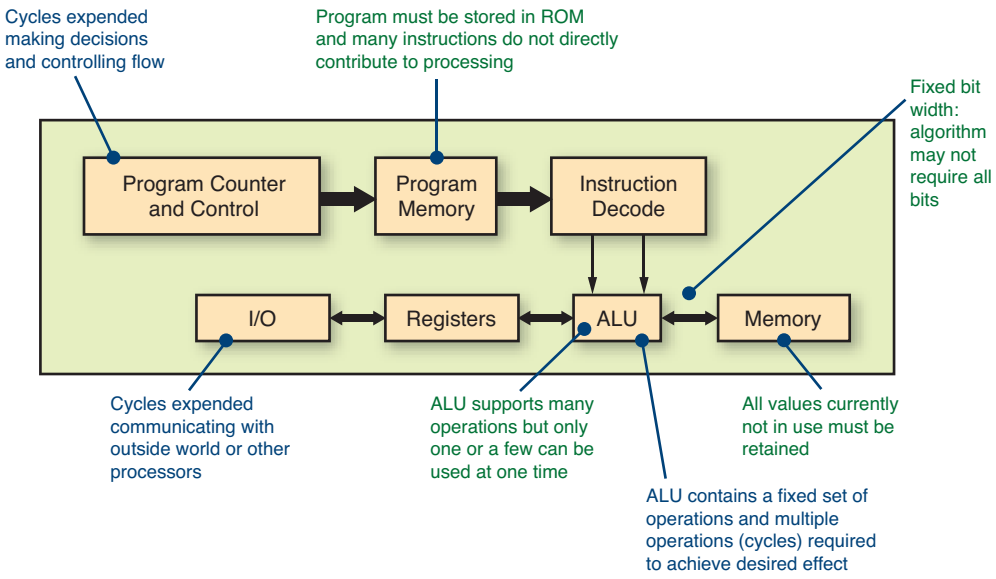


図 1 - 従来の DSP アーキテクチャ

例：デジタル FIR フィルター

デジタル信号処理では有限インパルス応答 (FIR) フィルターが広く使用されています。フィルター処理はデータ信号に含まれる振幅あるいは周波数を変更するために用いられ、一般には、サンプル データ スペクトラムの特定範囲の分離または強調が目的です。つまりフィルターは、信号のプリコンディショニング手法の 1 つと考えられます。一般的なフィルター アプリケーションでは、フィルター係数を使用して入力データをサンプリングし、フィルターの種類とインプリメンテーション方法に依存する数値演算を同期動作によって処理し、次の処理工程へとデータを転送します。データ ソースと宛先がアナログ信号の場合、データ サンプルはまず A/D コンバーターを通って、最後に D/A コンバーターから出力されます。

FIR フィルターの構成を単純化すると、直列に構成した遅延要素、乗算器、および加算ツリー / チェーンを用いて実装されます。

数学的にはシングル チャネル FIR フィルターは次の式で表されます。

$$Y_n = \sum_{i=0}^{N-1} k_{n-1} S_i$$

式の各項は、入力サンプル、出力サンプル、および係数に対応します。S は入力サンプルの連続ストリーム、Y は出力サンプルのフィルタリング後のストリーム、n と k は特定の時刻に対応します。つまり、時刻 n における出力サンプル Y(n) を計算するには、時間において異なる N 個のサンプルのグループ、つまり s(n)、s(n-1)、s(n-2)、… s(n-N+1) が必要です。N 個の入力サンプルのグループを係数 N で乗算し、それぞれを加算して、最終結果 Y を得ます。

図 2 は単純な 31 タップの FIR フィルターのブロック図です (長さ N=31)。

さまざまなデザイン ツールがフィルターの理想的な長さや係数の選択をサポートしています。必要なフィルター性能が得られる最適なパラメーターの選択が目的です。このようなパラメーター選択のツールには MATLAB® が最も使用されています。フィルター

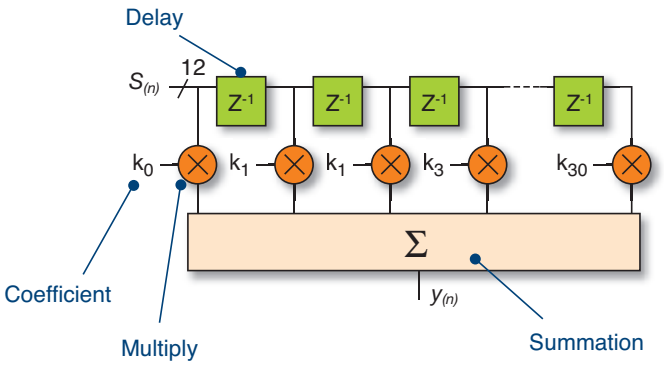


図 2 - 長さ 31 の FIR フィルター

パラメーターの選択後、数式に従ってインプリメンテーションを進めます。

FIR フィルターの基本的なインプリメンテーション手順は次のとおりです。

1. 入力データ ストリームをサンプリングします。
2. キャプチャした各サンプル データを各フィルター係数で乗算できるように、入力サンプルをバッファー内に揃えます。
3. 各サンプル データをそれぞれの係数で乗算し、すべての結果を累算します。
4. フィルター結果を出力します。

積和演算方式に従ってプロセッサ上にこの FIR フィルターをインプリメントする C プログラムのコード例を次に示します。

```
  /*
  * Capture the incoming data samples
  */
  datasample = input();
  /*
  * Push the new data sample onto the buf-
fer
  */
  S[n] = datasample;
  /*
  * Multiply each data sample by each coef-
   ficient and accumulate the result
  */
  y = 0;
  for (i = 0; i<N; i++)
  {
  y += k[i] * S[(n + i) %N];
  }
  n = (n+1) %N;
  /*
  * Output filtered result
  */
  output(y);
```

図 3 に示すインプリメンテーションは積和演算方式または MAC (multiply-and-accumulate) 方式として知られています。従来の DSP にフィルターをインプリメントする最も一般的な方法です。コアのクロック レートが 1.2GHz の典型的な DSP に 31 タップの FIR フィルターをこの方法でインプリメントした場合の最高性能はおよそ 9.68MHz となり、すなわち最大入力データレートは 9.68M サンプル毎秒 (MSPS) が得られます。

一方で FPGA を採用するとさまざまな方法でインプリメンテーションや最適化を実行できます。たとえば、きわめて高いリソース効率が求められる場合は、MAC エンジン手法でのインプリメンテーションが最適です。31 タップ フィルターを例に、フィルター仕様を満たすために必要なロジック リソースについて説明します。図 4

にインプリメンテーションのブロック図を示します。

データと係数の格納には内部メモリを使用しています。これには、FPGA 内部の RAM と ROM の組み合わせが適当です。データ サンプルには RAM を使用し、サイクリック RAM バッファーとして実装します。ワード数はフィルター タップ数と等しくし、ビット幅はサンプル サイズに設定します。ROM は係数に使用します。ワースト ケースでは、ワード数はフィルター タップ数に等しくなりますが、係数に対称性があればワード数を減らすことができます。ビット幅は最大係数をサポートできるだけの大きさが必要です。乗算器にはデータ サンプルと係数データの両方がサイクルごとに変わるため全乗算器が必要です。生成された結果はアキュムレータで加算します。フィルターがデータをサンプリングするとクロックごとにアキュムレータ出力が変化するため、キャプチャ レジスタを設けます。N 個のサンプルすべてを累算した最終結果が出力レジスタにキャプチャされます。

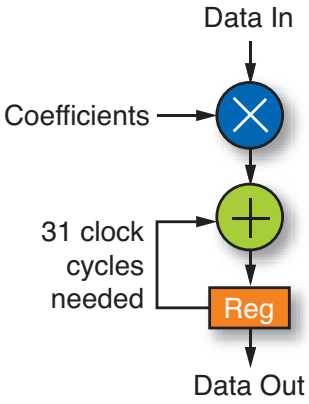


図 3 - 従来の DSP を使用した MAC インプリメンテーション

MAC モードを使用する場合は DSP48 が最適です。入力レジスタ、出力レジスタ、および加算ユニットが DSP48 スライス内に設けられています。31 タップの MAC エンジンのインプリメンテーションに必要なリソースは、DSP48 と 18Kb ブロック RAM がそれぞれ 1 個ずつ、そして 9 個のロジック スライスです。サンプル データと係数のアドレス生成および制御に、さらにいくつかのスライスが必要です。スピード グレードが -3 のザイリンクス 7 シリーズ デバイスで、FPGA 内に 600MHz クロックが存在するとき、このフィルターは 19.35MHz の入力サンプル レートで動作し、すなわち 19.35MSPS が得られます。

システム仕様としてより高性能な FIR フィルターが求められる場合は並列構成を選択します。図 5 に Direct Form Type I (直接型タイプ 1) インプリメンテーションのブロック図を示します。

Direct Form Type I フィルター構造は FPGA で最高レベルの性能を実現するインプリメンテーション方法です。この構造はシストリック FIR フィルターとも呼ばれ、パイプラインと加算器チェーンを使用して DSP48 スライスから最大限の性能を引き出します。入力はデータ サンプル バッファーとして機能するカスケード接続のレジスタに提供されます。各レジスタは入力サンプルを DSP48 に供給し、次に、対応する係数を使用して入力サンプルを乗算します。加算器チェーンには部分積が格納され、最後に連続的に加算処理を行って結果を求めます。

フィルターの実現に外付けロジックは不要であり、また、数多くの係数に対応できるように拡張も可能です。入力信号のファンアウトが少ないため、最大限の性能が得られます。31 タップ FIR フィルターのインプリメンテーションに必要なリソースは 31 個の DSP48 スライスのみです。スピード グレードが -3 の ザイリンクス 7 シリーズ デバイスで、FPGA 内に 600MHz クロックが存在するとき、このフィルターは 600MHz の入力サンプル レートで動作し、すなわち 600MSPS が得られます。

以上の例から、FPGA は従来の DSP を上回る優れた性能を発

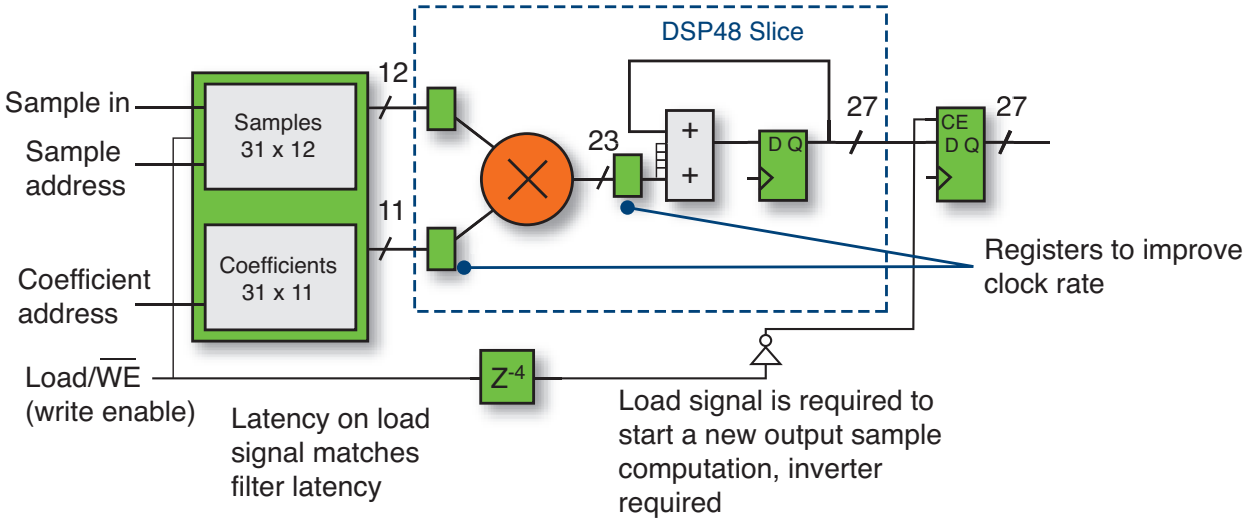


図 4 - MAC エンジンを用いた FPGA の FIR フィルター

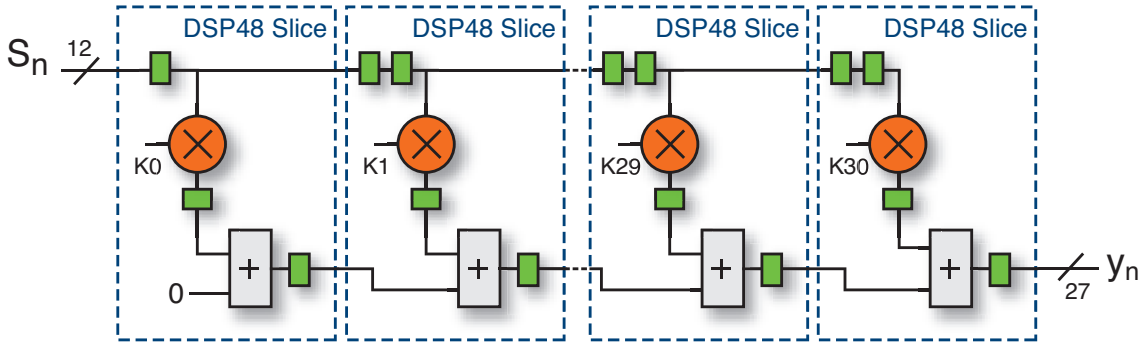


図 5 - Direct Form I を用いた FPGA の FIR フィルター

揮するだけでなく、同じ性能が得られるのであればクロック レートを大幅に下げることもわかります (消費電力も低下)。

今回は FPGA に FIR フィルターをインプリメントする 2 つの方法のみを示しました。データのサンプル レート要件に応じて、逐次的 MAC 型と完全な並列処理型の中間に位置するアーキテクチャを採用してもかまいません。また、係数の対称性、補完、デシメーション、マルチチャネル、マルチレートなど、性能と使用リソースのトレードオフを考慮することもできます。これらのさまざまな手法を検討するにあたって、ザイリンクス CORE Generator™ ツールや System Generator などのユーティリティが有効です。

従来 DSP と FPGA の選択判断

従来の DSP には実績があり、特定の課題に対して最適なソリューションとなる事例も存在します。システムのサンプル レートが数 kHz 以下で、かつ、シングル チャネル インプリメンテーションであれば、DSP は最適な選択肢となるでしょう。一方で、サンプル レートが数 MHz 高い場合、あるいはシステムがシングル チャネルよりも多くのチャネルを必要とする場合、FPGA はより魅力的なアプローチとなります。DSP ではデータ レートが高くなるほど、キャプチャ、処理、およびデータ出力を損失なく処理することが難しくなってきます。その理由は、バス、そして場合によってはプロセッサ内のコアさえもが共有リソースとなるためです。FPGA であれば各機能に専用リソースを割り当てることができます。

DSP はクロック基準ではなく命令基準で動作します。一般に、1 つのサンプル データを対象とする数値演算処理には、3 命令ないし 4 命令が必要です。データはまず入力点でキャプチャされ、次にプロセッシング コアに送信され、演算処理に応じてコア内を循環し、最後に出力されます。対照的に FPGA はクロック基準で動作するため、各クロック サイクルごとに入力データ ストリームに対して数値演算を適用できる能力を潜在的に備えています。

DSP は命令またはコードで動作するため、標準的な C 言語か、高性能を求めるのであれば下位レベルのアセンブリ言語でプログラミングを行います。コードは上位レベルのディシジョン ツリーまたは分岐動作として作成されると考えられますが、そのままでは FPGA にはインプリメントできません。DSP 用に、オーディオや電話コーデックなど、あらかじめ指定された機能や標準的な機能を実行するさまざまなコードが存在します。

高性能 DSP システムに FPGA を使用する利点は FPGA ベンダーやサードパーティ パートナーにも広く認知されており、ビデオ、画像処理、通信、オートモーティブ、医療、防衛アプリケーションなど、幅広い垂直市場を対象とした多くの IP コアが提供されていることです。DSP 用に C コードを書くよりも簡単に、ハイレベルのシステム ブロック図を FPGA モジュールや IP コアに落とし込めることも少なくありません。

DSP から FPGA への移行

従来 DSP と FPGA の選択を判断するいくつかの基準を確認していきましょう (表 1 を参照)。

設計者の数ではソフトウェア プログラマーの方がハードウェア設計者を上回ります。同様のことが DSP プログラマーと FPGA 設計者の数にもいえます。ただし、ソフトウェア エンジニアがハードウェアを設計することに比べれば、システム アーキテクトまたは DSP 設計者が FPGA の設計を担うことはそれほど難しくありません。DSP アルゴリズムの開発や FPGA のインプリメンテーションを短期間で習得できる、さまざまなリソースが用意されています。

障害となるのは、サンプル型あるいはイベント型のアプローチから、クロック型の問題記述およびソリューションへと発想を転換しなければならない点でしょう。システム アーキテクチャと設計工程の定義段階であれば、このような転換をより簡単に理解して適用することができます。また、別のエンジニアや数学者がそれぞれにシステム アーキテクチャ、DSP アルゴリズム、または FPGA のインプリメンテーションを定義することも珍しくありません。こういった場合は、各メンバーがほかのメンバーが直面している課題をある程度認識している方が、よりスムーズに作業を運ぶことができるでしょう。

適切な FPGA インプリメンテーションを行う際にアーキテクトが FPGA の設計に精通している必要はありません。デバイス、リソース、およびツールの基本を理解していれば充分です。これらの手順については、提供されている専門のセミナーなどを利用することで短時間に対応できます。

実際の設計手順はエンジニアの経験や専門性によって変わります。特に DSP については、アルゴリズム開発、効率的なインプリメン

	FPGA	DSP
システムのサンプル レート	サンプル レートが数 MHz 以上かマルチチャネルが必要な場合は FPGA が最適	サンプル レートが数 kHz のシングル チャネル アプリケーションにはスタンドアロン DSP の方がコスト効率が高い
システムの仕様	ブロック図を元に仕様を開発するか、ブロック図を包含して仕様を開発すれば、FPGA 内のモジュールに比較的容易にマッピングできる	C モデルに基づいてシステム仕様を定めている場合は、コードを DSP 用の機能モジュールに比較的簡単に落とし込める
データ レート	I/O データ レートが数 MB/s を超える場合は FPGA の方が効率良く、簡単にインプリメントできる	低データ レートで、通常は余剰なサイクルが存在する場合は、DSP でも充分な帯域を実現できる
データ幅または精度	高度な並列構造を実現できる FPGA はデータ幅の広いシステムや高精度システムに最適	データ幅がプロセッサバス幅と同じで、アルゴリズム内にビット拡張の可能性がない場合は、特に制限なし
ディシジョン ツリーまたは条件分岐	条件分岐の代わりにステート マシンが利用可能。ディシジョン ツリーが必要な場合は、組み込みプロセッサ コアまたは MicroBlaze ™ プロセッサの搭載が最適	DSP は一般に分岐のインプリメンテーションが容易
浮動小数点	浮動小数点コアを内蔵可能。高精度固定小数点のインプリメンテーションを浮動小数点の代わりに用いた方がより高性能なソリューションを実現できる場合もある	従来の DSP は浮動小数点コアを内蔵
従来のコード	C コードを FPGA に適応する手法が既に登場している	既存の C コードを再利用できればコード開発期間を短縮できる。ただし、従来のシステム仕様を超える性能が得られるかが課題
コア、ライブラリ、サードパーティ製 IP	パラメーター指定が可能で FPGA に最適化された豊富な IP コアやライブラリがザイリンクスとサードパーティから提供されている。FPGA のインプリメンテーションに関するライセンス契約は一般に単純で、交渉も容易	従来の DSP 用にさまざまなコアやライブラリが提供されている。出荷数あるいは売上げに応じたロイヤリティが設定されている場合があり、複雑な計算を要する

表 1 - DSP と FPGA の選択における判断基準

テーション方法、System Generator デザインなどのセミナーが提供されています。DSP デザインを FPGA に移行する技術を習得することが目的であれば、Hardent 社およびザイリンクスの認定トレーナから提供されている、「DSP デザイン基礎」、「ザイリンクス FPGA 向け DSP デザイン手法入門」、および「System Generator を使用した DSP デザイン」という 3 つのトレーニング コースの受講を推奨します。

Hardent 社は、ザイリンクスのデバイス、HDL デザイン入門言語、最適化テクニック、デザインおよびデバッグの進め方などに関連するセミナーも提供しています。上級者向けのコースおよびワークショップとしては、高速 I/O の課題、エンベデッド プロセッシング、DSP デザイン テクニックなどが用意されています。

Hardent 社のトレーニング コースのスケジュールについては www.hardent.com/training に掲載されています。

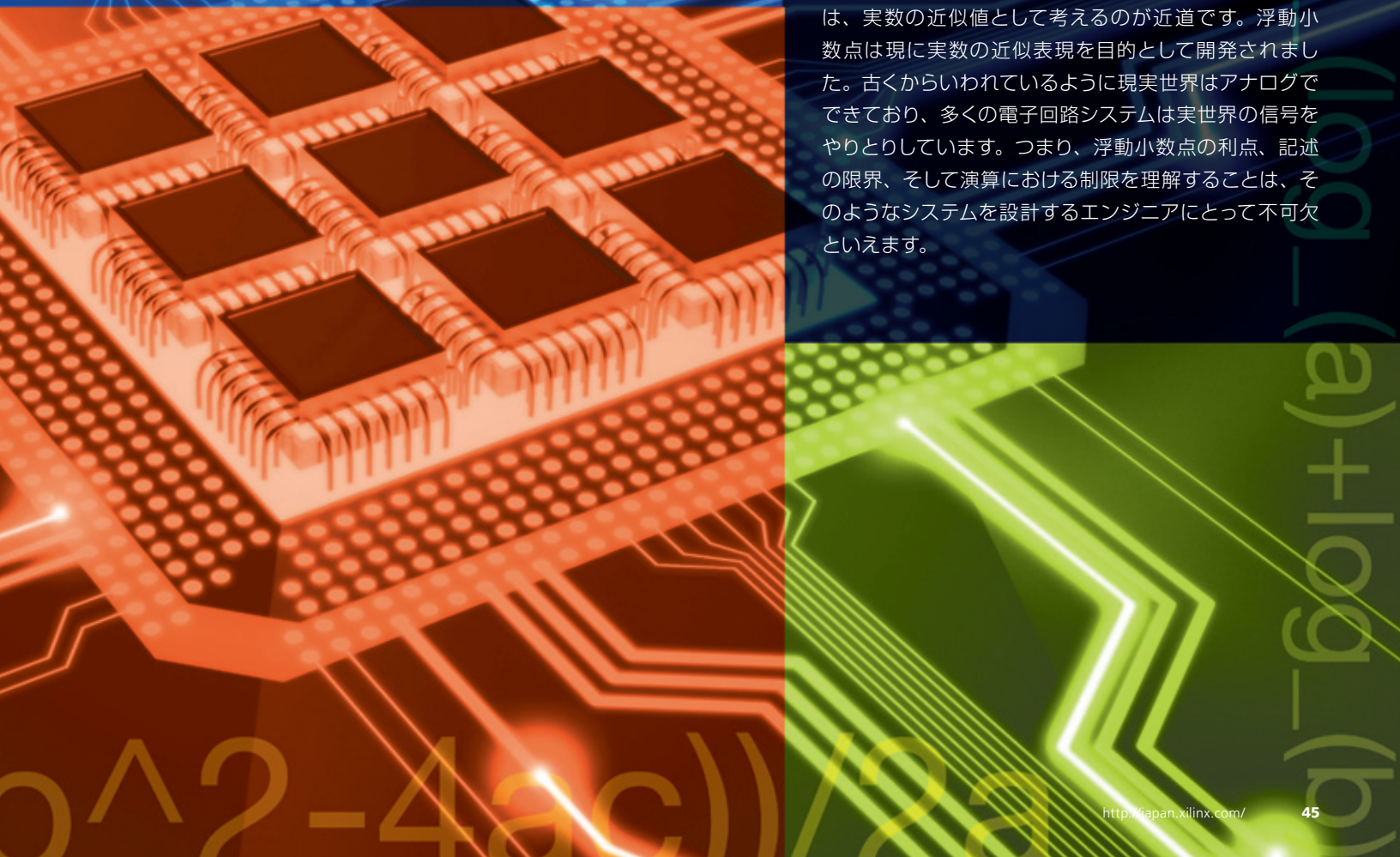
Tips and Tricks for Better Floating-Point Calculations in Embedded Systems

組み込みシステムにおける浮動小数点演算の活用のヒント

FPGA や FPGA ベースの組み込みプロセッサを使用して、整数演算の法則に従わない浮動小数点演算に対応するシステムを構築

Sharad Sinha
PhD Candidate
Nanyang Technological University, Singapore
sharad_sinha@pmail.ntu.edu.sg

エンジニアの多くは、ソフトウェア処理では演算に時間がかかり、ハードウェア実装では多くのリソースが必要になるといった理由からか、浮動小数点演算を敬遠しがちです。浮動小数点数およびその利点を理解するには、実数の近似値として考えるのが近道です。浮動小数点は現に実数の近似表現を目的として開発されました。古くからいわれているように現実世界はアナログでできており、多くの電子回路システムは実世界の信号をやりとりしています。つまり、浮動小数点の利点、記述の限界、そして演算における制限を理解することは、そのようなシステムを設計するエンジニアにとって不可欠といえます。



浮動小数点演算を詳しくみていきましょう。浮動小数点演算が整数演算ほど単純ではないことを説明するために、いくつかの計算例を示します。浮動小数点データを扱うシステムを設計する場合は、これらの点を考慮しなければなりません。たとえば、きわめて小さい浮動小数点数を操作するには対数を使用するというのが、1 つの重要なテクニックです。ここでは浮動小数点演算のいくつかの性質を紹介すると共に、設計上における課題を取り上げます。なお、この記事の最後に記載した参考文献に、より詳しい説明が述べられています。

インプリメンテーションに関しては、ザイリンクス LogiCORE ™ IP Floating-Point Operator を使用して、RTL デザインに浮動小数点演算子を生成し、インスタンシエートします。FPGA のエンベデッド プロセッサを利用して浮動小数点ソフトウェア システムを構築する場合は、Virtex-5 FXT の PowerPC 440 エンベデッド プロセッサ用に提供されている LogicCORE IP Virtex®-5 Auxiliary Processor Unit (APU) Floating Point Unit (FPU) を使用できます。

浮動小数点演算の基本

IEEE 754-2008 は浮動小数点演算に関する最新の IEEE 規格です [1]。 この規格は IEEE 754-1985 および IEEE 754-1987 の両バージョンを置き換えるもので、定義されている浮動小数点データの表記は次のとおりです。

- 符号付きゼロ : +0、-0
- 非ゼロの浮動小数点数は (-1)^s x b^e x m と記述する。ここで、

  s は +1 または +1、つまり数値が正または負であることを示し、

  b は基数 (2 または 10)、

  e は指数、

  m は形式 d0.d1d2-------dp-1 の数値で、ここで di は基数 2 のときは 0 または 1、基数 10 のときは 0 から 9 までのいずれか。なお小数点は d0 の直後に配置される。
- 符号付き無限大 : + ∞、- ∞
- 非数 (NaN) は 2 つの種類がある。
  qNaN (クワイエット NaN) および
  sNaN (シグナリング NaN)

項 d0.d1d2-------dp-1 は仮数と呼ばれ、e は指数と呼ばれます。仮数は p 桁で構成され、ここで p は表現の精度です。IEEE 754-2008 では 5 種類の基本的な表現形式が定められていて、3 種類が基数 2、2 種類が基数 10 です。また、拡張形式も規定されています。IEEE 754-1985 で規定されていた単精度数と倍精度数は、現在はそれぞれ binary32 と binary64 と呼ばれています。各形式において、指数の最小は emin、最大は emax です。

形式で表現できる有限値の範囲は、基数 (b)、精度 (p)、および emax で決定します。IEEE 754-2008 では emin を 1-emax に常に等しいと定義しています。

- m は範囲 0 から b^p-1 までの整数

  たとえば、p = 10 かつ b = 2 のとき、m は 0 から 1023 まで
- 与えられた数値に対して、e は次を満足しなければならない

  1-emax <= e+p -1 <= emax

  たとえば、p = 24、emax = +127 のとき、e は -126 から 104 まで

浮動小数点表現は必ずしも一意ではありません。たとえば、0.02 x 10^1 と 2.00 x 10^(-1) はいずれも同じ実数 0.2 を表しています。先頭桁、つまり d0 がゼロのとき、数値は正規化されているといいます。また、ある実数に対して浮動小数点表現が存在しない場合もあります。たとえば、数値 0.1 は 10 進表現では正確に表現できますが、2 進表現では小数点以下に 0011 のパターンが無限に続きます。したがって、2 進表現の浮動小数点形式で 0.1 を正確に表現することは困難です。表 1 に、IEEE 754-2008 の 5 種類の基本形式を示します。

浮動小数点演算の誤差

一定のビット数の中で実数を無数に表現しなければならないため、浮動小数点演算には丸め誤差がつきものです。このため、無限精度で計算したときを基準に、浮動小数点演算の結果がどれぐらい離れているかを示す指標が必要です。ここで、b = 10 かつ p = 4 の浮動小数点形式を考えてみます。実際の数値を .0123456 としたとき、浮動小数点形式では 1.234 x10^(-2) と表現されます。この表現には最終桁の単位 (ulp) に .56 (=0.0123456 - 0.01234) の誤差があることは明らかです。また、浮動小数点計算の結果が 4.567 x 10^(-2) で、無限精度で計算したときの結果が 4.567895 の場合は、結果は .895ulp だけ離れていることになります。

最終桁の単位はこのように計算誤差を規定する方法の 1 つです。浮動小数点数で実数を近似したときに発生する誤差を測定する指標として、相対誤差も使用されます。相対誤差は実数と浮動小数点表現との差を実数で除した値として定義されます。4.567895 を 4.567 x 10^(-2) と表現した場合の相対誤差は .000895/4.567895≈.00019 です。IEEE 規格はいずれの浮動小数点計算の結果も 0.5ulp 内で正確でなければならないと規定しています。

浮動小数点システムの設計

数値アプリケーション向けデザインを設計する際、入力データまたは定数が実数になり得るかどうかを把握しなければなりません。実数になり得る場合は、設計を進める過程で考慮する事項がいくつかあります。データセットから入力される数値を浮動小数点表現にしたときに、値が浮動小数点にきわめて近い場合、きわめて大き

パラメーター	2 進形式 (基数 b=2)			10 進形式 (基数 b=10)	
	binary32	binary64	binary128	decimal64	decimal128
P、桁数	24	53	113	16	34
emax	+127	+1023	+16383	+384	+6144

表 1 - IEEE 754-2008 が定める 5 種類の基本形式

い場合、またはきわめて小さい場合について検討します。例として二次方程式の根を考えてみます。2 つの根 α と β は次の式で表されます。

α=(-b+$\sqrt{}$(b^2-4ac))/2a ; β= (-b-$\sqrt{}$(b^2-4ac))/2a

b^2 および 4ac は明らかに、それぞれが浮動小数点のときに丸め誤差が生じる可能性があります。b^2 >> 4ac のとき、$\sqrt{}$(b^2) = ¦b¦ です。また、b > 0 のとき、β はなんらかの結果を与えますが、α は分子の項の桁落ちによりゼロになります。一方で、b < 0 のとき、β はゼロになりますが、α はなんらかの結果を与えます。浮動小数点計算によって根が意図せずにゼロになるため、これは誤差を含んだデザインと見なされ、デザインがある程度進んだ段階で使用すると、誤った結果を生み出すシステムが構築されてしまいます。

誤差の原因となるこのような桁落ちを防ぐには、α と β の分子と分母に -b-$\sqrt{}$(b^2-4ac) を乗じます。すると式は次のようになります。

α' = 2c/(-b-$\sqrt{}$(b^2-4ac)) ; β' = 2c/(-b+$\sqrt{}$(b^2-4ac))

ここで、b>0 のときは α' と β を使用して 2 個の根の値を求め、b<0 のときは α と β' を使用して 2 個の根の値を求めなければなりません。デザインによってこのような区別をするのは困難です。その理由は、データセットが整数であれば通常の等式から正しい結果が得られる一方で、浮動小数点のデータでは正しくない結果を返すことになるからです。

三角形の面積の計算でも興味深い現象が起こります。三角形の面積を求める Heron の公式を使用して、各辺の長さが既知のとき、三角形の面積 A は次の式で与えられます。

A= $\sqrt{}$(s(s-a)(s-b)(s-c))

ここで、a、b、c は三角形の各辺の長さで、s=(a+b+c)/2 です。

扁平な三角形の場合、つまり 2 辺の長さの和が 1 辺の長さとほぼ等しく a≈b+c のとき、s≈a となります。(s-a) の項でほぼ同じ 2 つの値が減算されるため、(s-a) はほぼゼロに等しくなります。あるいは、s だけが丸め誤差を持っている場合、または a だけが丸め誤差を持っている場合は大きな ulp 誤差が生じます。そこで Heron の公式を、数学者でありコンピューター サイエンティストでもある William Kahan 氏が 1986 年に提唱した次のような式 [2] に書き換えてみます。

A=($\sqrt{}$((a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))))/4 ;a≥b≥c

Heron と Kahan のそれぞれの式を使用して同じ三角形の面積を求めてみましょう。ここで、a = 11.0、b = 9.0、c = 2.04 とします。正確な s は 11.02 で、正確な A は 1.9994918 です。s の計算値はわずか 3ulp の誤差しか含まない 11.05 となりますが、Heron の公式を使用すると A は 3.1945189 となり、きわめて大きな ulp 誤差が生じていることがわかります。一方、Kahan の式を使用して A を求めると 1.9994918 が得られ、小数点以下 7 桁までが正確です。

小さな浮動小数点数と大きな浮動小数点数の取り扱い

きわめて小さい浮動小数点数を乗算すると結果はより小さくなります。これはシステムのアンダーフローを招き、最終結果は誤った値となってしまうでしょう。数理言語学や機械学習のような確率を扱う分野では特に深刻な問題です。確率は 0 から 1 の範囲の値しか取り得ません。整数を与えたときに正しい値を返す式にきわめて低い確率値を与えると、アンダーフローを引き起こす場合があります。このようなアンダーフローを防ぐには累乗と対数を使用します。

きわめて小さい値を効率的に乗算するには次の方法に従います。

a x b=exp (log (a)+log (b))

ここで、exp の底と log の底は同じでなければなりません。ただし、a と b の両方が確率値の場合、両者は 0 から 1 までの範囲にあるため、その対数は - ∞から 0 までの範囲となります。正の確率のみを扱う場合は、確率の対数に -1 を乗じ、結果を適切に解釈してください。対数で処理するもう 1 つの利点は実行速度にあり、乗算に比べて加算の方が高速に処理できます。

また、きわめて大きい値を連続して乗算しなければならない場合もあります。たとえば、

$$k = \prod_{i=1}^{v} m_i$$

場合によっては k は簡単にオーバーフローしてしまうでしょう。このような課題に対処するには、次のように k を計算します。

$$k = \exp\left(\sum_{i=1}^{v} \log\left(m_i\right)\right)$$

計算自体はやや難しくなりますが、見逃しやすい誤った結果を生まないことが重要であれば、デザイン側で対策することは決して大きな代償ではありません。

注意を要する浮動小数点演算の例

浮動小数点演算は整数が従う演算の法則に必ずしも準じません。加算と乗算に関連する結合法則は浮動小数点数には適用されず、分配法則も適用されません。それには 2 つの理由があります。まず、桁が変化するとアンダーフローまたはオーバーフローが生じること。もう 1 つが丸め誤差の問題です。浮動小数点数 a、b、c を対象にいくつかの例を挙げてみましょう。

例 1 : a x(b x c) ≠ (a x b) xc

式の右辺 (RHS) で a と b を乗算したときにオーバーフローまたはアンダーフローが起こることがあります。また、a と b の積を最初に計算するため、左辺 (LHS) での b と c の積と丸め誤差が異なる可能性があり、両辺から異なる結果が導かれます。

例 2 : a=(b-c)+ c は a=b とは等価ではない

右辺の (b-c) は浮動小数点表現が互いに近ければゼロになる場合があります。その場合、式は a = c となってしまいます。(b-c) での丸め誤差によっては、b とは異なる結果が導き出される可能性があります。

例 3 : a=b+b x c は a=b x(1.0+c) とは等価ではない

丸め誤差の影響で、(1.0+c) は最初の表現とは違う値に変化する可能性があります。

例 4 : b=c/10.0 は b=cx0.1 とは等価ではない

ここでも c/10.0 と cx0.1 の丸め誤差が異なる可能性があり、結果として両者の b に違いが生じます。また、基数 2 で 10.0 は正確に表現できますが、0.1 は正確には表現できません。0.1 は 2 進表現では 0011 のパターンが無限に繰り返されます。特定の基数にて有限的に表現できない数値はその基数における非正確数と呼ばれ、一方、表現できる数値は正確数と呼ばれます。

c が常に正確数として記述できると仮定して、一方は正確数で除算し、もう一方は非正確数を乗算した場合に、両者の結果は異なると考えられます。実際には c 自身は過去の計算に依存して正確数または非正確数のいずれかを取り得るため、上述の 2 つの式の等価関係は成立しません。

例 5 : x/x=1.0

x がゼロ、無限大、あるいは非数になり得る場合はこの式は成り立ちません。負数の平方根や、ゼロを別のゼロで除した場合など、無効な数値演算が実行されると NaN が発生します。NaN と判定された場合にどのような処理を行うかはプログラマーの責任です。IEEE 754-2008 では NaN の表現として、指数を emax + 1 とし、仮数を非ゼロとした浮動小数点数を使用することに注意してください。これを利用してシステム固有の情報を仮数部に書き込めるため、NaN は一意にはならず、さまざまな NaN が存在することになります。このように、浮動小数点コードを扱うときに、x がゼロ、無限大、あるいは NaN のいずれでもないことが確定されない限り、x /x を 1.0 で置き換えないでください。

例 6 : x-x=0.0

x が無限大または NaN のとき、この等式は成り立ちません。したがって、x の値が確定できない場合は x-x を 0.0 で置き換えないでください。プログラムの最適化手順においてコンパイラでこのような置き換えがインプリメントされる場合がありますが、ここに示すように、そのような最適化は誤った結果を導くことがあります。

例 7 : x+0=x

x が -0 のときはこの等式は成り立ちません (+0 と -0 が存在する)。IEEE 754-2008 では (-0)+(+0)=+0 がデフォルトの丸め方向、すなわち偶数丸めとなっているためです。

IEEE 規格では答えの符号を識別できるように +0 と -0 が定義されています。たとえば、4 x(+0)= +0 や (+0)/(-4)=-0 などです。比較目的では +0 = -0 であり、ゆえに、if(x = 0) のような命令文で +0 と -0 を区別する必要はありません。

より詳しく学ぶ

Kahan 氏の論文のほかに、浮動小数点データに関して知識を得るには、David Goldberg 氏の「What Every Computer Scientist Should Know About Floating-Point Arithmetic」(Revision A) (Mountain View, Calif.、1992 年 6 月) が役立ちます。また、IEEE 754-2008 の詳細を説明する参考文献に加え、浮動小数点デザインに影響を与える ISO/IEC 9899:201x などの規格が公開されています [3]。

航空宇宙や安全性が重要視される分野でのエンベデッド システムにおけるエレクトロニクスは、きわめて正確で、かつ誤差のないハードウェアおよびソフトウェアを必要としています。これらハードウェアの構成要素は大量の浮動小数点データを扱わなければならないため、浮動小数点演算と整数演算は同じではないという点を考慮してシステムを構築することが重要です。

参考文献

1. IEEE Std 754-2008, IEEE Standard for Floating Point Arithmetic
2. W. Kahan, "Calculating Area and Angle of a Needle-like Triangle," unpublished. Draft available at http://www.cs.berkeley.edu/~wkahan/
3. N1548, ISO/IEC 9899:201x, Programming languages - C

Dual-MicroBlaze Xilkernel System Eyes Automotive Apps

オートモーティブ アプリケーションをにらんだデュアル MicroBlaze Xilkernel システム

密結合されたマスター / スレーブ マルチプロセッサ システムのインプリメンテーションの鍵を握るメモリ セグメンテーションとデータ交換の戦略

Heiko Wilken
Graduate Student
Hamburg University
of Applied Sciences, Germany
wilken.heiko@gmail.com

Marco Kirschke
Graduate Student
Hamburg University
of Applied Sciences, Germany
kirschke.marco@gmail.com

Bernd Schwarz
Professor Dr.-Ing.
Hamburg University
of Applied Sciences, Germany
schwarz@informatik.haw-hamburg.de

ソフトコア プロセッサ ベースのマルチプロセッサ システムオンチップ (MPSoC) デザインは、エンベデッド システムの構築という観点から、その重要性が高まりつつあります。複数のプロセッサ間でデータを整合させるために、メモリ領域のアーキテクチャや管理には設計工数の大部分が費やされます。今回、筆者らはオートモーティブ アプリケーションのプロトタイプとして、Xilkernel 対応のデュアル Microblaze ™ システムの構築に着手しました。Xilkernel とは、ザイリンクスのエンベデッド プロセッサ用のカーネルで、2 つの MicroBlaze が両方共これに対応しています。

自律走行車で使用するステアリング プラットフォームのプロトタイプを構築するにあたり、ザイリンクスの仕様、リファレンス デザイン、デュアルプロセッサ デザイン スイート [1] を参照し、一般的な設計のワークフローとプロセスを検討しました。[2] 今回のデザインでは、衝突回避、[3] 車線検出、[4] 自動駐車などの複数の機能が組み合わされています。Xilkernel のインプリメンテーションは、QNX ベースの衝突回避アプリケーションを再利用および分割するために、Posix スレッドなどのライブラリ関数を提供します。

密結合

まったく同じコンフィギュレーションの MicroBlaze プロセッサを 2 つ緊密に結合させたプロセッサ システムを構築しました。各プロセッサは、ハーバード アーキテクチャによって 16KB ローカル BRAM に接続されています (図 1)。さらに、コードの格納とデータ交換用に、128MB の DDR2 メモリ モジュールにも接続されています。MicroBlaze の共有メモリへのアクセスは、Xilinx Platform Studio (XPS) Mutex コアによって同期が取られています。両プロセッサは、マスター / スレーブ方式およびパイプライン化されたプロセッサの設計概念に基づく開発をサポートする、1 組の Fast Simplex Links (FSL) を介して直接通信することもできます。接続するプロセッサが 2 つの MicroBlaze のみであれば、XPS Mailbox よりも FSL インターコネクトを使用した方が、リソースの使用量を減らすだけでなく性能も向上できます。

2 つのタイマー ペリフェラルが Xilkernel のスケジューリング インプリメンテーションに対応する割り込みソースを生成します。衝突回避システムの障害物検出機能のインプリメンテーションに備え、開発の第 1 段階では UART インターフェイスを介してレーザー スキャナーをシステムに接続しました。このシステムでは、直接接続のタイマー割り込みではなく、割り込みコントローラーを備えた Xilkernel を採用したため、ボード サポート パッケージ (BSP) の設定に Xilkernel 関連のパラメーターである sysintc_spec をシステムに適合するように変更しました。[5]

デバッグおよび microblaze_1 の JTAG UART を介して STDIO を出力する目的で、両方の MicroBlaze を MicroBlaze Debug Module に接続しました。このシステムについては、Spartan®-3A DSP 1800A FPGA を搭載した XtremeDSP ™ スタータ プラットフォームおよび Virtex®-6 XC6VLX240T FPGA を搭載した ML605 評価ボードを用いてインプリメンテーションを行い、それらを解析しました。

メモリおよびキャッシュのアーキテクチャ

マルチポート メモリ コントローラー (MPMC) が、2 つの MicroBlaze を Xilinx Cache Link (XCL) を介して外部の DDR2

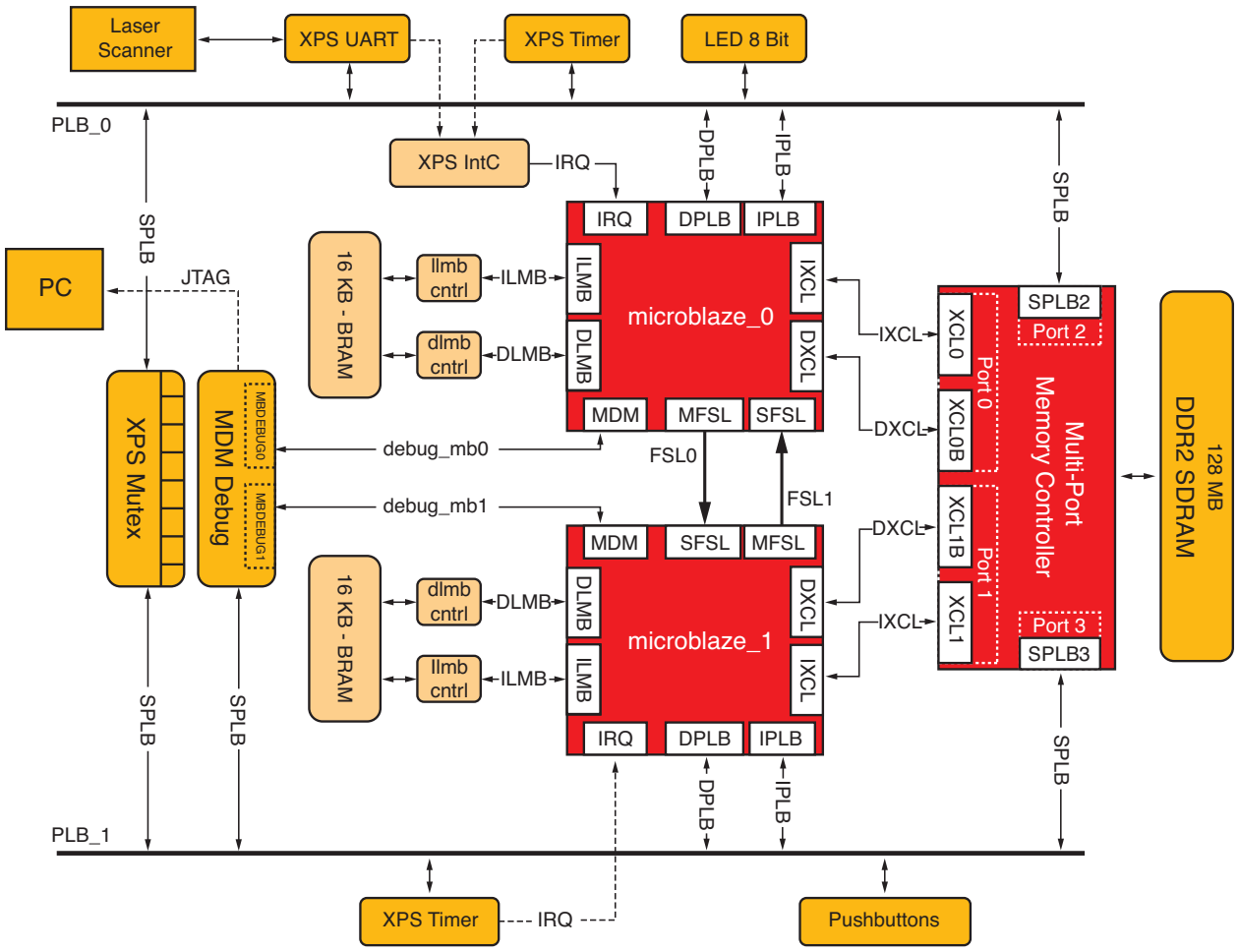


図 1 - XtremeDSP スタータ プラットフォームのデュアル MicroBlaze-Xilkernel コンフィギュレーション

メモリに接続し、ダイレクト メモリ アクセス (DMA) の割り当てを行います。[6] このように、同じメモリ アーキテクチャを採用していることから、各プロセッサのメモリのレイテンシとアクセス方法は同一のものとなります。両 MicroBlaze は、それぞれ 8KB の命令キャッシュと 8KB のデータ キャッシュを備えており、1 つの MPMC PIM に接続されます。この配線方式により、1 つの MPMC に最大 8 個の MicroBlaze を関連付けることができます。

MPMC と 外部メモリ間のデータ フローや命令フェッチのボトルネックは、MPMC のアドレス バスおよびデータ バス接続がシステムに 1 つしか存在しないために発生します。しかし、内部のラウンドロビン アービトレーションによって、すべての要求が確実にメモリに順次アクセスされます。さらに、キャッシュ対象外の共有メモリ領域にアクセスするために、MicroBlaze はプロセッサ ローカル バス (PLB) 経由による MPMC 接続も備えています。

マルチプロセッサ環境では、データ キャッシュのコヒーレンシを確保することが重要です。このため、外部メモリのアドレス範囲を 3 つのパーティションに分割しました (図 2)。各 MicroBlaze には専用のアドレス空間を占有させ、XCL はこの専用領域に格納されたデータのみをキャッシュします。第 3 のアドレス空間は、これらの専用領域とは別の領域を提供します。この領域は、PLB を介したアクセスが可能で、キャッシュされない共有データの交換に使用します。ソフトウェア エンジニアは、リンカー スクリプト内の定義済み変数によって、この領域のベース アドレスへのポインターを取得できます。キャッシュは、ライトスルー ストラテジに従って構成しました。ライトバックによるキャッシュ構成では、複数の書き込みアクセスの処理時にレイテンシが増加するからです。

リンカー スクリプトの構成

リンカー スクリプトは、ハードウェアの設計情報、対応するボード サポート パッケージ、およびソフトウェア アプリケーションの設定に基づいて、プロセッサ ベース システムのメモリ分割を定義します。XPS ツールは、各 MicroBlaze のリセット、割り込み、例外のベクターに対して、固定された領域を割り当てます。この領域は、プロセッサのブロック RAM に格納されます。一般的なアプローチでは、.heap や .stack 領域もローカル BRAM に格納し、スレッド実行時のローカル変数へのアクセスを高速化します。

リアルタイム オペレーティング システム (RTOS) のライブラリ関数については、.text および .data 領域が BRAM に格納するには大きくなりすぎるため、外部メモリに配置しました。しかし、この方法はマルチプロセッサ システムにおいて複数のプロセッサが同じメモリ領域のアドレスを指定した場合に事態を複雑化します。このため、開発者はメモリ分割を手動で適合させる必要があります。前述

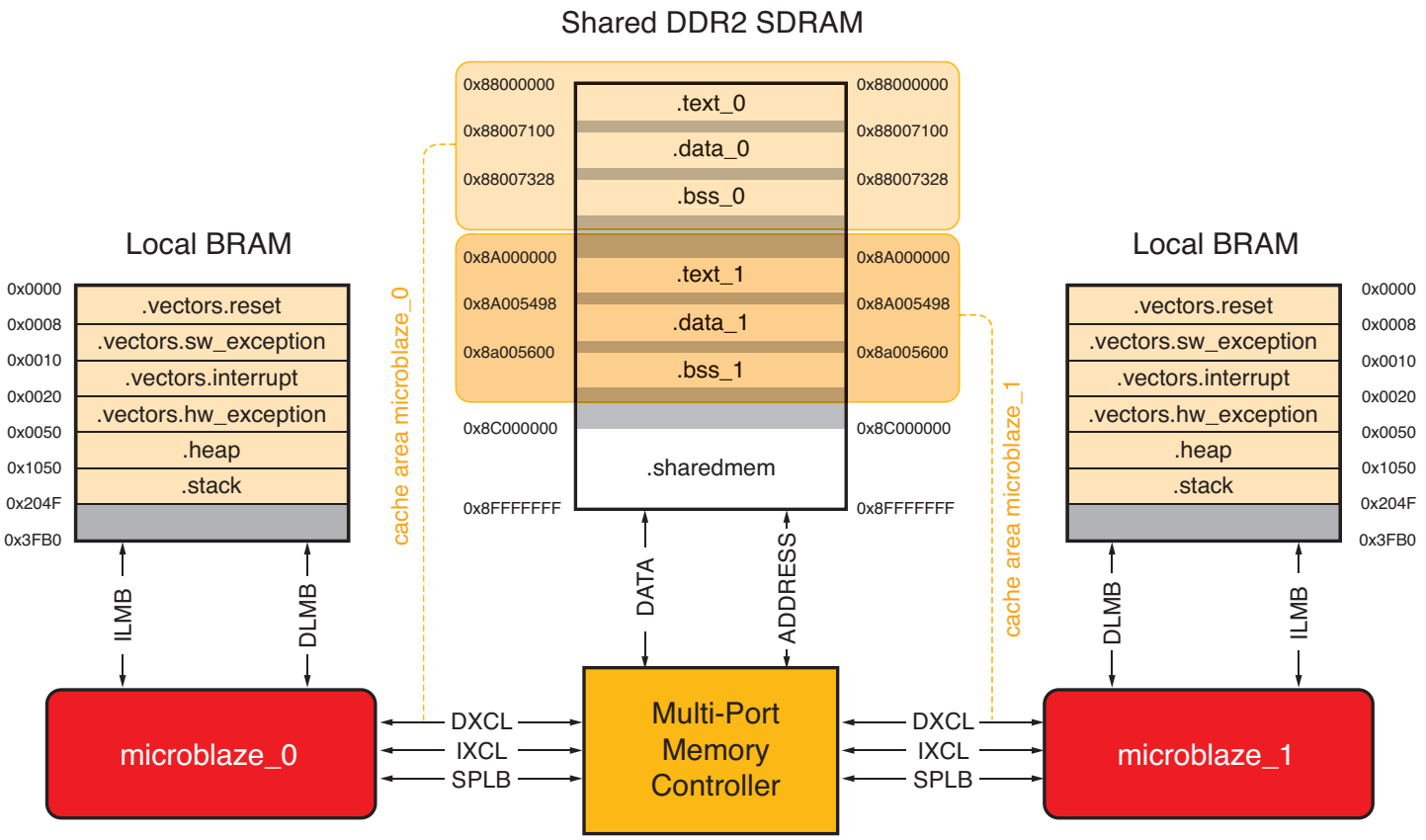


図 2 - DMX システムのメモリ マップ

のとおり、両方のプロセッサは、それぞれ専用のメモリ領域のデータのみをキャッシュし、コードおよびデータ セクションを超えた定義済み領域へのアクセスを共有します。この共有メモリ領域への同期アクセスは PLB 経由で XPS Mutex を要求することで処理し、MicroBlaze 間のデータ交換に使用します。マスター / スレーブ方式またはパイプライン化されたプロセッサ システム内の直接通信と同期には、FSL を採用しました。これにより、共有メモリへのアクセス時に複数のメモリ バスの割り当てによって生じるアクセス レイテンシの増大を防止できます。

最初のリンカー スクリプトは、SDK に含まれる Linker Script Generator によって作成しました。このスクリプトは、ヒープとスタックのサイズ、メモリ領域、各セクションに対するアドレスを定義した、3 つのパーティションから構成されるテキスト ファイルです。次に示すコードは、デュアル MicroBlaze Xilkernel システムの microblaze_0 用リンカー スクリプトです。コードおよびデータ セクションのメモリ領域のサイズを変更し、共有メモリ領域向けに新たなセクションを追加しました。これに応じて、microblaze_1 のリンカー スクリプトについても、コードおよびデータ セクションのメモリ領域の開始アドレスを 0x8A000000 に設定し、共有メモリ用に同様のセクションを追加します。

```
/* Define heap and stack sizes for microblaze_0 */

_STACK_SIZE = DEFINED(_STACK_SIZE) ? _STACK_SIZE : 0x1000;
_HEAP_SIZE = DEFINED(_HEAP_SIZE) ? _HEAP_SIZE : 0x1000;

/* Define memories in the system for microblaze_0 */
MEMORY
{
  ilmb_cntlr_dlmb_cntlr :
ORIGIN = 0x00000050, LENGTH = 0x00003FB0
  /* resize the memory region for code and data sections*/
  DDR2_SDRAM_MPMC_BASEADDR :
ORIGIN =  0x88000000, LENGTH = 0x02000000
  /* new partition for shared memory added*/
  DDR2_SDRAM_SHARED_BASEADDR :
ORIGIN =  0x8C000000, LENGTH = 0x04000000
}
[...]
/* Define the sections, and where they are mapped in memory */
SECTIONS
{
[...]
sharedmem : {
   __sharedmem_start = .;
   *(sharedmem)
   __sharedmem_end = .;
} > DDR2_SDRAM_SHARED_BASEADDR
[...]
}
```

MPSoC のダウンロードおよびデバッグ オプション

SDK 環境では、マルチプロセッサ システムのソフトウェアをダウンロード、実行、デバッグするために、2 つのアプローチが用意されています。これに対して、Eclipse ベースのツールは、ソフトウェア プロジェクト用にダウンロードと STDIO ビヘイビアーのコンフィギュレーションを提供します。また、これらの作業の実行には Xilinx Microprocessor Debug (XMD) コンソールも使用できます。

さらに、SDK は FPGA をビットストリームや .bmm ファイルでリコンフィギュレーションします。リンカー スクリプト内に定義されたセクションに従い、[Program FPGA] ダイアログ ボックスによって、必要に応じてソフトウェア セクション全体を特定のプロセッサに関連付けられた BRAM に直接転送します。既に述べたように、マルチプロセッサ システムでは一般的にこれらのセクションを外部メモリに配置します。Xilkernel ライブラリ 関数を使用するアプリケーションによって .elf ファイルが拡大するためです。この場合、[Program FPGA] ダイアログ ボックスの [Software Configuration] 設定でブートループを選択し、後続のソフトウェア ダウンロードが実行されるようにします。

SDK 環境

SDK には、[Project Explorer] ビューによるソフトウェア プロジェクトの選択や、[Run] メニューの [Run configurations...] を開くために、[Run] および [Debug] の設定があります。システム内のすべてのプロセッサに [Run] および [Debug] コンフィギュレーションが対応付けられています。最初のインスタンスで、コンフィギュレーション ダイアログ ボックスのメイン タブからプロジェクトの .elf ファイルを選択します。[Device Initialization] タブのリセット タイプには [Reset Processor Only] を選択し、リセット時にシステム全体がリセットされないようにします。

さらに、[STDIO Connection] タブの設定により、STDIO が SDK コンソールに直接出力されるようにします。個々のプロセッサの [Run] および [Debug] コンフィギュレーションをグループとして束ねることで、すべてのプロセッサが同時にソフトウェアの実行を開始します。最後に、開発者は SDK の Eclipse ベースの [Debug] ビューを使用し、ブレークポイントを挿入したり、変数およびメモリの割り当てに関する情報を入手することができます。

XMD コンソール

今回、最も重要な XMD コンソールのコマンドは、マルチプロセッサ システムの使用に関連するものです。コマンドのより詳細なリストは、XMD コンソールからヘルプ コマンドを投入することで、直接使用できるようになります。マルチプロセッサ システムのダウンロードやデバッグに XMD コンソールを使用するには、次のコマンドを投入します。

```
connect mb mdm –debugdevice cpunr <mb id>
```

このコマンドは、プロセッサを MicroBlaze Debug Module (MDM) 経由で XMD コンソールに接続し、さらにパラメーター <mb id> によって、各プロセッサの識別子を定義します。すべてのプロセッサの接続が完了したら、次のコマンドを実行します。

	Spartan-3A DSP 1800A		Virtex-6 LX240T	
	使用数	使用率	使用数	使用率
スライス DFF	7,674	23%	10,891	3%
LUT	9,790	29%	8,890	5%
BRAM	53	63%	49	11%
DSP48	6	7%	6	1%

表 1 – デュアル MicroBlaze Xilkernel システムのリソース使用率

```
target <trg no>
```

このコマンドは、現在のターゲット接続を処理する MicroBlaze を選択します。target コマンドは、接続されているターゲットの一覧を表示します。特定のプロセッサ ターゲットに .elf ファイルをダウンロードするには、次の dow コマンドでファイルのフル パスと名前を入力します。

```
dow <elf file>
```

選択したターゲットのソフトウェアは、run コマンドによって起動するか、stp コマンドによって 1 行ずつ実行できます。XMD コンソールのレポートに接続する PLB インターフェイスが MDM にある場合は、次のコマンドにより JTAG UART から出力させることができます。

```
read_uart start
```

JTAG UART サーバーは、次のコマンドによってポートを定義するだけでこれらの出力を受信できます。この機能を使用すれば、ローカル ホスト上のターミナル プログラム (HTerm や Tera Term など) で出力を使用できるようになります。

```
terminal jtag_uart_server <port>
```

パイプライン化されたシステム

デュアル プロセッサ プラットフォームの段階的なデザイン インプリメンテーションで構築した今回のシステムは、2 つの Xilkernel 対応 MicroBlaze プロセッサからなり、両プロセッサは FSL による 2 つの通信インターフェイスを共有し、共有メモリにアクセスできます。各プロセッサには、外部 DDR2 SDRAM モジュール内の互いに独立したメモリ領域に対して動作する、命令キャッシュとデータ キャッシュが設定されています。さらに、データの整合性を維持するために、XPS Mutex モジュールが共有メモリへのアクセスを同期させます。2 つの MicroBlaze のリンカー スクリプトは、両プロセッサの命令およびデータ セクション、および 1 つの共有データ セクションとして専用メモリ領域が割り当てられるように変更しました。

表 1 に、Spartan-3A DSP 1800A および Virtex-6 XC6VLX240T FPGA の両方について、デフォルトの合成制約を適用した場合のリソース割り当てと使用率を示します。中規模の Spartan デバイスでも、デュアル MicroBlaze システムの PLB または FSL インターフェイスに接続される追加のアクセラレータ IP 用に十分なスライス リソースを提供できます。

これらの未使用 FPGA リソースによって、レーザー スキャナーの生データを前処理するアクセラレータ ユニットを統合できます。そして次の段階で、プロセッサへの、QNX ベースの衝突回避アプリケーションの配置とスレッドの割り当てに注力します。

障害物の識別、環境のモデル化、距離制御は、条件分岐やシーケンシャルな処理などの一連の計算を構成するため、パイプライン化されたプロセッサ システムが適していると考えます。将来的には、カメラによる車線検出、自動駐車に加えて、車両走行距離計やクルーズ コントロールなどのその他にも提案されているコンポーネントの統合を進める予定です。

参考文献

1.『デュアル プロセッサ リファレンス デザイン スイート』- XAPP996 (2008 年)、http://japan.xilinx.com/support/documentation/application_notes/xapp996.pdf

2. “SVC-Collision Avoidance Assistant,” HAW Hamburg, 2011, http://www.informatik.haw-hamburg.de/ schwarz.html

3. D. Schetler, “Automatic collision avoidance with a laser scanner-based distance control for a driverless transport system,” Master's Thesis, HAW Hamburg, 2007.

4. D. Mellert, “Modeling of video-based lane detection with the Xilinx System Generator for an SoC-platform,” Bachelor's Thesis, HAW Hamburg, 2010.

5.『Xilkernel (v5.00.a)』- UG708 (2011 年)、http://forums.xilinx.com/xlnx/attachments/xlnx/EMBEDDED/4789/1/xilkernel_v5_00_a.pdf

6.『マルチポート メモリ コントローラー (MPMC) (v6.03.a)』– DS643 (2011 年)、http://japan.xilinx.com/support/documentation/ip_documentation/mpmc/v6_05_a/j_mpmc.pdf

Optimizing FPGAs for Power: A Full-Frontal Attack

FPGA の消費電力最適化：正攻法的な取り組み

7 シリーズ FPGA の最新ファミリを
開発サイクルの各段階で
低消費電力化する設計手法

Chandra Sekar Balakrishnan
Solutions Development Engineer
Xilinx, Inc.
cbalakr@xilinx.com

デザインにとっての完璧な FPGA を探究する上で、消費電力が最大の課題となりつつあります。パワー マネージメントは、ほとんどのアプリケーションにおいて不可欠です。カードあたり、あるいはシステムあたりの最大消費電力を規定する規格もあります。このため、設計者は今まで以上に早い設計段階 (多くの場合 FPGA の選定段階) から消費電力を検討しなければならなくなりました。

FPGA の消費電力を抑えると、電源レールの低電圧化、電源供給デザインや熱管理の簡略化、電力分配プレーンの要件緩和などによってボードのデザインが簡単になります。低消費電力化には、バッテリ寿命を延ばし、システムの信頼性を高める効果もあり、より低温な環境で動作するシステムは長持ちします。

消費電力に関する課題

トランジスタは、ムーアの法則に従ってプロセス テクノロジの世代が進むにつれて微細化されています。この状況は、個々のトランジスタにおけるリーク電流の増加という副作用をもたらし、スタティック消費電力、つまり FPGA が動作していない間に引き込まれる電流の増大へとつながります。一方、FPGA のパフォーマンス向上に伴うクロック速度の高速化は、ダイナミック消費電力を増加させます。スタティック消費電力がトランジスタのリーク電流によって決定するのに対し、ダイナミック消費電力はプログラマブル ロジックや I/O のスイッチ周波数に依存します。FPGA は、これら 2 種類の消費電力を増加させながら、製品世代ごとに容量を拡大させてきたのです。ロジックが増えるということは、デバイスあたりのリーク電流と、より高速で動作するトランジスタ数が増加することを意味します。

このような課題が存在するため、設計者は設計の初期段階から電源や熱管理の問題に対して、今まで以上に注意を払う必要があります。デバイスにヒート シンクを貼り付けるだけでは、適切な解決策は得られません。むしろ、デザインに含まれるロジックに削減の余地がないかを検討すべきです。

FPGA デザインの消費電力を低減するために、設計の各段階でどのような措置を講じることができるのか、その理解に役立ついくつかの指針を見ていきましょう。設計の早期段階で、これらの課題を詳細に把握すれば、その効果が大きくなるのは明白です。

図 1 に、FPGA の選定から低消費電力デザインの手法に至るまで、さまざまな設計段階を示します。

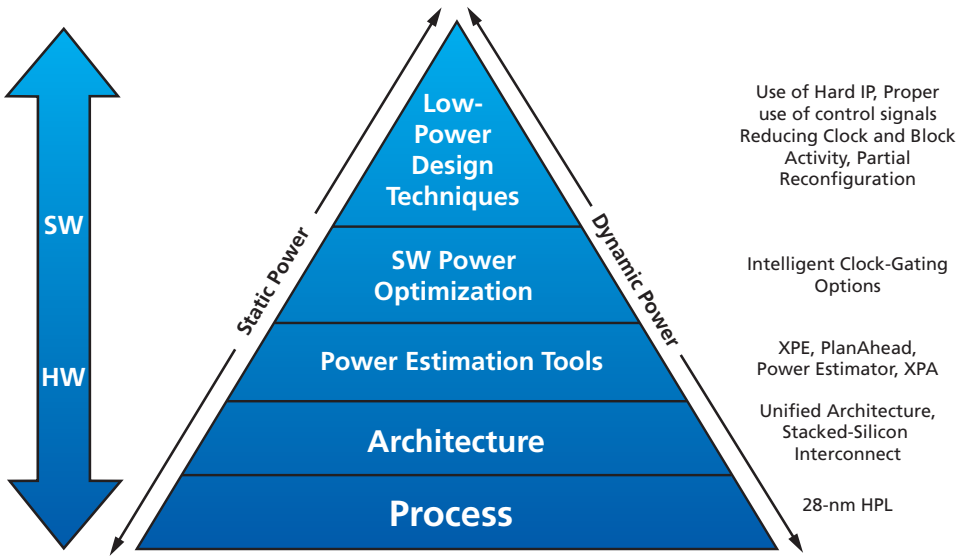


図 1 - デバイス選定に始まりツールやソフトウェアの最適化も含めた消費電力削減

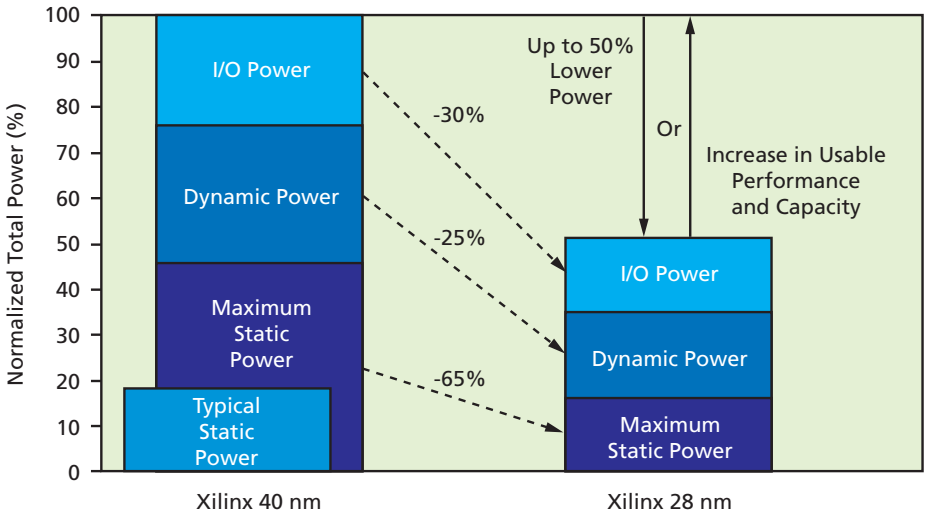


図 2 – 従来の 40nm プロセスで製造されたデバイスと比較して、消費電力の半減を実現したザイリンクス 7 シリーズ FPGA

7 シリーズのプロセス テクノロジ

FPGA の選定時にプロセス テクノロジを慎重に検討すると、デバイスのリーク電流とパフォーマンスを見極めることができます。ザイリンクスの 7 シリーズ FPGA は、FPGA がターゲットとする性能範囲において高性能モードで消費電力を半減する 28 HPL (28nm High-Performance, Low Power) プロセスを採用しています (Xcell Journal 75 & 76 合併号の「7 シリーズ FPGA で消費電力を半減」を参照)。リーク電流の小さい HPL プロセスによって製造されたデバイスを選択すると、FPGA デザインに複雑でコスト増を招くスタティック消費電力の低減策を組み込む必要がなくなります。

28 HP プロセスを採用した FPGA には 7 シリーズ FPGA を超えるパフォーマンス上のメリットはありませんが、競合他社の FPGA の中にはパフォーマンスを向上させるためにスタティック消費電力が 2 倍以上となり、リーク電流の低減に課題を抱えるものもあります。図 2 に、従来世代の 40nm FPGA デバイスに比べて総消費電力を半減させた、7 シリーズ ファミリの低消費電力化へのアプローチを示します。設計者は、開発時に大型の FPGA を選択し、続く製品ラインでより小型の FPGA に移行することもできます。小型の FPGA を選択すると、システムのコストを下げるだけでなく、消費電力も削減します。

すべての 7 シリーズ FPGA は統一されたアーキテクチャをベースに構築されています。この統一アーキテクチャによって、ザイリンクス 7 シリーズのポートフォリオに含まれるさまざまな FPGA デバイスやファミリ間でデザインを容易に移行できます。Virtex®-6 または Spartan®-6 デバイスから 7 シリーズ デバイス ファミリへのデザイン移行、または 7 シリーズ デバイス ファミリ間でデザイン移行を検討する際は、『7 シリーズ マイグレーション メソドロジ ガイド』(UG429) を参照してください。

7 シリーズ FPGA	C グレード デバイス	-2LE (1.0V)	-2LE (0.9V)
VCCINT	1.0V	1.0V	0.9V
スタティック消費電力	公称値	-45%	-55%
ダイナミック消費電力	公称値	公称値	-20%
パフォーマンス	-1, -2	-2	~ -1

表 1 – スタティック消費電力、ダイナミック消費電力、パフォーマンスの比較

ザイリンクスのスタックド シリコン インターコネクト テクノロジ

さらに大規模なシステムの場合、設計者はたいてい複数の FPGA を選択します。この種のアーキテクチャでは、各種 FPGA 間でデータを高速転送するという、細かく困難な作業が必要になることが少なくありません。そこで、ザイリンクスのスタックド シリコン インターコネクト (SSI) テクノロジを用いて作成された XC7V1500T や XC7V2000T デバイスなどの大型の 7 シリーズ FPGA を選択すれば、このような問題を回避できます。SSI テクノロジは簡単に説明すると、数万にも及ぶ配線を提供するシリコン インターポーザー上で複数のダイス間をつなぎ、大規模な単一デバイスを構成するものです。このテクノロジのメリットの 1 つは、同規模の標準的なモノリシック ダイと比較して、最大スタティック消費電力を減らせることです。

SSI テクノロジは I/O インターコネクトの消費電力も大幅に削減します。ボード上に複数の FPGA をインプリメントし、I/O とトランシーバーを用いて同等のインターフェイスを構築した場合と比べ、SSI テクノロジは I/O インターコネクトの消費電力を 100 分の 1 (帯域幅 / W) 削減する効果があります。これは、すべての接続をチップ上で行い、チップ外への信号を駆動する電力が不要となるためで、大幅な高速化と低消費電力化を可能にします。

電圧制御のために強化されたオプション

ザイリンクス 7 シリーズ FPGA には豊富な電圧制御オプションがあります。

7 シリーズ FPGA の -3 および -2L デバイスの両方には、拡張温度範囲 (0 ~ 100℃) オプション (E) が用意されています。28 HPL プロセスによって確保されたヘッドルームにより、-2LE デバイスは 1V または 0.9V 動作が可能です。これらのデバイスは、それぞれ -2L (1.0V) および -2L (0.9V) と呼ばれます。1.0V 動作の -2L デバイスは、-2I や -2C デバイスと同じスピード グレード性能を備えつつ、スタティック消

費電力が著しく低減されています。0.9V 動作の -2L デバイスでは、-1I や -1C デバイスの性能を保ちながら、スタティック消費電力に加えてダイナミック消費電力も削減されています。

0.9V 動作は、デバイス内の電圧降下だけでもスタティック消費電力を約 30% 低減します。電圧降下は性能低下につながる可能性がありますが、ザイリンクスではスピードとより厳しいリーク電流の仕様によって、これらの -2L (0.9V) デバイスをスクリーニングにかけています。これにより、ワーストケース条件下のプロセスでも、標準スピード グレード デバイスに比べて消費電力が 55% 削減されています。

-2L ファミリ デバイスを選択すれば、ダイナミック消費電力も削減されます。ダイナミック消費電力は VCCINT2 に比例するため、VCCINT を 10% 低くすると消費電力は 20% 低減されます。

消費電力概算ツール

現在、市場では、開発全体で FPGA デザインの放熱要件および電源要件を評価する支援ツールが多岐にわたって提供されています。図 3 に、FPGA 開発サイクルの各段階で利用可能なザイリンクスのツールを示します。

設計の開始時、設計前やインプリメンテント前の段階でさえ、XPower Estimator (XPE) スプレッドシート使用して消費電力を早期に評価できます。XPE はアーキテクチャの評価やデバイス選定をサポートし、アプリケーションに必要となる可能性がある、電源や熱管理向けのコンポーネントを適切に選択するのに役立ちます。

PlanAhead™ ツールは、RTL レベルでデザインの消費電力を概算します。設計者は、制約または GUI を使用してデバイスの動作環境、デザインの I/O 特性とデフォルトのアクティビティ レートを指定できます。続いて PlanAhead ツールは HDL コードを読み込み、必要なデザイン リソースを見積もり、各リソースのアクティビティの統計解析から消費電力の概算値をレポートします。PlanAhead™ ツールに含まれる RTL パワー エスティメーターは、デザインの意図に関してより詳細な情報を使用するため、XPower Estimator スプレッドシートよりも解析精度が高いものの、XPower Analyzer による配置配線後の解析と比較すると精度は落ちます。

XPower Analyzer (XPA) は、配置配線済みデザインの消費電力を解析する専用のツールです。消費電力だけでなく、指定された動作条件における熱情報も詳細に解析できる、包括的な GUI を提供します。

このツールでは、ブロック タイプ (クロック ツリー、ロジック、信号、I/O、あるいはブロック RAM や DSP ブロックなどのハード IP) ごとの表示とデザイン階層による表示の、2 つのビューを切り換えて消費電力を確認できます。これらのビューによって、消費電力を詳細に解析できます。このツールは、デザインに含まれるブロックまたはパーツの中で最も電力を消費するものを特定する、極めて効率的な方法を提供するため、電力最適化の作業にどこから着手すればよいかが明確になります。

ソフトウェアによる消費電力の最適化

ブロック RAM を使用するデザインの消費電力は、同時にアクティブになるブロック RAM ポートの数を最低限に抑えることで最適化できます。XST の -power yes オプションを用いて有効となるこの最適化は、複数のブロック RAM にまたがって分断された RAM または ROM の記述を修正します。さらに、アドレス ラインのほかにも、ポート イネーブルやライト イネーブル制御信号を調整し、デザインのタイミング制約を確実に満たしながら、各クロック サイクルでアクティブなブロック RAM のポート数を最小限に抑えます。

次に、パフォーマンスへの影響にかかわらず、最も電力効率のよいブロック RAM のマップを強制的に実行します。対象のメモリに関連するタイミング パスが重要でないことがわかっている場合は、ram_style 制約に block_power2 オプションを適用します。この方法による消費電力削減率は、15% から 75% に及びます。

XST の area 最適化モードも使用できます。このオプションは、デザインで使用されるリソースの数を最小限にします。エリア最適化を適用すると、パフォーマンスが犠牲となる可能性があるので注意が必要です。

これとは別に、高度なゲーティングと呼ばれる、アクティビティを意識した最適化方法があります。これらのアルゴリズムでは論理式が解析され、結果に影響しないソース レジスタがクロック サイクルごとに検出されます。続いて、FPGA ロジック内にある豊富なクロック イネーブル (CE) リソースを

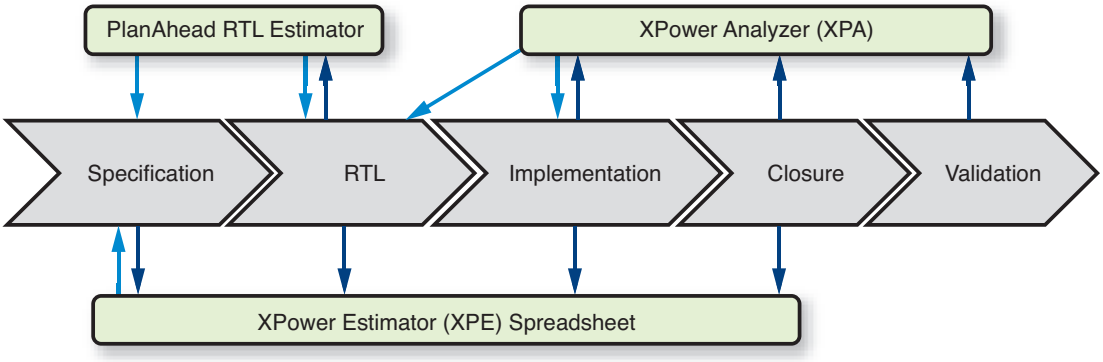


図 3 - ザイリンクスが提供する、各設計段階を対象とした消費電力の概算および解析ツール

使用し、無駄なスイッチング動作を無効にする、高精度のゲーティング信号が生成されます。この高度なクロック / データ ゲーティングは、map -power high オプションを使用して制御できます。コアの総ダイナミック消費電力を 15% 以上削減でき、ほとんどの場合、挿入されたゲーティング ロジックが原因でパフォーマンスが影響を受けることはありません。

さらに、消費電力削減を目的としたデザインで使用可能なアルゴリズムに、キャパシタンスを意識した最適化があり、主に次の 2 つの手法で行います。

- クロックの負荷をグループ化：このプロセスでは、フリップフロップや DSP ブロックなどの同期エレメントの配置が再編成され、各クロック ネットの範囲が最小限に抑えられます。クロック負荷を最少数の水平または垂直クロック スパインに沿って配置させると、クロック領域内で未使用の分岐をソフトウェアで無効にできます。これによって、クロック リソースとバッファーの要件を両方削減でき、コアのダイナミック消費電力が抑えられます。このプロセスは、map -power on オプションで制御します。

- データ負荷をグループ化：このアルゴリズムでは、パフォーマンス要件を確実に満たしながら、デザインに含まれるワイヤの全長が最短に抑えられます。ダイナミック消費電力はファンアウトおよび使用する配線構造の種類と長さに合わせて増大するため、データ負荷をグループにまとめることで消費電力を節減できます。このグループ化のアルゴリズムも、map - power on オプションで有効になり、関連ロジックを隣接して配置することで消費電力の削減を達成できます。

ISE® Design Suite には、合成、マップ、配置配線の各レベルで電力を最適化できるようにあらかじめ調整された、定義済みの目標およびストラテジが組み込まれています。これは、すべての合成制約にデフォルト以外の設定を適用するよりも優れた方法として代わりに活用できる可能性があります。ただし、このオプションを実行すると、さまざまなパスである程度の遅延時間が発生する場合があります。

最後に、ザイリンクスのインプリメンテーション ツールが未使用のトランシーバー、PLL、デジタル クロック マネージャーおよび I/O を自動的に遮断します。7 シリーズ デバイスでは、未使用ブロック RAM のパワー ゲーティング機能も追加されました。ブロック RAM のリーク電流は、デザイン内で使用中のブロックでのみ発生し、デバイス内のすべてのブロック RAM で発生するわけではありません。このため、デバイス内の電源はインスタンシエートされたブロック RAM にのみ配線し、未使用のブロック RAM では無効にします。

消費電力低減のための設計手法

FPGA デザインの消費電力の低減を目指して設計者が検討できるヒントやテクニックは数多く存在します。まず、ロジックのインプリメンテーションに CLB ではなく専用のハードウェア ブロックを使用することが選択肢として考えられます。消費電力を削減するには、デザイン内のロジックを減らす余地がないかを検討する必要があります。ロジックを削減して可能な限り小型のデバイスを使用できれば、スタティック消費電力の低減につながります。

専用ハード IP ブロックの適用は、スタティック消費電力とダイナミック消費電力の両方を低減し、同時にタイミング要件も満たすことができる、最も重要な方法の 1 つです。ハード IP は CLB ロジックによる等価コンポーネントよりもトランジスタの総数が少なくなるため、スタティック消費電力が抑えられます。

原則として、リソースはできるだけ推論する必要があります。推論されるリソースは、コードまたは制約ファイル内の属性を使用し、個別またはグループとして FPGA ファブリックやシリコン リソースへと誘導できます。ザイリンクスの CORE Generator™ ツール

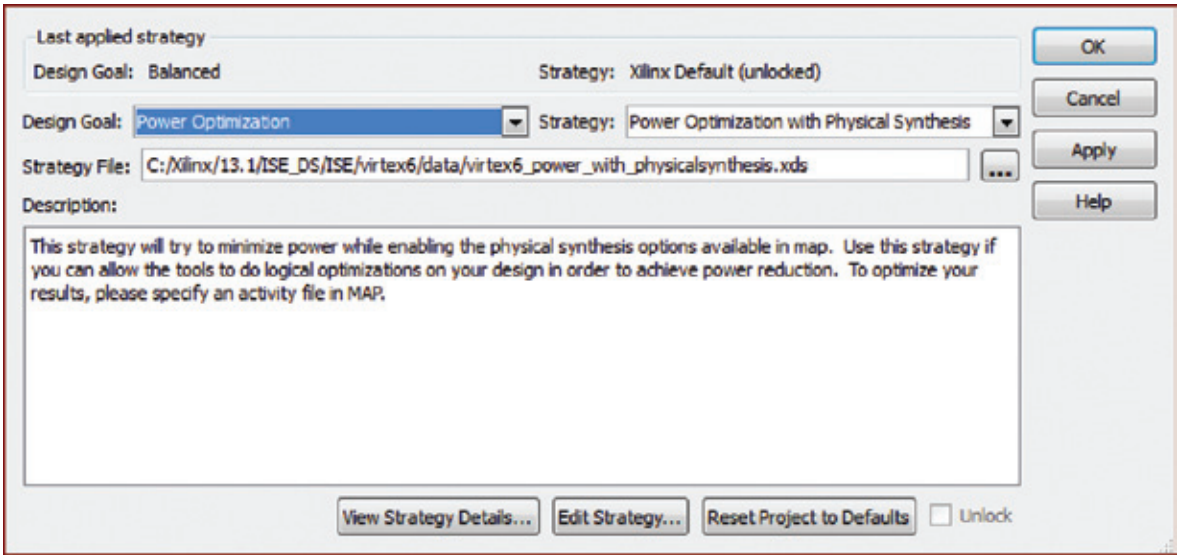


図 4 - 消費電力を最小にする「Design Goal (デザインの目標)」および「Strategy (ストラテジ)」がビルドされたザイリンクスの ISE Design Suite

を利用して専用ハードウェアをカスタマイズし、特定のリソースをインスタンシエートすることも可能です。

さらに、未使用のハード IP を不明瞭なほかのタスクに活用することもできます。DSP 48 スライスは、乗算器、加算器 / アキュムレータ、ワイド ロジック コンパレータ、シフター、パターン マッチャー、カウンターなど、多くのロジック ファンクションを提供します。ブロック RAM はステート マシン、数値演算機能、ROM、ワイド ロジック ルックアップ テーブル (LUT) として使用できます。

制御信号の最適な使用方法

制御信号 (クロック、セット、リセット、クロック イネーブルなどの同期エレメントを制御する信号) は、使用方法によってはデバイスの集積度、使用率、パフォーマンスに影響を及ぼす場合があります。次に示すガイドラインに従うことで、消費電力への影響は最小限に抑えることができます。

1 つ目のガイドラインは、レジスタまたはラッチでセットとリセットの両方を使用しないことです。ザイリンクス FPGA のフリップフロップは、非同期および同期のいずれのリセット / セット制御にも対応できますが、下位のフリップフロップのネイティブ構成では、一度に 1 つのセット、リセット、プリセット、クリアしかインプリメントできません。したがって、RTL コード内にこれらの機能を複数コーディングすると、ある条件はフリップフロップの SR ポートによってインプリメントされ、別の条件はファブリック ロジックにインプリメントされ、より多くの FPGA リソースが使用されることになります。

条件の 1 つが同期、別の条件が非同期の場合、非同期条件は SR ポートによってインプリメントされ、同期条件はファブリック ロジックにインプリメントされます。通常、複数のセット / リセット / プリセット / クリア条件は避けるべきです。また、スライス 1 つに含まれる 4 つのフリップフロップの各グループに対して、それらの SR ポートが同期または非同期であるかどうかは、そのグループに対応する属性 1 つのみで決定します。

2 つ目のガイドラインは、アクティブ High の制御信号を使用することです。レジスタの制御ポートはアクティブ High です。FPGA デザインでアクティブ Low のリセット信号の使用は推奨していません。アクティブ Low 信号は、レジスタの制御ポートを直接駆動する前に反転させる必要があるため、より多くのルックアップ テーブルを使用します。この反転動作は、LUT によって実行する必要があるため、LUT 入力を 1 つ占有します。

このためアクティブ Low の制御信号は、長時間に及ぶランタイムやデバイス使用率の低下の原因となり、ひいてはタイミングや消費電力に影響を与えます。

HDL コードやインスタンシエートされるコンポーネントでは、可能な限りアクティブ High の信号を使用してください。デザイン内の信号の極性を設定できない場合は、コードの最上位階層で信号を反転します。そうすれば、推論されるインバーターを I/O ロジック内に吸収でき、FPGA ロジックまたは配線を増やす必要がなくなり、使用率、パフォーマンス、消費電力が改善されます。

セットまたはリセットの多用を避ける

コード内に不必要なセットおよびリセットがあると、推論されるべきシフト レジスタ LUT (SRL)、LUT RAM、ブロック RAM、その他のロジック構造の推論を妨げる恐れがあります。多くの回路は自然にリセットされるように作成することができるか、そもそもリセットを必要としません。たとえば、レジスタはコンフィギュレーション完了時に自動的に初期化されるため、レジスタの初期化にのみ使用されるリセットは不要です。

不必要なセットまたはリセットを使用しないことで、デバイスの使用率が向上し、配置、パフォーマンス、消費電力を改善できます。

リセットに関する詳細は、http://issuu.com/xcelljournal/docs/xcell_journal_issue_76/44?viewMode=magazine&mode =embed を参照してください。

消費電力低減へのアプローチには、クロックとブロックの動作率について対策を講じることも必要です。BUFGMUX、BUFGCE、BUFHCE を最大限に活用して、クロック ドメイン全体をゲーティングすることで消費電力を削減できます。これらの制約により、クロック領域全体でクロックを一時停止できます。同様に、デザインの小さな領域のクロックのみを一時停止するアプリケーションでは、FPGA レジスタのクロック イネーブル ピンを使用します。

複数のクロック領域にまたがるデザインは、使用するクロック リソースが増加し、消費電力も増えます。断続的に使用されるロジックは、なるべく 1 つのクロック領域内にまとめて配置することで (図 5)、消費電力の低減効果が得られます。ツールは自動的にこのような配置を試みますが、デザインによっては手動による作業が必要になる場合があります。

データ移動を制限することでも消費電力を削減することが可能です (図 6)。FPGA 周りではオペランドを移動するのではなく、結果のみを移動してください。バスの数と配線長が抑えられ、キャパシタンスの低減、動作の高速化と消費電力の削減につながります。フロアプランに際しても、デザインのピン配置や対応するロジックの配置に注意を払う必要があります。

スタティック消費電力を低減するパーシャル リコンフィギュレーション

スタティック消費電力を削減する方法の 1 つに、小型デバイスの使用があります。パーシャル リコンフィギュレーションを用いることで FPGA を本質的に時分割でき、デザインの一部を個別に実行させることができます。デザイン全体が常に必要とされるわけではないため、使用するデバイスが小型化されます。

パーシャル リコンフィギュレーションによって、スタティック消費電力だけでなくダイナミック消費電力も低減できる可能性があります。たとえば、多くのデザインでは極めて高速な動作が要求されますが、その最大パフォーマンスが必要とされている時間の割合はとても低いものです。消費電力を節約するために、設計者は常時最大パフォーマンスで動作するデザインを設計するのではなく、パーシャル リコンフィギュレーションによってこの高性能なデザインを、同等機能を持ったより低消費電力のバージョンに切り換えることができます。システムが高パフォーマンスを要求した時点で、高性能バージョンのデザインに再度切り換えればよいのです。

この原理は I/O 規格にも適用可能で、具体的には、消費電力の大きなインターフェイスが常時必要とされない場合に有効です。LVDS は直流動作であるため、動作率にかかわらず多くの電力を消費するインターフェイスです。設計者は、パーシャル コンフィギュレーションを用いて、最大パフォーマンスが不要のときは LVDS から LVCMOS のような低消費電力の I/O 規格に切り換えることができます。そして、高速伝送が要

求された時点で LVDS に戻します。

低消費電力のデザインでは、タイミング制約を最大限に活用することも重要です。温度管理された環境での動作では、タイミングを満たすためにパーツをディレーティングできますが、そのパーツに対して規定の最大クロック レートを超える制約を課さないように注意が必要です。より高速のクロック レートを指定しても、より優れたデザインが生成されるわけではありません。通常は、リソースの共有が減るためにさらに多くのファブリック リソースが使用され、ロジック / レジスタの重複や配線が増加し、FPGA 専用機能の推論の減少を招くだけです。この結果、ダイナミック消費電力に著しい影響を及ぼします。

I/O の消費電力が総消費電力に占める割合は増加しています。特にメモリを多用するシステムでは、総消費電力の 50% が I/O に起因するデザインもあります。

プログラム可能なスルー レートおよび駆動電流によって、I/O ドライブのダイナミック消費電力が低減します。より高速な差動 I/O 機能が多くの場合好まれますが、すべてのインターフェイスでそのようなパフォーマンスが求められるわけではありません。FPGA 間の通信や、低速のメモリ インターフェイスにおいて消費電力を大幅に節約できる HSLVDCI のような規格も存在します。

ザイリンクス 7 シリーズのデバイスは、いずれもスルー レートおよび駆動電流をプログラムできます。ザイリンクスの FPGA は、トライステート化も可能にするデジタル制御インピーダンス (DCI) テクノロジをサポートします。DCI は FPGA からメモリへの書き込み時に、終端電源を必要としないため、デバイスが終端電源によって電力を消費するのは、読み出し時のみとなります。

7 シリーズのデバイスには、HSTL および SSTL 向けに、ユーザーによるプログラムが可能なリファレンス レシーバー モードが採用されています。これら 2 つのプログラマブル可能な電力モードは、I/O ごとに設定できるため、各 I/O で消費電力とパフォーマンス間のトレードオフを検討することで DC 消費電力の削減効果が得られます。

トランシーバーの消費電力に対する考え方

ザイリンクスは、7 シリーズ FPGA のトランシーバーを、高性能、低ジッターを実現できるように最適化しました。これらのトランシーバーは、低消費電力動作用の機能を備えています。設計者は、動作を柔軟にカスタマイズし、消費電力とパフォーマンスのバランスを調整できます。

7 シリーズ FPGA の共有 LC PLL の使用も、消費電力を大幅に低減します。ライン レートが同じ 4 レーンのデザイン (XAUI など) では、チャネル個別の PLL の代わりにクワッド PLL を使用すれば消費電力が抑えられます。同様に、PLL は範囲内の高いレートと低いレートの両方で動作可能なため、消費電力を節約するには低い方の動作範囲を選択することを推奨します。

TX/RXPOWERDOWN オプションを個別に有効にする方法もあります。PLL のパワーダウン機能は、消費電力が最小となるモード (PCIe® システムで使用されることが多いシステム D3 ステートなど) で有効に設定できます。

デザイン サイクルの各段階

コーディングを始める前に、消費電力を重視した設計手法を理解して実践することが、システム電力を低減する唯一、かつ最大のメカニズムであると言えます。ザイリンクスの各種ツールを適切な設計段階で活用することも、消費電力の仕様を満たす上で助けとなるだけでなく、電源の数、種類、大きさを選択するために必要な情報も提供されます。ザイリンクス 7 シリーズ FPGA は、プロセステクノロジとアーキテクチャ デザインを通じて、これまでにない高い電力効率を実現しました。

今回紹介した手法の多くは、FPGA の消費電力最適化に関するトレーニング コースで解説しています。ザイリンクスのトレーニング コースの詳細については、japan.xilinx.com/training をご覧ください。

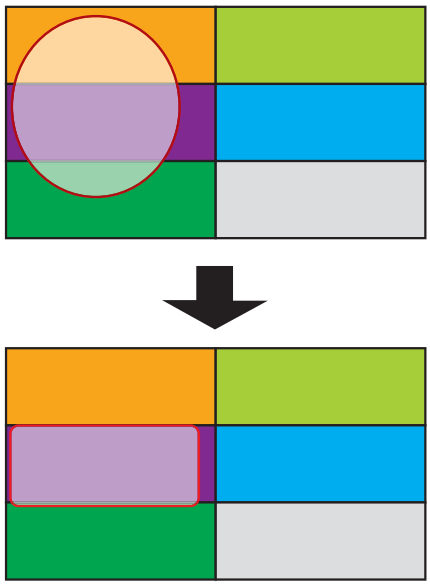


図 5 - 断続的に使用されるロジックは、可能な限り 1 つのクロック領域内にまとめて配置

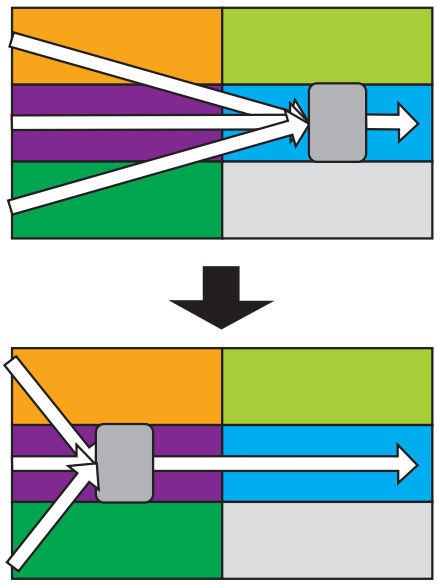
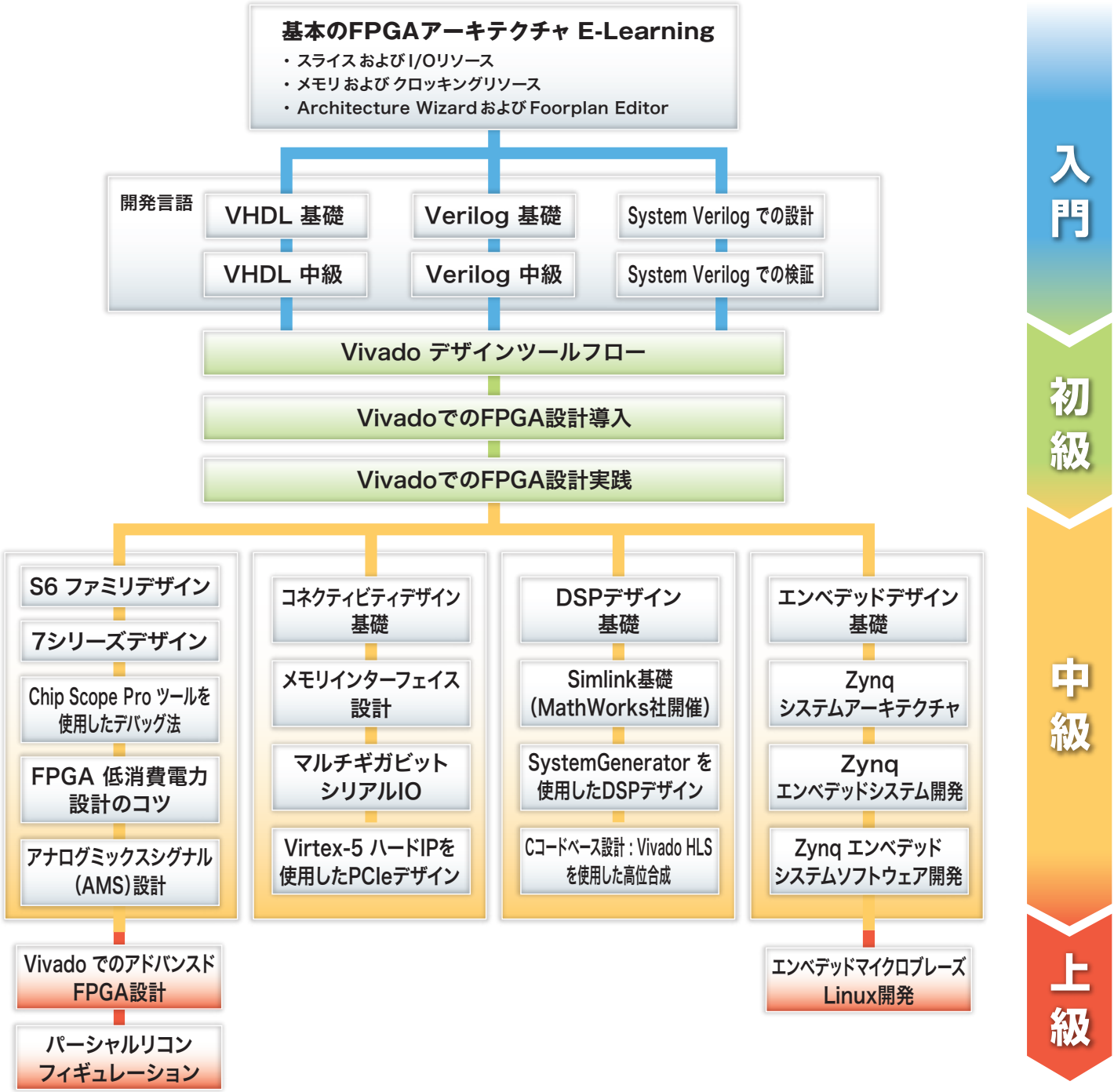


図 6 - データ移動の限定 - FPGA 周りではオペランドではなく結果のみを移動

Chandra Sekar Balakrishnan はザイリンクスのグローバル トレーニング ソリューション チームのソリューション開発エンジニアとして、ザイリンクスのトレーニング コースのコンテンツ制作を担当しています。専門は FPGA デザインとエンベデッドの分野です。2010 年 5 月にザイリンクスに入社する前は、Acme Technologies 社 (インド、チェンナイ) でプロジェクト リーダーを務めていました。同社では 8 年以上、日本の半導体組織向けに FPGA ベースのエンベデッド システムをインプリメントするプロジェクトに携わりました。
筆者への問い合わせは、cbalakr@xilinx.com まで。

ザイリンクス トレーニング カリキュラムパス

XILINX



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。

AVNET	アヴネット ジャパン	avnet.co.jp/design/training/
新光商事	新光商事	xilinx.shinko-sj.co.jp/training/index.html
東京エレクトロン デバイス	東京エレクトロンデバイス	ppg.teldevice.co.jp/
PALTEK	パルテック	www.paltek.co.jp/seminar/index.htm
FUJITSU	富士通エレクトロニクス	jp.fujitsu.com/fei/services/maker/xilinx/xtraining.html

詳細とご登録はこちらから

Japan.xilinx.com/training/

XILINX®

2倍のパフォーマンス、半分の消費電力

ザイリンクスの 7 シリーズ FPGA で妥協のない革新を。
新しい 7 シリーズ デバイスは統一されたアーキテクチャ上に構築しているため、
選択肢が広がりお客様のコンセプトをより現実的に設計できます。
また、高性能、低消費電力などさまざまなニーズを満たします。
次世代のISE® Design Suiteで開発時間をスピートアップし、
この高性能と柔軟性で世界を変える革新をおこしてください。

http://japan.xilinx.com/7

ARTIX 7™ 最も低いシステムコスト
KINTEX 7™ 最高の価格対性能
VIRTEX 7® 最高のシステム性能

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)443-4016 x2web@teldevice.co.jp ■アヴネット ジャパン(株) TEL(03)5792-8210 EVAL-KITS-JP@avnet.com ■(株)PALTEK TEL(045)477-2005 info_pal@paltek.co.jp
■富士通エレクトロニクス(株) TEL(045)415-5825 fei-xinq@cs.jp.fujitsu.com ■新光商事(株) TEL(03)6361-8086 X-Pro@shinko-sj.co.jp

©Copyright 2012 Xilinx, Inc. All rights reserved. ザイリンクスの名称およびロゴ、Artix、Kintex、Virtex、ISEは米国およびその他各国のザイリンクス社の登録商標および商標です。