

## FPGA デザインと DSP デザインの信頼性とメンテナンス性の比較

### はじめに

デジタル信号処理 (DSP) が、現代の無線通信、有線通信、医用診断装置、軍用システム、オーディオ / ビデオ装置、その他の多くの製品を支えており、消費者の生活に広く入り込むようになりました。半導体技術の進歩により、これまでになく複雑な DSP アルゴリズム、プロトコル、アプリケーションが実現可能になった反面、システムや製品は複雑になっています。システムが複雑になるほど、システムの信頼性は、平均故障間隔 (MTBF) といったハードウェア・プラットフォームの信頼性指標だけでは定義することができなくなりました。システムの信頼性は、ハードウェアとソフトウェアのアーキテクチャ、開発 / 検証プロセス、デザインのメンテナンス性によって決定されることがますます増えています。

アーキテクチャの基本的な問題の 1 つは、ハードウェア・プラットフォームのタイプです。DSP 機能は、一般に DSP プロセッサとフィールド・プログラマブル・ゲートアレイ (FPGA) の 2 つのタイプのプログラマブルなプラットフォーム上に実現されています。DSP プロセッサはマイクロプロセッサの特別な形式であり、これに対して FPGA は高度にコンフィギュラブルなハードウェア形式です。これまで、DSP プロセッサはコピキタスといえるほど広く採用されてきましたが、多くのアプリケーションのニーズが DSP プロセッサの処理能力 (百万単位で表した毎秒の命令実行回数: MIPS) を超えるようになると、FPGA の使用が急速に増えてきました。現在、DSP プロセッサより FPGA の方が選択される主な理由は、アプリケーションの MIPS 要求によります。このため、DSP プロセッサと FPGA との間の比較では MIPS の比較が中心になっています。これは確かに重要ですが、これだけが FPGA の利点ではありません。同様に重要ですが、しばしば見落とされるのは、製品の信頼性とメンテナンス性について FPGA が生来備えている利点です。この 2 つ目の利点をこのホワイトペーパーでは説明します。

### FPGA と DSP プロセッサの開発プロセス

ほぼすべての設計プロジェクト・マネージャは、次製品ソフトウェアの更新またはリリースの日付を回答できる準備をしています。大部分の技術会社には、ソフトウェア・バグまたは問題レポート、およびそれに対応するパッチまたは解決を含むソフトウェア・リリースの長い内部リストがあります。DSP コードなど、すべてのソフトウェアがあるレベルのバグを含むこと、それを小さく抑えることだけであることが知られるようになりました。一方、FPGA デザインの更新頻度は比較的小さくなる傾向があり、一般に、FPGA コンフィギュレーション・ファイルのフィールド・アップグレードを発行することはメーカにとってまれなことになっています。

この理由は、DSP プロセッサと FPGA との間の設計開発プロセスの違いにあります。どのタイプのプロセッサでも複雑なソフトウェアを開発しなければならないという基本的な課題があります。DSP プロセッサは本質的に、信号処理 / 制御またはプロトコル指向の強い、さまざまなタスクに対して常に再構成される特別な処理エンジンです。プロセッサ・コアのレジスタ、内部メモリ、外部メモリ、DMA エンジン、I/O ペリフェラルなどのリソースは、"スレッド" と呼ばれる多くのタスクによって共有されます。これらのリソース共有は、しばしば予期しない方法で、または不明確な方法で、タスク間の相互干渉を発生させます。さらに、大部分の DSP アルゴリズムは、"リアルタイム" で動作することが要求されます。そのために、予期できない遅延またはレイテンシによりシステム障害が発生することがあります。一般的な DSP ソフトウェア・バグの原因としては、次のようなものがあります。

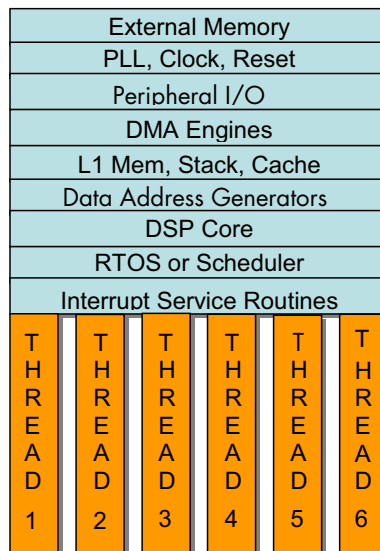
- 異なる機能を同時に開発 / インテグレーションする複数の技術者によってなされるプロセッサ・リソースについての不統一な仮定
- 完了時にプロセッサの状態を完全に復旧しない割り込み処理
- 重要な割り込み処理の実行が、別の割り込み処理もしくは割り込み不可プロセスにより妨害される場合
- ポインタの破壊が検出できないこと、またはポインタの初期化忘れ
- サーキュラ・バッファリング・アドレッシング・モード制御の失敗

- スレッドが終了した際にすべてのメモリを解放することに失敗したために、使用可能な揮発性メモリが徐々に消費されてしまうメモリ・リーク
- 特定の変数メモリ配置に対する DSP ルーチンの依存性
- メモリ・リンカーとコンパイラの最適化による予期しないメモリ再配置
- コア内での特別な DSP "コア・モード" 命令オプションの使用
- DMA、シリアル・ポート、L1、L2、外部 SDRAM メモリなどのペリフェラル・アクセスの競合や、大きなレイテンシ
- スタックまたはセマフォの破壊
- サブルーチン実行時間が、入力データやコンフィギュレーションに依存すること
- "C" または高級言語サブルーチンとアセンブリ言語サブルーチンの混在
- アセンブリ命令のパイプライン制約

### リソースの管理

マイクロプロセッサ、DSP プロセッサ、オペレーティング・システム (OS) の各ベンダは、タスク間またはスレッド間にさまざまなレベルの保護や独立性を設けることにより、これらの問題を解決しようとしてきました。許容実行時間、メモリ、または共通ペリフェラル・リソースなどのプロセッサ・リソースのアクセスを管理するために、オペレーティング・システムすなわちカーネルが使われますが、処理効率と OS が提供する保護レベルとは、もともと相反する性質を持っています。処理効率と確定的レイテンシが決定的に重要となる DSP プロセッサ (図 1) の場合には、結果的に OS によるタスク間の分離は最小限またはゼロになってしまいます。各タスクは効率良く動作するために、多くのプロセッサ・リソースを無制限にアクセスできることを要求します。

図 1. DSP プロセッサのブロック図



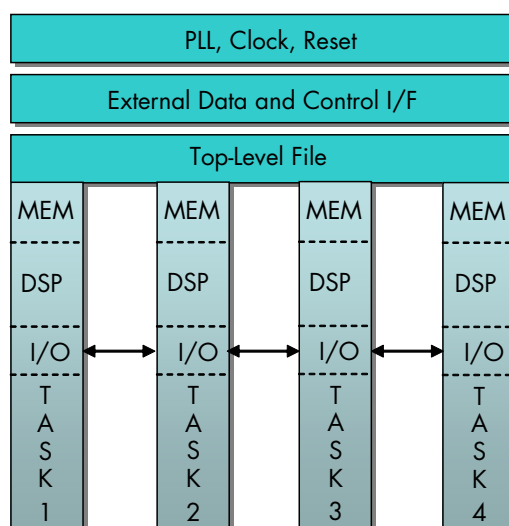
さらに、これらの開発課題がからみあうことで、初期開発段階や後続のコード・リリース時の回帰テストでの検証カバリッジが不完全になります。可能な組み合わせ ("コーナー・ケース" と呼ばれます) をすべてテストすること、さらにフィールドでの動作で発生する可能性のある、個々のタスクまたはスレッド間の相互干渉をすべてテストすることは、ほぼ不可能であるということが、ソフトウェア開発プロセスの恐らく最も困難な部分になっています。自動テスト・スクリプトを使ったとしても、可能なすべてのシナリオをテストすることは不可能です。既知のバグを修正するとき、または新しい機能を追加するときは、ソフトウェアの更新または修正を行う毎にこのプロセスを繰り返す必要があります。新しいソフトウェアをリリースする際にも、ときどき新しいバグが不注意のために導入されることがあり、新しいバグを修正するためにさらにもう 1 回リリースが必要になることがあります。製品が複雑になる

ほど、コードの行数が増えるために、プロセッサ・コア数も増え、開発労力の大部分がソフトウェアのテストに費やされています。

それでは、FPGA 開発プロセスはこのような課題を、どのように改善できるのでしょうか？ デザインに DSP プロセッサを使うか、または FPGA を使うかに関係なく、各タスクの複雑さは多かれ少なかれ同じです。両方法とも、一般的な信号処理アルゴリズム、インタフェース、プロトコルのサードパーティ・インプリメンテーションを使うというオプションを提供しています。また、いずれの方法でも、既存 IP を将来のデザインで再利用することが可能です。しかし、両者が同様なのはここまでです。

FPGA は、大部分の DSP アルゴリズムに対して、専用の処理回路を提供できます。各タスクには専用のリソースが割り当てられ、独立して実行されます。各ステップの専用リソースを使って組み立てラインのようなプロセスで連続的に流れる信号処理チェーンの各ステップを処理することが理にかなっていると、直感的に分かります。約 100 年前にヘンリー・フォードが発見したように、この方式は処理量を大幅に向上させます。

図 2. FPGA のブロック図



タスクの要求は論理的な分割に沿って分けることができますが、割り当てる FPGA リソース量を各タスクの要求に合わせて調整することができます。これにより、タスク間のインタフェースをうまく定めることができるようになるため、タスク間の不必要な干渉を大幅に削減することができます。データをバッファしてバッチで処理しなければならない DSP プロセッサ内部に比べると、各タスクの動作は連続しているため所要メモリははるかに少なく済みます。FPGA がデバイス内でメモリを配分するため、各タスクには必要とされる専用メモリが恒久的に割り当てられます。このためにタスク間の高度な分離が可能になり、1 つのタスクを修正しても、別のタスクで予期しない動作が発生するようなことはありません。このために、論理的かつ予測可能な方法でバグの分離と修正を容易に行うことができます。

## FPGA デザインの検証

基本レベルでは、FPGA のデザイン・ツールと検証ツールは ASIC 開発ツールに密接に関係しています。実際に、大部分の ASIC デザインの試作は FPGA を用いて行われています。ASIC 内ではバグは許されないため、これは重要なことです。ASIC 内のデザイン・バグを修正するフィールド・アップグレードはほぼ不可能に近く、可能であったとしても時間と開発コストが非常にかかるため、ASIC 開発者はデザイン検証を非常に厳密に行う必要に迫られています。このために考え得るすべての動作条件で、各ゲートのほぼ完全なカバリッジを提供するテスト方法を採用することになります。

FPGA の検証ツールは ASIC 検証ツールと密接な関係があり、ASIC 検証プロセスでの長年の成果から多くの恩恵を受けています。FPGA 分割機能、テスト・ベンチ、シミュレーション・モデルを使用することにより、問題の分離、開発プロセスの迅速化、製品メンテナンスの簡素化、機能追加の実現と回帰テストが非常に効果的になりました。DSP プロセッサ開発プロセスと比較した場合の、これらの FPGA の決定的な利点は、デザインの複雑さが増し、かつ開発チームが大きくなるほど、ますます重要になります。

例えば、アルテラの FPGA はインハウス・ツールとサードパーティ・ツールによって包括的にサポートされており、アーキテクチャ・パーティション機能、フロア・プラン機能、デザイン目標の実現、シミュレーション、タイミング・クロージャ、最適化、メンテナンス性に対する統一ツール・フローを提供しています。特に、パーティション機能は、デザイン入力プロセスに不可欠です。タイミング・クロージャ時と開発メンテナンス段階で、このパーティション（通常、パーティション内で必要とされるチップ・リソースをすべて含む）を結合することで、高度な分離を保証します。各論理パーティション、およびデザイン全体は、独立なテスト・ベンチとシミュレーション・モデルを持つことができます。

### まとめ

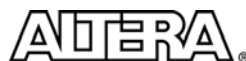
大規模なエレクトロニック・デザイン・オートメーション（EDA）業界は、FPGA と ASIC のテストおよび検証ツールの開発を継続的に促進しています。ソフトウェア開発業界には、これに対応する部分がありません。ソフトウェア検証における膨大なコストと問題に対する理解が進むと、この状況は変わると思われますが、現在は、最新パッチのダウンロードを続けることが実用的なソフトウェア・ソリューションになっています。

多くの設計マネージャは、バグを修正するためのソフトウェア更新頻度は、ほぼすべてのケースで同等な FPGA の更新頻度をはるかに超えていることを理解しています。組み込みソフトウェアのバグ修正が定期的に提供されることは、当然のことと考えられるようになってしまいました。それに対し、FPGA による開発は、低コストかつハイエンドの信号処理システム向けに最適化されたデバイスが提供され、広範囲な IP コアやハイレベルなデザイン入力方法があり、さらにはデザインおよび検証プロセスの固有な堅牢さにより、信号処理システムの開発において、より優れた選択肢としてますます注目されるものとなっています。

### 謝辞

- Michael Parker, Sr. Technical Marketing Manager, IP and Technology Product Marketing, Altera Corporation

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.