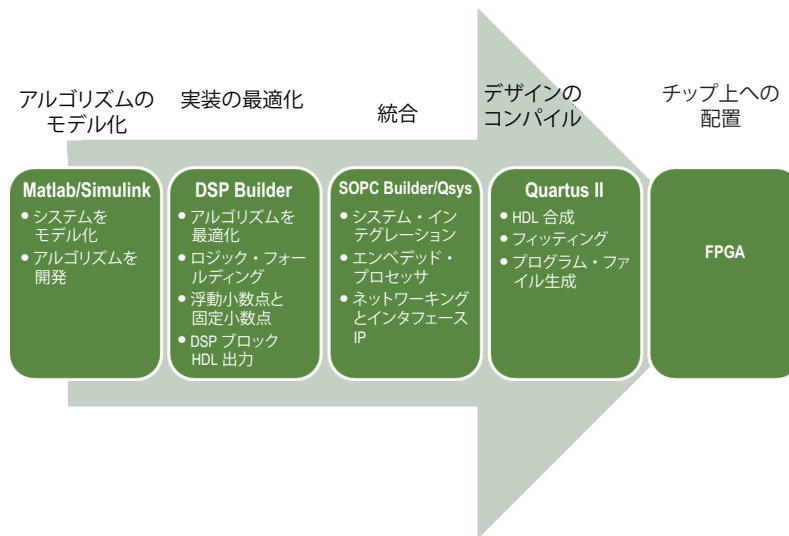


このホワイト・ペーパーでは、アルテラの FPGA が持つ適応性、可変精度 DSP、およびモーター制御デザインに対して統合されたシステム・レベルのデザイン・ツールを活用したデザイン・フローについて解説します。モーターで駆動する工業用機器の設計者は、このデザイン・フローの性能、集積度、および効率という利点を活用できます。

## はじめに

モーター駆動工業用機器は、工業用電力の 2/3 を超える電力を消費します。したがって、これらの電気機器の運用効率化は、工場の経費にとって重要な要素となります。モーター駆動システムの従来のドライブを可変速ドライブ (VSD) に置き換えると、最大 40% の電力削減に相当する大きな効率化になる可能性があります。アルテラの FPGA アーキテクチャは、図 1 に示す柔軟性、性能、統合、およびデザイン・フローの利点により、非常に効率的なプラットフォームを VSD システムに提供します。

図 1. 最適化されたモーター制御 FPGA デザイン・フロー



- 性能の拡大 — 並列処理と機能のスケーラビリティによって、さまざまなタイプのモーターでより高い性能と効率が得られます。
- デザイン統合 — エンベデッド・プロセッサ、エンコーダ・インタフェース、DSP モーション制御アルゴリズム、および産業用ネットワークを 1 つのデバイスに統合しています。
- デザインの柔軟性 — IP を再利用し、可変精度 DSP ブロックを活用します。制御パスのあらゆる部分に対して、固定小数点精度または浮動小数点精度を使用できます。

- 確定的レイテンシー モーターのアルゴリズムを確定的動作のハードウェアで実装できます。
- 簡素化された強力なツール — Simulink などのモデリング・ツールを、アルテラの **DSP Builder**、**Qsys** や **SOPC Builder** 内の汎用的な統合ツールと組み合わせて使用し、低コストの FPGA で完全なモーター・システムを最適化します。負荷をモニタし、位置、速度、その他のドライブ項目を調整する処理ループや制御ループを実装するには、市販のマイクロコントローラ (MCU) やデジタル信号プロセッサ (DSP) を使用するのが一般的です。しかし、マイクロコントローラには、拡張性と性能の不足による制約があります。これらの欠点は、高 MIPS 処理が要求され、ますます複雑化するアルゴリズムのシステムに最も顕著に現れます。さらに、ソフトウェアでアルゴリズムを記述すると、ハードウェアで最適化したシステムの要件に適合させることが困難です。

同様に、一般にハイエンド DSP はモーター制御のための計算能力を備えています。メモリ・インターフェース、信号インターフェース、信号フィルタリング、産業用イーサネット・プロトコル規格のサポートなど、時間的精度が要求される演算とタスク指向演算を同時に組み合わせたシステムには理想的とは言えません。

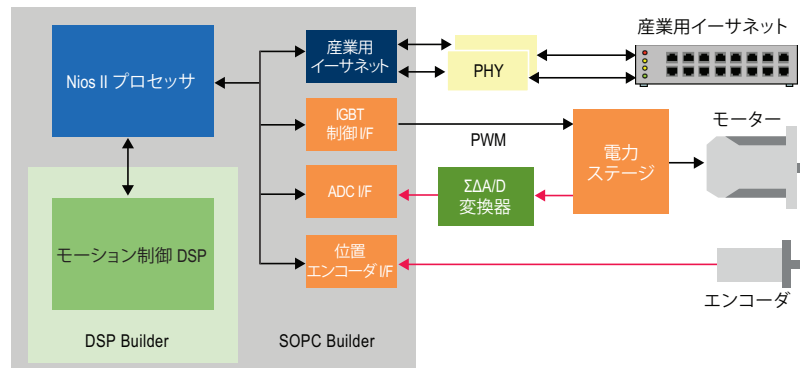
## 性能のスケールアップと統合化の利点

多くの市販 MCU や DSP は、汎用ドライブを操作するための基本要素を備えています。ところが、これらのデバイスが備えているのは、固定メモリ、狭いアナログ範囲、固定チャネルの PWM、および多軸システムへの限定されたサポートです。より高い性能とモーター効率の改善が要求される次世代ドライブは、処理と DSP 要件に対応する性能スケールアップ機能を提供し、システムの統合と最適化への柔軟性も併せて備えたプラットフォームを必要とします。

FPGA は、アプリケーション要件に基づいて性能を容易に拡張できます。設計者は、FPGA に複数のプロセッサを組み込んだり、FPGA の柔軟な DSP 機能を使用したりできます。その後、追加ロジック、カスタム命令、またはサポートされている多くの産業用ネットワーク・プロトコルの 1 つを利用します。アルテラの FPGA により、各サブシステムを独立して制御するいくつかのエンベデッド・プロセッサを実装できるようになります。アルテラ FPGA の並列機能は、ほとんどのモーター制御システムのビルディング・ブロックの統合をサポートします。例えば、アルテラの **Nios II** エンベデッド・プロセッサ (32 ビット RISC ソフト・プロセッサ) は、さまざまなインターフェース、センサー、エンコーダをすべて制御できます。設計者は、次に可変精度の浮動小数点 DSP ブロックを使用して、フィールド指向制御 (FOC) や演算量の多いその他のアルゴリズムを実行できます。

FPGA に統合して 1 つの「1 チップ・ドライブ」システムを生成できる、さまざまな要素を図 2 に示します。統合された IP 機能は並列動作が可能であり、シーケンシャル動作と遅延動作のいずれにもボトルネックがないことを保証します。

図 2. 複雑な演算アルゴリズムを含む FOC モデル



このデザイン・フローは、下記をはじめとする有用な IP の統合をサポートします。

- 位置フィードバック — EnDAT、Hiperface、BiSSなどの高精度位置フィードバック機能を備えたエンコーダで、10 倍の速度で位置データを得ることができます。
- IGBT 制御 — 絶縁ゲート・バイポーラ・トランジスタ (IGBT) を使用して、AC モーター駆動に必要な高電圧をスイッチングします。FPGA の空間ベクトル変調 (SVM) 手法を使用して IGBT のゲート入力をパルス幅変調 (PWM) し、モーター駆動に必要な正弦波電圧を生成します。IGBT は、2 レベル回路または 3 レベル回路が可能です。
- ADC インタフェース — モーターからのフィードバック電流を測定するための、外部アナログ - デジタル・コンバータ (ADC) へのインタフェース。シグマ - デルタ ADC は低ノイズで高駆動電圧を容易に光アイソレーションし、その出力をFPGAがサンプリングして高速かつ正確に読み取ることができるようにします。
- ネットワーキング・インタフェース — FPGA にリアルタイム・プロトコルを実装し、Ethernet/IP、PROFINET IO/IRT、EtherCAT などのアプリケーションに必要な産業用イーサネット・プロトコル規格に対応します。産業用イーサネットは、産業用ドライブの共通機能になりつつあります。

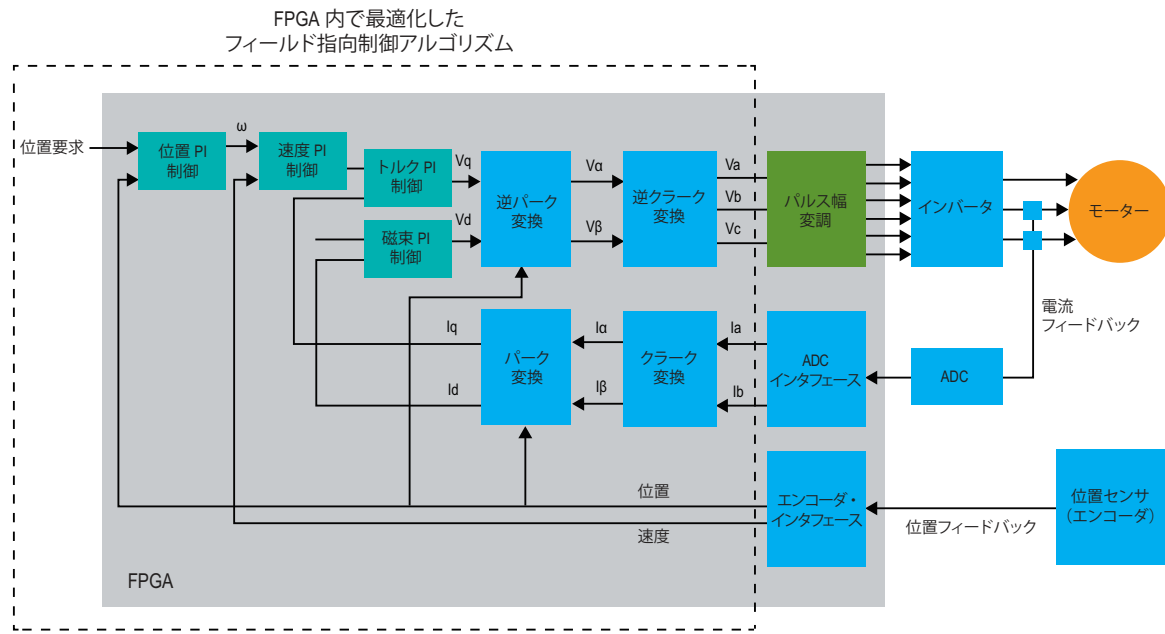
DSP をベースとするこれらのモーター制御機能、通信、およびインタフェース規格の普及により、FPGA が工業用モーター・ドライブの理想的なプラットフォームになります。

## 複雑な演算アルゴリズムの処理

ドライブ技術は、モーターの種類とアプリケーションによって大幅に変わります。例えば、ポンプやファンを制御するドライブは、CNC マシンや包装機械を制御するドライブとは異なる要件とフィードバック・メカニズムを備えています。これらのエンコーダやセンサーから収集したデータは、制御システムにフィードバックされ、ターゲット・システムの負荷要件とトルク要件に適した電圧レベルを決定する演算アルゴリズムに使用されます。

例えば、一般に使用される永久磁石同期モーター（PMSM）は、ベクトル制御とも呼ばれる演算量の多いフィールド指向制御（FOC）を制御ループのアルゴリズムの一部に使用します。FOC は、正確なトルク制御を必要とする工業用サーボ・モーターにも有用です。FOC 手法は、モーターのサイズ、コスト、および消費電力の削減に役立ちます。FOC は、電圧レベルおよび対応するモーター速度を正確に測定して速度およびトルク制御を向上させ、負荷が変動してもトルクを一定に保ちます。FOC には、トルク・リップルや電磁妨害を軽減する効果もあります。ただし、図 3 に示すようにこの計算モデルはかなり複雑なため、このアルゴリズムを超高速で実行するには大きなコンピューティング能力が必要です。

図 3. フィールド指向制御（FOC）モデル



FOC は、モーターの正弦波 3 相電流をリアルタイムで制御して、回転数が正弦波の周波数に対応する円滑な回転磁束パターンを生成します。この手法では、電流ベクトルの振幅を制御して、その位置を回転子の磁束軸に対して 90 度（「直交」電流）に保ちます。そのため、設計者は「直交」成分（0 度）をゼロに保ちながらトルクを制御できるようになります。このアルゴリズムは次のステップで構成されます。

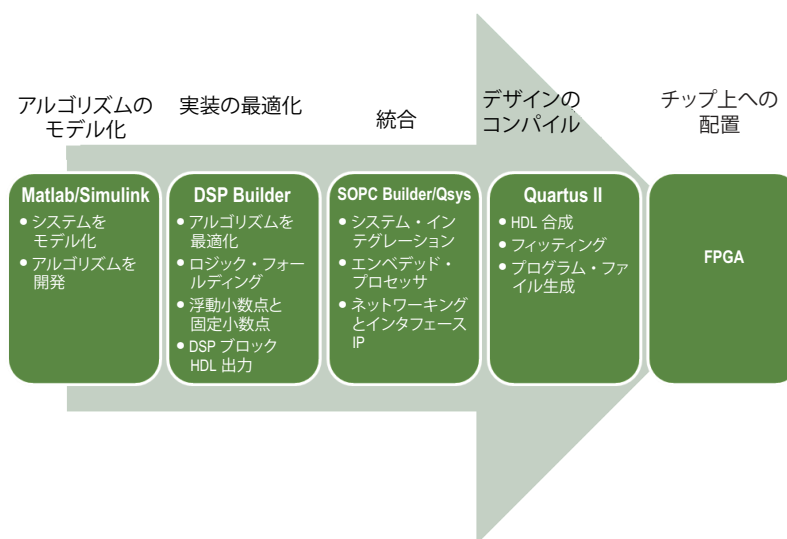
1. 3 相フィードバック電流入力とエンコーダからの回転子位置を、クラークおよびパーク変換によって「直交」成分と「直交」成分に変換します。
2. これらの電流成分を、並列動作する 2 つの比例および積分（PI）コントローラへの入力として使用して、「直交」電流をゼロに、「直交」電流を望ましいトルクにそれぞれ制限します。
3. PI コントローラからの「直交」および「直交」電流出力を、クラークおよびパーク逆変換によって 3 相電流に戻す変換を行います。

業界初の可変精度 DSP ブロックを備えたアルテラの FPGA は、要件に正確に適合する精度レベルを選択する柔軟性を提供するとともに、単精度または倍精度の浮動小数点タイプもサポートします。これらの要素によって、この DSP ブロックは、FOC 制御ループやその他の複雑な演算アルゴリズムを実装するのに理想的な選択肢になります。多くのアルテラ 28nm FPGA アーキテクチャが備える統合 DSP ブロックは、18 ビット・モードまたは高精度モードのいずれでもコンパイル時にコンフィギュレーション可能です。

## 強力な開発ツールの活用

モーター制御デザインを最適化するには、システムのモデル化とシミュレーションを支援し、複雑なアルゴリズムを低レイテンシで実装し、システムを統合して、モーター・ドライブのニーズに正確に合わせて性能を微調整する機能を持つ汎用ツール（および実用的なツール・フロー）が必要です。図 4 に示す統合ツール・フローを使用すると、設計者は開発時間を削減し、ドライブ・システムのさまざまなタイプに応じてスケーラブルな、柔軟かつ強力なモデルを提供する機能を活用できるようになります。

図 4. 最適化されたモーター制御 FPGA デザイン・フロー



アルテラは、エンベデッド業界の設計者に、Quartus® II 開発ソフトウェア、MegaCore® IP ライブラリなどの強力を使いやすい開発ツールを提供しています。アルテラは、図 4 に示すタスク指向操作の Qsys、SOPC Builder や、DSP 最適化用の DSP Builder などのシステム統合ツールも提供しています。さらに、FPGA ハードウェアを補完してデザイン・フローを簡素化する、Eclipse ベースの Nios® II エンベデッド・デザイン・スイート (EDS) も提供しています。

## Nios II エンベデッド・デザイン・スイート

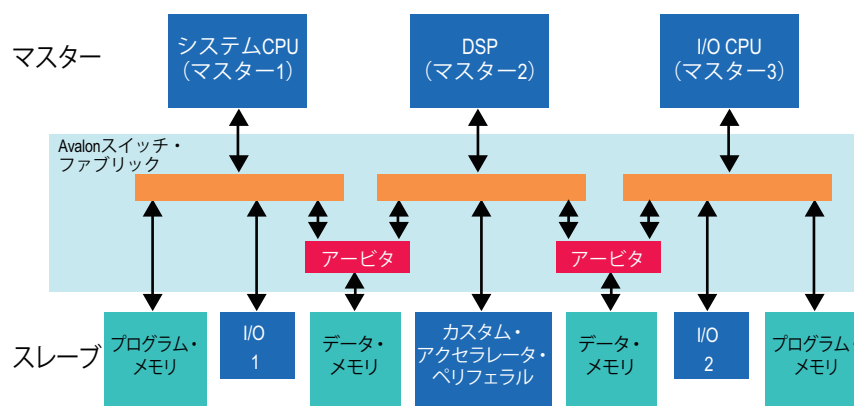
アルテラは、Eclipse ベースの Nios II EDS、Nios II エンベデッド・プロセッサなどの強力かつ使いやすいエンベデッド開発ツールのほか、よく知られた多くのベンダが提供する標準オペレーティング・システム (OS) やリアルタイム・オペレーティング・システム (RTOS) のサポートも提供しています。Nios II EDS は、複数の汎用 32 ビット RISC ソフト・プロセッサの実装をサポートします。これらのプロセッサは、最大 340 MIPS (Dhrystones 2.1) の性能を備え、独自のインストラクション・セット、データ・パス、およびアドレス空間によって独立して実行できます。

## Qsys および SOPC Builder システム統合ツール

アルテラの Quartus II 開発ソフトウェアは、最新の Qsys (およびレガシー SOPC Builder) システム統合ツールを含んでいます。これらのツールは、図 7 に示すハードウェア・コンポーネントの統合作業を自動化することによって、設計者が完全なシステム・オン・チップ (SoC) を定義して生成するのに役立ちます。Qsys や SOPC Builder は、HDL モジュールの定義と接続を手作業で行う従来の設計方法の代わりに、GUI でシステム・コンポーネントを定義するのに役立ち、インタコネクト・ロジックを自動的に生成します。これらのツールは、システムのすべてのコンポーネントを定義する HDL ファイルと、すべてのコンポーネントを接続するトップ・レベルの HDL ファイルを生成します。これらのツールは、Verilog HDL か VHDL のいずれかを生成します。

これらのシステム統合ツールは、Avalon インタフェースを使用して論理デバイス (オンチップまたはオフチップ) を接続します。モーター・システムでの Avalon インタフェースは、ソフト・プロセッサをドライブ・システムの他の要素と接続します。これは、システム統合の複雑さを軽減し、最適化のためのよりまとまった直感的なシステム提供します。インタコネクト・ファブリックは、スレーブ側を調停して同時マルチマスタリングを許容することにより、これらの接続を管理します。これらのツールは、各スレーブ・ポートの前に調停モジュールを挿入します。このモジュールは、さまざまなマスターからの要求を管理し、システムのインタコネクトの詳細をマスターおよびスレーブ・ポートの両方から抽象化します。システム内の複数のスレーブとマスターを接続するインタコネクト・ファブリックを図 5 に示します。

図 5. Qsys または SOPC Builder によるシステム・インタコネクトの管理



アルテラのシステム統合およびエンベデッド開発ツールは、DSP Builder で設計したハードウェア加速によるモーター制御アルゴリズムをプロセッサと接続するインタフェースを、設計者が迅速に構築するのに役立ちます。

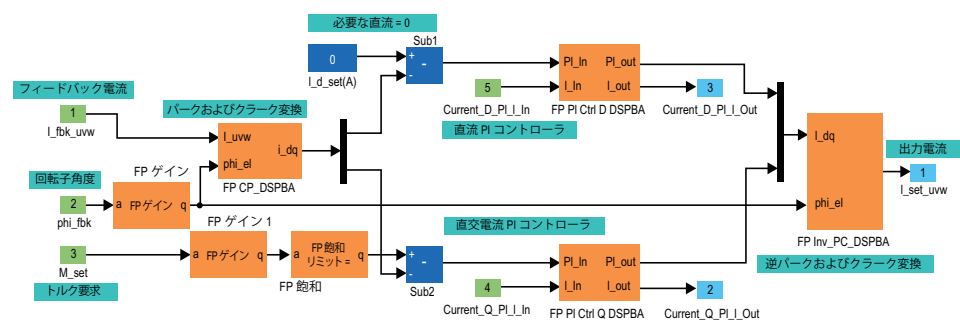
## DSP Builder および Matlab/Simulink

モーター制御システムの設計者は、演算量の多い高速モーター制御アルゴリズムに FPGA の DSP 機能を活用できます。アルテラは、DSP Builder を提供することによって、設計者がアルゴリズムに適した開発環境で DSP デザインのハードウェア記述を作成するのを支援し、DSP のデザイン・サイクルを短縮します。DSP Builder は、The Mathworks 社の MATLAB<sup>®</sup> および Simulink<sup>®</sup> システム・レベル・デザイン・ツールのアルゴリズム開発、シミュレーション、および検証機能と、アルテラ Quartus II ソフトウェアおよびサードパーティの合成およびシミュレーション・ツールを統合します。Simulink ブロックを DSP Builder ブロック、IP ブロックと組み合わせてシステム・レベルの仕様を検証し、シミュレーションを実行できます。

### FPGA 内の FOC のモデル化

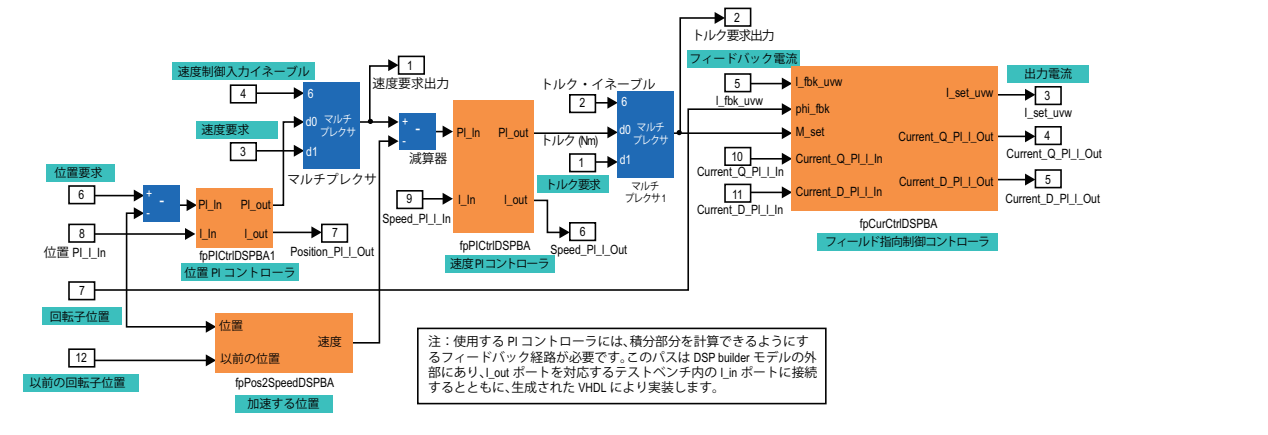
アルテラの統合 DSP Builder ツールにより、設計者は、図 6 に示すようなシステムを表現するプリミティブ・ブロックを接続するブロック図を作成して、FOC アルゴリズムを Matlab/Simulink 環境で直接モデル化できます。この例で使用するプリミティブ・ブロックは、ADD、SUB、MULTIPLY、CONSTANT、COSINE、および SINE です。Simulink では、モデルやシステムに対するアルゴリズムの振る舞いをビット精度で演算シミュレーションすることができます。アルゴリズムの開発が終了したら、DSP builder は、選択されたアルテラ FPGA を対象として最適化されたパイプライン化 RTL を自動的に生成します。

図 6. 永久磁石同期マシン用 FOC の例



このデザイン例には、図 7 に示す回転子の速度と角度の制御を可能にする、位置および速度制御ループも含まれています。代表的なモーター制御 IP システムは、空間ベクトル PWM、電流およびトルク制御ループ、速度および位置制御ループを備えています。設計者は、FPGA と CPU のリソース使用率に基づいて、これらの要素をハードウェア実装とソフトウェア実装に分配できます。

図 7. 永久磁石同期マシンの位置、速度、FOC コントローラ

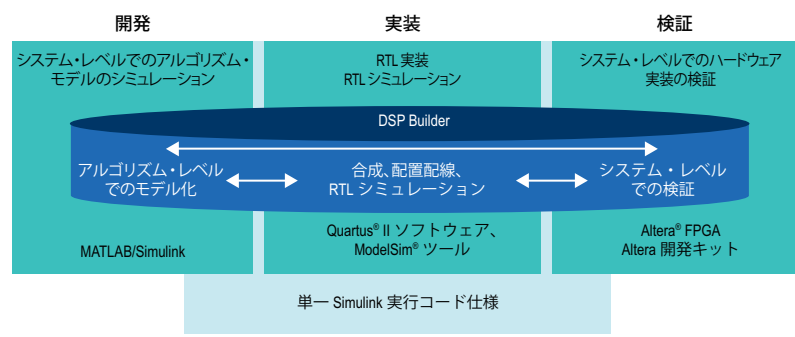


代表的なレガシー・デザイン・フローでは、電気およびモーター・システムのモデルを作成し、シミュレーションによってアルゴリズムを開発してから、DSP で実行するアルゴリズムを実装する「C」コードを記述していました。このデザイン・フローには、次のような重大な欠点があります。

- 多くの場合、アルゴリズムのモデル化を浮動小数点で行った後、DSP に実装するために固定小数点に変換します。従来、浮動小数点から固定小数点への変換は手作業で行われましたが、スケールとオーバーフローからの保護が煩雑でした。
- 「C」コードの実装を、モデルに対して再検証しなければなりません。
- アルゴリズムの実行時性能を向上させるには、次の追加ステップが必要です。
  - a. 「C」コードを手作業で最適化することによる性能向上
  - b. より高価な高速 DSP プロセッサへのアップグレード
  - c. 複数の DSP デバイスでのアルゴリズムの並列実行（可能な場合）


設計者は、DSP Builder によって FPGA ベースの DSP システム・デザインを最適化できます。DSP Builder は、パイプライン化やリソース共有などの最適化を実行して効率的な RTL 表現を生成します。図 8 に示すように、既存の MATLAB 機能と Simulink ブロックをアルテラの DSP Builder ブロックやその他の IP コアと組み合わせ、システム・レベルのデザインと実装を DSP アルゴリズム開発とリンクさせることができます。DSP Builder は、システム、アルゴリズム、およびハードウェアの各設計者が共通の開発プラットフォームを共有できるようにします。

図 8. DSP Builder のシステム・レベル・デザイン



DSP Builder のアドバンスド・ブロックセットは、固定小数点形式や単精度／倍精度浮動小数点形式でのアルゴリズムのモデル化をネイティブでサポートしています。設計者はまず、必要とするより高い精度を使用して Simulink でアルゴリズムをモデル化した後、精度を最終実装用ツールの中でスケールリングします。DSP Builder は、以下の具体的な利点を提供します。

- アドバンスド・ブロックセットによる、FPGA へのプッシュボタン式アルゴリズム実装。手作業の変換ステップは不要。
- ハードウェアで実行する前に、Simulink で実行時レイテンシ、データ・スループット、およびアルゴリズム使用率の結果を直接観察。Simulink でデザイン・スペースの探索を実行して最適な実装を選択。
- ソフトウェアで実装すると、一般に低速で実行時間が変動する関数 (SQRT や三角関数など) を含むプリミティブ演算子を、生成時に最適化して固定。この手法はアルゴリズム実行時間を予測可能にし、一部の演算子を大幅に加速。

 DSP Builder スタンダードおよびアドバンスドブロックセットについて詳しくは、[DSP ハンドブック](#)のサイトをご覧ください。

## フォールディング理論による性能向上

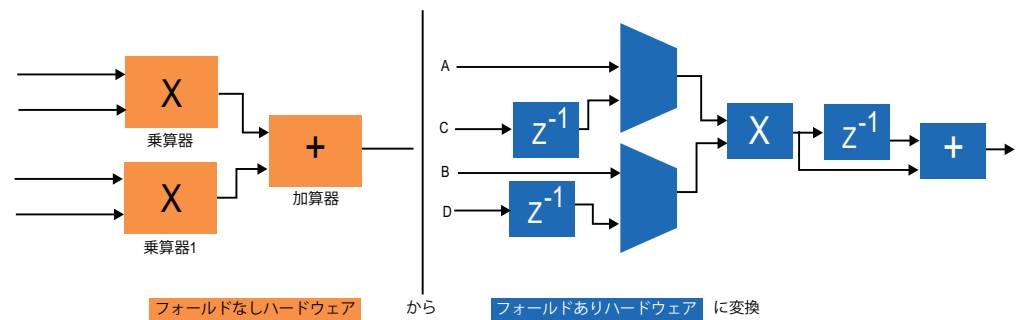
時分割多重化 (TDM) と密接に関連するフォールディング理論には、DSP モーター制御デザインを大幅に改善する可能性があります。システムのクロック・レートがデータ (サンプル) ・レートより速い場合、単一のハードウェア・コンポーネント (乗算器など) で複数のデータ・ポイントを処理できる可能性があります。フォールディング理論は、TDM 係数にアクセスするのと同様の方法で、複数のチャンネルが乗算器や加算器などのシステム・リソースにアクセスできるようにするため、リソースの節約につながります。

さまざまなデータ・ポイントが、TDM 係数を使用して共有ハードウェア・リソースにアクセスします。同様に、複数のパラレル・データ・ソースやデータ・チャンネルを備えるシステムでは、各チャンネルやデータ・ソースにハードウェアを重複して用意する必要はなく、1つのデータ・パスで複数のデータ・チャンネルを処理できます。

デフォルトでは、DSP Builder がプリミティブ・サブシステム用に生成するハードウェアは、クロック・サイクルごとに新しいデータを受け取って処理できます。一方、クロック・サイクルごとに計算する必要がないデザインもあります。サンプル・レートがクロック・レートより低いデザインでは、DSP Builder のアドバンスド・ブロックセットのフォールディング機能が、2つのレートの差を利用して、生成されたハード

ウェアの使用率を最適化できます。設計者は、フォールディングや TDM 係数がないものとして、最も直感的な方法でコア・アルゴリズムを実装できます。フォールディング理論によって、通常は手作業のフォールディング・デザインに必要な信号多重化およびデータ・バッファリング・スキームを明示的に実装する必要がなくなります。図 9 に示すように、全サイクルでは使用しないブロックをフォールディングによって組み合わせ、ハードウェアを削減できる可能性があります。

図 9. フォールディングなし、およびありのハードウェア例



DSP builder とプロセッサの性能を比較して正確なレイテンシとスループットの測定を保証するには、次の用語ガイドラインを使用します。

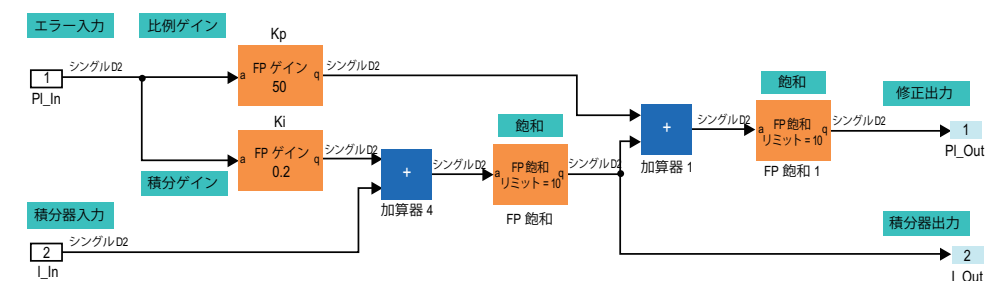
- 単一コア・プロセッサは、1つの演算の処理に  $x$  クロック・サイクルのレイテンシを必要とし、最初の演算が終了するまで新しい演算を開始できません。したがって、スループットは  $x$  クロック・サイクルあたり 1 演算となります。
- DSP builder システムは、1つの演算の処理に  $y$  クロック・サイクルのレイテンシを必要としますが、「フォールディング係数」クロック・サイクルごとに新しい演算を開始できます。したがって、スループットは「フォールディング係数」クロック・サイクルあたり 1 演算となります。

フォールディング係数をチューニングすることにより、再デザインせずに生成できるロジックのスループット、リソース使用率、およびレイテンシを相殺できます。次のセクションでは、フォールディング係数の影響を示すシステムとテストベンチの例を解説します。ここでの「フォールディング係数」クロック・サイクルは、 $x$  クロック・サイクルより小さくなります。

## FOC アルゴリズムのベンチマーク

アルテラは、既に解説したさまざまな特徴と機能をハイライトするために、FOC アルゴリズムをシミュレートして標準の「フォールディングなし」と「フォールディングあり」の実装結果を比較するベンチマークを開発しました。図 10 と図 11 に示すように、FOC アルゴリズムは、パーク／クラーク変換、および加減算、乗算、Sin、Cos 演算子を必要とする PI 制御ブロック、実装のための定数値と飽和ロジックで構成されます。代表的な FOC コントローラは、100 MHz の FPGA クロック・レートで容易に処理できる 10 ~ 100 ksps で入力をサンプリングします。100 ksps では、新しいサンプルを 10 $\mu$ s ごとに処理しなければなりません。この処理レイテンシを一定かつ最小に保つことが、制御アルゴリズムの性能に役立ちます。

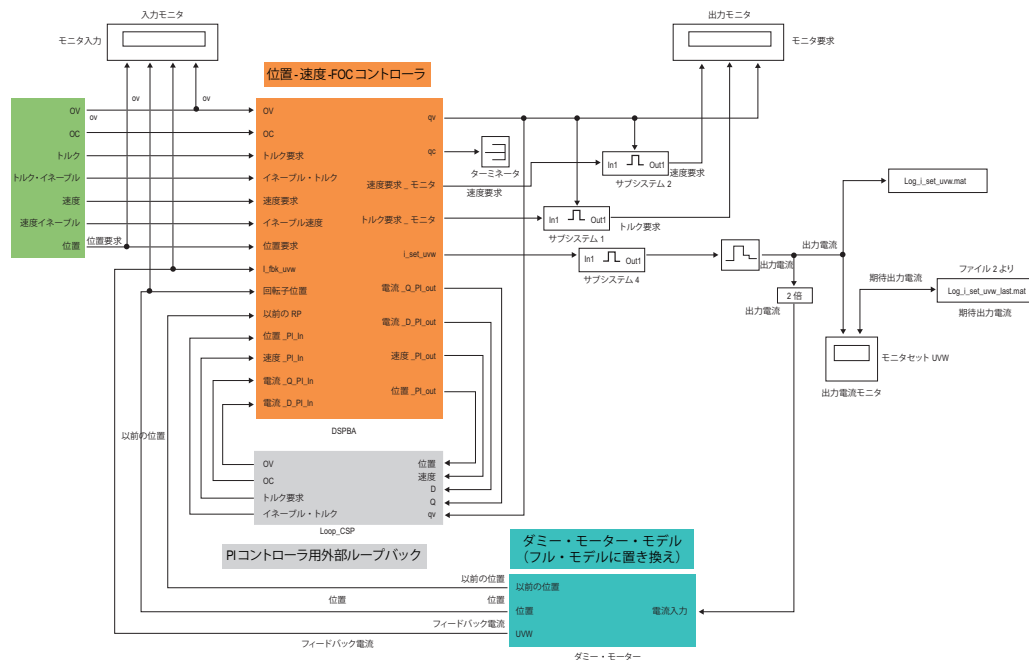
図 10. シンプルな比例および積分 (PI) コントローラ



テストベンチには、図 11 に示す以下の要素が含まれます。

- 入力スティミュラス (緑色) – 制御入力 (位置要求) を与えます。
- モーター・モデル (青緑色) – PMSM モーターをモデル化します。
- DSP Builder位置-速度-FOCコントローラ (黄色) – 制御アルゴリズムをモデル化します。
- 外部ループバック・モデル (灰色) – PI コントローラからの積分器フィードバック出力を入力にループさせます。

図 11. 永久磁石同期マシンの位置 - 速度 - フィールド指向制御コントローラ



## DSP Builder によるデザイン・チューニング

設計者は、Matlab ワークスペース変数から重要なシステム・パラメータを制御し、以下の方法でデザインをチューニングできます。

- フォールディング係数 — レイテンシ、スループット、およびリソース使用率のトレードオフをスイープできるようにして実装のスイートスポットを求めます。
- 固定小数点演算精度 — アルゴリズムの性能とリソース使用率に対する精度チューニングの効果を、アルゴリズム内のさまざまな段階で観察します。
- アルゴリズムのチューニング — プラント（モーター）の物理モデルに照らして実際のアルゴリズムをシミュレートし、モデル化段階の PID コントローラ、フィルタ、およびオブザーバのパラメータを調整します。

## ベンチマーク結果

以下のセクションでは、Cyclone IV デバイスに実装した単精度浮動小数点形式と固定小数点形式を使用する、Simulink でのモデリングから得られたアルゴリズムのベンチマーク結果の詳細を解説します。結果は、このデザイン例が必要とする 100 MHz のクロック・レート、リソース使用率、およびアルゴリズム・レイテンシの各要件に適合することを表わしています。

- 👉 Quartus II ソフトウェアでコンパイルに成功したら、設計者は Simulink 図の Quartus ブロック・リンクをクリックして正確なリソース情報を得ることができます。
- 👉 一般に、浮動小数点を提供する広いダイナミック・レンジを必要としないデザインは、固定小数点で実装されます。一方、浮動小数点はアルゴリズム開発やチューニング中の演算オーバーフローを防ぎます。

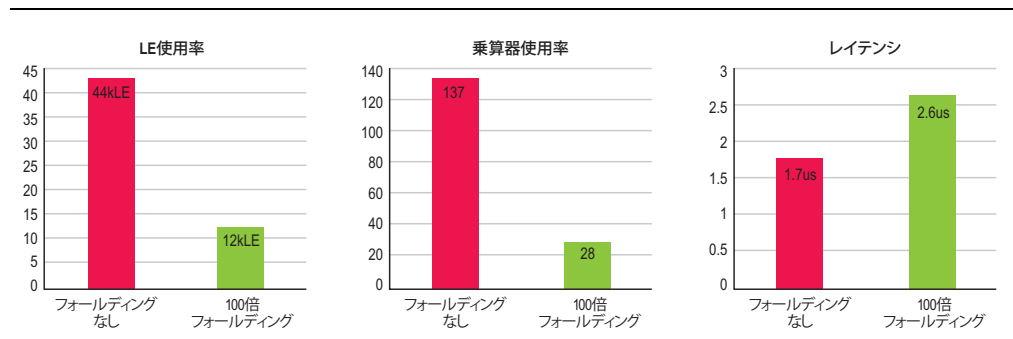
DSP builder は、クロック・サイクルごとに新しい入力値を受け入れ可能な、完全にパイプライン化された VHDL 表現をデフォルトで生成します。その後、この「フォールディングなし」構成で得られる結果を「完全フォールディング」構成と比較しました。

表 1 の結果は、フォールディング係数の結果として演算子が大幅に減ったことを示しています。そのため、集積度の低い Cyclone® IV デバイスを使用できるようになります。その上、レイテンシが増加してもアルゴリズムの許容範囲内に保たれています。制御ループの速度は、アルゴリズムのレイテンシと設定時間を加えたものです。5  $\mu$ s での結果は毎秒 20 万ループ (pwm 出力) であり、これは要求仕様を十分満たします。

表 1. フォールディング係数の利点

項目	フォールディングなし	フォールディング係数 100 倍
加減算ブロック	22	1
乗算器ブロック	22	1
Sin ブロック	4	1
最大スループット	100 Msps	1 Msps

図 12. システム・リソースとレイテンシ



DSP Builder は、固定小数点と浮動小数点のいずれも実装できるようにします。表 2 は、「完全にフォールドされた」固定小数点と浮動小数点の実装に必要なリソースの比較です。固定小数点の精度は、設計者がデザインの簡単な仮定実験を行えるようにする Matlab ワークスペース変数を使用して制御します。

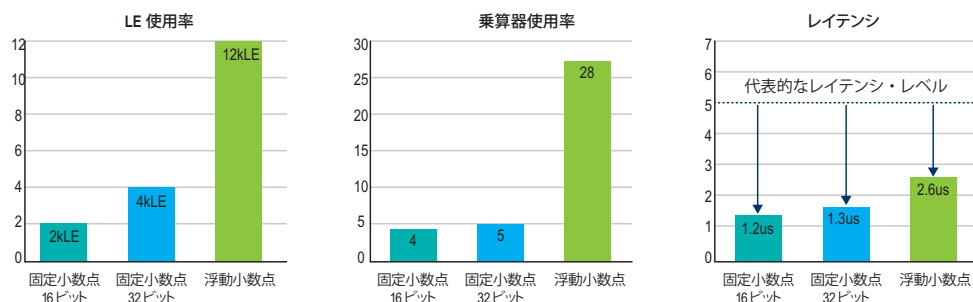
表 2. 浮動小数点と固定小数点の比較

項目	固定小数点 16 ビット	固定小数点 32 ビット	浮動小数点
ロジック・エレメント (LE) 数	2K	4K	12K
18 ビット乗算器	4	5	28
レイテンシ	1.21 $\mu$ s	1.36 $\mu$ s	2.65 $\mu$ s

**表 2 の注:**

- (1) 浮動小数点の結果は、浮動小数点 sine 関数実装を使用しています。代替の固定小数点実装を使用すると、LE の使用が 4,000、乗算器の使用が 16 個それぞれ削減されます。

図 13. システム・リソースとレイテンシ



## 結果の要約

ベンチマーク実験から次の結論が導かれます。

- FOCモデルと浮動小数点精度を使用すると、LEリソース使用率にわずかに影響があります。ただし、レイテンシの増加はわずか（依然として  $5\ \mu\text{s}$  以下）に留まり、この演算に許容されるレイテンシ・レベルです。
- 精度を 16 ビットに低下させると、データ・パスが狭くなってリソース使用率も低下します。
- フォールディング係数は、1 Msp のスループットを保ちながらハードウェア・リソースを最適化します。そのため、最大 10 チャンネルの 100 ksp FOC アルゴリズムをリアルタイム処理できるようになります。

## まとめ

今日の MCU と DSP は、次世代モーター制御システムでのその性能範囲から外れています。設計者は、モーター制御アルゴリズムを微調整してコストと消費電力を削減する柔軟性を必要とします。市販の DSP ソリューションは、システムを駆動するのに必要なその他のコンポーネントに適合しない、限定された固定小数点機能や浮動小数点機能しか持っていません。

一方、アルテラの FPGA は、動作全体を管理できるプロセッサ、カスタマイズされたサブシステムに容易に接続するための柔軟なインターフェース、複雑なモーター制御ループとアルゴリズムを同時に簡素化する、最適化されたデザイン・フローなどのコンポーネントを統合できるようにします。モーター・システムは、さまざまな高速制御ループ、時限出力パルス周波数、および必要なマルチセンサー・インターフェースとフィルタリングの組合せです。アルテラの FPGA 固有の並列処理能力と高性能可変精度 DSP ブロックは、ボトルネックを減らしてモーター制御システムに最適なソリューションを提供します。

このような FPGA 固有の利点に加えて、アルテラは最適なデザイン手法を提供しています。モーター・アルゴリズム最適化のためのアルテラの DSP Builder、システム統合のための Qsys または SOPC Builder、およびデザイン合成とフィッティングのための Quartus II ソフトウェアは、The MathWorks Simulink/Matlab ツールをモデル化に使用して、最も複雑なドライブ・システムにも対応する包括的な統合デザイン手法です。

## 詳細情報について

- アルテラの産業機器ウェブサイト  
<http://www.altera.co.jp/end-markets/industrial/ind-index.html>
- ホワイト・ペーパー：産業用アプリケーションにおける、設計資産保有の総コストの削減  
<http://www.altera.co.jp/literature/wp/wp-01122-tco-industrial.pdf>
- ホワイト・ペーパー：産業用イーサネットに対応する柔軟性の高いソリューション  
<http://www.altera.co.jp/literature/wp/wp-01037.pdf>
- ホワイト・ペーパー：アルテラのFPGAで構築する、TuV認定機能安全システム  
<http://www.altera.co.jp/literature/wp/wp-01123-functional-safety.pdf>
- オンライン・セミナー：産業機器設計における設計資産保有の総コスト (TCO: Total Cost of Ownership) の低減  
<http://www.altera.co.jp/education/webcasts/all/wc-2010-lower-tco-for-industrial-designs.html>
- ビデオ：3 Ways to Quickly Adapt to Changing Ethernet Protocols  
<http://www.altera.com/education/webcasts/videos/videos-adapt-to-changing-ethernet-protocols.html>
- その他の産業関連ビデオとオンライン・セミナー

## 謝辞

- Kevin Smith, Sr. Member of Technical Staff, Altera Corporation
- Wil Florentino, Sr. Technical Marketing Manager, Altera Corporation
- Jason Chiang, Sr. Technical Marketing Manager, Altera Corporation
- Stefano J. Zammattio, Product Manager, Altera Corporation

## アルテラについて

アルテラは、プログラマブル・ロジックのパイオニアとして、市場において革新性、差異化、成功を高い費用効果で迅速に実現するのに、システム設計者が信頼を寄せることができる革新的なテクノロジーを提供します。アルテラは、ファブレス・ビジネス・モデルによって、技術的に進歩したFPGA、CPLD、およびHardCopy® ASICの開発に注力することができます。

アルテラの工業用グレードのFPGAをコプロセッサまたはSoCとして使用すると、産業用アプリケーションに柔軟性がもたらされます。アルテラのFPGAは、多くの産業用製品に対して高度に統合された単一のプラットフォームを提供するため、大幅な開発期間の短縮とリスクの低減を実現できます。アルテラのFPGAは以下の利点をもたらします。

- ハードIPブロック、エンベデッド・プロセッサ、トランシーバ、およびその他の機能でデザインを統合することによるアプリケーション機能の向上と総コストの削減
- 進化しつつある産業用イーサネット・プロトコルと絶えず変化するデザイン要件をサポートする、現場でも対応可能な再プログラミング可能性
- エンベデッド・プロセッサ、カスタム命令、およびDSPブロックによる性能の拡大

- 産業機器の長いライフ・サイクルをサポートする、陳腐化からの保護と将来の FPGA ファミリへのマイグレーション・パス
- 使い慣れたツール、使い慣れた、強力かつ統合されたツールを使用して、デザインとソフトウェアの開発、IP 統合、およびデバッグを簡素化