

# FPGAベースのシステムで、モーター制御の性能を向上

著者：Andrei Cozma/Eric Cigan

はじめに

産業、民生、自動車など、電気モーターはさまざまな分野で活用されています。その制御は、入力電力を変化させてトルク、速度、位置を調整するモーター制御システムによって行われます。高性能のモーター制御システムを使えば効率が向上するとともに、より高速で高精度な制御が可能になります。高度なモーター制御システムには、制御用のアルゴリズム、産業用ネットワーク、ユーザー・インターフェース（UI）が統合されるため、すべてのタスクをリアルタイムに実行するには、強力な処理能力が必要になります。一般に、最新のモーター制御システムには、マルチチップのアーキテクチャが使われます。例えば、DSPによってモーター制御のアルゴリズムを実行し、FPGAによって高速I/Oとネットワーク・プロトコルの処理を行い、マルチプロセッサが実行制御の処理を担うといった具合です<sup>1</sup>。

Xilinx社は、汎用性の高いCPUと処理能力の高いFPGAを組み合わせたSoC（System on Chip）「Zynq All Programmable SoC（以下、Zynq）」を製品化しています。この製品を利用すれば、モーターを制御する機能とその他の処理タスクを単一のデバイスに統合することが可能です。制御アルゴリズムやネットワーク機能、集中的な処理を要するタスクは、プログラマブル・ロジックにオフロードすることができます。一方で、監視制御、システムの監視/診断、UI、コミッショニングなどはプロセッシング・ユニットによって実行することが可能です。プログラマブル・ロジックには、制御用のコアを複数搭載することができます。それらを並列に動作させれば、複数の軸に対応する機械や、複数の制御システムを実現することが可能です。1チップにコントローラのすべてを実装することで、よりシンプルで信頼性が高く、低コストのハードウェア設計が行えます。

一方、MathWorks®社は、ソフトウェア・モデリングとシミュレーションを行うためのツール「Simulink®」を提供しています。このツールを使用すれば、モデルの作成から実装までを完全に網羅する設計フローによって、モデル・ベースの設計を行うことが可能になります<sup>2</sup>。モデル・ベースの設計では、設計作業の場が研究施設や現場からデスクトップ環境へと移行できます。また、技術者や研究者が行う作業の方法にも変化をもたらします。最近では、プラントからコントローラまでを含むシステム全体をモデル化することが可能です。また、技術者はコントローラを現場に配備する前に、動作のチューニングを行うことができます。これによって、故障のリスクが軽減されるうえに、迅速なシステム統合が可能になり、装置の可用性に対する依存度が低下します。完成した制御用モデルは、Simulinkの環境において、制御システムで実行可能なC言語のコードまたはHDLのコードに自動変換することができ、時間短縮と手作業によるコーディング・エラーの回避につながります。システムのモデルを実際の条件下でコントローラの動作を観測できるラピッド・プロトタイプ環境とリンクさせることによって、リスクはさらに軽減されます。

ここでは、モーター制御の性能を向上させるための完全な開発環境を紹介します。この環境は、コントローラを実装するためのXilinx社のZynq SOC、モデル・ベース設計と自動コード生成のためのMathWorks社のSimulinkに加え、ドライブ・システムのラピッド・プロトタイプを行うためのアナログ・デバイス（ADI）の「Intelelligent Drives Kit」を使って構築されました。

## Zynqを用いたモーター制御ソリューション

高度なモーター制御システムでは、制御、通信、UIのタスクを組み合わせる必要があるため、処理に必要な帯域幅の要件やリアルタイム性に関する制約は

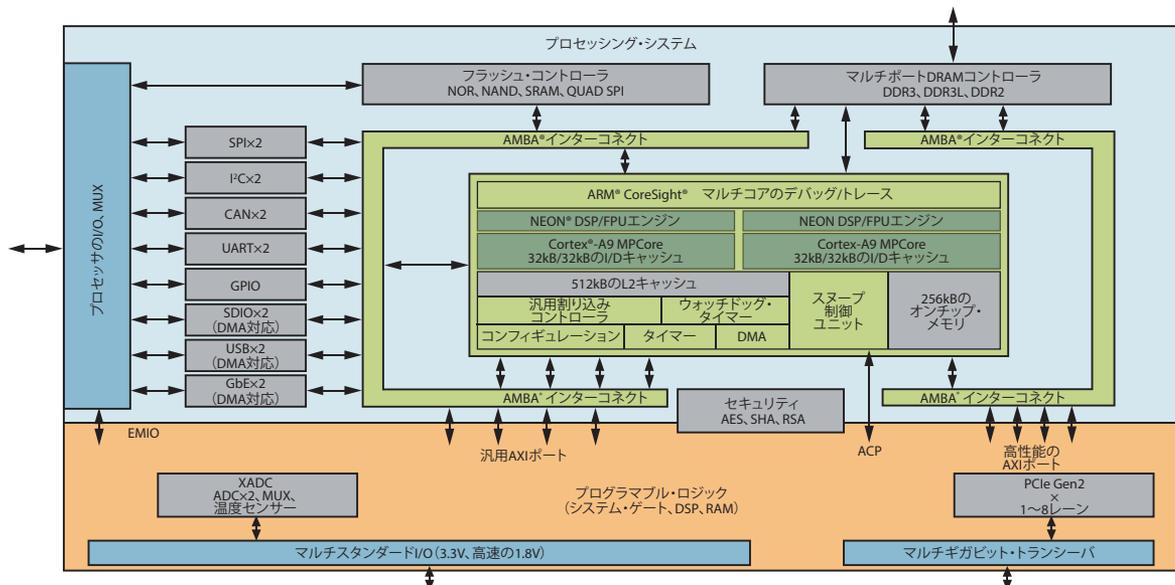


図1. Zynqのブロック図

タスクごとに異なります。このような制御システムの実装に使用するハードウェア・プラットフォームは堅牢かつ拡張可能なものである必要があります。加えて、システムにおいて将来的に生じる改善や規模の拡大にも対応できるものでなければなりません。

Zynqは、高性能のプロセッシング・システムとプログラマブル・ロジックを組み合わせたものです(図1)。この組み合わせによって、優れた並列処理能力、リアルタイム性能、高速演算性能、多くの用途に対応可能な接続性が実現されるため、上述したような要件を満たすことができます。Zynqは、システム内または外付けのアナログ・センサーを監視するために、Xilinx社製のA/Dコンバータ(XADC: Xilinx Analog to Digital Converter)を2個集積しています。

Zynqのプロセッシング・ユニットは、デュアルコアの「ARM Cortex-A9」、コプロセッサの「NEON」、ソフトウェアの実行を高速化するための浮動小数点向け拡張回路で構成されています。プロセッシング・システムは、ソフトウェアの実装にうまく適合した監視制御やモーション制御、システム管理、UI、遠隔保守機能などのタスクを実行します。システムの能力を活用するために、組み込みLinuxやリアルタイムOSを配備することが可能です。プロセッサは自己完結型のものであり、プログラマブル・ロジックを構成(コンフィギュレーション)していなくても使用可能です。このため、ソフトウェア開発者は、FPGAファブリックを設計するハードウェア技術者の作業が完了するのを待つことなくコードを開発することができます。

プログラマブル・ロジック部には、最大44万4000個のロジック・セルと2200個のDSPスライスを集積しており、これにより莫大な処理能力が得られます。FPGAファブリックはスケラブルなものなので、2万8000個程度のロジック・セルから成る小さなデバイスから、極めて負荷の高い信号処理に対応可能なハイエンドのデバイスまでの選択肢があります。プログラマブル・ロジックは、AMBA (Advanced Microcontroller Bus Architecture) -4 AXI (Advanced Extensible Interface) に対応する5つの高速インターコネクトによって、プロセッシング・システムに密に結合されています。これにより、3000本以上のピンを使用することに相当する有効帯域幅が実現されています。プログラマブル・ロジックは、リアルタイム対応の産業用イーサネット・プロトコルなど、タイム・クリティカルで集中的な処理を要するタスクの実装に適しています。また、複数の軸に対応する機械や、複数の制御システムに必要な並列処理を実現可能な複数の制御用コアを構築することもできます。

Zynqをベースとするソリューション/プラットフォームは、磁界方向制御(FOC: Field Oriented Control)のような今日の複雑な制御アルゴリズムにも対応できます。また、Xilinx社とQDESYS社が設計したRPFM (Regenerative Pulse Frequency Modulator) <sup>3</sup>のような複雑な変調方式において、タイミングと性能に関して求められる厳しい要件にも対応可能です。

### Simulinkを用いたモデル・ベース設計

Simulinkは、複数のドメインにまたがるシミュレーションとモデル・ベース設計に向けたブロック・ダイアグラム・ベースの開発環境です。制御用のアルゴリズムやプラントのモデルを含むシステムのシミュレーションに

非常に適しています。モーター制御に使うアルゴリズムは、高精度な位置決めを行うことを主な目的として、速度やトルクなどのパラメータを調整します。モーター制御の設計の適合性を判断するために、制御アルゴリズムの評価をシミュレーションによって行うというのは、非常に有効な手段です。コストのかかるハードウェア・ベースのテストに移行する前に、アルゴリズムについて十分に検証することで、開発にかかる時間とコストを削減できるからです。図2に示したのは、モーター制御アルゴリズムを開発するための効率的なワークフローです。

- モーター、駆動用電子回路、センサー、負荷のライブラリを使用して、コントローラとプラントの正確なモデルを構築する
- システムの動作をシミュレーションし、コントローラが想定どおりに機能することを検証する
- リアルタイム・テストと実装用にC言語/HDLのコードを生成する
- プロトタイプハードウェアを用いて制御アルゴリズムのテストを行う
- シミュレーションと、プロトタイプハードウェアで行うテストによって、制御システムが要件を満たすことを実証したうえで、コントローラを最終製品のシステムに配備する

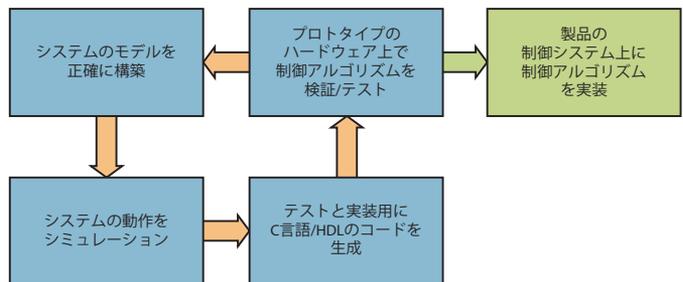


図2. モーターの制御アルゴリズムを開発するためのワークフロー

MathWorks社は「Control System Toolbox™」、「SimPowerSystems™」、「Simscape™」などの製品を提供しています。これらは、線形制御システムを体系的に解析/設計/チューニングするための業界標準のアルゴリズムとアプリケーションを備えています。加えて、機械、電気、油圧などの物理的ドメインにわたり、システムをモデル化してシミュレーションするためのコンポーネント・ライブラリと解析ツールも備えています。これらのツールによって、プラントやコントローラについて忠実度の高いモデルを作成し、実装を行う前に制御システムの動作と性能を検証することができます。シミュレーションは、機能のコーナー・ケースや極端な条件下における動作の検証に最適な手法です。シミュレーションでは、そのような状況にも対応できるコントローラを準備でき、実際の動作が装置とオペレータにとって安全であることを確認することができます。

シミュレーション環境では、「Embedded Coder」と「HDL Coder」の各ツールを用いて制御システムを完全に検証します。両ツールを使えば、C言語/HDLのコードを生成し、テストに使用するプロトタイプハードウェアに配備することができます。それらのコードは、検証後に最終製品のシステムに配備することも可能です。

この時点では、固定小数点やタイミングに関する要件など、ソフトウェアとハードウェアの実装についての仕様が定められます。コードの自動生成機能は、概念設計からシステム実装までに要する時間の短縮に貢献します。また、人為的なコーディング・エラーが削減されるので、実装がモデルとマッチしているかを保証します。図3は、Simulinkにおいてモーター用のコントローラをモデル化し、それを最終製品のシステムに移行するまでに必要な手順を示しています。

最初の作業は、Simulinkでコントローラとプラントをモデル化してシミュレーションを実施することです。この段階で、ソフトウェアで実装するブロックとプログラマブル・ロジックで実装するブロックに制御アルゴ

リズムを分割します。シミュレーションと分割が完了したら、Embedded CoderとHDL Coderを使用してコントローラのモデルをC言語のコードとHDLのコードに変換します。Zynqベースのプロトタイプ・システムによって制御アルゴリズムの性能を検証し、製造段階に移行する前に、コントローラのモデルをさらにチューニングします。製造段階では、自動生成されたC言語/HDLのコードを複雑な製品システムのフレームワークに実装します。このワークフローによって、製造段階に達した時点で制御アルゴリズムの検証/テストが完全に行われたことが保証されます。その結果、システムの堅牢性の面で高い信頼性を確保することが可能になります。

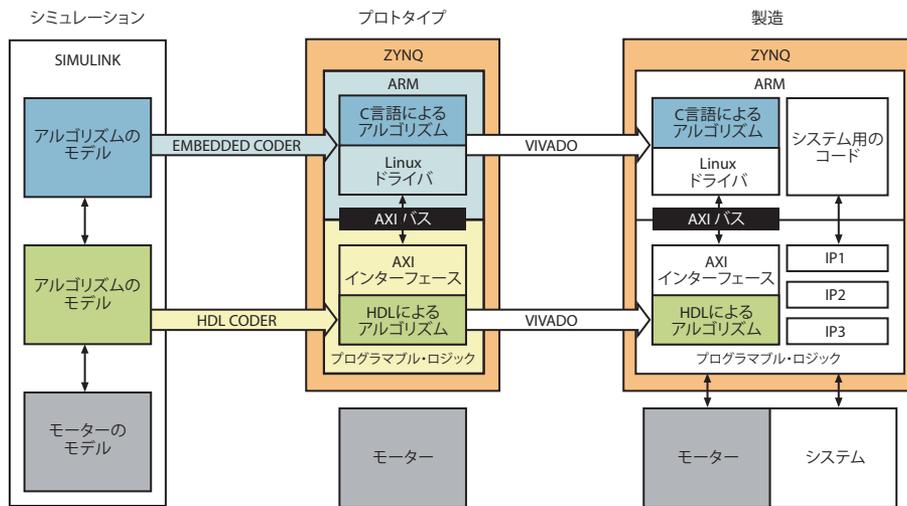


図3. シミュレーションから製造までの流れ

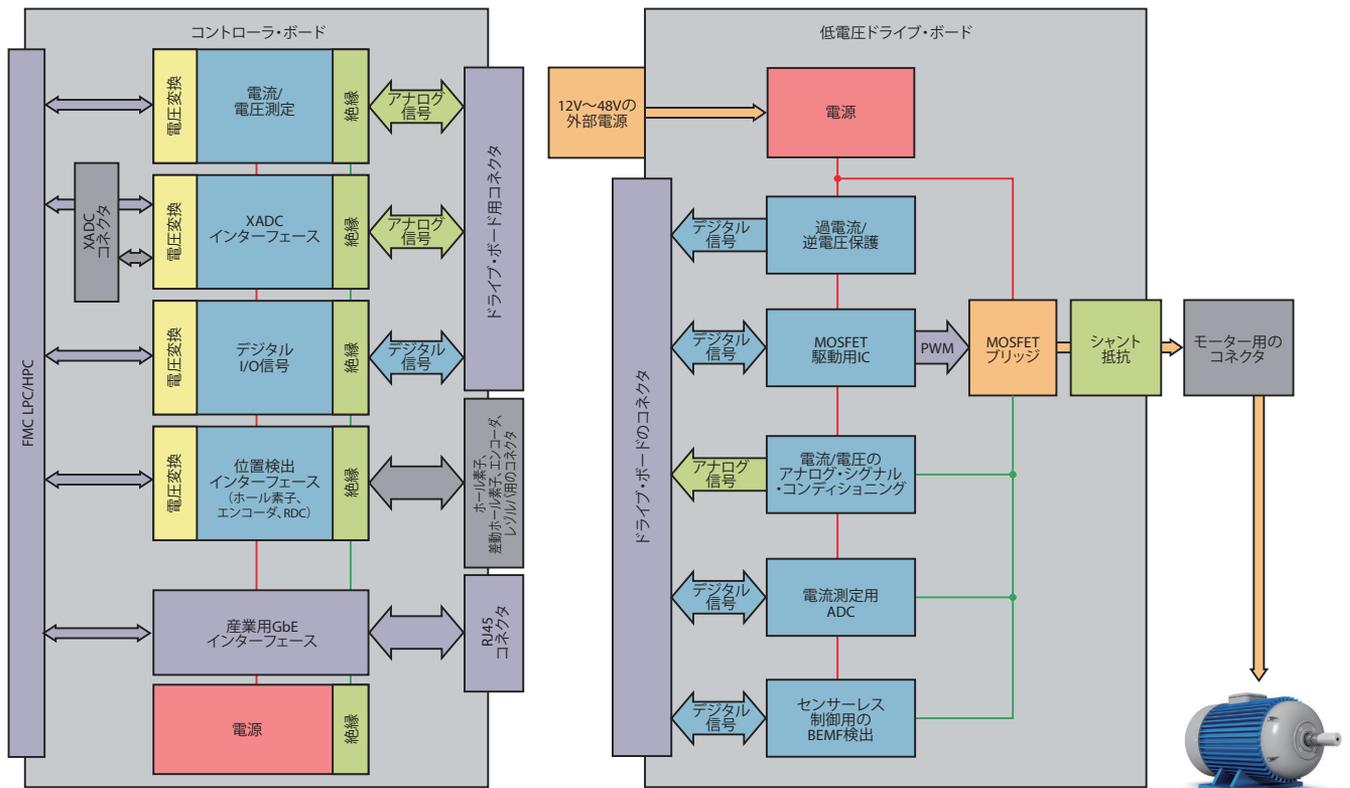


図4. AD-FMCMOTCON1-EBZのブロック図

## ADIのIDKによるラピッド・プロトタイピング

プロトタイプ用に適切なハードウェアを選定するのは、設計における最も重要な作業です。ADIのIntelligent Drives Kit (以下、IDK) は、迅速で効率的なプロトタイピングを可能にします。Avnet社はこのIDKとZynqを組み合わせた開発キット「Zynq-7000 All Programmable SoC/Analog Devices Intelligent Drives Kit」を提供しています。このキットでは、Zynqが備えるARM Cortex-A9に、28nmプロセスで製造されたプログラマブル・ロジック、ADIの最新世代の高精度データ・コンバータとデジタル・アイソレータを組み合わせています。これにより、高性能なモーター制御と、2GbE (2 Gigabit Ethernet) の産業用ネットワークへの接続が可能になります。また、このキットには、Avnet社のベースボード「ZedBoard 7020」と、複数種のモーターに対する効率的な制御機能を備えたADIの完全なドライブ・システム・モジュール「AD-FMCMOTCON1-EBZ」が付属しています。さらに、このキットは、ADIのダイナモメーター駆動システム「AD-DYNO1-EBZ」によって拡張することが可能です。AD-DYNO1-EBZは動的に調整が可能な負荷であり、モーター制御の性能をリアルタイムにテストする際に利用できます。AD-FMCMOTCON1-EBZは、図4に示すようにコントローラ・ボードとドライブ・ボードで構成されています。

コントローラ・ボードはミックスドシグナル対応のFPGAメザニン・カード (FMC) です。LPC (Low Pin Count) またはHPC (High Pin Count) のコネクタによってXilinx社製の任意のFPGAやSoCプラットフォームに接続できるように設計されています。このボードは以下のような機能を備えています。

- 絶縁型A/Dコンバータ (ADC) を用いた電圧/電流測定
- XADCに対する絶縁型インターフェース
- 完全絶縁型のデジタル制御信号/フィードバック信号
- ホール素子、差動ホール素子、エンコーダ、レゾルバ向けのインターフェース
- 産業用の高速通信プロトコル (EtherCAT、PROFINET、Ethernet/IP、Powerlink など) に対応可能な2GbEの物理層デバイス
- FMCの全電圧レベルにおけるシームレスな動作を実現するFMC信号電圧調節インターフェース

すべてのモーター制御システムでは、コントローラとユーザーを保護するために、絶縁が重要になります。コントローラ・ボード上のアナログ信号/デジタル信号を完全に絶縁することによって、モーター・ドライブ側で生じ得る危険な電圧からFPGAプラットフォームが常に保護されることが保証されます。

ドライブ・ボードは、モーターの駆動に必要なすべてのパワー・エレクトロニクスに加えて、電流/電圧の検出回路と保護回路を搭載しています。このボードは以下のような機能を備えています。

- ブラシレスDC (BLDC) モーター、永久磁石同期モーター (PMSM)、ブラシ付きDCモーター、ステッピング・モーターを、12V~48Vの範囲において最大18Aの電流で駆動
- 動的な制動機能、過電流/逆電圧に対する総合的な保護機能
- 絶縁型ADCを用いた相電流測定。プログラマブル・ゲイン・アンプによって電流測定用の入力範囲を最大限に拡大可能
- コントローラ・ボードに対するDCバス電圧、相電流、総電流フィードバック信号の供給
- PMSM、BLDCモーターに対するセンサーレス制御用の逆起電力 (BEMF: Back Electromotive Force) ゼロ交差検出機能

ダイナモメーターは、動的に調整が可能な負荷であり、モーター制御の性能をリアルタイムにテストする際に利用できます。固定的に直接結合された2つのBLDCモーターで構成されます。図5に示すように、一方のBLDCモーターは負荷として機能し、ダイナモメーターの組み込み制御システムによって制御されます。もう一方のモーターはADIのIDKによって駆動されます。このシステムは、負荷電流と速度に関する情報を表示するUIを備えており、さまざまな負荷プロファイルの設定が行えます。また、USBベースのオシロスコープ「Analog Discovery」(Digilent社製) を使用することで、外部から制御することができます。MathWorks社の「Instrument Control Toolbox™」を使用し、「MATLAB®」から直接、負荷の信号の取得と制御を行うことが可能です。

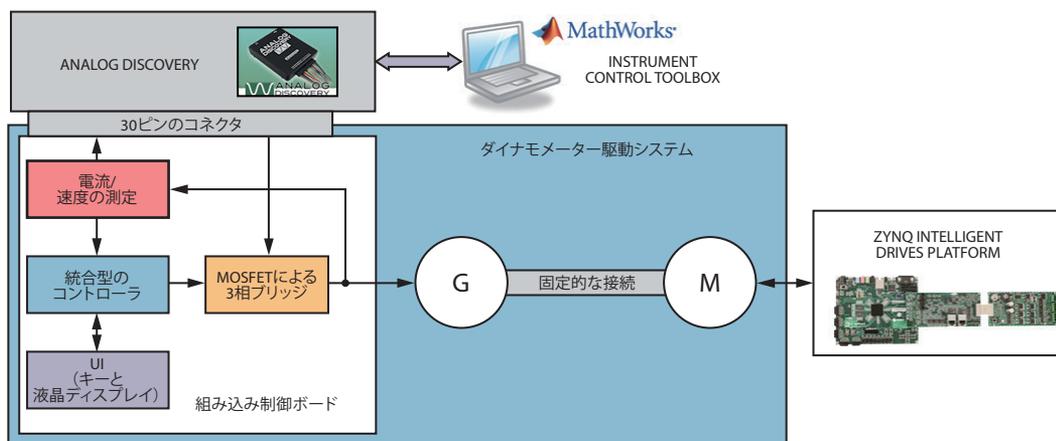


図5. ダイナモメーターの駆動システム

すべてのモーター制御システムにおいて性能を大きく左右するのは、モーターの電流と電圧の測定品質です。ADIのIDKは、高性能のアナログ・シグナル・コンディショニング用コンポーネントとADCを備えており、高い精度で電流/電圧の測定が行えます。図6に示すように、測定経路はコントローラ・ボードとドライブ・ボードの間で分離されています。

相電流はシャント抵抗の両端の電圧を測定することによって算出できます。言うまでもなく、測定精度は最大限に高めたいわけですが、ADCがシャント抵抗の近くにあるか否かに依存して測定経路は2種類考えられます。まずADCがシャント抵抗の近くにある場合には、信号経路は非常に短く、ノイズは結合しにくくなります。シャント抵抗からの小さな差動電圧は、絶縁型のシグマ・デルタ( $\Sigma\Delta$ )モジュレータ「AD7401」によって直接測定されます。インターフェースやシグナル・コンディショニング用の回路を追加する必要はありません。一方、ADCがシャント抵抗から離れている場合には、信号経路が長く、特に電源からのスイッチング・ノイズやモーターからのノイズが結合しやすくなります。このため、ADCとシャント抵抗の間に配置されるプリント回路基板のパターンとシグナル・コンディショニング回路が適切にシールドされるように、十分に注意を払う必要があります。シャント抵抗の小さな差動電圧は、ディファレンス・アンプ「AD8207」によってドライブ・ボード上で増幅されます。AD8207は、ノイズの結合を防ぐためにシャント抵抗の近くに配置します。信号の振幅はフルスケールで $\pm 125\text{mV}$ ですが、ノイズの影響を最小限に抑えるために

$\pm 2.5\text{V}$ に増幅します。増幅された信号は、プログラマブル・ゲイン計装アンプ(PGIA)「AD8251」を用いたもう1つの増幅段を経由します。これにより、入力範囲に合わせて適切にスケールリングされた信号がADCに入力されることが保証されます。増幅後のアナログ信号は、コネクタを介してコントローラ・ボードに伝送されます。コネクタには、各アナログ信号にノイズが結合されないようにするためにシールドが施されています。ドライブ・ボードからのアナログ信号は、オペアンプ「ADA4084-2」によってAD7401の入力範囲に適合するように減衰されます。

電圧/電流の情報をフィードバックするシグナル・チェーンにおいては、絶縁型の2次 $\Sigma\Delta$ モジュレータであるAD7401Aが最も重要な要素となります。この高性能のADCは、ミッシング・コードなしで、16ビットの分解能、13.3ビットのENOB(有効ビット数)、83dBのS/N比を実現します。2線式のデジタル・インターフェースは、20MHzのクロック入力と1ビットのデジタル・ビットストリーム出力で構成されます。ADCの出力は、 $\text{sinc}^3$ のデジタル・フィルタによって再構成(ビット数とサンプリング・レートのフォーマット)されます。データシートには、16ビット、78kHzのサンプリング・レートで出力する場合のフィルタのモデルとHDLによる実装が示されています。出力分解能とサンプリング・レートは、フィルタのモデルとデシメーション率を変更することによって調整可能です。78kHzのサンプリング・レートは、多くのアプリケーションに対して十分な高さですが、もっと高いレートが必要なケースもあり得ます。そのような場合には、図7に示すようにフィルタ・

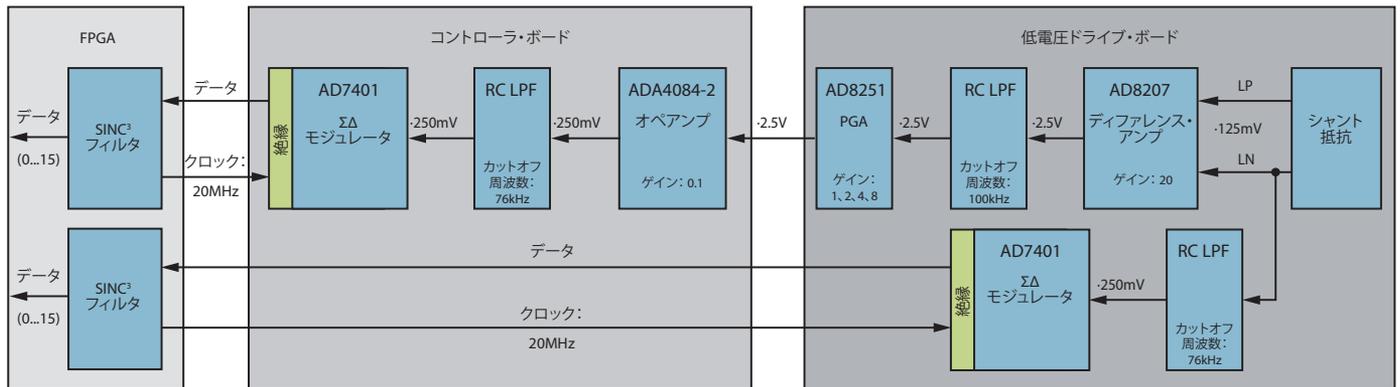


図6. 相電流のシグナル・チェーン

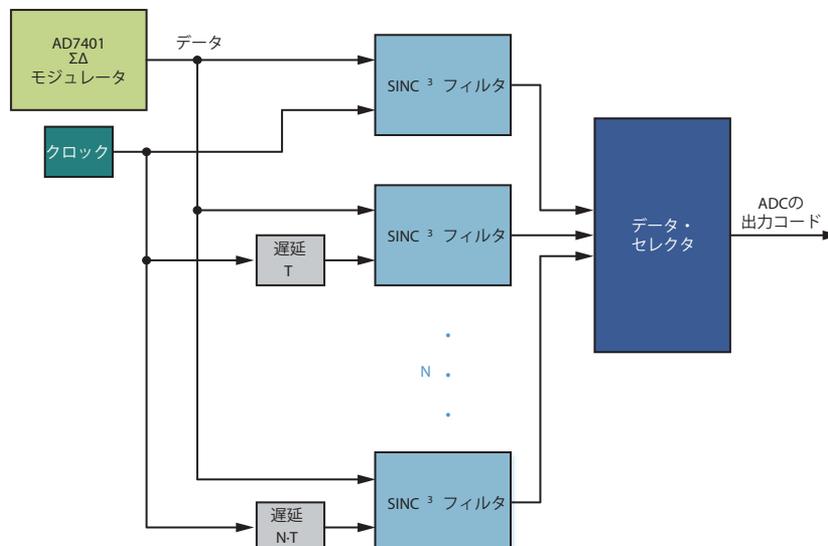


図7. フィルタ・バンク

バンクを使用し、16ビットの分解能でシステムのサンプリング・レートを10MSPS（メガ・サンプル/秒）まで高めることが可能です。フィルタ・バンクでは、 $n$ 個の  $\text{sinc}^3$  フィルタとサンプリング・クロックが使われます。各サンプリング・クロックには、それぞれ  $T$  ( $\text{sinc}^3$  フィルタにおける伝搬時間を  $n$  で割った数) の倍数の遅延が加えられます。データ用のセレクタは、周期  $T$  で A/D 変換の結果となるコードを出力します。

相電流の測定は、Zynq の XADC によって行うこともできます。XADC を使用する場合は信号測定チェーンは、通常の測定チェーンにおいて AD7401 の後段に再構成用のサレンキー型アナログ・フィルタを追加することで構成します。このフィルタは、オペアンプ「AD8646」を用いてコントローラ・ボード上に実装します（図8）。絶縁型の  $\Sigma\Delta$  モジュレータと再構成用のアナログ・フィルタを組み合わせることにより、測定品質を損なうことなく、XADC の入力信号のアナログ絶縁を容易かつ低コストで実現できます。

ADI の IDK には、Simulink ベースのコントローラ・モデル群、Xilinx 社の完全な「Vivado フレームワーク」、そして ADI の Linux インフラストラクチャが含まれています。IDK を使用すれば、シミュレーション、プロトタイプリング、製品のシステムの実装というモーター制御システムの開発に必要なすべての作業を実施することができます。

設計作業は、6ステップのコントローラと PMSM 磁界方向コントローラという2種類のコントローラのモデルを使用して開始することができます。図9に、これら2種類のコントローラの構成を示しました。6ステップのコントローラには、BLDC モーター用に台形波を制御する機能を実装します。FOC コントローラは、制御システムに統合するための FOC コアとして使用できます。

プラントのモデルとコントローラのモデルをシミュレーション段階で作成し、システム全体の動作をシミュレーションすることによって、コントローラが想定どおりに動作するか否かを検証することができます。コントローラのモデルは、C 言語で実装する部分と HDL で実装する部分に分割されます。その際には、コントローラのモデルがハードウェアによる実装と同じように動作するように、タイミング、固定小数点、サンプリング・レート、ループ時間などの制約について定義することになります。図10に、6ステップのコントローラをソフトウェアで実現する部分と HDL で実現する部分に分割した様子を示しました。

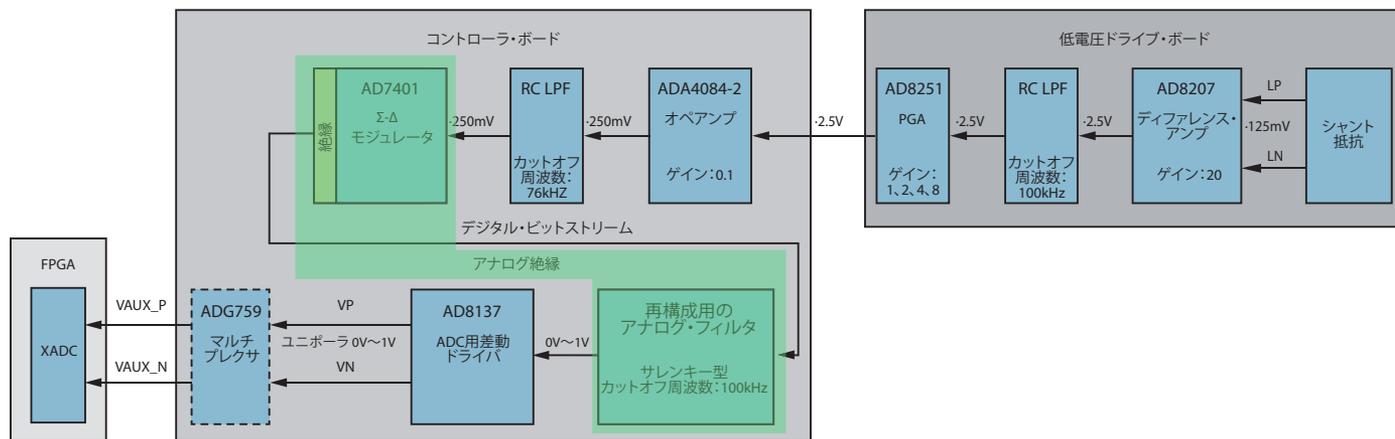


図8. XADC用の信号測定チェーン

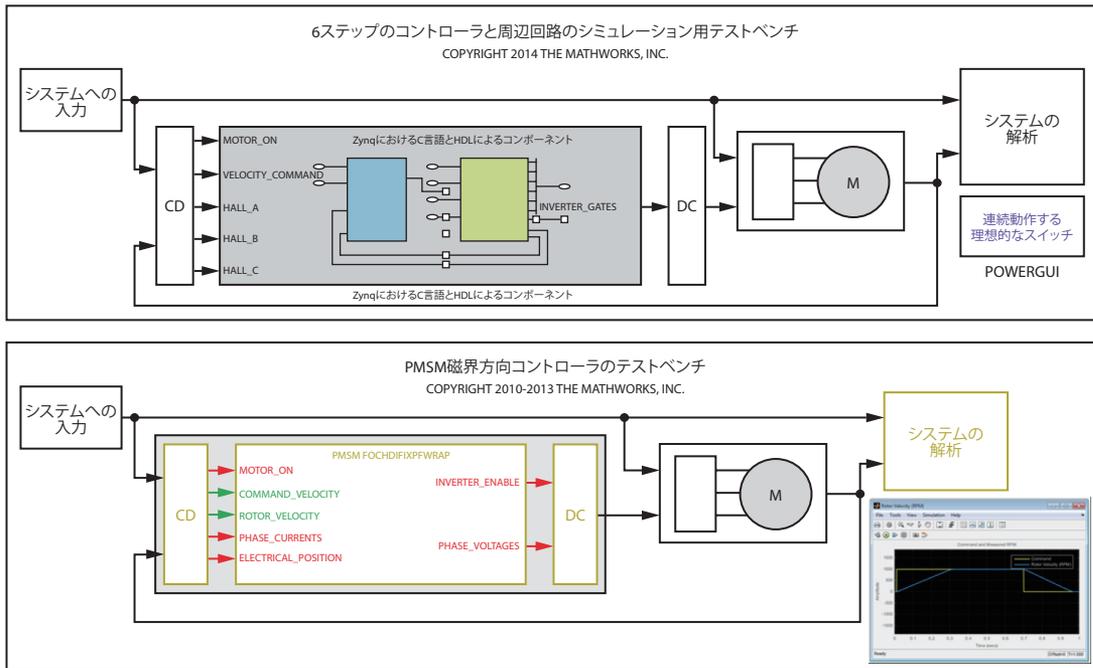


図9. Simulinkによるコントローラのモデル

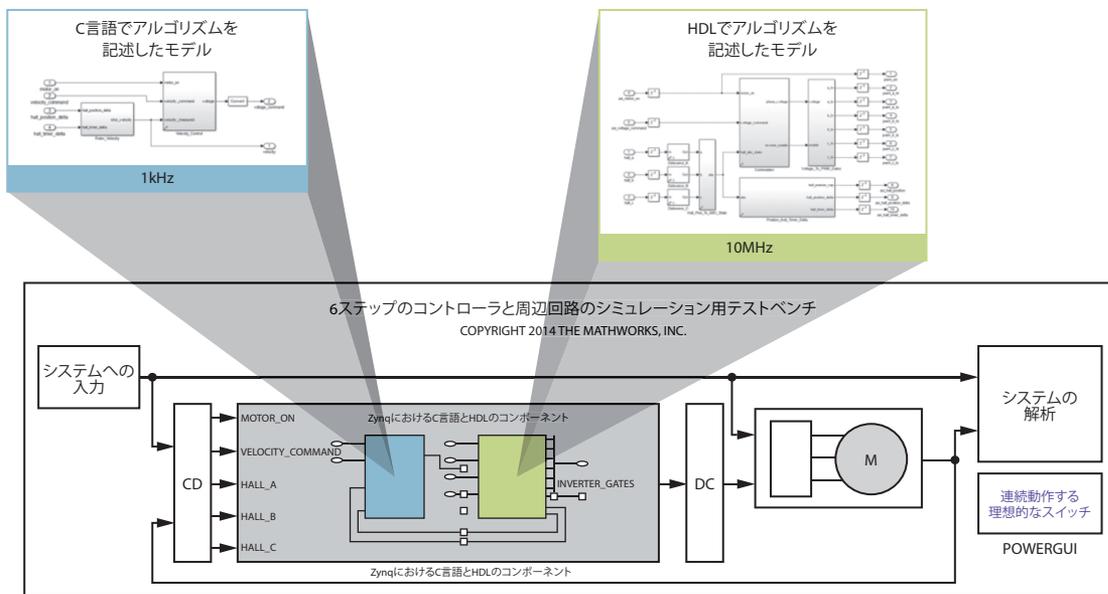


図10. コントローラをC言語による実装とHDLによる実装に分割した様子

シミュレーションによってコントローラを完全に検証したら、次にハードウェア・プラットフォーム上でそのプロトタイプを構築します。Zynqをベースとするワークフローにより、Cortex-A9とプログラマブル・ロジックのそれぞれをターゲットとしてサブシステムに分割されたSimulinkのモデルから、C言語のコードとHDLのコードが生成されます。このワークフローでは、プログラマブル・ロジックをターゲットとするHDLのコードがHDL Coderによって生成されます。一方、Cortex-A9をターゲットとするC言語のコードはEmbedded Coderによって生成されます。MathWorks社のZynqサポート・パッケージを使えば、モデルを基にしたアルゴリズムを実装したC言語のコードを含み、AXIバスとの通信を行うARM用の実

行可能ファイルが生成することができます。また、モデルを基にしたHDLのコードを含み、プログラマブル・ロジックのピンとAXIバスに接続されるビットストリームも生成することが可能です。図11に、コントローラの実装とIDKのハードウェアの関係を示します。

ビットストリームと実行可能ファイルをハードウェア上に読み込めば、コントローラの動作のテストを開始できます。その際には、Simulinkと、オープンソースのLinux OSが稼働する組み込みシステムとの間でイーサネット接続を使用し、HIL (Hardware-in-the-Loop) テストを実施することが可能です。シャフトの速度といったモーターのパラメータをSimulinkによって取得し、シミュ

レーション結果と比較することで、システムの実装がモデルと等価であることを確認します。制御アルゴリズムのテストが完了すれば、コントローラを製品のシステムにも適用することができます。

ADIはIDKと共に、プロトタイプと最終製品の両方に適用可能な完全なVivadoフレームワークとLinuxインフラストラクチャを提供しています。図12に、IDKをサポートするLinuxインフラストラクチャを示しました。このハイレベルの構成図には、ADIのリファレンス設計がZynq上で分割される様子が示されています。プログラマブル・ロジックには、ADCやポジション・センサー、モーター駆動段と接続するためのIP (Intellectual Property) コアを実装します。HDL Coderによって生成されたモーター制御アルゴリズムのHDLコードは、ADIのIPに組み込まれます。すべてのIPは、構成と制御のための低速AXI-Liteインターフェースと、DMA (Direct Memory Access) チャンネルを介してソフトウェアに対してリアル

タイムにデータを転送可能な高速AXI-Streamingインターフェースを備えます。高速イーサネット・インターフェースは、Cortex-A9が備えるMAC周辺回路 (ハード・マクロ)、またはプログラマブル・ロジックに含まれるXilinx社のイーサネットIPを使用して実装することができます。

Cortex-A9は、ADIが提供する「Ubuntu (Linux)」を搭載しています。これには、IDKのハードウェアとの通信に必要なLinux I/Oドライバ、監視/制御用のユーザー空間アプリケーション「IIO Oscilloscope (Scope)」、TCPを介したリアルタイムのデータ取得とシステム制御を可能にするサーバー・ソフト「libio」、リモートのコンピュータ上で動作するクライアント・ソフト、Embedded Coderによって生成されたC言語のコードが組み込まれるオプションのユーザー・アプリケーションが含まれています。

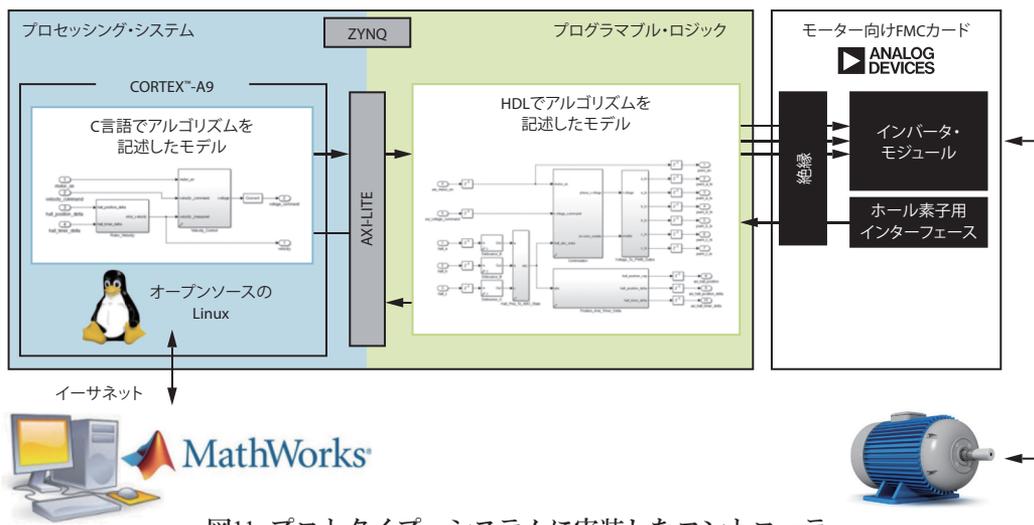


図11. プロトタイプ・システムに実装したコントローラ

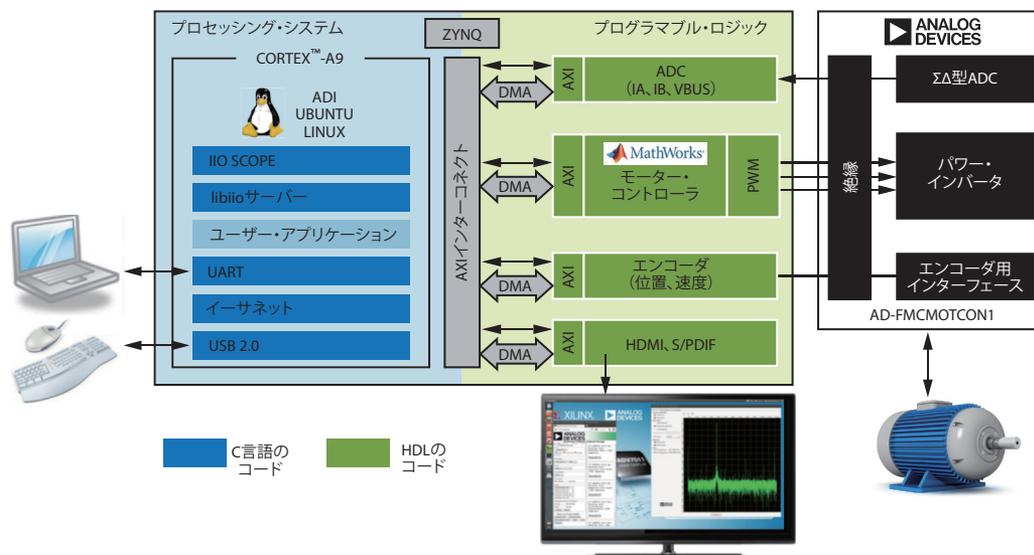


図12. ADIのLinuxインフラストラクチャ

ADIのLinuxドライバは、いずれも「Linux Industrial I/O (IIO) Subsystem」をベースにしています。現在、このサブシステムはすべての主要なLinuxカーネルに含まれています。IIO ScopeはADIが開発したオープンソースのLinuxアプリケーションです。Zynqが内蔵するCortex-A9上で動作し、Zynqのプラットフォームに接続された任意のADI製FMCカードから取得したデータをリアルタイムに表示することができます。表示形式としては、時間領域、周波数領域、コンスタレーションに対応します。また、取得したデータを保存してさらに詳しく解析できるように、CSVファイルやMATLAB形式ファイル（拡張子はmat）といったファイル形式をサポートしています。IIO Scopeは、ADIのFMCカードの構成を変更したり、再度読み込んだりするためのGUIを備えています。

リモートのコンピュータ上では、クライアント・ソフトが実行され、TCPを介したリアルタイムでのデータの取得とシステムの制御が行われます。これを可能にするのがlibioサーバーです。サーバーはLinuxを搭載する組み込みターゲット上で動作します。そして、ターゲットとリモート・クライアントの間で行われる、TCPを介したリアルタイムのデータ交換を管理します。IIOクライアントは、MATLABとSimulinkのネイティブ・アプリケーションに統合されるシステム・オブジェクトとして提供されます。Linuxのインターフェースは、HDMI出力を利用してモニターに表示されます。また、キーボードとマウスはUSB 2.0ポートによってシステムに接続できます。

ADIがIDK向けに提供しているLinuxソフトウェアとHDLインフラストラクチャは、MathWorks社とXilinx社が提供するツールを組み合わせることによって、モーター制御アプリケーションのプロトタイピングに最適な環境をもたらします。また、最終的な制御システムに搭載することのできる製品レベルのコンポーネントも含まれているため、概念設計から製造までにかかる時間を短縮し、コストを削減することが可能になります。

## まとめ

本稿では、FPGA対応の最新モーター制御システムの要件と動向について説明しました。MathWorks社、Xilinx社、ADIは、これらの要件を満たし、より効率的で高精度なモーター制御ソリューションを開発できるように支援するためにツール/システムを市場に提供しています。このツール/システムとは、MathWorks社が提供するモデル・ベース設計ツールとコードの自動生成ツールに、Xilinx社が提供する高性能のZynq、ADIが提供する絶縁/電源/シグナル・コンディショニング/測定用のソリューションを組み合わせたものです。これらを利用することにより、モーターを駆動するためのシステムの設計、検証、テスト、実装をかつてないほど効率的に実施できるようになります。結果として、モーターの制御性能が向上するだけでなく、製品を市場に投入するまでにかかる期間の短縮が可能になります。ADIのIDKとZynqを併用すれば、MathWorks社のSimulinkを使用して設計したモーター制御アルゴリズムに対する優れたプロトタイプ環境が得られます。IDKには、システムを評価する際の出発点として使用でき、任意のモーター制御プロジェクトに着手できるよう支援することを目的とした複数のリファレンス設計<sup>4</sup>が用意されています。

## 関連資料

1. Hill, Tom「Motor Drives Migrate to Zynq SoC with Help from Matlab」 Xcell Journal, Issue 87, Second Quarter 2014年
2. O'Sullivan, Dara/Sorensen, Jens /Frederiksen, Anders「Model Based Design Tools in Closed Loop Motor Control」 PCIM Europe, 2014年
3. Corradi, Dr. Giulio「Fpga High Efficiency, Low Noise Pulse Frequency Space Vector Modulation—Part I」 EDN Network, 2012年10月4日
4. AD-FMCMOTCON1-EBZ User Guide



### 著者：

Andrei Cozma ([andrei.cozma@analog.com](mailto:andrei.cozma@analog.com)) は、ADIのエンジニアリング・マネージャーとしてシステム・レベルのリファレンス設計の開発を支援しています。産業オートメーションと情報科学に関する学士号に加えて、電子工学と電気通信工学の博士号を取得しています。モーター制御、産業オートメーション、ソフトウェア無線、電気通信など、さまざまな分野にわたる設計/開発プロジェクトに従事した経験を持ちます。



Andrei Cozma

Eric Cigan氏 ([Eric.Cigan@mathworks.com](mailto:Eric.Cigan@mathworks.com)) は、MathWorks社の技術マーケティング部門で、SoCとFPGAの設計ワークフローに関する取り組みを行っています。同社に入社する前は、MathStar社、AccelChip社、Mentor Graphics社で技術マーケティングに携わっていました。マサチューセッツ工科大学で機械工学の学士号と修士号を取得しています。



Eric Cigan