

FPGAの電源に対する“世話と餌やり”——正しい方法とその理由

著者: Nathan Enger

Share on   

はじめに

最新のFPGAは、これまでに創り出された中でも特に複雑なICの1つです。最も高度なトランジスタ技術と最先端のアーキテクチャ構造を採用し、驚異的な柔軟性と最大限の性能が実現されています。時の流れと共に技術は進歩しているわけですが、その複雑さは、FPGAを使用するシステムの設計と実装に妥協を強いる要因になっています。その妥協が最も顕著に現れているのが電源です。FPGAが新しい世代へと進化するごとに、電源に対しては、それまで以上に優れた精度、アジリティ（俊敏性）、制御性が求められます。加えて、小型化、効率化、障害検知能力も必要になります。

本稿では、FPGAの代表的な例としてIntel®（旧Altera）の「Arria 10 FPGA」（以下、Arria 10）を取り上げます。最初に、同製品の仕様の中でも制約的なものをいくつか具体的に挙げ、それらが電源の設計に及ぼす影響について説明します。次に、アナログ・デバイセズのパワー・システム・マネージメント（PSM）IC「LTC3887」、「LTC2977」、「LTM4677」を使用して、最良の電力供給ソリューションを実現する方法を紹介します。更に、すべての仕様を適切に満たしつつ、FPGAが適切な効率、速度、消費電力で性能を発揮できるようにするためのプランを示します。

FPGA用の電源の要件——

データシートを読み解く

現代の技術者は、持てる時間の大半をプログラミングに注ぐべきです。適切な電源を設計することに時間とエネルギーを費やすべきではありません。実際、電力を供給するための最良の方法は、要件を満たし、プロジェクトの内容に応じて拡張が可能で、堅牢性と柔軟性に優れる、実績のある設計を踏襲することです。以下では、電源に関するいくつかの重要な仕様とその意味について詳しく説明します。

電圧の精度

FPGAのコア部に供給する電源電圧（以下、コア電圧）は、FPGAの消費電力と性能のバランスを図る上で、非常に重要な項目の1つです。仕様書には、電圧の許容範囲が記されていますが、範囲がわかればそれでよいというわけではありません。何事においてもそうであるように、トレードオフと最適化が必要になります。

表1に示したのは、広く使用されているArria 10¹のコア電圧の仕様例です。Arria 10は1つの例ですが、他のFPGA製品におけるコア電圧の要件を代表する値となっています。許容範囲は、公称電圧の±3.3%に相当します。この範囲内の電圧が供給されれば、FPGAは正しく動作するわけですが、全体像から考えると、より複雑な条件について検討する必要があります。

表1. Arria 10のコア電圧の仕様

シンボル	説明	条件	最小値	標準値	最大値	単位
V _{CC}	コア部の電源電圧	標準的かつ低消費電力	0.87	0.9	0.93	V
			0.92	0.95	0.98	V
		SmartVID	0.82		0.93	V

「SmartVID」と記された行の範囲が0.82V～0.93Vになっている点に注目してください。これは、FPGAがSmartVID²のインターフェースを介してコア電圧を要求する場合には、広範囲の電圧に対応可能であるということを表しています（詳細は後述）。このSmartVIDの仕様は、FPGAに関する基本的な事実を示しています。つまり、製造上の許容誤差と、実装されているロジック回路の設計によっては、標準的な仕様の範囲外の電圧でも動作が可能だということです。必要なスタティック電圧は、個々の製品ごとに異なる可能性があります。電源は、その状況に対応し、適応できるものでなければなりません。

目標は、プログラムされた機能を実行するのにちょうどよい性能レベルを実現しつつ、無駄な電力を消費しないようにすることです。半導体の物理的な特性については、IntelやXilinx®が公開しているデータを参照することで把握できます。例えば、図1を見ると、動的消費電力と静的消費電力の両方が、コア電圧を高くすると著しく増加することがわかります。したがって、FPGAの電源電圧としては、タイミングに関する要件をкаろうじて満たすものの、それ以上にはならないように抑えることが目標になります。過剰に電力を消費しても、性能は向上しないということです。それどころか、むしろ悪影響が生じる可能性の方が高くなります。温度の上昇と共にトランジスタのリーク電流が増え、不要な電力をより多く消費してしまうからです。このような理由から、システムの設計においては、実際の動作点としてのコア電圧を最適化することが不可欠となります。

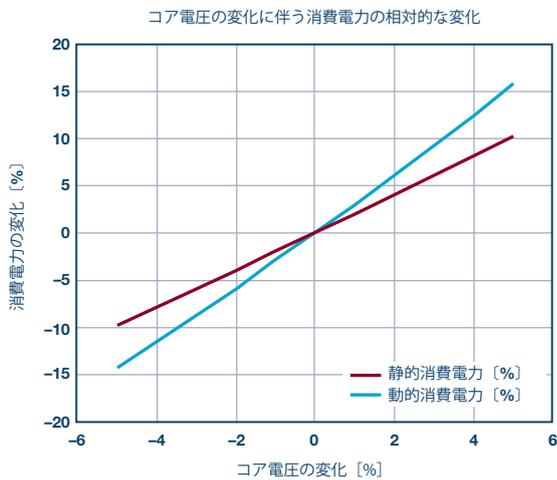


図1. 「Xilinx Virtex-5」の消費電力とコア電圧V_{CC}の関係

そうした最適化を適切に行うには、非常に精度が高い電源が必要です。レギュレータがどの程度不正確であるかを誤差のバジェットに織り込み、それらを使用可能な電圧範囲から差し引いた上で最適化を行わなければなりません。コア電圧が仕様上の最小値を下回ると、タイミング・エラーが発生して、FPGAが正しく動作しない可能性があります。逆に、コア電圧がドリフトして仕様上の最大値を上回ると、FPGAが破損したり、ホールド時間に関するロジック・エラーが生じたりする可能性があります。電源の許容誤差の範囲を考慮し、必ず仕様の範囲内になることが保証される電圧を使用することによって、そうしたあらゆる問題を防止しなければなりません。

問題なのは、ほとんどのレギュレータは十分に精度が高いわけではないという事実です。レギュレータの出力電圧（安定化電圧）は、指定された電圧（コマンド電圧）から許容誤差範囲内の任意の値をとります。また、負荷の条件、温度、使用年数によって、出力電圧がドリフトする可能性があります。±2%の許容誤差を保証している電源の場合、安定化電圧には4%の幅があるということを意味します。電圧が2%低い場合にそれを補償するには、タイミングの条件を満たすために、コマンド電圧として本来必要な値よりも2%高い値を設定する必要があります。その後、レギュレータの出力電圧がコマンド電圧よりも2%高い値にドリフトしたとします。その場合、レギュレータはその動作点に必要な最小電圧よりも4%高い電圧を出力することになります。それでもFPGAが許容可能な電圧範囲内にはあるのですが、かなりの電力を浪費してしまいます（図2）。

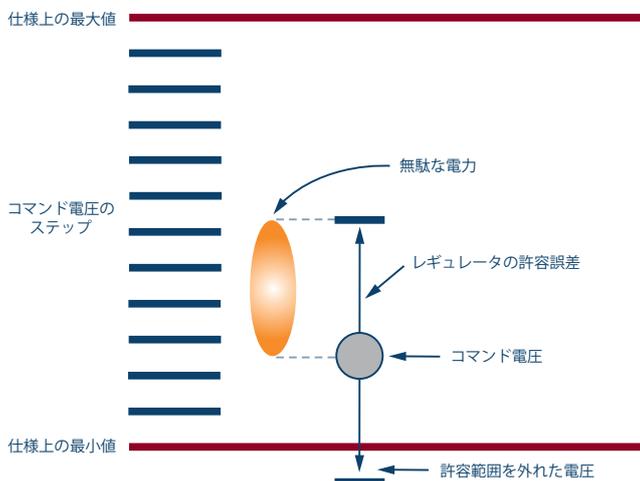


図2. レギュレータの誤差に関するトレードオフ

上記の問題に対する解決策は、許容誤差がはるかに狭いレギュレータを選択することです。例えば、許容誤差が±0.5%のレギュレータであれば、望ましい動作周波数において、仕様上の最小値に近いコマンド電圧を指定することができます。そうすれば、その電圧から1%の範囲内になることが保証されます。当然、FPGAは正常に動作しますし、その動作条件での消費電力を最小限に抑えることが可能になります。

LTC3887を含むDC/DCコントローラ・ファミリ（LTC388x）は、電圧の設定範囲が広いことを1つの特徴とします。また、設定した電圧から±0.5%以内の安定化電圧を出力することも保証されています。一方、LTC2977を含むPSM ICファミリ（LTC297x）は、レギュレータの誤差がトリム後には±0.25%以下になることを保証しています。これだけの精度があれば、任意のFPGAについて、消費電力と性能のトレードオフを最適に行うことができます。

熱の管理

電源の精度は、サーマル・バジェットにも微妙に影響を及ぼします。FPGAの静的消費電力は、無視できるレベルのものではありません。FPGAの温度は、電源を投入していれば、何も処理を実行していなくても上昇します。温度が上昇すれば静的消費電力は更に増加し、それによって動作温度が更に上昇します（図3）。電源電圧を無駄に高くすると、この問題は更に悪化します。電源の精度が低い場合、処理の実行に十分な電圧を確保するために、動作電圧に対するガード・バンドが必要になります。許容誤差のばらつきや、システム・アセンブリのばらつき、あるいは動作温度の変動に起因して電源電圧が一定にならない場合には、必要な最小電圧よりもはるかに高い電圧が生成される可能性があります。そうした電圧がFPGAに印加されると、熱による副作用が生じ、処理負荷が高くなったときに熱暴走に至るおそれもあります。

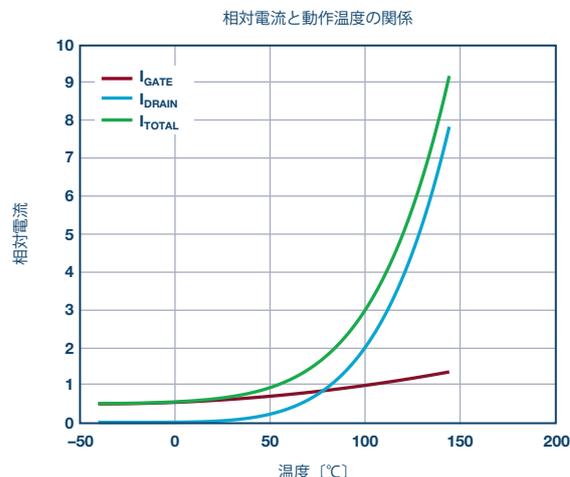


図3. 電源電流と動作温度の関係

この問題に対する改善策は、非常に高精度な電源を使用し、必要以上に高くない最適な電圧を生成することです。アナログ・デバイゼズのPSM ICは、まさにそのようなことを実現できることを特徴とします。

SmartVIDの概要

SmartVIDは、IntelのFPGAに固有の技術です。FPGA自身の要求に応じて電源電圧を設定し、そのFPGAを最適な電圧で動作させるというものです。各FPGAのレジスタには、そのFPGA自身が効率的に動作することが保証される固有の電圧値が（工場プログラムされて）格納されます。

FPGAが備えるコンパイル／書き込み済みIP (Intellectual Property) がそのレジスタの値を読み取り、その値どおりの電圧を供給するように、外部バスを介して電源に要求します (図4)。その電圧に達した後、この機能はスタティックな状態を維持し、FPGAに実装されたアプリケーションが動作します。

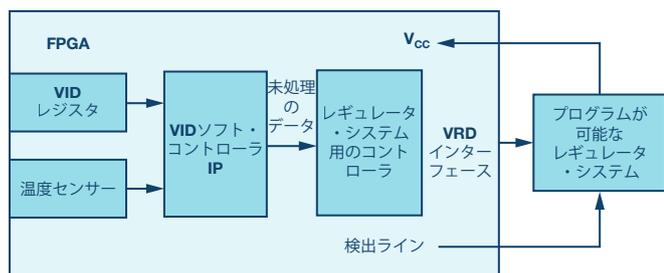


図4. SmartVIDの仕組み

SmartVIDを電源に適用するには、特定のバス・プロトコル、電圧の精度、速度などが必要です。バス・プロトコルは、FPGAからレギュレータに対して必要な電圧を伝達する方法の1つです。使用可能なプロトコルはいくつかありますが、最も柔軟性が高いのはPMBusです。これは、最も広範なパワー・マネージメントICに対応できます。SmartVIDのIPは、VOUT_MODEとVOUT_COMMANDという2つのPMBusコマンドを使用し、PMBusに対応するレギュレータに対して適切な電圧を出力するよう指示します。

レギュレータに対する電圧の精度と速度の要件としては、以下のようなものがあります。

- ▶ (PMBusがアクティブになる前に) 自律的にブート電圧を生成できること
- ▶ 10ミリ秒ごとに新しい電圧コマンドを受け付けられること
- ▶ 電圧の調整フェーズにおいては、10ミリ秒ごとに10mVのステップで電圧を調整できること
- ▶ 10ミリ秒のステップ時間の間に、目標とする電圧から30mV (約3%) の範囲内にセトリングすること
- ▶ 最終的にコマンド電圧を達成し、FPGAの動作中にはスタティックな状態を維持すること

IntelのFPGAではSmartVIDが使用されますが、ほぼ同じ結果が得られる類似の手法は、様々な分野で使われています。非常に簡単な方法としては、工場では各ボードのテストを行い、そのボードで最適な性能が得られる正確な電圧を、電源回路に含まれる不揮発性メモリにプログラムするというものが挙げられます。この方法であれば、それ以上全く手間をかけることなく、正しい電圧を出力するよう電源を動作させることができます。EEPROMを備えるパワー・マネージメントICやコントローラICを使用する場合、この方法によって大きなメリットが得られます。

LTC388xは、IntelのSmartVIDに対応するために必要なすべての要件を満たします。また、μModule® 製品である「LTM4675」、「LTM4676」、LTM4677を使えば、1つのコンパクトなモジュールという形態で、それらの要件を完全に満たすソリューションを実現できます。

タイミング・クロージャ

ロジック・ブロックの演算速度は、電源電圧に依存します。許容範囲内で最も高い電源電圧を選択すれば、演算

速度も最も高くなります。但し、最高速度を得るために、単純に最大動作電圧で動作させるわけにはいきません。図5に示すように、個々のアプリケーションに対して必要十分な電圧で動作させる必要があります。

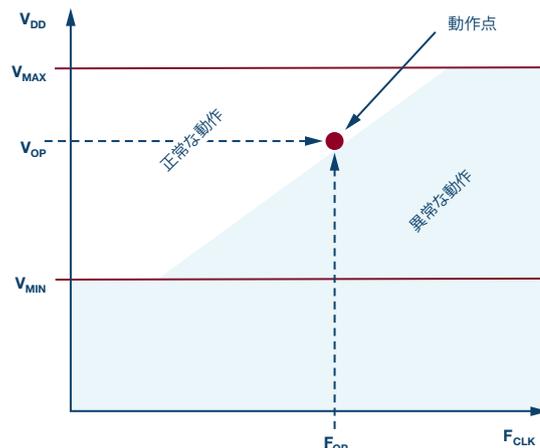


図5. FPGAの動作周波数とコア電圧V_{DD}のトレードオフ

図5は、ある設計がロジックのタイミング要件を満たさず、異常な動作を起こす領域に陥った場合にはどうすればよいのかという重要な情報を示しています。多くの場合、正常動作の領域と異常動作の領域の境界は、設計が確定してハードウェアが製作されるまでは、正確には定義できません。つまり、タイミング要件を満たす具体的な電圧を事前に定めることはできないということです。そのため、選択肢の1つは、最小値よりも十分に高い電圧を事前に定めておき、電力は浪費するものの、機能は保証するというものになります。もう1つの選択肢は、テストを実施する際、またはSmartVIDを起動する際に、ハードウェアのニーズに適応できる柔軟性の高い電源を設計することです。未知の要件に適応するには、アナログ・デバイス用のPSM ICのように高い精度が得られることがますます重要になります。高い精度が得られるなら、FPGAを実際に使用する際、開発の任意の段階で消費電力と性能のトレードオフを実施できるからです。

電源のシーケンス制御

ムーアの法則に従い、最新のFPGAでは、トランジスタの微細化が進んでいます。トランジスタは非常に高速で小型になりましたが、従来よりもはるかに脆弱にもなりました。このような微細なトランジスタを使用することが理由となって、1つのトレードオフが発生します。FPGAには、数億個にも達するトランジスタが集積されます。それらを設計／管理するためには、コア、ブロック、パーティションに分割しなければなりません。その結果、FPGAには、多数の電源ドメインが設けられるようになりました。最近のFPGAの中には、12以上の電源を使用するものもあります。それらすべてにおいて、電圧、電流、リップル、ノイズを管理する必要があります。加えて、起動時 (パワーアップ時) と停止時 (パワーダウン時) のシーケンスも管理しなければなりません。更には、何ら障害が発生していないことを確認する必要もあります。

最近のFPGAの仕様書を見ると、電源の起動時と停止時のシーケンスに関する具体的な要件が定められています。XilinxもIntelも、FPGAを正しくリセットし、消費電流を最小限に維持しつつ、電源電圧の遷移中にI/Oを適切なトライステート構成に維持するための順序とタイミングを推奨しています。各FPGAの電源の数からすると、このシーケンスの制御はかなり複雑になることは明らかです。

Arria 10では、電源を3つのグループ（1、2、3）に分け、1、2、3の順に起動し、3、2、1の順に停止すると定めています³。

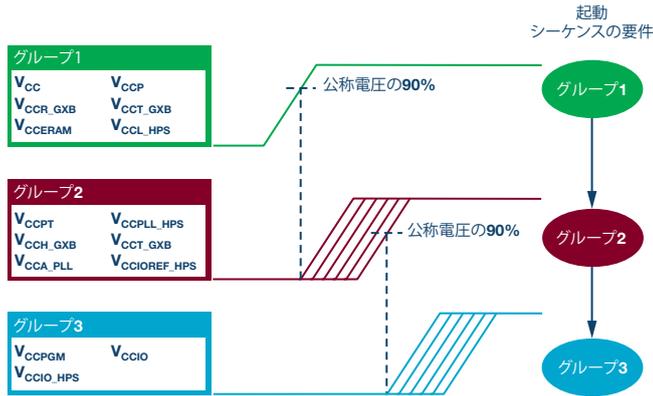


図6. Arria 10の起動シーケンス。起動する順番がグループ単位で定められています。

同様に、XilinxのFPGA「Virtex UltraScale」では、 V_{CCINT}/V_{CCINT_IO} 、 V_{CCBRAM} 、 V_{CCAUX}/V_{CCAUX_IO} 、 V_{CCO} の順に起動することを推奨しています。停止シーケンスは、起動シーケンスと逆になります⁴。

これらの例は、市場に提供されている多数のFPGAのうち、わずか2つにすぎません。ただ、最新のFPGAには、ほぼ間違いなく複数の電源レールがあり、それらをオン/オフする順序について、多くの質問が寄せられているはずです。シーケンスの要件が明示的に定められていない場合でも、確定的なシーケンスを適用することが推奨されます。適用可能な設計上の選択肢としては、以下のようになります。

- ▶ シーケンス制御を行わない：複数の電源を個々に起動/停止させます。但し、この方法には、明らかに問題があります。

- ▶ カスケードなシーケンス制御：ある電源を起動すると、その電源にハード・ワイヤードで接続された電源が次に起動されます。この方法は電源の起動時にしか適用できません。
- ▶ プログラマブル・ロジックによるシーケンス制御：プログラマブル・ロジック（CPLDやFPGA）を使用してカスタムのソリューションを構築します。この方法を使えば高い柔軟性が得られますが、設計者がすべての課題を解決する必要があります。
- ▶ イベント・ベースのシーケンス制御：カスケードなシーケンス制御に似ていますが、より柔軟性が高く、起動時にも停止時にも適用できます。プログラマブルな専用シーケンサICを使うことで、各種の障害が発生した状況やコーナ・ケースに対処することが可能になります。
- ▶ 時間ベースのシーケンス制御：定められた時間に各イベントをトリガします。包括的な障害管理手法と組み合わせることにより、柔軟かつ確定的で安全なシーケンス制御が可能になります。

以下、各選択肢について詳しく説明していきます。

シーケンス制御を行わない

この方法は、何の管理も行わずに電源システムを起動するというものです。主電源が投入される（電源のスイッチをオンにする）と、レギュレータによる安定化制御が始まります。電源が切断される（電源のスイッチをオフにする）と、レギュレータは安定化を停止します。当然のことながら、この方法には多くの問題があります。

まず、タイミングが確定していないと、システムに様々な影響が及ぶ可能性があります。実際、繊細なFPGAに多大な負荷が加わることになります。それによって、直ちに致命的な破損に至る場合もあれば、劣化が促進されて性能が徐々に低下する場合があります。いずれにせよ、好ましいことではありません。予測が不可能なパワーオン・リセットが働いたり、起動時にロジック回路の状態が不定になったりするおそれもあります。

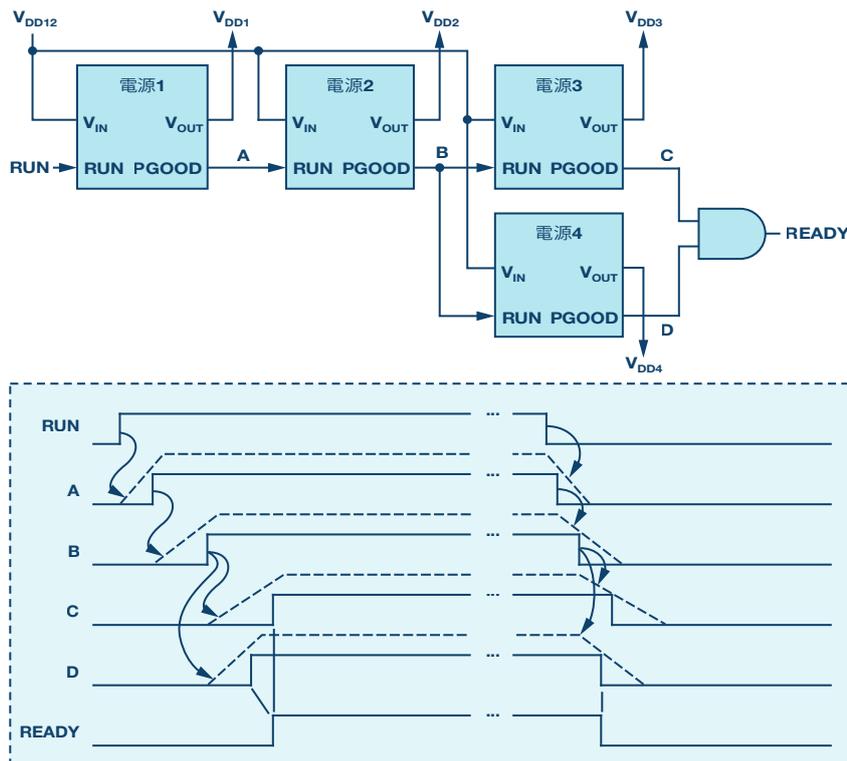


図7. PGOOD-to-RUNの形態のカスケードなシーケンス制御

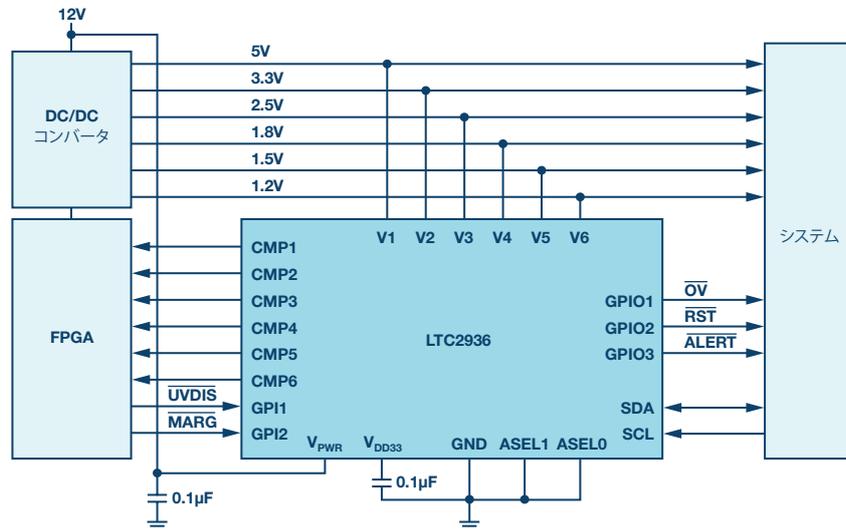


図8. プログラマブルな電圧スーパーバイザの使用例。
この回路ではLTC2936を採用しています。

結果として、システムの安定性が損なわれ、デバッグも困難になります。この方法では、障害の検出/対処、エネルギー管理、デバッグのサポートは全く行われません。一般に、電源のシーケンス制御を行わなければ、最悪の事態を招くことになります。

カスケードなシーケンス制御

上記の選択肢より、もう少し系統的な手法がカスケードなシーケンス制御です。図7に示すようなPGOOD-to-RUNの形態を、ハードワイヤードによる接続によって実現します。いわばドミノ倒しのように、複数の電源を順次起動するというシーケンスにより、定められた順序での起動が保証されます。この方法にはシンプルであるという利点がありますが、残念ながら欠点もあります。最大の欠点は、逆順（または任意の順）に停止することができないということです。また、1つのシーケンスにしか対応できないという欠点もあります。加えて、不確定な動作条件下において、障害への対処やエネルギー管理を適切に行うことができません。何らかの判断を下せるだけのインテリジェンスはなく、シーケンス中のどこかの段階で問題が生じると、その後何が起きるかわかりません。そのため、そうした問題のデバッグも困難になります。

プログラマブル・ロジックによるシーケンス制御

ボード上の補助的なCPLDやFPGAを使用し、電源のシーケンス制御を行う方法です。多くの設計者がこの方法を選択しています。実際、デジタル設計者によってデジタル設計者のために設計されたシステムにおいては、一定のメリットが得られます。そうしたケースでは、デジタル制御用のブロックを設計して補助的なFPGA上にプログラムし、別のFPGAの電源を制御するというのは自然な発想のようにも思えます。しかし、その考え方には落とし穴が潜んでいる可能性があります。電源システムは、デジタル制御を適用する対象としては少し複雑すぎるからです。

電源のシーケンス制御と管理の問題に徹底的に取り組むためには、まずはその複雑さについて十分に把握する必要があります。その多くについては既に説明しましたが、それ以外にも、マイクロ秒の時間単位で発生する可能性のある過電圧/低電圧状態の検出や対処、危険なレベルの電流や温度の検出、テレメトリによる測定結果やステ

ータスの記録といったことが必要になります。また、ハードウェア担当者の作業を支援する起動/デバッグ・サービスについても把握しておくべきです。そうしたすべての検討項目に対応した上で、デジタルのアルゴリズムに加えてアナログの専用ハードウェアを用意しなければなりません。

アナログ・デバイスは、このような難易度の高い課題に挑む設計者を支援するためのアナログ・フロント・エンド製品群を提供しています。「LTC2936」は、デジタル・ビットとアナログ電源の間のインターフェース向けに、堅牢で高精度な6個のアナログ・コンパレータを提供します。その閾値はプログラムが可能であり、高速のイベントを検出してそのステータス（デジタル値）をロジック回路に送信することができます。また、他の機能にも使用できるプログラマブルな3つのGPIOピンも備えています。このプログラマブルなICは、瞬時の起動を可能にするためのEEPROMを内蔵しています。このEEPROMには、I²C/SMBusのインターフェースを介し、テレメトリによって取得した障害に関する情報をデバッグ用に格納することができます。LTC2936の活用方法の例を図8に示しました。

テレメトリによる情報の収集には、高速コンパレータ機能に加えて、A/Dコンバータ（ADC）も必要になります。「LTC2418」は高速なセトリングを実現する24ビットのシグマ・デルタ（ $\Sigma\Delta$ ）方式ADCであり、実績のある選択肢の1つです。同ICが備える4線式のSPI（Serial Peripheral Interface）を使用することにより、最大16チャンネルのアナログ信号をモニタリングすることが可能です。このICをボードに実装したコントローラにより、システム内の多数のポイントをモニタリングし、測定値を即座にストリーミングすることができます。

一般に、CPLD/FPGAを使用した電源のシーケンス制御には、かなり多くの選択肢があります。適切に機能する方法には違いありませんが、これを採用すると1つのリスクを抱えることとなります。設計上の避けられないバグ、想定外のコーナ・ケースや障害の可能性、面倒なサポートの問題を含めて、デジタル設計とアナログ設計の両方に関するすべての責任を、採用を決めた設計者が負う必要があるからです。それよりも簡単に電源システムを構築する方法は、間違いなく存在するはずで

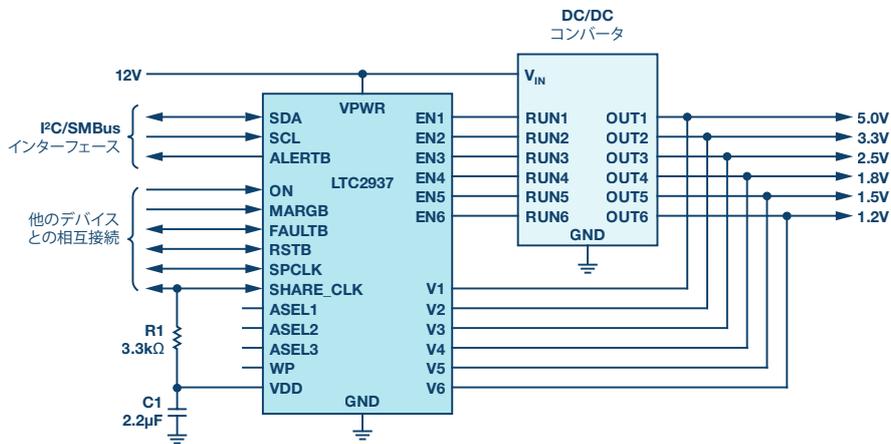


図9. 電源シーケンサ/スーパーバイザIC (LTC2937) の使用例

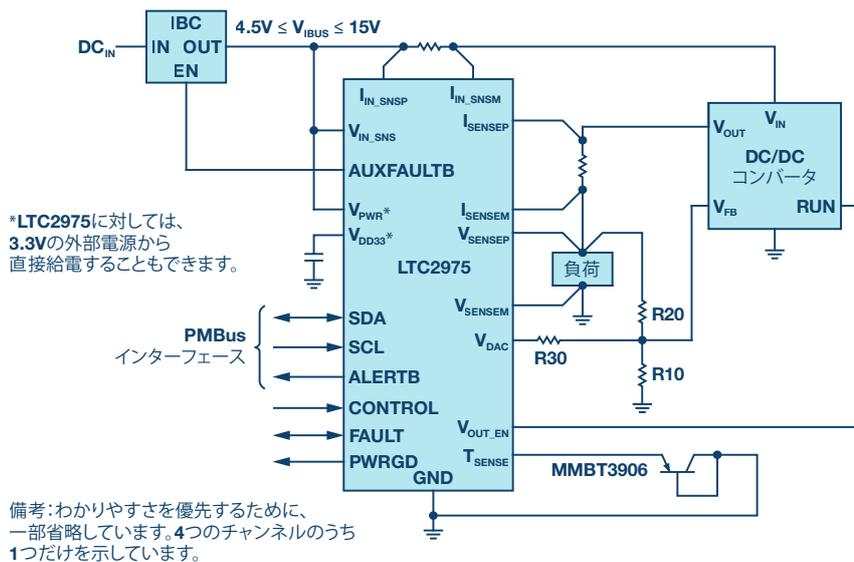


図10. 4チャンネルのPSM IC (LTC2975) の使用例

シンプルなシーケンサ/スーパーバイザの活用

この方法では、シンプルなシーケンサ/スーパーバイザICを使用します。それにより、堅牢なシーケンス制御を導入し、障害への対応を図るというものです。シーケンサ/スーパーバイザICは、電源レールに関するシーケンスを制御すると共に、電源電圧を監視してそれらを仕様の範囲内に維持するという重要な役割を担います。そうしたICの例としては「LTC2928」が挙げられます。これは、外部ピンの設定による構成（コンフィギュレーション）が可能な使いやすいシーケンサICです。起動シーケンスのタイミング（停止時は起動時の逆順）とスーパーバイザとしての機能における電圧の閾値を設定可能です。一方で、デジタル・プログラム機能やテレメトリ機能など、余分な機能は一切提供しません。それでも、同ICだけで要件を満たすことができる可能性はあります。

EEPROMを内蔵するプログラマブルなシーケンサ/スーパーバイザICとしては「LTC2937」が挙げられます。あらゆるデジタル・プログラム機能を備えており、イベント・ベースのシーケンス制御に加え、時間ベースのシーケンス制御にも対応可能です。任意の数の電源に対するシーケンス制御と監視を行い、障害を処理し、障害のステータスをEEPROMにログとして記録することができます。特に、電圧の管理とテレメトリを必要としない用途に対して有用なソリューションです。

PSM ICの活用

PSMのすべてのメリットを包括的かつ最大限に活用したいという場合には、アナログ・デバイゼズのPSM ICを採用するとよいでしょう。この種のICは、以下のような特徴を備えています。

- ▶ 任意の数の電源レールについて、起動/停止シーケンスを自律的に実行できます。
- ▶ 電源電圧を0.5%（場合によっては0.25%）の範囲内という高い精度で制御可能です。
- ▶ テレメトリによって電圧、電流、温度、ステータスを測定/把握し、伝達することができます。
- ▶ 複雑な障害の状況に対して協調的な対応を図ることができます。
- ▶ 障害に関する詳細な情報をEEPROMに記録することが可能です。

シーケンス制御は、タイミング・ハンドシェイクの仕組みを使って実行されます（時間ベースのシーケンス制御）。すべてのICの間でゼロ時間（シーケンスの開始時間）と時間ベースを共有し、シーケンスに関わるすべてのイベントがプログラムされた時間に発生します。それにより、任意の数の電源レールについて、起動/停止シーケンスを自律的に実行できます。

PSM ICの製品ファミリには、スイッチを駆動するドライバやアナログ・ループ制御回路をはじめ、スイッチング・レギュレータのあらゆる側面に対応するコントローラが用意されています。一方、電源マネージャ製品は、外部電源に対するサーボ・ループを備えています。この種のICを使用すれば、スイッチング・レギュレータからLDO（低ドロップアウト）レギュレータに至るまで、あらゆるレギュレータに対して、シーケンス制御、監視、モニタリングといったあらゆる電源管理機能を追加することができます。図10に、電源マネージャの一例として「LTC2975」を示しました。

μModule製品の活用

アナログ・デバイスでは、PSMの機能を備えるμModule製品も提供しています。これはBGAまたはLGAのフットプリントで、単位面積当たりの機能が最も多く、最も集積度の高いソリューションです。この種の製品は、コントローラIC、インダクタ、スイッチ、コンデンサなどを1つのパッケージに収めた完全な電源システムとなっています。「LTM4650」のようなデジタル機能を搭載していないμModuleレギュレータについては、LTC2975によってシーケンス制御機能と管理機能を追加することができます。同じμModuleレギュレータでも、「LTM4676A」などの製品は、独自のPSM機能を備えています。また、システムで使用している他のPSM ICと簡単に連携させることも可能です。

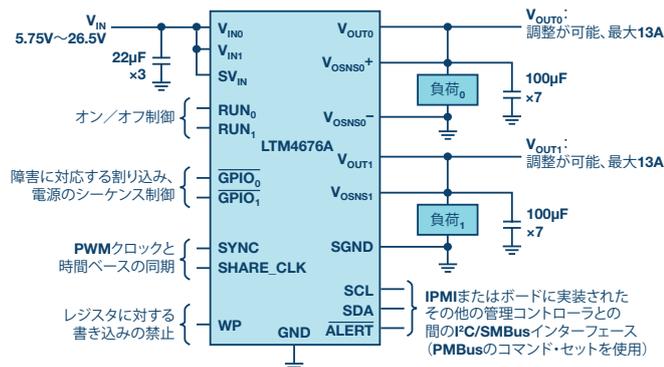


図11. LTM4676Aの概要。13A出力のデュアルPSM機能を備えるμModule製品です。

共有型のシーケンス制御

PSM機能を備えるμModule製品や、マネージャIC、コントローラICは、SHARE_CLKというシンプルな1本のバスを介してタイミング情報を共有することができます。それにより、すべてが連携した状態で起動/停止のシーケンス制御を行うことが可能になります。この1つの信号によって、ゼロ時間とクロック周期に関する情報に加え、シーケンス制御に影響を与える他のステータス情報が、すべてのPSM ICによって共有されます。システム内のすべてのSHARE_CLKピンを接続するだけで、そのような協調動作が可能になります。各ICは、シーケンスのタイミングについて個々にプログラミングされています。その状態で、時間ベースを共有しつつ、イネーブル、ディスエーブル、ランピング、障害が発生した場合のタイムアウトといったイベントのタイミングを、高い精度と信頼性で共有することが可能です。

SHARE_CLKピンは、最も基本的なレベルでは100kHzのクロックに対応するオープンドレイン・ピンとして使用さ

れます。オープンドレイン・ピンの使い方としては、アクティブにプルダウンすることもできますし、そのままの状態バスをフロートさせておくこともできます。バスにつながるすべてのデバイスをそのままの状態にした場合、プルアップ抵抗によって、電圧は3.3Vに引き上げられます。個々のデバイスでは、準備が整うまでの間プルダウンすることによって、クロックの入力を停止することができます。つまり、すべてのデバイスの間で合意が得られなければ、クロックはスタートしません。これは、ゼロ時間を伝達し、クロックの停止によってシーケンスのステータスを示す有効なメカニズムです。

共有型の障害処理

SHARE_CLKピンに似たものに、FAULTバスがあります。システム内の各PSM ICをFAULTという信号線に接続すると、オープンドレイン出力によって、それをロー・レベルに引き下げることができます。また、他のデバイスがロー・レベルに引き下げた場合には、それに対応することも可能です。これは、すべてのPSM ICの間で通信を行い、障害に迅速に対応するためのシンプルな手段となります。動作については完全にコンフィギュラブルです。シーケンス制御を行っている最中でも、定常状態にある場合でも、何らかの障害が発生したら協調してそれに対応することができます。また、障害が発生した際には、システムの状態や障害の原因に関する情報をEEPROMに記録しつつ、定められたタイミングに基づいて電源を停止し、起動シーケンスの再実行を試みるようにシステムを構成することができます。EEPROMに記録する情報は、後で処理を行うためにI²Cバスを介して読み出すことが可能です。

停止シーケンスの制御と蓄積されたエネルギーの管理

システムに電源の停止シーケンスを適用する際には、エネルギーの管理について検討する必要があります。電源の停止シーケンスにおいては、確定的なタイミングを供給することがますます重要になっています。そのためには、システム内に蓄積されたエネルギーがどこで消費されるのか、慎重に検討しなければなりません。大出力の電源は、恐らく数十個にも上る大容量の電界コンデンサを蓄電用のバルク素子として備えているはずですが、それらは電源電圧によって充電されているはずですが、このとき蓄積されているエネルギーは、条件によっては、保護が不十分なデバイスを破壊してしまうほどのレベルに達します。そのような事態を避けるために、FPGAのメーカーは、自社のFPGA製品を保護するための停止シーケンスを定めています。図12に、Arria 10の仕様で定められた停止シーケンスを示しました。

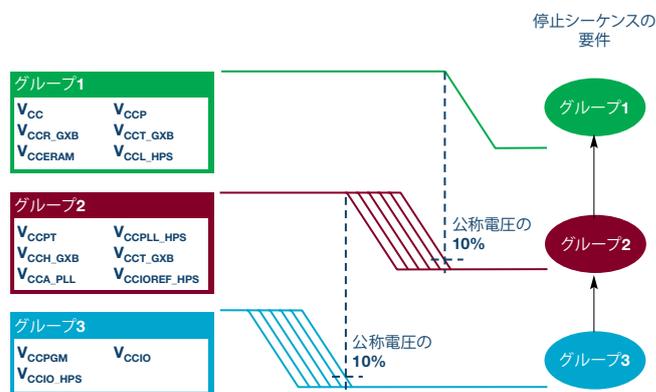


図12. Arria 10の停止シーケンス。グループごとに停止するよう、順序が定められています。

この停止シーケンスは、コンデンサに蓄積されたすべてのエネルギーをどこかに放出し、安全に消費しなければならないことを暗に示しています。これについてはいくつかの方法によって実現可能です。最も簡単な方法は、固定抵抗をコンデンサに並列に接続することです。この抵抗は電源が投入されている間、常に電力を消費します。そこで、抵抗値を十分に大きくし、相対的な損失を最小限に抑えつつ、RC放電の時定数を許容できるレベルまで小さくします。十分に放電するまでにかかる時間としては、RC時定数の倍数（一般的には5倍）が使われます。抵抗による静的な電力消費が許容できるレベル（例えば0.25W未満）になるように最適化しなければなりません。例えば、容量値が1mFで電源電圧が1.0Vの場合、4Ωの抵抗を使用すると時定数τは4ミリ秒となります。その場合、電圧が50mV未満になるまでにかかる放電時間は、13ミリ秒程度となります。抵抗の定格消費電力が少なくとも0.25Wあれば、この方法で十分です。システムは0.25Wという一定の損失で動作し、13ミリ秒で放電が完了するということになります。

多少複雑にはなりますが、上記の方法よりも更に安全な方法があります。それは、放電しなければならないときだけコンデンサに抵抗を接続するというものです。つまり、スイッチ（FET）を制御し、必要なときにバルク・コンデンサの電荷を放電するということです。追加した固定抵抗によって常に電力を消費するのではなく、その固定抵抗とFETの抵抗によって、安全に電力を消費することができます（図13）。

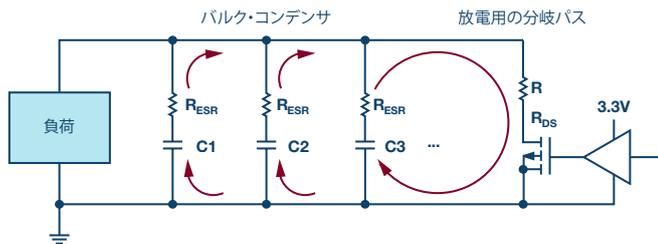


図13. コンデンサの放電回路。固定抵抗を接続したパスをスイッチ（FET）で制御します。

この方法を採用するにあたっては、制御、放電時間、消費電力など、いくつかの事柄について検討する必要があります。例えば、適切な時間に放電用のFETを閉じるには、そのための制御信号が必要です。また、スイッチ（NMOSトランジスタ）を制御する信号は、その閾値電圧V_{TH}を十分に上回っていなければなりません。一般的なNMOSトランジスタを使用する場合、そのゲートを駆動するためには3V~5Vもの電圧が必要になります。

標準的な電解コンデンサには数百mΩの等価直列抵抗（ESR）があります。コンデンサが放電する際には、それによってエネルギーの一部が消費されます。しかし、場合によってはトータルの容量が数十mFにも達するほど多くのコンデンサが並列に接続されるので、トータルのESRは数十mΩ以下に抑えられます。したがって、コンデンサのESRは、蓄積されたエネルギーのうち、ほんの一部しか消費しないと見なすべきです。

妥当な時間でコンデンサの放電を行うには、RCの時定数を、（電圧を数mV未満まで引き下げるために）必要な放電時間の1/5未満に抑えなければなりません。この値は、コンデンサのトータルの容量値、FETと直列抵抗Rのトー

タルの抵抗値、並列に接続されたR_{ESR}の合成抵抗値を使った簡単な計算によって得ることができます。

$$\tau = \left[\frac{R_{ESR} + R + R_{FET}}{N} \right] \times [C_1 + C_2 + \dots + C_N] \quad (1)$$

上式において、Nは並列に接続されたコンデンサの数を表します。

より大きな回路で、容量値が計50mFに達するコンデンサを備えており、FETのドレイン-ソース間の抵抗R_{DS}と固定抵抗Rの合計が500mΩであるとする、電圧が50mV未満に低下するまでに約125ミリ秒かかります。この間のピーク電流（または電力）は、1V÷500mΩ = 2A（または2W）です。蓄積されたエネルギーの大部分は、放電が始まってから時定数の2倍の時間までに消費されます。このことから、FETの安全動作領域のグラフを確認すれば、直列抵抗が必要かどうかを判断することができます。FETの安全動作領域の例を図14に示しました⁶。この例では、FETは破損する心配はなく、2Wのパルスに10秒間以上耐えることができます。しかし、このFETのR_{DS}は20mΩ未満なので、480mΩの直列抵抗が別途必要です。また、ほとんどの電力は、直列抵抗で消費されることとなります。したがって、直列抵抗はその際の発熱に耐えられる大きさのものでなければなりません。一般に、パルスの持続時間は、抵抗の熱時定数よりもはるかに短くなります。これについては、抵抗製品のデータシートで詳しい情報を確認してください。

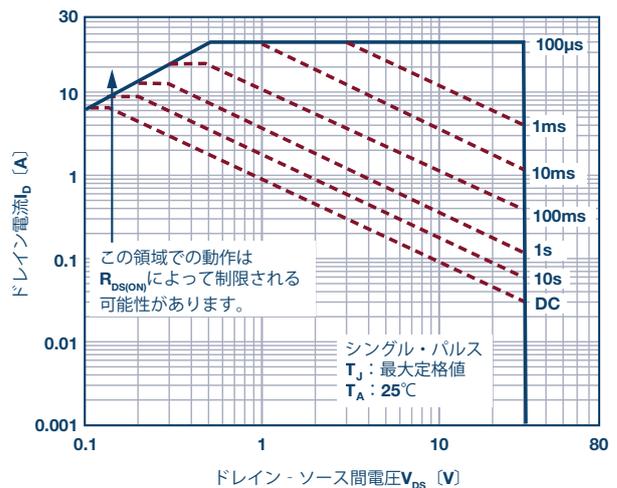


図14. FET（NMOS）の安全動作領域

広範な条件下で安全にエネルギーを放出できるものが、最も堅牢な放電回路だと言えます。図15に示したのは、確実に機能する放電回路です。ON Semiconductor®のFET「FDMC8878」と、1210サイズで値が0.5Ωの表面実装型抵抗を使用しています。

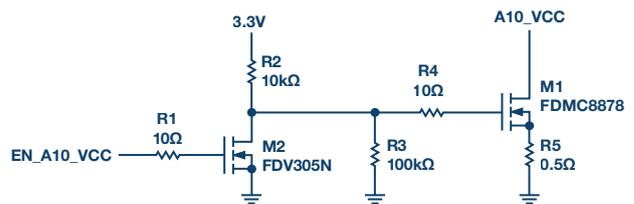


図15. FETを使って構成した放電回路

PSMによる課題への対処

ここまで説明したように、FPGA向けの電源に求められるすべての要件を満たすための最良のソリューションは、アナログ・デバイゼズのPSM ICを採用することです。PSM ICの製品群には、以下のような特徴があります。

- ▶ 電圧について、最高レベルの精度 ($\pm 0.5\%$ 未満) が得られます。
- ▶ EEPROMを利用することにより、完全な自律動作を実現できます。
- ▶ 完全にプログラマブルなシーケンス制御機能を搭載しています。そのため、システム全体にわたって、独立したタイミングで起動/停止を実行できます。
- ▶ 堅牢かつ統合された機能群により、システムのレベルで障害を管理することが可能です。
- ▶ 電圧、電流、温度、ステータスを網羅する包括的なテレメトリ機能を備えています。
- ▶ 協調動作が可能なICファミリー製品により、電源システムのあらゆる問題に対応可能です。

Intelの「Arria 10 SoC開発キット」には、「Arria 10 SoC」を対象とするアナログ・デバイゼズのPSMソリューションが採用されています (図16)。

同開発キットに実装されているのは、図17のような回路です。この回路は、コア電圧として0.95V/30Aを供給します。このように電力の要件が比較的緩い場合には、1個のLTM4677によって必要な電流 (最大36A) を簡単に供給することができます (図18)。より多くの電流が必要な要件の厳しいアプリケーションに対しては、LTM4677を最大4個並列に接続することで、最大144Aを供給することが可能です (図19)。



図16. Arria 10 SoC開発キット

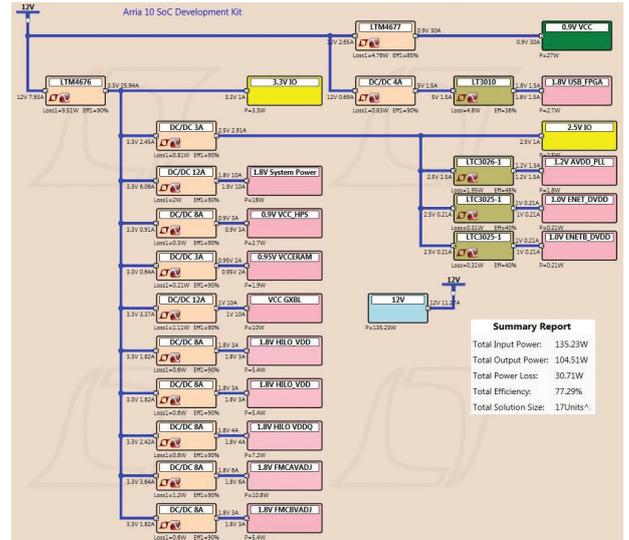


図17. Arria 10 SoC開発キットにおける電力の分配方法

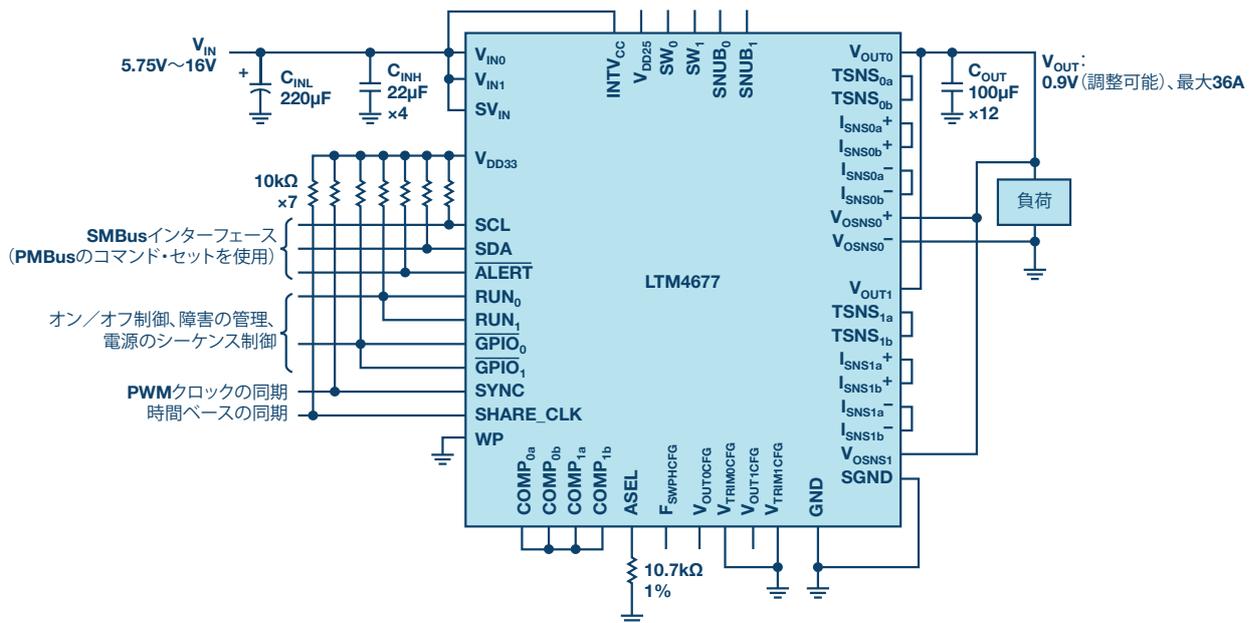


図18. LTM4677の使用例。
この1個のICで最大36Aの電流を供給できます。

LTM4677をベースとするソリューションを採用すれば、基板上のスペースを最も効率的に活用できます。μModule製品は集積度が高く、外部部品をほとんど必要としないからです。また、ハードウェアに変更を加えることなく、PMBusインターフェースを介して構成することも可能です。μModuleは、パワー・スイッチ、インダクタ、電流/電圧検出素子なども集積し、ループの安定性、熱など、アナログ回路における複雑な多くの問題に対処します。最も複雑さを緩和したソリューションだと言えるでしょう。

LTM4677はPSM機能を搭載しています。そのため、コア電圧は必ず目標とするDC電圧の±0.5%の範囲内に収まることが保証されます。また、FPGA内部のSmartVIDのIPと「LTpowerPlay®」のGUI（グラフィカル・ユーザ・インターフェース）のどちらによっても、PMBusインターフェースを介して電圧を調整可能です。これらの手段により、電源を完全に制御できるようになっています。

PSM機能を搭載していないレギュレータの管理には、LTC2977を使用するとよいでしょう。同製品は8チャンネルのPMBus対応PSM ICです。個々の電源に各チャンネルを接続することで、それぞれの電圧をプログラムされた目標値の0.25%の範囲内にサーボ制御することができます（図20）。LTM4677とシームレスに連携させてシーケンス制御と障害対応を行い、パワー・システム全体の一貫性を保つことも可能です。しかも、そうした機能を簡単にプログラムすることができます。

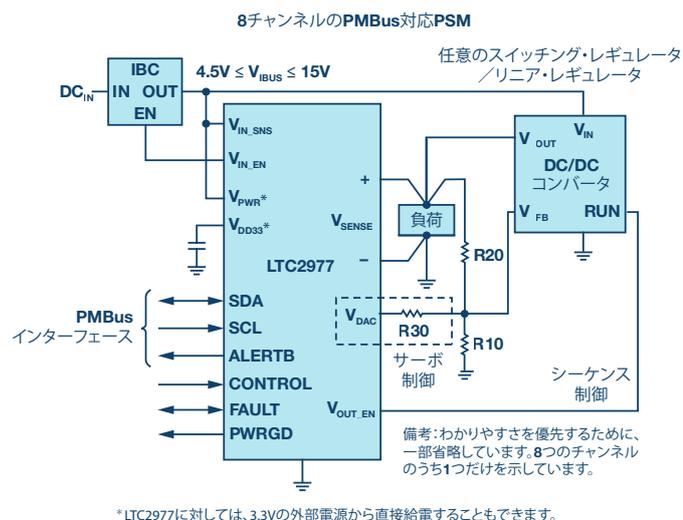


図20. LTC2977による任意の電源電圧の管理

システムの電源に対するシーケンス制御は、コア電圧を供給するLTM4677、3.3Vの電源電圧を供給するLTM4676A、ボード上のその他すべてのレギュレータを管理するLTC2977の協調的な連携動作によって行うことができます。これらのICは、起動/停止シーケンスを任意の順序、任意のタイミングで簡単に構成できるように、PMBusの同じタイミング・コマンドに対応しています（EEPROMに格納）。そのため、図6に示したグループ1、グループ2、グループ3の電源に対して、定義済みのイベントが自律的に正しいシーケンスで与えられることが保証されます。

高精度の電圧制御とシーケンスの制御に加えて、このボード上に実装されたLTM4677、LTM4676A、LTC2977を使うことで、完全な障害対応も行えます。1つ以上の電源レールに過電圧、低電圧、ブラウンアウト、過電流、または完全な故障が生じた場合、迅速かつ自動的に問題に対処するようシステムを構成できます。シャットダウンによって繊細なFPGAを保護し、可能であれば再起動するように構成することが可能です。

システム内のほとんどの電源レールに必要な電流（13A以下）と電圧の許容誤差は、それほど大きくありません。「LTM4620」のようなPSM機能を搭載しないデバイスで電力を供給し、LTC2977によってシーケンス制御と管理を行うことができます。その場合、ボードの面積、複雑さ、コストの間で非常に効果的にバランスをとることが可能です。

一方、フェーズ・ロック・ループ（PLL）やトランシーバーなどの電源レールについては、スイッチング・レギュレータよりもノイズが小さいリニア・レギュレータが適用されることが多いでしょう。このような用途に適しているのが「LTC3025-1」、「LTC3026-1」です。これらの製品は、スイッチングと負荷に起因するノイズを出力から除去します。これらもLTC2977で管理し、シーケンス制御、トリム、障害対応を行うことができます。

LTpowerPlay

包括的なGUIソフトウェアであるLTpowerPlayは、PSM ICの全ファミリ製品をサポートしています（図21）。ほとんどのPSM機能には、各PSM ICの内蔵EEPROMが備える構成用レジスタを介してアクセスします。そのアクセスを容易化するために、バス上のすべてのPSM ICをLTpowerPlayという単一のツール上に統合し、使いやすい単一のビューを提供するようになっています。LTpowerPlayは、設計/開発のすべてのフェーズで必要になる作業を支援する豊富な機能セットを備えています。同ツールでは、オフラインの状態でも、プログラムを実行する前のICの状態をビューとして表示することができます。また、同ツールを使えば、多数のPSM ICによって数百もの電源レールが制御されるシステム全体と、I²Cバスを介してリアルタイムに通信することも可能です。LTpowerPlayは、レジスタへのアクセスと機能に関する詳細な情報を提供することによって、複雑な構成の作業の簡素化と効率化を図ります。システムの稼働中に、システム内のすべての構成、ステータス、テレメトリに関する情報を、明確でわかりやすい形でグラフィカルに表示することができます。加えて、レジスタ全体のプログラミングとメンテナンスの作業を簡素化するために、Microsoft® Windows®が稼働するPC上で構成情報を作成して保存するための簡単な手段を提供します。電源に障害が発生した場合には、システム内の障害発生箇所、ステータス、テレメトリで取得した情報、障害の内容を示すログ情報をLTpowerPlayで簡単に確認することが可能です。一般的な障害が発生している場合に向けた便利なデバッグ機能も提供しています。必要に応じ、LTpowerPlay上でアナログ・デバイセズにサポートを要請することも可能です。その場合、ライブ・サポートを担当する技術者は、ユーザーが見ているのと同じGUI上の表示をリアルタイムに確認しながらサポートを提供することができます。

LTpowerPlayは[こちら](#)から無償ダウンロードできます。

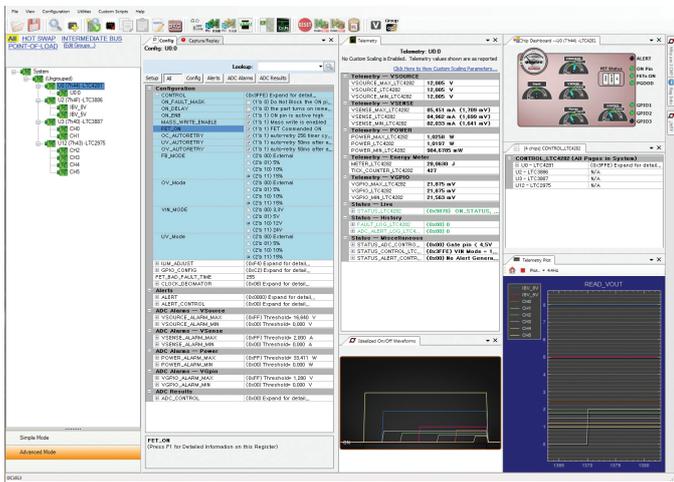


図21. LTpowerPlayのGUI

アナログ・デバイゼスは、Intel、Xilinx、NXP Semiconductors®のFPGAに対する包括的なデモ用プラットフォームを提供しています。完全な機能を提供するそれらのボードは、PSM ICを使用することにより、FPGAを搭載するシステムに対して最もクリーンで柔軟性と堅牢性の高い電源ソリューションを実現できることを示しています。また、アナログ・デバイゼスに問い合わせれば、PSM ICの製品群の中から適切なものを選択して使用することができます。アプリケーション・エンジニアからの支援を受けることができます。詳細な情報、参考資料のダウンロード、FPGA向けのボードの発注については、[こちら](#)をご覧ください。

最新FPGAが実現されるまでの歩み

ここまでの説明により、FPGAを使用するシステムに適切に電力を供給する方法については理解が得られたはずですが、ここからは、個々のケースに固有の問題は脇に置いておき、本質的な話として今日のFPGAを巡る状況を確認します。そのためには、まずこれまでの歴史を簡単に振り返る必要があります。

ムーアの法則

1965年に、Gordon Moore氏はElectronics Magazine誌に非常に有名な論文を寄稿しました⁸。同氏は、1つのチップ上に集積されるトランジスタの数が毎年2倍に増加していることを指摘し、少なくとも1975年まではその傾向が続くと予想しました。後に、エレクトロニクス市場におけるより大きな進化と観測に基づき、同氏は自身が考案したモデルを改訂しました。その後、チップ上に集積されるトランジスタの数は指数関数的に増加するという概念は、エレクトロニクス業界における基本的な法則として受け入れられました。これは、他の時代や他の業界には存在しない自己成就予言でした。実際、ムーアの法則は世界中の技術者にとっての基本的な動機付けとなりました。それによりイノベーションが創出され、Moore氏がそのシンプルな観測結果を最初に公開したときには想像もしていなかったようなトレードオフが行われるようになっていきます。

このように、技術者が自らにハードルを課して技術的な進歩を達成しようとした結果、意思決定のプロセスでは、コスト、消費電力、使いやすさ、耐久性を犠牲にしても、より小さなチップ面積により多くの素子を集積できるようにする技術が優先されるようになりました。技術的な進歩においては、何にも増してチップのサイズ（集積度）が重視されます。この傾向は、高度なチップはより多くの電力を消費し、リーク電流が多くなり、より脆

弱で繊細になり、管理と保護が格段に難しくなるということを示唆しています。

トランジスタの形成技術の変化

トランジスタの微細化がナノメートルのスケールで進行する中、それに伴う副作用はますます深刻なものになりつつあります。最も顕著なのは、電圧のヘッドルームです。トランジスタの電源電圧と言え、数十年前であれば5Vというのが適切な値でした。しかし、最近のFETに5Vの電源電圧を印加すると、すべての接合部と酸化物が破損してしまいます。微細化に伴って、トランジスタ内部の電界ははるかに強くなるため、破損を防ぐためには動作電圧は低く抑える必要があります。実際、新世代のトランジスタは、最高で約1.0Vの動作電圧にしか対応できません。また、電源電圧の許容誤差（絶対値）もそれに比例して小さくなります。1.0Vの2%は5Vの2%よりもはるかに小さく、どのようにして電源電圧の精度を高めるかということが、従来よりも差し迫った問題になっています。

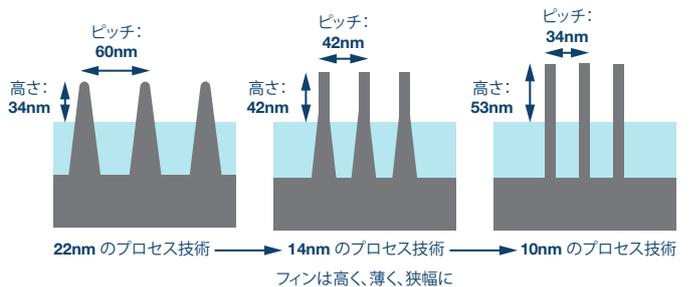


図22. トランジスタの微細化

動作電圧が低下する一方で、トランジスタの駆動能力（飽和ドレイン電流） I_{DSAT} はより高まっています。駆動能力を高めることには、少なくとも2つの効果があります。まず、ゲート電圧が低いトランジスタにおいて、高い周波数でのスイッチングに対応できるだけの多くの駆動電流を流せるようになります。そして、もう1つの効果は、トランジスタの物理的なサイズを小さくできることです。トランジスタはサイズが小さいほど高速です。但し、トランジスタの駆動能力を高めると、リーク電流が増加します。

チップ上のトランジスタの消費電力は2種類に分けられます。1つは動的消費電力です。これは、ある周波数で論理レベルの1と0の間でスイッチングする際に消費される電力のことです。トランジスタそのものの容量と、各トランジスタを互いに接続するチップ上の配線に付随する小さな寄生容量に対する充放電によって発生します。動的消費電力は、論理レベルが遷移する周波数と、電源電圧の2乗に比例します。

もう1つは、トランジスタのリーク電流によって消費される静的消費電力です。回路に電源が投入されている間は、アクティブな状態なのか、それともアイドル状態なのか、クロックが供給されているのか停止しているのかにかかわらず、必ずこのリーク電力が生じます。トランジスタの駆動能力を高めるためには、接合部などの構造は、より多くの電流を流せるように形成されます。逆に言うと、電流が流れないようにすることが難しくなり、リーク電流が増加するということです。駆動能力の高いトランジスタほど、リーク電流は多く流れます。そして、微細化が進むごとに、リーク電流の影響は増大します。トランジスタの形成技術（化学、金属工学、リソグラフィ、物理学を活用）と、精度、柔軟性に優れた電源管理を組み合わせることでしか、リーク電力を制御することはできません。

Moore氏は10年前にこの事実に気づき、2つの重要な事柄を指摘しました。1つは、動的消費電力が従来どおりのペースで増加し続けると、動作中のチップの接合部温度は太陽の表面温度に近いレベルまで上昇してしまうということです。もう1つは、何も対策を講じなければ、リーク電力は動的消費電力を上回る電力消費の最大の要因になり、消費電力の問題はより悪化するということです(図23)。これらの問題に対処するために、当時のIC業界は、複数の新しい手法を導入しました。その1つが、クロックを低速化するか停止することによって動的消費電力を抑えるクロック管理技術です。また、増加するトランジスタを有効に活用するために、1つのチップ上に複数のプロセッシング・コアを集積するという手法も考案されました。

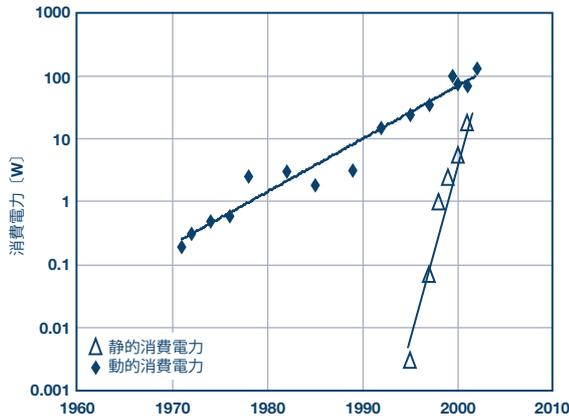


図23. 静的消費電力と動的消費電力の増加⁹⁾

ただ、そうした高度なアーキテクチャを採用しても、リーク電力はやはり面倒な問題のままです。トランジスタの形成技術は、リーク電力の増加を抑えるための手段として有効であることは確かですが、それだけでは十分ではありません。トランジスタの微細化が進むに連れて、電源電圧が引き下げられることから、動的消費電力の問題は対処できるレベルに維持されます。その一方で、トランジスタの駆動能力とリーク電力は増加を続け、チップ上の素子数も増え続けます。そこで重要になるのが電源電圧の管理です。電源電圧をきめ細かく制御すると共に、個々のデバイスのニーズに応じてアクティブな調整を可能にすることが重要になります。

高度なアーキテクチャ

2000年までのアーキテクチャは、次のような目的を達成すべく開発されていました。すなわち、1つのコアによって、できるだけ多くの演算を、できるだけ高速に実行できるように最適化するという事です。それ以上クロック・レートを上げると回路が破損してしまうというぎりぎりのレベルまで、最大動作周波数が高められていました。アーキテクチャの最適化も行われましたが、その目的は、個々のクロック・サイクルで、できるだけ高い性能を絞り出すことでした。

その後、消費電力が急激に重要視されるようになりました。結果として、単に処理速度を引き上げるのではなく、より細かいレベルで最適化を図ることに目が向けられるようになりました。従来のコンピューティング・アーキテクチャでは、世代ごとにクロック周波数がとどまることなく高められていました。現在ではその傾向に歯止めがかかり、トランジスタ当たりの性能の伸びも小さ

なっています(図24)。 V_{DD} から V_{SS} に流れる電流を素早く停止することが、動的消費電力の抑制に大きな効果を発揮しました。

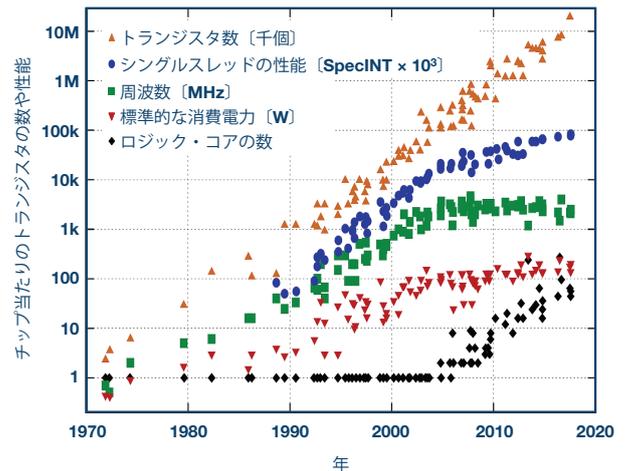


図24. コンピューティング・ハードウェアの各種指標の変化

一方で、Moore氏の予測に従い、1つのチップに集積されるトランジスタの数は、とどまることなく増加し続けました。それに対しても、何らかの対策が必要になりました。そこで、次に具現化されたイノベーションがマルチコア・アーキテクチャです。クロック周波数を高めようという動きが止まったのとほぼ同じ時期に、1つのチップ上に集積されるコアの数が増え始めました。マルチコアのメリットとしては、1つのコアを再利用することによってチップの設計が簡素化されること、使い慣れたビルディング・ブロックを対象とするのでソフトウェアの設計が簡素化されること、演算の負荷に応じ、各コアに対して個別の調整を適用できることなどが挙げられます。当初、マルチコアの革命は、固定のコンピューティング・プラットフォームによって実現されていました。FPGAが真価を発揮するようになったのは、コアの数を最大化することが最良の選択肢だと世界が気づいた、まさにそのときだったと言えるでしょう。同一のプログラマブル・ロジック・ブロックを無数に集積したFPGAよりも、多くのコアを搭載できるものなど存在しなかったからです。

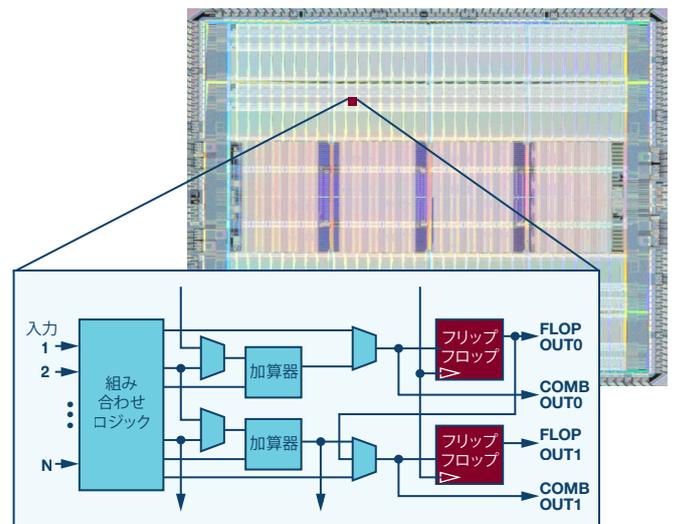


図25. FPGAが備える構成可能なファブリック

FPGAの構造

FPGAは、最も基本的なレベルでは、構成が可能で互いにメッシュ接続されたプリミティブなロジック・セルの集合体だと説明することができます。コンパイラと共に使用することで、組み合わせロジック・ブロックやシーケンシャル・ロジック・ブロックなど、想像し得るほぼすべての汎用デジタル機能に対応可能な非常に柔軟性の高い演算ファブリックが構成されています。トップ・レベルでは、これらのファブリックの周辺に、実現可能な機能を補助/増補するための回路が追加されています。例えば、バイパス回路、RAM、PLLなどの追加のブロックにより、チップ内部で実現できる機能を増やせるよう補助するということです。多種多様かつ構成が可能なGPIOセル、高速通信用のハード・マクロ（LVDS、DDR、HDMI、SMBusなど）、高速トランシーバーなどにより、チップ内部のロジックは、様々な電圧、速度、プロトコルで外部の世界と通信できるようになっています。CPUやDSPといった集積度の高いコア・ブロックは、共通的に必要とされる機能をサポートします。それらの機能は、消費電力、速度、サイズといった面で最適化されています。

FPGAのコア・ファブリックは、CLB（Configurable Logic Block）と呼ばれる無数のプリミティブなセルで構成されます。各CLBは、組み合わせロジックやシーケンシャル・ロジックの構成要素の集合です。それらが全体として基本的な演算を実行し、1個以上のフリップフロップに値を保持することができます。通常、組み合わせロジックは、数個の入力ビット・データから数個の任意の出力ビット・データへの変換を実現するプログラマブルなルックアップ・テーブル（LUT）によって実装されます。各LUTは、プログラムされたとおりに1つの基本的なロジック機能を実行します。その上で、後続の処理に向けて、得られた結果を構成可能なインターコネクタに引き渡します（図26）。CLBとLUTの具体的な設計方法は、FPGAファミリごとに異なります。その点が、ファミリごとの違いを生む1つの要因にもなっています。安価なFPGAには、入力、出力、インターコネクタ、フリップフロップの数が少ない単純なCLBが使用されます。一方、ハイエンドのFPGAは、それよりもはるかに複雑なCLBを採用しています。各CLBは、より多くの入力とロジックの組み合わせに対応し、より高速に動作します。このような最適化によって、CLB当たりの演算能力が増し、コンパイル後の回路の性能が更に最適化されます。当然のことながら、複雑なFPGAは入出力の数が多くなります。そのため、動的消費電力に関するトレードオフは、よりシンプルでインターコネクタの少ないデバイスとは異なる性質のものになります。

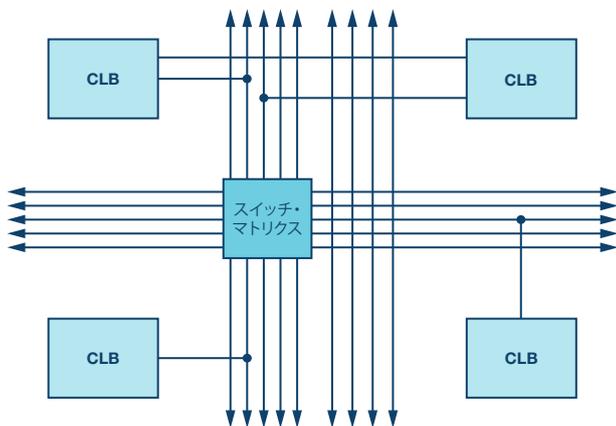


図26. CLB間に配置される構成が可能なスイッチ・マトリクス

構成が可能なロジック機能という基本的な概念は、コア・ファブリックだけでなく、その外部のI/Oセルにも適用されます。I/Oセルも柔軟に構成することが可能であり、幅広い電圧、駆動能力、ロジック方式（プッシュプル、トライステート、オープンドレインなど）に対応できます。構成が可能なLUTとインターコネクタのマトリクスと同様に、プログラマブルなI/Oは、起動時に構成用のメモリから構成情報を受け取ります。このことから、電源の起動シーケンスが重要であることがわかります。

汎用のCLBやGPIOでは実装できない、または実装してはならない機能ブロックもあります。いわゆるハード・マクロとして実現されているものがこれに該当します。つまり、最適化を図ることでより良いものになる機能や、十分な高速化や小型化が不可能なので、専用回路として用意した方がよい機能のことです。具体的な例としては、ギガビットのレベルの伝送レートに対応するトランシーバー、算術用ロジックやDSPエレメント、特殊なコントローラ、メモリ、専用のプロセッサ・コアなどがあります。この種の機能は、ソフトウェアのようにコンパイルし、構成が可能なファブリックに読み込むソフト・ブロックとしてではなく、ハード・マクロとして実現されます。通常、ハード・マクロには専用の電源が用意され、特定の電圧とタイミング要件で動作します。

このように、FPGA製品には多様な機能ブロックが集積されます。電源に関してはそれぞれに異なる要件があり、それに対応できる電源システムを用意する必要があります。通常、コア・ファブリックには、チップ上で最も低い電圧と最も大きな電力を供給することになります。最新のFPGAの場合、十分に活用されているファブリックに対しては、0.85Vの電源によって100Aを超える電流を供給しなければならない可能性があります。CPUコアでも同程度の電圧を使用しますが、電流量には差がありますし、シーケンスに関する要件も異なります。チップ上のアナログ機能は、3.3Vや1.8Vの電源電圧を使用します。これらの電圧は、他のブロックよりも先に供給する必要があります。GPIO群も3.3Vまたは1.8Vで動作しますが、コア・ファブリックのパワーオン・リセットが完了するまでは電力を供給してはなりません。システムにおいては、電源のシーケンスに関するそうした各種の要件を確実に満たす必要があります。

FPGAのアーキテクチャにおける最後の要素はツール・チェーンです（図27）。初期状態の構成可能なロジック・ファブリックを高性能の回路に変換するためには、まずVerilogまたはVHDLで記述したコードをロジック・ブロックに変換します。続いて、クロック信号やリセット信号、テストビリティを確保するためのリソースを割り当てます。更に、速度、消費電力、サイズの制約に応じて機能を最適化し、その結果をFPGAの構成用EEPROMに書き込みます。このような一連の作業を行うためには、包括的なツール・セットを利用することになります。そうしたツールがなければ、FPGAの潜在的な能力を最大限に活用することはできません。実際、ツールとプログラミング言語はとても重要な要素です。FPGAを機能させる上では、基本的な回路設計技術よりも、それらが重視される傾向にあります。技術者としては、適切な電源を用意することに時間とエネルギーを費やすのではなく、プログラミングに工数の大半を振り分けたいと考えます。

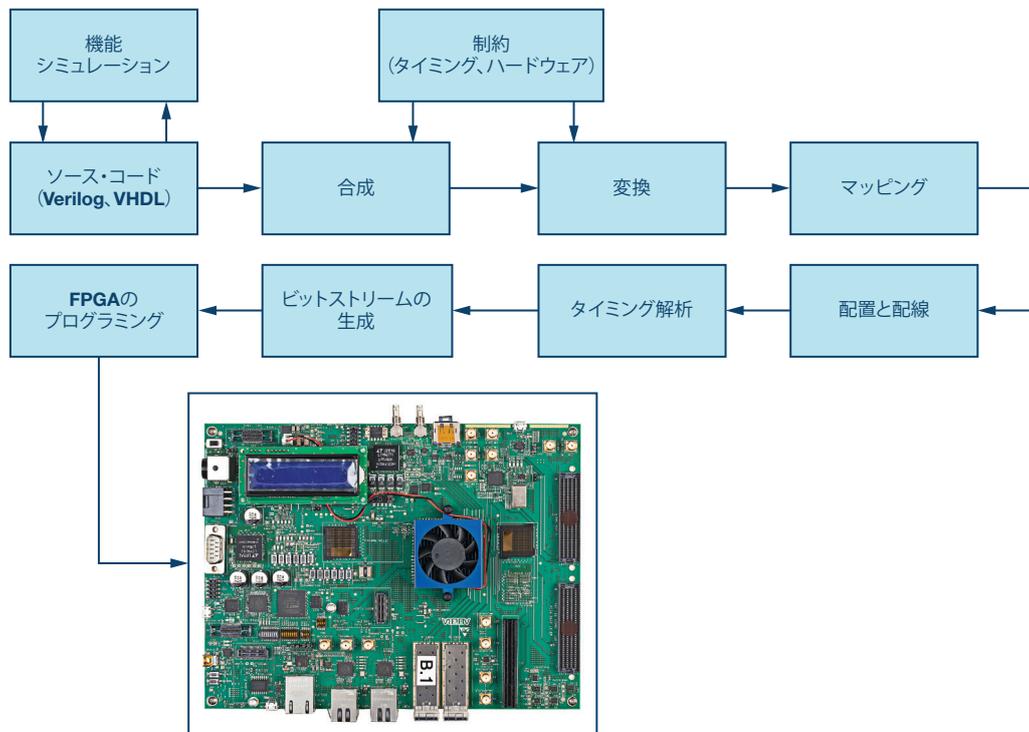


図27. FPGAの設計フロー。最終的にハードウェアが得られます。

つまり、それらのツールを使用した作業がメインになるわけですが、そこには1つの問題があります。それは、そうしたツールを使っていると電源に対する要件が示唆されるはずなのですが、設計者はそれを見落としがちだということです。あまりにも多くの労力がデジタル設計に注がれるため、回路のコンパイル後になって、電源に対する要件が明らかになるということが実際にあるのです。その段階で電源に問題があることが発覚しても手遅れです。ソフトウェア・ツールを活用してデジタル設計を行う場合でも、一般的なハードウェア設計を実施するケースと同様に、電源のアーキテクチャとしては柔軟性に優れたものを選択することが成功に向けた鍵になります。

FPGAを構成するトランジスタとアーキテクチャのトレンドを創出したのは、歴史、経済、人間です。どのレベルにおいても、どの設計段階においても、電源はFPGAをうまく活用する上で重要な役割を担います。精度、堅牢性、柔軟性が高く、コンパクトで使いやすいというのが、最良の電源が満たすべき条件です。アナログ・デバイス製のPSM IC製品は、それらすべての面で業界における基準となる性能を達成しています。

謝辞

図24を作製するためのデータ収集とグラフ化に協力していただいたM. Horowitz氏、F. Labonte氏、O. Shacham氏、K. Olukotun氏、I. Hammond氏、C. Batten氏、K. Rupp氏¹¹に感謝します。

付録A：「世話と餌やり (Care and Feeding)」

とは？

特に英語を母国語としない方にとっては、本稿のタイトルは違和感のあるものであるかもしれません。FPGAに対して「世話と餌やり (Care and Feeding)」というフレーズを使うのは、全くふさわしくないように感じられるでしょう。しかし、そのような意見に対する答えはいたって単純です。英語は面白い言語なのです。「Care and Feeding」というフレーズが広く使われるようになったのは、いつご

ろからなのでしょうか。正確な時期については様々な見解があると思います。ただ、現在よりもはるかにシンプルだった時代の農業に由来し、脆弱であったり予測不能であったりするあらゆるものに対して広く使われていた（乱用されていた）ことはよく知られています。本稿にとって、まさに的を射たフレーズだと思っています。FPGAに対して「feed」が必要であるかどうかは意見が分かれるところですが、間違いなく「care」は必要です。

「Care and Feeding」というフレーズは、インターネットの時代である現在、幼児、子供、夫、上司、国外居住者、科学的データ、更にはデジタル・パルス整形フィルタなど、様々なものに対して広く使われています。ここでは、最も古く、おそらく最も奇妙な参考文献の1つとして、図28に示した古典的な書物を紹介しておきます。タイトルだけでなく、インターネットを利用すればすぐに入手できるという点にも面白みを感じます。



図28. Geo. R. Simpson氏が執筆した「Care and Feeding of Indian Runner Ducks」(インディアン・ランナー種のアヒルの世話と餌やり)

参考文献

- ¹ Intel Arria 10 device data sheet (Intel Arria 10のデータシート) Intel、2018年6月
- ² 「AN 711: Power Reduction Features in Intel Arria 10 Devices (Intel Arria 10の消費電力を削減する機能)」2018年7月
- ³ Intel Arria 10 Core Fabric and General Purpose I/Os Handbook (Intel Arria 10のコア・ファブリックおよび汎用I/Oのハンドブック) Intel、2018年8月
- ⁴ 「Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics (Virtex UltraScale FPGAのデータシート: DC性能とACスイッチング性能)」Xilinx、2018年1月
- ⁵ 「AN 692: Power Sequencing Considerations for Arria 10 and Stratix 10 Devices (Arria 10とStratix 10の電源シーケンス制御に関する検討事項)」Intel、2018年4月
- ⁶ DMN1032UCB4 data sheet (DMN1032UCB4のデータシート) Diodes Incorporated、2015年1月
- ⁷ Joachim Von der Ohe 「Application Note: Pulse Load on SMD Resistors: At the Limit (アプリケーション・ノート: SMD抵抗のパルス負荷、その限界)」 Vishay Beyschlag、2015年8月
- ⁸ Gordon Moore 「Cramming More Components onto Integrated Circuits (ICに、より多くの部品を集積する)」 Electronics Magazine Vol. 38、No. 8、1965年4月19日
- ⁹ Gordon Moore 「No Exponential is Forever: But ‘Forever’ Can Be Delayed! (急成長は永遠には続かない。しかし、その終わりは先送りできる!)」 International Solid-State Circuits Conference、2003年
- ¹⁰ Samuel Fullers、Lynette Millett 「The Future of Computing Performance: Game Over or Next Level? (演算性能の未来: ゲーム・オーバーか、それとも次なるレベルが到来するのか?)」 National Academy of Science (ワシントン D.C.)
- ¹¹ Karl Rupp 「40 Years of Microprocessor Trend Data (マイクロプロセッサに関する過去40年間のトレンド・データ)」2018年2月

著者:

Nathan Enger (nathan.enger@analog.com) は、アナログ・デバイセズのミックスドシグナル・アプリケーション・エンジニアです。コロラド州コロラド・スプリングスの事業所に勤務しています。コロラド大学で電気工学の学士号、コロラド技術大学で電気工学の修士号を取得しています。Ford Motor、Intel、Marvell Technology Groupで、23年間にわたって高速PLLやトランシーバーなどのミックスド・シグナルICの設計/開発に従事してきました。



Nathan Enger