XILINX®

XAPP377 (v1.0) 2002 年 4 月 12 日

CoolRunner-II CPLD による低電力デザイン

概要

CoolRunner™ -II CPLD は、高速と低電力の両方を兼ね備えた唯一の次世代 CPLD です。このアプリケーション ノートでは、CoolRunner-II CPLD 独自の電力節約機能を使用して電力消費を最小限にするデザイン手法について説明します。

はじめに

CoolRunner-II CPLD は、従来の CoolRunner CPLD と同じように Fast Zero Power™ (FZP) 技術を利用していますが、Xilinx 独自のデザイン手法、より小さいジオメトリ、最先端のプロセス技術によって、低電力規格としてさらに進歩しています。他の CPLD メーカーは FZP 技術を利用した CoolRunner CPLD の CMOS 技術を再現しようとしていますが、CoolRunner のベンチマークと同等の結果を出すことはできていません。CoolRunner-II デバイスは、CPLD 業界で初めて高速と低電力を同時に実現するとともに、パワー ダウン モードを使用せずに業界最小のスタンバイ電流を実現しています。さらに、CoolCLOCK、DataGATE、DualEDGE フリップフロップなどのユニークな電力節約機能もあります。

従来の CPLD では、センス アンプ タイプの積項を使用して伝搬遅延時間を短くしていました。センス アンプ積項は、積項の論理状態の変化を示す、ワード ラインにおける電圧レベルの小さな変化を検出するようにバイアスがかけられます。バイアスをかけるトランジスタには、スタンバイ状態でも常に電流が流れます。したがって、このようなタイプの CPLD は、CoolRunner-II CPLD のような低電力ソリューションを提供できません。センス アンプ タイプのデバイスでは、サイズが大きくなってマクロセル数が増えると、電力を消費する積項が増えるので、電力消費が著しく増加します。

プロセス技術が微細になると、センス アンプ タイプの CPLD はその性質上、パフォーマンスを維持するためにより多くの電力を消費します。微細なプロセス技術では電源電圧を低くする必要があるので、センス アンプのゲインが小さくなります。さらに、ジオメトリが小さくなると、積項トランジスタからの漏れ電流が増加します。センス アンプ CPLD でパフォーマンスを維持するには、ワード ラインの小さい電圧変化を検出するようにゲインを増強し、バイアスが漏れ電流を補償するようにデザインする必要があります。バイアスを高くすると、バイアス ネットワークに流れる電流が増加するので、全体の電力消費が増加します。センス アンプ ベースの技術を使用する他のメーカーは、微細化し続けているプロセス技術の影響を補償するために、現在の製品を再デザインするというプロセスを避けることができません。

FZP 技術で使用されている CMOS の積項は、その性質上、状態が変化しないときは電力消費が少なくなります。CMOS ロジックではスタンバイ電流が小さいので、CoolRunner-II CPLD はこの技術を使用して低電力を可能にしています。さらに、CMOS には、ジオメトリが小さくなると消費電力が低下し速度が向上するという利点もあります。

電力節約機能

CoolRunner-II シリーズには、FZP 技術の電力節約機能を拡張する新しいアーキテクチャ上の機能が追加されています。このセクションでは、これらの機能について説明し、どのようにして電力を節約するかについて説明します。

以下で説明する新機能は、すべて XC2C128 以上のデバイスで使用できます。これより小さいデバイスの XC2C32 と XC2C64 では、DataGATE、クロック分周器、CoolCLOCK の機能は使用できません。

DataGATE

多くの場合、デバイスは、マスタ デバイスがアドレスしないデータ バスに接続されます。CPLD がこのような状況になると、CPLD が使用しないバスのデータが変化した場合に CPLD の内部ロジックも変化するので、必要以上に電力が消費されます。CPLD 内でロジックの状態が変化すると、電力が消費されます。したがって、アドレスされていないときに CPLD をデータ バスから切り離すと、電力が節約されます。

DataGATE は、外部信号の変化を内部ロジックから切り離してデバイスの電力消費を減らすことによって、この問題を解決します。これは、入力として設定した I/O ピンで、DataGATE グローバル ネットが制御する CoolRunner-II 独自のパス ゲートを使用することによって実現されます。DataGATE グローバル ネットは、特定のピン/マクロセルが起点となり、外部信号またはロジック エレメントを使用して内部で展開した信号のどちらかによって駆動できます。入力ピンのパス ゲートが DataGATE 制御ネットによって無効になると、内部ラッチが CPLD のロジック ネットワークを駆動して、DataGATE 制御ネットをアサートする直前に入力ピン上にあったものと同じロジック レベルを維持します。この結果、外部データの状態が変化しても CPLD の内部ロジックは状態が変化しなくなります。

たとえば、多くの場合、他のデバイスとデータ バスを共有する CoolRunner-II CPLD は、自分自身のアドレスを持っており、常にアドレスされるわけではありません。アドレスされていない CoolRunner-II CPLD をデータ バスから切り離すには、2 種類の方法があります。1 番目の方法は、チップ セレクト信号を使用してデバイスをアドレスする場合に、チップ セレクト信号を DataGATE 制御ピンに割り当てて、データ バスから入力を分離するために使用します。2 番目の方法は、アドレス バスを使用する場合に、デバイスのアドレスを CPLD の内部でデコードして、アドレスされたときにデータを受信できるように DataGATE によってデータ バスの入力を有効にします。CPLD がアドレスされていない場合、DataGATE は CPLD 内部でデコードしたアドレスを使用して外部データ バスの信号を切り離します。この場合、内部でデコードした結果は、バスの変化を切り離すために DataGATE ピンに接続されます。

DataGATE は、JTAG ピンと DataGATE ピン自体を除く I/O ピンの一部またはすべてに影響を与えるように設定できます。上記の例は、一部のマクロセル I/O ピンに影響を与えるように DataGATE を設定したものです。アプリケーションによっては、CPLD からのシステム クロックまたはクロックの切り離しが有効な場合があります。最も電力を消費する 2 つのエレメントは、出力バッファとクロック ツリーです。ある期間 CPLD が必要ない場合、DataGATE 機能を使用してクロックを切り離すことができます。このようにすると、電力消費が著しく削減されます。ただし、意図しないロジック遷移が発生する可能性があるので、クロックを切り離す場合は注意が必要です。

他の場合にも、CoolRunner-II 独自の DataGATE 機能が電力の削減に役立ちます。また、DataGATE は柔軟なので、アプリケーションによっては、デバイス全体またはデバイスの一部のみを外部信号から分離することができます。たとえば、CoolRunner-II CPLD でボードのトラブルシューティング手順を向上させることができます。

シュミット トリガ

CoolRunner-II のシュミット トリガ入力は、入力にヒステリシスを必要とするアプリケーションで役立ちます。このようなアプリケーションには、特定のピンでノイズが問題になるアプリケーションなどがあります。ヒステリシスは、ノイズ耐性を向上させます。この他に、CPLD の外部でオシレータ回路をインプリメントし、CPLD のロジックをオシレータ回路の一部のインプリメントに使用するアプリケーションでも有効です。

シュミット トリガ タイプのバッファは通常の入力バッファよりも電力を消費するので、CoolRunner-II CPLD で電力消費を最小にするには、シュミット トリガ入力を無効にします。シュミット トリガを無効にする場合は、CPLD の入力で意図しない遷移が発生しないようにするため、ノイズが少なくなるようにシステムのデザインと運用を行うことが重要です。

上記は入力信号のノイズが少なく立ち上がりエッジが高速である場合ですが、逆に入力信号の立ち上がりエッジが遅い場合は、シュミット トリガを有効にすると消費電力を抑えることができます。立ち上がりが遅い信号はロジックが中間の状態にある時間が長くなりますが、この遷移中は通常の入力バッファの方が電力を多く消費します。シュミット トリガにはヒステリシスが伴うため、この状態を回避することができます。

クロック分周器

CPLD では、グローバル クロック ネットワークが最も電力を消費する傾向があります。グローバル クロック ネットワークの周波数を小さくすると、システムの電力消費が著しく削減されます。このため、CoolRunner-II デバイスは、グローバル クロックの GCK2 にクロック分周ネットワークが含まれるようにデザインされています。このクロック分周器は、クロックを遅延させずに、2 ～ 16 の範囲にある偶数でシステム クロックを分周できます。図 1 を参照してください。

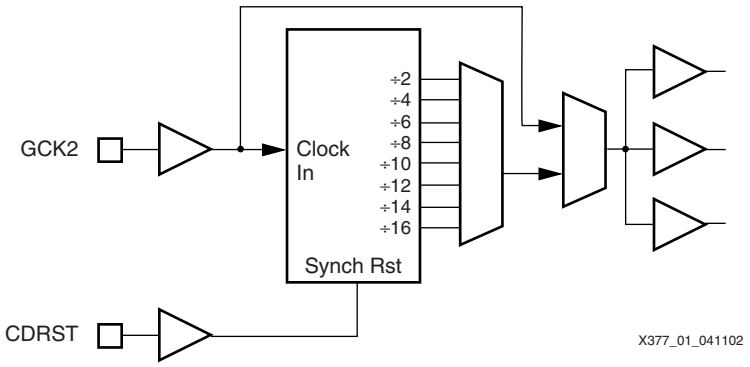

図 1: GCK2 のクロック分周回路

たとえば、システムによっては、最大速度の外部システム クロックを必要としないステート マシンを使用する場合があります。このような場合、クロック分周器はシステムの電力を削減するための最適なツールとなります。ユーザーがロジックを構築してクロック分周器を定義すると他のことに使用できるロジックを消費してしまいますが、クロック分周器を使用すれば、このようなことはありません。クロック分周器を使用してグローバル クロック ネットワークの周波数を低くすると、CoolRunner-II CPLD で消費される電力が少なくなります。

一般に、可能な限り遅いシステム クロックでデザインすると、電力消費が削減されます。このため、CoolRunner-II アーキテクチャで提供されるクロック分周器は、設計者にとって非常に役立ちます。

DualEDGE レジスタ

クロック信号の両エッジを利用できるようにすることによって、DualEDGE フリップフロップとして設定されたマクロセルは 2 倍の処理を行うことができます。図 2 に、DualEDGE フリップフロップとして設定したマクロセルを示します。DualEDGE フリップフロップを使用しないシステムでこのマクロセルと同じ出力を得るには、クロックの周波数を 2 倍にする必要があります。DualEDGE フリップフロップを使用するマクロセルはクロックの立ち上がりエッジと立ち下がりエッジの両方で動作するので、クロック ネットワークをより効率的に使用できます。その後、グローバル クロック ネットワークが低い周波数で動作するので、電力消費が削減されます。

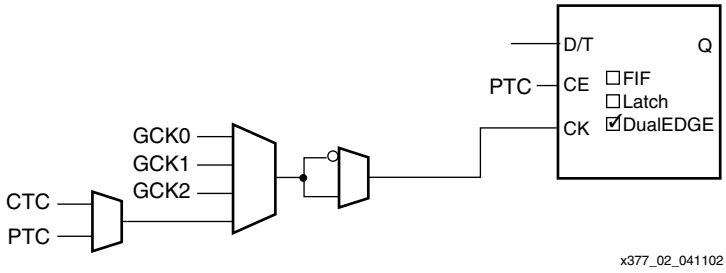

図 2: DualEDGE オプションを使用したマクロセルのクロック チェーン

DualEDGE フリップフロップはクロック分周器の機能を拡張できるので、電力がさらに節約されます。DualEDGE フリップフロップをクロック分周器とともに使用すると、グローバル クロックを 3、5、7 という奇数で分周できます。たとえば、クロックを 3 分周する必要があるシステムでは、6 分周するよ

うにクロック分周器を設定してから DualEDGE フリップフロップで 2 倍にすると、3 分周したことになります。クロックの周波数を低くすれば必ず電力が節約されるので、これは重要です。DualEDGE フリップフロップは、クロック分周ネットワークの機能を実質的に拡張させます。

デザインに 2 つのステート マシンが含まれているとします。一方のステート マシンはシステム クロックの 1/4 の周波数で動作し、もう一方のステート マシンは 1/8 の周波数で動作します。DualEDGE フリップフロップをクロック分周器とともに使用すると、このような状況に対応できます。システム クロックの 1/8 の周波数で動作するステート マシンは、8 分周するように設定したクロック分周器をそのまま使用できます。もう一方のステート マシンに割り当てられたマクロセルで DualEDGE フリップフロップを有効にして、このマクロセルに 8 分周したクロック分周器を割り当てると、クロックの周波数がシステム クロックの 1/4 になります。つまり、事実上 1 つのクロック分周器で 8 分周と 4 分周の 2 つの機能が得られます。これらのクロックは相互およびシステム クロックと同期しているので、2 つのステート マシンは同時に動作します。

以上をまとめると、CoolRunner-II でクロック分周器と DualEDGE フリップフロップを使用すれば、2、3、4、5、6、7、8、10、12、14、16 のいずれかの値で分周したクロックを得ることができます。さらに、マクロセルのグループでクロック分周器を使用し、そのマクロセルの一部で DualEDGE フリップフロップを使用すると、クロック分周ネットワークは 1-2 (後述する CoolCLOCK を使用)、2-4、3-6、4-8、5-10、6-12、7-14、8-16 のいずれかの値で分周した 2 種類のクロック周波数を提供できます。DualEDGE フリップフロップをクロック分周器とともに使用すると、電力を削減するだけでなく、デザインの動作効率を向上させることもできます。

CoolCLOCK

CoolRunner-II CPLD には、CoolCLOCK というユニークな機能もあります。この機能を使用すると、クロックの速度に影響を与えずにグローバル クロック ネットワークの電力消費をさらに削減できます。CoolCLOCK は、クロックを遅延させません。前述したように、グローバル クロック ネットワークの周波数を低くすると、電力消費が著しく削減されます。

CoolCLOCK は、グローバル クロック ネットワークに適用される前に外部クロック周波数を 2 分周することによって、グローバル クロック ネットワークが消費する電力を削減します。このクロック分周はクロック ツリーの初めの方のクロック入力バッファの近くで行われるので、分周されたクロックはクロック ネットワークの大部分に影響を与えます。グローバル クロック ネットワーク (GCK2) は内部容量が比較的大きいので、周波数を低くすると GCK2 が消費する電力が著しく削減されます。図 3 に示すように、分周されたクロック信号は、DualEDGE フリップフロップを使用するマクロセルで 2 倍になります。これによって、外部システムと同じクロック周波数がマクロセルに適用されます。DualEDGE フリップフロップを使用するように設定する必要があるのは、元のクロック周波数を必要とするマクロセルだけです。最大速度のクロックが不要な場合は、分周されたクロック周波数を使用するように他のマクロセルを設定できるので、電力消費がさらに削減されます。

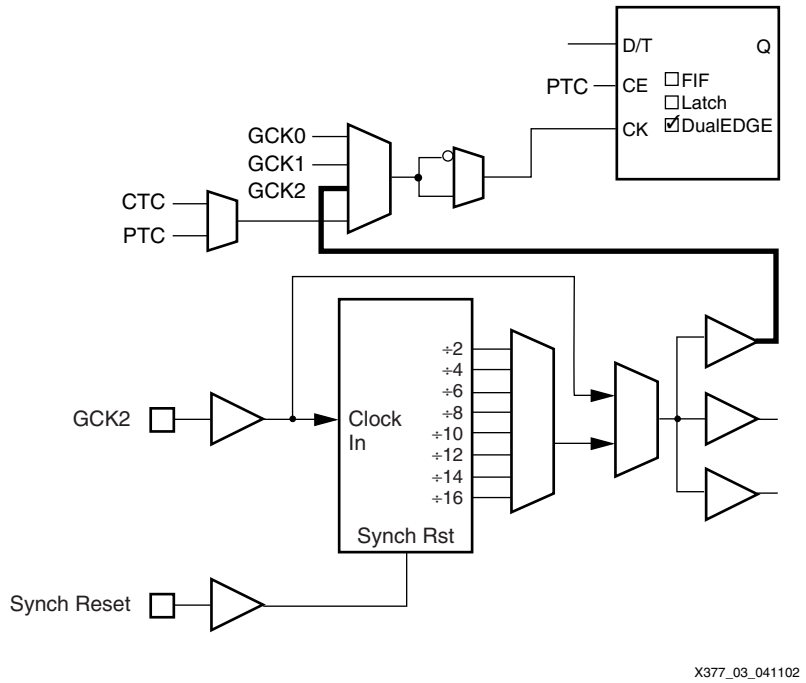

図 3: クロック分周器と DualEDGE フリップフロップをカスケード接続して作成した CoolCLOCK

ウィーク プルアップとバス ホールド

すべての CoolRunner-II CPLD には、外部でトライステート化されたバスによる I/O の電力消費を削減するための、I/O 終端処理オプションが含まれています。I/O 終端処理回路は、ウィーク プルアップ、バス ホールド、終端処理なしという 3 種類の方法で設定できます。ウィーク プルアップとバス ホールドはデバイス全体に適用され、両方を同時に使用することはできません。そして、選択した終端処理を使用するかどうか、ピンごとに指定します。ウィーク プルアップでは、ハイ インピーダンスの抵抗負荷を I/O ピンに接続して、ピンがフローティング状態にならないようにします。バス ホールドでは、バスがハイ インピーダンス状態になる前に、駆動されている I/O の最新の状態 (High または Low) で、I/O ピンを完全にラッチします。ソフトウェアでは、バス ホールドは「キーパ」と呼ばれます。

ロジックが High にも Low にもならない入力を、フローティング入力と呼びます。フローティング状態になると、入力バッファのゲートの電圧が High と Low の間で変動するので、過大な電力を消費する場合があります。このような場合、入力バッファでは P チャネル トランジスタと N チャネル トランジスタの両方がオン状態になっています。このような状況を回避するには、内部プルアップまたはバス ホールド回路を使用するように I/O を設定します。入力または双方向として設定された I/O が接続されているバスが通常の動作中にフローティング状態になることがわかっている場合は、その I/O でプルアップまたはバス ホールドを使用する必要があります。

ウィーク プルアップは適切はない場合があります。バスがほとんどプルアップされないデザインでは、電流がバッファを通ってウィーク プルアップからグランドに流れるので、バスが Low になってしまいます。このようなデザインでは、バス ホールド回路を使用することになります。

CMOS デバイスでは、入力がフローティング状態にならないようにしなければなりません。CoolRunner-II CPLD のウィーク プルアップおよびバス ホールドという機能は、入力ピンで過大な電力が消費される可能性を最小限にします。

電力消費を最小限にするには、各アプリケーションで I/O ごとに最適な I/O 終端処理を検討してください。可能な限り、入力がフローティング状態にならないようにしてください。

スルー レート

これは I/O 構造の機能で、出力バッファの状態が変化する速度を調節します。FAST および SLOW という 2 つのモードがあります。回路ボードのトレースにおける反射を削減する場合や RFI または EMI を最小限にする場合は、SLOW スルー レートを指定します。通常、低電力デザインを検討する場合は、システムの速度を遅くするので、反射は問題になりません。このような場合は、スルー レートを FAST に指定します。

出力として設定した I/O に入力バッファを接続すると、その I/O ピンでは電圧が変化します。このような出力を SLOW スルー レートに設定すると、出力電圧の状態が変換する速度が遅くなるので、標準的なロジック レベルになるまでの時間が長くなります。したがって、出力のスルー レートを FAST に設定した場合よりも、入力バッファの電圧が一定しない状態にある時間が長くなります。そして、入力バッファの P チャネル トランジスタと N チャネル トランジスタに電流が流れる時間が長くなるので、電力消費が大きくなります。このため、反射が問題ない場合は、出力のスルー レートを FAST に設定してください。

電力の節約方法

CMOS デバイスを使用する回路をデザインするときに電力消費を削減するには、いくつかの経験則に従う必要があります。このような経験則について説明する前に、電力消費の基本について理解する必要があります。そのため、初めに CMOS デバイスに流れる電流の計算式を説明します。電流の数学的なモデルを理解すると、経験則に従うのが容易になります。電力の節約を最大限にするため、CMOS デバイスをインプリメントするときはこのような考えを適用してください。

CMOS デバイスの電流計算式

CMOS デバイスは PMOS トランジスタと NMOS トランジスタから構成されているので、各トランジスタを動的にモデル化すると、単純な容量構造になります。キャパシタの基本的な構造は、2 つの電極の間にある誘電体です。CMOS デバイスの場合、キャパシタの誘電体はシリコン ウェハ上の酸化層であり、電極はポリまたはメタル ゲートとチャネル内の反転層です。内部を接続するメタル / ポリ配線も、キャパシタとしてモデル化されます。この場合、電極はこの配線自体と他の配線または導体で、誘電体は電極間の絶縁体です。そして、CMOS デバイス全体は、このようなキャパシタの集まりとしてモデル化されます。キャパシタに流れる電流の基本的な式は、次のようになります。

$$I = C \cdot \frac{dV}{dt}$$

この式を分解すると、周波数 (時間変化の逆数) に関する基本的な式を導くことができます。これを簡単にすると、次のようになります。

$$I = C \cdot dV \cdot \frac{1}{dt}$$

$$I = C \cdot dV \cdot f$$

CMOS デバイスの容量構造の電圧は High と Low の間で不連続に変化する矩形波として表され、理想的には High と Low 以外の値にはならないので、式をさらに簡単にできます。たとえば、電源電圧が V のシステムでは、電圧が V と 0 の間で変化するので、電圧変化 dV は次のように簡単に表現できます。

$$dV = V_2 - V_1$$

$$dV = V - 0$$

$$dV = V$$

したがって、各容量構造の電流式は、次のようになります。

$$I = C \cdot V \cdot f$$

変数の意味は、次のとおりです。

- I = 電流、単位はアンペア
- C = 容量構造の容量、単位はファラッド
- V = システム電圧、単位はボルト
- f = 容量構造の周波数、単位はヘルツ

動的なデバイスの電流は、すべての容量構造の時間変化を合計すると計算できます。電圧はすべての式で同じなので、定数として扱うことができます。したがって、容量構造が n 個あるデバイスの動的な電流は、次のようになります。

$$I_{Dynamic} = V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

デバイス全体の電流を得るには、動的な電流に静的な電流を加える必要があります。

$$I_{Total} = I_{Static} + V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

説明を簡単にするため、この式を次のように単純化します。

$$I_{CC} = I_{CCQ} + C \cdot V \cdot f$$

変数の意味は、次のとおりです。

- I_{CC} = デバイス全体の電流、単位はアンペア
- I_{CCQ} = デバイスの静的な電流、単位はアンペア
- C = デバイス全体の容量、単位はファラッド
- V = システム電圧、単位はボルト
- f = デバイスの平均周波数、単位はヘルツ

電力を計算するには、電流と電圧を掛けるとワット単位の値が得られます。

システム速度の低下

上記の式でデバイスの容量と電圧を一定にすると、デバイスが変化する平均的な速度を遅くすれば電力消費が少なくなることは明らかです。

システム クロックの速度とデータ バスの速度を小さい値に制限すると、デバイスが変化する平均的な速度が小さくなるので、電力消費が少なくなります。低電力デザインでは、必要な CPLD クロックの速度を慎重に解析することが重要です。必要以上にクロック周波数が大きい CPLD デザインでは、余計な電力が消費されます。CPLD ロジックに必要な最小限の速度を計算できれば、システム クロックが電力消費に与える影響が最小限になります。

入力を標準的なロジック レベルに駆動する

CMOS の入力バッファを標準的なロジック レベルに駆動しないと (つまり、フローティング状態の入力)、電力消費が大幅に増加します。電圧レベルが標準的なロジック レベルにない場合、入力バッファのトランジスタ (一般に P チャネルと N チャネル) が両方ともオン状態になるようにバイアスがかけられます。このようなバイアスがかけられると、このチャネルによって、デバイスの電源とグランドの間に大量の電流が流れます。したがって、これらの 2 つのトランジスタのいずれかをオフにするために入力バッファを High または Low に駆動して、このような状況にならないようにする必要があります。

CMOS ロジックの長所は、ロジック ゲートの入力レベルが特定の値に保たれていると電力消費がゼロに近くなるということです。CoolRunner-II CPLD の FZP 技術では、この原理を使用して他の CPLD よりも大幅に電力を節約しています。

さらに、入力レベルを完全な High または Low にすることも重要です。入力レベルを許容される電圧レベル (V_{IH} と V_{IL}) 内にするよりも、電圧を完全な High または Low にする方が流れる電流が少なくなるためです。この影響は、入力がフローティング状態になる場合ほどではありませんが、複数の I/O について電力消費を合計していくと無視できなくなります。

入力エッジ速度の増加

スイッチングが遅いソースによって駆動される CMOS デバイスの入力は、バイアスがかかって電圧が変化する時間が長くなるので、より多くの電力を消費します。標準的なロジック レベルの High と Low の間で入力にバイアスをかけると、電圧が変化します。このように CMOS の入力バッファにバイアスをかけると、P チャネル トランジスタと N チャネル トランジスタの両方がオンになるので、多くの電流がグランドに流れます。そして、バッファにバイアスをかける時間が長くなるほど、デバイスは多くの電力を消費します。したがって、CMOS デバイスの入力に割り当てる信号の切り替えは素早く行う必要があります。これは、すべてのクロック ピン、専用入力ピン、入力または双方向として設定された I/O に当てはまります。

バスの競合の回避

2 つの出力バッファが同時に逆方向にラインを駆動すると、バスの競合が発生します。これは、ロジックのパフォーマンスと電力消費に悪影響を与えます。2 つのドライバが反対の電圧レベルにバスを駆動しようとすると、過大な電流が流れます。

同じような状況は、プルアップ抵抗によって High になっているバスを、出力バッファが Low に駆動しようとする場合にも発生します。この場合、電流は電源からプルアップ抵抗と出力バッファの N チャネル トランジスタを通ってグランドに流れます。この場合の電流の値は、プルアップ抵抗の値と N チャネルのインピーダンスによって決定されます。このようなコンポーネントを使用する必要がある場合は、10KΩ程度のウィーク プルアップ抵抗の使用を推奨します。抵抗が大きい方が電力消費は少なくなりますが、バスがトライステート状態にあるときにプルアップ抵抗でバスを High レベルにする必要がある場合は、バスの応答時間が遅くなります。

電力消費を最小にするには、常に 1 つのデバイスについて同時にバス ラインを High または Low に駆動して、可能な限りプルアップ抵抗を使用しないようにします。状況によっては、このようにできない場合もあります。たとえば、SMBus や I2C SDA ラインはすべてのコンポーネントによって解放される必要がありますが、解放するときは High レベルにある必要があります。このような場合は、プルアップ抵抗が必要になります。

バスの終端処理

高速なデザインでは、反射を削減するためシャントによるバス終端処理が必要な場合に、プルダウン抵抗が必要になります。このような終端処理は、データ バス伝送ラインのインピーダンスと等しい値を持つプルダウン抵抗を使用して通常デザインされ、可能な限り負荷ピンの近くに配置されます。シャント抵抗がある伝送ラインを出力バッファが駆動する場合、P チャネル トランジスタが負荷に電流を供給するので、電力消費が増加します。

非常に短い伝送ラインを使用すれば、終端処理にシャント抵抗を使用する必要がなくなります。経験則から、短い伝送ラインの長さは立ち上がり時間の電気的長さを 1/6 未満にします。これは、次の式を使用して説明できます。

$$\text{Length} \ll \frac{1}{6} \cdot \frac{T_{rise}}{\sqrt{LC}}$$

変数の意味は、次のとおりです。

- Length = 最大のライン長、単位はインチ
- T_{rise} = 立ち上がり時間、単位は秒
- L = ラインのインダクタンス、単位はヘンリー /インチ
- C = ラインの容量、単位はファラッド/インチ

CoolRunner-II CPLD のスルー レート機能を使用して SLOW に設定すると、立ち上がり時間が長くなります。上記の式から、立ち上がり時間が長くなると、伝送ラインの長さが実質的に長くなり、反射も回避できます。スルー レートを SLOW にすることによって伝送ラインの長さが上記の式を満たして短くなれば、バス終端処理のシャント抵抗を使用する必要がなくなるので、電力が節約されます。

どちらの方法も採用できない場合は、伝送ラインのインピーダンスと同じ値の直列終端抵抗をソース ピンに接続します。この方法では、シャント負荷がないので過剰に電力が消費されず、ソースにおける反射を回避できます。ただし、負荷のインピーダンスが伝送ラインのインピーダンスと一致しないので、直列終端抵抗によって負荷における反射が発生します。これによって、入射波と負荷で発生する反射波という 2 つの遷移状態が、ソースと負荷の間にあるコンポーネントで発生します。

システム電圧の低下

全体的な電流の式で容量と平均周波数を一定にすると、システム電圧は推奨動作仕様内にある必要はありますが、システム電圧を低下すれば電力消費が削減されるのは明らかです。したがって、3.3V や 2.5V のデバイスの代わりに 1.8V のデバイスを使用することには意味があります。このため、CoolRunner-II CPLD は 1.8V デバイスです。また、CoolRunner-II CPLD のように微細なプロセス技術で製造されているデバイスは、一般に内部の全体的な容量が小さくなるので、さらに電力が節約されます。この 2 つの要因に加えて、システムの平均的な周波数を低くすると、さらに電力消費が削減されます。

どのような電圧のデバイスにも推奨動作範囲がありますが、この範囲内の低い電圧で動作する方が電力消費が削減されるので有利です。電圧が推奨動作範囲外になると、パフォーマンスが低下して電力が過剰に消費される可能性があります。

バス負荷の削減

CMOS デバイスに外部バスを接続すると、バスの負荷の影響によって電力消費が増加します。負荷の基本的な要因には、出力バッファに接続される容量性のバス コンポーネントと抵抗性のバス コンポーネントがあります。

容量性の負荷には、集中的なものと分散的なものがあります。一般に、集中的な容量は、他の CMOS デバイスの入力バッファのゲートによるものです。また、バスに接続されている容量性のエレメントによっても発生します。分散的な容量は、PCB 上のトレースの配線によって発生します。どちらの場合も、それまでのバスのロジックの状態に基づいて、ロジックが遷移するときに充電または放電が行われます。容量性の負荷があると、出力バッファがバスを駆動する CMOS デバイスから電流が流れるので、そのデバイスで明らかに電力消費が増加します。このような容量性負荷による電力消費を最小限にする

には、接続するデバイスの集中的な容量を小さくすることと、PCB のトレースを短くすることが必要です。このようにすると、反射が少なくなり、システムの速度が向上する可能性もあります。

抵抗性の負荷は、デバイスのインピーダンスより抵抗が小さい抵抗性のエレメントをバスに接続すると発生します。たとえば、プルダウン抵抗を PCB のトレースに接続すると、電流が CMOS デバイスの電源ラインから出力バッファ トランジスタの P チャネルを通り、抵抗を経由してグランドに流れるので、CMOS デバイスで電力消費が増加します。抵抗性の負荷を削減すると、電力消費も削減されます。

多くのコンポーネントは複数のタイプのインピーダンス エレメントから構成されていることに注意してください。たとえば、小規模ではありますが、キャパシタには抵抗性のエレメントと誘導性のエレメントも含まれています。

その他の電力の節約方法

ロジック デザインに関連するものなど、その他の電力の節約方法については、http://www.xilinx.com/xapp/xapp346.pdf にあるアプリケーション ノート XAPP346『Low Power Tips for CoolRunner Design (CoolRunner デザインでの低電力消費化のヒント)』を参照してください。XAPP346 は CoolRunner XPLA3 CPLD を対象にしていますが、基本的な原理は CoolRunner-II CPLD にも適用されます。

終わりに

CoolRunner-II 製品は、CPLD 業界で初めて高速ロジックと低電力の両立を実現しました。さらに、CoolCLOCK や DataGATE など CoolRunner-II CPLD 独自の機能によって、電力消費をさらに削減できます。これらの機能をこのアプリケーション ノートで説明したような優れたデザイン手法と組み合わせると、速度を犠牲にせずに電力を最大限に削減できます。

参考文献

1. Johnson H. および Martin G.、『High-Speed Digital Design: A Handbook of Black Magic』、1993 年、Prentice Hall PTR

改訂履歴

次の表に、このドキュメントの改訂履歴を示します。

日付	バージョン	改訂内容
04/12/02	1.0	初期リリース