XAPP583 (v1.0) 2012 年 5 月 31 日

スレーブ シリアル/SelectMAP モードで マイクロプロセッサを使用した 7 シリーズ FPGA のコンフィギュレーション

著者 : Matt Nielson

概要

エンベデッド システムの普及に伴い、コンポーネント数の削減と柔軟性の向上が追求されるようになりました。これらの目標は、システムに既に備わっているマイクロプロセッサを活用して FPGA をコンフィギュレーションすることで同時に達成できます。このアプリケーション ノートでは、マイクロプロセッサを使用したザイリンクスの 7 シリーズ FPGA のコンフィギュレーションについて詳しく説明します。ここには、スレーブ シリアルまたは SelectMAP モードを使用するアプリケーション例を示す C コードも含まれています。例では、MicroBlaze™ プロセッサを使用して Kintex™ XC7K325T デバイスをコンフィギュレーションします。

システムの概要

今日のシステムには、これまで以上に小型かつ安価で高い機能性が求められます。さらに、ザイリンクスの FPGA は各世代ごとにその性能および機能が向上しています。ザイリンクス FPGA は、サード パーティのフラッシュ メモリからの直接コンフィギュレーションをサポートしていますが、十分なメモリを持つ外部のエンベデッド プロセッサがシステムに存在する場合は、エンベデッド プロセッサ ベースのコンフィギュレーション ソリューションによって、高度な FPGA コンフィギュレーションの実行とボード スペース要件の縮小が可能になります。このようなコンフィギュレーションには、FPGA が機能する前にプロセッサが動作できる状態であることが必要です。

このアプリケーション ノートでは、エンベデッド プロセッサから FPGA をコンフィギュレーションする方法を説明します。FPGA コンフィギュレーション データ (別名コンフィギュレーション ビットストリーム) はザイリンクス デザイン ツールで生成します。生成されたビットストリームはコンフィギュレーション時に、上記ソリューションを使用して FPGA にロードされます。図 1 にシステム図を示します。

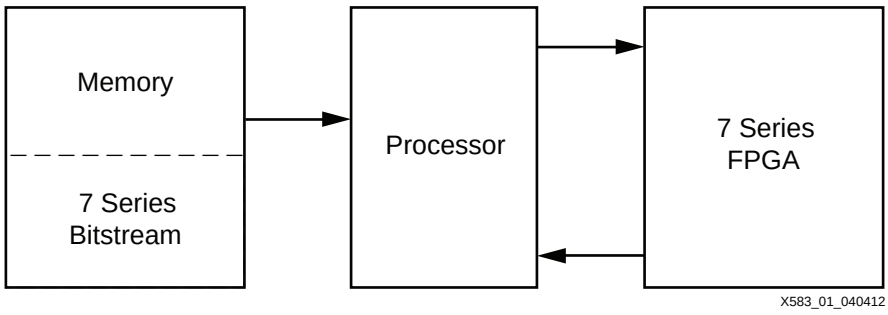


図 1 : システム図

その他タスクの実行を主な目的とするマイクロプロセッサは、ザイリンクス FPGA にコンフィギュレーション データをロードするためにも使用できます。プロセッサを使用すると、たとえば複数あるコンフィギュレーション ファイルのどれを用いて FPGA をプログラムするかを選択できるなど柔軟性が向上します。

コンフィギュレーションについて

マイクロプロセッサを使用した 7 シリーズ FPGA のコンフィギュレーションは、スレーブ シリアルまたは SelectMAP モードのいずれかで実行できます。これら 2 つのモードにはいくつかの共通点があります。中でも、基本的なコンフィギュレーション シーケンスが同じである点が最も重要です。

JTAG コンフィギュレーション インターフェイスもマイクロプロセッサで制御可能ですが、このアプリケーション ノートでは扱いません。

基本的なコンフィギュレーション シーケンス

図 2 に基本的なコンフィギュレーション シーケンスを示します。

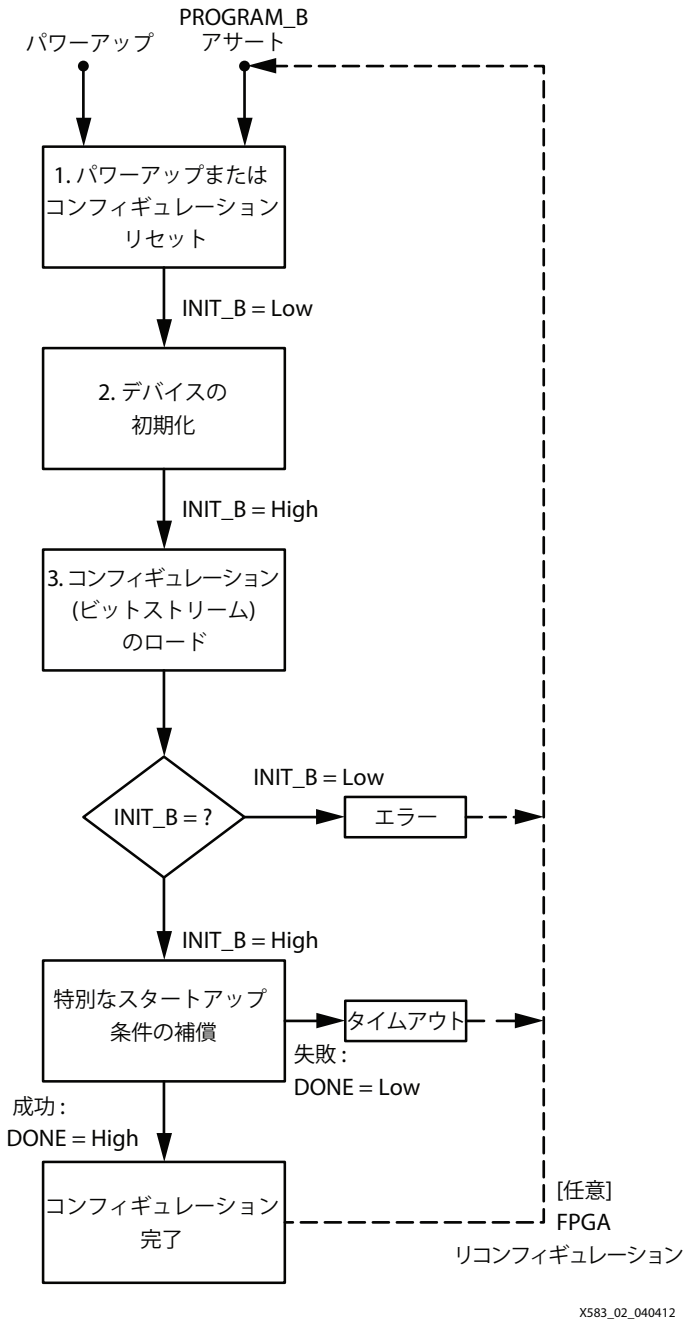


図 2 : 基本的なコンフィギュレーション シーケンス

1. パワーアップまたはコンフィギュレーション リセット：コンフィギュレーション シーケンスは、パワーアップまたはコンフィギュレーションのリセットで開始します。パワーアップとは、FPGA に最初に電源が投入されたときです。また、コンフィギュレーションのリセットは PROGRAM_B ピンがアサートされたときに発生します。
2. デバイスの初期化 ： コンフィギュレーション メモリの初期化はパワーアップまたはコンフィギュレーションのリセットによって開始されます。初期化中、FPGA の INIT_B ピンは Low に駆動され、内部のコンフィギュレーション ステート マシンがリセットされると共に、コンフィギュレーション メモリがクリアされます。初期化が完了すると、INIT_B ピンはリリースされて高インピーダンス状態になり、FPGA はこのピンが High になるまで待機します。

   リリースされて高インピーダンス状態にある INIT_B を High にするには、外部抵抗が必要です。INIT_B が High になると、FPGA はコンフィギュレーション モード ピン M[2:0] をサンプリングします。後続するコンフィギュレーション手順のモードはモード ピンによって決定します。M[2:0] が `111` の場合、FPGA はスレーブ シリアル コンフィギュレーション モードに設定されます。M[2:0] が `110` の場合は、スレーブ SelectMAP コンフィギュレーション モードになります。モード ピンのサンプリングが終わると、FPGA ではコンフィギュレーション データ (別名コンフィギュレーション ビットストリーム) の受信準備が完了します。
3. コンフィギュレーションのロード：スレーブ コンフィギュレーション モードの場合、外部のマイクロプロセッサはコンフィギュレーション シーケンスのこの手順でデバイスにビットストリームをロードできます。スレーブ シリアル モードの場合は、CCLK の立ち上がりエッジごとに FPGA の D01_DIN ピンを介して、ビットストリームが 1 ビットずつロードされます。スレーブ SelectMAP モードでは、FPGA の CSI_B および RDWR_B ピンが Low の場合に、CCLK の立ち上がりエッジごとに FPGA の D[31:00] ピンを介してデータがロードされます。CCLK の立ち上がりエッジごとにロードするデータ幅が 8、16 または 32 ビットか (それぞれ、SelectMAP D[07:00]、D[15:00] または D[31:00] を介する) を判断するビットストリームの自動バス幅検出パターンについては、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) を参照してください。

コンフィギュレーション シーケンスは、ビットストリームのロード終了で完了します。

コンフィギュレーション ビットストリームの詳細

ザイリンクス デザイン ツールを使用して、一致するターゲット デバイス ID を検索するために必要なすべてのコマンドおよびデータを含む FPGA コンフィギュレーション ビットストリームを作成し、コンフィギュレーション メモリ データをロードして FPGA の設計を開始します。ビットストリームの構成については、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』を参照してください。

一般に、プロセッサ コードでビットストリームの構成を把握している必要はありません。ただし、正しく FPGA の設計を開始し、プロセッサのコンフィギュレーション コードを検証してデバッグするには、ビットストリームに含まれるいくつかの主要コマンドを理解しておく必要があります。

同期ワード

ビットストリームの先頭には、いくつかのパディング ビットと自動バス幅検出パターンに続いて 32 ビットの同期ワード (`0xAA995566`) があります。ビットストリームを正しく FPGA にロードするには、プロセッサから FPGA に送信される同期ワードのビット順 (またはバイト順) が最も重要なサインです。

デバイス ID

ビットストリームには、同期ワードに続いてデバイス ID チェックがあります。これにより、ビットストリームが適切なターゲット デバイスで受信されているかが確認されます。デバイス ID の確認でエラーが発生すると、FPGA が INIT_B ピンを Low 駆動してコンフィギュレーション エラーを示します。

CRC チェック

ビットストリームの末尾近くには、ビットストリーム内の正しい値に対してコンフィギュレーションの内部の周期的冗長検査 (CRC) を確認するためのコマンドがあります。デバイスによって認識された CRC チェック コマンドで算出された CRC が正しい CRC 値と一致しない場合、FPGA が INIT_B ピンを Low 駆動してコンフィギュレーション エラーを示します。

スタートアップ

デバイスがビットストリームからのコンフィギュレーション データをすべて受信し、CRC チェックでエラーが検出されなければ、ビットストリームの末尾近くにあるスタートアップ コマンドによってロードされたデザインのスタートアップ シーケンスを開始します。スタートアップ シーケンスの詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) を参照してください。このシーケンス中、DONE ピンはリリースされて高インピーダンス状態になり、FPGA はこのピンが High になるまで待機します。DONE ピンを High にするには、強い外部のプルアップ抵抗 (または BitGen DriveDONE オプション) が必要です。スタートアップシーケンスは、FPGA が EOS (End Of Start-up) 状態に到達すると完了します。

注記：DONE 信号は EOS の前にリリースされます。そのため、プロセッサ コードは、DONE が High に遷移しても、ビットストリームや CCLK パルスを停止しない記述になっている必要があります。

プロセッサ コードはビットストリームの全ビット/ワードが FPGA に取り込まれるよう記述されている必要があります。通常、スタートアップ シーケンスはビットストリームの最後のビットが FPGA に取り込まれる前に EOS に到達して完了します。

特別なスタートアップ条件

BitGen オプションには、ビットストリームの最後に到達してもスタートアップ シーケンスを延長し、FPGA のスタートアップに影響を与えるものがあります。このようなオプションのいくつかでは、コンフィギュレーション ビットストリームが取り込まれても、FPGA に対して追加の CCLK パルスを送信します。スタートアップに影響を与える BitGen オプションの例としては、LCK_CYCLE や MATCH_CYCLE があります。

あらゆるビットストリームのスタートアップ オプションに対応するため、プロセッサ コードは次のように動作する記述になっている必要があります。

1. 全ビットストリーム データをロードする。
2. DONE がアサートして High になるまで (D01_DIN または D[31:00] のデータビットがすべて 1 の間)、CCLK サイクルを継続する。
3. FPGA のスタートアップ シーケンスが確実に終了するように、DONE がアサートして High になった後もさらに CCLK を 8 サイクル間継続する。

LCK_CYCLE または MATCH_CYCLE が選択されている場合、LCK_CYCLE または MATCH_CYCLE の値が DONE_CYCLE の値よりも小さければ上記シーケンスは正常に動作します。これにより、DONE はスタートアップ延長イベントの発生後にトグルします。その他の BitGen オプションと詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』および ISE® Design Suite マニュアルに含まれている『コマンドラインツールユーザーガイド』(UG628) の「BitGen」を参照してください。

スレーブ シリアル コンフィギュレーション

INIT_B が High になると、CCLK の立ち上がりエッジごとに (D01_DIN ピンに現れる) 1 ビットのスレーブ シリアル コンフィギュレーション データがコンフィギュレーション ロジックにロードされます。セットアップ タイムおよびホールド タイムの仕様については、該当するデータシートを参照してください。表 1 にスレーブシリアルコンフィギュレーションで使用されるピンを示します。

表 1 : スレーブ シリアルで使用されるピン

信号名	方向	説明
CCLK	入力	コンフィギュレーション クロックです。
PROGRAM_B	入力	コンフィギュレーション ロジックに対するアクティブ Low のリセット信号です。
INIT_B	入力/出力	アクティブ Low の FPGA 初期化ピンです。デバイスがコンフィギュレーション データを受信する準備ができていることを示します。また、コンフィギュレーション エラーも示します。外部から Low に保持してコンフィギュレーションを遅延させることができます。
DONE	入力/出力	コンフィギュレーションが完了したことを示します。外部から Low に保持してスタートアップを遅延させることができます。
M[2:0]	入力	コンフィギュレーション モードの選択ピンです。
D01_DIN	入力	シリアル コンフィギュレーションのデータ入力です。
DOUT	出力	シリアル デイジー チェーンのデータ出力です。

スレーブ SelectMAP コンフィギュレーション

CCLK の立ち上がりエッジごとに、D[07:00] バスに現れる 1 バイトのスレーブ SelectMAP x8 データバス コンフィギュレーション データがロードされます。詳細は、6 ページの「データ ファイルの形式とビット スワップ」を参照してください。表 2 に SelectMAP で使用されるピンを示します。

表 2 : スレーブ SelectMAP で使用されるピン

信号名	方向	説明
CCLK	入力	コンフィギュレーション クロックです。
PROGRAM_B	入力	コンフィギュレーション ロジックに対するアクティブ Low のリセット信号です。
INIT_B	入力/出力	アクティブ Low の FPGA 初期化ピンです。デバイスがコンフィギュレーション データを受信する準備ができていることを示します。また、コンフィギュレーション エラーも示します。外部から Low に保持してコンフィギュレーションを遅延させることができます。
DONE	入力/出力	コンフィギュレーションが完了したことを示します。外部から Low に保持してスタートアップを遅延させることができます。
M[2:0]	入力	コンフィギュレーション モードの選択ピンです。
D[31:00]	入力	パラレル コンフィギュレーションのデータ入力です。
CSI_B	入力	アクティブ Low のチップ セレクトです。
RDWR_B	入力	アクティブ Low の書き込みセレクトまたは読み出しセレクトです。

SelectMAP では、CSI_B と RDWR_B という追加の制御信号が 2 つ存在します。FPGA にコンフィギュレーション バイトを転送するには、これらの信号は共に Low にアサートされている必要があります。

注記：各信号のセットアップおよびホールド仕様については、該当する 7 シリーズ FPGA ファミリのデータシートを参照してください。このアプリケーション ノートのスレーブ SelectMAP コンフィギュレーションのリファレンス デザインでは、8 ビットの SelectMAP バスを介したコンフィギュレーションのみを示しています。7 シリーズ FPGA ファミリでは 16 ビットまたは 32 ビット幅の SelectMAP バスもサポートされています。必要箇所を修正し、ビット順に注意することで、このリファレンス デザインは 16 ビットまたは 32 ビット幅の SelectMAP バス用としても使用できます。詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) を参照してください。

データ ファイルの形式とビット スワップ

コンフィギュレーション ビットストリームはプロセッサに接続されたメモリにロードされるため、プロセッサ (またはメモリをプログラムする別のデバイス) が利用可能な形式にフォーマットされている必要があります。さまざまなソリューションをサポートできるように、ザイリンクス ツールは複数の異なる形式でファイルを生成できます (表 3 参照)。また、PROM ファイル生成ツールの PROMGen は、1 つ以上のビットストリーム ファイルを 1 つの PROM ファイルに変換します。PROM ファイルは PROM 以外でも使用可能です。これらは任意の場所に保存し、任意の手段で取り込むことができます。

表 3 : ザイリンクス ツールでサポートされているファイル形式

ファイル拡張子	説明
.bit	FPGA へのダウンロードが不要なヘッダー情報を含むバイナリ ファイルです。
.rbt	テキスト ヘッダーと ASCII 形式の 1 と 0 を含む ASCII ファイルです。
.bin	ヘッダー情報を含まないバイナリ ファイルです。
.mcs	アドレスおよびチェックサム情報を含む ASCII 形式の PROM ファイルです。
.hex	データのみを含む ASCII 形式の PROM ファイルです。

スレーブ シリアル コンフィギュレーションでのデータ順は非常に単純です。ロードは、ビットストリームの最初のビットに始まり、一度に 1 ビットずつ、ファイルの最後に到達するまで継続します。

一方、スレーブ SelectMAP コンフィギュレーションでのデータ順は若干複雑です。ここでは、8 ビット幅の SelectMAP パラレル バスでのビット順を説明します。16 ビットまたは 32 ビット幅の SelectMAP バス使用時におけるパラレル バスでのビット順については、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) を参照してください。コンフィギュレーション データは、CCLK の立ち上がりエッジごとに 1 バイトずつロードされ、各バイトの最上位ビット (MSB) が D[07] ピンではなく、D[00] ピンに現れます。このように順序が従来とは異なるため、.bin ファイルからのデータがそのまま現れると、不正なデータになります。これは、大半のプロセッサでは (D[00] ではなく) D[07] が各バイトの MSB と判断されるためです。プロセッサの D[07] を FPGA の SelectMAP データ バスの D[07] に接続すると、データは事実上逆にロードされることになり、適切にコンフィギュレーションが完了しません。このため、元のデータストリームのビットをスワップして、各バイトのビット順を逆にする必要があります。図 3 に 2 バイト (0xABCD) のビット スワップを示します。

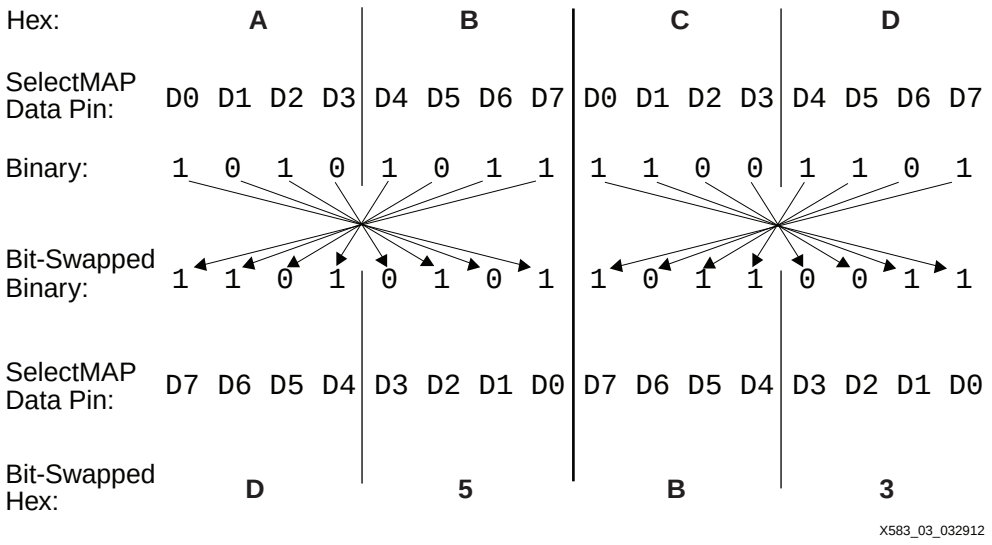


図 3 : x8 におけるビットのスワップ例

データの方向に関係なく、データにおける先頭バイトの MSB は D[00] に送信されます。ただし、ビット スワップしたデータでは右端のビット、ビット スワップしていないデータでは左端のビットが D[00]

に送信されます。ザイリンクス ツールを使用した場合、.mcs ファイルは常にビット スワップし、.bit、.rbt および .bin ファイルでビット スワップすることはありません。16 進形式のファイルは、PROMGen ツールのコマンド ライン オプションに従ってビット スワップ可能です。

注記：データのビット スワップが必要か否かは、プロセッサまたはアプリケーションに完全に依存し、一般にスレーブ SelectMAP アプリケーションに対してのみ適用されます。ビット スワップしていないデータはスレーブ シリアルでのダウンロードにのみ使用してください。

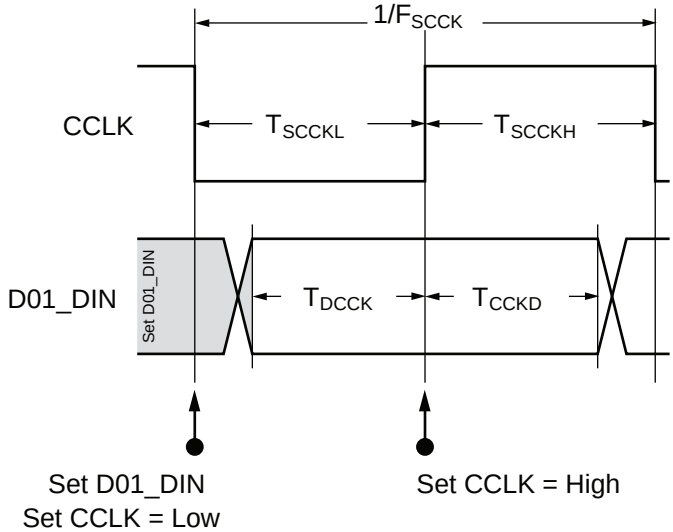
エラーおよびトラブルシューティング

正常にコンフィギュレーションされていない場合、すべてのデータがロードされても DONE ピンは High になりません。これにはさまざまな原因が考えられます。考えられるコンフィギュレーション エラーの原因については、3 ページの「コンフィギュレーション ビットストリームの詳細」を参照してください。

一般的なデバッグ法は次のとおりです。

- ザイリンクス コンフィギュレーション ケーブルと JTAG コンフィギュレーション モードを使用し、そのビットストリームで FPGA をコンフィギュレーションし、デバイスのステータス情報を読み出すことが可能かを確認する。
- ビット順と、必要であればビットがスワップされていることを確認する。ビット順の基準となる値は 3 ページの「同期ワード」を参照。
- データシートに記載されているコンフィギュレーションのタイミング要件 (図 4 参照) が満たされていることを確認する。
- ビットストリームが取り込まれても DONE が High にならない場合は、INIT_B ピンを確認する。
  - INIT_B ピンが High の場合
    - ビットストリームの先頭にある同期ワードが認識されなかった可能性があります。同期ワードが正しく取り込まれることを確認してください。
    - ビットストリームが最後まで受信されていない可能性があります。
  - INIT_B ピンが Low の場合
    - コンフィギュレーション中にエラーが検出された可能性があります。ビットストリームがターゲット デバイス向けのものであることを確認してください。

図 4 にコンフィギュレーションのタイミング要件を示します。



Symbols:

F_{SCCK} = Maximum CCLK frequency
$1/F_{SCCK}$ = Minimum CCLK period
T_{SCCKL} = Minimum CCLK low time
T_{SCCKH} = Minimum CCLK high time
T_{DCCK} = Minimum D01_DIN to CCLK rising-edge setup time
T_{CCKD} = Minimum CCLK rising-edge to D01_DIN hold time

Sample Sequence for a CCLK Cycle:

1. Set CCLK = Low and set D01_DIN = new_value.
2. Delay for Minimum[Maximum[T_{SCCKL}, T_{DCCK}]].
3. Set CCLK = High.
4. Delay for Minimum[Maximum[T_{SCCKH}, T_{CCKD}]].
5. Repeat from step 1.

X583_04_050412

図 4 : コンフィギュレーションのタイミング要件

図 5 にシリアル コンフィギュレーションのシーケンスを示します。

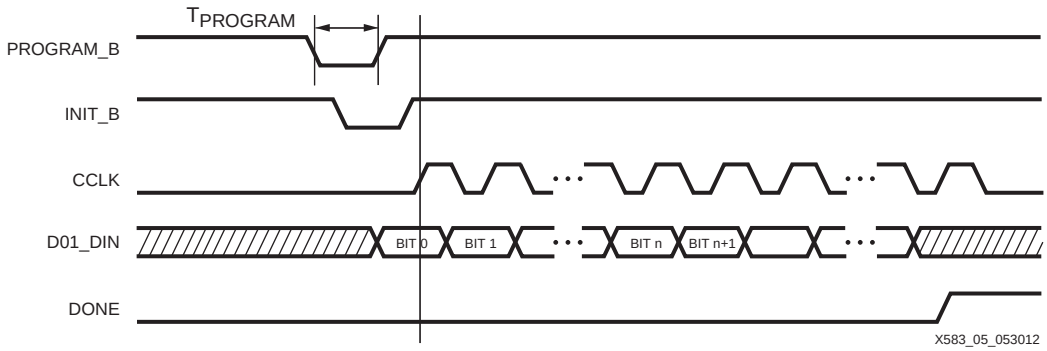


図 5 : シリアル コンフィギュレーションのクロック シーケンス

図 5 では、同期ワード前およびコンフィギュレーション完了後のデータは無視されています。

図 6 に SelectMAP コンフィギュレーションのシーケンスを示します。

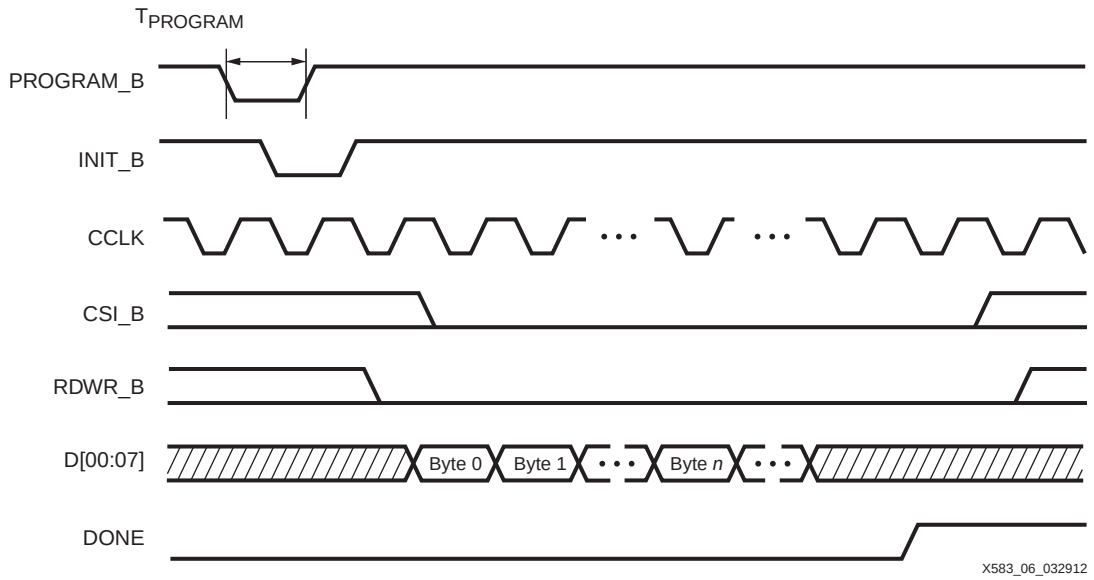


図 6 : SelectMAP コンフィギュレーションのクロック シーケンス

ハードウェアのインプリメンテーション

マイクロプロセッサ

リファレンス デザインでは、ザイリンクスの MicroBlaze プロセッサが使用されています。MicroBlaze プロセッサの詳細は、japan.xilinx.com/microblaze を参照してください。

電圧の互換性

プロセッサの I/O は、接続されている FPGA ピンと互換性のある電圧をサポートしている必要があります。コンフィギュレーション インターフェイスには、バンク 0 の JTAG とコンフィギュレーション専用ピン、およびバンク 14 の多目的ピンが含まれます。バンク 0 およびバンク 14 で適切なコンフィギュレーション インターフェイスの電圧をサポートするには、コンフィギュレーション用バンクの電圧セレクト ピン (CFGBVS) を High または Low に設定して、コンフィギュレーション用 I/O がそれぞれ 3.3V/2.5V または 1.8V で動作するように設定する必要があります。通常は、コンフィギュレーション インターフェイスのすべてのピンに一貫した I/O 電圧インターフェイスを確保するために、これら 2 つのバンクには同じ V_{CCO} 電圧が供給されます。Virtex-7 FPGA のバンク 14 は HP (High-Performance) バンクであるため、1.8V 以下の I/O 規格にのみ対応しています。デザインは表 4 に示す電圧要件を満

たす必要があります。

表 4 : 電圧の互換性要件

プロセッサ GPIO	FPGA バンクの V_{CCO} 要件		CFGBVS	FPGA ファミリのサポート			JTAG[1]
	バンク 0	バンク 14		Artix™-7	Kintex-7	Virtex™-7	
1.8V	1.8V	1.8V	GND	*	*	*	1.8V
2.5V	2.5V	2.5V	V_{CCO_0} = 2.5 V	*	*	N/A	2.5V
3.3V	3.3V	3.3V	V_{CCO_0} = 3.3 V	*	*	N/A	3.3V

注記 :

1. JTAG ピンはバンク 0 にあります。そのため、JTAG 信号の電圧はバンク 0 の電圧要件に準じます。

スレーブ SelectMAP のハードウェア

ここでは、スレーブ SelectMAP モードでマイクロプロセッサから FPGA デバイスをコンフィギュレーションするリファレンス デザインについて説明します。SelectMAP コンフィギュレーション モードは最速のコンフィギュレーション モードです。図 7 に示されているように、このデザインでは、外部メモリに格納されている FPGA コンフィギュレーション データが読み出され、ターゲット FPGA が直接コンフィギュレーションされます。

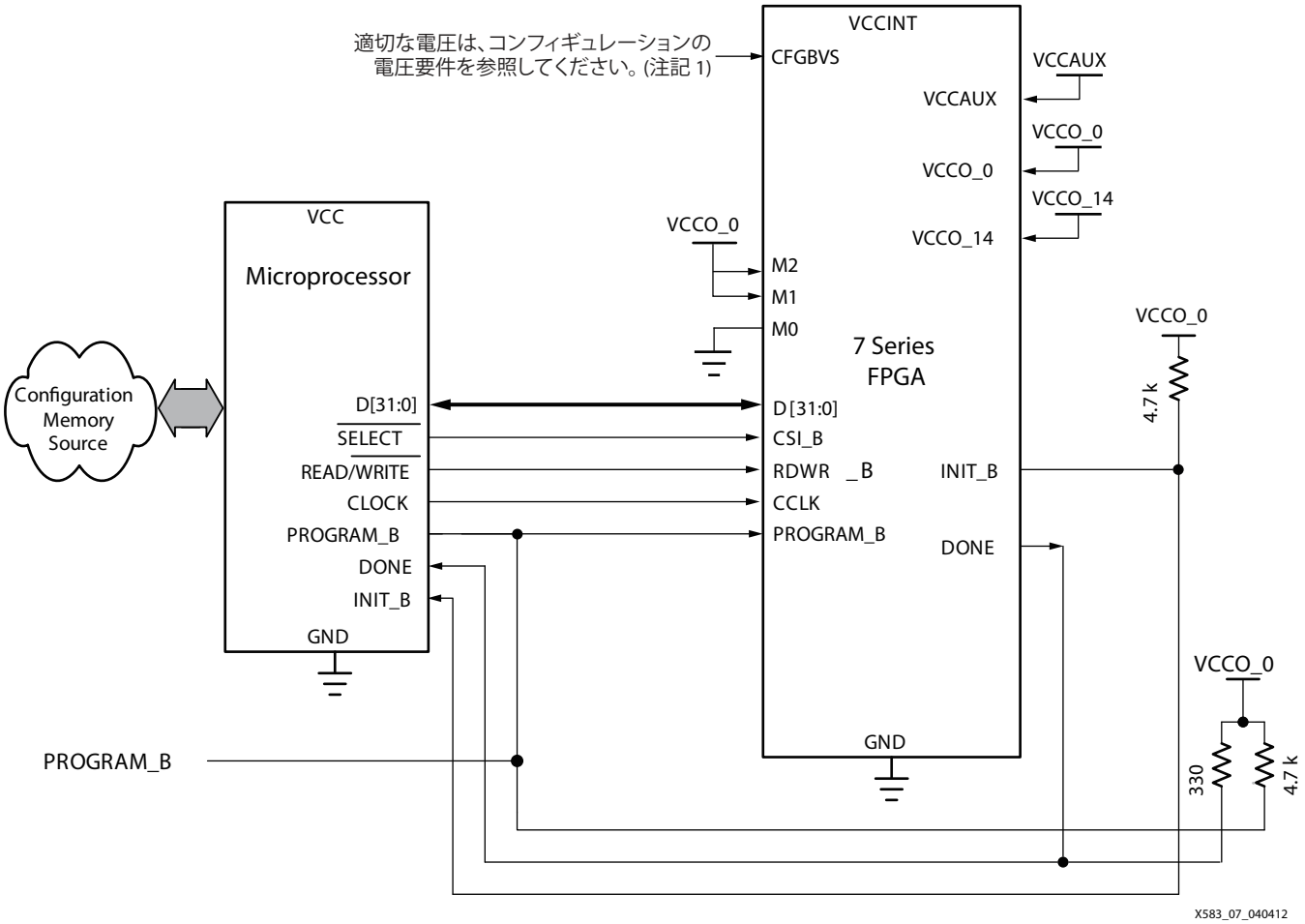


図 7 : スレーブ SelectMAP コンフィギュレーション

図 7 の注記

1. 電圧要件については、9 ページの表 4 を参照してください。

このリファレンス デザインでは、FPGA コンフィギュレーション データはマイクロコントローラーによって 1 ワード (32 ビット) ずつメモリから読み出されます。インターフェイスには、8 ビットの SelectMAP が使用されています。D[7:0] への正しいバイト順での送信やバイトごとの CCLK のアサートは、単純なソフトウェア プログラムで管理されます。ソフトウェア プログラムはまた、PROGRAM_B をアサートしてコンフィギュレーション シーケンスを開始し、DONE および INIT_B 信号を確認することでコンフィギュレーションのステータスやエラー確認も行います。

スレーブ SelectMAP の擬似コード

ここで説明する擬似コードによって、マイクロプロセッサで次の動作が可能になります。

- メモリから FPGA のコンフィギュレーション データを読み出す
- CCLK を生成する
- FPGA にバイト幅のデータを送信する

スレーブ SelectMAP の擬似コード内では、コンフィギュレーションは main() および shift_word_out() 関数によって実行されます。コンフィギュレーションを開始するには CSI_B ピン、RDWR_B ピンおよび PROGRAM_B ピンを Low にします。次に PROGRAM_B ピンがアサートされ、必要な PROGRAM_B パルス幅またはそれ以上が経過した後にディアサートされます。コードは必要に応じて、INIT_B のディアサートを待機します。

注記：PROGRAM_B パルス幅に対する個別の要件については、ターゲット FPGA のデータシートで $T_{PROGRAM}$ を参照してください。FPGA の電源投入が完了した、または PROGRAM_B のアサート以外の手段で FPGA がリセットされた場合、PROGRAM_B のディアサートは必須ではありません。

PROGRAM_B および INIT_B がディアサートされると、メモリに格納されているターゲット FPGA ビットストリームの各 32 ビットワードに対して shift_word_out() が呼び出されます。この関数は CCLK をディアサートし、現ワードの次のバイトを D[7:0] に送信すると、CCLK をアサートします。

main() は対象ビットストリームを送信し、ターゲット FPGA が DONE をアサートするまで待機します。また、特別なスタートアップ条件が満たされるように、追加で 8 サイクルの CCLK のアサートおよびディアサートをターゲット FPGA に送信します。

注記：次の擬似コードは、MicroBlaze プロセッサに実装されているようなメモリ マップされた I/O 構造を表します。リファレンス デザイン内の `slave_selectmap.c` ソース ファイルは簡単に移植できますが、読み出しコマンド、書き込みコマンド、およびアドレスを新しいシステム用に変更する必要がある場合があります。また、多くのシステムでは GPIO インスタンスが 1 つないし 2 つ存在するため、ピンを個別に制御するには、ビット マスクが必要な場合もあります。このリファレンス デザインの C コードはそれぞれにメモリマップされたペリフェラルとして I/O ピンにアクセスし、ポインターを使用してこれらの I/O ピンの読み出しおよび書き込みを行います。これにより、コードは単純かつ多様なマイクロプロセッサに容易に移植可能になります。

```
/* Global defines
 * Define the addresses for the I/O peripherals used to control and
 * monitor the target FPGA. Also define the location in memory the
 * bitstream is stored and its size. These are system dependent and
 * should be adjusted as needed
 */

/* Output GPIO addresses */
CCLK_GPIO_BASEADDR = 0x40020000
PROGRAM_B_GPIO_BASEADDR = 0x40030000
DATA_OUT_GPIO_BASEADDR = 0x40040000
```

```
RDRW_B_GPIO_BASEADDR = 0x40050000
CSI_B_GPIO_BASEADDR = 0x40060000

/* Input GPIO addresses */
INIT_B_GPIO_BASEADDR = 0x40070000
DONE_GPIO_BASEADDR = 0x40080000

/* Location in memory and size of the target bitstream */
MEMORY_BASEADDR = 0xC0000000
BITSTREAM_START_ADDR = MEMORY_BASEADDR + 0x2000000
BITSTREAM_SIZE_BYTES = 0xAEA68C

/* PROGRAM_B pulse width. Check the target FPGA data sheet for the
 * TPROGRAM pulse width. One microsecond is safe for any 7 series FPGA
 */
TPROGRAM = 1 /* Assumes sleep() is microseconds */

/* Serialize word and clock each bit on target's DIN and CCLK pins */
shift_word_out(data32)
{
   *cclk = CCLK_GPIO_BASEADDR
   *data_out = DATA_OUT_GPIO_BASEADDR

   /* Sequence the 32-bit word into bytes. The endianess can be either
    * little or big. The following assumes the sync word is read from
    * external memory as 0x665599AA (instead of 0xAA995566).
    */
   byte[0] = data32 >> 24
   byte[1] = data32 >> 16
   byte[2] = data32 >> 8
   byte[3] = data32

   *cclk = 0

   for (i = 0; i < 4; ++i) {
      *data_out = byte[i]
      shift_cclk(1)
   }
}

/* Assert and Deassert CCLK */
shift_cclk(count)
{
   *cclk = CCLK_GPIO_BASEADDR

   *cclk = 0
   for (i = 0; i < count; --i) {
      *cclk = 1
      *cclk = 0
   }
}

int main()
{
   bits_start = BITSTREAM_START_ADDR
   bits_size = BITSTREAM_SIZE_BYTES

   *program_b = PROGRAM_B_GPIO_BASEADDR
   *rdrw_b = RDRW_B_GPIO_BASEADDR
   *csi_b = CSI_B_GPIO_BASEADDR
   *init_b = INIT_B_GPIO_BASEADDR
   *done = DONE_GPIO_BASEADDR
```

```
    /* Bring csi_b, rdwr_b Low and program_b High */
    *program_b = 1
    *rdrw_b = 0
    *csi_b = 0
    /* Configuration Reset */
    *program_b = 0
    sleep(TPROGRAM)
    *program_b = 1

    /* Wait for Device Initialization */
    while(*init_b == 0)
       ;

    /* Configuration (Bitstream) Load */
    for (i = 0; i < bits_size ; i+=4) {
       shift_word_out(bits_start + i)
    }

    /* Check INIT_B */
    if (*init_b_pointer == 0) {
       return 1
    }

    /* Check INIT_B */
    if (*init_b == 0) {
       return 1
    }

    /* Wait for DONE to assert */
    while(*done == 0)
       ;

    /* Compensate for Special Startup Conditions */
    shift_cclk(8)
    return 0
}
```

スレーブ シリアルのハードウェア

ここでは、スレーブ シリアル モードでマイクロプロセッサから 7 シリーズ FPGA をコンフィギュレーションするリファレンス デザインについて説明します。スレーブ シリアル コンフィギュレーションでは、7 シリーズ FPGA にシリアル クロックが供給され、クロックの立ち上がりエッジごとに 1 ビットのデータがコンフィギュレーション ビットの最後まで送信されます。マイクロプロセッサは、外部メモリからコンフィギュレーション ビットストリームを読み出します。図 8 にスレーブ シリアル システムのレイアウトを示します。

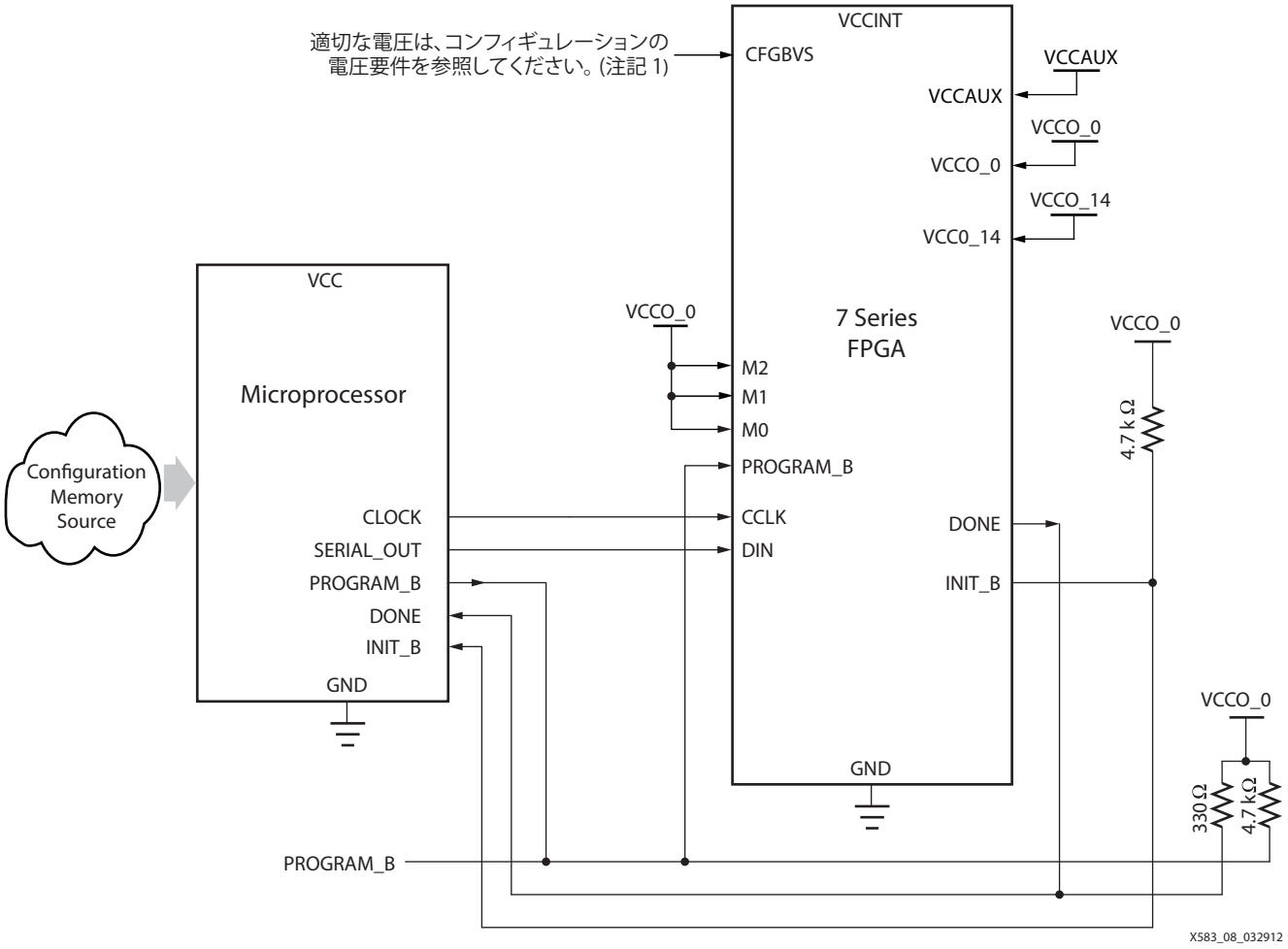


図 8 : スレーブ シリアル モードによるコンフィギュレーションの例

図 8 の注記

1. 電圧要件については、9 ページの表 4 を参照してください。

スレーブ シリアルの擬似コード

ここで説明する C コードによって、マイクロプロセッサで次の動作が可能になります。

- メモリから FPGA のコンフィギュレーション データを読み出す
- CCLK を供給する
- ビットストリームをシリアル化する

`slave_serial.c` ソースファイル内では、コンフィギュレーションは main() および shift_word_out() 関数によって実行されます。PROGRAM_B ピンがアサートされ、必要な PROGRAM_B パルス幅またはそれ以上が経過した後にディアサートされると、コンフィギュレーションが開始します。コードは必要に応じて、INIT_B のディアサートを待機します。

注記：PROGRAM_B パルス幅に対する個別の要件については、ターゲット FPGA のデータシートで $T_{PROGRAM}$ を参照してください。FPGA の電源投入が完了した、または PROGRAM_B のアサート以外の手段で FPGA がリセットされた場合、PROGRAM_B のディアサートは必須ではありません。

PROGRAM_B および INIT_B がディアサートされると、メモリに格納されているターゲット FPGA ビットストリームの各 32 ビットワードに対して shift_word_out() が呼び出されます。この関数は

CCLK をディアサートし、現ワードの次のバイトをシリアル化してそれをターゲット FPGA の DIN ピンに送信すると、CCLK をアサートします。

main() は対象ビットストリームを送信し、ターゲット FPGA が DONE をアサートするまで待機します。また、特別なスタートアップ条件が満たされるように、追加で 8 サイクルの CCLK のアサートおよびディアサートをターゲット FPGA に送信します。

注記：次の擬似コードは、MicroBlaze プロセッサに実装されているようなメモリ マップされた I/O 構造を表します。リファレンス デザイン内の `slave_serial.c` ソース ファイルは簡単に移植できますが、読み出しコマンド、書き込みコマンド、およびアドレスを新しいシステム用に変更する必要がある場合があります。また、多くのシステムでは GPIO インスタンスが 1 つないし 2 つ存在するため、ピンを個別に制御するには、ビット マスクが必要な場合もあります。このリファレンス デザインの C コードはそれぞれにメモリ マップされたペリフェラルとして I/O ピンにアクセスし、ポインターを使用してこれらの I/O ピンの読み出しおよび書き込みを行います。これにより、コードは単純かつ多様なマイクロプロセッサに容易に移植可能になります。

```
/* Global defines
 * Define the addresses for the I/O peripherals used to control and
 * monitor the target FPGA. Also define the location in memory the
 * bitstream is stored and its size. These are system dependent and
 * should be adjusted as needed
 */

/* Output GPIO addresses */
CCLK_GPIO_BASEADDR = 0x40020000
PROGRAM_B_GPIO_BASEADDR = 0x40030000
SERIAL_OUT_GPIO_BASEADDR = 0x40040000

/* Input GPIO addresses */
INIT_B_GPIO_BASEADDR = 0x40050000
DONE_GPIO_BASEADDR = 0x40060000

/* Location in memory and size of the target bitstream */
MEMORY_BASEADDR = 0xC0000000
BITSTREAM_START_ADDR = MEMORY_BASEADDR + 0x2000000
BITSTREAM_SIZE_BYTES = 0xAEA68C

/* PROGRAM_B pulse width. Check the target FPGA data sheet for the
 * TPROGRAM pulse width. One microsecond is safe for any 7 series FPGA
 */
TPROGRAM = 1 /* Assumes sleep() is microseconds */
/* Serialize a 32-bit word and clock each bit on the target's DIN and
 * CCLK pins */
shift_word_out(data32)
{
   *cclk = CCLK_GPIO_BASEADDR
   *serial_out = SERIAL_OUT_GPIO_BASEADDR
   *cclk = 0
   *serial_out = 0
   for (i = 31; i >= 0; --i){
      *serial_out = (data32 & 1 << i) ?1 :0
      shift_cclk(1)
   }
}

/* Assert and Deassert CCLK */
shift_cclk(count)
{
   *cclk = CCLK_GPIO_BASEADDR
```

```
   *cclk = 0
   for (i = 0; i < count; --i) {
      *cclk = 1
      *cclk = 0
   }
}

int main()
{
   bits_start = BITSTREAM_START_ADDR
   bits_size = BITSTREAM_SIZE_BYTES

   *program_b = PROGRAM_B_GPIO_BASEADDR
   *init_b = INIT_B_GPIO_BASEADDR
   *done = DONE_GPIO_BASEADDR

   /* Configuration Reset */
   *program_b = 0
   sleep(TPROGRAM)
   *program_b = 1

   /* Wait for Device Initialization */
   while(*init_b == 0)
      ;

   /* Configuration (Bitstream) Load */
   for (i = 0; i < bits_size; i+=4) {
      shift_word_out(bits_start + i)
   }

   /* Check INIT_B */
   if (*init_b_pointer == 0) {
      return 1
   }

   /* Wait for DONE to assert */
   while(*done == 0)
      ;

   /* Compensate for Special Startup Conditions */
   shift_cclk(8)
   return 0
}
```

リファレンス デザイン ファイル

このアプリケーション ノートのリファレンス デザイン ファイルは、次のサイトからダウンロードできます。

https://secure.xilinx.com/webreg/clickthrough.do?cid=188189

表 5 にリファレンス デザインの詳細を示します。

表 5 : リファレンス デザインの詳細

パラメーター	説明
全般	
開発者	Matt Nielson

表 5 : リファレンス デザインの詳細 (続き)

パラメーター	説明
ターゲット デバイス (ステッピング レベル、ES、プロダクション、スピード グレード)	7 シリーズ FPGA
ソース コードの提供	Yes
ソース コードの形式	C
既存のザイリンクス アプリケーション ノート、リファレンス デザイン、CORE Generator ツール、サードパーティからデザインへのコードおよび IP の使用	No
シミュレーション	
機能シミュレーションの実施	No
タイミング シミュレーションの実施	No
機能およびタイミング シミュレーションでのテストベンチの利用	No
テストベンチの形式	N/A
使用したシミュレータ ソフトウェア/バージョン	N/A
SPICE/IBIS シミュレーションの実施	No
インプリメンテーション	
使用した合成ソフトウェア ツール/バージョン	N/A
使用したインプリメンテーション ソフトウェア ツール/バージョン	mb-gcc (GCC) 4.1.2 20070214 (Xilinx 13.4)
スタティック タイミング解析の実施	No
ハードウェア検証	
ハードウェア検証の実施	Yes
検証に使用したハードウェア プラットフォーム	SP605 ボード、XM105 デバッグ カード、ザイリンクス Kintex-7 FPGA テスト プラットフォーム

まとめ

このアプリケーション ノートでは、コンフィギュレーションの概要を示し、スレーブ SelectMAP またはスレーブ シリアル モードでザイリンクス FPGA をコンフィギュレーションする 2 つのリファレンス デザインについて説明しました。ここで提供しているマイクロプロセッサの C コードはザイリンクスの MicroBlaze プロセッサ向けのものですが、移植性を考慮して記述されています。コードを別のプロセッサに移植するには修正が必要になりますが、デザイン ファイルはすべて幅広く応用できるものです。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2012年05月31日	1.0	初版リリース

Notice of Disclaimer

Automotive Applications Disclaimer

本資料は英語版 (v1.0) 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。
資料によっては英語版の更新に対応していないものがあります。
日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。