



ALL PROGRAMMABLE™

XAPP1188 (v1.0) 2014 年 9 月 23 日

マイクロプロセッサを使用する場合の SPI フラッシュ メモリからの FPGA コンフィギュレーション

著者 : Simon Tam、Matt Nielson、Mattics Phi

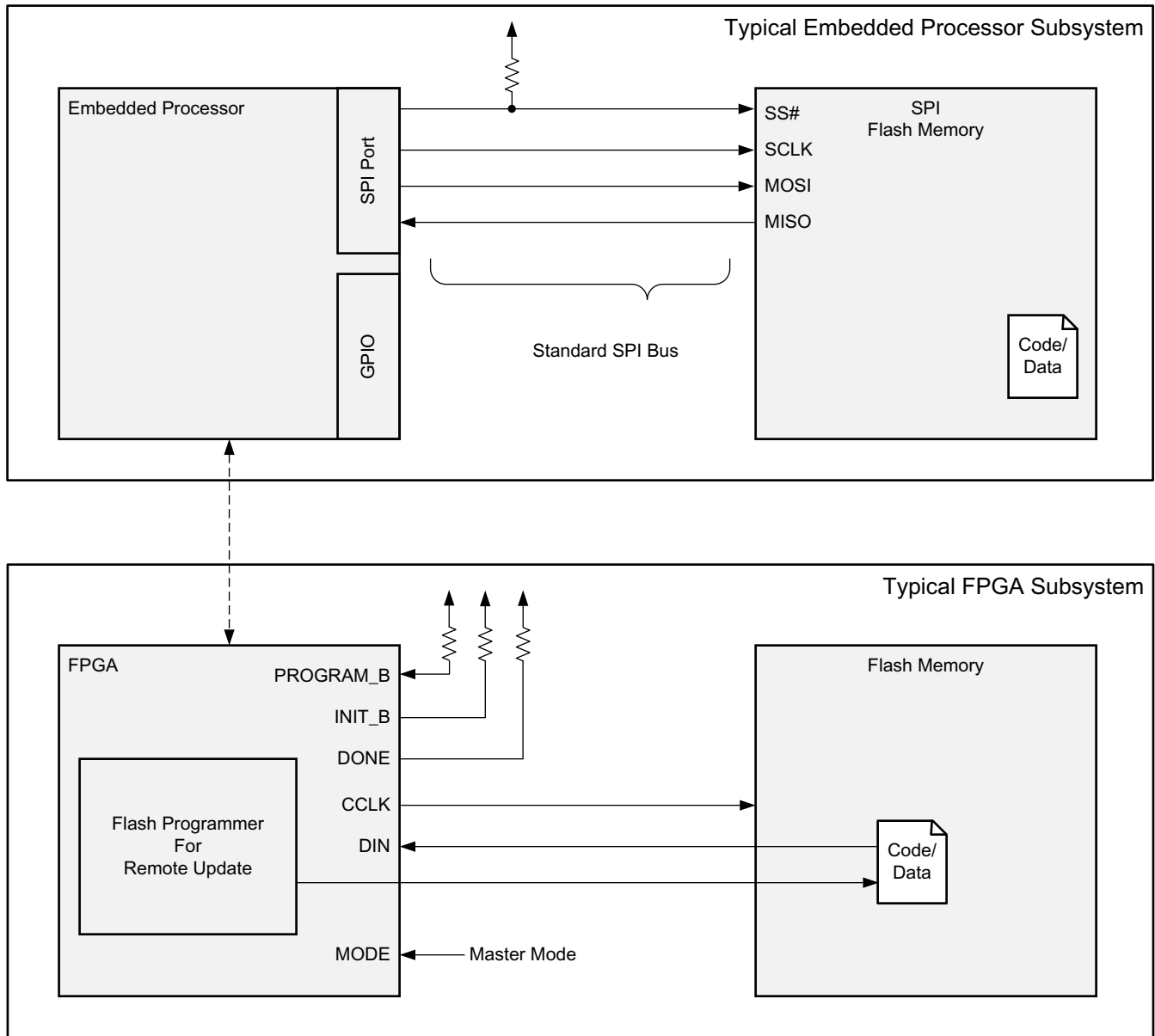
概要

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てシリアル ペリフェラル インターフェイス (SPI) フラッシュ メモリから FPGA デバイスをコンフィギュレーションするシンプルかつ効率的な方法について説明します。この方法によって、ハードウェア コンポーネント、ボード スペース、およびコストが削減できます。ここでは、リファレンス ハードウェア デザインとファームウェアを使用して説明します。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、ザイリンクスのウェブサイトからダウンロードできます。デザイン ファイルの詳細は、「[リファレンス デザイン](#)」を参照してください。

はじめに

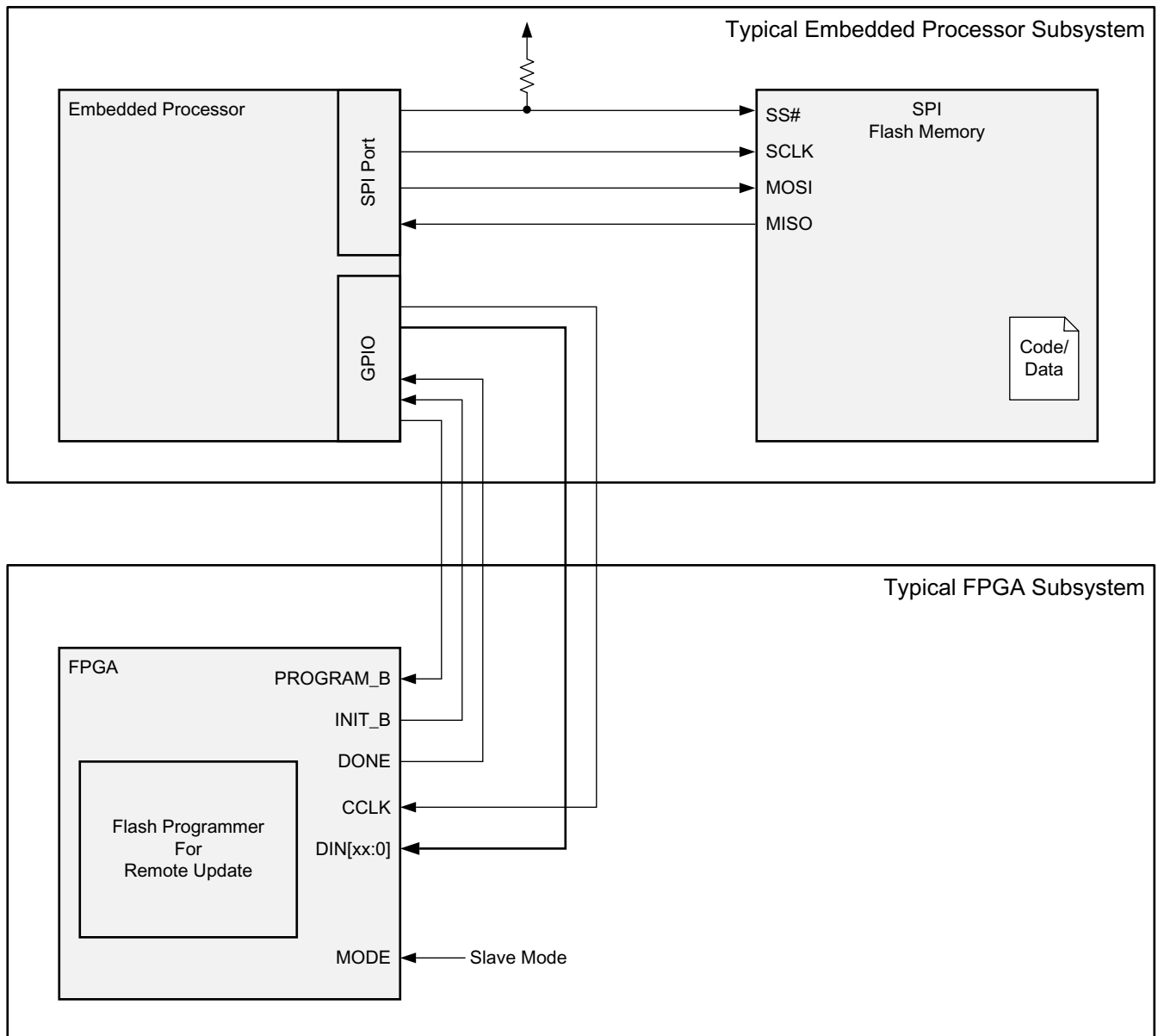
ディスクリート エンベデッド マイクロプロセッサと FPGA で構成されたシステムは一般的です。このようなシステムは、[図 1](#) のように 2 つの標準的なサブシステムから成ります。

マイクロプロセッサを利用して FPGA をコンフィギュレーションする推奨方法は、[XAPP583 \[参照 1\]](#) で説明しています。この方法では、マイクロプロセッサへ接続されたフラッシュ メモリ デバイスにユーザー ファームウェアとコンフィギュレーション ビット ファイルを格納します。マイクロプロセッサが SPI インターフェイスを介してビット ファイルを読み出し、スレーブ シリアル またはスレーブ SelectMAP インターフェイスを介して FPGA へビット ストリームを送信します。この場合、FPGA コンフィギュレーションのために追加で PROM を使用する必要がなくなります。[図 2](#) に、このブロック図を示します。



X1188_01_052814

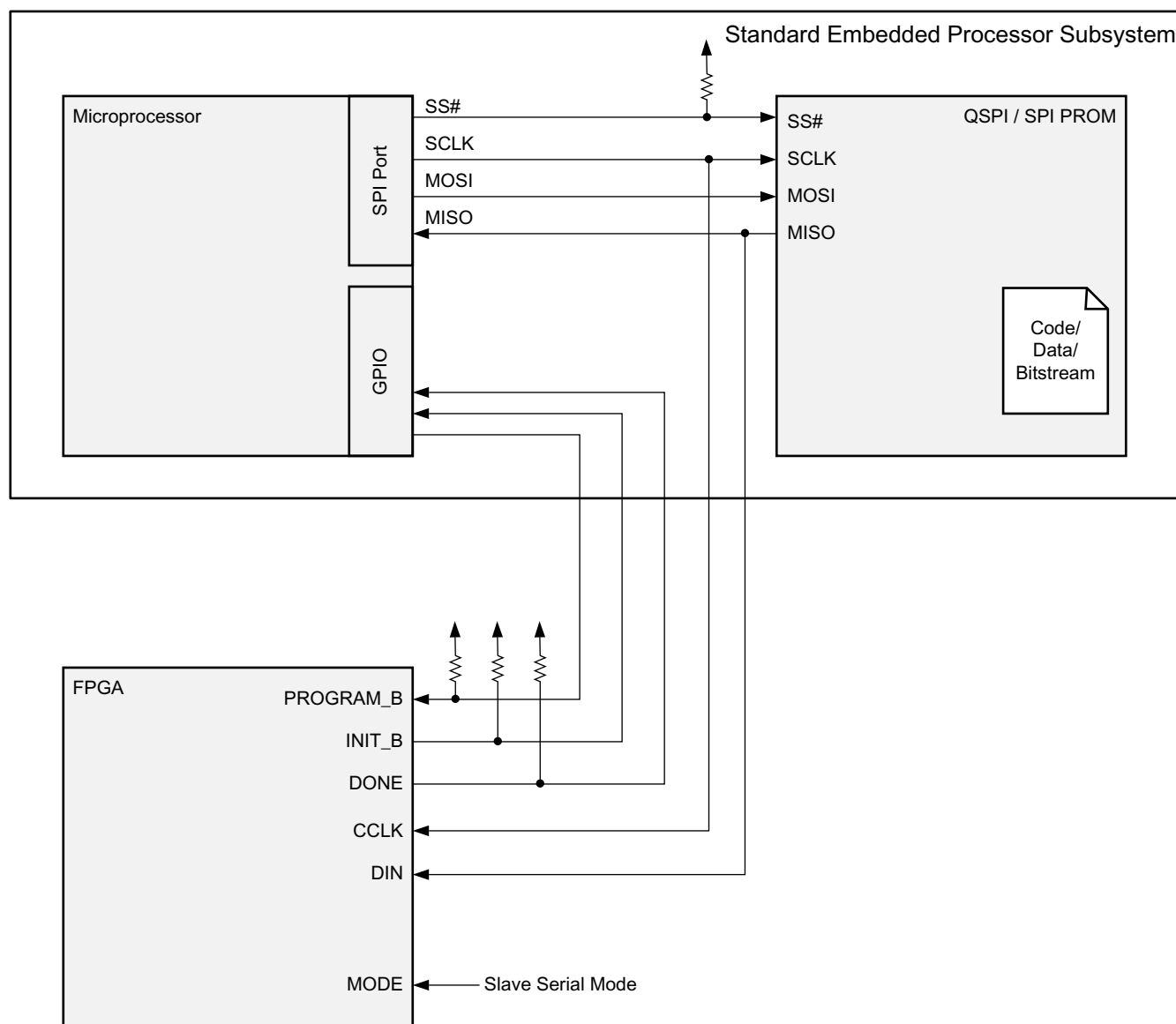
図 1 : FPGA を使用した一般的なエンベデッド マイクロプロセッサ システム



X1188_02_052814

図 2 : XAPP583 のブロック図

このアプリケーション ノートで説明する方法では、コンフィギュレーションがさらにシンプルになります。これは FPGA のスレーブ シリアル コンフィギュレーション ピンと標準の SPI バス信号の互換性を利用した方法です。図 3 に、マイクロプロセッサ、フラッシュ メモリ、および FPGA 間の接続を示します。後続のセクションでは、このデザインの詳細および動作について説明します。



X1188_03_061614

図 3 : XAPP1188 のブロック図

XAPP583 [参照 1] で説明されている方法と同様、マイクロプロセッサに接続されているフラッシュ メモリがユーザーファームウェアと FPGA ビットファイルを格納します。ただし、マイクロプロセッサは FPGA コンフィギュレーションポートを介して直接 FPGA をコンフィギュレーションしません。代わりに、FPGA のスレーブ シリアル DIN ピンと CCLK ピンを、フラッシュ メモリとマイクロプロセッサ間の SPI バスへ接続しています。スレーブ シリアル インターフェイスとコンフィギュレーションシーケンスが同時に SPI プロトコルと互換性を持つため、この方法でのコンフィギュレーションが可能になってます。この互換性の詳細は、「動作およびインプリメンテーションの詳細」で説明します。

ユーザー要件によって異なりますが、この方法はわずか2本の接続しか使用しません。コンフィギュレーションビットストリームをターゲット FPGA へ送信するために、次のピンを接続します。

- FPGA CCLK ピンを SPI バス SCLK ピンへ接続する。
- FPGA DIN ピンを SPI バス MISO ピンへ接続する。

コンフィギュレーションの制御またはモニタリング用に信号を追加できます。次の機能を有効にするには、各 FPGA ピンをマイクロプロセッサの GPIO ピンへ接続してください。

- PROGRAM_B – FPGA コンフィギュレーションシーケンスをリセットする。
- INIT_B – コンフィギュレーションの初期化またはエラー ステータスをチェックする。
- DONE – コンフィギュレーションの完了をモニターする。

このソリューションには、エンベデッド マイクロプロセッサとフラッシュを使用する図 1 に示す従来システムと比べて、次のようなメリットがあります。

- コンポーネント数の削減 - システムで必要なフラッシュ メモリが 1 つのみ。
- マイクロプロセッサの GPIO およびコード スペースの削減 - メモリ ソースからビットストリームを読み出して FPGA へ送信する動作を繰り返すエンベデッド マイクロプロセッサ ベースのコンフィギュレーションソリューションと比較した場合、コンフィギュレーション時間が短縮。
- マイクロプロセッサ ベースの FPGA コンフィギュレーション制御およびモニタリング機能 - ネットタイプの FPGA コンフィギュレーション機能より優れている。
- インシステム PROM のアップデート - 格納されたビットストリームのアップデートに、マイクロプロセッサで標準のフラッシュプログラミング ライブラリを使用できる。

このソリューションを説明するために、後続のセクションではすべてのマイクロプロセッサ タスクを実行するモデルとして、Zynq-7000 AP SoC のプロセッシング サブシステムを使用しています。

動作およびインプリメンテーションの詳細

このセクションでは、FPGA スレーブ シリアル コンフィギュレーション インターフェイスおよびシーケンスについて説明し、また Zynq-7000 AP SoC プロセッシング サブシステムを使用して FPGA をコンフィギュレーションする方法について解説します。Zynq-7000 AP SoC のプログラマブル ロジックはプロセッシング サブシステムからコンフィギュレーションする必要があるため、このソリューションでは Zynq-7000 AP SoC をターゲット デバイスとして使用しません。

FPGA スレーブ シリアル コンフィギュレーション モード

スレーブ シリアル モード (FPGA コンフィギュレーション モードの 1 つ) はシンプルな方法であり、非常に効率的といえます。主な特性を次に示します。このアプリケーション ノートで説明するコンフィギュレーション方法は、これらの特性を利用します。

- FPGA の内部コンフィギュレーション メモリのクリアとコンフィギュレーション モードのサンプリングが完了すると、FPGA はビットストリームをロードできる状態になり、DIN ピンをモニターします。
- コンフィギュレーション データ ビットは、CCLK の各立ち上がりエッジで DIN ピンに現れます。ただし、FPGA は、有効な同期ワード (0xAA995566) より前に受信したすべてのデータを破棄します。
- 同期ワードは、ビットストリームの開始点を示します。DIN ピンから有効な同期ワードを受信した後、FPGA はその後のビットストリームを有効なコンフィギュレーション データと見なします。
- FPGA は、スタートアップ コマンド (ビットストリームの終了直前) を受信すると、スタートアップ プロセスを開始します。このとき、FPGA はユーザー デザインを開始し、DIN ピンのモニタリングを停止します。

シリアル コンフィギュレーション ピンの詳細および電源投入後のコンフィギュレーション シーケンスは、『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) [参照 2] の「シリアル コンフィギュレーション データのクロッキング」および「コンフィギュレーション シーケンス」を参照してください。

シンプルなコンフィギュレーション方法

システムに電源が投入された後に FPGA を一度のみコンフィギュレーションする場合は、次のピンを接続するだけの非常にシンプルなコンフィギュレーションとなります。

- FPGA CCLK ピンを SPI バス SCLK ピンへ接続する。
- FPGA DIN ピンを SPI バス MISO ピンへ接続する。

Zynq-7000 AP SoC がコンフィギュレーションを開始する前、FPGA はコンフィギュレーション データを受信する準備が整っている必要があります。つまり、FPGA は、最初の電源投入後にコンフィギュレーションのハウス クリーニング ステージを完了している必要があります。FPGA をコンフィギュレーションする前のタイミング図は、UG470 [参照 2] の「デバイスの電源投入タイミング」を参照してください。

その後、Zynq-7000 AP SoC が SPI フラッシュ メモリへ読み出しコマンドを送信して、SPI フラッシュ メモリからすべてのビットストリームを読み出します。SPI フラッシュのシリアル読み出し動作では、ビットストリームが SPI バスの MISO ピンから FPGA の DIN ピンへシリアル送信されます。SPI フラッシュは、SCLK/CCLK の立ち上がりエッジで各シリアル データ ビットを出力します。FPGA は、SCLK/CCLK の次の立ち上がりエッジでシリアル データ ビットをキャプチャします。SPI フラッシュ メモリからすべてのビットストリームを読み出した時点で FPGA はコンフィギュレーションされた状態となります。

FPGA コンフィギュレーション インターフェイスの次の 3 つの特性によって、DIN ピンと CCLK ピンが SPI バスへ直接接続できるようになり、FPGA のコンフィギュレーションが可能になります。

- SPI バスのシリアル データ (MISO) およびクロック (SCLK) 信号は、FPGA のスレーブ シリアル データ (DIN) およびクロック (CCLK) ピンに対してそれぞれ互換性を持っています。
- FPGA コンフィギュレーション インターフェイスは、ビットストリーム読み出し動作の前に実行されるすべての SPI バス アクティビティを無視します。FPGA は、有効な 32 ビットの同期ワードを受信するまで全入力データを破棄するため、事実上すべての SPI バス アクティビティを無視します。
- FPGA コンフィギュレーション インターフェイスは、コンフィギュレーション動作が完了した後のすべての SPI バス アクティビティを無視します。FPGA は、コンフィギュレーション完了後に DIN 入力ピンのモニタリングを停止するため、ビットストリームの読み出し動作の終了後は事実上すべての SPI バス アクティビティを無視します。

図 4 に、SPI バスと FPGA コンフィギュレーション トランザクションの関係を示します。

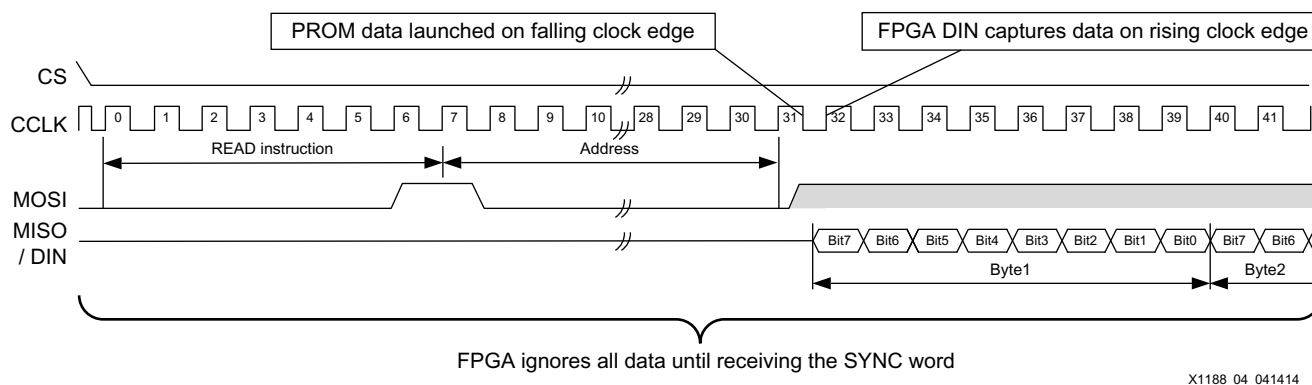


図 4 : SPI と FPGA のタイミング関係

追加で制御信号を使用するコンフィギュレーション方法

コンフィギュレーション動作および機能を強化するために、Zynq-7000 AP SoC を使用して FPGA コンフィギュレーション ピンを制御する方法もあります。たとえば、Zynq-7000 AP SoC は FPGA PROGRAM_B ピンを駆動する GPIO ピンを用いて FPGA のコンフィギュレーション メモリをクリアし、コンフィギュレーションを再開できます。これにより、システムの

動作状態に応じて、ユーザー デザインをリコンフィギュレーションできるようになります。この方法は、動作中に機能を変更する特殊なユーザー デザインに大きなメリットをもたらします。

さらに、GPIO ピンを FPGA の DONE ピンおよび INIT_B ピンに接続することで、Zynq-7000 AP SoC は FPGA コンフィギュレーションの準備状態や結果をモニターできます。INIT_B は、コンフィギュレーション エラーまたは準備が整ったことを示します。DONE 信号は、コンフィギュレーションの完了を示します。Zynq-7000 AP SoC は、これらのステータス信号による情報を効率的に活用し、信頼性の高いコンフィギュレーションを実行します。

たとえば、コンフィギュレーション エラーから回復するための一般的な方法として、コンフィギュレーション フォールバックがあります。フラッシュ メモリ内でターゲット ビットファイルが破損していると、ユーザー デザインをコンフィギュレーションできず、FPGA は機能しません。このような場合、ファームウェアが INIT_B 信号および DONE 信号を確認して状況を把握します。その後、フラッシュ メモリ内の信頼できる既知の(ゴールデン)ビットファイルを使用し、FPGA をリコンフィギュレーションして問題を解決します。これによって、FPGA は既知の動作状態に戻り、致命的なシステム エラーを回避できます。

SPI バスの SCLK 最大周波数

SPI バスのデータは、出力とキャプチャに逆のクロック エッジを使用します。SCLK のベース値が 0 (CPOL = 0) でクロック位相が 0 (CPHA = 0) の場合、データは立ち下がりエッジで出力されて立ち上がりエッジでキャプチャされます。したがって、SCLK 周波数の制限要因は SCLK/CCLK の最小 Low 時間です。これは、図 5 に示すように、SCLK の立ち下がりエッジから CCLK の立ち上がりエッジまでの、SPI フラッシュの MISO 出力データが有効な時間と FPGA DIN 入力セットアップ時間の合計で表すことができます。

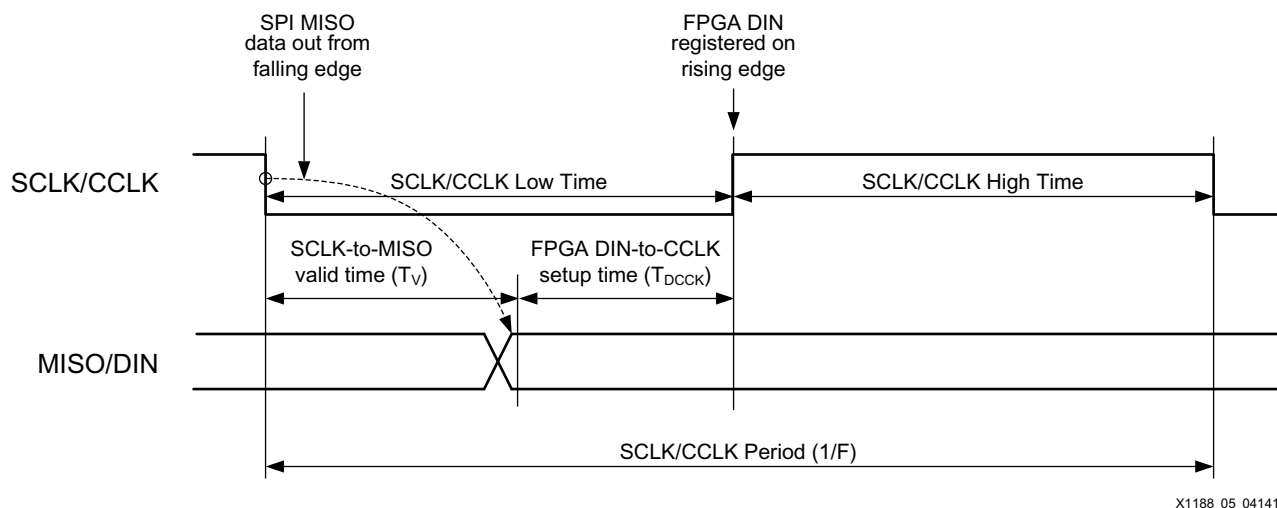


図 5 : SPI バスと FPGA のタイミング

次の式を使用して、FPGA コンフィギュレーションをサポートする SPI バスの最大 SCLK 周波数を概算します。

$$SCLK_{MaximumFrequency} \leq 1 / (T_V + T_{DCCK}) \times SCLK_{LowDutyCycle \% Minimum} \quad \text{式 1}$$

説明 :

- T_V = SPI フラッシュの SCLK から MISO データ出力が有効な時間
- T_{DCCK} = FPGA DIN 入力セットアップから CCLK セットアップまでの時間
- SCLK Low Duty Cycle % Minimum = クロック周期内における SCLK の最小パーセント

より正確な値を計算するには、SCLK/CCLK 信号および MISO/DIN 信号のトレースパスや伝搬遅延に伴うコンポーネントを上記の式に組み込む必要があります。

その他にも SCLK 信号 (FPGA CCLK ピン) のシグナル インテグリティが重要です。最大クロック周波数を達成するには、このクロック信号はベスト プラクティスを使用して設計し、Zynq-7000 AP SoC から SPI フラッシュ メモリおよび FPGA エンドポイントへ配線してください。

SCLK/CCLK のシグナル インテグリティ

SCLK 信号 (FPGA CCLK ピン) のシグナル インテグリティは非常に重要です。すべての SPI バス アクティビティは、SCLK の立ち上がりまたは立ち下がりエッジでトリガーされます。SPI フラッシュ メモリの SCLK ピンまたは FPGA の CCLK ピンにグリッジ (つまり、追加エッジ) をもたらす可能性がある信号トレース問題は回避する必要があります。このクロック 信号はベスト プラクティスを使用して設計し、Zynq-7000 AP SoC から SPI フラッシュ メモリおよび FPGA エンドポイントへ配線してください。たとえば、デュアルバッファを使用して SCLK の各デスティネーションにポイント間配線を行う、または適切な終端を用いてフライバイ配線を行います。各デスティネーション ピンに対して、個別のスタブまたは長いスタブを持つトレースは回避してください。

リファレンス デザイン

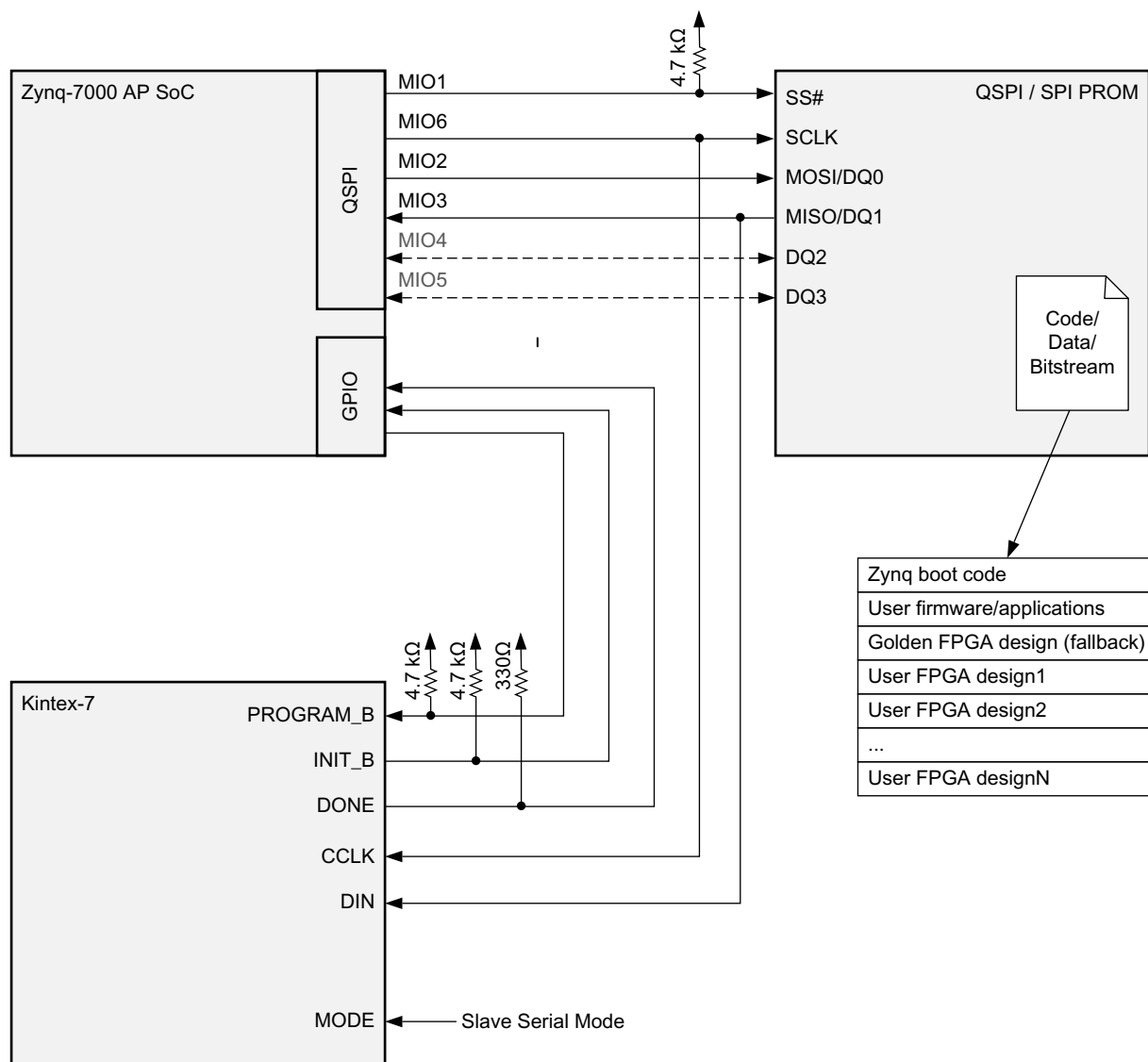
このアプリケーション ノートでは、デモンストレーション用のリファレンス デザインを提供しています。このデザインは、Zynq®-7000 All Programmable SoC (AP SoC) およびターゲット ボード上の FPGA を使用する、[図 2](#) のソリューションを実装します。このアプリケーション ノートの [リファレンス デザイン ファイル](#) は、ザイリンクスのウェブサイトからダウンロードできます。[表 1](#) に、リファレンス デザインの詳細を示します。

表 1: リファレンス デザインの詳細

パラメーター	説明
全般	
開発元	ザイリンクス
ターゲット デバイス	7 シリーズ FPGA UltraScale™ アーキテクチャ デバイス Zynq-7000 AP SoC
ソース コードの提供	あり
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/IP の使用	あり
シミュレーション	
論理シミュレーションの実施	なし
タイミングシミュレーションの実施	なし
論理シミュレーションおよびタイミング シミュレーションでのテストベンチの利用	なし
テストベンチの形式	N/A
使用したシミュレータ/バージョン	N/A
SPICE/IBIS シミュレーションの実施	N/A
インプリメンテーション	
使用した合成ツール/バージョン	Vivado® Design Suite 2013.4
使用したインプリメンテーション ツール/バージョン	Vivado® Design Suite 2013.4
スタティック タイミング解析の実施	なし
ハードウェア検証	
ハードウェア検証の実施	あり
使用したハードウェア プラットフォーム	Zynq-7000 AP SoC および Kintex®-7 FPGA 評価ボード

リファレンス ハードウェア

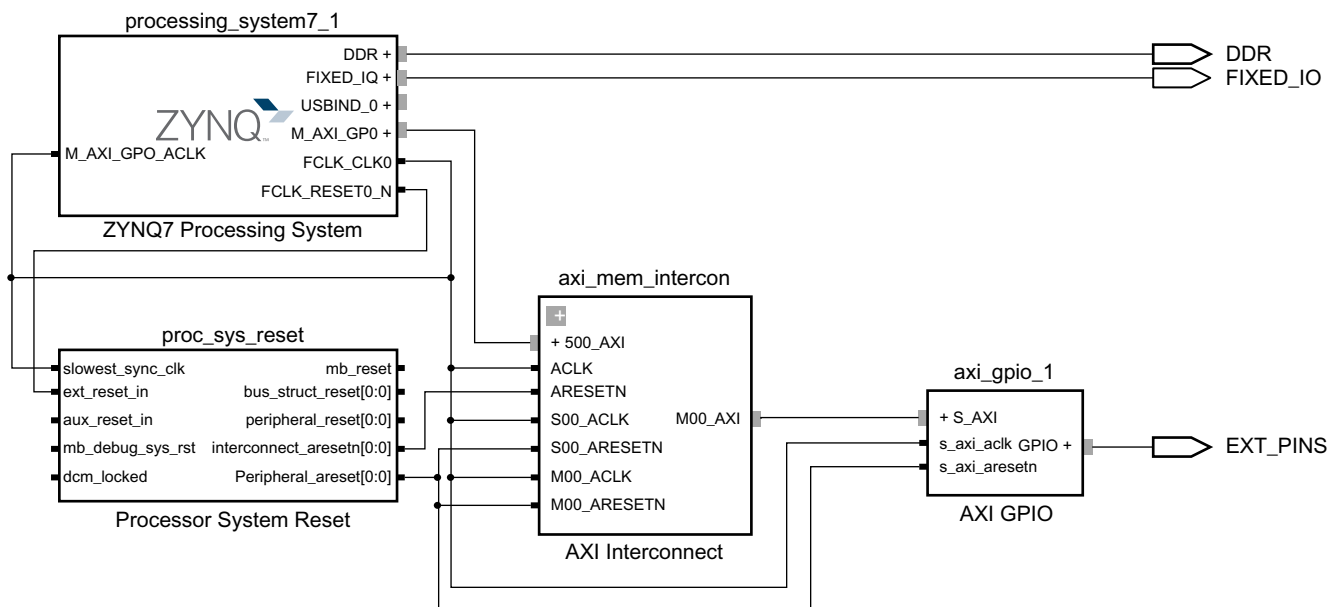
ハードウェア デザインは、Zynq-7000 AP SoC 7Z045 CLG484 デバイスをベースとしています。AP SoC Quad-SPI コントローラが、SPI x1、x2、または x4 幅をサポートする複数の I/O を備えたメモリ デバイス (例 : Micron/Numonyx 社製 N25Q128A13ESF40F) へ接続されます。Zynq Quad-SPI コントローラは、x1、x2、x4、および x8 幅をサポートしますが、このリファレンス デザインは、シングルビット バス幅を使用するレガシ SPI プロトコルで動作します。図 6 に、Zynq-7000 AP SoC、SPI PROM、および Kintex-7 FPGA 間のハードウェア接続を示します。



X1188_06_061714

図 6: リファレンス ハードウェアの接続

リファレンス ハードウェア デザインは、Vivado® Design Suite のプロジェクトとして提供されています。図 7 に、AP SoC 内のシステム ブロック図を示します。

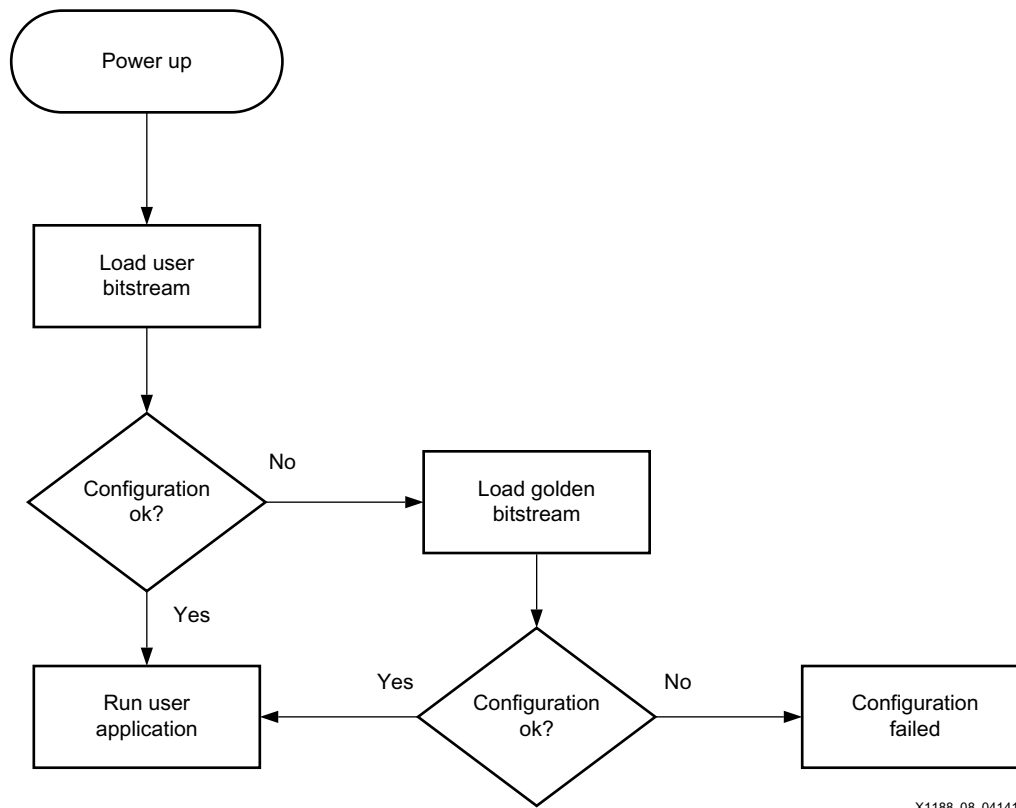


X1188_07_041414

図 7 : Vivado のブロック図

リファレンス ファームウェア

リファレンス デザインは、アップデートされた (最新のデザイン リビジョン) ビットストリームと SPI フラッシュ メモリ内に格納されているゴールデン (信頼性の高い既知の) ビットストリームを使用する Try-and-Fallback コンフィギュレーション スキームを実装しています。Zynq-7000 AP SoC ファームウェアは、まず最初に目的の最新のビットストリームをロードします。その後、コンフィギュレーションがエラーになると、コンフィギュレーションを再開してゴールデン ビットストリームをロードします。このプロセスは、フォールバック コンフィギュレーションとして知られています。図 8 に、ファームウェアのフロー図を示します。



X1188_08_041414

図 8: ファームウェアのフロー図

ファームウェア コンフィギュレーション アルゴリズムの詳細は次のとおりです。

- I/O モードを指定する Zynq-7000AP SoC Quad-SPI インターフェイス コンフィギュレーション レジスタを初期化します。
 - クロック分周器のセットアップ、チップ セレクト モード、およびコントローラーの有効化などのタスクを実行します。
- FPGA コンフィギュレーション メモリをクリアします。
 - PROGRAM_B 信号を少なくとも TPROGRAM より長い期間アサートし、その後ディアサートします。
- INIT_B がリリースされるまで待機します。
 - INIT_B ピンをポーリングし、FPGA がハウス クリーニング ステージを完了していることを確認します。
- Quad-SPI コントローラーの伝達関数を設定します。
 - デモでは、ポーリング データ伝達関数 XQspiPs_PolledTransfer() を使用します。関数プロトタイプは次のとおりです。

```
int XQspiPs_PolledTransfer(XQspiPs *InstancePtr, u8 *SendBufPtr, u8 *RecvBufPtr,
unsigned ByteCount);
```

- InstancePtr は、Quad-SPI インスタンスへのポインターです。
- SendBufPtr および RecvBufPtr は、送信および受信バッファ ポインターです。この関数には、FPGA をリコンフィギュレーションするビットストリームの長さと同じバイト数の送信バッファと受信バッファが必要です。
- ByteCount は、送信されるバイト数を示します。
- 表 2 に示すように、送信バッファの最初のワードには、コマンドとアドレスが含まれます。最初のバイトには、Quad-SPI メモリ デバイスのコマンドが含まれます。この場合、Read Byte コマンド (0x03) を送信します。すべての動作を完了するのに必要なコマンドは 1 つのみです。ターゲット ビットストリームの 24 ビットの開始アドレスは、3 つのバイトに分割されます。これは読み出し動作であるため、バッファの残りの内容は未定義です。

表 2: Quad-SPI 書き込みバッファの定義

バイト 0	バイト 1	バイト 2	バイト 3	残りのバイト
SPI コマンド	Address[23:16]	Address[15:8]	Address[7:0]	送信データ

- ポーリング伝達関数を呼び出す
 - 関数を呼び出した後、Quad-SPI コントローラーは送信バッファの内容を Quad-SPI メモリへ送信します。同時に、ビットストリームと同じ長さのバイト数を読み出します。FPGA の DIN および CCLK ピンは SPI コントローラーの MISO 信号と SCK 信号へ接続されているため、同期ワードが検出されると FPGA (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ている) はコンフィギュレーションを開始します。ビットストリームがすべて読み出されると、FPGA のコンフィギュレーションが完了します。
- エラーチェック
 - FPGA のコンフィギュレーション完了後、INIT_B および DONE がアサートされて読み出し動作にエラーがないことを示します。
 - 一方、INIT_B および DONE ピンのいずれかまたは両方が Low のままの場合は、ファームウェアが信頼性の高い既知のコンフィギュレーション (つまり、ゴールドンビットストリーム) をロードして FPGA をリコンフィギュレーションします。これは、コンフィギュレーションフォールバックとして知られています。

まとめ

このアプリケーション ノートでは、マイクロプロセッサ、フラッシュメモリ、および FPGA で構成される一般的なシステムのハードウェア/ファームウェア要件を削減する FPGA コンフィギュレーションソリューションについて説明しています。ここでは、FPGA のシリアルコンフィギュレーションモードと SPI メモリの互換性を利用しています。このソリューションはほとんどのマイクロプロセッサに適用できますが、アプリケーション ノートおよびリファレンス デザインでは、動作およびインプリメンテーションを詳しく示すために Zynq-7000 AP SoC のプロセッシング サブシステムを使用しています。

参考資料

1. 『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』([XAPP583](#))
2. 『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](#))
3. 『SPI フラッシュを使用した 7 シリーズ FPGA のコンフィギュレーション』([XAPP586](#))
4. 『Zynq-7000 All Programmable SoC (Z-7010、Z-7015、Z-7020) : DC 特性および AC スイッチ特性』([DS187](#))
5. 『Zynq-7000 All Programmable SoC (Z-7030、Z-7045、Z-7100) : DC 特性および AC スイッチ特性』([DS191](#))
6. 『Kintex-7 FPGA データシート : DC 特性および AC スイッチ特性』([DS182](#))

付録 A

デザインのトラブルシューティング ヒント

コンフィギュレーション エラーが生じた場合は、次のチェックリストに従います。

- ターゲット ボードに関して次の項目を確認します。
 - JTAG を使用して基本的な FPGA のコネクティビティおよび機能性をチェックします。

- ザイリンクスのプログラミング ツールおよび JTAG ケーブルを使用して、FPGA JTAG IDCODE を読み出します。
- ザイリンクスのプログラミング ツールおよび JTAG ケーブルを使用して、ビットストリームで FPGA をコンフィギュレーションします。
- JTAG コンフィギュレーション動作の完了時に、DONE ピンが High になっていることを確認します。
- 。 マイクロプロセッサが SPI フラッシュ メモリからビットストリームを読み出す前に、FPGA がコンフィギュレーションの準備ができていることを確認します。
 - デジタル オシロスコープを使用して、FPGA の PROGRAM_B 信号および INIT_B 信号をプローブします。電源投入後、または PROGRAM_B リセット プロセス中のこれらの信号をモニターします。これらの信号は、『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) [参照 2] の「デバイスの電源投入タイミング」で示すように遷移する必要があります。
- 。 マイクロプロセッサが SPI フラッシュ メモリからビットストリームを読み出す際の SPI フラッシュ データ ビット/バイトの順序を確認します。
 - デジタル オシロスコープまたはロジック アナライザを使用して、ビットストリームの最初から 256 ビットまでの間、FPGA の DIN (SPI MISO) 信号および FPGA の CCLK (SPI SCLK) 信号をモニターします。ビットストリームの同期 (SYNC) ワード (0xAA995566) は、ビットストリームの最初の 256 ビットの間に見れる必要があります。DIN ピンに到達する SYNC ワードのビット/バイト順序は、次のようになる必要があります (左側のビット値が最初)。

1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0

- 。 マイクロプロセッサが SPI フラッシュ メモリからビットストリームを読み出した後、FPGA のコンフィギュレーション ステータスを確認します。
 - ザイリンクスのプログラミング ツールおよび JTAG ケーブルを使用して、FPGA のコンフィギュレーション ステータスを読み出します。ステータスレポートを参照して、コンフィギュレーションの完了またはエラーを確認します。
 - ザイリンクスのプログラミング ツールおよび JTAG ケーブルを使用して、FPGA のコンフィギュレーション ステータスを検証します。
 - ザイリンクスのプログラミング ツールおよび JTAG ケーブルを使用して、FPGA のコンフィギュレーションをファイルヘリッドバックします。デバイスをコンフィギュレーションしたビットストリームまたはデバイスをコンフィギュレーションしなかったビットストリームのデータ パターンについて、ザイリンクス サポートを利用してファイルを確認します。
- 。 デジタル オシロスコープを使用し、FPGA CCLK ピンのできるだけ近くでプローブしてすべてのエッジに現れるグリッジを確認します。
- 。 回路図で次の項目を確認します。
 - 。 FPGA は、マイクロプロセッサおよび SPI フラッシュ メモリ SPI バスに対応する適切な I/O 電圧に設定されていることを確認します。
 - 。 SPI フラッシュ メモリ ピンが適切に終端処理されていることを確認します (write-protect#、hold#、および reset# ピンを含む)。
 - 。 SCLK 信号 (FPGA CCLK ピン) のシグナル インテグリティは非常に重要です。この信号はベスト プラクティスを使用して設計し、マイクロプロセッサから FPGA フラッシュ メモリおよび FPGA エンドポイントへ配線してください。すべての SPI バス アクティビティは、SCLK の立ち上がりまたは立ち下がりエッジでトリガーされます。SPI フラッシュ メモリの SCLK ピンまたは FPGA CCLK ピンでグリッジ (つまり、追加エッジ) をもたらす可能性がある問題は回避する必要があります。
 - 。 コンフィギュレーション開始前には PROGRAM_B 信号をパルスすることを推奨します。これによって、FPGA コンフィギュレーション ロジックが予期せぬステートへ遷移するなどの SPI バス アクティビティの潜在的問題を排除でき、FPGA ロジックはビットストリームを受信するための既知のステートに確実に遷移します (電源投入後の最初のコンフィギュレーションを含む)。
- 。 マイクロプロセッサのコンフィギュレーションまたはアプリケーション コードの次の項目を確認します。
 - 。 FPGA の PROGRAM_B 信号を制御あるいは FPGA の INIT_B および DONE 信号をモニターするために、GPIO が出力または入力として適切にコンフィギュレーションされていることを確認します。
 - 。 FPGA の PROGRAM_B リセット パルスが必要なタイミングを満たしているか、あるいは PROGRAM_B 信号が High ステートのままになっていることを確認します。

- 。コードが FPGA PROGRAM_B をパルスする場合、INIT_B が High に遷移するまで待機するか最小の TPL 時間待機することを確認します。TPL 時間の詳細は、FPGA のデータシートを参照してください。
- 。SPI バスの SCLK 周波数は、「SPI バスの SCLK 最大周波数」で計算される最大周波数より低速で動作するように設定されていることを確認します。
- 。SPI フラッシュ メモリの読み出しコマンドが、SPI フラッシュ メモリ内のビットストリーム位置の適切な開始アドレスと対になっていることを確認します。
- 。ビットストリームの読み出しを実行する SPI コマンドが、ビットストリームの全ビットを読み出すことを確認します。ビットストリームの長さは、『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) [参照 2] の表「ビットストリームの長さ」を参照してください。

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
2014 年 9 月 23 日	1.0	初版

法的通知

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.

© Copyright 2014 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

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、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。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。