



XAPP1257 (v1.0) 2015 年 9 月 30 日

UltraScale FPGA の SPI フラッシュを使用するマルチブートおよびフォールバック

著者 : Wendy Curran

概要

このアプリケーション ノートでは、UltraScale™ FPGA の SPI (シリアルペリフェラル インターフェイス) コンフィギュレーション モードで使用するマルチブート デザインを構築する場合の主な概念について説明します。UltraScale FPGA のマルチブート機能は、FPGA アプリケーションの制御下で複数の FPGA ビットストリームの読み込みを可能にします。ここでは、ビットストリーム設定を使用してフォールバック機能をインプリメントする手順、フォールバックを開始するさまざまな方法、さらにはブート ステータス (BOOTSTS) レジスタを使用してフォールバック動作をデバッグおよび検証する方法について説明します。このアプリケーション ノートには、SPI モードを使用する UltraScale FPGA のフォールバック機能を検証するためのリファレンス デザインが含まれています。

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

UltraScale FPGA のマルチブートおよびフォールバックは、フィールドでのシステム アップグレードをサポートする機能です。フィールドでビットストリーム イメージをアップグレードできるため、設計者には非常に大きなメリットとなります。FPGA のマルチブート機能は、オンザフライでイメージを切り替えることができます。マルチブート コンフィギュレーション プロセス中にエラーが検出されると、FPGA はフォールバック機能を開始して、検証済みのデザイン (ゴールデン イメージ) をデバイスに読み込むことができます。

この資料では、SPI (x1/x2/x4) コンフィギュレーション インターフェイスを使用する UltraScale FPGA のマルチブートおよびフォールバック機能について説明します。このアプリケーションでは、SPI x4 コンフィギュレーション モードを使用するザイリンクスの KCU105 開発ボード上で Micron 社製 N25Q256 シリアル NOR フラッシュ メモリ デバイスを使用しています。SPI x4 コンフィギュレーション インターフェイスの詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [\[参照 1\]](#) を参照してください。

マルチブートおよびフォールバックの基本

UltraScale アーキテクチャは、SPI x1、x2、および x4 でのマルチブートをサポートしており、FPGA は複数のビットストリームが格納されている SPI フラッシュ デバイスからビットストリームを読み込むことができます。このモードの場合、FPGA アプリケーションがマルチブート動作を開始して、別のビットストリームで FPGA をリコンフィギュレーションします。マルチブート動作が開始されると、FPGA は通常どおりにコンフィギュレーションプロセスを再開し、専用のマルチブート ロジック、ウォーム ブート開始アドレス (WBSTAR) レジスタ、および BOOTSTS レジスタ以外のコンフィギュレーション メモリを消去します。その後、SPI フラッシュ デバイスからの新しいビットストリームでリコンフィギュレーションを実行します。

フォールバックが開始される条件

コンフィギュレーション中に次のエラーが生じるとフォールバックが開始されます。

- IDCODE エラー
- 巡回冗長検査 (CRC) エラー
- ウォッチドッグ タイマーのタイムアウト エラー

フォールバックは、ビットストリーム オプションの `ConfigFallback` でも有効にできます。フォールバック リコンフィギュレーション中は、ウォッチドッグ タイマーが無効になります。フォールバック リコンフィギュレーションでエラーが発生すると、コンフィギュレーションが停止し、INIT_B と DONE の両方が Low に維持されます。

ゴールデン イメージ

FPGA デバイスに電源が投入されると、アドレス ロケーション 0 からゴールデン イメージが読み込まれます (図 1)。ゴールデン イメージには、WBSTAR (`next_config_addr`) レジスタで指定される上位アドレス空間があります。WBSTAR にデフォルト以外のアドレス値が設定されている場合は、IPROG が自動的にビットストリームに組み込まれます。電源が投入されると、ゴールデン イメージの読み込みが開始され、この上位アドレス空間からマルチブート イメージのブートが開始されます。マルチブート イメージは複数用意しておくことが可能で、任意のデザインからほかのイメージの読み込みを開始できます。マルチブート イメージのブート中にエラーが発生し、コンフィギュレーションを完了できなかった場合は、フォールバック回路がアドレス 0 からゴールデン イメージの読み込みを開始します。

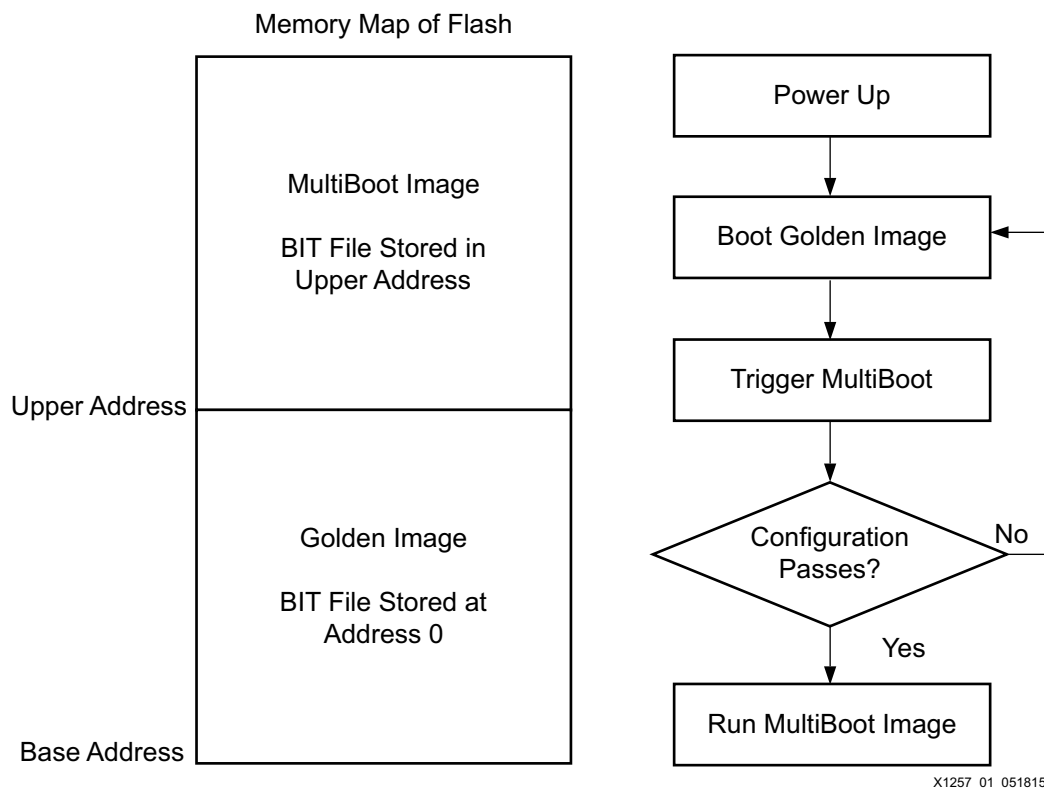


図 1: マルチブートおよびフォールバックのフロー

マルチブート イメージ

電源が投入されると、上位アドレス空間に格納されているマルチブート イメージが最初に読み込まれます。このイメージのコンフィギュレーションが正常に終了しなかった場合は、フォールバックが自動的に開始され、アドレス 0 に格納されているゴールデン イメージが読み込まれます。フォールバック機能は、マルチブート イメージの読み込みが正常終了しなかった場合にシステムを回復し、ゴールデン イメージの読み込みを可能にします。この段階でゴールデン イメージを読み込むことで、フラッシュ内のエラーを修正でき、マルチブート イメージから別のコンフィギュレーションを開始できます。

マルチブート フォールバック リファレンス デザイン

このセクションでは、マルチブート フォールバック リファレンス デザインの生成および検証方法について説明します。

マルチブート デザインの生成

マルチブート リファレンス デザインのビットストリーム設定

表 1 に、マルチブート デザインに必要なビットストリーム設定を示します。この表には、デフォルト値、設計可能な値、およびリファレンス デザインで使用されるオプションが含まれています。

表 1: ビットストリーム設定

設定	デフォルト値	設定可能な値	ゴールデンイメージ	マルチブートイメージ	説明
BITSTREAM.CONFIG.NEXT_CONFIG_ADDR	None	<文字列>	0x0400000	N/A	マルチブート セットアップで、次のコンフィギュレーションの開始アドレスを設定する。
BITSTREAM.CONFIG.NEXT_CONFIG_REBOOT	Enable	Enable、Disable	Enable	N/A	Disable に設定した場合、IPROG コマンドがビットファイルから削除される。
BITSTREAM.CONFIG.SPI_32BIT_ADDR	No	No、Yes	Yes	Yes	SPI 32 ビット アドレス形式を有効にする。256Mb またはそれより大規模なストレージを持つ SPI デバイスの場合に必要。
BITSTREAM.CONFIG.SPI_BUSWIDTH	None	NONE、1、2、4	4	4	サードパーティの SPI フラッシュ デバイスからのマスター SPI コンフィギュレーションの場合には、SPI バスをデュアル (x2) またはクワッド (x4) モードに設定する。
BITSTREAM.CONFIG.CONFIGFALLBACK	Enable	Enable、Disable	N/A	Enable	コンフィギュレーションが正常に完了しなかった場合、デフォルトのビットストリームの読み込みを有効または無効にする。
BITSTREAM.GENERAL.COMPRESS	False	True、False			複数のフレームをビットストリームに一度に書き込む機能を使用して、ビットストリーム (.bit) ファイルだけでなく、ビットストリームのサイズを小さくする。Compress オプションを使用しても、ビットストリーム サイズが小さくなるとは限らない。

注記：その他に利用できるビットストリーム オプションについては、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) [参照 2] を参照してください。

ビットストリームの設定は、Tcl コンソールから設定できますが、XDC ファイルで直接設定することも可能です。このリファレンス デザインに特有の設定については、後続のセクションを参照してください。

XDC ファイルを使用して設定

このセクションでは、XCD ファイルを使用した場合のビットストリームの設定を示します。

ゴールデン イメージ

```
set_property BITSTREAM.CONFIG.NEXT_CONFIG_ADDR 0x0400000 [current_design]
set_property BITSTREAM.CONFIG.NEXT_CONFIG_REBOOT ENABLE [current_design]
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
```

マルチブート イメージ

```
set_property BITSTREAM.CONFIG.CONFIGFALLBACK ENABLE [current_design]
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
```

注記:

- ターゲット SPI デバイスは、Micron 社製 N25Q256 シリアル NOR フラッシュ デバイスであるため、BITSTREAM.CONFIG.SPI_32BIT_ADDR が設定されています。
- BITSTREAM.GENERAL.COMPRESS はオプションですが、コンフィギュレーション時間を短縮する目的で使用されています。

write_cfgmem を使用して SPI フラッシュ ファイルを生成

コンフィギュレーション メモリ デバイスをプログラムするには、MCS ファイルが必要です。Vivado® Design Suite で MCS ファイルを生成する場合は、write_cfgmem Tcl コマンドを使用します。次に例を示します。

```
write_cfgmem -force -format MCS -size 32 -interface SPIx4 -loadbit "up 0x00000000
Golden.bit up 0x00400000 Update.bit" KCU105_multiboot_spix4.mcs
```

write_cfgmem コマンドの詳細およびすべての有効なオプションについては、『Vivado Design Suite ユーザー ガイド : プログラムおよびデバッグ』(UG908) [参照 2] および『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) [参照 3] を参照するか、Tcl コンソールで write_cfgmem コマンドのヘルプを実行してください。

ハードウェアでデザインを検証

このセクションでは、ハードウェアでマルチブート フォールバック リファレンス デザインを検証する方法を説明します。

ハードウェア要件

- KCU105 評価ボード
- KCU105 Diligent USB-to-JTAG モジュールへプラグインするための USB A-to-micro-B ケーブルまたはザイリンクスのプラットフォーム ケーブル USB II

ソフトウェア要件

- Vivado Design Suite 2015.1

予想されるイメージの動作

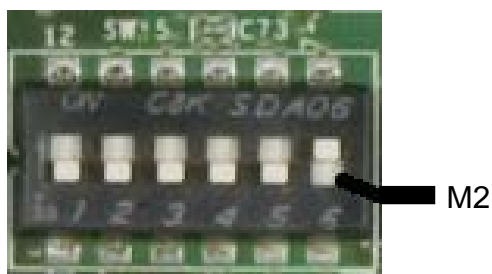
リファレンス デザインにはゴールデン イメージとアップデート イメージが含まれており (BIT ファイルと MCS ファイルの両方を提供)、これらを使用して各イメージの動作を検証できます。

- ゴールデン イメージは、GPIO LED [3:0] を左から右 (ボード上の 3 から 0) へ順番に点灯します。

- アップデート イメージは、GPIO LED [3:0] を右から左 (ボード上の 0 から 3) へ順番に点灯します。

ボードのセットアップ

SPI コンフィギュレーションを正常に実行するには、SW15 が正しい SPI コンフィギュレーション モードを示しているかを確認する必要があります。FPGA モード ピン M1 および M0 は、それぞれロジック 0 およびロジック 1 へ接続するように配線されます。FPGA モード ピン M2 は SW15 の位置 6 に接続され、M2 ネットがロジック 0 にプルダウンされて Quad SPI (QSPI) モードの選択が可能になります (図 2)。SW15.6 は、デュアル QSPI フラッシュにプログラムされたビットストリームを使用して、UltraScale FPGA のコンフィギュレーションを実行します。



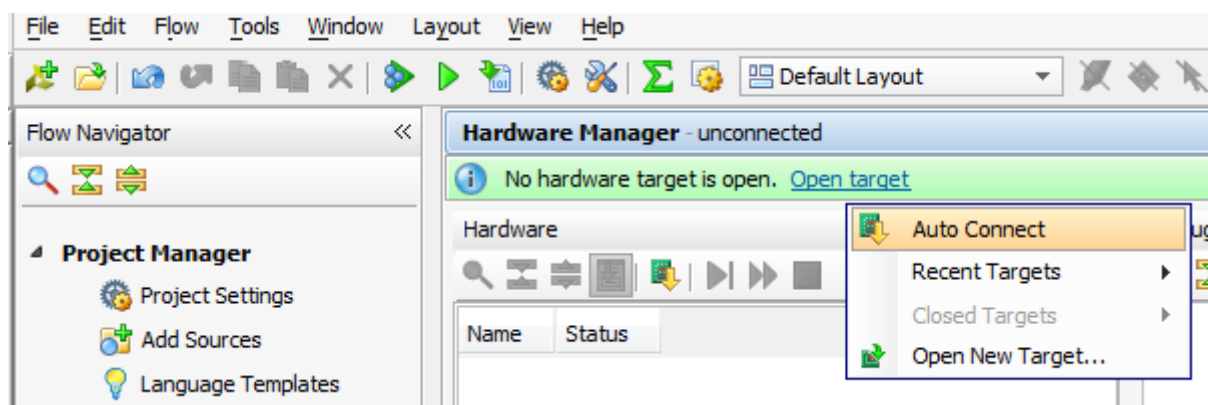
X1257_03_051815

図 2 : KCU105 SW15

フラッシュのプログラム

フラッシュ デバイスをプログラムする場合、まず最初に Vivado IDE (統合設計環境) でハードウェア ターゲットへの接続が必要になります。次の手順にしたがって、Vivado IDE でハードウェア ターゲットへ接続します。

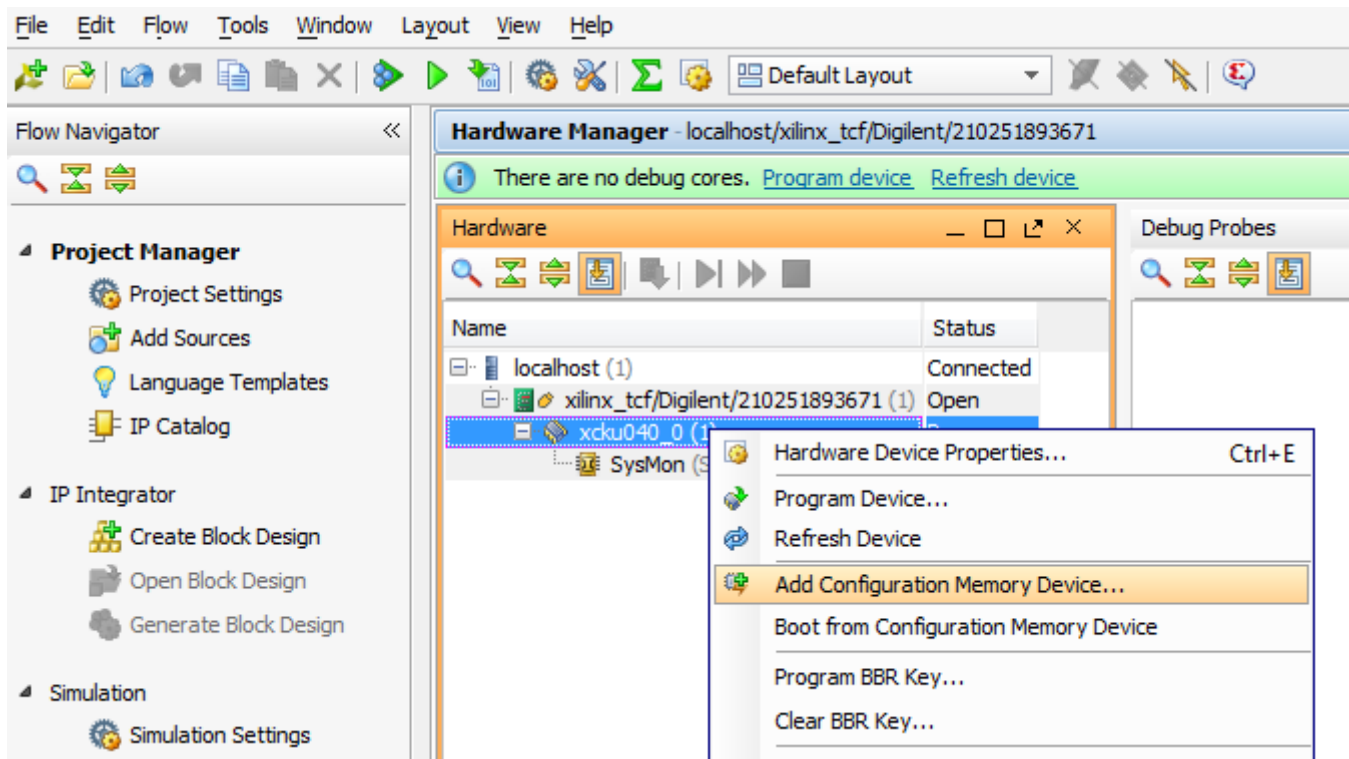
1. Vivado IDE の左側にある Flow Navigator の [Program and Debug] セクションで [Open Hardware Manager] をクリックします。
2. Hardware Manager が表示され、[Open Target] をクリックします。
3. [Auto Connect] をクリックして、デフォルト設定を使用してボードへの自動接続を実行します (図 3)。



X1257_04_051815

図 3 : Hardware Manager でボードへ自動接続

4. Hardware Manager コンソールに FPGA が表示されると、その FPGA を右クリックして [Add Configuration Memory Device] をクリックします (図 4)。



X1257_05_051815

図 4 : Vivado IDE でコンフィギュレーション メモリ デバイス オプションを追加

- このリファレンス デザイン用のデバイス (例 : N25Q256-1.8v-spi-x1_x2_x4) をクリックして [OK] をクリックします (図 5)。

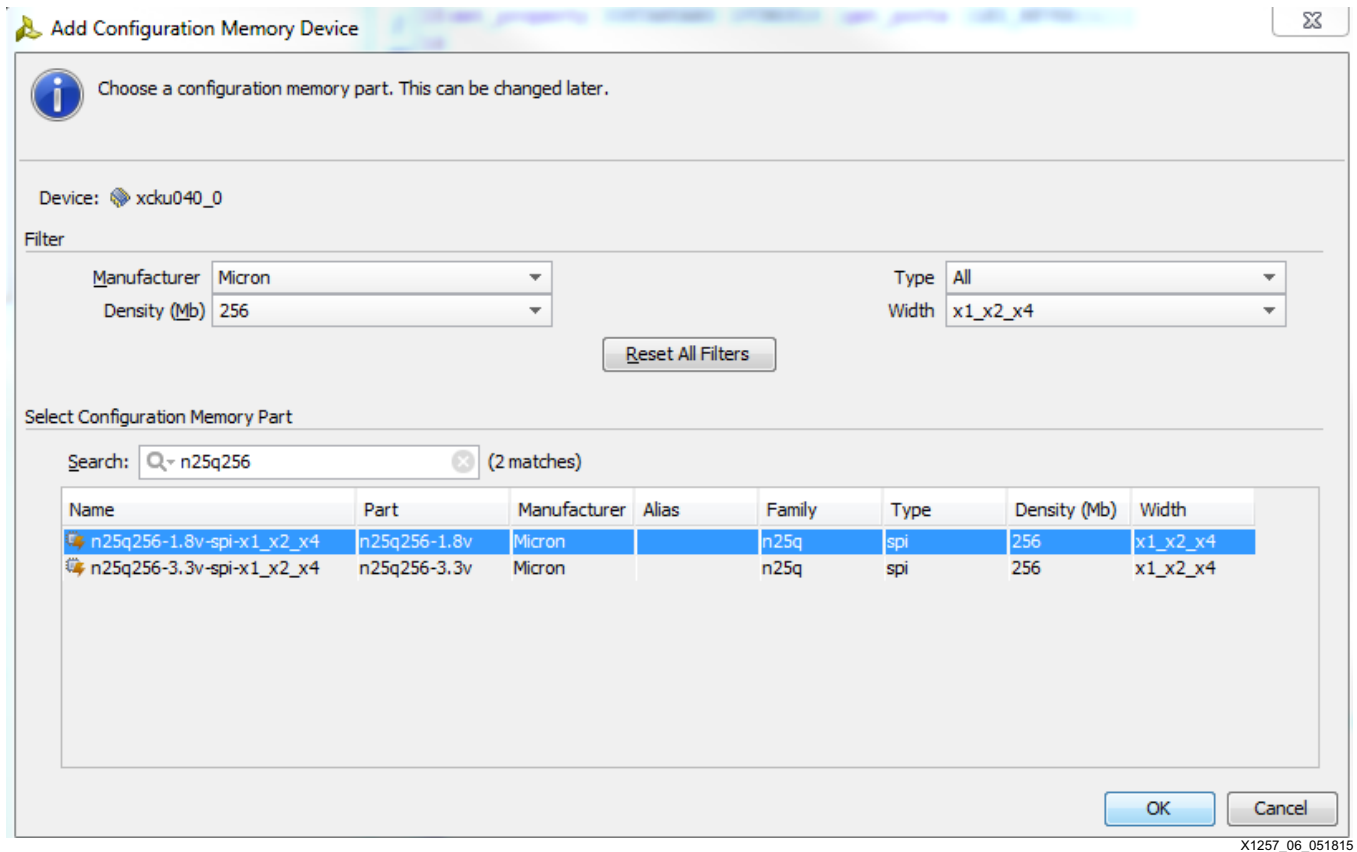
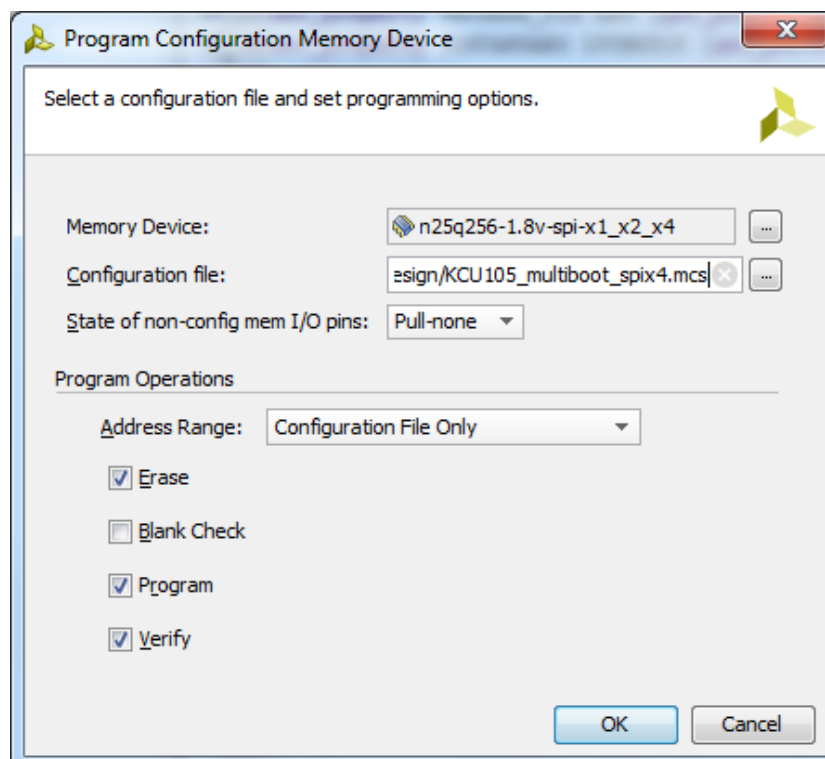


図 5 : Vivado IDE でコンフィギュレーション メモリ デバイスを追加

- コンフィギュレーション メモリ デバイスのプログラムを求めるプロンプトが表示されたら [OK] をクリックします。

7. 以前に生成された MCS ファイル (.../ready_to_download/KCU105_multiboot_spix4.mcs) を選択し、[OK] をクリックしてプログラムを開始します (図 6)。

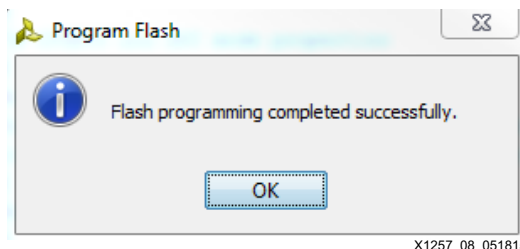


X1257_07_051815

図 6: Vivado IDE でコンフィギュレーション ファイルを追加

マルチブート動作の検証

フラッシュ デバイスのプログラムが正常に完了すると、Vivado IDE に図 7 と同じウィンドウが表示されます。

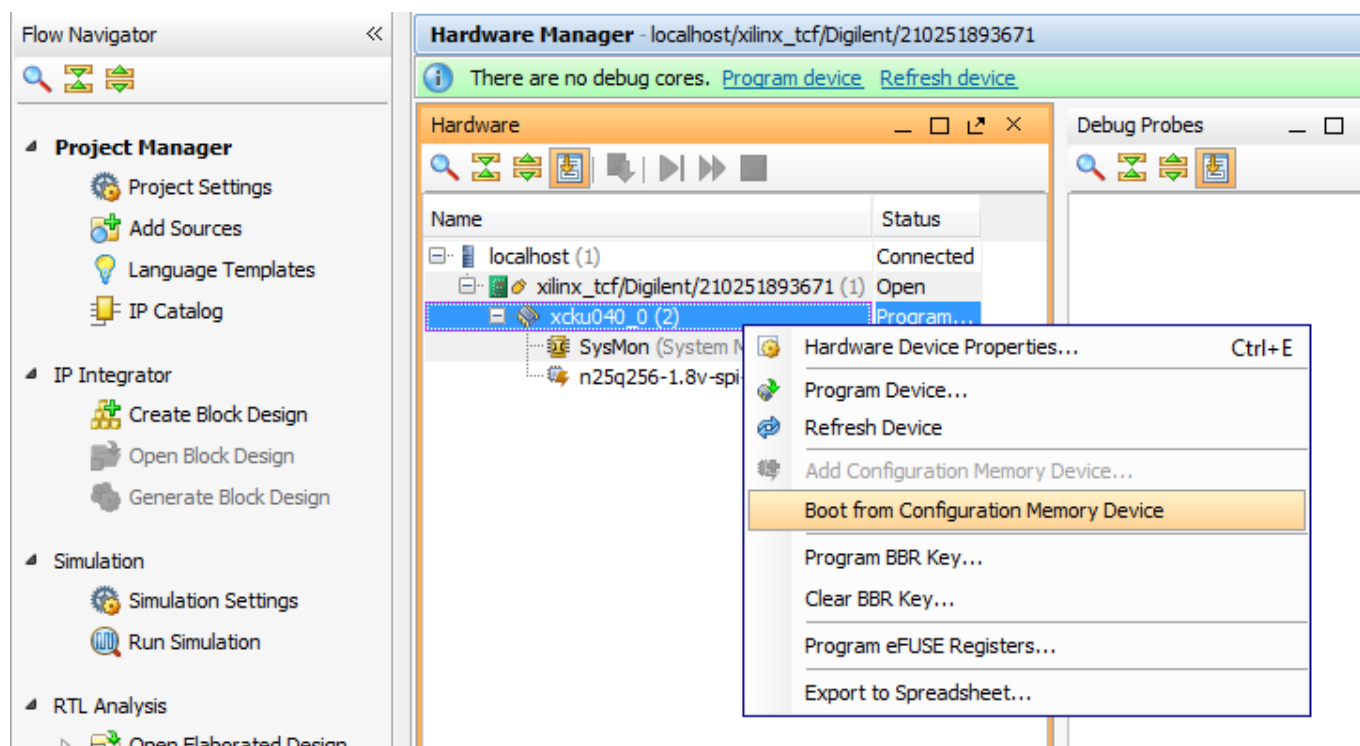


X1257_08_051815

図 7: フラッシュのプログラミング完了を示すウィンドウ

次のいずれかの方法で、フラッシュ デバイスにプログラムされたイメージを使用して FPGA を起動します。

1. KCU105 ボードの SW4 をパルスして、PROGRAM_B をパルスします。
2. boot_hw_device TCL コマンド (boot_hw_device [lindex [get_hw_devices] 0]) を使用します。
3. Vivado IDE でデバイスを右クリックして [Boot from Configuration Memory Device] をクリックします (図 8)

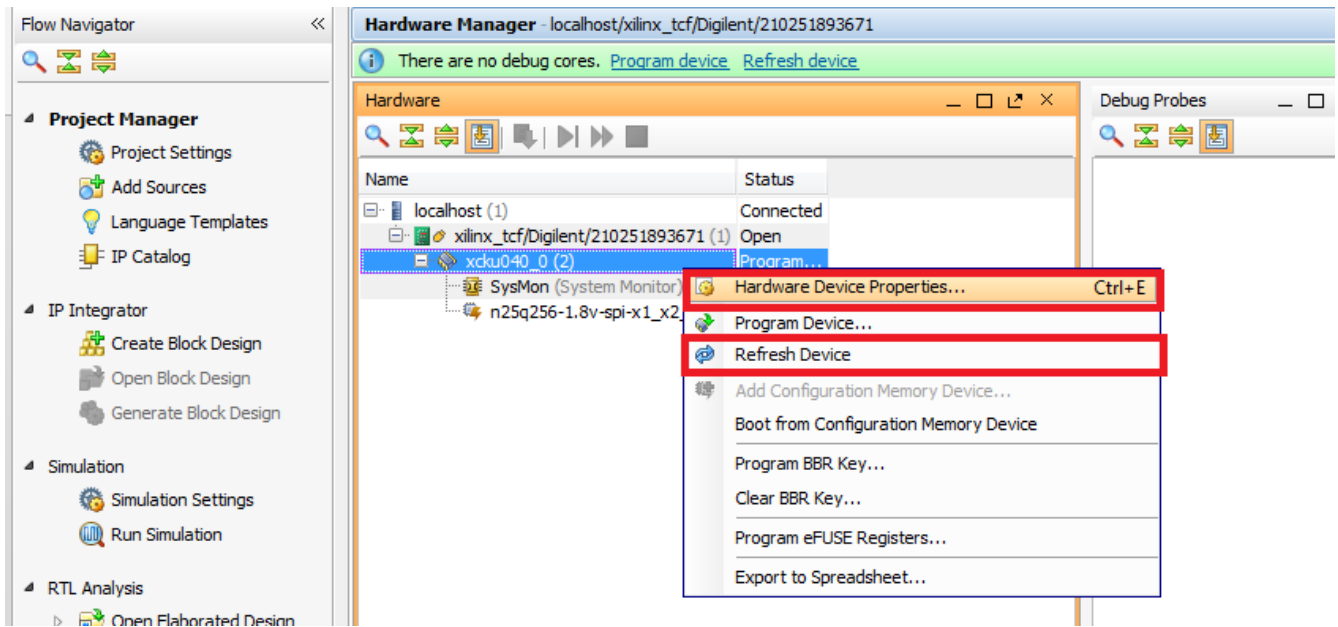


X1257_09_051815

図 8 : Vivado Hardware Manager の [Boot from Configuration Memory Device] オプション

次の方法で、FPGA が SPI フラッシュのアップデート ビットストリームで正常にコンフィギュレーションされているかを検証します。

- ボード上の DONE ピン LED が点灯しているかを確認します。
- GPIO LEDs [3:0] が右から左へ順番に点灯し、アップデート ビットストリームが正常に読み込まれたことを示しているかを確認します。
- Vivado IDE で FPGA を右クリックして [Hardware Device Properties] をクリックし、デバイスをリフレッシュします (図 9)。



X1257_10_051815

図 9 : Vivado Hardware Manager の [Hardware Device Properties] および [Refresh Device] オプション

- Vivado IDE の [Properties] ウィンドウで、[REGISTER] の下にある [BOOT_STATUS] と [CONFIG_STATUS] を展開表示します。BOOT_STATUS レジスタでは、アップデート ビットストリームへのジャンプをもたらした IPROG (INTERNAL_PROG) フラグが High であることを確認します。CONFIG_STATUS レジスタでは、DONE_PIN が High を示しています (図 10)。

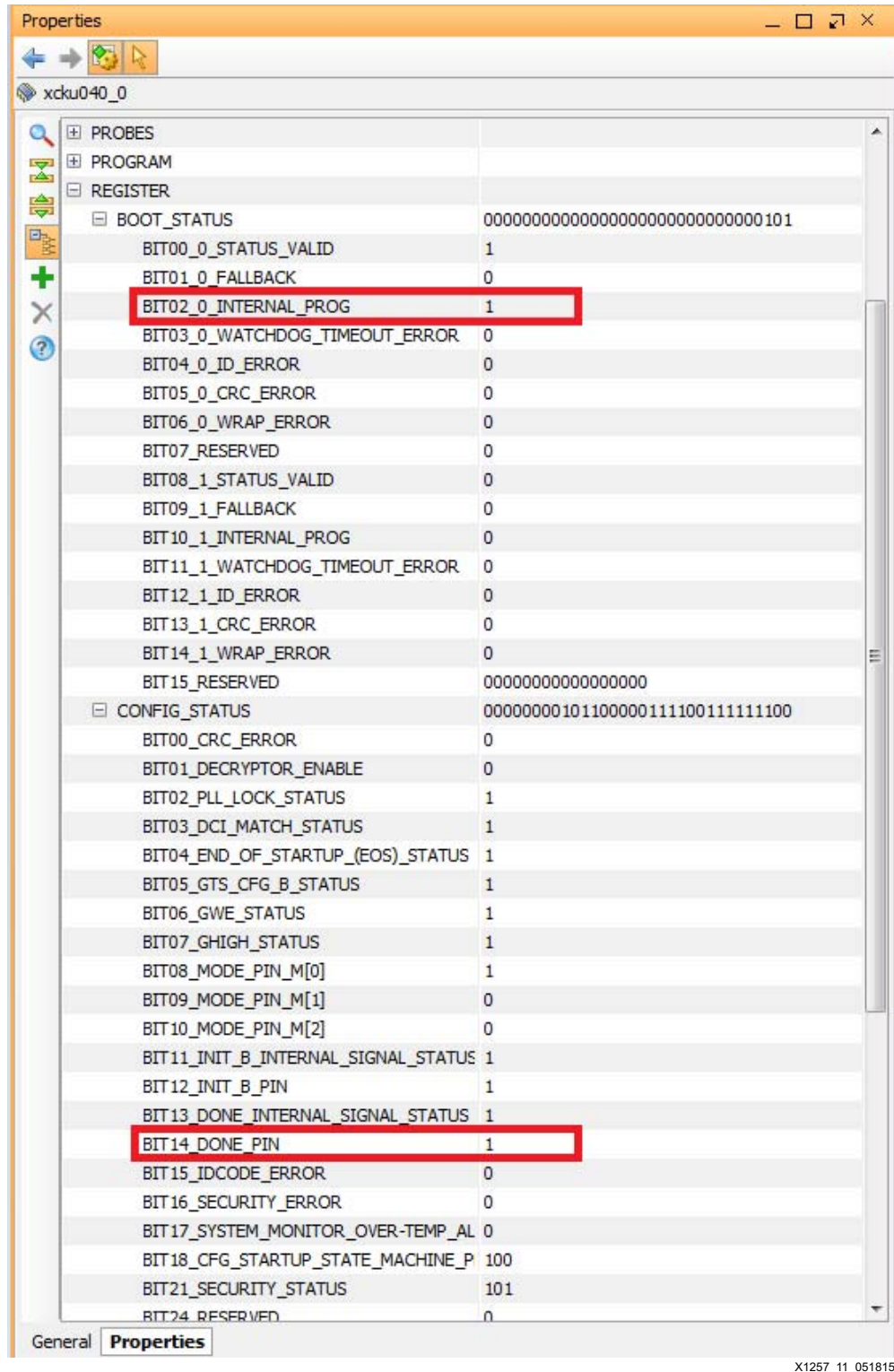
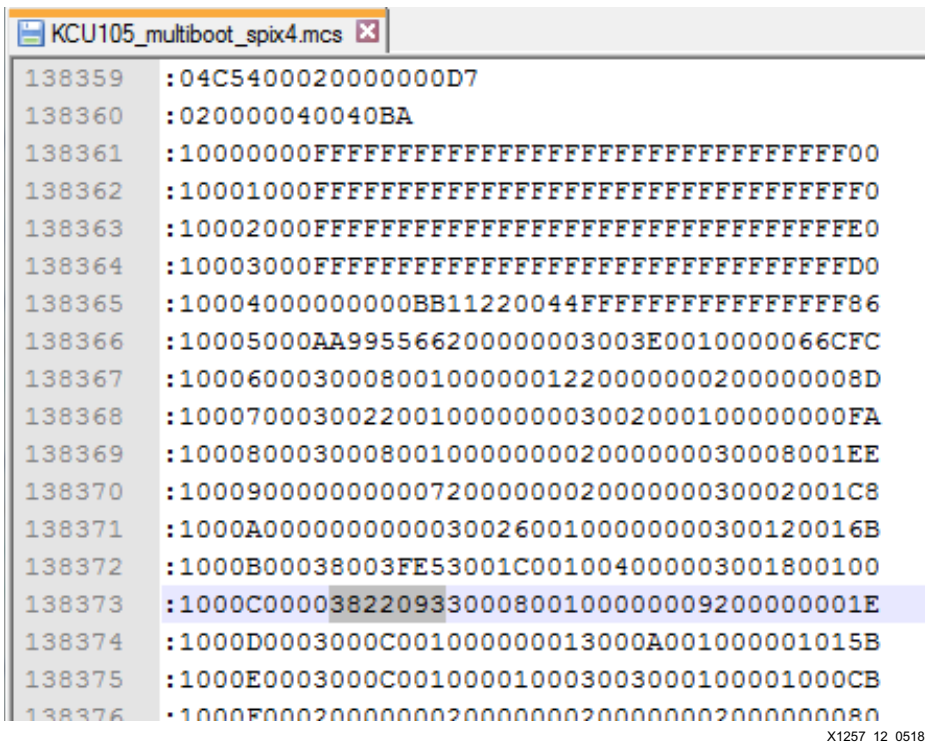


図 10: IPROG および DONE_PIN が High であることを示す BOOT_STATUS

フォールバックの例 - IDCODE エラー

フォールバックが正常に機能していることを確認する有効な方法として、アップデート イメージの IDCODE を故意に変更する方法があります。『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 1] から、XCKU040 デバイスの IDCODE は X3822093 であることがわかります。

- リファレンス デザインの KCU105_multiboot_spix4.mcs ファイルをエディター ウィンドウで開きます (図 11)。



```

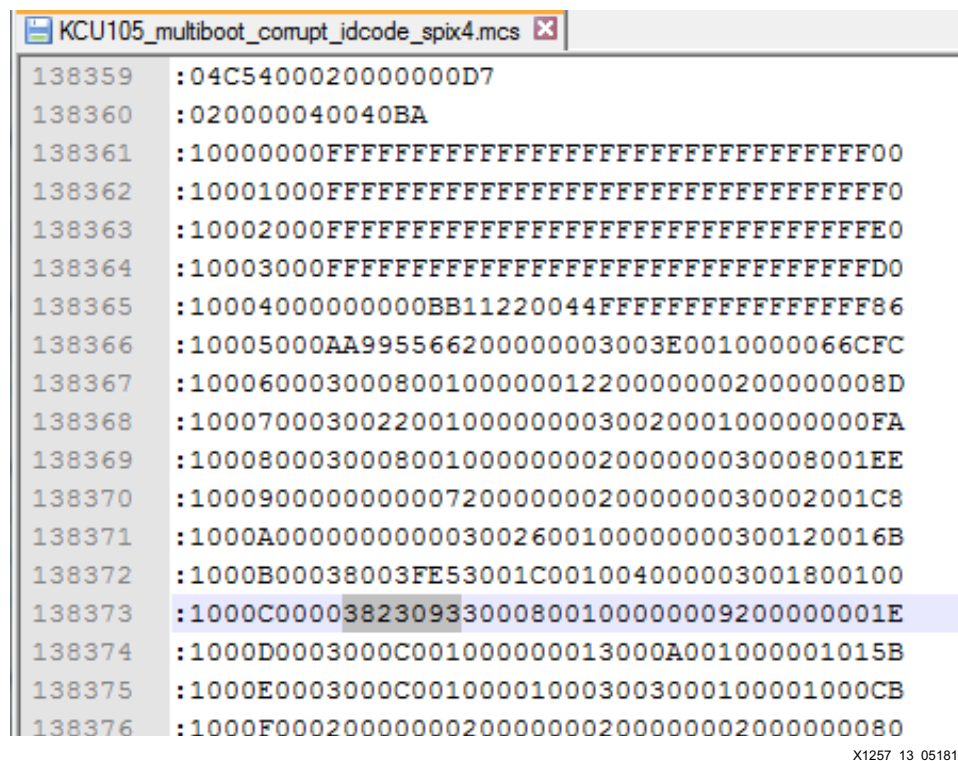
138359 :04C5400020000000D7
138360 :020000040040BA
138361 :10000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
138362 :10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
138363 :10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
138364 :10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
138365 :10004000000000BB11220044FFFFFFFFFFFFFFFFF86
138366 :10005000AA995566200000003003E0010000066CFC
138367 :10006000300080010000001220000000200000008D
138368 :1000700030022001000000003002000100000000FA
138369 :1000800030008001000000002000000030008001EE
138370 :10009000000000007200000002000000030002001C8
138371 :1000A00000000000003002600100000000300120016B
138372 :1000B00038003FE53001C001004000003001800100
138373 :1000C000038220933000800100000009200000001E
138374 :1000D0003000C001000000013000A001000001015B
138375 :1000E0003000C001000010003003000100001000CB
138376 :1000F000020000000020000000200000002000000080

```

X1257_12_051815

図 11: オリジナルのマルチブート イメージ

- アップデート イメージの IDCODE は MCS ファイル内に格納されています。X3822093 を X3823093 に変更して、.../ready_to_download/KCU105_multiboot_corrupt_idcode_spix4.mcs としてファイルを保存します (図 12)。



```

138359 :04C5400020000000D7
138360 :020000040040BA
138361 :1000000FFFFFFFFFFFFFFFFFFFFFFFFF00
138362 :10001000FFFFFFFFFFFFFFFFFFFFFFFFF0
138363 :10002000FFFFFFFFFFFFFFFFFFFFFFFFFE0
138364 :10003000FFFFFFFFFFFFFFFFFFFFFFFFFD0
138365 :10004000000000BB11220044FFFFFFFFF86
138366 :10005000AA995566200000003003E0010000066CFC
138367 :10006000300080010000001220000000200000008D
138368 :100070003002200100000000300200010000000FA
138369 :1000800030008001000000002000000030008001EE
138370 :1000900000000007200000002000000030002001C8
138371 :1000A000000000003002600100000000300120016B
138372 :1000B00038003FE53001C001004000003001800100
138373 :1000C00038230933000800100000009200000001E
138374 :1000D0003000C001000000013000A001000001015B
138375 :1000E0003000C001000010003003000100001000CB
138376 :1000F000200000000200000000200000002000000080

```

X1257_13_051815

図 12: IDCODE が不正なマルチブート イメージ

- 「[フラッシュのプログラム](#)」で説明した手順に従って、新しく作成された MCS ファイルで SPI フラッシュをプログラムします。
- FPGA が SPI フラッシュのゴールデン ビットストリームで正常にコンフィギュレーションされているかを検証します。検証方法は次のとおりです。
 - ボード上の DONE ピン LED が点灯しているかを確認します。
 - GPIO LEDs [3:0] が左から右へ順番に点灯し、ゴールデン ビットストリームが正常に読み込まれたことを示しているかを確認します。
 - GUI で FPGA を右クリックして [Hardware Device Properties] をクリックし、デバイスをリフレッシュします。

- 。 Vivado GUI の [Properties] ウィンドウで、[REGISTER] の下にある [BOOT_STATUS] と [CONFIG_STATUS] を展開表示します。BOOT_STATUS レジスタでは、アップデート ビットストリームへのジャンプをもたらした IPROG (INTERNAL_PROG) フラグが High であることを確認します。CONFIG_STATUS レジスタでは、DONE_PIN が High を示しています (図 13)。

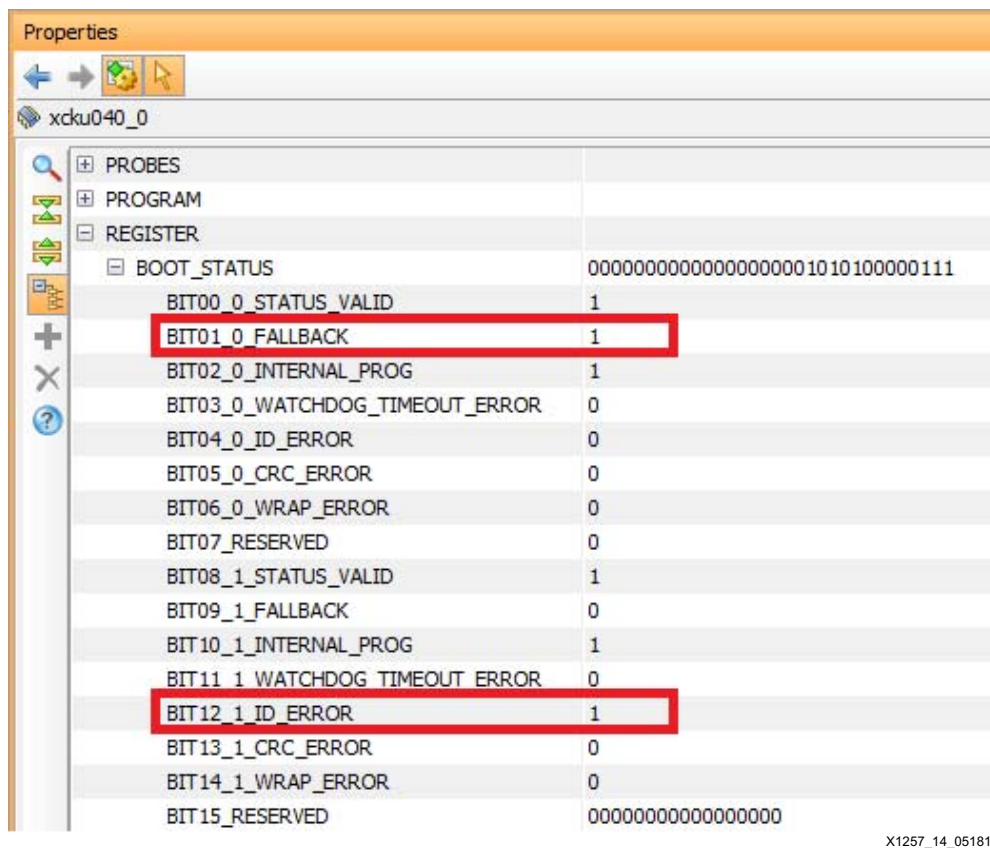


図 13: フォールバックと ID_ERROR を示す BOOT_STATUS

フォールバック - ウォッチドッグ タイマー

場合によっては、コンフィギュレーションが開始されない (おそらく、next_config_addr にイメージがないことが原因) またはコンフィギュレーションは開始されたが完了しない (おそらく、アップデート イメージの SYNC ワードが破損、またはコンフィギュレーション ソースの一部が破損していることが原因) ことがあります。このような場合には、ビットストリーム オプションでウォッチドッグ タイマーを使用すると、一定時間の経過後にコンフィギュレーションの再試行が可能になります。つまり、ウォッチドッグ タイマーがタイムアウトになると、コンフィギュレーション ロジックがフォールバック ビットストリームを読み込みます。ウォッチドッグ タイマーは、ビットストリーム プロパティ BITSTREAM.CONFIG.TIMER_CFG で設定できます。この設定は、ゴールデン イメージとアップデート イメージの両方で必要です。

TIMER レジスタはビットストリームの先頭からカウントダウンを開始し、スタートアップ シーケンスが完了すると無効になります。カウントが 0 に達すると、フォールバックが開始されます。考慮する必要がある遅延は、スタートアップ時のすべての待機時間 (MMCM のロック、DCI の一致、DONE 信号) です。TIMER レジスタの動作周波数は約 50MHz です。公称周波数が 50MHz の専用の内部クロック CFGMCLK を使用します。このクロックは、256 分周して供給されるため、ウォッチドッグ タイマーのクロック周期は約 5,120ns となります。ウォッチドッグ カウンターは 30 ビット幅のため、このタイマーを最大で約 5,500 秒までカウントできます。

デザインのデバッグ

このセクションでは、SPI フラッシュを使用するマルチブートの一般的な問題をデバッグする場合に役立つ手順を示します。

ファイル生成

- ゴールデン イメージとアップデート イメージの両方に対して、マルチブート ビットストリームのすべてのプロパティが適切に設定されていることを確認します (表 1 参照)。
- SPI ビットストリームのすべてのプロパティが適切に設定されていることを確認します (表 1 参照)。
- 「[write_cfgmem を使用して SPI フラッシュ ファイルを生成](#)」で説明したとおりに、フラッシュ プログラミング ファイルを生成するためのコマンドに適切なオプションがすべて含まれていることを確認します。
- アップデート イメージの開始アドレスがビットストリーム プロパティ NEXT_CONFIG_ADDR に指定したアドレスと同じであることを確認します。

コンフィギュレーション

- マルチブート プロパティを取り入れる前に、ゴールデン イメージとアップデート イメージを両方含むデザインが予想どおりに動作することを確認してください。これによって、問題の根本的原因がデザインにあるか、またはマルチブート プロパティにあるかを判断できます。リファレンス デザインに含まれる次の MCS ファイルを使用してコンフィギュレーションを実行することで、動作を確認できます。

- KCU105_golden_spix4.mcs
- KCU105_update_spix4.mcs

これらにはマルチブート プロパティが含まれていませんが、各イメージの動作を示すことは可能です。

- ゴールデン イメージは、GPIO LED [3:0] を左から右へ順番に点灯します。
- アップデート イメージは、GPIO LED [3:0] を右から左へ順番に点灯します。
- デザインでフラッシュのプログラミングを実行する前に、フラッシュ デバイスが完全に消去されていることを確認します。消去されたかは、ブランク チェック オプションで確認できます。
- マルチブート デザインのデバッグでは、BOOT_STATUS および CONFIG_STATUS レジスタを使用してエラーや動作の問題を確認します。レジスタを読み出す前に、デバイスのリフレッシュが完了していることを確認してください。

まとめ

このアプリケーション ノートは、フィールドでのシステム アップデートをサポートする UltraScale FPGA のマルチブート機能を使用する方法を説明し、SPI (x4) コンフィギュレーション インターフェイスでこの機能を実装する際のガイダンスを提供しています。また、マルチブート機能の動作を検証するためのリファレンス デザインも提供しています。

リファレンス デザイン

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 2 に、リファレンス デザインの詳細を示します。

表 2: 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	Wendy Curran
ターゲット デバイス	Kintex UltraScale FPGA
ソース コードの提供	あり
ソース コードの形式	VHDL
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、またはサードパーティからデザインへのコード/IP の使用	N/A
シミュレーション	
論理シミュレーションの実施	N/A
タイミングシミュレーションの実施	N/A
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	N/A
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 2015.1
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 2015.1
スタティック タイミング解析の実施	N/A
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	KCU105 評価ボード、256Mb Micron 社製シリアル NOR フラッシュ メモリ (N25Q256A)

参考資料

- 『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570: [英語版](#)、[日本語版](#))
- 『Vivado Design Suite ユーザー ガイド: プログラムおよびデバッグ』(UG908: [英語版](#)、[日本語版](#))
- 『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835: [英語版](#)、[日本語版](#))
- 『KCU105 ボード ユーザー ガイド』([UG917](#))

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
2015年9月30日	1.0	初版

法的通知

本通知に基づいて貴殿または貴社（本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ）に開示される情報（以下「本情報」といいます）は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で（with all faults）という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず（商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません）、すべての保証および条件を負わない（否認する）ものとし、また、(2) ザイリンクスは、本情報（貴殿または貴社による本情報の使用を含む）に関し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない（契約上、不法行為上（過失の場合を含む）、その他のいかなる責任の法理によるかを問わない）ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害（第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます）が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、<http://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うこととなります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。<http://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。

© Copyright 2015 Xilinx, Inc. Xilinx、Xilinx のロゴ、Artix、ISE、Kintex、Spartan、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他の各国のザイリンクス社の商標です。すべてのその他の商標は、それぞれの保有者に帰属します。

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。