

注意:この日本語版文書は参考資料としてご利用ください。 最新情報は必ずオリジナルの英語版をご参照願います。

# PolarFire® FPGA および PolarFire SoC FPGA プログラミング ユーザガイド

# はじめに

Microchip 社の PolarFire FPGA は不揮発性 FPGA の第 5 世代ファミリであり、最新の 28 nm 不揮発性プロセスを基に構築されています。PolarFire FPGA は、ミッドレンジの集積度でトップレベルの低消費電力性能を提供します。PolarFire FPGA は業界トップレベルの低消費電力 FPG ファブリック、Gbps トランシーバ レーン、低消費電力デュアル PCI Express Gen2 (EP/RP)、オプションのデータ セキュリティ(S)デバイス、低消費電力暗号コプロセッサを内蔵する事により、ミッドレンジ FPGA のコストを削減します。

Microchip 社の PolarFire SoC FPGA は、不揮発性 SoC FPGA デバイスの第 5 世代ファミリであり、最新の 28 nm 不揮発性プロセスを基に構築されています。PolarFire SoC ファミリは、Linux をサポートする業界初の RISC-V ベース SoC FPGA を提供します。PolarFire SoC ファミリのデバイスは、SiFive 社の U54-MC に基づく高性能 64 ビット 5x コア RISC-V マイクロプロセッサ サブシステム(MSS)と PolarFire FPGA ファブリックを 1 つのデバイスに実装しています。

PolarFire FPGA および PolarFire SoC FPGA は、幅広いエンドユーザ アプリケーションに対応するために各種のプログラミング オプションを提供します。表 1 に、PolarFire FPGA および PolarFire SoC FPGA ファミリ内でプログラミング可能なコンポーネントを示します。

#### 表 1. プログラミング可能 コンポーネント

A CONTRACTOR OF THE CONTRACTOR						
コンポーネント	PolarFire FPGA (MPF)	PolarFire SoC FPGA (MPFS)				
FPGA ファブリック	✓	1				
セキュア不揮発性メモリ(sNVM)	✓	1				
組み込み不揮発性メモリ(eNVM)	_	1				
ユーザ セキュリティ設定 (鍵、パスコード、ロック)	✓	✓				

どちらのデバイスファミリも、内蔵システム コントローラを使って専用の JTAG または SPI インターフェイス経由でプログラミングできます。使用するインターフェイスに応じて以下の 3 通りのプログラミング モードが使えます。

- JTAG
- SPI マスタ
- SPI スレーブ

JTAG および SPI スレーブ プログラミング モードの場合、外部マスタ(マイクロプロセッサ等)またはバージョン 5 以上の Microchip FlashPro プログラマを使ってデバイスをプログラミングできます。外部マスタは、外部メモリからプログラミング データ (ビットストリーム)をフェッチします。

SPI マスタモードの場合、システム コントローラがマスタとして機能して外部 SPI フラッシュメモリからビットストリームをフェッチする事により、デバイスをプログラミングします。このモードは 2 つのプログラミング機能(自動更新機能と IAP (In-Application Programming)機能)をサポートします。自動更新機能では、電源投入時にデバイスが自動的に再プログラミングを実行します。IAP 機能では、ユーザ アプリケーションによってデバイスのプログラミングを開始します。

#### 図 1. プログラミング モード

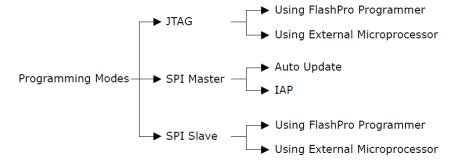
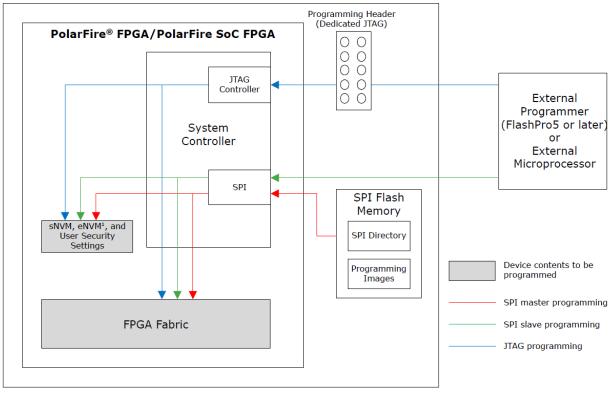


図2に、各種デバイス プログラミング モードと関連するインターフェイスのブロック図を示します。

#### 図 2. デバイス プログラミング モードとインターフェイス



<sup>&</sup>lt;sup>1</sup> Applicable for PolarFire SoC FPGA only.

Note: デバイスがシステム コントローラ サスペンドモードで使われる場合、シングルイベント アップセットによる 誤ったプログラミングからデバイスを保護するために、デバイス プログラミング機能は無効になります。サスペンドモードでは、システム コントローラはデバイスの初期化後にリセット状態に保持されるためセキュリティ、IAP、自動更新プログラミング等のシステム サービスを提供できません。デバイスがシステム コントローラ サスペンドモードを終了した後は、JTAG および SPI スレーブ プログラミング モードが利用可能となります。

# 参考資料

- sNVM、eNVM、セキュリティ設定に関する情報: PolarFire FPGA and PolarFire SoC FPGA Security User Guide.
- プログラミング サイクル数に関する情報: PolarFire FPGA and PolarFire SoC FPGA System Services User Guide.
- 回路初期化に関する情報: PolarFire FPGA and PolarFire SoC FPGA Device Power-Up and Resets User Guide.
- 電源要求とフィルタ コンデンサに関する情報: UG0726: PolarFire FPGA Board Design User Guide または UG0901: PolarFire SoC Board Design Guidelines User Guide
- PolarFire FPGA および PolarFire SoC FPGA 向け Libero SoC の使用法: Libero SoC Documentation.
- MSS に関する情報: PolarFire SoC FPGA MSS Technical Reference Manual.

# 目次

はし	〕めに		1
	1.	参考資料	3
1.	ビット	- ストリームの生成	6
	1.1.	ビットストリームの生成フロー	6
	1.2.	ビットストリームに sNVM データを追加する	7
	1.3.	ビットストリームに eNVM データを追加する(PolarFire SoC FPGA のみ)	
	1.4.	ビットストリームにユーザ セキュリティ設定を追加する	8
	1.5.	ビットストリーム コンポーネントの設定	
	1.6.	プログラミング ファイルのサイズ	13
2.	デバイ	′ス プログラミング フロー	16
	2.1.	プログラミング時間	17
3.	プログ	<sup>ず</sup> ラミング モード	18
	3.1.	JTAG プログラミング	18
	3.2.	SPI スレーブ プログラミング	22
	3.3.	SPI マスタ プログラミング	25
4.	バック	, レベル保護のバイパス	43
	4.1.	バックレベル保護バイパス機能が必要になる事例	45
5.	プログ	ブラミング中の I/O ステート	46
6.	プログ	ブラミング中の MSS ステート(PolarFire SoC FPGA のみ)	48
7.	プログ	<sup>ず</sup> ラミングに関する推奨事項	49
8.	プログ	<sup>ず</sup> ラミング中のブラウンアウト	50
9.	ゼロイ	<u></u>	51
10.	外部 5	SPI フラッシュのプログラミング	52
	10.1.	サポートする SPI フラッシュデバイス	52
		SPI ディレクトリ	
	10.3.	SPI フラッシュ のプログラミング方法	53
11.	シスラ	- ム コントローラ サスペンドモード	58
12.	補遺:	エラーコード	59
13.	改訂層	<u>履歴</u>	61
Mic	crochip	社の FPGA サポート	64
Mic	crochip	社ウェブサイト	64
製品	品変更通	<b>賃知サービス</b>	64

カスタマサポート	64
Microchip 社のデバイスコード保護機能	64
法律上の注意点	
商標	65
品質管理システム	66
各国の営業所とサービス	67

# 1. ビットストリームの生成

Libero® SoC デザインスイートは、各種プログラミング モード向けのビットストリームを生成します。プログラミング ビットストリームは、要件に応じて 1 つまたは複数のコンポーネントを格納できます。これらのコンポーネントに は以下が含まれます。

- FPGA ファブリック ロジック
- sNVM データ
- eNVM データ(PolarFire SoC FPGA のみ)
- ユーザ セキュリティ設定

表 1-1に、各種プログラミング モードで使われるインターフェイスと、ビットストリームの書式を示します。

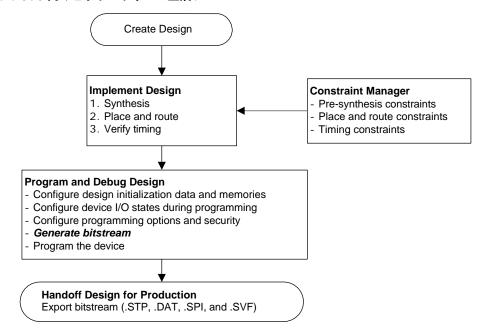
表 1-1. PolarFire FPGA および PolarFire SoC FPGA のプログラミング インターフェイスとビットストリームの書式

プログラミング モード	インターフェイス	マスタ	ビットストリームの書式
JTAG	システム コントローラ専用 JTAG	FlashPro プログラマ	STP
JTAG	システム コントローラ専用 JTAG	外部マイクロプロセッサ	DAT
SPIスレーブ	システム コントローラ専用 SPI	FlashPro プログラマ	DAT
SPIスレーブ	システム コントローラ専用 SPI	外部マイクロプロセッサ	DAT
SPIマスタ	システム コントローラ専用 SPI	システム コントローラ	SPI

# 1.1 ビットストリームの生成フロー

図 1-1 に、Libero SoC 内のビットストリーム生成フローを示します。

#### 図 1-1. Livero SoC 内のビットストリーム生成フロー

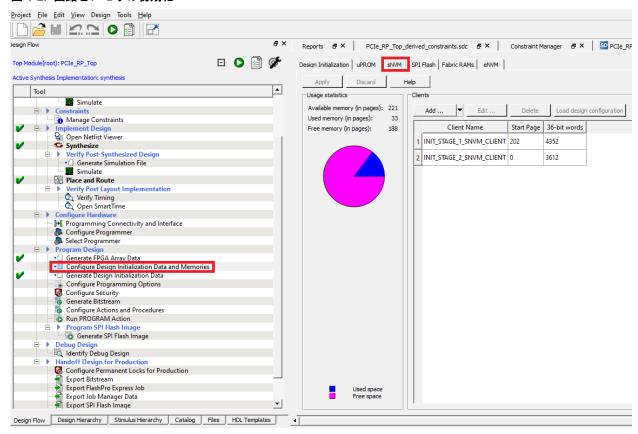


# 1.2 ビットストリームに sNVM データを追加する

sNVM は個別にプログラミング可能なユーザ用不揮発性フラッシュメモリです。各デバイスは 56K バイトの sNVM を備えています。

Libero SoC 内で複数の sNVM データクライアントをビットストリームに追加するには、**Design Flow > Program Design > Configure Design Initialization Data and Memories** と操作します(図 1-2 参照)。

#### 図 1-2. 回路とメモリの初期化

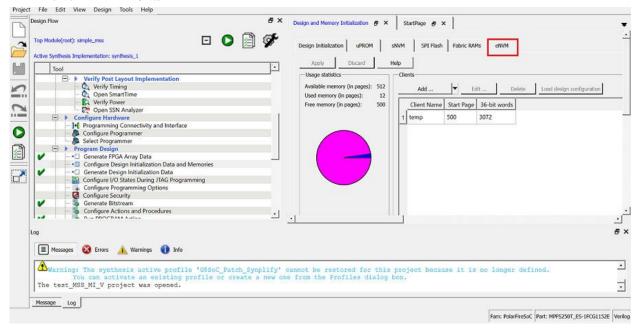


# 1.3 ビットストリームに eNVM データを追加する(PolarFire SoC FPGA のみ)

eNVM は個別にプログラミング可能なユーザ用不揮発性フラッシュメモリです。各 PolarFire SoC デバイスは 128K バイトの eNVM を備えています。

Libero SoC 内で複数の eNVM データクライアントをビットストリームに追加するには、**Design Flow > Program Design > Configure Design Initialization Data and Memories** と操作します(図 1-3 参照)。

#### 図 1-3. 回路とメモリの初期化

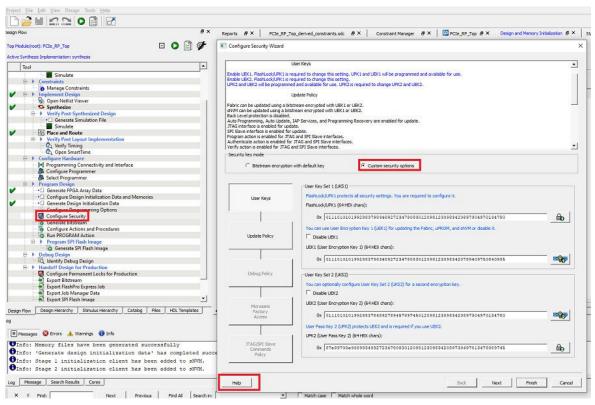


# 1.4 ビットストリームにユーザ セキュリティ設定を追加する

どちらのデバイスファミリも、一意のファクトリー鍵群を使ってプロビジョニングされます。加えて、Microchip 社が提供する鍵を一切使わずに、ユーザ独自のセキュリティ鍵を登録する事もできます。ユーザ暗号鍵 1 (UEK1)とユーザ暗号鍵 2 (UEK2)は、ユーザ定義の AES-2 対称鍵です。どちらの鍵もビットストリームの暗号化/復号および認証用のルート鍵として使えます。

ユーザ セキュリティ設定は、以下の手順によりビットストリームに追加できます。

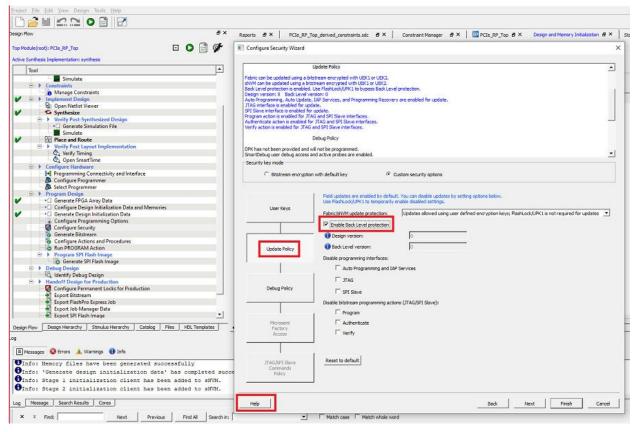
1. Libero SoC 内で **Design Flow > Program Design > Configure Security > Custom security options** と操作します(図 1-4 参照)。



## 図 1-4. ユーザ セキュリティの設定 - カスタム セキュリティ オプション

2. 更新ポリシー(Update Policy)を変更するために[Next>]をクリックします。すると「Configure Security Wizard」(図 1-5 参照)が開きます。

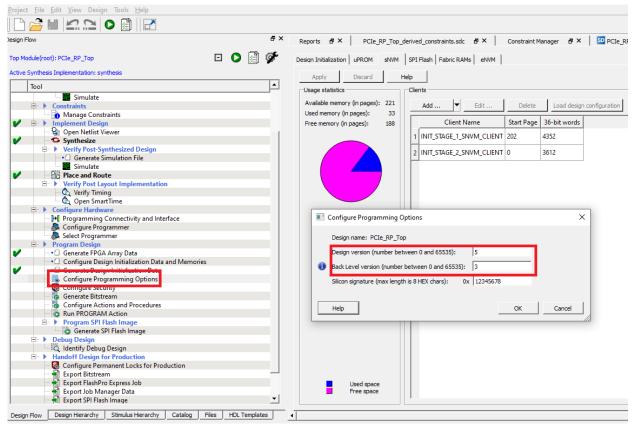
## 図 1-5. Configure Security Wizard - Update Policy



バックレベル保護機能([Enable Back Level protection])を有効にした場合、[Design version](プログラミングするビットストリームのバージョン) は[Back Level version]よりも高い事が必要です。[Help]をクリックすると設定フィールドの説明が表示されます。[Back Level version] の値は、デバイスが更新データとして受け付けるビットストリームのバージョンを制限します。デバイスに保存されている現在の[Back Level Version]値より大きな[Design version]値を持つビットストリームだけがデバイスへのプログラミングを許可されます。このバックレベル保護機能は FlashLock/UPK1 により管理され、バイパス可能です。[Back Level version]と[Design version]は、「Configure Programming Options」ダイログ内で変更できます。SNVM とセキュリティ設定の詳細は、『PolarFire FPGA and PolarFire SoC FPGA Security User Guide』を参照してください。

図 1-6 に、プログラミング オプションの設定を示します。

# 図 1-6. プログラミング オプションの設定



バックレベル保護機能のバイパス方法は、4.1. 「バックレベル保護機能のバイパス方法」を参照してください。

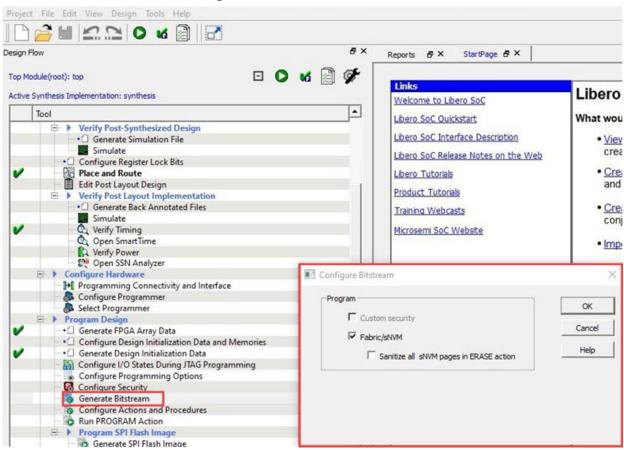
# 1.5 ビットストリーム コンポーネントの設定

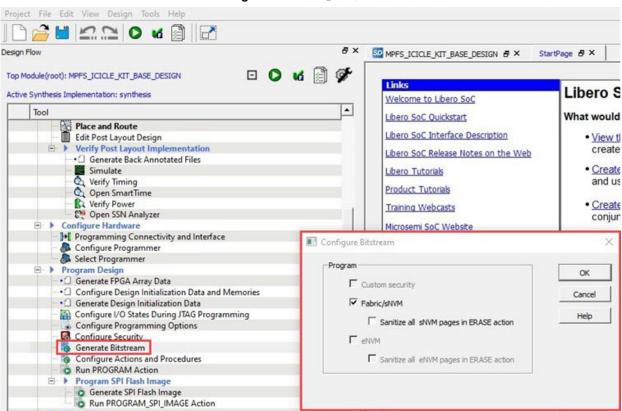
セキュリティ設定とビットストリーム コンポーネント (ファブリック、sNVM、eNVM(PolarFire SoC のみ)等) の設定手順は以下の通りです。

Libero SoC内で [Design Flow] > [Program Design] > [Program Design]と操作します。

- 1. [Generate Bitstream]を右クリックし、[Configure Options...]を選択します。
- 2. 「Configure Bitstream」ウィンドウが開きます。
- 3. [Custom security]、[Fabric/sNVM]、[eNVM](PolarFire SoC FPGA のみ)を選択します。
- 4. [OK]をクリックします。

# 図 1-7. PolarFire FPGA 向けの「Configure Bitstream」ウィンドウ





# 図 1-8. PolarFire SoC FPGA 向けの「Configure Bitstream」ウィンドウ

ビットストリーム ファイルをエクスポートするには、[Design Flow] > [Handoff Design for Production] > [Export Bitstream]と操作します。

Note: セキュリティ設定のみのビットストリームは、消去済みデバイスまたはブランクデバイスに書き込む必要があります。セキュリティ ビットストリームをプログラミング済み FPGA に書き込むと、FPGA アレイが無効になります。FPGA アレイを有効にするには、ファブリックを再書き込みする必要があります。

# 1.6 プログラミング ファイルのサイズ

プログラミング ファイルは、工場で書き込まれた鍵またはユーザ独自の鍵を使って暗号化されます。従って、それらのファイル(.dat または.spi)を圧縮してファイルサイズを削減する事はできません。表 1-2 に、「Configure Bitstream」ウィンドウ内で[Custom security] を無効にした場合の PolarFire FPGA 向けプログラミング ファイルのサイズを示します。

表 1-2. PolarFire FPGA 向けプログラミング ファイルのサイズ - [Custom security] 無効

		PolarFire FF	PGA		
		MPF100	MPF200	MPF300	MPF500
ファブリックと sNVM (kB)	STAPL	5585	9611	14772	23446
	DAT	3497	6043	9307	14789
	SPI	3496	6041	9305	14788

表 1-3 に、「Configure Bitstream」ウィンドウ内で[Custom security] を有効にした場合の PolarFire FPGA 向けプログラミング ファイルのサイズを示します。

表 1-3. PolarFire FPGA 向けプログラミング ファイルのサイズ - [Custom security]有効

			PolarFire F	PGA		
			MPF100	MPF200	MPF300	MPF500
カスタム セキュリティ、	STAPL	マスタファイル	5595	9621	14784	23456
ファブリック、sNVM (kB)		UEK1/UEK2	5585	9611	14774	23446
	DAT	マスタファイル	3502	6047	9312	14794
		UEK1/UEK2	3497	6043	9307	14789
	SPI	マスタファイル	3498	6044	9308	14790
		UEK1/UEK2	3496	6041	9305	14788
カスタム セキュリティ(kB)	STAPL	NA	84	84	84	84
	DAT	NA	8	8	8	8
	SPI	NA	4	4	4	4

例えばセキュリティ、ファブリック、sNVM を格納した MPF200 向けプログラミング ファイルのサイズ(SPI) は 6044 KB (約 6 MB)です。表 1-4 に、保存するプログラミング イメージの数に応じて必要となる外部 SPI フラッシュメモリのサイズを示します。

表 1-4. PolarFire FPGA - 外部 SPI フラッシュメモリの必要サイズ(概略値)

イメージの数	外部 SPI フラッシュのサイズ
1	6 MB
2	12 MB
3	18 MB
4	24 MB

表 1-5 に、「Configure Bitstream」ウィンドウ内で[Custom security] を無効にした場合の PolarFire SoC FPGA 向けプログラミング ファイルのサイズを示します。

表 1-5. PolarFire SoC FPGA 向けプログラミング ファイルのサイズ - [Custom security]無効

		PolarFire SoC FPGA <sup>1</sup>		
		MPFS250T		
ファブリック、sNVM、eNVM (kB)	STAPL	14717		
	DAT	9261		
	SPI	9259		
<sup>1</sup> MPFS250T 以外の PolarFire SoC FPGA のファイルサイズについては今後更新予定です。				

表 1-6 に、「Configure Bitstream」ウィンドウ内で[Custom security]を有効にした場合の PolarFire SoC FPGA 向けプログラミング ファイルのサイズを示します。

表 1-6. PolarFire SoC FPGA 向けプログラミング ファイルのサイズ - [Custom security] 有効

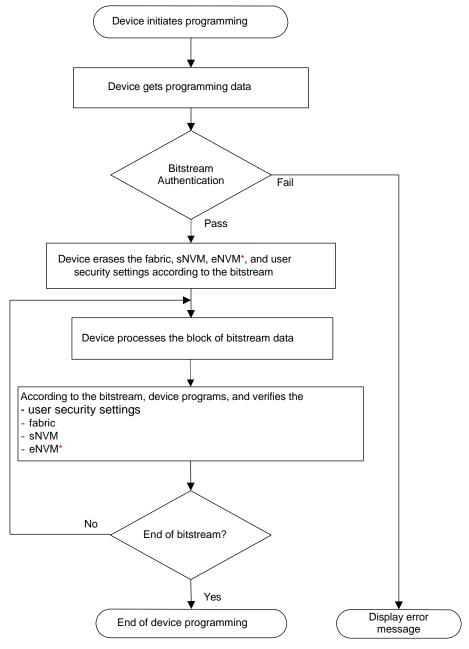
			PolarFire SoC FPGA <sup>1</sup>		
			MPFS250T		
カスタム セキュリティ、ファブリック、	STAPL	マスタファイル	14721		
sNVM (kB)		UEK1/UEK2	14714		
	DAT	マスタファイル	9263		
		UEK1/UEK2	9258		
	SPI	マスタファイル	9259		
		UEK1/UEK2	9257		
カスタム セキュリティ(kB)	STAPL	NA	85		
	DAT	NA	8		
	SPI	NA	4		
<sup>1</sup> その他の PolarFire SoC FPGA デバイスはアップデート予定です。					

# 2. デバイス プログラミング フロー

デバイスのプログラミング フローは、システム コントローラがデバイス プログラミング命令を受信または開始した時に始まり、ビットストリーム データが完全に転送および検証された時に終わります。システム コントローラは、デバイスに書き込むビットストリーム データをブロックごとにフェッチします。プログラミング フローには、ビットストリームの認証と書き込み済み内容の検証が含まれます。セキュリティ設定は、デバイス消去またはデバイス プログラミングの完了後に有効になります。プログラミングに成功するとシステム コントローラはデバイスをリセットし、書き込まれた内容で実行を開始します。プログラミング フローは、全てのプログラミング モードで同じです。

図 2-1 に、デバイス プログラミング フローの概要を示します。

#### 図 2-1. プログラミング フロー



<sup>\*</sup> Applicable for PolarFire SoC FPGA only

Note: デバイス消去動作でも書き込み回数のカウントはインクリメントします(消去動作も内部ではプログラミング動作と同じであるため)。書き込み回数のカウントについては、『PolarFire FPGA and PolarFire SoC FPGA System Services User Guide』を参照してください。

# 2.1 プログラミング時間

プログラミング時間とはデバイスの消去、ビットストリーム データの処理、デバイスへの書き込み、書き込んだ内容の検証に要する総時間です。書き込まれた内容は、次のデータブロックの書き込み中に検証されます。書き込みと検証を同時に実行する事により、総プログラミング時間が大幅に短縮されます。

どちらのデバイスファミリでも、総プログラミング時間は 60 秒未満です。特定デバイスにおける各プログラミングモードでのプログラミング時間については、『PolarFire FPGA Datasheet』または『PolarFire SoC FPGA Advance Datasheet』を参照してください。

# 3. プログラミング モード

以下では、3種類のプログラミングモードについて説明します。

# 3.1 JTAG プログラミング

PolarFire FPGA と PolarFire SoC FPGA は IEEE 1149.1 および IEEE 1532 規格に準拠する JTAG コントローラを内蔵しています。JTAG コントローラは、コマンドレジスタを使って JTAG 命令を送信し、128 ビット データバッファを使って全ての関連データを転送する事により、システム コントローラと通信します。

#### 3.1.1 JTAG プログラミング インターフェイス

PolarFire FPGA と PolarFire SoC FPGA の JTAG ピンは、専用 I/O バンク 3 VDDI に配置されています。JTAG プログラミング中の I/O ステートの詳細は、5. 「プログラミング中の I/O ステート」を参照してください。

JTAG バンクの動作電圧は 1.8 V、2.5 V、 3.3 V に設定可能です。表 3-1 に JTAG ピンの一覧を示します。

#### 表 3-1. JTAG ピン

ピン名	方向	Weak プルアップ /未使用時の状態	概要
TMS	入力	Yes/DNC	JATG テストモード選択
TRSTB	入力	Yes <sup>1</sup>	JTAG テストリセット デバイス動作中は Low に保持する必要があります。
TDI	入力	Yes/DNC	JTAG テストデータ入力 ATPG またはテストモードにおいて 4 ビット TDI バスを使う場合、この I/O は tdi[0]として使われます。
TCK	入力	No <sup>2</sup>	JTAG テストクロック
TDO	出力	No/DNC	JTAG テストデータ出力

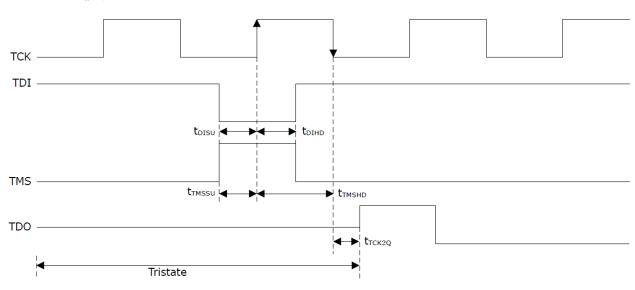
<sup>1.</sup> TRSTB が未使用かつアビオニクス モードである場合、内部 Weak プルアップをオーバーライドするために  $1 \text{ k}\Omega$  プルダウン抵抗を外付けするか、外部ソースから Low に駆動する必要があります。

#### 3.1.2 JTAG タイミング

JTAG プログラミングでは、図 3-1 に示す各 JTAG ピン信号のタイミング要件を満たす必要があります。推奨タイミング値については、『PolarFire FPGA Datasheet』または『PolarFire SoC Advance Datasheet』内の JTAG スイッチング特性を参照してください。

<sup>2.</sup> 未使用時は、10 kΩ 抵抗を介して VSS に接続する必要があります。

#### 図 3-1. JTAG 信号のタイミング

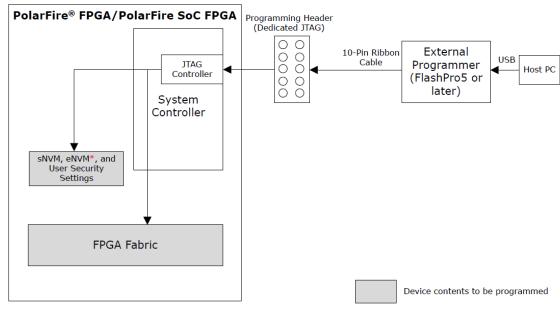


# 3.1.3 FlashPro を使った JTAG プログラミング

どちらのデバイスファミリも、Microchip 社の FlashPro プログラマ(バージョン 5 以上)を使って専用 JTAG インターフェイス経由でプログラミングできます。これには Libero SoC または FlashPro Express が使えます。

FlashPro プログラマは、FlashPro ケーブル(10 芯リボンケーブル)を使って 10 ピンのプログラミング ヘッダに接続します(図 3-2 参照)。

# 図 3-2. 外部プログラマを使った JTAG プログラミング



<sup>\*</sup> Applicable for PolarFire SoC FPGA only.

表 3-2 に、FlashPro ヘッダ信号の一覧を示します。

表 3-2. FlashPro ヘッダ信号

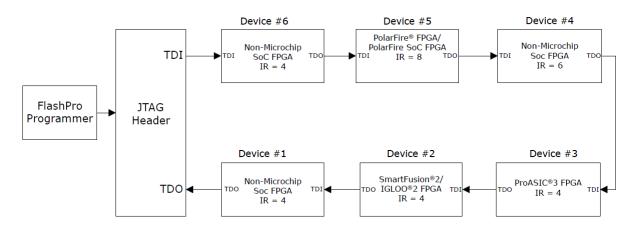
ピン番号	信号	FlashPro プログラマ から見た方向	概要
1	TCK/SCK	出力	JTAG/SPI クロック
2	GND	_	信号基準電位(GND ピンに接続)
3	TDO/MISO	入力	ターゲット デバイスからの JTAG/SPI データ出力
4	PROG_MODE	未接続	未使用
5	TMS/SS	出力	JTAG テストモード選択/SPI スレーブ選択
6	VJTAG/VSPI	_	ターゲット インターフェイス電圧入力
7	VPUMP	未接続	未使用
8	TRSTB	出力	JTAG テストリセット
9	TDI/MOSI	出力	ターゲット デバイスへの JTAG/SPI データ入力
10	GND	_	GND

1 つの FlashPro プログラマを使って、1 つの JTAG チェーン内で同一ファミリまたは異なるファミリからの複数の Microchip 社製 FPGA をプログラミングできます。JTAG チェーンは JTAG ヘッダの TDI ピンから始まり、最後のデバイスの TDO ピンから JTAG ヘッダの TDO ピンに戻ります。以下のタイプの FPGA を JTAG チェーンに追加できます。

- プログラミングのターゲットとなる Microchip 社製デバイス
- プログラミングのターゲットとならない Microchip 社製バイパスデバイス
- 他社製のバイパスデバイス

デバイスがバイパス モード中である場合、デバイスのデータ レジスタ長は自動的に 1 に設定され、デバイスはプログラミング命令への応答を停止します。デバイスをバイパスモードに設定するには、命令レジスタ(IR)の長さが既知である事が必要です。Microchip 社製 FPGA の IR 長は、FlashPro Express によって自動的に取得されます。他社製 FPGA の場合、バウンダリ スキャン コマンドとデータのシーケンスを格納した BSDL (Boundary Scan Description Language)ファイルをロードするか、FlashPro Express 内で IR 長を手動入力する必要があります。JTAG チェーン プログラミングの詳細は、『FlashPro User's Guide』を参照してください。

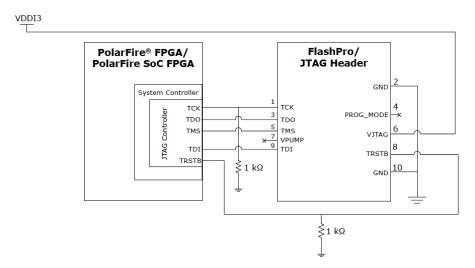
図 3-3. JTAG チェーンでのプログラミング



電源要件とフィルタ コンデンサに関する情報は、『UG0726: PolarFire FPGA Board Design User Guide』または『UG0901: PolarFire SoC Board Design Guidelines User Guide』を参照してください。

図 3-4 に、プログラミング ヘッダとデバイスの間の接続を示します。

# 図 3-4. FlashPro プログラマとデバイスの接続



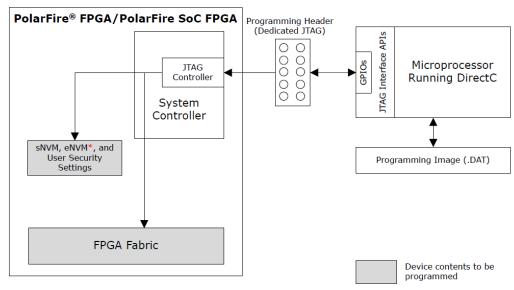
# 3.1.4 外部マイクロプロセッサを使った JTAG プログラミング

外部マイクロプロセッサを使って JTAG インターフェイス経由でデバイスをプログラミングする事ができます。この プログラミング方法では、外部マイクロプロセッサは DirectC (Microchip 社の FPGA 向けプログラミング ソリューション)を実行し、その GPIO ポートで JTAG インターフェイスを駆動する必要があります。

**Note:** DirectC ソリューションは FPGA ファブリック、sNVM、eNVM (PolarFire SoC FPGA のみ)、ユーザ セキュリティ設定のプログラミングをサポートします。DirectC を使うには、必要な API を追加してソースコードをコンパイルする事によりバイナリ実行可能ファイルを生成する必要があります。このバイナリ実行可能ファイルは、プログラミング データファイルと一緒に外部マイクロプロセッサにダウンロードされます。DirectC ユーザガイドの最新版は Microsemi 社の DirectC ソリューション ウェブページ から入手可能です。

セキュリティ設定のみのビットストリームは、消去済みデバイスまたはブランクデバイスに書き込む必要があります。 セキュリティ ビットストリームをプログラミング済み FPGA に書き込むと、FPGA アレイが無効になります。FPGA アレイを有効にするには、ファブリックを再書き込みする必要があります。 図 3-5 に、DirectC を実行する外部マイクロプロセッサを使ったデバイス プログラミングの実装例を示します。

# 図 3-5. 外部マイクロプロセッサを使ったプログラミング



<sup>\*</sup> Applicable for PolarFire SoC FPGA only.

# 3.2 SPI スレーブ プログラミング

どちらのデバイスファミリも、外部 SPI マスタ(外部マイクロプロセッサまたは FlashPro プログラマ)を使って SPI インターフェイス経由でプログラミングできます。システム コントローラ SPI をスレーブモードに設定する場合のピン設定は表 3-4 を参照してください。

SPI スレーブまたは SPI マスタのどちらのモードになるかは、デバイスのパワーオン リセット(POR)時の  $IO\_CFG\_INTF$  SPI ピンの状態によって決まり、動的に切り換える事はできません。 $IO\_CFG\_INTF$  ピンの設定(表 3-3 参照) によって SPI モードをスレーブからマスタ(またはその逆) に変更するには、電源の再投入またはデバイスリセット(DEVRST) が必要です。

SPI がスレーブモードである場合、ファブリックから SPI にアクセスする事はできません(SPI インターフェイスはシステム コントローラ専用となります)。

デバイスが SPI スレーブ プログラミング モードである場合、外部 SPI フラッシュからの回路初期化はサポートされません。回路初期化に関する情報は、『PolarFire FPGA and PolarFire SoC FPGA Power-up and Reset User Guide』を参照してください。

# 3.2.1 SPI スレーブ プログラミング インターフェイス

どちらのデバイスファミリも、SPI コントローラの設定用に2本のピン(SPI EN と IO CFG INTF)を備えています。

表 3-3 に、システム コントローラの SPI ピンの一覧を示します。この表には、未使用ピンに対して必要な措置(未使用時のピン状態)も記載しています。ピンの未使用時の状態と電源シーケンスに関する情報は、『UG0726: PolarFire FPGA Board Design User Guide』または『UG0901: PolarFire SoC FPGA Board Design Guidelines User Guide』を参照してください。

表 3-3. システム コントローラの SPI ピン

SPI ピン名	方向	概要	未使用時のピン状態
SCK	双方向	SPI クロック <sup>1</sup>	10 k $\Omega$ 抵抗を介して VSS に接続する必要があります。
SS <sup>2</sup>	双方向	SPI スレーブ選択 <sup>1</sup>	10 k $\Omega$ 抵抗を介して VSS に接続する必要があります。

SPI ピン名	方向	概要	未使用時の状態			
SDI	入力	SDI 入力 <sup>1</sup>	10 kΩ 抵抗を介して VDDI3 に接続する必要があります。			
SDO	出力	SDO 出力 <sup>1</sup>	DNC			
SPI_EN	入力	SPI イネーブル 0: SPI 出力を 3 ステートにする 1: SPI を有効にする 抵抗を介してプルアップまたはプルダウンするか、外部ソースから動的に駆動する事で、SPI I/O を有効または 3 ステートにします。	10 kΩ 抵抗を介して VSS に接続する必要があります。			
IO_CFG_INTF	入力	SPI I/O コンフィグレーション 0: SPI スレーブ インターフェイス 1: SPI マスタ インターフェイス 抵抗を介してプルアップまたはプルダウンします。	10 kΩ 抵抗を介して VSS に接続する必要 があります。			

1. システム コントローラと FPGA ファブリック/MSS の間で共有されます(MSS: PolarFire SoC FPGA のみが備えるマイクロプロセッサ サブシステム)。システム コントローラの SPI が有効かつマスタとして設定されている場合、システム コントローラはデバイスの電源投入後に SPI の制御権をファブリック/MSS に渡します。SPI\_EN ピンが Low(無効)である場合または SS ピンが High である場合、システム コントローラの SPI 出力は 3 ステートにされます。

2. システム コントローラの SS ピン信号はアクティブ Low です。未使用時の SS ピンは、VSS に接続してフローティング状態を防ぐ必要があります。

SPI\_EN および IO\_CFG\_INTF ピンは、基板上のジャンパまたはストラップピンによってデバイスの外部で設定する 必要があります。表 3-4 に、SPI スレーブ プログラミング向けの SPI\_EN ピンと CFG\_INTF ピンの設定を示します。

表 3-4. システム コントローラの SPI 設定 - SPI スレーブ

SPIピン		SPI スレーブ プログラミング	概要
SPI_EN	IO_CFG_INTF		
0	x	No	スレーブとマスタの間の動的な切り換えはできません。
1	0 (SPI スレーブモード)	Yes	IO_CFG_INTF ピンの設定(表 3-3 参照) によって SP モードを スレーブからマスタ(またはその逆) に変更するには、電源の再
1	1 (SPI マスタモード)	No	投入またはデバイスリセット(DEVRST_N)が必要です。

# 3.2.2 FlashPro プログラマを使った SPI スレーブ プログラミング

Microchip 社の FlashPro プログラマ(バージョン 5 以上)を使って専用 SPI 経由でデバイスをプログラミングできます。これには Libero SoC または FlashPro Express が使えます。FlashPro プログラマは、図 3-6 に示す通りにデバイスの SPI ポートに接続します。

ターゲットボードは VDD、VDD18、VDD25、VDDI3 電源を提供する必要があります。

#### VDDI3 $\leq_{1 k\Omega}$ ≥10 kΩ PolarFire® FPGA/ FlashPro PolarFire SoC FPGA Header (SPI Slave) SPI EN GND System Controller SCK SCK PROG MODE SDO MISO 5 5 7 VPUMI 9 MOSI SPI SS SDI FL\_GLD 8 10 GND IO\_CFG\_INTF $\leq_{1 \text{ k}\Omega}$ **≤**1 kΩ

#### 図 3-6. 外部プログラマを使った SPI スレーブ プログラミング

SPI スレーブ プログラミングを選択するには、Libero SoC Design Flow > Configure Hardware > Programming Connectivity and Interface と操作します。

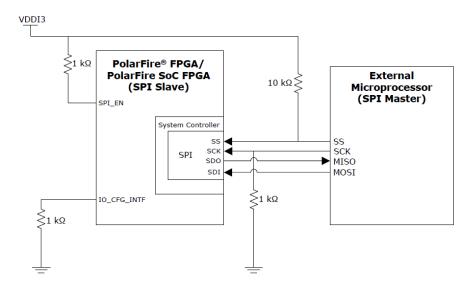
## 3.2.3 外部マイクロプロセッサを使った SPI スレーブ プログラミング

外部マイクロプロセッサ(ホスト PC または別の Microchip 社 FPGA 等)を使って専用 SPI ポート経由でデバイスをプログラミングできます(図 3-7 参照)。このタイプのプログラミングを行うには、Microchip 社の SPI-DirectC ソリューションを実行する外部マイクロプロセッサが必要です。外部マイクロプロセッサは、デバイスをプログラミングするために SPI\_EN、IO\_CNF\_INTF、DEVRST\_N ピンを制御する事もできます。

SPI-DirectC は FPGA ファブリック、sNVM、eNVM (PolarFire SoC FPGA のみ)、ユーザ セキュリティ設定のプログラミングをサポートします。SPI-DirectC を使うには、必要な API を追加してソースコードをコンパイルする事によりバイナリ実行可能ファイルを生成する必要があります。このバイナリ実行可能ファイルは、プログラミング データファイルと一緒に外部マイクロプロセッサにダウンロードされます。SPI-DirectC ユーザガイドの最新版は Microsemi社の DirectC ソリューション ウェブページから入手できます。[Downloads]タブからサンプル プロジェクト(Direct-C インストーラ)を入手する事もできます。

FlashPro ヘッダ信号については、表 3-2 を参照してください。

#### 図 3-7. 外部マイクロプロセッサを使った SPI スレーブ プログラミング



# 3.3 SPI マスタ プログラミング

システム コントローラ SPI をマスタとして設定した場合、デバイス自身がプログラミングを制御できます。SPI マスタ プログラミングの場合、SPI ディレクトリを使ってプログラミング イメージを外部 SPI フラッシュメモリに保存できます。SPI ディレクトリおよび外部 SPI フラッシュメモリへのプログラミングに関しては、10. 「外部 SPI フラッシュのプログラミング」を参照してください。

SPI マスタ プログラミングは自動更新機能と IAP (In-Application Programming)機能をサポートします。自動更新の場合、電源投入時に更新イメージのバージョンが現在プログラミングされているバージョンと異なる事が検出されると、システム コントローラは外部 SPI フラッシュメモリから更新イメージのビットストリームを読み出してデバイスをプログラミングします。IAP の場合、ユーザ アプリケーションがデバイスのプログラミングを開始し、システム コントローラが外部 SPI フラッシュメモリからビットストリームを読み出してデバイスにプログラミングします。自動更新と IAP はアトミックに実行され、JTAG または SPI スレーブコマンドによって割り込む事はできません。

自動更新機能は既定値では無効であり、必要な場合に Libero SoC を使って有効にする必要があります。SPI マスタモードは自動プログラミングおよび自動リカバリ機能もサポートします(表 3-5 参照)。これらの機能は既定値で有効となり、ユーザによる設定は不要です。

SPI マスタ プログラミング中の I/O ステートに関しては、5. 「プログラミング中の I/O ステート」を参照してください。表 3-5 に、SPI マスタ プログラミングがサポートする機能と、それらの開始方法を示します。

表 3-5. SPI マスタ プログラミングがサポートする機能

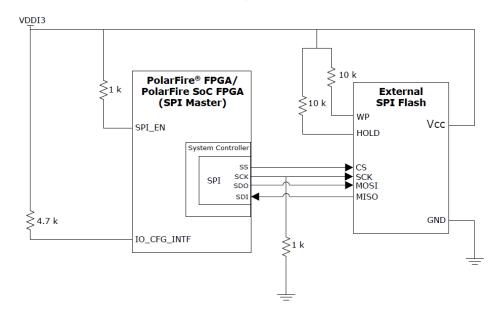
プログラミング機能	概要	開始方法	
自動プログラミング	ブランクデバイスをプログラミングします。	デバイスリセットまたは電源再投入	
自動更新	デバイスの内容を自動的に更新します。	デバイスリセット、電源再投入、 システムサービス要求	
IAP	ユーザの要求に応じてデバイスの内容を更新 します。	システムサービス要求	
自動リカバリ <sup>1</sup>	プログラミングの失敗からデバイスを自動的 にリカバリします。	プログラミング中のデバイス電源障害	

 $<sup>^1</sup>$  自動更新または IAP による eNVM (PolarFire SoC FPGA のみ)または sNVM の更新中に停電が発生すると、自動リカバリ機能がトリガされます。 eNVM または sNVM が完全に更新されなくてもデバイスは起動できます。しかし、eNVM または sNVM のプログラミングが不完全であった場合、ユーザ回路で誤動作が生じます。この場合、ユーザは別の方法(VERIFY、Degest Check 等)を使って、プログラミングが成功したかどうか確認する必要があります。

自動更新および IAP の実装に関しては、『AC466: PolarFire FPGA Auto Update and In- Application Programming Application Note』を参照してください。

図 3-8 に、SPI マスタ プログラミング向けに推奨する回路構成を示します。VDDI3 は、外部 SPI フラッシュのデータシートで指定されている電圧に適合する必要があります。

#### 図 3-8. SPI マスタ プログラミング向けに推奨する回路構成



## 3.3.1 SPI マスタ プログラミング インターフェイス

SPI\_EN および IO\_CFG\_INTF ピンは、基板上のジャンパまたはストラップピンによってデバイスの外部で設定する必要があります。表 3-6 に、SPI マスタ プログラミング向けの SPI\_EN ピンと CFG\_INTF ピンの設定を示します。

表 3-6. システム コントローラの SPI 設定 - SPI マスタ

SPIピン		SPI マスタ プログラミング		
SPI_EN	IO_CFG_INTF	IAP	自動更新	
0	х	No	No	
1	0 (SPI スレーブモード)	No	No	
1	1 (SPI マスタモード)	Yes	Yes	

#### 3.3.1.1 システム コントローラ SPI のモードとクロック

システム コントローラ SPI は、SPI フラッシュを読み出すためにデータ転送モード 3 (SPI モード 3)で動作します。このデータ転送モードでは、クロックパリティ(SPO/CPOL)とクロックフェイズ(SPH/CPHA) の両方を HIGH に設定する必要があります。システム コントローラの SPI は、20 MHz に固定されたクロックを使って動作します。

# 3.3.2 システムサービス

PolarFire FPGA と PolarFire SoC FPGA は、システム コントローラを使ってユーザからのシステムサービス要求に対応します。

ユーザ アプリケーションは、プログラミングに関連する以下のシステムサービスを要求できます。

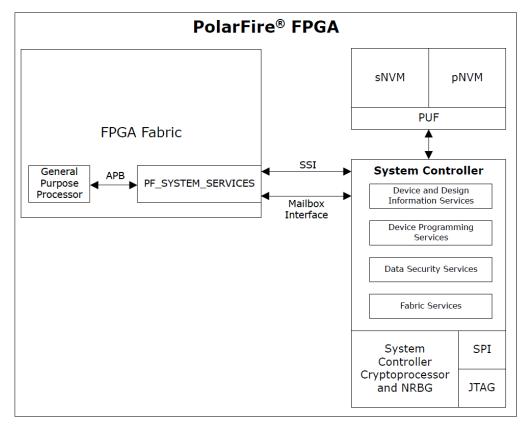
- ビットストリーム認証
- IAPイメージ認証
- 自動更新
- IAP

# 3.3.2.1 PolarFire FPGA のシステムサービス

PolarFire FPGA の場合、システムサービスはファブリック ユーザロジックからシステム コントローラのシステムサービス インターフェイス(SSI)を介して開始されるシステム コントローラ動作です。システムサービスを開始するた

めに、ファブリック ユーザロジックは PF\_SYSTEM\_SERVICES SgCore IP (Libero カタログ内で利用可能)を必要とします。図 3-9 に、ファブリックとシステム コントローラの間のインターフェイスを示します。

# 図 3-9. ファブリックとシステム コントローラの間のインターフェイス

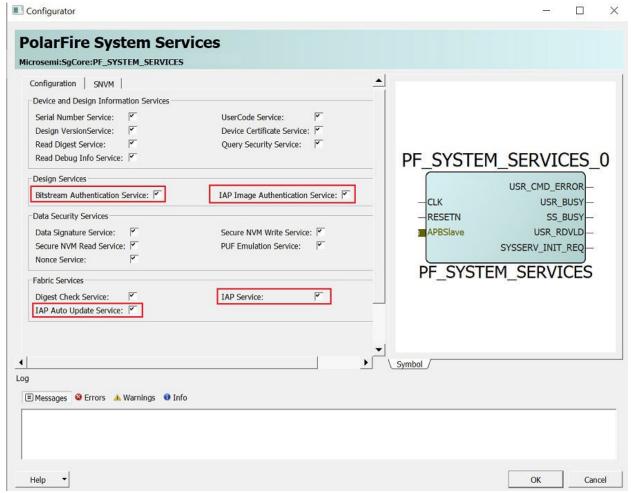


PolarFire FPGA システムサービス ドライバと SoftConsole サンプル プロジェクトに関する情報は、Libero SoC インストール パッケージに含まれる Firmware Catalog を参照してください。

#### 3.3.2.1.1 PolarFire システムサービス コンフィグレータ

図 3-10 に、PolarFire システムサービス コンフィグレータを示します。

#### 図 3-10. PolarFire FPGA コア システムサービス コンフィグレータ



ファブリック マスタは、APB インターフェイスを使って PF\_SYSTEM\_SERVICES コアに接続されます。 PF\_SYSTEM\_SERVICES コアは、Libero SoC 内の PolarFire システムサービス コンフィグレータを使って設定できます(図 3-10 参照)。詳細は『UG0848: PolarFire System Services User Guide』を参照してください。

#### 3.3.2.2 PolarFire SoC FPGA のシステムサービス

PolarFire SoC FPGA の場合、システムサービスは PolarFire SoC MSS によって開始されるシステム コントローラ動作です。MSS はシステム コントローラ ブリッジ(SCB) バスを介してシステム コントローラと連携します。図 3-11 に、MSS とシステム コントローラの間のインターフェイスを示します。

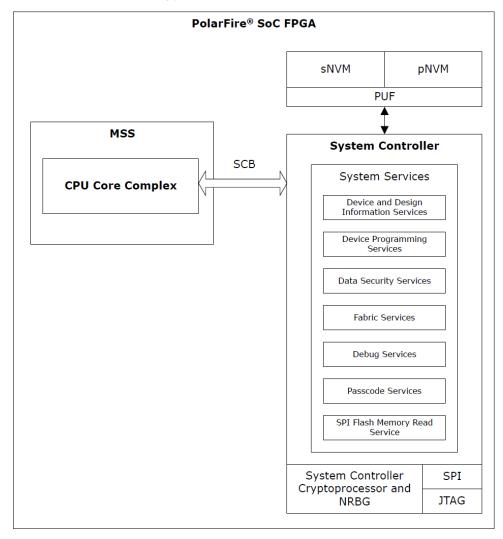


図 3-11. MSS とシステム コントローラの間のインターフェイス

PolarFire SoC FPGA MSS システムサービス ドライバと SoftConsole サンプル プロジェクトに関する情報は、GitHub を参照してください。

#### 3.3.2.3 システムサービス要求

PolarFire FPGA と PolarFire SoC FPGA は、16 ビットのシステムサービス ディスクリプタをシステム コントローラに渡す事により、システムサービスを要求します。ディスクリプタの下位 7 ビットは実行するサービスを指定し、上位 9 ビットはアドレス オフセットを指定します。システム コントローラへ渡される入力データとシステム コントローラから返される出力(サービス要求に対する応答)は、2 K バイトの内部メールボックス RAM メモリ空間を介して転送されます。メールボックス アドレスはサービスに固有のデータ構造(サービスへの追加入力、サービスからの追加出力)を指定します。サービスの完了時に、システム コントローラはサービスの正常な完了を示すステータス コードまたはエラーコードを書き込みます。表 3-7 に、システムサービス要求ディスクリプタのビットフィールドを示します。ファブリックからのメールボックス読み/書き通信に関する情報は、『UG0848 PolarFire System Services User Guide』を参照してください。

表 3-7. PolarFire FPGA および PolarFire SoC FPGA のシステムサービス要求ディスクリプタ

	·ステムサービス ディスク プタのビットフィールド	値	概要
15	5:7		メールボックス RAM 内のアドレス オフセットを指定します (最小 4 バイトのメモリにアクセス)。メールボックス アドレスは、ワードオフセット(0~511)を使って指定します。

続き				
システムサービス ディスク リプタのビットフィールド	値	概要		
6:0	SERVICECMD	システム コントローラに対して要求するサービスコマンドを指定します。		

システムサービスの詳細は、『PolarFire FPGA and PolarFire SoC FPGA System Services User Guide』を参照してください。

## 3.3.2.4 ビットストリームと IAP イメージの認証システムサービス

セキュリティと信頼性を確保するため、デバイスをプログラミングする前にプログラミング ビットストリームの認証と検証が必要です。認証に成功した場合、自動リカバリは発生しません。認証の実行中に、PolarFire FPGA 内のファブリック ユーザロジックおよび PolarFire SoC FPGA 内の MSS ユーザ アプリケーションは通常の動作(SPI フラッシュとシステム サービスへのアクセスを除く)を継続します。自動更新または IAP を使ってデバイスがプログラミングされる前に、ユーザ アプリケーションは認証システムサービスを実行できます。

**Note:** JTAG インターフェイスを介して新しいビットストリームをロードしている時にビットストリーム認証システムサービスが開始された場合、システムサービスが優先され、JTAG 動作は失敗します。

#### 3.3.2.4.1 ビットストリーム認証システムサービス

ビットストリーム認証システムサービスは、SPI フラッシュに保存されているビットストリームを解析し、そのインテグリティを検証します。表 3-8 に、ビットストリーム認証サービス要求向けのビットフィールドを示します。

#### 表 3-8. ビットストリーム認証サービス要求

システムサービス ディスクリ プタのビットフィールド	値	概要
15:7	MBOXADDR[10:2]	メールボックス アドレス (書式は表 3-9 参照)
6:0	23H	ビットストリーム認証コマンドコード

表 3-9 に、ビットストリーム認証サービス メールボックスの書式を示します。

#### 表 3-9. ビットストリーム認証サービス メールボックスの書式

オフセット	長さ(バイト)	パラメータ	方向	概要
0	4	SPIADDR		SPI フラッシュ内のビットストリームのアドレス 外部 SPI フラッシュ デバイスが 32 ビットアドレスをサポートしない場合、SPIADDR[31:24] は無視されます。

# 3.3.2.4.2 IAP イメージ認証システムサービス

IAP イメージ認証システムサービスは SPI フラッシュに保存されているイメージを解析し、イメージ ディスクリプタ、ビットストリーム、回路初期化データのインテグリティを検証します。

表 3-10 に、IAP イメージ認証サービス要求向けのビットフィールドを示します。

#### 表 3-10. IAP イメージ認証サービス要求

システムサービス ディスクリ プタのビットフィールド	値	概要
15	_	予約済み
14:7	IMAGEID[7:0]	SPI ディレクトリ内の認証が必要なイメージのインデックスを指定します。
6:0	(22H)です。	イメージ認証コマンドを指定します。

#### 3.3.2.4.3 認証サービスのステータスコード

ビットストリーム認証または IAP イメージ認証に成功した場合はステータスコードとして 0 が生成され、失敗した場 合は 8 ビットのエラーコードが生成されます。エラーコードの詳細は、12. 「補遺: エラーコード」を参照してくだ さい。

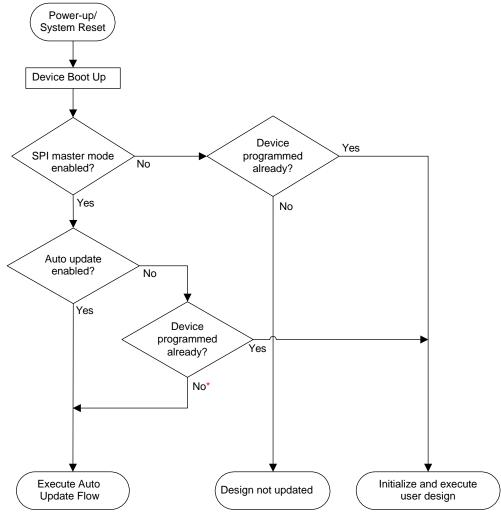
#### 3.3.2.4.4 認証システムサービスの使用法

プログラミング イメージはイメージ ディスクリプタ、ビットストリーム、回路初期化データ(オプション)を格納しま す。ビットストリーム認証システムサービスは、ビットストリームのみの認証用にしか使えません。これに対し、 IAP イメージ認証システムサービスは、上述の全てを含んだプログラミング イメージ全体の認証用に使えます。

#### 3.3.3 自動更新

自動更新を行うには、ユーザ回路内で自動更新機能を有効にする必要があります。電源投入時に、デバイスは SPI デ ィレクトリ内に保存されている最初の 2 つのイメージのうちのバージョンが新しい方のイメージを選択します。新し いバージョンのイメージがデバイスにプログラミング済みの現在のイメージと一致しない場合、自動更新が発生しま す。図 3-12 に、自動校正プログラミングの概略フローを示します。

#### 図 3-12. 自動更新の概略フロー

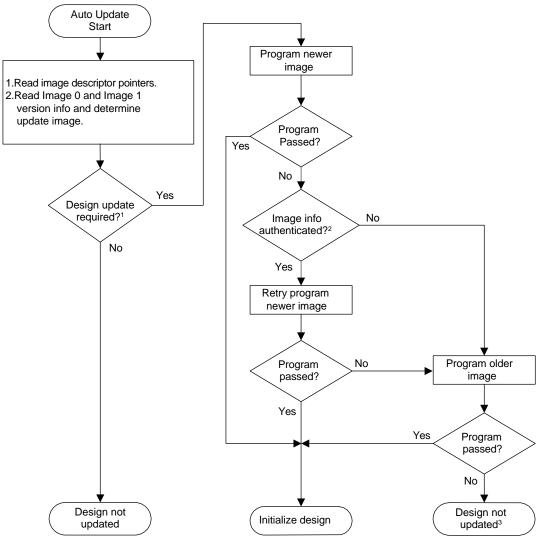


\*Different scenarios to reach here:

- Device is blank and auto update is initiated to program the device
- As part of IAP recovery when power fails during IAP or partially programmed with an invalid image
   As part of auto update recovery when power fails during auto update

図 3-13 に、自動更新プログラミングの詳細フローを示します。

# 図 3-13. 自動更新の詳細フロー



- Condition for update: version of the design differs from the update image or the device is blank.
   Device checks only BITS (starting bits of the bitstream) and AUTH (encryption keys information) components of the bitstream as part of the programming.
   The device is not programmed, and user intervention is required.

表 3-11 に、各種条件での自動更新の動作例を示します。

表 3-11. 各種条件での自動更新の動作例

デバイスで実行中の バージョン	SPI フラッシュ内の最初の 2 つのイメージのバージョン	バックレベル保護	自動更新用に選択される イメージ バージョン
ブランクデバイス	2, 3	無効	3
3	2, 3	無効	更新しない
3	1, 2	無効	2
2	1, 2	無効	更新しない
1	1, 2	無効	2
2	3, 4	有効、4に設定	更新しない
3	3, 5	有効、4に設定	5
2	3, 5	有効、4に設定	5
5	2, 3	有効、4に設定	更新しない

# 3.3.3.1 ブランクデバイスに対する自動更新(自動プログラミング)

ブランクデバイスの場合、SPI マスタモードを有効にすると、電源投入またはリセット時にデバイスは最新バージョンのイメージを使って自動的にプログラミングされます。このプロセスを「自動プログラミング」と呼びます。

自動プログラミングを使ってブランクデバイスにセキュリティを有効にしたビットストリームがプログラミングされた場合、その後のプログラミングにはカスタム セキュリティを有効にしたビットストリーム ファイル(UEK1/UEK2 鍵を使用)のみが使えます。セキュリティ有効ビットストリームの生成については、1.4. 「ビットストリームにユーザ セキュリティ設定を追加する」を参照してください。

## 3.3.3.2 プログラミング済みデバイスに対する自動更新

プログラミング済みデバイスでは、自動更新はシステムサービスを介して実行されます。デバイスがプログラミング済みである場合、更新イメージとプログラミング済みイメージのバージョンが比較されます。更新イメージのバージョンがプログラミング済みイメージのバージョンと異なる場合、自動更新プログラミングが開始されます。

プログラミング済みデバイスに対して自動更新を実行するには、ユーザ アプリケーションからシステムサービスを要求する必要があります。このサービス要求に応じて、システム コントローラはデバイスをプログラミングします。

以下の状況では、ユーザ アプリケーションはステータスコードを得る事ができません。

- 自動更新プログラミングが成功した時にデバイスが自動的に再起動し、新しいバージョンの回路を初期化した場合
- 自動更新プログラミングに失敗した時に、自動リカバリ機能が有効なイメージを使ってデバイスの再プログラミングを試みた場合

表 3-12 に、自動更新システムサービス要求向けのビットフィールドを示します。

#### 表 3-12. 自動更新システムサービス要求

<u>X • :=: 1                                </u>			
	システムサービス ディスク リプタのビットフィールド	値	概要
	15:7		予約済み
	6:0	46H	自動更新プログラミング コマンド

ユーザ回路で自動更新が有効にされていない場合、自動更新システムサービスを使ってデバイスのイメージを最新バ ージョンに更新する事はできません。

Note: SPI コントローラがマスタモードではない場合、自動更新システムサービスはエラーコードを生成しません。

# 3.3.3.3 自動更新プログラミング失敗からのリカバリ

自動更新プログラミング中に停電が発生した場合、次のブートサイクルで自動更新プログラミング フローが開始されてデバイスに最新イメージがプログラミングされます。

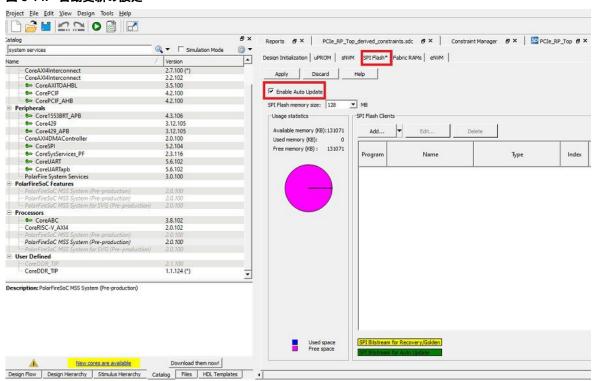
Note: 新しいイメージのプログラミングに失敗した場合、1度だけ再試行されます。それでも失敗した場合、旧バージョンのイメージがプログラミングされます。自動更新の終了後にデバイスがブランクのままであった場合、外部から失敗が判断できないため、ユーザの介入が必要です。

# 3.3.3.4 ユーザ回路での自動更新オプションの有効化

自動更新を有効にする手順は以下の通りです。

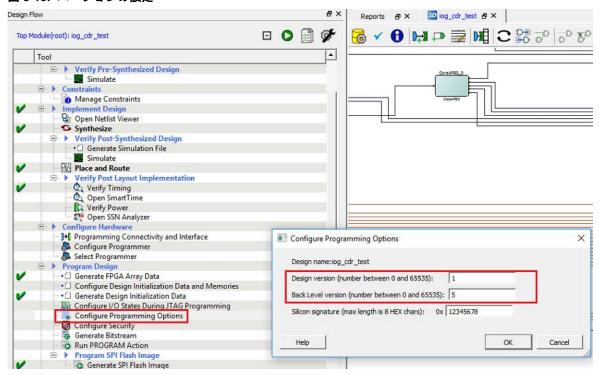
- 1. [Configure Design Initialization Data and Memories]をクリックし、[SPI Flash]タブを選択します。
- 2. [Enable Auto Update]チェックボックスを選択します。

#### 図 3-14. 自動更新の設定



3. [Configure Programming Options]をクリックし、[Desgn version]と[Back Level version]を指定します(図 3-15 参照)。

#### 図 3-15. バージョンの設定



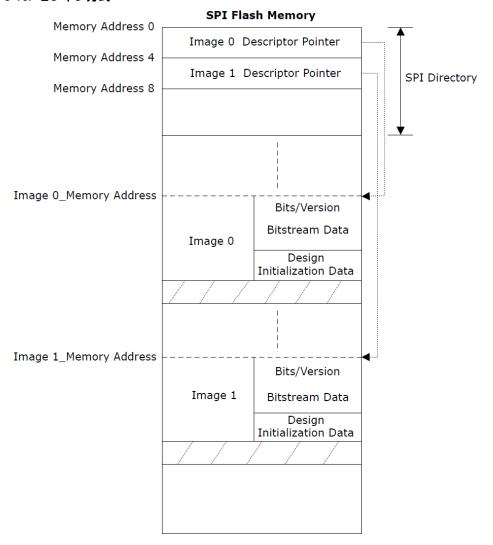
#### 3.3.3.5 自動更新の使用法

自動更新は、SPI フラッシュメモリ内にデバイス上の現在のバージョンとは異なるプログラミング イメージが存在する場合に開始されます。詳細は 10.2. 「SPI ディレクトリ」を参照してください。デバイスは、プログラミング イメージ内の Bits/Version 情報(図 3-16 参照)によりバージョンを特定します。Bits/Version 情報はビットストリームの先頭で提供されます。以下では、自動更新の 3 通りの使用法 (ピンポン、ゴールデン イメージ、シングルイメージ) について説明します。設計要件に応じてこれらの使用法を使い分けます。

#### ピンポン方式

自動更新は、SPI フラッシュメモリ内の最初の 2 つのイメージのうちの新しい方を使います。SPI フラッシュメモリに新しいイメージが追加された場合、既存の 2 つのイメージのうちの古い方が追加された新しいイメージで上書きされます。これを「ピンポン方式」と呼びます。この方式は、1 つ前のイメージ バージョンを最新イメージと一緒に保存しておく必要がある場合に使います。この方式を使うと、新しいイメージで問題が発生した場合に自動的に以前のイメージ戻す事ができます。図 3-16 に、ピンポン方式のイメージ配置を示します。

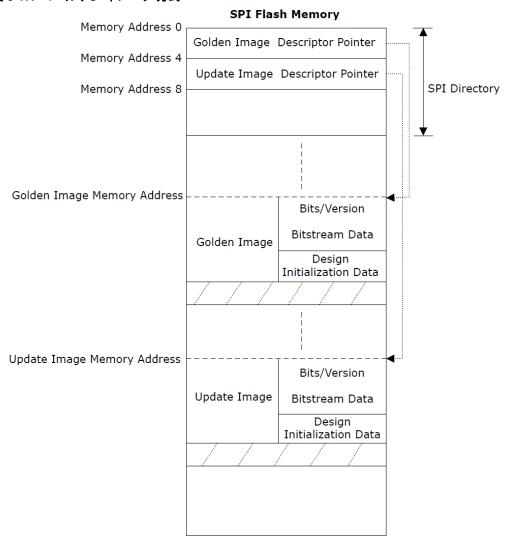
# 図 3-16. ピンポン方式



# ゴールデン イメージ方式

新しいバージョンのイメージで自動更新に失敗した場合、確実に動作可能なイメージ (これを「ゴールデン イメージ」と呼ぶ) をデバイスにプログラミングします。SPI フラッシュメモリに新しいイメージが追加された時にゴールデン イメージが上書きされてはいけません。図 3-17 に、ゴールデン イメージ方式のイメージ配置を示します。

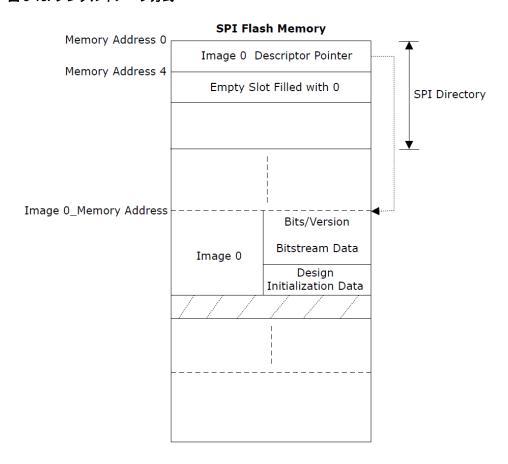
### 図 3-17. ゴールデン イメージ 方式



### シングルイメージ方式

これは、デバイスの更新用に使えるイメージが 1 つしか存在しない場合の使用法です。図 3-18 に、シングルイメージ方式のイメージ配置を示します。

#### 図 3-18. シングルイメージ方式



#### 3.3.4 IAP

IAP は、指定されたプログラミング イメージを使ってデバイスを再プログラミングします。IAP の場合、プログラミング イメージは、指定されたイメージ インデックスまたは SPI イメージアドレスに基づいて(イメージのバージョンとは無関係に)選択されます。PolarFire FPGA 内のファブリック ユーザロジックおよび PolarFire SoC FPGA 内のユーザ アプリケーションはプログラミング イメージを指定し、IAP システムサービスを使ってデバイスの再プログラミングを開始します。

#### 3.3.4.1 システムサービスによる IAP の実行

ユーザ アプリケーションは、PolarFire FPGA 内のファブリック ユーザロジックまたは PolarFire SoC FPGA 内の MSS ユーザ アプリケーションを使って、IAP システムサービスを要求します。システムサービスは、そのイメージを検証またはプログラミングのどちらの用途で使うのか指定します。システム コントローラは、SPI フラッシュから自動的にビットストリームを読み出して、デバイスの内容の検証またはプログラミングを行います。

#### IAP 検証動作

検証動作は、指定されたプログラミング イメージの内容とデバイスの内容を比較します。表 3-13 に、イメージ インデックスを指定して IAP の検証システムサービスを要求する場合のビットフィールドを示します。

表 3-13. イメージ インデックス指定による IAP 検証要求

システムサービス ディスク リプタのビットフィールド	値	概要
15	_	予約済み
14:7	SPI_IDX[7:0]	IAP 動作向けに SPI ディレクトリ内のイメージ インデック スを指定します。
6:0	44H	IAP 検証動作

SPI ディレクトリ内のイメージ インデックスの代わりに SPI フラッシュメモリ アドレスを指定する事もできます (表 3-14 参照)。

#### 表 3-14. イメージアドレス指定による IAP 検証要求

	システムサービス ディスク リプタのビットフィールド	値	概要
4	15:7	MBOXADDR[10:2]	メールボックス アドレス (書式は表 3-17 参照)
6	5:0	45H	IAP 検証動作

IAP 検証に成功すると、ステータスコードとして 0 が生成されます。失敗した場合は 8 ビットのエラーコードが生成されます。エラーコードの詳細は、12. 「補遺: エラーコード」を参照してください。

デバイスの内容のインテグリティを検証するには、IAP 検証動作の代わりに Digest Check システムサービスを使う事を推奨します。詳細は、『PolarFire FPGA Datasheet』または『PolarFire SoC AdvanceDatasresheet』を参照してください。

Note: プログラミング中に出力されるダイジェスト(\*.digest file の内容と同じ) は、ビットストリーム ペイロード ダイジェストです。このダイジェストにより、デバイスが正しいビットストリーム ペイロードを受信した事を確認できます。DEVICE\_INFO からエクスポートされるダイジェストは、実際のメモリ内容のダイジェストです。これは、暗号化されたビットストリーム ペイロードに含まれるその他のメタデータを含まないため、プログラミング中に生成されるダイジェストとは異なります。

#### IAP プログラミング動作

プログラミング動作により、指定されたプログラミング イメージを使ってデバイスの内容が更新されます。IAP プログラミング動作は、プログラミング実行前にイメージを認証しません。イメージは、IAP イメージ認証システムサービスを使って認証できます。詳細は、3.3.2.4.2. 「IAP イメージ認証システムサービス」を参照してください。

以下の状況では、ユーザ アプリケーションはステータスコードを得る事ができません。

- IAP が成功した時にデバイスが自動的に再起動し、新しい回路を初期化した場合
- IAP が失敗した時に IAP リカバリプロセスがイメージ 0 を使ってデバイスのプログラミングを試みた場合

**Note:** IAP リカバリプロセスは、SPI ディレクトリ内のイメージ 0 へのポインタの次のスロット(Addr 4)が Null である時に、イメージ 0 を使ってデバイスのプログラミングを試みます。詳細は 10.2. 「SPI ディレクトリ」を参照してください。

表 3-15 に、イメージ インデックスを指定して IAP のプログラミング システムサービスを要求する場合のビットフィールドを示します。

表 3-15. イメージ インデックス指定による IAP プログラミング要求

システムサービス ディスク リプタのビットフィールド	値	概要
15	_	予約済み
14:7	SPI_IDX[7:0]	IAP 動作向けに SPI ディレクトリ内のイメージ インデックス を指定します。
6:0	42H	IAP プログラミング動作

SPI ディレクトリ内のイメージ インデックスの代わりに SPI フラッシュメモリ アドレスを指定する事もできます (表 3-16 参照)。

#### 表 3-16. イメージアドレス指定による IAP プログラミング要求

システムサービス ディスク リプタのビットフィールド	値	概要
15:7	MBOXADDR[10:2]	メールボックス アドレス (書式は表 3-17 参照)
6:0	43H	IAP プログラミング動作

表 3-17 に、メールボックスの書式を示します。

### 表 3-17. メールボックスの書式

オフセット	長さ(バイト)	パラメータ	方向	概要
0	4	SPIADDR	入力	SPI フラッシュメモリ内のプログラミング イメージのアドレス
				接続した SPI フラッシュ デバイスが 32 ビットアドレスをサポートしない場合、SPIADDR[31:24]は無視されます。

#### 3.3.4.2 プログラミング失敗時のリカバリ動作

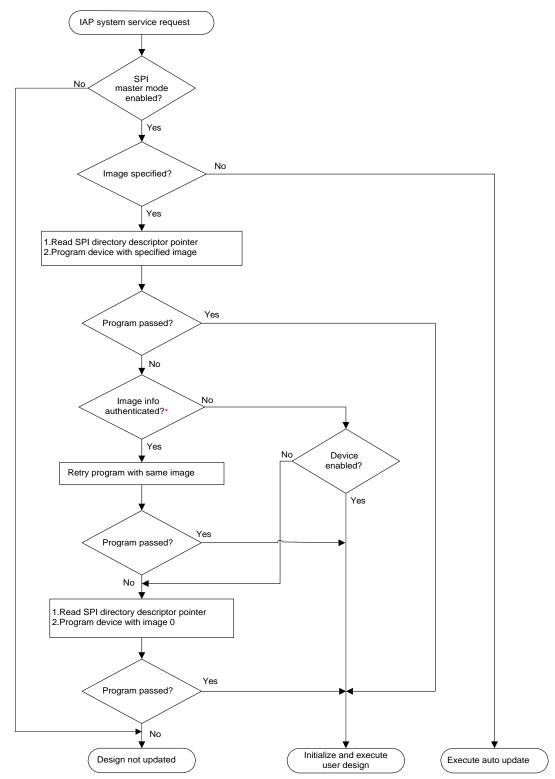
IAP の実行中に停電が発生すると、デバイスはイメージ 0 を使ってプログラミングします。

**Note**: 指定されたイメージのプログラミングに失敗した場合、1 度だけ再試行されます。それでも失敗した場合、イメージ 0 がプログラミングされます。IAP の終了時にデバイスがブランクのままであった場合、I/O から何の情報も得られず、ユーザの介入が必要です。

### 3.3.4.3 IAP フロー

図 3-19 に、IAP フローを示します。

#### 図 3-19. IAP フロー

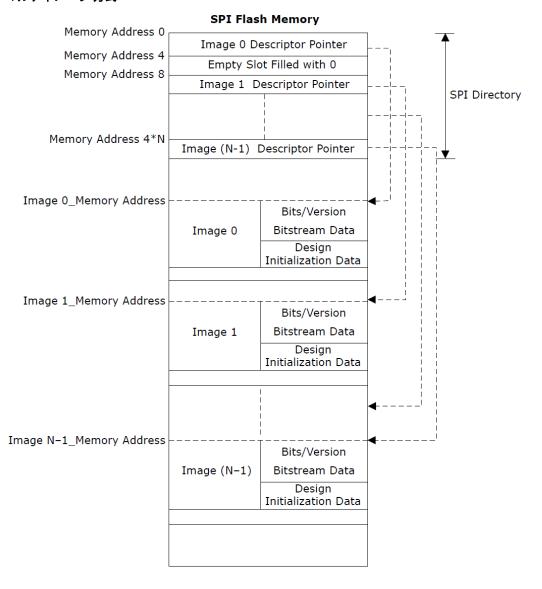


<sup>\*</sup> Device checks only BITS (starting bits of the bitstream) and AUTH (encryption keys information) components of the bitstream as part of the programming.

#### 3.3.4.4 IAP の使用法

どちらのデバイスファミリも、マルチイメージ IAP 方式(最大 255 個のイメージを SPI フラッシュメモリに保存可能)をサポートします。イメージ ディスクリプタ ポインタは、SPI フラッシュメモリのセクタ 0 に格納されます。SPI フラッシュに保存された任意のイメージを使ってデバイスをプログラミングでき、プログラミングに失敗した場合はイメージ 0 が使われます。イメージ 0 ポインタの次のスロットは Null (空のスロット)である事が必要です。マルチイメージ方式は、複数のイメージを用意しておき、そのうちの 1 つを指定してデバイスを更新する場合に使います。図 3-20 に、マルチイメージ方式のイメージ配置を示します。

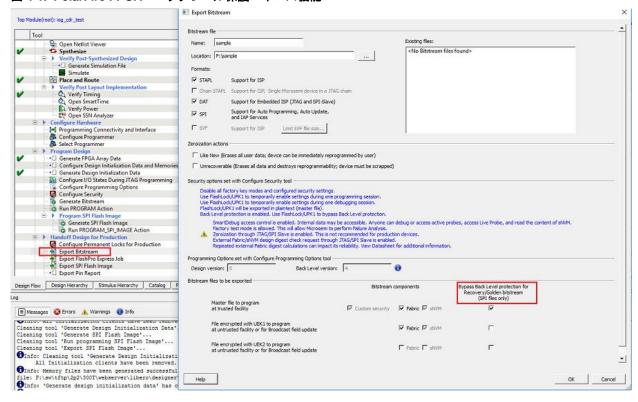
#### 図 3-20. マルチイメージ 方式



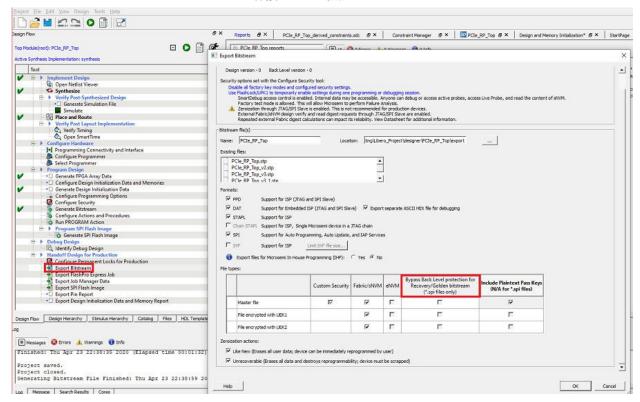
## 4. バックレベル保護のバイパス

Configure Security ツールでバック レベル保護が有効になっている場合、Libero を使ってビットストリームをエクスポートする際に、SPI ビットストリームのバック レベル保護をバイパスできます。プログラミング リカバリ エラーを回避するには、[Bypass the Back Level Protection for Recovery/Golden bitstream (SPI files only)] (図 4-1、図 4-2 参照)を有効にします。

#### 図 4-1. PolarFire FPGA - バックレベル保護バイパス機能

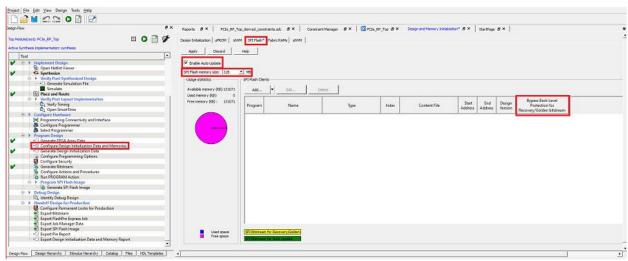


#### 図 4-2. PolarFire SoC FPGA - バックレベル保護バイパス機能



回路/メ モ リ 初期化デー タ を使って SPI ビ ッ トストリーム を SPI フ ラ ッ シュ に追加すると 、 Libero はバ ッ クレ ベル保護バ イパス機能のステータスを表示します(図 4-3 参照)。

#### 図 4-3. バックレベル保護パイパス機能のステータス



### 4.1 バックレベル保護バイパス機能が必要になる事例

表 4-1 に、バックレベル保護バイパス機能が必要となる事例を示します。

#### 表 4-1. バックレベル保護パイパス機能が必要になる事例

ステップ	SPI ビット ストリーム	動作	結果	ビット ストリーム バージョン	ビット ストリーム バックレベル パージョン	デバイス バックレベル バージョン
1	ゴールデン/リカバリ	自動プログラミング	成功	2	1	1
2	IAP/更新 ビットストリーム	自動更新 IAP	成功	3	2	2
3	IAP/更新 ビットストリーム	自動更新 IAP	失敗、 プログラミング リカバリを試行	4	無効	2

#### 上表の各ステップは以下の通りです。

- デバイスはビットストリーム バージョン = 2/ バックレベル バージョン =1 でデバイスをプログラミングします。 現在のデバイス バックレベル バージョンは 1 に設定されます。
- 次にデバイスはビットストリーム バージョン = 3/ バックレベル バージョン =2 でデバイスをプログラミングします。現在のデバイス バックレベル バージョンは 2 に設定されます。
- 3. デバイスはバージョン 4 のビットストリームを使って更新を試みますが、失敗します。この場合、デバイスは バージョン 2 のゴールデン/リカバリ ビットストリームを使ってリカバリを試みます。しかし、現在のデバイス バックレベル保護がバージョン 2 に設定されている(すなわち、ゴールデン/リカバリ ビットストリーム バージョンはデバイス バックレベル バージョンと等しい) ため、リカバリにも失敗します。バックレベル保護が原因でプログラミング リカバリに失敗する場合、ゴールデン/リカバリ ビットストリームに対してバックレベル保護バイパス機能(図 4-1 内の[Bypass Back Level protection for Golden bitstream])を有効にする事でリカバリの失敗を回避できます。

## 5. プログラミング中の I/O ステート

表 5-1 に、プログラミング中に適用される I/O ステートの一覧を示します。

表 5-1. 各種プログラミング モードの I/O ステート

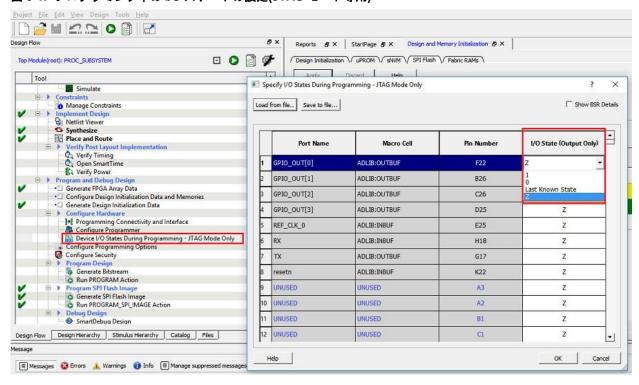
I/O タイプ	1/0 ステート		
	JTAG プログラミング	SPI スレーブ プログラミング	SPI マスタ プログラミング(IAP/自動更新)
システム コント ローラ I/O	有効	有効	有効
XCVR 参照クロッ ク入力	影響なし	影響なし	影響なし ループバック モードを使って IAP 中に動作を維持する事で、XCVR リ ンクをアクティブに保持可能
XCVR データ I/O	バウンダリ スキャンセルによ り設定	影響なし	影響なし ループバック モードを使って IAP 中に動作を維持する事で、XCVR リ ンクをアクティブに保持可能
GPIO、HSIO	I/O は有効だが、I/O ステート はバウンダリ スキャンセルを 使って設定可能	SPI スレーブ命令 ISC_ENABLE を使って Weak プルアップが可能	出力は3ステートにされ、Weak プルアップが適用される
MSS I/O (PolarFire SoC のみ)	I/O は有効だが、I/O ステート はバウンダリ スキャンセルを 使って設定可能	SPI スレーブ命令 ISC_ENABLE を使って Weak プルアップが可能	出力は3ステートにされ、Weak プルアップは適用されない

JTAG プログラミングの前に Libero SoC 内で I/O ステートを設定する事で、JTAG プログラミング中にこれらの I/O のステートを設定した状態に保持できます。I/O 出力ステートは以下の通りに設定できます。

- 1: I/O を論理 HIGH に設定します。
- 0: I/O を論理 LOW に設定します。
- Last Known State: プログラミング モードを開始する直前の I/O 状態をプログラミング中に保持します。
- Z: I/O を 3 ステートにします。

プログラミング中の I/O 出力ステートの設定方法を図 5-1 に示します。

#### 図 5-1. プログラミング中の I/O ステートの設定(JTAG モード専用)



6. プログラミング中の MSS ステート(PolarFire SoC FPGA の	み)
--	----

将来の版で更新されます。

## 7. プログラミングに関する推奨事項

プログラミングに関する推奨ガイドラインを以下に記載します。

- デバイスをプログラミングする前にビットストリームを認証する。
- プログラミング中にリセットピン(DEVRST\_N)をアサートしない(デバイス設定を破損する可能性あり)。
- 選択したプログラミングモードに適した設定とプログラミングインターフェイスを使う。
- JTAG プログラミングの前に設計要件に基づいてデバイス I/O ステートを設定する。詳細は 5. 「プログラミング中の I/O ステート」を参照してください。

# 8. プログラミング中のブラウンアウト

ブラウンアウトは、電源電圧が推奨レベルよりも低下した時に発生します。プログラミング中にブラウンアウトが発生すると、デバイスは自動的にプログラミング エラーから回復し(リカバリ機能は既定値により有効)、外部 SPI フラッシュに保存されている有効なプログラミング イメージを使ってプログラミングを実行します。

## 9. ゼロ化

どちらのデバイスファミリも、任意または全ての設定用ストレージをユーザ設定に従ってゼロ化(クリアおよび検証) する機能を内蔵しています。ゼロ化により、内部揮発性メモリ(LSRAM、 $\mu$ SRAM、システム コントローラ RAM 等) がクリアおよび検証されます。ゼロ化が完了した後に、JTAG/SPI スレーブ命令を使ってゼロ化証明を取得する事で、ゼロ化処理が正しく実行された事を確認できます。ゼロ化の詳細は、『PolarFire FPGA and PolarFire SoC FPGA Security User Guide』を参照してください。

### 10. 外部 SPI フラッシュのプログラミング

IAP および自動更新には外部 SPI フラッシュメモリが必要です。SPI フラッシュメモリはシステム コントローラの SPI と連携してプログラミング イメージを保存します。

SPI フラッシュメモリは複数のセクタに分割されます。最初のセクタ(セクタ 0) の 1 KB メモリは SPI ディレクトリとして使われ、プログラミング イメージのインデックス(ディスクリプタ ポインタ)を格納します。その他のセクタは、プログラミング イメージの保存用に使われます。

#### 10.1 サポートする SPI フラッシュデバイス

各社の SPI フラッシュ デバイスは、読み出し動作向けに標準命令セットを実装しています。システム コントローラファームウェアは、以下のコマンドを実行する事によりアドレス指定モード(3 バイトまたは 4 バイト)を識別します。

READ SERIAL FLASH DISCOVERY PARAMETER (5AH)

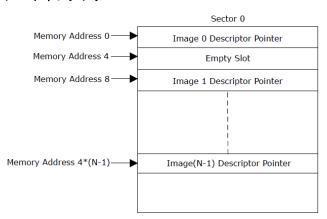
システム コントローラは Micron、Winbond、Macronix、Spansion 社製のデバイスをサポートします。JESD216 規格に準拠するその他のデバイスも使える場合があります。JESD216 に非準拠のデバイスであっても、3 バイトアドレス指定の FAST READ (OBH)コマンドをサポートしていれば、使える可能性があります。そのようなデバイスでは、フラッシュメモリの最初の 128 Mbit しか使えません。

### 10.2 SPI ディレクトリ

SPI ディレクトリは、各プログラミング イメージの先頭位置を指し示すイメージ ディスクリプタ ポインタを格納します。各ポインタは 4 バイトを使います。SPI フラッシュメモリ デバイスが 3 バイトアドレス指定モードのみサポートする場合、最初の 3 バイトが使われます。

電源投入時にイメージ 0 を選択する IAP リカバリ機能を使う場合、イメージ 0 ポインタの次のスロットは Null (空スロット)である事が必要です。そうではない場合、自動更新が選択されます。図 10-1 に、プログラミング イメージ ディスクリプタ ポインタを格納した SPI フラッシュ ディレクトリを示します。

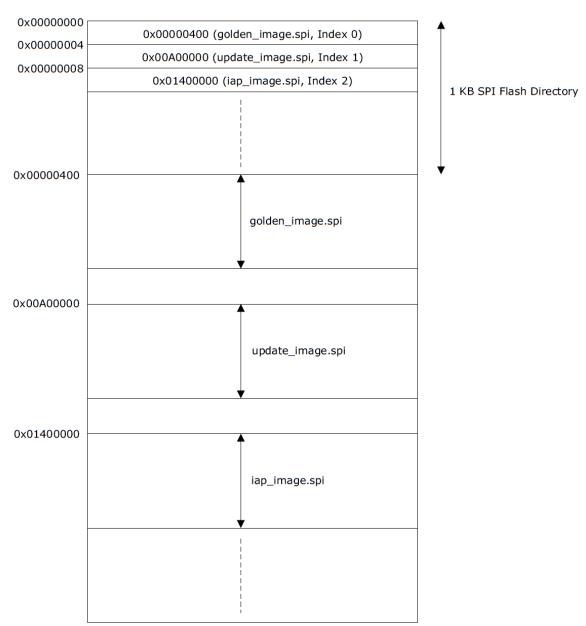
#### 図 10-1. SPI フラッシュ ディレクトリ



SPI ディレクトリは、各プログラミング イメージの開始アドレスを格納します。SPI ディレクトリは、外部 SPI フラッシュメモリのセクタ 0 から 1 KB の領域を使います。例えば、外部 SPI フラッシュメモリが 3 つのイメージ (ゴールデン イメージ、更新イメージ、IAP イメージ)を格納している場合、これらのイメージはそれぞれ開始アドレス 0x400、0xA00000、0x1400000 に保存されます。Libero コンフィグレータを使ってプログラミング イメージを外部 SPI フラッシュに書き込む場合、SPI ディレクトリの内容は自動的に書き込まれます。ユーザ アプリケーションを使ってプログラミング イメージを外部 SPI フラッシュに書き込む場合、ユーザ アプリケーションは各イメージの開始アドレスをアドレス 0 から始まる SPI ディレクトリに書き込む必要があります(図 10-2 参照)。

User Guide

図 10-2. SPI フラッシュメモリ



### 10.3 SPI フラッシュ のプログラミング方法

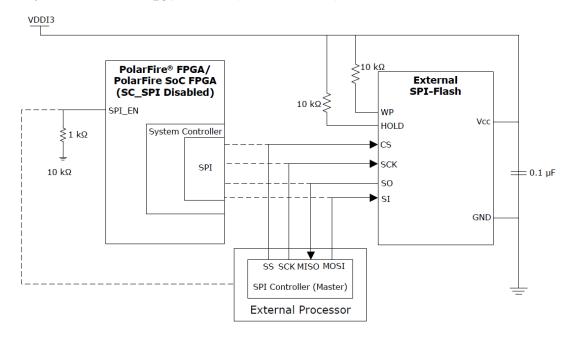
外部 SPI フラッシュは、JTAG またはシステム コントローラの SPI を使ってプログラミングできます。システム コントローラの SPI が有効かつ SPI マスタモードに設定されている場合、システム コントローラの SPI ポートはシステム コントローラと FPGA ファブリック マスタ/MSS (PolarFire SoC FPGA のみ)または JTAG の間で共有されます。以下では、外部 SPI フラッシュのプログラミング向けのユースモデルについて説明します。

#### 10.3.1 外部プロセッサを使って SPI フラッシュをプログラミングする

SPI\_EN ピンが無効(Low)の時にシステム コントローラの SPI 出力は 3 ステートになり、外部プロセッサから SPI ピンを駆動して SPI フラッシュに書き込む事が可能となります。この場合、システム コントローラおよびファブリック /MSS (PolarFire SoC FPGA のみ)から SPI インターフェイスを駆動する事はできません。外部プロセッサは、SPI\_EN ピンを Low に駆動する事により外部 SPI フラッシュに書き込む事ができます。

SPI\_EN ピンは、基板上のジャンパを使ってデバイスの外部で設定する事もできます。その場合、SPI フラッシュは外部プロセッサの SPI マスタ SCK に同期して書き込まれます。SCK の周波数は、外部プロセッサ アプリケーションを使って設定します。図 10-3 に、外部プロセッサを使って SPI フラッシュをプログラミングする場合の接続を示します。

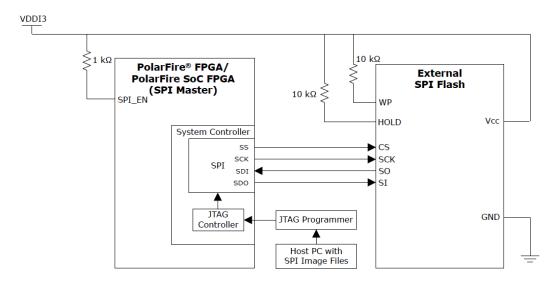
#### 図 10-3. 外部マイクロプロセッサを使った SPI フラッシュのプログラミング



#### 10.3.2 JTAG を使って SPI フラッシュをプログラミングする

外部 SPI フラッシュは、FlashPro プログラマ(バージョン 5 以上)を使ってシステム コントローラの JTAG インターフェイス経由でプログラミングできます。JTAG コントローラは、システム コントローラを介して外部 SPI フラッシュと連携するために、特殊な JTAG 命令(SPIPROG (IR=0xb0))を使います。どちらのデバイスファミリでも、JTAG コントローラはシステム コントローラの SPI 出力を直接駆動するためにこの命令をサポートします。図 10-4 に、JTAGを使って SPI フラッシュをプログラミングする場合の接続を示します。

#### 図 10-4. JTAG を使った SPI フラッシュのプログラミング



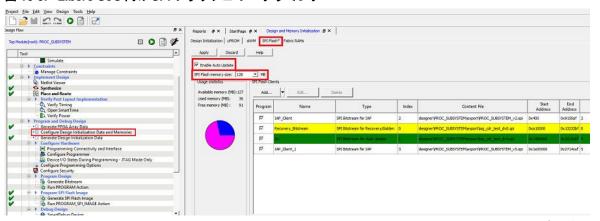
**Note**: FlashPro5 は、製品番号が MT25QL01GB の Micron 社製デバイスのみをサポートします。その他の SPI フラッシュ デバイスはサポートされません。FlashPro6 を使った場合にサポートされる SPI フラッシュメモリに関しては、ウェブページ(FlashPro and FlashPro Express Device Support)を参照してください。

#### 10.3.2.1 Lebero を使って SPI フラッシュをプログラミングする

Libero SoC ソフトウェアを使って外部 SPI フラッシュメモリにプログラミング イメージを書き込む事ができます。 SPI フラッシュメモリのプログラミング手順は以下の通りです。

1. [Design Flow] > [Program and Debug Design] > [Configure Design Initialization Data and Memories]と操作して[SPI Flash]タブを開きます(図 10-5 参照)。

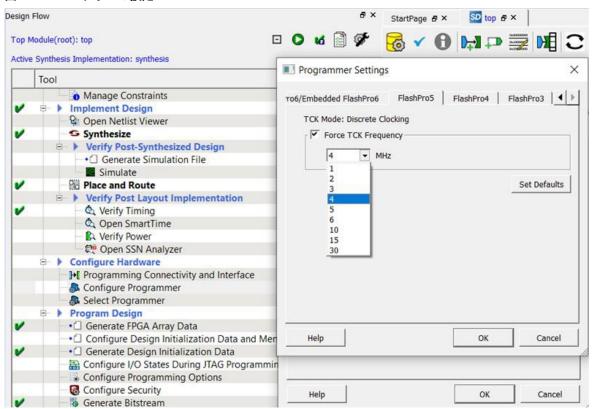
#### 図 10-5. Libero SoC 内の SPI フラッシュ プログラミング



**Note:** PolarFire FPGA の場合、FlashPro6 (Libero SoC v12.4 以上で対応)を使った SPI フラッシュ プログラミング サポートを簡素化するため、ベンダー情報はターゲットメモリの容量で置き換えられます。

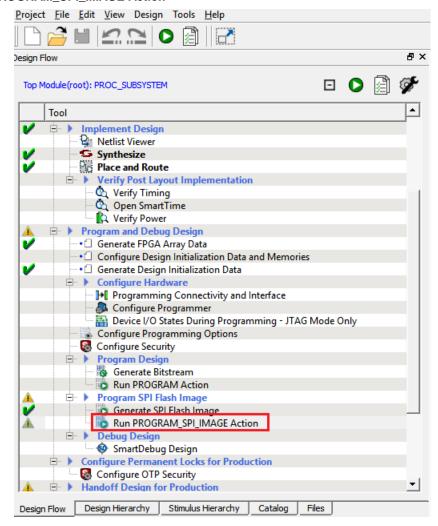
- 2. 「SPI Flash Clients」枠内で、必要なプログラミング イメージを追加した後に、[Apply]をクリックします。 [Help]をクリックすると、入力フィールドの値に関する説明が表示されます。
- [Design Flow] > [Configure Hardware] > [Configure Programmer] と操作し、[Flash Pro]タブ内で右クリックして[Programmer Settings]を選択します。[Force TCK Frequency]チェックボックスを有効にして TCK 周波数を選択する事で、SPI フラッシュの書き込み速度を調整できます。

#### 図 10-6. プログラマの設定



4. [Run PROGRAM\_SPI\_IMAGE Action]をダブルクリックする事で、SPI フラッシュに SPI ディレクトリとプログラミング イメージを書き込みます。

#### 図 10-7. Run PROGRAM\_SPI\_IMAGE Action



回路/メモリ初期化の詳細は、『PolarFire FPGA and PolarFire SoC FPGA Device Power-up and Reset User Guide』を参照してください。

Note: Libero を使った SPI フラッシュのプログラミングには、以下を推奨します。

- このツールはプログラミングの前に SPI フラッシュを消去します。このため、Libero SoC 以外のプログラミング 方法で SPI フラッシュに他のデータを書き込む必要がある場合は、最初に Libero を使って SPI フラッシュに書 き込む事を推奨します。そうしないと、先に書き込んだデータは消去されてしまいます。
- SPI フラッシュの部分的な更新は現在サポートしていません。
- SPI フラッシュ内のクライアントとクライアントの間のギャップを大きくしない事を推奨します。ギャップには「1」が書き込まれるため、プログラミング時間が増加します。
- **10.3.3** MSS を使って SPI フラッシュをプログラミングする(PolarFire SoC FPGA のみ) 将来の版で更新されます。
- 10.3.4 外部 SPI フラッシュの内容を MSS ユーザ アプリケーション メモリにコピーする(PolarFire SoC FPGA のみ) MSS SPI コピー システムサービスを使うと、外部 SPI フラッシュから MSS ユーザ アプリケーション メモリヘデー タをコピーできます。mss\_system\_services ドライバは、外部 SPI フラッシュから MSS ユーザ アプリケーション メモリヘデータをコピーするための機能を備えています。

mss\_system\_services ドライバと SoftConsole サンプル プロジェクトについては、GitHub を参照してください。この MSS SPI コピー システムサービスは、外部 SPI フラッシュメモリからの読み出し用にのみ使えます。

#### 10.3.5 ファブリック ユーザロジックを使って SPI フラッシュをプログラミングする

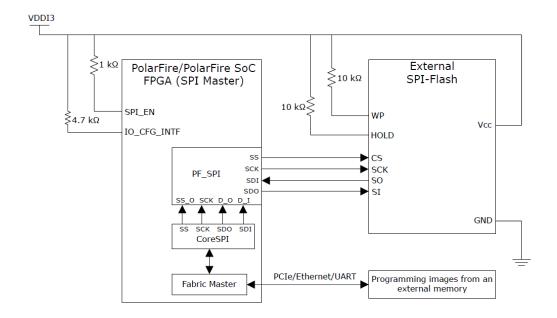
システム コントローラの SPI が有効かつマスタとして設定されている場合、デバイスの電源投入後にシステム コントローラは SPI の制御権をファブリックに渡します。システムサービス要求による SPI フラッシュへのアクセスや JTAG コントローラからのプログラミングのための SPI フラッシュ アクセス制御は、ファブリック側に引き継ぐ事ができます。

ファブリック ユーザロジックは、外部メモリソースからプログラミング イメージを取得します(図 10-8 参照)。ファブリック ユーザロジックは、Libero カタログ内で提供される CoreSPIcontroller と PF\_SPI macro を使って外部 SPI フラッシュにアクセスします。外部 SPI フラッシュは、SPI マスタ SCK クロックに同期して書き込まれます。SCK 周波数はユーザロジック内で設定できます。

システムコントローラは、専用 SPI I/O (SPI インターフェイス ピン) にのみアクセスできます。システム コントローラはファブリック I/O にアクセスできません。このため、システム コントローラからの SPI を使った全てのサービス (すなわちプログラミング) は、専用 SPI I/O しか使えません。ユーザは、Libero カタログ内で提供される PF\_SPI マクロを使う事で、ファブリックから専用 SPI I/O を介して SPI フラッシュメモリにアクセスできます(これはシステムコントローラが専用 SPI I/O を解放した後に可能となります)。

**Note:** プログラミング イメージをフェッチして外部 SPI フラッシュに書き込むには、デバイスファミリによらず、SPI 制御回路や書き込みイメージフェッチ回路を予めプログラミングしておく必要があります。詳細は、『AC466: PolarFire FPGA Auto Update and In-Application Programming Application Note』を参照してください。

#### 図 10-8. ファブリック ユーザロジックを使った SPI フラッシュのプログラミング



## 11. システム コントローラ サスペンドモード

PolarFire FPGA および PolarFire SoC FPGA デバイスはシステム コントローラ サスペンドモードを備えており、このモードを使ってデバイス初期化の完了後もシステム コントローラをリセット状態に保持できます。セーフティ クリティカル アプリケーションでは、シングルイベント アップセット(SEU) による誤ったプログラミングまたはゼロ化からデバイスを保護するために、このモードが不可欠です。ユーザが Libero SoC ツールを使ってシステム コントローラをサスペンドモードに設定した場合、一部のデバイス プログラミング オプションは無効になり、その他のオプションは JTAG\_TRST\_B ピンを制御する事により有効または無効にできます。表 11-1 に、システム コントローラサスペンドモードの設定(有効/無効)と利用可能プログラミング モードの対応を示します。サスペンドモード中に利用可能なデバイス機能の完全なリストは、『PolarFire FPGA and PolarFire SoC FPGA System Services User Guide』を参照してください。

表 11-1. システム コントローラ サスペンドモード中に利用可能なプログラミング モード (PolarFire FPGA/PolarFire SoC FPGA)

(1 otal 1 o 1 o o o o o o o o o o o o o o o o				
プログラミング	システム コントローラ	ラ サスペンドモード	概要	
モード	有効	無効		
JTAG	Yes <sup>1</sup>	Yes	_	
SPIスレーブ	Yes <sup>1</sup>	Yes	_	
自動更新 (POR/DEVRSTn)	Yes	Yes	自動更新機能を有効にした場合、電源投入後またはデバイスリセット(DEVRSTn) 後にプログラミングが実行されます。	
自動更新 (システムサービス)	No	Yes	サスペンドモード中は、システムサービスを介 して自動更新を要求する事はできません。	
IAP	No	Yes	_	

<sup>&</sup>lt;sup>1</sup>動作を開始する前に JTAG\_TRST\_B ピンを論理 High に駆動し、動作が完了するまで High に保持する必要があります。

FPGA のプログラミング中にシステム コントローラ サスペンドモード ビットがデバイスに書き込まれた場合、デバイスはシステム コントローラ サスペンドモードに設定されます。JTAG\_TRST\_B ピンの状態は、JTAG および SPI スレーブ プログラミング モードが利用可能かどうかに影響しますが、自動更新および IAP プログラミング モードには影響しません(表 11-1 参照)。

**Note:** PolarFire FPGA デバイスの場合、JTAG\_TRST\_B = 1 にアサートする事によりシステム コントローラはサスペンドモードによりリセット状態に保持され、PF\_INIT\_MONITOR マクロの出力は「0」に設定されます。このマクロの出力はユーザロジック回路のリセット用にしばしば使われるため、ユーザはこれらの状態に対応するよう回路を設計する必要があります。

補遺: エラーコード

## 12. 補遺: エラーコード

システム コントローラは、回路から要求されたシステムサービスを実行します。サービスの完了後に、ステータスコードがユーザ アプリケーションに返されます。サービスに成功した場合のステータスコードは 0 です。サービスに失敗した場合、8 ビットのエラーコードが返されます。表 12-1 にエラーコードの一覧を示します。

### 表 12-1. エラーコード

エラー コード	エラーテキスト	エラーの概要
1	Validator or hash chaining mismatch	ビットストリームが正しく構成されていないか、不正なセキュリティ鍵が使われました。
2	Unexpected data received	EOB (End of the Bitstream)コンポーネントの後でデータを受信しました。
3	Invalid/corrupt encryption key	要求された鍵モードが無効になっているか、鍵の読み出しまたは 再構成ができませんでした。
4	Invalid component header	ビットストリームが無効なコンポーネントデータを含んでいます。
5	Back level not satisfied	ビットストリームは、デバイス内の現在のバックレベルよりも古いバージョンです。
6	Illegal bitstream mode	要求されたビットストリーム モードは、ユーザ セキュリティによって無効にされています。
7	DSN binding mismatch	ビットストリームは拒絶されました。ビットストリーム内のデバイスシリアル番号(DSN)がデバイス上の DSN と一致しません。
8	Illegal component sequence	ビットストリームが不正です。ビットストリームが ERR ステートで終了しています。有効なビットストリームは BEGIN ステートで始まり、END ステートで終了する必要があります。
9	Insufficient device capabilities	ビットストリームは拒絶されました。ビットストリーム内で指定されている能力がターゲット デバイスの能力に適合しません。
10	Incorrect DEVICEID	ビットストリームは拒絶されました。ビットストリーム内で指定されている DEVICEID がターゲット デバイスの製品 ID フィールドと一致しません。
11	Unsupported bitstream protocol version (regeneration required)	ビットストリームは拒絶されました。旧式のデバイスが新しいビットストリーム バージョンのデコードを試みたか、新しいデバイスが旧式ビットストリーム バージョンのデコードを試みました。
12	Verify not permitted on this bitstream	暗号鍵を使ってデバイスにビットストリームをプログラミングした後に、他の暗号鍵で暗号化されたビットストリームを使ってデバイスの内容を検証する事はできません。
13	Invalid device certificate	デバイス証明書が存在しないか無効です。
14	Invalid DIB	デバイス インテグリティ ビットが無効です。
21	Device not in SPI master mode	IAP モードのプログラミングが試みられましたが、デバイスが SPI マスタとして設定されていません。
22	No valid images found (auto update)	自動更新モードのプログラミングが試みられましたが、有効なイメージポインタが見つかりません。
23	No valid images found (IAP)	IAP モード(インデックスでイメージを指定) のプログラミングが 試みられましたが、有効なイメージポインタが見つかりません。

補遺: エラーコード

続	続き				
エラー コード	エラーテキスト	エラーの概要			
24	Programmed design version newer than auto update image	自動更新モードのプログラミングが試みられましたが、プログラミング済みのバージョンは最新です。			
25	Reserved	予約済み			
26	Selected image invalid and no recovery performed because the device is running a valid design	自動更新または IAP モードのプログラミングが試みられましたが、選択されたイメージが無効です。			
27	Selected recovery image failed to program	自動更新または IAP モードのプログラミングが試みられましたが、選択されたリカバリ イメージのプログラミングに失敗しました。			
127	中止	ビットストリームのロード中に非ビットストリーム命令が実行されました。			
128	NVMVERIFY	ファブリック/セキュリティ鍵セグメントの検証に失敗しました。			
129	PROTECTED	デバイスのセキュリティ設定により、デバイスの不揮発性メモリ は変更できません。			
130	NOTENA	プログラミング モードが有効になっていません。			
131	SNVMVERIFY	sNVM の検証動作に失敗しました。			
132	SYSTEM	システム ハードウェア(PUF または DRBG)でエラーが発生しました。			
133	BADCOMPONENT	コンポーネントのペイロード内でエラーが検出されました。			
134	HVPROGERR	HV プログラミング サブシステムが動作に失敗しました。			
135	HVSTATE	エラーのために HV プログラミング サブシステムが予期せぬ状態 となりました。			

## 13. 改訂履歴

本書に適用された変更の履歴を下表に示します新しいリビジョンから順番に記載しています。

#### 表 13-1. 改訂履歷

及 13-1. 以前 1		
リビジョン	日付	概要
В	04/2022	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>システム コントローラ サスペンドモードに関する情報を追加しました (11. 「システム コントローラ サスペンドモード」参照)。</li> <li>プログラマと SPI フラッシュ製品番号に関する情報を追加しました(10.3.2. 「JTAG を使った SPI フラッシュのプログラミング」参照)。</li> </ul>
A	08/2021	本書の初版です。 このユーザガイドは、以下の文書を 1 つの文書に統合する事により作成されました。 • UG0714:PolarFire FPGA Programming User Guide • UG0914:PolarFire SoC FPGA Programming User Guide これらのデバイスファミリ ユーザガイドの改訂履歴は、将来の参照用として本書に記載します (表 13-2、表 13-3 参照)。

表 13-2 に、『UG0714:PolarFire FPGA Programming User Guide』の改訂履歴を示します。この表には、各リビジョンに適用された変更内容を記載しています。

『UG0714:PolarFire FPGA Programming User Guide』は廃刊となり、その内容は『PolarFire® FPGA and PolarFire SoC FPGA Programming User Guide』(本書)に統合されました。

表 13-2.『UG0714: PolarFire FPGA Programming User Guide』の改訂履歴

リビジョン	日付	概要
リビジョン 9.0	6/21	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>「ブランクデバイスに対する自動更新(自動プログラミング)」を更新しました。</li> <li>「SPI ディレクトリ」を更新しました。</li> <li>「外部プロセッサを使って SPI フラッシュをプログラミングする」を更新しました。</li> <li>「Lebero を使って SPI フラッシュをプログラミングする」を更新しました。</li> <li>「ファブリック ユーザロジックを使って SPI フラッシュをプログラミングする」を更新しました。</li> </ul>
リビジョン 8.0	3/21	動的再設定インターフェイス(PF_DRI)に関する情報を削除しました。
リビジョン 7.0	8/20	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>「SPIマスタ プログラミング向けの推奨基板構成」の図を更新しました。</li> <li>「ビットストリーム コンポーネントの設定」を更新しました。</li> <li>「ファブリック ユーザロジックを使って SPI フラッシュをプログラミングする」を更新しました。</li> <li>「Lebero を使って SPI フラッシュをプログラミングする」を更新しました。</li> <li>「Libero SoC コンフィグレータ内でのフラッシュ プログラミング」のスクリーン ショットを更新しました。</li> <li>「FlashPro プログラマと PolarFire FPGA の接続」の図を更新しました。</li> <li>「プログラミング ファイルサイズ」を更新しました。</li> <li>「SPI スレーブ プログラミング インターフェイス」を更新しました。</li> <li>「デバイス プログラミング フロー」を更新しました。</li> <li>「JTAG プログラミング インターフェイス」を更新しました。</li> </ul>

続き					
	日付	概要			
リビジョン 6.0	5/20	Libero SoC v12.1 向けに文書を更新しました。			
リビジョン 5.0	10/18	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>Libero SoC PolarFire v2.3 向けに文書を更新しました。</li> <li>フラッシュフリーズ機能を削除しました。</li> <li>バックレベル保護機能に関する情報を追加しました(「バックレベル保護のバイパス」参照)。</li> <li>ゼロ化に関する情報を追加しました(「ゼロ化」参照)。</li> <li>「プログラミング ファイルサイズ」の表を更新しました。</li> </ul>			
リビジョン 4.0	4/18	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>MPF300 デバイス向けプログラミング ファイルサイズの詳細を更新しました。</li> <li>「JTAG を使った SPI フラッシュ プログラミング」と「ファブリック ユーザロジックを使った SPI フラッシュ プログラミング」の図内の SPI ブロックのポートの詳細を更新しました。</li> </ul>			
リビジョン 3.0	11/17	<ul> <li>このリビジョンでの主な変更内容は以下の通りです。</li> <li>Libero SoC PolarFire v2.0 向けに文書を更新しました。</li> <li>MPF300 デバイス向けプログラミング ファイルサイズの詳細を更新しました(「プログラミング ファイルサイズ」参照)。</li> <li>XCVR 参照クロック入力、XCVR データ I/O、GPIO、HSIO の I/O ステートに関する情報を更新しました。</li> <li>外部 SPI フラッシュをプログラミングするための手順を追加しました(「Libero を使って外部 SPI フラッシュをプログラミングする」参照)。</li> <li>JTAG ピンの詳細を更新しました。</li> <li>SPI マスタ プログラミングの推奨設定を示す図を更新しました。</li> <li>文書全体を通して VDDIx を VDDI3 に置換しました。</li> </ul>			
リビジョン 2.0	6/17	このリビジョンでの主な変更内容は以下の通りです。  ・ ビットストリームの生成に関する情報を更新しました。  ・ 自動更新に関する情報を更新しました。			
リビジョン 1.0	2/17	『UG0714: PolarFire FPGA Programming User Guide』の初版です。			

表 13-3 に、『UG0914: PolarFire SoC FPGA Programming User Guide』の改訂履歴を示します。この表には、各リビジョンに適用された変更内容を記載しています。

『UG0914: PolarFire SoC FPGA Programming User Guide』は廃刊となり、その内容は『PolarFire® FPGA and PolarFire SoC FPGA Programming User Guide』(本書)に統合されました。

表 13-3. 『UG0914: PolarFire SoC FPGA Programming User Guide』の改訂履歴

リビジョン	日付	概要
リビジョン 4.0	6/21	このリビジョンでの主な変更内容は以下の通りです。     「ブランクデバイスに対する自動更新(自動プログラミング)」を更新しました。     「SPI ディレクトリ」を更新しました。     「外部プロセッサを使って SPI フラッシュをプログラミングする」を更新しました。     「Lebero を使って SPI フラッシュをプログラミングする」を更新しました。     「ファブリック ユーザロジックを使って SPI フラッシュをプログラミングする」を更新しました。
リビジョン 3.0	3/21	「MSS を使って SPI フラッシュをプログラミングする」を更新しました。

続き				
リビジョン	日付	概要		
リビジョン 2.0	10/20	このリビジョンでの主な変更内容は以下の通りです。  「SPI マスタ プログラミング向けの推奨基板構成」の図を更新しました。  「ビットストリーム コンポーネントの設定」を更新しました。  「MSS を使って SPI フラッシュをプログラミングする」を更新しました。  「FlashPro プログラマと PolarFire SoC FPGA の接続」の図を更新しました。  「SPI スレーブ プログラミング インターフェイス」を更新しました。  「デバイス プログラミング フロー」を更新しました。		
リビジョン 1.0	4/20	『UG0914: PolarFire SoC FPGA Programming User Guide』の初版です。		

## Microchip 社の FPGA サポート

Microchip 社の FPGA 製品グループはカスタマサービス、カスタマ技術サポート、ウェブサイト、世界各地の営業所を含む各種のサポートサービスを提供しています。サポートチームにお問い合わせになる前に、弊社のウェブサイトをご覧ください。多くの場合、お客様に必要な情報は弊社ウェブサイトで見つかります。

技術サポートセンターにはウェブページ(www.microchip.com/support)からお問い合わせください。Microchip Support Case を作成する際は、FPGA 製品番号を指定し、適切なケースカテゴリを選択し、設計ファイルをアップロードしてください。

その他の製品サポート(価格、アップグレード、更新情報、注文ステータス、認証等)につきましては、カスタマサービスにお問い合わせください。

- 北米からのお問い合わせ電話番号: 800-262-1060
- その他の地域からのお問い合わせ電話番号: 650-318-4460
- 全世界からのお問い合わせ FAX 番号: 650-318-8044

### Microchip 社ウェブサイト

Microchip 社はウェブサイト(www.microchip.com)を通してオンライン サポートを提供しています。当ウェブサイトでは、お客様に役立つ情報やファイルを簡単に見つけ出せます。以下を含む各種の情報をご覧になれます。

- **製品サポート** データシートとエラッタ、アプリケーション ノートとサンプル プログラム、設計リソース、ユーザガイドとハードウェア サポート文書、最新のソフトウェアと過去のソフトウェア
- 技術サポート FAQ(よく寄せられる質問)、技術サポートのご依頼、オンライン ディスカッション グループ、 Microchip 社のデザイン パートナー プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** 製品セレクタと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い合わせ先(営業所/正規代理店)の一覧

### 製品変更涌知サービス

Microchip 社の製品変更通知サービスは、お客様に Microchip 社製品の最新情報をお届けする配信サービスです。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

http://www.microchip.com/pcn にアクセスし、登録手続きをしてください。

## カスタマサポート

Microchip 社製品をお使いのお客様は、以下のチャンネルからサポートをご利用になれます。

- 正規代理店
- 技術サポート

サポートは正規代理店にお問い合わせください。各地の営業所もご利用になれます。本書の最後のページに各国の営業所の一覧を記載しています。

技術サポートは以下のウェブページからもご利用になれます。www.microchip.com/support

## Microchip 社のデバイスコード保護機能

Microchip 社製品のコード保護機能について以下の点にご注意ください。

- Microchip 社製品は、該当する Microchip 社データシートに記載の仕様を満たしています。
- Microchip 社では、通常の条件ならびに仕様に従って使った場合、Microchip 社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。

- Microchip 社はその知的財産権を重視し、積極的に保護しています。Microchip 社製品のコード保護機能の侵害は 固く禁じられており、デジタル ミレニアム著作権法に違反します。
- Microchip 社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。コード保護機能は常に進歩しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。

### 法律上の注意点

本書および本書に記載されている情報は、Microchip 社製品を設計、テスト、お客様のアプリケーションと統合する目的を含め、Microchip 社製品に対してのみ使用する事ができます。それ以外の方法でこの情報を使用する事はこれらの条項に違反します。デバイス アプリケーションの情報は、ユーザの便宜のためにのみ提供されるものであり、更新によって変更となる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。#TOIN#

その他のサポートは Microchip 社正規代理店にお問い合わせ頂くか、www.microchip.com/en-us/support/ design-help/client-support-services をご覧ください。

Microchip 社は本書の情報を「現状のまま」で提供しています。Microchip 社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。

いかなる場合も Microchip 社は、本情報またはその使用に関連する間接的、特殊的、懲罰的、偶発、的または必然的 損失、損害、費用、経費のいかんにかかわらず、また Microchip 社がそのような損害が生じる可能性について報告を 受けていた場合あるいは損害が予測可能であった場合でも、一切の責任を負いません。法律で認められる最大限の範 囲を適用しようとも、本情報またはその使用に関連する一切の申し立てに対する Microchip 社の責任限度額は、使用 者が当該情報に関連して Microchip 社に直接支払った額を超えません。

Microchip 社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途に Microchip 社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、 Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。特に明記しない場合、暗黙的あるいは明示的を問わず、Microchip 社が知的財産権を保有しているライセンスは一切譲渡されません。

### 商標

Microchip 社の名称とロゴ、Microchip ロゴ、Adaptec、AnyRate、AVR、AVR ロゴ、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi ロゴ、MOST、MOST ロゴ、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST ロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGA は米国およびその他の国における Microchip Technology Incorporated の登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus、ProASIC Plus、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、TrueTime、WinPath、ZL は米国における Microchip Technology Incorporated の登録商標です。

Adjacent Key Suppression AKS Analog-for-the-Digital Age Any Capacitor Anyln AnyOut Augmented Switching BlueSky BodyCom CodeGuard CryptoAuthentication CryptoAutomotive CryptoCompanion CryptoController dsPICDEM dsPICDEM.net Dynamic Average Matching DAM ECAN Espresso T1S EtherGREEN GridTime IdealBridge In-Circuit Serial Programming ICSP, INICnet Intelligent Paralleling Inter-Chip Connectivity JitterBlocker Knob-on-Display maxCrypto maxView memBrain Mindi MiWi MPASM MPF MPLAB Certified D MPLIB MPLINK MultiTRAK NetDetach NVM Express NVMe Omniscient Code Generation PICDEM PICDEM.net PICkit PICtail PowerSmart PureSilicon QMatrix REAL ICE Ripple Blocker RTAX RTG4 SAM- ICE Serial Quad I/O simpleMAP SimpliPHY SmartBuffer SmartHLS SMART-I.S., storClad SQI SuperSwitcher

SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、

VeriPHY、ViewSpan、WiperLock、XpressConnect、ZENA は米国およびその他の国における Microchip Technology Incorporated の商標です。

SQTP は米国における Microchip Technology Incorporated のサービス マークです。

Adaptec ロゴ、Frequency on Demand、Silicon Storage Technology、Symmcom、Trusted Time はその他の国における Microchip Technology Incorporated の登録商標です。

GestIC は、米国以外の国における Microchip Technology Inc.の子会社である Microchip Technology Germany II GmbH & Co. KG の登録商標です。

その他の本書に記載されている商標は各社に帰属します。

© 2022, Microchip Technology Incorporated and its subsidiaries.All Rights Reserved.

ISBN: 978-1-6683-0248-4

## 品質管理システム

Microchip 社の品質管理システムについては www.microchip.com/quality をご覧ください。



# 各国の営業所とサービス

#### 北米

本社

2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel:480-792-7200 Fax:480-792-7277

技術サポート:

http://www.microchip.com/ support

URL:

www.microchip.com

アトランタ Duluth, GA Tel:678-957-9614 Fax:678-957-1455

オースティン、TX Tel:512-257-3370

ポストン Westborough, MA Tel:774-760-0087 Fax:774-760-0088

シカゴ Itasca, IL Tel:630-285-0071 Fax:630-285-0075

**ダラス** Addison, TX Tel:972-818-7423 Fax:972-818-2924

デトロイト Novi, MI Tel:248-848-4000

ヒューストン、TX Tel:281-894-5983

インディアナポリス Noblesville, IN Tel:317-773-8323 Fax:317-773-5453 Tel:317-536-2380

ロサンゼルス Mission Viejo, CA Tel:949-462-9523 Fax:949-462-9608 Tel:951-273-7800

ローリー、NC Tel:919-844-7510

ニューヨーク、NY Tel:631-435-6000

サンノゼ、CA Tel:408-735-9110 Tel:408-436-4270

カナダ - トロント Tel:905-695-1980 Fax:905-695-2078

#### アジア/太平洋

オーストラリア - シドニー Tel:61-2-9868-6733

**中国 - 北京** Tel:86-10-8569-7000

**中国 - 成都** Tel:86-28-8665-5511

中国 - 重慶

Tel:86-23-8980-9588

**中国 - 東莞** Tel:86-769-8702-9880

**中国 - 広州** Tel:86-20-8755-8029

**中国 - 杭州** Tel:86-571-8792-8115

**中国 - 香港 SAR** Tel:852-2943-5100

**中国 - 南京** Tel:86-25-8473-2460

**中国 - 青島** Tel:86-532-8502-7355

**中国 - 上海** Tel:86-21-3326-8000

**中国 - 瀋陽** Tel:86-24-2334-2829

**中国 - 深**圳 Tel:86-755-8864-2200

**中国 - 蘇州** Tel:86-186-6233-1526

**中国 - 武漢** Tel:86-27-5980-5300

**中国 - 西安** Tel:86-29-8833-7252

**中国 - 厦門** Tel:86-592-2388138

**中国 - 珠海** Tel:86-756-3210040

#### アジア /太 平洋

インド - バンガロール Tel:91-80-3090-4444

インド - ニューデリー Tel:91-11-4160-8631

インド - プネ Tel:91-20-4121-0141

**日本 - 大阪** Tel:81-6-6152-7160

**日本 - 東京** Tel:81-3-6880- 3770

**韓国 - 大邱** Tel:82-53-744-4301

**韓国 - ソウル** Tel:82-2-554-7200

マレーシア - クアラルンプール Tel:60-3-7651-7906

マレーシア - ペナン Tel:60-4-227-8870

フィリピン - マニラ Tel:63-2-634-9065

シンガポール Tel:65-6334-8870

**台湾 - 新竹** Tel:886-3-577-8366

**台湾 - 高雄** Tel:886-7-213-7830

**台湾 - 台北** Tel:886-2-2508-8600

タイ - バンコク Tel:66-2-694-1351

ベトナム - ホーチミン Tel:84-28-5448-2100

#### ヨーロッパ

オーストリア - ヴェルス Tel:43-7242-2244-39 Fax:43-7242-2244-393

デンマーク - コペンハーゲン Tel:45-4485-5910 Fax:45-4485-2829

フィンランド - エスポー Tel:358-9-4520-820

フランス - パリ Tel:33-1-69-53-63-20 Fax:33-1-69-30-90-79

ドイツ - ガーヒング Tel:49-8931-9700

ドイツ - ハーン Tel:49-2129-3766400

ドイツ - ハイルブロン Tel:49-7131-72400

ドイツ - カールスルーエ Tel:49-721-625370

ドイツ - ミュンヘン Tel:49-89-627-144-0 Fax:49(-89/-627)-144/-44

ドイツ - ローゼンハイム Tel:49-8031-354-560

イスラエル - ラーナナ Tel:972-9-744-7705

イタリア - ミラノ Tel:39-0331-742611 Fax:39-0331-466781

イタリア - パドヴァ Tel:39-049-7625286

オランダ - ドリューネン Tel:31-416-690399 Fax:31-416-690340

ノルウェー - トロンハイム Tel:47-7288-4388

ポーランド - ワルシャワ Tel:48-22-3325737

ルーマニア - ブカレスト Tel:40-21-407-87-50

スペイン - マドリッド Tel:34-91-708-08-90 Fax:34-91-708-08-91

スウェーデン - ヨーテボリ Tel:46-31-704-60-40

スウェーデン - ストックホルム Tel:46-8-5090-4654

イギリス - ウォーキンガム Tel:44-118-921-5800 Fax:44-118-921-5820