

PolarFire® FPGA および PolarFire SoC FPGA 電源投入/リセット 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 FPGA 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 FPGA は不揮発性 SoC 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 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 ファミリに基づく高性能 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、確実な起動を保證するために先進の電源投入回路を備えています。パワーオン リセット(POR)回路とシステム コントローラにより、デバイスへの電源投入時にシステムチェックな POR が確保されます。システム コントローラはデバイスのブートと回路の初期化を行います。本書では、デバイスの電源投入およびリセットの全体的なプロセスについて説明します。

表 1 に、PolarFire FPGA および PolarFire SoC FPGA ファミリの電源投入およびリセット時のステートを示します。

表 1. 電源およびリセット時のステート

コンポーネント/機能	PolarFire FPGA (MPF)	PolarFire SoC FPGA (MPFS)
パワーオン	✓	✓
デバイスブート	✓	✓
回路/メモリ初期化	✓	✓
• μ PROM	✓	✓
• sNVM	✓	✓
• SPI フラッシュ	✓	✓
• eNVM	—	✓
MSS プリブート	—	✓
MSS ユーザブート	—	✓
デバイスリセット	✓	✓
MSS リセット	—	✓

参考資料

- 組み込みメモリブロックの詳細: 『[PolarFire FPGA and PolarFire SoC FPGA Fabric User Guide](#)』
- MSS ブートの詳細: 『[PolarFire SoC Software Development and Tool Flow User Guide](#)』
- PCIe 初期化プロセスの詳細: 『[PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide](#)』
- PUFT(Power-Up to Functional Timing)の詳細: 『[PolarFire FPGA Datasheet](#)』 または 『[PolarFire SoC FPGA Advance Datasheet](#)』
- MSS の詳細: [PolarFire SoC FPGA MSS Technical Reference Manual](#)
- 電源投入シーケンスの要件と推奨手順: 『[UG0726: PolarFire FPGA Board Design User Guide](#)』 または 『[UG0901: PolarFire SoC Board Design Guidelines User Guide](#)』

目次

はじめに.....	1
1. 参考資料.....	2
1. 略語の一覧.....	5
2. 電源投入.....	6
2.1. パワーオン.....	6
2.2. デバイスブート.....	
2.3. 回路とメモリの初期化.....	7
2.4. MSS プリブート(PolarFire SoC FPGA のみ).....	24
2.5. MSS ユーザブート(PolarFire SoC FPGA のみ).....	30
2.6. HSIO/GPIO バンクの初期化.....	30
2.7. I/Oの再キャリブレーション.....	32
2.8. トランシーバの初期化.....	33
2.9. ユーザ PLL/DLL の初期化.....	34
2.10. PCIe の初期化.....	34
2.11. 電源投入中のブロックのステート.....	34
3. PolarFire FPGA のリセット.....	36
3.1. ハードリセット.....	36
3.2. ユーザリセットの生成手順.....	37
4. PolarFire SoC FPGA のリセット.....	40
4.1. ハードリセット.....	40
4.2. MSS リセット.....	40
4.3. ユーザリセットの生成手順.....	42
5. システム コントローラ サスペンドモード.....	45
5.1. デバイスのプログラミングとシステムサービス.....	45
5.2. PolarFire SoC のリブート.....	45
6. 補遺: 電源.....	47
7. 改訂履歴.....	48
Microchip 社の FPGA サポート.....	51
Microchip 社ウェブサイト.....	51
製品変更通知サービス.....	51
カスタマサポート.....	51
Microchip 社のデバイスコード保護機能.....	51
法律上の注意点.....	52
商標.....	52
品質管理システム.....	53

各国の営業所とサービス.....	54
------------------	----

1. 略語の一覧

本書で使う略語の一覧を表 1-1 に示します。

表 1-1. 略語の一覧

略語	意味
AMBA	ARM 先進マイクロコントローラ バス アーキテクチャ
eNVM	組み込み不揮発性メモリ
MSS	マイクロプロセッサ サブシステム
POR	パワーオン リセット
SCB	システム コントローラバス
sNVM	セキュア不揮発性メモリ
HSIO	高速 I/O
GPIO	汎用 I/O
PLL	位相ロックループ
DLL	遅延ロックループ
FIC	ファブリック インターフェイス コントローラ
PCIe	Peripheral Component Interconnect Express
SCSM	システム コントローラ サスペンドモード
PUFT	Power-Up to Function Time: 電源投入から動作開始までの時間

2. 電源投入

デバイスの電源投入プロセスは以下の一連の手順を含みます。

- 2.1. パワーオン
- 2.2. デバイスブート
- 2.3. 回路/メモリの初期化
- 2.4. MSS プリブート(PolarFire SoC FPGA のみ)
- 2.5. MSS ユーザブート(PolarFire SoC FPGA のみ)
- 2.6. HSIO/GPIO バンクの初期化
- 2.8. トランシーバの初期化
- 2.9. ユーザ PLL/DLL の初期化
- 2.10. PCIe の初期化
- 2.11. 電源投入中のブロックのステート

2.1 パワーオン

デバイスに電源が投入されると、POR 回路は電圧検出回路を使って VDD、VDD18、VDD25 電源レール電圧の立ち上がりを検出します。電源の一覧は 6. 「補遺: 電源」を参照してください。システム コントローラは、電圧がしきい値レベルに達するまでリセット状態を維持します。システム コントローラは、FPGA ファブリックおよび関連 I/O の有効化を行います。

電圧が最低しきい値レベルに達した事を正確に検出するため、デバイス内部の電圧検出回路は高精度でキャリブレーションされます。デバイスを POR から解除するために必要なしきい値電圧レベルに関しては、『PolarFire FPGA Datasheet』または『PolarFire SoC FPGA Advance Datasheet』内の「Power-on Reset Voltages」を参照してください。全ての電源レールの電圧がそれぞれのしきい値電圧レベルに達すると、遅延時間(TCALIB: 20 μ s~50 ms の範囲で設定可能)の後にデバイスブートが始まります。

どちらのデバイスファミリも、I/O バンク電源ごとに別々の電圧監視回路を備えています。POR 中は、専用 I/O バンクに電源が投入され、シリアル トランシーバとファブリックの電源は遮断され、HSIO/GPIO バンクは 3 ステートにされます。バンク 3 向け I/O バンク コントローラ内の電圧検出回路によって VDDI3 が入力の有効化に必要な電圧レベルに達した事が検出されると、200 ns の遅延後に、専用 I/O バンクの出力(SPI コンフィギュレーションと JTAG I/O を含む)の有効化が可能となります。

電源投入シーケンスの詳細は、『UG0726: PolarFire FPGA Board Design User Guide』または『UG0901: PolarFire SoC Board Design Guidelines User Guide』内の「Core Power Supply Operations」を参照してください。

2.2 デバイスブート

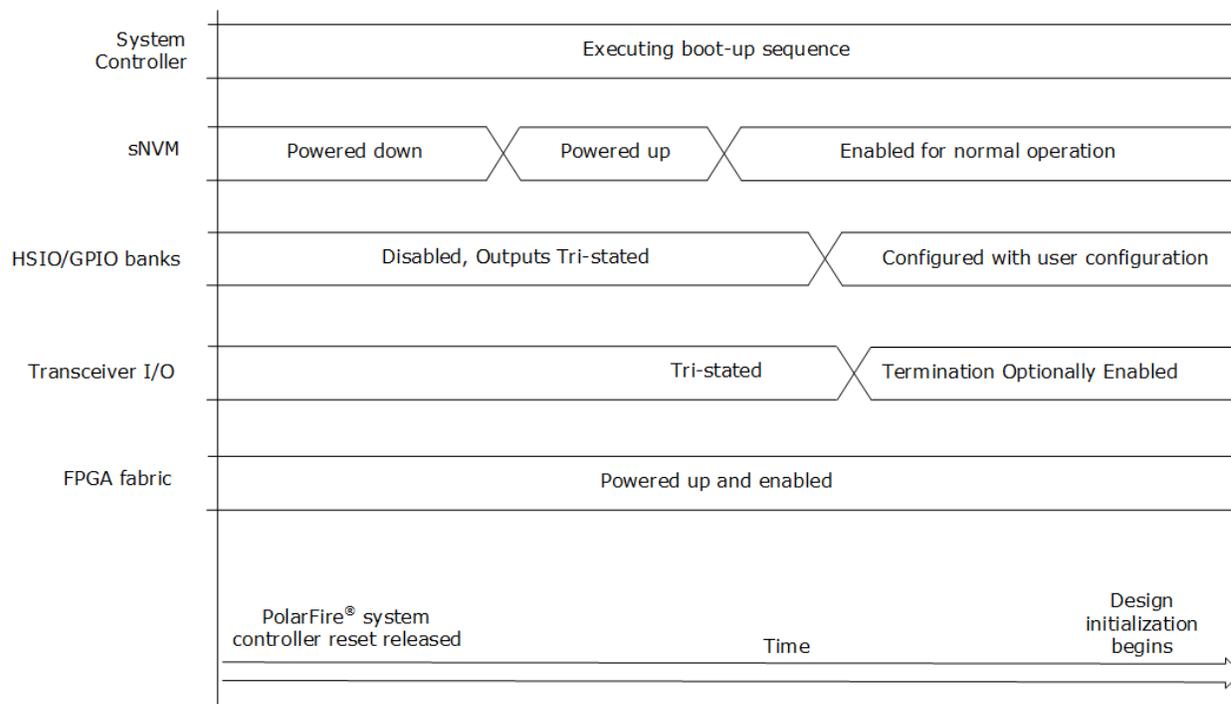
システム コントローラは、POR 回路によってリセット状態から解除された後に、デバイス ブートアップ プロセスを実行する事により FPGA ファブリックおよび関連 I/O を起動します。システム コントローラは、ユーザ回路に関係なく、常に同じデバイス ブートアップ シーケンスを実行します。

デバイス ブートアップ時に以下のイベントが発生します。

- sNVM の電源投入と通常動作向けの有効化
- トランシーバ I/O の有効化
- ユーザ電圧検出回路の有効化
- FPGA ファブリックの電源投入と有効化
- HSIO および GPIO バンクの設定(Libero[®] SoC 内のユーザ設定に基づく)
 - キャリブレーション前は GPIO のみが使用可能
- MSS の電源遮断と MSSIO の 3 ステート化(PolarFire SoC FPGA のみ)

図 2-1 に、プログラミング済み PolarFire FPGA デバイスのブートアップ シーケンスを示します。

図 2-1. PolarFire FPGA のブートアップ



2.3 回路とメモリの初期化

デバイス ブートプロセスが完了した後に、ファブリック RAM ブロック (LSRAM と μ SRAM) は 0 (既定値) へ初期化されます。どちらのデバイスファミリでも、必要に応じてファブリック RAM ブロックを特定の値に初期化する事が可能です。回路内で使われる PCIe および XCVR ブロックは、電源投入時にユーザ設定データを使って初期化されます。システム コントローラは、電源投入シーケンス中に回路とメモリの初期化を実行します。メモリ初期化データは μ PROM、sNVM、外部 SPI フラッシュに保存しておく事ができます。初期化データの保存位置は、Liberio SoC **Design Flow** 内で選択します。初期化データは暗号化して外部 SPI フラッシュに保存できます。

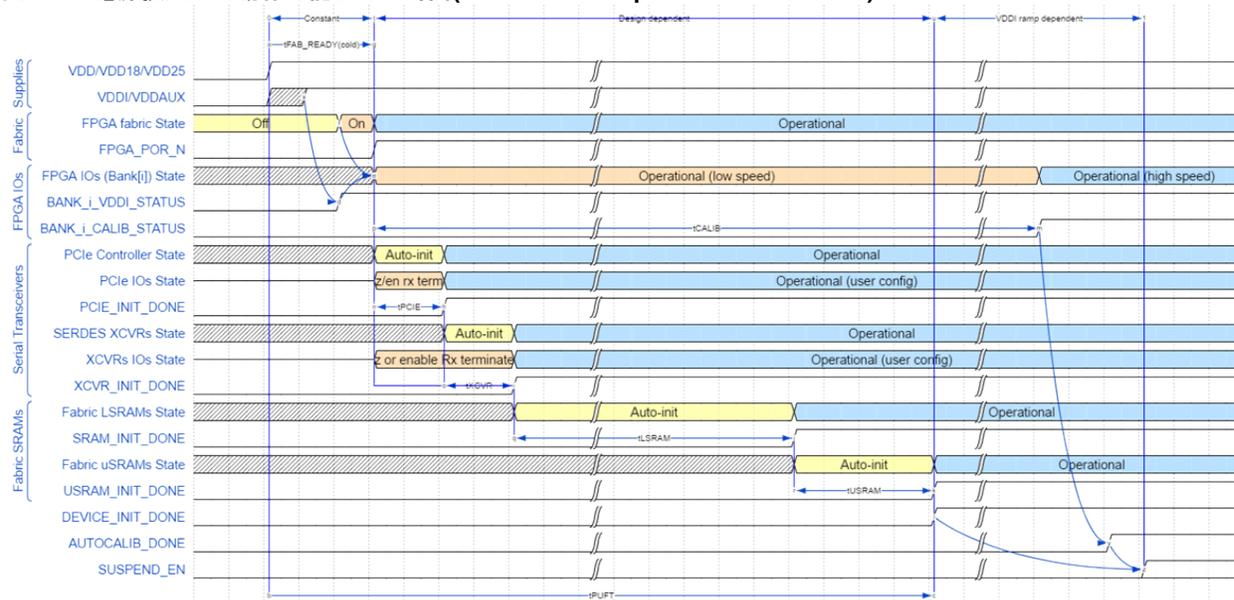
図 2-2 に、ファブリック、PCIe、トランシーバ、LSRAM、 μ SRAM を自動的に初期化するシーケンスを示します。このシーケンスは、ユーザ回路内でどのリソースがインスタンス化されているかに応じてカスタマイズされます。例えばユーザ回路が PCIe を含まない場合、PCIE_INIT_DONE はアサートされません。この場合、シーケンスは PCIe の初期化をスキップして次の手順へ進みます。PolarFire SoC FPGA の MSS (マイクロプロセッサ サブシステム) は、この段階ではリセットされたままです。

Liberio SoC の場合、以下のいずれかの方法によりメモリを初期化できます。

- Place and Route の実行後に、**[Configure Design Initialization Data and Memories]** オプションの**[fabric RAMs]**タブを使って content ファイルをインポートする (詳細は 2.3.7. 「回路/メモリ初期化の設定方法」参照)
- Place and Route の実行前に、**LSRAM and μ SRAM Configurator** を使って content ファイルをインポートする (詳細は 2.3.9. 「Place and Route の前の RAM 初期化」参照)

ユーザは初期化モニタを使って回路初期化ステータスを監視できます。

図 2-2. 電源投入から動作可能までの時間(PUFT: Power-Up to Functional Time)



電源投入から動作可能になるまでに要する総時間(tPUFT) は下式により定義されます。

$$t_{PUFT} = t_{FAB_READY(cold/warm)} + \max((t_{PCIE} + t_{XCVR} + t_{LSRAM} + t_{USRAM}), t_{CALIB})$$

PUFT は回路設定に応じて異なります。PUFT の代表値については、『PolarFire FPGA Datasheet』または『PolarFire SoC Advance Datasheet』を参照してください。

Note:

- **PUFT (Power-up To Functional Time)** は、I/O バンクの VDDI/VDDAUX の電源投入タイミング (VDD/VDD18/VDD25 より前か後か) に応じて異なります。I/O バンクの有効化時間は、VDD/VDD18/VDD25 のアサート時点から計測されます。VDD/VDD18/VDD25 の電源投入から十分な時間が経過した後に I/O の VDDI/VDDAUX が投入される場合、I/O バンクの有効化時間は VDDI/VDDAUX のアサート時点から計測されます。この場合の I/O 動作は、FABRIC_POR_N のネゲートに対して相対的に計測されるのではなく、BANK_#_VDDI_STATUS のアサートにより示されます。
- AUTOCALIB_DONE のアサートは、DEVICE_INIT_DONE のアサートより前で発生する場合もあれば、後で発生する場合もあります。AUTOCALIB_DONE のアサートに要する時間は以下によって決まります。
 - VDD/VDD18/VDD25 のパワーオン後に VDDI/VDDAUX が投入されるタイミング
 - 自動キャリブレーション向けに指定された各 I/O バンクの VDDI ランプ時間
 - PCIe、SerDes トランシーバ、ファブリック LSRAM 向けに実行される自動初期化の時間
 - 自動キャリブレーション用に指定された I/O バンクのいずれかで、自動キャリブレーション タイムアウト ウィンドウ内に VDDI/VDDAUX がパワーオンされていない場合、その後 VDDI/VDDAUX がパワーオンされるたびに自動キャリブレーションが実行されます。そのような I/O バンクで正確なキャリブレーションを実行するには、ファブリックからの CALIB_START を使って再キャリブレーションを開始する必要があります。
- サスペンドモードが有効である場合、DEVICE_INIT_DONE または AUTOCALIB_DONE のアサートから約 100 システムコントローラクロックサイクル後に SUSPEND_EN がアサートします。
- 両デバイスファミリとも、電源電圧を監視するためのタンパー検出機能により、しきい値電圧の最低または最高値を検出するためのフラグが提供されます。これらのフラグは、回路の初期化後にのみ有効であり、POR 中は無効です。システムコントローラ サスペンドモードが有効である場合、TAMPER フラグをラッチして、DEVICE_INIT_DONE がアサートされた後、SUSPEND_EN がアサートされる前に、ファブリック回路によって読み取るようにする必要があります。

回路の初期化中に以下の信号がアサートされます。

- **DEVICE_INIT_DONE:** 回路初期化の完了時にアサートされます。
- **FABRIC_POR_N:** ファブリックが動作可能になった時点でディアサートされます。
- **PCIE_INIT_DONE:** PCIe コントローラが初期化されるまで PCIe 関連のファブリック ロジックをリセット状態に保持するために使われます。PCIE_INIT_DONE は、PCIe クワッド内に配置された PCIe レーン インスタンスの初期化後にアサートされます。XCVR レーンが PCIe 対応クワッド内に配置されている場合、XCVR_INIT_DONE がアサートされます。

- **XCVR_INIT_DONE:** XCVR ブロックが初期化され時にアサートされます。
- **SRAM_INIT_DONE:** LSRAM ブロックが初期化された時にアサートされます。
- **USRAM_INIT_DONE:** μ SRAM ブロックが初期化された時にアサートされます。
- **BANK_#_CALIB_STATUS:** ユーザロジックはこの信号を使って各 I/O バンクのキャリブレーションが完了した事を検出できます。#はバンク番号(0/1/7/8/9)です。
- **BANK_#_VDDI_STATUS:** この信号は、特定 I/O バンクの VDDI 電源のステータスを監視するために使えます。この信号は、対応するバンクのいずれかが選択されている場合に INIT_MONITOR IP から出力されます。#はバンク番号(0/1/7/8/9)です。
- **SRAM_INIT_FROM_SNVM_DONE:** sNVM から SRAM が初期化された時にアサートされます。
- **USRAM_INIT_FROM_SNVM_DONE:** sNVM から μ SRAM が初期化された時にアサートされます。
- **SRAM_INIT_FROM_UPROM_DONE:** μ PROM から SRAM が初期化された時にアサートされます。
- **USRAM_INIT_FROM_UPROM_DONE:** μ PROM から μ SRAM が初期化された時にアサートされます。
- **SRAM_INIT_FROM_SPI_DONE:** SPI から SRAM が初期化された時にアサートされます。
- **USRAM_INIT_FROM_SPI_DONE:** SPI から μ SRAM が初期化された時にアサートされます。

2.3.1 PolarFire 初期化モニタ

PolarFire 初期化モニタ(PF_INIT_MONITOR) は、デバイス設定ステータスを FPGA ファブリックに提供するために使われます。この IP は、全ての回路において FPGA ファブリック内でインスタンス化される必要があり、デバイス初期化が完了するまでユーザ ファブリック ロジックの動作を抑制するために使われます。DEVICE_INIT_DONE のアサートは、デバイス設定が完了した事を示します。PolarFire および PolarFire SoC デバイスはシステム コントローラ サスペンドモード(SCSM)を備えており、このモードを使ってデバイスの初期化完了後もシステム 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、このモードを使ってシングル 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事が不可欠です。サスペンドモードを使う場合、ラッチシステム コントローラ出力オプションを有効にする必要があります。この場合、CLK_160_MHZ ポートを内部 160 MHz RCOSC に接続する必要があります。PF_INIT_MONITOR IP は、Clock and Management の下の IP カタログ内で利用できます (図 2-3 参照)。

Note: PolarFire および RT PolarFire デバイスでシステム コントローラ サスペンドモードを使う場合、JTAG_TRST_B ピンが論理 High にアサートされると PF_INIT_MONITOR マクロの全ての出力は 0 に設定されます。この状況は、ユーザが SmartDebug を使ってデバイスの再プログラミングまたはデバッグを試みた場合に発生します。PF_INIT_MONITOR マクロ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。システム コントローラ サスペンドモードの詳細は、『[PolarFire FPGA and PolarFire SoC FPGA System Services User Guide](#)』を参照してください。

図 2-3. PolarFire 初期化モニタ コンフィグレータ

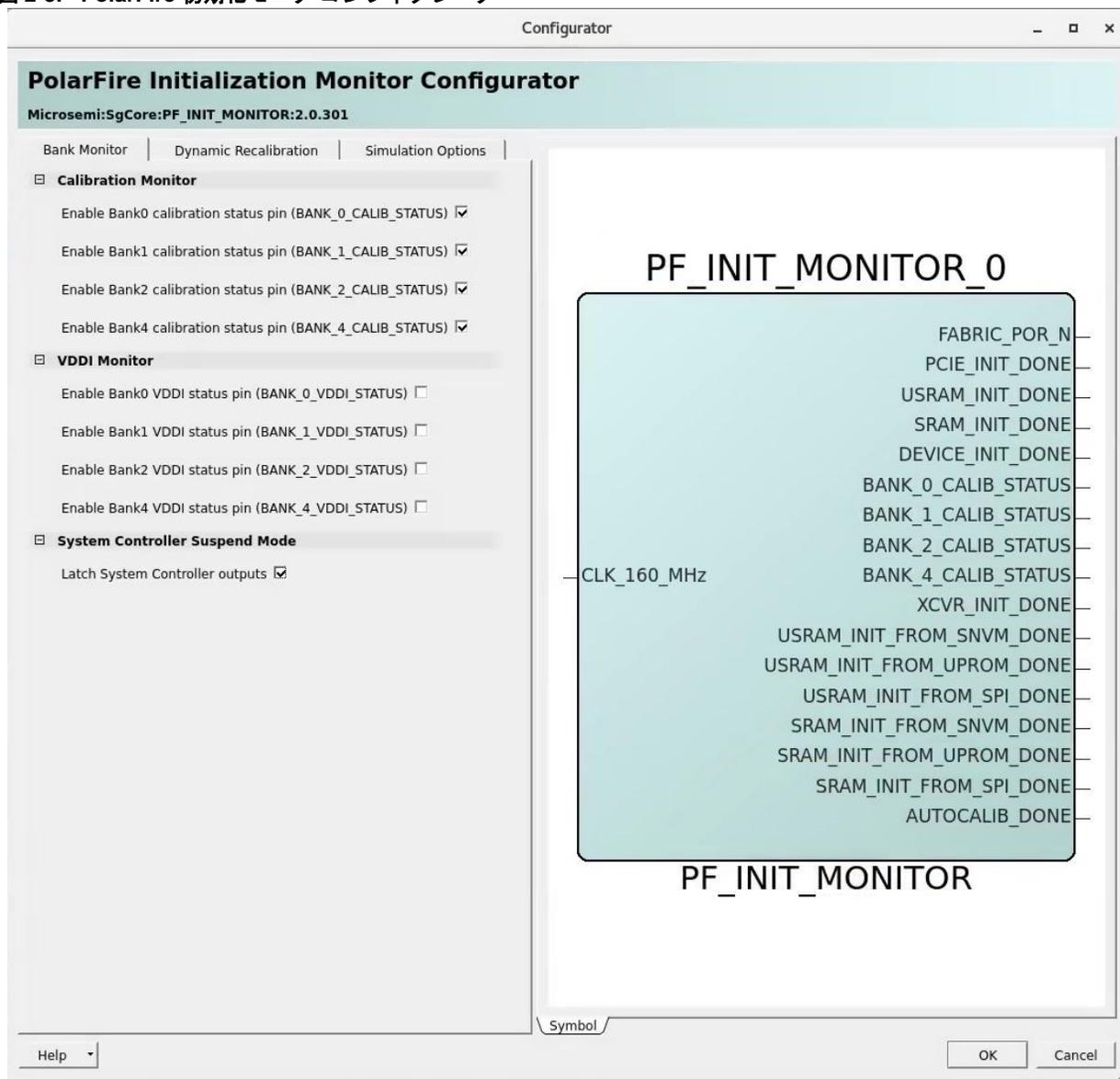
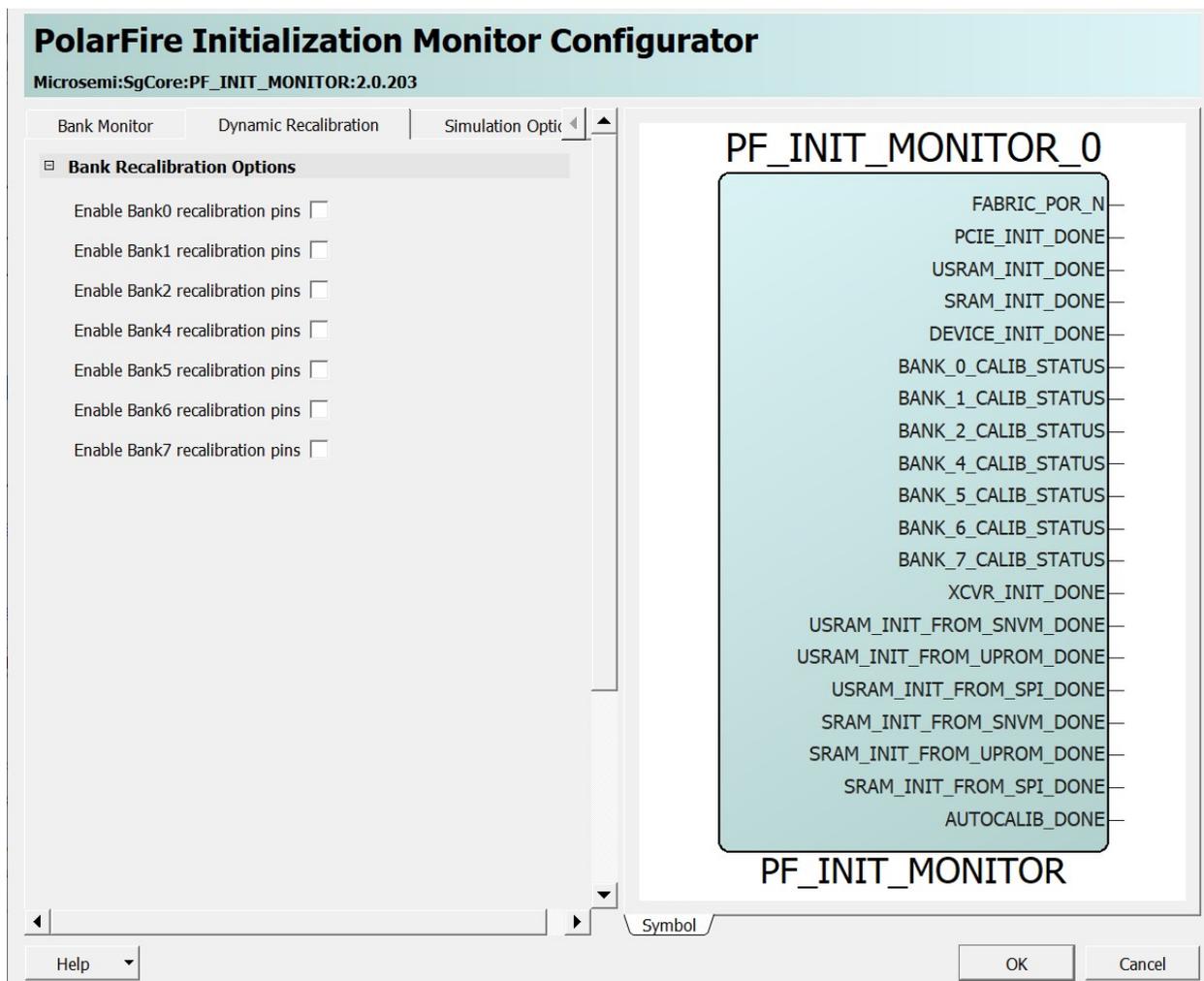


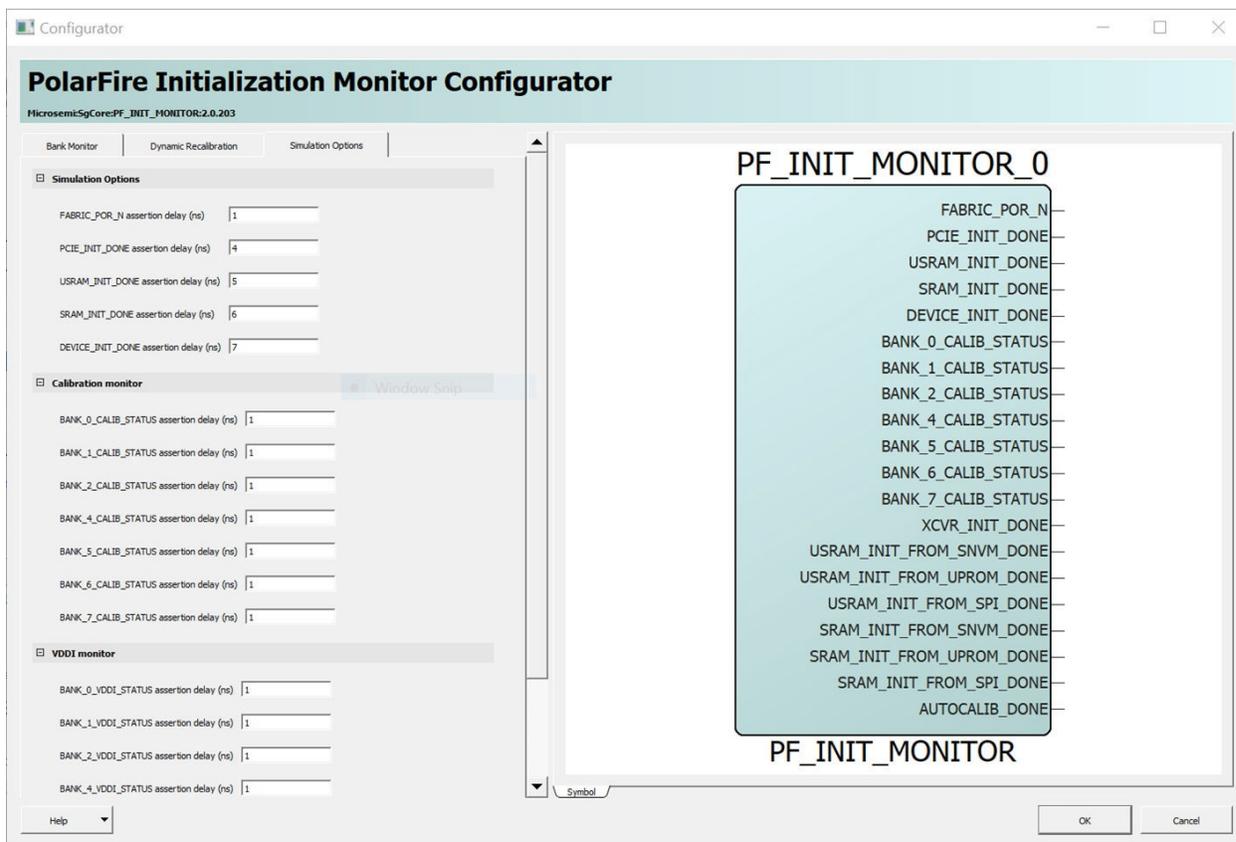
図 2-4 に、[Dynamic Recalibration]タブを示します。

図 2-4. PolarFire 初期化モニタ コンフィグレータ - [Dynamic Recalibration]タブ



PolarFire 初期化モニタはシミュレーションをサポートします。[Simulation Options]タブを使って、シミュレーション開始から出力信号リリースまでの時間を指定します。図 2-5 に、[Simulation Options]タブを示します。

図 2-5. PolarFire 初期化モニタ コンフィグレータ - [Simulation Options]タブ



Note: DDR コントローラのトレーニング ロジックを開始する前に、I/O をキャリブレートする必要があります。このために、PFSOC_INIT_MONITOR IP の DEVICE_INIT_DONE 信号と BANK_#_CALIB_STATUS 信号の間の AND (論理積) によってリセット信号を生成する必要があります。BANK_#は、DDR サブシステムが配置されているバンクを示します。

2.3.2 PolarFire SoC 初期化モニタ

PolarFire SoC 初期化モニタ(PFSOC_INIT_MONITOR) は、FPGA ファブリックにデバイス設定ステータスを提供します。この IP は、全ての回路において FPGA ファブリック内でインスタンス化される必要があります。デバイス初期化が完了するまでユーザ ファブリック ロジックの動作を抑制するために使います。DEVICE_INIT_DONE のアサートは、デバイス設定が完了した事を示します。PolarFire および PolarFire SoC デバイスはシステム コントローラ サスペンドモード(SCSM)を備えており、このモードを使ってデバイスの初期化完了後もシステム 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。サスペンドモードを使う場合、ラッチシステム コントローラ出力オプションを有効にする必要があります。この場合、CLK_160_MHZ ポートを内部 160 MHz RCOSC に接続する必要があります。PFSOC_INIT_MONITOR IP は、Clock and Management の下の IP カタログ内で利用できます。

図 2-6. PolarFire SoC 初期化モニタ コンフィグレータ

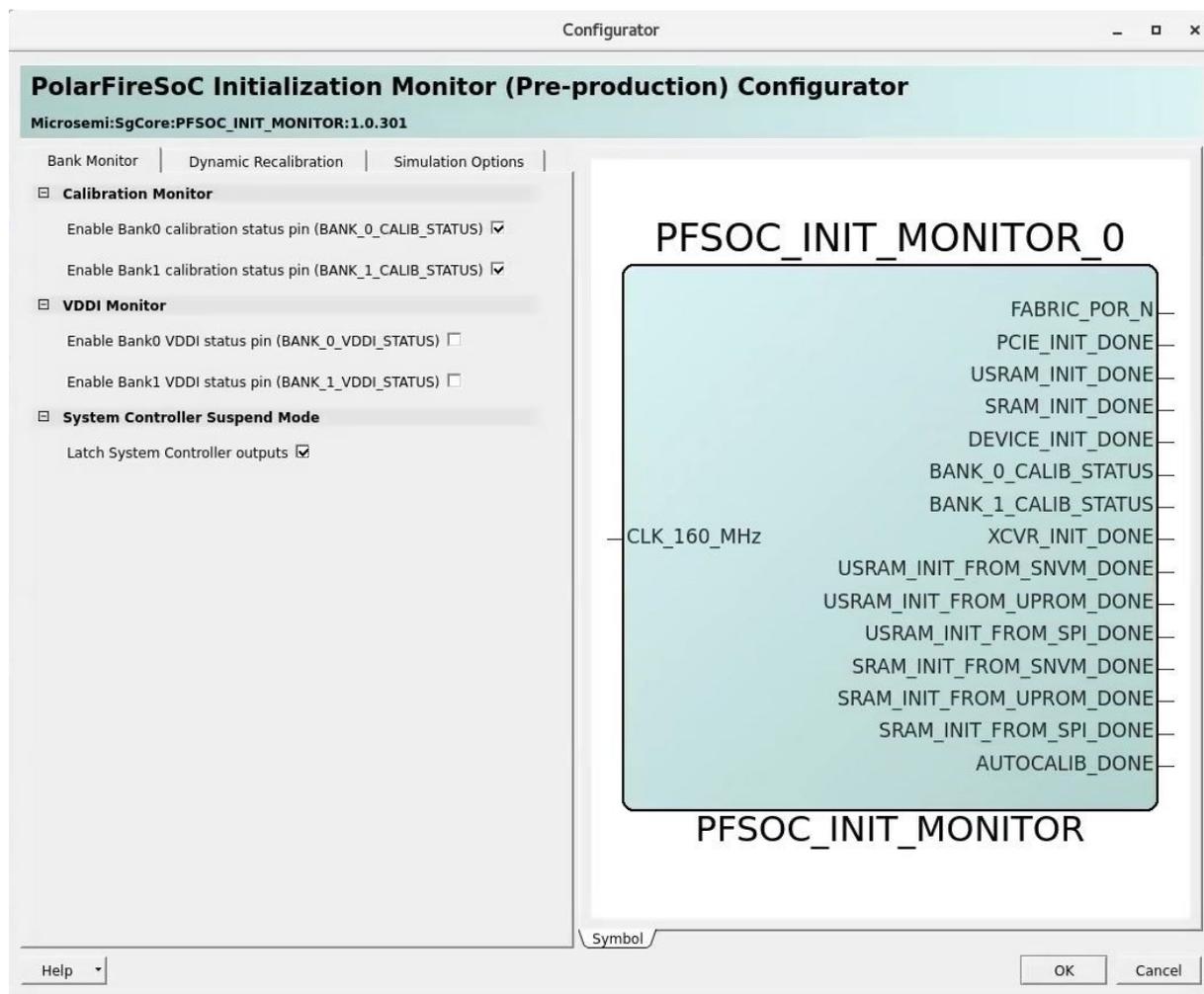
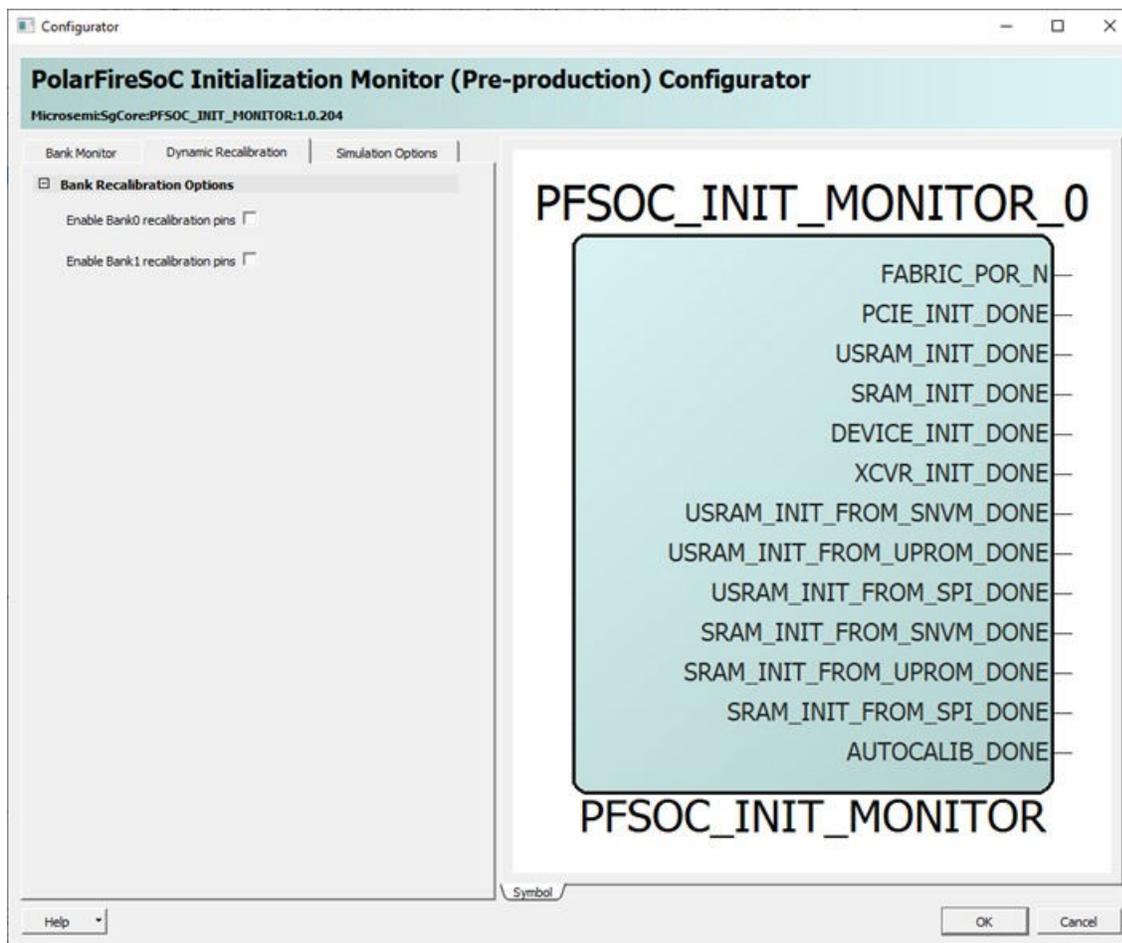


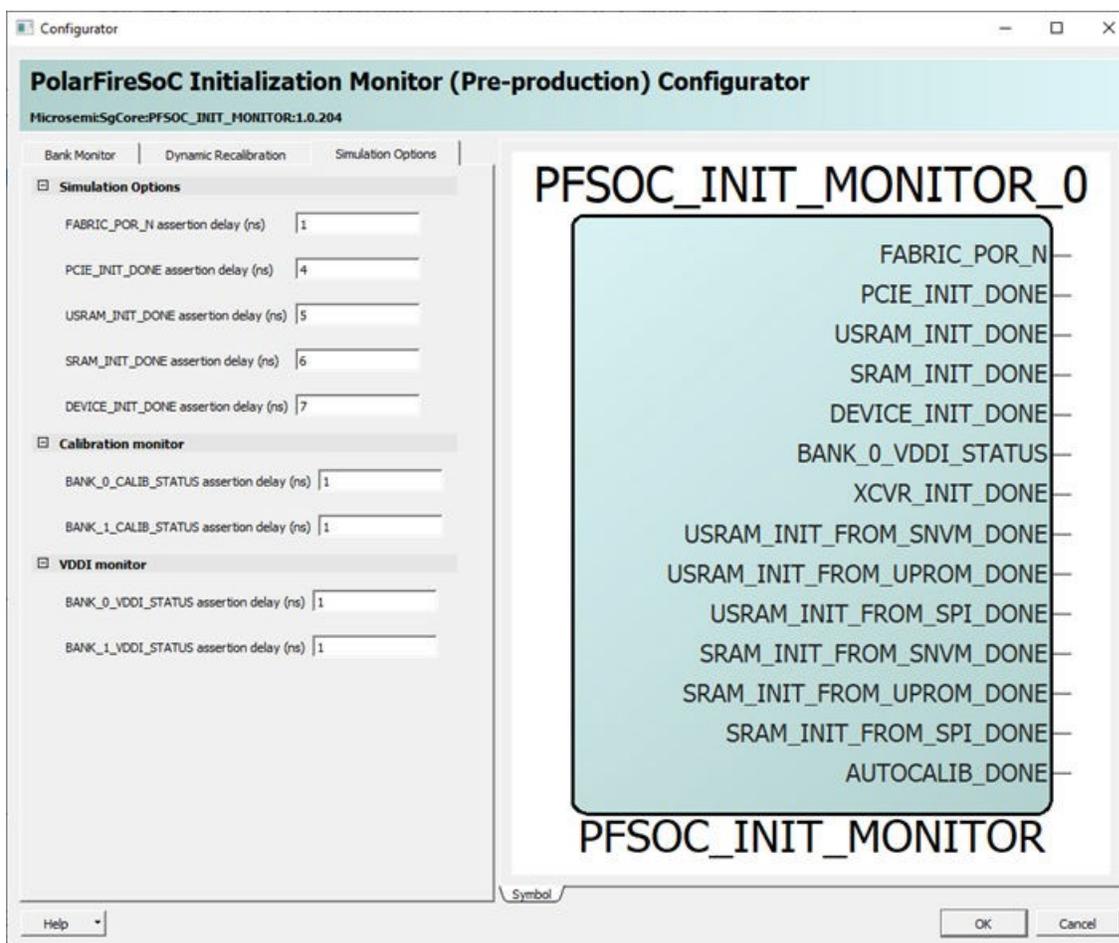
図 2-7 に、[Dynamic Recalibration] タブを示します。

図 2-7. PolarFire SoC 初期化モニタ コンフィグレータ - [Dynamic Recalibration]タブ



PolarFire SoC 初期化モニタはシミュレーションをサポートします。[Simulation Options]タブを使って、zero time instance から出力信号リリースまでの時間を指定します。図 2-8 に、[Simulation Options]タブを示します。

図 2-8. PolarFire SoC 初期化モニタ コンフィグレータ - [Simulation Options]タブ



Note: DDR コントローラのトレーニング ロジックを開始する前に、I/O をキャリブレートする必要があります。このために、PFSOC_INIT_MONITOR IP の DEVICE_INIT_DONE 信号と BANK_#_CALIB_STATUS 信号の間の AND (論理積)によってリセット信号を生成する必要があります。BANK_#は、DDR サブシステムが配置されているバンクを示します。

2.3.3 セキュア不揮発性メモリ(sNVM)

各デバイスは 56 K バイトの sNVM を備えています。このメモリは、データを平文として保存するか暗号化/認証済みデータとして保存するかに応じて 236 バイトまたは 252 バイト x 221 ページの構成になります。このメモリには、システム コントローラに対してシステムサービスを呼び出す事によりアクセスできます。sNVM 内のページは、ビットストリーム プログラミング中に ROM として指定できます。sNVM の内容は、LSRAM と μ SRAM を初期化するためのセキュアデータとして使えます。

自動初期化される LSRAM または μ SRAM がユーザ回路内に少なくとも 1 つ存在する場合、sNVM から RAM ブロックを初期化するのに要する時間(単位は ms) は下式により求められます。

$$t_{\text{LSRAM_pt_SNVM}} = ((14.0875 + (541.4125 \times L)) / 1000) \pm 6\%$$

$$t_{\text{USRAM_pt_SNVM}} = ((14.0875 + (29.325 \times L)) / 1000) \pm 6\%$$

L = 自動初期化される LSRAM の数

U = 自動初期化される μ SRAM の数

Note: $t_{\text{LSRAM_pt_SNVM}}$ および $t_{\text{USRAM_pt_SNVM}}$ 内の「pt」は、「平文(plaintext)」を意味します。

2.3.4 組み込み不揮発性メモリ(eNVM) (PolarFire SoC FPGA のみ)

PolarFire SoC FPGA は、1 つの組み込み不揮発性メモリ(eNVM)ブロック (サイズは 128 KB)を備えています。eNVM は 1 ビットエラー訂正/2 ビットエラー検出(SECDED)により保護され、長いデータ保持期間を有します。詳細は『[PolarFire FPGA and PolarFire SoC FPGA Security User Guide](#)』を参照してください。

2.3.5 μPROM

どちらのデバイスファミリもファブリックのボトムに配置された μPROM 行により最大 513K バイトの読み出し専用不揮発性メモリを提供します。アドレスバスは 16 ビット幅であり、データ読み出しバスは 9 ビット幅です。ファブリック ロジックは μPROM データの全体にアクセス可能です。

μPROM から自動初期化される LSRAM または μSRAM がユーザ回路内に少なくとも 1 つ存在する場合、初期化に要する時間(単位は ms) は下式により求められます。

$$t\text{LSRAM_pt_UPROM} = ((30.1325 + (663.7125 \times L)) / 1000) \pm 6\%$$

$$t\text{USRAM_pt_UPROM} = ((30.1325 + (28.75 \times U)) / 1000) \pm 6\%$$

L = 自動初期化される LSRAM の数

U = 自動初期化される μSRAM の数

Note: tLSRAM_pt_UPROM および tUSRAM_pt_UPROM 内の「pt」は、「平文(plaintext)」を意味します。

2.3.6 外部 SPI フラッシュ

システムコントローラの SPI インターフェイスを介して SPI フラッシュメモリを接続し、そこにプログラミングイメージを保存できます。システムコントローラは他社(Micron、Winbond、Spansion 社等)のデバイスをサポートします。

SPI フラッシュ内のファブリック SRAM (LSRAM と μSRAM)は認証なし平文初期化データ、認証済み平文初期化データ、認証/暗号化済み初期化データを使って初期化できます。

ユーザ回路が FPGA ファブリック LSRAM の自動初期化を要求しない場合、tLSRAM パラメータは 0 です。ユーザ回路が FPGA ファブリック μSRAM の自動初期化を要求しない場合、tUSRAM パラメータは 0 です。

2.3.6.1 認証なし平文初期化データ

SPI フラッシュから自動初期化される LSRAM または μSRAM がユーザ回路内に少なくとも 1 つ存在する場合、初期化に要する時間(ms) は下式により求められます。

$$t\text{LSRAM_pt} = [(((\text{ROUNDUP}(4.034 \times L) + 1) \times 8192 / f) + (130 \times L)) / 1000] + 1 \pm 6\%$$

$$t\text{USRAM_pt} = [(((\text{ROUNDUP}(0.144 \times U) + 1) \times 8192 / f) + (25 \times U)) / 1000] + 1 \pm 6\%$$

L = 自動初期化される LSRAM の数

U = 自動初期化される μSRAM の数

f = SPI クロックの周波数(MHz)

Note: tLSRAM_pt および tUSRAM_pt 内の「pt」は、「平文(plaintext)」を意味します。

2.3.6.2 認証済み平文初期化データ

認証済み平文初期化データを選択した場合、tLSRAM_pt および tUSRAM_pt タイミング パラメータに 103 ms ±6% を追加する必要があります。

Note: tLSRAM_pt および tUSRAM_pt 内の「pt」は、「平文(plaintext)」を意味します。

2.3.6.3 認証/暗号化済み初期化データ

暗号化した初期化データを使う場合、LSRAM の初期化に要する時間(ms) は下式により求められます。

$$t\text{LSRAM_enc} = t\text{LSRAM_pt} + t\text{LSRAM_auth} + ((\text{ROUNDUP}((L \times 2560) / 1024, 1) + 1) \times 1024 \times 8 / \text{Dlsram}) / 1000$$

tLSRAM_pt 内の「pt」は、「平文(plaintext)」を意味します。

Note: Dlsram は SPI SCK 周波数によって決まります。

表 2-1 に、LSRAM 暗号化データの分周比(Dlsram)設定を示します。

表 2-1. LSRAM 暗号化データの分周比(Dlsram)設定

SPI_SCK 周波数(MHz)	Dlsram
13.33	180
20	30
40	15

暗号化した初期化データを使う場合、μSRAM の初期化に要する時間(ms) は下式により求められます。

$$t\text{USRAM_enc} = t\text{USRAM_pt} + t\text{USRAM_auth} + ((\text{ROUNDUP}((U \times 2560) / 1024, 1) + 1) \times 1024 \times 8 / \text{Dusram}) / 1000$$

- $t\text{LSRAM_pt} = t\text{LSRAM}$
- $t\text{USRAM_pt} = t\text{USRAM}$
- $t\text{LSRAM_auth}$ および $t\text{USRAM_auth}$ 内の「auth」は、「認証済み平文(Authenticated Plaintext)」を意味します。
- $t\text{USRAM_pt}$ 内の「pt」は、「平文(plaintext)」を意味します。
- $t\text{LSRAM_auth} = t\text{USRAM_auth} = 103\text{ms} \pm 6\%$

Note: Dusram は SPI SCK 周波数によって決まります。

表 2-2 に、μSRAM 暗号化データの分周比(Dusram)設定を示します。

表 2-2. USRAM 暗号化データの分周比(Dusram)設定

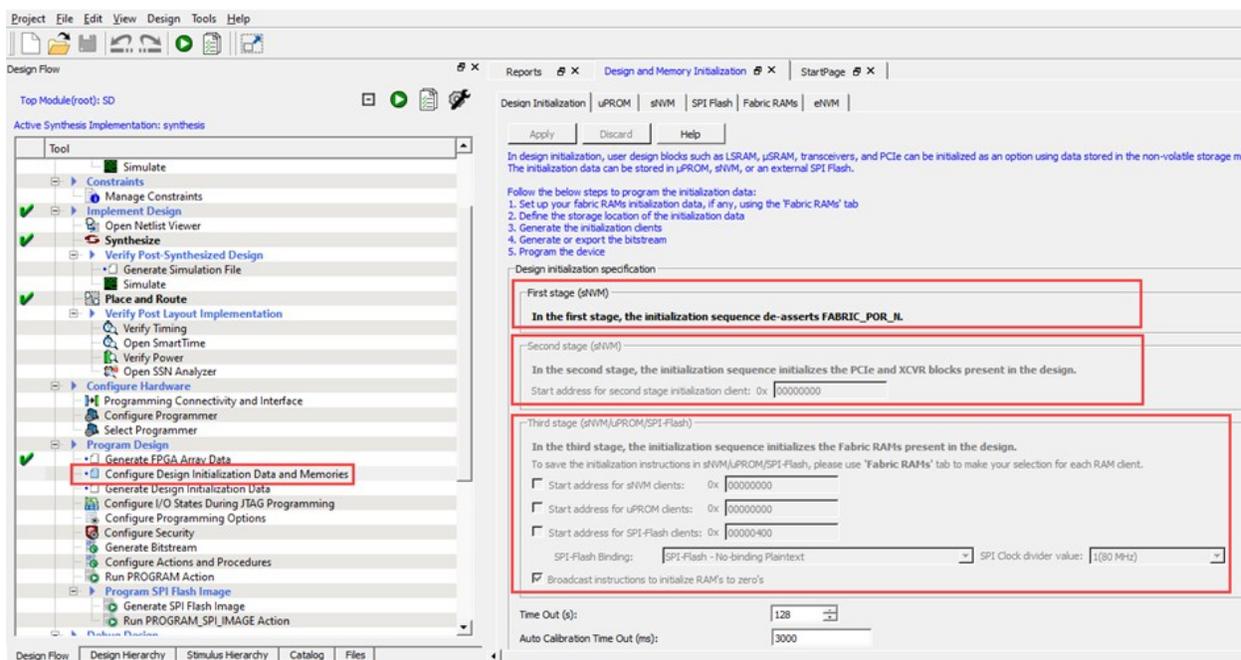
SPI_SCK 周波数(MHz)	Dusram
20	45
40	20

SPI_SCK 周波数が 13.3 MHz である場合、 $t\text{USRAM_enc} = t\text{USRAM_pt} + t\text{USRAM_auth} + 0.01$ となります。

2.3.7 回路/メモリ初期化の設定方法

以下では、Libero SoC 内の[Configure Design Initialization Data and Memories] オプションを使って PCIe、トランシーバ、ファブリック RAM ブロックを初期化する方法について説明します。回路/メモリの初期化は、図 2-9 に示す通り、3 つステージに分割されます。

図 2-9. 回路/メモリ初期化



1. ステージ 1 クライアントは FPGA ファブリックおよび関連 I/O を起動した後に、FABRIC_POR_N 信号をデassertします。このクライアントは、sNVM 内でアドレス空間の先頭に保存されます。
2. ステージ 2 クライアントは回路内に存在する PCIe および XCVR ブロックを初期化します。このクライアントは sNVM 内に保存されます(クライアントの開始アドレスは設定可能)。
3. ステージ 3 クライアントは回路内に存在するファブリック RAM を初期化します。回路内の各ロジカル RAM は各種ストレージタイプ(sNVM、μPROM、SPI フラッシュのいずれか)から初期化可能です。これらのストレージの開始アドレスは設定可能です。

Note: ステージ 2 クライアントは回路内に存在する PCIe および XCVR ブロックを初期化します。PCIe/XCVR レジスタの既定値を変更するには、カスタム設定用のテキストファイルをインポートします。これにより、ステージ 2 で生成されるアセンブリ ファイルは既定値フローから変更されます。レジスタの内容を変更するために使うテキストファイルの書式は以下の通りです。

Instance_Name/Register:Field_Name [Hex 値] (Hex 値の前にスペースが必要)

例: PF_PCIE_0/PCIESS_LANE0_Pipe_AXI0_SER_DRV_CTRL:TXDRVTRIM 0xFFFFFFFF

Note: SPI フラッシュから RAM を初期化する場合、IO_CFG_INTF ピンを「1」に設定する事によってシステム コントローラ SPI インターフェイスをマスターモードにする必要があります。

Note: SPI クロック分周値は、SPI フラッシュから初期化データを読み出すために必要な SPI SCK 周波数を設定します。SPI クロック分周値は、外部 SPI フラッシュの動作周波数レンジに基づいて選択する必要があります。

表 2-3. SPI クロック分周値

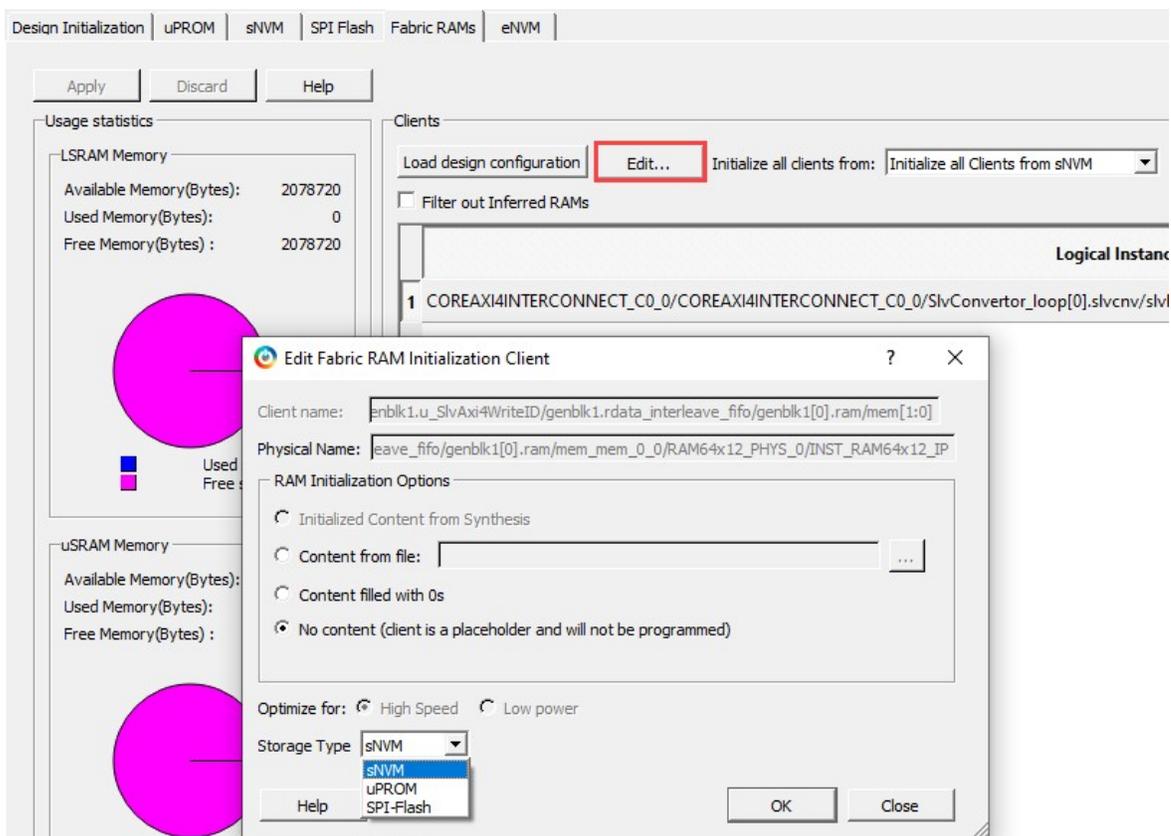
SPI クロック分周値	SCK 周波数
1	80 MHz
2	40 MHz
4	20 MHz
6	13.3 MHz

電源投入時のファブリック RAM 初期化方法は、以下の手順により設定します。

1. **[Fabric RAMs]** タブから必要なロジカル RAM を選択して**[Edit]**をクリックします。「**Edit Fabric RAM Initialization Client**」ウィンドウは以下のオプションを提供します。
 - **Content from file:** Intel-Hex (*.hex)、Simple-Hex (*.shx)、Motorola-S (*.s)、Microchip Binary (*.mem)のどれからクライアントを初期化するのか選択します。

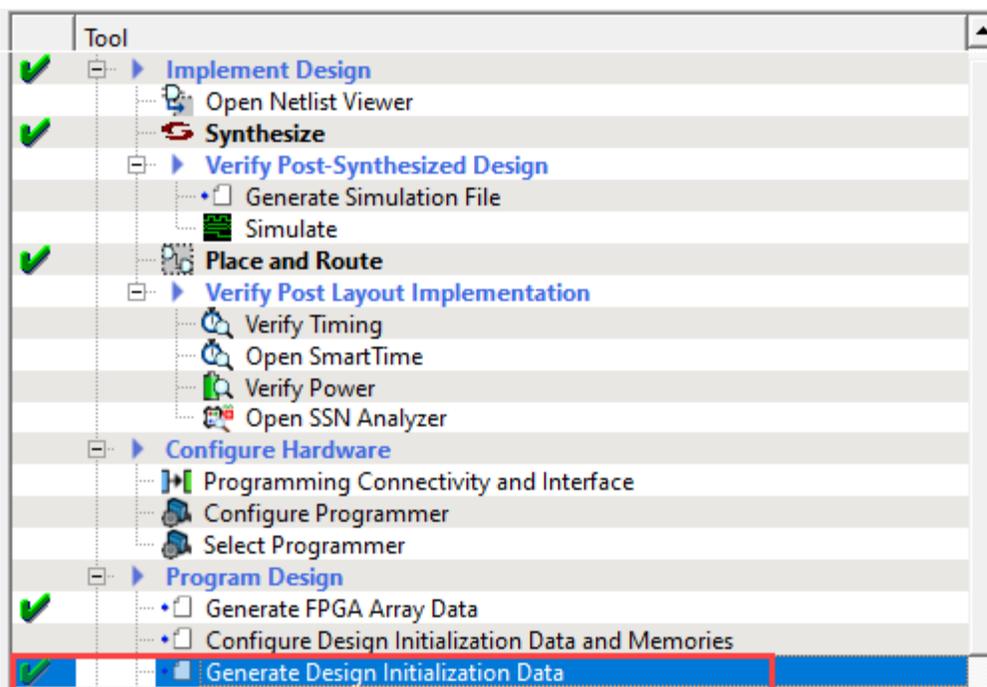
- **Content filled with 0s:** クライアントを 0 で初期化します。
- **No content:** 内容を持たないプレースホルダとしてクライアントを作成します。
- **Storage Type:** クライアント向けにストレージのタイプを選択します。

図 2-10. ファブリック RAM



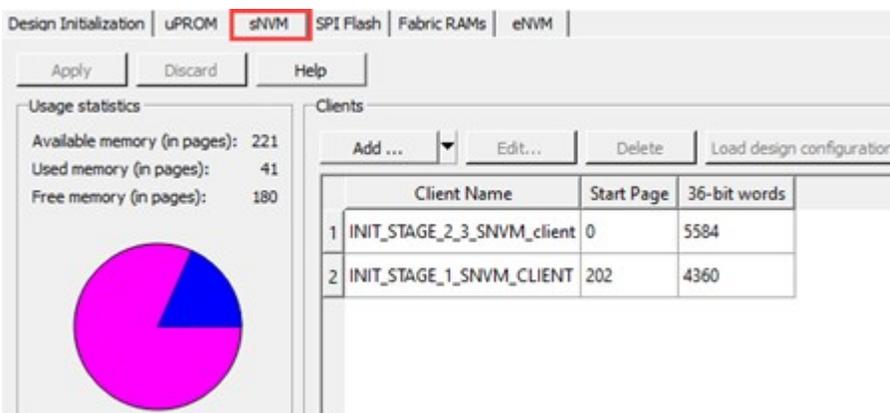
2. RAM 初期化クライアントを設定した後に、**[Fabric RAMs]**タブ上で**[Apply]**をクリックします。
3. **[Design Flow]**タブの下で**[Generate Design Initialization Data]**を選択します。これにより、自動的にステージ 1/2/3 の初期化クライアントが生成され、ユーザが選択した不揮発性メモリに追加されます。図 2-11 に示す通り、**[Generate Design Initialization Data]** がハイライト表示されます。

図 2-11. Generate Design Initialization Data



初期化クライアントは、図 2-12 内の対応するタブ([uPROM]、[sNVM]、[SPI Flash]、[Fabric RAMs]、[eNVM]のいずれか)に追加されます([eNVM]タブは PolarFire SoC FPGA 向けにのみ利用可能)。

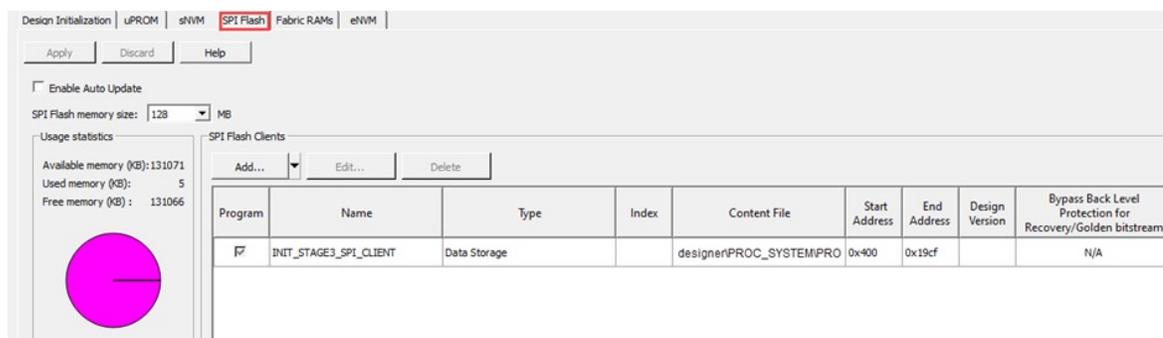
図 2-12. sNVM 内で生成された初期化クライアント



Note: ユーザ回路が PCIe または XCVR ブロックを含んでいる場合、ステージ 2 のクライアントは[sNVM]タブに追加されます。

ストレージタイプとして SPI フラッシュが選択されている場合、図 2-13 に示す通り、初期化クライアントは[SPI Flash]タブに追加されます。

図 2-13. SPI フラッシュ上のステージ 3 初期化クライアント



4. ステージ 3 向けに外部 SPI フラッシュが選択された場合、図 2-14 に示す[Run PROGRAM Action]を完了する前に、[Design Flow]タブから[Generate SPI Flash Image]と[Run PROGRAM_SPI_IMAGE Action]を生成する必要があります。

図 2-14. Program SPI Flash Image



これらの手順により、回路内に存在する PCIe、XCVR、ファブリック RAM は、ユーザの選択に基づいて不揮発性メモリ内に配置された初期化クライアントを使って、電源投入中に初期化されます。

2.3.7.1 SPI フラッシュ クライアントの設定

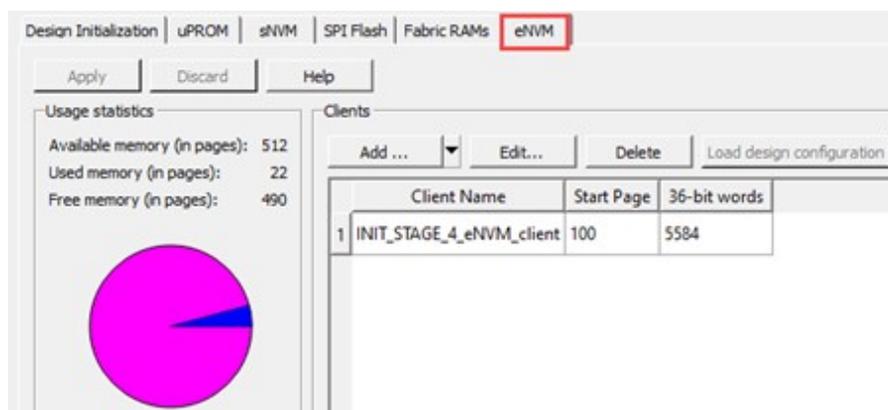
SPI フラッシュに保存されたメモリ初期化データは、暗号化してデバイスにバインドできます。[Design Initialization]タブは、以下の暗号化/バインド オプションを提供します。

- **SPI Flash – Binding Encrypted with Default Key**
KLK が認証および暗号化/復号用のルート鍵として使われます。
- **SPI Flash – Binding Encrypted with User Encryption Key 1 (UEK1)**
UEK1 が認証および暗号化/復号用のルート鍵として使われます。
- **SPI Flash – Binding Encrypted with User Encryption Key 2 (UEK2)**
UEK2 が認証および暗号化/復号用のルート鍵として使われます。

2.3.7.2 eNVM クライアントの設定(PolarFire SoC FPGA のみ)

ストレージタイプとして eNVM が選択されている場合、図 2-15 に示す通り、初期化クライアントは[eNVM]タブに追加されます。初期化クライアントが[eNVM]タブに追加されたら、[Design Flow]タブから[Generate Bitstream]をダブルクリックします。

図 2-15. eNVM 内のステージ 4 初期化クライアント



2.3.8 PUFT (Power-up To Functional Time)データレポート

Libero SoCの「Design Initialization Data and Memories」レポートは、PUFT タイミング データに関する情報を提供します。電源投入後のデバイス初期化時に、各ブロック(PCIe、XCVR、RAM 等)の初期化完了を示す信号がアサートされます。例えば、PCIeに関連する全てのレジスタが設定された後に PCIE_INIT_DONE 信号がアサートされます。初期化の最後に DEVICE_INIT_DONE 信号がアサートされます。

LSRAM および μ SRAM ブロックの場合、初期化に要する時間はデバイス上で計測されるブロックの数に応じて異なります。信号の PUFT を計算するために、1 ブロックの初期化に要した平均時間が使われます。システム コントローラは、最初のブロックの初期化データをコピーする時にだけ一定の時間(Constant_Copy_Time)を必要とします。後続ブロックの初期化データは、直前のブロックの初期化中にバックグラウンドでコピーされます。

$PUFT_{\text{signal}} = \text{Constant_Copy_Time} + (\text{Number_Of_Blocks} \times \text{Average_Time_to_Init_One_Block}) \text{ ns}$

これらの信号は順次アサートされるため、各信号の PUFT は以下に依存します。

- シーケンス内の直前のブロックがユーザ回路内でインスタンス化されるかどうか
- ブロックの設定(設定によっては追加のレジスタが必要になる事がある)

LSRAM と μ SRAM は各種ストレージ(sNVM、 μ PROM、SPI のいずれか)から初期化可能です。SPI から初期化する場合、SPI クロック分周値と暗号化タイプに応じて追加のデータを収集する必要があります。

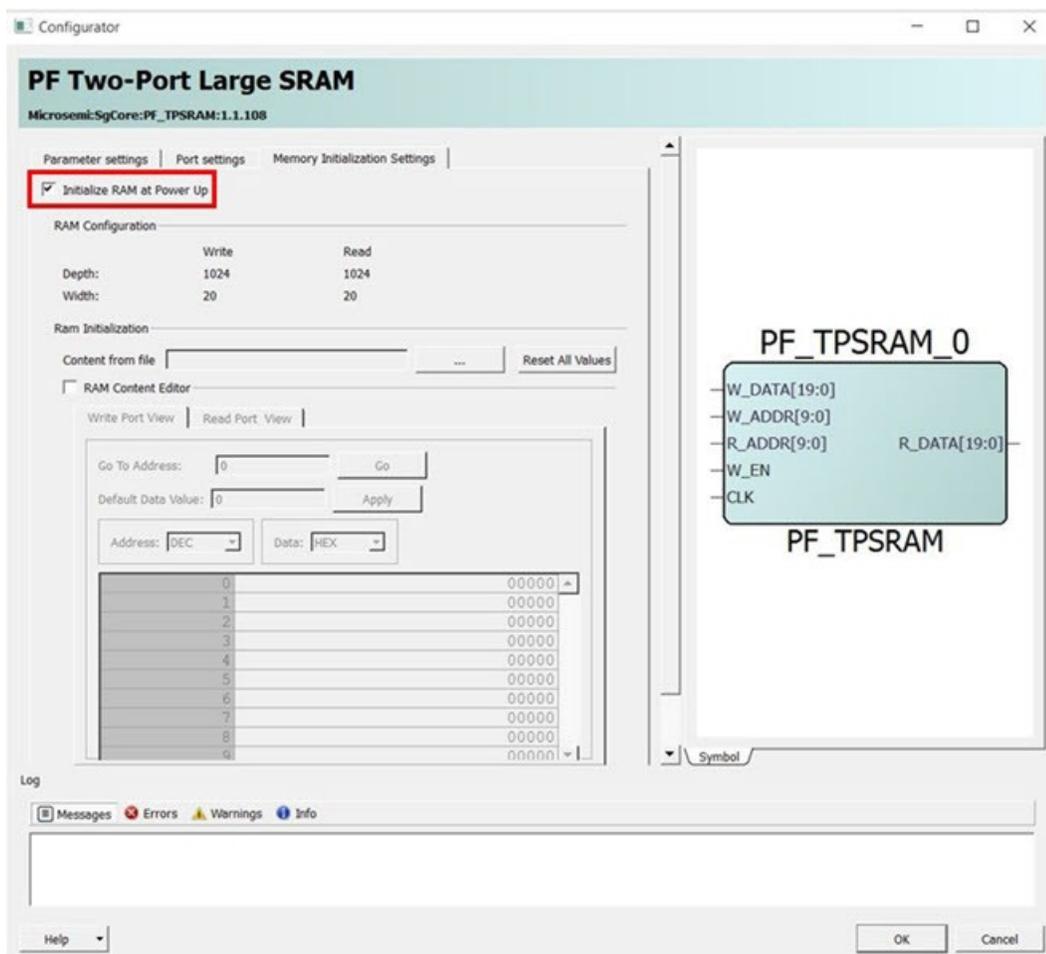
2.3.9 Place and Route 前の RAM 初期化

ファブリック RAM IP コアの作成時に、シミュレーション用の content ファイルをインポートできます。content ファイルのパスは保存され、回路/メモリ初期化ステージに渡されます。

LSRAM および μ SRAM コンフィグレータを使ってファブリック RAM を初期化する手順は以下の通りです。

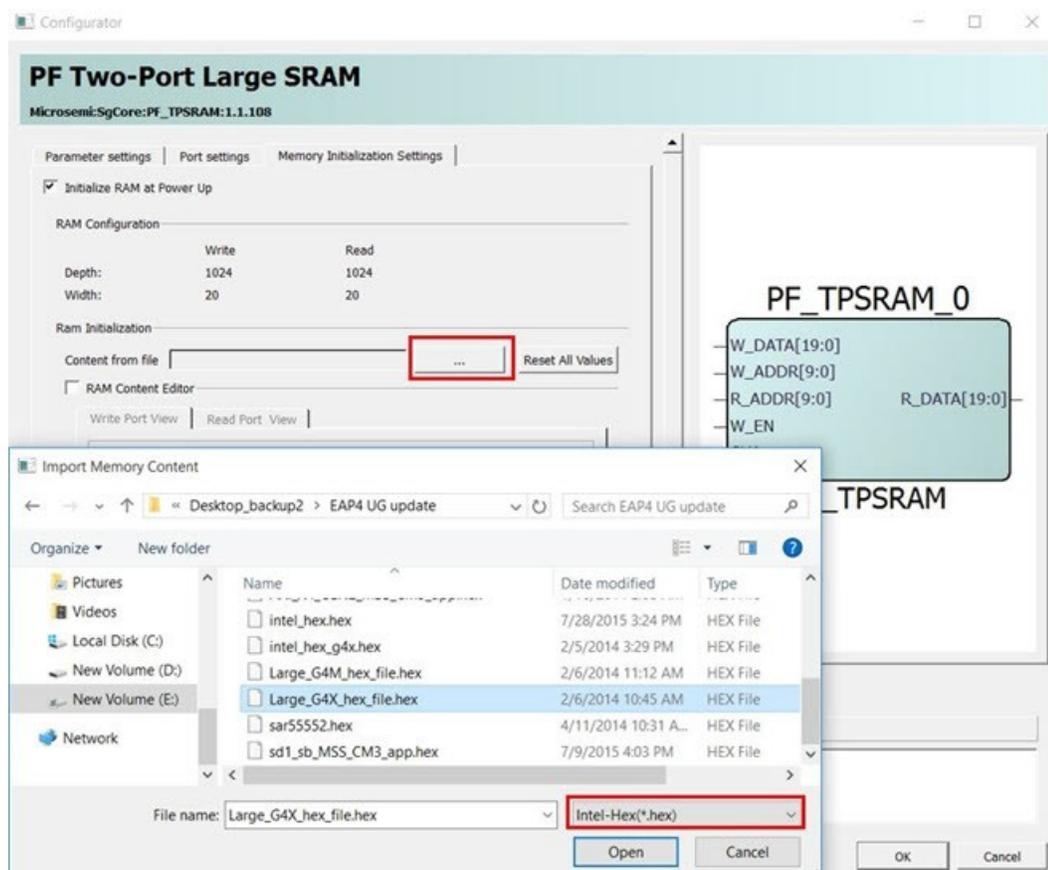
1. 「PF Two-Port Large SRAM Configurator」ウィンドウ内で[Memory Initialization Settings]タブを選択します。次に、図 2-16 に示す[Initialize RAM at Power-up]チェックボックスを有効にします。

図 2-16. 「PF Two-Port Large SRAM Configurator」 ウィンドウ



2. **[Import File]** オプションを選択し、「**Import Memory Content**」ダイアログボックスからメモリ content ファイル(Intel-Hex)をインポートします(図 2-17 参照)。インポート時にファイル拡張子は **Intel-Hex** ファイル向けの「*.hex」に設定されます。インポートされたメモリ内容は「**RAM Content Editor**」枠内に表示されます。

図 2-17. 「Import Memory Content」 ダイアログ ボックス



3. Place and Route 後に、content ファイルのストレージタイプを選択し、2.3.7. 「回路/メモリ初期化の設定方法」に記載した手順で初期化クライアントを生成します。

LSRAM および μ SRAM コンフィグレータとユーザオプションの詳細は、『PolarFire FPGA and PolarFire SoC FPGA Fabric User Guide』内の「Embedded Memory Blocks」を参照してください。

2.4 MSS プリブート(PolarFire SoC FPGA のみ)

回路の初期化が完了して DEVICE_INIT_DONE がアサートされると、MSS プリブートの実行が始まります。全ての起動プロセスが完了した後に、MSS はリセット状態から解除されます。システム コントローラはデバイスプログラミング、初期化、設定を行います。ES (Engineering Sample) デバイスの場合、プログラミング済みデバイスがシステム コントローラ サスペンドモード向けに設定されていると MSS プリブートは発生しません。

初期化の MSS プリブート フェイズは、システム コントローラ ファームウェアにより管理されますが、プリブートシーケンスの特定部分を実行するために MSS コア コンプレックス内の E51 (RISC-V Monitor Core)を使う事もできます。

MSS プリブート ステージ中に以下のイベントが発生します。

- MSS 組み込み不揮発性メモリ(eNVM) の電源投入
- MSS コア コンプレックス L2 キャッシュの ECC に関する回路の初期化
- ユーザ ブートコードの認証(ユーザ セキュアブート オプションが有効である場合)
- ユーザ ブートコードへの設定済み MSS の引き渡し

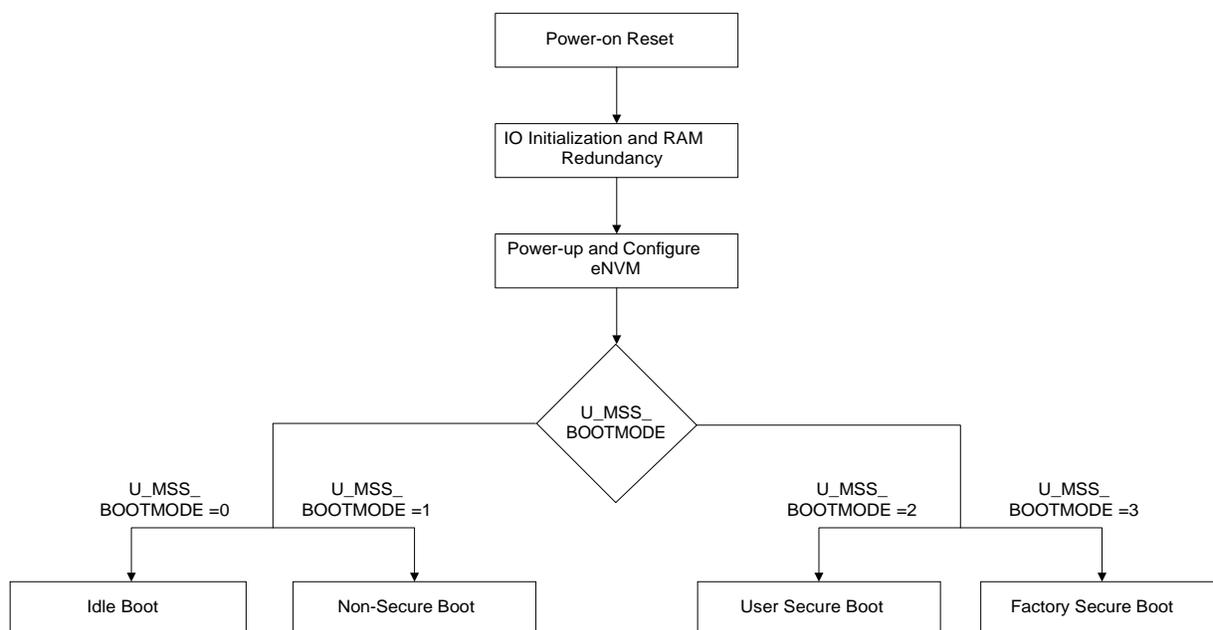
MSS コア コンプレックスは、4 通りのモードでブートできます。表 2-4 に、設定可能な MSS プリブート オプションの一覧を示します。これらの設定は sNVM に書き込む事ができます。ブートモードは、ユーザ パラメータ U_MSS_BOOTMODE[1:0] により定義されます。モードに依存するその他のブート設定データは、ユーザ パラメータ U_MSS_BOOTCFG (表 2-6、表 2-8 参照) により定義されます。

表 2-4. MSS コア コンプレックスのブートモード

U_MSS_BOOTMODE[1:0]	モード	概要
0	アイドルブート	MSS が設定されていない場合、MSS コア コンプレックスはブート ROM からブートします。
1	非セキュアブート	コア コンプレックスは U_MSS_BOOTADDR によって定義されたアドレスから直接ブートします。
2	ユーザ セキュアブート	MSS コア コンプレックスは sNVM からブートします。
3	ファクトリ セキュアブート	MSS コア コンプレックスは工場で設定済みのセキュアブート プロトコルを使ってブートします。

ブート オプションは、Libero Design Flow の一部として選択されます。モードを変更するには、新しい FPGA プログラミング ファイルを生成する必要があります。

図 2-18. MSS プリブートフロー

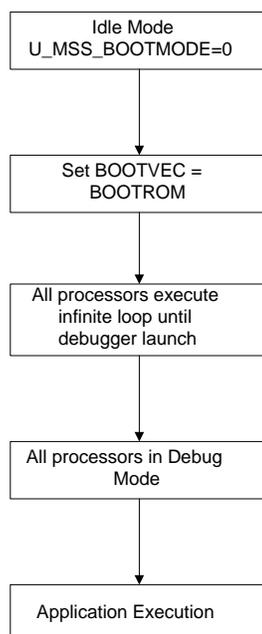


2.4.1 アイドルブート

MSS が設定されていない場合(例: ブランクデバイスの場合)、MSS コア コンプレックスはブート ROM プログラムを実行します。このプログラムは、デバッガがターゲットに接続されるまで全プロセッサを無限ループさせて保持します。ブートベクタ レジスタは、デバイスがリセットされるか新しいブートモード設定が書き込まれるまで値を保持します。設定済みデバイスの場合、このモードは Libero コンフィグレータ内の U_MSS_BOOTMODE=0 ブートオプションを使って実装できます。

Note: アイドルブート モードでは U_MSS_BOOTCFG は使われません。図 2-19 に、アイドルブート フローを示します。

図 2-19. アイドルブートフロー



2.4.2 非セキュアブート

このモードでは、MSS コア コンプレックスは指定された eNVM アドレスから認証なしで実行します。これは最速のブートオプションですが、コードイメージの認証は行われません。アドレスは、Libero コンフィグレータ内の U_MSS_BOOTADDR により指定できます。このモードを使うと、FIC (Fabric Interface Controller)を介して任意の FPGA ファブリックメモリアドレスからブートすることもできます。このモードを実装するには、U_MSS_BOOTMODE=1 ブートオプションを使います。

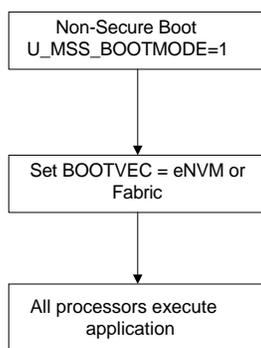
MSS コア コンプレックスは、U_MSS_BOOTCFG によって定義されたブートベクタ(表 2-5 参照)を使ってリセットから解除されます。

表 2-5. 非セキュアブートモードにおける U_MSS_BOOTCFG の設定

オフセット (バイト)	サイズ (バイト)	名称	概要
0	4	BOOTVEC0	E51 向けブートベクタ
4	4	BOOTVEC1	U540 向けブートベクタ
8	4	BOOTVEC2	U541 向けブートベクタ
16	4	BOOTVEC3	U542 向けブートベクタ
20	4	BOOTVEC4	U543 向けブートベクタ

図 2-20 に、非セキュアブート フローを示します。

図 2-20. 非セキュアブートフロー



2.4.3 ユーザセキュアブート

このモードを使うと、ユーザ独自のカスタムセキュアブートを実装し、ユーザセキュアブートコードを sNVM 内に格納できます。sNVM は 56K バイトの不揮発性メモリであり、内蔵 PUF (Physically Unclonable Function) により保護可能です。sNVM ページは改変不可能な ROM として指定されているため、このブートモードはセキュアであると見なされ、電源投入時にシステムコントローラは sNVM 内のユーザセキュアブートコードを E51 モニタコアの DTIM(Data Tightly Integrated Memory)へコピーした後、即座に E51 がユーザセキュアブートコードの実行を開始します。

ユーザセキュアブートコードのサイズが DTIM のサイズより大きい場合、ブートコードを 2つのステージに分割する必要があります。ユーザブートシーケンスの第 2 ステージも sNVM に格納し、ユーザ認証/復号アルゴリズムを使って認証できます。

認証済みまたは暗号化済みページが使われる場合、全ての認証/暗号化済みページに対して同じ USK 鍵(すなわち U_MSS_BOOT_SNVN_USK)が使われる必要があります。

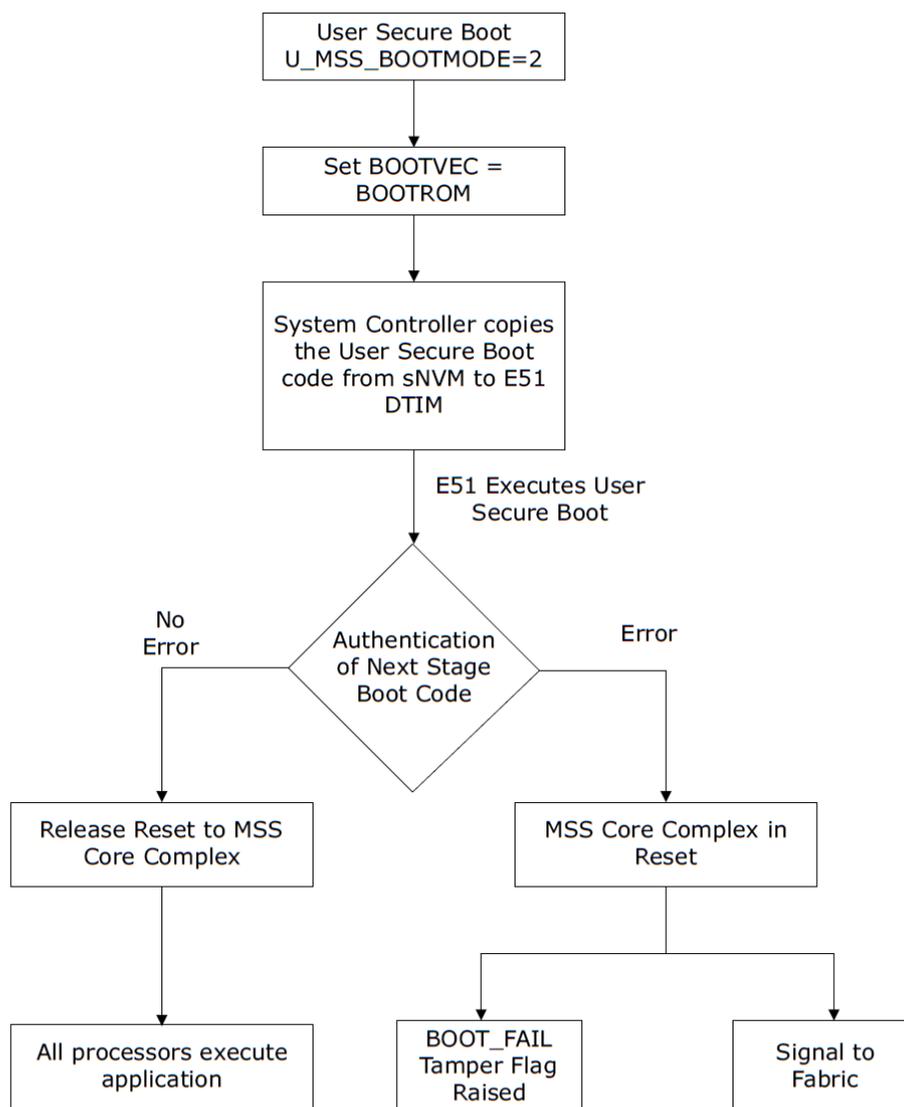
認証に失敗した場合、MSS コアコンプレックスをリセット状態にして BOOT_FAIL タンパーフラグをセットできます。このモードを実装するには、U_MSS_BOOTMODE=2 ブートオプションを使います。

表 2-6. ユーザセキュアブートモードにおける U_MSS_BOOTCFG の設定

オフセット (バイト)	サイズ (バイト)	名称	概要
0	1	U_MSS_BOOT_SNVN_PAGE	sNVM 内の開始ページ
1	3	RESERVED	アライメント用
4	12	U_MSS_BOOT_SNVN_USK	認証/暗号化済みページ向けに使う USK 鍵

図 2-21 に、ユーザセキュアブートフローを示します。

図 2-21. ユーザ セキュアブート フロー



2.4.4 ファクトリ セキュアブート フロー

このモードでは、システム コントローラは eNVM から SBIC (Secure Boot Image Certificate)を読み出して検証します。検証に成功すると、システム コントローラは非公開セキュアメモリ領域内のファクトリ セキュアブート コードを E51 モニタコアの DTIM へコピーします。既定値セキュアブートは、eNVM 内に保存されている SBIC を使って、eNVM イメージの署名を確認します。エラーが報告されなければ、MSS コア コンプレックスはリセットから解除されます。エラーが報告された場合、MSS コア コンプレックスはリセット状態に置かれ、BOOT_FAIL タンパーフラグがセットされます。次にシステム コントローラはタンパーフラグをアクティブにする事で、FPGA ファブリックへの信号をアサートします。ユーザはこの信号を監視する事で、エラーを検出できます。このモードを実装するには、U_MSS_BOOTMODE=3 ブートオプションを使います。

SBIC は、保護されたバイナリ BLOB のアドレス/サイズ/ハッシュ/ECDSA (Elliptic Curve Digital Signature Algorithm)署名を格納します。ECDSA は、楕円曲線暗号を使うデジタル署名アルゴリズム(DSA)です。SEBIC は、システム内の各ハードウェア スレッド(Hart)/コア/プロセッサコア向けのリセットベクタも格納します。

表 2-7. SBIC (Secure Boot Image Certificate)

オフセット(バイト)	サイズ(バイト)	値	概要
0	4	IMAGEADDR	MSS メモリマップ内の UBL(User Boot Loader) のアドレス

.....続き			
オフセット(バイト)	サイズ(バイト)	値	概要
4	4	IMAGELEN	UBL のサイズ (バイト数)
8	4	BOOTVEC0	UBL 内の E51 向けブートベクタ
12	4	BOOTVEC1	UBL 内の U540 向けブートベクタ
16	4	BOOTVEC2	UBL 内の U541 向けブートベクタ
20	4	BOOTVEC3	UBL 内の U542 向けブートベクタ
24	4	BOOTVEC4	UBL 内の U543 向けブートベクタ
28	1	OPTIONS[7:0]	SBIC オプション
28	3	RESERVED	—
32	8	VERSION	SBIC/イメージバージョン
40	16	DSN	オプションの DSN バインド
56	48	H	UBL イメージ SHA-384 ハッシュ
104	104	CODESIG	DER エンコード形式の ECDSA 署名
Total	208	バイト	—

DSN

DSN フィールドの値が非 0 である場合、その値はデバイスのシリアル番号と比較されます。比較に失敗した場合、BOOT_FAIL タンパーフラグがセットされ、認証は中止されます。

VERSION

U_MSS_REVOCATION_ENABLE によって SBIC 失効が有効にされている場合、VERSION の値が失効しきい値未満であれば SBIC は拒絶されます。

SBIC 失効オプション

U_MSS_REVOCATION_ENABLE によって SBIC 失効が有効にされ、かつ OPTIONS[0] が「1」である場合、VERSION 値未満の SBIC バージョンは SBIC の認証完了時に失効します。失効しきい値は、OPTIONS[0] = 1 かつより高い VERSION フィールド値を持つ SBIC によって更新されるまで、現在の値を保持します。失効しきい値はこの方法によってのみ更新(インクリメント)可能であり、ビットストリームによってのみリセット可能です。

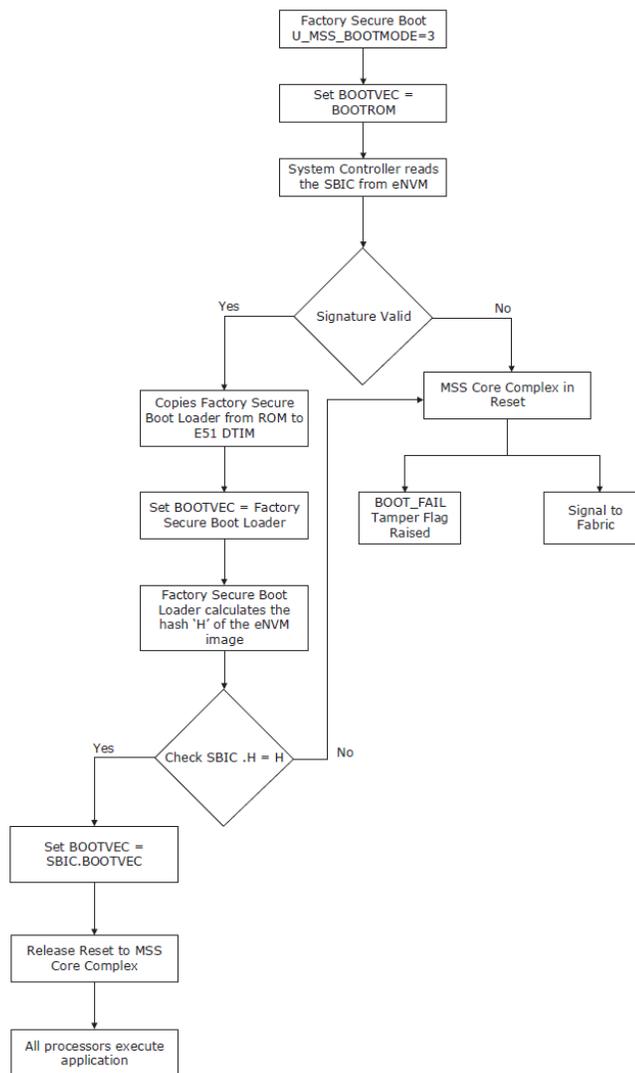
失効しきい値が動的に更新される場合、しきい値はパスワード向けに用いられる冗長ストレージ方式を使って保存されるため、デバイスブート中に停電が発生しても電源が回復した後にデバイスを正常にブートできます。失効しきい値の更新に失敗した場合、しきい値は新しい値または 1 つ前の値に確実に設定されます。

表 2-8. ファクトリ セキュアブート モードにおける U_MSS_BOOTCFG の設定

オフセット(バイト)	サイズ(バイト)	名称	概要
0	4	U_MSS_SBIC_ADDR	MSS アドレス空間内の SBIC のアドレス
4	4	U_MSS_REVOCATION_ENABLE	非 0 値である場合に SBIC 失効は有効

図 2-22 に、ファクトリ セキュアブート フローを示します。

図 2-22. ファクトリ セキュアブート フロー



2.5 MSS ユーザブート(PolarFire SoC FPGA のみ)

MSS ユーザブートは、制御がシステム コントローラから MSS コア コンプレックスに渡された時に発生します。MSS プリブートに成功すると、システム コントローラは MSS コア コンプレックスをリセットから解除します。MSS は、以下のいずれかによりブートアップできます。

- ベアメタル アプリケーション
- Linux アプリケーション
- AMP アプリケーション

MSS ブートの詳細は、『[PolarFire SoC Software Development and Tool Flow User Guide](#)』を参照してください。

2.6 HSIO/GPIO バンクの初期化

未使用 GPIO/HSIO バンクには電源を投入してもしなくても構いません。デバイスの電源投入中に、使用される GPIO/HSIO バンクの電源は他の全ての電源と同時に投入されます。ファブリックの電源投入時に、全てのバンクはフラッシュ コンフィグレーション ビットを使って自動的に初期化されます。電源が投入される全てのユーザ I/O (HSIO/GPIO)では、電源投入時に初期 PVT キャリブレーションが実行されます。

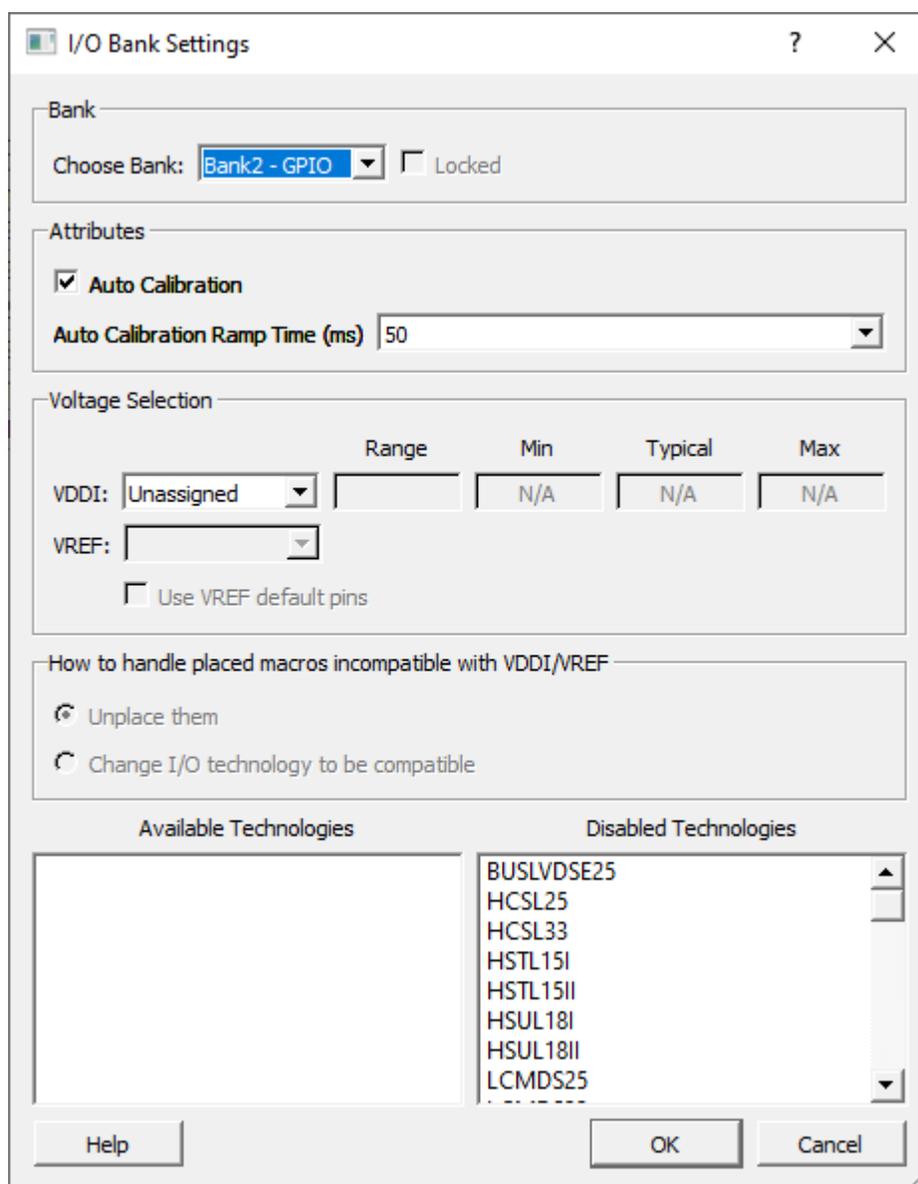
I/O が動作可能になるタイミングは、以下の要因により決まります。

- デバイスブート
- I/O バンクに印加される電源のランプアップ時間
- I/O のキャリブレーション時間 (例: DDR インターフェイス)

100 MHz 未満の低速動作の場合、I/O は I/O バンクに印加された電源電圧がしきい値レベルを超えた後に動作可能となります。GPIO が OUTPUT PAD 信号に割り当てられた場合、初期キャリブレーションの完了時に駆動強度が低減されます。なぜなら、I/O キャリブレーション ステータス(AUTOCALIB_DONE)信号がアサートされる前の初期駆動強度はキャリブレーション用の最大値に設定されるからです。駆動強度は変動するため、信号が低速であっても高速であっても、I/O キャリブレーション ステータス(AUTOCALIB_DONE)信号を監視する事を推奨します。

デバイスブートの完了後即座に I/O を使う必要がある場合、I/O 電源のランプ時間が十分に短い必要があります。I/O 電源のランプアップを低速にする(または遅延させる)事により、I/O バンクが使用可能になるまでの時間を遅らせる事も可能です。I/O バンクに低ランプレートの電源を使う場合、ファブリック内のユーザロジックは、I/O が低速 I/O として使用可能になった事を検知するために I/O バンクの状態を監視する必要があります。

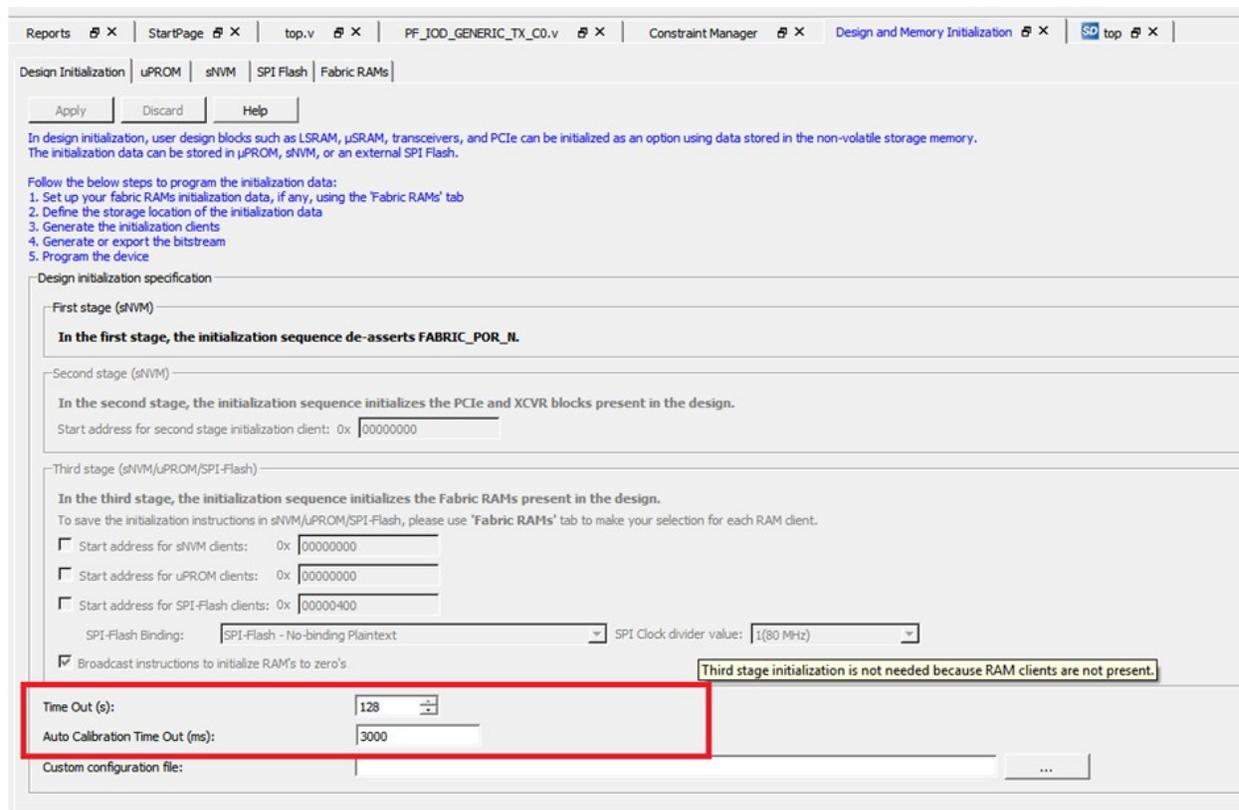
図 2-23. ランプアップ時間



高速 I/O キャリブレーション プロセスは自動的に発生します。I/O を高速アプリケーション向けに使う場合、FPGA ファブリック内のユーザ回路は I/O キャリブレーションの完了(AUTOCALIB_DONE のアサート)を監視する必要があります。

I/O キャリブレーションとバンク電源のステータスは、PolarFire/PolarFire SoC 初期化モニタ IP のステータス信号を使って監視できます。図 2-24 に、自動キャリブレーションのタイムアウト設定を示します。[Time Out (ms)] オプションは、ユーザ回路内で選択されている GPIO または HISO に基づいて設定します。

図 2-24. 自動キャリブレーションのタイムアウト



PolarFire/PolarFire SoC 初期化モニタは、BANK_#_CALIB_STATUS および BANK_#_VDDI_STATUS 信号をファブリックに対してアサートします。ユーザロジックは BANK_#_CALIB_STATUS を使って、各 I/O バンクのキャリブレーションが完了したかどうかを判別できます。BANK_#_VDDI_STATUS 信号は、各 I/O バンクの VDDI 電源を監視するために使えます。

DRI クロック(DRI_CLK) は、デバイスが完全に初期化されて DEVICE_INIT_DONE がアサートされるまで遮断する必要があります。

2.7 I/O の再キャリブレーション

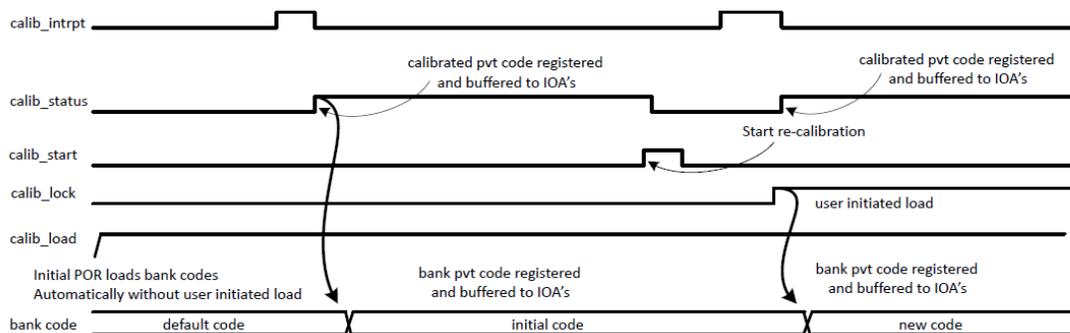
I/O 再キャリブレーションの制御用または初期 I/O キャリブレーションの監視用に PF_INIT_MONITOR IP および PFSOC_INIT_MONITOR IP が使われます。ユーザは I/O 再キャリブレーション機能を使ってキャリブレーションをシステムの起動シーケンスに合わせて(再)実行する事で、I/O 性能に対する VT(電圧/温度)変動の影響に対処できます。図 2-25 に再キャリブレーション動作を示します。

I/O キャリブレーションを有効にすると、以下のポートが PFSOC_INIT_MONITOR IP 内で利用可能となります。

- BANK_#_CALIB_STATUS
- BANK_#_CALIB_INTERRUPT
- BANK_#_CALIB_LOCK
- BANK_#_CALIB_LOAD
- BANK_#_CALIB_START

図 2-25 に再キャリブレーション動作を示します。

図 2-25. 再キャリブレーション動作



再キャリブレーション動作の手順は以下の通りです。

1. 新しいキャリブレーションを開始します。
 - ユーザが「`bank#_calib_start = 1`」に設定する事によりキャリブレーション シーケンスを開始します。
2. キャリブレーション エンジン新しいコードが存在する事を示します。
 - キャリブレーション エンジンは信号「`bank#_calib_interrupt = 1`」を生成します。
 - この時点で新しいキャリブレーション コードは使用可能になりますが、ユーザが要求するまで I/O に適用されずに保持されます。
3. ユーザは、新しいコードを I/O へ適用するようキャリブレーション エンジンに要求します。
 - ユーザが「`bank#_calib_lock = 1`」に設定する事により、新しいコードがラッチされて I/O へ適用されます。
4. キャリブレーション エンジンは新しいコードのラッチが完了した事を示します。
 - キャリブレーション エンジンは信号「`bank#_calib_status = 1`」を生成する事で、キャリブレーション が完了した事をユーザと I/O に知らせます。

Note: `bank#_calib_load` は High に固定する必要があります。

2.7.1 I/O Editor

バンクレベル コードの代わりに LANECTRL からのコードを使うように I/O を設定することができます。コードは、初期キャリブレーション後に LANECTRL に書き込まれます。書き込まれたコードは、後続の再キャリブレーションの影響を受けません。これにより、I/O を再キャリブレーションから保護できます。新しいコードのロード中に出力でグリッチが発生する可能性があるため、動作中に I/O が 3 ステートにならないよう、またはグリッチによって誤った出力リセットやクロックパルスが生じないよう、I/O を保護する事が重要です。I/O Editor 内の[**Use IO calibration from the lane**] は、そのバンクに対して初期キャリブレーションが有効である場合にのみ、LANECTRL からのコードを選択します。I/O を再キャリブレーションから保護し、かつ LANECTRL コードを使う場合、IOA PCBIT(USE_LANE_CALIB_CODE) を '1'b1 に設定する必要があります。その他の場合、既定値('1'b0)から変更してはいけません。

2.8 トランシーバの初期化

トランシーバの電源投入には VDDA、VDDA25、VDD_XCVR_CLK が影響します。

VDD_XCVR_CLK は、トランシーバ向けに外部参照クロックが使われる場合に適用可能です。電源の一覧は、6. 「補遺: 電源」を参照してください。電源投入中に、参照クロックとデータビットでグリッチが発生する可能性があります。トランシーバはフラッシュ コンフィグレーション ビットと回路初期化クライアントにより、sNVM から初期化されます。

トランシーバが完全に設定されると、PFSOC_INIT_MONITOR からの XCVR_INIT_DONE/DEVICE_INIT_DONE 信号が High に遷移します。XCVR クロックを使うユーザロジックは、XCVR_INIT_DONE 信号がアサートされるまでリセット状態を保持する必要があります。

トランシーバのデータピンは、電源投入時にホットプラグ モードになります。プログラミング ビットを使って早期に TX および RX 終端抵抗を検出する事で、PCI Express 等の規格における高速なレーバ検出が可能となります。

トランシーバ サブシステムの初期化に要する時間は、設定する必要のあるレーンと高速シリアル プロトコルの数によって決まります。1 レーンの QUADO が Libero 既定値を使って初期化される場合、XCVR 初期化時間は 282 μ s です(表 2-9 参照)。

最悪条件での遅延時間は、ステージ 2 アセンブリ ファイル内の全ての PCS および PMA レジスタ書き込み時間を含めて計算されます。ステージ 2 の詳細は、「[回路/メモリの初期化](#)」を参照してください。

表 2-9. XCVR 初期化時間

フロータイプ	ステージ 2 アセンブリ ファイル内のレジスタ書き込み数	XCVR 初期化時間
既定値フロー (生成ファイルに対する変更なし)	61	282 μ s
アセンブリ ファイル変更フロー	136	594 μ s

2.9 ユーザ PLL/DLL の初期化

ファブリックの電源投入時に、PLL と DLL の両方がフラッシュ コンフィグレーション ビットを使って自動的に初期化されます。

2.10 PCIe の初期化

PCIe の初期化要件を満たすため、物理層はフラッシュ コンフィグレーション ビットを使って設定されます。その他の設定は、不揮発性メモリに保存されたユーザデータを使って回路初期化中に行われます。

PCIESS をリセットするために PCIe_x_PERST_N サイドバンド リセット入力を使います。

ドライバを使って PCIESS (PCIe SubSystem) をリセットするには、ホットリセット(インバンド リセット)を使います。ホットリセットは、複数の TS1(トレーニング シーケンス 1) パケット(シンボル 5 の bit 0 をアサート)を送信する事により、1 つのリンクから隣のリンクへとインバンドで伝播します。これらの TS1 は全てのレーンで送信されます。TS1 が送信されると、ホットリセットの Tx と Rx は LTSSM (Link Training State Machine) ステートの検出で終了します。ホットリセットは、ルートポート ブリッジ制御コンフィグレーション レジスタ内のセカンダリバスリセットビットのセットにより開始されます。

初期化に要する時間は、設定する必要のあるレーンの数と PCI コントローラの数によって決まります。PCIE0 コントローラが Libero 既定値を使って初期化される場合、PCIE 初期化時間は 440 μ s です。PCIE0 と PCIE1 の両方が有効である場合、PCIE 初期化時間は 782 μ s です。

最悪条件での PCIE 初期化時間は、ステージ 2 アセンブリ ファイル内の全ての PCIe コントローラおよびブリッジレジスタ書き込み時間を含めて計算されます(表 2-10 参照)。

表 2-10. PCIe 初期化時間

PCIE コントローラ の選択	既定値フロー (生成ファイルに対する変更なし)		アセンブリ ファイル変更フロー	
	ステージ 2 アセンブリ ファイル内のレジ スタ書き込み数	PCIe 初期化時間	ステージ 2 アセンブリ ファイル内のレジ スタ書き込み数	PCIe 初期化時間
PCIE0 は有効、 PCIE1 は無効	99	440 μ s	388	937.5 μ s
PCIE0 と PCIE1 の 両方が有効	185	782 μ s	745	1759 μ s

PCIe 初期化プロセスの詳細は、『[PolarFire FPGA and PolarFire SoC FPGA PCI Express User Guide](#)』を参照してください。

2.11 電源投入中のブロックのステート

表 2-11 に、デバイス電源投入時の各種ブロックのステートを示します。

表 2-11. デバイス電源投入時の既定値ステート

ブロック	POR	デバイスブート	回路/メモリ初期化ステート
システムコントローラ	リセット状態に保持	ブートアップシーケンスを実行	回路/メモリ初期化を実行
sNVM	リセット状態に保持	電源投入シーケンス後に動作可能	動作可能
FPGA ファブリック アレイ	電源 OFF	電源投入シーケンス後に動作可能	動作可能
LSRAM	電源 OFF	電源 ON、未初期化	ユーザデータ(設定されている場合)を使って初期化
μSRAM	電源 OFF	電源 ON、未初期化	ユーザデータ(設定されている場合)を使って初期化
μPROM	電源 OFF	電源 ON	動作可能
演算ブロック	電源 OFF	電源 ON	動作可能
トランシーバと TX PLL	電源 OFF	電源 ON、動作不可、終端抵抗はオプションで有効化可能	ユーザデータを使って初期化、動作可能
GPIO/HSIO - 低速(電源 ON 時)	入力バッファは無効、出力バッファは 3 ステート、GPIO バッファはホットプラグモード	電源 ON、動作不可、GPIO バッファはホットプラグモード、HSIO バッファはホットプラグ機能をサポートしない	I/O 電源と I/O 補助電源が電圧しきい値を超えていれば動作可能
GPIO/HSIO - 高速(電源 ON 時)	入力バッファは無効、出力バッファは 3 ステート、GPIO バッファはホットプラグモード	電源 ON、使用不可、GPIO バッファはホットプラグモード、HSIO バッファはホットプラグ機能をサポートしない	I/O 電源と I/O 補助電源が ON の場合、I/O キャリブレーションの完了後に高速動作可能
PCIe	電源 OFF	電源 ON、動作不可	ユーザデータを使って初期化、動作可能
トランシーバ I/O	3 ステート、ホットプラグモード	3 ステート、ホットプラグモード、終端抵抗はオプションで有効化可能	終端抵抗はオプションで有効化可能
MSSIO (PolarFire SoC FPGA のみ)	3 ステート	3 ステート	3 ステート
MSS (PolarFire SoC FPGA のみ)	電源 OFF	電源 OFF	電源 OFF

Note: コールドブートおよびウォームブートでの電源投入から動作可能になるまでの時間については、『[PolarFire FPGA Datasheet](#)』または『[PolarFire SoC FPGA Advance Datasheet](#)』内の「Power-Up to Functional Timing」を参照してください。

3. PolarFire FPGA のリセット

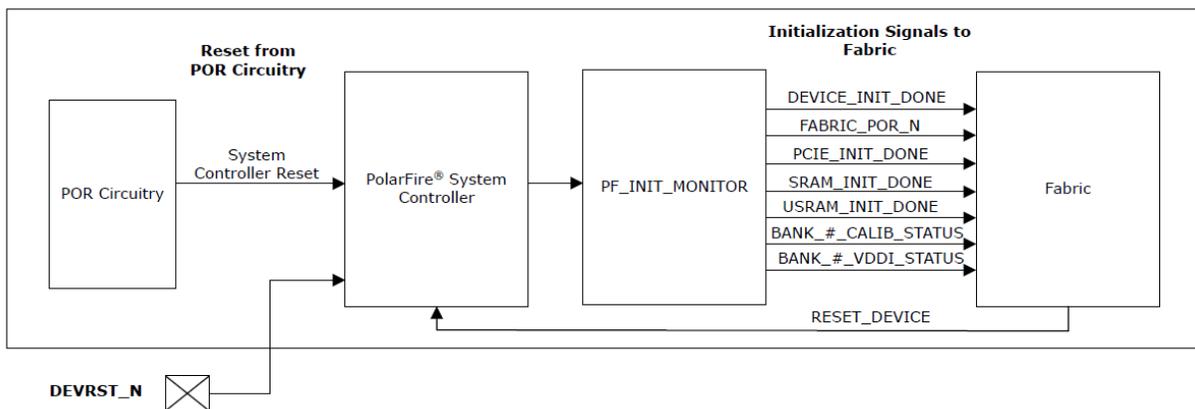
デバイスの電源投入後に、PolarFire FPGA システム コントローラはデバイスの初期化を行います。ファブリック フリップフロップの電源投入時の状態は不確定です。適正動作を得るには、回路にリセットロジックを含める必要があります。フリップフロップの初期状態を既知の値に設定するためにリセットパルスが必要です。以下では、PolarFire ハードリセットとユーザリセットを生成する方法について説明します。

3.1 ハードリセット

PolarFire FPGA は、以下のいずれかのソースによりリセットできます。

- DEVRST_N ピン
- POR 回路
- ファブリック

図 3-1. リセットの概略ブロック図



3.1.1 Device Reset Pad (DEVRST_N)

DEVRST_N (デバイスリセット)は、専用 I/O バンクから給電されます。DEVRST_N のアサートにより、デバイスは完全に再初期化されます。これには PCIe へのユーザ設定データの書き込みとトランシーバ、MSS、ファブリック LSRAM および μ SRAM の再初期化が含まれます。

デバイスの完全なリセットとリブートをスケジューリングするために、DEVRST_N を外部ソースによって Low にする事ができます。これは非同期リセットではありませんが、システム コントローラ内のプロセッサに対してノンマスカブル割り込み(NMI)をアサートした後に、(途中停止できない) デバイスリセットをスケジューリングするためにウォッチドッグ タイマを起動します。このリセットは、ファームウェアが I/O を無効にしてファブリックの電源を安全に遮断した後にアサートされます。

専用の DEVRST_N ピンまたは任意の GPIO/HSIO を使った汎用リセット信号をグローバルなシステムリセット用に使う事で、信頼性の高いシステムを設計できます。以下の場合、DEVRST_N をデバイスのウォームリセット用に使えます。

- 動作中にユーザ回路が自動初期化されたファブリック RAM または PCIe 設定を変更する場合
- ユーザ回路がトランシーバまたは UserCrypto を使う場合

PCIe の場合、PCIESS をリセットするために PCIe_x_PERST_N 入力を使います。PCIe_x_PERST_N はサイドバンド リセット入力です。ドライバを使って PCIESS をリセットするには、ホットリセット(インバンド リセット)を使います。

ホットリセットは、複数の TS1(トレーニング シーケンス 1)パケット(シンボル 5 の bit 0 をアサート)を送信する事により、1 つのリンクから隣のリンクへとインバンドで伝播します。これらの TS1 は全てのレーンで送信されます。TS1 が送信されると、ホットリセットの Tx と Rx は LTSSM (link training state machine)ステートの検出で終了します。ホットリセットは、ルートポートブリッジ制御コンフィグレーション レジスタ内のセカンダリバス リセットビットをセットする事によりソフトウェアによって開始されます。

上記を除く全ての状況では、任意の GPIO/HSIO I/O を介する汎用リセット信号を使う事を推奨します(汎用リセット信号を使う事で、回路がリセットを完了するまでの時間を大幅に短縮できるため)。

専用 DEVRST_N ピンをウォームリセット用に使わない場合、DEVRST_N ピンは以下のいずれかの方法で設定する必要があります。

- システム/クロックが安定してデバイスが正常に起動するまで、外付けの POR デバイスまたは外部デバイスによって DEVRST_N をアサート状態に駆動する。
- 1 kΩ 抵抗を介して DEVRST_N を VDDI3 に接続する(この抵抗を他のピンと共有してはいけません)
 - この場合、ユーザ回路がパワーオン リセットから解除される前にデバイスに供給される全てのクロックが安定する事をユーザが保証する必要があります。電源投入後に FPGA ファブリック内の回路がアクティブになるまでに要する最小時間は、『PolarFire SoC FPGA Advance Datasheet』内の「Power-Up To Functional Timing」を参照してください。

3.1.2 POR 回路からのリセット

POR 回路は、全ての電源電圧(VDD、VDD18、VDD25)がそれらの最低しきい値レベル以上で安定した時にシステムコントローラをリセット状態から解除します。いずれかの電源電圧が最低しきい値レベルを下回ると、デバイスリセットが発行されます。

3.1.3 ファブリックからのデバイスリセット

完全なデバイスリセットおよびリブートを開始するために、ユーザロジックはファブリックからの RESET_DEVICE 信号で High パルスを生成する場合があります。これは、タンパー レスポンス制御信号の1つです。この信号は、ファブリック内で検出されたタンパーイベントに対するタンパー レスポンスとして使えます。ファブリックからの RESET_DEVICE リセット信号がアサートされた時に、システム コントローラは以下のシーケンスでデバイスの電源を遮断します。

1. リセット信号はノンマスカブル割り込みとしてシステム コントローラへ伝播し、最初に全ての I/O を無効にします。
2. デバイスリセットをスケジューリングするためにウォッチドッグ タイマが起動します。
3. ファブリックの電源を遮断します。

リセットは全ての周辺モジュール(MSS (PolarFire SoC のみ)、ファブリック、トランシーバ、PCIe、PLL、DLL 等)に対して発行されます。

3.2 ユーザリセットの生成手順

ユーザリセットの生成手順は以下の通りです。

- 外部リセット入力を使う場合、リセット信号の入力バッファが動作可能であると検出されるまでリセット入力を無視する必要があります。以下の両方の条件が成立する場合に入力バッファは動作可能です。
 - FABRIC_POR_N がネゲートされる
 - BANK_x_VDDI_STATUS がアサートされる(x は入力バッファを格納した I/O バンクの番号)
- 外部参照クロック入力を備えた PLL を使う場合、外部参照クロックが安定し、かつ外部参照クロックの入力バッファが動作可能であると検出されるまで、FPGA ファブリックは PLL を電源遮断状態に保持する必要があります。入力バッファは、以下の2つの条件の両方が成立した時に動作可能となります。
 - FABRIC_POR_N がネゲートされる
 - BANK_y_VDDI_STATUS がアサートされる(y は入力バッファを格納した I/O バンクの番号)
- DRI_CLK が FPGA ファブリック内のフリップフロップ(例: クロック分周回路)によって生成される場合、このフリップフロップは非同期に(例: FABRIC_POR_N によって)リセットされる必要があります。
- FPGA ファブリック内で DRI_PSEL を駆動するフリップフロップは、非同期に(例: FABRIC_POR_N によって)リセットされる必要があります。
- PLL がロック信号のアサートまでクロックパルスを出力しないよう設定されている場合、PLL ロック信号を直接リセット信号として使ってはいけません。この場合、リセット中にクロックエッジが発生しないため、FPGA ファブリック内の全ての同期リセットロジックは正しくリセットされません。

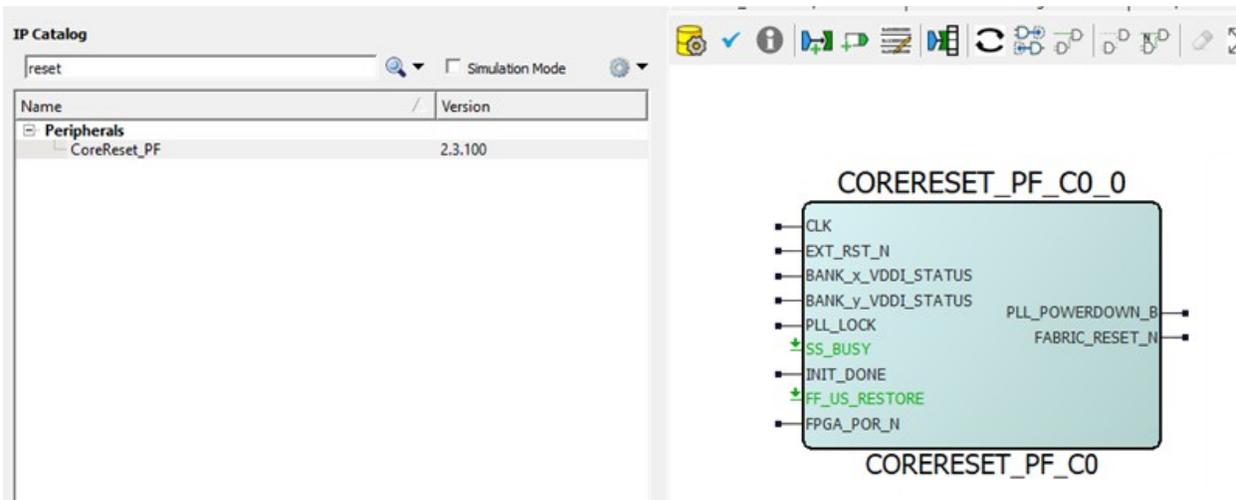
これらの要件を満たすため、[図 3-3](#)の通りに CORERESET_PF を使う事を推奨します。

CORERESET_PF IP コアは Libero IP カタログに含まれています([図 3-2](#) 参照)。この IP コアは、ユーザが指定したクロックドメイン内のダウンストリーム ロジックに対するリセットをクロックに同期してディアサートします。従って、リセットのアサートは非同期ですが、ネゲートはクロックに同期します。この IP コアにより回復時間要件が満たされ、同一クロックパルスに同期して全てのフリップフロップがリセットから解除されます。

CORERESET_PF IP は、複数ソース(例: 外部 GPIO、PLL ロック、PF_INIT_MONITOR ブロック等)からのリセットを統合します。CORERESET_PF IP は、ファブリック ロジック向けにシステムレベルの同期リセット(FABRIC_RESET_N)を生成します。電源投入時のファブリック フリップフロップの状態は不確定です。

フリップフロップの初期状態を既知の値に設定するためにリセットパルスが必要です。このリセットには、FABRIC_RESET_N を使う事を推奨します。

図 3-2. CORERESET_PF IP

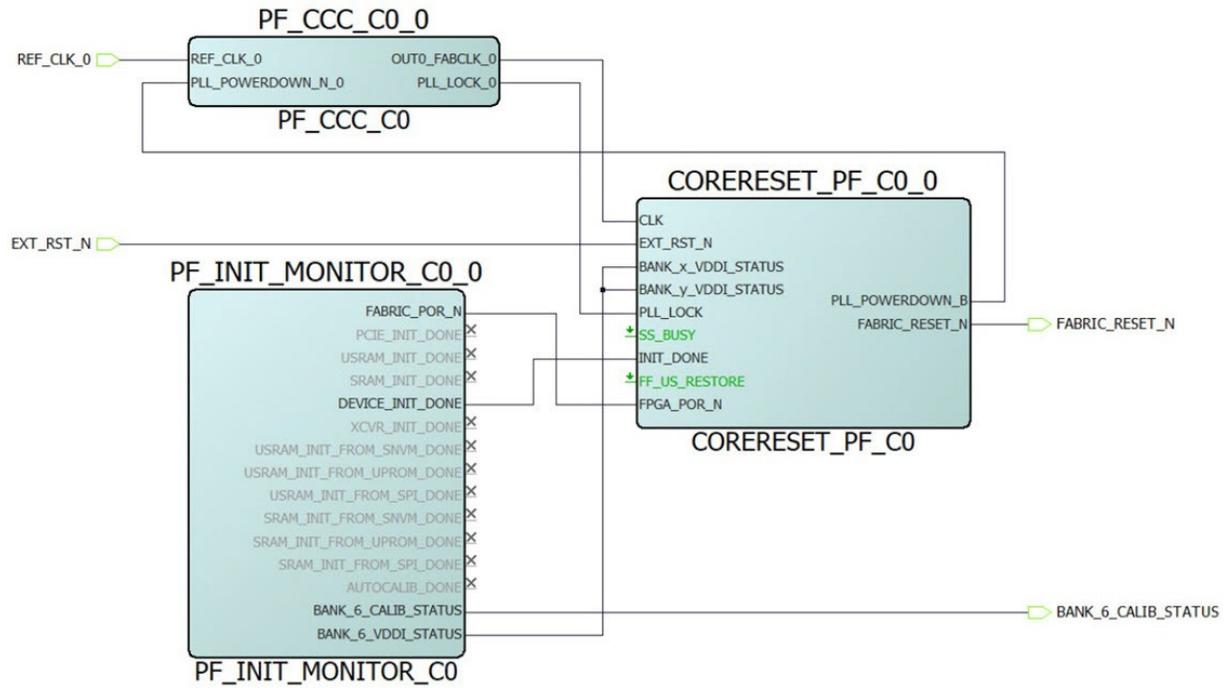


Note: CORERESET_PF IP の詳細は、Libero カタログ内の CORERESET_PF ハンドブックを参照してください。

PolarFire 初期化モニタ(PF_INIT_MONITOR コンポーネント)を全ての回路でインスタンス化する必要があります。インスタンス化したモニタはユーザロジックをリセットするために使えます。図 3-3 に、PF_INIT_MONITOR の使用例を示します。この例では、ユーザロジックに対して同期リセット信号を提供するために、DEVICE_INIT_DONE 信号を RESET_GEN_0 ブロック (CORERESET_PF IP) の INIT_DONE 信号に接続しています。DEVICE_INIT_DONE 信号は、デバイス初期化の完了後にアサートされます。

この例では、EXT_RST_N と REF_CLK をバンク 6 に接続しています。また、PF_INIT MONITOR IP で Bank_6_VDDI_STATUS を有効にする事により、BANK_x_VDDI_STATUS と BANK_y_VDDI_STATUS を Bank_6_VDDI_STATUS に接続しています。Bank_6_CALIB_STATUS は、バンク 6 に接続した GPIO のキャリブレーションステータスを監視するために使えます。

図 3-3. PolarFire 初期化の例



4. PolarFire SoC FPGA のリセット

デバイスの電源投入後に、PolarFire SoC FPGA システム コントローラはデバイスの初期化を行います。以下では、PolarFire SoC のリセットについて説明します。

4.1 ハードリセット

PolarFire SoC FPGA のハードリセット アーキテクチャは PolarFire FPGA と同じです。ハードリセットの詳細は、[3.1. 「ハードリセット」](#) を参照してください。

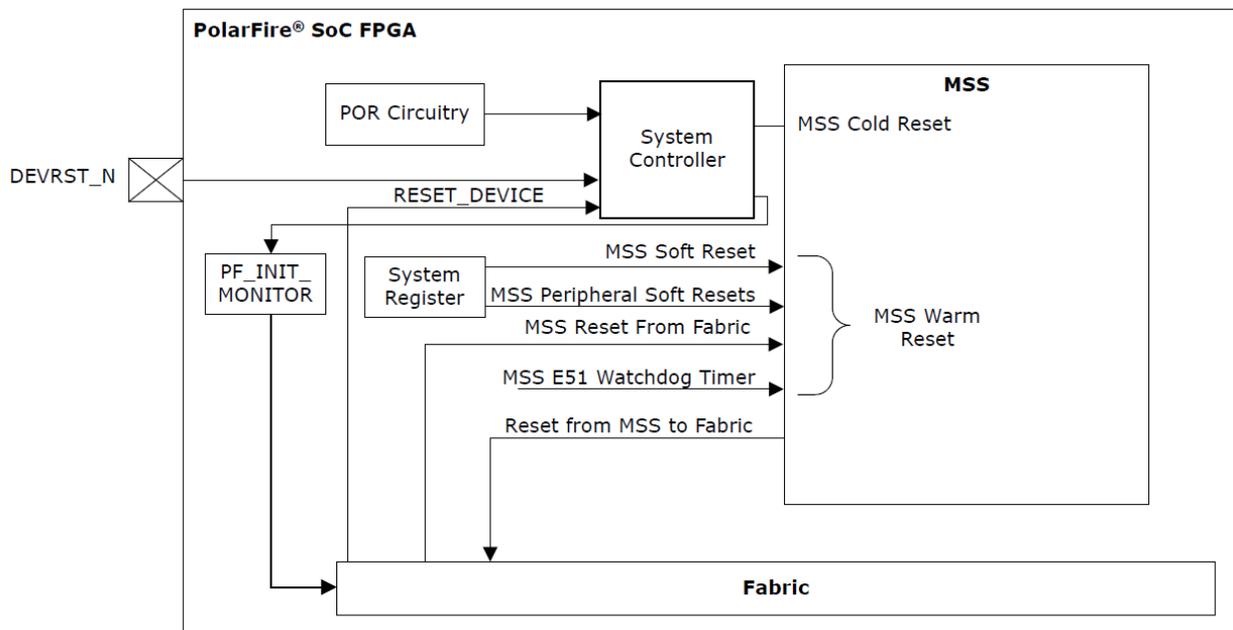
4.2 MSS リセット

デバイスの電源投入後に、PolarFire SoC システム コントローラはデバイスの初期化を行います。MSS ウォームリセット後に、ファームウェアは MSS をリセットから解除します。PolarFire SoC のリセットには以下が含まれます。

- MSS コールドリセット
- MSS ウォームリセット
- MSS 周辺モジュール ソフトリセット
- ユーザリセット

MSS の詳細は、『[PolarFire SoC FPGA MSS Technical Reference Manual](#)』を参照してください。

図 4-1. MSS リセットのブロック図



4.2.1 MSS コールドリセット

MSS コールドリセットは、デバイスの電源投入時に POR (パワーオン リセット) 回路により開始されます。MSS コールドリセットは、eNVM を除く MSS 内の全ての機能をリセットします。eNVM は SOFT_RESET_CR レジスタを使ってリセットできます。

4.2.2 MSS ウォームリセット

4 種類の MSS ウォームリセットにより、MSS 全体と全ての周辺モジュールをリセットできます。ウォームリセットにより、MSS 内の全ての機能(MSSIO コンフィグレーションと IOMUXe は除く)が非同期にリセットされます。MSS GPIO 周辺モジュールがファブリックによってリセットされるよう設定されている場合、MSS GPIO 周辺モジュールはリセットされません。MSS は、ウォームリセットの原因が取り除かれるまで内部でリセット状態を維持します。

ウォームリセット信号が解除されると、MSS ウォームリセット解除イベントが発生した事を示すために、システムコントローラに対して割り込みが生成されます。この後に、システム コントローラ ファームウェアは MSS をリセットから解除します。

以下では、各種ソースによる MSS ウォームリセットについて説明します。

4.2.2.1 MSS ソフトリセット

アプリケーション コードから MSS_RESET_CR ソフトリセット レジスタに特別な値を書き込む事により、MSS を完全にリセットできます。表 4-1 に、MSS ソフトリセット レジスタの概要を示します。

表 4-1. MSS_RESET_CR

	レジスタ名	タイプ	既定値	フィールドの概要
31:16	予約済み	RO	0x0000	予約済み
15:0	RESET_VALUE	RW	0x0000	16'hDEAD が書き込まれると、完全な MSS リセットが発生します。リセット時にこのレジスタはクリアされます。このレジスタには任意の値を書き込めますが、16'hDEAD を書き込んだ場合にのみリセットが発生します。

4.2.2.2 ファブリックからの MSS リセット

ファブリック内のユーザロジックがリセット信号 MSS_RESET_N_F2M をアサートすると、MSS は非同期にリセットします。

4.2.2.3 MSS E51 プロセッサ ウォッチドッグ タイムアウト リセット

E51 プロセッサのウォッチドッグ タイマがタイムアウトした時に MSS がリセットされます。

MSS リセットの原因

MSS は前述の通りに各種の方法でリセットできます。ユーザは 32 ビット RESET_SR レジスタにアクセスする事で、MSS リセットをトリガした原因を特定できます。表 4-2 に MSS リセットの原因を示します。

表 4-2. MSS リセットの原因

原因	リセット原因ビット	アサート元	注釈
SCB_PERIPH_RESET	0	SCB	これはパワーオンリセットです。これは MSS を完全にリセットします(eNVM トリム値を含む)。SOFT-RESET レジスタ内の追加ビットを使って SCB レジスタをリセットする事もできます。
SCB_MSS_RESET	1	SCB、CPU、MSS	これはコア コンプレックス、周辺モジュール、全ての AXI インフラストラクチャを含む MSS をリセットします。eNVM トリム値と SCB レジスタはリセットしません。
SCB_CPU_RESET	2	SCB、CPU、MSS	これはコア コンプレックスのみリセットします。MSS は未処理の AXI トランザクション等のクリアと同時にリセットを要求するため、通常はこのリセットを使ってはいけません。
DEBUGGER_RESET	3	デバッガ	これはコア コンプレックス デバッガによってアサートされ、SCB_MSS_RESET と同じ効果を有します。
FABRIC_RESET	4	ファブリック	これはファブリックによってアサートされ、SCB_MSS_RESET と同じ効果を有します。このリセットは、システムレジスタ ビットによってリセット時に無効にされ、再度有効にしない限り機能しません。

.....続き			
原因	リセット原因ビット	アサート元	注釈
WDOG_RESET	5	ウォッチドッグ	これはウォッチドッグ リセットがアクティブになった事を示します。
GPIO_RESET	6	ファブリック	これはファブリック GPIO リセットがアサートされた事を示します。これは GPIO ブロックをリセットします(GPIOがこの信号によってリセットされるよう設定されている場合)。
SCB_BUS_RESET	7	ファブリック	SCB バスリセットが発生した事を示します。
CPU_SOFT_RESET	8	CPU	CPU がソフトリセット レジスタを使って MSS をリセットした事を示します。
予約済み	31:9		予約済み

4.2.2.4 MSS 周辺モジュール ソフトリセット

各 MSS 周辺モジュールには、MSS システムレジスタ内のソフトリセット レジスタ(SOFT_RESET_CR) のビットが割り当てられています。MSS 周辺モジュールを使うには、そのモジュールに対応するビットに「1」を書き込んだ後に「0」を書き込む必要があります。MSS がリセットすると、これらの全てのリセットがアサートされます。

表 4-3. MSS 周辺モジュール ソフトリセット

アドレス	レジスタ	フィールド	ビット	タイプ	リセット値
x88	SOFT_RESET_CR	ENVM	0	RW	0x0

MSS 周辺モジュール ソフトリセットには以下の例外が存在します。

MSS GPIO ソフトリセット: デバイスがプログラミング済みである場合、3つの各 MSS GPIO ブロックはファブリックからの MSS ウォームリセットまたは MSS GPIO リセット信号によってリセットされるよう設定できます。

MSS ウォームリセット(既定値)を使うよう設定した場合、それらは MSS システムレジスタ内の MSS GPIO ソフトリセット レジスタによってもリセットされます。

GPIO ファブリック リセットを使うよう設定した場合、MSS GPIO レジスタの状態は MSS GPIO ソフトリセット レジスタへの書き込みによる影響を受けません。しかし、これらの MSS GPIO レジスタは、システム コントローラ ファームウェアによって MSS ウォームリセット イベントの処理中にリセットされます。

4.2.3 MSS eNVM リセット

eNVM のリセットは、システム コントローラによって処理されます。

4.2.4 MSS からファブリックへのリセット ステータス信号

MSS のリセット ステータスを示すため、MSS からファブリックへステータス信号 MSS_RESET_N_M2F が提供されます。ファブリック ロジックは、この信号を使って MSS とファブリックの間のデータ転送を中止できます。MSS_RESET_N_M2F は、MSS_RESET_N_F2M がアサートされた時にアサートされ、MSS ユーザ ソフトウェアによりディアサートされます。

4.3 ユーザリセットの生成手順

ユーザリセットの生成手順は以下の通りです。

- 外部リセット入力を使う場合、リセット信号の入力バッファが動作可能であると検出されるまでリセット入力を無視する必要があります。以下の両方の条件が成立する場合に入力バッファは動作可能です。
 - FABRIC_POR_N がネゲートされる
 - BANK_x_VDDI_STATUS がアサートされる(x は入力バッファを格納した I/O バンクの番号)
- 外部参照クロック入力を備えた PLL を使う場合、外部参照クロックの入力バッファが動作可能になるまで、FPGA ファブリックは PLL を電源遮断状態に保持する必要があります。入力バッファは、以下の 2つの条件の両方が成立した時に動作可能となります。

- FABRIC_POR_N がネゲートされる
- BANK_y_VDDI_STATUS がアサートされる(y は入力バッファを格納した I/O バンクの番号)
- DRI_CLK が FPGA ファブリック内のフリップフロップ(例: クロック分周回路)によって生成される場合、このフリップフロップは非同期に(例: FABRIC_POR_N によって)リセットされる必要があります。
- FPGA ファブリック内で DRI_PSEL を駆動するフリップフロップは、非同期に(例: FABRIC_POR_N によって)リセットされる必要があります。
- PLL がロック信号のアサートまでクロックパルスを出力しないよう設定されている場合、PLL ロック信号を直接リセット信号として使ってはいけません。この場合、リセット中にクロックエッジが発生しないため、FPGA ファブリック内の全ての同期リセットロジックは正しくリセットされません。

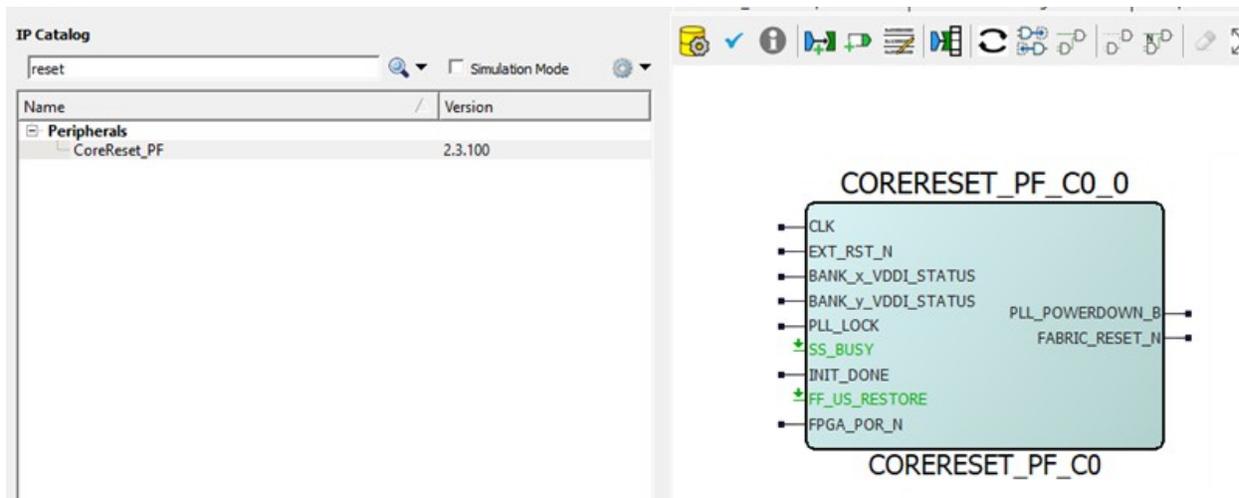
これらの要件を満たすため、[図 4-3](#) の通りに CORERESET_PF を使う事を推奨します。

CORERESET_PF IP コアは Libero IP カタログに含まれています([図 4-2](#) 参照)。この IP コアは、ユーザが指定したクロックドメイン内のダウストリーム ロジックに対するリセットをクロックに同期してディアサートします。従って、リセットのアサートは非同期ですが、ネゲートはクロックに同期します。この IP コアにより回復時間要件が満たされ、同一クロックパルスに同期して全てのフリップフロップがリセットから解除されます。

CORERESET_PF IP は、複数ソース(例: 外部 GPIO、PLL ロック、PF_INIT_MONITOR ブロック等)からのリセットを統合します。CORERESET_PF IP は、MSS とファブリック ロジック向けにシステムレベルの同期リセット(FABRIC_RESET_N)を生成します。

電源投入時のファブリック フリップフロップの状態は不確定です。フリップフロップの初期状態を既知の値に設定するためにリセットパルスが必要です。このリセットには、FABRIC_RESET_N を使う事を推奨します。

図 4-2. CORERESET_PF IP

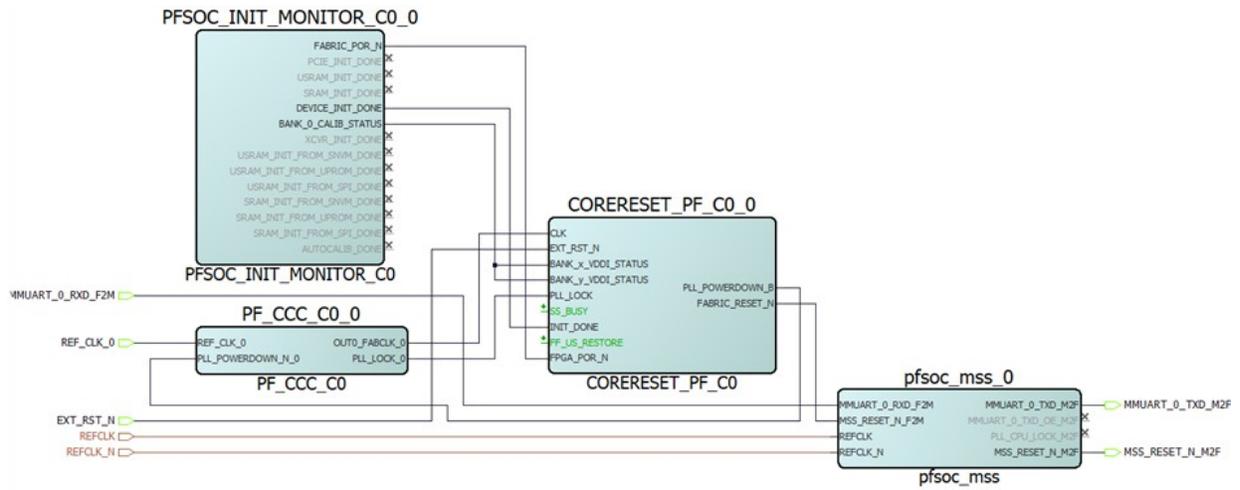


Note: CORERESET_PF IP の詳細は、Libero カタログ内の CORERESET_PF ハンドブックを参照してください。

PolarFire SoC 初期化モニタ(PFSOC_INIT_MONITOR コンポーネント)を全ての回路でインスタンス化する必要があります。インスタンス化したモニタはユーザロジックをリセットするために使えます。図 4-3 に、PFSOC_INIT_MONITOR と PFSOC_MSS_C0_0 の使用例を示します。この例では、ユーザロジックに対して同期リセット信号を提供するために、DEVICE_INIT_DONE 信号を RESET_GEN_0 ブロック(CORERESET_PF IP)の INIT_DONE 信号に接続しています。

この例では、EXT_RST_N と REF_CLK をバンク 0 に接続しています。また、PFSOC_INIT MONITOR IP 内で Bank_0_VDDI_STATUS を有効にする事により、BANK_x_VDDI_STATUS と BANK_y_VDDI_STATUS を Bank_0_VDDI_STATUS に接続しています。Bank0_CALIB_STATUS は、バンク 0 に接続された GPIO のキャリブレーション ステータスを監視するために使えます。CORERESET_PF_C0 IP の FABRIC_RESET_N 信号は PFSOC_MSS_C0_0 の MSS_RESET_N_F2M に接続しています。MSS_RESET_N_M2F 出力は、MSS FIC を介して接続されたファブリック周辺モジュール等、MSS に関連するファブリック ロジック サブシステムに接続されて、MSS とファブリック サブシステムの間で確実にリセット同期が行われるようになります。DEVICE_INIT_DONE 信号は、回路初期化の完了後にアサートされます。

図 4-3. PolarFire SoC 初期化の例



5. 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 ファミリの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を防ぐために、このモードを使ってデバイスを保護する事が不可欠です。システム コントローラ サスペンド モードを使う場合、システム コントローラ出力ラッチ オプションを有効にして必要な PF_INIT_MONITOR または PFSOC_INIT_MONITOR IP をインスタンス化および設定する必要があります。この場合、CLK_160_MHZ ポートを内部 160 MHz RCOSC に接続する必要があります。

以下では、システム コントローラ サスペンド モードを有効にした場合の、デバイスの挙動への影響について説明します。

5.1 デバイスのプログラミングとシステムサービス

システム コントローラ サスペンド モードを有効にしてデバイスをプログラミングした場合、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ます。その他のオプションは、JTAG_TRST_B ピンの制御により有効または無効にできます。システム コントローラ サービスモードで利用可能なデバイス機能とこのモードの動作については、『[PolarFire FPGA and PolarFire SoC FPGA System Services User Guide](#)』を参照してください。

Note: PolarFire および RT PolarFire デバイスでシステム コントローラ サスペンドモード機能を使う場合、JTAG_TRST_B ピンが論理 High にアサートされると PF_INIT_MONITOR マクロの全ての出力は 0 に設定されます。この状況は、ユーザが SmartDebug を使ってデバイスの再プログラミングまたはデバッグを試みた場合に発生します。PF_INIT_MONITOR 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。このような状況は PolarFire SoC デバイスでは発生しません。

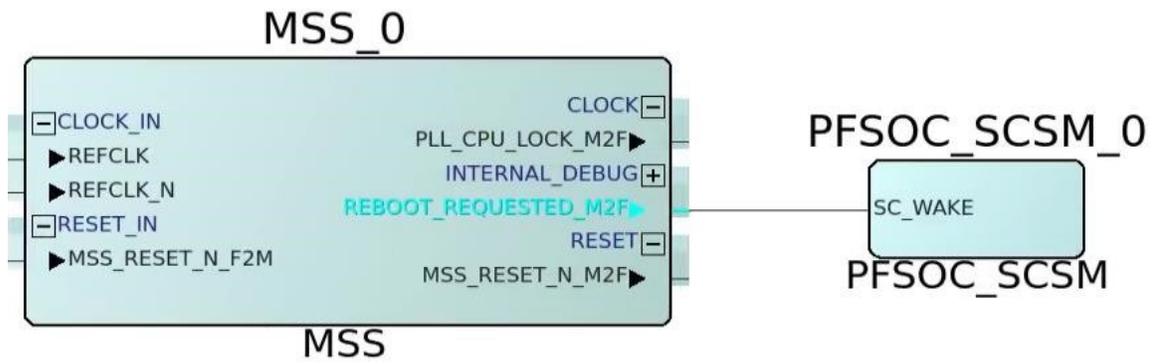
5.2 PolarFire SoC のリブート

システム コントローラは、MSS のブートで重要な役割を果たします。システム コントローラ サスペンドモードが有効である場合、デバイスの電源投入またはリセット時にシステム コントローラは MSS をブートした後にシステム コントローラ サスペンドモードへ移行します。システム コントローラ サスペンドモード中はシステム コントローラがリセット状態となり、通常ではサポートするサービス(MSS ブートを含む)をサポートできなくなります。動作中にリブートが必要になった場合、MSS はシステム コントローラにサスペンドモードを終了させる必要があります。これを行うには PFSOC_SCSM マクロをインスタンス化し、その入力を MSS REBOOT_REQUESTED_M2F 出力に接続する必要があります(図 5-1 参照)。MSS REBOOT_REQUESTED_M2F ポートは、MSS コンフィグレータ内の[Expose Feedback ports to Fabric]チェックボックスで有効にします。このポートに対するその他の接続はサポートされません。ユーザの FPGA ファブリック回路にこの接続を追加した場合、MSS REBOOT_REQUESTED_M2F 出力がアサートされるとシステム コントローラはサスペンドモードを終了し、保留中の MSS リブート要求を処理します。MSS のブート後に REBOOT_REQUESTED_M2F 出力はディアサートし、システム コントローラはサスペンドモードに戻ります。システム コントローラのステータスは、SC_STATUS マクロにより監視できます。

Note: この PFSOC_SCSM マクロは PolarFire SoC デバイスファミリのみサポートします。

Note: 量産用デバイスのみサポートします。PolarFire SoC の ES (Engineering Sample)デバイスはサポートしません。

図 5-1. PolarFire SoC のリブート



6. 補遺: 電源

表 6-1 に電源の一覧を示します。

表 6-1. PolarFire SoC 内の電源

電源	概要
VDD	ファブリックコアとトランシーバ/PCIe ブロック向けの電源
VDD18	ファブリック プログラミングと RC コントローラ向けの電源
VDD25	コーナー位相ロックループ(PLL)と内部不揮発性メモリ(sNVM)向けの電源
VDDIx	I/O バンク向けの電源
VDDAUXx	GPIO/HSIO バンク向けの電源
VDDA	トランシーバ向けの電源
VDDA25	トランシーバ PLL 向けの電源
VDD_XCVR_CLK	トランシーバ参照クロック入力バッファ向けの電源

7. 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。

リビジョン	日付	概要
C	2022年4月	<p>このリビジョンでの主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> システムコントローラ サスペンドモードが有効である時のタンパーフラグ出力のラッチに関する情報を追加しました (2.3. 「回路/メモリの初期化」参照)。 システムコントローラ サスペンドモードが有効である時の PF_INIT_MONITOR の動作に関する情報を追加しました (2.3.1. 「PolarFire 初期化モニタ」と 2.3.2. 「PolarFire SoC 初期化モニタ」参照)。 CORERESET_PF IP に関する情報を追加しました (3. 「PolarFire FPGA のリセット」と 4. 「PolarFire SoC FPGA のリセット」参照)。機能に変更はありません。 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 (5. 「システムコントローラ サスペンドモード」参照)。 RESET_DEVICE 信号のリセットのブロック図を更新しました (図 3-1 と図 4-1 参照)。
B	2021年8月	<p>以下の文書を本書に統合しました。</p> <ul style="list-style-type: none"> UG0725: PolarFire FPGA Device Power-up and Reset User Guide PolarFire SoC FPGA Power-up and Reset User Guide <p>これらの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。詳細は表 7-1 と表 7-2 を参照してください。</p>
A	2021年3月	<p>このリビジョンでの主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> トランシーバの初期化時間に関する情報を追加しました (「トランシーバの初期化」参照)。 PCIe の初期化時間に関する情報を追加しました (「PCIe の初期化」参照)。 I/O の再キャリブレーションに関する情報を追加しました (「I/O の再キャリブレーション」参照)。 本書の書式を Microchip 社のテンプレートに適合させました。文書番号を 50200890 から DS60001676 へ変更しました。

表 7-1 に、『UG0725: PolarFire FPGA Device Power-up and Reset User Guide』に適用された変更の履歴を示します。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。

Note: 『UG0725: PolarFire FPGA Device Power-up and Reset User Guide』は廃刊となり、その内容は『PolarFire® FPGA and PolarFire SoC FPGA Device Power-up and Reset User Guide』(本書)に引き継がれました。

表 7-1. 『UG0725: PolarFire FPGA Device Power-up and Reset User Guide』の改訂履歴

リビジョン	日付	概要
リビジョン 8.0	7/21	<p>このリビジョンでの主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> SUSPEND_EN に PUFT (Power-up Function Timing) タイミングパラメータに関する情報を追加しました (「回路/メモリの初期化」参照)。 「回路/メモリ初期化の設定方法」の内容を更新しました。

.....続き		
リビジョン	日付	概要
リビジョン 7.0	2/21	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none"> I/O の再キャリブレーションに関する情報を追加しました(「I/O の再キャリブレーション」参照)。 トランシーバの初期化時間に関する情報を追加しました(「トランシーバの初期化」参照)。
リビジョン 6.0	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。 <ul style="list-style-type: none"> 「回路/メモリ初期化の設定方法」に関する情報を更新しました。 ランプアップ時間に関する情報を追加しました(「HSIO/GPIOバンクの初期化」参照)。 低速 I/O のキャリブレーションに関する情報を追加しました(「HSIO/GPIOバンクの初期化」参照)。
リビジョン 5.0	3/19	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none"> Libero SoC v12.0 向けの文書を更新しました。 ファブリック RAM の初期化手順を更新しました(「回路/メモリ初期化の設定方法」参照)。
リビジョン 4.0	4/18	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none"> Libero SoC PolarFire v2.1 向けの文書を更新しました。 「デバイスのブート」を更新しました。 「回路/メモリの初期化」を更新しました。 「回路/メモリ初期化の設定方法」を更新しました。 「HSIO/GPIOバンクの初期化」を更新しました。 「トランシーバの初期化」を更新しました。 「電源投入中のブロックのステート」を更新しました。
リビジョン 3.0	4/18	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none"> 文書内のスクリーンショットを Libero SoC PolarFire v2.0 リリース向けに更新しました。 「回路/メモリの初期化」から SPI スレーブ プログラミング モードに関する注釈を削除し、PolarFire 初期化モニタの使い方に関する情報を追加しました。 μPROM と外部 SPI フラッシュに関する項目を追加しました。 BANK_#_CALIB_STATUS および BANK_#_VDDI_STATUS 信号を説明するため、「HSIO/GPIOバンクの初期化」を編集しました。 「Power-Up to Functional Time (PUFT)」に、PolarFire FPGA データシートへの参照を示す Note を追加しました。 「電源投入」から図「Top-Level Device Power-Up」を削除しました。 「リセットの概略ブロック図」で GPIO_ACTIVE および HSIO_ACTIVE ピンを削除し、BANK_#_CALIB_STATUS および BANK_#_VDDI_STATUS ピンを追加しました。 DEVIRST_N にデバイスリセットの使用に関する推奨を追加しました。 図「Power-up To Functional Time (PUFT)」を更新しました。
リビジョン 2.0	11/17	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none"> Libero SoC PolarFire v1.1 SP1 に導入された機能と拡張を反映するため本書を更新しました。 PolarFire 初期化モニタに関する情報を追加しました。詳細は、「ユーザーリセットの生成手順」を参照してください。
リビジョン 1.0	2/17	『UG0725: PolarFire FPGA Device Power-up and Reset User Guide』の初版です。

『PolarFire SoC FPGA Device Power-up and Reset User Guide』に適用された変更の履歴を表 7-2 に示します。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。

Note: 『PolarFire SoC FPGA Device Power-up and Reset User Guide』は廃刊となり、その内容は『PolarFire® FPGA and PolarFire SoC FPGA Device Power-up and Reset User Guide』(本書)に引き継がれました。

表 7-2. 『PolarFire SoC FPGA Device Power-up and Reset User Guide』の改訂履歴

リビジョン	日付	概要
3.0	10/2020	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none">ランプアップ時間に関する情報を追加しました (2.6. 「HSIO/GPIOバンクの初期化」参照)。低速 I/O のキャリブレーションに関する情報を追加しました (2.6. 「HSIO/GPIOバンクの初期化」参照)。2.4. 「MSS ブリブート(PolarFire SoC FPGA のみ)」と 2.5. 「MSS ユーザブート(PolarFire SoC FPGA のみ)」を追加しました。
2.0	04/2020	このリビジョンでの主な変更内容は以下の通りです。 <ul style="list-style-type: none">2.3. 「回路/メモリの初期化」を更新しました。2.3.7. 「回路/メモリ初期化の設定方法」に関する情報を追加しました。4.3. 「ユーザーリセットの生成手順」に関する情報を追加しました。
1.0	—	本書の初版です。

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

製品変更通知サービス

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。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、経費のいかににかかわらず、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 ロゴ、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、AnyIn、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 ロゴ、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。

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。

Adapttec 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: 978-1-6683-0247-7

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

ボストン

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 - ハイムブロン

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 - ドリュエーン

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 - ウォーキングム

Tel:44-118-921-5800
Fax:44-118-921-5820