

PolarFire® FPGA PCIe®ルートポート アプリケーションノート

AN4664



はじめに (質問する)

Microchip PolarFire® FPGA は、トランシーバの物理層インターフェイス(PHY)を使う最適化された組み込みコントローラ ブロックと完全に統合された PCIe エンドポイントおよびルートポート サブシステムをサポートします。各 PolarFire デバイスは 2 つの組み込み PCIe®サブシステム(PCIESS)ブロックを備えています。これらのブロックは、Libero® SoC ソフトウェア内で PF_PCIE IP コンフィグレータを使って個別にまたはペアとして設定することができます。

PF_PCIE コアは PCI Express Base 仕様リビジョン 3.0 および Gen1/2 と互換です。このコアは、PCIe 空間にに対するメモリマップド AMBA® (Advanced Microcontroller Bus Architecture) AXI4 (Advanced eXtensible Interface 4) アクセスと、メモリマップド AXI4 空間にに対する PCIe アクセスを実装します。詳細は [『PolarFire Family PCI Express User Guide』](#) を参照してください。

目次

はじめに.....	1
1. PolarFire FPGA PCIe ルートポート.....	3
1.1. デモデザインの要件.....	3
1.2. 前提条件.....	4
1.3. デモデザイン.....	4
1.4. クロック構造.....	9
1.5. リセット構造.....	10
2. Libero デザインフロー	11
2.1. 論理合成.....	11
2.2. 配置配線.....	12
2.3. タイミングの検証.....	13
2.4. FPGA 配列データ.....	13
2.5. デモデザインの初期化データとメモリの設定.....	13
2.6. ビットストリームの生成.....	16
2.7. デバイスのプログラミング	16
3. デモのセットアップ	18
3.1. 2つのボードの接続.....	18
4. デモの実行	20
4.1. GUI のインストール.....	20
4.2. エニュメレーションデータの表示.....	20
4.3. コントロール プレーン コマンドの実行	22
4.4. データプレーン コマンドの実行	28
4.5. PolarFire DMA スループットのまとめ	32
5. 補遺 1: FlashPro Express を使ってデバイスにプログラミングする方法.....	33
6. 補遺 2: DDR4 の設定.....	35
7. 補遺 3: TCL スクリプトの実行	38
8. 補遺 4: 参考資料.....	39
9. 改訂履歴.....	40
Microchip 社の FPGA サポート	42
Microchip 社の情報.....	42
商標.....	42
法律上の注意点.....	42
Microchip 社のデバイスコード保護機能	43

1. PolarFire FPGA PCIe ルートポート (質問する)

本書は、Mi-V ソフトプロセッサを使った PolarFire FPGA PCIe コントローラのルートポート機能について説明します。本書では、PCIe ルートポート デモ GUI アプリケーションを使って、エンドポイント デバイスのエニュメレーションや低速および高速データ転送といった PCIe ルートポート機能のデモを実行します。

本デモデザインは、PCIe コントロール プレーンおよびデータ プレーン機能を初期化/開始する MiV ソフトプロセッサを備えています。PCIe ルートポート回路の実装、必要ブロック、Libero SoC 内でインスタンス化される IP コアの詳細は [「1.3. デモデザイン」](#) を参照してください。

本デモデザインは下記の方法でプログラミングできます。

- **.job ファイルを使う場合:** デモ関連ファイルと一緒に提供される .job ファイルを使ってデバイスをプログラミングする方法は [「3. デモのセットアップ」](#) を参照してください。
- **Libero SoC を使う場合:** Libero SoC を使ってデバイスをプログラミングする方法は [「2. Libero デザインフロー」](#) を参照してください。デモデザインを変更する場合はこちらの方法を使います。

デモを実行するには下記の手順が必要です。

- ルートポート用に使う PolarFire 評価用ボードにルートポート デモデザインをプログラミングする
- エンドポイント用に使うもう 1 つの PolarFire 評価用ボードにエンドポイント デモデザインをプログラミングする
- PCIe アダプタカードを使って両方の評価用ボードを接続する

PCIe ルートポート デモのセットアップの詳細は [「3. デモのセットアップ」](#) を参照してください。

1.1. デモデザインの要件 (質問する)

表 1-1 に、本デモデザインのハードウェアおよびソフトウェア要件を示します。

表 1-1. デモデザインの要件

要件	バージョン
オペレーティング システム	64 ビット Windows® 10 および 11
ハードウェア	
2x PolarFire® 評価用キット (MPF300TS-FCG1152I)	リビジョン D 以上
Microchip PCIe アダプタカード	PCIe-ROOTPORT-AD
ソフトウェア	
Libero® SoC ¹	本リファレンス デザインで使うソフトウェア バージョンの詳細は、デモ関連ファイルに含まれる <code>readme.txt</code> ファイルを参照してください。
SoftConsole®	
Microchip FPGA_GUI_Pack ²	バージョン 1.0



重要:

1. 本書に記載した Libero® SmartDesign と設定画面のスクリーンショットは説明用であり、実際の表示と異なる場合があります。最新バージョンについては Libero Design を参照してください。
2. Microchip FPGA_GUI_Pack には、Microchip FPGA デモ GUI アプリケーションの動作に必要なランタイム エンジンとドライバが含まれています。

1.2. 前提条件 (質問する)

デモを開始する前に以下の手順を実行する必要があります。

1. Libero SoC Documentation から Libero® SoC をダウンロードし、ホスト PC にインストールします。



重要: ModelSim®、Synplify Pro®、FTDI ドライバの最新バージョンは Libero SoC インストール パッケージに含まれています。

2. www.microchip.com/en-us/application-notes/AN4664 から PCIe ルートポート デモ関連ファイル(mpf_an4664_df.zip)をダウンロードし、www.microchip.com/en-us/application-notes/AN4597 から PCIe エンドポイント デモ関連ファイル(mpf_an4597_df.zip)をダウンロードします。
3. Libero プロジェクトの作成方法は「[補遺 3: TCLスクリプトの実行](#)」を参照してください。
4. ダウンロードしたファイルを解凍し、「\mpf_an4664_df\GUI」フォルダから「setup.exe」ファイルを実行する事により、PolarFire® PCIe ルートポート デモ GUI アプリケーションをインストールします。お客様のシステム内で FPGA_GUI_Pack が既に利用可能となっていない場合、インストールの最後で FPGA_GUI_Pack をダウンロードしてインストールするよう促される場合があります。
5. FPGA_GUI_Pack は、[Microchip FPGA_GUI_Pack](#) から手動でダウンロードしてインストールする事もできます。

1.3. デモデザイン (質問する)

図 1-1 に、PCIe ルートポート デザインの概略ブロック図を示します。PolarFire PCIe ルートポートは、任意の PCIe エンドポイントまたはブリッジとの PCIe リンクを確立できます。ユーザ アプリケーションは、ECAM (Enhanced Configuration Access Mechanism)機能を使って、エンドポイント デバイスをエミュレートします。ユーザ アプリケーションは、ルートポートからの AXI トランザクションも開始します。これらの AXI トランザクションは、エンドポイントへの PCIe コンフィグレーション空間トランザクション(CfgWr/CfgRd)またはメモリ トランザクション(MWr/ MRd)に変換されます。

図 1-1. ブロック図

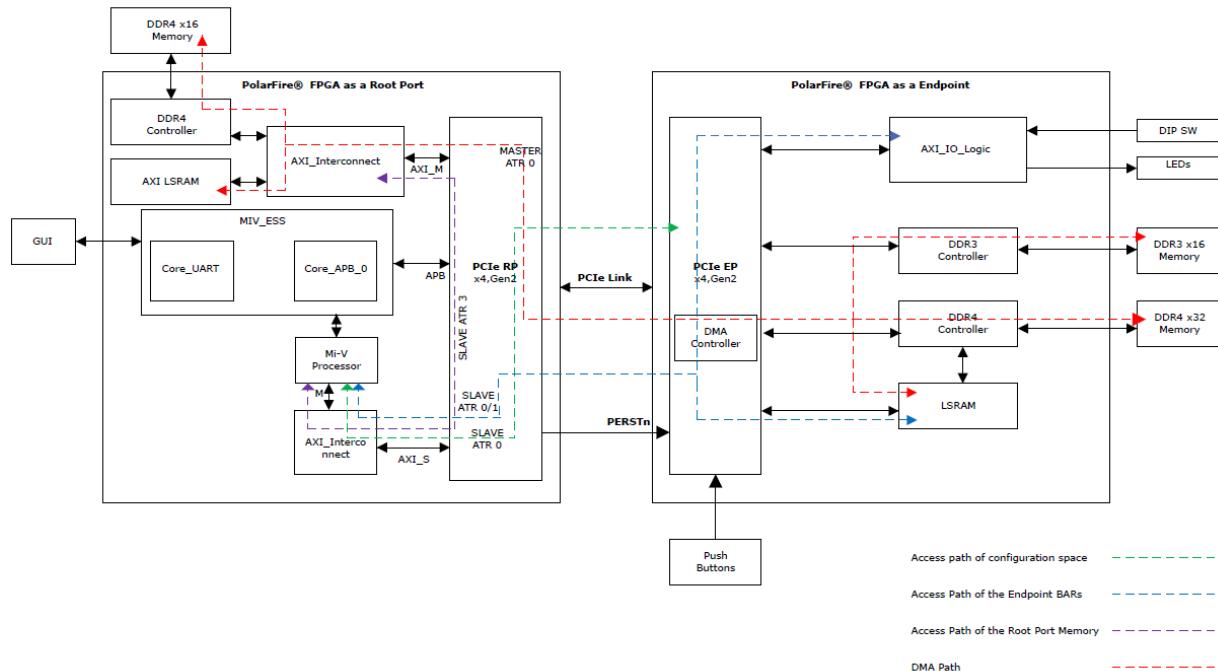


図 1-1 内の PCIe ルートポート デザイン内のデータフローの要点:

1. MIV_ESS_C0 UART ブロックは GUI と通信する
2. Mi-V ソフトプロセッサはコア MIV_ESS_C0 APB を介して MIV_ESS_C0 UART ブロックに対するデータの読み書きを行う
3. Mi-V ソフトプロセッサは PCIe_APB/PCIe_AXI スレーブ インターフェイスを介して対応する PCIe コマンドを PF_PCIE_0 ブロックに渡す
4. PCIe 要求およびコンプリഷン TLP(Transaction Layer Packet)は、ルートポートとエンドポイントの間でシリアルリンクを介して送受信される
5. インバウンド TLP は、PF_PCIE の AXI_1_MASTER ポート上で AXI トランザクションとして反映される
6. Mi-V ソフトプロセッサは、PCIe_AXI バス インターフェイスを使って AXI_1_SLAVE からデータを読み出す
7. Mi-V ソフトプロセッサは、コア MIV_ESS_C0 APB を使ってデータを MIV_ESS_C0 UART スレーブ インターフェイスに書き込む事により GUI にデータを渡す

図 1-1 内のルートポートからエンドポイントへの DMA フローの要点:

1. Mi-V ソフトプロセッサは、SLAVE ATR0 パスを介してルートポートおよびエンドポイント コンフィグレーション空間にアクセスする事により、エンドポイントをエニュメレートする



重要: ATR (Address Translation Register)は、PCIe アドレス空間から AXI マスターへのアドレス変換を実行します。ATR の詳細は [『PolarFire Family PCI Express User Guide』](#) を参照してください。

2. Mi-V ソフトプロセッサは、SLAVE ATR 0/1 パスを介してエンドポイント BAR 0/2 にアクセスする
3. Mi-V ソフトプロセッサは、SLAVE ATR 3 パスを介してルートポート LSRAM メモリにアクセスする
4. DMA は、GUI アプリケーション上のユーザによる選択に応じて実行される

1.3.1. メモリと周辺モジュールのアドレスマップ (質問する)

ルートポート デモデザインのメモリと周辺モジュールのアドレスマップを以下に記載します。

Mi-V 周辺モジュールとメインメモリのアドレスマップは下記の通りです。

- APB インターフェイス: 0x60000000~0x6FFFFFFF
- AXI インターフェイス: 0x70000000~7FFFFFFF
- TCM(Tightly Coupled Memory)メモリ インターフェイス: 0x80000000~0x8FFFFFFF

表 1-2 に、Mi-V を PF_PCIE へ接続するバス インターフェイスのアドレスマップを示します。

表 1-2. Mi-V と PF_PCIE のアドレスマップ

バス インターフェイス/ コンポーネント	概要	メモリマップ
PCIe_APB	このバス インターフェイスは PCIe レジスタにアクセスするために使われます。	0x63000000~0x63FFFFFF
MIV_ESS_C0 UART	このブロックは、Mi-V プロセッサを外部へ接続するための UART インターフェイスを確立します。	0x61000000~0x61FFFFFF
PCIE_1_PERST_OUT_N	これはルートポートに接続されたリンクパートナー向けの PERSTn 信号を生成するためには使われます。	0x6300A150

表 1-2. Mi-V と PF_PCIE のアドレスマップ(続き)

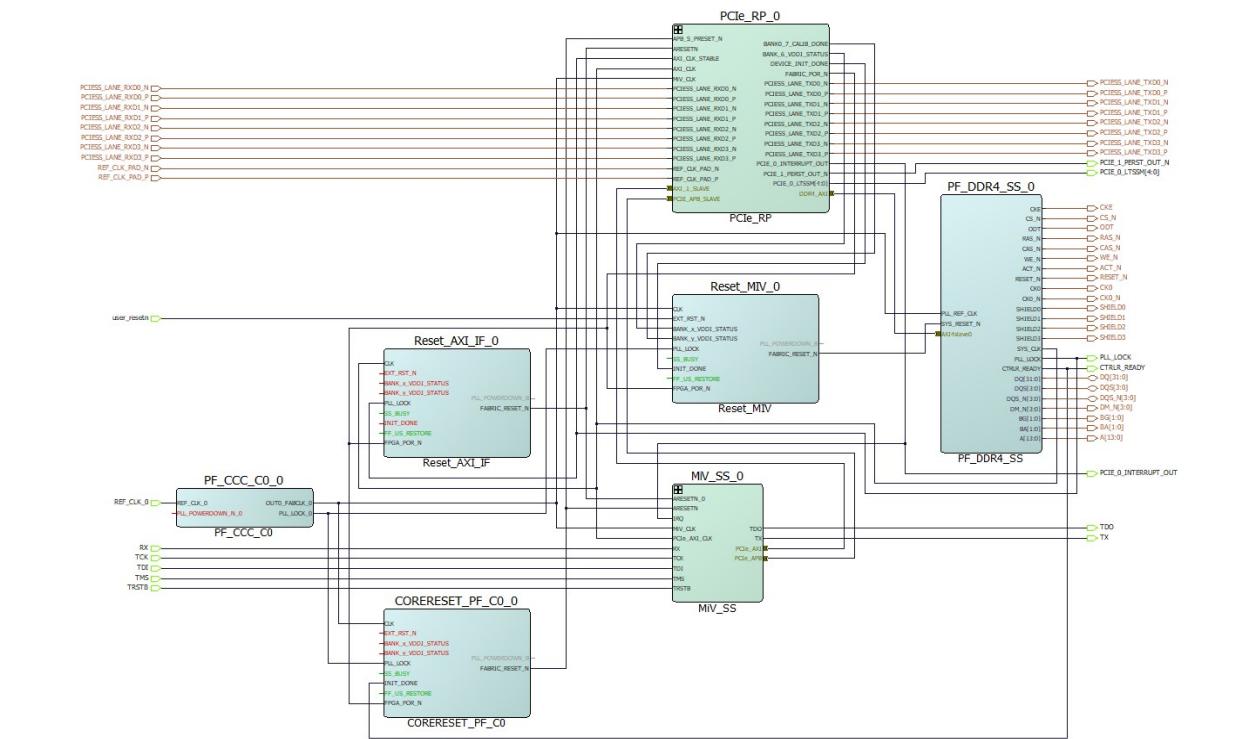
バス インターフェイス/ コンポーネント	概要	メモリマップ
PCIe_AXI	このバス インターフェイスは、EP コンフイグレーション空間または BAR 空間アクセス用の PCIe AXI スレーブです。	0x70000000～0x7000FFFF: コンフィグレーション空間(Mi-V は PCIe APB を介して設定) 0x71000000～0x7100FFFF: EP BAR0 空間 0x72000000～0x7200FFFF: EP BAR2 空間 0x73000000～0x73FFFFFF: RP AXI マスター-LSRAM/DDR4 (Mi-V は PCIe APB を介して設定)
TCM	このブロックは Mi-V プロセッサのメインメモリです。	0x80000000～0x8FFFFFFF

PF_PCIE ブロックは、AXI_1_master バス インターフェイスを介して DDR4 および LSRAM ブロックへ接続します。DDR4 のアドレスマップは 0x10000000～0x1FFFFFFF、LSRAM のアドレスマップは 0x00000000～0x00000FFF です。

1.3.2. デモデザインの実装(質問する)

図 1-2 に、PCIe ルートポート デモデザインの Libero SoC ソフトウェア実装を示します。

図 1-2. PCIe ルートポート デモデザイン



最上位層の設計は下記の SmartDesign コンポーネントとメモリ コントローラ サブシステムを含みます。

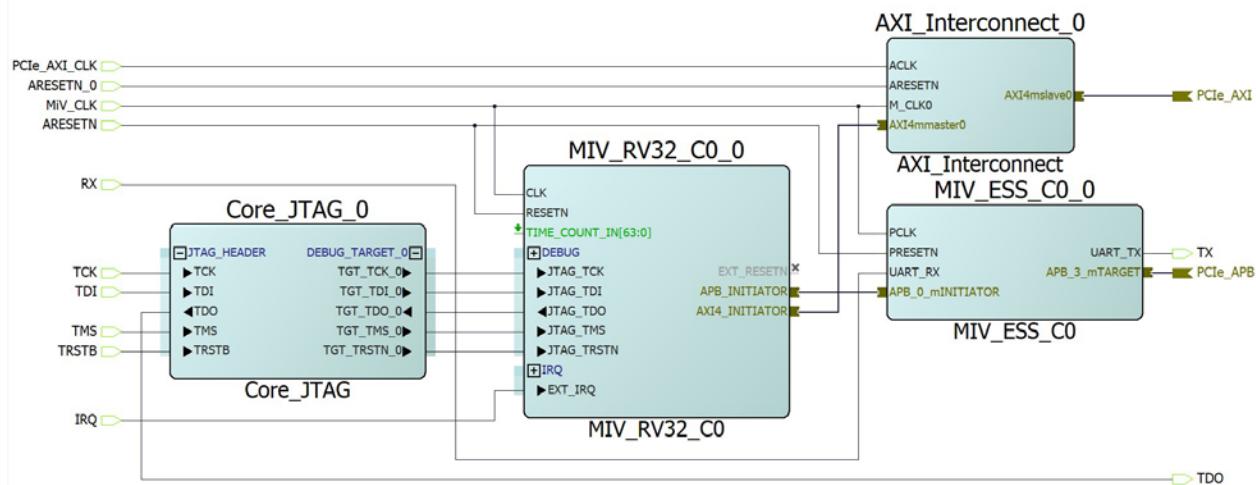
- MIV_SS_0
- PCIe_RP_0
- DDR4

重要: PERSTn は、PCI Express Base 仕様と PCI Express Card Electromechanical 仕様の両方で定義された基本的リセット信号です。このリセット信号は、PCIe ファブリック階層の全体をリセットするためにルートポートから PCIe スロットを介して発行されます。MI-V プロセッサ上で動作するルートポート フームウェアは、PCIE_APB_SLAVE インターフェイスを介して PERSTn 信号をアサートできます。ホストの電源を再投入すると、ルートポート内の PCIe コントローラが初期化されるまで、PERSTn 信号が PCIE_INIT_MONITOR_0 によってアサートされます。

1.3.2.1. Mi-V サブシステム (質問する)

図 1-3 に、MIV_SS_0 のサブブロックを示します。

図 1-3. MIV_SS_0 SmartDesign



1.3.2.1.1. MIV_RV32_C0 (質問する)

MIV_RV32_C0 (MIV_RV32_C0) は、0x80000000 のリセットベクタ アドレスで設定されます。リセット後に、プロセッサはこのアドレスから命令の実行を開始します。プロセッサのメインメモリのアドレス範囲は 0x80000000～0x8FFFFFFF です。

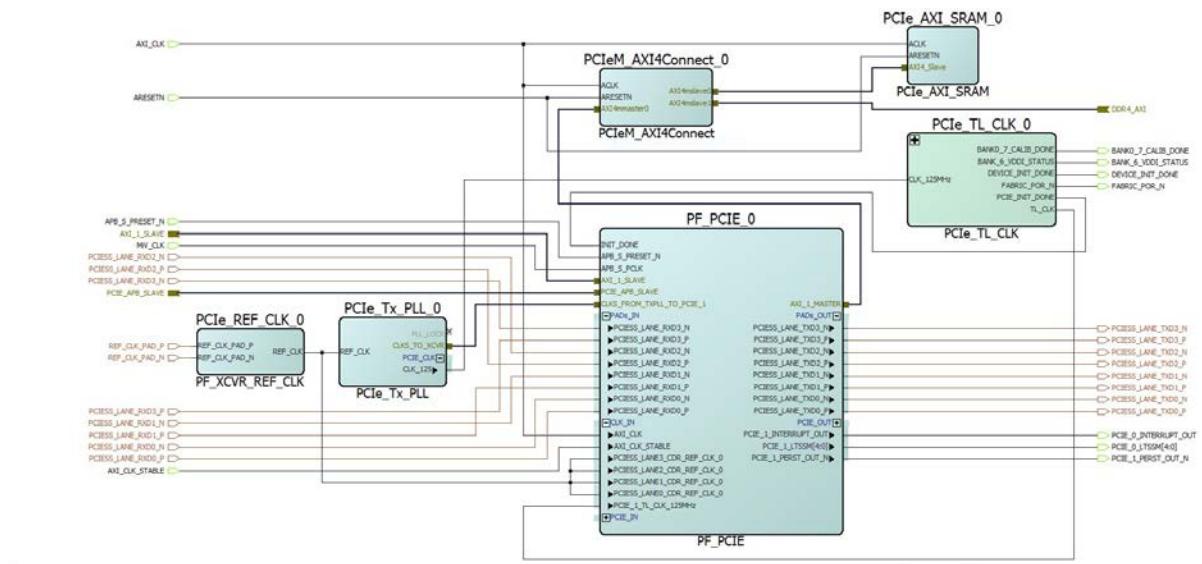
AXI_Interconnect_0 および MIV_ESS_C0 ブロックは、MIV_RV32_C0 ブロックを下記へ接続します。

- PCIe_APB スレーブ インターフェイスを介して PF_PCIE_0 ブロックに接続
MIV_RV32_C0 ブロックは PCIe_APB スレーブ インターフェイスを介して PCIe 制御レジスタにアクセス
 - PCIe_AXI スレーブ インターフェイスを介して PF_PCIE_0 ブロックに接続
 - APB スレーブ インターフェイスを介して MIV_ESS_C0 UART ブロックに接続
- これらのスレーブは下記のアドレスで接続されます。
- PCIe_APB スレーブ: 0x63000000
 - PCIe_AXI: 0x70000000
 - MIV_ESS_C0 UART_APB スレーブ: 0x61000000

1.3.2.2. PCIe ルートポート サブシステム(質問する)

図 1-4 に、PCIe RP 0 のサブブロックを示します。

図 1-4. PCIe_RP SmartDesign



1.3.2.2.1. PF_PCIE_0 (質問する)

PF_PCIE_0 IP ブロックは、PCIESS をルートポート(PClE 1)として設定するために使われます。PCIESS ブロックは、PCIe コントローラ アクセス用に 4 レーン、5 Gbps データレート、APB インターフェイス 向けに設定されます。

1.3.2.2.2. PCIe_Tx_PLL_0 (質問する)

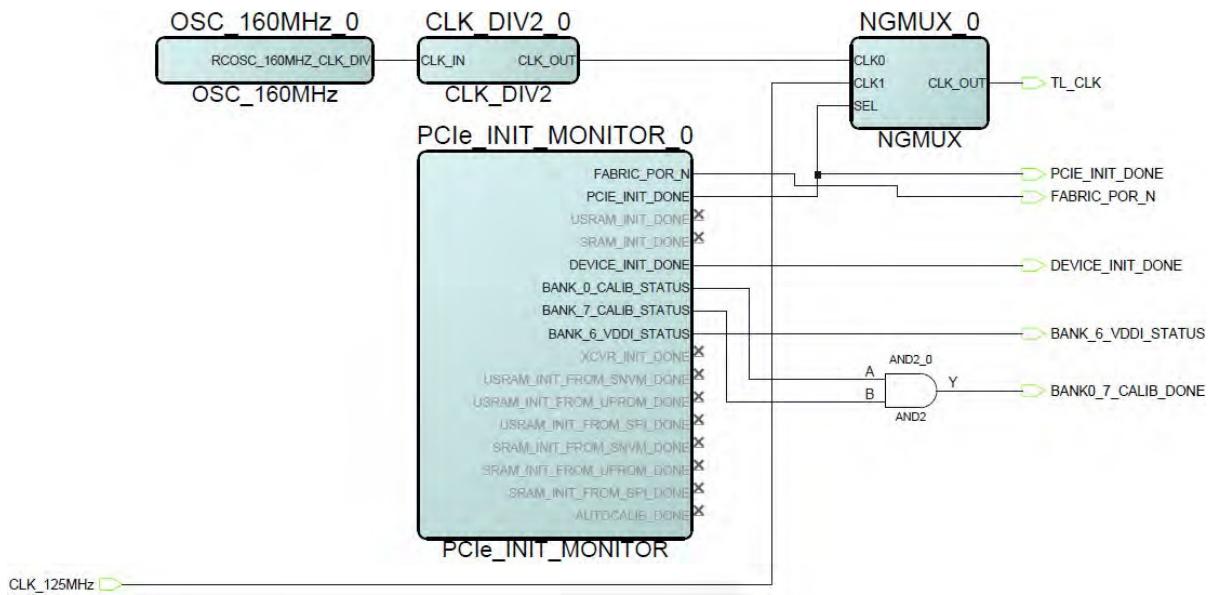
PCIe_Tx_PLL_0 (送信 PLL) は、100 MHz の参照クロックと 5000 Mbps 速度対応のビットクロック向けに設定されます。

PolarFire FPGA トランシーバはハーフレート アーキテクチャを採用しており、内部高速バスでクロックの両方のエッジを使います。これによりクロックはデータレートの 1/2 の速度で動作できるため、動的消費電力を削減できます。PCIe モード中のトランシーバは 2500 MHz ビットクロックを要求します。

1.3.2.2.3. PCIe_TL_CLK_0 SmartDesign (質問する)

図 1-5 に、PolarFire デバイス向けの PCIe TL CLK を実装する PCIe_TL_CLK SmartDesign を示します。PCIe TL CLK は、Tx PLL の CLK_125 MHz に接続する必要があります。PolarFire デバイスでは、TL CLK は PCIe の初期化後にのみ利用可能です。PCIe の初期化中に TL CLK を駆動するために、80 MHz クロックが内部 160 MHz オシレータから生成されます。PCIe 初期化後にこのクロックを必要な CLK_125 MHz へ切り換えるために、NGMUX が使われます。

図 1-5. PCIe_TL_CLK_0 SmartDesign



1.3.2.2.4. PCIeM_AXI4Connect_0 (質問する)

PCIeM_AXI4Connect_0 (CoreAXI4Interconnect)バスは 1x マスタ/2x スレーブ向けに設定され、DMA 動作向けに PF_PCIE_0 を PCIe_AXI_SRAM_0 および DDR4 と接続するために使われます。

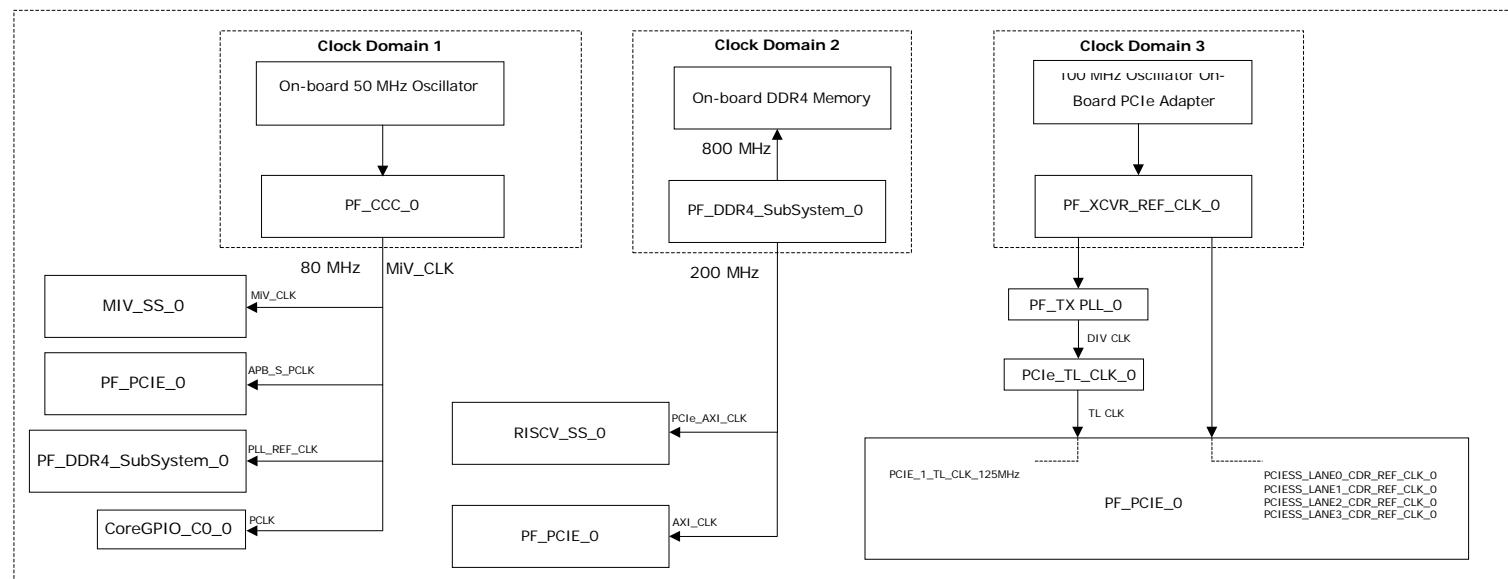
1.3.2.3. DDR4 サブシステム (質問する)

DDR4 サブシステムは、AXI4 64 ビット インターフェイスを介して 32 ビット DDR4 メモリにアクセスするよう設定されます。DDR4 メモリの初期化およびタイミング パラメータは、PolarFire 評価用キット上の DDR4 メモリごとに設定されます。

1.4. クロック構造 (質問する)

図 1-6 に、デモデザインのクロック構造を示します。

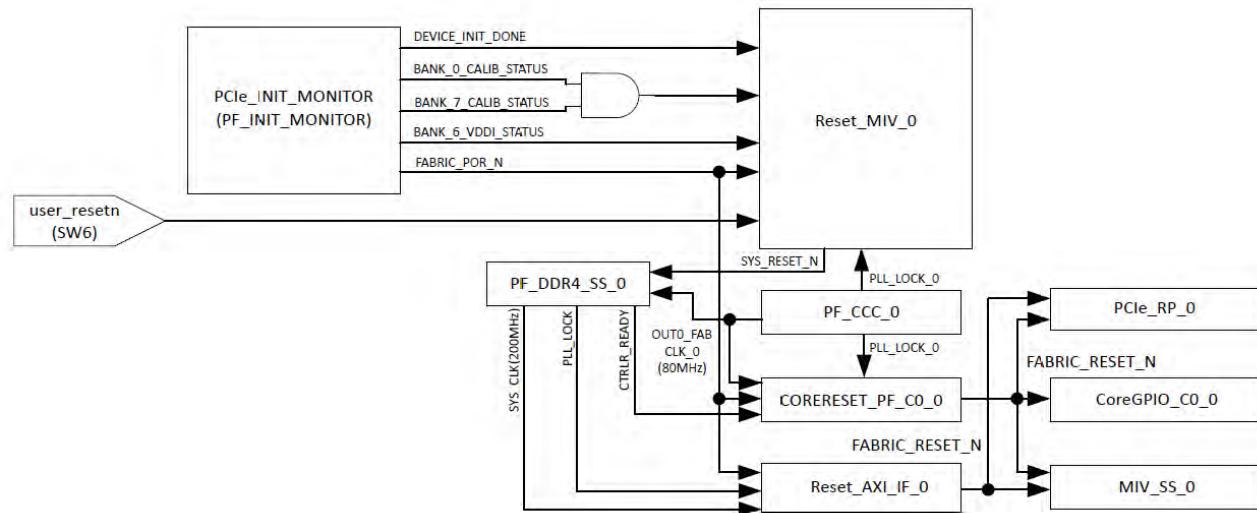
図 1-6. クロック構造



1.5. リセット構造 (質問する)

図 1-7 に、PCIe ルートポート デモデザインのリセット構造を示します。

図 1-7. リセット構造



Reset_AXI_IF_0 (CoreReset_PF) ブロックは、PCIe_RP_0 および MIV_SS_0 ブロックを駆動する FABRIC_RESET_N 信号を生成するために、PF_DDR4_SS_0 IP の「PLL_LOCK」信号を DDR4 システムクロック(200 MHz)に同期させます。

PF_DDR4_SS ブロックを駆動する SYS_RESET_N を生成するために、Reset_MIV_0(CoreReset_PF) ブロックは外部の user_resetn (PolarFire 評価用ボード上の SW6) と DEVICE_INIT_DONE(PF_INIT_MONITOR) を共に RISCV システムクロック(80 MHz)に同期させます。

CORERESET_PF_C0_0(CoreReset_PF) ブロックは PCIe_RP_0 と MIV_SS_0 を駆動する FABRIC_RESET_N 信号を生成するために、PF_DDR4_SS_0 IP の「CTRLR_READY」信号を RISCV システムクロック(80 MHz)に同期させます。

デバイス初期化の詳細は [『PolarFire Family Power-Up and Resets User Guide』](#) を参照してください。

CoreReset_PF IP コアの詳細は、Libero カタログから [『CoreReset_PF Handbook』](#) を参照してください。

2. Libero デザインフロー (質問する)

以下では、本デモデザインの Libero デザインフローについて説明します。Libero デザインフローには下記の手順が含まれます。

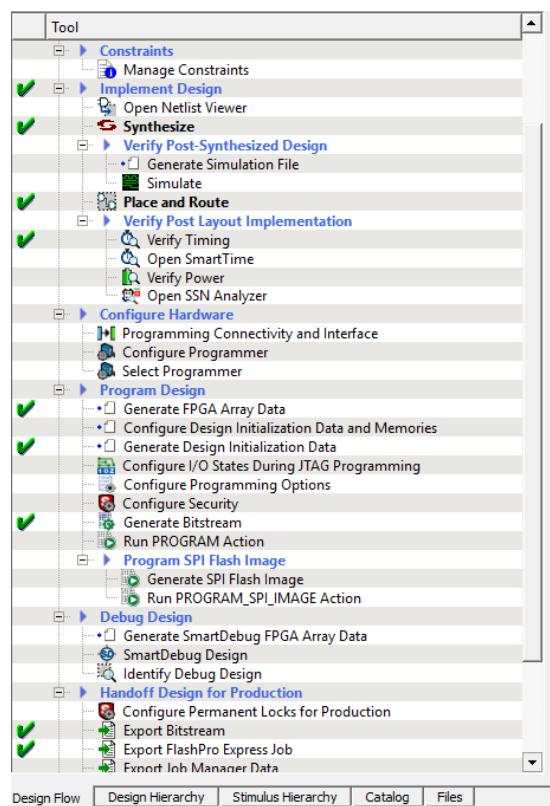
1. 論理合成
2. 配置配線
3. タイミングの検証
4. デザイン初期化データとメモリの設定
5. ビットストリームの生成
6. デバイスのプログラミング



重要: システム コントローラを使って PolarFire 内の TCM を初期化するには、論理合成の前に `miv_rv32_subsys_pkg.v` ファイル内でローカル パラメータ `I_cfg_hard_tcm0_en` を 1'b1 に設定する必要があります。『MIV_RV32 Handbook』内の TCM の説明を参照してください。この文書は Libero SoC カタログからダウンロードできます。

図 2-1 に、[Design Flow]タブ内のこれらのオプションを示します。

図 2-1. Libero デザインフロー オプション



2.1. 論理合成 (質問する)

デザインを論理合成する手順は以下の通りです。

1. 「Design Flow」 ウィンドウから [Synthesize] をクリックします。論理合成が成功すると、緑のチェックマークが表示されます。

2. [Synthesize]を右クリックして[View Report]を選択すると、論理合成レポートとログファイルが[Reports]タブに表示されます。

論理合成エラーとコンパイルエラーをデバッグするために top.srr と top_compile_netlist.log ファイルを確認する事を推奨します。

2.2. 配置配線(質問する)

回路ブロックの配置配線を行うには、I/O エディタを使って送信 PLL (TX_PLL)、XCVR_REF_CLK、PF_XCVR TX および RX レーン、PF_DDR4_SS_0 を配置する必要があります。

配置配線の手順は以下の通りです。

1. 「Constraints Manager」ウィンドウから、I/O エディタを使って送信 PLL、XCVR_REF_CLK、PF_XCVR TX および RX レーンを配置します。図 2-2 を参照してください。

図 2-2. I/O エディタ - [XCVR View]



2. NORTH_NW の位置に PF_DDR4_SS_0 を配置します(図 2-3 参照)。

図 2-3. PF_DDR4_SubSystem_0 の配置



3. 「Design Flow」ウィンドウから、[Place and Route]をダブルクリックします。配置配線が成功すると緑のチェックマークが表示されます(図 2-1 参照)。
4. [Place and Route]を右クリックして[View Report]を選択すると、[Reports]タブに配置配線レポートとログファイルが表示されます。

配置配線の制約達成範囲を top_place_and_route_constraint_coverage.xml ファイルで確認する事を推奨します。

2.2.1. リソース使用率(質問する)

リソース使用率レポートは、top_layout_log.log ファイル([Reports]タブ > [RP_Demo_Top report] > [Place and Route])に書き込まれます。表 2-1 に、配置配線後の回路リソース使用率を示します。これらの値は Libero の実行条件、設定、シード値に応じて多少変化する可能性があります。

表 2-1. リソース使用率

タイプ	使用量	総量	使用率(%)
4LUT	31943	299544	10.66
DFF	23193	299544	7.74

表 2-1. リソース使用率(続き)

タイプ	使用量	総量	使用率(%)
I/O レジスタ	0	1536	0

2.3. タイミングの検証 (質問する)

タイミングの検証手順は以下の通りです。

- 「Design Flow」ウィンドウから[Verify Timing]をクリックします。
- デザインがタイミング要件を満たすと緑のチェックマークが表示されます(図 2-1 参照)。
- [Verify Timing]を右クリックして[View Report]を選択すると、[Reports]タブにタイミング検証レポートが表示されます。

2.4. FPGA 配列データの生成 (質問する)

FPGA 配列データを生成するには、「Design Flow」ウィンドウ内で[Generate FPGA Array Data]をダブルクリックします。

FPGA 配列データが正常に生成されると緑のチェックマークが表示されます(図 2-1 参照)。

2.5. 回路初期化データとメモリの設定 (質問する)

回路初期化データとメモリを設定するため、以下の手順により TCM 初期化クライアントを生成し、選択された不揮発性メモリのタイプに基づいてそれを sNVM、μPROM、外部 SPI フラッシュのいずれかに追加します。以下の例では、TCM は μPROM から初期化されます。

この処理には、デバイスへの電源投入後に TCM を初期化するための入力として、ユーザ アプリケーション実行ファイル(hex ファイル)が必要です。hex ファイルは設計情報ファイルと一緒に提供されます。

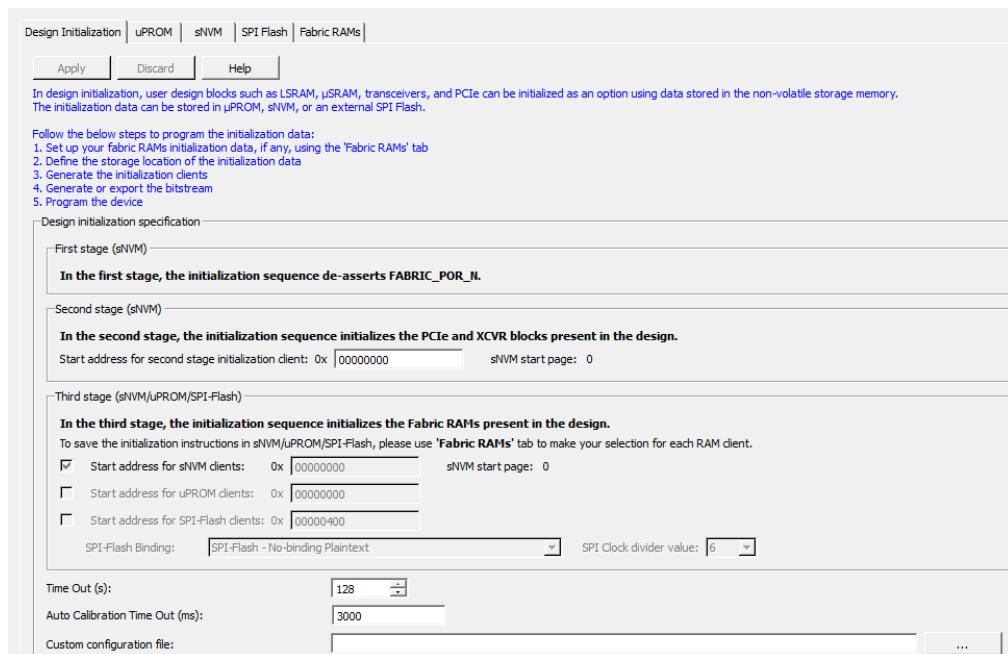
以下の手順により不揮発性メモリを選択して初期化クライアントを生成します。

- [Design Flow]タブ上で[Configure Design Initialization Data and Memories]をダブルクリックします。「Design and Memory Initialization」ウィンドウが開きます。
- [Third stage (uPROM/sNVM/SPI-Flash)] の下で μPROM を選択します(図 2-4 は sNVM を選択した場合を示す)。[Third Stage]枠内で μPROM を不揮発性メモリとして選択します(開始アドレスは既定値 0x00000000 のまま)。



重要: μPROM 内で指定された初期化クライアントは他に存在しないため、開始アドレスは既定値 0x00000000 のままです。

図 2-4. 「Design and Memory Initialization」 ウィンドウ



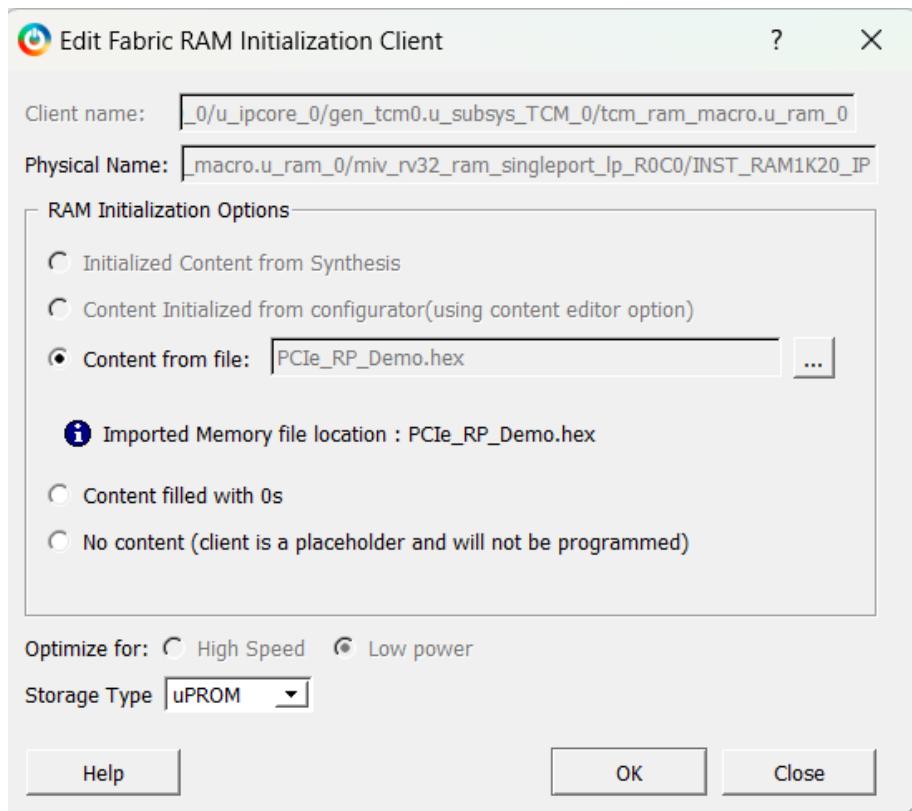
3. [Fabric RAMs]タブ上で、ロジカルインスタンスのリストから
MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0 を選択し、[Edit]をクリックします(図 2-5 参照)。
MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0 インスタンスは Mi-V プロセッサのメインメモリです。システムコントローラは、電源投入時にインポートされたクライアントを使ってこのインスタンスを初期化します。

図 2-5. [Fabric RAMs]タブ

Fabric RAMs						
Clients						
<input type="button" value="Load design configuration"/> <input type="button" value="Edit..."/> Initialize all clients from: <input type="button" value="User Selection"/>		<input checked="" type="checkbox"/> Filter out Inferred RAMs				
Logical Instance Name		PORTA Depth * Width: 8192x32 PORTB Depth * Width: 8192x32 Memory Content: PCIe_RP_Demo.hex Storage Type: uPROM Memory Source: Configurator				
1	MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0	1024x80	1024x80	No content	sNVM	Configurator
2	PCIe_RP_0/PCIe_AXI_SRAM_0					

4. 「Edit Fabric RAM Initialization Client」ダイアログ ボックス内で、[Content from file]フィールドの右隣にある「...」ボタンをクリックします(図 2-6 参照)。

図 2-6. 「Edit Fabric RAM Initialization Client」 ダイアログ ボックス



開いたダイアログ ボックス内で `mpf_an4664_df/HW/src/softconsole/ PCIe_RP_Demo.hex` ファイルを見つけてダブルクリックします。

重要: Mi-V アプリケーション コードを変更した場合、SoftConsole プロジェクトを Release モードでリビルドします。

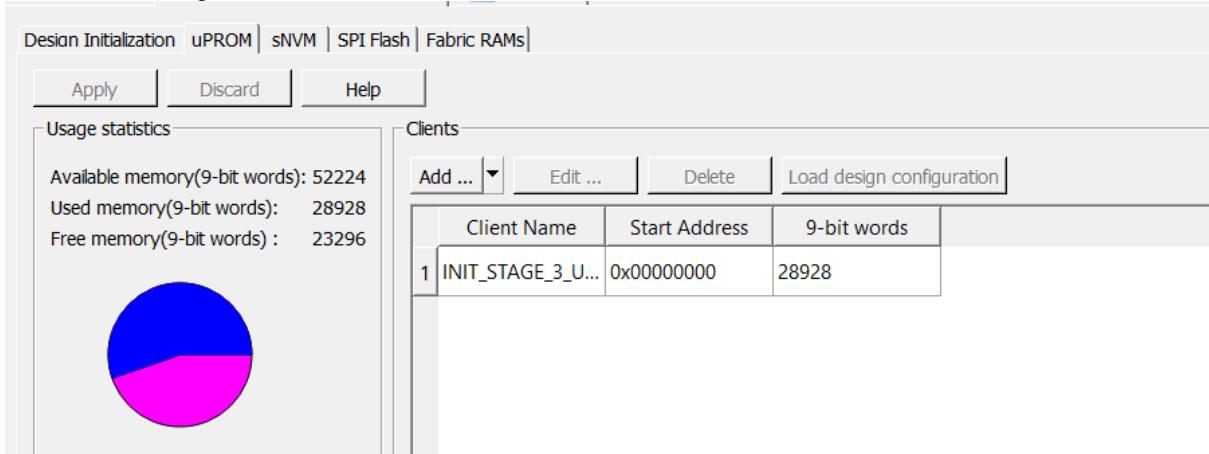
5. 「Edit Fabric RAM Initialization Client」 ウィンドウ内で[OK]をクリックします。
6. [Fabric RAMs]タブ上で[Apply]をクリックします(図 2-7 参照)。

図 2-7. ファブリック RAM 内容の適用

Design Initialization		uPROM	sNVM	SPI Flash	Fabric RAMs																		
		Apply	Discard	Help																			
Usage statistics																							
LSRAM Memory																							
Available Memory(Bytes):	2437120	Used Memory(Bytes):	104960	Free Memory(Bytes):	2332160																		
Clients																							
<input checked="" type="radio"/> Load design configuration <input type="button" value="Edit..."/> Initialize all clients from: <input type="button" value="User Selection"/>																							
<input type="checkbox"/> Filter out Inferred RAMs																							
<table border="1"> <thead> <tr> <th>Logical Instance Name</th> <th>PORTA Depth * Width</th> <th>PORTB Depth * Width</th> <th>Memory Content</th> <th>Storage Type</th> <th>Memory Source</th> </tr> </thead> <tbody> <tr> <td>1 MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0</td> <td>8192x32</td> <td>8192x32</td> <td>PCIe_RP_Demo.hex</td> <td>uPROM</td> <td>Configurator</td> </tr> <tr> <td>2 PCIe_RP_0/PCIe_AXI_SRAM_0</td> <td>1024x80</td> <td>1024x80</td> <td>No content</td> <td>sNVM</td> <td>Configurator</td> </tr> </tbody> </table>						Logical Instance Name	PORTA Depth * Width	PORTB Depth * Width	Memory Content	Storage Type	Memory Source	1 MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0	8192x32	8192x32	PCIe_RP_Demo.hex	uPROM	Configurator	2 PCIe_RP_0/PCIe_AXI_SRAM_0	1024x80	1024x80	No content	sNVM	Configurator
Logical Instance Name	PORTA Depth * Width	PORTB Depth * Width	Memory Content	Storage Type	Memory Source																		
1 MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0	8192x32	8192x32	PCIe_RP_Demo.hex	uPROM	Configurator																		
2 PCIe_RP_0/PCIe_AXI_SRAM_0	1024x80	1024x80	No content	sNVM	Configurator																		

7. `MIV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0` インスタンス向けの初期化クライアントは `uPROM` 内に生成され保存されます。この手順を検証するため、作成された「Third stage」クライアントが `[uPROM]` タブ内に表示される事を確認します(図 2-8 参照)。

図 2-8. Third Stage INIT クライアント



既定値により、「First stage」および「Second stage」クライアントは sNVM 内で生成され保存されます。

2.6. ビットストリームの生成 (質問する)

ビットストリームは以下の手順により生成します。

1. [Design Flow]タブ上で[Generate Bitstream]をクリックします。
ビットストリームが正常に生成されると緑のチェックマークが表示されます(図 2-1 参照)。
2. [Generate Bitstream]を右クリックし、[View Report]を選択すると、[Reports]タブに対応するログファイルが表示されます。

2.7. デバイスのプログラミング (質問する)

ビットストリームを生成した後に、PolarFire デバイスをプログラミングする必要があります。PolarFire デバイスのプログラミング手順は以下の通りです。

1. ルートポート デバイスとして使われる評価用ボード上のジャンパが図 2-2 の通りに設定されている事を確認します。

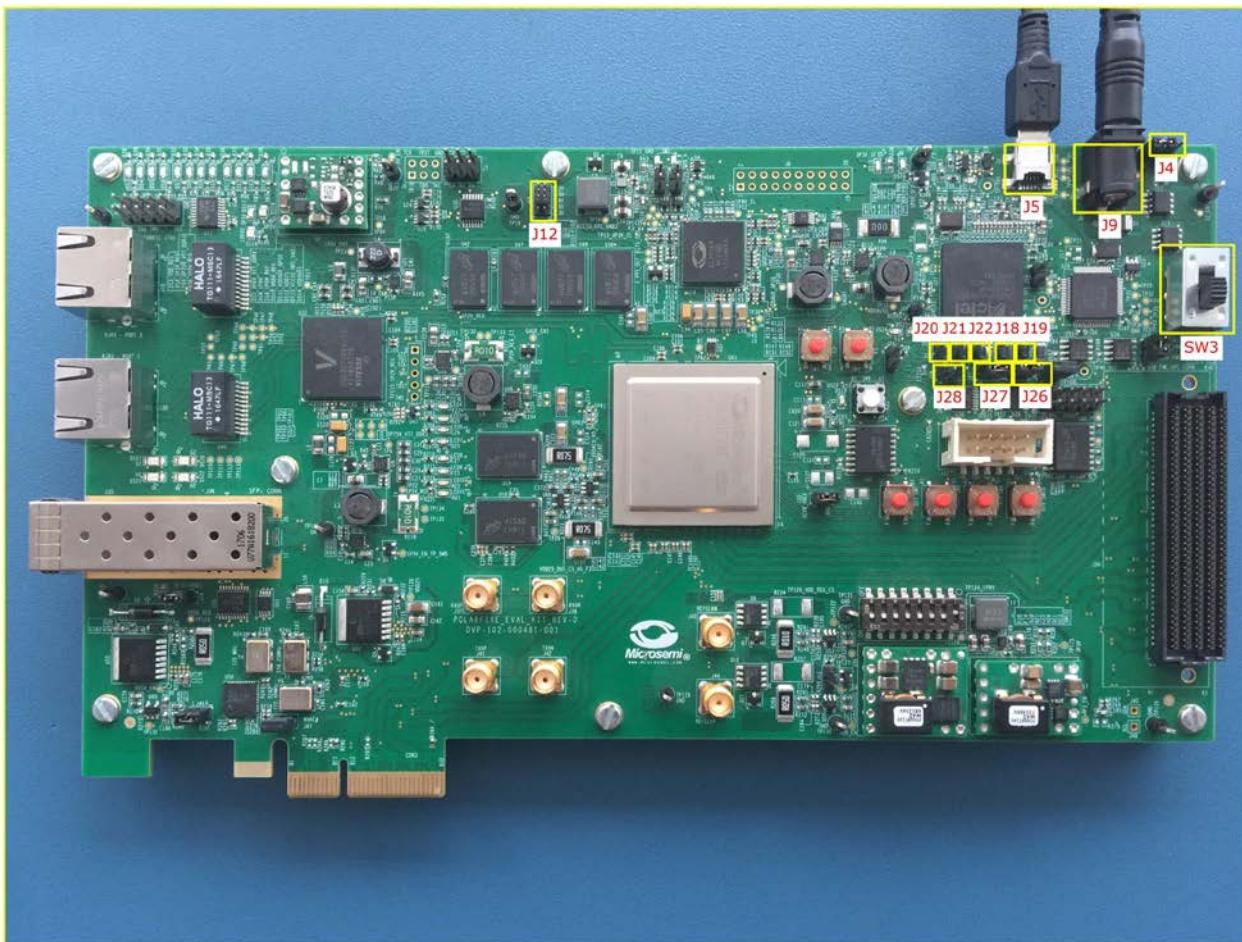
表 2-2. ジャンパ設定

ジャンパ	説明	既定値
J18、J19、J20、J21、J22	ピン 2-3 間を短絡: FTDI 経由で PolarFire® FPGA をプログラミングする	短絡
J28	ピン 1-2 間を短絡: 内蔵 FlashPro5 経由でプログラミングする	開放
J4	ピン 1-2 間を短絡: SW3 を使って手動で電源を ON/OFF する	短絡
J12	ピン 3-4 間を短絡: 2.5V	短絡

2. 電源ケーブルをボード上の J9 コネクタに接続します。
3. ホスト PC からの USB ケーブルをボード上の J5 (FTDI ポート)に接続します。
4. SW3 スライドスイッチを使ってボードを電源 ON にします。
5. Libero 上の[Design Flow]タブ内で[Run PROGRAM Action]をダブルクリックします。

デバイスが正常にプログラミングされると緑のチェックマークが表示されます(図 2-1 参照)。デバイスのプログラミングが完了したら「3. デモのセットアップ」に進みます。

図 2-9. 評価用ボード



3. デモのセットアップ (質問する)

デモのセットアップ手順は以下の通りです。

1. 2つの評価用ボード上の PolarFire デバイスをプログラミングします。
2. PCIe アダプタカードを介して 2つの PolarFire 評価用ボードを接続します。

以下では 2つのボードを下記の通りに表記します。

- ボード A: ルートポート設計を実行するボード
- ボード B: エンドポイント設計を実行するボード

3.1. 2つのボード間の接続 (質問する)

下記の手順により、Microchip PCIe アダプタカードを使って 2つのボード間を接続します。

1. PCIe アダプタカード上の J1 ジャンパのピン 1-2 間が短絡されている事を確認します。
2. PCIe アダプタカード上の J2 ジャンパのピン 1-3 間が開放されている事を確認します。
3. アダプタカードの CON1 をボード A の CON3 (PCIe スロット)に接続します。
4. アダプタカードの CON2 をボード B の CON3 (PCIe スロット)に接続します。
5. ホスト PC からの USB ケーブルをボード A の J5 (FTDI ポート)に接続します。
6. ホスト PC からの USB ケーブルをボード B の J5 (FTDI ポート)に接続します。
7. 電源ケーブルを PCIe アダプタカードの J3 コネクタに接続します。
8. SW3 スライドスイッチを使ってボード A とボード B の電源を ON にします。
9. SW1 スライドスイッチを使って PCIe アダプタカードに電源を投入します。

ボード A とボード B は PCIe アダプタカードを介して給電されます。図 3-1 に、アダプタカードを介して接続された 2つの評価用ボードを示します。

図 3-1. デモのセットアップ



4. デモの実行 (質問する)

PCIe ルートポート デモの実行に必要な GUI をインストールして使用する方法を下記の 4 項目に分けて説明します。

- GUI のインストール
- エニュメレーションデータの表示
- コントロール プレーン コマンドの実行
- データ プレーン コマンドの実行

4.1. GUI のインストール (質問する)

GUI をインストールする手順は以下の通りです。

1. `mpf_an4664_df.zip` ファイルを解凍します。`mpf_an4664_df\GUI` フォルダから `setup.exe` ファイルをダブルクリックします。
2. インストール ウィザードに表示される指示に従います。

インストールが成功すると、ホスト PC デスクトップの [Start] メニューに `PCIe_Root_Port_GUI` が表示されます。GUI アプリケーションがインストール済みではない場合、「1.2. 前提条件」を参照してください。

4.2. エニュメレーションデータの表示 (質問する)

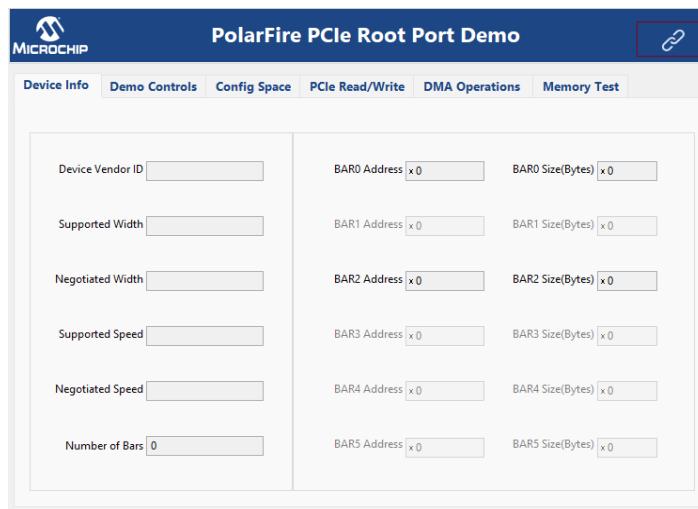
デモを開始する前に以下を確認します。

1. 一方のボード上の PolarFire FPGA には PCIe ルートポート デザインがプログラミングされ、他方のボード上の PolarFire FPGA には PCIe エンドポイント デザインがプログラミングされている事
2. 2 つのボードが Microchip PCIe アダプタカードを介して接続され給電される事
3. ルートポート ボード上の LED9/10/11 が点灯している事
 - これは PCIe リンクが起動済みである事を示します。点灯しない場合はボードの電源を再投入する必要があります。

GUI を起動してエニュメレーションデータを表示するために以下の手順を実行します。

1. タスクバーから [Start] ボタンをクリックし、[PCIe_Root_Port_GUI] を選択します。
2. [Connect] をクリックする事により、GUI をルートポート ボードへ接続します(図 4-1 参照)。

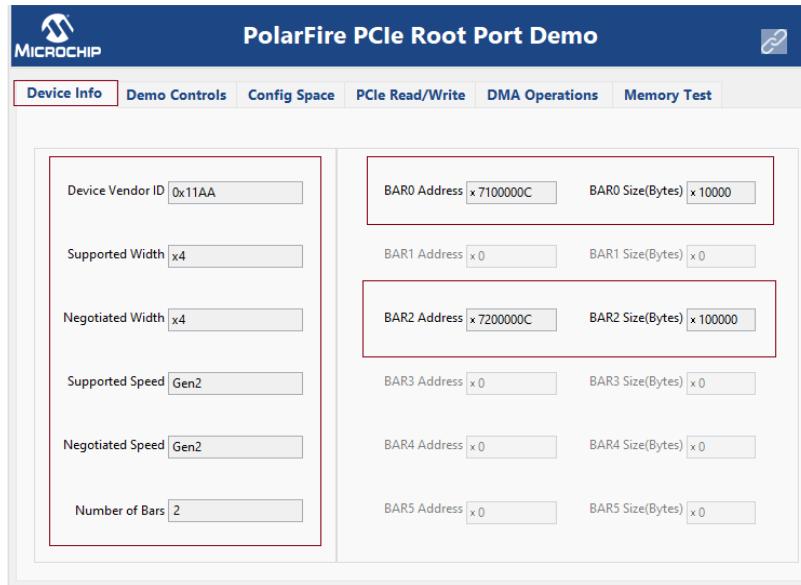
図 4-1. PCIe ルートポート GUI



GUI はルートポート デバイスの UART COM ポートの検出を開始します。

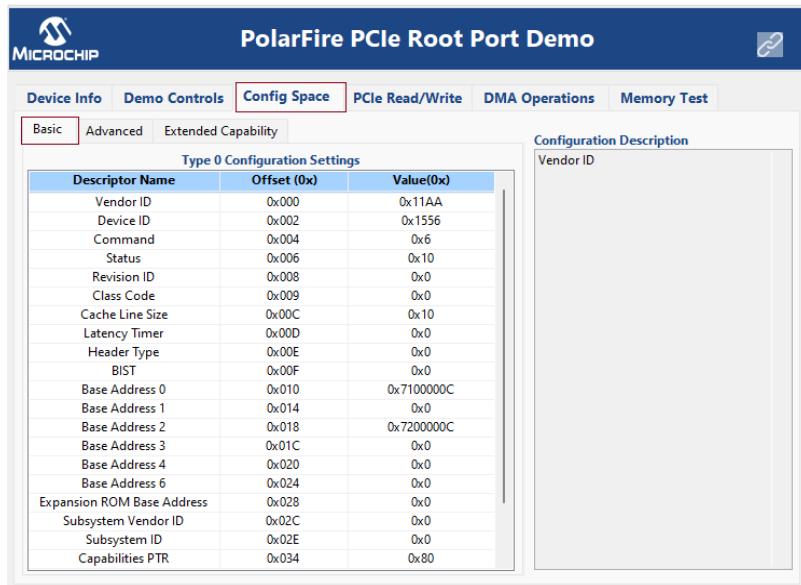
3. COM ポートへ正常に接続された後に、Mi-V ソフトプロセッサは PCIe EP デバイスをエニュメレートし、コンフィグレーション空間データを GUI へ送信します。
4. [Device Info]タブをクリックするとエンドポイント デバイスの情報が表示されます。図 4-2 を参照してください。

図 4-2. エンドポイント デバイスの情報



5. [Config Space]タブをクリックするとエンドポイントの基本的な Type 0 コンフィグレーション設定が表示されます。

図 4-3. エンドポイント コンフィグレーション空間 - [Basic]



6. [Advanced] タブをクリックすると、エンドポイントの「MSI(Message Signaled Interrupts) Capabilities」が表示されます(図 4-4 参照)。

図 4-4. エンドポイントコンフィグレーション空間 - [Advanced]

Descriptor Name	Offset (0x)	Value
Cap ID	0xE0	0x5
Next pointer	0xE1	0xF8
Message Control	0xE2	0xA5
Message Address	0xE4	0x190
Message upper Address	0xE8	0x0
Message data	0xEC	0x0

7. 同様に、[Power Management Capability]および[PCIe Capability]タブをクリックすると、関連データが表示されます。

4.3. コントロールプレーンコマンドの実行(質問する)

本デモでは、ルートポートは下記のコントロールプレーン動作を開始します。

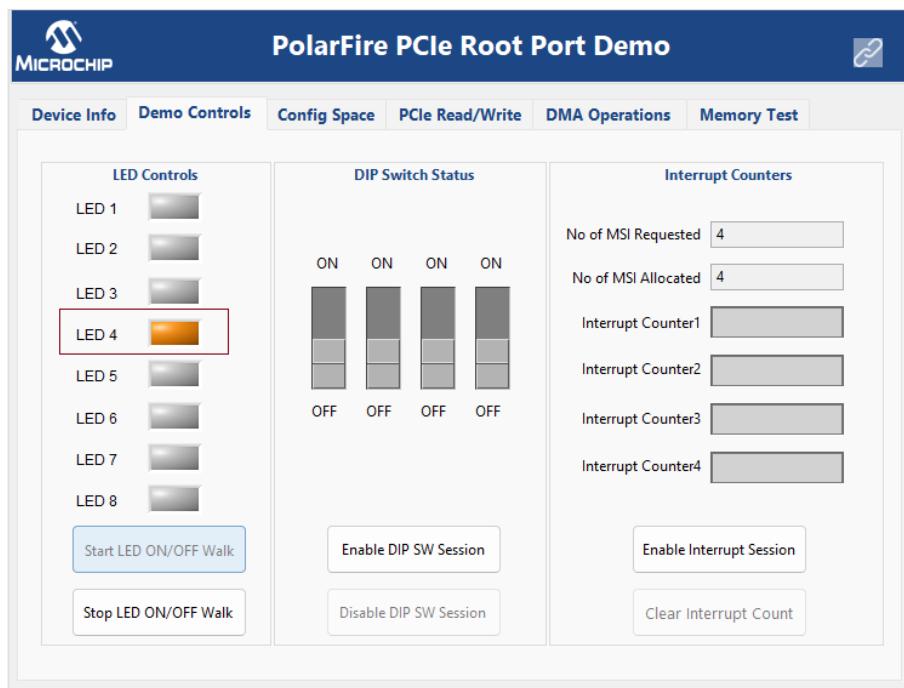
- ・エンドポイントLEDの制御
- ・DIPスイッチステータスの読み出し
- ・MSIカウント値の読み出し
- ・BAR2メモリ読み書きコマンド

4.3.1. エンドポイントLEDの制御(質問する)

ルートポートはエンドポイントのLEDを設定数に応じて順次点灯(ON/OFF Walk)します。LEDコマンドは以下の手順により発行します。

1. [Demo Controls]タブをクリックします。
2. いずれか1つのLEDを選択します。LED3を選択した場合の例を図4-5に示します。

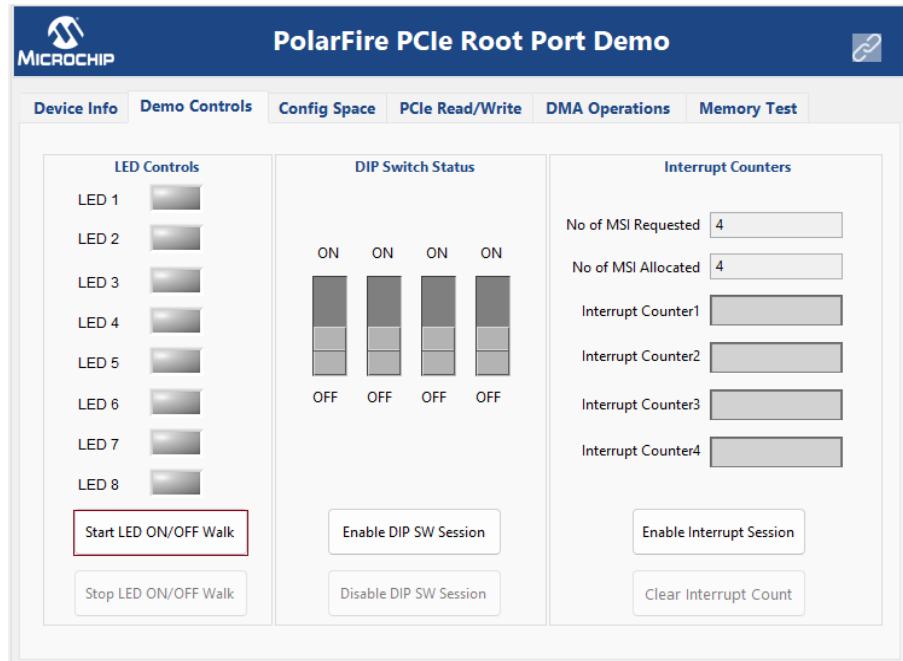
図 4-5. 単一 LED の制御



GUI は RISC-V プロセッサに対して LED 点灯要求を開始し、この要求は PF_PCIE_0 ブロックに渡されます。PF_PCIE_0 は BAR2 MWr パケットをエンドポイントへ送信します。結果として、エンドポイントボード上の LED3 が点灯します。

3. [Start LED ON/OFF Walk]ボタンをクリックします。

図 4-6. LED ON/OFF Walk

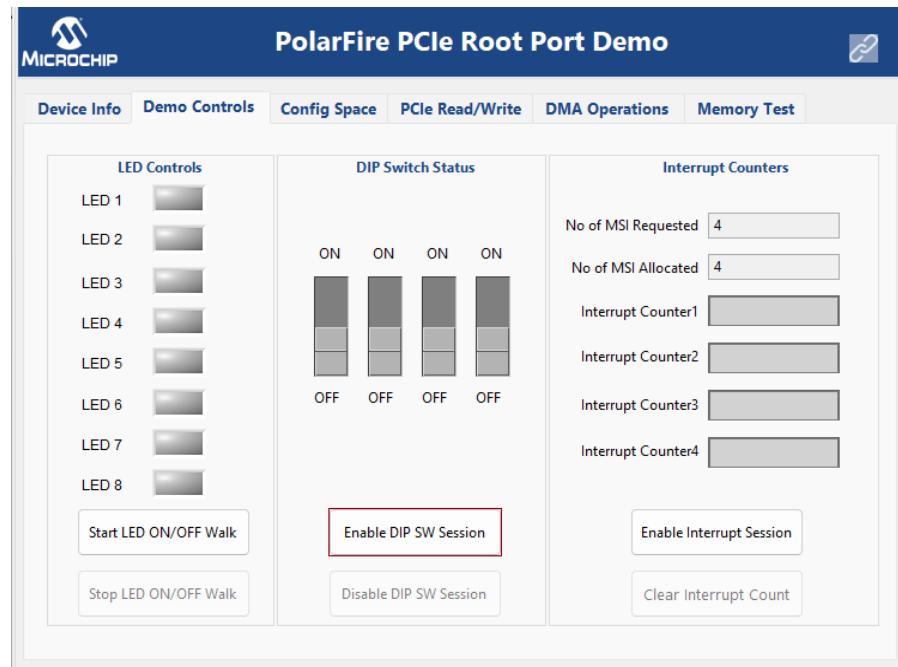


GUI は LED ON/OFF Walk 要求を開始します。結果として、LED の ON/OFF が LED 番号順に実行された後で逆順に実行されます。

4.3.2. エンドポイント DIP スイッチ ステータスの読み出し(質問する)

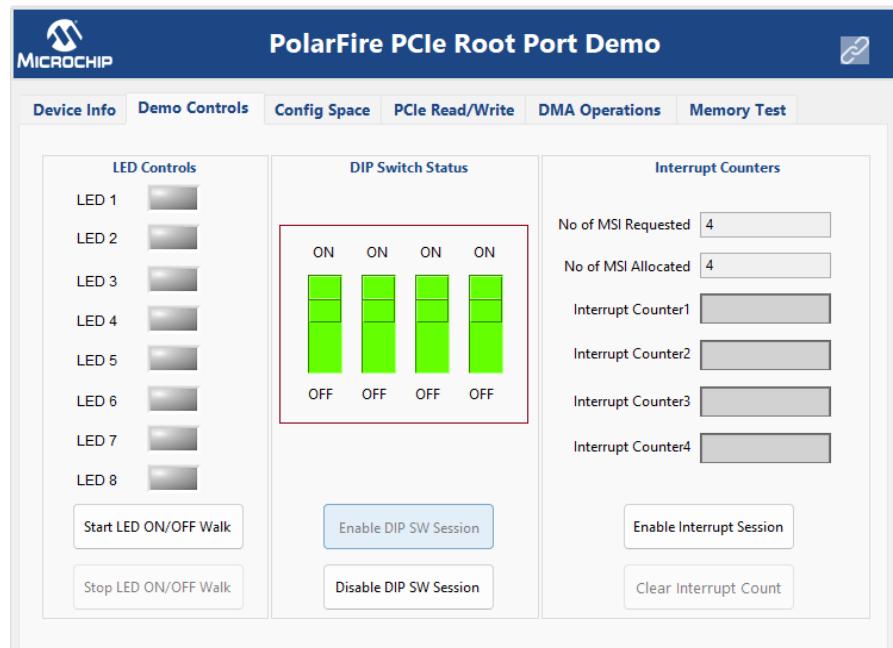
DIP スイッチ ステータスを読み出すには、[Enable DIP SW Session]ボタンをクリックします(図 4-7 参照)。

図 4-7. [Enable DIP SW Session]オプション



GUI は DIP スイッチのステータス読み出し要求を開始します。結果として、エンドポイント ボード上の DIP スイッチ ステータスが表示されます(図 4-8 参照)。エンドポイント ボード上の DIP スイッチ位置を変更し、GUI 内でそのステータスを観察します。

図 4-8. エンドポイント DIP スイッチのステータス



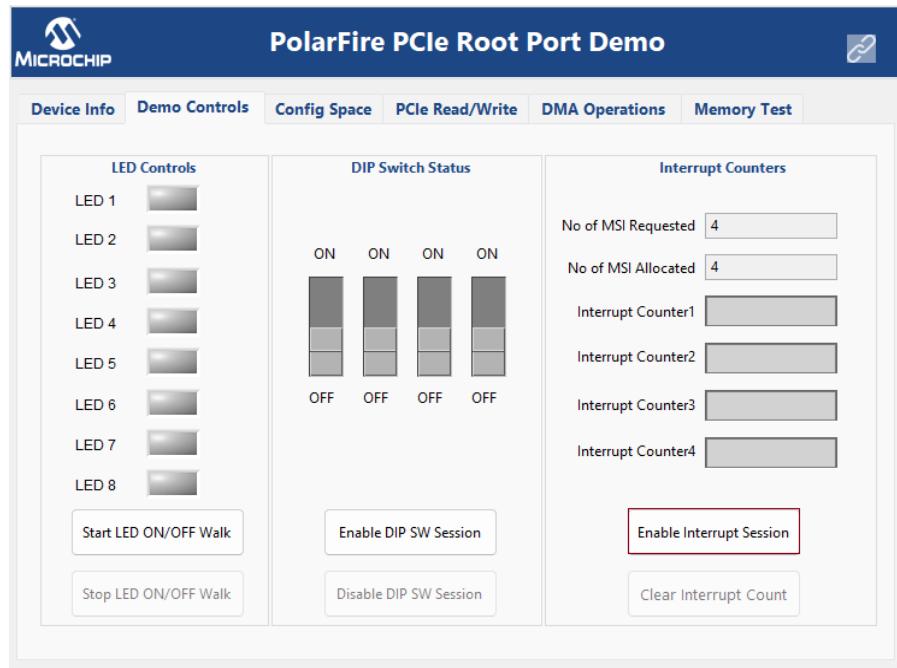
4.3.3. MSI カウント値の読み出し(質問する)

本デモでは、ルートポートはエンドポイント ボード上の押しボタン割り込み用に MSI カウント値を読み出す事ができます。

MSI カウント値を読み出すには以下の手順を実行します。

1. [Enable Interrupt Session]ボタンをクリックします(図 4-9 参照)。

図 4-9. [Enable Interrupt Session]オプション



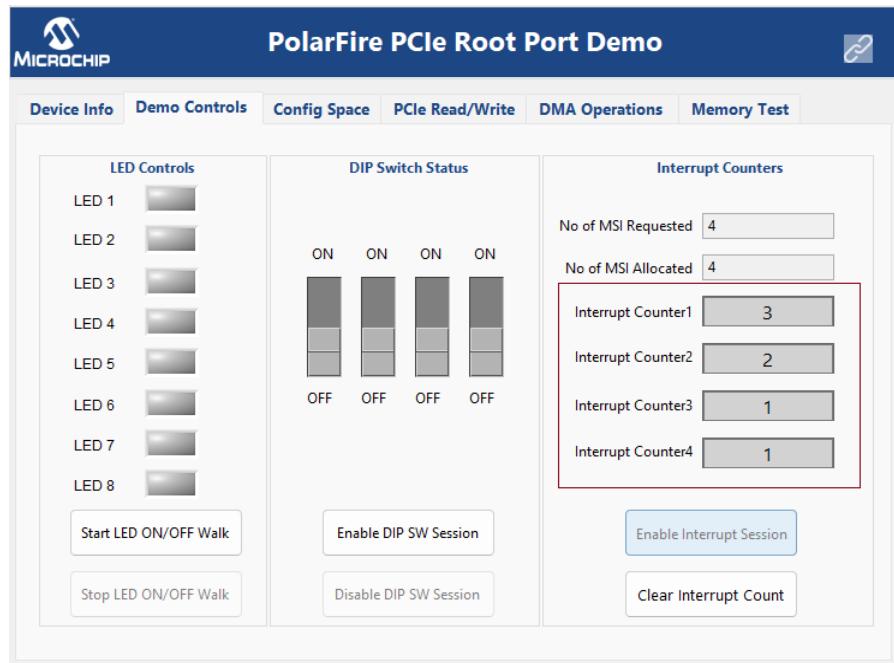
割り込みセッションが有効になると、GUI は RISC-V プロセッサに対して割り込みセッション イネーブル要求を送信します。PF_PCIE_0 は、エンドポイントによって要求された MSI の数を受信します。表 4-1 に、リファレンス デザイン内でルートポートによって割り当てられる MSI の 4 つのタイプを示します。

表 4-1. 割り当てられる MSI

MSI 番号	エンドポイント ボード上の割り込みタイプ	GUI 上で割り当てられる割り込みカウンタ
1	SW10	Interrupt Counter1
2	SW9	Interrupt Counter2
3	SW8	Interrupt Counter3
4	SW7	Interrupt Counter4

2. スイッチを押して割り込みカウンタを観察します。

図 4-10. Interrupt Counter4



3. [Clear Interrupt Count]ボタンを押すと、GUI 上の全ての割り込みカウンタがクリアされます。

4.3.4. BAR2 メモリ読み書きコマンドの実行(質問する)

本デモでは、ルートポートは BAR2 メモリ読み書きコマンドを発行する事により、エンドポイントの LSRAM/DDR3/DDR4 に対する読み書きを実行できます。

これらのコマンドを発行するため GUI 上の [PCIe Read/Write] タブを使います。エンドポイント LSRAM/ DDR3/DDR4 メモリが最初に読み出された後に、特定位置に値を入力する事で書き込みコマンドを開始できます。

BAR2 読み書きコマンドの実行手順は以下の通りです。

1. [BAR2-LSRAM]オプションを選択し、[Read]ボタンをクリックします(図 4-11 参照)。

図 4-11. BAR2-LSRAM の読み出し

	0x0	0x4	0x8	0xC
0x000				
0x010				
0x020				
0x030				
0x040				
0x050				
0x060				
0x070				
0x080				
0x090				
0x0A0				
0x0B0				
0x0C0				
0x0D0				
0x0E0				
0x0F0				
0x100				
0x110				
0x120				

Read Progress [progress bar] Read

2. 任意のメモリ位置を選択し、その位置の値を編集します。図 4-12 に例を示します。

図 4-12. BAR2-LSRAM の書き込み

	0x0	0x4	0x8	0xC
0x000	1	2	3	4
0x010	5	6	7	8
0x020	9	C	B	C
0x030	D	E	F	10
0x040	11	12	13	14
0x050	15	16	17	18
0x060	19	1A	1B	1C
0x070	1D	1E	1F	20
0x080	21	22	23	24
0x090	25	26	27	28
0x0A0	29	2A	2B	2C
0x0B0	2D	2E	2F	30
0x0C0	31	32	33	34
0x0D0	35	36	37	38
0x0E0	39	3A	3B	3C
0x0F0	3D	3E	3F	40
0x100	41	42	43	44
0x110	45	46	47	48
0x120	49	4A	4B	4C

Read Progress [progress bar] Read

3. 編集されたメモリ位置は緑に変化し、入力された値はエンドポイント LSRAM メモリ位置に書き込まれます(図 4-13 参照)。

図 4-13. BAR2-LSRAM の書き込みに成功

Address	Value
0x000	1
0x010	5
0x020	C
0x030	D
0x040	11
0x050	15
0x060	19
0x070	1D
0x080	21
0x090	25
0x0A0	29
0x0B0	2D
0x0C0	31
0x0D0	35
0x0E0	39
0x0F0	3D
0x100	41
0x110	45
0x120	49

4. 同様の手順でその他のメモリ位置も変更できます。
5. [Read]ボタンをクリックし、そのメモリ位置が直前の書き込み値を格納しているか確認します。
6. 同様の手順で BAR2-DDR3 および BAR2-DDR4 メモリの読み書きも実行できます。

4.4. データプレーン コマンドの実行(質問する)

本デモでは、ルートポートは下記のデータプレーン コマンドを実行するためにエンドポイント DMA エンジンの動作を開始します。

- DMA 動作の実行
- メモリテストの実行

4.4.1. DMA 動作の実行(質問する)

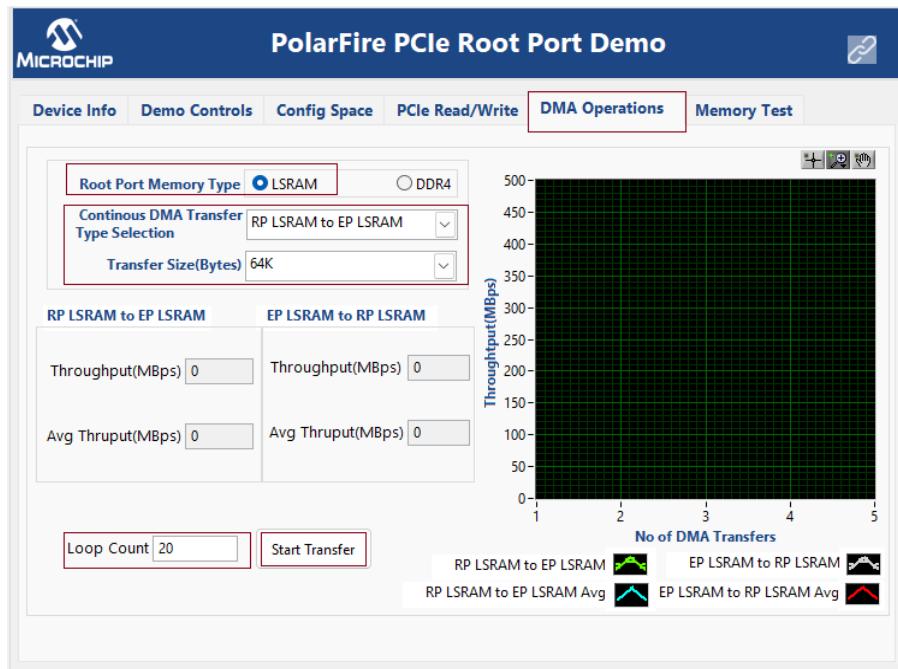
ルートポート(RP)が DMA 動作を開始した時に、Mi-V ソフトプロセッサは BAR0 を介してエンドポイント (EP) DMA レジスタをアクティブにします。エンドポイント DMA エンジンは下記の DMA 動作を実行できます。

- ルートポート LSRAM/DDR4 からエンドポイント LSRAM\DDR3\DDR4 への DMA 動作
- エンドポイント LSRAM\DDR3\DDR4 からルートポート LSRAM\DDR4 への DMA 動作

これらの DMA 動作は下記の手順により実行します。

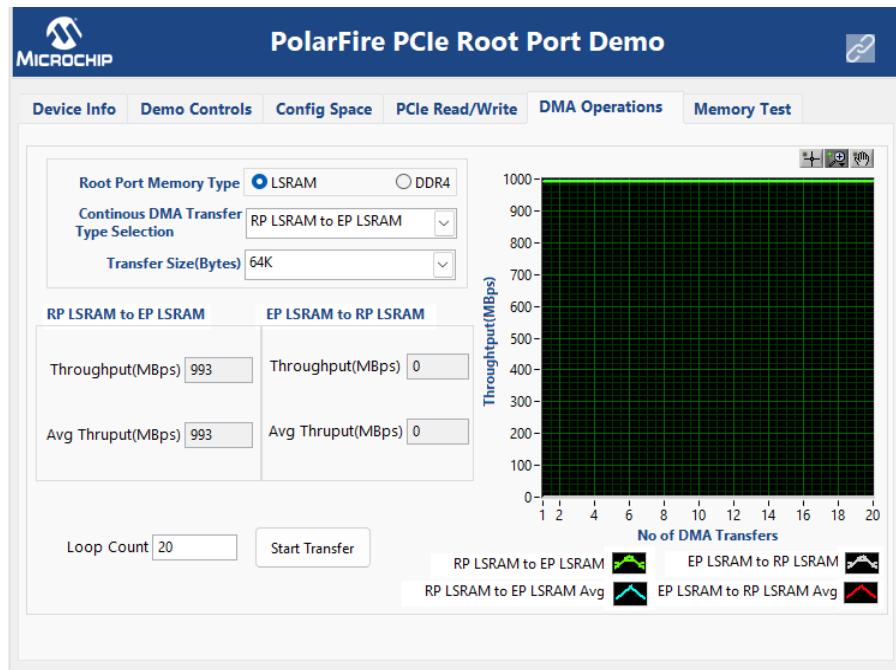
1. [DMA Operations]タブをクリックします(図 4-14 参照)。
2. 下記を実行します。
 - ドロップダウンリストから **RP LSRAM to EP LSRAM** を選択します。
 - [Transfer Size (Bytes)] ドロップダウンから **64K** を選択します。
 - [Loop Count] を 20 に設定します(「Transfer Size」と「Loop Count」は調整可能な一般パラメータです)。
 - [Start transfer] をクリックします。

図 4-14. RP LSRAM から EP LSRAM への DMA 動作の開始



GUI は対応するスループットの値とグラフを表示します(図 4-15 参照)。

図 4-15. RP LSRAM から EP LSRAM へのスループット



3. DMA トランザクションを開始するには以下を実行します。
 - ドロップダウンリストから **Both RP LSRAM <-> EP LSRAM** を選択します。
 - **[Transfer Size (Bytes)]** ドロップダウンから **64K** を選択します。
 - **[Loop Count]** を 20 に設定します(「Transfer Size」と「Loop Count」は調整可能な一般パラメータです)。

– [Start Transfer]をクリックします。

4. 同様に、ドロップダウンから **RP LSRAM to EP DDR3** と **RP LSRAM to EP DDR4** を選択し、スループットの値を確認します。
5. ルートポートのメモリタイプとして **DDR4** を選択し、エンドポイントの転送先メモリタイプを選択する事により DMA 動作を実行します。

4.4.2. メモリテストの実行(質問する)

[Memory Test]タブはメモリテスト機能を提供します。この機能も DMA 動作です。[Memory Test]タブは、ルートポートおよびエンドポイント メモリタイプ(LSRAM、DDR3、DDR4)間の DMA トランザクションを有効にします。この機能はデータパターン オプションを提供します。このオプションを使ってルートポート メモリが初期化され、その後に DMA 動作が実行されます。

メモリテストでは、ユーザ アプリケーションは下記の一連の動作を実行します。

1. 指定されたデータパターンを使ってルートポート メモリを初期化する
2. ルートポート メモリからエンドポイント メモリへの DMA 転送を実行する
3. ルートポート メモリ内のデータパターンを消去する
4. エンドポイント メモリからルートポート メモリへの DMA 転送を実行する
5. ルートポート メモリ内のデータと選択したデータパターンを比較する

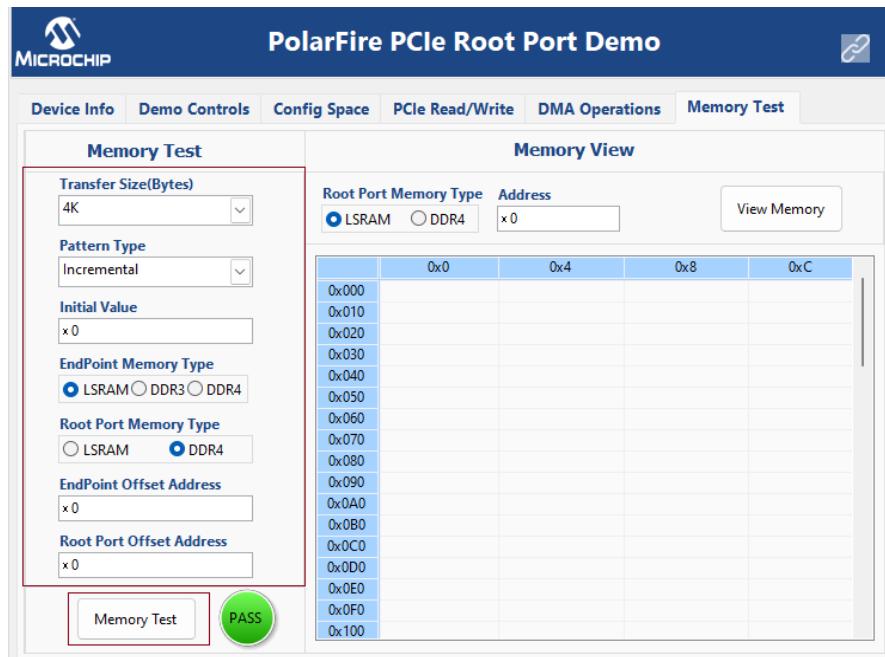
メモリテストを実行するには下記の手順を実行します。

1. GUI 上の各種 DMA パラメータ(Transfer Size (Bytes)、Pattern Type、Endpoint Memory Type、RootPort Memory Type、EndPoint Offset Address、RootPort Offset Address)を選択します(図 4-16 参照)



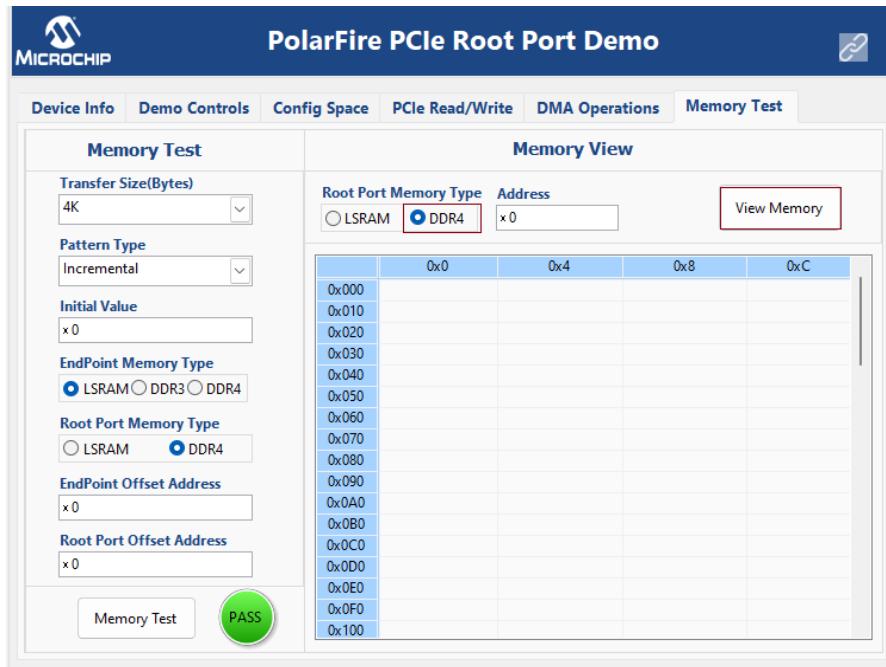
重要: ルートポート スレーブ ATR3 は 1 MB 向けに設定されます。従って、最大エンドポイント オフセット アドレスは F80000、最大ルートポート アドレスは 0x80000 です。

図 4-16. メモリテスト機能



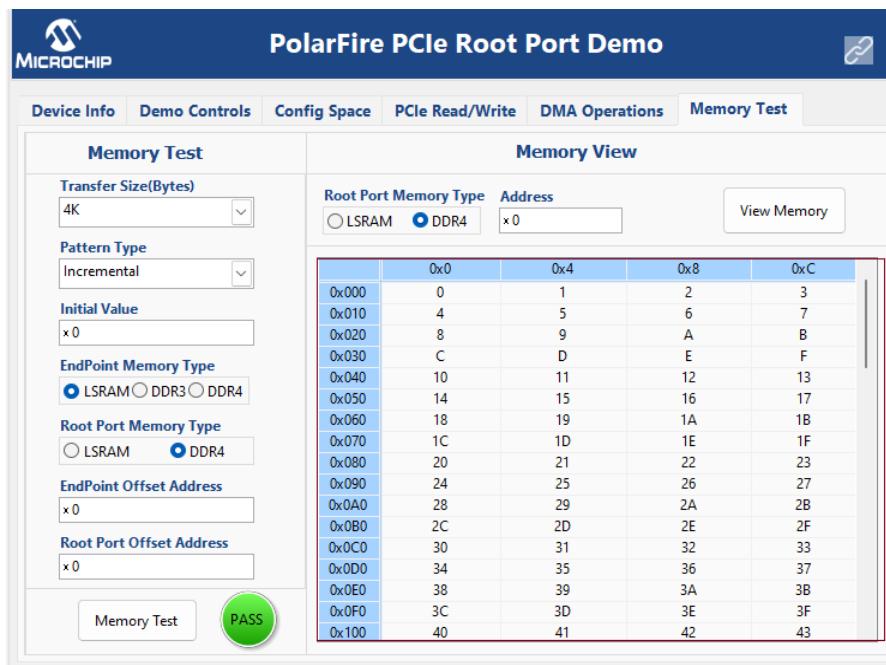
2. [Memory Test]をクリックします。
3. 「Root Port Memory Type」で DDR4 を選択し、[View Memory]をクリックすると、ルートポート DDR4 が読み出されます(図 4-17 参照)。

図 4-17. [View Memory]オプション



4. GUI にはルートポート DDR4 に書き込まれたデータパターンが表示されます(図 4-18 参照)。

図 4-18. ルートポート DDR4 メモリの内容



5. [PCIe Read/Write]タブを選択し、[Read]をクリックすると、エンドポイント LSRAM に書き込まれたデータパターンが表示されます。

図 4-19. エンドポイント LSRAM メモリの内容

PolarFire PCIe Root Port Demo				
Device Info	Demo Controls	Config Space	PCIe Read/Write	DMA Operations
<input checked="" type="radio"/> BAR2-LSRAM <input type="radio"/> BAR2-DDR3 <input type="radio"/> BAR2-DDR4	PCIe Offset Address x0			Note: All data is in hex format
	0x0	0x4	0x8	0xC
0x000	0	1	2	3
0x010	4	5	6	7
0x020	8	9	A	B
0x030	C	D	E	F
0x040	10	11	12	13
0x050	14	15	16	17
0x060	18	19	1A	1B
0x070	1C	1D	1E	1F
0x080	20	21	22	23
0x090	24	25	26	27
0x0A0	28	29	2A	2B
0x0B0	2C	2D	2E	2F
0x0C0	30	31	32	33
0x0D0	34	35	36	37
0x0E0	38	39	3A	3B
0x0F0	3C	3D	3E	3F
0x100	40	41	42	43
0x110	44	45	46	47
0x120	48	49	4A	4B

4.5. PolarFire DMA スループットのまとめ (質問する)

表 4-2 に、連続 DMA モード中に計測されたスループット値を示します。

表 4-2. スループットのまとめ

DMA 転送タイプ	DMA サイズ	スループット(MBps)	平均スループット(MBps)
RP LSRAM → EP LSRAM	512K	1022	1022
EP LSRAM → RP LSRAM	512K	779	779
RP LSRAM → EP DDR3	512K	773	773
EP DDR3 → RP LSRAM	512K	328	328
RP LSRAM → EP DDR4	512K	998	998
EP DDR4 → RP LSRAM	512K	391	391
RP DDR4 → EP LSRAM	512K	540	540
EP LSRAM → RP DDR4	512K	779	779
RP DDR4 → EP DDR3	512K	540	540
EP DDR3 → RP DDR4	512K	328	328
RP DDR4 → EP DDR4	512K	540	540
EP DDR4 → RP DDR4	512K	391	391

5. 補遺 1: FlashPro Express を使ってデバイスにプログラミングする方法

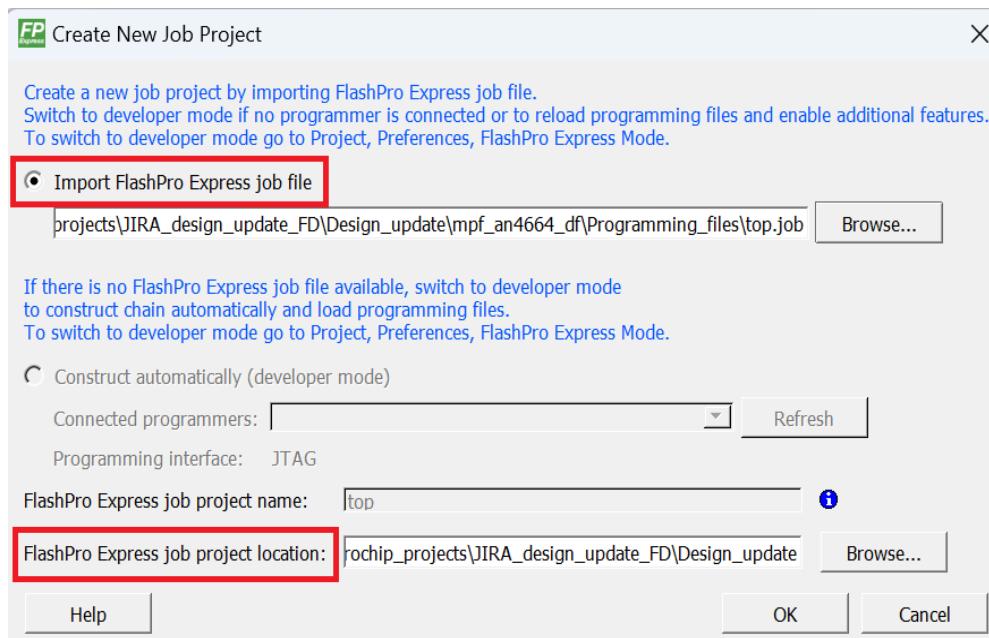
(質問する)

ルートポート デザインをボード A にプログラミングし、エンドポイント デザインをボード B にプログラミングする必要があります。

プログラミングの手順は以下の通りです。

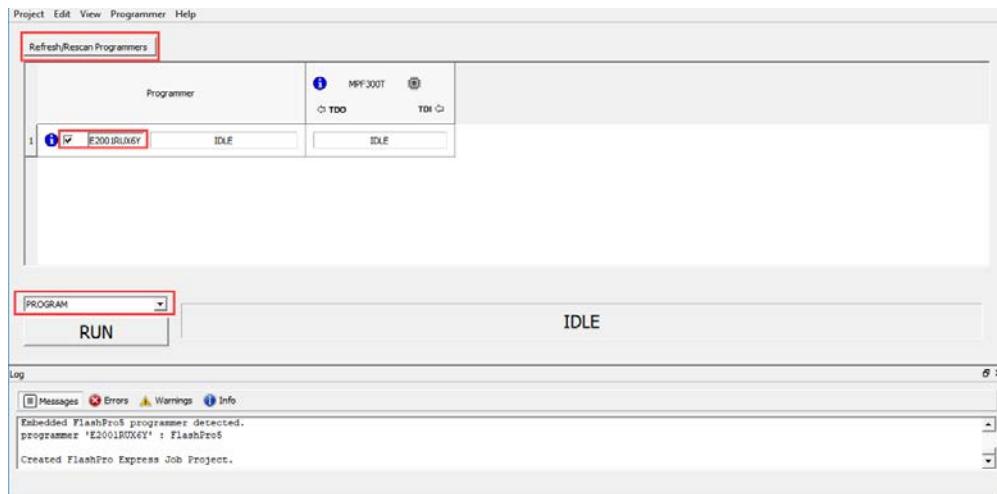
1. ボード A のジャンパが表 2-2 の通りに設定されている事を確認します。
2. 電源ケーブルをボード A の J9 コネクタに接続します。
3. ホスト PC からの USB ケーブルをボード A の J5 (FTDI ポート)に接続します。
4. SW3 スライドスイッチを使ってボード A に電源を投入します。
5. ホスト PC 上で **FlashPro Express** ソフトウェアを起動します。
6. 新規ジョブを作成するため、[New]をクリックするか、[Project]メニュー内で[New Job Project]を選択します。
7. 「New Job Project from FlashPro Express Job」ダイアログ ボックスで下記を指定します。
 - **Import FlashPro Express job file:**
[Browse]をクリックし、.job ファイルを見つけて選択します。このファイルの既定値パスは <ダウンロード先フォルダ>\mpf_an4664_df\Programming_files です。
 - **FlashPro Express job project location:**
[Browse]をクリックし、プロジェクトの保存位置を指定します。

図 5-1. FlashPro Express からの新規ジョブ プロジェクトの作成



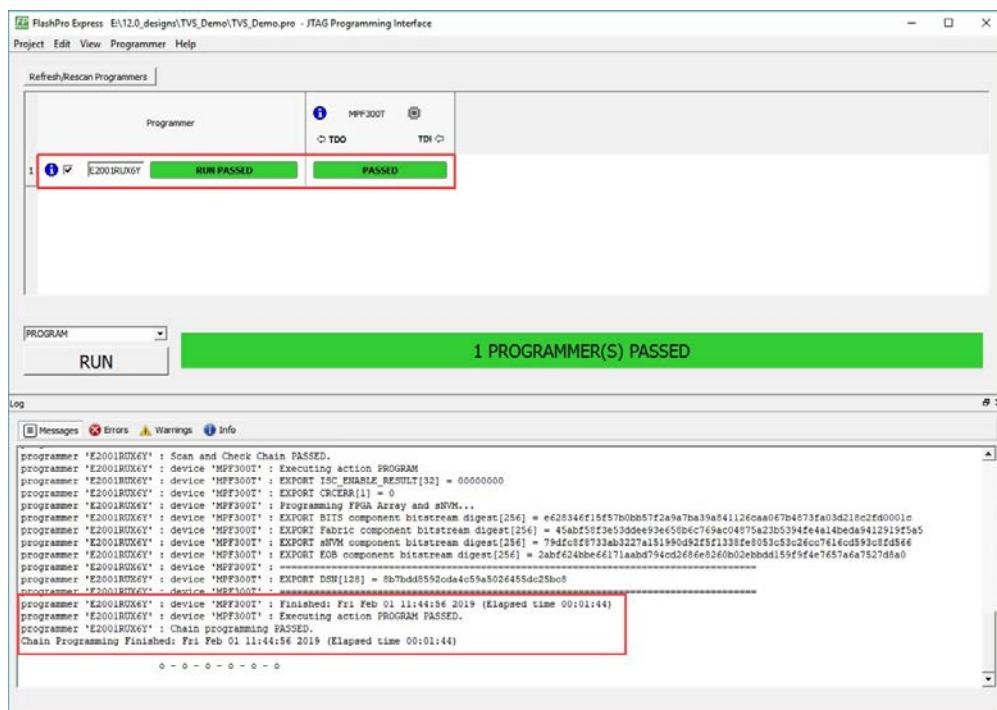
8. [OK]をクリックします。必要なプログラミング ファイルが選択され、デバイスへのプログラミング準備が整います。
9. 図 5-2 に示す「FlashPro Express」ウィンドウが開きます。[Programmer]フィールドにプログラマ番号が表示される事を確認します。表示されない場合、ボードの接続を確認し、[Refresh/Rescan Programmers]をクリックします。

図 5-2. デバイスのプログラミング



10. [RUN]をクリックします。デバイスが正常にプログラミングされると「RUN PASSED」ステータスが表示されます(図 5-3 参照)。PCIe ルートポート デモを実行する方法は「[4. デモの実行](#)」を参照してください。

図 5-3. FlashPro Express - RUN PASSED



11. FlashPro Express を閉じるか、[Project]タブ内で[Exit]をクリックします。以上により、ルートポート デザインがボード A にプログラミングされます。
12. 同様に、ボード B にエンドポイント デザインをプログラミングします。このプログラミング ファイルの既定値パスは `mpf_an4597_df\Programming_files\top_Eval_Kit.job` です。

6. 補遺 2: DDR4 の設定 (質問する)

DDR4 サブシステムは、AXI4 64 ビット インターフェイスを介して 32 ビット DDR4 メモリにアクセスするよう設定されます。DDR4 メモリの初期化およびタイミング パラメータは、PolarFire 評価用キット上の DDR4 メモリごとに設定されます。図 6-1 に、DDR4 メモリの一般的な設定を示します。

図 6-1. PF_DDR4 コンフィグレータ - General

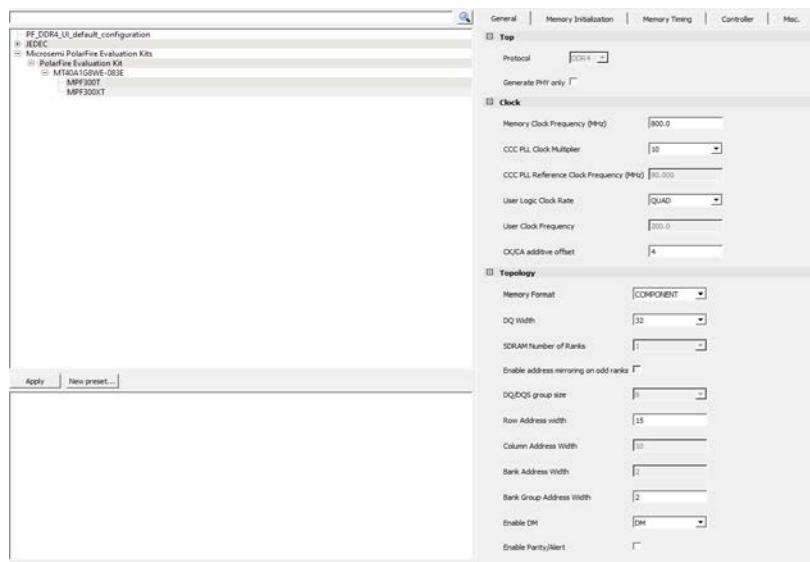


図 6-2 に、DDR4 メモリの初期化設定を示します。

図 6-2. PF_DDR4 コンフィグレータ - Memory Initialization

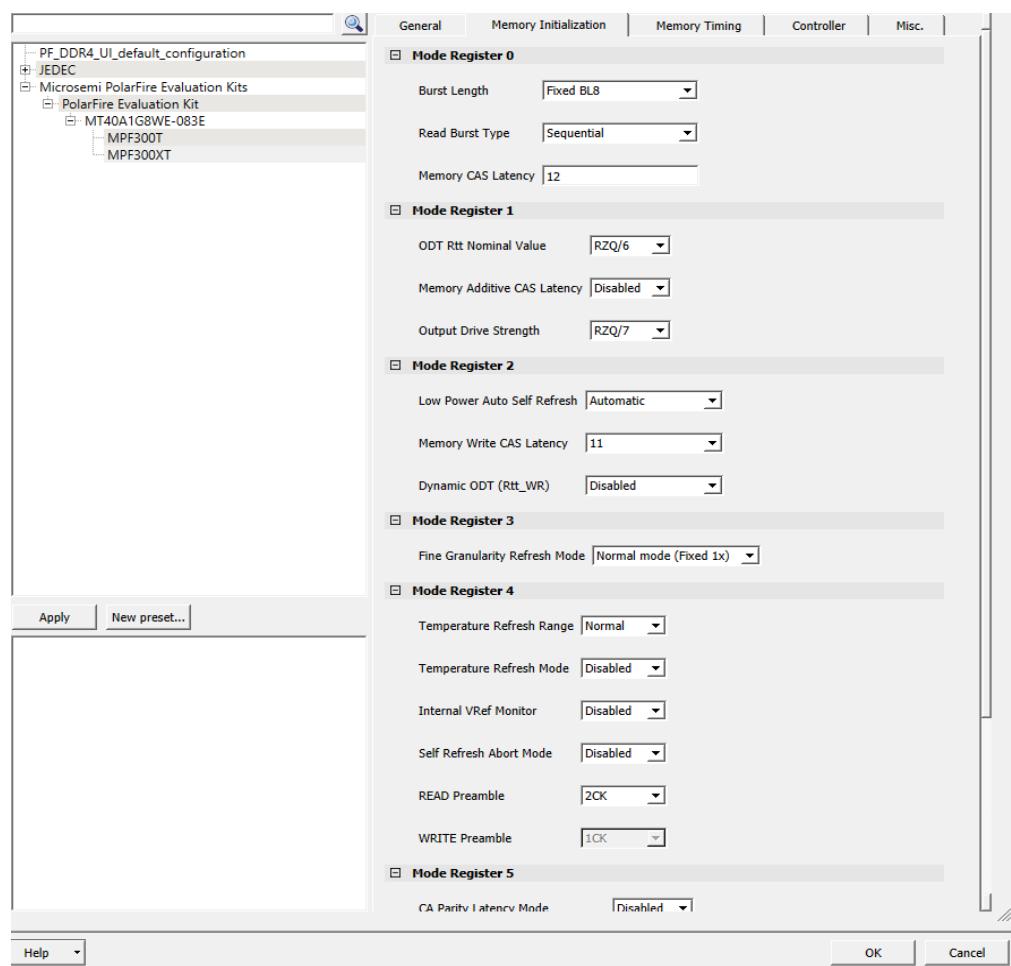


図 6-3 に、DDR4 メモリのタイミング設定を示します。

図 6-3. PF_DDR4 コンフィグレータ - Memory Timing

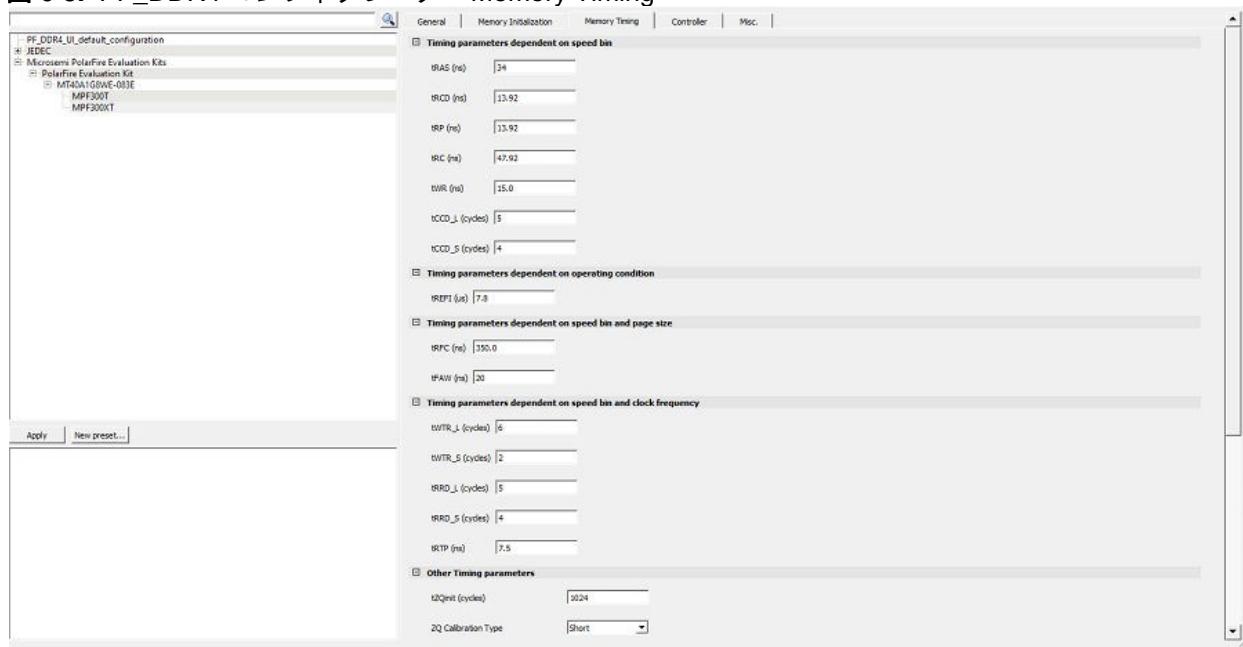
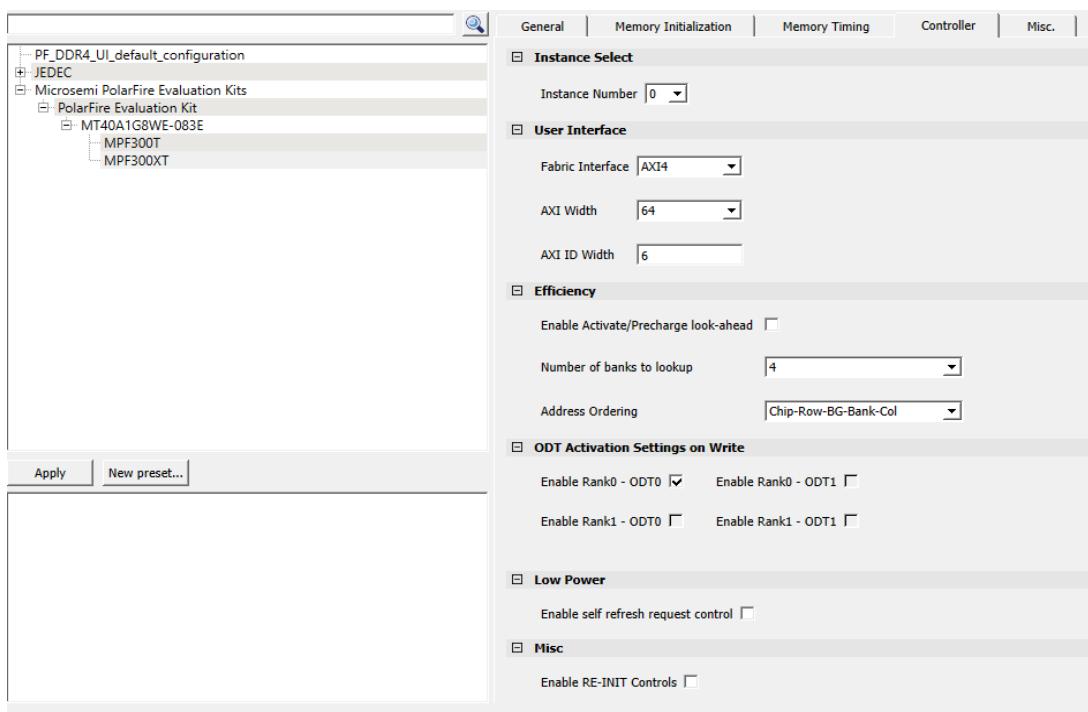


図 6-4 に、DDR4 メモリのコントローラ設定を示します。

図 6-4. PF_DDR4 コンフィグレータ - Controller



7. 補遺 3: TCL スクリプトの実行 (質問する)

TCL スクリプトはダウンロードしたデモ関連ファイルに含まれています(ディレクトリ「HW」内)。必要に応じ、.job ファイルの「Design Implementation till generation」からデザインフローを再生成できます。

TCL の実行手順は下記の通りです。

1. Libero ソフトウェアを起動します。
2. **Project > Execute Script....**と選択します。
3. **[Browse]**をクリックし、ダウンロードした「HW」ディレクトリから `script.tcl` を選択します。
4. **[Run]**をクリックします。

TCL スクリプトの実行に成功すると、Libero プロジェクトが「HW」ディレクトリ内に作成されます。

TCL スクリプトの詳細は `mpf_an4664_df\HW\TCL_Script_readme.txt` を参照してください。TCL スクリプトの実行中に生じた疑問点については、弊社の技術サポートまでお問い合わせください。

8. 補遺 4: 参考資料 (質問する)

以下に記載する参考資料は、本リファレンス デザインで使われる PCIe エンドポイントと IP コアに関する追加の情報を提供します。

- PolarFire トランシーバ ブロック、PF_TX_PLL、PF_XCVR_REF_CLK の詳細: [PolarFire Family Transceiver User Guide](#)
- PF_PCIE の詳細: [PolarFire Family PCI Express User Guide](#)
- PF_CCC の詳細: [PolarFire Family Clocking Resources User Guide](#)
- DDR3 メモリの詳細: [PolarFire Family Memory Controller User Guide](#)
- Libero、ModelSim、Synplify の詳細: ウェブページ [「Libero SoC Documentation」](#)
- PolarFire FPGA 評価用キットの詳細: [UG0747: PolarFire FPGA Evaluation Kit User Guide](#)
- CoreAHBLite の詳細: [CoreAHBLite Handbook](#)
この文書は Libero SoC カタログからダウンロードできます。
- CoreAHBtoAPB3 の詳細: [CoreAHBtoAPB3 Handbook](#)
この文書は Libero SoC カタログからダウンロードできます。

9. 改訂履歴 (質問する)

以下の改訂履歴には、本文書に加えられた変更内容を最新版から順にリビジョンごとに記載しています。

表 9-1. 改訂履歴

リビジョン	日付	変更内容
D	2025 年 9 月	<p>リビジョン D の主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> Libero® v2025.1 向けに文書を更新しました。 「補遺 1: FlashPro Express を使ってデバイスにプログラミングする方法」内の図 5-1 を更新しました。 「補遺 2: DDR4 の設定」内の図 6-2 と図 6-4 を更新しました。
C	2025 年 1 月	<p>リビジョン C の主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> Libero® v2024.2 向けに文書を更新しました。 文書全体を通して .job ファイルのパスと TCL スクリプトファイルのパスを更新しました。 「1.1. デモデザインの要件」に Microchip FPGA_GUI_Pack を追加しました。 「1.2. 前提条件」を更新しました。 「1.3.2. デモデザインの実装」内の図 1-2 を更新しました。 「2.2.1. リソース使用率」内の表 2-1 を更新しました。 「4. デモの実行」内の GUI の図を更新しました。 「補遺 2: DDR4 の設定」内の図 6-1 を更新しました。
B	2024 年 6 月	<p>リビジョン B の主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> 図 1-1 を更新しました。 表 1-2 を更新しました。 図 1-3 を更新しました。 図 1-4 を更新しました。 図 1-6 と図 1-7 を更新しました。 「補遺 1: FlashPro Express を使ってデバイスにプログラミングする方法」内の .job ファイルとプログラミング ファイルのパスを更新しました。 「補遺 3: TCL スクリプトの実行」内の TCL スクリプト ファイルのパスを更新しました。
A	2022 年 8 月	<p>リビジョン A の主な変更内容は以下の通りです。</p> <ul style="list-style-type: none"> 本書に Microchip 社のテンプレートを適用しました。 文書番号を 50200802 から DS00004664A へ変更しました。 文書 ID を DG0802 から AN4664 へ変更しました。 「1.2. 前提条件」に新しい行を追加しました。 図 1-1 を更新しました。 表 1-2 を更新しました。 図 1-2 を更新しました。 図 1-3 を更新しました。 図 1-4 を更新しました。 図 1-6 と図 1-7 を更新しました。 「4.1. GUI のインストール」内の GUI インストール リンクを置き換えました。 「補遺 1: FlashPro Express を使ってデバイスにプログラミングする方法」内の .job ファイルとプログラミング ファイルのパスを更新しました。 「補遺 3: TCL スクリプトの実行」内の TCL スクリプト ファイルのパスを更新しました。

表 9-1. 改訂履歴(続き)

リビジョン	日付	変更内容
7.0	—	<p>このリビジョンの主な変更内容は以下の通りです。</p> <ul style="list-style-type: none">文書を Libero SoC v2021.2 向けに更新しました。PERSTn 信号生成向けのサポートを追加しました。図 1-1~図 1-7 を差し替えました。表 1-2 と表 2-1 を更新しました。「1.3.2. デモデザインの実装」を更新しました。MiV サブシステムから AHBtoAXIAPB サブシステムを削除しました。MiV サブシステムに AXI_Interconnect_0 と Core_APB_0 を追加しました。MiV コア上の AXI イニシエータ インターフェイスと APB イニシエータ インターフェイスを有効にしました。リセット構造を更新しました。
6.0	—	<p>このリビジョンの主な変更内容は以下の通りです。</p> <ul style="list-style-type: none">Libero SoC v12.6 向けに文書を更新しました。Libero バージョン番号への参照を削除しました。
5.0	—	文書を Libero SoC v12.0 向けに更新しました。
4.0	—	文書を Libero SoC PolarFire v2.3 向けに更新しました。
3.0	—	文書を Libero SoC PolarFire v2.2 向けに更新しました。
2.0	—	文書を Libero SoC PolarFire v2.1 向けに更新しました。
1.0	—	本書の初版です。

Microchip 社の FPGA サポート

Microchip 社 FPGA 製品部門では、カスタマサービス、カスタマ技術サポートセンター、ウェブサイト、世界各地の営業所等、各種のサポートサービスで製品をバックアップしています。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。技術サポートケースを作成する際は、FPGA デバイスの製品番号を明記し、適切なケースカテゴリを選択し、デザインファイルをアップロードしてください。

製品の価格、製品のアップグレード、更新情報、注文状況、認証等、技術的でない製品のサポートについてはカスタマサービスにお問い合わせください。

- 北米のお客様は **800 262 1060** にお電話ください。
- その他の地域のお客様は **650 318 4460** にお電話ください。
- FAX 番号は世界共通で **650 318 8044** です。

Microchip 社の情報

商標

「Microchip」社の名称とロゴ、「M」のロゴ、およびその他の名称、ロゴ、ブランドは、米国および他の国における Microchip Technology Incorporated またはその関連会社および/または子会社の登録商標および未登録商標です（「Microchip 社の商標」）。「Microchip 社の商標」に関する情報は <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks> に記載されています。

ISBN: 979-8-3371-2215-1

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。その他のサポートについては、弊社または代理店にお問い合わせになる前に、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、非侵害性、商品性、特定目的への適合性の暗黙的保証、または状態、品質、性能に関する保証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、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する Microchip 社に直接支払った額を超えません。法律で認められる最大限の範囲を適用しようと、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する 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。

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。