

はじめに

MachXO2™は SRAM ベースのプログラマブル・ロジックデバイスですが、内部フラッシュメモリが内蔵されているため、MachXO2 は不揮発性デバイスと同様な動作が可能です。MachXO2 は、FPGA のプログラミング及びコンフィグレーションのための豊富な機能セットを提供します。数多くのオプションがあり、実際のニーズに適したプログラミング・ソリューションの構築が可能です。本テクニカルノートでは、ニーズに合わせてプログラミング及びコンフィグレーション・ソリューションを組み合わせられるよう、それぞれのオプションについて詳述します。

MachXO2 の機能

MachXO2 デバイスの主要なプログラミング及びコンフィグレーション機能には、次のものがあります。

- ・ 内部フラッシュメモリからのインスタントオン・コンフィグレーション ～ ミリ秒単位で起動
- ・ セキュアなシングルチップ・ソリューション
- ・ 複数のプログラミング及びコンフィグレーション・インターフェイス
 - 1149.1 JTAG
 - セルフダウンロード
 - スレーブ SPI (以下 "SSPI")
 - マスタ SPI (以下 "MSPI")
 - デュアルブート
 - I²C
 - WISHBONE バス
- ・ ユーザフラッシュメモリ (UFM) による不揮発性データストレージ。用途例は以下
 - コンフィグレーション・フラッシュメモリ (CFM) のオーバフロー
 - EBR 初期化データ
 - アプリケーション固有のデータ
- ・ 不揮発性メモリの透過的 (transparent) プログラミング
- ・ 外付け SPI メモリによるオプションのデュアルブート
- ・ デザイン保護のためのオプションのセキュリティビット

用語定義

本書では、共通機能の説明で次の用語を使用します。

- ・ **BIT** ～ BIT ファイルは、外付け SPI フラッシュメモリに格納される、MachXO2 のコンフィグレーション・データです。バイナリファイルになっており、このまま SPI フラッシュにプログラムされます。
- ・ **コンフィグレーション** ～ MachXO2 の SRAM メモリセルの状態を変更することです。
- ・ **コンフィグレーション・データ** ～ 不揮発性メモリから読み出され、FPGA のコンフィグレーション SRAM にロードされるデータです。ビットストリーム、またはデバイス・ビットストリームとも呼ばれます。
- ・ **コンフィグレーション・モード** ～ MachXO2 が不揮発性メモリからコンフィグレーション・データを取得する方法を定義します。
- ・ **内部フラッシュメモリ** ～ JED またはビットファイルを直接内部フラッシュ・セクタにプログラムすることができます。ユーザはこのコンフィグレーションデータが実際にはどの先頭ページからなのかを知ることができます。

る必要はありません。MachXO2 内のコンフィグレーション・エンジンがフラッシュメモリから SRAM への転送処理を行います。

- ・ **JEDEC** ~ JEDEC ファイルには、MachXO2 の UFM、UFM、フィーチャ行、及びフィーチャビットにプログラムされるコンフィグレーション・データが格納されます。フォーマットについては、このテクニカルノートで後述します。
- ・ **オフライン (Offline) モード** ~ 不揮発性メモリのプログラミングとコンフィグレーション SRAM のコンフィグレーションの両方に該当する用語です。オフラインモード・プログラミング / コンフィグレーションを使用すると、不揮発性またはコンフィグレーション SRAM の内容は更新されますが、MachXO2 はオフラインモードのプログラミング / コンフィグレーションが完了するまで、ユーザモードで論理動作を実行しません。
- ・ **数値フォーマット** ~ 基数の表現には、次の表記法を使用します。
 - **0x** : 先頭に '0x' が付く数値は 16 進数
 - **b** : 末尾に 'b' が付く数値は 2 進数
 - その他の全ての数値は 10 進数
- ・ **ポート** ~ プログラミング及び一部のコンフィグレーション動作に使用される物理接続のことです。MachXO2 のポートには、JTAG、SPI、I²C、WISHBONE などの物理接続があります。
- ・ **プログラミング** ~ 内蔵または外付け不揮発性メモリの内容変更に使用されるプロセスです。
- ・ **透過 (Transparent) モード** ~ MachXO2 をユーザモードにしたまま、CFM 及び UFM を更新するために使用されます。
- ・ **ユーザモード** ~ 設計された論理機能で MachXO2 は動作すること。コンフィグレーションが完了すると MachXO2 はユーザモードになります。

コンフィグレーションの詳細

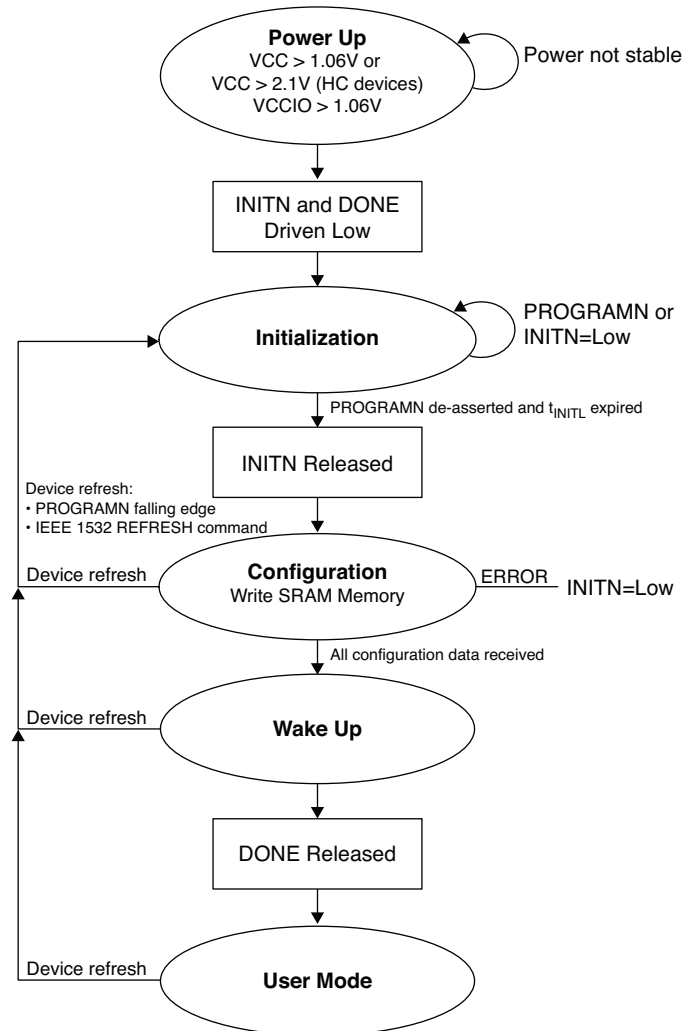
MachXO2 デバイスにはコンフィグレーション用に 2 種類のメモリ (SRAM 及びフラッシュ) が内蔵されています。SRAM には、MachXO2 の動作を定義する "ヒューズ" となる、アクティブ・コンフィグレーションが格納されます。多くの場合、アクティブ・コンフィグレーションは不揮発性メモリから取得されます。不揮発性メモリにはコンフィグレーション・データが格納され、それが MachXO2 の SRAM にロードされます。MachXO2 には SRAM にロードされるコンフィグレーション・データを格納する、フラッシュメモリが内蔵されています。

コンフィグレーションのプロセスとフロー

MachXO2 は初期化、コンフィグレーション、ウェイクアップといった一連のステートを経て、動作可能になります。

MachXO2 の sysCONFIG ポートは、プログラミングとコンフィグレーション用に業界標準の通信プロトコルを提供します。表 14-1 に示す各プロトコルは、MachXO2 デバイスの内部フラッシュメモリにアクセスする手段、またはコンフィグレーション SRAM にロードするための手段を提供します。各 sysCONFIG ポートの機能については、「メモリ空間のアクセス」の項で説明します。

図 14-1. コンフィグレーション・フロー



フラッシュメモリにアクセス可能な sysCONFIG ポートを表 14-1 に示しますが、これには優先順位があります。MSPI コンフィグレーション・ポートにはフラッシュメモリ空間を変更する機能がないため、sysCONFIG ポートの優先順位スキームには含まれません。優先順位の低い sysCONFIG ポートを使用したコンフィグレーション・ロジックの動作は、優先順位の高い sysCONFIG ポートによって割り込まれることがあるため、この優先順位スキームを理解することは重要です。優先順位が低い sysCONFIG ポートが、優先順位の高い sysCONFIG ポートから割り込まれた場合の、コンフィグレーション・ロジックの動作は（デバイスとして）定義されていません。sysCONFIG ポートを使用したコンフィグレーション・ロジックへの同時アクセスを許容しないでください。

電源起動シーケンス

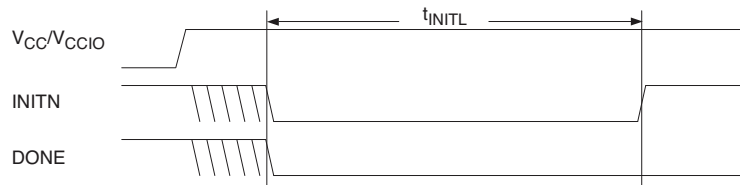
MachXO2 が動作するには、デバイスに電源を供給する必要があります。システムに供給される電圧が上昇すると短時間の間、MachXO2 は未定義のステートになります。

電源がさらに上昇すると、MachXO2 内部の POR (Power On Reset) 回路がアクティブになり、外部 I/O ピンをハイインピーダンス状態にし、また V_{CC} 及び V_{CCIO0} 入力電位を監視します。POR 回路は、次の条件になるまで待ちます。

- ・ $V_{CC} > 1.06V$ (= ZE デバイス。HC デバイスでは 2.1V)
- ・ $V_{CCIO0} > 1.06V$

これらの条件が満たされると、POR 回路は内部リセットストローブを解放し、デバイスが初期化プロセスを開始できるようにします。即ち MachXO2 は INITN をアクティブ Low に、そして DONE を Low にドライブします。その後、図 14-1 に示すようにデバイスは初期化ステートに移行します。

図 14-2. POR (パワーオンリセット) タイミングからのコンフィグレーションの開始



初期化

MachXO2 は POR 回路が INITN 及び DONE ステータスピンを Low にドライブすると、すぐにメモリ初期化ステートに移行し、MachXO2 内部のコンフィグレーション SRAM を全てクリアします。

次の全ての条件が満たされるまで、MachXO2 は初期化ステートのままとなります。

- ・ t_{INITL} 期間が経過した
- ・ PROGRAMN ピンが High にネゲートされた
- ・ INITN ピンが High (外部マスタから Low にドライブされない)

専用 INITN ピンは、初期化中に 2 つの機能を提供します。1 つは、MachXO2 が現在コンフィグレーション SRAM をクリア中であることを示します。もう 1 つは、入力として機能し、Low の場合は初期化ステートからコンフィグレーション・ステートへの移行を防止します。

t_{INITL} 期間中、MachXO2 はコンフィグレーション SRAM をクリアします。MachXO2 がデバイスチェーンに含まれている場合、各デバイスが持つ t_{INITL} 初期化時間はそれぞれ異なります。最も遅い t_{INITL} パラメータを持つデバイスによって、チェーン内の他のデバイスはコンフィグレーションの開始を待たされます。複数デバイスのチェーンで INITN の解放が早すぎると、チェーン内の一部のデバイスがコンフィグレーションにときどき失敗することがあります。

初期化信号 INITN はアクティブ Low、オープンドレインで、初期化完了時に High になるように外付け抵抗でプルアップする必要があります。複数の FPGA のコンフィグレーションを同期するには、1 つまたは複数の INITN ピンをワイアード AND 接続する必要があります。1 つまたは複数の FPGA、あるいは外部デバイスが INITN を Low に保った場合、MachXO2 は初期化ステートのままとなります。

コンフィグレーション

INITN ピンの立ち上がりエッジで、MachXO2 はコンフィグレーション・ステートに移行し、開発ツールで作成されたビットストリームを受け付けます。

MachXO2 は、不揮発性メモリからのコンフィグレーション・データのフェッチを開始します。MachXO2 のコンフィグレーションに使用されるメモリは、内部フラッシュか外付け SPI フラッシュのどちらかです。有効なコンフィグレーション・データが格納されたメモリがない場合、MachXO2 はコンフィグレーション・ステートのままとなります。MachXO2 内部または接続された不揮発性メモリをプログラムするか、JTAG ポートを使用して MachXO2 をコンフィグレーションする必要があります。デバイスがブランク時か消去状態の場合にプログラム可能なモードは JTAG と SSPI、および I²C のみです。

MachXO2 がコンフィグレーション・データを受信中に、INITN 制御ピンはその最後の機能を果たし、コンフィグレーション・データ内にエラーが存在することを示すために使用されます。INITN がプルアップによ

り High のままであれば、コンフィグレーションは問題なく続行されています。Low にドライブされた場合は何らかのエラーが発生しており、MachXO2 が動作しなくなります。

ウェイクアップ

ウェイクアップとは、コンフィグレーション・モードからユーザモードに移行することです。MachXO2 の固定された 4 段階のウェイクアップ・シーケンスは、デバイスがコンフィグレーション・データを全て受信すると開始されます。全てのコンフィグレーション・データを受信すると、内部 DONE ステータスビットをアサートし、ウェイクアップ・ステートマシンが 4 本の信号の順序制御を行います。この 4 つの制御ストロブには、次のものがあります。

- ・ 外部 DONE ピン
- ・ 内部グローバル・ライトディセーブル (Global Write DISable: GWDISn)
- ・ 内部グローバル出力イネーブル (Global Output Enable: GOE)
- ・ 内部グローバル・セット / リセット (Global Set / Reset: GSR)

ウェイクアップ・プロセスの最初のステップでは、MachXO2 が GOE を解放します。GOE がアサートされると、MachXO2 の I/O がハイインピーダンス・ステートを終了し、プログラムされた出力機能を果たすようになります。MachXO2 の入力には常にアクティブになっています。GSR がアサートされていると、入力信号が MachXO2 のフリップフロップに対してアクションを実行できません。

ウェイクアップ・プロセスの第 2 ステップでは、GSR 及び GWDISn の制御が解放されます。

GSR は内部ストロブで、アサートされると **GSR イネーブル** 属性を持つ全ての I/O フリップフロップ、ファブリック・フリップフロップ、分散 RAM 出力フリップフロップ、及びブロック RAM 出力フリップフロップが、そのハードウェア記述言語の定義に従ってセット / クリアされます。

GWDISn は、MachXO2 内部の全ての RAM ロジックのライトイネーブル・ストロブをオーバーライド（無効に）する制御です。上述のように入力は常にアクティブになっています。GWDISn をアサートしたままであれば、インスタンスされた RAM リソースの偶発的な書き換えを防止できます。

ウェイクアップ・プロセスの最後のステップは、外部 DONE ピンのアサートです。外部 DONE ピンはイネーブルされたときに限り、双方向オープンドレイン I/O になります。外部デバイスが外部 DONE ピンを Low に保持すると、MachXO2 のウェイクアップ・プロセスが続行されなくなります。外部 DONE がイネーブルされ、High になった場合のみ、ウェイクアップの最終ステップが完了します。外部 DONE ピンがイネーブルされていない場合は、ウェイクアップが中断なく完了します。

ウェイクアップの最終ステップが完了すると、MachXO2 はユーザモードになります。

ユーザモード

MachXO2 はウェイクアップ・シーケンスが完了すると直ちにユーザモードになり、ユーザ論理の動作を開始します。次のいずれかのイベントが発生するまで、MachXO2 はこのステートのままとなります。

- ・ PROGRAMN 入力ピンが Low にドライブされる
- ・ いずれかのコンフィグレーション・ポートから REFRESH コマンドを受信する
- ・ 電源が再投入される（一旦電源オフ後に再度オンする）

コンフィグレーション・メモリのクリアと再初期化

MachXO2 のユーザモード機能動作は、意図的にクリアされるか電源がオフされるまで継続されます。MachXO2 の内部コンフィグレーション SRAM をクリアする方法はいくつかあります。1 つは、電源をオフしてから再び投入すること、もう 1 つの方法は、PROGRAMN ピンをトグルすることです。また Refresh コマンドでメモリを再初期化する方法もあります。Refresh コマンドは、アクティブな任意のコンフィグレーション・ポートを使用して与えられます。

- ・ PROGRAMn 入力をドライブする
- ・ MachXO2 の電源を再投入する
- ・ コンフィグレーション・ポートを使用して Refresh コマンドを送信する

これらの方法のいずれかを実行すると、MachXO2 は INITN 及び DONE を Low にドライブし、初期化ステートに入ります。

メモリ空間のアクセス

MachXO2 のフラッシュと SRAM という 2 つの内部メモリは、読み出しと書き込みが可能です。MachXO2 の各ポートは、各メモリ空間への異なるアクセスレベルを持っています。表 14-1 に MachXO2 のポートと、それがアクセス可能なメモリ空間のクロスリファレンスを示します。

表 14-1 に示すとおり、JTAG ポートは両方の内部メモリ空間の読み出しと書き込みが可能です。これ以外にコンフィグレーション SRAM を読み出すことができるポートはありません。JTAG ポートは、オフラインモードと透過モードのどちらでも 2 つのメモリ空間にアクセスできます。それ以外のポートは、実行できる機能が制限されています。

表 14-1. 各種ポートのメモリ空間アクセス

ポート	オンチップフラッシュメモリ		SRAM	
	Read	Write	Read	Write
JTAG	Yes	Yes	Yes	Yes
SPI ポート	Yes	Yes	No	Refresh ²
I ² C ポート	Yes	Yes	No	Refresh ²
内部 WISHBONE	Yes ¹	Yes ¹	No	No

1. 透過 (Transparent) モードの場合のみ

2. 「コンフィグレーション・メモリのクリアと再初期化」項を参照してください

オンチップ・フラッシュメモリのプログラミング

表 14-1 に示す通り、オンチップ・フラッシュメモリは異なるモードでプログラミングできます。これらにモードについては次章以降で説明します。それぞれにはオフラインモードとバックグラウンドモードの異なる方法があります。

オフライン・プログラミング：この方法ではデバイスがプログラミングモードに入る必要があります。プログラミングモード中、プログラミングが終了するまでデバイスは動作を停止します。Diamond プログラマを使用する場合でオペレーションとして "FLASH" で始まる名称のオペレーションを指定すると、本モードが選択されます。オペレーションについて特に言及しない場合、フラッシュメモリとしてアクセスされるセクタはフィーチャ行 (Feature Row) とコンフィグレーション・フラッシュ、および UFM です。

バックグラウンド・プログラミング：この方法では、コンフィグレーション回路がオンチップ・フラッシュメモリにアクセスしている最中でも、デバイスはユーザモードの動作を継続します。プログラミングが完了すると、REFRESH 命令によってプログラムされたデータが SRAM に展開されます。Diamond プログラマを使用する場合でオペレーションとして "XFLASH" で始まる名称のオペレーションを指定すると、本モードが選択されます。オペレーションについて特に言及しない場合、フラッシュメモリとしてアクセスされるセクタはフィーチャ行 (Feature Row) とコンフィグレーション・フラッシュ、および UFM です。

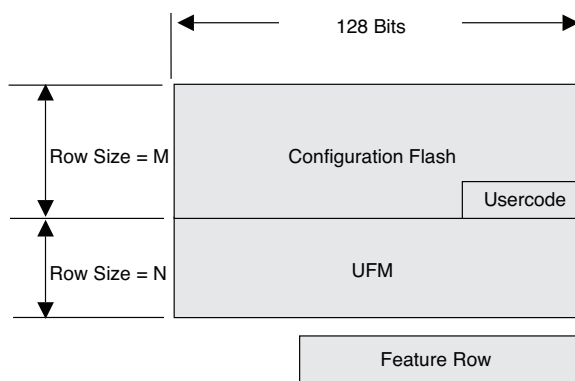
PLL を二個集積しているデバイス (XO2-2000U、4000、7000) でバックグラウンド・プログラミングを実装する場合は留意が必要です。デバイス内右側にある RPLL と呼んでいる PLL は、プログラミング中にリセット状態になければなりません。必要となるプログラミング時間については TN1246 (Using User Flash Memory and Hardened Control Functions in MachXO2 Devices) の図 17-97 をご参照ください。左側にある LPLL はその必要はなく、ユーザ動作が可能です。

ビットストリームとフラッシュメモリのサイズ

MachXO2 は SRAM ベースの FPGA です。コンフィグレーション SRAM には、全てのコンフィグレーション・データを格納可能な不揮発性メモリからロードする必要があります。コンフィグレーション・データのサイズは一定していません。MachXO2 で使用可能なロジックの量と、コンフィグレーション時に初期化される EBR (Embedded Block RAM、組み込みブロック RAM) コンポーネント数に依存します。最大量のストレージが必要になるのは、最大規模のデバイスを使用し、各 EBR がそれぞれ固有のデータ値で初期化され、圧縮をオンにしないで生成された MachXO2 デザインです。

MachXO2 の内部フラッシュメモリにコンフィグレーション・データを格納する際は、特別な注意が必要です。MachXO2 のフラッシュメモリは 3 セクタで構成されています。最初のセクタは圧縮コンフィグレーション・データの保持専用で使用され、コンフィグレーション・フラッシュメモリ (以下 CFM) と呼ばれます。2 番目のセクタはユーザフラッシュメモリ (以下 UFM) と呼ばれ、3 種類の機能を提供します。コンフィグレーション・データイメージが大きい場合に、CFM を補うストレージとなり、或いは EBR の内容を格納でき、また汎用フラッシュメモリとして使用できます。3 番目のセクタはフィーチャ行です。

図 14-3. MachXO2 デバイスのフラッシュメモリ空間



殆どのデザインでは、CFM はコンフィグレーション SRAM にロードされる、圧縮したコンフィグレーション・データの格納に十分な大きさがあります。しかしデザイン内のロジックサイズと初期化される EBR 数が増加すると、コンフィグレーション・データのサイズも増加し、コンフィグレーション・データが UFM セクタにオーバーフローすることがあります。また、可能性は低いものの、内部フラッシュメモリ全体に収まらないほどコンフィグレーション・データが大きくなることもあります。大きくなりすぎて、CFM と UFM メモリ空間を組み合わせても収まらなくなった場合は、デザインを変更してサイズを小さくするか、外付け SPI フラッシュメモリを使用する必要があります。コンフィグレーション・データを生成するソフトウェアの入力を指定することで、UFM へのオーバーフローを防止できます。

表 14-2. 最大コンフィグレーション・ビット数

デバイス	EBR なしの非圧縮ビット ストリーム・サイズ	EBR ありの非圧縮ビット ストリーム・サイズ	最大内部フラッシュ サイズ	単位
LCMXO2-256	0.09	N/A	0.071	Mb
LCMXO2-256 HC	0.09	N/A	0.071	Mb
LCMXO2-256 ZE	0.09	N/A	0.071	Mb
LCMXO2-640	0.19	0.20	0.17	Mb
LCMXO2-640HC	0.19	0.20	0.17	Mb
LCMXO2-640 ZE	0.19	0.20	0.17	Mb
LCMXO2-640UHC	0.19	0.20	0.17	Mb
LCMXO2-1200	0.35	0.41	0.33	Mb
LCMXO2-1200HC	0.35	0.41	0.33	Mb
LCMXO2-1200ZE	0.35	0.41	0.33	Mb
LCMXO2-1200UHC	0.35	0.41	0.33	Mb
LCMXO2-2000	0.51	0.58	0.47	Mb
LCMXO2-2000HC	0.51	0.58	0.47	Mb
LCMXO2-2000HE	0.51	0.58	0.47	Mb
LCMXO2-2000ZE	0.51	0.58	0.47	Mb
LCMXO2-2000U	0.51	0.58	0.47	Mb
LCMXO2-2000UHC	0.51	0.58	0.47	Mb
LCMXO2-2000UHE	0.51	0.58	0.47	Mb
LCMXO2-4000	0.93	1.02	0.80	Mb
LCMXO2-4000 HE	0.93	1.02	0.80	Mb
LCMXO2-4000 HC	0.93	1.02	0.80	Mb
LCMXO2-4000 ZE	0.93	1.02	0.80	Mb
LCMXO2-7000	1.47	1.70	1.38	Mb
LCMXO2-7000 HE	1.47	1.70	1.38	Mb
LCMXO2-7000 HC	1.47	1.70	1.38	Mb
LCMXO2-7000 ZE	1.47	1.70	1.38	Mb

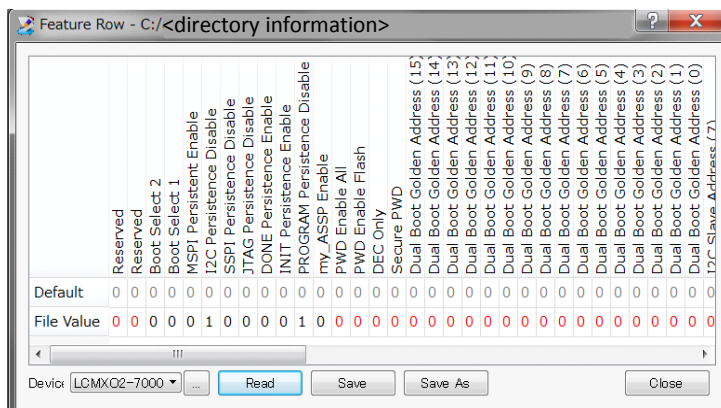
CFM と UFM メモリを合わせたサイズよりも、コンフィグレーション・データが大きくなった場合は、デバイスのビットストリームを外付け SPI フラッシュに格納する方法があります。表 14-2 に、SPI フラッシュを選択するための、最大非圧縮ビットストリーム・サイズを示します。

フィーチャ行 (Feature Row)

MachXO2 に内蔵されるフィーチャ行は、FPGA リソースの制御に使用されます。例えば、フィーチャ行は MachXO2 のコンフィグレーション SRAM をロードする方法を定義するために使用されます。他の FPGA では、この動作制御に外部 I/O ピンが使用されます。フィーチャ行はコンフィグレーション機能の選択や、デバイスの I/O 数を増やしてハードウェアに変更を加える必要性を排除するなどの柔軟性を与えます。

フィーチャ行の消去やプログラミングは独立に実行できます。消去されると、フィーチャ行の値は HW (ハードウェア) デフォルトモード・ステートに戻ります。フィーチャ行の編集は、デタッチしたプログラマーで **Design→Utilities→Programming File Utility** を順に選択してユーティリティツールを立ち下げ、その後 **Tools→Feature Row Editor** を指定して行います。

図 14-4. フィーチャ行エディタの表示例



フィーチャ行オプションと、Diamond スプレッドシートビューに関して表 14-3 と表 14-4 に示します。

表 14-3. フィーチャ行オプションと Diamond スプレッドシートビュー

MASTER_SPI_PORT	CONFIGURATION	BOOT_SEL[2:1], MSPI_Persistent_Enable
ENABLE	CFG	001
ENABLE	EXTERNAL	011
EFB_USER	CFG *1	000
DISABLE	CFG *1	000

1: "CFG" には CFG、CFG_EBRUFM、または CFGUFM を含む

表 14-4. MachXO2 のフィーチャ行要素

フィーチャ	SW デフォルトモード・ステート (プログラム済み)	HW デフォルトモード・ステート (消去)
PROGRAMN Persistence	Disabled	Enabled
INITn Persistence	Disabled	Disabled
DONE Persistence	Disabled	Disabled
Custom IDCODE	0x00000000	0x00000000
TraceID™	00000000	00000000
Security ¹	OFF	OFF
JTAG Port Persistence	Enabled	Enabled
SSPI Port Persistence	Disabled	Enabled
I ² C Port Persistence	Disabled	Enabled
MSPI Port Persistence	Disabled	Disabled
I ² C Programmable Primary Configuration Address ²	yyyyxxxx00	1111000000
UFM OTP	OFF	OFF
SRAM OTP	OFF	OFF
Config Flash OTP	OFF	OFF
my_ASSP Enable	OFF	OFF

1. CONFIG_SECURE 制約を使用してイネーブル / ディセーブルします

2. "x" および "y" は IPexpress™ でユーザが指定できます

フィーチャ行を変更するのは開発時、及びフィールドでのアップグレード時（やむをえない場合）のみとすることを強く推奨します。これは、フィーチャ行は使用可能にするコンフィグレーション・ポートの制御を行うためです。アクティブなコンフィグレーション・ポートが使用できなくなり、その後のアップデートを行えなくなる可能性があります。

フィーチャ行の変更により、MachXO2 がコンフィグレーションできなくなることがあります。 PROGRAMN、INITN、DONE の各制御及びステータスピンは、フィーチャ行を使用してイネーブル及びディセーブル制御されます。PROGRAMN 入力ピンは、汎用 I/O としての使用が回復されるかもしれません。フィーチャ行ステートを消去すると、PROGRAMN 入力は汎用 I/O ではなく、PROGRAMN として機能しますので、汎用 I/O として Low にドライブされていると、MachXO2 がコンフィグレーション・プロセスを完了できなくなります。

I²C インターフェイスにおいても SSPI インターフェイスを GPIO として用いる場合は同様の注意が必要です。フィーチャ行の消去と再プログラミングすることによって、一時的に全コンフィグレーション・インターフェイスがイネーブルされます。SSPI チップセレクトである SN ピンが GPIO として用いられる場合で、Low にドライブされる或いは接続されると、I²C インターフェイスよりも優先度の高い SSPI インターフェイスがイネーブルされるため、プログラミングやコンフィグ・プロセスが割り込まれる結果となり、プロセスが無事終了できなくなります。

フィーチャ行は Diamond Programmer によって消去や（再）プログラミングが可能で、消去・プログラム・ベリファイのシーケンスで行われます。バックグラウンド・フラッシュ・プログラミング時にフィーチャ行の消去と再プログラムを望まない時は、"XFLASH Erase, Program, Verify, Refresh" 操作を選択することを推奨します。

フィーチャ行の設定は、Diamond の Spreadsheet View（スプレッドシート・ビュー）を使用することで変更できます。Spreadsheet View では、MachXO2 のコンフィグレーション設定を編集し、その設定は LPF（Lattice Preference File）に書き出されます。この設定はマッピングや配置配線の各処理ステップで参照され、MachXO2 のコンフィグレーション・データに反映されます。

主な機能・動作のまとめ

- ・ 実動作環境にて更新することは期待せず、開発時のみの書き換えを意図している
- ・ フィーチャ行の値を変更することはアクティブに設定しているコンフィグレーションポートが使用できなくなる可能性がある
- ・ Diamond Programmer や Diamond Spreadsheet View で書き換えが可能
- ・ フラッシュ・アップデート時に消去・再プログラミングされるため、フィーチャ行の値を変えないようにすること

コンフィグレーション・モード

MachXO2 のコンフィグレーション SRAM メモリに有効なコンフィグレーション・データをロードしないと動作しません。MachXO2 でコンフィグレーション・データを SRAM にロードする方法は、4 つあります。表 14-5 に示すこれらの方法には、それぞれ利点があります。

表 14-5. コンフィグレーション・モード

モード	ピン数	最大周波数
1149.1 JTAG	4 (5)	25 MHz
SDM（セルフダウンロード = 内部）	0	N/A
外部ダウンロード	4	50 MHz
デュアルブート・ダウンロード	0/4	N/A / 50 MHz

殆どの MachXO2 デザインのコンフィグレーション・モードは、セルフダウンロード・モード（Self-Download Mode、以下 "SDM"）です。この方法では、内部クロックが外付けメモリの速度よりも高速となるため、コン

フィグレーション速度に関して利点があります。また製品コストを増加させる不揮発メモリが不要です。JTAG ポートを使用して、外部プログラマから SRAM にロードする必要もありません。

外部ダウンロード（External Download）モードの利点は、UFM の全てを使用できることです。コンフィグレーション・イメージが UFM にオーバフローすることや、使用可能な内部フラッシュメモリがオーバフローする心配がありません。

デュアルブート・モード（Dual-boot Mode）の利点は、MachXO2 のコンフィグレーションの信頼性を上げられることです。MachXO2 は、最初に内部フラッシュメモリ・イメージをロードし、それに失敗した場合でも、フェイルセーフ・コンフィグレーション・データイメージを MachXO2 の SRAM にダウンロードできるため、動作し続けることができます。MachXO2 のコンフィグレーション失敗の最大の原因は内部メモリの再プログラム失敗（一般的には、電源が失われたことによる）です。

JTAG ポートの利点は、MachXO2 システムのプログラミング、コンフィグレーション、及びテスト用の多様な機能を提供することです。

sysCONFIG™ ポート

表 14-6. MachXO2 プログラミング及びコンフィグレーション・ポート

インターフェイス	ポート	記 述
JTAG	JTAG (IEEE 1149.1 および IEEE 1532 互換)	4 線または 5 線式 JTAG インターフェイス
sysCONFIG	SSPI	SSPI (Slave Serial Peripheral Interface)
	MSPI	MSPI (Master Serial Peripheral Interface)
	I ² C	I ² C (Inter-integrated Circuit) インターフェイス
内部	WISHBONE	内部 WISHBONE バスインターフェイス

sysCONFIG ピン

MachXO2 にある sysCONFIG I/O ピンは、デバイスのプログラム及びコンフィグレーションに使用できます。sysCONFIG ピンはグループにまとめられてポート（JTAG、SSPI、I²C、MSPI）を構成し、内蔵されたリソースのプログラミング、コンフィグレーション、及びアクセスのための MachXO2 との通信に使用されます。コンフィグレーションポート・グループ内の sysCONFIG ピンは、アクティブにしてプログラミングに使用できますが、再コンフィグレーションして汎用 I/O として機能することもできます。

コンフィグレーション・ポートピンを汎用 I/O として回復するには、次のガイドラインに従う必要があります。

- ・ 未使用ポートをディセーブルする必要があります。そのためには、Diamond の Spreadsheet View にある Global Preferences（グローバル制約）タブを使用します。各コンフィグレーション・ポートは、sysCONFIG オプションツリーにリストされます。
- ・ 外部ロジックが、デバイス・プログラミングに影響を与えないようにする必要があります。MachXO2 がフィーチャ行 HW デフォルトモード・ステートのときに、回復された sysCONFIG ピンがドライブされないようにします。例えば、MachXO2 がフィーチャ行 HW デフォルトモード・ステートになったら、PROGRAMN を Low にドライブします。PROGRAMN をディセーブルした状態でフィーチャ行の再プログラムに失敗すると、MachXO2 をコンフィグレーションすることも、ユーザモードにすることもできなくなります。
- ・ JTAGENB を使用して、JTAG ポートを選択的にイネーブル及びディセーブルする際は注意が必要です。JTAG I/O に接続された外部ロジックが、JTAG プログラミングポートと競合してはいけません。

表 14-7 に共有 sysCONFIG ピンのデフォルト・ステートを示します。この表に示したとおり、フィーチャ行デフォルトモードのデバイスは JTAG、SPI スレーブ、及び I²C ポートがイネーブルされます。ユーザモードになると、MachXO2、SSPI のデフォルト・ステート、及び I²C sysCONFIG ピンは汎用 I/O になります。その結果、デフォルトの sysCONFIG ポート設定を使用すると、SSPI または I²C を使用して MachXO2 をプログラ

ムできなくなります。SSPI または I²C sysCONFIG ピンをユーザモードのままにするには、Diamond の Spreadsheet View エディタを使用して、それらのピンを必ずイネーブルする必要があります。

特記のない限り、sysCONFIG ピンには V_{CCIO0} 電圧が印加されます。バンク 0 に接続された他のロジックを用意する場合は、このことに十分注意してください。

以下、各 sysCONFIG ピンの機能について詳しく説明します。

表 14-7. sysCONFIG ピンのデフォルト・ステート

ピン名	関連する sysCONFIG ポート	フィーチャ行デフォルトモードでデフォルトのピン機能 (コンフィグレーション・モード)	ピンの方向 (コンフィグレーション・モード)	ユーザモードフィーチャ行のデフォルト機能
PROGRAMN ¹	SDM	PROGRAMN	弱いプルアップのある入力	ユーザ定義 I/O
INITN	SDM	I/O	弱いプルアップのある I/O	ユーザ定義 I/O
DONE	SDM	I/O	弱いプルアップのある I/O	ユーザ定義 I/O
MCLK/CCLK	SSPI/MSPI	SSPI	弱いプルアップのある入力	ユーザ定義 I/O
SN	SSPI/MSPI	SSPI	弱いプルアップのある入力	ユーザ定義 I/O
SI/SISPI	SSPI/MSPI	SSPI	入力	ユーザ定義 I/O
SO/SPISO	SSPI/MSPI	SSPI	出力	ユーザ定義 I/O
CSSPIN	MSPI	I/O	弱いプルアップのある I/O	ユーザ定義 I/O
SCL	I ² C	I ² C	双方向	ユーザ定義 I/O
SDA	I ² C	I ² C	双方向	ユーザ定義 I/O

表 14-8. 各ポートの Diamond におけるデフォルト設定値

sysCONFIG ポート	デフォルト ¹
SDM_PORT	Disable
SLAVE_SPI_PORT	Disable
I2C_PORT	Disable
MASTER_SPI_PORT	Disable
JTAG_PORT	Enable
1. 本オプション設定は Diamond の Spreadsheet View 内、Global Preference タブで変更が可能	

セルフダウンロード・ポートピン

PROGRAMN: MachXO2 のコンフィグレーションに使用される制御入力ピン。イネーブルされると、PROGRAMN ピンは High から Low の遷移にセンシティブで、内部的に弱くプルアップされます。PROGRAMN が Low にドライブされると、MachXO2 はユーザモードを終了し、前述のように初期化ステップのデバイス・コンフィグレーション・シーケンスを開始します。PROGRAMN ピンを Low に保持すると、MachXO2 は初期化ステップを終了できなくなります。PROGRAMN には、ドライブされたことを MachXO2 が認識するためのドライブ期間の最小パルス幅があり、MachXO2 ファミリデータシートの AC タイミング項に記載されています。

PROGRAMN ピンがイネーブルされている場合は、次の特殊ケースに注意してください。

- ・ デバイスが JTAG からプログラム中の場合は、JTAG モード・プログラミング・シーケンスが完了するまで PROGRAMN は無視されます
- ・ デバイスのコンフィグレーション中に PROGRAMN ピンをトグルすると、プロセスが中断されて、コンフィグレーション・サイクルが再始動されます。ただし、デバイスのコンフィグレーションが進行中

は PROGRAMN ピンはデアサート (High) 状態に保持しておく必要があります

- ・ フィーチャ行デフォルトモード状態のデバイスで PROGRAMN をドライブすると、SSPI 及び I²C ポートがディセーブルされます。SSPI または I²C のプログラミング動作は、PROGRAMN のネゲート後に開始してください
- ・ PROGRAMN は汎用 I/O として確保されている場合でも電源起動時はアクティブです。V_{CC} がその (min) に達してから INITN が立ち上がるまでの期間より早く PROGRAMN に接続された入力信号が High から Low に遷移させてはなりません。さもないと、しばしば MachXO2 がコンフィグレーションに失敗することとなり、リセット状態のままになります。図 14-5 を参照してください
- ・ PROGRAMN は GPIO として用いられる場合でも、フィーチャ行が消去・再プログラミングされる場合はネゲートされている必要があります。アサートされているとコンフィグレーション手順が完了しない可能性があります

図 14-5. PROGRAMN が常時監視される周期

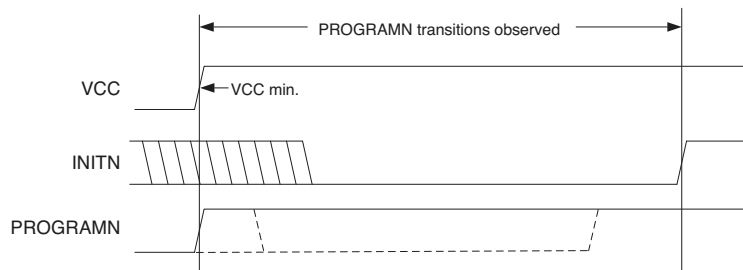
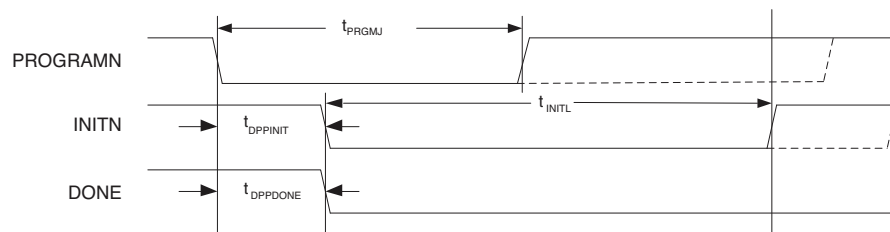


図 14-6. PROGRAMN からのコンフィグレーション・タイミング



INITN: INITN ピンは双方向のオープンドレイン制御ピンで、次の機能があります。

- ・ 電源投入後、PROGRAMN のドライブ後、または REFRESH コマンド実行後に Low になり、コンフィグレーション SRAM の消去中であることを示します。Low のドライブ時間は、 t_{INITL} パラメータで規定されます。
- ・ t_{INITL} 期間の経過後、INITN ピンがネゲートされて (High) コンフィグレーションの準備ができたことを示します。MachXO2 は、内部フラッシュメモリまたは外付け SPI フラッシュから、コンフィグレーション・データのロードを開始します。
- ・ INITN を t_{INITL} 期間の経過前に外部デバイスから Low にドライブすることにより、コンフィグレーションを開始することを防止できます。複数のプログラマブルデバイスがチェーン接続されている場合は、これが役立ちます。 t_{INITL} 時間が最も長いプログラマブルデバイスは、そのデバイスの準備ができるまで、チェーン内の他の全てのデバイスにデータ取得開始を待たせることができます。
- ・ INITN が持つ最後の機能は、コンフィグレーション・データの読み出し中にエラーを通知することです。 t_{INITL} が経過し、INITN ピンが一度 High になった後に INITN が Low になることは、コンフィグレーション中に MachXO2 がエラーを検出したことを示します。

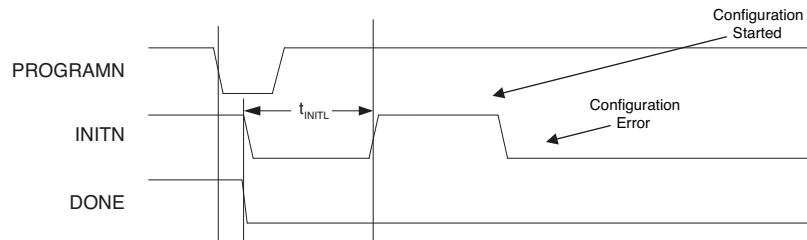
INITN は次の条件でアクティブになり、初期化ステートがアクティブであることを示します。

- ・ 電源の供給が開始された
- ・ PROGRAMN の立ち下がりエッジが発生した

- ・ スレーブ・コンフィグレーション・ポート（JTAG、SSPI、I²C、または WISHBONE）を使用して、IEEE 1532 REFRESH コマンドが送信された

エラー条件によって INITN ピンがドライブされた場合は、コンフィグレーション・ビットストリームを修正し、MachXO2 を強制的に初期化状態にすると、エラーを解消できます。

図 14-7. コンフィグレーション・エラーの通知



フィーチャ行 HW デフォルトモード・ステートにある場合、MachXO2 デバイスの INITN の状態はデバイス外部から認識できません。このモードの INITN ピンは、デフォルトで High にプルアップされます。図 14-7 に示す INITN の動作が MachXO2 の外部から認識されるのは、INITN ピンがイネーブルされた場合だけです。

INITN は汎用 I/O として回復できます。デフォルトでは、INITN ピンはディセーブルされます。イネーブルするには、Diamond の Spreadsheet View を使用します。

ビットストリームの読み出し時にエラーが検出された場合は、INITN が Low になり、内部 DONE ビットはセットされず、DONE ピンが Low のままとなって、デバイスはウェイクアップしません。次の場合、デバイスはコンフィグレーションに失敗します。

- ・ ビットストリーム CRC エラーが検出された
- ・ 無効なコマンド（エラー）が検出された
- ・ オンチップフラッシュからのロード時にタイムアウトエラーが発生した
- ・ オンチップ・コンフィグレーション SRAM またはオンチップ・フラッシュメモリの末尾に達したときに、プログラム完了（Program Done）コマンドを検出しなかった

DONE: 弱くプルアップされた双方向オープンドレインで、MachXO2 がユーザモードであることを通知します。DONE がユーザモードへの移行を最初に示すことができるのは、内部 DONE ビットがアサートされた後だけです。内部 DONE ビットは、MachXO2 のウェイクアップ・ステートの開始を示します。

DONE ピン出力は SDM_PORT コンフィグレーション・パラメータで制御され、Diamond の Spreadsheet View（スプレッドシート・ビュー）で変更できます。デフォルトでは、MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合に、DONE ピンは汎用 I/O になります。デフォルトモードでは、内部 DONE ビットのアサート後、MachXO2 が自動的にウェイクアップ・シーケンスを順に実行します。MachXO2 がウェイクアップを停止し、DONE ピンが High にドライブされるまで待つことはありません。

外部デバイスが DONE ピンを Low にドライブし続けると、その期間はユーザモードに移行しない状態に保持できます。そのためには、SDM_PORT で DONE I/O をイネーブルし、MachXO2 のフィーチャ行をプログラムする必要があります（この「ストール機能」は Diamond 3.5 以降でサポートされます。これ以前のバージョンでは、SDM_POT が DONE I/O をイネーブルしても、本機能はイネーブルになりません）。DONE を Low にドライブしたままにする一般的な理由は、複数の FPGA を全てコンフィグレーションできるようにすることです。各 FPGA が DONE ステートに達するとユーザ動作を開始できます。（チェーンにして DONE ピンをワイヤードオア接続した場合、）最後の FPGA がコンフィグレーションされると、全ての FPGA が一斉に動作を開始します。

MachXO2 が初期化モードになると、INITN ピンと連動して DONE ピンが Low にドライブされます。既に説明したとおり、電源が投入されるか、PROGRAMN がドライブされるか、アクティブなコンフィグレーション・ポートから IEEE 1532 Refresh コマンドを受信するとこの状態になります。

DONE ピンをサンプリングすることで、外部デバイスが MachXO2 のコンフィグレーション終了を知る方法として利用できます。ただし、IEEE 1532 JTAG を使用して SRAM をコンフィグレーションすると、DONE ピンがバウンダリスキャン・セルによってドライブされるため、IEEE 1532 JTAG によるコンフィグレーション中は DONE ピンの状態が意味を持たなくなります(コンフィグレーションが完了すると、DONE はフィーチャ行の SDMPORT 設定で定義された動作となります)。MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合も、DONE ピンは High にプルアップされます。この動作によって、DONE ピンをモニタしている他のロジックに、コンフィグレーションの完了を認識させることができます。

MSPI 及び SSPI コンフィグレーション・ポートピン

表 14-9. MSPI コンフィグレーション・ポートピン

ピン名	機能	方向	記 述
MCLK/CCLK	MCLK	弱いプルアップのある出力	MachXO2 のコンフィグレーション・ロジックからスレーブ SPI フラッシュメモリへの、データの送信 / 受信のタイミングに使用されるマスタクロック。外部及びデュアルブート・コンフィグレーション・モードでは、MCLK に 1K のプルアップ抵抗を推奨
CSSPIN	CSSPIN	出力	コンフィグレーション・データが格納された外付け SPI フラッシュメモリのイネーブルに使用されるチップセレクト
SI/SISPI	SISPI	出力	MachXO2 コンフィグレーション・ロジックからスレーブ SPI フラッシュメモリに出力データを伝達する
SO/SPISO	SPISO	入力	スレーブ SPI フラッシュメモリから MachXO2 コンフィグレーション・ロジックに出力データを伝達する
SN	SN/IO	入力	MachXO2 コンフィグレーション・ロジックの SSPI チップセレクト入力。MSPI ポートがアクティブの場合は常に、外部で High にプルアップする

表 14-10. SSPI コンフィグレーション・ポートピン

ピン名	機能	方向	記 述
MCLK/CCLK	CCLK	弱いプルアップのある入力	外部 SPI マスタデバイスから MachXO2 コンフィグレーション・ロジックへのデータの送信 / 受信のタイミングに使用されるクロック
SI/SISPI	SI	入力	外部 SPI マスタから MachXO2 コンフィグレーション・ロジックに出力データを伝達する
SO/SPISO	SO	入力	MachXO2 コンフィグレーション・ロジックから外部 SPI マスタに出力データを伝達する
SN	SN	弱いプルアップのある入力	MachXO2 コンフィグレーション・ロジックの SSPI チップセレクト入力。SN はアクティブ Low 入力

MCLK/CCLK: MCLK/CCLK がアクティブな場合は、MachXO2 のコンフィグレーション・データをシーケンシャルにロードするためのクロックとして使用されます。このピンの機能は次のとおりです。

フィーチャ行 HW デフォルトモード・ステートの MachXO2 では、MCLK/CCLK ピンはデフォルト・ステートでコンフィグレーション・クロック (CCLK) として機能します。このステートでは、外部 SSPI マスタコントローラが MachXO2 をプログラムできます。最大 CCLK 周波数及びセットアップ / ホールド・パラメータは、MachXO2 ファミリデータシートの AC タイミング項に記載されています。SSPI ポートを使用して、ユーザモードになった MachXO2 をプログラムする場合は、SSPI ポートをイネーブルするようフィーチャ行を構成する必要があります。

MachXO2 がデュアルブートまたは外部ブートモードに設定されている場合、MCLK/CCLK ピンはマスタクロック (MCLK) として機能します。これらのモードを使用する際は、1K の外付けプルアップ抵抗を推奨します。MCLK が出力となり、MachXO2 の MSPI コンフィグレーション・ポートに接続された SPI フラッシュ

の基準クロックを提供します。全てのコンフィグレーション・データを受信するまで、MCLK はアクティブにドライブされます。MachXO2 がユーザモードになると、MCLK 出力はトライステートになり、汎用 I/O として使用できるようになります。しかしコンフィグレーション後の多くの使い方としては、MCLK は外部 SPI フラッシュメモリによるメモリ・トランザクション実行用の基準クロックとして確保されます。

MachXO2 は内部オシレータから MCLK を発生します。通常、MCLK の初期周波数は 2.08MHz です。この MCLK 周波数は、MCCLK_FREQ パラメータを使用して変更できます。MCCLK_FREQ の選択には、Diamond の Spreadsheet View を使用します。対応している MCLK 周波数の全リストは、表 14-11 を参照してください。

表 14-11. MachXO2 MCLK の有効な周波数 (MHz)

2.08	9.17	33.25
2.46	10.23	38.00
3.17	13.30	44.33
4.29	14.78	53.20
5.54	20.46	66.50
7.00	26.60	88.67
8.31	29.56	133.00

デバイス・コンフィグレーションの初期化ステップの中で、MCCLK_FREQ を使用して指定された周波数値が MachXO2 にロードされます。MachXO2 が新しい MCLK_FREQ 値を受け取ると、MCLK 出力は選択された周波数でのドライブを開始します。MCCLK_FREQ を選択するときは、SPI フラッシュメモリまたは PCB の周波数仕様を超過しないよう注意し、また MachXO2 ファミリーデータシートで MachXO2 の AC 仕様を確認してください。

SN: SSPI ポートのチップセレクト入力ピンです。外部 SPI バスマスタは、MachXO2 のプログラミング及びコンフィグレーション・ロジックを使用して動作を実行するため、SN ピンをアクティブ Low にドライブします。SN ピンを使用できるのは、MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合、及び SSPI ポートが ENABLE に設定されたステートのユーザモードの場合です。SSPI ポートが DISABLE に設定されている場合、SN ピンは汎用 I/O になります。

MachXO2 が正しく動作するには、SN ピンを正しいステートに保つことが重要です。

- ・ MSPI を使用してコンフィグレーションする場合は、SN をネゲート（High に固定）する必要があります。SN ピンは電源立ち上がり時にノイズが乗らないようにする必要があります。フラッシュメモリからのダウンロードが不成功になる恐れがあります。スレーブ SPI モードを使用してコンフィグレーションしている期間中、SN ピンはアサートしていなければいけません。
- ・ MachXO2 がユーザモードで内部 WISHBONE バスを使用して SPI メモリ・トランザクションを初期化する場合は、SN をネゲートする必要があります。
- ・ I²C を使用して MachXO2 内のコンフィグレーション・ロジックにアクセスする場合は、SN をネゲートする必要があります。
- ・ SN がドライブされた場合は、CSSPIN をネゲートし、共有 SPI ピンをハイインピーダンス状態にする必要があります。
- MSPI ポートと SSPI ポートは、SI/SISPI、SO/SPISO、MCLK/CCLK という 3 本のピンを共有します。MachXO2 では、両方のポートを使用できますが、両方に同時にアクセスすることはできません。SSPI 及び MSPI ポートの両方をイネーブルした場合は、これらのポートを時分割で使用する必要があります。
- ・ SN ピンは GPIO として用いられる場合でも、フィーチャ行が消去・再プログラミングされる場合はネゲートされている必要があります。アサートされているとコンフィグレーション手順が完了しない可能性があります。

SN ピンを外部から High にプルアップすることで、内部の弱いプルアップを強化することを推奨します。

CSSPIN: MSPI コンフィグレーション・モードで外付け SPI フラッシュをイネーブルするため使用される、アクティブ Low のチップセレクト入力ピンです。外部またはデュアルブート・モードでコンフィグレーションするよう MachXO2 がプログラムされると、接続された SPI フラッシュに対して CSSPIN ピンがドライブされます。MachXO2 は、全てのコンフィグレーション・データバイトがロードされるまで CSSPIN をアサートし続け、その後 CSSPIN はハイインピーダンス状態になります。

MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合、CSSPIN ピンは弱くプルダウンされた汎用 I/O となります。外部及びデュアルブート・コンフィグレーション・モードで使用する場合は、外付けのプルアップ抵抗が必要です。CSSPIN の電位は、SPI フラッシュメモリ VCC の立ち上がり・下がりに連動する必要があります。MachXO2 がユーザモードになると、汎用 I/O のままになります。内部 SPI マスタロジックが CSSPIN を使用できるよう予約するには、MSPI ポートをイネーブルする必要があります。

外部 SPI フラッシュメモリからコンフィグレーションする場合は、SPI フラッシュメモリと MachXO2 の V_{CCIO2} が同じ電圧にしてください。また、SPI フラッシュメモリの電圧範囲が推奨動作電圧内であることも確実にしてください。

SPI フラッシュメモリによっては、チップセレクト入力フラッシュメモリの VCC と連動して立ち上がり・下がりする必要があります。デフォルトで、CSSPIN ピンには内部に弱いプルダウン抵抗が付いていますので、4.7 ~ 10K Ω のプルアップ抵抗を付加することを推奨します。

SI/SISPI: デュアル機能の双方向ピンで、その方向はマスタまたはスレーブのどちらのモードがアクティブかで決まります。SI/SISPI は、SSPI モード使用時は入力データピンになり、MSPI モード使用時は出力データピンになります。MSPI モードでは、全てのコンフィグレーション・データバイトがロードされるまで MachXO2 が SI/SISPI をドライブし、その後 SI/SISPI はハイインピーダンス・ステートになります。

このピンを SI/SISPI として確保し、SPI インターフェイスへのアクセスを許可するには、sysCONFIG 制約の SLAVE_SPI_PORT または MASTER_SPI_PORT の少なくとも 1 つを ENABLE に設定する必要があります。

SO/SPISO: デュアル機能の双方向ピンで、その方向は、マスタまたはスレーブのどちらのモードがアクティブかで決まります。MSPI モード使用時は入力データピンになり、SSPI モード使用時は出力データピンになります。

このピンを SO/SPISO として確保し、SPI インターフェイスへのアクセスを許可するには、sysCONFIG 制約の SLAVE_SPI_PORT または MASTER_SPI_PORT の少なくとも 1 つを ENABLE に設定する必要があります。

I²C コンフィグレーション・ポートピン

SCL: MachXO2 には I²C コンフィグレーション・ポートがあります。SCL は I²C シリアルクロック・ピンで、I²C バス・トランザクションの開始とタイミングに使用されます。双方向のオープンドレイン信号で、MachXO2 の I²C コントローラがバストランザクション・マスタの場合は出力、外部 I²C マスタが MachXO2 内部リソースにアクセスする場合は入力になります。SCL が動作するためには、外付けプルアップ抵抗が必要です。

SCL ピンを使用できるのは、MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合です。I²C ポートがユーザモードでも使用できるようにするには、I2C_PORT をイネーブルする必要があります。I²C ポートを ENABLE に設定しなかった場合、SCL ピンは汎用 I/O になります。

SDA: I²C シリアルデータ入出力ピンです。双方向のオープンドレインで、動作するためには外付けプルアップ抵抗が必要です。このピンの方向は、I²C バス上のデータトランザクション中に動的に変化します。状態は、その時点のバスマスタがどれかと、そのマスタが実行中の動作で決まります。

SDA ピンを使用できるのは、MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合です。I²C ポートがユーザモードでも使用できるようにするには、I2C_PORT をイネーブルする必要があります。I²C ポートを ENABLE に設定しなかった場合、SDA ピンは汎用 I/O になります。

JTAG コンフィグレーション・ポートピン

JTAG ピンは標準 IEEE 1149.1 TAP (Test Access Port) を提供します。JTAG ポートは MachXO2 で唯一の、コンフィグレーション、プログラミング、及び複数デバイス・コンフィグレーションの各機能の実行が可能なコンフィグレーション・ポートです。JTAG ポートによるプログラミングとコンフィグレーションでは、IEEE 1532 準拠のコマンドが使用されます。IEEE 1532 機能に加えて、MachXO2 には必須の IEEE 1149.1 TAP コマンドが全て用意され、プリント基板アセンブリの検証が可能です。

JTAG ポートは、MachXO2 がフィーチャ行 HW デフォルトモード・ステートの場合に、デフォルトでイネーブルになります。他の全てのコンフィグレーション・ポートピンと同様、JTAG ピンも汎用 I/O にすることができます。他のポートとは異なり、JTAG ポートのデフォルト・ステート (ENABLE ステート) はユーザモードでもアクティブなままになります。JTAG_PORT 制約を DISABLE ステートに設定すると、JTAG ピンを汎用 I/O として回復できます。JTAG ポートは、専用プログラムピンのままにすることを推奨します。

JTAG ポートを DISABLE ステートに設定すると、JTAGENB 入力がいネーブルされ、JTAG ピンを多重化できます。JTAGENB を High にドライブすると、JTAG ピンは IEEE 1149.1 の特性を持ちます。JTAGENB をネゲート (Low にドライブ) すると、JTAG ポートピンが汎用 I/O になります。JTAG ポートピンの多重化を利用する場合は、JTAG ポート回路の設計には注意が必要です。JTAG ポートに接続されたロジックとの競合を防止してください。

デバイスが IEEE 1149.1 制御でプログラムされている場合は、プログラミングの進行状況の判定に、DONE などの sysCONFIG プログラミングピンを使用できません。これはバウンダリスキャン・セルの状態によるピンのドライブが、通常の内部ロジックではなく IEEE JTAG 標準に従って行われるためです。

表 14-12. JTAG ポートピン

ピン名	ピンの機能 (コンフィグレーション・モード)	ピンの方向 (コンフィグレーション・モード)	デフォルト機能 (ユーザモード)
TDI	TDI	弱いプルアップのある入力	TDI
TDO	TDO	弱いプルアップのある出力	TDO
TCK	TCK	入力	TCK
TMS	TMS	弱いプルアップのある入力	TMS
JTAGENB	I/O	弱いプルダウンのある I/O	I/O

TDO (Test Data Output) : シリアルテスト命令及びデータをシフトアウトするために使用されます。TDO が内部回路からドライブされていない場合、このピンはハイインピーダンスになります。ハイインピーダンスでなくなるのは、JTAG ステートマシンがシフト IR またはシフト DR ステートの場合だけです。このピンは、JTAG コネクタの TDO、または JTAG チェーンの下流デバイスの TDI に配線する必要があります。TDO ピンにはプルアップ抵抗が内蔵されており、VCCIO バンク 0 (V_{CCIO0}) にプルアップされています。

TDI (Test Data Input) : シリアルテスト命令及びデータをシフトインするために使用され、JTAG コネクタの TDI、または JTAG チェーンの上流デバイスの TDO に配線する必要があります。TDI ピンにはプルアップ抵抗が内蔵されており、V_{CCIO0} にプルアップされています。

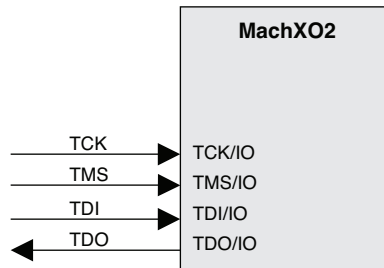
TMS (Test Mode Select) : 1149.1 準拠ステートマシンのステート遷移を制御する入力ピンで、TCK の立ち上がりエッジでサンプリングされます。JTAG ステートマシンは、現在の TAP ステートとサンプル時の TMS 入力レベルによって、そのステートを維持するか、または新しい TAP ステートに遷移します。JTAG 仕様に従って、TMS にはプルアップ抵抗が内蔵されており、V_{CCIO0} にプルアップされています。

TCK (Test Clock) : 他の JTAG ポートピンのタイミングに使用されるクロックです。データは TCK の立ち上がりエッジで命令レジスタまたはデータレジスタにシフトインされ、TCK の立ち下がりエッジでシフトアウトされます。TAP は静的デザインとなっていて、TCK は High または Low ステートで停止することができます。TCK の最高入力周波数は、MachXO2 ファミリーデータシートの DC 及びスイッチング特性の項に記載さ

れています。TCK ピンはプルアップされていません。MachXO2 の電源投入時に TAP コントローラの予期しないクロック動作を防止するため、4.7KΩ の外付けプルダウン抵抗を推奨します。

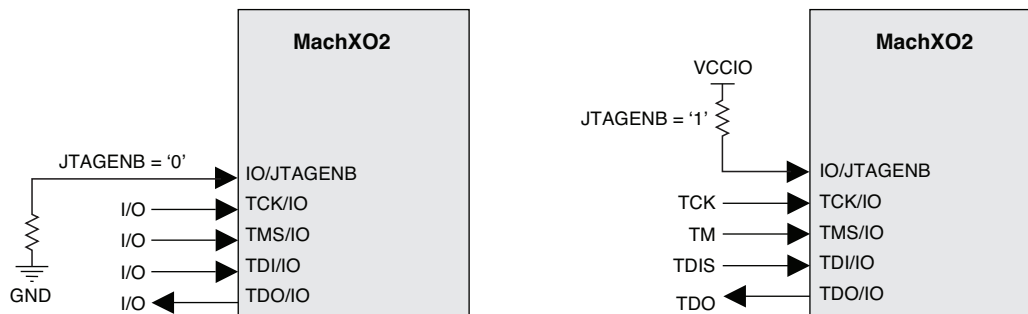
JTAGENB: JTAG ENABLE ピンは IEEE 1149.1 準拠ピンとも呼ばれ、JTAG ポートの多重化に使用可能な入力ピンです。JTAGENB ピンはユーザモードでのみアクティブになります。JTAG ポートが ENABLE ステートのときに、JTAGENB ピンはユーザ I/O になります。図 14-8 に、MachXO2 デバイスの JTAG ポートのデフォルト動作を示します。

図 14-8. JTAG_PORT = ENABLE 時の JTAG ポートのデフォルト



JTAG ポートは汎用 I/O にできます。それには、JTAG_PORT の制約を Diamond の Spreadsheet View で DISABLE に設定します。JTAG ポートが DISABLE ステートの場合、JTAGENB ピンは専用入力になります。JTAGENB を Low にすると JTAG ポートがディセーブルされ、4 本の JTAG ピンが汎用 I/O になります。JTAGENB 入力を High にすると、JTAG ポートがイネーブルされます。図 14-9 に、JTAGENB で制御された JTAG ポートの動作を示します。

図 14-9. JTAG_PORT = DISABLE 時の JTAG ポートの動作



JTAGENB 機能を使用する際は、JTAG I/O ピンに接続されたロジックが、JTAG プログラミングシステムと競合しないようにすることが非常に重要です。外部ロジックは、外部プログラミングシステムで実行される全ての JTAG トランザクションを無視する必要があります。

ラティスのパラレルポートまたは USB ダウンロードケーブルには、ispEN という出力があります。この ispEN 信号を JTAGENB 入力に接続すると、JTAG ポートを使用するかどうかを制御できます。JTAGENB 入力を制御する別のメカニズムとしては、必要に応じて取り付けと取り外しが可能なジャンパを使用する方法があります。

コンフィグレーション・モード

MachXO2 には、不揮発性メモリからコンフィグレーション SRAM にロードする複数のオプションが用意されています。前の項では、MachXO2 コンフィグレーション・ロジックとの通信に必要な物理インターフェイスについて説明しました。この項では、様々なコンフィグレーション・モードそれぞれの機能を中心に説明します。Diamond の Spreadsheet View (スプレッドシート・ビュー) で必要な、重要な設定についても説明します。

SDM モード

SDM は MachXO2 の主要なコンフィグレーション・モードで、その利点としては次のものがあります。

- ・ **速度**：デバイス規模にもよりますが、MachXO2 が数ミリ秒で実行可能な状態になります。
- ・ **セキュリティ**：SRAM へのロード中、コンフィグレーション・データがデバイス外部に露出することはありませんので、内部メモリが読み取られることを防止できます。
- ・ **低コスト**：MachXO2 のプログラミング専用の不揮発メモリを購入する必要がありません。
- ・ **ボード面積の削減**：外付け不揮発メモリが不要となるため、ボードを小型化できます。
- ・ **信頼性の向上**：システムアップデート時に内部フラッシュメモリに問題がある場合、MachXO2 を外付けフラッシュメモリから起動できます。

SDM を使用している場合、MachXO2 はコンフィグレーション・データを内部フラッシュメモリから取得します。SDM がトリガされるのは、電源投入時、REFRESH コマンドの受信時、または PROGRAMN ピンのドライブ時です。図 14-6 に示したように、内部フラッシュメモリには 3 つのセクタがあります。ほとんどの場合、最初のセクタには MachXO2 デバイスのコンフィグレーション・データを格納できるだけの十分な容量があります。コンフィグレーション・データサイズは、圧縮の程度、及びデザイン内で初期化する EBR コンポーネント数に依存して変動します。コンフィグレーション・データのサイズが増加すると、CFM セクタがオーバーフローする可能性があります。コンフィグレーション・データが UFM セクタにオーバーフローできるようにすると、このオーバーフローに対処できます。まれなケースとして、コンフィグレーション・データが CFM と UFM を合わせたサイズからオーバーフローすることがありこの場合、SDM が使用できません。CFM/UFM オーバーフローの発生時は、MSPI コンフィグレーション・モードを使用する必要があります。

通常の状況では、コンフィグレーション・データが CFM セクタに完全に収まります。初期化 EBR がそれほど多数ないデザインは、ほぼ必ずこの条件に適合しますので、UFM は内部フラッシュメモリ・アレイとしてユーザ用途に使用できます。CONFIGURATION オプションは CFG に設定することを推奨します。このように設定すると、コンフィグレーション・データが UFM にオーバーフローすることが防止され、UFM に格納されたデータがデバイスの更新時に上書きされることがなくなります。

第 2 のフラッシュセクタである UFM には、3 種類の使用モデルがあります。

- ・ CFM からのコンフィグレーション・データのオーバーフロー
- ・ EBR の初期化とユーザ定義のストレージ
- ・ ユーザ定義のストレージ

デフォルトで Diamond は、初期化 EBR データをコンフィグレーション・データイメージ内に作成します。その結果、コンフィグレーション・データが UFM セクタにオーバーフローすることがあります。デフォルト・ステートを変更するには、Diamond の Spreadsheet ツールを使用して sysCONFIG の CONFIGURATION エントリを修正する必要があります。CONFIGURATION エントリのデフォルト・ステートは“CFG”に設定されています。

コンフィグレーション・データは論理的に分割し、EBR の初期化データを UFM に配置することができます。CONFIGURATION オプションを CFG_EBRUFM に設定すると、Diamond ソフトウェアがコンフィグレーション・データは CFM に配置し、EBR 初期化データを UFM に配置するようになります。この場合、EBR 初期化データが UFM の先頭ページに格納されます。IPexpress を使用すると、メモリ初期化ファイルを UFM に対応付けることができます。なお、現行の Diamond 開発ツールには、UFM に格納された EBR 初期化データを、MachXO2 ファブリック内の対応する EBR にマッピングする手段は用意されていません。

初期化 EBR データの自動割り当てだけでなく、ユーザ固有の目的にデータブロックを追加することもできます。このデータは、EBR 初期化データとの競合を防止するため、UFM の末尾のメモリ位置に格納されます。

ユーザ定義ストレージモードの動作では、UFM セクタを汎用フラッシュメモリのように動作させることができます。IPexpress を使用してデータを初期化することも、消去可能セクタが 1 つの個別フラッシュメモリ・デバイスであるかのように、このセクタを使用することもできます。

この 3 つのいずれの場合も、UFM セクタ全体を消去しなければなりません。コンフィグレーション・データ、EBR 初期化データ、及びインプリメンテーション固有のデータの回復を必ず行う必要があります。すなわち、UFM 内の全てのデータを読み出し、変更する部分とマージし、UFM を消去し、新しいデータを UFM に書き戻す必要があります。

マスタ SPI コンフィグレーション・モード (MSPI)

マスタ SPI (MSPI) コンフィグレーション・モードは、MachXO2 で使用可能な唯一の、もう一つの自己制御コンフィグレーション・モードです。MachXO2 で MSPI がイネーブルになると、外付け SPI フラッシュから自動的にコンフィグレーション・データを取得できるようになります。MachXO2 がフィーチャ行 HW デフォルトモード・ステートになっていると、MSPI コンフィグレーション・ポートを使用できません。MSPI モードでコンフィグレーションを行うときは、必ずフィーチャ行で MSPI ポートをイネーブルしてください。プログラミングエラー時に MachXO2 を回復できるように第二のコンフィグレーション・ポートを用意し、MachXO2 がフィーチャ行デフォルトモード状態（ブランク時や消去状態）のときにアクティブにすることを推奨します。

MachXO2 が MSPI コンフィグレーション・モードで正しく動作することを確実にするために、以下を確認してください：

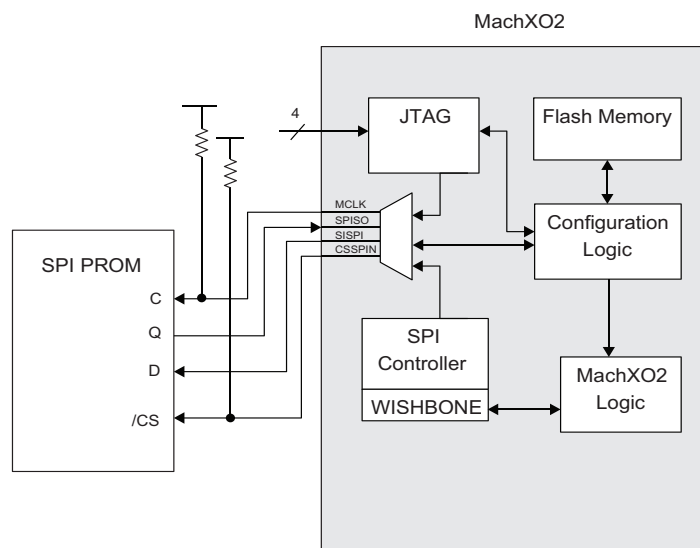
- ・ SPI フラッシュメモリのパワーオンリセット解除レベルが MachXO2 のそれよりも低いこと、或いは SPI フラッシュメモリが最初に電源投入されて起動されること
- ・ SPI フラッシュメモリの Fmax が MachXO2 の MCLK Fmax よりも大きいこと
- ・ MachXO2 のセットアップ時間およびホールド時間要件（データシート参照のこと）を満たすことができるボード配線

表 14-13. MSPI ポートピン

ピン名	I ² C 機能
MCLK	MachXO2 のコンフィグレーション・ロジック及び MSPI コントローラからのクロック出力。MCLK は、スレーブ SPI デバイスの SCLK 入力に接続する
SISPI	MachXO2 から SSPI SI 入力へのシリアルデータ出力
SPISO	SSPI SO から MachXO2 のコンフィグレーション・ロジックへのシリアルデータ入力
CSSPIN	MachXO2 のコンフィグレーション・ロジックから、MachXO2 のコンフィグレーション・データが格納されたスレーブ SPI フラッシュへのチップセレクト出力

表 14-2 に MachXO2 のコンフィグレーション・データに必要なメモリ量を、デバイス規模ごとに示します。リードオペコード 0x03 を受け付け可能な SPI フラッシュを選択してください。MachXO2 で使用可能なリードオペコードは 0x03 だけです。

図 14-10. MSPI コンフィグレーション・モード



電源投入時、REFRESH コマンドの受信時、または PROGRAMN ピンがトグルされると、MachXO2 は SPI フラッシュからのコンフィグレーション・データの取得を開始します。MCLK/CCLK I/O は MCLK (Master Clock、マスタクロック) 機能となり、SPI フラッシュの SCLK 入力に公称 2.08MHz クロックのドライブを開始します。CSSPIN は Low にドライブされ、SI/SISPI 出力から SPI フラッシュにコマンドが送信されて、データが SO/SPISO 入力ピンで SPI フラッシュから読み出されます。全てのコンフィグレーション・データが SPI フラッシュから取得されると、CSSPIN ピンがネゲートされ、MSPI 出力ピンがトライステートになります。

コンフィグレーション・データのダウンロードは MCLK が必ず公称周波数 2.08MHz から開始されます。MCCLK_FREQ パラメータ (Spreadsheet View を使用してアクセス) を使用することで、コンフィグレーション周波数を高速にできます。SPI フラッシュ内のコンフィグレーション・データにはある程度のパディングビットがあり、MCLK の基本周波数を変更するデータが読み込まれた後、MachXO2 は新しい MCLK 周波数を使用して残りのコンフィグレーション・データバイトを読み込みます。

MachXO2 がユーザモードになると、MSPI コンフィグレーション・ポートはトライステートになります。その結果、データを SPI 経由で転送できるようになります。SPI バスによるデータ転送に使用可能な方法は、主に 2 つあります。1 つは、MachXO2 の EFB (Embedded Function Block、組み込み機能ブロック) をイネーブルする方法です。IPexpress を使用して EFB をインスタンス化し、アクティブにする機能を選択します。EFB で使用可能な機能の 1 つは SPI マスタコントローラです。EFB 内部の SPI マスタコントローラは、MSPI コンフィグレーション・ポートピンに直接接続され、SPI バス・トランザクションを初期化するための一連のステータス、制御、及びデータレジスタがあります。レジスタは、内部 WISHBONE データバスを使用してアクセスします。MachXO2 のファブリックに組み込まれるロジックは、WISHBONE ブリッジを経由する EFB とのトランザクションを実行し、それにより SPI バス・トランザクションを生成します。

MSPI コンフィグレーション・ポートのトランザクションを実行するもう 1 つの方法は、JTAG ポートから制御する方法です。MachXO2 には JTAG から MSPI へのパススルー回路が内蔵されているため、スレーブの SPI フラッシュメモリを消去 / プログラム / リードできます。接続された SPI フラッシュをプログラミングする主要な方法は、Diamond の Programmer を使用して PC からコンフィグレーション・データを転送する方法です。この方法は、ボードの開発及びデバッグ時に役立ちます。**注：JTAG から MSPI へのパススルー・プログラミングモードに対応するためには、MCLK に 1kΩ の外付けプルアップ抵抗が必要です。**

JTAG ポートを使用して SPI フラッシュをプログラムするもう 1 つの方法は、Lattice ispVME ソリューションを使用する方法です。ispVME は組み込みマイクロプロセッサ用に書かれた C プログラムです。マイクロプロセッサは Diamond Deployment ツールで作成されたデータファイルを読み取り、ispVME プログラムを実行します。ファームウェアが、ポート I/O を使用して MachXO2 の JTAG ポートをドライブすることで、デー

タが MSPI ポートに渡されます。マイクロプロセッサを使用した、外付け SPI フラッシュの更新の詳細については、ispVME ツールスイートを参照してください。

MachXO2 の外付け SPI フラッシュのプログラミングに JTAG ポートを使用することの利点は、MachXO2 がフィーチャ行 HW デフォルトモード・ステートになっていても構わないことです。JTAG は、フラッシュモード・フィーチャ行またはユーザモード・フィーチャ行ステートでデバイスをプログラムできます。そのためには、MSPI ポートピンをイネーブルする必要があります。パススルーは、JTAG TAP システムに不可欠な部分です。この方法が成功するためには、JTAG ポートが使用できる必要があることは明白です。

MSPI コンフィグレーション・モードを使用して動作するよう MachXO2 を設定するには、以下を行う必要があります。

- ・ コンフィグレーション・データ全体を外部 SPI フラッシュに格納する
- ・ SPI フラッシュ内のデータの開始位置のオフセットが 0x000000 になっていること
- ・ 表 14-14 に示す設計制約を設定する
- ・ Diamond の Process（処理）枠で JEDEC ファイル作成をイネーブルする
- ・ Export Files（ファイルのエクスポート）処理を実行し、デザインを作成する

表 14-14. MSPI コンフィグレーションのソフトウェア設定

制約	設定
MASTER_SPI_PORT	ENABLE
CONFIGURATION	EXTERNAL

Export Files 処理では、JEDEC ファイルと BIT ファイルの両方が生成されます。両方のファイルを使用することが重要です。フィーチャ行に書き込むには、JEDEC ファイルを MachXO2 にプログラムする必要があります。JEDEC ファイルでは、MSPI コンフィグレーション・ポートがイネーブルされます。

外部 SPI フラッシュには BIT ファイルをプログラムする必要があります。データを SPI フラッシュに格納する方法は、次のように幾つかあります。

- ・ JTAG ダウンロードケーブルを使用して、Diamond の Programmer で SPI フラッシュデータを送出する
- ・ マイクロプロセッサで ispVME を実行する
- ・ ATE（自動テスト装置）で JTAG を使用して SPI フラッシュをプログラムする
- ・ プログラム済み SPI フラッシュメモリ PCB に実装する

MachXO2 のフィーチャ行がプログラムされ、SPI フラッシュにコンフィグレーション・データが格納されると、コンフィグレーションをテストできるようになります。PROGRAMN のドライブ、REFRESH コマンドの送信、またはボードの電源再投入を行うと、MachXO2 が外付け SPI フラッシュからコンフィグレーションされます。

デュアルブート・コンフィグレーション・モード

デュアルブート・コンフィグレーション・モードは、SDM と MSPI コンフィグレーション・モードを組み合わせたものです。デュアルブート・モードに設定された MachXO2 は、最初に SDM を使用して内部フラッシュメモリからコンフィグレーションしようとしています。SDM コンフィグレーションに失敗すると、MachXO2 は MSPI モードを使用して自分自身をコンフィグレーションしようとしています。このロード順序は逆にできません。

内部データはプライマリ・コンフィグレーション・データ（プライマリイメージ）で、外部データはゴールデン・コンフィグレーション・データ（ゴールデンイメージ）です。プライマリイメージでは、次のいずれかの障害が発生することがあります。

- ・ ビットストリームの CRC エラーが検出された
- ・ 内部フラッシュからのロード時にタイムアウトエラーが発生した

CRC エラーは、内部フラッシュメモリに誤ったデータが書き込まれることが原因で発生します。データは行単位でフラッシュメモリから読み取られます。各行がコンフィグレーション・エンジンに取り込まれるたびに、データの CRC 整合性がチェックされます。CRC が正しい場合に限り、データがコンフィグレーション SRAM に格納されます。CRC に何らかの誤りがあると、デバイスはコンフィグレーション SRAM を消去し、外部 SPI フラッシュからコンフィグレーション・データを取得し直します。

CRC 計算の観点では誤りデータは訂正が可能です。機能的には正しくなりません。エラーになると、内部 DONE ビットがアクティブになることはありません。MachXO2 は、パワーオンリセット信号の解放後に発生したマスタクロックのパルス数をカウントします。内部 DONE ビットがアクティブにならないままカウントが終了すると、MachXO2 はコンフィグレーション・データを外部 SPI フラッシュから取得しようとします。

SPI フラッシュメモリのパワーオンリセット (POR) 解除が MachXO2 の内部 POR 解除より遅い場合、以下の事象が生じます。

1. MachXO2 が動作を開始する
2. MCLK が MCLK のトグルを開始
3. SPI フラッシュメモリからのプリアンプル入力が遅れる
4. MachXO2 がタイムアウトして起動に失敗する

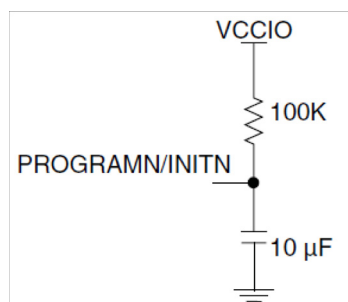
こうした状況の回避策としては、SPI フラッシュメモリの POR 解除より MachXO2 の内部 POR 解除を遅くすることですが、その方法としては幾つか考えられます。

- ・ 外部コントローラで INITN を Low に保持する
- ・ 外部コントローラで PROGRAMN を Low に保持する
- ・ RC 時定数で INITN を遅延させる (外部コントローラが使用できない場合の推奨。図 14-11)
- ・ RC 時定数で PROGRAMN を遅延させる

(注) RC 回路を付加して PROGRAMN を遅延させる場合、その遅延量は大き過ぎないようにします。デバイスが正常に起動しない恐れがあります。

RC 遅延回路を採用する場合、ボードの浮遊容量も考慮に入れるようにします。

図 14-11. RC 遅延回路例



デュアルブート・コンフィグレーション・モードには上記 2 つのイメージファイルが必要です。1 つはほとんど更新されることがないゴールデン (フェイルセーフ) イメージで、外部 SPI メモリに格納されます。もう 1 つは定期的に更新されるプライマリ (ワーキング) イメージで、CFM に格納されます。1 つの Diamond プロジェクトを使用して、ゴールデン / プライマリの両コンフィグレーション・データファイルを作成できます。単一 Diamond プロジェクト内に、例えば **working** という名前のインプリメンテーション (Implementation、注: Diamond 固有の呼称) と、もう一つ **failsafe** という名前のインプリメンテーションを定義・構成し、それ

ぞれデータファイルを生成します。Diamond インプリメンテーションの使用の詳細については、Diamond のオンラインヘルプをお読みください。

デュアルブート・デザインを作成するには、次の設計制約を使用します。

制約	デュアルブート設定
CONFIGURATION	CFG CFG_EBRUFM CFGUFM
MASTER_SPLPORT	ENABLE
COMPRESS_CONFIG	ON OFF

Diamond では、内部フラッシュメモリに格納されるプライマリイメージ用の JEDEC ファイルが作成されます。外部 SPI フラッシュに格納されるゴールデンイメージ用には、BIT ファイルが作成されます。ゴールデンイメージは、外部 SPI フラッシュの開始アドレス 0x010000 に配置する必要があります。これは、コンフィグレーション・データをオフセット 0x000000 で格納する必要がある、シングルイメージの MSPI コンフィグレーション・モードと異なっています。

以下はデュアルブートモードを用いる場合の、外部・内部プログラミングにおけるそれぞれの推奨手順です。

オプション A : 外部フラッシュメモリのプログラミングにバックグランド・モードを用いる

1. フラッシュ・プログラミングモードを用いて内部フラッシュをプログラム。SPI ポートがイネーブルされ、かつパーシスタンス (persistent) がオンであること
2. バックグランド・モードで外部フラッシュをプログラム
3. リフレッシュ、または電源をオフしてオン

オプション B : 外部フラッシュメモリのプログラミングにオフライン・モードを用いる

1. (非バックグランド・プログラミングモードを用いて) 外部 SPI フラッシュをプログラム
2. (フラッシュ・プログラミングモードを用いて) 内部フラッシュをプログラム
3. リフレッシュ、または電源をオフしてオン

ユーザ MSPI コントローラを使用する場合に、MachXO2 がデュアルブート・モードを使用しないようにするには、MASTER_SPL_PORT 設計制約を EFB_USER に設定します。このようにすると、MSPI コンフィグレーション・ポートピンが予約され、デュアルブートが防止されます。

スレーブ SPI モード (SSPI)

MachXO2 には SSPI コンフィグレーション・ポートがあり、コンフィグレーション・ロジックが提供する機能にアクセスできます。CFM、UFM、及びフィーチャ行の再プログラミング、コンフィグレーション・ロジックブロック内のステータス / 制御レジスタのアクセスが可能です。オフライン動作と透過的動作のどちらを使用してもフラッシュメモリにアクセスできます。コンフィグレーション SRAM には直接アクセスできません。REFRESH コマンドを送出して、新しいフラッシュイメージを SRAM にロードする必要があります。

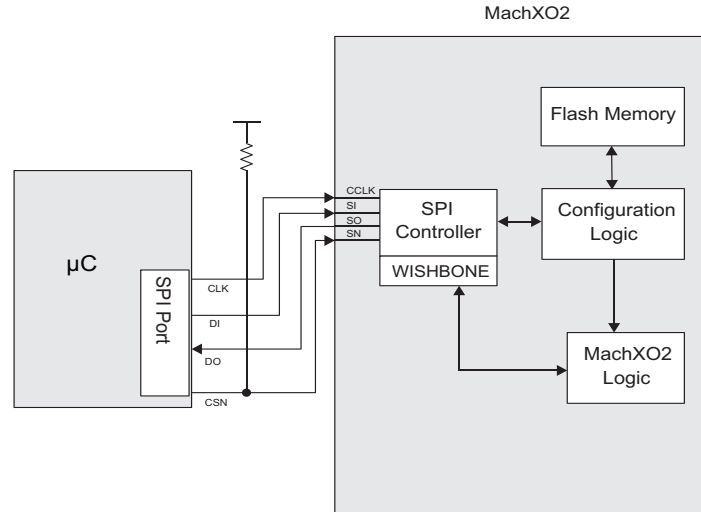
表 14-15. SSPI ポートピン

ピン名	記 述
CCLK	SPI マスタコントローラからドライブされるコンフィグレーション・クロック入力
SI	コマンド及びデータ用のコンフィグレーション・ロジックへのシリアルデータ入力
SO	コンフィグレーション・ロジックからのシリアルデータ出力
SN	コンフィグレーション・ロジックをイネーブルするチップセレクト

SSPI モードでは、MCLK/CCLK ピンは CCLK (コンフィグレーション・クロック) になります。SI ピンからの入力データは、CCLK の立ち上がりエッジで MachXO2 デバイスに読み込まれます。SO ピンの出力データは、CCLK の立ち下がりエッジで有効になります。SN はチップセレクト信号として機能します。SN が High

になると、SSPI インターフェイスが選択解除され、SO/SPIISO ピンがトライステートになります。SN が Low にドライブされると、MachXO2 へのコマンドの書き込みと、データの読み出しが可能になります。MachXO2 の SSPI ポートは、コンフィグレーション・ロジックへのモード 0 バス・トランザクションのみを受け付けます。

図 14-12. SSPI コンフィグレーション・モード



SSPI ポートがアクティブになるのは、MachXO2 がフィーチャ行 HW デフォルトモード状態（ブランク時や消去状態）の場合です。SLAVE_SPI_PORT に対する Diamond のデフォルト設計制約は、このポートをディセーブルするようになっています。Spreadsheet View（スプレッドシート・ビュー）を使用して、デザインの SLAVE_SPI_PORT 制約を ENABLE に設定し、ユーザモードで SSPI ポートがアクティブなままになるようにしてください。SSPI ポートが誤ってディセーブルされた場合に備えて、第二のプログラミングポートをアクティブ・ステートに保つことを推奨します。

SSPI ポートは、CFM、UFM、及びフィーチャ行の消去 / プログラム / ベリファイに使用されます。コンフィグレーション SRAM に直接アクセスする機能はありません。フィーチャ行が誤って消去されることを防止するため、SSPI ポートではフラッシュメモリの透過モードによる更新を推奨します。SSPI ポートで REFRESH コマンドを発行し、新たにプログラムされたイメージをアクティブにすることができます。REFRESH 動作は SN がネゲートされるまで開始されないため、REFRESH コマンドは、MachXO2 が外部またはデュアルブート・コンフィグレーション・モードのときに安全に使用できます。

SSPI ポートを使用した MachXO2 のプログラミングは複雑です。MachXO2 のプログラミングの複雑さを軽減するため、ラティスでは SSPIEmbedded と呼ばれる C プログラムをご用意しています。MachXO2 のフラッシュメモリを再プログラムするときは、SSPIEmbedded を使用することを推奨します。

フラッシュメモリの再プログラミング以外に、SSPI ポートは MachXO2 のいくつかのステータス及び制御レジスタへのアクセスに使用できます。そのレジスタで使用可能なコマンドのリストと説明は、“TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices (MachXO2 デバイスの UFM とハードマクロ化制御機能の使用法)” にあります。ステータス・レジスタへのアクセスはそれほど複雑ではなく、SSPIEmbedded プログラムを使用する必要はありません。

I²C コンフィグレーション・モード

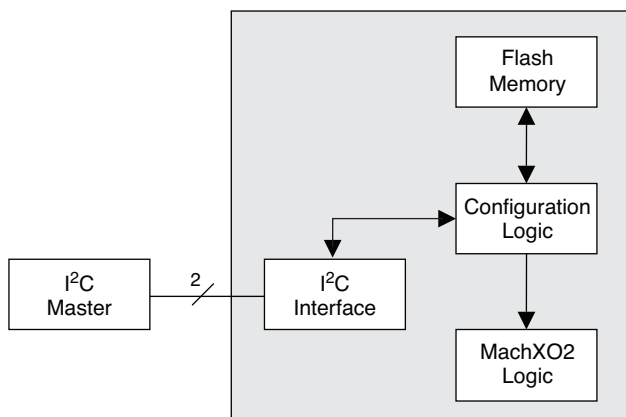
MachXO2 には、コンフィグレーション・ロジックのアクセスに使用する I²C コンフィグレーション・ポートがあります。I²C マスタは、10 ビットまたは 7 ビット・アドレッシングモードを使用して、コンフィグレーション・ロジックと通信できます。I²C SCL 入力、最高 400KHz のクロック周波数に対応できます。CFM、UFM、及びフィーチャ行の再プログラミング、コンフィグレーション・ロジックブロック内のステータス / 制御レジスタのアクセスが可能です。フラッシュメモリの再プログラミングは、オフライン動作と透過的動作のどちらでも実行できます。コンフィグレーション SRAM を直接更新することはできません。コンフィグ

レーション SRAM を更新するには、フラッシュメモリの再プログラミング後に、REFRESH コマンドを送信する必要があります。

表 14-16. I²C ポートピン

ピン名	記 述
SCL	I ² C バスクロック
SDA	I ² C バス・データライン

図 14-13. I²C コンフィグレーション・ロジック



I²C コンフィグレーション・ポートを使用できるのは、MachXO2 がフィーチャ行 HW デフォルトモード状態（ブランク時や消去状態）の場合です。Diamond デザインソフトウェアでは、I2C_PORT がデフォルト・ステートとして DISABLE ステートに設定されます。ユーザモードで I²C インターフェイスをアクティブなままにするには、I2C_PORT を ENABLE ステートに設定する必要があります。誤って I²C ポートがディセーブルされた場合に回復できるよう、第 2 のコンフィグレーション・ポート（JTAG）を使用できるようにしておくことを推奨します。

MachXO2 デバイスには、プライマリとセカンダリの 2 つのハードマクロ I²C コントローラがあります。プライマリ I²C コントローラは、MachXO2 コンフィグレーション・ロジックへのインターフェイスと、Wishbone レジスタへのアクセスを提供します。Wishbone レジスタへのアクセスはユーザモード I²C と呼ばれます。コンフィグレーション・ロジックへのアクセスを許可されているのは、プライマリ I²C コントローラだけです。セカンダリ I²C コントローラは、常にユーザモード I²C コントローラになります。

MachXO2 がフィーチャ行 HW デフォルトモード状態の場合 I²C ポートはイネーブルされ、プライマリ I²C コントローラと通信できます。I²C ポートがイネーブルされているときは、いつでもコンフィグレーション・ロジックにアクセスできます。ユーザモードでコンフィグレーション・ロジックにアクセスするためには、EFB をインスタンス化する必要があります。さらに、EFB の "wb_clk_i" 入力にはアクティブなクロック源に明示的に接続し、その周波数が I2C インターフェイス周波数の 7.5 倍より大きい 必要があります（I2C が 400kHz 動作の場合は >3.0MHz）。

プライマリ I²C コントローラは、次の場合にコンフィグレーション・ロジックへのアクセスができます。

- ・ MachXO2 がフィーチャ行 HW デフォルトモード・ステートになっている
- ・ EFB がインスタンスされている。"wb_clk_i" 入力にはアクティブなクロック源に明示的に接続され、周波数が I2C インターフェイス周波数の 7.5 倍より大きい。I²C ポートピンが ENABLE になっている

EFB I²C のベースアドレスが変更されている場合を除いて、外部 I²C マスタはアドレス 1000000（7 ビットモード）または 1111000000（10 ビットモード）を使用してコンフィグレーション・ロジックにアクセスします。プライマリ及びセカンダリ I²C コントローラが応答するアドレスを変更するには、Spreadsheet View ではな

く、IPexpress を使用します。アドレスを変更するには、EFB をインスタンス化する必要があります。このアドレスは、プライマリとセカンダリ I²C コントローラで共有されます。

表 14-17 に MachXO2 の両 I²C リソースへのアクセスに使用されるアドレスデコードを示します。

表 14-17. I²C ポートのスレーブアドレス

スレーブアドレス	I ² C 機能
yyyxxxxx00	プライマリ I ² C コントローラのコンフィグレーション・ロジックのアドレス。常に 7 ビットまたは 10 ビットアドレスで応答
yyyxxxxx01	ユーザモード・プライマリ I ² C コントローラのアドレス
yyyxxxxx10	ユーザモード・セカンダリ I ² C コントローラのアドレス
yyyxxxxx11	プライマリ I ² C コンフィグレーション・ロジックのリセット。常に 7 ビットまたは 10 ビットアドレスで応答

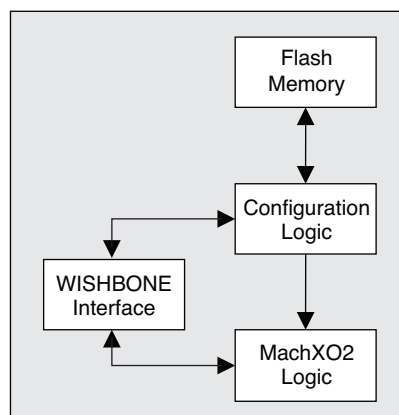
プライマリ I²C コアは UFM へのアクセスと CFM のプログラミングに使用できます。しかしプライマリ I²C コアは、単一のデザインでは UFM/CFM かユーザファンクションのどちらか一方にしか用いることはできません。ユーザモード・プライマリ及びユーザモード・セカンダリ I²C コントローラの動作については、“TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices (MachXO2 デバイスの UFM とハードマクロ化制御機能の使用法)” に説明があります。I²C スレーブデバイスとの通信については、本書で説明しません。

MachXO2 の以下に示す第 4 の I²C リソースはオフセット 3 にあります。インスタンスによっては、コンフィグレーション・ロジックへの I²C メモリ・トランザクションが中断されたり、放棄されたりすることがあります。コンフィグレーション・ロジックが受け付け可能なコマンドの中には、コンフィグレーション・ロジックがデータで応答するものがあります。I²C メモリ・トランザクションが中断または放棄された場合でも、コンフィグレーション・ロジックはキュー内のデータを返すことができます。新しく受信した I²C コマンドは、パディングバイトと見なされたり、誤って解釈されたりすることがあります。このステートを解消するには、任意の値をオフセット 3 に書き込みます。これで、コンフィグレーション・ロジックのコマンド・インタプリタがリセットされ、キュー内の全てのデータがフラッシュされて、その後のコンフィグレーション・ロジックへの I²C メモリ・トランザクションが正常に動作するようになります。

WISHBONE コンフィグレーション・モード

内部 WISHBONE バスから CFM、UFM、およびフィーチャ行にアクセスできます。WISHBONE バスを使用するには、デザインに EFB（組み込み機能ブロック）を組み込む必要があります。EFB とインターフェイスするロジックを設計してから、WISHBONE バス・トランザクションを実行して、コンフィグレーション・ロジックに接続されたリソースにアクセスします。

図 14-14. WISHBONE コンフィグレーション・モード



WISHBONE インターフェイスにアクセスするには、MachXO2 がユーザモードになっている必要があります。コンフィグレーション・ロジックによって利用可能となったリソースのアクセスと更新は、透過モードで行う必要があります。オフラインモードでコンフィグレーション・ロジックにアクセスしようとする、MachXO2 がユーザモードでなくなるため、デッドロックが発生します。

MachXO2 の WISHBONE インターフェイスの詳細については、“TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices (MachXO2 デバイスの UFM とハードマクロ化制御機能の使用法)” を参照してください。

JTAG モード

JTAG ポートは、MachXO2 で使用可能な最も柔軟なコンフィグレーション及びプログラミングポートです。JTAG では、以下が可能です。

- ・ オフラインのフラッシュメモリ・プログラミング
- ・ 透過的フラッシュメモリ・プログラミング
- ・ オフラインのコンフィグレーション SRAM のコンフィグレーション
- ・ MachXO2 コンフィグレーション・ロジックへのフルアクセス
- ・ デバイスのチェーン接続
- ・ IEEE 1149.1 テスト機能
- ・ IEEE 1532 準拠のプログラミング

JTAG ポートを使用できるのは、MachXO2 がフィーチャ行 HW デフォルトモード状態（ブランク時や消去状態）の場合です。Diamond 1.4 では、デフォルトでこのポートがイネーブルされます。MachXO2 の JTAG ポートピンは、IEEE 1149.1 TAP 機能の実行専用ではありません。JTAG ポートは、汎用 I/O としての使用が回復されることがあります。汎用 I/O として使用するための JTAG ピンの回復の詳細については、「sysCONFIG ピン」項を参照してください。

MachXO2 の JTAG ポートはその柔軟さから、価値のある機能となっています。システム及びデバイスのデバッグに、最も機能を発揮します。全ての MachXO2 デザインにおいて、JTAG ポートはアクセス可能なように保つことを推奨します。JTAG ポートをアクティブにしておくことの利点としては、次の点があります。

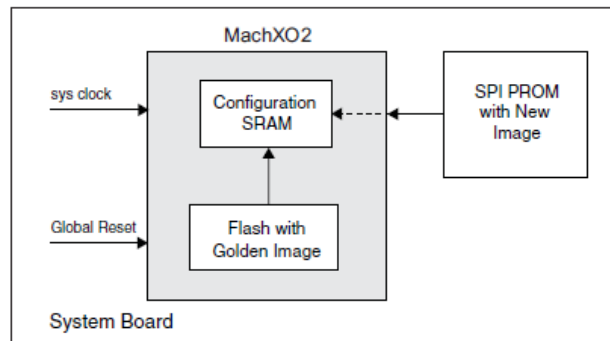
- ・ **マルチチェーン・アーキテクチャ**：JTAG ポートは、MachXO2 を他のプログラマブルロジック・チェーンと結合可能な、唯一のコンフィグレーション及びプログラミングポートです。
- ・ **Reveal によるデバッグ**：Lattice Reveal デバッグツールは、組み込み可能なロジックアナライザ・ツールです。外部ロジックアナライザでボードレベルのロジックを解析する場合と同様の方法で、MachXO2 内部のロジックを解析できます。Reveal には、MachXO2 の JTAG ポートからのみアクセスできます。
- ・ **SRAM リードバック**：JTAG ポートは、MachXO2 のコンフィグレーション SRAM に直接アクセス可能な唯一の sysCONFIG ポートです。SRAM ベースの FPGA の故障解析を行うために、直接アクセスが必要になることがしばしばあります。コンフィグレーション SRAM を読み出すと、故障解析のための主要コンポーネントを呼び出すことができます。JTAG ポートがイネーブルされていないと、この種の故障解析ができません。
- ・ **バウンダリスキャン・テスト機能**：IEEE 1149.1 JTAG を使用して行われるボードレベルの接続テストは、アセンブル済みのプリント基板の品質を保証するための重要な機能です。バウンダリスキャン・テストを行うためには、MachXO2 の JTAG ポートを確保することが不可欠です。ラティスのウェブサイトには、MachXO2 用のバウンダリスキャン記述言語ファイルが用意されています。

TransFR 動作

MachXO2 には他の Lattice FPGA と同様に、TransFR™ 機能があります。TransFR については、“TN1087 Minimizing System Interruption During Configuration Using TransFR Technology (TransFR 技術を使用したコン

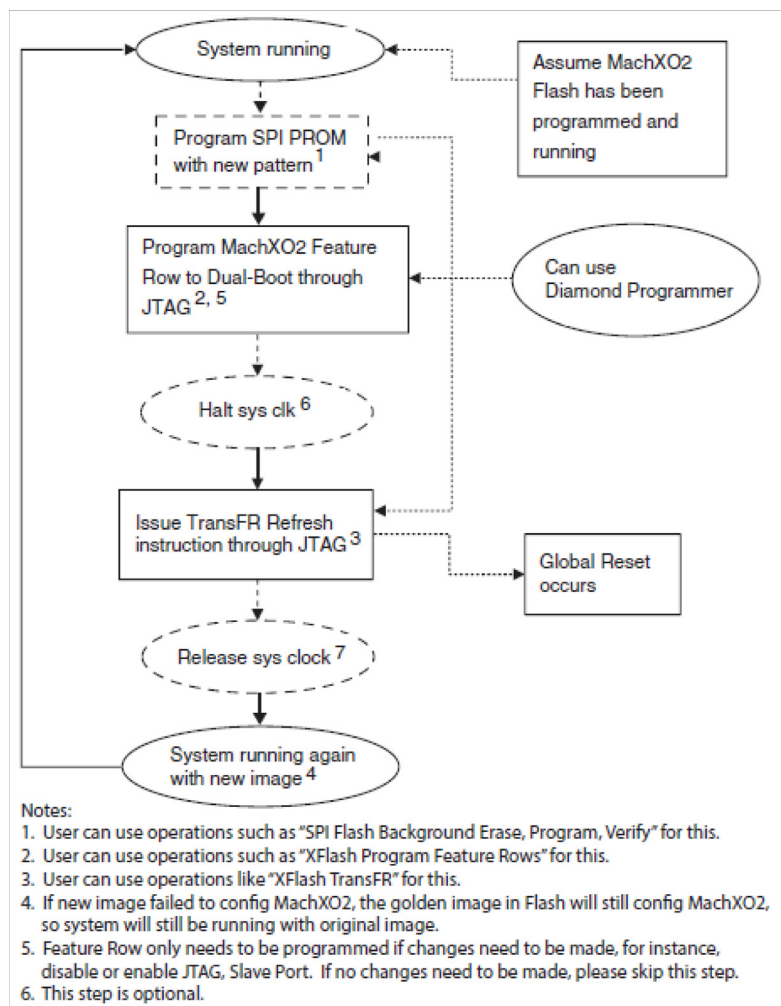
フィグレーション時のシステム中断の最小化) ” に説明があります。次に本機能を用いた MachXO2 のビットストリームを更新する例です。

図 14-15. *TansFR* を用いたビットストリームの更新



この例ではシステムの起動のために、ゴールデン・イメージがフラッシュメモリにストアされていて、システム動作に影響を与えずにイメージを外部 SPI フラッシュメモリからイメージをロードすることを想定しています。にその目的のための処理フローを示します。

図 14-16. プロセスフローの例



本フローを用いる場合の留意点は以下の通りです。

”リフレッシュ命令”が発行された後、デバイスのウェイクアップ手順で内部グローバルリセット (GSR) がアサートされます。従って、次のような I/O 設計を行う場合は注意が必要です。

- ・ レジスタ出力のピンの状態
- ・ 値が変わった際のシステム (ボード) への影響
- ・ 内部レジスタがセット / リセットされる影響

上記 I/O の場合、TransFR リフレッシュでは I/O ステートは変化しませんが、リフレッシュ完了後にユーザーモードに移行する時点では変化するかもしれません。そうした状況を回避するためには以下の事項を考慮するようにします。

- ・ クリティカルな I/O については GSR を適用しない
- ・ クリティカルな I/O で GSR を適用せざるを得ない場合、”GSR でセットまたはリセットする”オプションを用いる

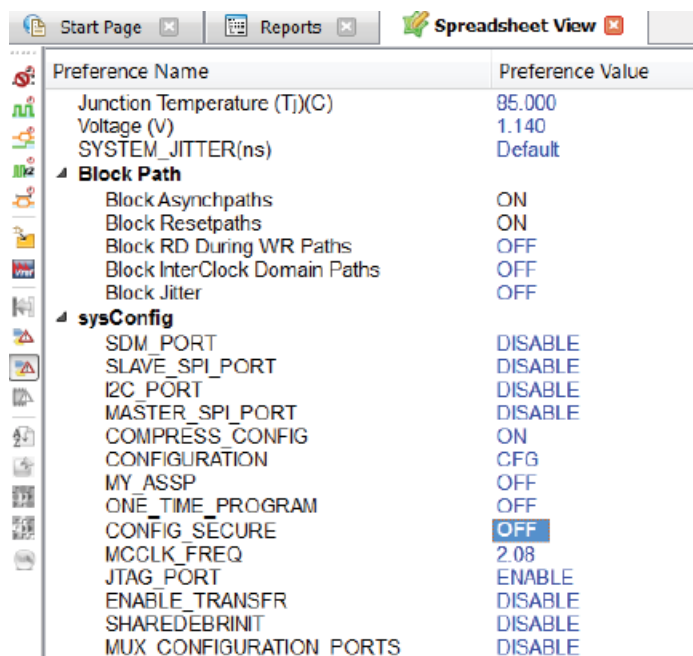
ソフトウェアで選択できるオプション

MachXO2 コンフィグレーション・ロジックの動作は、Diamond デザインソフトウェアで選択したオプションによって管理されます。他の FPGA では、コンフィグレーション・モードの選択用に専用 I/O が用意されています。MachXO2 では、不揮発性フィーチャ行を使用してコンフィグレーション方法が選択されます。フィーチャ行のデフォルト・ステートは、ほぼ全てのデザインごとに変更する必要があります。Diamond の Spreadsheet View (スプレッドシート・ビュー) を使用すると、MachXO2 のフィーチャ行の動作を変更でき、コンフィグレーション・ロジックの動作が変化します。

コンフィグレーション・ロジックの制約には、Spreadsheet View を使用します。Global Preferences (グローバル制約) タブをクリックし、sysCONFIG ツリーを確認してください。sysCONFIG セクションを図 14-17 に示します。sysCONFIG の制約は、次の 3 つのカテゴリに分類されています。

- ・ コンフィグレーション・モード及びポートに関するもの
- ・ ビットストリーム生成に関するもの
- ・ セキュリティに関するもの

図 14-17. Diamond Spreadsheet View の Global Preferences タブにある sysCONFIG 制約



Preference Name	Preference Value
Junction Temperature (Tj)(C)	85.000
Voltage (V)	1.140
SYSTEM_JITTER(ns)	Default
Block Path	
Block Asynchpaths	ON
Block Resetpaths	ON
Block RD During WR Paths	OFF
Block InterClock Domain Paths	OFF
Block Jitter	OFF
sysConfig	
SDM_PORT	DISABLE
SLAVE_SPI_PORT	DISABLE
I2C_PORT	DISABLE
MASTER_SPI_PORT	DISABLE
COMPRESS_CONFIG	ON
CONFIGURATION	CFG
MY_ASSP	OFF
ONE_TIME_PROGRAM	OFF
CONFIG_SECURE	OFF
MCCLK_FREQ	2.08
JTAG_PORT	ENABLE
ENABLE_TRANSFR	DISABLE
SHAREDDEBRINIT	DISABLE
MUX_CONFIGURATION_PORTS	DISABLE

コンフィグレーション・モード及びポートオプション

コンフィグレーション・モード及びポートオプションでは、MachXO2 デバイスがユーザモードになった後も引き続き動作するコンフィグレーション・ポートを、どれにするかを決定できます。使用可能なステータスピン、及びコンフィグレーション・データを外部 SPI フラッシュから読み取る速度も制御できます。ここで行った選択はフィーチャ行に保存され、フィーチャ行が消去されるまで有効なままになります。唯一の例外として、MCCLK_FREQ パラメータはコンフィグレーション・データに格納されます。

コンフィグレーション及びポートオプションは、任意の組み合わせで使用できます。

表 14-18. コンフィグレーション・モード / ポートオプション

オプション名	デフォルト設定	全設定値
JTAG_PORT	ENABLE	DISABLE, ENABLE
SLAVE_SPI_PORT	DISABLE	DISABLE, ENABLE
MASTER_SPI_PORT	DISABLE	DISABLE, ENABLE, EFB_USR
I2C_PORT	DISABLE	DISABLE, ENABLE
SDM_PORT ^{1, 2}	DISABLE	DISABLE, PROGRAMN, PROGRAMN_DONE, PROGRAMN_DONE_INITN
MCCLK_FREQ	2.08	(p.14-15 参照)
ENABLE_TRANSFR	DISABLE	DISABLE, ENABLE

1. ispLEVER 8.1 SP1 及び Diamond 1.1 では、SDM_PORT のデフォルトが PROGRAMN でした

2. 32 QFN パッケージには INITN ピンがありません。そのため、オプション SDM_PORT = PROGRAMN_DONE_INITN を使用できません

JTAG ポート

JTAG_PORT 制約では、MachXO2 デバイスがユーザモード時の JTAG コンフィグレーション・ポートピンの動作を決定できます。JTAG_PORT は、次の 2 つのステートに設定できます。

- ・ **ENABLE** ~ JTAG I/O が専用となり、IEEE 1149.1 JTAG インターフェイスを提供します
- ・ **DISABLE** ~ JTAGENB ピンを使用して JTAG I/O ピンが動的に制御されます

JTAGENB ピンを使用できるのは、JTAG_PORT が DISABLE ステートの場合だけです。JTAGENB が High にされると、4 本の JTAG I/O が IEEE 1149.1 JTAG ポートとして機能するようになります。JTAGENB が Low にされると、4 本の I/O を汎用 I/O として使用できるようになります。

再プログラミングによって MachXO2 でプライマリ・コンフィグレーション・ポートがディセーブルされても、JTAG ポートにアクセスできるようなデザインにすることを推奨します。

SSPI ポート

SLAVE_SPI_PORT では、MachXO2 デバイスがユーザモードになった後も、SSPI コンフィグレーション・ポートを確保できます。SLAVE_SPI_PORT 制約は、次の 2 つに設定できます。

- ・ **ENABLE** ~ MachXO2 デバイスがユーザモード時に SPI ポート I/O が確保され、外部 SPI マスタがコンフィグレーション・ロジックと通信できるようになります。この設計制約では、ポートピンに I/O を重複して割り当てることも防止されます
- ・ **DISABLE** ~ SPI ポートピンがコンフィグレーション・ロジックから切断されます。自動的にポートピンが汎用 I/O になることはありません。SPI ポートピンを汎用 I/O にするには、SLAVE_SPI_PORT と MASTER_SPI_PORT の両方を DISABLE ステートにする必要があります

SLAVE_SPI_PORT は、MASTER_SPI_PORT と同時にイネーブルできますが、内部 SPI マスタコントローラが、外部 SPI マスタと同時に SPI トランザクションを実行しないことを確実にする必要があります。2 つの SPI マスタの同時動作を必ず防止してください。

MSPI ポート

MASTER_SPI_PORT では、MachXO2 デバイスがユーザモードになった後も、SPI コンフィグレーション・ポートを確保できます。MASTER_SPI_PORT 制約は、次の 3 つに設定できます。

- ・ **ENABLE** ~ MachXO2 がユーザモード時に SPI ポート I/O が確保されます。この設計制約では、外部またはデュアルブート・コンフィグレーションモードがアクティブになります。この設計制約を CONFIGURATION = EXTERNAL と組み合わせて使用すると、外部ブートモードがイネーブルになり、また CFG、CFG_EBRUFM、CFGUFM と組み合わせると、デュアルブート・モードがイネーブルになります。ユーザモードになると、EFB 内の SPI コントローラが SPI ポートにアクセスし、SPI バス・トランザクションを実行できます。この設計制約では、ポートピンに I/O を重複して割り当てることも防止されます
- ・ **EFB_USER** ~ MachXO2 がユーザモード時に SPI ポート I/O が確保されます。ユーザモードになると、EFB 内の SPI コントローラが SPI ポートにアクセスし、SPI バス・トランザクションを実行できます。この設計制約では、ポートピンに I/O を重複して割り当てることも防止されます
- ・ **DISABLE** ~ SPI ポートピンがコンフィグレーション・ロジックから切断されます。自動的にポートピンが汎用 I/O になることはありません。SPI ポートピンを汎用 I/O にするには、SLAVE_SPI_PORT と MASTER_SPI_PORT の両方を DISABLE ステートにする必要があります

MASTER_SPI_PORT は SLAVE_SPI_PORT と同時にイネーブルできます。しかし内部 SPI マスタコントローラが、外部 SPI マスタと同時に SPI トランザクションを実行しないことを確実にする必要があります。2 つの SPI マスタの同時動作を必ず防止してください。

I²C ポート

I2C_PORT では MachXO2 デバイスがユーザモードになった後も、I²C コンフィグレーション・ポートを確保できます。I2C_PORT 制約は、次の 2 つに設定できます。

- ・ **ENABLE** ~ MachXO2 がユーザモード時に I²C ポート I/O が確保され、外部 I²C マスタがコンフィグレーション・ロジックと通信できるようになります。この設計制約では、ポートピンに I/O を重複して割り当てることも防止されます
- ・ **DISABLE** ~ I²C ポートピンがコンフィグレーション・ロジックから切断されます。このポートピンは汎用 I/O になります

EFB 内のプライマリ及びセカンダリ I²C コントローラを使用するには、I2C_PORT を ENABLE にする必要があります。

SDM ポート

SDM_PORT では MachXO2 デバイスがユーザモードになってから、プログラミング・ステータスピンを選択できます。SDM_PORT 制約は、次の 4 つに設定できます。

- ・ **DISABLE** ~ PROGRAMN、DONE、及び INITN ステータスピンが汎用 I/O になります
- ・ **PROGRAM** ~ MachXO2 デバイスがユーザモード時に PROGRAMN ピンが確保されます。このピンをアクティブ Low にすると、MachXO2 が再コンフィグレーションされます。DONE 及び INITN ピンは汎用 I/O になります
- ・ **PROGRAM_DONE** ~ MachXO2 デバイスがユーザモードになっても PROGRAMN 及び DONE ピンが確保されます。INITN は汎用 I/O になります
- ・ **PROGRAM_DONE_INITN** ~ ユーザモード時に PROGRAM、DONE、及び INITN が確保されます

MSPI またはデュアルブート・コンフィグレーション・モードを使用する場合は、SDM_PORT を PROGRAMN に設定することを推奨します。別のコンフィグレーション・ポートを使用して REFRESH コマンドを送出可能な場合を除いて、PROGRAMN ピンは、MachXO2 デバイスを "ウォーム" 再コンフィグレーションする唯一の手段です。

MCCLK の周波数

MCLK_FREQ 制約では、EXTERNAL またはデュアルブート・コンフィグレーション・モード使用時に、外部 SPI フラッシュからのデータ取得に使用される MCLK 周波数を変更できます。MachXO2 では、外部 SPI フラッシュからのデータ取得開始時に、公称 2.08MHz (+/- 5.5%) のクロック周波数が使用されます。MCLK_FREQ の値は、受信するコンフィグレーション・データに格納されていて、フィーチャ行には格納されていません。MachXO2 デバイスは、連続したパディングビット、データ開始ワード (0xBDB3)、及びコントロールレジスタ値を読み出します。制御レジスタには新しい MCLK_FREQ 値が格納されます。MachXO2 は MCLK_FREQ 値を受信するとすぐに、新しいクロック周波数に切り替わります。MCLK_FREQ で使用可能な周波数範囲は、2.08MHz から最高 133 MHz までです (表 14-11 を参照)。実際の SPI フラッシュまたはプリント基板の最高クロックレートを超えないよう注意してください。

仕様を外れるクロック周波数を指定した場合に備えて、バックアップ・コンフィグレーション・ポートを用意しておくことを推奨します。

ENABLE_TRANSFR

MachXO2 で使用される TransFR 機能では、コンフィグレーション SRAM にロードされたコンフィグレーション・データ、及び内部フラッシュメモリにロードされた将来のコンフィグレーション・データファイルで、ENABLE_TRANSFR が ENABLE ステートに設定されている必要があります。MachXO2 での TransFR の使用の詳細については、"TransFR の動作" 項、及び "TN1087 Minimizing System Interruption During Configuration Using TransFR Technology (TransFR 技術を使用したコンフィグレーション時のシステム中断の最小化)" を参照してください。

ビットストリーム生成オプション

Bitstream Generation (ビットストリーム生成) オプションでは、Diamond 開発ツールで MachXO2 デバイス用のコンフィグレーション・データを作成する方法を決定できます。CONFIGURATION、USERCODE、CUSTOM_IDCODE、及び SHAREDEBRINIT の各設定はフィーチャ行に保存され、フィーチャ行が消去されるまで有効なままになります。その他のオプションでは、Diamond で生成される JEDEC 及び BIT ファイルを制御できます。表 14-17 はそれらをまとめたものです。

COMPRESS_CONFIG

COMPRESS_CONFIG 制約は JEDEC 及び BIT ファイルの生成方法を変更します。COMPRESS_CONFIG のデフォルト設定は ON です。

JEDEC ファイルは生成時に必ず圧縮されます。外部 SPI フラッシュからコンフィグレーション・データを読み込むと、コンフィグレーション時間を多少短縮でき、Diamond ツールでは内部フラッシュメモリにプログラム可能な JEDEC ファイルが作成されます。

CONFIGURATION

CONFIGURATION 制約では CFM 及び UFM セクタを制御できます。CONFIGURATION 制約には、次の 4 つの設定が可能です。

- ・ **CFG** ～コンフィグレーション・データ作成時のデフォルトモードです。コンフィグレーション・ビットストリームは CFM に格納され、UFM セクタへのオーバーフローは許可されません。コンフィグレーション・データには EBR 初期化データが含まれます。UFM セクタは、ユーザモードで汎用フラッシュメモリとして使用できます。
- ・ **CFG_EBRUFM** ～ この設計制約で作成されるコンフィグレーション・データは、CFM に格納されます。EBR 初期化データは、UFM セクタの最下位ページアドレスに格納されます。UFM セクタはユーザモードで使用できます。正しい動作を保証するため UFM に変更を加えた場合は、EBR 初期化データを回復する必要があります。
- ・ **CFGUFM** ～ この設計制約で作成されるコンフィグレーション・データは、CFM に格納されます。このモードは、UFM へのオーバーフローが許可されるという点が CFG と異なっています。デザインに EBR 初期化データが追加されると、コンフィグレーション・データのサイズが増加します。
- ・ **EXTERNAL** ～ この設計制約で生成されるコンフィグレーション・データは、外部メモリに格納されます。UFM セクタは、ユーザモードで汎用フラッシュメモリとして使用できます。

Diamond ソフトウェアのリリース 1.4 では、CONFIGURATION 制約のデフォルトが CFG ステートです。Diamond デザインソフトウェアで JEDEC ファイルが生成されるのは、デザイン全体が CFMCFM に収まる場合だけです。MachXO2 デバイスがユーザモードになると、UFM を使用できることが保証されます。

コンフィグレーション・データが CFM に収まらない場合は、CFG_EBRUFM 制約を使用してみてください。デザインに多数の初期化する EBR があり、しかも UFM ページにアクセスしてデータを格納する場合は、この設計制約が有効です。初期化する EBR の数次第では、UFM にデータ格納用の十分な容量が残っています。

CFM にコンフィグレーション・データを格納できるだけのサイズがなく、独自データの格納に UFM を使用する必要がない場合は、CFGUFM オプションを使用してください。まれなケースとして、CFM と UFM の合計よりも、コンフィグレーション・データのサイズが大きくなる場合があります。

MSPI コンフィグレーション・モードで使用するコンフィグレーション・データの作成には、EXTERNAL 制約を使用してください。CFM と UFM で使用可能な容量の合計よりも、コンフィグレーション・データが大きい場合は、EXTERNAL モードに切り替える必要があります。EXTERNAL モードでは、CFM または UFM リソースがまったく使用されません。ユーザモードで UFM を使用できます。

MachXO2-256 デバイスには UFM が内蔵されていません。このデバイスで使用可能なコンフィグレーション・オプションは、CFG モードと EXTERNAL モードだけです。

USERCODE

MachXO2 コンフィグレーション・フラッシュセクターにはユーザ定義の値を格納するための 32 ビットレジスタが内蔵されています。このレジスタに格納されるデフォルト値は 0x00000000 です。USERCODE 制約を使用すると、このレジスタに任意の値を割り当てることができます。推奨する用途としては、コンフィグレーション・データのバージョン番号、製造元 ID コード、組み立ての日付、JEDEC ファイルのチェックサムなどがあります。

USERCODE フィールドの形式は、USERCODE_FORMAT 制約を使用して制御されます。データ入力は、2 進、16 進、ASCII のいずれの形式でも可能です。

USERCODE_FORMAT

USERCODE_FORMAT 制約では USERCODE 制約で値の割り当てに使用するデータフィールドの形式を選択し、次の 3 つのオプションがあります。

- ・ **Binary** (2 進) ～ 32 個の '1' または '0' 文字を使用して、USERCODE を設定します
- ・ **Hex** (16 進) ～ 8 個の 16 進数 (0 ～ 9、A ～ F) を使用して、USERCODE を設定します
- ・ **ASCII** ～ 4 個の ASCII 文字を使用して、USERCODE を設定します

CUSTOM_IDCODE

CUSTOM_IDCODE 制約はフィーチャ行内にある 32 ビットレジスタの割り当てに使用されます。CUSTOM_IDCODE フィールドがアクティブとなるのは、MY_ASSP 制約が ON ステートの場合だけです。割り当てる値は、CUSTOM_IDCODE_FORMAT 制約に従って、2 進または 16 進で入力できます。CUSTOM_IDCODE 制約への値の割り当ての詳細については、「MY_ASSP」の項を参照してください。

CUSTOM_IDCODE_FORMAT

CUSTOM_IDCODE_FORMAT 制約では CUSTOM_IDCODE 制約での値の割り当てに使用するデータの形式を選択し、次の 2 つのオプションがあります。s:

- ・ **Binary** (2 進) ～ 32 個の '1' または '0' 文字を使用して、CUSTOM_IDCODE を設定します
- ・ **Hex** (16 進) ～ 8 個の 16 進数 (0 ～ 9、A ～ F) を使用して、CUSTOM_IDCODE を設定します

SHAREDEBRINIT

に設定すると、この設計制約によって固有のメモリ初期化ファイルを 1 つ、フラッシュメモリに格納できます。この初期化値は複数の EBR 間で共有できます。そうすることで、デザインのビットストリーム・サイズが縮小され、UFM 容量が節約されて、他のアプリケーションに利用できるようになります。

MUX_CONFIGURATION_PORTS

MUX_CONFIGURATION_PORTS は全コンフィグレーション・ポートがディセーブルされている場合に用いられます。全てディセーブルにすることは、MachXO2 が「ライトワンス」デバイスになることを意味します。MUX_CONFIGURATION_PORTS は全てのコンフィグレーション・ポートを無くすることを確実にします。制御は全コンフィグレーション・ポートが DISABLE 状態になった場合にのみアクティブになります。MUX_CONFIGURATION_PORTS を ENABLE 状態にすると JTAGENB 入力ピンが有効になり、JTAG ポートピンが多重化されます。MUX_CONFIGURATION_PORTS を DISABLE 状態にすると、Diamond がビルドする際に全コンフィグレーション・ポートを無くすることとなり、MachXO2 がライトワンス・デバイスになることができます。

セキュリティオプション

セキュリティオプションでは、MachXO2 デバイスのトラッキングとセキュリティ保護用の様々なオプションを選択できます。表 14-19 はそれらをまとめたものです。

表 14-19. セキュリティオプション

オプション名	デフォルト設定	全設定値
TRACEID	<all zero>	8-bit arbitrary
MY_ASSP	OFF	OFF, ON
CONFIG_SECURE	OFF	OFF, ON
ONE_TIME_PROGRAM	OFF	OFF, FLASH, FLASH_UFM, FLASH_UFM_SRAM

TRACEID

MachXO2 では TraceID と呼ばれる新機能が導入されました。TraceID では、MachXO2 ごとに固有の 64 ビット ID が付けられます。2 つの MachXO2 デバイスに同じコンフィグレーション・データがロードされても、任意の 2 デバイスが同じ TraceID 値を持つことはありません。この点がコンフィグレーション・データ内にある USERCODE とは異なっています。USERCODE を使用してコンフィグレーション・データを受け取る各デバイスは、同じ USERCODE 値を持ちます。

TraceID は長さが 64 ビットで、下位 56 ビットは変更不可のデータとなっています。この 56 ビットは、ウェハのロット、ウェハ番号、及びウェハ上のダイ位置を示す X/Y 座標を組み合わせたものです。最上位の 8 ビットはユーザが指定し、フィーチャ行に格納されます。TraceID は、Diamond の Spreadsheet View を使用して変更します。TraceID フィールドに固有の 8 ビット 2 進値を入力し、コンフィグレーション・データを生成してください。

TraceID 機能の詳細については、“TN1207, Using TraceID in MachXO2 Devices (MachXO2 の TraceID の使用)” をご参照ください。

MY_ASSP

ラティスの全てのデバイスは、デバイスファミリ、デバイス規模、及びその他のパラメータ（電圧、デバイスステッピングなど）を識別する固有の ID コードを持っています。このコードには、MachXO2 のコンフィグレーション・ポートからアクセスできます。IDCODE レジスタに格納された値からは、ラティスのデバイスを一意に識別できます。

MY_ASSP 制約では、MachXO2 から IDCODE コードを読み取った場合に返される値を変更できます。MY_ASSP 制約を ON に設定すると、CUSTOM_IDCODE 制約がイネーブルされます。

CUSTOM_IDCODE

CUSTOM_IDCODE は MachXO2 デバイスのデフォルトの IDCODE を上書きするために割り当てる値です。MY_ASSP 制約が ON の場合に入力できるのは、32 ビットの 16 進または 2 進値だけです。

IDCODE を上書きすると、ラティスのプログラミング・ソフトウェアが MachXO2 デバイスを識別できなくなり、その結果 Programmer で MachXO2 デバイスを直接プログラムできなくなります。MY_ASSP がイネーブルされた MachXO2 デバイスをプログラムするには、SVF (Serial Vector Format) ファイルが生成されるようマイグレーションする必要があります。

CONFIG_SECURE

この設計制約を ON に設定すると、SRAM メモリ及び CFM のリードバックがブロックされます。UFM のリードバックは、ビットストリームが UFM ブロックにオーバーフローした場合もブロックされます。消去しない限り、MachXO2 デバイスをリードバックすることも、プログラムすることもできなくなります。セキュリ

ティ設定をリセットするには、デバイスを消去する必要があります。CONFIG_SECURE ヒューズと CFM は、連動して消去されます。セキュリティヒューズをリセットすると、デバイスを再びプログラムできます。

ONE_TIME_PROGRAM

MachXO2 には OTP (One Time Programmable、ワンタイム・プログラマブル) ヒューズがあり、オンチップメモリの消去またはプログラムの防止に使用できます。MachXO2 デバイスには、それぞれが SRAM、CFM、及び UFM の各メモリセクタに対応した 3 つの OTP があります。この設計制約によって、各メモリセクタに OTP セキュリティを設定することができます。

- ・ FLASH ~ CFM を消去またはプログラムできません
- ・ FLASH_UFM ~ CFM と UFM を消去またはプログラムできません
- ・ FLASH_UFM_SRAM ~ CFM、UFM、及び SRAM を消去またはプログラムできません

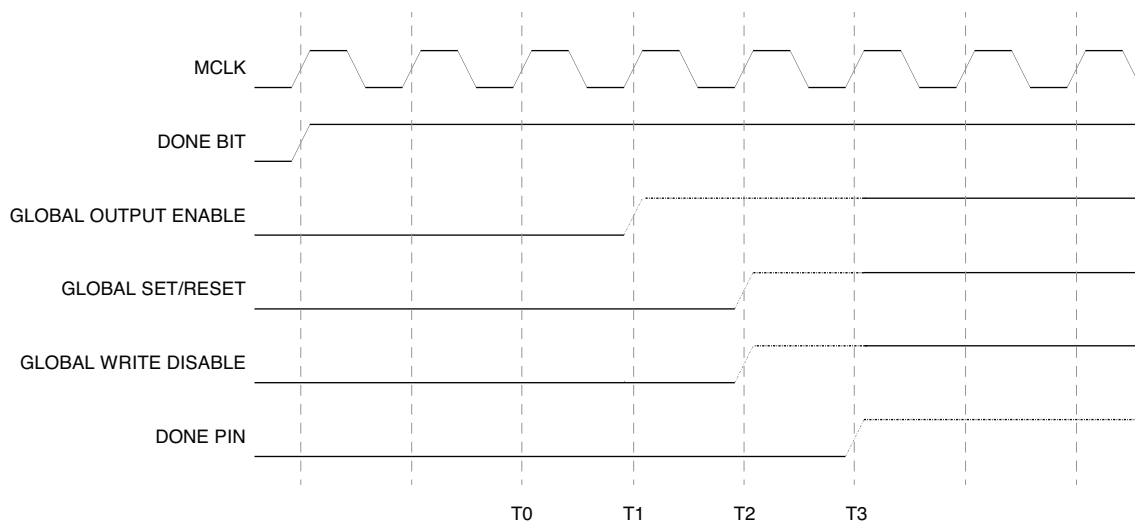
フラッシュメモリの ONE_TIME_PROGRAM 制約が設定されると、内部フラッシュメモリを消去またはプログラムできなくなります。コンフィグレーション・データの変更が防止されますが、SDM モードを使用してデバイスをコンフィグレーションすることは可能です。

ONE_TIME_PROGRAM 機能制約が FLASH_UFM_SRAM メモリに設定されると、デバイスは ASIC のような動作になります。内蔵フラッシュまたは UFM を再プログラムできなくなり、JTAG ポートから SRAM を変更できなくなります。その状態でも、内部フラッシュメモリまたは外部 SPI フラッシュからの SRAM のコンフィグレーションは可能です。

デバイスのウェイクアップ・シーケンス

コンフィグレーションが完了すると (SRAM がロードされると)、デバイスは予測可能な方法でウェイクアップします。MachXO2 デバイスがチェーン内の唯一のデバイス、またはチェーン内の最後のデバイスとなっている場合、コンフィグレーションの完了によってウェイクアップ・プロセスが起動される必要があります。コンフィグレーションが完了すると、内部 DONE ビットが設定され、ウェイクアップ・プロセスが開始されます。図 14-18 に、内部クロックを使用したウェイクアップ・シーケンスを示します。

図 14-18. 内部クロックを使用したウェイクアップ・シーケンス



ウェイクアップ信号

3つの内部信号 GSR、GWDIS、GOE によってウェイクアップ・シーケンスが決まります。

- ・ GSR はデバイスのコアの設定とリセットに使用されます。GSR はコンフィグレーション中にアサートされ (Low)、ウェイクアップ・シーケンスでネゲートされます (High)。
- ・ GWDIS 信号が Low のときは、デバイス内の RAM ブロックと LUT の整合性が保全されます。デバイスのウェイクアップ前、この信号は Low になっています。この制御信号は、デバイスのプライマリ入力ピンを制御しませんが、EBR 及び LUT の特定の制御ポートを制御します。
- ・ GOE 信号 が Low のときは、デバイスの I/O バッファがピンをドライブできなくなります。GOE は**出力ピンだけ**を制御します。内部 DONE ビットがアサートされると、MachXO2 は入力データに応答するようになります。
- ・ DONE ピンが High のときは、コンフィグレーションが完了し、エラーが検出されなかったことを示します。

ウェイクアップ・クロックの選択

ウェイクアップ・シーケンス中の 4 つのステート遷移を完了させるクロックソースは、ユーザが選択可能です。MachXO2 は、コンフィグレーションが終わるとウェイクアップ・ステートになり、コンフィグレーション・モードからユーザモードに移行します。このシーケンスはクロックソースと同期しますが、デフォルトのクロックソースは sysCONFIG 使用時に MCLK/CCLK、JTAG 使用時に TCK になります。

Verilog または VHDL 内で START マクロをインスタンス化することで、ウェイクアップで使用するクロックが変更できます。MachXO2 はウェイクアップ・シーケンスが完了するまで内部動作を開始しないため、このクロックは外部入力ピンから供給する必要があります。デバイスが最後の 4 つのステート遷移を実行可能になったことは、外部に示されませんので、フリーランクロック周波数を指定するか、デバイスがウェイクアップ可能になったことが保証されるまで待つしかありません。START マクロを使用することで、1 つまたは複数のプログラマブルデバイスのコンフィグレーションが完了してから、それらのデバイスを同期して起動するためのもう 1 つのメカニズムが得られます。

Verilog

```
module START (STARTCLK);  
  input  STARTCLK;  
endmodule  
  
START u1 (.STARTCLK(<clock_name>)) /* synthesis syn_noprune=1 */;
```

VHDL

```
COMPONENT START  
  PORT (  
    STARTCLK      :  IN STD_ULOGIC  
  );  
END COMPONENT;  
  
attribute syn_noprune: boolean ;  
attribute syn_noprune of START: component is true;  
  
begin  
  u1: START port map (STARTCLK =><clock name>);
```


コンフィグレーションの詳細について

フラッシュ・プログラミング

MachXO2 の内蔵フラッシュメモリは、このデバイスのコンフィグレーション・システムの心臓部です。柔軟で、デバイスのコンフィグレーション・データを格納でき、デザイン固有のデータを UFM に格納できます。正確な消去とプログラミングのシーケンスを必要とするリソースでもあります。ラティスでは、MachXO2 フラッシュメモリのプログラミング用にいくつかの方法を用意しています。

- ・ **Diamond Programmer** (プログラマ) : JTAG または SSPI のプログラミング
- ・ **VMEEmbedded** : JTAG ポートを制御する組み込みマイクロプロセッサで使用する 'C' ソースコード
- ・ **SSPIEmbedded** : SSPI ポートを制御する組み込みマイクロプロセッサで使用する 'C' ソースコード
- ・ **Custom** : この項で説明する内容、及び "TN1246, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide (MachXO2 デバイスの UFM とハードマクロ制御機能の使用法リファレンスガイド)" の説明に従うと、カスタム・ソリューションを作成できます。

フラッシュメモリ空間には、JTAG ポート、I²C ポート、SPI ポート、または WISHBONE バスからアクセスできます。これらのコンフィグレーション・ポートをオフラインまたは透過プログラミングモードで使用する、MachXO2 フラッシュメモリ・リソースを消去 / プログラム / ベリファイできます。WISHBONE インターフェイスでは、透過プログラミング動作のみを使用できます。コンフィグレーション・ロジックに渡すコマンドのシーケンスとタイミングは、全てのコンフィグレーション・ポートで同一です。コマンド及びデータを送信するときの通信プロトコル規格による違いは多少あります。最初に、全てのコンフィグレーション・ポートで共通のコマンドとタイミングフローについて説明します。その後、プロトコルの違いについて説明します。

それぞれの MachXO2 には、一定量の CFM と UFM が内蔵されています。このメモリサイズは MachXO2 のデバイス規模によって異なります。図 14-20 に各規模の MachXO2 デバイスで使用可能なフラッシュメモリのページ数を示します。各ページは 128 ビットのデータよりなります。

表 14-20. MachXO2 ファミリのフラッシュメモリのページ数

MachXO2 デバイス規模	CFM (ページ数)	UFM (ページ数)	CFG + UFM 連結 ¹ (使用可能なページ数)
7000	9,212	2,048	11,257
4000, 2000U	5,758	768	6,524
2000, 1200U	3,198	640	3,836
1200, 640U	2,175	512	2,686
640	1,151	192	1,342
256	575	0	575

1. デバイスの制限により、CFG+UFM (CONFIGURATION = CFGUFM) のページ数は各部の合計よりも小さくなることがあります

MachXO2 JEDEC の JEDEC ファイル形式

ラティスの全ての不揮発性デバイスは JEDEC ファイルに対応しています。開発ソフトウェアに含まれるユーティリティを使用すると、JEDEC ファイルを STAPL、SVF、ビットストリーム (16 進または 2 進) など他のプログラミングファイル形式に変換できます。JEDEC ファイルに関連する詳細情報について、次の表で詳しく説明します。

表 14-21. MachXO2 JEDEC ファイル形式

JEDEC フィールド	文法	記 述
Don't Care (ドントケア)	My design	^B (Control-B、0x02) 文字よりも前にある文字は無視。ここには、^B を除く全ての文字セットまたは内部言語を使用可能
Start-of-text (テキストの先頭)	^B	^B は JEDEC ファイルの開始を示す。^B の後ろは ASCII 文字のみ有効。文字 "*" は JEDEC フィールドの終わりを示すデリミタ。CR 及び LF は、JEDEC ファイルでは通常の空白として扱われ、デリミタ機能を持たない
Header (ヘッダ)	My design	最初のフィールドは、開始を示す識別子を持たないヘッダ。^B の後ろは ASCII 文字のみ有効。ヘッダはアスタリスク文字 "*" で終端される
Field Terminator (終端)	*	JEDEC ファイル内の各フィールドはアスタリスクで終端される
Note (コメント)	NOTE my design	キーワード N はコメントの開始を示す。JEDEC ファイル内の任意の位置に記述できる。ラティスの JEDEC ファイルでは NOTE という分かりやすい単語となっている
Fuse Count (フューズ数)	QF3627736	キーワード QF はデバイスの実際の合計ヒューズ数を示す ¹
Default Fuse State (デフォルト・フューズステート)	F0 or F1	キーワード F は、リンクフィールドに含まれていないヒューズのヒューズ状態を示す。F0 = ゼロ (0) で埋め、F1 = 1 で埋める。JEDEC ファイルサイズ削減のため定義されたもの。ラティスの JEDEC ファイルでは意味を持たない。ファイルサイズの削減には、圧縮の使用を推奨する
Security Setting (セキュリティ設定)	G0 or G1	セキュリティのプログラミング用に JEDEC 標準では G<0,1> が定義されている (0= なし、1= あり)
OTP and Security Setting (セキュリティ設定)	G0, G1, G2, or G3	ラティスでは OTP ヒューズのプログラミングにも対応するため、G フィールドを拡張した。G<0= どちらもなし、1= セキュリティのみあり、2=OTPのみあり、3= どちらもあり>
Link Field (リンクフィールド)	<pre> L0000000 101011...100011 111111...101100 110 101011...100011 111111...101100 110 101011...100011 111111...101100 110* NOTE SED_CRC* L3627704 111111...111111* CC1B9 </pre>	キーワード L は、空白の後に続くヒューズパターンの先頭ヒューズアドレスを示す。L キーワードに続く、ここに示された桁数は、QF フィールドと一致していること。この例では QF3627736 が 7 桁であるため、L00000000 には 7 個の 0 が必要。 従来、ヒューズアドレスは 0 からカウントしていた。 リンクフィールドにはプログラミング・パターンが格納され、JEDEC ファイルで最も重要な部分である。シリコンの物理アレイ・レイアウトが正確に再現されるよう、プログラミングデータがこのフィールドに書き込まれる。 行アドレスは、昇順で上から下に書き込まれる。 先頭 = 行 0、末尾 = 最終行 列アドレスは、昇順で左から右に書き込まれる。 左端 = ビット 0、右端 = 最終ビット INIT_ADDRESS コマンドでは行 0 が最初に選択される。プログラミングのためデバイスに最初にシフトインされるビットはビット 0。ベリファイ時にデバイスから最初にシフトアウトされるビットもビット 0。 CFM データの末尾は "NOTE END CONFIG DATA*" で示される。全て '0' 値を含んだページデータをプログラムする必要はない。 UFM ページが JEDEC 内にある場合は、その前に "NOTE TAG DATA*" 行が付く。 JEDEC ファイルが暗号化される場合は、リンクフィールド内の全データが暗号化される。列サイズパケット (パケットあたり 128 ビットまたは 16 バイト) の境界を示すようフィラービットを含めると、列サイズが増加する

表 14-21. MachXO2 JEDEC ファイル形式 (Continued)

JEDEC フィールド	文法	記 述
Fuse Checksum (フューズチェックサム)	CC1B9	ヒューズのチェックサム＝ヒューズ数。全てのヒューズのヒューズ状態はリンクフィールドで確認できる。リンクフィールドに指定されていない場合は、その位置にあるデフォルト・ヒューズ状態を使用する。JEDEC ファイルが暗号化される場合は、暗号化後にヒューズ・チェックサムが計算される。暗号化前のヒューズ・チェックサムは、いずれかのコメントにある
U Field (U フィールド)	UA Home	32 ビット USERCODE はここに格納される。32 ビット USERCODE は UA = ASCII、UH = ASCII Hex、U = Binary で表現できる。暗号化された JEDEC2 の CRC 値を格納できるよう、ラティスはこのフィールドを拡張した ²
E Field (E フィールド)	EH 012..ABCDEF	JEDEC 規格の規定では、このフィールドにアーキテクチャ・ヒューズが格納される。ラティスはこのフィールドを、フィーチャ行及び FEABITS の格納に使用する。フィーチャ行データは先頭行に置かれる。FEABITS 値は行 2 にある
End-of-text (テキスト末尾)	^C	^C (CTLG) は JEDEC ファイルの末尾を示す
Transmission Checksum (送信のチェックサム)	ABCD	^B から ^C までのファイル全体のチェックサム。チェックサム計算には、^B 及び ^C を含む全ての文字と空白が含まれる

1. 暗号化 JEDEC ファイルでは、USERCODE の最初の 16 ビットは行 0 のみから計算された CRC 値で、2 番目の 16 ビットが行 0 から最終行までで計算された CRC 値となります

MachXO2 の非暗号化 JEDEC ファイルの例を、図 14-19 に示します。

図 14-19. 非暗号化 JEDEC ファイルの例

[illegible]

[illegible]

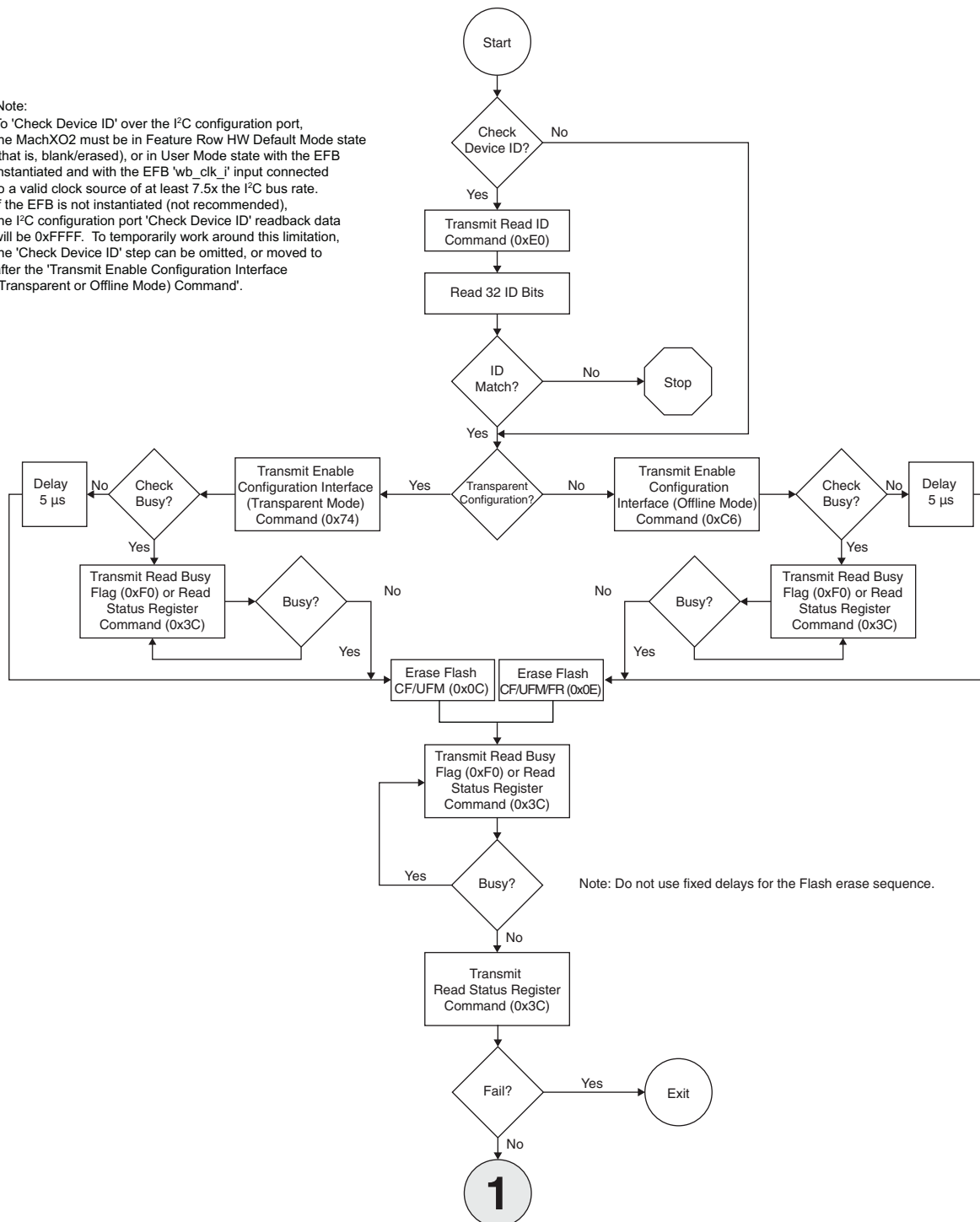
MachXO2 のフラッシュメモリ・プログラミングフロー

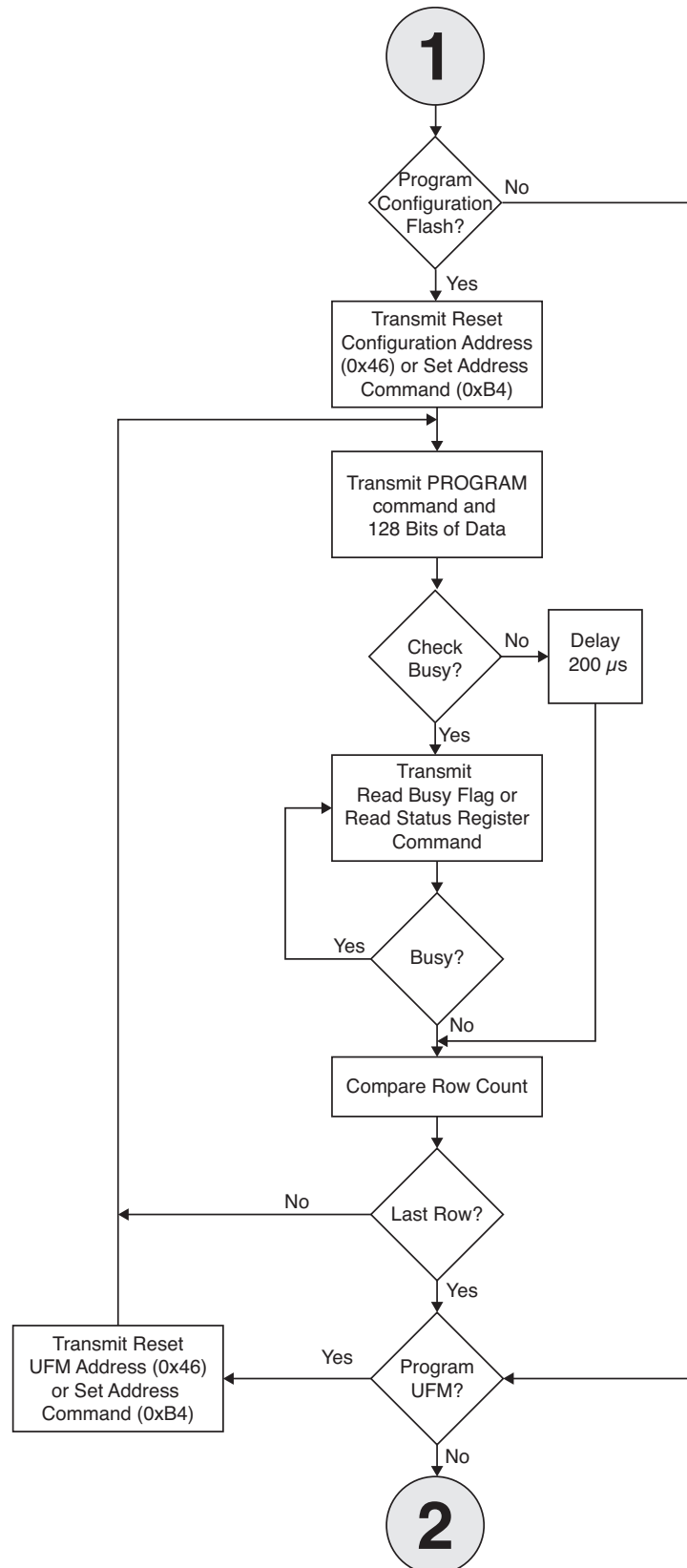
MachXO2 のフラッシュメモリの消去とプログラミングには、特定の一連のステップとタイミングが必要です。この項のフローチャートは、フラッシュを正しくプログラミングするために必要なコマンドシーケンスとタイミングを示しています。コマンドとタイミングは、全てのコンフィグレーション・ポートで共通です。タイミングとは異なり、使用するコンフィグレーション・ポートによっては、プロトコルに多少の違いがあります。例外については、コンフィグレーション・ポートごとの項で説明します。

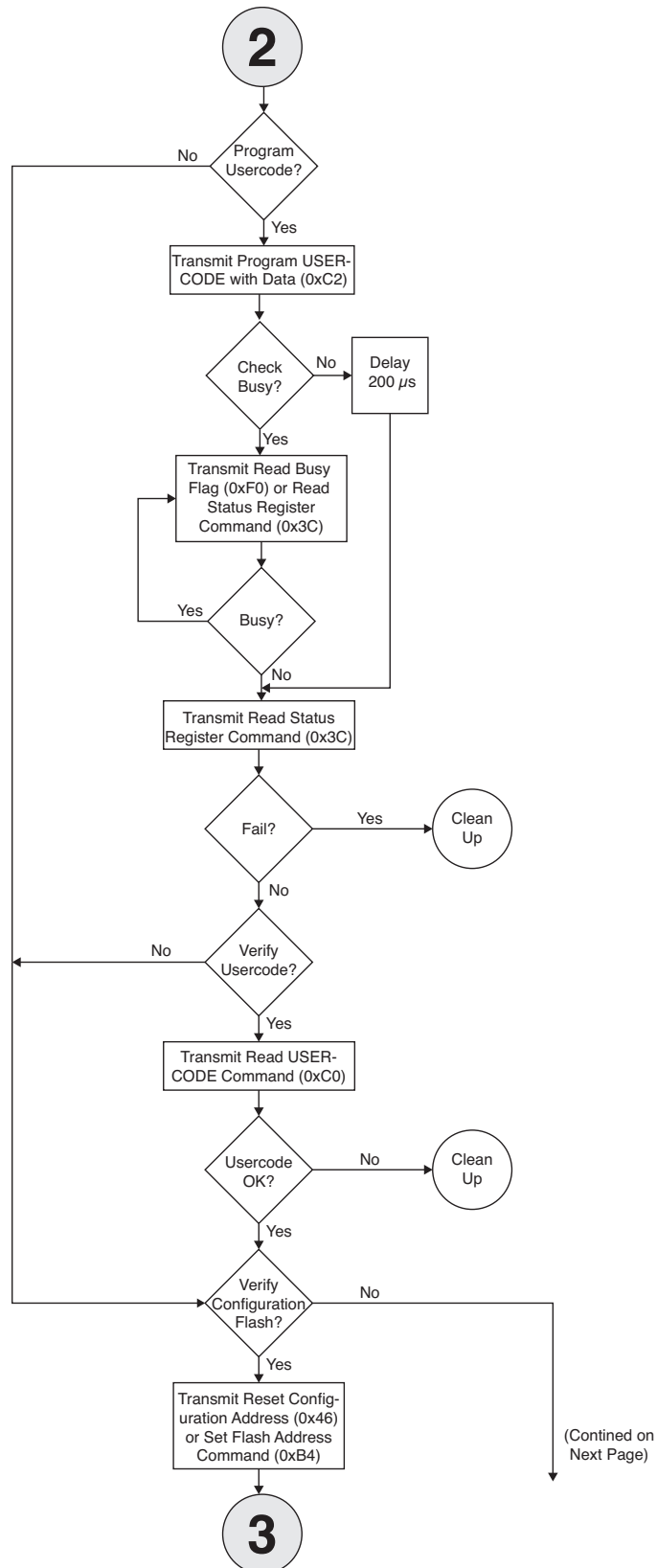
図 14-20. MachXO2 のフラッシュメモリ・プログラミングフロー

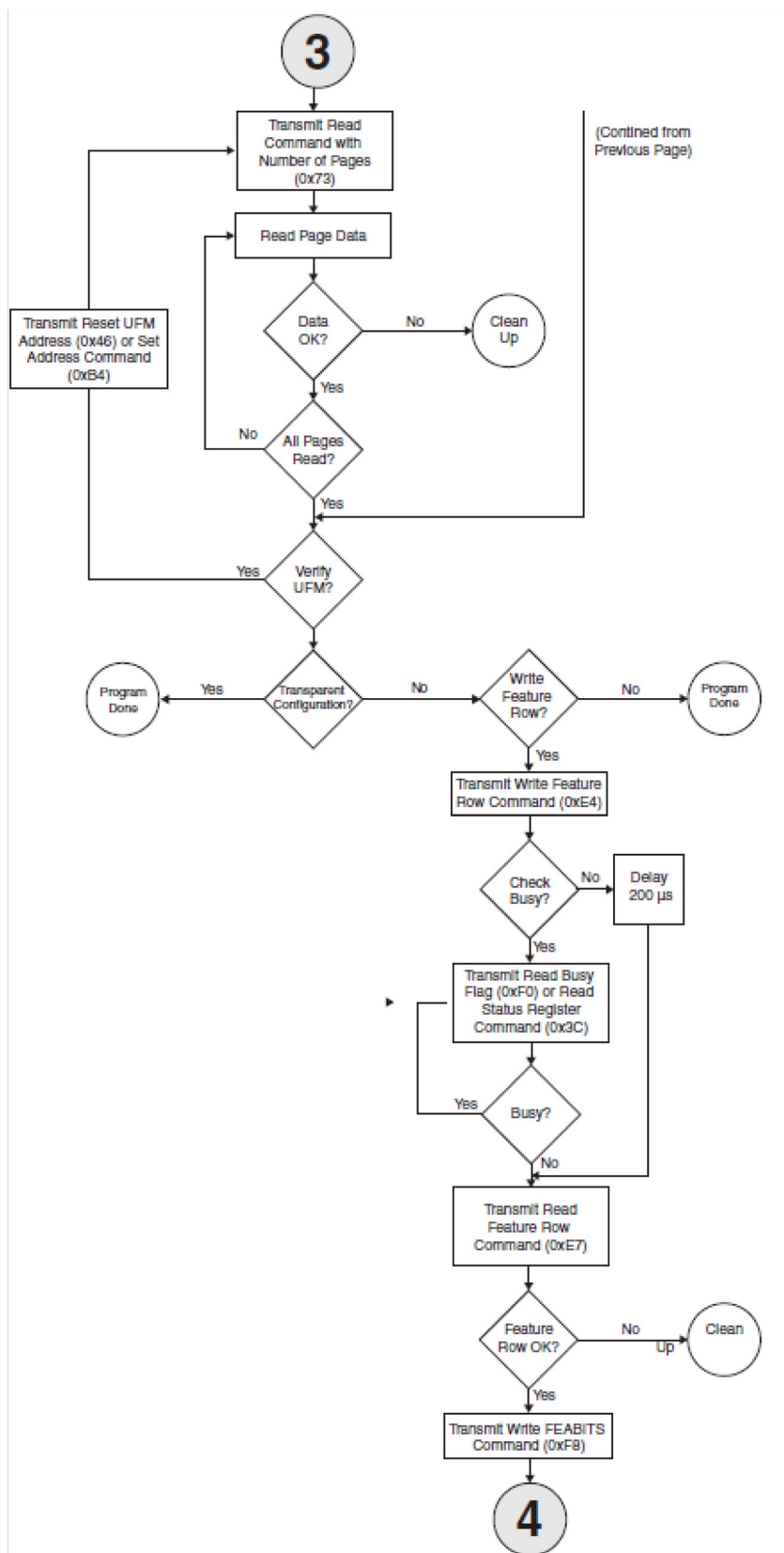
Note:

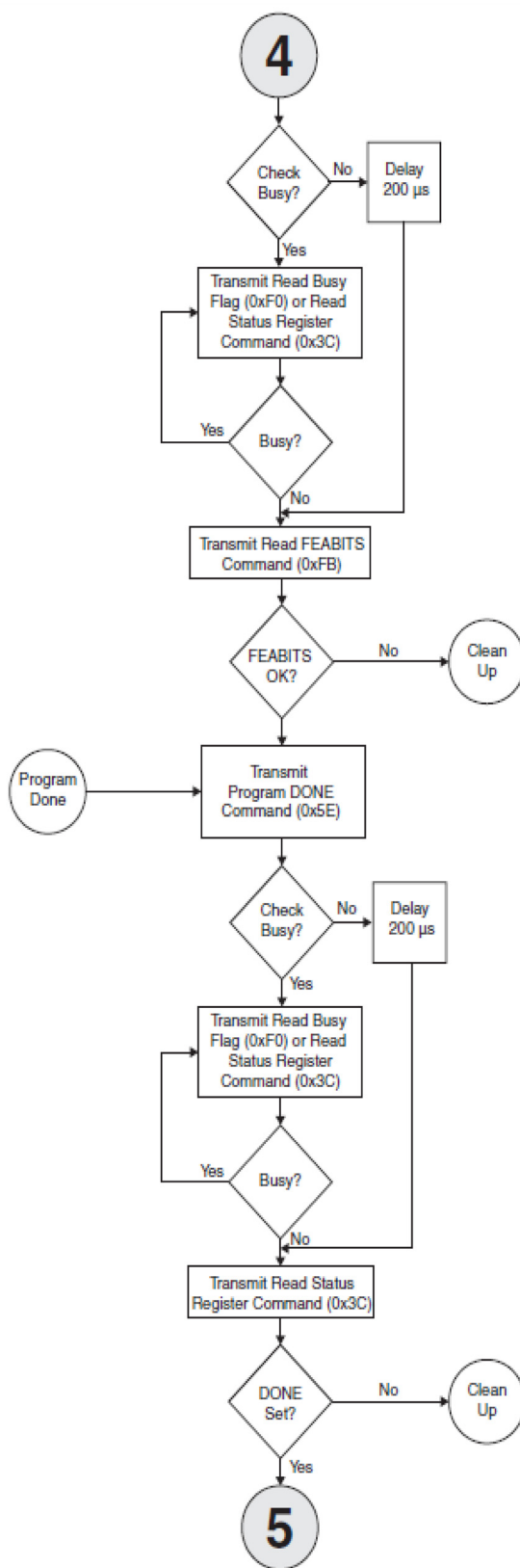
To 'Check Device ID' over the I²C configuration port, the MachXO2 must be in Feature Row HW Default Mode state (that is, blank/erased), or in User Mode state with the EFB instantiated and with the EFB 'wb_clk_i' input connected to a valid clock source of at least 7.5x the I²C bus rate. If the EFB is not instantiated (not recommended), the I²C configuration port 'Check Device ID' readback data will be 0xFFFF. To temporarily work around this limitation, the 'Check Device ID' step can be omitted, or moved to after the 'Transmit Enable Configuration Interface (Transparent or Offline Mode) Command'.

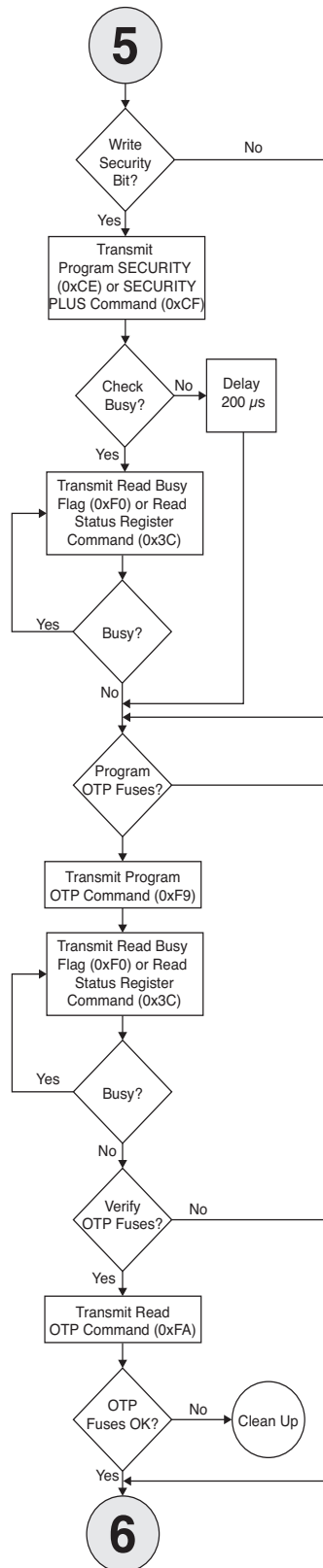


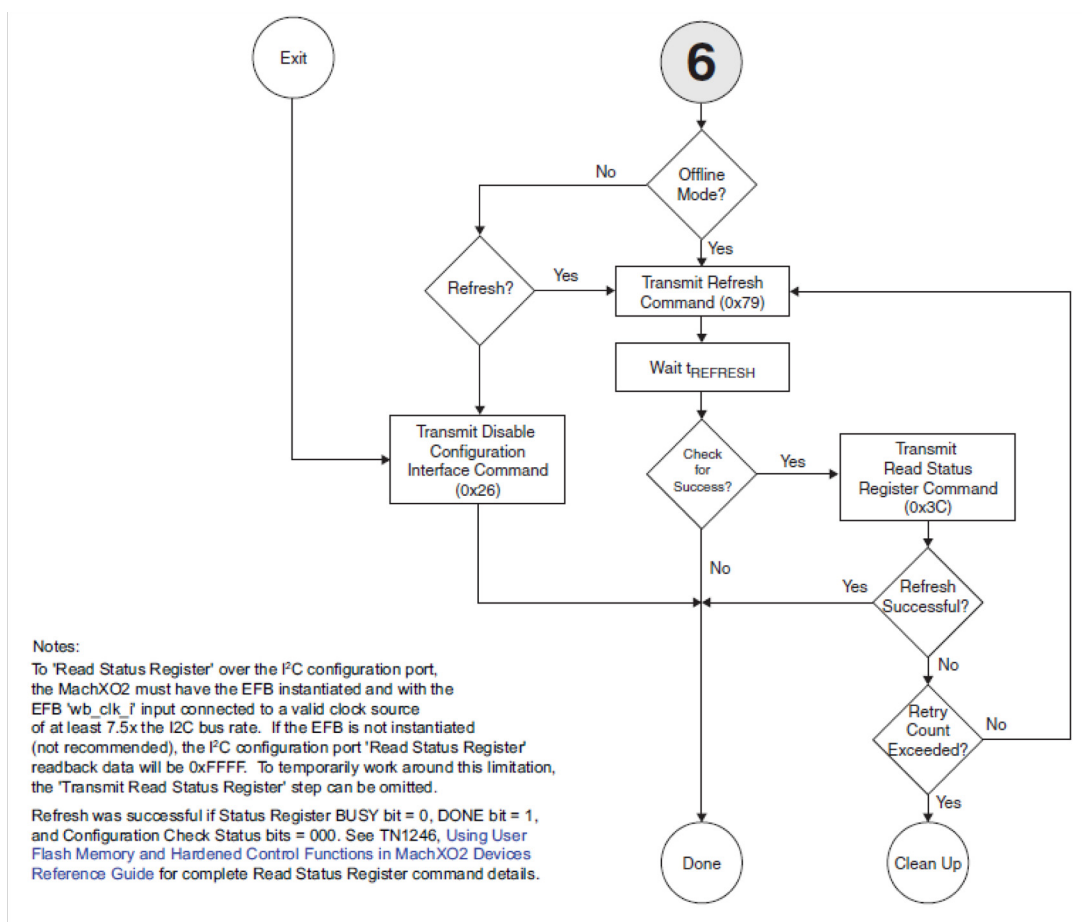












1. リフレッシュの成功 (BUSY = 0, DONE = 1, Configuration Check Status bits = 000)。リードステータス等詳細は [TN1246](#) を参照

MachXO2 のプログラミングコマンド

表 14-22. MachXO2 sysCONFIG のプログラミングコマンド

コマンド名 [SVF Synonym]	コマンド	オペランド	ライトデータ	リードデータ	注
Read Device ID [IDCODE_PUB]	0xE0	00 00 00	N/A	YY YY YY YY	YY 文字は、デバイス固有の ID コードを表す
Enable Configuration Interface (Transparent Mode) [ISC_ENABLE_X]	0x74	08 00 00 ¹	N/A	N/A	透過モードでのデバイス・プログラミング用にコン フィグレーション・ロジックをイネーブルする ¹
Enable Configuration Interface (Offline Mode) [ISC_ENABLE]	0xC6	08 00 00 ¹	N/A	N/A	オフラインモードでのデバイス・プログラミング 用にコンフィグレーション・ロジックをイネー ブルする
Read Busy Flag [LSC_CHECK_BUSY]	0xF0	00 00 00	N/A	YY	Bit 7 1 Busy 0 Ready
Read Status Register [LSC_READ_STATUS]	0x3C	00 00 00	N/A	YY YY YY YY	Bit 12 1 Busy 0 Ready 13 Fail OK
Erase [ISC_ERASE]	0x0E	0Y 00 00	N/A	N/A	Y = 消去するメモリ空間。Y はビット単位の OR (1 = イネーブル) Bit 16 SRAM の消去 Bit 17 フィーチャ行の消去 Bit 18 CFM の消去 Bit 19 UFM の消去
Erase UFM [LSC_ERASE_TAG]	0xCB	00 00 00	N/A	N/A	UFM セクタのみを消去する
Reset Configuration Flash Address [LSC_INIT_ADDRESS]	0x46	00 00 00	N/A	N/A	CFM セクタの開始位置へのページアドレス・ポ インタを設定する
Set Address [LSC_WRITE_ADDRESS]	0xB4	00 00 00	M0 00 PP PP	N/A	PP PP フィールドの下位 14 ビットで指定されたフ ラッシュページへのページアドレス・ポインタを 設定する。 'M' フィールドは、アクセス先のフラッシュメモリ 空間を定義する Field 0x0 0x4 M CFM UFM
Program Page [LSC_PROG_INCR_NV]	0x70	00 00 01	YY * 16	N/A	フラッシュの 1 ページをプログラムする。CFM ま たは UFM のプログラムに使用可能
Reset UFM Address [LSC_INIT_ADDR_UFM]	0x47	00 00 00	N/A	N/A	UFM セクタの開始位置へのページアドレス・ポ インタを設定する
Program UFM Page [LSC_PROG_TAG]	0xC9	00 00 01	YY * 16	N/A	UFM の 1 ページをプログラムする
Program USERCODE [ISC_PROGRAM_USERCODE]	0xC2	00 00 00	YY * 4	N/A	USERCODE をプログラムする
Read USERCODE [USERCODE]	0xC0	00 00 00	N/A	YY * 4	32 ビット USERCODE 値を取得する
Write Feature Row [LSC_PROG_FEATURE]	0xE4	00 00 00	YY * 8	N/A	フィーチャ行ビットをプログラムする
Read Feature Row [LSC_READ_FEATURE]	0xE7	00 00 00	N/A	YY * 8	フィーチャ行ビットを取得する
Write FEABITS [LSC_PROG_FEABITS]	0xF8	00 00 00	YY * 2	N/A	ビットをプログラムする
Read FEABITS [LSC_READ_FEABITS]	0xFB	00 00 00	N/A	YY * 2	ビットを取得する
Read Flash [LSC_READ_INCR_NV]	0x73	M0 PPPP	N/A	“フラッシュペ ージの読み出し” 項 を参照	PPPP カウントページを取得する。PP PP の下位 14 ビットのみが使用される。 'M' フィールドは、フラッシュメモリの読み出しに 使用するコンフィグレーション・ポートに応じて 設定すること 0x0 I ² C 0x1 JTAG/SSPI/WB

表 14-22. MachXO2 sysCONFIG のプログラミングコマンド (Continued)

コマンド名 [SVF Synonym]	コマンド	オペランド	ライトデータ	リードデータ	注
Read UFM Flash [LSC_READ_UFM]	0xCA	M0PPPP	N/A	“フラッシュページの読み出し”項を参照	PPPP カウントの UFM ページを取得する。ページカウントには PP PP の下位 14 ビットのみが使用される ‘M’ フィールドは、UFM の読み出しに使用するコンフィグレーション・ポートに応じて設定すること 0x0 I ² C 0x1 JTAG/SSPI/WB
Program DONE [ISC_PROGRAM_DONE]	0x5E	00 00 00	N/A	N/A	SDM をイネーブルする DONE ステータスビットをプログラムする
Program OTP Fuses [LSC_PROG_OTP]	0xF9	00 00 00	UCFSUCFS	N/A	選択したメモリ空間をワンタイム・プログラマブルにする。OTP 機能を有効にするには、対応するビットを連動してセットすること Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 CF OTP CF Writable 3, 7 UFM OTP UFM Writable
Read OTP Fuses [LSC_READ_OTP]	0xFA	00 00 00	N/A	UCFSUCFS	ワンタイム・プログラマブル (OTP) ヒューズの状態を読み出す Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 CF OTP CF Writable 3, 7 UFM OTP UFM Writable
Disable Configuration Interface [ISC_DISABLE]	0x26	0000	N/A	N/A	オフラインまたは透過プログラミングモードを終了する。このコマンドを送信すると、オフラインモード終了時に MachXO2 が再コンフィグレーションされる。Disable コンフィグレーション・インターフェイス・コマンドを送信する前に、コンフィグレーション SRAM をクリアしておくこと
Bypass [ISC_NOOP]	0xFF	FF FF FF	N/A	N/A	何も実行せず、デバイスをウェイクアップする
Refresh [LSC_REFRESH]	0x79	0000	N/A	N/A	MachXO2 を強制的に再コンフィグレーションする。REFRESH を送信すると、PROGRAMN をドライブした場合と同様に、MachXO2 が再コンフィグレーションされる
Program SECURITY [ISC_PROGRAM_SECURITY]	0xCE	00 00 00	N/A	N/A	セキュリティビット (CFG セクタを保護) をプログラムする ²
Program SECURITY PLUS [ISC_PROGRAM_SECPLUS]	0xCF	00 00 00	N/A	N/A	セキュリティプラス (Security Plus) ビットをプログラムする (CFM と UFM セクタを保護) ²
Read TraceID code [UIDCODE_PUB]	0x19	00 00 00	N/A	YY*8	R64 ビットの TraceID を読み出す

1. I²C ポート使用時は、コマンドのオペコードとオペランドの先頭 2 バイトを送信します。オペランドの最終バイトを送信してはいけません

2. SECURITY と SECURITY PLUS コマンドは相互に排他的です

フラッシュページの読み出し

CFM 及び UFM ページを読み出すには、そのための手順が必要です。CFM 及び UFM ページには、MachXO2 の任意のコンフィグレーション・ポートからアクセスできます。JTAG、SSPI、WISHBONE の各コンフィグレーション・ポートは、リード動作の実行に関して同一の動作となります。I²C ポートでは、修正されたアクセスプロトコルが必要です。ポートごとのデータフローの概要を、図 14-22 に示します。

リードプロセスの開始時には、リードフラッシュ / リード UFM フラッシュ・コマンドを送信する方法を、全てのコンフィグレーション・ポートが実行します。コンフィグレーション・ロジックがコマンドバイトを受け取ると、MachXO2 はリードプロセスを開始します。MachXO2 から返される最初のページは、ページアドレス・ポインタで決定されます。単一ページリード動作で、読み出される最初のページが有効であるためには、XO2-2000 とこれより大きい規模のデバイスでは 240ns 以上、XO2-1000 とこれより小規模デバイスでは 360ns 以上待たなければなりません (retrieval delay)。この時間の定義は図 14-21 のように、コマンドバイト

の終わりから第一オペランドバイトの終わりまでです。インタークロックの速度が遅い場合は、付加的な遅延を挿入する必要が無い程度に、時間が経過するかもしれません。

図 14-21. 単一ページリード時のタイミング遅延 (retrieval delay) 要件

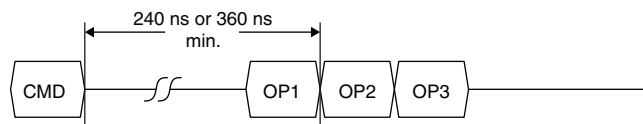
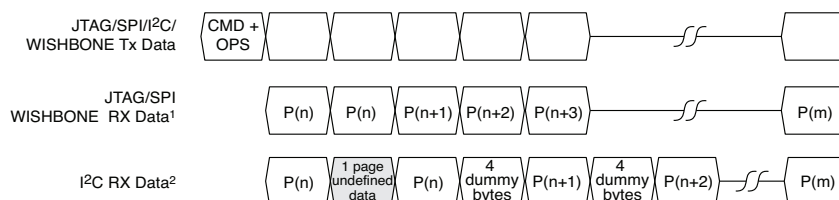


図 14-22. フラッシュページ・コマンドとデータシーケンス



Notes:

1. JTAG/SSPI must transmit data in order to read data back. The data sent by the JTAG/SSPI master is not specified (i.e. don't care).
 2. The I²C must use RESTART between sending the CMD and reading the data. (Issuing a STOP terminates a CMD and resets the I²C state machine.)
- CMD + OPS = Read Flash or Read UFM Flash command byte + 3 operand bytes.

図 14-22 は複数ページのリードシーケンスを示しています。リードページまたはリード UFM ページ・コマンドが MachXO2 に送信され、240ns の遅延後に有効なページデータが得られます。図 14-22 から分かるとおり、全てのインターフェイスで一定の遅延後すぐに、ページアドレス・ポインタ位置のページが返されます。全てのコンフィグレーション・ポートは、最初のページの最終バイトの転送直後に、単一ページリードを終了できます。I²C インターフェイスは、リードフラッシュ / リード UFM フラッシュのオペランドバイトのみが異なっています。

複数のページを読み取るには、特別な処理が必要です。複数ページリードでは、ページアドレス・ポインタで選択したページが複製されます。このような動作となるため、ページカウントは目的のページ数よりも 1 大きくする必要があります。例えば 2 ページを読み取るには、リードフラッシュ / リード UFM フラッシュ・コマンドに指定するページカウントに、値 3 を割り当てます。ページアドレス・ポインタが 0000 の場合、MachXO2 はページ 0、ページ 0、ページ 1 の計 3 ページを返します。

I²C インターフェイスには、フラッシュメモリ・ページの読み出し時にさらにオーバーヘッドがあります。図 14-17 を見ると、複数ページリード要求時にどのようにデータが提示されるかが分かります。ページカウントが 3 で、ページアドレス・ポインタが 0000 の場合、I²C インターフェイスはページ 0、未定義の 16 バイト、ページ 0、ダミーの 4 バイト、ページ 1 を返します。最後のダミー 4 バイトの読み出しはオプションです。

参照資料

- ・ MachXO2 Family Data Sheet
- ・ TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices
- ・ TN1207, Using TraceID in MachXO2 Devices

テクニカルサポート支援

e-mail: techsupport@latticesemi.com

インターネット : www.latticesemi.com

日本語版改訂履歴 (Ver.2.0 以降)

日付	バージョン	新ページ	改訂内容
2012 年 6 月	02.0	-	Ver.1.X から大幅・全面改訂。特に [1] プログラミング・アルゴリズムについて更新、[2] フィーチャ行の記述、[3] コンフィグレーション・ポート管理について追記
2012 年 7 月	02.1	46	図 14-17 MachXO2 フラッシュメモリ・プログラミングフローにて SECURITY/SECURITY PLUS について明記
		48	表 14-21 sysCONFIG プログラミング・コマンド追加
		50	図 14-18 単一リード時のタイミング要件追加、記述追加
			フラッシュメモリ・プログラミングフローにて 'Program DONE' ステップを追加
2012 年 9 月	02.2	11, 12	PROGRAMN ピン情報を更新、図 14-4 を追加
		28	TransFR 動作記述更新
		32, 34	MUX_CONFIGURATION_PORTS 記述を追加、表 14-16 に追記
		41 - 47	プログラミングフローを更新
2012 年 10 月	02.3	9	ページ中程の表に番号なし --> 表 14-4 付与、その後の表番全て +1
		15	表 14-9 タイトルに周波数単位追記
		26	表 14-15 の直後のパラグラフ先頭記述追記
2013 年 10 月	02.4 ~ 02.6	2	用語定義、内部フラッシュメモリを追加
		7 ~ 8	フィーチャ行記述、全面的に記述変更・更新。表 14-3 カラム追加、脚注 2 削除
		(19 カ所)	用語 " デフォルトモード・フィーチャ行ステート " を " フィーチャ行デフォルトモード・ステート " に変更
		16	CSSPIN 記述、第三パラグラフ追加
		32	CONFIGURATION 記述、CFGUFM 項、最終行削除
		47	フロー図にコメントを追加
		48	表 14-22 内、Set Address の Write Data を "M0 00 0P PP" から "M0 00 PP PP" に修正
		20	マスタ SPI モード記述、ブレット注記を追加。表 14-12、GENERATE_BITSTREAM 削除。図 14-9 MSPI コンフィグレーション・モード図、差し替え
		23	ページ上部表の内、GENERATE_BITSTREAM 削除、図 14-10 削除
		24	新図 14-10 SSPI コンフィグレーション・モード図、差し替え
		32	ビットストリーム生成オプション節、第一パラグラフ最終行と削除。表 14-17 削除
		32	GENERATE_BITSTREAM オプション記述、削除。COMPRESS_CONFIG オプション記述、更新。表 14-18 削除
		32	CONFIGURATION オプション記述、最終パラグラフの前のパラグラフ最終行 (GENERATE_BITSTREAM について) 削除
		10	表 14-6、脚注削除。なお、PROGRAMN ポートのユーザモード・デフォルト状態がオリジナル英語版では PROGRAMN とあるが、これは誤り。次版で元に戻し、User-defined I/O とする予定。(従って、日本語版更新せず)
		11	表 14-7 追加
		29	図 14-13 差し替え
2014 年 2 月	2.7-3.0	1 ~ 2	用語定義の項目順序を入れ替え
		8	フィーチャ行プログラミングの節、第三パラグラフ追加
		10	(英語版のみ) 表 14-6 PROGRAMN のデフォルト表記を更新
		11	PROGRAMN ピンの記述に第五ブレット追加
		15	SN ピンの記述に第五ブレット追加
		23	デュアルブート節、推奨オプション A / B 手順を追記
		44, 45, 47	図 14-16、ステップ 3 / 4 で FEABITS 名に訂正、ステップ 3 のコードを 0xFB に訂正。同ステップ 6 の Refresh フロー部記述を更新
		48, 49	表 14-21、Erase 注記 b16 追記、ISC_DISABLE 注記追加・更新
2014 年 7 月	3.1	6	「オンチップ・フラッシュメモリのプログラミング」節追加
		34	MUX_CONFIGURATION_PORTS 項、英語版に合わせて記述復活

日付	バージョン	新ページ	改訂内容
2014 年 9 月	3.2、3.3	5	最上のパラグラフに記述一行追加
		6, 7	オフライン / オンライン両モードのパラグラフに、それぞれ一行記述を追加
		8, 9	フィーチャ行記述一行追加、図 14-4 追加、表 14-3 追加、表 14-4 内上から 11 番目の機能で旧 "I2C Slave Address" 表記と記述を "I2C Programmable Primary...." として記述と共に更新
		—	用語 "フィーチャ行デフォルトモード・ステート" を "フィーチャ行デフォルトモード状態" に変更、一部には英語版にそって適宜 " (ブランク時や消去状態) " の注記を追加
		23, 24	デュアルブート項、POR 解除タイミングに関する記述追加、図 14-11 追加
2015 年 2 月	3.4	8	表 14-2 を更新
		13	PROGRAMN ピン記述、注意事項第二点について追記
		17	SN ピン記述、注意事項第一点について追記
		28	I2C コンフィグレーションモード記述、EFB インスタンスとクロックについての記述を更新・追加
		43, 46, 9	プログラミングフロー図、差し替え 3 ページ分
		51, 52	Retrieval Time についての記述を更新、図 14-19 を差し替え
2015 年 4 月	3.5、3.6	14	DONE ピン記述、第三パラグラフ内、Diamond バージョンの注記を追加
		16	SN ピン記述、注記第一ブレット、電源立ち上がり時の注意を追記
		24	POR 解除タイミングに関して、記述更新
		29	"JTAG モード" 項の直前のパラグラフ、" 付録 C、、、" 行を削除
		30-31	TransFR 項を更新、図 2 つ追加