

第 14 章 プログラマー

本章ではデバイスの書き込みツールであるプログラマー (Programmer) について記述します。Diamond 組み込みバージョンについての操作を基本として記述しています。スタンドアローン版では、操作や表示が全く同じではない場合があることに、ご注意ください。

14.1 基本的な書き込み手順

FPGA デバイスのプログラミング、またはコンフィグレーション (以下 ”書き込み ” という表記と併用の ”初回の ” 基本手順は次の通りです。ここでデバイスへの書き込み用ファイル (.bit / .jed、ビットストリーム、パターン、イメージなどとも表現される) は生成済みのものとします。

1. ダウンロード・ケーブルをターゲットボードに接続 (JTAG。MachXO2/3 などでは SPI / I2C も)
2. プログラマーの起動とケーブルの (指定・) 検出
3. デバイス・スキャン (デバイス検出)
4. デバイスへの書き込みファイルとアクセスモード、オペレーションを指定
5. 書き込み (設定したオペレーション) の実行
6. チェインファイル (プロジェクト・ファイル) XCF の保存

チェインファイル (*.xcf) を保存しておくこと、二回目以降同じ (JTAG チェイン) 構成の場合は、ステップ 2 の後に xcf を呼び出すことでステップ 3 と 4 は省略できます。チェインファイルはプログラマー・プロジェクトと呼ぶことがあり、(当該 JTAG チェインの) オペレーションに関する設定情報を全て含んでいます。

ここで、ラティス定義では ”プログラミング ” をオンチップ・フラッシュメモリーや SPI フラッシュメモリーに対する書き込み (.jed ファイル使用)、 ”コンフィグレーション ” をデバイス内部コンフィグレーション SRAM への書き込み (.bit ファイル使用) と呼び、書き込み対象によって異なる用語になっています。これはラティスのデータシートやテクニカルノート全般で共通です。

14.1.1 ダウンロード・ケーブルの接続

書き込みの準備として、まずダウンロード・ケーブルをターゲットボードに接続します。最新のダウンロード・ケーブルは USB 対応 HW-USBN-2B です (種類や詳細についてはユーザーガイド『UG48 Programming Cables』をご参照ください)。図 14-1 に USB ダウンロード・ケーブル (右) と、その前バージョンの USB ケーブル HW-USBN-2A (左) の外観図を示します。

図 14-1. USB ダウンロード・ケーブル (左 : HW-USBN-2A、右 : HW-USBN-2B)



註 : 本 Lattice Diamond 日本語マニュアルは、日本語による理解のため一助として提供しています。作成にあたっては各トピックについて可能な限り正確を期しておりますが、必ずしも網羅的あるいは最新でない可能性や、オリジナル英語版オンラインヘルプや各種ドキュメントと不一致がある可能性があり得ます。疑義が生じた場合は技術サポート担当者にお問い合わせ頂くか、または最新の英語オリジナル・ソースを参照するようお願い致します。

ケーブルドライバーが正しくインストールされていない場合や使用ケーブルタイプの設定が異なると、使用するPC環境で最初の実行する場合にエラーやウォーニングが表示されます。通常はDiamondのインストール時にドライバーのインストールを促すプロンプトが表示されますので、インストール作業が正常終了すれば、準備が整います。

MachXO2/3 シリーズや Crosslink ファミリーの機能として、プログラマーを用いて PC の USB ポートからデバイスの（ハードマクロが備える）SPI や I2C ポートを介してのプログラミング・アクセスが可能です。以下、特に言及しない場合は JTAG 接続について記述するものとします。

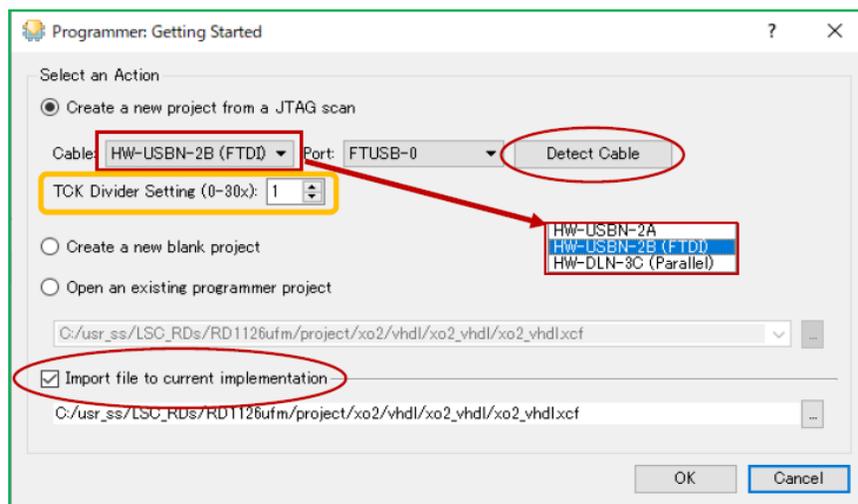
14.1.2 プログラマーの起動

プログラマーの起動には幾つかの方法があります。

- ① Windows スタートメニューから起動。[スタート] → [すべてのプログラム] → [Lattice Diamond 3.12] → [Diamond Programmer] と選択して起動します
- ② Diamond を立ち上げてから Programmer を指定して起動。プログラマーのアイコン  をクリックするか、メニューバーで [Tools] → [Programmer] を選択して起動します
- ③ インプリメンテーションのファイルリスト・ビュー内の「Programming Files」セクションに、既存のチェーンファイル (.xcf) がある場合、これをダブルクリックして起動します

インプリメンテーションで初めてプログラマーを起動した時は図 14-2 のような画面が立ち上がります。

図 14-2. インプリメンテーションで初回起動後の表示例



14.1.3 ケーブルの指定・検出

図 14-2 において、初回は『Create a new project from a JTAG scan』ボタンが選択されています。ケーブルがボードに接続されていないとコンソール（Diamond GUI 下部の”Output”ウィンドウ）にエラーメッセージが出力されます。このウィンドウでの”project”とはプログラマー・プロジェクトのことを指しています。

”Cable”部で選択可能なケーブルタイプは三つあります（図 14-2 内の右）が、プルダウン・メニューから直接指定することができます。『Detect Cable』ボタンをクリックすることで、ケーブルタイプを自動判別する（スキャンする）ことも可能です。一旦プログラマー・プロジェクトを作成後に変更することも可能です。

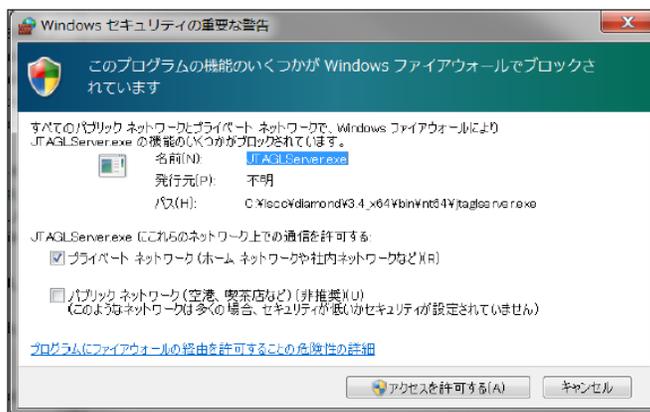
JTAG インターフェイスのない Crosslink の SSPI/I2C や、MachXO2/3 シリーズで SPI/I2C 接続している場合は「Open a new blank project」をチェックして起動することを推奨します。

デフォルトでは最下部の「Import file to current implementation」にチェックが入っています（インポートしない場合は、チェックをはずします）。通常はこのまま”OK”をクリックします。

「TCK Divider Setting」は、スキャンでデバイスが検出できないなどの問題があることが既知ケースでは、この値を大きくすることで期待動作する場合があります。チェインファイル作成後でも操作することができます（図 14-6）。

初めて実行した際に、図 14-3 のような Windows の警告メッセージが出る場合がありますが、右下『アクセスを許可する』をクリックします。

図 14-3. 初回アクセス時の Windows 警告メッセージ



検出が完了するとその旨を通知する図 14-4 のようなメッセージが出ますので、”OK”をクリックします。

図 14-4. 自動検出アクションでの通知画面

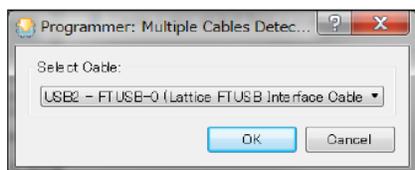
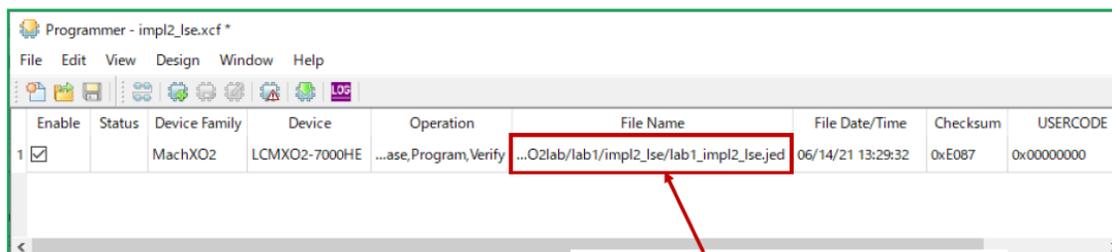


図 14-5 は起動後の表示例です。殆どの場合 [File Name] カラムに作業中のインプリメンテーションで生成した書き込みファイル (.bit, .jed) が自動的に選択・表示されます。[File Name] カラムが空白の場合は、まず当該セル内を一度クリックします。これによって、セル右端にブラウズ操作用のボタンが表れますので（図 14-5 内の下部）、これをクリックしてターゲットファイルをブラウズ後に選択します。

図 14-5. プログラマー・プロジェクト起動後の表示画面例



空白時はセルを一度クリックする

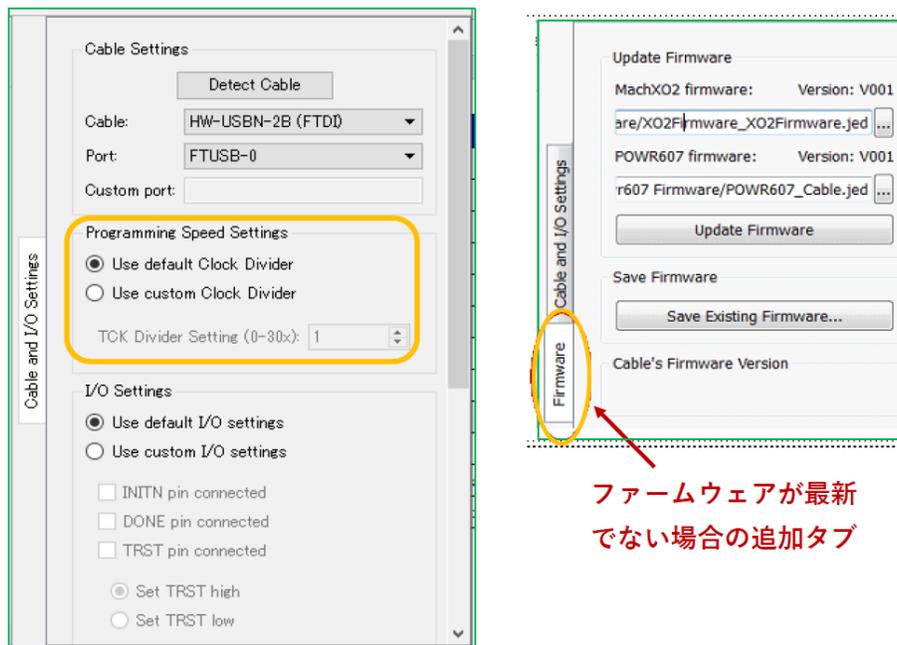
ケーブル設定に関して、プログラマーのウィンドウ右側に図 14-6 左図のような表示『Cable and I/O Settings』セクション（図示は一部のみ）があり、図 14-2 と同様の設定ができます。スキャン動作が不成功の場合、本セクションで設定変更することが可能です。図 14-2 の ”TCK Divider Settings” 設定は、このウィンドウでは ”Use custom Clock Divider” をチェックしてイネーブルした後に指定します。

また、ケーブルが ”HW-USBN-2B” の場合で、かつファームウェアが最新でない場合に限り、図 14-6 右図に示すような [Firmware] タブが表示されます。適切なファームウェアを入手して更新します。

14.1.4 スキャン (デバイスの検出)

新規デザインの書き込みや、ターゲットボードが新規または変更後（後述するチェーンファイル xcf を作成済みでない場合など）の操作です。前項 14.1.3 で記述した「Create a new project from scan」で起動した場合、図 14-5 の「Device Family」および「Device」カラムは、同時に実行されるスキャン結果を反映して表示されますのでスキャン操作は不要です。

図 14-6. ケーブル設定セクション (左、一部) とファームウェア更新タブ (右)



スキャンは、プログラマー内上部のアイコンをクリックする (図 14-7) か、プログラマーをデタッチして、メニューの [Design] → [Scan] を選択します。スキャンが正常に行われると (チェーンに接続されている) デバイス名が表示されます。他社デバイスは全て JTAG-NOP と表示されます。

図 14-7. スキャン・アイコン



JTAG チェインが複数デバイスよりなる場合や、例えば「Create a new blank project」を選択・起動後の初期表示デバイスと異なる作業をする場合 (当該インプリメンテーションと直接関連のないボードやデバイスのプログラミングの場合) など、意図する行のどこかで右クリックして当該行を削除し、その後スキャンすることも可能です。

プログラマーはデバイスから ”Device ID” を読み出すことでデバイスの識別を行います。同一ファミリーの派生品間で ID 番号を共有するケースでは、厳密に正確なデバイス名表記にならない事があります (例えば MachXO2-7000HE に対して MachXO2-7000ZE など)。こうした場合、注意を促す意味でデバイスのセルが黄色で表示されます。セルをクリックすると正しいデバイスの候補がプルダウン表示されますので、選択します。

14.1.5 アクセスモードとオペレーションの指定

「Create a new project from a JTAG scan」で立ち上がると、図 14-8 左側のような設定ウィンドウが表示されますので、当該デバイス行の「Operation」セルか、または行のどこかをダブルクリックして**アクセスモード**と**オペレーション**を指定します。図 14-8 は ECP5 の例で、デフォルトではアクセスモードが”JTAG 1532 Mode”、オペレーションが”Fast Program”になっています。それぞれクリックすると表示されるリストから、意図するものを選択します（図 14-9）。ECP5 ファミリーで集積されているにはコンフィグレーション SRAM のみですので、”JTAG 1532 Mode”、”Fast Program” が基本的なアクセスモードとオペレーションになります。

図 14-8. ECP5 の場合の画面表示例

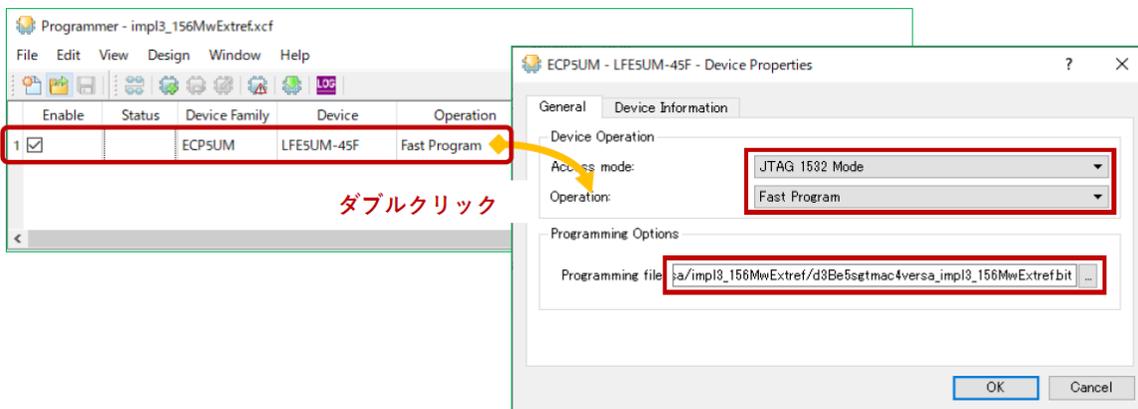
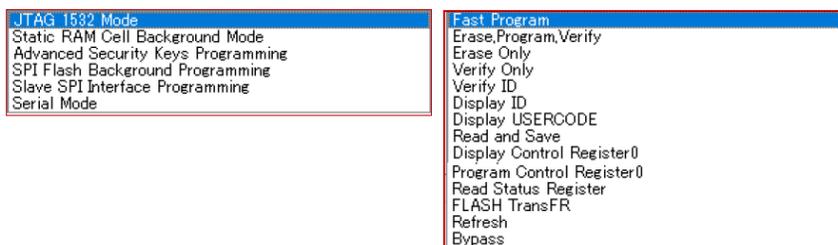
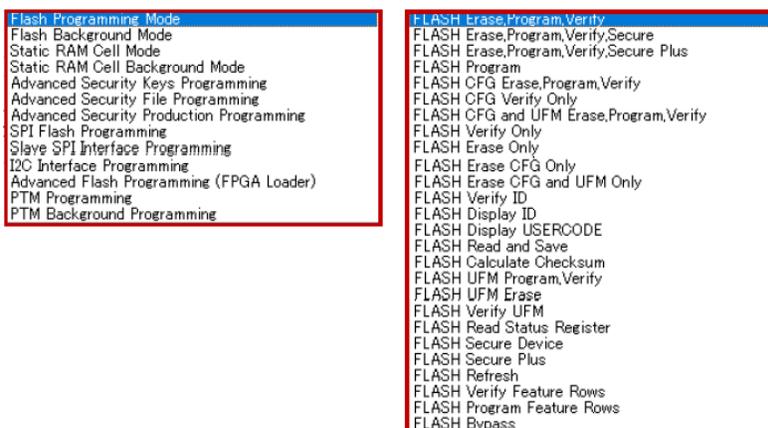


図 14-9. ECP5 ファミリーのアクセスモード（左）と JTAG1532 Mode のオペレーション（右）



MachXO2 ファミリーの場合を図 14-10 に示します。

図 14-10. MachXO2 ファミリーのアクセスモード（左）と Flash Programming Mode のオペレーション

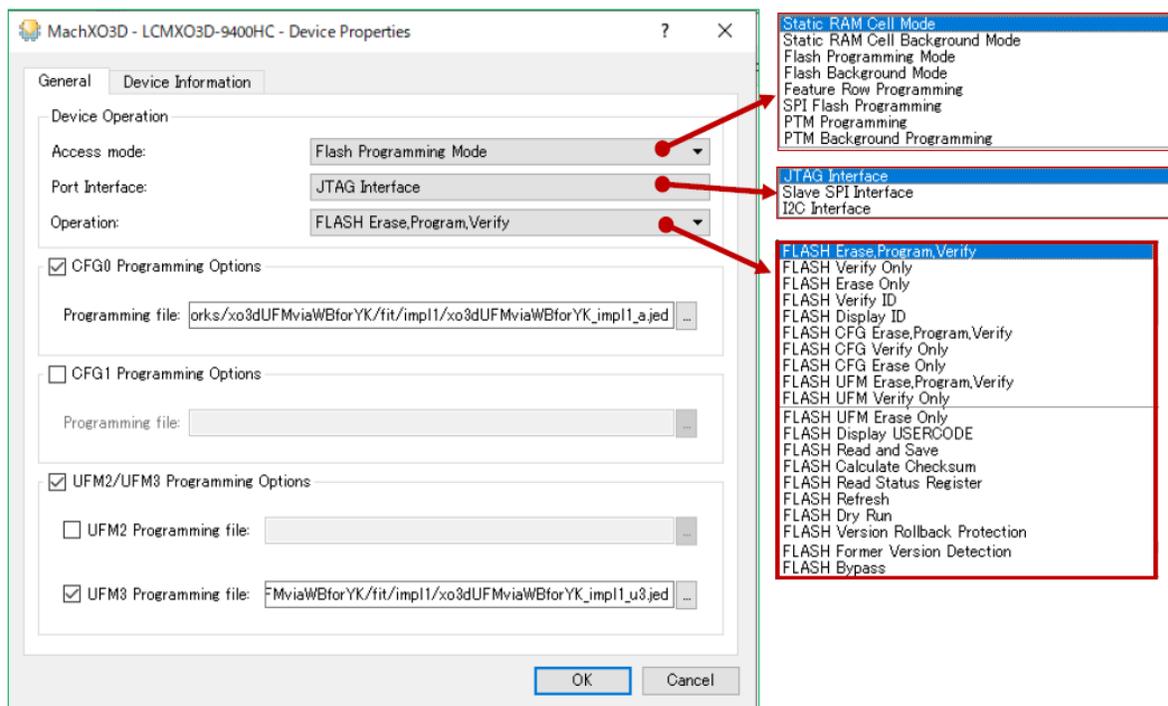


MachXO2/3 シリーズでは不揮発メモリー（フラッシュメモリーや NVCM）が集積されています。ECP5 同様 ”Fast Program” は有効ですが、不揮発メモリーへのプログラミングには「Access Mode」として”Flash

Programming Mode” または ”Flash Background Mode” を選択します。

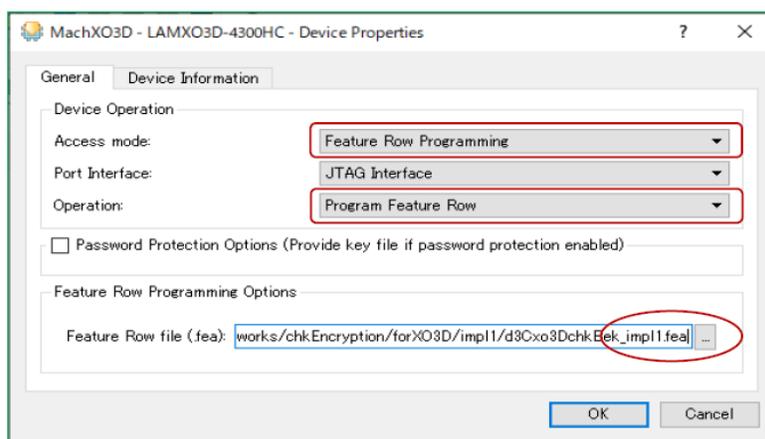
MachXO3D ファミリーの場合を図 14-11 に示しますが、「Port Interface」項目が追加されています。MachXO3D にはオンチップのコンフィグレーション用フラッシュメモリーが二面（CFG0/CFG1、または FLASHA/FLASHB）あり、UFM セクタも複数あります（UFM0～UFM3）。コンフィグレーション・モードの設定にも依存しますが、デザイン内でそれらがイネーブルされている場合は、サブプロセス [JEDEC File] を実行すると、ファイル末尾が “xxx_a.jed / xxx_b.jed”、”xxx_u2.jed / xxx_u3.jed” などの書き込みファイルが複数生成されます。この時、フィーチャー行用の “xxx.fea” という拡張子のファイルが、これらとは別に生成されます。さらに ”xxx.jed_list” というテキストファイルも生成されて、生成ファイルの一覧が書き出されます。図 14-11 は jedec ファイルの指定例です。該当するオプションをチェックしてファイル指定します。

図 14-11. MachXO3D ファミリーのアクセスモード (左) と Flash Programming Mode のオペレーション



どのようなデザインでも基本的に一度は、ファイル ”xxx.fea” もフィーチャー行を対象にプログラムする必要があります。「Access mode」は ”Feature Row Programming”、「Operation」は ”Program Feature Row” です。

図 14-12. MachXO3D ファミリーのフィーチャー行プログラム例

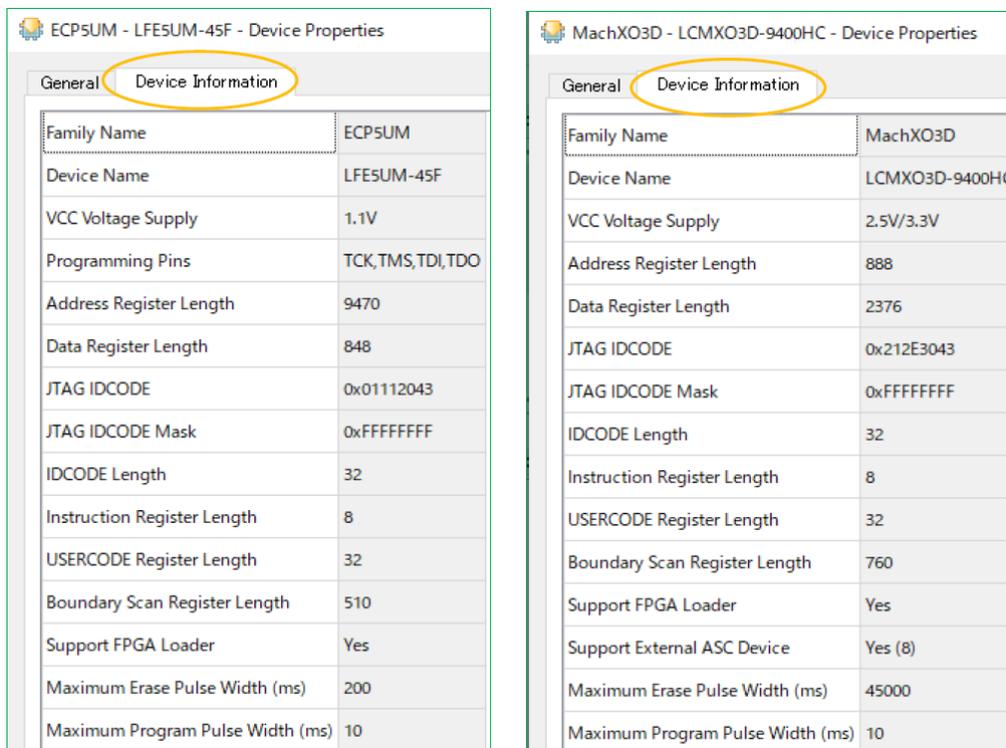


なお、フィーチャー行の書き換えは極力避けることを推奨しています。

このようにデバイス・ファミリーによって「Access mode」（アクセスモード）と「Operation」（オペレーション）の初期表示、および選択できる項目は異なります。詳細はオンラインヘルプの”Programming the FPGA”→”Programmer Options”→”Device Properties Dialog Box - Device, Access Mode, and Operation Options”をご参照ください。デバイス・ファミリー毎のオペレーションがすべて記述されています。

なお、図 14-8 右側や図 14-11 左側の”Device Properties”ウィンドウには [Device Information] タブがあります。デバイスの各属性情報が確認できます（図 14-13）。

図 14-13. Device Information タブ表示の例（左：ECP5、右 MachXO3D）



これらの情報は PCB レベルのバウンダリ スキャン・テストやインシステム・プログラミングなどを行う際に参照され得ます。

14.1.6 書き込みファイルの指定

次に書き込みファイルの指定を行います。起動・スキャン後、通常は「File Name」セルが自動フィルされますが、ブランクの場合は図 14-5 のように [File Name] セルをクリック後、ブラウザボタンを押して所望のファイルを指定します。「File Date/Time」「Checksum」「USERCODE」の各セルにはファイルから読み取られた情報が入ります。複数の xcf ファイルで、デザインの異なる同名の書き込みファイルを使い分けるケースなどでは、タイムスタンプ情報やチェックサム値で識別できます。

MachXO3D に関しては、オンチップのコンフィグレーション用フラッシュメモリーが二面あり、UFM セクタも複数ありますので、デザイン依存ですがそれらがイネーブルされている場合は、ファイル指定が他の XO2/3 シリーズとは異なります（図 14-11）。デザインを実装し、Export Files プロセスまで完了すると、該当する場合は複数の書き込みファイル（.jed）が生成されますので、それぞれに対応するファイル名を選択・指定するようにします（ファイル末尾が“xxx_a.jed / xxx_b.jed”、“xxx_u2.jed / xxx_u3.jed”など）。

書き込みファイルの拡張子は次の通りです：

- .jed MachXO2/3 シリーズのフラッシュメモリーなど、オンチップ不揮発メモリー対象

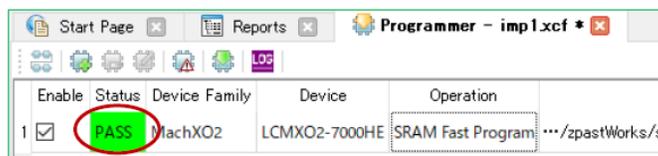
- .bit MachXO3L NVCM、iCE40 シリーズ NVCM、その他 ECP5 ファミリー (LatticeECP シリーズ) などのコンフィグレーション SRAM、および FPGA 外付け SPI フラッシュメモリー (単独) 対象
- .mcs 外付け SPI フラッシュメモリー (デュアル / マルチブートなど複数の bit ファイル) 対象

なお、使用するファイルはプログラマーによる (JTAG/SPI/I2C ポートを介した) 書き込みだけではなく、外部コントローラによる場合も同様です。ただし、後者の場合で特にキャラクタベースのインターフェイスである SPI/I2C を介する場合は、バイナリファイルであるビットファイルを扱いやすい 16 進数に変換して用います。

14.1.7 書き込みオペレーションの実行

各種設定が完了しましたのでメニューアイコン列から  をクリックして、書き込み (コンフィグレーション、またはプログラミング) を実行します。プログラマーをデタッチし、メニューで [Design] → [Program] と選択することでも可能です。

図 14-14. オペレーション正常終了の例



プログラマーを実行後、「Status」セルに緑色で PASS (もしくは黄色で DONE) と表示されれば、オペレーションが正常に終了したことを示します (図 14-14)。FAIL (赤) が表示された場合には、何らかの理由により、オペレーションが失敗しています。Diamond 下部の [Output] ウィンドウのメッセージをチェックするか、失敗の原因が不明の場合は、ログファイルを確認します。

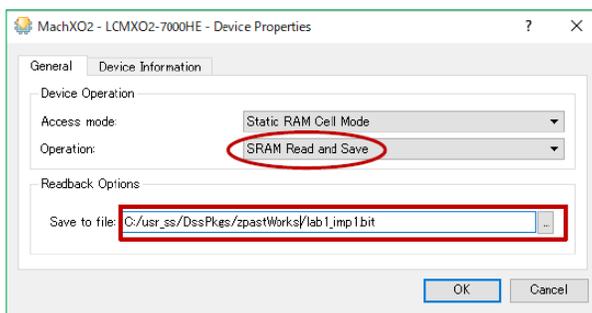
ログファイルの表示はアイコン  をクリックします。ログのクリアはデタッチ状態で [Design] → [Clear Log File] と指定します。

14.1.8 チェインファイルの保存

作業したプログラマー・プロジェクトは JTAG チェイン情報を含む設定一式であり、”チェインファイル” (拡張子 xcf) として保存します。保存アイコン  をクリックするか、デタッチ状態でメニューから [File] → [Save <Implementation 名>.xcf] または [Save <Implementation 名>.xcf as ...] を選択します。保存したチェインファイルは、Diamond のファイルリスト・ビュー内 ”Programming Files” セクションに自動的にアクティブなチェインファイル (太字) として取り込まれます。次回以降ダブルクリックすれば、同じ設定で起動します。

14.1.9 Read & Save オペレーションの実行

図 14-15. Read and Save オペレーション



書き込みオペレーションの前に、直前の状態で FPGA に書き込まれているコンフィグレーション・データをリードバックしてファイルに保存したい場合があります。このオペレーションが ”(SRAM/Flash) Read and Save” です。図 14-15 に MachXO2 シリーズで SRAM 対象の場合の例を示します。リードバック後に保存するファイル名を「Save to file」セルに入力し、『OK』ボタンでメインウィンドウに戻ります。

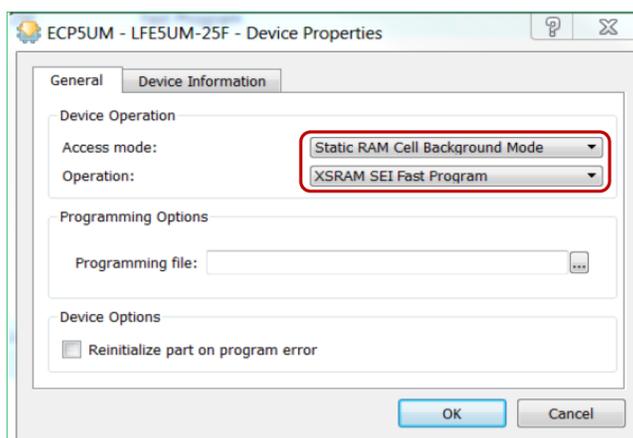
その後アイコン  をクリックして実行します。プログラマーのステータスが”DONE”となれば完了です。

なお、スプレッドシート・ビューの [Global Preferences] タブにある ”CONFIG_SECURE” オプションがイネーブルされている書き込みファイルがデバイスに書かれている場合、リードバック結果は全て ”1” (デバイスによっては全て ”0”) になります。

14.2 ソフトエラー検出機能検証時のオペレーション

Diamond にはソフトエラーを模擬して機能検証することを可能にする ”SEI エディター” があります。このツールで生成したビットストリームのロードはバックグラウンド・モードを用います。「Access mode」を ”Static RAM Cell Background Mode” に、「Operation」を ”XSRAM SEI Fast Program” にします (図 14-16)。

図 14-16. ソフトエラー検証時のプログラマー設定



なお、プログラミングの実行前に SED マクロ機能はオフにしておく必要があります。SEI エディターの操作については第 22 章をご参照ください。

14.3 その他のオペレーション

その他オペレーションの詳細については、オンラインヘルプの ”Programming the FPGA” → ”Programmer Options” → ”Device Properties Dialog Box” をご参照ください。

14.4 外付け SPI フラッシュメモリーのプログラミング

オンチップにコンフィグレーション SRAM のみを集積する ECP5 ファミリー (LatticeECP シリーズ) では、外付け SPI フラッシュメモリーからコンフィグレーション・データをロードして起動するモードが最も良く利用されます。また、オンチップ不揮発メモリーを集積する MachXO2/3 シリーズや Crosslink では、デュアルブート機能に対応しており、外付け SPI フラッシュメモリーを接続することが少なくありません。

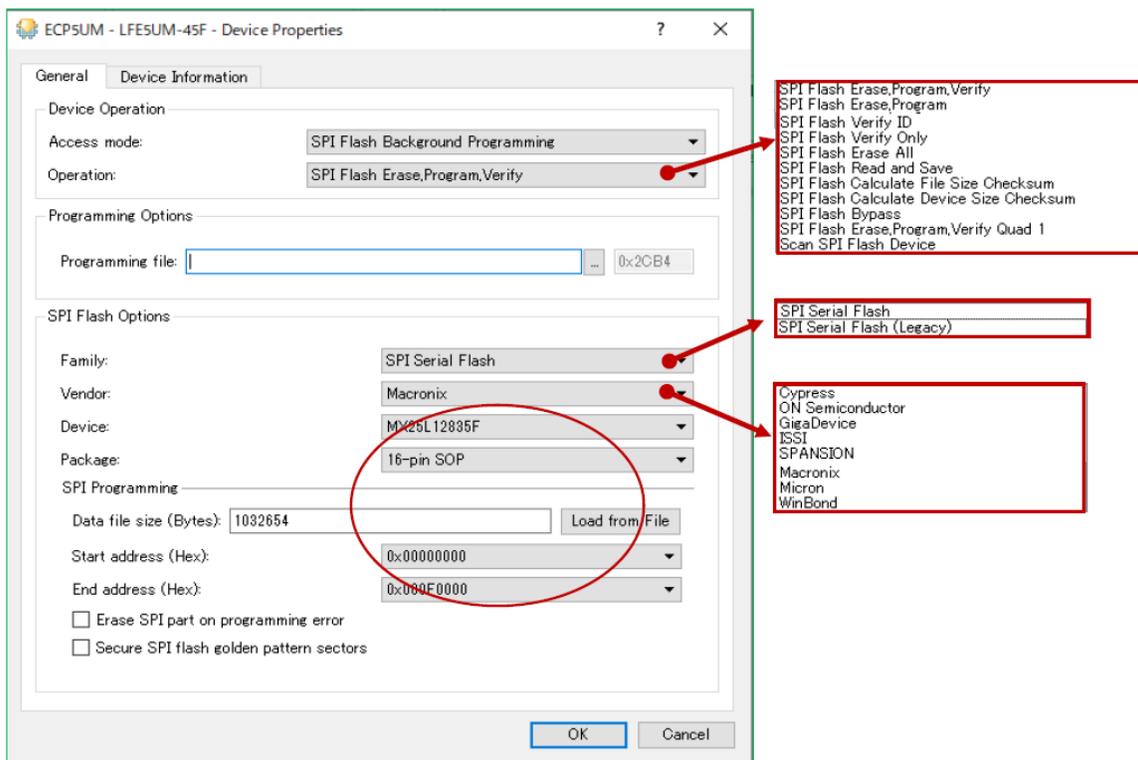
14.4.1 コンフィグレーション用外付け SPI フラッシュメモリー

ECP5 ファミリー (LatticeECP シリーズ) および MachXO2/3 シリーズでは、FPGA 経由で外付け SPI フラッシュメモリーを JTAG ポートからプログラミング (書き換え) することができます (Crosslink は SPI フラッシュメモリーに直接アクセスします)。デバイスは MSPI コンフィグレーション・モードに設定します。図 14-17 は ECP5 の場合で、「Device Operation」の「Access mode」は ”SPI Flash Background Programming” を選択します。MachXO2/3 ファミリーの場合 (図 14-10、図 14-11)、「Access mode」は ”SPI Flash Programming” です。

「Operation」は、それぞれで選択可能なオペレーションからいずれかを指定します。書き換える場合は”SPI Flash Erase, Program, Verify”の一連の動作を指定します。

使用するファイルは .bit または .mcs です。Export プロセスのサブプロセス [Bitstream File] で .bit ファイルを生成します。以降に記述するデュアルブートやマルチブートではなく、単独の mcs ファイルが必要な場合、ECP5 ファミリーではサブプロセス [PROM File] で生成できます。MachXO2/3 シリーズでは Export プロセスでは生成できませんので、デプロイメント・ツールで生成します。

図 14-17. オペレーションと SPI フラッシュメモリの選択例 (ECP5)



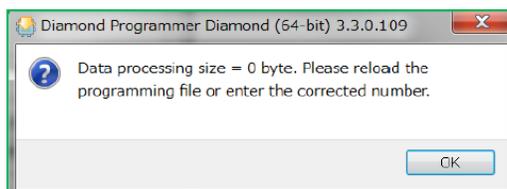
次に FPGA 外付けの SPI フラッシュメモリーを「SPI Flash Options」部で指定します。「Family」は”SPI Serial Flash”は各メモリーベンダーの現行シリーズ一覧から指定する場合に、“SPI Serial Flash (Legacy)”をレガシーな（旧シリーズの）SPI フラッシュメモリーから指定する場合に選択します。

次いで「Vendor」を選択後、適切な「Device」と「Package」を指定します。「Data file size (Bytes)」セルは上部「Programming file」を指定すると自動で入力されますが、ブランクのままの場合は『Load from File』ボタンをクリックして読み出します。このセルがブランクのまま『OK』ボタンを押すと、図 14-18 のようなメッセージが表示されて先に進むことができませんので、ご注意ください。

また bit ファイルが指定した SPI フラッシュメモリーに収まらないとか、SPI フラッシュメモリーの型番が不一致の場合も、それぞれエラーメッセージが出力されますので、解消するようにします。

この MSPI コンフィグレーション・モードでは、開始アドレス「Start Address (Hex)」は”0x000000”です。

図 14-18. 「Data file size」セルが 0 の場合のメッセージ

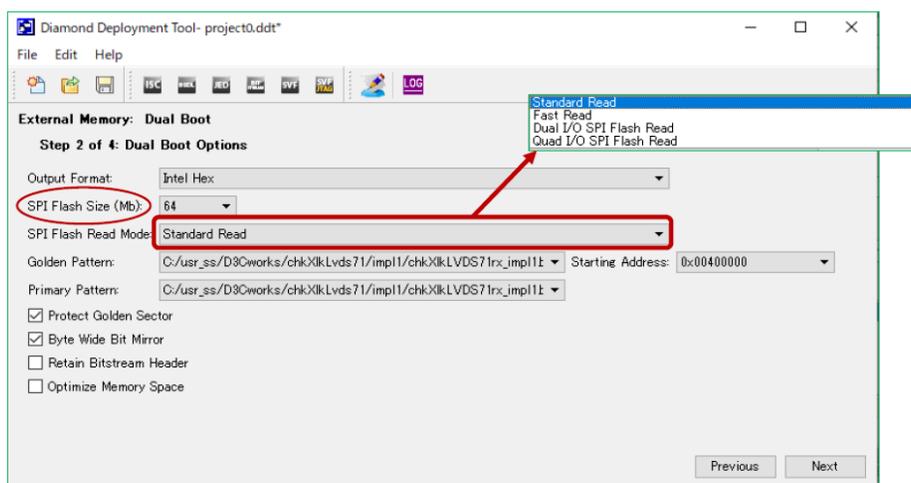


14.4.2 マルチブート用外部 SPI フラッシュ : ECP5 ファミリー

ECP5 ファミリーは、外付け SPI フラッシュメモリーに複数のコンフィグレーション・データ（ゴールデン・イメージとプライマリ・イメージ、および最大4本の代替イメージ）を保存する、いわゆるデュアル・ブート（マルチブート）機能をサポートしています。詳細はテクニカルノート TN-02039 をご参照ください。

オペレーションは前項と同様に「SPI Flash Background Programming」を選択し、「Programming File」としてデュアル/マルチ・ブート対応の書き込みファイル（.mcs）を指定します。mcs ファイルはプログラミング・ユーティリティであるデプロイメント・ツール（Deployment Tool）を使用して事前に作成しておきます。図 14-19 にデプロイメント・ツールでの「Step 2 of 4」の設定例を示します。「Golden Pattern」と「Primary Pattern」にそれぞれビットストリーム・ファイルを設定します。「Byte Wide Bit Mirror」はチェックします。

図 14-19. デプロイメント・ツールによる mcs ファイルの生成オプション設定例



デプロイメント・ツールによるデュアルブート用 mcs 生成の詳細については第 15.1.3 節をご参照ください。

14.4.3 デュアルブート用外部 SPI フラッシュ : MachXO2/3 シリーズ

MachXO2/3 シリーズはデュアルブート・モードに対応します。オンチップ不揮発メモリーと外付け SPI フラッシュメモリーにそれぞれ1本のイメージを保存し、マスターSPIモードとSDMモードの組み合わせで実現します。MachXO2ではオンチップメモリーにプライマリ・イメージ、SPIフラッシュメモリーにゴールデン・イメージが保存されます。MachXO3L/LFの場合は逆です。ともに起動は必ずプライマリ・イメージからで、ユーザーがその順序を変える方法はありません。

MachXO2/3 シリーズの場合、「Operation」の「Access mode」（図 14-10 の左下）は「SPI Flash Programming」ですが、外部 SPI フラッシュメモリーに置くイメージの開始アドレス「Start Address (Hex)」は「0x010000」でなければなりません。その他 SPI フラッシュメモリーの指定方法などは ECP5 ファミリーと同様です。

MachXO2/3 シリーズのなかで、MachXO3D ファミリーに限っては、前述のとおりオンチップのコンフィグレーション用フラッシュメモリーが二面あり（CFG0、CFG1）、内部 CFG0/1 のみのデュアルブートや外部 SPI フラッシュメモリーが関与する動作、さらにはセキュリティ機能を強化しているアーキテクチャに関する機能など、動作はやや複雑です。詳細はテクニカルノート TN-02069 をご参照ください。

なお、Diamond スプレッドシート・ビューで設定するグローバル制約（Global Preference）のコンフィグレーション指定については、第 16.3.1 項をご参照ください。

ここで、ボード設計に関して注意点を追記しておきます。MachXO2/3 シリーズの場合、SPI マクロを集積する EFB（組み込みファンクションブロック）が SPI マスターになります。クロック信号 `spi_clk` は出力（MCLK）になり、外部プルアップ抵抗 $1k\Omega$ （必ずプルアップ）を付加することが推奨されています。それぞれテクニカルノート「Hardware Checklist」（TN1208、TN1291）にも明記されていますので、ご参照ください。

14.4.4 デュアルブート用外部 SPI フラッシュ : Crosslink ファミリー

Crosslink ファミリーでは、ECP5 ファミリーや MachXO2/3 シリーズと異なり、デバイスを経由して外付け SPI フラッシュメモリーをプログラムする機能には対応していません。オフラインで SPI フラッシュメモリーをプログラミングして実装するか、プログラマーから直接アクセスできる PCB 設計にしておきます (Crosslink の MSPI 動作と競合しないように留意します)。

プログラマーの「Access mode」は「SPI Flash Programming」にします (図 14-20)。この場合の「Operation」は他と同様です (図 14-21)。なお、「Port Interface」はプログラマーから Crosslink にアクセスする際のポートを指定するもので、SPI フラッシュメモリーへのアクセスでは非該当です。

図 14-20. Crosslink のアクセスモードとポート・インターフェイス設定

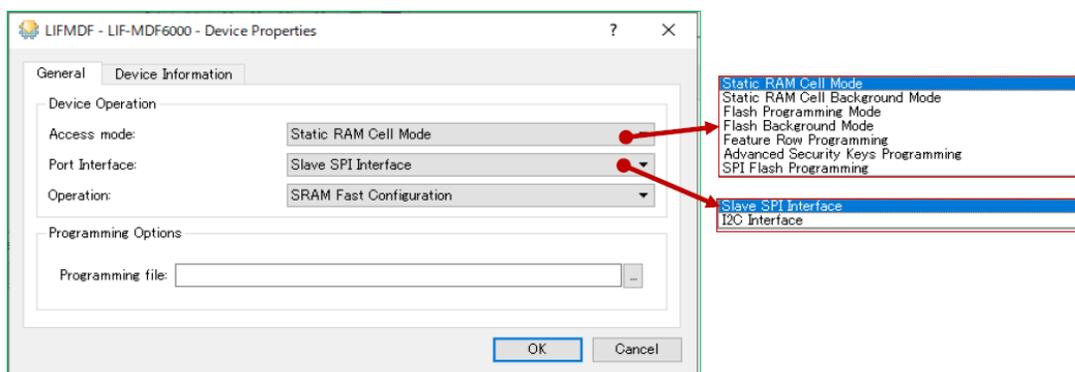
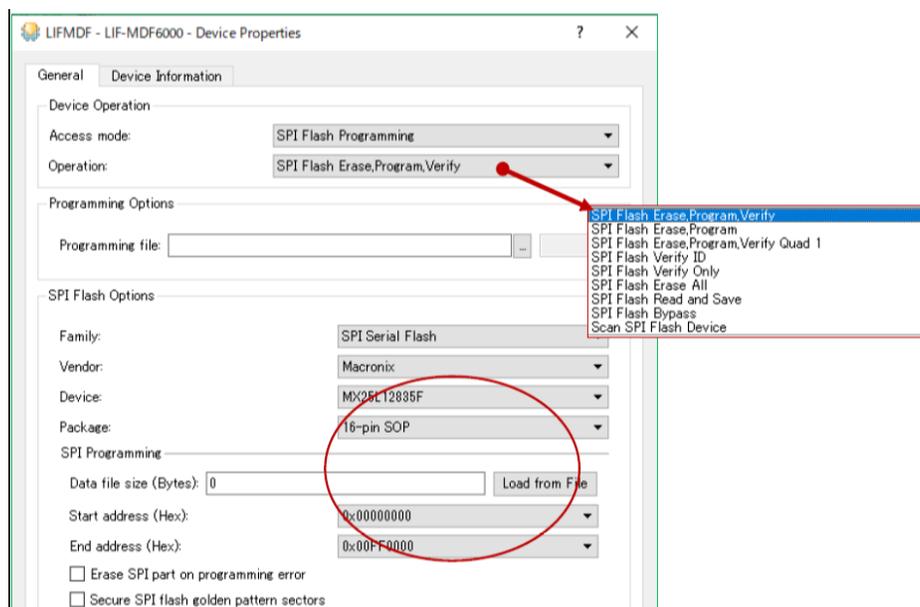


図 14-21. Crosslink の SPI Flash Programming オペレーション設定など



外部 SPI フラッシュメモリーに置くイメージの開始アドレス「Start Address (Hex)」は「0x000000」です。その他の指定方法などは XO2/3 ファミリーなどと同様です。

Crosslink ファミリーにおけるデュアルブート・モードの実現は、MSPI モードにし (MASTER_SPI_PORT = ENABLED) てスプレッドシート・ビューのグローバル制約 (Global Preference) で「BOOT_UP_SEQUENCE」を意図する動作によって NVCM-EXT / EXT-NVCM / EXT-EXT のいずれかに設定します。

「Programming file」セルは、オンチップ NVCM のイメージをデュアルブートの一つとする場合は、Diamond の Export プロセスで生成した .bit ファイルをそのまま指定します。「EXT-EXT」にした場合は、外部 SPI フ

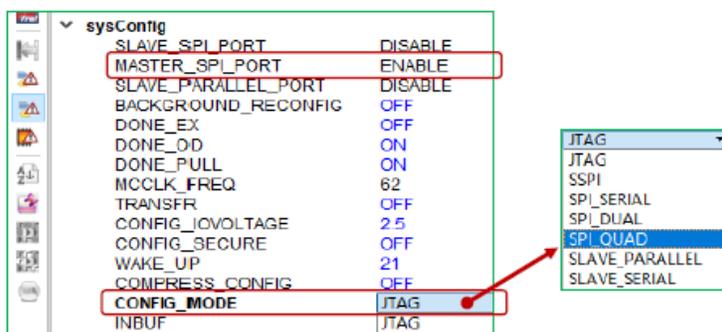
ラッシュメモリに二つのイメージをストアしますので、ECP5 のマルチブートと同様に、Deployment Tool であらかじめ二つのイメージを統合した mcs ファイルを生成しておき、これを指定します。

14.4.5 Dual/Quad リードモード用 SPI フラッシュメモリ : ECP5

ECP5 のマスター SPI モードを用いた外付け SPI フラッシュメモリによるコンフィグレーションでは、データ線を二本 (Dual) または四本 (Quad) 用いることで起動時間を短くすることが可能です (Dual/Quad リードモード)。手順として三つのポイントに留意する必要があります。

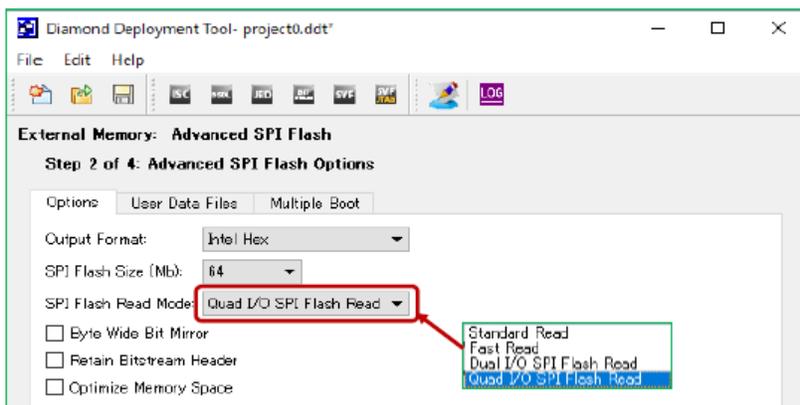
最初はスプレッドシート・ビュー [Global Preference] の "sysConfig" 部設定です。"MASTER_SPI_PORT" をイネーブルにすることが必須です。また、ECP5 の SPI インターフェイス端子を専用ピンとして保持 (preserve) する場合は、"SYS_CONFIG" モードを "SPI_DUAL" 或いは "SPI_QUAD" にする必要があります (図 14-22)。

図 14-22. Dual/Quad SPI リード用 sysConfig 設定例



次にビットストリーム・ファイルの変換です。ECP5 のマルチブートと同様に、デプロイメント・ツールで mcs ファイルに変換します。図 14-23 にデプロイメント・ツールでの "Step 2 of 4" の設定例を示します。

図 14-23. デプロイメント・ツールによる Dual/Quad SPI リード用 mcs 生成設定例

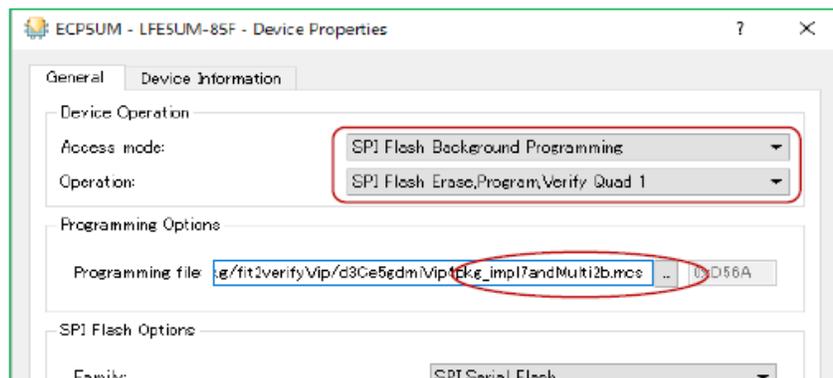


デプロイメント・ツールによる Dual/Quad SPI リード SPI フラッシュメモリ用 mcs 生成の詳細については第 15.1.5 項をご参照ください。

最後はプログラマーの設定です。例を図 14-24 に示します。「Access mode」は "SPI Flash Background Programming" を、「Operation」は "SPI Flash Erase,Program,Verify Quad 1" を選択します。「Programming file」セルにはデプロイメント・ツールで変換・生成した mcs ファイルをブラウザして指定します。図 14-24 では省略していますが、ウィンドウ下部「SPI Flash Options」セクションは SPI フラッシュメモリのデバイス指定です。

全て設定後、プログラマーに戻り、実行します。

図 14-24. Quad SPI リードのプログラマー設定例

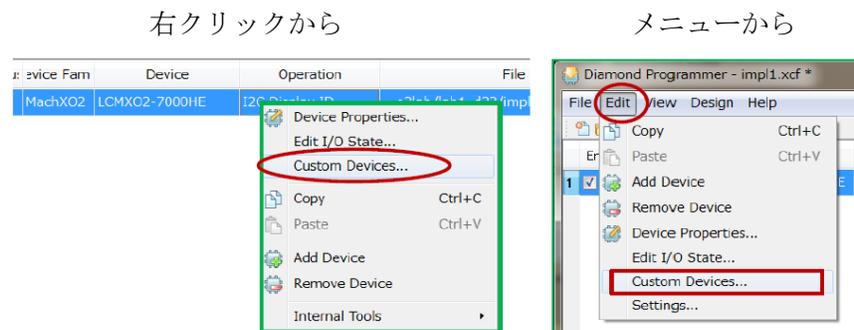


14.4.6 SPI フラッシュメモリーの登録、使用

”Device Properties” の「SPI Flash Options」部で指定する SPI フラッシュメモリー候補に、意図するメモリーがない場合の機能です。最初にカスタム品として手元で登録すると、その後に使用できるようになります。

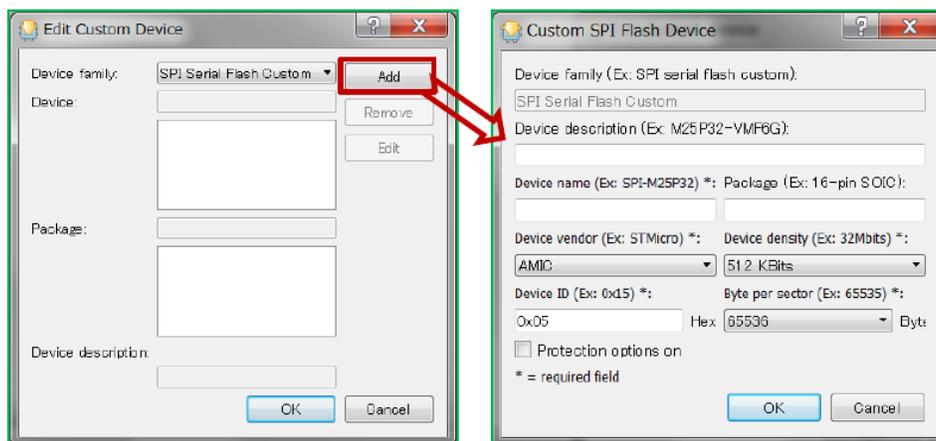
プログラマーのデバイス行を選択後右クリックすると表示されるドロップダウン・リストか（図 14-25、左）、デタッチして [Edit] メニューから（図 14-25、右）、いずれも ”Custom Devices ...” を選択します。

図 14-25. カスタム SPI フラッシュメモリー登録ウィンドウの呼び出し



表示されるウィンドウで『Add』ボタンをクリックし（図 14-26、左）、次いで適宜任意の名称とデバイス名、パッケージ、デバイス ID など必要な値やパラメータを選択・入力します。名称は後で選択する場合に判別がつくような名称にして登録します。

図 14-26. SPI フラッシュメモリー登録ウィンドウ



14.5 I2C・SPIポートからのプログラミング

MachXO2/3シリーズやCrosslinkファミリーではHW-USBN-2Bケーブルを用いてプログラマー(PCのUSBポート)からデバイスが備えるコンフィグレーション用のI2CポートやSPIポートを介してアクセスすることができます。MachXO2/3シリーズのI2CインターフェイスはEFB(ハードマクロ)のプライマリ・ポートです。

14.5.1 ケーブルの接続

ケーブルをファイワイヤーでターゲットボードに接続する際の、I2C / SPIに関わる信号線の割り当てのみを以下に示します。詳細は『FPGA UG-02042-26.4 Programming Cables』をご参照ください。

表 14-1. プログラミングケーブル HW-USBN-2B のピンと定義

信号	ケーブルでの名称	線材の色	PCからの方向	デバイスピン	信号の説明
Test Data Output	SDO/TDO	茶	Input	SPISO	SPI データ入力
Test Data Input	SDI/TDI	橙	Output	SISPI	SPI データ出力
Enable	ispEN/Enable/PROG/SN	黄	Output	SN	SPI チップセレクト
Test Clock	SCLK/TCK	白	Output	MCLK	SPI クロック
I2C Clock	SCL	白黄ストライプ	Output	SCL	I2C クロック
I2C Data	SDA	白緑ストライプ	Output	SDA	I2C データ

14.5.2 グローバル制約の設定

これら機能を有効にするためには、コンフィグレーション関連グローバル制約の1つであるオプションをそれぞれイネーブルした書き込みファイルを生成・プログラミングしておく必要があります。Diamondのデフォルト(いわゆる”SW Default”)はどちらもDISABLEです。

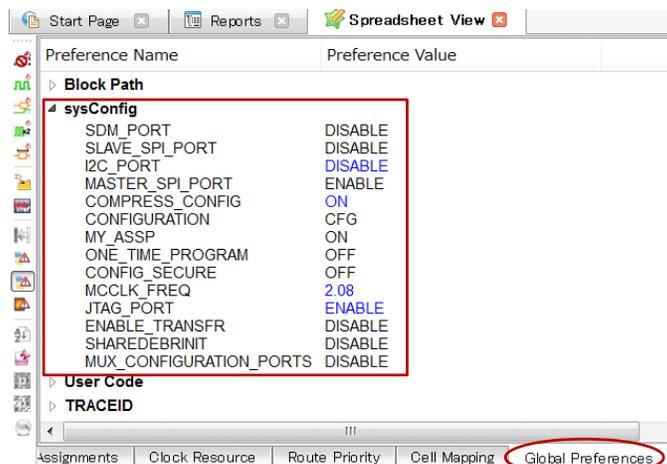
SPI アクセス SLAVE_SPI_PORT = ENABLE

I2C アクセス I2C_PORT = ENABLE

ちなみにブランクデバイスのデフォルト(いわゆる”HW Default”)は、どちらもENABLEです。

スプレッドシートビューでの設定を保存すると、設計制約ファイルlpfに書き出されますので、その後はテキストエディターによる編集も可能です。図 14-27 に MachXO3L の ”sysConfig” セクションを示します。

図 14-27. [Global Preference] の sysConfig セクション (MachXO3L)



14.5.3 プログラマーの設定

プログラマーのアクセスモードとオペレーションの設定例は図 14-28 および図 14-29 のようになります。アクセスモードは I2C ポート経由の場合は ”I2C Interface Programming”、SPI ポートの場合は ”Slave SPI Interface Programming” です。

図 14-28. I2C ポート・アクセスのオペレーション (MachXO2 の例)

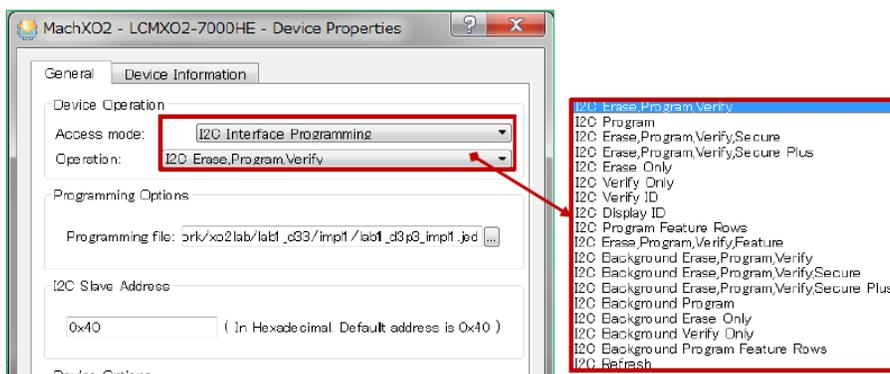
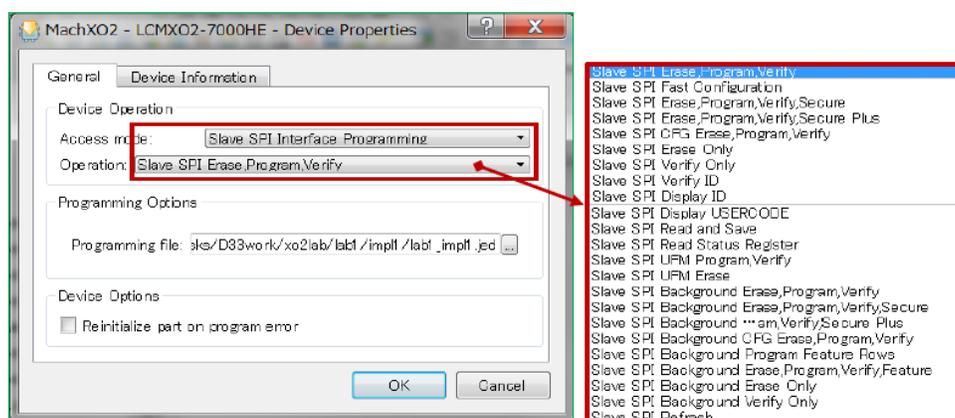


図 14-29. SPI ポート・アクセスのオペレーション (MachXO2 の例)



14.6 セキュリティー機能関連のオペレーション

14.6.1 ラティス FPGA とセキュリティー対応

前項で言及したように、ラティスの FPGA は全てのファミリーではスプレッドシート・ビューで ”CONFIG_SECURE” オプションをイネーブルすることで、デバイスから書き込み済みコンフィグレーション・データの読み出しを不可能にする基本機能に対応します。この機能を無効にするためには、デバイスを消去するか、本オプションを OFF にした書き込みデータで更新する手段しかありません。

なお、CONFIG_SECURE オプションがイネーブルされていると、通常の ”Erase, Program, Verify” オペレーションは選択できなくなり、”Erase, Program, Verify, Secure (Plus)” になりますので、ご注意ください。

この基本機能に加えて ECP5 ファミリー (LatticeECP2S/ECP3 ファミリー)、MachXO3L/LF ファミリー、および Crosslink では、”セキュリティーキー” による保護、書き込み (コンフィグレーション / プログラミング) データの暗号化 (Encryption) などに対応しています (どの機能に対応しているかはファミリーによって異なります)。さらに MachXO3D ファミリーではこれに加えて改ざん防止など、セキュリティー機能が大幅に強化されています。

14.6.2 暗号化対応パッケージのインストール

セキュリティー対応の機能を実現するためには、暗号化パッケージ（Encryption Package）が必要です。スタンドアロン・プログラマー用には別のパッケージを用意しています。Diamond のページ下部から入手します（図 14-30）。

これ以降は、パッケージのインストールが完了しているものとして記述します。ビットストリーム・ファイルの暗号化手順についてはデプロイメント・ツールで記述します（第 15.1.7 項参照）。

なお、ECP5 ファミリー（LatticeECP シリーズ）はコンフィグレーションメモリーが揮発性のため、暗号キーを保持するメモリーは OTP（一度のみプログラム可能）です。キーを一度プログラムした後は変更（書き換え）ができませんので、ご注意ください。

図 14-30. 暗号化対応パッケージのダウンロード

Software Downloads & Documentation							
Quick Reference	Technical Resources	Information Resources	Downloads				
Application Note	Known Issues	Product Brochure	Downloadable Software				
Installation Guides	Product Change Notification	Release Notes					
User Manual		Tutorials					
		Help					
TITLE	NUMBER	VERSION	DATE	FORMAT	SIZE		
<input type="checkbox"/> Diamond 3.12 64-bit Encryption Pack for Linux		3.12	12/8/2020	RPM	1.3 MB		
<input type="checkbox"/> Diamond 3.12 64-bit Encryption Pack for Windows		3.12	12/8/2020	ZIP	3.6 MB		
<input type="checkbox"/> Diamond 3.12 64-bit for Linux		3.12	12/8/2020	RPM	1.8 GB		
<input type="checkbox"/> Diamond 3.12 64-bit for Windows		3.12	12/8/2020	ZIP	1.6 GB		
<input type="checkbox"/> LatticeMico System for Diamond 3.12 64-bit Linux		3.12	12/8/2020	RPM	342.2 MB		
<input type="checkbox"/> Programmer Standalone 3.12 64-bit for Linux		3.12	12/8/2020	RPM	59.5 MB		
<input type="checkbox"/> Programmer Standalone 3.12 64-bit for Windows		3.12	12/8/2020	ZIP	56.4 MB		
<input type="checkbox"/> Programmer Standalone Encryption Pack 3.12 64-bit for Linux		3.12	12/8/2020	RPM	108.2 KB		
<input type="checkbox"/> Programmer Standalone Encryption Pack 3.12 64-bit for Windows		3.12	12/8/2020	ZIP	2.6 MB		
<input type="checkbox"/> Reveal Standalone 3.12 64-bit for Linux		3.12	12/8/2020	RPM	51.2 MB		
<input type="checkbox"/> Reveal Standalone 3.12 64-bit for Windows		3.12	12/8/2020	ZIP	55.6 MB		
<input type="checkbox"/> <input type="checkbox"/> Select All	Notify Me of Changes *		Download Selected as Zip File				

14.6.3 暗号化に用いるキーファイルの生成

最初のステップとして、セキュリティー機能を実装する際のベースになる暗号キーファイル（Encryption Key File）を生成しておきます。

暗号化対応パッケージをインストールしている状態で、セキュリティー対応デバイスをターゲットとした Diamond プロジェクトを作成すると、図 14-31 のようにツールメニューからセキュリティー設定ツール（Security Setting）が起動できるようになります。未インストール状態やセキュリティー非対応デバイスのプロジェクトではメニューの候補に表示されません。

最初のウィンドウは暗号キーファイルに対するアクセスを制限するためのパスワード設定です。

図 14-31. セキュリティー設定ツールの起動

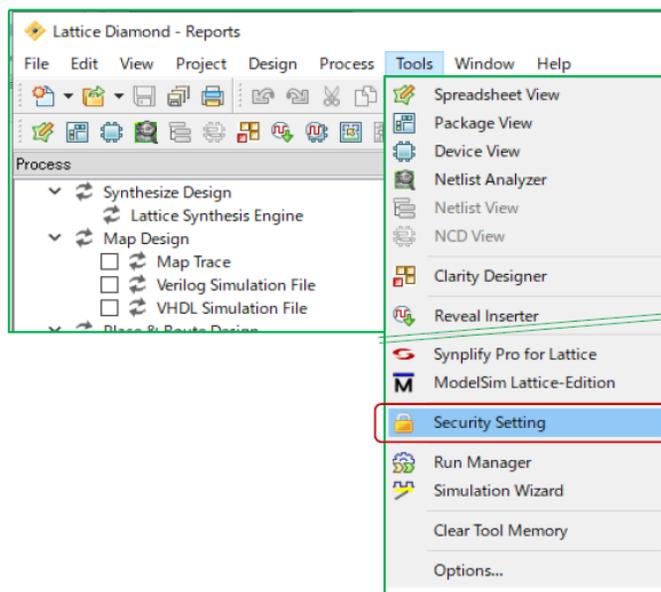


図 14-32 のようにデフォルトの "LATTICESEMI" が自動入力されて表示されます。所望のワードに変更する場合は『Change Password...』をクリックし、小ウィンドウで二回入力して『OK』をクリックします。二つが一致しないと、メッセージ (The passwords entered do not match. Please re-enter.) が表示されて先に進めません。問題なければ元のウィンドウで『OK』をクリックします。

図 14-32. 暗号キーファイルに対するパスワードの設定



次に表示されるウィンドウは暗号キーの設定です。始めは図 14-33 左のように「Encryption Key」セルは入力できない状態になっていますので、チェックボックスにチェックします。「Key Format」を ASCII/Hex/Binary から選択した後に「Encryption Key」セルに暗号キーを入力します。ASCII は 16 文字以下、Hex は 32 文字以下、Binary は 128b 以下です。問題なければメッセージ (Encrypted files are produced successfully.) が表示されて完了です。

図 14-33. 暗号キーの設定

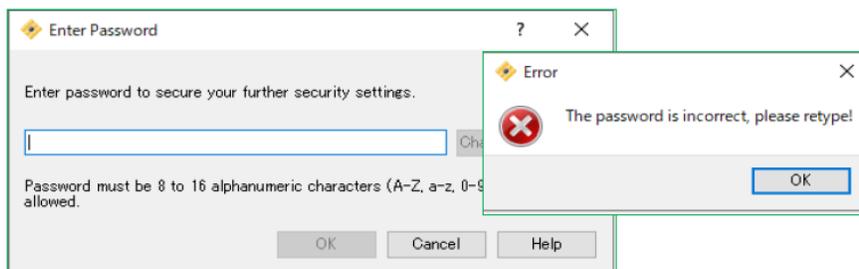


以上により、ECP5 では Diamond プロジェクト・フォルダー直下に “<project>.sec” および “<project>.bek” の二つのバイナリファイルが生成されます（.bek が暗号化キーファイル）。MachXO3L/LF ではファイル “<project>.key” も生成されます。bek ファイルはこれ以降の処理で適宜呼びだされます。これらはプロジェクト単位で共通で、インプリメンテーションごとに使い分けることなどはできません。

一旦パスワードをデフォルトから変更してセキュリティー設定が行われた場合で、二回目以降に “Security Setting” を起動すると図 14-34 のようなウィンドウが現れ、パスワード入力が求められます。不一致だとエラーメッセージが出て、先に進めませんので、ご注意ください。

パスワードが一致すれば、図 14-34 のようなウィンドウになりますので、暗号キーの変更が可能になりますが、パスワードを紛失してしまった場合、再現・回復できません。.sec/.bek 両ファイルを手動で削除して、やり直すしかするしか手段はありませんので、ご注意ください。

図 14-34. 二回目以降のセキュリティー設定の起動（パスワード変更後）



14.6.4 ビットストリームの暗号化

ECP5 に限っては、一旦セキュリティー・ツールで暗号化キーファイルを生成すると、そのプロジェクトは [Export Files] のサブプロセス [Bitstream File] で生成される bit ファイルは暗号化されたものが生成されます。ストラテジー・オプションなど、明示的な設定方法はあります。複数のインプリメンテーションがあっても、全て適用されます。暗号化をしない場合は、生成済みキーファイル一式を手動で削除します。

ECP5 ファミリー以外や別フローでビットストリームの暗号化が必要な場合はデプロイメント・ツールをします。第 15.1.7 項をご参照ください。

14.6.5 MachXO3D におけるセキュリティー設定

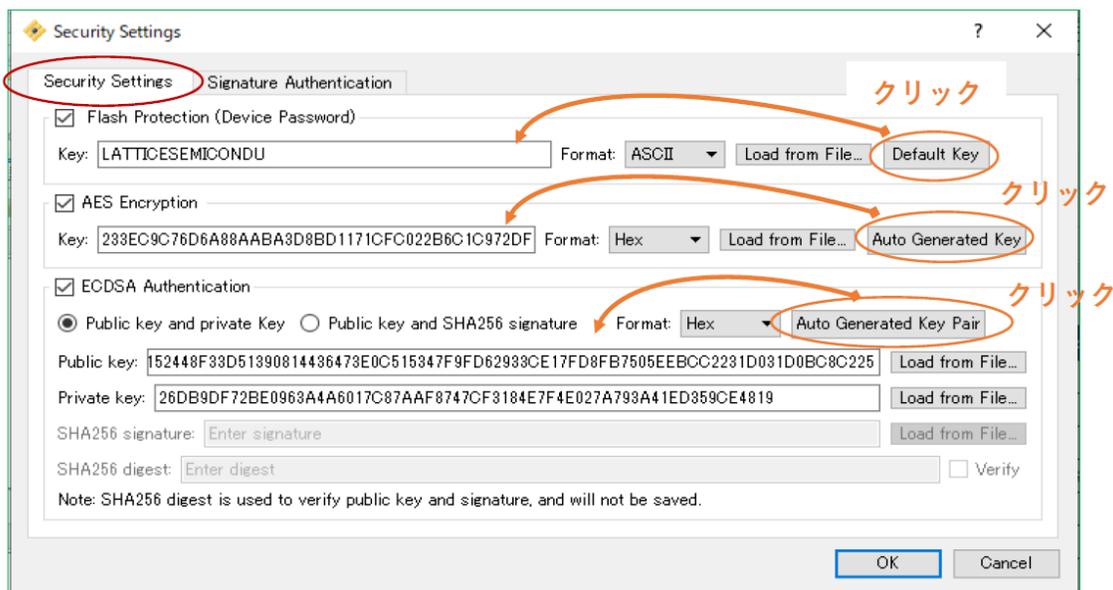
MachXO3D も暗号化や認証機能対応のための設定はセキュリティー・ツールをしますが（図 14-35 は設定ウィンドウ例）、セキュリティーレベルが何段階かあるため、スプレッドシート・ビュー [Global Preference] の sysConfig オプション設定やプログラマーのオペレーションを含めて、機能・動作がやや複雑です。詳細はテクニカルノート TN-02069 MachXO3D Programming and Configuration Usage Guide、TN-02091 MachXO3D Embedded Security Block 他をご参照ください。

- ・「Flash Protection」はパスワードによるアクセスの保護の設定です（“xxx.key” を生成）。プログラマーの “Advanced Security Keys Programming” アクセスの “Security Program Password Key” オペレーションで、保護する対象を指定するオプションがあります。これらはフィーチャー行にストアされます。

- ・ビットストリームの暗号化は “AES Encryption” で指定するキーで行います（“xxx.bek” を生成）
- ・ビットストリームの認証は “SHA256 ハッシュ符号” で生成したビットストリームのダイジェストに対して秘密鍵で ECDSA 符号化したシグニチャーと、これを公開鍵で復号化したものを比較して一致・不一致を検出することで行います

『Default Key』や『Auto Generated Key (Pair)』ボタンをクリックすると、それぞれツールが自動的にキーを生成して各セルを満たします。ユーザー入力も可能です。

図 14-35. MachXO3D のセキュリティー設定ウィンドウ例



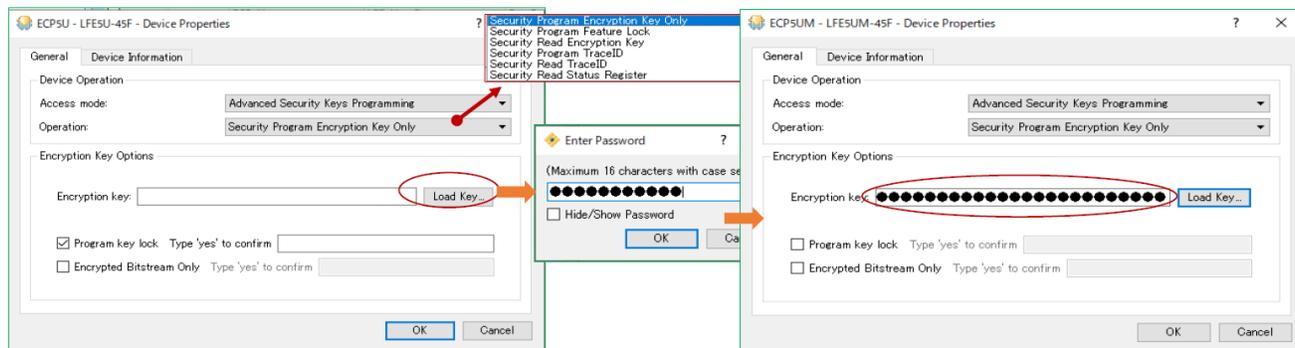
セキュリティー・ツールでの設定が完了すると、プロジェクト下に ”security_setting” という名称のサブフォルダーが生成され、”<project>.secproj” および ”<project>.bek” と ”<project>.key” の三つのバイナリファイルが生成されます。公開鍵 (Public Key) と秘密鍵 (Private Key) を生成した場合は、”<project>.pub” と ”<project>.prv” のキーファイルが二つ生成されます。

14.6.6 暗号キーのプログラミング : ECP5

暗号化したビットストリームをプログラミングをするためには、予め暗号キーをプログラミングしておく必要があります。ECP5 より前の LatticeECP3 ファミリーまでのオペレーションとは異なりますので、注意が必要です。暗号キーと、そのキーで暗号化済みの書き込みファイルを一度にプログラミングすることはできません。暗号キーが未書き込みの状態、暗号化済みビットストリームを用いてコンフィグレーションすることはできません。

暗号キーのみのプログラミングは、図 14-36 に示すように「Access mode」を ”Advanced Security Keys Programming”、「Operation」を ”Security Program Encryption Key Only” に指定します。

図 14-36. 暗号化キーのプログラミング設定



既に作成済みの暗号化キーファイル .bek を『Load Key...』ボタンで読み込みますが、図のようにパスワード入力を求められます。一致すれば図 14-36 の右のように「Encryption Key」セルが ”●” でフィルされますが、一致しないと先には進めません。

ここで、「Program key lock」オプションをチェック（イネーブル）すると、暗号キーファイルをリードバックする手段がなくなります。その場合、キーが正しいかどうかは、暗号化ビットストリームをプログラムすることによってのみ確認できます。それでも良い場合は「Type ‘yes’ to confirm」セルに “yes” とタイプします。

また「Encrypted Bitstream Only」オプションは、このオペレーション後にデバイスは暗号化済みビットストリームのみを受け付けるようになります。外付け SPI フラッシュメモリに保存済みの暗号化していないビットストリームによる MSPI モードでの起動もできません。上と同様に “yes” を入力して誤設定しないようにします。

既に記述した通り、暗号キーは OTP ですので、一度プログラムすると、書き換えはできません。「Encrypted Bitstream Only」オプションも一旦設定すると取り消しできませんので、ご注意ください。

設定完了後、このウィンドウを抜けて、プログラマーでオペレーションを実行します。プログラマーの [Status] が “PASS” と表示されれば成功です。

14.6.7 暗号化済みビットストリームの書き込み : ECP5

暗号化済みビットストリームの書き込み設定は通常と同じで、アクセスモードは “ISC1532 Mode”、オペレーションは “Fast Program” を選択します。暗号キーをプログラムしていない場合はオペレーションに失敗します。前項で示す暗号キーをプログラム済みの場合にのみ可能です。

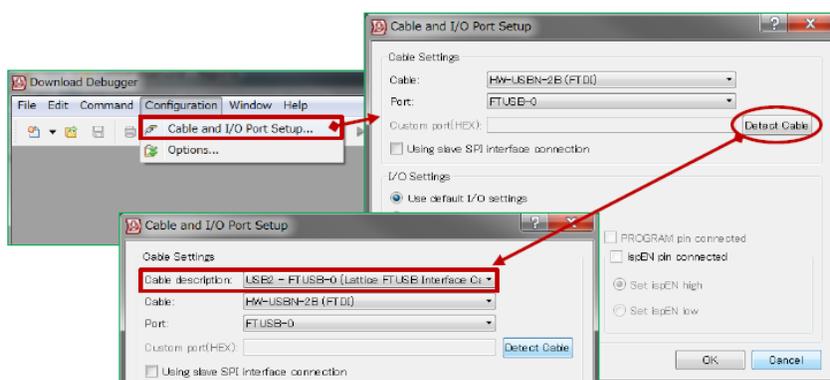
外付け SPI フラッシュメモリにプログラムする際にはアクセスモードが [SPI Flash Background Programming]、オペレーションは [SPI Flash Erase, Program Verify]などを指定します。

14.7 その他の機能

14.7.1 ダウンロード・ケーブルと I/O ポートの設定

デバイス・スキャンが成功しない場合、ダウンロード・ケーブルの設定が正しくない可能性があります、その確認には、“Download Debugger” が有用です。プログラマーをデタッチして Design → Utilities → Download Debugger と選択して起動後、Configuration → Cable and I/O Port Setup... と選択します。『Cable Detect』ボタンをクリックして、意図する状態と相違がないかを確認します (図 14-37)。

図 14-37. ケーブル設定の再確認



14.7.2 JTAG 信号線接続の確認

ダウンロード・ケーブルの設定が正しくてもデバイス・スキャンが失敗する場合、フライワイヤ形式のケーブルや PCB 配線が各 JTAG 信号線の期待する配線と合致していないことが考えられます。この場合 GUI を用いて接続テストをすることができます。

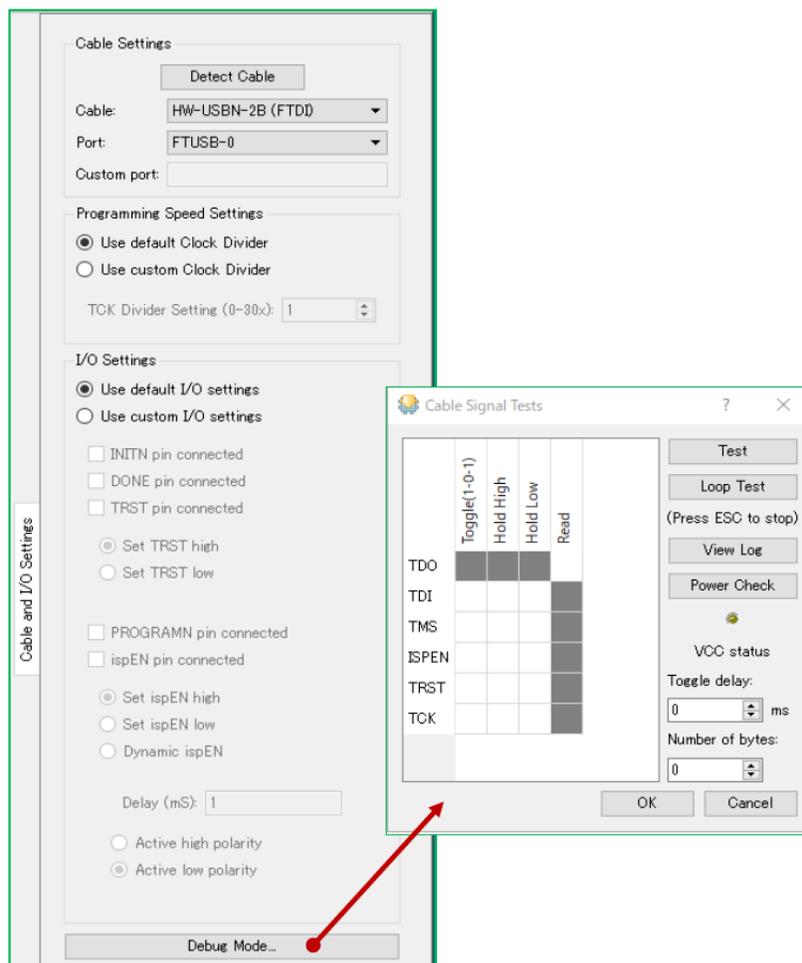
プログラマーのウィンドウ右側に図 14-38 の左部のようなセクションがあります。『Debug Mode ...』ボタ

をクリックすると右部のようなウィンドウが表示されますので、『Test』ボタンなどで接続状態が確認できます。アクションと対象信号は左側の表内の白色セルをクリックして指定します。

例えば TDO の Read セルをクリックするとチェックマークが表示されますので、その後『Test』をクリックします。リード結果がログファイルに書き出されますので、実際の論理レベルと合致するかが確認できます。

TDO 以外は全て出力信号ですので、トグル (Toggle)・High 固定 (Hold High)・Low 固定 (Hold Low) のいずれかアクションを意図する信号に対するセルをチェックします。『Test』ボタンをクリックすると、指定したアクション (レベル) に出力がドライブされますので、接続の正しさを確認できます。

図 14-38. 信号接続テスト GUI

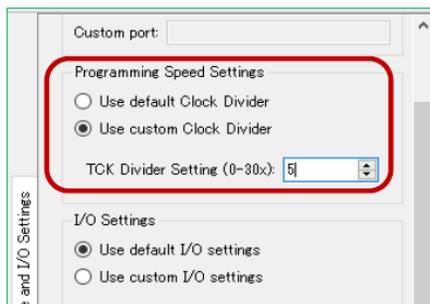


14.7.3 JTAG クロック周波数の変更

レガシーデバイスとか、何らかの事由で JTAG からのオペレーションが失敗する際に、JTAG クロック (TCK) の周波数を遅くすることで問題が解消する場合があります。

プログラマーをデタッチした後、図 14-39 のように Edit --> Settings... とたどると右のような画面が表示されます。「Pulse Width Delay Settings」部の『Use custom pulse width delay』ボタンをイネーブルし、[TCK low pulse width delay (0-10x):] 行の値を変更します。デフォルトは 1 で、大きくするほど周波数が遅くなります。

図 14-39. TCK 周波数の変更



周波数の算出式は次の通りです。

HW-USBN-2A Freq. = 6 / (1 + TCK Delay) [MHz] '0' → 6MHz, '1' → 3 MHz, etc.

HW-USBN-2B Freq. = 4 / (1 + TCK Delay) [MHz] '0' → 4MHz, '1' → 2MHz, etc.

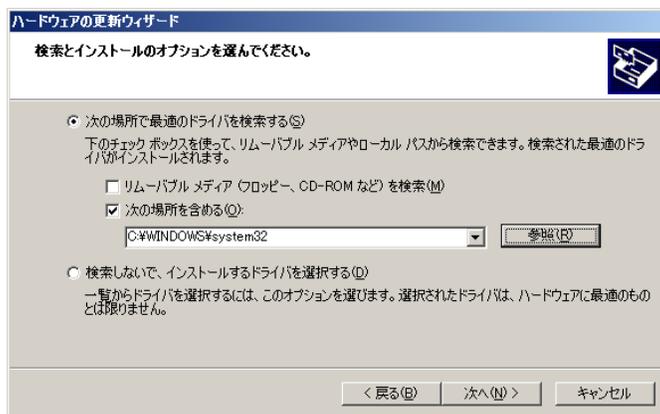
14.8 USB ドライバーのインストール

図 14-40. USB ドライバーのインストール要求初期画面



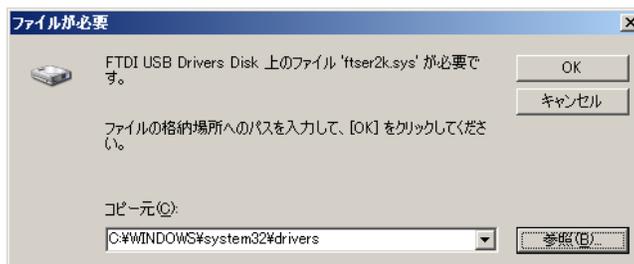
ターゲットボードを初めて USB ポートに接続した場合に、USB ドライバーのインストールが事前に完了していないと、インストールを促す Windows のメッセージが表示されます。『一覧または指定の場所からインストールする』を選択し次に進みます。(Windows は Administrator 権限でログインしている必要があります)。

図 14-41. USB ドライバーのフォルダー指定



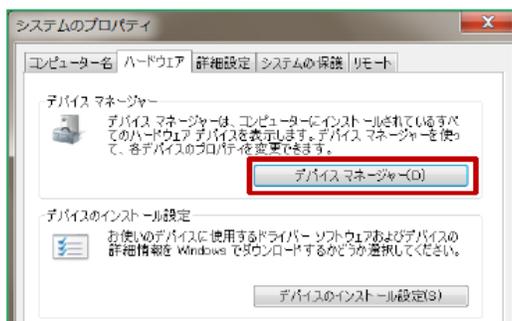
次の表示で”次の場所を含める”を選択して、『参照』ボタンでブラウズして”C:\WINDOWS\system32”を指定します(図 14-41)。その後『次へ』をクリックします。

図 14-42. FTDI USB ドライバーのコピー元指定



FTDI USB スレーブデバイスがボード上にある場合、続いて図 14-42 に示すような画面が表示されます。『参照』ボタンでブラウズして "C:\WINDOWS\system32\drivers" を指定し、『OK』をクリックします。これ以外に "ファイルが必要" と促される場合がありますが、同様に "C:\WINDOWS\system32" か "C:\WINDOWS\system32\drivers" を指定して手順を完了します。

図 14-43. デバイスマネージャの起動 (ウインドウの一部)



その後、システム・プロパティ (コントロールパネル) のデバイスマネージャ (図 14-43) で、ボードを接続した状態で正常に認識されていることを確認します (図 14-44)。

図 14-44. デバイスマネージャの USB ドライバー表示例 (左: FTDI あり、右: なし)

