LATTICE

プログラマーとユーティリティ

2015 年 5 月

Lattice Diamond 日本語ガイドライン

第21章 プログラマーとユーティリティ

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) と、付随するユーティリティ・ツー ルについて記述します。本節は Diamond 組み込みの場合についてのオペレーションを前提に記述しています。 スタンドアローン版ではオペレーションや表示が全く同じではない場合がありますのでご留意ください。

21.1 基本的な書き込み手順とオペレーション

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
- 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 (JTAG チェイン内のデバイス検出)
- 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(選択オペレーションによってはベリファイまで)
- 6. チェインファイルの保存

一度作業を終了する前に、後述のチェインファイル "<file name>.xcf" を保存しておくと、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。次 節以降、それぞれのステップについて詳細を記述します。

なお、ここで"プログラミング"とはオンチップ・フラッシュメモリや SPI フラッシュメモリに対する書 き込み、"コンフィグレーション"はデバイス内部コンフィグレーション SRAM への書き込み、という意図 であり、書き込み対象によって用語の使い分けをしています。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。

21.1.1 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ガイド『UG48 Programming Cables』 をご参照ください)。図 21-1 に、最新 USB ダウンロードケーブル(右)と、最も出荷数の多いその前バージョンの USB ケーブル(左)の外観図を示します。



^{© 2015} Lattice Semiconductor Corp. (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。その作成にあたっては各ト ピックについて、それぞれ可能な限り正確を期しておりますが、必ずしも網羅的ではなく、或いは最新でない可能性があります。また、意図せずオリジナル英語版 オンラインヘルプやリリースノートなどと不一致がある場合もあり得ます。疑義が生じた場合は、ラティスセミコンダクター正規代理店の技術サポート担当にお問 い合わせ頂くか、または極力最新の英語オリジナル・ソースドキュメントを併せて参照するようにお願い致します。)

www.latticesemi.com

Lattice Diamond 日本語ユーザガイド

ケーブルドライバ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 環境で最初に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、 用意が整っているはずです。問題がある場合は、21.xx 節をご参照ください。

MachXO シリーズの機能として、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 EFB 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。HW-USBM-2A では SPI ポートからのアクセスのみが可能です。HW-USBM-2B は Diamond 3.1 以降でサポートされており、I2C ポー トからのアクセス機能もサポートするようになっています。

21.1.2 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.x]→[Accessories]→[Diamond Programmer] と選択して起動
- ② Diamond を立ち上げてから Programmer を指定して起動

プログラマのアイコン 🌆 をクリックするか、"Tools" メニューバーから "Programmer" を選択して起動

③ Diamond の File List 内の "Programming Files" セクションに (アクティブな) 既存のチェインファイル (.xcf) がある場合、これをダブルクリックして起動

プロジェクトで初めて起動した時は図 21-2 のような画面が立ち上がります。

図 21-2. プロジェクトで初めてプログラマーを起動した後の表示

Separate Programmer: Getting Started	? ×
Select an Action	
Oreate a new project from a scan	
Cable HW-USBN-2B (FTC •) + FTUSB-0 • Detect Cable	
Create a new blank project	
○ Open an existing programmer project HW-USBN-2A HW-USBN-2B (ETDI)	
C:/aUSER_SStasks/D33work/xo2lab/lab1/impl1/impl1.xcf	•
Import file to current implementation	
C:/aUSER_SStasks/D33work/xo2lab/labl /impl1 /cf	
ОК	Cancel

21.1.3 ケーブルの指定・検出

図 21-2 において、初回は『Create a new project from 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(プロジェクトナビゲータ下部の『Output』窓枠)にエラーメッセージが 出力されます。

"Cable" 部で選択可能なケーブルタイプは三つあります(図 21-2 内の右)が、プルダウンで直接指定する ことができます。或いは『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勿論可能です。

『Create a new blank project』を選択した場合で、作業しているインプリメンテーション下に既存の xcf ファ イルが存在する場合、デフォルトで最下部の『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。

初めて実行した際に、図 21-3 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』をクリックして先に進みます。 図 21-3. 初回アクセス時の Windows 警告メッセージ

✓ Windows セキュリティの重要な警告
このプログラムの機能のいくつかが Windows ファイアウォールでブロックさ れています
すべてのパブリックネットワークとブライベート ネットワークで、Windows ファイアウォールにより JTAGLServerexe の機能のいくつかがブロックされています。 名前(N): <u>JTAGLServerexe</u> 発行元(P): 不明 パス(H): C:Viscc¥diamond¥3.4_x64¥bin¥nt64¥jtagserverexe
JTAGLServerexe にこれらのネットワーク上での通信を許可する: ▼ プライベート ネットワーク (ホーム ネットワークや 社内ネットワークなど XR) ■ パブリック ネットワーク (空港、喫茶店など) (非推奨XU) (このようなネットワークは多くの場合、セキュリティが低いかセキュリティが設定されていません)
プログラムにファイアウォールの経由を許可することの危険性の詳細

検出が完了するとその旨を通知する図 21-4 のようなメッセージが出ますので、"OK" をクリックして抜けます。

図 21-4. 自動検出アクションでの通知画面

🚱 Programmer: Multiple Cables Detec 🛛 🗙
Select Cable:
USB2 - FTUSB-0 (Lattice FTUSB Interface Cable 🔻
OK Cancel

図 21-5 は『Create a new blank project』を選択した場合の起動後の表示例です。『File Name』欄に作業中の 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・表示されますが、本図 のように『File Name』セルがブランクの場合は、まず枠内をクリックします。これによって、枠の右端にブ ラウズ操作ができるようにボタンが表れますので(図 21-5 内の下部)、これをクリックしてターゲットファ イルを選択します。

図 21-5. ブランクプロジェクトとして立ち上がり直後の表示画面

	Start Pag	ze 🖸 🚺	Reports	🛛 🛛 😵 Pro	grammer * 🔀	
00	88	🖗 🚷 🍪	L03			
E	Enable 3	tatus levice	Famil	Device	Operation	File Name
1 🗸]	Mach)	(02 LCN	1XO2-7000HE	FLASH Erase,Program,Verify	_
•	3	なち上に	がりī	道後の表	示例	枠内をクリック
			05			
90						
En	nable Sta	atus evice F	amil	Device	Operation	File Name

なお、プロジェクト起動後でもケーブル設定を変更可能です。プログラマー起動後のウィンドウで右側に のような表示セクションがあります(図 21-6 左)。図 21-2 と同様の操作ができるようになっています。次 項で説明するスキャン動作が不成功の場合、本セクションで正しい設定に変更することが可能です。

また、図 21-6(右)に示すように Diamond 3.2 以降は『Firmware』タブが追加になっています。ケーブルが "HW-USBN-2B" の場合で、ファームウェアが最新でない場合に限り、本タブが表示されますので、容易 に更新作業が可能です。

Cable Settings Detect Cable Cable: HW-USBN-2B (FT • Port: FTUSB-0 • Custom port: I/O Settings © Use default I/O settings © Use custom I/O settings © Use custom I/O settings INIT N pin connected DONE pin connected TRST pin connected © Set TRST high	III	Firmware Cable and I/O Settings	Update Firmware MachXO2 firmware: Version: V001 are/XO2Firmware_XO2Firmware.jed POWR607 firmware: Version: V001 r607 Firmware/POWR607_Cable.jed Update Firmware Save Firmware Save Firmware Cable's Firmware Version
---	-----	---------------------------------	--

図 21-6. ケーブル設定変更セクション(左)とファームウェア更新タブ(右)

21.1.4 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の(後述するチェインファイル xcf を 作成済みでない)場合などの操作です。

前項 21.1.3 で記述した『Create a new project from scan』で立ち上げた場合、図 21-5 の『Device Family』および『Device』欄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。

なお、デバイスの識別はデバイスから『Device ID』を読み出すことで行います。同一ファミリの派生品間で ID 番号を共有するケースでは、厳密に正確な Device 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。

図 21-7. デバイスの削除



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(図 21-7)、その後 スキャンすることも可能です。

図 21-8. スキャンアイコン

	00 00	i	2	Δ	\$	LOG	
スキャン	Enable		Statu	IS	le	vice	Famil

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 21-8)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(チェイ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。

21.1.5 書き込みファイルとアクセスモードの設定

まず**アクセスモードとオペレーションを指定**します。当該デバイス行の『Operation』セルか、行のどこか をダブルクリックすると、図 21-9 のような設定ウィンドウが立ち上がります(LatticeECP3 の例)。デバイ スファミリによって『Access mode』("アクセスモード")と『Operation』("オペレーション")の(初期) 表示項目は異なります。

図 21-9. LatticeECP3 の場合の初期画面

😔 Programmer *	
File Edit View Design Window Help	CalticeECP3 - LFE3-35EA - Device Properties
	General Device Information
Enable Status Device Family Device Operatio	Device Operation
1 V LatticeECP3 LFE3-35EA Fast Program	Access mode: JTAG 1532 Mode 🔹
	Operation: Fast Program
	Programming Options
	Programming file:
	Device Options
	Reinitialize part on program error
🔒 Analysis Files	OK Cancel

LatticeECP3ファミリの場合のAccess mode選択肢と"JTAG 1532 Mode"指定時のOperation選択肢を、図 21-10 に例として示します。"JTAG 1532 Mode"以外のモードを選択した場合、それぞれのアクセスモードで個別のオプションがありますが、個々のオペレーションの意味はほぼ自明ですのでここでの説明は割愛します (本章末尾を参照)。

なお、Diamond 3.2 以降では図 21-9 の画面で『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 21-10. LatticeECP3 の場合の各オプション(一部)

JTAG 1532 Mode 指定時の Operation (右) Access mode (左) Advanced Security Keys Programming Erase,Program,Verify Advanced Security Production Programm Verify Only Erase Only Slave SPI Interface Programming SPI Flash Background Programming Verify ID Dual Boot SPI Flash Programming Display ID Static RAM Cell Background Mode Display USERCODE Serial Mode Read and Save SSPI Flash Programming Read DONE bit Display Status Register Display Control RegisterO Refresh from FLASH Bypass

同様に MachXO2 ファミリの場合の例を図 21-11 に示します。

次に**書き込みファイルの指定**を行います。起動・スキャン後に『File Name』セルがブランクの場合、図 21-5のように Data File セルをクリック後ブラウズボタンを押して所望のファイルを指定するのが一般的です。 LatticeECP シリーズの場合は、拡張子が bit(または rbt)のファイルを、MachXO シリーズの場合は、拡張 子が jed(または isc)のファイルを指定します。

或いは、図 21-9の設定ウィンドウを抜ける時に、中央部の "Programming File"欄にブラウズして指定する ことも可能です。この場合、指定ファイルがターゲットデバイス用の書き込みファイルでない場合は、エラー メッセージが直ちに表示されます。プログラマーの Data File セルで指定した場合は、オペレーション(書き 込み)実施時になってようやく誤りがあれば通知されます。



図 21-11. MachXO2 ファミリの場合の各オプション(一部)

書き込みファイルを指定すると、『File Date/Time』『Checksum』『USERCODE』の各セルがファイルから読み取られた情報で埋められます。次項で説明する xcf ファイルで既存プロジェクトを起動した場合などで、かつ同名称の書き込みファイルを繰り返し用いる場合には、特にタイムスタンプ情報は最新かどうかを確認する手助けとなります。

リードバックしてファイルに保存するオペレーションでは、"Programming file"の欄は保存するファイル名にしておきます。

なお、コンフィグレーション用書き込みファイルの標準的なファイルタイプは以下の通りです。

- .jed MachXOシリーズ・オンチップフラッシュなど
- .bit 外付け SPI フラッシュ(単独)、LatticeECP シリーズオンチップ SRAM、MachXO3L NVCM、iCE40 シリーズ NVCM など

.mcs 外付け SPI フラッシュ(デュアルブート)

21.1.6 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(設定したオペレー ション)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 21-12. オペレーション正常終了の例

🚯 Start I	Page 🛛 🖸	3 📃 Re	ports 🗵 🛛 😂 F	Programmer – impl1.xcf * 🗵	
8) 🖉 🛛	à 🎥 🚾			
Enable	Statu	evice Fam	Device	Operation	
1 🗸	PASS	MachXO2	LCMXO2-7000H	E ···se,Program,Verify	···o2la

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、プログ ラミング(オペレーション)が正常に終了したことを示します。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。オペレーション失敗の原因が不明の場合は、ログファイ ルにヒントがある場合がありますのでチェックします。ログファイルのクリアはデタッチ状態で [Design] → [Clear Log File] と指定して行います。

21.1.7 Read & Save (オペレーション)の実行

生成した書き込みファイルをプログラミングする前に、直前の状態で FPGA に書き込まれているデータを 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。この場合のオペレーションが "Read and Save" で す。図 21-13 に LatticeECP シリーズ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入力し、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。

図 21-13. Read and Save オペレーション

🚷 LatticeECP3 - LFE3-35EA - Device Properties
General Device Information
Device Operation
Access mode: JTAG 1532 Mode
Operation: Read and Save
Readback Options
Save to file: ks/D33work/xo2lab/lab1d33versa/versa_original.bit
OK Cancel

21.1.8 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一連の設定であり、"チェインファイル" (拡張子 xcf) として保存できます。保存アイコン \blacksquare 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 File List ウィンドウ内 "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して同じ設定で起動可能です。

21.2 外付け SPI フラッシュメモリのプログラミング

ラティスの FPGA ではコンフィグレーション用やデュアルブート機能対応のために外部 SPI フラッシュメ モリを接続できます。本節ではこれらに関するオペレーションについて記述します。

21.2.1 コンフィグレーション用外部 SPI フラッシュ(LatticeECP シリーズ)

オンチップにコンフィグレーション SRAM のみを集積する LatticeECP シリーズ(LatticeECP2M / ECP3 / ECP5) では、外付け SPI フラッシュメモリ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。

このような場合、オンボードの SPI フラッシュメモリを FPGA 経由で 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。図 21-14 は LatticeECP3 の場合です。プログラマーで『Device Operation』を 設定する際に、[Access mode] 行で "SPI Flash Background Programming" を選択し、[Operation] 行では図中に 示す選択可能なオペレーションからどれかを指定します。例えば書き換える場合は "SPI Flash Erase, Program, Verify" の一連の動作を指定します。

次にオンボードで接続されている SPI フラッシュメモリを『SPI Flash Options』部で指定します。[Family] 行で、"SPI Serial Flash" がラティスで機能検証済みのメモリー覧から選択する場合、"SPI Serial Flash Beta" は ツールに登録されているもののラティスにて実デバイスでの機能検証が完了していないメモリから選択する 場合です。後者は機能検証が完了次第 Diamond の後のバージョンで "Beta" から移動する可能性があります。 プログラマの機能動作としては同一です。

図 21-14. オペレーションと SPI フラッシュメモリの違	<i>睪択</i>
----------------------------------	-----------

🚱 LatticeECP3 - LFE3-35E	A - Device Properties	? X	
General Device Inform	nation		
Device Operation			
Aaaaaa mada:	DI Elech Peolemund Desembing	SPI Flash Er	ase.Program,Verify
Access mode.	FI Flash Dackground Frogramming	SPI Flash En	ase,Program arifu Ophu
Operation: S	PI Flash Erase,Program,Verity	SPI Flash En	ase All
Programming Options		🛰 SPI Flash Re	ad and Save
		SPI Flash Ca	alculate File Size Checksum
Programming file:		SPI Flash By	/Dess
		Scan SPI Fla	ash Device
Device Options			
Deinitialize part on on			
E Reinitianze part on pr	bgrantenor		
SPI Flash Options			SPI Serial Flash
			SPI Serial Flash Beta SPI Sarial MPAM Bata
Family:	SPI Serial Flash		
Vendor:	AMIC		AMIC
Device:	SPL A25L10P		ATMEL
Package:	209 mil 8-pin SOP		INTEL Macmpix
SPI Programming			NexFlash
Data file eize (Butee)	: 0 Load from	File	Numonyx
Chart address (Hau):			Micron
Start address (Hex).	000000000		SPANSIUN
End address (Hex):	0x00018000		STMicro
Erase SPI part on	programming error		WinBond
Secure SPI flash	golden pattern sectors		
	ОК	Cancel	

次いで [Vendor] を選択後、適切な [Device] と [Package] を指定します。[Data file size (Bytes)] 行は上部にある [Programming file] を指定後、(ファイルを指定しても自動で読み出されない場合)『Load from File』ボタン をクリックすることで入力します。このセルがブランクのまま『OK』ボタンを押すと、図 21-15 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、ご留意ください。

また bit ファイルが指定した SPI フラッシュメモリに収まらないとか、SPI フラッシュメモリ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、入力して解消するようにします。

図 21-15. "Data file size" セルが値 0 の場合のメッセージ



なお、次節で記述するデュアルブートモードで、一方のビットストリームのみを書き換える場合、"Start Address"に当該先頭アドレスを設定します。この情報が正しくないと正常に起動しないなどの障害となりま すので、ご留意ください。

21.2.2 デュアルブート用外部 SPI フラッシュ(LatticeECP シリーズ)

LatticeECP シリーズでは外付け SPI フラッシュメモリに二本のコンフィグレーション・データ (ゴールデ ン・ビットストリームとプライマリ・ビットストリーム)を保存する、いわゆるデュアルブート機能をサポー トします。また、デバイスファミリによって、両ビットストリームのストアするセクタ (開始アドレス)や ジャンプコマンド (0x00400000)のストアされるセクタが異なります。詳細はテクニカルノート TN1216 な どをご参照ください。 プログラマーのオペレーションは図 21-16 のように "Dual Boot SPI Flash Programming" を選択します。

図 21-16. デュアルブート・オペレーションの選択

Jeneral Device	Information	
-Device Operation -		
Access mode:	Dual Boot SPI Flash Programming	•
Operation:	SPI Flash Erase.Program.Verify	•

ここで、書き込みファイル("PROM Hex"ファイルとも呼ばれる)を事前に作成しておく必要があります。 これはユーティリティであるデプロイメント(Deployment)ツールを使用します。21.4節をご参照ください。

なお、Diamond 3.1 以降では、Deployment Tool でマルチブート『Multiple Boot』もサポートされています (LatticeECP5)。

21.2.3 デュアルブート用外部 SPI フラッシュ (MachXO シリーズ)

MachXO シリーズもデュアルブート・モードをサポートしますが、SRAM ベースの LatticeECP シリーズの ように外付けフラッシュメモリに2本のビットストリームを保持する機構とは異なります。オンチップ不揮 発メモリと外付け SPI フラッシュメモリにそれぞれ1本づつのビットストリームを保存しておき、マスター SPI モードと SDM モードの組み合わせで実現しています。オンチップメモリには MachXO2 の場合がプライ マリ、MachXO3L の場合はゴールデンが保存されます。起動は必ずプライマリからで、これらの順序をユー ザが変える手段はありません。

MachXO シリーズの場合、『Operation』の [Access mode] 行(図 21-11 の左下を参照) で選択するのが "SPI Flash Programming" であること、および外部 SPI フラッシュメモリにおくイメージの開始アドレスが "0x010000" でなければならないことを除いて、SPIフラッシュメモリの指定方法などは 21.2.1 項の Lattice ECP シリーズと同様です。

なお、Diamond のスプレッドシートビューなどで指定するグローバル制約(Global Preference)のコンフィ グレーション指定は、MachXO3Lの場合(MASTER_SPI_PORT = ENABLE)、

デュアルブート~ CONFIGURATION = CFG、(TN1279、Dual Boot 項)

マスター SPI モード(単一イメージのみを外部 SPI に置く) ~ CONFIGURATION = EFB_USER

MachXO2 の場合、

デュアルブート~ CONFIGURATION = CFG、CFG_EBRUFM、CFGUFM のどれか(TN1204、Dual Boot 項) マスター SPI モード(単一イメージのみを外部 SPI に置く)~ CONFIGURATION = EFB_USER

なお、MachXO 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)は 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 1kΩ を負荷すること が推奨されています。この際に抵抗をプルダウンにすると、プログラマーからのプログラミングが失敗しま すので、ご留意ください。**外部抵抗は必ずプルアップ**です。それぞれテクニカルノートの "Hardware Checklist" にも明記されていますので、ご参照ください。

21.2.4"カスタム"SPIフラッシュメモリの登録・使用

プログラマーの『SPI Flash Options』部で指定する SPI フラッシュメモリ候補に、意図するデバイスがない 場合に対応するための機能です。最初にカスタム品として手元で登録した後に使用できるようになります。

意図するデバイス行を選択後右クリックするか(図 21-17、左)、デタッチしたプログラマーで [Edit] メ ニューから(図 21-17、右)、いずれも "Custom Devices ..."を選択します。





表示されるウィンドウで最初に Add ボタンをクリックし(図 21-18、左)、次いで適宜名称の登録と詳細な パラメータを選択・入力します(後に呼び出す場合に判別がつくような登録名称とするようにします)。

洋细彩稳面面

図 21-18. SPI フラッシュメモリ登録ウィンドウ

初期画面

			印油五點四田
Sedit Custom I	Device	? ×	🐼 Custom SPI Flash Device
Device family: Device:	SPI Serial Flash Custom 🔻	Add	Device family (Ex: SPI serial flash custom): SPI Serial Flash Custom
		Edit	Device description (Ex: M25P32-VMF6G); Device name (Ex: SPI-M25P32) *: Package (Ex: 16-pin SOIC);
Package:			Device vendor (Ex: STMicro) *: Device density (Ex: 32Mbits) *: AMIC 512 KBits
Device descriptio	n:		Device ID (Ex: 0x15) *: Byte per sector (Ex: 65535) *: 0x05 Hex 65536 Byte Protection options on
	OK	Cancel	* = required field OK Cancel

21.2.5 SPI メモリ内にユーザデータを共存

外付け SPI フラッシュメモリ内に、コンフィグレーションデータとユーザデータを共存することが可能で す。この場合、特段に留意するべき事項はなく、以下の基本的な点を押さえるのみです。

- ・使用するデバイス(ファミリと規模)によってビットストリーム・ファイルや、デュアルブート時の ジャンプコマンドの開始アドレスは一義的に決まるセクターは避ける
- ・バイナリかテキスト (Hex)ファイルかで LSB MSB 順が逆になる

本機能に関してラティスが提供するツールはありませんので、いずれにしろ SPI アクセスするためにファ イル処理手段は、論理回路にしろ組み込みプログラムにしろ、ユーザが作成することになります。

21.3 拡張·特定機能

21.3.1 各種 ID 値の設定(User Code / TraceID / Custom ID Code)

ラティス FPGA では、デバイス内にユーザー固有のコードを格納することが可能です。は MachXO2 の例 で、スプレッドシートビューの『Global Preference』タブを示します。User Code (USERCODE)、TraceID、 そして Custom ID Code がデザインのコンパイル時に設定可能です。これら ID 機能はデバイスファミリによっ て異なり、LatticeECP3 では User Code のみが対応しています。

User Code は 32 ビットで、例えば実装するデザインのバージョン番号など、履歴管理や任意のコードを設定します。MachXO シリーズの TraceID は 64 ビットで、そのうち 56 ビットはラティス出荷時に製造上の固

有の値が入ります(個体レベルで識別できるものです)。残りの8ビットをユーザが任意に用いることが可能 です。

そして同様に MachXO シリーズの Custom ID Code は図 21-19 の Global Preference 内の "sysConfig" 項に含 まれる『MY_ASSP』オプションが ON の時に有効になります。21.1.4 項で言及した "Device ID" は標準品では 規定の値ですが、MY_ASSP=ON 時はこの Custom ID Code がこれに置き換わります。

図 21-19. スプレッドシートビューによるの各 ID 値設定(MachXO2)



上記スプレッドシートビューによる設定以外に制約ファイル 〈file name〉.lpf での指定も可能です。

USERCODE ASCII "abcd";

TRACEID "01010101";

CUSTOM_IDCODE HEX "13579bdf";

このような Diamond の標準フローの一部として設定した ID 情報は、Diamond が生成する書き込みファイル に反映されます。これに対して生成済みファイルやプログラム済みデバイスの ID 情報を書き換える場合は、 ユーティリティツールを用いることができます。USERCODE は『USERCODE エディタ』での編集を(21.4.3 項参照)、TraceID および Custom ID Code の編集は『FEATURE ROW エディタ』(21.4.2 項参照) がサポート しています。

21.3.2 I2C・SPI ポートからのプログラミング (MachXO シリーズ)

MachXO2/3L ファミリでは、ハードマクロ EFB の備える I2C(プライマリ)ポートや SPI ポートから、PC の USB 2.0 ポートを介してプログラマーを用いたプログラミングが可能です。本項ではケーブルとして HW-USBN-2B を用いる前提で記述します。

21.3.2.1 ケーブルの接続

本ケーブルをフライワイヤでターゲットボードに接続しますが、I2C / SPI に関わる信号線のみの割り当て を以下に示します(電源・グランドも省略)。詳細は『UG48 Programming Cables』をご参照ください。

表 21-1. プログラミングケーブルのピンと定義

信 号	ケーブルでの名称	線材の色	PC からの方向	XO2/XO3L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(HW-USBN-2Bのみ)	白・黄ストライプ	Output	SCL	I2C、クロック
I2C Data	SDA(HW-USBN-2Bのみ)	白・緑ストライプ	Output	SDA	I2C、データ

21.3.2.2 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ビットストリームを生成・プログラミングしておく必要があります。

I2C アクセス I2C_PORT = ENABLE

SPI アクセス SLAVE_SPI_PORT = ENABLE

スプレッドシートビューで設定すると制約ファイル <file>.lpf に書き出されますので、手編集も可能です。 MachXO3L に "sysCONFIG" セクション下のアイテムを示します。

図 21-20. スプレッドシートビューの sysCONIFIG 部制約アイテム (MachXO3L)

C) Start Page 🔝 🔃 Reports 🗵	🌠 Spreadsheet View 🗵
ø	Preference Name	Preference Value
лů	Block Path	
🎋 🎽 🎁 🚚 📓 📓 🖓	sysConfig SDM_PORT SLAVE_SPI_PORT i2C_PORT i2C_PORT MASTER_SPI_PORT COMPRESS_CONFIG CONFIGURATION MY_ASSP ONE_TIME_PROGRAM CONFIG_SECURE MCCLK_FREQ JTAG_PORT ENABLE_TRANSFR SHADEDFEDEDINT	DISABLE DISABLE DISABLE ENABLE ON CFG OFF OFF 2.08 ENABLE DISABLE DISABLE
2 11	MUX_CONFIGURATION_PORTS	DISABLE
9	•	4 III
	Assignments Clock Resource Rou	ite Priority Cell Mapping Global Preferences

21.3.2.3 プログラマーのオペレーション

これまでと同様にプログラマーを立ち上げ、アクセスモードとオペレーションを設定します。I2C ポート 経由の場合を図 21-21 に、SPI ポートの場合を図 21-22 に示します。 図 21-21. I2C ポートのアクセス・オペレーション



指定するアクセスモードは "I2C Interface Programming"です。オペレーションは適宜意図するアクションを選択します。

図 21-22. SPI ポートのアクセス・オペレーション



指定するアクセスモードは"Slave SPI Interface Programming"です。オペレーションは適宜意図するアクションを選択します。

21.3.3 暗号化キーと暗号化済み書き込みファイルのプログラミング

本機能はLatticeECPシリーズのみの対応で、MachXOシリーズはサポートしていません。

まず、書き込みファイル(ビットストリーム)の暗号化を実行するには通常の Diamond パッケージに加え て暗号化機能用のパッケージ(Encryption Package)が必要です。ユーザがラティスのウェブサイトからまず リクエストに対する許可の申請をします。その後ダウンロード先 URL を含むメールが送付されてくるので、 ダウンロード、インストールを済ませます。こうした手順は米国の法律に基づいた対処であり、必ず必要で す。日本国内のユーザであれば、問題なく許可されるはずです。

暗号化パッケージをインストールしていないと。本節で紹介しているようなスクリーンダンプ例が得られ ませんのでご留意ください。本項では、パッケージのインストールが完了しているものとして、暗号化キー と暗号化されている書き込み(ビットストリーム)ファイルのプログラミングについて記述します。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(21.4.1.3参照)。書き 込みデータの暗号化処理自体は、Diamond本体のフローやパラメータ設定ではサポートされていません。

ちなみに、コンフィグレーションメモリが揮発性のこれらファミリにおいて、唯一暗号キーを保持するオ ンチップメモリのみ不揮発性であり、かつ OTP(一度のみプログラム可能)です。すなわちキーを一度プロ 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。

	Diamond 3.3 32-bit Encryption Pack for Linux	3.3	10/6/2014		714 KB
	Diamond 3.3 32-bit Encryption Pack for Windows	3.3	10/6/2014		2.9 MB
	Diamond 3.3 32-bit for Linux	3.3	10/6/2014	RPM	1.1 GB
	Diamond 3.3 32-bit for Windows	3.3	10/6/2014	ZIP	1.5 GB
1	Diamond 3.3 64-bit Encryption Pack for Linux	3.3	10/6/2014		720 KB
\sim	Diamond 3.3 64-bit Encryption Pack for Windows	3.3	10/6/2014		2.9 MB
	Diamond 3.3 64-bit for Linux	3.3	10/6/2014	RPM	1.2 GB
	Diamond 3.3 64-bit for Windows	3.3	10/6/2014	ZIP	1.5 GB
1	Diamond Programmer 1.3 Encryption Installer for Linux	1.3	7/18/2011		27 KB
	Diamond Programmer 1.3 Encryption Installer for Windows	1.3	7/18/2011		1.5 MB
	Diamond Programmer 1.3 Standalone for Linux	1.3	7/18/2011	RPM	100.4 MB
	Diamond Programmer 1.3 Standalone for Linux md5 checksum	1.3	7/18/2011	MD5	0.1 KB
	Diamond Programmer 1.3 Standalone for Windows	1.3	7/18/2011	ZIP	34.3 MB

図 21-23. 暗号化対応パッケージのダウンロード先

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、以下の二つのオプションがあります。

1. 暗号キーのみをプログラミングする

2. 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

それぞれで指定するオプションが異なります。以下それらの手順を記述します。

21.3.3.1 暗号キーのみのプログラミング

暗号キーのみをプログラミングする場合は、プログラマーの設定で、図 21-24 に示すようにアクセスモード は [Advanced Security Keys Programming]、オペレーションは [Security Program Encryption Key] を選択します。

図 21-24. アクセスモードとオペレーションの指定

LatticeECP3 - LFE3-3	5EA - Device Propert	ies ? X	
General Device Info	rmation		
Device Operation			
Access mode:	Advanced Security Key	s Programming 🔹	
Operation:	Security Program Encry	rption Key 🔹 🔹	
Device Options		Security Program Encrypt	ion Kev
Reinitialize part on	program error	Security Verify Encryption Security Read Encryption Security Program Key Lo	n Key Key sk
Encryption Key Option	s	Doodanty Programmo y 20	
Enter key:		Load Key	
Confirm key:		Save Key	
Format: ASCII 💌		Show key	
🔽 Program key lock	Type 'yes' to confirm		
		OK Cancel	

書き込み(ビットストリーム)ファイルを暗号化する際に、指定した暗号化キーをファイルに保存できます(<file name>.bek)。同ファイルが存在する場合には、『Load Key...』ボタンで読み込み指定が可能です(図 21-25 のように暗号キーファイルのパスワード入力を求められます)。キーファイルが存在しない場合、 [Enter key] と[Confirm key]欄に暗号キーを入力後、『Save Key...』ボタンをクリックすることで保存できます。

オプションと暗号キーを入力完了後、本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。

なお、"Program key lock" をチェック(イネーブル)した場合、暗号キーファイルをリードバックする手段 は存在しません。したがって、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。別の暗号キーにするか(デバイスにプログラムしていない場合)、同じ操作を実行 してキーファイルを上書きする(キーではなく、キーファイル)の操作は勿論可能です。

図 21-25. 『Load Key...』をクリック後のパスワード入力画面

🚸 Enter Password	? ×
(Maximum 16 characters with	case sensitive)
Hide/Show Password	
ОК	Cancel

21.3.3.2 暗号化済み書き込みファイルのプログラミング / ダウンロード

暗号化済みビットストリーム(書き込みファイル)をダウンロードする場合、選択肢は二つあります。

1. まだ暗号キーをプログラムしていない場合

2. 前項で示す暗号キーをプログラム済みの場合

前者の場合、暗号キーと暗号化済みビットストリームを一度のオペレーションでプログラミング(ダウンロード) します。選択するオプションは図 21-26 のようにアクセスモードが [Advanced Security Production Programming]、オペレーションが [Security Fast Program with Encryption Option] です。

図 21-26. 暗号キーとビットストリームのプログラミング

😓 LatticeECP3 - LAE3-35EA - Device Properties	? X
General Device Information	
Device Operation	
Access mode: Advanced Security Production Program	nming 🔻
Operation: Security Fast Program with Encryption O)ption 🔻
Programming Options	
Programming file: /impl1/d3p5versaE3demo_impl1_encryJTA	G.bit]
Device Options	
🔲 Reinitialize part on program error	
Encryption Key Options	
Enter kev:	Kev
Confirm key:	Key
Format: ASCII 🔹 📃 Sho	ow key
Program key lock Type 'yes' to confirm yes	
ОК	Cancel

同図下部にあるように、『Load Key …』ボタンで書き込みファイルの暗号化時に保存したキーファイルが あれば読み込む、などをして、暗号キーを同時に指定します。[Programming file:] で指定する書き込みファイ ルは、暗号化時にコンフィグレーション・モードを JTAG として選択しておく必要があります (21.4.1.3 項)。 暗号化していない場合 (図 21-27) や、モードが JTAG でない場合などはそれぞれエラーメッセージが表示 されて、本ウィンドウ入力が完了できませんので、ご留意ください。

最下段の [Program key lock] は前項に記述したとおりです。

なお ECP5 では、このオペレーション完了後に別途書き込みファイル(ビットストリーム)をダウンロードする場合に、暗号化済みビットストリームのみを受け付けるオプションが追加されています。

全て設定後、プログラマーに戻りオペレーションを実行します。

図 21-27. キー未書き込みで暗号化済みファイルを指定



次に第二の場合で、暗号キーをプログラムした後に、別オペレーションで暗号化済み書き込みファイルを ダウンロードするケースです。

コンフィグレーション SRAM に直接ダウンロードする際のアクセスモードは [ISC1532 Mode]、オペレー ションは [Fast Program] を選択します。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。

ここで、暗号化の際に指定するコンフィグレーション・モードもチェックされませんので、任意で構いま せん。また、LatticeECP3 ファミリの場合、コンフィグ SRAM にしろ SPI フラッシュメモリにしろ、暗号化済 みでも暗号化していないファイルでもどちらでも問題なく終了できます。ECP5 で既に言及したオプション がある点とは異なります。

21.4 各種ユーティリティツール

21.4.1 デプロイメントツール

書き込みファイル (.jed、.bit) を組み込みコントローラが扱うファイルフォーマットに変換する際や、デュ アルブート機能対応 PROM ファイルを生成する場合などのファイル変換ユーティリティがデプロイメント (Deployment) ツールです。

21.4.1.1 デプロイメントツールの起動

デプロイメントツールの起動は、デタッチしたプログラマーから [Design]-->[Utilities]-->[Deployment Tool] と選択するか(図 21-28、左)、WindowsのStart メニューからLattice Diamond 3.xを選択し、その下"Accessories" を展開すると表示される [Deployment Tool] を選択します (図 21-6、右)。

図 21-28. デプロイメントツールの起動方法



起動後の初期画面は図 21-29 のようになります。[Function Type] として選択できる四つのタイプと、それ ぞれの生成ファイル形式を合わせて示しています。

図 21-29. デプロイメントツール起動後とファンクションタイプ

Diamond Deployment Tool - Getting Started Create New Deployment Function Type: File Conversion Output File Type EEE 1532 EC Data File Open an Existing Deploym Recent Files: Cr/aUSER_SStasks/D33work/xa2lab/tp1_d33/project0.ddt Image: Create Starter Create New Deployment Recent Files: Cr/aUSER_SStasks/D33work/xa2lab/tp1_d33/project0.ddt Image: Create Starter Image: Create Star	Edit Help		
Diamond Deployment Tool - Getting Started Oreate New Deployment File Conversion Output File Type EEE 1532 ISC Data File Open an Existing Deploym Recent Files: C:/sUSER_SStasks/D33work/.xo2lab/tbol_d33/project0.ddt File Conversion Tester Embedded System External Memory Hex Conversion Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Shim VME Embedded		ISC exc JED AT SVF	
Diamond Deployment Tool - Getting Started Create New Deployment Function Type: File Conversion Output File Type EEE 1532 ISO Data File Open an Existing Deploym Recent Files: C:/sUSER_SStasks/D33work/xo2lab/tb1_d33/project0.ddt Elie Conversion Tester Embedded System External Memory SVF = JTAG Chain SVF = JTAG Chain SVF = JTAG Chain SVF = UTAG Chain Advanced SPI Flash sysCONFIG Daisy Chain JTAG Slim VME Embedded	_		
Create New Deployment Eunction Type: File Conversion Output File Type EEE 1532 ISO Data File Open an Existing Deploym Recent Files: Cr/aUSER_SStasks/D33work/xo2lab/ls1_d33/project0.ddt Embedded System File Conversion Tester Embedded System External Memory SVF - JTAG Chain STAFL - Single Device STAFL - Sing	Diamond	Deployment Tool - Getting	3 Started
Function Type: File Conversion Output File Type EEE 1532 ISC Data File Open an Existing Deploym Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d33/project0.ddt Recent Files: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d33/project0.ddt Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d33/project0.ddt Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d32_ISC Data File Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d1_d2 Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d33/project0.ddt Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d33/project0.ddt Image: Conversion Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d1_d12_ISC Data File Image: Cr/aUSER_SStasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work/xo2lab/tb1_d13/dt2_Stasks/D93work	Oreate N	ew Deployment	
Output File Type FEE 1532 ISC Data File Open an Existing Deploym Recent Files: O:/aUSER_SStasks/D93work/xo2lab/jb1_d33/project0.ddt OK Close File Conversion OK Tester Exbedded System External Memory SVF - JTAG Chain SVF - Single Device SVF - Single Device SVF - Single Device STAPL - Single Device SVF - JTAG Chain STAPL - JTAG Chain JTAG Full WARE Embedded JTAG Slim VME Embedded	Function Type	Eile Conversion	
Output Hie Type [EEE 1532 ISC Data File Open an Existing Deploym Recent Files: C:/aUSER_SStasks/D03work/xco2lab/lpb1_d03/project0.ddt CK Close File Conversion CK Tester Exbedded System External Memory SVF - Single Device SVF - Single Device SVF - Single Device SVF - Single Device STAPL - Single Device SVF - Single Device STAPL - Single Device SVF - JTAG Chain STAPL - Single Device JTAG Full WALE Embedded JTAG Slim VME Embedded			
● Open an Existing Deploym Recent Files: C:/aUSER_SStasks/D03work/xxo2lab//b1_d03/project0.ddt OK Close File Conversion OK Close Tester Exbedded System Bitstream External Memory SVF - Single Device SVF - Single Device Dual Boot Advanced SPI Flash SyCONFIG Daisy Chain JTAG Full WAE Embedded	Output File Ty	P ^e IEEE 1532 ISC Data File	· · · · · · · · · · · · · · · · · · ·
Recent Files: C:/aUSER_SStasks/D03work/xo2lab//b1_d33/project0.ddt CK Close CK Close File Conversion Tester Embedded System External Memory Hex Conversion Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain SVF - Single Device SVF - Single Device STAFL - Si	🔘 Open an	Existing Deploym	
OK Close File Conversion Application Specific BSDL File Tester JEDEC File Embedded System Bitstream External Memory SVF - Single Device SVF - JTAG Chain STAPL - JTAG Chain Advanced SPI Flash SYFCONFIG Daisy Chain	Recent Files:	C:/aUSER_SStasks/D33work/>	xo2lab/lab1_d33/project0.ddt 🔹 🛄
Elle Conversion EEE 1532 ISC Data File Tester Application Specific BSDL File External Memory SVF - Single Device SVF - Single Device SVF - Conversion Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Full WAE Embedded			
ELEE 1532 ISC Data File Tester Application Specific BSDL File Embedded System JEDEC File External Memory SVF - Single Device SVF - JTAG Chain STAPL - Single Device Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Full WME Embedded			UK Close
File Conversion Effective Solution Tester Application Specific BSDL File Embedded System Bitstream External Memory SVF - Single Device SVF - JTAG Chain STAPL - Single Device Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Full Wite Embedded	_		
Embedded System External Memory Hex Conversion Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain	T	ester	Application Specific BSDL File
External Memory SVF - Single Device SVF - JTAG Chain SVF - JTAG Chain STAPL - Single Device STAPL - Single Device Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Slim VME Embedded	Er	mbedded System	JEDEC File Bitstream
Hex Conversion Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain	E	xternal Memory	SVF - Single Device
Dual Boot Advanced SPI Flash sysCONFIG Daisy Chain JTAG Sim VME Embedded			SVF - JTAG Chain
Advanced SPI Flash sysCONFIG Daisy Chain JTAG Sim VME Embedded	_	ex Conversion	STAPL - JTAG Chain
JTAG Slim VME Embedded		Juai Boot	
	С Д	Idvanced SPI Flash	ATE
	L A S	Juai Boot Idvanced SPI Flash IysCONFIG Daisy Chain	ATE JTAG Full VME Embedded JTAG Slim VME Embedded Slave SPI Embedded I2C Embedded

21.4.1.2 デュアルブート用 PROM ファイル生成

LatticeECP シリーズをターゲットとする、デュアルブート用の二本のビットストリーム(ゴールデンとプライマリ)は生成済みのものとします。統合されるファイルはHex フォーマットで、通常拡張子はmcs です(後述)。

図 21-30. デュアルブート用 PROM ファイル生成

🚼 Diamond [Deployment Tool - Getting Started
Oreate Net	e w Deployment
Function Type	External Memory 👻
Output File T	E Dual Boot
🔘 Open an E	ixisting Deploym
Recent Files:	C:/aUSER_SStasks/D33work/xo2lab/lab1_d33/project0.ddt 💌 💷
	OK Close

まず最初のウィンドウで、[Function Type] として "External Memory" を、[Output File Type] として "Dual Boot" を選択します (図 21-30)。

図 21-31. ビットストリーム・ファイル 2 本の指定例

ile E	dit Help								
P	1	ISC PERK	ÆD	12	SVF	SVF The	2		
xtern	al Memory:	Dual Bo	ot						
Ste	p1 of 4:5	elect Inp	ut File	(s)					
		File N	lame (*.jed	*.bit	*.rb	<u>t)</u>	Device Family	Device
1 EF	R_SStasks/	File N D33vork,	lame (/Versa	*.jed _chk_[*.bit Dual	*.rb Boot	/platform1_platform1.bit	LatticeECP3	Device LFE3-35EA

OK をクリックすると、使用するファイル2つの指定ウィンドウが表れますので、意図するファイルをブラウズして入力します。上がゴールデン、下がプライマリです。Next ボタンをクリックして進みます。

図 21-32. 生成ファイルのオプション指定例

External Memory: [Jual Boot
Step 2 of 4: Dua	Il Boot Options
Output Format:	Intel Hex
SPI Flash Size (Mb):	64 -
Golden Pattern:	0:/aUSER_SStasks/D33work/Versa_chk_DualBoot/platform1_platform1.bit 🔹
Primary Pattern:	C:/aUSER_SStasks/D03work/Versa_chk_DualBoot/platform1_platform1_bit 💌
📝 Protect Golden S	ector
📝 Byte Wide Bit Mir	Tor
📃 Retain Bitstream	Header
📃 Optimize Memory	Space

次のウィンドウは生成するファイルのオプションです。[Output Format]はIntel Hex、Motorola Hex、Extended Tektronix Hex から選択します。通常は Intel Hex です(拡張子 mcs)。フラッシュメモリのサイズを選択し、 ゴールデンとプライマリが正しいことを確認します。その下に4つのオプションがあります。

- · Protect Golden Sector ゴールデンに保護を掛けます(万が一のため、オンにしておきます)
- ・ Byte Wide Bit Mirror ビット反転。Hexファイル出力では基本的に反転ですのでチェックします
- · Retain Bitstream Header ヘッダ情報を保持する場合にチェックします。通常は残します(オフ)
- · Optimize Memory Space SPI フラッシュメモリのサイズに余裕がない場合です。オンにしなければい けない状況は推奨しません

通常は図 21-32 の例のようにゴールデンの保護とビット反転(ミラーリング)をオンにします。Next ボタン を押して先に行きます。生成ファイルとフォルダ位置の確認・指定ウィンドウが表示されますので(図 21-33)、問題なければ右下の Next ボタンをクリックします(図 21-33 では割愛)。

図 21-33. ファイル名とフォルダ指定



因みに、コンフィグレーション用ビットストリームのファイル内のバイトデータ表記は、バイナリファイルの場合が LSB から MSB の順で書かれ、Hex ファイルの場合は MSB から LSB の順で書かれた形式でなければなりません。例えば、ビットストリームのプリアンブル(先頭を示すキーワード)は元々バイナリファイルでの16 進表記 0xBDB3 ですが、Hex ファイルでの表記は 0xBDCD となります(0xBD の反転が 0xBD、0x03は 0xCD)。

次にファイル生成します。右下の Generate ボタンをクリックするか、上部のアイコン 🌌 をクリックしま す。ウィンドウ下部のコンソールにログ表示されますので、"... successfully"となれば正常終了です。右上の "View File" ボタンをクリックすると、テキストビューアーが立ち上がり、生成したファイルが閲覧できます。 図 21-34. ファイル生成とチェック

🛐 Diamond Deployment Tool- project0.ddt*
File Edit Help
2 10 II
External Memory: Dual Boot Step 4 of 4: Generate Deployment
View File
Deployment Tool Summary
Input File 1: C:/aUSER_SStasks/D33work/Versa_chk_Dual
"C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/platform1_platform1.bit" -format int -flashsize 64 -protect -mirror -of "C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/platform1_platform1.mcs
•
C:¥aUSER_SStasks¥D33work¥Versa_chk_DualBoot¥platform1_platform1.mcs
Lattice Diamond Deployment Tool has exited successfully.
III
Previous

また アイコンをクリックするなどしてログファイルをチェック(保存)することを推奨します。内容としてジャンプコマンド、プライマリとゴールデン両ビットストリームの開始・終了アドレス(セクター)

としてジャンプコマンド、ブライマリとゴールデン両ビットストリームの開始・終了アドレス(セクター) 情報が書き出されています。特に開始セクター(アドレス)は一方のビットストリームのみを更新して書き 換える場合に必要となります。

デバイスファミリ毎に両ビットストリームとジャンプコマンドのセクタは決められています。デプロイメ ントツールは自動的に対処してくれますが、ユーザが何らかの別の手段で生成する場合は注意が必要です。 詳細はテクニカルノート TN1216 Table.5 をご参照ください。

終了するには、ウィンドウ右上隅のXをクリックしますが、プロジェクトファイルddtとして保存するかどうかの確認を促されます。保存しておけば、後で繰り返し作業する場合などに有用です。

図 21-35. SPI フラッシュメモリの保護解除(MachXO3L)

	indere in the second seco		
Jevice Operation			
Access mode:	SPI FI	lash Programm	ing 🔻
Operation:	SPI Flash U	Jnlock Device	
SPI Flash Options			
Family:		SPI Serial Fl	ash 🔻
Vendor:		SPANSION	•
Device:		SPI-S25FL2	04K 🔻
Package:		8-pin SOIC	•
SPI Programming			
Data file size (Bytes): 131072		Load from File
Start address (Hex):		0x00000000	•
End address (Hex):		0x0001 0000	•
Erase SPI part on programming error			
📃 Secure SPI flash	golden pat	tern sectors	

ここで、MachXO3L で保護をかけたプライマリ・イメージを更新して書き換える際には、始めに保護の解除が必要です。[Operation] で [SPI Flash Unlock Device] を選択します。最下段の "Secure SPI flash golden pattern sectors" はチェックしないようにします (MachXO2 は未サポートです)。

21.4.1.3 書き込みファイルの暗号化

書き込みファイル(ビットストリーム)の暗号化を実行するには通常の Diamond パッケージに加えて暗号 化機能用のパッケージ(Encryption Package)が必要です。ユーザがラティスのウェブサイトからまずリクエ ストに対する許可の申請をします。その後ダウンロード先 URL を含むメールが送付されてくるので、ダウン ロード、インストールを済ませます。こうした手順は米国の法律に基づいた対処であり、やむを得ません。 日本国内のユーザであれば、問題なく許可されるはずです。

既に言及したとおり、暗号化パッケージをインストールしていないと。本節で示すようなスクリーンダン プ例が得られませんのでご留意ください。

作業手順は次の通りです。まずデプロイメントツールを起動し、[Function Type] は "File Conversion" に、 [Output File Type] は "Bitstream" を選択します (図 21-36)。既に作成済みデプロイメントツール・プロジェ クトがあれば、下部の『Open an Existing Deploym』をチェックして ddt ファイルをブラウズすることで指定 します。OK ボタンをクリックします。

図 21-36. 暗号化作業の開始

🛃 Diamond [Deployment Tool - Getting Started
Oreate Net	w Deployment
Function Type:	File Conversion 🔹
Output File Typ	e Bitstream 🔻
🔘 Open an E	xisting Deploym
Recent Files:	C:/aUSER_SStasks/D33work/xo2lab/lab1_d33/project0.ddt 💌 💷
	OK Close

次のステップは暗号化するファイルの指定です(図 21-37)。ブラウズして Next ボタンで進みます。

図 21-37. 暗号化対象ファイルの指定

Diamond Deployment Tool- project0.ddt*	198.34		. 🗆 🗙
File Edit Help			
🐣 🔂 🔚 ISC 🚥 🗷 🕬 🕅	u 🎽 🔤		
File Conversion: Bitstream Step 1 of 4: Select Input File(s)			
File Name (*.jed *.bit *.rbt)	Device Family	Device	
1 ityDDRG/impl1/clarity_ddrg_impl1.bit	ECP5UM	LFE5UM-45F	
		Previous	Next

次はオプションの詳細設定です(図 21-38)。通常は特に変更しません。USERCODE を編集することも可能です。

最下部の『Encryption』ボタンをチェックして有効にし、[Configuration Mode]の選択をハードウェアや Diamondのオプション指定と合致させます。非暗号化ビットストリームはこのコンフィグレーション・モー ドには非依存ですが、暗号化ビットストリームでは合致することが要件です。プログラマーを用いて JTAG からコンフィグレーションする場合は"jtag"です。 図 21-38. 暗号化詳細設定

Diamond Deploym	ient Tool- project0.ddt* 📃 🗖 🔀
File Edit Help	
💾 🖆 🖶 📑	C 🔤 JED 🚛 SVF 🌃 🌽 🔟
File Conversion: Bit	tstream
Step 2 of 4: Bits	tream Options
Ounut Format:	Binary Bitstream
Program Security Bit:	
Verify ID Code:	Default
Erequency:	
Compression:	
CRC Calculation:	
USERCODE Format:	
USERCODE:	000000000 slave_pcm
📃 Byte Wide Bit Min	ror 7 isni
📃 Retain Bitstream I	Header spim
Encryption	slave spi
Configuration Mode	spi
Encryption Key:	
	Edit Key Save Key
	Previous Next

次に暗号化キーを入力します。『Edit Key』ボタンをクリックして表示される図 21-39 のウィンドウで指定 します。入力後 "OK" をクリックして抜けると、図 21-38 で『Save Key』ボタンがアクティブになりますの で、(必要に応じて)保存します。

図 21-39. 暗号化キーの編集・入力

Encryption Key Setup		? X
Encryption Key:	•	ОК
Hide Encryption Key		Cancel
ASCII	•	Load From File
ASOII		.11

図 21-40. 暗号化キーの保存

		, , , , , , , , , , , , , , , , , , , ,	-15
整理▼ 新しいフォル	レダー		• •
涬 お気に入り	▲ 名前 [▲]	更新日時	種類
	👢 backup	2015/03/30 10:04	ファイル フォ
门 ライブラリ	≡ 📙 chk_enc_impl1.dir	2015/03/30 10:04	ファイル フォ
	👢 coreip	2015/03/30 9:59	ファイルフォ
🚜 ホームグループ	📜 dm	2015/03/30 9:59	ファイルフォ
	📕 📕 physical_plus	2015/03/30 9:59	ファイル フォ
💵 コンピューター	👢 synlog	2015/03/30 9:59	ファイル フォ
	👢 syntmp	2015/03/30 10:04	ファイルフォ、
🧞 US (C:)	▼ <		+
ファイル名(N):			-
ファイルの種類(T): En	cryption Key Files (*.bek *.BEK)		-

ファイル名を入力して『保存』をクリックすると、パスワードの入力を求められます。知る必要のない人 に見られないようにするための措置です。問題なければ "Save key successfully" というメッセージの小ウィン ドウがポップアップします。

図 21-41. 暗号化キーファイルのパスワード

Encryption File Password	? X
Password (Maximum 16 characters− Case sensitive) ●●●●●●●●●●●●	ОК
V Hide password	Cancel

次にキーが入力された状態で図 21-38 に戻りますので、『Next』で次ステップに進み、出力ファイル名の 指定・確認を行います(図 21-42)。元のビットストリーム・ファイルと異なる名称にする場合にファイル名 とフォルダを明記します。『Next』で次に進みます。

図 21-42. 出力ファイル名の指定・確認

Diamond Deployment Tool- project2.ddt*	
File Edit Help	
* * -	
File Conversion: Bitstream	
Step 3 of 4: Select Output File(s)	
Output File1: C:/aDiWork/D33ugj/encry/chk_enc_impl1_encrptd.bit	▼
Previou	s Next

図 21-43. 暗号化ビットストリームの生成

File Conversion: Bitstream Step 4 of 4: Generate Deployment View File USERCODE: 00000000 Byte Wide Bit Mirror: Off Retain Bitstream Header: Off Encryption: On Configuration Mode: slave_scm BEK File: C:/aDiWork/D33ugj/encry/impl1/chkencrypt_ke: Encryption Key: .LatticeSemiTest Encryption Key Format: ASCII BEK File Password: latticesemi Output File: C:/aDiWork/D33ugj/encry/chk_enc_impl1_encrptd.bitv Command Line Command Line Command Line File C:YaDiWork/D33ugj/encry/impl1 Ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/impl1 Ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/timpl1 Ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/timpl1 Ychk_enc_impl1.bit was generated successfully. The File C:YaDiWork/D33ugj/encry/thk_enc_impl1_encrptd.bit was generated successfully. The File C:YaDiWork/D33ugj/encryYchk_enc_impl1_encrptd.bit was generated Successfully. The File C:YaDiWork/D33ugj/encryYchk_enc_impl1_encrptd.bit was generated Successfully. The File Diversed Declarement Teal her exclude the processing the C:YaDiWork/D33ugj/encryYchk_enc_impl1_encrptd.bit was generated Successfully. The File C:YaDiWork/D33ugj/encryYchk_enc_impl1_encrptd.bit was generat	File Edit		
Step 4 of 4: Generate Deployment View File USERCODE: 00000000 Byte Wide Bit Mirror: Off Retain Bitstream Header: Off Encryption: On Configuration Mode: slave_scm BEK File: C:/aDiWork/D33ugj/encry/imp11/chkencrypt_ket Encryption Key: .LatticeSemiTest Encryption Key Format: ASCII BEK File Password: latticesemi Output File: C:/aDiWork/D33ugj/encry/chk_enc_imp11_encrptd.bit v Command Line "Orlean/Glamond/33_x64/bin/nt64/ddtcmd" -oft -bit -dev LFE2M20SE -if Cost colspan="2">Cost colspan="2">Cost colspan="2">Estream Processing file C*ApDiWorkVD33ugWencryVimp1 Vchk_enc_imp1 bit processed successfully. File C.YaDiWorkVD33ugWencryVimp1 Vchk_enc_imp1 bit processed successfully. The file C*ApDiWorkVD33ugWencryVimp1 Vchk_enc_imp1 encrytd.bit was generated successfully. The file C*ApDiWorkVD33ugWencryVehk_enc_imp1 encrytd.bit was generated successfully.	File Conve	ersion: Bitstream	
View File USERCODE: 00000000 Byte Wide Bit Mirror: Off Retain Bitstream Header: Off Encryption: On Configuration Mode: slave_scm BEK File: C:/aDiWork/D33ugj/encry/impl1/chkencrypt_ke: Encryption Key: LatticeSemiTest Encryption Key Format: ASCII BEK File Password: latticeSemi Output File: C:/aDiWork/D33ugj/encry/chk_enc_impl1_encrptd.bit Command Line Command Line File C:YaDiWork/D33ugj/encry/impl1 Ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/impl1 ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/impl1 ychk_enc_impl1.bit File C:YaDiWork/D33ugj/encry/chk_enc_impl1.bit was generated successfully. The file C:YaDiWork/D33ugj/encry/chk_enc_impl1.encrptd.bit was generated	Step 4	of 4: Generate Deployment	
USERCODE: 0000000 Byte Wide Bit Mirror: Off Retain Bitstream Header: Off Encryption: On Configuration Mode: slave_scm BER File: C:/aDiWork/D3Jugj/encry/impl1/chkencrypt_key Encryption Key Format: ASCII BER File Password: latticeSemi Output File: C:/aDiWork/D3Jugj/encry/chk_enc_impl1_encrptd.bit + Command Line C-/soc/damond/3.3.x04/bin/nt64/ddtcmd ^{eff} -oft -bit -dev LFE2M20SE -if Generating Binary Bitsream Frocessing file C:#aDiWork/D3Jugi/encry/impl1 ychk_enc_impl1_bit File C:#aDiWork/D3Jugi/encry/impl1 ychk_enc_impl1_bit File C:#aDiWork/D3Jugi/encry/chk_enc_impl1_bit was generated successfully. The file C:#aDiWork/D3Jugi/encry/chk_enc_impl1_encrptd.bit was generated successfully.			View File
Command Line C./Isoc/diamond/3.3_x64/bin/nt64/ddtcmd ²¹ -oft -bit -dev LFE2M20SE -if Generating Binary Bitstream Processing file C.¥aDiWork¥D33ugi¥encry¥impl1 ¥chk_enc_impl1.bit File C.¥aDiWork¥D33ugi¥encry¥chk_enc_impl1_encrptd.bit was generated successfully. The file C.¥aDiWork¥D33ugi¥encry¥chk_enc_impl1_encrptd.bit was generated successfully.	Output	Retain Bitstream Header: Off Encryption: On Configuration Mode: slave_scm BEK File: C:/aDiWork/D33ugj/encry/impl1/chkenc Encryption Key: .Lattice3emiTest Encryption Key Format: ASCII BEK File Password: latticesemi File: C:/aDiWork/D33ugj/encry/chk enc impl1 enc	rypt_kes
Command Line "Or/Isco/diamond/3.3_x64/bin/nt64/ddtcmd" -oft -bit -dev LFE2M20SE -if Generating Binary Bitstream Processing file C¥aDiWork¥D33ugi¥encry¥impl1¥chk_enc_impl1.bit File C¥aDiWork¥D33ugi¥encry¥impl1¥chk_enc_impl1_bit processed successfully. The file C¥aDiWork¥D33ugi¥encry¥chk_enc_impl1_encrptd.bit was generated successfully.	<	III	
Lattice Diamond Deployment 1001 has exited successfully.	Commar "C:/Isco. Generatin Processin File C:¥al The file C successf Lattice D	d Line (diamond/3.3,x64/bin/nt64/ddtcmd ² -oft -bit -dev LFE2M2OSE -if (uk4800 ²) Ig Binary Bitstream Ig file 0:¥aDiWork¥D33ug¥encry¥impli ¥chk_enc_impli .bit DiWork¥D33ug¥encry¥impli ¥chk_enc_impli .bit processed successful 0:¥aDiWork¥D33ug¥encry¥chk_enc_impli _encrptd.bit was generated ully. iamond Deployment Tool has exited successfully.	lly.

最後が暗号化ビットストリームの生成です(図 21-43)。右下の『Generate』ボタンか上部の 🍊 アイコン をクリックします。下段の窓枠で図のように"... excited successfully"と表示されれば正常終了です。

アイコン 🧧 をクリックしてログの確認や、『View File』をクリックしてテキストエディタで生成物の内容 をチェックできます。

最後にデプロイメント・プロジェクト ddt を保存後、ウィンドウ外枠の右上すみの X 印をクリックして終 了します。

21.4.2 フィーチャ行エディタ(MachXO2/XO3L)

ハードマクロ EFB がサポートする I2C や SPI インターフェイスを介して、外部コントローラから最初に (デバイスがブランク時や消去状態で)プログラミングする際には、フィーチャ行(Feature Row)は通常最 低一度はアクセス(ライト)することになります。しかし、フィーチャ行を変更するのは開発時、及び量産 (出荷)時のみとすることを強く推奨しています。これは、特に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。ノイズが多いなど望ましくない特定の環境下では、意図せぬア クセスのために誤った値を書き換えてしまい、従って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、結果的にその後のアップデートを行えなくなるケースが生起する可能性が否定できません。

何らかのやむを得ない事由(含デバッグ等)でフィーチャ行に書かれているビットの設定値を確認したり、 編集することが避けられない場合はフィーチャ行エディタが活用できます。起動するには、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 (図 21-44)、その後 Tools --> Feature Row Editor を指定して行います。また Programming File Utility について は、図 21-28 に示すと同様の手順で Windows のスタートメニューから立ち上げることもできます。

図 21-44. フィーチャ行エディタの起動



図 21-45 左のような起動直後、赤丸で示すブラウズボタンで jed ファイルを指定し、デバイスタイプ等を 読み取ってアクセスの準備をします(MachXO2 の jed ファイルを指定した例)。次に "Read" ボタンをクリッ クしてファイルからフィーチャ行情報を抽出して、同右のように表示します。

図 21-45. 起動後の表示例



IEDEC ファイルを読み込んだ後

黒色で表記されているビットのいづれかをクリックするたびに、当該ビット表示がトグルします。赤色表記のビットは編集できません。編集後、"Save"や"Save As"で Jedec ファイルとして(別名で)保存します。

なお、フィーチャ行エディタでは 21.3.1 項で言及した TraceID および Custom ID Code の編集もサポートします。図 21-45 では示していませんが、本ウィンドウの右側にフィールドがあります。

21.4.3 USERCODE エディタ

特にユーザーコードを編集するツールが USERCODE エディタです。起動するには、図 21-44 のウィンド ウで [Tools]-->[USERCODE Editor...] と選択します。

操作手順は前項のフィーチャ行エディタと同様です。表示されるウィンドウ(右上)で [Device] 行右端の ブラウズボタンで該当ビットストリーム・ファイルを指定します。デバイスタイプが読み込まれ、表示に反 映されます(左下)。"Read"ボタンをクリックして値を抽出して表示します(右下)。これで所望の値に編集 して保存し直します。デフォルトでは16進表記になっています。

図 21-46. USERCODE エディタの起動から編集まで

🥰 Feature Row Editor	
🔜 Control Register0 Editor	Device:
USERCODE Editor	USERCODE Fo Hex
1 Security and Persistent Fields Editor	USERCODE:
-	Read Bave Save As Close
USERCODE Editor	USERCODE Editor
Device: LCMX02-7000HE	Device: LCMX02-7000HE
Device: LCMXO2-7000HE	USERCODE Editor
USERCODE Editor	USERCODE Editor

21.4.4 ダウンロード・デバッガ

意図せぬ理由で、ISP/JTAG プログラミングが失敗する場合、いったん書き込みファイル(.bit、.jed)を SVF ファイルに変換し、これを用いたデバッグを実行することで原因を究明できる場合があります。ダウン ロード・デバッガ(Download Debugger)は、ダウンロードケーブルで接続された実デバイスを動作させなが ら、SVF ファイルをデバッグするためのツールです。組み込みコントローラ用 VME ファイルのデバッグで も使用できますが、本節では SVF ファイルを例にとり記述します。

図 21-47. デプロイメントツールのファンクションタイプ選択

🚼 Diamond D	eployment Tool - Getting Started		? X
Oreate New	Deployment		
Function Type:	Tester		•
Output File Type	SVF - Single Device		-
Open an Ex	<u>SVF – Single Device</u> SVF – JTAG Chain STAPL – Single Device STAPL – JTAG Chain		
Recent Files.	ATE	ок	Close

SVF ファイルが未生成の場合、まずデプロイメントツールでフォーマット変換します。起動方法は 21.4.1 節をご参照ください。ファンクションタイプは "Tester"、出力ファイルタイプは SVF を選択します。JTAG チェイン内にターゲットの単一デバイスのみか、ターゲットを含む複数デバイスかによりオプション指定等 が異なります。

ステップ1でソースファイルを指定後、ステップ2で図21-48のようにオペレーション (Operator) と各オプ ションを指定します。選択できるオペレーションはターゲットデバイスに依存します。

図 21-48. SVF 変換の各種オプション指定

Diamond Deployment Tool- project0.ddt	*			
File Edit Help				
🐣 🖻 🔚 ISC 📼 IED 🕮 SVF	🔤 🔀 🚾			
Tester: SVF - Single Device				
Step 2 of 4: SVF Options				
Operation Fast Program	▼			
🔽 Write Header and Comments				
🔲 Rev D Standard SVF				
RUNTEST from Rev C				
🔲 For Erase, Program, and Verify Operations, Skip Verify				
Include RESET at the End of the SVF File				
🔲 Set Maximum Data Size per Row (Kbits):	64 *			

その後ステップ3で出力ファイル名とフォルダを指定し、ステップ4で変換を実行(Generate)します(図21-49)。『... successfully』と表示されれば終了です。作業を繰り返す際に備えてデプロイメント・プロジェクト (.ddt)として保存しておきます。次回からはファイルのオープンで一連の作業が迅速に行えます。

図 21-49. SVF ファイルの生成(変換)

🔀 Diamond Deployment Tool- project0.ddt*
File Edit Help
🗠 🖻 🖃 🔤 📼 🔤 🖼 🔛
Tester: SVF - Single Device
Step 4 of 4: Generate Deployment
Launch Debugger
Deployment Tool Summary
Input File: C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx
Options:
Operation: Fast Program Write Header and Comments: On
4
"C:/luSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_test_tx_rx_test.bit" -dev "C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_test_tx_rx_test.bit" -dev LFE3-35EA -op "Fast Program" -of "C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_test_tx_rx_test.svf"
Generate Single SVF file: Start
Device 1 LFE3-35EA:Fast Program
Build SVF File Operation: Successful.
Lattice Diamond Deployment Tool has exited successfully.
Previous Generate

以上で準備が整いましたのでダウンロード・デバッガを起動します。図 21-49 中の右上にある『Launch Debugger』をクリックすることで連続して作業ができます。この場合初期画面は図 21-51 のようになり、生成した SVF をロードした状態で立ち上がります。

或いは一度デプロイメント・ツールを終了し、図 21-50 のように、プログラマーから起動することも可能 です。この場合、何もファイルをロードしていないブランク状態でデバッガが立ち上がりますので、File ---> Open と辿りデバッグのターゲットとする SVF ファイル (別作業の場合は STP、VME ファイルなどが選択 可)を指定します。これにより、図 21-51 のような初期状態になります。

図 21-50. プログラマーからダウンロード・デバッガの起動

😔 Programmer - J	platform1.xcf	
File Edit View	Design Window Help	
Enable Status	Scan Check XCF Project	Operation
1 🔽	掛 Program	Flash Erase,Program,Verify
	Log	
	💈 Clear Log File	
	Utilities 🔹 📢	Deployment Tool
	BSCAN Configuration	Download Debugger
		Model 300 Programmer
		Programming File Utility

図 21-51. SVF ファイル読み込み後のデバッガ初期画面

File Edit Command Configuration Window Help <td< th=""><th>Do 🕑</th><th>wnloa</th><th>d Debugger -</th><th>[C:/aUSI</th><th>R_SSt</th><th>asks/D3</th><th>Bwork/</th><th>Versa</th><th>_chk_</th><th>Dual</th><th>Boot/</th><th>tx_rx</th><th>_test_t</th><th>x_rx_te</th><th></th><th></th><th>X</th></td<>	Do 🕑	wnloa	d Debugger -	[C:/aUSI	R_SSt	asks/D3	Bwork/	Versa	_chk_	Dual	Boot/	tx_rx	_test_t	x_rx_te			X
<pre></pre>	File	e Edi	t Command	Configu	ration	Windov	v Help	0								Ŀ	- 8
<pre>1 2 2 2 2 2 2 2 2 3 2 2 3 2 2 3 2 3 2 3</pre>	<u>@</u>	- 🗗		с° %	Ē	۰ ا	5	*		GO	2	8	LOG				
<pre> 2 ! Lattice Semiconductor Corp. 4 ! Serial Vector Format (.SVF) File. 5 ! User information: 6 ! File name: C:\aUSER_SStasks\D33work\Versa_chk_DualBoot\tx_rx_test_tx_rx_test.svf 7 ! CREATED BY: Diamond Deployment Tool 3.3 8 ! CREATION DATE: Fri May 08 16:59:50 2015 9 ! Device: LFE3-35EA Fast Program C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t 10 ! LATTICE_NOTE "Device" "LFE3-35EA" 11 12 13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 ! Devidet LFE3-140 18 ! NewIMO Mark the Life Version 1.40 18 ! NewIMO Mark the Life Version 1.40 19 ! Devidet Lformather CRC: 0xFBCF 20 21 ! Initialize 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 IDTR 0; 37 TIR 0; 38 ENDDR DEPAUSE; 39 ENDDR IRPAUSE; 40 PREQUENCY 1.00e+006 HZ; 41 ' Check the IDCODE 44 ' Check the IDCODE 44 ' Check the IDCODE 44 44 ' Check the IDCODE 44 44 ' Check the IDCODE 45 45 45 45 45 45 45 45 45 45 45 45 45</pre>	1	L															
<pre>1 Lattice Semiconductor Corp. 2 Listil Vetor Format (.SVF) File. 1 User information: 2 File name: C:\aUSER_SStasks\D33work\Versa_chk_DualBoot\tx_rx_test_tx_rx_test.svf 2 (CREATED BY: Diamond Deployment Tool 3.3 3 (CREATION DATE: Fri May 08 16:59:50 2015 3) Device: LFE3-35EA Fast Program C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t 1 1 1 1 1 LATTICE_NOTE "Device" "LFE3-35EA" 1 1 1 LATTICE Semiconductor Corporation Bitstream 1 1 Lattice Semiconductor Corporation Bitstream 1 1 Lattice Semiconductor Corporation Bitstream 1 1 Lattice Semiconductor Corporation 3.4.0.80 1 Elistream Status: Final Version 1.40 1 Device: LFE3-35EA Fast Program C:/aUSER_Stasks/D33work/Versa_chk_DualBoot/tx_rx_t 1 Lattice Semiconductor Corporation Bitstream 1 1 ReadDoot: 6ff 2 Lattice Semiconductor Corporation 1.40 1 Device: fraction for the transment for the formation for the fo</pre>	2	2															
<pre>Secial vector formation: S = 1 User information: F = File name: C:\AUSER_SStasks\D33work\Versa_chk_DualBoot\tx_rx_test_tx_rx_test.svf CREATED BY: Diamond Deployment Tool 3.3 CREATED NDATE: Fri May 08 16:59:50 2015 Device: LFE3-35EA Fast Program C:\AUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t LATTICE_NOTE "Device" "LFE3-35EA" LATTICE_NOTE "Device" "LFE3-35EA" LATTICE_Semiconductor Corporation Bitstream S = Version: Diamond Version 3.4.0.80 E = Bitstream Status: Final Version 1.40 L = ReadDatk: off ReadDatk: off E = Bitstream CRC: 0xFBCF I = ReadDatk: off E = Bitstream CRC: 0xFBCF I = Row_Width :3412 A = Address_Length :1 HDR 0; HIR 0; HIR 0; FREQUENCY 1.00e+006 HZ; FREQUENCY 1.00e+006 HZ; TATE IDLE; I = Calif</pre>		3	Lattice Se	emicondu	ctor (orp.	10										
<pre> File name: C:\aUSER_SStasks\D33work\Versa_chk_DualBoot\tx_rx_test_tx_rx_test.svf CREATED BY: Diamond Deployment Tool 3.3 CREATION DATE: Fr: May 08 16:59:50 2015 Device: LFE3-35EA Fast Program C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t LATTICE_NOTE "Device" "LFE3-35EA" LATTICE_NOTE "Device" "LFE3-35EA" Lattice Semiconductor Corporation Bitstream I Lattice Semiconductor Corporation 1.40 Design Nume: top: test tx rx test.ncd ReadBack:</pre>	5	5	User info:	rmation:	mac (.												
<pre>7 ! CREATED BY: Diamond Deployment Tool 3.3 8 ! CREATION DATE: Fri May 08 16:59:50 2015 9 ! Device: LFE3-35EA Fast Program C:/AUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t 10 ! LATTICE_NOTE "Device" "LFE3-35EA" 11 12 13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 Design nume: ty my test iv ry test.nod 18 ReadBack: Sff 19 ReadBack: Sff 20 Bitstream CRC: 0xFBCF 27 28 29 30 ! Initialize 31 Address_Length :1 32 Row_Width :3412 33 Address_Length :1 34 HDR 0; 35 HIR 0; 36 HDR 0; 36 HDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 44 Check the IDCODE 44 WE</pre>	6	5	File name	: C:\aUS	ER SSt	asks\D3	3work	\Vers	a ch	k Dua	alBoo	t\tx	rx te	st tx	rx tes	t.svf	
<pre>8 CREATION DATE: Fri May 00 16:59:50 2015 9 Device: LFE3-35EA Fast Program C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t 10 LATTICE_NOTE "Device" "LFE3-35EA" 11 12 13 14 Lattice Semiconductor Corporation Bitstream 15 Version: Diamond Version 3.4.0.80 16 Bitstream Status: Final Version 1.40 17 Deaton name: to one test tx rx test.ncd 18 ReadDack: Off 19 Security: Off 26 Bitstream CRC: 0xFECF 27 28 29 30 Initialize 31 Row_Width :3412 32 Row_Width :3412 33 Address_Length :1 44 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 Check the IDCODE 44 Check the IDCODE 45 Check the IDCODE 45</pre>	7	7	CREATED B	Y: Di	amond	Deploym	ent T	ool 3	3.3	_					_		
<pre>9 ! Device: LFE3-35EA Fast Program C:/aUSER_SStasks/D33work/Versa_chk_DualBoot/tx_rx_t 10 ! LATTICE_NOTE "Device" "LFE3-35EA" 11 12 13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 1 Design mamma to the the two rests.ned 18 ! ReadDack: Off 19 1 Design mamma to the two rests.ned 19 1 Design mamma to the two rests.ned 24 ! ReadDack: Off 25 ! Security: Off 26 ! Bitstream CRC: 0xFECF 27 28 29 29 29 29 29 29 20 20 21 Initialize 21 22 ! Row_Width :3412 23 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 36 ENDDR DEPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 41 STATE IDLE; 42 44 ! Check the IDCODE 4</pre>		3	CREATION	DATE: Fr	i May	08 16:5	9:50	2015									
<pre>11 LATTICE_NOTE "Device" "LFE3-35EA" 11 12 13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 I DESIGN NAME: transment transme</pre>	9	9	! Device: L	FE3-35EA	Fast	Progra	um C:/	aUSEI	_sst	asks.	/D33v	ork/	Versa_	chk_Du	alBoot	/tx_r	x_t
<pre>12 13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 ! Design name, by that tx rx test.ncd 24 ! ReadBack: Off 25 ! Security: Off 26 ! Bitstream CRC: 0xFBCF 27 28 29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 30 ENDIR IRPAUSE; 31 ENDIR IRPAUSE; 32 ENDIR IRPAUSE; 33 ENDIR IRPAUSE; 44 ! Check the IDCONE 45 ! Check the IDCONE 45</pre>	10		LATTICE_N	OTE "Dev	ice" '	'LFE3-35	EA"										
<pre>13 14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 I Design name: transmost tx rx test.nod 24 ! ReadBack: Off 25 ! Security: Off 26 ! Bitstream CRC: 0xFBCF 27 28 29 30 ! Initialize 31 22 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDDR DRPAUSE; 39 ENDDR IRPAUSE; 39 ENDDR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the INCONE</pre>	12																
<pre>14 ! Lattice Semiconductor Corporation Bitstream 15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 17 Design name. + + + + + + + + + + + + + + + + + + +</pre>	13	3															
<pre>15 ! Version: Diamond Version 3.4.0.80 16 ! Bitstream Status: Final Version 1.40 24 ! ReadDack: Off 25 ! Security: Off 26 ! Bitstream CRC: 0xFBCF 27 28 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 41 STATE IDLE; 42 44 ! Check the INCONE</pre>	14		Lattice S	emicondu	ctor (Corporat	ion B	itsti	eam								
<pre>16 ! Bitstream Status: Final Version 1.40 17 I Design name: toget to rx test.nod 24 ! ReadBack: off 25 ! Security: Off 26 ! Bitstream CRC: 0xFBCF 27 28 39 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the INCODE 44 ! Check the INCODE 45 ENDIR INCOME INC</pre>	15	5	Version:		Diamor	nd Versi	on 3.	4.0.8	80								
<pre>11 Design name, by the bat ty ry test.ned 24 ReadDack: off 25 Security: Off 26 Bitstream CRC: 0xFBCF 27 28 29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 44 ! Check the IDCODE 44 ! Check the IDCODE 44 ! Check the IDCODE 45 ! Check the</pre>	16	5	! Bitstream	Status:	Final	L Versio	on 1.4	0									
<pre>25 ! Security: Off 26 ! Bitstream CRC: 0xFBCF 27 28 29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the IDCODE 44 ! Check the IDCODE 45 Col:1 INFE</pre>	<u>17</u> 24	7	Pesion na	ne: 1.x 1	to at	ty ry	test.	ncd									
<pre>26 ! Bitstream CRC: 0xFBCF 27 28 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the INCONF 44 ! Check the INCONF 44 ! Check the INCONF 45 Col:1 INFE</pre>	25	5	Security:	Off													
<pre>27 28 29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 43 44 ! Check the INCODE 44 ! Check the INCODE 45 Col:1 INCODE 45 Col:</pre>	26	5 !	! Bitstream	CRC: 0x	FBCF												
<pre>29 29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDER DEPAUSE; 39 ENDIR IRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the INCODE 44 ! Check the INCODE 45 Color 1 INCODE 46 INCODE 45 INCODE 4</pre>	27	7															
<pre>29 30 ! Initialize 31 32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 45 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 43 44 ! check the IDCODE 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</pre>	28	3															
<pre>30 : Initialize 31 32 : Row_Width : 3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 ! Check the IDCODE 45 Col:1 INFE</pre>	29	9															
32 ! Row_Width :3412 33 ! Address_Length :1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42	30		! Initialize	e													
1 Address_length :1 33 1 34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE 42 1 44 1 44 1 44 1 44 1 44 1 44 1 44 1	32	2	Row Width	• 3412													
34 HDR 0; 35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 43 44 1 Check the IDCODE 1 Check the IDCODE		3	Address L	ength :1													
35 HIR 0; 36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 III	34	i i	HDR 0;	-													
36 TDR 0; 37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 44 44 1 Check the IDCODE 11 INE	35	5 F	HIR 0;														
37 TIR 0; 38 ENDDR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 43 44 1 check the IDCODE 1 III	36	5 3	PDR 0;														
36 ENDIR DRPAUSE; 39 ENDIR IRPAUSE; 40 FREQUENCY 1.00e+006 H2; 41 STATE IDLE; 42 43 44 ! Check the IDCODE 1 III			FIR 0;														
40 FREQUENCY 1.00e+006 HZ; 41 STATE IDLE; 42 44 44 I Check the IDCODE 1 IDCODE			SNUDR DRPA	AUSE; AUSE:													
41 STATE IDLE; 42 43 44 1 Check the IDCODE (III	40		FREQUENCY	1.00e+0	06 H7:												
42 43 44 L Check the IDCODE (III)	41	L	STATE IDL	E;	,												
43 44 I check the IDCODE	42	2		-													
41 I check the IDCODE III III III INC	43	3															
	44	•	Check the	TDCODF													Þ
	·		Cali	-		NC					001001001						P

デバッガの操作は、組み込みソフトウェアをデバッグする際のプリミティブな手法と類似しています。上 部にあるアイコン列を活用してステップ実行、ブレークポイントの設定・解除、ブレークポイントまでの実 行、などを行います。アイコンとオペレーションは次の通りです。

アイコン	オペレーション	アイコン	オペレーション
•	Toggle Breakpoint	1	Next Breakpoint
5	Previous Breakpoint	-	Clear All Breakpoint
•	Step	GO	GO
	Reset		

なお、チェイン内に複数のデバイスがある場合、オプション設定が必要です。 C アイコンをクリックするか、ダウンロード・デバッガのメニューで Debugger -->Options... と辿り、SVF タブを設定することでオプション設定画面を表示させます。

図 21-52. SVF オプション設定

Ø	Options				2	X
	General SVF STAPI	-				
	🔲 Disable SVF syntax cheo	cker				
	🔲 Ignore IR and DR header	∵⁄tra	iler ir	n SVF		
	🔲 Continue on error					
	Mixed chain			_		
	Instruction register header:	0	-	FCK frequency:	1	-
	Instruction register trailer:	0	-	SVF vendor:	JTAG STANDARE	•
	Data register header:	0	-	Starting TAP state:	TLR	-
	Data register trailer:	0	-	_		
				•		
					UK Can	cel

JTAG チェイン内にあるターゲット・デバイスをデバッグする場合、前(header:ヘッダ)と後ろ(trailer: トレイラ)にあるデバイスの命令レジスタ(Instruction Register)長[単位:bit]を指定します(図 21-52、赤枠)。ない場合は0です。図 21-53 にヘッダ長・トレイラ長の設定例を示します。

図 21-53. JTAG チェインの構成とレジスタ長の例



21.5 その他の機能

21.5.1 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。念のため Configuration --> Cable and I/O Port Setup... と選択後、『Cable Detect』ボタンをクリックして自動検出して、意図する設定済みの状態と相違ないことを確認します。

図 21-54. ケーブル設定の再確認

	🔊 Cable and I/O P	ort Setup	? X
	Cable Settings		
🔊 Download Debugger	Cable:	HW-USBN-2B (FTDI)	•
File Edit Command Configuration Window Help	Port:	FT USB-0	•
💁 🗸 📷 👝 🛛 🚔 🎤 Cable and I/O Port Setup	Custom port(HEX)		Detect Cable
🗊 Options	🔲 Using slave SP	1 interface connection	
	I/O Settings		
	Ose default I/	O settings	
Cable and I/O Port Setup	?	X	
	/	PROGRAM pin	connected
Cable Settings			
Cable description: USB2 - FTUSB-0 (Lattice FTUSB Inter	face C⊱ ▼	Set ispEN hig	h
Cable: HW-USBN-2B (FTDI)	•	Set ispEN low	v
Port: FT USB-0	•		
Custom port(HEX):	Detect Ca	ble	OK Concel
Using slave SPI interface connection			Cancer

21.5.2 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 21-55 の左部のようなセクションがあ ります。『Debug Mode ...』ボタン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。

図 21-55. 信号接続テスト GUI

	Cable Settings Detect Cable Cable: HW-USBN-2B (FT • Port: FTUSB-0 • Custom port:	
	I/O Settings	😔 Cable Signal Tests
	Use default I/O settings	
Cable and I/O Settings	 Use custom I/O settings INITN pin connected DONE pin connected TRST pin connected Set TRST high Set TRST low 	Test Loop Test P
	ispEN pin connected	TCK Vumber of bytes:
	 ● Set ispEN high ○ Set ispEN low 	OK Cancel
	Debug Mode	

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。

21.5.3 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、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。本項はその方法を記述します。

プログラマーをデタッチした後、図 21-56 のように Edit --> Settings... と辿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、大きくするほど等価的に周波数が遅くなります。

図 21-56. TCK 周波数の変更

Programmer - impl1.xcf *	Settings
File Edit View Design Window Help Pile Copy Ctrl+C Paste Ctrl+V Paste HE I Image: Composition of the past of th	Programming Mode Turbo mode Sequential mode JTAG State Settings Use default JTAG states (TLR/TLR) Avoid test logic reset (TLR) state Use custom JTAG states Initial TAP state: TLR Final TAP state: TLR Pulse Width Delay Settings Use default pulse width delay Use default pulse width delay Use custom pulse width delay Use custom pulse width delay Check cable setup before programming Continue download on error OK Cancel

目安の周波数は以下の通りです。

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.

21.6 (参考) プログラミング時間

参考として各ファミリのオペレーション所用時間を示します。コンフィグレーション SRAM への書き込み 時間はファミリに拘わらず数秒程度で終了します。MachXO シリーズの内蔵不揮発メモリへのオペレーショ ン時間を以下に転記します。詳細はそれぞれのテクニカルノートをご参照ください。 図 21-57. MachXO3L のプログラミング時間(TN1294、Table 98)

NVCM/Flash Performance

Table 98. NVCM/Flash Performance in MachXO3L/LF Device¹

		MachXO3L/LF -640	MachXO3L/LF -1300	MachXO3L/LF -1300 256 Ball Package	MachX03L/LF -2100	MachXO3L/LF -2100 324 Ball Package	MachXO3L/LF -4300	MachXO3L/LF -4300 400 Ball Package	MachXO3L/LF -6900
CFG Erase	Min.	800	800	1100	1100	1800	1800	2800	2800
(tEraseCFG)	Max.	1400	1400	1900	1900	3100	3100	4800	4800
CFG Program	All	500	500	740	740	1400	1400	2200	2200
(tProgramCFG)	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
NVCM1/UFM Erase	Min.	400	400	500	500	600	600	900	900
(tEraseNVCM1/UFM)	Max.	700	700	900	900	1000	1000	1600	1600
NVCM1/UFM Program	Ali	110	110	140	140	180	180	480	480
(tProgramNVCM1/UFM)	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2

1. All times are averages, in (ms). SRAM erase times are < 0.1 ms.

注; "CFG" はコンフィグレーション用不揮発メモリを指す

図 21-58. MachXO2 のプログラミング時間(TN1264、Table 97)

Flash Memory Erase and Program Performance

Table 97. Flash Memory (UFM/Configuration) Performance in MachXO2 Devices¹

		MachXO2 -256	MachXO2 -640	MachXO2 -640U	MachXO2 -1200	MachXO2 -1200U	MachXO2 -2000	MachXO2 -2000U	MachXO2 -4000	MachXO2 -7000
CFG Erase	Min.	400	600	800	800	1100	1100	1800	1800	2800
(tEraseCFG)	Max.	700	1100	1400	1400	1900	1900	3100	3100	4800
CFG Program	All	130	270	500	500	740	740	1400	1400	2200
(tProgramCFG)	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
UFM Erase	Min.	_	300	400	400	500	500	600	600	900
(tEraseUFM)	Max.	_	600	700	700	900	900	1000	1000	1600
UFM Program	All	_	40	110	110	140	140	180	180	480
(tProgramUFM)	1 page	_	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
UFM Program (tProgramUFM)	All 1 page	_ _ _	40	110 0.2	110 0.2	140 0.2	140 0.2	180	180	480

All times are averages, in (ms). SRAM erase times are < 0.1ms.

なお、通常は上に示す様な数値に近くなるはずですが、使用する PC によっては、USB ドライバとクロック生成の関連で上に示す値より大幅に大きくなるケースも報告されていますので、ご了解ください。

外付け SPI フラッシュメモリへのプログラミング時間は種々のパラメータが関与するので一義的に示すことはできませんが、上記値より極端に大きくなることはないようです。

21.7 USB ドライバのインストール

USB ドライバのインストールが事前に完了していない場合、ターゲットボードを初めて USB ポートに接続 した場合に、インストールを促す Windows のメッセージが表示されます。 図 21-59. USB ドライバのインストール要求初期画面



『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 21-60. USB ドライバのフォルダ指定

から検索できます。検索された最適のドラ
(<u>M</u>)
▼ 参照(B)
されたドライバは、ハードウェアに最適のもの

『次の場所を含める』を選択して、"参照"ボタンで "C:\WINDOWS\system32" をブラウズして指定します。" 次へ"をクリックします。

FTDI USB スレーブデバイスがボード上にある場合、続いて図 21-61 に示すような画面が表示されます。

図 21-61. FTDI USB ドライバのコピー元指定

ファイルが必要 🛛 🔀 🔀 🗶 🗶 🗶 🗶 🗶 🗶 🗶 🗶 🗶 🗶 🗶 🗶				
	FTDI USB Drivers Disk 上のファイル 'ftser2k.sys' が必要で す。	ОК		
	ファイルの格納場所へのパスを入力して、[OK] をクリックしてくださ い。	キャンセル		
	コピー元(©): C.¥WINDOWS¥system32¥drivers			

"参照"ボタンで "C:\WINDOWS\system32\drivers" をブラウズして指定し、"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。

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 21-62. デバイスマネージャ



図 21-63. デバイスマネージャの USB ドライバ表示例(左: FTDI あり、右: なし)



21.8 改訂履歴

Ver.	Date	page	内容
3.3 v1.0	June 2015		初版(Diamond JUG Ver.3.3.1)

--- *** ----