

2021 年 12 月

Lattice Diamond 日本語ユーザーガイド

第15章 プログラミング・ユーティリティ

15.1 デプロイメント・ツール (Deployment Tool)

デプロイメント・ツールはファイル変換ユーティリティです。デュアルブート / マルチブート機能対応で 外付け SPI フラッシュメモリーに書く mcs ファイル ("PROM ファイル")を生成する場合や、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場合、或いは ECP5 ファ ミリーや Crosslink ファミリーが対応する Dual-/Quad-SPI フラッシュメモリー・インターフェイスを用いた り、またチェインを構成する場合など、種々のケースで用います。

15.1.1 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] \rightarrow [Utilities] \rightarrow [Deployment Tool] と選択します (図 15-1、左)。または Windows の Start メニューから Lattice Diamond 3.12 を展開すると 表示される [Deployment Tool] を選択しても起動できます (図 15-1、右)。

図 15-1. デプロイメント・ツールの起動



図 15-2. 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- project0.ddt	- 🗆 X
File Edit Help	
	IEEE 1532 ISO Data File Application Specific BSDL File EPDF File Bistream
Diamond Deployment Tool - Getting Started	SVF - JTAG Device SVF - JTAG Device STAPL - SInce Device STAPL - JTAG Drain
Greate New Deployment Function Type: File Conversion	ATE
Output File Type: THEE 1532 ISC Data File 🗸	Slave SPI Embedded 190 Embedded sysCONPIG Embedded
Open an Existing Deployment	Hav Drowy city
Recent Files: 0./usr.ps/D3.0works/chikkluds71/XLKduelboot.ddt 🗸	Dual Boot Advanced SFI Flash sycCONRG Daisy Chain
OK Okose	
Previo	us Next

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にお願い致します。

起動後の初期画面は図 15-2 のようになります。「Function Type」として四つが選択でき、それぞれに対す る生成ファイル形式があります。殆どの作業でステップ1(対象ファイル入力)、ステップ3(出力ファイル 名とフォルダーの指定)、およびステップ4(ファイル生成・実行)は共通の操作になります。ステップの進 行はそれぞれウィンドウ下部の『Next』ボタンで行います。戻る場合は『Previous』ボタンをクリックしま す。開始画面には戻れませんが、ステップ1までは戻ることができます。

15.1.2 jedec から hex への変換

しばしば jedec ファイルを 16 進表記にして確認・作業する必要がある場合があります。デプロイメント・ ツールではこの変換が容易にできます。起動後、最初のウィンドウで、「Function Type」として "File Conversion" を、「Output File Type」として "JEDEC to Hex" を選択して『OK』をクリックします

図 15-3. jedec から hex 変換の開始



次にステップ1で対象のファイルをブラウズして指定します(図15-4)。

図 15-4. 対象ファイルの指定

File Edit Help	Diamond Deployment Tool- project1.ddt*			_	×
Pile Conversion: JEDEC to Hex Step 1 of 4: Select Input File(s) File Name (*.jed) Device Family Device 1:ryption/forXO3D/impl1/d3Cxo3DchkBek.impl1_a.jed[ile Edit Help				
File Conversion: JEDEC to Hex Step 1 of 4: Select Input File(s) File Name (*.jed) Device Family Device 1 :ryption/forXO3D/impl1/d3Cxo3DchkBek.impl1_a.jed[MachXO3D LAMXO3D-4300HC	^ 🗠 🔚 📧 📼 📼 🚟 🌌				
File Name (*.jed) Device Family Device 1 :ryption/forXO3D/impl1/d3Cxo3DchkBek_impl1_a.jed[MachXO3D LAMXO3D-4300HC	ile Conversion: JEDEC to Hex Step 1 of 4: Select Input File(s)				
1 ryption/forXO3D/impl1/d3Cxo3DchkBek_impl1_a.jedl MachXO3D LAMXO3D LAMXO3D-4300HC	File Name (*.jed)	Device Family	Device		
	1 :ryption/forXO3D/impl1/d3Cxo3DchkBek_impl1_a.jed	MachXO3D	LAMXO3D-4300HC		

ステップ2は出力フォーマットの指定です(図15-5)。通常は"ASCII Row Hex"です。

図 15-5. 出力フォーマットの確認

Diamond Deployment Tool- project1.ddt*	-	\times
File Edit Help		
😬 🖻 🔲 🔤 📼 🖼 🔤 🔛		
File Conversion: JEDEC to Hex		
Step 2 of 4: JEDEC to Hex Options ASOII Raw Hex (*hex) Binary Raw Hex (*bin)		
Output Format: ASCII Raw Hex (*hex)		
Generate digest file		

ステップ3は出力ファイル名とフォルダーの指定です(図15-6)。

図15-6. 出力ファイル名とフォルダーの指定

Diamond Deploy	ment Tool- project1.ddt*	-		\times
File Edit Help				
🐣 🖻 🔒	201 🐹 🔤 🔛 📷			
File Conversion:	JEDEC to Hex			
Step 3 of 4:S	elect Output File(s)			
Output File1:	C:/usr_ss/D3Cworks/chkEncryption/forXO3D/imp11/d3Cxo3DchkBek_imp11_ahex		~ .	-



ステップ4がファイル生成・実行です(図 15-7)。上部のアイコン 🌌 をクリックするか、右下の『Generate』 ボタンをクリックします。ステータス窓で"... successfully."となっていることを確認して終了します。

図 15-7. 変換実行と結果確認

Diamond Deployment Tool- project1.ddt*	-		×
File Edit Help			
😬 🔛 🚥 🚥 🚥 🚥 🎬			
File Conversion: JEDEC to Hex			
Step 4 of 4: Generate Deployment			
		View File	
Deployment Tool Summary			^
Input File: C:/usr_ss/D3Cworks/chkEncryption/forX03D/impll/d3Cxo3DchkBek_impll_a.jed			
Options:			
Output Format: ASCII Raw Hex (*.hex)			
Output File: C:/usr_ss/D3Cworks/chkEncryption/forX03D/impl1/d3Cxo3DchkBek_impl1_a.hex			\checkmark
Command Line "C:/IscCidiamond/3.12/bin/nt64/ddtcmd" -oft jed2hex -dev LAMX03D-4800HC -jf "C:/usr_ss/D3Cworks/chkEncryption/forX03D/imp11/d3Cxo3DchkBek_imp11_ajed" -of "C:/usr_ss/D3Cworks/chkEncryption/forX03D/imp11/d3Cxo3DchkBek_imp11_ahex"			
Deployment Generation Status Lattice Diamond Deployment Tool 3.12 Command Line			
Loading Programmer Device Database			
Generating RAW Hex File			
Device Name: LAMXO3D-4300HC Reading Input File: C:/usr_ss/D3Cworks/chkEncryption/forXO3D/imp11/d3Cxo3DchkBek_imp11_a.jed			
Output File: C:/usr_ss/D3Cworks/chkEncryption/forXO3D/impl1/d3Cxo3DchkBek_impl1_ahex Generating Raw Hex File			
User Code: 0x00000000 File C.¥usr ss¥D3Cworks¥chkEncryption¥forXO3D¥imp11¥d3Cxo3DchkBek_imp11_ahex generated successfully.			
Lattice Diamond Deployment Tool has exited successfully.			
Pr	evious	Gener	ate

15.1.3 デュアルブート用 mcs ファイル生成: ECP5、Crosslink

ECP5 ファミリーと Crosslink が対応するデュアルブート機能用に、外付け SPI フラッシュメモリーにスト アする二本のビットストリーム・ファイルを結合して *.mcs ファイルを生成します。デュアルブート機能は、 CRC エラーなど何らかの理由で"プライマリ"ビットストリームでの起動に失敗すると、自律的に"ゴール デン"ビットストリームで再起動を試みるものです。

図 15-8. デュアルブート用ファイル生成の開始

🛐 Diamond De	ployment Tool - Getting Started	?	×
) Create New	Deploymen t		
Function Type:	External Memory	•	
Output File Type;	Dual Boot	•	
🔿 Open an Ex	isting Deployment		
Recent Files:	C:/usr_ss/D3Cworks/E5gdmi4pkg/E5multi5boot.ddt	~	
	ОК	Clo	se

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" を選択して『OK』をクリックします (図 15-8)。



図 15-9. ビットストリーム・ファイル2本の指定例

MACNICA

🖸 Diamond Deployment Tool- project0.ddt*	-		×
File Edit Help			
* 🖻 🖶 📧 🚥 🕮 🚟 🌌			
External Memory: Dual Boot			
Step 1 of 4: Select Input File(s)			
File Name (*.jed *.bit *.rbt) Device Family Device			
1 C:/usr_ss/D3Cworks/chkXlkLvds71/imp11/chkXlkLVDS71rx_imp11.bit LIFMD LIF-MD6000			
2 Cr/usr_ss/D3Cworks/chkXlkLvds71/impQ/chkXlkLVDS71rr_impl2.bit LIFMD LIF-MD6000			
Peri	avicus	N	evt

次に使用するファイルの指定ウィンドウが表れます(図 15-9)。生成済みの .bit 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。

次のウィンドウは生成するファイルのオプション設定です(図 15-10)。「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 モリーのサイズを選択し、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 フラッシュメモリー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 15-10 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、『Next』ボタンを押して次に進みます。「SPI Flash Read Mode」はデフォルトが "Standard Read" です。第 15.1.5 項に記述 する、リードデータ線を複数とする "Dual/Quad SPI リード"の場合は "Dual (Quad) I/O SPI Flash Read" を選 択します。

図 15-10. 生成ファイルのオプション指定例

Diamond Deployme	nt Tool- project0.ddt*	-		×
File Edit Help				
🐣 🖻 🔲 🔤	📼 📧 🔤 🔛 🔀 🚺			
External Memory: Du	al Boot Fast Read			
Step 2 of 4: Dual	Boot Options Quad I/O SPI Flash Read			
Output Format	Intel Hey			
SPI Flash Size (MD):	04			
SPI Flash Read Mode	Standard Read 👻			
Golden Pattern:	C:/usr_ss/D3Cworks/chkXlkLvds71/imp11/chkXlkLVDS71rx_imp11t 🔻 Starting Address: 0:	<00400000	•	
Primary Pattern:	C:/usr_ss/D3Cworks/chkXlkLvds71/imp11/chkXlkLVDS71rx_imp11E +			
🗹 Protect Golden Sec	tor			
🗹 Byte Wide Bit Mirro	r			
🗌 Retain Bitstream H	eader			
Optimize Memory S	pace			
		Dention		_
		Previous	Nex	τ

生成ファイル(拡張子.mcs)とフォルダー位置の確認・指定ウィンドウが表示されますので入力し(図 15-11)、右下の『Next』ボタンをクリックします(図 15-11 では割愛)。

図 15-11. ファイル名とフォルダー指定例

MACNICA

) ×	-	🛐 Diamond Deployment Tool- project0.ddt*
		File Edit Help
		* 🖻 🗔 📧 📼 📼 🔤 🌌
		External Memory: Dual Boot
		Step 3 of 4: Select Output File(s)
	\sim	Output File 1: C:/usr_ss/D3Cworks/chkXlkLvds7(2hkXlkLVDS71rx_dualBoot.mc
		External Memory: Dual Boot Step 3 of 4: Select Output File(s) Output File1: C:/usr_ss/D3Cworks/chkXlkLvds? ChkXlkLVDS?1rx_dualBoot.mc

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"... successfully"とログ表示されれば正常終了です(図 15-12)。

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。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。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) 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・ファミリー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、ユーザーが何らかの別の手段 で生成する場合は注意が必要です。

図 15-12. ファイル生成とチェック

Diamond Deployment Tool- project0.ddt*	- 🗆	×
le Edit Help		
* 🖻 🕞 🔤 🔤 🖭 🔤 🌌		
xternal Memory: Dual Boot		
Step 4 of 4: Generate Deployment		
	View File	
Deployment Tool Summary		~
Terret File 1. C. free of (EQCorple (=) Wildersda 21 (i==11 (=) Wild WE 21 (
Input File 7: C:/Usr_ss/D3Cworks/chkXikLvds/1/inpi1/chkXikLvD5/ir: Input File 7: C:/Usr_ss/D3Cworks/chkXikLvds71/inpi2/chkXikLvD571r:	x_impli.bit	
		¥
<	>	
"G/asc/aamond/3.12/bin/nt64/ddtcmd" -ofi -boot -dev LIF-MD6000 -solden "G/asc/ss/D3Gworks/chk3kLvds71/imp1/chk3kLvD571rx;imp11bit" -soldenadd tx0040 "G/asr_ss/D3Gworks/chk3kLvds71/imp11/chk3kLvD571rx;imp11bit" -format int -f1shs -mirror -of "G/asr_ss/D3Gworks/chk3kLvds71/krk3kLvD571rx;imp11bit" -format int -f1shs	0000 -primary ize 64 -protect	^
		¥
Start generation. Generating Intel Hex PRDM File Souccessfully generate file(s) C#usr_ss¥D3Gworks¥chkXlkLvds71¥chkXlkLVD571rx_dualBoot.mcs		^
Lattice Diamond Deployment Tool has exited successfully.		
• · · · · · · · · · · · · · · · · · · ·		
		¥

デプロイメント・ツールの終了は、ウィンドウ右上隅の"X"印をクリックしますが、プロジェクトファ イル(拡張子.ddt)を保存するかどうかの確認を促されます。保存しておけば、後で修正などの繰り返し作 業する場合などに有用です。

15.1.4 マルチブート用 mcs ファイル生成: ECP5

ECP5 ファミリーは、最大四本までの代替えビットストリーム・ファイルを外付け SPI フラッシュメモリー に格納し、それらを自律的に切り替えるマルチブート機能があります。そのための mcs ファイルを生成しま す。Dual Boot のように出力ファイルタイプとしては定義されていませんので、図 15-13 のように最初のウィ ンドウでは「Function Type」を "External Memory"、「Output File Type」を "Advanced SPI Flash" にします。

図15-13. マルチブート用ファイル作成の開始

Oreate Net	ø Deployment
Function Type:	External Memory
Output File Type	Advance d SPI Flash
Open on E	ásting Deployment
Recent Files:	C:/usr_ss/D3Bworks/e5ohkBek/e5enorypt.ddt 🗸 🔍

図 15-14. マルチブート用ファイル作成: ステップ1

	iamon	d Deployment Tool- project0.ddt*	-		×
File	Edit	Help			
2	Ľ	🔚 📧 📼 🕮 🐺 🔤			
Exte	rnal M tep 1	emory: Advanced SPI Flash of 4: Select Input File(s)			
		File Name (*.bit *.rbt *.bin *.hex) Device Family		Device	e
13	32b/d3	Ce5gdmiVip4pkg_impl6_extclk135u150m32b.bit ECP5UM	LFE5UN	1-85F	

次のステップ1 (図 15-14) では"ゴールデン"ビットストリーム・ファイルを指定します。ステップ2 で変更も可能です。指定後、ウィンドウ下部の『Next』ボタンで次に進みます。

図 15-15. マルチブート用ファイル作成:ステップ2

Diamond Deployment Tool-	project0.ddt*				-		×
OA no II ma							
		j 🍊 📟					
External Memory: Advanced Step 2 of 4: Advanced Sf Options User Data Files (Multiple Boot Golden Pattern: C/usr_ss/	PI Flash PI Flash Options Multiple Boot	t/fit2verifyVersa/impi	1_extref158u200m/d3Ce	5∉dmiVersa4pkg_in	pl1_extref15t ◄	• -	
Starting Address:	0×00400	0000 •	•				
Protect Golden Sector Number of Alternate Pattern	ns:	2	\supset				
Alternate Pattern 1:	C:/usr_ss/D3Cworks/	E5gdmi4pkg/fit2verify	Versa/impl2_extref125u	150m/d3Ce5gdmiV	ersa4pkg_ir 💌	-	
Starting Address:	0×00140000		•				
Next Alternate Pattern to	Configure:	Primary Pattern		•			
Alternate Pattern 2	C:/usr_ss/D3Cworks/	E5gdmi4pkg/fit2verify	/Versa/impl3_150u200m/	/d3Ce5ødmiVersa4p	okg_imp13_1 🔻	-	
Starting Address:	0×00290000		•				
Next Alternate Pattern to	Configure:	Primary Pattern		•			
Alternate Pattern &					7	-	
Starting Address:			~				
Next Alternate Pattern to	Configure:	Primary Pattern		Ψ.			
Alternate Pattern 4:					Ψ.	-	
Starting Address:			w.				
Next Alternate Pattern to	Configure:	Primary Pattern		÷			
					Previous	Nex	t

ステップ2ではマルチブート・モードの指定、ビットストリーム・ファイルの指定を行います(図 15-15)。 タブ [Multiple Boot] を選択して最上部の「Multiple Boot」チェックボックスをチェックしてイネーブルしま す。「Number of Alternate Patterns」のプルダウン (2~4) からゴールデン以外のパターン数を指定し、ぞれ ぞれのセルに対してファイルをブラウズして指定します。ゴールデンにはステップ1で与えたファイルが 入っていますので、このステップで変更することも可能です。

指定パターン数に対して SPI フラッシュメモリーのサイズが不足する場合は、ウォーニング・メッセージ が表示されますので、[Options] タブを選択して「SPI Flash Size」部を適宜変更します。「Starting Address」は 特段の理由がない限り変更しないことを推奨します。

この後のステップはデュアルブートと同様です(図15-11、図15-12)。

15.1.5 Dual/Quad SPI リード用 mcs ファイル生成: ECP5

ECP5 ファミリーは、MASTER SPI インターフェイスで外付け SPI フラッシュメモリーからコンフィグレー ションする際に、データ線を二本(Dual)または四本(Quad)にすることができます(Dual/Quad SPI リー ド)。この場合にも SPI フラッシュメモリーに格納するビットストリーム・ファイルは単独ですが、やはりデ プロイメント・ツールで mcs に変換します。

作業開始のウィンドウではマルチブートと同様で、「Function Type」を "External Memory"、「Output File Type」を "Advanced SPI Flash" にします (図 15-16)。次のステップ1でのファイル指定も同じです。

異なるのはステップ2で、のように「SPI Flash Read Mode」を "Dual (Quad) I/O SPI Flash Read" にします。

図 15-16. Dual/Quad SPI リード用ファイル作成: ステップ2



ステップ3と4はデュアルブートと同様です。

15.1.6 MSPI/SSPI デイジーチェイン対応 SPI 用 mcs ファイルの生成: ECP5

ECP5 には種々コンフィグレーション・モードがありますが、図 15-17 のように MSPI モードの ECP5 を先 頭にして SSPI モードの ECP5 を複数個、同一チェインにする構成(デイジーチェイン)があり得ます(各信 号線にはプルアップ / ダウン抵抗が必要ですが割愛しています。DONE/INITn 信号線は Wired-OR 接続して います)。この場合に SPI フラッシュメモリーには各デバイスのビットストリーム・ファイルを結合した mcs ファイルを格納し、これはデプロイメント・ツールで作成します。

なお、本構成での各ビットストリーム・ファイルの作成は、Diamond において次のような設定が必要です。

- ・SPI フラッシュメモリを接続する先頭の ECP5 はスプレッドシート・ビュー [Global Preferenecs] タ ブの sysCONFIG セクションで、MASPTER_SPI_PORT=ENABLE、DONE_EX=ON にします
- ・二個目以降の ECP5 はスプレッドシート・ビュー [Global Preferences] タブの sysCONFIG セクショ ンで、SLAVE_SPI_PORT=ENABLE、DONE_EX=ON にします
- ・アクティブなストラテジー設定を立ち上げ、"Bitstream" プロセスの "Chain Mode" オプションを全



てのデバイスで [Bypass] にします

以上の設定でそれぞれ "Export Files" プロセスまで実行してビットストリーム・ファイルを生成します。その後、これらファイルに対して第15.4 節に記述する操作で MCLKB ビットを編集します。これら手順で準備した対象ファイルを用いて次のステップ以降を行います。

デプロイメント・ツールでの開始はマルチブートと同じように(図 15-13)、「Function Type」を "External Memory"、「Output File Type」を "Advanced SPI Flash" にします。ステップ1のファイル指定も図 15-14 と同様ですが、これには先頭の MSPI モードの ECP5 用ビットストリーム・ファイルを与えます。

図 15-17. ECP5 の MSPI/SSPI デイジーチェイン構成例



ステップ2が各ファイルの指定など詳細設定です(図 15-18)。[User Data Files] タブで「User Data Files」 ボックスと「Byte Wide Bit Mirroring」ボックスを共にチェックします。「Number of User Data Files」 プルダ ウンを操作し、SSPI モードのデバイス数に一致する値にします。値に相当する各フィールドがアクティブに なりますので、それぞれブラウズ・ボタンをクリックして適切なビットストリーム・ファイルを指定します。

図 15-18. MSPI/SSPI チェイン用 mcs 生成のステップ2の例

	D	iamon	d Deploy	ment	Tool- j	projec	t1.ddt	*							
	File	Edit	Help												
	2	Ċ		ISC	D SEL	JED	1.07 	SVF	SV F JTAG	2	LOG				
	Exte	rnal M	lemory:	Adv	anced	SPI	Flast	n							
l	S	tep 2	of 4: A	dvanc	ed Sl	PI Fla	ash O	ption	s						
l	0	ptions	Use	r Data	Files	∕м	ultiple	Boot							
l		🔽 Use	er Data I	Files											
	ſ	🗹 By	te Wide	Bit Mir	roring										
	Ľ	Numbe	r of Use	er Data	File	4	+	D							
						-	_	_						 ~	
		User	Data Fi	ile 1:	C:/u	isr_ss	4 D 3Cw	/orks	E5gdr	ni4pkg/1	iit2verif	yVip∕ir	•		
		Start	ting Add	ress:	8×01	01E00	00		•				$\overline{}$		
		User	Data Fi	ile 2:	C:/u	isr_ss.	/D3Cw	/orks	E5gdr	ni4pk∉∕1	it2verif	yVip∕ir	• 👻		
		Start	ting Add	ress:	0×0	03C00	00		•						
		User	Data Fi	ile 3:	C:/u	isr_ss.	/D3Cw	/orks/	E5gdr	ni4pkg/1	it2verif	yVip∕ir	•		
		Star	ting Add	ress	0×0	05A00	00		•						
		User	Data F	ile 4:	G:/u	isr_ss.	/D3Cw	orks/	E5gdr	ni4pkg/1	it2verif	yVip∕ir			
		Start	ting Add	ress:	0×01	07800	80		•		_				
		User	Data Fi	ile 5:									v		

「Starting Address」各セルは自動的に値が入りますので、ユーザーが編集する必要はありません。

[Options] タブ (図 15-18 では図示していません)の「SPI Flash Size」部で SPI フラッシュメモリーのサイ

ズを指定しますが、全ファイルのサイズに対して SPI フラッシュメモリーが小さい場合は、ウォーニング・ メッセージが表示されますので、適宜変更します。

また、[Options] タブ内にも「Byte Wide Bit Mirror」オプションがありますが、こちらはチェックしないで おきます。次のステップ3の出力ファイル名の設定とステップ4 "Generate" は他と同様です。

15.1.7 ビットストリーム・ファイルの暗号化

ビットストリーム・ファイル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(第 14.6.1 項に関連記述がありますので、ご参照ください)。 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 15-19)。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 15-19. 暗号化ステップの開始

🛃 Dia	amon	d Deploym	ent Tool-	projec	t0.dd	t					_		\times
File	Edit	Help											
2	Ŕ		5C 920	JED	-	SVF	SV F JTAG		LOG				
	6	Diamond	Deployn	nent To	ol - G	etting	Start	ed			?	×	
) Create M	le# Depl	oymen	t								
	F	unction Type	File	e Conve	rsion						-		
	0	utput File Ty	pe: Bit	stream							-		
	0) Open an	Existing	Deplo	ymen	t							

次のステップは暗号化対象ファイルの指定ですが、これは他と同様の方法でブラウズして『Next』ボタン で進みます。次のステップ2はオプションの詳細設定です(図 15-20)。

図 15-20. 暗号化のステップ2

Diamond Deployment	Tool- project0.ddt*			\times	
File Edit Help					
🐣 📔 🔚	9 SCL JED MALE SVF	🜃 📝 LOG			
File Conversion: Bitstr	ream				
Step 2 of 4: Bitstrea	am Options				
Ouput Format:	Binary Bitstream	•			
Program Security Bit:	On	~			
Verify ID Code:	Default	•			
Frequency:	Default	~			
Compression:	Default	•			
CRC Calculation:	Default	Encryption File Password		?	×
USERCODE Format:	Hex	Password (Maximum 16 characters- Cas	e sensitive)		
USERCODE:		•••••	L	0	ί.
Byte Wide Bit Mirror		Hide password	[Can	cel
Retain Bitstream Head	der				
Encryption					
Configuration Mode:			~		
Encryption Key:					
		Load Key			

最下部の「Encryption」ボックスをチェックして有効にして暗号化キーを入力します。『Load Key』ボタン をクリックして表示されるファイル・ブラウザのウィンドウで暗号化キーファイル (.bek) を指定すると、 図 15-20 内の小ウィンドウが表示されますので、暗号化キーファイルのパスワードを入力します。パスワー ドが一致すれば、『OK』をクリックして抜けた後に「Encryption Key」セルが"●"で満たされます。『Next』 ボタンで次に進みます。

パスワードが一致しないとこのステップを完了できませんので、ご注意下さい。また「Encryption Key」セルを直接編集することやパスワードの変更などはできません。パスワードの設定と暗号化キーファイルの生成は Diamond の "Security Tool" で行います。第14.6.3 項をご参照下さい。

ステップ3は出力ファイル名と場所の指定です(図 15-21)。暗号化したことが容易に判別できる名称にするようにします。『Next』で次に進みます。

図 15-21. 暗号化ステップ3

D 🕄	iamon	d Deploymen	t Tool- pi	roject0.do	lt*						_		×
File	Edit	Help											
2	P	E Isc	1974	JED	SVF	SV F JTAG	2	LOG					
File	Conve	ersion: Bits	stream										
s	tep 3	of 4: Selec	t Output	t File(s)									
Out	put File	e 1: C:	/usr_ss/[)3Bworks	/e5chk	:B&{/	13Ce5ch	<bek_im< td=""><td>pl1_encry</td><td>ptedbit</td><td></td><td>~</td><td>></td></bek_im<>	pl1_encry	ptedbit		~	>

ステップの最後が暗号化ビットストリームの生成です。右下の『Generate』ボタンか上部の 🌌 アイコン をクリックします。他と同様に"... excited successfully"と表示されれば正常終了です。

15.1.8 SPI フラッシュメモリー内にユーザーデータを共存

外付け SPI フラッシュメモリー内に、コンフィグレーションデータとユーザー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ありませんが、以下の基本的な点を押さえておきます。

- ・使用するデバイス(ファミリーと規模)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
- ・デバイスによっては SPI アクセスの信号線はコンフィグレーション用ポートが共用できないため、注 意深くピン配置を行う

SPI フラッシュメモリーのユーザーデータにアクセスするための手段は、ユーザーが用意します。共存させるためのツールとしてデプロイメント・ツールが使用できます。MSPI/SSPI チェインの場合と同様の手順ですが、図 15-18 においてビットストリーム・ファイルではなく、意図するユーザーデータを指定します。また、開始アドレスの変更はユーザー依存です。

15.2 フィーチャー行エディター (MachXO2/XO3L/XO3LF)

注:フィーチャー行のみの意図的な変更は開発時(及び量産出荷時)に限ることを強く推奨しています。

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、フィーチャー行 (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。これは、特にフィーチャー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。

何らかのやむを得ない事由(含デバッグ等)でフィーチャー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ー行エディターが活用できます。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 15-22)、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 15-1 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 15-22. フィーチャー行エディターの起動



起動後、まず①ブラウズボタンで jed ファイルをブラウズ後に指定します。次に②『Read』ボタンをクリックすることでファイルからフィーチャー行情報を抽出すると、図 15-23 のように表示されます。

図 15-23. フィーチャー行エディター、jedec リード後の表示例 (MachXO3L)

	Reserved	Reserved	Boot Select 2	Boot Select 1	MSPI Persistent Enable	2C Persistence Disable	SSPI Persistence Disable	JTAG Persistence Disable	DONE Persistence Enable	INIT Persistence Enable	PROGRAM Persistence Disable	my_ASSP Enable	PWD Enable All	PWD Enable	DEC Only	Secure PWD																	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	
Chip Value	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0																	
Default Chip Value	 Dual Boot Golden Address (15) 	 Dual Boot Golden Address (14) 	 O Dual Boot Golden Address (13) 	 Dual Boot Golden Address (12) 	 Dual Boot Golden Address (11) 	 Dual Boot Golden Address (10) 	 Dual Boot Golden Address (9) 	 Dual Boot Golden Address (8) 	 Dual Boot Golden Address (7) 	 Dual Boot Golden Address (6) 	 Dual Boot Golden Address (5) 	 Dual Boot Golden Address (4) 	 Dual Boot Golden Address (3) 	 Dual Boot Golden Address (2) 	 O Dual Boot Golden Address (1) 	 O Dual Boot Golden Address (0) 	 I2C Slave Address (9) 	 I2C Slave Address (8) 	 I2C Slave Address (7) 	 I2C Slave Address (6) 	 I2C Slave Address (5) 	 I2C Slave Address (4) 	 I2C Slave Address (3) 	 I2C Slave Address (2) 	 O Tracel D (7) 	O O TracelD (6)	 TraceID (5) 	 TraceID (4) 	 TracelD (3) 	 TraceID (2) 	 O TracelD (1) 	O D TracelD (0)	
Default Chip Value	 Custom ID Code (31) 	 Custom ID Code (30) 	 O Custom ID Code (29) 	 O Custom ID Code (28) 	 O Custom ID Code (27) 	 Custom ID Code (26) 	 Custom ID Code (25) 	 O Custom ID Code (24) 	 O Custom ID Code (23) 	O Custom ID Code (22)	 O Custom ID Code (21) 	 Custom ID Code (20) 	 O Custom ID Code (19) 	 O Custom ID Code (18) 	O Custom ID Code (17)	O Custom ID Code (16)	 O Custom ID Code (15) 	 Custom ID Code (14) 	 Custom ID Code (13) 	O Custom ID Code (12)	O Custom ID Code (11)	O Custom ID Code (10)	 Custom ID Code (9) 	 Custom ID Code (8) 	 O Custom ID Code (7) 	 O Custom ID Code (6) 	O Custom ID Code (5)	 O Custom ID Code (4) 	 Custom ID Code (3) 	 Custom ID Code (2) 	 Custom ID Code (1) 	O Custom ID Code (0)	
unp value	0	0	0	0	U	0	0	0	0	0	U	0	0	0	0	0	U	0	0	0	0	0	0	0	0	0	0	•	0	0	0	0	
								6	D	_					(2)	_										- (3)	_				

MachXO3D の場合は <xxx_a.jed><xxx_b.jed> ファイルではなく、<xxx.fea> ファイルを対象に指定します。表示されるウィンドウは図 15-23 とはやや異なります。

黒色で表記されているビットのいず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、③『Save As』で Jedec ファイルとして別名で保存します。

なお、フィーチャー行エディターでは、第16.3.2 項で記述する TraceID および Custom ID Code の編集ができます。図 15-23 の中に含まれていることが理解できます。

意図せず不明瞭な理解に基づくアクセスを行うと、不適切な値に書き換えてしまい(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状態になる可能 性が否定できません。注意の上に注意が必要です。

15.3 USERCODE エディター

特にユーザーコードに限定して編集するツールが USERCODE エディターです。起動は図 15-24 のように [Tools] → [USERCODE Editor...]を選択します。

表示されるウィンドウの右上「Device」行右端のブラウズボタンで、意図するビットストリーム・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(図 15-24 右下)。『Read』ボタンをクリックして値を抽出して表示します(図 15-24 左下)。これで所望の値に編集し、『Save As』で別名保存します。デフォルトでは 16 進表記になっていますが、選択が可能です。

図 15-24. USERCODE エディターの起動から編集までの例

Z Programming File Utility	💐 USERCODE Editor ? 🗙
File Edit Command Tools Configuration Window Help Image: Security and Persistent Fields Editor Image: Security and Persistent Fields Editor Image: Security and Persistent Fields Editor	Device:
욿 USERCODE Editor ? 🗙	💐 USERCODE Editor 🛛 💡 🗙
Device: LFE5UM-45F USERCODE Format Hex USERCODE: 00000000 Read Save Save As Close	Device: LFE5UM-45F

15.4 コントロール・レジスタ0エディター

第 15.1.6 項で記述した ECP5 MSPI/SSPI デイジーチェイン・コンフィグレーション用の mcs ファイル生成 に関連して、チェインに接続される全ての ECP5 は、このコントロール・レジスタ0エディターを用いてビッ トストリーム・ファイルを編集する必要があります。

それ以外でユーザーが用いる必要があるケースは特に認識されていません。

図 15-25. コントロール・レジスタ0 エディターの起動

🏄 Р	rogran	nming F	ile Uti	ility			
File	Edit	Comm	nand	Tools	Configuration	Window	Help
1		ß	\$	%	Feature Row Editor	r	
3				CTril0	Control Register0 E	ditor	
				ا 📀	Control Register1 E	ditor	
				UES	USERCODE Editor		
				1	Security and Persis	tent Fields	Editor

起動は Tools → Control Register0 Editor... と選択します(図 15-25)。

起動後、まず①ブラウズボタンで bit ファイルをブラウズ後に指定します。次に②『Read』ボタンをクリッ クすることで図 15-26 のようにファイルから情報が抽出され表示されます。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、③『Save As』で bit ファイルを別名で保存します。

ECP5 MSPI/SSPI チェインの構成時は、"MCLKB" ビットを'1'に変更する必要があります。編集後の ビットファイルが、mcs ファイルとして結合する対象です。『Save』または『Save As』で保存して使用します。



図 15-26. コントロール・レジスタ 0 エディターの作業例 (ECP5)



15.5 ダウンロード・デバッガー

何らかの理由で 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、これを用いてデバッグを実行することによって原因を究明できる場合があります。ダウンロー ド・デバッガー (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 ファイルを実行するためのツールです。"svf (serial vector format)"は元来 PCB テスターで用いられる書 式です。

図 15-27. デプロイメントツールのファンクションタイプ選択

Diamond De	ployment Tool - Getting Started		? ×
● Create Ne w	Deployment		
Function Type:	Tester	•	·
Output File Type:	SVF - Single Device		•
🔿 Open an Exi	SVF - Sindle Device SVF - JTAG Chain STAPL - Single Device STAPL - JTAG Chain		
Recent Files:	ATE		
		ОК	Close

SVF ファイルを未生成の場合、まずデプロイメントツールでフォーマット変換します。「Function Type」は "Tester"、「Output File Type」は "SVF" を選択します (図 15-27)。JTAG チェイン内にターゲットの単一デ バイスのみでは "SVF - Single Device" を、ターゲットを含む複数デバイスの JTAG チェインの場合は "SVF -JTAG Chain"です。

図 15-28. SVF 変換の各種オプション指定

Diamond Deployment Tool- project0.ddt*	_	×
File Edit Help		
* *		
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 🗸		

ステップ1でソースファイルを指定後、ステップ2で図15-28のようにオペレーション (Operation) 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。

その後ステップ3で出力ファイル名とフォルダーを指定し、ステップ4で変換を実行(Generate)します (図15-29)。"... 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。

図 15-29. SVF ファイルの生成・変換例

Diamond Deployment Tool- project0.ddt*	-		\times
File Edit Help			
* 🖻 🗔 🔤 📼 🔤 🔤 🌌			
Tester: SVF - Single Device			
Step 4 of 4: Generate Deployment			
	Launcl	h Debugg	er
Deployment Tool Summary			^
Input File 1: C:/usr_ss/DssPkgs/E5gdmi/d3Ce5gdmiVersa4pkg/impll_ext Input File 2:	ref156	1200m/d	3(
Ontioner			~
<		3	>
C/usr_ss/DssPkgs/E5gdmi/d3Ce5gdmiVersa4pkg/impl1_extref156u200m/d3Ce5gdmiVersa -dev LFE5UM-45F -op "Fast Program" -of "C./usr_ss/DssPkgs/E5gdmi/d3Ce5gdmiVersa4pkg/impl1_extref156u200m/d3Ce5gdmiVersa	4pke_imp 4pke_imp	1_extref1 1_extref1	5
٢		2	>
Device 1 LFE5UM-45F:Fast Program			^
Build SVF File Operation: Successful.			
Lattice Diamond Deployment Tool has exited successfully			
			~
<		2	>
Pr	evious	Gene	rate

以上で準備が整いましたのでダウンロード・デバッガーを起動します。図 15-29 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。表示は生成した SVF をロードした状態で立ち上がり、ブレークポイントを設定したりできるようになります(図 15-31)。

または一度デプロイメント・ツールを終了し、図 15-30 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ーが立ち上がりますので、File → Open とたど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 15-31 のような初期状態になります。

図 15-30. プログラマーからダウンロード・デバッガーを起動

File Edit View	Desi	gn Window Help			
Enable Status D	38 (*)	JTAG Scan Check XCF Project Program			
	LOG	Log Clear Log File			
		Utilities	•		Deployment Tool
		BSCAN Configuration.		Ø	Download Debugger
			-	*	Model 300 Programmer Programming File Utility

デバッガーの操作は、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図中に示す通りです。

図 15-31. SVF デバッガー表示例

😰 Dow	nload Debugger - [C:/usr_ss/DssPkgs/E5gdmi/d3Ce5gdmiVersa4pkg/impl1_extref156u200m/d3Ce5gdmiVersa4p –	- 🗆 🗙
SWF File	Edit Command Configuration Window Help	- 8 ×
1 😷 👻	📴 🗔 🚍 🗅 🐰 ሴ 🕸 🖕 🌹 🏘 🕨 GO 🕋 🔯 🔤	
19 20 21 22 23 24 25 26	! Part: LFESUM-45F-8CABGA381 ! Date: Tue Mar 30 09:23:13 2021 ! Rows: 9470 ! Cols: 846 ! Bits: 8011620 ! Readback: Off ! Security: Off ! Security: Off Previous Breakpoint	^
27	Next Breakpoint	
28	Toggle Breakpoint	
30	! Initialize	
31 32	Row Width :848	
	! Address_Length :1	
34	HDR 0;	
35	HIR 0;	
36	TDR 0;	
38	ENDDR DRPAUSE:	
39	ENDIR IRPAUSE;	
40	FREQUENCY 1.00e+006 HZ;	
41	STATE IDLE;	
42		
43		
44	! Check the IDCODE	
46	! Shift in IDCODE PUB(0xE0) instruction	
47	SIR 8 TDI (EO);	
48	SDR 32 TDI (0000000)	
49	TDO (01112043)	
50	MASK (FFFFFFF);	
51		
52	A Deserve Deserves and see	
53	: Frogram bscan register	
55	Shift in Preload(0x1C) instruction	
56	SIR 8 TDI (1C);	
		TEFFFFFFF Y
× .		3

図 15-32. SVF オプション設定

Ontions	Download	🔊 Download Debugger - [C:/usr_ss/DssPkgs/E5gdmi/d3Ce5gdn					
e options	SWF File Edit	Command	Cor	figuration	Window	Help	
General SVF STAPL	0 - FR		Z	Cable and	I/O Port Se	tup	
Disable SVF syntax checker	: •••••		Î¥	Options			
☐ Ignore IR and DR header/trailer in SVF							
Continue on error							
Mixed chain							
Instruction register header: 0 🗢 TCK frequency:	1	-					
Instruction register trailer: 0 🕏 SVF vendor:	JTAG STAND	ARD 🔻					
Data register header: 0 🗲 Starting TAP st	ate: TLR	-					
Data register trailer: 0 😫							
	OK	Cancel					

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: ヘッダー)と後ろ (trailer: トレイラー) 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 15-32、 赤枠)。ない場合は0です。図 15-33 にヘッダー長、トレイラー長の設定例を示します。

図 15-33. JTAG チェインの構成とレジスタ長の例



---- *** ----