# macnica

モジュール / IP 生成

Lattice Diamond

2021 年 12 月

Lattice Diamond 日本語ユーザーガイド

## 第4章 モジュール / IP 生成

## 4.1 概要

Lattice FPGA ファミリー固有のモジュールや Lattice オリジナル IP のパラメータ設定と生成用ツールは、 IPexpress および Clarity Designer です。カウンターや、汎用的な機能モジュールを実装する場合、論理合成 ツールに依存する推論(Inferencing)ベースの手法は、ターゲットフリーという長所はありますが、IPexpress/Clarity Designer で生成したモジュールを RTL 記述で明示的にインスタンスした方法に比較して、エリ アやスピード的に劣る場合が一般的です。

ECP5 ファミリーと Crosslink では Clarity Designer が、その他のファミリーでは IPexpress が用いられます。 これらの機能比較概要を表 4-1 に示します。

		IPexpress	Clarity Designer
°-, 6	モジュール	Yes	Yes
ハフメータ設定、 生成	IP	Yes	Yes
/~	IP ダウンロード	Yes	Yes
	デザインルールチェック	No	Yes
ы́ II Lí	接続情報の生成	No	Yes
	配線支援	No	Yes
	再利用性	No	Yes
	論理合成前処理	No	Yes
	配置支援	No	Yes
J = / GDDR	デザインルールチェック	No	Yes
	GUI 対応	No	Yes

#### 表 4-1. IPexpress と Clarity Designer の機能比較

Clarity Designer については第4.3節で詳述しますが、IPexpress が単一のモジュールや IP を個別に生成する のみであるのに対して、Clarity Designer は"サブシステム"と呼ばれる、複数のモジュールやマクロを相互 接続指定して、上位階層の機能ブロックとして扱うことが可能な RTL と設定ファイルを生成します。また、 Clarity Designer はモジュール /IP のパラメータ設定と生成に加えて、『ビルド』および『プランニング』機能 が追加されています。

特に ECP5 における DDR メモリー・インターフェイスの実装時は、PCB 設計の前に最適なポート配置を Clarity Designer のプランニングで決定することを強く推奨します。『バーチャル VCCIO (GND)』の配置も推 奨(ツールが自動的に選択)されていますので、関連テクニカルノートを参照するとともにご留意ください。

## 4.2 IPexpress

## 4.2.1 ウィンドウの起動

Clarity Designer 非対応デバイスでは IPexpress を起動します。メニューバーの Tools をクリックすると表示 されるツール群の中から IPexpress を選択する (図 4-1、左)か、アイコンメニューから ピ をクリックし ます (同、右)。

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 た場合は正規代理店の技術サポート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。



図 4-1. IPexpress の起動(左:アイコンから、右:メニューバーから)



初期画面は図 4-2 のようになります。

#### 図 4-2. IPexpress 初期画面 (MachXO3L の例)

Name Name Module Module Modules M	Versic 4.2 5.7 5.7 2.0 5.7 5.7	Module/IP Name Macro Type: Version: IP Name: Project Path: File Name: Nodule Output: Device Family: Part Name:
Name A OB Module A OB Architecture_Modules A OD A A DDR A DDR_GENERIC A DDR_MEM A DDR A	4.2 5.7 5.7 2.0 5.7 5.7 5.7	Module/IP Name Macro Type: Version: IP Name: Project Path: Browse File Name: Module Output: Part Name:
<ul> <li>▲ Module</li> <li>▲ Architecture_Modules</li> <li>▲ ▲ IO</li> <li>♣ DDR</li> <li>⊕ DDR_GENERIC</li> <li>⊕ DDR_MEM</li> <li>₩ DQS</li> <li>₩ GDDR_7:1</li> <li>₩ SDR</li> </ul>	4.2 5.7 5.7 2.0 5.7 5.7 5.7	Macro Type:       Version:         IP Name:       IP Name:         Project Path:       Browse         File Name:       Browse         Module Output:       *         Device Family:       Part Name:
닳 DLL         닳 Digital CDR         닳 Dynamic Bank Controller         닳 EFB         닳 ORCAstra         닳 PCS         잫 PLL         잫 Power Controller         잫 Power Guard         잫 System_Bus         닳 Tag Memory         잫 Arithmetic_Modules         잫 Adder	3.5 1.0 1.4 1.2 1.0 8.2 5.6 1.2 1.2 1.2 4.0 2.0 3.4	Synthesis: SynplifyPro Please enter a project directory/ Customize

左枠の Name 部は Module セクションと IP セクションに分かれており、それぞれがサブセクションに分割 されています。サブセクションの構成は図 4-3 のようになっています(意図的に詳細を隠した状態。IP セク ションは一例)。

Architecture\_Modules 下には図 4-3 のようなマクロが表示され、これは各デバイスファミリー固有の機能ブロックが分類されています。従って表示アイテムはファミリー毎に異なります。図 4-2 は MachXO3L の場合です。

Arithmetic\_Modules と DSP\_Modules、および Memory\_Modules 下のアイテムは図 4-4 のようになります。 Arithmetic\_Modules はカウンターや演算機能を LUT+FF で構成する場合の機能ブロックです。DSP\_Modules は 演算機能一式の機能ブロックで、パラメータを指定する際の個別詳細設定 GUI で DSP 用ハードマクロを用い るか、LUT+FF で実現するかを選択できます。Memory\_Modules セクションはオンチップメモリーの生成用で、 Distributed\_RAM 部は LUT+FF を用いた構成(分散メモリー)、EBR\_Components はブロックメモリー(EBR) を用いた構成です。どちらにも属さない FIFO と FIFO\_DC および RAM\_Based\_Shift\_Register は個別詳細設定 GUI で EBR を用いるか、LUT+FF で実現するかを選択します。

図 4-3. IPexpress の Name 枠の標準的なセクション(項目)構成

MACNICA

E 🎜 🚱 🔣 🐁 地		
Name	Version	Module/IP Name
<ul> <li>Module</li> <li>Architecture_Modules</li> <li>Arithmetic_Modules</li> <li>DSP_Modules</li> <li>Memory_Modules</li> <li>IP</li> <li>Connectivity</li> <li>DSP</li> <li>Processors, Controllers and Peripherals</li> </ul>	5	Macro Type: IP Name: Project Path: File Name: Module Output: Device Family: Part Name: Synthesis:SynplifyPro

なお、各名称行頭にあるアイコンのうち 🚧 印は、選択しているデバイスでは未サポートを意味します。

デバイスの選択が正しいか(意図したものか)を含めて、確認します。例えば MachXO3L には DSP マクロがありませんので、DSP\_Modules 部にある各マクロは同印が表示されて選択できません。

図 4-4. Module 部サブモジュール毎のマクロ項目(MachXO3Lの例)



Name 枠の下部は IP セクションになっています。ここに表示されるアイテム・構成はデバイス毎に異なり、 また後述のようにユーザがダウンロード、インストールする(インストール済み) IP に依存します。

## 4.2.2 モジュール生成手順

カウンター・モジュール生成の具体例を用いて、手順・方法の概要を記述します。

まず該当するセクションから Counter を選択(クリック)します。右枠の中に所定の情報が表示されます。 ユーザが入力するのは赤枠長方形内です。Project Path はデフォルトでインプリメンテーション・フォルダー がロードされています。変更する場合には Browse ボタンをクリックして所望のフォルダーを指定します。 File Name はモジュールの名称(インスタンス名)です。Module Output は初期表示は図 4-5 のようにブラン クですが、行の一部をクリックするとプルダウン形式で HDL 言語を選択できます(図 4-6)。VHDL を選択 した場合は、本体記述が Verilog でその上のラッパー・モジュールが VHDL という構造になります。

これらを全て入力・指定後、右下の赤丸内に示す『Customize』ボタンをクリックします。いずれかが入力 されていないとグレーアウトのままで次に進めませんので留意します。その後に表示されるウィンドウでモ ジュールのパラメータ指定を行います。



## 図 4-5. カウンターマクロの生成準備

Name	Version _	Counter 4.5
<ul> <li>Module</li> <li>Architecture_Modules</li> <li>Arithmetic Modules</li> </ul>		Macro Type: Module Version: 4.5 Module Name: Counter
Adder	3.4	Project Path: C.¥aUSER_SStasks¥D33work Browse
₩ Adder_Subtractor	3.4	File Name: counter32b
a Comparator	3.5	Module Output
鐸 Complex_Multiplier	1.7 =	Device Family: LatticeECP3
辞 Convert	1.3	Part Name: LFE3-35EA-8FN484C
🐺 Counter	4.5	Synthesis: SynplifyPro
ə FFI_Butterfly	1.4	
錘 LFSR	3.6	Please select "Module Output"!
每 Mult_Add_Sub	2.7	Customize
CLASSE AND C.L. C	2.2	

## 図 4-6. モジュール記述言語の指定

Module Output:	*
Device Family:	Verilog
Part Name:	VHDL CALOTIVIONO

カウンターの場合、図 4-7 がパラメータ設定ウィンドウになります。対象モジュールに拘わらず、数値入 カセルとプルダウンによるオプションや値の選択のセルに分かれますが、基本的に全て所望の設定を行う必 要があります。本例では上部に『Configuration』タブが一つあるのみですが、モジュールによっては複数の タブがありますので、全タブを確認します。

入力・指定完了後、下部の『Generate』ボタンをクリックすると、指定した言語の RTL と設定ファイル (拡張子 .ipx) とともに、付随する複数のファイルも生成されます。このときに、図 4-7 の下部左にある 「Import IPX to Diamond Project」にチェックすることで、.ipx ファイルが作業中のインプリメンテーションに 自動的にインポートされます (図 4-9)。チェックしない場合は、手動でインポートします。

図 4-7. Counter のパラメータ設定(Configuration) ウィンドウ

Counter	Configuration \			
	Specify the data width of	the counter:		
	8	(1 - 64)		
	Specify the direction of the	e counter:		
→ Clk_En 0[7:0]→	Up	•		
	Continuized for speed			
Acir	Lower count value	0	(0 - 2^8-1)	
	Upper count value	255	(0 - 2^8-1)	
	Enable load input			
Estimated Resource Usage: UT:10 Reg:8				
Big Endian (MSB:LSB)				

Generate ボタンをクリックした後は、図 4-8 のような GUI が表れます。必ず Total Warnings / Errors が 0 であることを確認後、Close ボタンで終了します。0 でない場合は何らかの問題があります。

問題がなく『Close』ボタンをクリックすると、自動インポート指定の場合は当該インプリメンテーションの "Input Files" セクションに.ipx が追加されます。

図 4-8. モジュール Generate 後の表示例

Configuration	Generate Log	
Generated Log	Errors/Warnings Msg	
EDIF output Verilog output Verilog temple Verilog temple Verilog purpo Bus notation Report output Estimated Re	Counter32b edn : counter32b v the counter32b, tmpl v thch th <sub>0</sub> counter32b, tmpl v is or synthesis and simulation : big endian : counter32b srp : counter32b srp : counter32b srp : counter32b srp	
Reg:8		
END SCUBAN	ladule Synthesis	
File: counter32b	lpc created.	
End process: co	mpleted successfully.	
Total Warnings	•	
Total Errors: 0		
$\sim$		
*		
	c	lose Help

.ipx ファイルはツールが必要なパラメータ情報を全て含む、Diamond 固有のテキストファイルです(ユー ザーは編集しないようにします)。ipx ファイルを取りこむ利点は、パラメータの変更が容易な点があります。 プロジェクト進捗に伴い、或いはその他要因で変更する可能性がある場合が少なくありませんので、.v /.vhd ではなく、.ipx のインポートを推奨します。

図 4-9. IPX チェック後生成して自動で取り込む例 (File List 枠内 Input Files 部)



設定変更は意図する.ipx 行をダブルクリックします。設定済みパラメータをロードした状態で図 4-7 のパラメータ設定ウィンドウが表示されます。

RTL は上位モジュールでインスタンス記述する場合(同時に生成されるテンプレート・ファイルの使用を 推奨)や、論理シミュレーションで用います。RTL 記述をチェックしたい場合も.v/.vhd を参照します。

インプリメンテーションには、単一モジュールに対して .v /.vhd と .ipx 両方をインポートする事も可能で すが、必ず一方を右クリックして [Exclude from Implementation] 指定するようにします (グレーアウトされ る)。両方をアクティブなままにしておくと、予期しない問題が生じる恐れが否定できません。

## 4.2.3 特定モジュール生成時の留意事項

#### 4.2.3.1 PLL モジュール

PLL モジュールの生成時のパラメータ設定ウィンドウ例を図 4-10 に示します。

[Configuration] タブ表示の直下ではパラメータ指定モードが選択できます。"Frequency Mode"は入出力クロックの指定セルに周波数を与える場合(デフォルト)、"Divider Mode"は入出力ポートに付随する分周器の値を直接与える場合に選択するものです。通常は前者を使用します。後者を使用する場合は分周値を求める

際に十分な知識が求められます。その詳細やパラメータ入力項目についての定義などは、各デバイスファミリーのクロック /PLL 関連テクニカルノートをご参照ください。

図 4-10. PLL モジュールのパラメータ入力例(MachXO3L)



通常のモジュール生成では一通りパラメータを設定し終えると『Generate』ボタンがアクティブになりま す。PLL に限り、パラメータ入力後に右下の赤枠で示す『Calculate』ボタンをクリックする必要があります (デバイス・ファミリーによってボタンの位置は異なります)。これによって DRC (ルールチェック)が実行 され、入力された周波数や分周値で PLL が規定内動作可能かどうかをツールがチェックします。周波数値が 所望にならなかったり ([Tolerance] で指定)、最低・最高周波数範囲に収まらなかったり、ということがな いと確認された場合に限り、『Generate』ボタンが有効になり、クリックできます。無効の場合は、設定内容 を再吟味する必要があります。

なお、Calculate 後に左上部 CLKI セクションの下部にある [~Bandwidth] と [VCO Freq.] セルに計算結果が 表示されますが、これらはそれぞれ閉ループ帯域幅と VCO 周波数です。

## 4.2.3.2 ROM モジュール

分散メモリ(Distributed)ベースでも EBR ベースでも該当する留意事項についてです。ROM はコンフィグレーション時にメモリー内の値が初期化され、リードアクセスのみが許容されます。従って、一般のモジュールと異なり、ROM モジュール生成時に初期化データを与える必要があります。図 4-11 の赤枠が該当する入力部です。

初期化ファイルは"拡張子を〈任意〉.mem とするテキストファイル"という約束があります。「Memory File」セルに、ブラウズしてファイルを選択・指定します。テキストファイルの初期値を記述するフォーマッ トが決められていて、図中のように Binary か Hex、または "Addressed Hex" のいずれかである必要がありま す。それぞれのフォーマットは、図 4-12 に示すような形式です。詳細は各デバイス・ファミリーのメモリー 関連テクニカルノートをご参照ください。



図 4-11. ROM モジュールのパラメータ入力ウィンドウ例(MachXO3L)

ROM	Configuration \
	Specify the size of the ROM
Address[4:0]	Address Depth 32 (2-65536) Date Width 16 (1-256)
OutClock	F Enable Output Register
Q[15:0] →	Optimization C Area @ Speed
OutClockEn	Reset Mode
	C Async C Async C Async C Sync
	Initialization
Estimated Bespurce Usage:	Memory File
corer	Memory File Format & Binary C Hex C Addressed Hex
	Enable ECC (not supported for Data Width > 64)
	Pipeline Stages for O and EHHOH Outputs IV
Bus Ordering Style:	
Big Endian (MSB LSB)	

これらを全て入力後『Generate』クリックして、モジュールを生成しますが、形式が一致、かつ初期化ファイルに記述されるデータ語長とデータ語数が、GUI中の[Address Depth] と [Data Width] の値と一致するようにします。設定より mem ファイルが大きい(長い)と、生成ステータス表示ウィンドウで結果がエラーになります。そのまま抜けてもモジュールは生成されていません。設定より mem ファイルが小さい(短い)と、エラーにはならず、不足分は全て"0"としてモジュールが生成されます。

図 4-12. 初期化.mem ファイルの書式概要(左: Binary、中: Hex、右: Addressed Hex)

00100000010000000010000001000000 0000000	A001 0B03 1004 CE06 0007 040A 0017 02A4	–A0 : 03 F3 3E 4F –B2 : 3B 9F
---	--	----------------------------------

## 4.2.4 IP の生成準備

Diamond インストール直後はラティス提供の IP が使用できる状態にはなっていませんので、初めに意図する IP を (個別に) ダウンロードしてインストール、その後パラメータ設定を行い、生成します。

まず、ダウンロードするためにラティスの IP サーバーにアクセスします。 Ø をクリックし (図 4-13、 左)、次に "IP (Click to get IP information)" をクリックします (同図、右)。

#### 図 4-13. ラティス IP のダウンロード

🊯 Start Page 🗵 📑 Reports 🗵 🔠 IPexpress 🖾	🚯 Start Page 🛛 🛄 Reports 🔍	🔡 IPexpress 🛛
S S & S & S	2 2 2 2 2	
Name Version	Name	Version
	IP (Click to get IP information)	

なお、ここで PC 環境はネットワークにアクセスしている必要があります (Proxy 設定など、何らかの問題 があるとメッセージが表示されます。

## macnica

IP サーバーが応答後は図 4-14 左のように、ターゲットとしているデバイスに対応する IP のリストがバー ジョン情報と共に表示されます(グレー表示や古いバージョンは対象としないことを推奨します)。カテゴ リーは 5 つに分類されています。"Connectivity" には PCI Express や SDI (SMPTE) などが、"DSP" には FIR Filter / Scaler / Viterbi / CFC Filter / CORDIC / CSC / FFT / NCO などが、"Processors, Cont..." には DDR2/3 Controller などが含まれています。

図 4-14. IP サーバー応答後の表示例(MachXO2、左:表示直後、右:カテゴリの操作後)

_				_					
Na	me		Version		Nam	ne			Version
~	$\bigcirc$	IP			~ (	3	IP		
	$\sim$	lease Audio, Video and Image Processing				>		Audio, Video and Image Processing	
		12 4:1 CSI-2 to CSI-2	1.0			>		Communications	
		12 4:1 CSI-2 to CSI-2	1.1			>		Connectivity	
		P Byte to Pixel Converter	1.0			~	<u></u>	DSP	
		🌈 Byte to Pixel Converter	1.1				1	IP 2D Edge Detector	1.0
		P Byte to Pixel Converter	1.2				1	B 2D Edge Detector	1.1
		P Byte to Pixel Converter	1.3				1	1P 2D FIR Filter	1.0
		P CMOS to D-PHY	1.1				1	IP 2D FIR Filter	1.1
		P CMOS to D-PHY	1.2				1	IP 2D Scaler	1.0
		P CMOS to D-PHY	1.3				1	1P 2D Scaler	2.0
		P CSI-2 to CSI-2 Repeater/Splitter	1.0				1	IP 2D Scaler	2.1
		P CSI-2 to CSI-2 Repeater/Splitter	1.1				1	B 2D Scaler	2.2
		P CSI-2 to CSI-2	1.1				1	P Block Convolutional Encoder	3.5
		P CSI-2 to CSI-2	1.2				1	IP Block Convolutional Encoder	3.6
		P CSI-2 to CSI-2	1.3				1	P Block Convolutional Encoder	3.7
		P CSI-2/DSI D-PHY Receiver	1.0					IP Block Viterbi Decoder	4.5
		CSI-2/DSI D-PHY Receiver	1.1				1	IP Block Viterbi Decoder	4.6
		P CSI-2/DSI D-PHY Receiver	1.2				1	IP Block Viterbi Decoder	4.7
		P CSI-2/DSI D-PHY Receiver	1.3				1	IP CFR	1.0
		P CSI-2/DSI D-PHY Receiver	1.4				1	P CIC Filter	3.1
		P CSI-2/DSI D-PHY Receiver	1.5					P CIC Filter	3.2
		B CSL 2/DSLD BHV Transmitter	1.0				-	P CIC Filter	3.3

IP をダウンロードするには、所望の IP / バージョン行を選択してアイコン 🗈 をクリックするか、選択 行を右クリックすると表示される [Download <IP 名、バージョン番号]"を選択します。この場合は、ダウン ロードしたパッケージを元に、手動でインストールする必要があります。ダウンロードとインストールを連 続して自動で行うには 💽 アイコンをクリックするか、選択行を右クリック後表示される [Install <IP 名、 バージョン番号 >] を選択します。ダウンロードのみを選択した場合は、最初にダウンロード先のフォルダー

ハーション番号 汀を選択します。タリンロードのみを選択した場合は、最初にタリンロード先のフォルター 選択を促す表示が出ますので指定します。デフォルトの "C:¥LatticeCore" が適切(推奨)です。ダウンロー ド・ファイルは実行形式 "<IP 名、バージョン番号 >.exe" のようになっていますので、ダブルクリックして インストーラを実行します。

手動でインストールした場合で、IPexpress や Clarity Designer を起動してもインストール済み IP リストに 表示されないときには、"C:¥lscc\_env" にあるテキストファイル "ipsetting.lst" を編集します。書式例は次の通 りです:

[Installed\_IP] pci\_express\_endpoint\_v6.4=C:¥LatticeCore sgmii\_gbepcs\_v4.1=C:¥LatticeCore ddr3\_sdram\_common\_v3.1=C:¥LatticeCore

## 4.2.5 IP のパラメータ設定と生成

DDR3 メモリーコントローラ IP (LatticeECP3) の具体例を示すことで、手順・方法の概要を記述します。 まず Name 枠内で当該 IP 行を選択し、右枠に図 4-5 と同様の要領で必要な入力 (Project Path、File Name、 Module Output) を完了後、『Customize』ボタンをクリックします。

## 

MACNICA

Name	DDR3 SDRAM Controller 3 Geer
<ul> <li>Module</li> <li>IP</li> <li>Connectivity</li> <li>PCI Express Endpoint Core</li> <li>PCI Express Endpoint Core</li> <li>PCS PIPE</li> <li>DSP</li> <li>CORDIC</li> <li>Processors, Controllers and Peripherals</li> <li>DDR3 SDRAM Controller</li> </ul>	Macro Type: User Configurable IP Version: 3.0esr IP Name: DDR3 SDRAM Controller Project Path: C:¥aUSER_SStasks¥D03work Browse File Name: ddr3ip64b Module Output: Verling • Device Family: LatticeECP3 Part Name: LFE3-35EA-8FN484C Synthesis: SynplifyPro

次に表示されるのがパラメータ入力・設定用 GUI です。これは勿論 IP ごとに異なります。DDR3 メモリー コントローラ(LatticeECP3)の場合は図 4-16 のようになります。

#### 図 4-16. DDR3 メモリーコントローラ IP のパラメータ設定

DDR3 SDRAF = rst_n = scik_2x = eck = scik = scik	4 Controller em_ddr_clk→ em_ddr_clk→ em_ddr_cck→ em_ddr_csn→ em_ddr_csn→ em_ddr_csn→ em_ddr_csn→ em_ddr_csn→ em_ddr_csn→ em_ddr_csn→	Type \ Setting \ Memory Device Timing \ Pin Selection \ Design tools Option & Into \       Device Information       Select Memory (Micron DDR3 1Gb -{SE       PatClock(MHz): 100       MemOlock(MHz): 400       Device: LafficeECP3       Part LFE3-35EA-8FN484C       Memory Configuration       Memory Type: Unbuffered DIMM       Memory Date Bus Size: 32       Configuration: 68       CKE Width: 1       DIMM Type: Single Rank       Additionel Configuration       F 2T MODE	
<ul> <li>M_ent</li> <li>read_data[127:0]</li> <li>read_data_valid</li> <li>int_dicat_valid</li> <li>int_dicat_rdy</li> <li>dista_rdy</li> <li>dista_rdy</li> <li>dista_rdy</li> <li>w_int_datapath</li> <li>uddontin</li> </ul>	em_ddr_dm(3.0) em_ddr_dqs(3.0) em_ddr_dqs(3.0) em_ddr_reset_n →	Deta_rdy to Write Data Delay: 1	
,			

モジュール生成時と同様に、各セルに適宜入力した後、『Generate』ボタンをクリックします。GUI 上部の 赤枠のように、IP によっては複数のタブがありますので、留意します。また、前出と同様に「Import IPX to Diamond Project」オプションがあり、チェックすれば.ipx ファイルを自動的にインポートします。

IP の場合、Generate をクリック後それまでに論理合成が実行されますので、ある程度の処理時間を必要と します。図 4-8 と同様な "Total Errors 0" が表示される前に『Close』ボタンをクリックしないように注意し ます。誤ってクリックした場合は、再度手順を繰り返す必要があります。処理時間は IP やパラメータに依存 して変わります。

なお、図 4-15 では右枠最下段の [Synthesis] 部表示が "Synplify Pro" になっています。プロジェクト生成時 やインプリメンテーションの設定時に選択して有効になっているツール名が表示されます。



## 4.3 Clarity Designer

## 4.3.1 Clarity Designer & IPexpress

IPexpress が生成するモジュールの出力ファイルは Diamond 固有のフォーマットである <module\_name>.ipx と RTL (<module\_name>.v /.vhd) などです。これに対し、Clarity Designer では <sub\_system\_name>.sbx と RTL などです。Clarity Designer ではやや上位の概念で"サブシステム"という語を用います。単一の .sbx ファイルには含む複数のモジュール (又はコンポーネント) が存在することが許容されます。

Clarity Designer 生成サブシステム(.sbx)は基本的に下位モジュールとして使用します。実装設計の柔軟性としての観点から、Clarity Designer 生成サブシステム sbx をトップとして据える構成は推奨しません。その留意点は以下の通りです。

- ・ sbx に相当する RTL トップをユーザ記述 HDL 内でインスタンスする
- ・インスタンス時はテンプレート (<subsystem>\_tmpl.v/\_tmpl.vhd) を元にすることが推奨

## 4.3.2 Clarity Designer で可能な作業

Clarity Designer で可能なタスクは以下の通りです(詳細は後述します)。

- ・ IP のダウンロードとインストール
- ・モジュールや IP のパラメータ設定
- ・ IPexpress ファイル .ipx をインポートして .sbx に変換
- ・サブシステムとして下位モジュール相互を接続指定([Builder]タブ)
- ・ PCS/SERDES や DDR エレメントの配置指定([Planner] タブ)

他方、PCS / DDR 以外の汎用 IO の配置指定はできませんので、従来通りスプレッドシート・ビューか LPF 制約ファイルで行います。また、Clarity Designer 対応のプロジェクト / インプリメンテーションに関する留 意点は以下です。

- ・ 単一 sbx ファイルには複数のモジュールが存在可能。複数のモジュール間を接続する信号線がある場合はサブシステム内で行う(外部引き出しも可能。図 4-17 の例では色枠内の "EBR+DSP")
- · PCS や DDR モジュールが複数の場合は、各々個別のサブシステムにし、適切な配置指定を行う
- ・ サブシステム内の(複数)モジュールを設定後、終了する前に必ず『Generate』する
- ・サブシステムのパラメータ等、何らかの編集や変更した場合は、必ず再度 Generate する

図 4-17. サブシステムの想定・非想定ユースケース



## macnica

特に重要な"非推奨事項"があります。サブシステム内に複数の独立したモジュールを含めて Generate 後、 生成された Verilog 記述の中から、特定モジュールに相当する部分のみを抜き出して、上位モジュール記述で インスタンス使用すること(図 4-17 右)は想定されていません。単独で用いるモジュールは単独のサブシ ステムを生成して用いるようにします(図 4-17 左)。

## 4.3.3 Clarity Designer の起動

FIFO\_DC モジュール生成の例を用いて手順概要を記述します。

## 図 4-18. Clarity Designer の起動



Clarity Designer の起動はアイコンをクリックするか、メニューから [Tools] → [Clarity Designer] とたどりま す (Generate 後、File List に取り込まれる .sbx 行をダブルクリックしても立ち上がります)。

初めに立ち上がるのは図 4-19 のようなウィンドウです。赤枠内を選択・記述します(下部の Diamond Project セクションが正しいことを確認します)。「Design Name」セルには"サブシステム"的な意味での名称を入力します。完了後、『Create』ボタンをクリックします。

## 図 4-19. サブシステム作成の始め

	🖁 Clarity Designer		? )	×
	Create new Clarity d	lesign		
	Design Location:	C:¥usr_ss¥cathandsOct182021_	Browse	]
	Design Name:	MIPIcsi2rx		
	HDL Output:	Verilog	•	•
	Open Clarity design			
	Design File: C:/usr_ss/	/cathandsOct182021_/byte2pix_fudo/IPcat/byte: 👻	Browse	
IPexpress ライクな	Start Clarity Designer Clarity Designer will en click the module on Cla IP generation is not ava generation Diamond Project	er to generate a single Component SBX ter single module generation mode. In this mode, you rity Designer catalog to start generating a single mo illable in this mode, please do not enable this mode f	can double- dule SBX. or IP.	
モジュール生成時	Diamond Design Name:	xInk_byte2pix		
にチェックする	Diamond Design Path:	C:/usr_ss/cathandsOct182021_/byte2pix_fudo/fit		
	Part Name:	LIF-MD6000-6KMG80I		
	Synthesis:	LatticeLSE		
		Create Cancel	Help	

ここで、ウィンドウ中央にオプション「Start Clarity Designer to generate a single Component SBX」があり ます。サブシステムには単一のモジュールしか含まない場合に限り、これをチェックすることで IPexpress と 同じ扱いで生成することができます。

## 4.3.4 モジュールのパラメータ設定 (Configuration)

次に表示されるのが図 4-20 のようなウィンドウです。

図 4-20. カタログタブ (Module 部各セクションを畳んだ後の例)

	🄄 Start Page 🗵 🛛 🛄 Repo	ts 🗵	ቶ Clarity	Design	er 🗵	
	🗞 DRC 🔗 Generate					
	Catalog Builder Planner					
	Name		v	ersion		
	🔺 🖄 Module 🔵					
	Architecture_Modules					
	Arithmetic_Modules					
	DSP_Modules					
	4 🖄 Memory_Modules					
	Distributed_RAM					
	#					
	묠 ram_dp		6	.4		
	₽ ram_dp_true		7	.4		
	æram_dq		7	.4		
	룓 rom		5	.3		
	霽 fifo		5	.0		
	₩ fifo_dc		5	.7		
	<pre>zw ram_based_snift_r</pre>	egister	5	.1		
	4 🖄 IP					
	Connectivity					
	🚮 pci express endpoi	nt core	6	.0_asr		
- 1		ener I	mont IP	Diamo	bo	
			in port if	D-amoi		

IPexpress とほぼ同じ画面ですが、上部には三つのタブ、[Catalog] と [Builder]、および [Planner] があり、 『Catalog』タブが選択された状態で立ち上がります。"Module" 部は IPexpress と同様です。その下には IP 部 があり、その時点でインストール済みの IP リストが表示されます。

ここで、"コンフィグレーション(Configuration)"という語は、パラメータ設定、或いはオプション設定 をすることを意味します。本章では同じ意味で"パラメータ設定"と既述します。[Catalog] タブでモジュー ルや IP のパラメータ設定を行います。これは IPexpress と同様です。

下部には四つのタブがあり、[Lattice IP] タブが選択された状態です。新たにラティスの IP サーバーから IP をダウンロード・インストールする場合は、[Lattice IP Server] タブを選択して作業します(第4.2.4節参照)。

#### 図 4-21. インスタンス名入力ウィンドウ

Name		
Instance Path:	/D33work/e	cp5pcie/e5pipe32b/impl_proj/fifodc_32x16
Instance Name:	fifodc_32x16	
fifo_dc 5.7		
Macro Type:	Module	Version: 5.7
Maco Name:	fifo_dc	
Module Output:	Verilog	-
Device Family:	ECP5UM	
Part Name:	LFE5UM-45F-6	6BG554C
Synthesis:	synplify	
		where we are set to the second set of a second

例として "fifo\_dc" モジュール行をダブルクリックします。図 4-21 のようなウィンドウが表示されますの で、インスタンス名を「Instance Name」セルに入力後、『Customize』ボタンをクリックします。図 4-22 のよ うなパラメータ設定ウィンドウが立ち上がります。

ここで、「Instance Name」入力は先に入力したサブシステム名(Design Name)と別にする必要があります。

同一の場合は『Customize』ボタンがアクティブになりません。

## 図 4-22. FIFO\_DC モジュールの生成

FIFO_DC	Configuration
→ Deta[11:0] O(47:0] → → WrClock Empty →	FIFO Implementation     C     LUT Only       © EBR Only     C     LUT Only       Write Address Depth     32     ✓     Date Width       Read Address Depth     8     ✓     Date Width       IF     Enable Output Register     ✓     Controlled by RdEn
→ WrEn Full → PidEn AlmostEmpty → Reset	Flag Control         Static - Dual Threshold         Image: Control Contron Control Control Contro Contron Control Control Co
AlmostFull AlmostFull	Reset Mode     Assertion     Release       If Assertion     If Assertion     If Assertion       If Assertion     If Assertion     If Assertion       If Data Count Options     If Data Count (Synchronized with White Clock)     If Data Count (Synchronized with Read Clock)
Bus Ordering Style: Big Endian (MSB:LSB)	Enable ECC (not supported for Data Width > 64)

IPexpress との第一の違いはウィンドウ下部にあるボタンで、『Generate』ではなく『Configure』です。 Generate は後述するビルドステップで行います。第二の違いは IPexpress でウィンドウ左下にあったオプション [Import IPX to Diamond Project] がない点です (Clarity Designer では『Generate』後に sbx ファイルが自動 的に取り込まれます)。

## 図 4-23. モジュール生成後の GUI 例

Configuration	Generate Log			
Generated Log	Errors/Warnings Msg			
Issued comm Circuit name Module type Module Versi Ports	iana :: Uysocyaiamono(3.3) : filodo_rx8to32 : ebfilo ion : 5.7	xx+v/sptpgalpim/m+v/scuba.ex	kerwin trooc_txotosz Heng W	briog-synth synpil
UD buffer : EDIF output Verilog output	: O(47:0), WrGock, Ho : O(47:0), Empty, Full, Als not inserted :ftodc_rx8to32.edn A :ffodc_rx8to32.v	usok, witzi, Hatzi, Heset, He nostEmpty, AlmostFull	Heset	
Verilog templ Verilog testbo Verilog purpo Bus notation	late:fifodc_n @to32_tmpl.v ench:tb_fifodc_n @to32_tmpl ose:forsym thesis and simu :big endian	v lation		
Estimated Re LUT : 129 EBR : 2 Reg : 74	t : filodc_n/8to32.srp esource Usage: i			
END SCUBA	Module Synthesis			
File: filodc_n/8k	o32.lpc created.			
End process: o	ompleted successfully.			
Total Warnings	c 0			
Total Errors: 0				
•	ш			
			Close	Halo
			GIUSE	neip

IPexpress 同様にモジュール生成が正常終了したことを確認してから『Close』します。

Clarity Designer 特有の機能で、単一サブシステム配下に別のモジュールをインスタンスして相互接続する 場合などは、『Configure』ボタンで設定完了後に、[Catalog] タブを選択して、二つ目以降のモジュールにつ いて同じ手順を繰り返します。

## 4.3.5 ビルド

Clarity Designer では、[Builder] タブでの設定後に『Generate』ボタンをクリックしないとサブシステムの 生成は行われませんので、注意が必要です。本タブで行う作業は以下の通りです:

- ・ウィンドウ下部の [Components] タブを選択して、コンポーネントの I/O ポートをサブシステム外
- (上位)に引き出す("Export する")かどうかの設定
- ・ 複数コンポーネント (インスタンス)間の接続

デフォルトではコンポーネントの全ポートは Export セルがチェックされていて、上位に引き出されていま す (Type = "fabric")。その必要がない場合はチェックボックスをクリックして Type を "un-connected" とし ます。一旦 Generate した後に設定変更してポートが増える場合、基本的にそのポートの初期設定は "unconnected" になります。上位モジュールで必要なケースでは、Export します。

Type カラムにはこれ以外に "internal" と "pad-pin" があります。前者は [Planner] タブでの接続指定、後者 はデバイスのパッケージ I/O との直接接続を意味します。なお、ポートを 0/1 固定レベルに接続することは できません。また、論理式(回路)と接続することもできません。Connection 列のセルはユーザが任意に名 称を変更できます(モジュールのポート名になる)。

#### 図 4-24. FIFO\_DC モジュールの接続設定例

🙆 Start Page 🛛 🔚 Re	ports 🗵 🛛 📅 Clarity Designe	r 🗵	
😸 🔍 🧠 🕵 🕷 DRC 🕻	🌛 Generate		
Catalog Builder Planne	r		
Components	Connection	Туре	Export
▲ fifodc_rx8to32			
AlmostEmpty	fifodc_rx8to32_AlmostEmpty	fabric	<b>v</b>
AlmostFull	fifodc_rx8to32_AlmostFull	fabric	<b>v</b>
<ul> <li>Empty</li> </ul>		un-connected	
G Full		un-connected	
RPReset	fifodc_rx8to32_RPReset	fabric	<b>v</b>
RdClock	fifodc_rx8to32_RdClock	fabric	<b>v</b>
RdEn	fifodc_rx8to32_RdEn	fabric	<b>v</b>
Reset	fifodc_rx8to32_Reset	fabric	<b>v</b>
WrClock	fifodc_rx8to32_WrClock	fabric	<b>v</b>
WrEn	fifodc_rx8to32_WrEn	fabric	<b>v</b>
Data[11:0]	fifodc_rx8to32_Data[11:0]	fabric	<b>v</b>
▷ 😫 Q[47:0]	fifodc_rx8to32_Q[47:0]	fabric	<b>v</b>
Components Schematic			

次にコンポーネントのポート同士を接続する例を図 4-25 に示します。

"Control" キーを押しながらクリックすることで複数の信号を選択し、そのいずれかの上でマウス右クリック後に [Connect] を選択します。同一のソース信号に接続する複数の入力信号は何本でも同時に選択できますが、出力信号は一本のみです。この場合、入力信号を一本のみ "Export" して他を un-connected にしないと "Connect" アクションが選択できません。一本を Export する場合、出力信号は必ずしも選択しなくて構いません。

"Connect" されると、"un-connected" 指定した入力ポートも Type 表示が "fabric" に変わります。接続を解除するには、再度右クリックして [Disconnect] を選択します。

図 4-25. 入力ポート同士を接続する例

Catalog Builder	Planner					
Component	ts	Conr	nection	Туре	Export	
< Almost Eul	pcy	field 16v16	AlmortFull	fabric		
< Annoscrun		fiede 16x16	Ference	fabric	<b>V</b>	
Empty		nrodc_16x16_	Empty	rabric		
- Full		htodc_16x16_	Full	fabric	×	
RPReset		fifodc_32x16v	v_RPReset	fabric		
RdClock		fifodc_32x16v	v_RdClock	fabric		
RdEn				un-connected		
Reset	Cor	nnect	Reset	fabric	<b>v</b>	
WrClock	Disc	connect	WrClock	fabric	<b>v</b>	
WrEn	Der	auit Name	WrEn	fabric	<b>v</b>	
Data[15:0	]	fifodc_16x16_	Data[15:0]	fabric	<b>v</b>	
Q[15:0]		fifodc_16x16_	Q[15:0]	fabric	<b>v</b>	
▲ fifodc_32x16w					<b>v</b>	
AlmostEm	pty	fifodc_32x16v	v_AlmostEmpty	fabric	<b>v</b>	
AlmostFull		fifodc_32x16v	v_AlmostFull	fabric	<b>v</b>	
Empty		fifodc_32x16v	v_Empty	fabric	<b>v</b>	
- Full		fifodc_32x16v	v_Full	fabric	¥	
RPReset		fifodc_32x16v	v_RPReset	fabric	<b>v</b>	
RdClock		fifodc_32x16v	v_RdClock	fabric	<b>v</b>	
■ RdEn		fifodc_32x16v	v_RdEn	fabric	<b>V</b>	
Reset		fifodc_32x16v	v_Reset	fabric	<b>v</b>	
WrClock		fifodc_32x16v	v_WrClock	fabric	<b>v</b>	
WrEn		fifode 32v16v	w WrEn	fabric	4	

ウィンドウ下部の [Schematic タブをクリックすると、図 4-26 のようにコンポーネントレベルの I/O ポート図が表示されます。単一コンポーネントの場合は、このウィンドウは閲覧(確認)用のみです。

図 4-26. Builder → Schematic タブの表示例

Catalog Builder	Planner	1			
Design					
Image: Second	(32gear8				
Ports(12)					
Nets(12)		fifodo_tx32gear8_Data[47:0]	Data[47:0]		
		fifodc_tx32gear8_RPReset		AlmostEmpty fifodc_tx32gear8_AlmostEn	npty
		fifodc_tx32gear8_RdClock 🗁		AlmostFull fifodc_tx32gear8_AlmostFu	11
		fifodc_tx32gear8_RdEn 🗁	RdEn	Empty fifodo_tx32gear8_Empty	
		fifodc_tx32gear8_Reset 🗁	Reset	Full fifodo_tx32gear8_Full	
		fifodc_tx32gear8_WrClock 🗁	WrClock	Q[11:0] fifodo_tx32gear8_Q[11:0]	
		fifodo_tx32gear8_WrEn 🗁	WrEn		
Property			fifodo	tx32gear8	
General Conn	ections				
Components S	Schematic				
General Conn Components S	ections Schematic				

複数コンポーネントがある場合には、Schematic ウィンドウ上でドラッグ&ドロップによって接続指定が可能です。

入力ポート同士や入力・出力ポート間を接続する場合、まず [Components] タブで一方を "un-connected" 指 定します。次に [Schematic] タブを選択し、意図する一方のポートを選択し (クリックを離す)、接続先のポー トにカーソルを移動します。許可される接続の場合は配線が赤色となり、同時に (わかりにくいですが)緑 色 (水色)のチェックマークがポート近辺に表示されます。良ければ赤色に変わった配線の上で左クリック して確定します。接続情報が描画され直して表示されます。左クリックだけでは接続指定のアクションが継 続していますので、他とも接続する場合は繰り返します。終了する場合には、左クリックに続いてさらに右 クリックします (図 4-27 下)。

ポート接続の設定を完了したら(プランニングしない場合は)[Catalog] / [Builder] / [Planner] タブ表示の 上にある『Generate』ボタンをクリックしてモジュールを生成します。Generate しないとモジュールが生成 されないことにご留意ください。

## 図 4-27. Schematic ウィンドウで接続指定する例



正常に生成後は sbx が自動的に取りこまれ、ファイルリスト・ビューの "Input Files" セクションに表示されます(図 4-28)。

#### 図 4-28. モジュール生成後の Input Files セクション例



一度生成した後に、sbx ファイルのパラメータを変更する場合は、まず Input Files 内の当該 sbx 行をダブル クリックします。図 4-29 のようなウィンドウが表示されます。

変更したい当該コンポーネント行を選択して(本図では一つしかない)右クリックすると、三つのアクションが選択できます。[Reset]はパラメータを初期状態に戻し、[Config]を選択すると図4-22のような生成時にパラメータを設定した状態で Clarity Designer が立ち上がります。適宜変更・修正して再度『Generate』をクリックします。Generate しないと変更は有効になりません。

ここで留意事項があります。サブシステム・フォルダー下にある何らかのソースファイルを編集したり、 ユーザ作成のファイルを置いた場合で、"Input Files" セクションにインポートしている .sbx をダブルクリッ クして Clarity Designer を起動すると、その時点で(Generate していなくても)全てのファイル / サブフォル ダーが元の状態に上書きされます。必要なファイルは別の場所にバックアップしておくことを推奨します。

図 4-29. sbx ダブルクリック後、選択して右クリック時の GUI 表示例

Clarity	Designer(	C:/aUSER	_SStasks/D33wo	rk/ecp5pcie	e/e5p —	x I
File Vie	w Windo	w Help				
S DRC	🗟 Genera	te				
Catalog	Builder	Planner				
Co	omponent	s	Connection	Туре	Export	
fifodc	_tx32gea	r8				
				Reset		
				Delete		
				oning		
Compone	nts Sch	ematic				
-						

## 4.3.6 プランニング

[Planner] タブでは、特定のエレメントをドラッグ&ドロップ形式でターゲットとして選択しているチップの物理リソースに配置指定(割り当て)することができます。本機能には ECP5 の PCS/SERDES (DCU, Extref) と Generic DDR および DDR3/LPDDR3 メモリー・インターフェイスのみが対応しています。

前節までのようなデバイス・アーキテクチャ固有のマクロ(モジュール)生成ではプランニングは必須で はなく、ビルド作業後に "Generate" して問題ありません。他方、PCS ではプランニングしないと Generate で きません。『DRC』ボタンをクリックして確認できます。

詳細フローはそれぞれの例を扱う第4.3.8項、第4.3.9項、第4.3.10項で記述します。

Clarity Designer が生成するファイル sbx は、ターゲットとするデバイスサイズ (LUT 規模)、およびパッ ケージに依存する情報を含んでいます。サブシステム生成後、或いはプロジェクト (インプリメンテーショ ン)を再利用したり、変更する場合、或いは既存 sbx ファイルをインポートした場合などに、デバイスやパッ ケージが異なると、図 4-30 のようなメッセージが表示されます。再度 "Generate" し直す必要がありますの で、ご留意ください。

同様にインプリメンテーション(プロジェクト)で論理合成ツールの指定を変更した場合も、全てのサブシステム.sbxを再度 Generate する必要がありますので、ご留意ください。

図 4-30. デバイスサイズ (左) やパッケージ (右) を変えた場合のウォーニング例



## 4.3.7 ビルド / プランニング後のリソース情報

ビルド / プランニングを終え Generate すると、図 4-31 のように [Resource] タブのあるウィンドウ枠内 にハードウェア・リソースの使用・割り当て情報が表示されます。右上のアイコン 5 をクリックすると



ウィンドウがデタッチできますので、拡大表示して確認します。図 4-32 は "DDR\_Generic" サブシステムの 例です。

図 4-31. リソース表示ウィンドウ枠

Usage	G
	Glot 🔦
PLL	0/4
DDRDLL	1/4
CLKDIV	2/4 *
🎨 Resource	

#### 図 4-32. デタッチ状態のリソース表示ウィンドウ

Usage		-	(, pro. )	offer (30)	11,041.1	ы.		
	Global	Bank0	Bank1	Bank2	Bank3	Bank6	Bank7	Bank8
PLL	0/4							
DDRDLL	1/4							
CLKDIV	2/4							
ECLKBRIDGE	0/2							
Edge Clocks				1/2	0/2	0/2	1/2	
Number of IOs	20/245	0/32	0/40	10/32	0/48	0/48	10/32	0/13
Number of Differential Pins		0/16	0/20	5/16	0/24	0/24	5/16	0/6
TRUE LVDS Outputs				10/16	0/24	0/24	6/16	
VCCIO of the Bank				2.5			1.2,1.35,1.5,1.8,2.5,3.3	
DQS Groups				0/2	0/3	0/3	0/2	
PCLK Pins		0/2	0/2	0/2	0/2	0/2	1/2	
Dedicated PLL Input Pins		0/3	0/3	0/3	0/3	0/3	1/3	
Virtual VCCIO Pins		0/4	0/6	0/4	0/6	0/6	1/4	

## 4.3.8 プランニング例 1 ~ PCS (SERDES)

本節では SERDES サブシステムのプランニング(配置指定)を例に手順を既述します。サブシステムには 二つのモジュール、PCS と Extref(外部基準クロック入力マクロ)を含むものとします(一般的な構成)。

#### 図 4-33. PCS モジュールの生成開始

Catalog Builder Planner	9, 9, <u>0</u> ,	
Name	Version	PCS
<ul> <li>▲ Module</li> <li>▲ Architecture_Modules</li> <li>▷ ■ IO</li> <li>☞ dynamic bank controller</li> </ul>	1.4	PCS (Physical Coding Sub-layer) supports adv Ethemet, Fibre Channel, and PCI Express. Devices Supported: ECPSUM, LatticeECP2M, LatticeECP3, LatticeS Version:
a extref	1.1	Revision History:
叇 pcs	8.2	8.2: Added ECP5 support.
نغة pli A 🕾 Arithmetic Modules	5.6	8.1: Added Control Setup tab to control hie ge 8.0: Added Frequency attribute for LatticeECP

まず Clarity Designer を起動し、サブシステム名等の初期設定を行います(図 4-19 参照)。次に PCS モジュールを生成します。図 4-33 に示すように [Catalog] タブで "pcs" を選択します (ウィンドウ下部のタブ は [Lattice IP])。

ダブルクリックすると図 4-34 が立ち上がります。上部にタブが五つありますが、該当する所は全て設定 します。その中でも基本設定は [Instance Setup] タブの内容で、その上部を抜き出したものが図 4-35 です。



図 4-34. PCS コンフィグレーション  $\sim$  Instance Setup タブ



ここで対応プロトコルとチャネル数、送受信対応等を指定します。選択するプロトコルに従って、その他 タブの表示や選択できるパラメータ等が変わります。各パラメータの詳細説明は ECP5 のテクニカルノート TN1261 をご参照ください。

#### 図 4-35. PCS の基本設定 (Instance Setup の上部)



次に Extref マクロの設定を行います。図 4-36 のように [Catalog] タブで "extref" 行をダブルクリックしま す。本モジュールはパラメータが少ないため、詳細説明は割愛します。

#### 図 4-36. extref モジュールの生成開始

Catalog Builder Planner	କ୍ ବ୍ ପ୍ର		
Name  Module  Module  Modules  D  IO  Modules  D  Modu	Version	*	EXTREF Description: Interface for external reference clock. Devices Supported: ECP5UM Version:
寧 extref 碑 pcs	1.1		Revision History: 1.1: Added ECP5 support. 1.0: Original released version
硬 pll 4 l Arithmetic Modules	5.6		References TN1261 - ECP5 SERDES/PCS Usage Guide

両モジュール (コンポーネント) をコンフィグレーション後の [Builder] タブは図 4-37 のようになります (モジュールのポート名表示行を展開した後)。

第4.3.5 項で既述したとおり、各ポートをファブリックや外部ピンに引き出すかどうかの指定や、モジュール(コンポーネント)間の接続指定などを行います。例えば"extref"出力"refclko"ポートを"PCS"の"pll\_refclki"入力ポートと接続し、かつ外部に Export します。

🗷 4-37. PCS	Sとextrefをコン	フィグレーショ	ン後のサブシステム	(Builder タブ)
-------------	-------------	---------	-----------	--------------

Catalog Builder Planne	r		
Components	Connection	Туре	Export
extref			<b>V</b>
refclkn	extref_refclkn	pad-pin	$\checkmark$
refclko	extref_refclko	fabric	<b>V</b>
refclkp	extref_refclkp	pad-pin	$\checkmark$
▲ PCS1			
cyawstn_dual0		un-connected	
Ism_status_ch0_s		un-connected	
Ism_status_ch1_s		un-connected	
pll_lol	PCS1_pll_lol	fabric	<b>V</b>
pll_refclki		un-connected	
rsl_disable	PCS1_rsl_disable	fabric	<b>V</b>
rsl_rst	PCS1_rsl_rst	fabric	<b>V</b>
rst_dual_c	PCS1_rst_dual_c	fabric	<b>V</b>

次に [Planner] タブをクリックすると、初期表示は図 4-38 のようになります(各リソース名の行を展開して表示した後)。[Placement] カラムはまだブランクです。

図 4-38. PCS と extref をコンフィグレーション後のサブシステム (Planner タブ)



PCS の配置指定は、"Resource" 枠内のコンポーネント印のある [DCUCHANNEL] を選択し、ドラッグ&ド ロップして行います。図 4-39 はドラッグ中のスクリーンダンプです。ドラッグ先で"禁止マーク"が出ず、 リソース表示が図のように緑色に変化すれば問題ありません。



図 4-39. PCS (DCU) のドラッグによる配置指定



ドロップ後は図 4-40 のように物理リソースの色が青色になります。また、Placement カラムにはドロップ したリソースの物理位置情報が表示されます。

図 4-40. PCS (DCU) のドロップ後の表示例



次に同様に extref マクロもドラッグ&ドロップします。基本的に extref は同一 DCU にします。

特にマルチプロトコル対応のサブシステムの場合は、ここで DCU のクロック入力設定を確認します。配置した DCU をダブルクリックして "DCU Settings" ウィンドウを表示させます(図4-42)。プルダウン形式で TX PLL とレシーバの入力ソースを選択します。本例では単一チャネルですので、選択肢は二つしか表れません。

DCU を二つ使用する場合は、外部クロック入力の選択候補が増えて表示されます。



図 4-41. extref をドロップ後の表示例



図 4-42. DCU クロックソースの設定例



以上で各種構成指定・設定は終了です。最後に『Generate』ボタンをクリックしてサブシステムを生成後、 エラーがなければ終了します。エラーのある場合は解決して先に進みます。

なお、Clarity Designer で生成したサブシステム(モジュール、或いは"コンポーネント")に関わる設計 制約は、IPexpress の場合 LPF 制約ファイルにマニュアルで記載(転記)する必要がありました。Clarity Designer ではその必要は無く、物理制約ファイル PRF(<implementation>.prf)に反映される形式になってい ます(スプレッドシート・ビューで確認してもわかりません)。配置指定するプランニング情報とともに配置 配線エンジンにパラメータを直接渡すためです。タイミングに関わる制約については、レポートファイルで 確認することができます。

## 図 4-43. Generate 後、コンソールのメッセージ例

Output							
INFO -	C:/aDi	Work/Clarity	/0514/cl	test33r	edo/vhdl/s	ource/block1	L/PCS1/PCS1 rsl.v(70,1-694
INFO -	C:/lsco	c/diamond/3.	4 x64/i	spfpga/	userware/N	T/SYNTHESIS	HEADERS/ecp5um.v(1751,1-1
Done:	design l	load finishe	d with	(0) err	ors, and (	0) warnings	
Error*	Output	T cl Console	Warning	Info*	Find Results		

また、IPexpress では 〈PCS モジュール名 〉.txt という名称で PCS/SERDES 設定用の『自動コンフィグレー ションファイル』(Auto Configuration File) がテキスト形式で RTL と共に出力されていました。Clarity Designer では同様なファイルは存在しません。従来ユーザがこのファイルを編集してパラメータ変更等をすることも

## macnica

可能でしたが、Clarity Designer 対象デバイスではできなくなりました。

## 4.3.9 プランニング例 2 ~ DDR Generic インターフェイス

ここでは ECP5 用マクロ DDR Generic のプランニング例を示します(Crosslink ではプランニングに対応していません)。サブシステム名(<name>.sbx)設定は省略します(図 4-19 参照)。下部[Lattice IP] タブを選択した状態で、上部の[Catalog] タブを選択し、さらに "DDR Generic"を選択して(図 4-44) ダブルクリックすると、図 4-45 が立ち上がります。

## 図 4-44. DDR Generic モジュールの生成開始

Senerate Seferesh Catalog Builder Planner			
Name  A  Module  A  A  A  A  A  A  A  A  A  A  A  A  A	Version	•	DDR Generic Description: Allows you to generate an output DDR for a generic appli Devices Supported:
▲ ③ IO 母 ddr_generic	5.8		ECP5U, ECP5UM, LatticeEC, LatticeECP, LatticeECP2, Lat LatticeXP, LatticeXP2, MachXO2, MachXO3L, Platform Ma Version:
鐸 ddr_mem	5.8	11	5.8 Revision History:
每 gddr_7:1	5.8		5.8: Added support for IOL on ECP5 with uneven sides.
₩ sdr 5.8			5.7: Added MachXO3L MIPI support. ECP5 update.
Advantige dynamic bank controller 1.4			5.5: Added MachXO3L support.
母 extref	1.1	Ŧ	The DEL_ADJ and DEL_VAL attributes can no longer be a
Lattice IP 🚯 Lattice IP Server	Import IP Diamor	d	About

図 4-45. DDR\_Generic のコンフィグレーション ~ [Pre-Configuration] タブ

Generate Log			
DDR_GENERIC	Pre-Configuration Configuration		
	Interface Type	Receive	-
dont(7:0)	I/O Standard for this Interface	LVCM0S25	•
	Bus Width for this Interface	8	(1 - 256)
	Clock Frequency for this Interface	400	(100 - 400MHz)
sync_clk	Interface Bandwidth (Calculated)	6400	Mbits/s
	Clock to Data Relationship at the Pins	Edge-to-Edge	• •
			Next >
q[31:0] → datain[7:0]			

パラメータ設定(Configuration)用に、ウィンドウ上部に二つのタブがあります。通常は[Pre-configuration] での設定で完了します。データパス遅延が固定値(デフォルト)以外の必要があるなど、特定のケースでな ければ[Configuration]タブでの設定は不要です。詳細は ECP5の"High-Speed I/O"に関するテクニカルノー トをご参照ください。

Pre-configuration の基本設定としては図 4-46 のように、Tx か Rx か(註:基本的に [Receive MIPI] は選択 しないこと)、I/O インターフェイスタイプ、データレーン数、データレート、そしてクロックとデータの位 相関係などがあります。



図 4-46. DDR Generic 基本パラメータ



コンフィグレーション完了後、ウィンドウ左下の『Configure』ボタンをクリックします(本例でのコン ポーネント名は "dddr\_rx")。[Builder] タブをクリックしてコンポーネント名表示の行を展開するると、図 4-47 のようになります(ウィンドウ下部は[Component] タブ)。

図 4-47. DDR\_Generic コンフィグレーション後の ddr\_rx コンポーネント ([Builder] タブ)

🌯 DRC 🔒 Ge	enerate			
Catalog Built	der Planne	r		
Compo	nents	Connection	Туре	Export
⊿ ddr_rx				<b>V</b>
alignw	d	ddr_rx_alignwd	fabric	<b>V</b>
clkin		ddr_rx_clkin	pad-pin	V
ready		ddr_rx_ready	fabric	<b>V</b>
sclk		ddr_rx_sclk	fabric	<b>V</b>
sync_c	lk	ddr_rx_sync_clk	fabric	<b>V</b>
sync_r	eset	ddr_rx_sync_reset	fabric	<b>V</b>
update	e	ddr_rx_update	fabric	<b>V</b>
🛛 🖒 📕 datain	[7:0]	ddr_rx_datain[7:0]	pad-pin	$\checkmark$
b a dcntl[2]	7:0]	ddr_rx_dcntl[7:0]	fabric	
> 🗧 q[31:0	)]	ddr_rx_q[31:0]	fabric	
Components	Schematic			

次に [Planner] タブをクリックします。Resource 枠最上行の"ddr\_rx"を展開して下位を表示させ、eclk\_group0 をドラッグ&ドロップして配置指定します。図 4-48 はバンク 2 にドラッグ中の様子です。ここでは同バン ク下部の領域にカーソルを移動していますが、該当ピンとエッジクロック・ネットワーク(ECLK)とクロッ ク分周器(CLKDIV)などの所用リソースが同じくピンク色に変わり、配置可能なことを示します。ピンの 選択は自動で判断されます。

この状態でドロップする(マウスを離す)と、色が青に変わり、指定が受け付けられたことを示します。 同時に、左枠 "Placement" カラムに割り当て結果としての物理位置情報が表示されます。



図 4-48. DDR\_Generic (eclk\_group) のドラッグによるバンク指定 ([Planner] タブ)



なお、コンポーネントやドラッグ先によって、付随して使用されるリソース位置が異なるのは勿論ですが、 場合によっては、例えば1本で済むべきクロックネットを2本使用する結果になったりしますので、ドロッ プ先は注意深くします。

図 4-49. DDR\_Generic (eclk\_group) のドロップ後の表示例



この状態ではまだクロック関連の指定が行われていません。Clarity Designer の考え方として、クロック入 力は必ずマニュアルで別途指定することが必要です。そこで [clkin] ポートを選択し、先ほどと同じバンク2 の下部ピン近辺にドラッグします。配置可能なピンがあれば、付随して使用される DDRDLL や遅延素子 DLLDEL などのクロック関連リソースと共にピンク色に変わります(図 4-49)ので、ドロップします。配置 されたリソースが青色になり(図 4-50)、[Placement] カラムが配置後のリソース情報で満たされます。

次に Generate する前に、[Catalog] タブ表記の上部にある『DRC』 ボタンをクリックしてルールチェック を実行します。問題がなければプランニングは終了です。例えば上記 "clkin" の指定をする前に DRC を実行 すると、指定が必要であることを示すメッセージがコンソールに表示されます。

最後に『Generate』すると、"ddr\_rx.sbx" が "Input Files" 部に取りこまれていることが分かります。その後は Diamond フローに戻ります。



#### 図 4-50. クロック入力ポート (clkin) の配置指定例



## 4.3.10 プランニング例 3 ~ DDR3 メモリー・インターフェイス

ここではラティスの DDR3 メモリー・インターフェイス IP を使用したプランニング例(ECP5)を示しま す。DDR3 IP のコンフィグレーションは終了しているものとします。

図 4-51. "eclk\_group0"のドラッグによる配置先を探す



ラティスの IP の場合、二つの上位リソース "eclk\_group0" および "sclk\_group1" が必要な構成になっています(自動生成)。プランニングはそれぞれについて行います。

まず "eclk\_group0" をドラッグして配置先を探します。図 4-51 のように、有効なロケーションがあると パッドや DLL など所用リソースの色がピンク色に変わります。ドラッグ先(のポート位置)を変えるとそ れらも変化しますので、所望の所でドロップします。ここの例はメモリーとのデータバス幅が 64 ビットの 場合のため、かなりのリソースを消費します。データポートは反時計回りに順にポートが割り当てられます。 従って配置先バンクの版時計回りで先頭のポートにドラッグ先を移動すると最もバンクの使用効率的には 最良になります。

ドロップ指定するリソースの順序や何らかの理由で、すでに指定済みの位置を取り消す場合、図 4-52 のようにリソース名を右クリックして "Reset" を選択します。

#### 図 4-52. 配置指定済みのリソースを取り消し



もう一つのリソース "sclk\_group1" についても同様にドラッグ&ドロップ指定します。

次に、二つの上位リソース "eclk\_group0" および "sclk\_group1" をプランニングしたので『DRC』をクリッ クしてみます。その結果(図 4-53)、"clkin" ポートが未指定である旨のメッセージが表示されました(コン ポーネント "ddr3ip64b400" 内インターフェイス "eclk\_group0" のリソース "clk\_in" が配置指定されていない)。

## 図 4-53. 未指定リソースがある場合のメッセージ例



前節の例で記述したように、Clarity Designer の考え方として、クロック入力は必ずマニュアルで別途指定 することが必要です。Resource カラムで "eclk\_group0" を展開し、リソース "clk\_in" をドラッグします。ド ロップ可能なパッドにドラッグすると、図 4-54 のように関連リソースがピンク色になります。所望の位置 でドロップします。なお、"clk\_in" は差動入力ですが反転入力 "~clk\_in" をドラッグせず、必ず非反転側信号 を扱います。



図 4-54. "clk\_in"の候補先ポートを探す



以上で DRC がパスしたら、『Generate』して Diamond フローに戻ります。

---- \*\*\* ----