

Lattice Diamond 1.1 Startup Manual rev1.1

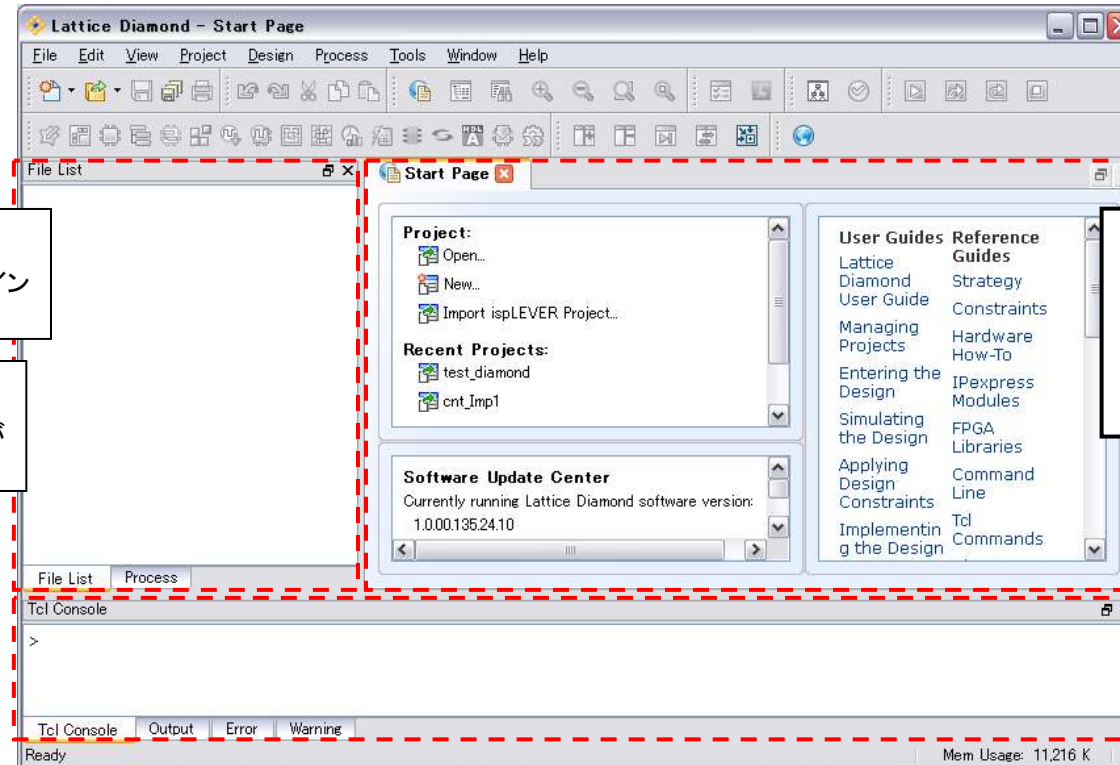


- 本マニュアルはFPGAデバイス設計時のDiamondのオペレーションフローマニュアルです。
- 新規プロジェクトの作成からデバイスへのフィッティングまでの一連のフローを解説したものです。
- 各項目の詳細については、別途Latticeのユーザーマニュアルもしくは弊社技術サポートまでお問い合わせください。



- スタートメニューより、プログラム＞Lattice Diamond 1.x＞Lattice Diamondを起動させます。
メインウィンドウは大きく分けて3つのウィンドウで構成されます。各ウィンドウの機能は以下の通りです。

Diamondメインウィンドウ



File Listウィンドウ

選択したデバイス、設計デザインファイルがリストされます。

Processウィンドウ

コンパイル等の実行コマンドがリストされます。

Start Pageウィンドウ

プロジェクトの新規作成、既存プロジェクトのOpenが可能です。Diamondのバージョン、マニュアル等を閲覧することも可能です。

Tcl Consoleウィンドウ

実行したコマンドが表示されます。直接コマンドを記述し、コンパイル等を実行することも可能です。

Outputウィンドウ

実行した結果のログが表示されます。

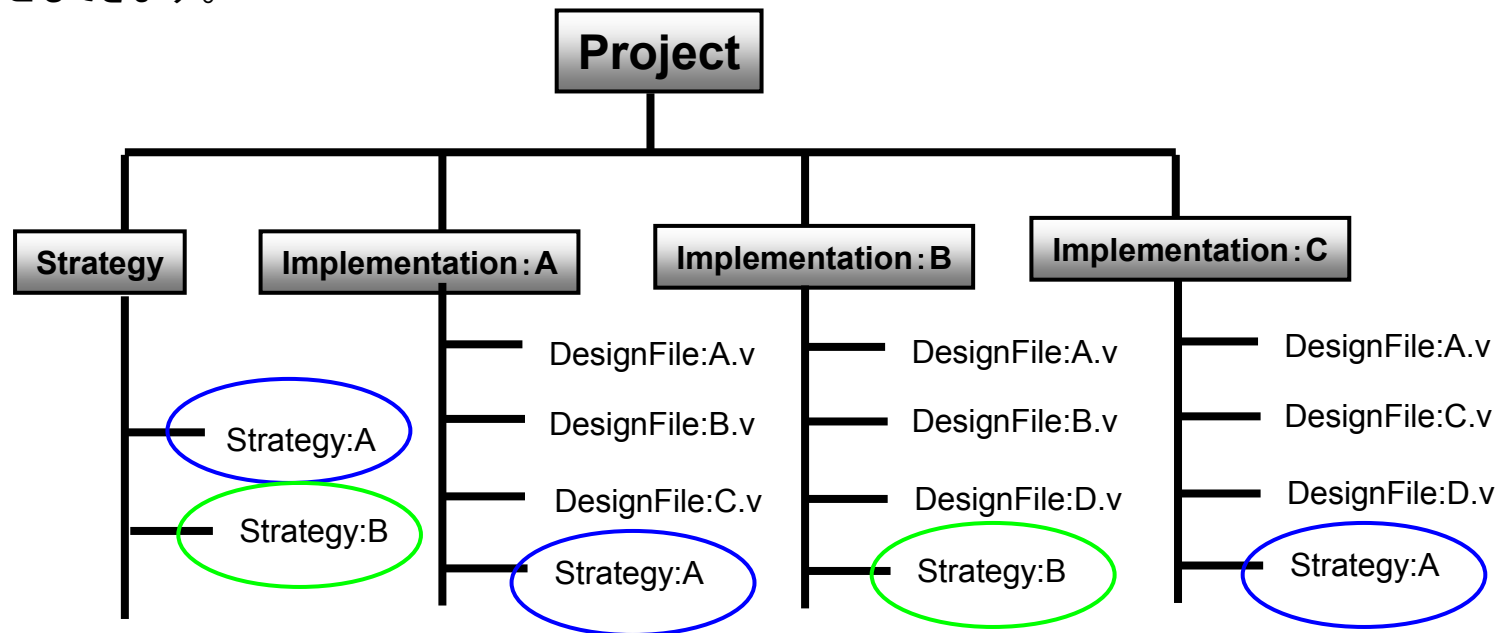
Errorウィンドウ

Outputウィンドウのエラーログのみが抽出され、表示されます。

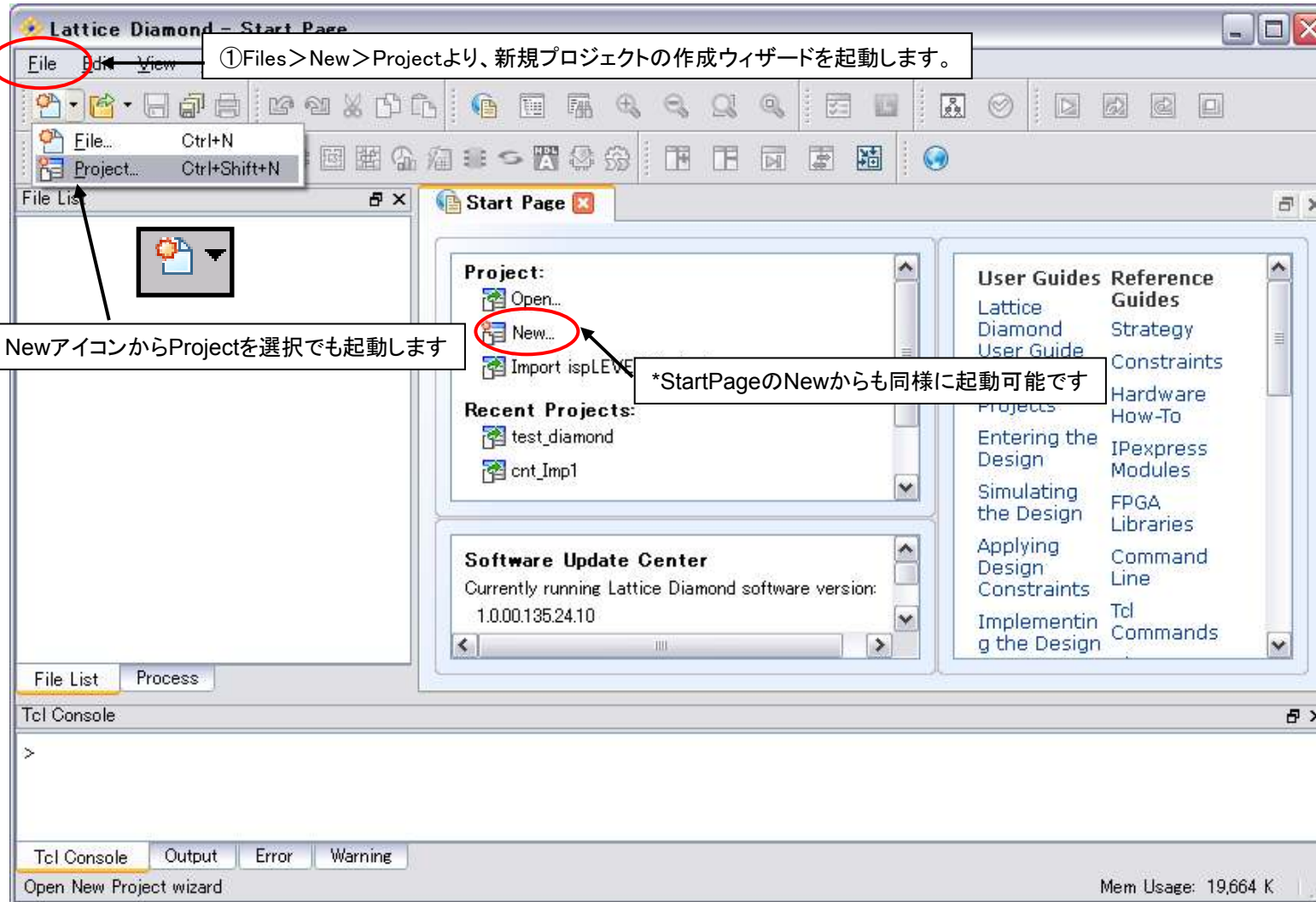
Warningウィンドウ

Outputウィンドウのワーニングログのみが抽出され、表示されます。

■ Lattice Diamondはユーザー様で作成したデザインProject単位で管理を行います。LatticeDiamondからはProjectの下に複数のサブProjectを構成でき、サブProjectごとに違う設計データを持たせてデザインの管理を行うことができます。このサブProjectはImplementationと呼びます。Implementation毎にStrategy(ツールのProperty、コンパイルオプション)や制約を持たせることができ、全てのImplementationを並列にコンパイルさせ、それぞれのコンパイル結果を比較することもできます。

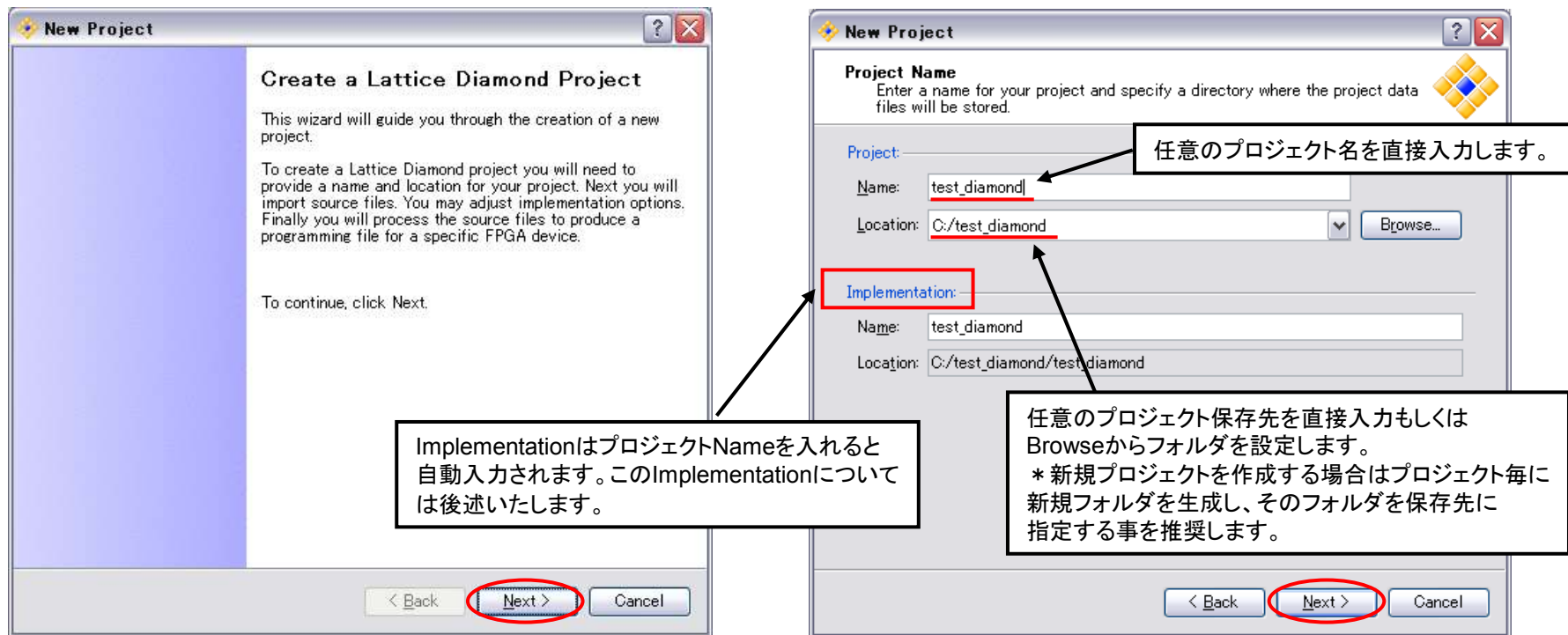


- Diamondで設計の際はProjectの作成から始めます。Projectで設計ソース、設計制約ファイル、その他各種設定情報等を一元管理いたします。



②New Projectウィンドウが開いたらNextをクリックして下さい。

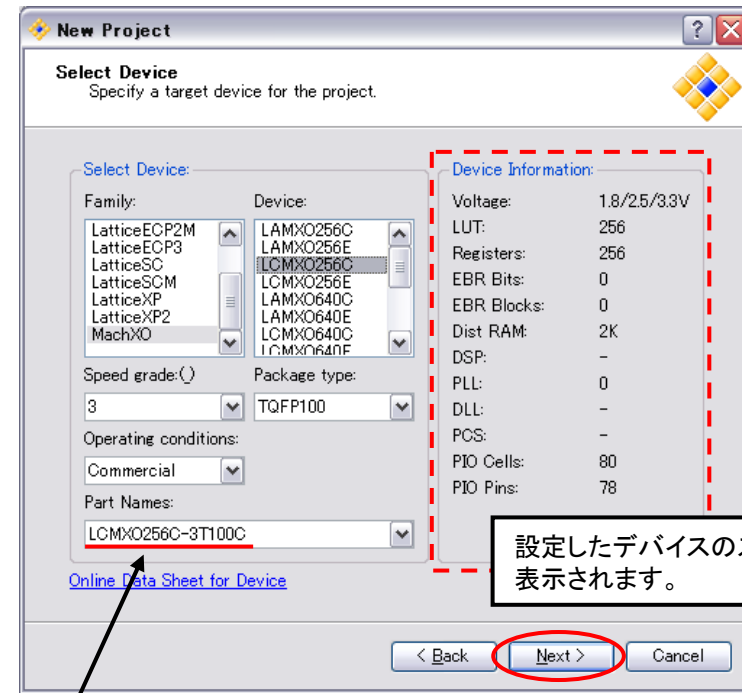
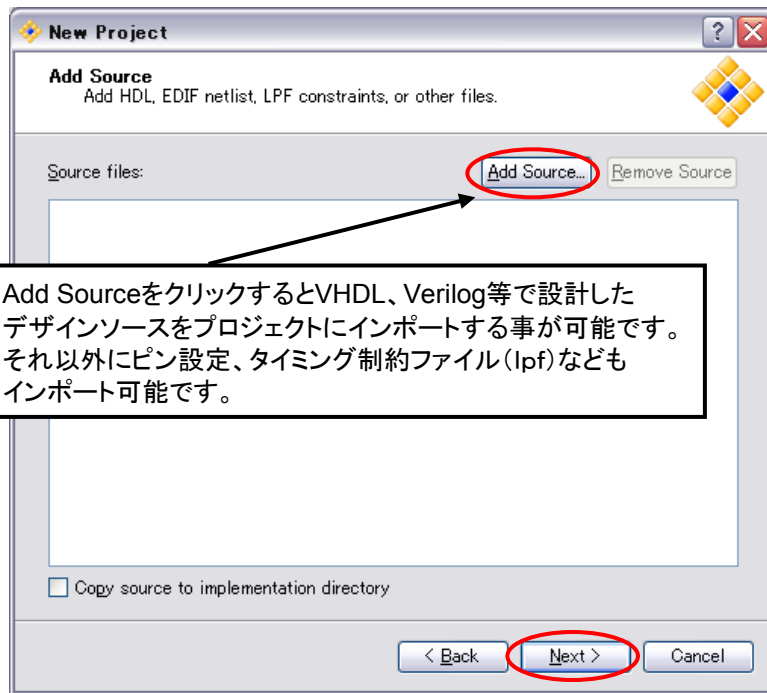
③次にプロジェクト名と保存先を設定し、Nextをクリックします。



*** 注意**

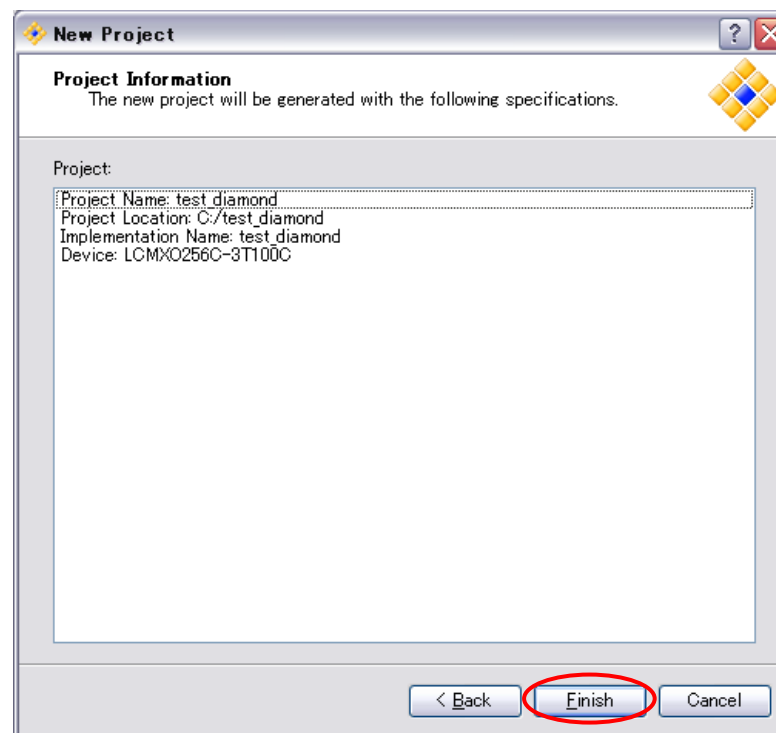
- ・プロジェクト名は英字、数字、_ (アンダースコア)のみを使用するようにして下さい。特に日本語フォント(全角)は絶対に使用しないで下さい。
- ・プロジェクトの保存先に日本語フォントが入らないようにして下さい。(デスクトップなど)

- ④既存のデザインソースがある場合は、プロジェクト作成時にインポートする事が可能です。(Add Sourceより)
既存のデザインソースが無い場合は何もせずにNextをクリックして下さい。
- ⑤次に設計するデバイスを選択します。Familyよりデバイスファミリーを選択し、Deviceから詳細のデバイスを選択します。Speed gradeでデバイスのスピードグレード、Package typeよりパッケージ、Operating conditionsより温度グレードを設定し、Nextをクリックして下さい。

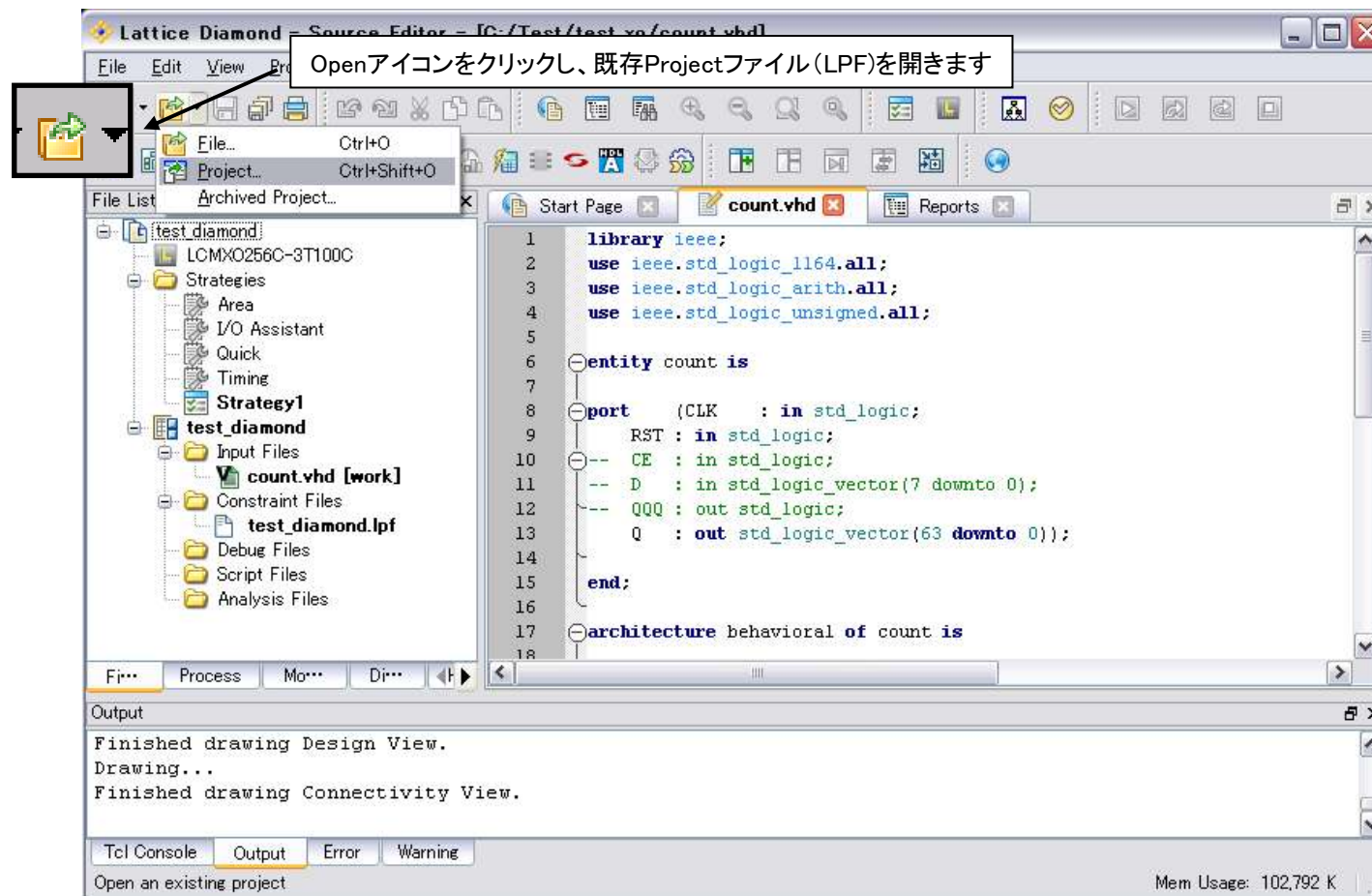


設定したデバイスの型式が表示されます。

- ⑥New Projectウインドウで設定した内容が表示されます。間違いが無いか確認下さい。
設定を変更する場合はBackをクリックし、変更箇所のウインドウまで戻って下さい。変更が無い場合はそのままFinishをクリックします。これで新規プロジェクトの作成は完了です。



- 既存のプロジェクトをオープンする場合はFiles>Open>Projectより既存のプロジェクトファイル(.lpf)を指定します。DiamondのOpenアイコンよりProjectを選択、もしくはStart PageのProject:Openをクリックする事でも同様に既存のプロジェクトをオープンする事が可能です。



①Files>New>Fileより、新規デザインファイルの作成ウィザードを起動します。

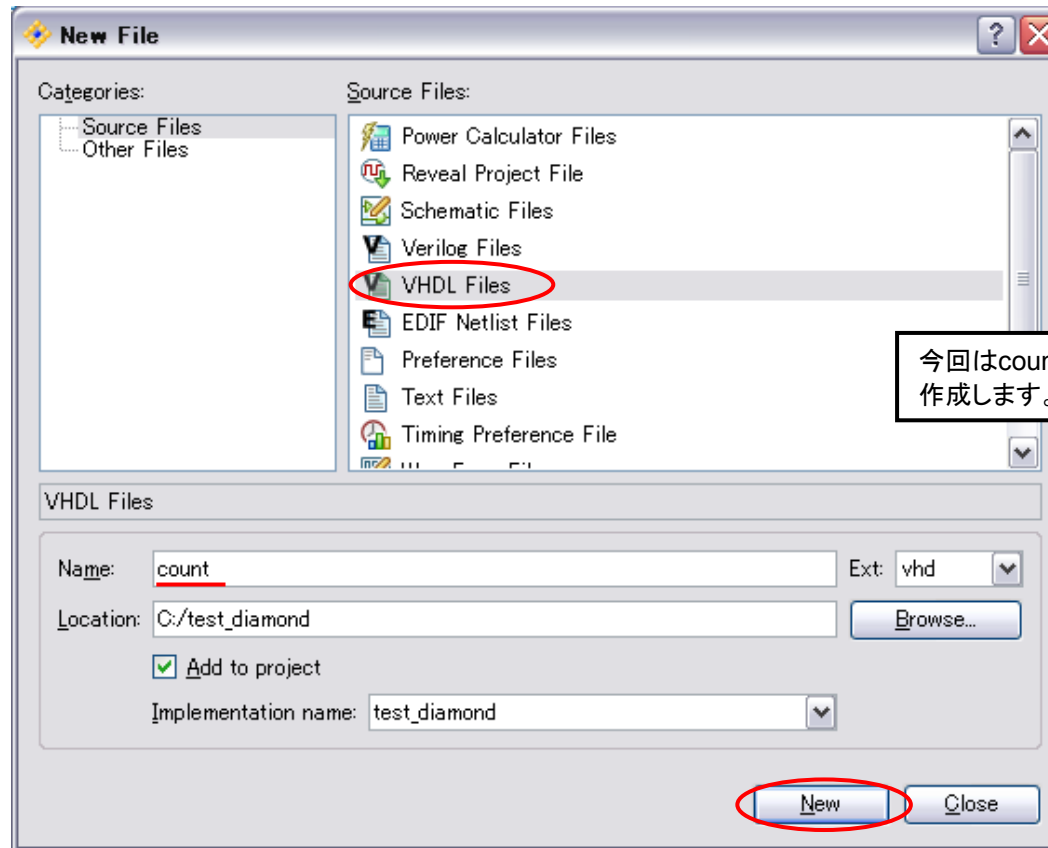
DiamondのNewアイコンよりFileを選択する事でも同様に新規デザインファイルの作成ウィザードを起動する事が可能です。

test_diamond project summary			
Module Name:	test_diamond	Synthesis:	SynplifyPro
Implementation Name:	test_diamond	Strategy Name:	Strategy1
Last Process:		State:	
Target Device:	LCMX0256C-3T100C	Device Family:	MachX0
Device Type:	LCMX0256C	Package Type:	TQFP100
Speed grade:	3	Operating conditions:	COM
Logic preference:	test_diamond.lpf		

```

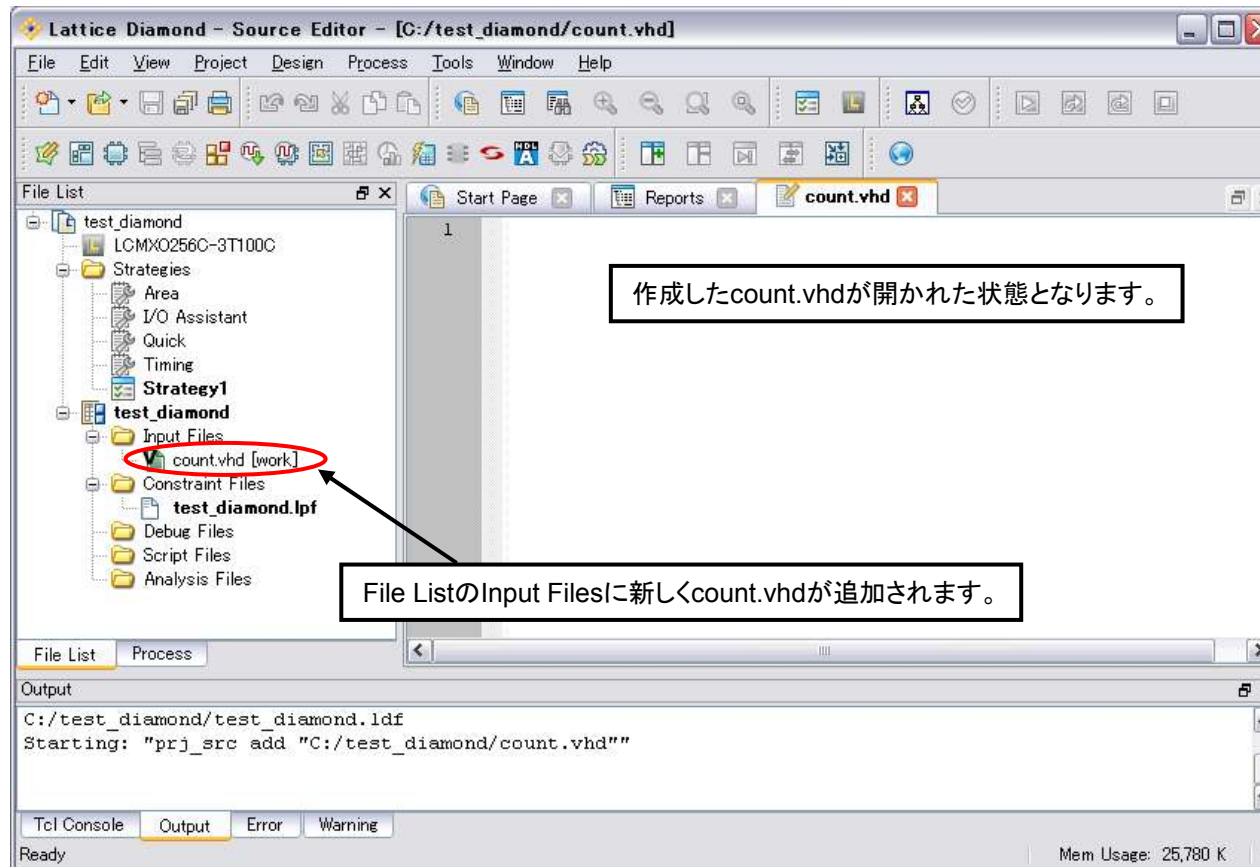
Tcl Console
> prj_project new -name "test_diamond" -impl "test_diamond" -dev LCMX0256C-3T100C
C:/test_diamond/test_diamond.lpf
>
  
```

- ②New Fileウインドウが開いたら、新規作成したいファイルの種類を選択します。
Name項にファイル名を任意の名前で直接入力し、Newをクリックして下さい。



今回はcountという名前のVHDLファイルを新規作成します。

③プロジェクトに新規で作成したデザインファイルが追加されます。



- ④新規作成したデザインファイルにデザインを設計します。
設計が完了したらSAVEもしくはSAVE ALLのアイコンをクリックして下さい。

SAVEは右ウインドウで現在開いているファイルのみをSaveします。
SAVE ALLは右ウインドウのファイルすべて(タグに表示されるすべてのファイル)をSaveします。

ファイルを編集するとファイル名の最後に * 印が表示されます。
Saveすると * 印は消えるため、複数のファイルを開いた際に、どのファイルを編集したかをタグで判別することが出来ます。

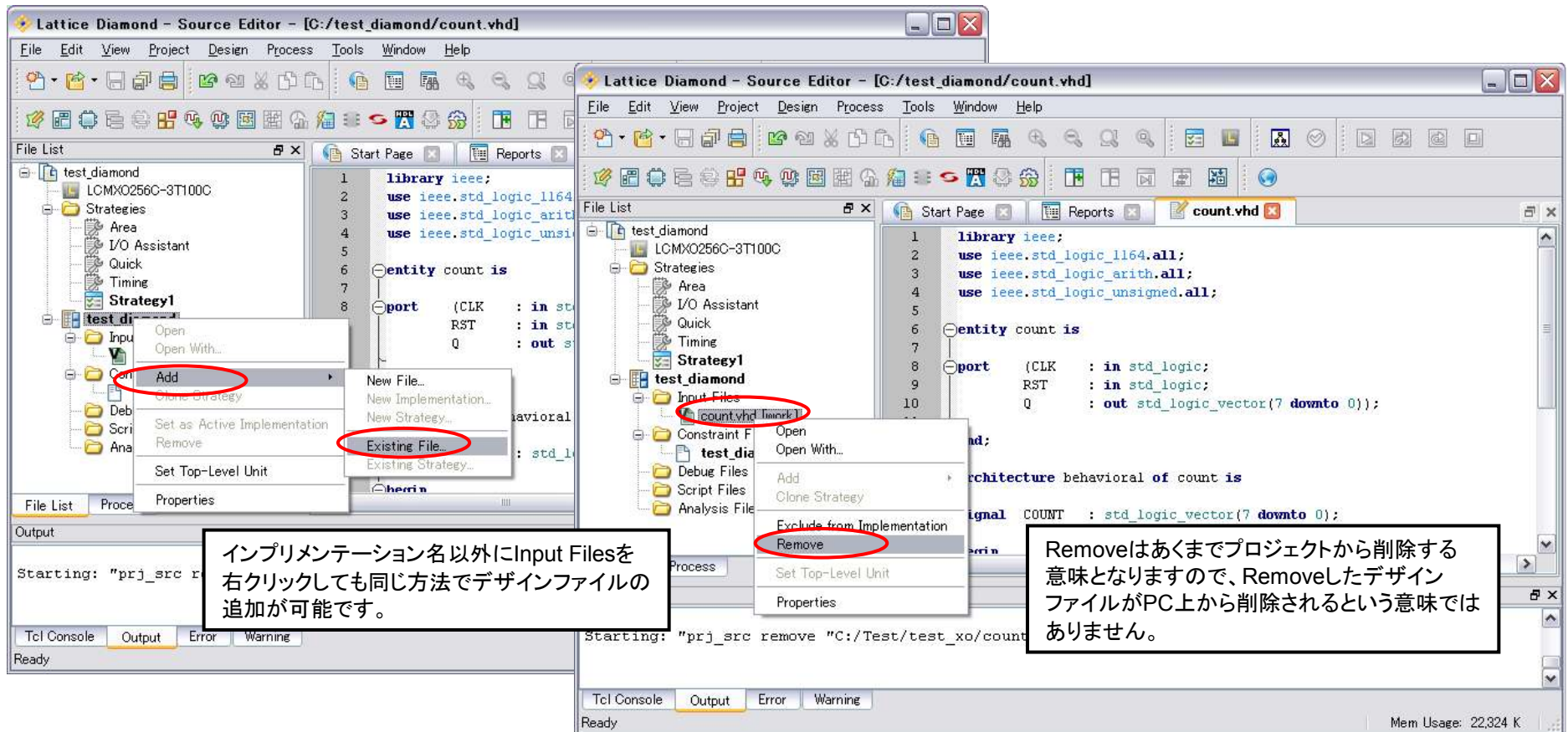
デザインを直接入力で設計します。
予約語はエディタ上で青や水色で色分けされます。
エンティティ部、アーキテクチャ部はブロック毎に表示/非表示を設定が可能です。

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  use ieee.std_logic_unsigned.all;
5
6  entity count is
7
8  port (CLK      : in std_logic;
9       RST      : in std_logic;
10      Q        : out std_logic_vector(7 downto 0));
11
12  end;
13
14  architecture behavioral of count is
15
16  signal COUNT  : std_logic_vector(7 downto 0);
17
18  begin

```

- 既存デザインファイルをプロジェクトに追加する場合は、インプリメンテーション名をクリックした状態で右クリック Add>Existing Fileより追加するデザインファイルを選択します。
デザインファイルをプロジェクトから削除する場合は、削除したいデザインファイルをクリックした状態で右クリック Removeより削除が可能です。



- 作成したデザインを再度確認するにはFile List Windowを開きます。このFile List Windowには作成したデザインの他のプロジェクトフォルダ内の各種ファイルが表示されます。

Device: 現在選択されているデバイスが表示されます。変更する場合はデバイス名をダブルクリック致します。

Strategies: Strategyが複数ある場合はそれぞれが表示されます。Strategyを変更する場合にはActiveにしたいStrategyを右クリック⇒「Set As Active Strategy」を選択します

DesignFiles: 設計ソースが格納されます。DiamondはMixed言語対応ですので、VHDL, Verilog, Edifが混在している場合には複数のファイルが表示されます

Constraint File: 各種制約情報が記載されているファイル.lpfが表示されます。

DebugFile: 各種デバッグファイルが格納されます。

Implimentation: Implimentationが複数存在する時に表示されます。Implimentationを変更する場合にはActiveにしたいImplimentationを右クリック⇒「Set As Active Implimentation」を選択します。

■ ispLEVERでは、ソースの改造構造がDefaultで表示されるGUIでしたが、Diamondでは階層構造の表示はされません。階層構造の解析を行う場合にはGenerate Hierarchyを使用します。

①Generate Hierarchyボタンをクリック。ツールが自動的にソースを解析し、階層構造を表示します

Hierarchyウインドウ

DesignView

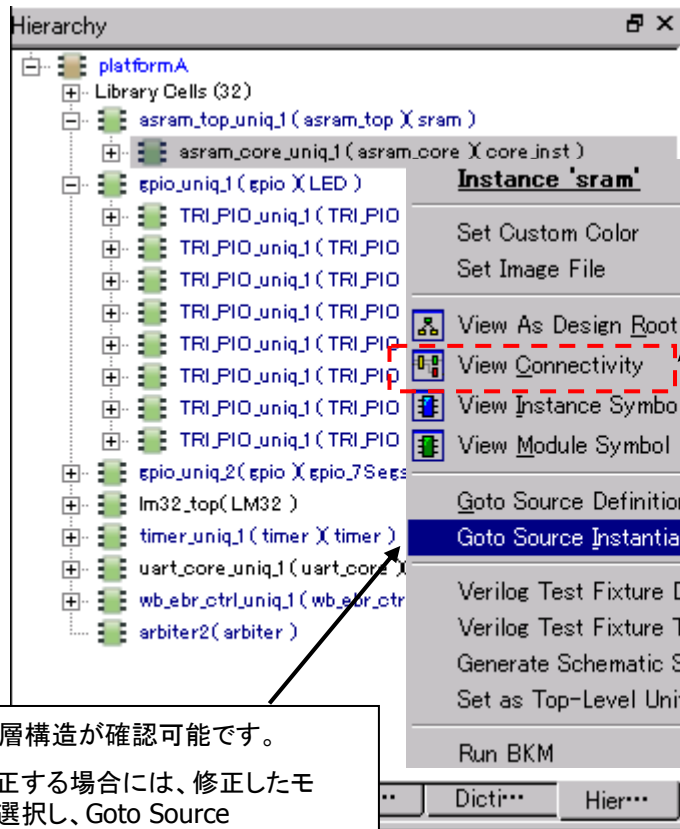
The screenshot shows the Lattice Diamond HDL Diagram interface. A red dashed box highlights the 'Generate Hierarchy' button in the toolbar and the 'Hierarchy' window. The 'Hierarchy' window on the left lists components like 'platformA', 'Library Cells (32)', 'arbitrator2(arbitrator)', 'asram_top_uniq_1(asram_top Xsram)', 'epio_uniq_1(epio XLED)', 'epio_uniq_2(epio Xepio_7Segs)', 'lm32_top(LM32)', 'timer_uniq_1(timer Xtimer)', 'uart_core_uniq_1(uart_core Xuart)', and 'wb_ebr_ctrl_uniq_1(wb_ebr_ctrl Xebr)'. The 'DesignView' on the right shows a hierarchical tree structure of the design, with levels 1 through 5 indicated on the right side. The root node is 'platformA', which branches into various sub-modules like 'cpu', 'icache', and 'memories'.

ispLEVERのように自動的に階層構造を表示させたい場合にはプルダウンメニューのDesign⇒AutoGenerateHierarchyを選択します。Projectの起動時、ソースをProjectにインポート時等で自動的にGenerate Hierarchyが起動します。

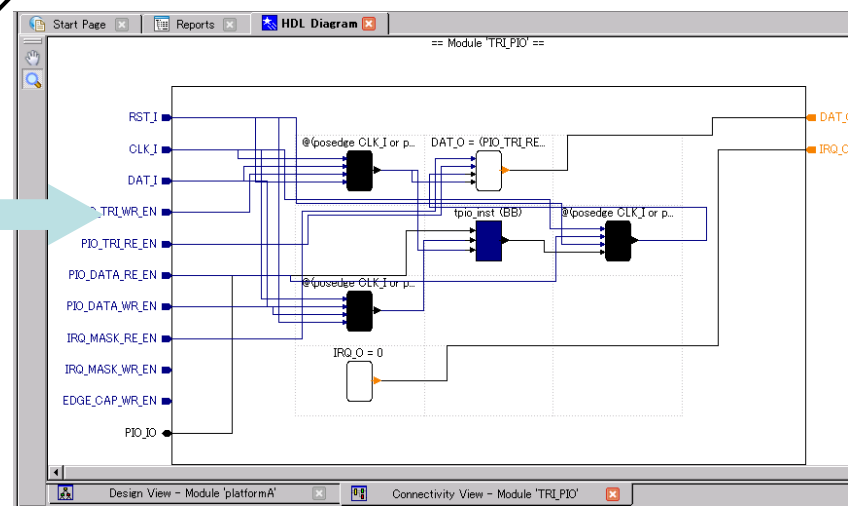
Design

- Generate Hierarchy
- Run BKM Check
- Reset Documentation Attributes
- Auto Generate Hierarchy
- Auto Run BKM Check

■ Generate Hierarchyで各種情報が確認可能です。

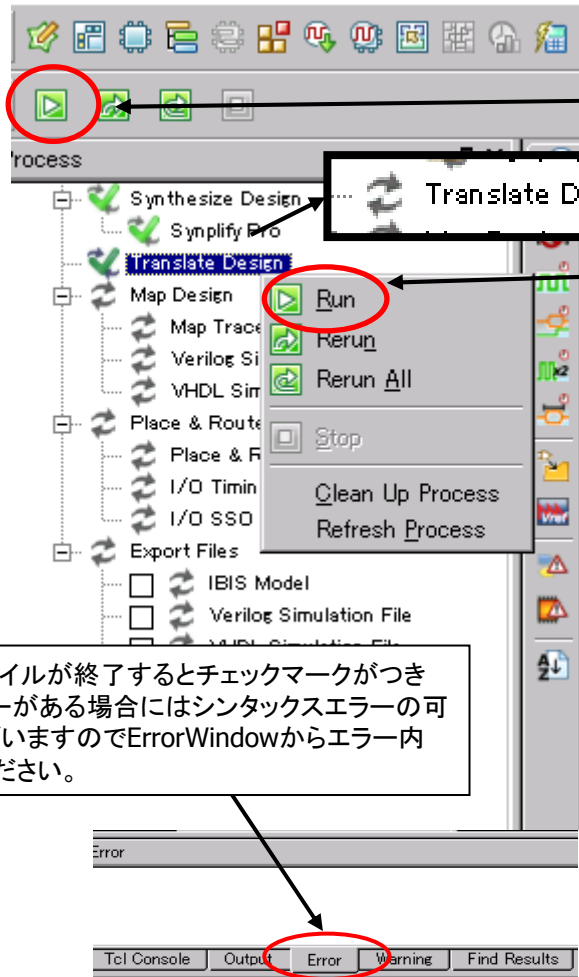


View Connectivityを選択してシンボル同士
の接続を回路図で確認することも可能です。



ソースの階層構造が確認可能です。
ソースを修正する場合には、修正したモジュールを選択し、Goto Source Instantiationを選択します。





- デザインを作成し終えた後、デバイスに対してピン配置の制約、タイミングの制約を行います。これら制約をする為にはSpreadSheetViewというツールを使用します。起動、実行させる為には、一度論理合成を行い論理合成後のEDIF(ネットリスト)ファイルからLattice独自のネットリスト形式に変換する必要があります



実行には

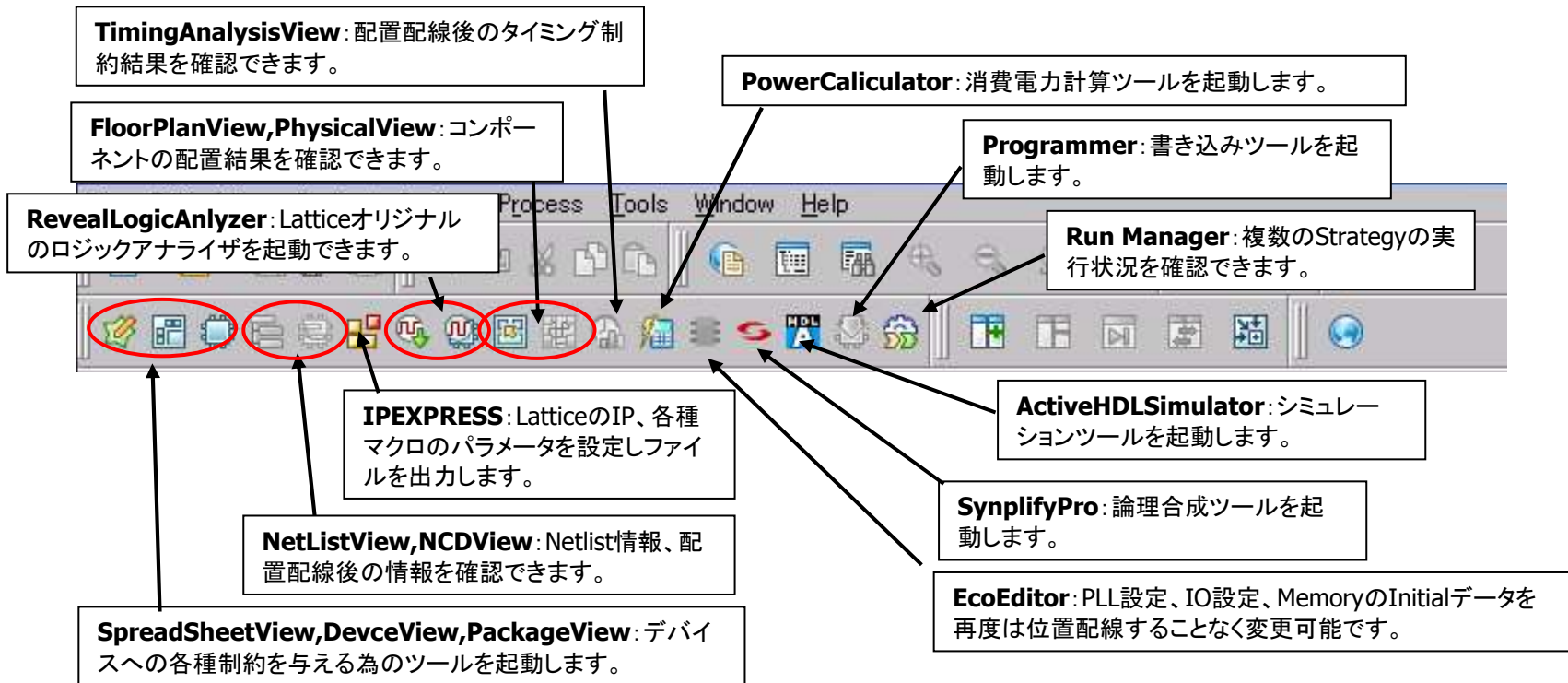
- ① TranslateDesignをダブルクリック
 - ② TranslateDesignをクリックし上部RUNボタン
 - ③ TranslateDesignを右クリック、Window内からRunを選択
- で可能です

■ チェックマーク

-  現在実行中のコマンドを示します
-  正常にコンパイルは終了しました。
-  ワーニングがあります。
ログをWarningウインドウに切り替え、ワーニングを確認して下さい。
-  エラーがあります。
ログをErrorウインドウに切り替え、エラーを確認し、修正して下さい。

無事コンパイルが終了するとチェックマークがつきます。エラーがある場合にはシンタックスエラーの可能性がございますのでErrorWindowからエラー内容を参照ください。

■ TranslateDesignを実行すると上部ツールバー上の“SpreadSheetView”ボタンを押すことができます。Diamondで使用可能な各種ツールはこちらのツールバーに集約されており、起動させることが可能です。



グレーで表示されていて実行できないツールについては、現在のProcess(Synthesis, Map, Place&Route)では実行できない(必要とならない)ツールになります。**Processが進んでいくとカラー表示になり、実行することが可能です。**

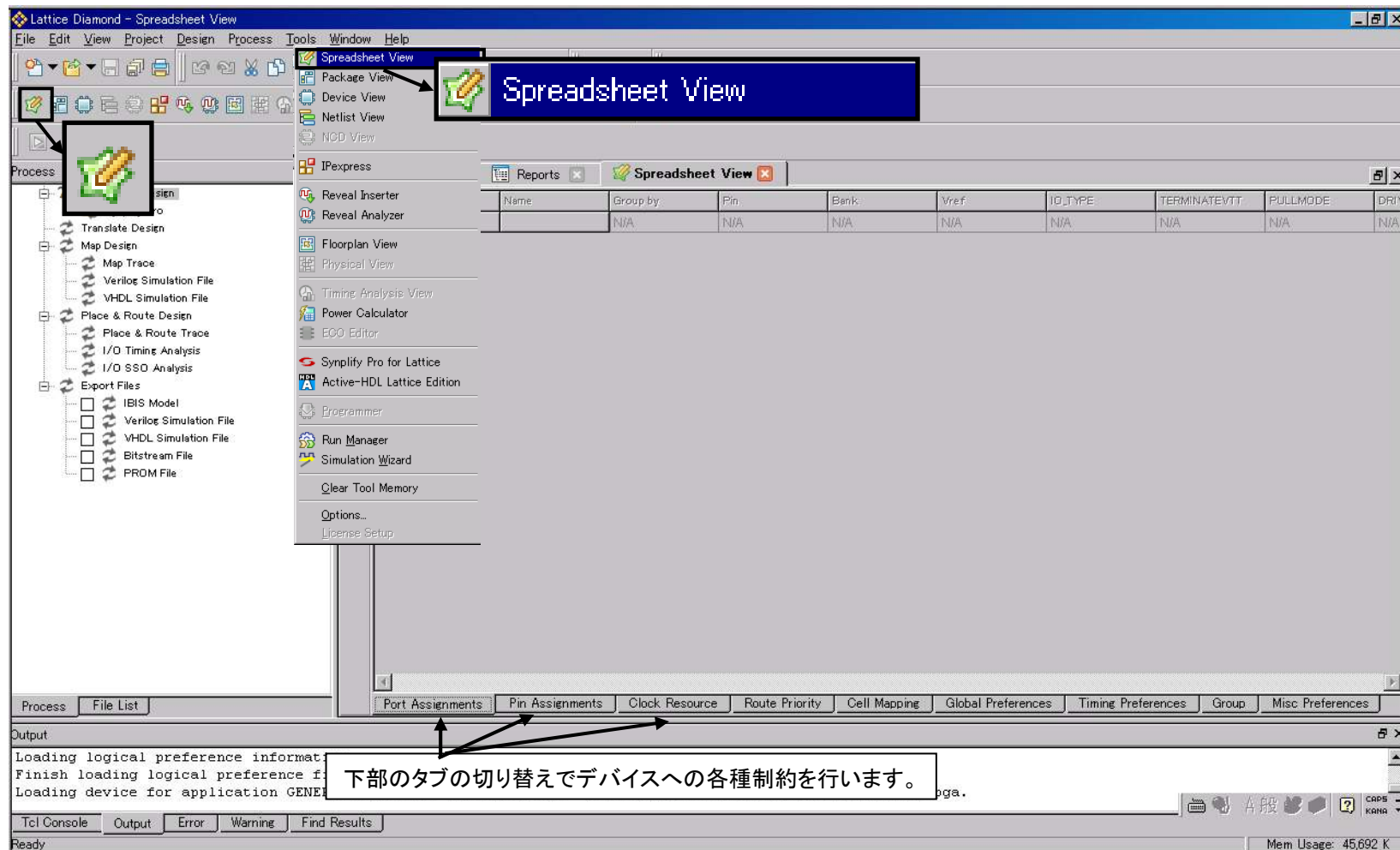


実行可能ツール



実行不可能ツール

- Tools > SpreadSheet Viewをクリック、もしくはSpreadSheet Viewアイコンをクリックし、SpreadSheet Viewを開きます。このツールを使用して、ピン配置、IO設定、クロックソース、タイミング等の各種制約を付加することが可能です。下部のタブを切り替えてデバイスの各種制約を行います。それぞれのタブの詳細を次ページより説明します。



■ Pin Assignment画面

信号の配置ができます。

PinAssignmentタブを選択してピンに対して信号をアサインしていきます。

SignalNameセルをダブルクリックでAssignmentSignalsダイアログが開きます。左のような画面が開きますので信号のアサインを行います。

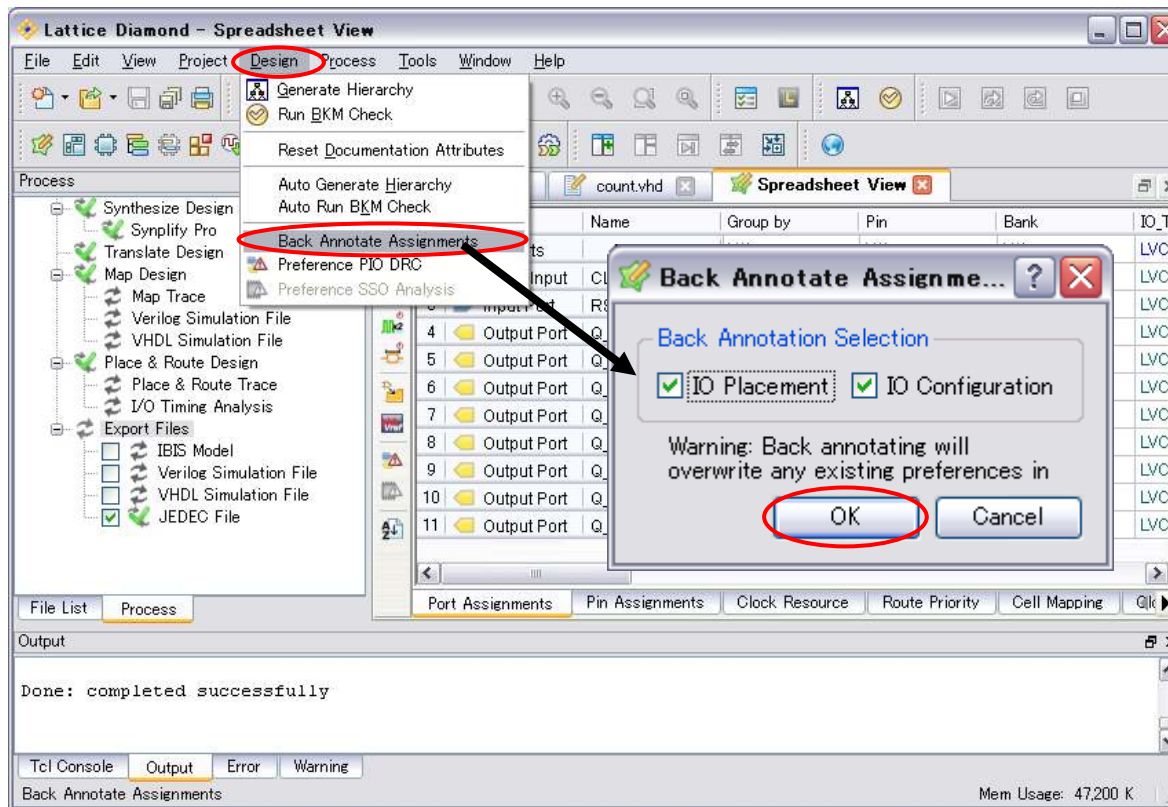
下部のタブによりPinAssignmntを選択します

Pin	Pad Name	Bank	Polarity	Signal	Type	Signal Name	IOTYPE
A15	FIO:PT53B	8	N		Output Port	EMPTY	LVCNMOS25
					Output Port	FULL	LVCNMOS25
					Output Port	HOLDN	LVCNMOS25
					Input Port	SPMADR_0	LVCNMOS25
					Input Port	SPMADR_1	LVCNMOS25
					Input Port	SPMADR_2	LVCNMOS25
					Input Port	SPMADR_3	LVCNMOS25
					Clock Input	SPMCLK	LVCNMOS25
					Input Port	SPMOSN	LVCNMOS25
					Input Port	SPMDI_0	LVCNMOS25
					Input Port	SPMDI_1	LVCNMOS25
					Input Port	SPMDI_10	LVCNMOS25

(参考情報)Back Annotationについて

Place & Route Designまで実行した後にSpreadSheet Viewを開くと下図のように開発ツールが自動で割り付けたピン配置が表示されます。(自動で割り付けたピン配置は()で表示されます。)

開発ツールが自動で割り付けた仮のピン配置を今後使い続けたい場合には、制約ファイルに反映させる必要があります。反映させる場合はDesign>Back Annotation Assignmentsより、IO PlacementとIO Configurationにチェックを入れてOKをクリックする事で本設定に変更する事が可能です。(この作業を行わないと次にPlace&Routeした際にピン配置が変わります)



■ Clock Resource画面

クロックへの制約を与えることが可能です。

クロック信号に対して、各種制約を与えます(タイミングの制約はClockResource画面ではできません)

	Clock Type	Clock Name	Selection	Quadrant	DCS/Pure
1	Clock Net	HOLDN_c		N/A	N/A
2	Clock Net	SPMCLK_c			
3	CE/LSR Net	SPMRSTN_c			

設計中でクロック信号として使用されている信号の一覧が表示されます。

制約を与えたい信号のSelectionセルを右クリックしその信号をデバイス内のクロックリソースのどこに割り当てたいのか制約を与えます。Prohibitxxxは逆に制約を避けたい場合に使用します。

- Primary、Secondary→プライマリ、セカンダリクロックにアサインしたい場合
- Prohibit Primary,Secondary→プライマリ、セカンダリにアサインしたくない場合
- ProhibitBoth→クロックリソースのどれにもアサインしたくない場合

下部のタブによりClock Resourceを選択します

■ タイミング制約画面

各種タイミング制約を与えることが可能です

Type	Name	Group by	Pin
All Ports			
Input Port	SPMRST		
Clock Input	SPMCLK		
Input Port	SPMCSN	N/A	
Input Port	SPMWEN		
Input Port	SPMREN		
Input Port	SPMADR_1	N/A	
Input Port	SPMADR_2	N/A	
Input Port	SPMADR_3		
Input Port	SPMADR_4		
Input Port	SPMDI_15	N/A	
Input Port	SPMDI_14	N/A	
Input Port	SPMDI_13		
Input Port	SPMDI_12		
Input Port	SPMDI_11		
Input Port	SPMDI_10		
Input Port	SPMDI_9		
Input Port	SPMDI_8		
Input Port	SPMDI_7	N/A	
Input Port	SPMDI_6	N/A	
Input Port	SPMDI_5	N/A	
Input Port	SPMDI_4	N/A	

■ タイミング制約～PERIOD/FREQUENCY Preference

選択したクロックに対して最大動作周波数の設定を行います。

時間(周期)か周波数で制約をかけるかを選択

内部の配線からのクロックか入力ポートからのクロックかを選択

クロックの選択

制約周波数(もしくは時間)を入力

HoldTimeのマーヅンを入力

OKをクリックすると設定が表示される

■ タイミング制約～ BLOCK Preference

タイミング解析しない信号の選択をします。ここで選択した信号に繋がるパスは全てタイミング解析上無視されます。

ブロック(無視)する信号名選択

ブロックする対象をネットかグループで選択

OKをクリックすると設定が表示される

■タイミング制約～ INPUT_SETUP/CLOCK_TO_OUT Preference

クロックに対して選択したポートのセットアップ/ホールドタイムとクロック入力から出力までの許容できる遅延値の設定を行います。

制約の選択

- ・クロックに対するセットアップとホールドタイム
- ・クロック入力からの出力遅延

制約をかける対象の選択

- ・全てのポートを選択
- ・個別のポートを選択
- ・グループの選択(グループ化した場合)

対処となるポートの選択

クロックの選択

制約時間を入力

OKをクリックすると設定が表示される

■タイミング制約～ MULTICYCLE Preference

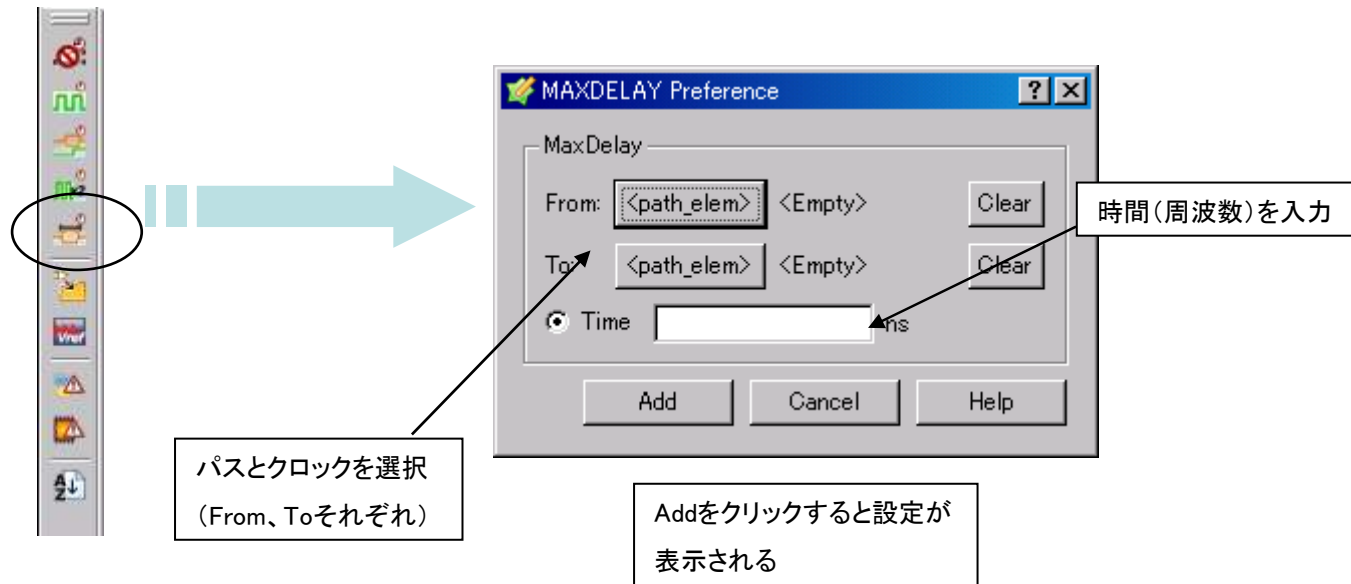
MULTICYCLE (General、Clock Net to Clock Net、Slow/Fast Path Exception)によって選択した非同期パスへ遅延を与えます。MAXDELAYによって選択したパスに許容できる最大遅延の設定を行います。

The image shows the MULTICYCLE Preference dialog box with several callouts explaining its components:

- 非同期クロック間に対する遅延設定**: Points to the "Clock Net to Clock Net" radio button option.
- 異なるグループ間に対する遅延設定**: Points to the "Slow/Fast Path Exception" radio button option.
- 一般的なパス(レジスタ間やピン間)に対する遅延設定**: Points to the "General" radio button option.
- パスとクロックを選択 (From、Toそれぞれ)**: Points to the "From:" and "To:" fields, which include dropdown menus for path elements and "Clear" buttons.
- 時間(周波数)を入力**: Points to the input field for the delay value, which is currently set to "ns".
- OKをクリックすると設定が表示される**: Points to the "Add" button at the bottom of the dialog.

■タイミング制約～ MAX Delay Preference

MAXDELAYによって選択したパスに許容できる最大遅延の設定を行います。



■タイミング制約～ 設定済みの制約の確認

Spreadsheet Viewの“Timing Preferences”タブから設定済みのタイミング制約を一覧できます。

修正がある場合、修正箇所をクリックで修正が可能です。

Preference Name	Preference Value	Preference Unit
BLOCK		
FREQUENCY		
PORT "SEMCLK"		
Frequency	200.000000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
PERIOD		
INPUT_SETUP		
ALLPORTS CL...		
Time	10.000000	ns
Hold Time	0.000000	ns
Clock Offset	0.000000	
PLL Phase...	OFF	
Input Delay	not Input Delay	
CLOCK_TO_OUT		
ALLPORTS CL...		
Time	10.000000	ns
Min Time	0.000000	ns
Clock Offset	0.000000	
PLL Phase...	OFF	
Output Delay	not Output Delay	
MULTICYCLE		
MAXDELAY		

(参考情報)

Spread Sheet Viewで設定した各種制約は.LPFファイルに情報が格納されます。こちらのファイルを直接編集して設定することも可能です。

The screenshot displays the Lattice IDE interface. On the left, the 'File List' pane shows a project tree with 'mult1' selected. Under 'Constraint Files', the file 'mult1/source/multi.lpf' is highlighted. On the right, the 'Spreadsheet View' shows a table of constraints for the 'SPMCLK' pin, including frequency, I/O type, and timing parameters like setup and hold times. A callout box points to the 'multi.lpf' file in the File List, stating: 'File Listにタブを切り替えて"ConstraintFiles"の中の.lpfファイルを選択' (Switch to the 'File List' tab and select the .lpf file in 'ConstraintFiles'). Another callout box points to the constraint table, stating: 'ピンの配置情報、デバイスへのタイミング制約情報等が格納されており、直接編集可能' (Pin configuration information, timing constraints to the device, etc. are stored here and can be edited directly).

- ①File ListタブをProcessタブに切り替えるとコンパイルの実行コマンドリストを表示します。
* 各コマンドの詳細は以下をご参照ください。

The screenshot shows the 'Process' tab in a design tool. The 'Process' tab is selected and highlighted in yellow. The list of steps includes: Synthesize Design, Synplify Pro, Translate Design, Map Design, Map Trace, Verilog Simulation File, VHDL Simulation File, Place & Route Design, Place & Route Trace, I/O Timing Analysis, Export Files, IBIS Model, Verilog Simulation File, VHDL Simulation File, and JEDEC File. Five callout boxes provide detailed descriptions for Synthesize Design, Translate Design, Map Design, Place & Route Design, and Export Files.

Processタブをクリックしコマンドリストを表示させます
Processタブが表示されていない場合は、View>
Toolbars>Processにチェックを入れて下さい。

Synthesize Design
論理合成ツールSynplify Proを使用し、デザインファイルをゲートレベルのEDIFネットリストに最適化します。

Translate Design
論理合成ツールから吐き出されたEDIFを受け取りLattice独自のInternal Data Baseを作成します。
Lattice Module,ispLEVERCore IP,Userで作成したMacro等が存在した場合、Lattice独自の形に変換を行います。

Map Design
EDIFネットリストを使用しデバイス独自のComponent (PFU,PLC,PIO.SLIC)へと変換を行います。
Mapping時に配置情報、Region情報、グループ情報等のAttributeが存在していた場合には、この情報を含めてMappingを行い、*_map.ncdファイルを生成します。

Place & Route Design
Mapping情報を元にデバイスに対して配置、配線を行い、ncdファイルを生成します。
Preference file中にタイミング制約があった場合にはそれを満たすように配置、配線を行います。

Export Files
チェックマークを付けたファイルを生成します。
IBISシミュレーション用のモデルを生成する場合はIBIS Modelにチェックを、タイミングシミュレーションの遅延ファイルを生成する場合は使用言語のSimulation Fileにチェックを、書き込み用データ(.jed .bit)を生成する場合はJEDEC Fileにチェックを入れます。

②コンパイルの実行は

その1. 各コマンドをダブルクリック、或いは各コマンドを右クリックし、Runで実行します。

その2. コマンドを一度クリックしハイライトさせた後、上部アイコンのRUNボタンを押します。

各コマンドは1つずつ手動で実行する必要は無いため、JEDECファイルを生成する場合は、JEDEC Fileチェックボックスにチェックを入れてExport Filesのみを実行します。するとSynthesize Designから順にコンパイルコマンドが自動で実行されます。コンパイル後は各コマンドにコンパイル結果のチェックマークが表示されます。

実行方法その1
コマンドをダブルクリック
OR
「右クリック」→「RUN」

実行方法その2
コマンドをハイライト後、上部
RUNボタンで実行





ワーニング、エラー発生時はログウインドウを切り替えて内容を確認

File List | Process | Output
Running BKM Check
BKM Check finished with (0) error

Tcl Console | Output | **Error** | Warning

Ready
Diamond1.x_startup.ppt Page: 34

■チェックマーク

-  現在実行中のコマンドを示します
-  正常にコンパイルは終了しました。
-  ワーニングがあります。
ログをWarningウインドウに切り替え、ワーニングを確認して下さい。
-  エラーがあります。
ログをErrorウインドウに切り替え、エラーを確認し、修正して下さい。



* エラー発生時にエラーログをダブルクリックするとデザインのエラー発生箇所にジャンプします。

■コンパイル後の結果を確認するにはレポートビューを使用します。レポートビューはDiamondメイン画面
上部のアイコンから起動します。

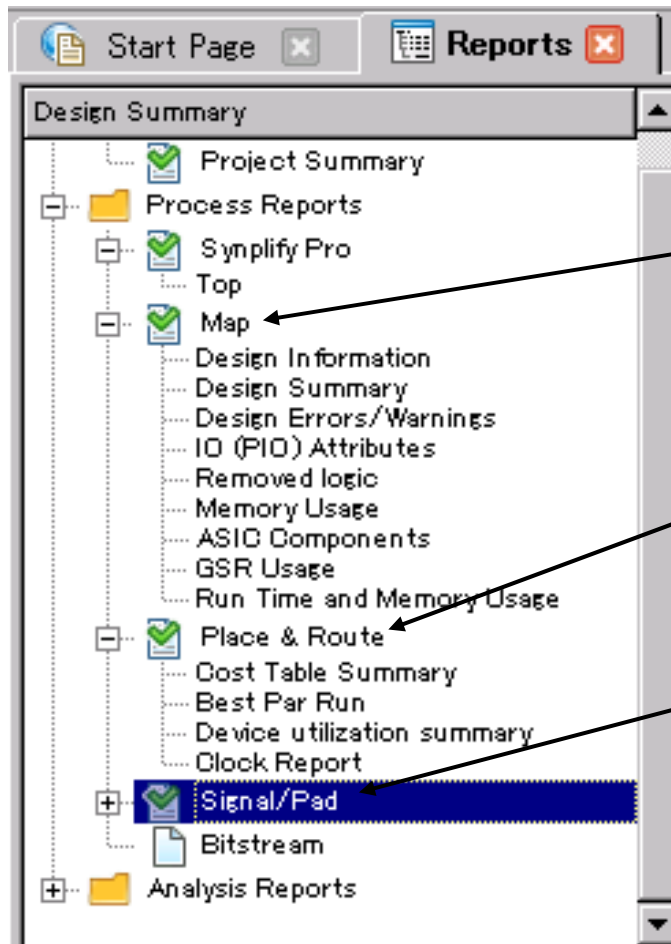


Synthesis and Ngdbuild Report
 #Build: Synplify Pro for Lattice D-2009.12L-1, Build 040R, Jan 20 2010
 #install: C:\lsc\lsc\diamond\1.0\synpbase
 #
Process Report:
 論理合成結果、マッピング結果、配置配線結果、ピン配置結果、書き込みファイル生成結果が確認できます
 #Tue Oct 05 10:31:04 2010

Analysis Report:
 Map後のタイミング解析結果、配置配線後のタイミング解析結果、Setup/Hold,Tcoのタイミング解析結果、SSO解析結果が確認できます

-  確認することができるレポートが存在することを示します。
-  確認することができるレポートが存在していないことを示します。コンパイルを実施すると確認できます。

■ Process Reportでは、マッピング結果、配置配線結果、IO配置結果について確認可能です



Map:
作成したデザインをFPGA内部のLUT, Slice, PFU, Memory, PLL等にマッピングされた結果を示します
このレポートにより、デザインの規模、内部の専用メモリブロック、DSPブロック、PLL等のリソースの使用数、使用率が確認することが可能です

Place&Route:
配置配線実施時のコンパイル結果が確認可能です。
内部配線リソースの使用率、クロックリソースの使用率等が確認できます。

Signal/Pad:
ピン配置結果に関して確認することが可能です。

■ Analysis ReportsではSpreadSheetViewで設定したタイミング制約に対する結果が確認可能です

各種Traceレポートを確認する際は
ProcessWindowからそれぞれのProcessごとの
Trace名をダブルクリックしてください。

Map Trace:
マッピング後のタイミング制約に対する結果を確認
できます。あくまでも配線をしておりませんので配線
については見積もり値を使用します。* 見積もり値
はツール上で設定可能です

Place&Route Trace:
配置配線実施後のタイミング制約に対する結果を
確認できます。Setup側の解析結果、Hold側の解
析結果を確認できます。

IO Timing Report:
IOのSetup/Hold時間、Tco時間を確認できます。
Place&Route Traceでも確認できますが、デバイ
スのMinimum遅延に-M品を含んでおりません。よっ
てIOのタイミング結果はこちらのレポートを参照く
ださい。

* -M品とは設計上の理論値としてデバイスがとりえるであろう
最小遅延値になります。

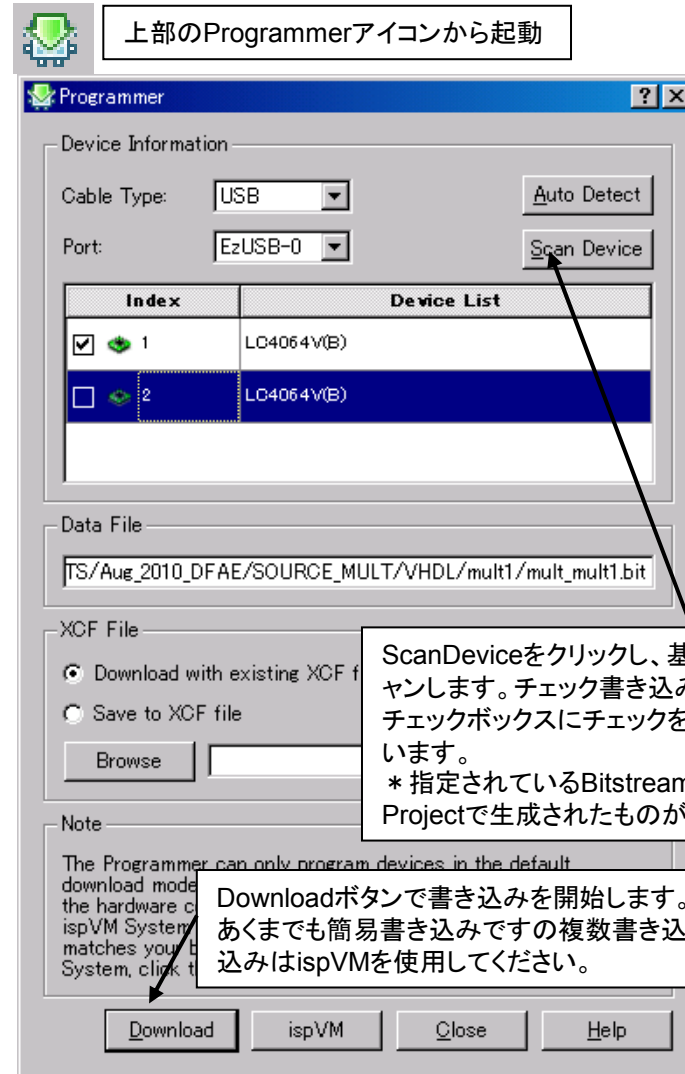
■書き込みファイルを生成する際には、ProcessWindowから生成を行います。生成後、別ツールのispVMsystem 或いはDiamondにバンドルされているProgrammerを使用して簡易書き込みが可能です。



Bitstream Fileのチェックボックスにチェックを入れます。以下の方法でBitsreamデータを生成します。

- ①Bitstream Fileの項目をダブルクリック
- ②Bistream Fileを選択し、RUNボタンをクリック
- ③BistreamFileを選択し、右クリックからRUNを選択

Processウインドウを選択



上部のProgrammerアイコンから起動

ScanDeviceをクリックし、基板上のDeviceをスキャンします。チェック書き込みたいデバイスのチェックボックスにチェックを入れ、書き込みを行います。
* 指定されているBitstreamファイルは現在のProjectで生成されたものが指定されます。

Downloadボタンで書き込みを開始します。Programmerはあくまでも簡易書き込みですので複数書き込み、Flash書き込みはispVMを使用してください。

■ ispLEVERでは、テストベンチファイルの雛形を作成する機能がございました。Diamondでも同様にテストベンチの雛形を作成することが可能です。生成の際にはGenerate Hierarchyを起動します。

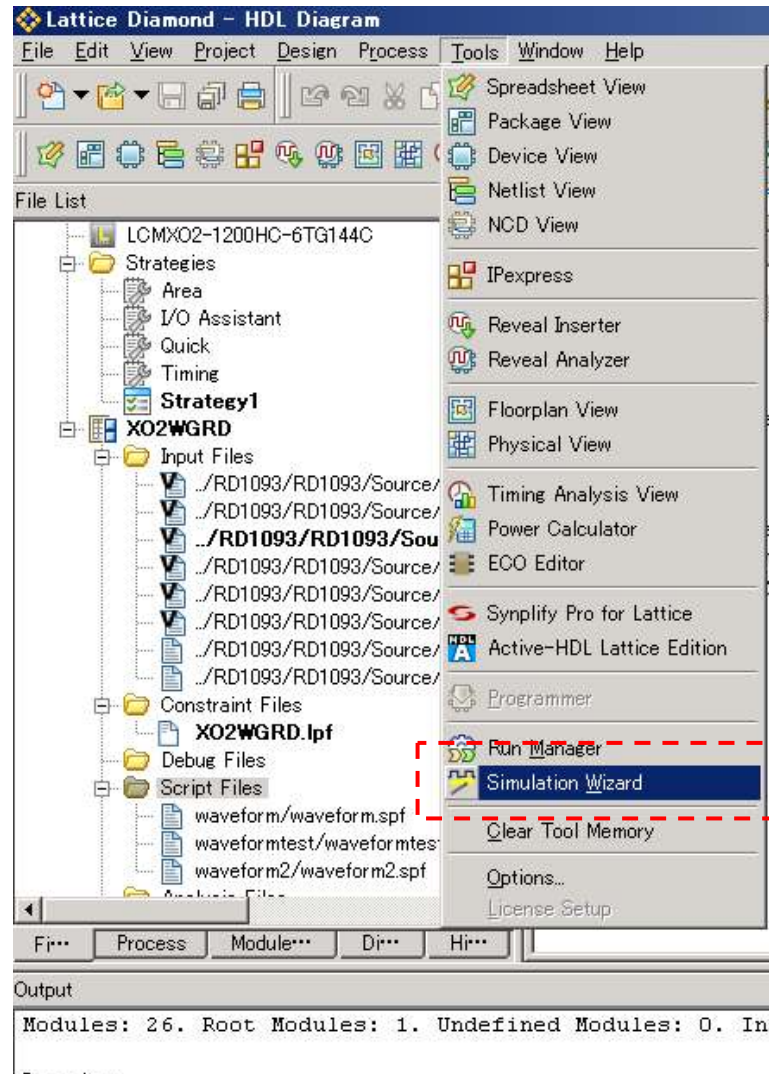
①Generate Hierarchyを起動

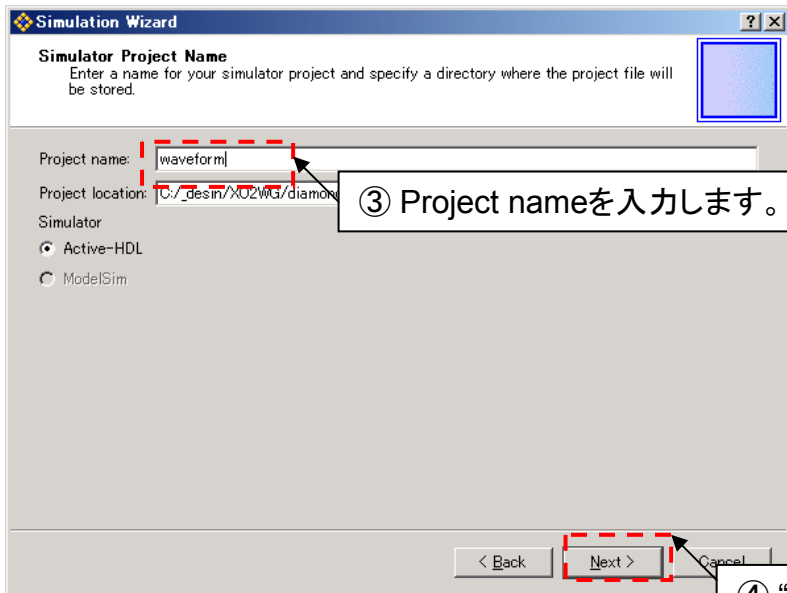
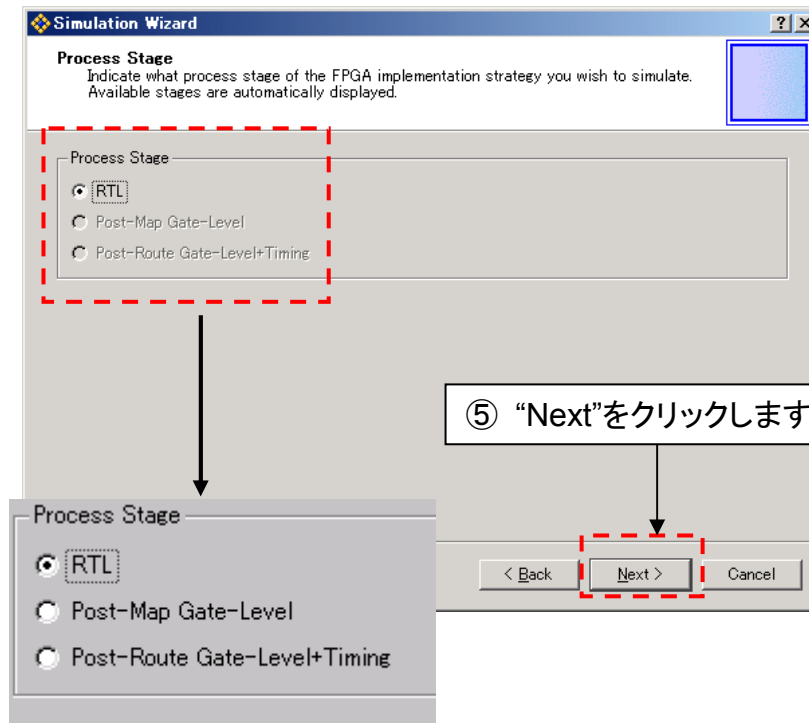
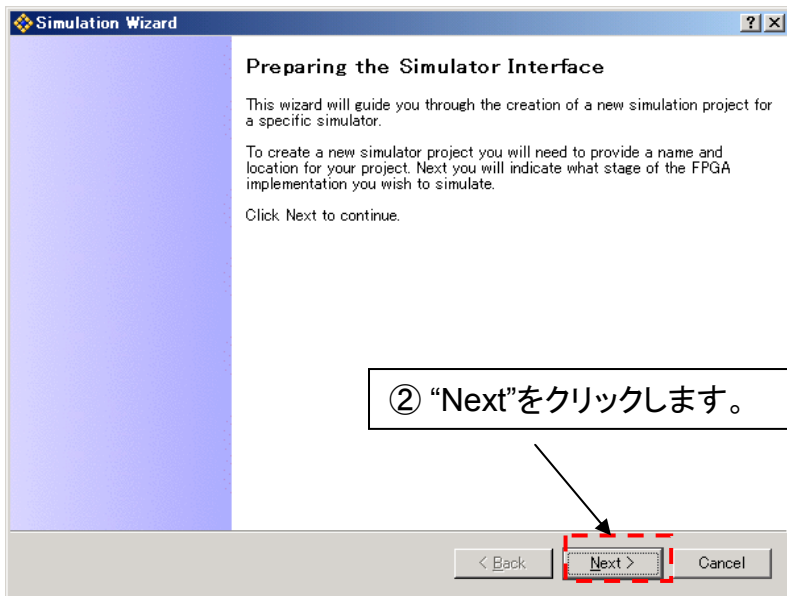
②Hierarchyウインドを選択

③ソースのTOPファイルを選択し
右クリック⇒Verilog Test Fixture
Templateを選択します。

④Projectフォルダ内にTFTファイルが
生成されます。この雛形を使用してテ
ストベンチを作成することができます。

■シミュレーションを実施するには、Simulation Wizardを使うのが便利です。以下にSimulation Wizardを使用したシミュレーション方法を記載します。



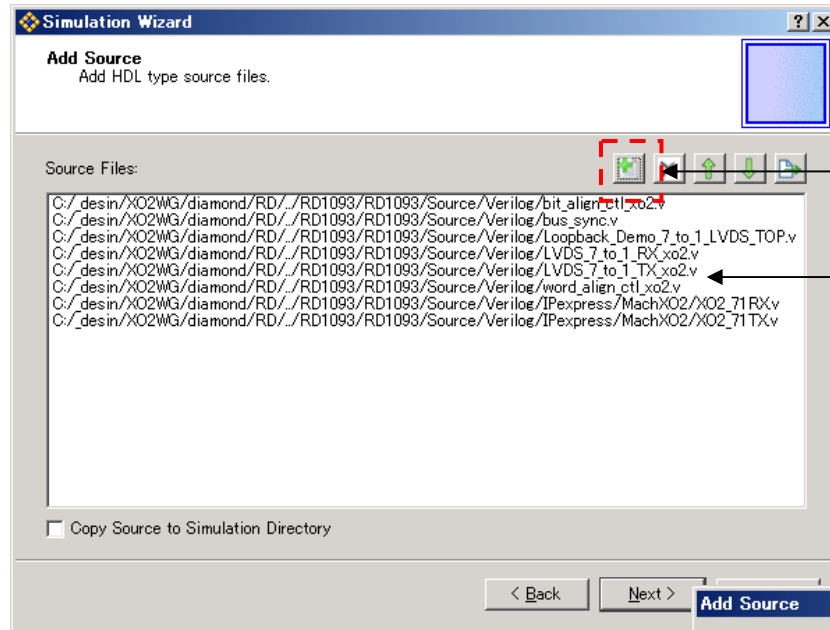


シミュレーションの方法を選択します

RTL:
シミュレータがRTLをインポートし、コンパイルを行いシミュレーションを実施します。

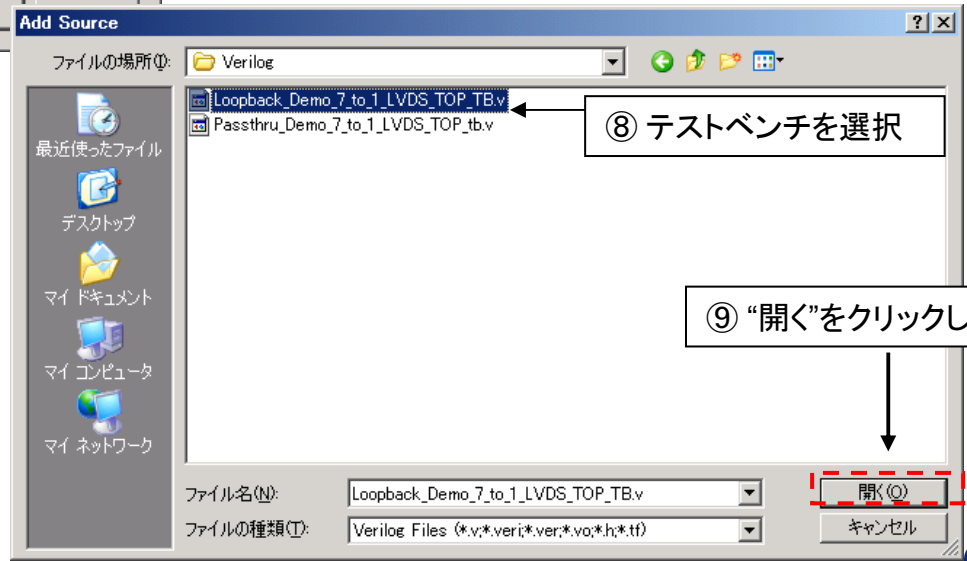
Post-Map Gate-Level:
Mapping後生成されるネットリストファイル(VO,VHOファイル)を使用してシミュレーションを実施します。

Post-Map Gate-Level+Timing:
上記ネットリストファイル+遅延情報ファイル(SDFファイル)を取り込んでシミュレーションを行います。一番実機の動きに近いシミュレーションが可能ですがシミュレーションの実行に時間がかかります



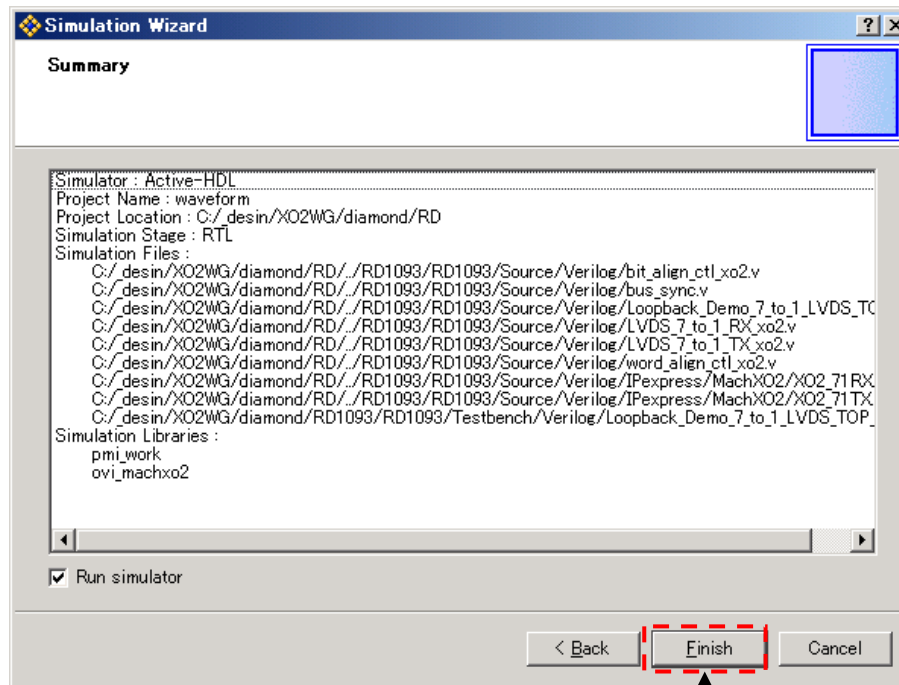
⑥ このアイコンをクリックします。

⑦ 全てのデザインソース及び
テストベンチが含まれているか確認をします。
テストベンチが別のロケーションに含まれる場合は、
上記+マークのアイコンをクリックします。

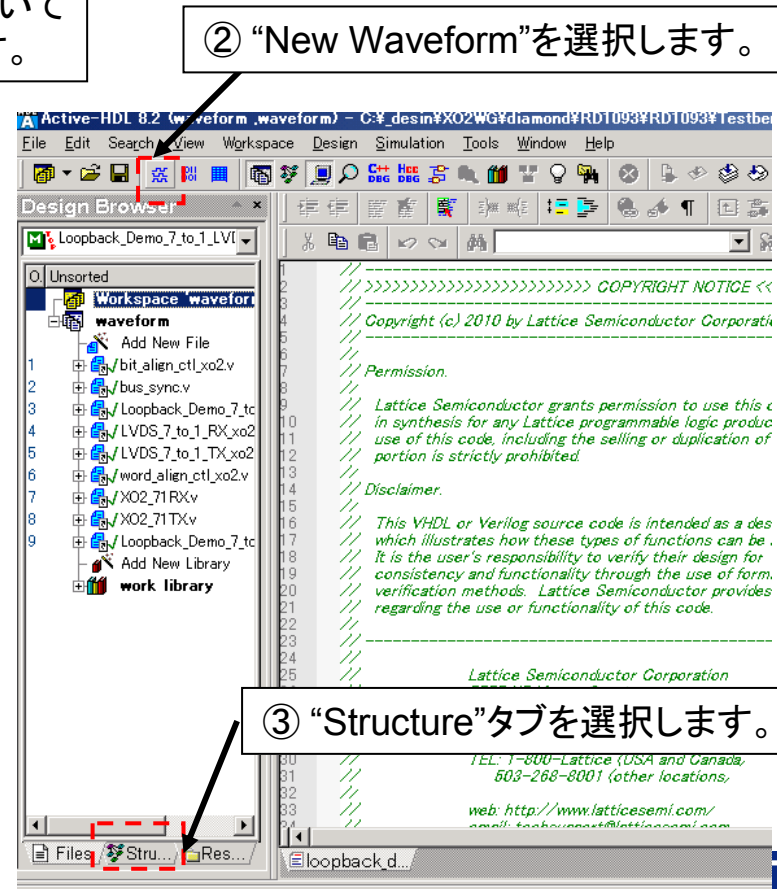
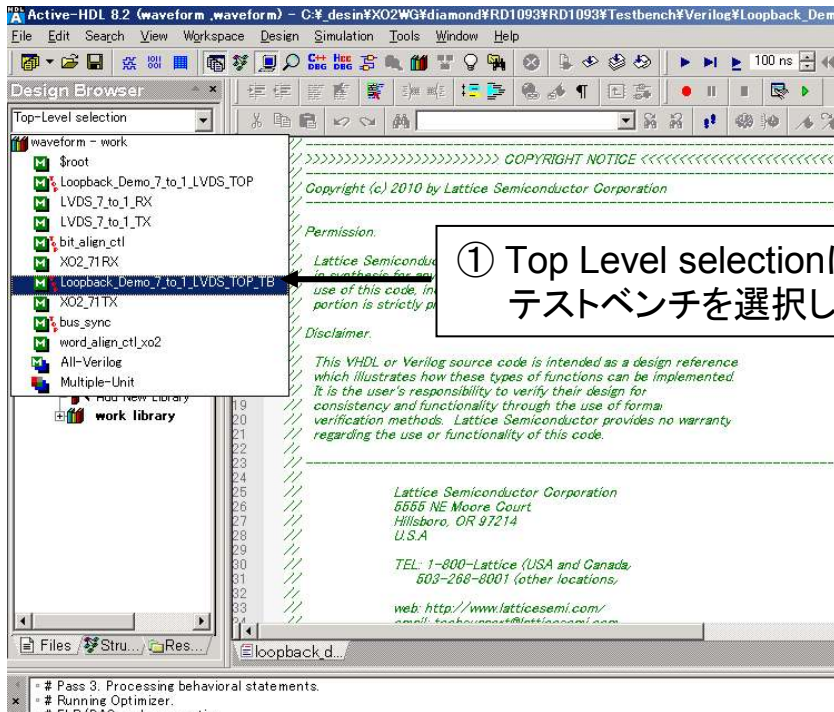


⑧ テストベンチを選択

⑨ “開く”をクリックします。



⑩ “Finish”をクリックします。



確認したい信号が存在するインスタンスを選択します。

Structureウインド内に信号が表示されますので、Waveformウインドにドラック&ドロップを行います。

信号をWaveformウインドに移動させた後、上部実行アイコンをクリックしてシミュレーションを開始させます。

Name	Value	Sti...
R= clk_tx		
R= gsrn_jn		
lvds_clk		
lvds_a		
lvds_b		
lvds_c		

Name	Value
R= errc	Unavailable
R= errd	Unavailable
berr_a	Unavailable
berr_b	Unavailable
berr_c	Unavailable
berr_d	Unavailable
tx_sclk	Unavailable
RST	Unavailable
stop1	Unavailable
stop2	Unavailable
stop3	Unavailable
stop	Unavailable
blink_count	Unavailable
blink_time	Unavailable
BBITS	Unavailable

Name	Value	Sti...
R= sim time		
P= ...		
TX_eclk		
GSRN		
TCLK_out		
TA_out		
TB_out		
TC_out		
TD_out		
RCLK_jn		
RA_jn		
RB_jn		
RC_jn		
RD_jn		

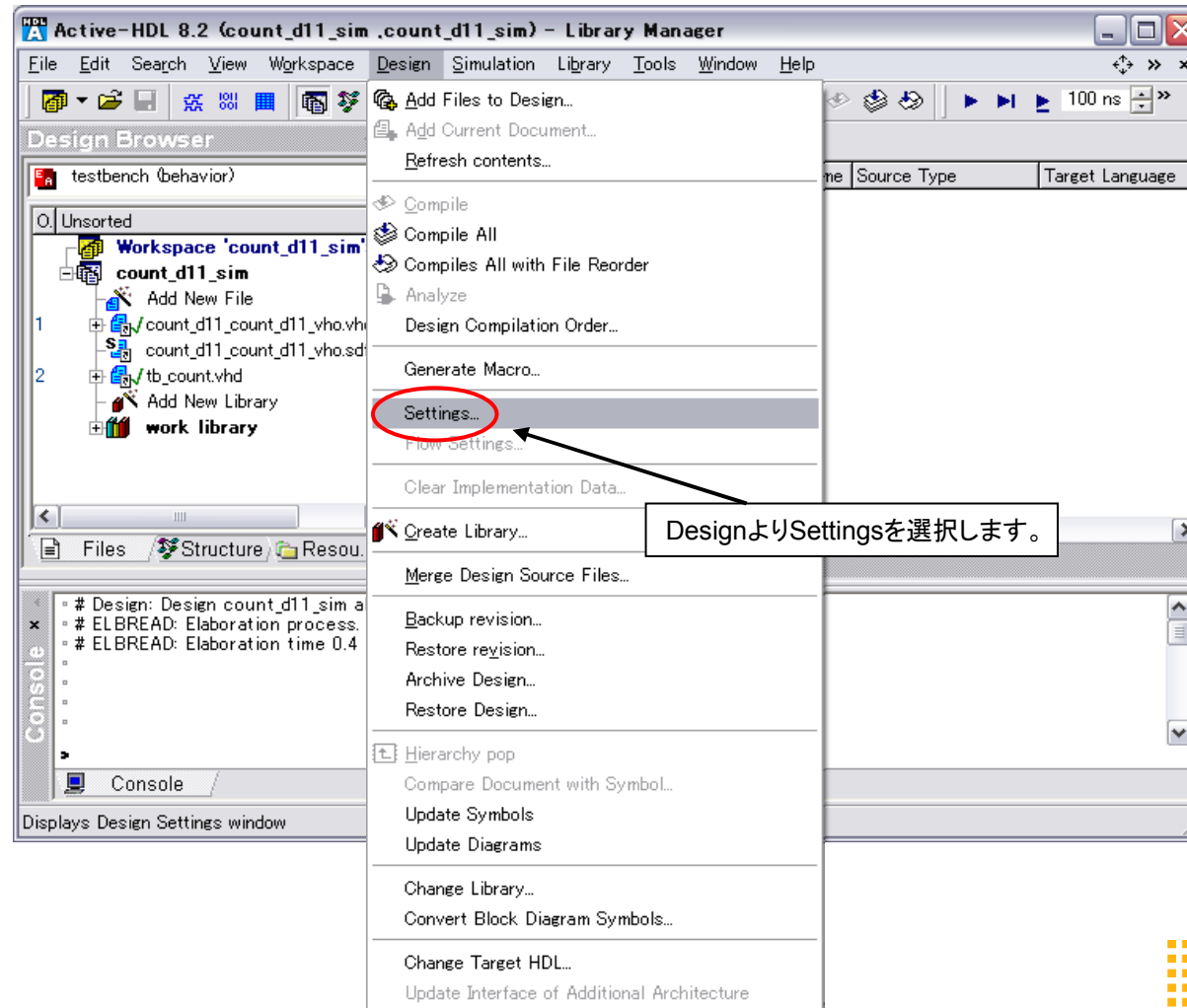
Files Structure Resources

loopback_d... waveform ...

```

# ELBREAD: Warning: Module 'bus_sync' does not have a `timescale directive, but previous modules do.
# ELBREAD: Warning: Module 'word_align_ctl_xo2' does not have a `timescale directive, but previous modules do.
# ELBREAD: Elaboration time: 0.1 [s]
    
```

■ タイミングシミュレーションを実施するには、Simulation WizardにてPost-Map Gate-Level+Timingを選択します。また、Waveformを表示する前に以下設定が必要になります。設定を行わない場合、SDFファイルが無効となり内部遅延がシミュレーションに反映されません。



Design Settings

Category:

- General
 - Top-level
 - Compilation
 - Simulation
 - SDF**
 - VHDL
 - Verilog
 - Access to Design Objects
 - Parameters
 - Trace/Debug
 - Expression Coverage
 - Code Coverage/Profiler
 - Verilog PLI Application
 - C Code Debug Settings
 - Handel-C Code Debug and Co-simul

SDF

Files - Region	Value	Load
./count_d11#count_d11_count_d11_who.sdf	/uut	Average

①SimulationのSDFを選択します。

②LoadをドロップダウンメニューからYesに設定します。

③OKで設定完了です。

Buttons: Default, OK, Cancel, Apply

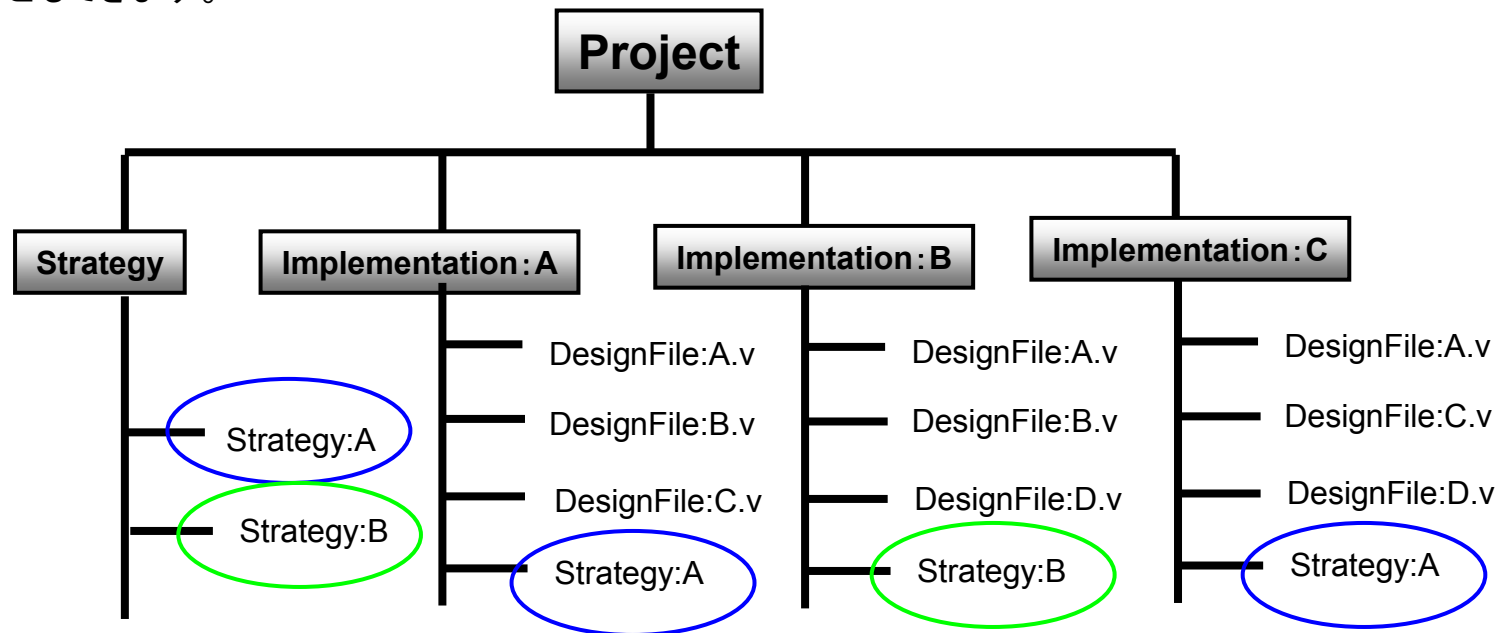
- デザイン変更を行い、再度シミュレーションを行う場合には、以下の方法でシミュレーションを実施することが可能です。但し、デザインファイルが追加された場合などはうまく起動ができませんのでご注意ください。

The image shows a software interface with two main windows. On the left is the 'File List' window, which displays a hierarchical tree of project files. Under the 'Script Files' folder, the file 'test_tb/test_tb.spf' is highlighted with a red circle. Below the File List are three tabs: 'File List', 'Process', and 'Hierarchy'. On the right is the 'Simulation Wizard' window, which has a 'Summary' section. This section contains the following text: 'Simulator : Active-HDL', 'Project Name : test_tb', 'Project Location : C:/lsc/diamond/1.0/examples/attributes', 'Simulation Stage : Post-Map Gate-Level', 'Simulation Files : C:/lsc/diamond/1.0/examples/attributes/attributes/attributes_attributes_mapvo.vo, C:/lsc/diamond/1.0/examples/attributes/attributes/attributes_attributes_mapvo.sdf, C:/lsc/diamond/1.0/examples/attributes/simulation/enc_dec_tb.v', and 'Simulation Libraries : pmi_work, ovi_xp2'. At the bottom of the Simulation Wizard, there is a checked checkbox 'Run simulator' and three buttons: '< Back', 'Finish', and 'Cancel'. The 'Finish' button is circled in red. A callout box with a white background and black border points to the 'Finish' button and contains the following Japanese text: 'Summaryが表示されます。こちらでデザインファイルが正しく読み込まれていることを確認し、Finishを押します。後は自動でシミュレータが立ち上がります。'

FileList内のScriptFilesの項目から.spfファイルを選択。ダブルクリックを行います。

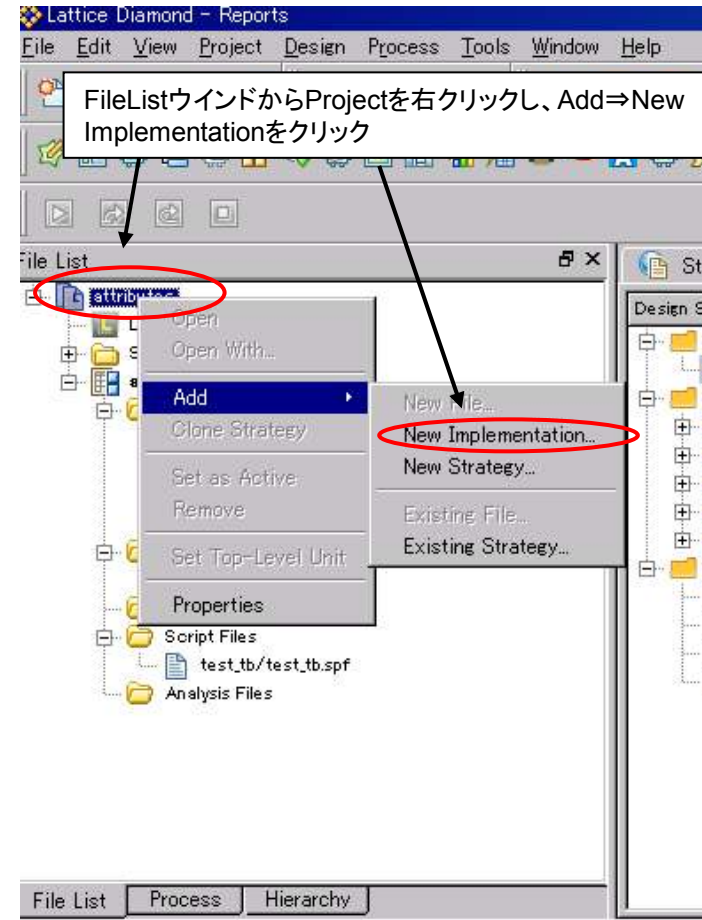
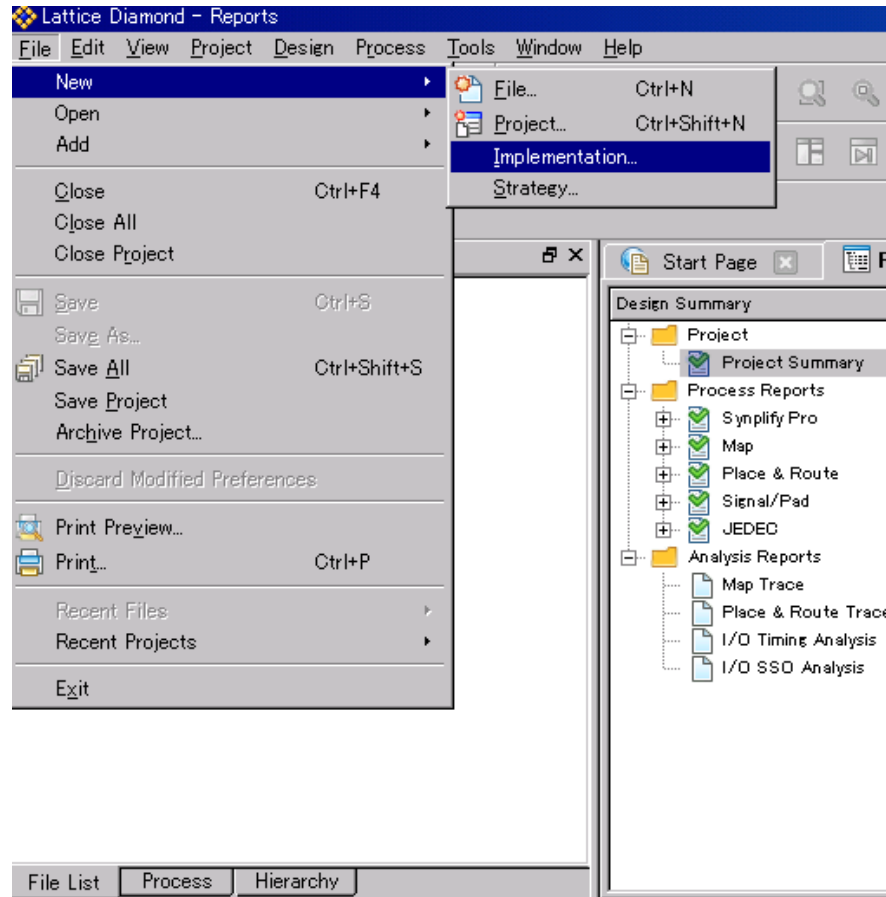
Summaryが表示されます。
こちらでデザインファイルが正しく読み込まれていることを確認し、Finishを押します。
後は自動でシミュレータが立ち上がります。

■ Lattice Diamondはユーザー様で作成したデザインProject単位で管理を行います。LatticeDiamondからはProjectの下に複数のサブProjectを構成でき、サブProjectごとに違う設計データを持たせてデザインの管理を行うことができます。このサブProjectは**Implementation**と呼びます。Implementation毎にStrategy(ツールのProperty、コンパイルオプション)や制約を持たせることができ、全てのImplementationを並列にコンパイルさせ、それぞれのコンパイル結果を比較することもできます。



例えば、同じデザインに対してImplementationを複数作成し、それぞれ違うStrategyを設定行い、並列にコンパイルを行う良い結果を選択して使用することが可能です。これにより、設計期間の短縮化を図ることができます。

- Implementationを新規で作成します。上部プルダウンメニューから「File」⇒「New」⇒「Implementation」を選択
 或いは、FileListウインド内のProjectを右クリックして⇒「Add」⇒「New Implementation」を選択します



① Implementation名を入力します。

② Strategyを選択します。
* 詳しいことは後述致します。とりあえず現在のままで構いません。

③ ソースを取り込みます。
ソースの取り込みは2種類の方法がございます。
1. Browseで特定のソースを指定する
2. 現在存在するImplementationからソースをまるまるImportする

④ OKを押すとImplementationがFileListウインド内に作成されます。

新たにImplementationを作成するとそのImplementation専用フォルダが作成されます。チェックが入っているとそのフォルダにソースがコピーされます。

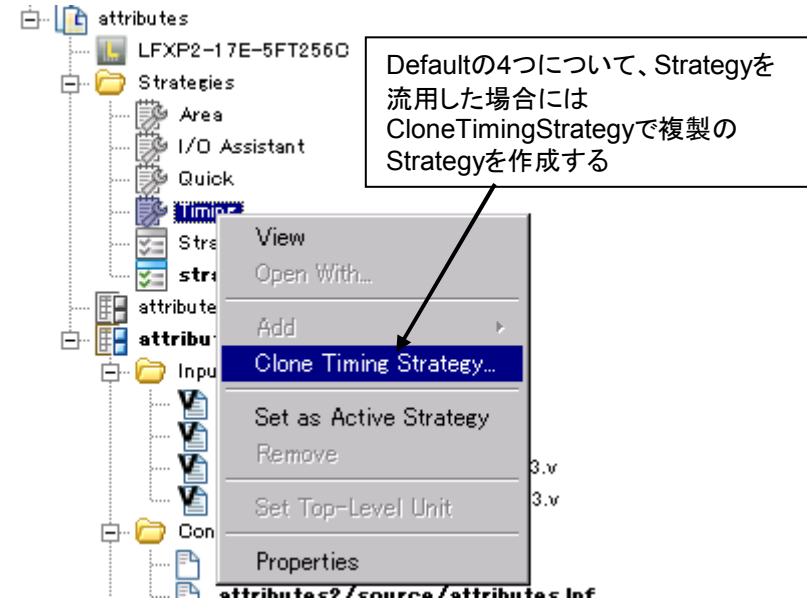
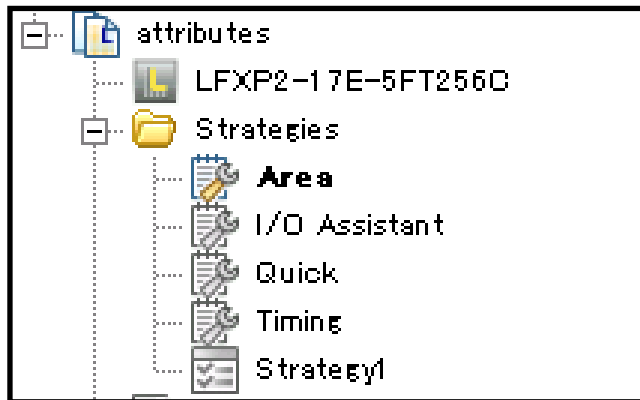
File List Process Hierarchy

- Implementationを作成した後、内部のデザインファイル、Strategy、各種制約に対して変更を加える場合には、該当のImplementationをアクティブにする必要があります。

変更を加えたいImplementationをFileListウインドから選択し、右クリック。Set as Active Implementationを選択

選択したImplementationがアクティブになり、Implementation内の各種ファイルが表示される

- Strategyとはコンパイル時のオプション設定リストになります。複数のStrategyを作成し、Implementation毎に共有させて使用することが可能です。Defaultの状態では5つのStrategyが用意されています。



Area	: 必要なリソース(SLICE)数が最小になるよう設定されたStrategy
I/O Assistant	: I/O Assitant機能(IOの配置ルールチェック)を実行すうためのStrategy
Quick	: プロセスの処理時間を優先するよう設定されたStrategy
Timing	: タイミング要求を優先するよう設定されたStrategy(全てデフォルト)
Strategy1	: Timingと同じ

- * 上部四角で囲んだStrategyについてはコンパイルオプション内容を変更できません。流用したい場合には、流用させたいStrategyを右クリックし、Clone Timing Strategyを選択して新規にStrategyを作成します。

- StrategyをTargetとしているImplementation内で使用するにはアクティブにする必要があります。

アクティブにしたいStrategyを選択

アクティブになるとStrategyの文字が太字になります

ターゲットとしているImplementationをActiveにした後、使用したいStrategyを右クリックし、Set as Active Strategyを選択します。

■ Strategyの変更方法を以下に記載いたします。

Strategyをダブルクリックし、StrategyWindowを表示します。変更したいコンパイルオプションをこのウインドから選択します。

Name	Type	Value
Allow Duplicate Modules	T/F	False
Area	T/F	True
Arrange VHDL Files	T/F	False
Default Enum Encoding	List	Default
Disable IO Insertion	T/F	False
FSM Encoding	T/F	True
Fanout Limit	Num	100
Fix Gated Clocks [0-3]	Num	3
Fix Generated Clocks [0-3]	Num	3
Force GSR	List	False
Frequency (MHz)	Num	200
Input SDC Constraint File(.sdc)	File	
Number of Critical Paths	Num	3
Number of Start/End Points	Num	0
Output Preference File	T/F	False
Pipelining and Retiming	List	None
Push Tristates	T/F	True
Resource Sharing	T/F	True
Use Clock Period for Unconstrained I/O	T/F	True
Verilog Input	List	Verilog 2001

Allows the use of duplicate modules in your design.

- Diamondでは複数のImplementationを並列処理でコンパイルすることが可能です。並列処理には“Run Manager”を使用します。以下よりRun Managerの使い方を示します。

Run ManagerはRun Managerアイコンから起動します。

Status
Implementation毎の実行状況

- Completed : 処理完了済み
- Running : 処理の実行中
- Waiting : 処理の実行待ち
- Failed : エラーの為に処理が中断
- Cancelled : 処理中にユーザによりRun Managerが強制停止

Unrouted
未配線の信号数

Implementation<Strategy>	Status	Progress	Start	Run Time	Score	Unrouted	Level/Cost	Description
<input type="checkbox"/> attributes<Strategy1>	Ready	100%	木 19:56:30	7	3935	0	5,1	attributes
<input type="checkbox"/> attributes2<Area>	Ready	0%	Unknown...	0	NA	NA	NA	attributes2

Process
処理の進捗状況

Run Managerを起動すると現在のProjectに存在するImplementationが全て表示されます。(この例では2つのImplementationが存在します)

RunTime
処理に要した時間。単位は秒

Score
置配線結果でもっとも良かった(小さかった)Timing Score

Level/Cost
最も良いTiming Scoreを出したLevelとCost Tableの値

■ Run Managerの実行方法を記載します

①実行したいImplementationのチェックボックスにチェックを入れます。全てのImplementationを選びたい場合、右クリック⇒SelectALLで全てを選択可能です

Implementation<Strategy>	Status	Progress	Start	Run Time	Score	Unrouted	Level/Cost	Description
<input checked="" type="checkbox"/> attributes<Strategy1>	Ready	100%	木 19:56:30	7	3935	0	5_1	attributes
<input checked="" type="checkbox"/> attributes2<Area>	Running	3%	金 20:28:08	11	NA	NA	NA	attributes2

②初めて実行する場合にはRUNボタンを押します。
2度目以降はRERUNボタンを押します。
あとは自動で論理合成⇒配置配線まで処理されます

【注意】

Run Managerでは複数のImplementationを並列で実行可能ですが必ずお使いのPCのCPUコア数以上にならないよう並列処理数を調整して下さい。
コア数以上になりますと逆に処理速度が低下する可能性があります。

より詳細なお問合せ、ご質問等に関しましては、技術サポート貴社担当FAE
または下記技術サポート窓口までお気軽にお問い合わせ下さい。

株式会社 マクニカ テクスターカンパニー ラティス製品 技術サポート窓口
電話 045-470-9841/FAX 045-470-9842
Email lattice@macnica.co.jp
URL <http://www.tecstar.macnica.co.jp/contact/index.html>

