



ネットリスト・IPの扱い方について





本資料はLattice製FPGAを対象とし、Diamondにてネットリストの取り扱い方法について記載したマニュアルです。

ネットリスト作成時や合成時の手順及び波形シミュレーション時の注意点を、順を追って説明していきます。

まずは、下記手順に従って、ブラックボックス化したいIPのみをimportしたDiamondプロジェクトを作成して下さい



Agenda

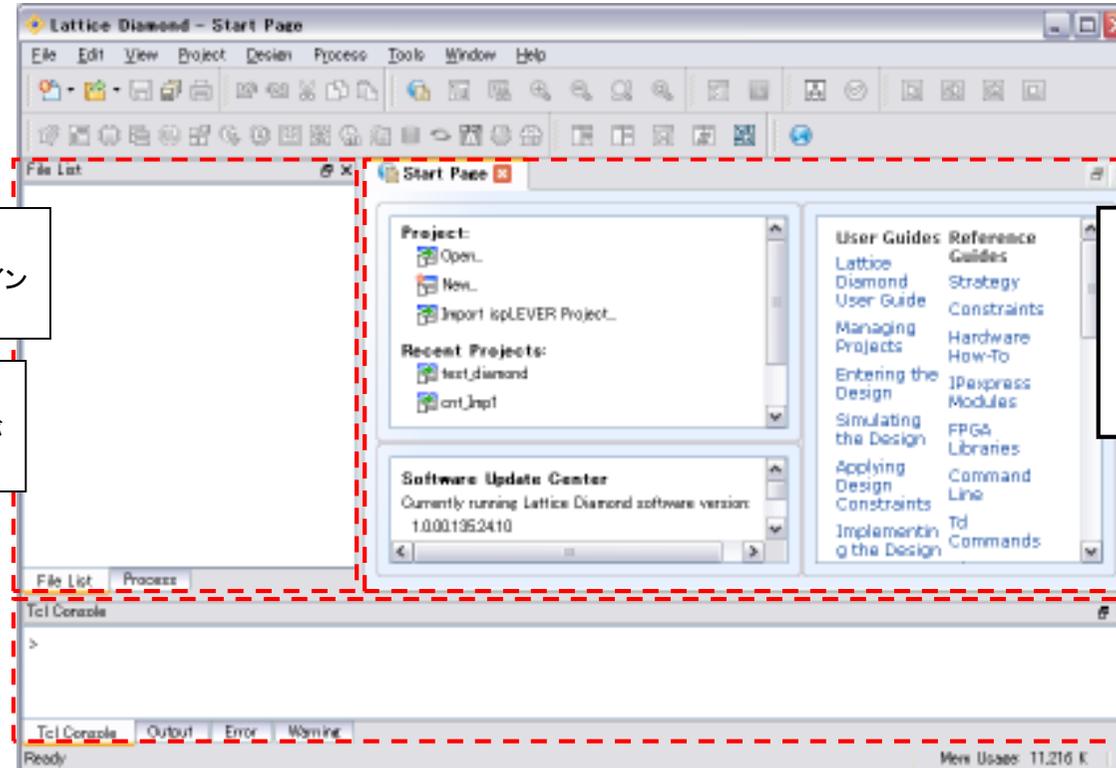
1. Diamondの起動方法について
2. ネットリストファイル(.ngo)の生成方法について
3. ネットリスト(NGOファイル)を取り込んだデザインの作成方法について
4. ネットリスト(NGOファイル)デザインに対する
Active-HDL波形シミュレーション方法について

1. Diamondの起動方法について



- 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事も可能です。

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事も可能です。

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。



- 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。

①Files>New>Projectより、新規プロジェクトの作成ウィザードを起動します。

* NewアイコンからProjectを選択でも起動します

* StartPageのNewからも同様に起動可能です

File List Process

Tcl Console

Tcl Console Output Error Warning

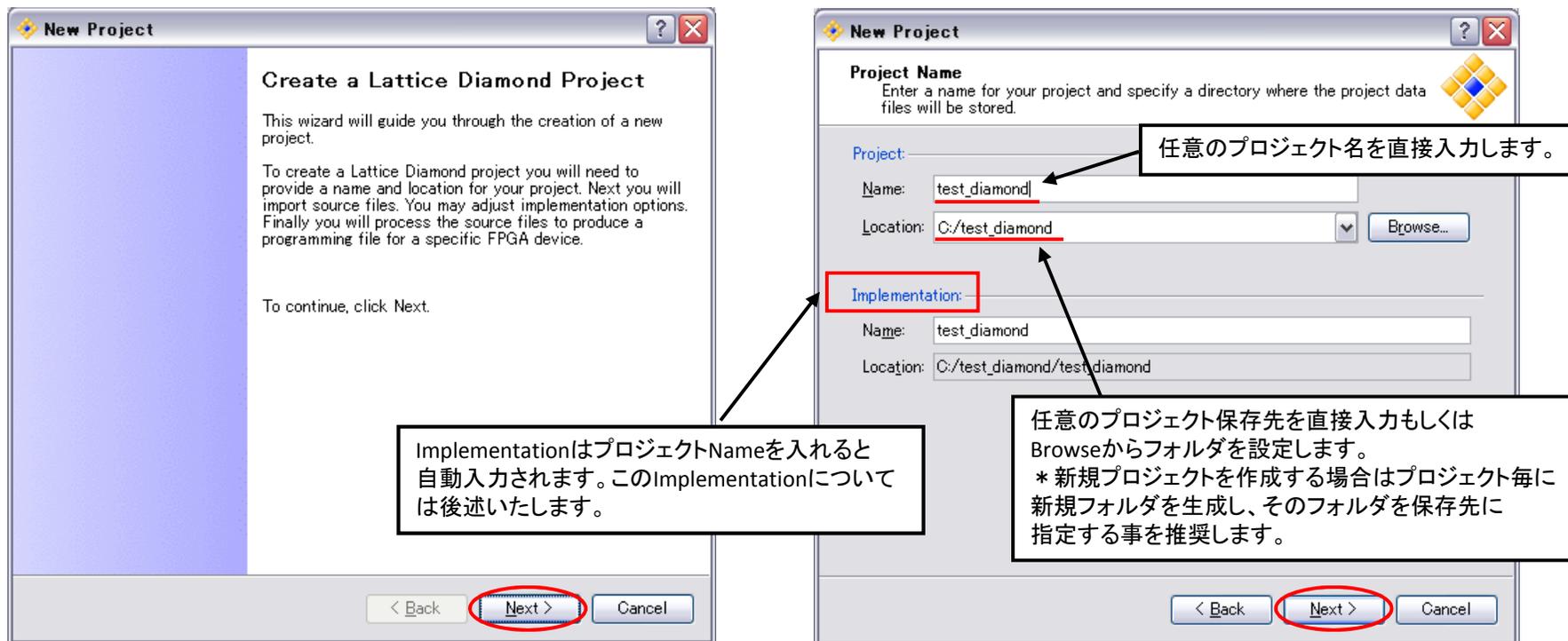
Open New Project wizard

Mem Usage: 19,664 K



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。

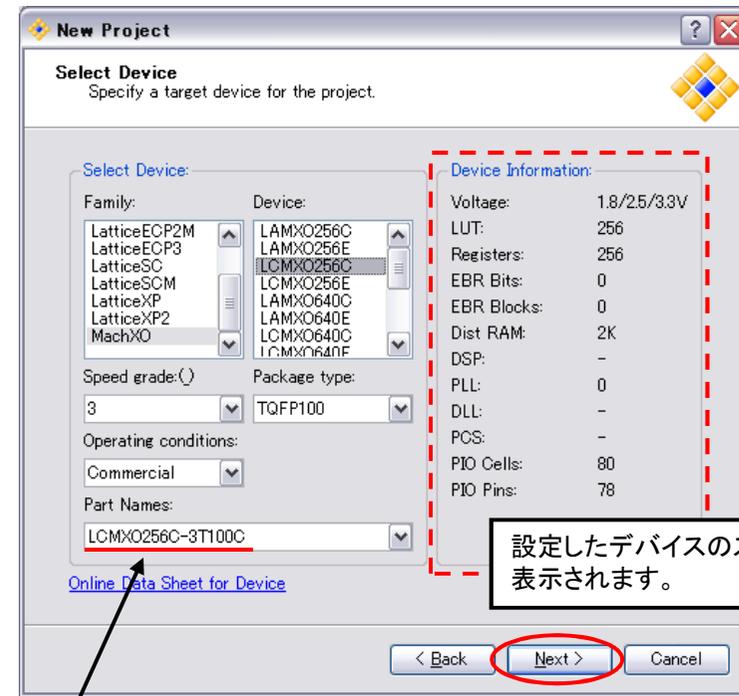


* 注意

- ・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)



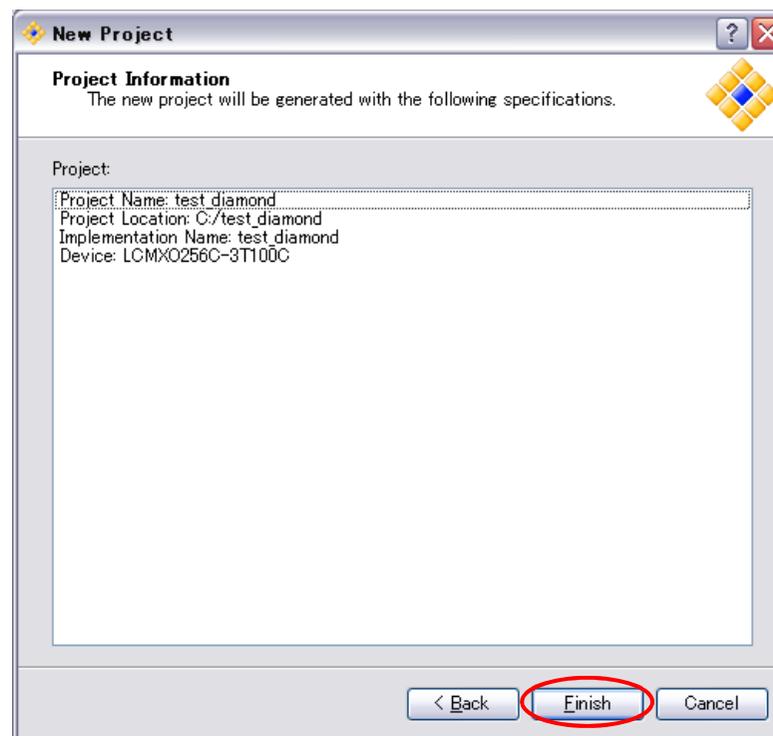
- ④既存のデザインソースがある場合は、プロジェクト作成時にインポートする事が可能です。(Add Sourceより)
既存のデザインソースが無い場合は何もせずにNextをクリックして下さい。
- ⑤次に設計するデバイスを選択します。Familyよりデバイスファミリーを選択し、Deviceから詳細のデバイスを選択します。Speed gradeでデバイスのスピードグレード、Package typeよりパッケージ、Operating conditionsより温度グレードを設定し、Nextをクリックして下さい。



設定したデバイスの型式が表示されます。



- ⑥ New Project ウィ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、変更箇所のウィ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。

インプリメンテーション名以外にInput Filesを右クリックしても同じ方法でデザインファイルの追加が可能です。

Removeはあくまでプロジェクトから削除する意味となりますので、RemoveしたデザインファイルがPC上から削除されるという意味ではありません。

※上記デザインはサンプル例です。

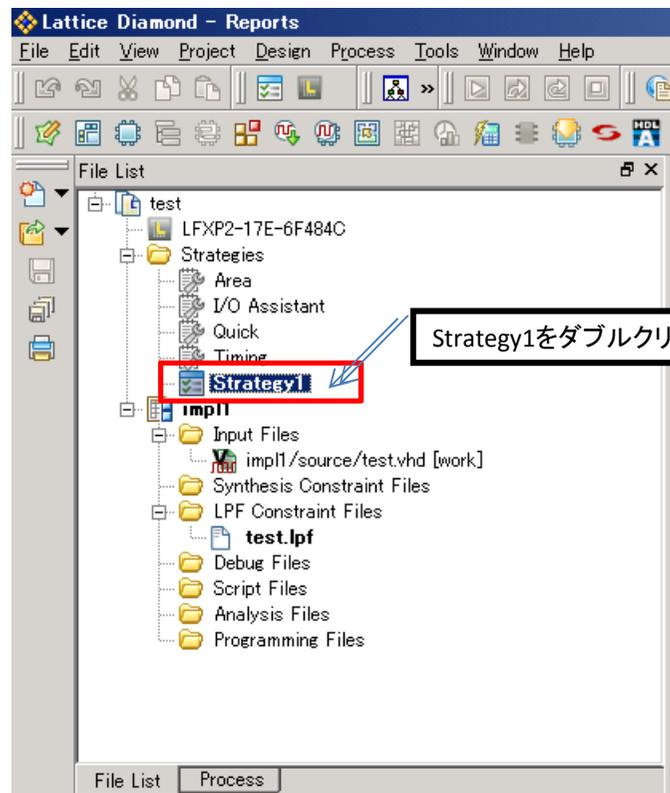
2. ネットリストの生成方法について



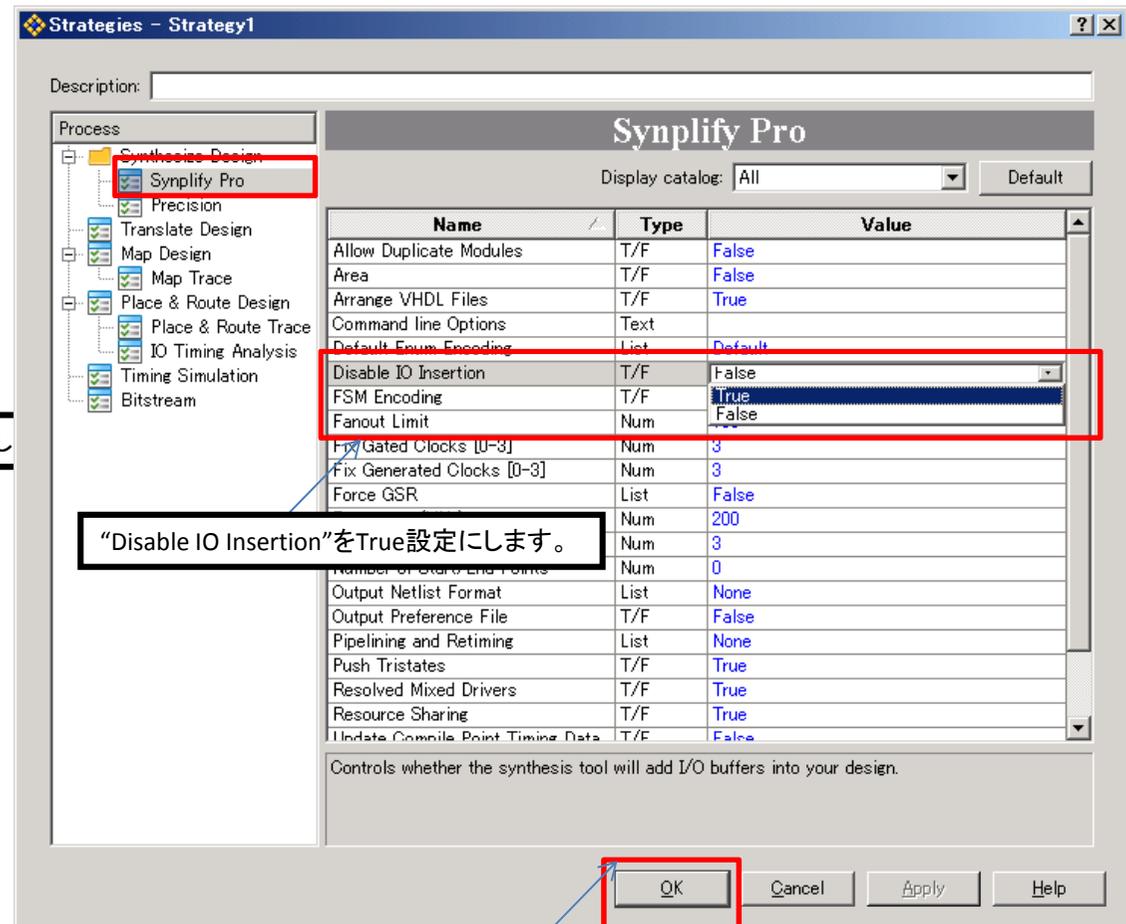
※このプロセスはIPのネットリストを作成・提供する場合に必要となります。

作業② そのプロジェクトにおいて、Strategy内のSynplify Proのオプションにおいて、
”Disable IO Insertion”を”True”に変更して下さい。

(この作業を行わないとI/Oバッファが追加されてしまい、トップソース以下の階層に
IPデザインを置いた場合にエラーが発生します。)



Strategy1をダブルクリックし



“Disable IO Insertion”をTrue設定にします。

設定が終了しましたら、OKをクリックしてください。



③Translate Designを実行して下さい。
上記手順で、“モジュール名_インプリメンテーション名.ngo”というファイルが生成されます。

The screenshot shows the Lattice Diamond Reports window with the 'Process' tree on the left and a file explorer window on the right. The 'Translate Design' step is highlighted in the process tree. The file explorer shows the contents of the 'impl1' folder, with several files highlighted in red boxes: 'test_impl1.edi', 'test_impl1.fse', 'test_impl1.ngd', 'test_impl1.ngo', and 'test_impl1.srd'. A callout box points to 'Translate Design' with the text: "Translate_Design"をダブルクリックします。 Another callout box points to the generated files with the text: プロジェクトフォルダ内に"モジュール名_インプリメンテーション名.ngo"と"モジュール名_インプリメンテーション名.edi"が生成されます。

名前	サイズ	種類	更新日時
synlog		ファイル フォルダ	2012/01/12 11:28
synlog		ファイル フォルダ	2012/01/12 11:26
synlog		ファイル フォルダ	2012/01/12 11:28
synlog		ファイル フォルダ	2012/01/12 11:28
synwork		ファイル フォルダ	2012/01/12 11:28
build_status	2 KB	BUILD_STATUS ファ...	2012/01/12 11:29
recordref	0 KB	RECORDREF ファイル	2012/01/12 11:28
automake.log	2 KB	テキスト ドキュメント	2012/01/12 11:29
run_options.txt	2 KB	テキスト文書	2012/01/12 11:28
scratchproject.prs	2 KB	PRS ファイル	2012/01/12 11:28
stdout.log	2 KB	テキスト ドキュメント	2012/01/12 11:28
test_impl1_areasrr	1 KB	AREASRR ファイル	2012/01/12 11:28
test_impl1.edi	6 KB	EDI ファイル	2012/01/12 11:28
test_impl1.fse	0 KB	FSE ファイル	2012/01/12 11:28
test_impl1.ngd	6 KB	NGD ファイル	2012/01/12 11:29
test_impl1.ngo	3 KB	NGO ファイル	2012/01/12 11:29
test_impl1.srd	4 KB	SRD ファイル	2012/01/12 11:28
test_impl1.srt	11 KB	SRF ファイル	2012/01/12 11:28
test_impl1.srl	3 KB	SRL ファイル	2012/01/12 11:28
test_impl1.srm	8 KB	SRM ファイル	2012/01/12 11:28
test_impl1.srs	3 KB	SRS ファイル	2012/01/12 11:28



IPを使用する際の設計対象がVerilogの場合、例えば以下のようなエンティティのみ記述されたトップソースを用意する必要があります (VHDLの場合は必要ありません)。このトップソースは実際にVerilogソースとネットリストを混在設計する際に、プロジェクトにインポートして使用します。

```

0000001 module top_demo(
0000002     rst_n,
0000003     core_clk,
0000004     hdinp0,
0000005     hdinn0,
0000006     hdoutp0,
0000007     hdoutn0,
0000008     error_inject,
0000009     clear_error_cnt,
0000010     err_inj_LED,
0000011     activity_LED,
0000012     pass_LED,
0000013     fail_LED,
0000014     tdi,
0000015     tdo,
0000016     tck,
0000017     tms
0000018 );
0000019
0000020
0000021 input    rst_n ;
0000022 input    core_clk ;
0000023 input    error_inject,clear_error_cnt;
0000024 input    hdinp0 ;
0000025 input    hdinn0 ;
0000026 output   hdoutp0 ;
0000027 output   hdoutn0 ;
0000028 output   activity_LED;
0000029 output   pass_LED;
0000030 output   fail_LED;
0000031 output   err_inj_LED;
0000032 input    tdi;
0000033 input    tck;
0000034 input    tms;
0000035 output   tdo;
0000036
0000037 endmodule

```

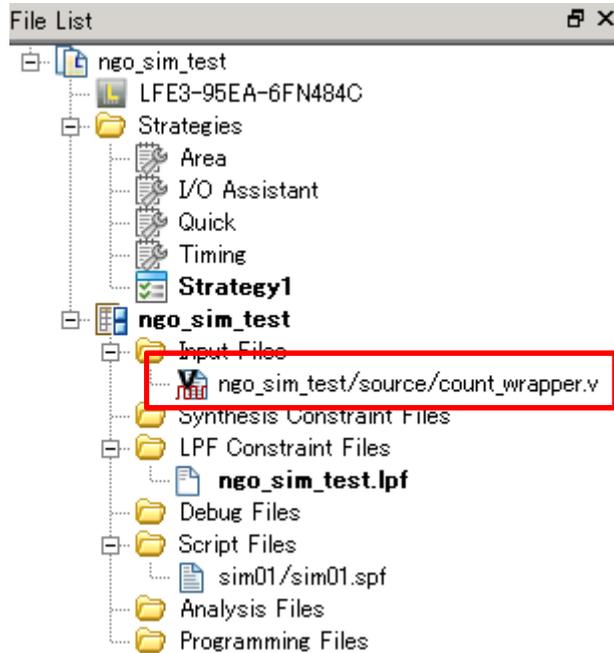


3. ネットリスト(NGOファイル)を取り込んだ デザインの作成方法について



新規にプロジェクトファイルを作成します。必要となるソースは以下2つになります。

1. TOPソース
2. ngoファイル



TOPソースは以下の様に、
ブラックボックスのモジュール名とポートのみ定義した
ラッパーモジュールを作成して下さい。
(例えば以下のようなモジュールになります。)

```

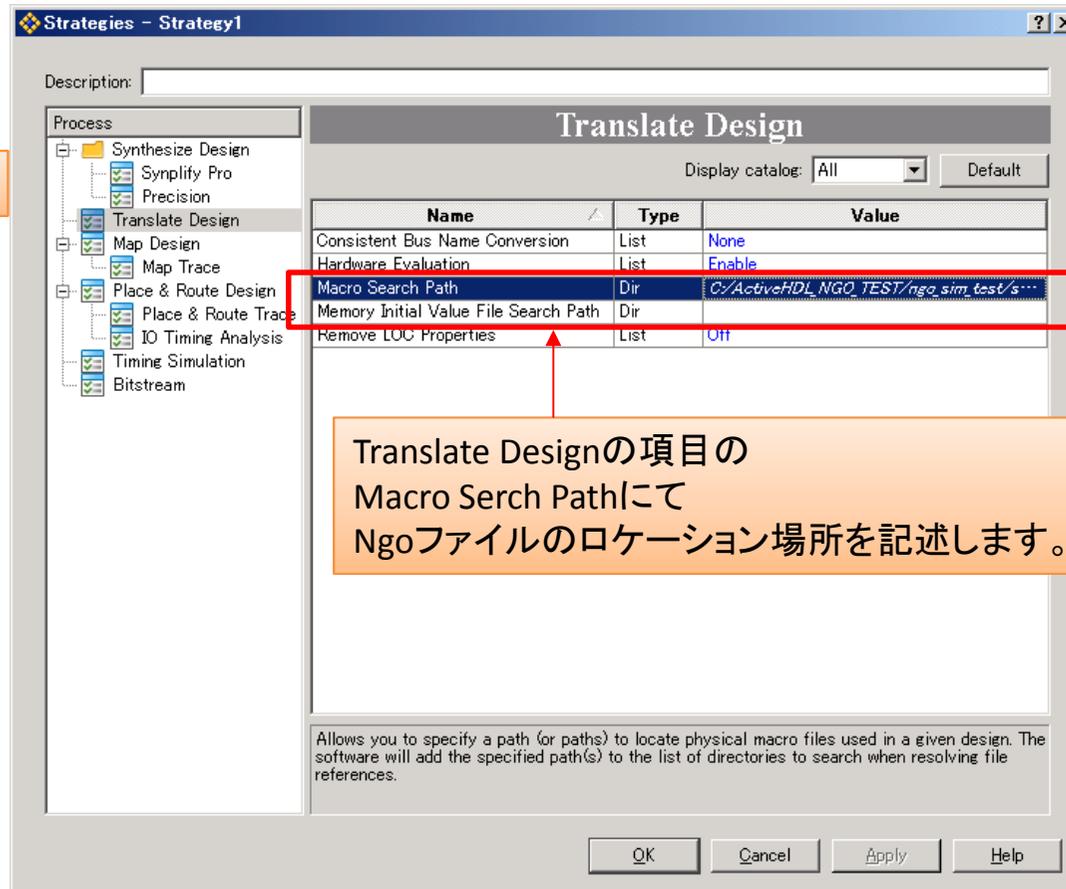
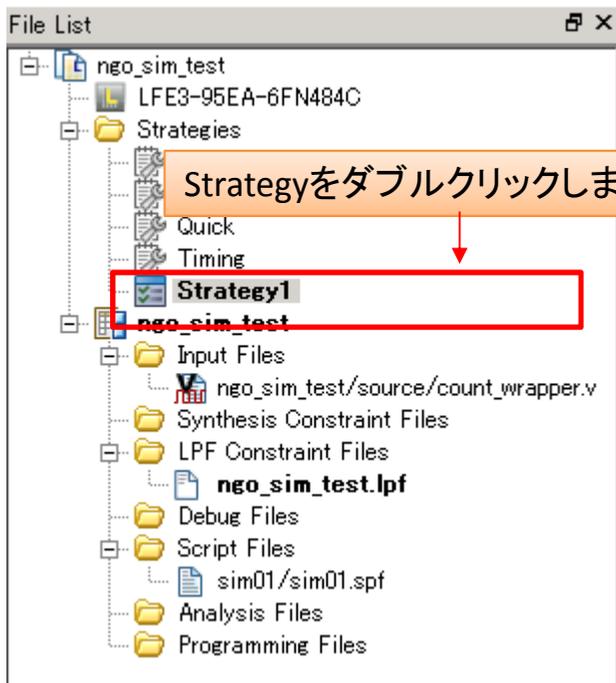
module count↓
(
  c,↓
  clk,↓
  rst↓
)/* synthesis FILE="ngo_sim_test_ngo_sim_test.ngo" */;↓
  input          clk;↓
  input          rst;↓
  output [2:0]   c;↓
↓
endmodule[EOF]
  
```

Moduleの最後の)には
*/ synthesis FILE = "○○.ngo" */;
とngoファイルを指定します。

また、ngoファイルはldfファイルと同じフォルダに保存しておきます。



Ngofファイルが正常に反映されない場合には？
以下手順にて指定してください。

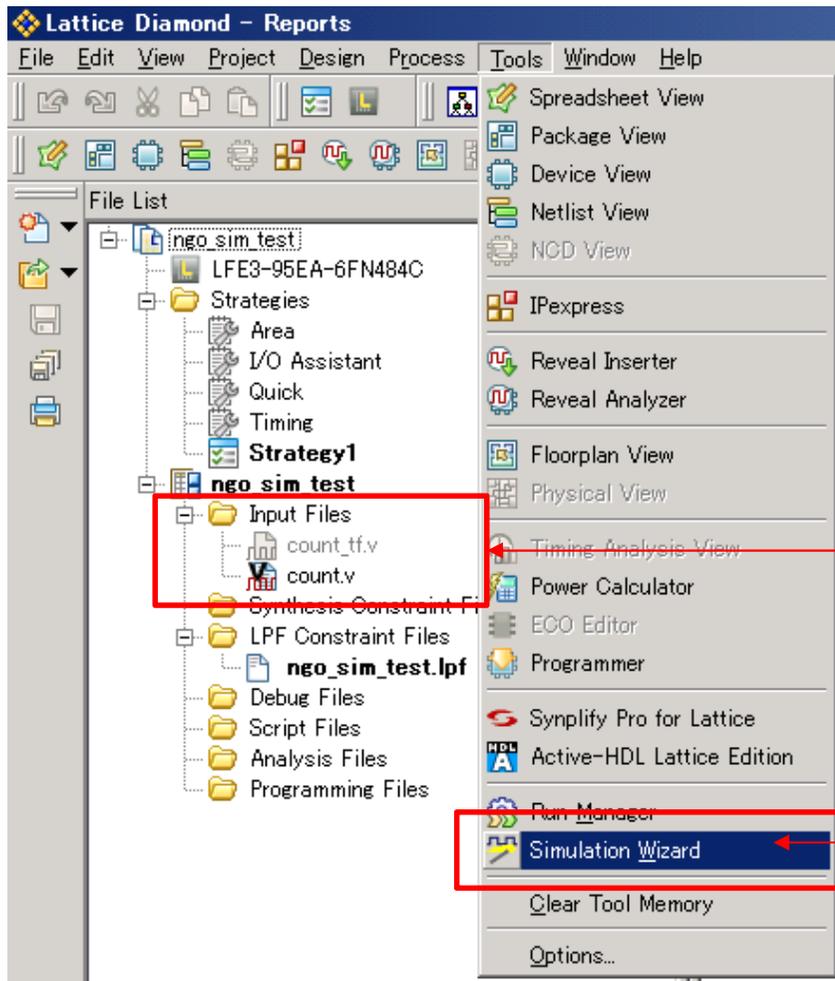




4. ネットリスト(NGOファイル)デザインに対する Active-HDL波形シミュレーション方法について



ネットリストデザイン(ngo)を用いてActive-HDL波形シミュレーションを行う場合
ネットリスト(ngo)ファイルを使用する前のデザインソースからActiveHDL上にてLibraryを作成する必要があります。
以下は、Libraryを構成する方法について説明致します。

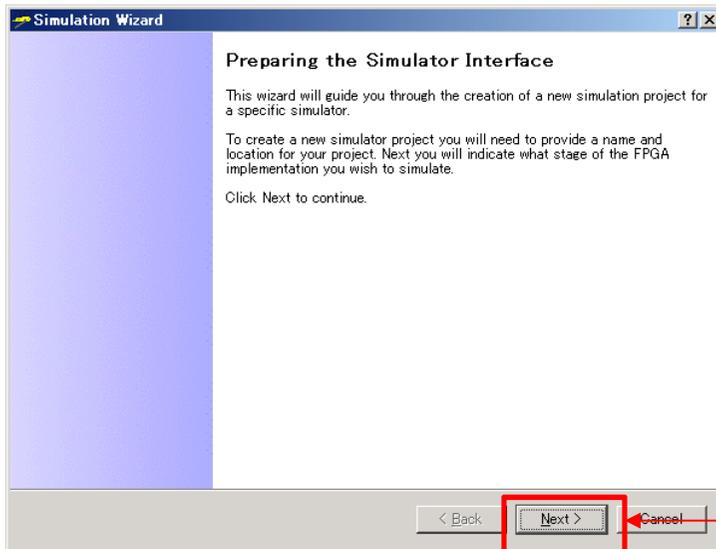


※ ngoファイルを使用する前の
デザインソースを使用致します。

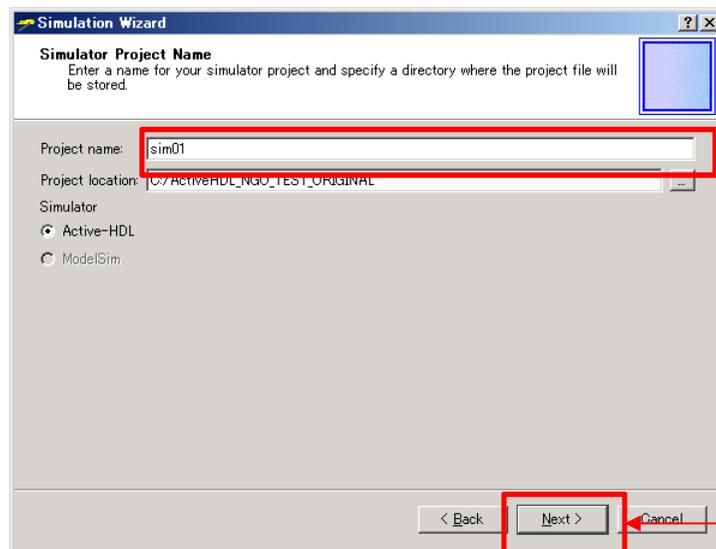
① Diamond上でTool > Simulation Wizard
を実行します。



※こちらはDiamondでの設定画面です。



② Next > をクリックします。

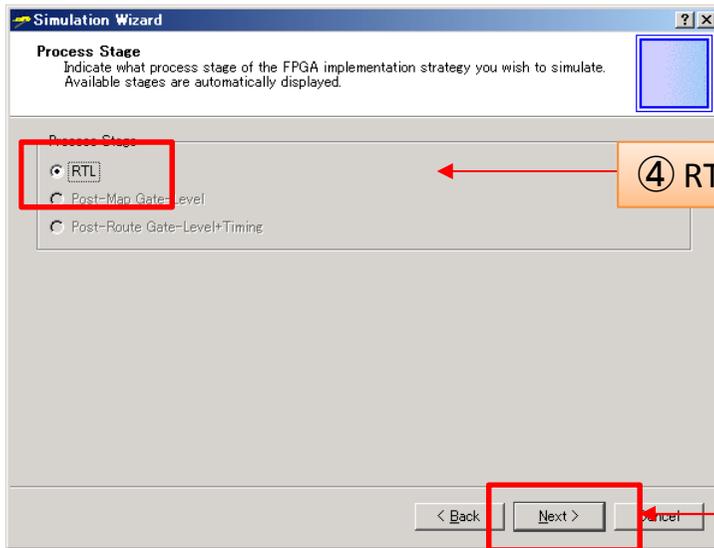


③ Project nameに名前を入力します。

③ Next > をクリックします。

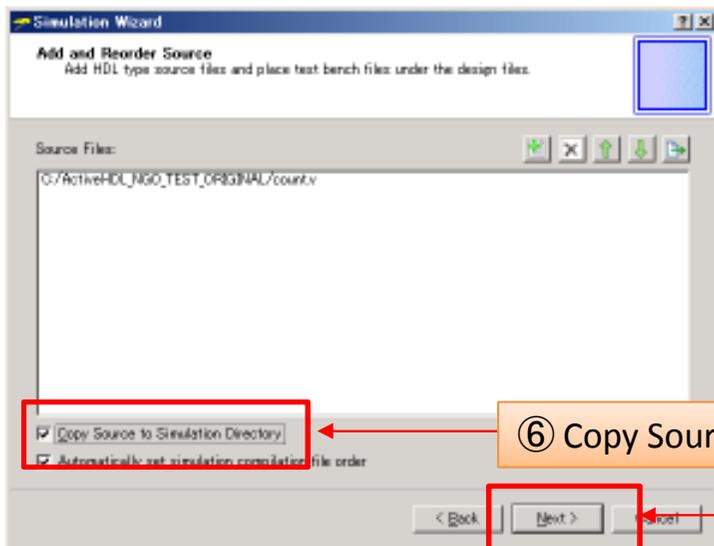


※こちらはDiamondでの設定画面です。



④ RTL をクリックします。

⑤ Next > をクリックします。



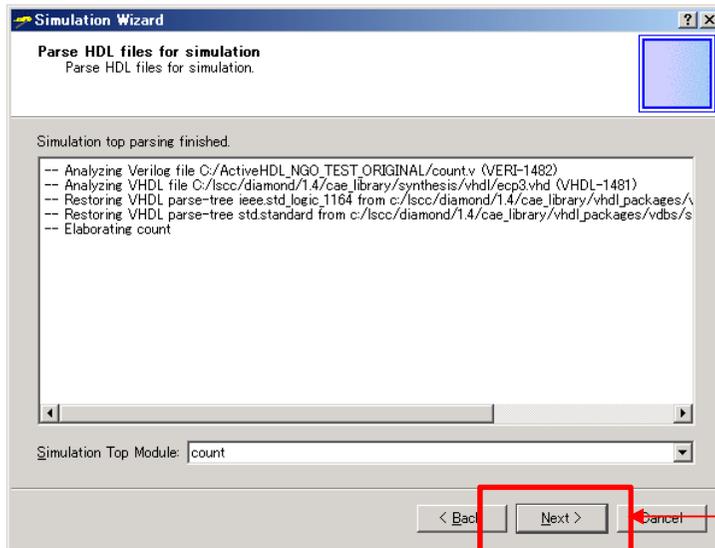
※デザインが読み込まれます。
ngo化する全てのデザインが存在しているか
確認してください。
この時、テストベンチファイルは必要ありません。

⑥ Copy Source Simulation Directory をクリックします。

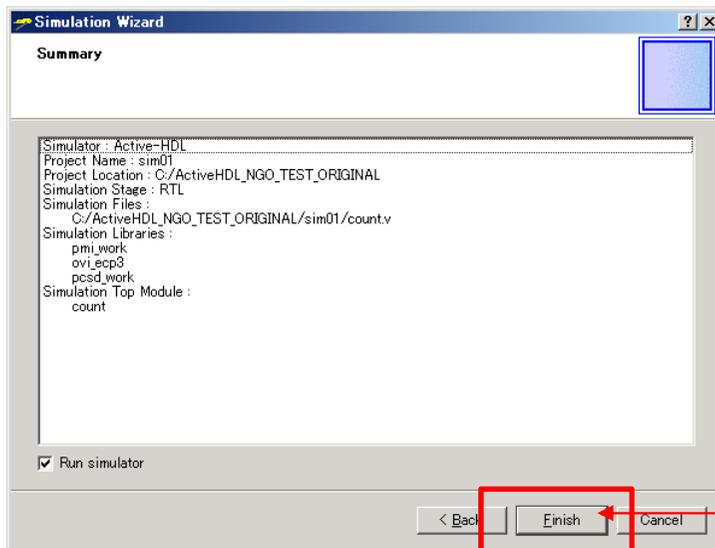
⑦ Next > をクリックします。



※こちらはDiamondでの設定画面です。



⑧ Next> をクリックします。



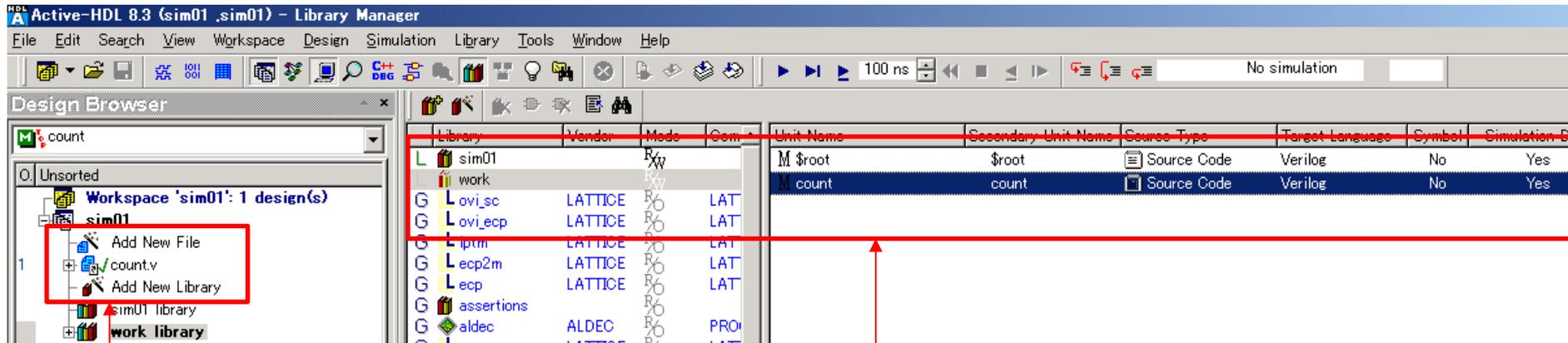
⑨ Finish をクリックします。

※Run Simulatorにチェックが有効な場合
ActiveHDLが自動で起動します。



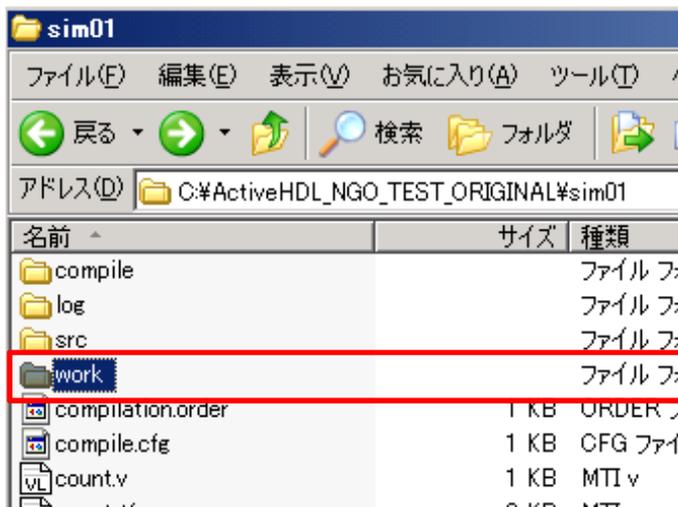


※こちらはActiveHDLでの設定画面です。



※デザインファイル〇〇.vに が付いている場合
コンパイル完了を示します。

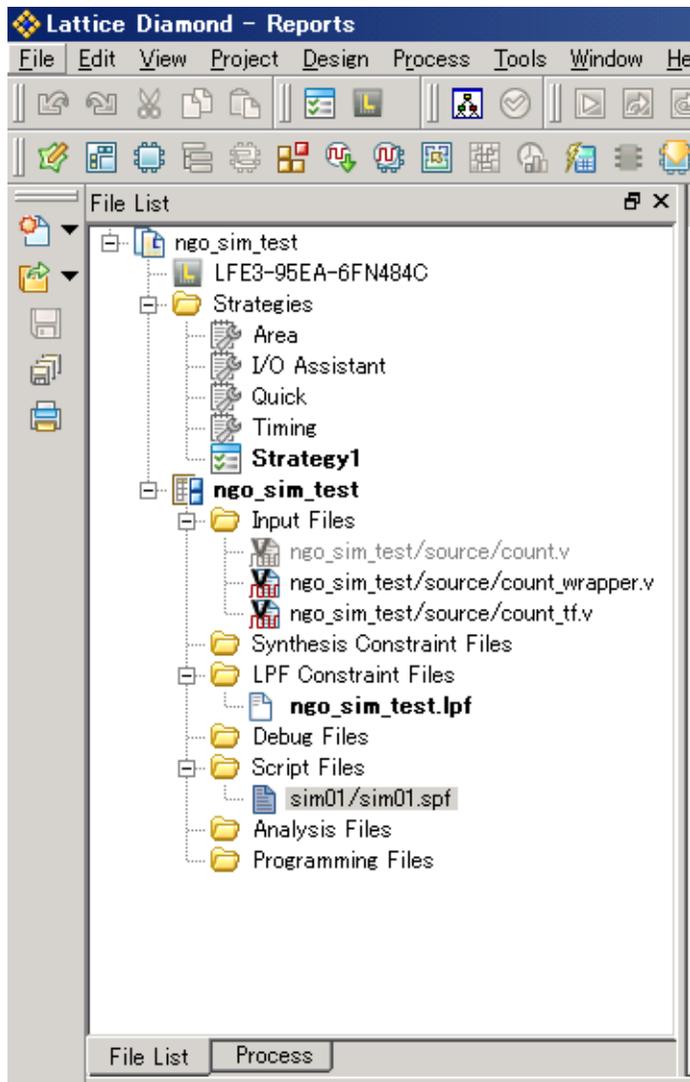
※コンパイルが完了致しますと、
workという名前のLibraryが構成されます。



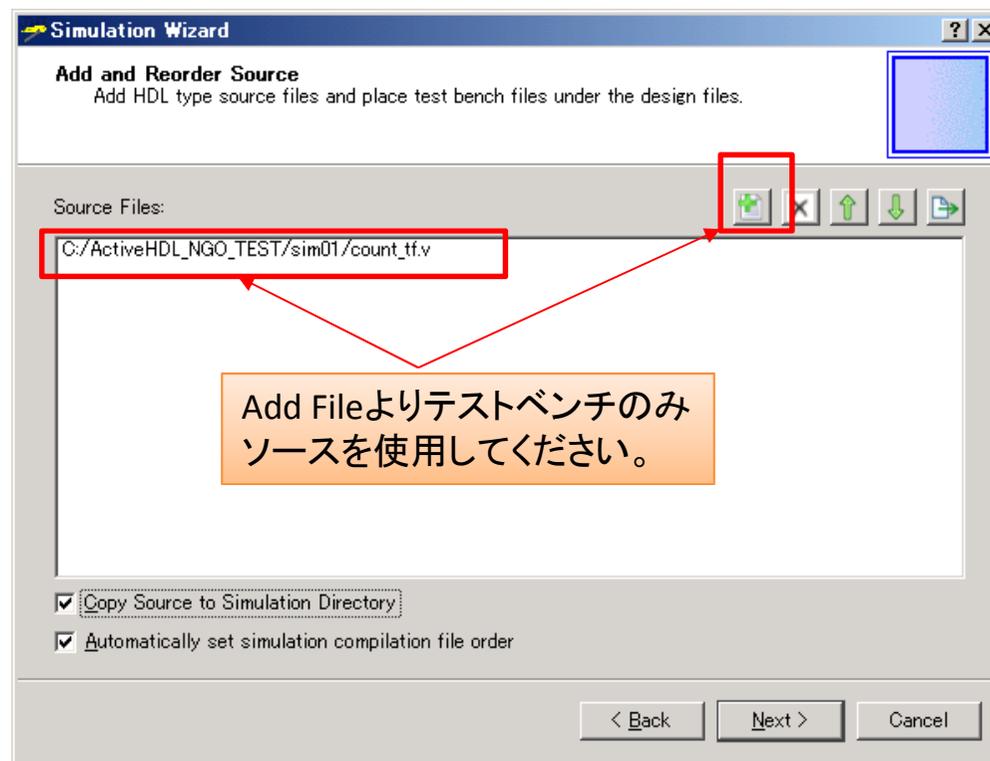
ngoファイルを用いたデザインにおいて
ActiveHDLでの波形シミュレーションを行うためには、
この作成したWorkフォルダを使用します。
こちらをコピーしてください。



次に、ngoファイルを用いたデザインを開きます。
前回と同様に、Simulation WizardでActiveHDLを起動する手順を行います。

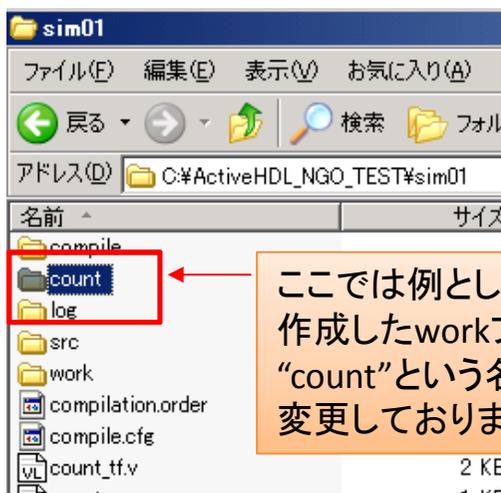


※ 前回とは1箇所以下の作業が必要となります。





Windows上にて、先程生成したWorkフォルダを今回使用するSimulationフォルダに保存します。
 ※ただし、workという名前のフォルダは存在しておりますので、名前を変更して保存します。
 同様に、変更したフォルダ内のwork.libファイル名前を変更致します。

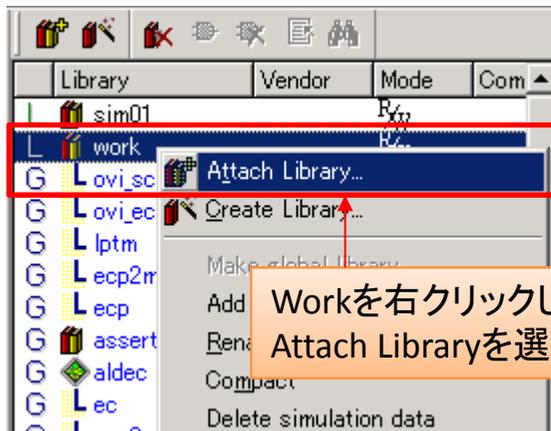


ここでは例として、
作成したworkフォルダを
“count”という名前に
変更しております。

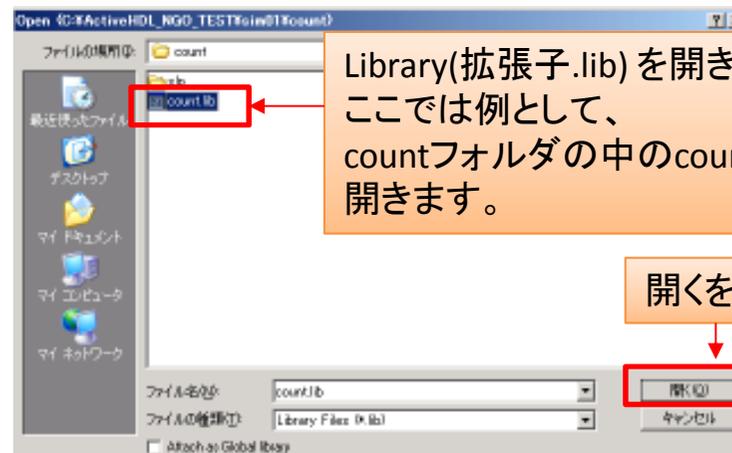


ここでは例として、
作成したcountフォルダ内の
“work.lib”を“count.lib”という名前に
変更しております。

次にActiveHDL画面にて、Libraryに保存したLibraryを読み込みます。



Workを右クリックし、
Attach Libraryを選択します。



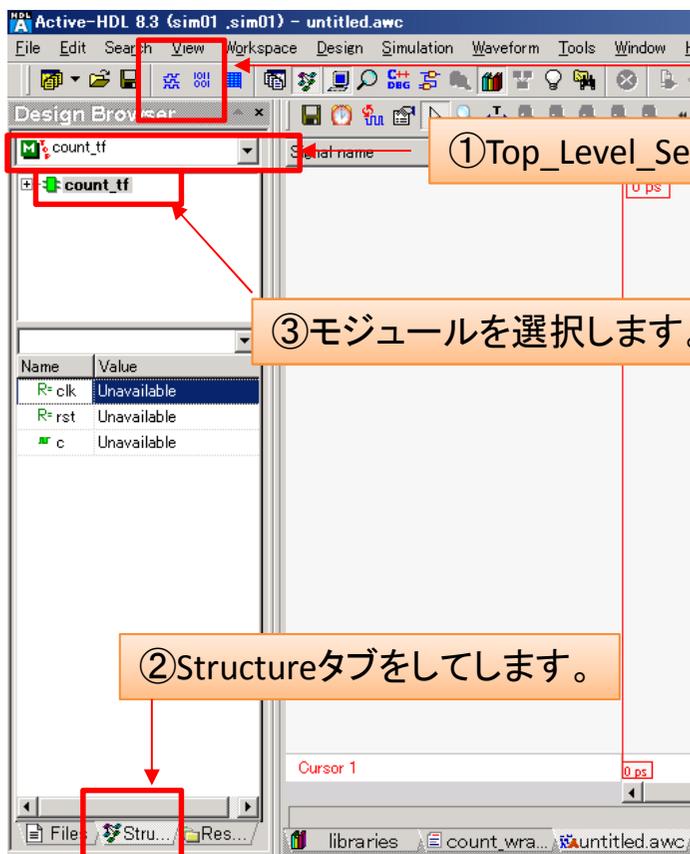
Library(拡張子.lib)を開きます。
ここでは例として、
countフォルダの中のcount.libを
開きます。

開くをクリックします。



ActiveHDL画面にて、以下に示す様にLibraryが読み込まれます。

Library	Vendor	Mode	Com	Unit Name	Secondary Unit Name	Source Type	Target Language	Symbol	Simulation D.
L sim01		R,W		M \$root	\$root	Source Code	Verilog	No	Yes
L count		R,W		M count	count	Source Code	Verilog	No	Yes
L work		R,W							
G L ovi_sc	LATTICE	R,O	LAT						
G L ovi_eop	LATTICE	R,O	LAT						
G L lptm	LATTICE	R,O	LAT						
G L ...	LATTICE	R,L	LAT						



④ New Wave formを選択し、
awcファイルを開きます。

① Top_Level_Selectionをテストベンチファイルを指定します。

③ モジュールを選択します。

② Structureタブをしてします。



The screenshot shows the Active-HDL 8.3 interface. The top toolbar contains various icons, with the 'Run' icon (a blue play button) highlighted by a red box. Below the toolbar is the 'Design Browser' showing a project named 'count_tf'. The 'Console' window at the bottom displays simulation logs, with a red box highlighting the command 'asim -O5 -L ovi.ecp3 -L pcds.work -PL pmi.work +access +r count_tf'. The main workspace shows a signal name 'count_tf' with a value of '0 ps'.

①Runをクリックします。

②Console画面に"asim"で始まる以下の様なコマンドを探し、保存します。

```
asim -O5 -L ovi.ecp3 -L pcds.work -PL pmi.work +access +r count_tf
```



③Console画面の最終行に、先程のコマンドを以下の様に追記して入力します。

```

Console x
# SLP: Finished: 1.5 [s]
# SLP: 0 primitives and 2 (100.00%) other processes in SLP
# SLP: 4 (100.00%) signals in SLP and 0 interface signals
# ELAB2: Elaboration final pass complete - time: 1.5 [s].
# KERNEL: SLP loading done - time: 0.1 [s].
# KERNEL: Warning: You are using the Active-HDL Lattice Edition. The performance of simulation is running at a reduced rate.
# KERNEL: Warning: Contact Aldec for available upgrade options - sales@aldec.com.
# KERNEL: SLP simulation initialization done - time: 0.0 [s].
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 5728 kB (elbread=1023 elab2=4615 kernel=89 sdf=0)
# KERNEL: ASDB file was created in location C:\ActiveHDL_NGO_TEST\sim01\src\wave.asdb
# 11:23, 2012年3月29日
# Simulation has been initialized
# Selected Top-Level: count_tf (count_tf)
run
# KERNEL: Simulation has finished. There are no more test vectors to simulate.
▶ asim -O5 -L ovi_ecp3 -L pcds_work -PL pmi_work -L count +access +r count_tf
    
```

asim -O5 -L ovi_ecp3 -L pcds_work -PL pmi_work -L count +access +r count_tf

④+accessの前に、
-L OO(コピーしたLibrary名)を追加し、Enterで
実行させます。



以上の操作により、ActiveHDL画面にて、ネットリストを用いたシミュレーションが可能となります。

The screenshot shows the Active-HDL 8.3 simulation environment. The Design Browser on the left lists the design 'count_tf' and its signals: 'clk', 'rst', and 'c'. The main window displays a waveform for these signals over time. The 'clk' signal is a periodic square wave. The 'rst' signal is a single pulse. The 'c' signal shows a sequence of values: 'x', '0', '1', '2', '3', '4', '5'. A cursor is positioned at the end of the waveform, and a red box highlights the value '1' in the 'c' signal column.

Signal name	Value
clk	0
rst	0
c	5

※ 正常にシミュレーション出来ていない場合は、出力が全てHi-Z表示となっております。



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。

< お役立ち参考URL >:

■ Lattice 製品サポートページはこちら【pass-word: lattice】

<http://www.tecstar.macnica.co.jp/product/lattice/LatticeSupportHP/index.htm>

■ Lattice日本語ドキュメント

<http://www.latticesemi.co.jp/docs/index.cfm>



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>

