Lattice Radiant Startup Manual

MACNICA

はじめに

- 本マニュアルはLattice社FPGAデバイス設計ツールRadiant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ツールのHelpをご参照頂くか弊社技術サポートまでお問い合わせ下さい
- 本マニュアルとLattice社マニュアルに差異があった場合、Lattice社マニュアルを正として扱うようお願い致します

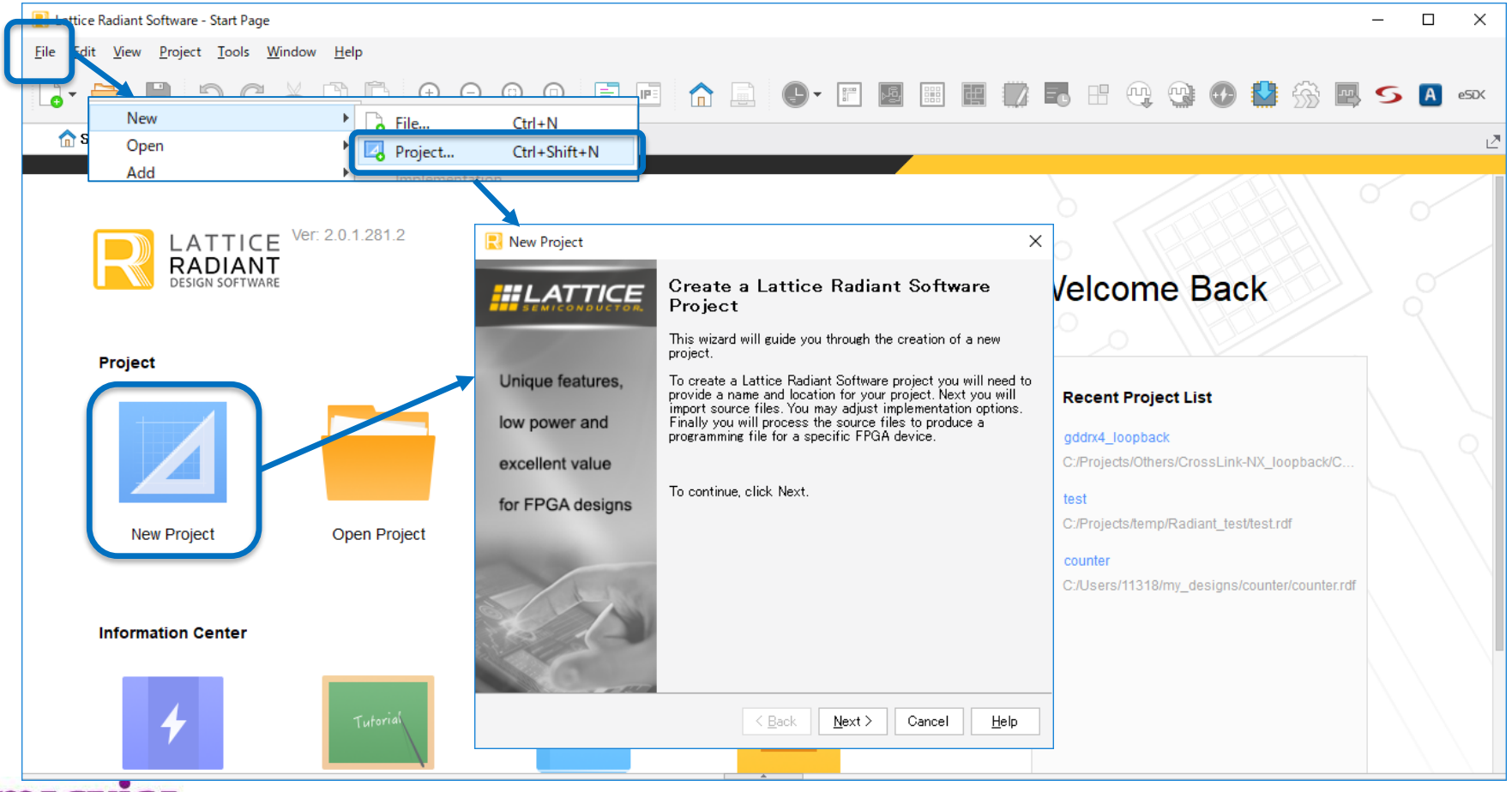
Contents

1. プロジェクト作成～デザインエントリー ・・・P3
2. タイミング制約の設定 ・・・P24
3. I/Oアサイメント、デバイス全般の設定 ・・・P32
4. デザインコンパイル～レポート確認 ・・・P40
5. 消費電力の見積もり ・・・P60
6. Functionシミュレーション ・・・P67
7. デザイン書き込み ・・・P74
8. 実機上での内部波形確認 ・・・P81

1. プロジェクト作成～デザインエントリー

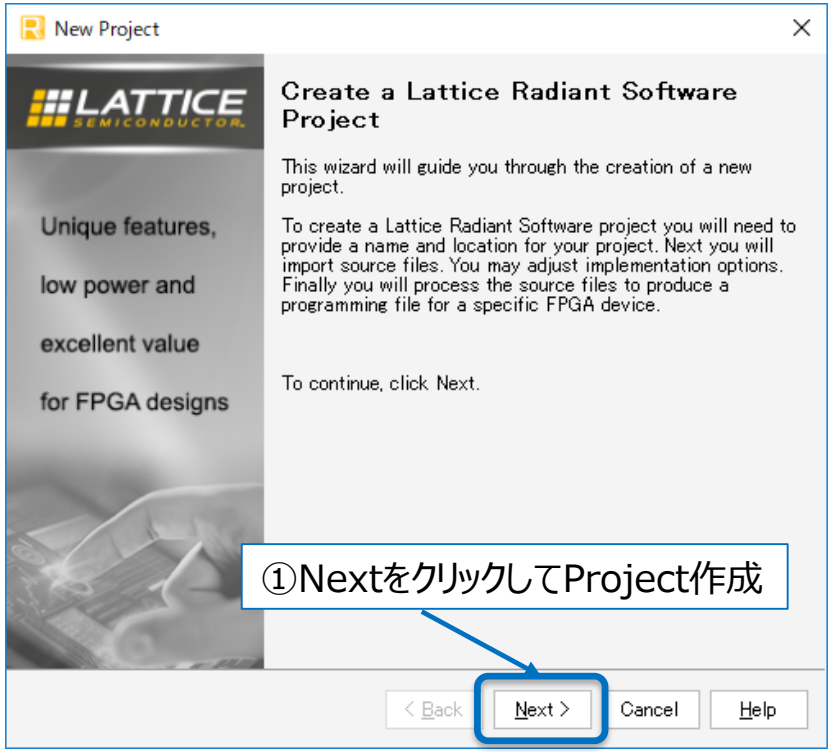
Lattice Radiant Softwareの起動～新規Project作成

- Windowsのスタート > 全てのプログラム> Lattice Radiant Software x.x > Radiant Softwareまたは、デスクトップやタスクバーに配置したショートカットアイコンからRadiantを起動します。
- 画面上のNew Projectアイコンをクリック、またはFile > New > Projectでプロジェクト作成Windowを開きます。



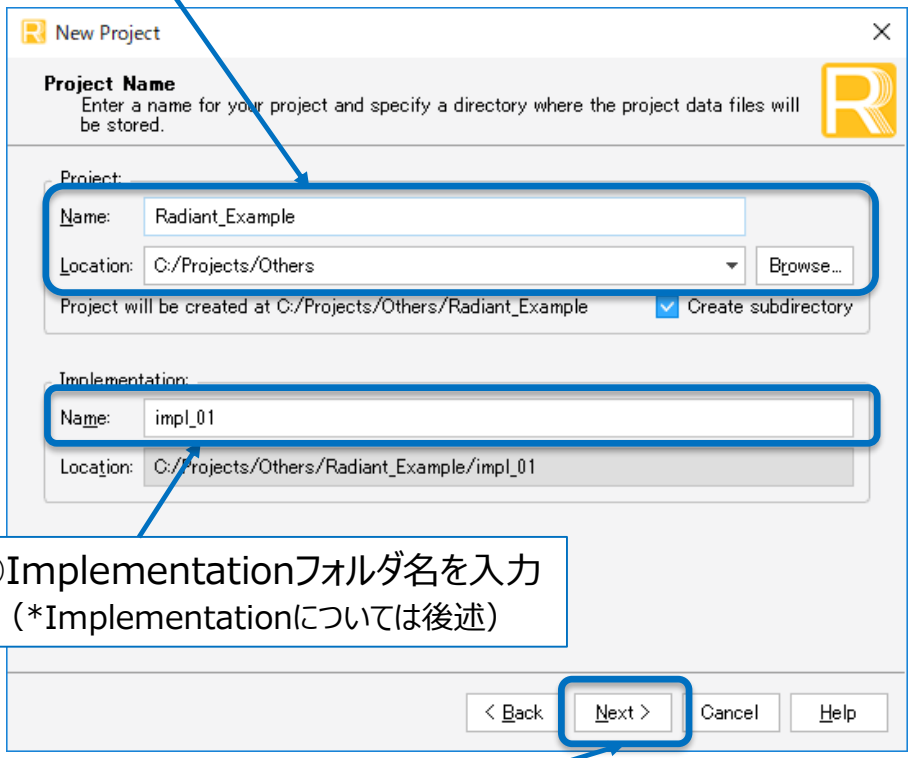
Lattice Radiant Softwareの起動～新規Project作成

- 以下のフローに従い新規プロジェクトを作成します
- フォルダ名は半角英数のみで入力し、フォルダパスに全角やスペースが入らないようにして下さい



①Nextをクリックして Project作成

②Projectフォルダ名とProjectフォルダのロケーションを入力
ロケーションはBrowseで選択することが可能です



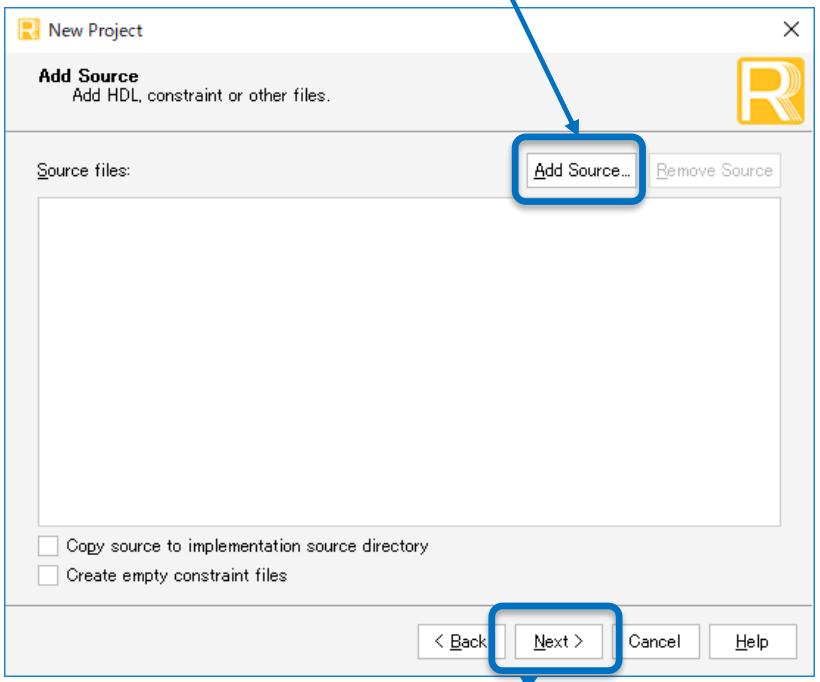
③Implementationフォルダ名を入力
（*Implementationについては後述）

④Nextをクリック

Lattice Radiant Softwareの起動～新規Project作成

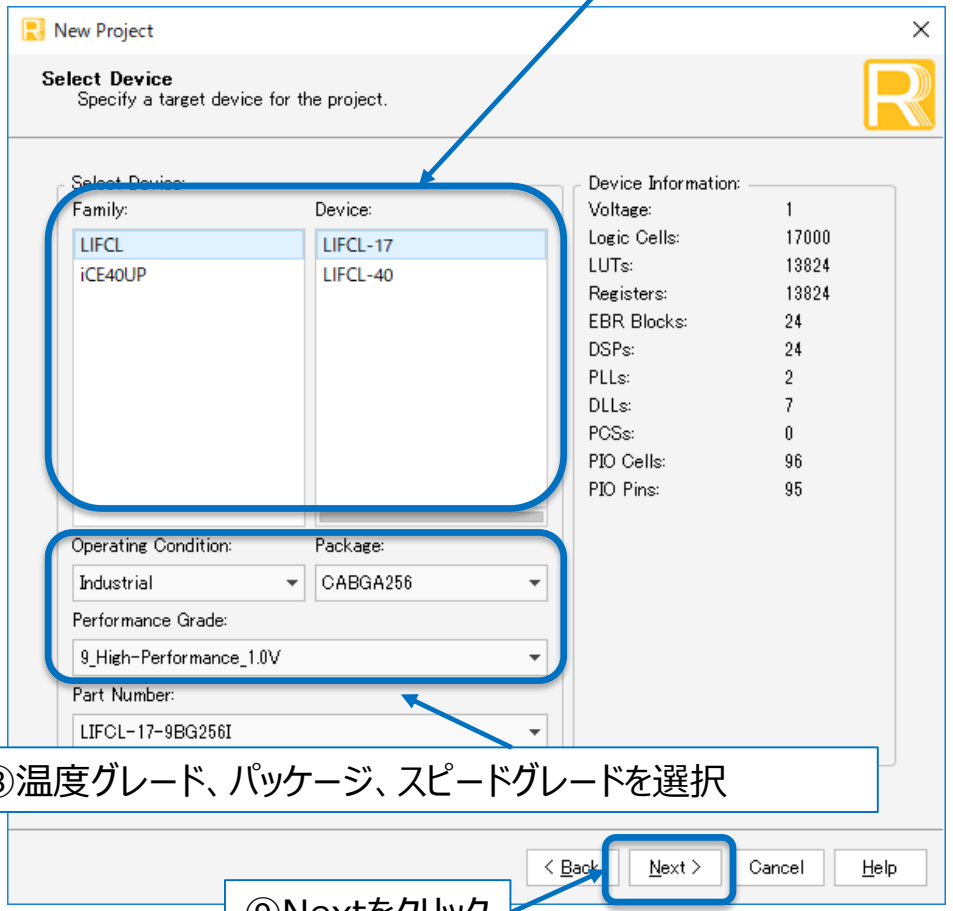
■ 以下のフローに従い新規プロジェクトを作成します

⑤プロジェクトにインポートするソースファイルを選択
（プロジェクト作成後でもインポート可能です）



⑥Nextをクリック

⑦設計対象のデバイスファミリー、デバイス型名を選択

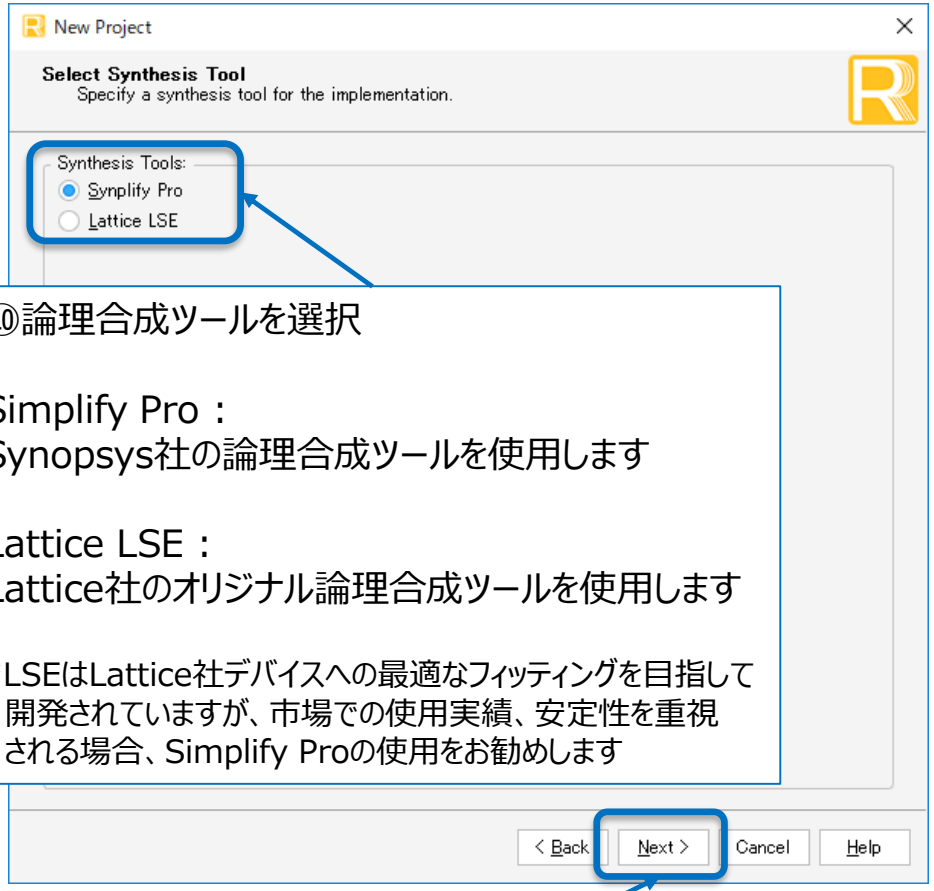


⑧温度グレード、パッケージ、スピードグレードを選択

⑨Nextをクリック

Lattice Radiant Softwareの起動～新規Project作成

■ 以下のフローに従い新規プロジェクトを作成します



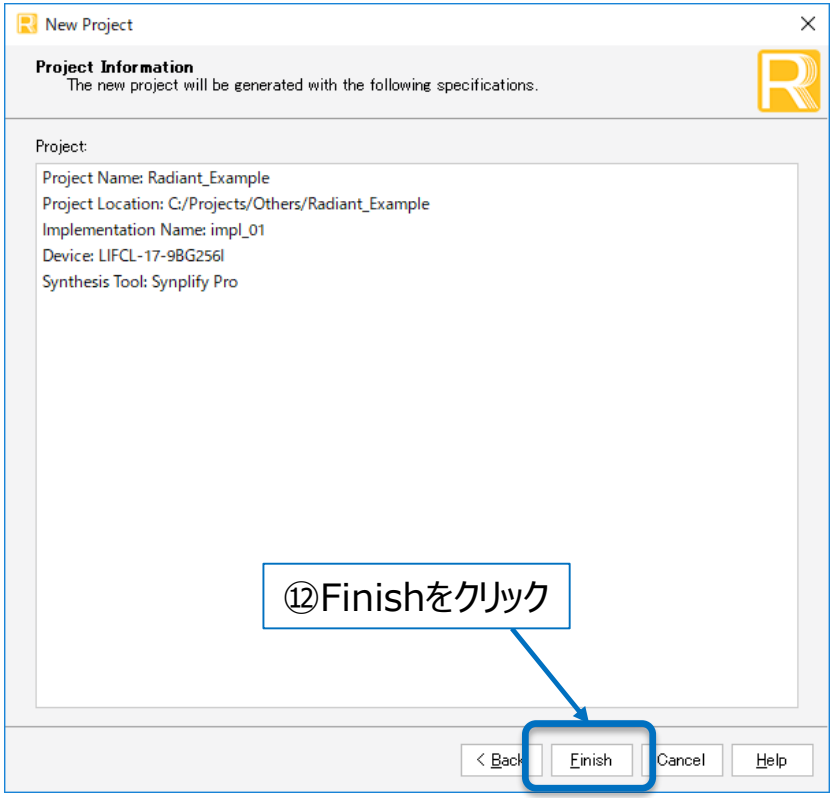
⑩論理合成ツールを選択

Simplify Pro：
Synopsys社の論理合成ツールを使用します

Lattice LSE：
Lattice社のオリジナル論理合成ツールを使用します

*LSEはLattice社デバイスへの最適なフィッティングを目指して開発されていますが、市場での使用実績、安定性を重視される場合、Simplify Proの使用をお勧めします

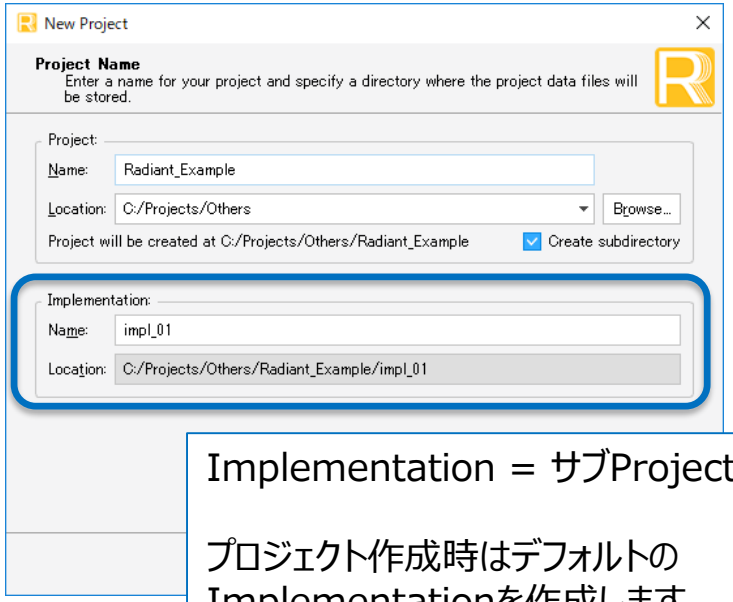
⑪Nextをクリック



⑫Finishをクリック

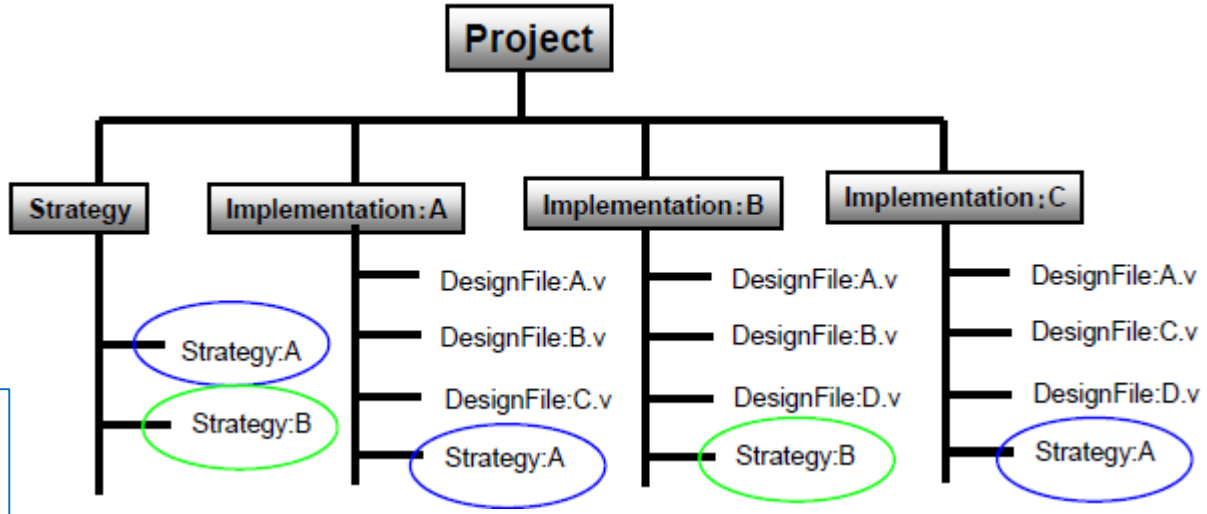
（参考）Implementationについて

- Radiantは作成したProject単位で管理を行います
- Projectの下にさらに複数のサブ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をImplementationと呼びます
- Implementation毎にStrategy（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



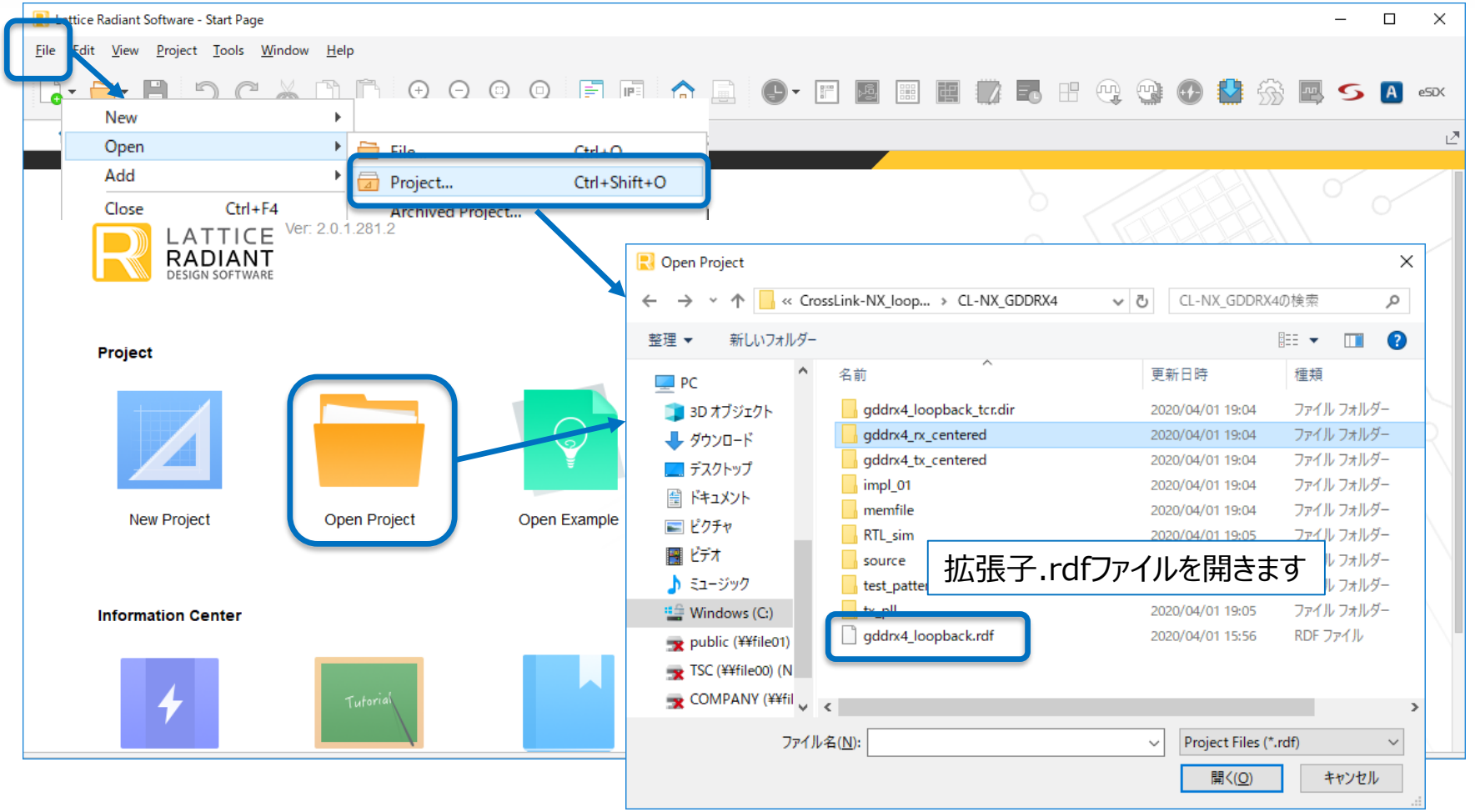
Implementation = サブProject

プロジェクト作成時はデフォルトのImplementationを作成します
作成後に他のImplementationを追加することが可能です



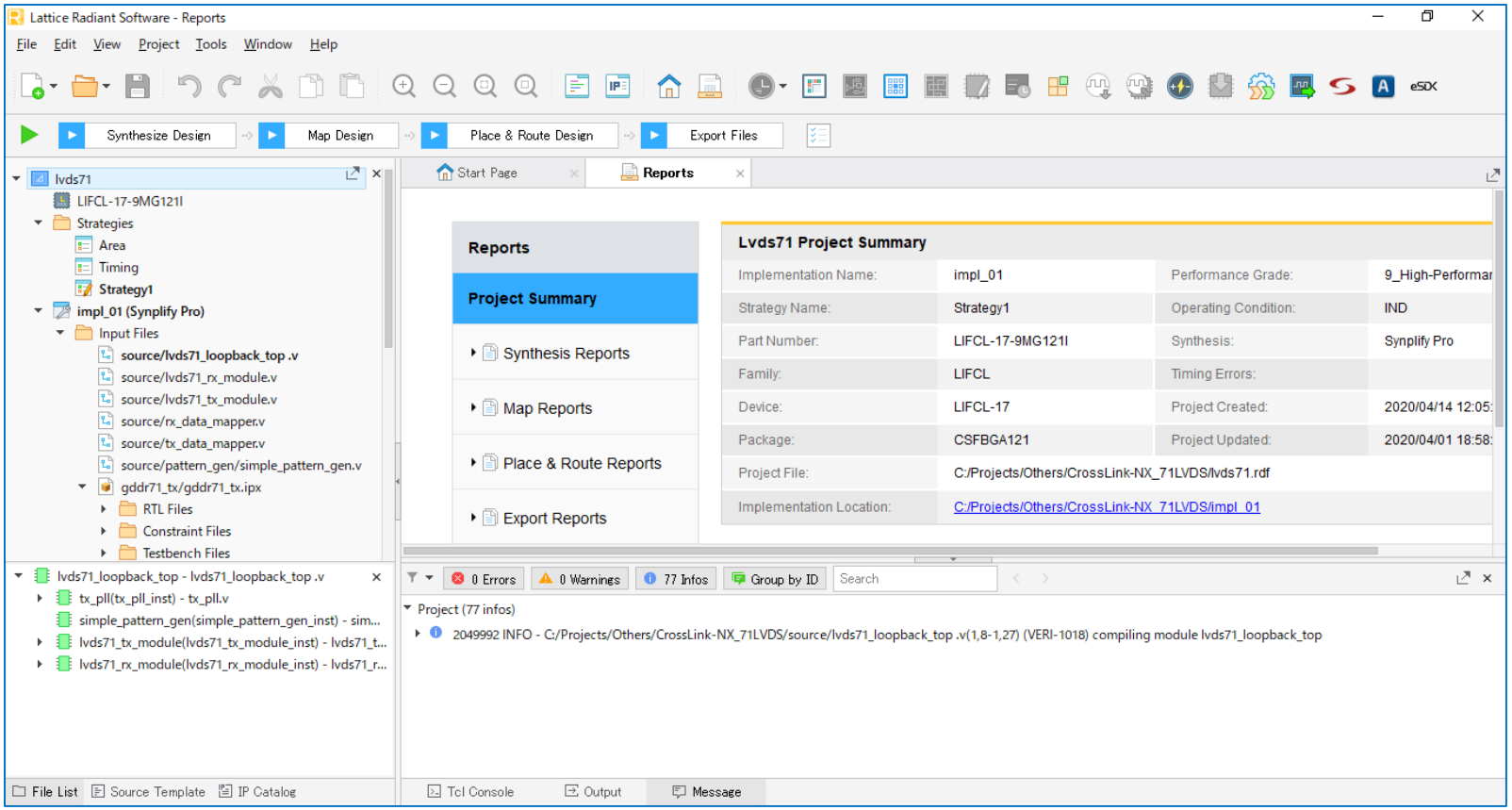
既存プロジェクトのオープン

- 画面上のOpen Projectアイコン をクリック、またはFile > Open > Projectで既存プロジェクトの拡張子.rdfファイルを開きます



Project Navigator

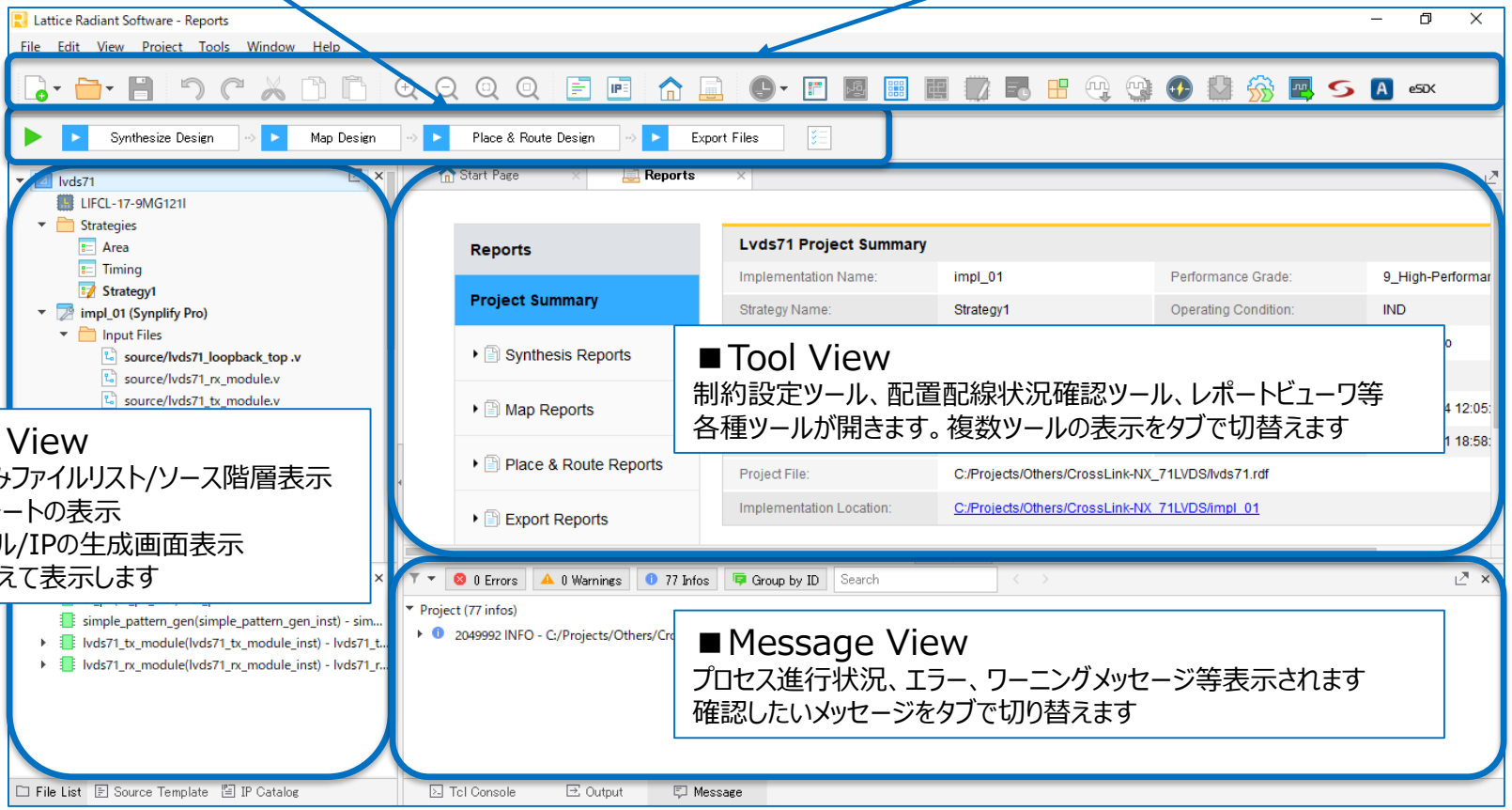
- プロジェクトを新規作成、または既存プロジェクトをオープンすると以下のProject Navigatorの GUIが開きます



Project Navigator

■Process Toolbar
論理合成、マッピング、配置配線、ファイル出力のプロセスはここで実行します

■Toolbar
ファイル追加、制約設定、消費電力見積もり、シミュレーション等を実行する各種ツールのアイコン



■Project View
・インポート済みファイルリスト/ソース階層表示
・ソーステンプレートの表示
・専用モジュール/IPの生成画面表示
をタブで切り替えて表示します

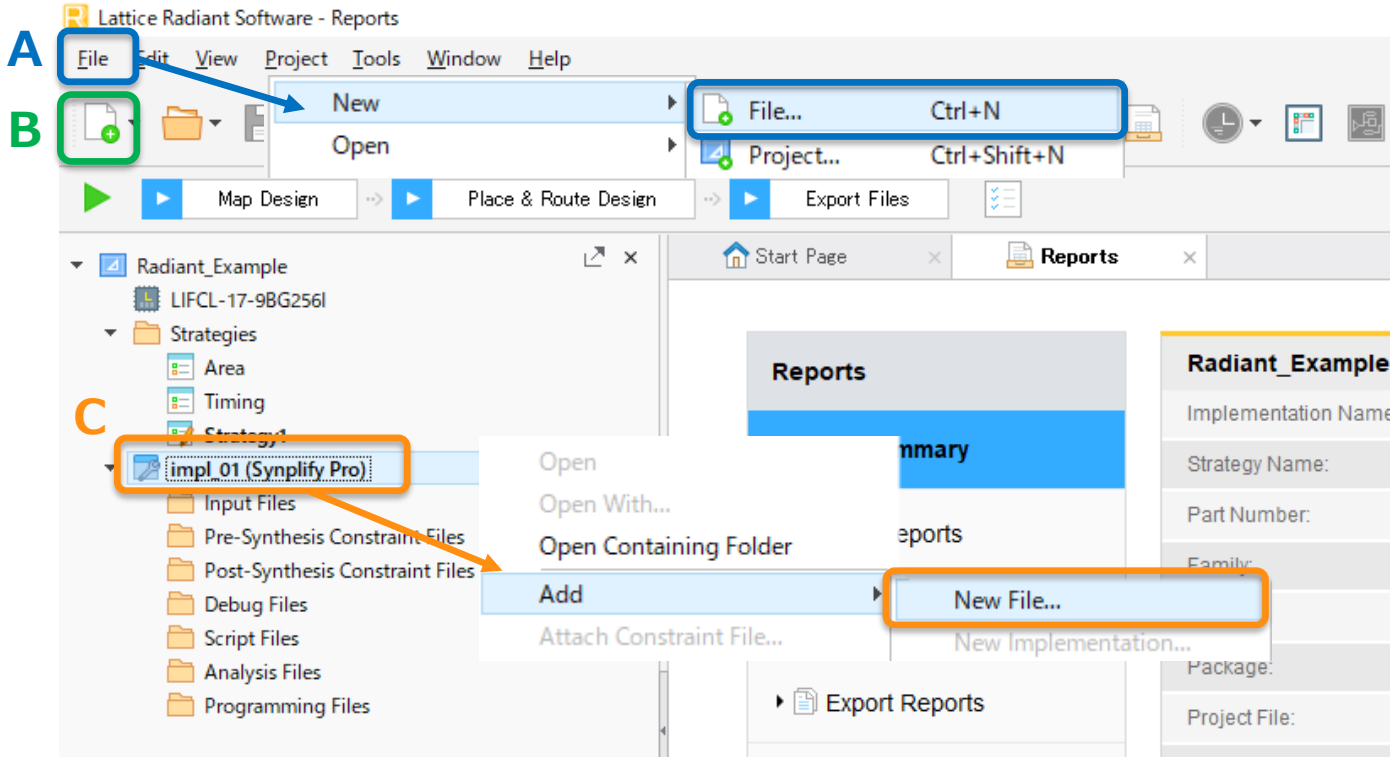
■Tool View
制約設定ツール、配置配線状況確認ツール、レポートビューワ等各種ツールが開きます。複数ツールの表示をタブで切替えます

■Message View
プロセス進行状況、エラー、ワーニングメッセージ等表示されます
確認したいメッセージをタブで切り替えます

*各ビュー右上のDetach Viewボタン でビューを別ウィンドウに切り離して大きく表示できます

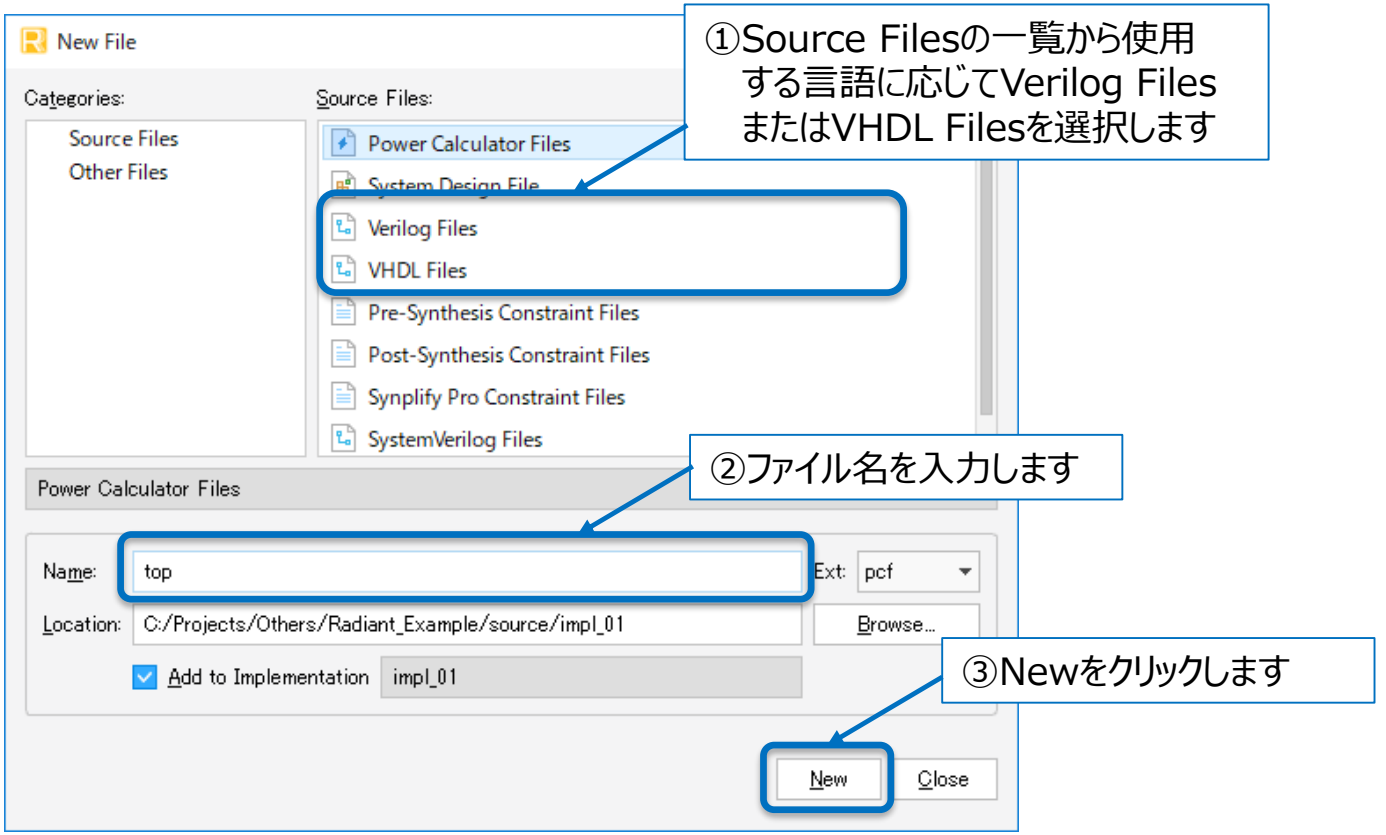
新規デザインエントリー

- New Fileウィンドウを以下のいずれかの方法で開きます
  - A) File > New > File を選択
  - B) Newアイコン をクリック
  - C) Implementation名を右クリック > Add > New File



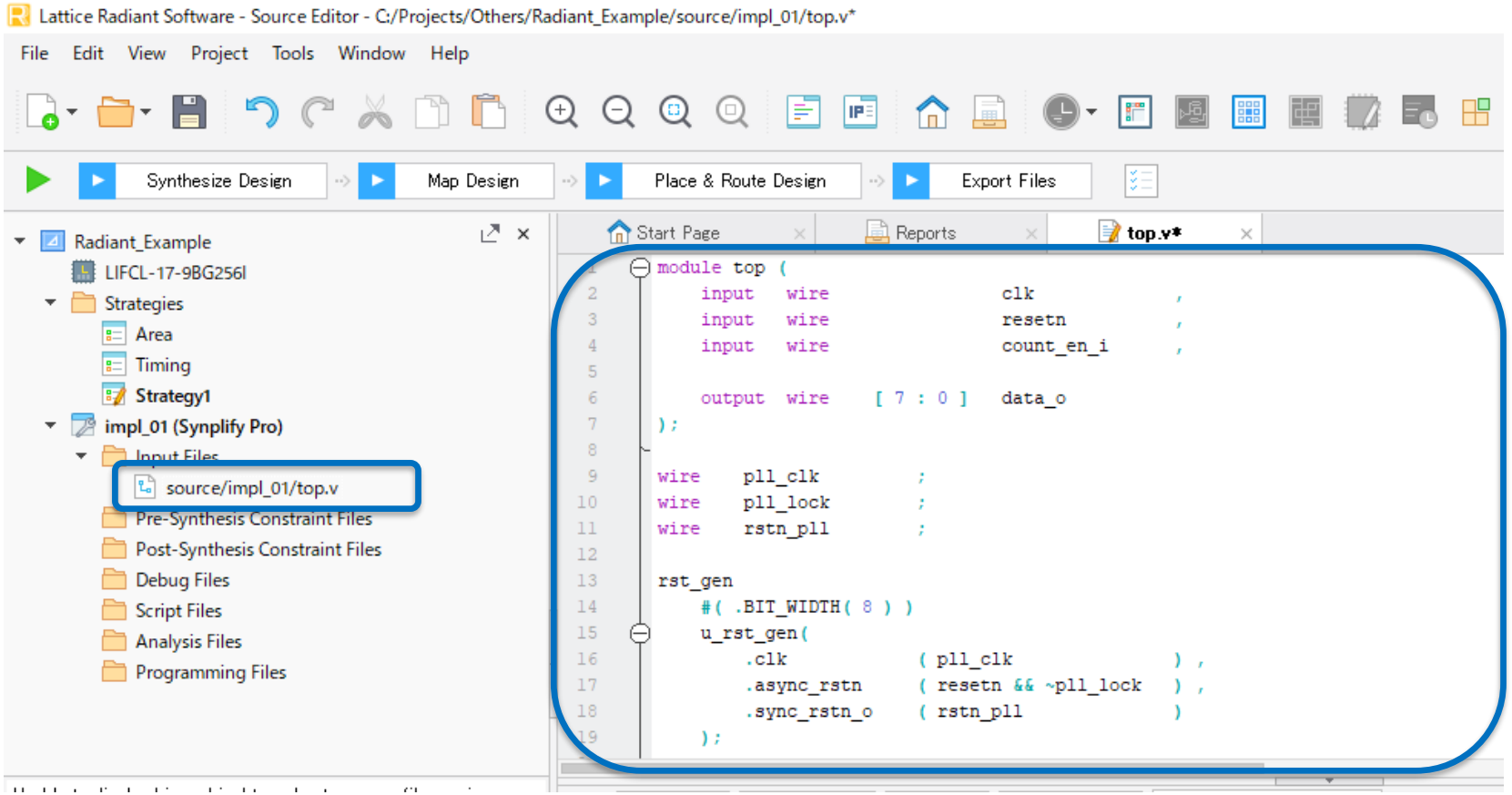
新規デザインエントリー

■ New Fileウィンドウが開いたら以下の手順で新規ソースコードを作成します



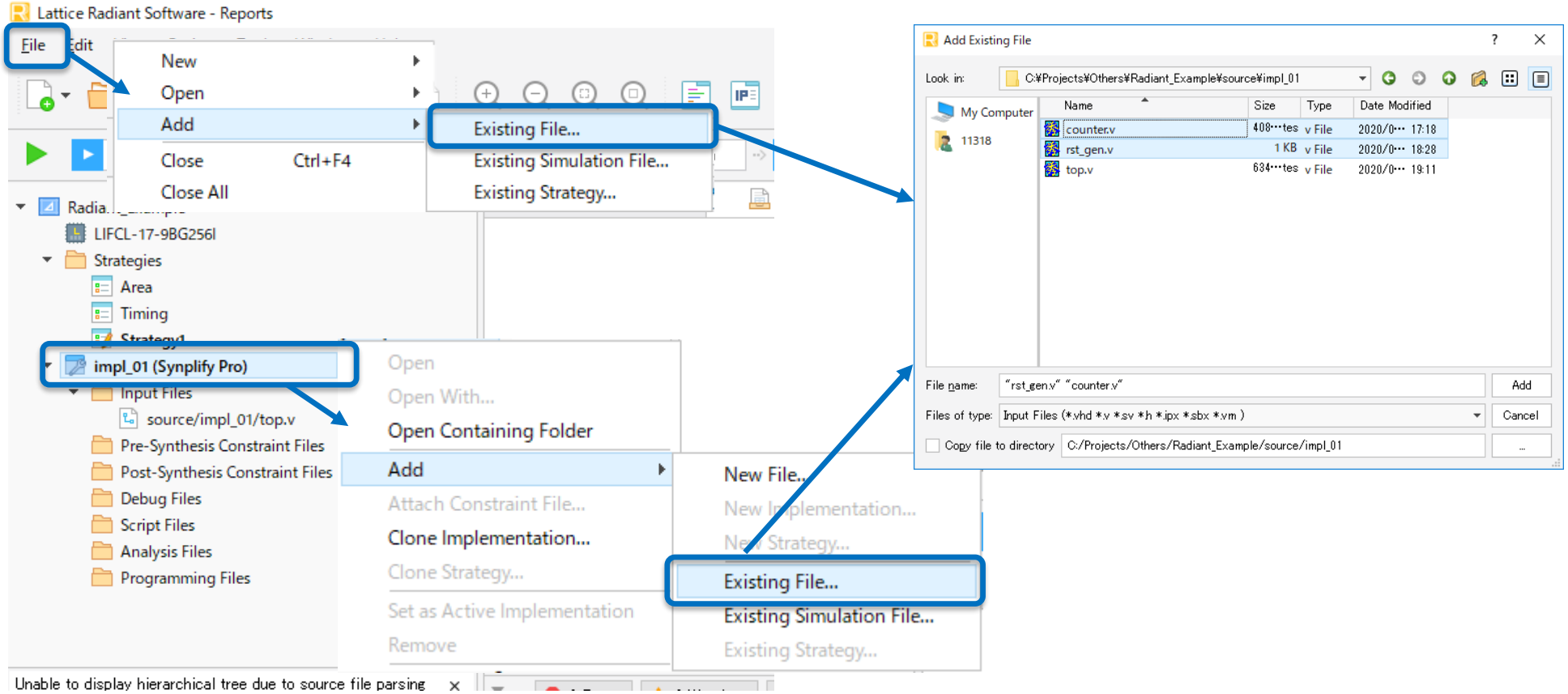
新規デザインエントリー

- 新規ソースコードは自動でプロジェクトにインポートされます
- 同時にSource Editorが開き、そのままソースコード編集が可能です



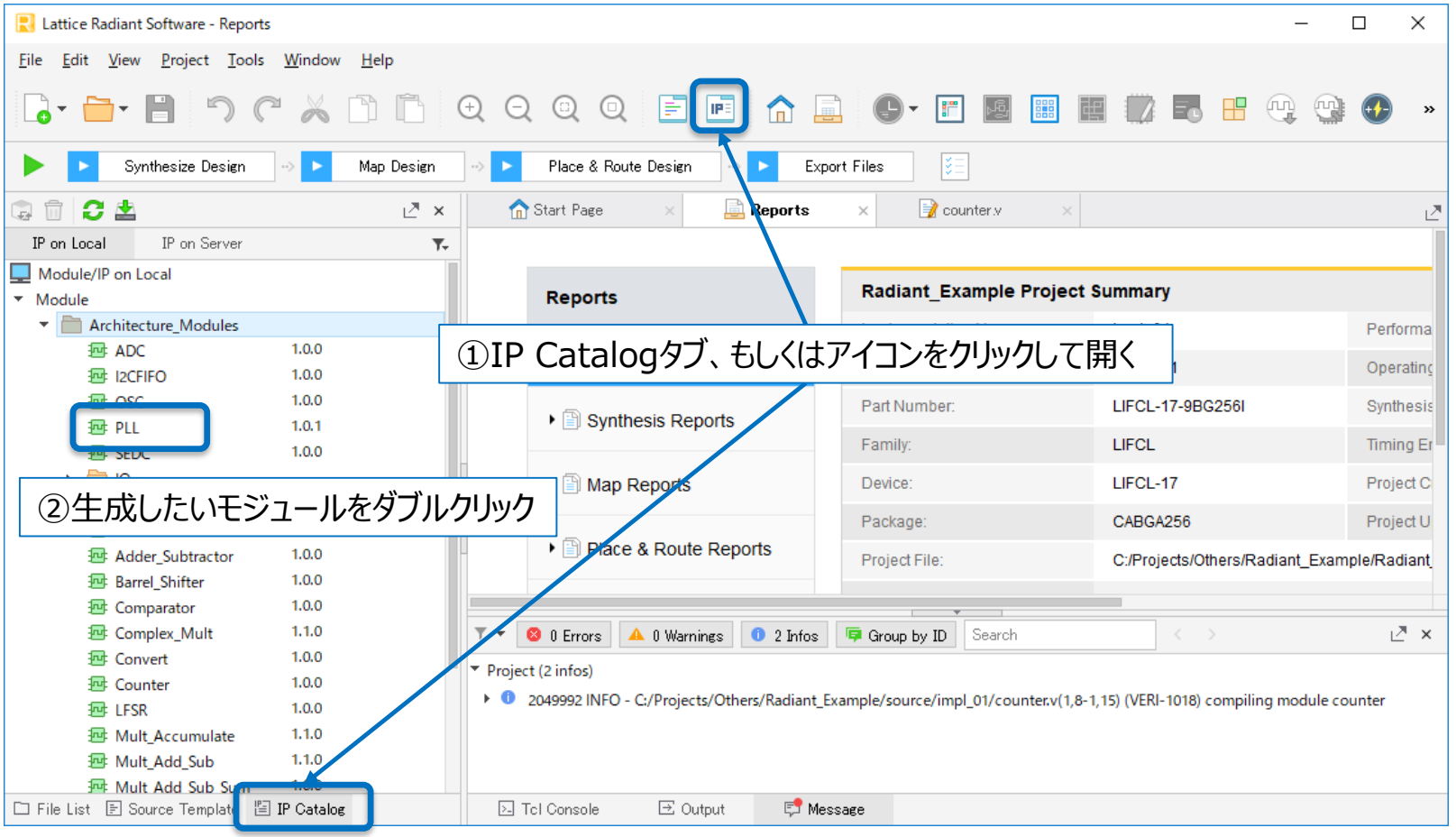
既存デザインエントリー

- 既存のソースコードはFile > Add > Existing Fileもしくは Implementation名を右クリックして Add > Existing Fileを選択します
- Add Existing Fileウィンドウが開きますので、インポートしたいソースコードを選択します



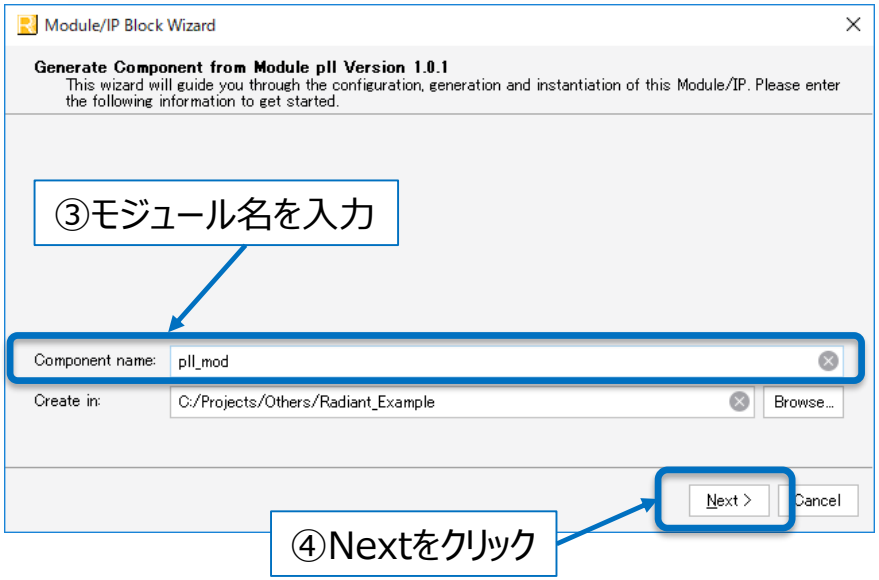
専用モジュールの生成

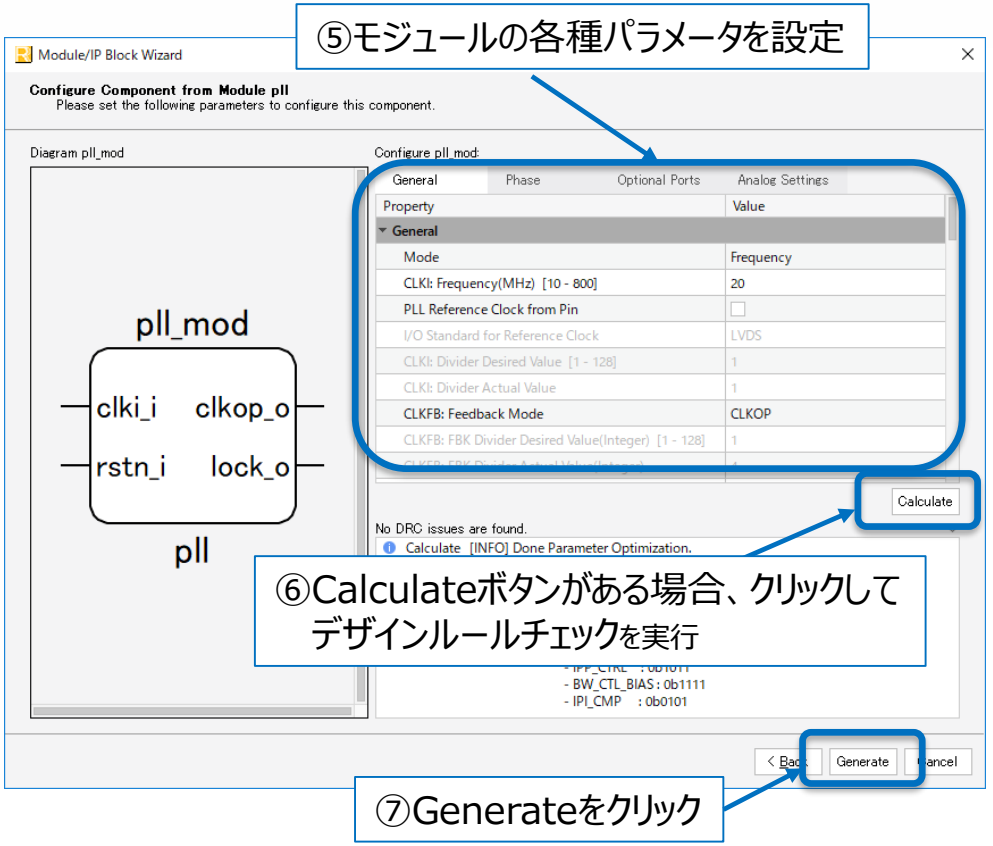
- Project View下部のタブからIP Catalogを選択、もしくはToolbarからIP Catalogアイコン を クリックします
- 生成したいモジュールをダブルクリック（ここでは例としてPLLを選択）



専用モジュールの生成

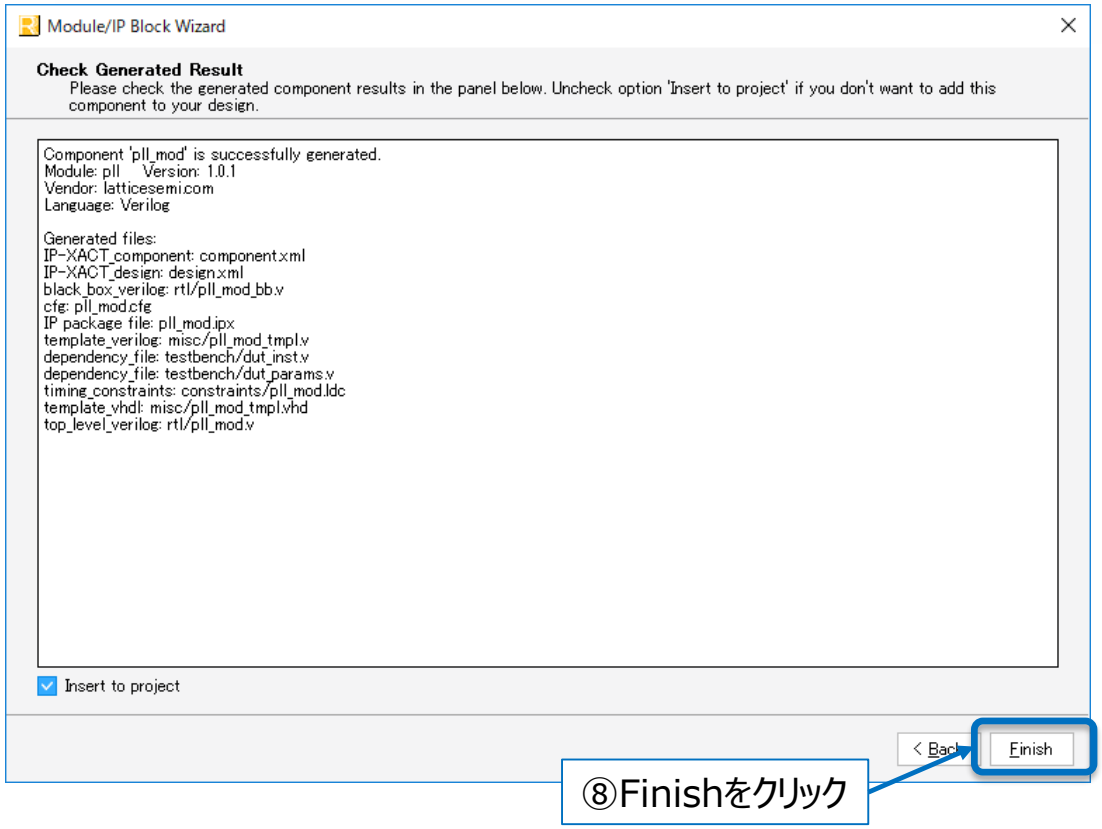
■ Module/IP Block Wizardが開きますので、以下の手順でモジュールを生成します

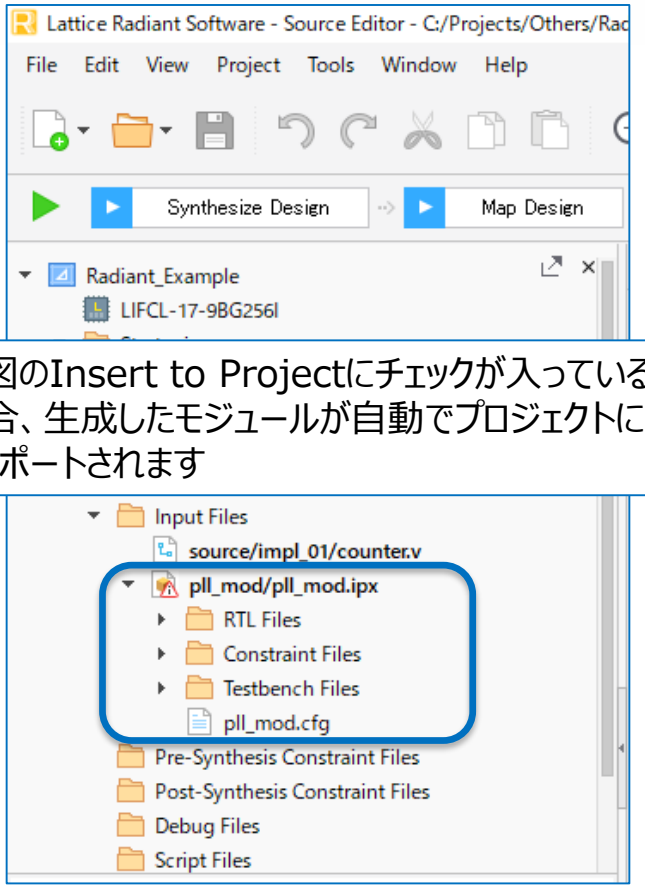




*モジュールのパラメータ内容の詳細に関しては Lattice社HPに掲載されている各デバイスの専用モジュール向けに用意されたApplication NoteやProject Navigator上部のHelpからLattice Radiant Software Helpをご参照下さい

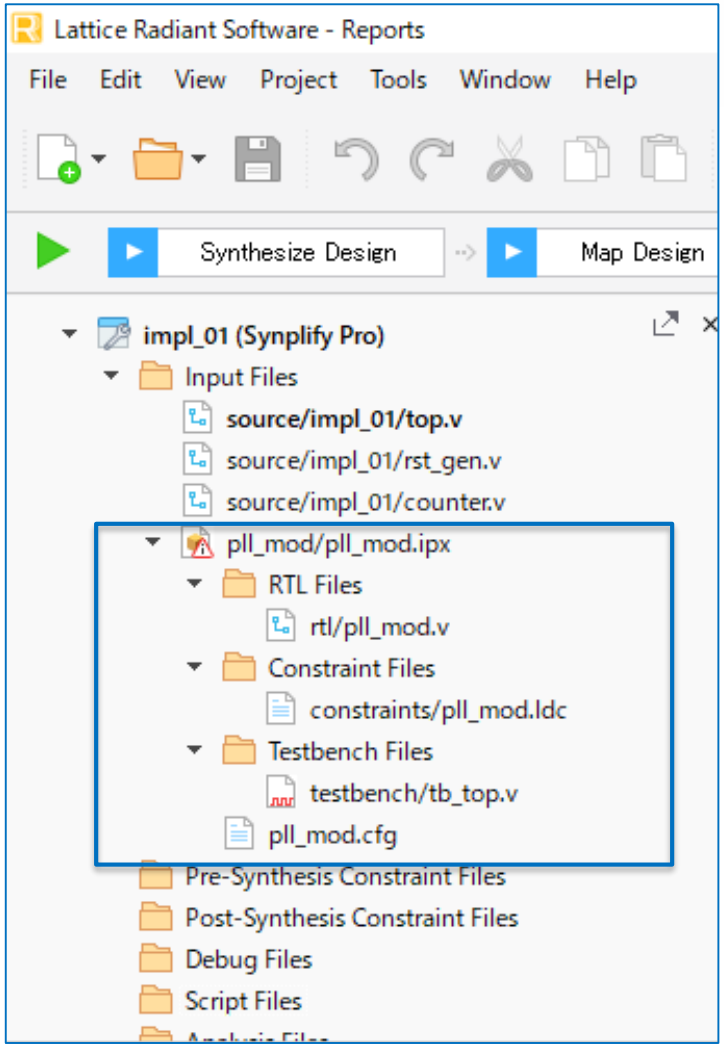
専用モジュールの生成





左図のInsert to Projectにチェックが入っている場合、生成したモジュールが自動でプロジェクトにインポートされます

（参考）専用モジュールのファイル構成



xxx.ipx
→ 生成したモジュールの管理ファイルです。ダブルクリックするとモジュールの編集画面が開きます。このファイルがインポートされている場合のみ以下の構成ファイルがプロジェクトビューに同時表示されます

xxx.v
→ モジュールのソースファイルです。このソースファイルかipxファイルのいずれかがプロジェクトにインポートされている必要があります

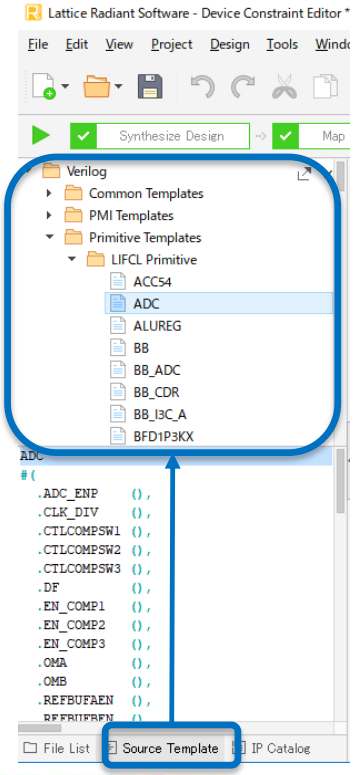
xxx.Ldc
→ モジュールに付帯する制約ファイルです

xxx.tb_top.v
→ 生成したモジュールのシミュレーション用テストベンチファイルです

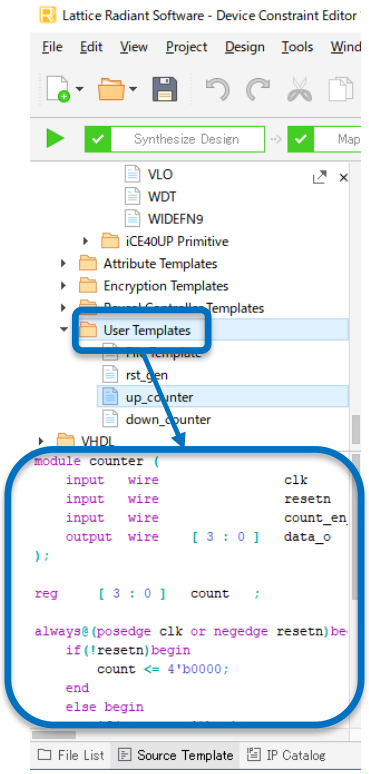
xxx.cfg
→ モジュールのコンフィグレーションファイルです。モジュール構成の概要を示しています

（参考）Source Template

- Project View下部のSource Templateを開くと各デバイスに用意されたコンポーネント（専用モジュールやFF、I/Oバッファ等）のプリミティブ記述を参照したり、ユーザーがよく使うソース記述をテンプレートとして記憶させることができます
- 主ににユーザーのテンプレート記憶用途として使用して下さい。こちらに用意されているプリミティブ記述をソースファイルに記述して使用することは特定の場合を除き基本的に推奨致しません。必要な専用モジュールはIP Catalogから生成して使用して下さい。



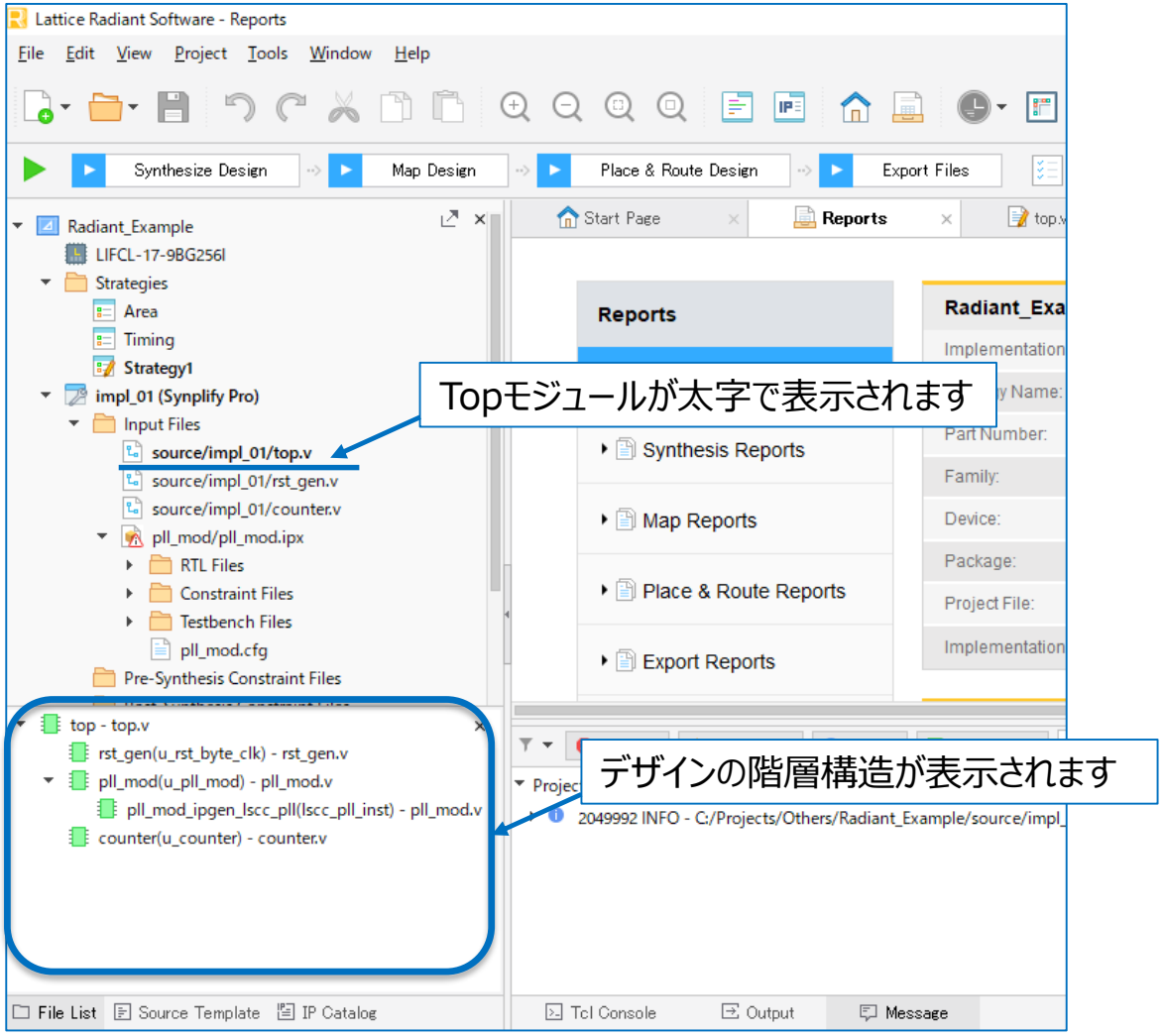
Source Templateからプリミティブ記述を参照できますが、特定用途を除き、基本的に使用を推奨しておりません。専用マクロはIP Catalogから生成して下さい



User Templateに新規テンプレートを追加し、Copy & Pasteで使用することは問題ありません

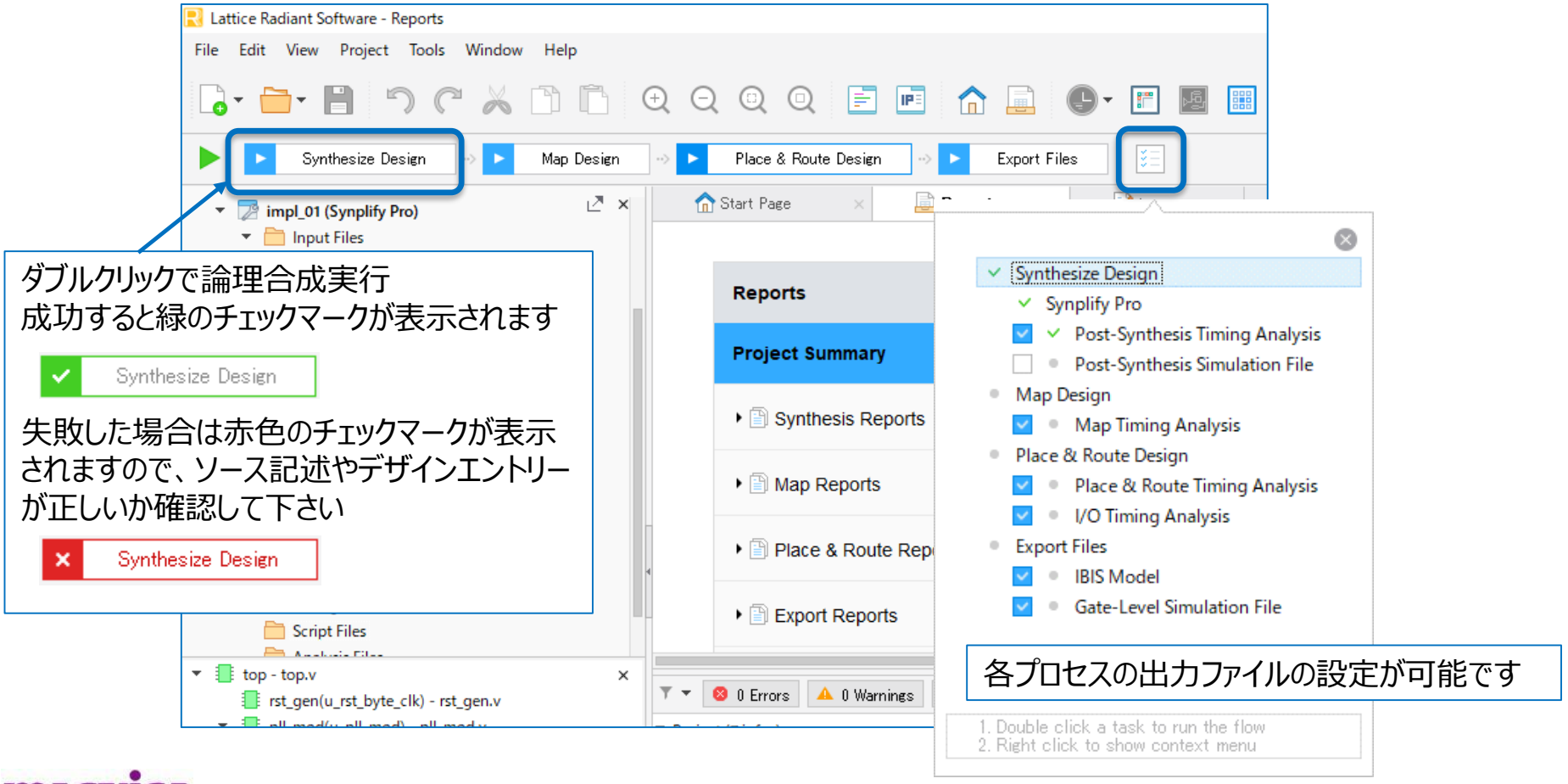
デザインエントリー確認

- デザインエントリーが正しく行われた場合、ツールがTopモジュールを自動認識し、太字で表示します
- 同時にProject View下部のモジュール階層構造が正しく表示されます



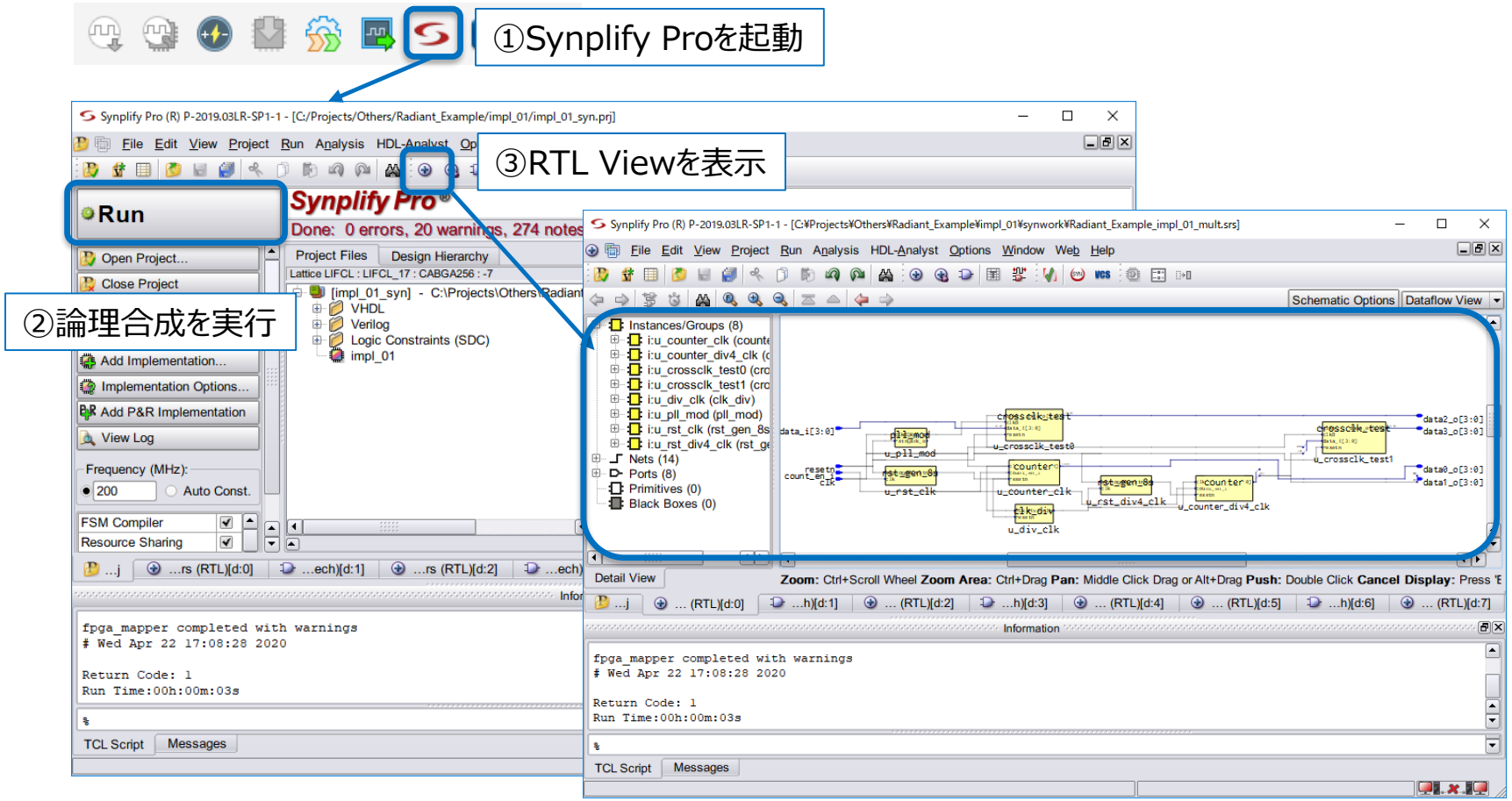
論理合成の実行

- Process ToolbarのSynthesize Designをクリックして論理合成を実行します
- 論理合成が成功すると緑のチェックマークが表示されます
- 各プロセス実行時に出力するファイルはExport Files右側のアイコンをクリックすることで設定可能です



（参考）Synplify ProによるRTL Viewの確認

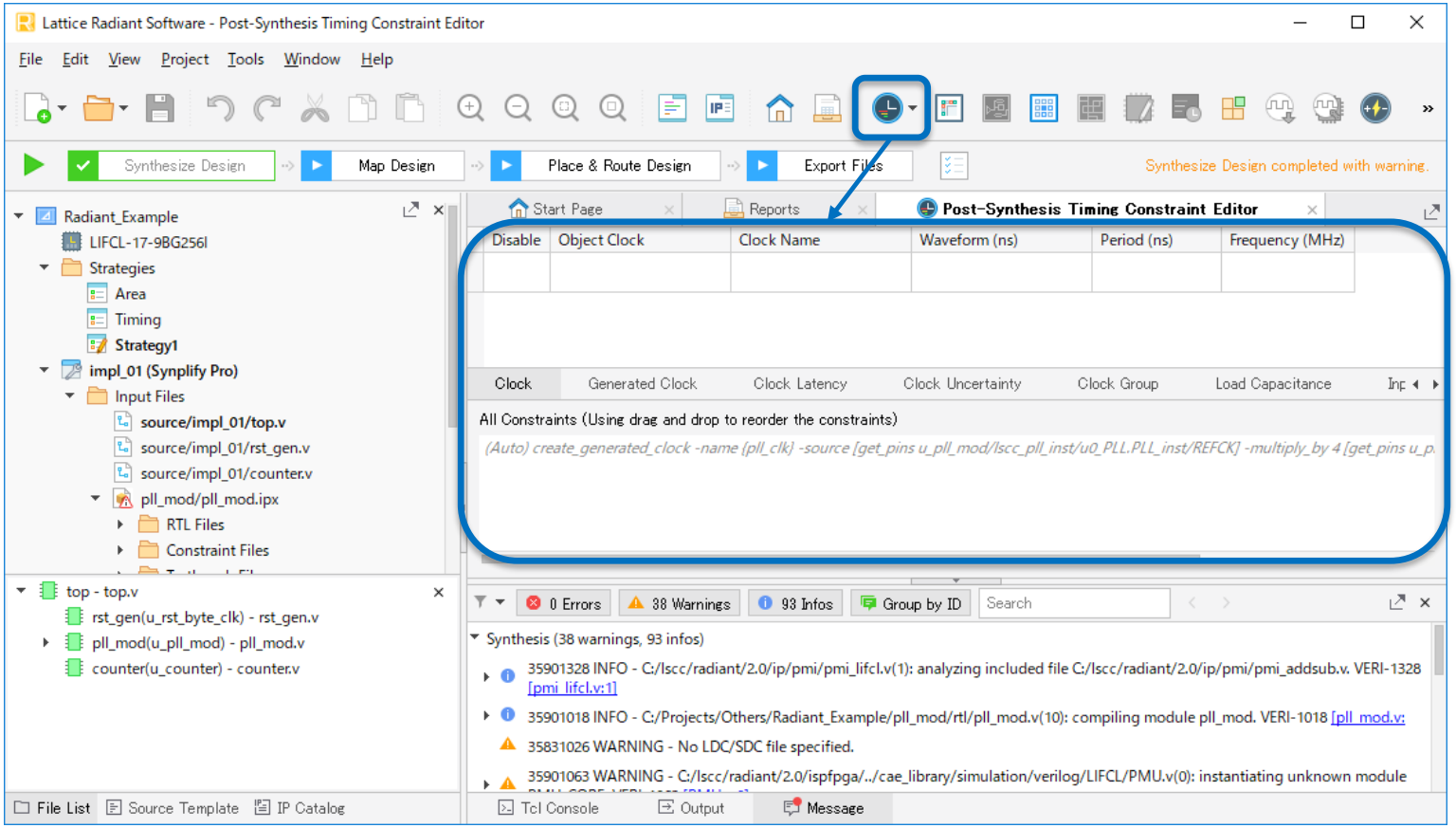
- Synplify Proに付属のRTL Viewで論理合成後の回路がシンボル接続で表示できます
- Process ToolbarのSynplify Proアイコンをクリックして立ち上げ、論理合成を実行します
- Synplify Pro上のRTL Vewアイコンをクリックします



2. タイミング制約の設定

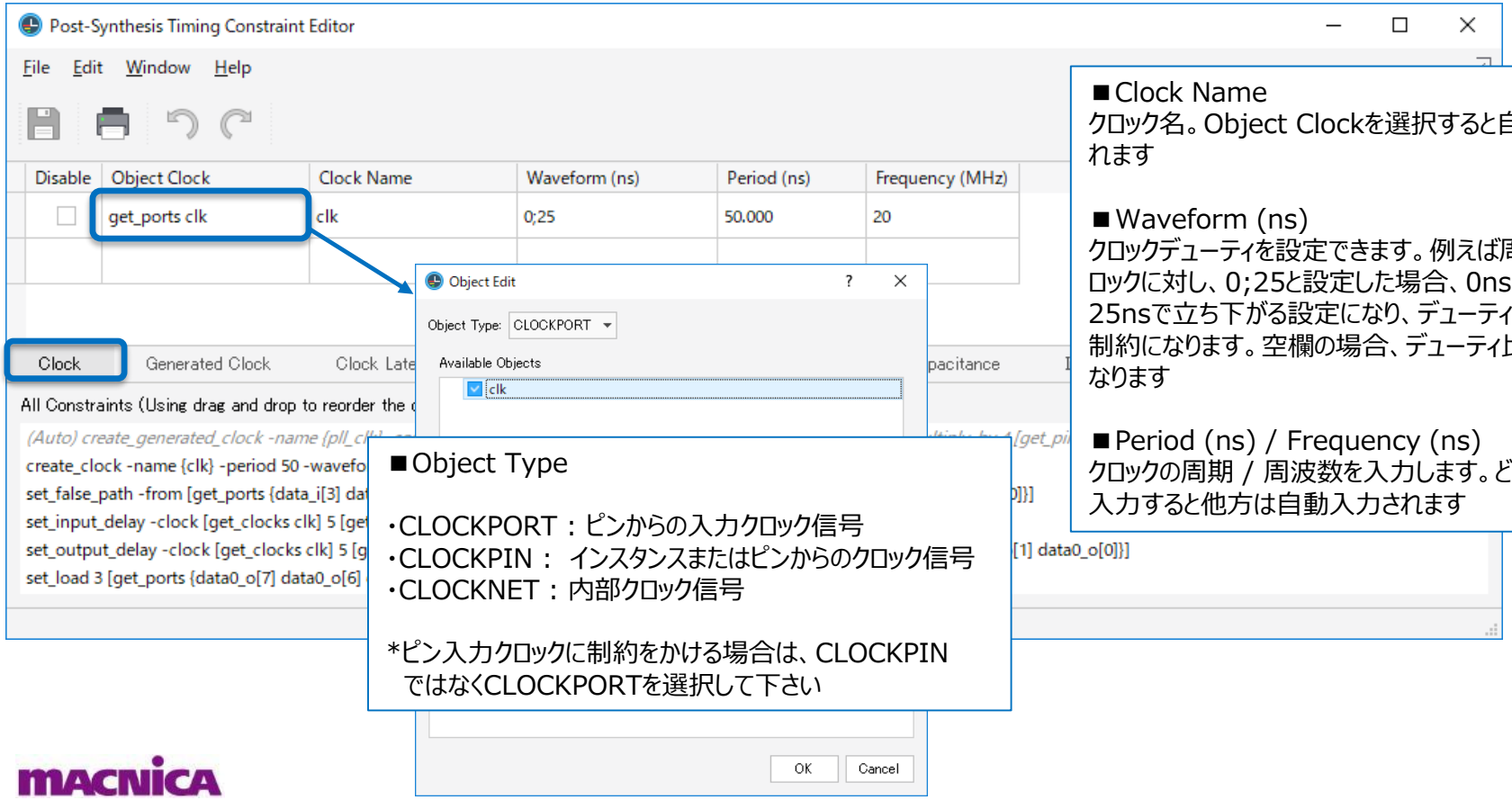
Timing Constraint Editor

- タイミング制約の設定はTiming Constraint Editorで行います
- Synthesize Designが完了している状態でTools > Timing Constraint Editor > Post Synthesis Timing Constraint Editorもしくは、ToolbarのTiming Constraint Editor アイコン からPost Synthesis Timing Constraint Editorを開きます



Fmax（最大動作周波数）制約

- Fmax制約の設定はClockタブもしくはGenerated Clockタブで行います
- Object Clockセルをダブルクリックすると、クロックタイプの選択画面を表示させることができます
- 下図を参照してクロックタイプを選択し、その他のパラメータを設定します
- 制約を削除する場合は右クリックでRemove Rowを選択するか、Disableにチェックを入れることで設定そのものは残したまま制約を外すことができます



■Clock Name
クロック名。Object Clockを選択すると自動で入力されます

■Waveform (ns)
クロックデューティを設定できます。例えば周期50nsのクロックに対し、0;25と設定した場合、0nsで立ち上がり、25nsで立ち下がる設定になり、デューティ比50:50の制約になります。空欄の場合、デューティ比は50:50になります

■Period (ns) / Frequency (ns)
クロックの周期 / 周波数を入力します。どちらか片方を入力すると他方は自動入力されます

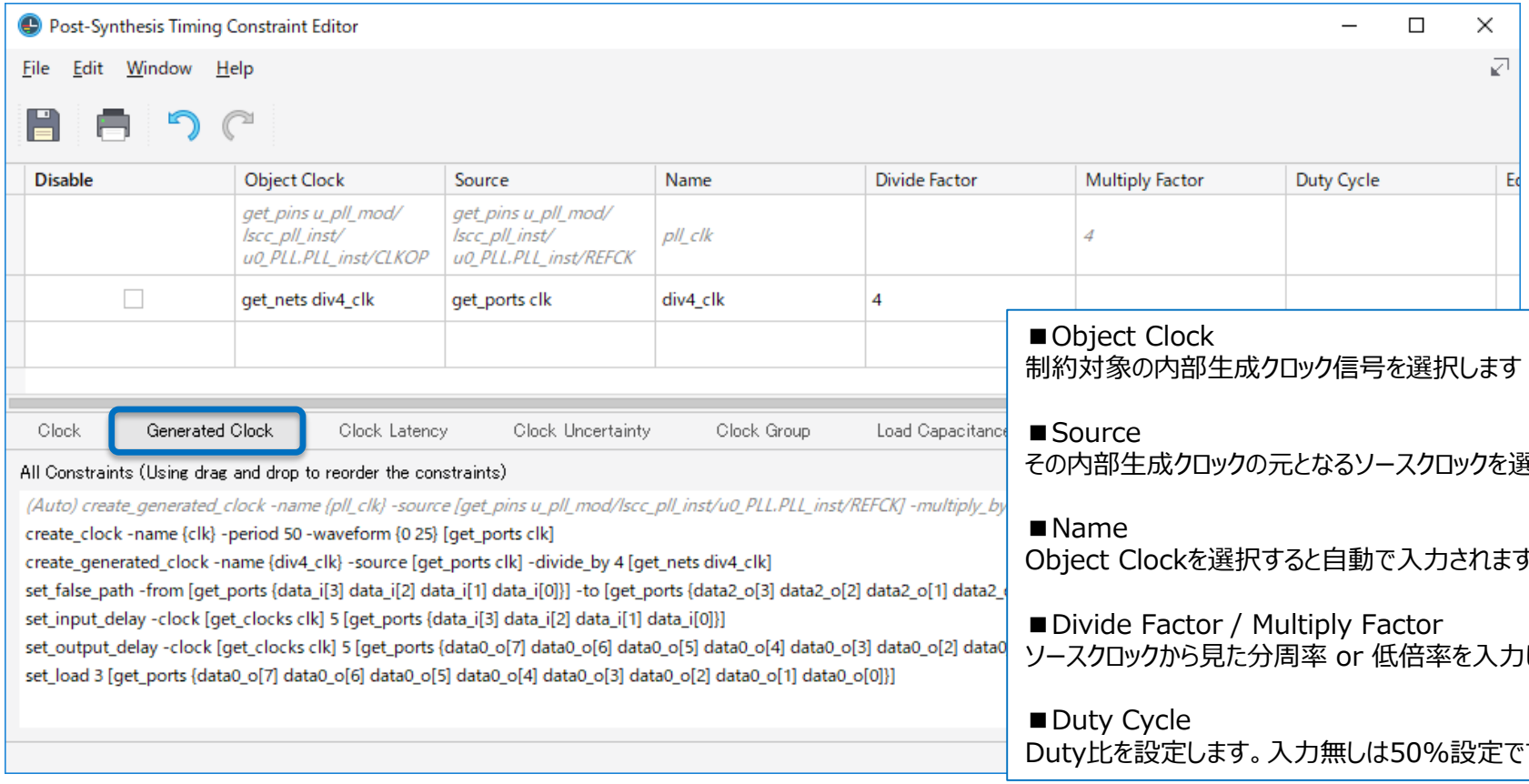
■Object Type

・CLOCKPORT：ピンからの入力クロック信号
・CLOCKPIN： インスタンスまたはピンからのクロック信号
・CLOCKNET：内部クロック信号

*ピン入力クロックに制約をかける場合は、CLOCKPINではなくCLOCKPORTを選択して下さい

Fmax（最大動作周波数）制約

- Generated Clockタブでは内部生成クロックに対して制約を設定できます
- Object Clockセルをダブルクリックすると、クロックタイプの選択画面を表示させることができます
- 内部PLLで生成するクロック等はここに自動で制約が入力されます



■Object Clock
制約対象の内部生成クロック信号を選択します

■Source
その内部生成クロックの元となるソースクロックを選択します

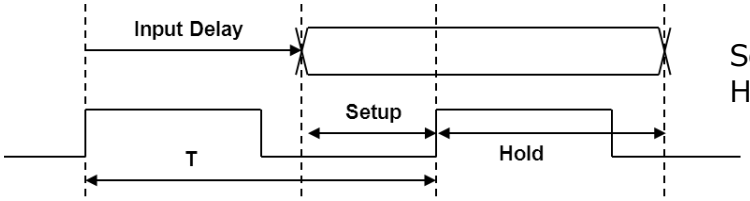
■Name
Object Clockを選択すると自動で入力されます

■Divide Factor / Multiply Factor
ソースクロックから見た分周率 or 低倍率を入力します

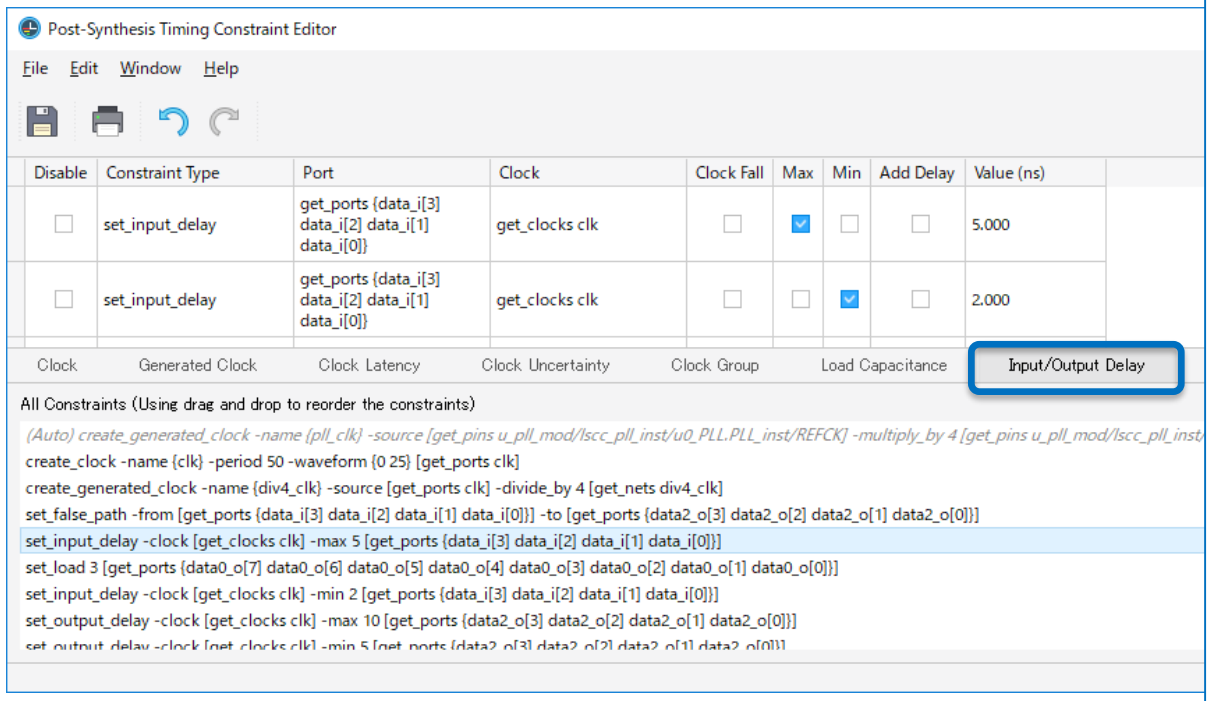
■Duty Cycle
Duty比を設定します。入力無しは50%設定です

Input Delayの設定

- Input Delayとは入力データのクロックに対する遅延設定で、クロック周期からSetup Timeを引いた値、つまり前段デバイスのClock to Output Timeに相当します
- 下図を参照して制約を与えるポートと基準クロック、パラメータを設定します



Setup Time = T - Input Delay
Hold Time = Input Delay



■Constraint Type
ここではInput Delay制約を選択します

■Port
制約対象となるデータ入力ポートを選択します

■Clock
制約の基準クロックを選択します

■Clock Fall
クロックの立下りエッジを基準として制約をかける場合にチェックを入れます

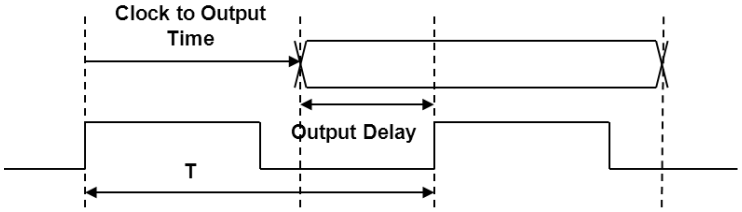
■Max / Min
制約が最大遅延（Max）に対するものか、最小遅延（Min）に対するものかを選択します

■Value
遅延量を設定します。デバイスのMin Setup Timeの値が T - Input Delay(Max)より小さくならないように設定します。また、デバイスのMin Hold TimeがInput Delay (Min)より小さくならないように設定します

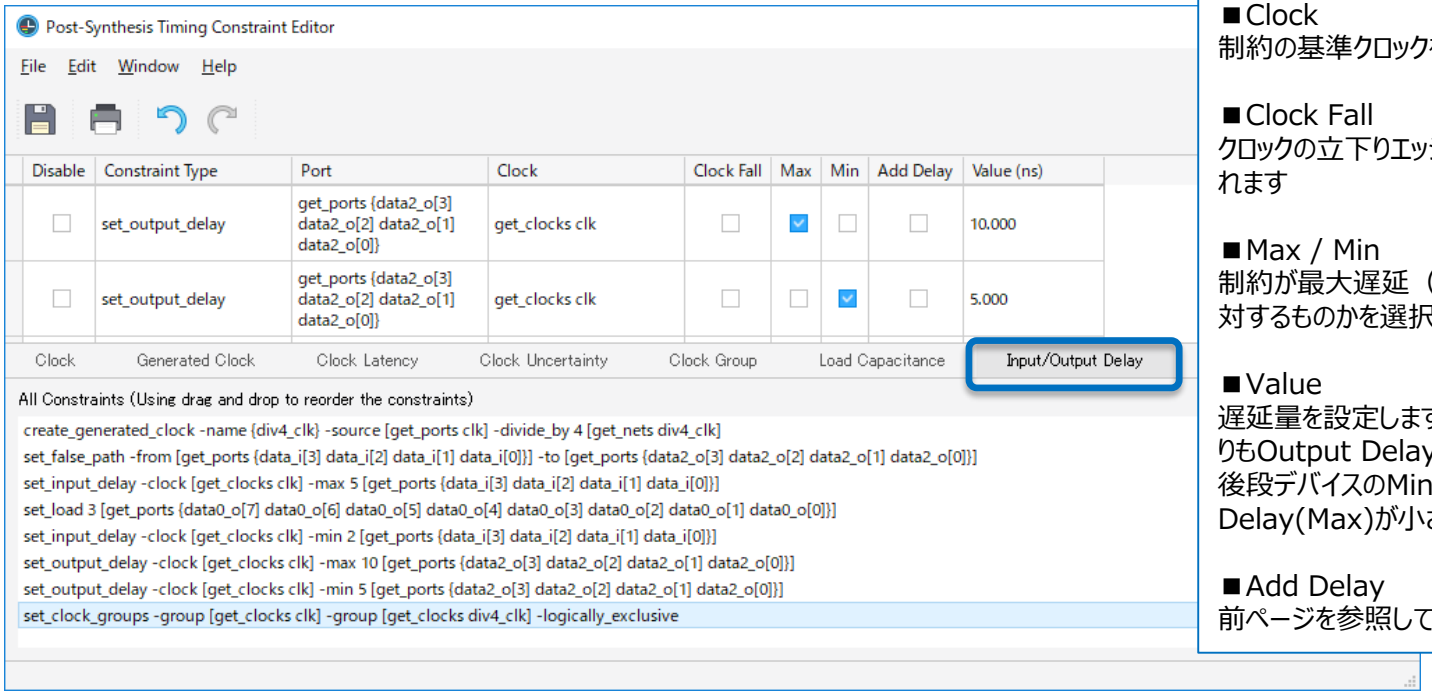
■Add Delay
同一データパスに複数の制約をかける際にチェックします。例えば同一のデータパスに同じクロックの立上がりと立下りの両方に対して制約をかける場合、Add Delayにチェックを入れないと前の制約が後に追加されている制約に上書きされます

Output Delayの設定

- Output Delayとは出力データをラッチするクロックエッジのデータに対する遅延の設定で、クロック周期からClock To Output Timeを引いた値、つまり後段デバイスのSetup Timeに相当します
- 下図を参照して制約を与えるポートと基準クロック、パラメータを設定します



Clock to Output Time
= T - Output Delay



■Constraint Type
ここではOutput Delay制約を選択します

■Port
制約対象となるデータ出力ポートを選択します

■Clock
制約の基準クロックを選択します

■Clock Fall
クロックの立下りエッジを基準として制約をかける場合にチェックを入れます

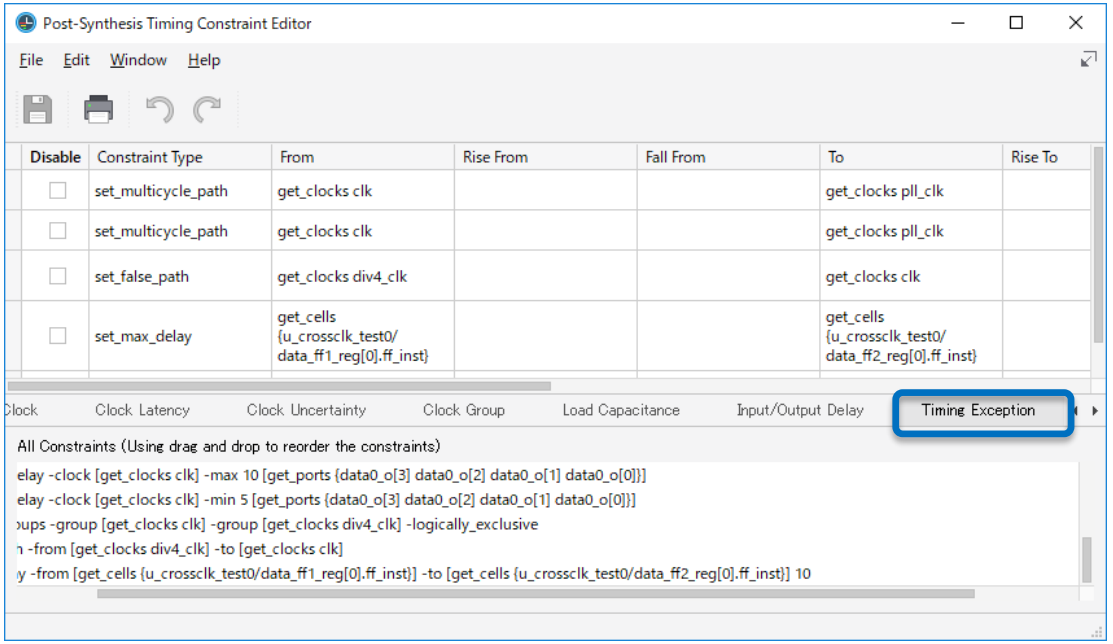
■Max / Min
制約が最大遅延（Max）に対するものか、最小遅延（Min）に対するものかを選択します

■Value
遅延量を設定します。後段デバイスのMin Setup Timeの値がよりもOutput Delay(Min)が小さくならないように設定します。また、後段デバイスのMin Hold TimeよりもT - Output Delay(Max)が小さくならないように設定します

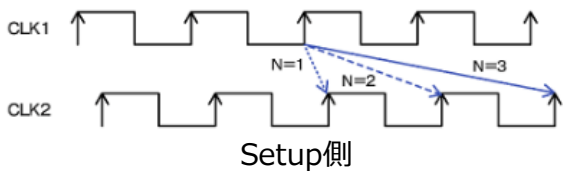
■Add Delay
前ページを参照して下さい

その他のタイミング制約(Min/Max Delay, Multicycle, False Path)

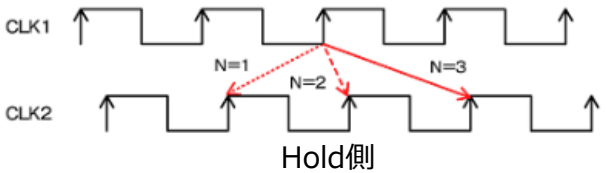
- 詳細なパス間の最大・最小遅延制約、マルチサイクル制約、False Path制約はTiming Exceptionタブから設定可能です
- 非同期クロックドメインのデータ受け渡しの際のタイミング制約や、解析対象から外したいパス等はここから設定します



*タイミング解析における解析エッジ



Setup側



Hold側

■Constraint Type
set_max_delay：指定パス間の最大遅延制約設定
set_min_delay：指定パス間の最小遅延制約設定
set_multicycle_path：マルチサイクルパス設定
set_false_path：解析対象外パスの設定

■From / To
制約の起点と終点を設定します

■Max / Min delay
Max Delay制約、Min Delay制約の制約値を設定します

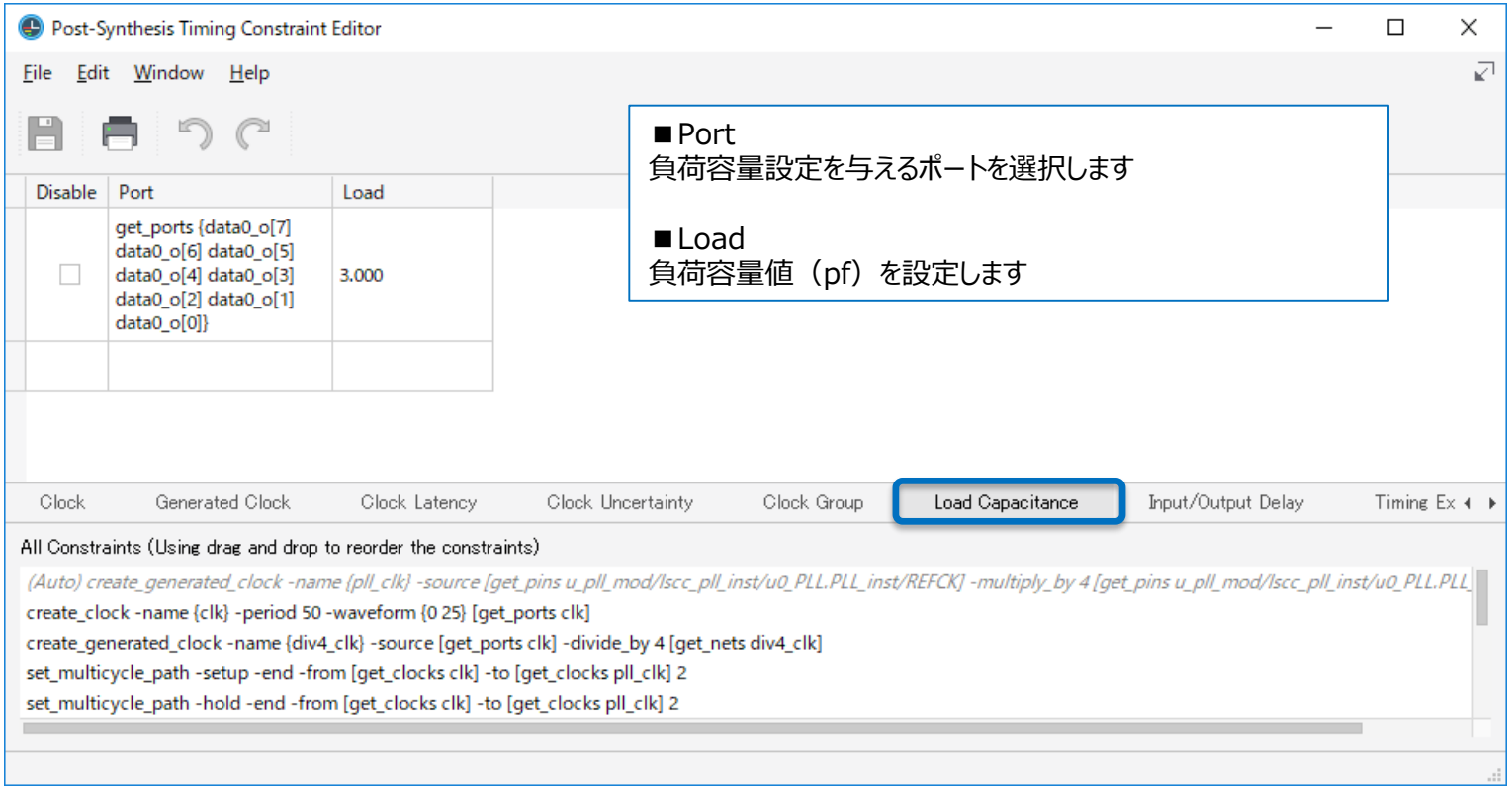
■Setup / Hold
マルチサイクル制約において、サイクル数Nを増すエッジをSetup側解析エッジにするかHold側解析エッジにするか選択します（左図の解析エッジ参照）Setup側に制約をかけると、同時にHold側に自動で制約がかかり、タイミングエラーとなる場合があります。この場合Hold側にも個別に制約をかける必要があります

■Start / End
マルチサイクル制約のサイクル数を送信側クロックでカウントする(start)か、受信側クロックでカウントするか(end)を選択します

■Multiplier
マルチサイクル制約のサイクル数Nを設定します

その他の制約(Load Capacitance)

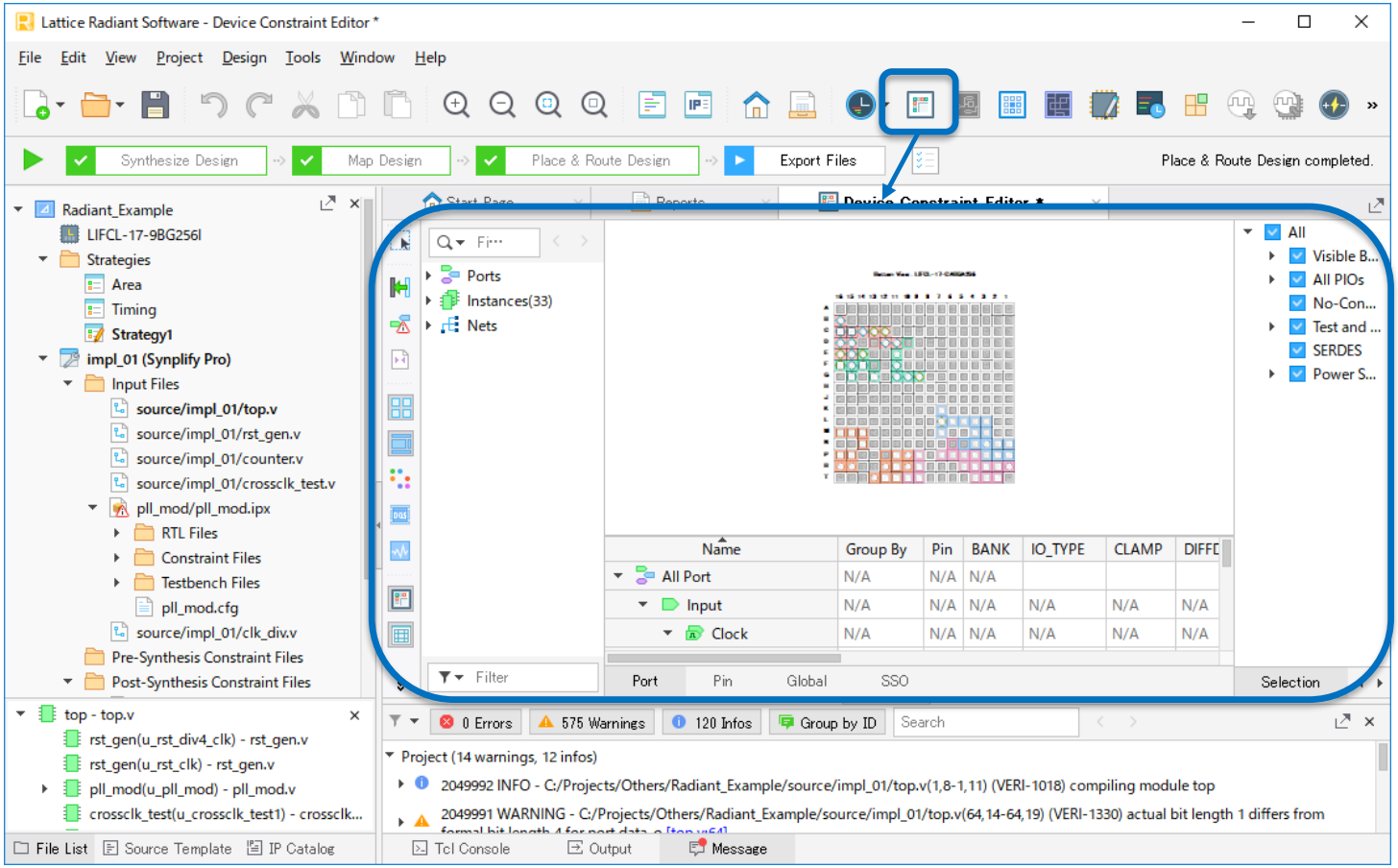
- 出力ポートに対して、接続先の負荷容量の設定が可能です。出力タイミング解析時に負荷容量を加味した解析が行われるようになります
- タイミング解析の他に、SSOアナライザを使用した同時スイッチングノイズの影響度解析にもここで設定された負荷容量が使用されます



3. I/Oアサイメント、デバイス全般の設定

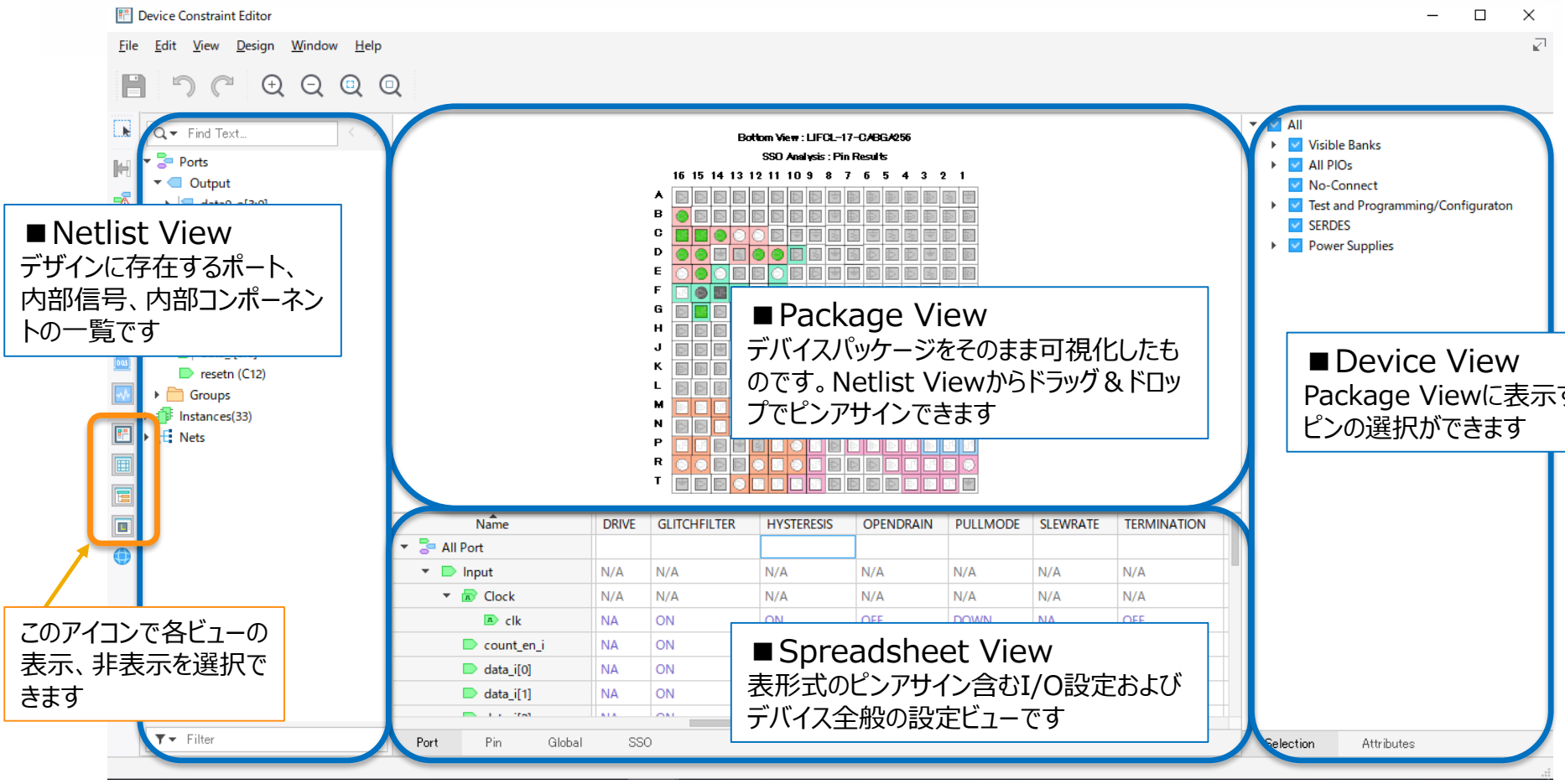
Device Constraint Editor

- 入出力信号のI/OへのアサイメントDevice Constraint Editorで行います
- Synthesize Designが完了している状態でTools > Device Constraint Editorもしくは、ToolbarのDevice Constraint Editorアイコン から開きます



Device Constraint Editor

- Device Constraint EditorのGUIは以下のようになっています



■Netlist View
デザインに存在するポート、内部信号、内部コンポーネントの一覧です

このアイコンで各ビューの表示、非表示を選択できます

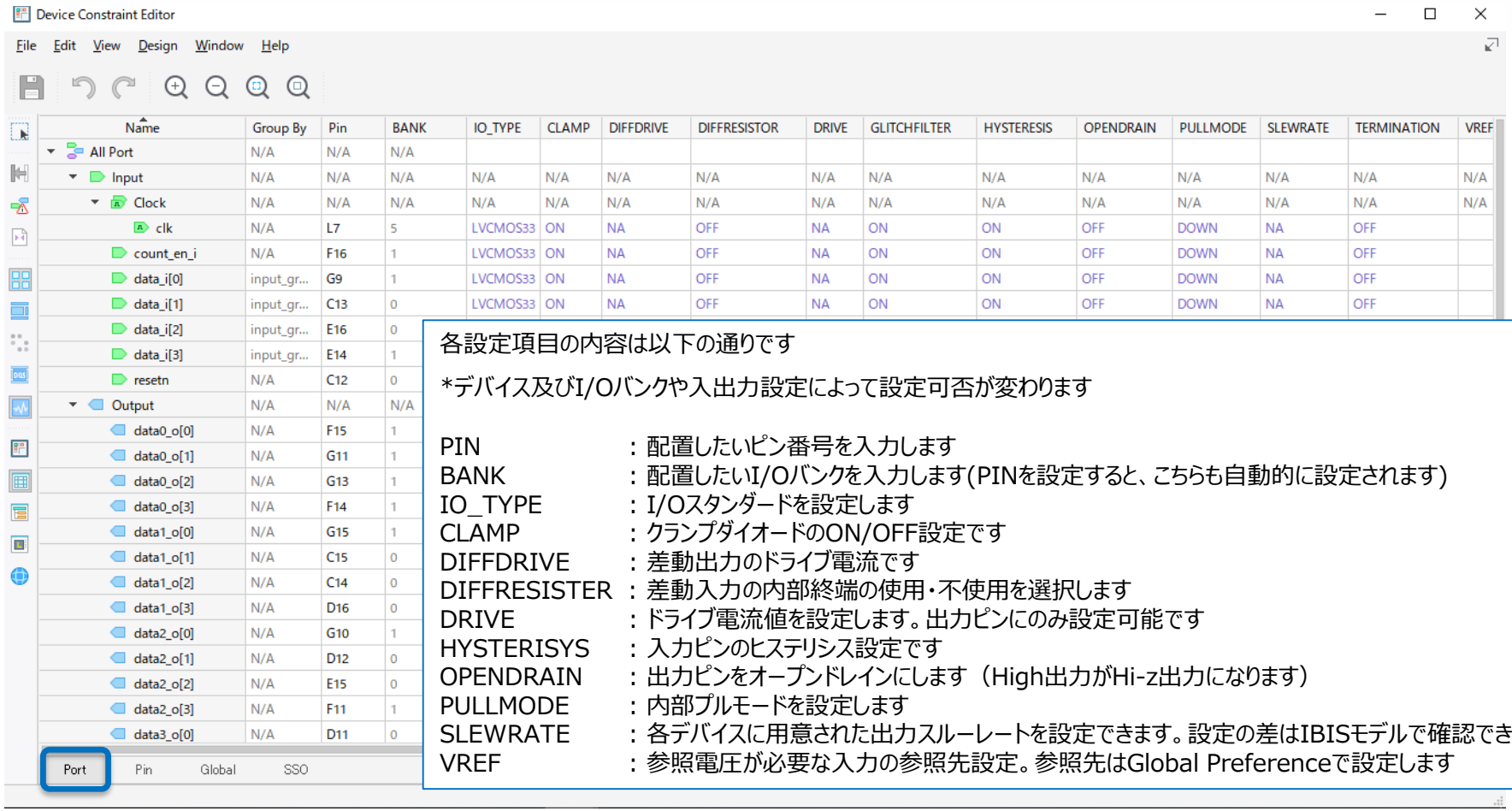
■Package View
デバイスパッケージをそのまま可視化したものです。Netlist Viewからドラッグ＆ドロップでピンアサインできます

■Spreadsheet View
表形式のピンアサイン含むI/O設定およびデバイス全般の設定ビューです

■Device View
Package Viewに表示するピンの選択ができます

Spreadsheet View

- Spreadsheet ViewのPortタブではピンアサインやI/O Type（電圧スタンダード）等を表形式のシートで細かく設定できます



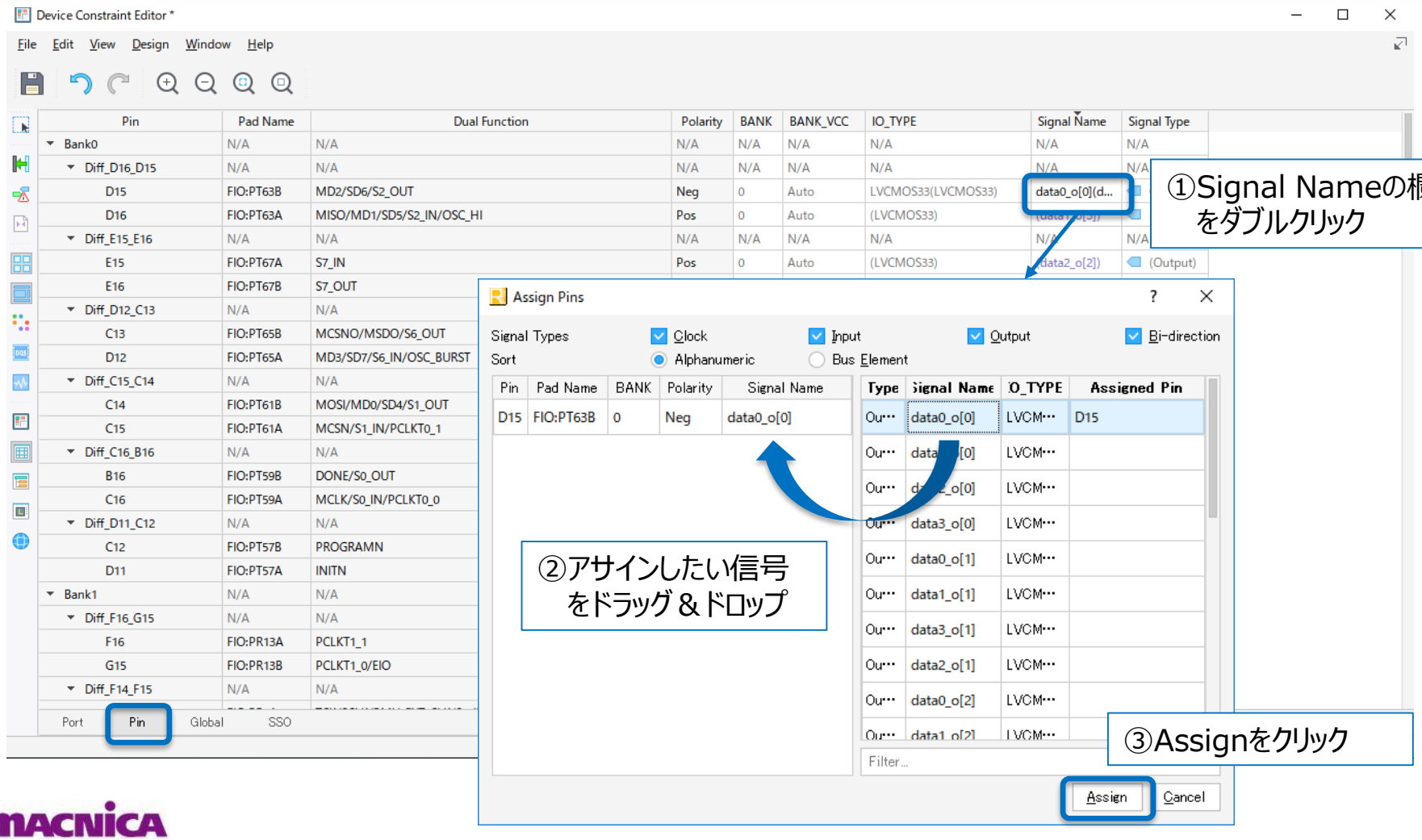
各設定項目の内容は以下の通りです

*デバイス及びI/Oバンクや入出力設定によって設定可否が変わります

PIN	：配置したいピン番号を入力します
BANK	：配置したいI/Oバンクを入力します(PINを設定すると、こちらも自動的に設定されます)
IO_TYPE	：I/Oスタンダードを設定します
CLAMP	：クランプダイオードのON/OFF設定です
DIFFDRIVE	：差動出力のドライブ電流です
DIFFRESISTER	：差動入力の内部終端の使用・不使用を選択します
DRIVE	：ドライブ電流値を設定します。出力ピンにのみ設定可能です
HYSTERISYS	：入力ピンのヒステリシス設定です
OPENDRAIN	：出力ピンをオープンドレインにします（High出力がHi-z出力になります）
PULLMODE	：内部プルモードを設定します
SLEWRATE	：各デバイスに用意された出力スルーレートを設定できます。設定の差はIBISモデルで確認できます
VREF	：参照電圧が必要な入力の参照先設定。参照先はGlobal Preferenceで設定します

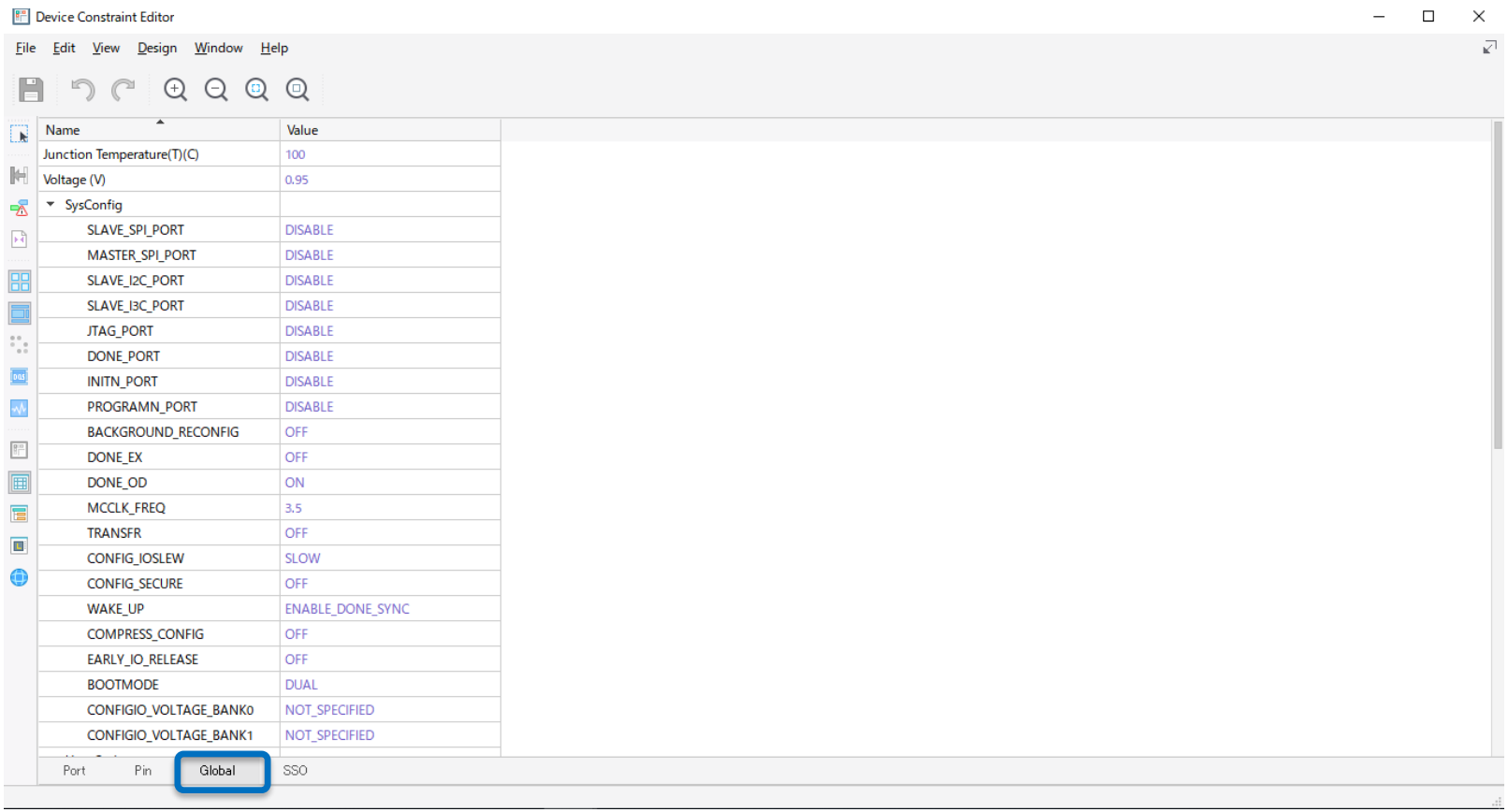
Spreadsheet View

- Spreadsheet ViewのPinタブではI/OのDual Functionを確認しながらピンアサインが可能です
- I/O TYPE等の細かい設定はPortタブで行う必要があります



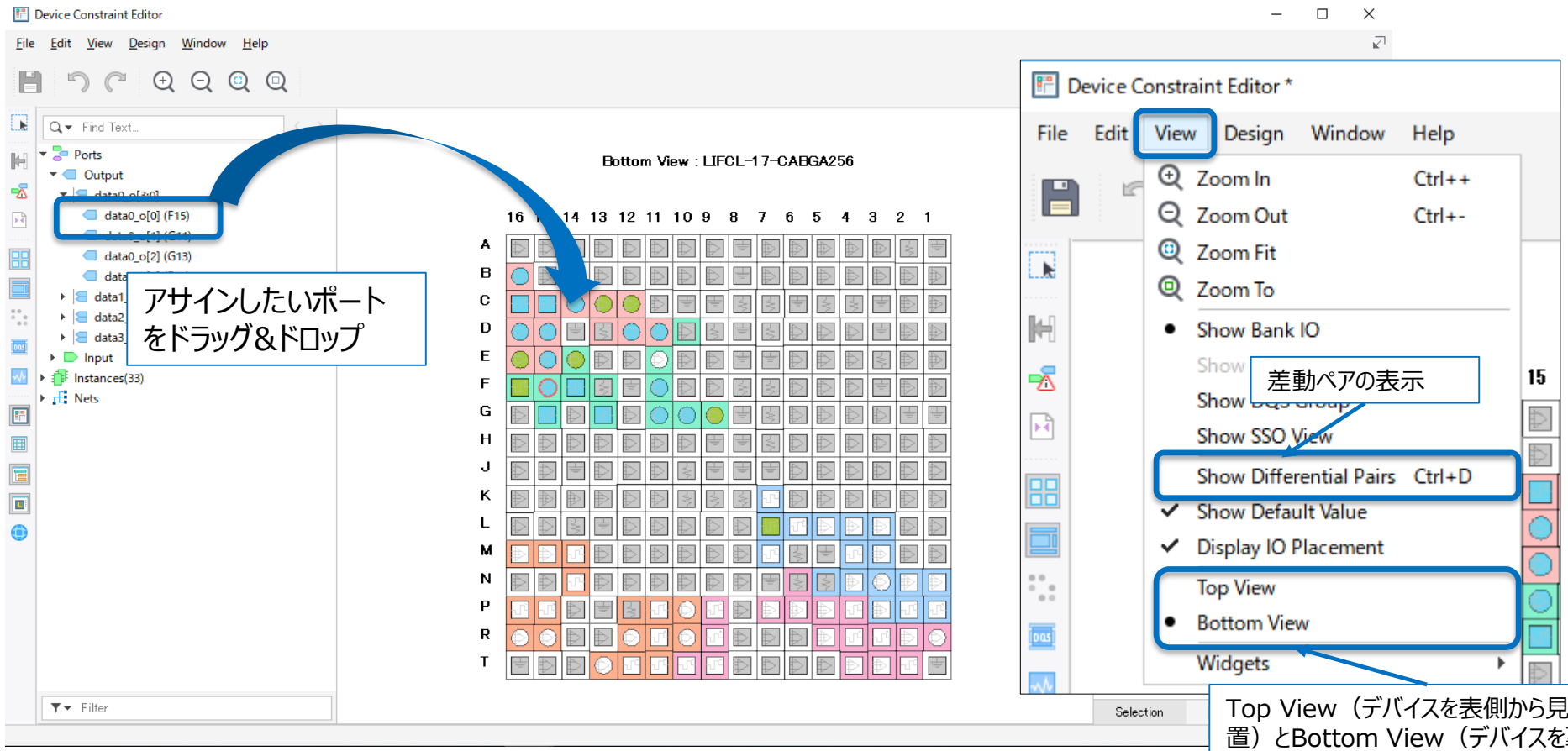
Spreadsheet View

- Spreadsheet ViewのGrobalタブではデバイスのコンフィグレーション関連ピンの有効・無効設定等の全般的な設定を行うことができます



Package View

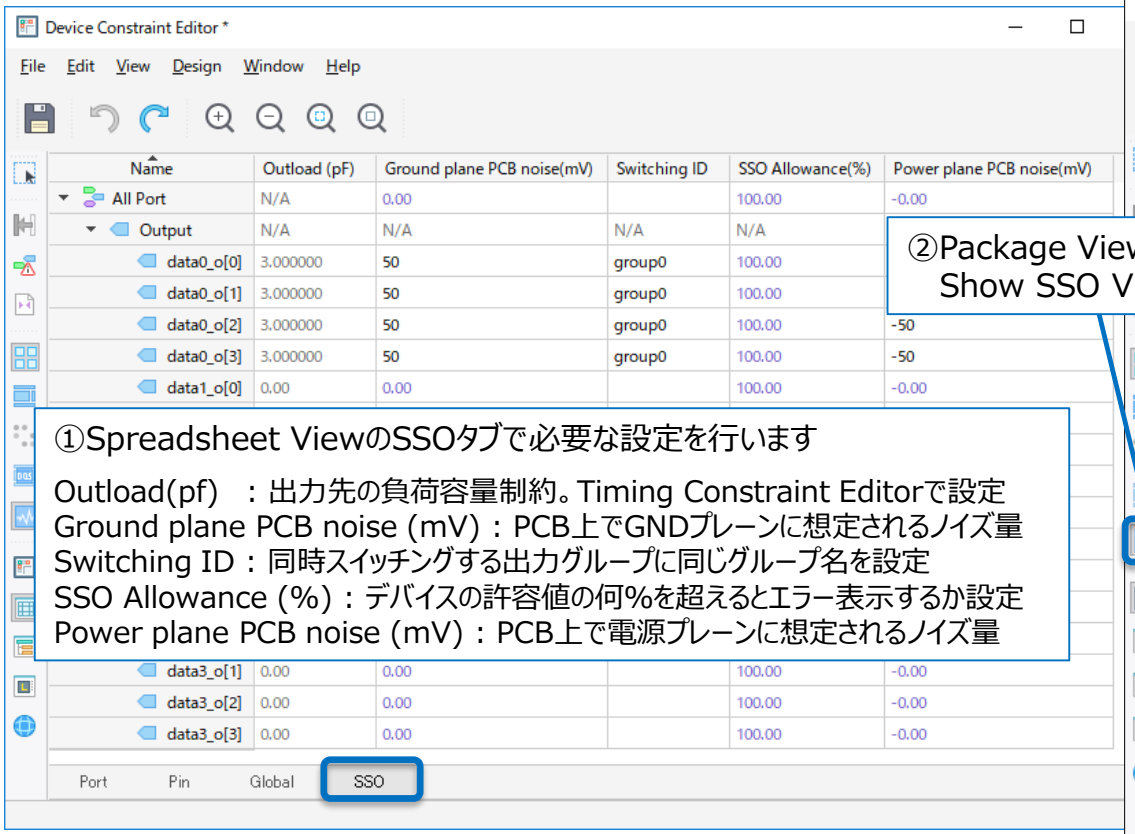
- Package ViewではNetlist Viewに表示されているPortの信号をドラッグ&ドラッグによってグラフィカルにピンアサインが行えます
- ViewでTop View/Bottom Viewの切り替えや、差動ペアの表示も可能です



Top View（デバイスを表側から見た配置）とBottom View（デバイスを裏側から見た配置）の切り替え

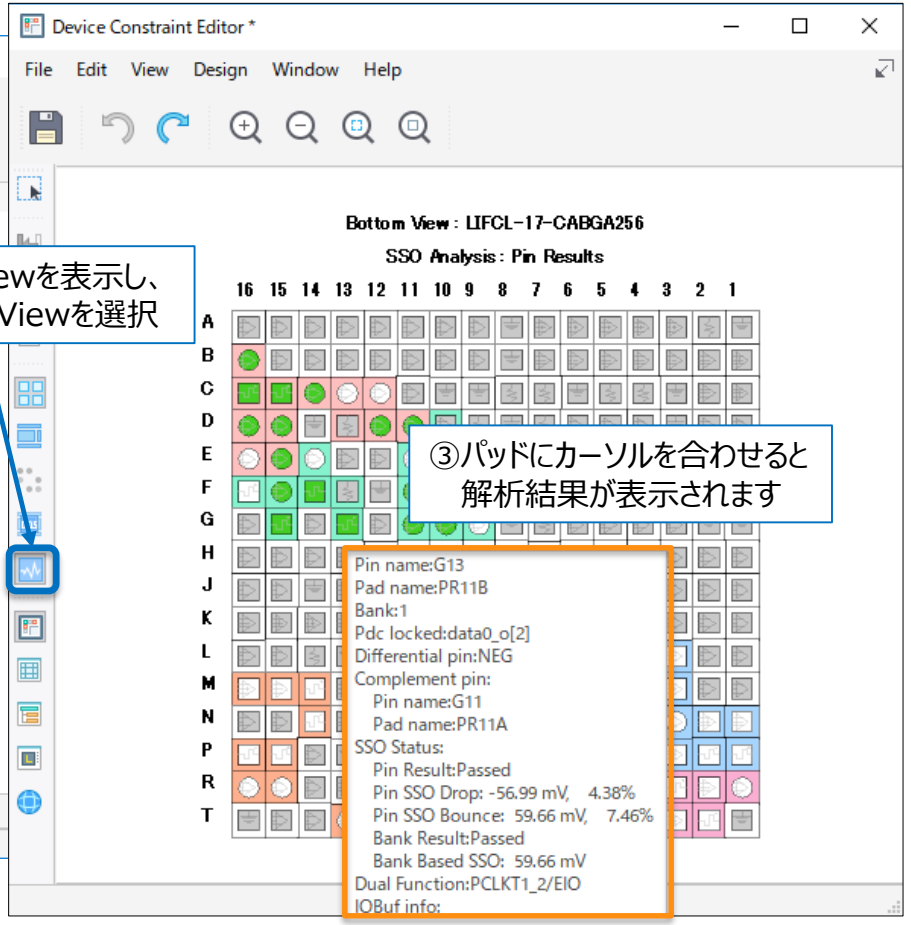
（参考）SSO解析

- Spreadsheet ViewのSSOタブでSSO（同時スイッチング出力）の影響確認用の設定を行い、Package View上で影響度の確認を行うことができます
- 予めTiming Constraint EditorのLoad Capacitance設定で出力先の負荷容量の制約をかけておく必要があります



①Spreadsheet ViewのSSOタブで必要な設定を行います

Outload(pf) ：出力先の負荷容量制約。Timing Constraint Editorで設定
Ground plane PCB noise (mV)：PCB上でGNDプレーンに想定されるノイズ量
Switching ID：同時スイッチングする出力グループに同じグループ名を設定
SSO Allowance (%)：デバイスの許容値の何%を超えるとエラー表示するか設定
Power plane PCB noise (mV)：PCB上で電源プレーンに想定されるノイズ量



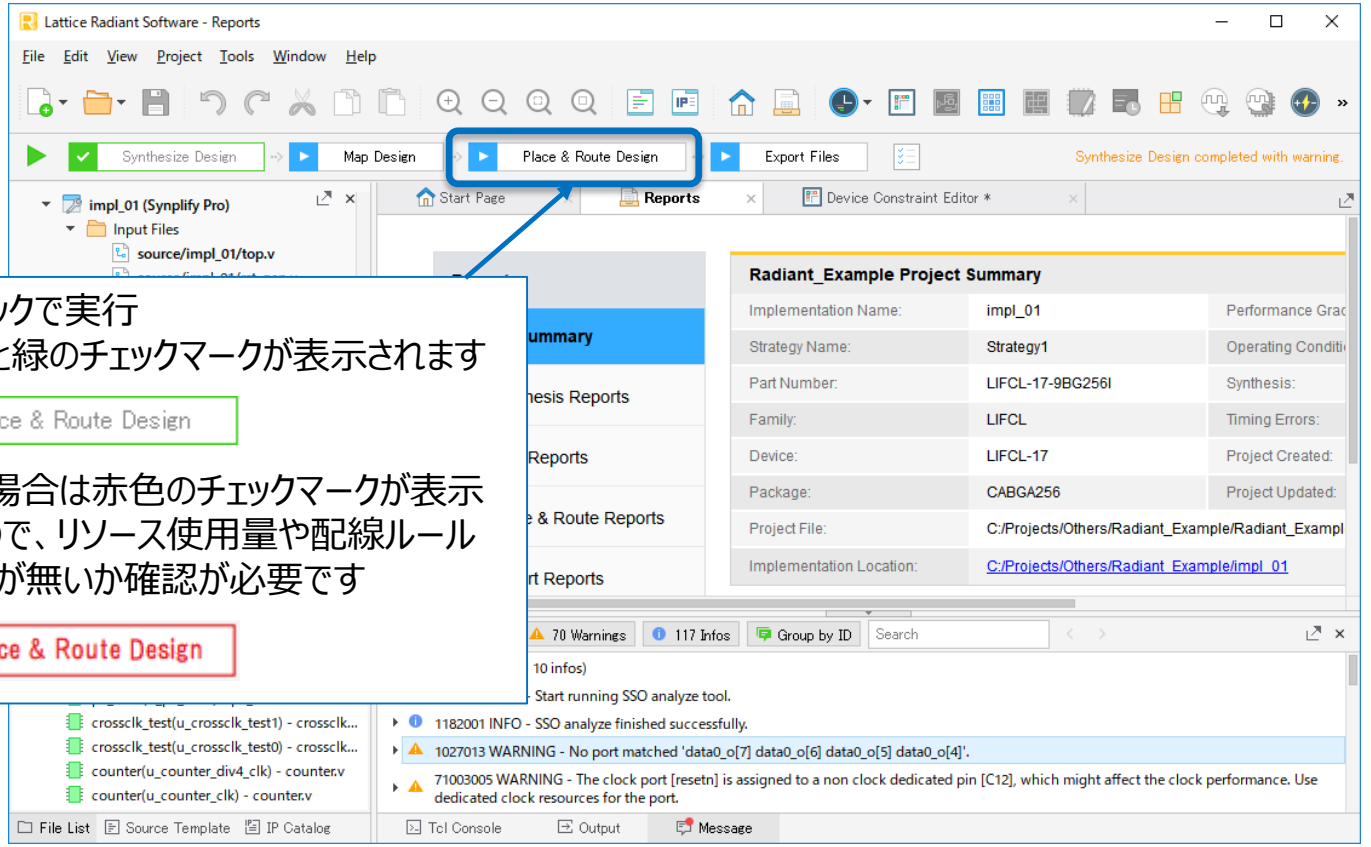
②Package Viewを表示し、Show SSO Viewを選択

③パッドにカーソルを合わせると解析結果が表示されます

4. デザインコンパイル～レポート確認

配置配線の実行

- Process ToolbarのPlace & Route Designをクリックして配置配線を実行します
- Map Design（論理回路の実リソースへのマッピング）から順番に行っても良いですが、Place & Route Designを実行すれば自動的にMap Designも実行されます
- Map Design, Prace & Route Designが成功すると緑のチェックマークが表示されます



ダブルクリックで実行
成功すると緑のチェックマークが表示されます

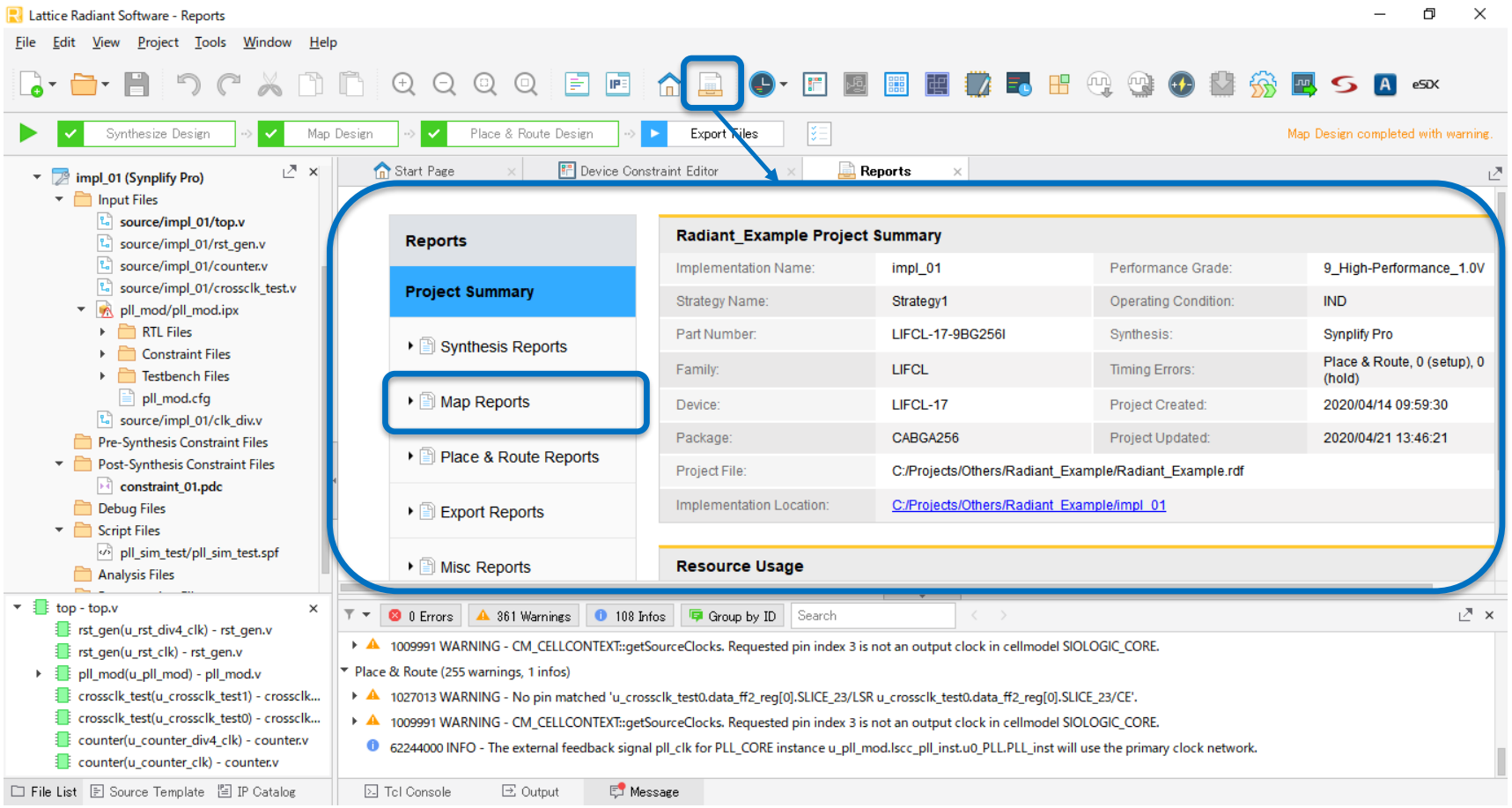
Place & Route Design

失敗した場合は赤色のチェックマークが表示されますので、リソース使用量や配線ルール等に問題が無いか確認が必要です

Place & Route Design

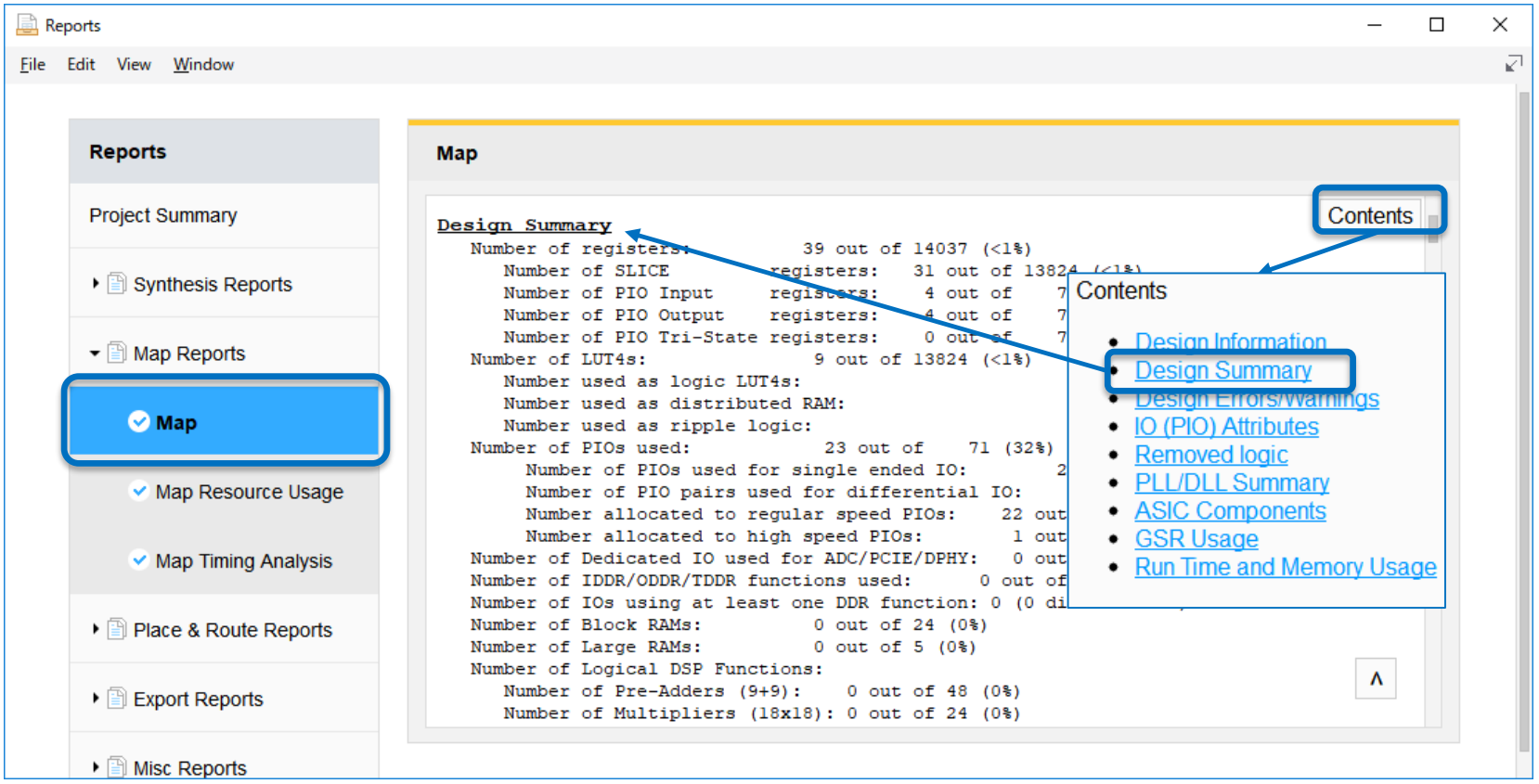
Map Reportの確認

- Map Reportではリソース使用状況の確認ができます
- Reportsアイコン をクリックし、レポート画面を表示します
- Reportsの項目でMap Reportsを選択します



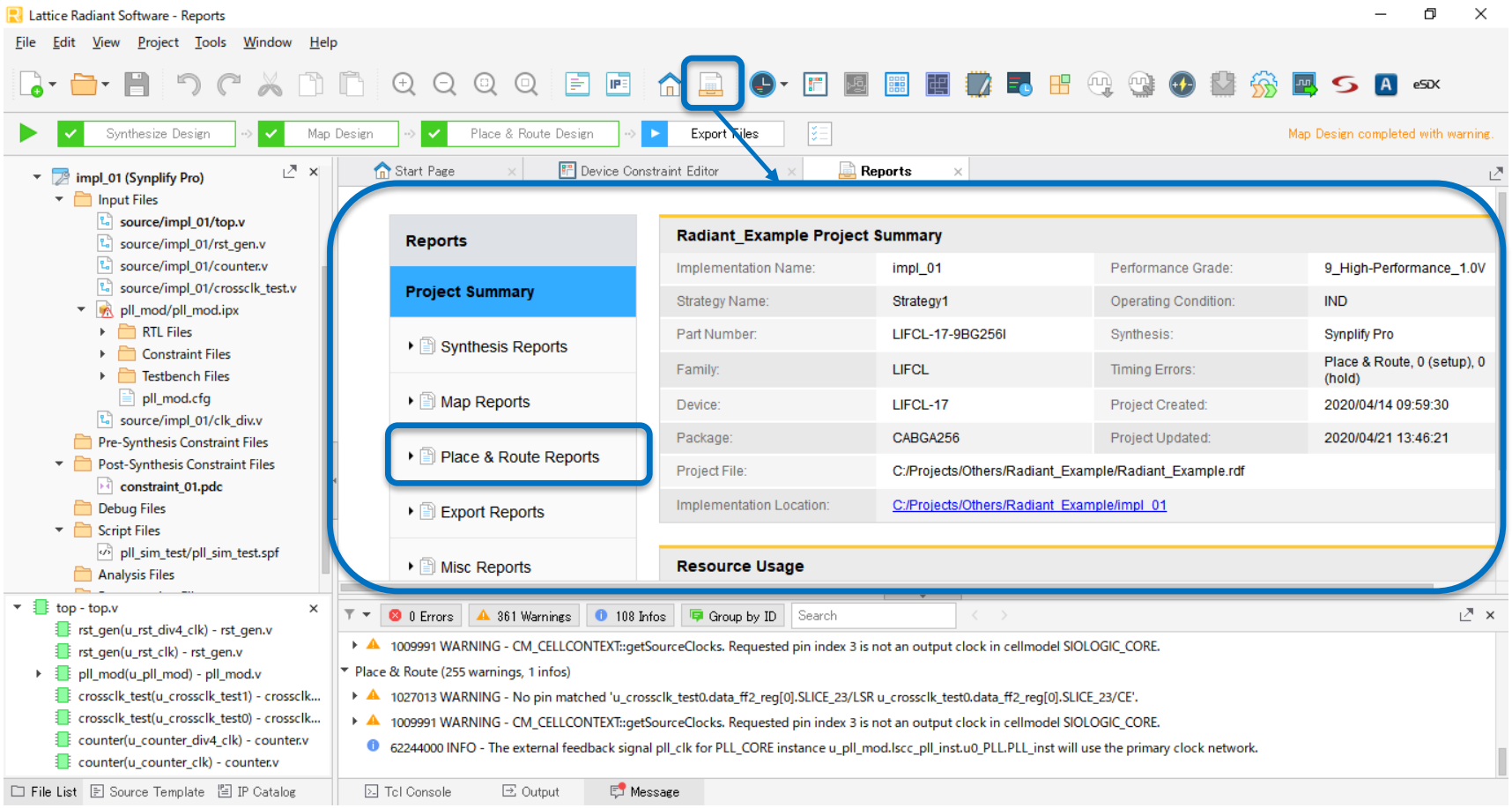
Map (Map Report) の確認

- デバイスの基本的なリソース（LUT、SLICE、Register、EBR、DSP、PLL、OSC、I/O Logic等）の使用状況の確認はMapのDesign Summaryで行います
- 右上のContentsボタンにカーソルを合わせると、レポートコンテンツが表示されます
- 確認したいコンテンツをクリックすると、その記載箇所にジャンプします



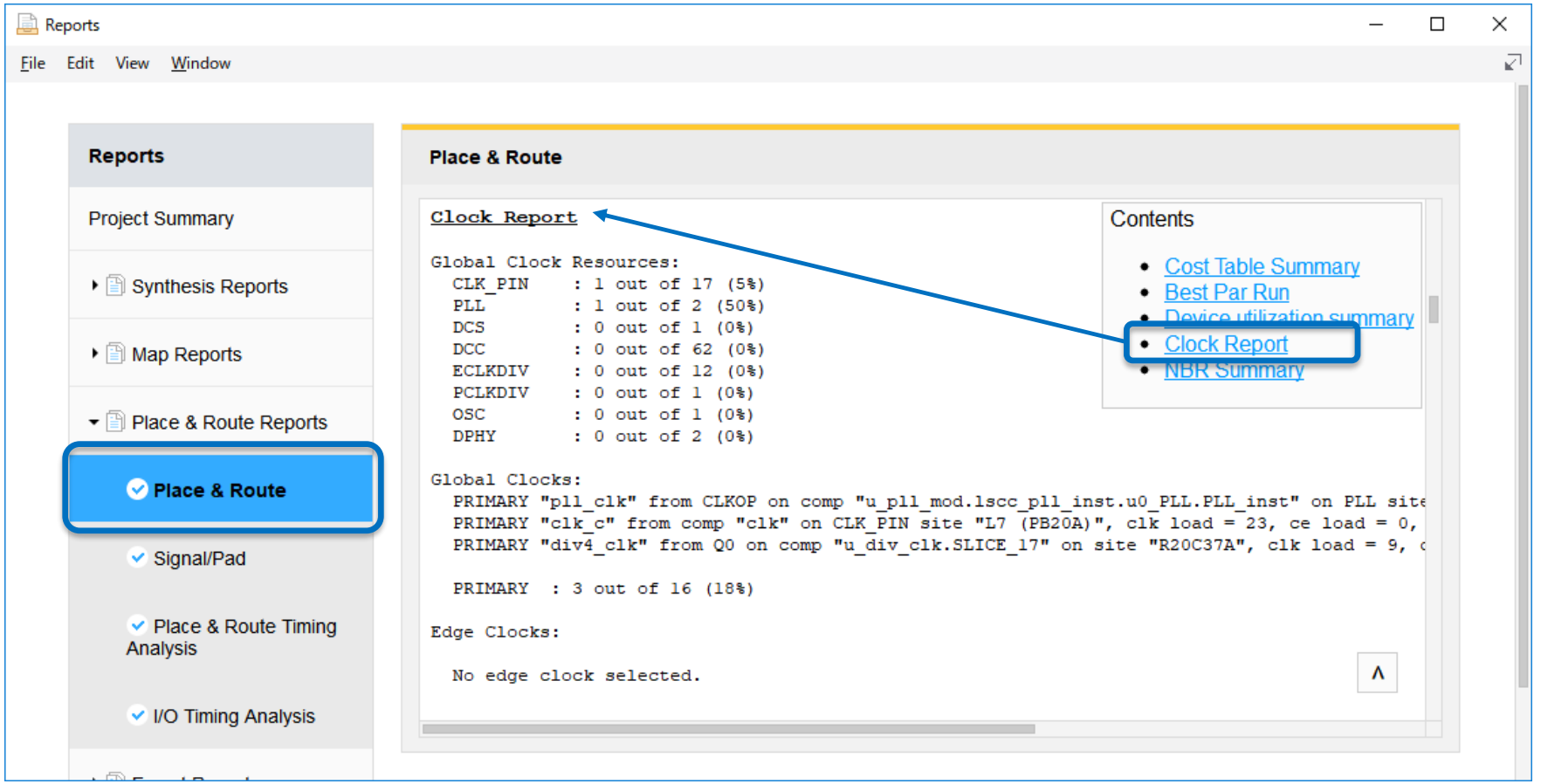
Place & Route Reportの確認

- Place & Route Reportではリソース使用状況の確認ができます
- Reportsの項目でPlace & Route Reportsを選択します



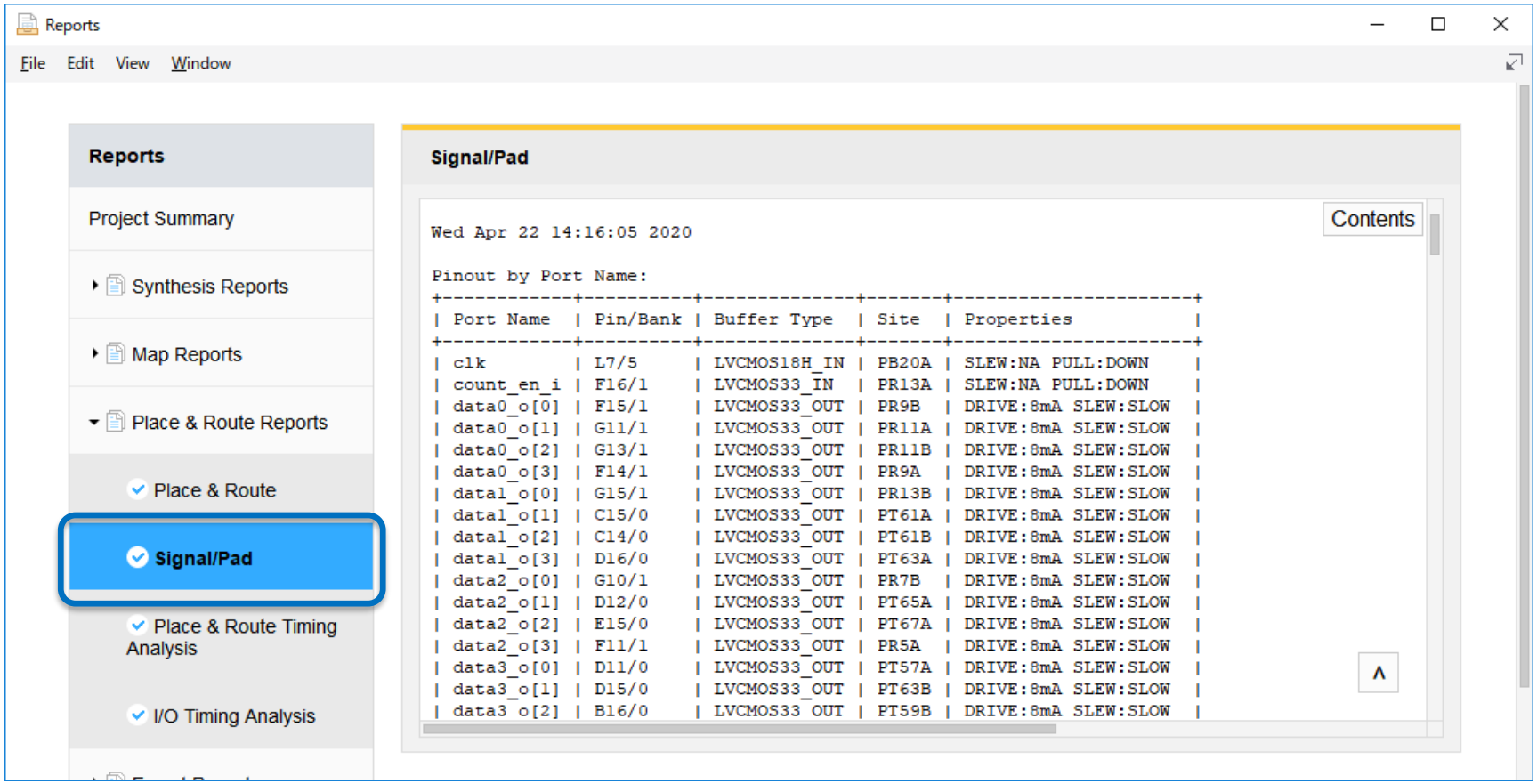
Place & Route (Place & Route Report) の確認

- Place＆Routeではデザインが使用しているリソースの使用率が確認できます
- リソースの使用率確認は基本的にはMap Reportで確認しますが、Place & RouteではMap Reportで確認できないクロック関連リソースの使用状況やクロック専用配線のアサイン状況が確認できます



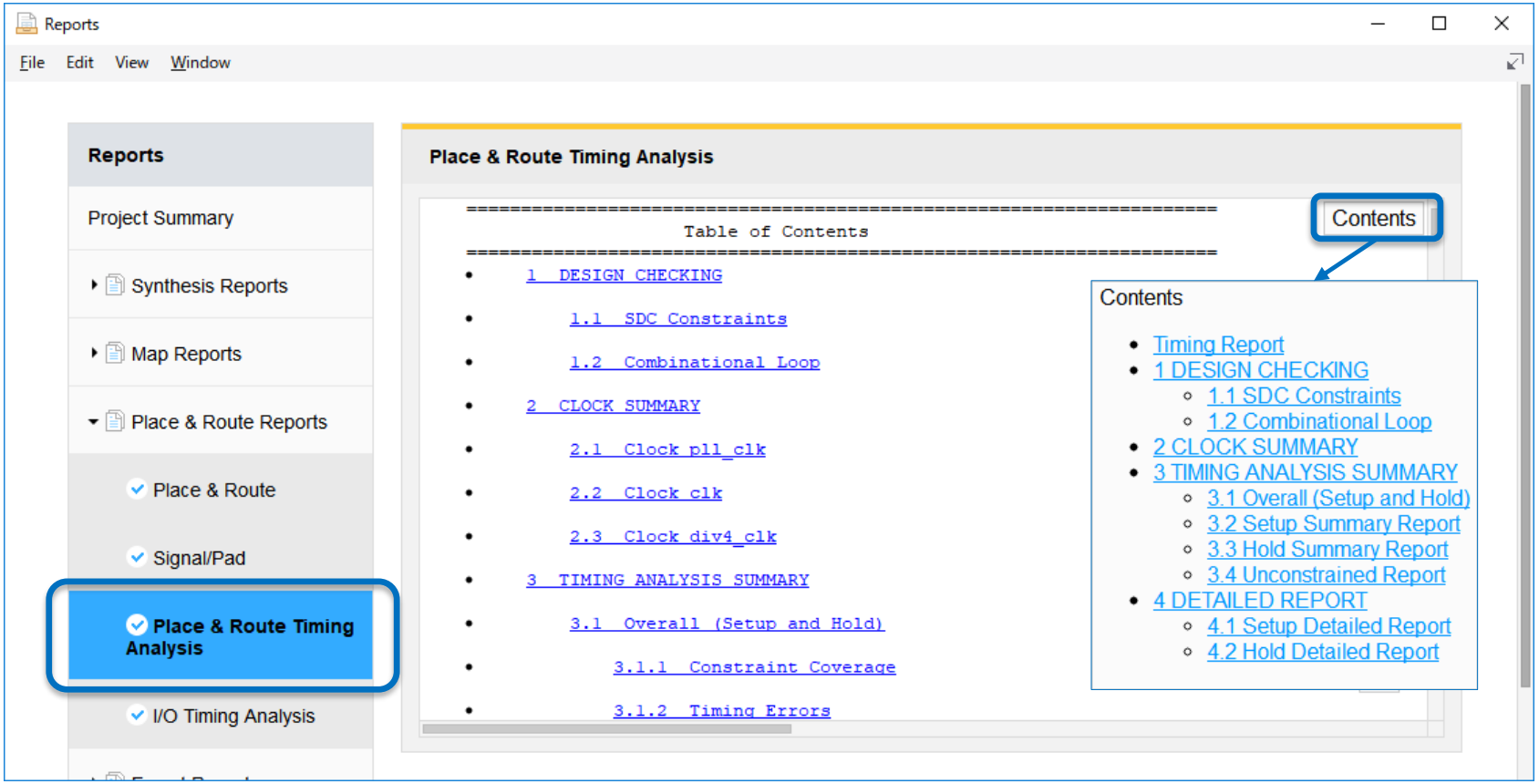
Signal/Pad (Place & Route Report) の確認

- Signal/Padでは配置配線後の実際のピン配置結果、Buffer Typeやその他I/O設定の反映結果が確認できます



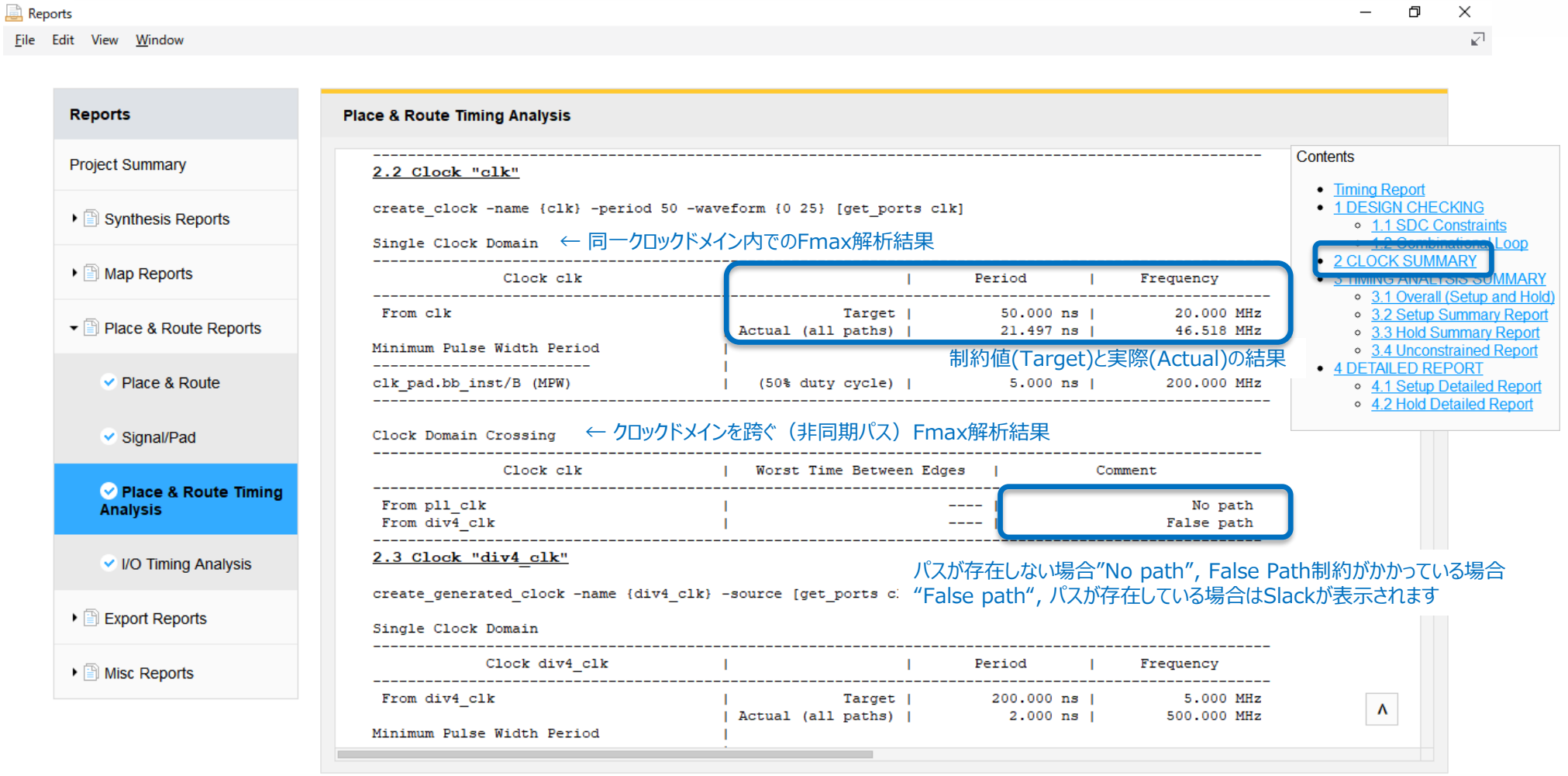
Place & Route Timing Analysis (Place & Route Report) の確認

- Place & Route Timing Analysisでは配置配線後の静的タイミング解析（STA）結果を確認できます
- 右上のContentsボタンから確認したいレポートを選択します



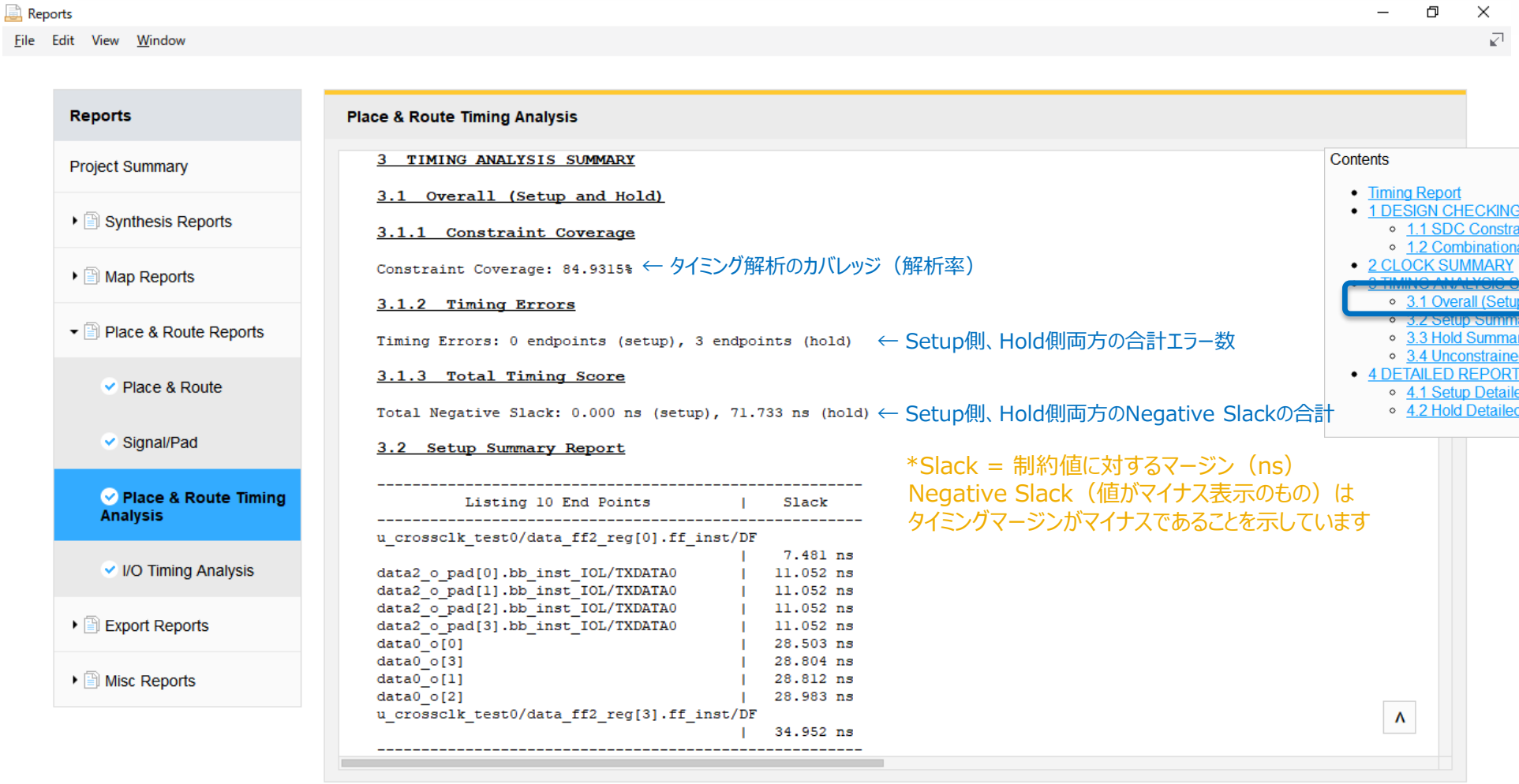
Place & Route Timing Analysis (Place & Route Report) の確認

■ 以下はCLOCK SUMMARY（Fmax解析結果）の表示例です



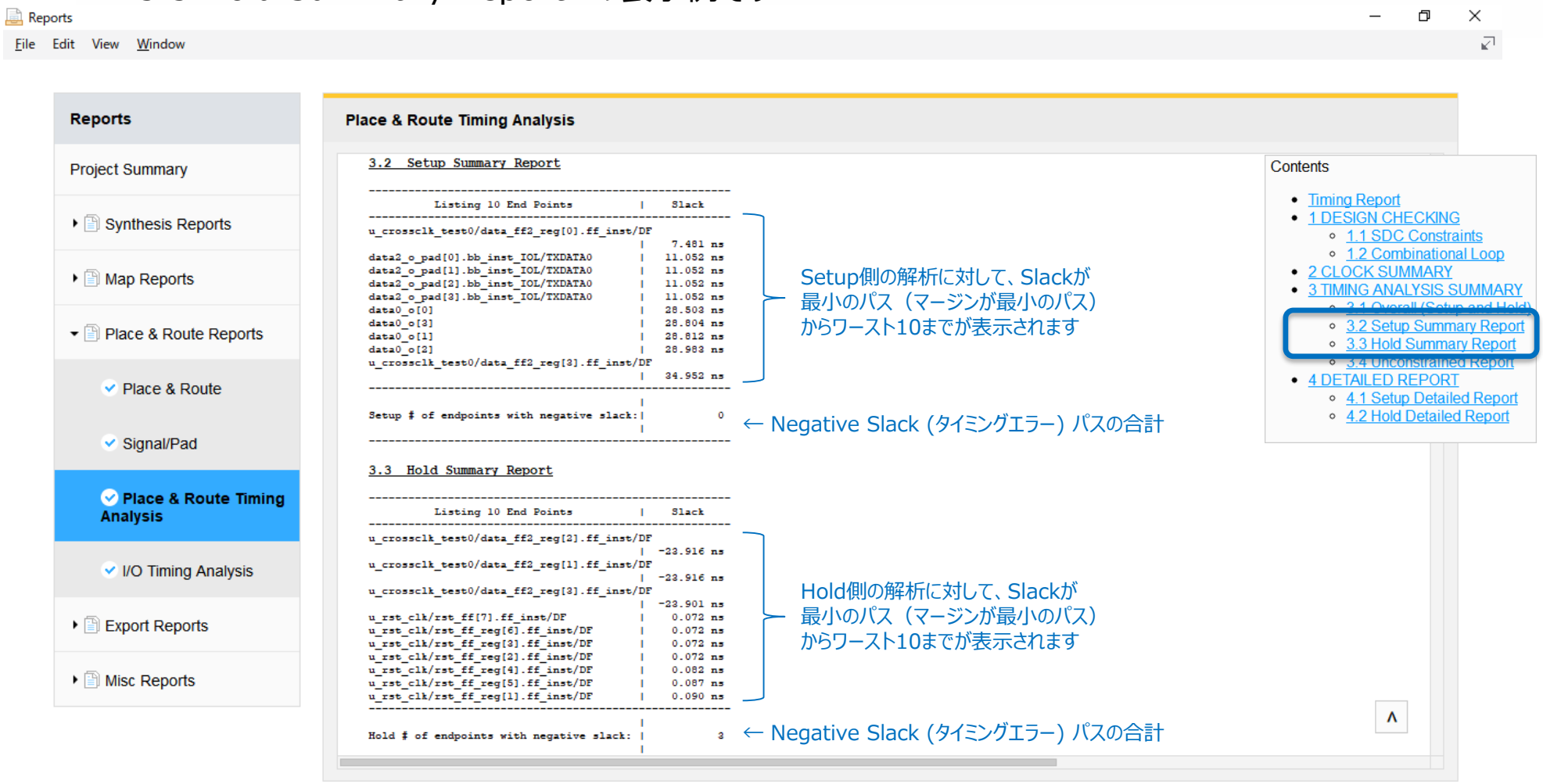
Place & Route Timing Analysis (Place & Route Report) の確認

■ 以下はTIMING ANALYSIS SUMMARYの ”3.1 Overall (Setup and Hold)” の表示例です



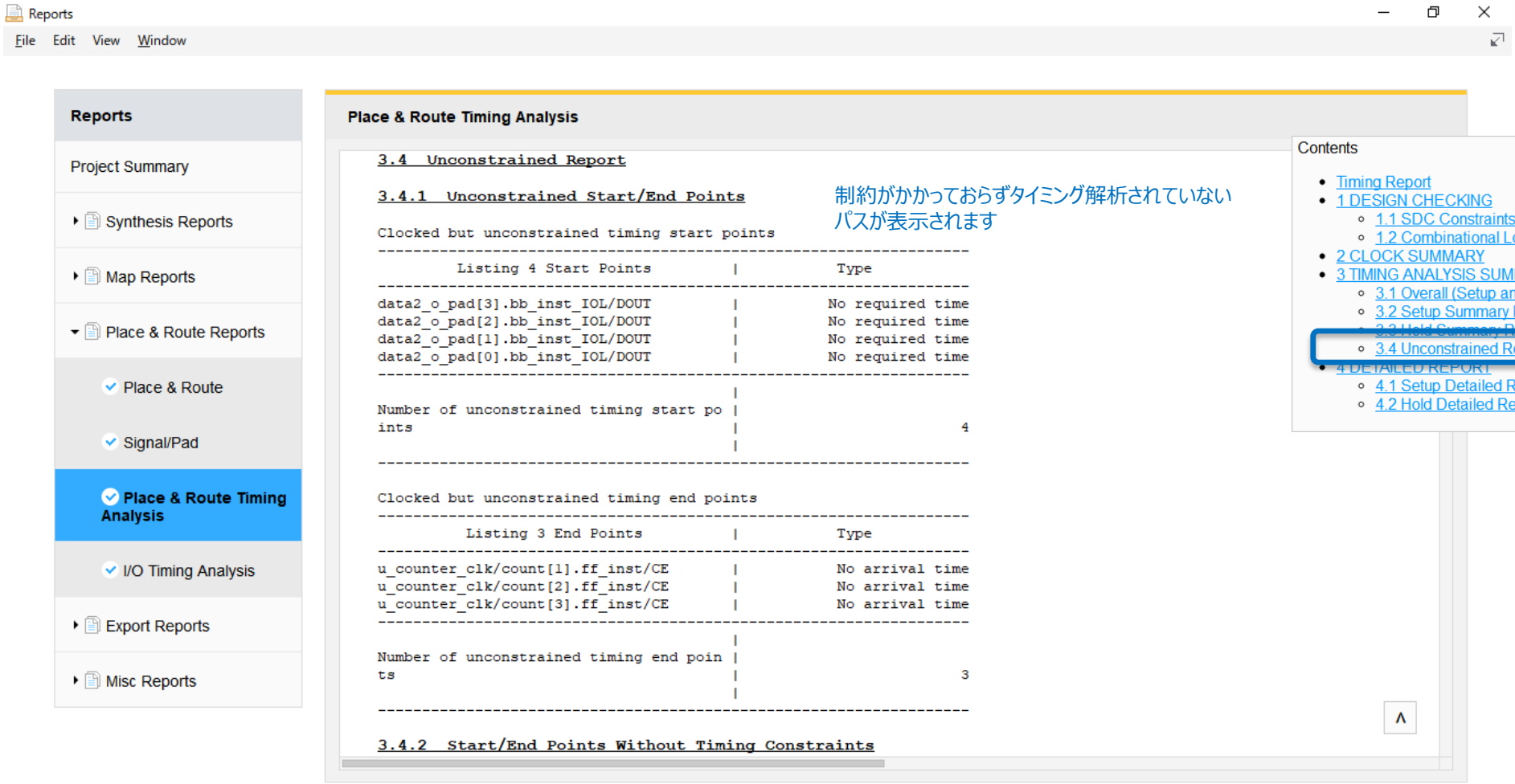
Place & Route Timing Analysis (Place & Route Report) の確認

■ 以下はTIMING ANALYSIS SUMMARYの ”3.2 Setup Summary Report” と “3.3 Hold Summary Report” の表示例です



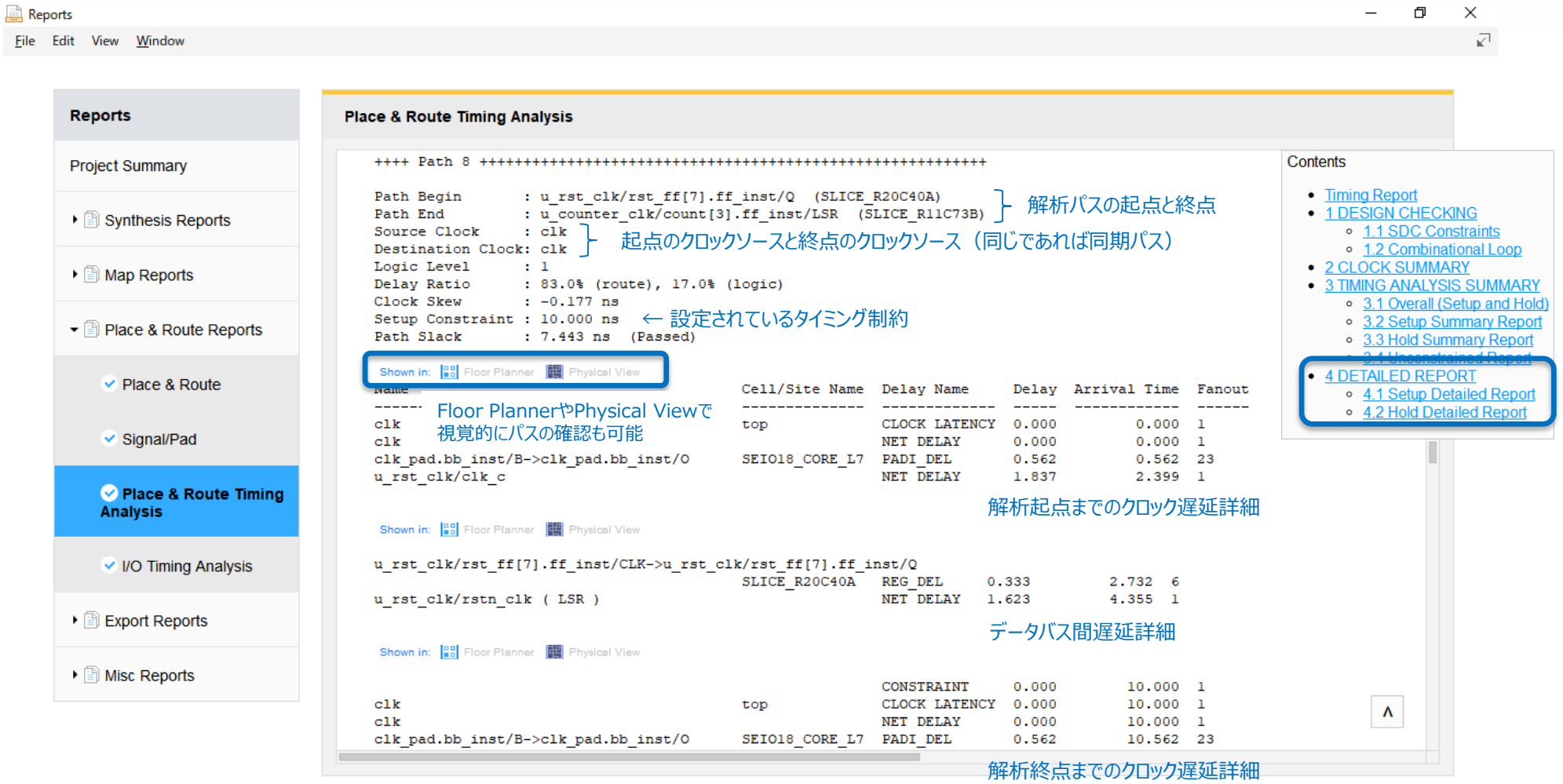
Place & Route Timing Analysis (Place & Route Report) の確認

■ 以下はTIMING ANALYSIS SUMMARYの ”3.4 Unconstrainted Report” の表示例です



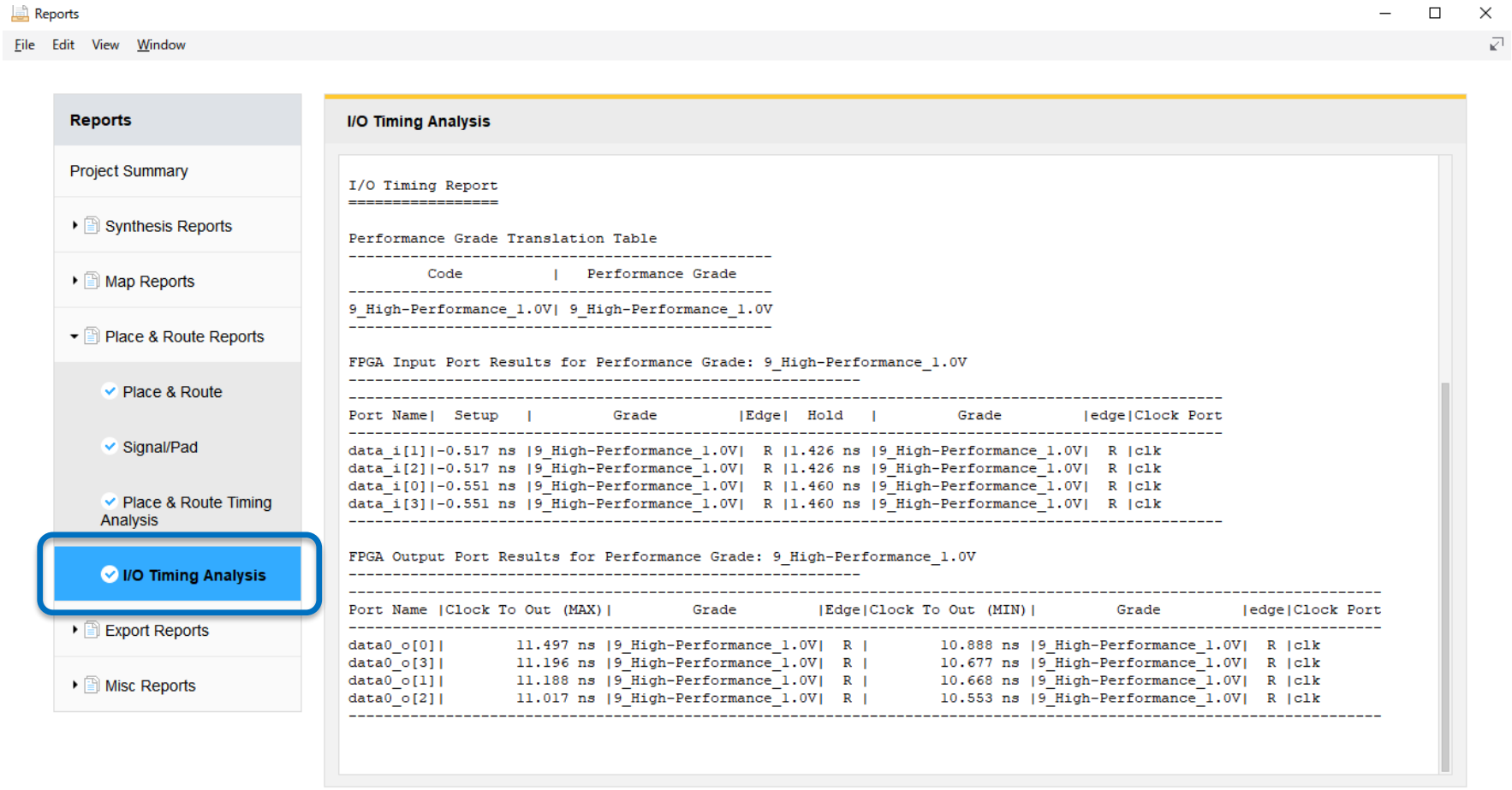
Place & Route Timing Analysis (Place & Route Report) の確認

- 以下はTIMING ANALYSIS SUMMARYの ”4 DETAILED REPORT” の表示例です
- この例では同期クロック間でのFmax解析の詳細を示しています



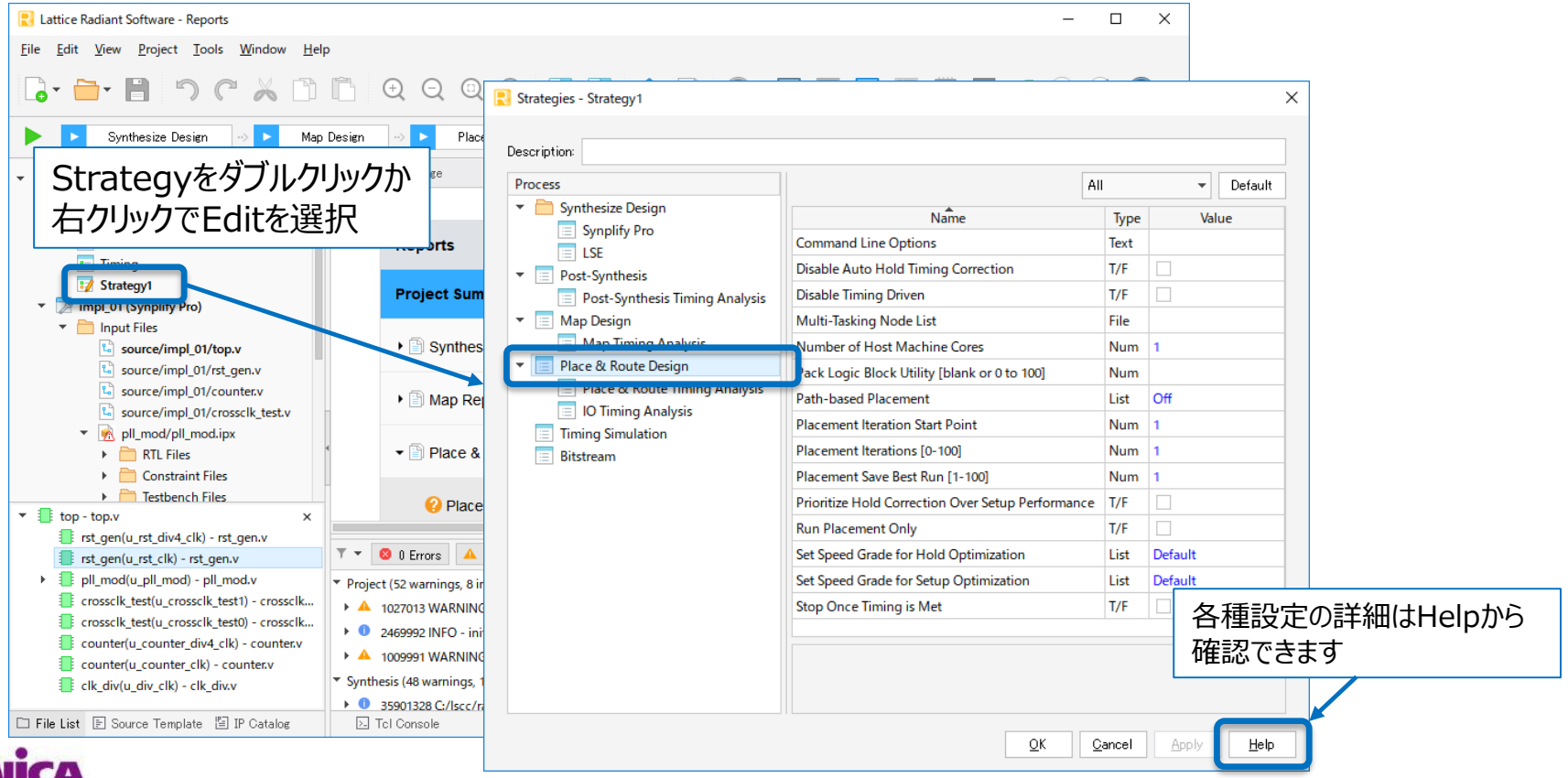
I/O Timing Analysis (Place & Route Report) の確認

- I/O Timing Analysisでは配置配線後のI/Oタイミング解析（STA）結果を確認できます
- 入力ポートのワーストSetup/Hold Timeと解析クロックエッジ、出力ポートのワーストTcoと解析クロックエッジを以下のように確認できます



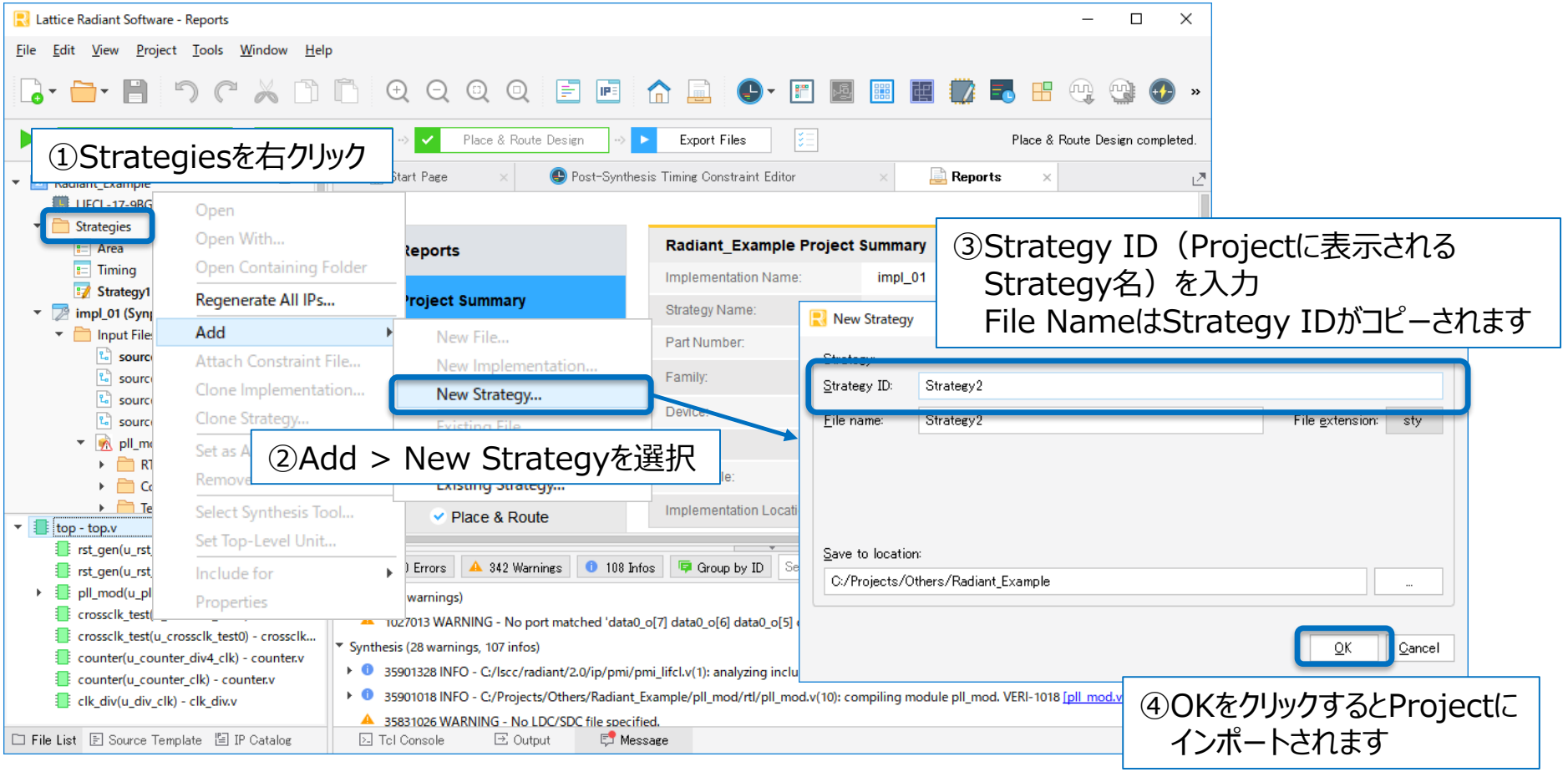
（参考）Strategy設定

- タイミングエラーが出た場合の対処として、Strategy(コンパイルオプション)を変更することができます
- Strategyを変更してPlace & Route Designを実行することで配置配線結果に変化を与えます
- デフォルトで “Area”, “Timing”, “Strategy1”が用意されており、Strategy1がアクティブになっています
- Project ViewのStrategiesの下にあるStrategy1をダブルクリックするか右クリックからEditを選択すると設定画面が開きます
- 配置配線時に与えるコンパイルオプションはPlace & Route Designを選択します



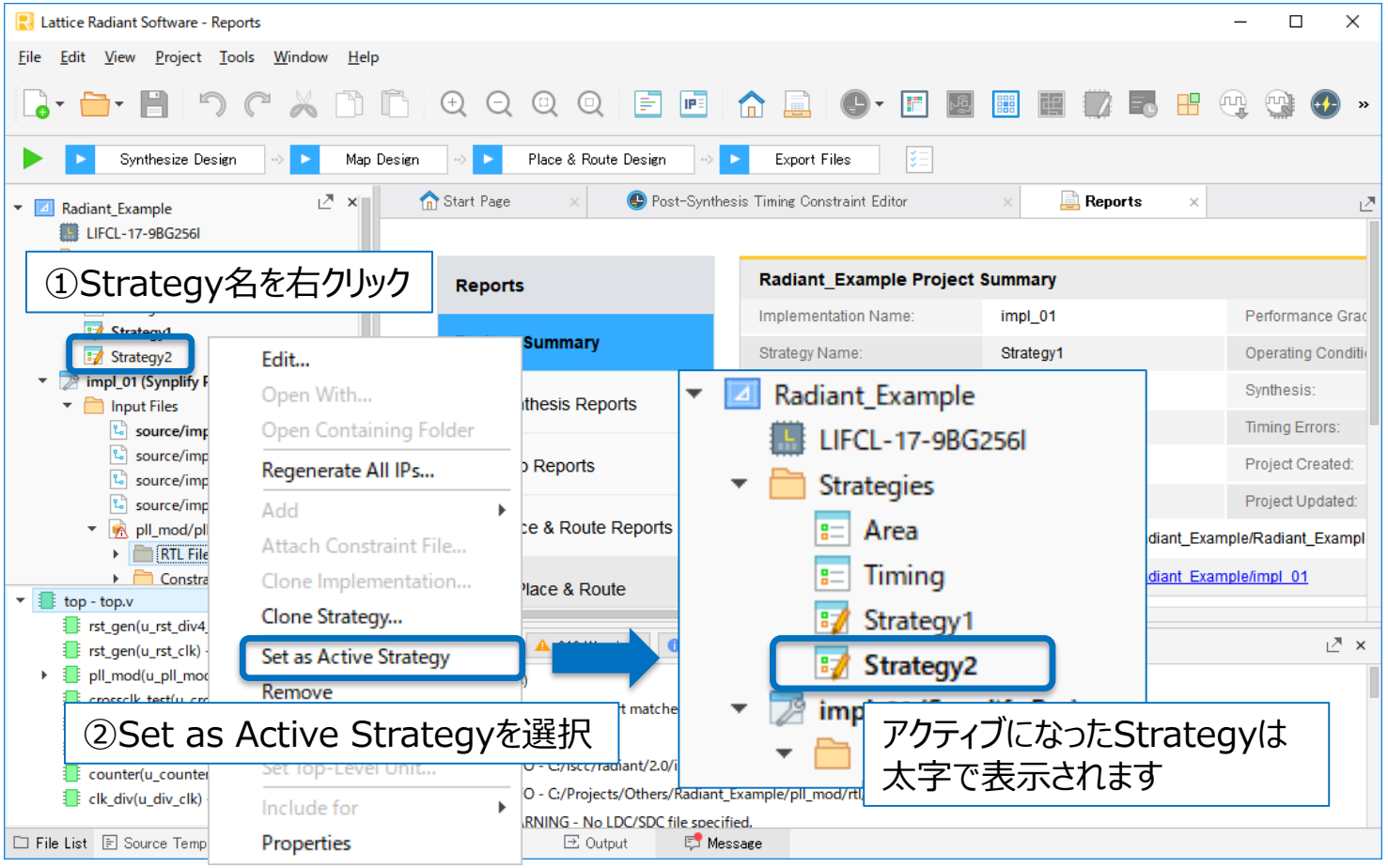
（参考）Strategyの追加

- Strategiesを右クリック > Add > New Strategyで新しいStrategyを作成することができます
- 設定の異なるStrategyを複数用意し、切り替えてコンパイル結果を比較することができます



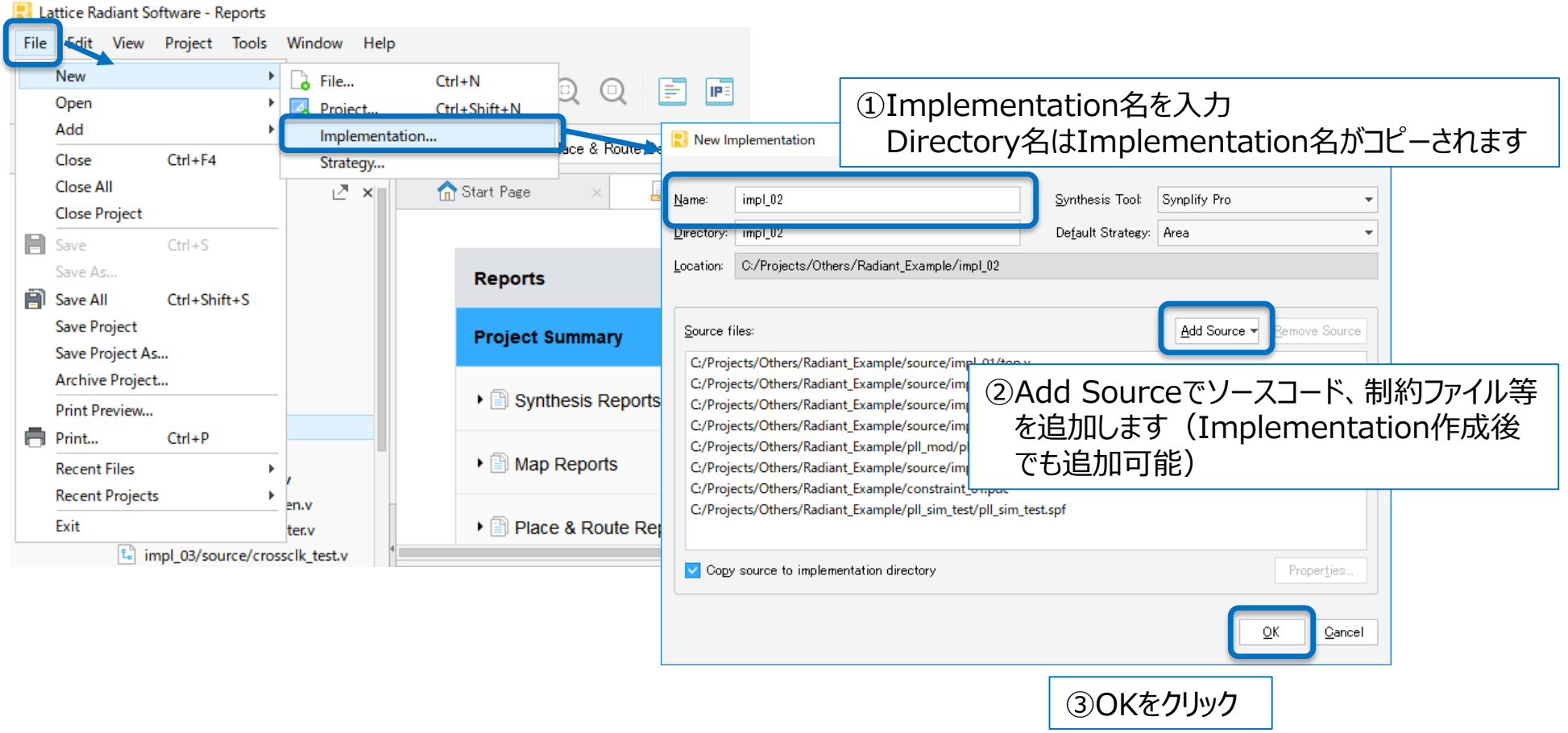
（参考）Strategyの追加

■ アクティブなStrategyを切り替えるにはStrategy名を右クリックし、”Set as Active Strategy”を選択します



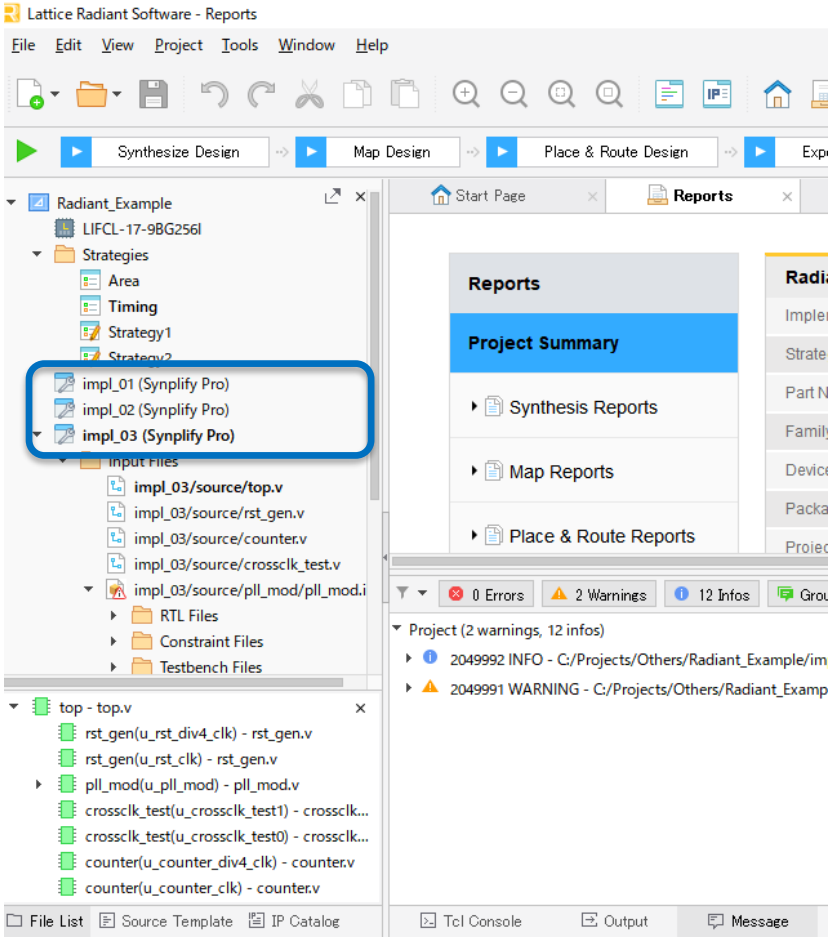
（参考）Run Manager

- 複数のImplementation（サブProject）を作成し、それぞれに異なるStrategyや制約を設定し、同時にコンパイルを回すことができます
- まず、File ＞ New ＞ Implementationから新しいImplementationを作成します



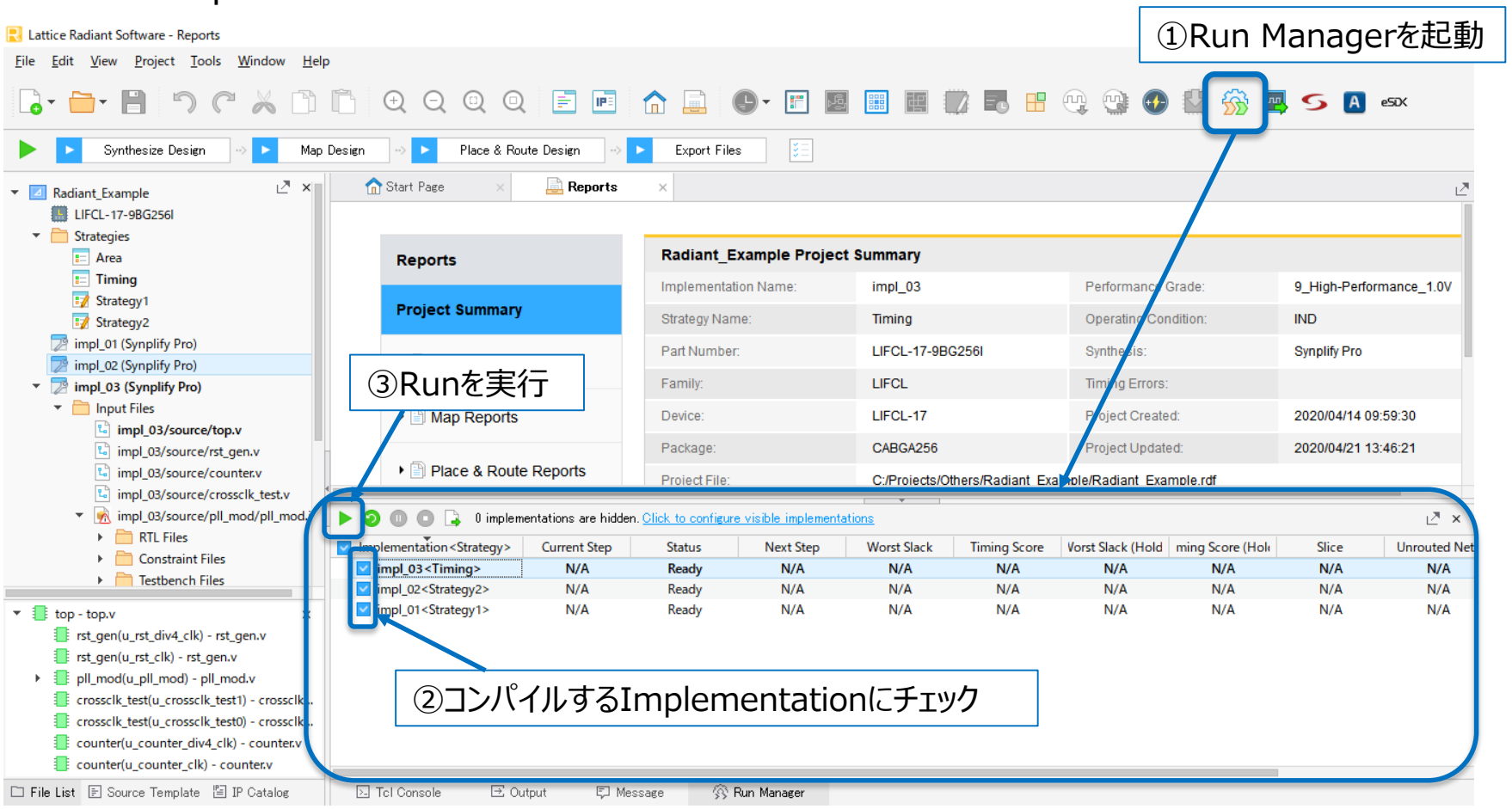
（参考）Run Manager

- 作成したImplementationがProject Viewにインポートされます
- アクティブなImplementationを切り替えてImplementation毎にStrategyやソースコード、制約設定を変更することができます



（参考）Run Manager

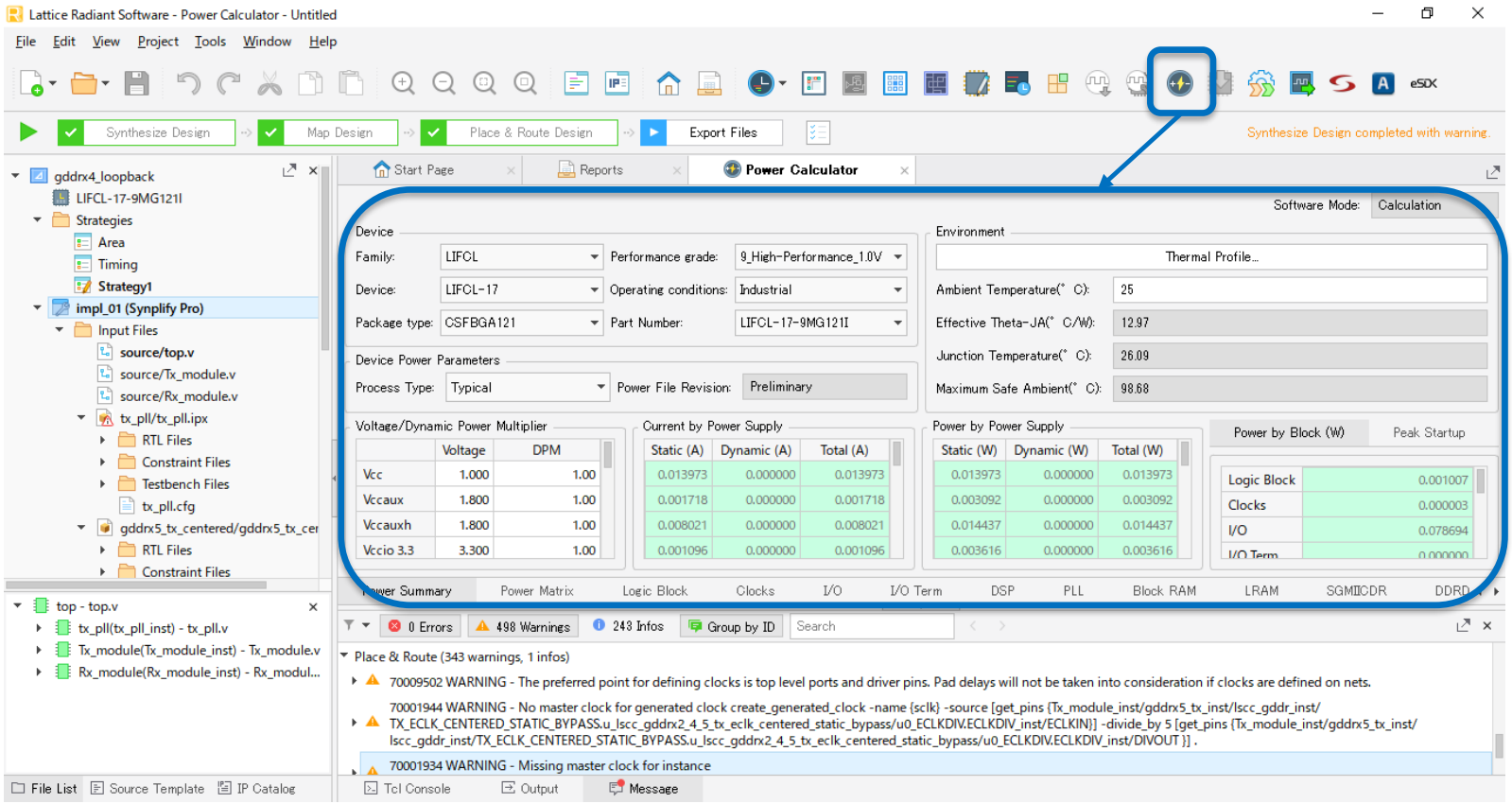
- Tools > Run Manager またはToolbarのRun Managerアイコン からRun Managerを起動します
- コンパイルを実行するImplementationにチェックを入れ、Runボタンをクリックします
- レポートはImplementationを切り替えて確認します



5. 消費電力の見積もり

Power Calculator

- 消費電力の見積もりはPower Calculatorを使用します
- Power Calculatorには実際のデザインコンパイル結果を用いるCalculationモードと、使用リソースを概算で入力して見積もるEstimationモードが用意されていますが、ここではCalculationモードについて説明します
- Tool > Power Calculator もしくはToolbarのPower Calculatorアイコン から起動します



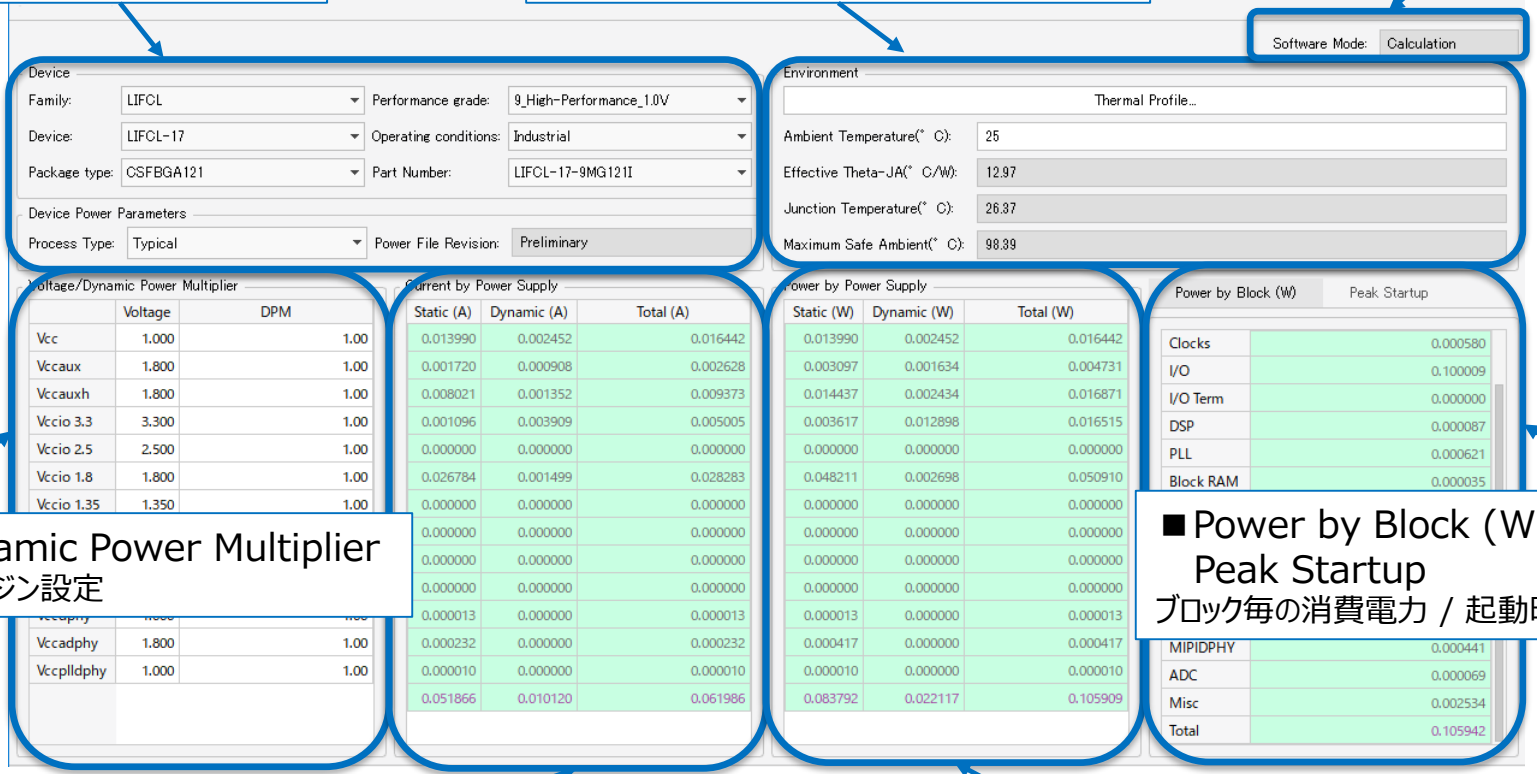
Power Calculator

■ Power CalculatorのGUIは以下のようになっています

■Device
見積もり対象デバイスの情報、設定入力

■Environment
周囲温度、実装条件等の入力及び
ジャンクション温度、最大周囲温度等の表示

■見積もりモード表示



■Voltage/Dynamic Power Multiplier
電源電圧設定とマージン設定

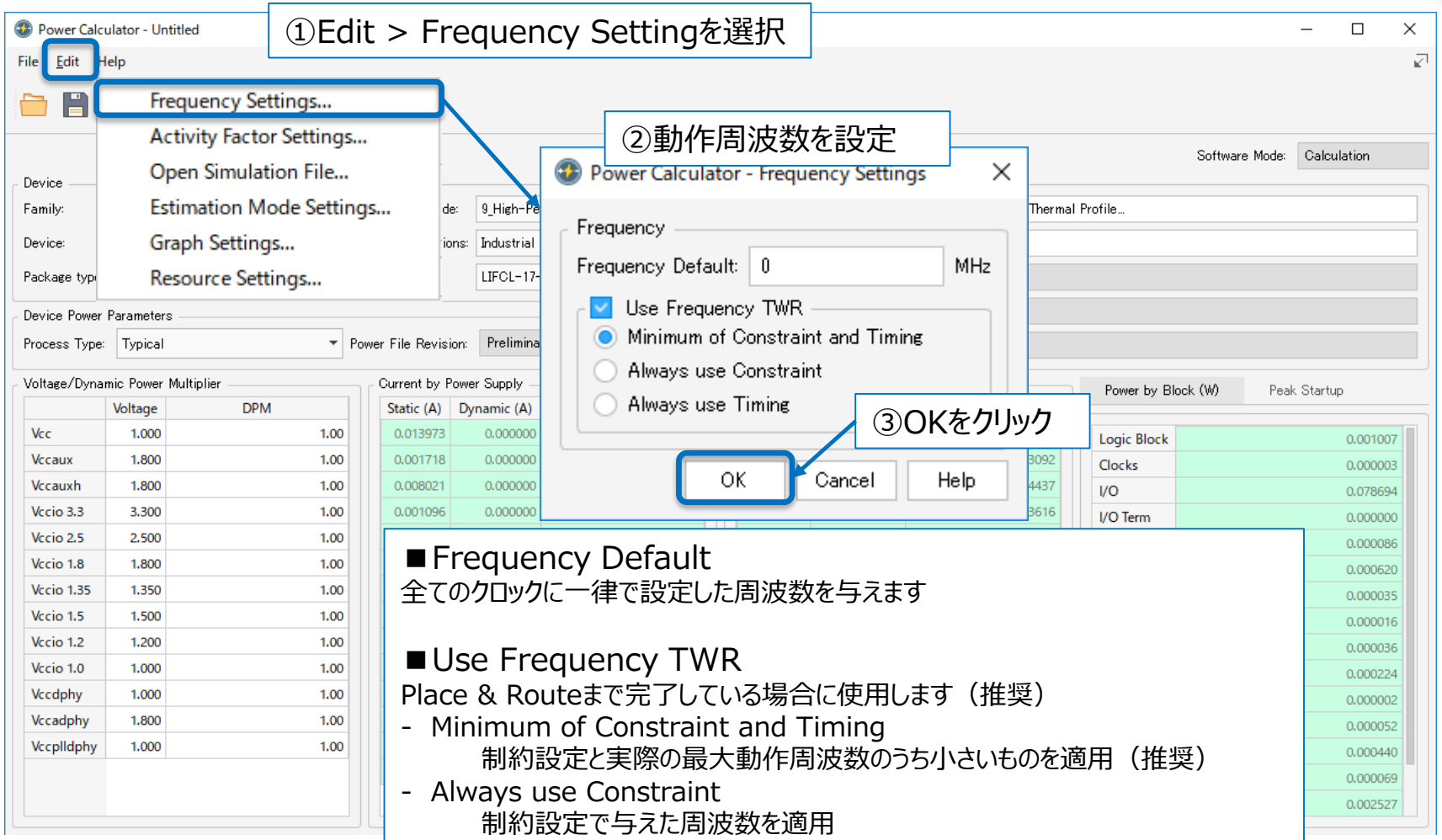
■Power by Block (W) / Peak Startup
ブロック毎の消費電力 / 起動時ピーク電流

■Current by Power Supply
電源電圧毎の消費電流

■Power by Power Supply
電源電圧毎の消費電力

Power Calculator

■ 以下のフローで動作周波数設定を行います

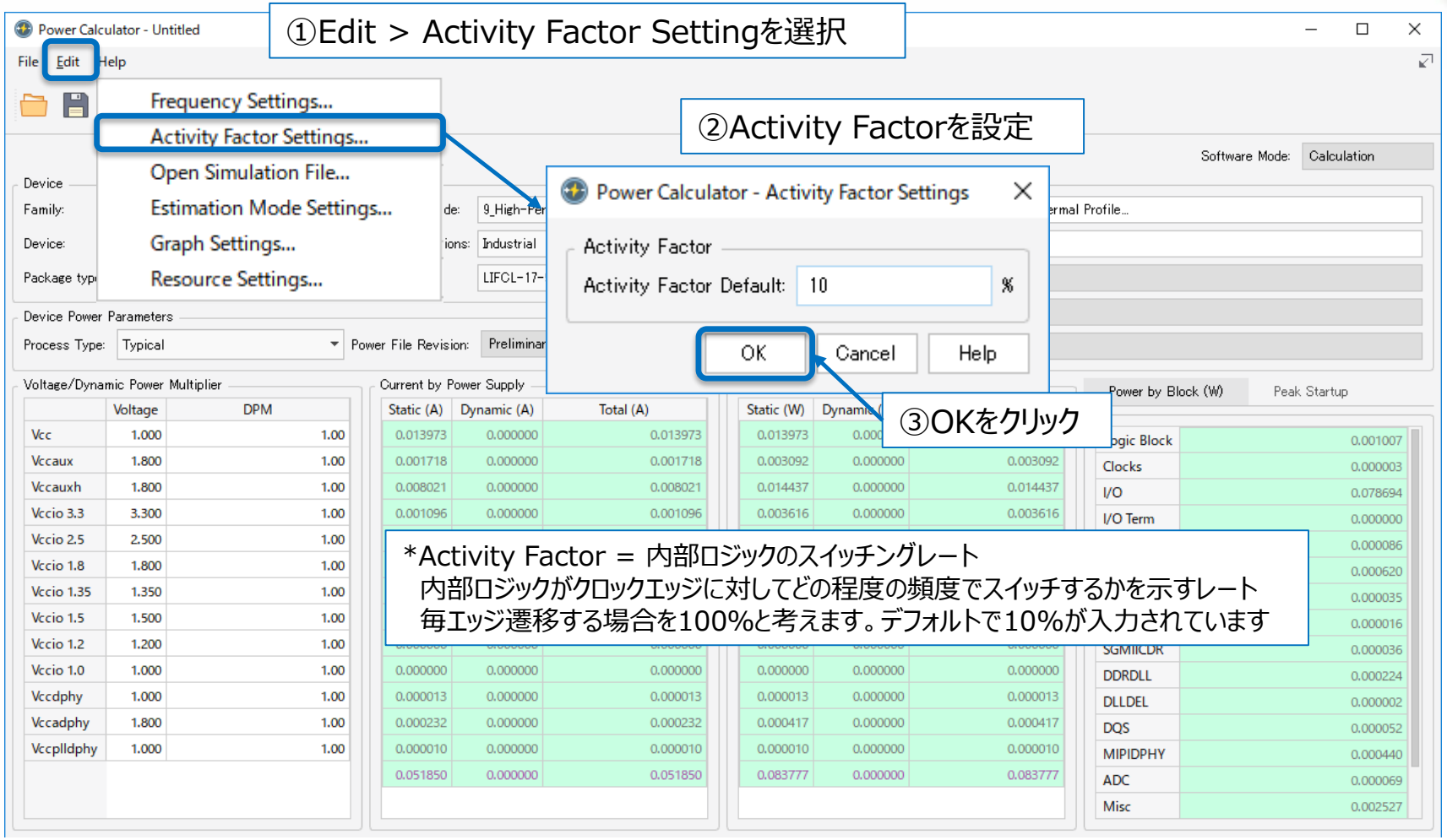


■Frequency Default
全てのクロックに一律で設定した周波数を与えます

■Use Frequency TWR
Place & Routeまで完了している場合に使用します（推奨）
- Minimum of Constraint and Timing
  制約設定と実際の最大動作周波数のうち小さいものを適用（推奨）
- Always use Constraint
  制約設定で与えた周波数を適用
- Always use Timing
  最大動作周波数を適用

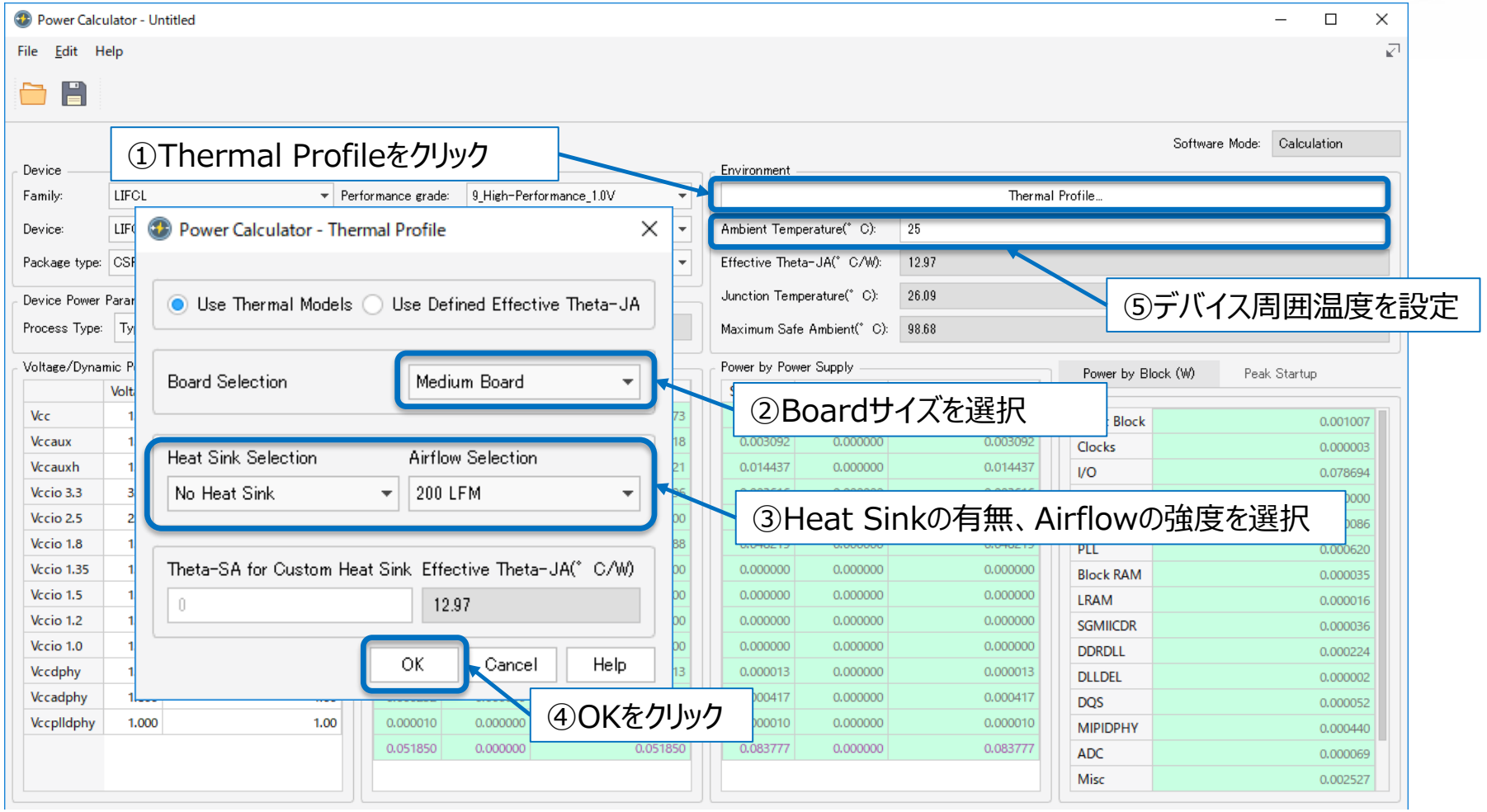
Power Calculator

■ 以下のフローでActivity Factorを設定します



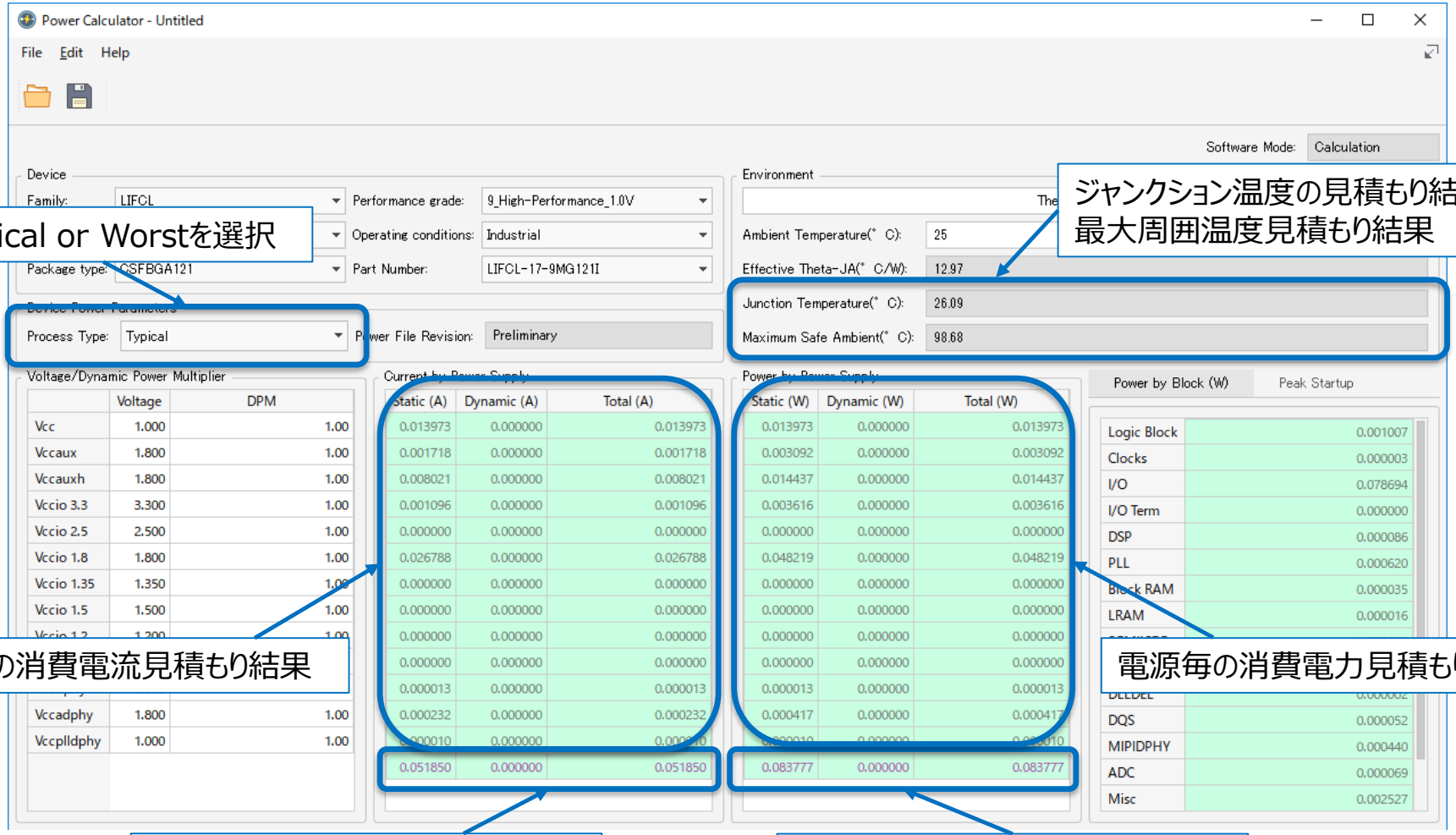
Power Calculator

■ 以下のフローで周囲環境を設定します



Power Calculator

■ デバイスプロセス条件のTypical / Worstを設定し、見積もり結果を確認します



①Typical or Worstを選択

ジャンクション温度の見積もり結果
最大周囲温度見積もり結果

電源毎の消費電流見積もり結果

電源毎の消費電力見積もり結果

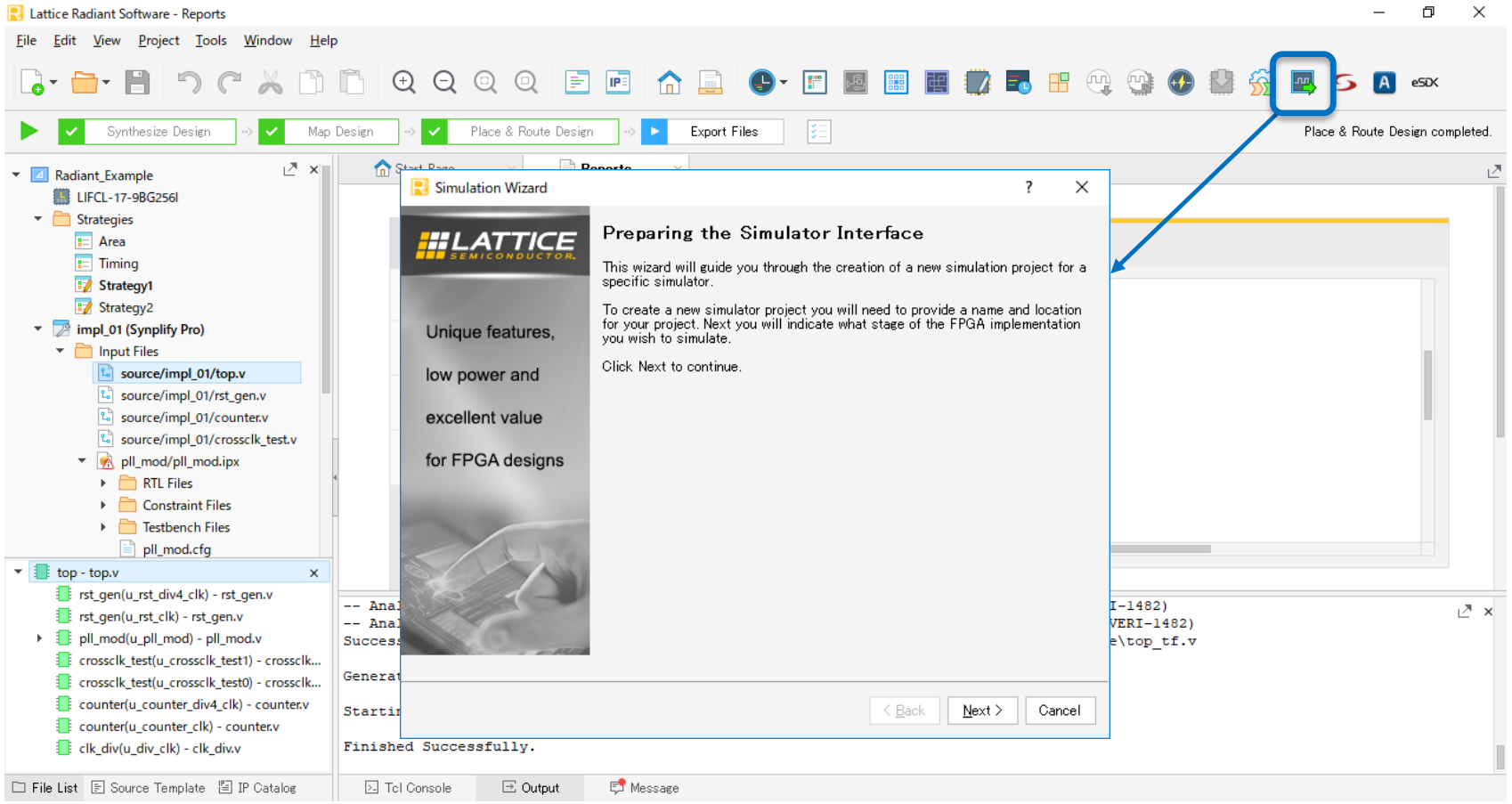
トータル消費電流見積もり結果

トータル消費電力見積もり結果

6. Functionシミュレーション

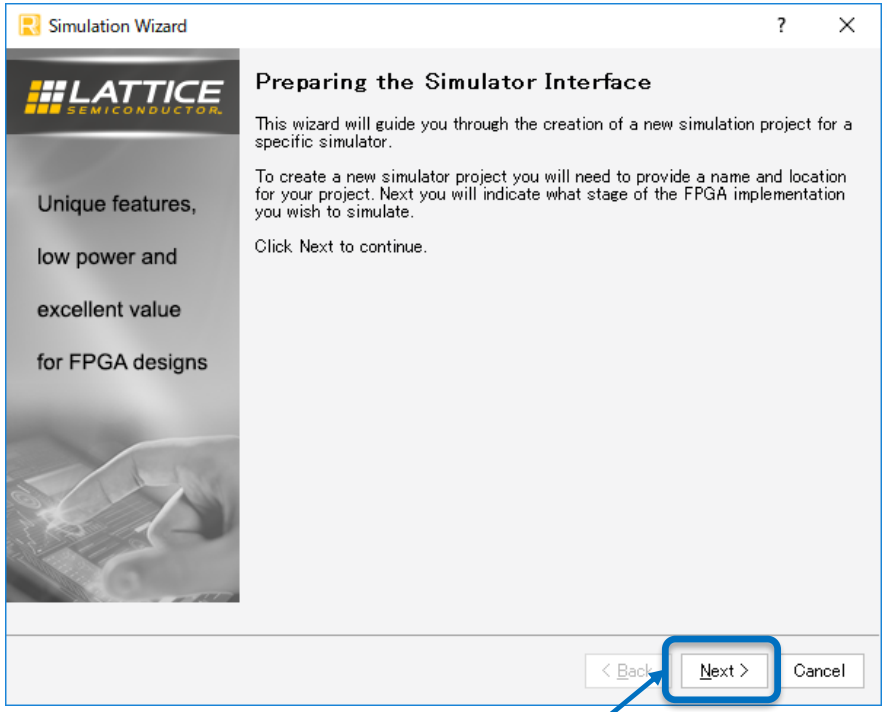
Simulation Wizardの実行

- RadiantではActive-HDLでのシミュレーションをSimulation Wizardを使用して実行することが出来ます
- Tool > Simulation Wizard またはToolbarからSimulation Wizardアイコン をクリックしてSimulation Wizardを起動します

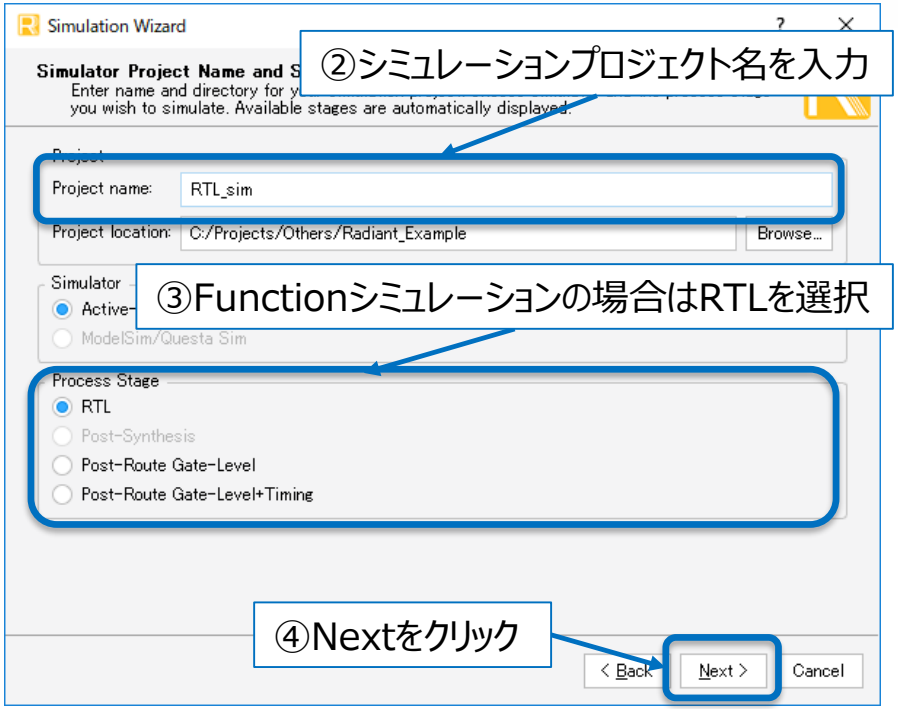


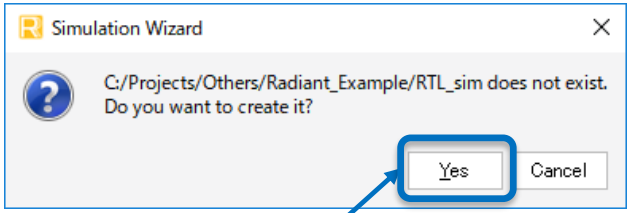
Simulation Projectの作成

■ 以下のフローに従い、シミュレーションプロジェクトを作成します



①Nextをクリック

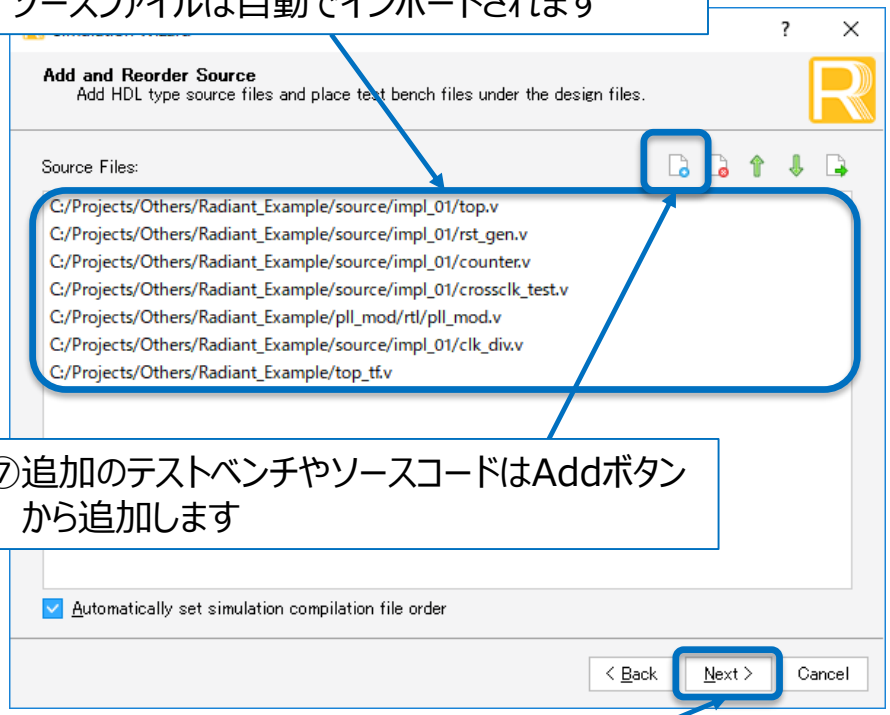




⑤Yesをクリック

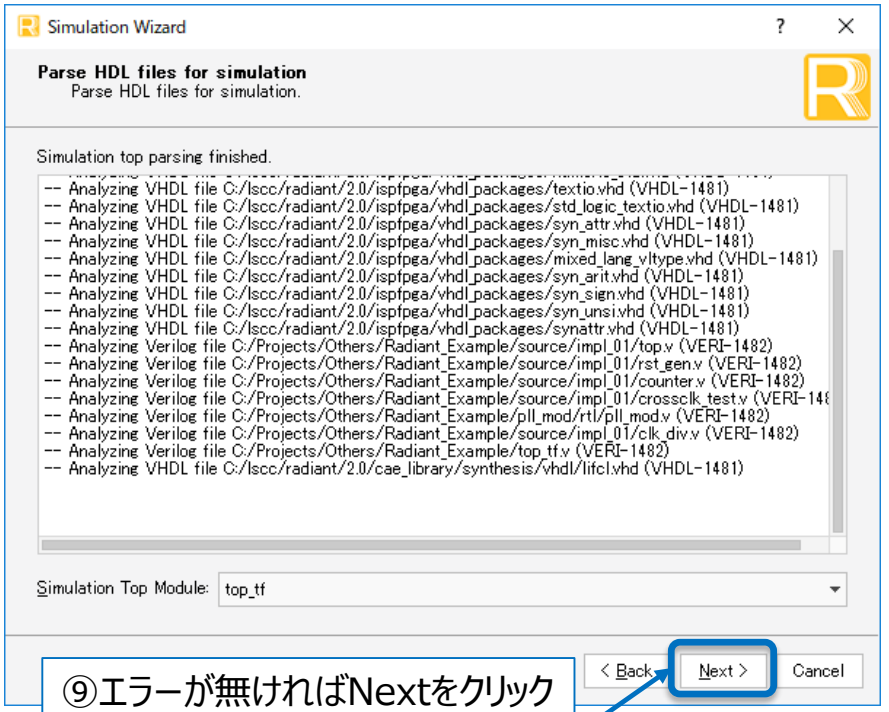
Simulation Projectの作成

⑥Radiantのプロジェクトにインポートされている
ソースファイルは自動でインポートされます



⑦追加のテストベンチやソースコードはAddボタン
から追加します

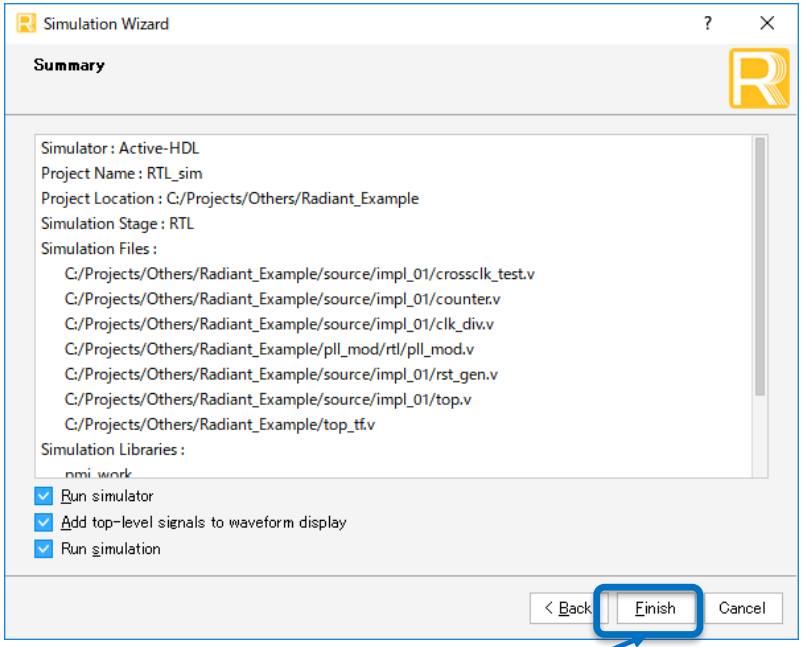
⑧Nextをクリック



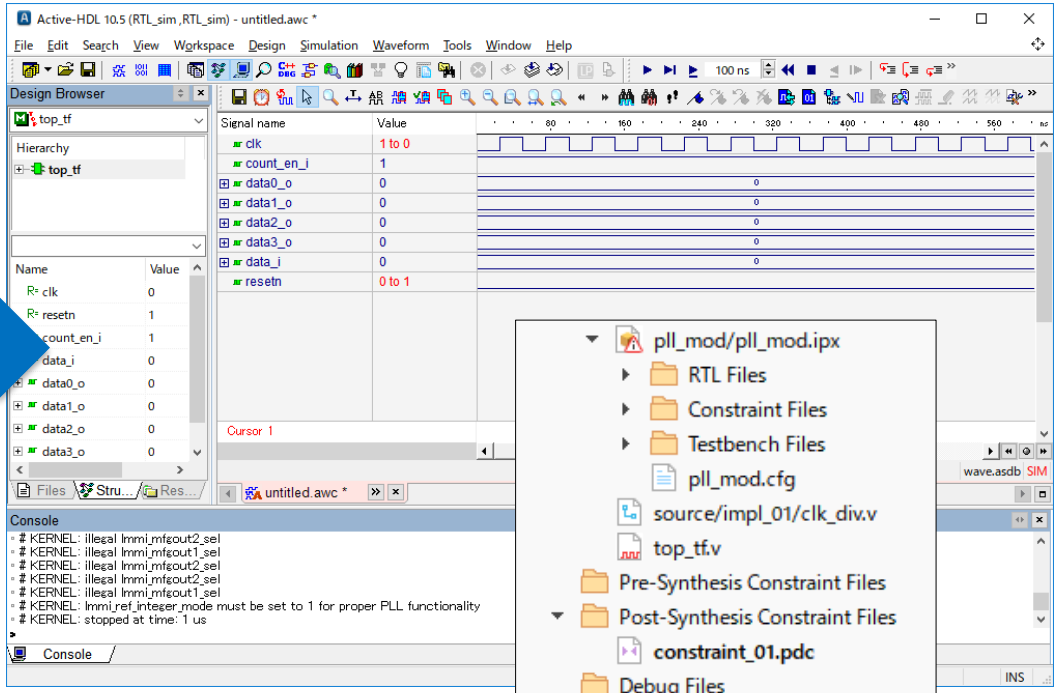
⑨エラーが無ければNextをクリック
エラーがある場合はソースコード
を修正して下さい

Simulation Projectの作成

- FinishをクリックするとActive-HDLが立ち上がります
- 作成したシミュレーションプロジェクトはRadiantのProject View内のScript Fileにインポートされ、このファイルをダブルクリックすることでいつでもシミュレーションプロジェクトを起動できます

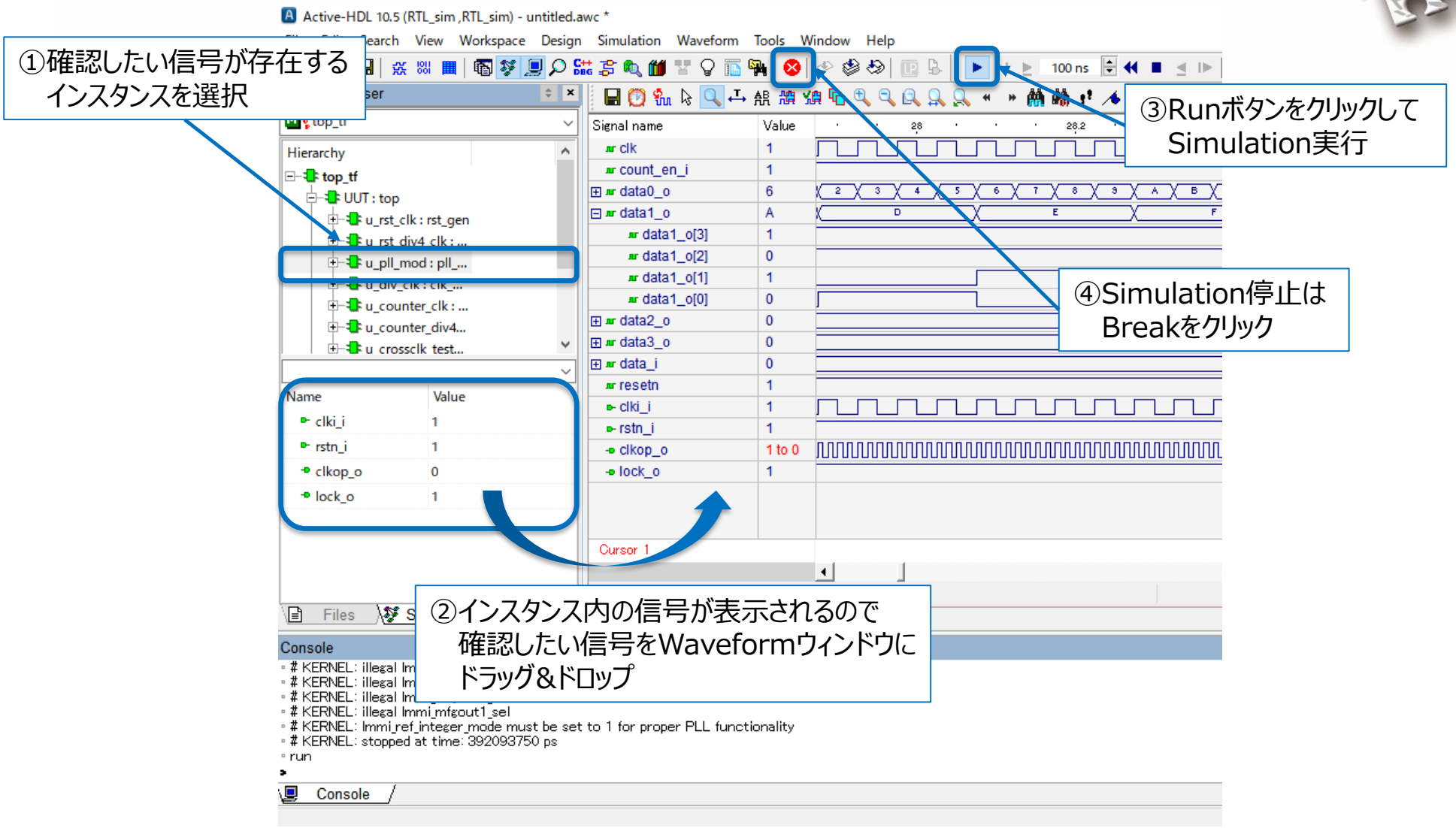


⑩Finishをクリック

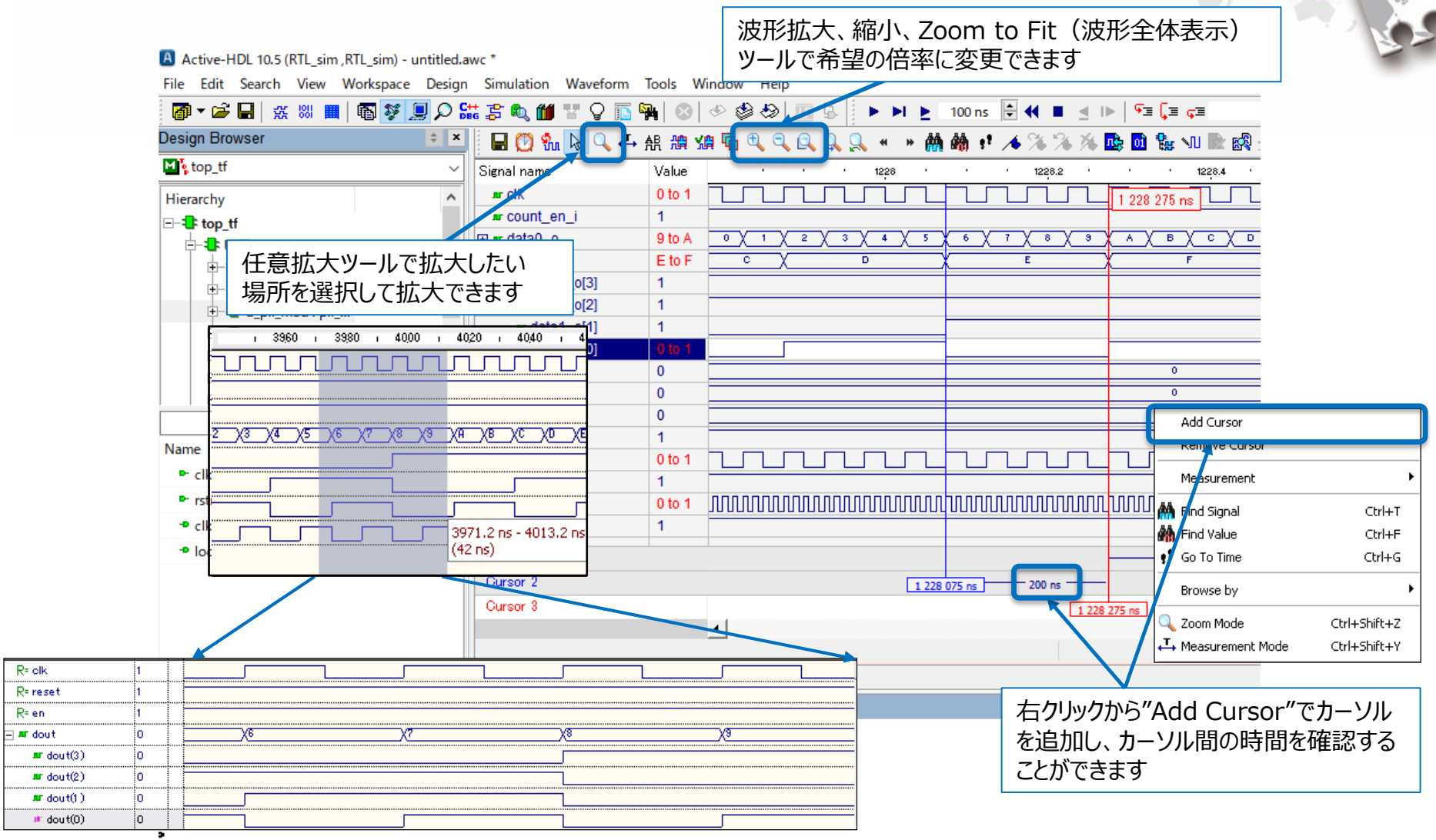


シミュレーション起動ファイル(.spf)がインポートされます

Active-HDLでの波形確認



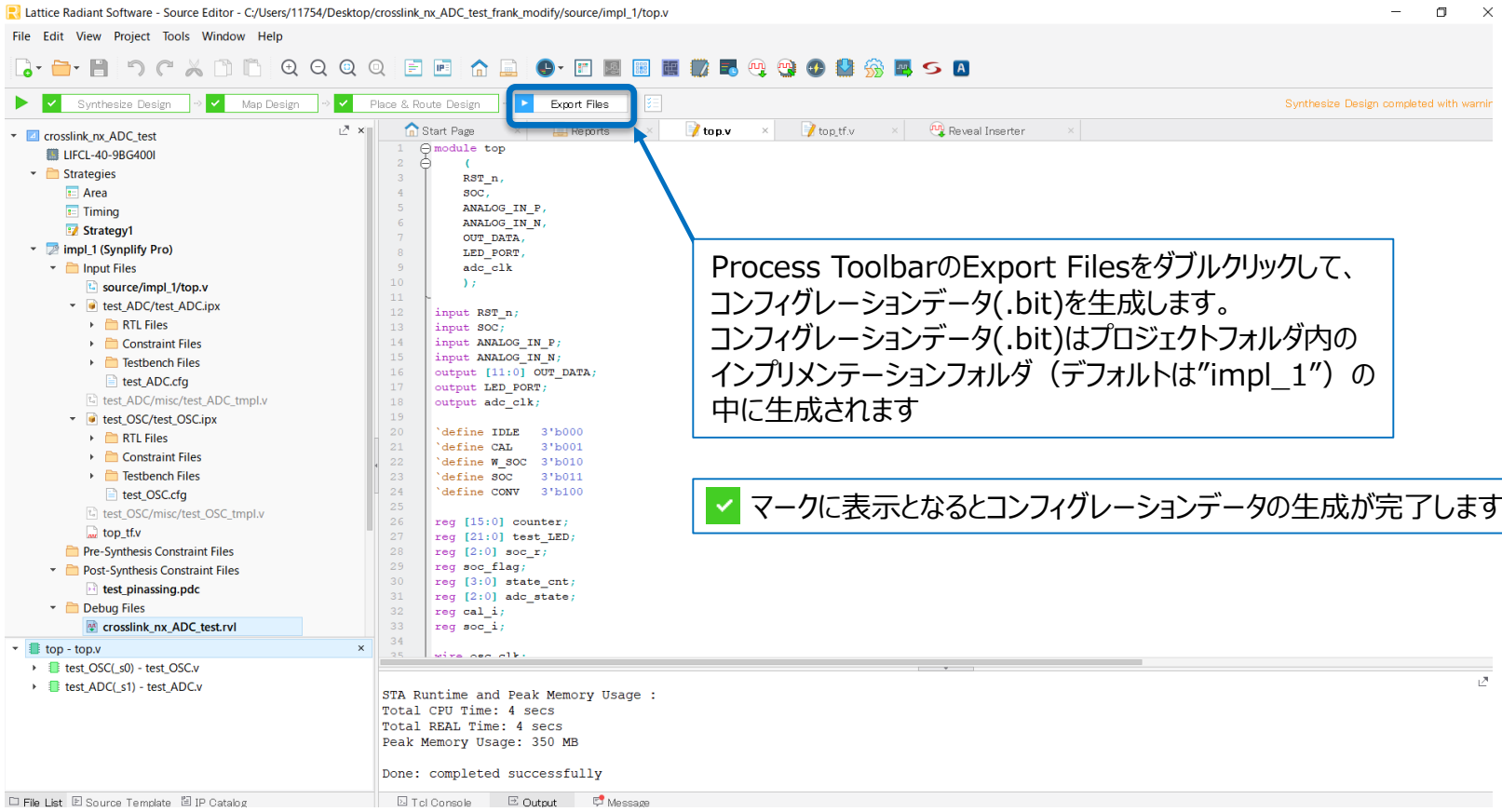
Active-HDLでの波形確認



8. デザイン書き込み

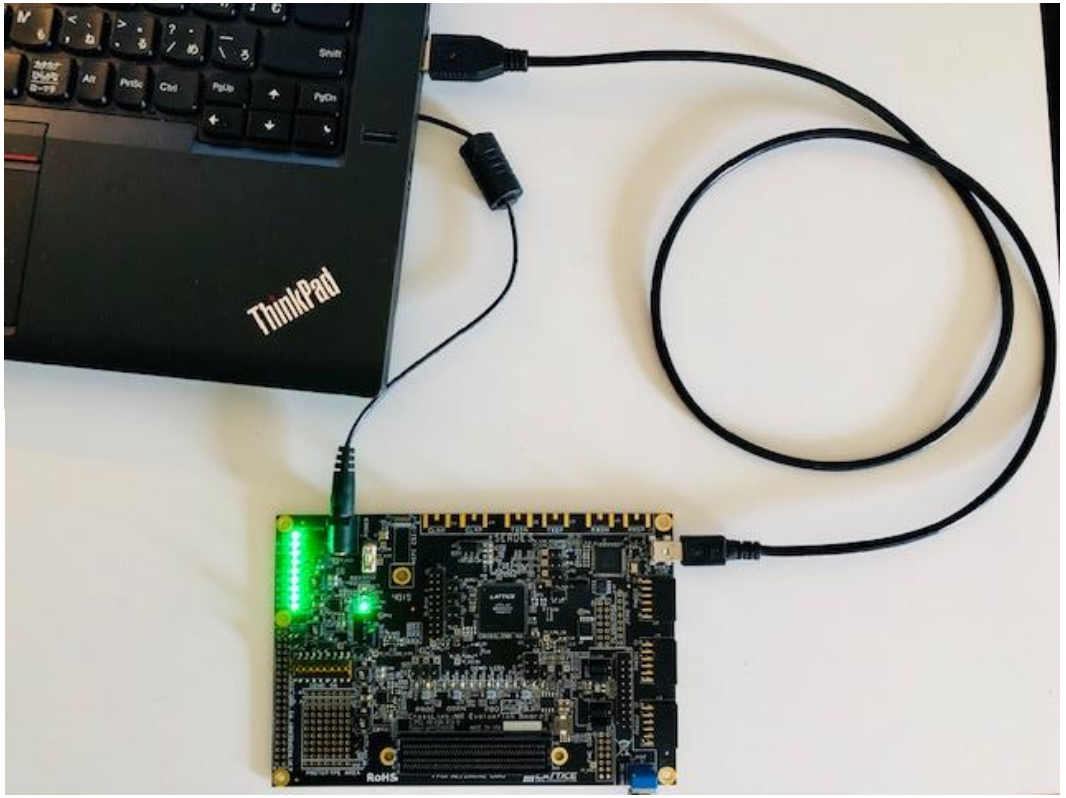
デザイン書き込み

- RADIANTでは、コンフィグレーションデータの生成から書き込みまで同ツール内で行うことが可能です
- ここでは、コンフィグレーションデータの生成と書き込みフローについて説明します



デザイン書き込み

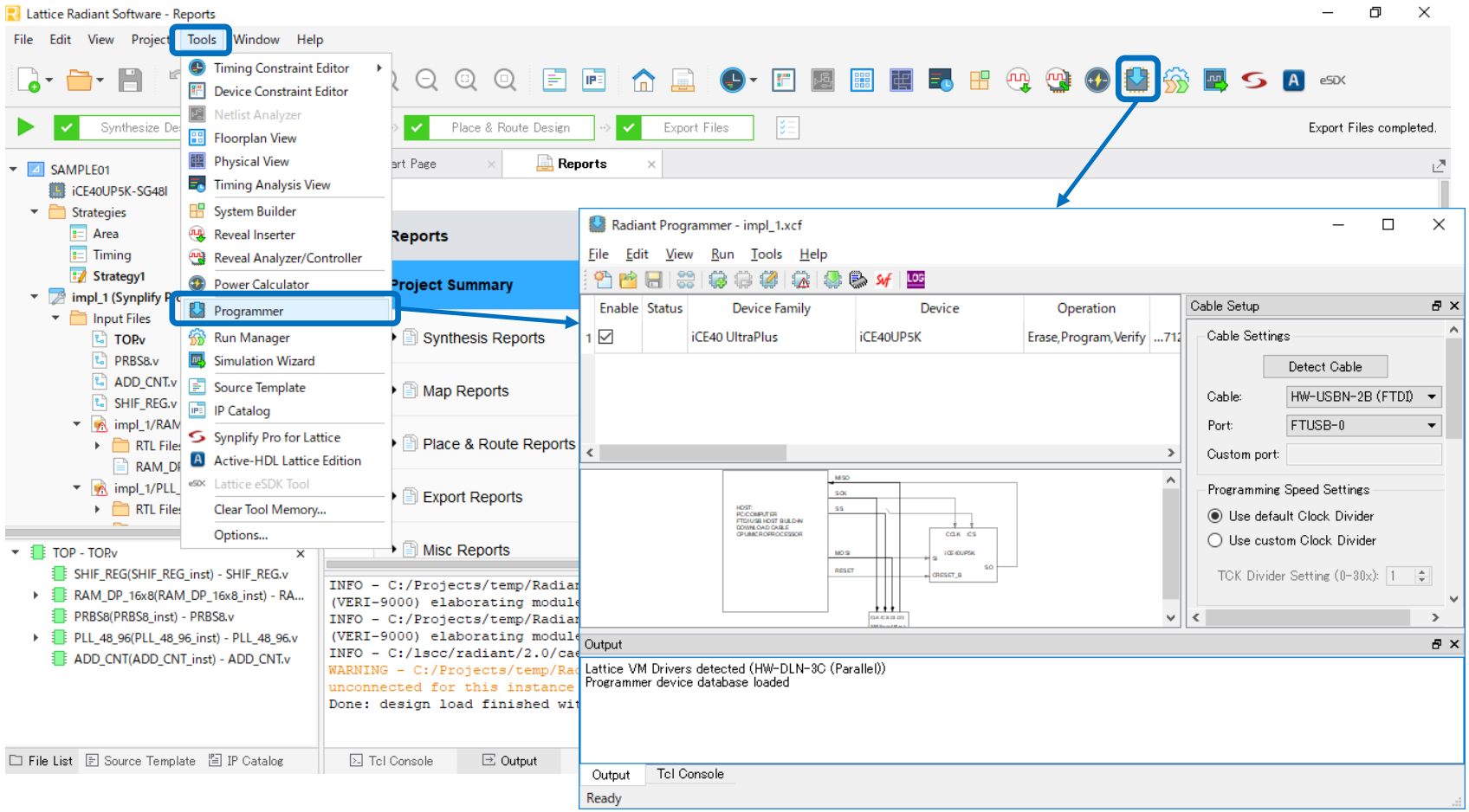
- PCとデバイスが実装されたボードを接続します
- ここでは、Lattice評価ボードを使用した場合を紹介していますが、お客様ボードでの書き込みフローも操作方法は共通となります



お客様ボードの場合、USBダウンロードケーブル
HW-USBN-2B（JTAG/SPI/I2C対応）を使用します

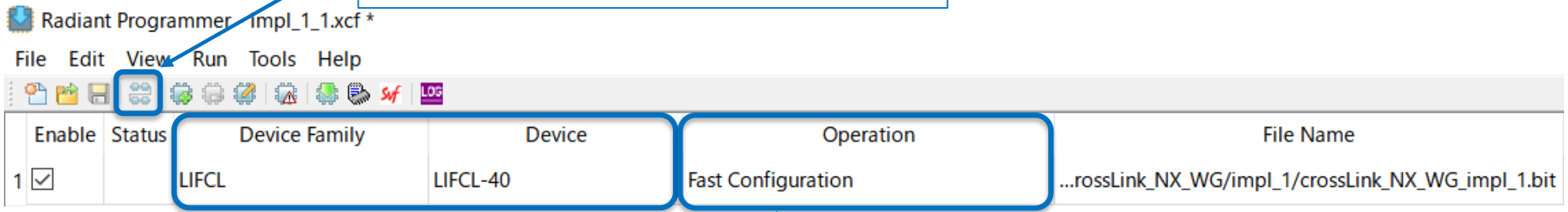
デザイン書き込み

■ Tools > Programmer もしくはToolbarからProgrammer アイコンをクリックし、Programmerを起動します



デザイン書き込み

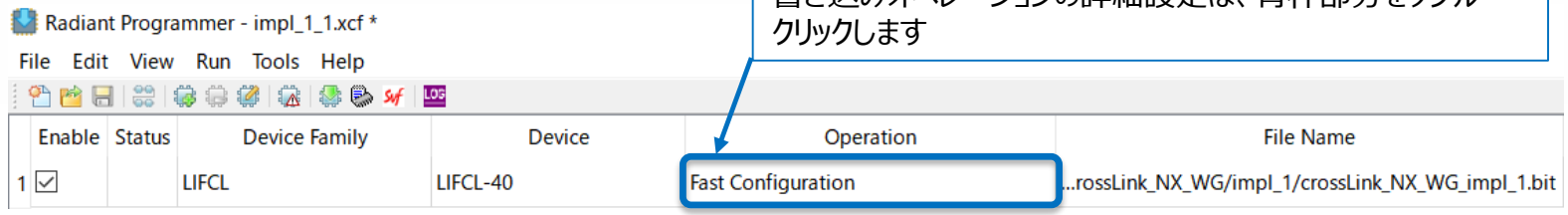
Programmerを起動すると自動で接続されているデバイスがスキャンされますが、正しくデバイスのスキャンが完了していない場合は、”Scan Device”をクリックします

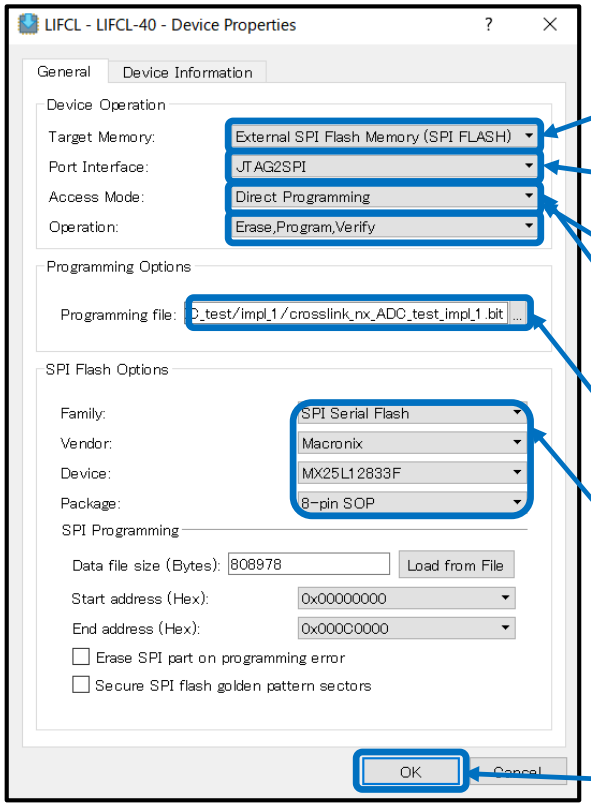


デバイスのスキャンが正常に完了すると、”Device Family”, ”Device”に接続されているデバイス型番が反映されます

デバイスへの書き込みオペレーションを設定します。書き込みオペレーション詳細については、次ページに記載しています

デザイン書き込み





書き込みオペレーションの詳細設定は、青枠部分をダブルクリックします

コンフィグレーションメモリの書き込み先を選択します。外部SPI-FLASH、内部NVCM、SRAMから選択可能です

書き込む際のI/Fを選択します。JTAG、又はSPIから選択可能です。デバイスとケーブルを接続しているI/Fに合わせて設定してください

アクセスモードを選択します。通常のコンフィグレーションデータ書き込みであれば、”Direct Programming”を選択します

書き込み時のオペレーション、その他実施したいオペレーションメニューから選択します。通常、外部SPI-FLASHに書き込む際は、”Erase Program Verify”を選択します

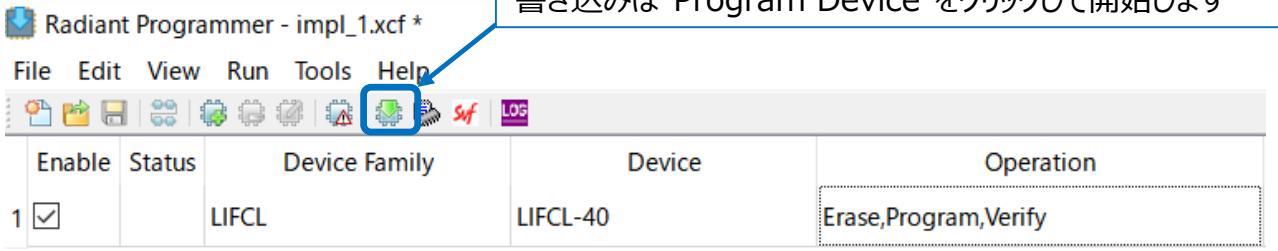
書き込むコンフィグレーションデータを選択します

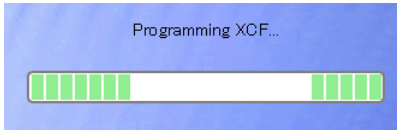
SPI-FLASHへ書き込む際は、型番に合わせて各種設定します。設定が誤っていると書き込みエラーが発生します

すべての設定が完了したらOKをクリックします

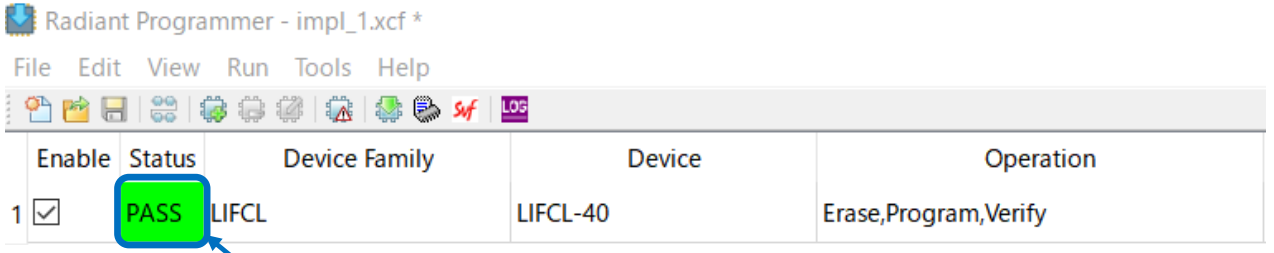
デザイン書き込み

書き込みは”Program Device”をクリックして開始します





書き込みには数十秒ほどかかります。

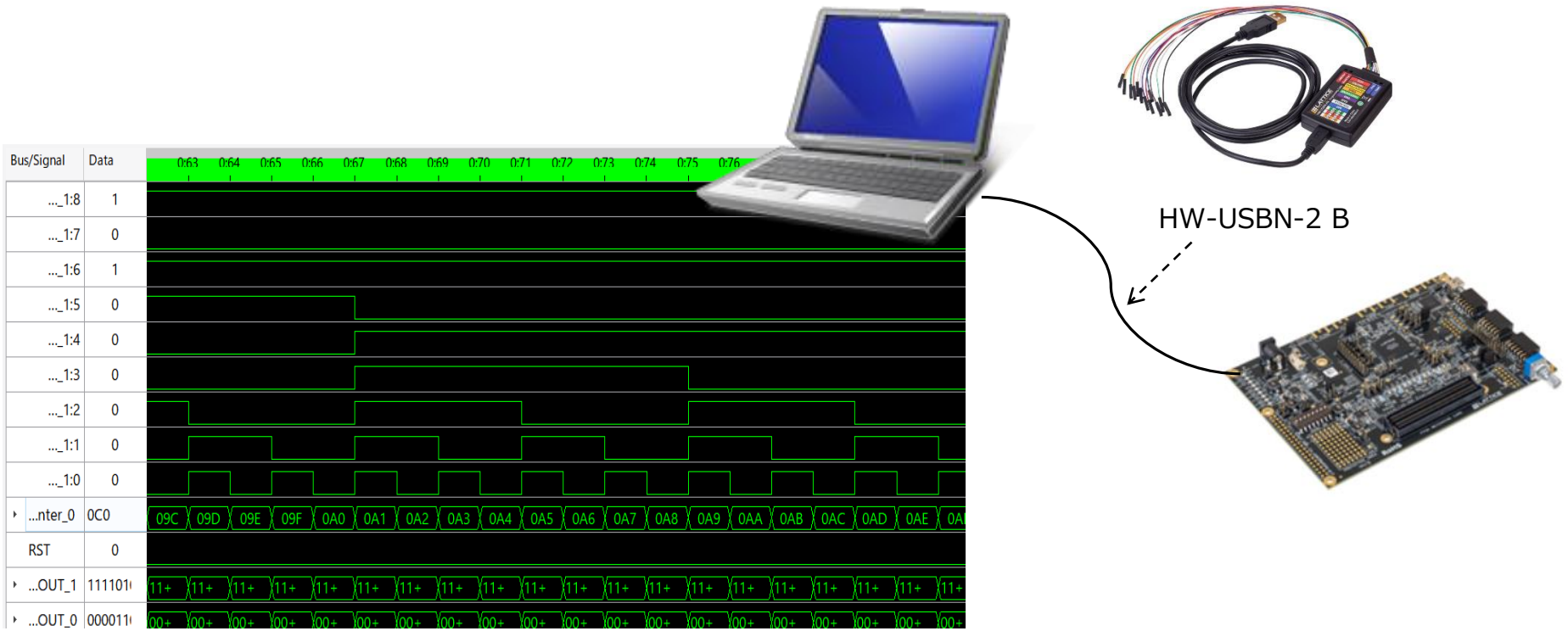


書き込みが正常に完了すると、”PASS”と表示されます

7. 実機上での内部波形観測

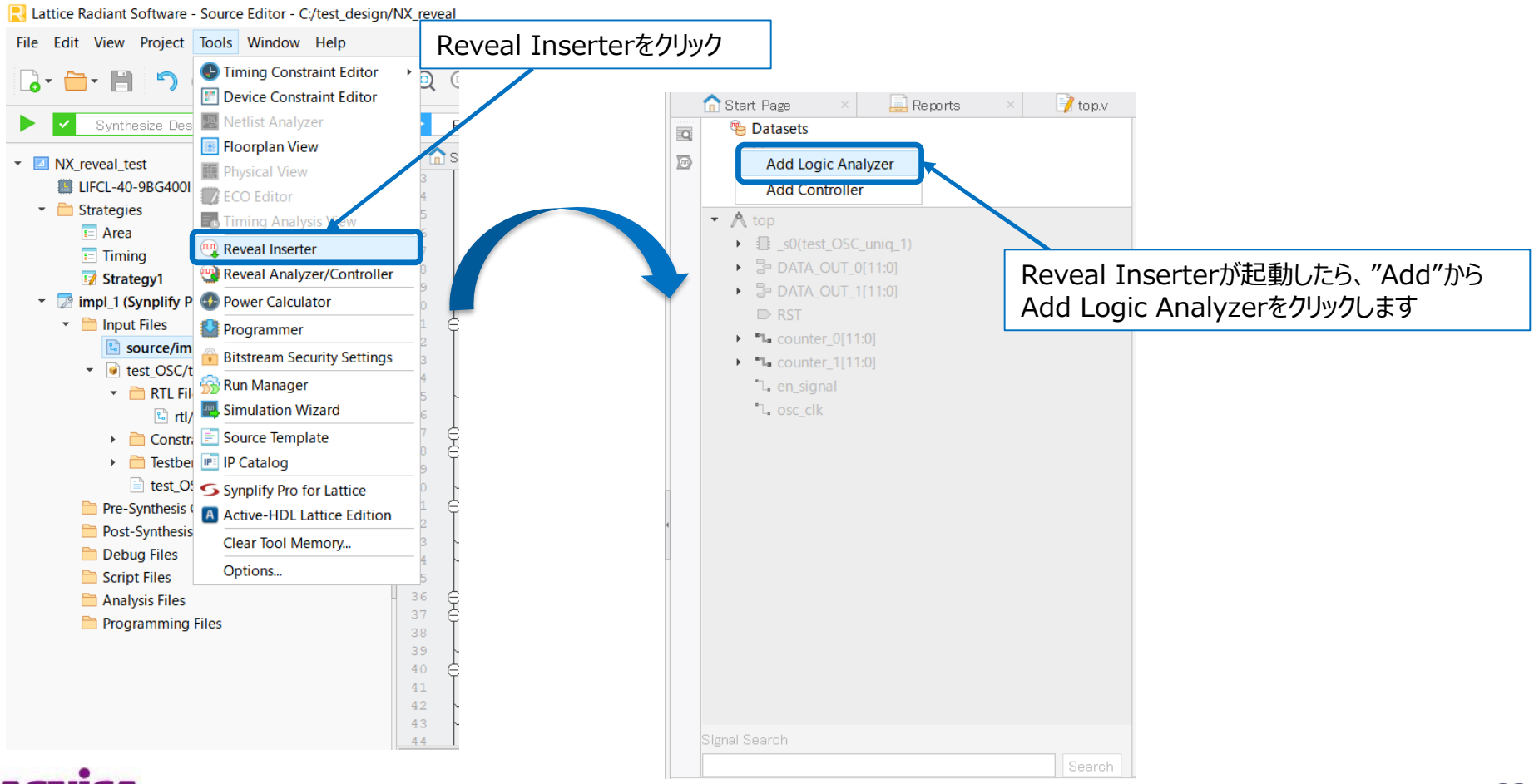
実機上での内部波形観測

- RADIANTでは、デバイス内部の実機波形を観測できるデバッグ用機能、Revealが用意されています
- Revealを使用した実機上での波形観測は、下図のようにPCと、デバイスが実装されたボードを接続します
- ここでは、Revealを使用した内部波形観測フローを紹介します



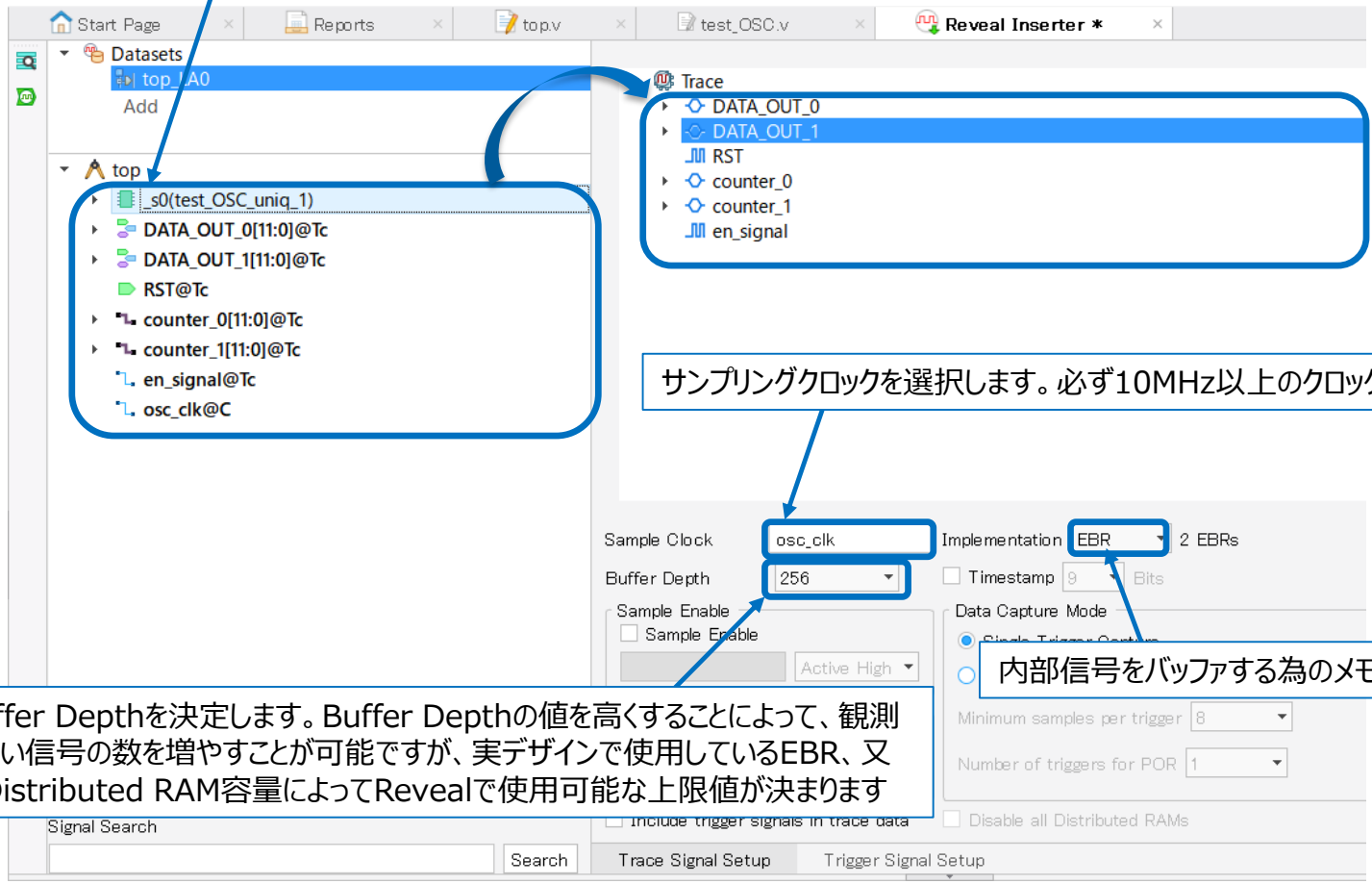
実機上での内部波形観測

- Revealを使用するには、デバイス内部にRevealコアを生成＆実装する必要がある為、Reveal InserterでRevealコアを生成します
- Tools > Reveal Inserter もしくはToolbarからReveal Inserterアイコン をクリックします



実機上での内部波形観測

観測可能な信号がリストで表示されます。これらの信号から観測した信号を選択し、Traceの下にドラッグ&ドロップします



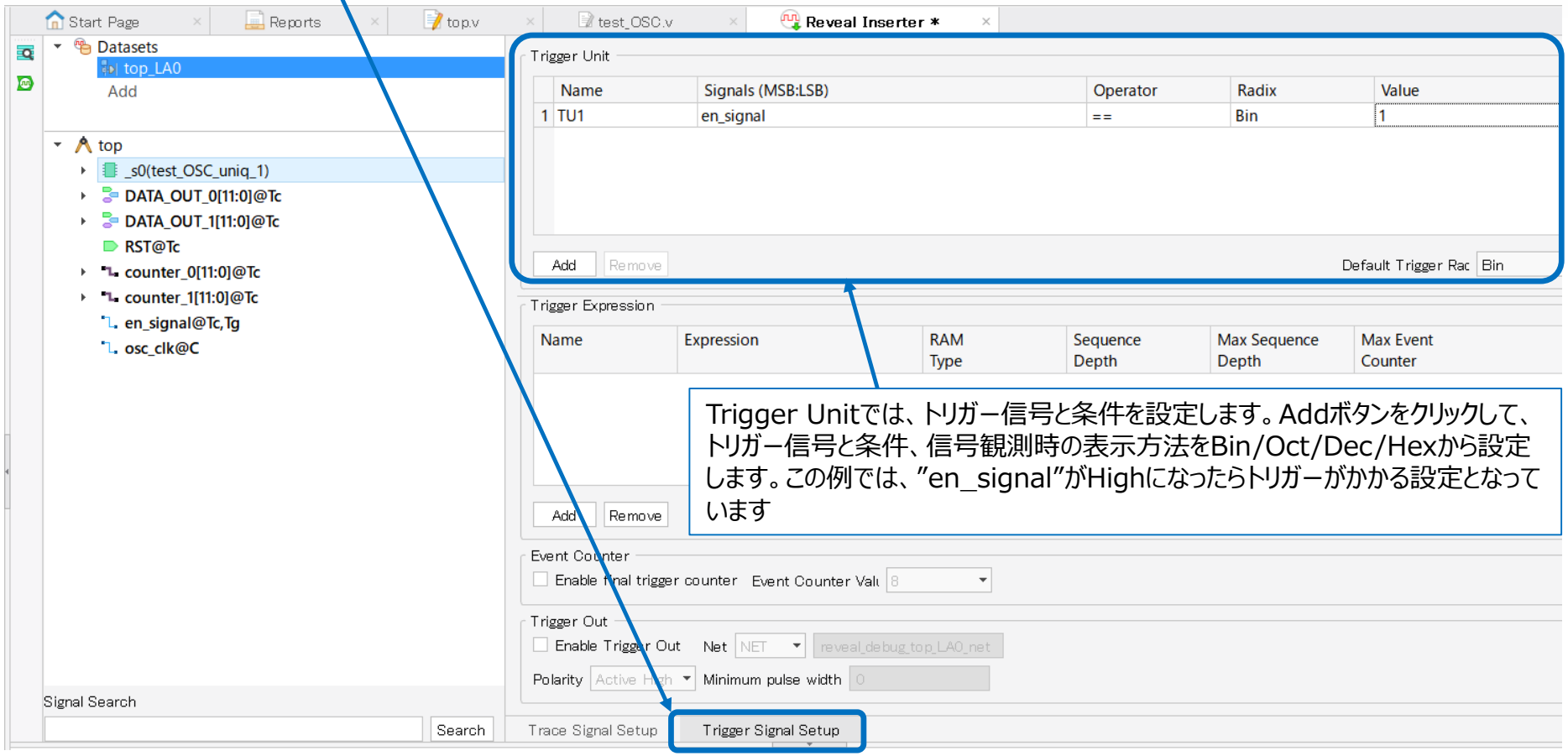
サンプリングクロックを選択します。必ず10MHz以上のクロックを選択してください

内部信号をバッファする為のメモリタイプを選択します

Buffer Depthを決定します。Buffer Depthの値を高くすることによって、観測したい信号の数を増やすことが可能ですが、実デザインで使用しているEBR、又はDistributed RAM容量によってRevealで使用可能な上限値が決まります

実機上での内部波形観測

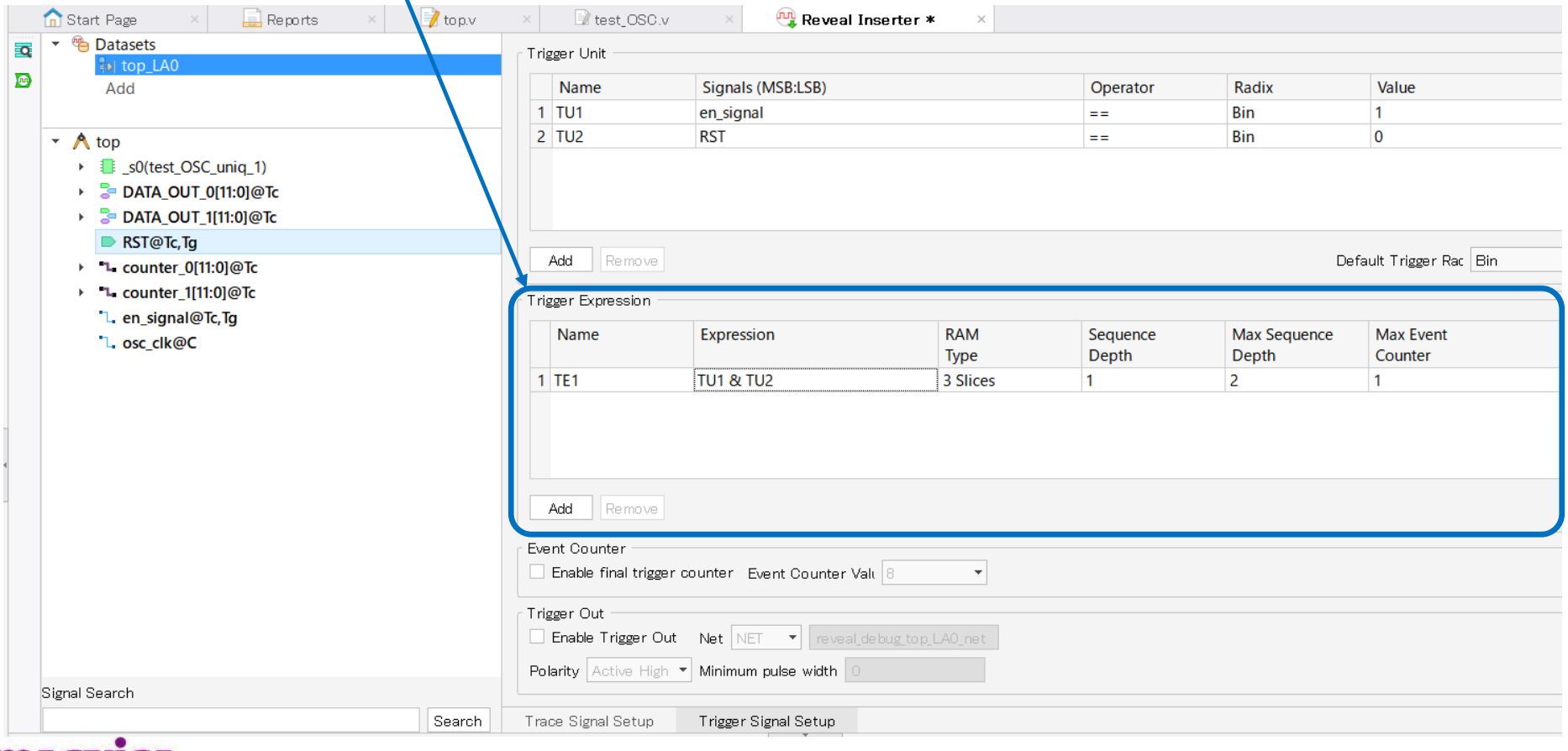
下タブをTrigger Signal Setupに切り替えます



Trigger Unitでは、トリガー信号と条件を設定します。Addボタンをクリックして、トリガー信号と条件、信号観測時の表示方法をBin/Oct/Dec/Hexから設定します。この例では、”en_signal”がHighになったらトリガーがかかる設定となっています

実機上での内部波形観測

Trigger Expressionでは、Trigger unitで設定した複数のトリガーを組み合わせて論理を組むことが可能です
この例では、”en_signal”がHighかつ、”RST”がLowの時にトリガーがかかる設定となっています
論理はExpressionのセルに直接記述します。論理の記述方法は次ページを参照ください
尚、Trigger Expressionでは、論理の組み合わせが不要な場合もExpressionのセルにTrigger Unitで設定したトリガー条件を入力する必要があります。この例では、TU1又はTU2のいずれかを記述します



実機上での内部波形観測

演算子	演算子の意味
&	前後のTUの and 論理
|	前後のTUの or 論理
^	前後のTUの xor 論理
!	後に続くTUの not 論理
()	括弧内の論理式を優先
next	前のTU成立後、次のサンプルで後のTUが成立
then	前のTU成立後、後のTUが成立
#	前に指定されたTUが、後に指定された回数成立
##	前に指定されたTUが、後指定された回数連続して成立

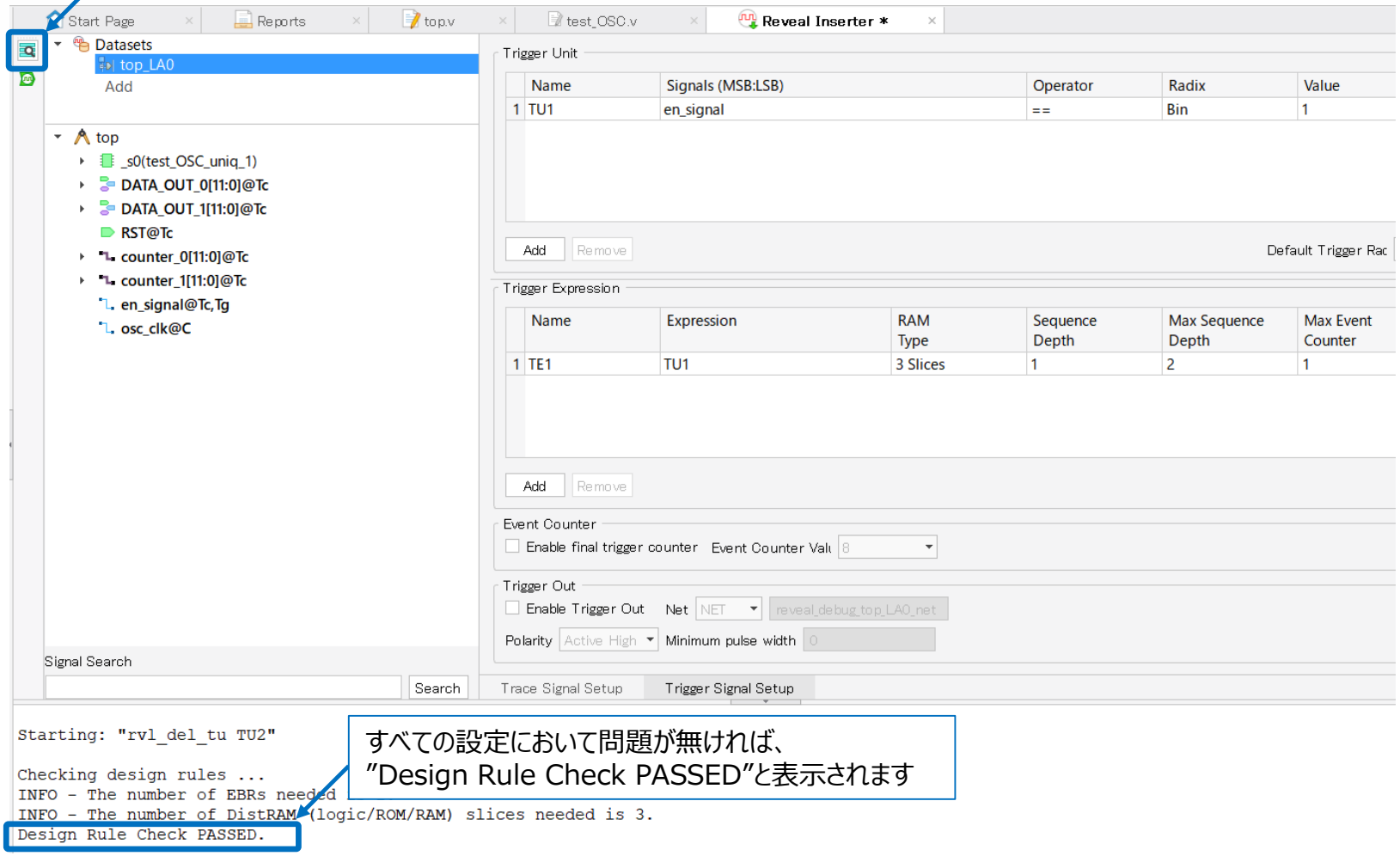
■Trigger Expression記述例

```
TU1 & TU2          : TU1とTU2が同時にTrueの場合にトリガ生成
TU1 | TU2          : TU1もしくはTU2がTrueの場合にトリガ生成
!TU3               : TU3がTrueでない場合にトリガ生成
TU1 & !TU4         : TU1がTrueかつTU4がTrueでない場合にトリガ生成
TU3 ^ TU1          : TU3もしくはTU1のどちらか一方がTrueの場合にトリガ生成
TU1 next TU2       : TU1がTrueになった次のサンプリングでTU2がTrueの場合にトリガ生成
TU1 then TU2       : TU1がTrueになった次のサンプリング以降にTU2がTrueの場合にトリガ生成
TU5 #2             : TU5が2回Trueになった場合にトリガ生成
TU5 ##2            : TU5が2回連続してTrueになった場合にトリガ生成
```

* #および##演算子で繰り返し回数を指定する場合は“Max Event Counter“セルの設定が必要
* thenおよびnext演算子でシーケンシャル・トリガを生成する場合は”Max Sequence Depth”セルの設定が必要

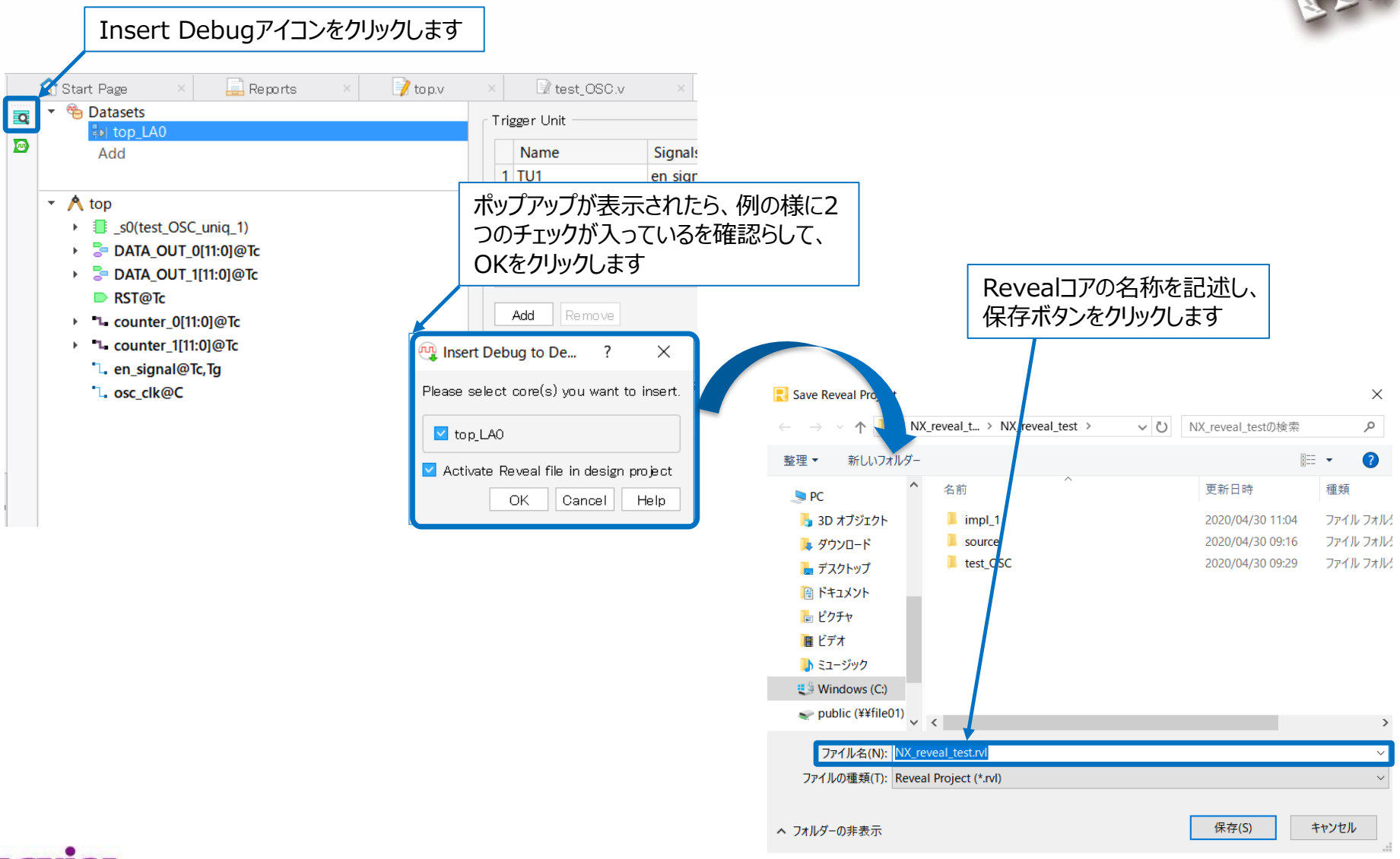
実機上での内部波形観測

すべての設定が完了したら、Design Rule Checkアイコンをクリックします



すべての設定において問題が無ければ、
”Design Rule Check PASSED”と表示されます

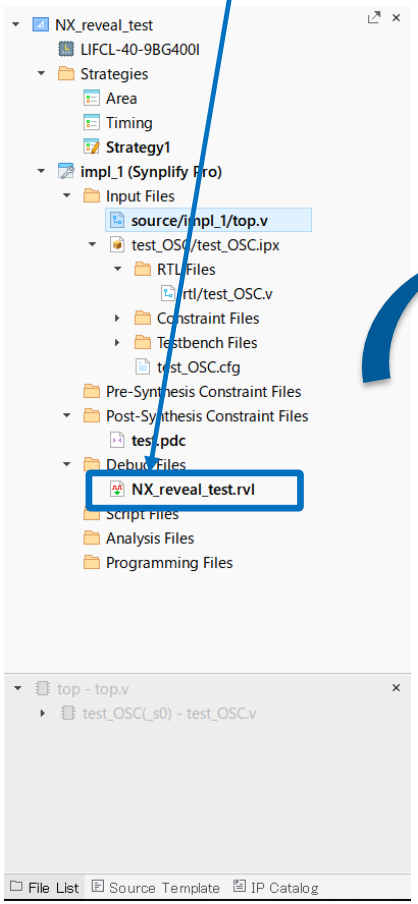
実機上での内部波形観測

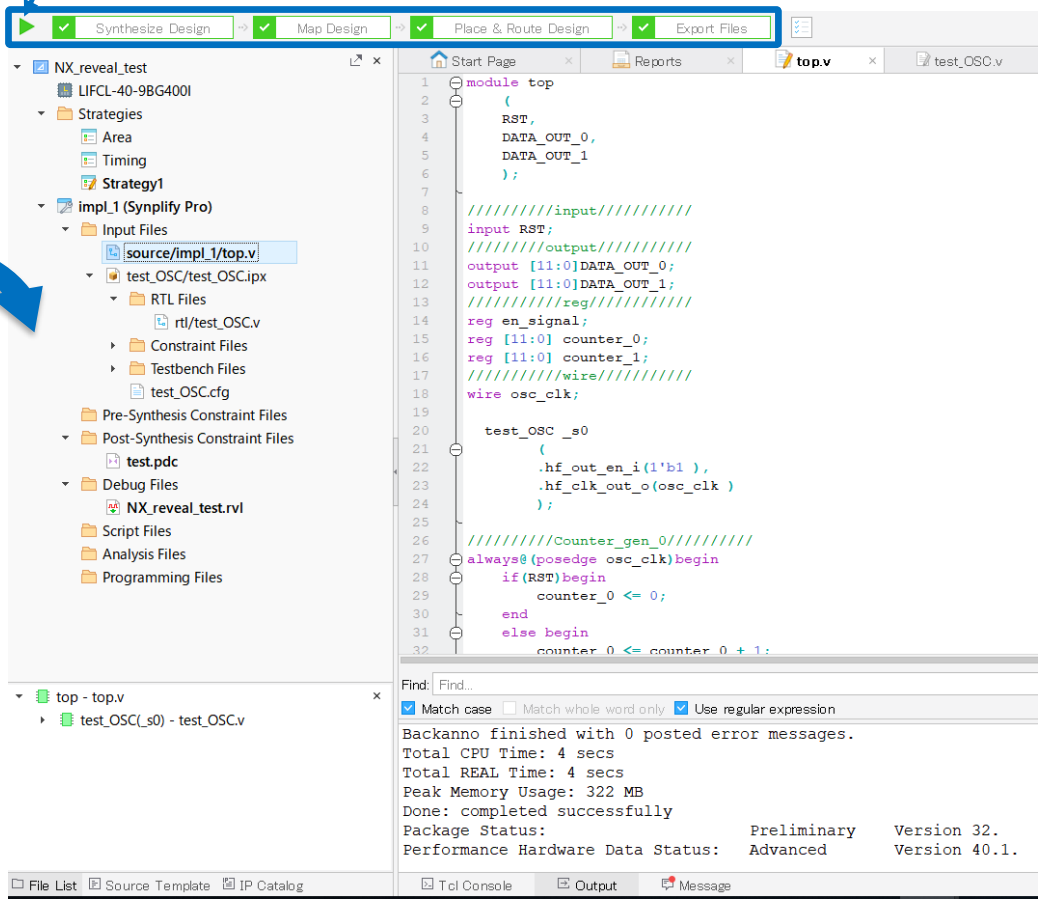


実機上での内部波形観測

File ListタブにRevealコアファイル(.rvl)が追加されている事を確認します

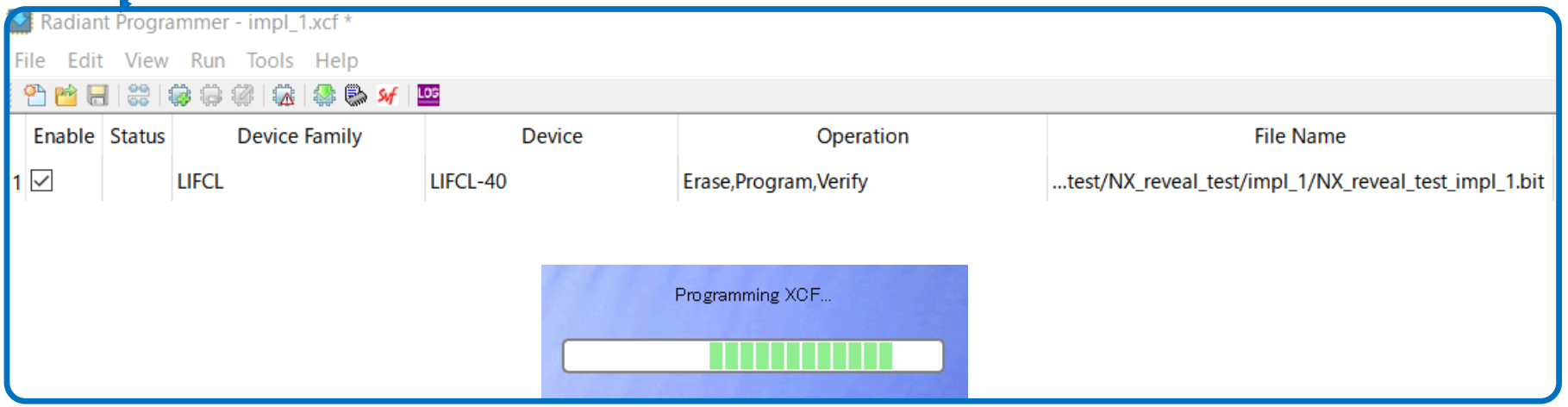
Export Filesプロセスまで完了させ、Revealコアが実装されたコンフィグレーションデータを生成します





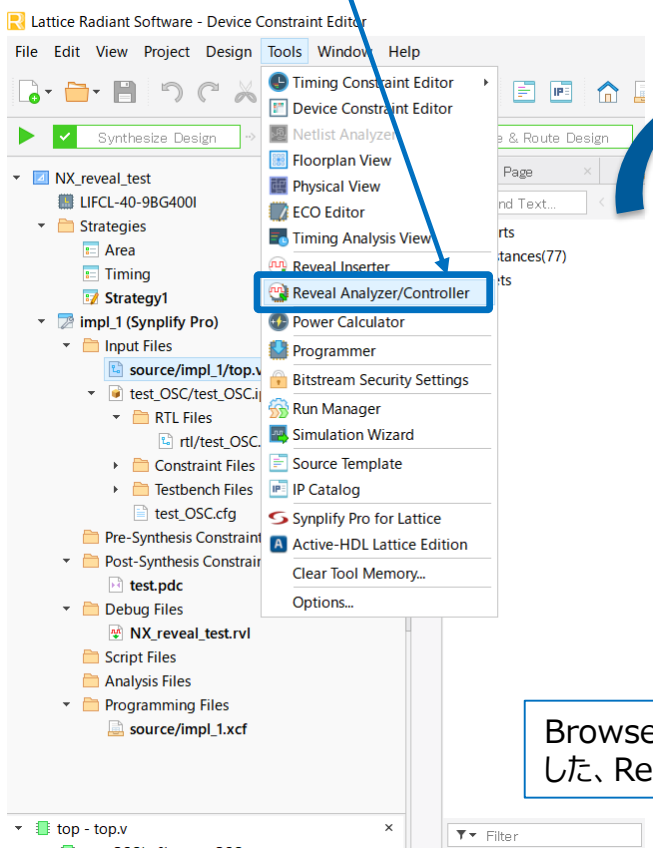
実機上での内部波形観測

Revealコアが実装されたコンフィグレーションデータをProgrammerを使用して書き込みます



実機上での内部波形観測

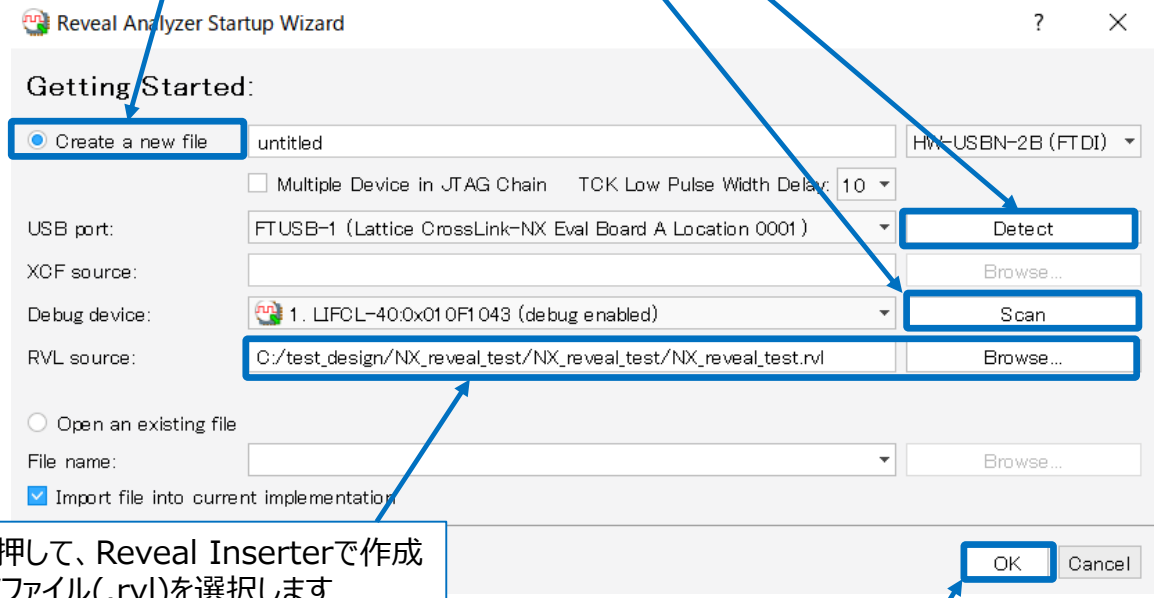
書き込みが完了したら、Tools > Reveal Analyzer/Controller もしくはToolbarから Reveal Analyzer/Controllerアイコン を クリックします



ポップアップ画面が出てきたら、Create a new file にチェックを入れます。以降、デバイスとPCは接続した状態で作業を行う必要があります

Detectボタンをクリックし、PCと接続されているケーブルを認証させます

Scanボタンを押して対象デバイスを認識させます

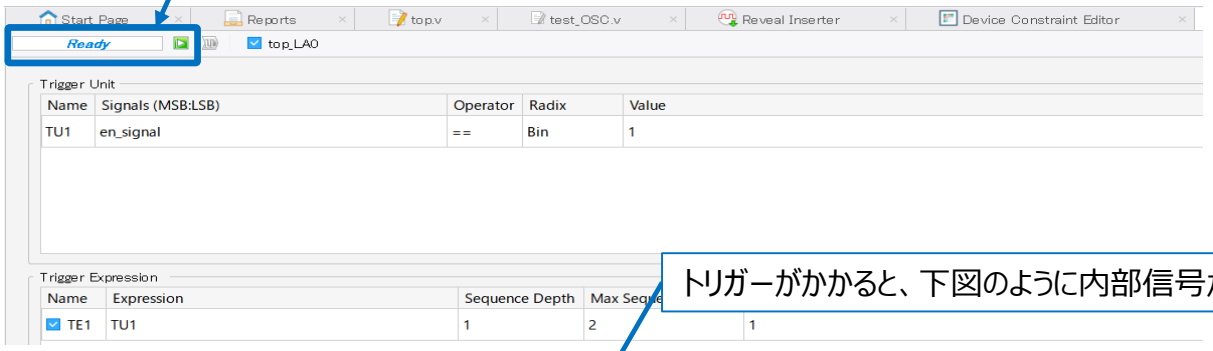


Browseボタンを押して、Reveal Inserterで作成した、Revealコアファイル(.rvl)を選択します

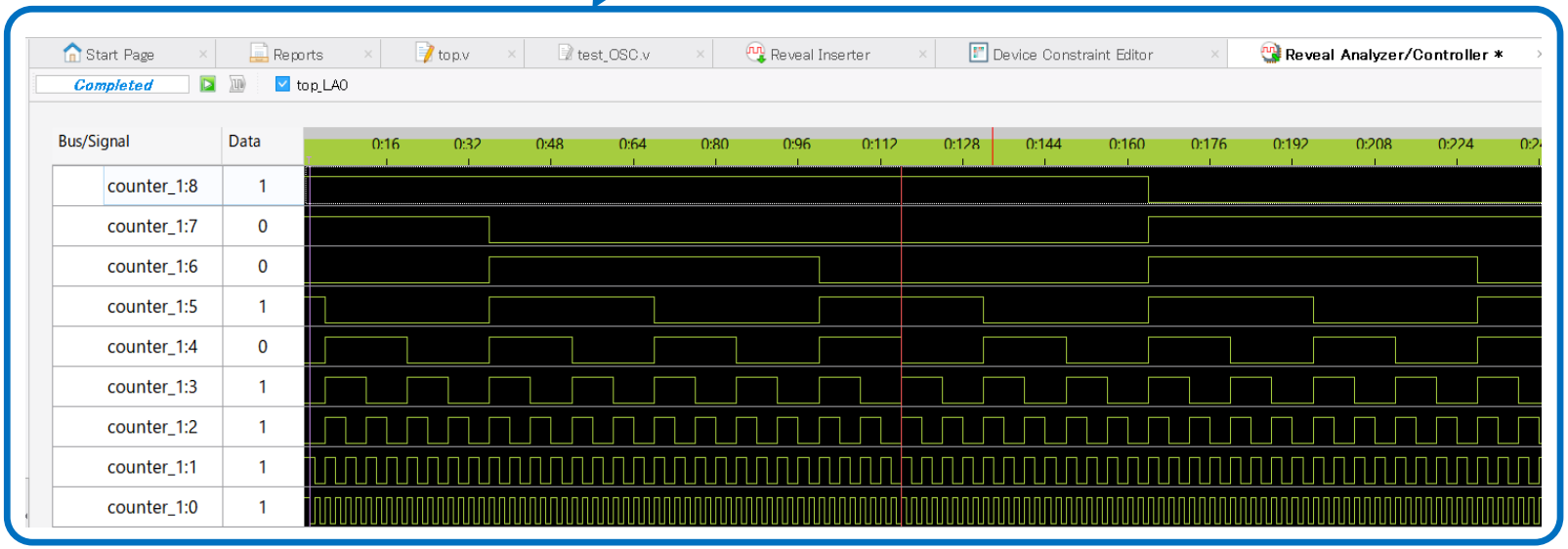
上記設定が完了したらOKボタンをクリックします

実機上での内部波形観測

Reveal Analyzerが起動したら、Ready表示に
なっていることを確認して、 アイコンをクリックします

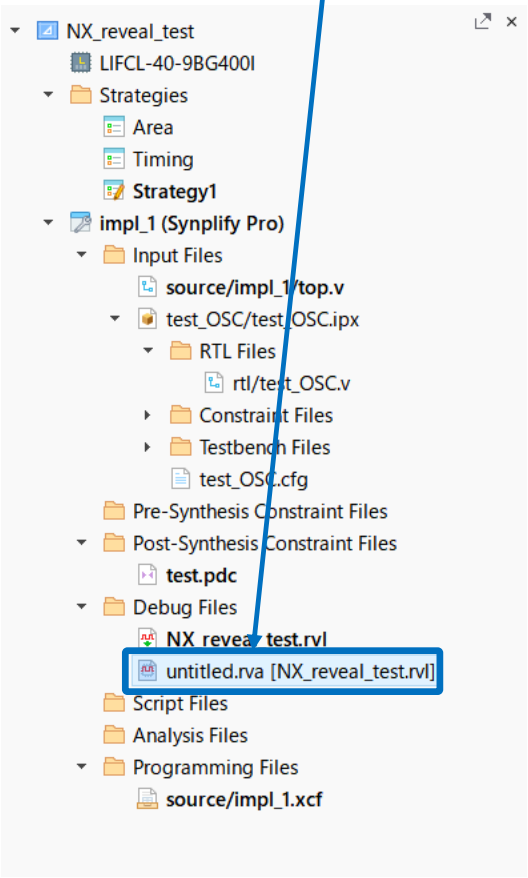


トリガーがかかると、下図のように内部信号が表示されます

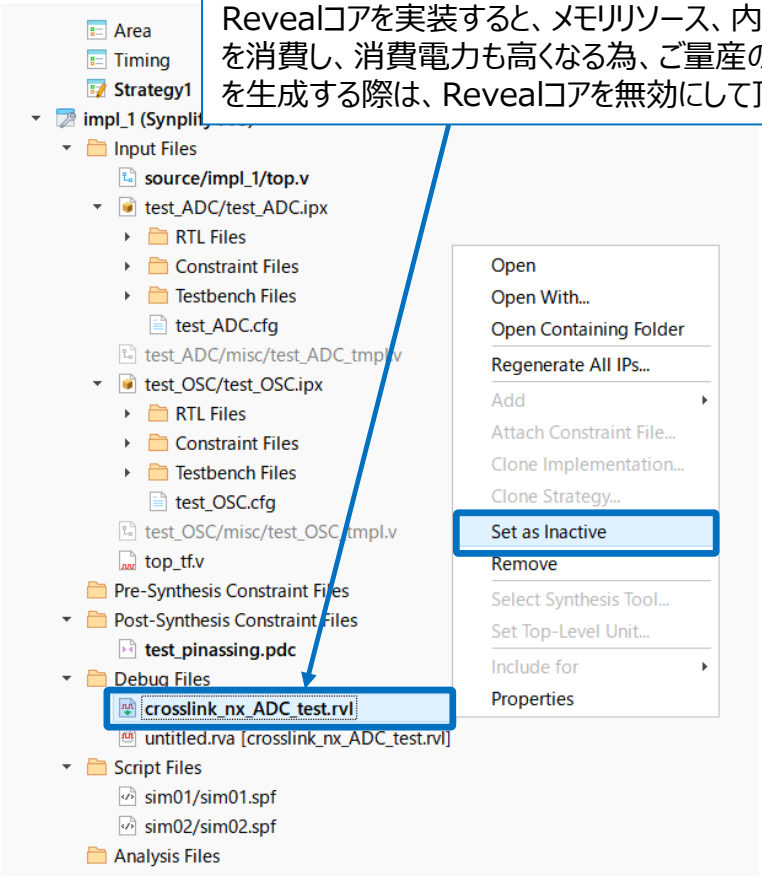


実機上での内部波形観測

初回移行、再度Reveal Analyzerを起動するときは、File ListのReveal Analyzer起動ファイル(.rva)をダブルクリックするだけで、起動させることが可能です



Revealコアは、File ListのRevealコアファイル(.rvl)を右クリック後、Set as Inactiveを選択することによって無効にすることが可能です。Revealコアを実装すると、メモリリソース、内部ロジック、配線、リソースを消費し、消費電力も高くなる為、ご量産のコンフィグレーションデータを生成する際は、Revealコアを無効にして頂く事をお勧め致します



Revision History

Date	Revision	Page	Change Information	Updated by
2020/04/24	1.0		First Revision	W. Nakatsuka
2020/04/30	1.1	P74～P94	デザイン書き込み方法、内部波形観測方法を追加	K. Ishigaki
2021/08/02	1.2	P19	誤記修正	H. Nogawa