

第 2 章 プロジェクト管理

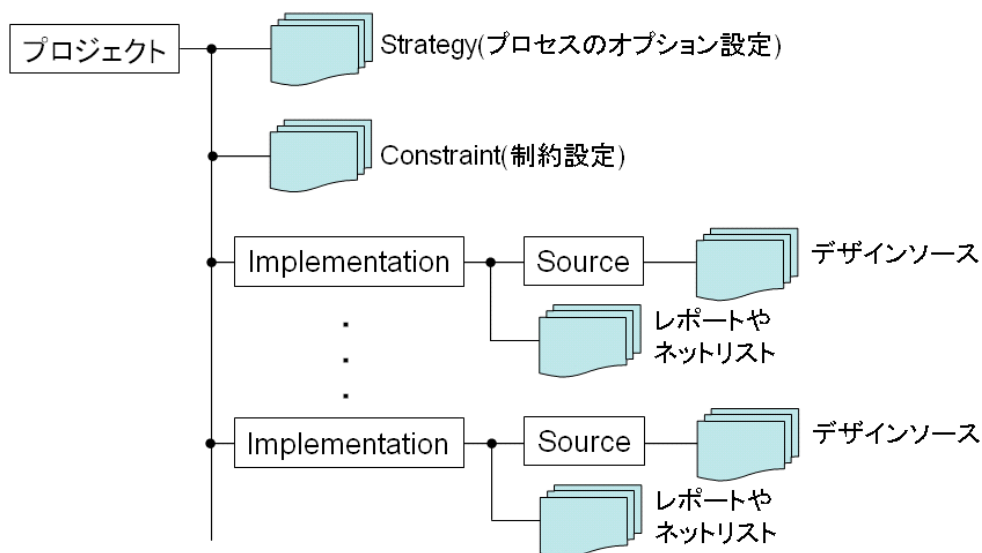
本章では Lattice Diamond のプロジェクト・ナビゲータ (Project Navigator) 上でのデザインプロジェクトの管理方法について説明します。Lattice Diamond またはプロジェクト・ナビゲータと記述されている場合はどちらも Lattice Diamond Project Navigator を表すものとします。

2.1 プロジェクトの構造

本節では Lattice Diamond のプロジェクトの構造について説明します。Lattice Diamond のプロジェクト・ナビゲータでは、「プロジェクト」単位でデザインを管理します。プロジェクトは主に「インプリメンテーション (Implementation)」、「ストラテジ (Strategy)」と「設計制約 (Constraint)」から構成されます (図 2-1)。

- ・ Lattice Diamond で用いる「設計制約」には、論理合成の際に使用する制約「Synthesis Constraint」と論理合成後のプロセスで使用する制約「LPF Constraint」があります。このドキュメントでは、特に指定の無い場合の「Constraint」は、両方を指すものとします。Constraint 自体は HDL アトリビュートを含む用語です
- ・ 設計制約として全く同じ意味で「Preference」も用います。制約ファイル「Preference File」としては「LPF (Logical Preference File)」や「PRF (Physical pReference File)」があります
- ・ 「Synthesis Constraint」については、論理合成のマニュアルで説明します

図 2-1. プロジェクト構造

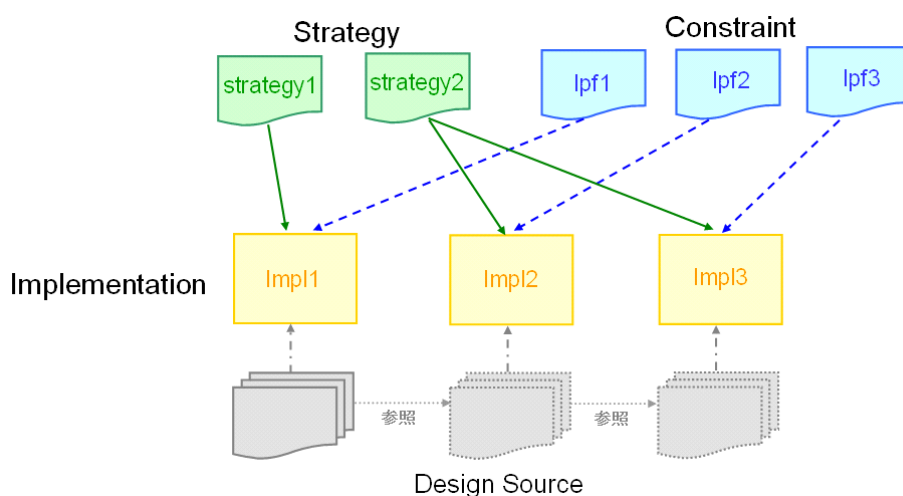


論理合成や配置配線といったプロセスはインプリメンテーション単位で行われます。この際に使用される各プロセスのオプション設定が「ストラテジ」、タイミング等の制約設定が「Constraint」です。ストラテジと Constraint はプロジェクト単位で管理されます。それぞれ複数作成することができ、プロジェクト内のインプリメンテーションでこれらを共有します。使用するストラテジと Constraint はインプリメンテーションごとに各 1 つを選択します。

デザインソース・ファイル (HDL や回路図等) は、インプリメンテーションごとに個別に管理されますが、インポートの際に他のインプリメンテーションと同じソースを参照する設定にすれば複数のインプリメンテーション間で共有することも可能です (図 2-2)。ソースファイルのインポート方法については、**2.6.2 項**を参照してください。

- ・ ランマネージャ (Run Manager) を使用することにより、同時に複数のインプリメンテーションを処理・実行することができます。ランマネージャについては**第 5 章**を参照してください

図 2-2. 複数インプリメンテーションでの各要素の共有例



2.2 プロジェクトの管理

2.2.1 新規プロジェクト作成

Lattice Diamond ではプロジェクト単位でデザインの管理を行います。本項では、プロジェクトの作成方法について説明します。

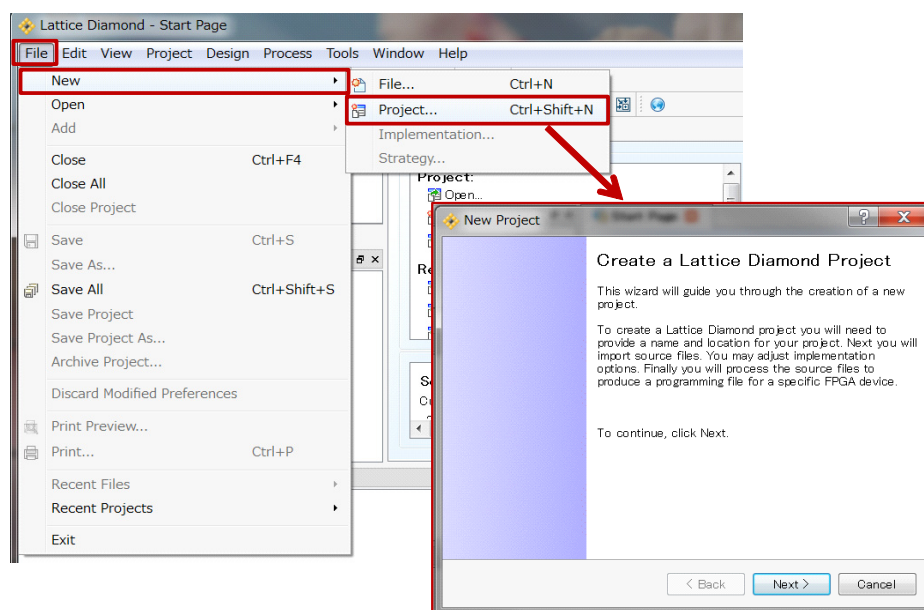
Lattice Diamond のプロジェクトの作成には、新たに Lattice Diamond プロジェクトを作成する方法と、ispLEVER のプロジェクトをインポートして自動的に Lattice Diamond のプロジェクトに変換する方法があります。新たにプロジェクトを作成する際は、メニューバーから [File]=[New]=[Project] の順に選択するか、[Start Page] タブ上で [Project :New] を選択します。これで Project Wizard が起動します (図 2-3) ので、その指示に従ってデバイス等を選択すればプロジェクトが生成されます。

起動した Project Wizard には、プロジェクトの生成に必要なフォルダやデバイスを指定する旨のメッセージが表示されていますので、そのまま [Next>] ボタンをクリックします。次に表示されるウィンドウでは、プロジェクト名とプロジェクトのフォルダパスを設定します (図 2-4)。

[Project:Name] 欄にプロジェクト名を入力します。プロジェクト名として使用できる文字は、アルファベット、数字と _ (アンダースコア) です (先頭の 1 文字目はアルファベットのみ) 使用できない文字はキーを

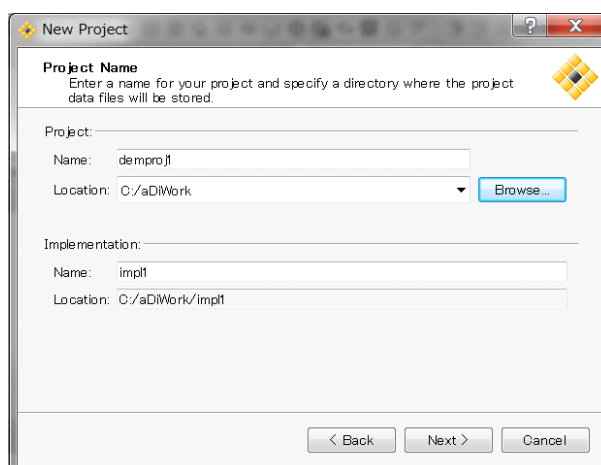
打っても入力されません。

図 2-3. プロジェクトの新規作成 ～ Project Wizard の起動



[Project:Location] 欄ではプロジェクトで使用するフォルダを指定します。直接パスを入力するか、[Browse...] ボタンをクリックして立ち上がるウインドウ上で適当なフォルダを選択します。フォルダパスを直接入力した場合、指定したフォルダが存在しなければ自動的に新しいフォルダが生成されます。

図 2-4. プロジェクトの新規作成 ～ プロジェクト名とフォルダパスの設定



[Implementation: Name] 欄にはインプリメンテーション名を入力します。デフォルトでは [Project:Name] 欄と同じ名前が自動的に入力されます。

*****-----

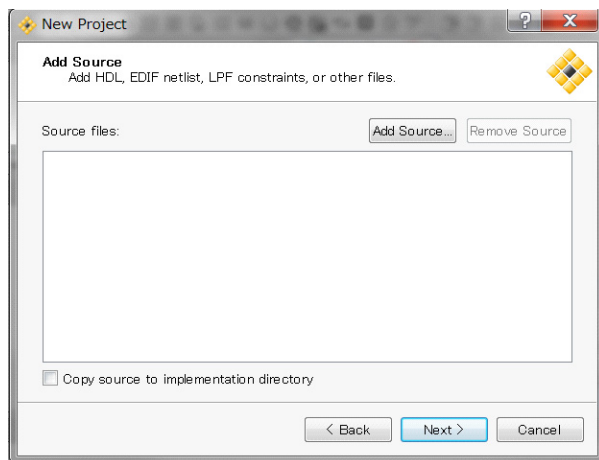
- ・ [Implementation: Location] 欄は、実装処理で使用するソースやネットリスト、各種レポートが格納されるフォルダで、「プロジェクトフォルダ/インプリメンテーション名」になります

*****-----

[Implementation: Name] 欄は Project 欄を入力すれば自動的に設定されますが、変更することもできます。[Implementation: Name] 欄を変更すると [Implementation: Location] 欄のパスが自動的に変更されます。

設定が完了後、[Next >] ボタンをクリックすると、次はソースのインポートウインドウが開きます(図 2-5)。

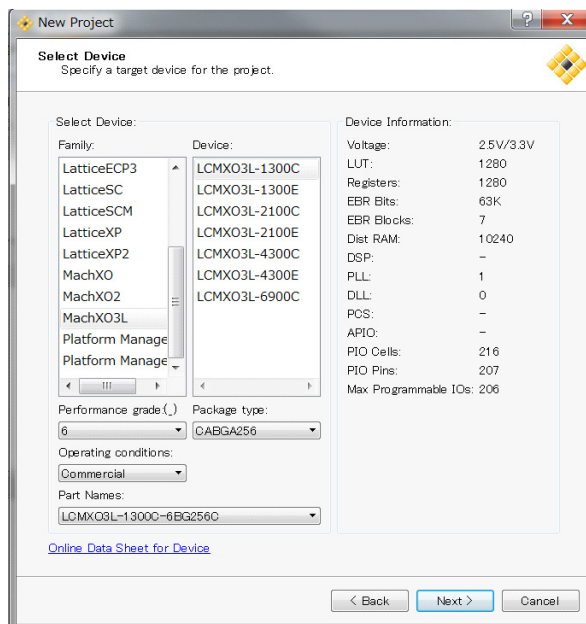
図 2-5. プロジェクトの新規作成 ～ ソースファイルのインポート



その後、[Add Source..] ボタンをクリックすると起動するウインドウ上で、必要なファイル（既存の HDL ソース、edif ファイルや制約ファイル）を選択します。この際、ウインドウ左下の [Copy source to ...] チェックボックスにチェックが入っていると、選択したファイルがプロジェクトのフォルダへコピーされ、それがインポートされます。チェックが入っていない場合は選択したファイルがインポートされます。

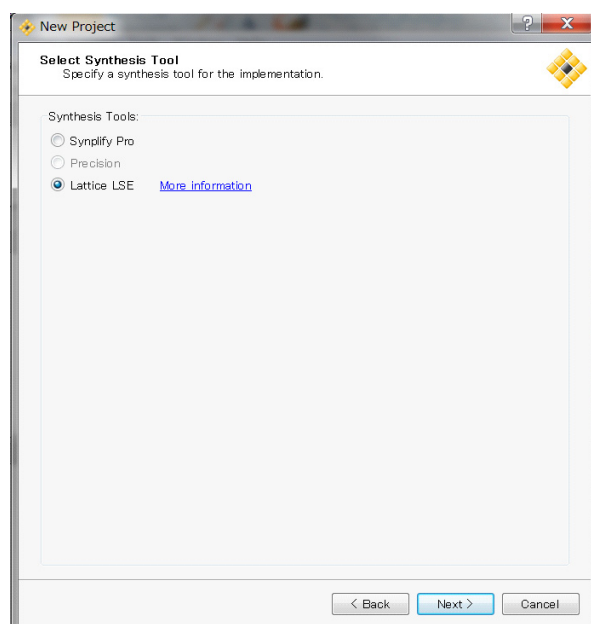
ソースのインポートはプロジェクト作成後でもできるので、ここで全てのソースをインポートする必要はありません。インポートするソースの選択完了後、[Next >] ボタンをクリックすると、次は使用するデバイス選択ウインドウが開きます（図 2-6）。

図 2-6. プロジェクトの新規作成 ～ デバイス選択



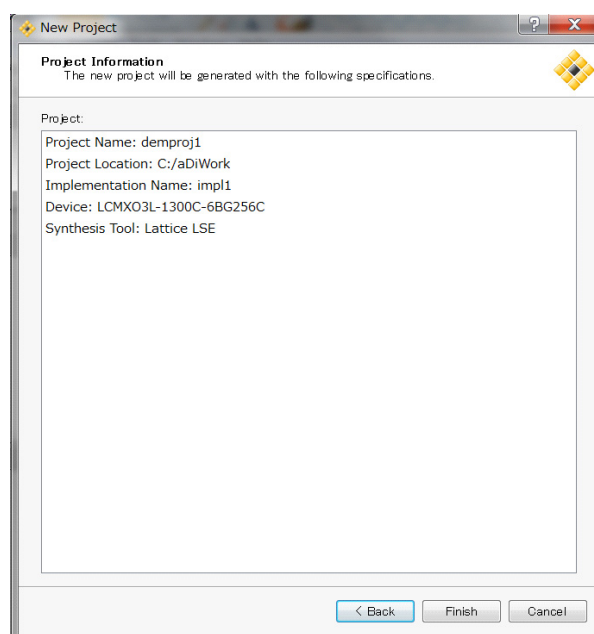
このウインドウ上で、使用するデバイスやスピードグレード、パッケージを選択します。ライセンスのタイプによって選択できるファミリーに違いがあります。有償（Subscription）ライセンスでは全てのファミリーが選択できますが、無償（Free）ライセンスでは、SERDES を内蔵しているデバイスは表示されないため選択できません。デバイスやパッケージはプロジェクト作成後でも変更できます。確定していない条件については適当なものを選択しておき、確定後に変更してください

図 2-7. 論理合成ツールの選択



デバイス選択完了後、[Next >] ボタンをクリックすると、論理合成ツールの選択ウインドウが開きます。MachXO シリーズと ECP5 のデフォルト論理合成ツールは LSE（Lattice Synthesis Engine）です。Synplify Pro を選択する事も可能です。また、プロジェクト生成後でも、メニューバーの Project => Synthesis Tool ... と辿ることで、変更することが可能です。

図 2-8. 新規プロジェクトの作成 ～ 設定確認

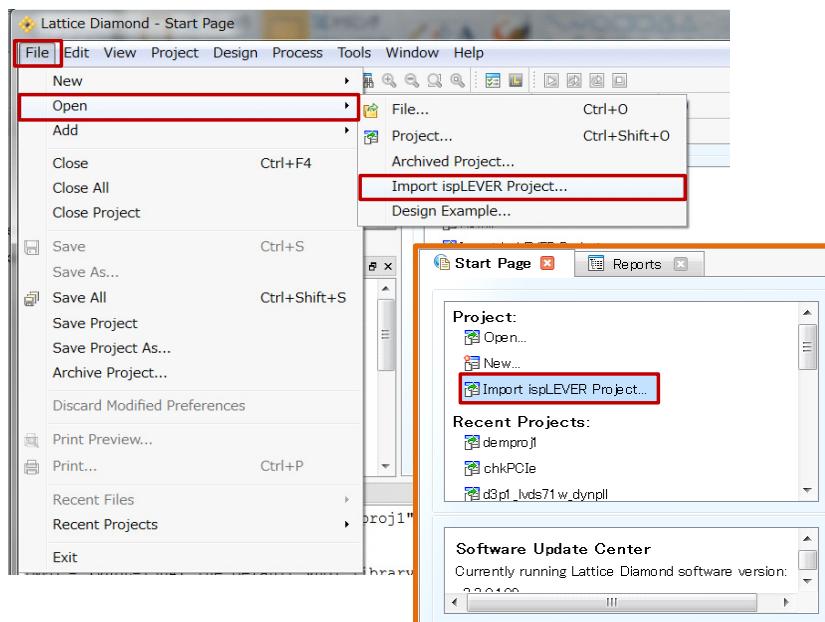


論理合成ツールの指定後 [Next >] ボタンをクリックすると、これまでの設定の確認ウインドウが開きます。[Finish] ボタンをクリックすれば、設定した内容に従ってプロジェクトが作成されます。設定内容を変更する場合は、[< Back] ボタンをクリックして適当なウインドウまで戻り設定の変更を行ってください。作成されたプロジェクト情報（インプリメンテーションやインポートしたファイル等）は、プロジェクトフォルダに「プロジェクト名 .ldf」というファイル名で保存されます。

2.2.2 ispLEVER プロジェクトのインポート

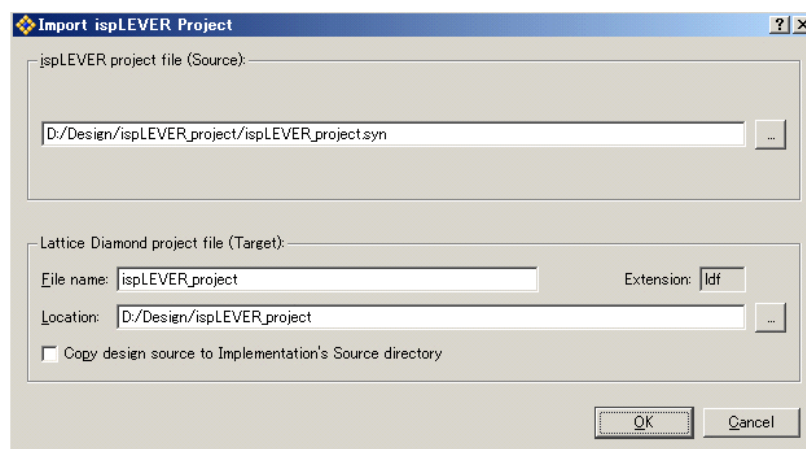
Lattice Diamond では、Lattice の旧デザインツールである ispLEVER のプロジェクトをインポートし、Lattice Diamond 上で処理を行うこともできます。インポートした場合、デザインファイルや設定ファイル、各プロセスのオプション設定等が引き継がれます。

図 2-9. ispLEVER プロジェクトのインポート ～ インポートするプロジェクトの選択



メニューバーから [File] => [Open] => [Import ispLEVER Project..] の順に選択するか、[Start Page] の [Import ispLEVER Project] をクリックすると（図 2-9）、インポートする ispLEVER のプロジェクトファイル（*.syn）を選択するウインドウが起動します。このウインドウ上で、インポートする ispLEVER のプロジェクトファイルを選択すると、Location やソースのインポート方法の確認ウインドウが開きます。

図 2-10. ispLEVER プロジェクトのインポート ～ ロケーション設定



ウインドウの下側が Lattice Diamond のプロジェクトファイル名（＝プロジェクト名）とロケーションの設定です。デフォルトではプロジェクトファイル名とロケーションは ispLEVER のプロジェクトと同じものが設定されています。このまま [OK] ボタンをクリックすると、元のプロジェクトと同じフォルダ内に Lattice Diamond のプロジェクトが作成されます。プロジェクト名とロケーションは必要に応じて変更してください。

*****-----

- ・インプリメンテーション名は自動的に設定されます（変更できません）。プロジェクト名を変更しなかった場合は [impl1]、プロジェクト名を変更した場合は「ispLEVER のプロジェクト名」がインプリメンテーション名になります

*****-----

ウインドウ左下の [Copy design source …] チェックボックスは、ソースファイルのインポート方法に関する設定です。チェックが入っていれば、選択したファイルがインプリメンテーション毎に生成されるフォルダ下にコピーされ、それらがインポートされます。チェックが入っていない場合は選択した元々のファイルがインポート（参照）されます。

ispLEVER プロジェクトをインポートする際の重要注意事項

ispLEVER プロジェクトを Lattice Diamond にインポートする際に、ソースファイルを Lattice Diamond にコピーする設定を行っても、コピーされるのはデザインソースとして ispLEVER にインポートされているファイルのみです。このため以下の場合に必要なファイルがコピーされずに処理中にエラーや Warning が発生することがあります。ソースファイルのタイプごとの対処方法は以下のとおりです。

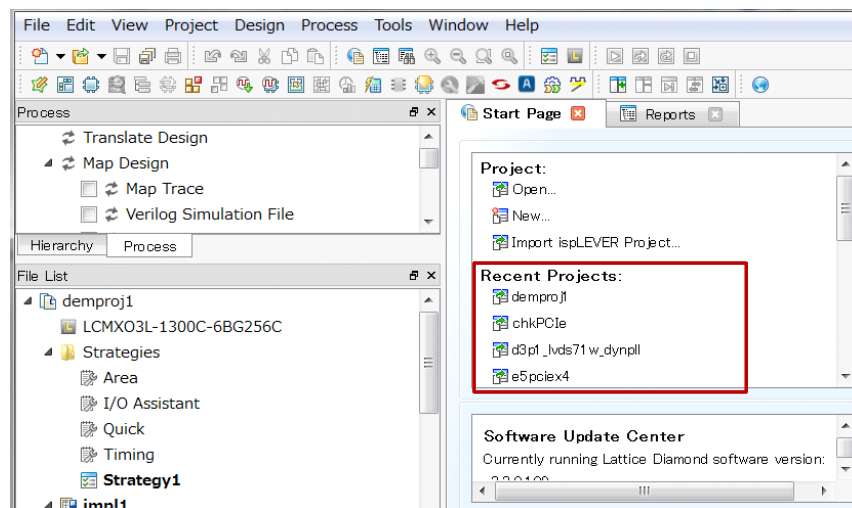
表 2-1. 使用ファイルタイプごとの問題と対処方法まとめ

ソースファイル	発生する問題	回避策
プロジェクトに IPexpress で生成した lpc ファイルをインポートしている場合	ツールは論理合成時に lpc ファイルと同じフォルダにあるはずの HDL ソースを使用しますが、インポートの際に HDL ソースはコピーされないためファイルが見つからずエラーになる事がある	プロジェクトフォルダ内にある HDL ソースをインプリメンテーションの source フォルダにコピーするか、Lattice Diamond の IPexpress で lpc ファイルからソースを再生成する
プロジェクトで SERDES/PCS を使用している場合	SERDES/PCS の設定が記述されたテキストファイル (.txt) がコピーされないため、ファイルが見つからない旨のメッセージが出力される。その後各プロセスでデフォルト設定値で処理されるため、期待動作しない	ispLEVER プロジェクトのフォルダ内にあるテキストファイルをインプリメンテーションのフォルダにコピーするか、Lattice Diamond の IPexpress で lpc ファイルからソースを再生成する
プロジェクトで ngo ファイル形式のネットリスト (IP コア) を使用している場合	ngo ファイル形式のネットリストは、Translate Design プロセスで論理合成結果とマージされるが、インポートの際にこのファイルはコピーされないためファイルが見つからずエラーになることがある	プロジェクトで使用していた ngo ファイルをインプリメンテーションのフォルダにコピーするか、Lattice Diamond のストラテジで、Translate Design の [Macro Search Path] オプションに ngo ファイルのあるパスを記入する
プロジェクトの PAR または Map プロセスで NCD Guide File オプションとして ncd ファイル形式のネットリストを参照する設定を行っている場合	参照されるネットリストがインポートの際にはコピーされないため、ファイルが見つからずエラーになることがある	プロジェクトで使用していた ncd ファイルをインプリメンテーションのフォルダにコピーするか、Lattice Diamond のストラテジで PAR または Map プロセスの [NCD guide file] オプションにオリジナルの ncd ファイル名をフルパスで指定する
プロジェクトで、回路図内に IPexpress または ispLEVER で HDL ソースから生成したシンボルを使用している場合	スキマティック・シンボルがコピーされないため、Schematic Editor を開いた際に該当するシンボルが見つからない旨のメッセージが表示される	プロジェクトで使用していたシンボル・ファイル (*.sym) を、インプリメンテーションの source フォルダにコピーする
プロジェクトで、プロジェクトにインポートされていないファイルを verilog HDL ソース内でインクルードしている場合	インクルードされているだけのファイルはコピーされないため、論理合成の際に該当するファイルが見つからずエラーになることがある	プロジェクトで使用している Verilog HDL ソース内でインクルードしていたファイルを、インプリメンテーションの source フォルダにコピーする

2.2.3 既存プロジェクトのオープン

既存の Lattice Diamond プロジェクトを開くには、メニューバーから [File] => [open] => [Project...] の順に選択するか、Start Page の Project:Open をクリックします。また、Start Page には、[Recent Project] として最近使用したプロジェクト名が幾つか表示されています。開きたいプロジェクトがこの中にあれば、そのプロジェクト名をクリックします。

図 2-11. 既存プロジェクトのオープン



[Recent Project] として表示されるプロジェクト数は、プロジェクト・ナビゲータのオプション設定で変更することができます。

2.2.4 プロジェクトのクローズ

プロジェクトをクローズする場合は、メニューバーから [File] => [Close Project] の順に選択します。他のプロジェクトをオープンするか、Lattice Diamond をシャットダウンすると自動的にクローズされます。メニューの close や close all からではプロジェクトはクローズされません。

2.2.5 クローン・プロジェクトの生成

プロジェクトを複製する機能が「クローン生成」です (Diamond 3.1 以降)。File メニューから [Save Project As...] を選択して表示されるウィンドウで、保存フォルダと別名で名称を指定するだけです (図 2-12)。

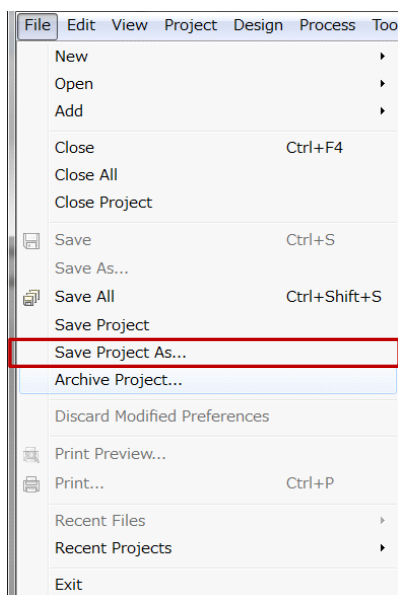
2.2.6 プロジェクトのアーカイブ作成

プロジェクトを保存する場合や何らかの都合でフォルダを移動させる場合等に、プロジェクトのアーカイブを作成することができます。メニューバーから [File] => [Archive Project] を選択すると、[Archive project] ウィンドウが開くのでこのウィンドウ上で、アーカイブの保存先を選択します。

この際、ウィンドウ左下にある [Archive all files ...] にチェックが入っている場合は、プロジェクトとの関連に関わらずプロジェクトフォルダ以下にあるフォルダとファイル全てが含まれるアーカイブが作成されます。チェックが入っていない場合は、プロジェクトに関連しているフォルダとファイルのみのアーカイブが作成されます。

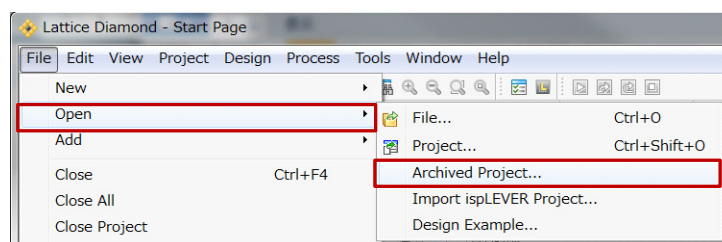
ラティスのテクニカルサポートにプロジェクトを送付して技術支援を受ける場合などに、本アーカイブ機能を用いると容易にやりとりができます。

図 2-12. クローン・プロジェクトの生成



なお、アーカイブ後の .zip されたファイルを Windows 対応の解凍ツールを用いて解凍すると、一部のファイルが復元できなくなる場合も報告されています。アーカイブファイルの復元は [File] => [Open] => [Archived Project...] の手順で行うようにします。

図 2-13. アーカイブ・プロジェクトの復元



2.3 インプリメンテーションの管理

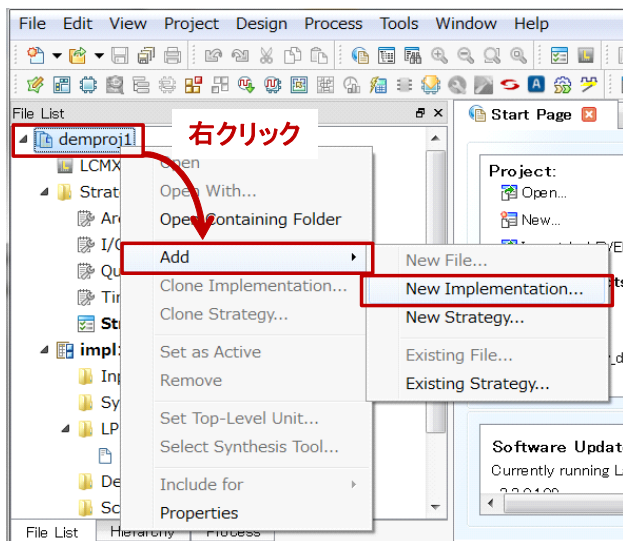
「インプリメンテーション」とは、プロジェクト内で設定・定義する、ソースファイルやモジュール、或いはオプション設定や各設計制約の組み合わせの呼称です。Diamond では、単一プロジェクト内に複数のインプリメンテーションを定義することができます。それぞれのインプリメンテーションでは異なるソースファイルや制約ファイル、あるいはオプションを用いて論理合成や配置配線を行うことができます。従来複数プロジェクトを生成して行っていた、実装に係わるトライアル・アンド・エラーを、Diamond では単一プロジェクト内で行う事が可能です。このため最適な実装の「推敲」が容易になります。

本節ではインプリメンテーションの追加方法や切り替え方法について説明します。インプリメンテーション内でのストラテジや制約の指定・選択方法等については、**3.5.4 項**または **3.6.3 項**を参照してください。

2.3.1 新規インプリメンテーションの作成

インプリメンテーションはプロジェクトを作成した際に必ず 1 つ作成されます。2 つ以上のインプリメンテーションが必要な場合、メニューバーから [File] => [New] => [Implementation] の順に選択するか、[File list] ウィンドウの一番上に表示されているプロジェクト名を右クリックし [Add] => [new Implementation] の順に選択します (図 2-14)。

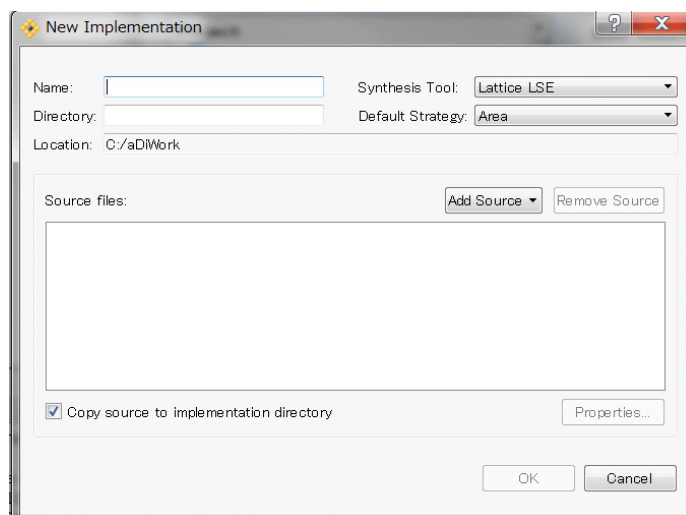
図 2-14. インプリメンテーションの追加



これで New Implementation Wizard が立ちあがります (図 2-15)。**[Name]** 欄は追加するインプリメンテーションの名前、**[Directory]** 欄はインプリメンテーションで使用するファイルを格納するフォルダ名を入力します。名前を先に入力するとフォルダ名も自動的に同じものが入力されますが、変更することもできます。

[Default Strategy] 欄はインプリメンテーションで使用するストラテジを選択します。右側の▼アイコンをクリックすると、プロジェクトに登録されているストラテジの一覧が表示されるので、適当なものを選択してください (インプリメンテーション作成後に変更することも可能です)。

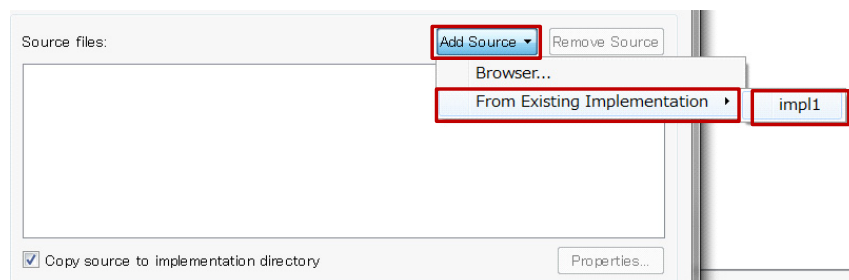
図 2-15. New Implementation Wizard



インプリメンテーション作成時にソースファイルのインポートを行うこともできます。**[Add Source]** ボタンをクリックするとプルダウンメニューで **[Browser…]** と **[From Existing …]** が表示されます。ブラウザを使用して既存のファイルを選択する場合は **[Browser…]** を選択します。既存のインプリメンテーションにインポートされているソースを選択する場合は、**[From Existing..]** を選択すると既存のインプリメンテーション名が全てリストされますので、その中から 1 つを選択します (図 2-16。この図では一つしかない)。

この際、ウインドウ左下の [Copy source to ...] にチェックが入っていると、選択したソースは新規に作成したインプリメンテーションのフォルダにコピーされ、そちらがインポートされます。チェックが入っていないと選択したファイルがインポートされます。

図 2-16. 既存インプリメンテーションからのソースのインポート



*****-----

- ・ コピーした場合、ソースを変更しても他のインプリメンテーションのソースには反映されません
- ・ コピーしなかった場合、複数のインプリメンテーション間で同一のソースを使用できます

*****-----

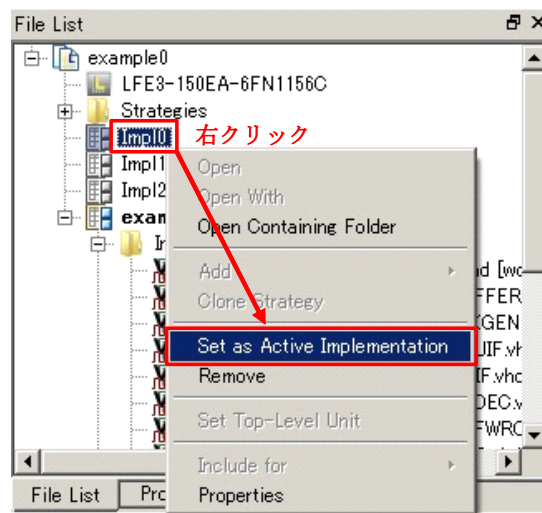
必要なソースを選択後、OK ボタンをクリックすると新しいインプリメンテーションが追加されます。

2.3.2 アクティブなインプリメンテーションの選択

プロジェクト・ナビゲータ上では、基本的に 1 つのインプリメンテーションで操作（オプション設定やファイルのインポート等）ができます。このため複数のインプリメンテーションを持つプロジェクトでは、アクティブにして操作するインプリメンテーションの選択を行う必要があります。

File Lists ウィンドウで、アクティブにしたいインプリメンテーション名を右クリックし、メニューから [Set as Active Implementation] を選択するとインプリメンテーションがアクティブになります。

図 2-17. アクティブにするインプリメンテーションの選択



2.3.3 インプリメンテーションの削除

不要なインプリメンテーションをプロジェクトから削除するには、まずそのインプリメンテーションを非アクティブ化（他のインプリメンテーションをアクティブ化）します。次に削除したいインプリメンテーショ

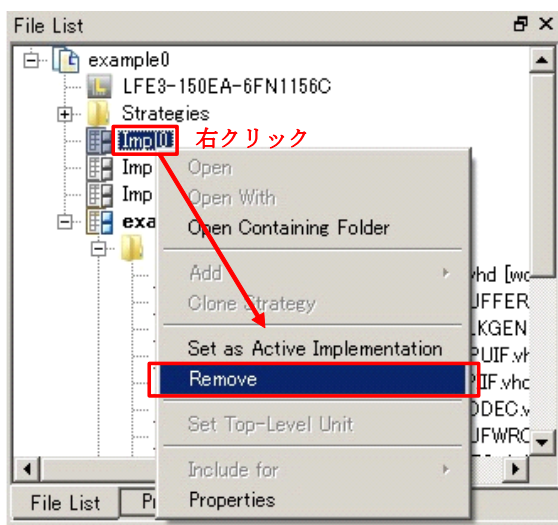
ン名を右クリックし、[Remove] を選択するか、選択した状態でキーボードの [Delete] キーを押すとインプリメンテーションがプロジェクトから削除されます。

*****-----

- ・ アクティブなインプリメンテーションは削除できません
- ・ 「削除」は「インポート情報の削除」であり、ファイルやフォルダの削除ではありません

*****-----

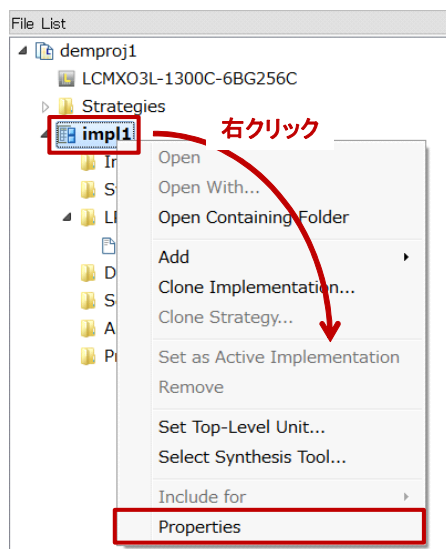
図 2-18. インプリメンテーションの削除



2.3.4 インプリメンテーションのプロパティ

プロパティ設定は図 2-19 のように行います。

図 2-19. インプリメンテーションのプロパティ設定



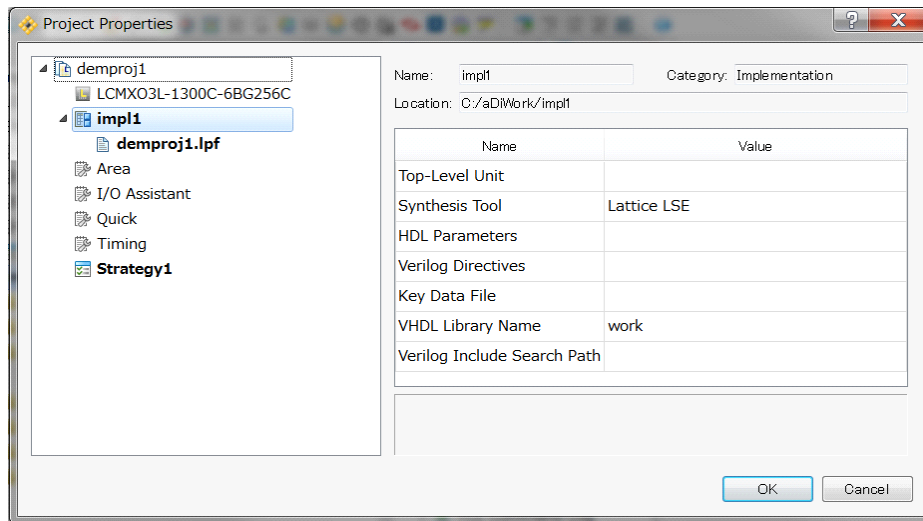
インプリメンテーションはプロパティとして、以下の設定を行うことができます。

- ・ 最上位モジュール名
- ・ HDL のパラメータ

- ・ VHDL ファイルのコンパイルライブラリ名
- ・ Verilog HDL ソースでインクルードされているファイルがあるフォルダ指定

これらの設定を行う場合は、メニューバーから [Project] => [Property Pages] の順に選択するか、File List ウィンドウでインプリメンテーション名を右クリックして [Properties] を選択します (図 2-19)。これで、Project Properties ウィンドウが立ち上がります。

図 2-20. Project Properties ウィンドウ ~ インプリメンテーション



Project Properties ウィンドウでは、左側にプロジェクトに登録されているインプリメンテーションやインポートされているソースが表示されています。ここで現在アクティブなインプリメンテーションを選択すると、ウィンドウ右側に設定項目が表示されます。

非アクティブなインプリメンテーションを選択した場合は、何も表示されません。インプリメンテーションのプロパティとして設定できるのは以下の 4 つの項目です。

Top Level Unit

インポートしたデザインソースの最上位階層の entity/module 名を入力します。論理合成ツールとして Synplify Pro が選択されていれば、ほとんどの場合で最上位階層の entity/module 名はインポートされたソースから自動的に検出されます。しかし VHDL/Verilog HDL 混在の場合には、最上位階層が自動検出されない場合があるので、その際にはこの欄に最上位階層の entity/module 名を入力してください。

論理合成ツールとして LSE を選択している場合は、設定が必要です。

HDL Parameter

HDL ソースにグローバルで作用するパラメータ (VHDL の generic、Verilog HDL の parameter やコンパイラ指示子) を指定することができます。これ以外のパラメータは HDL ソース内の値が適用されます。設定は以下の様に記述します。複数定義する場合はセミコロンで分離します。

ー 記述ルール

パラメータ = 設定値 [; パラメータ = 設定値]

ー 記述例

g_bus_width=16; comm_id=8;

VHDL Library Name

インプリメンテーションで、デフォルトで使用する VHDL ソースのコンパイル先ライブラリ名の設定です。デフォルトでは VHDL ソースは work ライブラリにコンパイルされます。VHDL ソース単位でコンパイル先のライブラリ名が異なる場合は、デザインソースのプロパティで設定してください (2.6.6

項参照)

Verilog Include Search Path

Verilog HDL ソース内でインクルードしているファイルを検索するパスの設定です。インクルードされるファイルが、ソース内にフルパスでファイル名が記述されていたり、プロジェクト内の適当なフォルダに保存されていたりする場合は、設定する必要はありません。

2.3.5 インプリメンテーションの複製（クローン）

インプリメンテーションの複製を生成する機能が「クローン生成」です。意図するインプリメンテーションを選択後、右クリックで [Clone Implementation...] を選びます (図 2-21)。その後表示されるウィンドウ (図 2-22) で名称を入力します。ソースファイルを参照するか、コピーするか指定も行います。

図 2-21. インプリメンテーションの複製

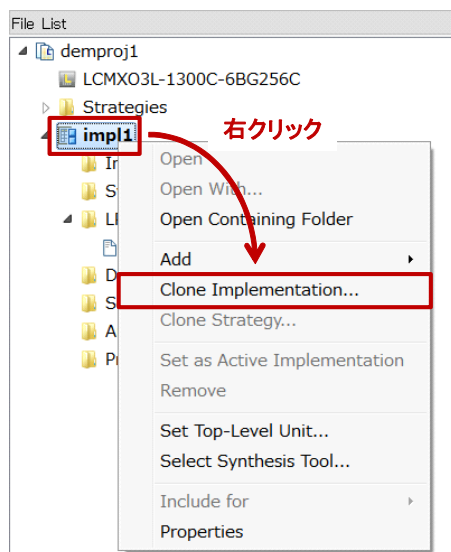
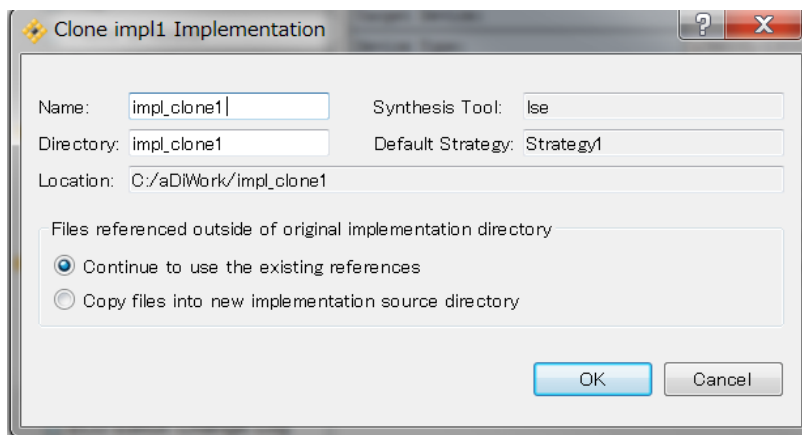


図 2-22. クローン・インプリメンテーションの設定



2.4 ストラテジの管理

ストラテジとは各プロセスに対して与える設計制約・オプション全てを集約したものの呼称です。Diamond では、単一プロジェクト内に複数のストラテジを持つことができ、それぞれのストラテジでは異なるオプション設定を定義することができます。本項ではストラテジの追加方法や切り替え方法について説明します。

2.4.1 デフォルトで作成されるストラテジ

プロジェクトを作成した際に、必ず以下の 5 つのストラテジが作成されます。

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

*****-----

- ・ Strategy1 以外は設定を変更することができません
- ・ I/O Assistant は IO のルールチェックだけのプロセスで、配線処理は行われません

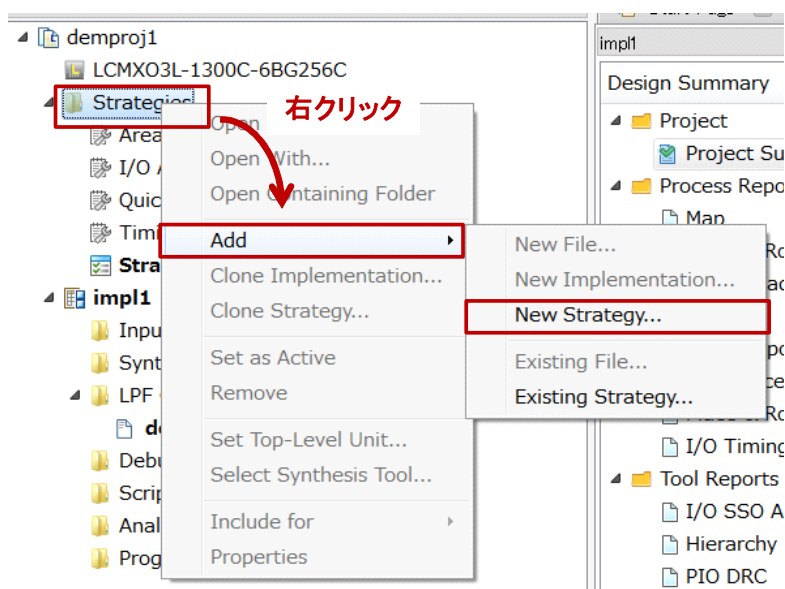
*****-----

2.4.2 新規ストラテジの作成

ストラテジを追加するには、新たに作成する方法と、既存のストラテジの clone (コピー) を作成する方法があります。本項では、新たに作成する方法について説明します。

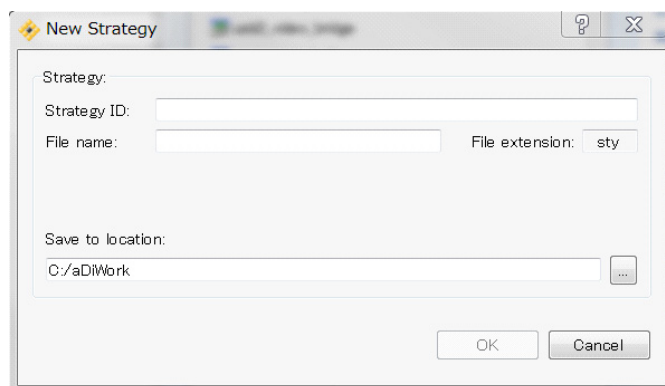
ストラテジを作成する場合は、メニューバーから [File] => [New] => [Strategy...] の順に選択するか、File list ウィンドウのストラテジシンボルを右クリックし、[Add] => [New Strategy...] の順に選択します (図 2-23)。これで、New Strategy Wizard が立ち上がります (図 2-24)。

図 2-23. ストラテジの追加



New Strategy Wizard の Strategy ID: 欄にはストラテジ名を入力します。Lattice Diamond 上にはこの名前が表示されます。File name: 欄にはデフォルトでストラテジ ID と同じ名前が設定されますが、変更することもできます。Save to Location: 欄はファイルを保存するフォルダです。デフォルトではプロジェクトフォルダが設定されていますが、変更することも可能です。

図 2-24. New Strategy Wizard

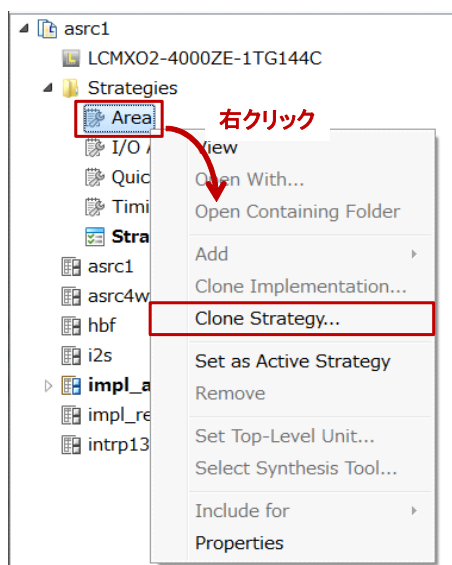


全ての設定完了後、OK ボタンをクリックすると、新規ストラテジ（設定は全てデフォルト）が作成されます。作成したストラテジは「strategy 名 .sty」というファイル名で指定したフォルダに保存されます。

2.4.3 ストラテジの複製

Lattice Diamond では既存のストラテジの複製を作成する機能が「クローン (Clone) 生成」です。複製したいストラテジを右クリックし、メニューから [Clone Strategy...] を選択します (図 2-25)。これで、図 2-24 と同様の『Clone *** Wizard』(“***” は複製元のストラテジ名) が立ち上がりますので、新ストラテジ名を入力します。

図 2-25. Clone Strategy の作成



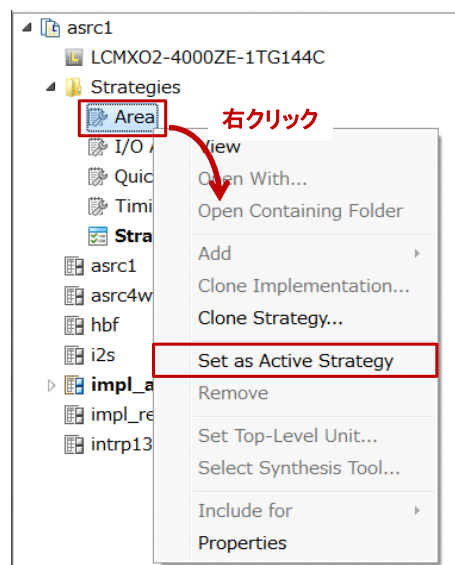
New Strategy Wizard の設定は、新規にストラテジを追加する場合と同じですが、作成されるストラテジの設定値は、デフォルトではなく複製元と同じになっています。プロジェクト作成時にデフォルトで生成される設定変更できないストラテジでも、Clone は設定を変更できます。

2.4.4 使用するストラテジの選択

インプリメンテーションでは、プロジェクトに複数あるストラテジの内、アクティブ化されている 1 つが使用されます。このため「ストラテジのアクティブ化」が「使用するストラテジの選択」になります。

ストラテジをアクティブ化するには、File List ウィンドウでストラテジを右クリックし、メニューから [Set as Active Strategy] を選択します。これでそのストラテジがアクティブ化され、以後そのインプリメンテーションで使用されます (図 2-26)。アクティブ化されたストラテジは太字で表示されます。

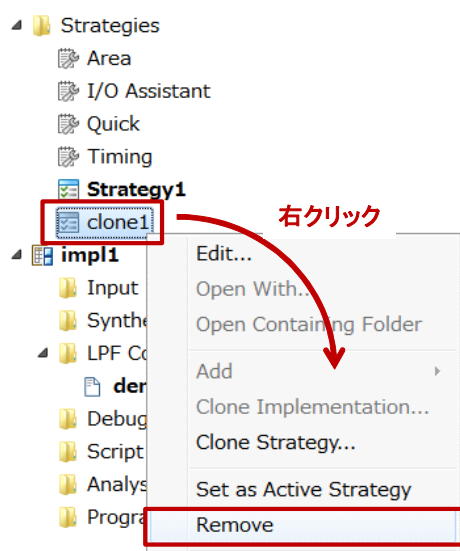
図 2-26. ストラテジのアクティブ化



2.4.5 ストラテジの削除

プロジェクトで不要になったストラテジは削除することもできます。File List ウィンドウで削除したいストラテジを右クリックし [remove] を選択するか、キーボードの [Delete] キーを押すと、ストラテジがプロジェクトから削除されます (図 2-27)。

図 2-27. ストラテジの削除

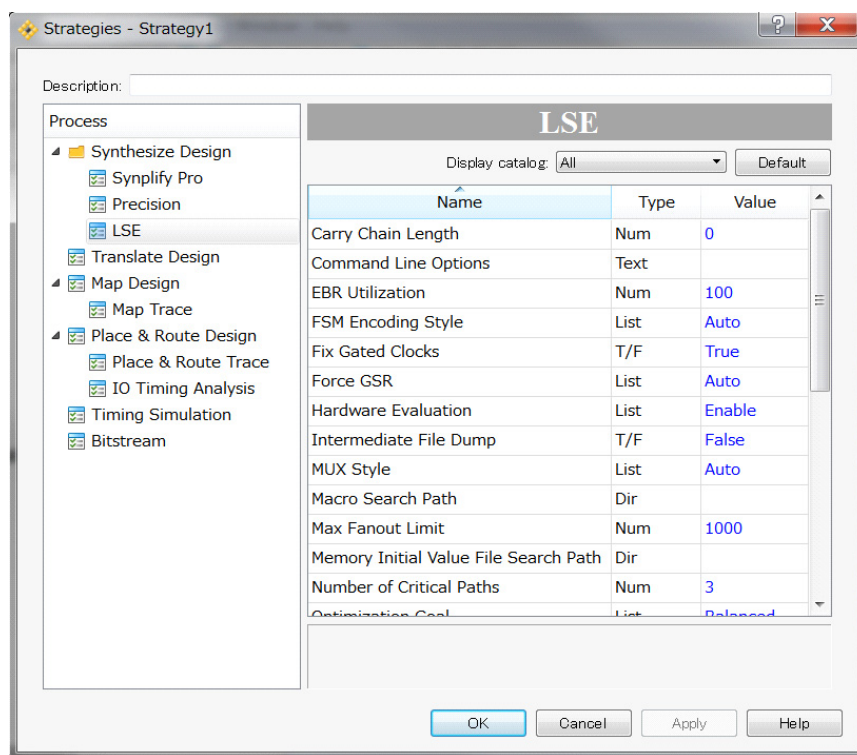


なお、非アクティブなインプリメンテーションでもアクティブなストラテジは削除できません。また、プロジェクト作成時に自動生成されるものは、[Strategy1] を含めて、全て削除できません。「プロジェクトから削除」することは「インポート情報の削除」であり、ファイルの削除ではありません。

2.4.6 ストラテジの設定変更

File List ウィンドウでストラテジ名をダブルクリックするか、右クリックしてメニューから [Edit] を選択すると、ストラテジの設定ウィンドウが立ち上がります (図 2-28)。

図 2-28. ストラテジの設定ウィンドウ (LSE の例、一部)



このウィンドウで各プロセスのオプション設定を変更することができます。なお、オプション設定の詳細については、各プロセスのマニュアルを参照してください。

2.5 LPF 制約ファイルの管理

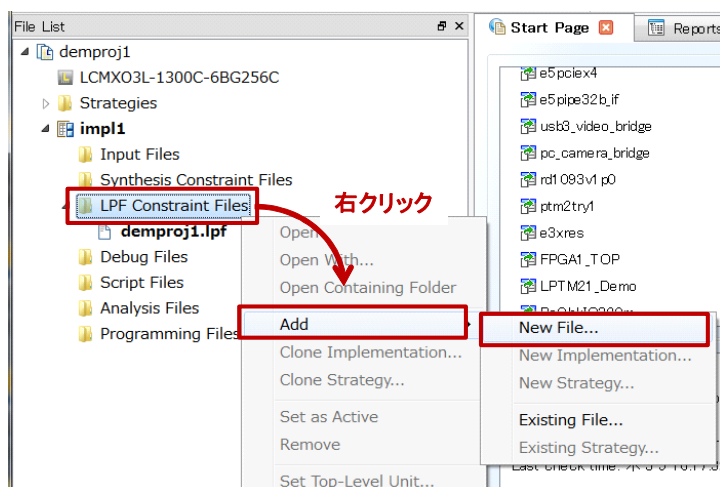
LPF (Logical Preference File) 制約ファイル ("LPF Constraint") は設計制約を記述するファイルで、拡張子が lpf です。プロジェクトとして配置配線に係わる全ての制約を書き出した制約ファイル .prf (Physical Preference File) と区別されます。テキストファイルですので、ユーザが任意に編集できます。

1つのプロジェクトは1つ以上の LPF 制約ファイルを持ち、それぞれの制約ファイルでは異なる制約の設定を行うことができます。本項では LPF 制約ファイルの追加方法や切り替え方法について説明します。

2.5.1 新規 LPF 制約ファイルの作成

プロジェクト作成時には、自動的に LPF 制約ファイルが1つ生成されます。内容は基本的な BLOCK 制約が二つ記載してあるのみです。名前は「プロジェクト名 .lpf」になります。これ以外に制約ファイルを追加する場合は、メニューバーから [File] => [New] => [File...] の順に選択するか、File List ウィンドウでフォルダシンボル (ストラテジ以外) を右クリックし、[Add] => [New File...] の順に選択します。これで、New File Wizard が起動します。

図 2-29. LPF 制約ファイルの追加



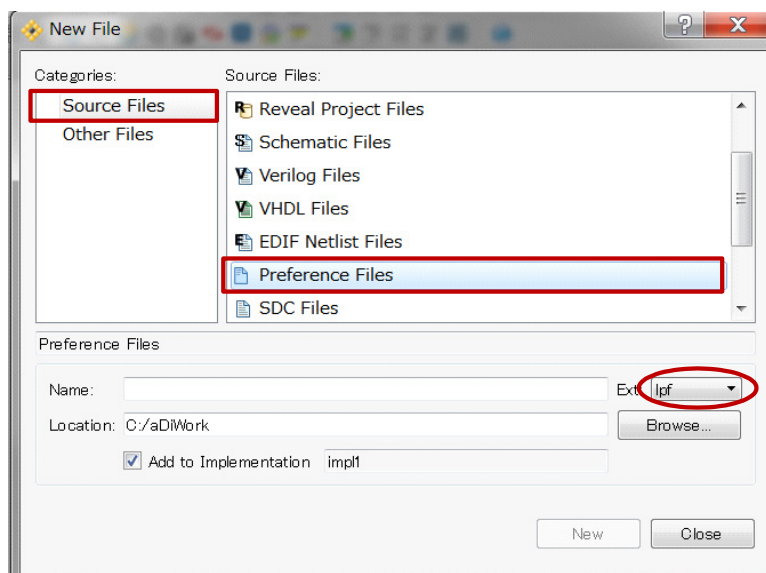
LPF 制約ファイルを追加する場合は、Categories 欄で [Source Files]、Source Files 欄で [Preference Files] を選択し、Name 欄にファイル名を入力します。Ext. 欄 (丸印) で拡張子を選択することもできますが、基本的には [lpf] を選択してください。設定完了後に New ボタンをクリックすると、新しい LPF 制約ファイルが作成されテキストエディタが立ち上がり、編集後に自動的にインプリメンテーションに取りこまれます。

*****-----

- ・ 新たに作成した LPF 制約ファイルは、完全なブランクファイルです。プロジェクト作成時に自動で生成される LPF 制約ファイルに記述されているデフォルトの BLOCK 設定も記述されていません
- ・ 「LPF 制約ファイルのコピー」機能はないので、必要な部分は他の LPF 制約ファイルを開き、マニュアルで記述のコピーを行ってください
- ・ LPF 制約ファイルを追加する場合、新規に作成するより既存のものを別名で保存し、それをインポートした方が、手間がかからない場合があります

*****-----

図 2-30. New File Wizard (LPF 制約ファイルの追加)



2.5.2 既存 LPF 制約ファイルのインポート

既存の LPF 制約ファイルをインポートする場合は、メニューバーから [File] => [Add] => [Existing File...] の順に選択するか、File List ウィンドウ上で * Files のフォルダシンボルを右クリックし、[Add] => [Existing File...] の順に選択します (図 2-31)。

これで、インポートするファイルの選択ウィンドウ (図 2-32) が立ち上がるので、まず File of Type として [LPF Constraint Files (*.lpf)] を選択します。これでリストに LPF 制約ファイルが表示されるので、適当なファイルを選択し [Add] ボタンをクリックします。

図 2-31. 既存 LPF 制約ファイルのインポート

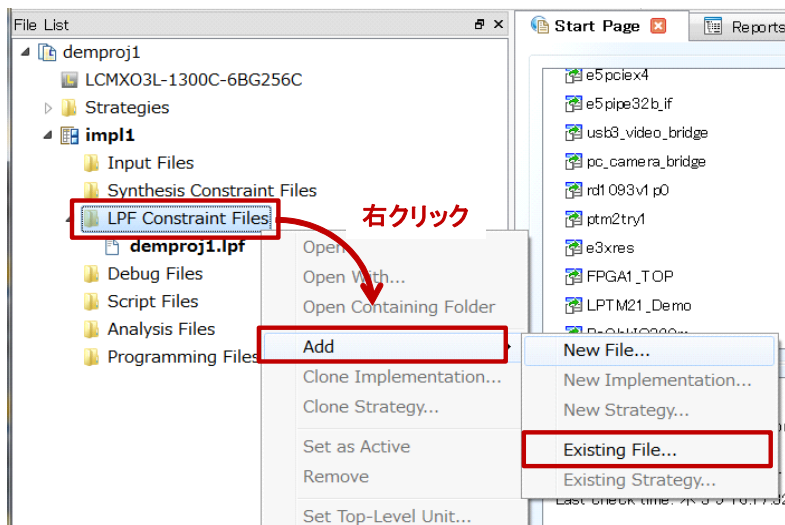
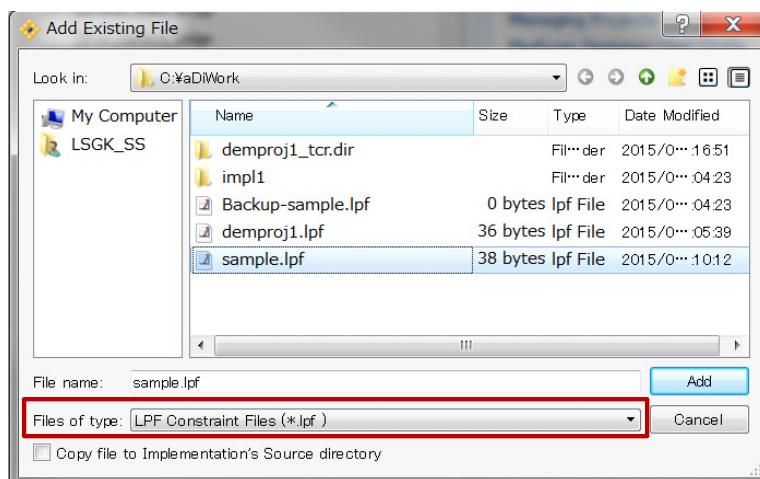


図 2-32. インポートする LPF 制約ファイルの選択



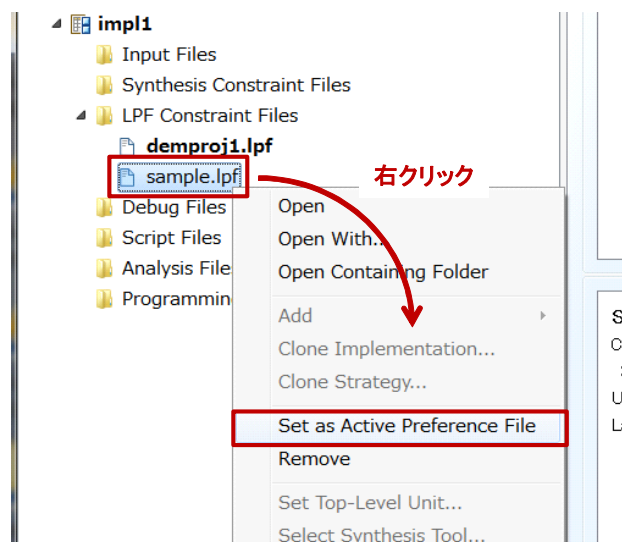
この際、ウィンドウ左下の [Copy file to ...] にチェックが入っていると、インプリメンテーションフォルダにコピーが作成され、そちらがインポートされます。チェックが入っていない場合は、選択したファイルがインポートされます。

2.5.3 使用する LPF 制約ファイルの選択

インプリメンテーションでは、プロジェクトに複数ある LPF 制約ファイルの内、アクティブ化されている 1 つが使用されます。このため「LPF 制約ファイルのアクティブ化」が「使用する LPF 制約ファイルの選択」

になります。アクティブ化するには、File List ウィンドウでアクティブでない LPF 制約ファイルを右クリックし、[Set as Active Preference File] を選択します (図 2-33)。以後そのインプリメンテーションで使用されます。アクティブ化された LPF 制約ファイルは太字で表示されます。

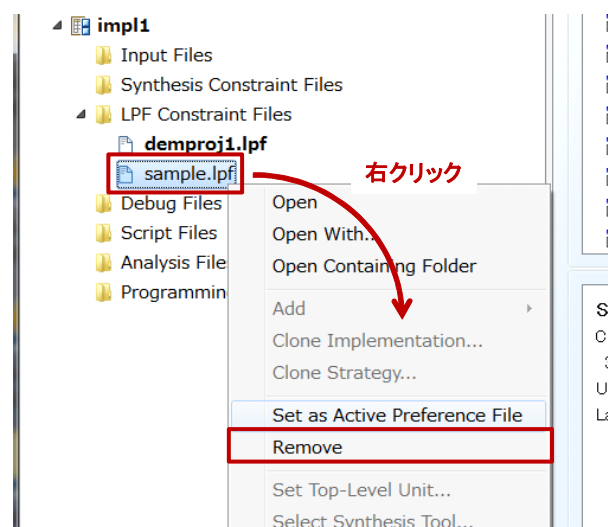
図 2-33. LPF 制約ファイルのアクティブ化



2.5.4 LPF 制約ファイルの削除

不要になった LPF 制約ファイルは、プロジェクトから削除することもできます。

図 2-34. LPF 制約ファイルの削除



File List ウィンドウで削除したい LPF 制約ファイルを右クリックし [remove] を選択するか、選択した状態でキーボードの [Delete] キーを押します (図 2-34)。インプリメンテーションでアクティブになっている LPF 制約ファイルは削除できません。

なお、非アクティブなインプリメンテーションでもアクティブな LPF 制約ファイルは削除できません。LPF 制約ファイルが 1 つしかインポートされていない場合は削除できません。「プロジェクトから削除」は「インポート情報の削除」であり、ファイル自体の削除ではありません。

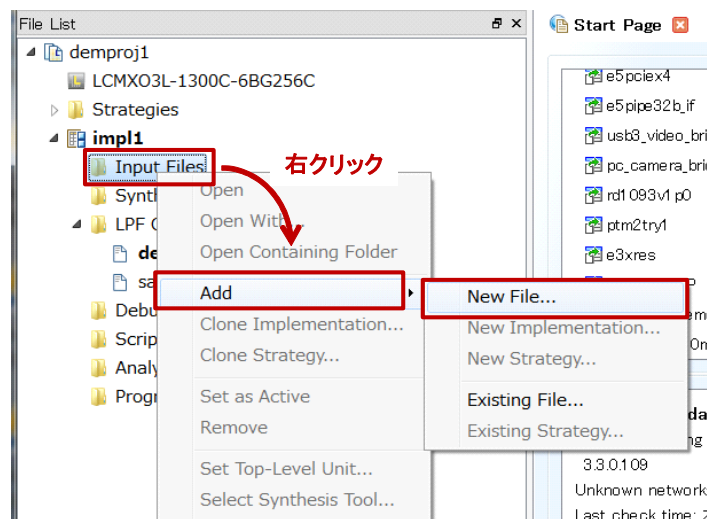
2.6 デザインファイルの管理

本項ではプロジェクト内でデザインファイル（HDL ソースや回路図等）を作成したりインポートしたりする方法について説明します。なお、RAM、PLL、SERDES といったマクロの作成方法については、別のドキュメントで説明します。

2.6.1 新規デザインファイルの作成

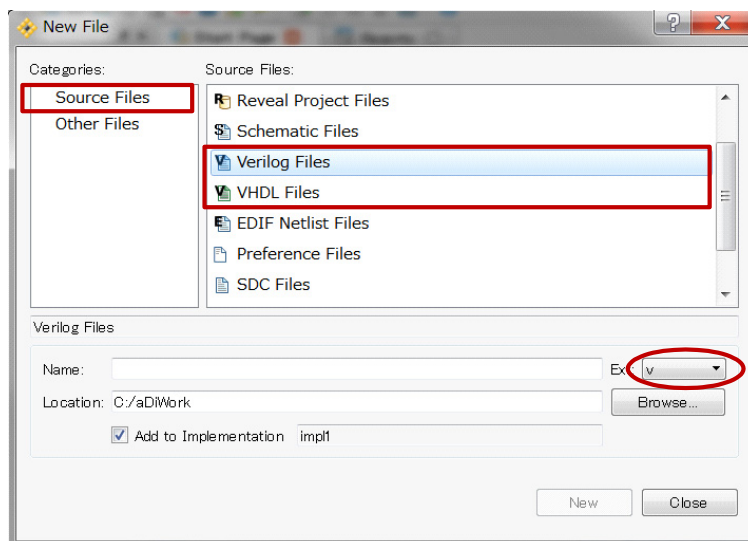
新規デザインファイルを作成する場合、メニューバーから [File] => [New] => [File...] の順に選択するか、もしくは File List ウィンドウでフォルダシンボル（ストラテジ以外）を右クリックし、[Add] => [New File...] の順に選択します。これで、New File Wizard が起動します。

図 2-35. 新規デザインファイルの作成



デザインファイルを追加する場合は、Categories 欄で [Source Files]、Source Files 欄で [VHDL Files]、[Verilog Files] または「Schematic Files」を選択し、Name 欄にファイル名を入力します。Ext. 欄（丸印）で拡張子を選択することもできます（選択したファイルに応じて適当なものが幾つか表示されます）。

図 2-36. デザインファイルの追加 (New File Wizard)



設定完了後に [New] ボタンをクリックすると、ファイルが作成され適当なエディタ（テキストエディタか回路図エディタ）が立ち上がります。なお、[Add to Implementation] ボックスにチェックが入っていると、ファイルは作成した時点でプロジェクトにインポートされます。

2.6.2 既存のデザインファイルのインポート

既に作成済みのデザインファイルをプロジェクトにインポートする場合は、メニューバーから [File] => [Add] => [Existing File...] の順に選択するか、もしくは File List ウィンドウでフォルダシンボル（ストラテジ以外）を右クリックし、[Add] => [Existing File...] の順に選択します（図 2-37）。これで、ファイルの選択ウィンドウが起動します。

図 2-37. 既存のデザインファイルのインポート

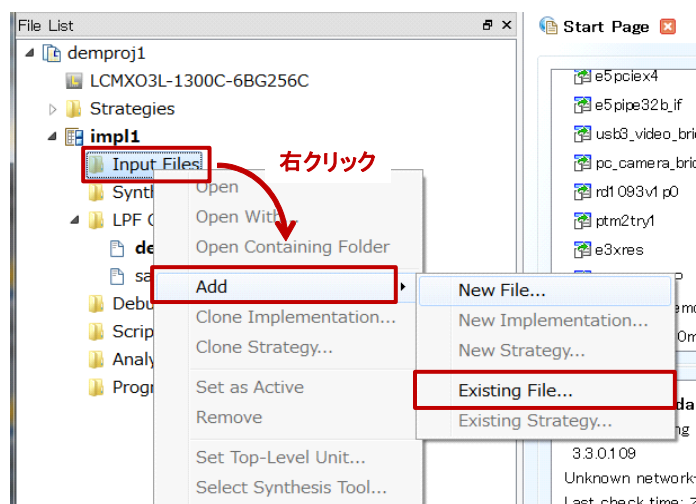
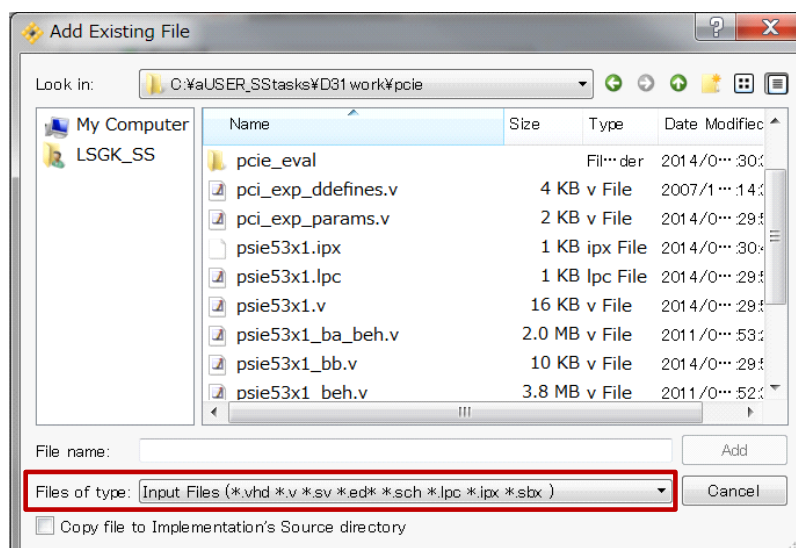


図 2-38. インポートするデザインファイルの選択



インポートするファイルの選択ウィンドウ（図 2-38）が立ち上がるので、まず Files of Type として [Input Files (*.vhd *.v ...)] を選択します。これでリストにデザインファイルが表示されるので、適当なファイルを選択し [Add] ボタンをクリックします。

この際、ウインドウ左下の [Copy file to ...] にチェックが入っていると、インプリメンテーションの source フォルダにコピーが作成され、そちらがインポートされます。チェックが入っていない場合は、選択したファイルがインポートされます。

2.6.3 対象プロセスの選択

インポートしたデザインファイルを使用するプロセス（シミュレーション、論理合成、または両方）を選択することができます。

*****-----

- ・ インポートした際のデフォルト設定は、シミュレーションおよび論理合成の対象として設定されています

*****-----

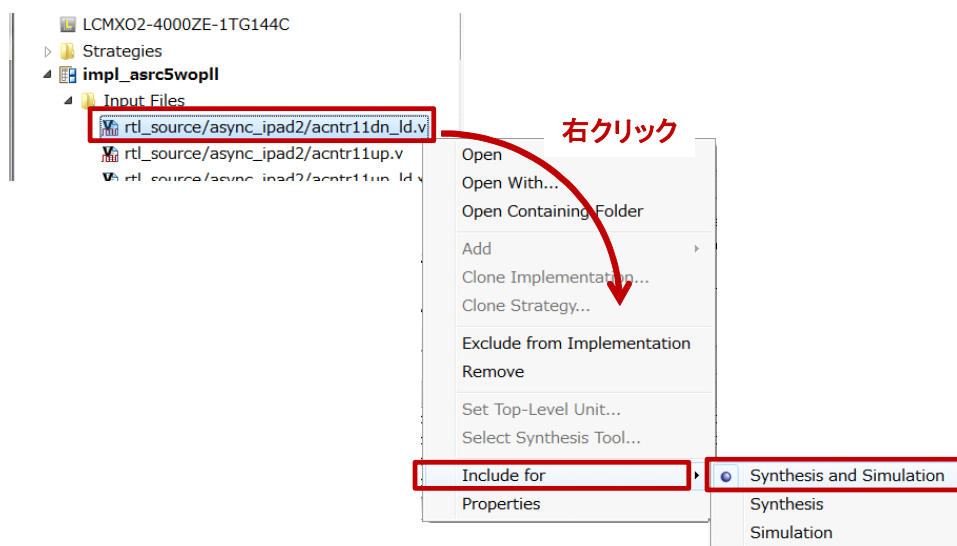
デザインファイルを右クリックし、[Include For] を選択すると対象プロセスのリストが表示されます。この中から 1 つを選択します。

[Synthesis] または [Synthesis and Simulation] を選択したソースは、論理合成の対象となります。[Simulation] または [Synthesis and Simulation] を選択したソースは、Simulation Wizard でシミュレーション用コンパイルスクリプト作成時に、デフォルトでコンパイル対象としてリストアップされます。

使用例

- ・ テストベンチをインポートして [Simulation] に設定
- ・ IP コアを使用する場合に、シミュレーション用のビヘイビアモデルと合成用のブラックボックスモデル両方をインポートしてそれぞれ [Simulation]、[Synthesis] に設定

図 2-39. 対象プロセスの選択



2.6.4 使用しないデザインファイルの選択

インポートしたデザインファイルは、インプリメンテーションごとに特定のファイルを処理の対象から除外することもできます。

*****-----

- ・ 例えば（内部の記述が異なるが）同じ entity/module 名のファイルを複数インポートしておき、インプリメンテーション毎に使用するファイルを切り替える場合などに使用する機能です
- ・ remove と違い、インポート情報が削除されません

*****-----

File List ウィンドウで処理の対象から外したいファイルを右クリックし、[Exclude from Implementation] を選択すると、そのファイルは現在のインプリメンテーションの処理対象から除外されます (図 2-40)。

対象から除外されたファイルは、薄い文字で表示されます。

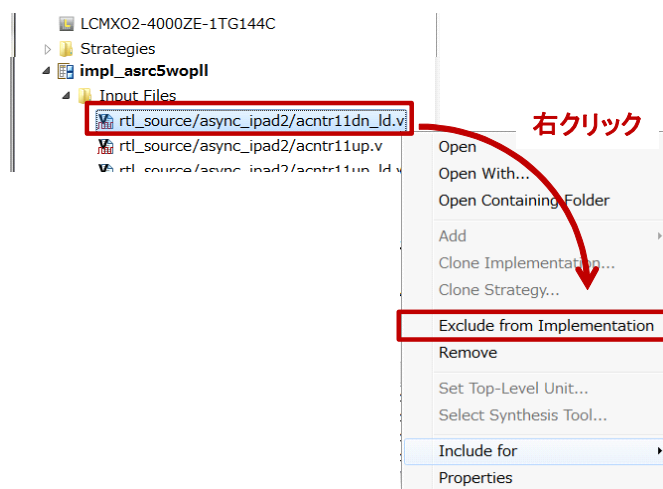
*****-----

- ・ この設定は選択されているインプリメンテーションでのみ有効です。他のインプリメンテーションには反映されません

*****-----

なお、一度除外したファイルを再び処理対象に戻す場合は、ファイルを右クリックして [Include in Implementation] を選択します。

図 2-40. デザインファイルの除外 (Exclude)



2.6.5 デザインファイルの削除

不要になったデザインファイルは、プロジェクトから削除することもできます。File List ウィンドウで削除したいデザインファイルを右クリックし [remove] を選択するか、選択した状態でキーボードの [Delete] キーを押すと、デザインファイルがプロジェクトから削除されます。ファイル自体は削除されません。図 2-40 で『Remove』を選択します。

2.6.6 デザインソースのプロパティ設定

インポートされている HDL ソースのプロパティとして、個別に以下の設定を行うことができます。

- ・ VHDL ソースのコンパイル先ライブラリ名
- ・ Verilog HDL ソース内でインクルードされているファイルの検索パス

これらの設定を行う場合は、メニューバーから [Project] => [Property Pages] の順に選択するか、File List ウィンドウでデザインファイルを右クリックして [Properties] を選択します (図 2-41)。これで、Project Properties ウィンドウが立ち上がります。

このウィンドウの左側でプロパティを設定したいデザインファイルを選択すると、ウィンドウ右側に設定可能な項目が表示されます。

なお、インプリメンテーション作成時に既存のインプリメンテーションにインポートされているソースを参照した場合、これらのプロパティ設定も反映されます。従ってインプリメンテーション毎に設定しなおす必要はありません。

図 2-41. デザインファイルのプロパティ設定

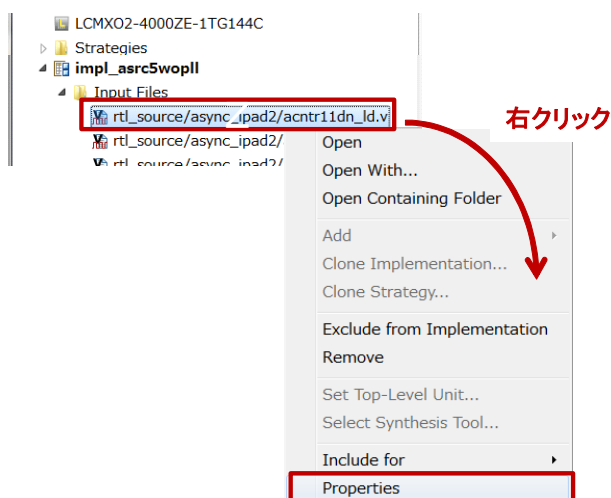
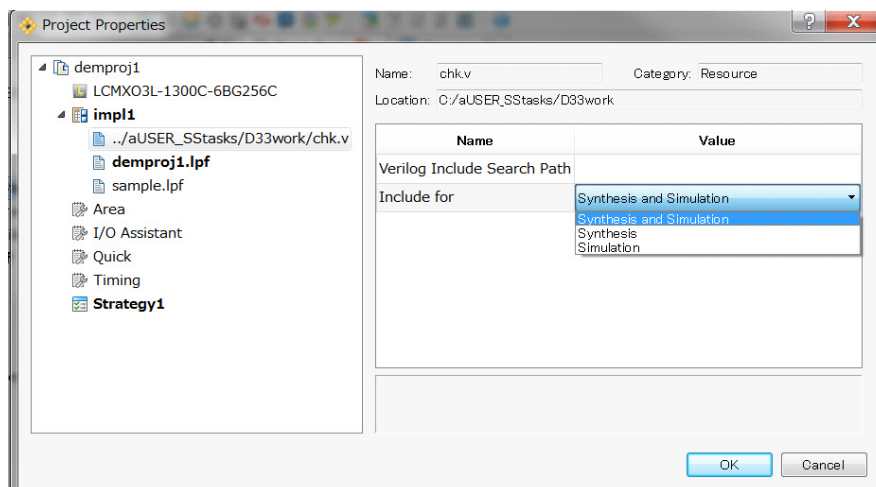


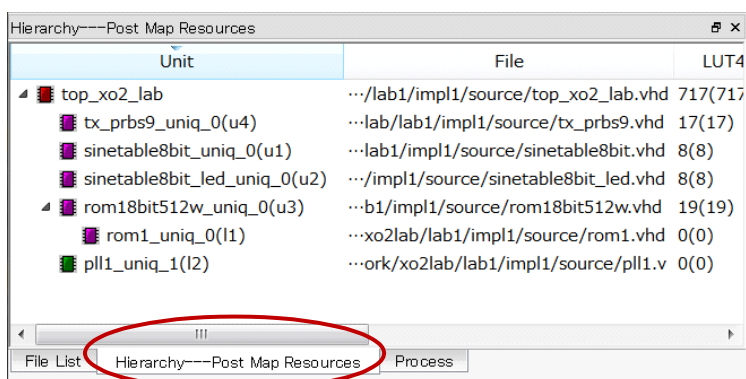
図 2-42. プロパティ・ウィンドウ ~ デザインの設定



2.6.7 デザイン（ファイル）の階層構造表示

インポートしたファイルは File List ウィンドウではフラットに表示されるため、ファイルの過不足が直感的には分かりません。

図 2-43. Hierarchy ビューによるモジュールの階層構造表示

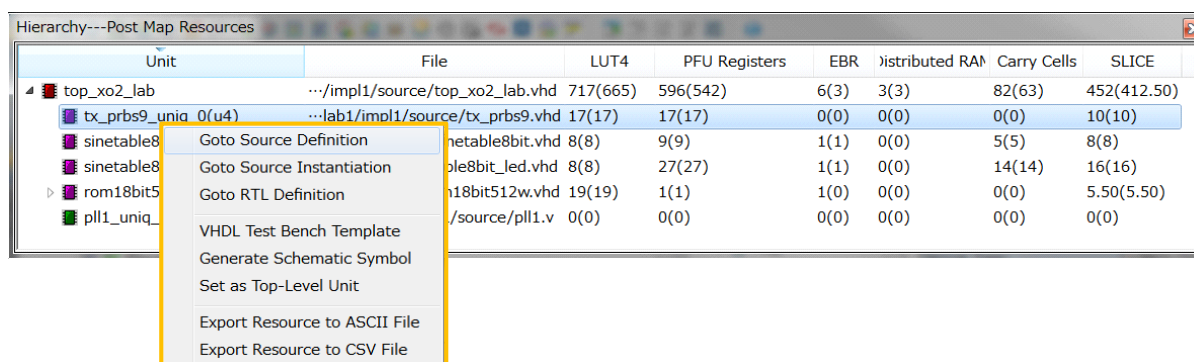


Diamond 3.2 以降はグラフィカル表示ツールの HDL Diagram がありませんので、図 2-43 のようにハイアラーキ（Hierarchy）ビューでモジュールの階層構造が期待通りかどうかを確認できます。

ハイアラーキ・ビューは論理合成前後で表示内容が変わります。論理合成後は LUT、FF など各リソースの使用数が各モジュール毎、階層ごとに表示されます（図 2-44）。RTL 既述の想定する規模に合致するか、大凡の判断が可能です。また、回路規模を目標に収めるための方策を検討するデータとなります。

図 2-44 のようにモジュールを選択して右クリックすると、種々メニューが表示されます。『Goto Source Definition / Instantiation』等を選択すると、設定しているエディタが開き、モジュール既述やインスタンス箇所にジャンプして表示します。『Export Resource to ASCII / CSV File』ではハイアラーキビューの表示内容がテキストファイルとして書き出すことができます。

図 2-44. ソースとのリンク、エクスポートなど



2.7 プロジェクト管理の注意点

2.7.1 Windows と Linux 間のデータ移転

Windows マシンで作成したプロジェクトを Linux マシンにコピーした場合、HDL ソースや制約ファイル等のインポート情報が Linux 上の Lattice Diamond 上で正しく認識されない場合があります。これは、Windows 上で作成したファイルを（そのまま）Linux 上へコピーした際に、フォントや文字コードの変換が原因でファイル名の太文字／小文字が変更されてしまうことがあるためです。Linux は太文字小文字を区別しますので、ファイル名の太文字／小文字が変換されると正しく認識できなくなります。

回避策は 2 つあります。

1. Linux にコピーしたソースファイル名を全て変更する（拡張子は小文字のまま）
2. Windows からコピーするファイルを tar と gzip で圧縮（拡張子が tgz or tar.gz）し、コピーした圧縮ファイルを Linux 上で展開する

2.7.2 自動生成マクロのパラメータファイル

SERDES や一部のマクロの HDL ソースは、GUI ツール IPexpress で生成しますが、この際に HDL ソースと共に動作モード設定等のパラメータを記述したテキストファイルも生成します。配置配線やタイミング解析プロセス、或いは書き込みデータ生成時には、このファイルを読み込んでパラメータを抽出します。これらのパラメータファイルは、**インプリメンテーション・フォルダに置いておく必要があります**。書き込みデータ生成プロセスでは、プロジェクトフォルダにあっても適用されますが、ファイル管理の観点からインプリメンテーション・フォルダに置くことを推奨します（IPexpress のデフォルト設定では、ファイルの出力先はインプリメンテーション・フォルダではなくプロジェクトフォルダです）。

プロセス実行時にインプリメンテーションフォルダにテキストファイルが無い場合、ツールは以下の様なメッセージを出力し、マクロはデフォルトのパラメータで処理が行われます。

WARNING – Cannot open configure file: [モジュール名].txt. Use the default setting !

このメッセージが表示された場合、IPexpress で設定したファイルの出力先から該当するテキストファイルをインプリメンテーション・フォルダにコピーしてください。テキストファイルを Lattice Diamond にインポートしても適用されませんので、注意してください。

2.8 改訂履歴

Ver.	page	内容
3.1	2-9	2.2.5 クローン・プロジェクトの生成、追加
	2-10	2.2.6 プロジェクト・アーカイブ項、復元について追記
	2-16	2.3.5 クローン・インプリメンテーションの生成、追加
3.3	--	バージョン 3.3 (3.2) 用に GUI キャプチャを全面的に更新
	2-26	HDL Diagram 既述を削除し、Hierarchy ビューによる既述に更新

--- *** ---