第 5 章　論理合成プロセス

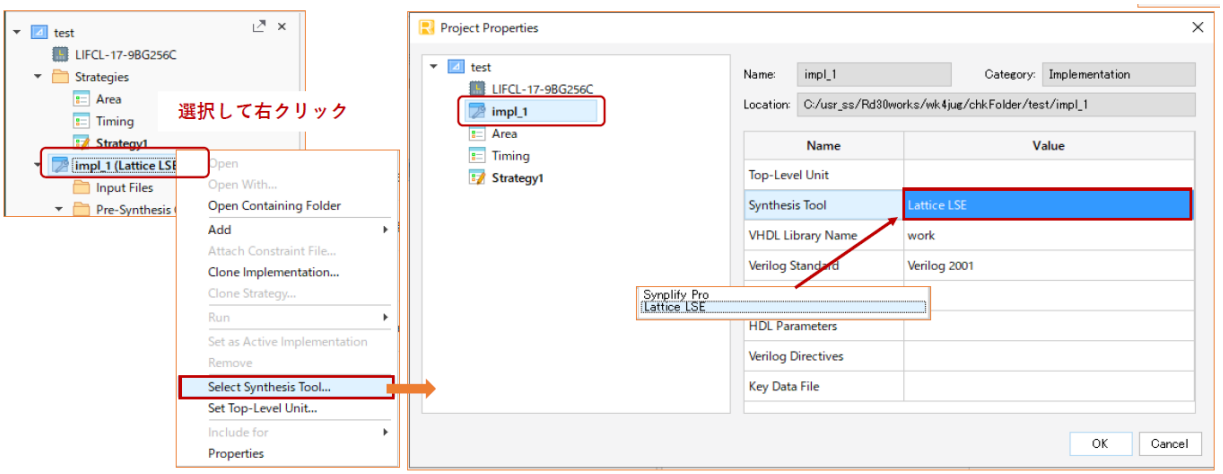
5.1 論理合成ツールについて

Lattice Radiant の論理合成ツールには Lattice Synthesis Engine（以下 LSE）と Synplify Pro for Lattice（以下 Synplify Pro）があります。LSE が新規プロジェクト作成時のデフォルトですが（図 2-7 参照）、指定を変えることができます（第 2.2.4.2 項にも関連記述）。

5.1.1 論理合成ツールの選択

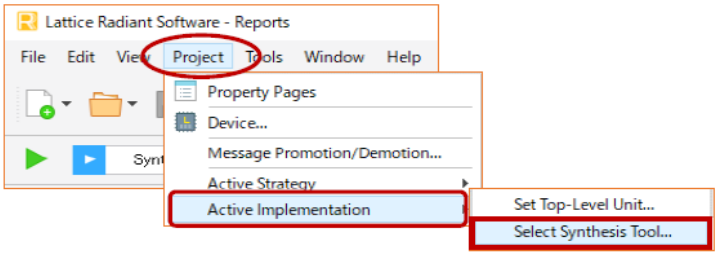
論理合成ツールはインプリメンテーションごとに指定できますが、プロジェクト作成後にツールを変更する方法は幾つかあります。第一の方法では、まずアクティブなインプリメンテーション名を選択後右クリックし、[Select Synthesis Tool...] または [Properties] を選択します（図 5-1）。

図 5-1. 論理合成ツールの変更操作 1



表示される ”Project Properties” ウィンドウで、左枠でアクティブなインプリメンテーションが選択されている状態で、右枠表内 Name カラムの ”Synthesis Tool” 行に、その時点で選択されている論理合成ツールが表示されます。[Value] カラムのどこかをクリックすると、プルダウン形式で選択可能なツール名が表示されます（Synplify Pro と LSE）。所望のツールを選択後、『OK』ボタンをクリックします。

図 5-2. 論理合成ツールの変更方法 2



別の方法として、Radiant のメニューバーから [Project] → [Active Implementation] → [Select Synthesis Tool...]

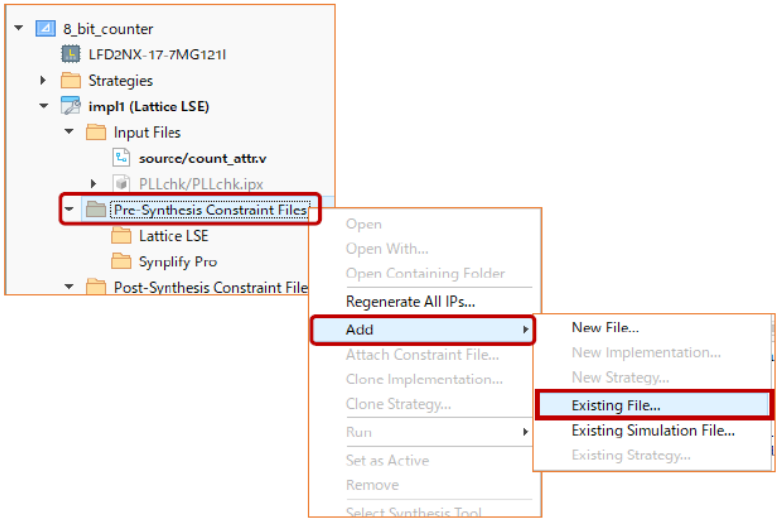
を選択します（図 5-2）。図 5-1 と同じプロパティ・ウィンドウが表示されますので、同様に行います。

5.1.2 論理合成制約ファイルの取り込み

一般的なフローでは、論理合成用ストラテジーの各オプション設定がデザイン全体（グローバル）に適用されます。例えば、デザインに 25MHz と 125MHz の二系統のクロックがある場合でも、ストラテジーの Frequency 制約として 125MHz を与えると、両方に適用される動作になります。

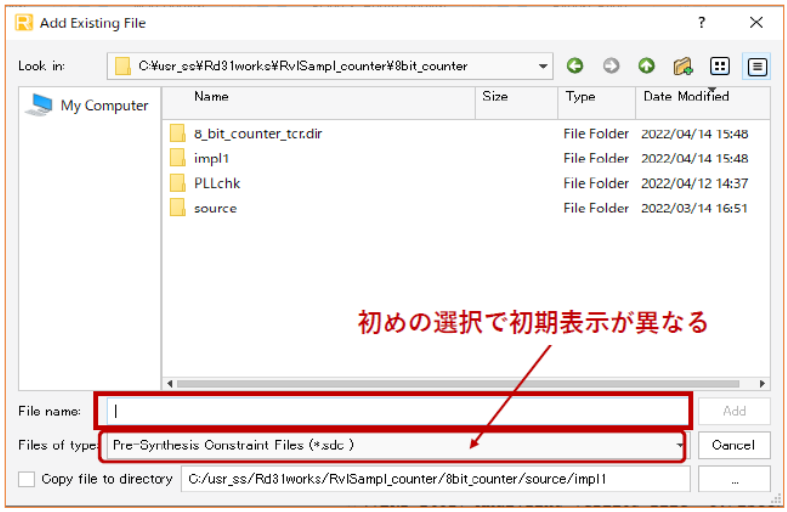
一方、論理合成ツール用の制約ファイル (Pre-Synthesis Constraint File) を用いることで、より詳細なコントロールが可能になります。この例の場合、25MHz のクロックには 25MHz を、125MHz のクロックには 125MHz の制約というように、クロックネットごとに個別に周波数制約を与えることが可能です。制約ファイルがなくても Radiant フローの処理自体には支障ありませんが、論理合成結果は異なります。

図 5-3. 既存の論理合成制約ファイルのインポート



プロジェクト作成時には、論理合成制約ファイルがない状態です。作成済み論理合成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...] と選択するか、”File List” ウィンドウ内 [Pre-Synthesis Constraint Files] 行か、その下にある [Lattice LSE] 行か [Synplify Pro] 行をのいずれかを選択後に右クリックし [Add] → [Existing File...] を選択します（図 5-3）。

図 5-4. インポートする制約ファイルの選択



ファイル・インポートのウィンドウ(Add Existing File)が立ち上がりまず。図 5-4 は [Pre-Synthesis Constraint Files] 行を選択した場合の例で、「File of Type」は ”*.sdc” が選択されています。ここでは示しませんが、[Lattice LSE] 行の場合は ”*.ldc” が、[Synplify Pro] 行の場合は ”*.fdc” がファイルタイプになります。このバーをクリックすることで、自動選択されている拡張子は変更することができます。ブラウズして適切なファイルを選択後、『Add』ボタンをクリックします。

このとき、ウィンドウ下部の「Copy file to directory」ボタンにチェックを入れると、その右セルで指定するフォルダーにコピーされ、それをインポートします。デフォルトはチェックが入っていない状態で、選択ファイルをコピーせずに、そのままインポートします。

各インプリメンテーションには複数の論理合成制約ファイルをインポートできますが、アクティブ（有効）なものは唯一です。アクティブな制約ファイルは太字で、非アクティブな制約ファイルはグレーで表示されます。

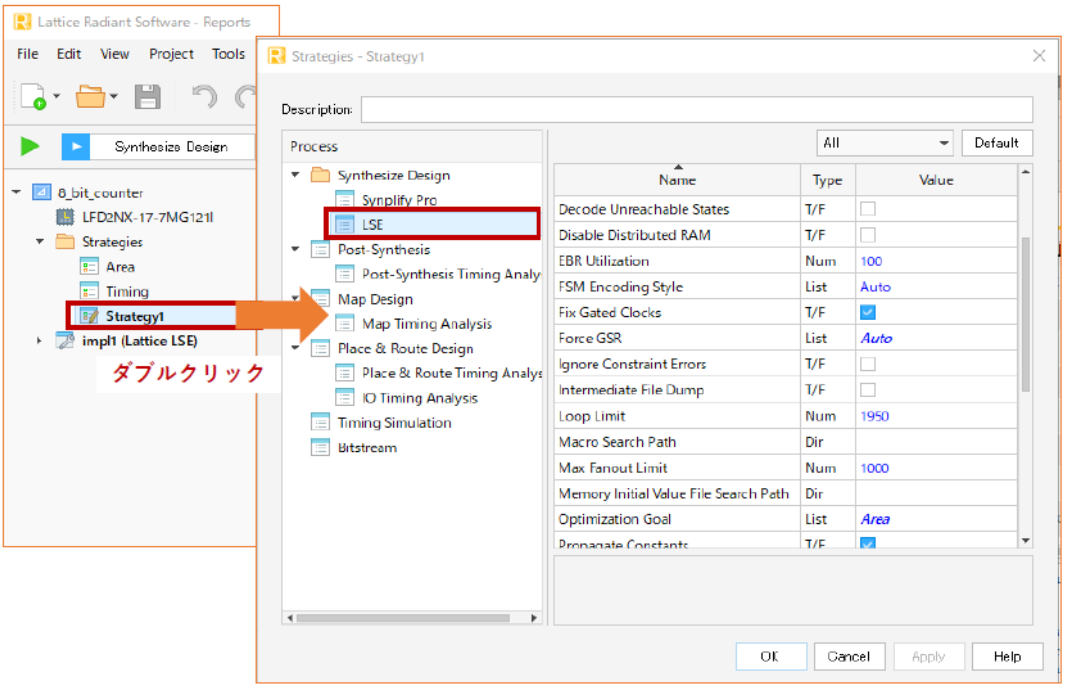
アクティブ化と非アクティブ化の切り替えは、意図するファイルををクリックして選択し、右クリックすると表示されるプルダウンから [Set as Active] または [Set as Inactive] を選択します。制約ファイルは削除できます。同様に選択後に右クリックで [Remove] を選択するか、選択した状態でキーボードの『Delete』キーを押します。” プロジェクトから削除 ” はインポート情報の削除で、ファイル自体の削除ではありません。

5.2 LSE

5.2.1 ストラテジー・オプション項目

GUI 左側のファイルリスト・ビューの上部は「Strategies」セクションで、プロジェクトで定義されているストラテジーの一覧が表示されています。太字表示が、インプリメンテーションに適用されるアクティブなものです。論理合成ツールとして LSE を指定している状態でストラテジー名をダブルクリックすると、LSE のストラテジー設定ウィンドウが開きます（図 5-5）。

図 5-5. LSE のストラテジー設定ウィンドウ



以降に個々のオプションについて記述します。オリジナルはオンラインヘルプ [Reference Guide]→[Strategy Reference Guide] → [LSE Options] と選択することでご参照いただけます。

Allow Duplicate Modules

パラメータ：True / False　　デフォルト：False

同一モジュールが重複している場合、デフォルトではエラーになり、処理を終了します。[True] の場合はウォーニングを出力しますが、モジュール定義の最後のものを用いて処理を行います（それ以前の定義は用いません）。

Carry Chain Length

パラメータ：数値　　デフォルト値 ：0（制限なし）

カウンターや演算回路で使用されるキャリー（桁上がり／下がり）チェイン・プリミティブ　CCU（Carry Chain Cells）の使用数に関するオプションで、単一キャリー・チェインの最大数を指定します。

この設定は [Use Carry Chain] オプションが [True] の場合のみ有効です。

Command Line Options

パラメータ：文字列　　デフォルト：ブランク

GUI のリストに表示されていない特別なオプションを使用する場合に、直接引数等を記述します。コマンドの詳細についてはオンラインヘルプの [Running SYNTHESIS from the Command Line] 項をご参照ください（[Reference Guides] → [Command Line Reference Guide] → [Command Line Tool Usage]）。

DSP Style

パラメータ：DSP / Logic　　デフォルト：DSP

演算機能の実装方法を指定します。デフォルトは DSP マクロを使用し、[Logic] では汎用ロジック（LUT+FF）を使用します。

DSP Utilization

パラメータ：数値　　デフォルト値：100

DSP マクロ使用率 [%] の上限値を指定します。

Decode Unreachable States

パラメータ：True / False　　デフォルト：False

[True] にすると、FSM（ステートマシン）が未定義ステートに遷移した場合の動作記述（Verilog の ”default”/VHDL の ”others” ステート）を残します（セーフ・リカバリー動作）。

Disable Distributed RAM

パラメータ：True / False　　デフォルト：False

ボックスをチェックして[True]にすると、メモリー推論記述RTLで分散メモリー(Distributed Memory)を使用せずに、EBR を用います。

EBR Utilization

パラメータ：数値　　デフォルト値：100

EBR 使用率 [%] の上限値を設定します。設定した値を超える場合、残りは分散メモリー（Distributed Memory）で実現されます。

FSM Encoding Style

パラメータ：Auto / Binary / Gray / One-Hot　　デフォルト：Auto

FSM のステート・エンコード方式に関する設定で、Auto（デフォルト）では、RTL 記述ステート数から LSE が決定します。Auto 以外では、その指定に従ったエンコードの FSM が生成されますが、Gray は 4 ステート未満でのみ指定が有効です。

Fix Gated Clocks

パラメータ：True / False　　デフォルト：True

デフォルトでは、RTL 記述でゲーティングしているクロック信号を、FPGA 実装に適する構成に変換します。ただし、対象クロックは *.ldc ファイルに ”create_clock” 制約で明記されている必要があります。また、変換するためには全てのゲーティング論理が分解できること、など一定の条件があります。変換できない場合は、スキューの大きいクロックネットとして実装されますので、注意が必要です（基本的にクロック・ゲーティングはしないことが重要です)。

変換されたクロックと関連情報はログファイル ”synthesis.log” に書き出されます。

Force GSR（LFCPNX, LFD2NX, LIFCL, UT24C, UT24CP）

パラメータ：Auto / Yes / No　　デフォルト：Yes

GSR（Global Set/Reset）使用に関して、デフォルト [Yes] では使用し、[No] では使用しません。[Auto] では LSE が自動的に判断します。

Ignore Constraint Errors

パラメータ： True / False　　デフォルト：False

デフォルトでは制約記述にエラーがあると処理を終了し、エラーメッセージを出力します。[True] ではエラー記述は無視し、処理を継続します。

Intermediate File Dump

パラメータ： True / False　　デフォルト：False

[True] にすると暗号化 Verilog HDL ファイルを出力します（拡張子 *.ve)。デフォルト（False）では出力されません。何らかの問題が生じた際などに、デザインのソース記述を開示せずにブラックボックスとして提供し、Lattice から技術支援を受けることができます。

Loop Limit

パラメータ：数値　　デフォルト値：1950

RTL ソース記述内の ”for” や ”while” ループ回数の最大値を設定します。ループ・インデックスが定数ではなく変数として定義されている場合にのみ適用されます。小さすぎる値は無視されます。大き過ぎると、スタック・オーバーフローを生じる可能性がありますので、ご注意ください。

Macro Search Path

パラメータ：フォルダーパス　　デフォルト：ブランク

デザインで IP コアなどのマクロファイル（*.ngo）を使用している場合のみ、そのフォルダーパスを指定します。ただしプロジェクト・フォルダーかインプリメンテーション・フォルダーに当該ファイルがある場合、或いは RTL 内で *.ngo ファイルの（相対パスではなく）絶対パスを ”FILE” アトリビュート指定で明記している場合は、ここで指定する必要はありません。

複数のフォルダーを指定する場合は、” ; ” で区切ります（間にスペースは入れない)。パスの記述は相対パスでも有効です。

Max Fanout Limit

パラメータ：最大ファンアウト数　　デフォルト値：1000

ファンアウト上限数を制約します。ファンアウトがこの値を超えた場合は、ドライバーを複製してファンアウト数を減らします。

Memory Initial Value File Search Path

パラメータ：フォルダーパス　　デフォルト：ブランク

EBR（ブロックメモリー）の初期値設定を行う際に、初期値を記述したファイル（*.mem）が保存されているフォルダーパスを指定します。何も指定しない場合は、インプリメンテーション・フォルダーが参照されます。EBR の初期値設定を行わない場合は、記述する必要はありません。また、モジュール生成ツールで生成する際に初期化ファイルを指定している場合は、ここで再度指定する必要はありません。

Optimization Goal

パラメータ：Area / Timing　　デフォルト：Timing

デフォルト [Timing] はロジック段数を少なくする速度重視で最適化します。*.ldc に ”create_clock” で制約が記述されている場合、[Target Frequency] オプションは無視します。

[Area] は使用リソース数が少なくなるように最適化します。[Use IO Registers] オプションが [Auto] になっている場合、入出力レジスターは I/O パッドセルのものを使用します。

Propagate Constant

パラメータ：True / False　　デフォルト：True

デフォルトは（論理やモジュールの）入力に固定値になっている場合に、関連する回路を最適化（削除）します。[False] を選択すると、回路は最適化されずに残ります。

RAM Style

パラメータ：Auto / Block_RAM / Distributed / Registers　　デフォルト値：Auto

RAM を推定する場合の実現方法（リソース）を設定します。[Block RAM] では EBR が、[Distributed] は分散 RAM が、[Registers] ではフリップフロップがそれぞれ使用されます。デフォルトは LSE が最適なものを選択します。

ROM Style

パラメータ：Auto / EBR / Logic　　デフォルト：Auto

ROM を推定する場合の実現方法を設定します。[EBR] では EBR を使用し、[Logic] では分散メモリかスライス内のロジックリソースで構成します。デフォルトは LSE が最適なものを選択します。

Remove Duplicate Registers

パラメータ：True / False　　デフォルト：True

全く同じ機能（論理）のフリップフロップが複数ある場合、デフォルトは冗長な分を最適化（削除）してリソース数を削減します。[False] では削除せずに残します。

Remove LOC Properties

パラメータ：On / Off　　デフォルト：Off

RTL ソース内記述の配置制約（LOC）に関して、[On] にするとその指定は削除されます。

Resolve Mixed Drivers

パラメータ：True / False　　デフォルト：False

[True] を選択すると、特定のネットがアクティブな信号と VCC（または GND）から共にドライブされている場合、VCC（または GND）に固定します。

Resource Sharing

パラメータ：True / False　　デフォルト：True

同一機能の回路が複数ある場合、使用リソース数を減らして共用する回路を生成（リソース・シェアリング）します。[False] ではこの処理は行いません。

Target Frequency (MHz)

パラメータ：数値（周波数）　　デフォルト値：ブランク

クロック周波数（MHz）ターゲットの設定で、デザイン内に複数のクロックがある場合でも、全てが対象となります。LDC 制約で個別に指定しているクロックには適用されません。

Use Carry Chain

パラメータ：True / False　　デフォルト：True

カウンターや演算回路に含まれる加算器に関して、キャリー（桁上がり／下がり）チェインの使用をイネーブルします。[False] では使用しません。

Use IO Insertion

パラメータ：True / False　　デフォルト：True

論理合成時に IO パッドと GSR を挿入します。[False] では挿入しません。

Use IO Registers

パラメータ :Auto / False / True　　デフォルト：Auto

[Tue] ではタイミング制約を踏まえて、I/O パッドセル内の I/O レジスタを使用することを強制します。[False] では I/O レジスタを使用しません。デフォルト（Auto）では、[Optimization Goal] オプションが [Area] の場合に I/O レジスタを使用しますが、[Timing] では使用しません。

VHDL 2008

パラメータ：True / False　　デフォルト：False

[False] にすると、VHDL 言語仕様として VHDL 2008 を適用します。

5.2.2 LSE 制約ファイルの生成

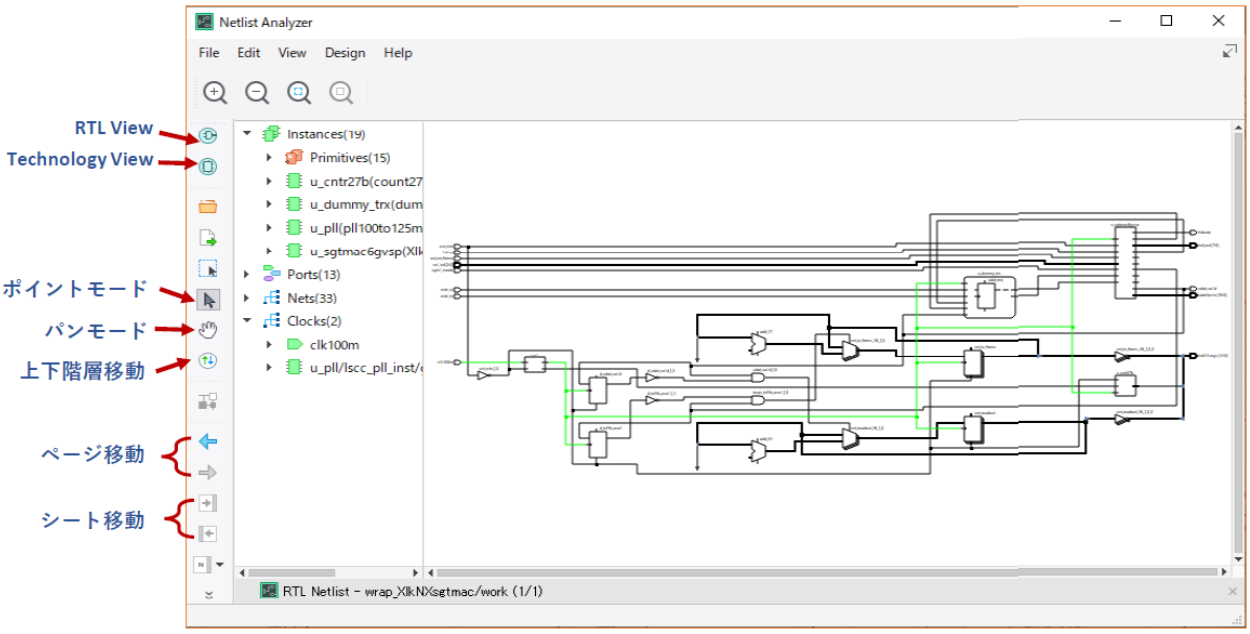
LSE 用の論理合成制約ファイル *.ldc はブランク状態から作成できますが、これは LDC 書式に習熟している場合にのみ推奨します。基本的にはタイミング制約エディター（Timing Constraint Editor: TCE）を用いることをお勧めします。その詳細については第 15 章をご参照ください。

新規作成は、メニューバーから [File] → [New] → [File...] の順に選択すると表示される「New File」ウィンドウで [LSE Design Constraints Files] を選択して（ファイル名を入力後に）開始します。または、インプリメンテーションのファイルリスト（File List）枠内「Pre-Synthesis Constraint Files」セクション下「Lattice LSE」を選択後に右クリックして [Add] → [New　File...] と選択しても「New File」ウィンドウが表示されます。

5.2.3 ネットリスト・アナライザー

LSE を論理合成に選択した場合、回路情報をグラフィカルに閲覧するツールがネットリスト・アナライザーです。起動するにはアイコンバーの中の  をクリックするか、メニューから [Tools] → [Netlist Analyzer] を選択します。

図 5-6. ネットリスト・アナライザーの表示例とアイコン



起動後の表示（デタッチしたウィンドウ）は図 5-6 の例のようになり、RTL View スケマティックが表示されます。アイコン列はウィンドウの左端に並んでいて、上部には RTL View / Technology View アイコンや、階層移動、ページ移動などのアイコンがあります。

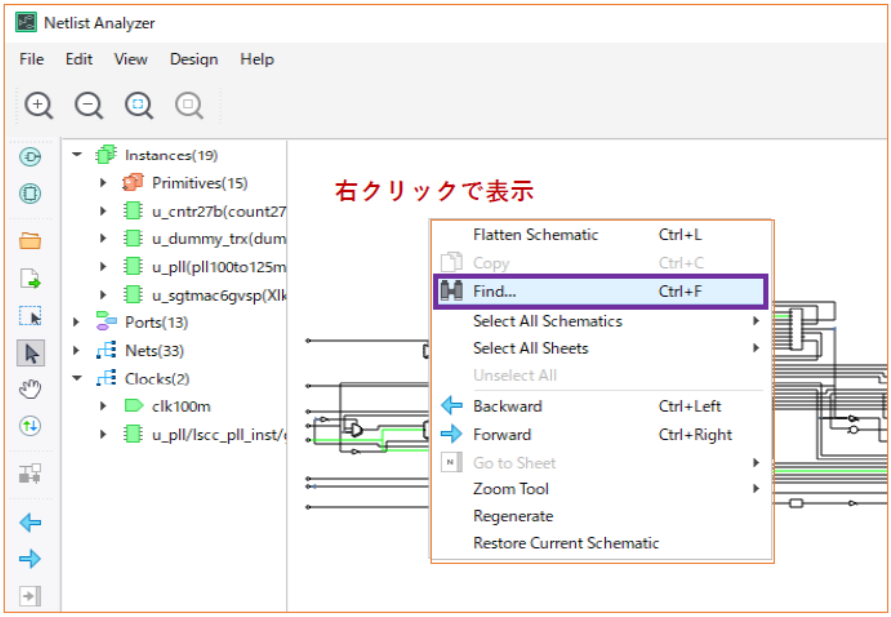
その右はリソースリストの表示ウィンドウで、Instances / Ports / Nets / Clocks の４つに分かれています。図 5-6 は Instances と Clocks 行先頭の▸をクリックして展開した状態の例です。他も同様に展開して意図するリソースを選択すると、スケマティック内で当該リソースが赤色でハイライトされます。

デザインの実装は階層構造にするのが一般的ですが、モジュール階層の上下移動のアイコンをクリックしてから、スケマティック表示内の（サブ）モジュールの箱の上にアイコンを移動すると、下位階層がある場合はアイコンが下向き青矢印表示になります。クリックすると下位階層のスケマティック表示になります。上位階層への移動は何も回路素子がない位置にアイコンを移動する事で、上向き緑矢印が表示されますのでこれをクリックして戻ることができます。

5.2.3.1 オブジェクトのサーチ

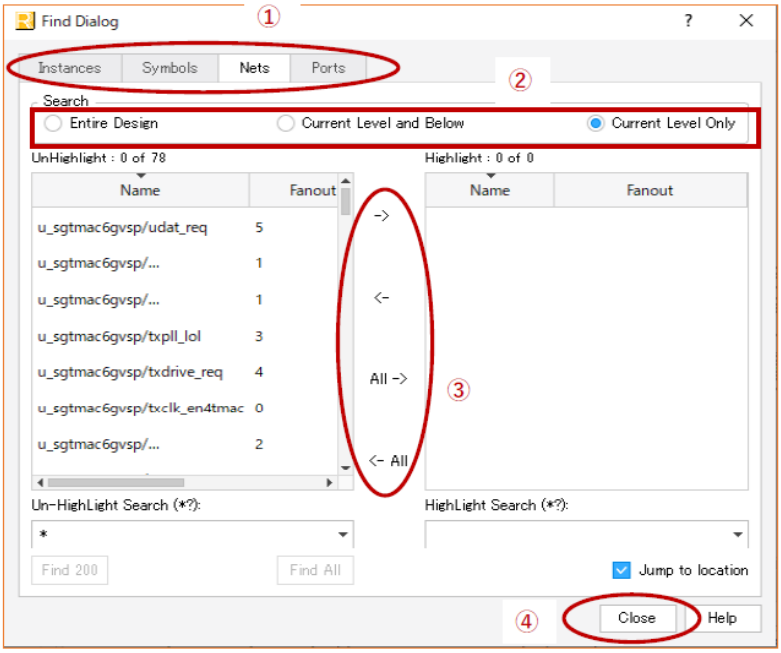
特定のリソースやネットを探すときのサーチ機能があります。スケマティック内ブランク領域にマウス・ポインターを移動し、右クリックすると表示されるメニューから [Find...] を選択します。

図 5-7. Find ダイアログの起動



表示されるウィンドウ（図 5-8）で、まず①上部タブ [instances] / [Symbols] / [Nets] / [Ports] から意図する対象リソースを選択します。次に② ”Search” セクションでサーチ対象の階層を指定します。デフォルトは ”Current Level Only”（ダイアログを立ち上げた際にマウスクリックした階層）です。その後、③左側に表示される対象候補を選択して矢印『->』などを操作して ”Highlight” 枠に移動させて『Close』をクリックします。各矢印で選択指定と解除を適宜行います。

図 5-8. Find Dialog ウィンドウ



5.3 Synplify Pro

5.3.1 ストラテジー・オプション項目

オプション設定ウィンドウの起動と操作方法は LSE と同じです（図 5-5 参照）。以降にストラテジー・オプションの意味について記述します。オリジナルはオンラインヘルプ [Reference Guide] → [Strategy Reference Guide] → [Syplify Pro Options] と選択することでご参照頂けます。

Allow Duplicate Modules

パラメータ：True / False　　デフォルト：False

同一モジュールの重複時、デフォルトでは処理を終了します。[True] にすると処理を継続します。

Area

パラメータ：True / False　　デフォルト：False

[True] にすると [Frequency] オプションの設定値を無効にし、使用リソースが最少となるように回路を最適化します。*.sdc/*.fdc 制約ファイルでクロック周波数（周期）を制約している場合は、その設定が優先されます。

Arrange VHDL Files

パラメータ：True / False　　デフォルト：True

デフォルトでは VHDL ファイルのコンパイル順を自動的に操作します。[False] にすると Radiant のファイルリスト・ビューで表示されている順に、上からコンパイルされます。

Automatic Read/Write Check Insertion for RAM

パラメータ：True / False　　デフォルト：False

推論記述の RAM に関して、[True] にすると、同一アドレスにリードとライト同時アクセスが生じる場合のシミュレーション・ミスマッチを防ぐため、グルーロジックを生成して挿入します。

Clock Conversion

パラメータ：True / False　　デフォルト：False

[True] にすると、ゲーテッド・クロックや分周クロックの記述に対して、クロック・イネーブルを用いる回路に変換します。

Command Line Options

パラメータ：文字列　　デフォルト：ブランク

GUI のリストに表示されていない、特別なオプションを指定する場合に記述します。

Default Enum Encoding

パラメータ：Default / Gray / Onehot / Sequential　　デフォルト：Default

VHDL の列挙（enum）型変数が定義されている場合、その実装形式に関する設定です。デフォルト（Default）ではメンバー数に応じて自動的に割り振られます。4 以下では ”Sequential”、40 以上では ”Gray”、それ以外は ”Onehot” です。

[Gray] では各メンバーに一度の変化で 1 ビットのみが変化するような値が割り当てられ（例 :000, 001, 011, 111, 110, 010）、[Sequential] では一度に複数 bit が変化するような値（基本的にはバイナリー・カウンター）が割り当てられます。[Onehot] では各メンバに全ビット中 1 ビットのみが ’1’ となる値が割り当てられます（例 :0001, 0010, 0100, 1000）。

Disable IO Insertion

パラメータ：True / False　　デフォルト：False

デフォルトでは論理合成時に IO パッドを挿入します。[True] にすると I/O パッドのないネットリストを生成しますが、マッピング以降の通常の処理には使用できませんので、ご注意ください。ユーザーモジュールのブラックボックス化など、意図する場合のみ [True] にします。

Disable Register Replication during S/R Optimization (LFCPNX, LFD2NX, LIFCL, UT24C, UT24CP)

パラメータ：True / False　　デフォルト：False

デフォルトでシフトレジスターの推論記述で RAM を生成した場合に、アドレスポインターを複製します。[True] にすると複製しません。

Export Radiant Settings to Synplify Pro GUI

パラメータ：No / Only on First Launch / Yes　　デフォルト：Yes

Synplify Pro GUI を Tools メニューやアイコンから立ち上げて論理合成を行う場合に、ストラテジー設定各オプションをどのように渡す（エクスポートする）かを指定するオプションです。

デフォルトでは、Synplify Pro GUI が起動されるごとにストラテジー設定を渡し、それ以前のセッションで保存されている設定は無効になります。[Only on First Launch] では、初回起動時のみにストラテジー設定を渡し、それ以後はそのセッションでの設定やデフォルトを用いて起動します。[No] は渡しません。

FSM Compiler

パラメータ：True / False　　デフォルト：True

Synplify Pro にはステートマシン用の解析・合成エンジンである FSM（Finite State Machine）コンパイラーが組み込まれています。デフォルトでは使用され、[False] にすると使用されません。

Fanout Guide

パラメータ：最大ファンアウト数　　デフォルト値：1000

ファンアウト上限数を設定します。ファンアウトがこの値を超えた場合は、ドライバを複製してファンアウトを減らします。

Force GSR

パラメータ：Auto / False / True　　デフォルト：Auto

論理合成時の GSR（Global Set/Reset）使用に関する設定です。[True] では GSR バッファを使用し、[False] では使用しません。[Auto] では自動的に判断します。

Frequency（MHz）

パラメータ：周波数　　デフォルト値 :200

クロック周波数（MHz）ターゲットの設定で、[Area] オプションが [False] の場合のみ有効です。デザイン内に複数のクロックがある場合、全てが対象となります。*.sdc/*.fdc 制約ファイルで個別に設定しているクロックは除外されます。ブランク時はクロック周波数が最速になるような回路を合成するように動作します。

Library Directories

パラメータ：ディレクトリー　　デフォルト：ブランク

デザインに含まれる Verilog ライブラリーのディレクトリー・パスを指定します。ファイル名とモジュール名が一致している必要があります。

Number of Critical Paths

パラメータ：パス数　　デフォルト値：なし

論理合成結果のタイミング・レポートに出力するクリティカルパス数を指定します。

Number of Start/End points

パラメータ：数値　　デフォルト値：なし

論理合成結果のタイミング・レポートに出力するパス数の設定です。この値が 1 以上の場合のみに [Number of Critical Paths] が有効です。

Output Netlist Format

パラメータ：None / VHDL　　デフォルト：None

論理合成後シミュレーション用のネットリスト出力設定です。[VHDL] にすると暗号化 VHDL ネットリスト（*.vhm）が生成されます。本オプション設定に関わらず暗号化 Verilog ネットリスト（*.vm）は生成されます。

Pipelining and Retiming

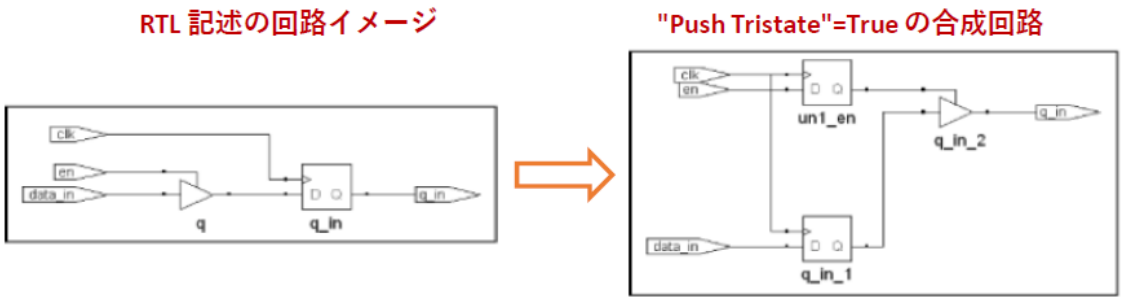
パラメータ：None / Pipelining Only / Pipelining and Retiming　　デフォルト：Pipelining Only

論理合成中に行うタイミングの最適化処理 ”Retiming”（リタイミング）と ”Pipelining”（パイプライン）に関する設定です。

リタイミングは組み合わせ回路を移動してレジスタ間同士の遅延量のバランスをとるようにする処理です。回路全体に対してこの処理が行われます。パイプライン処理は主に演算回路に対して行う処理で、回路の中間にレジスタ（パイプライン・レジスター）を挿入するものです。後者では入力から出力が得られるまでのサイクル数が、RTL 記述とは異なる可能性があります。

デフォルトはパイプライン処理のみ、[Pipelining and Retiming] は両方を行います。[None] ではどちらも行いません。

図 5-9. Push Tristate オプションの作用



Push Tristates

パラメータ：True / False　　デフォルト：True

デフォルト（True）では、ハイインピーダンス 'Z' がレジスターを介してポートに出力されるような記述に対して、トライステート制御用レジスターを追加し、IO バッファでトライステート制御を行うように回路を論理合成します（図 5-9）。

[False] では、FPGA 内部で 'Z' が生成されないような回路になります。

Resolve Mixed Drivers

パラメータ：True / False　　デフォルト：False

[True] を選択すると、特定のネットがアクティブな信号と VCC（または GND）から共にドライブされている場合、VCC（または GND）に固定します。

Resource Sharing

パラメータ：True / False　　デフォルト：True

同一機能の回路が複数ある場合、使用リソース数を減らして共用する回路を生成（リソース・シェアリング）します。[False] ではこの処理は行いません。

Update Compile Point Timing Data

パラメータ：True / False　　デフォルト：False

コンパイルポイント（compile point）はインクリメンタル・デザインフローに関する機構です。階層構造の下位モジュールが変更された場合に、デフォルト（False）ではその上位モジュールは再合成されませんが、[True] を選択した場合は、上位モジュールが再び論理合成されます。下位モジュールの変更に伴い、タイミング・データとタイムスタンプが更新されます。

Use Clock Period for Unconstrained I/O

パラメータ：True / False　　デフォルト：False

SDC/FDC ファイルでタイミング制約が与えられている I/O についてのみ制約を適用します。[True] にすると未制約の I/O にも同じ制約が与えられます。

VHDL 2008

パラメータ：True / False　　デフォルト：False

[True] にするとプロジェクトの VHDL 言語仕様として VHDL 2008 を適用します。

5.3.2 論理合成のタイミング解析とストラテジー項目

論理合成プロセスでのタイミング解析にははスライス（LUT/FF）などのエレメント遅延、データパスの配線遅延、およびクロック配線遅延などが関わります。エレメント遅延はツールにロードされている所定の値を用いますが、データパスとクロックの配線遅延については、特定のアルゴリズムで推定値を使用します。

ストラテジー・オプションによってタイミング解析条件やレポートスタイル等を変更することができます。アクティブなストラテジーをダブルクリックして設定ウィンドウを表示し、左側 "Process" 枠で "Post-Synthesis Timing Analysis" を選択します。

Number of End Points

パラメータ：数値　　デフォルト :10

クリティカル・エンドポイント・サマリーでのエンドポイント数を指定します。

Number of Paths per Constraint

パラメータ：数値　　デフォルト :10

詳細パスレポートのパス数を指定します。

Number of Paths per Endpoint

パラメータ：数値　　デフォルト :1

各エンドポイントに対する最大パス数を指定します。

Number of Unconstrained Paths

パラメータ：数値　　デフォルト :10

タイミング制約でカバーされていないパスのレポート数を指定します。

Report Format

パラメータ：Lattice Standard / Diamond Styleデフォルト :Lattice Standard

[Lattice Standard] は業界標準ツールの書式に準じたレポート・フォーマットです。[Diamond Style] は既存ツール Lattice Diamond のタイミング・レポートと同じフォーマットです。

Timing Analysis Options

パラメータ：Hold Analysis / Standard Setup Analysis / Standard Setup and Hold Analysis

デフォルト：Standard Setup and Hold Analysis

デフォルトでセットアップ / ホールド時間解析の両方を同時に行います。[Hold Analysis] にすると、ホールド解析のみを行い、[Standard Setup Analysis] セットアップ解析のみを行います。

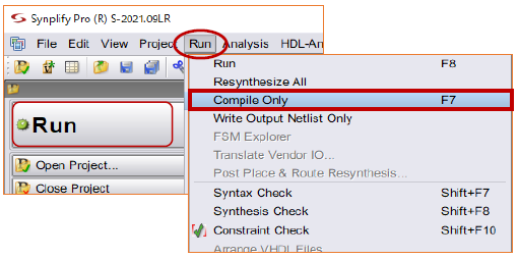
5.3.3 SCOPE エディターによる制約ファイルの生成

Synplify Pro が備える制約エディタ GUI である ”SCOPE エディター” を用いて ”Synopsis Design Constraint” としての制約ファイルが生成できます。以下に論理合成時に適用する制約ファイル ”*.fdc”（FPGA Design Constraints）を新規に生成する方法を記述します。RTL の文法エラー等はすべて解消されているものとします。

なお、Synplify Pro には sdc から fdc への変換コマンド ”sdc2fdc” がサポートされていますので、これについてもドキュメントをご参照ください。

5.3.3.1 SCOPE エディターの起動

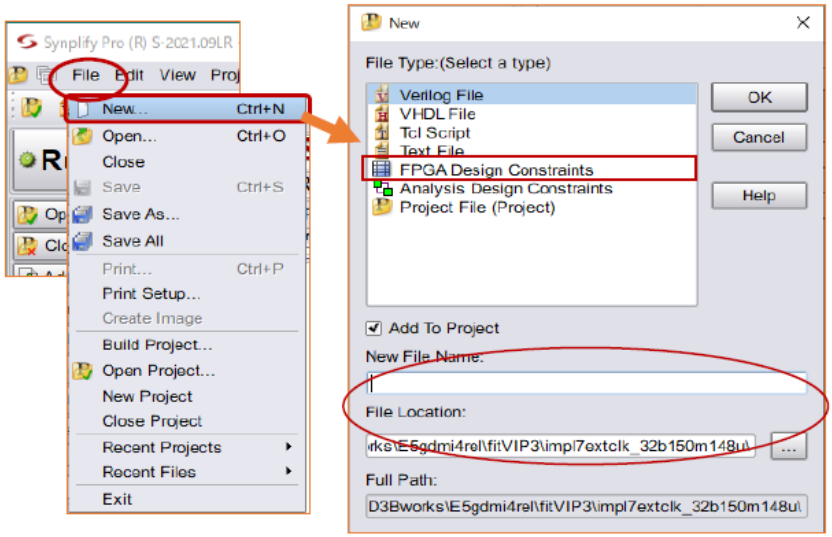
図 5-10. 『Run』またはコンパイルのみを実行



まず Lattice Radiant から Synplify Pro GUI を立ち上げます。その後 GUI トップで左上の大きなボタン『Run』をクリックして論理合成を実行するか、[Run] メニューから [Compile Only] を選択してコンパイルのみを実行します（図 5-10）。エラーがある場合は、これ以降の作業ができませんので、ご注意ください。

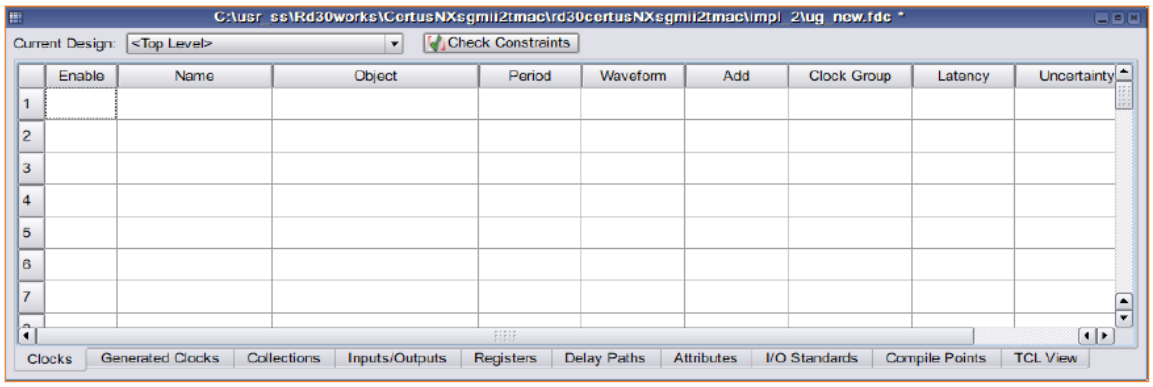
次にメニューから [File] → [New...] を選択して ”New” ウィンドウを立ち上げます（図 5-11）。fdc ファイル生成のため ”FPGA Design Constraints” を選択し、丸印内の「New File Name」セルにファイル名を入力し、「File Location」をブラウズして指定した後、『OK』をクリックします。

図 5-11. SCOPE エディタの起動



制約設定用ワークシート ”SCOPE エディター ” が立ち上がります（図 5-12）。

図 5-12. “FPGA Design Constraints” 入力用ワークシート GUI



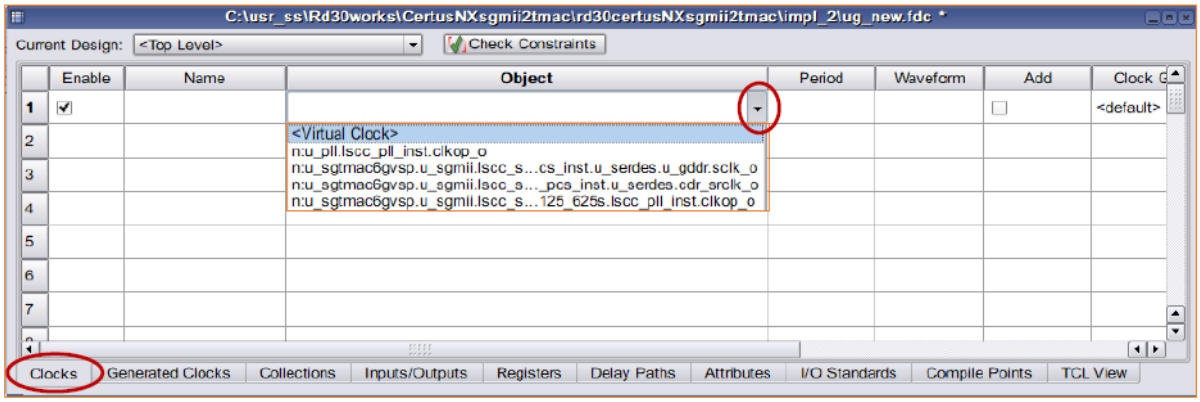
ウィンドウの下部には複数のタブがあります。与えたい制約のタブを選択した後に、それぞれを入力していきます。

5.3.3.2 クロック制約の設定

クロックについては、[Clocks] タブを選択後に空白の ”Object” セルをクリックすると、その右端に▼印が表示されます。これをクリックして現れるプルダウンの候補から意図するものを選択します。その後最低限 ”Name” に適切な名称を、”Period” セルの周期 [ns] を与えます。”Enable” セル表示に✔印が付いていることを確認します。無効にする場合はこの印を消します。

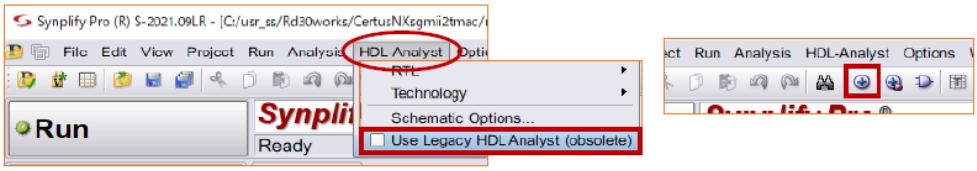
プルダウンに意図する候補が表示されない場合は、クロックとして認識されない RTL 記述か回路設計に何らかの要因がありますが、クロックネットやポートのオブジェクトを RTL スケマティック・ビュー GUI からドラッグ＆ドロップする入力方法があります。その場合は、まず上部メニューバー内の RTL ビューアイコン（右図の赤枠）をクリックして、RTL スケマティック・ビューを表示させます。

図 5-13. クロック・オブジェクトの選択



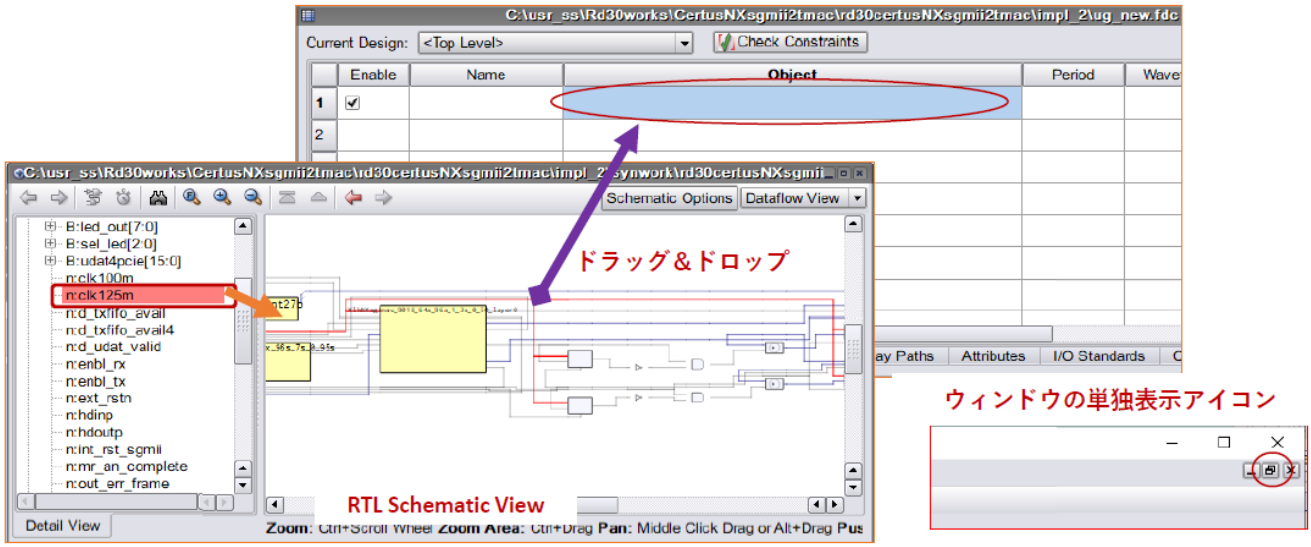
ここで、ビューの表示形式を旧バージョン相当にしたい場合は、メニューから [HDL Analyst] → [Use Legacy HDL Analyst (obsolete)] を選択しておきます（図 5-14 左図）。

図 5-14. RTL ビューの起動（右）と ”レガシー” 表示の選択（左）



次に図 5-15 右下の丸印で示す、ビューの右上端にある ”Restore Down” アイコン（どの表示でも構わない）をクリックして、複数のウィンドウ間でマウス移動ができるようにします。FPGA Design Constraints ワークシートでは [Clocks] タブを選択し、適切な表示位置に移動しておきます。

図 5-15. オブジェクトのドラッグ&ドロップとウィンドウの個別表示アイコン（右下）



- RTL ビューの左枠で Instances や Nets 項を展開し、意図するクロックネットを選択します（図 5-15 左枠内の赤色表示は、クロックネットが選択された状態）
- RTL ビューで赤線の当該クロックネットをクリックして選択した状態でドラッグしてワークシートの ”Object” セルにドロップします（図内の中央上）

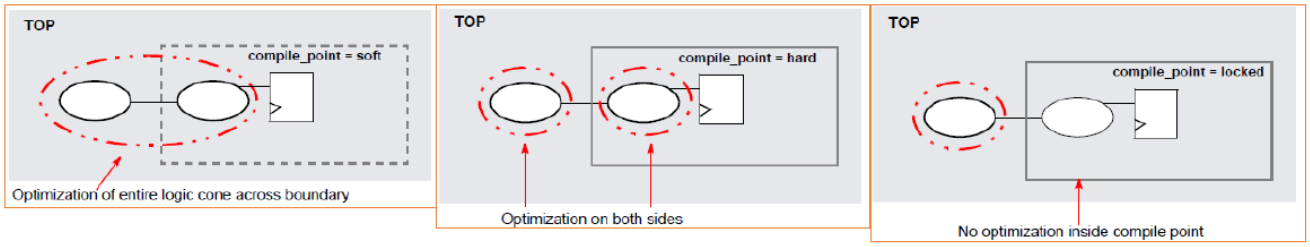
”Name” と ”Period” などを適宜入力します。これを必要なクロックネット分だけ繰り返します。

ワークシートで I/O タイミングなど、その他についても適宜タブを選択して入力を完了し、保存しておきます。

5.3.3.3 コンパイルポイントの設定

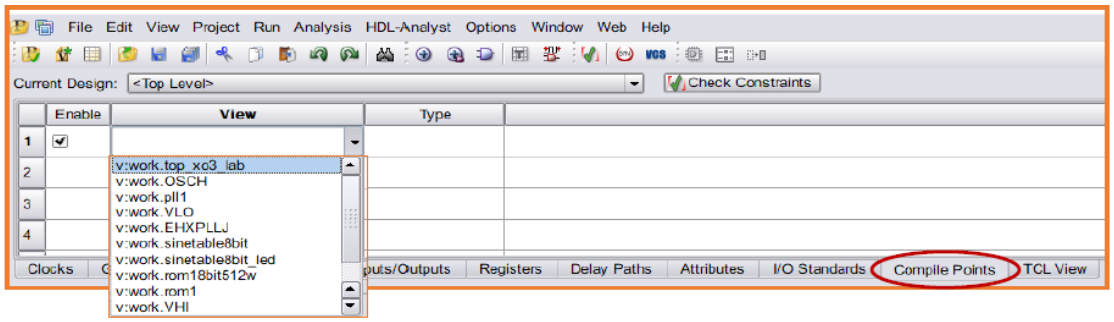
モジュールにコンパイルポイント（Compile point）を設定して論理合成すると、特に規模の大きなデザインでは論理合成時間を短縮できます。論理合成の実行時にコンパイルポイントごとに作成されるデータベースを参照して、変更されているソースファイルのみの論理合成を行います。

図 5-16. コンパイルポイント・タイプによる最適化の違い



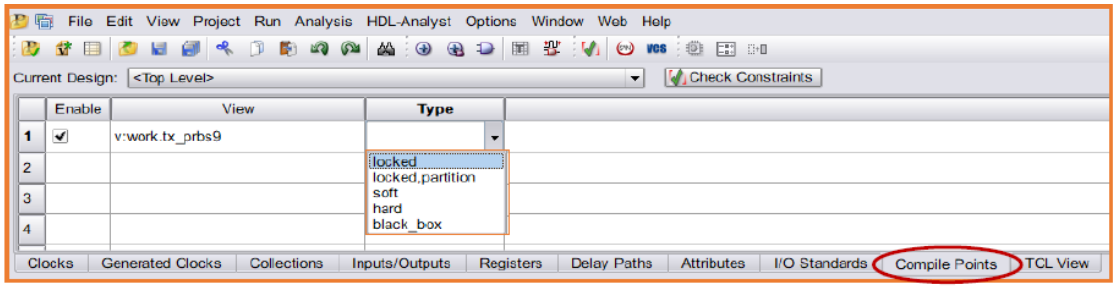
”soft” はモジュール間の接続分の回路をまとめて最適化します。”hard” の場合は接続部分の回路をコンパイルポイントごとに個別に最適化します。”locked” では接続部分の最適化は行われません。

図 5-17. SCOPE によるコンパイルポイントの設定 1



SCOPE ワークシートで設定する場合、[Compile Point] タブを選択後、空白の ”View” セルをクリックするとその右端に▼印が表示されますので（図 5-17）、これをクリックして現れるプルダウンの候補から選択します。

図 5-18. SCOPE によるコンパイルポイントの設定 2

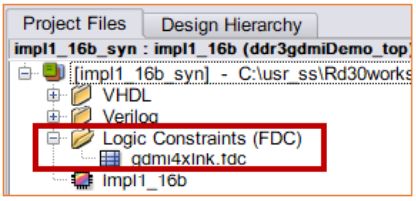


次に ”Type” セルをクリックし、同様にプルダウンから意図するタイプを指定します（図 5-18）。”Enable” セル表示に✔印が付いていることを確認します。無効にする場合はこの印を消します。

5.3.3.4 fdc 制約ファイルのプロジェクトへの取り込み

制約ファイル保存時に作業プロジェクトに含めるかどうかを問われます。生成が完了した後、プロジェクト表示ウィンドウ（[Project Files] タブ）で図 5-19 のように ”Logic Constraints (FDC)” 部にファイル名が表示されます。なお、以上の SCOPE エディターによる作業は、GUI 起動して作業した論理合成プロジェクトのみに対してであることに留意する必要があります。

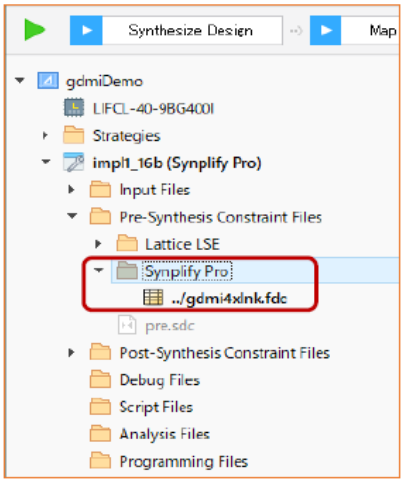
図 5-19. 制約ファイル fdc のリスト表示



5.3.3.5 fdc 制約ファイルのアクティブ化

作成した fdc ファイルを、Synplify Pro GUI の起動ではない、通常の Radiant フローにも適用されるようにする場合の手順です。Radiant のファイルリスト・ビュー内のインプリメンテーション下 ”Pre-Synthesis Constraint File” セクションに *.fdc を取り込み、アクティブ化（太字表示）します（図 5-20）。複数ある場合でも、有効（アクティブ）にできるのは一つです。これはインプリメンテーションごとに行います。

図 5-20. fdc 制約ファイルの取り込み例



5.3.4 Synplify Pro ドキュメント

Synplify Pro の使用方法やオプション設定に関する各種のガイドライン・ドキュメントは、以下のフォルダーにインストールされています(Radiant 3.2 をデフォルト設定でインストールした場合)。これらは、Synplify Pro GUI のメニューから [Help] → [PDF Documents] でも呼び出すことができます。

C:¥lscc¥Radiant¥3.2¥synpbase¥doc

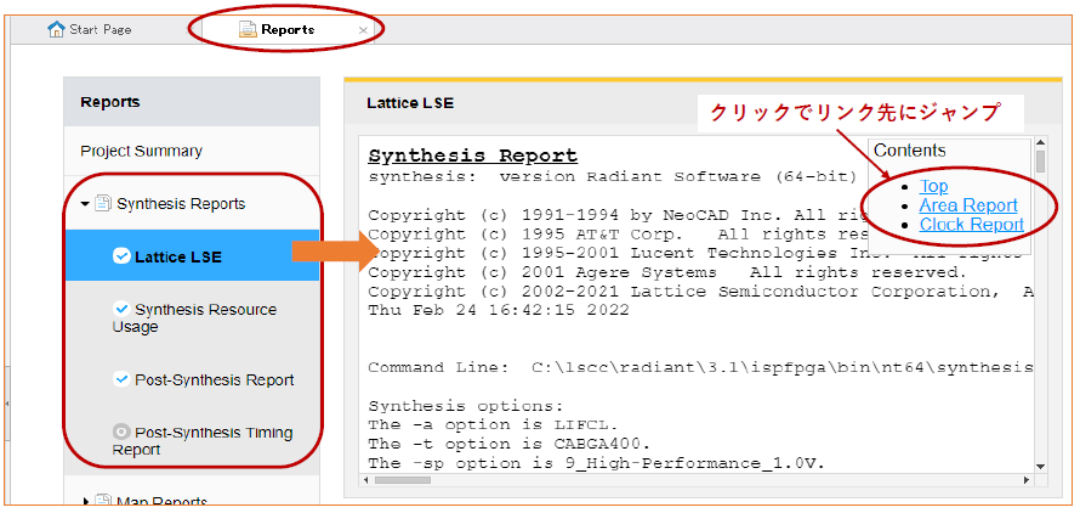
fpga_user_guide.pdf : 主に GUI 操作を含むデザインフロー全般についての記述
fpga_hdl_reference.pdf : HDL 言語サポート機能についての詳細記述（Verilog/VHDL/System Verilog）
fpga_attribute_reference.pdf : 論理合成アトリビュート（属性）についての記述

5.4 論理合成プロセス・レポート

5.4.1 プロセス実行ログ

論理合成プロセスを実行すると、レポート・ビューにレポートが表示されます。GUI 上部の [Reports] タブをクリックし、ウィンドウ左側のセクションで [Synthesis Reports] をクリックすると（図 5-21）、サブ項目として四つがリストされています。最初は選択しているツールによって ”Lattice LSE” または ”Synplify Pro” です。この項目と ”Post-Synthesis Report” 項の内容は基本的にツールの実行ログです。

図 5-21. 論理合成プロセス・レポート （LSE）

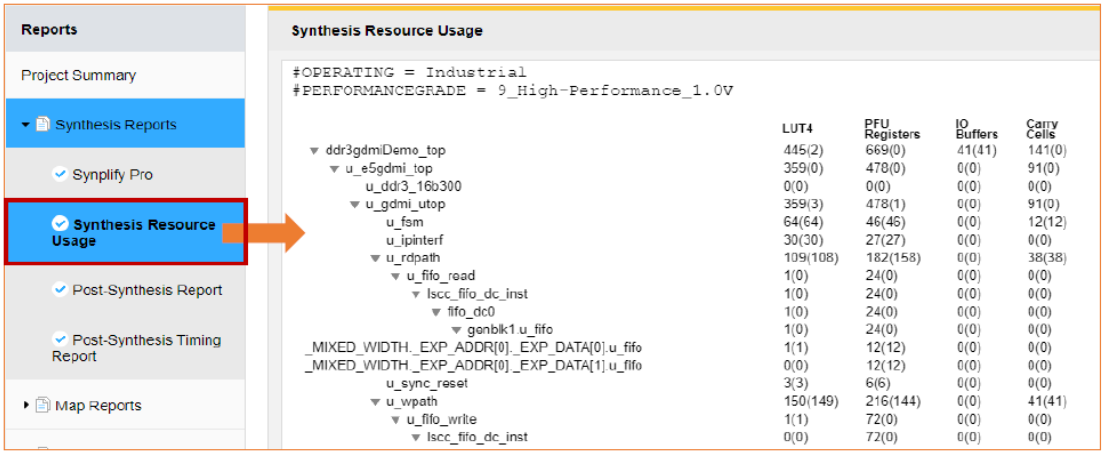


LSE の場合には ”Lattice LSE” 項を選択時の右側表示内で、右上に ”Contents” として三つのレポート内リンクがあります。これらをクリックしてクロックやエリア（リソース）の表示箇所にジャンプできます。Synplify Pro ではこのリンクはありません。

5.4.2 リソース・レポート

”Synthesis Resource Usage” 項をクリックすると、右側にはのようにデザイン階層ごとに使用されているリソースが表形式でレポートされます。階層表示はデフォルトでは下位階層すべて展開されていますが、▼印をクリックすれば展開しないでその階層についてのリソース行のみが表示されます。

図 5-22. 論理合成プロセスのリソース・レポート例



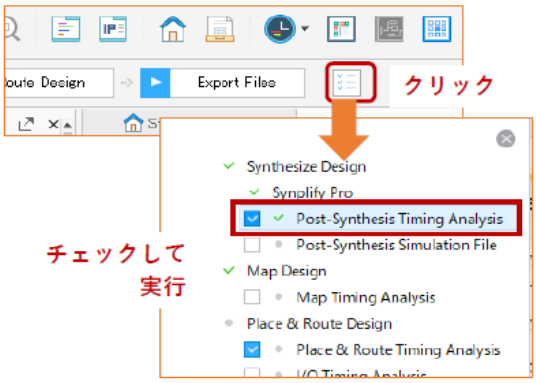
各リソース数は、数値とともに ”(値)” のように括弧付きの数値を伴っています。これは下位モジュールではなく、当該階層そのレベルでのリソース数を示しています。

5.4.3 論理合成後のタイミング・レポート

”Post-Synthesis Timing Report” 項は、他のサブ項目の行頭が 印で閲覧可能であることを示しているのに対して、デフォルトでは  印になっていて、有効なレポートが閲覧できる状態ではないことを示しています。

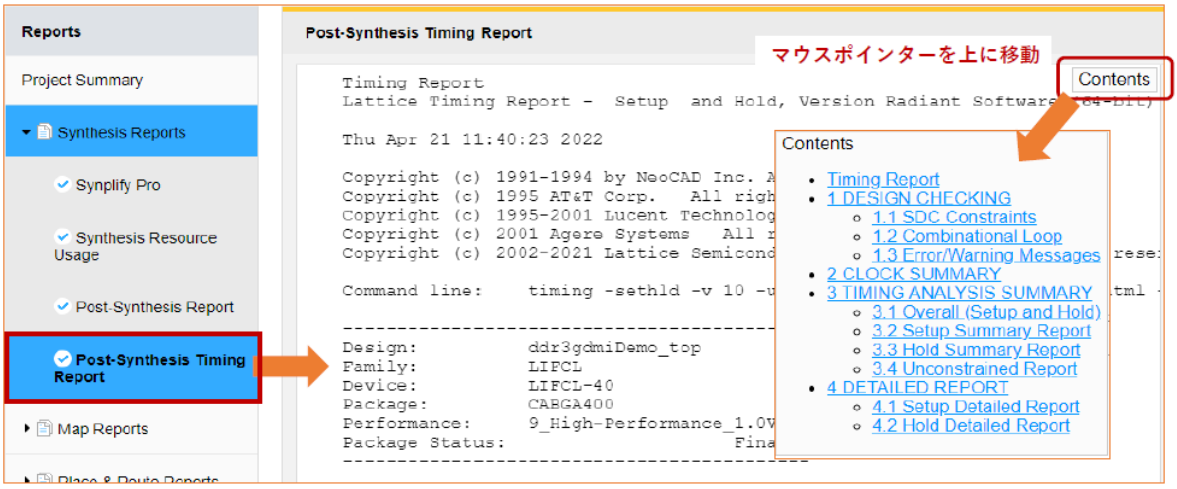
レポートを出力するためには、図 5-23 のように ”Task Detail View” アイコンをクリックすると現れるビュー内で該当するサブプロセス ”Post-Synthesis Timing Analysis” ボックスをクリックして有効化してから論理合成を実行するか、或いは論理合成を実行後にこのサブプロセスのみをダブルクリックして解析を実行します。

図 5-23. 論理合成後タイミング解析サブプロセスの実行



タイミング解析レポートの例を図 5-24 に示します。右側のレポート表示内で、右上に ”Contents” というボタンがあります。マウス・ポインターをこの上に移動すると、内容項目のリストを示すリンクが表示されますので、これらをクリックして所望の表示箇所にジャンプすることができます。

図 5-24. 論理合成プロセスのタイミング・レポート



1. DESIGN CHECKING

1.1 SDC Constraints	*.sdc/*.fdc 制約ファイルで与えられている制約をリストします
1.2 Combinational Loop	組み合わせ回路のループがある場合にレポートされます。ない場合はブランクです

1.3 Errors/Warning Messages エラーやウォーニングがある場合のみレポートされます。何もない場合は、このサブ項目はありません

2. CLOCK SUMMARY

クロック・ネットワークごとにサブ項目でリストしてターゲットと結果の周波数（周期）をレポートします。またほかのドメインからの（domain crossing）パスがある場合、クロックエッジ間のタイミングをレポートします

3. TIMING ANALYSIS SUMMARY

3,1 Overall (Setup and Hold) 制約カバレージ、タイミングエラー、およびタイミングスコアです

3.2 Setup Summary Report セットアップ時間をスラックの小さい順にレポートします。パス数はストラテジー項目の ”Number of End Points” に準じます。

3.3 Hold Summary Report ホールド時間をスラックの小さい順にレポートします。パス数はストラテジー項目の ”Number of End Points” に準じます。

3.4 Unconstrained Report 未制約パス（Unconstrained Start/End Points）、および未制約の I/O ポート（Start/End Points Without Timing Constraints）をレポートします。パス数はストラテジー項目の ”Number of Unconstrained Paths” に準じます。

4. DETAILED REPORT

4.1 Setup Detailed Report セットアップ時間詳細を、スラックの小さい順にレポートします。パス数はストラテジー項目の ”Number of Paths Per Constraint” に準じます。

4.2 Hold Detailed Report ホールド時間詳細を、スラックの小さい順にレポートします。パス数はストラテジー項目の ”Number of Paths Per Constraint” に準じます。

5.5 論理合成 HDL アトリビュート記述例

LSE と Synplify Pro でサポートする HDL アトリビュートについて、詳細についてはオンラインヘルプで [Reference Guides] → [Constraints Reference Guides] → [HDL Attributes] よりご参照いただけます。以下ご参考までに、一部について例を示します。

VHDL 例

（定義した信号・ノードを論理圧縮で削除しないで残す）

```
attribute syn_keep       : boolean; -- ワイヤ（ノード）の保持
attribute syn_keep of <signal_name>: signal is true; -- “signal_name” は signal 定義した信号名

attribute syn_preserve   : boolean; -- FF の保持
attribute syn_preserve of <signal_name>: signal is true; -- “signal_name” は signal 定義した信号名

attribute syn_noprune    : boolean; -- 出力がないリソース削除の禁止
attribute syn_noprune of <signal_name>: signal is true; -- “signal_name” は signal 定義した信号名
```

（FSM のコーディング方式）

```
attribute syn_encoding   : boolean; -- [sequential, binary, onehot], 他に safe（”default” 記述有効化）
attribute syn_encoding of <fsm_nodes>: signal is “safe, Boolean”; --“fsm_nodes” は signal 定義した FSM 名
```

Verilog HDL 例

（I/O タイプ指定、ポート宣言行）

```
input CLKIN /* synthesis IO_TYPE="LVCMOS18" */ ;
output CLKOUT /* synthesis IO_TYPE="LVDS25E" */ ;
```

(定義した信号・ノードを論理圧縮で削除しないで残す。ノード定義行)

```
reg [x:0] <FF_name> /* synthesis syn_preserve=1 */;
wire [y:0] <wire_name> /* synthesis syn_keep=1 */;
```

(FSM のコーディング方式。FSM ノード定義行)

```
reg [z:0] <FF_FSM_name> /* synthesis syn_encoding="safe, binary" */;
```

--- *** ---