

第 4 章 モジュール / IP 生成

4.1 概要

Lattice FPGA ファミリー固有のモジュールや Lattice オリジナル IP のパラメータ設定と生成用ツールは、IP Catalog です。カウンターや、汎用的な機能モジュールを実装する場合でも、論理合成ツールに依存する推論 (Inferencing) ベースの手法は、ターゲットフリーという長所はありますが、IP Catalog で生成したモジュールを RTL 記述で明示的にインスタンスした方法のほうが、エリア (リソース使用) やスピード的に少なからず優れる場合が一般的です。

4.2 IP Catalog での作業開始


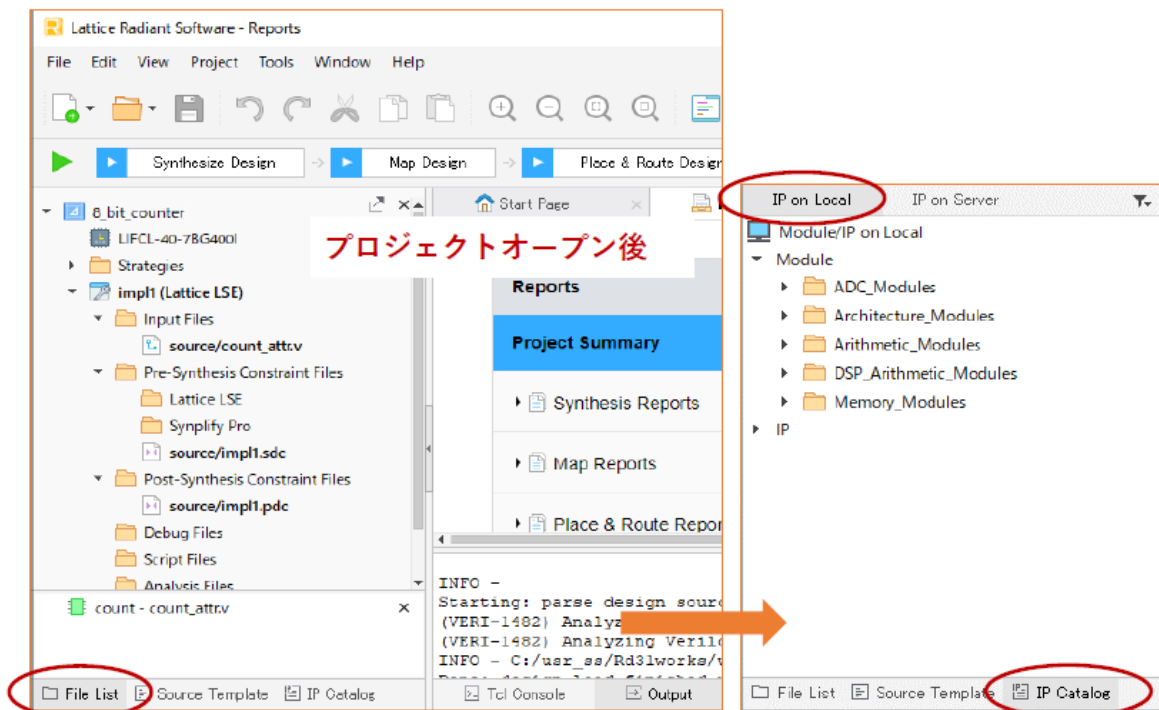
Radiant プロジェクトをオープンすると、GUI 左側は下部で [File List] タブが選択された表示になっています (図 4-1、左)。**[IP Catalog]** タブを選択することで IP Catalog での作業が開始できます (図 4-1、右)。或いはメニューバーから [Tools] → [IP Catalog] を選択するか、アイコン  をクリックしても同様です。

図 4-1. IP Catalog 初期画面例



4.2.1 Module セクション

[IP Catalog] タブでの表示は、ウィンドウ上部の [IP on Local] タブが選択されています。これは Radiant を実行している PC にインストールされている Module と IP を、それぞれサブセクションに分類して表示しています。何も IP をインストールしていない初期状態では、IP セクションにリストされているものはありません。Module セクションの構成は図 4-1 右のようになっていて、五つのサブセクションがあります。

註：本 Lattice Radiant 日本語マニュアルは、日本語による理解のため一助として提供しています。作成にあたっては各トピックについて可能な限り正確を期しておりますが、必ずしも網羅的あるいは最新でない可能性や、オリジナル英語版オンラインヘルプや各種ドキュメントと不一致がある可能性があります。疑義が生じた場合は正規代理店の技術サポートにお問い合わせ頂くか、または最新の英語オリジナル・ソースを参照するようお願い致します。

各サブセクション下の典型的なアイテム（モジュール）は図 4-2 のようになっています。各デバイスファミリー固有の機能ブロックが分類されてリストされます（ファミリー毎に違いがあります）。

Arithmetic_Modules はカウンターや演算機能を LUT+FF で構成する場合の機能ブロックですが、他方 DSP_Arithmetic_Modules は同じ数値演算機能を DSP ハードマクロで実装するものです。Memory_Modules セクションの Distributed_RAM は LUT+FF を用いた構成（分散メモリー）、EBR_Components はブロックメモリー（EBR）を用いた構成です。どちらにも属さない FIFO と FIFO_DC（'Dual-Clock'）および Shift_Register は個々の詳細設定ウィザードで EBR（"Area-Optimized(HW)"）を用いるか、LUT+FF（"Feature-Rich(LUT)"）で実装するかを選択します。

図 4-2. Module 部サブモジュール毎のマクロ項目例

<ul style="list-style-type: none"> ADC_Modules <ul style="list-style-type: none"> ADC_Sequencer 1.4.0 Architecture_Modules <ul style="list-style-type: none"> ADC 1.1.0 I2CFIFO 1.2.0 MPCS 1.3.0 OSC 1.3.0 PLL 1.6.0 SEDC 1.2.0 IO <ul style="list-style-type: none"> GDDR 7:1 1.5.0 DDR_MEM 1.4.0 DDR_Generic 1.5.0 LPDDR4_MEM 1.3.0 MIPLDPHY 1.5.0 SDR 1.2.0 	<ul style="list-style-type: none"> Arithmetic_Modules <ul style="list-style-type: none"> Adder 1.3.0 Adder_Subtractor 1.2.0 Barrel_Shifter 1.2.0 Comparator 1.2.0 Complex_Mult 1.3.0 Convert 1.2.0 Counter 1.1.0 FFT_Butterfly 1.2.0 LFSR 1.2.0 Mult_Accumulate 1.3.0 Mult_Add_Sub 1.3.0 Mult_Add_Sub_Sum 1.3.0 Multiplier 1.3.0 Sin_Cos_Table 1.2.0 Subtractor 1.3.0 	<ul style="list-style-type: none"> DSP_Arithmetic_Modules <ul style="list-style-type: none"> 1D_Filter 1.2.0 Adder_Tree 1.2.0 DSP_Mult_Accumulate 1.2.0 DSP_Mult_Add_Sub 1.2.0 DSP_Mult_Add_Sub_Sum 1.2.0 DSP_Mult_Mult_Accumulate 1.2.0 DSP_Multiplier 1.2.0 	<ul style="list-style-type: none"> Memory_Modules <ul style="list-style-type: none"> FIFO 1.3.0 FIFO_DC 1.3.0 Shift_Register 1.2.0 Distributed_RAM <ul style="list-style-type: none"> Distributed_DPRAM 1.2.0 Distributed_ROM 1.2.0 Distributed_SPRAM 1.2.0 EBR_Components <ul style="list-style-type: none"> RAM_DP 1.3.0 RAM_DP_True 1.2.0 RAM_DC 1.4.0 RAM_ROM 1.3.0 Large_RAM <ul style="list-style-type: none"> Large_RAM_DP 1.2.0 Large_RAM_DP_True 1.2.0 Large_RAM_SP 1.2.0 Large_ROM 1.2.0
---	--	--	--


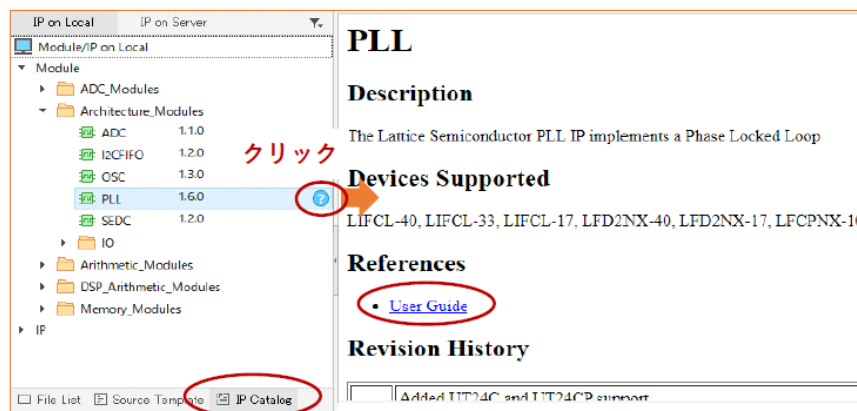
各マクロに関するドキュメントは容易に入手できます。図 4-3 のように、意図する行にマウス・ポインターを移動すると、当該行の右端に  印が現れますので、これをクリックします。すると右側のウィンドウに HTML 表示で選択したマクロの説明が出ますので、その中から「User Guide」をリンクします。自動的に Lattice のウェブサイトからダウンロードされ、保存されます。

図 4-3. マクロのドキュメントを入手する例



4.2.2 IP セクション

Module セクションの下は IP セクションです。ここに IP が表示されるようになり、IP を生成してデザインに実装するためには、Lattice の IP サーバーからユーザーが逐次ダウンロードしてインストールする必要があります。まず [IP Catalog] タブ選択状態で、ウィンドウ上部にある [IP on Server] タブをクリックします。


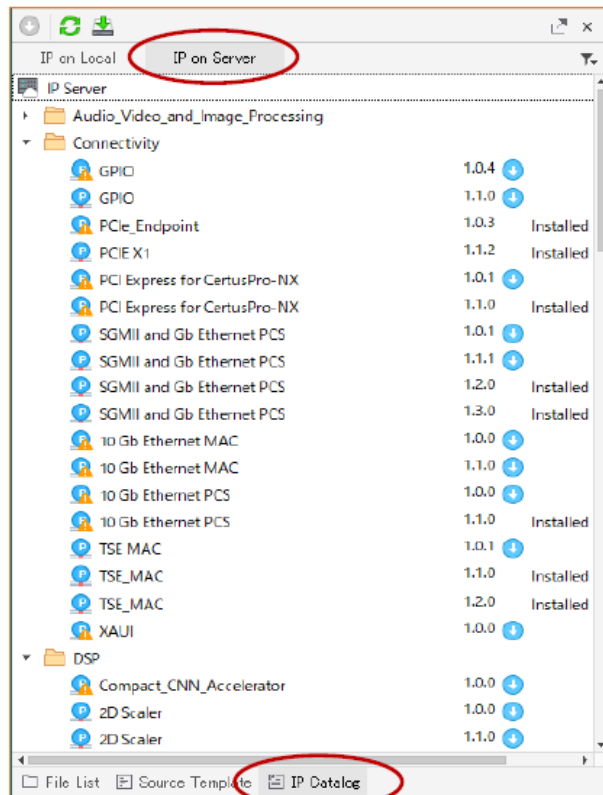
Radiant 起動時に自動的に Lattice の IP サーバーにアクセスしてリストを表示できるようになっています。ネットワーク設定は Radiant の環境設定で行います (第 3.5.5 項参照)。リスト表示を更新したい場合は、上部にあるリフレッシュ・アイコン  をクリックします。

図 4-4. IP リストの表示例





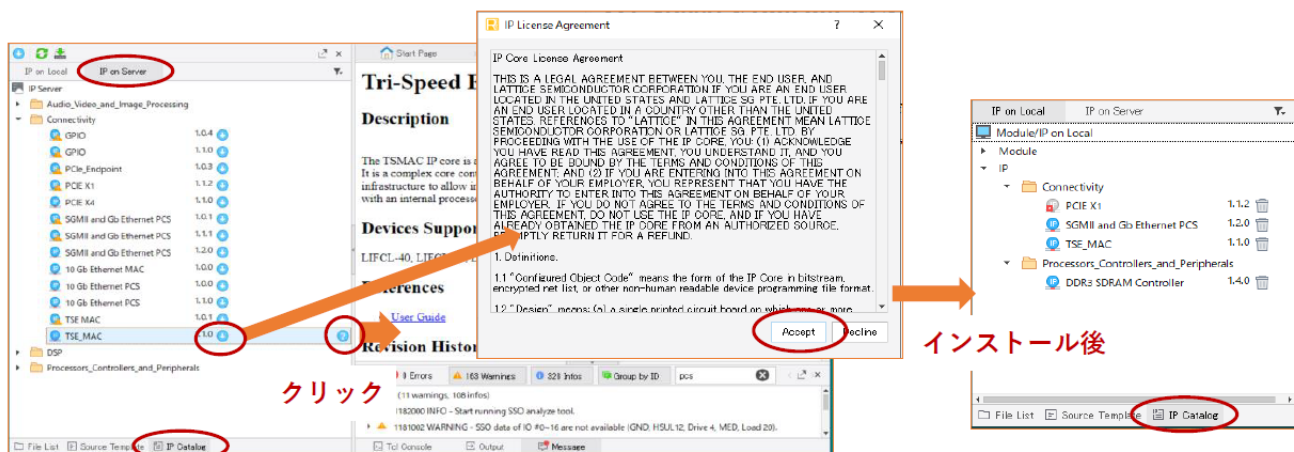
カテゴリ別に IP 名がバージョン情報と共に表示されます。IP をダウンロードするには、所望の IP / バージョン行右側にある  印をクリックします (図 4-5、左)。当該行右端の  印はモジュールと同じです。

図 4-5. IP のインストール



プロジェクトのターゲット・デバイスに対応しない IP は先頭に  アイコンが表示されます。また、古

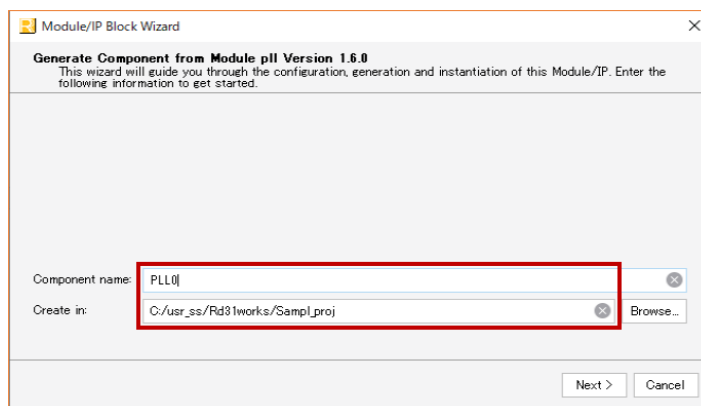
いバージョンは特に理由がない限りは対象としないことを推奨します。

ダウンロードが開始される前にライセンス同意を促すウィンドウが表示されますので (図 4-5、中央)、支障がなければ『Accept』をクリックして先に進みます。問題なくダウンロードとインストールが終了すると、上部の [IP on Local] タブを選択した状態で IP セクション下に IP が追加・表示されます (図 4-5、右)。

4.3 モジュール生成の基本手順

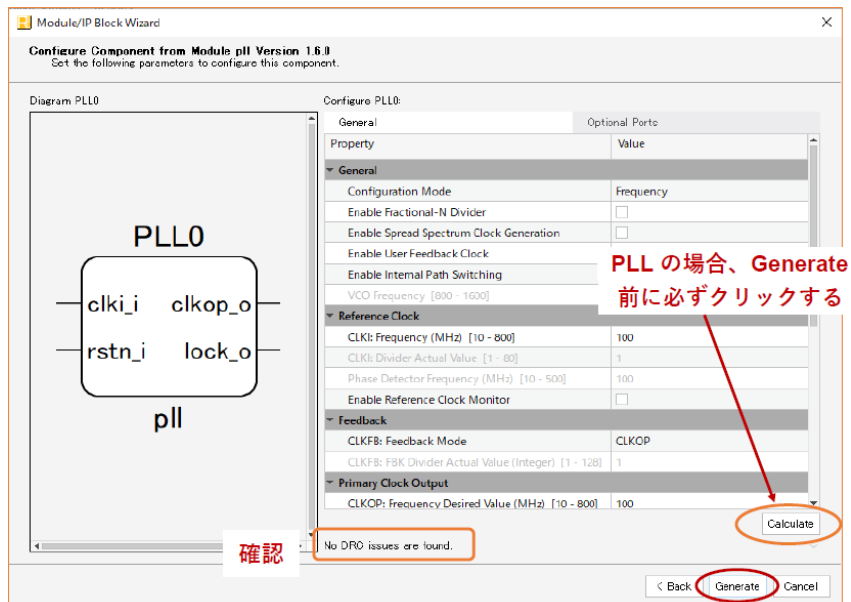
各モジュール共通の手順概要を PLL 生成の例で記述します。Radiant では常に Verilog HDL 記述のモジュールが生成されます。まず IP Catalog 左枠 (図 4-2) の該当するセクションから意図するモジュール (ここでは PLL) をダブルクリックします。

図 4-6. 生成開始ウィンドウ (PLL の場合)



起動するウィザードの最初に表示されるウィンドウ (図 4-6) でマクロ名 (Component name) と生成するフォルダ (Created in) 両セルに入力します。デフォルトではインプリメンテーション・フォルダーが表示されます。変更する場合には必ず Browse ボタンで所望のフォルダーを指定します (存在しないフォルダー名を記述しないようにします)。

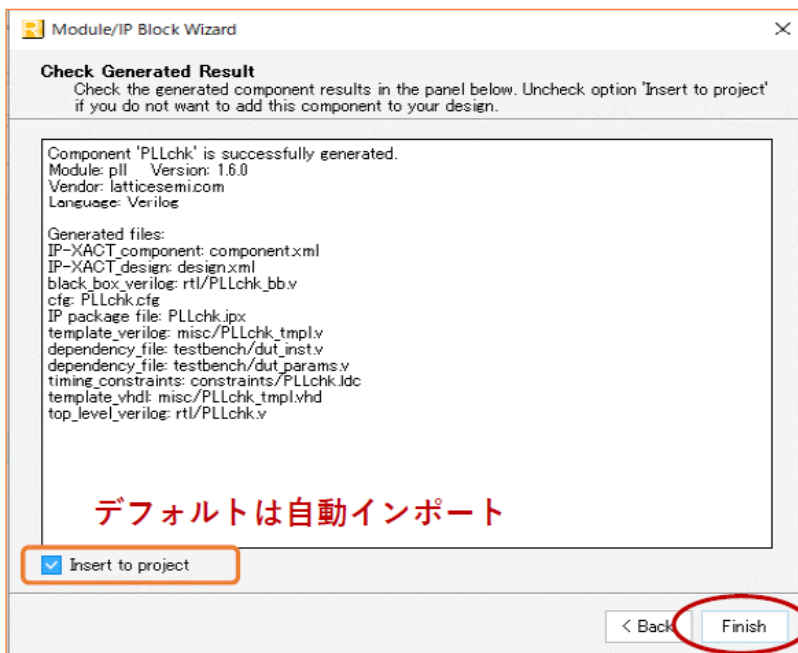
図 4-7. PLL のパラメータ設定ウィンドウ



パラメータ設定は、対象モジュールに拘わらず数値入力セルとプルダウンによるオプションや値の選択で行うようになっています。基本的にツールが期待する全ての設定を行う必要があります。図 4-7 は PLL の例ですが、ウィザード上部に複数のタブがあるモジュールによっては複数のタブがありますので、全タブを確認します。

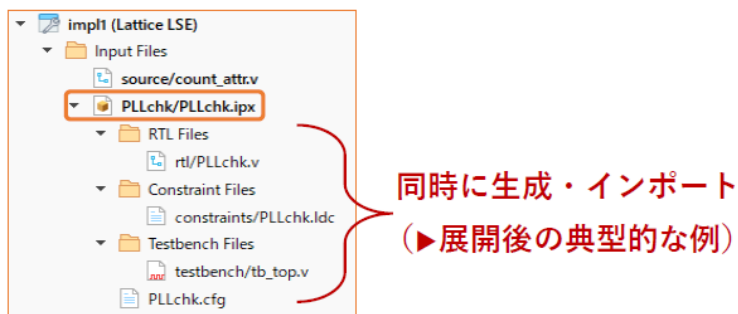
入力・指定完了後、下部の『Generate』ボタンをクリックすることで、モジュールを生成します。PLL に限っては図 4-7 の下部にある『Calculate』ボタンをクリックして DRC がパスしないと『Generate』ボタンがアクティブになりません。他の一部モジュールでもこれに類似したケースがありますので、ご注意ください。ウィンドウ下部に「No DRC Issues are found」と表示されていれば、基本的には問題なく生成が可能です。

図 4-8. Generate 後の表示 (PLL の例)



生成されると次に図 4-8 のようなウィザード表示になります。必ずエラーなどが無いことを確認します。下部左にある「Insert to project」ボタンにデフォルトでチェックが入っていますが、手動でインポートする場合はチェックを外します。『Finish』ボタンで終了します。自動インポート指定の場合は当該インプリメンテーションの”Input Files”セクションに*.ipx、および*.cfgなどの付随する各ファイルがサブフォルダ (RTL Files/Constraint Files/Testbench Files) を含めて追加されます (図 4-9)。

図 4-9. インポートされたモジュール関連のファイル



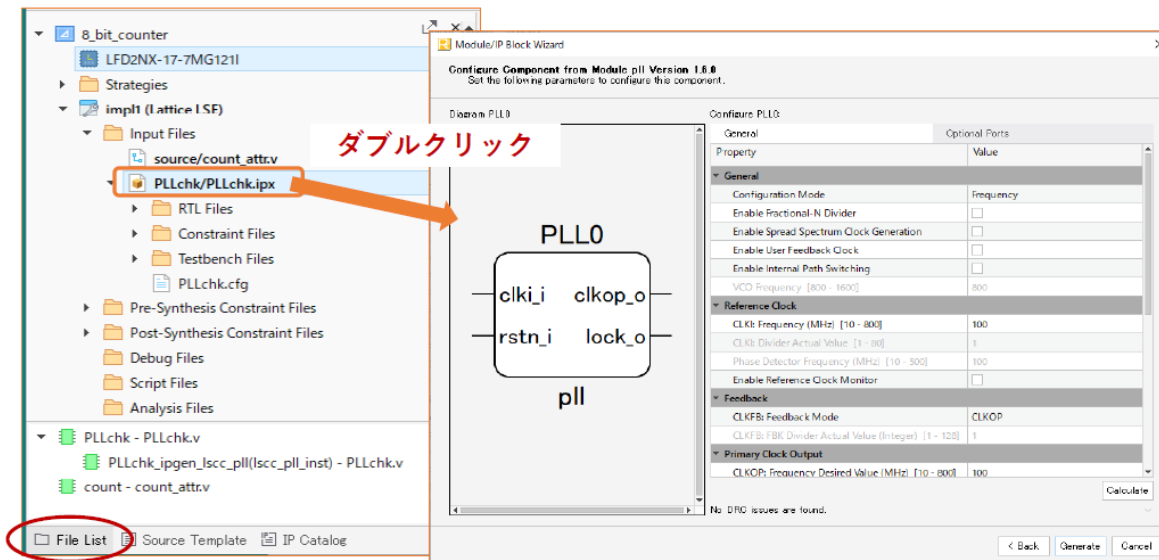
手動でインプリメンテーションにインポートする場合は、この *.ipx ファイルのみを指定します。これ以外を個別に扱うことは避けるようにします。.ipx ファイルはツールが必要なパラメータ情報を全て含む、

Radiant 固有の設定ファイルです。これを指定すると、自動インポートの場合と同様に、付随する他のファイルも全て自動的にインポートされます。

4.4 モジュールのパラメータ変更（再生成）

モジュールのパラメータや設定を変更する場合、インプリメンテーションにインポートされている当該モジュールの *.ipx 行をダブルクリックします（図 4-10）。生成時のパラメータと設定をロードした状態で、図 4-7 と同じウィンドウが表示されます。

図 4-10. パラメータ変更の操作



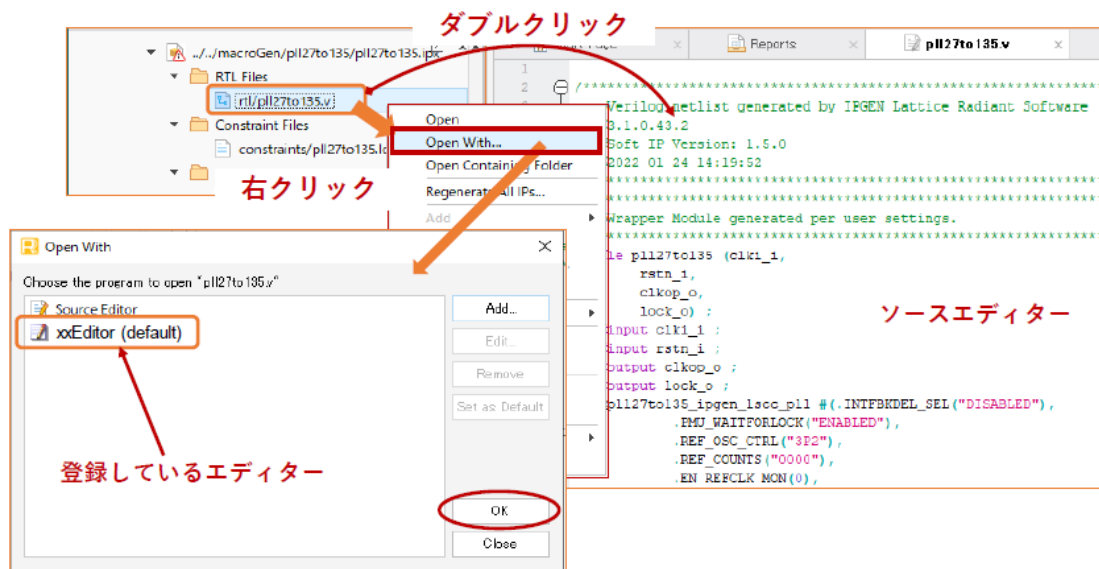
4.5 生成モジュールのソースファイル確認

生成したモジュールやIPのRTLやコンフィグレーションのソース記述を確認したい場合がありますが、その場合はまず [RTL Files]/[Constraint Files]/[Testbench Files] 各セクション行頭の”▶”をクリックして展開します（図 4-11 は PLL の一例）。通常は当該ソースファイルをダブルクリックすることで、Radiant GUI 右側の領域にソースエディターが開きますので、内容を確認できます。

他方、環境設定の第 3.5.3 項に示す方法でユーザー所望のテキスト・エディターを登録していて、それで表示させたい場合は、対象ファイルを選択後に右クリックして [Open With...] を選択します。”Open With”ウィンドウが表示されますので、候補から登録したエディターを選択して『OK』をクリックすることで表示されます。なお、ユーザーの参照のみに用いるなどの場合を除いて、ツールが生成したソースファイルをユーザーが編集して実装に用いることは推奨していないことにご留意ください。

他方、インポートされているユーザー記述 RTL をテキストエディターで確認したい場合は、この二段階の操作は不要です。対象ファイルをダブルクリックすることで、ただちに登録しているエディターが立ち上がります。

図 4-11. エディターによるソースファイル確認



4.6 特定モジュールの生成に関して

本ユーザーガイドでは個々のモジュールについて全てカバーするのではなく、特に基本的な留意事項があるものについてのみ記述しています。それぞれの詳細については、該当するデバイス・ファミリーのテクニカルノートなどをご参照ください。

4.6.1 ROM モジュール

分散メモリ (Distributed) ベースでも EBR ベースでも該当する、ROM モジュールに関する留意事項です。ROM はデバイスのコンフィグレーション時にメモリー内の値が初期化され、リードアクセスのみが許容されるモジュールです。従って他とは異なり、モジュール生成時に初期化データを与える必要があります。

図 4-12. ROM モジュールの初期化指定部

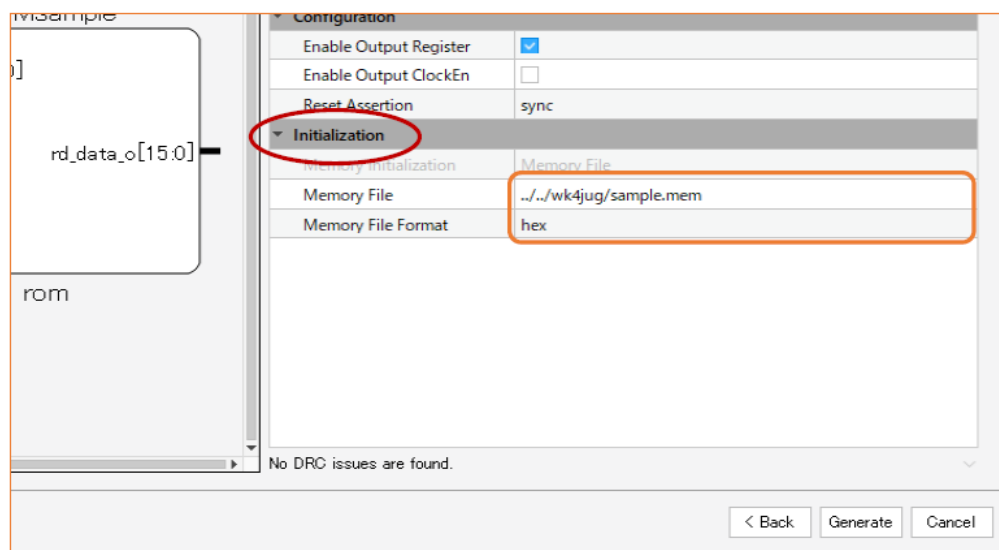


図 4-12 は設定ウィザード画面の右下部を示します。"Initialization" セクションで指定する初期化ファイル

は「拡張子を *.mem とするテキストファイル」という規定があります。「Memory File」セルに、ブラウズして mem ファイルを選択・指定します。

テキストファイルの初期値を記述するフォーマットも決められていて、Binary (0 と 1 のみの表記によるテキスト) か Hex (16 進表記)、または Addressed Hex (アドレス指定あり 16 進表記) のいずれかである必要があります。それぞれのフォーマットは、図 4-13 に示すような形式です。詳細は各デバイス・ファミリーのメモリー関連テクニカルノートをご参照ください。


図 4-13. 初期化ファイル *.mem のフォーマット (左 : Binary、中 : Hex、右 : Addressed Hex)

0010000011001110	A001	A0 : 03 F3 3E 4F
0101110011100010	09EB	B2 : 3B 9F
1110000011111000	3725	B3 : 2A A5 34 22
0010000101000111	BC0E	B4 : 11 77 FF A0
0000110011001110	51FF	C0 : 0F 55

初期化ファイルを指定後に『Generate』で ROM モジュールを生成しますが、パラメータ設定より *.mem ファイルが大きい (深い) と、生成ステータス表示ウィンドウで結果がエラーになります。そのまま終了してもモジュールは生成されません。設定より *.mem ファイルが小さい (浅い) と、エラーにはならず、不足分は全て "0" とした ROM モジュールが生成されます。特に ROM モジュールの場合は『Generate』をクリックした後は、図 4-8 のような結果表示を必ず確認するようにします。

4.7 IP のパラメータ設定と生成

インストールした IP のパラメータ指定・設定と生成手順はモジュール生成と基本的に同様です。設定ウィザードのタブやウィンドウ内の構成や個々の IP に依存しますので、それぞれのユーザーガイドをご確認ください。

ユーザーガイドの入手方法は、図 4-5 の場合と同様に、意図する行にマウス・ポインターを移動すると、当該行の右端に  印が現れます。これをクリックすると、右側に IP についての説明が表示されます。リンクが張られている "User Guide" をクリックするとダウンロードして自動的に保存されます。
