

ModelSim

アルテラ・シミュレーション・ライブラリ 作成および登録方法

ver.10

ModelSim – アルテラ・シミュレーション・ライブラリ作成および登録方法

目次

1. はじめに.....	3
2. 操作方法.....	6
2-1.Quartus II におけるシミュレーション・ライブラリの作成	6
2-2.ライブラリの登録	10
2-3.ライブラリの選択	14
3. NativeLink での活用方法	15
Appendix: ライブラリの作成および登録(上級者向け).....	17
Type A: プロジェクト共通でライブラリを作成.....	17
Type B: プロジェクトごとにライブラリを作成	21

1. はじめに

この資料では、メンター・グラフィックス社のシミュレーション検証ソフトウェア ModelSim[®] SE/PE/DE（以下、ModelSim と表記）において、アルテラ・デバイスをターゲットにした際のシミュレーションに必要なライブラリの作成および登録方法をご紹介します。

本資料では、下記バージョンを対象として作成しています。

Quartus[®] II v12.1 SP1

ModelSim SE 10.1d

ModelSim において、アルテラ・メガファンクションを使用したデザインのファンクション・シミュレーションや、配置配線後にタイミング・シミュレーションを実行する場合は、アルテラ・シミュレーション・ライブラリが必要なため、従来、ユーザが自ら必要なライブラリのラインナップやモデルファイルを調べ、それらをコンパイルし、ライブラリを作成する必要がありました。

ModelSim でアルテラ・デバイスを対象としたシミュレーションを実行する場合、シミュレーション・ライブラリの作成方法から 3 つに分類できます。

- A. Quartus II NativeLink を使用
- B. Quartus II でシミュレーション用ライブラリを作成し、ModelSim にてシミュレーションを実行
- C. ModelSim でシミュレーション用ライブラリを作成し、シミュレーションを実行

ライブラリの作成からシミュレーションまでのフローを大枠に図 1-1 に記載します。

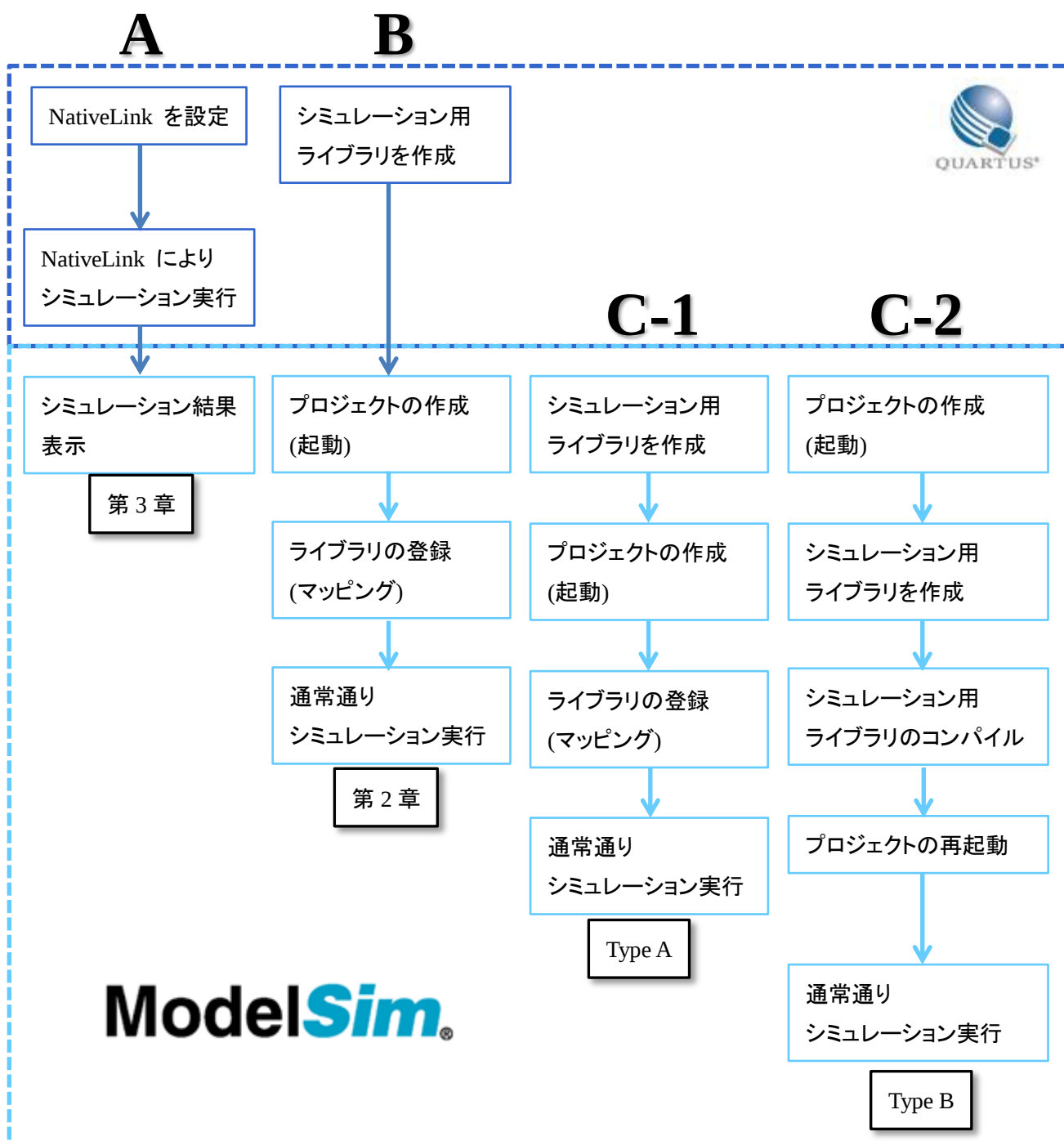


図 1-1 シミュレーション・ライブラリ作成と ModelSim におけるシミュレーション・フロー・イメージ

アルテラ・シミュレーション・ライブラリの作成など煩わしい作業を簡易的にする機能が Quartus II にあります。Quartus II の“EDA Simulation Library Compiler” 機能です。

EDA Simulation Library Compiler は EDA 言語シミュレータ・ツールのための機能で、アルテラ・シミュレーション・ライブラリをコンパイルするためのツールです。EDA Simulation Library Compiler は、ターゲット・デバイスを指定するだけでユーザが必要なファイルをコンパイルし、シミュレーション・ライブラリを生成してくれます。そのためユーザは ModelSim において、ライブラリの登録（マッピング）をするだけとなり、作業が簡素化されます。また必要なライブラリを調べることもないため、アルテラ・デバイスをはじめて使用するユーザにも簡単にシミュレーションを実行していただけます。この機能は Quartus II で NativeLink シミュレーションを行う場合にも活用できます。

なお EDA Simulation Library Compiler をサポートしている EDA シミュレータ・ツールは ModelSim の他に、Questa[®]、Active-HDL[™]、Riviera-PRO[™] があります。

Quartus II において、EDA Simulation Library Compiler を使用せずに ModelSim でライブラリを作成および登録する方法を希望される場合には、本資料の『Appendix: ライブラリの作成および登録(上級者向け)』をご覧ください。

2. 操作方法

Quartus II にてアルテラ・シミュレーション・ライブラリを作成し、ModelSim へ登録します。

2-1. Quartus II におけるシミュレーション・ライブラリの作成

Quartus II の EDA Simulation Library Compiler を使用し、シミュレーション・ライブラリを作成します。

- ① Quartus II を起動します。
- ② Tools メニュー ⇒ Launch Simulation Library Compiler を選択します。
- ③ EDA Simulation Library Compiler が起動します。(図 2-1-1)

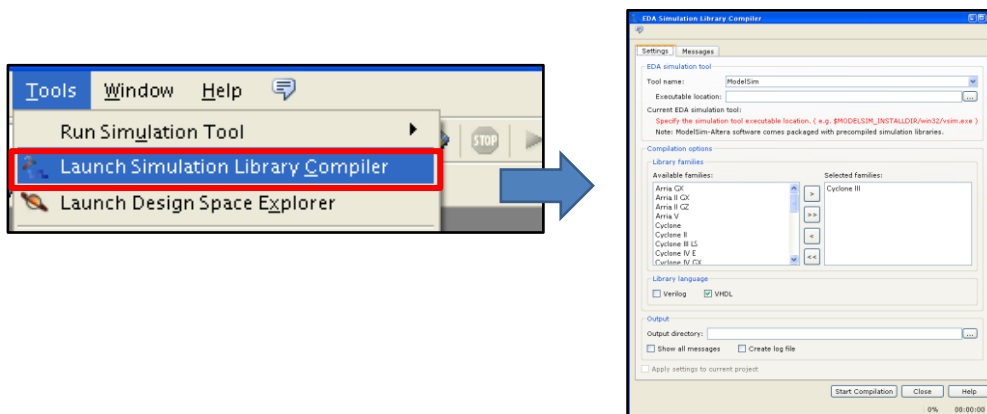


図 2-1-1 EDA Simulation Library Compiler の起動

- ④ Settings タブ ⇒ EDA simulation tool セクション ⇒ “Tool name” において、プルダウンリストから ModelSim を選択します。(図 2-1-2)

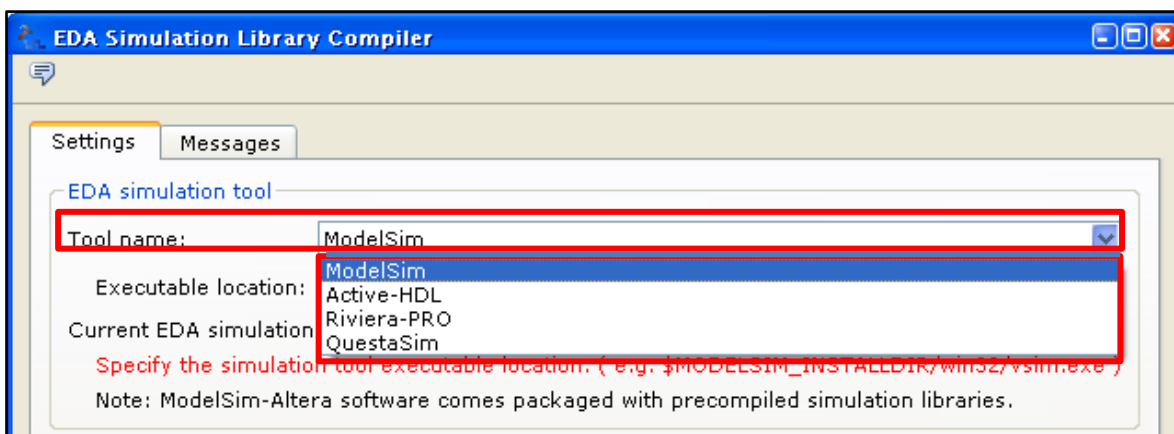


図 2-1-2 Tool name

- ⑤ “Executable location” で、ModelSim の起動プログラム・ファイル (modelsim.exe) が保存されているフォルダを指定します。(ブラウザボタンを使用してください。)(図 2-1-3)

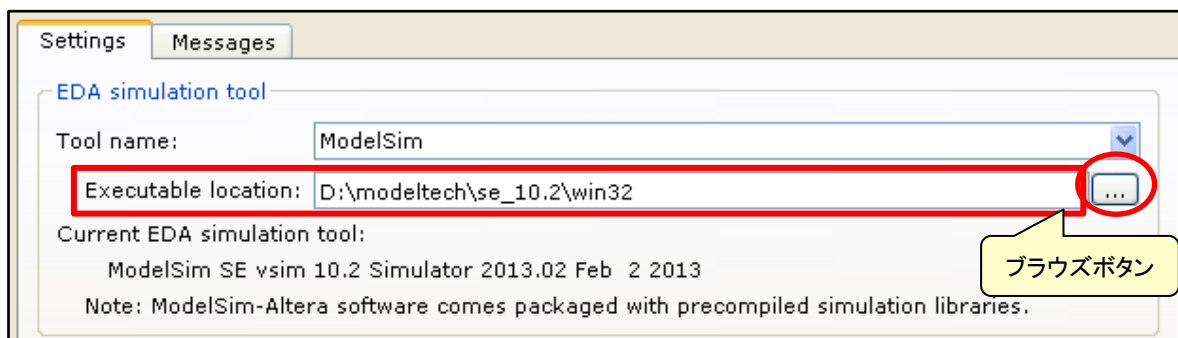


図 2-1-3 Executable location

- ⑥ Compilation options セクション ⇒ Library families セクション ⇒ “Available families” のリストより、ターゲット・デバイス・ファミリを選択し、> ボタンをクリックして “Selected families” へ移行します。(図 2-1-4)

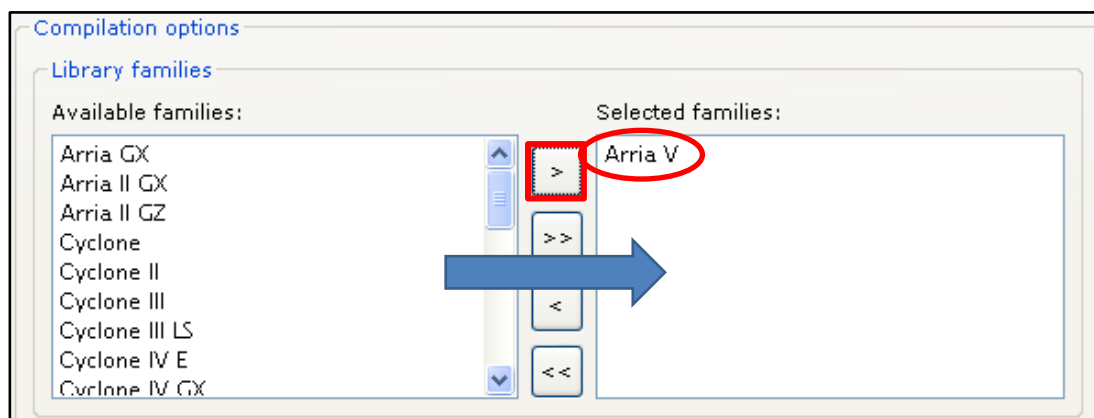


図 2-1-4 Selected families

<<補足>>

Selected families 欄に複数のデバイスを選択することも可能です。指定した分のデバイス用ライブラリが一括で生成されます。ただし、生成されたライブラリの格納先は共通であるため、ライブラリが混在し重複するライブラリは上書きされます(図 2-1-5 参照)。そのため、デバイス・ファミリごとに管理できるよう、デバイス・ファミリ単位でシミュレーション・ライブラリを生成することを推奨します。

参考図: Arria® V と Stratix® IV 用ライブラリを同時に生成した場合のフォルダ

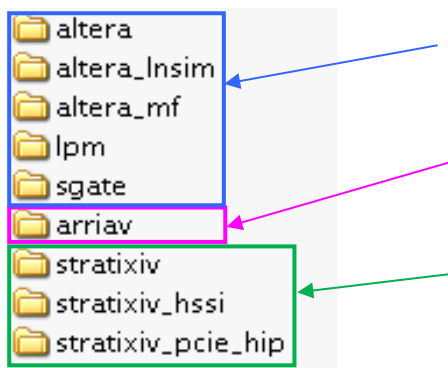


図 2-1-5 ライブラリ生成時のフォルダ・イメージ

- ⑦ Compilation options セクション ⇒ Library families セクション ⇒ “Library language” より、生成するライブラリの言語を指定します。(図 2-1-6)

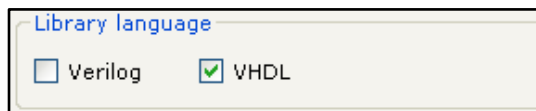


図 2-1-6 Library Language

- ⑧ Output セクション ⇒ “Output directory” に、ライブラリの保存先を指定します。(図 2-1-7)

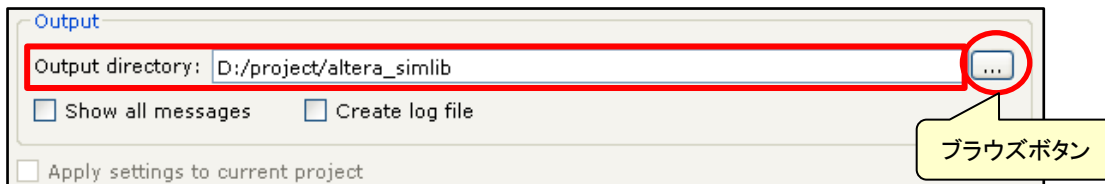


図 2-1-7 Output directory

- ⑨ その他、必要に応じてオプションを設定します。
⑩ Start Compilation ボタンをクリックし、ライブラリを生成します。(図 2-1-8)

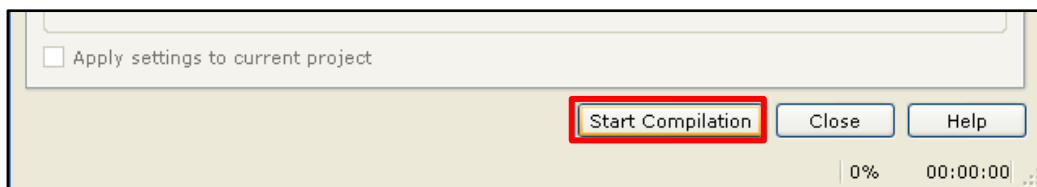


図 2-1-8 Start Compilation

- ⑪ バックグラウンドで ModelSim によりライブラリのコンパイルが実行されます。
⑫ 無事に終了すると、“Compilation was successful” のメッセージが表示されます。

- ⑬ 指定フォルダに“指定言語_libs”フォルダが生成され、指定デバイス用のアルテラ・シミュレーション・ライブラリが生成されます。
- ⑭ Close ボタンをクリックし、Quartus II によるアルテラ・シミュレーション・ライブラリの作成は終了です。

なお、このフォルダ内には、ファンクション・シミュレーションで使用するライブラリとタイミング・シミュレーションで使用するライブラリそれぞれが混在しています。

2-2. ライブラリの登録

第 2-1 章『Quartus II におけるシミュレーション・ライブラリの作成』で作成したライブラリを ModelSim で登録(マッピング)します。マッピング作業は、ModelSim のプロジェクトを作成した後に行います。

- ① ModelSim を起動しプロジェクトを作成します。プロジェクトが既存にある場合は、プロジェクトを起動します。

※ プロジェクト作成の操作方法は、担当する代理店の技術サイトにおいて以下の資料をご参考ください。

『ModelSim-Altera Edition RTL シミュレーション』
または『ModelSim-Altera Edition タイミング・シミュレーション』

- ② File メニュー ⇒ New ⇒ Library より、Create a New Library ダイアログボックスを表示します。
- ③ 作成済みのライブラリにそれぞれライブラリ名を付けます。(図 2-2-1)

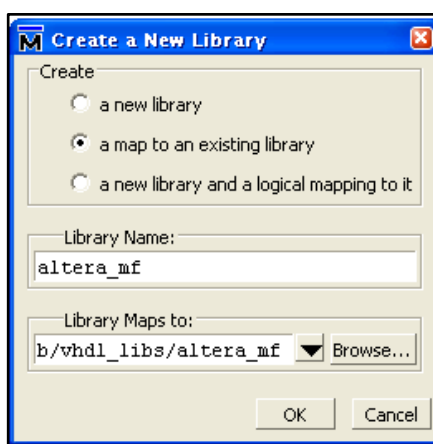


図 2-2-1 Create a New Library

- **Create:** “a map to an existing library”を選択します。(既存ライブラリに対してマッピングする。)
- **Library Name:** ライブラリ名を入力します。

ライブラリの名称を何にするかは、開発言語が VHDL / Verilog HDL のどちらか、またこれから検証するシミュレーションがファンクションなのかタイミングなのかにより異なります。

➤ VHDL

- ファンクション・シミュレーションの場合

MegaWizard Plug-In Manager の EDA タブでアナウンスされたライブラリ名

- タイミング・シミュレーションの場合

VHO に宣言されているライブラリ名

➤ Verilog HDL

- 任意のライブラリ名でマッピング(推奨は VHDL と同様)
- マッピングしなくてもライブラリ指定はできます。

※ライブラリ名に関する詳細は、次頁の補足をご参考ください。

- **Library Maps to:** 先ほど作成したライブラリのフォルダをブラウズボタンにより指定します。

ライブラリに対する作業は以上です。ライブラリ登録完了後は、ModelSim において通常のシミュレーショ

ン・フローに従ってください。

《補足:ライブラリ名に関する詳細事項》

Create a New Library ダイアログボックスの Library Name にどのようなライブラリ名をつければ良いかは、開発言語により異なります。また、これから検証するシミュレーションがファンクションなのかタイミングなのかによっても異なります。以下を参考にし、適切なライブラリ名を付けてください。

■ VHDL の場合

- ファンクション・シミュレーションの場合

ユーザ・デザイン内に存在するアルテラ・メガファンクション (Quartus II の MegaWizard Plug-In Manager で作成したファンクション) を作成する際に、MegaWizard Plug-In Manager の EDA タブでアナウンスされたライブラリ名を用います。(図 2-2-2)

メガファンクションごとにライブラリが異なりますので、それぞれにライブラリのマッピングを行います。

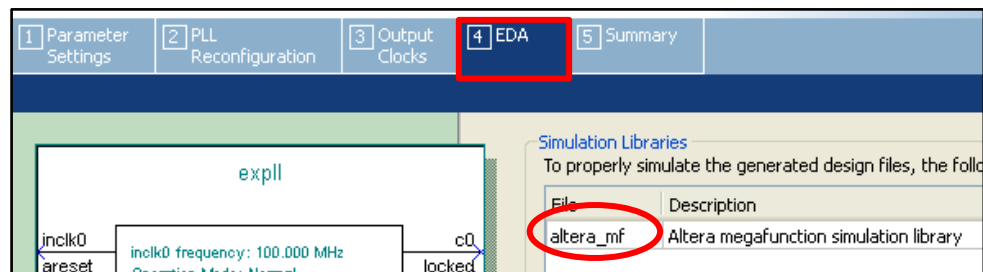
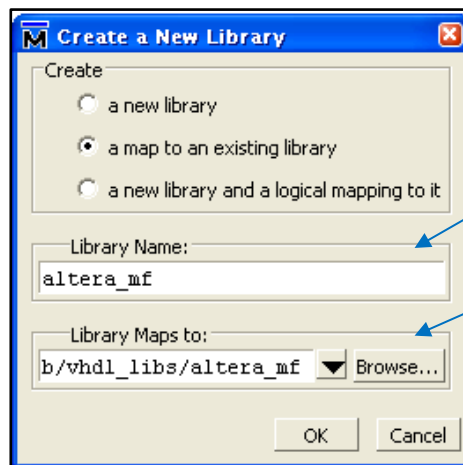


図 2-2-2 Simulation Libraries

上記例の場合は、ModelSim でのライブラリ名とフォルダ指定は以下のようになります。(図 2-2-3)



同じ名称を入力

同じ名称のフォルダを指定
(ブラウズボタンの使用必須)

図 2-2-3 Create a New Library

– タイミング・シミュレーションの場合

Quartus II のコンパイルで生成されたタイミング・シミュレーション用ネットリストファイル VHO に宣言されているライブラリ名を用います。主にデバイス・ファミリ名ですが、ファミリによりライブラリが複数必要な場合もありますので、ご注意ください。

```
LIBRARY ALTERA;  
LIBRARY ARRIAII;  
LIBRARY IEEE;  
USE ALTERA.ALTERA_PRIMITIVES_COMPONENTS.ALL;  
USE ARRIAII.ARRIAII_COMPONENTS.ALL;  
USE IEEE.STD_LOGIC_1164.ALL;
```

ENTITY (以下省略)

(ターゲット・デバイスが Arria II GX の場合)

上記例の場合は、ModelSim でのライブラリ名とフォルダ指定は 2 種類作成することになり、それぞれ以下ようになります。(図 2-2-4)

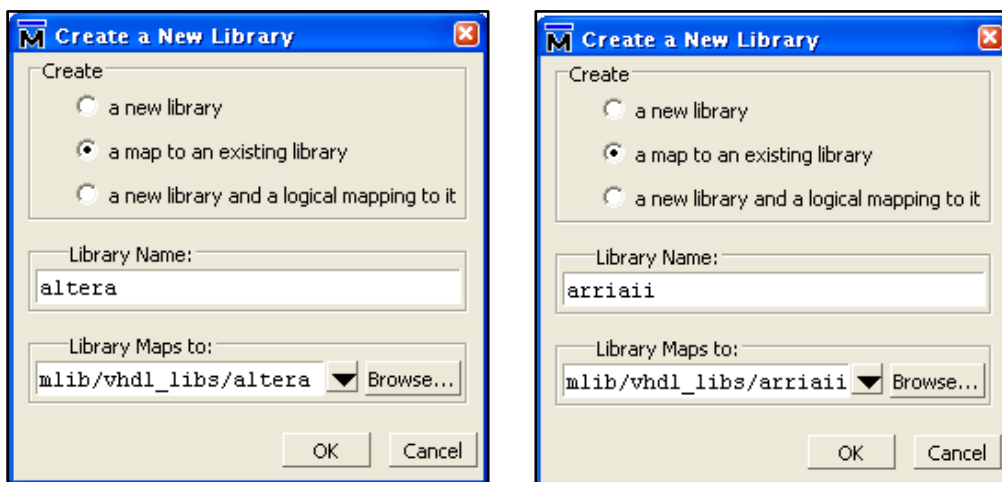


図 2-2-4 Create a New Library

■ Verilog HDL の場合

Verilog HDL の場合ライブラリをマッピングせずにシミュレーションが行えますが、操作を簡素化するため、マッピングすることをお勧めします。以下、操作手順です。

– ファンクション・シミュレーションの場合

Verilog HDL の場合にはライブラリという概念がないため、任意のライブラリ名を付けてマッピングすることが可能です。また、Library Maps to で指定するフォルダは VHDL と同様に Quartus II で生成したライブラリのフォルダを指定します。

ただし混乱をさけるため、弊社では VHDL と同様のフローを推奨しています。

– タイミング・シミュレーションの場合

上記ファンクション・シミュレーション同様、任意のライブラリ名を付けることが可能です。ただ、Verilog HDL では Quartus II のコンパイルで生成されたタイミング・シミュレーション用ネットリストファイル VO にライブラリ名が明記されないため、どのライブラリが必要なのかわかりません。そこで、サンプルとして VHDL 用のタイミング・シミュレーション・ネットリストファイル (VHO) を生成させ、ライブラリ宣言部分を参考にされると良いでしょう。

VHO の生成は、Quartus II においてコンパイルが完了していれば容易に行えます。操作は以下のとおりです。

- ① Quartus II の Assignments メニュー ⇒ Settings ⇒ EDA Tool Settings ⇒ Simulation において、出力フォーマットを一時的に VHDL に変更します。このとき、出力先のフォルダは、本来必要なネットリストファイルと区別するため、下図のように別フォルダを用意することをお勧めします。(図 2-2-5)

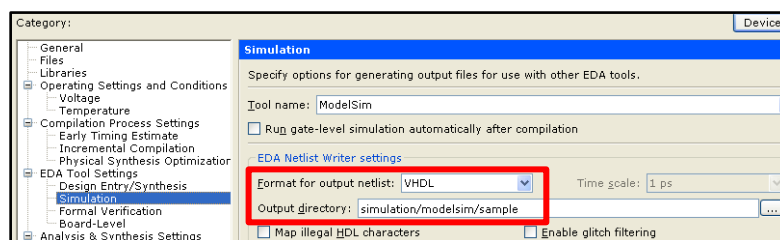


図 2-2-5 Simulation

- ② ネットリストファイルのみを生成させるため、Processing メニュー ⇒ Start ⇒ Start EDA Netlist Writer を実行します。
- ③ 指定フォルダに生成された VHO を開き、“LIBRARY” 宣言しているもののうち、IEEE 以外をメモに取ります。これらがライブラリ名に当たります。
- ④ ModelSim でタイミング・シミュレーション・ライブラリをマッピングします。
 - ・Library Name → 上記 3) でメモした名称を入力
 - ・Library Maps to → Quartus II で作成した同一名のライブラリ用フォルダを指定

2-3. ライブラリの選択

ModelSim においてライブラリ登録が完了すれば、あとは通常操作となりますが、Verilog HDL の場合のみデザインのロード実行前にライブラリの選択が必要です。

■ VHDL

- 特別な作業は必要なし。
- アルテラ用ライブラリ作成 & マッピングが完了していれば、MegaWizard Plug-In Manager で作成した VHDL ファイル、あるいは配置配線後のシミュレーション用ネットリスト VHO をコンパイル可能です。
 - 生成された VHDL、VHO には、アルテラ用のライブラリ/パッケージ宣言が記述済みであるため

■ Verilog HDL

- デザインのロード前に、作成したアルテラ・シミュレーション用ライブラリを選択
 - ① Simulate メニュー ⇒ Start Simulation ⇒ Libraries タブを選択します。
 - ② Add ボタン ⇒ Select Library においてライブラリを指定します。その際、事前にライブラリがマッピングされていれば、▲ボタンをクリックしてプルダウンリストから作成したライブラリ名を選択してください。(図 2-3-1)

もし、ライブラリをマッピングしていない場合には、Browse ボタンより、Quartus II で作成したライブラリ群のうち、適切な名称のフォルダを指定してください。

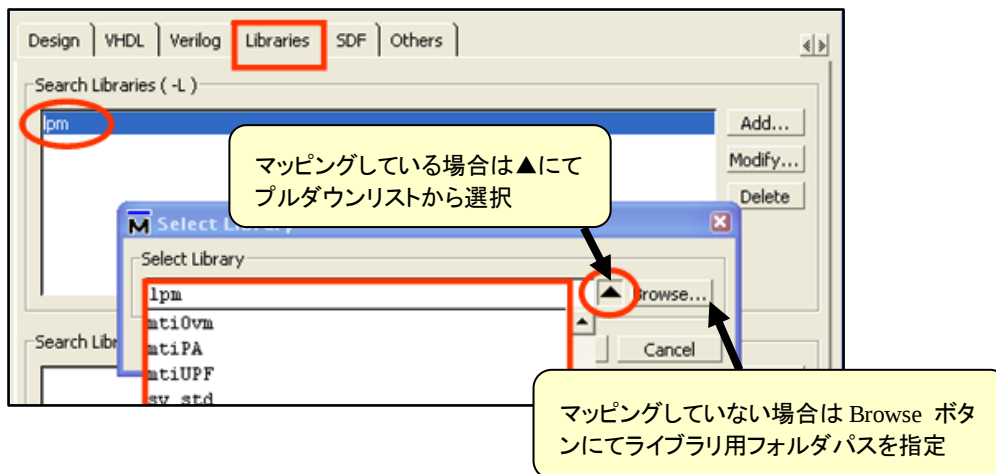


図 2-3-1 Libraries

3. NativeLink での活用方法

NativeLink 機能とは、Quartus II 操作フローに EDA 論理合成ツールや EDA シミュレータ・ツールの実行を統合させたツール・インタフェースのことです。従来、アルテラ・シミュレーション・ライブラリを NativeLink 用に設定するには、do ファイルや tcl ファイルなどのスクリプト・ファイルに記載するか、あらかじめ作成した ModelSim のプロジェクト・ファイルへ記述する必要がありました。

そのようなとき Quartus II の EDA Simulation Library Compiler を活用すると、NativeLink 経由での ModelSim シミュレーションの際、GUI でアルテラ・シミュレーション・ライブラリを指定することができ、さらに容易に操作ができます。

NativeLink の操作に関しては、担当する代理店の技術サイトにおいて以下の資料をご参考ください。

『Quartus II はじめてガイド – EDA ツールの設定』(こちらの“第4章 NativeLink の設定”をご覧ください。)

NativeLink で ModelSim シミュレーションをする際に、Quartus II の EDA Simulation Library Compiler で作成したライブラリを指定する方法は、以下のとおりです。

- ① Quartus II Assignments メニュー ⇒ Settings ⇒ EDA Tool Settings ⇒ Simulation を選択します。
- ② Tool name のプルダウンリストより ModelSim を選択し、Run gate-level simulation automatically after compilation にチェックを入れます。(図 3-1)

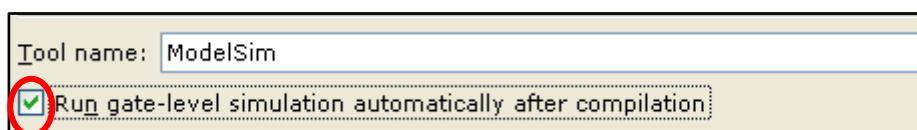


図 3-1 Simulation

- ③ 必要に応じ、各種オプションを設定します。
- ④ NativeLink settings セクションから実行したいフローを選択します。
- ⑤ More NativeLink Settings ボタンをクリックします。Existing option settings より “Location of user compiled simulation library” オプションを選択し、Setting 欄をダブルクリックします。あらかじめ EDA Simulation Library Compiler で作成したライブラリが保存されているフォルダ(“指定言語_libs” フォルダが保存されているフォルダ)を指定します。(図 3-2)

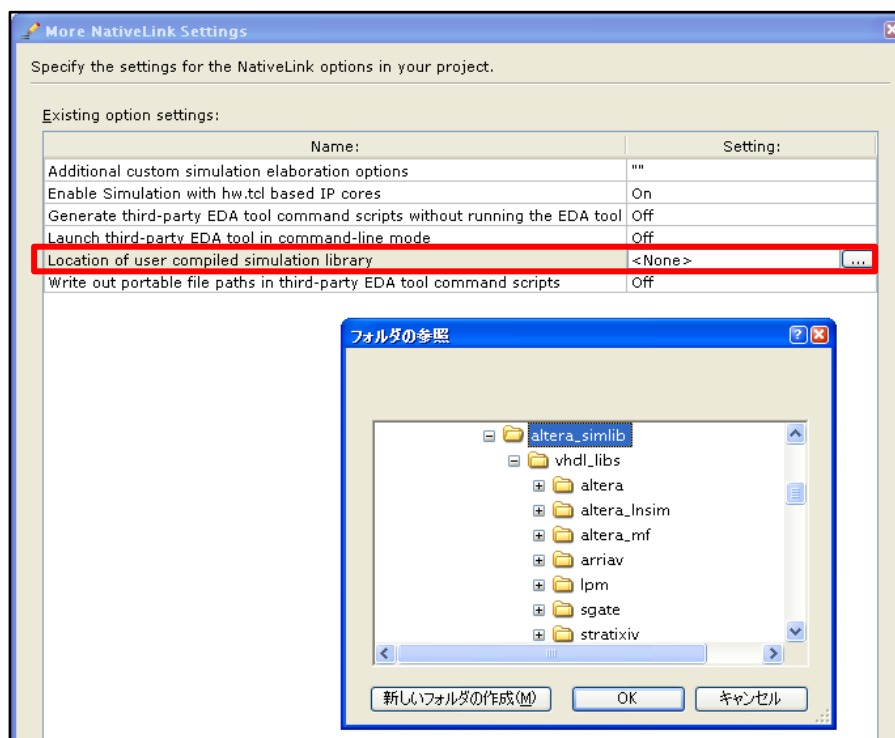


図 3-2 More NativeLink Settings

以上で設定は終了です。その後は通常の NativeLink シミュレーションを実行してください。

《実行例》

- Quartus II のコンパイル・フローの一環として ModelSim でシミュレーションする場合
 - Processing メニュー ⇒ Start Compilation
 - ※ この場合、ファンクション・シミュレーション は実行されません。
- NativeLink シミュレーションを単独で実行する場合 (図 3-3)
 - Tools メニュー ⇒ Run Simulation Tool ⇒ RTL Simulation

または

- Tools メニュー ⇒ Run Simulation Tool ⇒ Gate Level Simulation...

※ RTL Simulation を実行の場合には、Quartus II において、Analysis & Elaboration が終了している必要があります。

※ Gate Level Simulation を実行する場合には、配置配線、タイミング検証まで終了している必要があります。

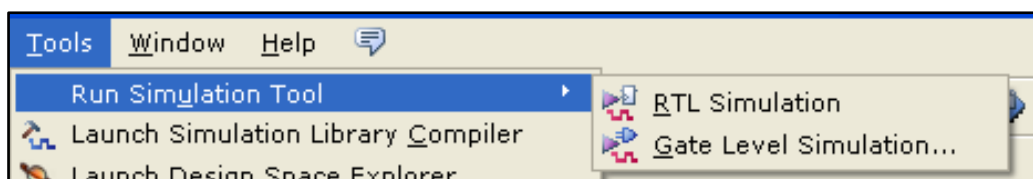


図 3-3 Run Simulation Tool

Appendix: ライブラリの作成および登録(上級者向け)

本資料では、ModelSim で必要なライブラリの作成を Quartus II の EDA Simulation Library Compiler を使用したフローでご紹介しました。Appendix では、ModelSim でライブラリを作成および登録するフローをご紹介します。なお、このフローは ModelSim の操作に慣れているユーザ向けであるため、弊社では本編でご案内している EDA Simulation Library Compiler を活用した方法を推奨いたします。

ModelSim でライブラリを作成および登録する際、2 つのタイプで構成できます。ユーザの作業プロジェクトに合ったタイプでご利用ください。

Type A: プロジェクト共通でライブラリを作成し、プロジェクトごとにライブラリ登録を行う。

Type B: プロジェクトごとにライブラリを作成し、プロジェクトごとにライブラリ登録を行う。

各タイプの操作方法は、以下にご紹介します。なお、各タイプともライブラリの作成方法のみをご案内し、共通の操作である“プロジェクトごとにライブラリ登録を行う”は、本資料 2-2 項“ライブラリの登録”をご参照ください。

Type A: プロジェクト共通でライブラリを作成

この方法は、使用するコンピュータのあるフォルダにライブラリを作成し、各プロジェクトではライブラリ登録のみを行うため、複数プロジェクトでライブラリを共有することが可能です。つまり、ライブラリの複製がないので、コンピュータのファイルサイズを抑えることができます。

① プロジェクトを作成(または起動)していない状態で、File メニュー ⇒ Change Directory を選択し、ライブラリを作成するフォルダを指定します。

② ライブラリ用フォルダを作成するため、File メニュー ⇒ New ⇒ Library を選択します。(図 A-1)

- ・ Create: a new library を選択
- ・ Library Physical Name: ライブラリを保存するフォルダ名を入力

OK ボタンをクリックすると、指定先フォルダに指定した名称のフォルダが生成されます。複数ライブラリを作成する場合には、この操作を繰り返してください。

なお作成するライブラリのフォルダ名は任意ですが、推奨はアルテラ・シミュレーション・ライブラリ名です。詳細は、本資料 7 ページ「補足:ライブラリ名に関する詳細事項」をご参考ください。



図 A-1 Create a New Library

- ③ Compile メニュー ⇒ Compile を選択し、Compile Source Files ダイアログボックスを表示します。
- ④ “Library” のプルダウンリストから、先ほど作成したライブラリ名を選択します。(図 A-2)

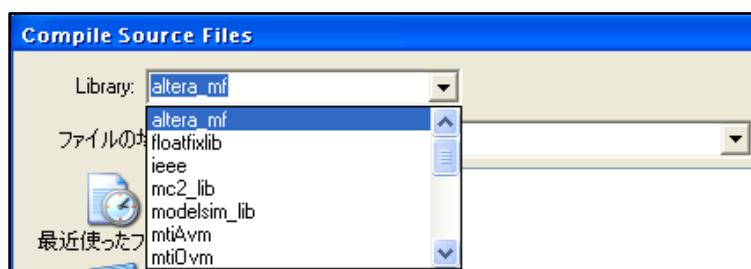


図 A-2 Compile Source Files

- ⑤ “ファイルの場所”では、④で指定したライブラリに該当するアルテラのシミュレーション用ライブラリのモデルファイルを指定します。アルテラのシミュレーション用ライブラリのモデルファイルの保存場所は、Quartus II のインストール・フォルダ(以下参照)にあります。(図 A-3)

¥¥Quartus II インストール・フォルダ¥eda¥sim_lib

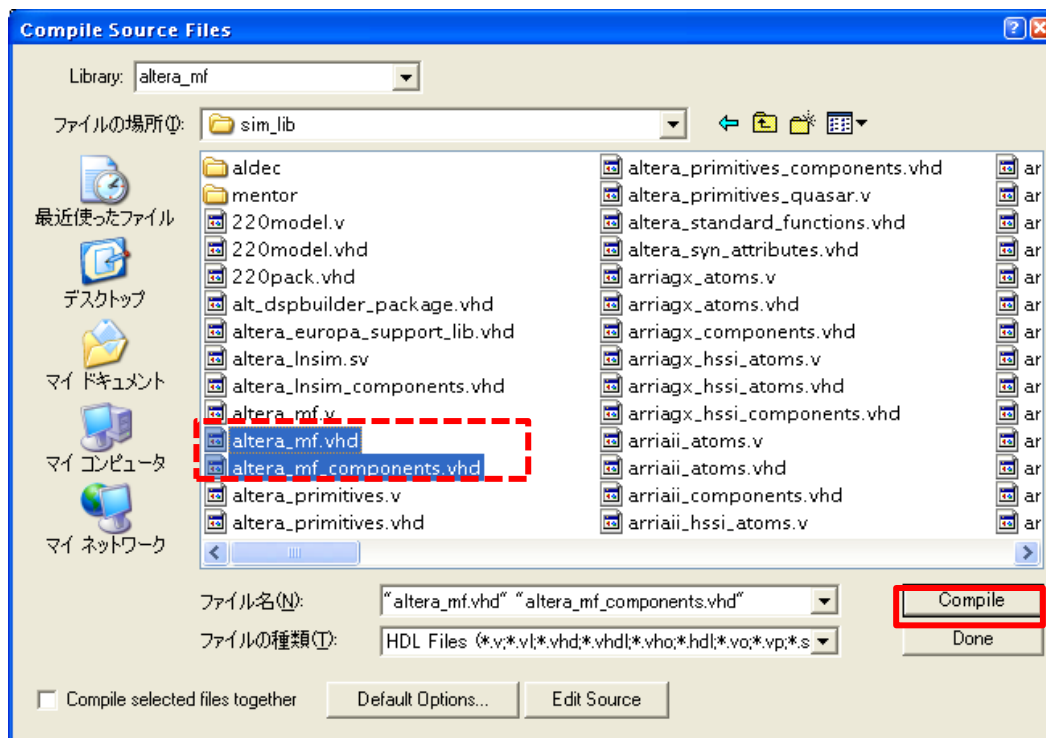


図 A-3 Compile Source Files

sim_lib フォルダ内から目的のモデルファイルを指定します。その後 Compile ボタンをクリックし、コンパイルを実行します。コンパイルが完了すると、Library で指定したフォルダに指定したモデルファイルのコンパイル情報が保存されます。

sim_lib フォルダには、VHDL 用モデルファイルと Verilog HDL 用モデルファイルが混在しているため、取り扱いに注意が必要です。また、VHDL の場合、1 つのライブラリを作成する際に 2 つのモデルファイルが必要になります。

指定するモデルファイル名は、ライブラリ名と同じもの（例外あり）、もしくはファイル名の冒頭にライブラリ名が使用されているものをコンパイルしてください。詳細は、次頁のファンクション・シミュレーション用ライブラリの対応表をご確認ください。タイミング・シミュレーション用ライブラリの対応表は、Quartus II のヘルプ [Quartus II Help v12.1 > Altera Simulation Models](#) をご覧ください。なお、対応表では、Verilog HDL 用ライブラリの表記を VHDL と見やすく区分けするためフォルダ末尾に “_ver” と記載していますが、実際に作業する際は、_ver を付ける、付けないはユーザの管理になりますので、ご自由にご利用ください。

下表は、アルテラ・シミュレーション・ライブラリにおけるファンクション・シミュレーション用ライブラリの対応表です。タイミング・シミュレーション用ライブラリの対応表は、Quartus II のヘルプ [Quartus II Help v12.1 > Altera Simulation Models](#) をご覧ください。

ライブラリ	モデルファイル	コンパイルの目的	備考
altera	altera_primitives.vhd	アルテラ・プリミティブ	
	altera_primitives_components.vhd		
altera_ver	altera_primitives.v	アルテラ・メガファンクション	
altera_mf	altera_mf.vhd	アルテラ・メガファンクション	VHDL-93 用
	altera_mf_components.vhd		
altera_mf_ver	altera_mf.v	アルテラ・メガファンクション	
lpm	220model.vhd	アルテラ LPM	
	220pack.vhd		
lpm_ver	220model.v	アルテラ LPM	
altera_Insim	altera_Insim.sv	Stratix® V 以降の Stratix シリーズに必要	VHDL/Verilog HDL シミュレーション用 SystemVerilog ファイル
	altera_Insim_components.vhd		
altera_Insim_ver	altera_Insim.sv (メンター製品以外)	ファミリに関係なく必要	VHDL/Verilog HDL シミュレーション用 SystemVerilog ファイル
	¥mentor¥altera_Insim_for_vhdl.v		
sgate	sgate_pack.vhd	ハイレベル・プリミティブ用ライブラリ	
	sgate.vhd		
sgate_ver	sgate.v	ハイレベル・プリミティブ用ライブラリ	

- ⑥ ライブラリが複数ある場合は、操作④～⑤を該当するフォルダおよびファイルに指定しなおして実行してください。
- ⑦ ライブラリのコンパイル終了後、Done ボタンをクリックし、ライブラリ作成は終了です。
- ⑧ プロジェクトを新規作成(または既存プロジェクトを起動)します。
- ⑨ プロジェクトにおいてライブラリの登録をします。操作方法は、本資料 2-2 項“ライブラリの登録”をご参照ください。
- ⑩ 以後は、通常のシミュレーション作業を行います。

以上がプロジェクト共通でライブラリを作成し、プロジェクトごとにライブラリ登録を行う方法です。

Type B: プロジェクトごとにライブラリを作成

この方法は、プロジェクトを作成したフォルダごとにライブラリを作成し、ライブラリ登録を行うため、作業プロジェクトにおいて Quartus II バージョンごとのライブラリ管理が可能です。ただし、コンピュータ内に同一のライブラリが共存するため、コンピュータのファイルサイズが増加します。

- ① File メニュー ⇒ New ⇒ Project から、新規プロジェクトを作成(または既存プロジェクトを起動)します。
- ② ライブラリ用フォルダを作成し同時に(事前に)ライブラリ登録も行うため、File メニュー ⇒ New ⇒ Library を選択します。(図 B-1)
 - ・ Create: a new library and a logical mapping to it を選択
 - ・ Library Name: ライブラリ名を入力
 - ・ Library Physical Name: 自動的に Library Name で入力した名称が表示される

OK ボタンをクリックすると、プロジェクトのフォルダに指定した名称のフォルダが生成します。複数ライブラリを作成する場合には、この操作を繰り返してください。

作成するライブラリのフォルダ名は任意ですが、推奨はアルテラ・シミュレーション・ライブラリ名です。詳細は、本資料 7 ページ《補足:ライブラリ名に関する詳細事項》をご覧ください。

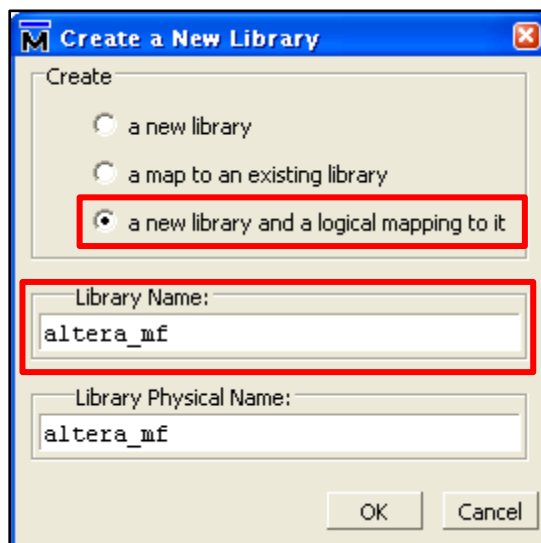


図 B-1 Create a New Library

- ③ ライブラリを作成する(ライブラリ用モデルファイルをコンパイルする)ために、File メニュー ⇒ Close Project より一度プロジェクトを終了します。
- ④ Compile メニュー ⇒ Compile を選択し、Compile Source Files ダイアログボックスを表示します。

- ⑤ “Library” のプルダウンリストから、先ほど作成したライブラリ名を選択します。(図 B-2)

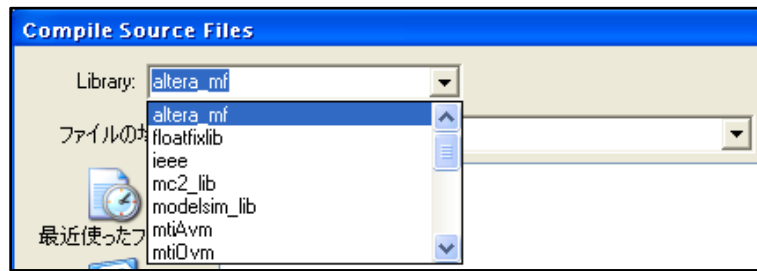


図 B-2 Compile Source Files

- ⑥ “ファイルの場所”では、⑤で指定したライブラリに該当するアルテラのシミュレーション用ライブラリのモデルファイルを指定します。アルテラのシミュレーション用ライブラリのモデルファイルの保存場所は、Quartus II のインストール・フォルダ(以下参照)にあります。(図 B-3)

¥¥Quartus II インストール・フォルダ¥eda¥sim_lib

sim_lib フォルダ内から目的のモデルファイルを指定します。その後 Compile ボタンをクリックし、コンパイルを実行します。コンパイルが完了すると、Library で指定したフォルダに指定したモデルファイルのコンパイル情報が保存されます。

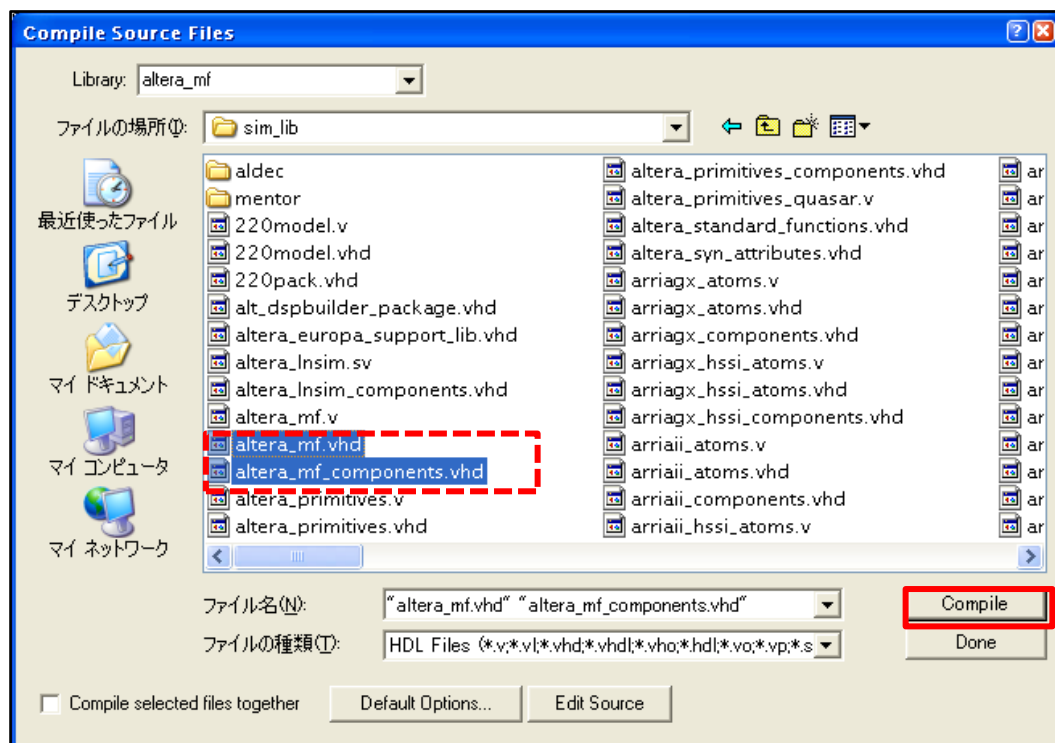


図 B-3 Compile Source Files

sim_lib フォルダには、VHDL 用モデルファイルと Verilog HDL 用モデルファイルが混在しているため、取り扱いに注意が必要です。また、VHDL の場合、1 つのライブラリを作成する際に 2 つのモデルファイルが必要になります。

指定するモデルファイル名は、ライブラリ名と同じもの（例外あり）、もしくはファイル名の冒頭にライブラリ名が使用されているものをコンパイルしてください。詳細は、下記のファンクション・シミュレーション用ライブラリの対応表をご確認ください。タイミング・シミュレーション用ライブラリの対応表は、Quartus II のヘルプ [Quartus II Help v12.1 > Altera Simulation Models](#) をご覧ください。なお、対応表では、Verilog HDL 用ライブラリの表記を VHDL と見やすく区分けするためフォルダ末尾に “_ver” と記載していますが、実際に作業する際は、_ver を付ける、付けないはユーザの管理になりますので、ご自由にご利用ください。

下表は、アルテラ・シミュレーション・ライブラリにおけるファンクション・シミュレーション用ライブラリの対応表です。タイミング・シミュレーション用ライブラリの対応表は、Quartus II のヘルプ [Quartus II Help v12.1 > Altera Simulation Models](#) をご覧ください。

ライブラリ	モデルファイル	コンパイルの目的	備考
altera	altera_primitives.vhd	アルテラ・プリミティブ	
	altera_primitives_components.vhd		
altera_ver	altera_primitives.v	アルテラ・メガファンクション	
altera_mf	altera_mf.vhd	アルテラ・メガファンクション	VHDL-93 用
	altera_mf_components.vhd		
altera_mf_ver	altera_mf.v	アルテラ・メガファンクション	
lpm	220model.vhd	アルテラ LPM	
	220pack.vhd		
ライブラリ	モデルファイル	コンパイルの目的	備考
lpm_ver	220model.v	アルテラ LPM	
altera_Insim	altera_Insim.sv	Stratix® V 以降の Stratix シリーズに必要	VHDL/Verilog HDL シミュレーション用 SystemVerilog ファイル
	altera_Insim_components.vhd		
altera_Insim_ver	altera_Insim.sv (メンター製品以外)	ファミリに関係なく必要	VHDL/Verilog HDL シミュレーション用 SystemVerilog ファイル
	¥mentor¥altera_Insim_for_vhdl.v		
sgate	sgate_pack.vhd	ハイレベル・プリミティブ用ライブラリ	
	sgate.vhd		
sgate_ver	sgate.v	ハイレベル・プリミティブ用ライブラリ	

- ⑦ ライブラリが複数ある場合は、操作⑤～⑥を該当するフォルダおよびファイルに指定しなおして実行してください。
- ⑧ ライブラリのコンパイル終了後、Done ボタンをクリックし、ライブラリ作成は終了です。
- ⑨ プロジェクトを再度起動します。
- ⑩ 以後は、通常のシミュレーション作業を行います。

以上がプロジェクトごとにライブラリを作成し、ライブラリ登録を行う方法です。

改版履歴

Revision	年月	概要
1	2013 年 3 月	新規作成

免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。

株式会社アルティマ : 〒222-8563 横浜市港北区新横浜 1-5-5 マクニカ第二ビル

TEL: 045-476-2155 HP: <http://www.altima.co.jp>

技術情報サイト EDISON :

<https://www.altima.jp/members/index.cfm>

株式会社エルセナ : 〒163-0928 東京都新宿区西新宿 2-3-1 新宿モノリス 28F

TEL: 03-3345-6205 HP: <http://www.elsena.co.jp>

技術情報サイト ETS :

<https://www.elsena.co.jp/elspear/members/index.cfm>
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる場合は、英語版の資料もあわせてご利用ください。