

旧バージョン開発ツールでの ModelSim Lattice Edition使用方法



はじめに

- 本マニュアルはLattice社FPGAデバイス設計ツール Diamond 3.12以降、及びRadiant2.2以降にバンドルされたModelSim Lattice Editionを用いて、旧バージョンの開発ツール（Diamond 3.11以前、及びRadiant2.1以前）にてシミュレーションを行う場合のオペレーションマニュアルです
- 本マニュアルでは旧バージョンのDiamond及びRadiantにて、Simulation Wizardから ModelSim Lattice Editionによるシミュレーション環境を作成、立ち上げ、シミュレーション実行の方法について説明しています
- 本マニュアルの内容の確認は以下のツールで実施しています
 - Diamond編：Diamond3.11 SP3
 - Radiant編：Radiant2.1 SP1
- 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

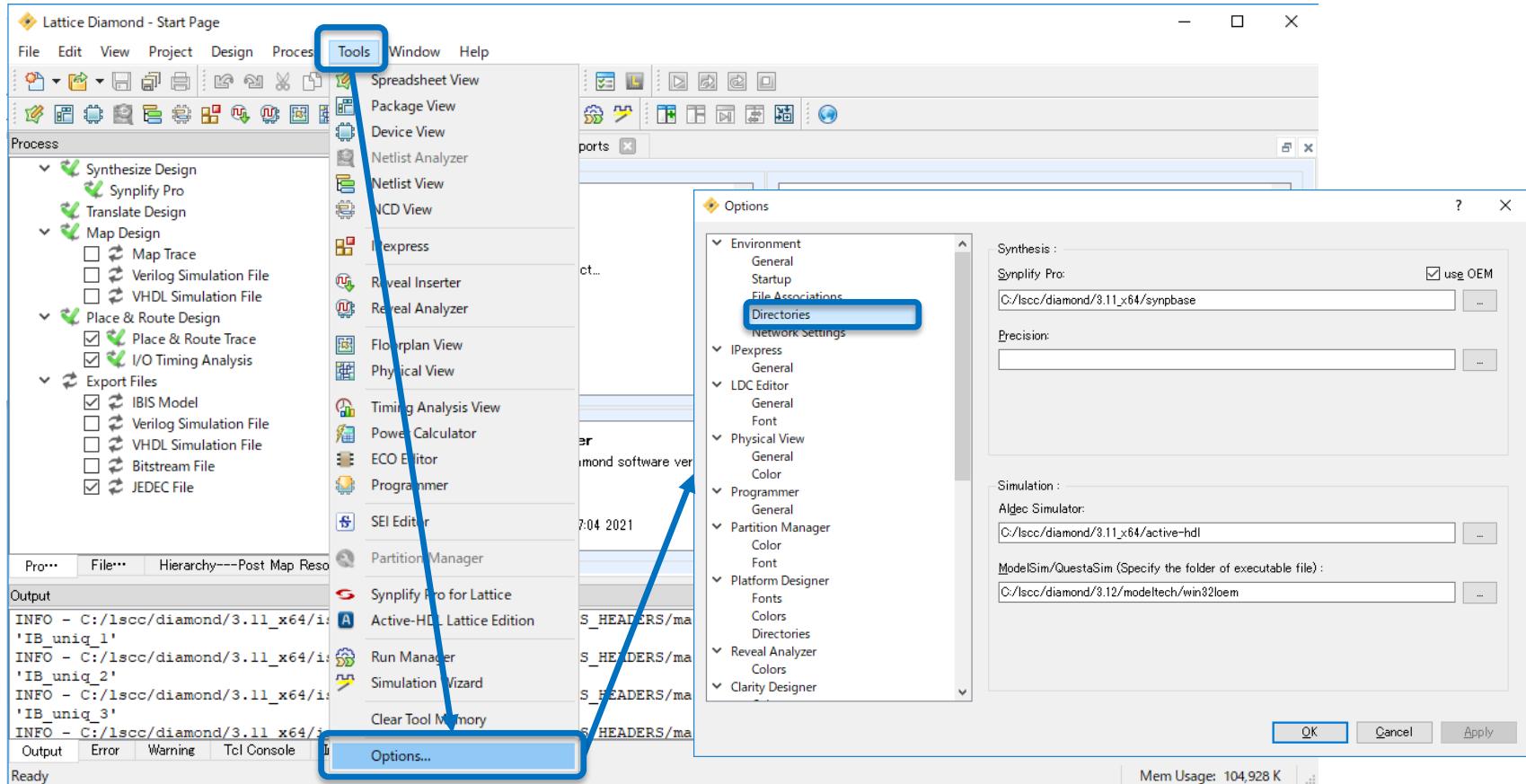


Diamond編

1. ModelSim Lattice EditionのDirectory指定



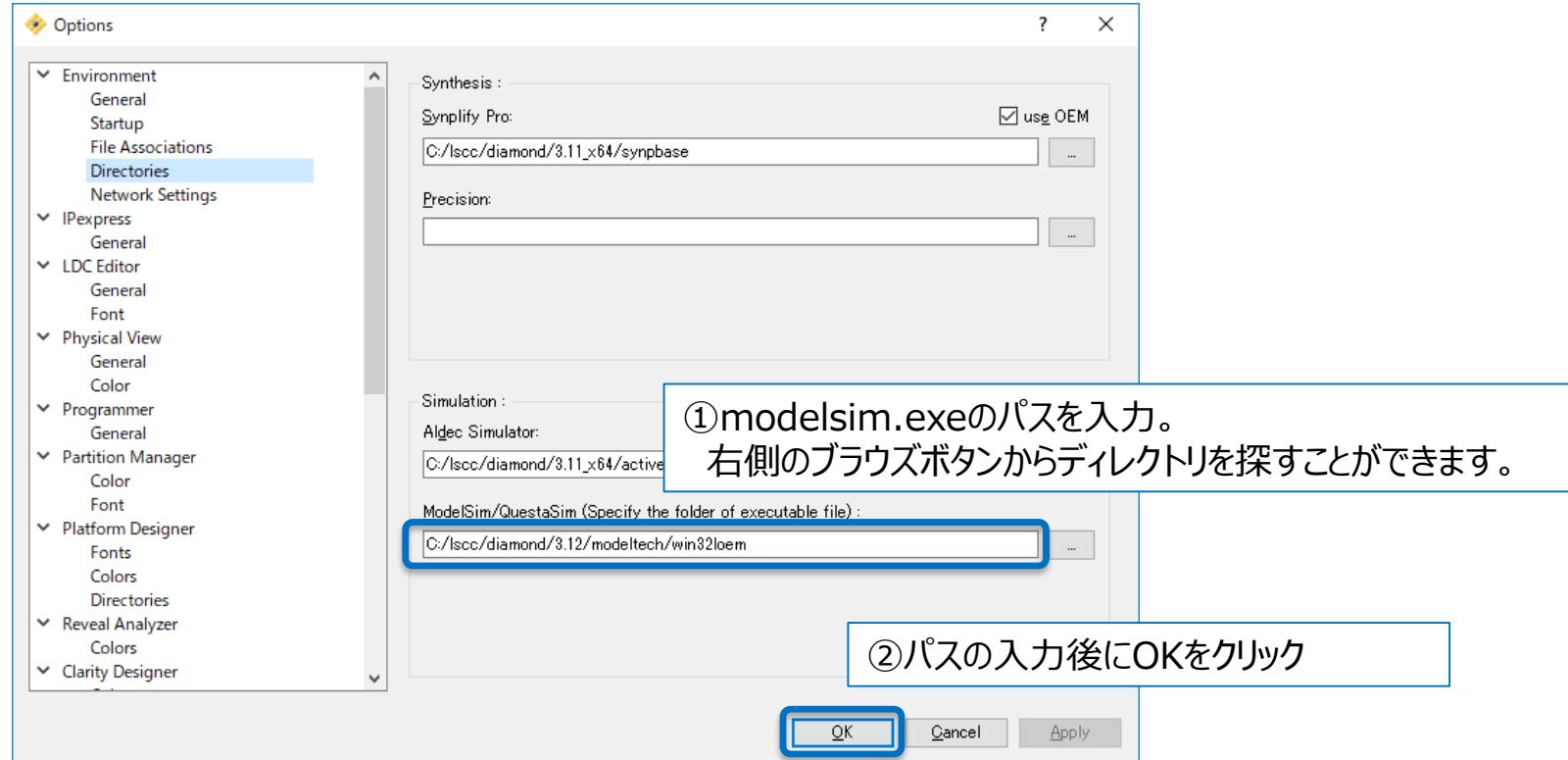
- 最初にDiamond3.11以前のバージョンでmodelsim.exeファイルのパスを指定します
Tools > OptionsでOption画面を開き、Environment下のDirectoriesを選択します



1. ModelSim Lattice EditionのDirectory指定

- Directory設定でDiamond3.12以降にバンドルされているModelSim Lattice Editionの modelsim.exeのパスを指定します。Diamond3.12の場合は、modelsim.exeはDiamond インストールフォルダの以下のパスに存在しています

¥lscc¥diamond¥3.12¥modeltech¥win32loem





1. ModelSim Lattice EditionのDirectory指定

- 注意：旧バージョンの Diamond で指定する modelsim.exe のパスは、ModelSim がバンドルされている Diamond 3.12 のインストール・ディレクトリ下から指定してください
- Radiant 2.2 のインストール・ディレクトリ下から指定した場合、Diamond とはサポートするデバイスが異なるため、ModelSim がロードするライブラリも異なり、エラーになります
 - 下に初期化ファイル modelsim.ini の一部を示します。それぞれ青枠は Verilog ライブラリです

Diamondバンドル版のmodelsim.ini
(ModelSimライブラリ参照ファイル)

```

9 [Library]↓
10 std = $MODEL_TECH/..../std↓
11 ieee = $MODEL_TECH/..../ieee↓
12 vital2000 = $MODEL_TECH/..../vital2000↓
13 verilog = $MODEL_TECH/..../verilog↓
14 std_developerskit = $MODEL_TECH/..../std_developerskit↓
15 synopsys = $MODEL_TECH/..../synopsys↓
16 modelsim_lib = $MODEL_TECH/..../modelsim_lib↓
17 sv_std = $MODEL_TECH/..../sv_std↓
18 floatfixlib = $MODEL_TECH/..../floatfixlib↓
19 ; Lattice Primitive Libraries↓
20 ;↓
21 ovi_ec = $MODEL_TECH/..../lib/ovi_ec↓
22 ovi_ecp = $MODEL_TECH/..../lib/ovi_ecp↓
23 ovi_ecp2 = $MODEL_TECH/..../lib/ovi_ecp2↓
24 ovi_ecp3 = $MODEL_TECH/..../lib/ovi_ecp3↓
25 ovi_ecp5u = $MODEL_TECH/..../lib/ovi_ecp5u↓
26 ovi_lfmnx = $MODEL_TECH/..../lib/ovi_lfmnx↓
27 ovi_lifmd = $MODEL_TECH/..../lib/ovi_lifmd↓
28 ovi_lifmdf = $MODEL_TECH/..../lib/ovi_lifmdf↓
29 ovi_lptm = $MODEL_TECH/..../lib/ovi_lptm↓
30 ovi_lptm2 = $MODEL_TECH/..../lib/ovi_lptm2↓
31 ovi_machxo = $MODEL_TECH/..../lib/ovi_machxo↓
32 ovi_machxo2 = $MODEL_TECH/..../lib/ovi_machxo2↓
33 ovi_machxo3d = $MODEL_TECH/..../lib/ovi_machxo3d↓
34 ovi_machxo3l = $MODEL_TECH/..../lib/ovi_machxo3l↓
35 ovi_prism = $MODEL_TECH/..../lib/ovi_prism↓
36 ovi_sc = $MODEL_TECH/..../lib/ovi_sc↓
37 ovi_xp = $MODEL_TECH/..../lib/ovi_xp↓
38 ovi_xp2 = $MODEL_TECH/..../lib/ovi_xp2↓
39 ovi_ecp2 = $MODEL_TECH/..../lib/ovi_ecp2↓
40 ovi_ecp2s = $MODEL_TECH/..../lib/ovi_ecp2↓
41 ovi_ecp2ns = $MODEL_TECH/..../lib/ovi_ecp2↓
42 ovi_scm = $MODEL_TECH/..../lib/ovi_sc↓
43 ovi_ecp5um = $MODEL_TECH/..../lib/ovi_ecp5u↓

```

Radiantバンドル版のmodelsim.ini

```

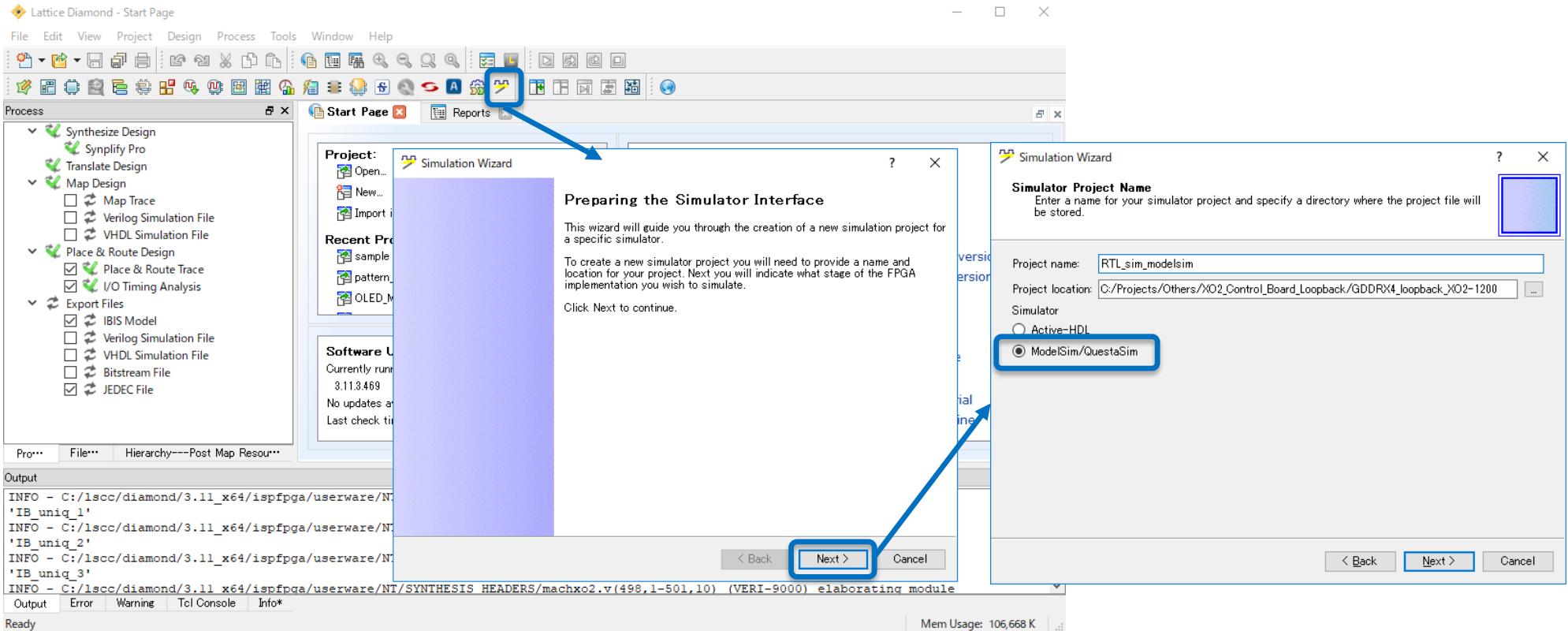
9 [Library]↓
10 std = $MODEL_TECH/..../std↓
11 ieee = $MODEL_TECH/..../ieee↓
12 vital2000 = $MODEL_TECH/..../vital2000↓
13 verilog = $MODEL_TECH/..../verilog↓
14 std_developerskit = $MODEL_TECH/..../std_developerskit↓
15 synopsys = $MODEL_TECH/..../synopsys↓
16 modelsim_lib = $MODEL_TECH/..../modelsim_lib↓
17 sv_std = $MODEL_TECH/..../sv_std↓
18 floatfixlib = $MODEL_TECH/..../floatfixlib↓
19 ; Lattice Primitive Libraries↓
20 ;↓
21 ovi_lifcl = $MODEL_TECH/..../lib/ovi_lifcl↓
22 ovi_lfd2nx = $MODEL_TECH/..../lib/ovi_lfd2nx↓
23 ovi_lce40up = $MODEL_TECH/..../lib/ovi_lce40up↓
24 pmi_work = $MODEL_TECH/..../lib/pmi_work↓
25 ; VHDL Section↓
26 ;↓
27 lifcl = $MODEL_TECH/..../lib/lifcl↓
28 lfd2nx = $MODEL_TECH/..../lib/lfd2nx↓
29 lce40up = $MODEL_TECH/..../lib/lce40up↓
30 mach = $MODEL_TECH/..../lattice/vhdl/mach↓
31 vfl1 = $MODEL_TECH/..../lattice/vhdl/vfl1↓
32 lat_vhd = $MODEL_TECH/..../lattice/vhdl/lat_vhd↓
33 lat_viti = $MODEL_TECH/..../lattice/vhdl/lat_viti↓

```

それぞれのツールでサポートするデバイスが異なるため、用意されているライブラリと参照ファイルの内容が異なります

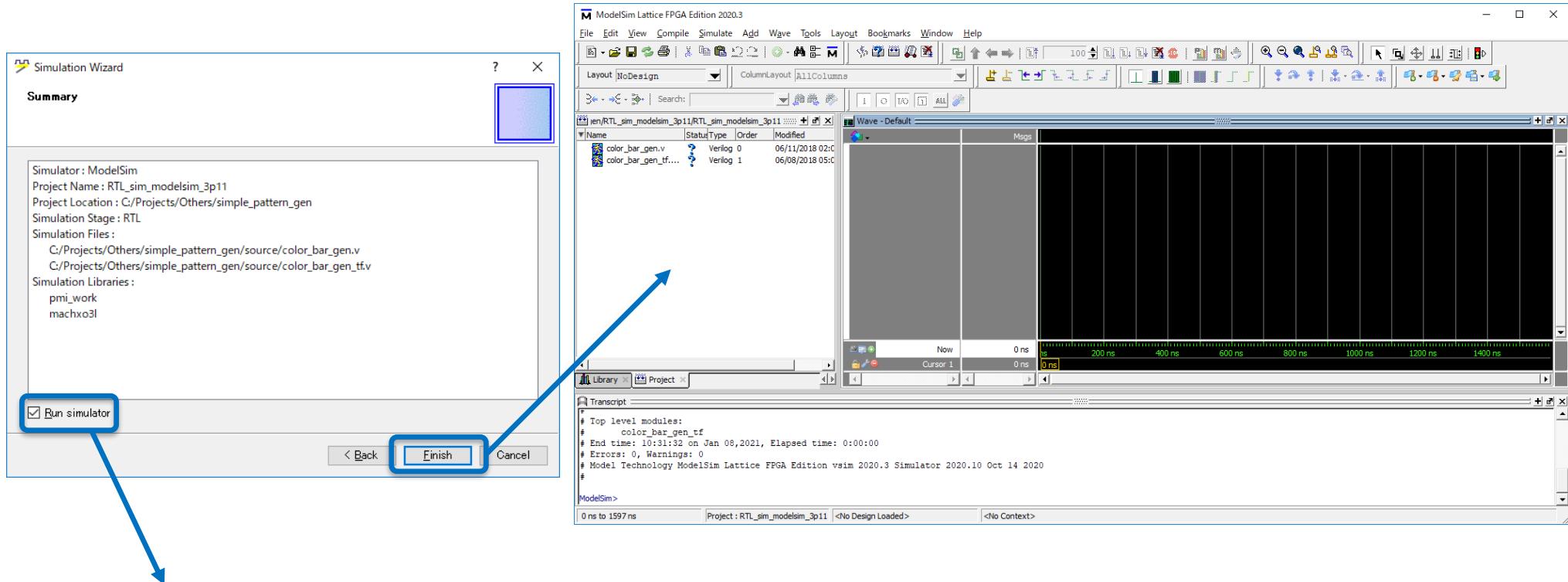
2. Simulation Wizardの起動

- Diamond GUI上のアイコンからSimulation Wizardを起動します
- Simulation Wizardの次の画面でModelSim/QuestasimをSimulatorとして選択します
- その後のModel Sim起動までのSimulation Wizardの操作はActive-HDLの場合と同じです



2. Simulation Wizardの起動

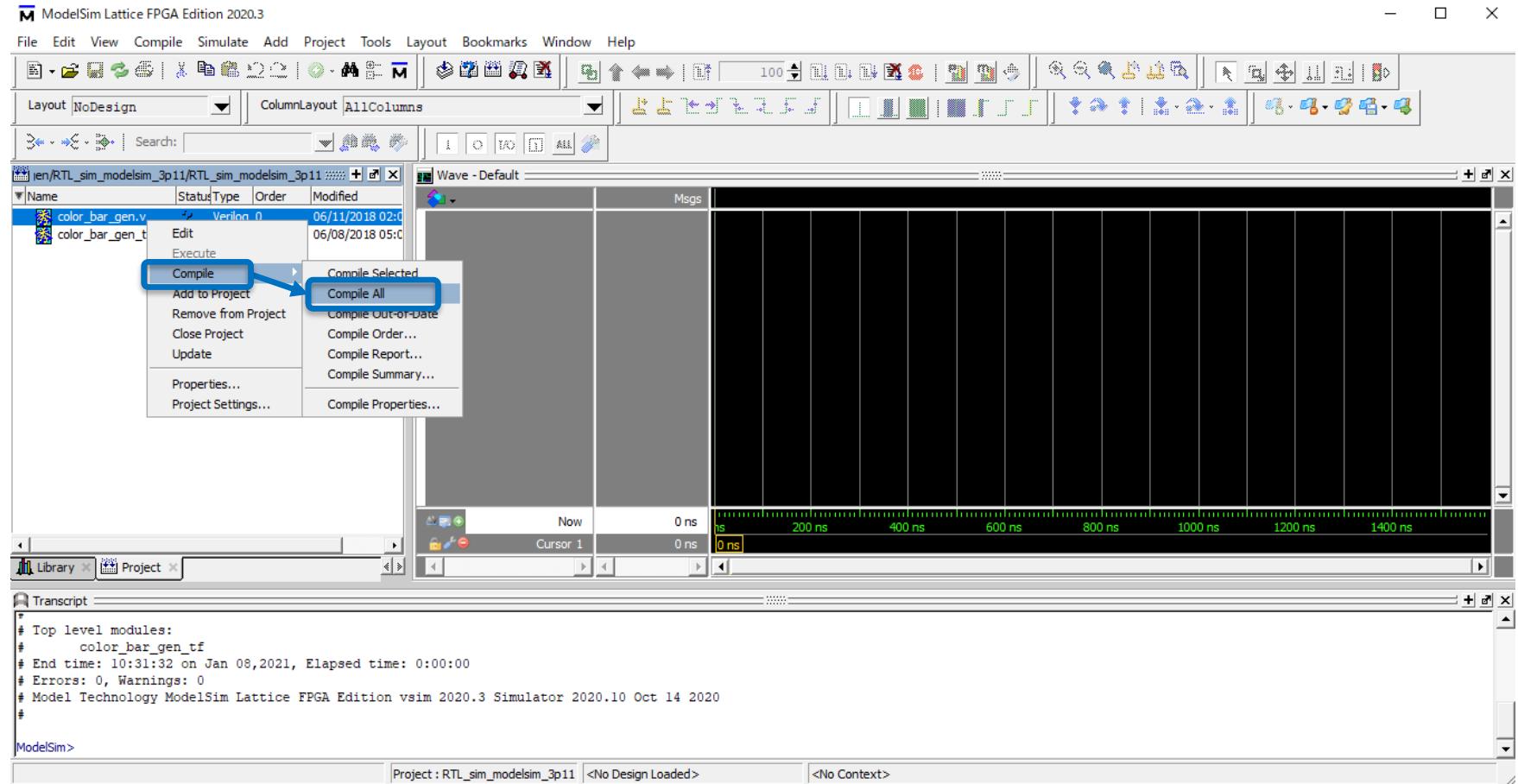
- Simulation Wizardの最後の画面でエラー無くFinishをクリックするとModelSimが起動します



Diamond 3.12 ではこれ以外に
Add top-level signals to waveform display
Run Simulation
もデフォルトでインエーブルされてシミュレーション実行まで行われます

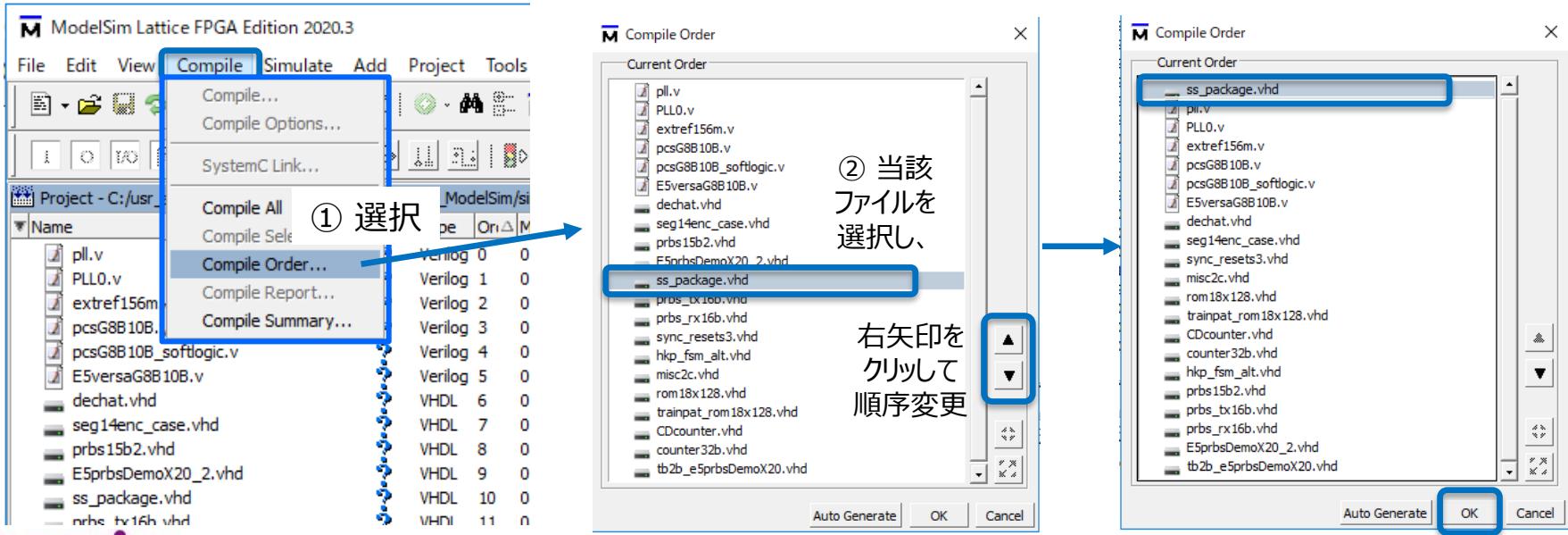
3. ソースコンパイル (Verilog)

- Project Viewからいずれかのソースファイルを右クリックし、Compile > Compile Allを実行します
- Compile Allが完了すると全てのソースファイルの右側のステータスアイコンが ? から ✓ に変わります



3. ソースコンパイル (VHDL)

- VHDLベースのプロジェクトでは、ソースファイルのコンパイル順に留意する必要があります
 - ユーザ定義パッケージを記述したソースファイルを用いる場合は、最初にコンパイルするように設定します
 - 想定される順序と異なる場合はコンパイルエラーになります
 - モジュール階層に関しては、特に下位モジュールからのコンパイルが必須ではありません（任意）
- ① メニュー Compile > Compile order... を選択します
 - ② 表示ウィンドウで当該ファイルを選択し、右側の矢印をクリックしてコンパイル順を変更します
 - ③ 完了したら OK ボタンをクリックして抜けます
 - ④ その後、前ページのように Compile > Compile All を選択します





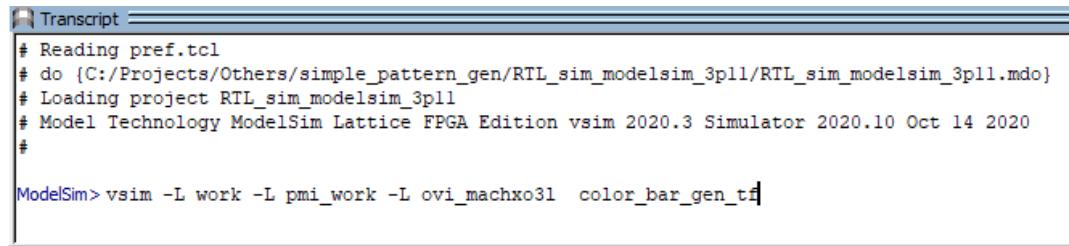
4. シミュレーション初期化

- コンソール (Transcript) から以下のvsimコマンドを実行し、シミュレーションを初期化します

vsim -L work -L (ライブラリ) -L (ライブラリ) ... (テストベンチモジュール名)

- 例えばXO3デバイスに対してcolor_bar_gen_tf (テストベンチソース内で記述したテストベンチ名) というモジュールのシミュレーション初期化の場合、以下のような記述になります

vsim -L work -L pmi_work -L ovi_machxo3l color_bar_gen_tf

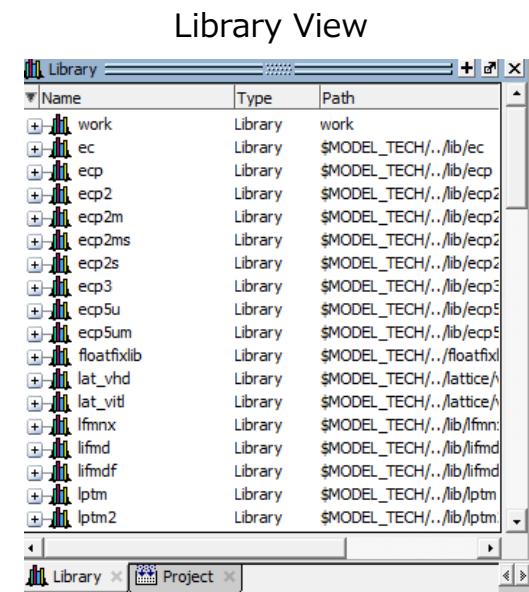


```
# Transcript
# Reading pref.tcl
# do {C:/Projects/Others/simple_pattern_gen/RTL_sim_modelsim_3pll/RTL_sim_modelsim_3pll.mdo}
# Loading project RTL_sim_modelsim_3pll
# Model Technology ModelSim Lattice FPGA Edition vsim 2020.3 Simulator 2020.10 Oct 14 2020
#
ModelSim> vsim -L work -L pmi_work -L ovi_machxo3l color_bar_gen_tf
```

- VHDLのみの場合は以下になります（テストベンチ名が同じと仮定）

vsim -L work -L pmi_work -L machxo3l color_bar_gen_tf

- 参照するライブラリは概ね Library View のリストから予測できますが、分からぬ場合は Diamond 3.12 以降で Simulation Wizard から ModelSim Lattice Edition を立ち上げ、シミュレーションフォルダ内に生成される .mdo ファイルを参照すると分かります（次ページ参照）



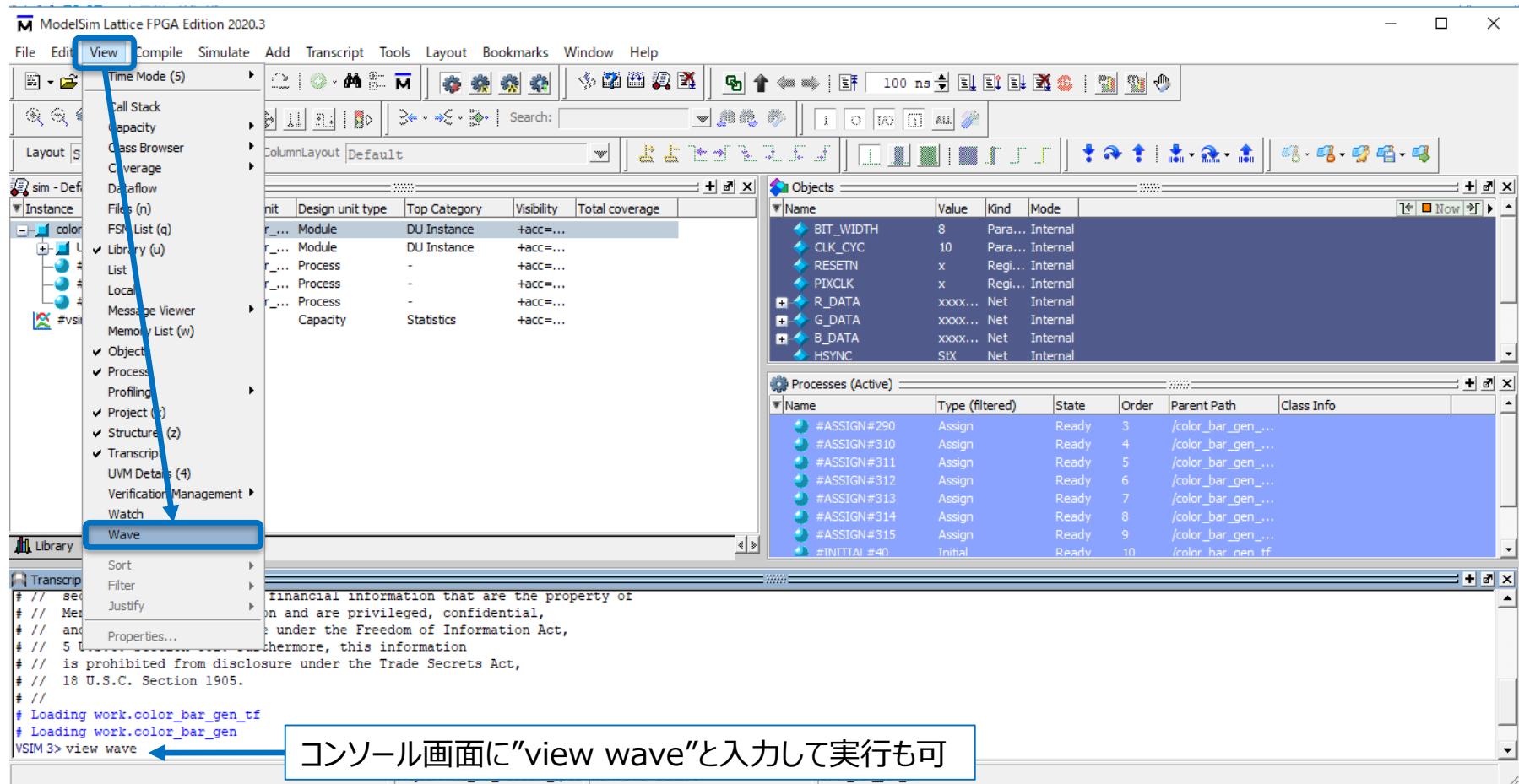
(参考)vsimコマンドの確認 .mdoファイルの参照

- .mdoファイルはDiamondプロジェクトフォルダ内にある、シミュレーションプロジェクトフォルダ (Simulation Wizardで最初に指定したプロジェクト名のフォルダ) の中に存在します
- 例としてDiamond3.12で生成された.mdoファイルを参照すると、以下のようにになっており、青線部にシミュレーション開始時に実行するvsimコマンドの内容が記述されています

```
1 if {[file exists "C:/Projects/Others/simple_pattern_gen/RTL_sim_modelsim/RTL_sim_modelsim.mpf"]} { ↓
2   project new "C:/Projects/Others/simple_pattern_gen/RTL_sim_modelsim" "RTL_sim_modelsim" ↓
3   project addfile "C:/Projects/Others/simple_pattern_gen/source/color_bar_gen.v" ↓
4   project addfile "C:/Projects/Others/simple_pattern_gen/source/color_bar_gen_tf.v" ↓
5   vlib work ↓
6   vdel -lib work -all ↓
7   vlib work ↓
8   vlog +incdir+C:/Projects/Others/simple_pattern_gen/source -work work "C:/Projects/Others/simple_pattern_gen/source/color_bar_gen.v" ↓
9   vlog +incdir+C:/Projects/Others/simple_pattern_gen/source -work work "C:/Projects/Others/simple_pattern_gen/source/color_bar_gen_tf.v" ↓
10 } else { ↓
11   project open "C:/Projects/Others/simple_pattern_gen/RTL_sim_modelsim/RTL_sim_modelsim" ↓
12   project compileoutofdate ↓
13 } ↓
14 vsim -L work -L pmi_work -L ovi_machxo3l color_bar_gen_tf ↓
15 view wave ↓
16 add wave /* ↓
17 run 1000ns ↓
18 [EOF]
```

5. 波形ウィンドウの表示

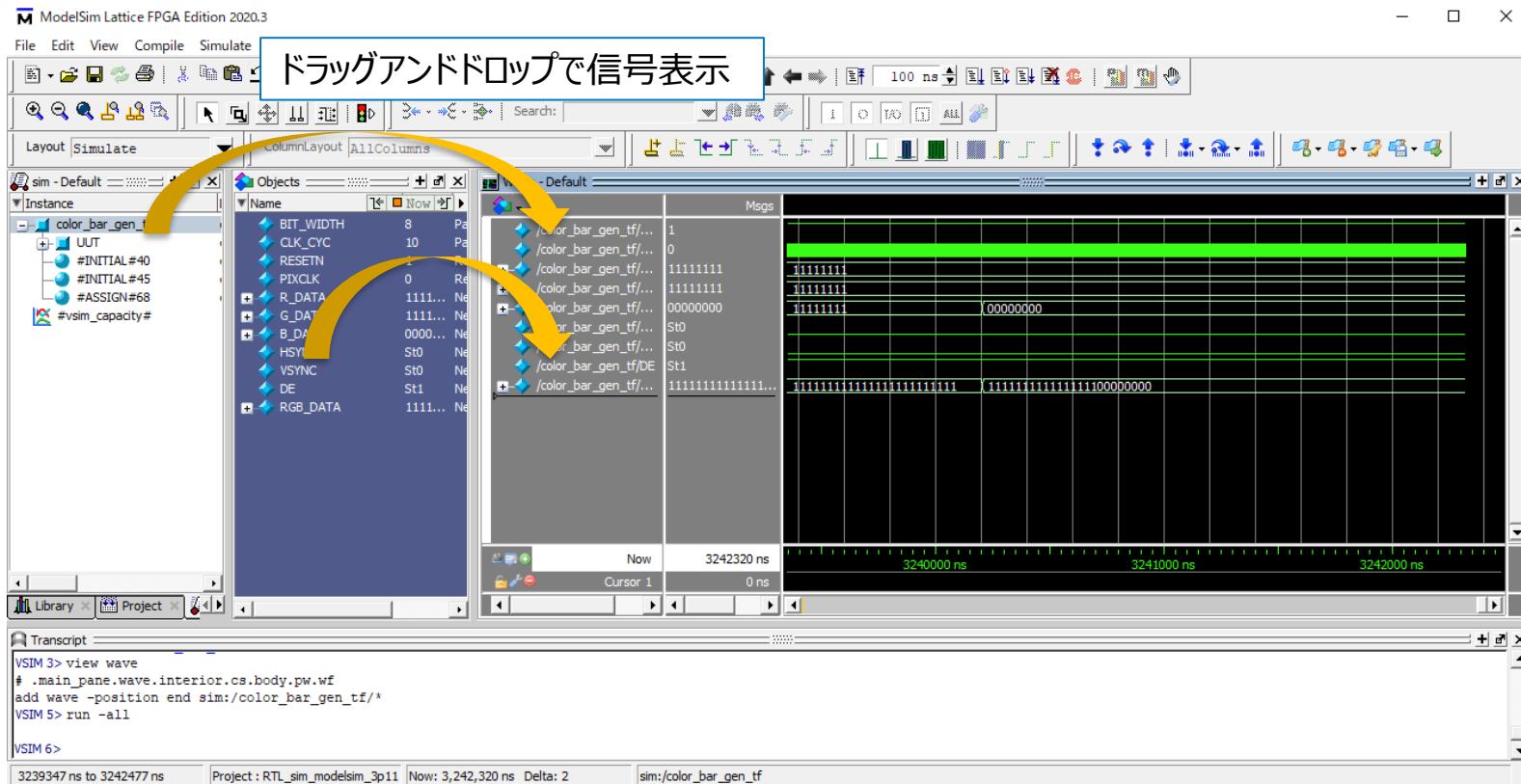
- シミュレーションの初期化完了後、波形ウィンドウを表示します。View > Waveを選択、もしくはコンソール画面に“view wave”と入力します



6. 信号の表示、シミュレーションの実行

NOWHERE,
but **HERE.**

- Object Viewやストラクチャ構造を表示するウィンドウからドラッグアンドドロップで信号を波形ウィンドウに表示します



- 信号が表示されたらRunやRun Allを実行するとシミュレーションが開始されます

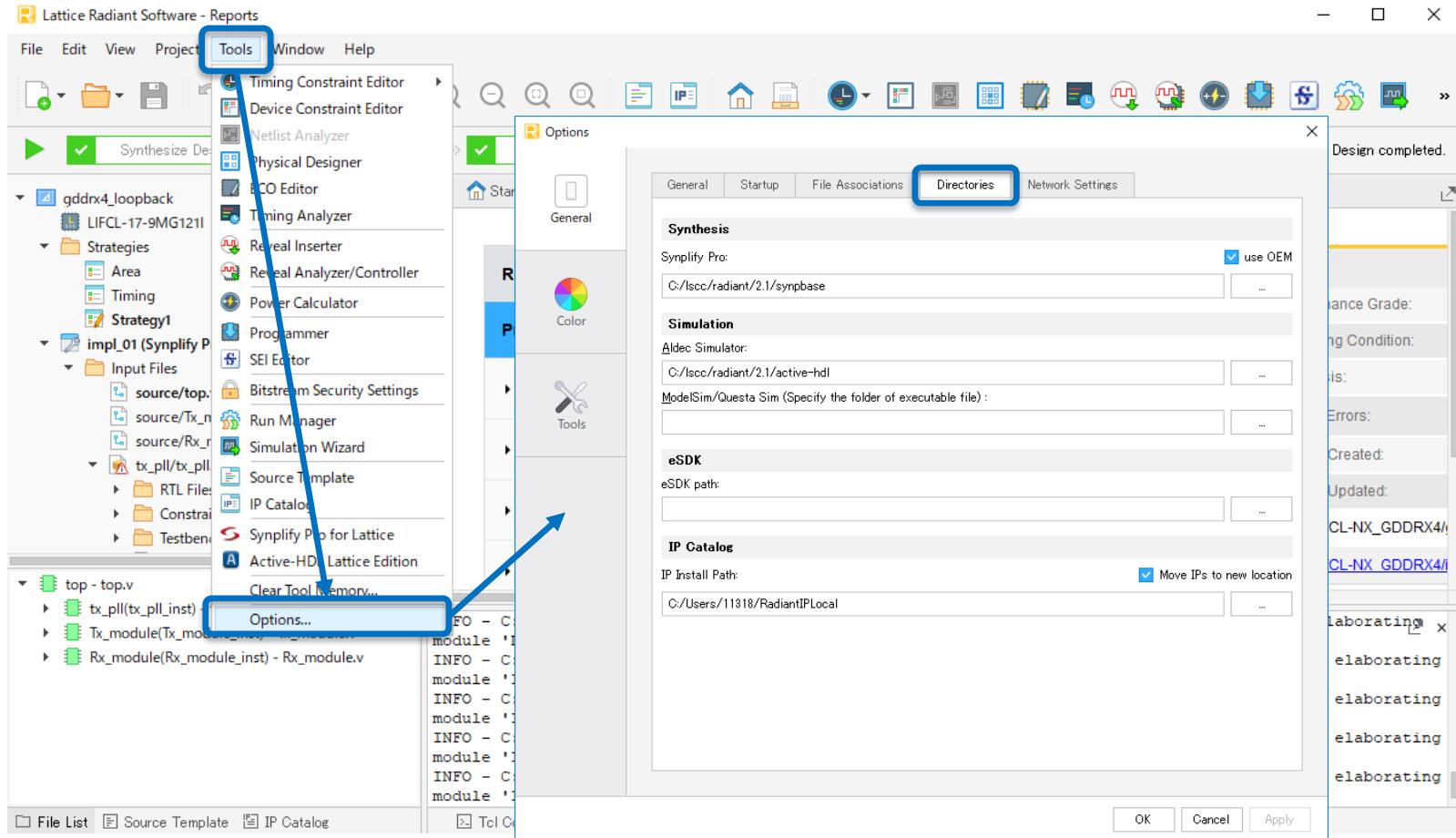


Radiant編

1. ModelSim Lattice EditionのDirectory指定



- 最初にRadiant2.1以前のバージョンでmodelsim.exeファイルのパスを指定します
Tools > OptionsでOption画面を開き、上部のタブからDirectoriesを選択します

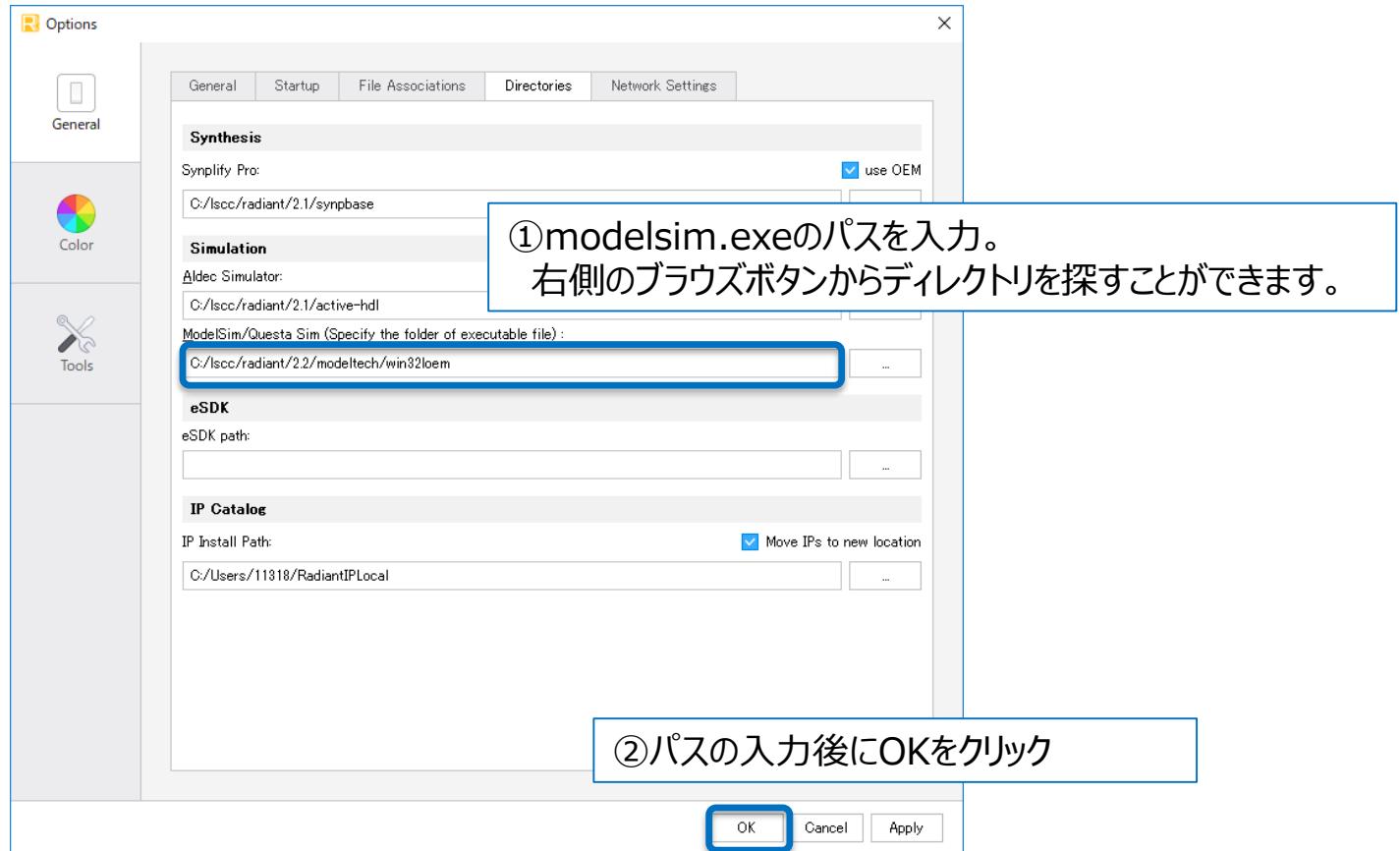


1. ModelSim Lattice EditionのDirectory指定



- Directory設定でRadiant2.2以降にバンドルされているModelSim Lattice Editionの modelsim.exeのパスを指定します。Radiant2.2の場合は、modelsim.exeはRadiantインストールフォルダの以下のパスに存在しています

¥lscc¥radiantradiant2.2¥modeltech¥win32loem





1. ModelSim Lattice EditionのDirectory指定

- 注意：旧バージョンの Radiant で指定する modelsim.exe のパスは、ModelSim がバンドルされている Radiant 2.2 のインストール・ディレクトリ下から指定してください
- Diamond 3.12 のインストール・ディレクトリ下から指定した場合、Radiant とはサポートするデバイスが異なるため、ModelSim がロードするライブラリも異なり、エラーになります
 - 下に初期化ファイル modelsim.ini の一部を示します。それぞれ青枠は Verilog ライブラリです

Diamondバンドル版のmodelsim.ini
(ModelSimライブラリ参照ファイル)

```

9 [Library]↓
10 std = $MODEL_TECH/..../std↓
11 ieee = $MODEL_TECH/..../ieee↓
12 vital2000 = $MODEL_TECH/..../vital2000↓
13 verilog = $MODEL_TECH/..../verilog↓
14 std_developerskit = $MODEL_TECH/..../std_developerskit↓
15 synopsys = $MODEL_TECH/..../synopsys↓
16 modelsim_lib = $MODEL_TECH/..../modelsim_lib↓
17 sv_std = $MODEL_TECH/..../sv_std↓
18 floatfixlib = $MODEL_TECH/..../floatfixlib↓
19 ; Lattice Primitive Libraries↓
20 :↓
21 ovi_ec = $MODEL_TECH/..../lib/ovi_ec↓
22 ovi_ecp = $MODEL_TECH/..../lib/ovi_ecp↓
23 ovi_ecp2 = $MODEL_TECH/..../lib/ovi_ecp2↓
24 ovi_ecp3 = $MODEL_TECH/..../lib/ovi_ecp3↓
25 ovi_ecp5u = $MODEL_TECH/..../lib/ovi_ecp5u↓
26 ovi_ifmnx = $MODEL_TECH/..../lib/ovi_ifmnx↓
27 ovi_ifmd = $MODEL_TECH/..../lib/ovi_ifmd↓
28 ovi_ifmdf = $MODEL_TECH/..../lib/ovi_ifmdf↓
29 ovi_lptm = $MODEL_TECH/..../lib/ovi_lptm↓
30 ovi_lptm2 = $MODEL_TECH/..../lib/ovi_lptm2↓
31 ovi_machxo = $MODEL_TECH/..../lib/ovi_machxo↓
32 ovi_machxo2 = $MODEL_TECH/..../lib/ovi_machxo2↓
33 ovi_machxo3d = $MODEL_TECH/..../lib/ovi_machxo3d↓
34 ovi_machxo3l = $MODEL_TECH/..../lib/ovi_machxo3l↓
35 ovi_prism = $MODEL_TECH/..../lib/ovi_prism↓
36 ovi_sc = $MODEL_TECH/..../lib/ovi_sc↓
37 ovi_xp = $MODEL_TECH/..../lib/ovi_xp↓
38 ovi_xp2 = $MODEL_TECH/..../lib/ovi_xp2↓
39 ovi_ecp2m = $MODEL_TECH/..../lib/ovi_ecp2↓
40 ovi_ecp2s = $MODEL_TECH/..../lib/ovi_ecp2↓
41 ovi_ecp2ms = $MODEL_TECH/..../lib/ovi_ecp2↓
42 ovi_scm = $MODEL_TECH/..../lib/ovi_sc↓
43 ovi_ecp5um = $MODEL_TECH/..../lib/ovi_ecp5u↓

```

Radiantバンドル版のmodelsim.ini

```

9 [Library]↓
10 std = $MODEL_TECH/..../std↓
11 ieee = $MODEL_TECH/..../ieee↓
12 vital2000 = $MODEL_TECH/..../vital2000↓
13 verilog = $MODEL_TECH/..../verilog↓
14 std_developerskit = $MODEL_TECH/..../std_developerskit↓
15 synopsys = $MODEL_TECH/..../synopsys↓
16 modelsim_lib = $MODEL_TECH/..../modelsim_lib↓
17 sv_std = $MODEL_TECH/..../sv_std↓
18 floatfixlib = $MODEL_TECH/..../floatfixlib↓
19 ; Lattice Primitive Libraries↓
20 :↓
21 ovi_lifcl = $MODEL_TECH/..../lib/ovi_lifcl↓
22 ovi_lfd2nx = $MODEL_TECH/..../lib/ovi_lfd2nx↓
23 ovi_ICE40UP = $MODEL_TECH/..../lib/ovi_ICE40UP↓
24 pmi_work = $MODEL_TECH/..../lib/pmi_work↓
25 ; VHDL Section↓
26 :↓
27 lifcl = $MODEL_TECH/..../lib/lifcl↓
28 lfd2nx = $MODEL_TECH/..../lib/lfd2nx↓
29 ICE40UP = $MODEL_TECH/..../lib/ICE40UP↓
30 mach = $MODEL_TECH/..../lattice/vhdl/mach↓
31 vfl = $MODEL_TECH/..../lattice/vhdl/vfl↓
32 lat_vhd = $MODEL_TECH/..../lattice/vhdl/lat_vhd↓
33 lat_viti = $MODEL_TECH/..../lattice/vhdl/lat_viti↓

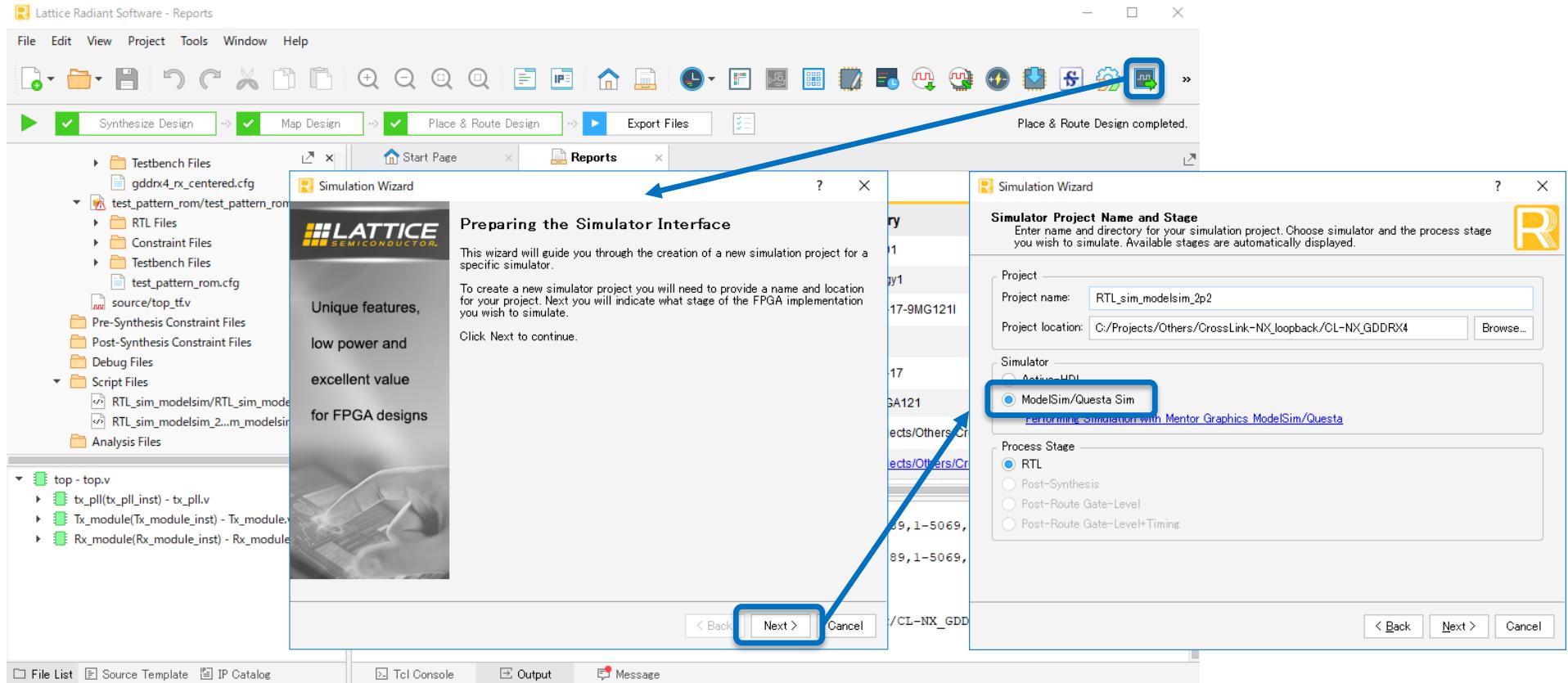
```

それぞれのツールでサポートするデバイスが異なるため、用意されているライブラリと参照ファイルの内容が異なります



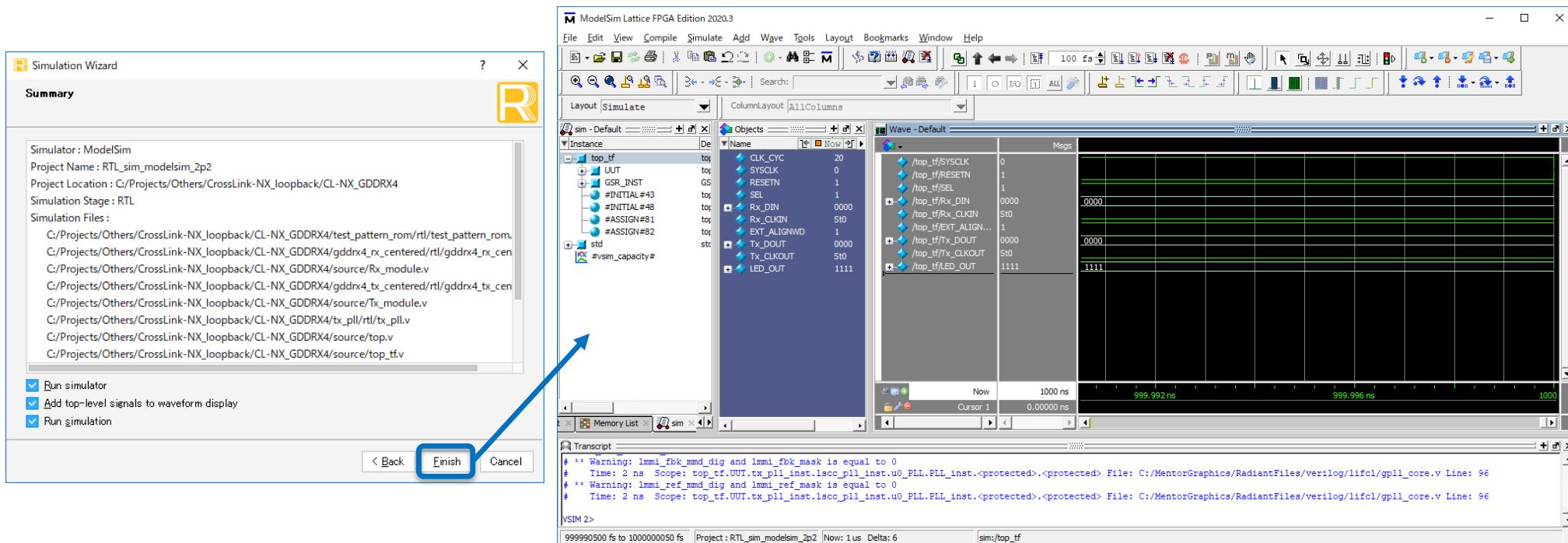
2. Simulation Wizardの起動

- Radiant GUI上のアイコン  からSimulation Wizardを起動します
- Simulation Wizardの次の画面でModelSim/QuestasimをSimulatorとして選択します
- その後のModel Sim起動までのSimulation Wizardの操作はActive-HDLの場合と同じです



2. Simulation Wizardの起動

- Simulation Wizardの最後の画面でエラー無くFinishをクリックするとModelSimが起動します
- 起動後に自動的にソースコンパイル、シミュレーションの初期化、波形ウィンドウの表示、テストベンチ上の信号表示まで行われます



- その後RunやRun Allを実行してシミュレーションを開始します

Revision History



| Date | Revision | Page | Change Information | Updated by |
|------------|----------|----------------|--|--------------|
| 2021/01/08 | 0.1 | | First Revision | W. Nakatsuka |
| 2021/01/14 | 0.2 | 6,10,11, 18 | Added VHDL related info (Minor change on p.6, p.11 & p.18, added p.10) | S. Sugawara |
| 2021/01/14 | 0.3 | | Modified the small mistake | H.Nogawa |
| 2021/01/14 | 0.4 | 17 | Fix miss-text from "Diamond" to "Radian" | S.Katube |
| | | | | |