

ModelSim Lattice Edition ユーザガイド (Rev.0.4)



- 本ユーザガイドは Lattice 社 FPGA デバイス設計ツール [Diamond 3.12](#) 及び [Radiant 2.2](#) にバンドルされた [ModelSim Lattice Edition](#) (以降 [ModelSim LE](#)) を用いる場合の操作ガイドです
 - 主にマウス操作による基本的な作業についてのみ記述しています
 - 詳細は `C:\%lscc%\<radiant / diamond install-dir>%modeltech%\docs\pdfdocs` 下の各ドキュメントをご参照ください
- ModelSim LE による論理シミュレーション実行には、三つの方法があります
 1. Diamond / Radiant に組み込まれている “[Simulation Wizard](#)” から自動的に ModelSim LE を立ち上げ、シミュレーションを実行
 2. まず ModelSim LE GUI を立ち上げ、主にマウス操作によって各ステップを処理してシミュレーションを実行
 3. まず ModelSim LE GUI を立ち上げ、予め用意するテキスト『do マクロ』（TCL スクリプト）を読み込むことでシミュレーションを実行
- 本ユーザガイドは上記手順 2 についての記述です
 - 上記 1 については別途『[ModelSim_LE_for_old_version_tools](#)』をご参照ください
 - 上記 3 については別途『[ModelSim_LE_do-macro](#)』（準備中）をご参照ください
- 本ユーザガイドと同等の 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

内容

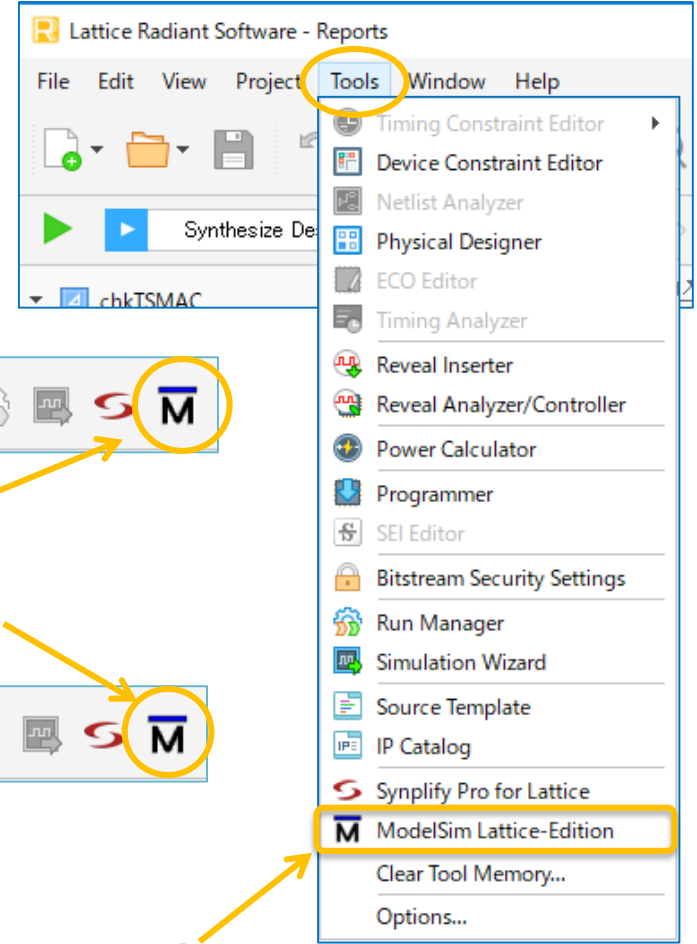
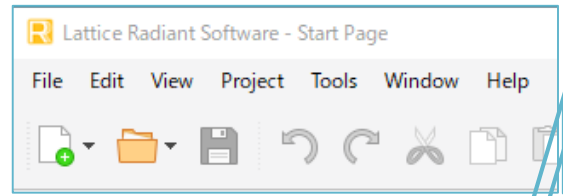
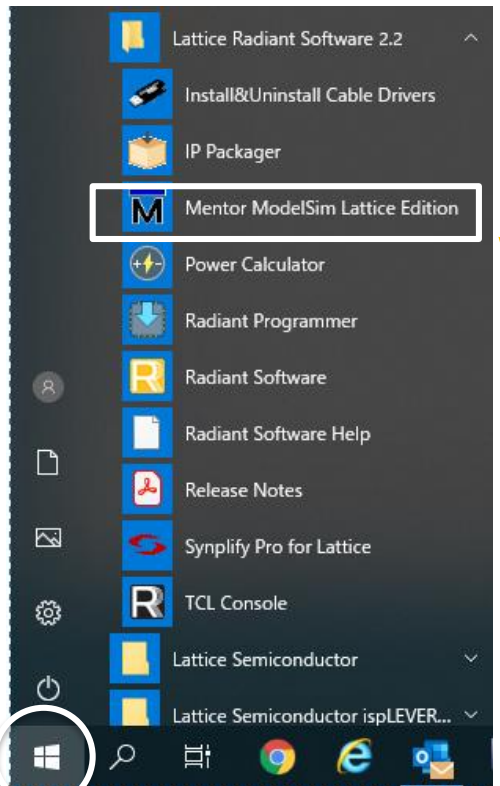


1. ModelSim Lattice Edition の起動	p. 4
2. 作業手順の概要	p. 6
3. 手順詳細	
ステップ 1 作業ディレクトリへの移動	p. 7
ステップ 2 プロジェクトの作成	p. 8
ステップ 3 RTL ソースファイルのインポート	p. 9
ステップ 4 RTL ソースファイルのコンパイル	p. 11
ステップ 5 シミュレーションの初期化	p. 13
ステップ 6 観測する信号の指定	p. 15
ステップ 7 シミュレーションの実行	p. 20
4. 補足事項	
補足 1 シミュレーションの再実行	p. 21
補足 2 波形表示 Wave 窓のデタッチ	p. 23
補足 3 波形表示の信号をグループ化	p. 24
補足 4 波形表示の信号属性を指定	p. 25
補足 5 波形表示の信号リスト書き出し	p. 26
補足 6 作業の終了	p. 27
補足 7 プロジェクトの再オープン	p. 28
補足 8 コマンド履歴の参照・書き出し	p. 29
5. 付加情報	p. 30
付加情報 1 ライブラリ名	付加情報 2 アイコン名称
付加情報 3 (参考) ModelSim と Active-HDL との動作的差分について	
変更履歴	p. 33



1. ModelSim LE GUI の起動 (Radiant 2.2~)

- 一般のアプリケーションと同様に ModelSim LE GUI の単独起動方法は複数あります
 - Windows スタートメニューで **Mentor ModelSim Lattice Edition** を選択 (①)、又は
 - Radiant を起動後、トップのアイコン列からクリック (②)
 - Radiant を起動後、何らかのプロジェクトをオープンした状態で：
 - ② **M** アイコンをクリック、または
 - ③ **Tools** メニュー → **ModelSim Lattice-Edition** を選択



①

②

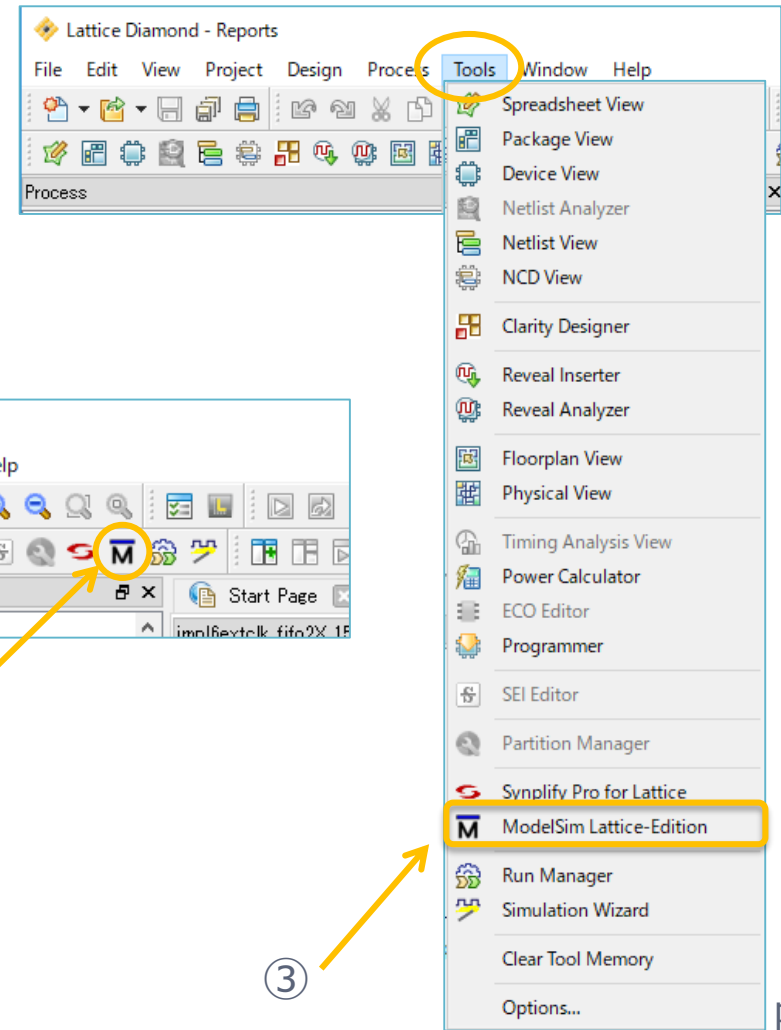
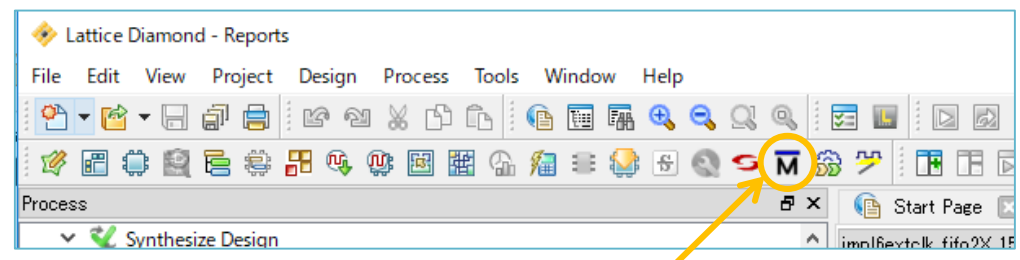
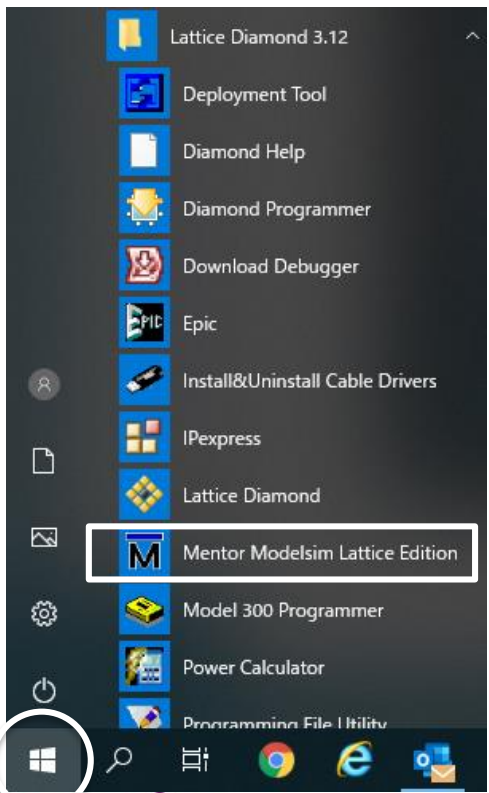
③

④



1. ModelSim LE GUI の起動 (Diamond 3.12)

- 一般のアプリケーションと同様に ModelSim LE GUI の単独起動方法は複数あります
 - Windows スタートメニューで **Mentor ModelSim Lattice Edition** を選択 (①)、又は
 - Diamond を起動後は、何らかのプロジェクトをオープンした状態で：
 - ② **M** アイコンをクリック、または
 - ③ **Tools** メニュー → **ModelSim Lattice-Edition** を選択





2. 作業手順の概要

- 新規にシミュレーションを実行するまでのステップ概要は以下のとおりです（次ページ以降に詳細）
 1. 作業ディレクトリに移動
 - ✓ 4～5 ページで示した複数の GUI 起動方法に準じて、起動直後のディレクトリ位置は異なります
 2. プロジェクトの作成
 - ✓ ModelSim シミュレーション・プロジェクトの作成です
 3. RTL ソースファイルのインポート
 - ✓ RTL は別途テキストエディタで全て作成済として説明します
 4. RTL ソースファイルのコンパイル
 - ✓ 文法エラーなどはこのステップで検出されます
 - ✓ VHDL でユーザ定義パッケージがある場合は、当該ソースを最初にコンパイルするように順序変更します
 5. シミュレーションの初期化
 - ✓ トップモジュール（エンティティ）やターゲットデバイスのライブラリを指定します
 6. 観測する信号の指定
 - ✓ シミュレーションで確認する表示波形の設定をします
 7. シミュレーションの実行



3. ステップ^o1 : 作業ディレクトリへの移動

3.1 - 1 : File → Change Directory... の選択から始めるのが第一の方法です

- 『フォルダーの選択』画面でブラウザして所望のディレクトリを選択します
- ☞ 事前に作業フォルダを作成しておくことをお勧めします

3.1 - 2 : File → Recent Directory → <dir リスト> から選択するのが第二の方法です

- 既に作業した Sim. プロジェクトがあればリストに候補が表示されます
- ☞ 移動後、Transcript 枠で意図するディレクトリであることが確認できます

☞ dir 変更は Transcript 窓で次のコマンドをタイプすることと等価です
 ModelSim > `cd C:/xxx/yyy/zzz`

第二の方法

第一の方法

クリック操作でブラウザして移動します

所望のディレクトリに移動後クリックして終了します

移動先が意図するものであることが Transcript 枠で確認できます →

```

Transcript
# Reading pref.tcl
cd C:/lsc/radiant/2.2/bin/nt64
cd C:/macWeb_lsc_UG/sim_work
ModelSim>
  
```

直近に作業したディレクトリのリスト例

- 1 C:/lsc/.../modeltech/win32loem
- 2 C:/usr_ss/LSC_RDs/RD1126ufm/ModelSim_gui
- 3 C:/usr_ss/.../chkADC/ModelSim_simwiz
- 4 C:/lsc/.../modeltech/win32loem
- 5 C:/usr_ss/LSC_RDs/RD1126ufm/ModelSim_simwiz
- 6 C:/usr_ss/Rd22works/chkTSMAC/ModelSim_GUI
- 7 C:/usr_ss/Rd22works/chkTSMAC/ModelSim_simwiz
- 8 C:/macWeb_lsc_UG/sim_work

MACNICA ModelSim_LE GUI JUG (rev.0.4, p. 7)

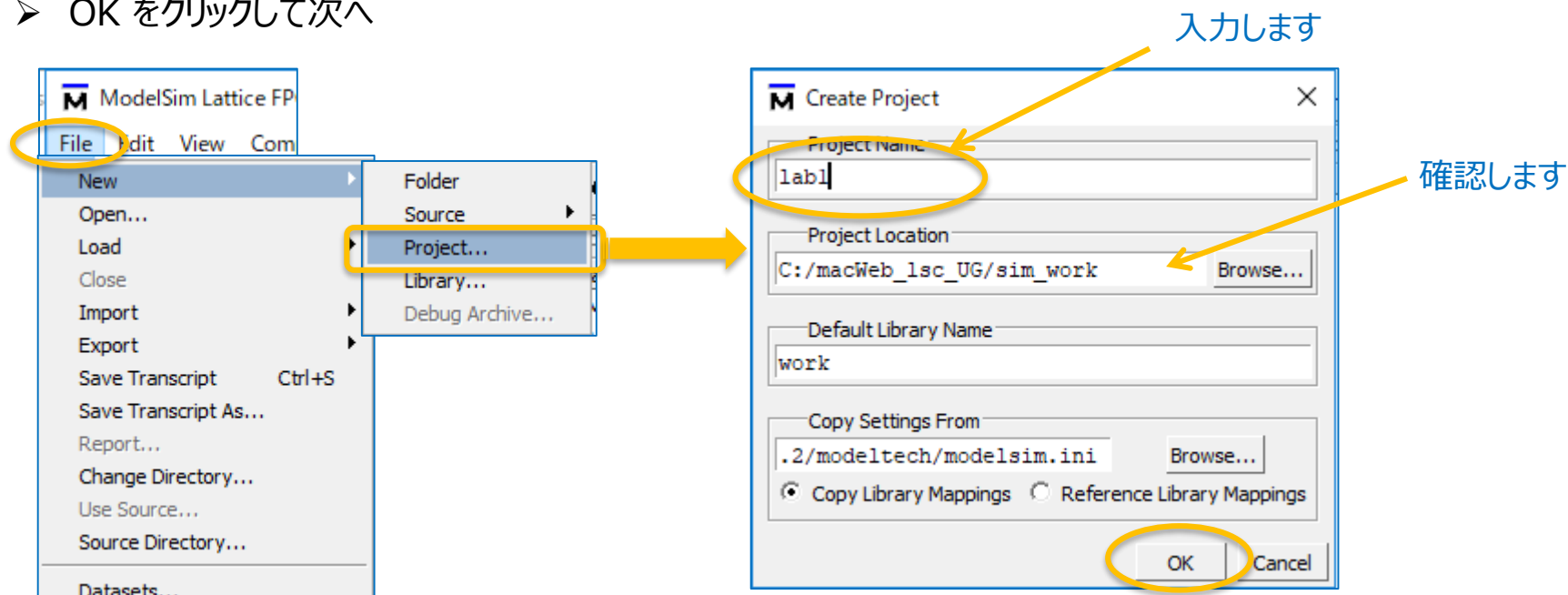
3. ステップ^o2 : プロジェクトの作成

3.2 – 1 : File → New → Project... を選択します

👉 アイコン  は “New File” であり、これをクリックしてもプロジェクトは作成できません

3.2 – 2 : Create Project 窓 Project Name 欄にプロジェクト名を入力します

- Project Location のフォルダが意図するディレクトリであることを確認します
- デフォルトのシミュレーション・ライブラリ名は常時 work です（敢えて変更しなくても良い）
- OK をクリックして次へ



3. ステップ^o3 : RTL ソースファイルのインポート

3.3 - 1 : プロジェクト作成すると自動的に表示される “Add items to the Project” 窓で作業します

- ① “Add Existing File” をクリック
- ② “Add file to Project” 窓で “Browse...” をクリック
- ③ ファイル・ブラウザで所望のファイルを選択し “開く” をクリック
- ④ “Add file to Project” で “OK” をクリック
- ⑤ 全てのファイル指定が終わるまで①～④を繰り返し、終了したら “Close” をクリック

3.3 - 2 : 上の作業を一旦終了後に、ファイルの変更や追加をする必要がある場合は次ページへ

①

②

③

④

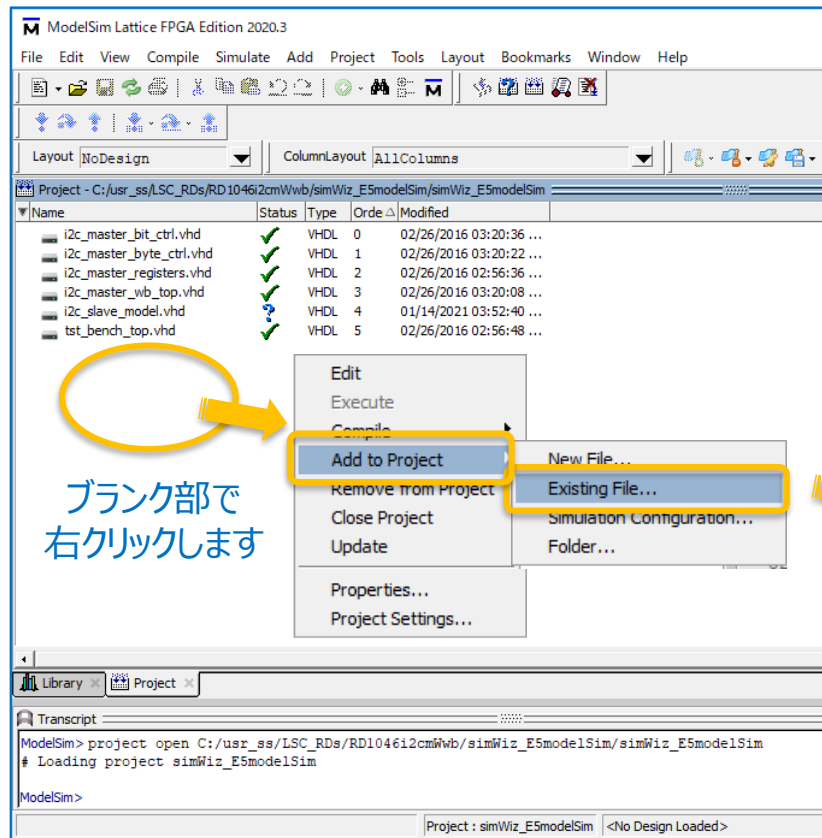
ファイルを選択して『開く』をクリックします
(複数選択可)

全てのファイル指定が
終わるまで繰り返します

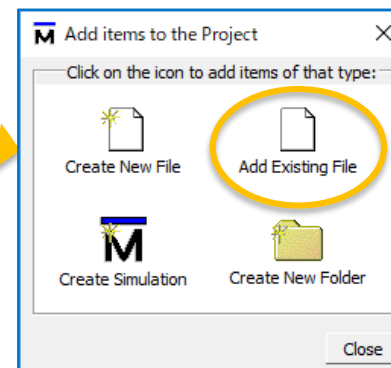
3. ステップ^o3 : RTL ソースファイルのインポート (つづき)

3.3 - 2 : 新たにファイルを追加する必要がある場合の手順は以下のとおりです

- ① Project 枠の空白領域で右クリックします
- ② 表示メニュー Add to Project → Existing File... を選択します
- ③ “Add items to the Project” 窓が表示されるので、以下前ページの要領でファイルを追加します
- ④ 全て終了したら “Close” をクリックします



Verilog ソースファイル内で相対パス記述している `include 対象のファイルがある場合は、当該ファイルをカレント・ディレクトリ (* .mpf があるディレクトリ) にコピーしておく必要があります



前ページと同じ操作手順で追加ファイルをインポートします



3. ステップ^o4 : RTL ソースファイルのコンパイル

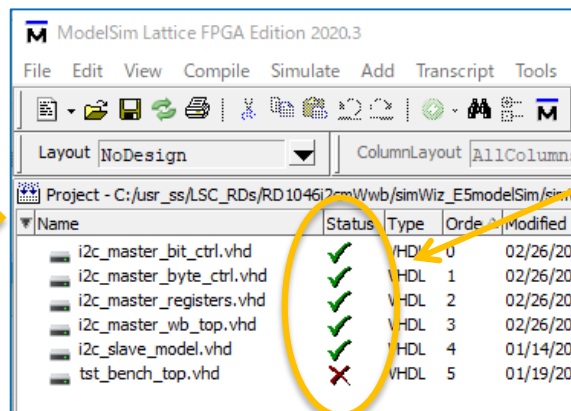
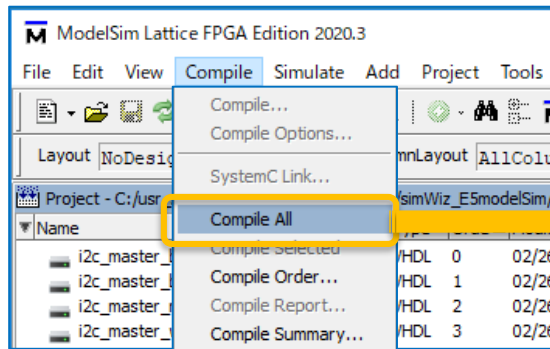
3.4 - 1 : アイコン をクリック、又はメニューバーで File → Compile All を選択します

3.4 - 2 : コンパイル結果が Status 列に規定アイコンで表示されます (右下)

➤ ステータスのファイルを単独コンパイルするには、アイコン をクリックします

☞ 当該ファイルを選択してからメニューバーで Compile → Compile Selected を選択しても同じです

➤ コンパイルエラーがある場合、Transcript 枠にその詳細はレポートされないので、赤字行をダブルクリックします



Status アイコン



コンパイル成功

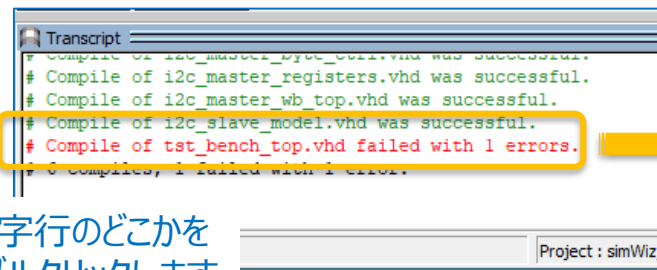


未コンパイル



コンパイルエラー

コンパイルエラーがあると Transcript 枠に赤字のメッセージが出ます



赤字行のどこかをダブルクリックします



エラー箇所と理由の詳細がレポートされます

3. ステップ 4 : RTL ソースファイルのコンパイル (VHDL)

- VHDL ベースのプロジェクトで、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指定します

- ☞ 期待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
- ☞ モジュール階層関係について、下位から上位へのコンパイル順とすることは任意です

- ① メニューバーで 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て、Compile > Compile All を実行します

コンパイル順序に問題ありエラーのケース例

Name	Status	Type	Orde	Modified
ufm_wb_top.vhd	✗	VHDL	0	01/21/2015 05:45:
EFB_UFM.vhd	✓	VHDL	1	01/21/2015 05:45:
USR_MEM.vhd	✓	VHDL	2	01/21/2015 05:45:
ufm_wb_tb.vhd	✓	VHDL	3	01/21/2015 05:45:
efb_define_def.vhd	✓	VHDL	4	01/21/2015 05:45:

2. 対象ファイルを選択し、右矢印をクリックして順序変更後 OK をクリックします


3. 再コンパイル結果 OK

Name	Status	Type	Orde	Mod
efb_define_def.vhd	✓	VHDL	0	01
EFB_UFM.vhd	✓	VHDL	1	01
USR_MEM.vhd	✓	VHDL	2	01
ufm_wb_top.vhd	✓	VHDL	3	01
ufm_wb_tb.vhd	✓	VHDL	4	01

1. 選択します



3. ステップ^o5 : シミュレーションの初期化 (1)

3.5 - 1 : アイコン  をクリック、又はメニューバーで Simulate → Start Simulation... を選択します

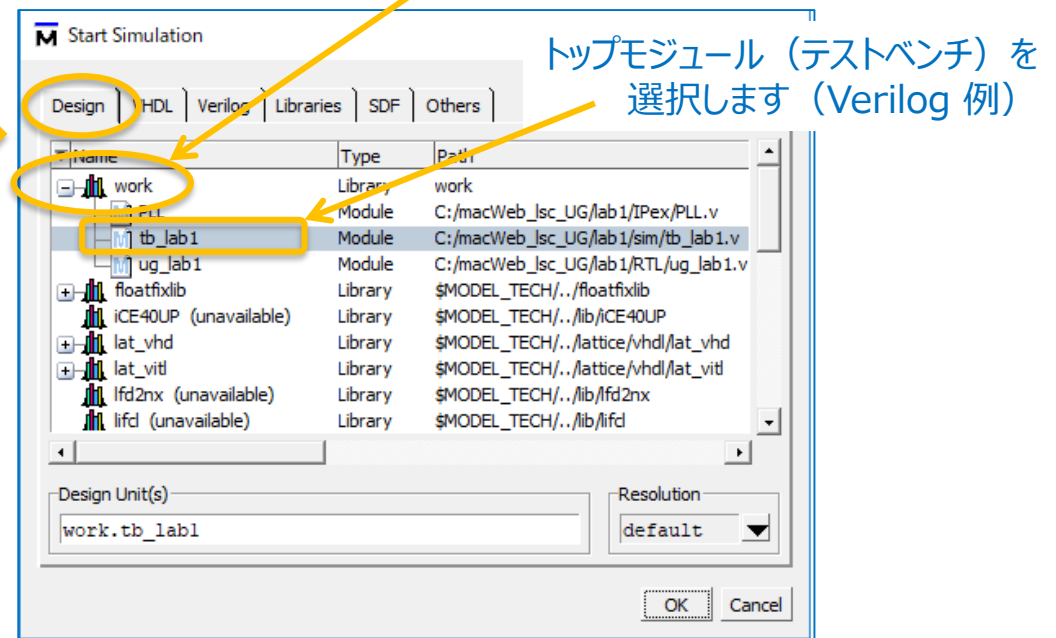
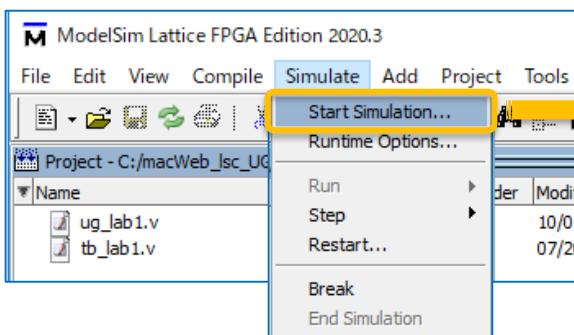
3.5 - 2 : “Start Simulation” 窓の “Design” タブで作業プロジェクトのライブラリ名 (デフォルトは work) を展開し、トップモジュール (= テストベンチ名) を選択します

☞ Verilog でトップモジュールの Type = Module

☞ VHDL でトップモジュールの Type = Entity

3.5 - 3 : (次ページへ)

“work” 左端の  を展開します



“Start Simulation” 窓が立ち上がらない場合があります。Radiant から ModelSim LE を起動した場合で Diamond サポートのデバイスのマクロが RTL 記述にある、あるいは その逆、ということがないかどうか、ご確認ください

3. ステップ^o5 : シミュレーションの初期化 (2)

3.5 – 3 : “Start Simulation” 窓の “Libraries” タブでターゲットデバイスのライブラリを指定します

☞ 一度シミュレーション実行した後など、“Search Libraries” にライブラリが選択済みの場合は確認のみです

① “Search Library” 部右側の “Add...” をクリックします

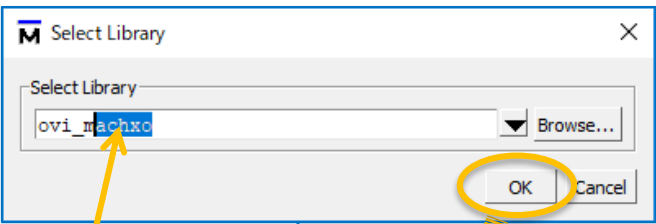
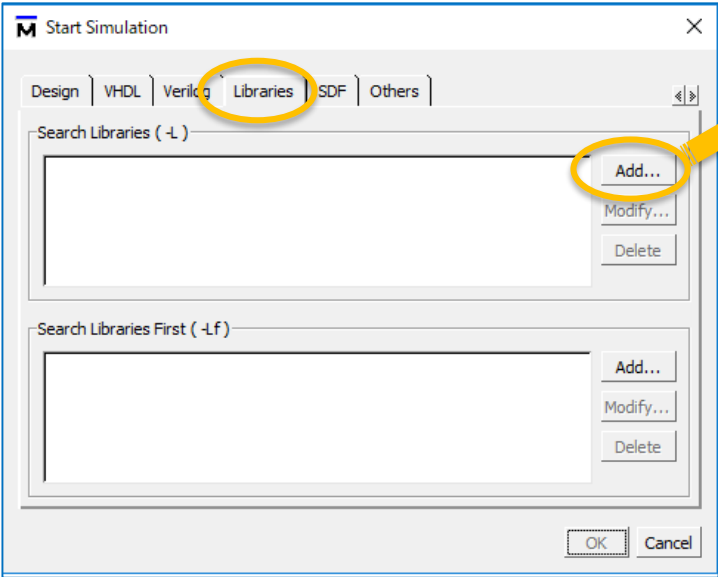
② “Select Library” 窓でライブラリ名をタイプして OK をクリックします

☞ ライブラリ名は事前に “Design” タブ内のリストをチェックし、メモしておくことを推奨します (p.30 も参照)

☞ タイプを開始すると、その途中で候補ライブラリ名が文字補間で表示される補助機能あります

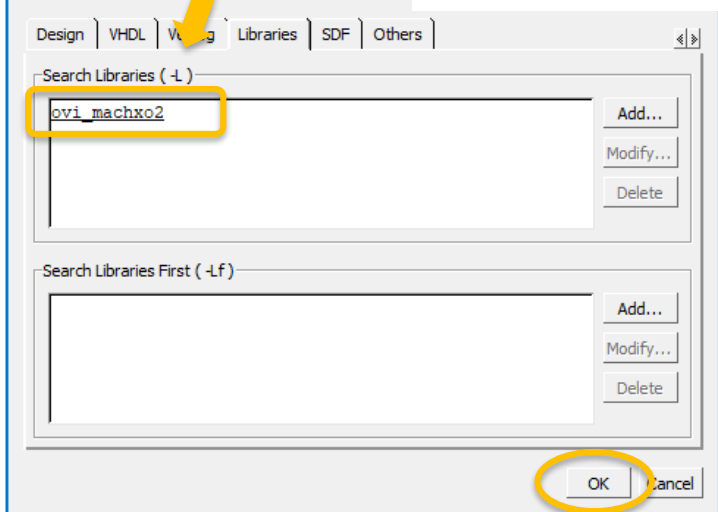
③ “Start Simulation” 窓 “Library” タブ表示に戻り、意図通りであることを確認し OK をクリックします

☞ SERDES やハードマクロ、暗号化 IP (Radiant) を含むデザインでは “pmi_work” ライブラリも必要です



(MachXO2 Verilog
ライブラリの例)

タイプし始めると候補
が補間表示されます

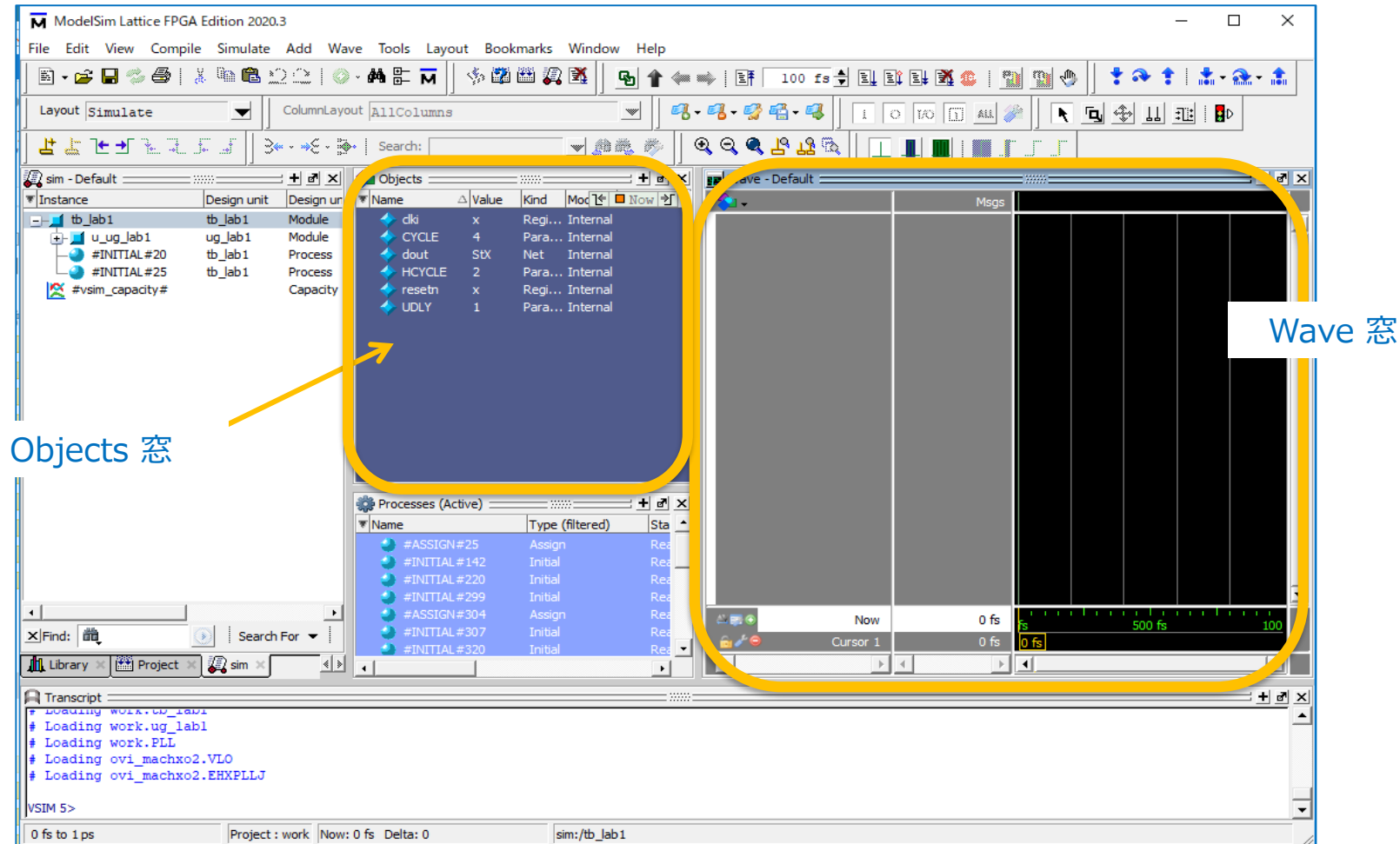


macnica ModelSim_LE GUI JUG (rev.0.4)

3. ステップ^o6 : 観測する信号の指定 (1)

3.6 - 1 : シミュレーション初期化が問題なく実行されると、初期表示は下図のようになります

- ☞ “Objects” 窓が表示されない場合はメニューバーで View → Objects を選択します
- ☞ “Wave” 窓が表示されない場合はメニューバーで View → Wave を選択します

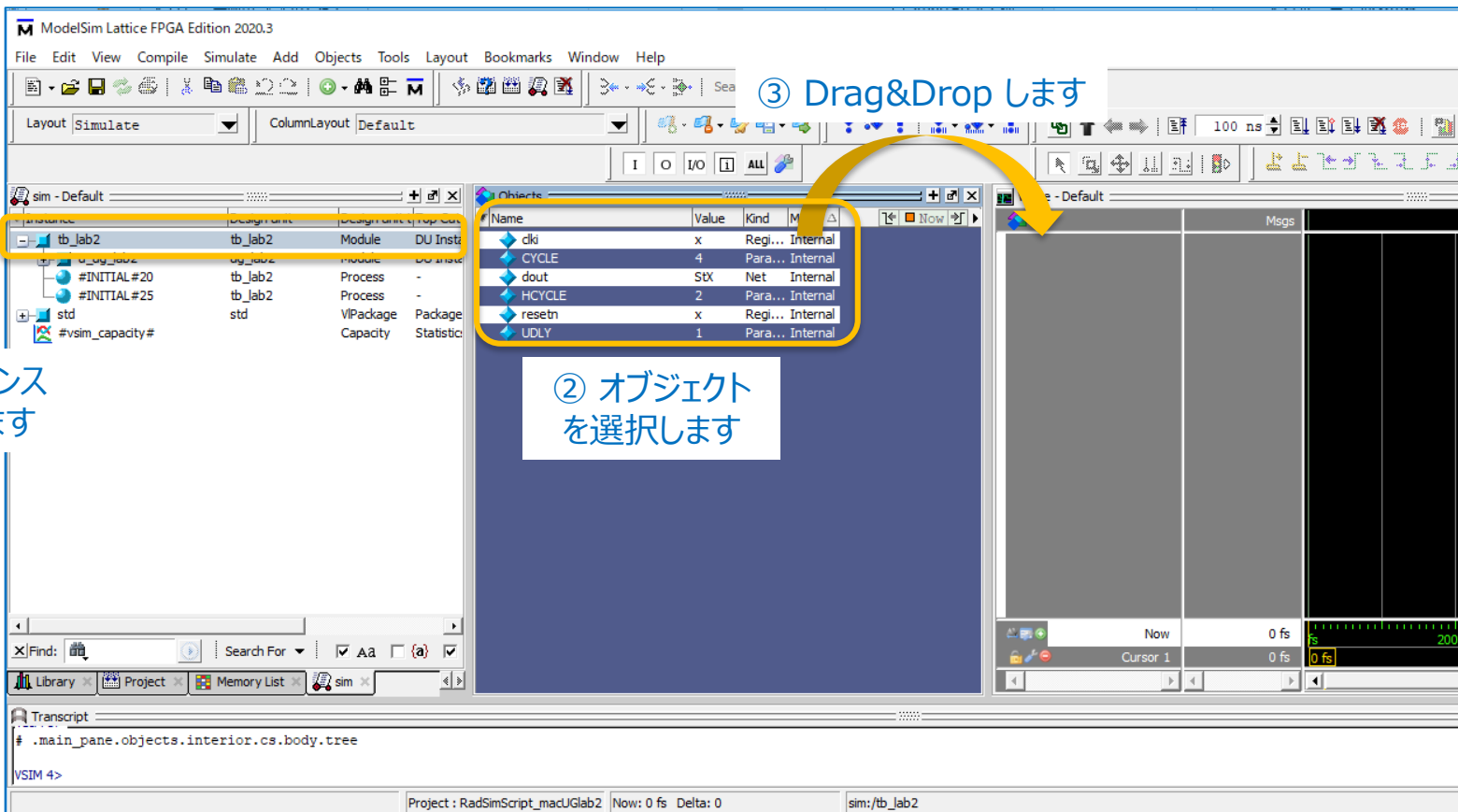




3. ステップ^o6 : 観測する信号の指定 (2-1)

3.6 - 2 (第一の方法) ドラッグ&ドロップ操作による信号表示の指定方法です

- ① “Wave” 窓で観測する信号を含むモジュール (エンティティ) のインスタンスを “Instance” 窓で選択します
- ② 選択されたインスタンスのモジュール記述に含まれる信号 = オブジェクト (ネット、ポート) が “Objects” 窓にリストされますので、観測したい信号 (複数可) を選択します
- ③ “Wave” 窓にドラッグしてドロップします



① インスタンス
を選択します

② オブジェクト
を選択します

③ Drag&Drop します

3. ステップ^o6 : 観測する信号の指定 (2-3)

3.6-2 (第三の方法) メニューバーを用いる方法です

- ① “Wave” 窓で観測する信号を含むモジュール (エンティティ) のインスタンスを “Instance” 窓で選択します (前頁と同様)
- ② 選択された行の上で右クリックして Add to → Wave → All Items in region を選択します
 - ☞ 選択モジュール (アーキテクチャ) 階層にある信号 (reg / wire / signal / port) が**全て**挿入されます
 - ☞ “All Items...” の他の二つのいずれかを選択しても良いですが、やや煩雑になります

① 選択します

② 右クリックして
選択します

当該モジュール (アーキテクチャ) 階層にある
オブジェクト = 信号 (reg / wire / signal
/ port) が全て追加されます

3. ステップ°6：観測する信号の指定（2-4）

3.6 - 2（第四の方法）メニューバーを用いるもう一つの方法です

- ① “Wave” 窓で観測する信号を含むモジュール（エンティティ）のインスタンスを “Instance” 窓で選択します（前頁までと同様）
- ② メニューバーの Add → To Wave → All Items in region を選択します
 - ☞ “All Items...” の他の二つのいずれかを選択しても良いですが、やや煩雑になります

① 選択します

② Add をクリックして選択します


Wave - Default

	Msgs		
◆ /tb_lab1/dki	x		
◆ /tb_lab1/resetn	x		
◆ /tb_lab1/dout	StX		

当該モジュール（アーキテクチャ）階層にあるオブジェクト=信号（reg / wire / signal / port）が全て追加されます

3. ステップ^o7 : シミュレーションの実行

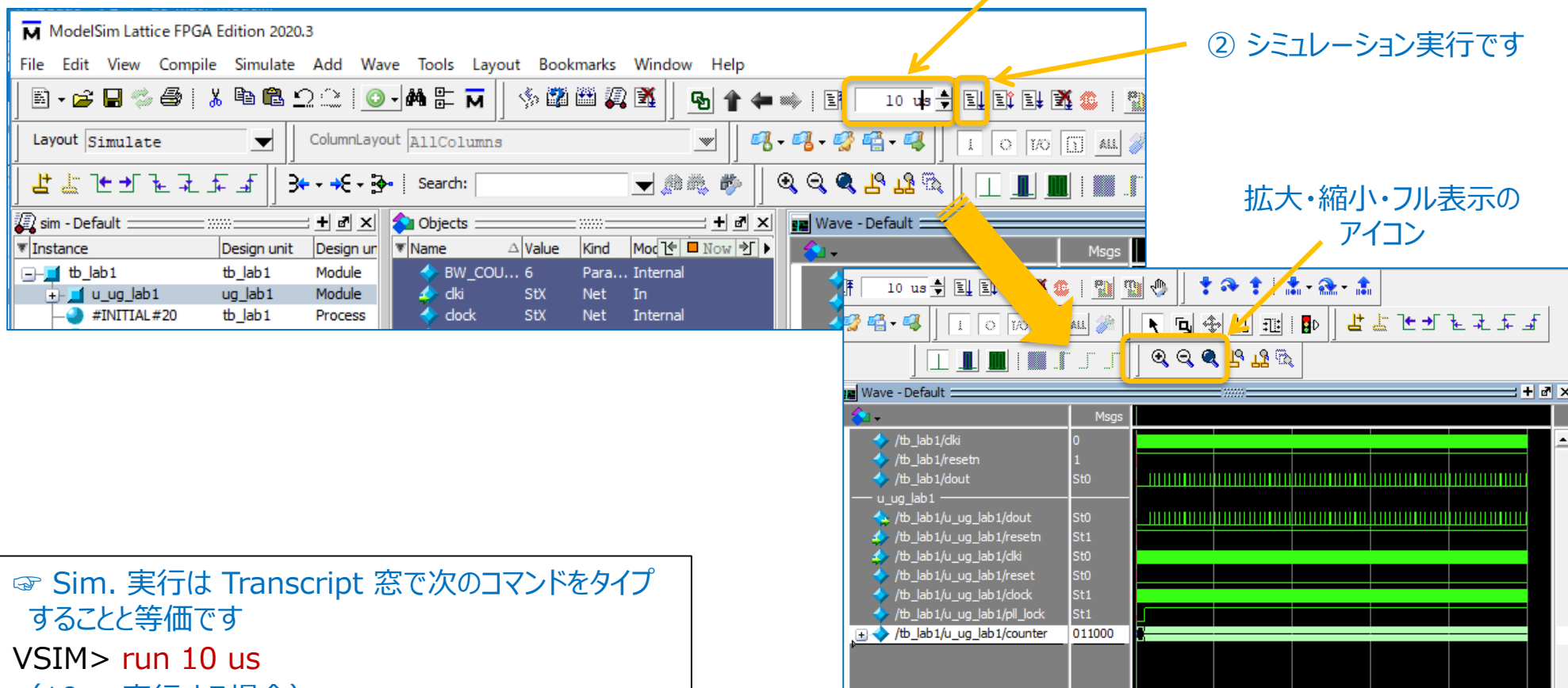
3.7 - 1 : 全て準備ができたのでシミュレーション実行します

- ① ニューバーの下にある “Run Length” セルに実行時間を単位と共に入力します
- ② その隣にある  アイコンをクリックして実行を開始します

① “Run Length” セルに実行時間を指定します

② シミュレーション実行です

拡大・縮小・フル表示の
アイコン




The screenshot shows the ModelSim Lattice FPGA Edition 2020.3 interface. The top menu bar includes File, Edit, View, Compile, Simulate, Add, Wave, Tools, Layout, Bookmarks, Window, and Help. The toolbar contains various icons for simulation control. The Run Length field is set to 10 us. The Run button is highlighted with a yellow box. The Wave window shows a timing diagram with green traces.

👉 Sim. 実行は Transcript 窓で次のコマンドをタイプ
することと等価です
VSIM> run 10 us
(10us 実行する場合)

4. 補足 1 : シミュレーションの再実行 (1)

■ RTL ソースは一切編集・変更せず、観測する信号を追加・削除するのみの場合は以下です

① まず Wave 窓に観測・表示する信号 (順序、追加・削除、属性など) を更新します

② アイコン  をクリックするか、又はメニューバーの Simulate → Restart を選択します (下図左)

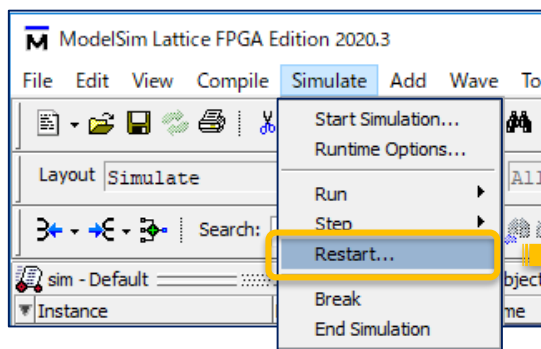
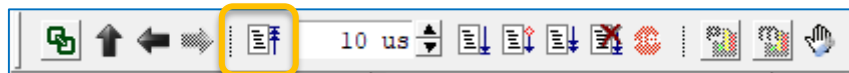
② “Restart” ウィンドウがポップアップ (下図中) しますので、OK をクリックします

👉 シミュレーションが初期化されます (p.15 のシミュレーション実行前の表示に戻ります)

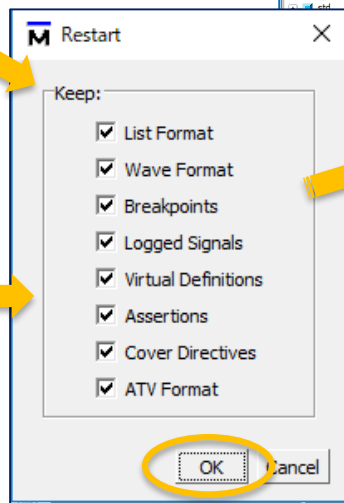
👉 Transcript 窓の次コマンドと等価です : VSIM> restart -f -nowave (波形表示が同じなら restart のみ)

③ ステップ 3.7-1 (p.20) の手順でシミュレーション実行します

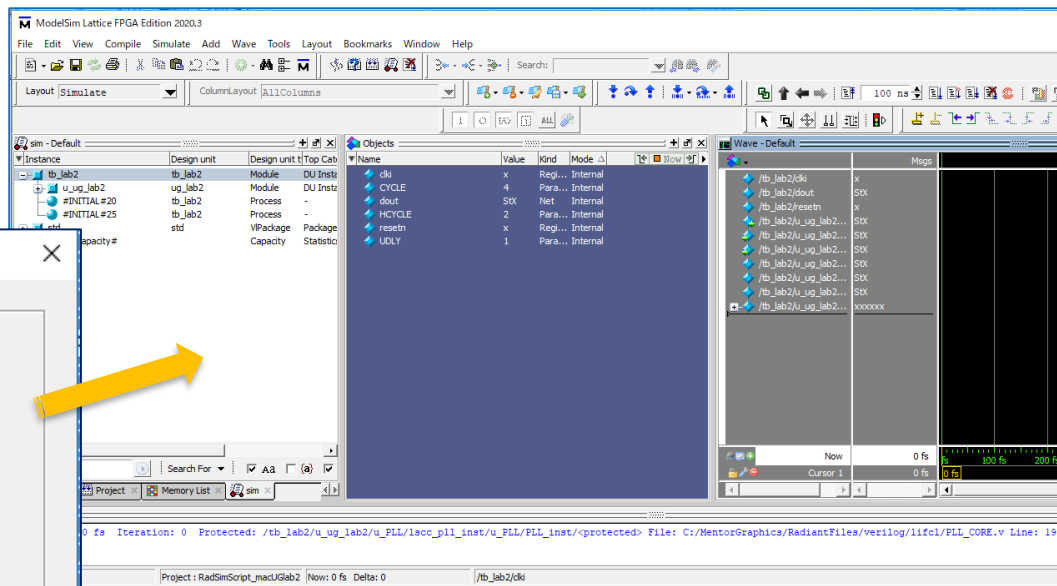
② Restart アイコンをクリック、または ↓



② Restart メニューを選択



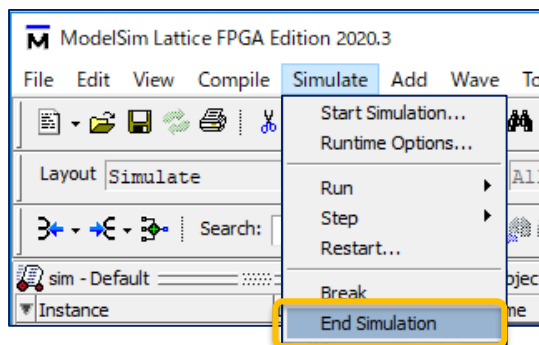
← Restart 窓
(OK をクリック)



シミュレーション実行前
の状態に戻ります



4. 補足 1 : シミュレーションの再実行 (2)

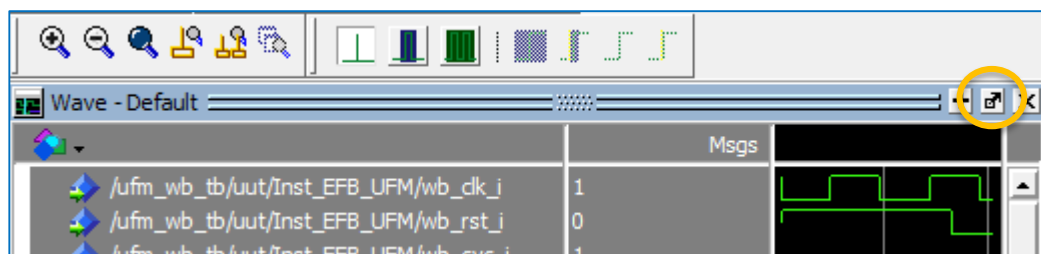
- RTL ソースのいずれかを編集したり、ファイルを追加・削除など変更した場合は以下です
 - ① 一旦シミュレーションを終了します (p.10 の状態に戻ります。等価コマンドは p.27)
 - ✓ メニューバーで Simulate → End Simulation を選択します (下図左)
 - ② ステップ 3.3 からステップ 3.7 までを実行します
 - ✓ RTL インポート / 削除 / 編集 → コンパイル → シミュレーション初期化 → 表示波形設定 → Sim. 実行



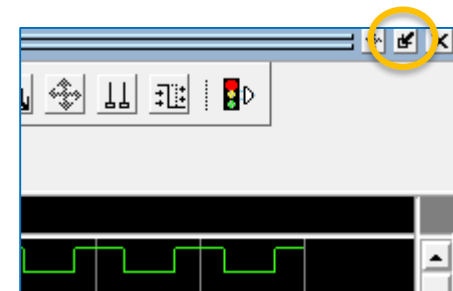
① Sim. 終了操作

4. 補足 2 : 波形表示 Wave 窓のデタッチ

- タイリング表示されている Wave 窓をデタッチして単独表示することで、波形を見やすくできます
- デタッチ（単独ウィンドウでの表示）：下図左
 - Wave 窓右上隅の  アイコンをクリックします
- アタッチ（元に戻す）：下図右
 - デタッチされた Wave 窓右上隅の  アイコンをクリックします



デタッチ

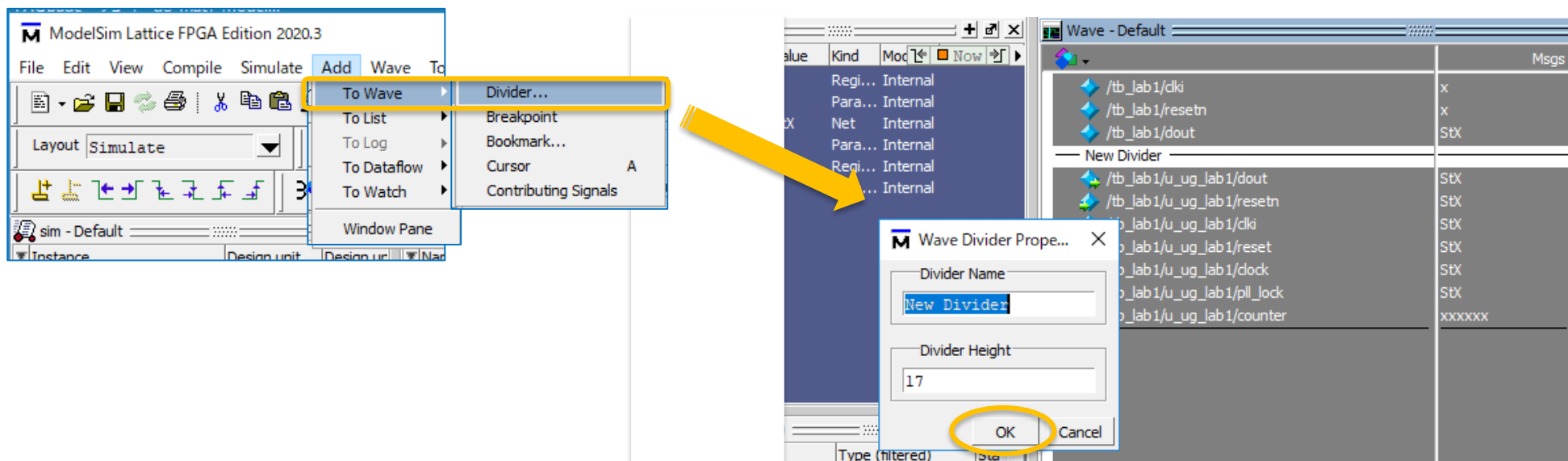


アタッチ



4. 補足3：波形表示の信号をグループ化

- Wave 窓で意図する任意の2信号間に“Divider”を挿入できます
 - ☞ 表示する信号をモジュールや階層ごとに区別することで、見やすくなります
- 挿入手順は以下のとおりです
 - ① Wave 窓で Divider を挿入する位置の信号を選択します
 - ☞ Divider は選択した信号の上に挿入されます
 - ② メニューバー Add → To Wave → Divider を選択します（下図左）
 - ✓ “Wave Divider Prope...” ポップアップ・ウィンドウで、Divider 名称を入力し、OK をクリックします
 - ③ Divider 挿入後に名称を変更する場合は、その行をダブルクリックし、ポップアップ窓で編集します



4. 補足 4 : 波形表示の信号属性を指定

- Wave 窓に表示する信号、特に多ビット/バス信号の属性変更は頻繁に行います

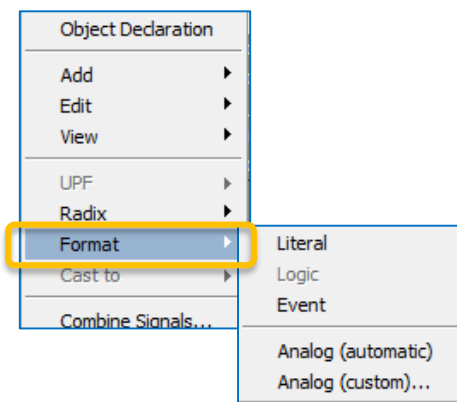
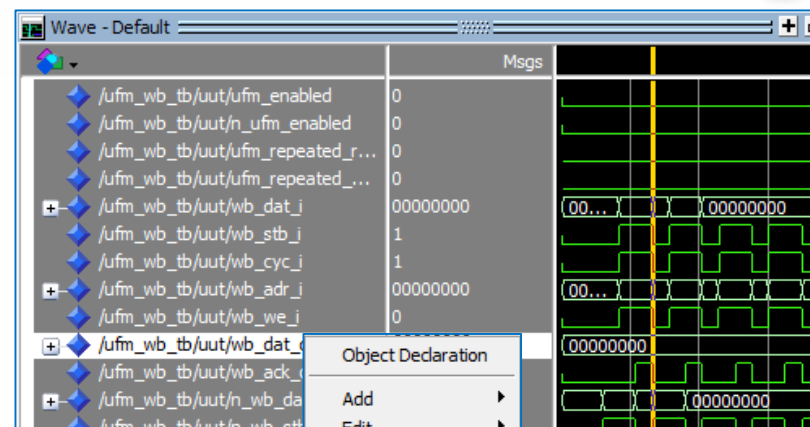
☞ デフォルトはバイナリです

- 変更手順は以下のとおりです

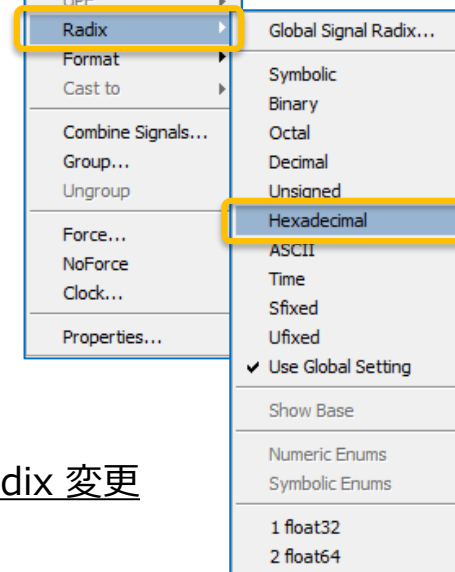
- Wave 窓で属性変更する信号を選択します
- 右クリックして Radix、または Format などを選択します
 - ✓ Radix → Hexadecimal などを指定します

☞ Format 変更の頻度は高くありません

(これ以外にも種々属性指定ができます)



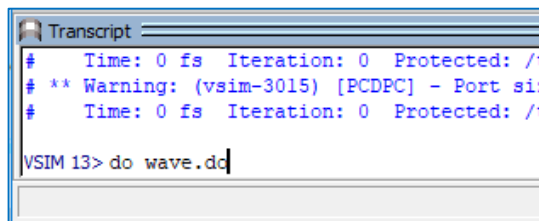
Format 変更



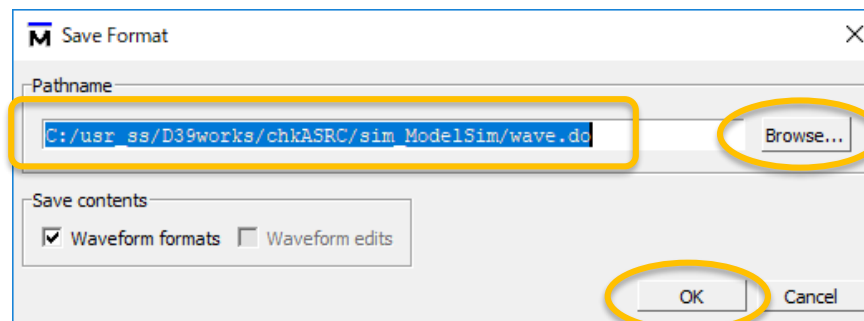
Radix 変更

4. 補足5：波形表示の信号リスト書き出し

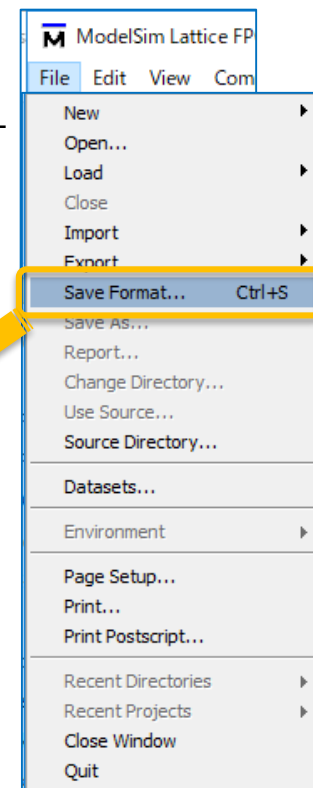
- Wave 窓で表示した信号の順序・リスト・属性をテキストファイル（マクロ）に書き出します
 - ☞ プロジェクトを再オープンしての作業や、RTL ソースを再コンパイルしてシミュレーション初期化した後などに、Wave 窓の表示を復元するために利用できます
- 手順は以下のとおりです
 - ① メニューバーで File → Save Format... を選択します
 - ☞ メニューに “Save Format...” がいない場合は Wave 窓内のどこかを一度クリックします
 - ② “Save Format” ポップアップ・ウィンドウの Pathname 欄にパスとファイル名を入力後 OK をクリックします
 - ☞ デフォルト wave.do が自動的に表示されています（編集可）
 - ☞ 慣例的に拡張子は .do です
 - ③ p.15 シミュレーション初期化後 Transcript 窓で “do wave.do” のようにタイプします
 - ☞ 下左図。波形表示書式がロードされます
(p.15 の Wave 窓の表示がない状態でタイプしても良いです)
 - ☞ メニューバー File → Load → Macro File... でファイル指定しても同じです



波形表示をロード（復元）
するコマンド入力



パスとファイル名を入力して OK をクリック
(必要に応じてブラウズする = 非推奨)





4. 補足6 : 作業の終了

- シミュレーションの終了:
 - メニューバーで Simulation → End Simulation を選択します
 - ☞ Transcript 窓に以下コマンドをタイプしても等価です

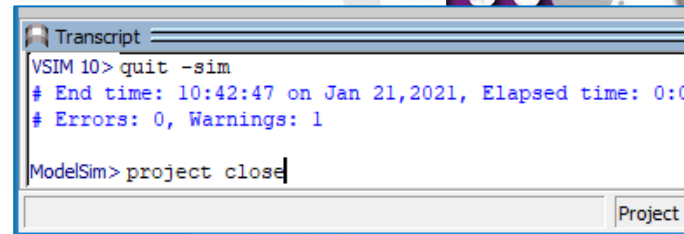
VSIM> quit -sim

- プロジェクトのクローズ:
 - Transcript 窓に以下コマンドをタイプします
 - ☞ メニューバーの File → Close Window では終了しません

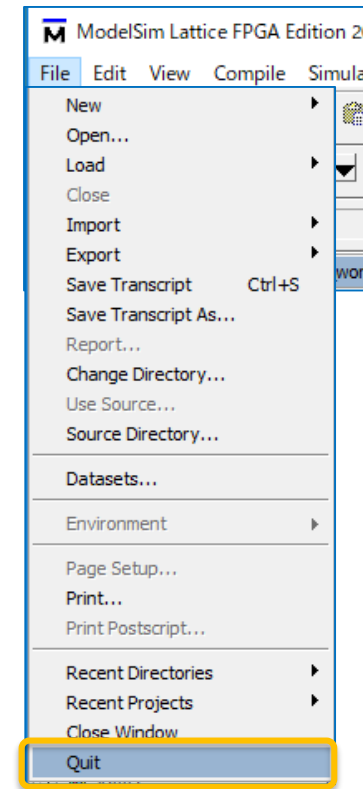
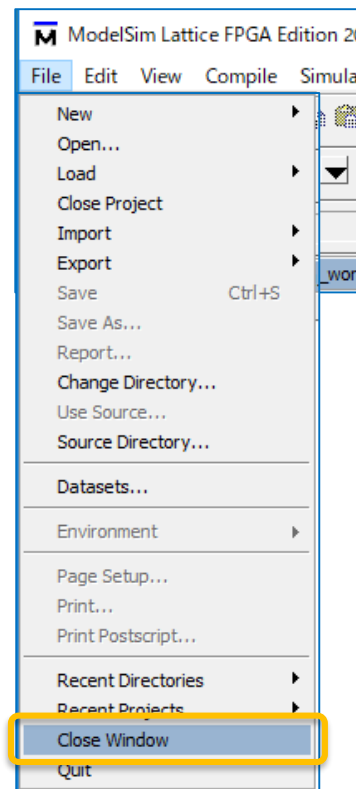
VSIM> project close

- ModelSim LE の終了:
 - メニューバーで File → Quit を選択します
 - ☞ Transcript 窓に以下コマンドをタイプしても等価です

VSIM> quit



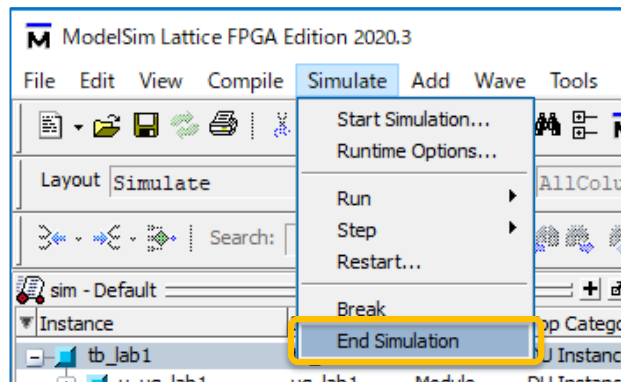
プロジェクトのクローズ



“Close Window”ではプロジェクトはクローズしません


ModelSim LE の終了

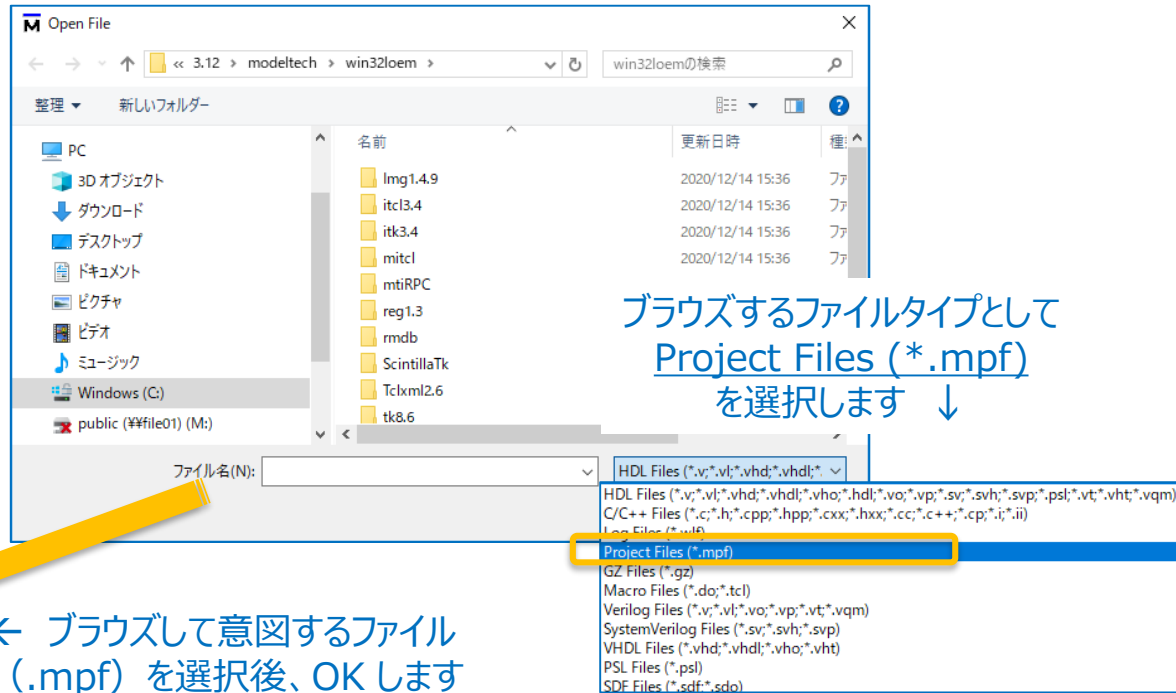
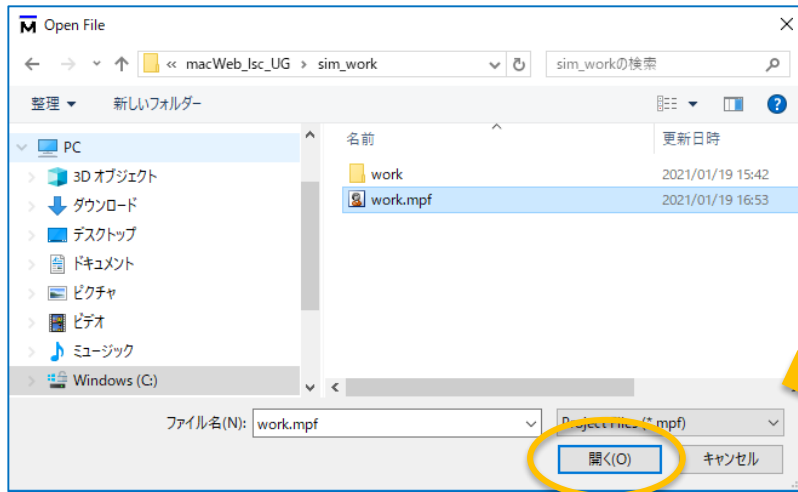
シミュレーションの終了





4. 補足7 : プロジェクトの再オープン

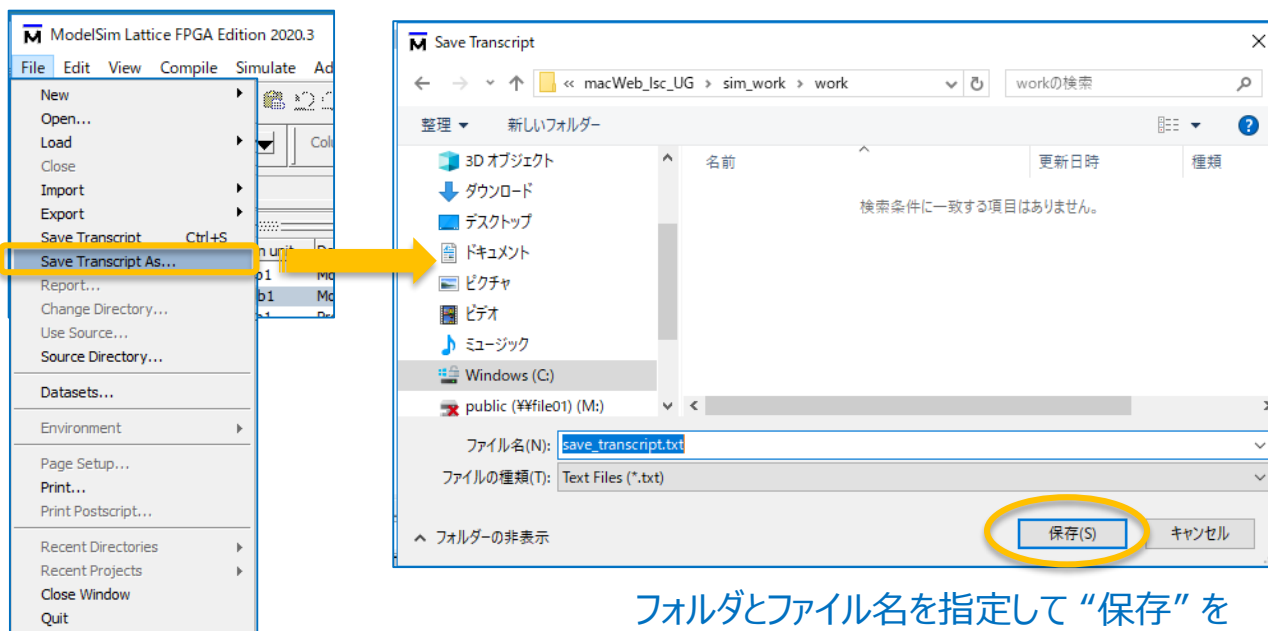
- File → Recent Projects で表示されるプロジェクトから選択します
 - プロジェクト名は Diamond / Radiant 両環境が識別されずに、混在してリストされることにご留意ください
 - ☞ ツールが併存する環境下ではプロジェクト名を工夫し、識別できるようにすることを推奨します
 - プロジェクト・リストにない場合は以下です
 - ✓ メニューバーの File → Open... を選択するか、アイコン  をクリックします
 - ✓ 表示される “Open File” の右下ファイルタイプを “Project Files (*.mpf)” にし、ブラウズ・選択して OK します
- ☞ 作業していたプロジェクトをクローズしないで ModelSim LE を終了した際は、その次に立ち上げると終了直前のプロジェクトを自動的にオープンした状態になることにご留意ください
 - ModelSim 終了前のクローズを推奨します
 - ✓ Transcript 窓に “project close” と入力



4. 補足8 : コマンド履歴の参照・書き出し

- マウスベースの各操作に該当するコマンドを Transcript 窓でチェックできることがあります
(操作と等価なコマンドがその都度必ず表示されるとは限りません)
 - “VSIM x>” の右をクリック後、キーボードの『↑』キーをタイプすると直前の実行コマンドが表示されます
 - ☞ ↑の回数分、コマンド履歴が遡って呼び出され、表示されます
 - ☞ 呼び出されたコマンドに対して Enter キーを打てば、実行できます
- その時点までの Transcript 窓の表示情報をテキストファイルに書き出せます (下図)
 - 書き出すファイル名の入力では、拡張子も指定します
 - ☞ テキストエディタでコマンドのみの抽出が容易です (→ do スクリプト作成時の参照)

保存したファイルの内容例



フォルダとファイル名を指定して“保存”を
クリックします

```
save_transcript.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
# Reading pref.tcl
# Loading project work
vsim -gui work.tb_lab1 -L ovi_machxo2
# vsim -gui work.tb_lab1 -L ovi_machxo2
# Start time: 14:30:26 on Jan 19, 2021
# // ModelSim - Lattice FPGA Edition 2020.3 Oct 14 2020
# //
# // Copyright 1991-2020 Mentor Graphics Corporation
# // All Rights Reserved.
# //
# // ModelSim - Lattice FPGA Edition and its associated c
# // secrets and commercial or financial information that
# // Mentor Graphics Corporation and are privileged, conf
# // and exempt from disclosure under the Freedom of Info
# // 5 U.S.C. Section 552. Furthermore, this information
# // is prohibited from disclosure under the Trade Secret
# // 18 U.S.C. Section 1905.
# //
# Loading work.tb_lab1
# Loading work.ug_lab1
# Loading work.PLL
# Loading ovi_machxo2.VLO
# Loading ovi_machxo2.EHXPLLJ
add wave sim:/tb_lab1/*
add wave sim:/tb_lab1/u_ug_lab1/*
run
```

5 付加情報 1 : ライブラリ名



- Radiant 2.2~ / Diamond 3.12 のサポートする各デバイス・ファミリのライブラリ名は以下のとおりです

Radiant

Family	Verilog	VHDL
iCE40 Ultra Plus	ovi_iCE40UP	iCE40UP
Crosslink-NX	ovi_lifcl	lifcl
Certus-NX	ovi_lfd2nx	lfd2nx

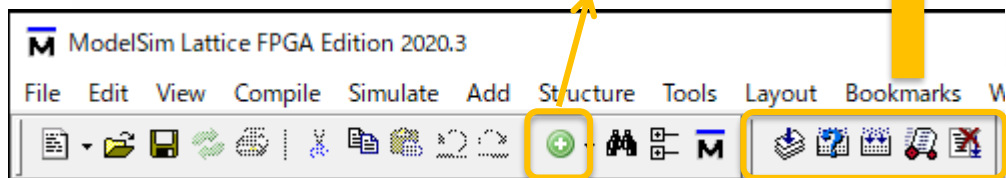
Diamond

Family	Verilog	VHDL
Crosslink	ovi_lifmd	lifmd
Crosslink Plus	ovi_lifmdf	lifmdf
Mach-NX	ovi_lfmnx	lfmnx
MachXO2	ovi_machxo2	machxo2
MachXO3L/LF	ovi_machxo3l	machxo3l
MachXO3D	ovi_machxo3d	machxo3d
ECP5U	ovi_ecp5u	ecp5u
ECP5UM	ovi_ecp5um	ecp5um
LatticeECP3	ovi_ecp3	ecp3



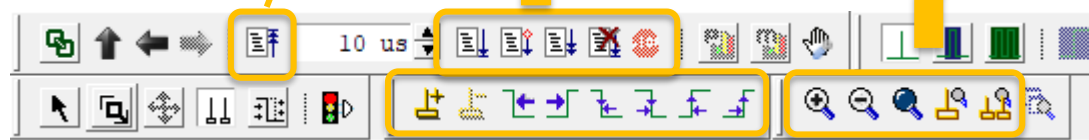
5 付加情報 2 : 各種アイコン

Add Selected to Window



- Compile
- Compile Out of Date
- Compile All
- Simulate
- Break

Restart



- Run
- Continue Run
- Run All
- Break
- Stop

- Zoom In 拡大
- Zoom Out 縮小
- Zoom Full フル表示
- Zoom In on Active Cursor
- Zoom Between Cursors

- Insert Cursor
- Delete Cursor
- Find Previous Transition
- Find Next Transition
- Find Previous Falling Edge
- Find Next Falling Edge
- Find Previous Rising Edge
- Find Next Rising Edge

Revision History



Date	Revision	Page	Change Information	Updated by
2021/1/25	0.2		Draft Version for review	S.S.
2021/1/25	0.3		Another draft Version for review	S.S.
2021/2/1	0.4, 0.5	(many)	Major revision based on the feedback	S.S.