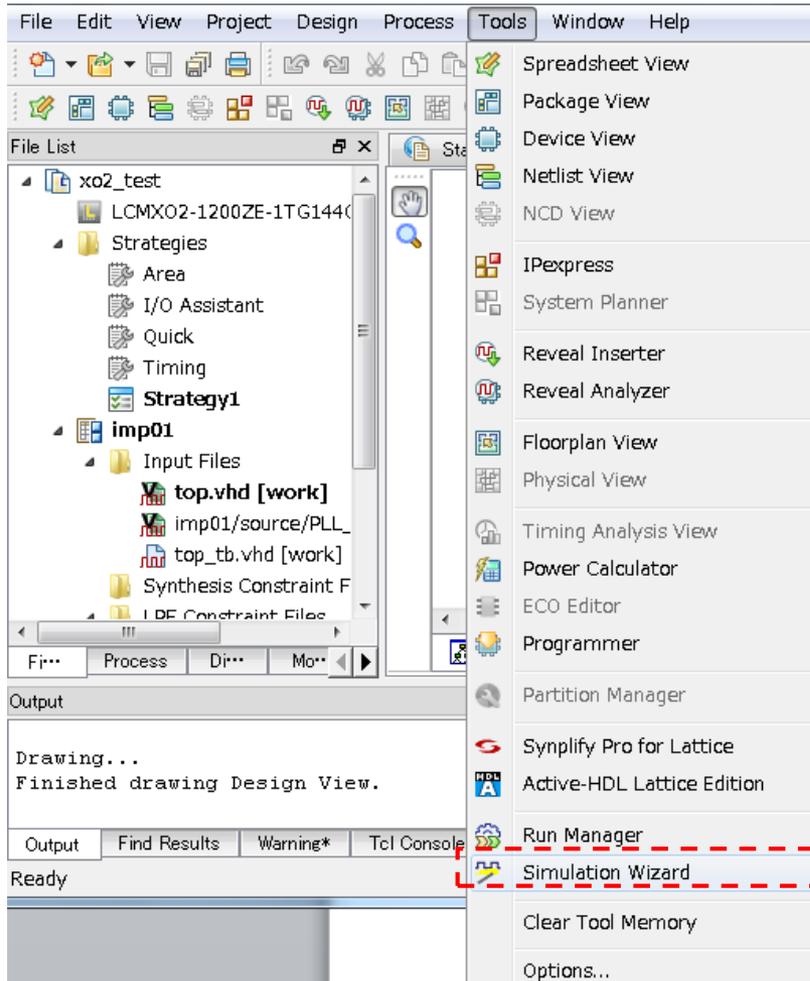


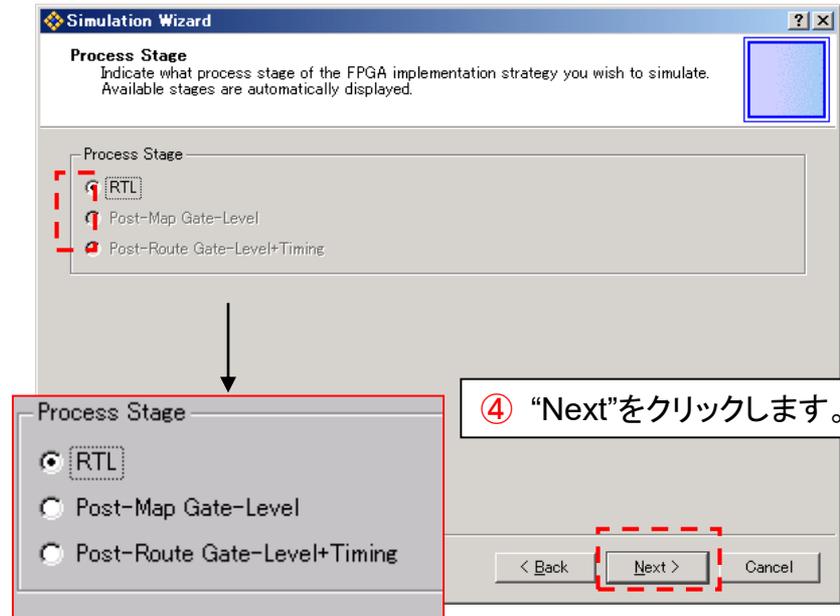
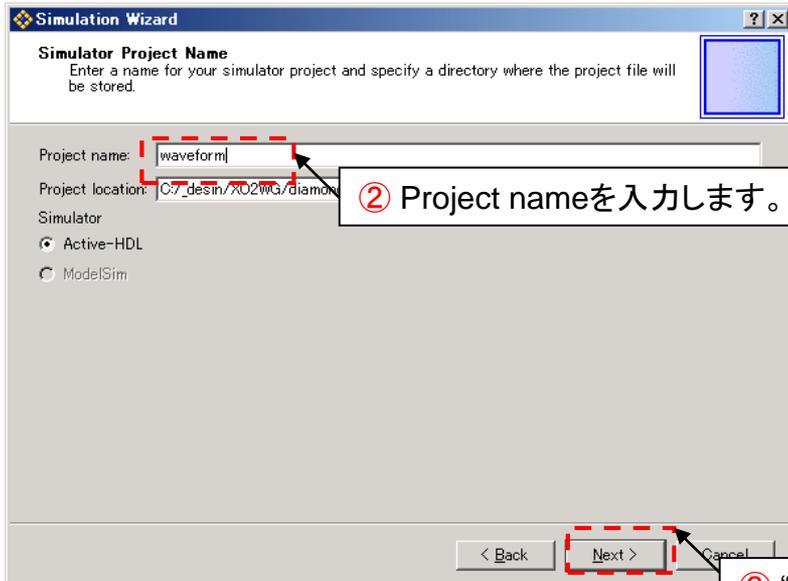
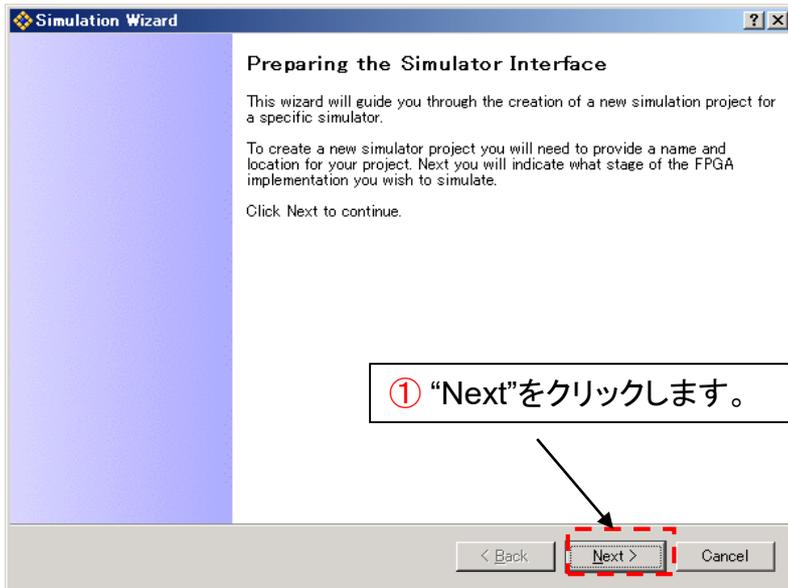
Lattice Diamond 「Active-HDLシミュレーション」

Functionシミュレーション

■ Simulation Wizardを使用し、シミュレーションを実行



①Tools ⇒ Simulation Wizardを選択します。



・RTL

シミュレータがRTLをインポートし、コンパイルを行いシミュレーションを実施します。

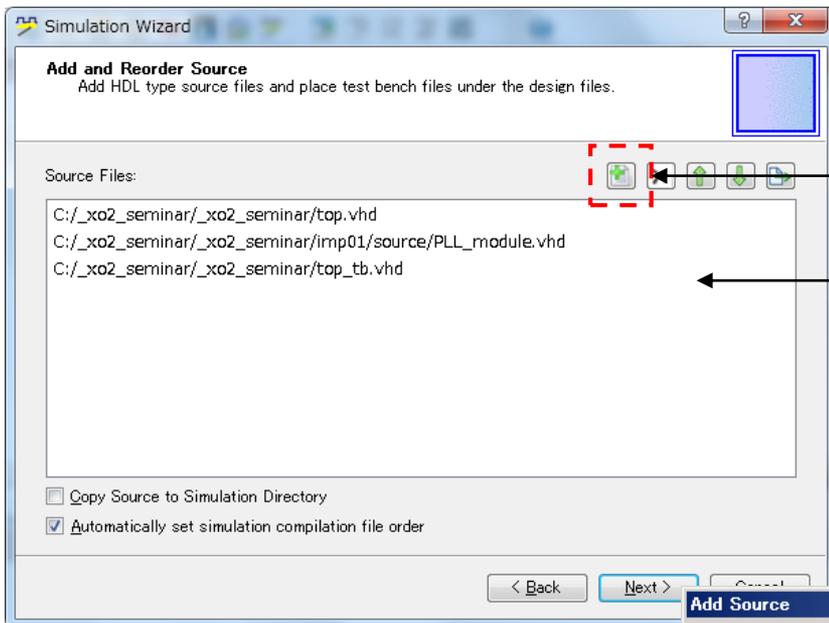
・Post-Map Gate-Level

Mapping後生成されるネットリストファイル(VO,VHOファイル)を使用してシミュレーションを実施します。

・Post-Map Gate-Level+Timing

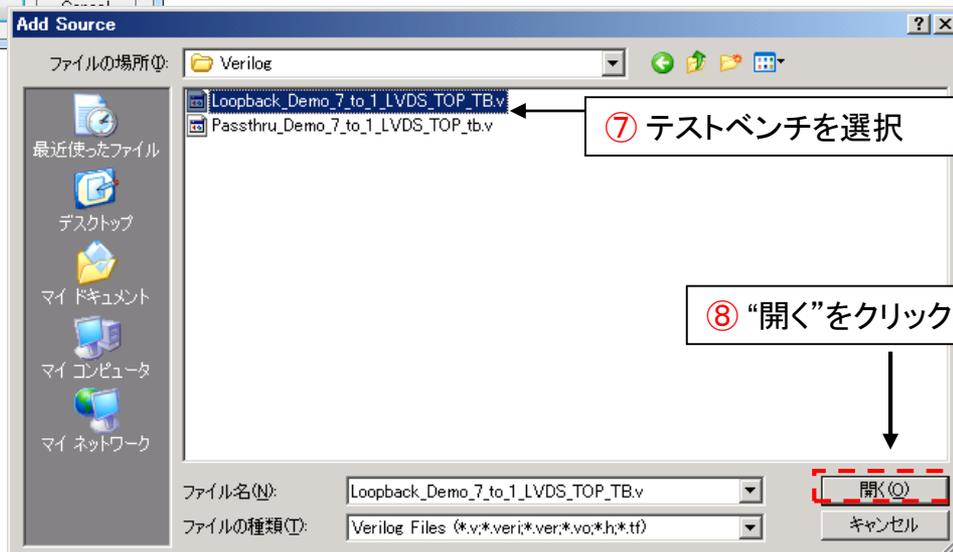
上記ネットリストファイル+遅延情報ファイル(SDFファイル)を取り込んでシミュレーションを行います。実機の動きに近いシミュレーションが可能です。シミュレーションの実行に時間がかかります。

③ “Next”をクリックします。



⑤ ソースを追加する場合はこのアイコンをクリックします。

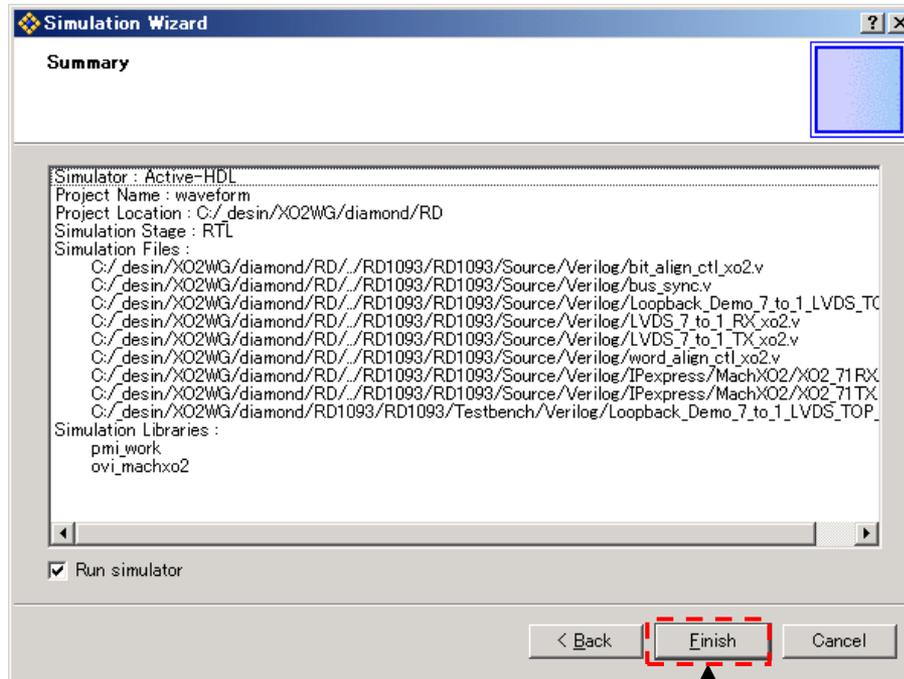
⑥ 全てのデザインソース及びテストベンチが含まれているか確認をします。テストベンチが別のロケーションに含まれる場合は、上記+マークのアイコンをクリックします。



⑦ テストベンチを選択

⑧ “開く”をクリックします。



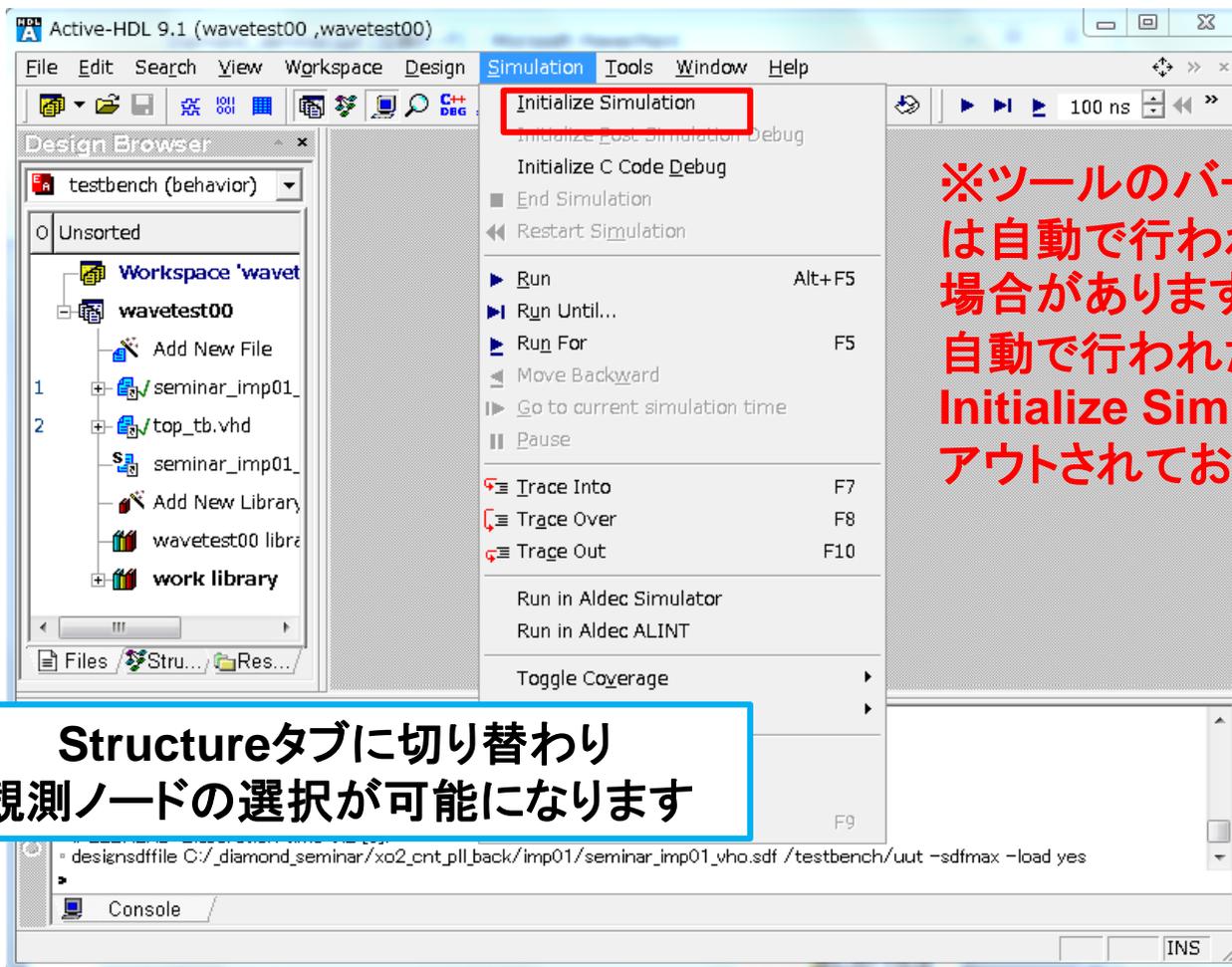


⑨ “Finish”をクリックします。

FinishをクリックするとActive-HDLが自動で立ち上がります。

■Active-HDLを初期化

Simulationメニューより "Initialize Simulation" をクリックします



① “New Waveform”を選択します。

② “testbenchをWaveform Editorにドラッグ & ドロップします。”

※バージョンによっては自動で初期の信号が追加されます。

Active-HDL 9.1 (wavetest00)

File Edit Search View New Workspace Design Simulation Waveform Tools Window Help

Design Browser

testbench (behavior)

- testbench (behavior)
- std.standard
- std.TEXTIO
- ieee.std_logic_116
- vital2000.VITAL_Ti
- ieee.std_logic_arit
- ieee.VITAL_Primiti
- machxo2.global

Name Value

CLK U

untitled.awc

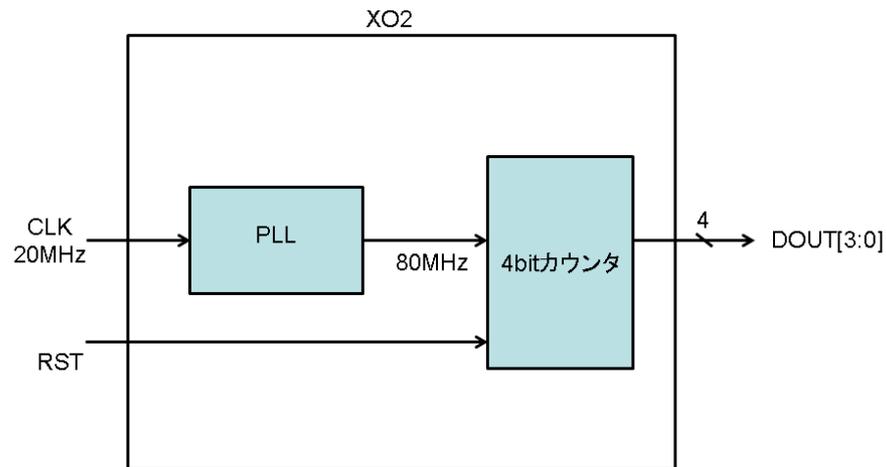
Console

```

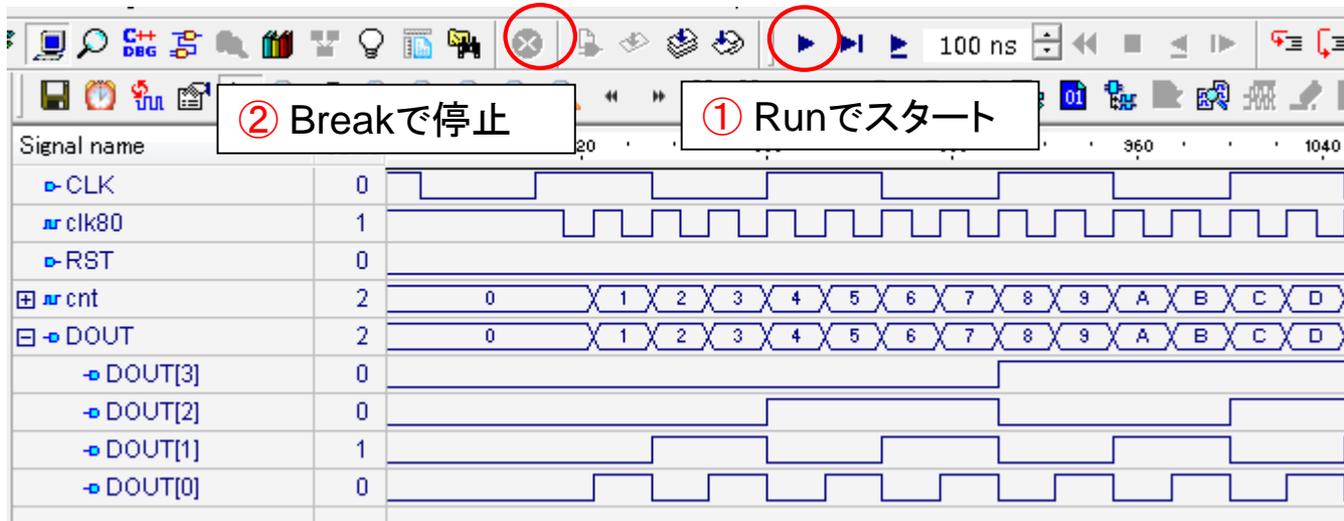
# SDF: SDF load & update time: 0.1 [s].
# SDF: 74 SDF entries loaded while 74 entries read
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 6764 kB (elbread=1024 elab2=5334 kernel=112 sdf=292)
# KERNEL: ASDB file was created in location C:\#_diamond_seminar#xo2_cnt_pll_back#wavetest00#src#wave.asdb
# 17:05, 2013年1月7日
# Simulation has been initialized
# Selected Top-Level: testbench (behavior)
    
```

Shows simulation step for Run For command

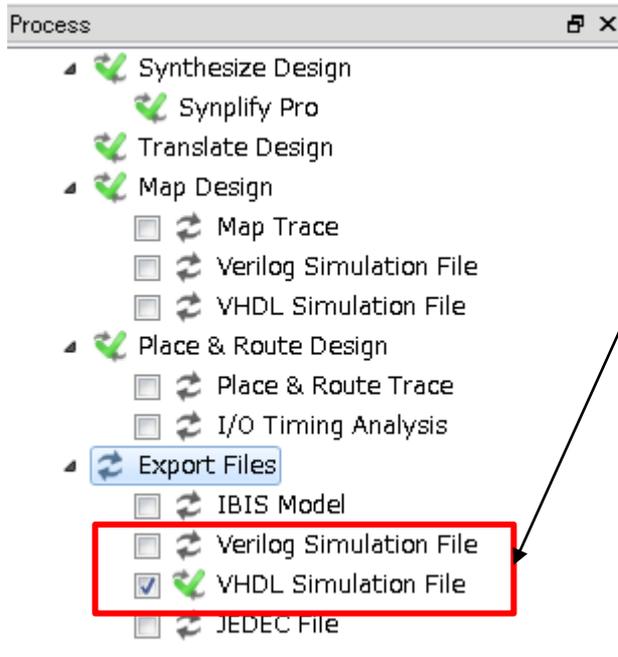
INS



■ シミュレーションの波形



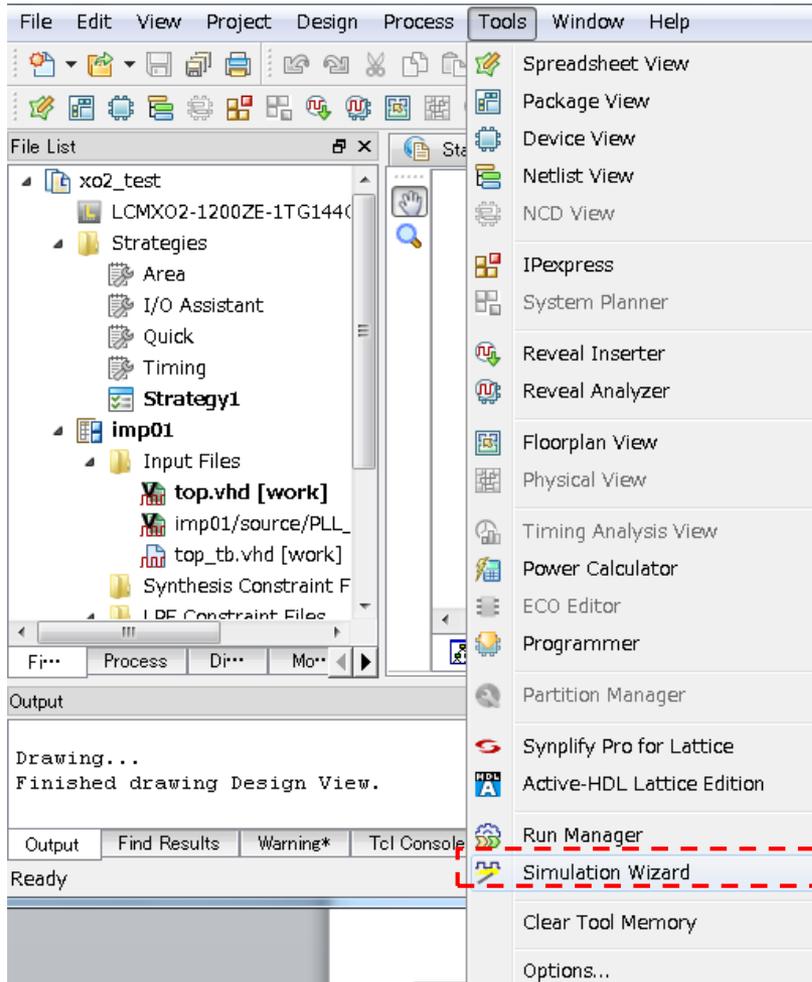
Timingシミュレーション



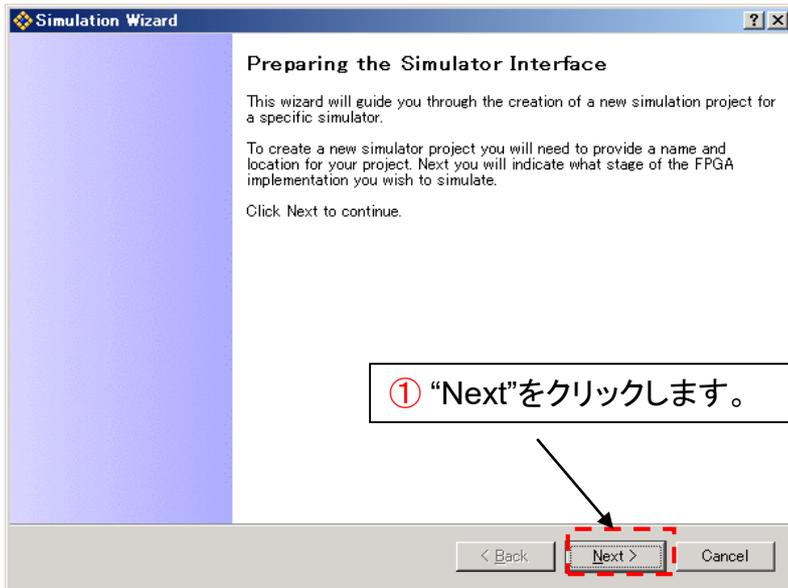
*.sdf 、 *.vhoファイルが生成されます。
タイミングシミュレーションを行う場合には、上記ファイルを使用するため、ここまでシミュレーションを通しておく必要があります。

sdf ファイル : 遅延情報ファイル
vho ファイル : ネットリストファイル
(Verilogの場合には*.vo ファイル)

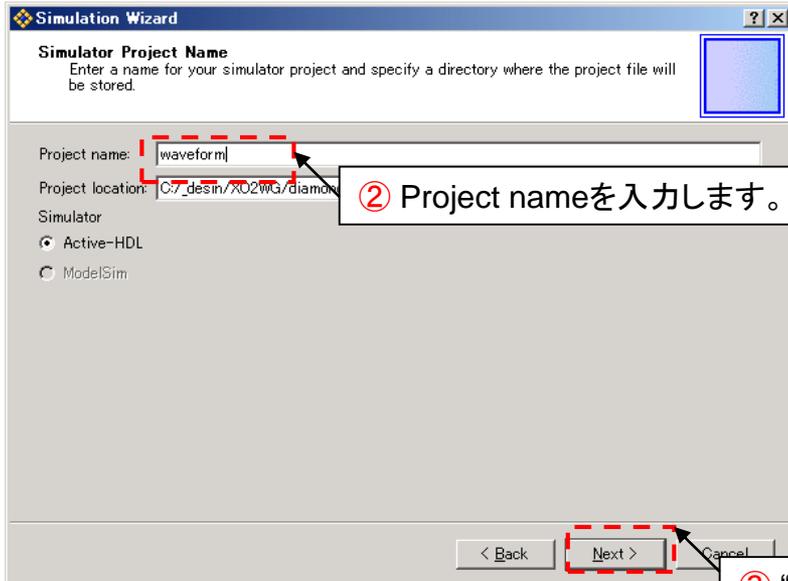
■ Simulation Wizardを使用し、シミュレーションを実行



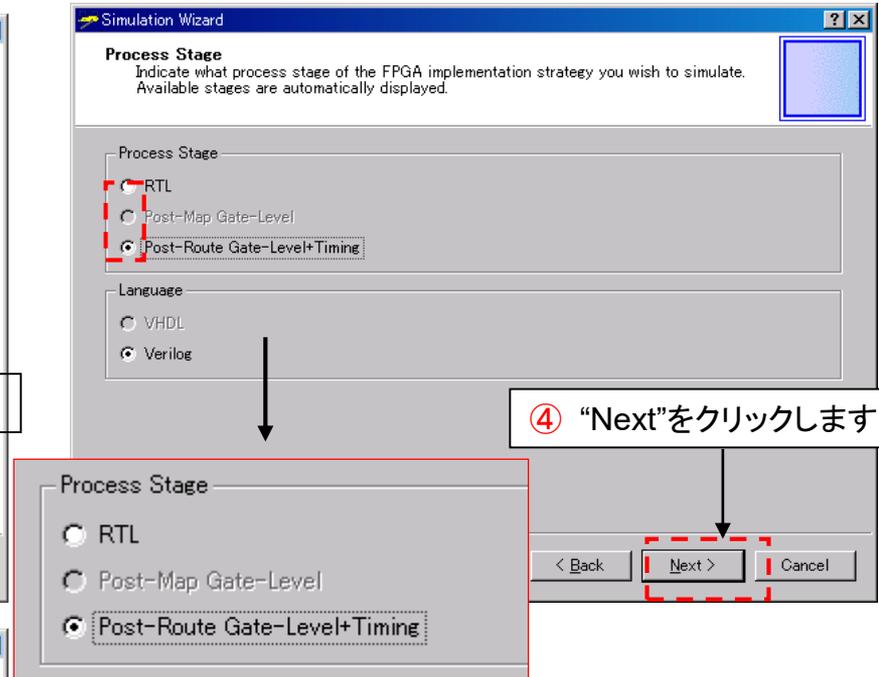
①Tools ⇒ Simulation Wizardを選択します。



① “Next”をクリックします。



② Project nameを入力します。



④ “Next”をクリックします。

・RTL

シミュレータがRTLをインポートし、コンパイルを行いシミュレーションを実施します。

・Post-Map Gate-Level

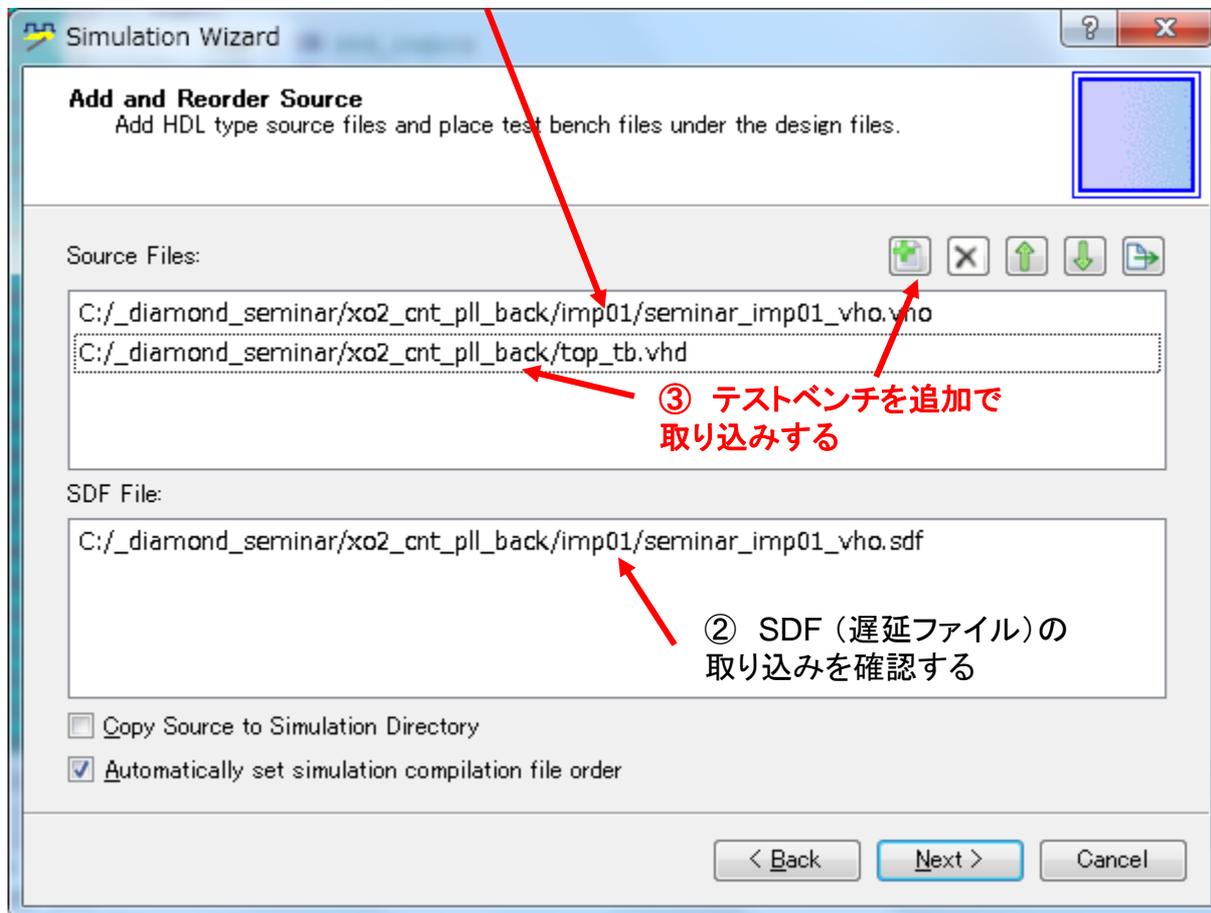
Mapping後生成されるネットリストファイル(VO,VHOファイル)を使用してシミュレーションを実施します。

・Post-Map Gate-Level+Timing

上記ネットリストファイル+遅延情報ファイル(SDFファイル)を取り込んでシミュレーションを行います。実機の動きに近いシミュレーションが可能です。シミュレーションの実行に時間がかかります。

③ “Next”をクリックします。

① vo, vho ネットリストの
取り込みを確認する

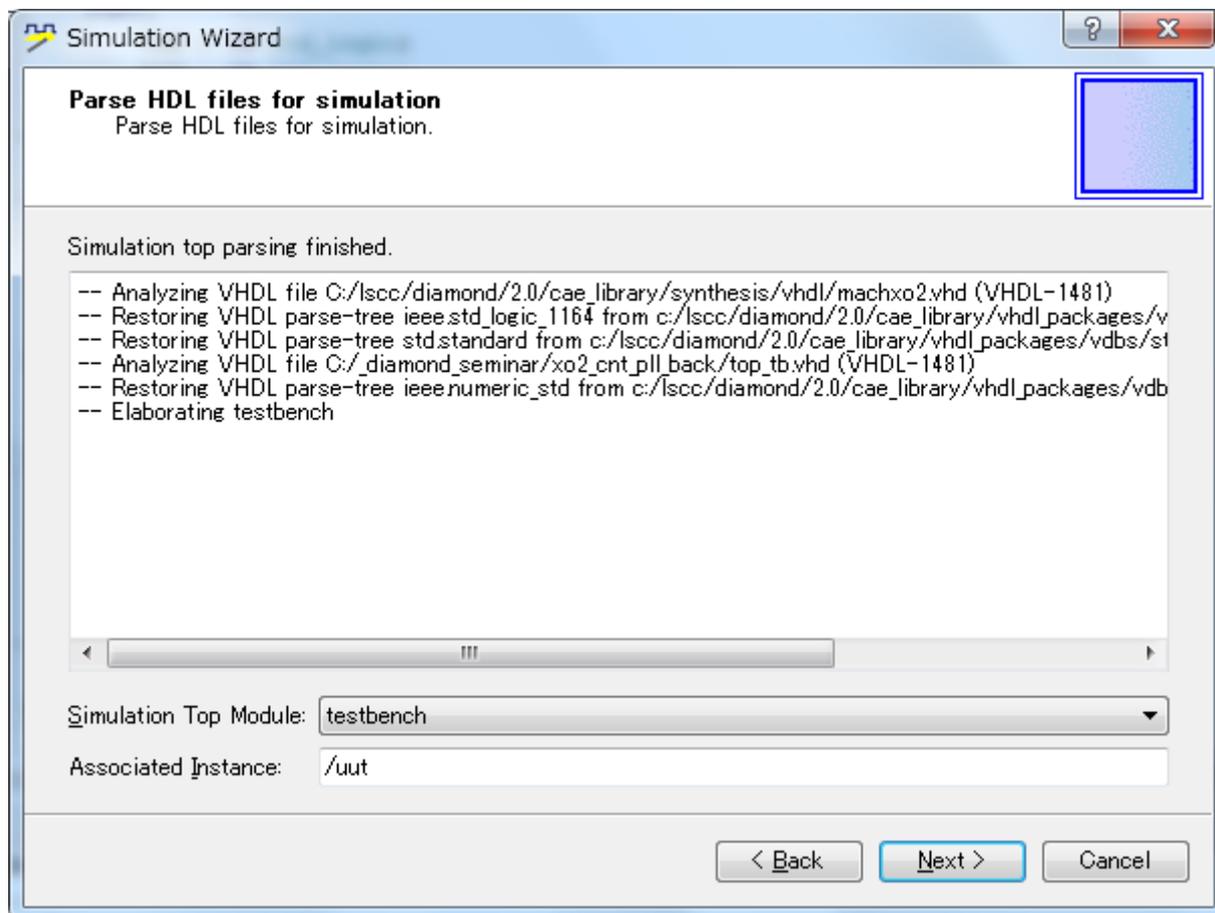


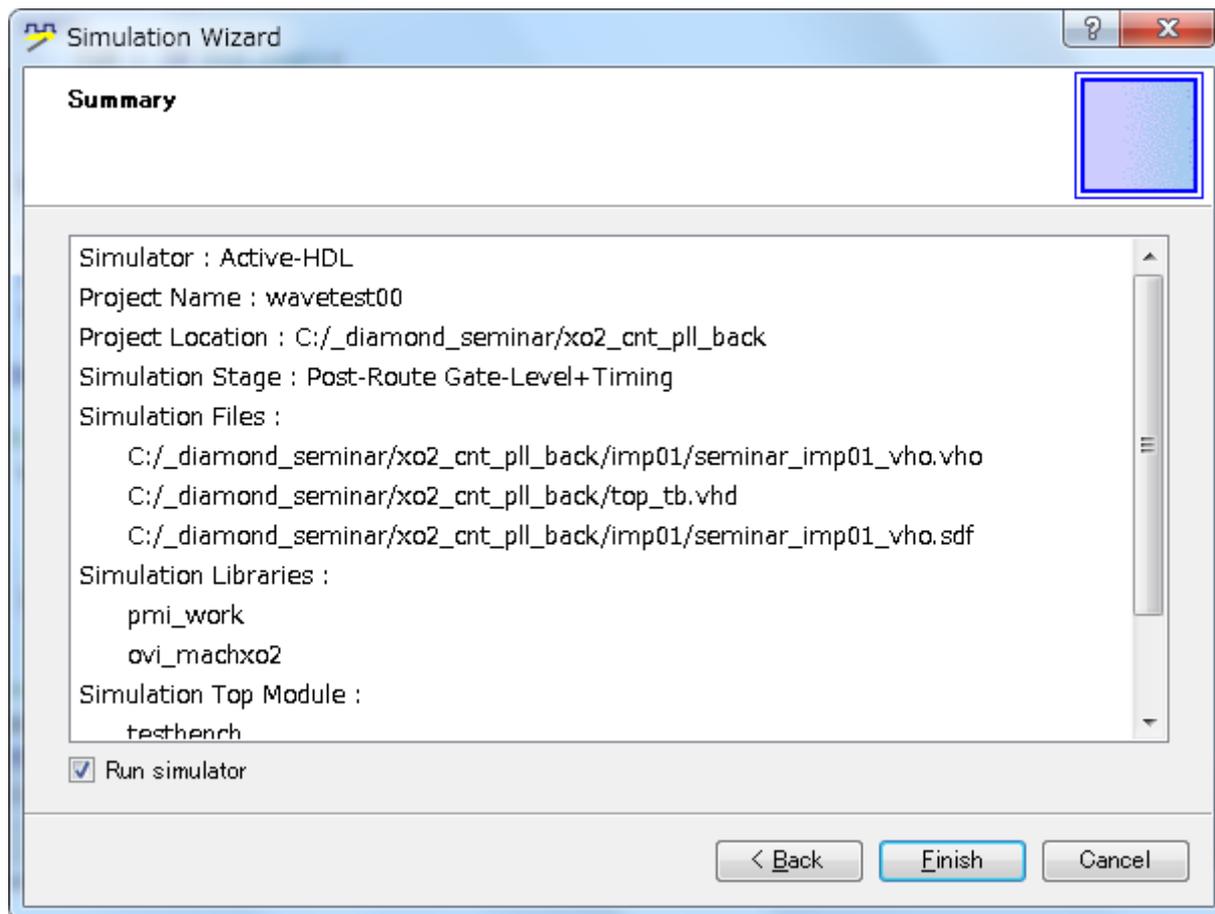
③ テストベンチを追加で
取り込みする

② SDF (遅延ファイル)の
取り込みを確認する

※デザインのソースは取込まないで
ください。デザインのソースを取込
んでしてしまうと、Functionシミュ
レーションとなってしまいます。

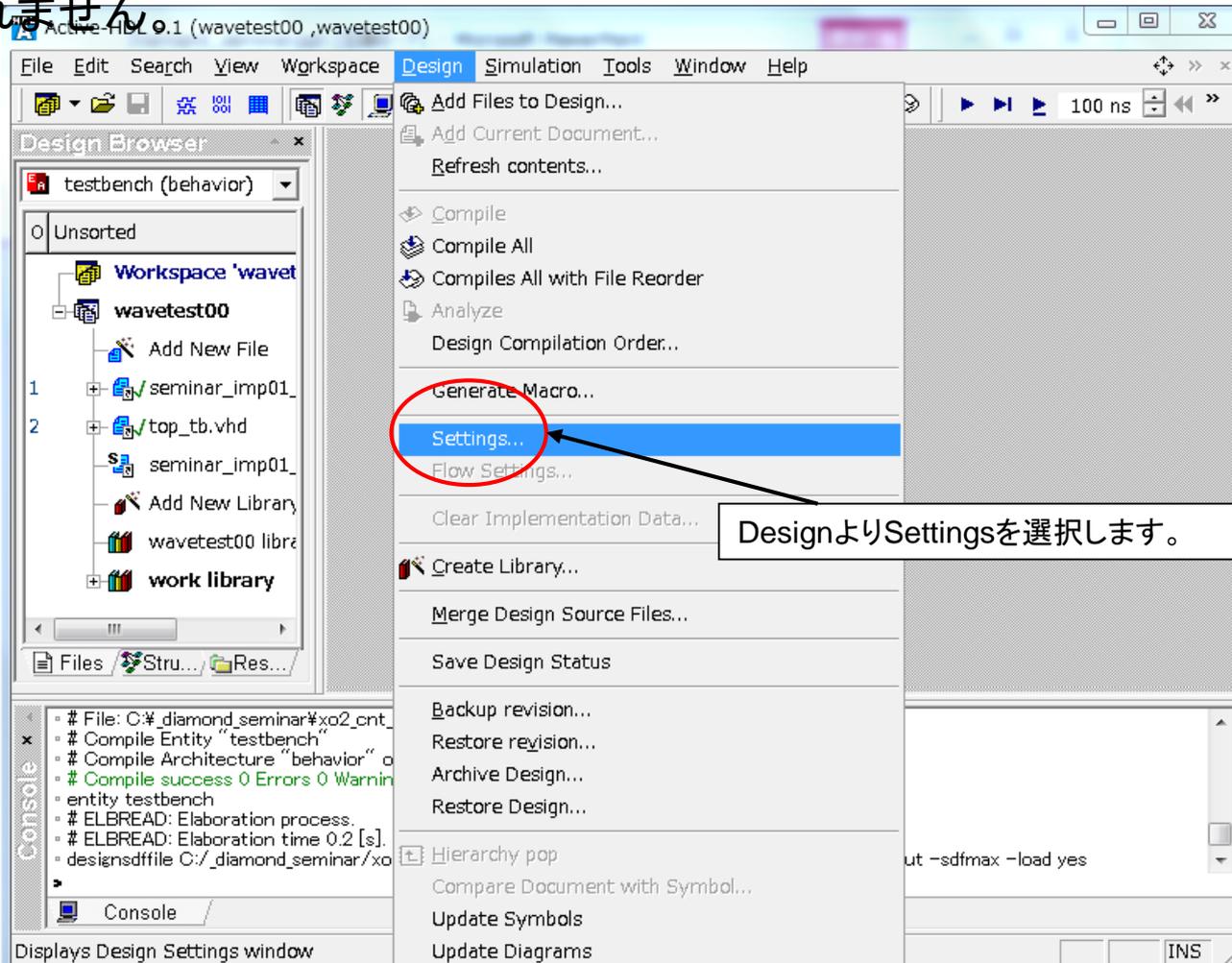
Timingシミュレーションでは、コンパ
イル済みの vo, vho 使用します。

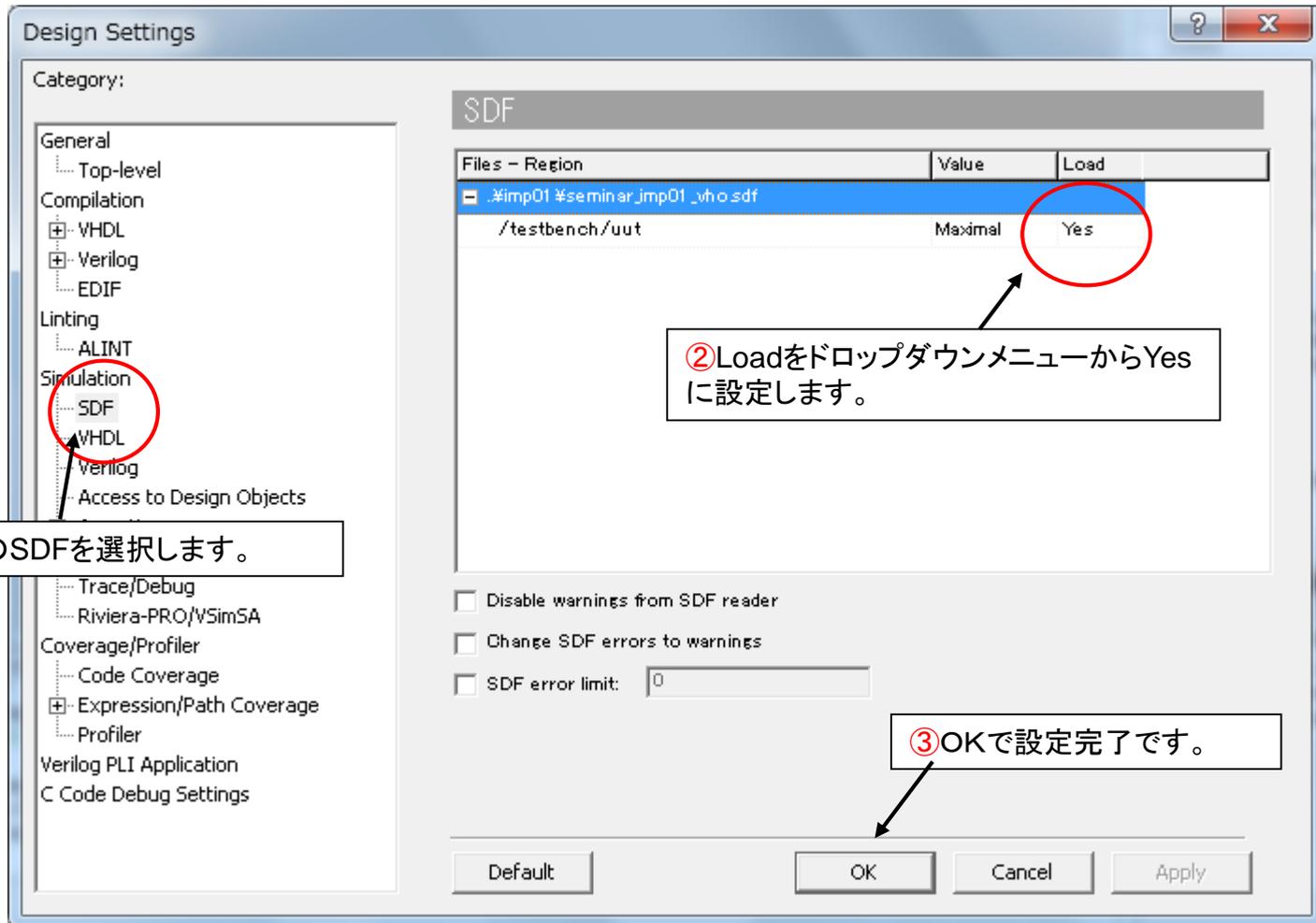




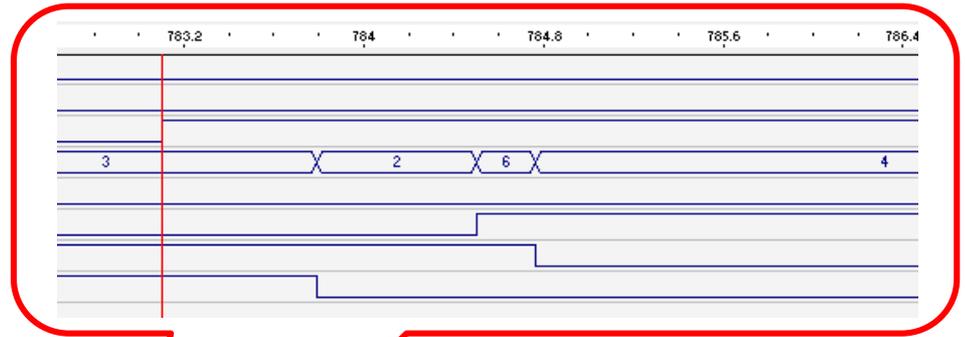
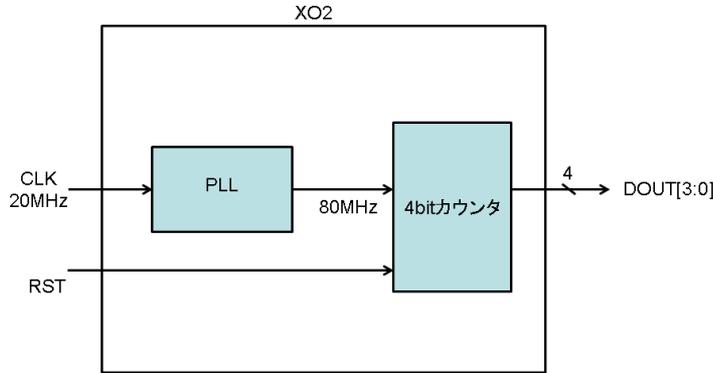
FinishをクリックするとActive-HDLが自動で立ち上がります。

- Waveform を表示する前に以下設定が必要になります。
 - 設定を行わない場合、SDFファイルが無効となり内部遅延がシミュレーションに反映されません。

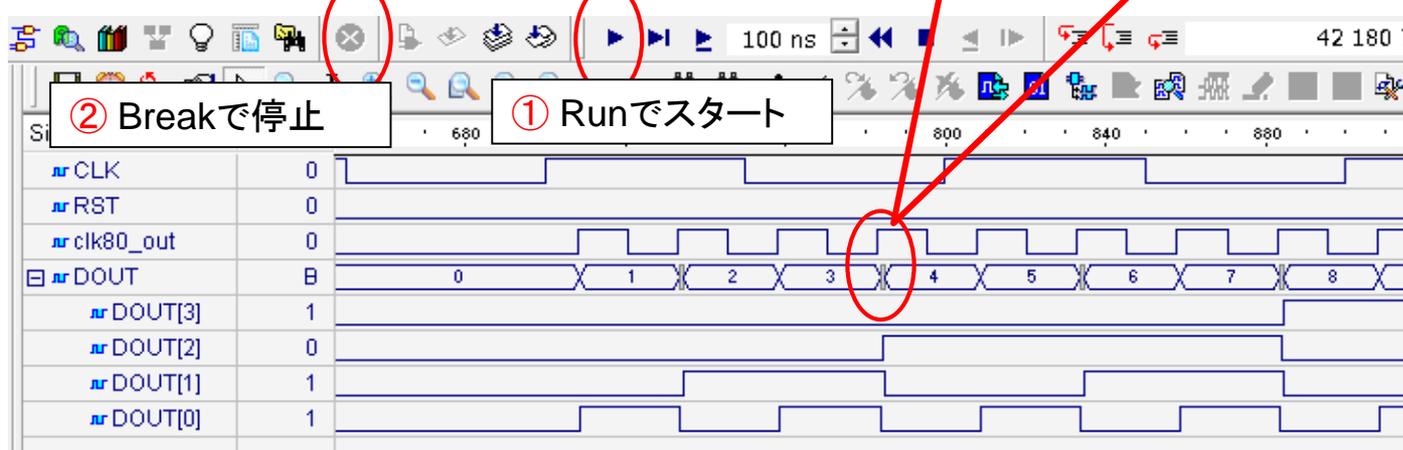




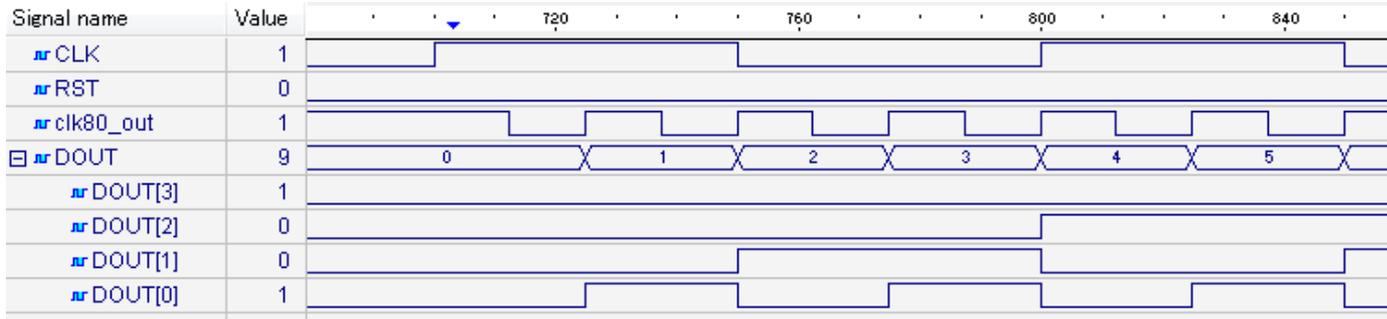
※ツールのバージョンによっては初めから設定されております。



■ Timingシミュレーションの波形



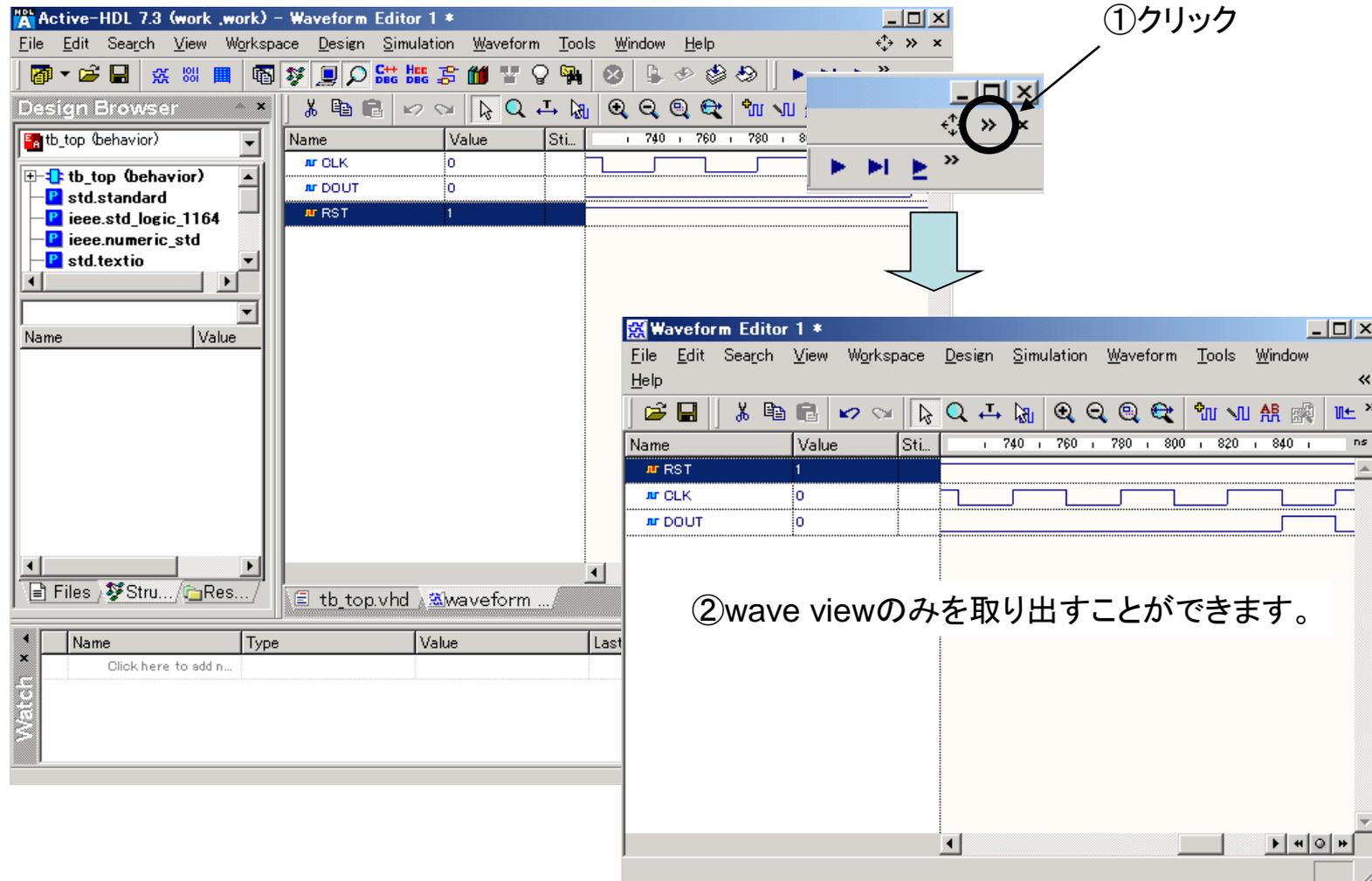
■ Functionシミュレーションの波形



その他、基本操作

Undockボタン

Undockをクリックして、wave画面を単独に取り出すことができます。



信号の移動

移動したい信号をドラッグ & ドロップで簡単に移動できます。

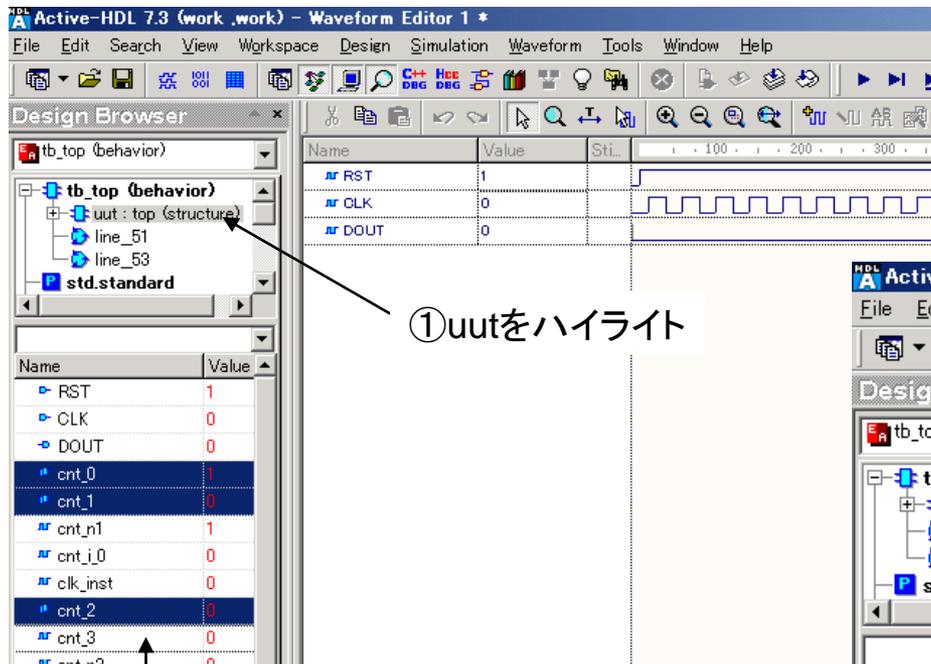
The screenshot shows the ActiveHDL 7.3 Waveform Editor interface. On the left is the Design Browser showing the project structure for 'tb_top (behavior)'. The main window displays a table of signals and their values. In the top part of the table, RST is at the bottom. In the bottom part of the table, RST has moved to the top. A large blue arrow indicates this movement. Red arrows point to the RST row in both tables. Text annotations explain the process: '①ドラッグ & ドロップ' (Drag & Drop) and '②RST信号が移動します。' (RST signal moves).

Name	Value	Sti...
CLK	0	
DOUT	0	
RST	1	

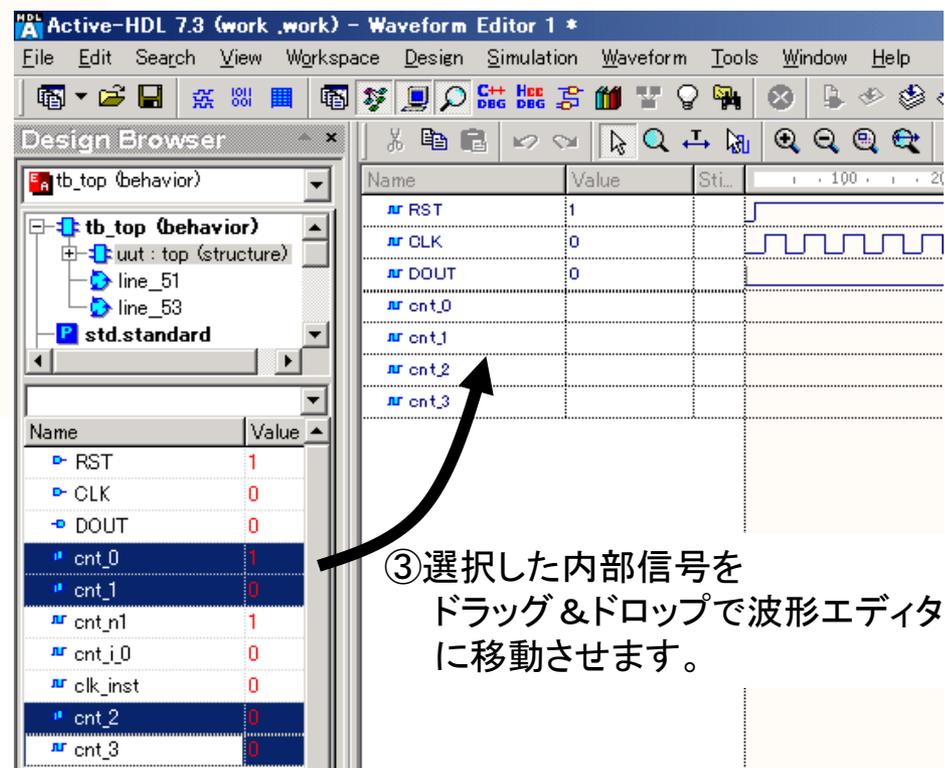
Name	Value	Sti...
RST	1	
CLK	0	
DOUT	0	

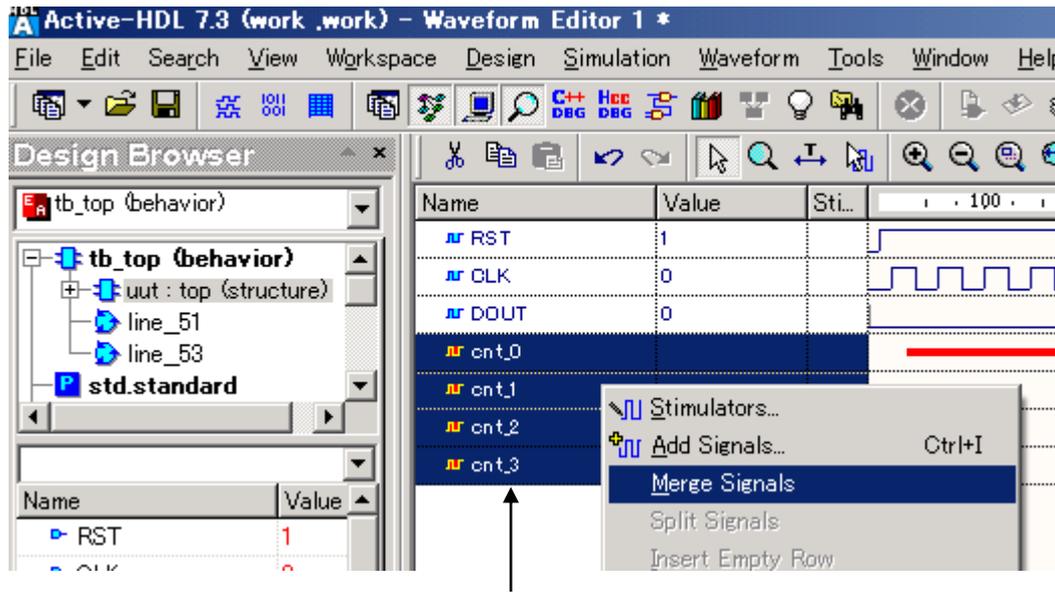
内部信号の観測1

観測したい内部信号を簡単に観測することが可能です。

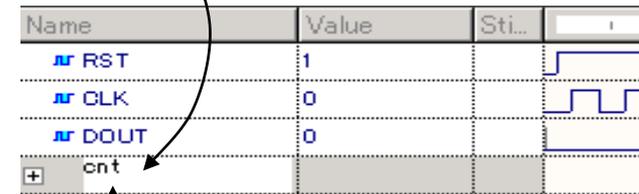
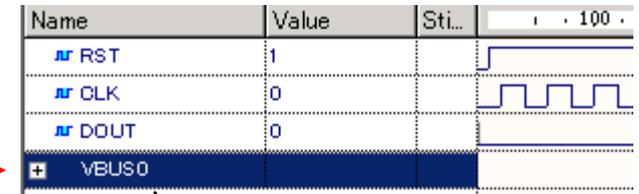


② uutをハイライトすると、内部信号が表示されます。
内部カウンタcnt0～cnt3を観測したい場合はCntIキーを押しながら選択すると、複数の信号を選択できます。





④取り出した内部信号をバス化したい場合は対象となる信号を選択し、[右クリック]→[Merge Signals]を選択すると、右図のようにバス化されます。



⑤バス名を変更したい場合は、自動的に作成されたバス名をクリックすると、名前の変更が可能です。

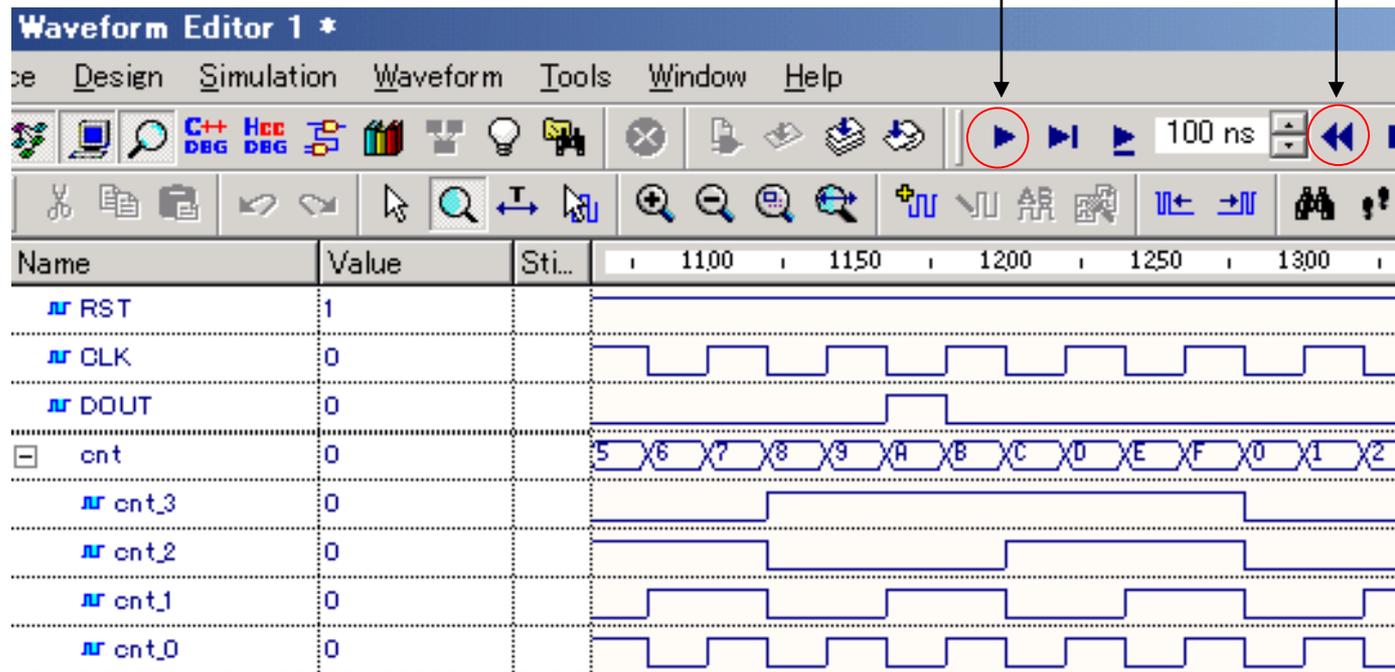


⑥バス名の前の+ボタンをクリックすることで内部バスの信号全てを表示できます。また、内部信号の並び替えもドラッグ&ドロップで可能です。

ドラッグ&ドロップで並び換え可能。

観測したい内部信号を全て取り出した後、restart を実行し、Run をかけることによって、取り出した内部信号の波形を観測することが可能です。

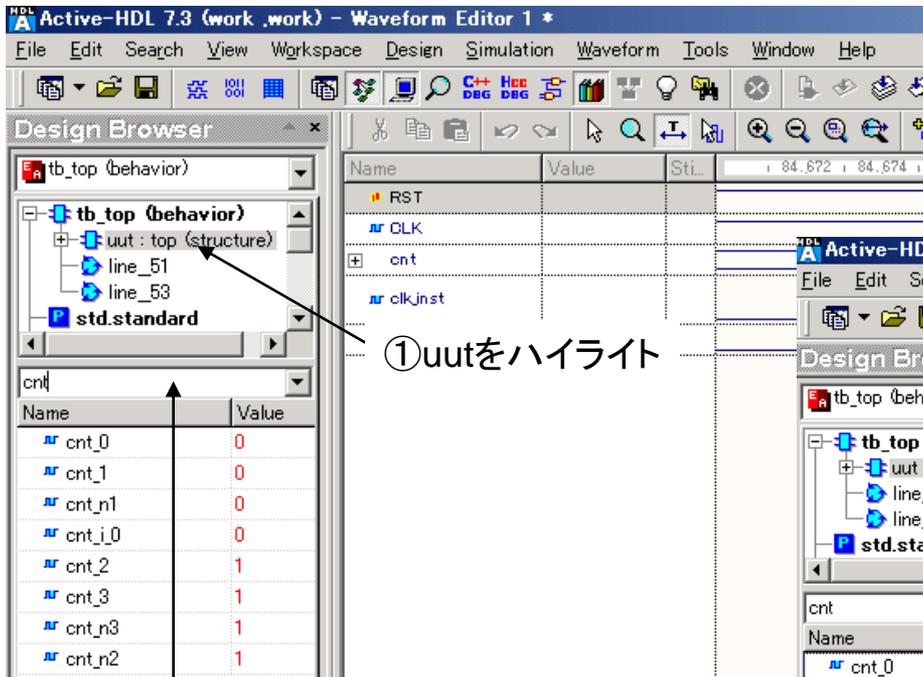
⑧Runをクリック ⑦restartボタンをクリック



取り出した内部信号波形を確認できる

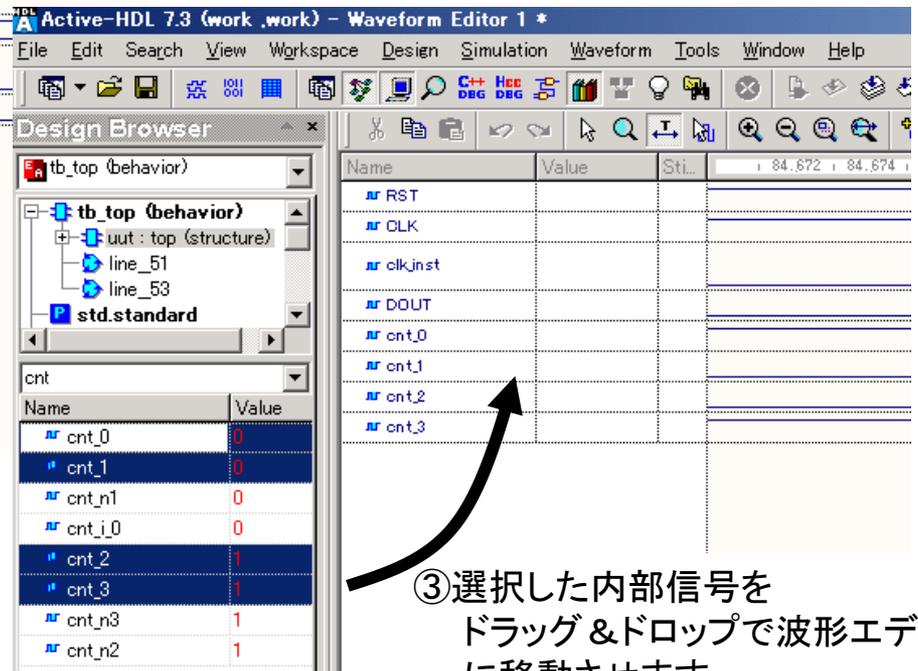
内部信号の観測2

内部信号が多い場合は、観測したい内部信号を簡単に検索することが可能です。



① uutをハイライト

② cntという内部信号を検索したい場合はこの部分に入力し、リターンを押します。リターン後にcntを含む内部信号が表示されます。

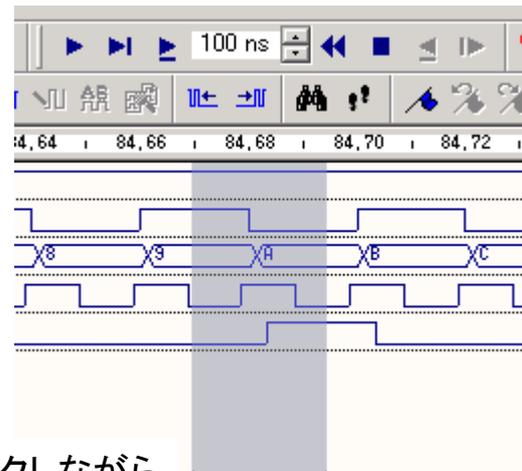
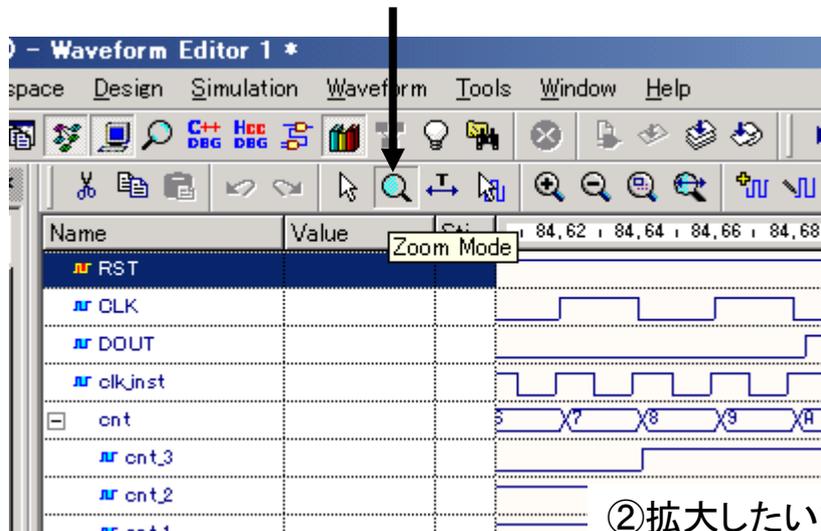


③ 選択した内部信号をドラッグ & ドロップで波形エディタに移動させます。

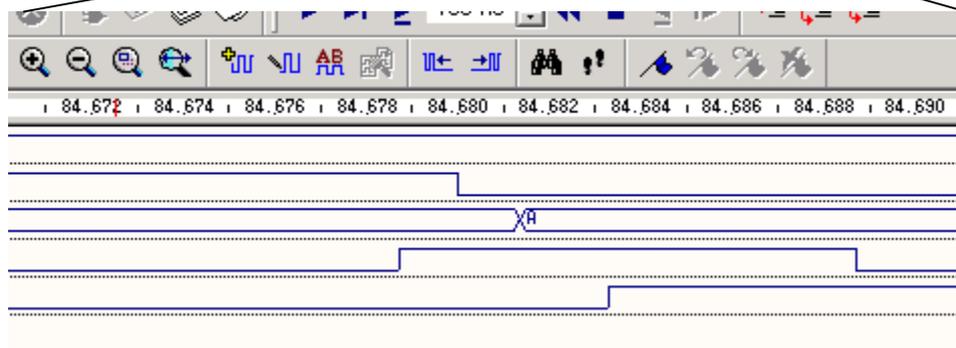
cnt

2. Waveform editor から遅延を簡単に確認することができます。

①まず、Zoom Mode をクリック

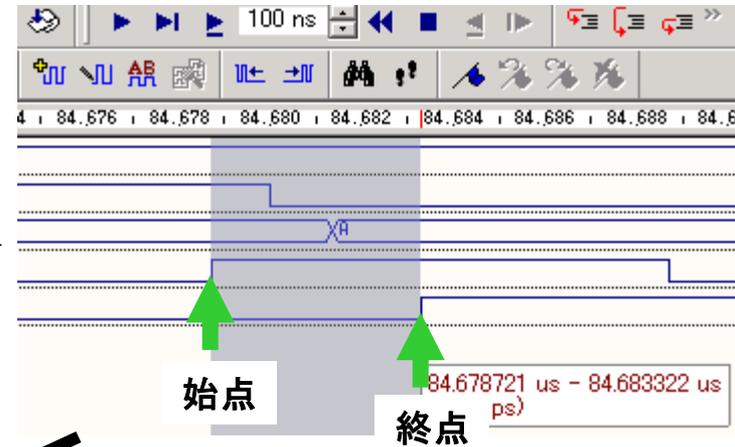
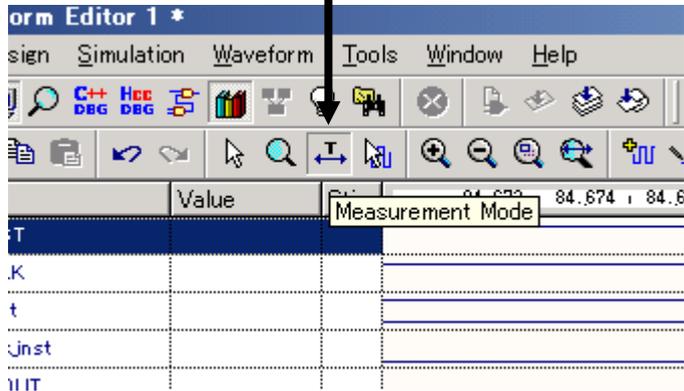


②拡大したい範囲をクリックしながら
選択します。

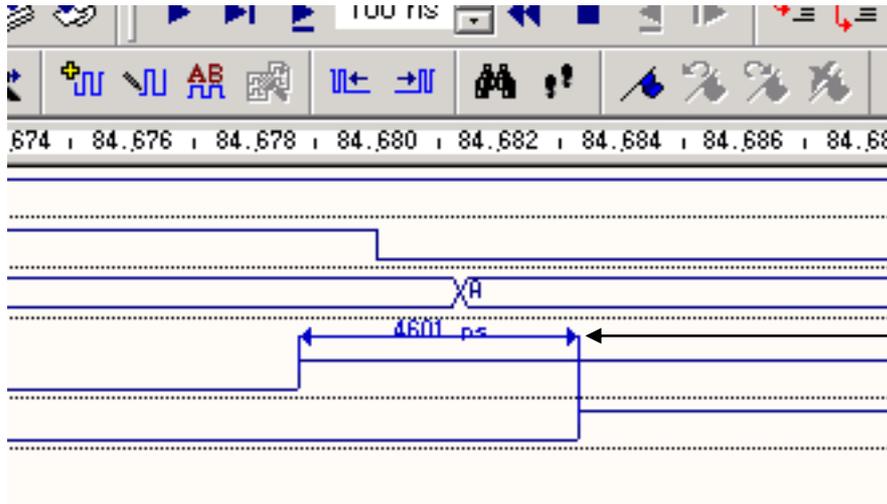


②を実行すると右図のように
拡大されます。

- ③遅延を測定する為に、Measurement Mode をクリック。



- ④測定したい遅延の始点をクリックした後、クリックした状態のまま、終点まで持ってゆき、離します。始点、終点のエッジ部でカーソルが緑色に変化します。その箇所を選択してください。



- ⑤選択した範囲の遅延が表示され、4.6nsであることが確認できます。

◇ブレークポイントの設定

1.ブレークポイントを設定すると希望する状態で検証プロセスを停止させることができます。

①Simulation → Breakpoints
を選択します。

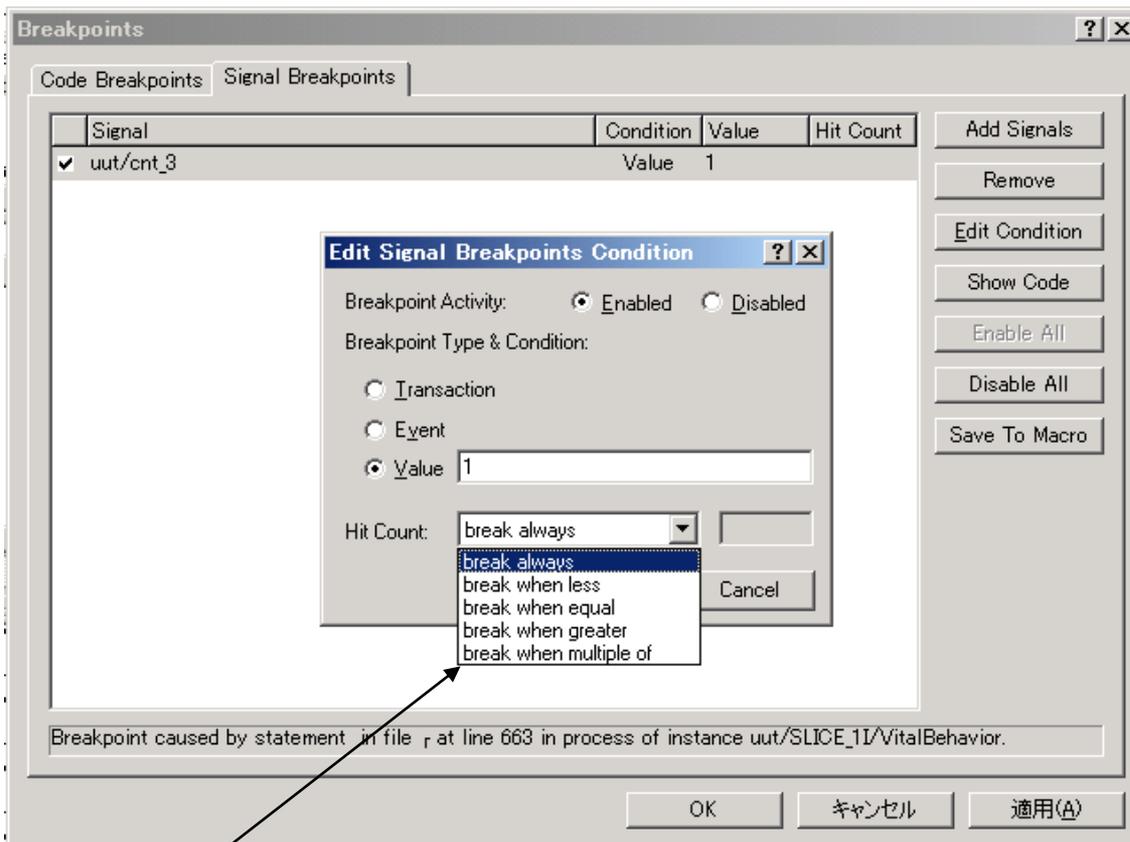
②Signal Breakpointsを選択し、信号を選択。
(今回は内部ノードのcnt_3を選択)

Signal	Condition	Value	Hit Count
✓ uut/cnt_3		Value 1	

Name	Value
cnt_1	0
cnt_n1	0
cnt_i_0	0
clk_inst	1
cnt_2	0
cnt_3	1
cnt_n3	1
cnt_n2	0

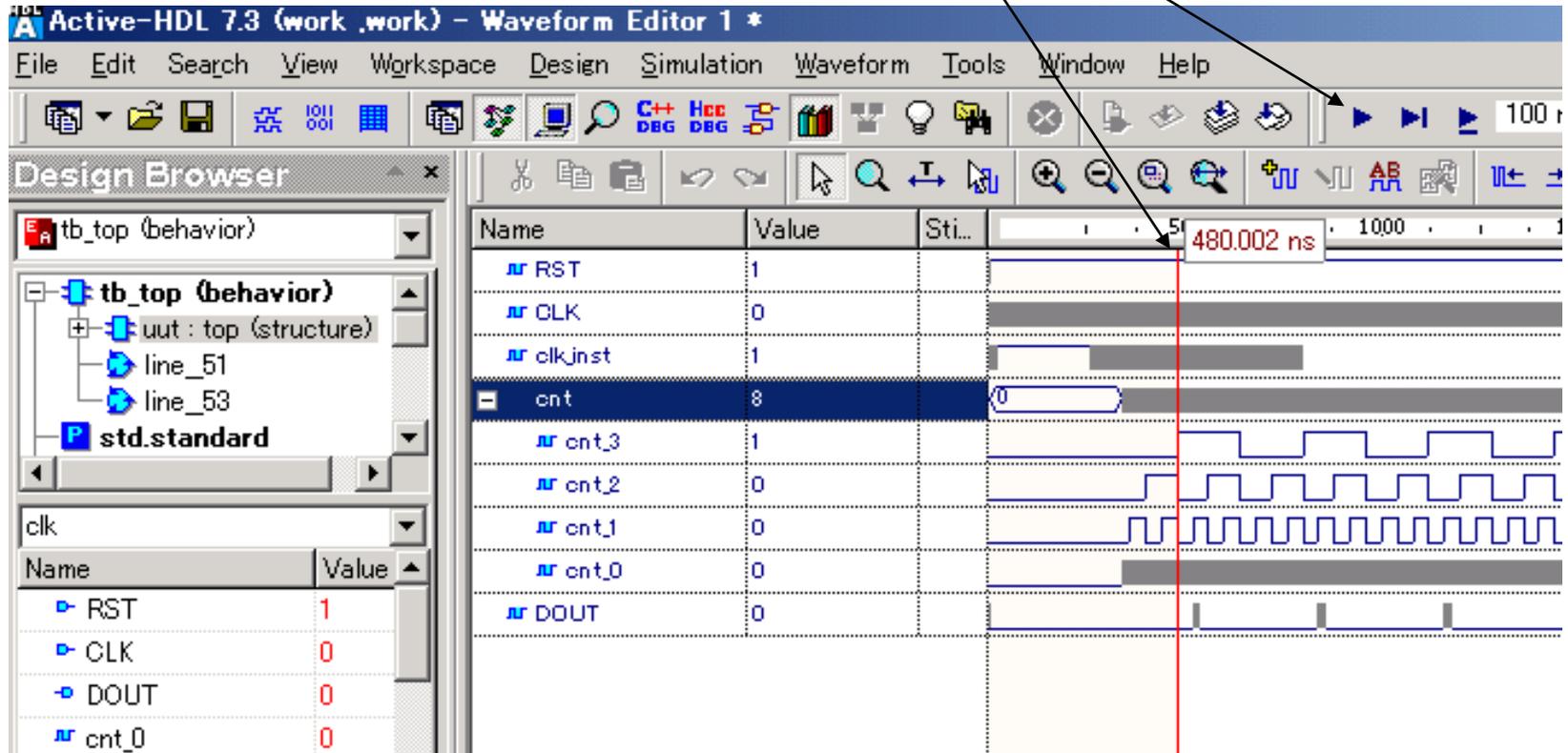
Breakpoint caused by statement in file r at line 663 in process of instance uut/Slice_11/VitalBehavior.

OK キャンセル 適用(△)



- ③
- break always : 指定したラインが実行されたら停止
 - break when less : 指定したHit countよりも少ない場合に停止
 - break when equal : 指定したHit countと同じ場合に停止
 - break when grater : 指定したHit countよりも多い場合に停止
 - break when multiple of : 指定したHit countの整数倍で停止

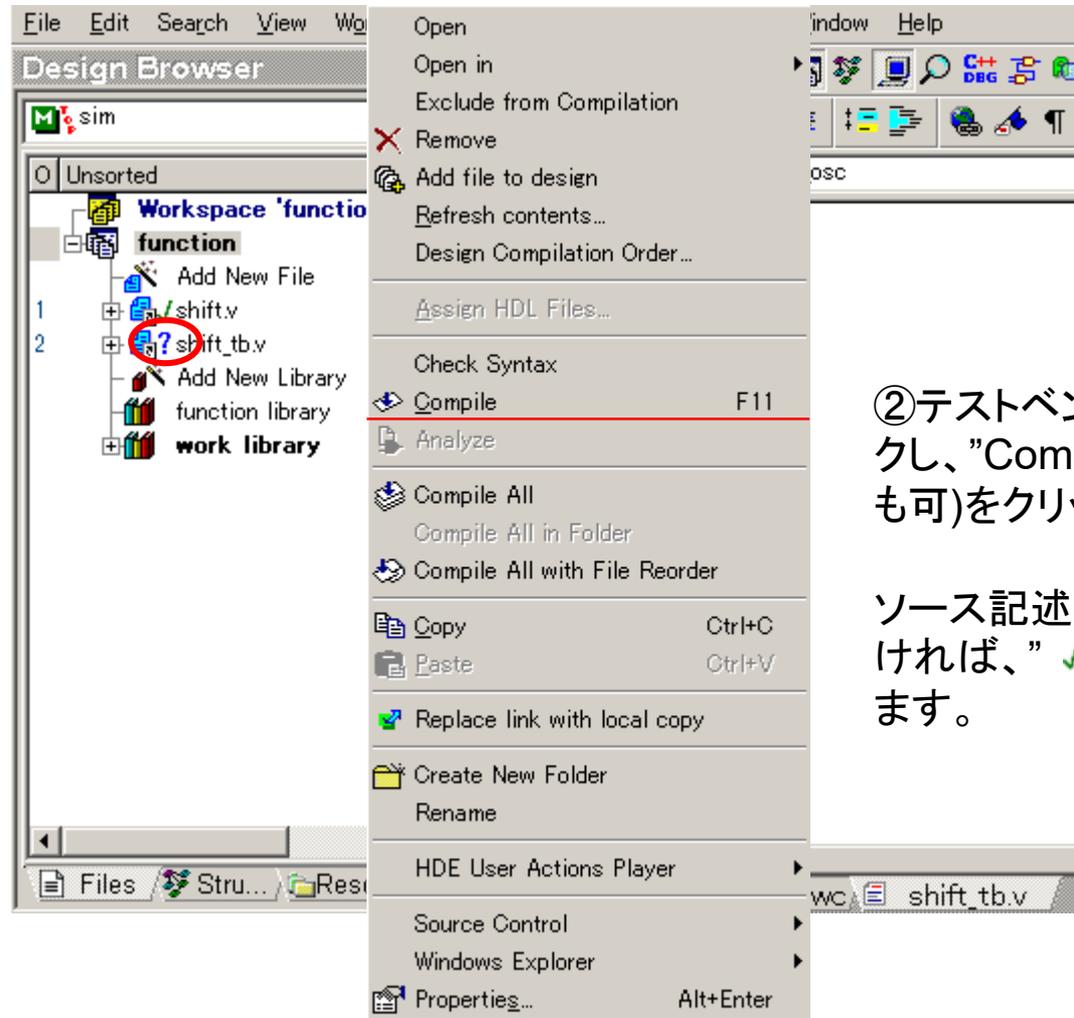
④runボタンをクリックすると、cnt3 = 1の時にBreakされ、赤いラインで表示されます。



再度runボタンをクリックすると、次のcnt3 = 1時にBreakされます。

テストベンチソースを修正し、 再度シミュレーションを行う方法

①テストベンチソースを修正し保存すると、ソース名の左側の” ✓ ”が” ? ”マークに変わります。



②テストベンチを右クリックし、”Compile”(F11でも可)をクリックします。

ソース記述に問題が無ければ、” ✓ ”に変わります。

⑤ Break

④ Run

③ Restart Simulation



この操作で、Waveformに表示している信号名はそのまま、ソース修正後の再シミュレーションが可能です。