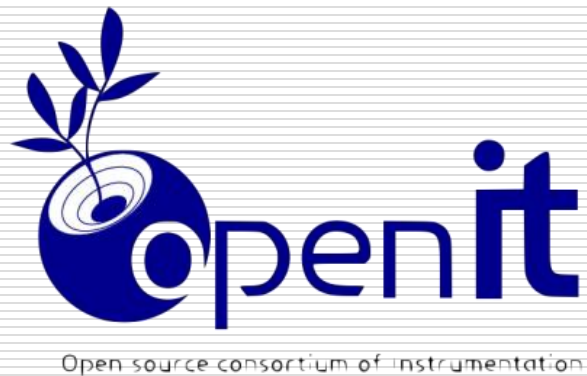


# Open-It FPGAトレーニングコース(入門編)

## 2.3 論理シミュレーション(組み合わせ回路)

---



第3.3版

2016年06月22日

# シミュレーション

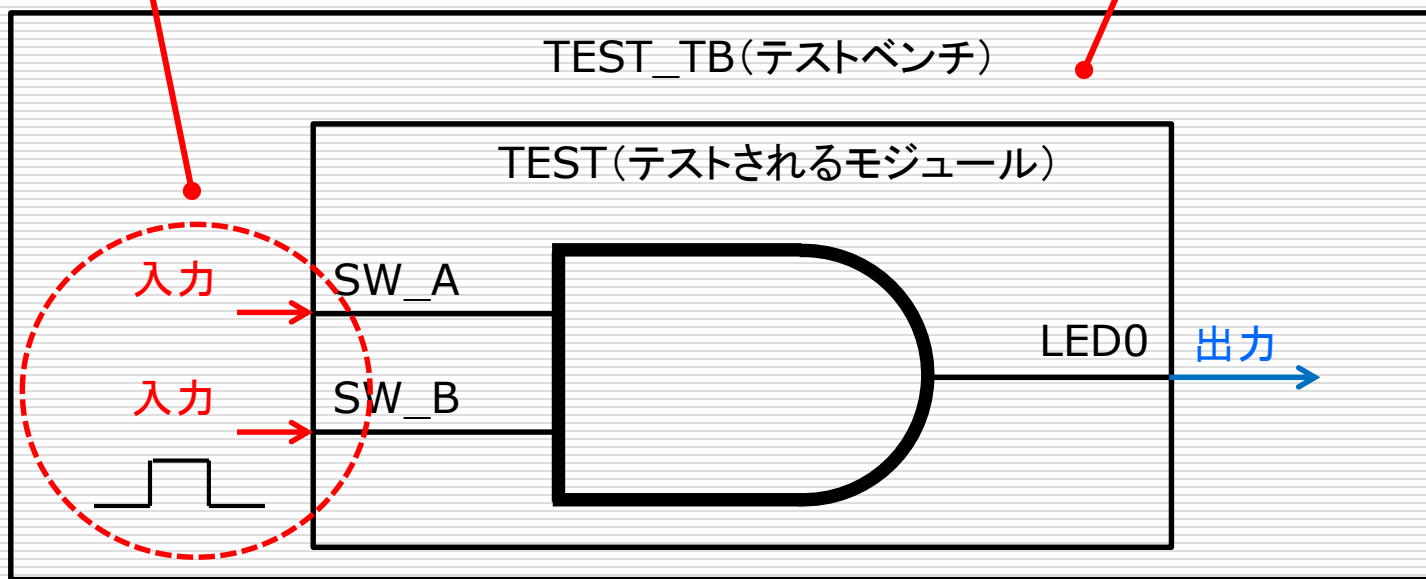
---

- PC上で設計した回路を仮想的に動作させること
  
- 新たなツールが必要
  - 無料、有料様々なものがある
  - このコースではVivadoに付属しているシミュレータを使用します
  
- テストベンチと呼ばれるテスト環境をPC上に作る
  - 何故テストベンチが必要なのか？
  - 開発する回路は外部と通信するための信号を必ず持っている。それらの信号を与える必要がある
  - 想定される入力を設計した回路に与え、その応答が設計通りの動作か否かを確認する
  - 例えば、TEST.vモジュールをテストするためにはSW\_A, SW\_Bに信号を与える必要がある

# テストベンチ

SW\_A, SW\_Bに信号を与える

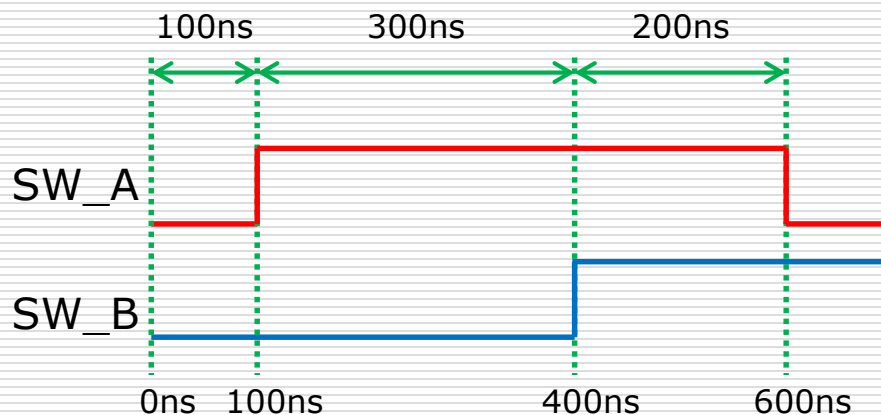
テストするためのシミュレーション環境



- 入力を作り、出力を確認するのがテストベンチ
- これから実際にテストベンチを作り、入力、出力を確認します

# 入力信号、出力信号

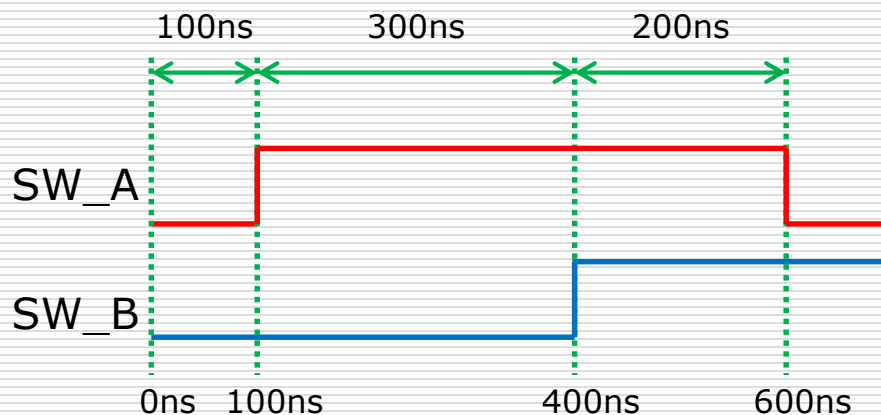
- 今回は以下のような入力信号を作ります



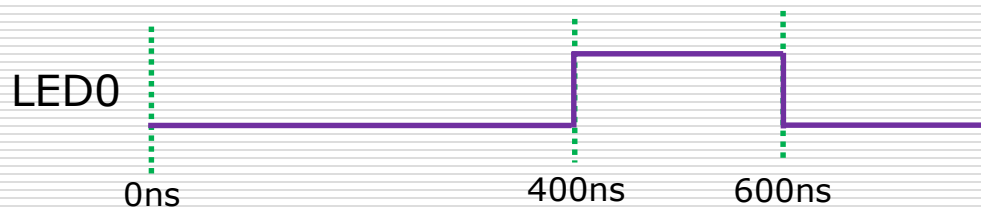
- 出力信号は？

# 入力信号、出力信号

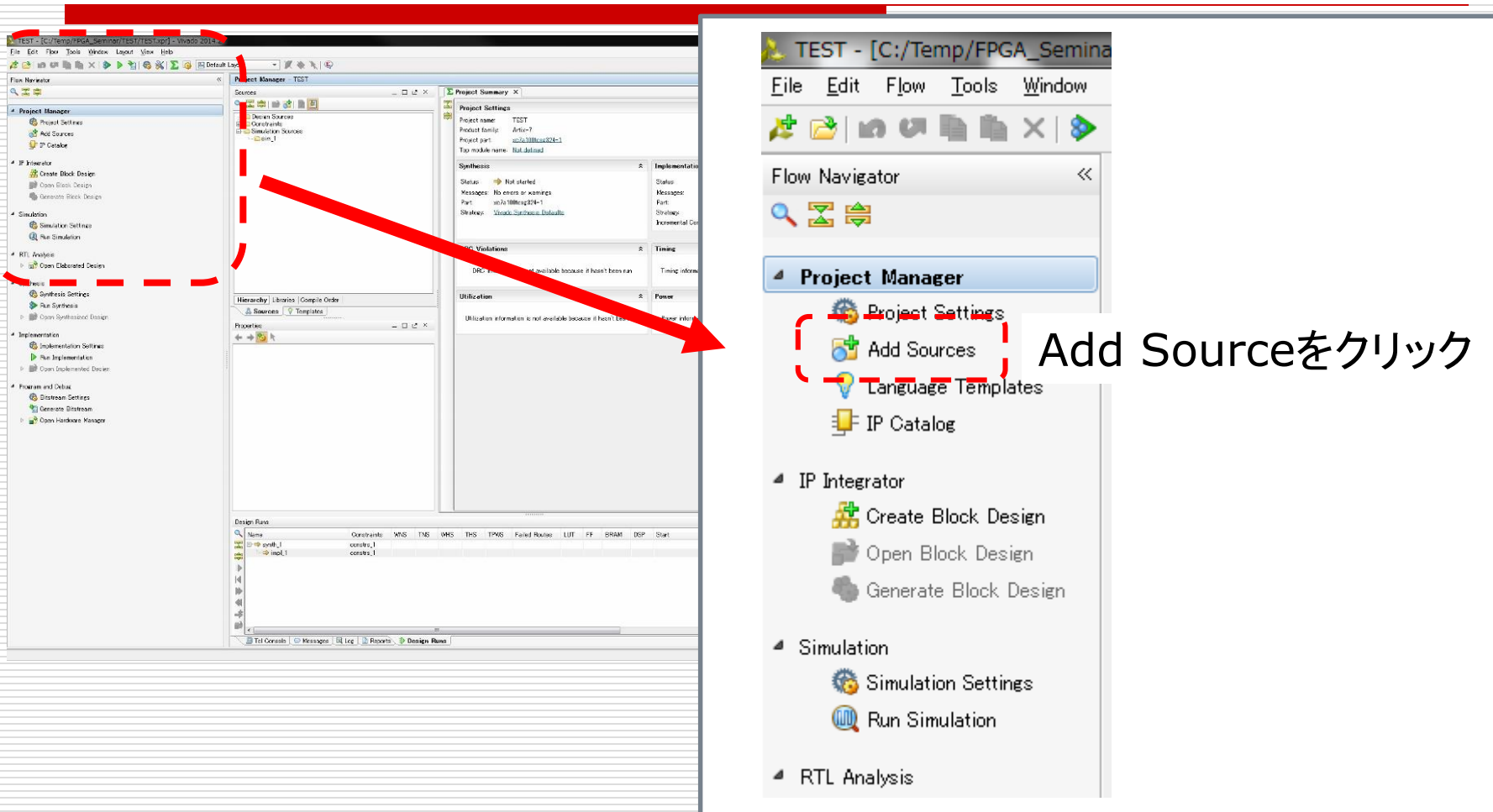
- 今回は以下のような入力信号を作ります



- 出力信号は？

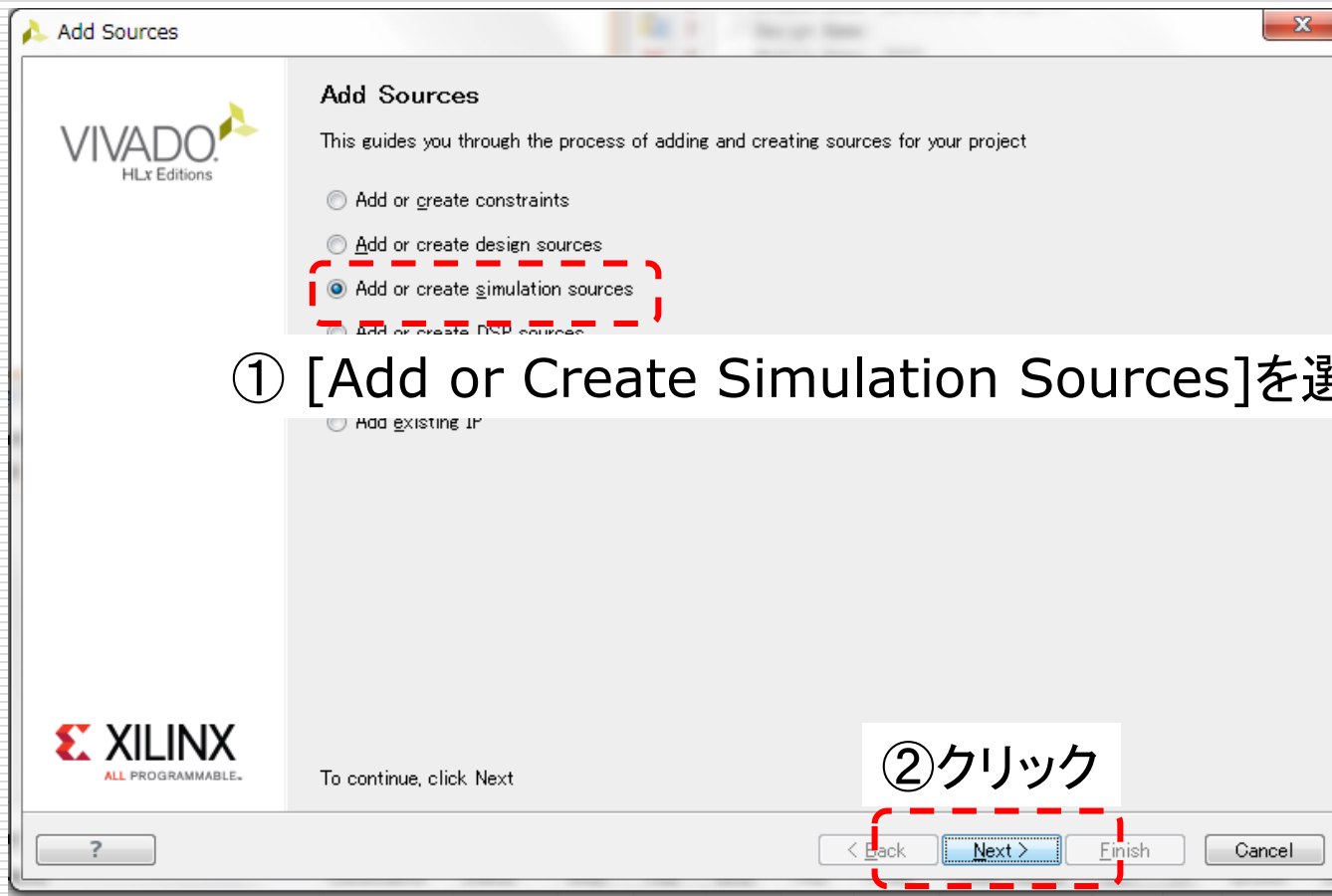


# テストベンチの作成

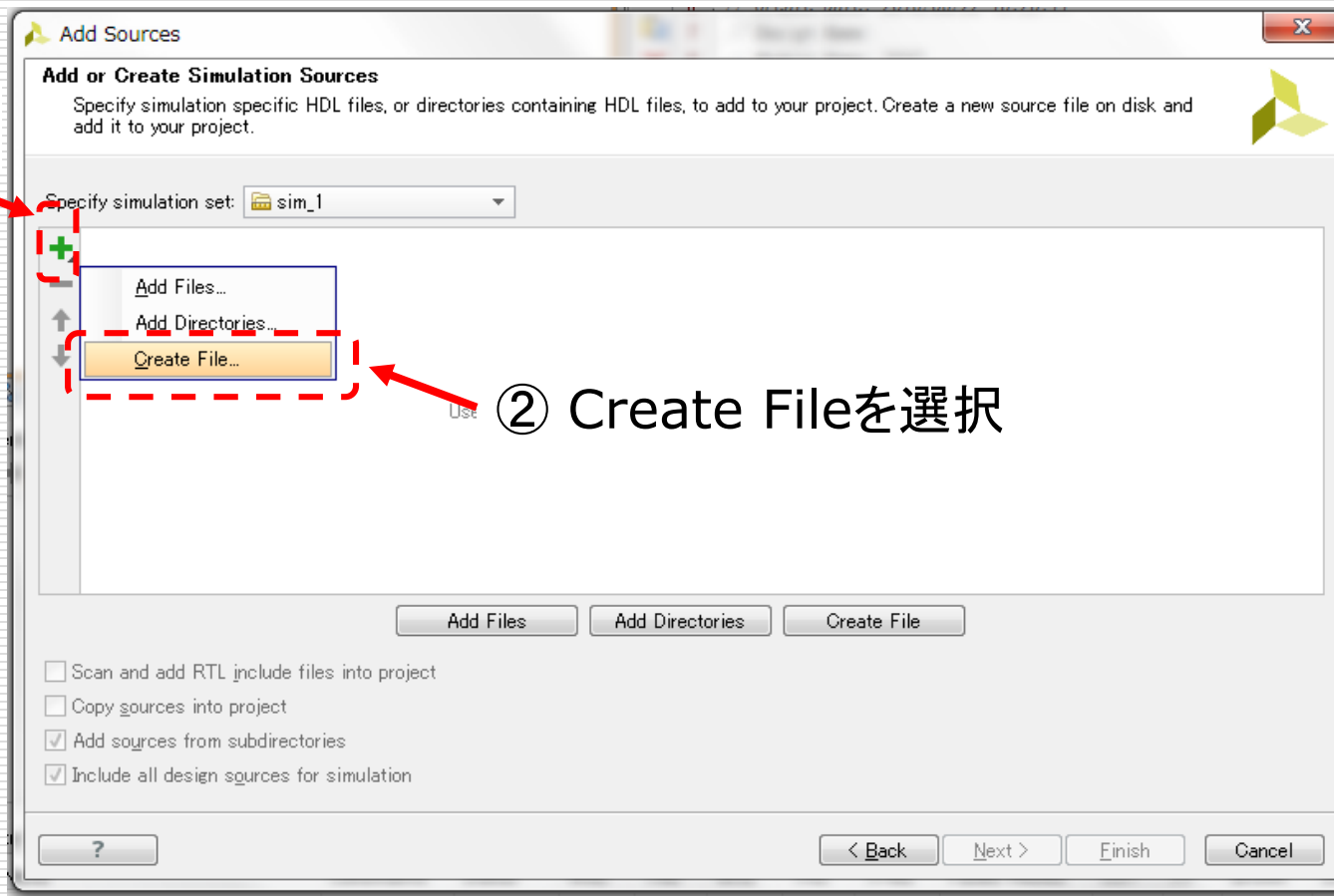


The image shows a screenshot of the Open-It IDE interface. The main window displays the Project Manager for a project named 'TEST'. The 'Add Sources' button is highlighted with a red dashed box. A red arrow points from this button to a larger, zoomed-in view of the Project Manager window on the right. In this zoomed view, the 'Add Sources' button is again highlighted with a red dashed box, and a text box next to it says 'Add Sourceをクリック' (Click Add Source).

# ウィザード開始

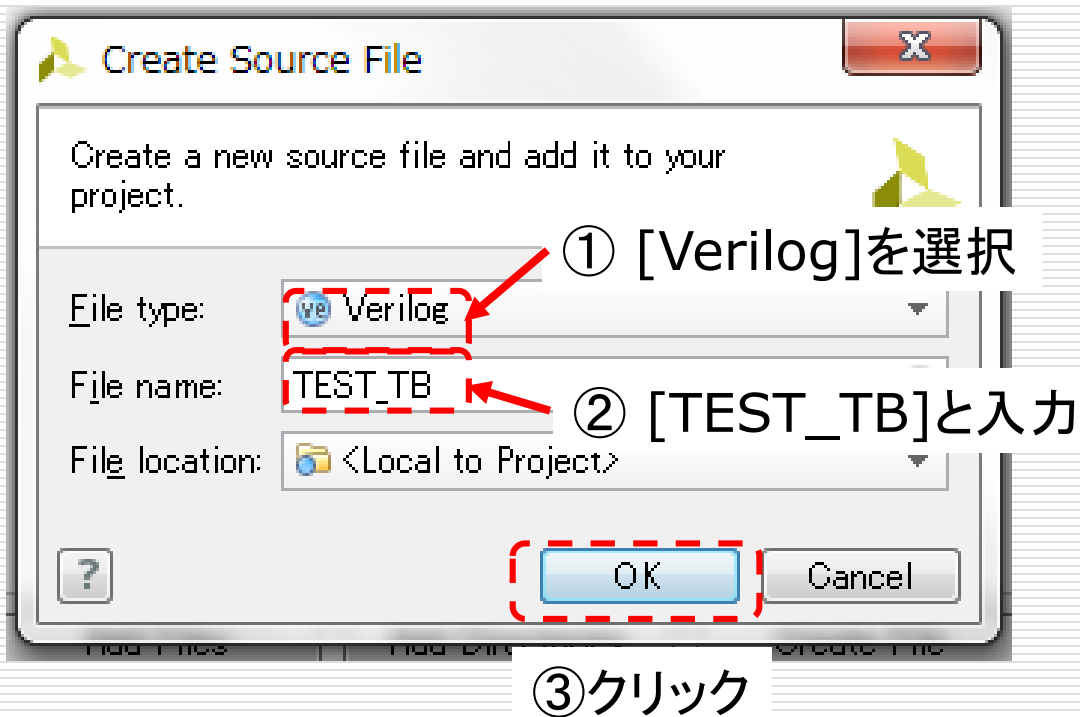


# テストベンチ・ファイル作成

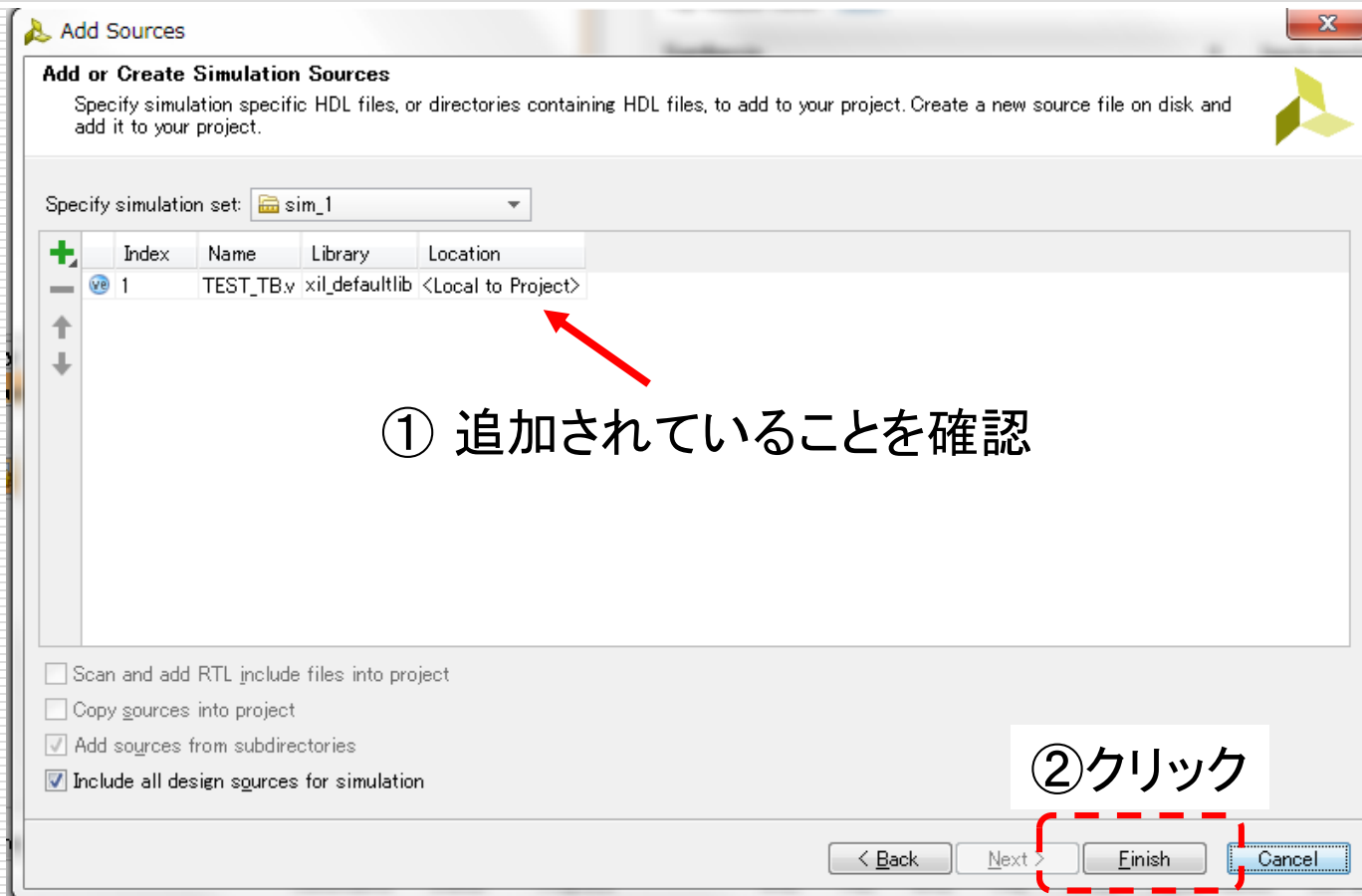




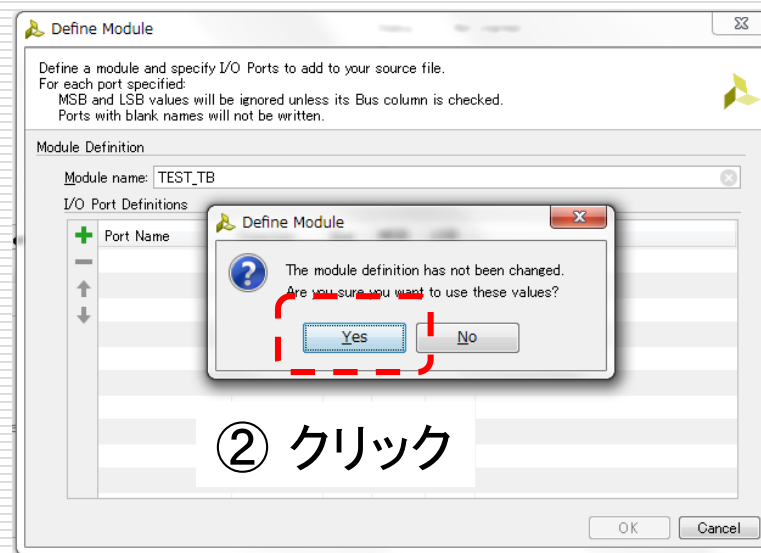
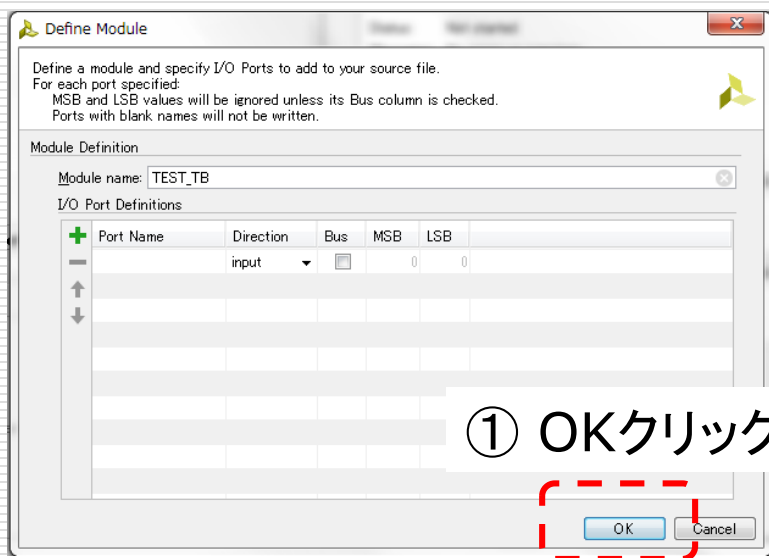
# ファイル名の指定



# ファイル名などの確認

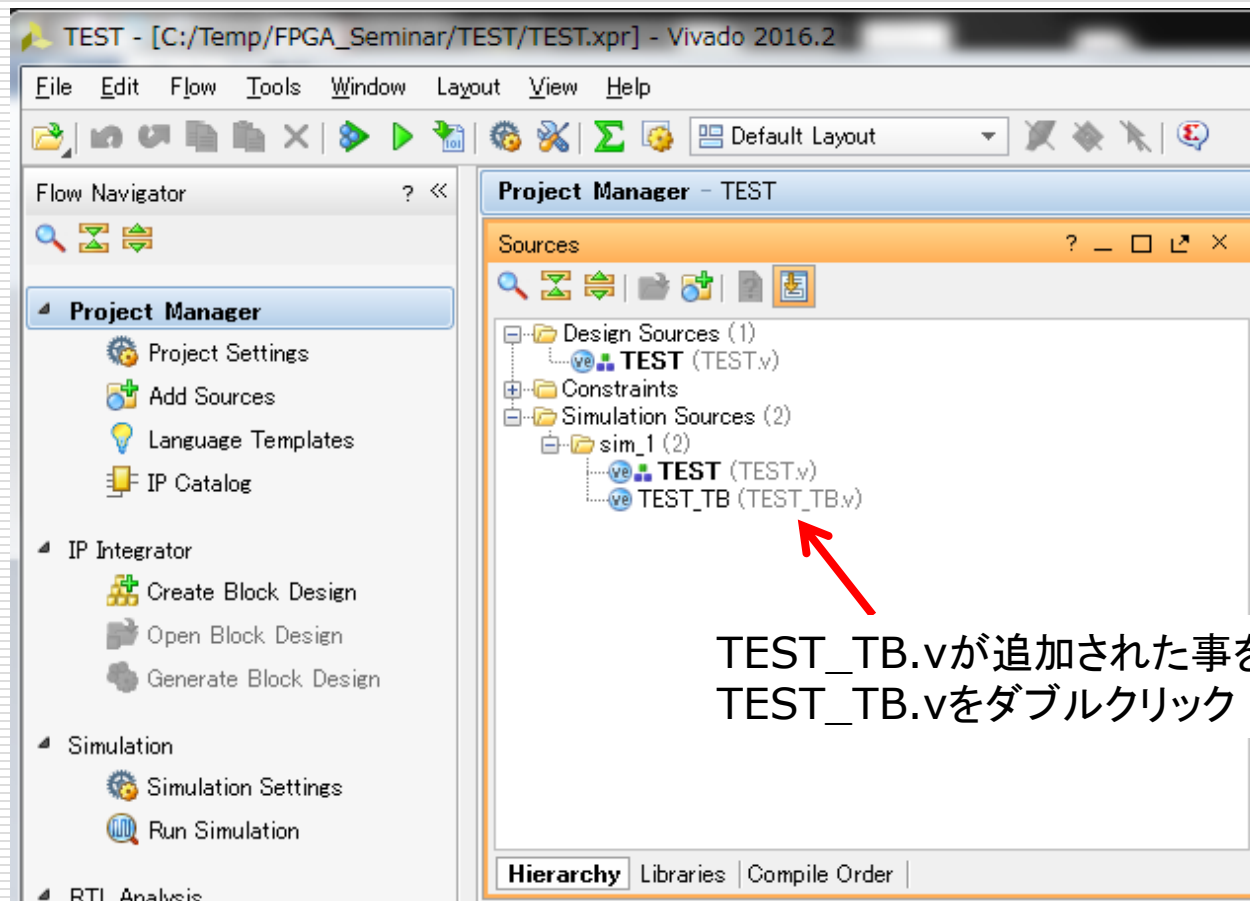


# ポート設定



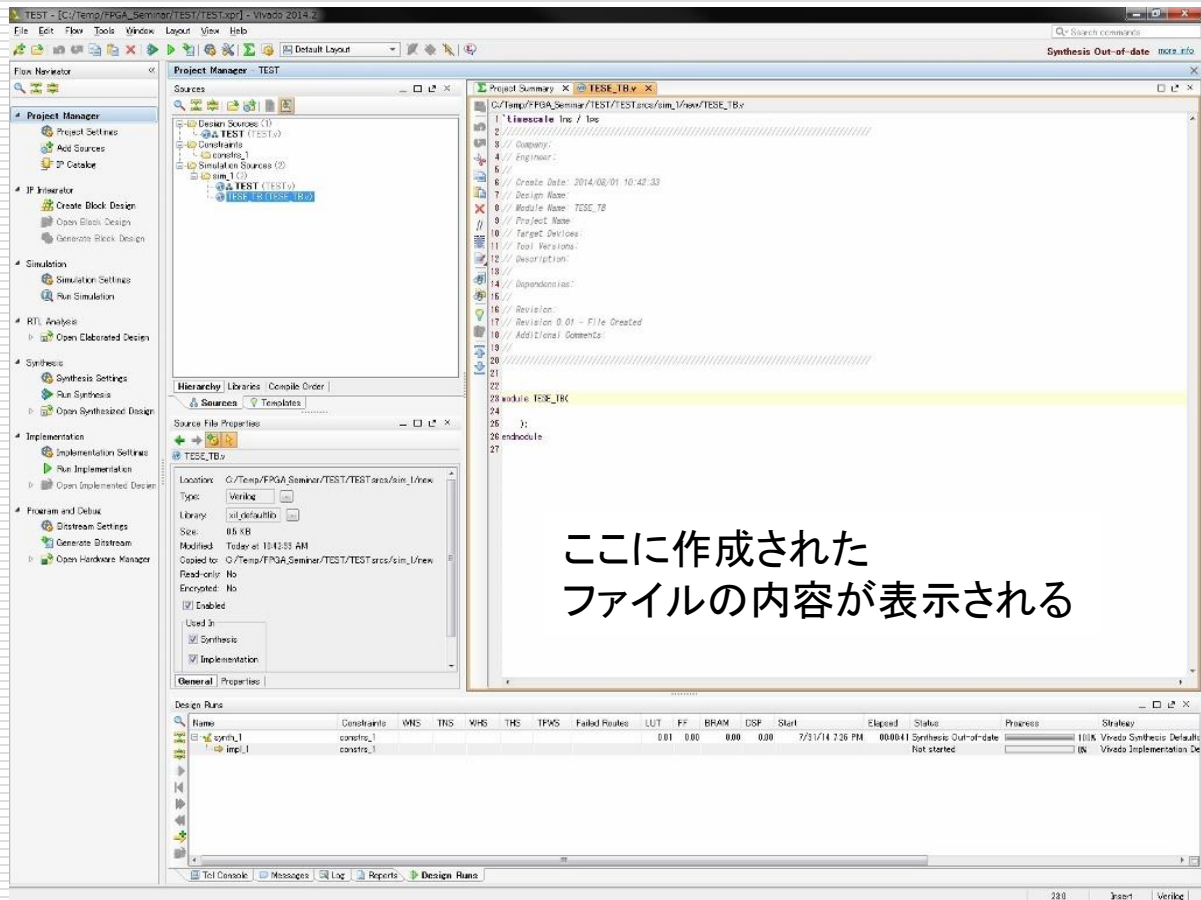
テストベンチは入出力がないので  
何も入力せずにOKをクリック

# 作成したファイルの確認



TEST\_TB.vが追加された事を確認し  
TEST\_TB.vをダブルクリック

# 作成したファイルを開く



ここに作成された  
ファイルの内容が表示される

Name	Constraints	WNS	TNS	WHS	THS	TPMS	Failed Routes	LUT	FF	BRAM	DSP	Start	Elapsed	Status	Progress	Strategy
const_1	const_1							0.01	0.00	0.00	0.00	7/31/14 7:26 PM	00:00:41	Synthesis Out-of-date	100%	Vivado Synthesis Default
impl_1	const_1													Not started	0%	Vivado Implementation De

# 作成したファイル内容

```

Project Summary x TEST_TB.v x
C:/Temp/FPGA_Seminar/TEST/TEST.srcs/sim_1/new/TEST_TB.v
1 `timescale 1ns / 1ps
2 //////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 2016/06/22 13:49:34
7 // Design Name:
8 // Module Name: TEST_TB
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21
22
23 module TEST_TB(
24
25 );
26 endmodule
27
  
```

テストベンチ内のユニット単位:  
時間を表す数字を用いたときに  
採用される単位

シミュレーションの精度:  
この時間単位でシミュ  
レーションされる

← テストベンチのモジュール名

ここにテストベンチの内容を書く

# ポートリストの削除

テストベンチは製作した回路に必要な全ての入力を与え、全ての出力を観測するので外部信号は存在しない

```
22
23 module TEST_TB;
24
25 |
26
27 endmodule
28
```

ポートリストの括弧を削除してください

# 入力信号の定義

設計した回路に入力される信号をregタイプで定義  
設計した回路から出力される信号をwireタイプで定義

```
23 module TEST_TB;  
24  
25     reg SW_A;  
26     reg SW_B;  
27     wire LED0;  
28  
29 endmodule  
30
```

regタイプについて:

記憶素子が出力する信号であることを意味する。  
記憶素子なので一度値を設定すると次回設定するまで  
設定した値を保持し続ける

wireタイプについて:

組み合わせ回路出力または単なる配線を意味。  
配線による接続なので設定した値が直ぐに現れる  
値を保持する機能は無い

補足) テストベンチの信号名は自由。設計した回路の入出力名と異なっても良い。



# TESTモジュールの組み込み

```

22
23 module TEST_TB;
24
25     reg SW_A;
26     reg SW_B;
27     wire LED0;
28
29     TEST uut(
30         .SW_A(SW_A),
31         .SW_B(SW_B),
32         .LED0(LED0)
33     );
34
35 endmodule
36

```

下位モジュールの  
モジュール名

テストベンチ内での名前  
インスタンス名と言う

ポート名の前には  
ドットを必ず付ける

同じモジュールを複数組み込んだときに  
区別できるようにモジュール名と異なる名前を使用する事ができる  
もちろんモジュール名と同じでも良い

下位モジュールのポート名

次がある場合はコンマでつなぐ

括弧の中はテストベンチの信号名

# 与える入力の記述

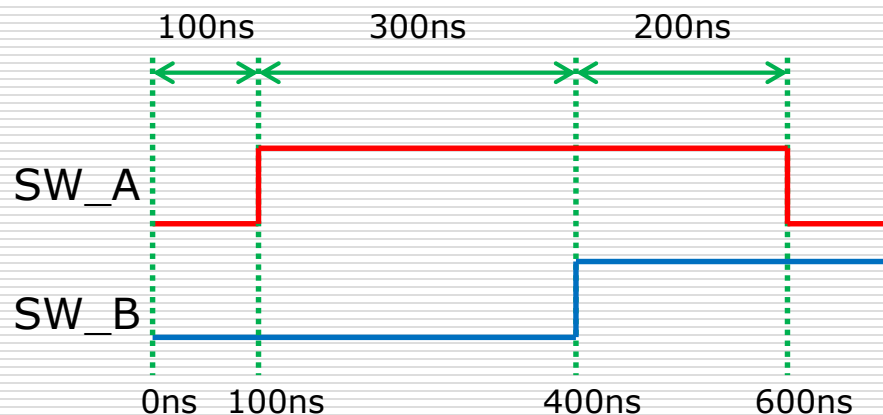
```

28
29 TEST uut(
30     .SW_A (SW_A),
31     .SW_B (SW_B),
32     .LED0 (LED0)
33 );
34
35 initial begin
36     SW_A = 1'b0;
37     SW_B = 1'b0;
38
39     #100 SW_A = 1'b1;
40     #300 SW_B = 1'b1;
41     #200 SW_A = 1'b0;
42 end
43
44 endmodule

```

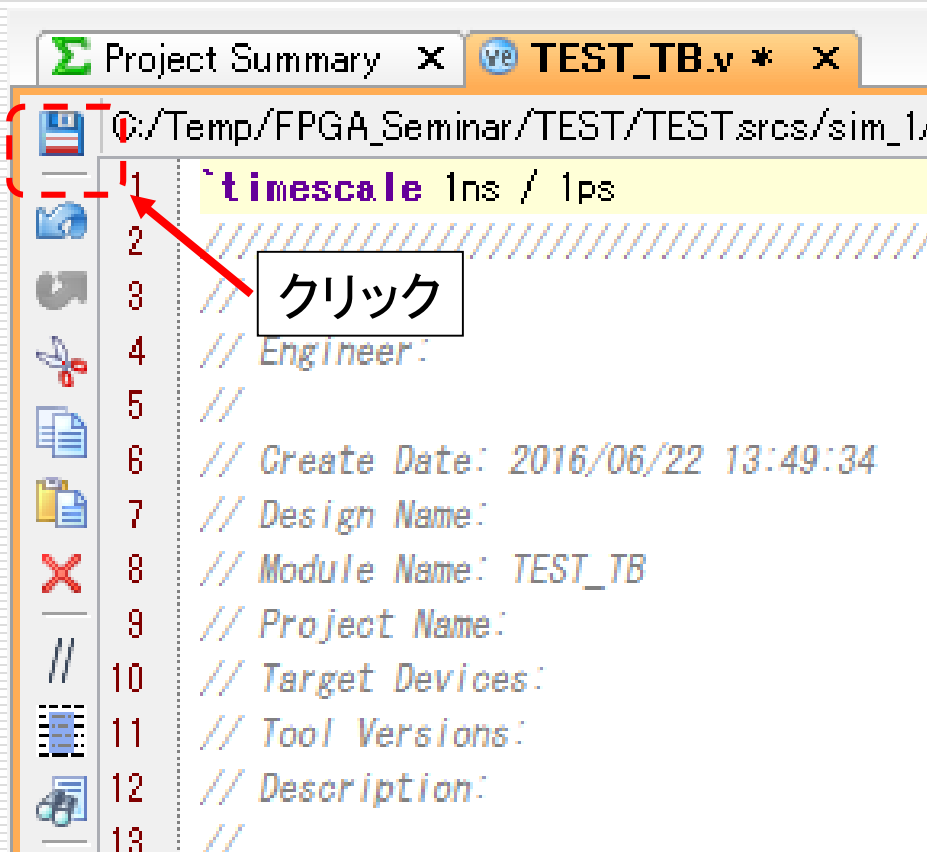
initial文 : initial begin ~ end  
 シミュレーション開始時に上から下に順番に一度だけ実行される  
 (シミュレーション専用文)

# 数字は遅延を表す  
 数字で指定されたユニット時間遅延する  
 今回ユニットの単位はナノ秒(1行目参照)なので  
 100nsの遅延が発生する

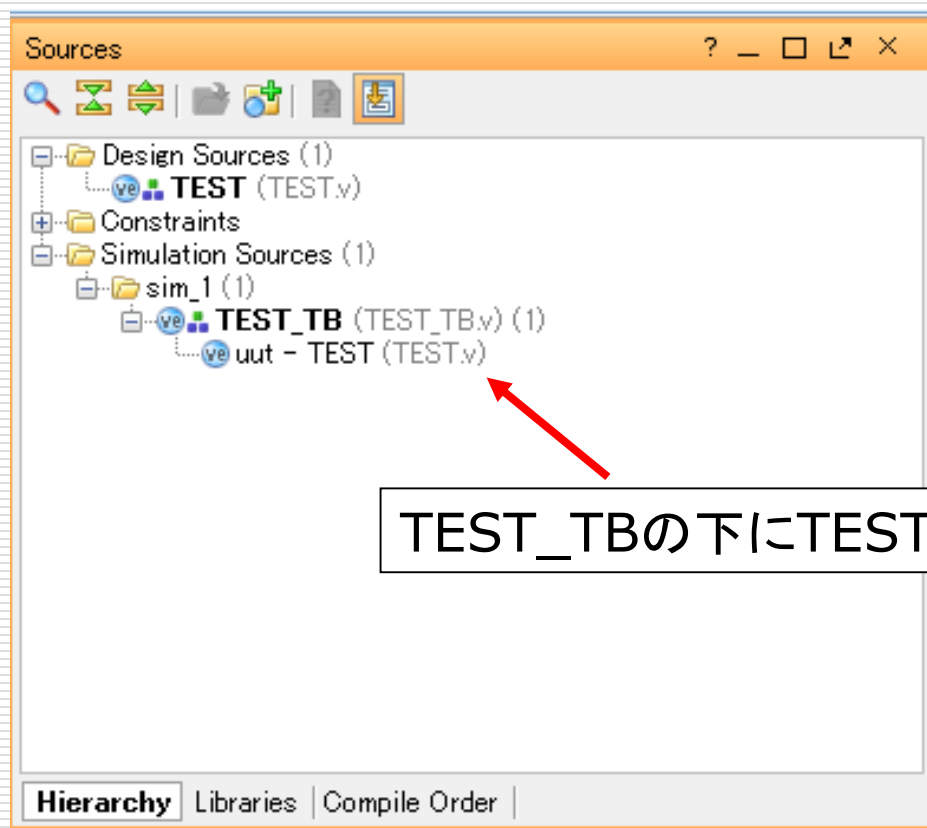


補足) initial文中の遅延を入れていない部分は同時に実行される、例えば36、37行。

# ファイルの保存

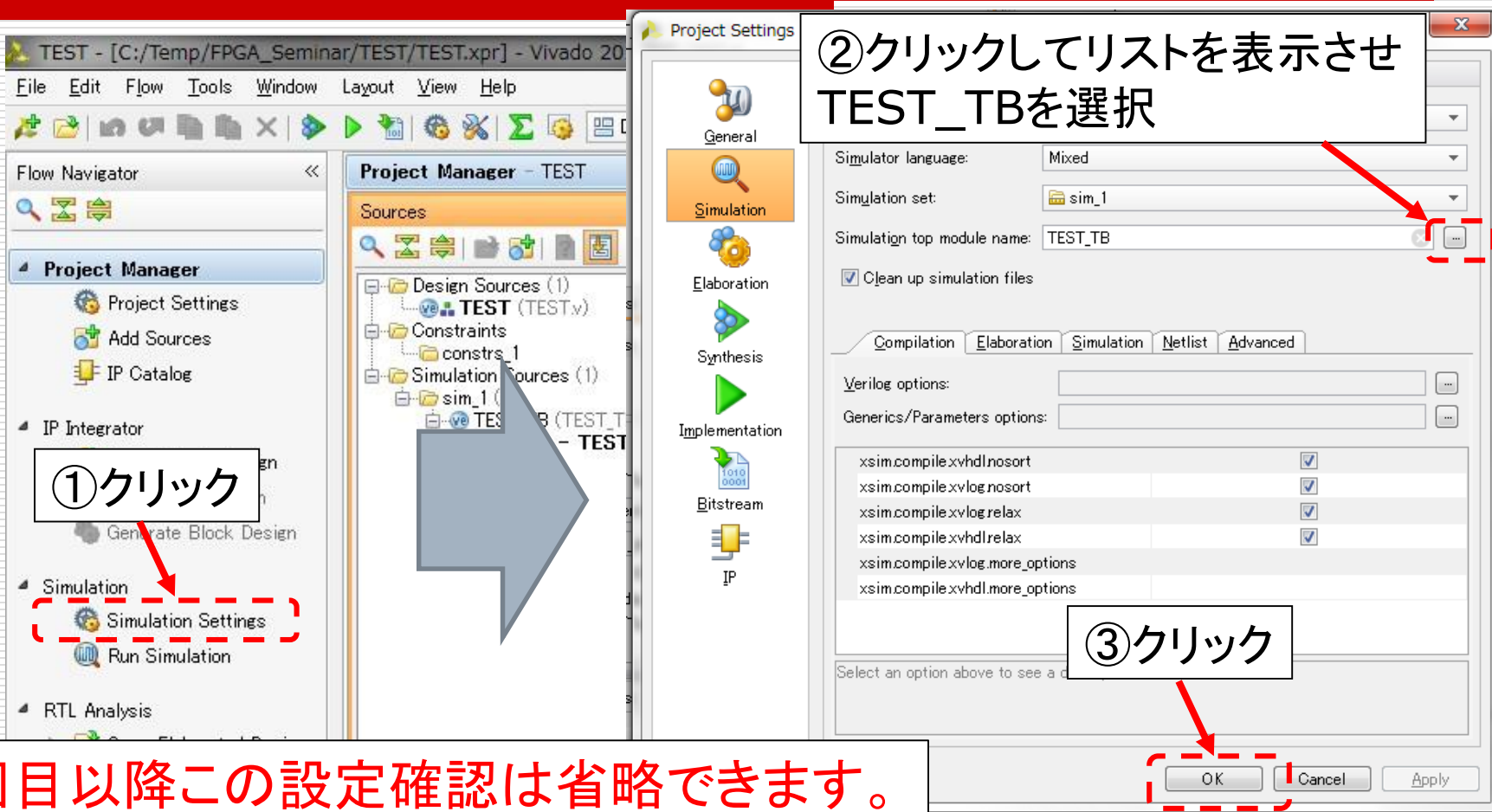


# 階層の確認



TEST\_TBの下にTESTが組み込まれた

# シミュレーション設定



①クリック

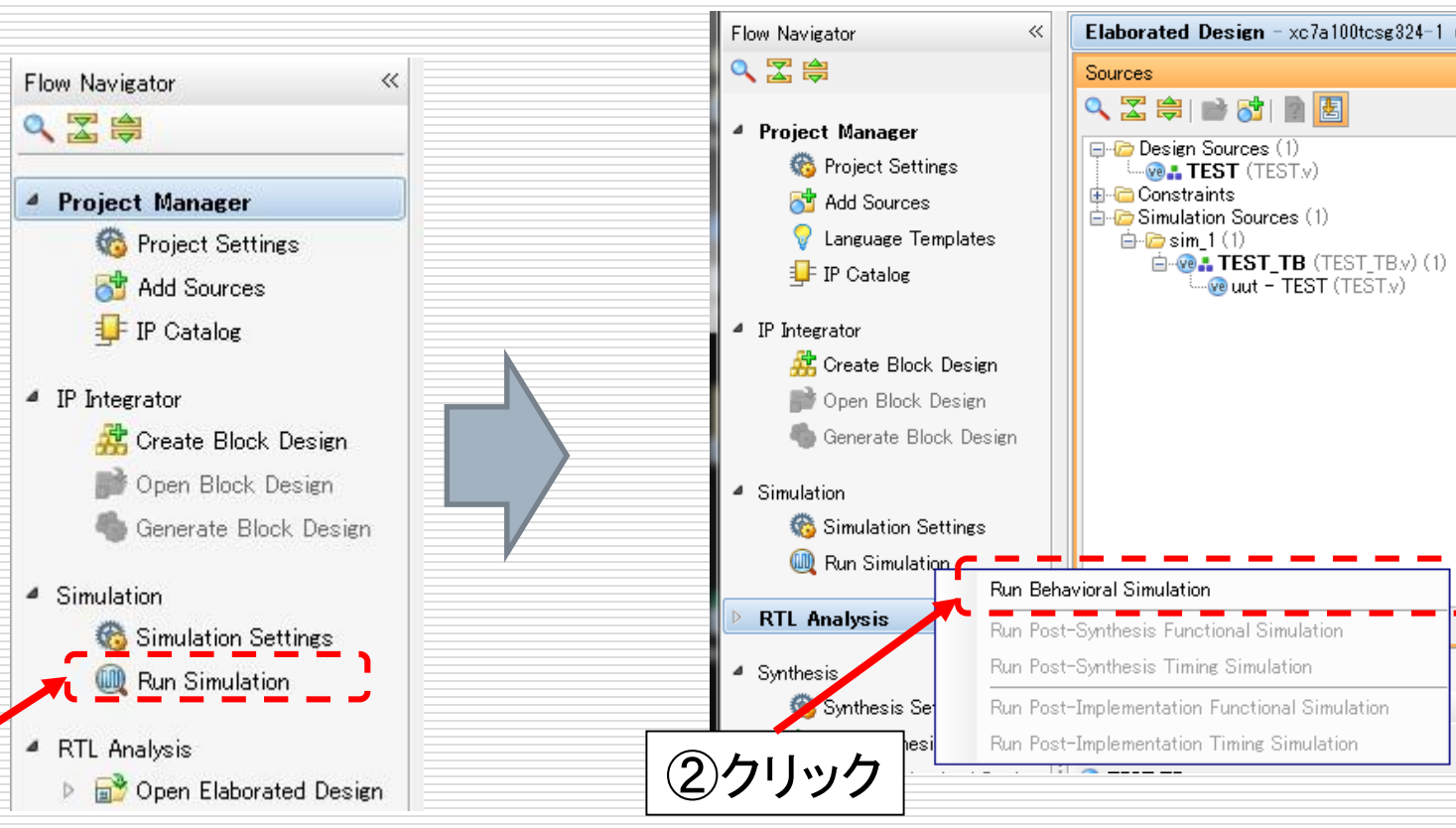
②クリックしてリストを表示させ TEST\_TBを選択

③クリック

2回目以降この設定確認は省略できます。

# シミュレータ起動

ファイアウォールの許可を求めてきたときは許可してください

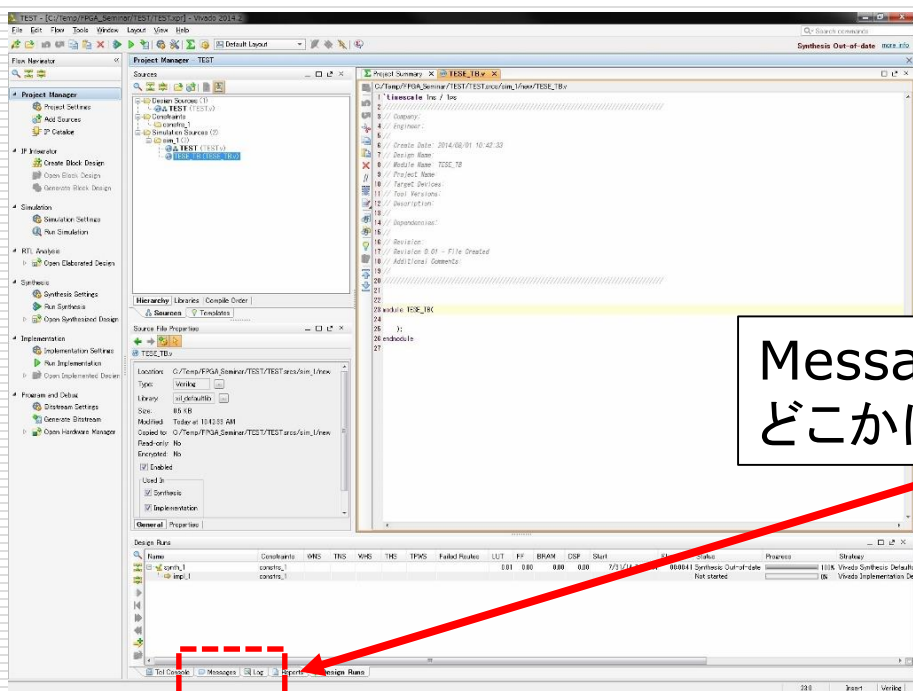


①クリック

②クリック

# シミュレータが起動しない場合 1

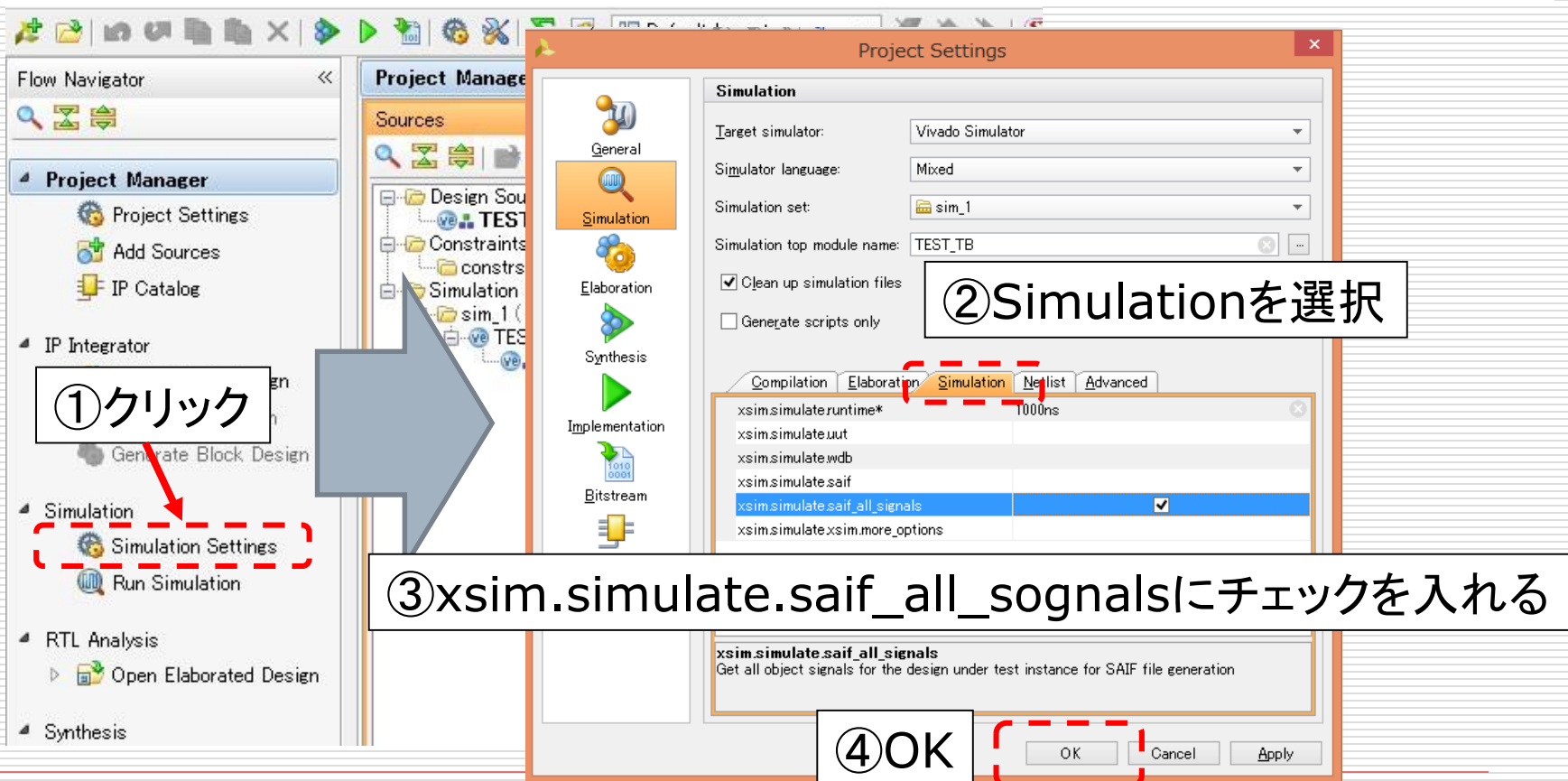
□ 多くの場合はテストベンチの書き間違いです。



Messageタブを選択して内容を確認  
どこかにエラーがあるはず

# シミュレーターが起動しない場合 2

エラーも何も表示されずにシミュレータが起動しない場合は下の設定を試してください



The image shows the Vivado Project Settings dialog box with the Simulation tab selected. The following steps are annotated:

- ①クリック**: A red dashed box highlights the "Simulation Settings" option in the left-hand Project Manager tree.
- ②Simulationを選択**: A red dashed box highlights the "Simulation" tab in the top navigation bar of the dialog box.
- ③xsim.simulate.saif\_all\_sognalsにチェックを入れる**: A red dashed box highlights the checkbox next to the "xsim.simulate.saif\_all\_sognals" option in the list of simulation options.
- ④OK**: A red dashed box highlights the "OK" button at the bottom of the dialog box.

The Simulation tab settings are as follows:

- Target simulator: Vivado Simulator
- Simulator language: Mixed
- Simulation set: sim\_1
- Simulation top module name: TEST\_TB
- Clean up simulation files
- Generate scripts only

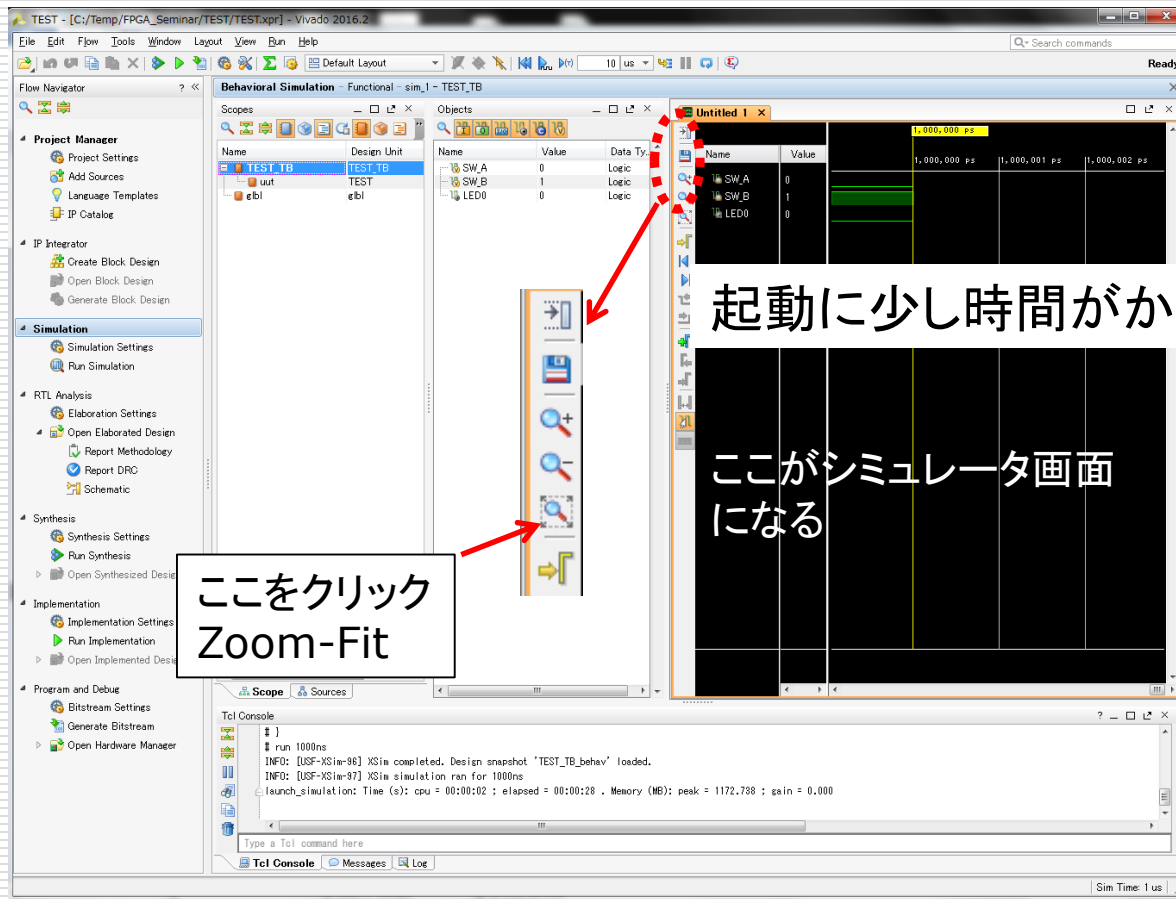
The list of simulation options includes:

- xsim.simulate.runtime\* (1000ns)
- xsim.simulate.uut
- xsim.simulate.wdb
- xsim.simulate.saif
- xsim.simulate.saif\_all\_sognals** (checked)
- xsim.simulate.xsim.more\_options

The description for the checked option is: **xsim.simulate.saif\_all\_sognals**  
Get all object signals for the design under test instance for SAIF file generation



# Vivadoシミュレータ画面



The screenshot shows the Vivado Behavioral Simulation window. The left sidebar contains the Project Manager, IP Integrator, Simulation, RTL Analysis, Synthesis, Implementation, and Program and Debug sections. The main area is divided into Scopes, Objects, and a waveform viewer. The Objects table is as follows:

Name	Value	Data Type
TEST_TB	TEST_TB	
TEST	TEST	
out		
SW_A	0	Logic
SW_B	1	Logic
LEDD0	0	Logic

The waveform viewer shows a signal trace for SW\_A, SW\_B, and LEDD0 over time. Annotations include:

- A red dashed circle around the simulation controls in the top right.
- A red arrow pointing to the Zoom-Fit button in the bottom right toolbar, with a callout box: **ここをクリック Zoom-Fit**.
- A white callout box: **起動に少し時間がかかります** (It takes a little time to start).
- A white callout box: **ここがシミュレータ画面になる** (This is where the simulator screen becomes).

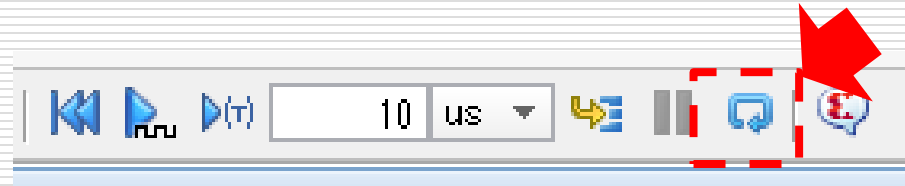
The Tcl Console at the bottom shows the following output:

```

run 1000ns
INFO: [DUSF-XSIm-96] XSIm completed. Design snapshot 'TEST_TB_behav' loaded.
INFO: [DUSF-XSIm-97] XSIm simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:02 ; elapsed = 00:00:28 . Memory (MB): peak = 1172.738 ; cain = 0.000
  
```

# コードを修正した時

- コードを修正した場合、再度シミュレーション用中間データを生成する必要がありますので再起動が必要です
- ツールバーの再起動 (relaunch) を実行してください

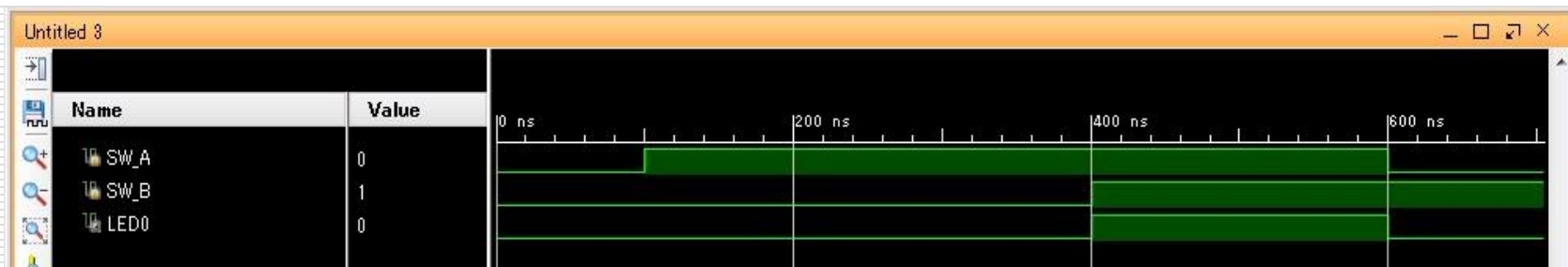


- 再起動すると確認窓が表示されますのでOKを選択してください

# シミュレーション結果

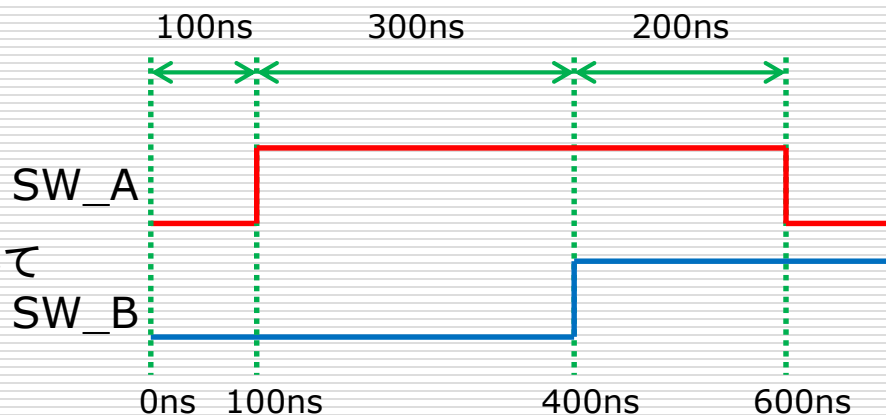
シミュレーション全体が見れます

ISimを起動するとデフォルトで1000ns分シミュレーションを行います

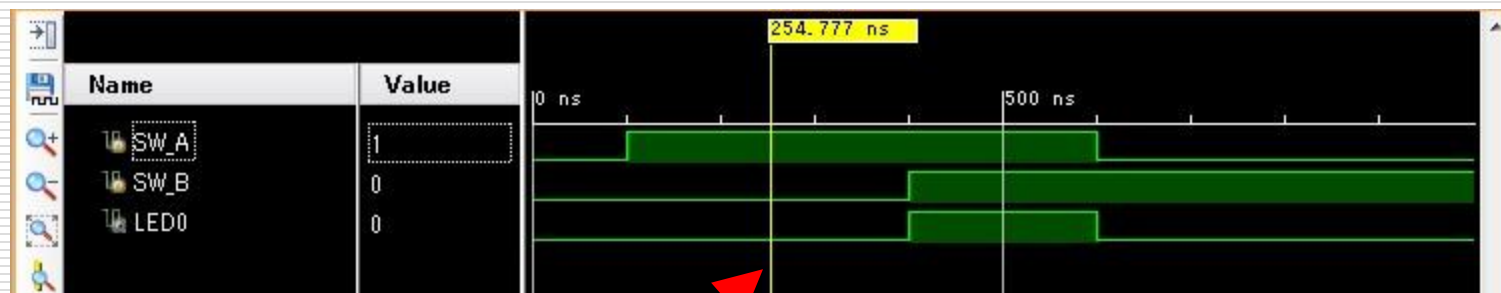


次にシミュレーションが正しいかどうかを確認します

ここで正しいとは  
 自分の思った通りに信号が入力されていて  
 設計した通りに出力しているか  
 ということです



# カーソルの表示



①この窓で左クリックすると黄色のカーソルが表示されます

# カーソルを波形のエッジに合わせて合わせる

カーソルの上で左クリックしたままマウスを移動させると  
カーソルが左右に動く

カーソルを100ns付近のSW\_Aの立ち上がりエッジに合わせてみます



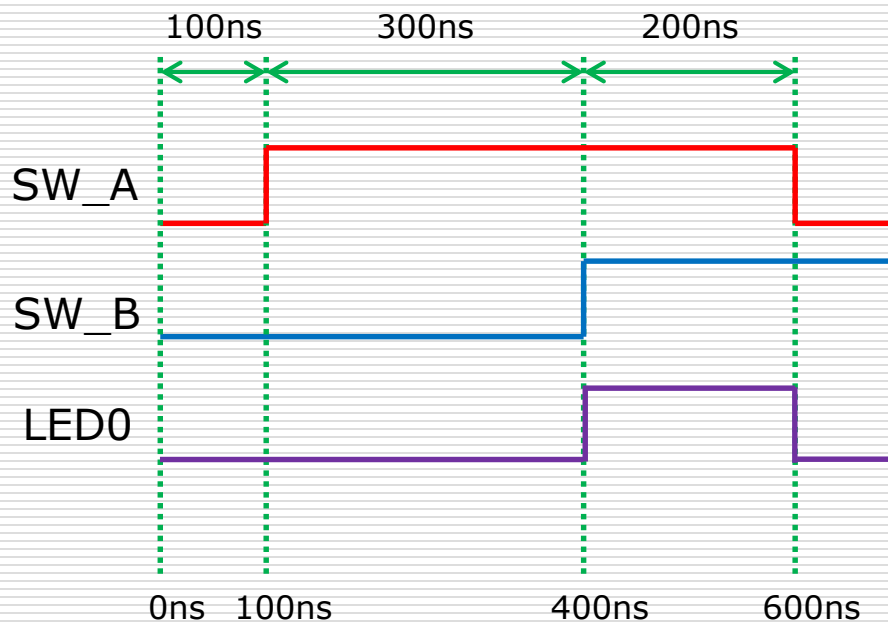
①カーソルがエッジに合うと円の中が白く塗りつぶされる

②この状態でクリックを止めてください

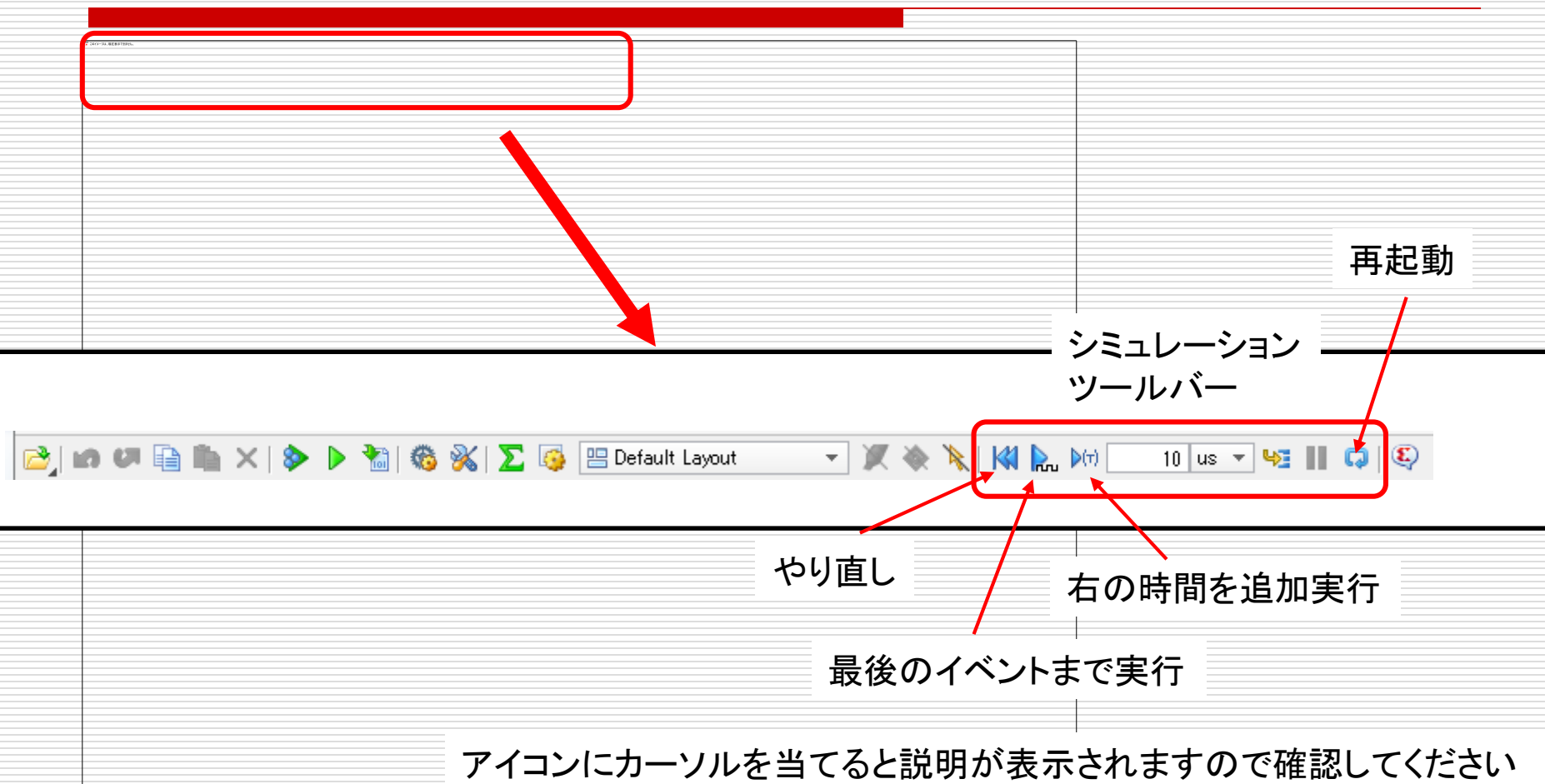
③100.000nsと表示されるはずですが

# シミュレーション結果の確認

カーソルを使って入力波形が以下のように生成されていることを確認してください。



# ツールバー



The image shows a software interface with a toolbar. A red box highlights a search field at the top left. A red arrow points from this box to the simulation toolbar. The toolbar contains various icons for simulation control. A red box highlights a specific set of icons: a double left arrow, a play button, a right arrow, a text input field containing '10 us', a refresh icon, and a help icon. Red arrows point from text labels to these icons.

再起動

シミュレーション  
ツールバー

やり直し

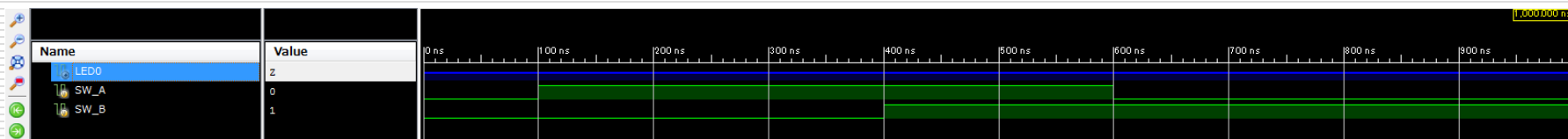
最後のイベントまで実行

右の時間を追加実行

アイコンにカーソルを当てると説明が表示されますので確認してください

## その他のエラーとして

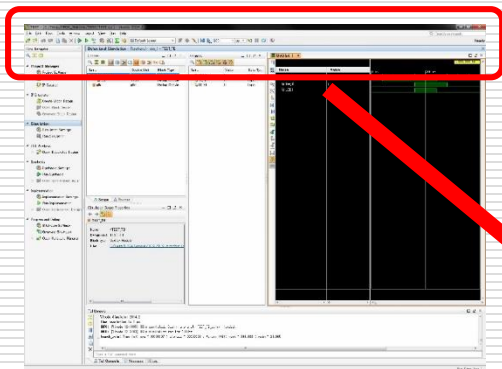
- 駆動されていない信号は下ののように青くなります。多くの場合、未接続、接続するのを忘れている。



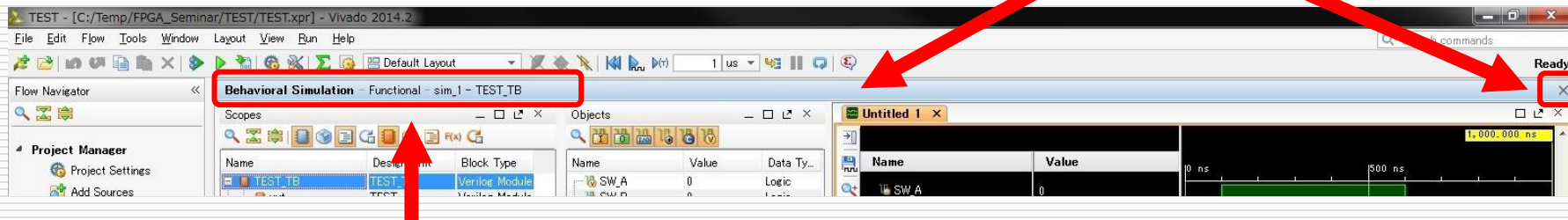
- この場合エラーにならないので自分で間違いを探さなければいけません
  - エラーにならない理由: 意図して未接続なのか意図せず未接続なのかツールは判定できない



# シミュレータの終了



②この帯で右クリック→close  
または  
右端のxをクリック



①[Behavioral Simulation]と表示されていることを確認

# 参考文献

---

- Xilinx, UG937, Vivado Design Suite Tutorial: Logic Simulation

# 履歴

---

- 2012/5/17 第1.0版 ISE13.4対応 内田智久(Esys, KEK/総研大), 林達也(大阪大学)
- 2014/8/7 第2.0版 Vivado2014対応、章構成変更 内田智久(Esys, KEK/総研大)
- 2015/7/31 第3.0版 Vivado2015対応、章構成変更 内田智久(Esys, KEK/総研大)
- 2015/12/03 第3.1版 誤字訂正 内田智久(Esys, KEK/総研大)
- 2016/1/27 第3.2版 Vivado2015.4対応、内田智久(Esys KEK/総研大)
- 2016/6/22 第3.3版 Vivado2016.2対応、内田智久(Esys KEK/総研大)