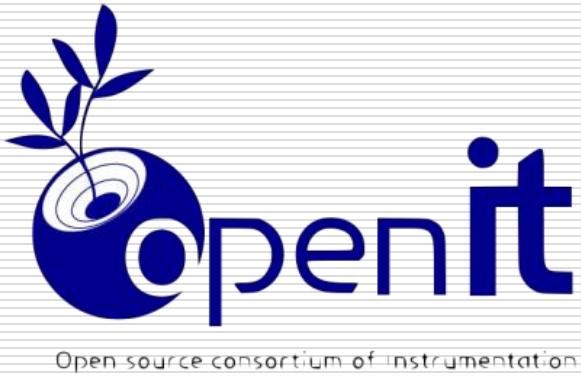
Open-It FPGAトレーニングコース（入門編）

2.2 HDL入力とRTL解析



第3.4版

2016年06月22日

注意

- 実習時に入力するコードなどの内容は「実習時の参考資料」を参照してください
  - スライドでは細かい部分が分かりにくいので。

Vivado

- VivadoはXilinx社のFPGA開発統合環境
  - 様々なツールを一つの窓から操作できる
- Vivadoを使って回路入力やFPGAデータ生成を行います。

注意：
Windowsアカウントのユーザー名

- 英字のアカウントで作業してください
- 日本語が入っていると正しく動作しない時があります

プロジェクト作成

作業をするめに必要な環境を設定します

作業内容

- 作業ディレクトリの作成
- Vivadoプロジェクトの作成
  - 作業ディレクトリの指定
  - 使用デバイスの指定

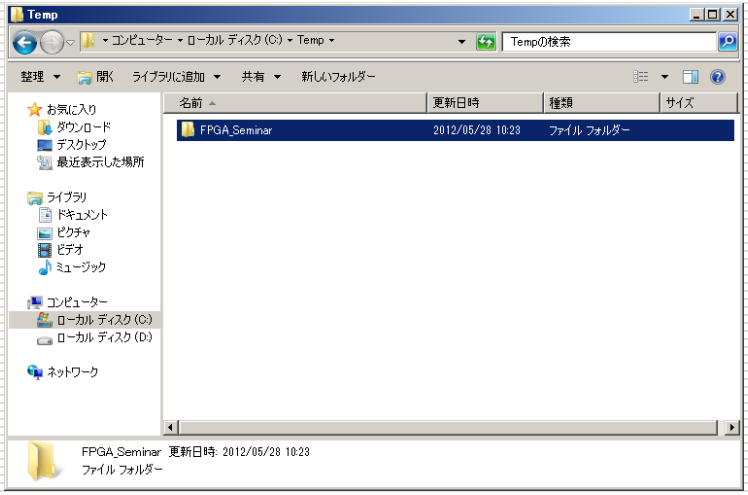
プロジェクトは一つのFPGAで一つ作ります

プロジェクトについて

- FPGA単位で一つ作ります
  - 機能が異なる回路は異なるプロジェクトとしてください。
  - 機能が同じでも使用するFPGAの型番が異なる場合は異なるプロジェクトとしてください。

作業ディレクトリの作成

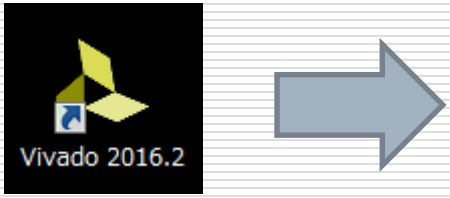
- 以下を演習用のディレクトリとします
  - C:¥Temp¥FPGA_Seminarを作成して下さい



- この資料では上を作業ディレクトリして使用しますが、他のフォルダでも構いません。
- フォルダ名に日本語やスペースがあるディレクトリは避けた方がよいです

Vivadoを起動しましょう！

起動に少し時間がかかります



アイコンをダブルクリック！

または、

Windows 7:
全てのプログラム→
Xilinx Design Tools →
Vivado 2015.4 →
Vivado 2015.4

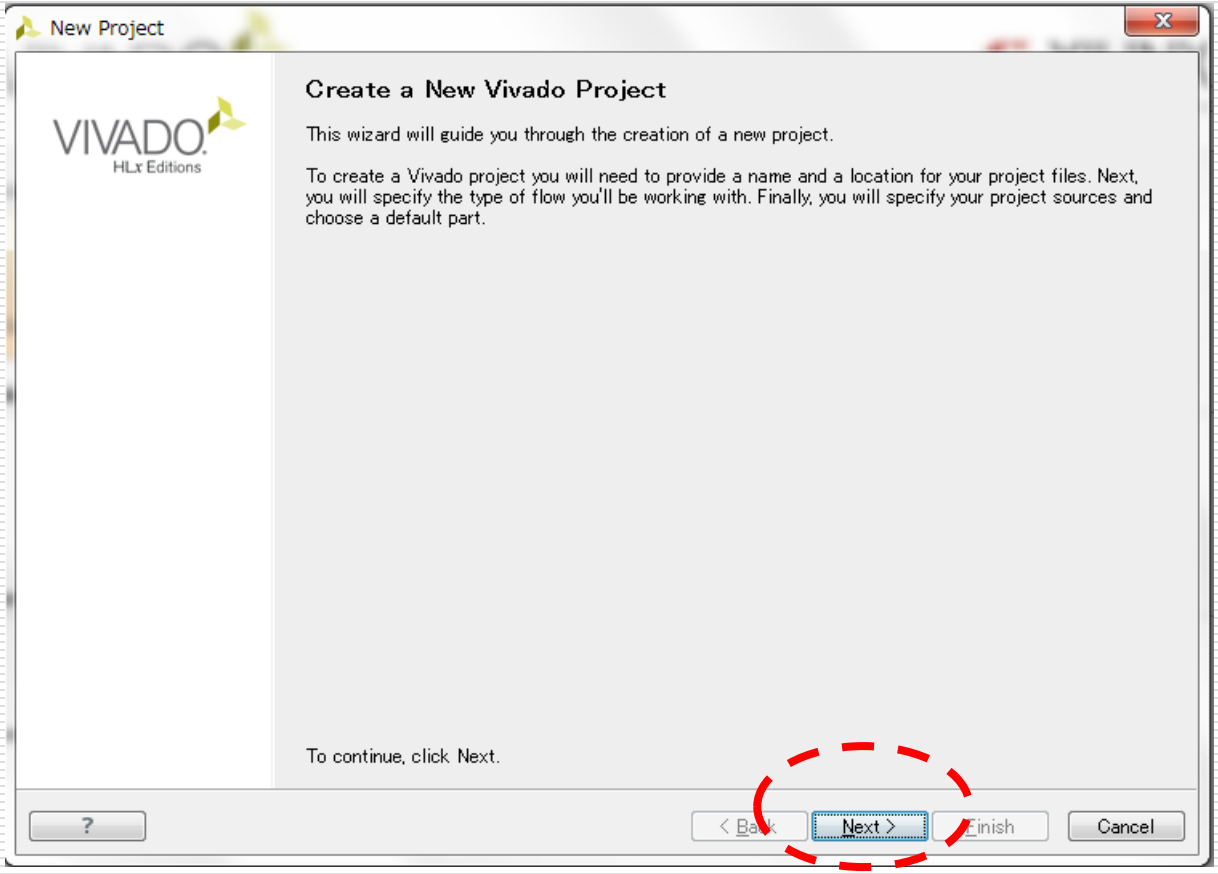
Windows8:
Vivadoで検索

Vivadoプロジェクトの作成

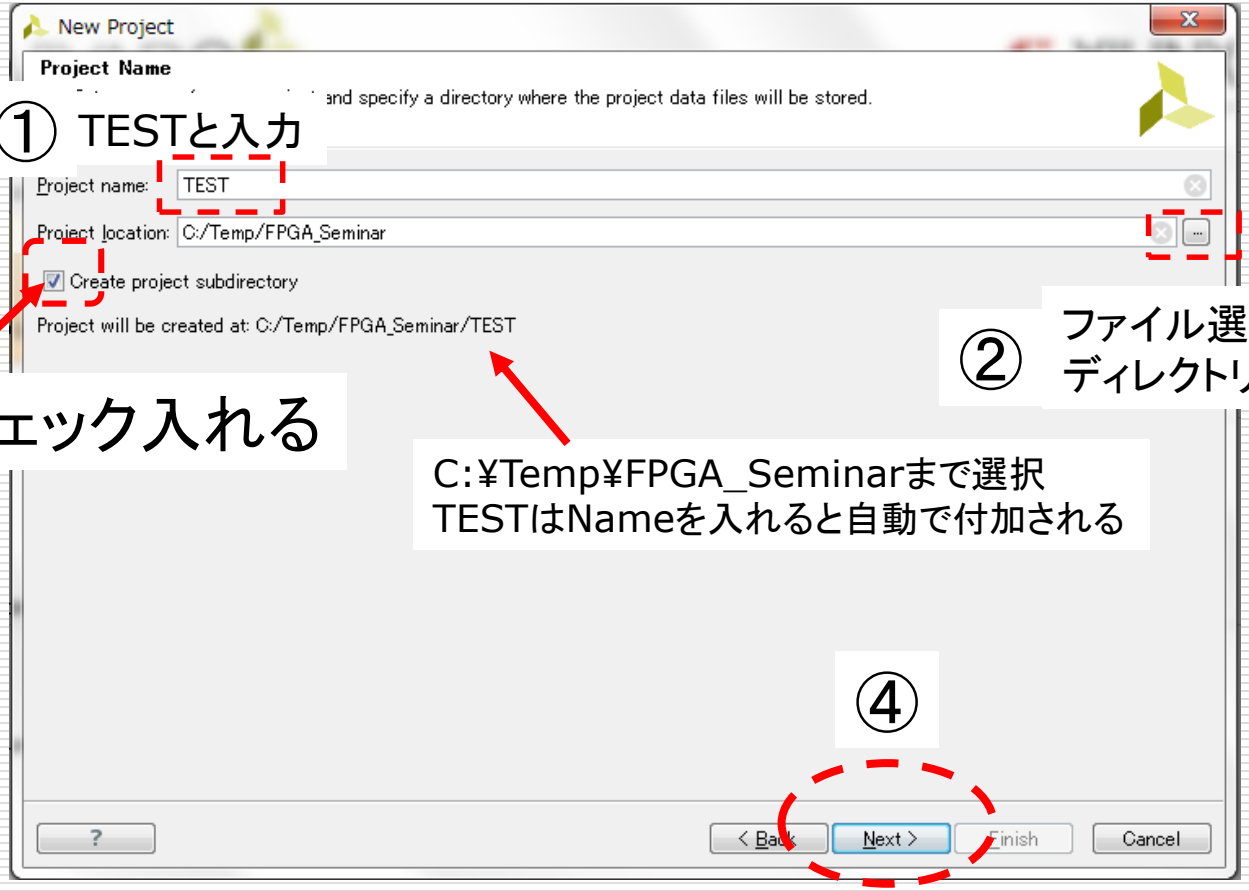
Create New Project
をクリック

（補足）既存のプロジェクトを開くときはOpen Project

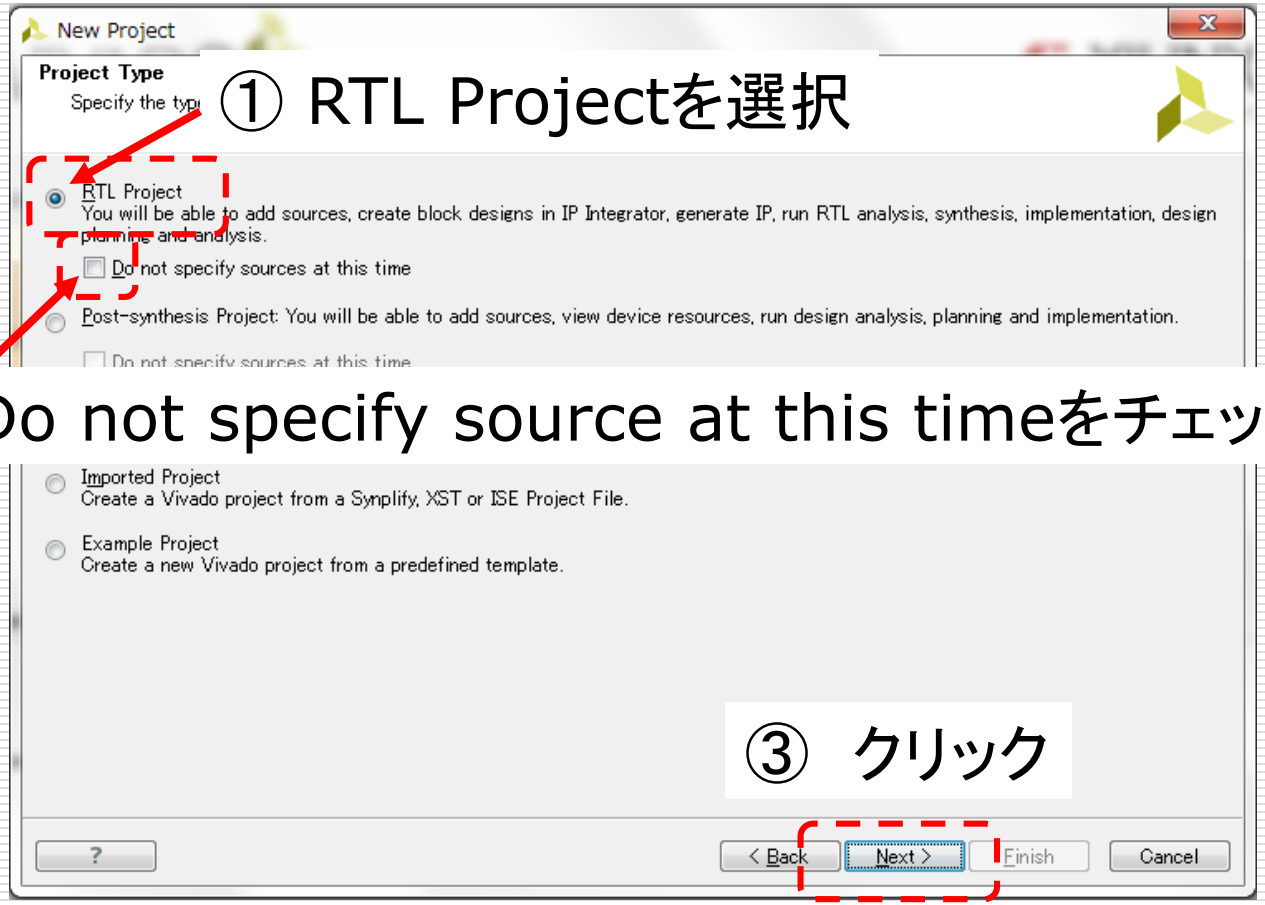
ウィザードの開始


New Project
VIVADO
HLx Editions
Create a New Vivado Project
This wizard will guide you through the creation of a new project.
To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.
To continue, click Next.
?
< Back
Next >
Finish
Cancel

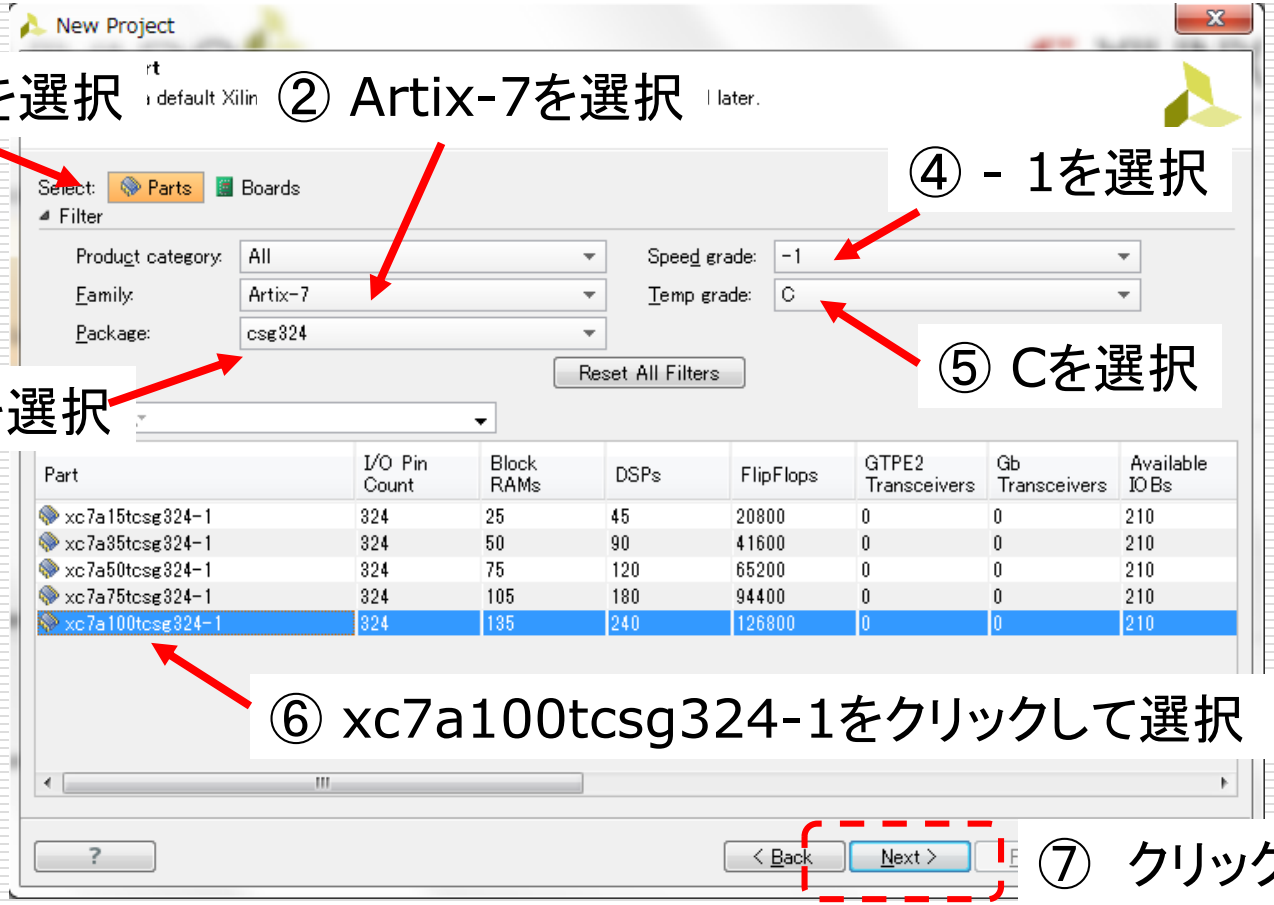
プロジェクト名と作業フォルダの指定



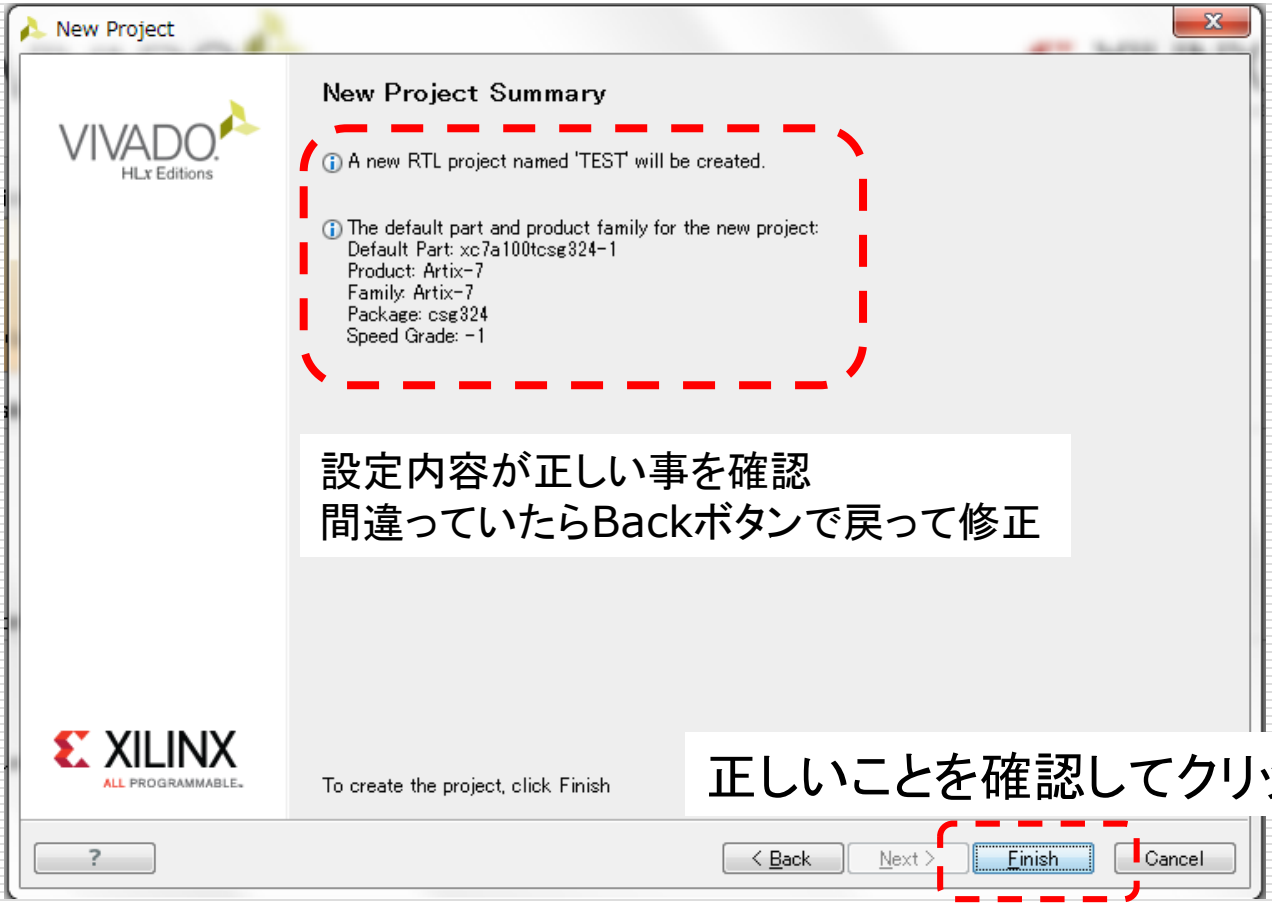
プロジェクトの種類を選択



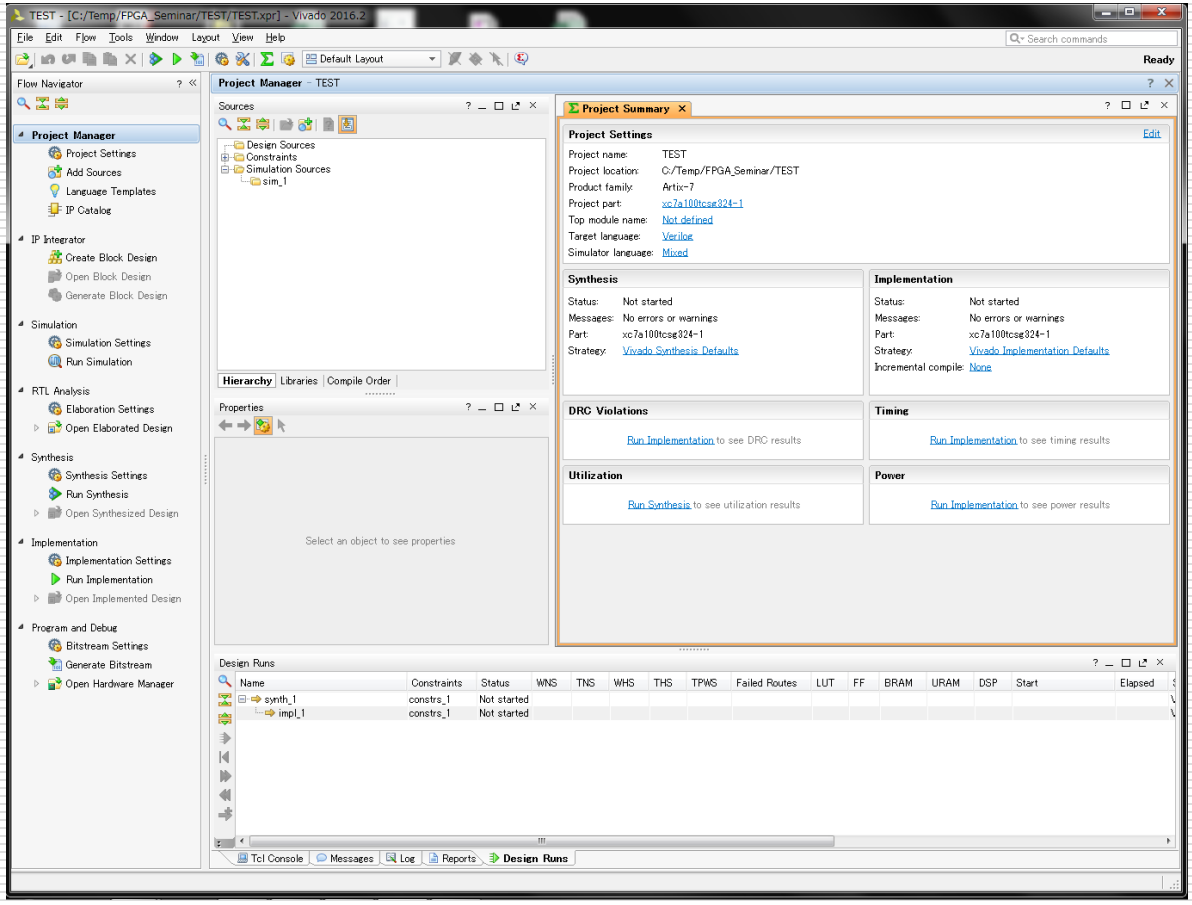
使用するデバイスの指定



入力確認


New Project
VIVADO
HLx Editions
New Project Summary
A new RTL project named 'TEST' will be created.
The default part and product family for the new project:
Default Part: xc7a100tcsg324-1
Product: Artix-7
Family: Artix-7
Package: csg324
Speed Grade: -1
設定内容が正しい事を確認
間違っていたらBackボタンで戻って修正
XILINX
ALL PROGRAMMABLE.
To create the project, click Finish
正しいことを確認してクリック
?
< Back
Next >
Finish
Cancel

プロジェクト作成終了


TEST - [C:/Temp/FPGA_Seminar/TEST/TEST.xpr] - Vivado 2016.2
File Edit Flow Tools Window Layout View Help
Default Layout
Ready
Flow Navigator
Project Manager
Project Settings
Add Sources
Language Templates
IP Catalog
IP Integrator
Create Block Design
Open Block Design
Generate Block Design
Simulation
Simulation Settings
Run Simulation
RTL Analysis
Elaboration Settings
Open Elaborated Design
Synthesis
Synthesis Settings
Run Synthesis
Open Synthesized Design
Implementation
Implementation Settings
Run Implementation
Open Implemented Design
Program and Debug
Bitstream Settings
Generate Bitstream
Open Hardware Manager
Project Manager - TEST
Sources
Design Sources
Constraints
Simulation Sources
sim_1
Hierarchy Libraries Compile Order
Properties
Select an object to see properties
Project Summary
Project Settings
Edit
Project name: TEST
Project location: C:/Temp/FPGA_Seminar/TEST
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: Not defined
Target language: Verilog
Simulator language: Mixed
Synthesis
Status: Not started
Messages: No errors or warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Implementation
Status: Not started
Messages: No errors or warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Implementation Defaults
Incremental compile: None
DRC Violations
Run Implementation to see DRC results
Timing
Run Implementation to see timing results
Utilization
Run Synthesis to see utilization results
Power
Run Implementation to see power results
Design Runs
Name Constraints Status WNS TNS WHS THS TPWS Failed Routes LUT FF BRAM URAM DSP Start Elapsed
synth_1 constrs_1 Not started
impl_1 constrs_1 Not started
Tcl Console Messages Log Reports Design Runs

Verilog-HDLモジュール作成

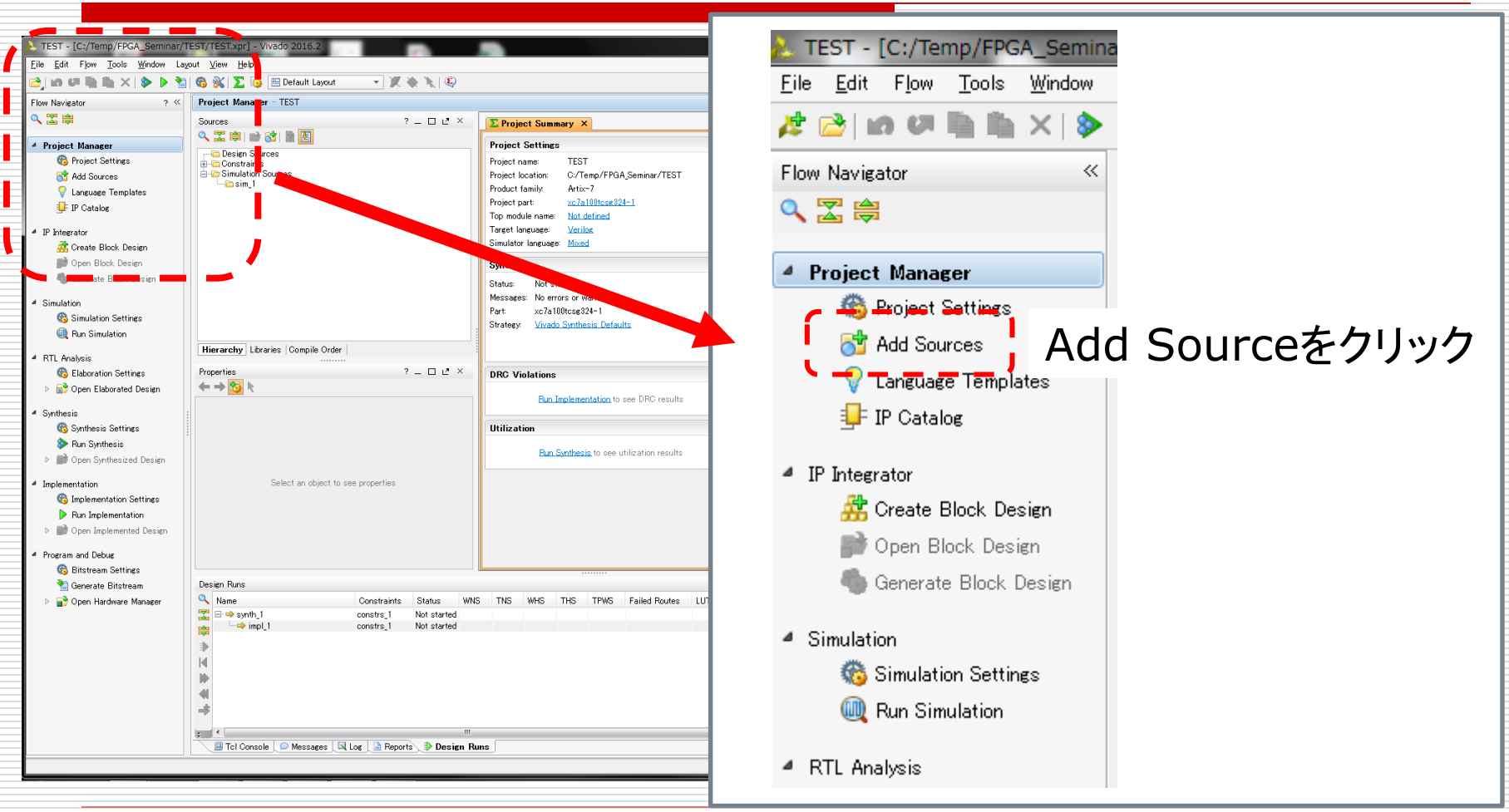
- AND回路をHDLで記述します。

- ファイル名：TEST.v
- モジュール名：TEST

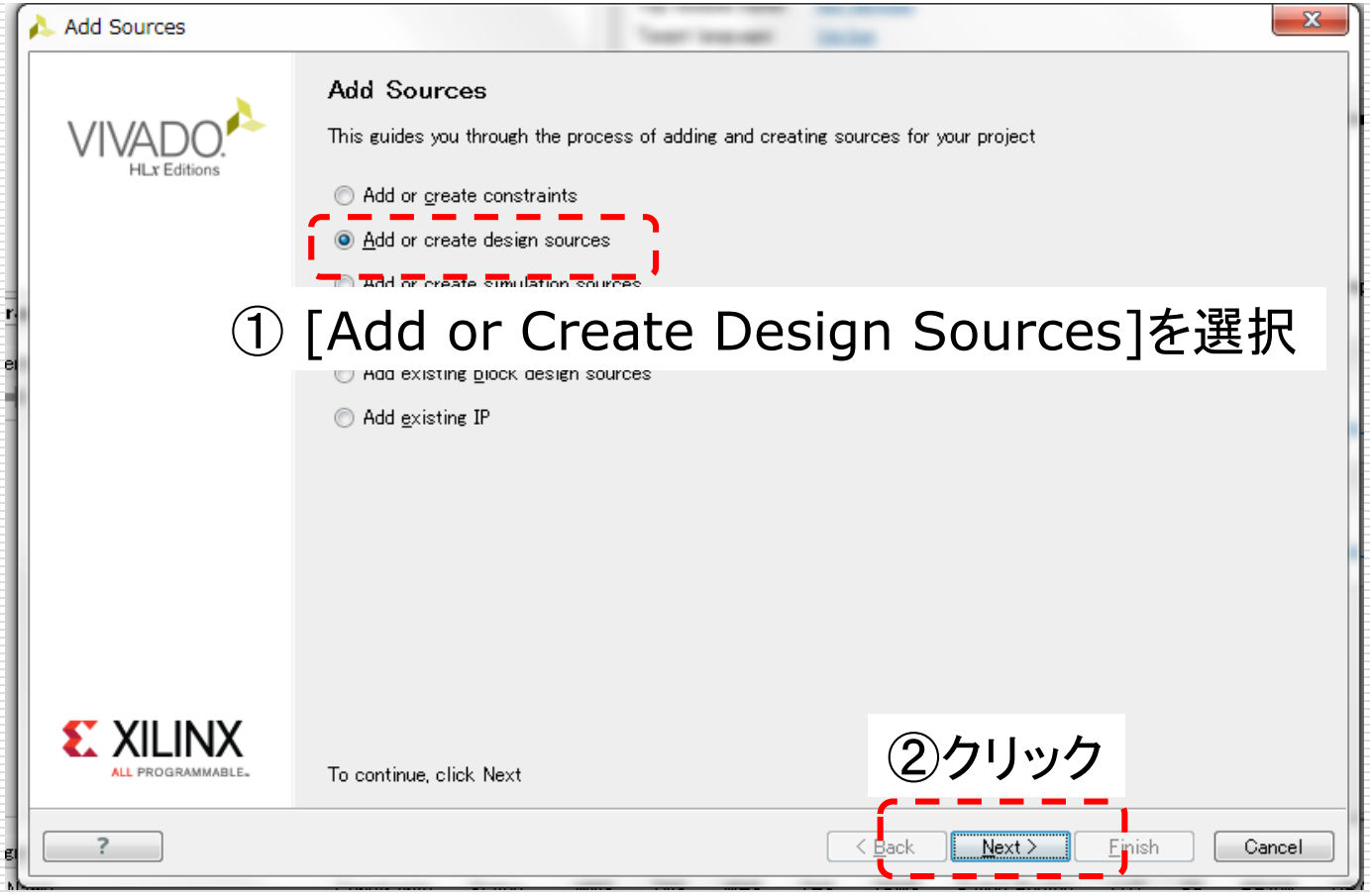
Vivadoに付属しているテキストエディターを使用して編集します。
HDLキーワードは文字が青くなるので打ち間違いなどに気が付きやすくなる利点があります

ちなみに、サクラエディタ、Emacs、ViなどでもVerilog-HDLキーワード設定ができます。

新規Verilogモジュールの追加

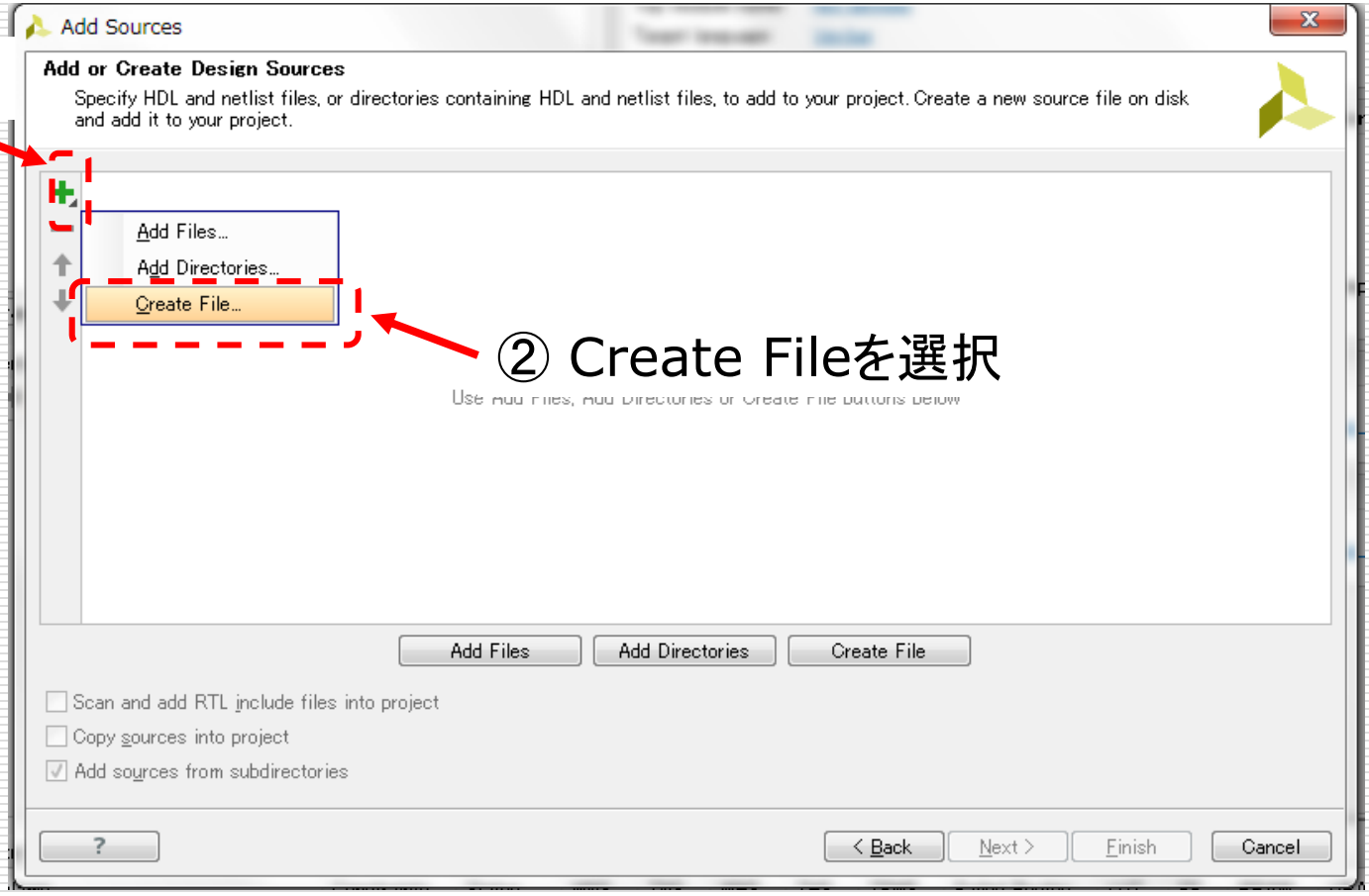


ウィザード開始

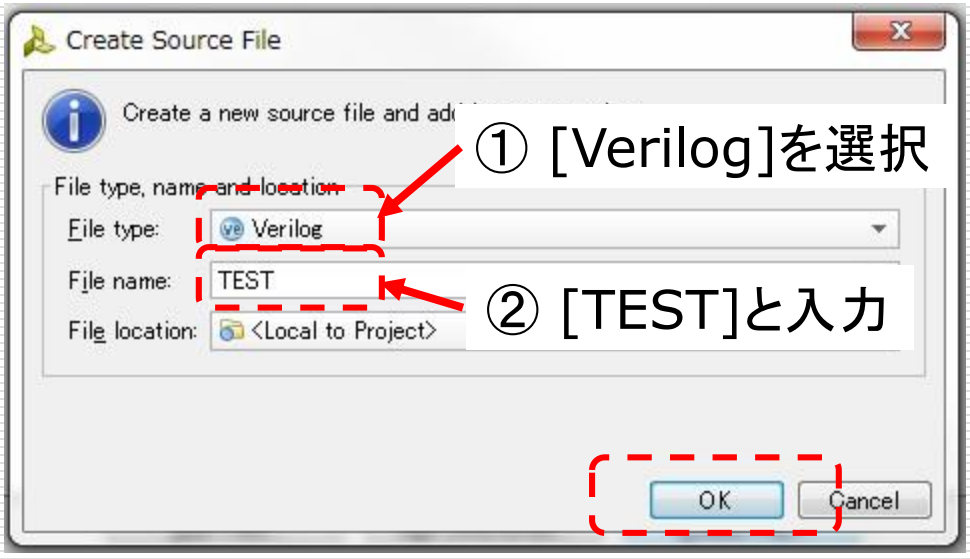

Add Sources
VIVADO
HLx Editions
Add Sources
This guides you through the process of adding and creating sources for your project
Add or create constraints
Add or create design sources
Add existing IP
① [Add or Create Design Sources]を選択
XILINX
ALL PROGRAMMABLE
To continue, click Next
②クリック
?
< Back
Next >
Finish
Cancel

新規ファイル作成

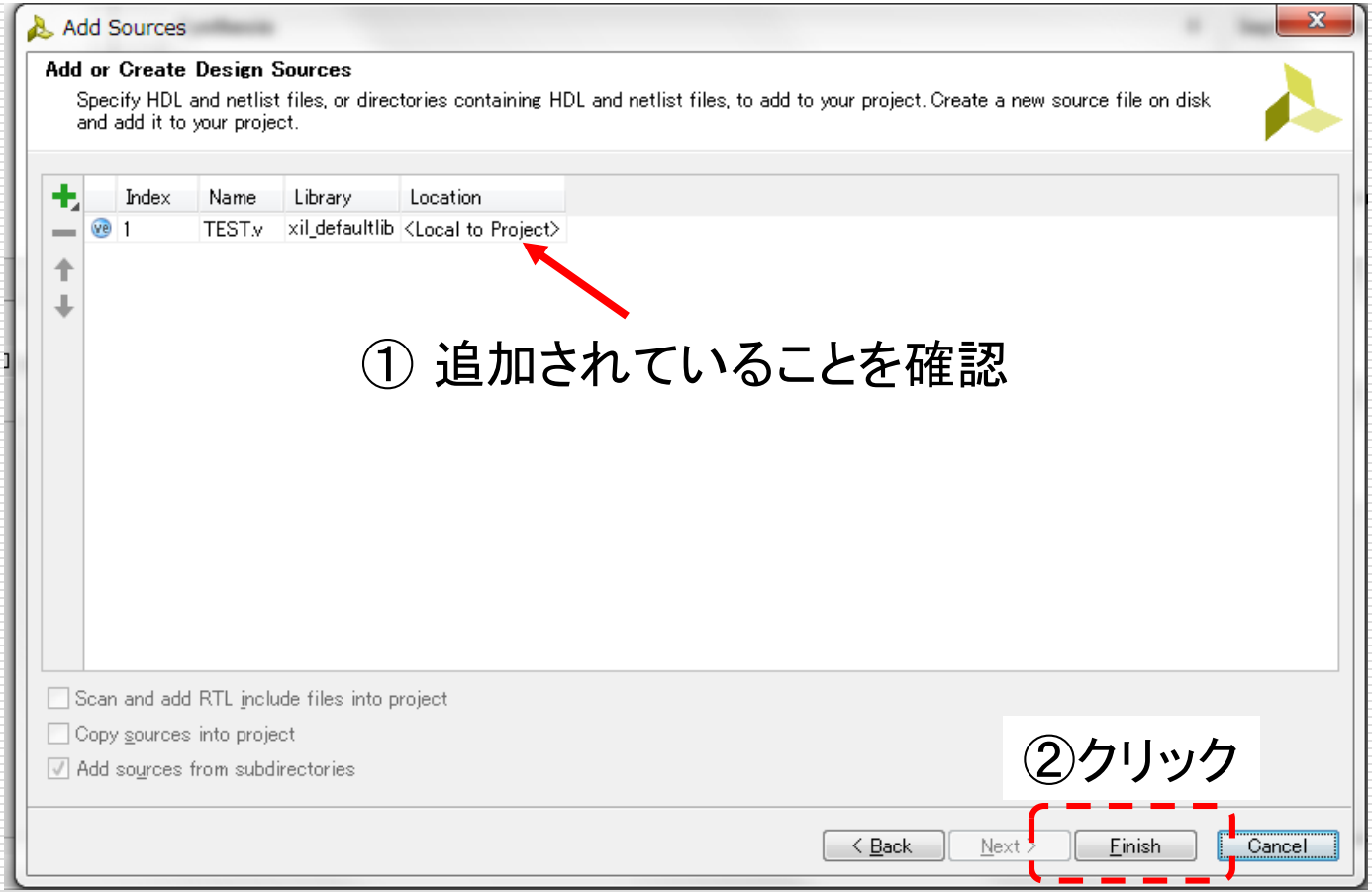
①クリック



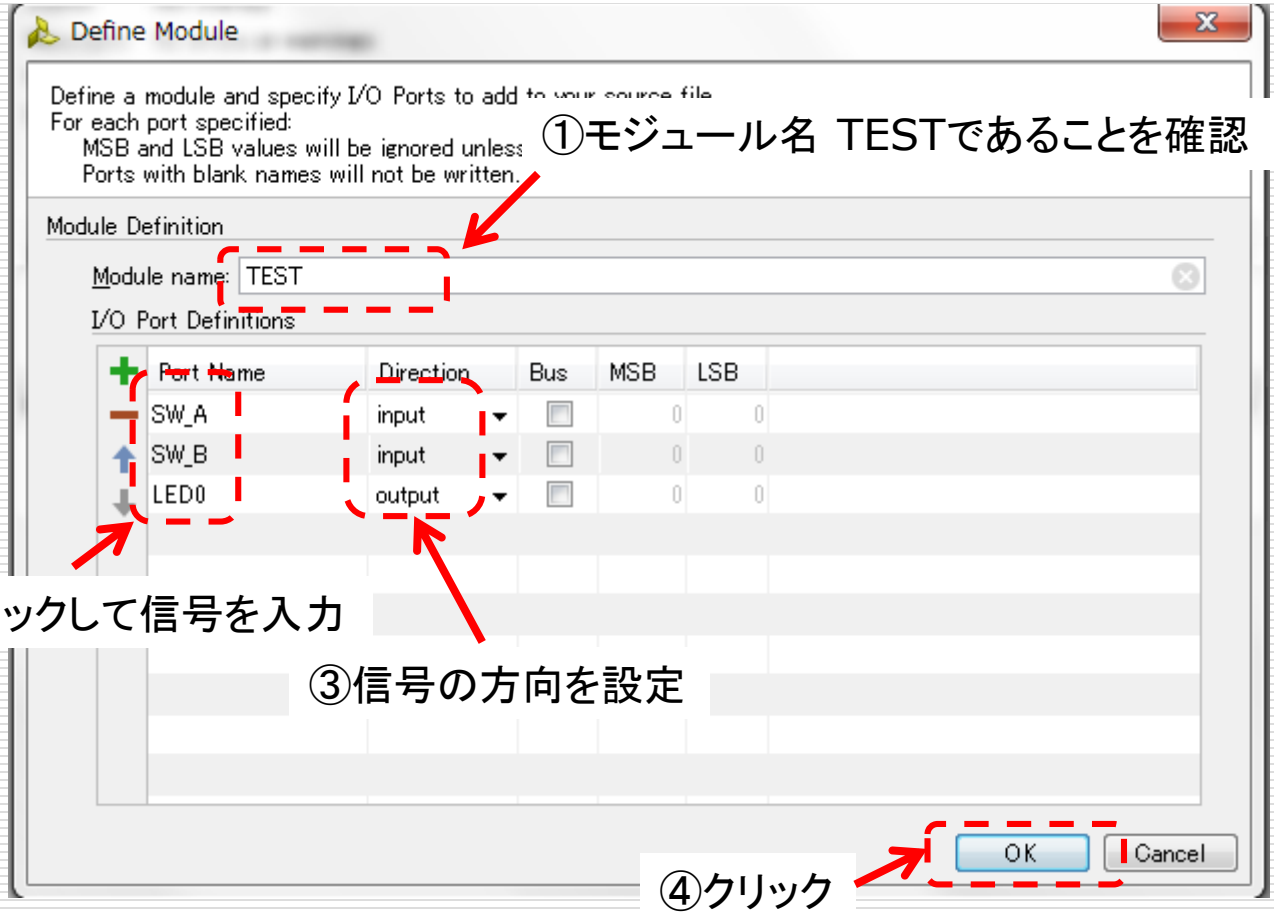
ファイル名の指定


Create Source File
Create a new source file and ad
File type, name and location
File type:
Verilog
File name:
TEST
File location:
<Local to Project>
OK
Cancel
① [Verilog]を選択
② [TEST]と入力
③クリック

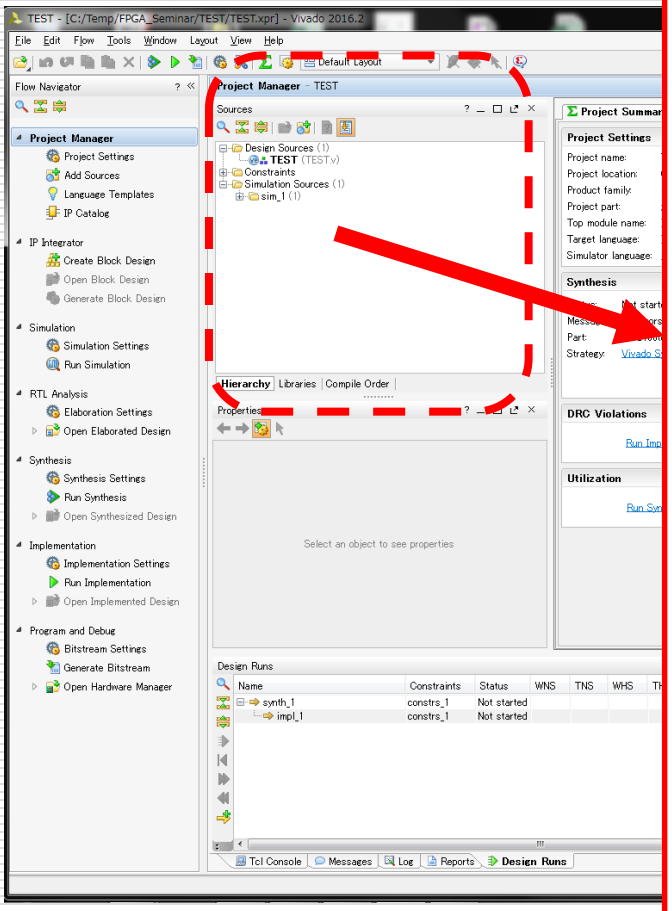
ファイル名などの確認

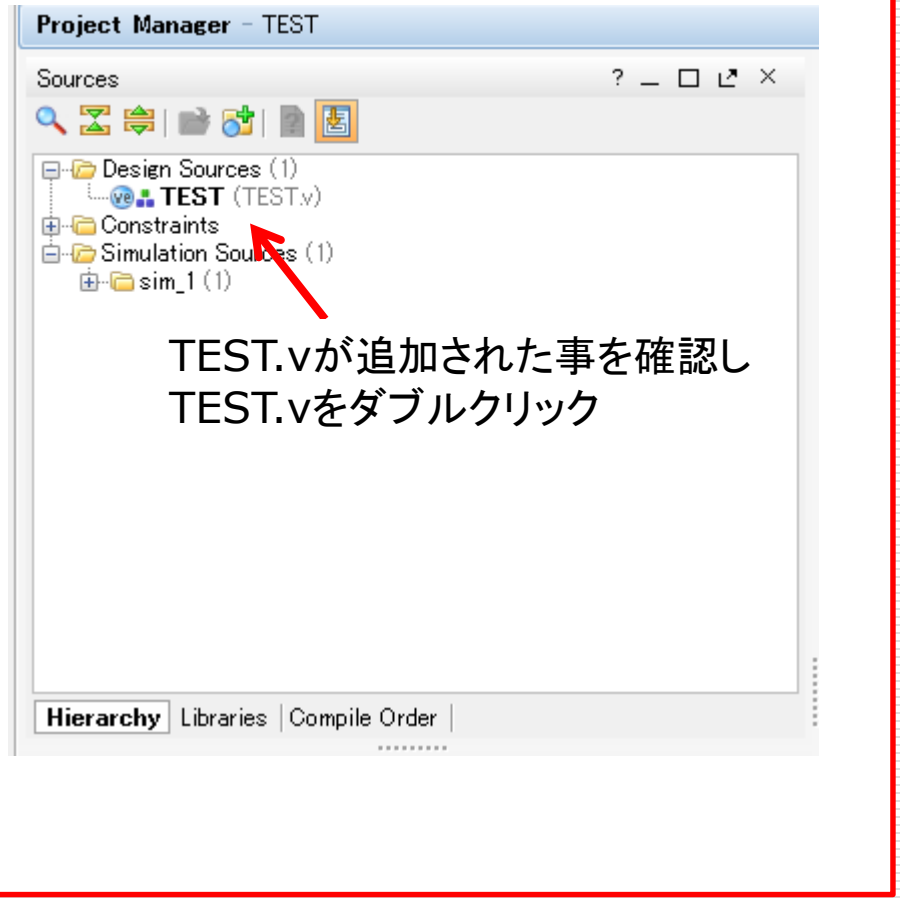

Add Sources
Add or Create Design Sources
Specify HDL and netlist files, or directories containing HDL and netlist files, to add to your project. Create a new source file on disk and add it to your project.
Index
Name
Library
Location
1
TEST.v
xil_defaultlib
<Local to Project>
① 追加されていることを確認
Scan and add RTL include files into project
Copy sources into project
Add sources from subdirectories
②クリック
< Back
Next >
Finish
Cancel

モジュール名とポート名の指定

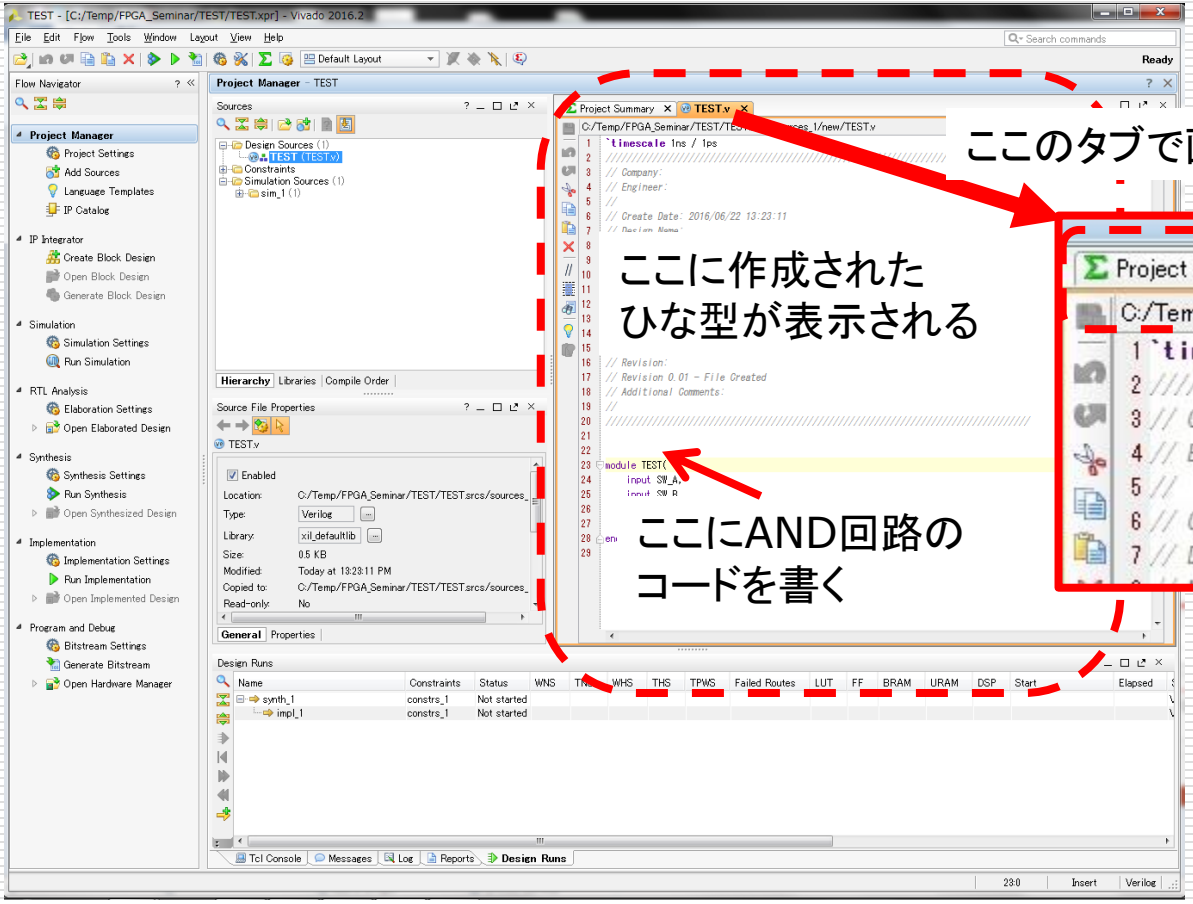


TEST.vがDesign Sourceに追加された事を確認

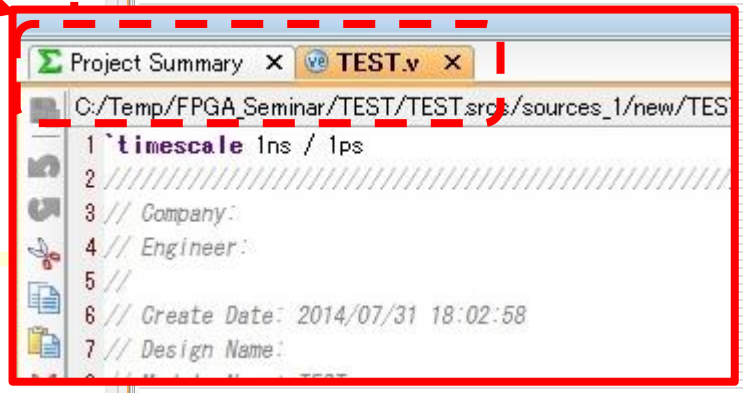




TEST.vの内容表示

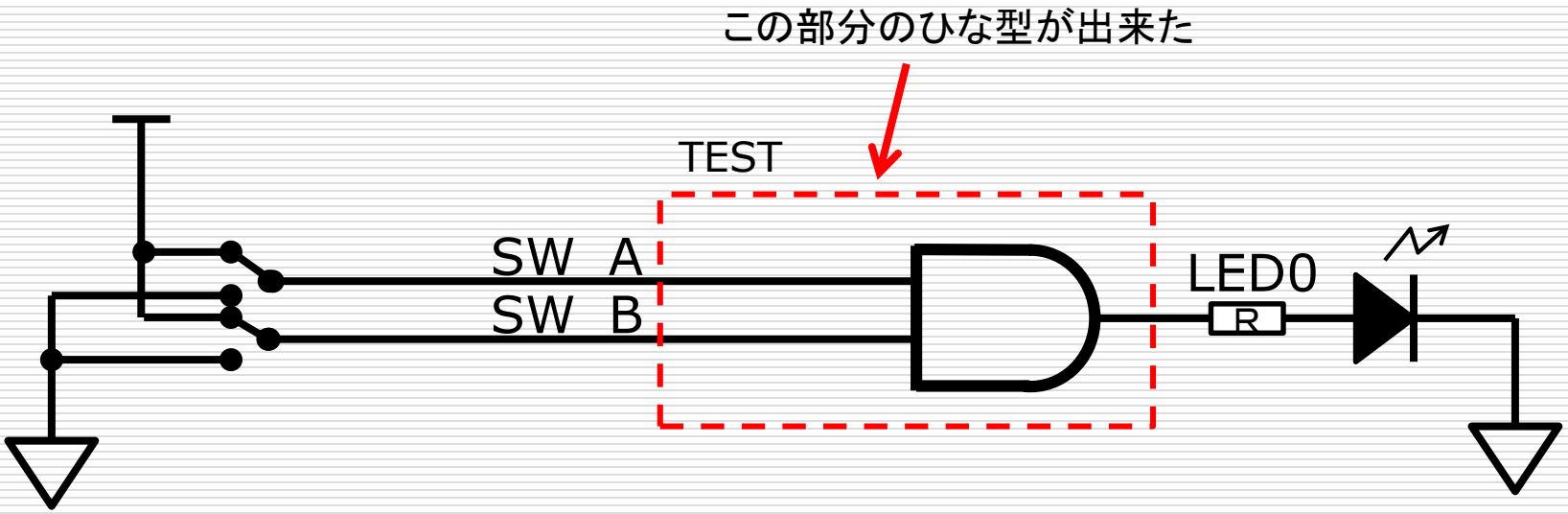


ここのタブで画面を切り替える事が出来ます



TEST.vタブが
選択されていることを確認

AND回路の再確認



ANDを記述して下さい

TEST.vにAND回路を追加

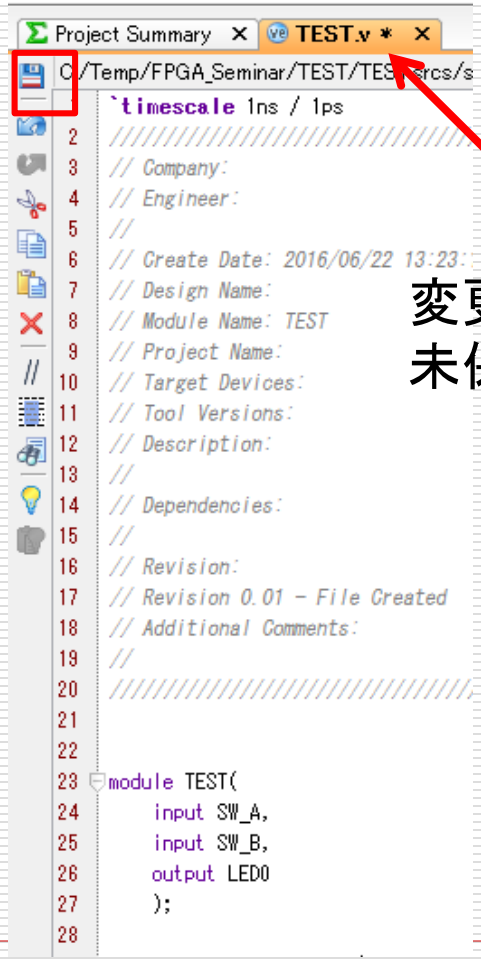
```
22
23  module TEST(
24      input SW_A,
25      input SW_B,
26      output LED0
27      );
28
29      assign LED0 = SW_A & SW_B;
30
31  endmodule
```

outputはデフォルトでwireなので
wire宣言は要らない

書き加えるのはこの1行

組み合わせ回路はassign文で書きます

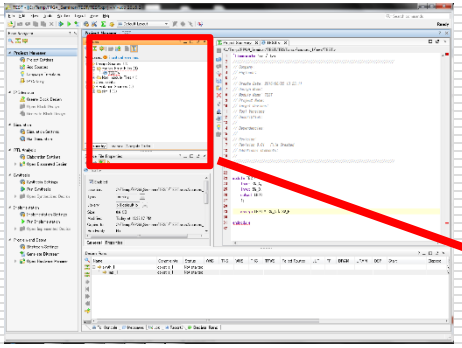
書き終わったらセーブ

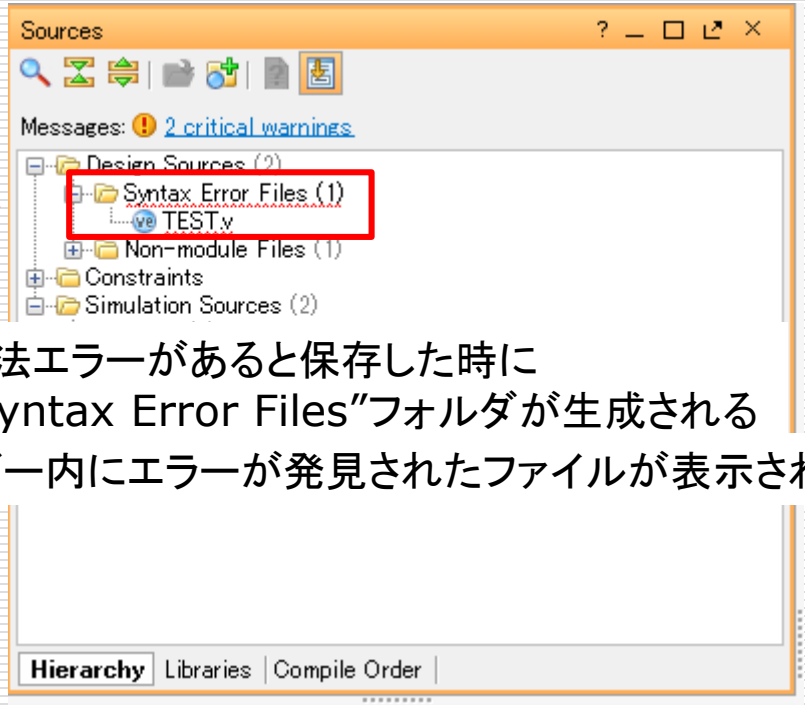


クリック

変更されているが
未保存状態の時＊印が表示される

文法エラー



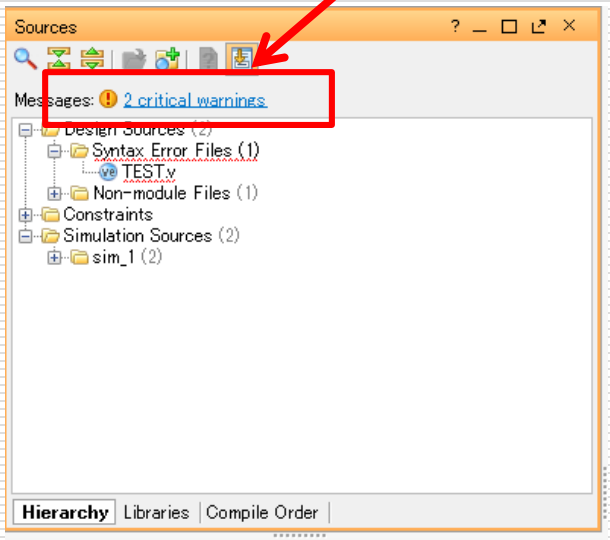


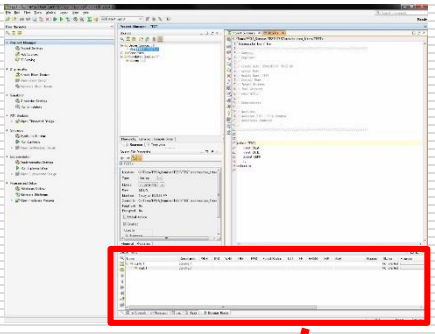
文法エラーがあると保存した時に
”Syntax Error Files”フォルダが生成される
フォルダー内にエラーが発見されたファイルが表示される

補足：セミコロンがないなど、明確にエラーと認識できる時のみこのようになります。
微妙な書き間違いなどはエラーと認識されないことがあるので注意

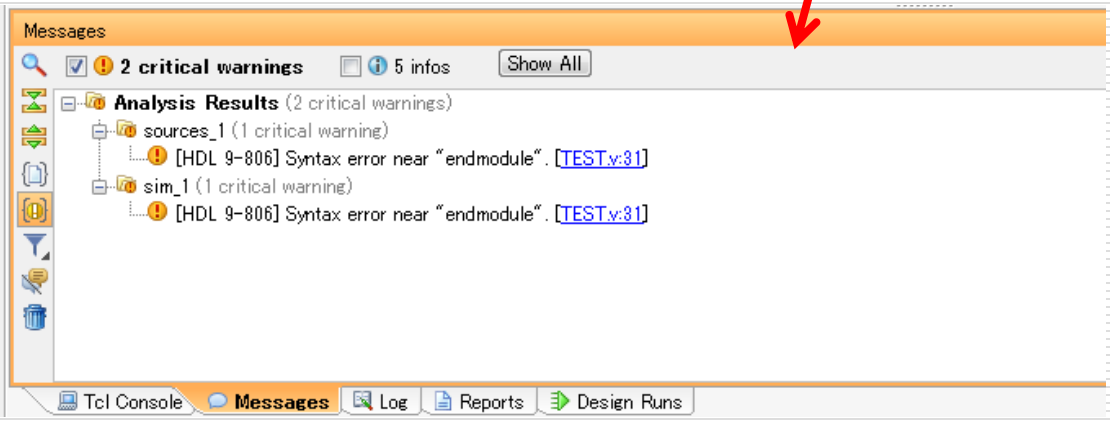
エラー内容の表示

①エラー内容の表示はここをクリック

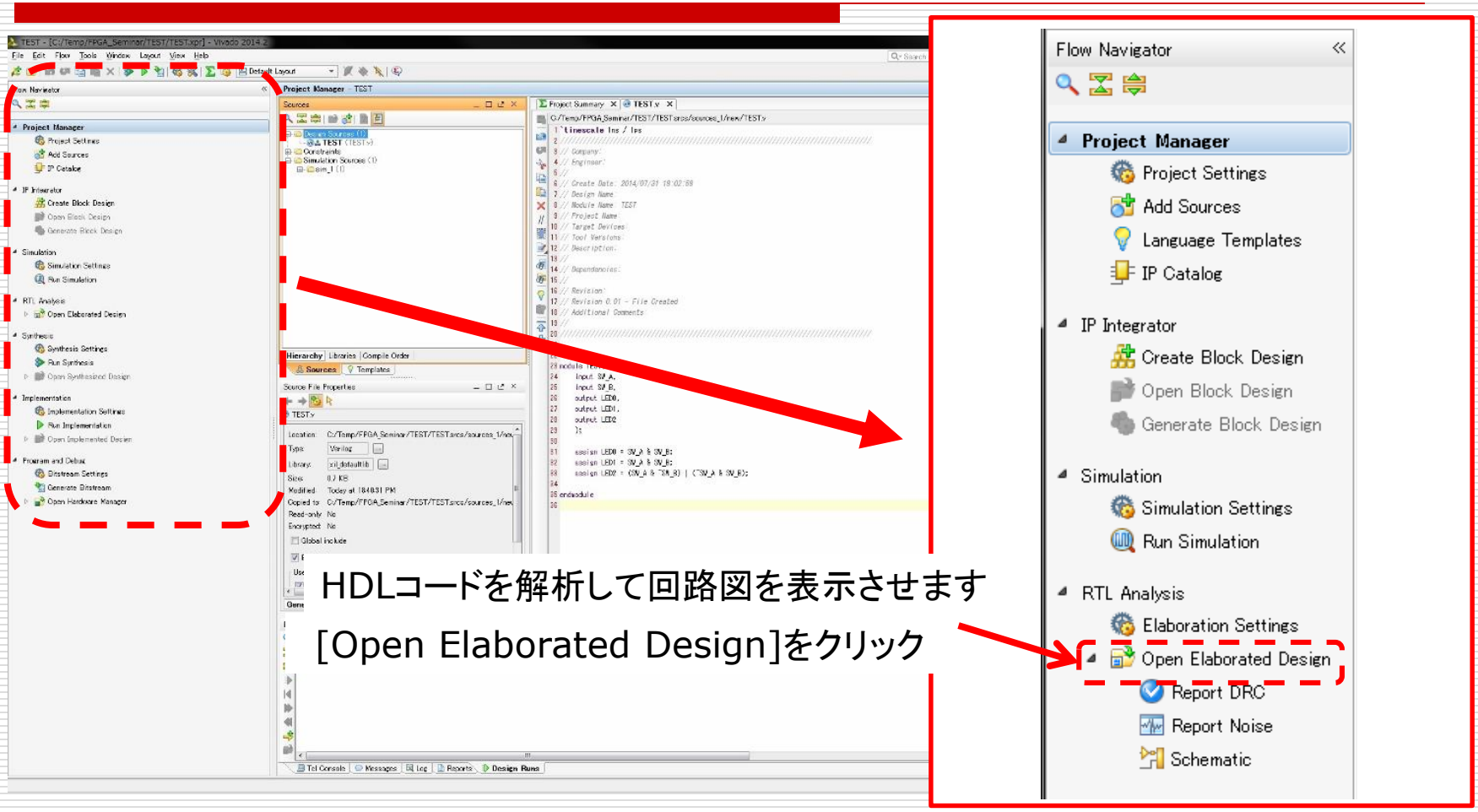




②メッセージ窓に内容が表示される



RTL解析しましょう！

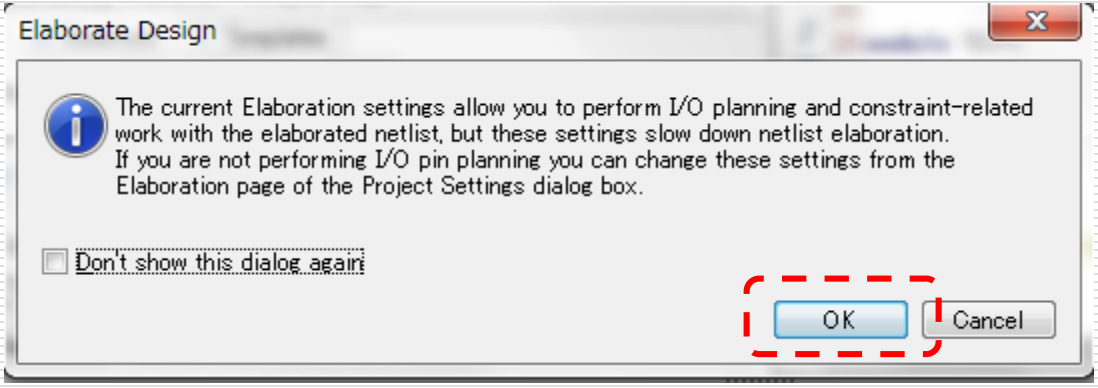


HDLコードを解析して回路図を表示させます

[Open Elaborated Design]をクリック

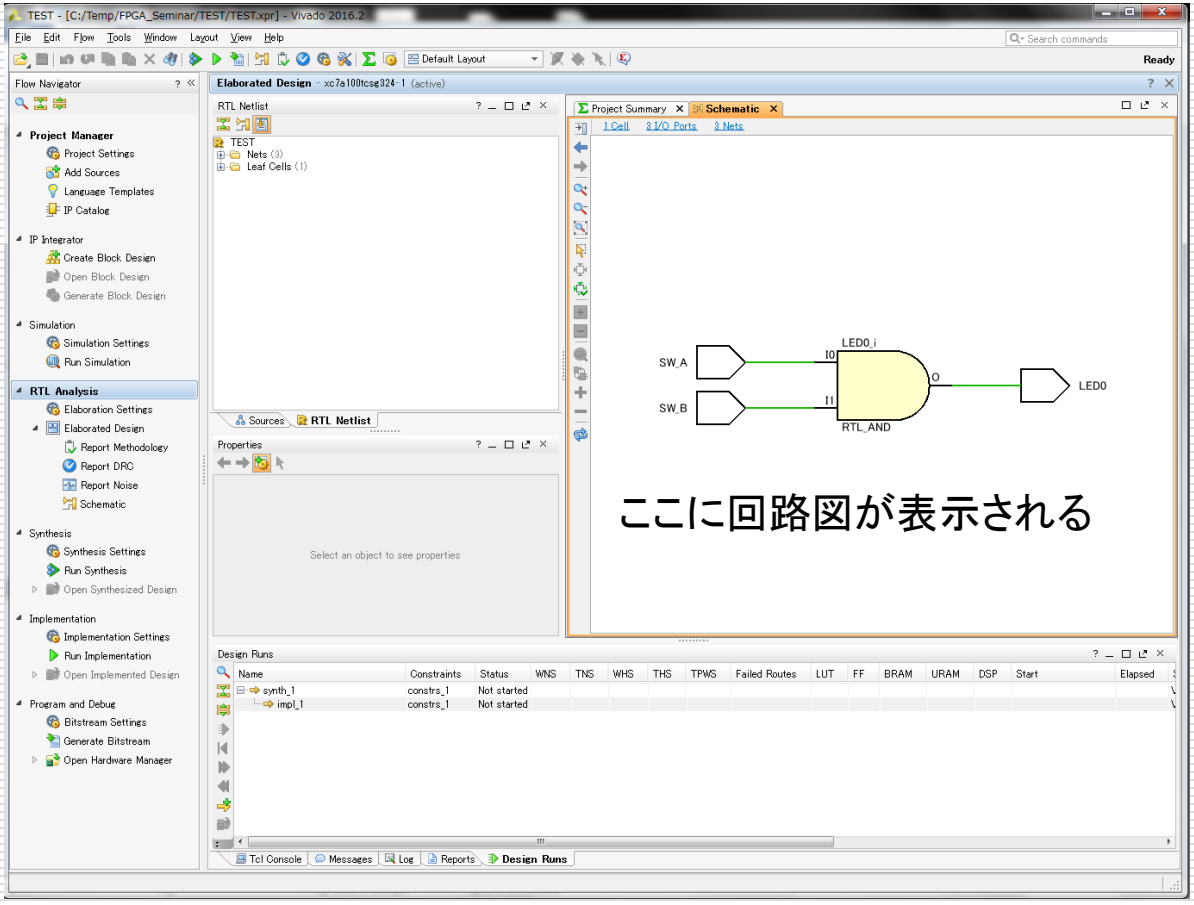
確認画面

この様な確認画面が現れたらOKをクリック

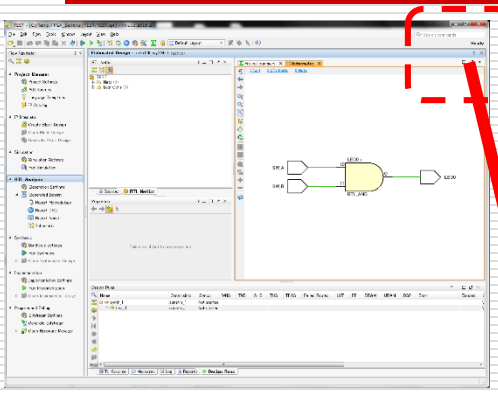


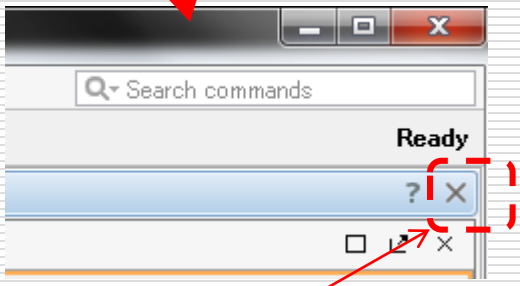
評価方法を変更することで時間短縮できると伝えている。デフォルト設定で良いのでOKをクリック

回路図の確認

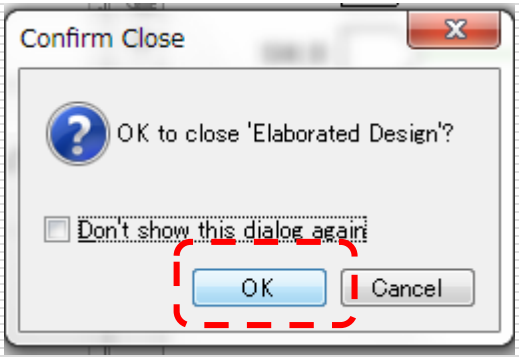

TEST - [C:/Temp/FPGA_Seminar/TEST/TEST.xpr] - Vivado 2016.2
File Edit Flow Tools Window Layout View Help
Default Layout
Ready
Flow Navigator
Project Manager
Project Settings
Add Sources
Language Templates
IP Catalog
IP Integrator
Create Block Design
Open Block Design
Generate Block Design
Simulation
Simulation Settings
Run Simulation
RTL Analysis
Elaboration Settings
Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
Synthesis
Synthesis Settings
Run Synthesis
Open Synthesized Design
Implementation
Implementation Settings
Run Implementation
Open Implemented Design
Program and Debug
Bitstream Settings
Generate Bitstream
Open Hardware Manager
Elaborated Design - xc7a100tcsg324-1 (active)
RTL Netlist
TEST
Nets (3)
Leaf Cells (1)
Sources
RTL Netlist
Properties
Select an object to see properties
Project Summary
Schematic
1 Cell
3 I/O Ports
3 Nets
SW_A
SW_B
I0
I1
LED0_i
O
LED0
RTL_AND
ここに回路図が表示される
Design Runs
Name
Constraints
Status
WNS
TNS
WHS
THS
TPWS
Failed Routes
LUT
FF
BRAM
URAM
DSP
Start
Elapsed
synth_1
constrs_1
Not started
impl_1
constrs_1
Not started
Tcl Console
Messages
Log
Reports
Design Runs

Elaborated Designを閉じる





②確認画面



クリック

①Elaborated Designと書かれている
帯の右端にあるXをクリックして閉じる

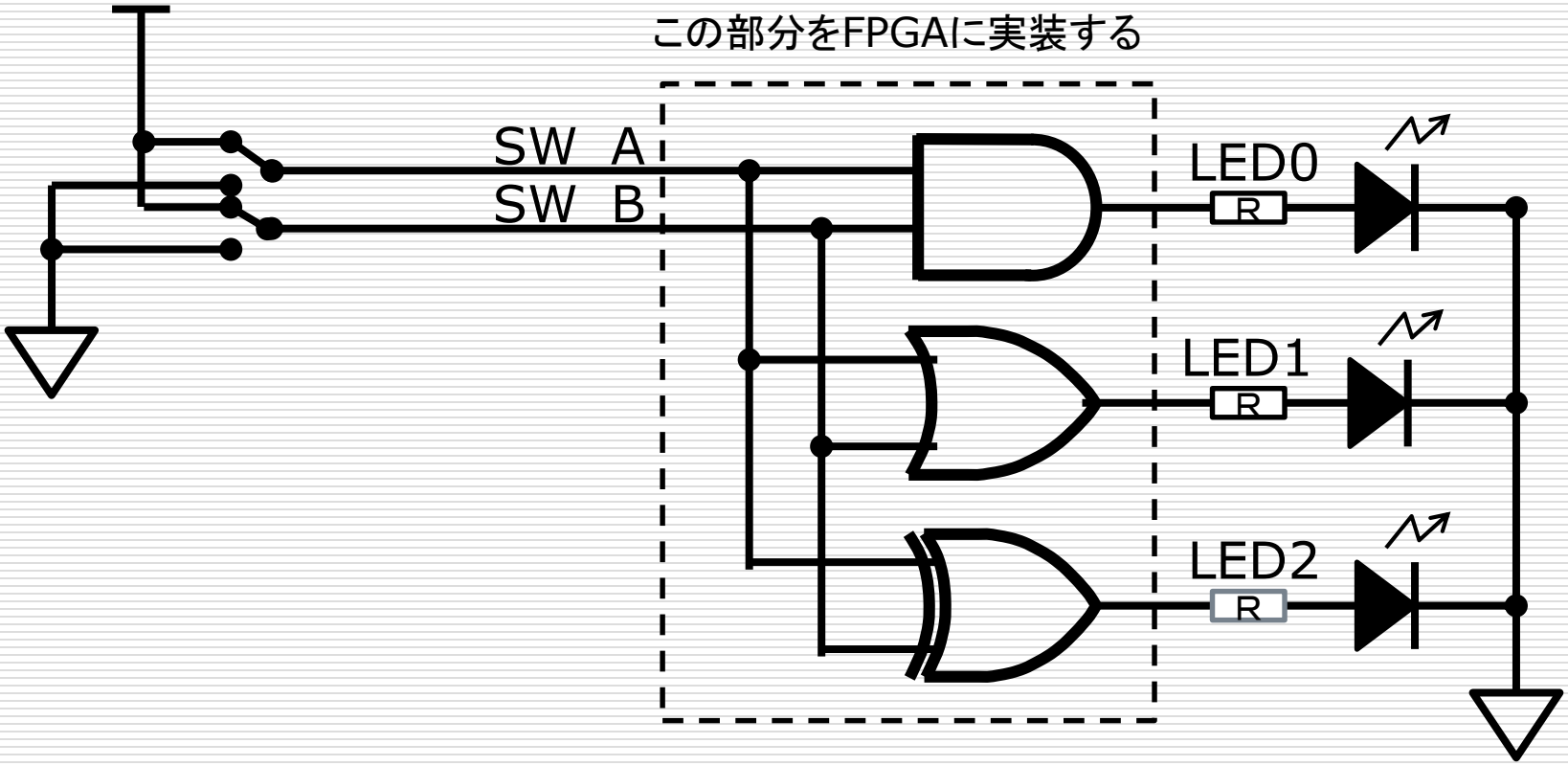
課題

終わった人は下の課題を行って下さい

- ORをTEST.Vに加えてください
- さらにXORをTEST.Vに加えてください

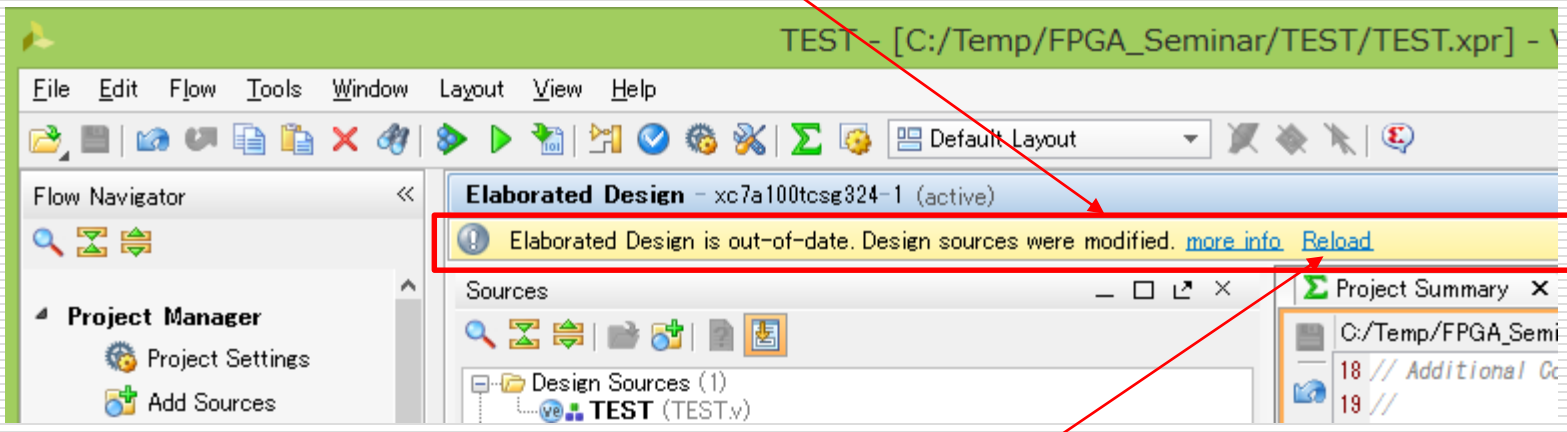
以後課題の説明をします

全回路図



データの再読み込み

コードなどを変更した場合、下のような帯が表示される時があります。
（Elaborated Design画面を閉じずに変更した時など）



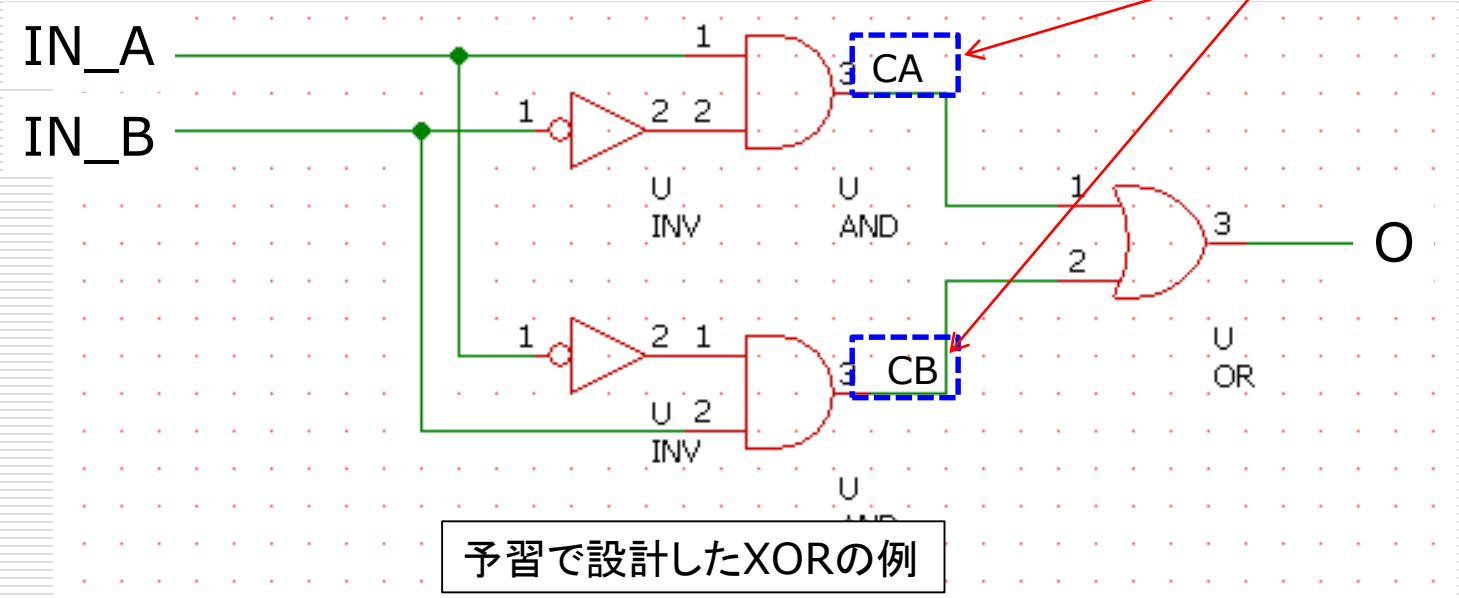
Reloadをクリックしてください

内部信号

モジュール内限定で使用する信号

内部信号は頻繁に使いますので覚えてください

内部信号の定義を加えた
好きな信号名を使ってください



内部信号の使い方

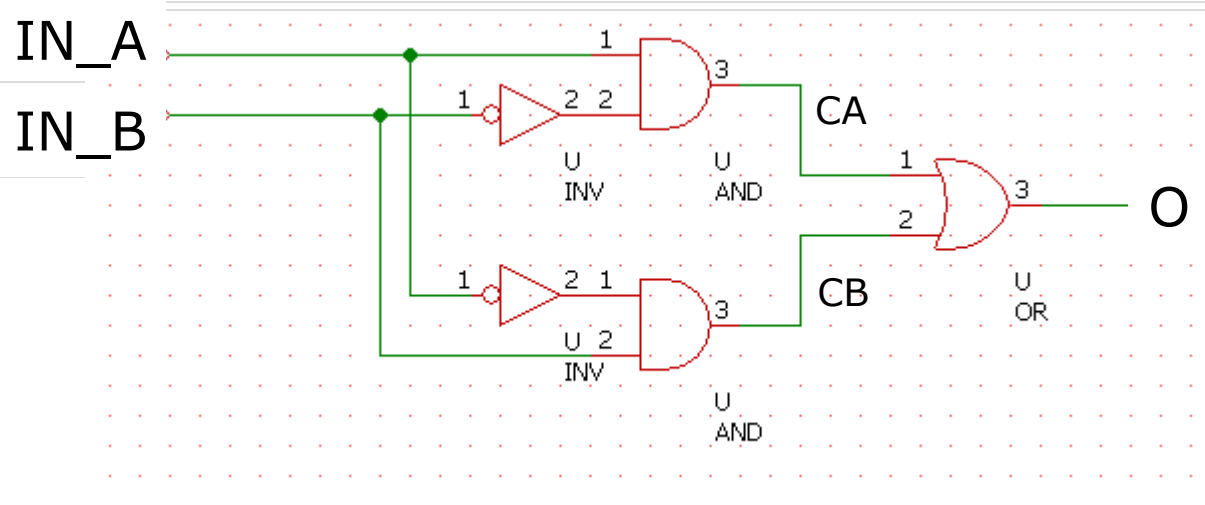
```
module X_OR2(
    input IN_A,
    input IN_B,
    output O
);

  wire CA;
  wire CB;

    assign  CA =  IN_A & ~IN_B;
    assign  CB = ~IN_A &  IN_B;

    assign O = CA | CB;

endmodule
```



内部信号を使うときには
新たに信号を定義する“wire”

“wire”は組み合わせ回路の出力
である事を意味する

組み合わせ回路例　XOR

```
module X_OR2(
    input IN_A,
    input IN_B,
    output O
);

    wire CA;
    wire CB;

    assign  CA =  IN_A & ~IN_B;
    assign  CB = ~IN_A &  IN_B;

    assign O = CA | CB;

endmodule
```

Module名　回路の名前
区別できる好きな名前をつけてよい、ファイル名と同じにした方が良い

()の中に入出力信号線リストを書く

入出力信号線リストの属性定義
input, output, inoutの3種類

回路は
moduleで始まり
endmoduleで終わる

組み合わせ回路の出力はwireで定義
使うassignより上に書く

ポート信号はwireなので宣言は不要

回路図の線の代わりに信号名を使用する

回路はassignを使用して書く、右辺が入力、左辺が出力

これが組み合わせ回路の最低限の知識です

OR, XORを追加したコード

```
module TEST(
    input SW_A,
    input SW_B,
    output LED0,
    output LED1,
    output LED2
    );

    assign LED0 = SW_A & SW_B;
    assign LED1 = SW_A | SW_B;
    assign LED2 = (SW_A & ~SW_B) | (~SW_A & SW_B);

endmodule
```

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、章変更、内田智久(Esys KEK/総研大)
- 2015/11/03 第3.1版 補足説明追加、内田智久(Esys KEK/総研大)
- 2015/12/03 第3.2版 誤字修正、補足説明追加、内田智久(Esys KEK/総研大)
- 2016/1/27 第3.3版 Vivado2015.4対応、内田智久(Esys KEK/総研大)
- 2016/6/22 第3.4版 Vivado2016.2対応、内田智久(Esys KEK/総研大)