



# マイクロプロセッサ実験

## 実験の概要

情報工学科 松谷 宏紀  
[matutani@arc.ics.keio.ac.jp](mailto:matutani@arc.ics.keio.ac.jp)

# IoT(組込み)システムの基礎を学ぶ

- 基盤システム: プロセッサ、周辺回路、ソフトウェア

現実空間



サイバー空間



データ収集



ビッグデータ (大量データ)

データ利活用

IoT (センサ+アクチュエータ)

異常検知、ロボット、自動制御、...

←

社会をより良い方向へ

AI (人工知能)

自動化など新たな価値

この授業ではIoTのシステム面(ソフト、ハード)の基礎を実際に実装しながら学びます。

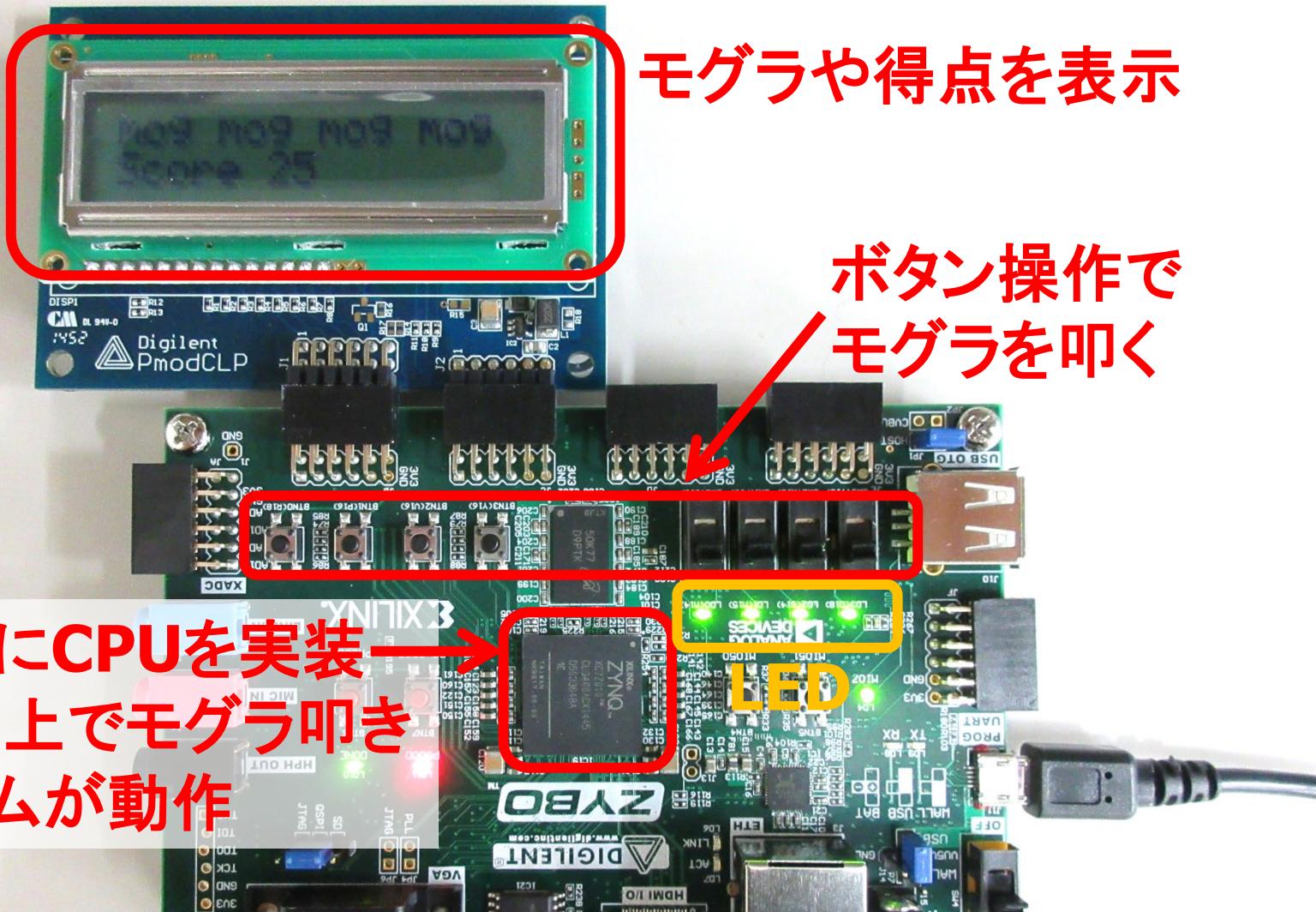
# IoT(組込み)システムの基礎を学ぶ

- 題材: FPGAボードを用いてゲーム機を作る
  - モグラ叩き、テニスゲーム、音ゲー、シューティング
  - LED、液晶モニタ(LCD)、押しボタン、スライドスイッチ
  - 拡張I/O(キーパッド、ブザー、回転スイッチ、LED、...)



- グループワーク(1チーム3~4名)
  - メンバー同士の協力、プロジェクト管理が成功の鍵

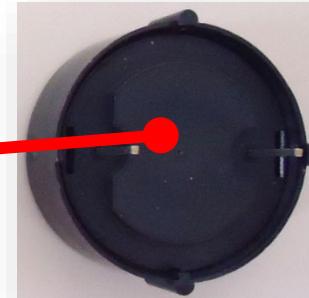
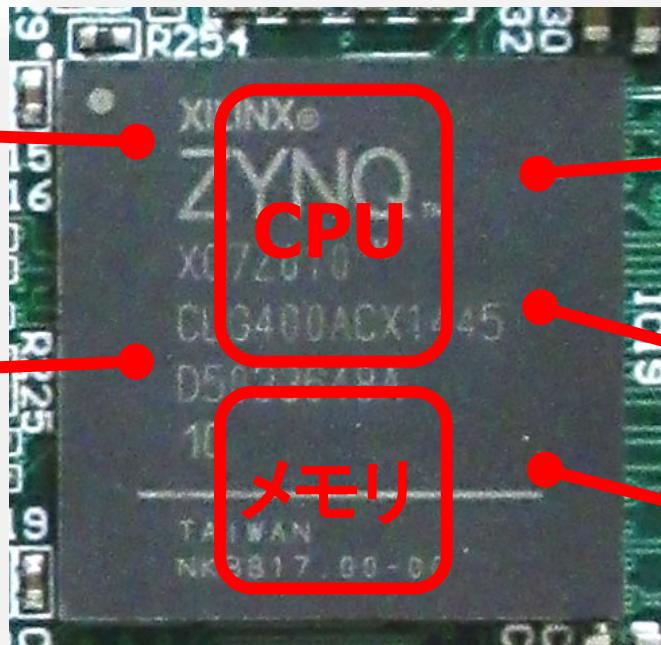
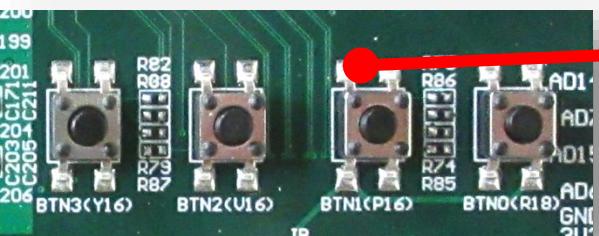
# 実験に使用するFPGAボード



具体的には、このFPGAボードを上記のように使ってゲーム機を実現する(モグラ叩きゲームの例)

# IoT(組込み)ハードウェア設計

- FPGA上にプロセッサ+周辺回路を載せる
  - MIPS互換プロセッサのVerilog HDLモデルは提供
  - プロセッサとFPGAボード上の各種I/Oをつないで、I/Oを利用可能にする

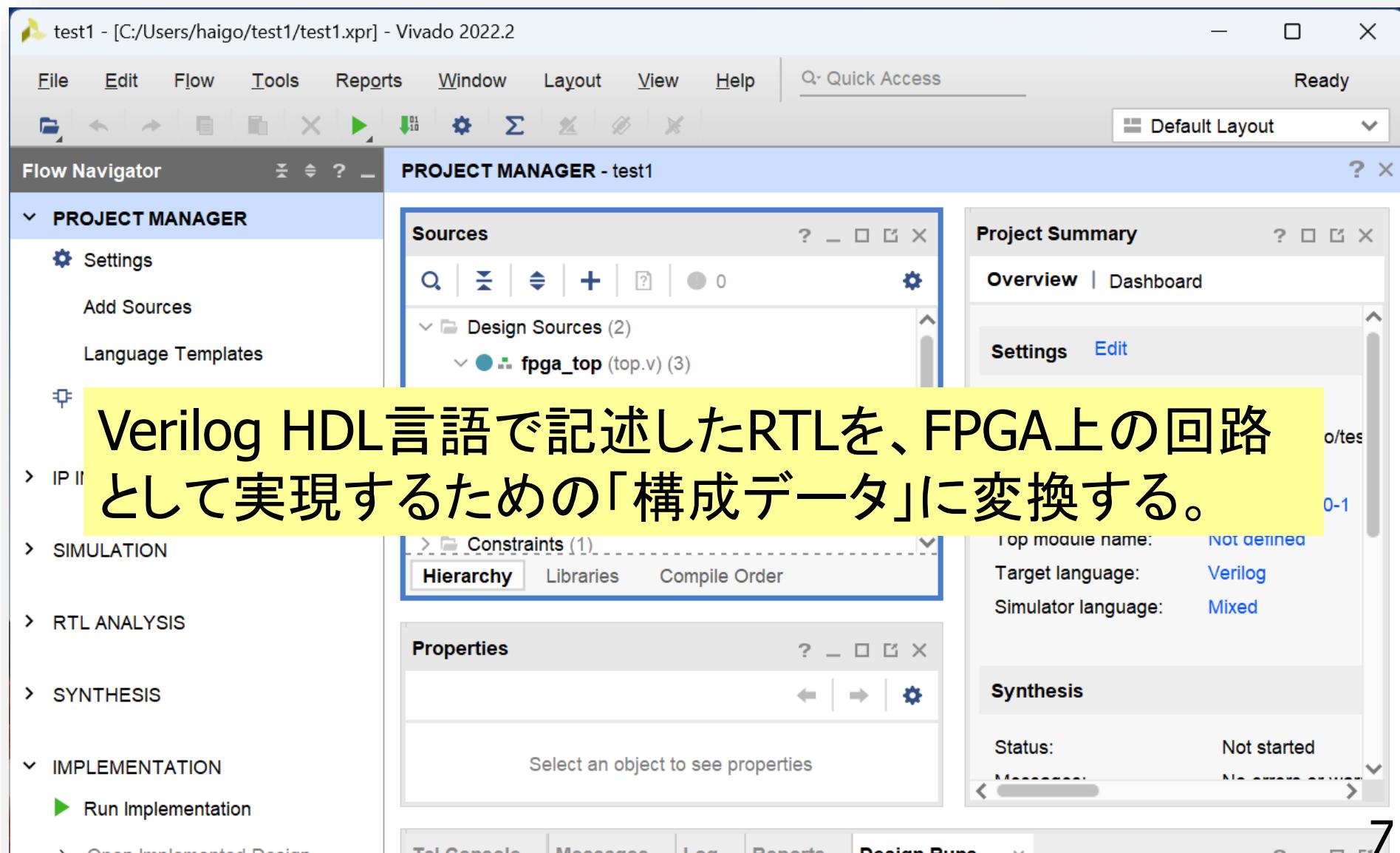


- 習得する内容

- Verilog HDLを用いたハードウェア設計
  - Xilinx Vivadoを用いた論理合成、配置配線

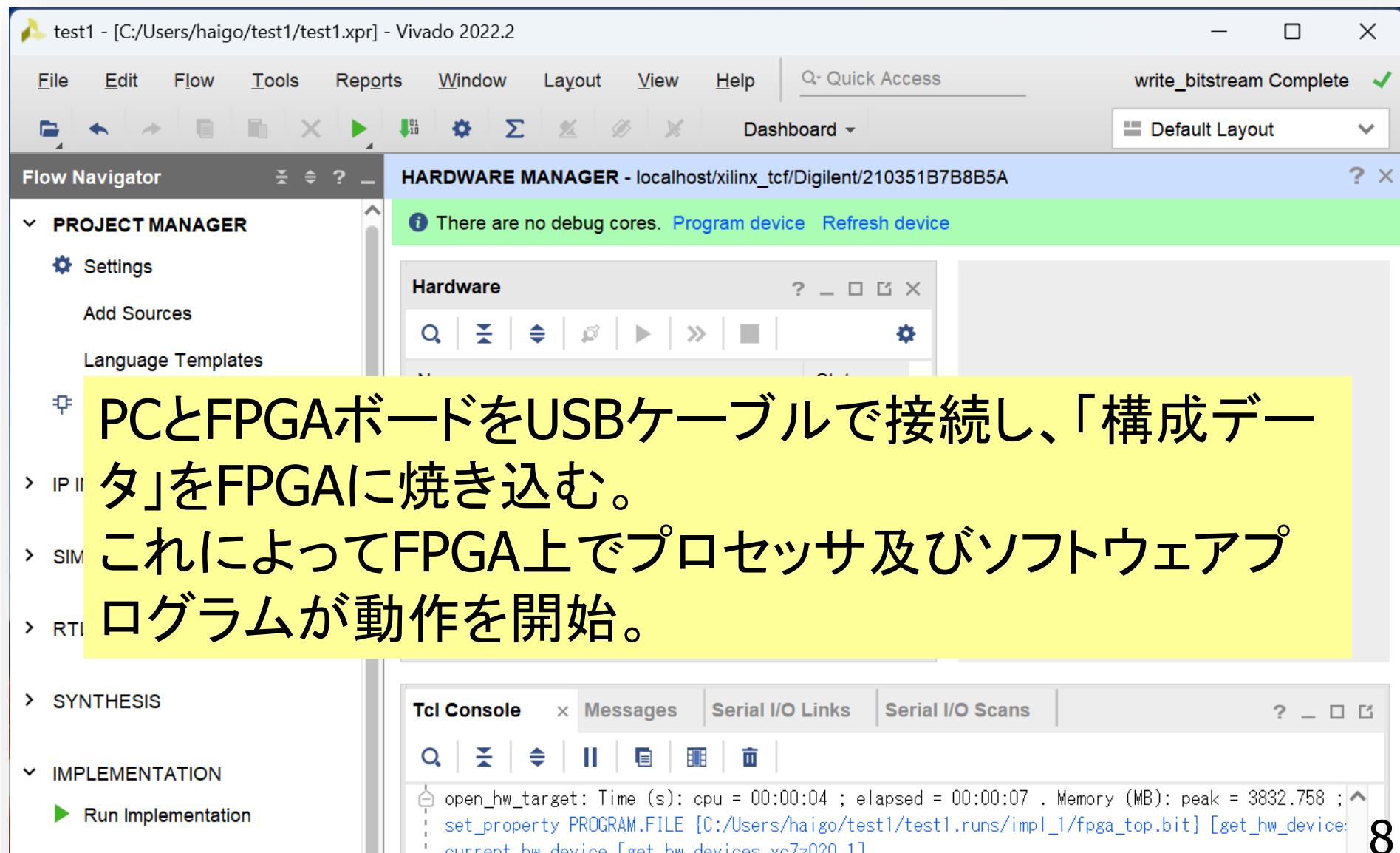
# ハードウェア設計に使用するツール

- 論理合成、配置配線(Xilinx Vivado)



# ハードウェア設計に使用するツール

- FPGAへの焼き込み(Xilinx Vivado)



# IoT(組込み)ソフトウェア設計

- ・ プロセッサ上で動作するプログラムを開発
  - モグラ叩き、テニスゲーム、音ゲー、シューティング
  - FPGAボード上の各種I/O(LED、LCD、スイッチ)制御

/\* 押されたスイッチの値をLEDに表示 \*/

```
void main() {
    int *sw_ptr = (int*)0xff04;
    int *led_ptr = (int*)0xff08;
    for (;;) *led_ptr = *sw_ptr;
```



- ・ 習得する内容
  - C言語による組込みソフトウェア開発
  - 標準的なクロス開発環境(gcc, as, ld, objdump)を用いた組込みソフトウェア開発フロー

# IoT(組込み)ソフトウェア設計

- ・ プロセッサ上で動作するプログラムを開発
  - モグラ叩き、テニスゲーム、音ゲー、シューティング
  - FPGAボード上の各種I/O(LED、LCD、スイッチ)制御



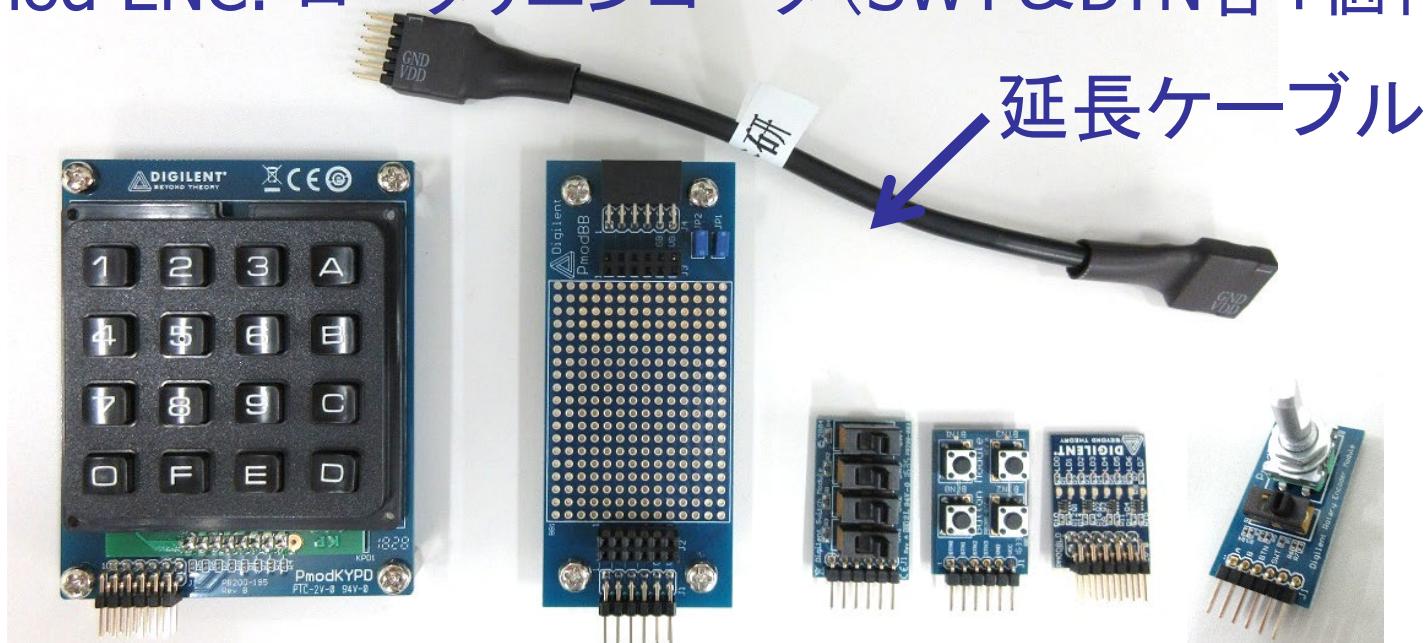
/\* 液晶モニタに文字を表示する \*/

```
lcd_data(0x6d);      /* m */  
lcd_data(0x6f);      /* o */  
lcd_data(0x67);      /* g */  
lcd_data(0x20);      /*   */  
lcd_data(0x6d);      /* m */  
lcd_data(0x6f);      /* o */
```

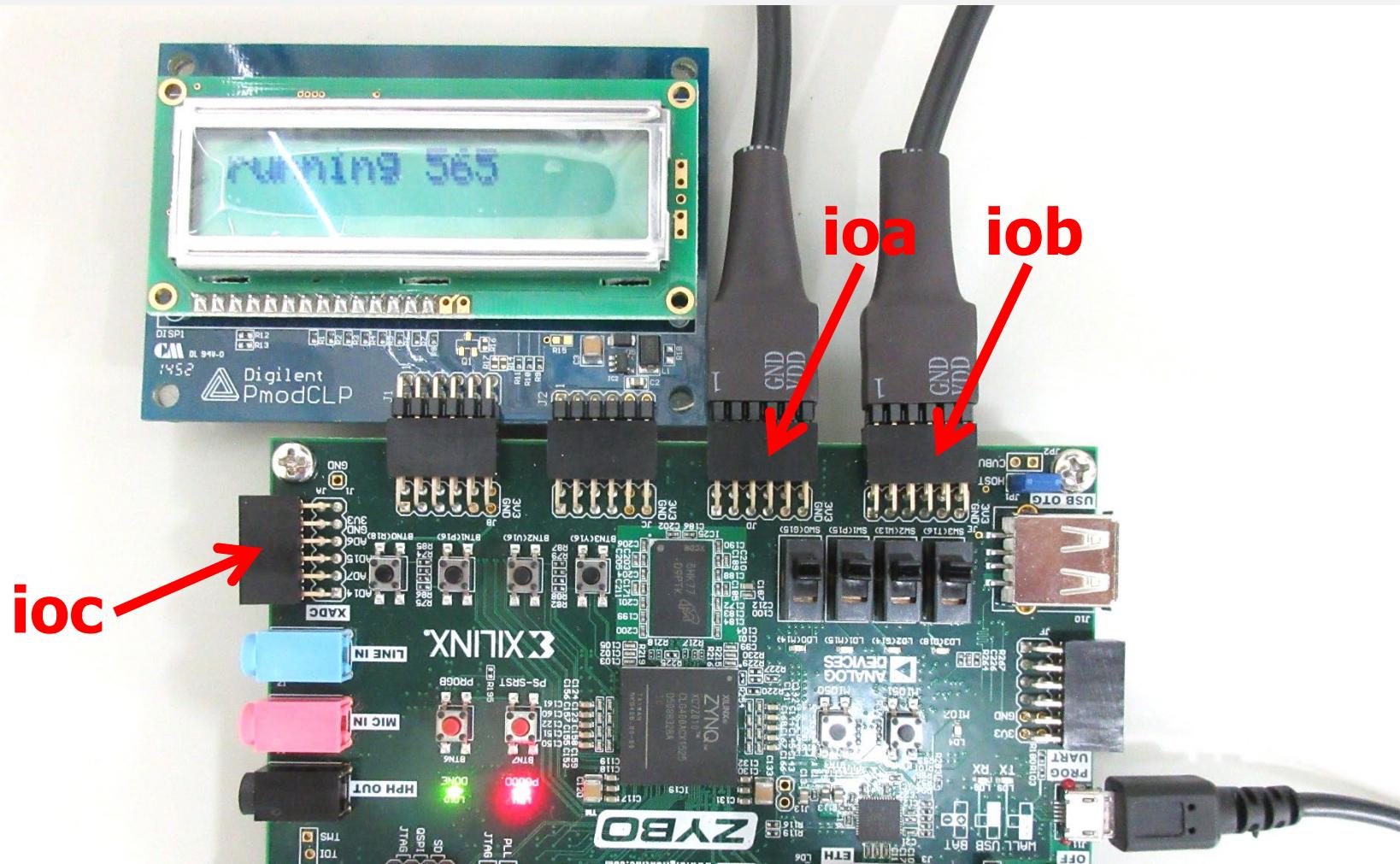
- ・ 習得する内容
  - C言語による組込みソフトウェア開発
  - 標準的なクロス開発環境(gcc、as、ld、objdump)を用いた組込みソフトウェア開発フロー

# 拡張I/Oボード

- ・ ゲーム内容に応じて自由にFPGAボードを拡張
  - Pmod KYPD: 16ボタンのキーパッド
  - Pmod LED: 4個の追加LED ※8個品も混じってる？
  - Pmod SWT: 4個の追加スライドスイッチ
  - Pmod BTN: 4個の追加押しボタン
  - Pmod BB: ブレッドボード(ブザーを実装)
  - Pmod ENC: ロータリエンコーダ(SWT & BTN各1個付き)



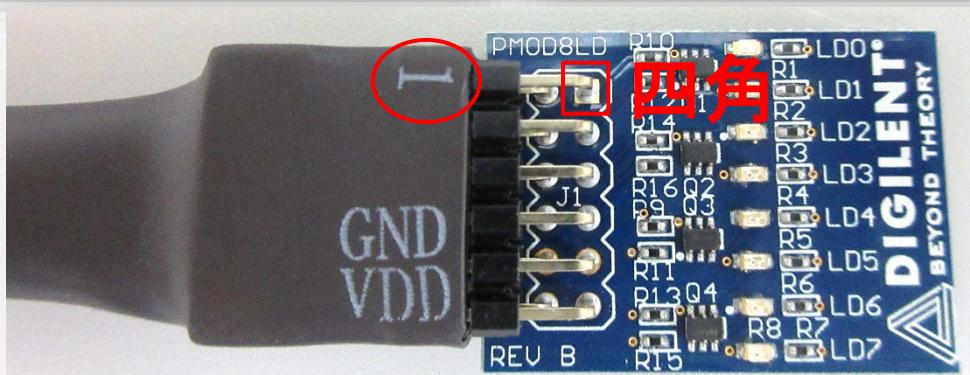
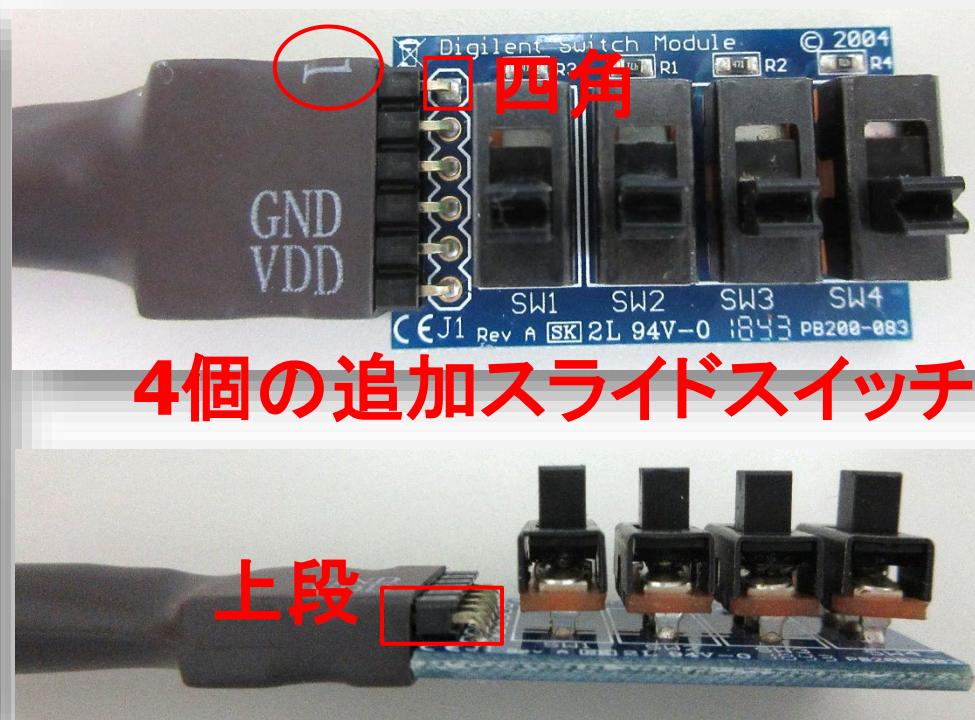
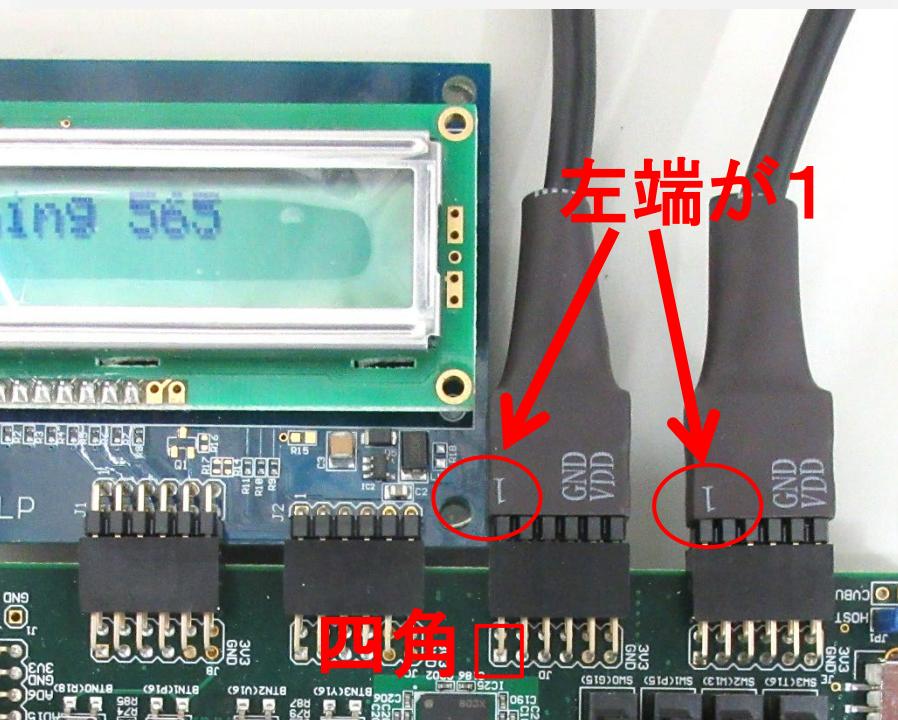
# 拡張I/Oボードの使い方



拡張ボードは、ポートioa、job、iocに挿入して使う。  
それぞれ8-bit幅。例えば、ここに0x0fを書き込むと、  
下位4-bitがすべてオンになる(0000\_1111)。

# 拡張I/Oボードの注意点

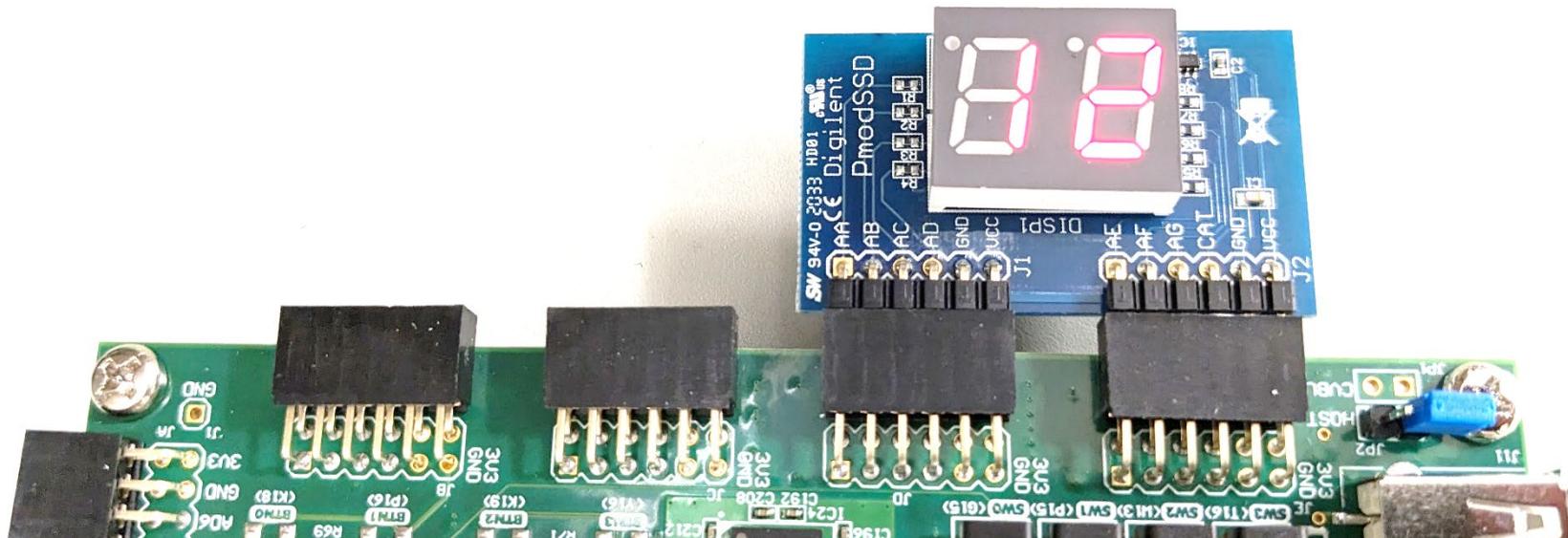
- 延長ケーブルの極性は要注意



# 拡張I/Oボード(オマケ)

- ・ ゲーム内容に応じて自由にFPGAボードを拡張
  - 4行LCD: 20文字4行の大画面LCD ※各班1個ずつ
  - Pmod SSD: 7セグメントLED ※各班1～2個ずつ
  - ...

4行LCDは電気系実験室謹製  
7セグLEDは今年試験導入



# マイクロプロセッサ実験の特徴

- 実験の趣旨
  - 実際に動作するIoT(組込み)システムを開発
  - ソフト開発、ハード開発、プロジェクト管理の基礎を学ぶ
- プロジェクト型の実験
  - 成績は、製品の出来によって決まる
  - メンバー同士の協力、作業分担が成功の鍵
- 実験の流れ

第1週	チュートリアル、グループ(3~4人)で構想
第2週	ディスカッション 単機能でいいので、実機上で何とか動かす
第3週	ゲームとして作り込み、みんなの前で発表

# タイムテーブル(1週目)

10:45–11:30	本実験の概要、出席確認、班分け
11:30–12:15	チュートリアル1
12:15–13:00	全員で昼休み
13:00–13:45	チュートリアル2
13:45–15:45	班ごとに演習課題(ハード、ソフト、LCD)をやる サンプルゲームの動作確認
15:45–16:15	ゲームの構想を練る
16:15–	<b>サンプルゲームが動作し、ゲームの内容が決 まった班は解散</b>

※休憩は各自、自由に取る

# タイムテーブル(2週目)

10:45－11:30	出席確認 外部仕様(ゲームの取説)、内部仕様(フローチャート)を作成
11:30－12:15	FPGA実機ボードを用いたゲーム開発を開始 <b>教員とディスカッション(外部仕様・内部仕様を提出し、企画を口頭でプレゼン)</b>
12:15－13:00	班ごとに昼休み
13:00－16:15	FPGA実機ボードを用いたゲーム開発を継続 <b>LCDを使ったアニメーション、スイッチを使った当たり判定ができた班は教員に見せてから解散</b>
備考	ここで頑張って3週目の作業量を減らすのが良策

※休憩は各自、自由に取る

# タイムテーブル(3週目)

10:45－12:15	出席確認 FPGA実機ボードを用いたゲーム開発を継続 (必要に応じて)拡張ボードの追加実装
12:15－13:00	班ごとに昼休み
13:00－16:15	FPGA実機ボードを用いたゲーム開発の仕上げ
16:15－17:00	<b>みんなの前でデモ発表</b>
備考	レポート作成方法は、実験手順書の「最終レポートに関する注意」を <b>熟読</b> してください また、アンケートに答えるのを忘れずに

※休憩は各自、自由に取る

# 過去の作品(動画)集

<https://www.arc.ics.keio.ac.jp>

The screenshot shows a web browser window with the URL <https://www.arc.ics.keio.ac.jp/lecture>. The page features a large image of a SLAM (Simultaneous Localization And Mapping) algorithm's internal state, showing a map of a campus with various colored particles representing sensor data and estimated positions. Overlaid on the map are the words "MATSUTANI LAB.", "DEPT OF ICS, KEIO UNIV", and "SLAM AT YAGAMI CAMPUS" in green. Below the image is a dark navigation bar with white text and icons. The "講義" (Lecture) tab is highlighted. Other tabs include ホーム (Home), メンバ (Members), 研究 (Research), 業績 (Achievements), アクセス (Access), B 3 向け (For B3), 内部向け (Internal), English, and a search bar.

講義

情報工学科(学部)向け

- アルゴリズム第2
- 情報工学実験第2 マイクロプロセッサ
- VLSI設計演習

機械工学科、管理工学科、数理科学科、物理学科(学部)向け

- 計算機基礎

ユーザ名「jikken」  
パスワード「2023f」



# マイクロプロセッサ実験

Xilinx Vivadoの使い方

情報工学科 松谷 宏紀

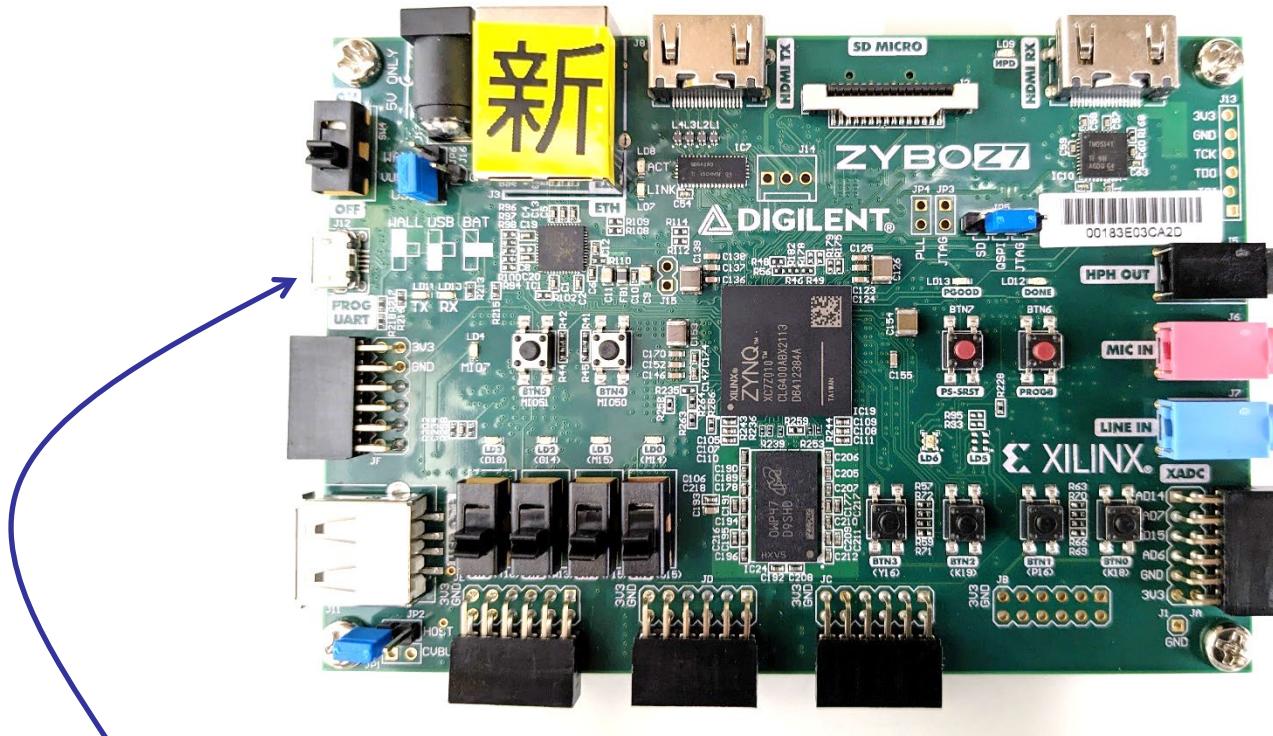
[matutani@arc.ics.keio.ac.jp](mailto:matutani@arc.ics.keio.ac.jp)

# 参考：自宅で作業する場合

- Linuxサーバ(remote.educ.cc.keio.ac.jp)を利用  
詳細は下記参照(塾内ネットワーク or 要keio.jp認証)  
[https://secure.itc.keio.ac.jp/c/a/st/ja/com\\_remote\\_gui\\_st.html](https://secure.itc.keio.ac.jp/c/a/st/ja/com_remote_gui_st.html)
- SSH、WinSCPによる接続ができるか？
- RDPによる接続ができるか？
  - Windowsなら、「Windowsアクセサリ」→「リモートデスクトップ接続」
  - 接続先は「remote.educ.cc.keio.ac.jp:22385」

# 新FPGAボードについて

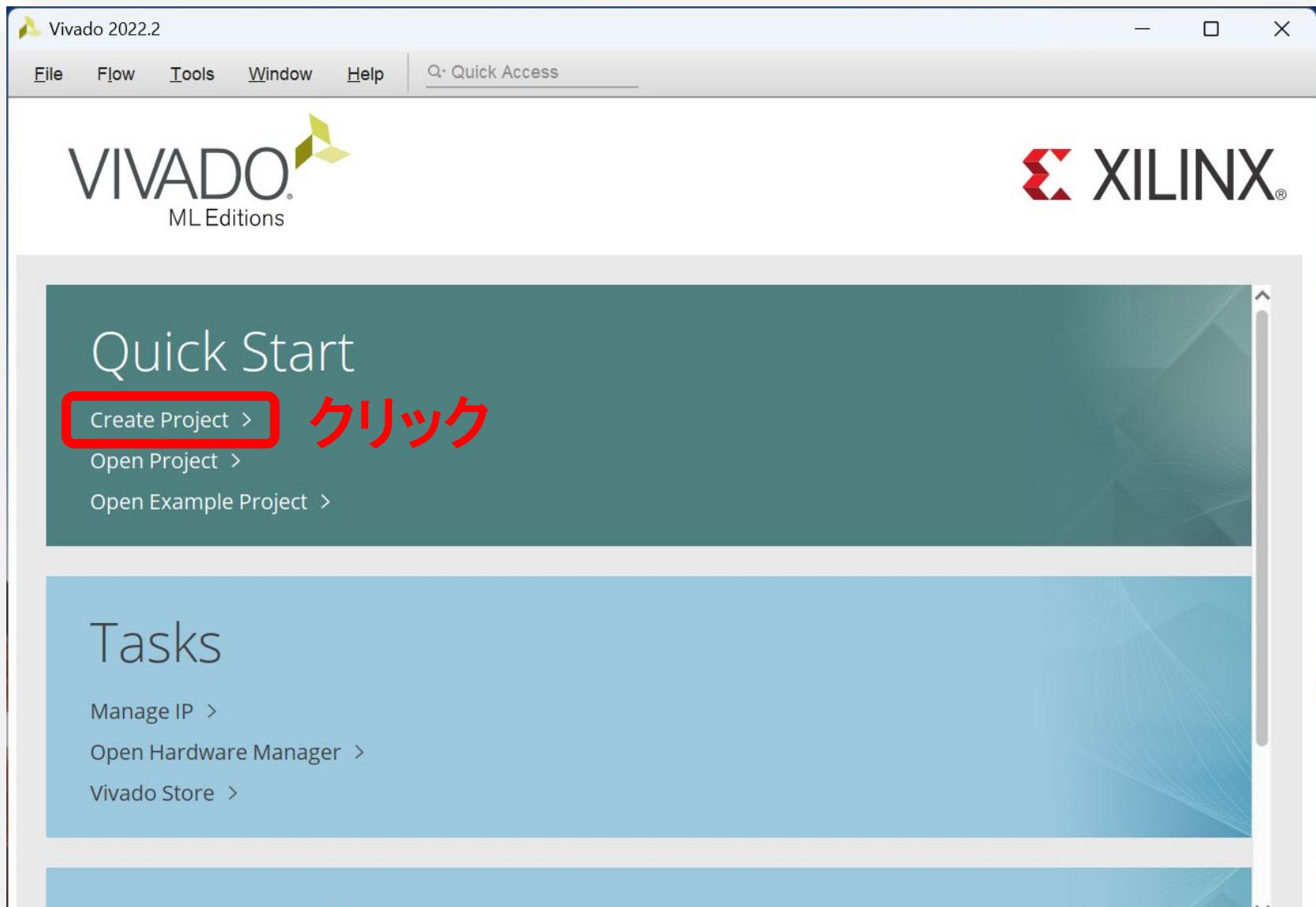
- ・ 黄色の「新」シールが貼ってあるFPGAボード
  - 数は少ないものの、設計フローが僅かに違うため注意



以前と同じFPGAボードを調達できなかったため、2種類のボードが混在することになってしまいました。USBコネクタにケーブルをさした状態でケーブルを持ち上げると、USBコネクタが破損し、ボードが使えなくなるので大事に扱ってください！

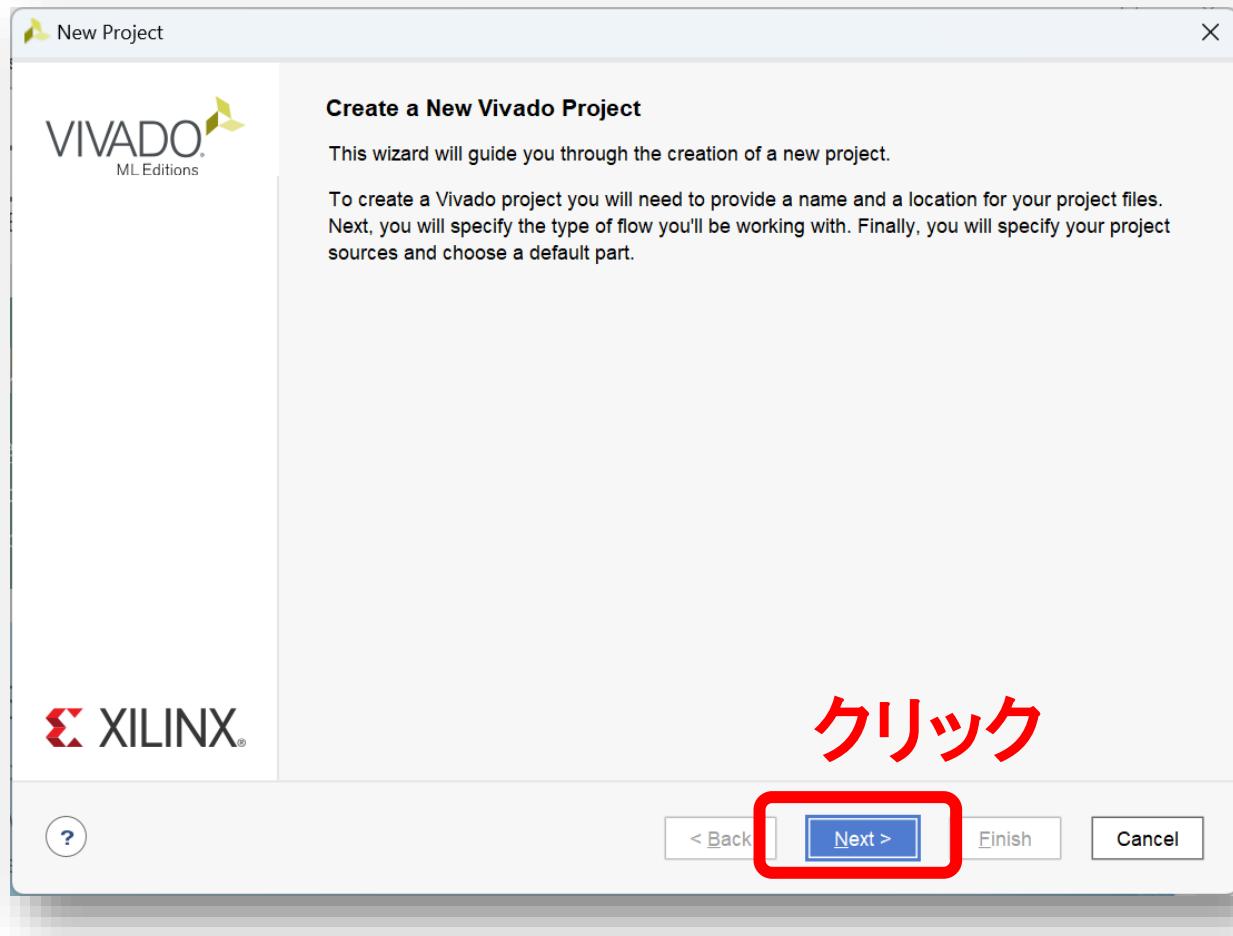
# プロジェクトの新規作成(1/14)

- 「Create Project」→各種質問に答える



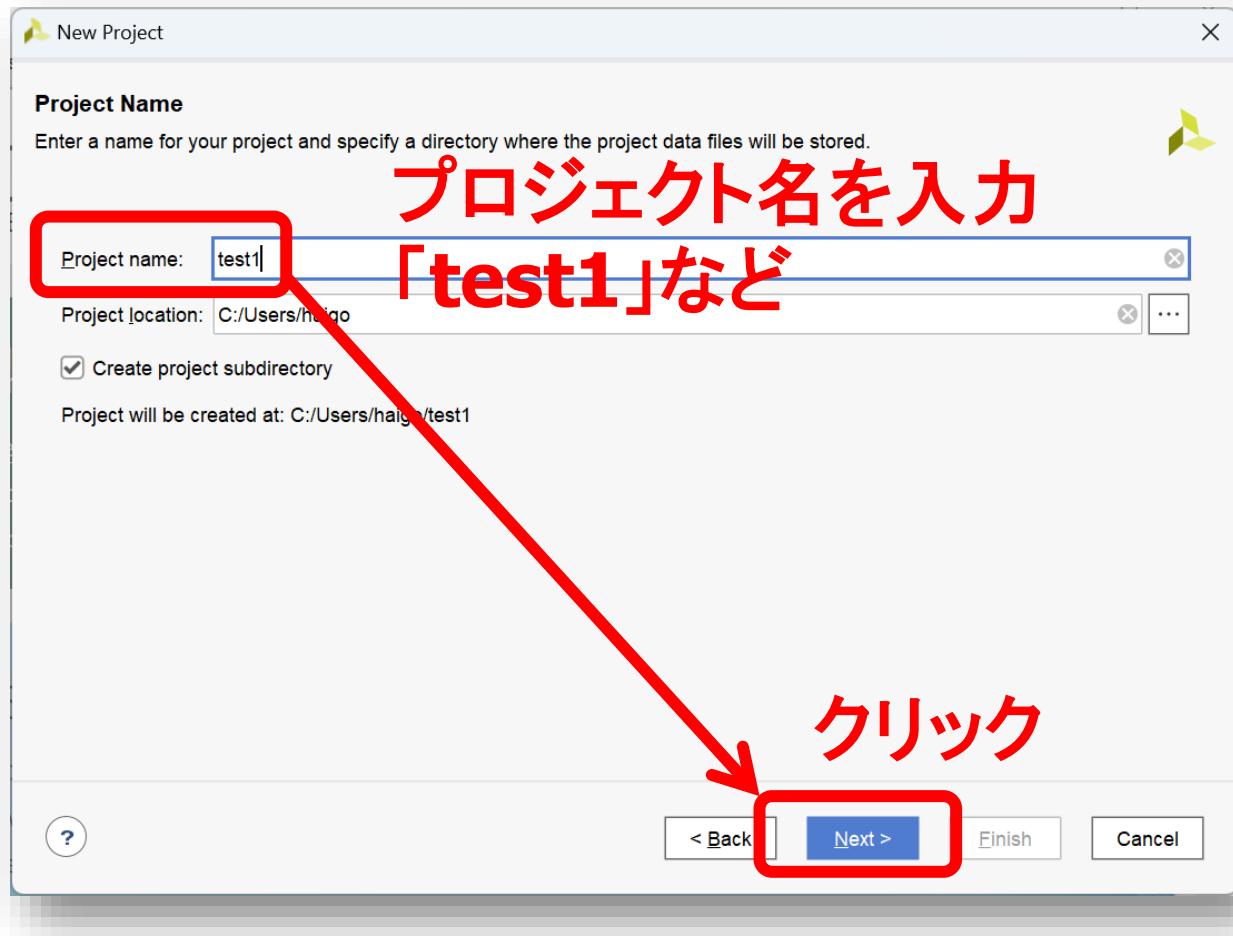
# プロジェクトの新規作成(2/14)

- 「Create Project」→各種質問に答える



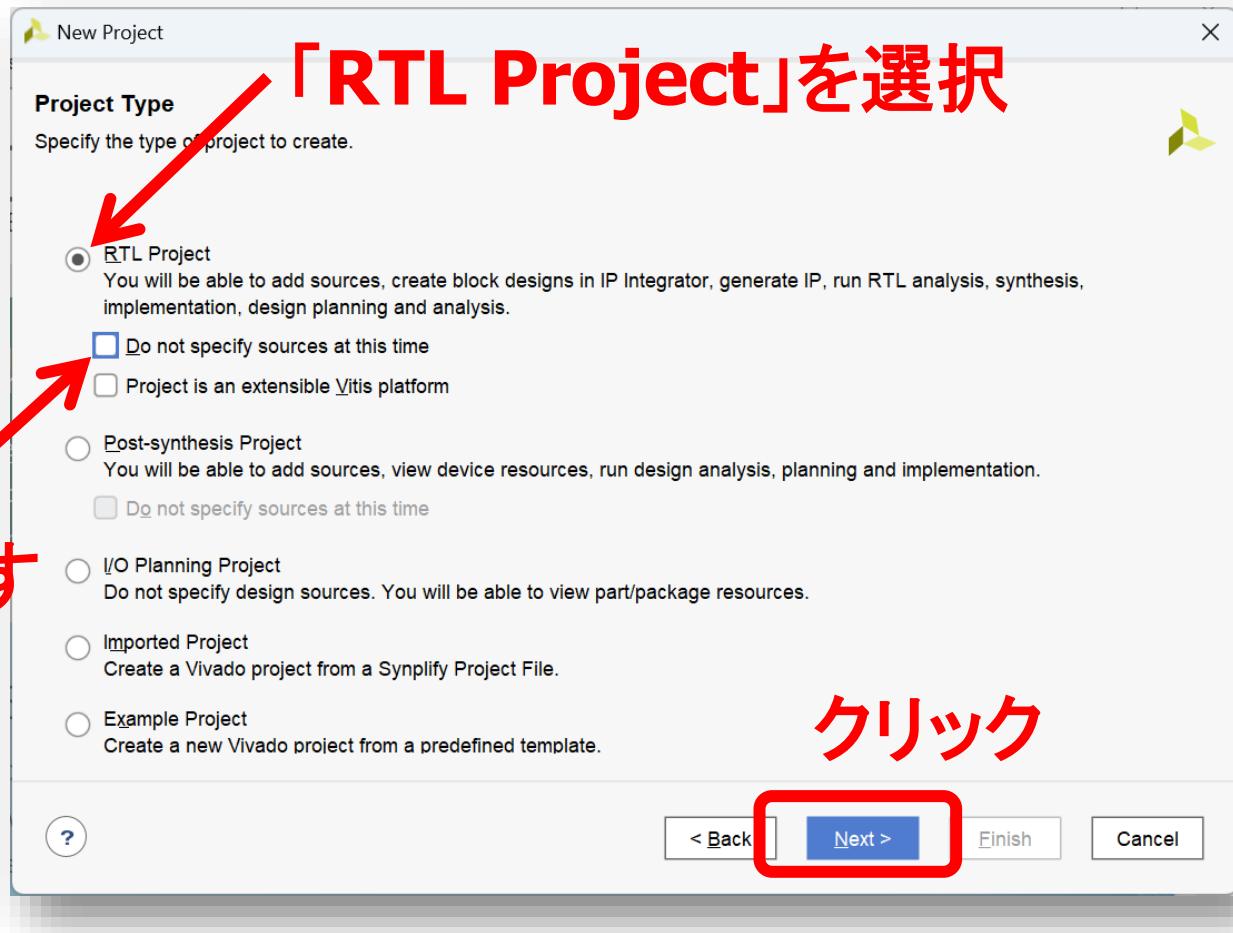
# プロジェクトの新規作成(3/14)

- 「Create Project」→各種質問に答える



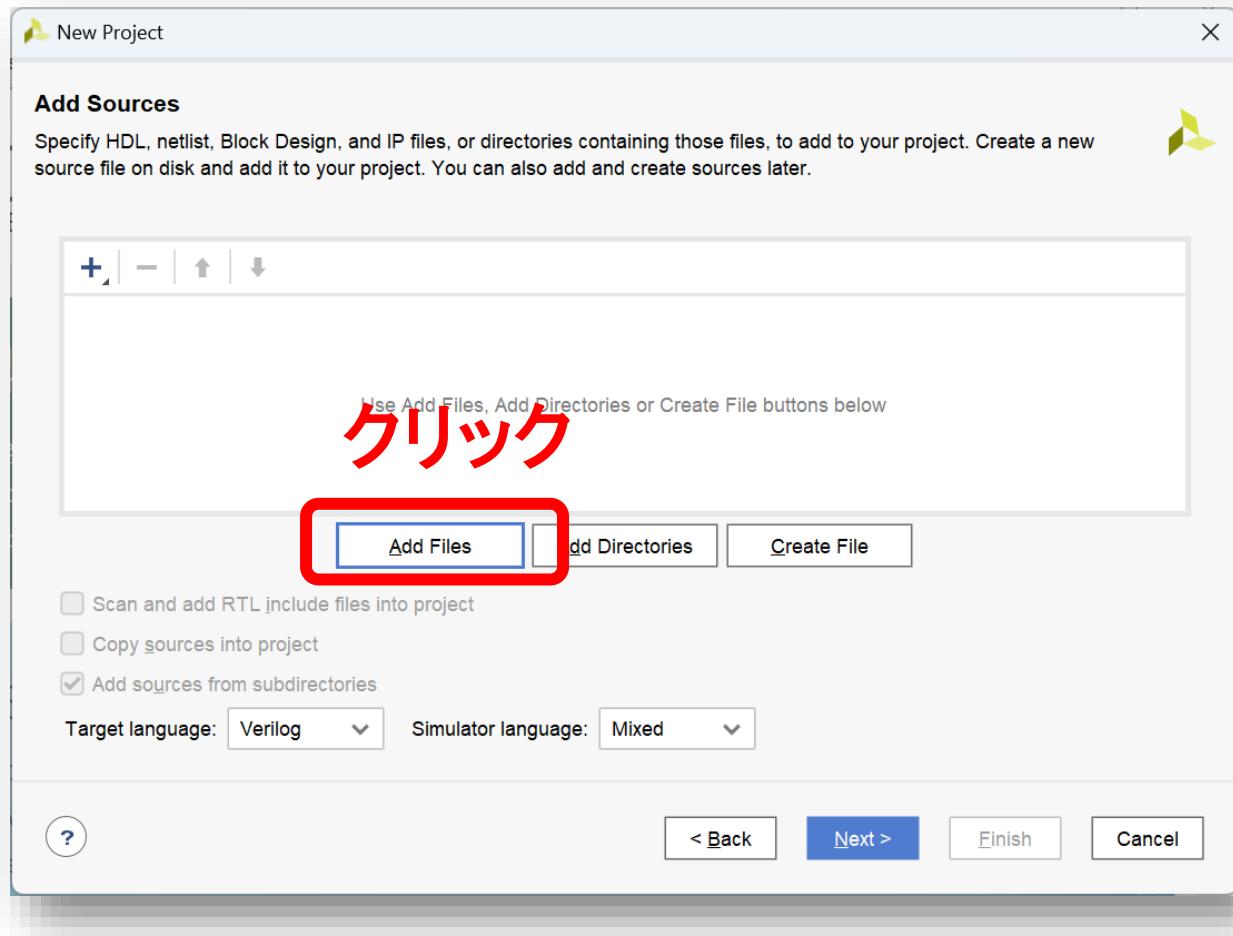
# プロジェクトの新規作成(4/14)

- 「Create Project」→各種質問に答える



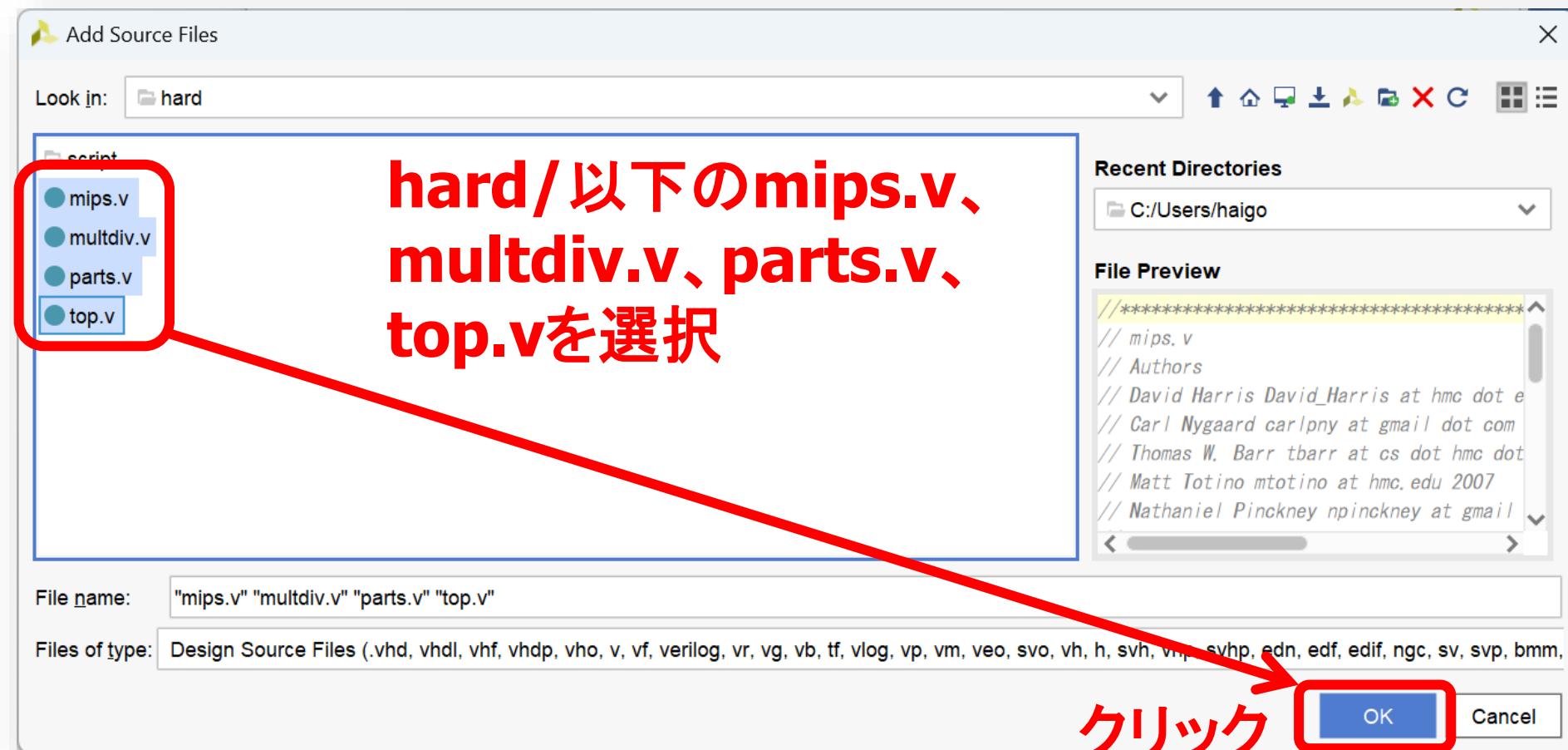
# プロジェクトの新規作成(5/14)

- 「Create Project」→各種質問に答える



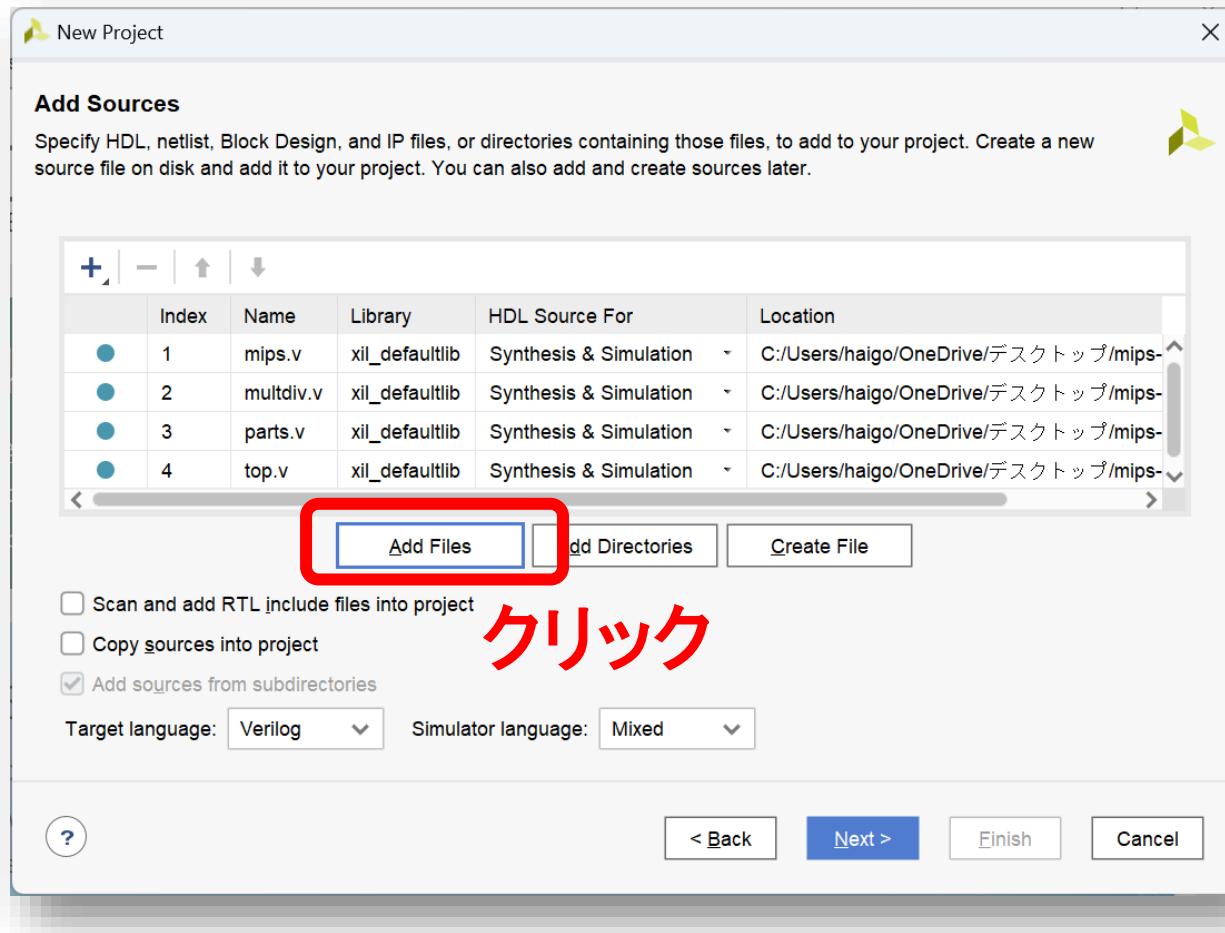
# プロジェクトの新規作成(6/14)

- 「Create Project」→各種質問に答える



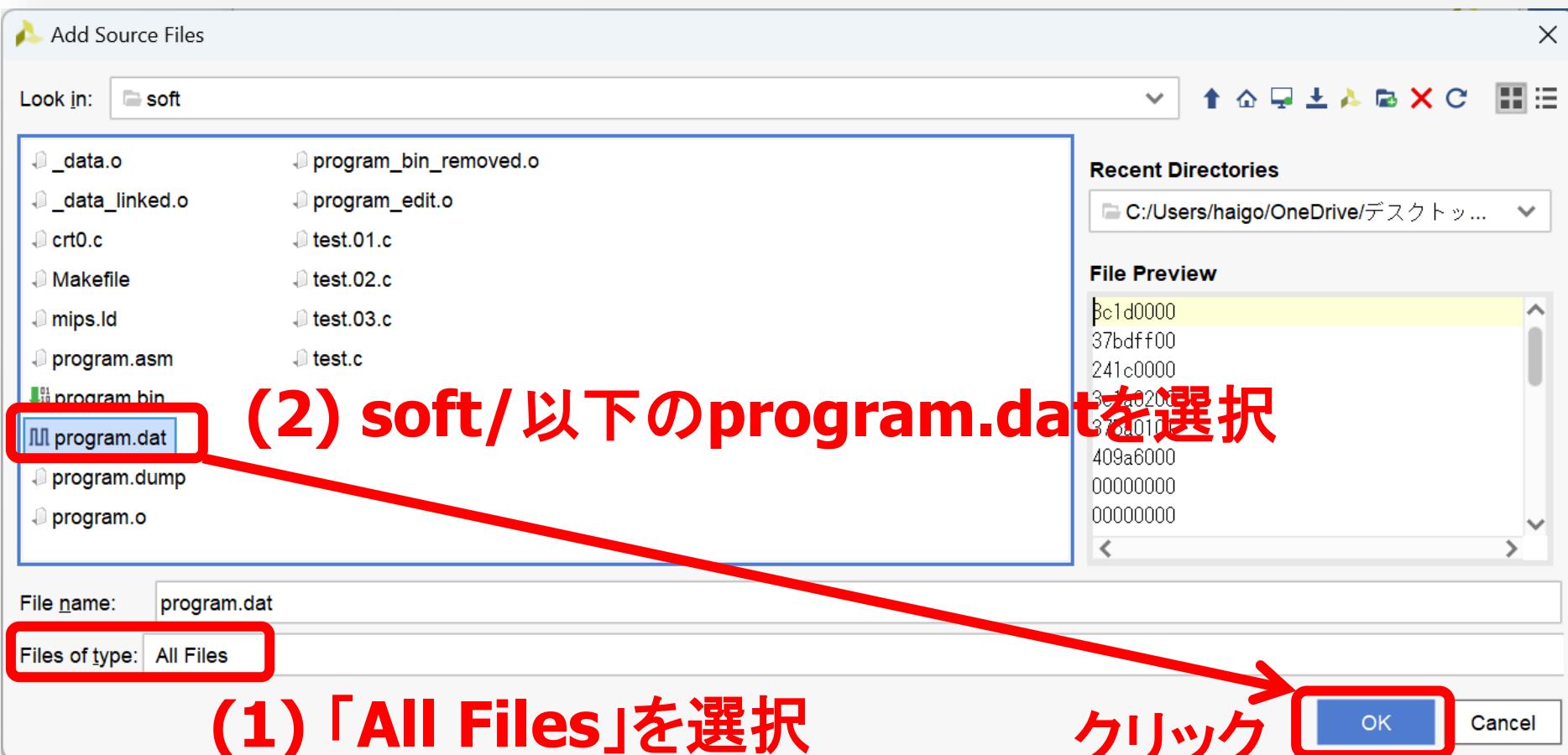
# プロジェクトの新規作成(7/14)

- 「Create Project」→各種質問に答える



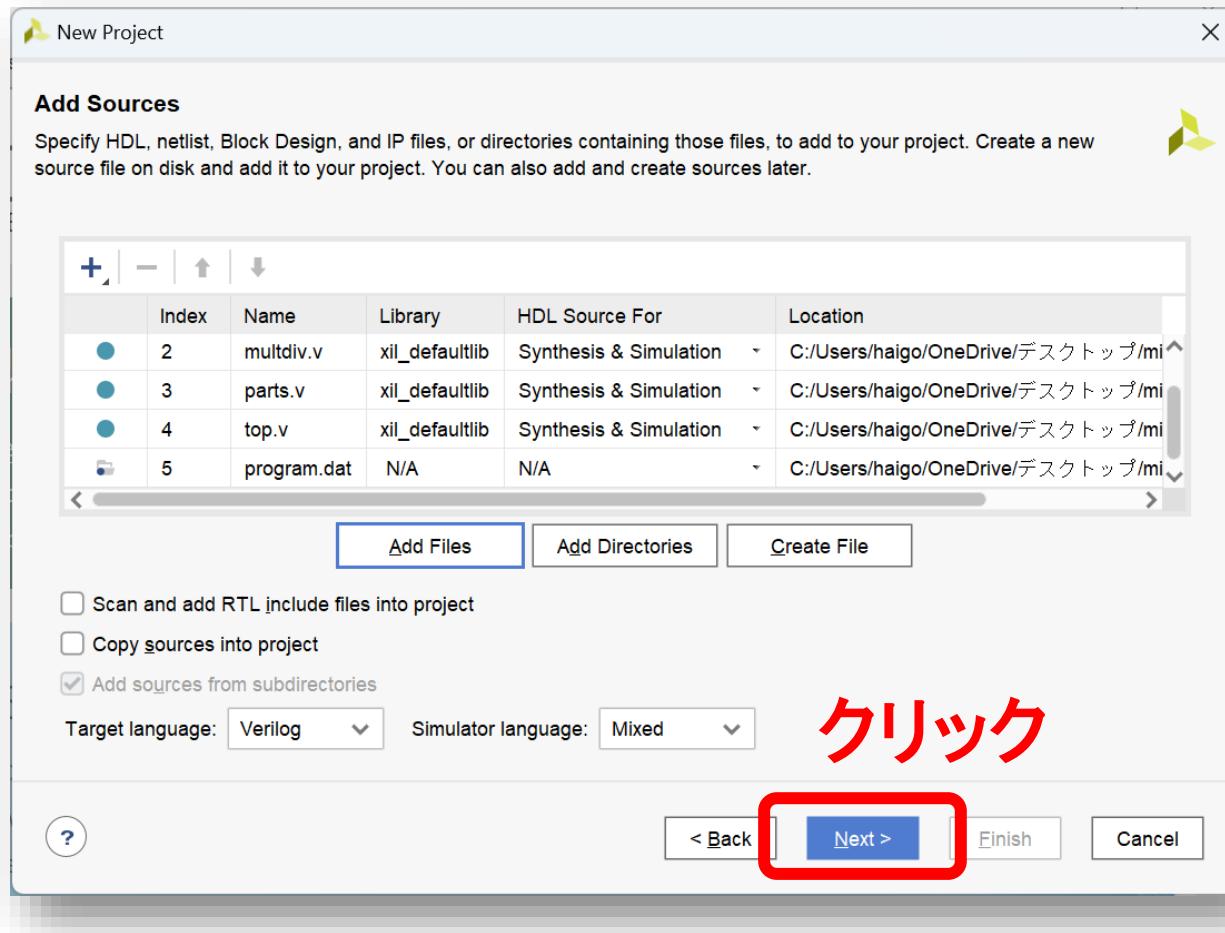
# プロジェクトの新規作成(8/14)

- 「Create Project」→各種質問に答える



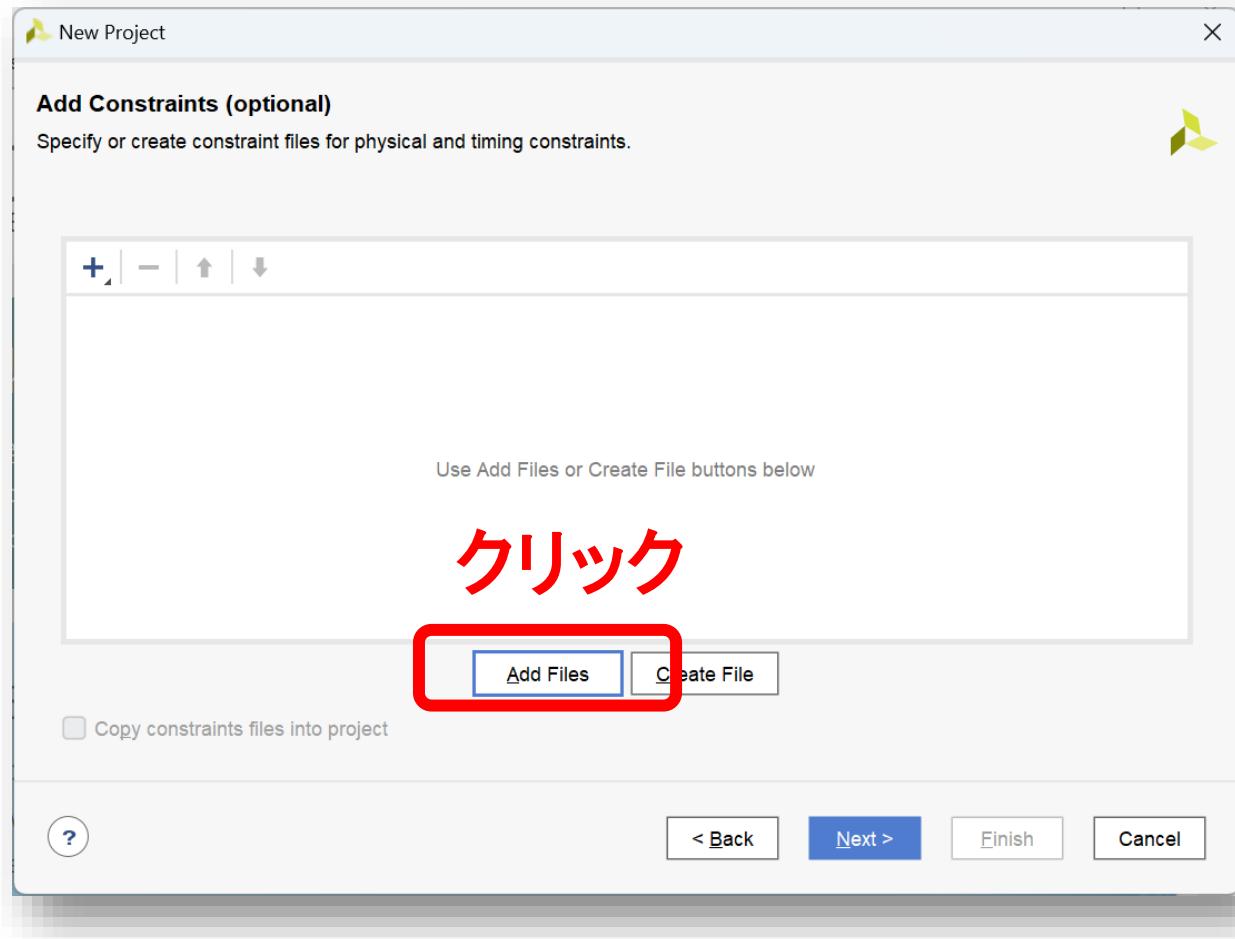
# プロジェクトの新規作成(9/14)

- 「Create Project」→各種質問に答える



# プロジェクトの新規作成(10/14)

- 「Create Project」→各種質問に答える

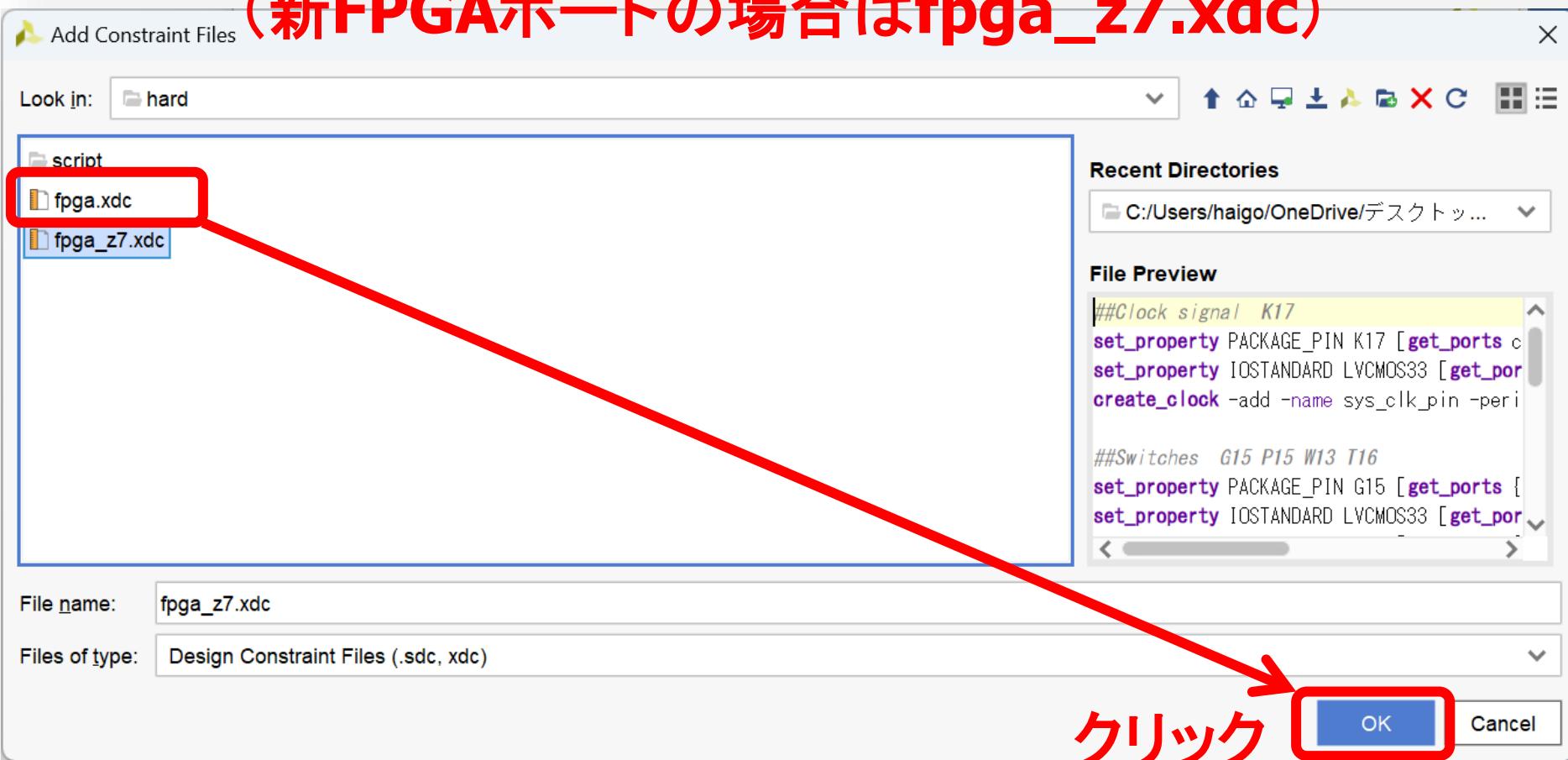


# プロジェクトの新規作成(11/14)

- 「Create Project」→各種質問に答える



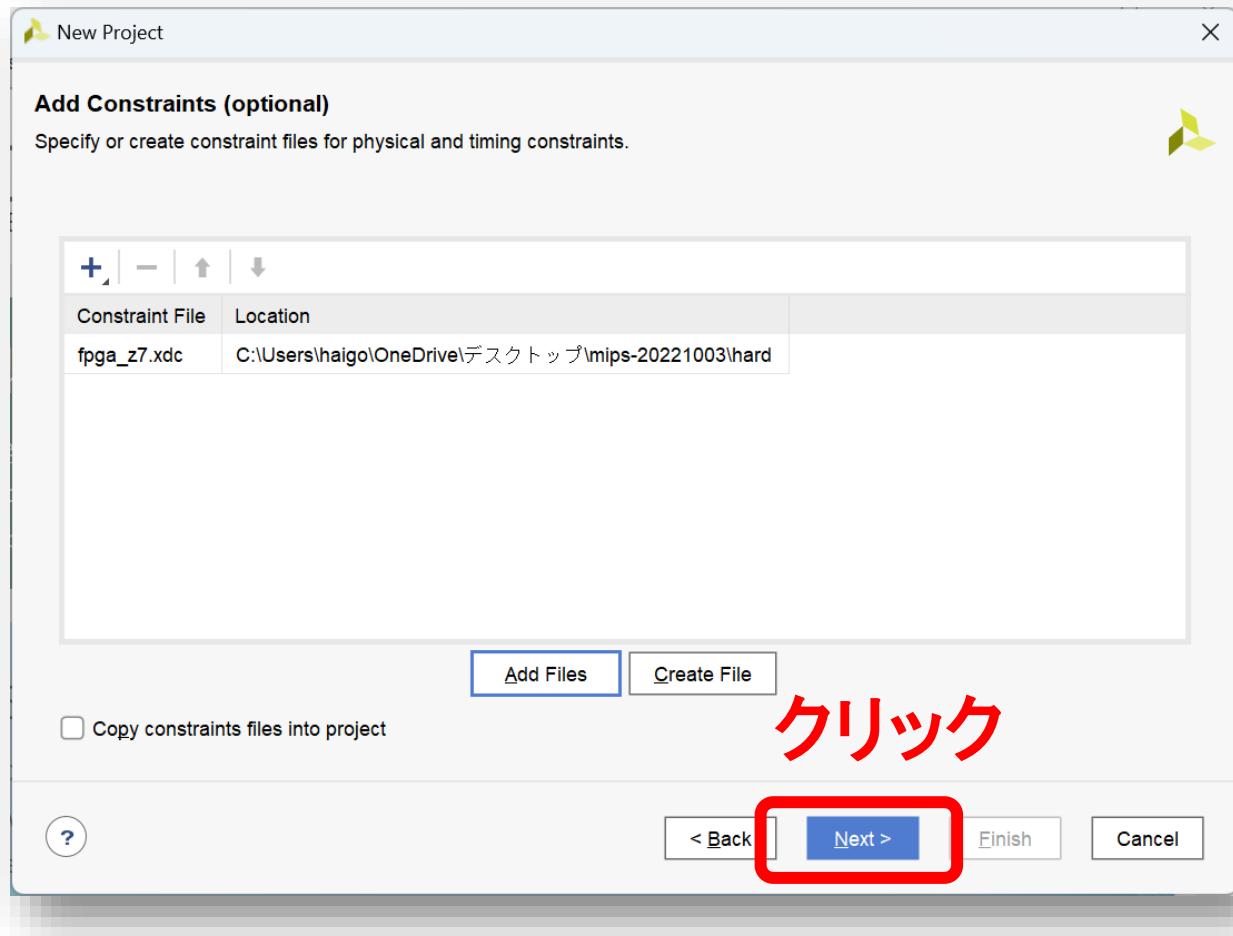
hard/以下のfpga.xdcを選択  
(新FPGAボードの場合はfpga\_z7.xdc)



クリック

# プロジェクトの新規作成(12/14)

- 「Create Project」→各種質問に答える



# プロジェクトの新規作成(13/14)

- 「Create Project」→各種質問に答える

(2) xc7z010clg400-1を選択

(新FPGAボードの場合はxc7z020clg400-1)



(1) 下記のように選択

Default Part  
Choose a default Xilinx part or board for your project.

Parts | Boards

Reset All Filters

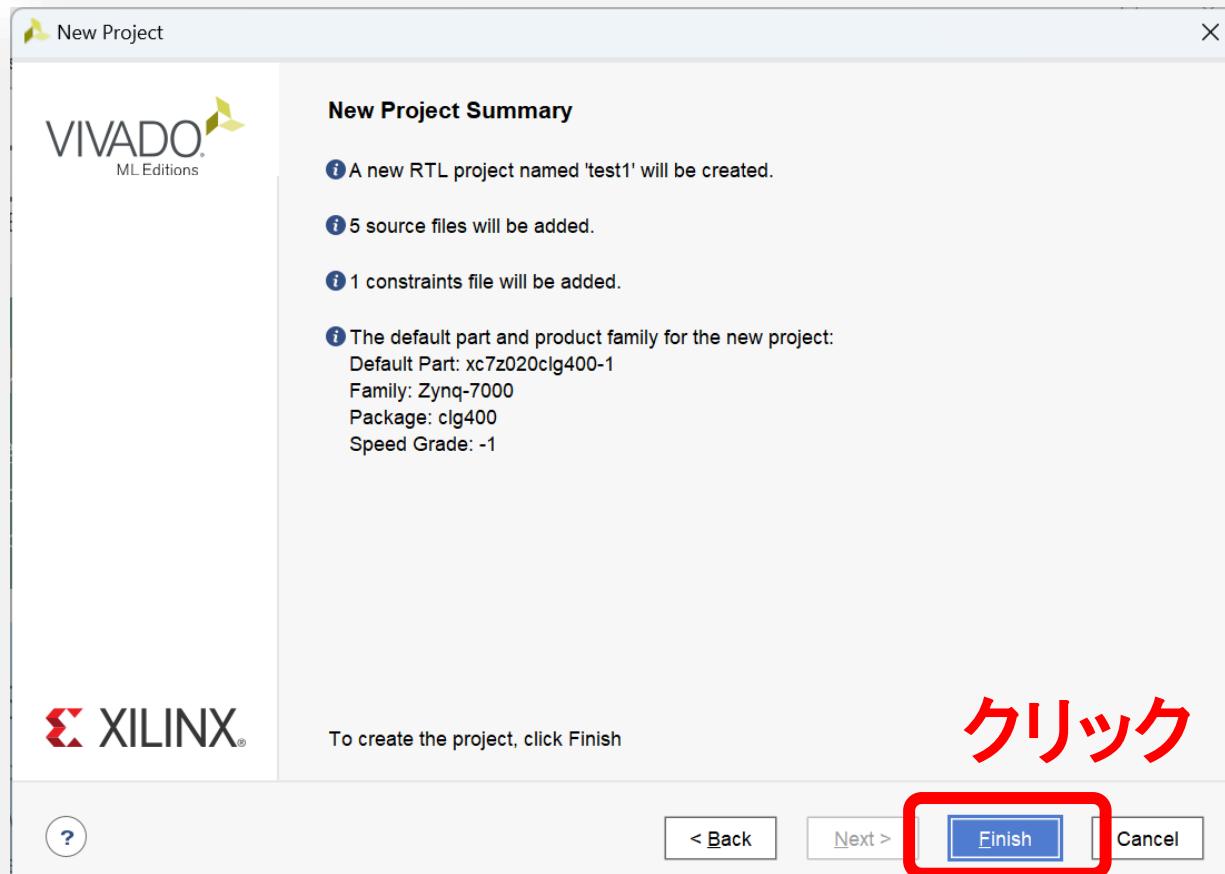
Category:	General Purpose	Package:	clg400	Temperature:	All Remaining
Family:	Zynq-7000	Speed:	-1	Static power:	All Remaining
Search:	Q-				
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs
xc7z007scl400-1	400	100	14400	28800	50
xc7z010clg400-1	400	100	17600	35200	60
xc7z014scl400-1	400	125	40800	81200	107
xc7z020clg400-1	400	125	53200	106400	140

< Back Next > Finish Cancel

(3) クリック

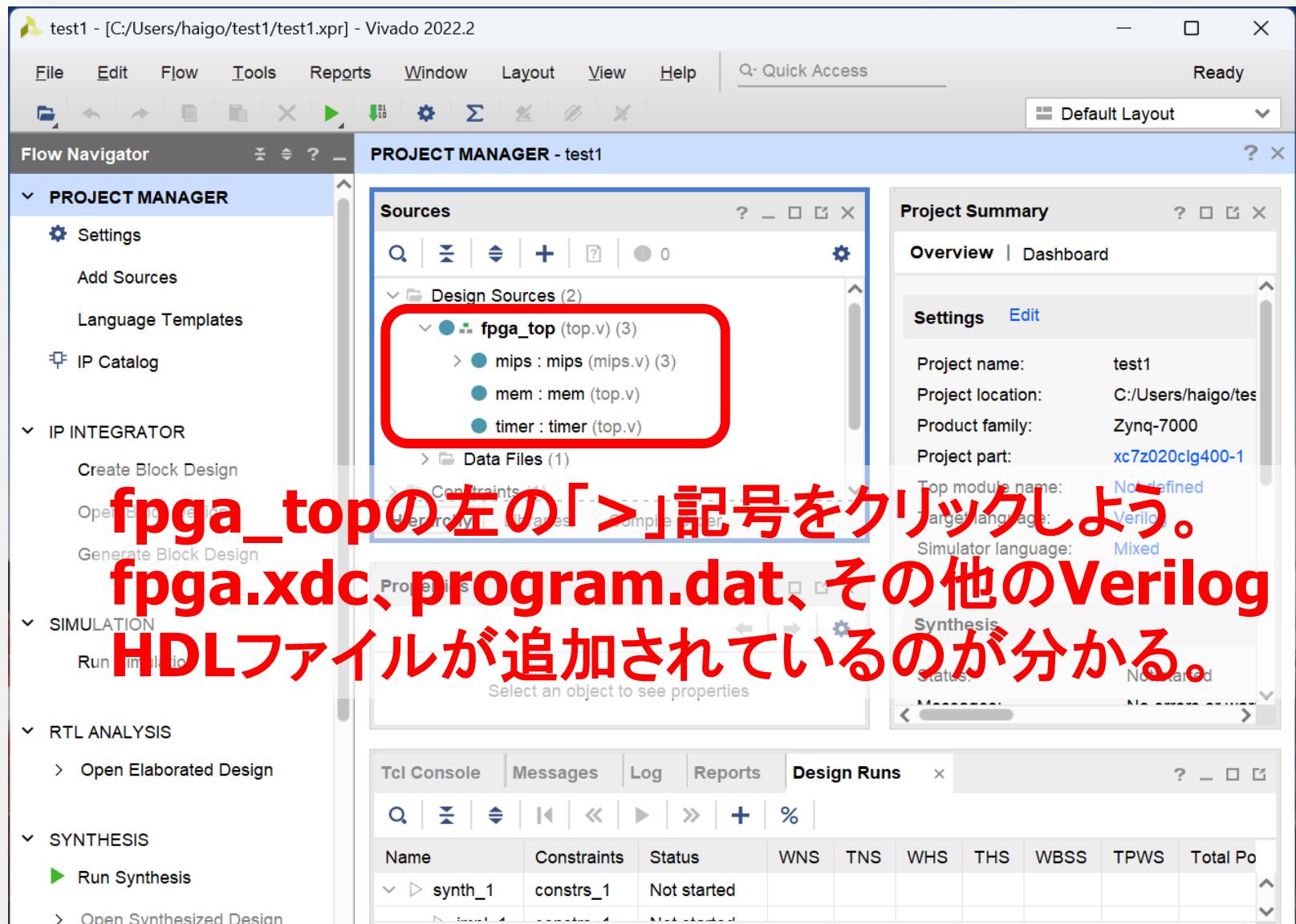
# プロジェクトの新規作成(14/14)

- 「Create Project」→各種質問に答える



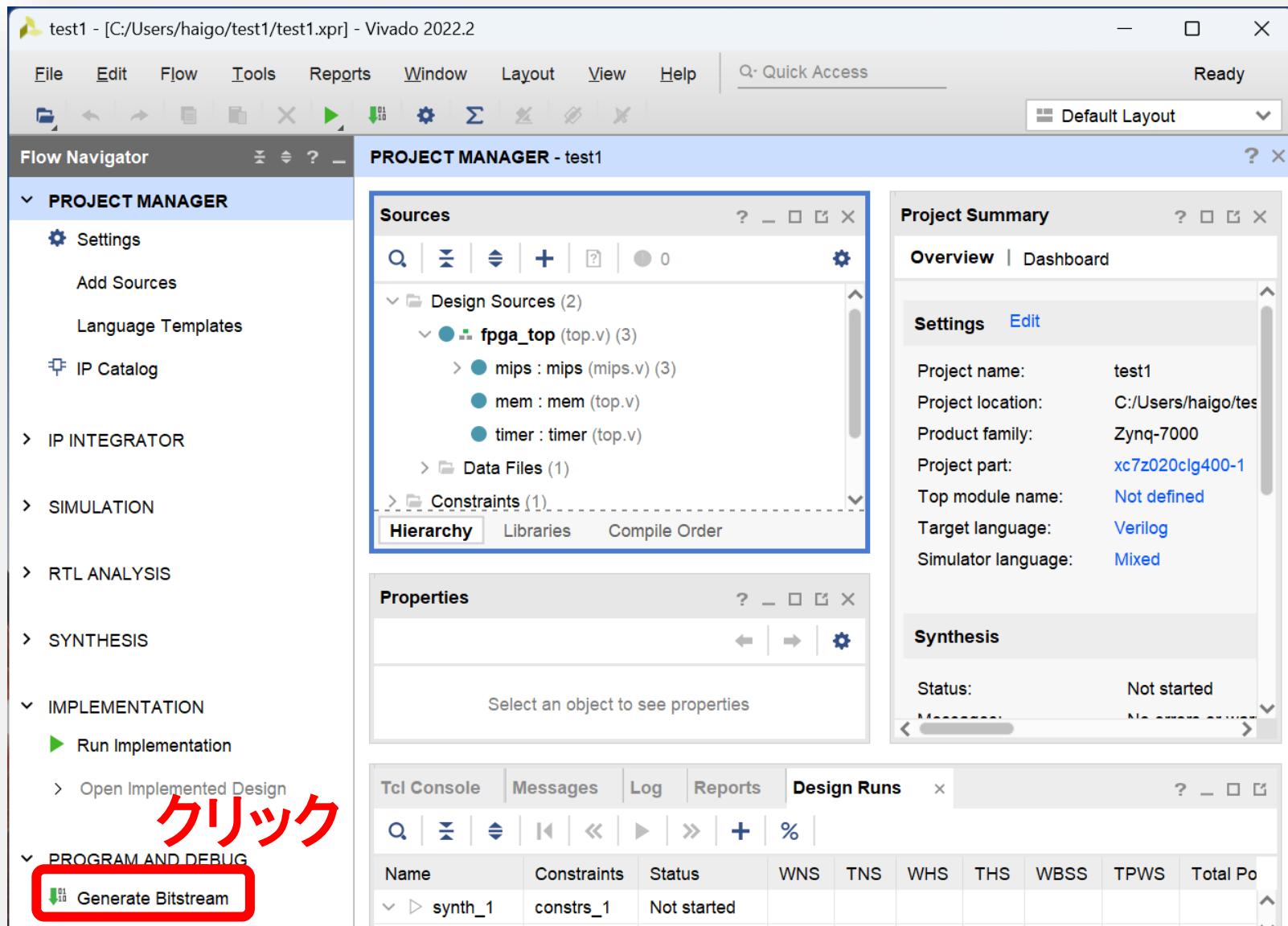
# ファイルの追加が完了

- Sourcesウィンドウにファイルの依存関係が表示



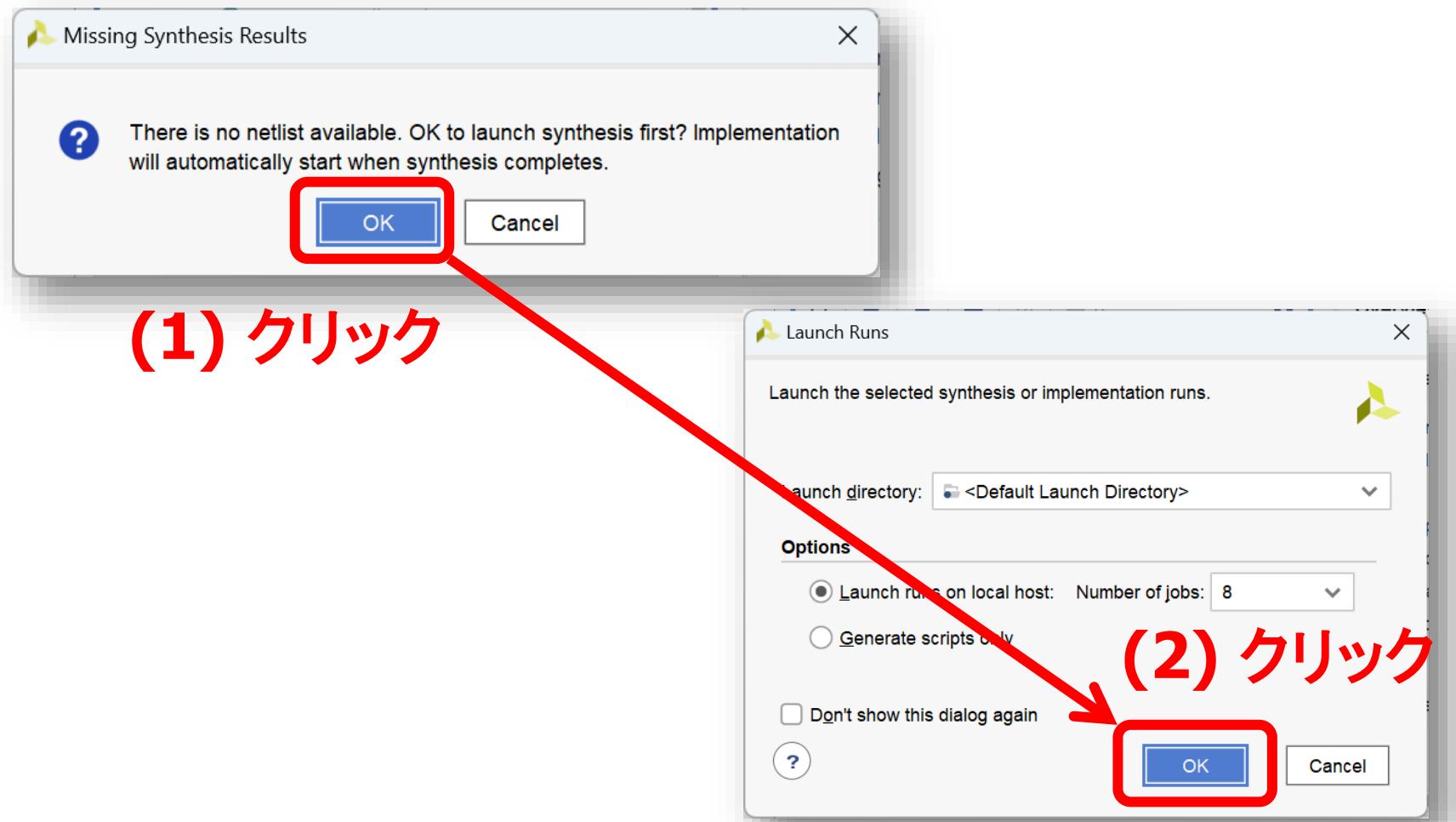
# 論理合成・配置配線(1/3)

- 「Generate Bitstream」をクリック



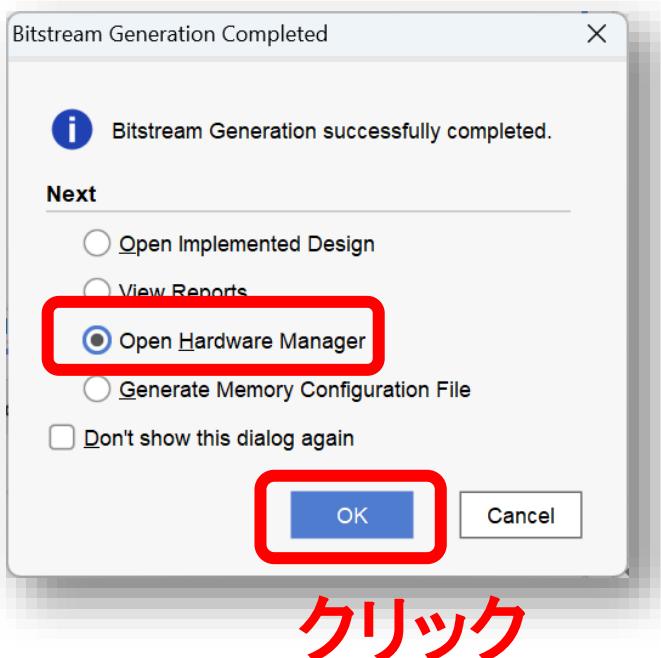
# 論理合成・配置配線(2/3)

- 「Generate Bitstream」をクリック
- すぐに確認ウィンドウが出る→「Yes」をクリック



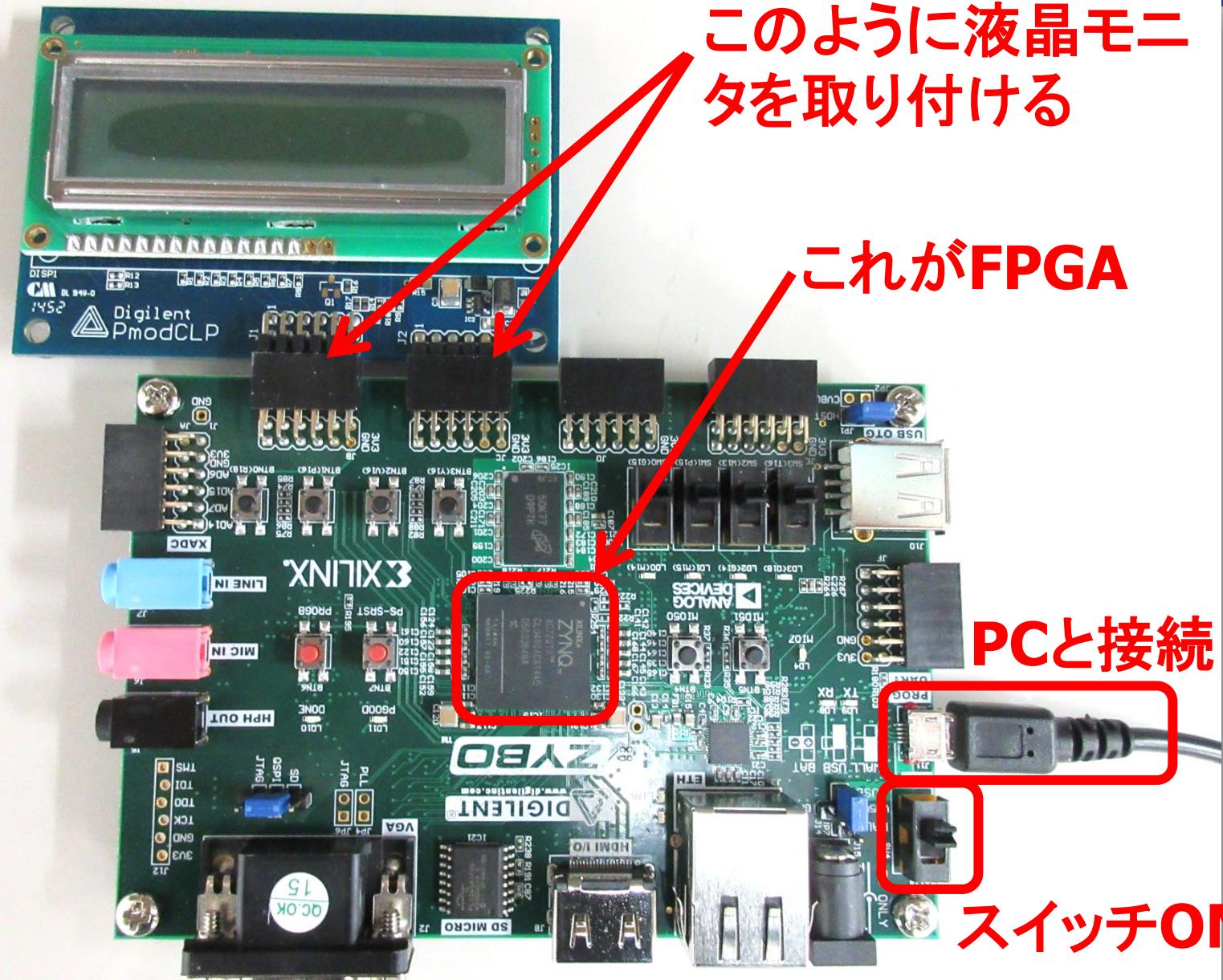
# 論理合成・配置配線(3/3)

- 配置配線まで終わると別の確認ウィンドウが出る  
→「Open Hardware Manager」を選んで「OK」



クリック

# FPGAボードとPCをつないで電源オン



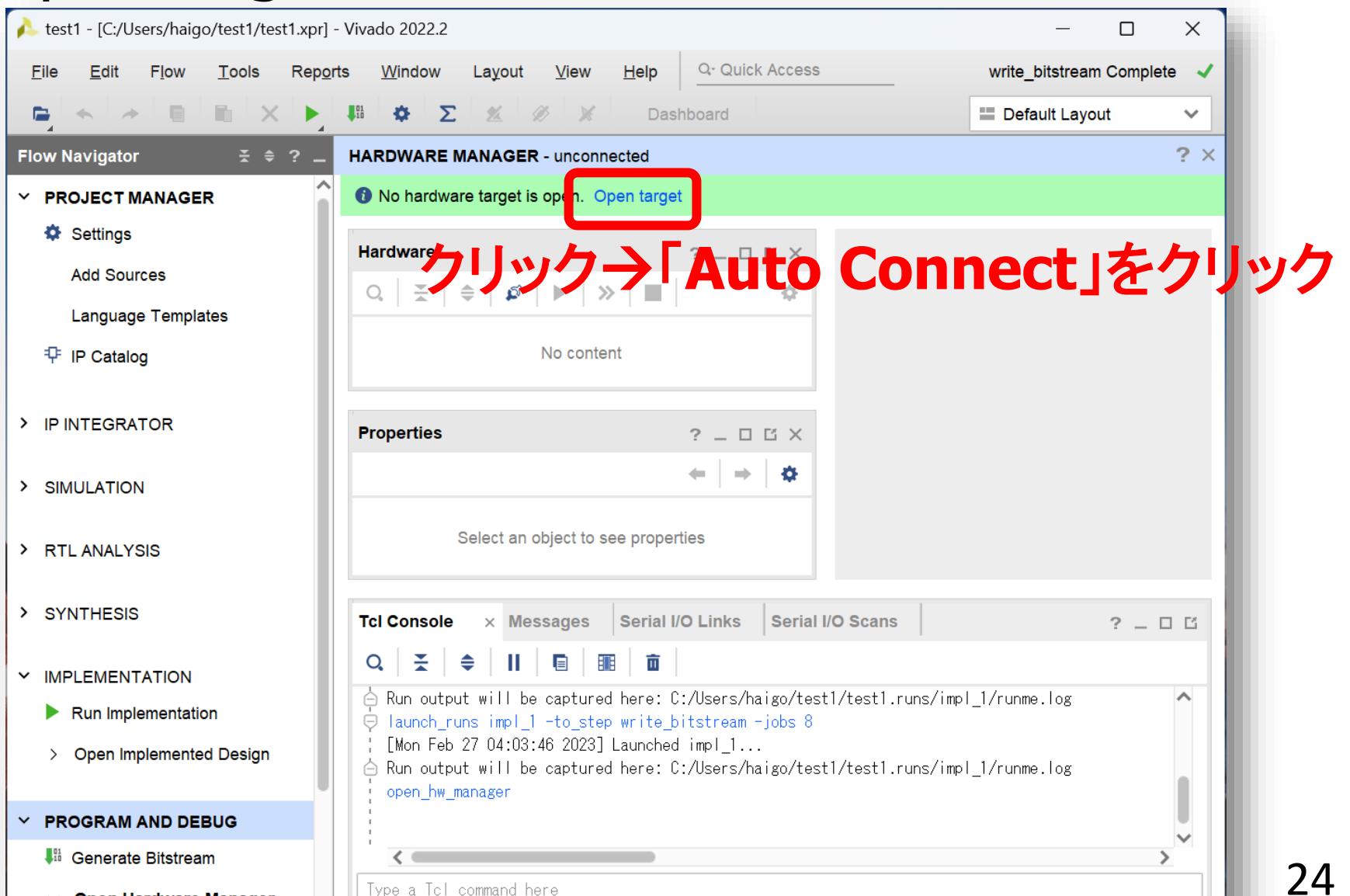
# FPGAへの書き込み

- ITCマシン(現地参加)
  - スライド24へ
- 自宅マシン
  - WindowsマシンにDigilent社のAdept2をインストール
  - 下記URLから `digilent.adept.system_v2.19.2.exe` をダウンロード

<https://reference.digilentinc.com/reference/software/adept/previous-versions>

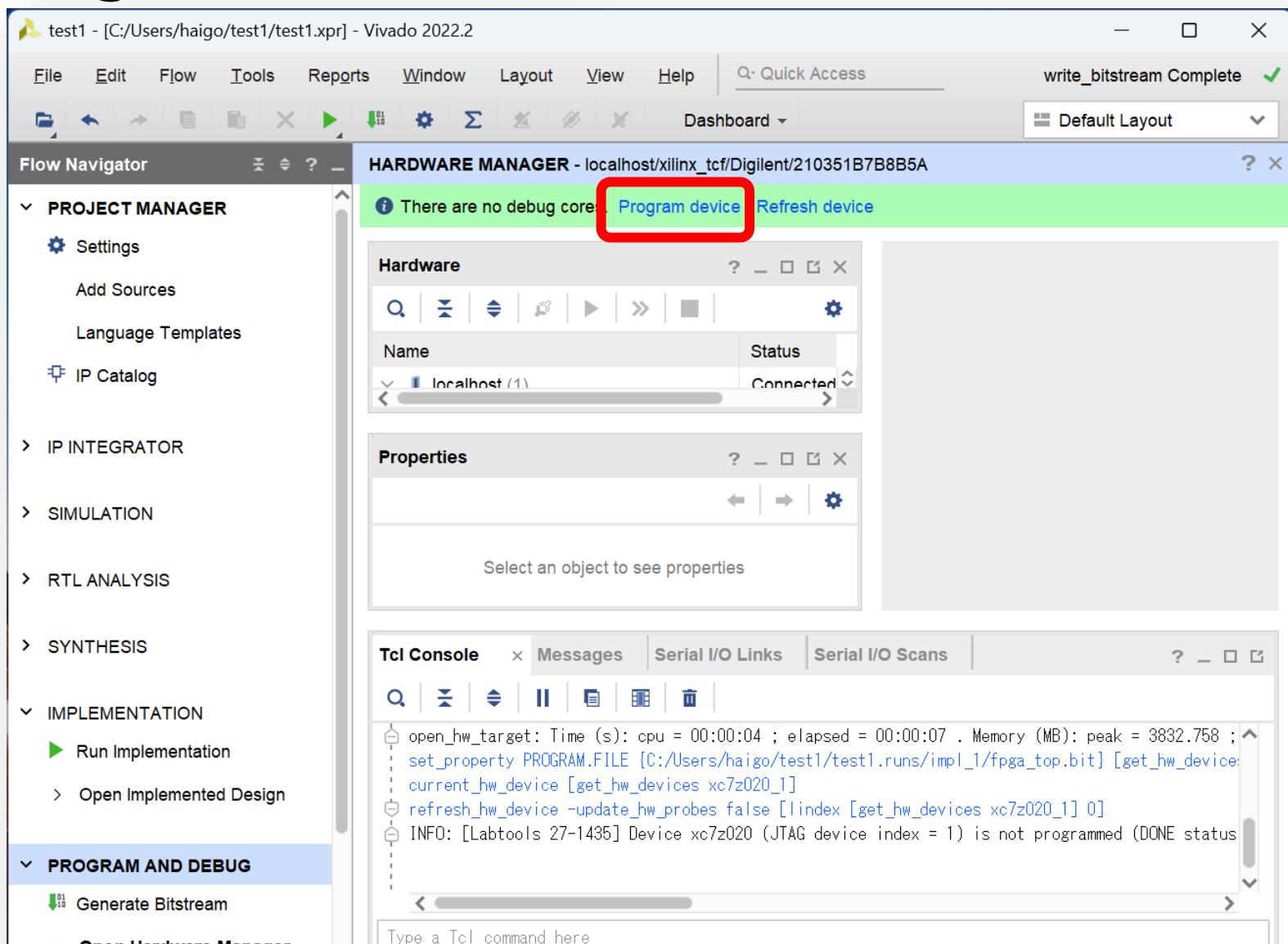
# FPGAへの書き込み@ITCマシン(1/3)

- 「Open target」→「Auto Connect」の順にクリック



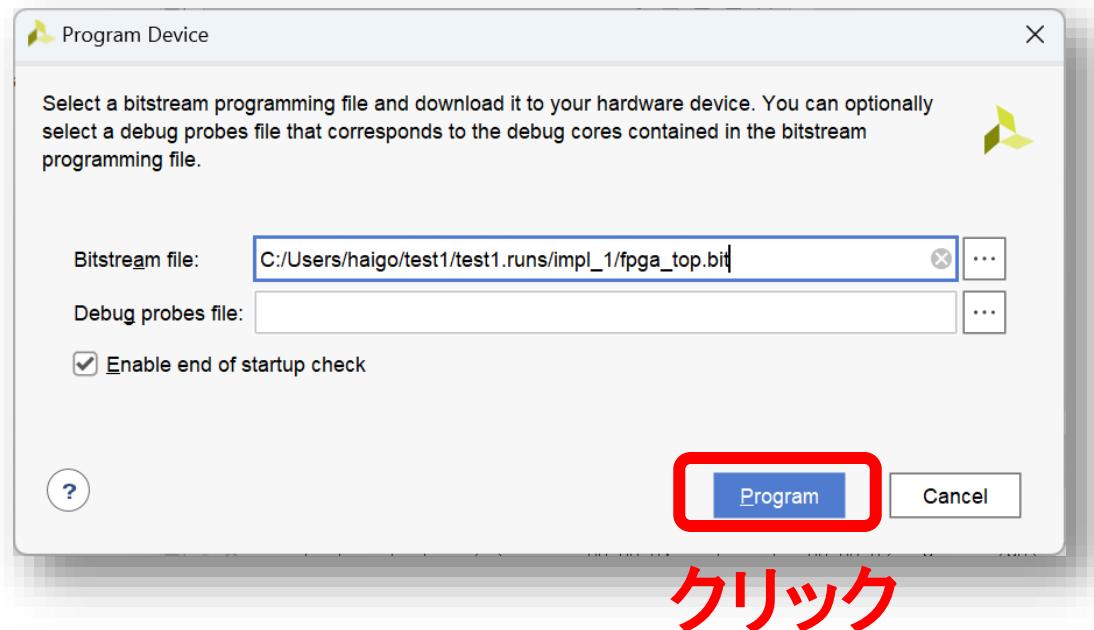
# FPGAへの書き込み@ITCマシン(2/3)

- 「Program device」をクリック



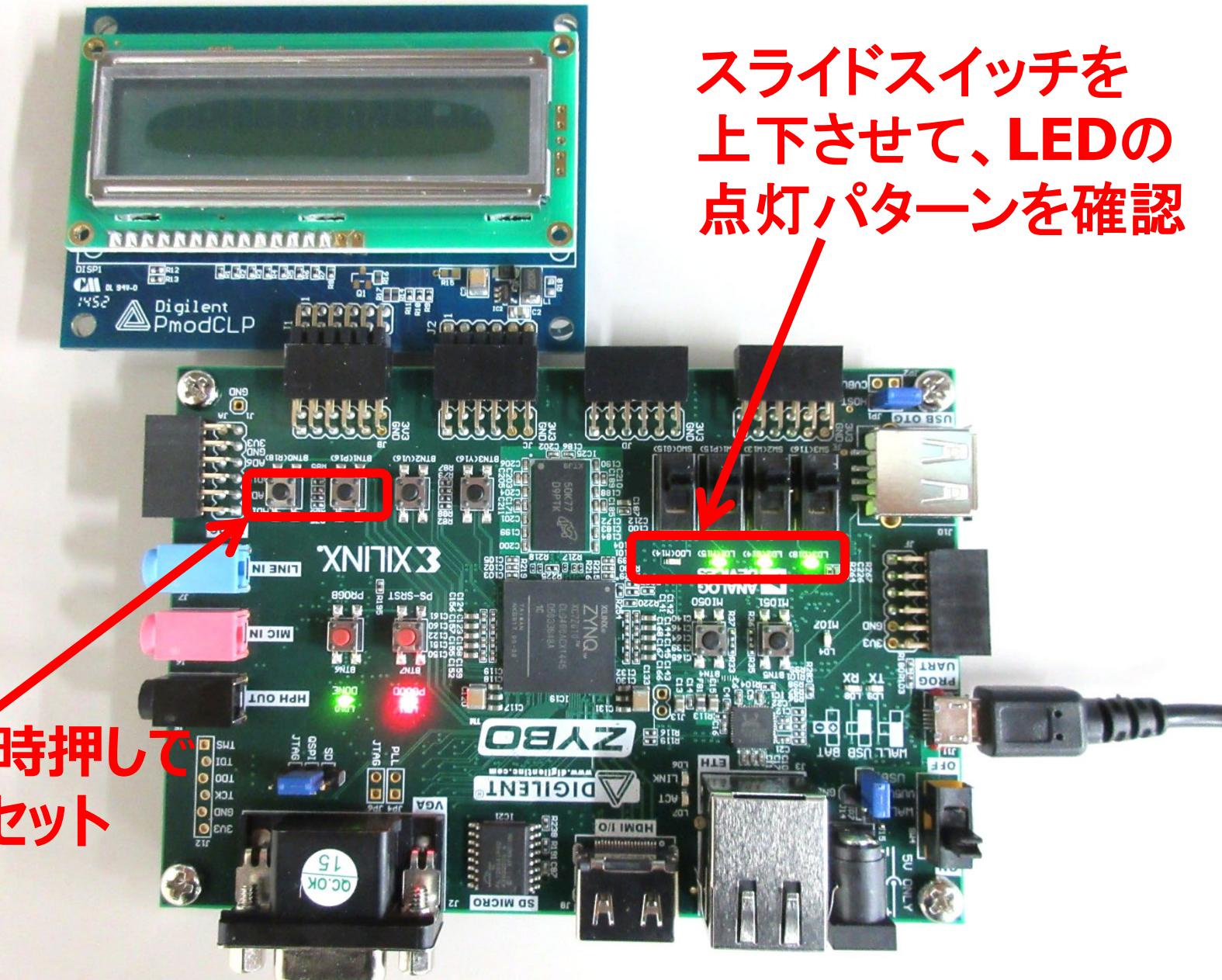
# FPGAへの書き込み@ITCマシン(3/3)

- Bitstream file選択画面が出たら「Program」をクリック



クリック

# FPGAの動作確認



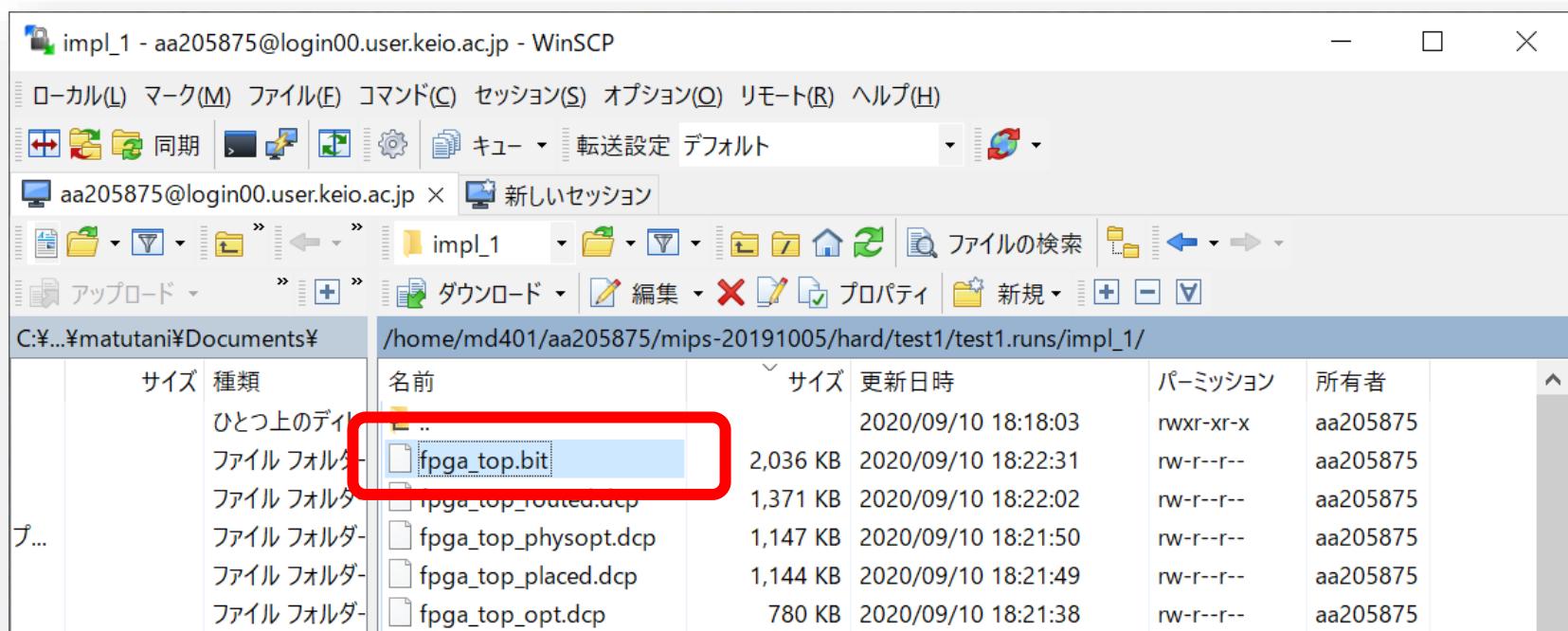
# FPGAへの書き込み@自宅マシン(1/4)

- ITCマシンで作成したビットストリームファイルを手元(自宅マシン)にコピー
  - ファイル名はfpga\_top.bitだが、ファイルの場所は論理合成配置配線の仕方によって異なるので要注意※
  - GUI(チュートリアル1だけ)の場合  
<Vivadoを起動したディレクトリ>/test1/test1.runs/impl\_1/fpga\_top.bit
  - CUI(チュートリアル2以降)の場合  
hard/ディレクトリの直下(つまり、hard/fpga\_top.bit)

※現地参加であればITCマシンだけで完結していたが、自宅から試す場合はFPGAへの書き込みだけ別マシン(=自宅マシン)で行うため、手順が煩雑になる。

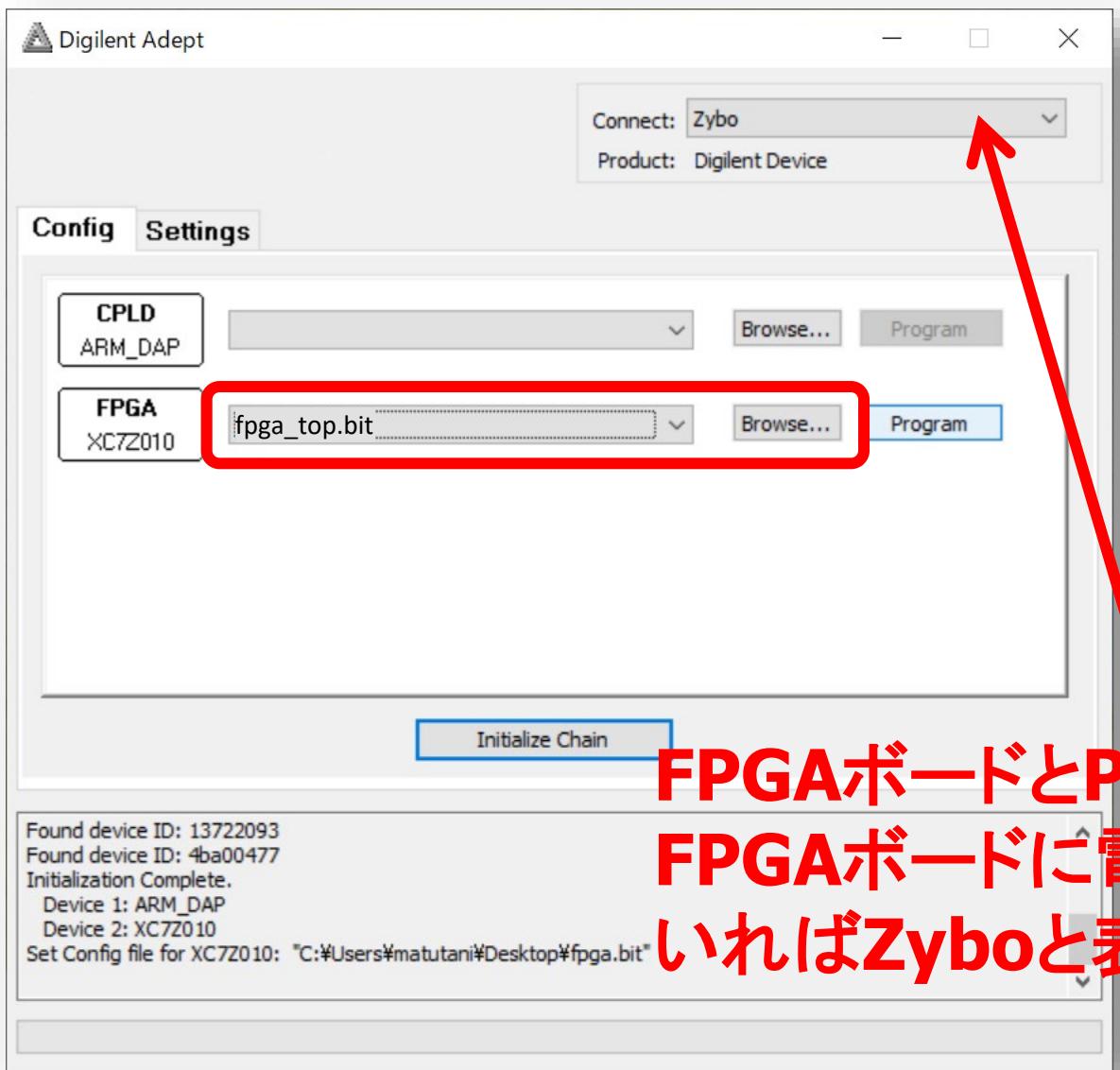
# FPGAへの書き込み@自宅マシン(2/4)

- ITCマシンで作成したビットストリームファイルを手元(自宅マシン)にコピー
  - ファイル名はfpga\_top.bitだが、ファイルの場所は論理合成配置配線の仕方によって異なるので要注意※
  - WinSCPによるファイル転送



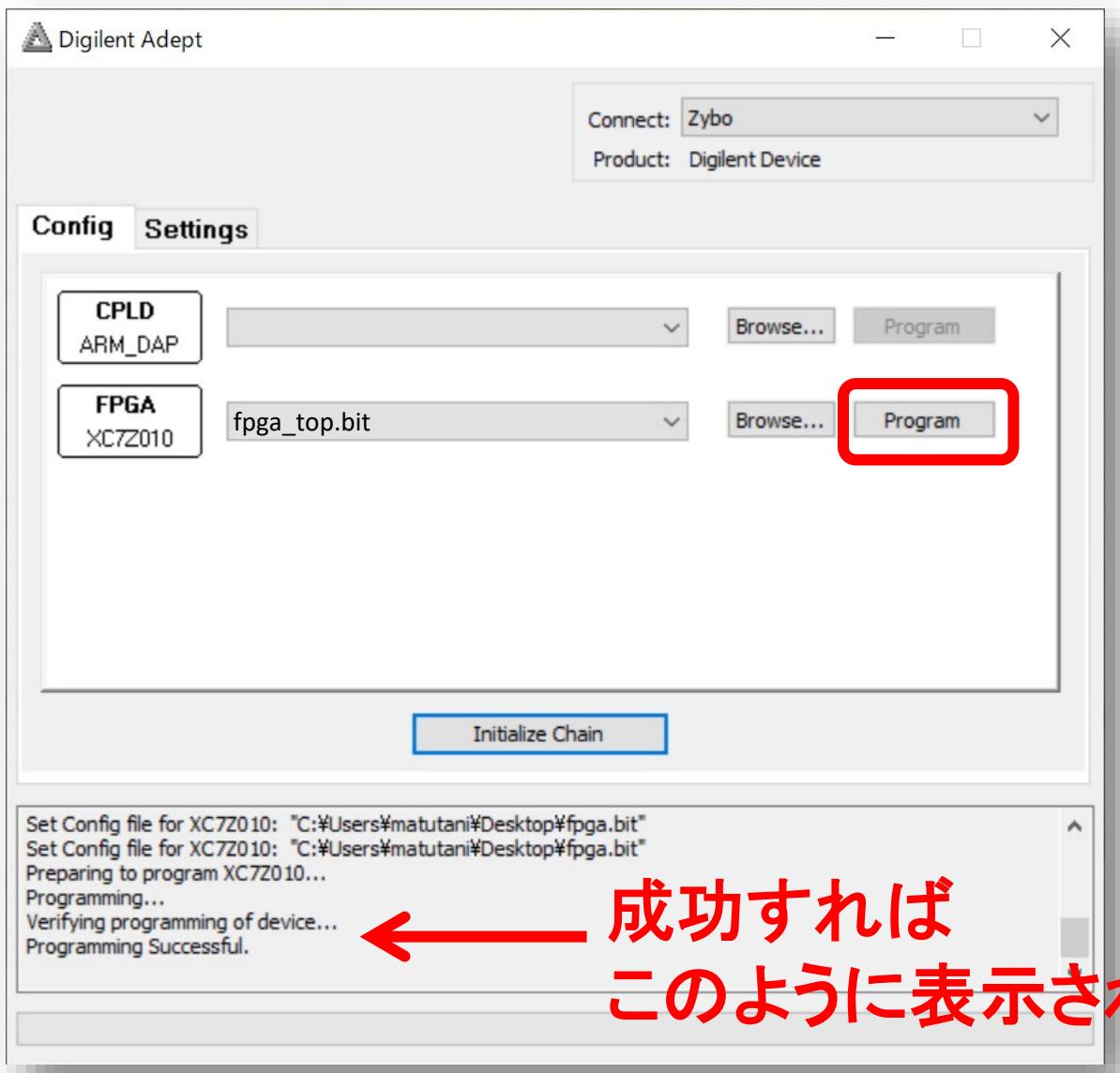
# FPGAへの書き込み@自宅マシン(3/4)

- Browseボタンを押してfpga\_top.bitを指定



# FPGAへの書き込み@自宅マシン(4/4)

- Programボタンを押す



# FPGAの動作確認

