



SWEST/ACRi 共同企画セッション

FPGA の「さわれる」 遠隔学習システムの利用体験

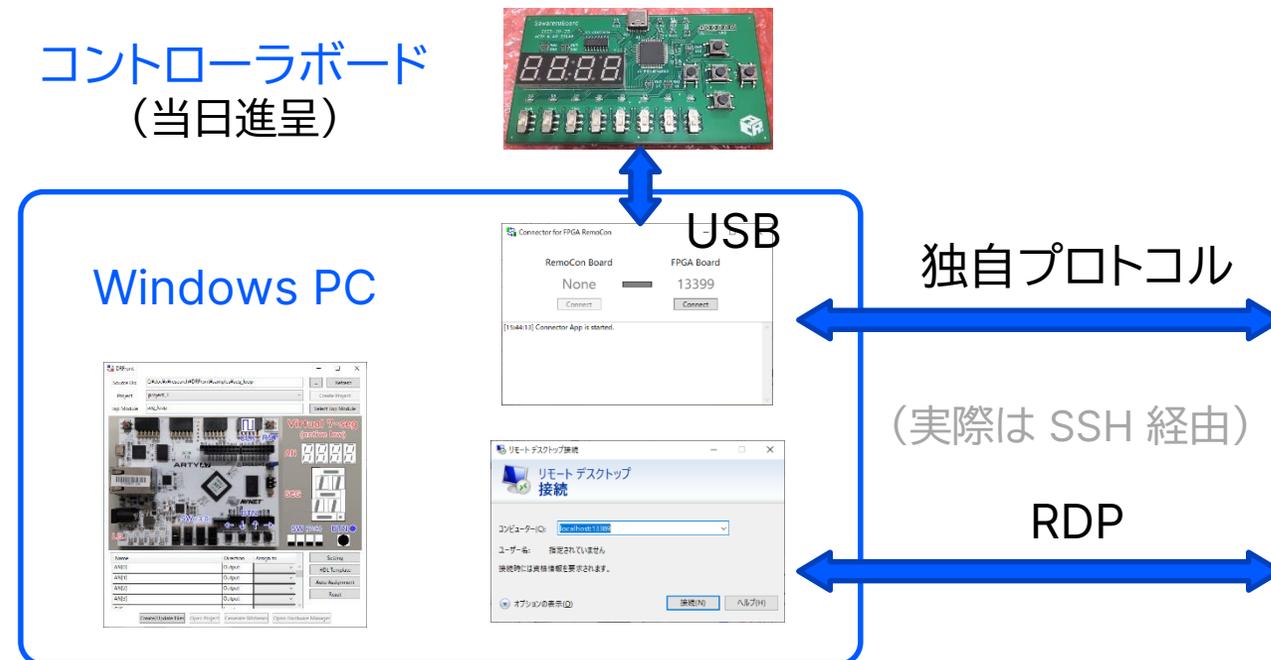
事前準備について、
あらかじめスライド2~6を確認し、
対応をお願いします

2024-08-30 SWEST26

藤枝 直輝(愛知工業大学)

今回の前提

- ◆ **ACRi ルーム**のアカウントを持っていること
 - リモートデスクトップでのログイン, **Vivado** の起動まで経験があること
- ◆ **Windows PC** と **USB Type-C ケーブル**を持参すること
 - PC はインターネット接続可能であること



ACRi ルーム



ACRi ルーム

◆ アカウント作成・サーバ予約は下記 URL から

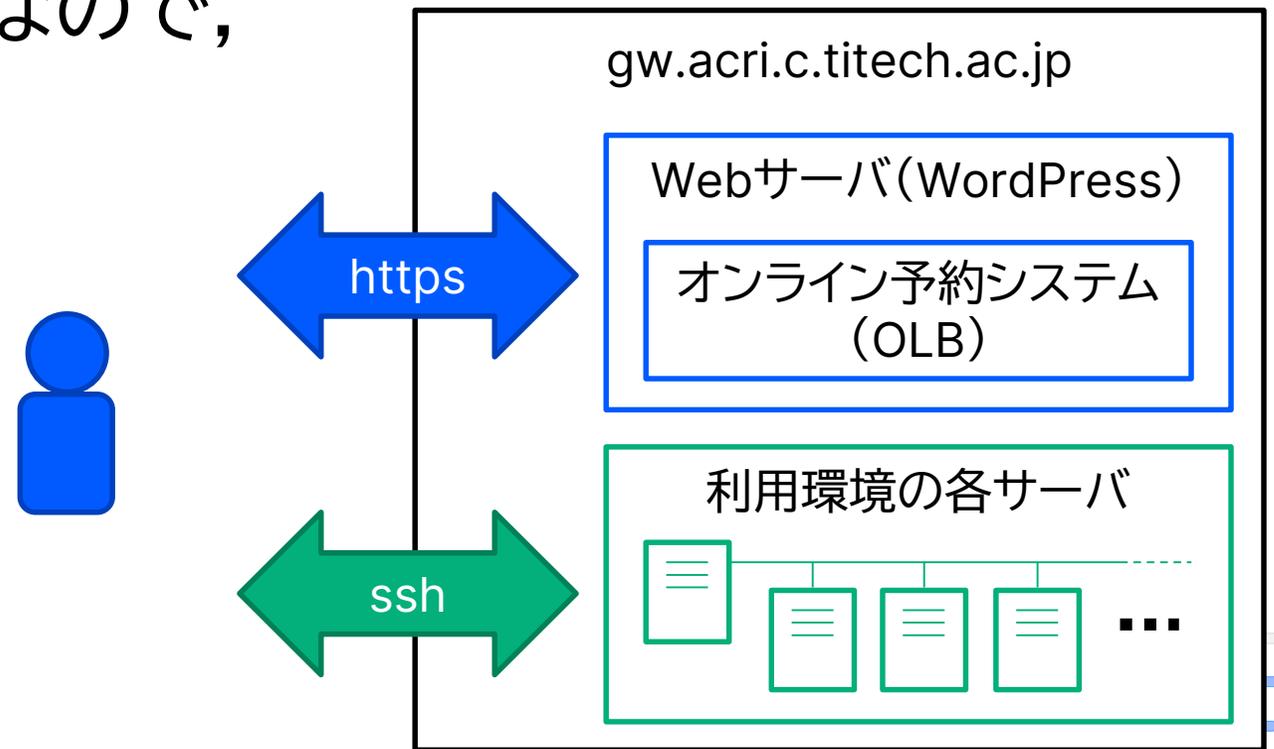
<https://gw.acri.c.titech.ac.jp/wp/>



The screenshot shows the ACRi website homepage. At the top is a dark blue header with the ACRi logo and name. Below the header is a main content area with a post titled "ACRi ルームへようこそ!". The post includes a date range (2023.07.19 to 2020.06.14), a welcome message, a link to the account application page, and a maintenance notice for August 10-16, 2023. Below the post is a "日別スケジュール" (Daily Schedule) section with navigation buttons for previous/next days and a dropdown menu for server selection. On the right side, there is a sidebar with a search bar, a navigation menu with links like "ようこそ", "予約ページトップ", and "ログイン/ログアウト", and a "ACRi ルームの利用説明" (ACRi Room Usage Guide) link at the bottom.

2種類のアカウ

- ◆ アカウ申請は数日以内に受理
 - 予約システムのアカウについてのメールが届く
 - 利用環境サーバのアカウについてのメールも届く
- ◆ それぞれパスワードが別なので、混同に注意



当日に向けた準備 (1)

- ◆ 8月30日 ~~12:00~15:00~~ 09:00~12:00 のサーバを予約
 - サーバは vs で始まるものであればどれでもよい
 - 日付を選択して [移動] ボタンでその日の予約状況が表示される

日別スケジュール

<前日> 2023-08-10 * <翌日> 移動 サーバ: 全て表示

サーバ	as001 (U200)	as002 (U250)	as003 (U280-ES1)	as004 (U50)	as005 (VCK5000)	vs001	vs002	vs003
00:00	Open	Open	Open	Open	Open	Open	Open	Open
03:00	Open	Open	Open	Open	Open	Open	Open	Open
06:00	Open	Open	Open	Open	Open	Open	Open	Open
09:00	Open	Open	Open	Open	Open	Open	Open	Open
12:00	Open	Open	Open	Open	Open	Open	Open	Open
15:00	Open	Open	Open	Open	Open	Open	Open	Open
18:00	-	-	-	-	-	-	-	-
21:00	-	-	-	-	-	-	-	-

当日に向けた準備 (2)

- ◆ システムの配布パッケージをダウンロード・展開しておく

<https://github.com/nfproc/SawareruSys>

N. Fujieda and A. Okuchi. A Novel Remote FPGA Lab Platform Using MCU-based Controller Board. In 12th International Conference on Teaching, Assessment and Learning for Engineering (TALE 2023), pp. 188-193, 2023, DOI: [10.1109/TALE56641.2023.10398409](https://doi.org/10.1109/TALE56641.2023.10398409).

ユーザ向けの説明は、ユーザマニュアルを参照してください。マニュアルは、本リポジトリの Manual/main.pdf または配布パッケージの UserManual.pdf にあります。

配布パッケージのダウンロード: [ZIP \(37.3 MiB\)](#)

本リポジトリの内容:

- Board/ : コントローラボードの KiCad による設計ファイル一式
 - v2/ : 「FPGA リモコンボード V2」用
 - v4/ : 「FPGA リモコンボード V4」および「SawareruBoard V1」用

謝辞(および ACRI について)

◆ Adaptive Computing Research Initiative

- 要は, 日本の FPGA を盛り上げる会
- 2020年開設, 現在5年目
- 多くの企業からの協賛金により運営されています
(企業の皆様ありがとうございます)
- 本セッションは SWEST と ACRI の共同で企画されたものです
(SWEST 実行委員会の皆様ありがとうございます)

創設企業



ACRI参加企業



リアル ACri ルーム @ 東工大

◆ 基礎学習用

- 物理マシン10台 x
仮想マシン10台 = 100台
- 80台は Arty A7 搭載
- 10台は CMod A7 搭載
- 10台は Nexys A7 搭載

◆ アクセラレータ評価用

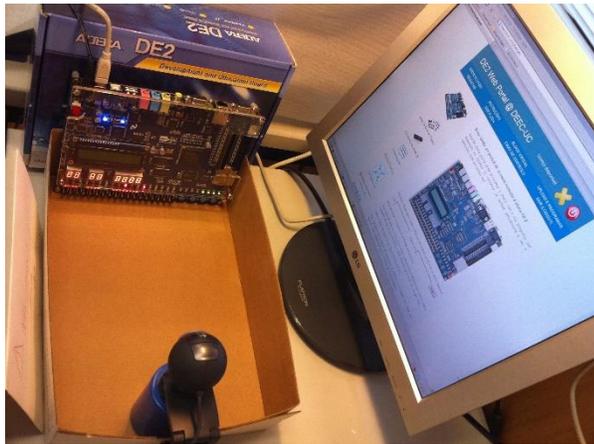
- 物理マシン5台
- Alveo または MI210 搭載



遠隔ゆえの難しさ.....

- ◆ FPGA ボードの入出力にアクセスする手段が乏しい
- ◆ WebCam や Web UI を使った先行研究はあるが.....
 - 利用者がハードウェアに触れていると感じづらい

WebCam を使った研究の例 [1]



[1] J. Soares and J. Lobo, Remote FPGA Laboratory for Digital Design Students, in REC 2011, pp. 95-98, 2011.

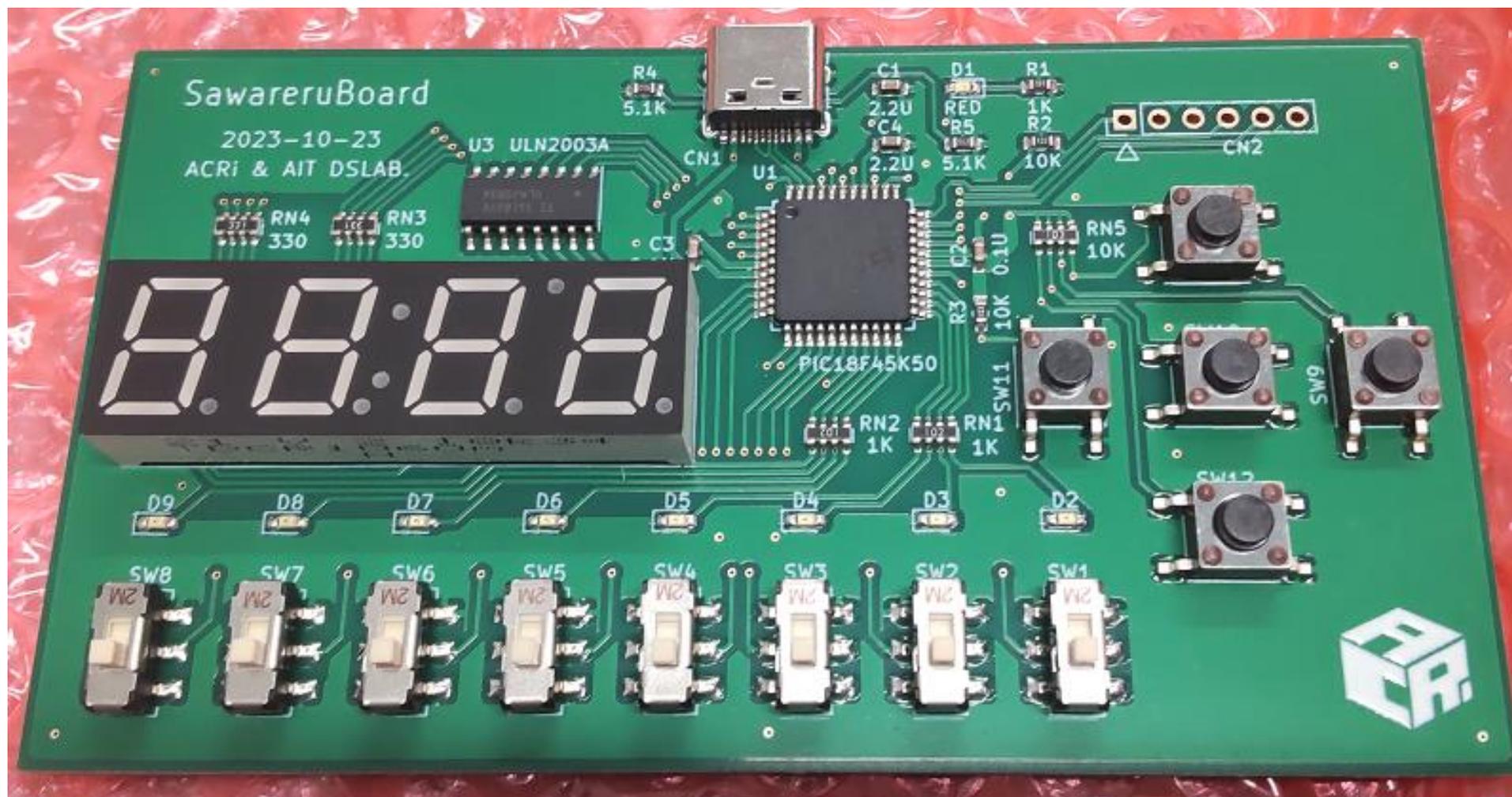
Web インタフェースを使った研究の例 [2]



[2] Y. Zhang et al., Remote FPGA lab platform for computer system curriculum, in ACM TUR-C '17, no. 3, pp. 1-6, 2017.

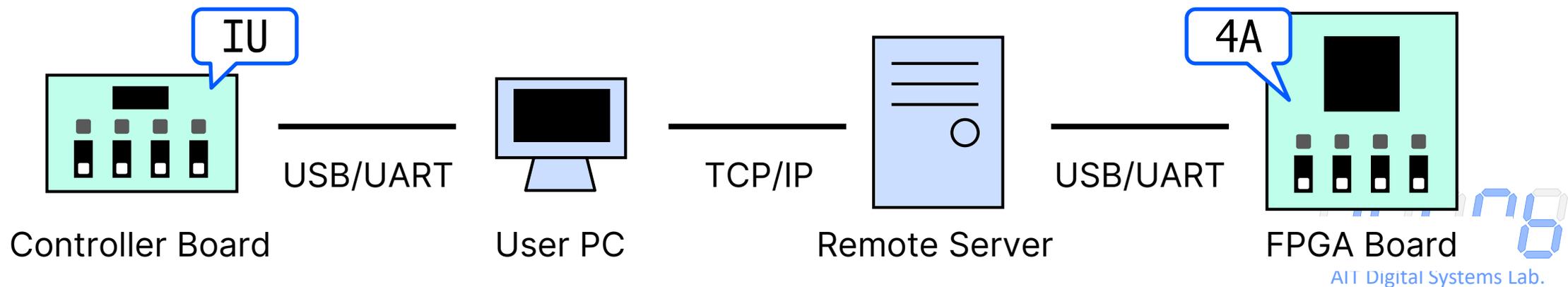
SawareruSys のアプローチ

- ◆ 安価な**コントローラボード**を配布/貸与する



動作原理(簡潔に)

- ◆ コントローラボードには USB 対応の PIC マイコン
- ◆ コントローラボードの**入力**・FPGA ボードの**出力の変化**を**コマンド文字列**に変換して送信
 - IU: 右端のスイッチがオンになった
 - 4A: 右端の LED が点灯した
- ◆ ユーザ PC と遠隔サーバは**コマンド**を TCP/IP で中継



必要なツールなど

◆ DRFront

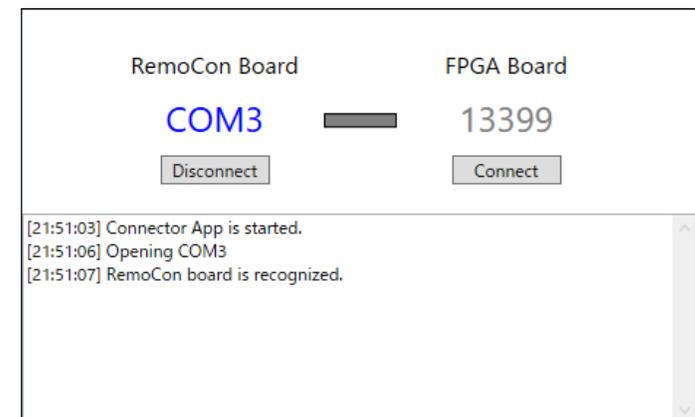
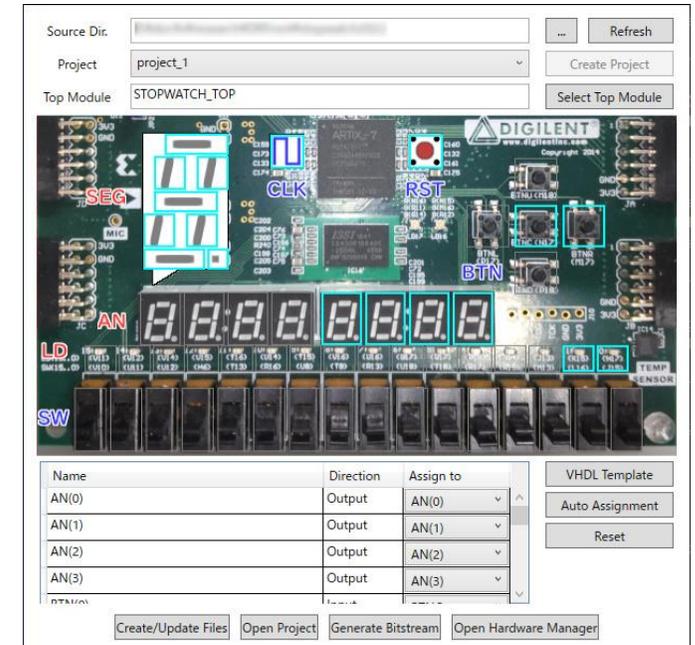
- DRFront¥DRFront.exe
- 動的再構成(Dynamic Reconfiguration)を用いたワークフローを支援するフロントエンド(Frontend)
- 事前に用意された入出力制御回路との接続回路や、Vivado 用のスクリプトを自動生成

◆ コネクタアプリ

- Connector¥Client¥Connector.exe
- ユーザ PC 側でのコマンド中継プログラム

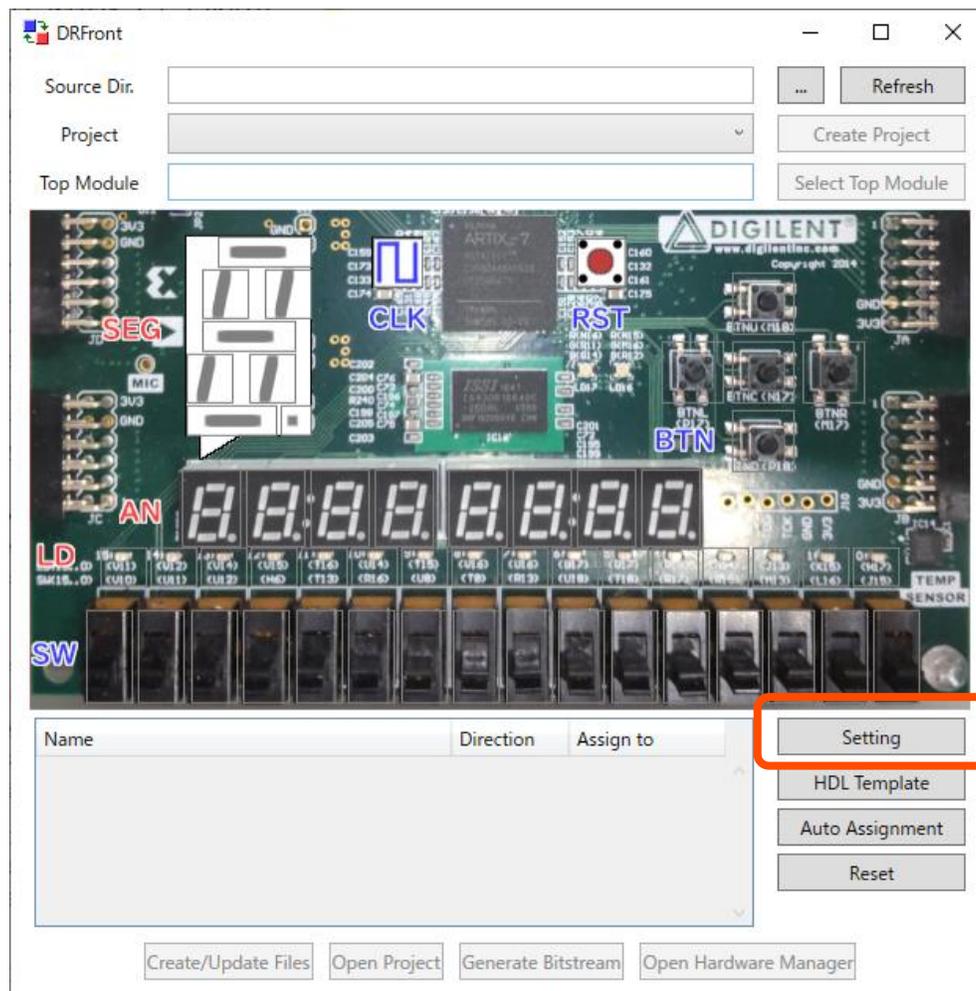
◆ 作業用のフォルダ

- 日本語を含まないパスで



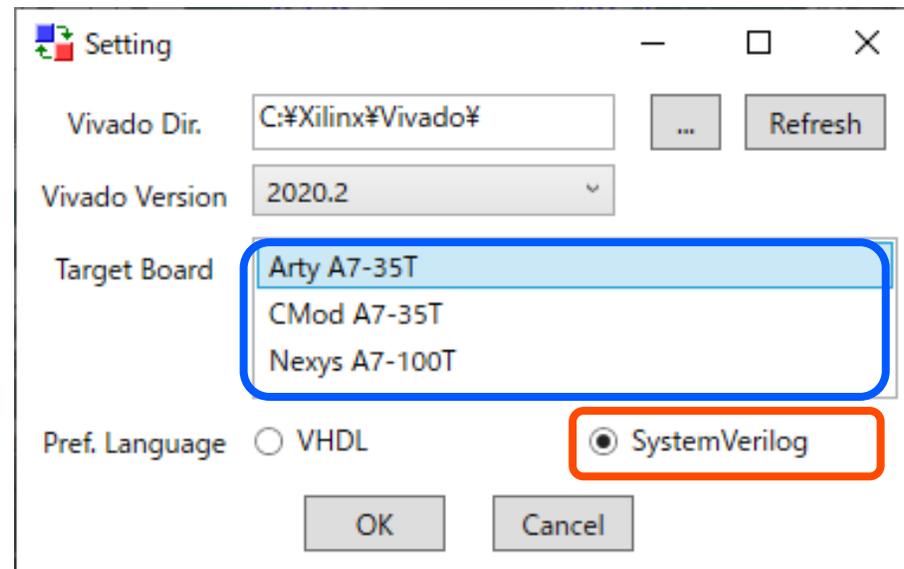
DRFront を起動

- ◆ 初期画面では, まずは **Setting** を押す



DRFront の初期設定

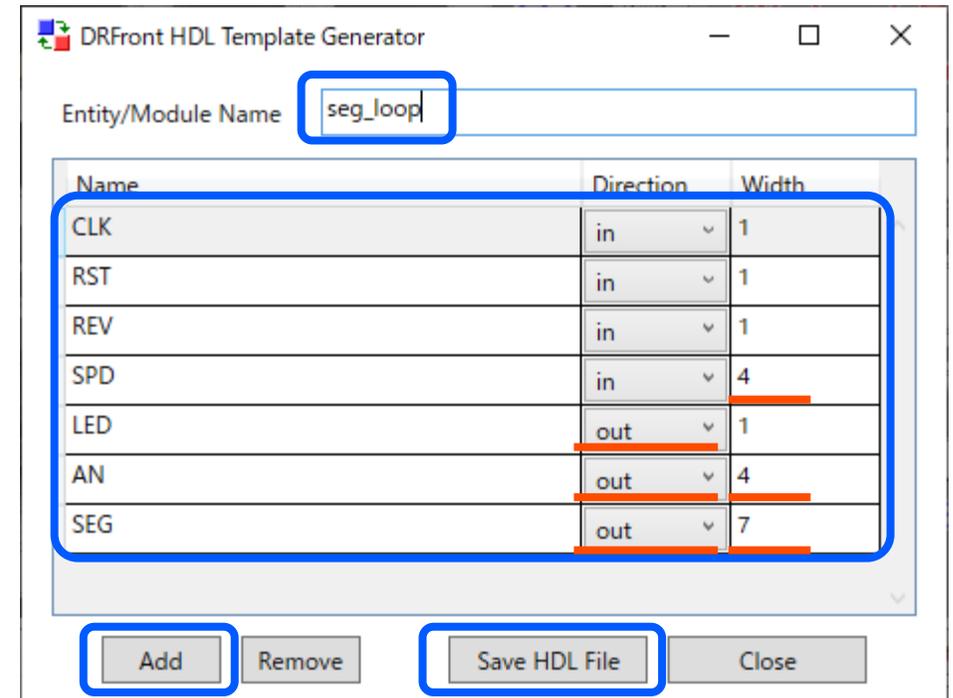
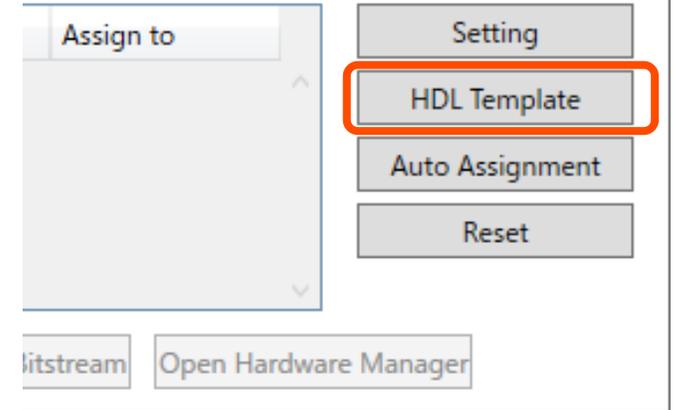
- ◆ Target Board を以下のいずれかに
 - vs001~vs710 を予約した場合: Arty A7-35T
 - vs801~vs810 を予約した場合: CMod A7-35T
 - vs901~vs910 を予約した場合: Nexys A7-100T
- ◆ Pref. Language を **SystemVerilog** に



回路のテンプレート作成

◆ HDL Template 画面を開く

- 回路名を適当に設定(ここでは `seg_loop`)
- Add ボタンを6回押す
- 表の各行をそれぞれ以下の通り設定
 - ◆ CLK, in(入力), 1ビット
 - ◆ RST, in(入力), 1ビット
 - ◆ REV, in(入力), 1ビット
 - ◆ SPD, in(入力), 4ビット
 - ◆ LED, out(出力), 1ビット
 - ◆ AN, out(出力), 4ビット
 - ◆ SEG, out(出力), 7ビット
- Save HDL File で作業用フォルダに保存



回路記述をコピーペースト

- ◆ Slack に掲示の回路記述をコピー
- ◆ 保存したテンプレートの endmodule の前にペースト

```
// NOTE: Template was generated by DRFront automatically.
```

```
module seg_loop (  
    input logic CLK,  
    input logic RST,  
    input logic REV,  
    input logic [3:0] SPD,  
    output logic LED,  
    output logic [3:0] AN,  
    output logic [6:0] SEG  
);
```

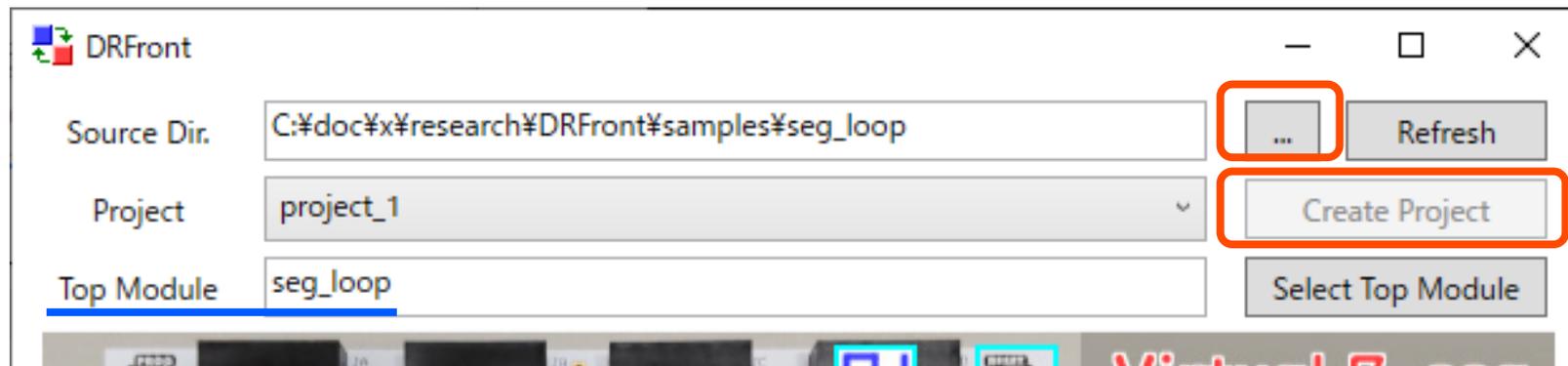
ここにペースト

```
endmodule
```



入出力の割当て (1)

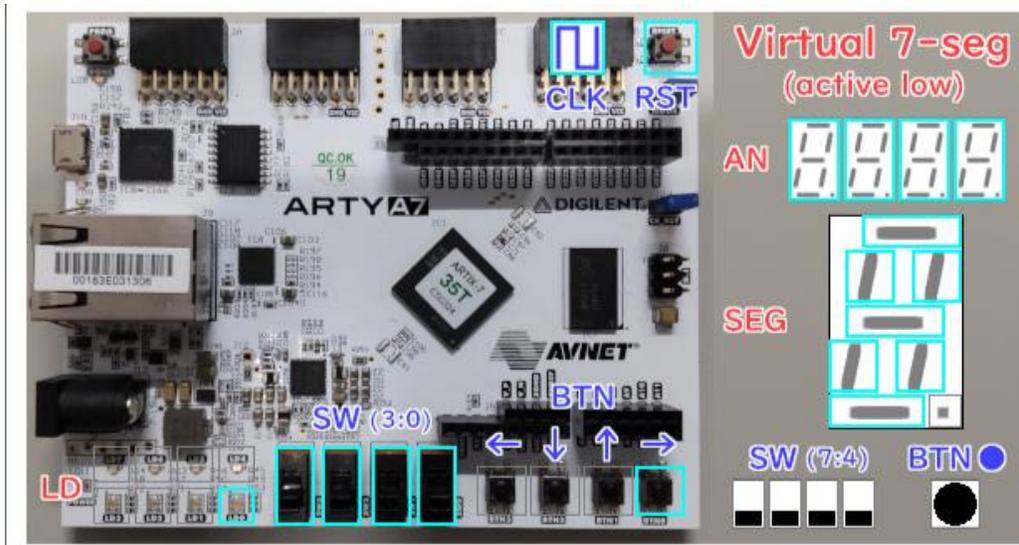
- ◆ Source Dir. 横の ... をクリック
 - 作業フォルダを選択
 - 回路記述に問題がなければ, 先ほど作成した回路が Top Module として認識されているはず
- ◆ Create Project でプロジェクトを作成



入出力の割当て (2)

◆ 右表の通りに, 入出力を ボードの各部品に割当て

- プルダウン選択 or マウสดラッグで
- ボードによっては, ボード上にない部品を仮想的に利用可能



Name	Direction	Assign to
AN[0]	Output	AN[0]
AN[1]	Output	AN[1]
AN[2]	Output	AN[2]
AN[3]	Output	AN[3]
CLK	Input	CLK
LED	Output	LD[0]
REV	Input	BTNR
RST	Input	RST
SEG[0]	Output	CA
SEG[1]	Output	CB
SEG[2]	Output	CC
SEG[3]	Output	CD
SEG[4]	Output	CE
SEG[5]	Output	CF
SEG[6]	Output	CG
SPD[0]	Input	SW[0]
SPD[1]	Input	SW[1]
SPD[2]	Input	SW[2]
SPD[3]	Input	SW[3]

Create/Update Files

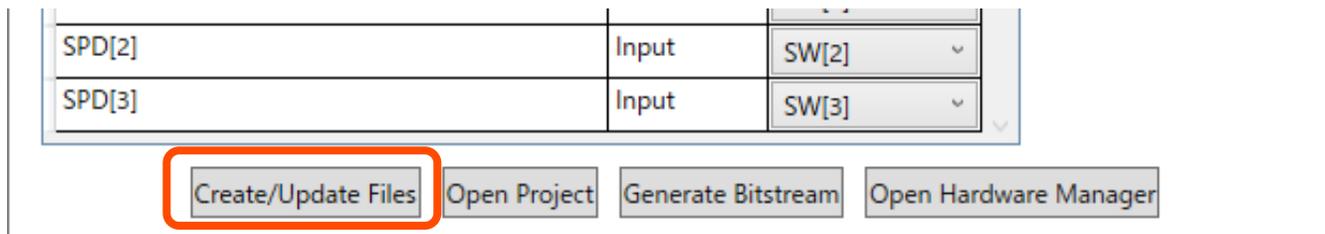
Open Project

Generate Bitstream

Open Hardwar

必要なファイル群を生成

- ◆ **Create/Update Files** をクリック
 - その後, **作業フォルダ**が右下に示す**構成**になっていることを確認
- ◆ WinSCP 等を用いて, **作業フォルダ**をフォルダごと ACRI ルームのサーバに**アップロード**
 - SawareruSys の配布パッケージの **Connector¥server¥connector_serv.py** も
 - WinSCP は SawareruSys の配布パッケージにも同梱



```
seg_loop.sv
project_1
  logs
  dr_top.sv
  dr_testbench.sv
  OpenProject.tcl
  GenerateBitstream.tcl
  OpenHW.tcl
```



SSH で ACRI ルームに接続

◆ PowerShell 等で ACRI ルームに SSH 接続

- ただし, 通常のリモートデスクトップに必要な設定に加えて,
-L 13399:<サーバ名>:3399 も追加する

```
ssh -L 13389:<サーバ名>:3389 -L 13399:<サーバ名>:3399 <ユーザ名>@gw.acri.c.titech.ac.jp
```

- サーバ名は vs + 3桁の数字で, 各自が予約したもの

◆ リモートデスクトップでサーバに接続

- 接続先は localhost:13389

◆ サーバ上で Vivado を起動

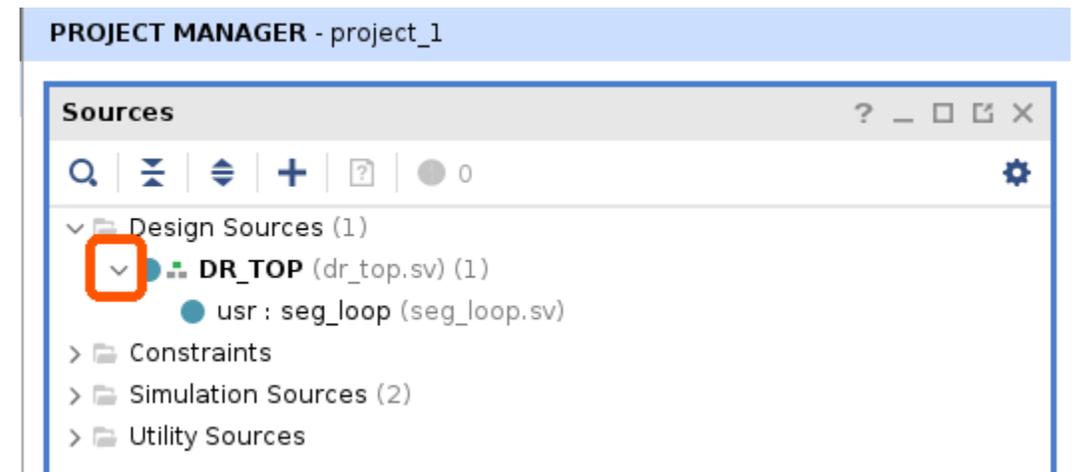
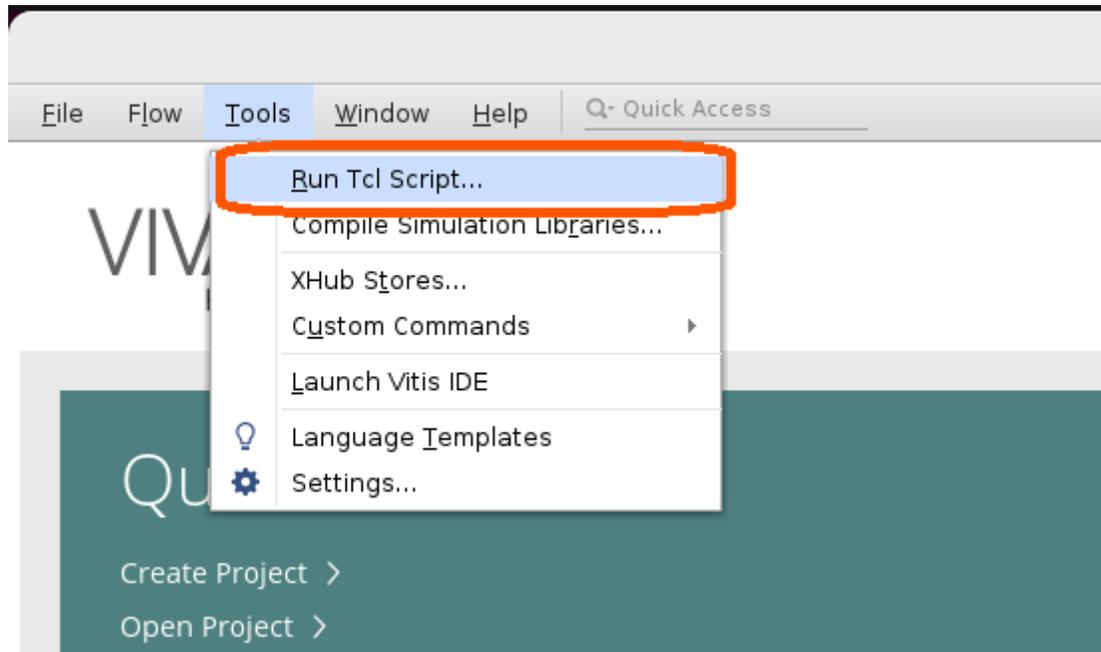
- Ctrl+Alt+T でターミナルを開いてから, 以下のコマンドを入力

```
source /tools/Xilinx/Vivado/2020.2/settings64.sh  
vivado &
```

Vivado で論理合成の準備

リモートデスクトップ上

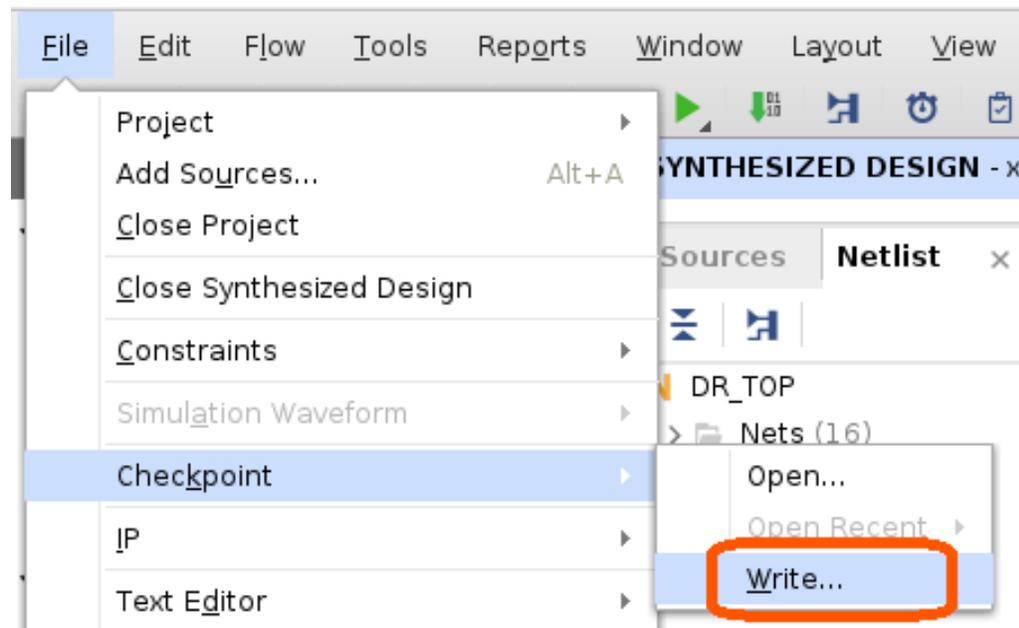
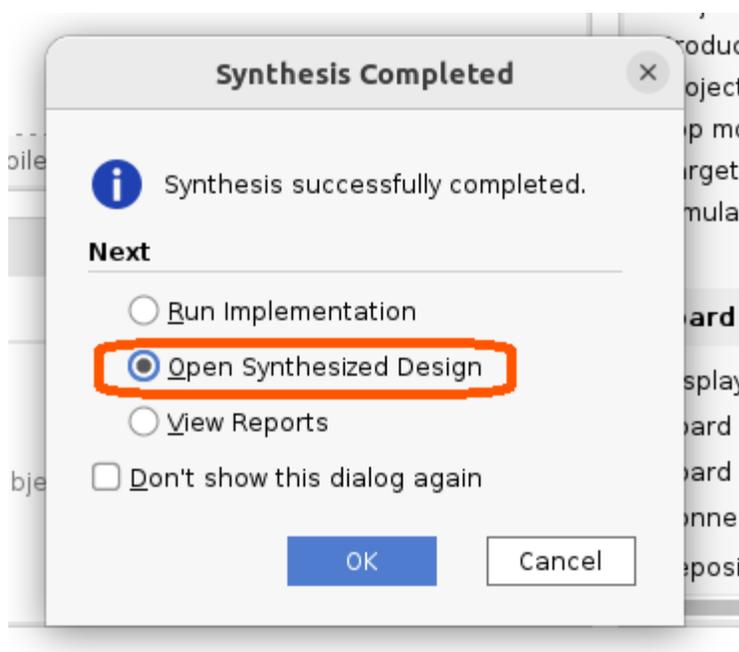
- ◆ Vivado が開いたら **Tools → Run Tcl Script**
 - アップロードした作業フォルダの `project_1/OpenProject.tcl` を選択
 - **Sources** タブ上で, **DR_TOP** の下に作成した回路(`seg_loop`)があることを確認



Vivado で論理合成

リモートデスクトップ上

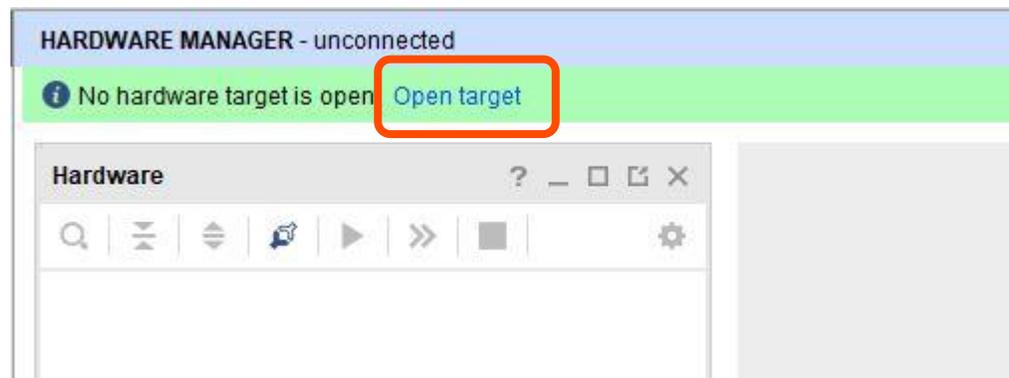
- ◆ SYNTHESIS → Run Synthesis
 - 論理合成に成功したら Open Synthesized Design を選択
- ◆ メニューの File → Checkpoint → Write
 - ファイル名はそのまま OK を押す



Vivado で配置配線～接続

リモートデスクトップ上

- ◆ Tools → Run Tcl Script → GenerateBitstream.tcl
 - 配置配線・ビットストリーム生成が自動で行われ, 初期画面に戻る(約2分)
- ◆ Tools → Run Tcl Script → OpenHW.tcl
 - Hardware Manager が開く
 - Open target → Auto Connect でボードと接続



接続中継の準備(サーバ側)

リモートデスクトップ上

- ◆ リモートデスクトップのターミナル上で, `connector_serv.py` をアップロードしたフォルダに移動
- ◆ 以下のコマンドを入力
 - ログに Serial port opened と表示されれば OK

```
python3 connector_serv.py /dev/ttyUSB1 3399
```

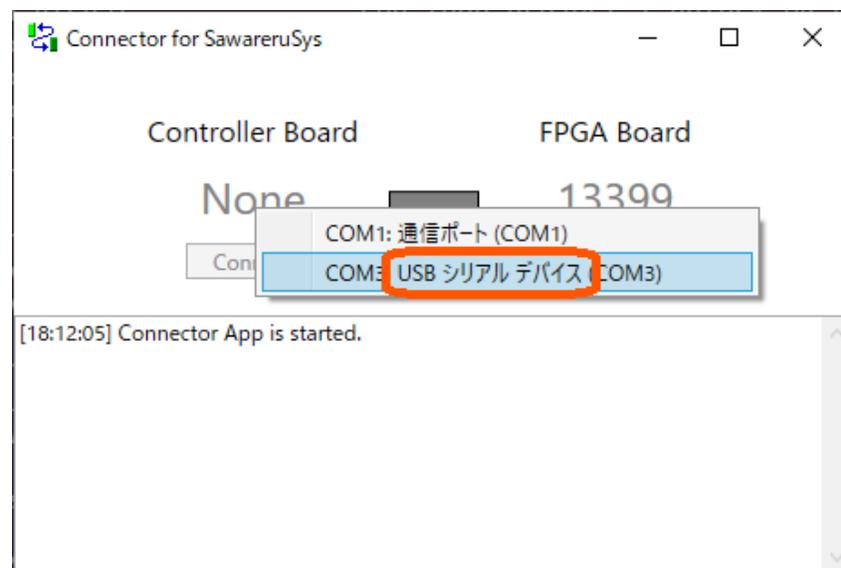
```
seg_loop$ python3 connector_serv.py /dev/ttyUSB1 3399  
[2024-05-29 18:09:45] Connector for SawareruSys started.  
[2024-05-29 18:09:45] Server started listening.  
[2024-05-29 18:09:45] Serial port opened.
```



接続中継の準備(PC側)

各自のPC上

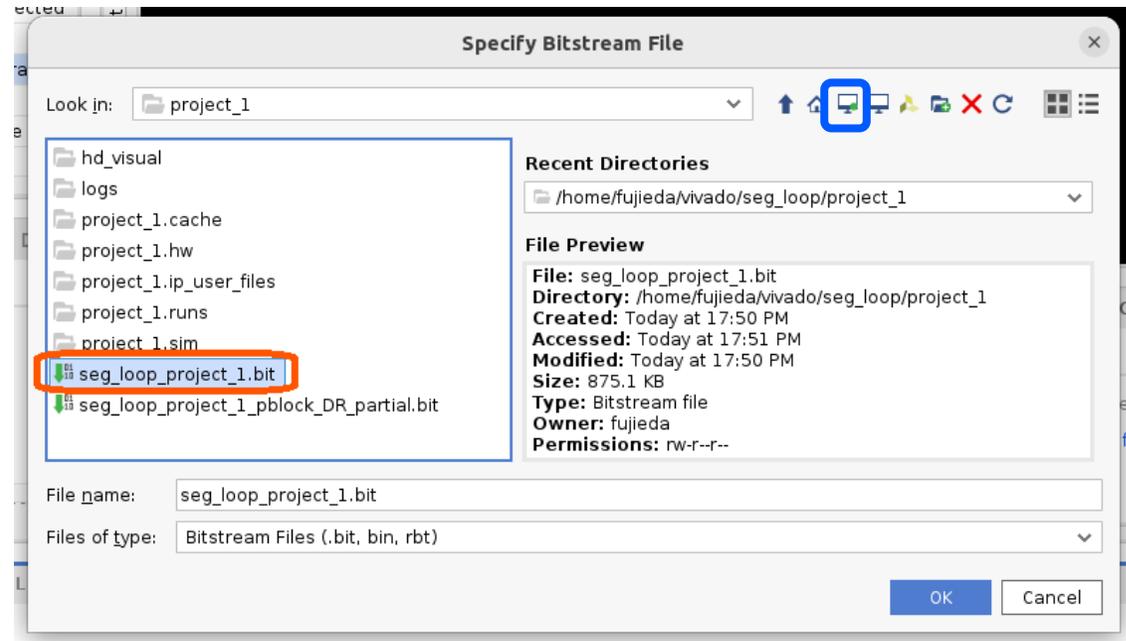
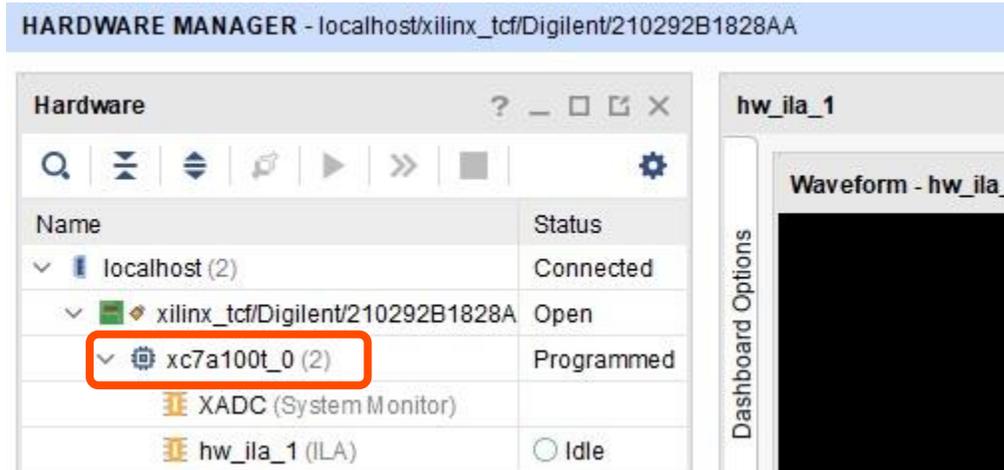
- ◆ コントローラボードを PC と接続
- ◆ コネクタアプリを開く
 - Controller Board の **None** の欄をクリック → **USB シリアル デバイス** を選択
 - 2つの **Connect** ボタンを両方クリック
 - 中央のグレーのバーが青になれば, 接続中継の準備 OK



FPGA に書き込み

リモートデスクトップ上

- ◆ xc7a(35|100)t_0 を右クリック → Program Device
 - 作業フォルダ( で移動)の seg_loop_project_1.bit を選択
- ◆ 回路の動作がコントローラボード上で確認できる(はず)



後始末

◆ PC 側

- コネクタアプリを終了
- コントローラボードの USB ケーブルを抜く

◆ リモートデスクトップ側

- Vivado を終了
- サーバ側のコネクタアプリを実行中のターミナルで Ctrl + C
- デスクトップ右上の電源マークからログアウト

◆ PC 側

- SSH 接続を切断

まとめと展望

◆ SawareruSys の利用体験

- FPGA ボードの入出力に疑似的に「さわれる」
- ACRI ルームともども、ぜひご活用を

◆ 今後の展望

- まだちょっと手順が多いので、何とかしたい.....
- コントローラボード単体での活用法もありそう？
- 実は、簡単な回路であれば FPGA の実機は不要かも？
- などなど(面白そうなアイデアがあればぜひご連絡を)