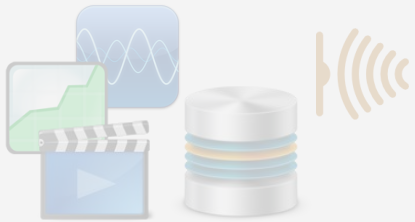
MATLABで楽々FPGA/ASIC実装

MathWorks Japan
アプリケーションエンジニア部

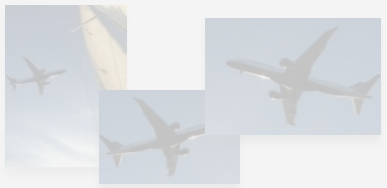
信号処理・AIシステム開発のワークフロー

1. データセットの作成
2. 前処理と変換
3. 予測モデルの開発
4. 実システムへの展開

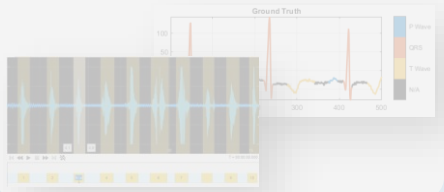
■ データソース



■ シミュレーション
■ オーグメンテーション

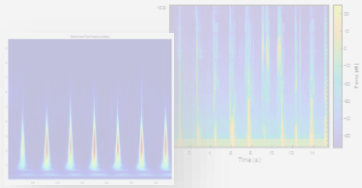


■ データラベリング



■ 前処理

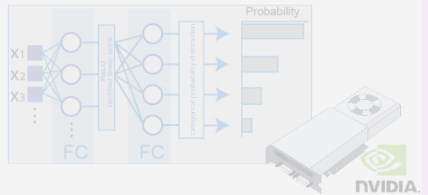
■ 変換



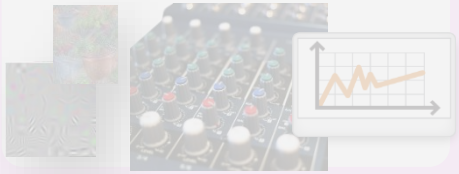
■ 特徴抽出

■ リファレンスモデルの流用
■ オリジナルモデルの作成

■ GPUによる高速な学習

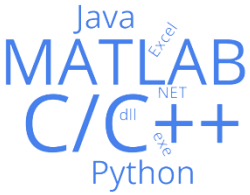


■ ハイパーパラメータの
解析とチューニング



■ デスクトップアプリ

■ エンタープライズシステム



■ エッジ（組み込み）デバイス

アルゴリズムが動くことは分かったけど…

FPGAへの実装はハードルが高い…

開発用PC

準備のハードル
・ハードウェアのアクセスが大変

実装のハードル
・ハードウェアやインターフェイス周りの知識が必要
・ハンドコーディング時にバグが混入

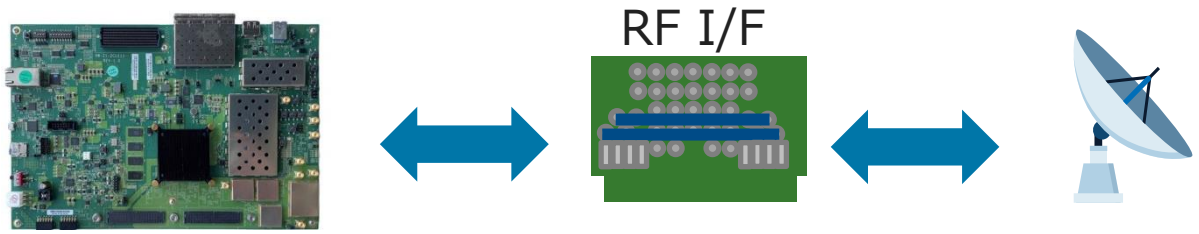
検証のハードル
・アルゴリズム開発環境と検証環境／言語が異なり、性能検証するのが難しい

FPGA

MATLAB／Simulinkを使うことでこれらのハードルを大きく下げ、“本質”に集中してFPGAを開発しましょう！

デモ：ドップラーレーダの信号処理をFPGAへ実装

- ゴール：
  - ドップラーレーダの信号処理アルゴリズムをFPGA（Xilinx Zynq®）へ実装



- ハードル：既存の開発方法

準備 ハードウェアへのアクセス

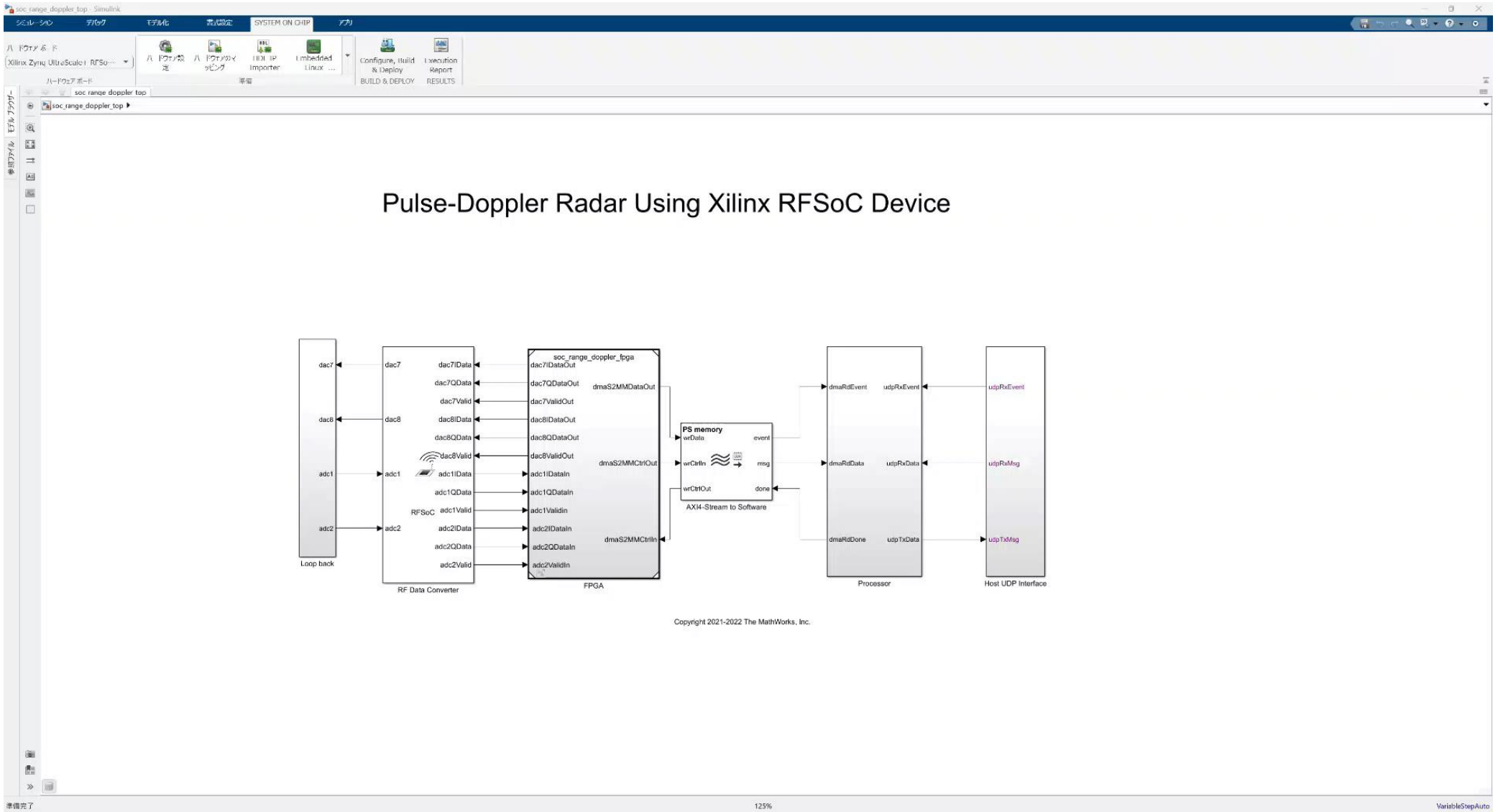
実装 ハンドコーディング

検証 開発環境と検証環境の相違

- Simulinkを活用する方法
  - サポートパッケージでXilinx Zynq開発環境構築
  - 開発したアルゴリズムと等価なHDLコードを自動生成
  - 同一環境上で性能検証

アルゴリズム開発という“本質”に集中することが出来る

デモ：ドップラーレーダシステム



Agenda

準備

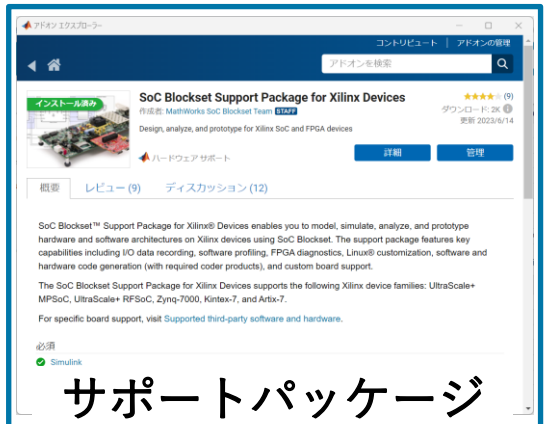
実装

検証

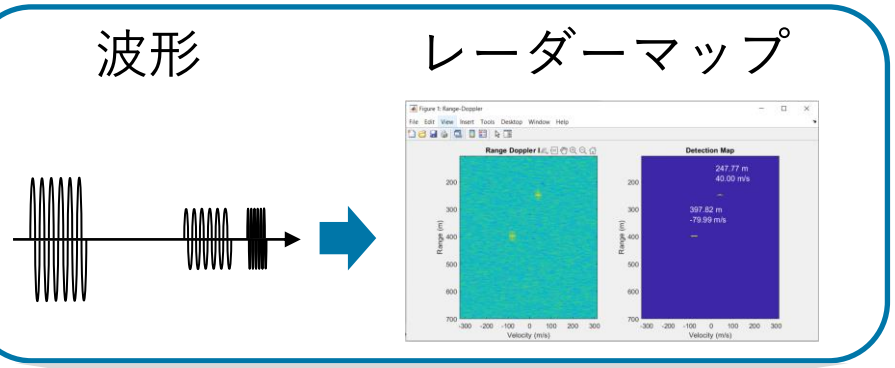
– サポートパッケージでXilinx Zynq開発環境構築

サポートパッケージでXilinx Zynq開発環境構築

お困りごと：ハードウェアへのアクセスが難しい・・・➡サポートパッケージで環境構築支援

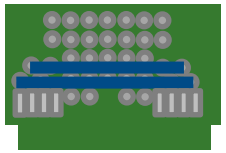


サポートパッケージ



開発用PC

Xilinx ZCU111



XM500 balun card

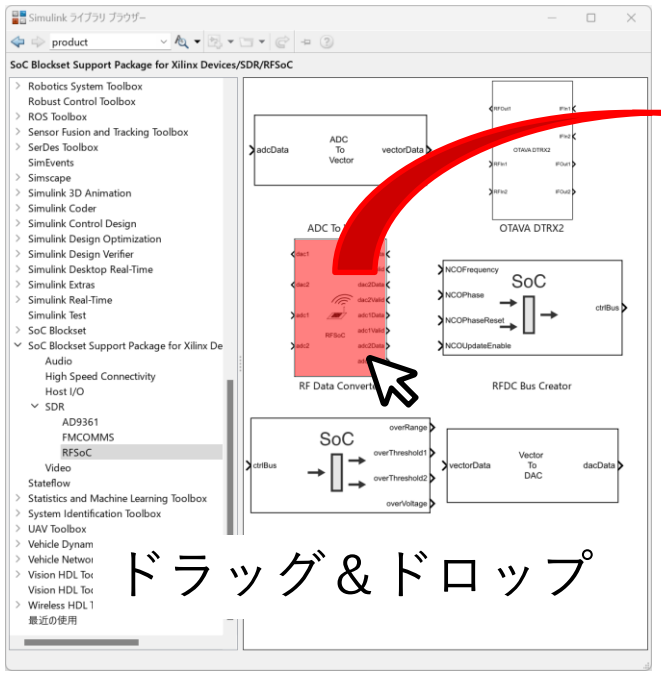
アンテナ

サポートしている周辺ハードウェア（ボード毎に異なる）
　RF Data Converter(XM500 balun card), FMCOMMS, HDMIポート 等

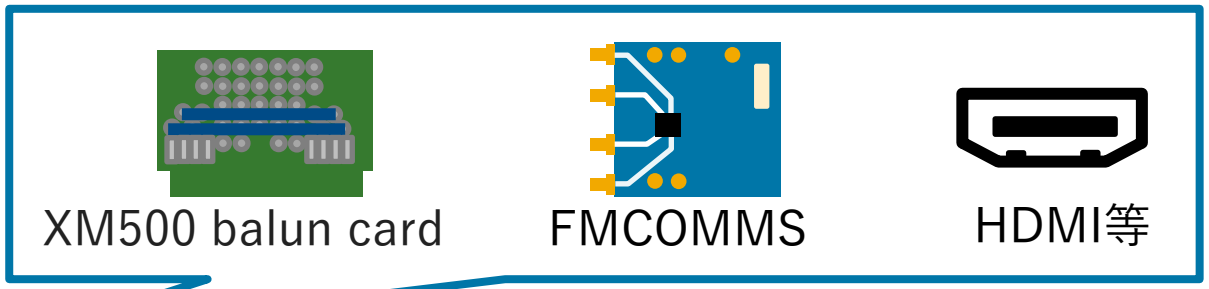
Simulinkとの連携に対応しているボード・ベンダ
　Xilinx®、Intel®、Microchip®

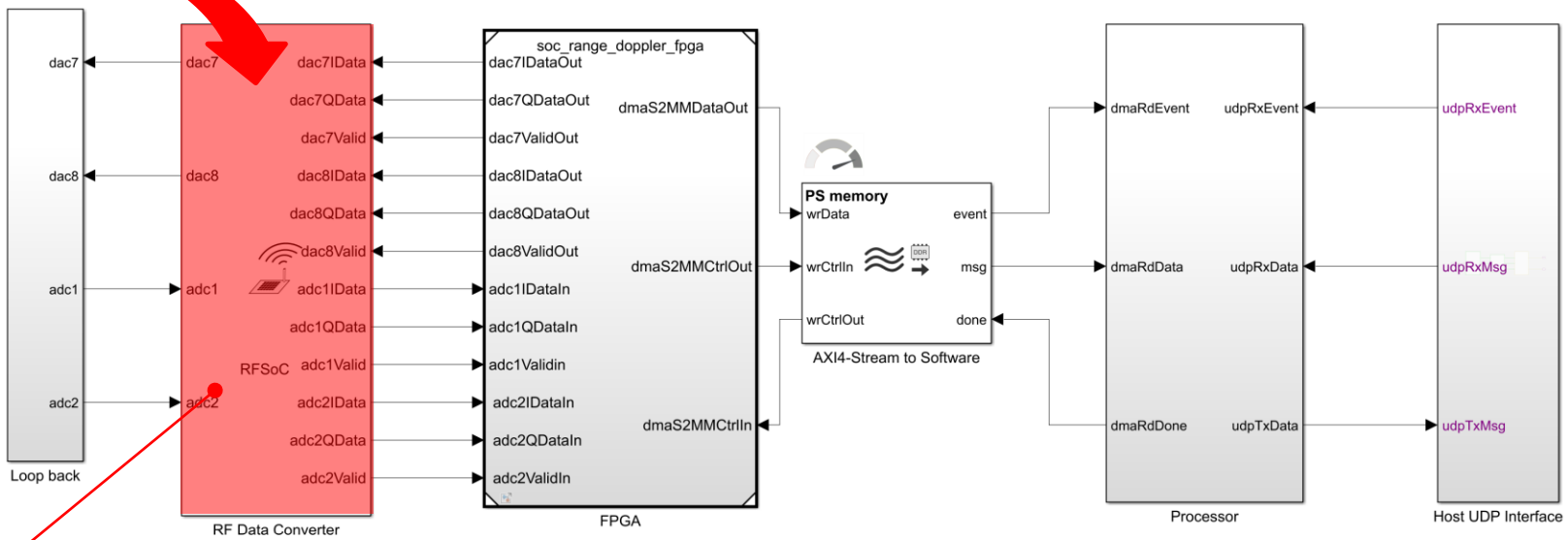
低レイヤを意識せずにペリフェラル周りにアクセスし、素早くラピットプロトタイピングに移れる

サポートパッケージを活用したハードウェアへのアクセス



ドラッグ＆ドロップ





RFドーターボードへ
のI/F用IPコアを生成

ブロック線図ベースでハードウェアへアクセス出来、SoC FPGAのシステムの設計が出来る

Agenda

準備

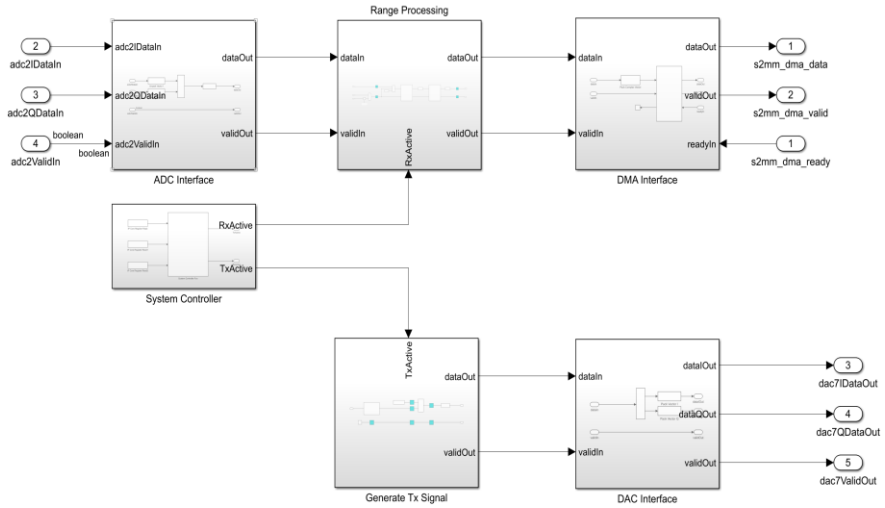
実装

検証

- サポートパッケージでXilinx Zynq開発環境構築
- 開発したアルゴリズムと等価な
  HDLコードを自動生成

ブロック線図ベースの設計によりアルゴリズム開発にフォーカス

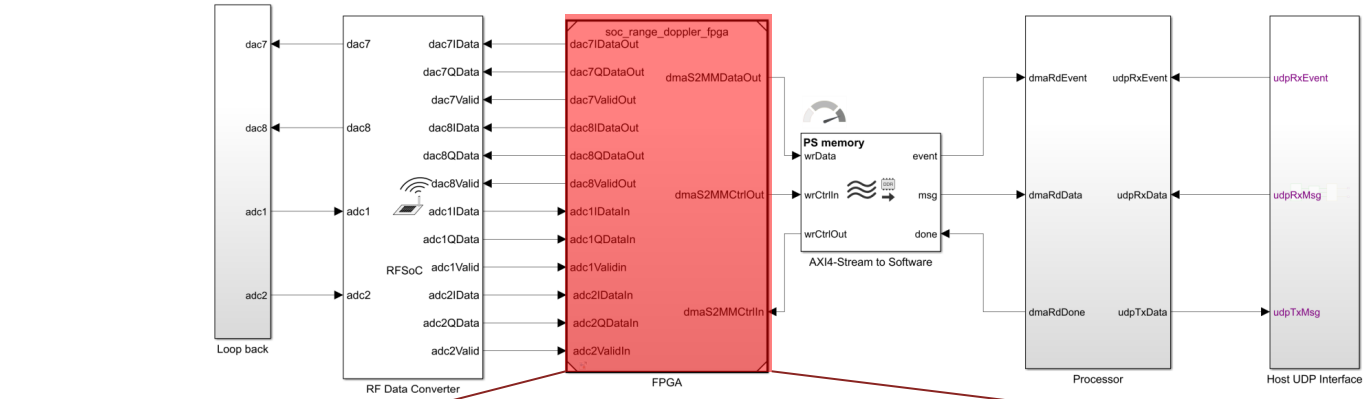
```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY range_doppler_tx_rx_ip IS
  PORT( IPCORE_CLK                    :   IN    std_logic;  -- ufix1
        IPCORE_RESETN                 :   IN    std_logic;  -- ufix1
        AXI4_Stream_0_Master_TREADY   :   IN    std_logic;  -- ufix1
        adc2IDataIn                   :   IN    std_logic_vector(31 DOWNTO 0);  -- ufix32
        adc2QDataIn                   :   IN    std_logic_vector(31 DOWNTO 0);  -- ufix32
        adc2ValidIn                   :   IN    std_logic;  -- ufix1
        AXI4_Lite_ACLK                :   IN    std_logic;  -- ufix1
        AXI4_Lite_ARESETN             :   IN    std_logic;  -- ufix1
        AXI4_Lite_AWADDR              :   IN    std_logic_vector(15 DOWNTO 0);  -- ufix16
        AXI4_Lite_AWVALID             :   IN    std_logic;  -- ufix1
        AXI4_Lite_WDATA               :   IN    std_logic_vector(31 DOWNTO 0);  -- ufix32
        AXI4_Lite_WSTRB               :   IN    std_logic_vector(3 DOWNTO 0);  -- ufix4
        AXI4_Lite_WVALID              :   IN    std_logic;  -- ufix1
        AXI4_Lite_BREADY              :   IN    std_logic;  -- ufix1
        AXI4_Lite_ARADDR              :   IN    std_logic_vector(15 DOWNTO 0);  -- ufix16
        AXI4_Lite_ARVALID             :   IN    std_logic;  -- ufix1
        AXI4_Lite_RREADY              :   IN    std_logic;  -- ufix1
        AXI4_Stream_0_Master_TDATA    :   OUT   std_logic_vector(63 DOWNTO 0);  -- ufix64
```

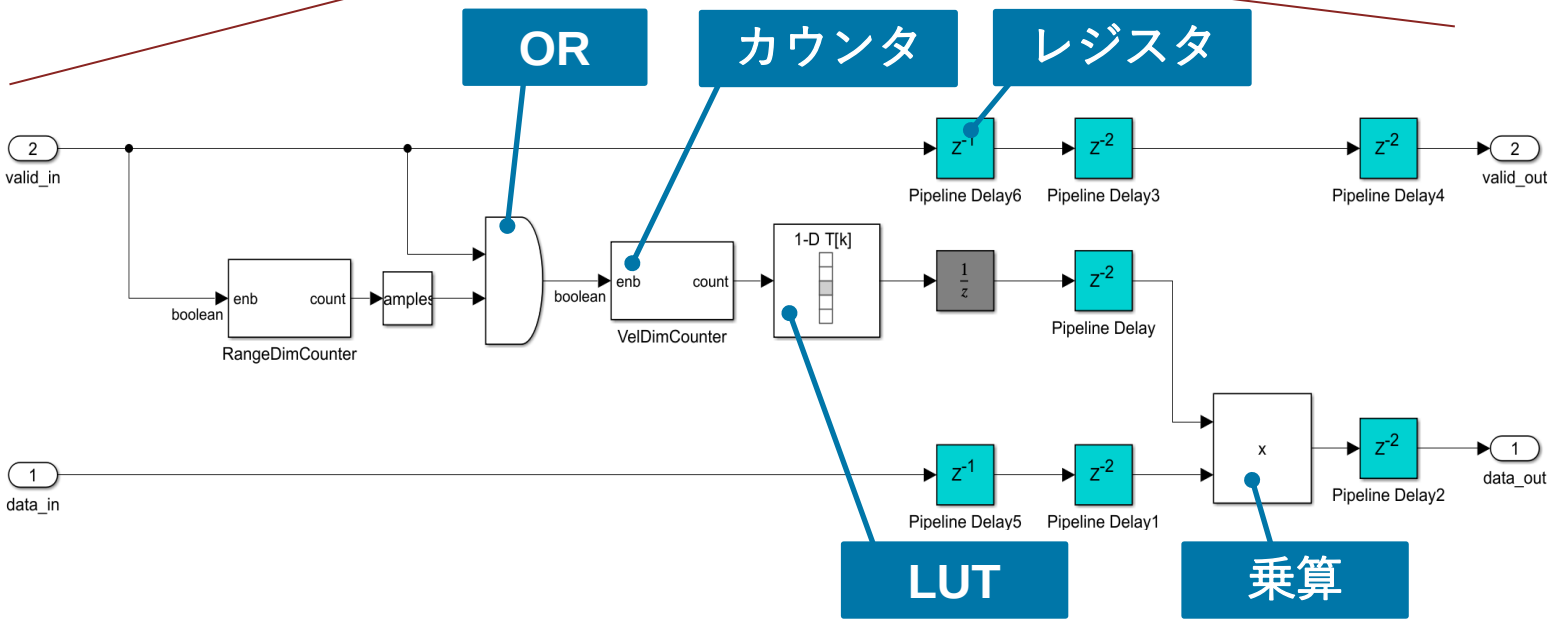


- ハンドコーディングでRTLの設計
  - HDLの習得が必要
  - ロジックIPの接続関係が分かりにくい
  - コーディング時にバグ混入

- ブロック線図を活用した設計
  - ノーコード、ブロック線図ベースでの設計
  - ロジックの接続関係の見通しが良い
  - 等価なコード生成でバグ混入未然防止

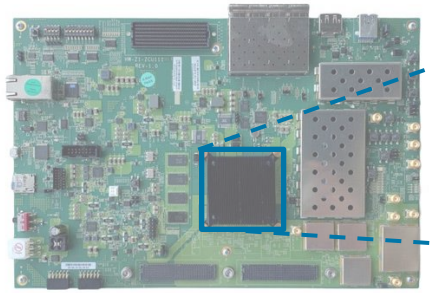
具体的なSimulink上でのFPGAロジック設計方法

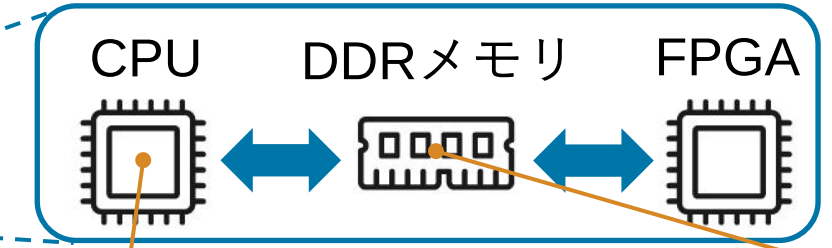




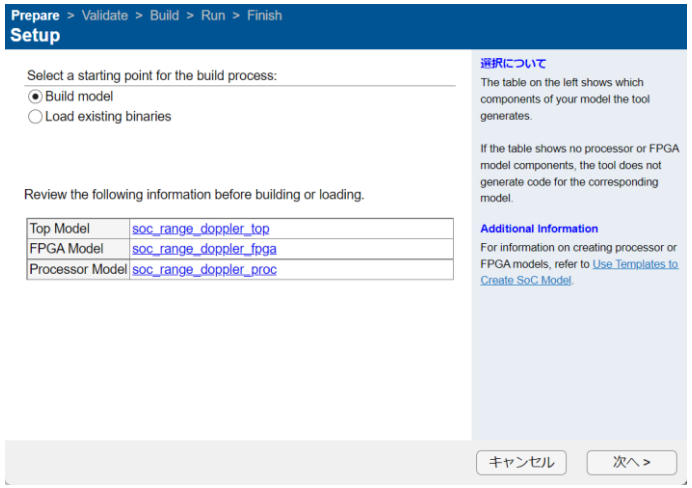
RTLをブロック線図ベースで設計

開発したアルゴリズムと等価なHDLコードを自動生成（１／２）





HDLコード生成対象選択

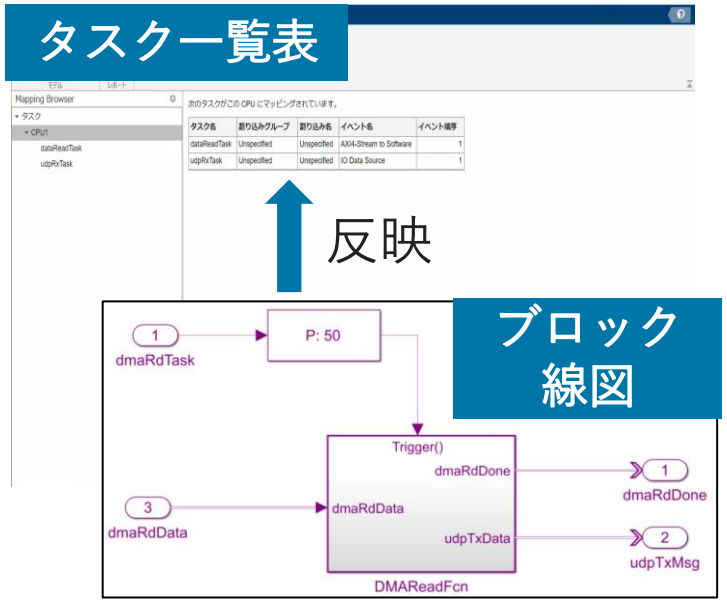


①起動方法

>> socBuilder('modelname')
もしくは

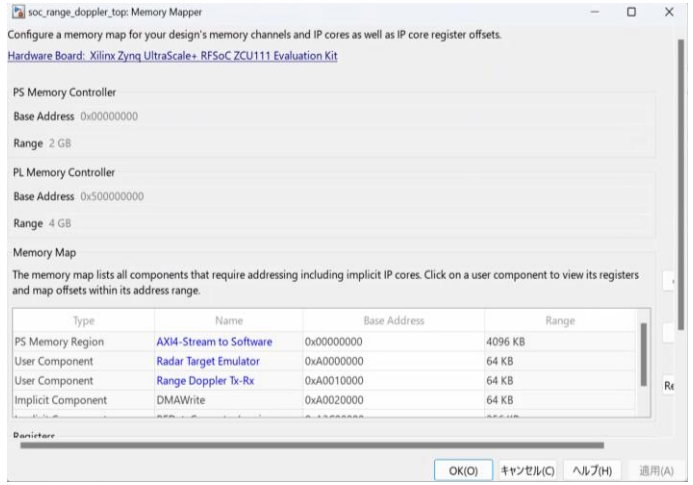
②FPGAに実装するモデルを選択

CPUでのタスクマッピング



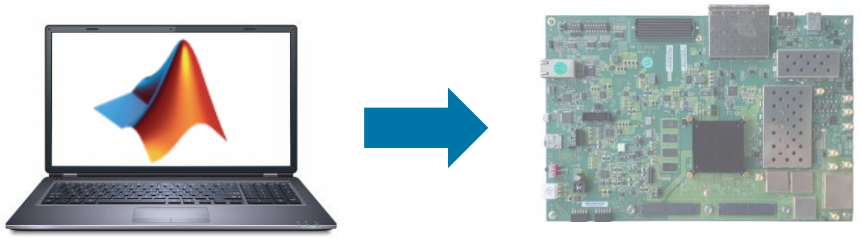
③CPUで実行するタスクと、
タスクを駆動する割り込みを確認

メモリアドレスマッピング

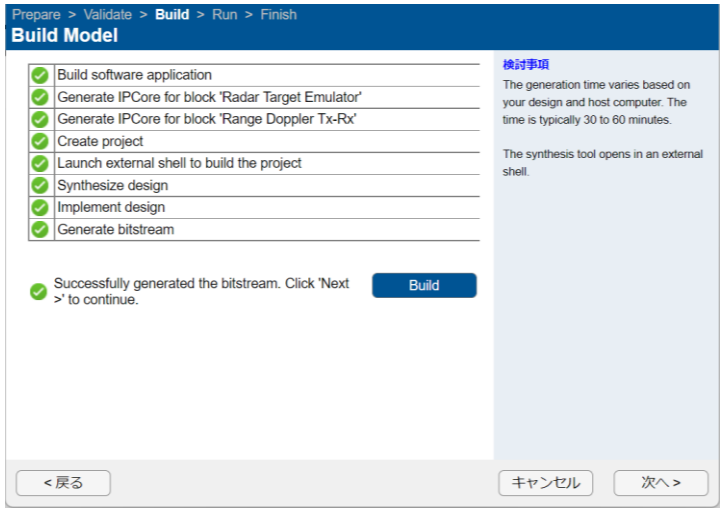


④メモリアドレス、範囲を指定

開発したアルゴリズムと等価なHDLコードを自動生成（２／２）

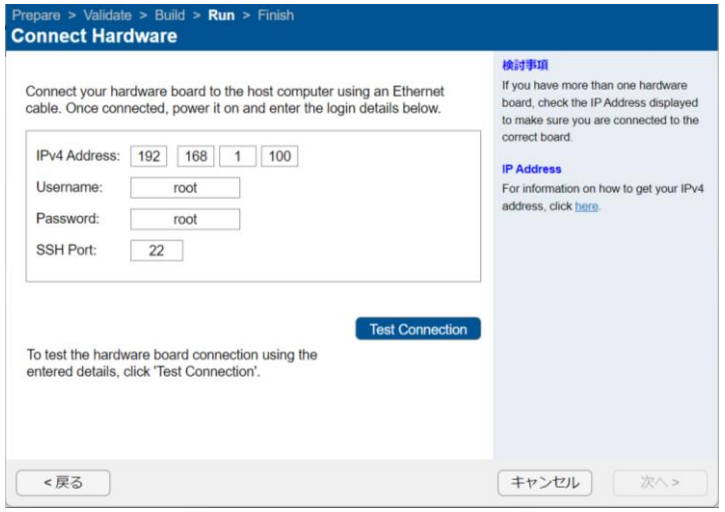


コード生成



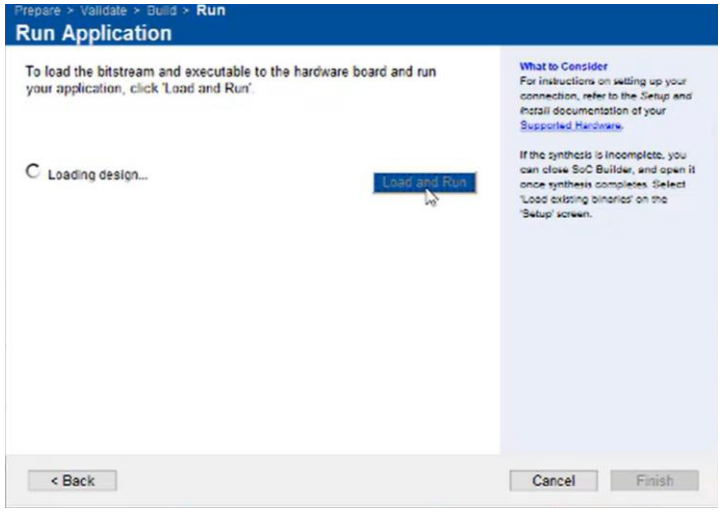
⑤コード生成、ビットストリームの生成

実装先選択



⑥実装先のFPGAのアドレスを選択・通信確認

システムの書き込みと実行



⑦ソフトウェア・ビットストリームの書き込み

GUIベースのアプリでSoC FPGAシステム生成し、実装可能

Agenda

準備

実装

検証

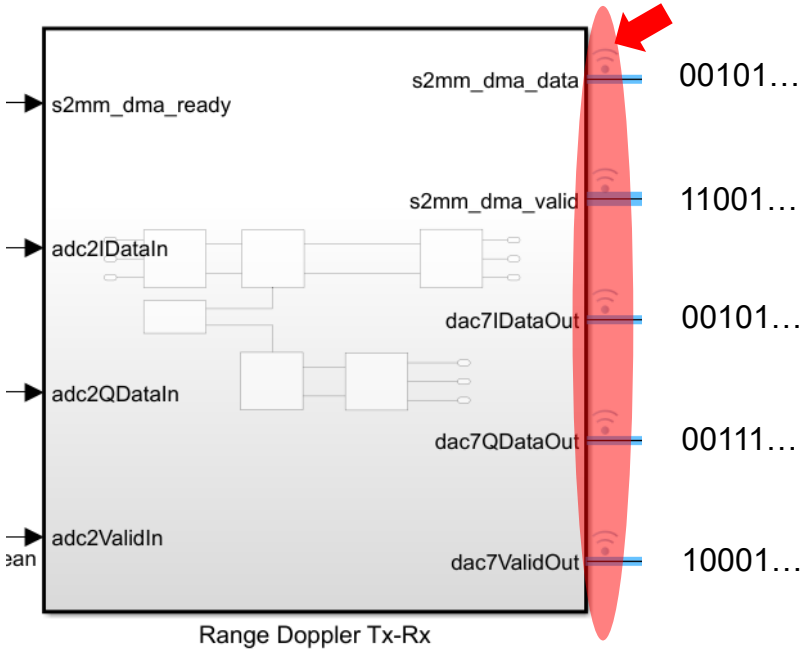
- サポートパッケージでXilinx Zynq開発環境構築
- 開発したアルゴリズムと等価な
  HDLコードを自動生成
- 同一環境上で性能検証

ロジックアナライザを用いたSimulink上からの検証

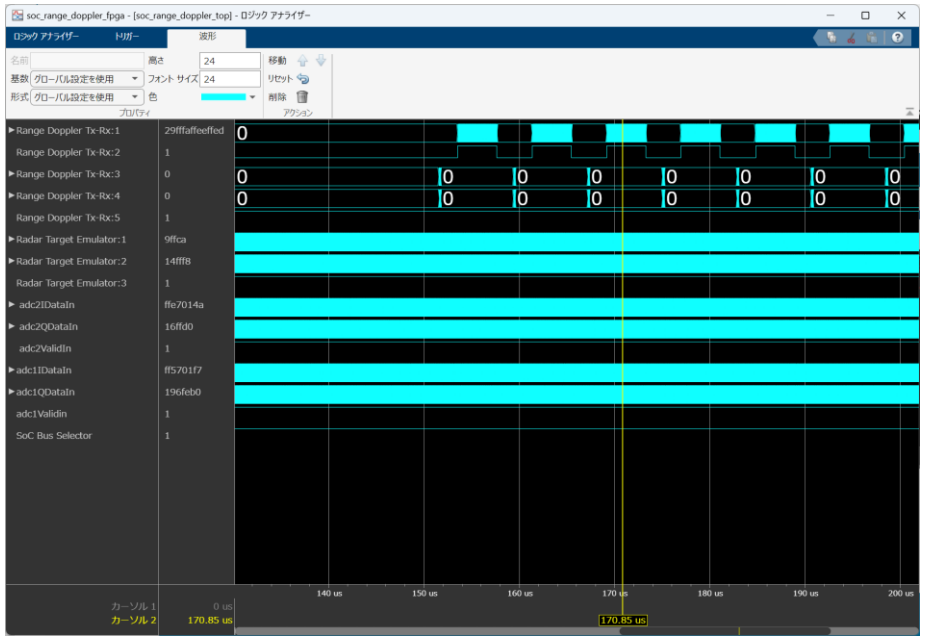
お困りごと：FPGAの性能改善・デバッグのためロジックの動作検証したい

➡ ロジックアナライザを用いて信号線の時系列データを確認

信号の取得箇所選択



信号の値を確認

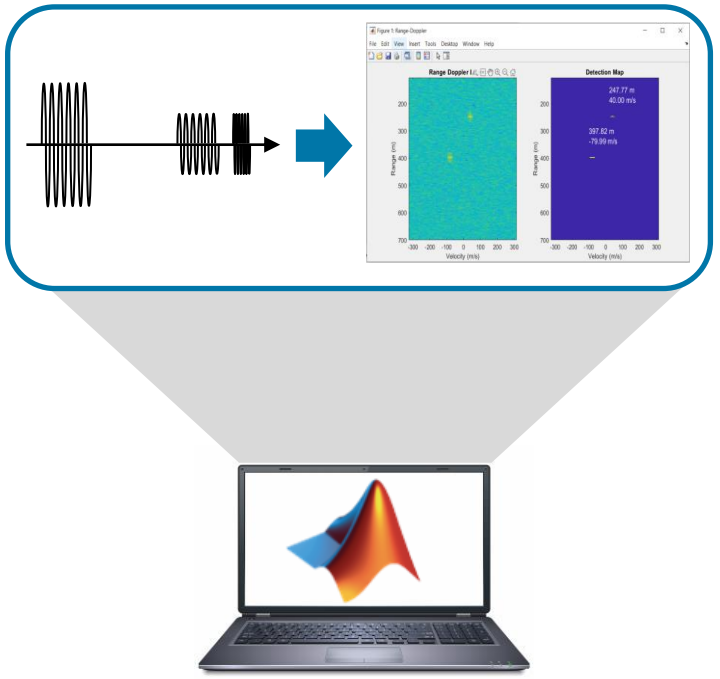


多数の信号の同時トレースによりデバッグ、タイミングの検出が容易に

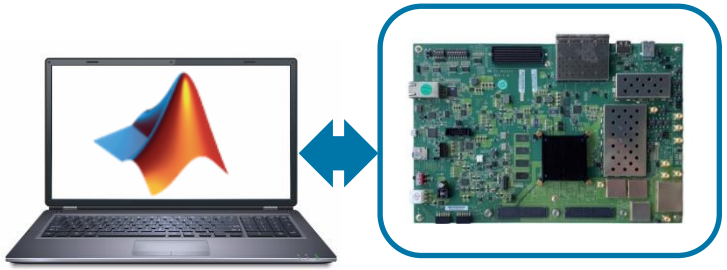
FPGAの動作をロジックアナライザにて検証可能

実機実装の前に効率的な検証が可能

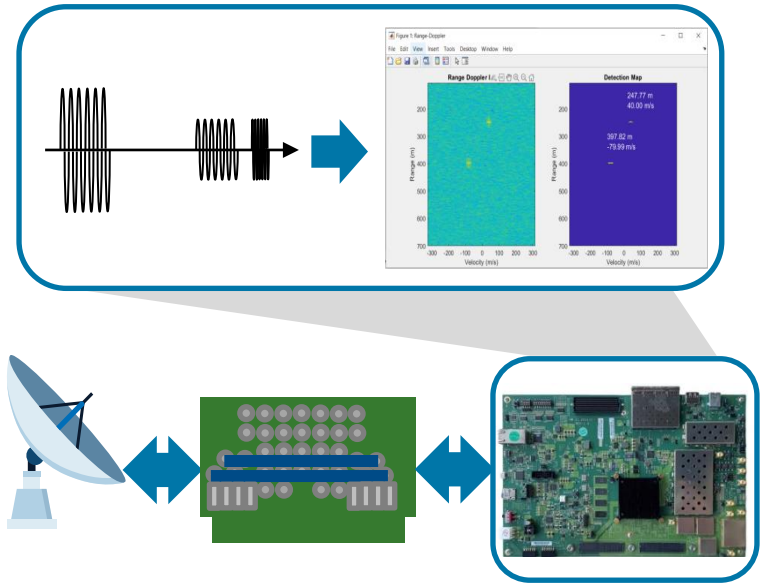
シミュレーション → FPGA-in-the-Loop → External Mode → 実機実装



開発用PC上でコード実行



開発用PC上で動作を確認しつつ実行

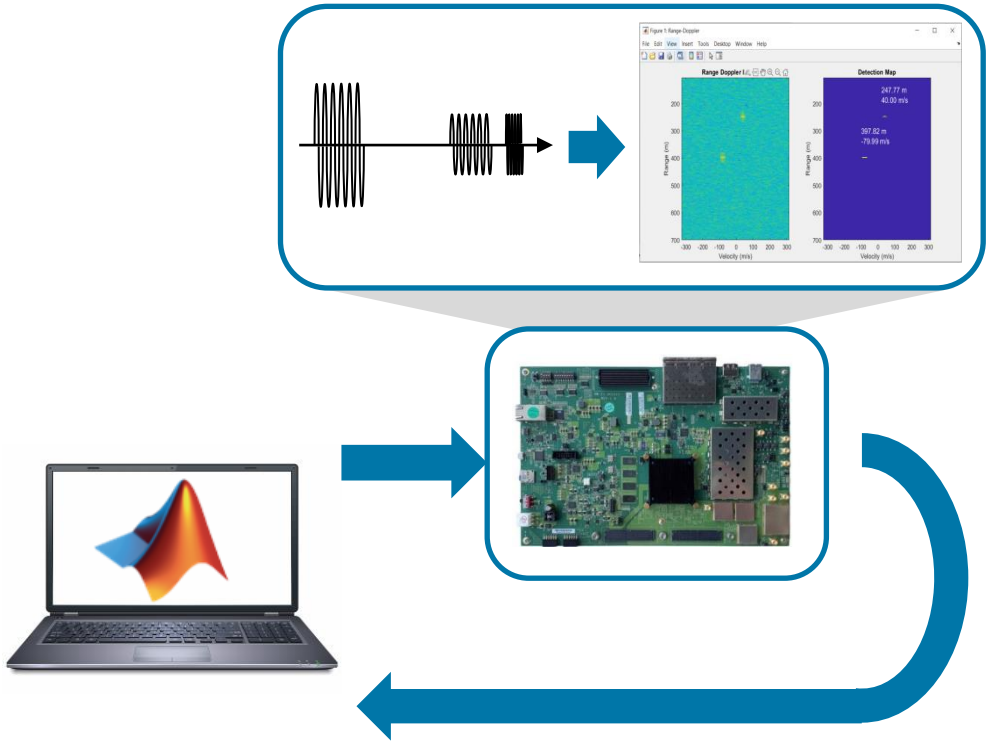


実機上でコード実行

MATLAB/Simulink環境上で効率的に動作検証

実機実装の前に：FPGA-in-the-Loop機能を用いた動作検証

シミュレーション | **FPGA-in-the-Loop** | External Mode | 実機検証

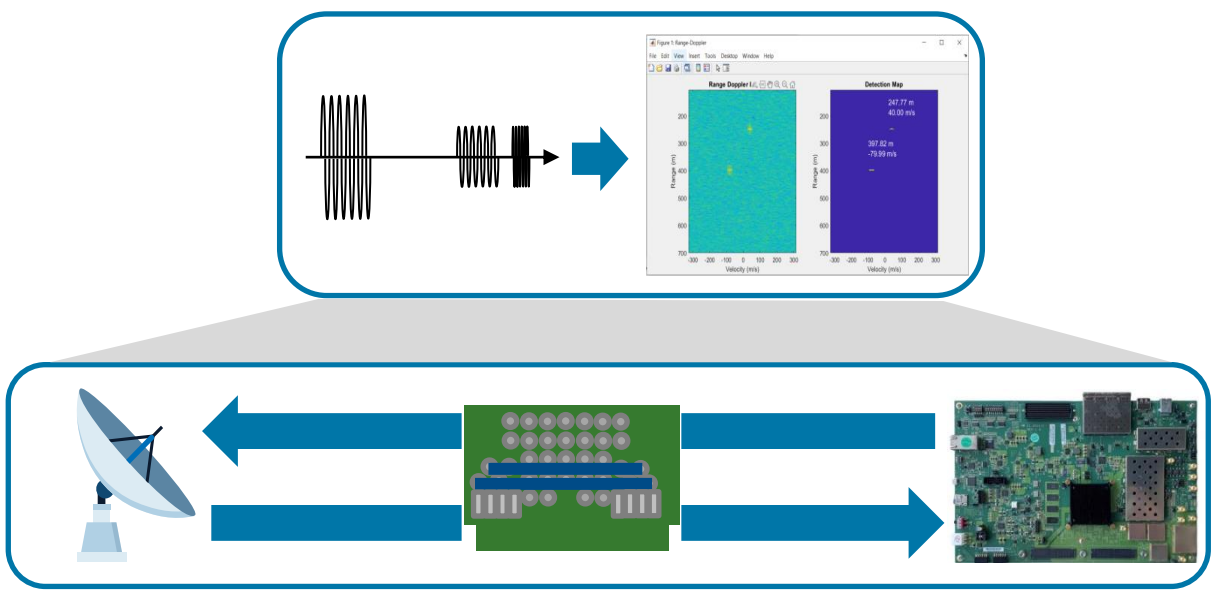


HDLコード化対象部分のみ実行（非リアルタイム）

MATLAB/Simulink環境上で効率的に動作・等価性検証

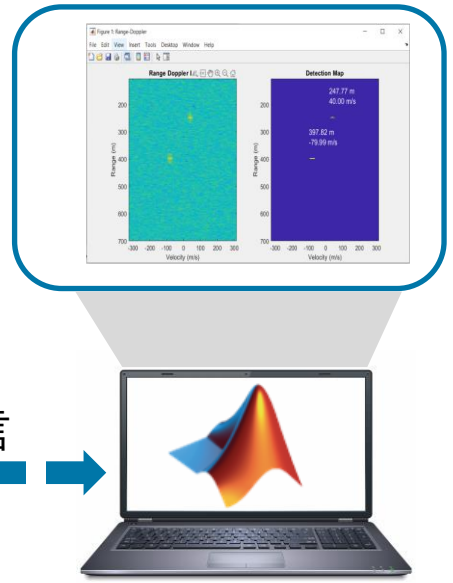
実機実装の前に：External Mode機能を用いた動作検証

シミュレーション → FPGA-in-the-Loop → **External Mode** → 実機検証



ドップラーレーダシステム
（リアルタイム実行）

通信

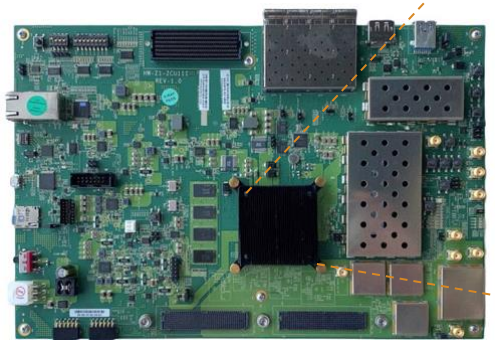


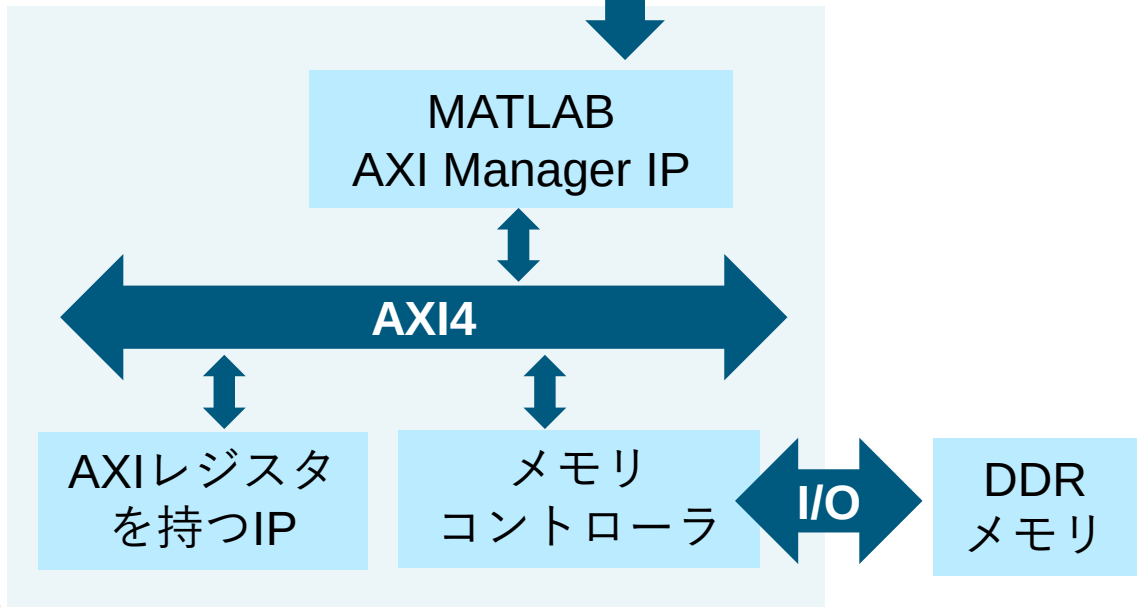
信号データの監視
CPUを介してFPGAのパラメータの書き換え

External Mode機能を活用することで効率的に実機調整

Simulink上からの実機実行時の動作検証

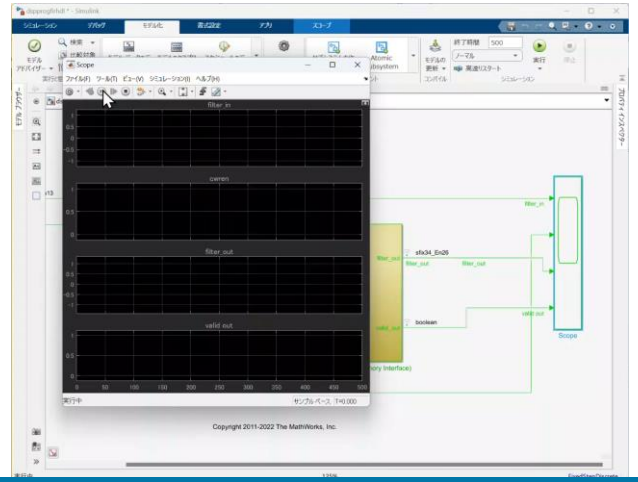
- AXI Managerを使った動作検証
  - MATLAB/Simulink環境から
    オンボードメモリへのアクセス
  - 動作時のメモリ／IPのレジスタの値を
    読み出し・書き込み



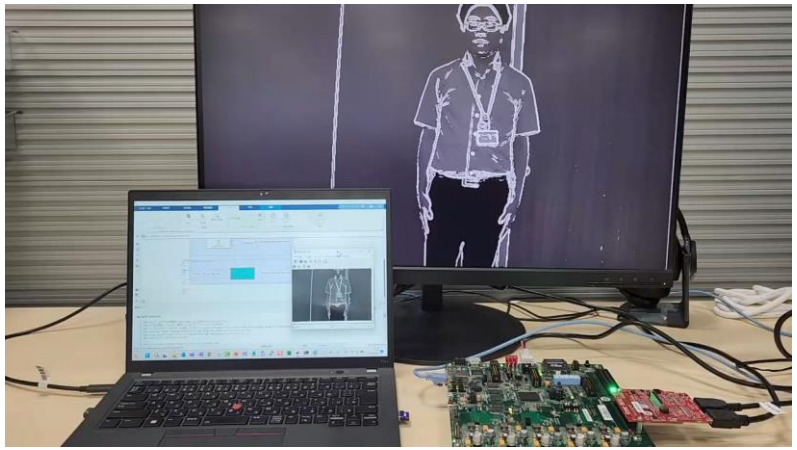


実装後の動作検証も強力にサポート

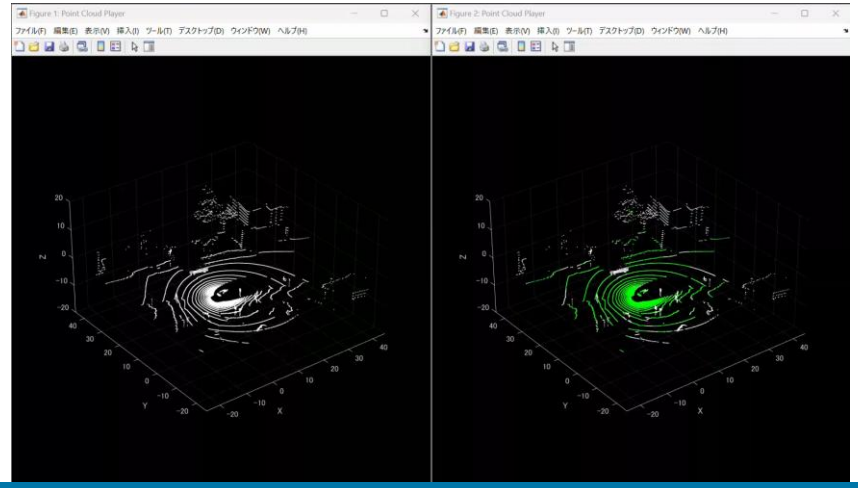
すぐにお試しいただける製品例題



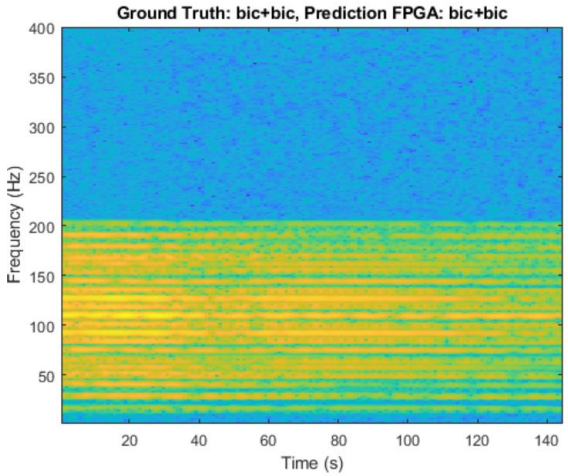
FPGA 用のプログラム可能な FIR フィルター



ビデオカメラ入力のエッジ強調処理

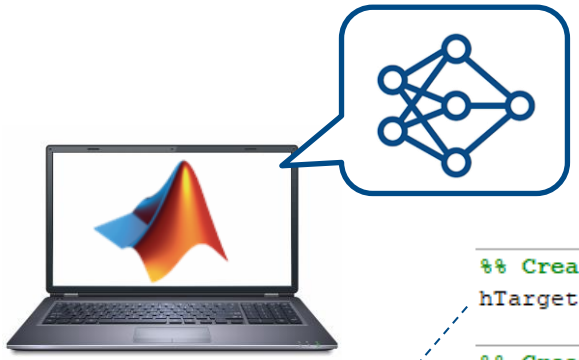


FPGAを用いたLiDARデータからの地面検知



FPGAと深層学習による人・自転車分類

MATLAB活用でディープラーニングのFPGA実装可能

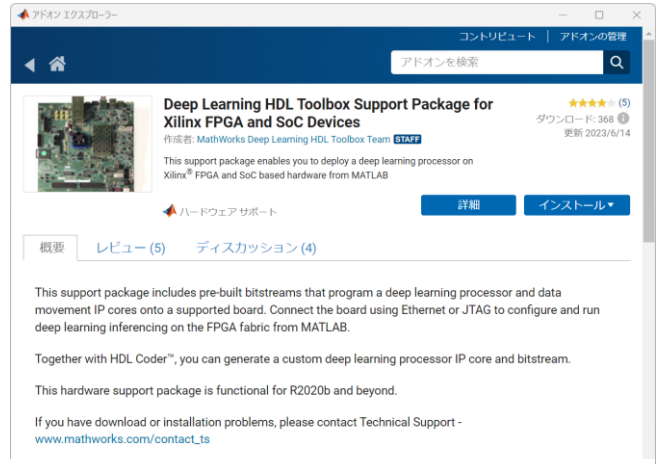


開発用PC

FPGA

```
%% Create Target Object
hTarget = dlhdl.Target('Xilinx','Interface','Ethernet','IPAddress','10.10.10.15');

%% Create WorkFlow Object
hW = dlhdl.Workflow('network', snet, 'Bitstream', 'zcu102_single','Target',hTarget);

%% Compile the Lanenet series Network
dn = hW.compile;

%% Program Bitstream onto FPGA and Download Network Weights
hW.deploy;

%% Run prediction for one frame
outputs = hW.predict(inputImg);
```

MATLAB⇒FPGAへ展開するスクリプト



サポートパッケージ

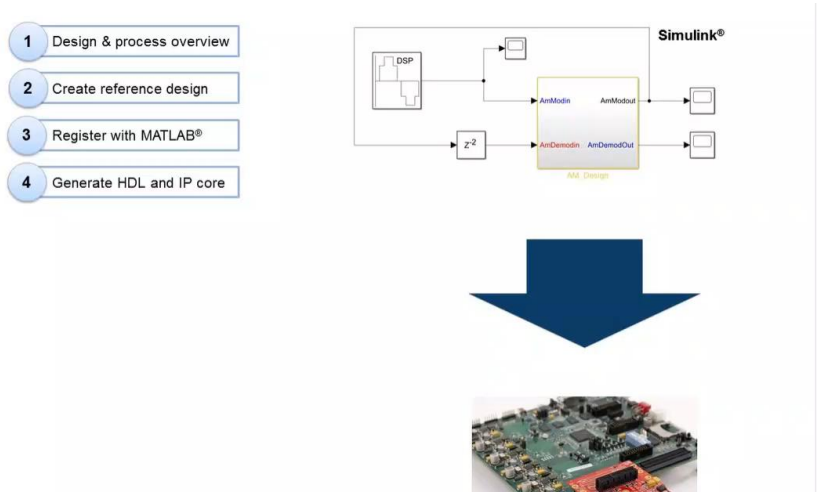
Deep Learning HDL Toolbox Support Package for Xilinx FPGA and SoC Devicesで
サポートしているハードウェア・デバイスファミリ
　　Xilinx Zynq® UltraScale+™ MPSoC ZCU102, Xilinx Zynq-7000 ZC706

サポートしている学習済みネットワーク
　　AlexNet, LogoNet, ResNet-based YOLO v2, SqueezeNet-based YOLO v3等

細かいカスタマイズや最適化も可能

- 想定しているボードがあるんだけど・・・
  - リファレンスデザインをカスタマイズし、Simulinkと連携可能

充実したドキュメント



カスタムボードの手順ビデオ

お客様の想定しているボード

カスタムボード設定可能、お客様が使いたいボードに実装も可能

全体のまとめ

MATLAB® SIMULINK® を活用することで

準備のハードル	・ハードウェアへのアクセスが大変	・サポートパッケージで Xilinx Zynq開発環境構築
実装のハードル	・ハードウェアやインターフェイス周りの知識が必要 ・ハンドコーディング時にバグが混入	・開発したアルゴリズムと等価な HDLコードを自動生成
検証のハードル	・アルゴリズム開発環境と検証環境／言語が異なり、性能検証するのが難しい	・同一環境上で性能検証

FPGA

MATLAB／Simulinkを用いることで“本質”に集中し、ラピットプロトタイピングが可能！

技術トレーニングコース（FPGA）

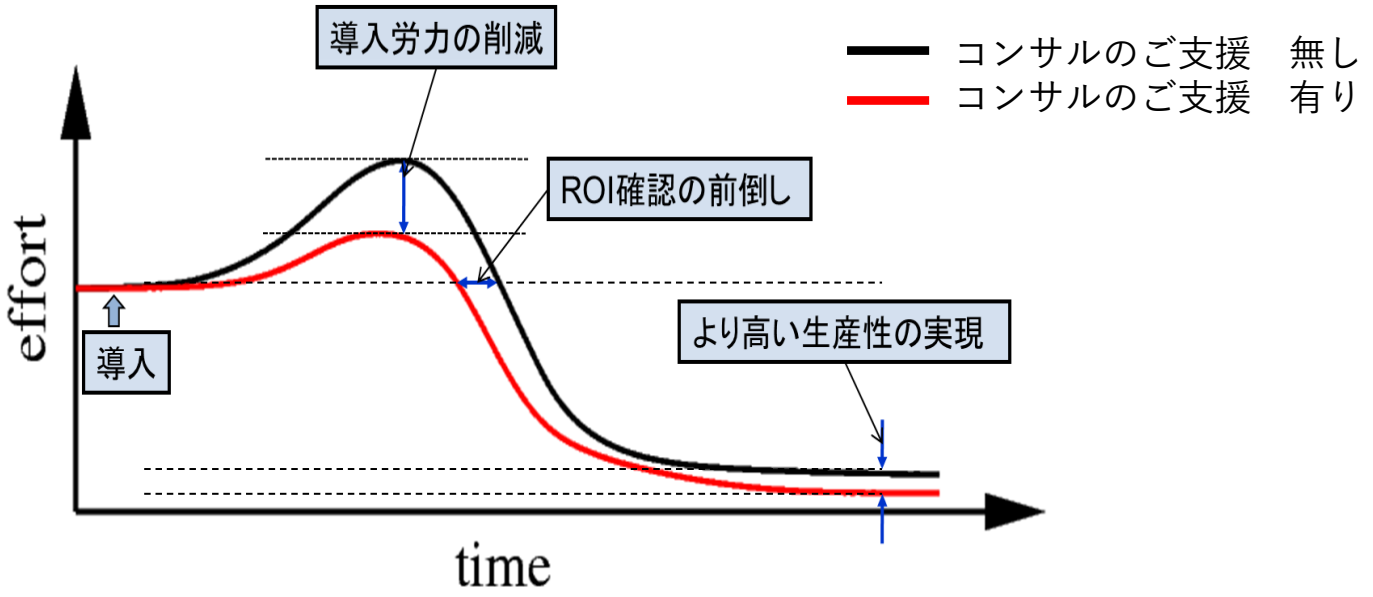
- MathWorks技術トレーニングコース
  - パブリックコース（東京、名古屋、大阪）、オンサイトコース、オンライン
- トレーニングコース受講のメリット
  - 短期間で豊富な機能を効率的に習得
  - 実用的な演習問題により、理解度向上
- 関連コース
  - HDL CoderによるHDLコード生成
  - カスタムトレーニング（オンデマンド）は相談いだければ打ち合わせの上、実施検討

https://jp.mathworks.com/services/training/index.html

FPGA開発に向けた立ち上がりを強力にサポート

MathWorks 技術コンサルティング・サービスのメリット

- 技術コンサルティング・サービスを活用することにより、**より早く、より少ない労力で、より効果的に弊社製品を導入**できます
  - お客様の仕様書、モデルやデータをもとに、ご支援します
  - 他のプロジェクトや業界での経験を活かして、すばやく軌道に乗れるようにご支援します
  - お客様社内にノウハウの蓄積ができます



MathWorks 技術コンサルティング・サービスのプロジェクトのご支援内容

- 技術コンサルティング・サービスは、受託開発ではなく、**アドバイザリー**（助言や顧問的な活動）をご提供します
  - 例えば、全てのプログラムやモデルの開発作業をお客様の代わりに実施することではありません

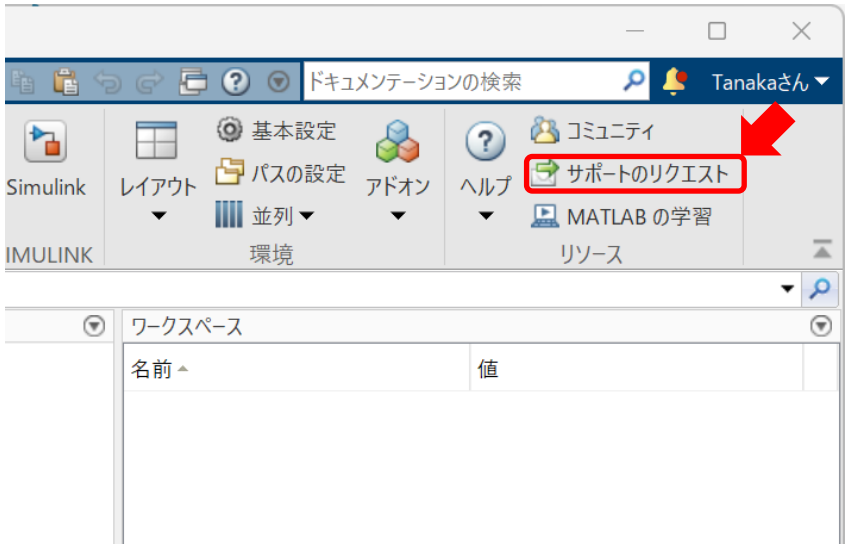
- 技術コンサルタントは、**お客様自身で、弊社製品を効果的に使える**ようになっていただくために、下記のようなご支援を行います
  1. お客様の仕様書、モデルやデータをもとに、実際に開発を行って模範例をご提供します
  2. その中で、サンプル・プログラムやモデル、説明資料をご提供します
  3. 必要に応じて、演習を実施し、お客様にご理解を深めていただきます

サポートの紹介

- MathWorksのサポートエンジニア
  - お客様の環境への導入支援
  - 技術的な成立性確認をサポート

- 詰まりがちなポイントをサポート可能
  - お客様の開発環境構築
  - お客様で解決が難しいエラー
  - 弊社製品についての技術的ご相談

- お問い合わせ方法
  - MATLAB デスクトップの右上にある“サポートのリクエスト” を押下
  - 表示されたダイアログ内に、質問のタイトルと内容（詳細）を記入



専門のエンジニアによる的確なサポート