



MATLAB EXPO
JAPAN

2020年9月28日—10月2日 | オンライン

FPGA回路規模の見積もり精度向上と 実機デバッグの高機能化

横河電子機器株式会社

(2020年10月1日より 株式会社YDKテクノロジーズ)

秦 直哉

アジェンダ

- 会社紹介
- FPGA開発フローの改善 (MBDの導入)
 1. 抽象度の異なるエンジニア間のコミュニケーション改善
 - 回路規模とレイテンシの早期見積り
 - リソース共有ツールによる最適化
 - 浮動小数点HDLコードの利用
 2. 実機デバッグの高機能化と効率化
 3. Webアプリによる実験結果の共有
- マイクロマウス競技大会の紹介

本セッションの関連Toolbox 一覧

- MATLAB®
- Simulink®
- HDL Coder™
- HDL Verifier™
- Fixed-Point Designer™
- DSP System Toolbox™
- Embedded Coder®
- MATLAB® Coder™
- MATLAB® Compiler™
- MATLAB® Web App Server™
- Polyspace® Bug Finder™
- Polyspace® Code Prover™

社名変更

2020年10月1日 横河電子機器 は
YDKテクノロジーズ へ社名を変更します。



事業紹介



 **マイナビ2022** をご覧いただくか、採用担当までご連絡ください。

当社採用担当メールアドレス: jinji@ydktechs.co.jp

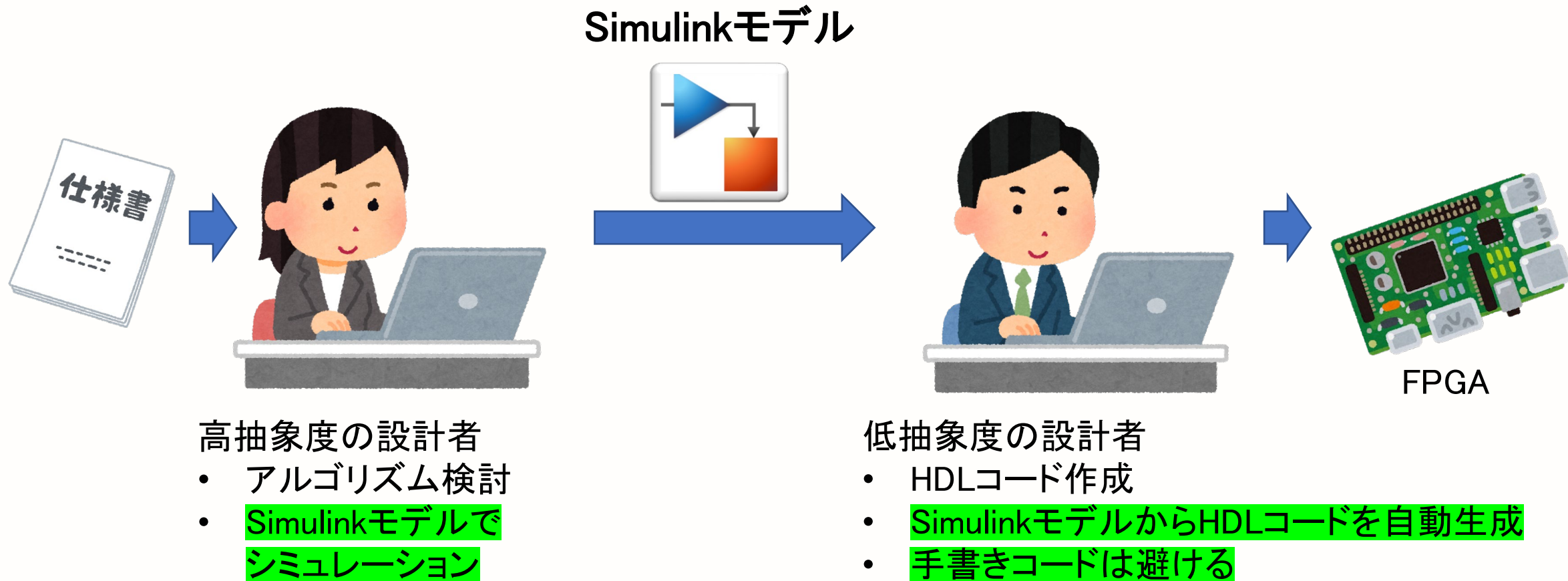
- 新卒採用、キャリア採用、ともにメールアドレスは共通です。
-  で知ったと伝えて頂くとスムーズです。

【冬季インターンシップ】

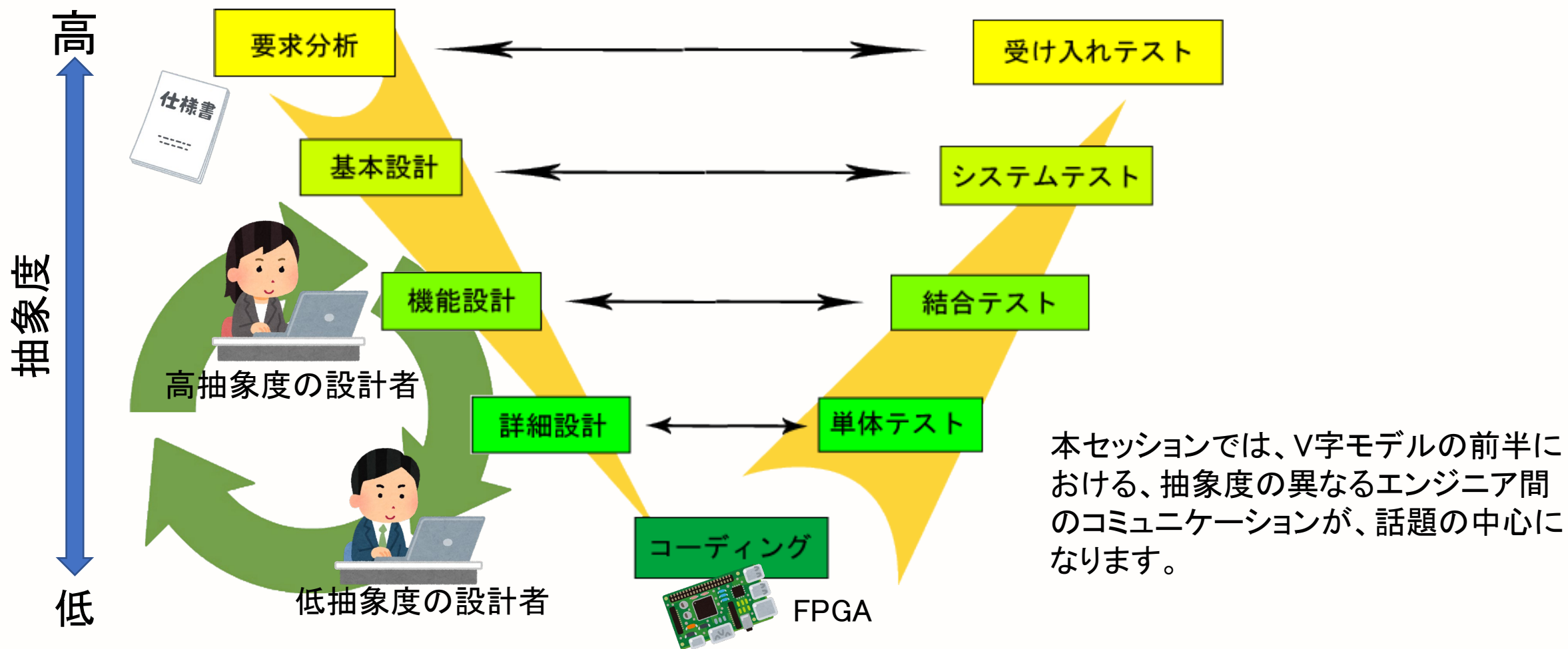
- 時期: 2021年1～2月
- 場所: オンライン開催
- 内容: MATLAB と関連した内容で検討中。(予定)



現在のFPGA開発フロー



V字モデルとの対応

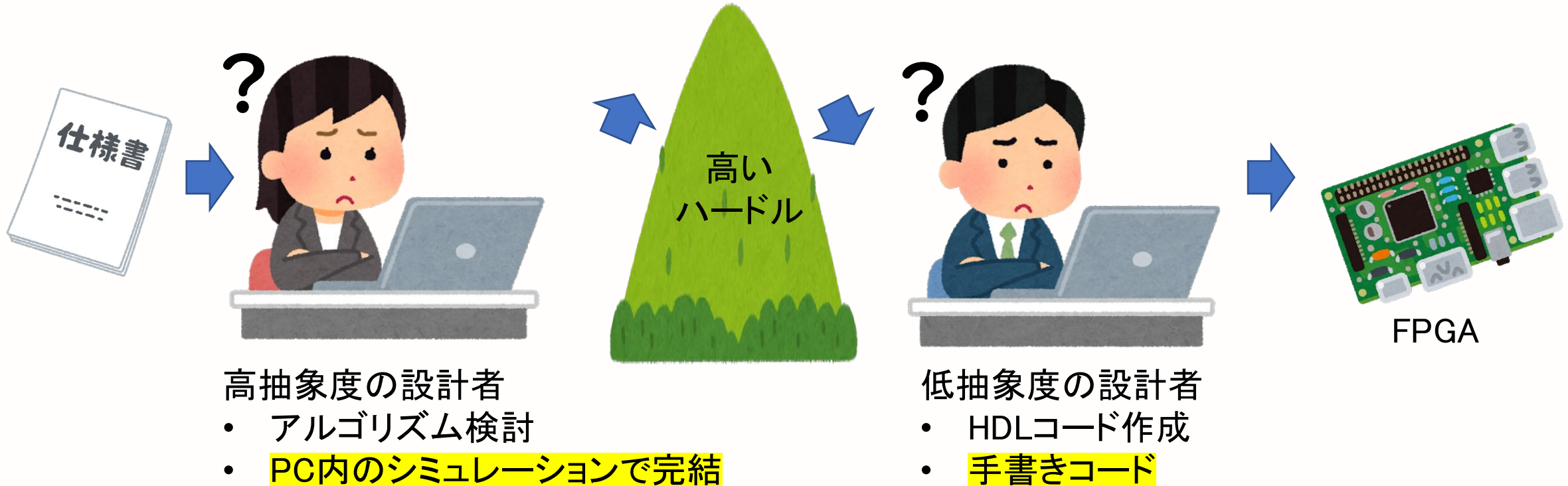


従来のFPGA開発フロー

アルゴリズム設計のシミュレーションは、アルゴリズム設計者のパソコン上で完結していたため、HDLコード設計者とは、分断されていました。

HDLコード設計者は、アルゴリズム設計の結果を、参考にしますが、ほぼゼロから、手書きコードを作成しなければいけませんでした。

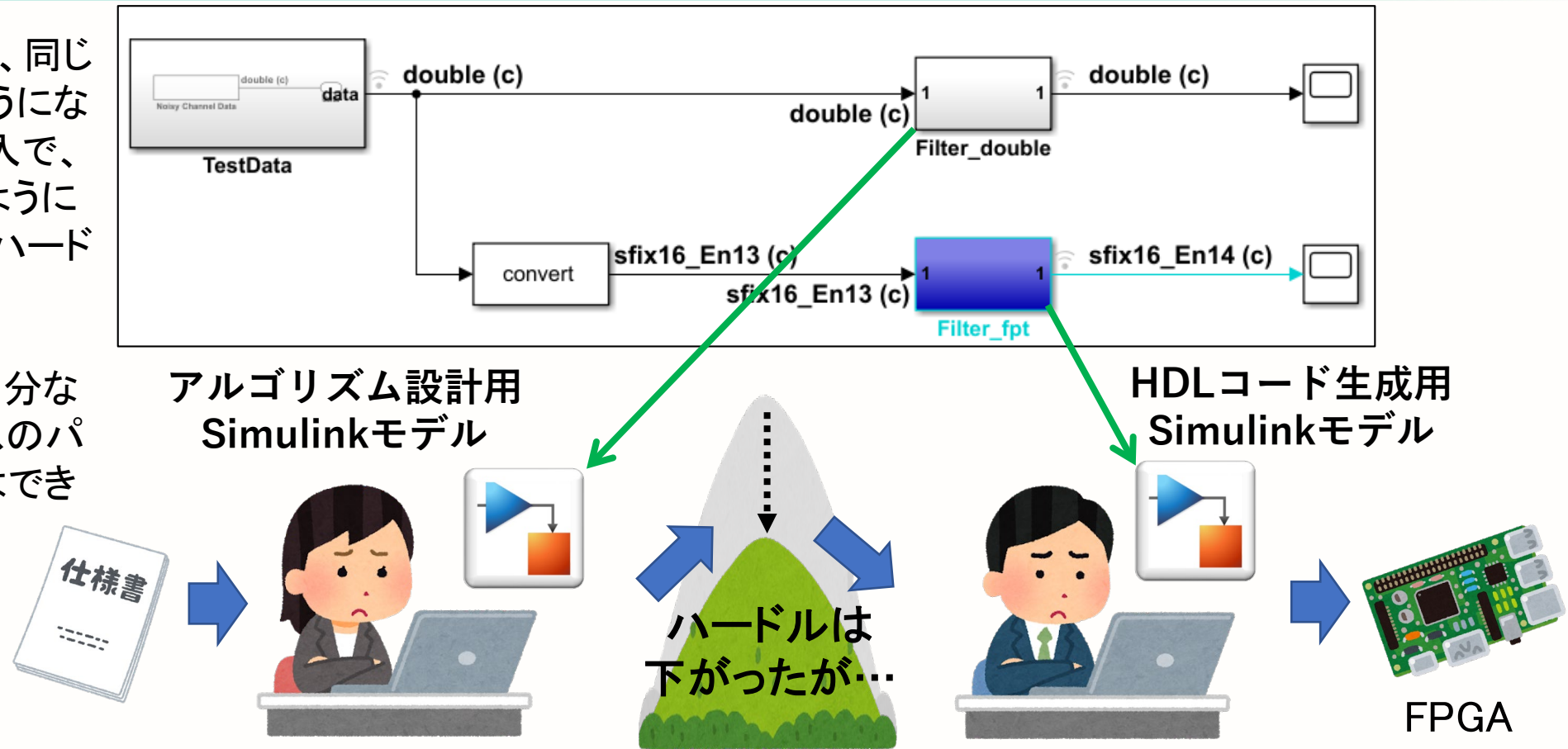
従来の開発フローでは、アルゴリズムの設計と、手書きHDLコードの作成が、全く異なる開発環境で、行われていました。その結果、回路規模の見積りや、等価性検証が困難でした。



HDL Coder の導入

MATLAB/Simulinkの導入で、同じ開発環境で設計ができるようになりました。HDL Coder の導入で、コードの自動生成も出来るようになりました。エンジニア間のハードルは確かに下がりました。

しかし、これだけでは必要十分な設計に落とし込み、デバイスのパフォーマンスを引き出す事はできませんでした。



必要十分な回路規模

分野に限らず、多くの設計者は自然と自分の担当する部分に、余裕を持たせる設計をしがちです。

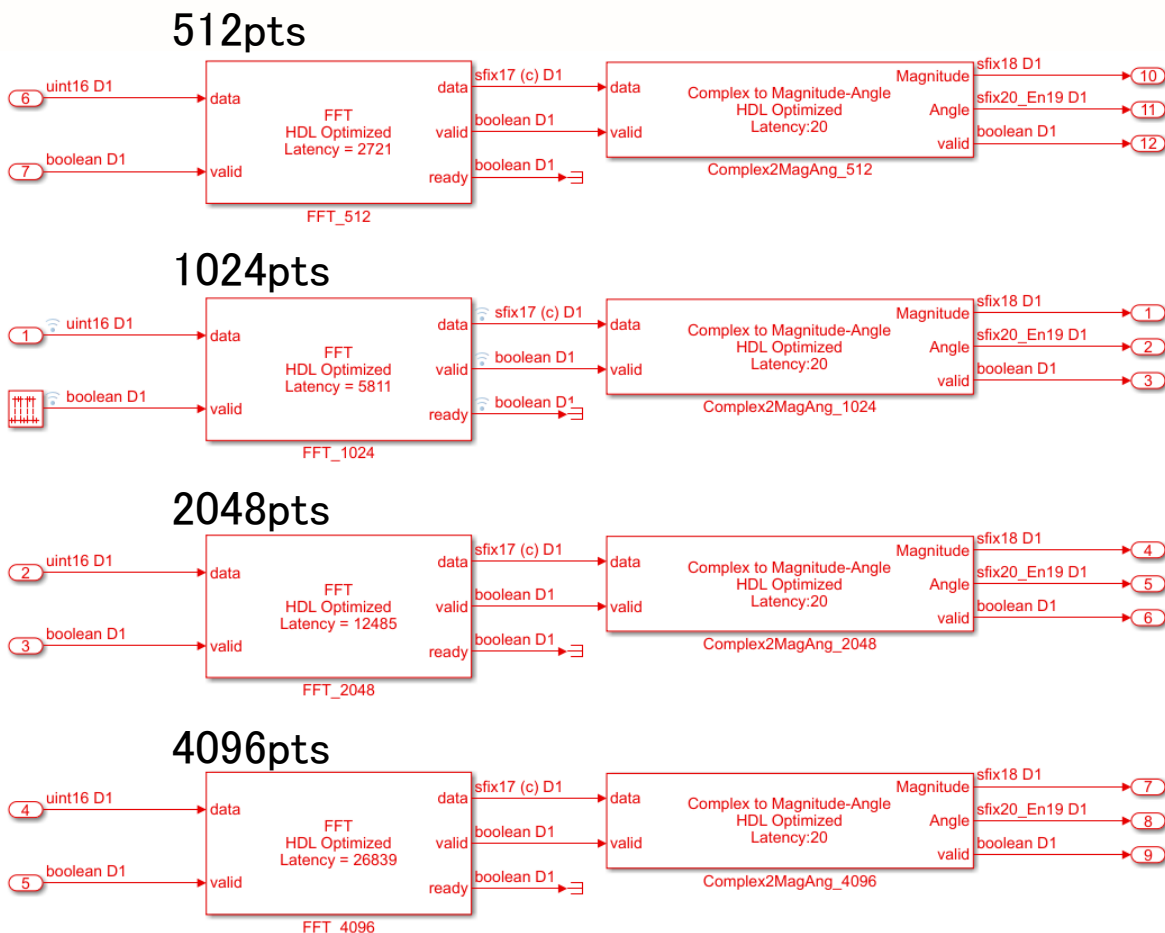
デジタル信号処理のアルゴリズム設計者の場合、デジタルフィルタの次数を上げる、FFTポイント数を増やす、などです。これはFPGAのリソースを多く消費する設計です。

一方、HDLコード設計者は、これとは逆にFPGAのリソースに余裕をもたせる設計をしがちです。

両者のコミュニケーションが取れていないと、必要十分な設計から外れてしまう事がよくあります。



回路規模とレイテンシの見積もり



FFT ポイント数	レイテンシ (処理サイクル数)	処理時間 @50MHz(usec)	Logic elements	乗算器	Fmax (MHz)
512	2721	54.42	2761	3	81.76
1024	5811	116.22	2941	3	88.48
2048	12485	248.70	3225	3	84.52
4096	26839	536.78	3717	3	82.45

FFT処理の回路規模やレイテンシを検討した際のSimulinkモデルです。HDLコード設計者は、上記のようなリソース消費の目安表を、アルゴリズム設計者に示します。

レイテンシはコンパイル(Ctrl+D)すれば表示されます。乗算器は高水準リソースレポート、回路規模(Logic elements)やFmaxは配置配線まで実行して求めています。

開発フローの早い段階で、リソースの目安を情報共有する事は、必要十分な設計に落とし込むのに重要です。

HDLワークフローアドバイザーの利用

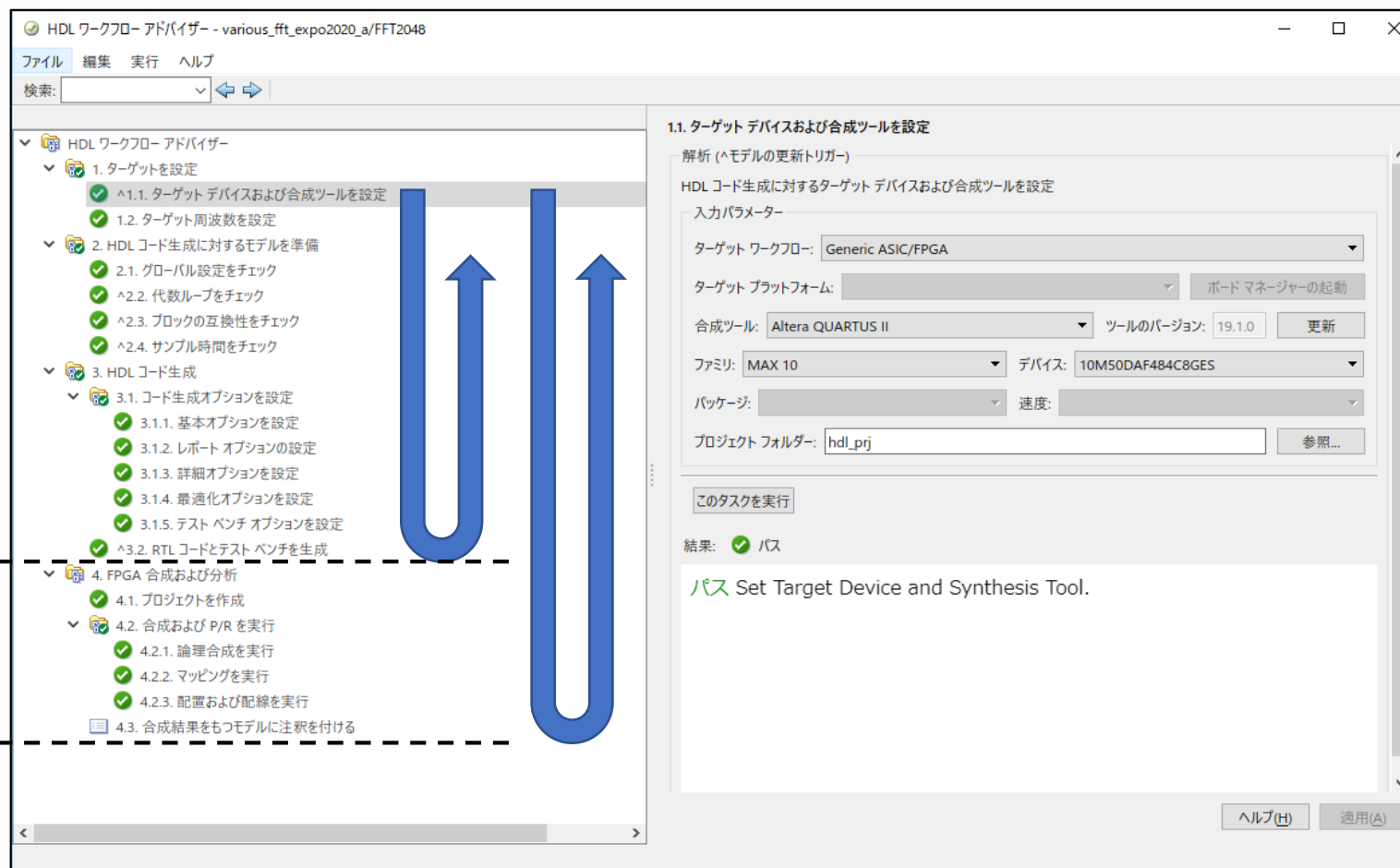
HDLワークフローアドバイザーを利用して、FPGAのリソース情報を共有しています。

アルゴリズム設計者でも、コード設計者の力を少し借りれば、コード生成可能です。高水準リソースレポートを気にするだけでも、情報共有の意義があると思います。

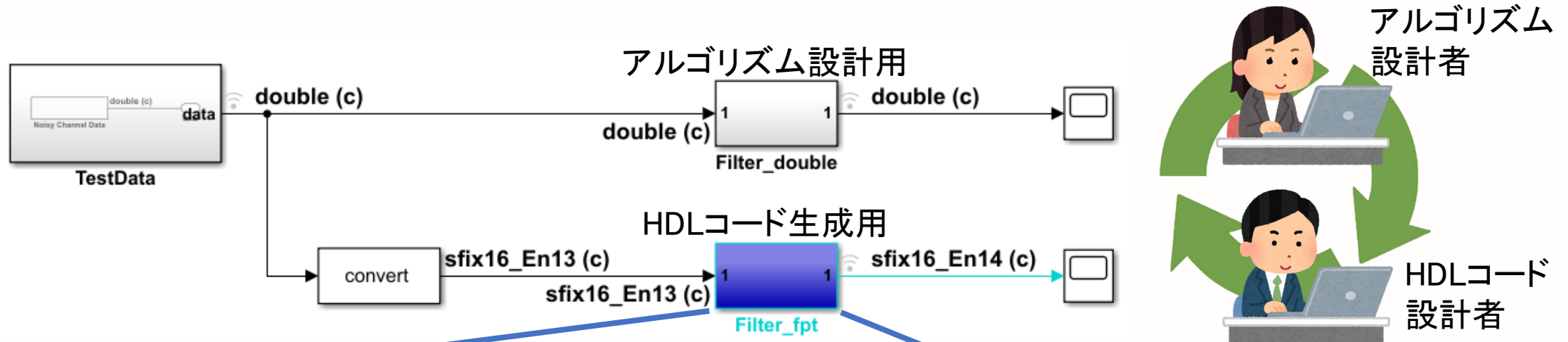


HDLコード生成
(高水準レポート)

配置配線



リソース共有ツールによる最適化



リソースとパフォーマンスの目安表

OutputPipeline	5	5	5	10	10	10	15	15	15
SharingFactor	1	2	4	1	2	4	1	2	4
Logic elements	3112	3133	3206	3069	3009	3230	3080	3112	3140
乗算器	26	13	7	26	13	7	26	13	7
Fmax (MHz)	66.54	73.87	80.94	102.21	95.83	103.19	147.47	132.73	127.75

通常、この目安表はHDLコード設計者が作成しますが、アルゴリズム設計者が作ろうと思えば作れます。早い段階でリソース情報を共有して、建設的な打ち合わせができます。手書きでHDLコードを書いている頃には、まず不可能な事でした。

浮動小数点HDLコードの利用

アルゴリズム設計者は
浮動小数点 で考える。



64 ビット浮動小数点 (double型) :
表現レンジ $-1.7977 \times 10^{308} \sim 1.7977 \times 10^{308}$
最小ステップ: 2.2251×10^{-308}

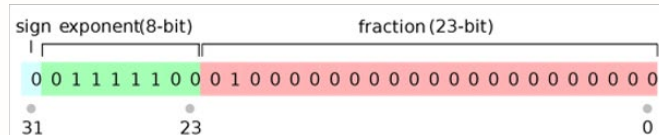
浮動小数点HDLコード

- 短時間で実装可能、リソース消費は大。
- 使い所を絞ればローエンドFPGAでも利用可能。

32 ビット浮動小数点 (single型) :
表現レンジ $-3.4028 \times 10^{38} \sim 3.4028 \times 10^{38}$
最小ステップ: 1.1754×10^{-38}

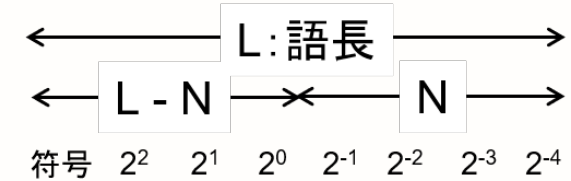


HDLコード設計者も
浮動小数点 で考えられる。

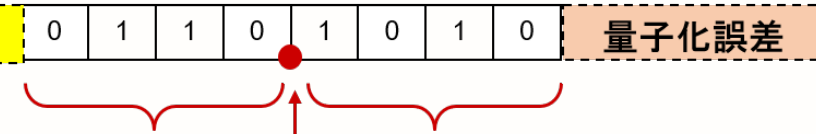


固定小数点化

- 正攻法だが、不向きなアルゴリズムもある。



オーバーフロー



整数部 小数点 小数部

表現レンジ: $-8.0000 \sim 7.9375$
最少ステップ: $2^{-4} = 0.0625$



HDLコード設計者は
固定小数点 で考える。

浮動小数点HDLコードの利用



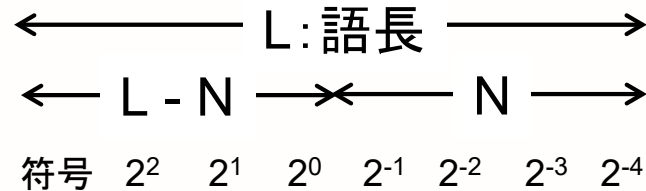
64 ビット浮動小数点:

表現レンジ $-1.7977 \times 10^{308} \sim 1.7977 \times 10^{308}$

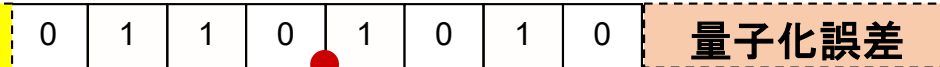
最小ステップ: 2.2251×10^{-308}

アルゴリズム設計者は
浮動小数点 で考える。

固定小数点化



オーバーフロー



量子化誤差

整数部 小数点 小数部

表現レンジ: $-8.0000 \sim 7.9375$

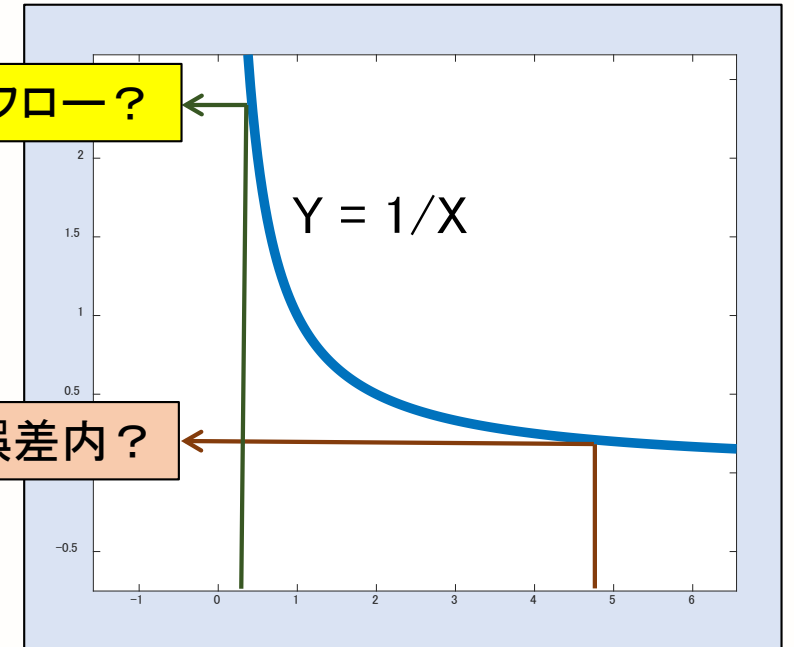
最少ステップ: $2^{-4} = 0.0625$



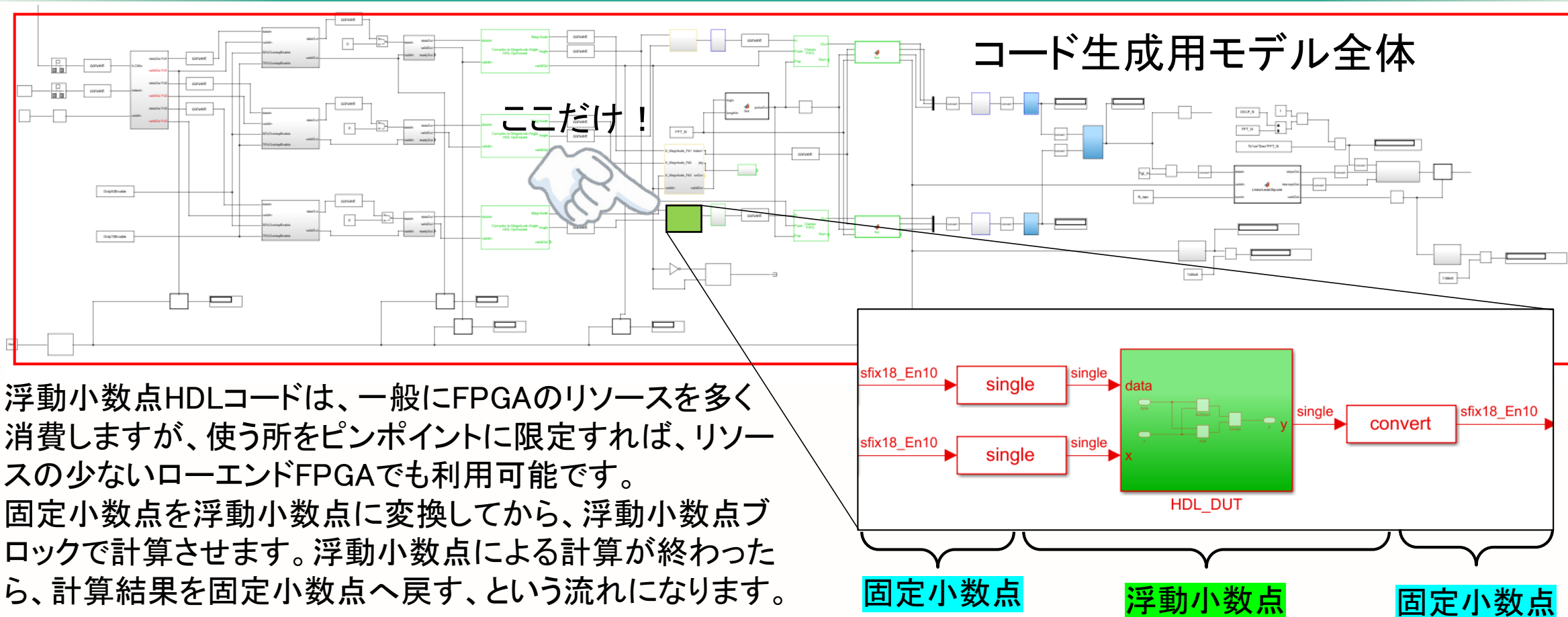
HDLコード設計者は
固定小数点 で考える。

簡単な計算に見えても、固定小数点化に不向きなアルゴリズムがあります。
アルゴリズム設計者は浮動小数点で考えるので、HDLコード設計者が何を悩んでいるのか分からないことがあります。

オーバーフロー?



浮動小数点HDLコードの利用



コミュニケーション補助ツールのまとめ

1. HDLワークフローアドバイザー

- HDLコード設計者はもちろん、アルゴリズム設計者でも操作できる。
- 上から順番に実行していけば、HDLコード生成や配置配線まで可能です。

2. リソース共有ツール

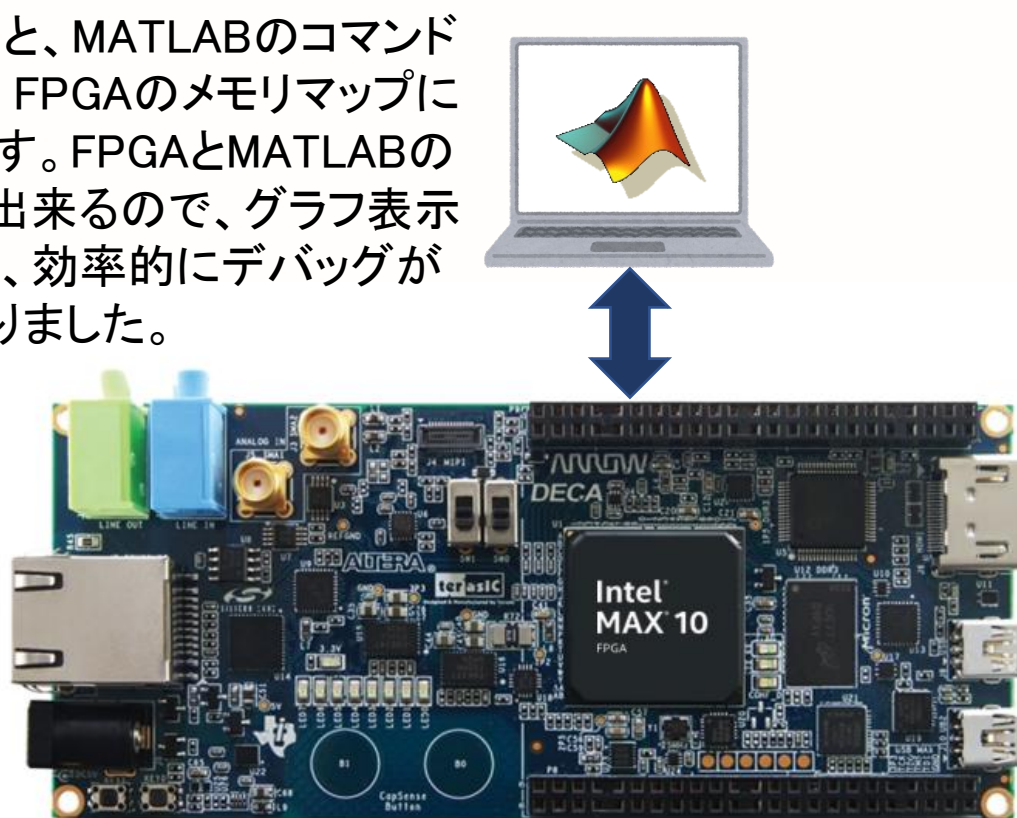
- 主に、乗算器の数の最適化で利用しています。

3. 浮動小数点HDLコード

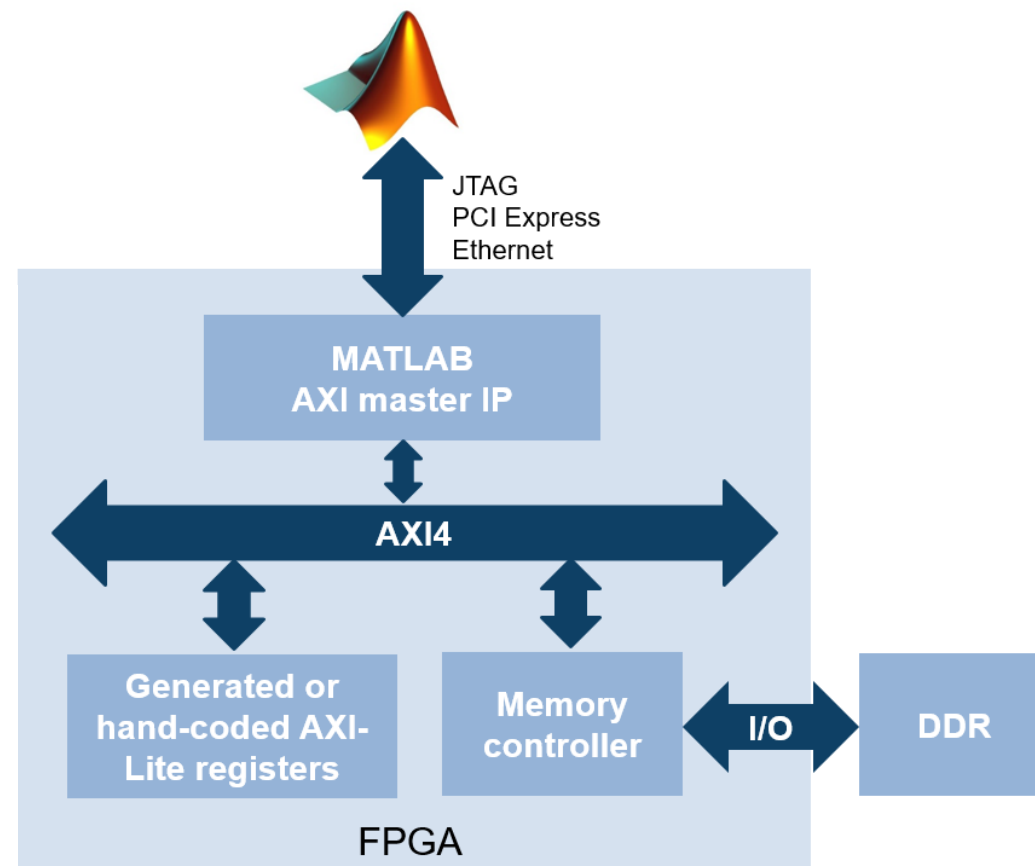
- 固定小数点化に不向きなアルゴリズムで利用しています。

実機デバッグの高機能化 (MATLAB as AXI Master)

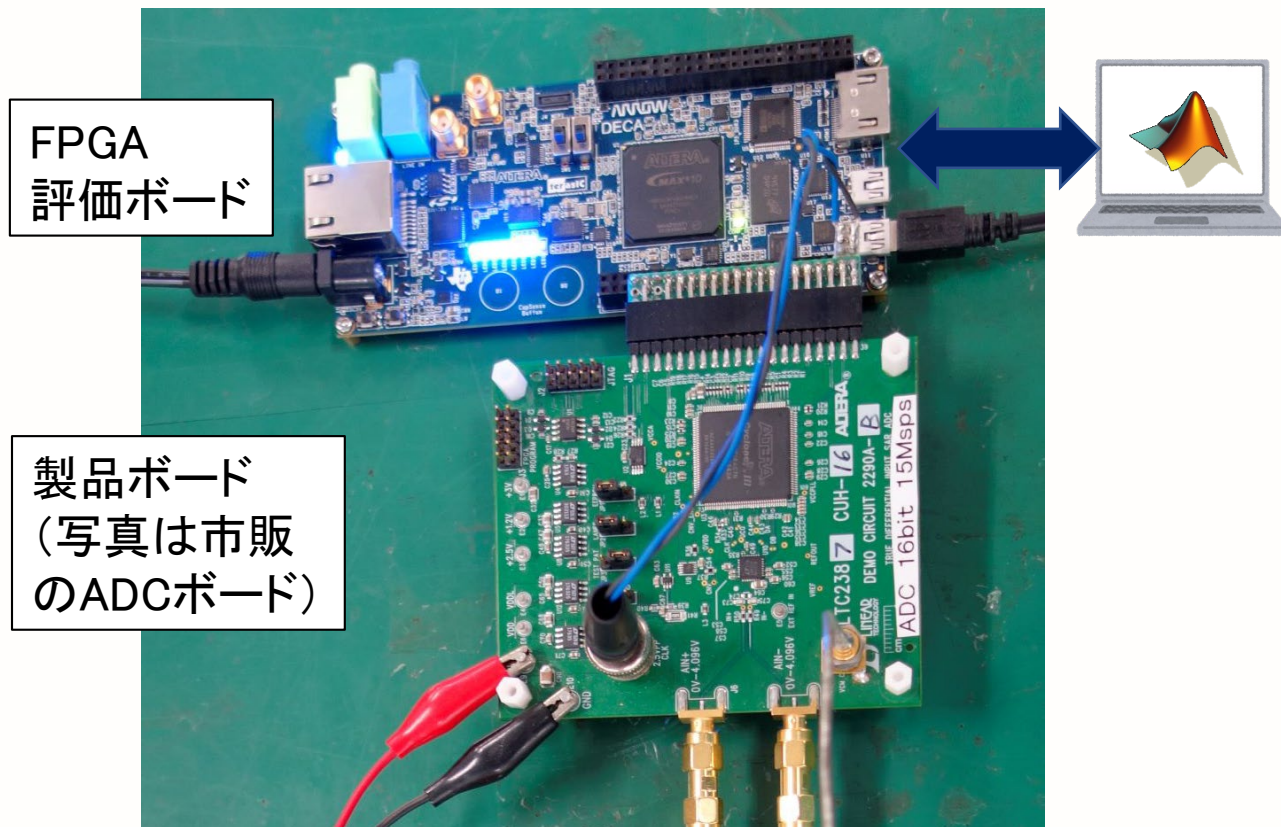
この機能を使うと、MATLABのコマンドウィンドウから、FPGAのメモリマップにアクセスできます。FPGAとMATLABの通信が簡単に出来るので、グラフ表示などを利用して、効率的にデバッグができるようになりました。



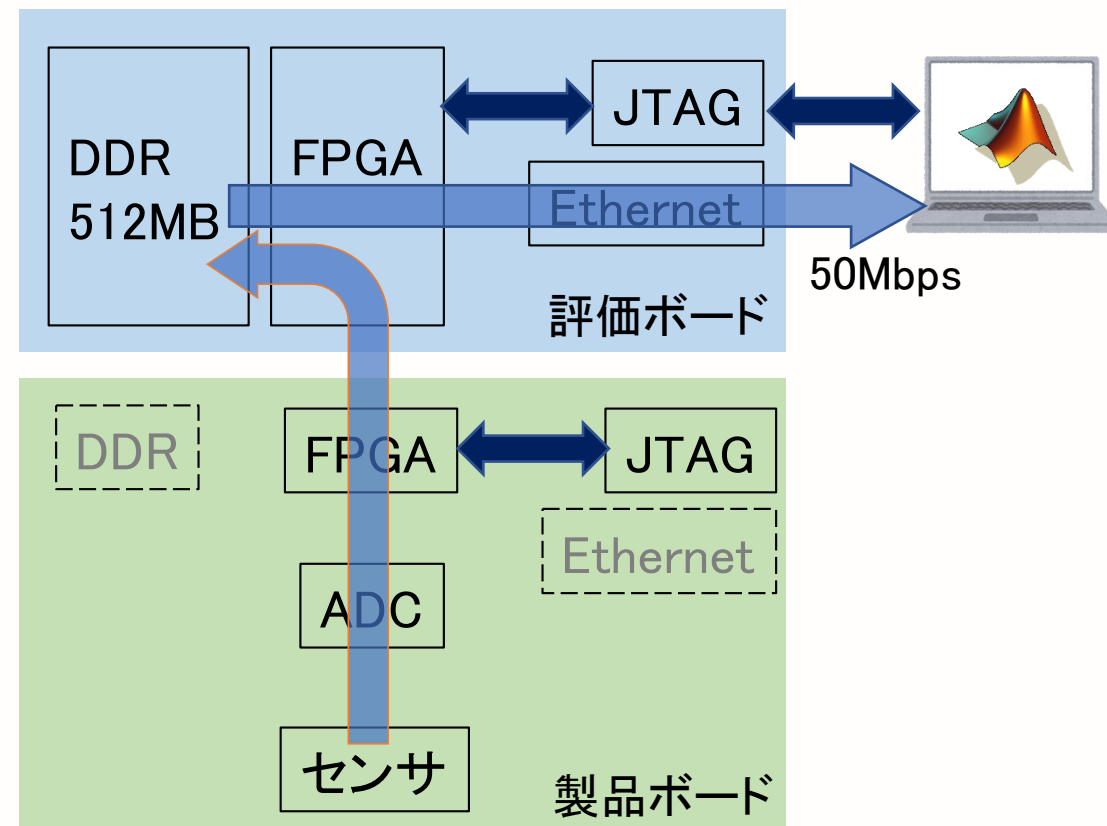
FPGA評価ボード



応用例①: 簡易データロガー

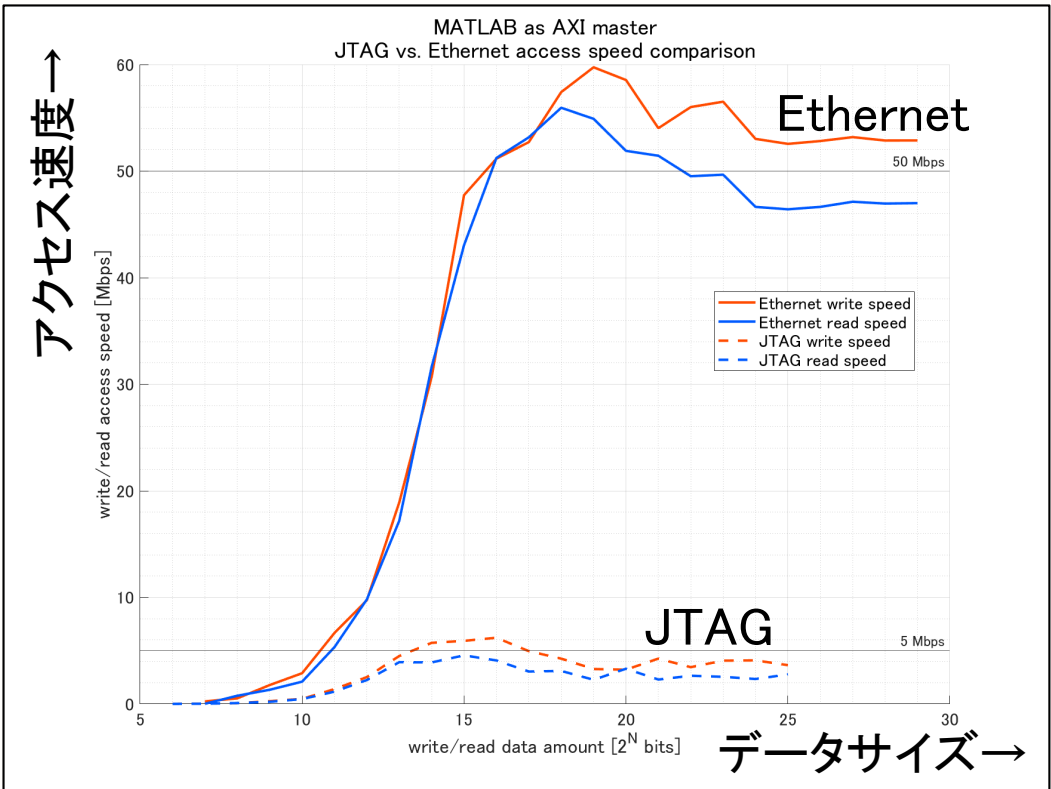
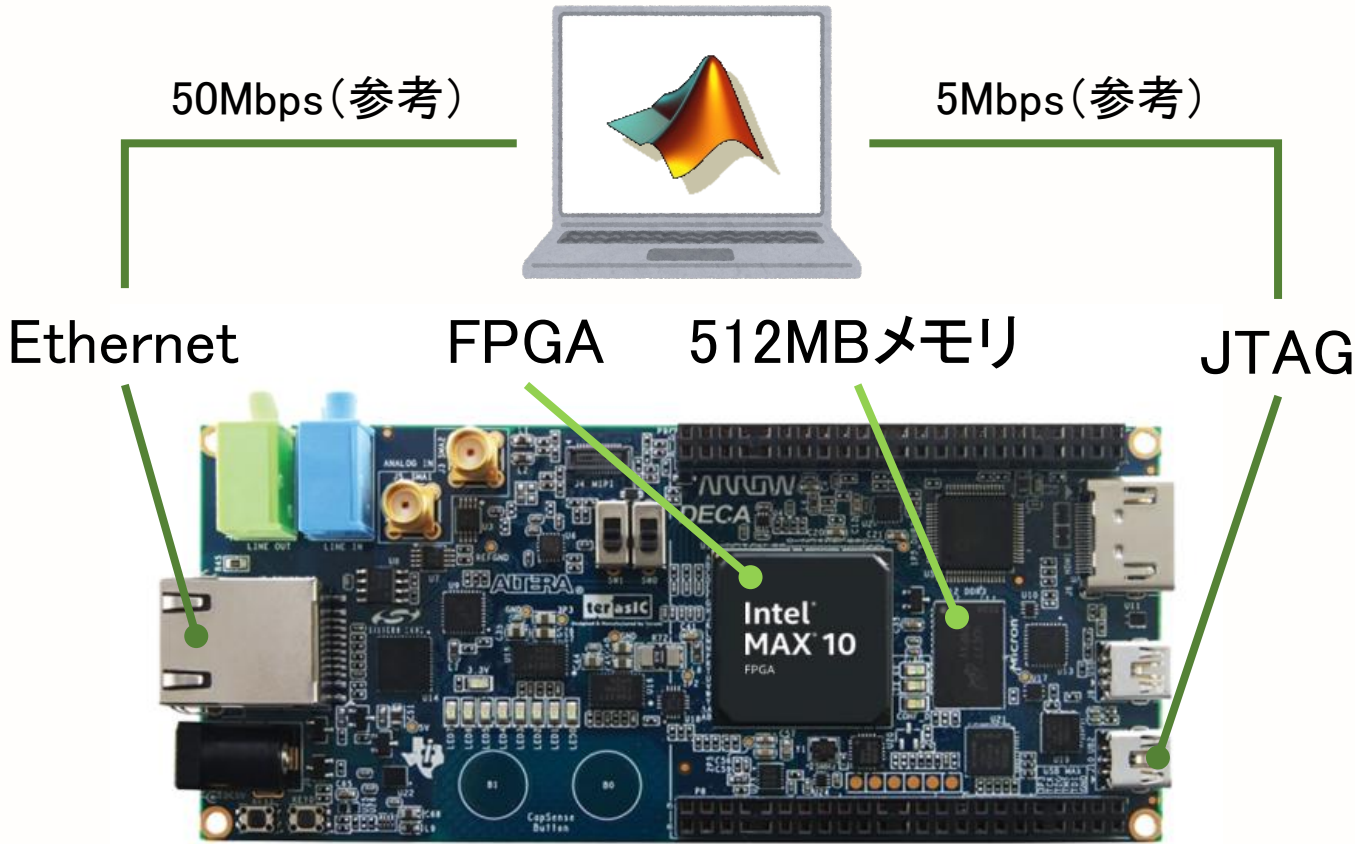


FPGA評価ボードと他のボードを接続



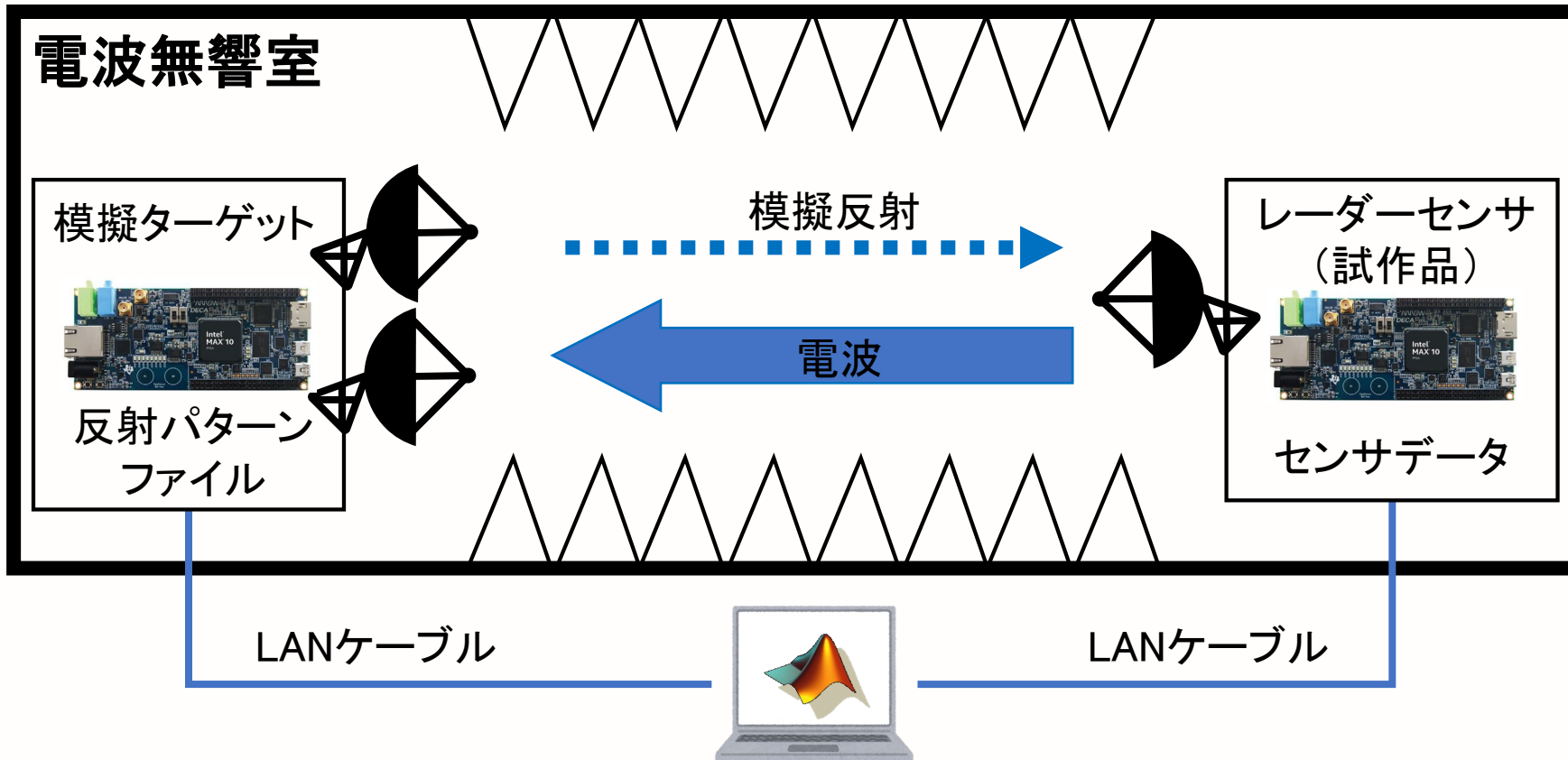
ブロック図(製品ボードに無いハードを補う)

通信速度の比較



通信速度の比較 JTAG vs Ethernet (当社測定)

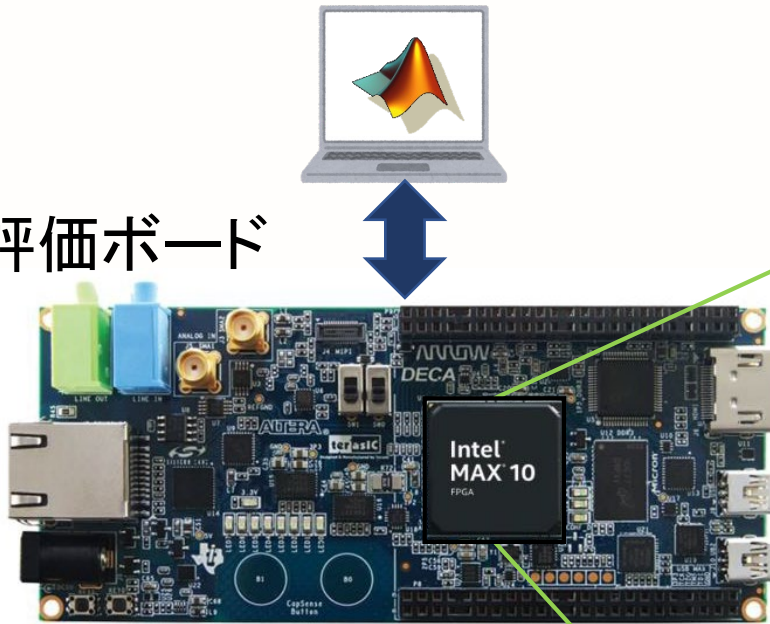
応用例②: 模擬ターゲット



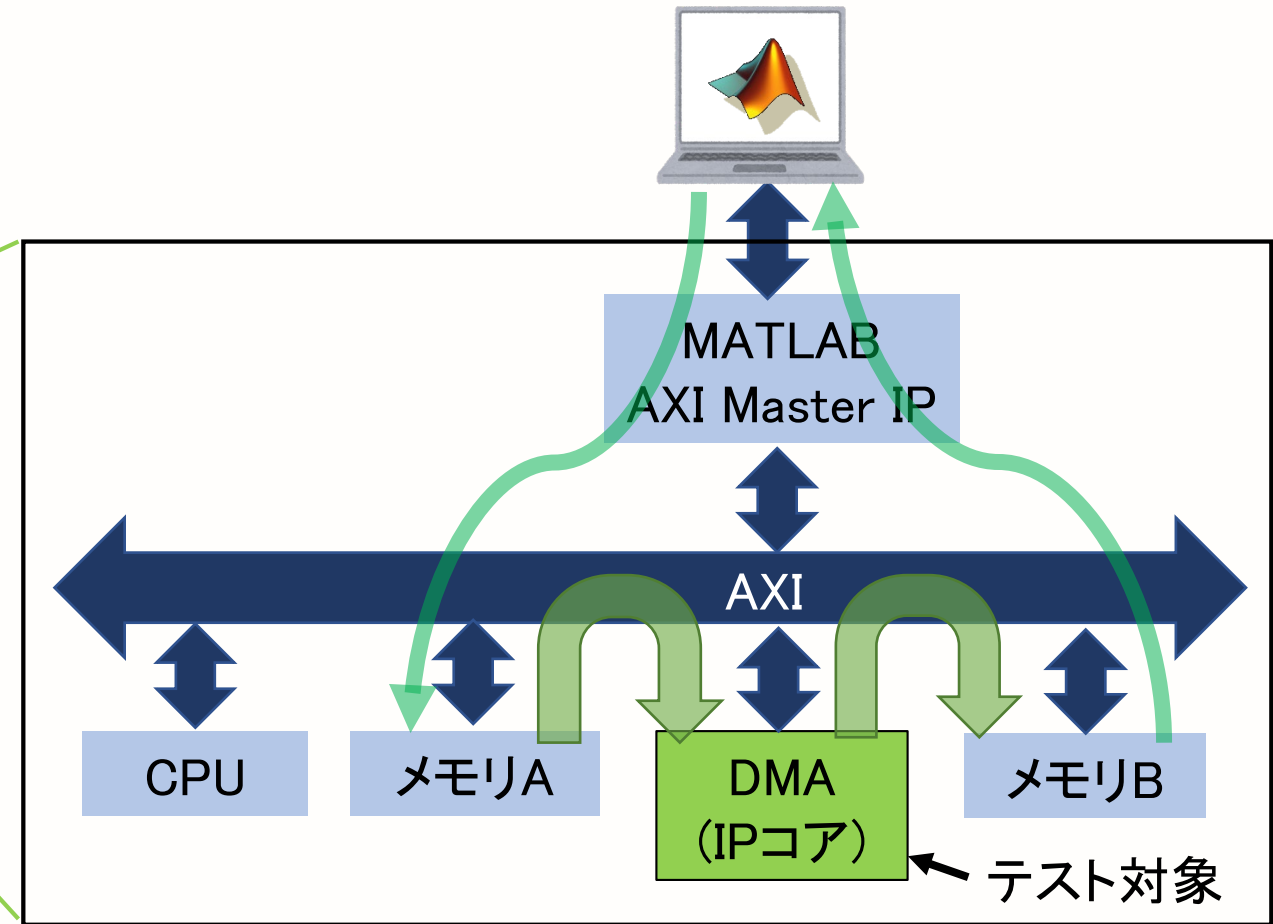
レーダーセンサの模擬ターゲットの作成に利用しました。模擬ターゲットとは、本来存在しないターゲットの反射を模擬する装置です。車載レーダーで例えると、いきなり走行している本物の車で実験するのは非現実的なので、このような装置を用意します。模擬ターゲットに書き込む反射パターンのデータ、レーダーセンサ本体が取得するデータ、共にサイズが大きいのですが、高速転送してMATLABで解析可能です。PCは電波無響室の外に置くので、LANケーブルが使える点も便利でした。

IPコアの動作確認

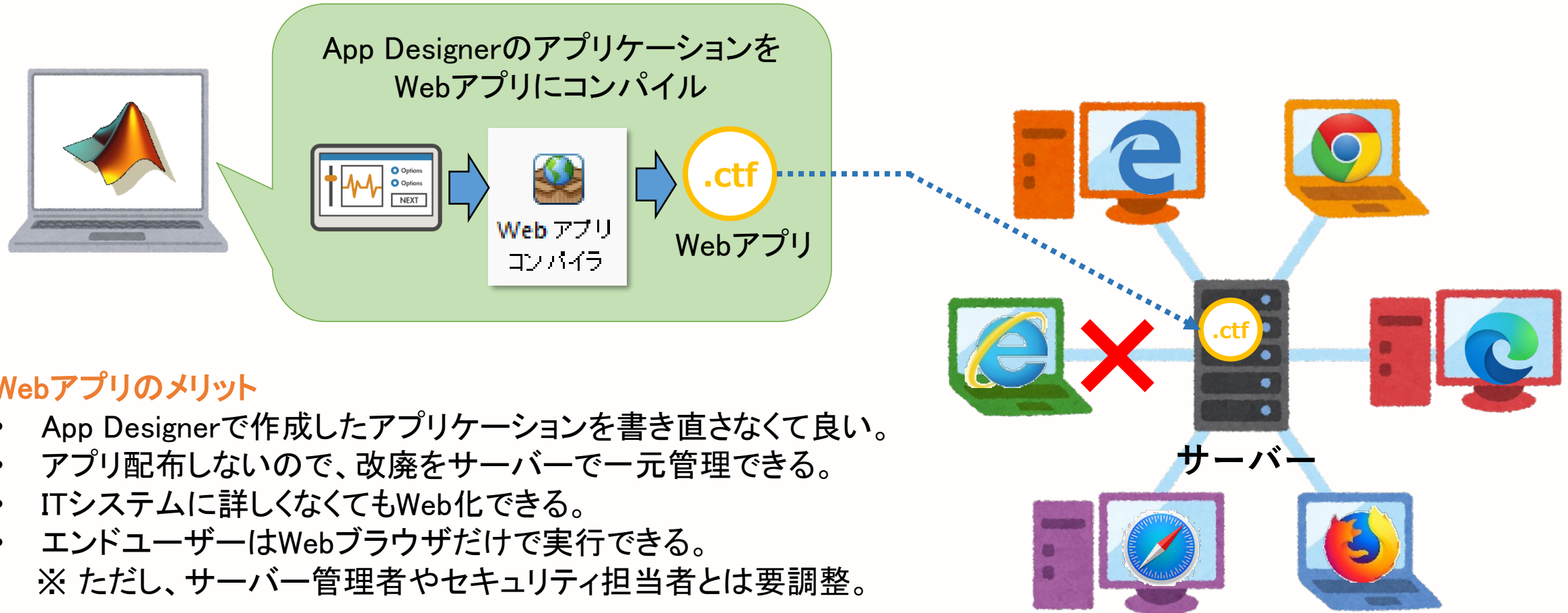
FPGA評価ボード



他の要素を排除して、MATLABからIPコアの動作確認をできるのが魅力です。従来はデバッグ用のHDLコードやCコードにバグが入り込み、何をデバッグしているのか迷走する事がありました。



Webアプリによる実験結果の共有



Webアプリのメリット

- App Designerで作成したアプリケーションを書き直さなくて良い。
 - アプリ配布しないので、改廃をサーバーで一元管理できる。
 - ITシステムに詳しくなくてもWeb化できる。
 - エンドユーザーはWebブラウザだけで実行できる。
- ※ ただし、サーバー管理者やセキュリティ担当者とは要調整。

Webアプリ化の思わぬ効能



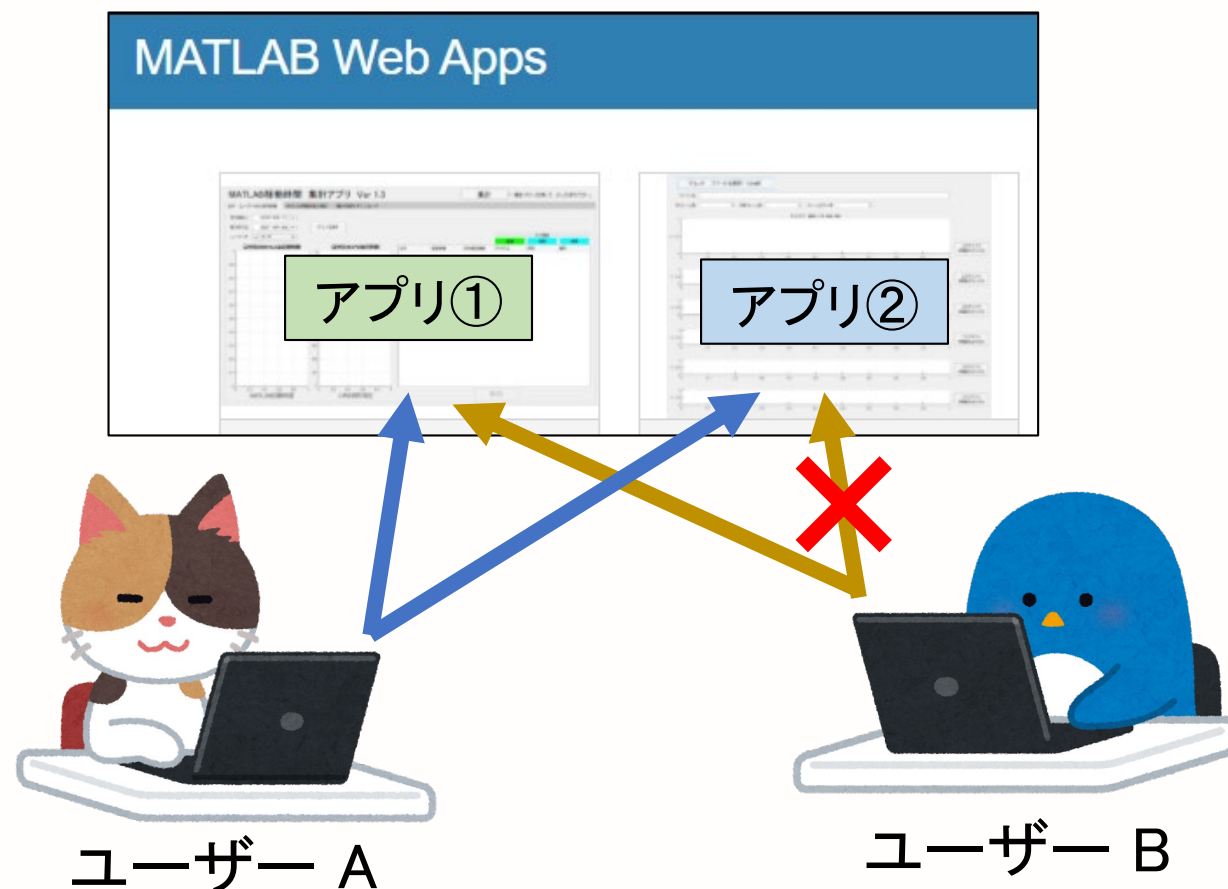
新型コロナウイルスの影響で、1つのPCの前に大勢が集まる事が難しくなりました。MATLABに慣れている社員が、テレワークで出社していないケースも増えています。そんな状況でも実験結果をWebアプリで共有できるのは、思わぬ効能でした。

開発版と製品版

R2020a

特徴	開発版 MATLAB Web App Server	製品版 MATLAB Web App Server	新製品の利点
ユーザー認証 ／アクセス制御	×	✓	Webアプリへのユーザーアクセスを 制御できます。
MATLAB Runtime のバージョン	1つのみ	複数	MATLABをバージョンアップしたときに、 Webアプリをコンパイルし直さなくても これまでのRuntimeで動作し続けら れます。
同時利用セッション数	32	無制限	- サーバーハードウェアの容量でサ ポートできる限りアプリへのアクセス 数に制限がありません。 - IT管理者側でユーザーベースの 制御ができます。

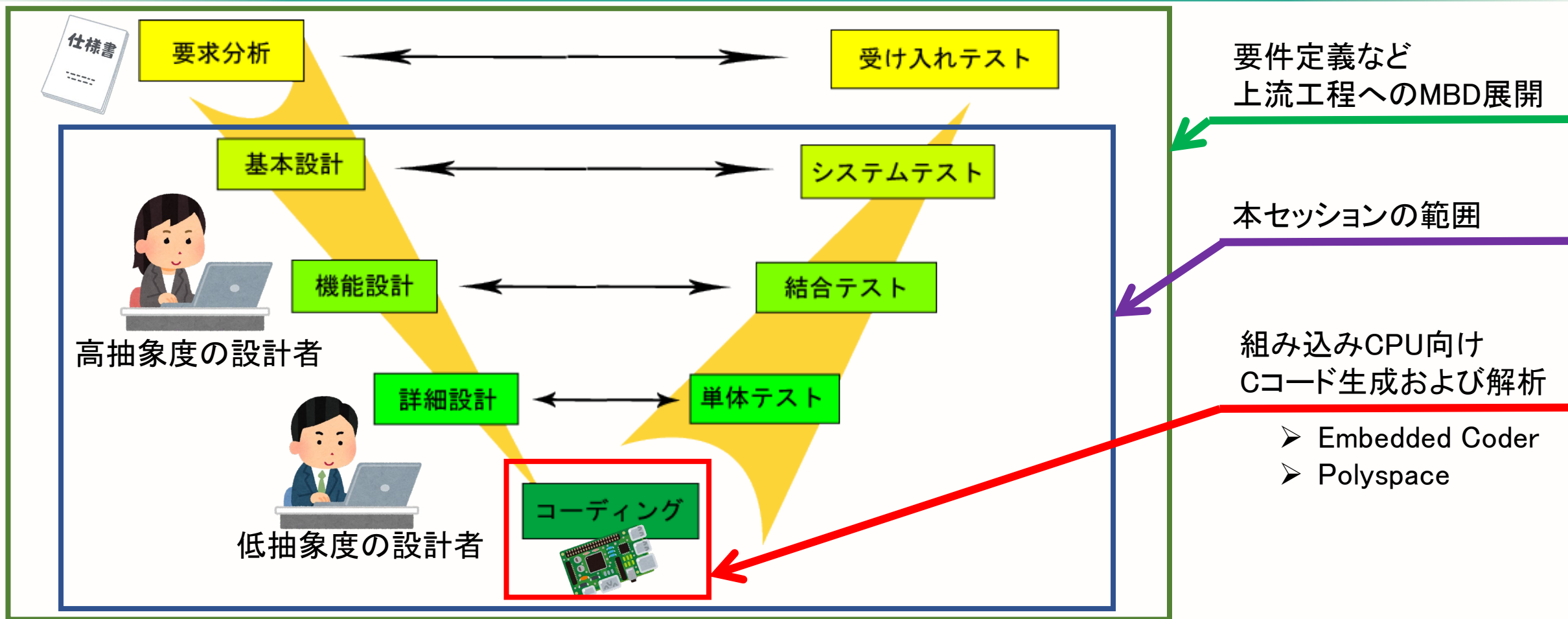
Webアプリへの要望



ユーザーごとにアクセスできるアプリを指定できるようにして欲しいと思っています。特定の開発チームだけが利用できるようにしたいアプリもあれば、複数の部署から利用したいアプリもあるためです。

製品の試験結果をグラフ化するようなアプリは部署内のみで共有。MATLABの稼働時間を集計するようなアプリは、他の開発チームとも共有といった使い方を想定しています。

MBDに関する現在の取り組み



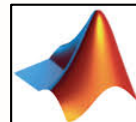
まとめ

- MBD導入の動機
 - 手書きHDLコードの限界
 - 後工程につながらないアルゴリズム設計の限界
- MBD導入初期に見えてきた課題(コード自動生成が出来るようになってから)
 - 抽象度の異なるエンジニア間のコミュニケーションが重要(MBDに限らず)
 - 回路規模、レイテンシの早期見積もりと情報共有
 - リソース共有ツールや浮動小数点HDLコードの活用
- 実機デバックの高機能化と効率化
 - MATLAB as AXI Master の活用
 - 簡易データロガーやテスト治具を短時間で構築
 - IPコアのテスト効率化
- 実験結果をWebアプリで共有化
 - テレワーク時代におけるWebアプリ化の可能性

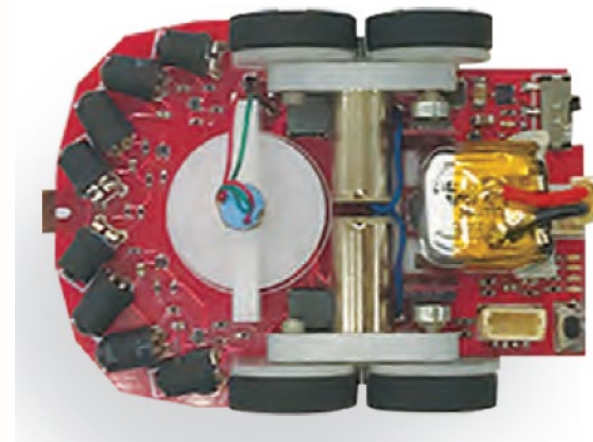
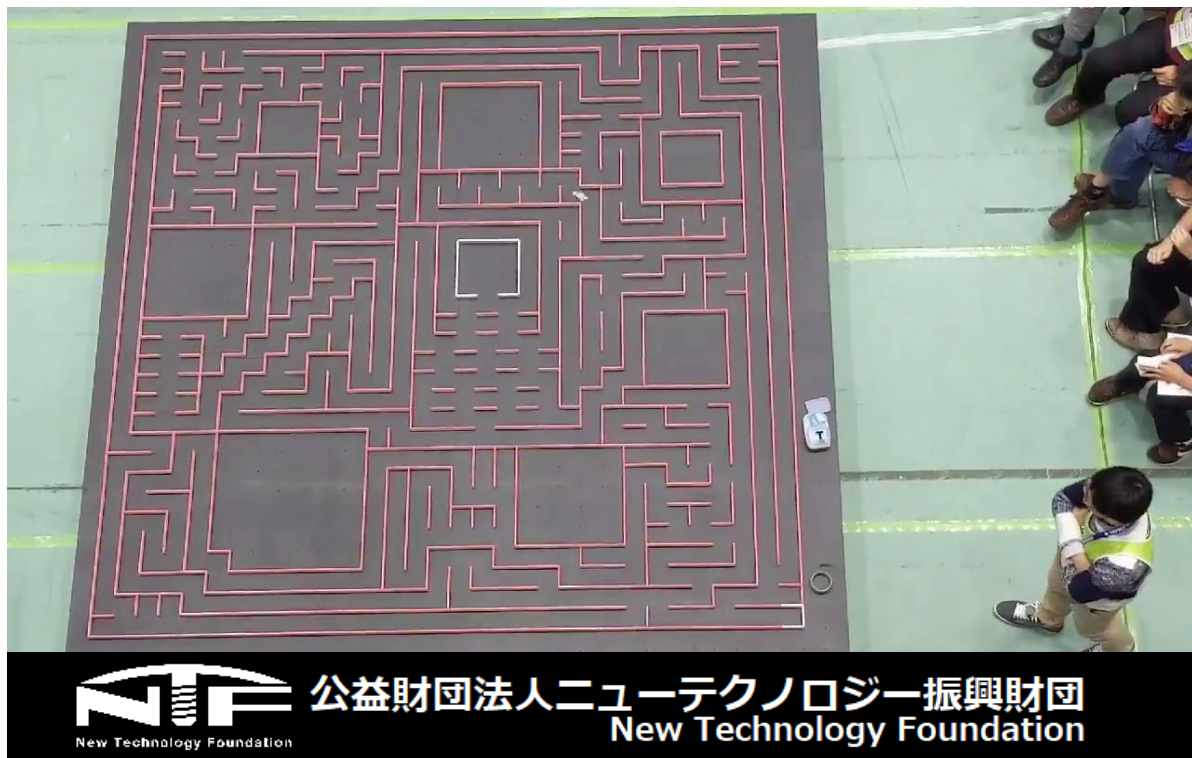
マイクロマウス

協賛企業:

YDK Technologies



MathWorks® など



2019年全国大会の優勝ロボット

長さ52mm x 幅38mm x 高さ20mm 重量11.8g

MathWorks Micromouse Contest : <https://jp.mathworks.com/academia/student-competitions/micromouse-contest.html>
(公財)ニューテクノロジー振興財団: <https://www.ntf.or.jp/>

ご視聴ありがとうございました



YDK Technologies
Sensing with You