



**TOSHIBA**

MATLAB EXPO 2023 JAPAN

セッション：パワーエレクトロニクスとエネルギーマネジメント

# SoC/FPGA評価基板を活用した パワエレ制御開発向けHILS環境の構築

東芝インフラシステムズ株式会社

インフラシステム技術開発センター

電機応用・パワエレシステム開発部

モータ・ドライブシステム技術担当

齋藤 亮介

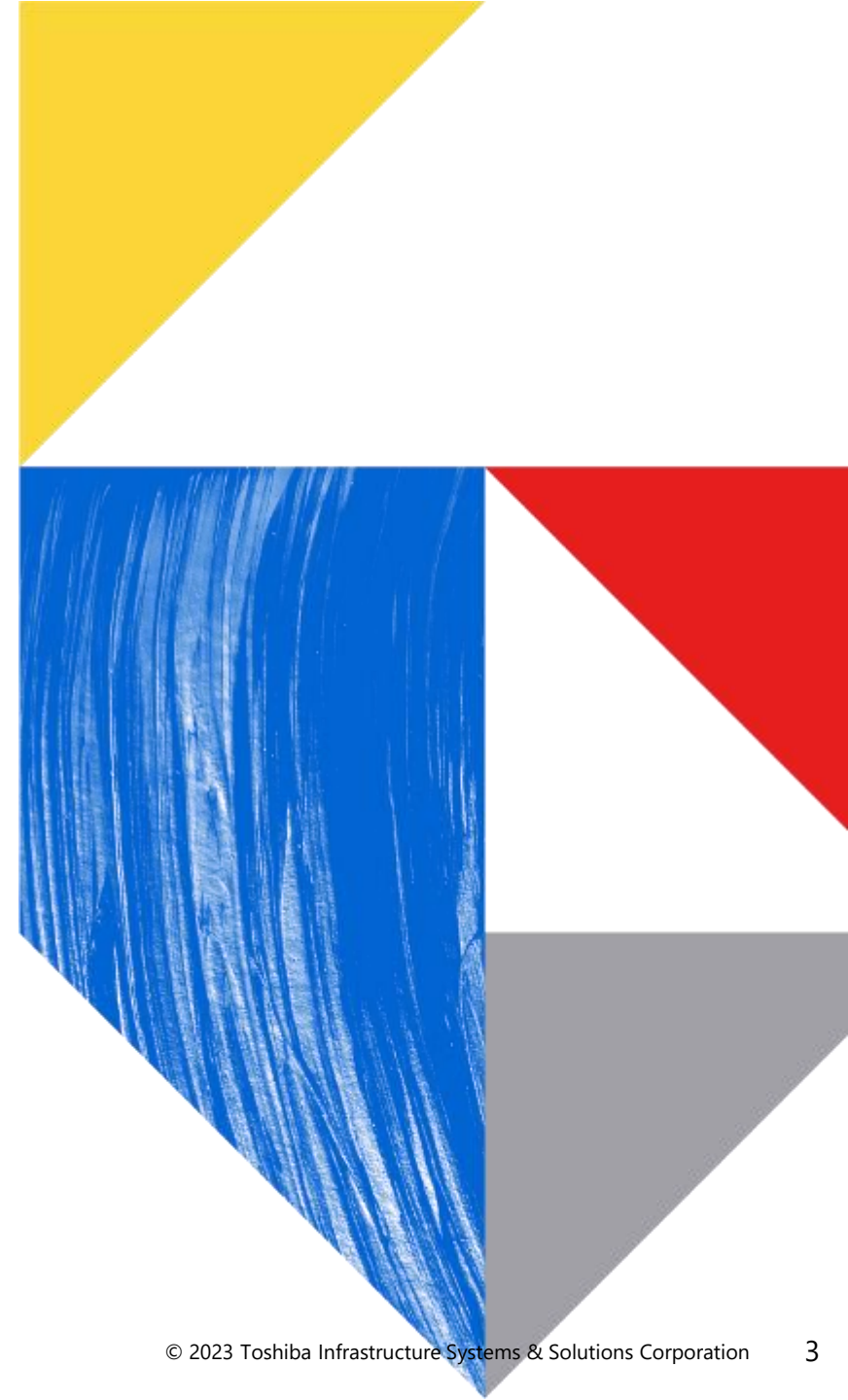
2023.05.31

# Contents

- 01 パワエレ制御へのHILS検証のモチベーション
- 02 HILS検証の導入・課題（2008年～）
- 03 市販SoC/FPGAを活用したRTS開発
- 04 MATLAB<sup>®</sup>/Simulink<sup>®</sup>を活用した電気モデル開発
- 05 まとめ・今後の展開

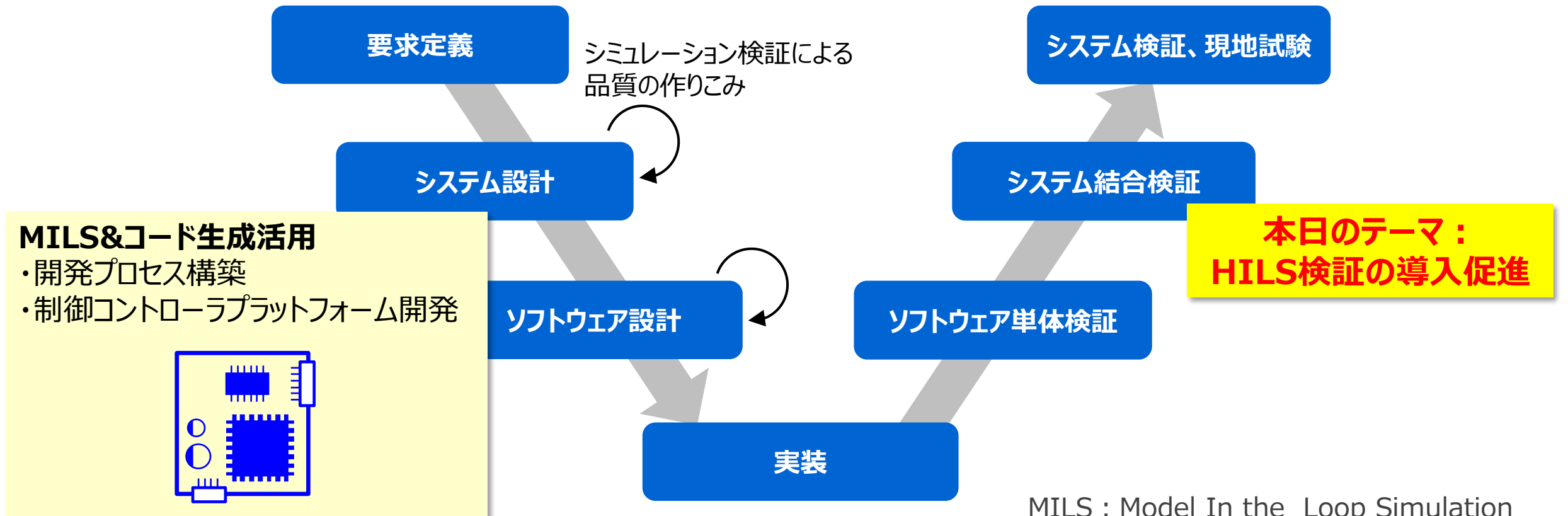
# 01

## パワエレ制御へのHILS検証のモチベーション



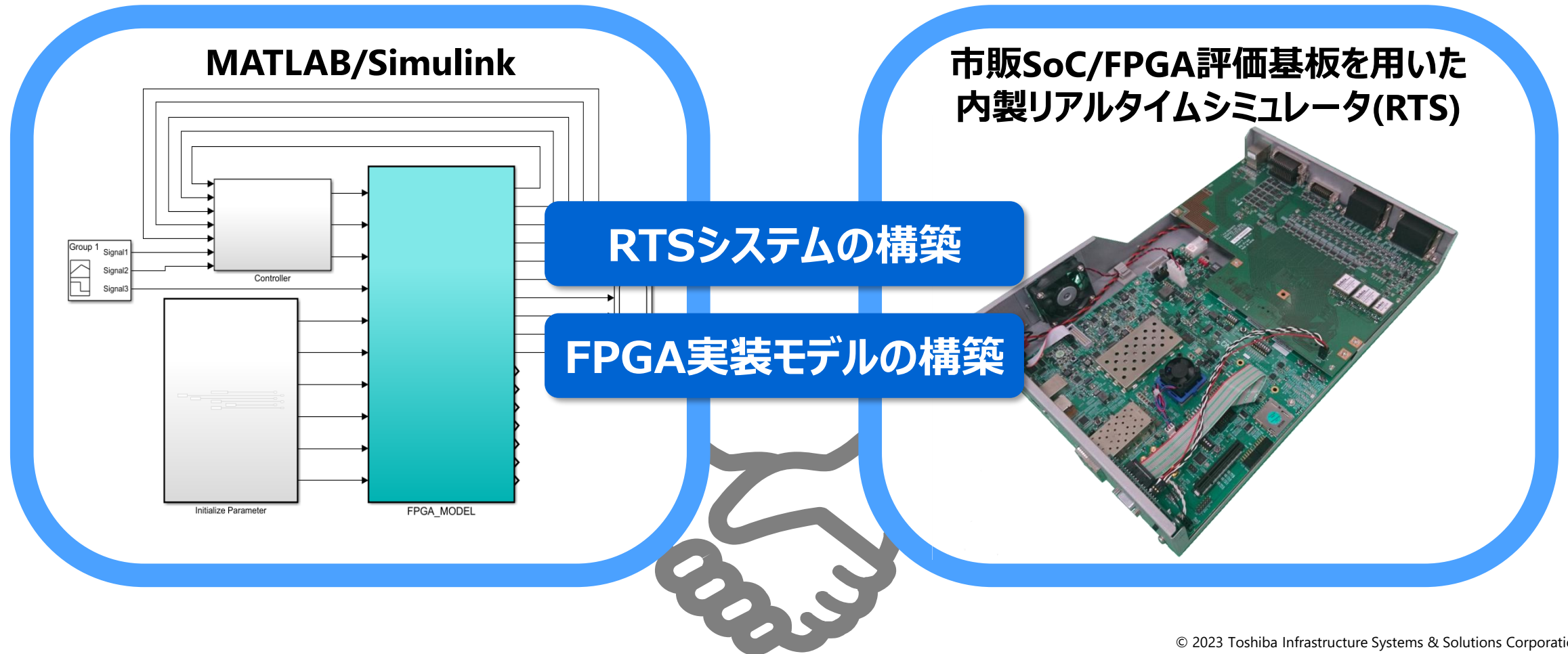
# モデルベース開発の取り組み（対象：パワーエレクトロニクス制御の開発検証）

開発効率と品質を両立すべく、設計・検証へのシミュレーション技術導入を進めている。  
→システム検証の負荷軽減が喫緊の課題、HILS検証の導入促進が重要。



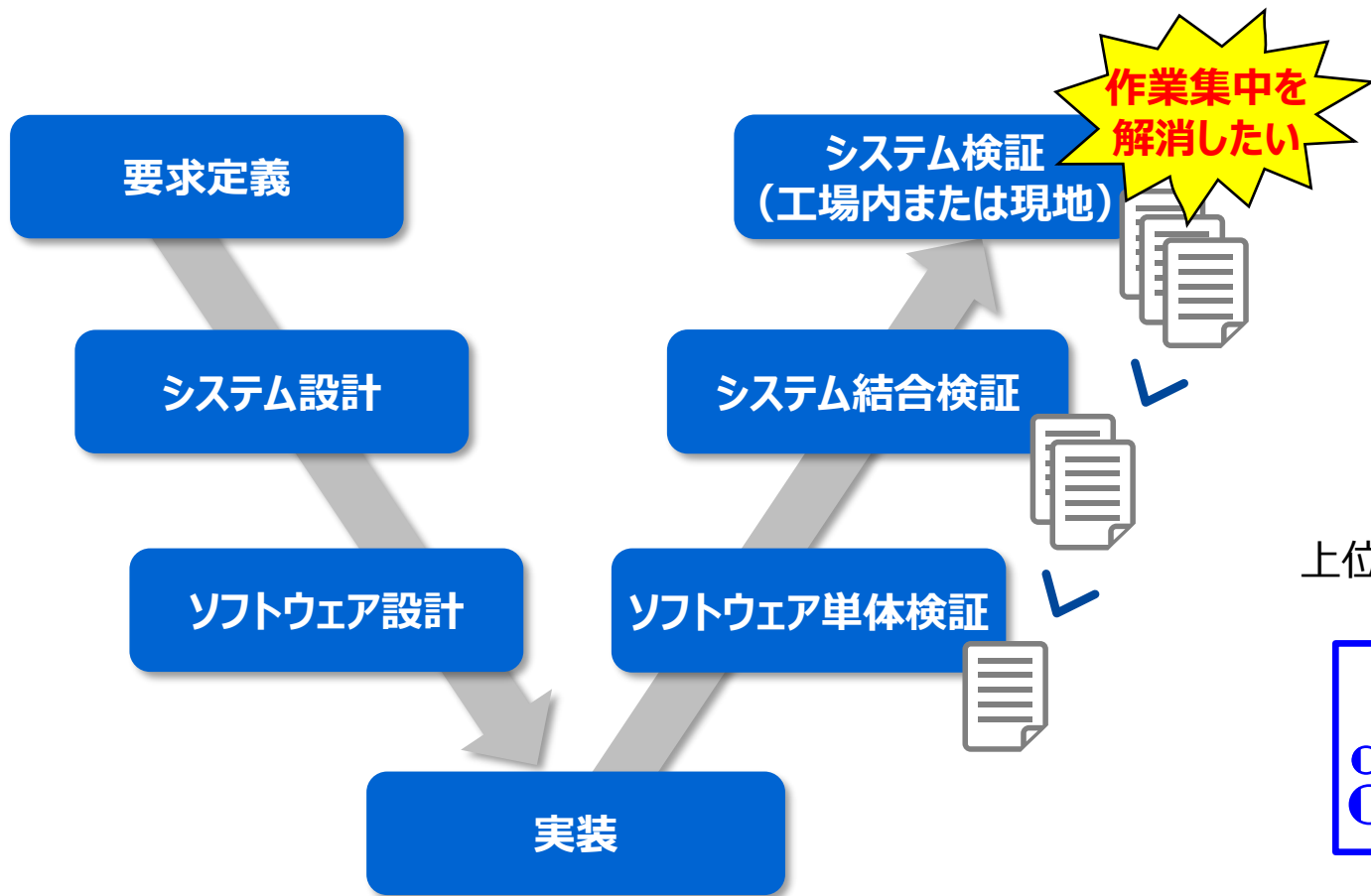
MILS : Model In the Loop Simulation  
HILS : Hardware In the Loop Simulation

パワエレ制御のHILS検証導入促進を目的に、  
「**MATLAB®/Simulink® + 内製リアルタイムシミュレータ**」のHILS検証環境を構築。



# HILS検証導入のモチベーション ～パワエレ制御検証における傾向～

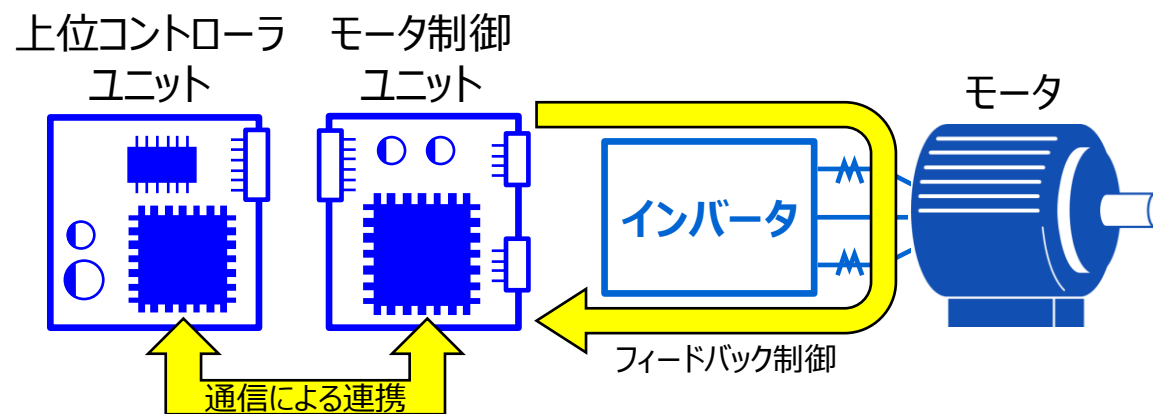
パワエレ制御ソフトウェア検証の特徴：検証作業がシステム検証に集中しやすい傾向。  
→システム検証での手戻りリスクが大きく、工程に大きな影響を与える。



## 検証作業が集中する要因

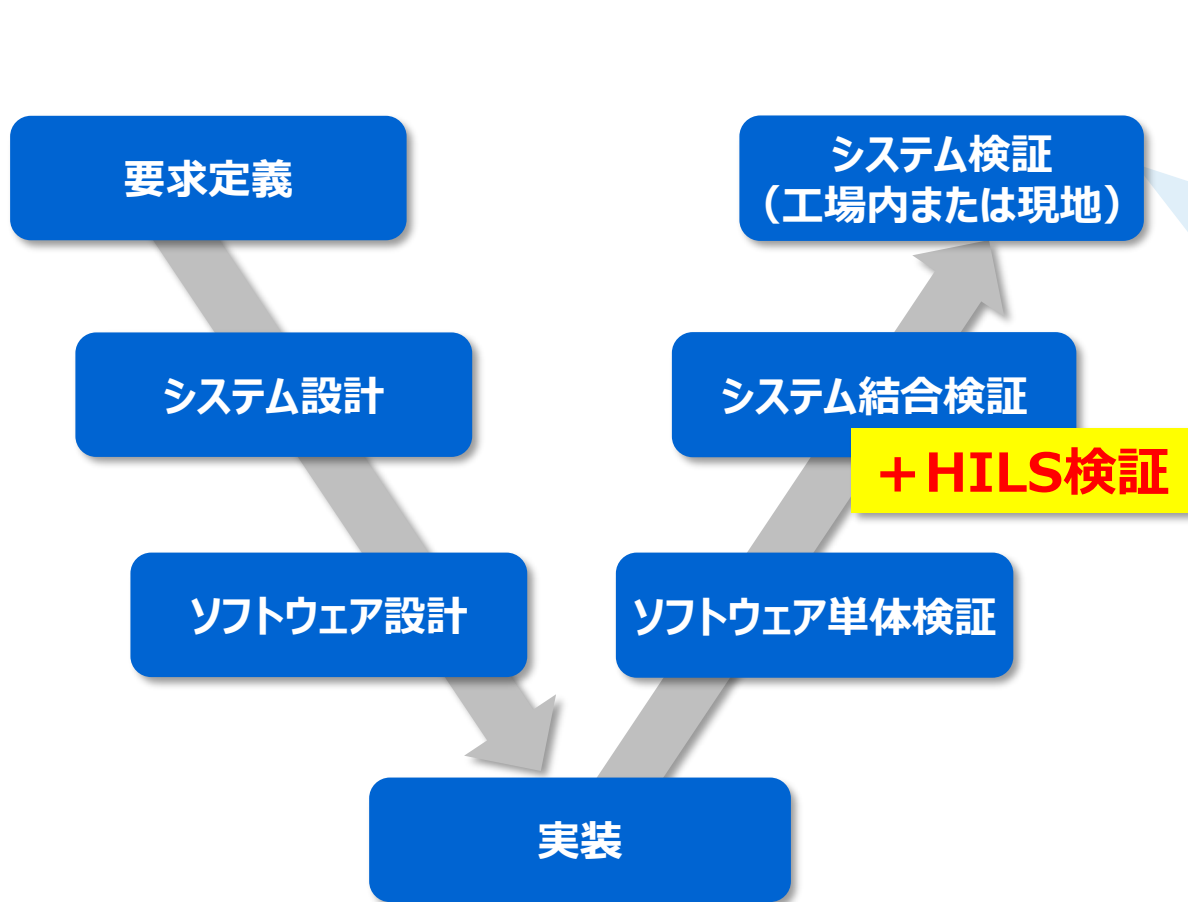
- ・制御装置が複数ユニットから構成される  
→機能の多くがユニット結合後に検証可能となる。
- ・フィードバック制御を多用する  
→制御対象（モータ、インバータなど）と接続して、動的な性能検証が可能となる。

例：モータ制御装置の構成



# HILS検証導入のモチベーション ～大型・高出力インフラ製品ならではの傾向～

大型・高出力製品では検証スケジュール変更が難しく、手戻りを極力抑制したい。  
→システム検証負荷を軽減し、前工程での品質確保のためHILS検証を導入。



大型・高出力なインフラ製品では



- ・試験場所、設備のスケジュール確保が難しい  
(大容量の電源や負荷など)
- ・安全性確保のため複数の人員の確保が必要
- ・運搬、設置、撤去などの日程確保が必要

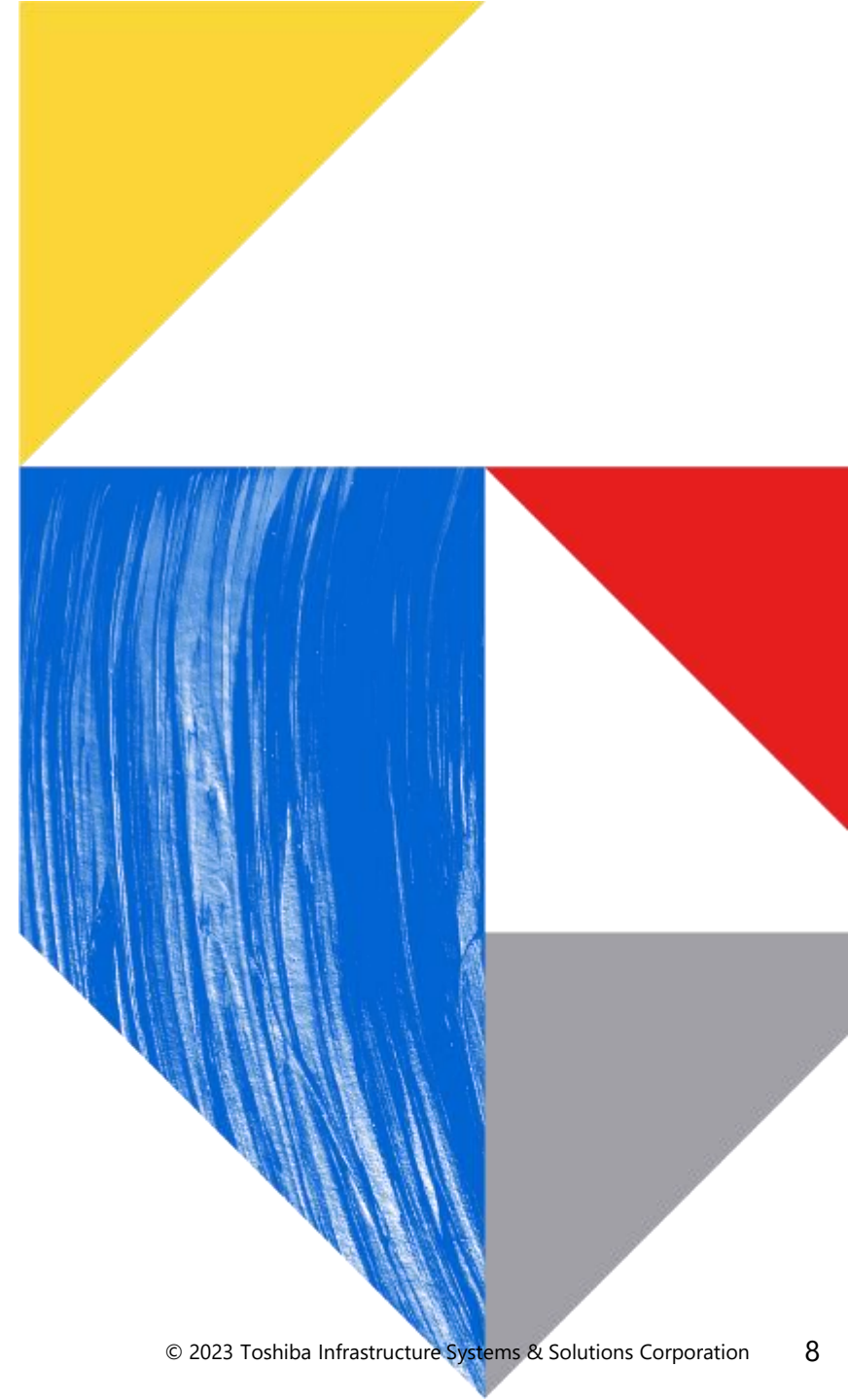
## HILS検証の導入

システム検証からの手戻りを抑制するため、  
システム結合検証の実施範囲を広げて事前に検証する。

複数ユニット結合後のソフトウェア検証  
フィードバック制御の検証 } システム結合検証に  
前倒して実施する

# 02

## HILS検証の導入・課題（2008年～）



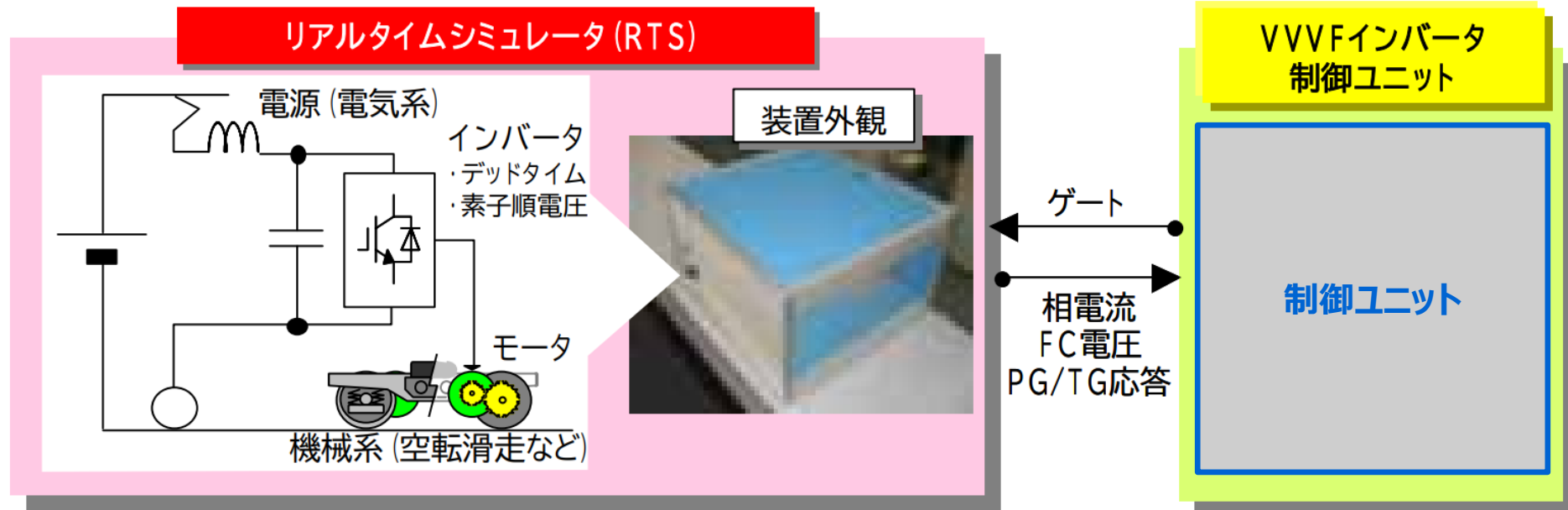


# HILS検証導入（第一ステップ：2008年～）

内製RTSを開発、シーケンス/フィードバック制御のデバッグやゲイン調整に活用。  
HILS検証が浸透したケース/展開できなかったケースがみられた。

## 内製RTSの概要

- ・数kHz以上の高調波電流の再現のため、モータやインバータなど電気モデルをFPGAに実装
- ・電気モデル演算周期：約1 $\mu$ s



結城、他、「鉄道車両ドライブ装置向けリアルタイムシミュレータの開発」、平成20年電気学会全国大会（2008）

# HILS検証環境導入（第一ステップ）で得られた効果・知見

HILS検証の導入によって「システム検証の負荷低減」に貢献。  
初期コストの低いRTSを多数台導入することで検証作業を効率化。

## メリット

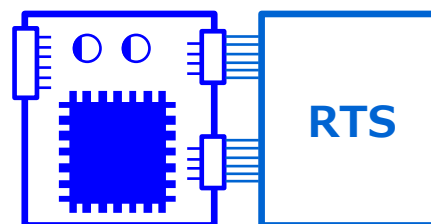
### システム検証の負荷低減に貢献

- ・シーケンス/フィードバック制御のデバッグ、ゲイン調整
- ・いじわる試験に対するソフトウェアの事前検証
- ・システム検証におけるトラブル早期解決の補助  
(システム検証と並行したデバッグ作業が可能)

### RTS多数台導入による検証効率化

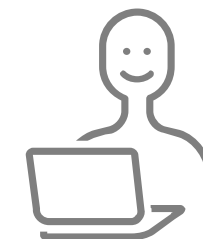
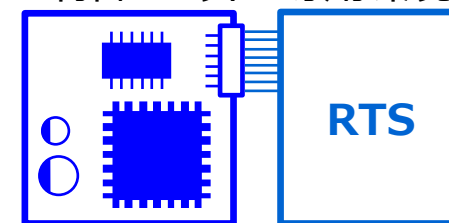
- ・制御ユニット専用の検証環境を用意、  
並行して複数の検証作業が実施可能
- ・検証環境を長期間保管でき、立上げ作業を省力化  
→初期コストの低い内製RTSの多数台導入で実現

制御ユニットA専用環境



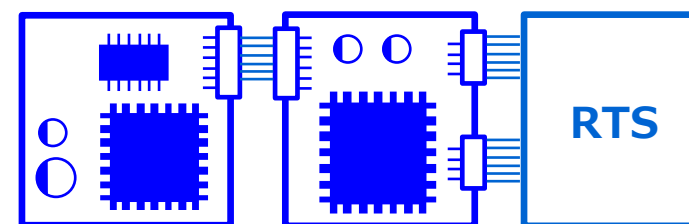
システム検証の前にソフトウェアの動的な挙動まで確認しておこう！

制御ユニットB専用環境



制御ユニット専用環境があるから、スムーズに検証に着手できる！

制御ユニットC専用環境



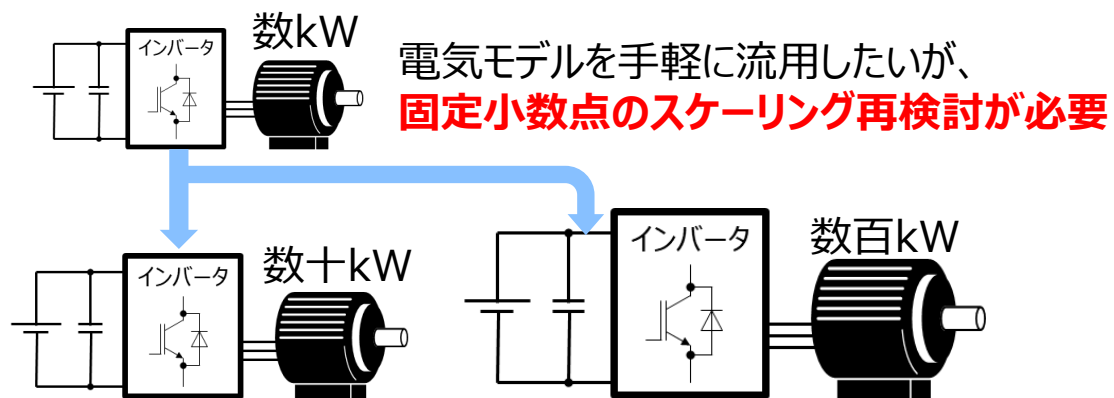
# HILS検証環境導入（第一ステップ）でみえた課題・要望1

## ユーザー（＝ソフトウェア設計者）による電気モデル（FPGA）の開発が長期化。

### HILS検証導入の阻害要因

#### 電気モデルの開発が長期化

- ①モデルの流用に手間がかかる（スケーリング再検討）
- ②新規モデル開発がシステム結合検証に間に合わない



#### 仕様・アルゴリズム検討

回路方程式や  
モータの電圧方程式から  
アルゴリズムを検討

#### モデル検討

- ・スケーリング検討
- ・HDLコーディング

ソフトウェア設計者には  
馴染みが薄い...  
HDLコード?  
RTLシミュレーション?



#### 論理シミュレーション

- ・テストベンチ作成
- ・論理シミュレーション

- ・コーディングミスによる手戻り
- ・テストベンチ作成

#### 実装検証/ リリース



## モデル仕様書の整備だけでは、電気モデルの動的挙動が把握しにくい。

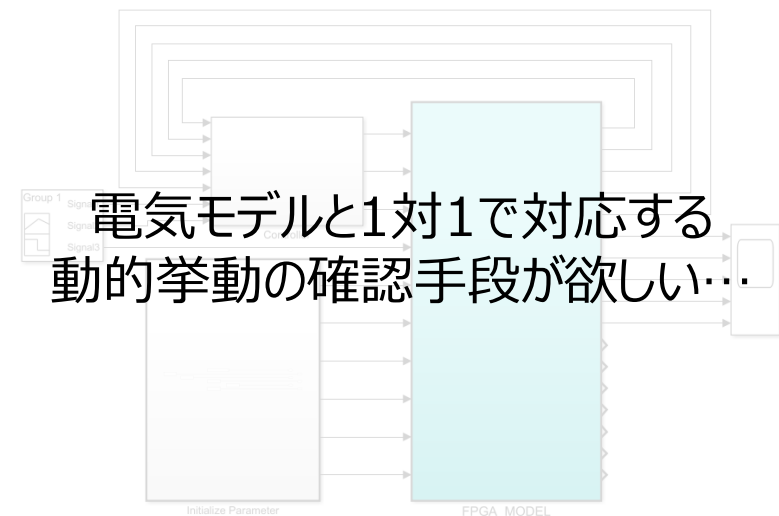
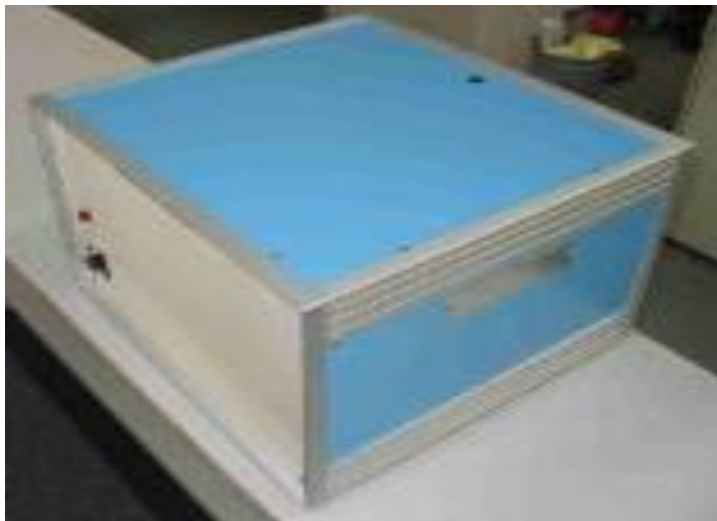
### HILS検証導入の阻害要因

#### 電気モデルの挙動把握が困難

- ③モデル仕様書とHDLコードだけでは挙動が把握しにくい  
(派生や改良が重なり、モデルの全容把握が困難)



RTSを使う前に挙動把握したいが仕様書やHDLコードだけでは動的な挙動が読み解きにくい…



# 第一ステップの課題解決に向けて

電気モデルに対する課題・要望の解決策として、  
第二ステップでは開発環境改善と浮動小数点設計の採用を決定した。



FPGA設計ツールに不慣れ

## HILS検証導入の阻害要因

電気モデルの開発が長期化

電気モデルの挙動把握が困難



## 開発環境改善（MATLAB/Simulink活用）

### シミュレーション検証

- ・モデル開発上流からのシミュレーション検証による開発効率向上
- ・動的挙動の確認手段

### HDLコード生成

- ・ブロック図記述による可読性向上
- ・コーディング不要、コーディングミス抑制

## モデルの流用簡便化

### 浮動小数点設計の採用

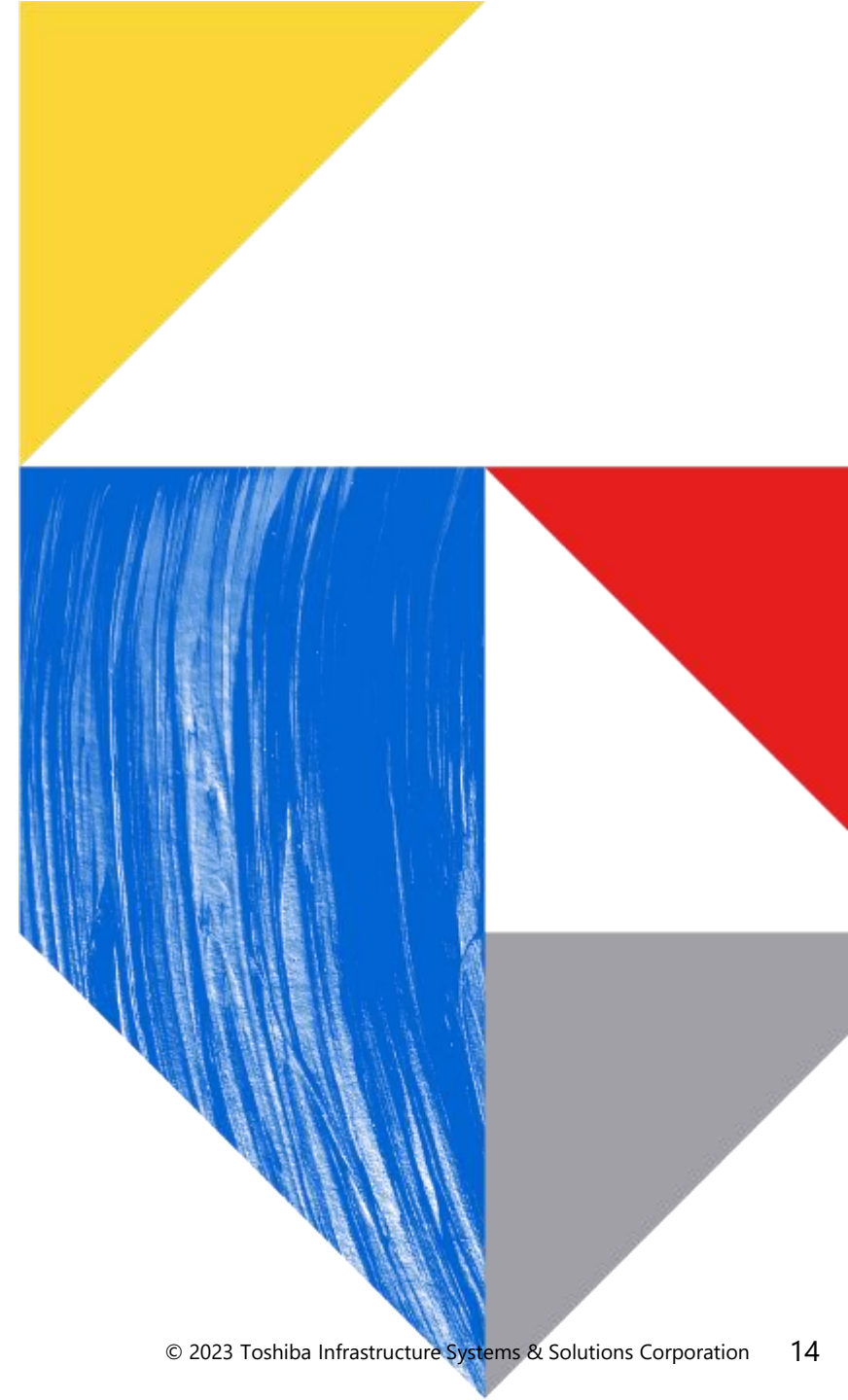
- ・流用時の固定小数点のスケーリング再検討を低減



ソフトウェア設計ツールのデファクトスタンダードで扱い易い

# 03

## 市販SoC/FPGAを活用したRTS開発



## 第二ステップ：HILS検証環境の改善方針

### 1 多数台導入による 検証効率化

- ・市販製品の組み合わせによる  
RTS開発要素の絞り込み
- ・初期コストの低減

#### RTSの概要を紹介



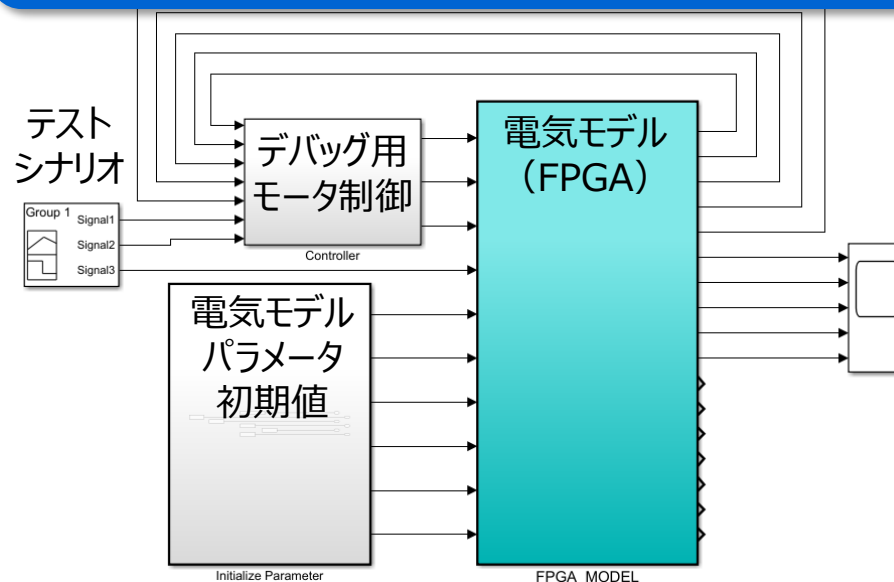
### 2 電気モデルの 開発環境改善

- ・シミュレーション検証による  
電気モデル開発の効率化
- ・HDLコード生成による  
コーディングミス抑制、手戻り低減

### 3 電気モデルの 流用性向上

- ・浮動小数点設計を採用し、  
スケーリング検討の省力化

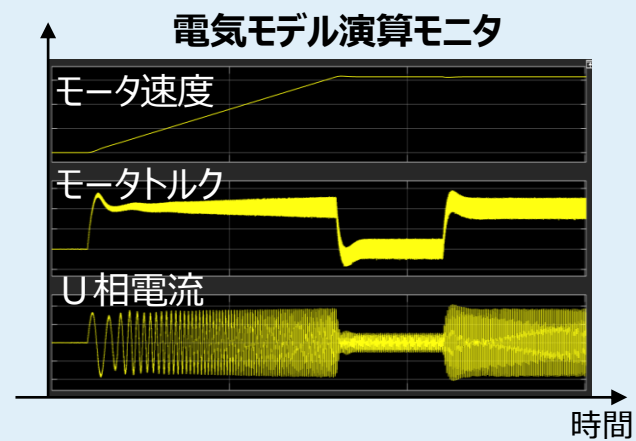
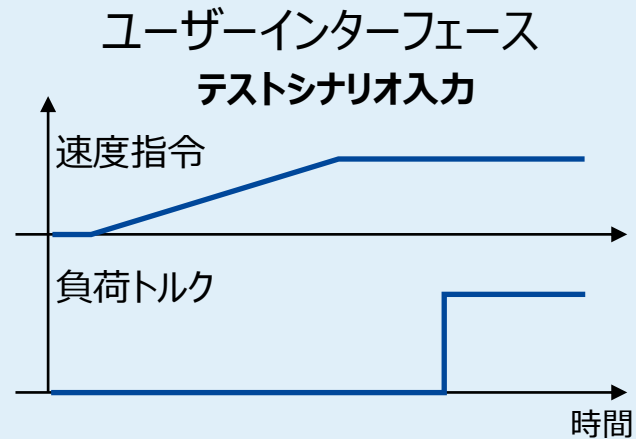
#### 電気モデル開発フローと工数低減効果を紹介



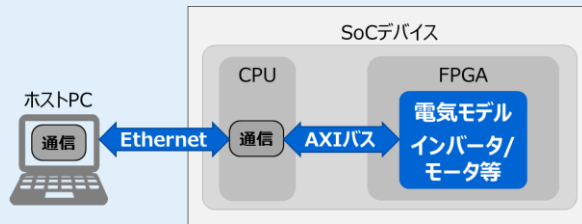
## 第二ステップ：RTSシステム概要

コンセプト：市販製品を組み合わせることで自前開発要素を最小限に抑制

### MATLAB/Simulink



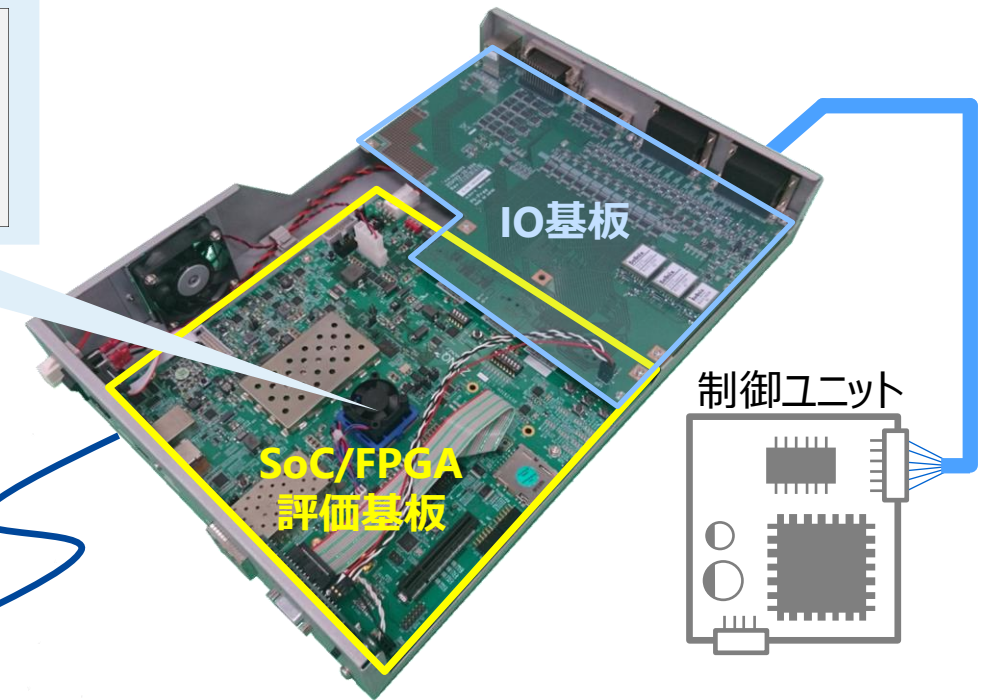
### ホストPC⇔CPU/FPGA通信



### ホストPC



### 市販SoC/FPGA評価基板





# RTSのシステム構成要素

基本的な構成要素5つのうち、**ユーザーに依存する3つの要素の開発に注力したい。**  
「CPU・FPGA基板」、「ホストPC⇔CPU/FPGA通信」は市販製品を活用。

## 構成要素 (ハードウェア)

CPU・FPGA基板

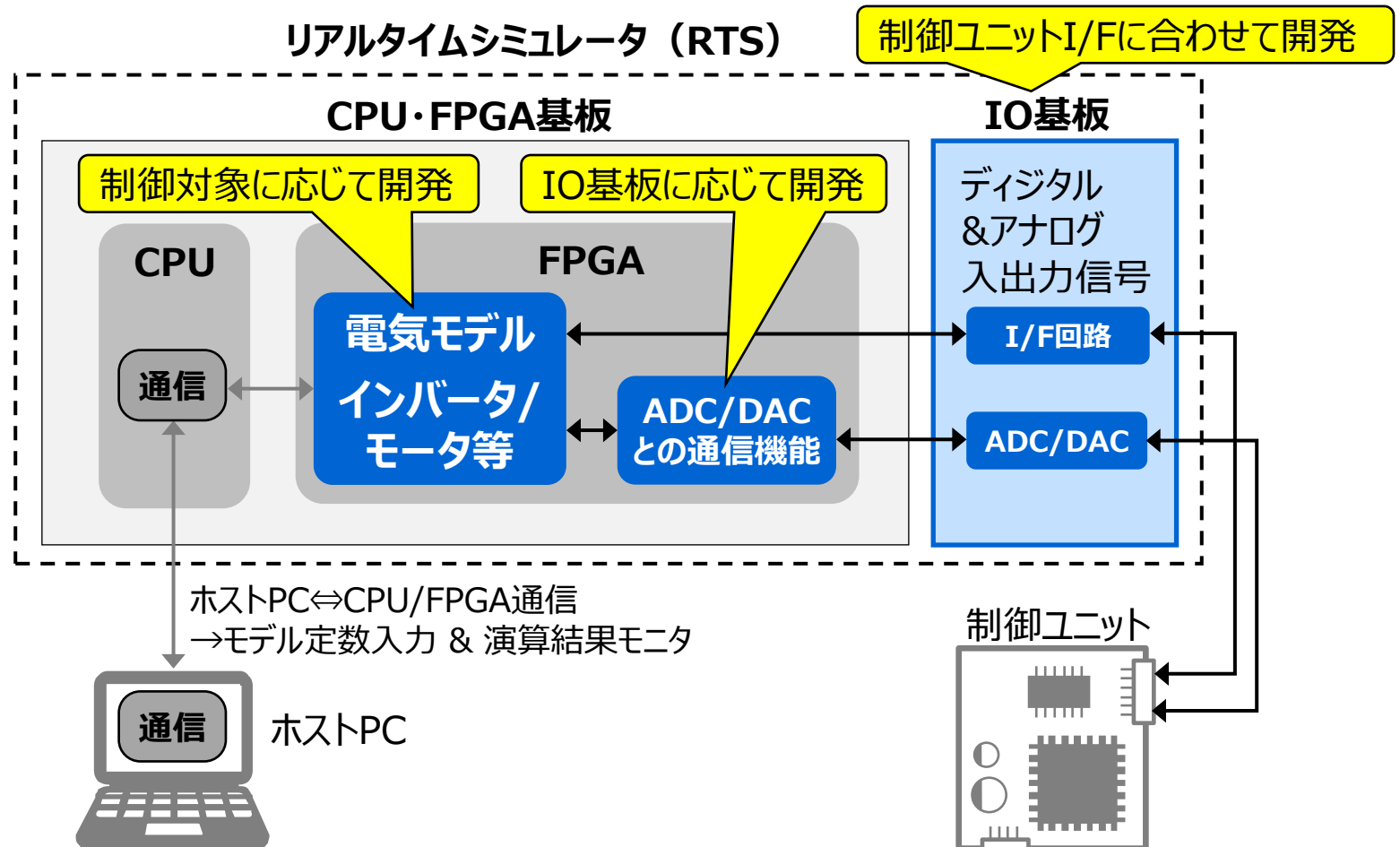
IO基板  
(デジタル/アナログ入出力)

## 構成要素 (ソフトウェア)

電気モデル

IO基板との通信

ホストPC⇔CPU/FPGA通信



## Embedded Coder<sup>®</sup>/HDL Coder<sup>™</sup>のSupport Packageに対応した市販のSoC/FPGA評価基板を使うことで、構成要素2つを準備できる。

### 構成要素 (ハードウェア)

CPU・FPGA基板

IO基板  
(デジタル/アナログ入出力)

### 構成要素 (ソフトウェア)

電気モデル

IO基板との通信

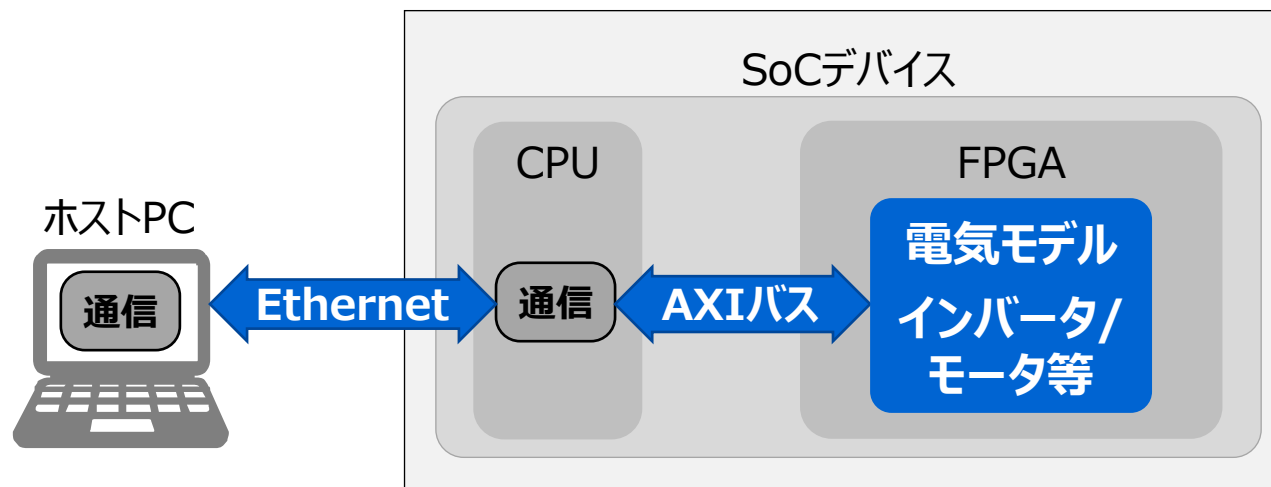
Host PC ↔ CPU/FPGA通信

Support Package対応評価基板であれば、

- ・SoC/FPGA基板開発を省略できる
- ・Host PC ↔ CPU/FPGA通信も準備済み、開発を省略できる

→ **利用者に依存する3つの要素の開発に注力できる。**

### 例：SoC評価基板



# 本開発で使用した評価基板 Zynq™ UltraScale+ MPSoC ZCU102

Support Package対応およびハードウェアリソース要求仕様から、評価基板を選定。  
IO基板でアナログ入出力機能を補い、RTSとして構築した。

## 構成要素 (ハードウェア)

CPU・FPGA基板

IO基板  
(デジタル/アナログ入出力)

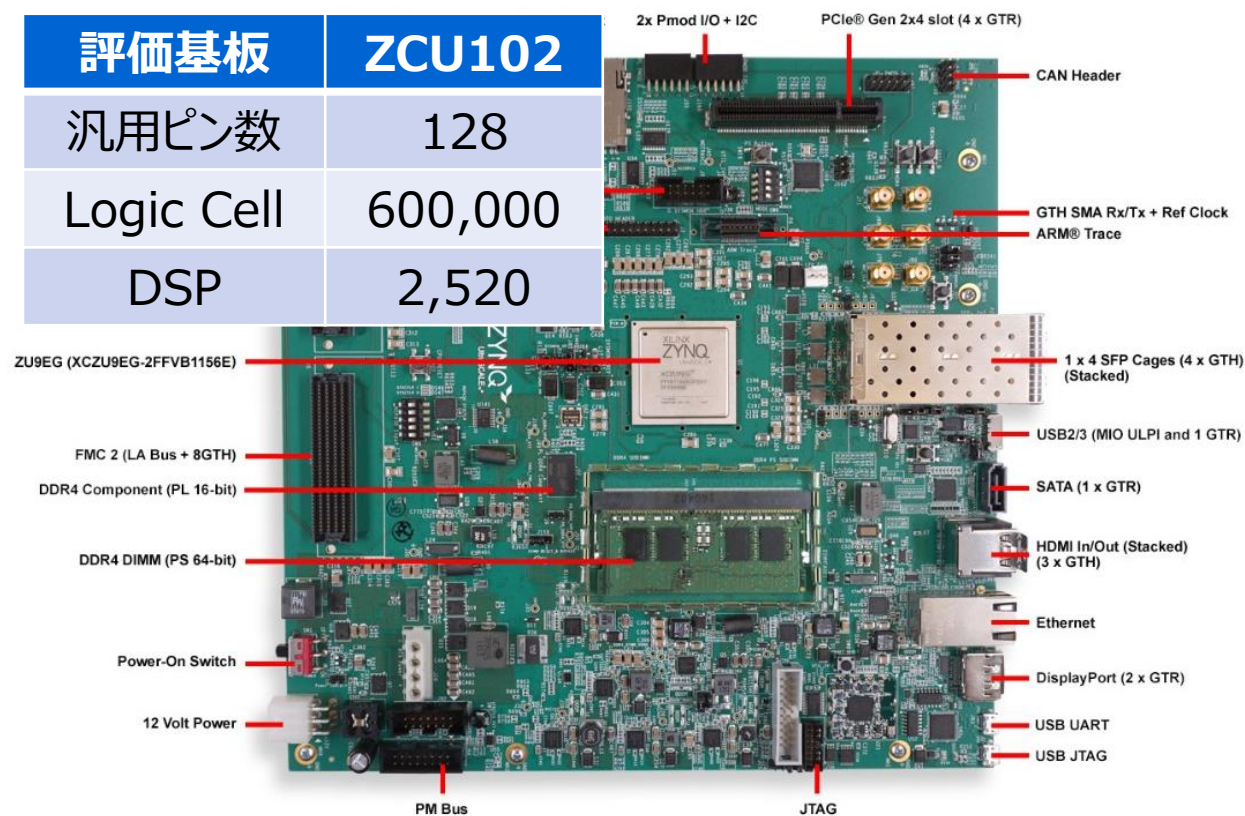
## 構成要素 (ソフトウェア)

電気モデル

IO基板との通信

ホストPC⇔CPU/FPGA通信

| 評価基板       | ZCU102  |
|------------|---------|
| 汎用ピン数      | 128     |
| Logic Cell | 600,000 |
| DSP        | 2,520   |



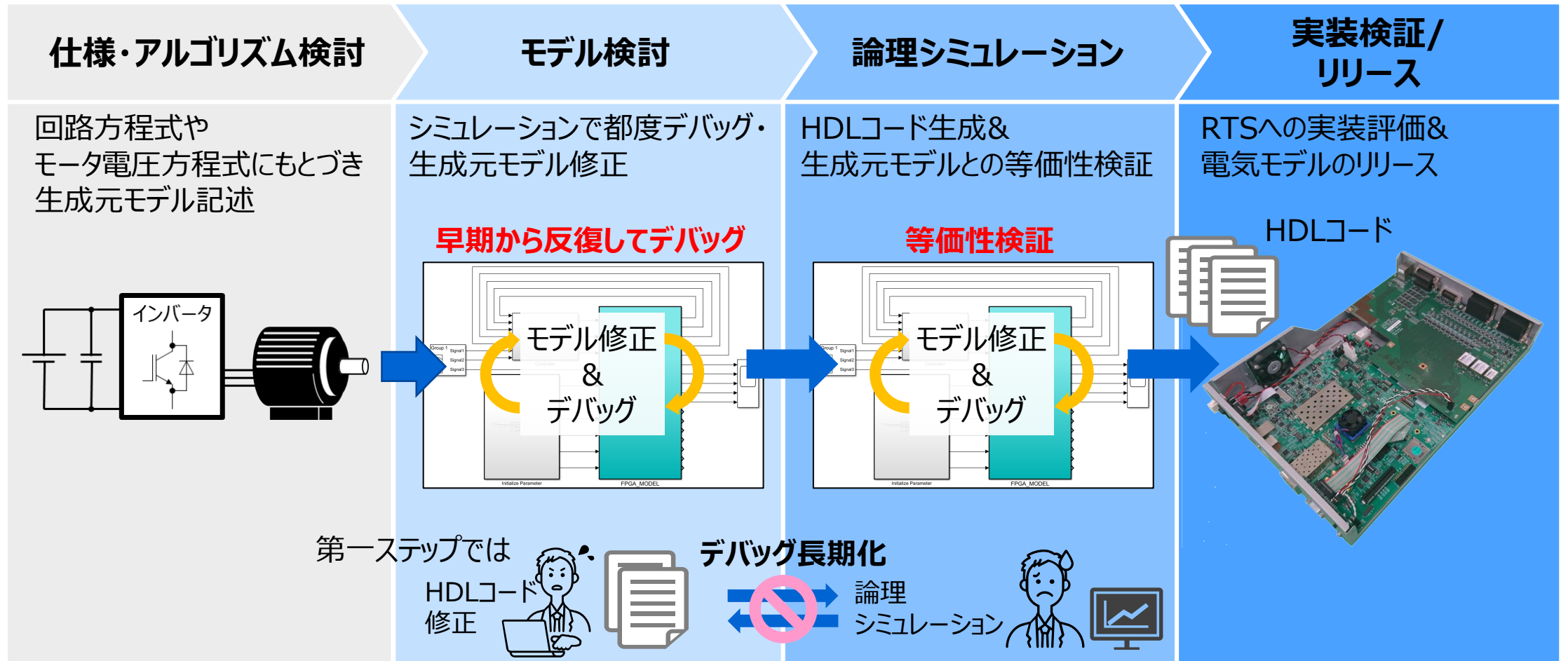
出典: <https://japan.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html>

# 04

## MATLAB/Simulinkを活用した電気モデル開発

# 電気モデル開発における開発環境改善のねらい

シミュレーションを用いた早期デバッグによって開発効率を向上。  
HDLコード生成によるコーディングミスと手戻りの低減。



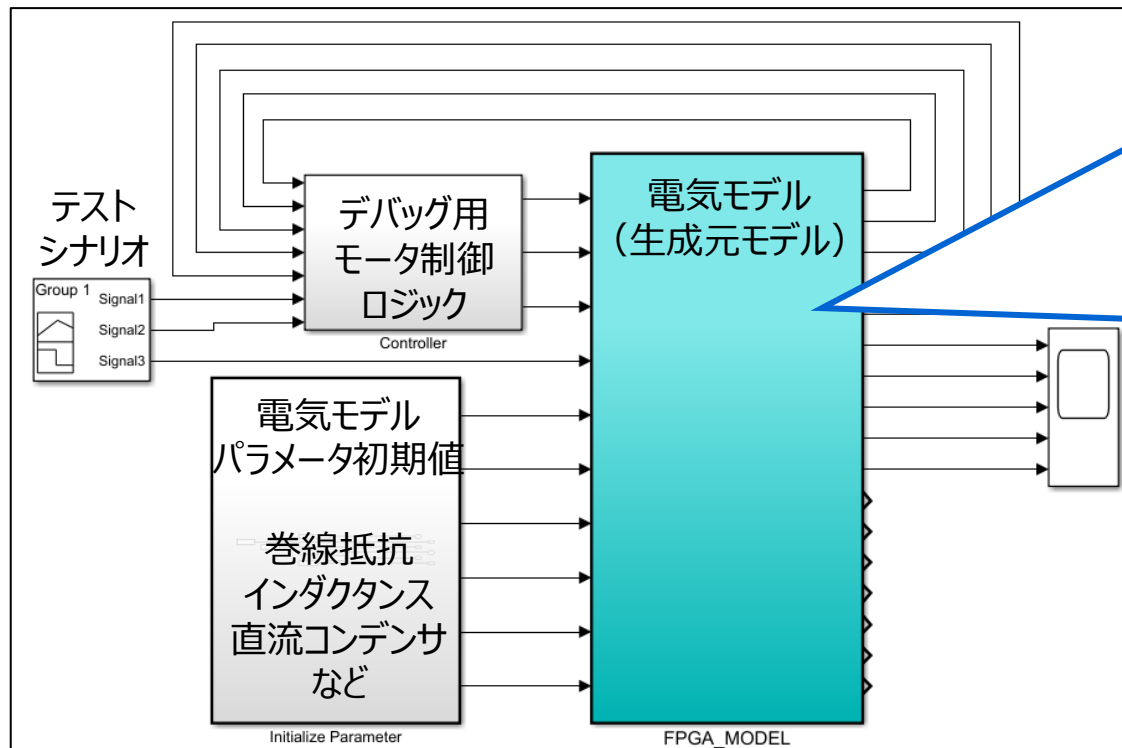
**HDLコード生成のため電気モデルをSimulinkブロックで記述。  
流用性向上のために浮動小数点で設計、演算誤差を検討する。**

仕様・アルゴリズム検討

モデル検討  
(生成元モデル)

論理シミュレーション  
(HDL生成モデル)

実装検証/  
リリース



- **HDL Coder™対応のモデル記述 (生成元モデル)**  
HDLコード生成対応のSimulinkブロックで回路方程式やモータの電圧方程式を記述。



- **浮動小数点演算の検討**  
浮動小数点演算 (IEEE 754準拠) でモデル検討、情報落ちや変数型など演算誤差のケアを実施。

**シミュレーションで都度検証、確認が可能**

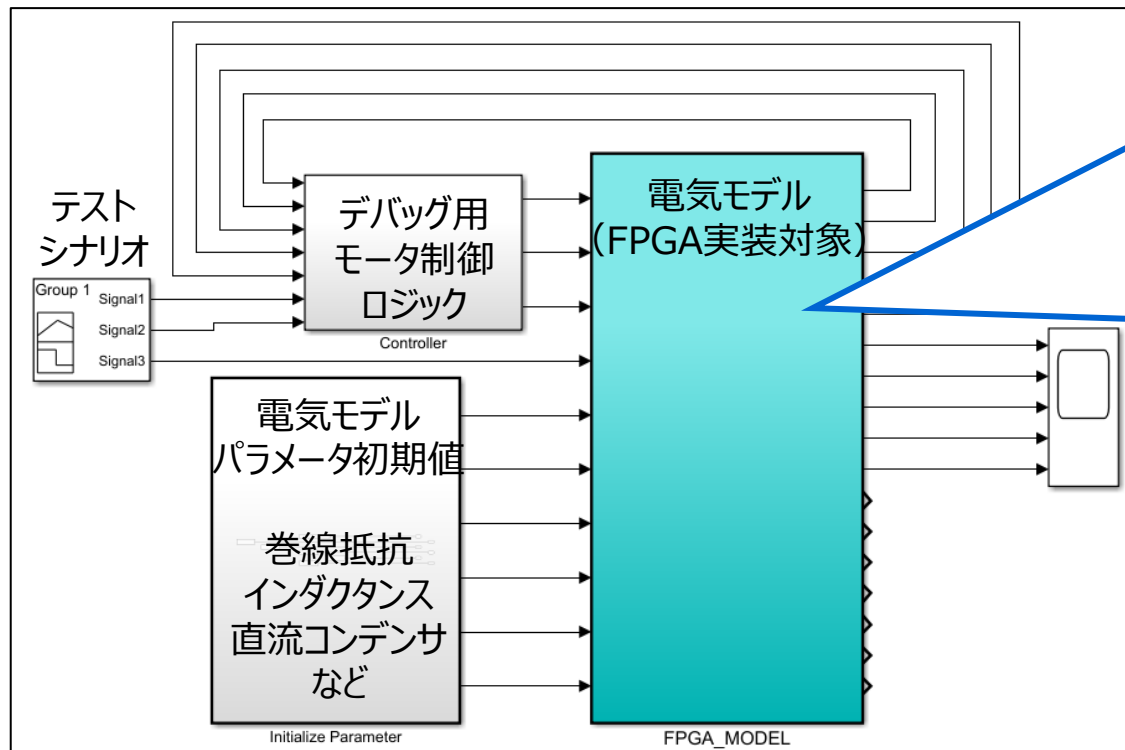
## シミュレーションを用いた早期デバッグによる開発効率化 & HDLコード生成によるコーディングミスの抑制

仕様・アルゴリズム検討

モデル検討  
(生成元モデル)

論理シミュレーション  
(HDL生成モデル)

実装検証/  
リリース



### • FPGAクロック、パイプライン検討

電気モデルの演算周期と演算子に必要なレイテンシ、パイプライン数にもとづき、FPGAクロックを検討。

### HDLコード生成

(コードと等価なHDL生成モデルを自動生成)

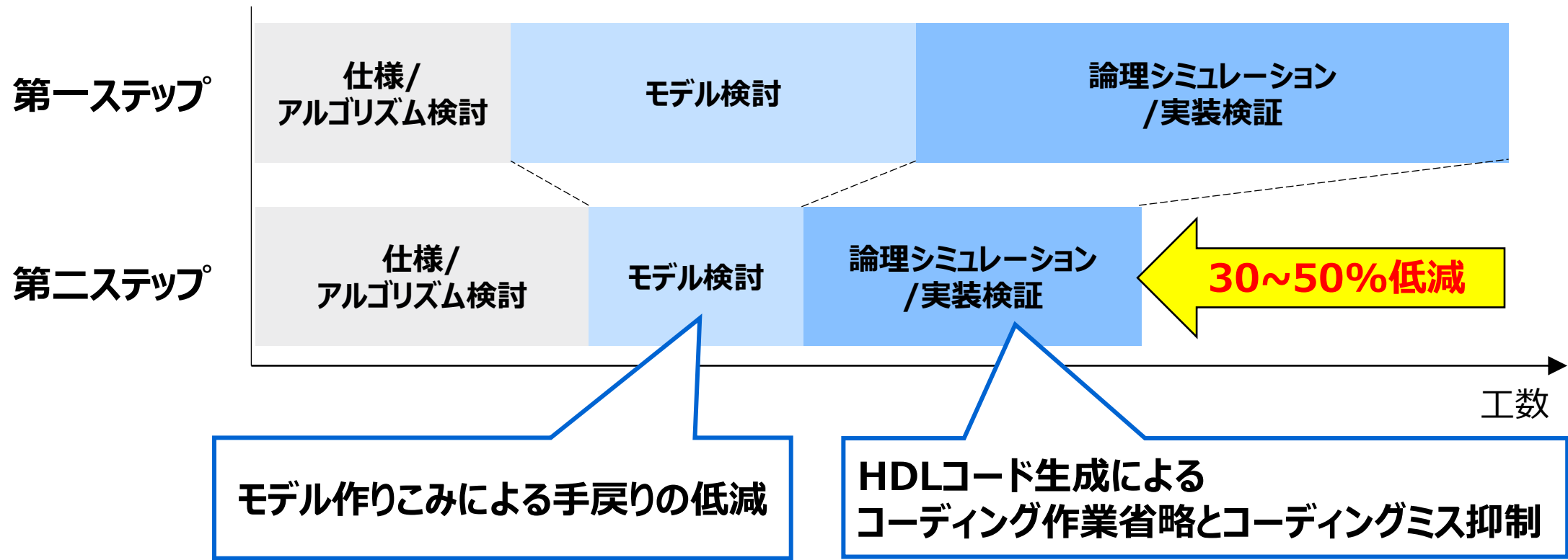
### • 等価性検証

パイプライン自動挿入の影響を確認するため、生成元モデルとHDL生成モデルの誤差を検証。

等価性確認もシミュレーションで実施

# 電気モデル構築における工数低減効果

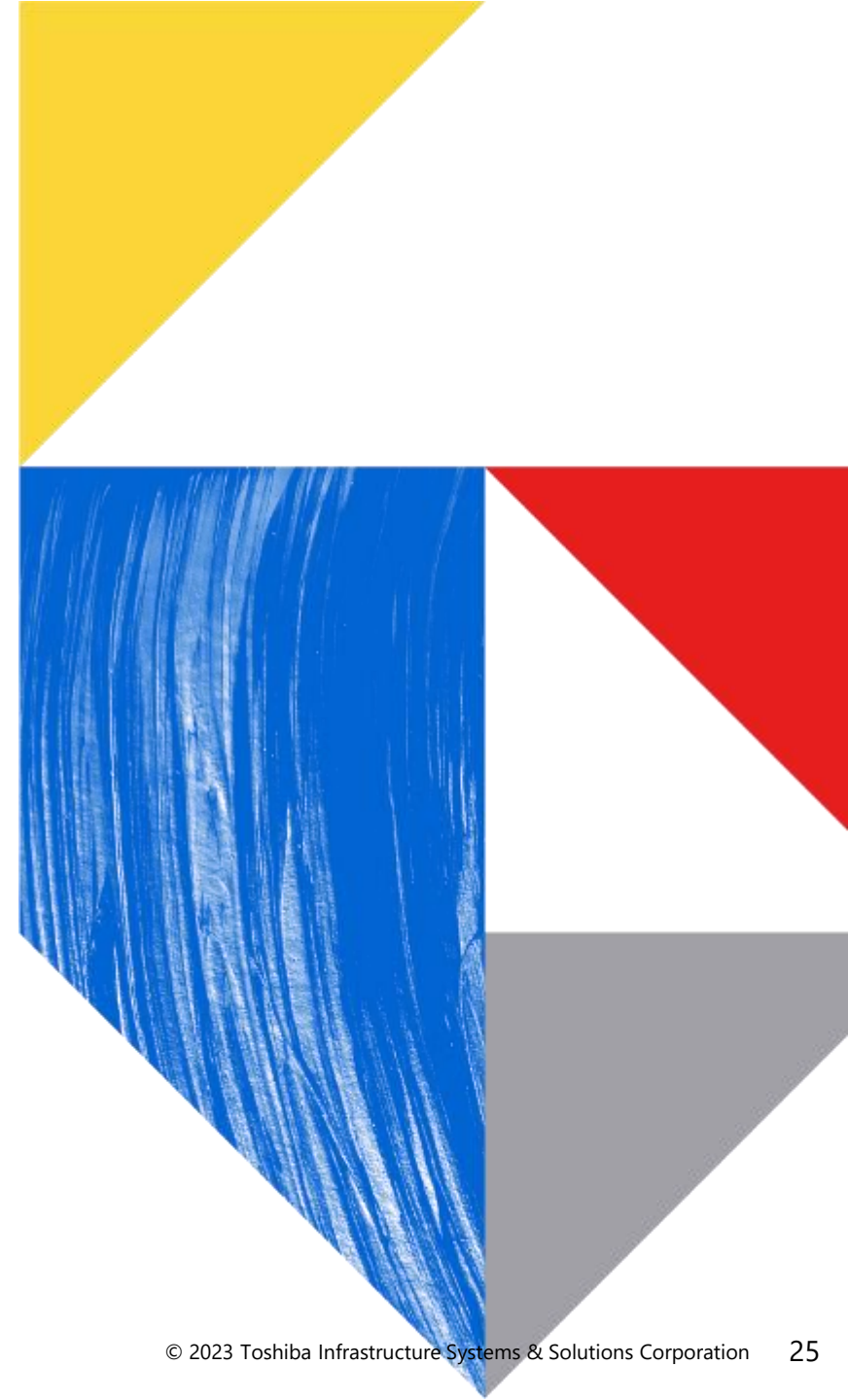
シミュレーションによるモデルの作りこみと、HDLコード生成を活用することで、  
30～50%の工数削減を実現。





# 05

まとめ・今後の展開



## 1 多数台導入による 検証効率化

- ・市販製品の組み合わせによる  
RTS開発要素の絞り込み
- ・初期コストの低減

## 2 電気モデルの 開発環境改善

- ・シミュレーション検証による  
電気モデル開発の効率化
- ・HDLコード生成による  
コーディング不要、ミス抑制

## 3 電気モデルの 流用性向上

- ・浮動小数点設計を採用し、  
スケーリング検討の省力化

- ✓ 3つの方針にそってHILS検証環境の改善を実施し、RTSおよびモデル開発環境を整備した。
- ✓ 「電気モデルの開発長期化（第一ステップ）」に対して、30～50%の工数低減効果が得られた。

**HILS検証導入を促進し、パワエレ制御ソフトウェアのさらなる品質向上に貢献します。**

# HDLコード生成への期待

さらなるモデル開発工数の低減に向けて、MATLAB/SimulinkのHDLコード生成に期待すること。

仕様・アルゴリズム検討

モデル検討  
(生成元モデル)

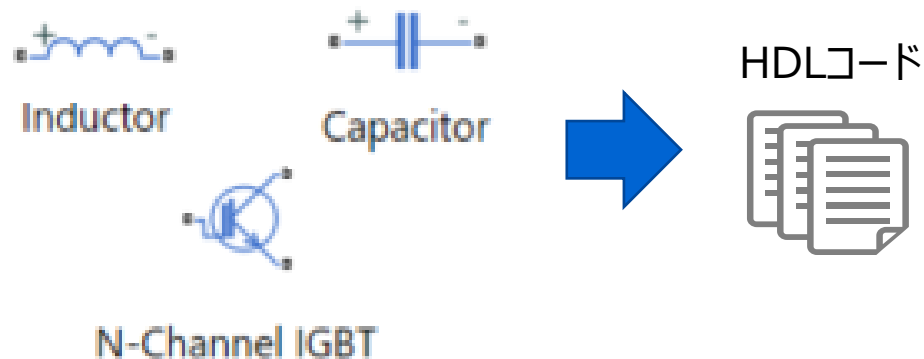
論理シミュレーション  
(HDL生成モデル)

実装検証/  
リリース

## 1. Simscape™によるHDLコード生成

- ✓ Simscapeブロックを用いて回路図記述のような直感的なモデル構築。
- ✓ SimscapeブロックからのHDLコード生成。

モータドライブなどの事例の拡充を期待。



## 2. 生成元モデルへの遅延自動挿入機能

- ✓ 電気モデルは積分器を多く含み、Simulinkブロックのフィードバックループが点在。
- ✓ HDL Coder™の制約上、フィードバックループには遅延自動挿入が非適用。
- ✓ 人手での遅延ブロック設置に労力がかかる。

遅延ブロック設置を補助する機能追加に期待。

**TOSHIBA**