

MATLAB EXPO 2018 Japan

無線モデムFPGA/SoC開発における HDL Coder™の活用事例

2018/10/30

NECネットワーク・センサ株式会社

技術開発本部 通信ネットワーク技術部

主任/プロダクトスペシャリスト

住田 憲昭

\Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

目次

1. 会社紹介/自己紹介
2. 防衛事業の技術課題とHDL Coder導入経緯
3. 無線機構成と開発ツール
4. 従来型開発プロセスとその問題点
5. HDL Coder導入後の開発プロセスとその効果
6. HDL Coder設計結果の例
7. 留意すべき点
8. 所感と今後の展開

1.1. 会社紹介

■ NECネットワーク・センサ株式会社

- 設立：1999年（NEC100%出資）
- 本社：東京都府中市
- 日高事業所：埼玉県日高市
- 相模原事業所：神奈川県相模原市
- 白石事業所：宮城県白石市

■ NECグループの主に航空・宇宙・防衛の事業の上流設計～詳細設計、ものづくり～メンテナンスまで、幅広い役割を担う

■ 防衛用通信機器の近代化に貢献

■ 学生求む！

- <https://www.necnets.co.jp/>
- <https://job.rikunabi.com/2019/company/r749500017/>
（写真：航空自衛隊、海上自衛隊サイトより引用）



1.2. 自己紹介

主に防衛用・公共用無線通信機器の設計

無線機アーキテクチャ/変復調アルゴリズム/ハードウェア/FPGA設計

MATLAB/Simulink・FPGA使用歴

	～2006	2007～2012	2013	2014～
シミュレーション ツール・ FPGA設計環境	MATLAB/ Simulink + ハンドコード VHDL	MATLAB/Simulink + ISE System Generator 5製品/5波形出荷		MATLAB/Simulink + HDL Coder + Vivado System Generator 6製品/7波形出荷 4製品開発中
	ISE		Vivado	
使用デバイス	Xilinx Spartan-3E	Xilinx Virtex-5	Xilinx Artix-7	Xilinx Kintex-7 / Virtex-7 / Zynq-7000 Zynq Ultrascale+

2.1. 防衛通信事業の技術課題

■ 少量多品種

- 電氣的/構造的な標準化が難しい、大量生産でなくても対応可能な組織である必要

■ 多種多様な通信方式（独自規格）

- TDD / FDD, SCPC / TDMA / CDMA / CSMA/CA

■ 耐環境性能（温度/湿度/防水/衝撃/振動）

■ 民生通信トレンドとの乖離

- 民生：どんどん広帯域に
- 防衛/公共：民生に追従する広帯域化ニーズもあるも、狭帯域通信も根強い
 - ・ 狭帯域デジタル通信は、位相ノイズ特性に敏感なので、広帯域向けに最適化された部品が狭帯域で使えるとは限らない

■ 品質の担保（インターネット経由のアップデート不可）

■ 旧世代品との互換性（無線、有線）

- ドキュメント不足、市場からの部品/IP消滅、技術継承者不在

■ 短納期化/低コスト化（設計費/製品費）/小型化/低消費電力化

- 防衛事業においても、例外ではない

2.2. HDL Coder導入経緯

■ アルゴリズムからの新規開発案件がほとんど

- アルゴリズム開発に時間をかけ、無駄なコーディング時間を削減する必要性

■ FPGAの大規模化と機能統合により、1デバイス1担当者（社）での開発が限界に

- 担当者間の共通言語としてのSimulinkの有効活用

■ 顕著な短納期化

- 信号処理から制御回路まで、短期間で設計する必要

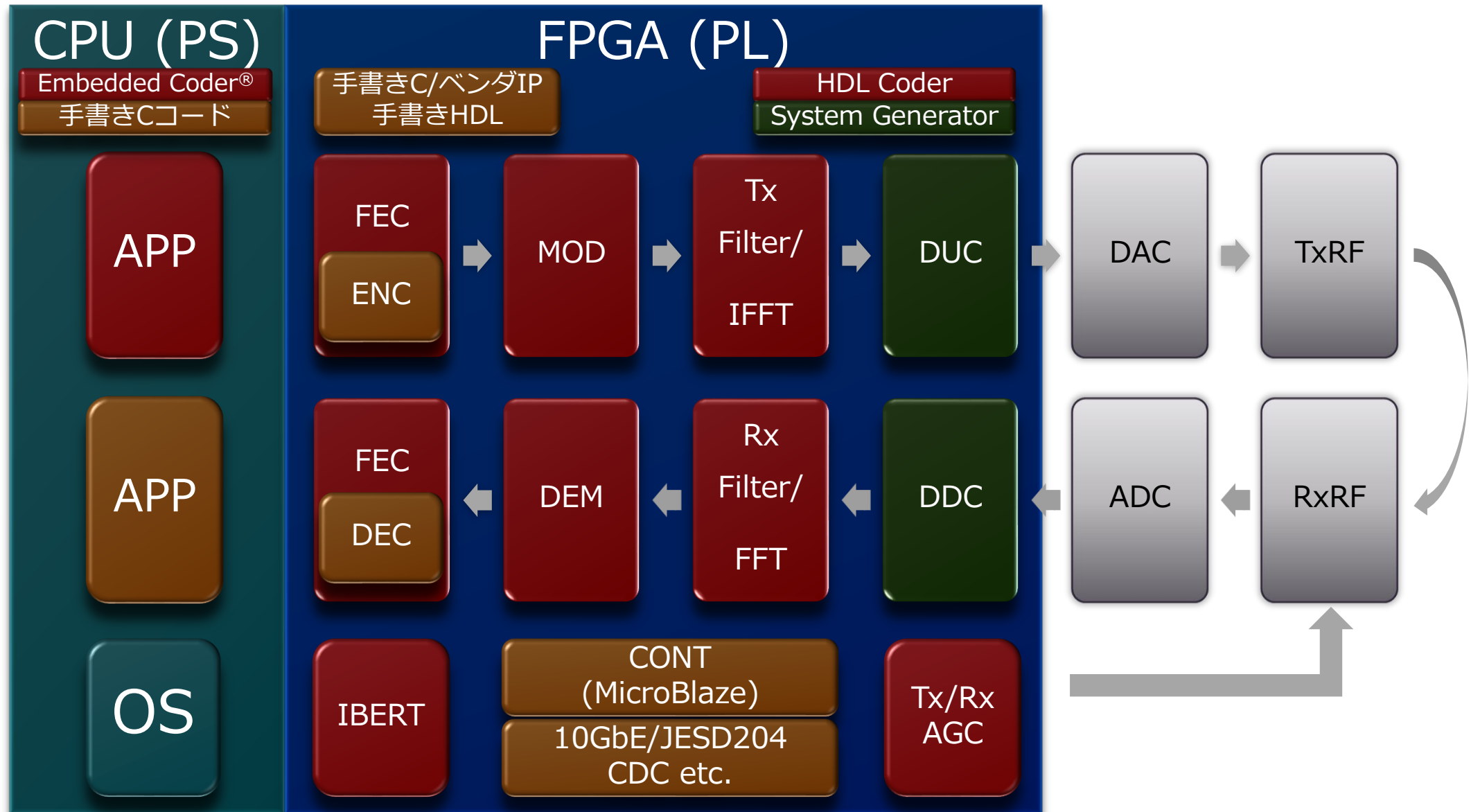
■ ターゲットデバイス依存性の低減

- S/WとH/Wの切り分けを柔軟にする必要性
- CPU/DSP⇔FPGAの移植性向上
- FPGAベンダ切替時の移植性向上

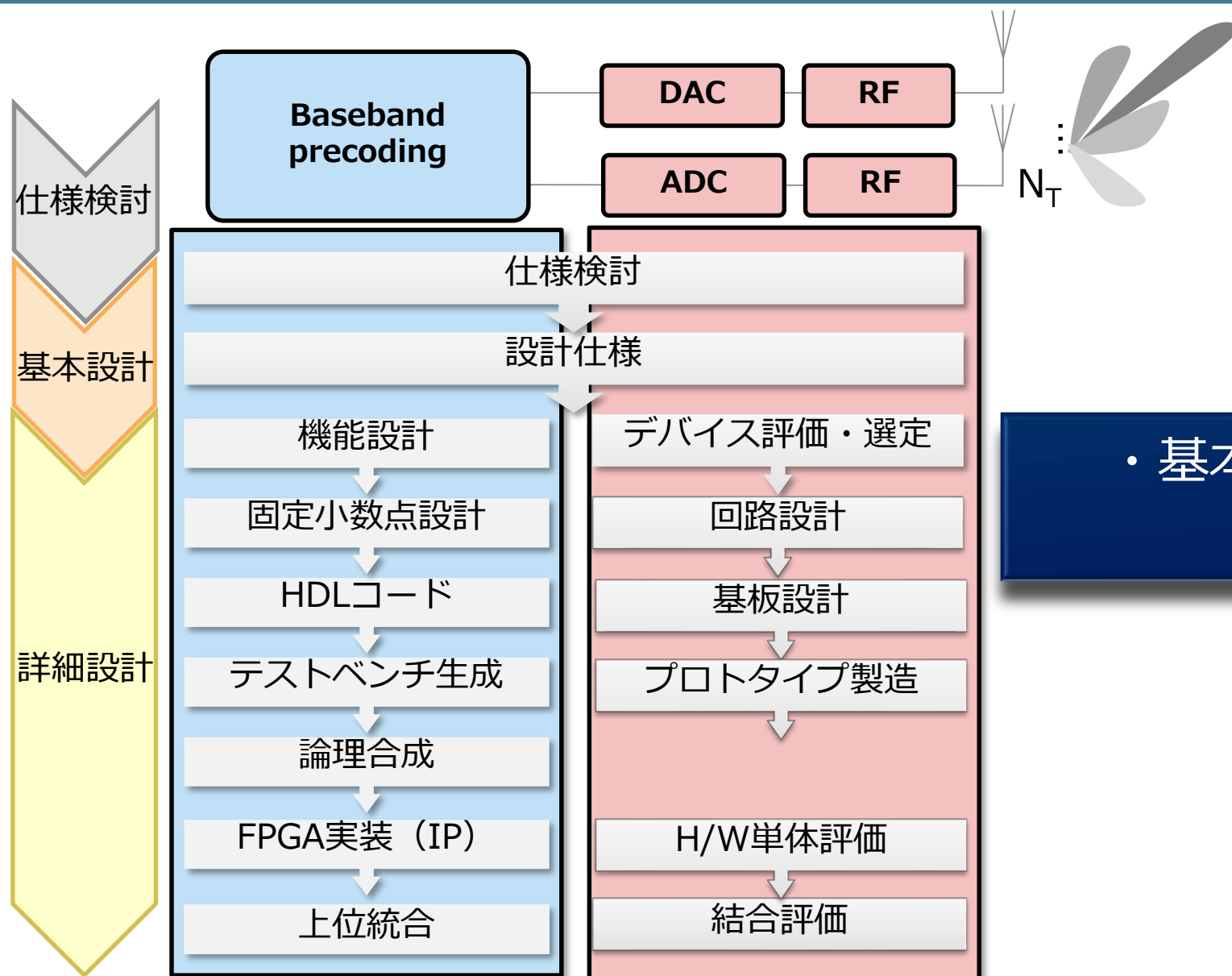


HDL Coderの導入

3. 無線機構成と開発ツール

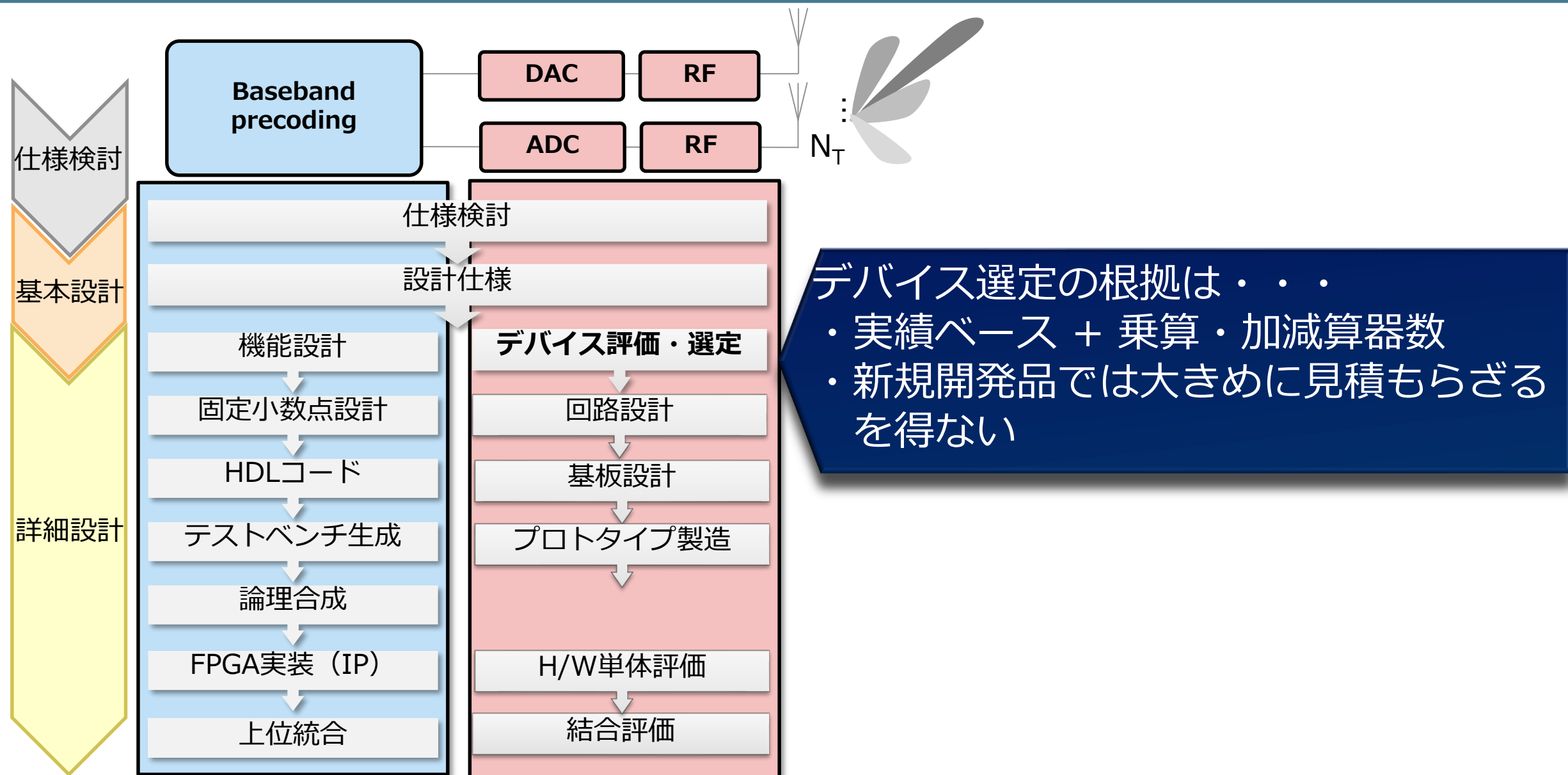


4.1. 従来型開発プロセス

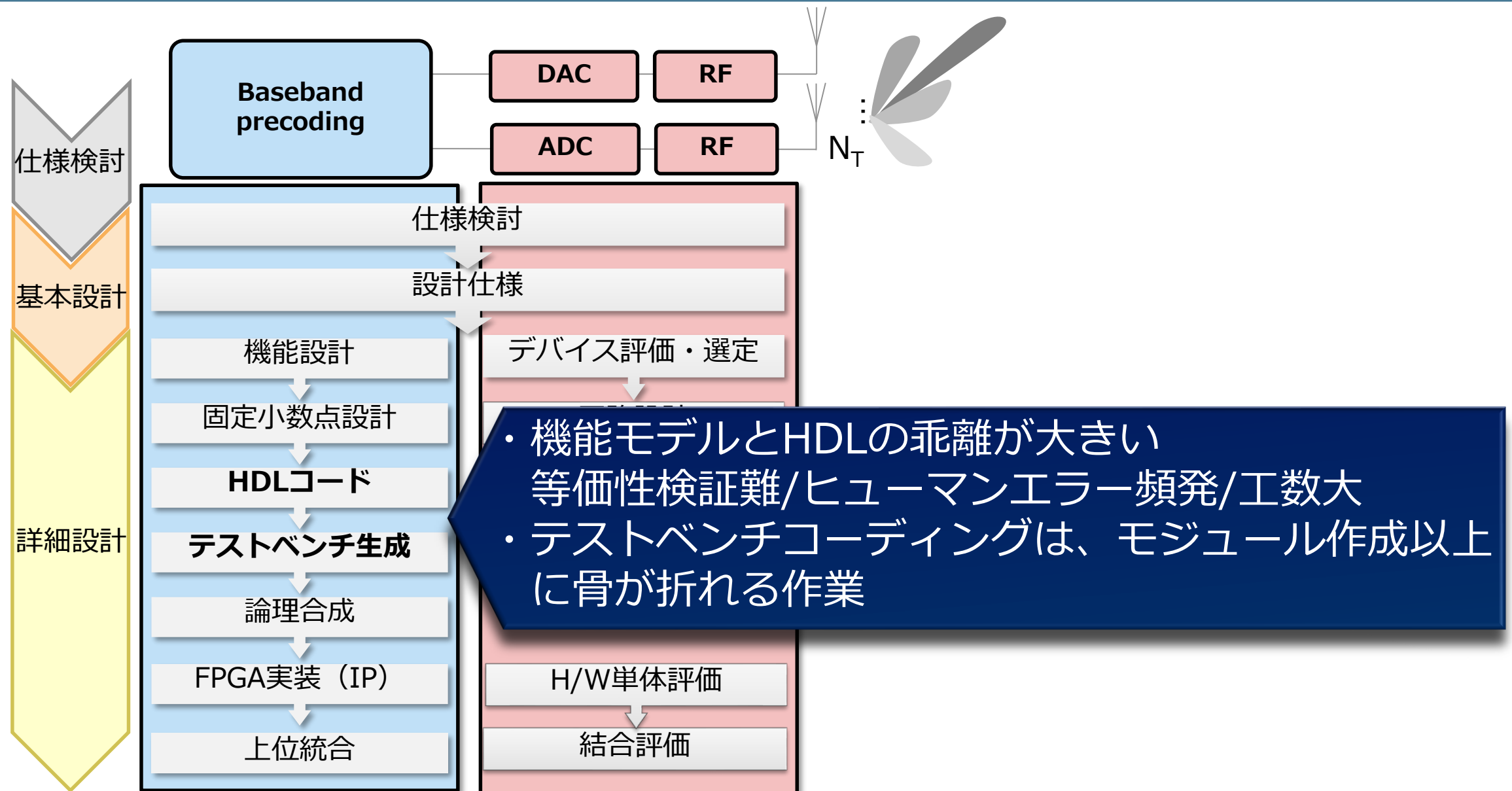


- ・ 基本的なウォーターフロー開発
- ・ ハンドコードに依存

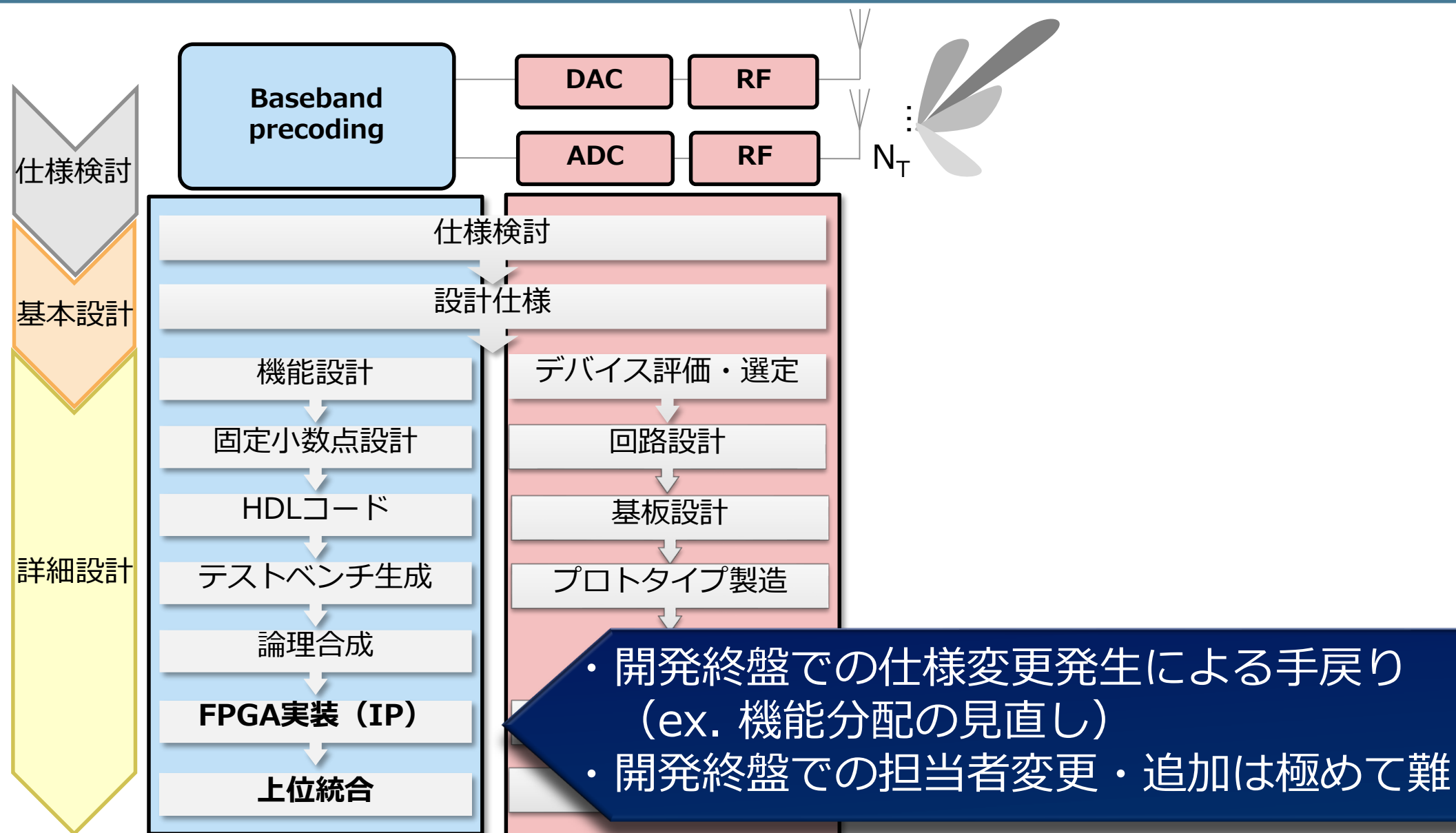
4.2. 従来型開発プロセスの問題点①



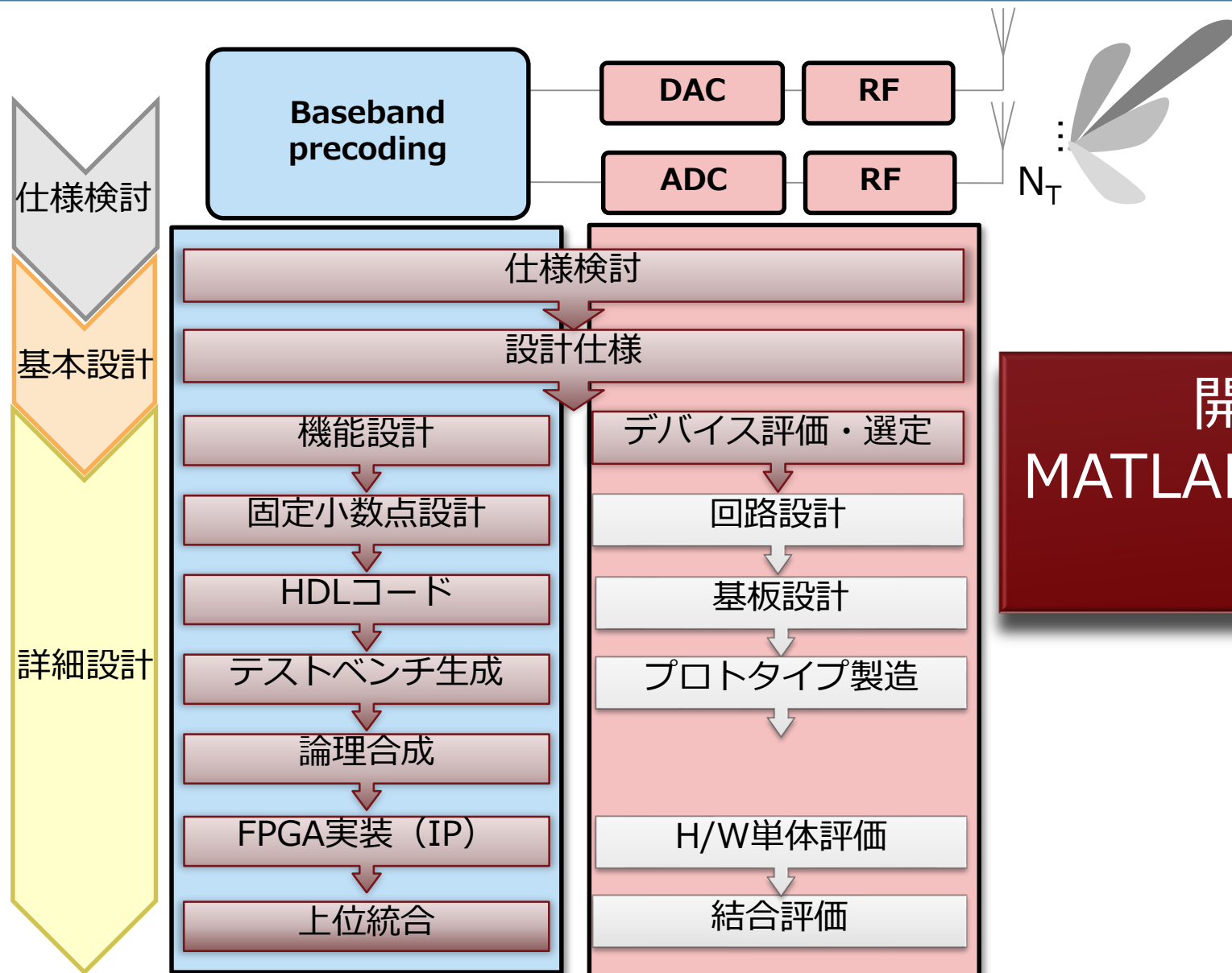
4.2. 従来型開発プロセスの問題点②



4.2. 従来型開発プロセスの問題点③

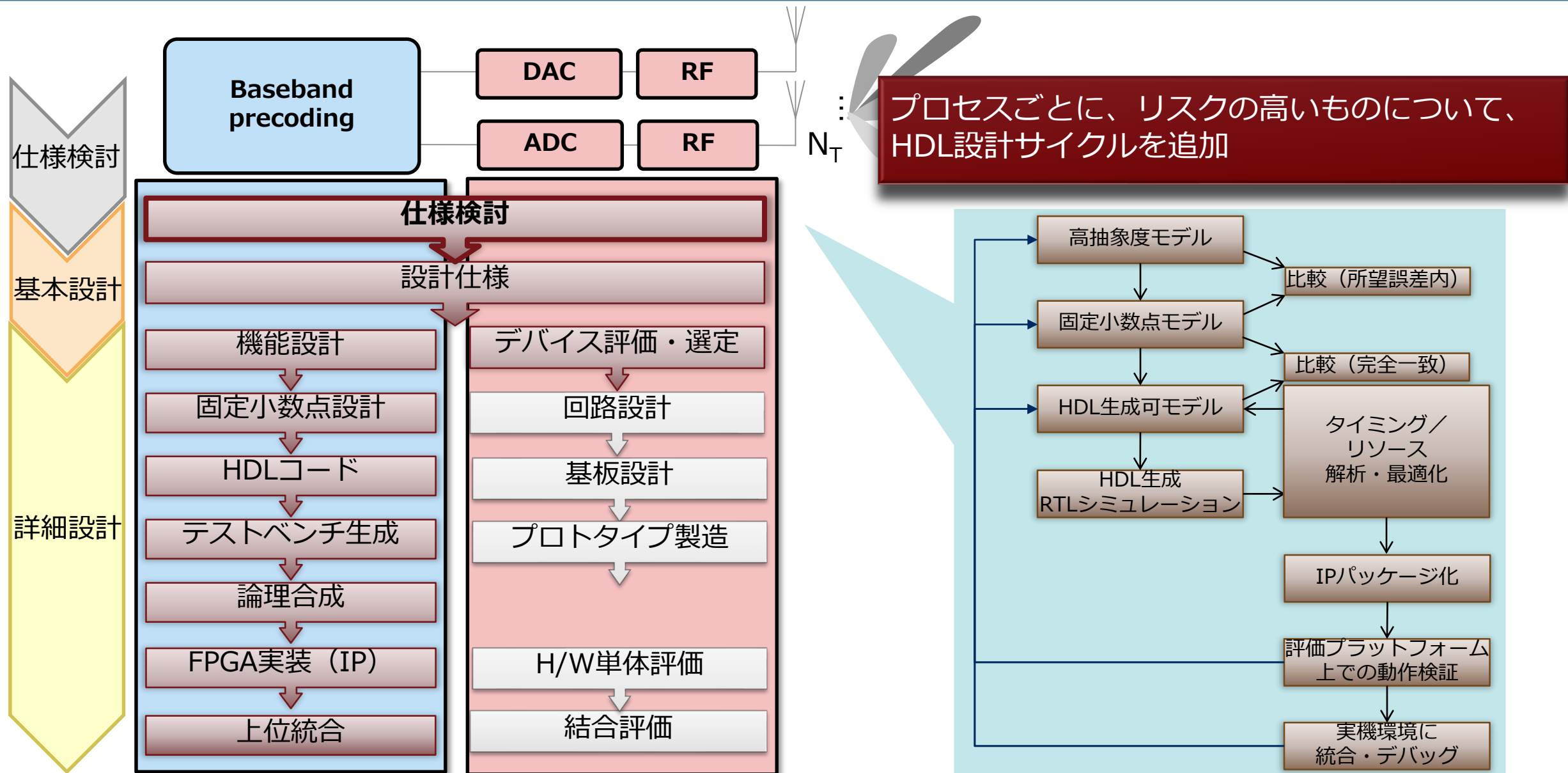


5.1. HDL Coder導入後の開発プロセス

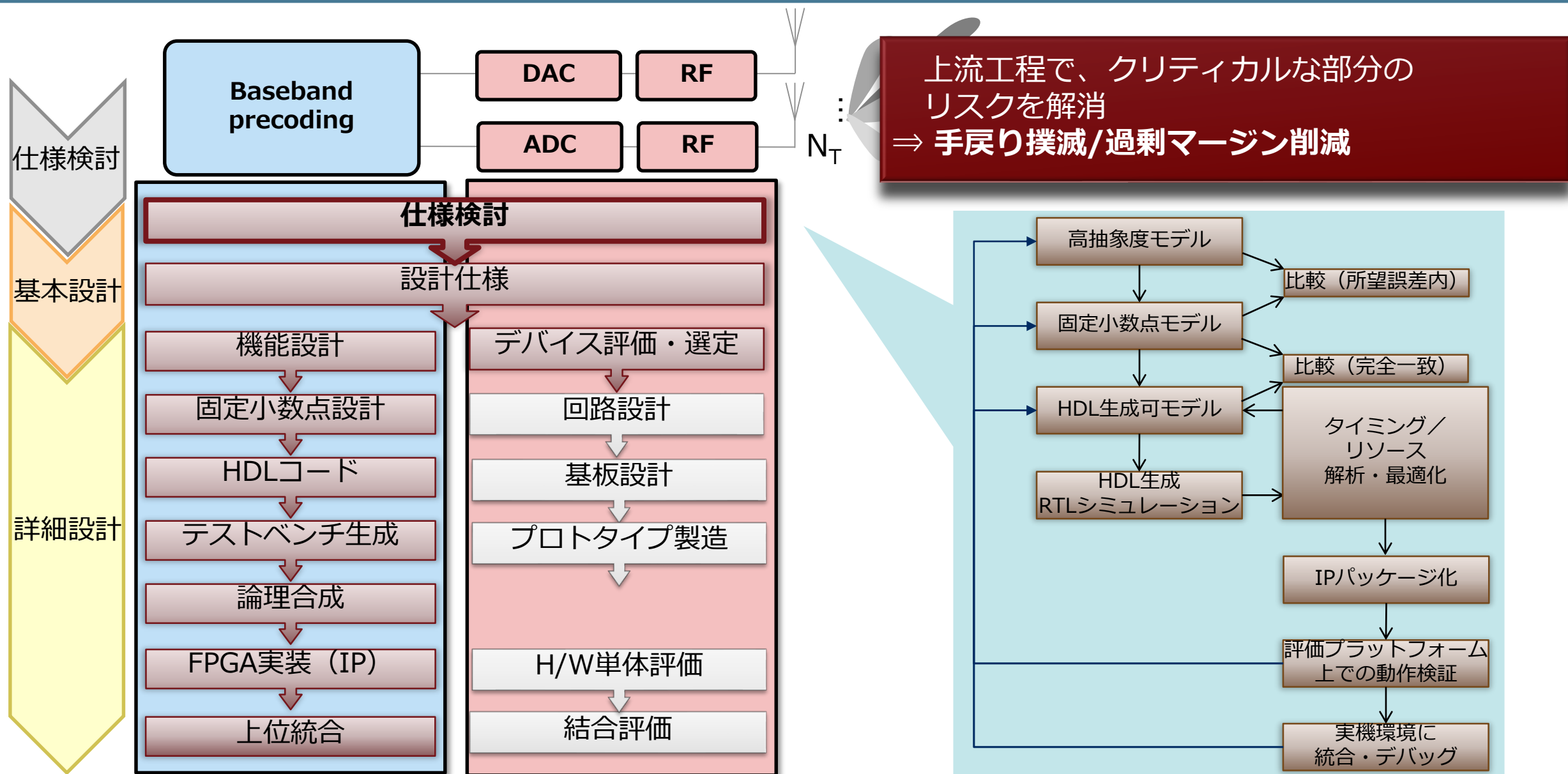


開発工程の大部分で、
MATLAB®/Simulink®/HDL Coder
を使用

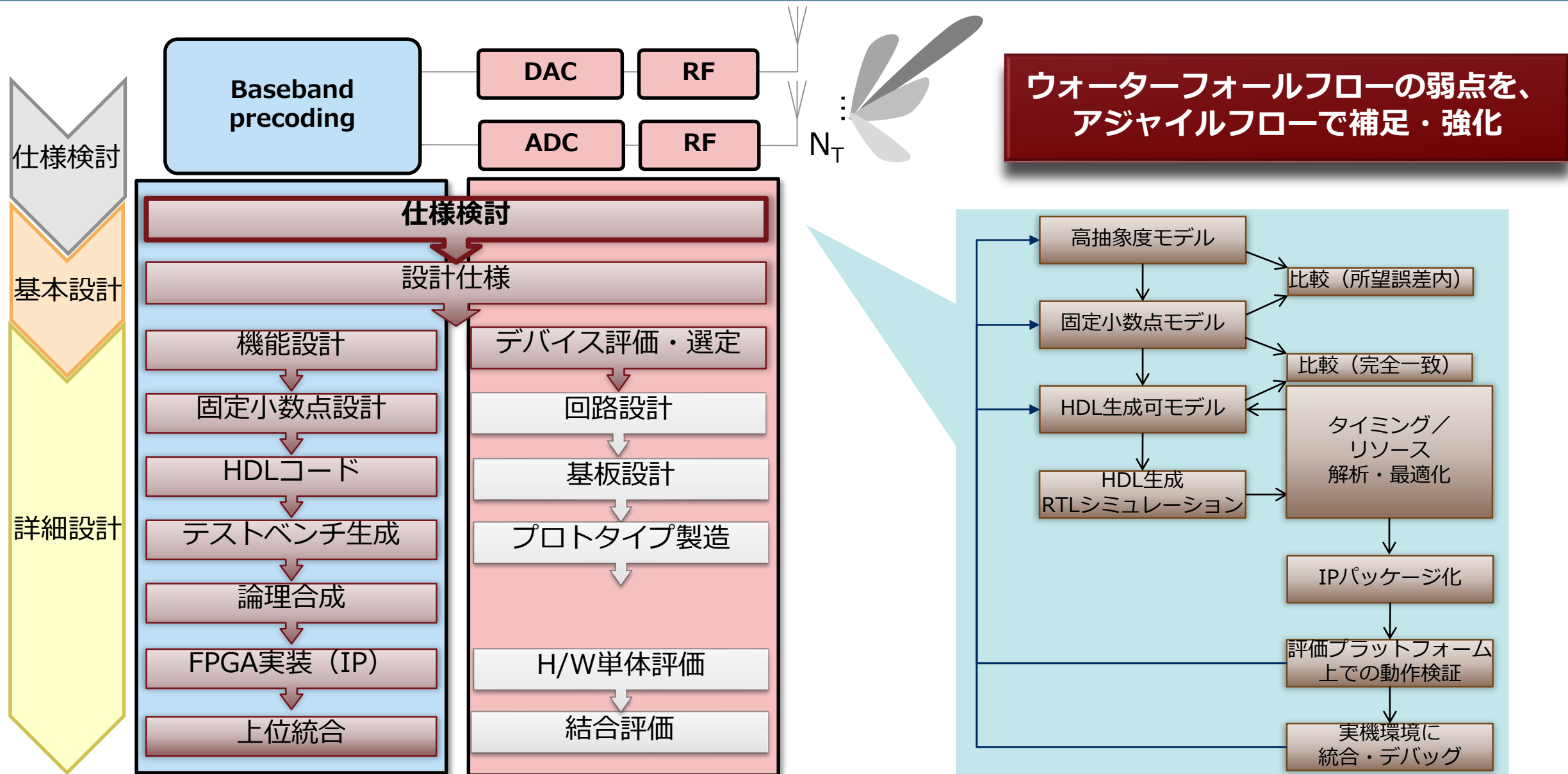
5.1. HDL Coder導入後の開発プロセス



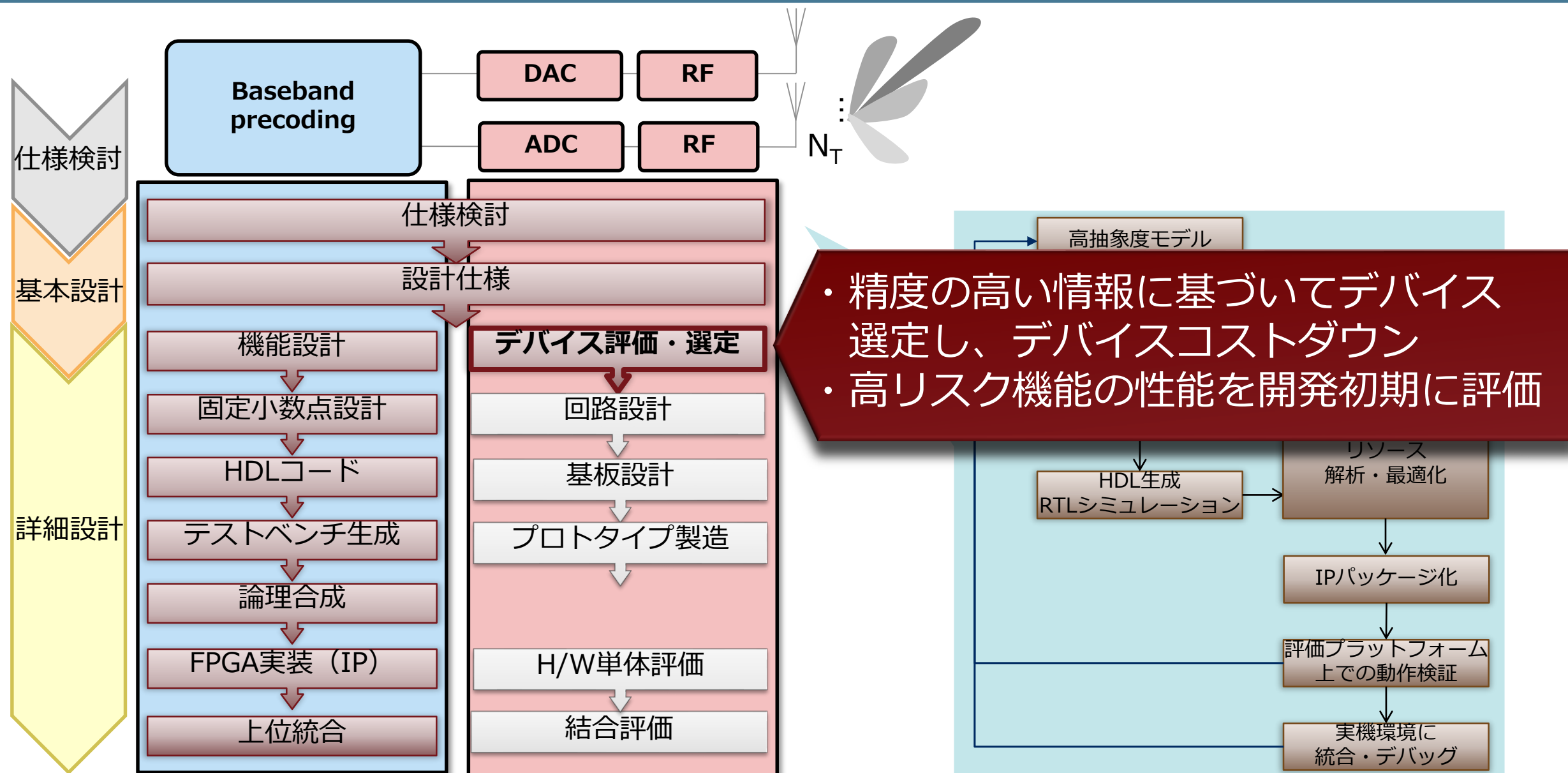
5.1. HDL Coder導入後の開発プロセス



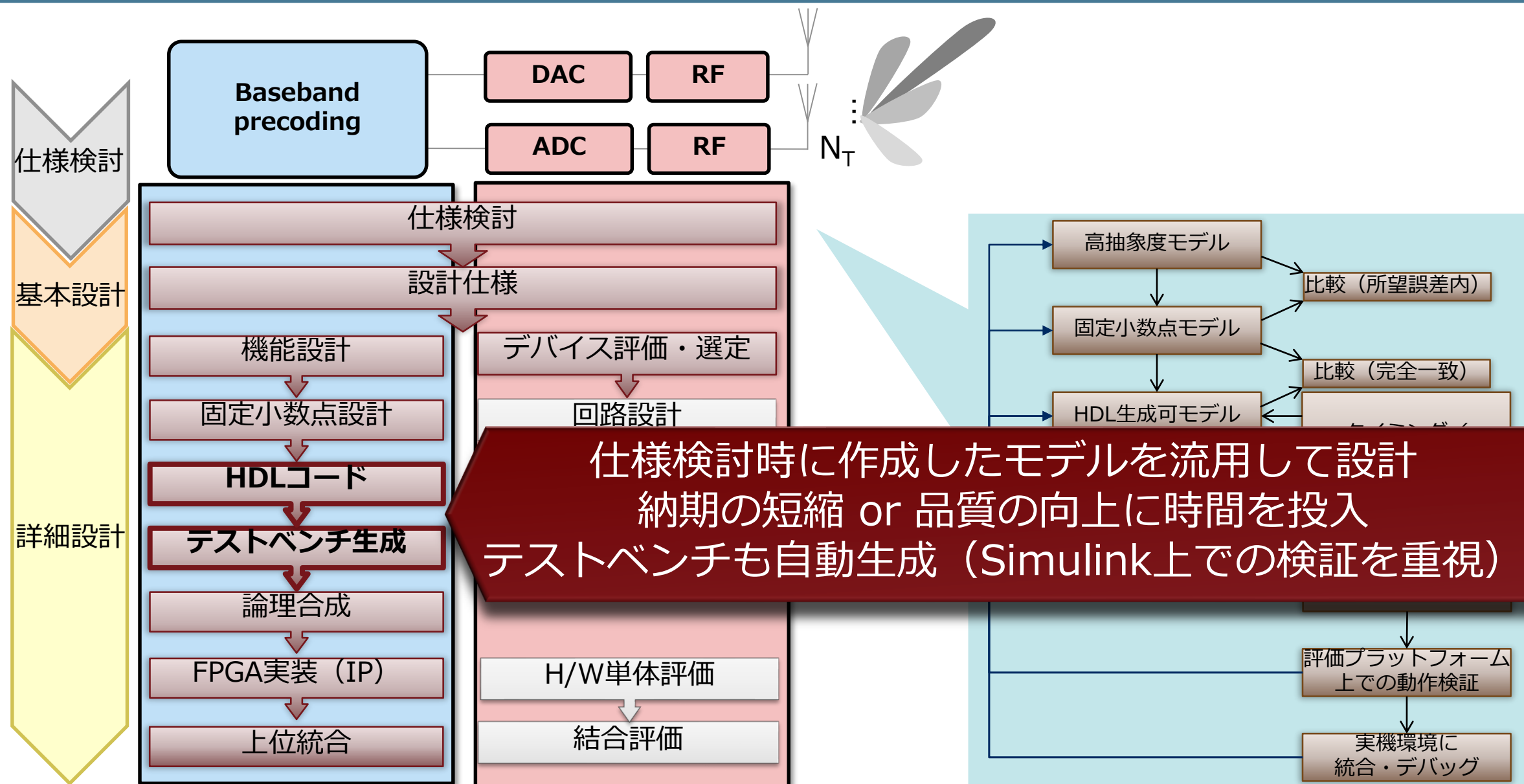
5.1. HDL Coder導入後の開発プロセス



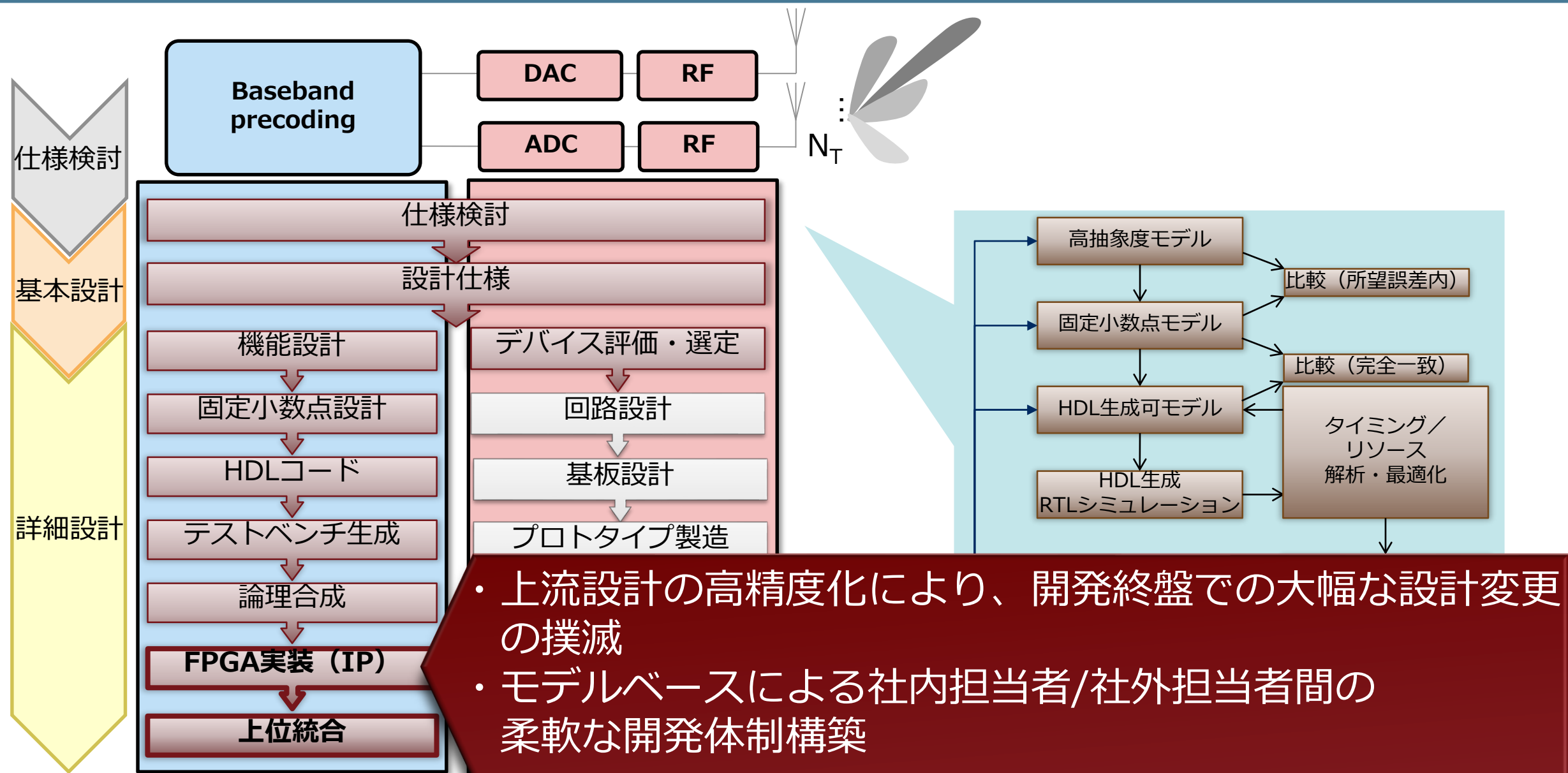
5.2. HDL Coder導入の効果①



5.2. HDL Coder導入の効果②

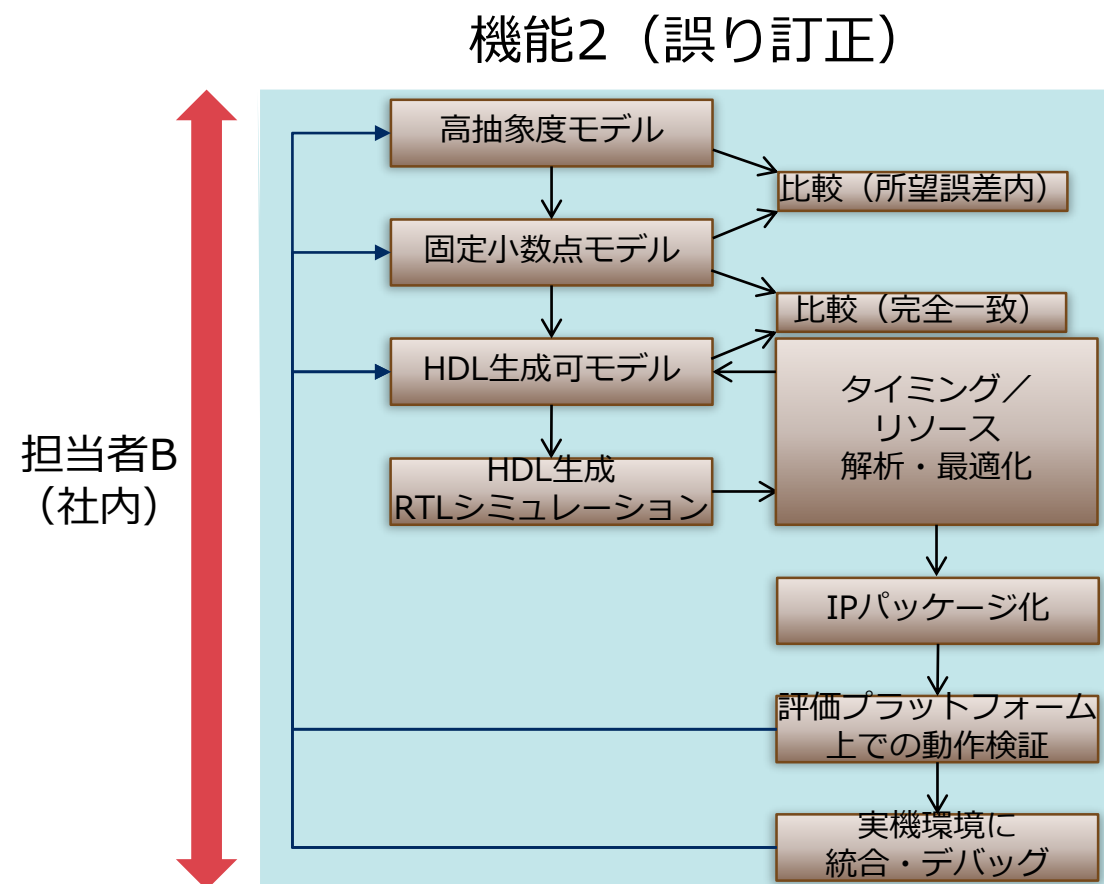
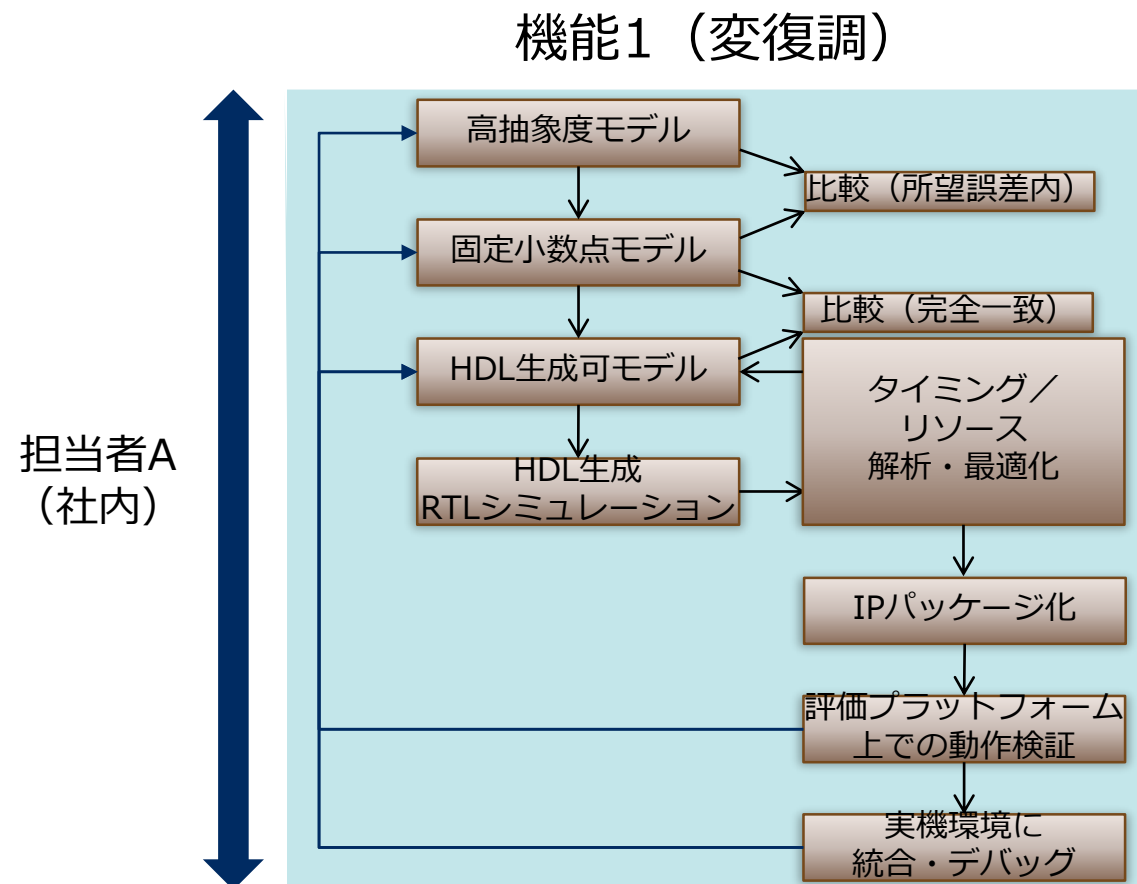


5.2. HDL Coder導入の効果③



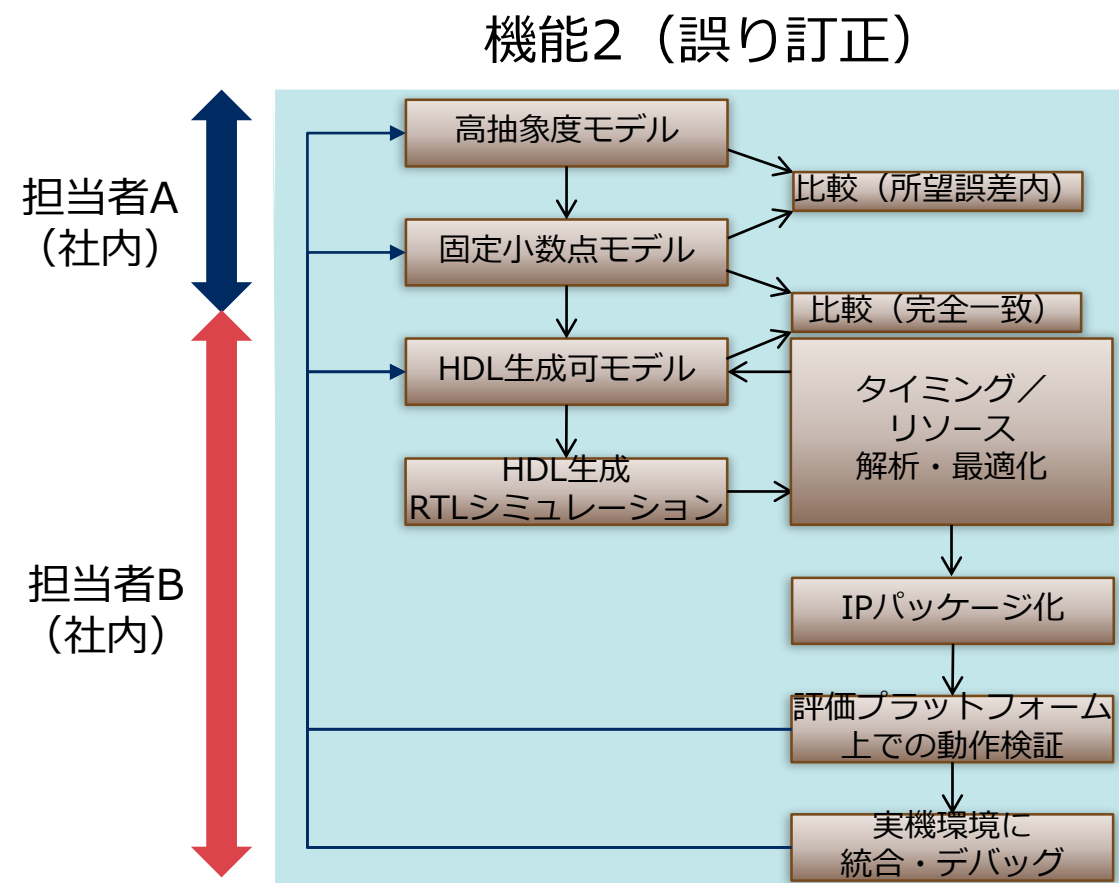
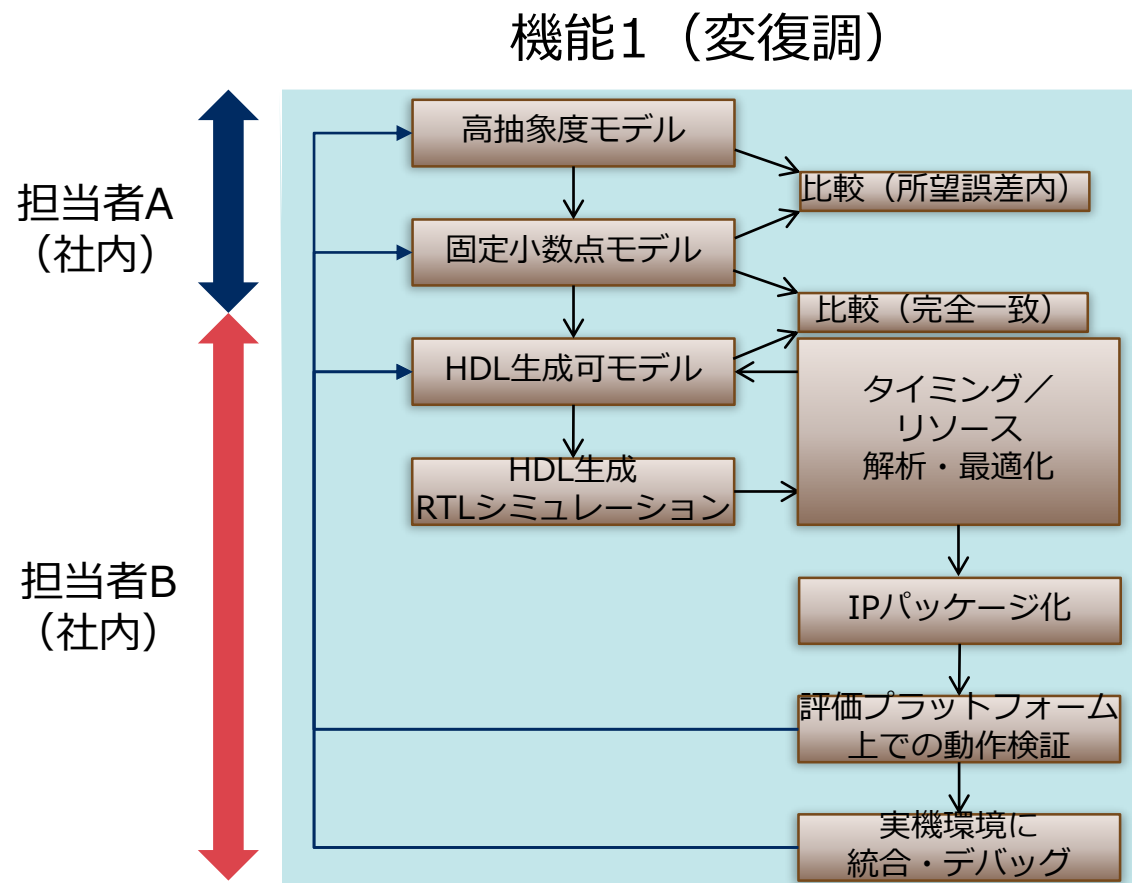
5.2. HDL Coder導入の効果③

チームでの協調開発の柔軟性向上の実例①



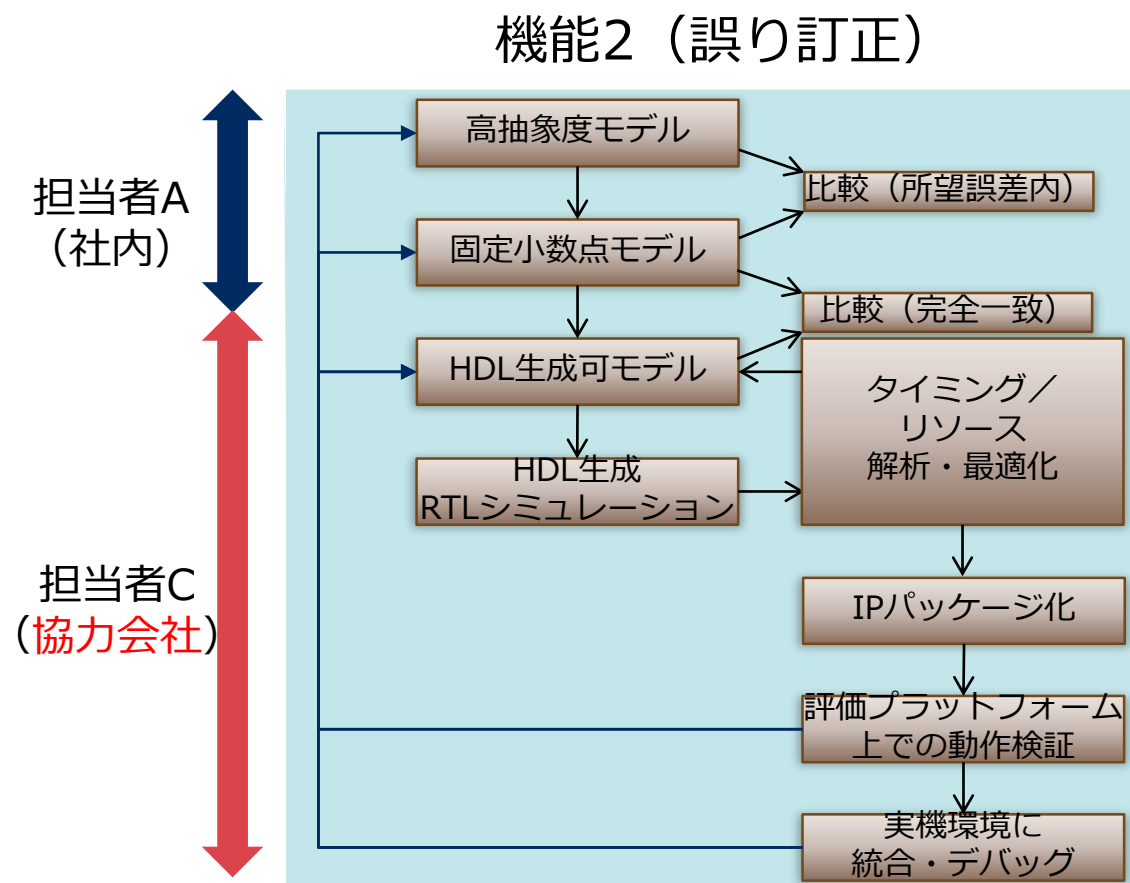
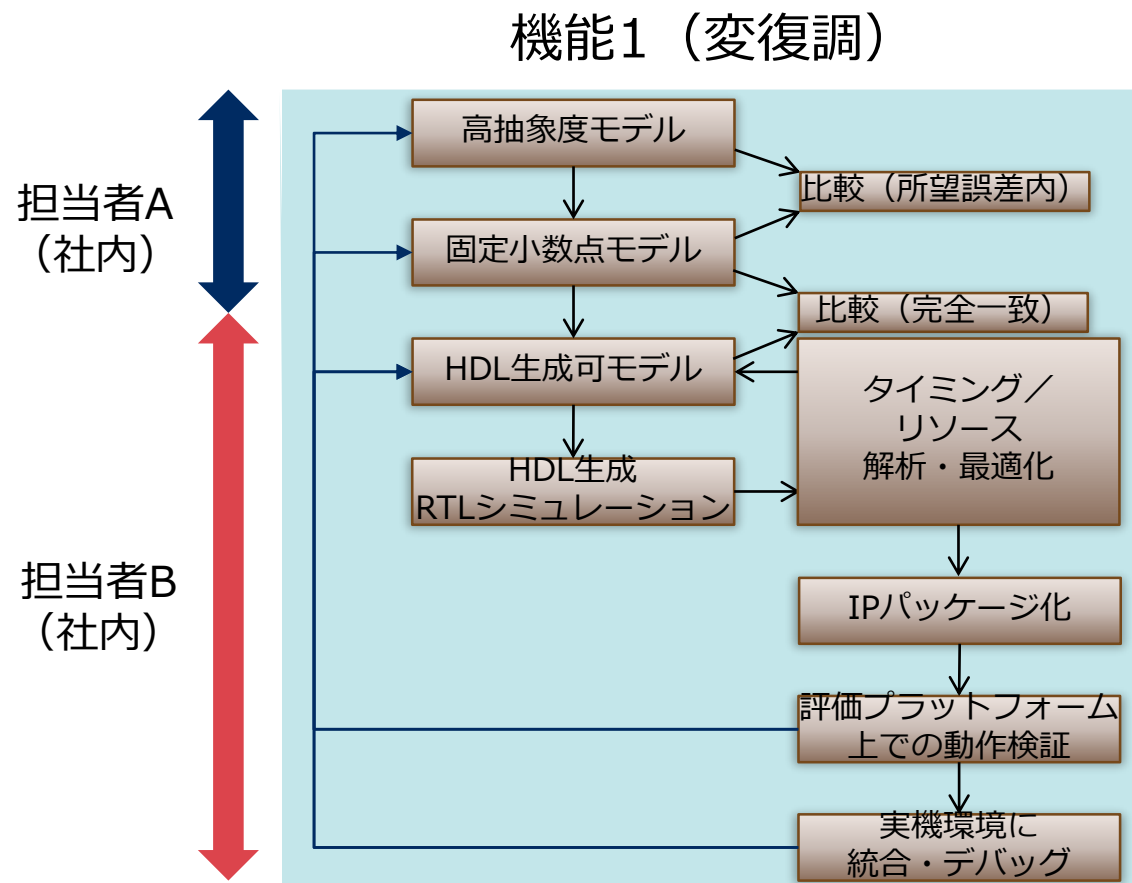
5.2. HDL Coder導入の効果③

チームでの協調開発の柔軟性向上の実例②



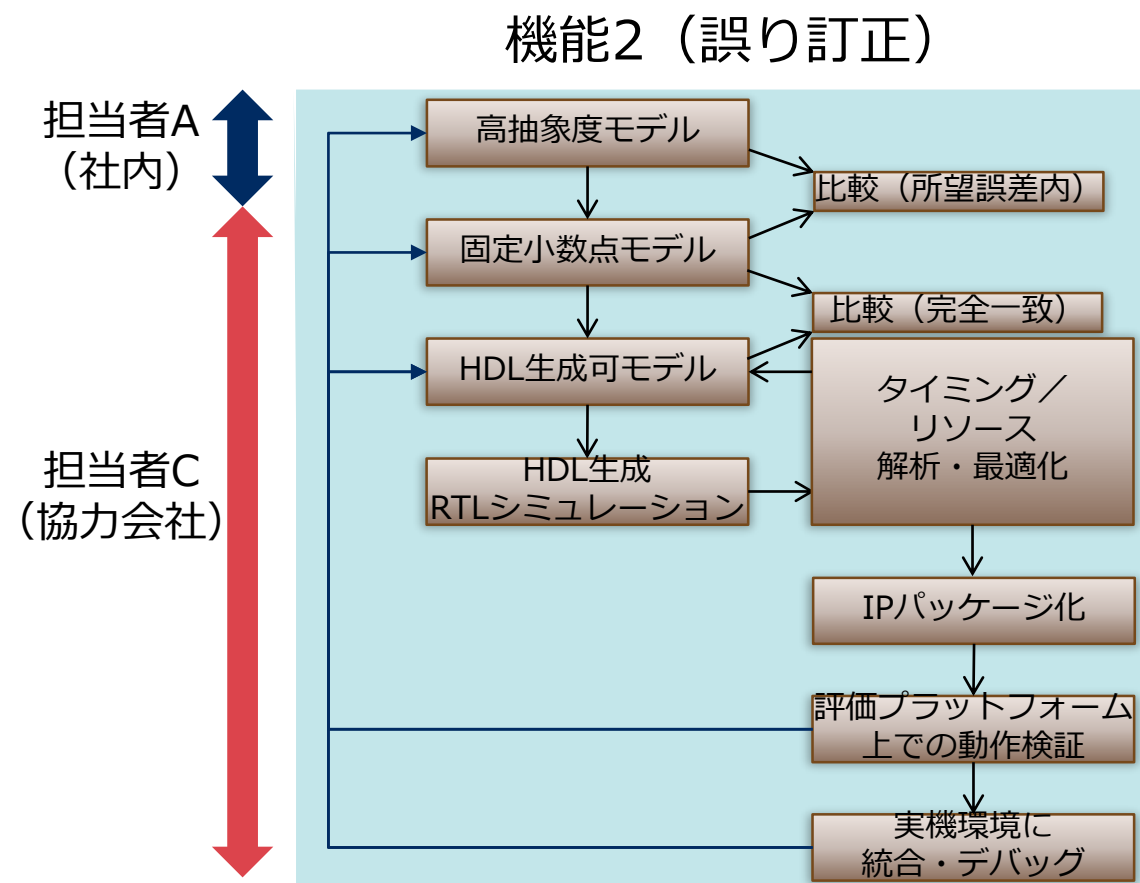
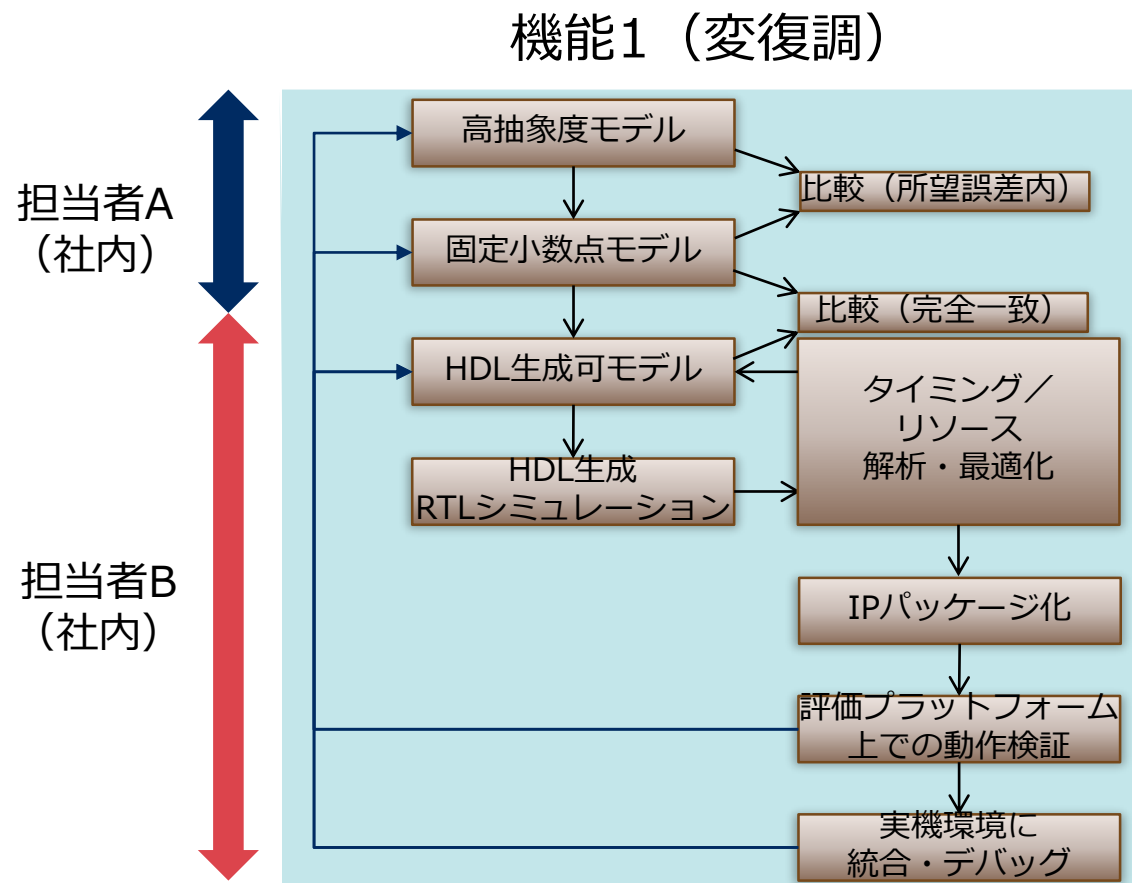
5.2. HDL Coder導入の効果③

チームでの協調開発の柔軟性向上の実例③



5.2. HDL Coder導入の効果③

チームでの協調開発の柔軟性向上の実例④



5.3. HDL Coder導入の効果（まとめ①）

仕様検討

基本設計

詳細設計

■ アルゴリズム調査・シミュレーション

- 「これできる？」に、正確・迅速に対応
- 上位システム設計者による顧客提案活動/上流設計の技術的根拠を提供
- 技術的/時間的/費用的に「できること/できないこと」を、高い精度で明確化
 - ・ 「（定性的に）できる」を、「**（定量的に）できる**」に

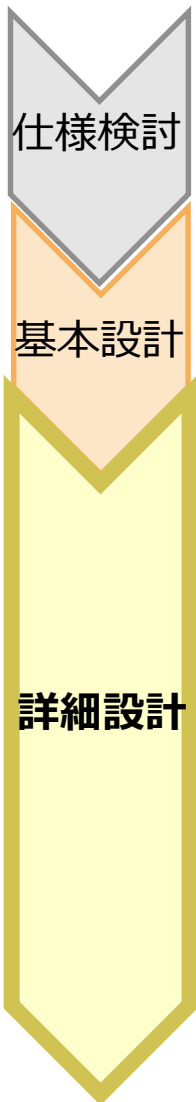
■ 仕様検討での見積り

- 大きすぎる/小さすぎるデバイス選定の抑止
- 開発費と製品費見積もり精度の向上

■ 基本設計段階でのリスク低減

- 高リスク機能（新規開発のもの、要件が難しいもの）を、開発初期で目途を付ける

5.3. HDL Coder導入の効果（まとめ②）



チームでの協調開発

- 社内担当者間の役割分担

- 変復調担当者A ⇔ 誤り訂正担当者B
- アルゴリズム設計～機能検証担当者A ⇔ HDL生成モデル最適化担当者B

- 協力会社との役割分担

- 高抽象度モデルまで社内で作成 ⇒ 協力会社でHDL生成可モデル作成～実装
- HDL生成可能モデルまで社内で作成 ⇒ 協力会社でHDL生成～実装
- 類似HDL生成可能モデルを提供 ⇒ 協力会社でカスタマイズしてIP化 ⇒ 社内実装

プロジェクトの状況により、柔軟に開発スタイルを選択可能

- モデルベースなら理解が圧倒的に速い

5.3. HDL Coder導入の効果（まとめ③）

■ 初心者は中級者に、中級者は上級者に、上級者は超上級者に

■ FPGA設計教育を、あるべき姿に

- FPGA設計教育≠HDL記述教育
- HDL記述の教育は、FPGA設計教育としては不足
- 論理回路が書ける、イメージできることが本質
- Simulinkにより、合成後の回路イメージを持ちやすくなり、技術習熟が早い
- 試行錯誤のサイクルが短いので、技術習熟が早い



技術者の成長促進

6. HDL Coder設計結果の例①

FIR補間フィルタの実装比較

- 161タップ/ポリフェーズ8倍補間/フルシリアル非対称/16bit複素数入出力
- タップ係数、補間ファクタ、ビット幅は任意のパラメータ
- 工数にはマスタモデル作成、モデル最適化、IPコアパラメータ最適化を含む

	HDL Coderで自作	Xilinx IP
LUT	88	89
FF	132	167
BRAM	0.5	0.5
DSP	2	2
Fmax	413MHz	681MHz
工数	4h	30min
メリット	ベンダIPと遜色ない 設計フローが統一できる	最適/最速 パラレル/部分シリアルアーキテクチャにも すぐ変更できる
デメリット	アーキテクチャ変更はカスタマイズが必要 DSP推論の仕様により、Fmaxが制限される (ハンドコードでも同様)	Xilinxターゲットのみ ベンダIPが使用できないプロジェクトでは 使えない フィルタ自体の検証はMATLAB/Simulinkで行う

6. HDL Coder設計結果の例②

MODEM FPGAケース1（製品化済）

- クロック周波数：384MHz
- Kintex-7 420T -2
- Slice 55% / DSP 40% / BRAM 30%
- Vivado 2018.1 / Xeon 3.4GHz 3コア6スレッド
- インプリメント時間：38分（デフォルト設定）

MODEM FPGAケース2（製品化済）

- クロック周波数：245MHz
- Kintex-7 325T -1
- Slice 40% / DSP 40% / BRAM 50%
- Vivado 2017.4 / Xeon 3.4GHz 3コア6スレッド
- インプリメント時間：21分（デフォルト設定）

インプリメント時間が短い＝各モジュールの設計が適切であることが絶対条件

HDL Coderがなければ達成は難しい

- 最適化作業/タイミング収束作業も、圧倒的に早いため

7. 留意すべき点

■ 最適な結果を得るためには、スキルが必要

- デバイスアーキテクチャを意識した設計ができるスキル
- 意図通りに合成ツールに推論させるスキル
 - 合成制約・attribute
 - ベンダ提供IPに依存している場合は要習得
- タイミング制約をはじめとする制約のスキル
- ⇒ 通常のFPGA設計と共通

8.1. 所感

HDL CoderなしのFPGA設計には戻れない

ここ2年のアップデートにより、「できないこと」がほぼなくなった

- 同期セマンティクス対応によるenable, reset動作の適正化
- グローバルリセットの削除等

高抽象度ブロックがHDL生成未対応でも、低抽象度ブロックを組み合わせれば、短時間で作成できる

- 数リリース後には機能追加により自作する必要性がなくなっていることもある

アップデートでどんどん進化し、よりスマートに、よりシンプルに使えるようになる改善が継続的に投入

- ユーザの要望ヒアリング⇒改善のサイクルが早い

8.2. 今後の展開

■ 浮動小数点のHDL生成対応により、上流工程プロセスが更に加速

- 固定小数点設計をスキップしての粗検討が可能

■ 社内・協力会社への展開を推進し、開発効率向上を継続する

- 社内
- グループ会社内
- 協力会社間

■ S/W開発への展開

- CPU部分は大部分が従来通りのハンドコード開発
- HDL Coder+FPGAの進化による開発の革新により、S/Wとの生産性差が縮少
- S/W開発でもコード生成をフル活用し、キャッチアップを図る

 **Orchestrating** a brighter world

NEC

MATLAB EXPO 2018

無線モデムFPGA/SoC開発における HDL Coder™の活用事例

NECネットワークセンサ株式会社
技術開発本部 通信ネットワーク技術部
主任
プロダクトスペシャリスト
住田 憲昭

MathWorks
田中明美



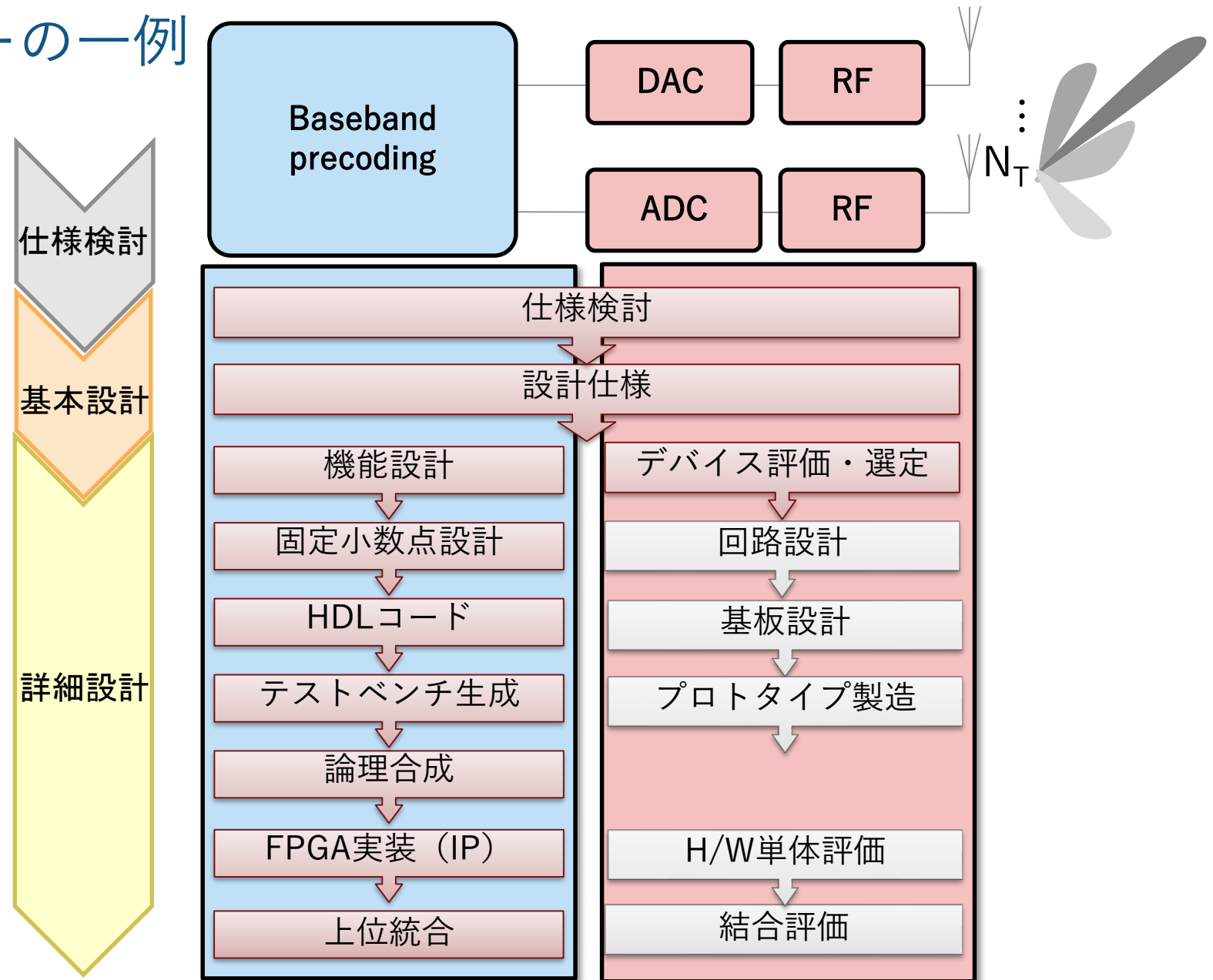
アジェンダ

- 関連オプション製品の紹介
- 関連新機能の紹介

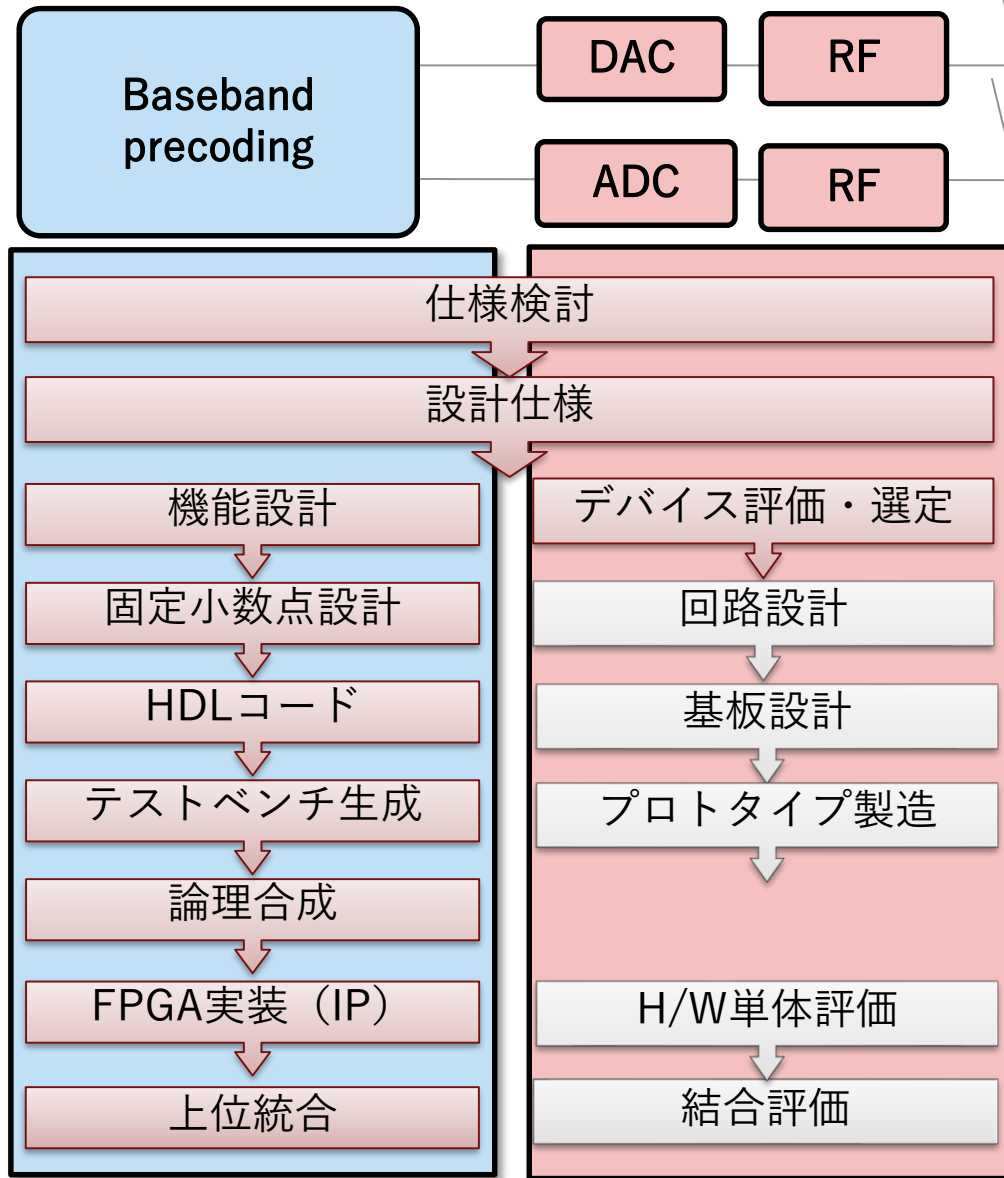
アジェンダ

- 関連オプション製品の紹介
- 関連新機能の紹介

無線機開発ワークフローの一例



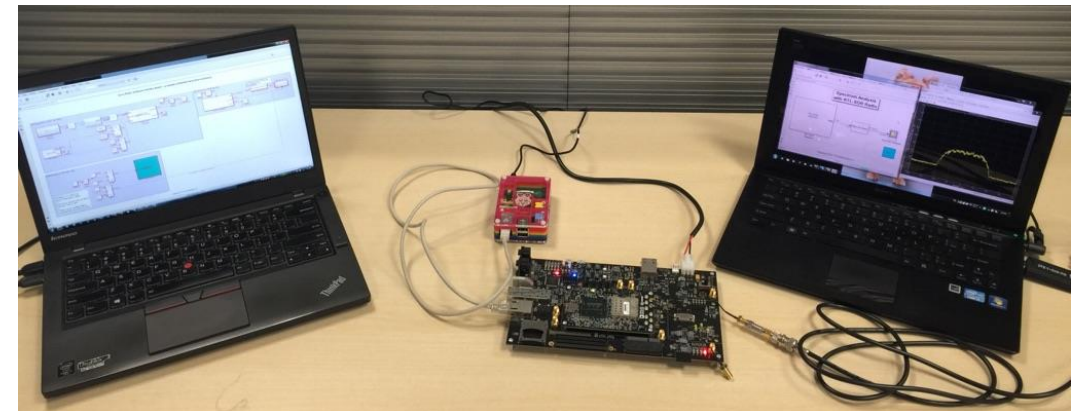
無線機開発ワークフローの一例



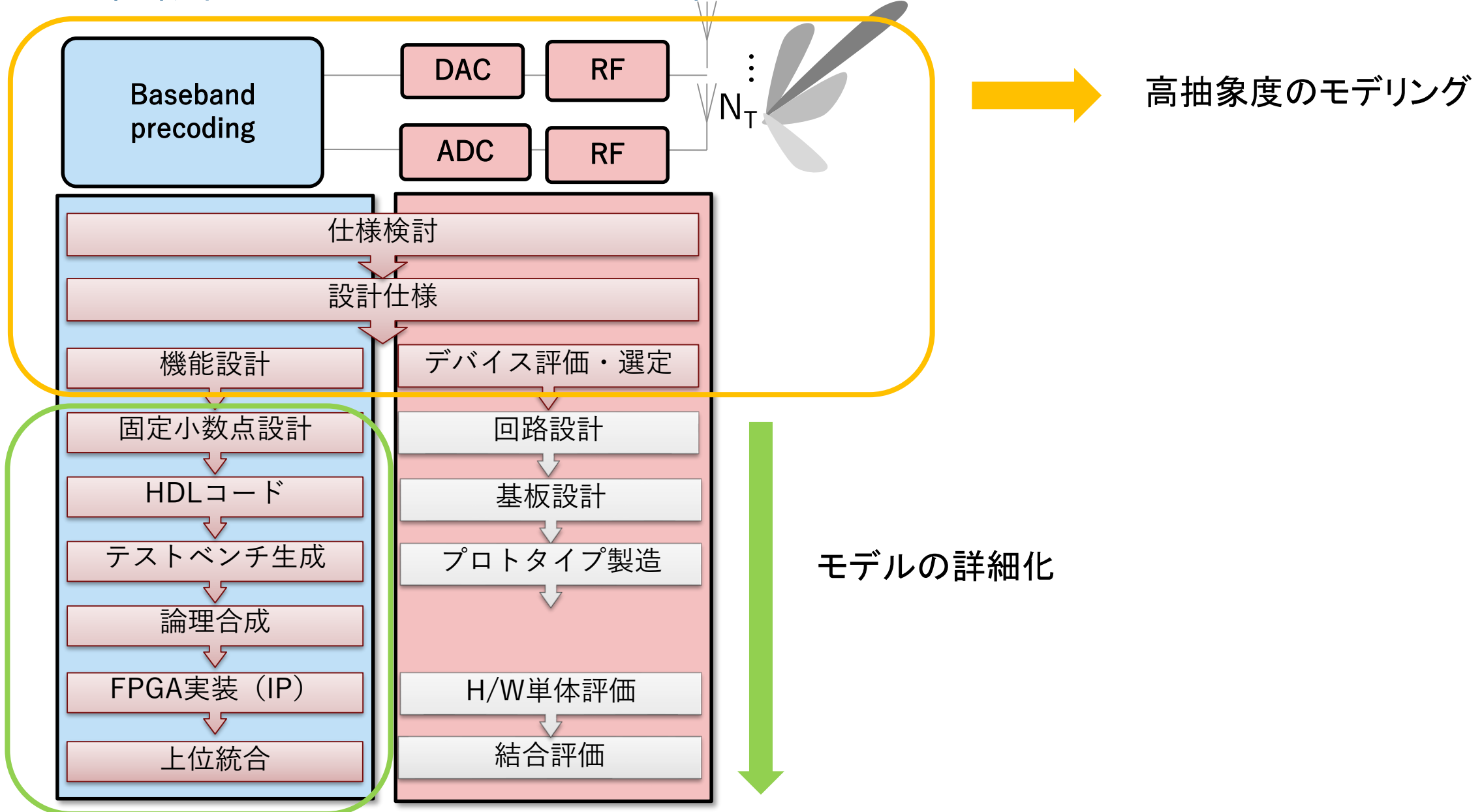
PicoZed SDR SOM



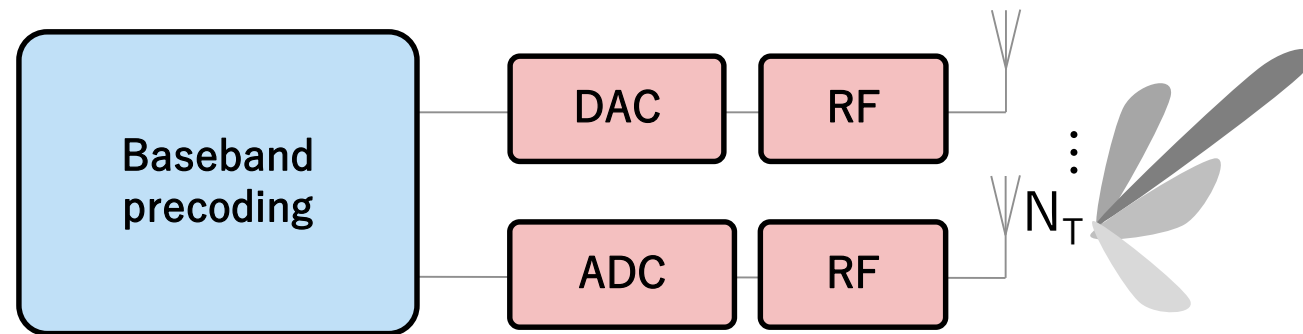
Zed Board & FMCOMMS



無線機開発ワークフローの一例



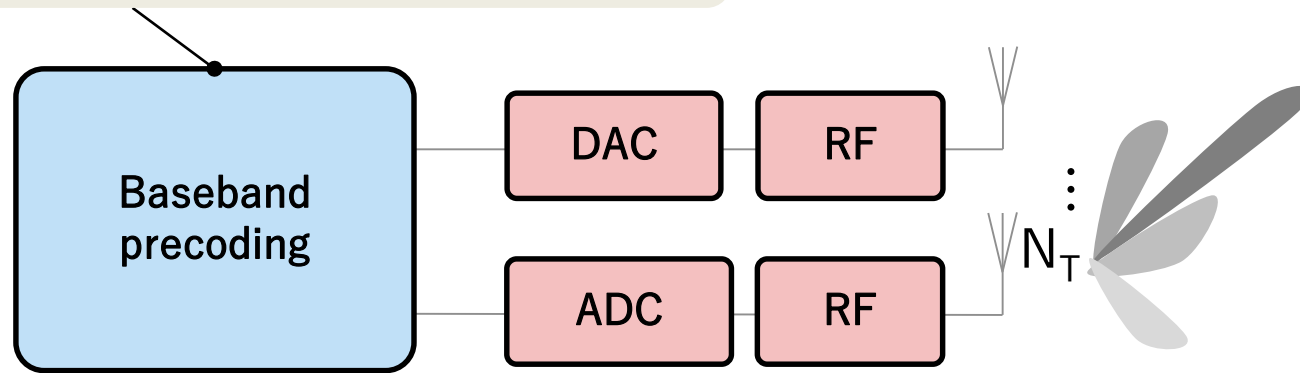
高抽象度のモデリング



高抽象度のモデリング

ベースバンドデジタル信号処理

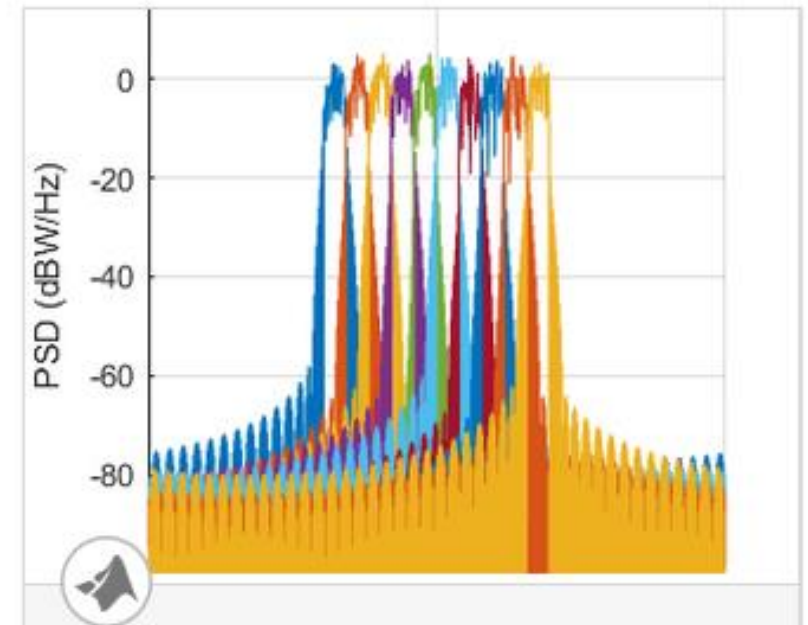
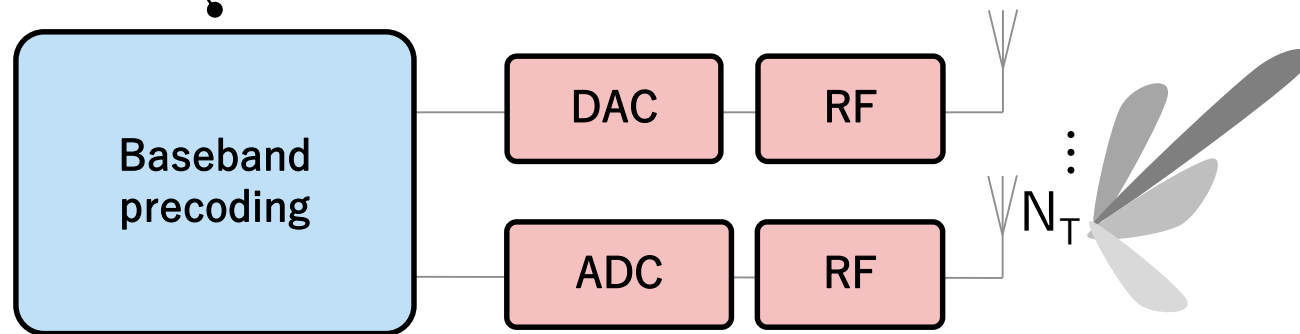
- 変復調
- エラー訂正
- フィルタ
- 同期
- 性能評価



高抽象度のモデリング

ベースバンドデジタル信号処理

- Communications Toolbox™



UFMC vs. OFDM Modulation

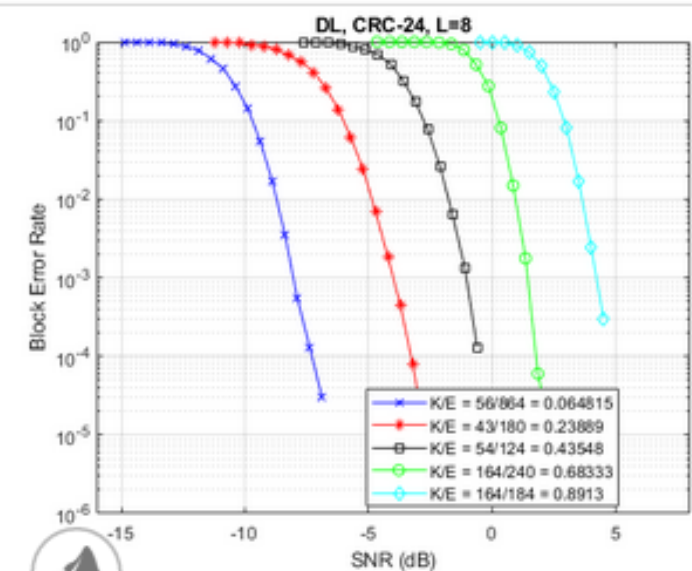
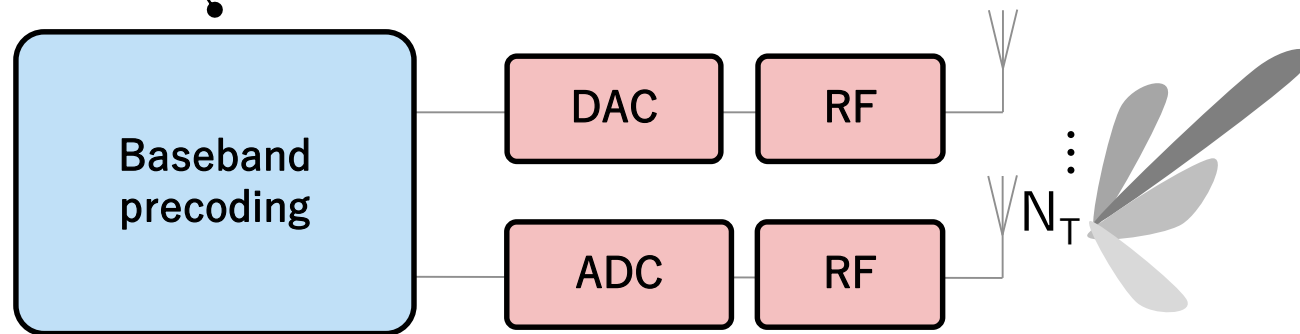
Compares Universal Filtered Multi-Carrier (UFMC) with Orthogonal Frequency Division Multiplexing (OFDM) and highlights the merits of

[Open Script](#)

高抽象度のモデリング

ベースバンドデジタル信号処理

- Communications Toolbox™
- 5G Toolbox™



5G New Radio Polar Coding

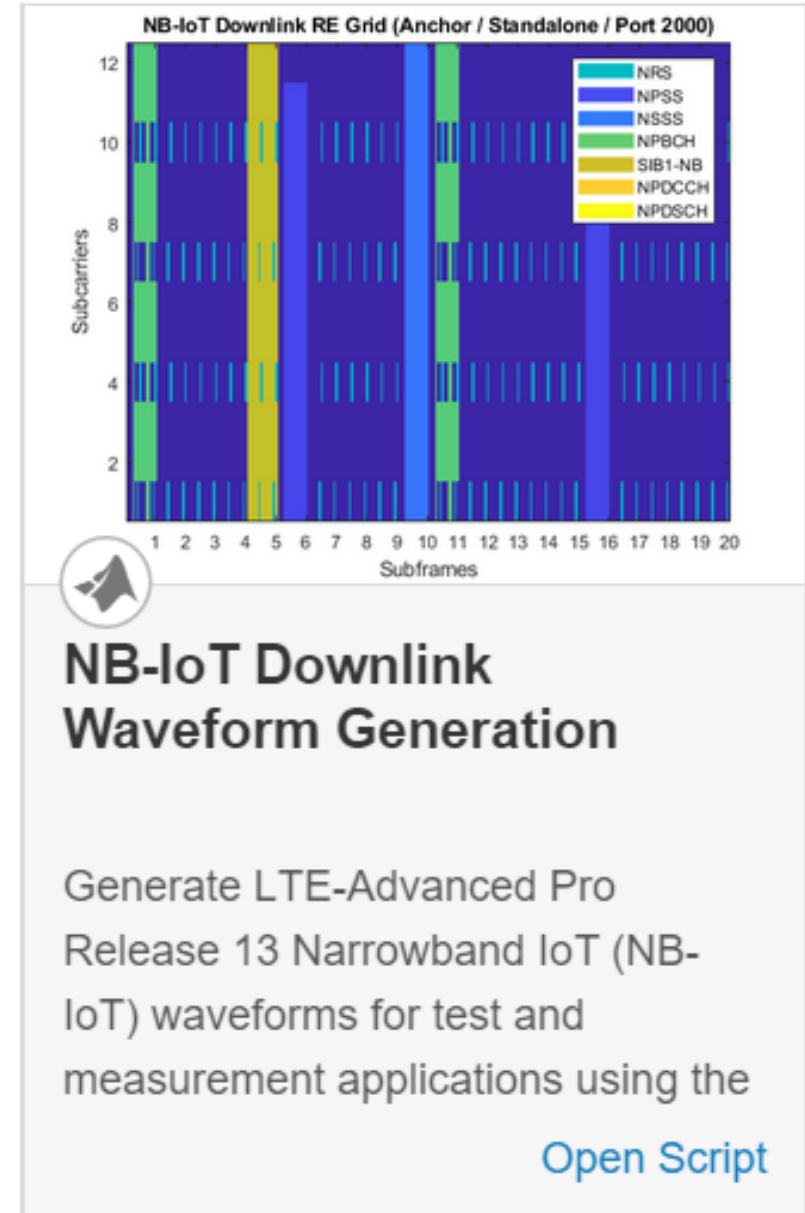
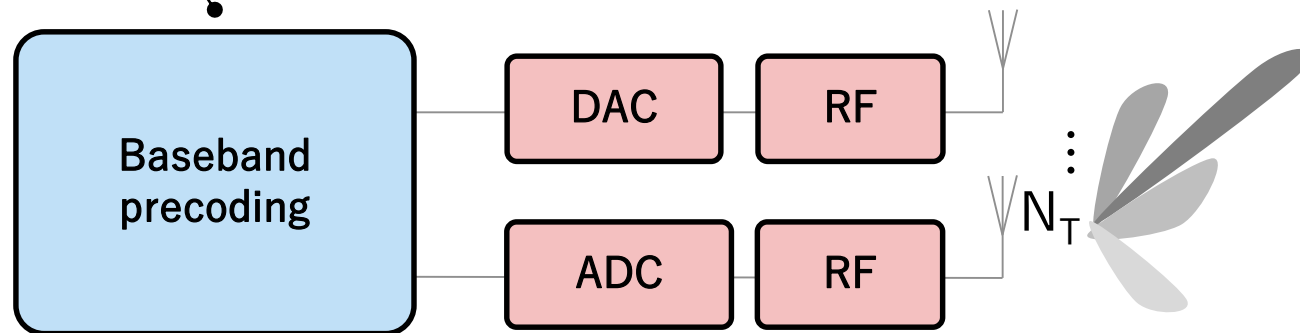
Highlights the new polar channel coding technique of NR communications system by modeling the CRC-aided polar (CA-

[Open Script](#)

高抽象度のモデリング

ベースバンドデジタル信号処理

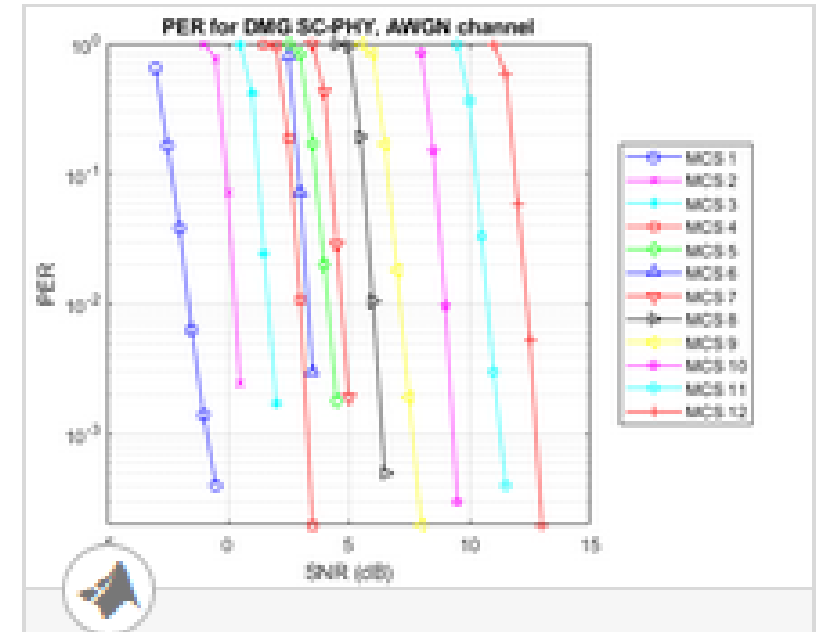
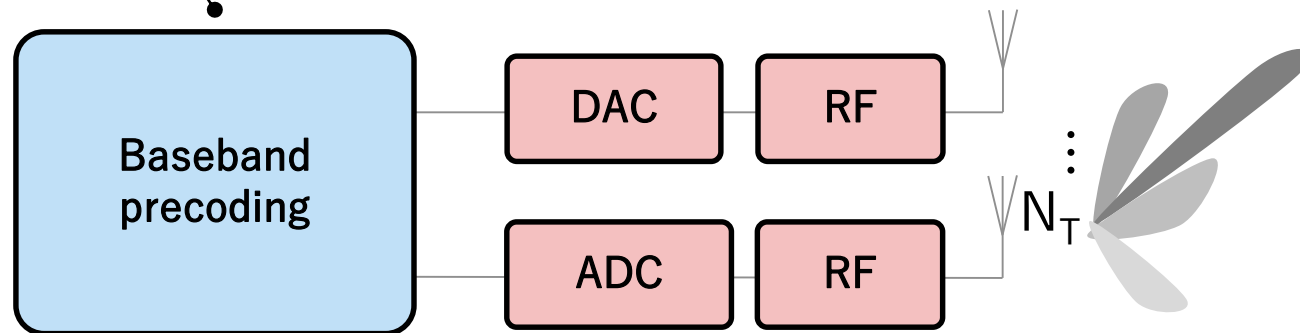
- Communications Toolbox™
- 5G Toolbox™
- LTE Toolbox™



高抽象度のモデリング

ベースバンドデジタル信号処理

- Communications Toolbox™
- 5G Toolbox™
- LTE Toolbox™
- WLAN Toolbox™



802.11ad Packet Error Rate Simulation for Single Carrier PHY

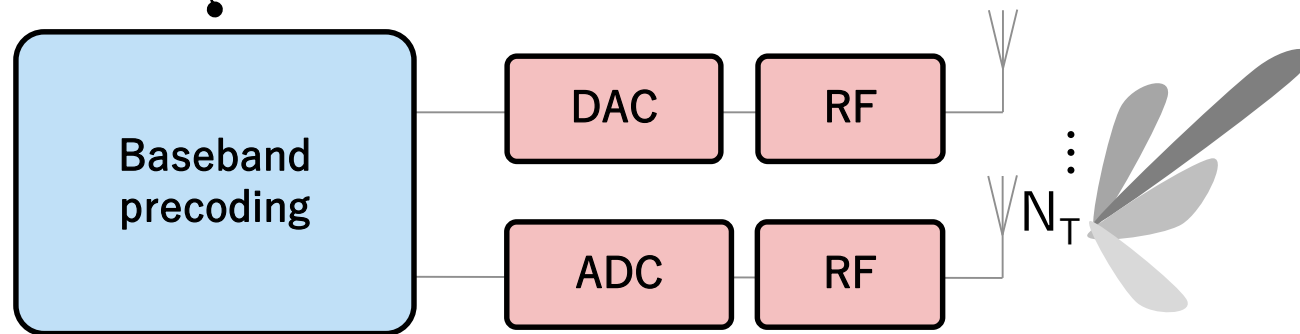
Measure the packet error rate of an IEEE® 802.11ad™ DMG single carrier (SC) PHY AWGN link using an end-to-end simulation.

[Open Script](#)

高抽象度のモデリング

MAC層のシミュレーション

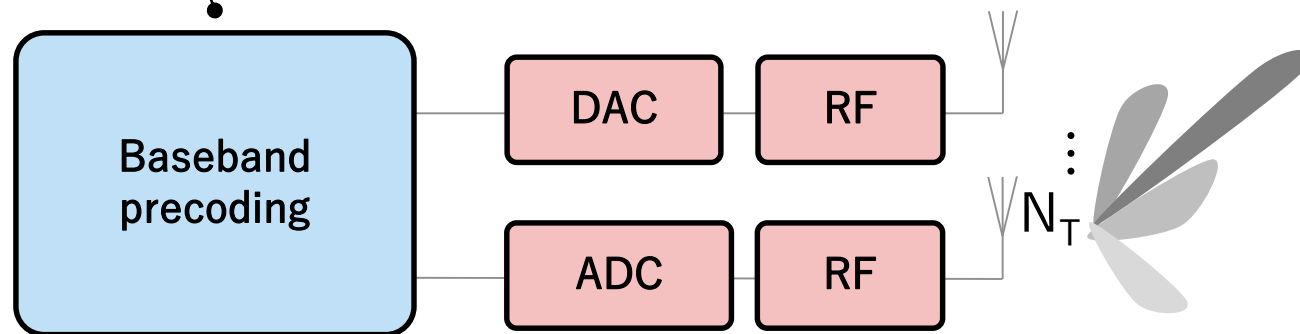
- スループット
- ACK
- 衝突



高抽象度のモデリング

MAC層のシミュレーション

- Stateflow®



A diagram illustrating a wireless network. A central blue cloud labeled "Fading Network Channel" is connected by double-headed blue arrows to three nodes: "Radio Node2" (top, orange car icon), "Radio Node1" (bottom right, blue car icon), and "Radio Node" (bottom left, orange car icon). To the right of the cloud, a sub-diagram shows a "MAC" block receiving inputs from "ALOHA" and "CSMA" blocks. Below the diagram, the text "ALOHA and CSMA/CA Packetized Wireless Networks" is displayed. Below this, a paragraph states: "Simulate a basic ALOHA or CSMA/CA MAC using Simulink®, Stateflow® and the Communications Toolbox™." At the bottom right, a blue link labeled "Open Model" is present.

ALOHA and CSMA/CA Packetized Wireless Networks

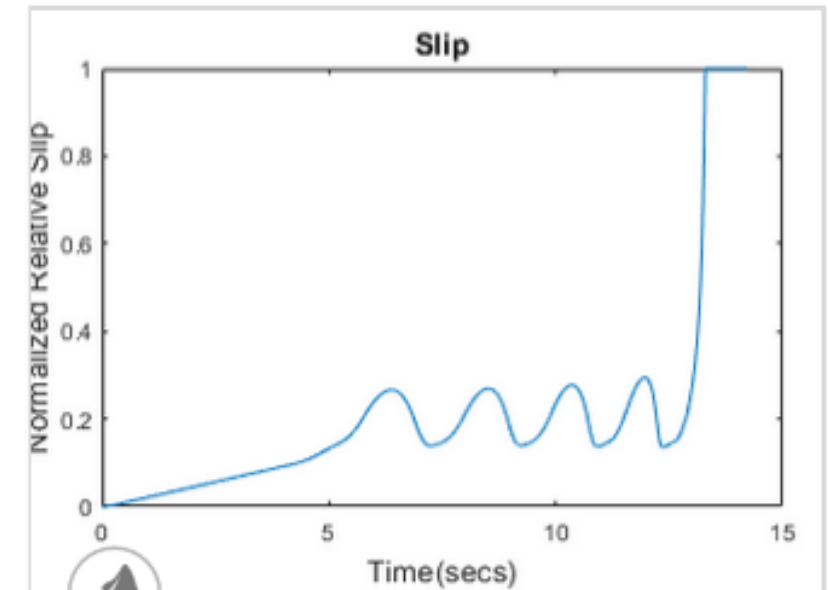
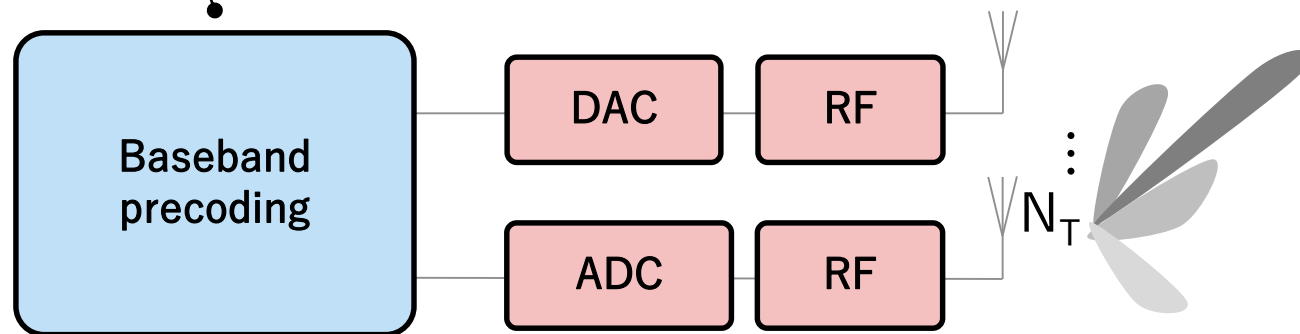
Simulate a basic ALOHA or CSMA/CA MAC using Simulink®, Stateflow® and the Communications Toolbox™.

[Open Model](#)

高抽象度のモデリング

MAC層のシミュレーション

- Stateflow®
- SimEvents®



Effects of Communication Delays on an ABS Control System

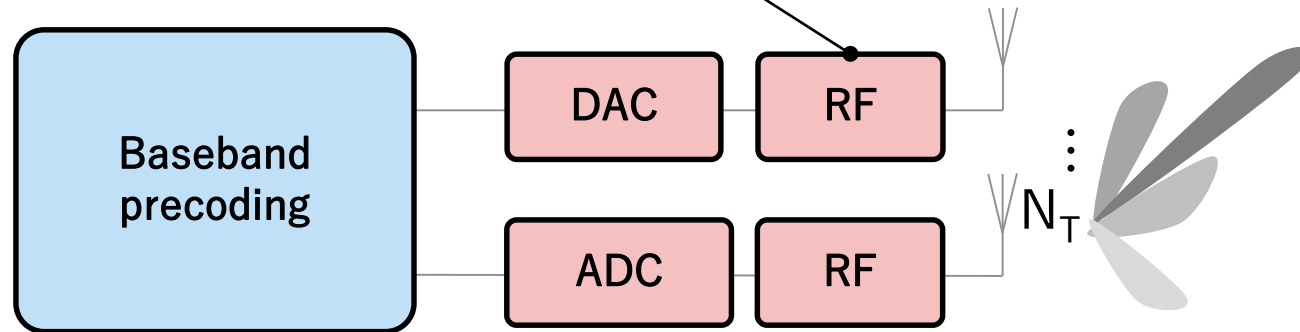
How stochastic network traffic causes timing latency and uncertainty in an anti-lock braking system (ABS) that uses Control

[Open Model](#)

高抽象度のモデリング

RF損失の影響を調査する

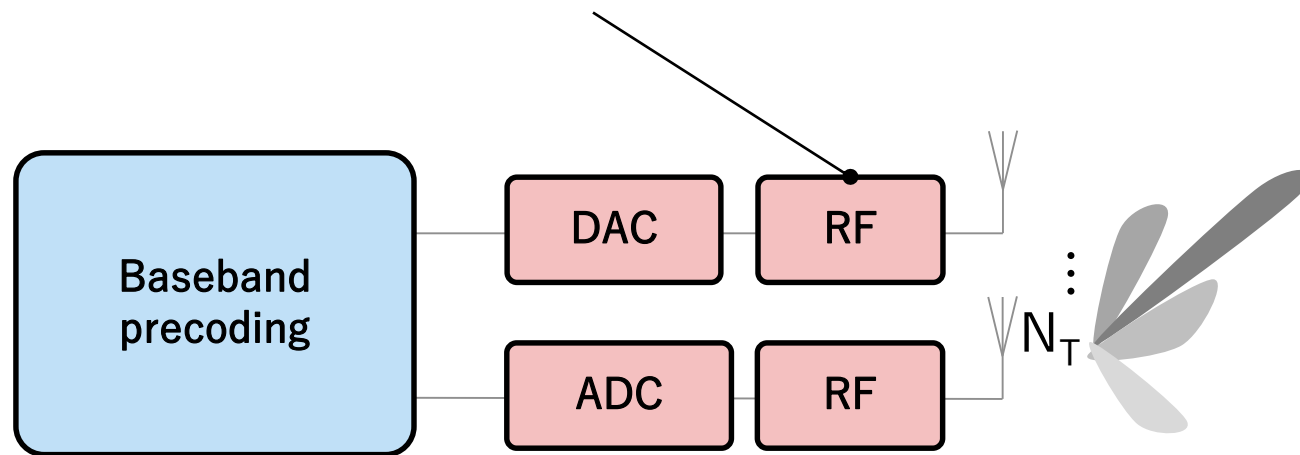
- 周波数依存性
- 非線形性
- ミスマッチとカップリング



高抽象度のモデリング

RF損失の影響を調査する

- RF Toolbox™
- RF Blockset™



Impact of an RF Receiver on Communication System Performance

The diagram shows a detailed RF receiver model. The signal path starts from an antenna, goes through a block labeled "SL RF", then an "LNA" (Low Noise Amplifier) block, followed by an "Image Reject" block, a "LO" (Local Oscillator) block, and finally an "RF" block. The output of the "RF" block is labeled "SimRF Output". A noise source block labeled "Noise Figure 6.3218 dB" is connected to the signal path, and its output is labeled "Combined Noise". The signal path also includes a block labeled "SF_Gain_LNA+SF_Gain_Mixer". The final output is labeled "RX Out".

Impact of an RF Receiver on Communication System Performance

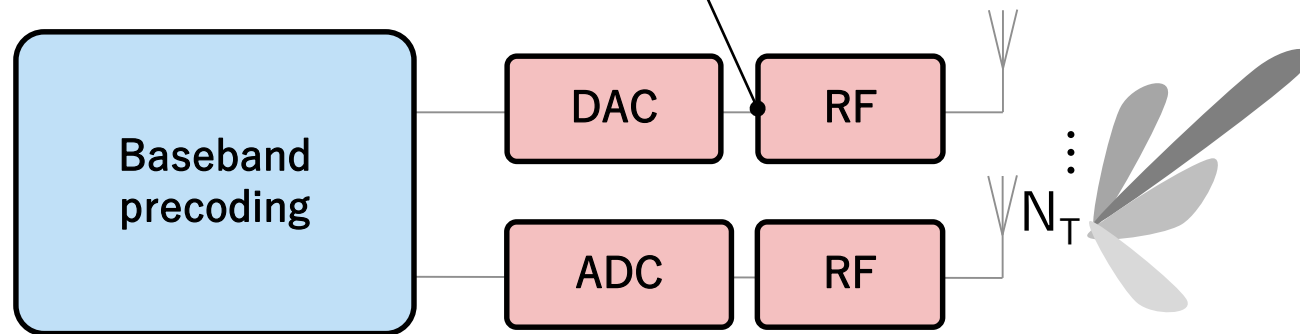
Use the RF Blockset™ Circuit Envelope library to measure the effect of thermal noise on the bit error rate (BER) of a

[Open Model](#)

高抽象度のモデリング

ビームフォーミングのトレードオフを探る

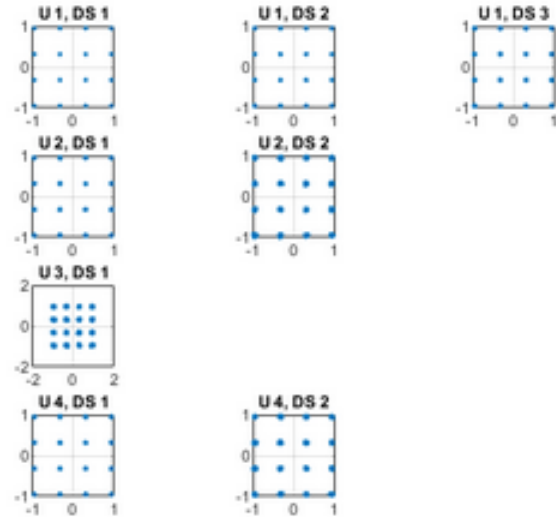
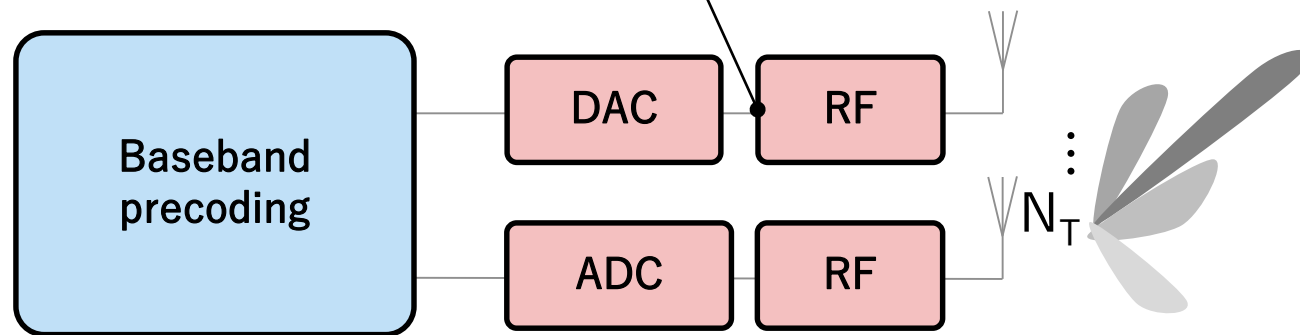
- ベースバンド、アナログ、ハイブリッドビームフォーミング
- 性能限界をシミュレーション



高抽象度のモデリング

ビームフォーミングのトレードオフを探る

- Phased Array System Toolbox™



Massive MIMO Hybrid Beamforming

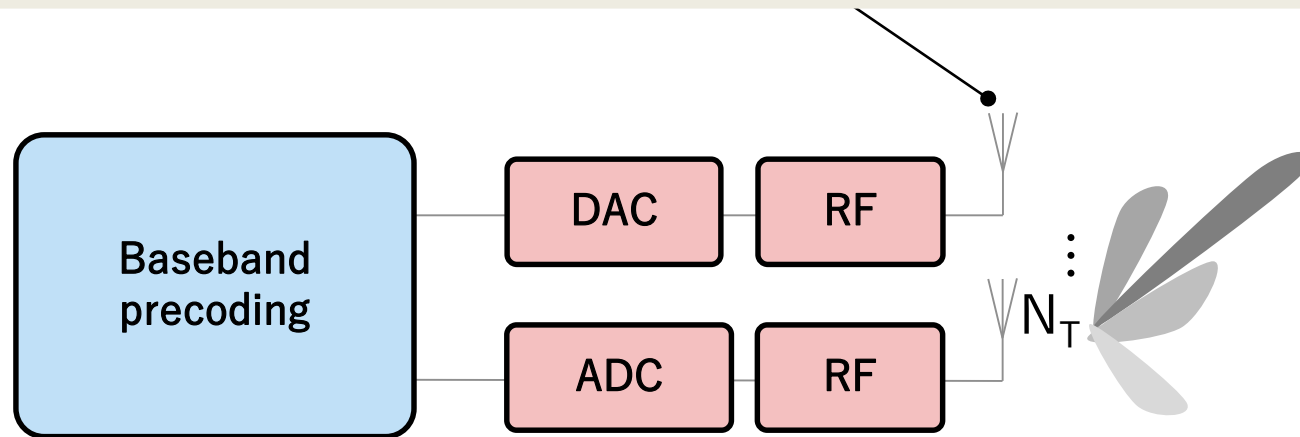
How hybrid beamforming is employed at the transmit end of a massive MIMO communications system, using techniques for both

[Open Script](#)

高抽象度のモデリング

アンテナアレイの設計/評価、FR伝搬モデルを考慮した解析

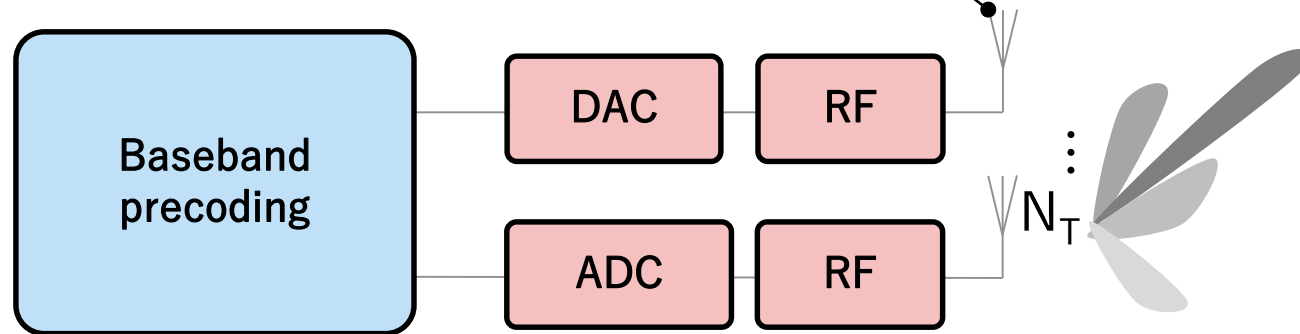
- エレメントカップリング
- エッジ効果
- 気象ベースモデル
- カバレッジ



高抽象度のモデリング

アンテナアレイの設計/評価、FR伝搬モデルを考慮した解析

• Antenna Toolbox™



SINR Map for a 5G Urban Macro-Cell Test Environment

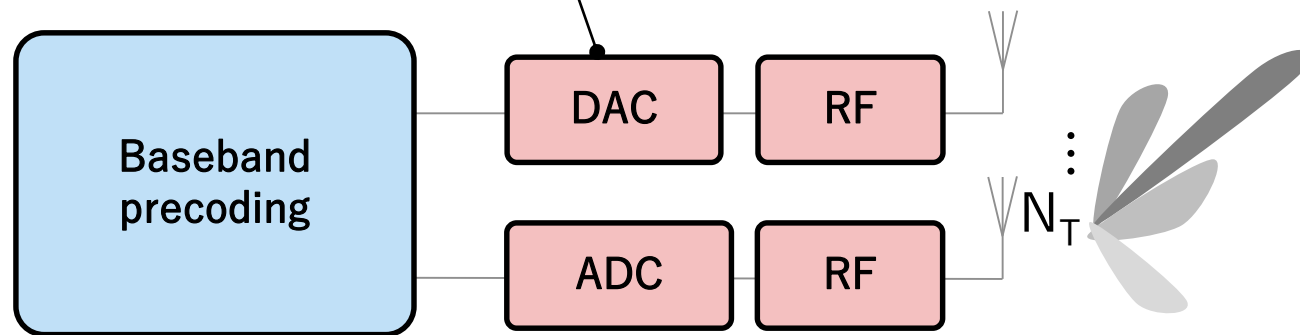
Construct a 5G urban macro-cell test environment and visualize the signal-to-interference-plus-noise ratio (SINR) on a map. The test

[Open Live Script](#)

高抽象度のモデリング

ADC/DACの振る舞いを模擬

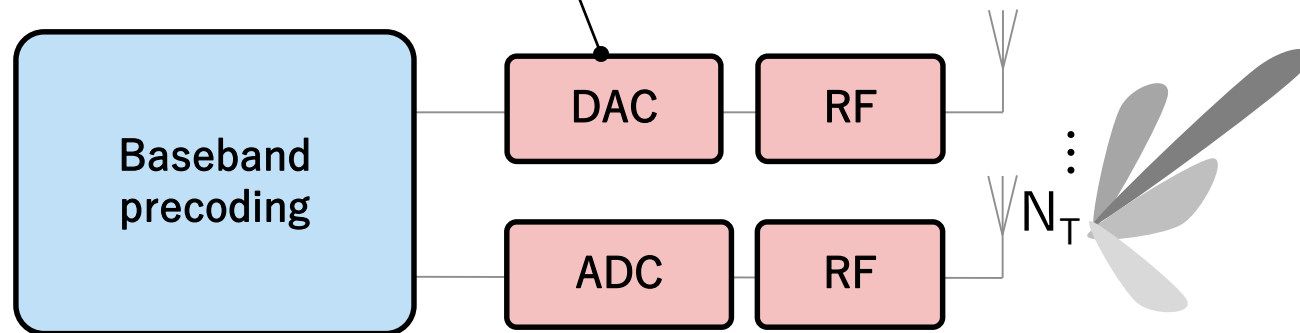
- デジタル、アナログ混在
- AGCのモードによる遅延
- 特定のチップの振る舞い



高抽象度のモデリング

ADC/DACの振る舞いを模擬したい

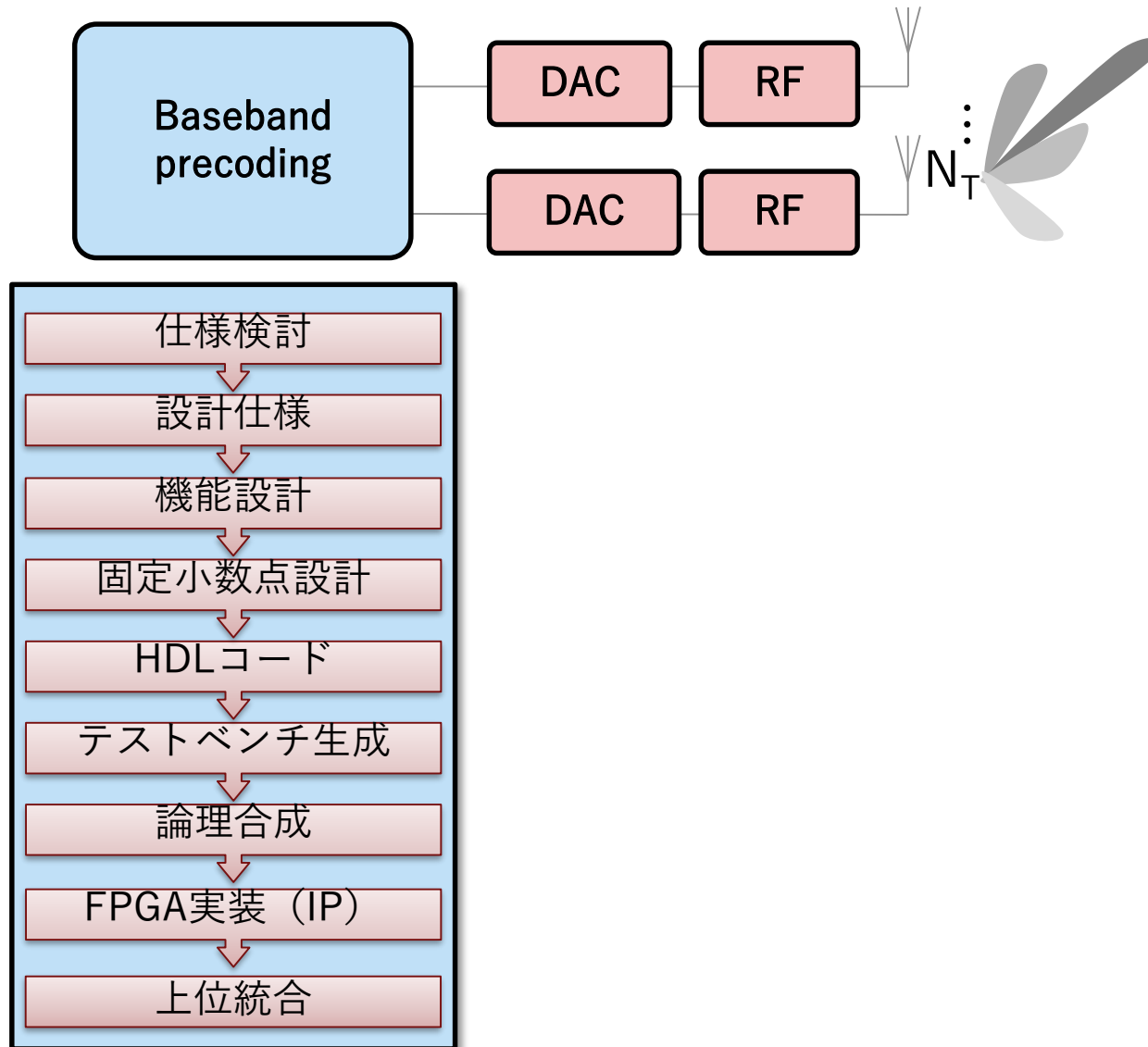
- サポートパッケージ



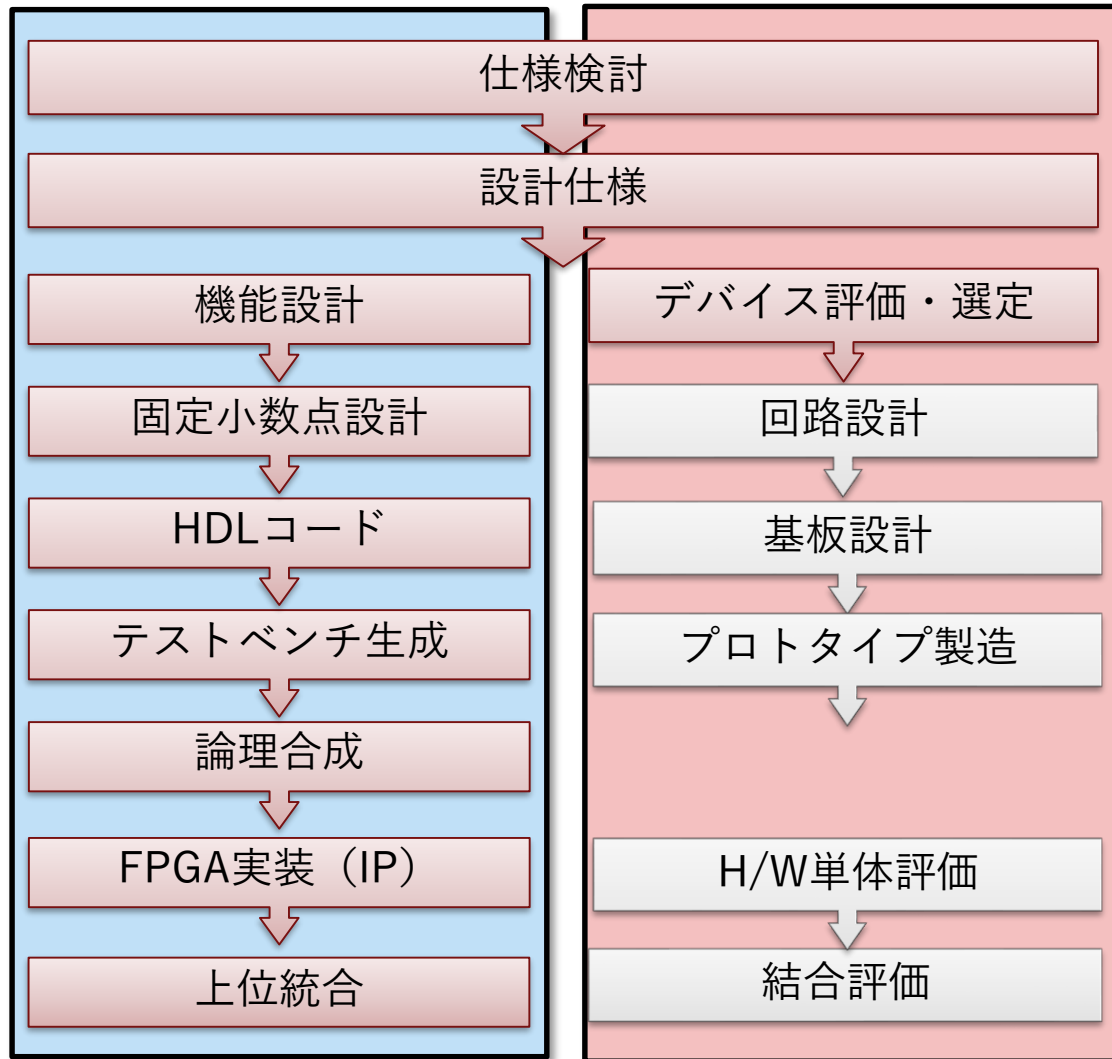
The screenshot shows the "アドオン エクスプローラー" (Add-on Explorer) window. It displays the "RF Blockset Models for Analog Devices RF Transceivers" add-on. The add-on is marked as "インストール済み" (Installed). The author is listed as "MathWorks RF Products Team". The description states: "Simulate and verify agile RF transceivers and predict impact of RF imperfections using test signals". The MathWorks logo and "オプション機能" (Optional Feature) are also visible.

The screenshot shows the Simulink library browser for the "ad9371_models" library. The library contains four blocks: "ad9371 Transmitter", "ad9371 Receiver", "ad9371 Observer RX", and "ad9371 Sniffer RX". Each block has input and output ports labeled "BB", "LO", "RF", and "Out". The copyright notice at the bottom reads "Copyright 2016-2017 The MathWorks, Inc.".

モデルの詳細化

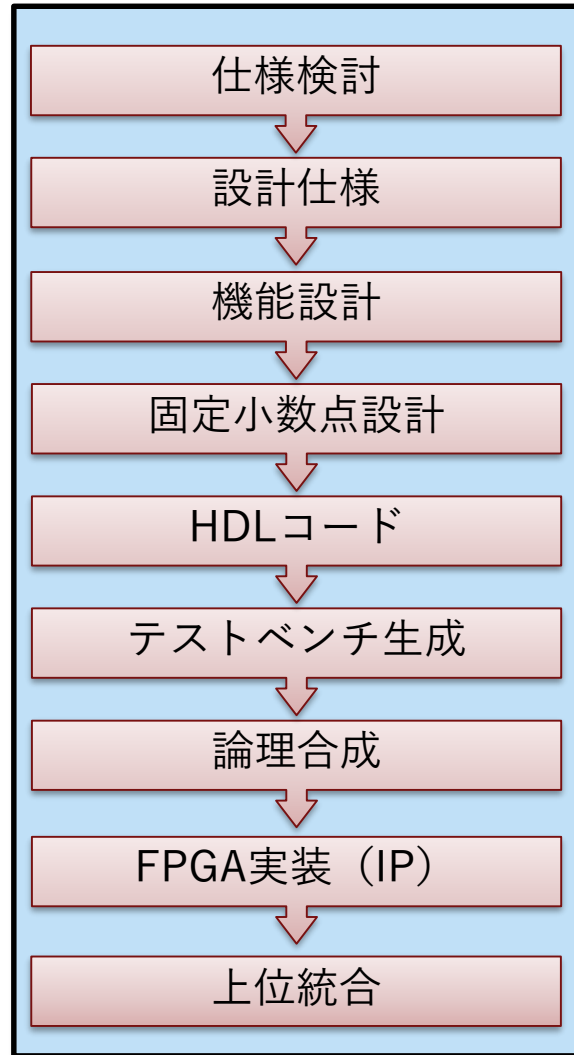


モデルの詳細化

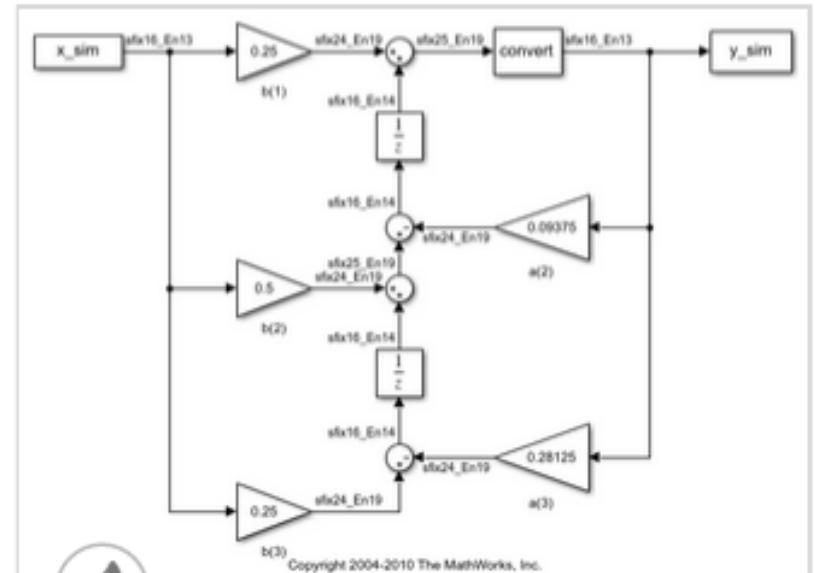
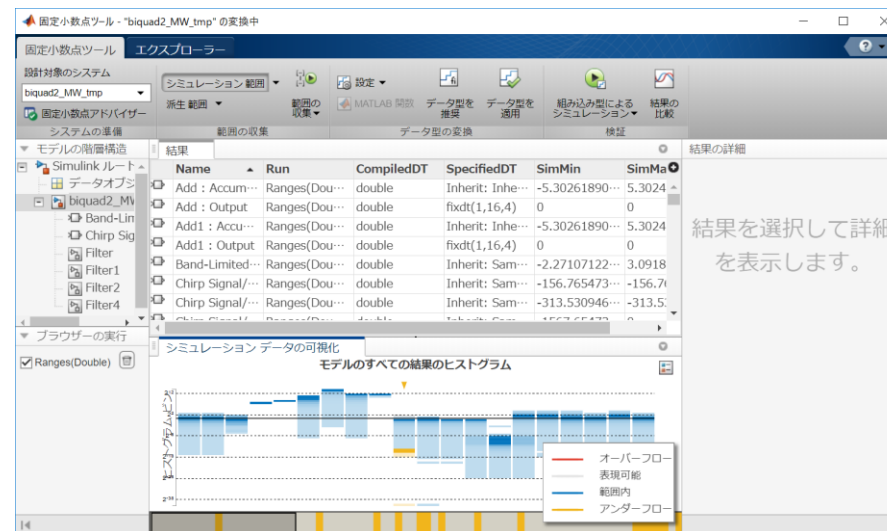


- Communications Toolbox™
- 5G Toolbox™
- LTE Toolbox™
- WLAN Toolbox™
- RF Blockset™
- Antenna Toolbox™
- Phased Array System Toolbox™
- Stateflow®
- SimEvents®

モデルの詳細化：固定小数点化



Fixed-Point Designer™

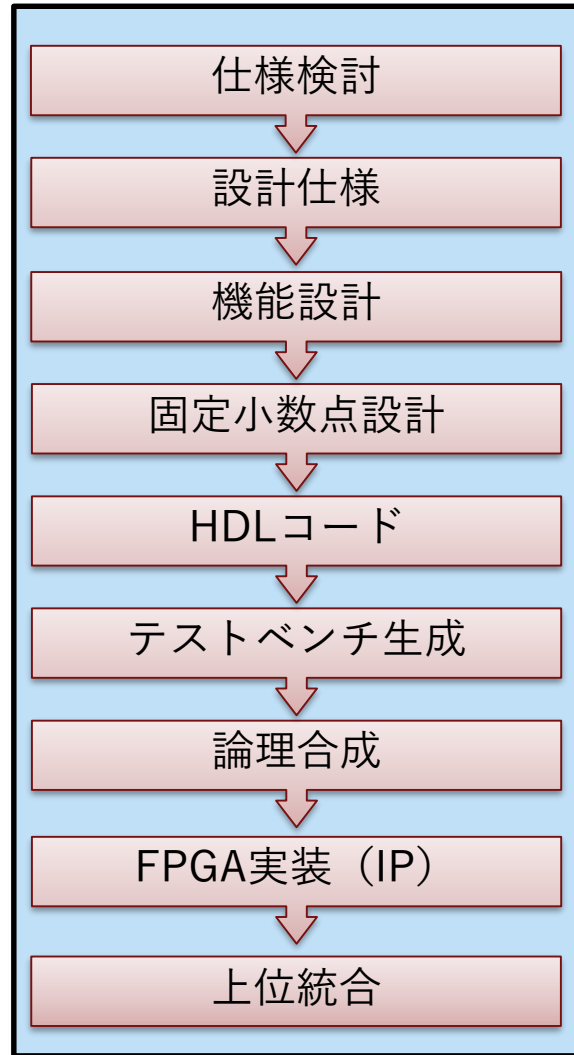


Develop Fixed-Point Algorithms

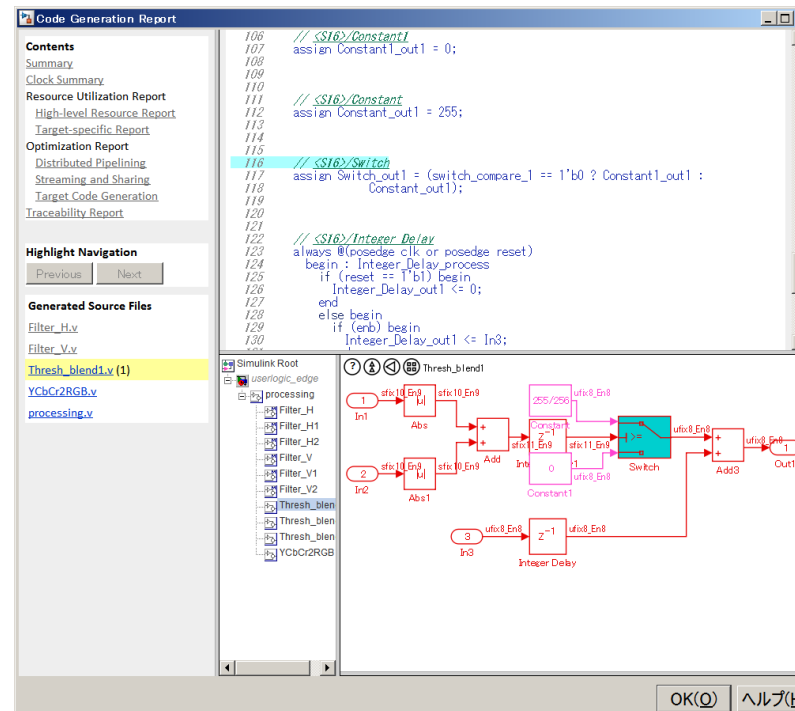
Develop and verify a simple fixed-point algorithm.

[Open Script](#)

モデルの詳細化 : HDL コード生成



HDL Coder™



HDL Implementation of LTE OFDM Modulator and Detector

The diagram illustrates the HDL implementation of an LTE OFDM Modulator and Detector. It shows two main blocks: 'LTE_Modulator_HDL' and 'LTE_Detector_HDL'. The modulator block takes inputs like 'dataIn' and 'freqOffset' and outputs 'dataOut' and 'freqOffset'. The detector block takes inputs like 'dataIn' and 'freqOffset' and outputs 'dataOut' and 'freqOffset'. The diagram also shows the 'Open LTE initialization script' and 'Open LTE initialization script' blocks. The copyright notice 'Copyright 2015 The MathWorks, Inc.' is visible.

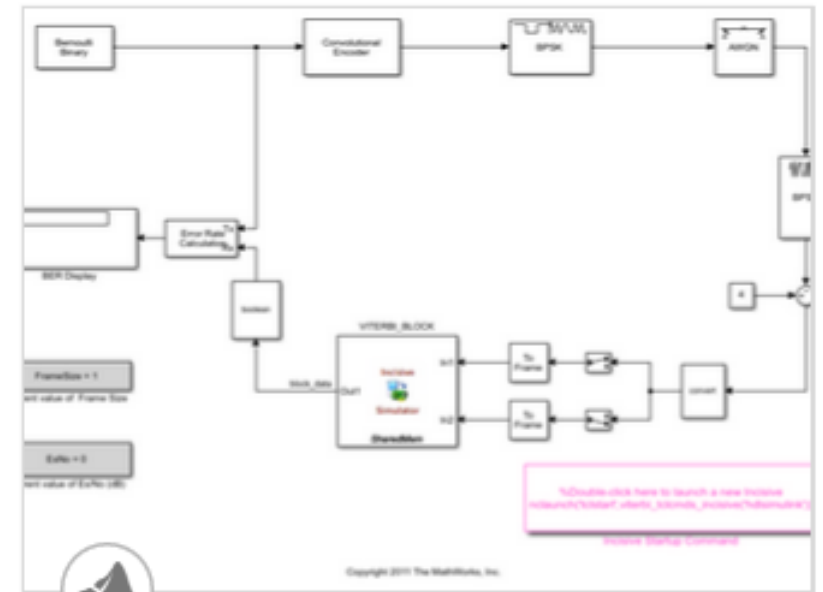
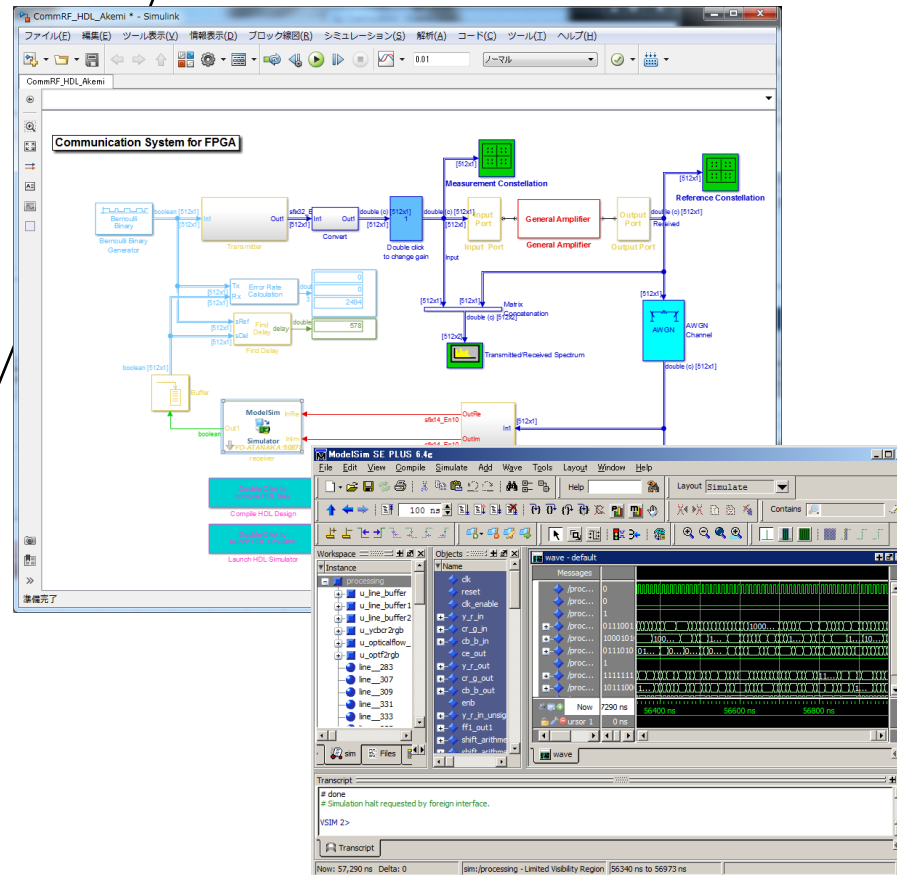
HDL Implementation of LTE OFDM Modulator and Detector

Build an LTE compliant OFDM Modulator and Detector for implementation with HDL Coder™, and use LTE Toolbox™ to verify the

[Open Model](#)

モデルの詳細化 : HDL 検証

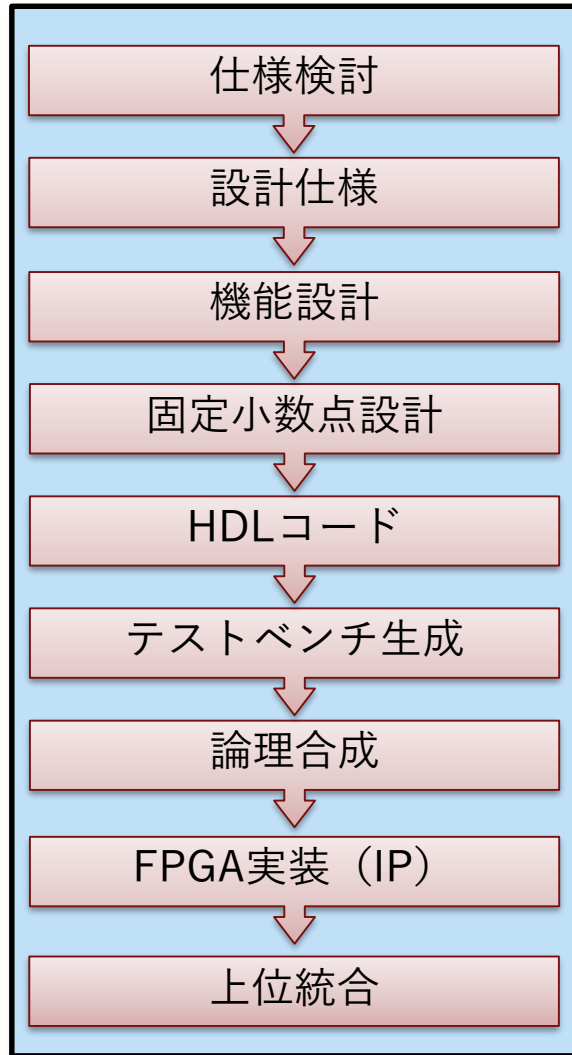
HDL Verifier™



Verify Viterbi Decoder Using HDL Cosimulation

Use HDL Verifier™ in conjunction with Mentor Graphics ModelSim®/QuestaSim® or Cadence Incisive®/Xcelium® to

モデルの詳細化 : FPGA In the Loop Simulation



HDL Verifier™

Host
Compute

Simulink

FIL
Block

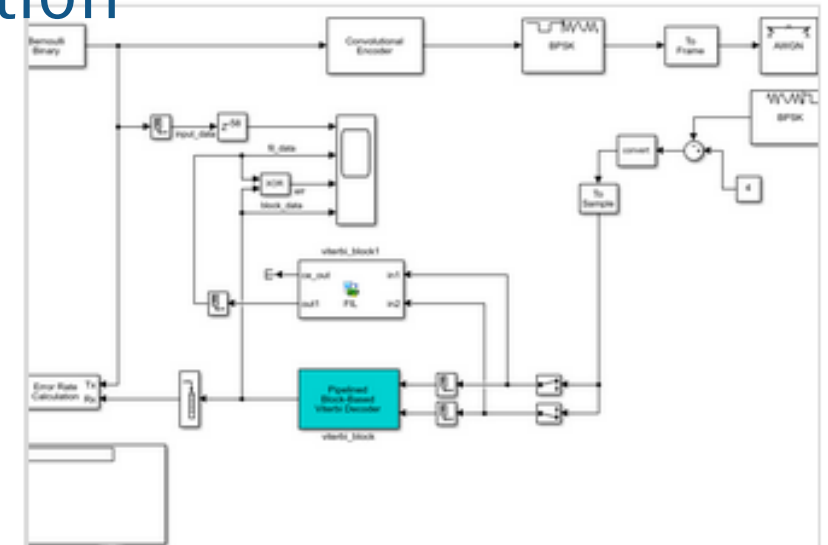
Ethernet

Board

FPGA

FIL
Interface

User
Design

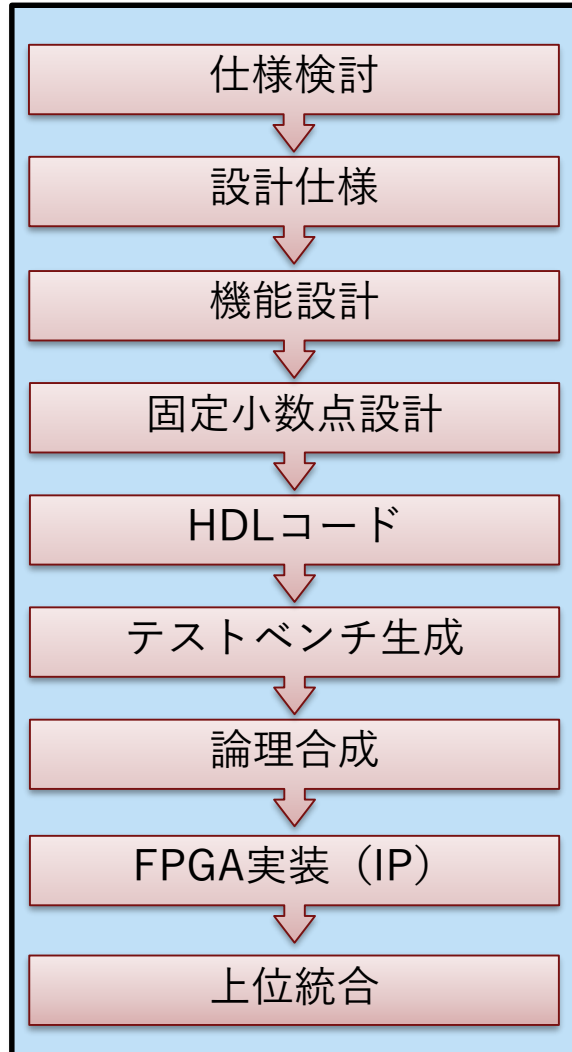


**Accelerating
Communications System
Simulation Using FPGA-in...**

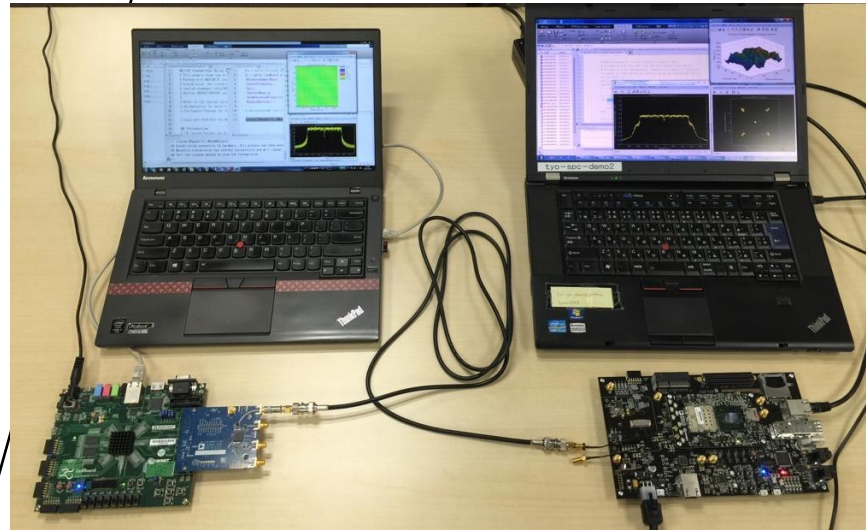
Uses FPGA-in-the-Loop (FIL) simulation to accelerate part of a communications system. The application uses the Viterbi

[Open Model](#)

モデルの詳細化：プロトタイピング



Embedded Coder®
ハードウェアサポートパッケージ



rm

RF Signal Generation

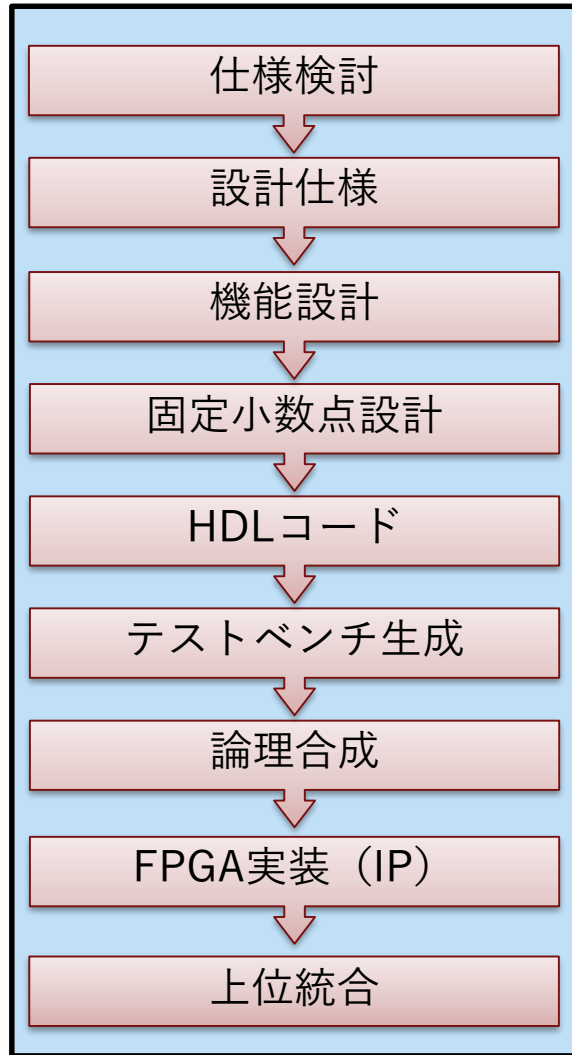
HW Support package

SDR platform

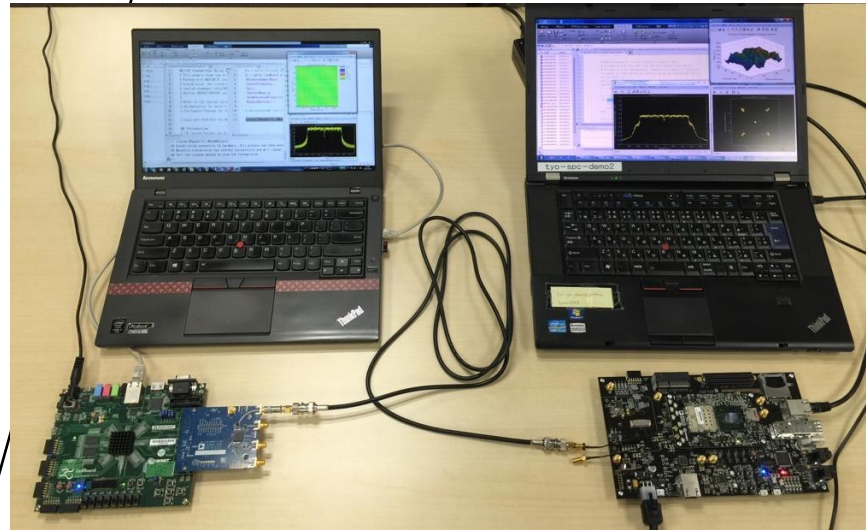
LTE Transmitter using Zynq-based Software-Defined Radio (SDR)

Use the Xilinx® Zynq-based radio support package with MATLAB® and LTE Toolbox™ to generate an LTE transmission. If you have a

モデルの詳細化：プロトタイピング



Embedded Coder®
ハードウェアサポートパッケージ



RF Signal Capture

HW Support package

SDR platform

LTE Receiver using Zynq-based Software-Defined Radio (SDR)

Use the Xilinx® Zynq-based radio support package with MATLAB® and LTE Toolbox™ to decode the master information block (MIB) and

アジェンダ

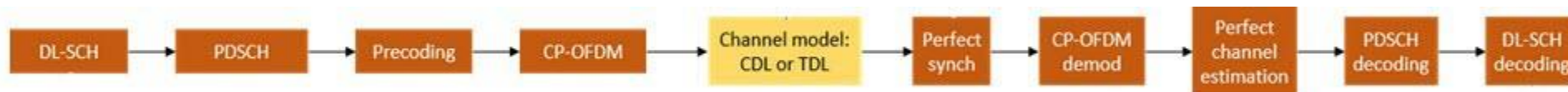
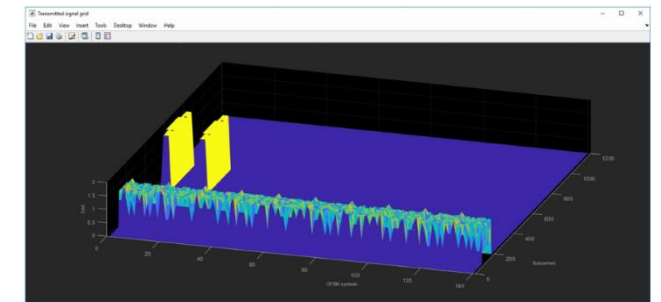
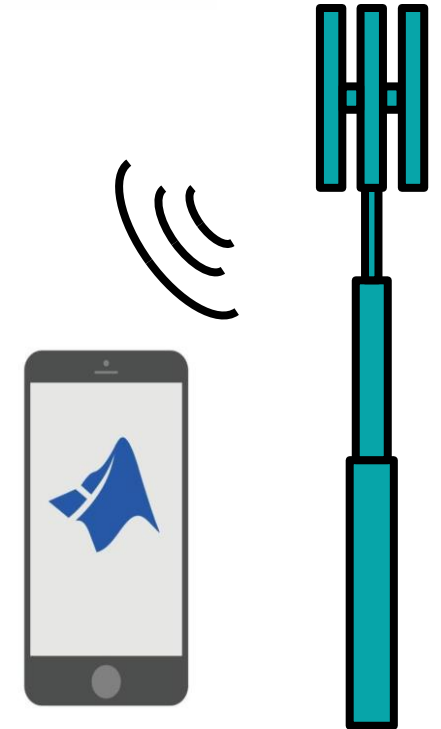
- 関連オプション製品の紹介
- 関連新機能の紹介

5G Toolbox™ リリース

R2018b

- 波形生成
- ダウンリンク処理 - 送信および受信
- TDLおよびCDLチャネルモデル
- 物理チャネルと信号
- リンクレベルのシミュレーションとスループットの測定
- 同期バースト
- セル検索手順
- リファレンスデザイン

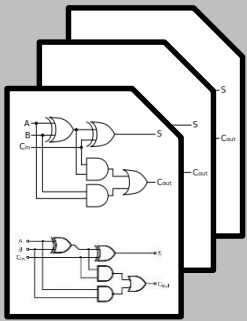
すべての機能がオープンでカスタマイズ可能なMATLABコード
C/C++コード生成対応



HDL Coder™ 新機能

Verilogのインポート

R2018b



Custom Verilog files

Import
HDL

Simulink

```

reg signed [31:0] Delay6_out1; // int32
wire signed [31:0] Add2_out1; // int32
reg signed [31:0] Delay8_out1; // int32

assign X_re_1 = X_re;
assign Y_re_1 = Y_re;
assign Product_out1 = X_re_1 * Y_re_1;

assign enb = clk_enable;

always @(posedge clk)
begin : Delay3_process
if (reset == 1'b1) begin
Delay3_out1 <= 32'sb00000000000000000000000000000000;
end
else begin
if (enb) begin
Delay3_out1 <= Product_out1;
end
end
end

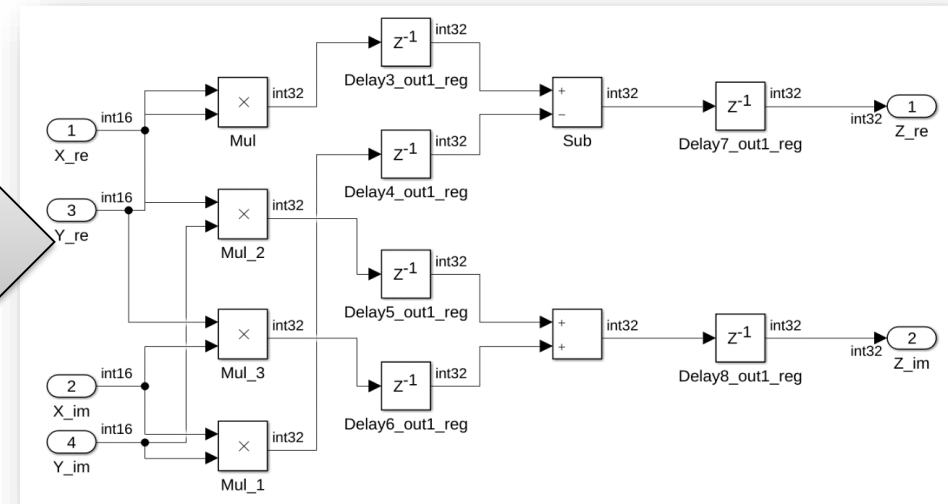
assign X_im_1 = X_im;
assign Y_im_1 = Y_im;
assign Product1_out1 = X_im_1 * Y_im_1;

always @(posedge clk)
begin : Delay4_process
if (reset == 1'b1) begin

```

Pipelined complex
multiplier
(Verilog code snippet)

既存の資産利用がより簡単に
Simulinkでシミュレーション、最適化

Import
HDL

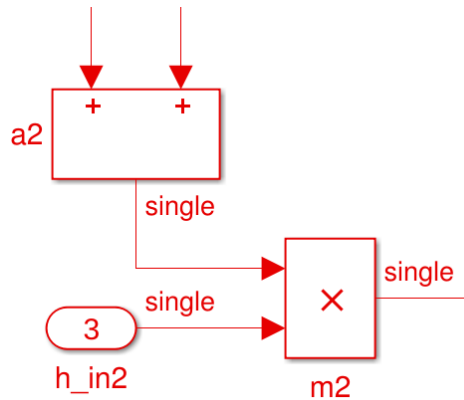
Auto-generated Simulink
model in 18b

HLD Coder™ 新機能

サポートデータタイプの拡張 (Native Floating Point)

R2016b

Single Precision



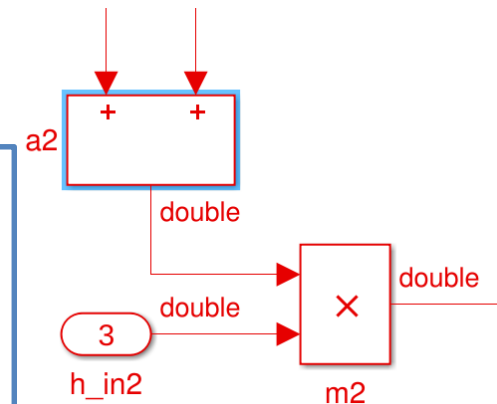
Rich operator support

- Sum
- Add
- Subtract
- Bias
- Gain
- Product
- Divide
- Abs
- Unary Minus
- MinMax
- Sum of Elements
- Product of Elements
- Dot Product
- Sqrt
- Reciprocal Sqrt
- Assignment
- Vector Concatenate
- Reshape
- Complex to Real-Imag
- Real-Imag to Complex
- Math Function
- Trigonometric Function

exp
Log
Log10
10^u
Pow
hypot

R2018b

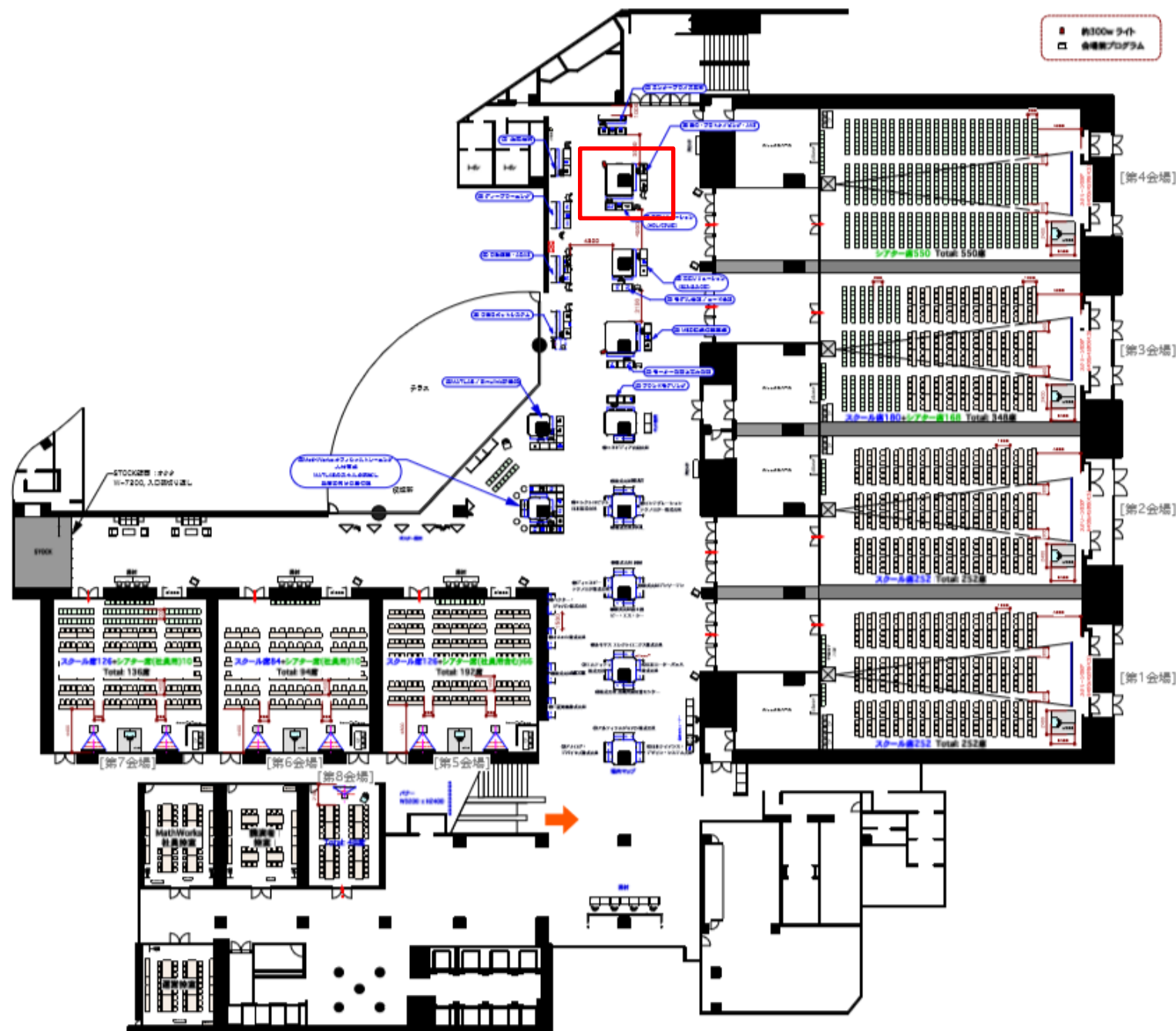
Double Precision



Radar Applications

- Add
- Sum
- Subtract
- Product
- Gain
- Divide
- Reciprocal
- Relational Operator
- Data Type Conversion
- Abs
- Unary Minus

展示ブース



まとめ

- 高抽象度でのシミュレーション環境
 - Communications Toolbox™
 - 5G Toolbox™
 - LTE Toolbox™
 - WLAN Toolbox™
 - etc.
- プロダクト化までの開発環境
 - Fixed-Point Designer™
 - HDL Coder™
 - HDL Verifier™
 - etc.
- バージョンアップごとにお客様のニーズを反映した製品のリリースと新機能