

FPGAによるインバータ制御 設計の基礎と実際

2020年2月21日

Mywayプラス株式会社

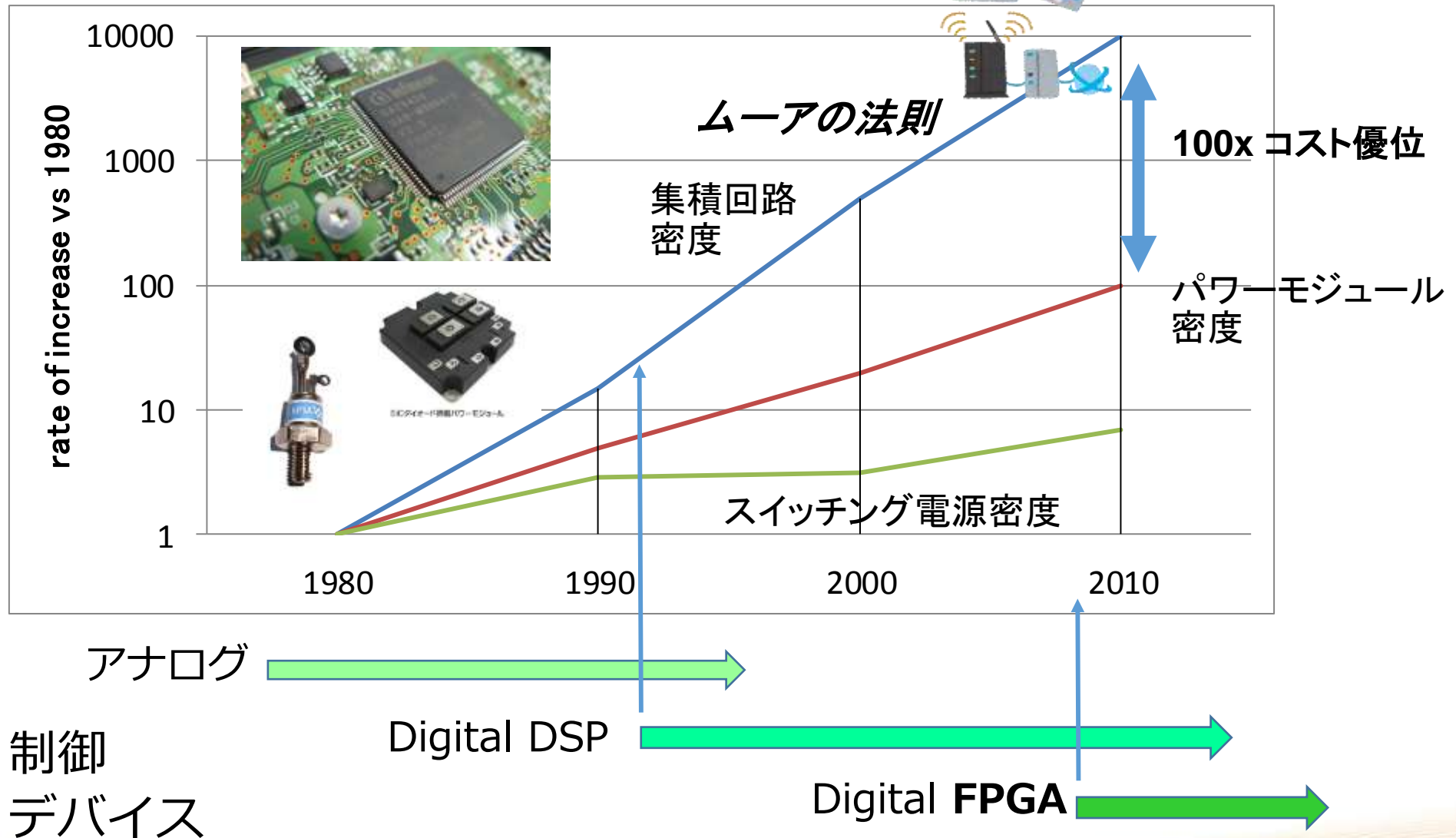
松野 知愛

目次

1. FPGA概要
 - 1-1. デジタルデバイス全般
 - 1-2. FPGAの構造と設計フロー
 - 1-3. 効率的な開発体制
2. システム設計の勘所
 - 2-1. FPGAを含むシステム
 - 2-2. FPGAのタスク切り出し
 - 2-3. パイプライン化と並列化
3. コーディングの勘所
 - 3-1. アーキテクチャの基本形
 - 3-2. センスの良いコーディング
 - 3-3. より高度なコーディング
4. FPGAプラットフォーム紹介
 - 4-1. PE-Expert4ご紹介
 - 4-2. 製品開発事例
 - 4-3. アプリケーション事例(参考)

途中、演習および休憩を入れます

時代はデジタルリッチへ (自己紹介)



1. FPGA概要

設計フローと必要スキル

1. FPGA概要

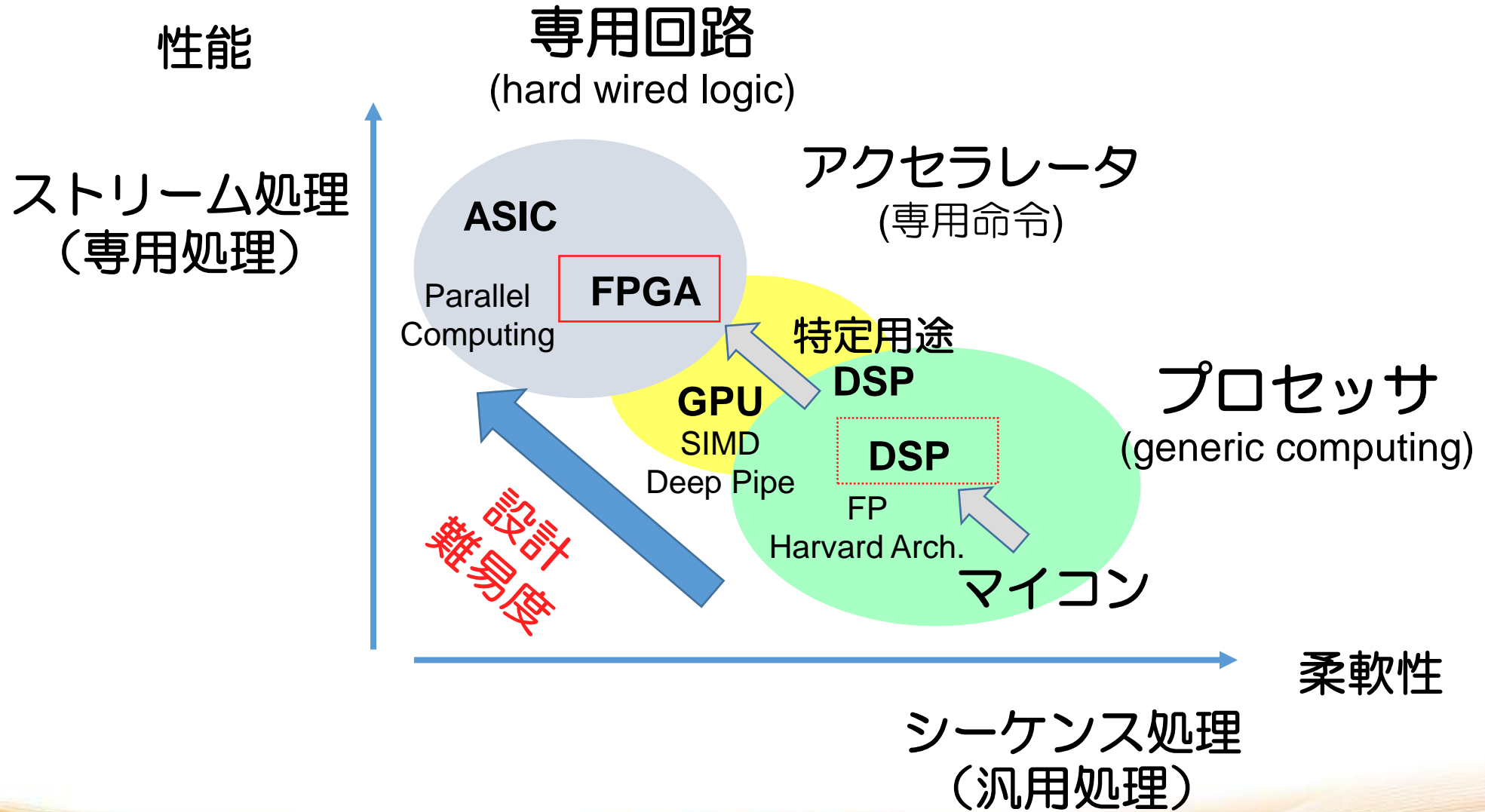
1-1. デジタルデバイス全般

1-2. FPGAの構造と設計フロー

1-3. 効率的な開発体制

デジタルデバイス全般

様々な演算デバイス



パワーレ制御に必要なスペック

- 演算速度
 - 動作周波数, コア数, 専用命令
- レイテンシ
 - 割り込み遅延, 内部バス
- インターフェース
 - 周辺回路(AD, GPIO) 、専用回路(PWM) 、外部バス

パワエレ制御～演算デバイス比較～

	マイコン	DSP	FPGA	ASIC	アナログ
演算能力	△	○→◎	◎	◎	◎
レイテンシ	△	○	◎	◎	◎
インターフェース	△	△→○	◎	◎	○
汎用性(適用アルゴリズム柔軟性)	◎	◎	◎	◎	×
開発工数	◎	◎	△→○	×	○
部品単価	◎	○	△→○	◎	○
パワエレ用途	◎	◎	→◎	-	◎

× ←不利 有利→◎

高速処理・多ゲート制御の利点を活かせる
先端アルゴリズムに最適

FPGAによるイノベーション

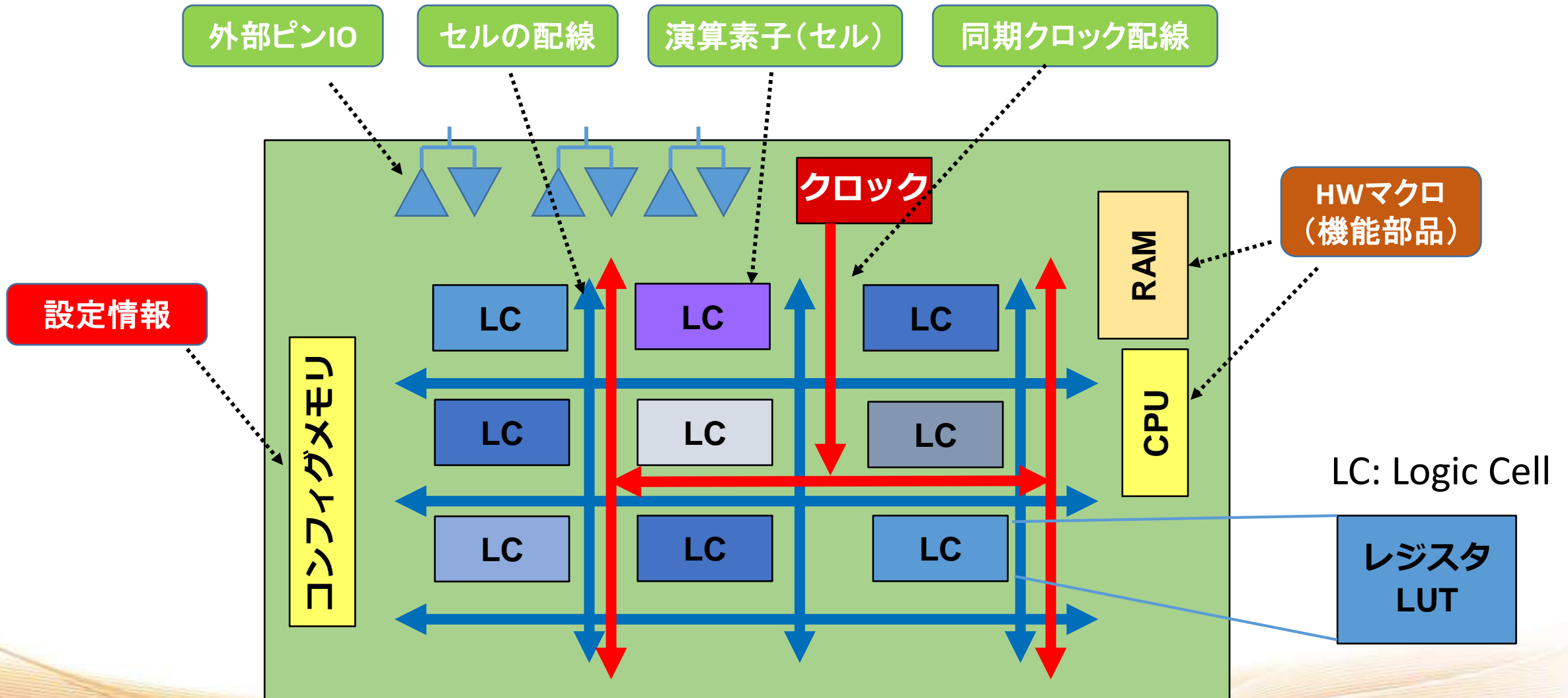
- 既存機能の置き換え：
 - 高速フィードバック制御
 - システム同期制御
- 新たな付加価値：モデルベース・デジタルツイン
 - 数学モデルのリアルタイム演算
 - 例、回路モデルの数値計算
 - モデルデータのパターンマッチング
 - 例、インダクタンス空間分布テーブルによる位置推定

FPGAによるパターンマッチングの例 (名古屋大 道木先生)

出典: 電気学会
自動車研究会資料
2011

FPGAの構造と設計フロー

FPGAの内部構成

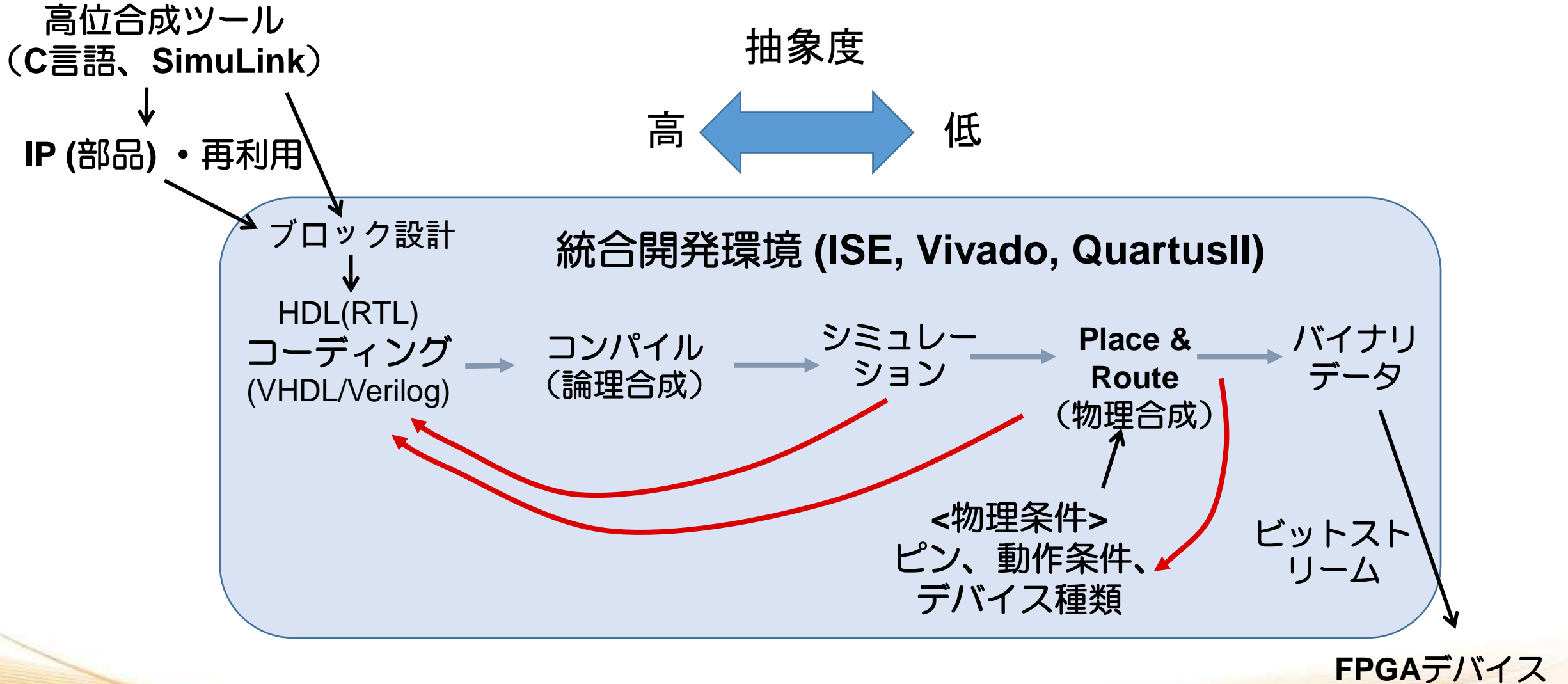


設計で注意するポイント

- クロック （同期設計、クロックソース）
- リセット （起動シーケンス）
- 電源 （起動シーケンス、ノイズ、安定動作）
- I/O （バンク設定、初期値、同期設計、ノイズ）
- RAM （ブロックメモリ・IP、使用率、チップスコープ）
- JTAG/デバッグ（Flash選定、システム構成）
- 機能 （タイミングチャート、ステートマシン）

全てFPGA設計仕様書でカバーすべき項目

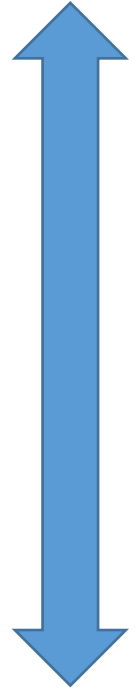
FPGAの設計フロー



効率的な開発体制

デジタルシステムの開発抽象度

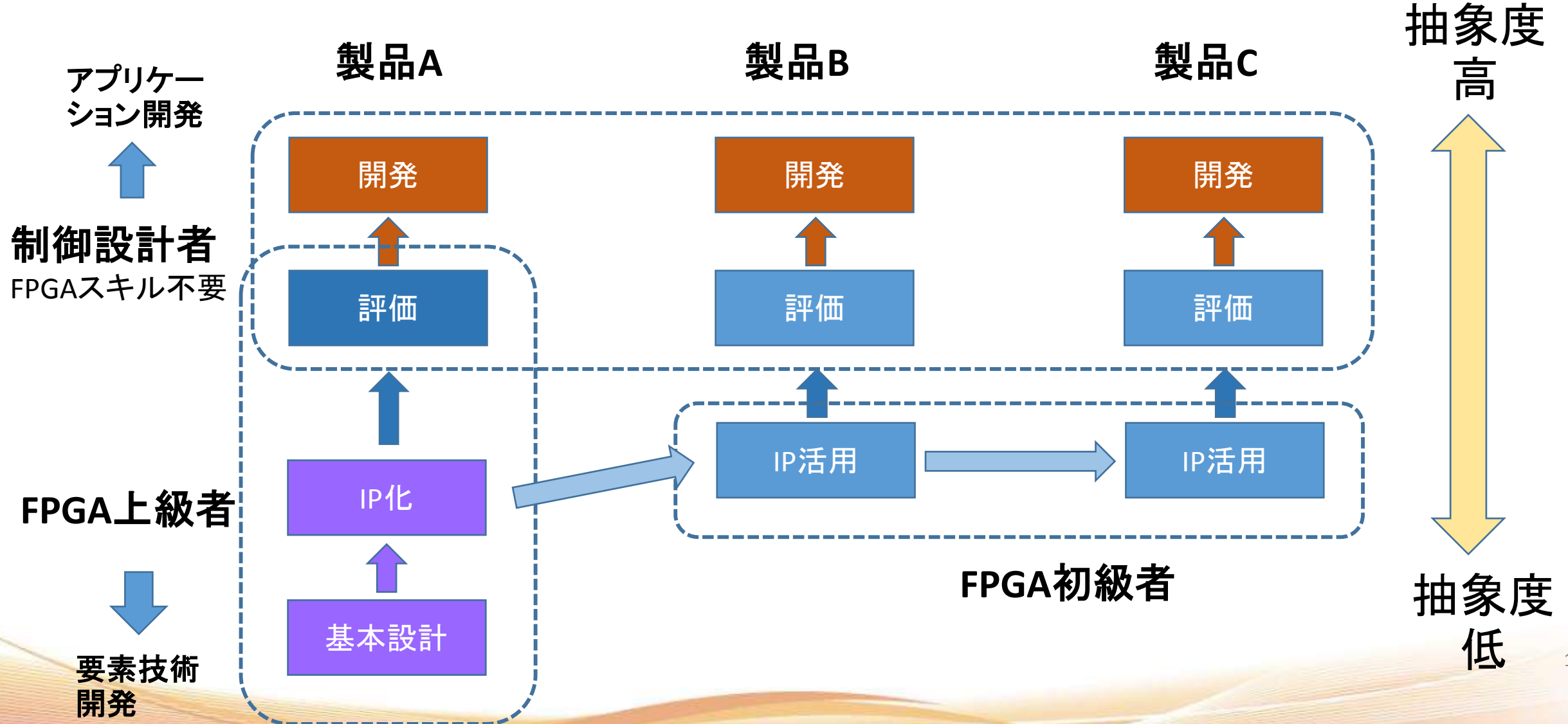
抽象度 高 (システム難易度 大)



- ビヘイビアモデル(イベント・ドリブン)
- (命令セットレベル (プロセッサ、ISA))
- RTL(レジスタ・トランスファ・レベル)(サイクル・アキュレート)
- ネットレベル、ゲートレベル(配置配線)
- トランジスタレベル(半導体、マスク、プロセス)

抽象度 低 (デバイス難易度 大)

抽象度と理想的な開発体制



演習1 VHDL設計の基本

設計環境の基本操作

演習資料を参照下さい

2. システム設計の勘所

2. システム設計の勘所

2-1. FPGAを含むシステム

2-2. FPGAのタスク切り出し

2-3. パイプライン化と並列化

FPGAを含むシステム

既存デバイスの限界

- 制御周期
 - キャリア割り込みルーチン許容時間
- 高速・精密・多ゲート制御
 - マルチレベル、インターリーブ、位相制御
- 高速システム同期制御
 - ユニット間高速通信によるシステム並列化

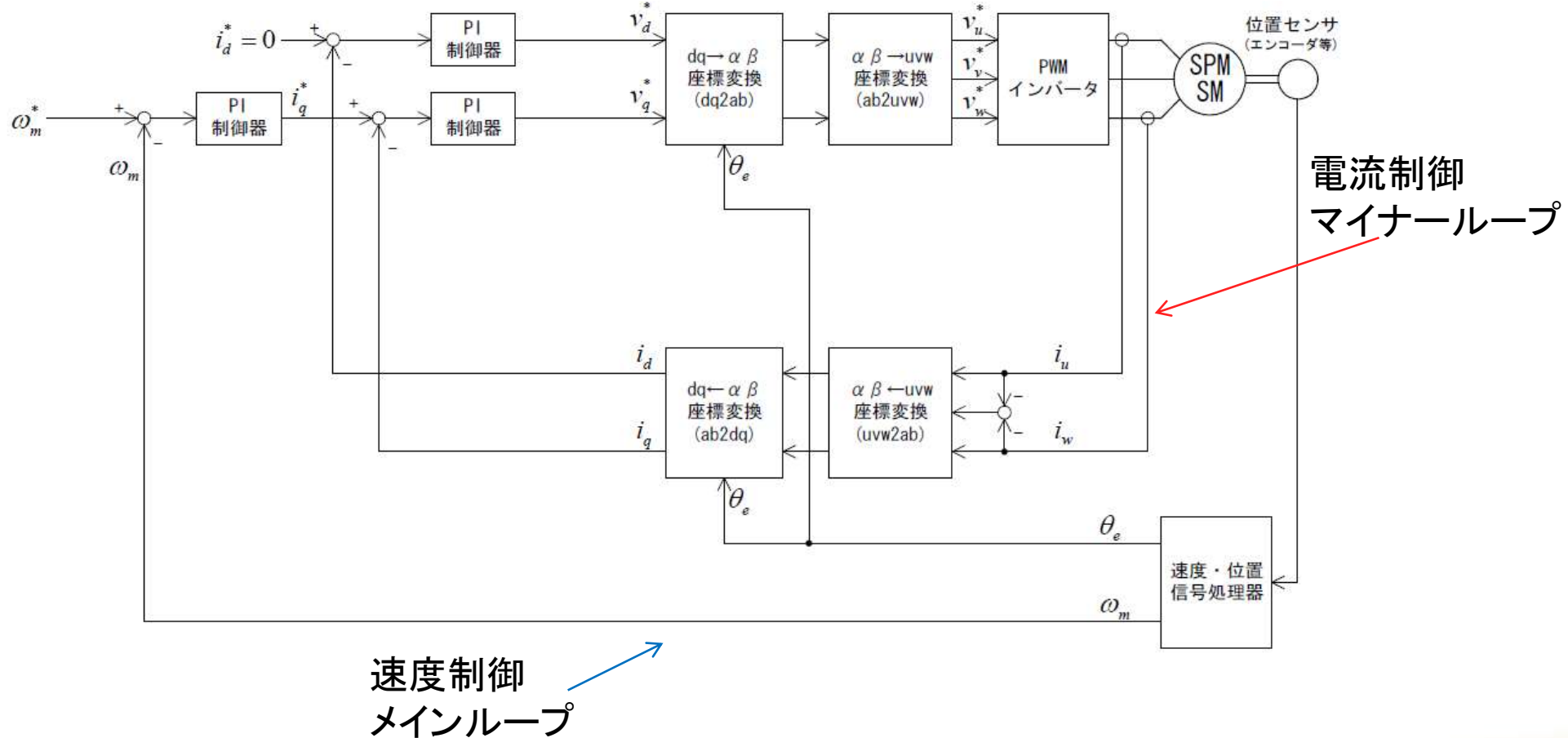
FPGAだからできること

時間と空間を自由に操る

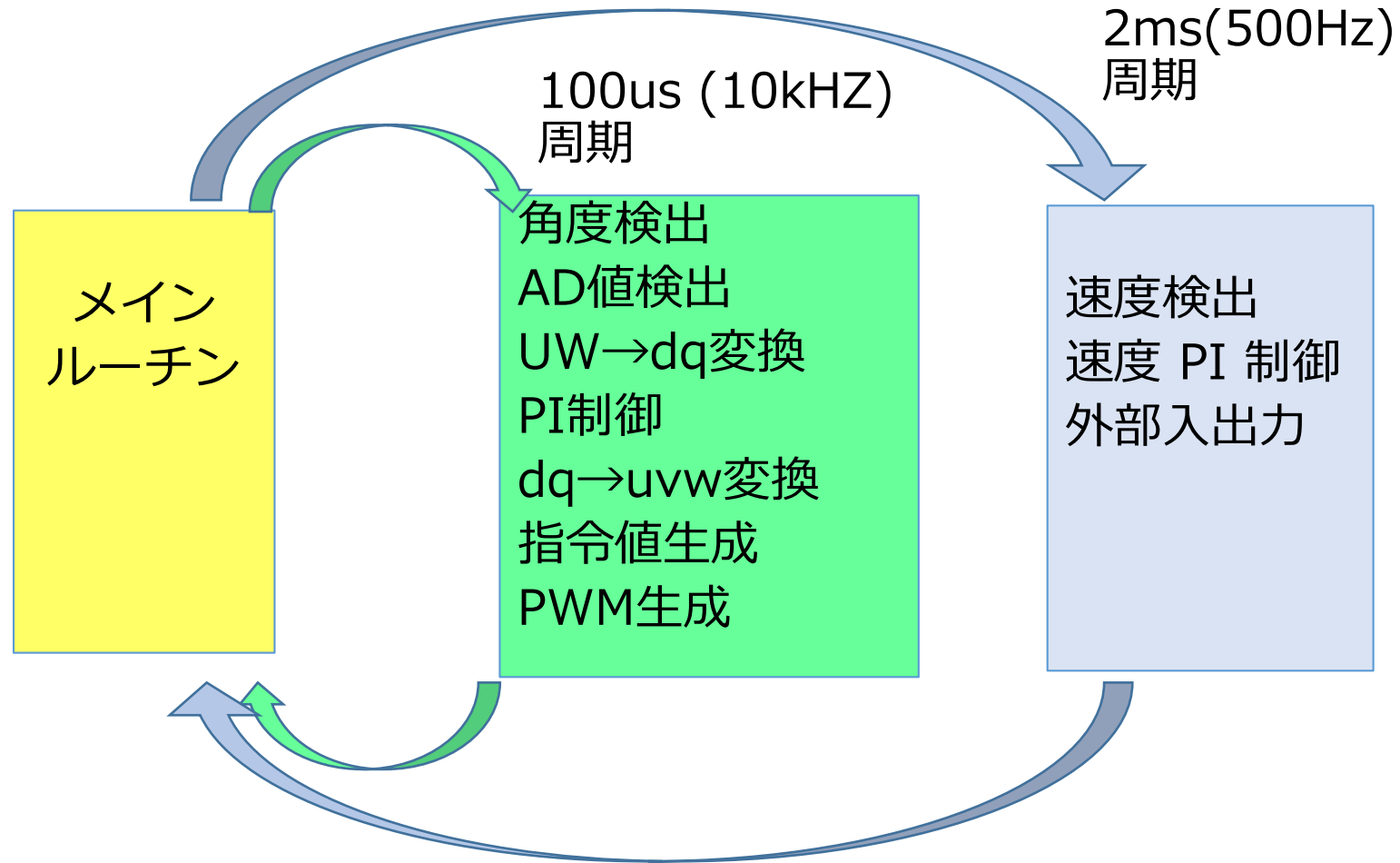
- 時間的(クロック)
 - パイプライン処理 (高速) → 通信
 - クロック単位の制御(精度) → 位相制御
- 空間的(ビット)
 - 超並列処理 (高速) → MMC、多ゲート
 - ビット単位の制御 (精度) → 高精度AD

FPGAのタスク切り出し

システム例：PM同期モータ ベクトル制御

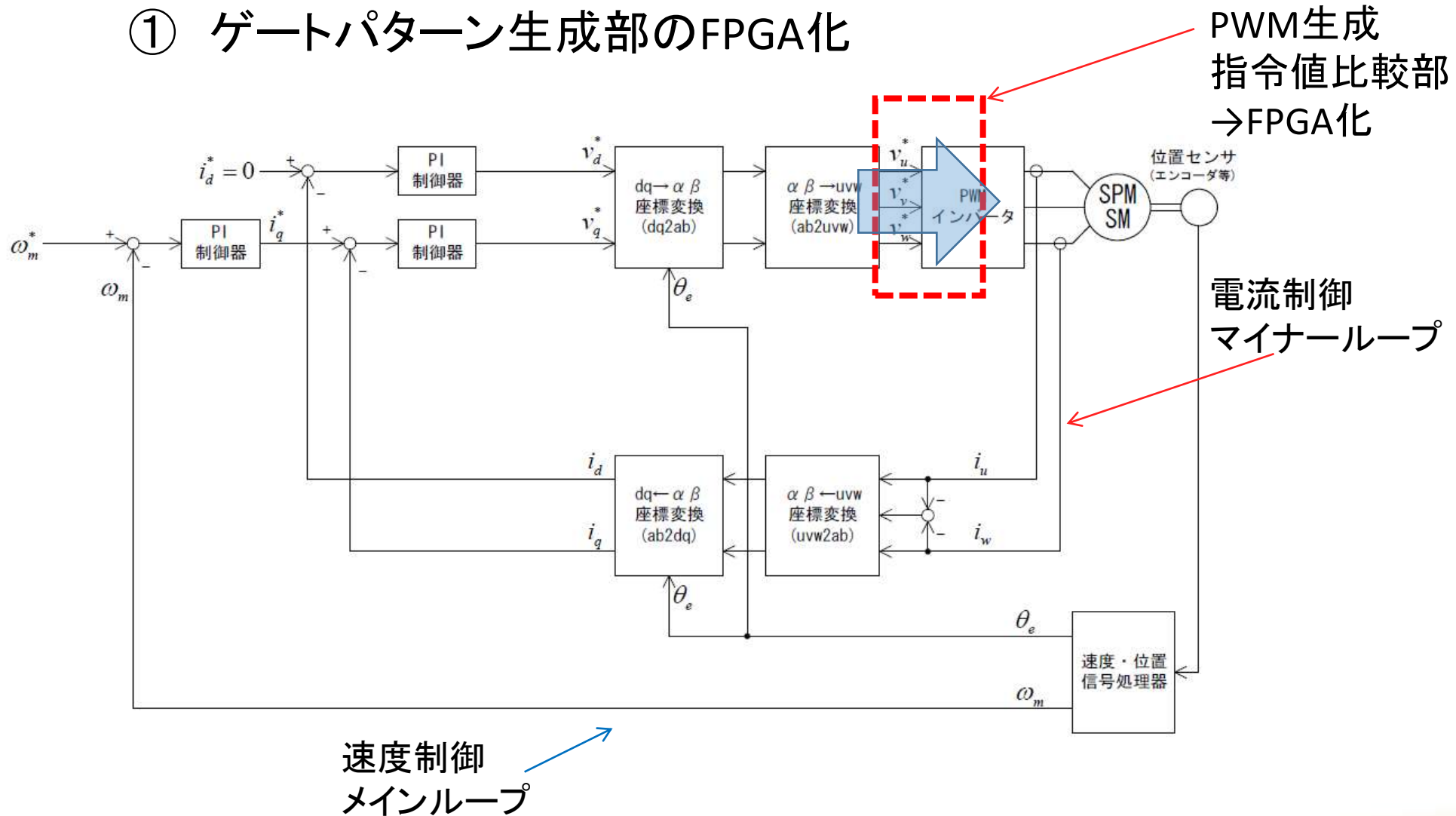


DSP 制御ソフトウェア構成

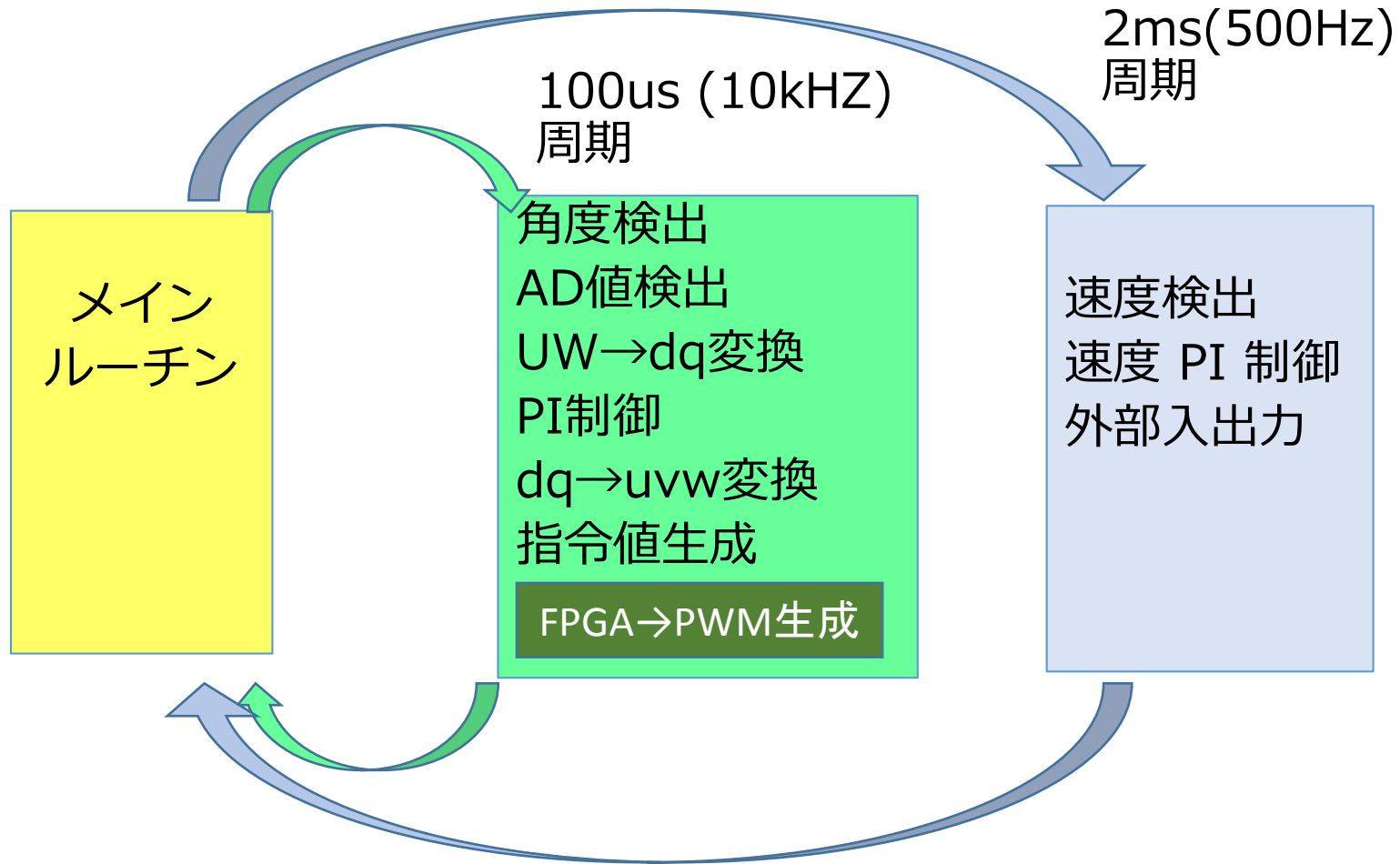


システム例：PM同期モータ ベクトル制御

① ゲートパターン生成部のFPGA化

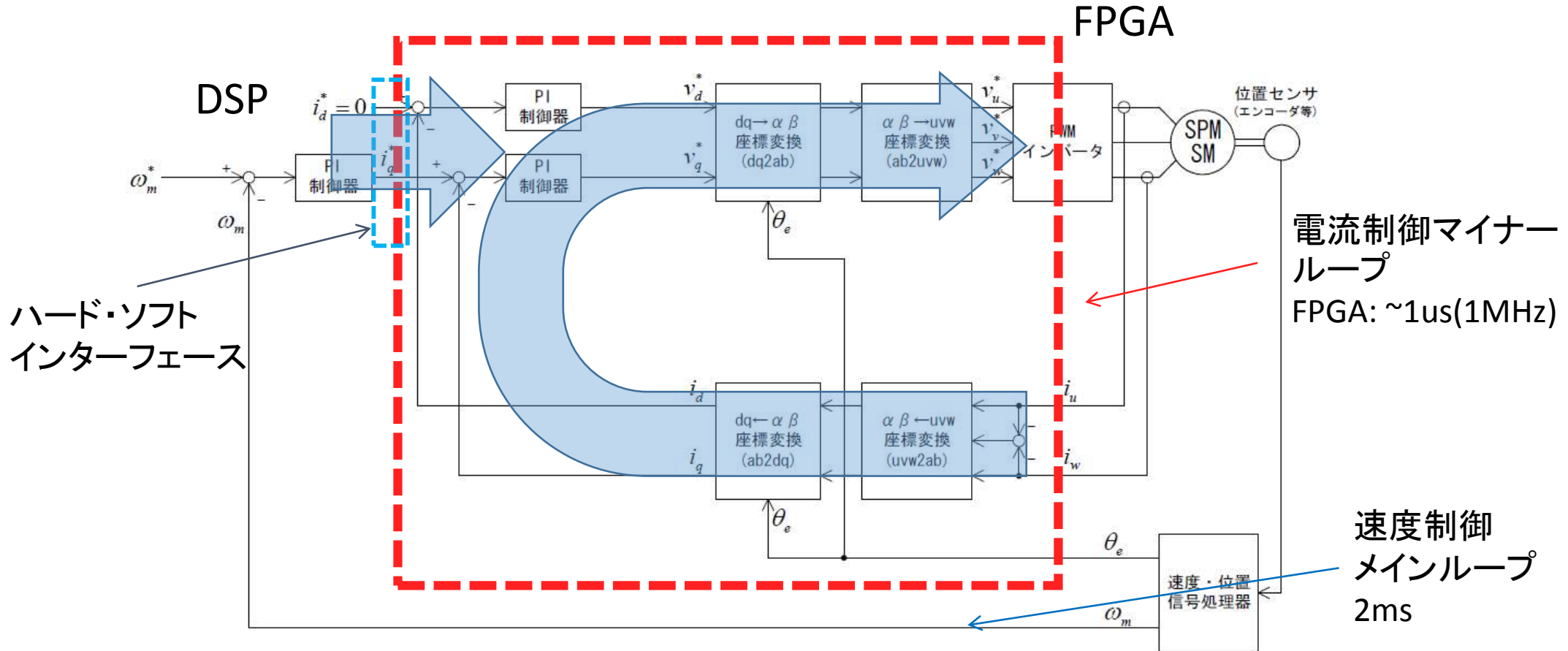


DSP + FPGA PWM生成ロジック



システム例：PM同期モータ ベクトル制御

②マイナーループ演算部のFPGA化の例



a) ハード・ソフトインターフェース b) 処理見積もり c) ビット精度

マイナーループPFGA化

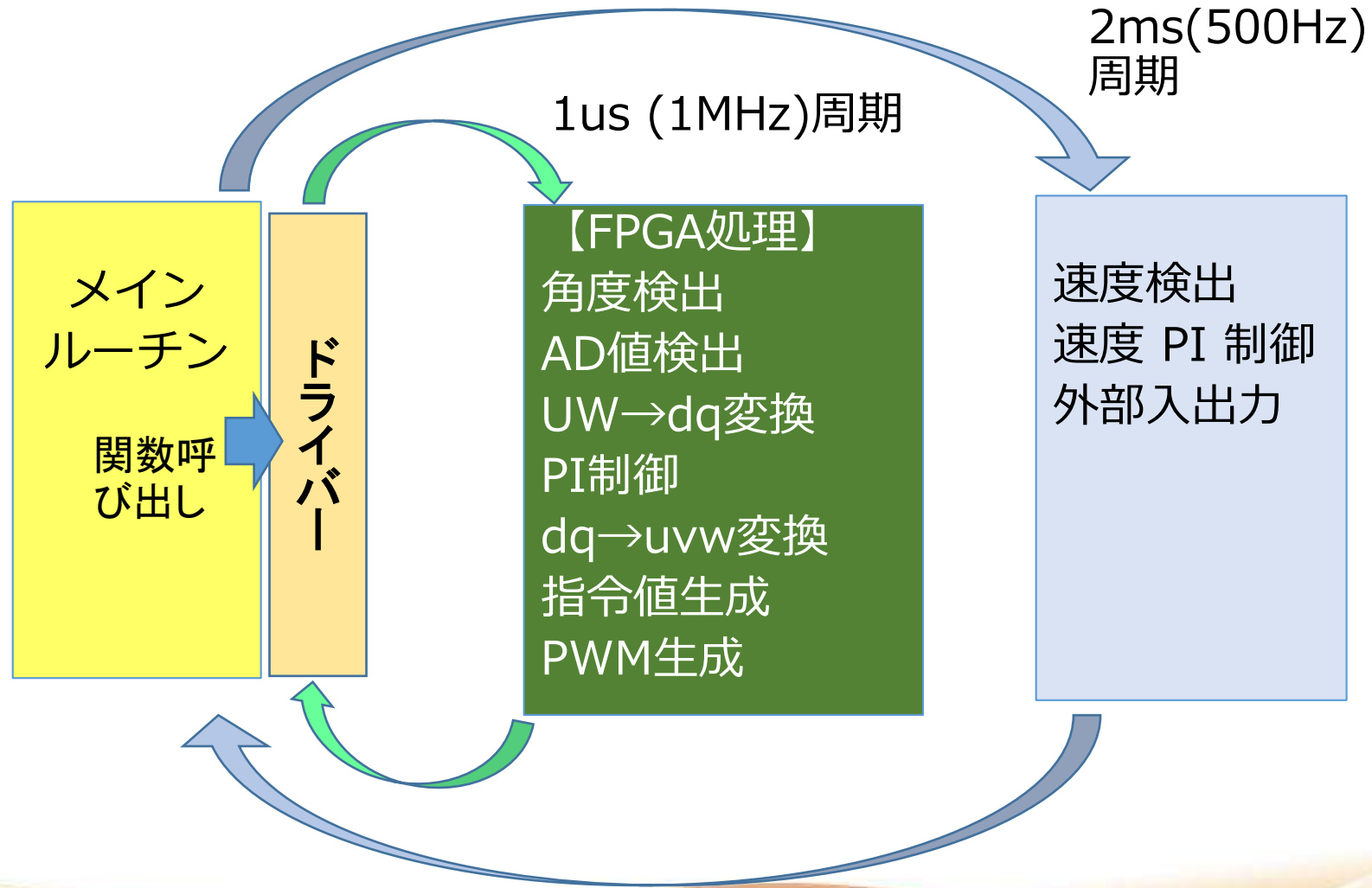
a) ハード・ソフトインターフェース

- 開発効率化のために
 - 仕様の明確化・標準化(機能、性能、使い方)
 - ドライバソフトウェア・試験環境をあわせて管理
 - シミュレーションモデル作成(ビット精度含む)



- 抽象度による階層化でインターフェースを定義し役割分担
 - FPGAエンジニア不在でも開発を進める

FPGA処理のドライバ化



b) 遅延見積もり

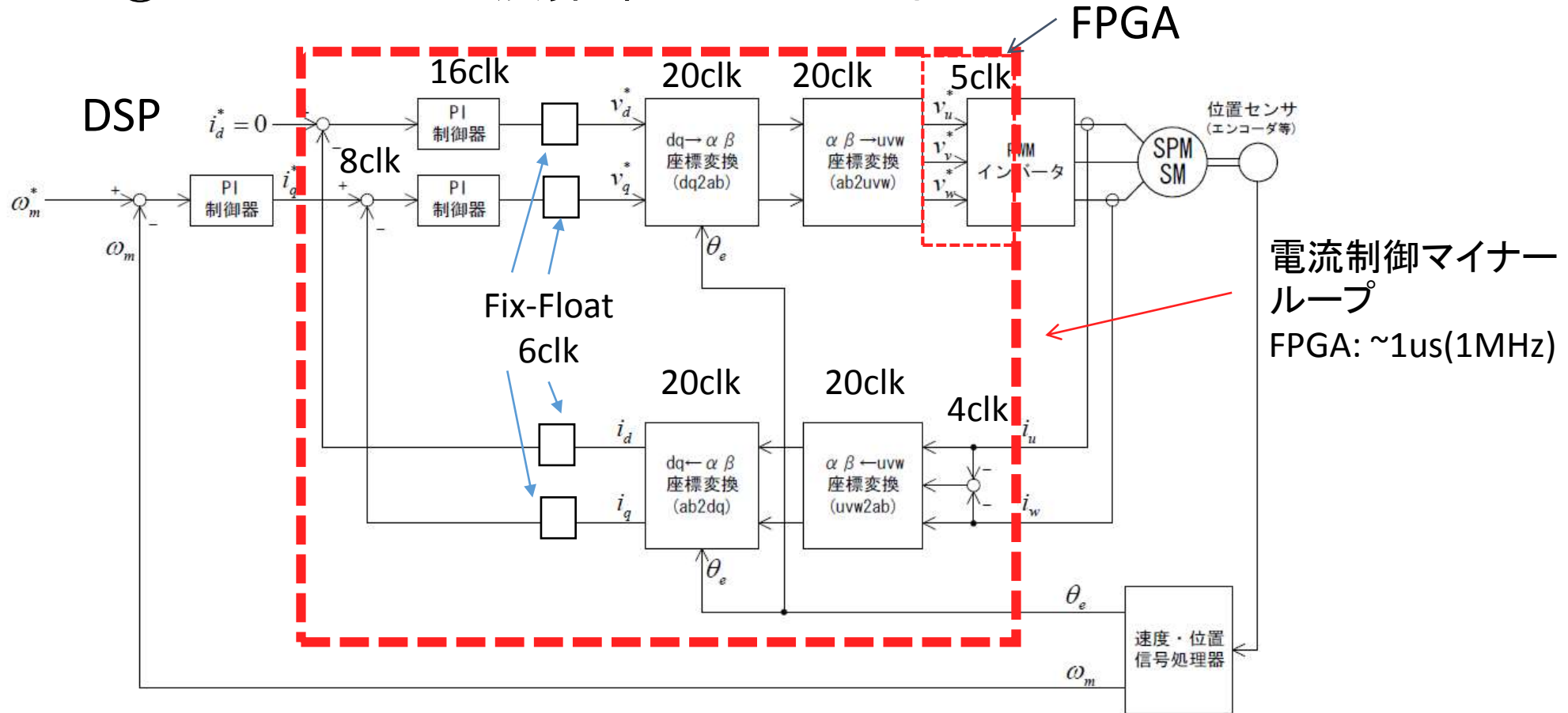
- 要求仕様に有った遅延
 - 100ns? 1us? 10us? 100ns? 実装が劇的に変化
 - オーバスペックに注意 → 不必要なリソース消費 (コストアップ・開発遅延)
- FPGA以外の遅延要素に注意 (むしろ支配的)
 - ADデバイス、シリアル通信、絶縁デバイス遅延、他
- さらなる高速化にはパイプライン化も検討 (後述)



- 明示的な遅延設計目標と適切な実装手段
- 推定・仮定よりも最新の实装結果 (最終コンパイル結果) で判断

演算遅延見積もりの例

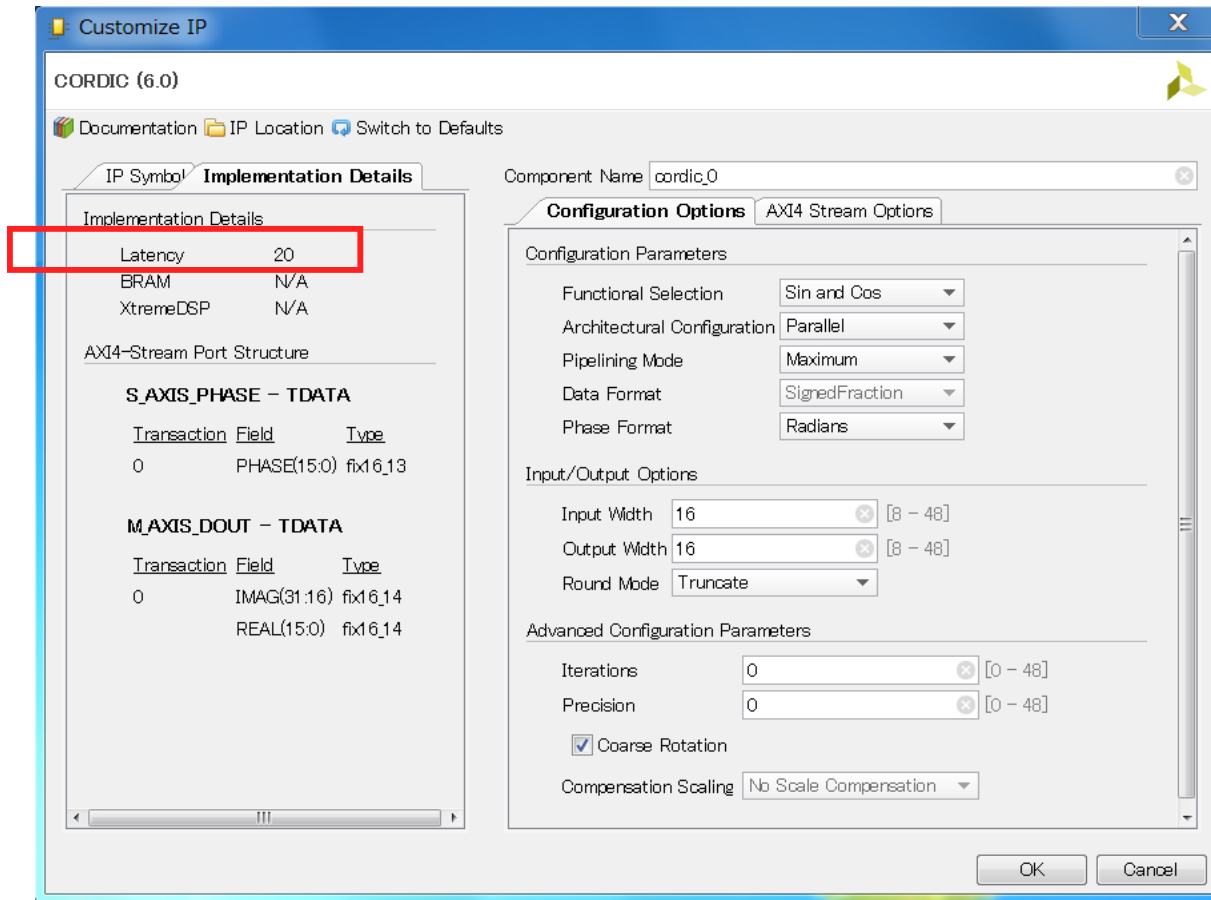
② マイナーループ演算部のFPGA化の例



遅延計算

(logic) $4+20+20+6+8+16+6+20+20+5 = 125 \text{ cycle} \times 5 \text{ (ns @200MHz)} = \sim 650\text{ns}$

IPブロックの遅延(例)



Vivado IP Integrator

各種演算器の作成
(Sin/cos の例)

Latency 20 clock

※演習4において
「IP Catalog」で確認下さい

c) 精度見積もり

- 事前に高位言語モデル(C、SimuLink)で評価
 - 高位合成ツール等
- 浮動小数点モジュールの実装
 - 特にPI制御などダイナミックレンジが大きい場合に有効
 - IP Catalog参照
- タイミングクリティカルならば固定化も検討

パイプライン化と並列化

さらなる高速化を目指す

パイプラインと並列処理

高速化手法

- 並列処理: 空間方向に処理分割
- パイプライン: 時間方向に処理分割

高速化の例

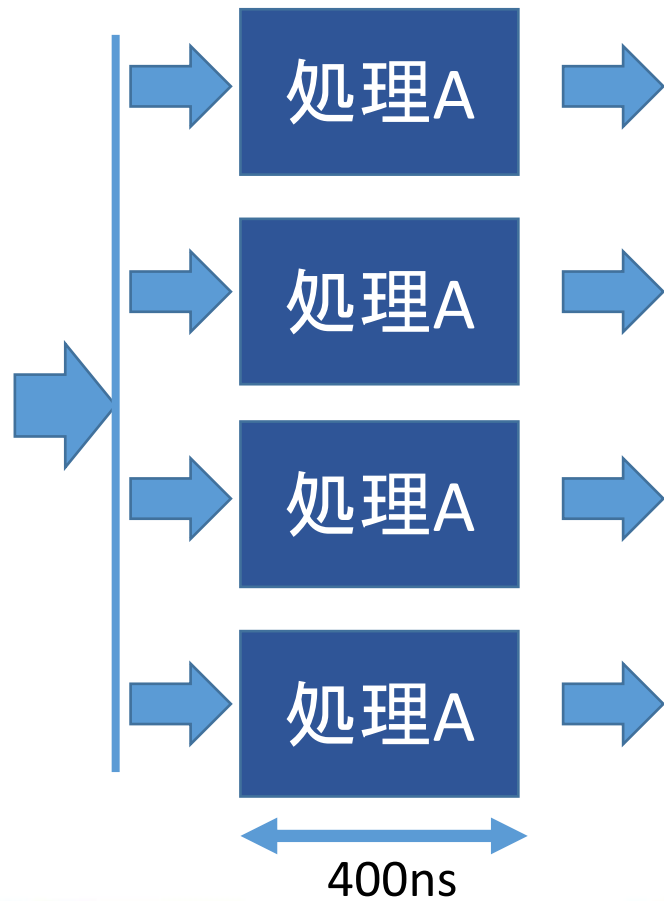
ロジック



スループット: 2.5MHz
レイテンシ: 400ns
リソース: x1

高速化 ① 並列処理

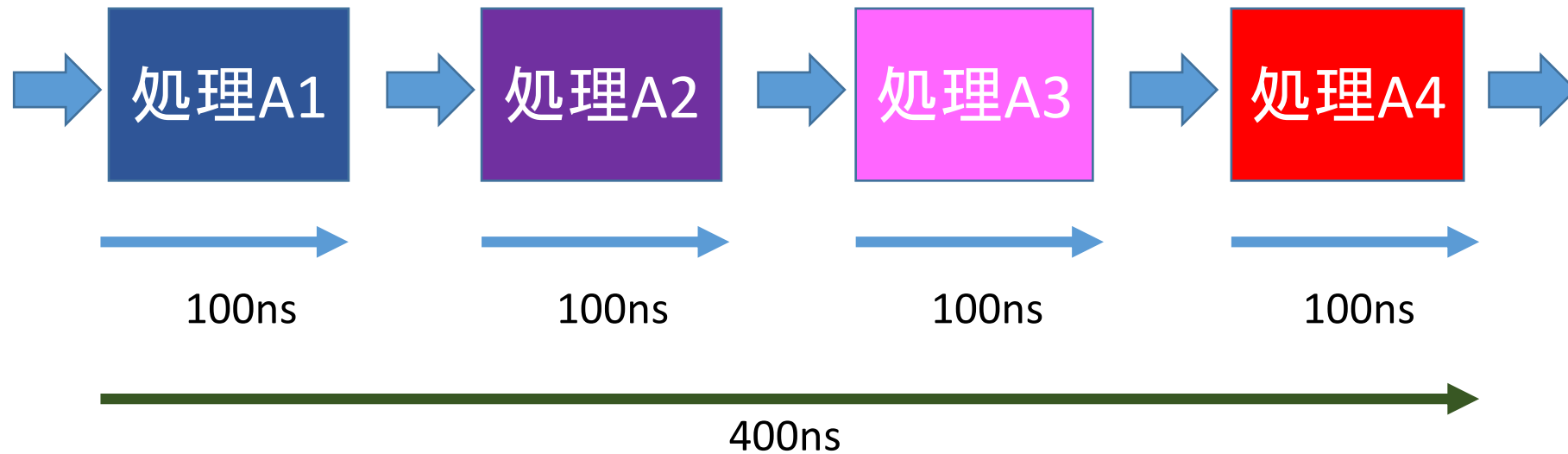
「依存関係のない」処理は分割し空間方向へ拡張



- スループット: 2.5MHz x4 → 10MHz
- レイテンシ: 400ns
- リソース: **x4**

高速化 ②パイプライン処理

「時間方向」の並列処理



スループット: 2.5MHz x 4倍 → 10MHz

レイテンシ: 400ns

リソース: **x1**

3. コーディングの勘所

「センスの良い」設計

3. コーディングの勘所

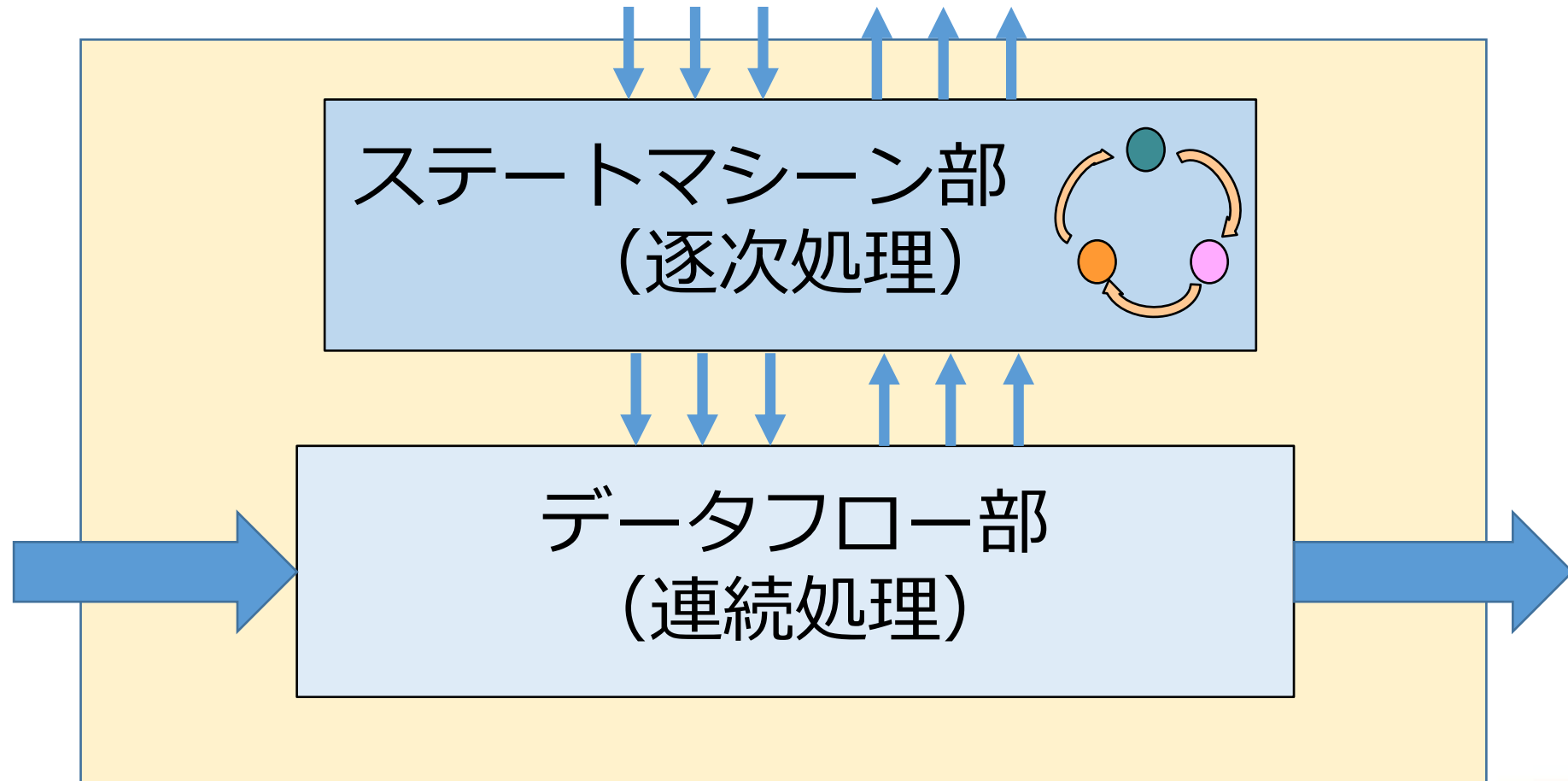
3-1. アーキテクチャの基本形

3-2. センスの良いコーディング

3-3. より高度なコーディング

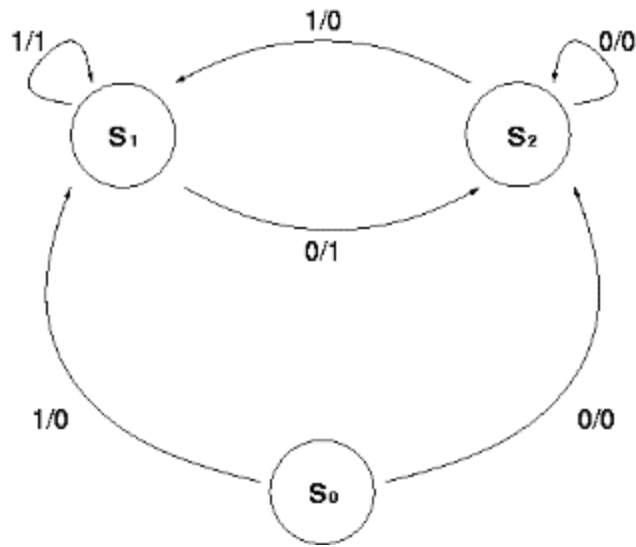
アーキテクチャの基本形

FPGA 一般的な論理回路構成

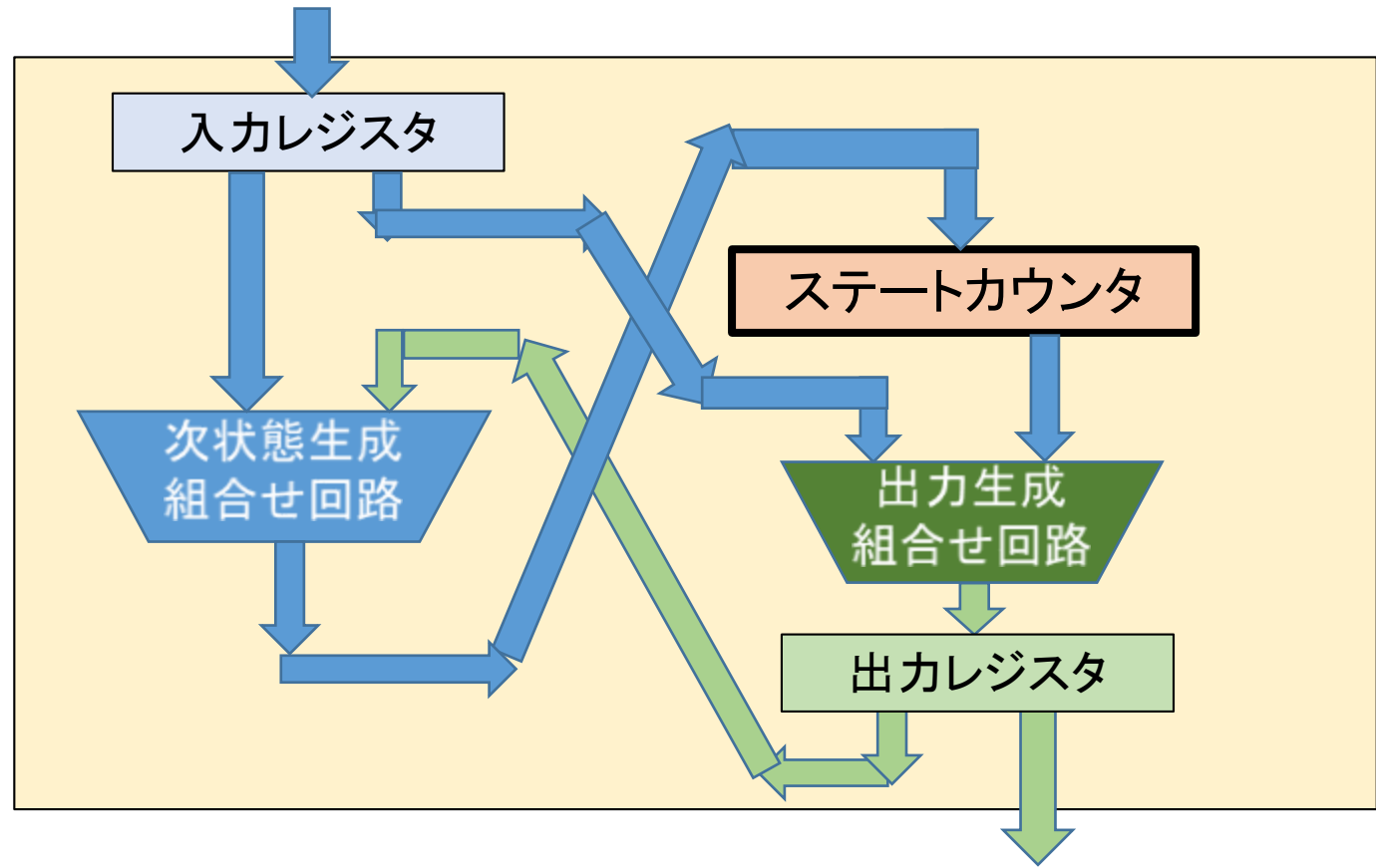


ステートマシン部

入力信号(内部状態、外部入力、設定パラメータ)

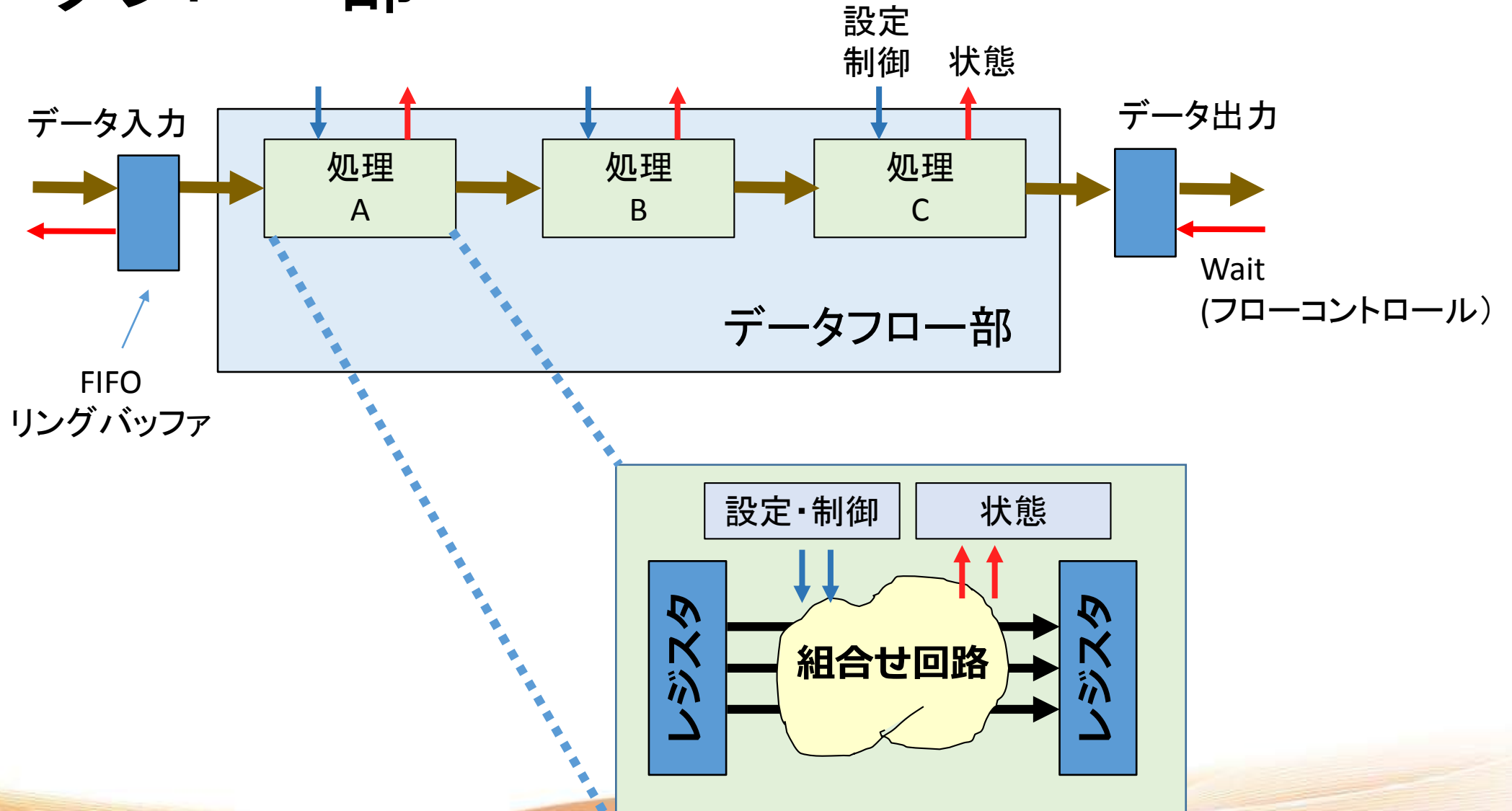


ステートマシン(例)
(状態遷移図)

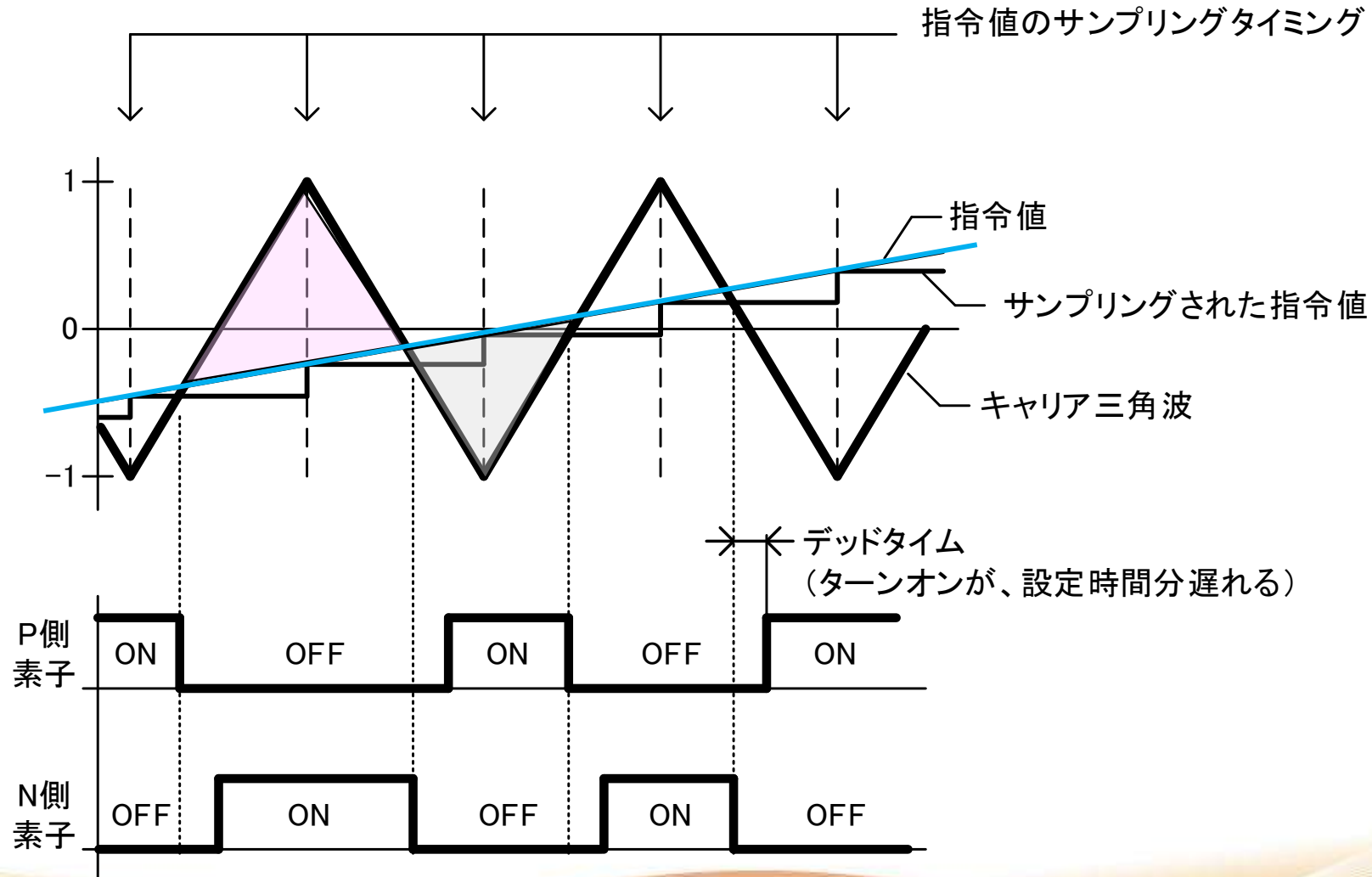


出力信号(内部設定、制御指令、外部出力)

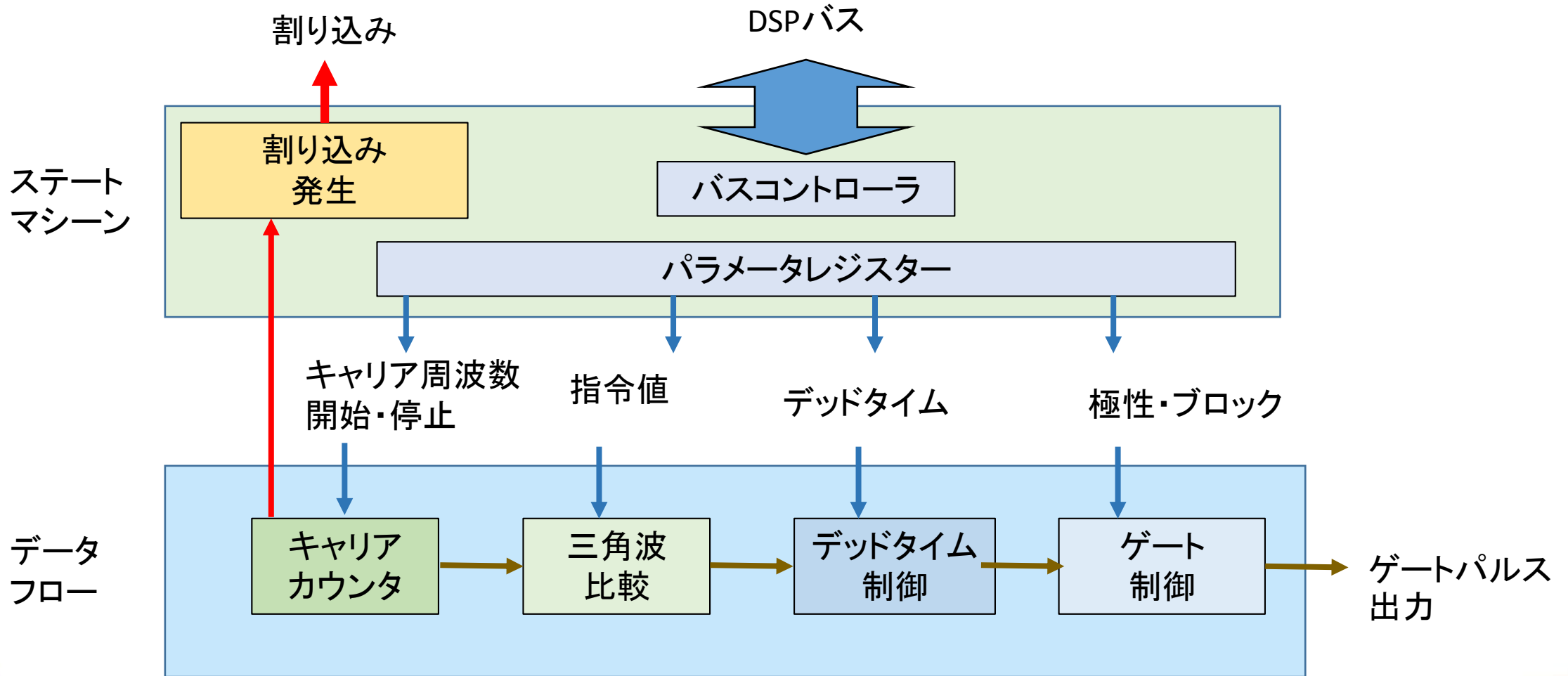
データフロー部



PWMゲートパルス生成部の例



PWMゲートパルス生成部ブロック図



演習2 PWM生成回路の構成例

シミュレーションとデバッグ作業

演習資料を参照下さい

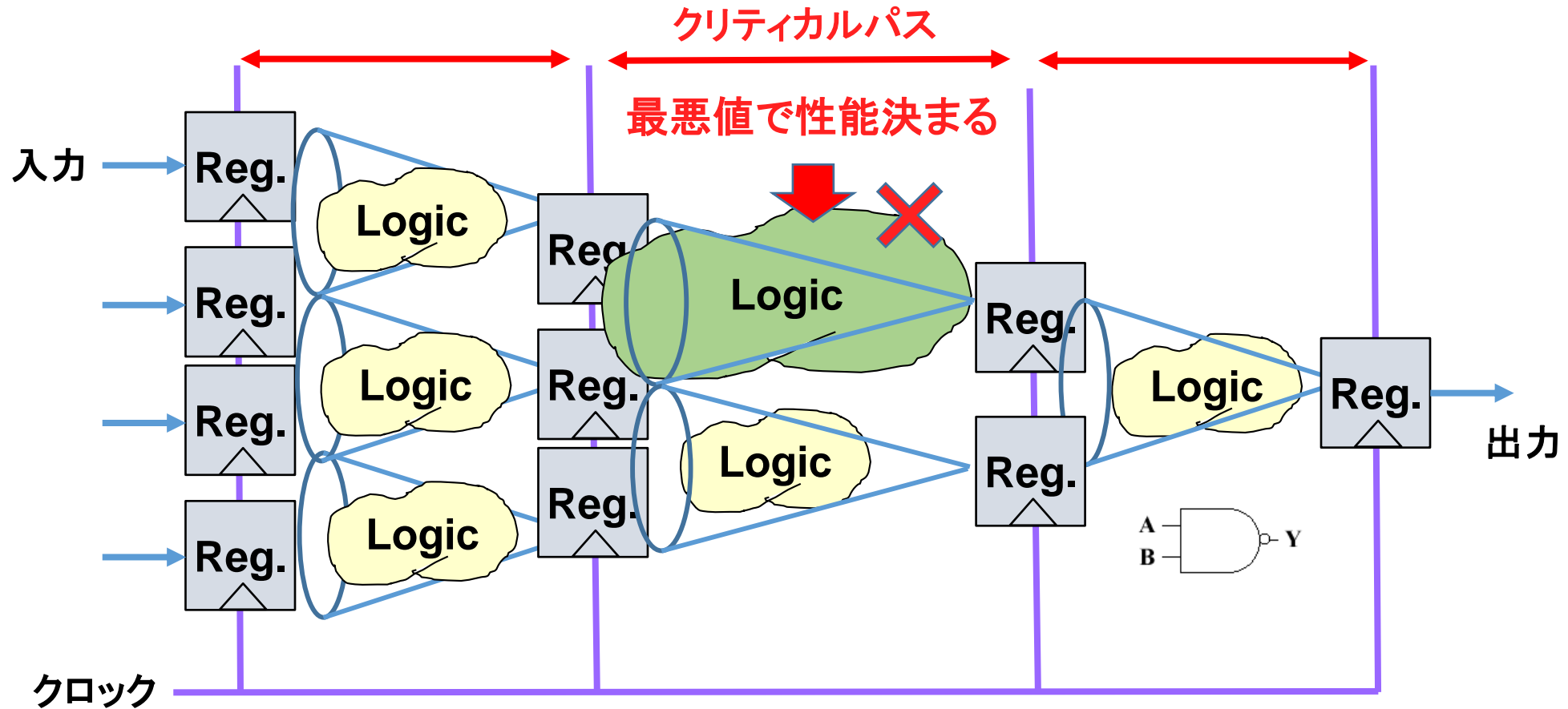
演習3 階層化

モジュール呼び出し

演習資料を参照下さい

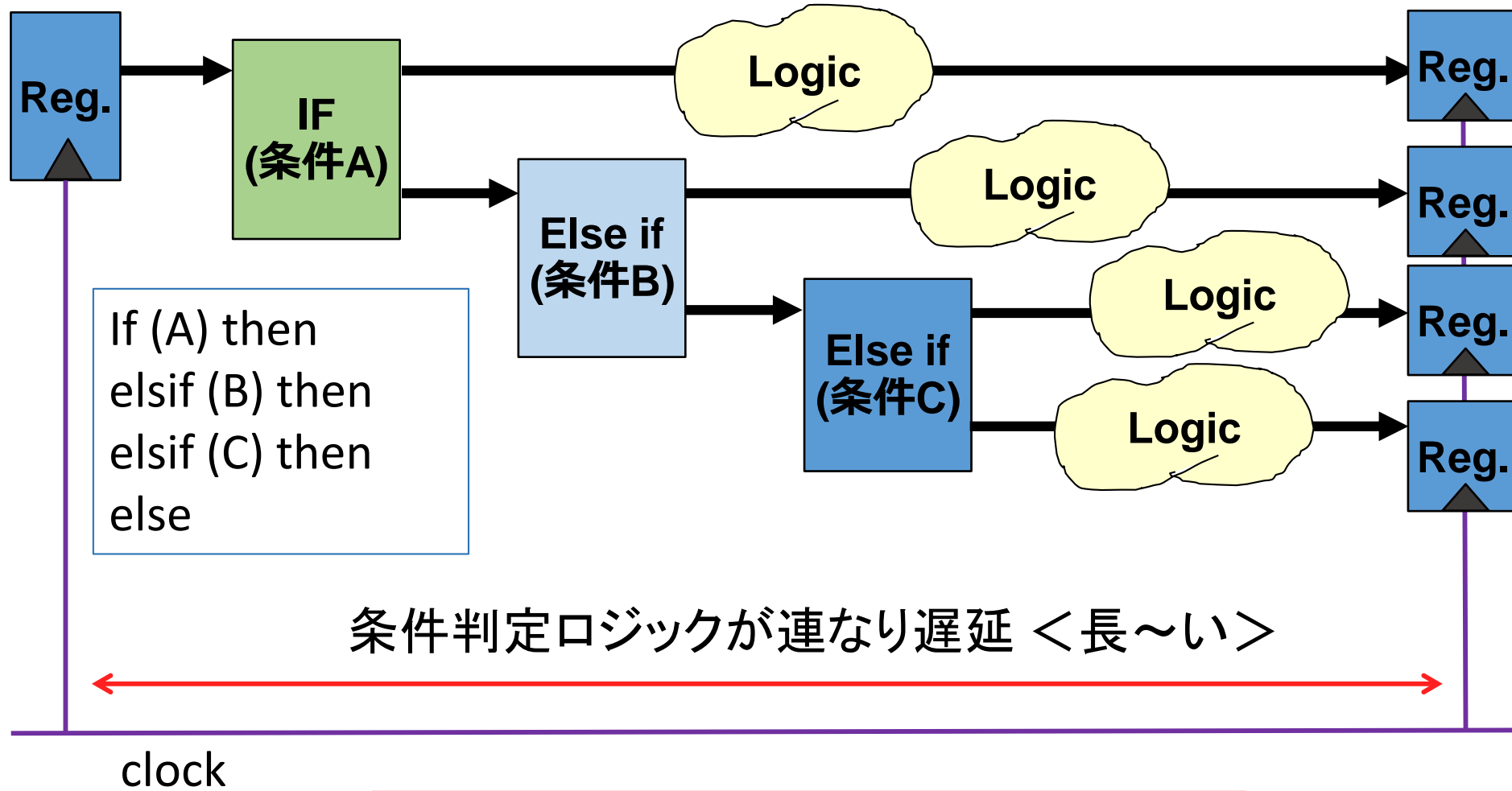
センスの良いコーディング

合成結果の回路を常に意識



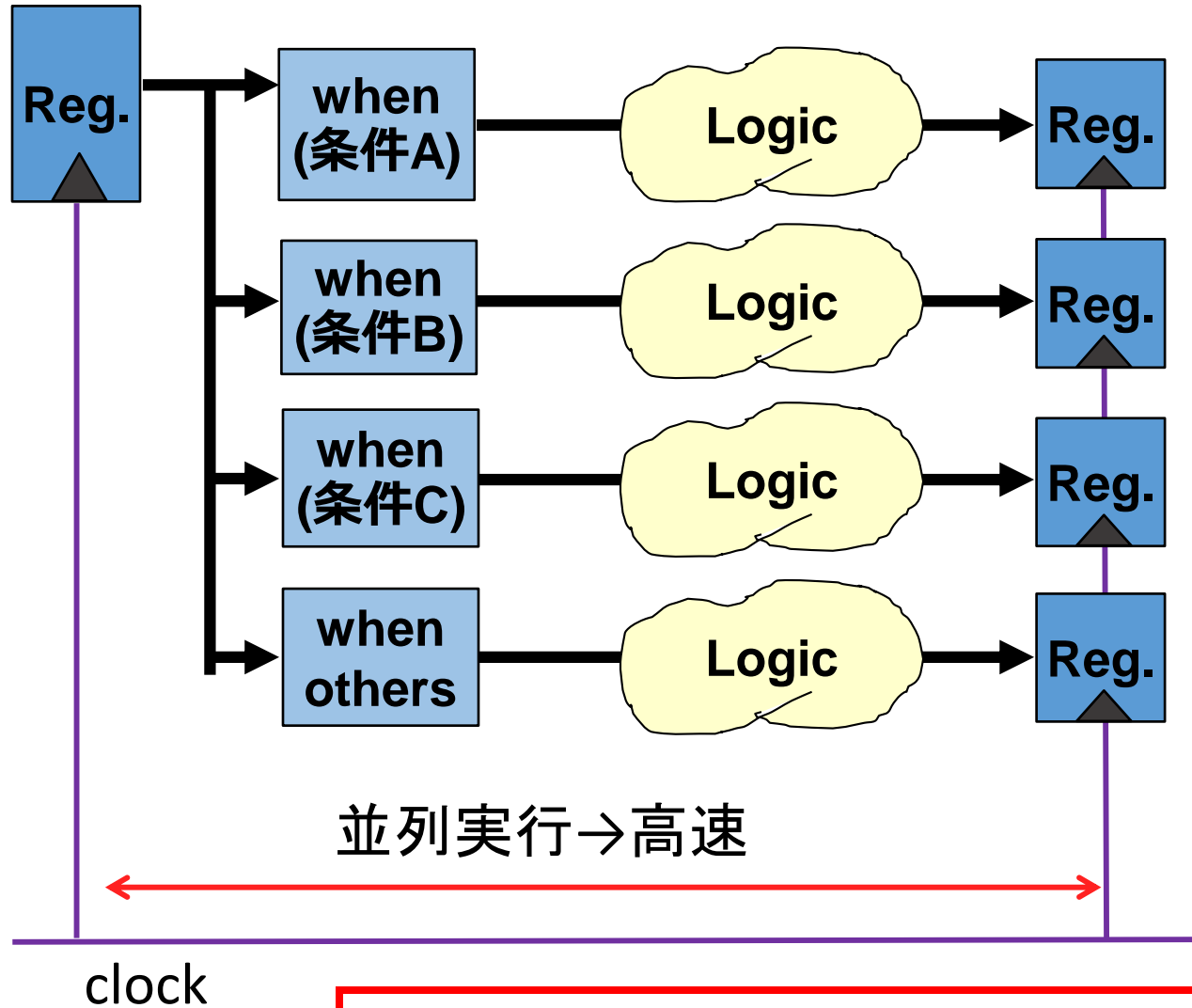
合成結果の回路をイメージして「バランス設計」
「最悪値」(クリティカルパス)が全てを台無し

プライオリティエンコーダは要注意 △



IFの連続に注意

Case文を活用しよう ○



条件成立が排他

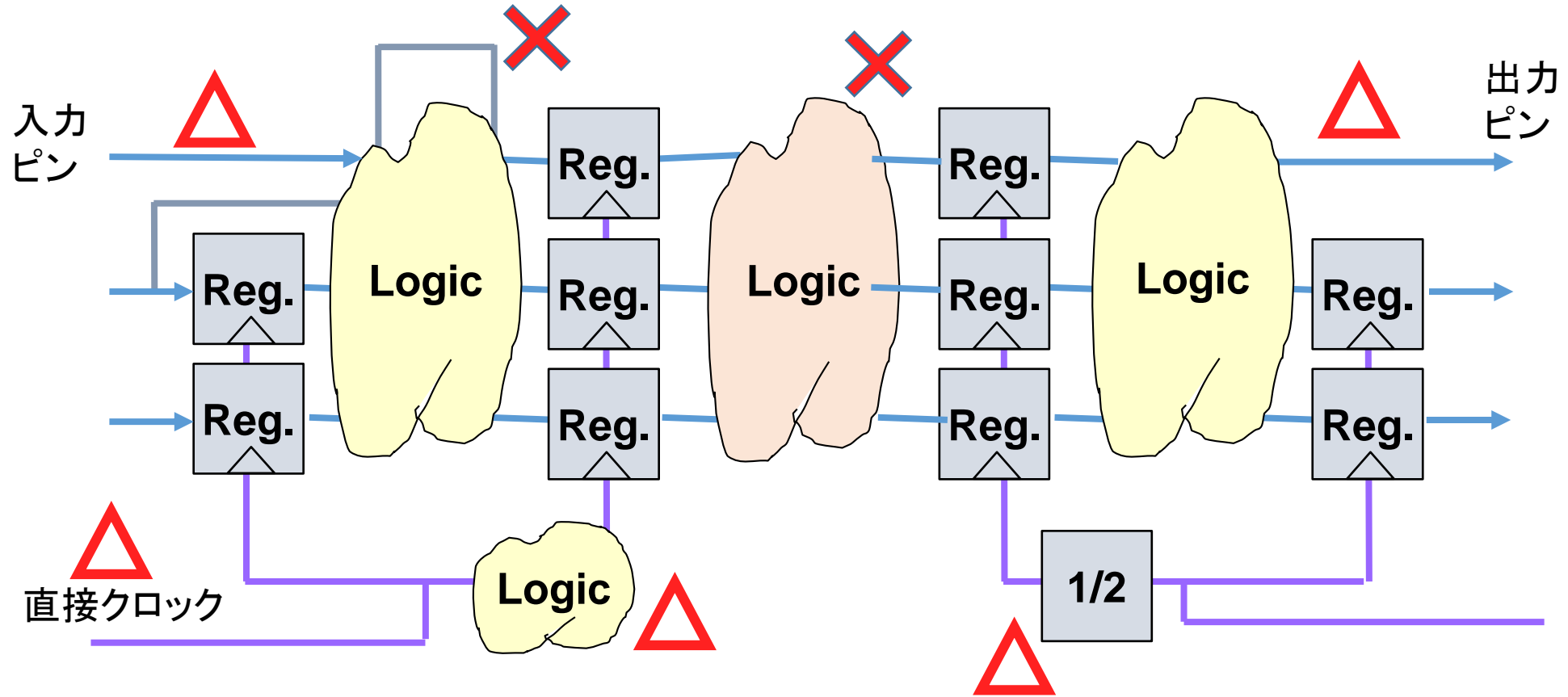
Case ABC is
when A
when B
when C
when others

ステートマシンの基本形

こんなことはありませんか？

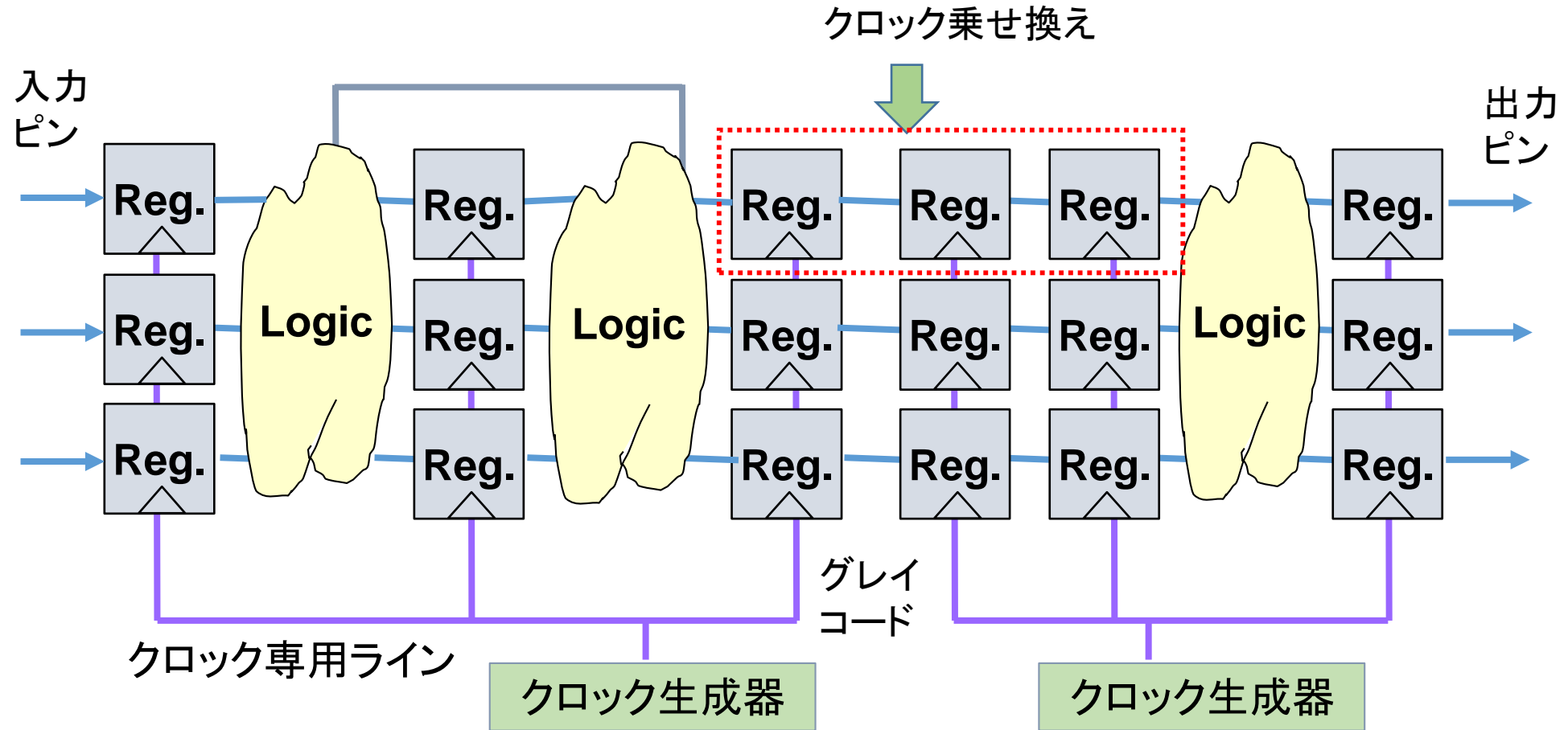
- コンパイルの度に動作が変わる
- 昨日は動いたのに、今朝は動かない
- 設計変更したら、無関係のはずの機能がおかしくなった
- 個体差で動作にばらつきがある

原因はこれかも・・・ 非同期設計



メタステーブル、ラッチ、レーシング、ヒゲ

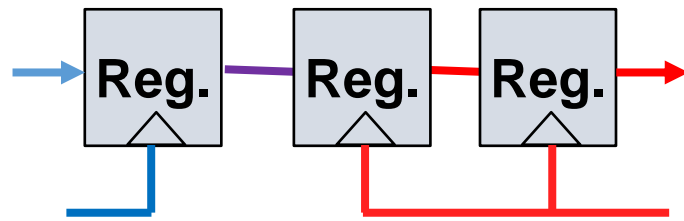
同期設計 ○



単一クロック推奨、非同期は丁寧に乗せ換え
外部入出力はレジスタ受け・出し

非同期対応

- 2段のFFで受ける(「2FFシンクロナイザ」、「メタスケルチ」)



- 非同期FIFO

- VivadoのIPを活用
- オーバフロー、アンダーフローに注意
- バスの場合はグレイコードを使用
 - バイナリコード: 0001 → 0010
 - 「メタる」と 0011になる可能性
 - グレイコード: 0001 → 0011
 - 「メタっても»: 0001, 0011のどちらか



Vivado IP Catalog
FIFO Generator

より高度なコーディング

さらにより高度な設計へ

- IP活用、構造化設計
 - 再利用性、IPツール活用、パワエレ専用IP(Myway)
- 多様・高速なゲートパターン生成
 - 大規模高速カウンタ(分散・並列化)・テーブル活用
- 最後は物理設計
 - PAR(Place & Route)で高速回路(500MHz超 PWM)
- SoC (System on Chip)
 - Xilinx Zync DSP+FPGA バスを自在に
- 高位合成ツール？

高位合成ツール(参考)

- Mathworks社
 - HDL Coder: SimuLinkアルゴリズム設計→汎用HDL
- Xilinx社
 - System Generator: 信号処理マクロ+ブロックセット
 - Vivado HLS: C言語によるIP作成

ツールの限界 ≠ 自分の限界

開発の効率化 ≠ 習得の効率化

効率化の対象、開発フェーズ、費用対効果で活用

演習4 モジュールのブロック化

IP作成による効率的な設計

演習資料を参照下さい

演習5 ADインターフェースの設計

順序回路によるシリアルデバイスとの通信

演習資料を参照下さい

4. FPGAプラットフォーム紹介

4. FPGAプラットフォーム紹介

4-1. PE-Expert4ご紹介

4-2. 製品開発事例

4-3. アプリケーション事例(参考)

PE-Expert4ご紹介

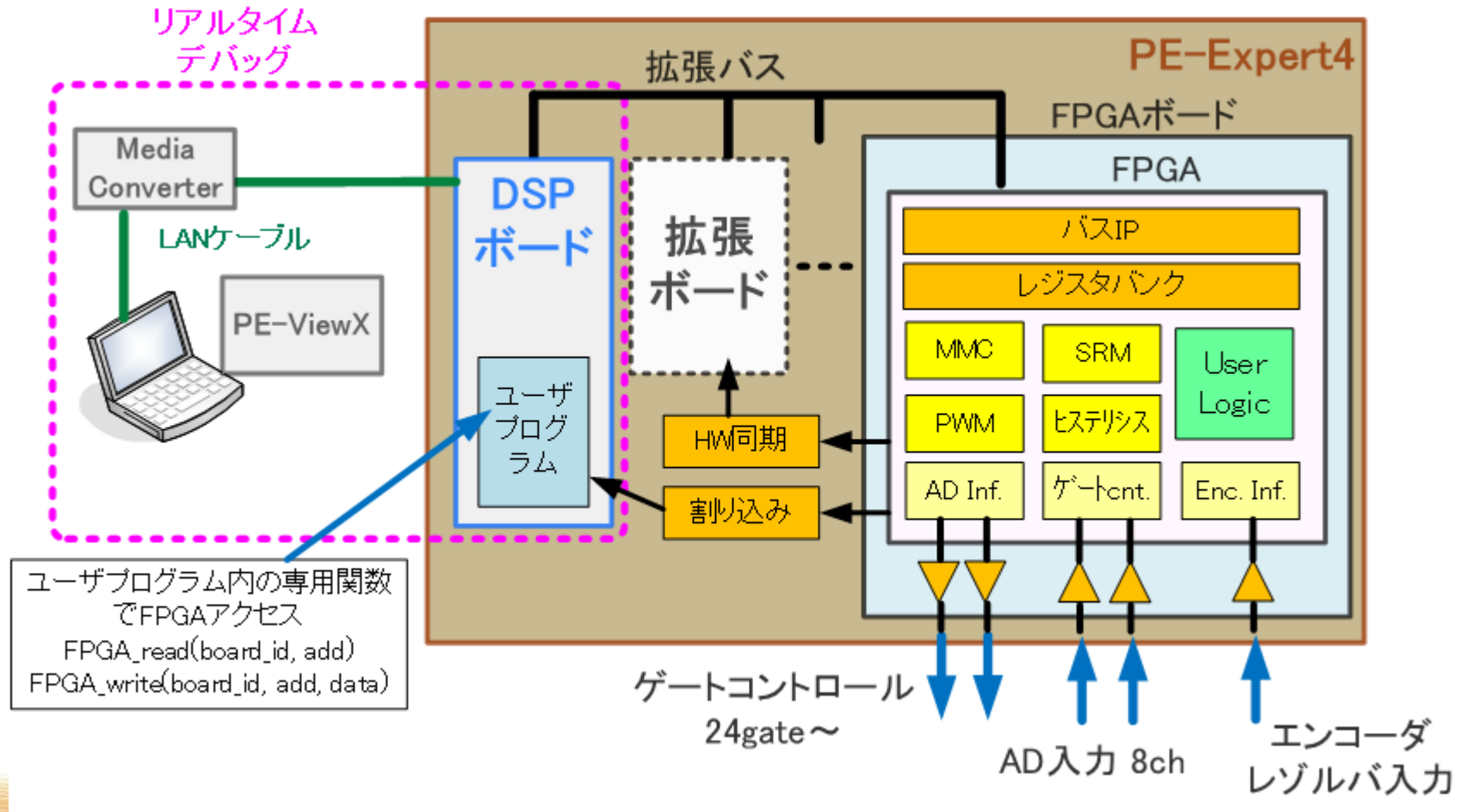
パワエレFPGAプラットフォーム

- PE-Expert4によるFPGA制御システムのプロトタイピング (RCP)
- TyphoonHILによるパワーステージのリアルタイムモデリング (HIL)

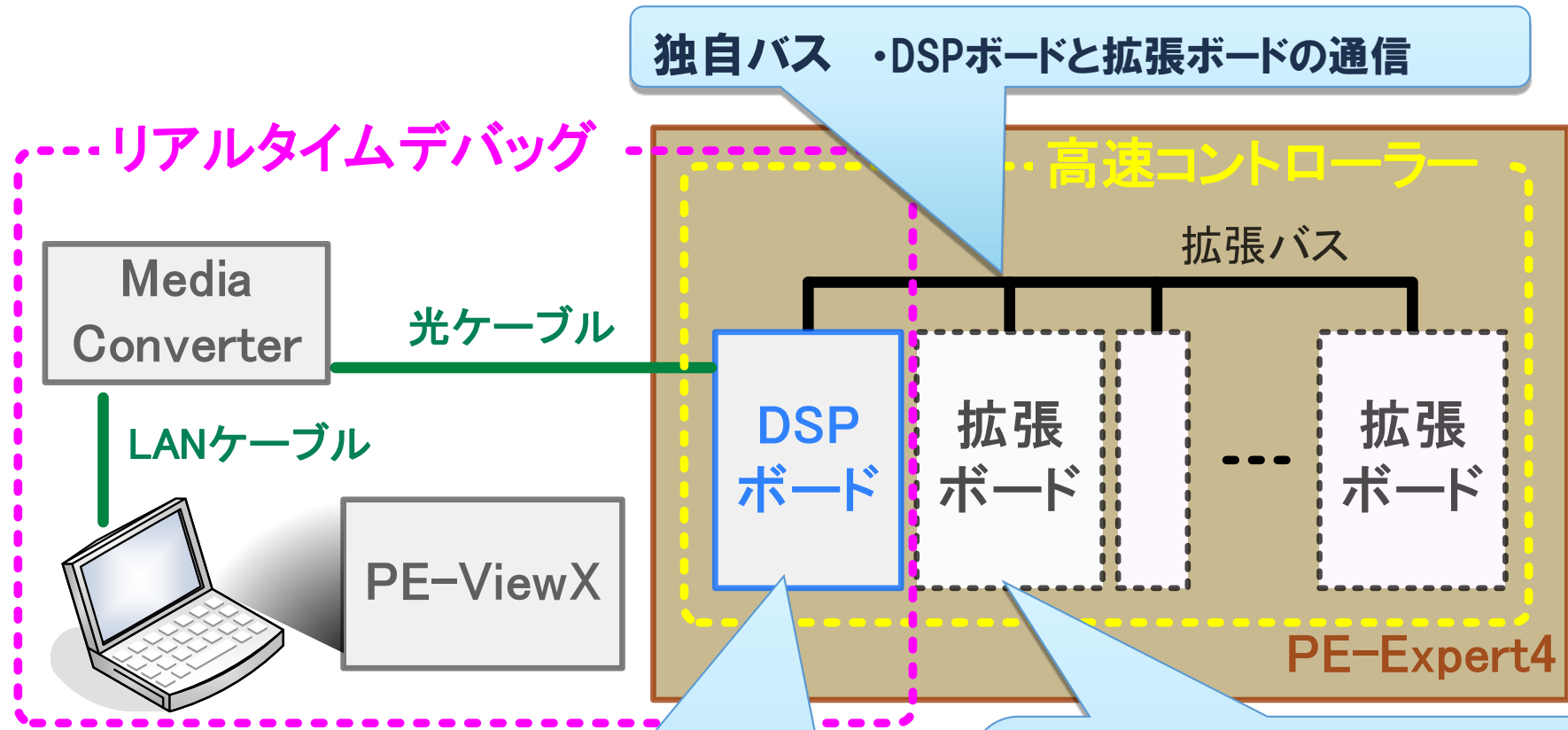


PE-Expert4 FPGAボード

パワエレ専用FPGAサンプル回路(IP)で効率開発



システム構成



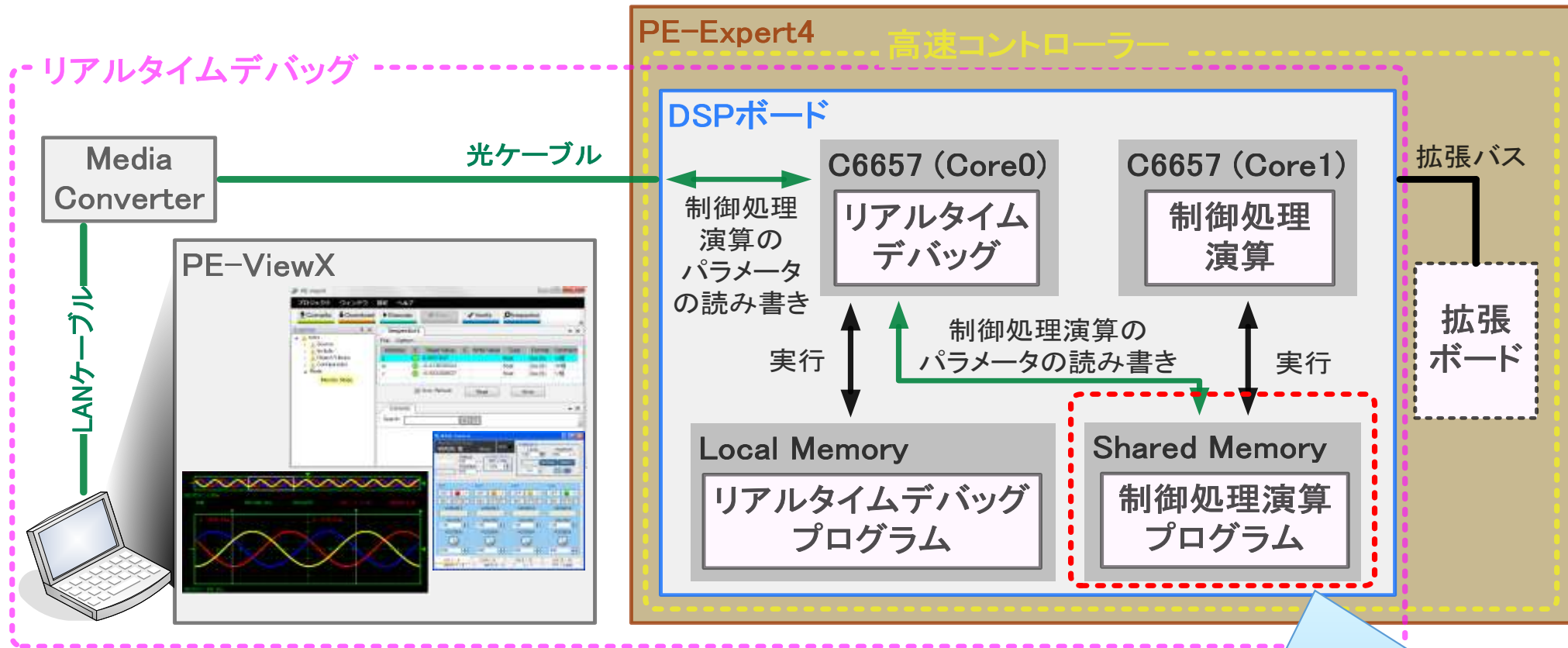
DSPボード

- ・制御処理演算 (1.25GHz)
- ・リアルタイムデバッグ (Dual Core)
(TI TMS320C6657)

- ・PEVボード PWM制御
- ・ADCボード: AD変換
- ・DACボード: DA変換
- ・PIOボード: デジタルインターフェース
- ・FPGAボード: Xilinx FPGA搭載

リアルタイムデバッグ

制御とリアルタイムデバッグを、デュアルコアで独立に実現



- 両方のコアから同時アクセスできる
- 両方のコアは独立実行できる

製品開発事例

製品事例

大容量バッテリー充放電システム

MWCDS-060-040A1/MWCDS-075-04

選べる2つのモデル



電流リップル重畳機能により電池評価機能が向上

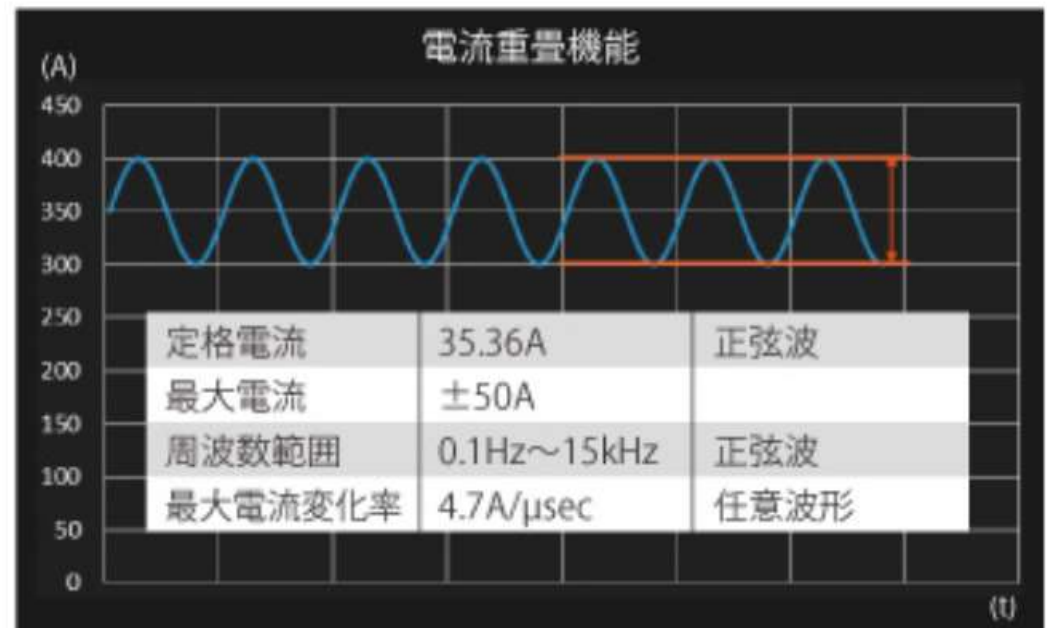
MWCDS-060-040A1

600V版

(600V/±400A/±100kW)

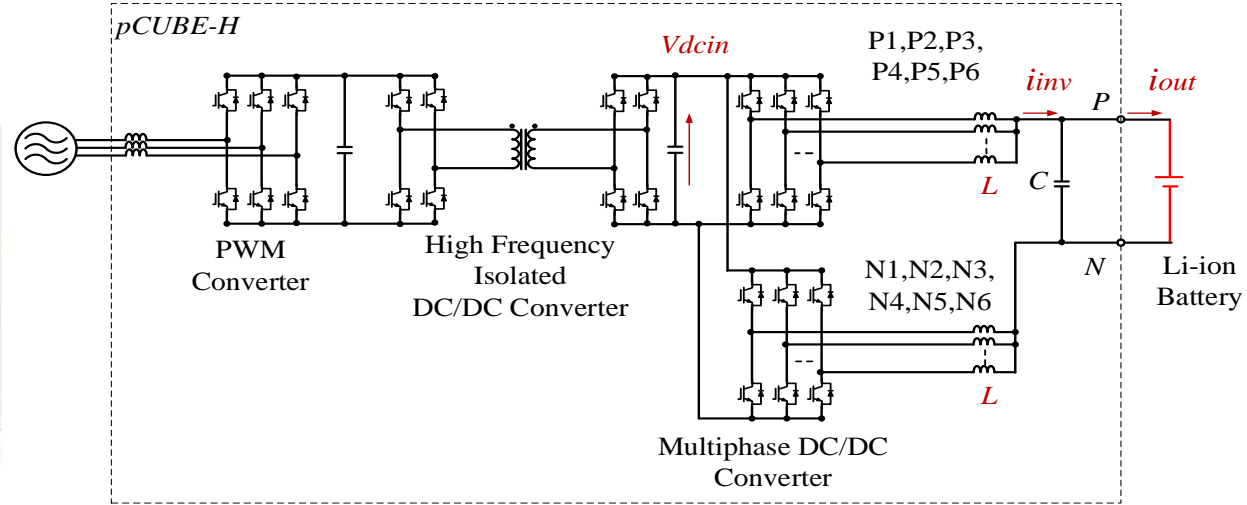
- 関連論文 (ECCE2018)
- Design Method of DC Power Supply for Superposing 20kHz/100A Peak to Peak Sinusoidal Current with Several Hundred DC Current to Analyze Battery AC Impedance
- [10.1109/ECCE.2018.8558032](https://doi.org/10.1109/ECCE.2018.8558032)

電流リップル重畳機能 (600Vモデルのみ)



Coordination control: CPU and FPGA

Controller(Expert4)



CPU Board(TMS320C6657)

- System Mode Controller
- Output Voltage Controller/Compensator
- Limitation and protection Controller
- Other Controllers

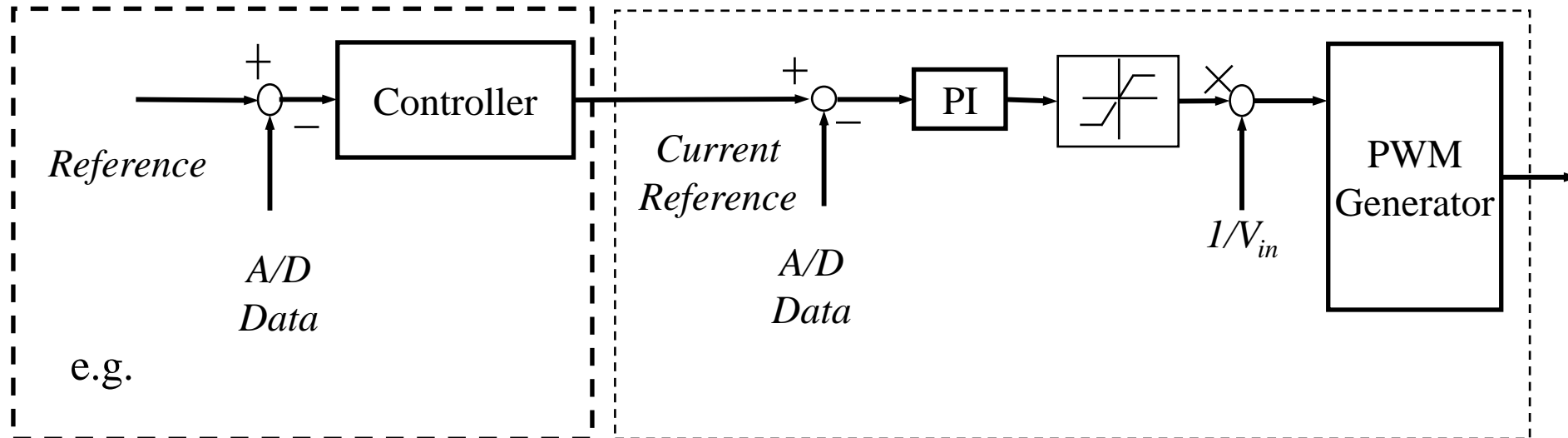


FPGA Board(KINTEX-7)

- 20bit A/D Converter
- Output Current Controller

High speed current control method

Brief control blocks between CPU and FPGA



CPU

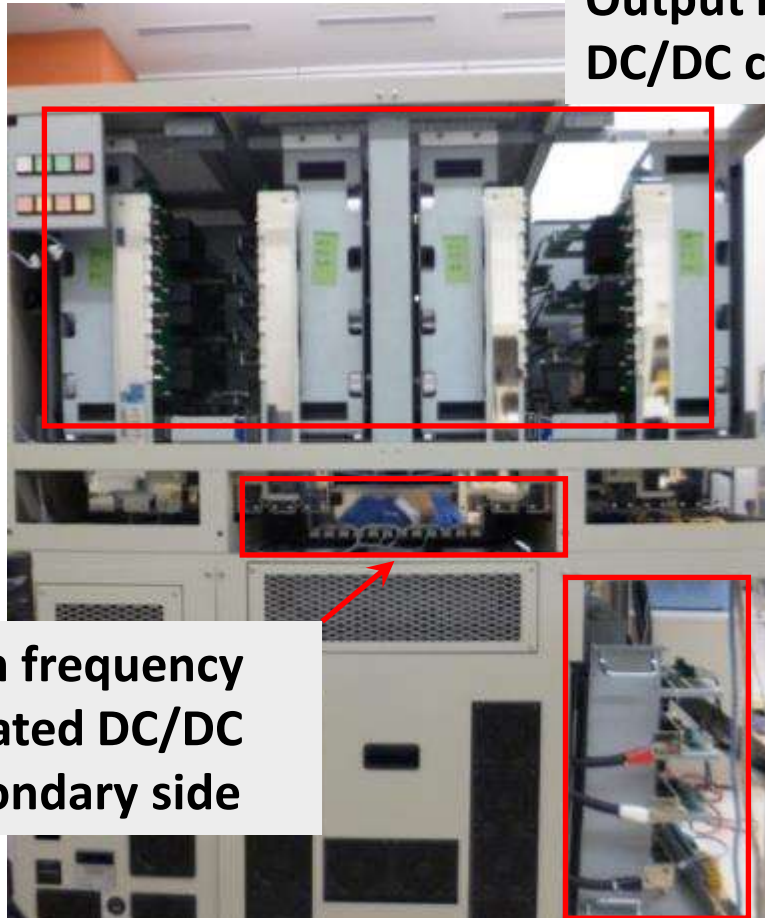
Control period: 50kHz

FPGA

Control period: 600kHz

Experimental results

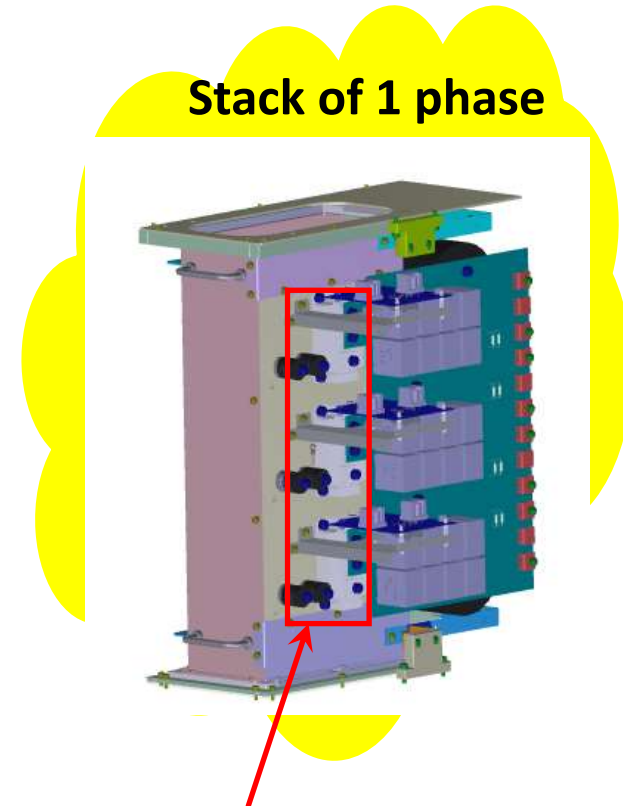
Bidirectional DC power supply(pCUBE-H)



Output multiphase DC/DC converter

High frequency isolated DC/DC secondary side

Three-phase PWM converter



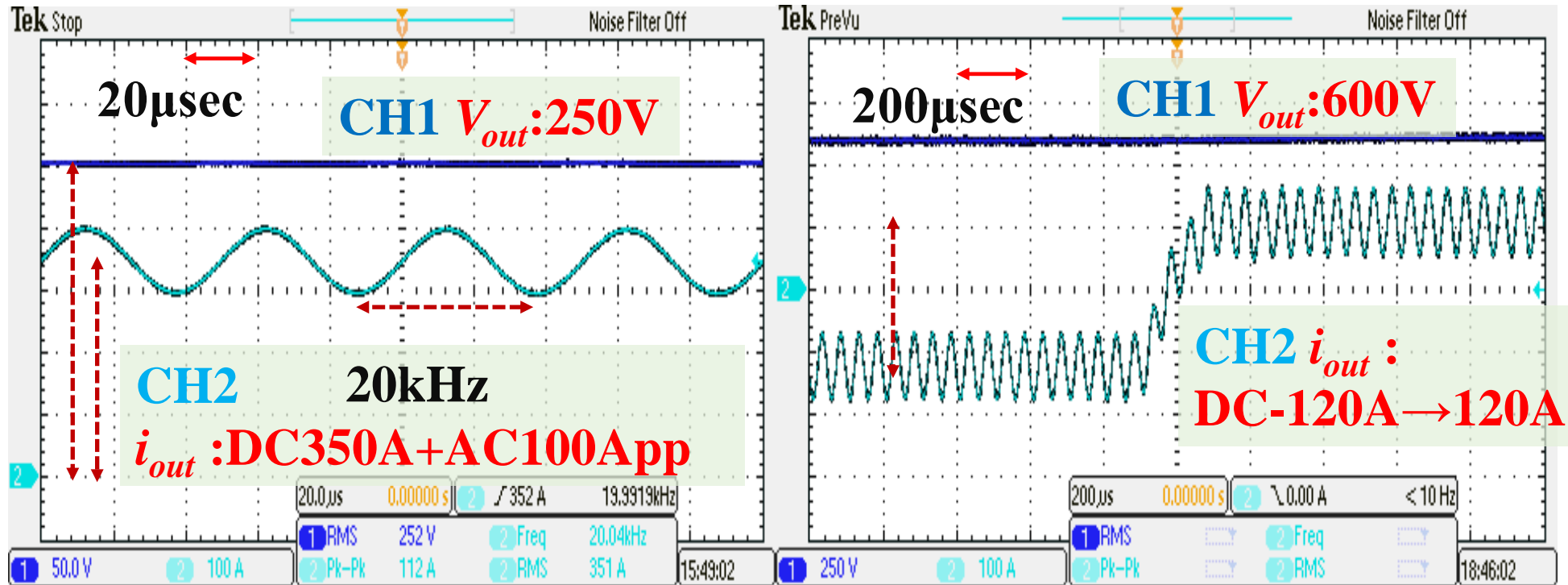
Stack of 1 phase

SiC MOSFET

Specification (**ALL SiC devices**): $V_{out} = 600V$ $i_{out} = 400A_{max}$ $P=100kW$

Experimental results

Electronic load using pCUBE-H



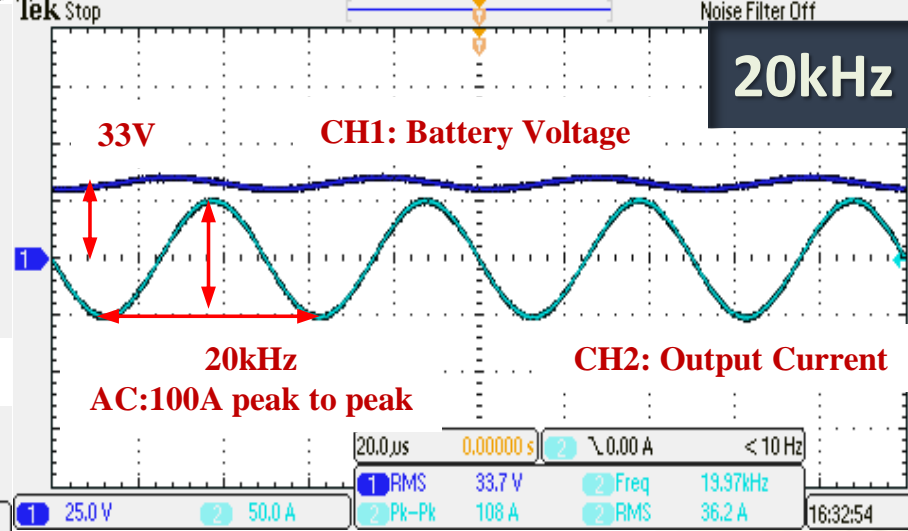
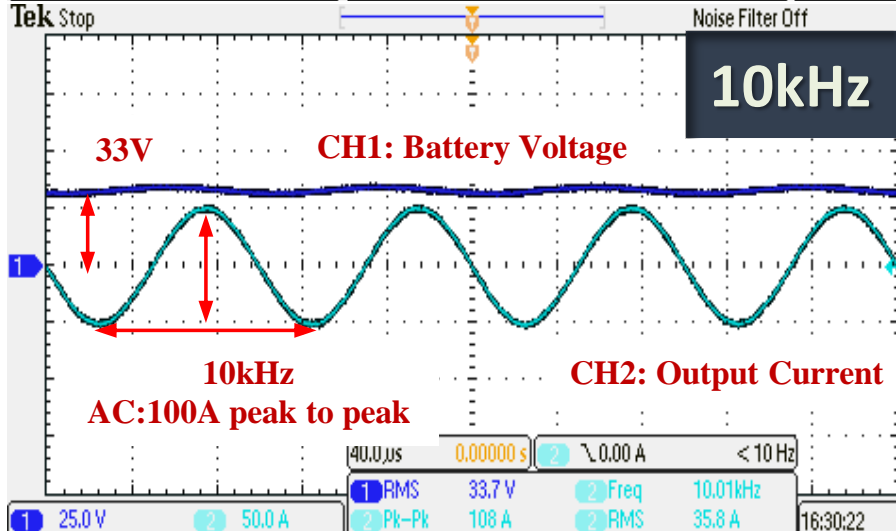
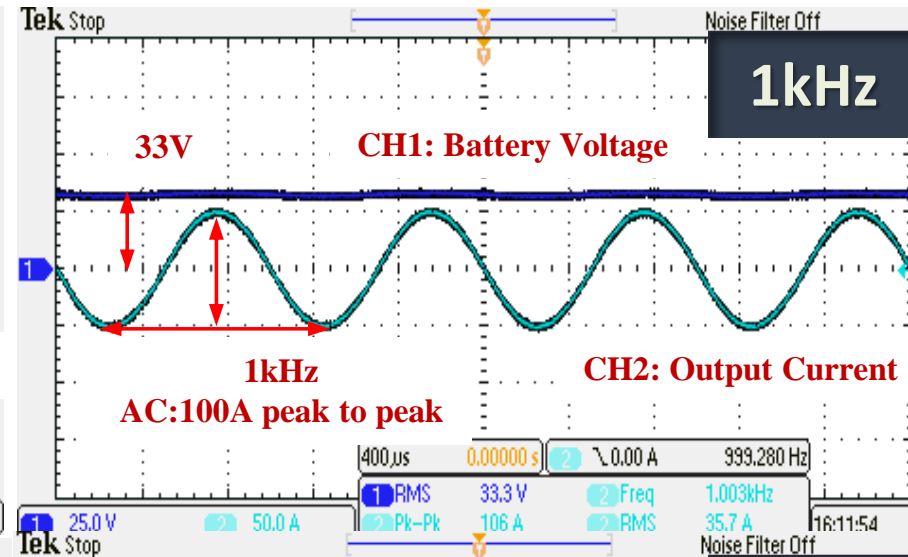
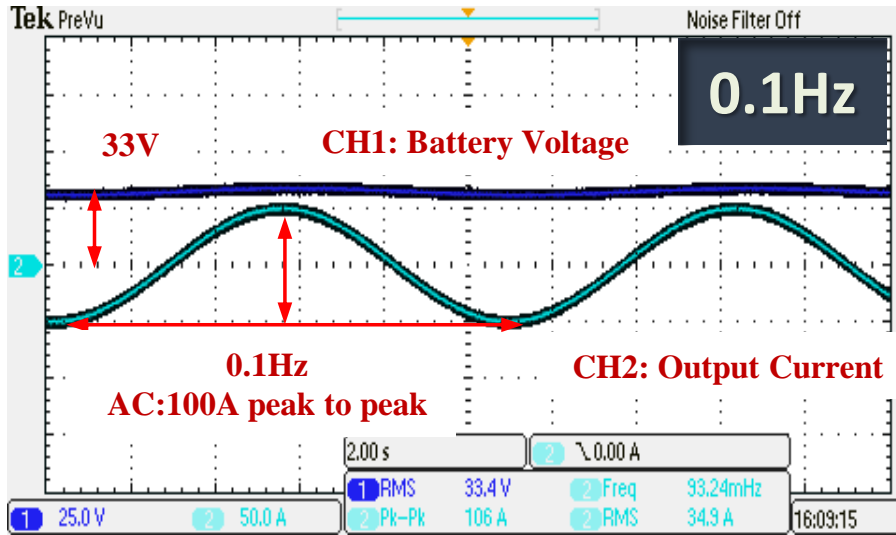
$V_{out} = 250V$, $i_{out} = \text{DC}350A$
AC current is 100A_{pp} at 20kHz

$V_{out} = 600V$, $i_{out} = \text{DC} -120A \rightarrow 120A$
AC current is 100A_{pp} at 20kHz

The AC current can be **regulated instantaneously**

Experimental results

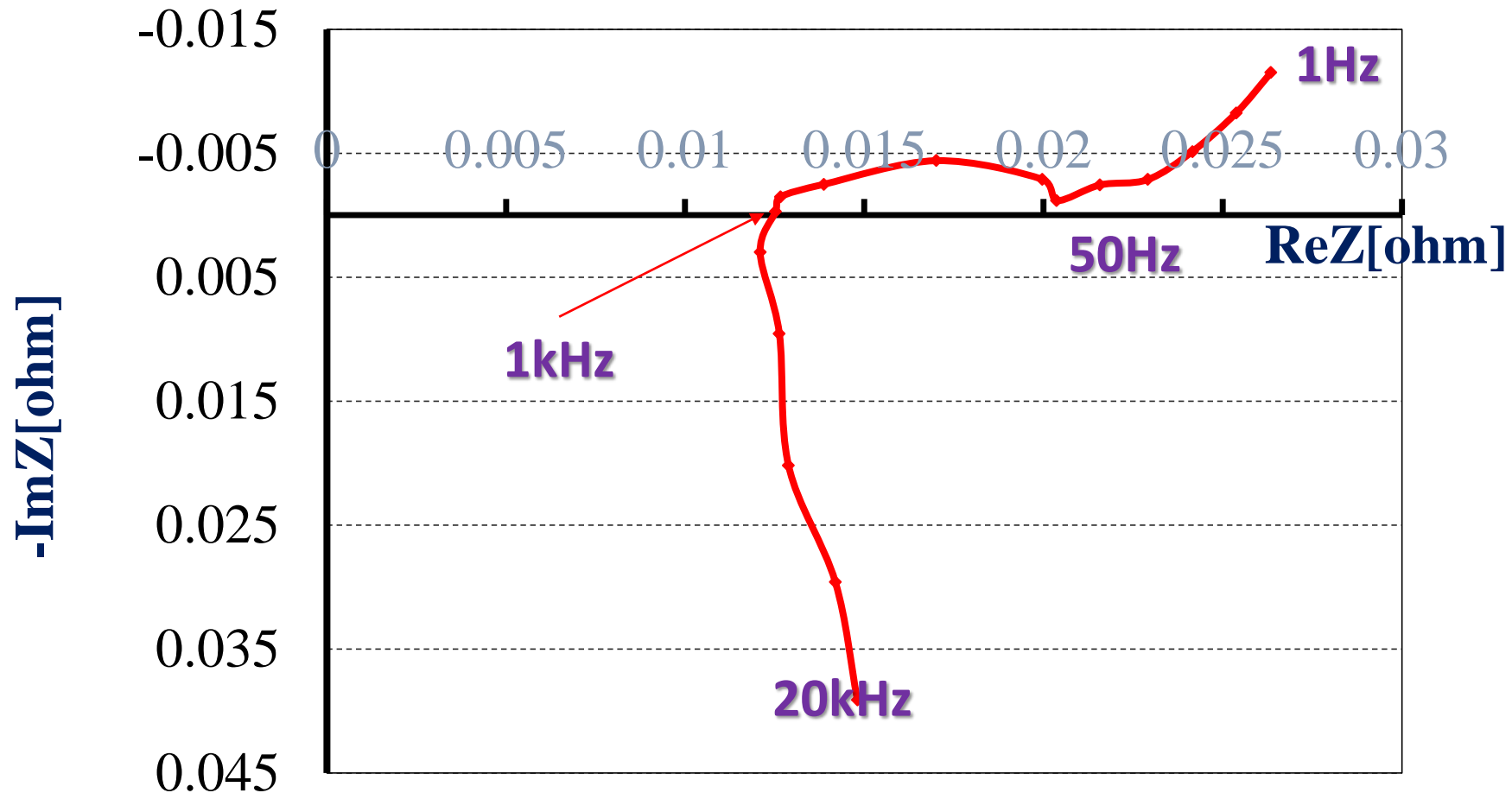
Battery load



$V_{battery} = 33V$, AC current is 100App from 0.1Hz to 20kHz

Experimental results

Battery impedance

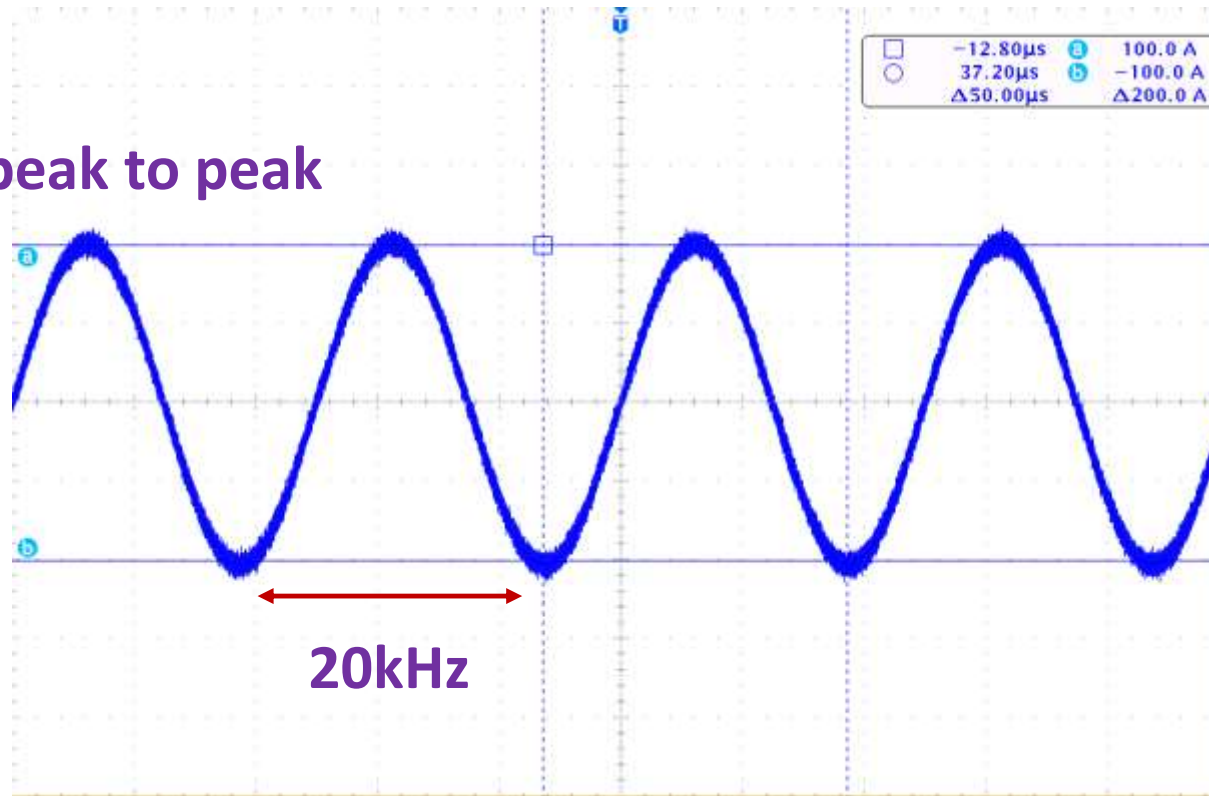


**the proposed method can satisfy the demands of analyzing
battery AC impedance**

Experimental results

Parallel connection

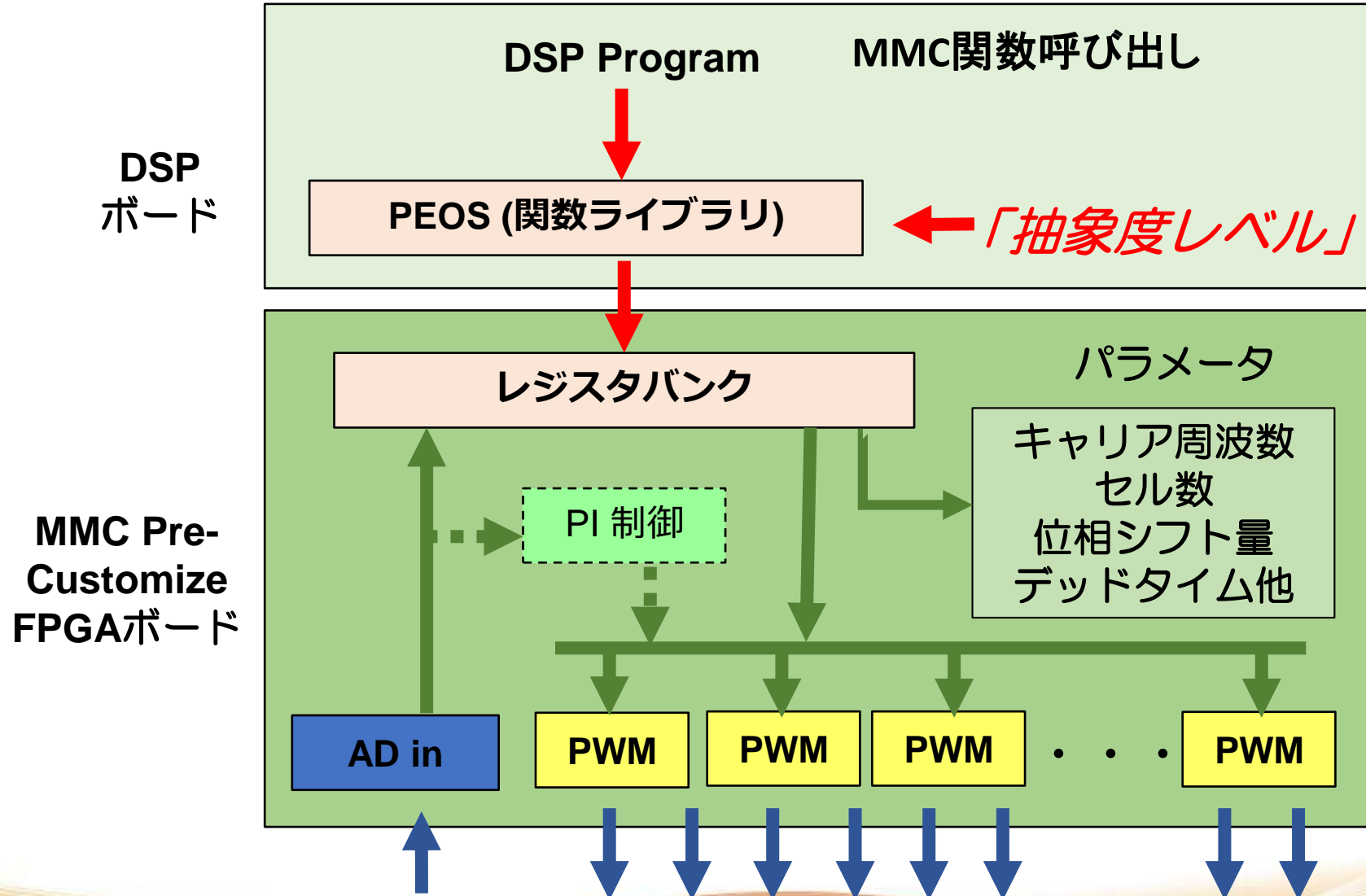
AC200A peak to peak



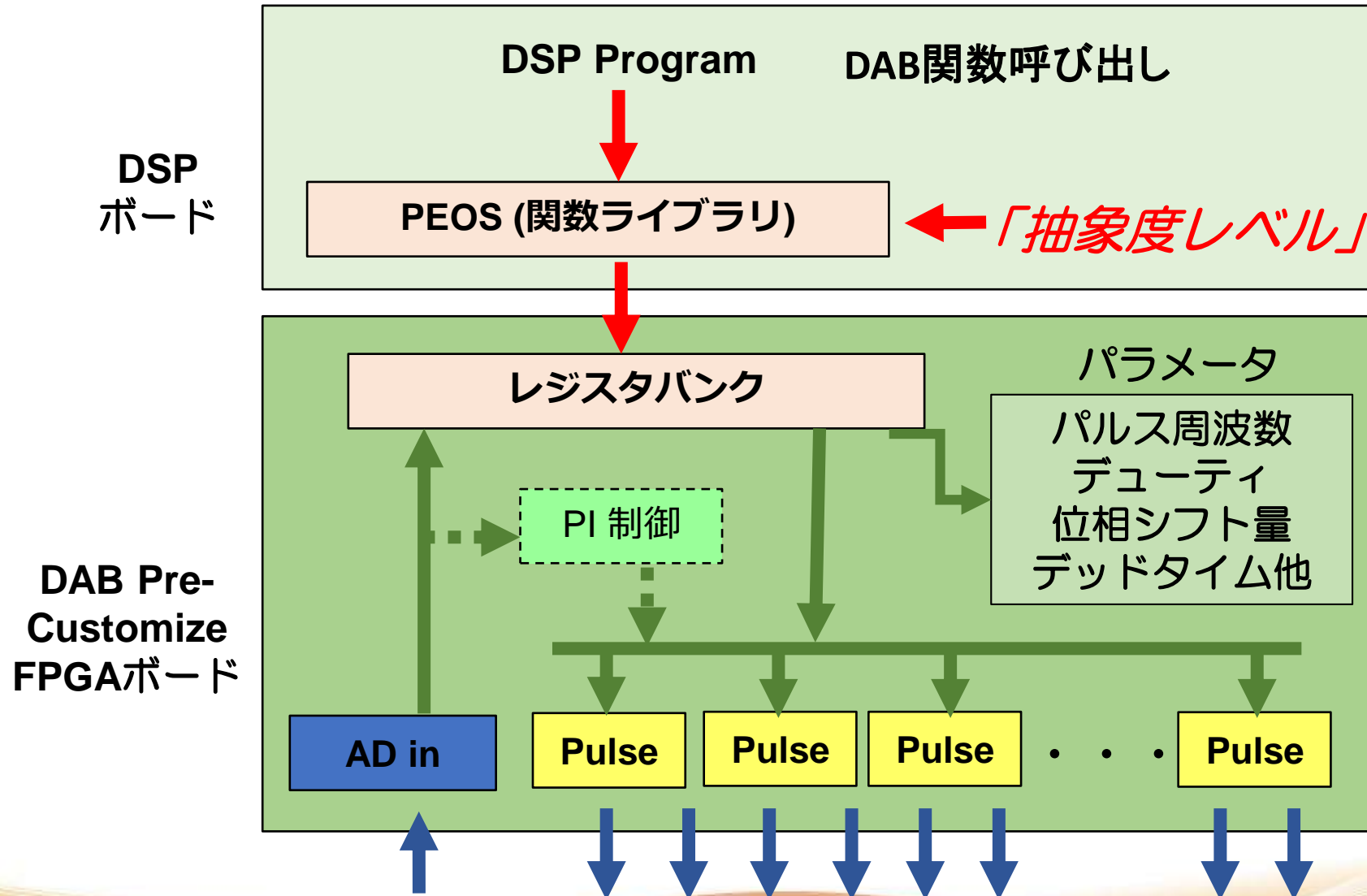
- **Low current distortion**
- **Phase synchronization**

アプリケーション事例（参考）

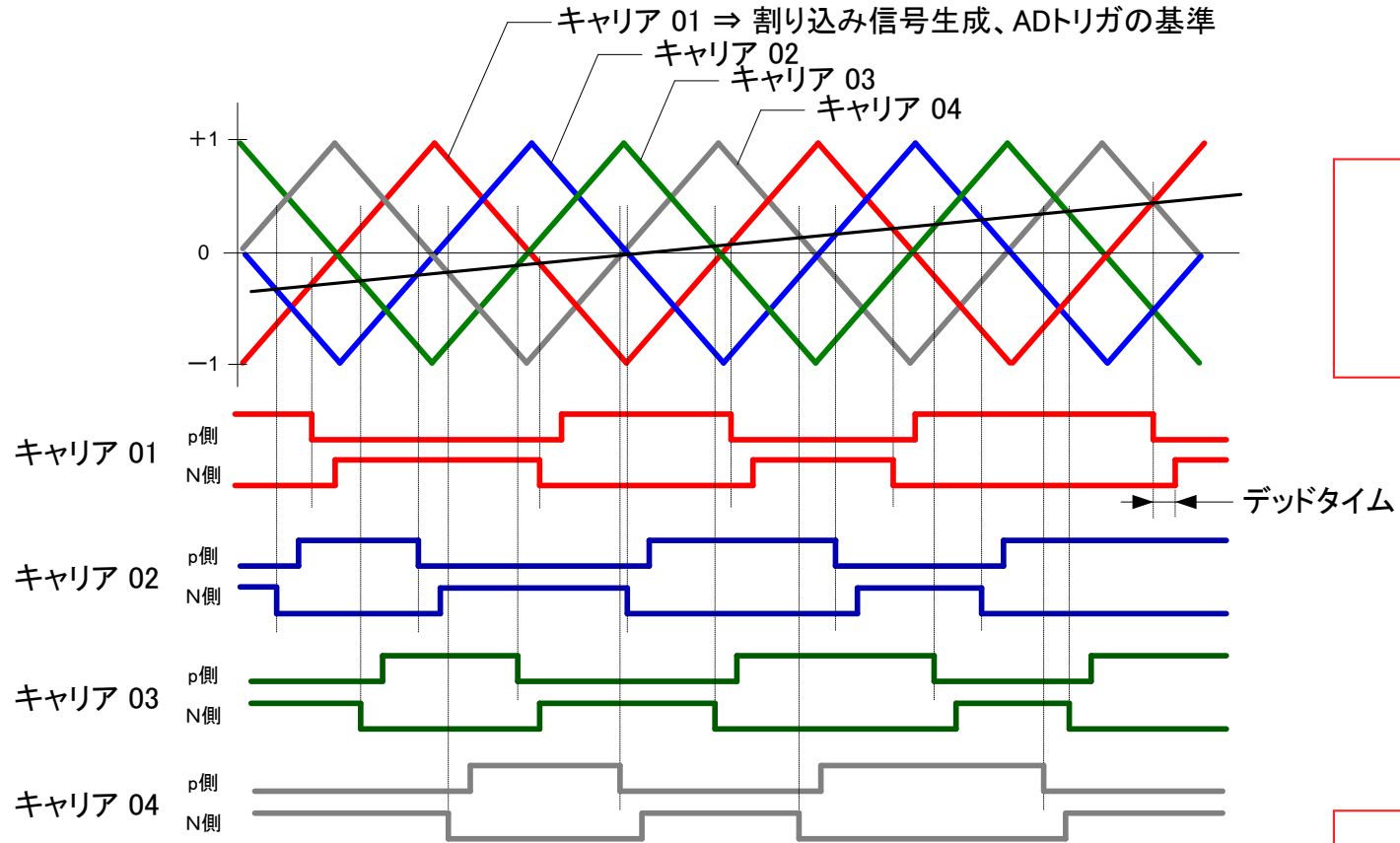
専用C関数からFPGAを制御



高精度位相制御 DAB



マルチフェーズ キャリアシフトPWM制御



空間的拡張性
の活用
(多ゲート制御)

HWロジックとSWの切
り分け(適材適所)

SWインターフェース

関数機能: マルチレベルPWM生成機能の初期化

```
INT32 MLPWM_inverter_init ( INT32 bdn, INT32 fc, INT32 level, INT32 dt);
```

関数機能: 三角波キャリア比較変調の指令値設定

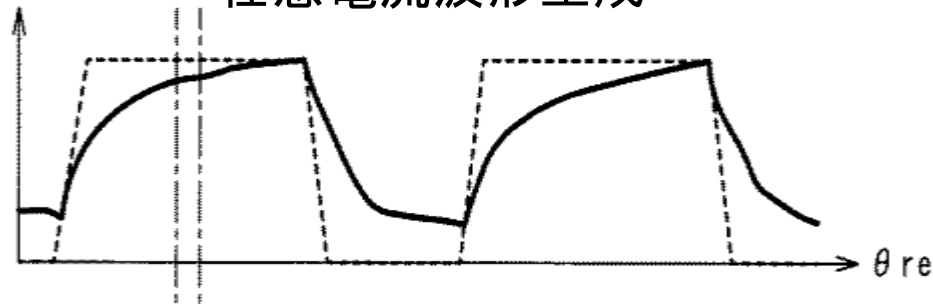
```
INT32 MLPWM_inverter_set_uvw ( INT32 bdn, FLOAT32 u, FLOAT32 v, FLOAT32 w);
```

FPGA活用例 任意波形

事例紹介

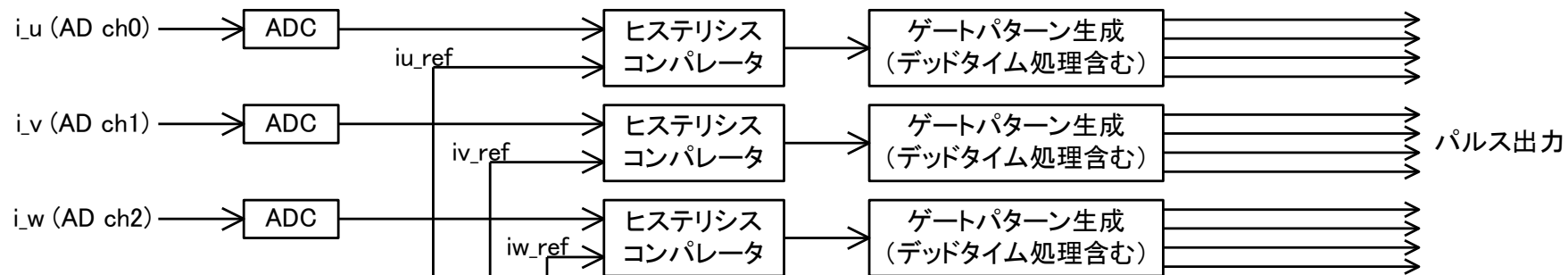


任意電流波形生成



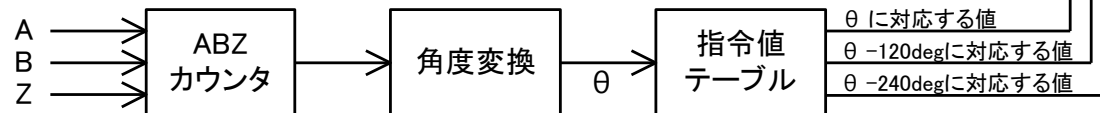
電流値

FPGAブロック図



Hブリッジゲート制御
(正・負・ゼロ)

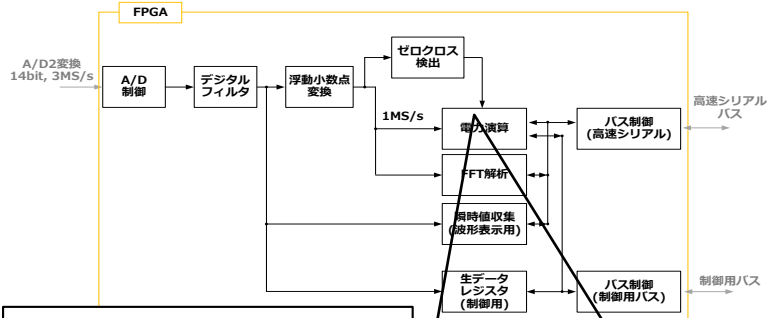
角度
情報



任意波形テーブル(各相個別)

パワーメータ機能の概要(計測項目)

事例紹介



電力LC能
(サンプリング: 1MS/s)

$$X_{rms}(i) = \sqrt{\frac{1}{M} \sum_{s=1}^M (X(i)s)^2}$$

$$X_{mn}(i) = \frac{\pi}{2\sqrt{2}} \frac{1}{M} \sum_{s=1}^M |X(i)s|$$

$$S(i) = U_{rms}(i) \times I_{rms}(i)$$

$$P(i) = \frac{1}{M} \sum_{s=1}^M (U(i)s \times I(i)s)$$

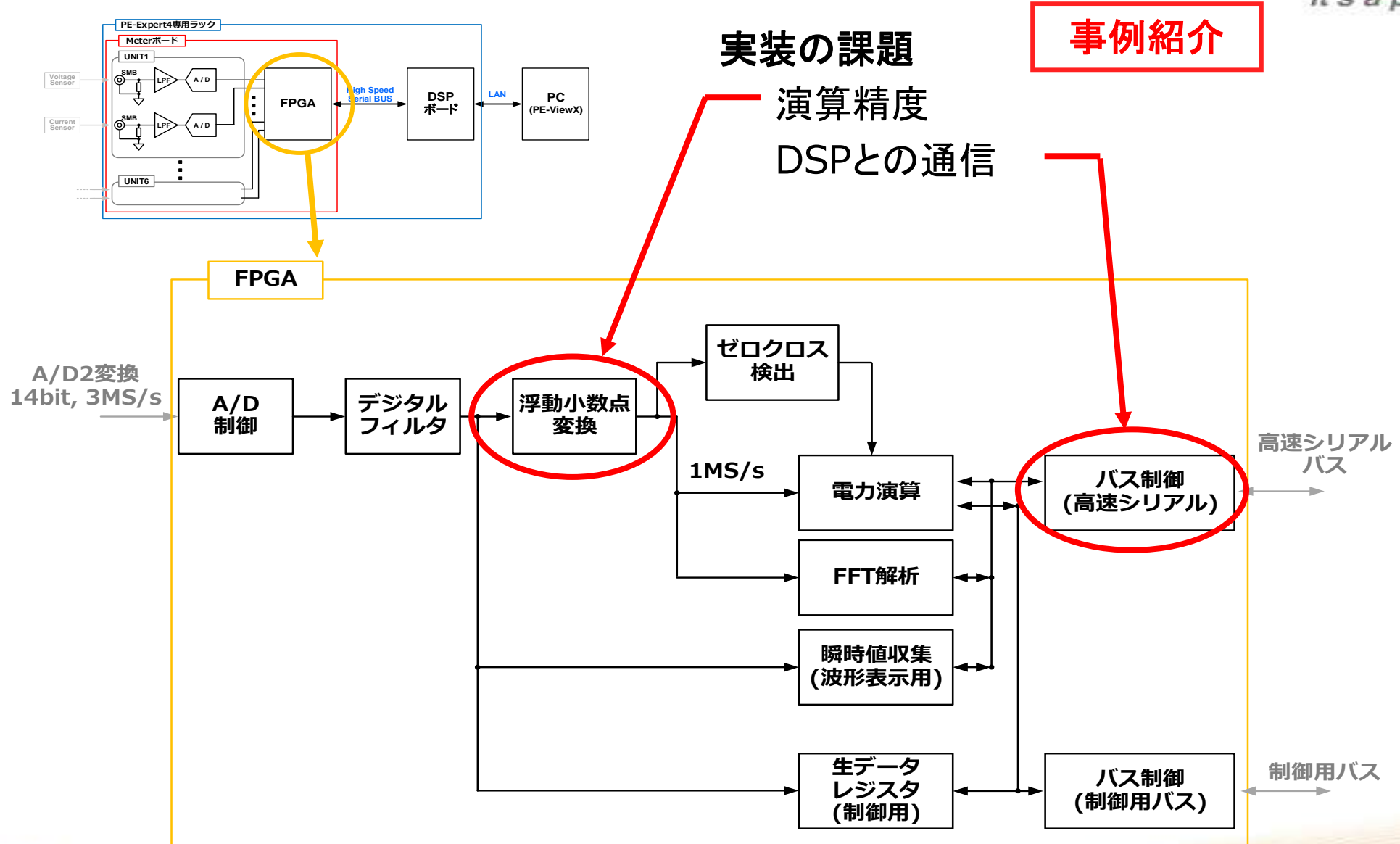
$$Q(i) = si(i) \sqrt{S(i)^2 - P(i)^2}$$

⋮

計測項目	記号	単位
電力系計測項目		
有効電力	P	W
無効電力	Q	var
皮相電力	S	VA
位相角	φ	deg.
力率	λ	
電圧実効値	Urms	V
電流実効値	Irms	A
電圧平均値整流実効値換算	Umn	V
電流平均値整流実効値換算	Imn	A
電圧直流成分	Udc	V
電圧交流成分	Uac	V
電流直流成分	Idc	A
電流交流成分	Iac	A
周波数	f	Hz
電圧ピーク+	Upk+	V
電圧ピーク-	Upk-	V
電流ピーク+	Ipk+	A
電流ピーク-	Ipk-	A
電圧瞬時値	U	V
電流瞬時値	I	A
効率	η1~η4	%
高調波解析計測項目		
高調波電圧値	—	V
電圧高調波含有率	—	%(for Fud)
高調波電流値	—	A
電流高調波含有率	—	%(for Fud)
高調波有効電力	—	W
総合高調波電圧歪率	THD	%
総合高調波電流歪率	THD	%
機械系計測項目		
回転		
トルク		
温度計測項目		
温度	temp	℃

既存の計測機器と同様の計測項目を実装

パワーメータ機能の概要 (システム構成)



参考情報

The screenshot shows the PWEL website interface. At the top left is the PWEL logo (power electronics association) and the text '一般社団法人 日本パワーエレクトロニクス協会'. To the right are buttons for 'お問い合わせフォーム', '会員登録', and 'ログイン'. A navigation bar contains links for '協会について', '入会案内', 'お知らせ', 'コミュニティ', 'eラーニング', '技術情報', 'セミナー情報', '推薦図書', and 'マイページ'. Below this is a search bar with the text 'パワエレ専門のコミュニティサイト' and a search button. The main content area is divided into two columns. The left column is titled '人気の記事(閲覧数順)' and lists five articles. The right column is titled '質問カテゴリ' and lists various categories like '回路シミュレータPSIM', '統合開発環境', '組込基板開発', etc. Below that is a '投稿カテゴリ' section with links for 'パワエレ教材制作コンテスト', '自作回路・試作品紹介', and 'お知らせ'. At the bottom right, there is a '新着 (質問・投稿一覧)' section.

「日本パワエレ協会」「PWEL」で検索(pwel.jp)→「コミュニティ」をクリック

DSP+FPGA実践セミナー開催予定！！

- 実際のシステム設計、制御C言語、VHDLコーディングを実践
- 実機での動作確認
- すぐに使えるソースコード(C、VHDL)をお持ち帰り
- 基礎から、教科書に載っていないテクニックまで
- 実開発経験豊富なエンジニアが直接指導！！