

次世代SoCプラットフォームの開発 : SystemMorphとRedefis

吉松, 則文
福岡県産業・科学技術振興財団福岡知的クラスター研究所

<https://hdl.handle.net/2324/9142>

出版情報 : SLRC プレゼンテーション, 2006-07-19. 九州大学システムLSI研究センター
バージョン :
権利関係 :

次世代SoCプラットフォームの開発 ～SystemMorphとRedefis～

2006年7月19日

福岡県産業・科学技術振興財団

福岡知的クラスター研究所

吉松 則文

次世代SoCプラットフォームの開発 ～背景～

SoC (System-On-a-Chip)の高性能化、高機能化とSoCのコストへの対応

■ 製造コスト

- 半導体プロセス技術の微細化による、マスク費用、プロセス費用の上昇
- 高集積、高機能、高性能なSoCのテスト費用の上昇
- 多品種少量生産による開発投資の非効率さに伴うコスト高

■ 設計コスト

- SoCの高集積化、複雑化による開発コストの上昇
- 市場化時期を逸しない開発、短TAT(Turn-Around-Time)化の要求による開発コストの上昇

■ 運用コスト

- 高信頼性化、耐故障化への対応
- 消費電力

高性能化、高機能化しながら、これらSoCに関わるコストに対応可能なSoCプラットフォームの実現

SoCプラットフォーム

■ SoC

システムを構成する各種機能のうち、汎用プロセッサ上では、**性能上の理由**で実現出来ない機能をカスタム・ロジックとしてチップ上に集積

■ プラットフォーム・ベース設計

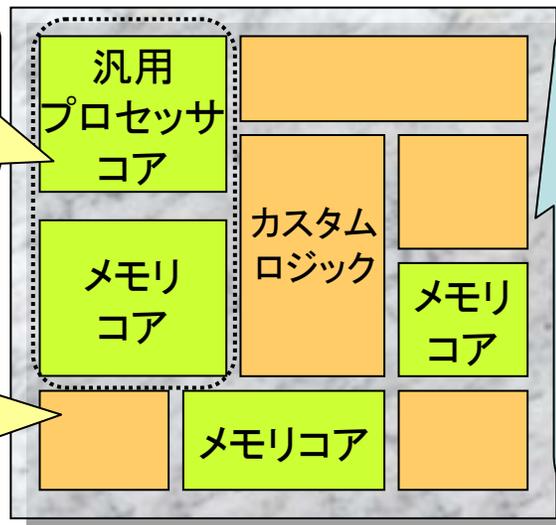
「カスタム・ロジック」の違いに拠らない「**共通の土台**」、プラットフォーム上で「カスタム・ロジック」を設計

汎用機能の実現法

- ・ 汎用プロセッサ
+ソフトウェア

カスタムロジックの
設計に集中

- ・ 短TAT化



SoCプラットフォーム

プロセッサ+ソフトウェア

- ・ CELL (ソニー/東芝/IBM)

コンフィギャラブル・プロセッサ+ソフトウェア

- ・ MeP (東芝)
- ・ UniPhier (松下電器産業)

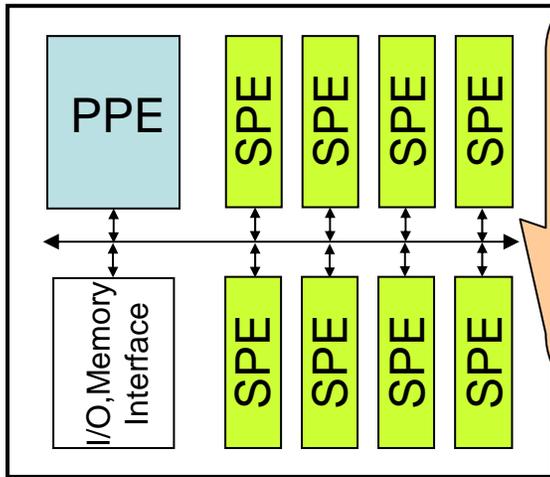
リコンフィギャラブル・プロセッサ+ソフトウェア

- ・ Redefis/SystemMorph (CLUSS)

プラットフォーム・ベースによるSoC

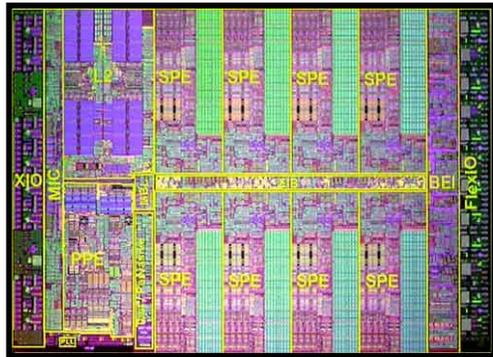
～ CELL ～

CELL ブロック図



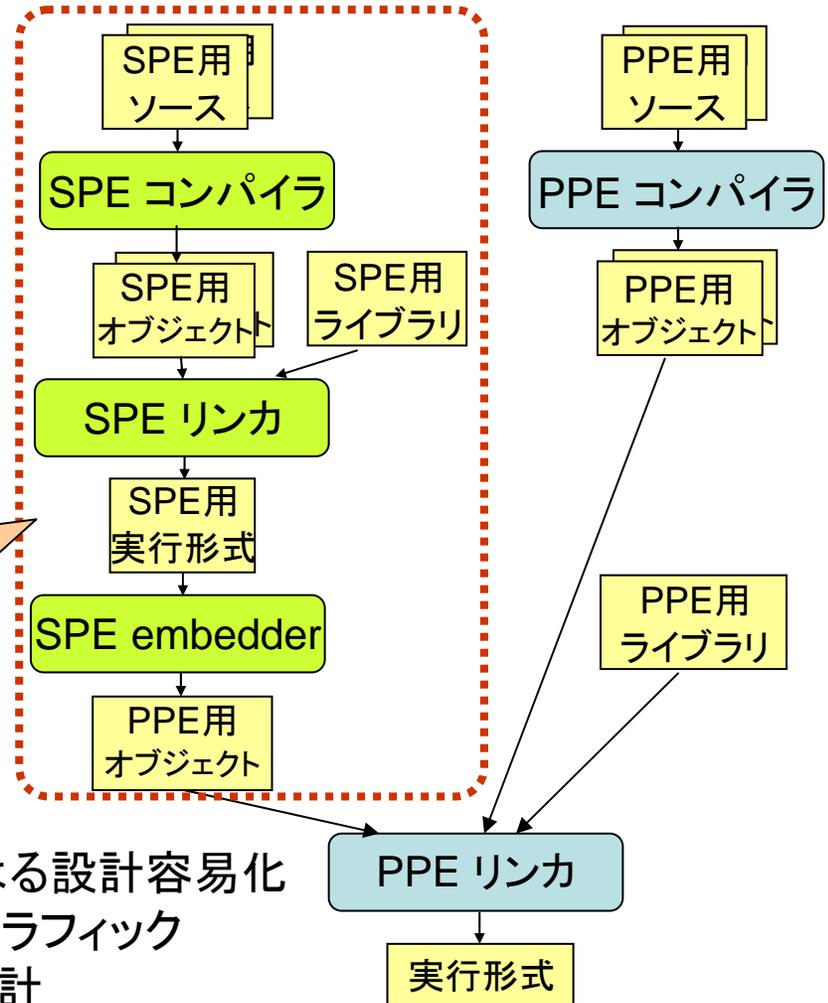
PPE:
汎用プロセッサ
SPE:
メディア処理、
3Dグラフィックス
処理に適した、
専用プロセッサ

SPE用コード
の生成



- ・プログラム開発による設計容易化
- ・メディア処理、3Dグラフィック
処理等に最適化設計

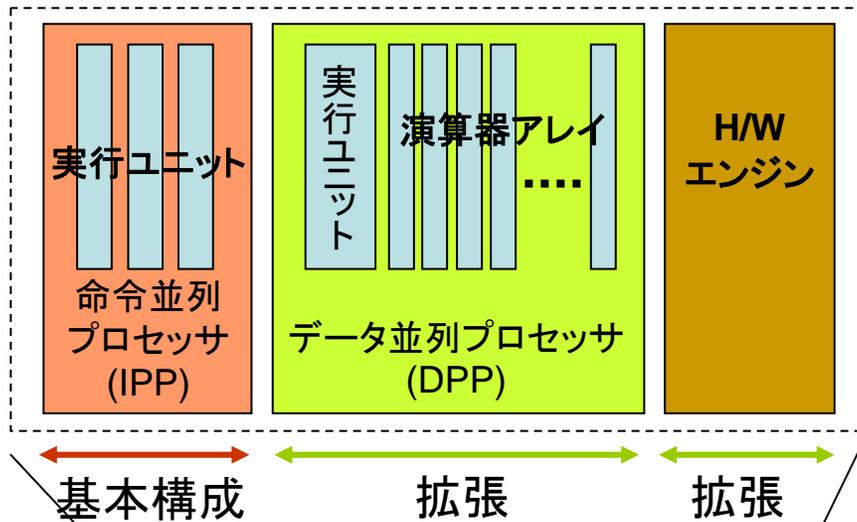
コンパイラ サポート



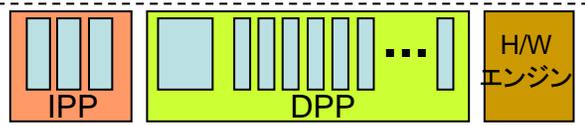
プラットフォーム・ベースによるSoC

～ UniPhier ～

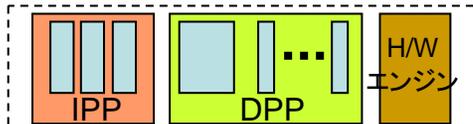
UniPhier ブロック図



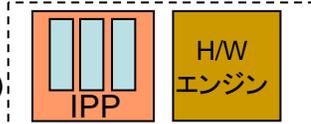
SoC A
(据え置き
車載AV機器向け)



SoC B
(携帯型
AV機器向け)



SoC C
(携帯電話機向け)



対象とするシステムの要求
に対しSoCを
最適化

ソフトウェア、ハードウェア
設計環境

SoC A SoC B SoC C

ソフトウェア
プラットフォーム



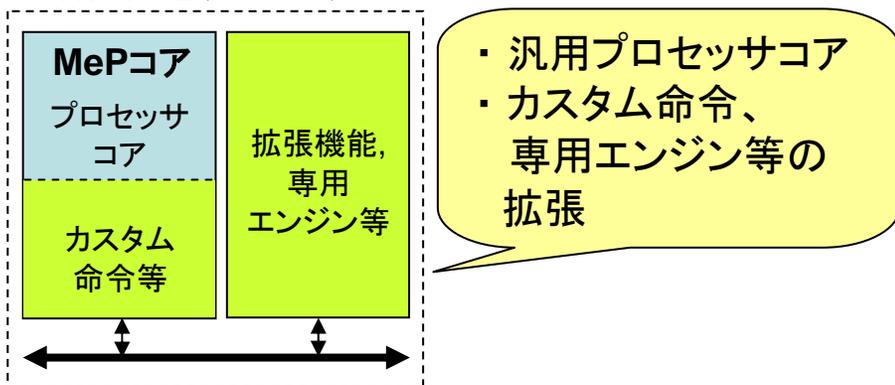
- ・ 多様なシステムに向けたH/Wの最適化
- ・ ソフトウェア開発の容易化

プラットフォーム・ベースによるSoC

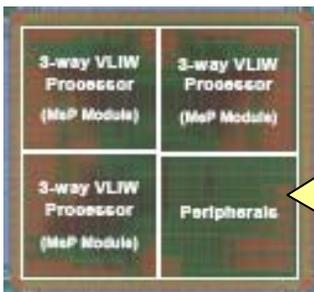
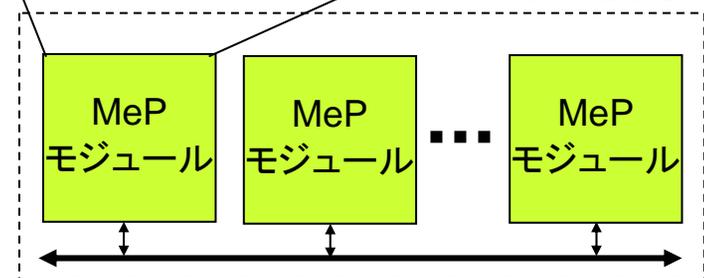
～ MeP ～

MePブロック構成

MePモジュール

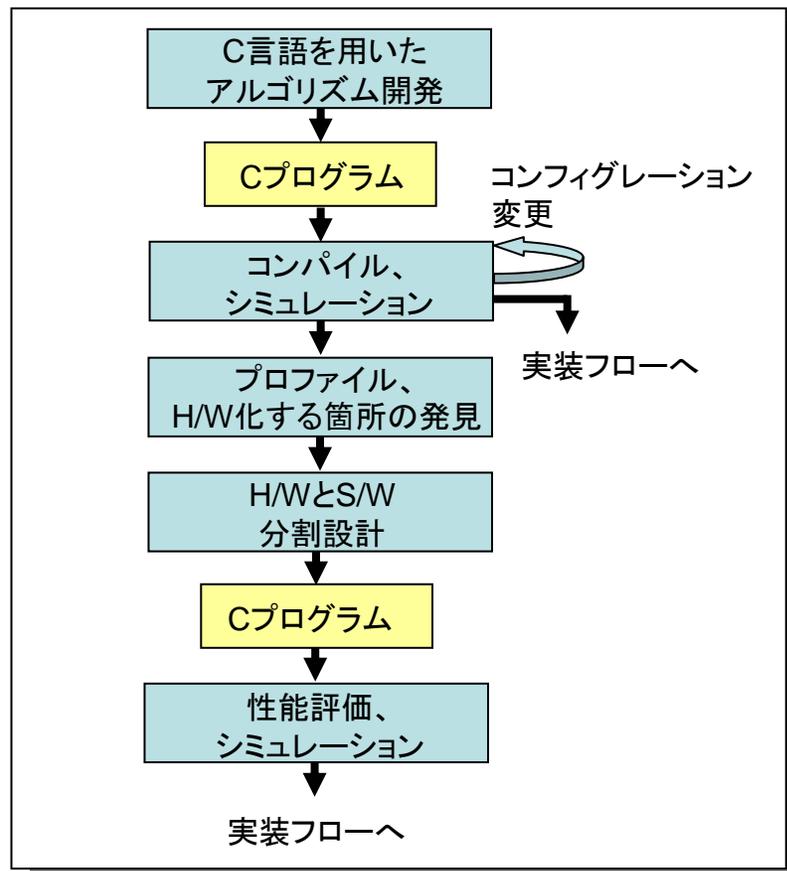


MeP SoC



画像処理用に3つの
カスタムされたMeP
モジュールと周辺機能
で構成

アプリケーション実装フロー



- ・ 多様なシステムに向けたH/Wの最適化設計
- ・ プログラムによる設計の容易化

プラットフォーム・ベースの各設計手法の比較

■ プロセッサ+ソフトウェア

- ☺ Cなど、プログラム言語を用いた開発による設計生産性の高い開発が可能
- ☹ アプリケーションを実装するハードウェア(命令セット)が固定

■ コンフィギャラブル・プロセッサ+ソフトウェア

- ☺ プロセッサの命令のカスタム化、H/Wの拡張によりアプリケーションに最適化
- ☹ カスタム化に対するハードウェア設計が必要、製造後の設計変更に対応しない

■ リコンフィギャラブル・プロセッサ+ソフトウェア

リコンフィギャラブル(実装する論理をソフトウェアで再構成可能な機能)を使う事で、

- ☺ カスタム化はSoCの製造後にも適用可能
開発の短TAT化、SoCの共通化による少品種大量生産化
- ☹ 再構成のために、面積、性能のオーバヘッドが必要
- ☺ 再構成機能を生かした高並列処理など、カスタム化による性能の向上が可能
- ☺ カスタム化命令では、プログラム言語との親和性が高く、Cなどを用いた設計生産性の高い開発が可能

RedefisとSystemMorphによるSoCプラットフォーム

Redefis

SoCにおけるカスタムロジックの実装

- リコンフィギャラブル・プロセッサを用い、C言語で記述されたソースコードから自動的にアプリケーションにカスタム化した命令セットを生成し、プログラムを実行

SystemMorph

最適化設計の自動化

- プログラム実行をモニタし、最適化のヒントとなるプログラム箇所を発見、最適化を自身で適用

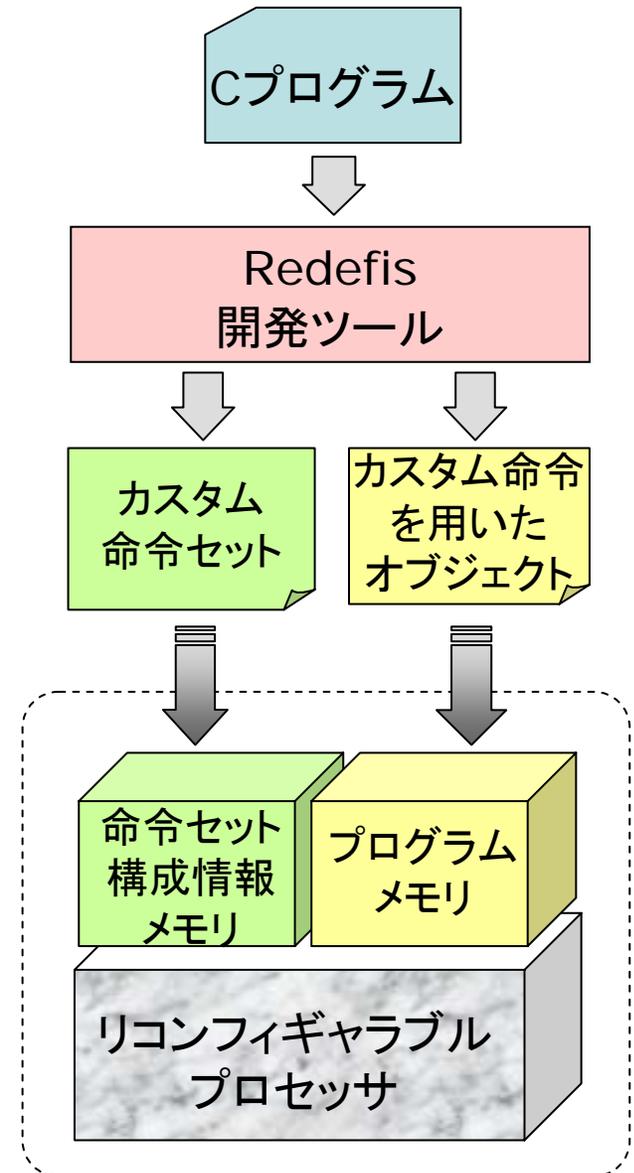
Redefisとは?

実装すべきシステム機能に応じて
ISA(命令セット・アーキテクチャ)を
「再定義可能」としたリコンフィギャラブル・
プロセッサ

設計者が、SoC上に実装したい
機能をC言語で記述

Redefis用の開発ツールにより
Cプログラムに応じ、プロセッサの
命令セットを自動的にカスタム化

実装したいシステム機能を
「専用プロセッサ+ソフトウェア」に
よる構成で実現



Redefisとは?

■ 性能向上

- アプリケーションに応じたカスタム命令セットの生成による高性能化

■ コスト、短TAT化への対応

- カスタムロジックをSoC上に実現する、プラットフォーム・ベース設計による開発の効率化
- C言語を用いたアプリケーションの実装開発による設計生産性向上
- リコンフィギャラブル・プロセッサにより、カスタムロジックをSoCの製造後に実装
 - SoC開発の短TAT化
 - 少品種大量生産化

RedefisとSystemMorphによるSoCプラットフォーム

Redefis

SoCにおけるカスタムロジックの実装

- リコンフィギャラブル・プロセッサを用い、C言語で記述されたソースコードから自動的にアプリケーションにカスタム化した命令セットを生成し、プログラムを実行

SystemMorph

最適化設計の自動化

- プログラム実行を用い、最適化のヒントとなるプログラム箇所を発見、最適化を自身で適用

SystemMorphとは?

～ 最適化のヒントの発見 ～

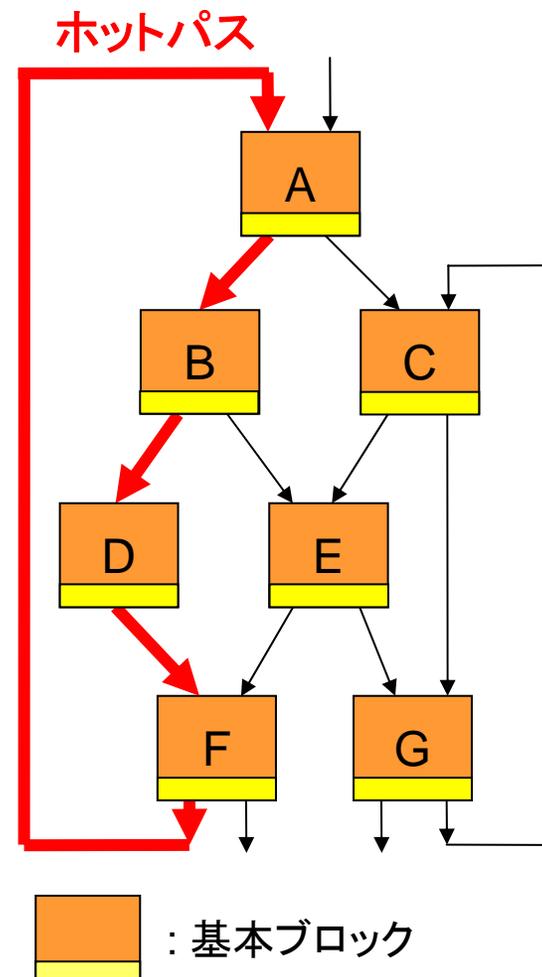
■ ホットパス・プロファイリング

• ホットパス

プロセッサによるプログラム実行で高い頻度で繰り返し実行される命令列

• ホットパス・プロファイリング

プログラムの実行トレースを使い、ホットパスを推定

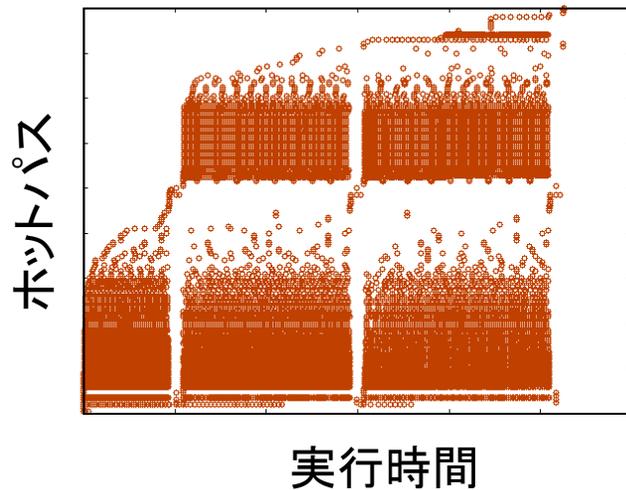


SystemMorphとは？

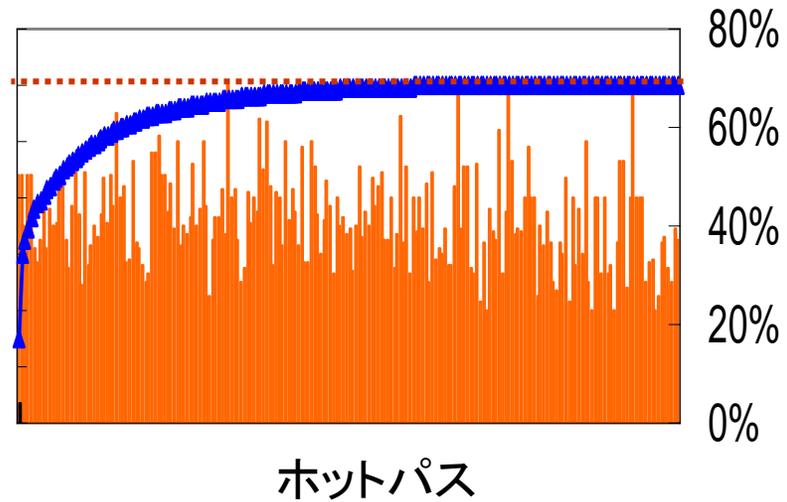
～ホットパスの検出～

- **ホットパスの検出例**
 - H.264エンコーダのホットパス

検出されたホットパス



ホットパスの
実行時間に占める割合



RedefisとSystemMorphによるSoCプラットフォームの応用 ～ ソフトウェア無線用SoCプラットフォーム ～

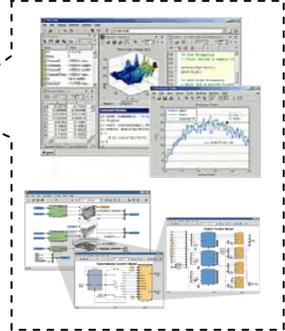
① Matlab/Simulinkを用い無線アルゴリズムを開発

- 無線用関数ライブラリ
- ブロック線図による開発
- コード生成

Matlab/Simulink
無線通信アルゴリズム開発

H/W, S/W分割設計自動化

汎用処理



Cプログラム

HW記述

Cプログラム

② Cプログラムに適したカスタム命令セットを生成

- ソフトウェアによる無線アルゴリズムの実装

Redefis開発ツール
SystemMorph最適化

H/W開発
IP適用

CPU用
開発ツール

④ ホットパス情報による最適化

- カスタム命令の最適化
- S/W, H/W分割

ソフトウェア無線アーキテクチャ・プラットフォーム
性能シミュレーション

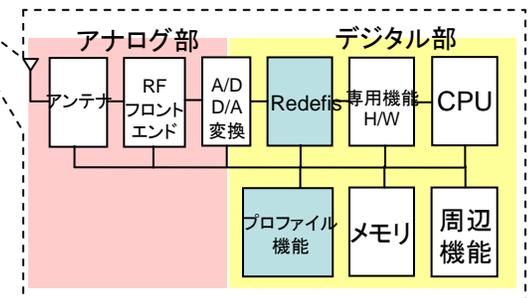
- プロファイル取得、性能最適化へフィードバック

ソフトウェア無線
アーキテクチャ、
及び評価環境

③ ホットパス・プロファイリングによる最適化適用箇所の発見

実評価環境

リCONFIGャラブル・プロセッサを用いた、ソフトウェア無線用SoC



RedefisとSystemMorphによるSoCプラットフォームの応用 ～ ソフトウェア無線とは ～

無線通信では多数の規格が並存し、無線通信毎にSoC、それらを使った端末が必要

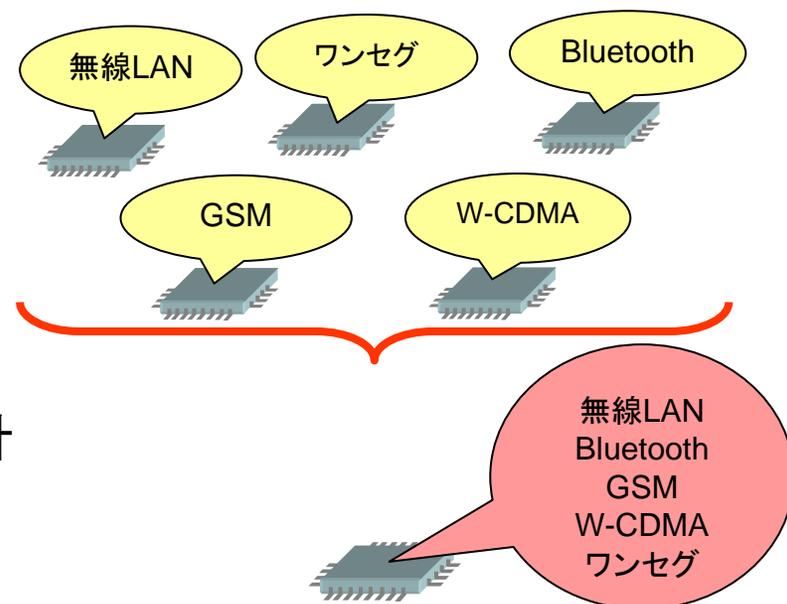
• ソフトウェア無線

ソフトウェアを入れ替える事で、同じSoCで複数の無線通信規格を利用可能にする技術

• ソフトウェア無線用SoCの開発

市場ニーズに応じた短期間でのSoC開発

- SoC設計のプラットフォーム化
- ソフトウェアによる柔軟性とハードウェア性能の実現
- ... リコンフィギュラブル機能を用いた設計

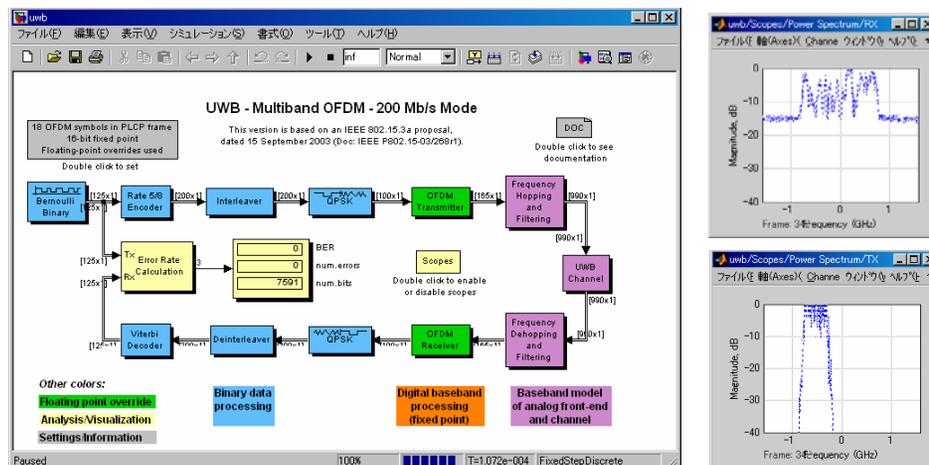


RedefisとSystemMorphによるSoCプラットフォームの応用 ～ Matlab/Simulink ～

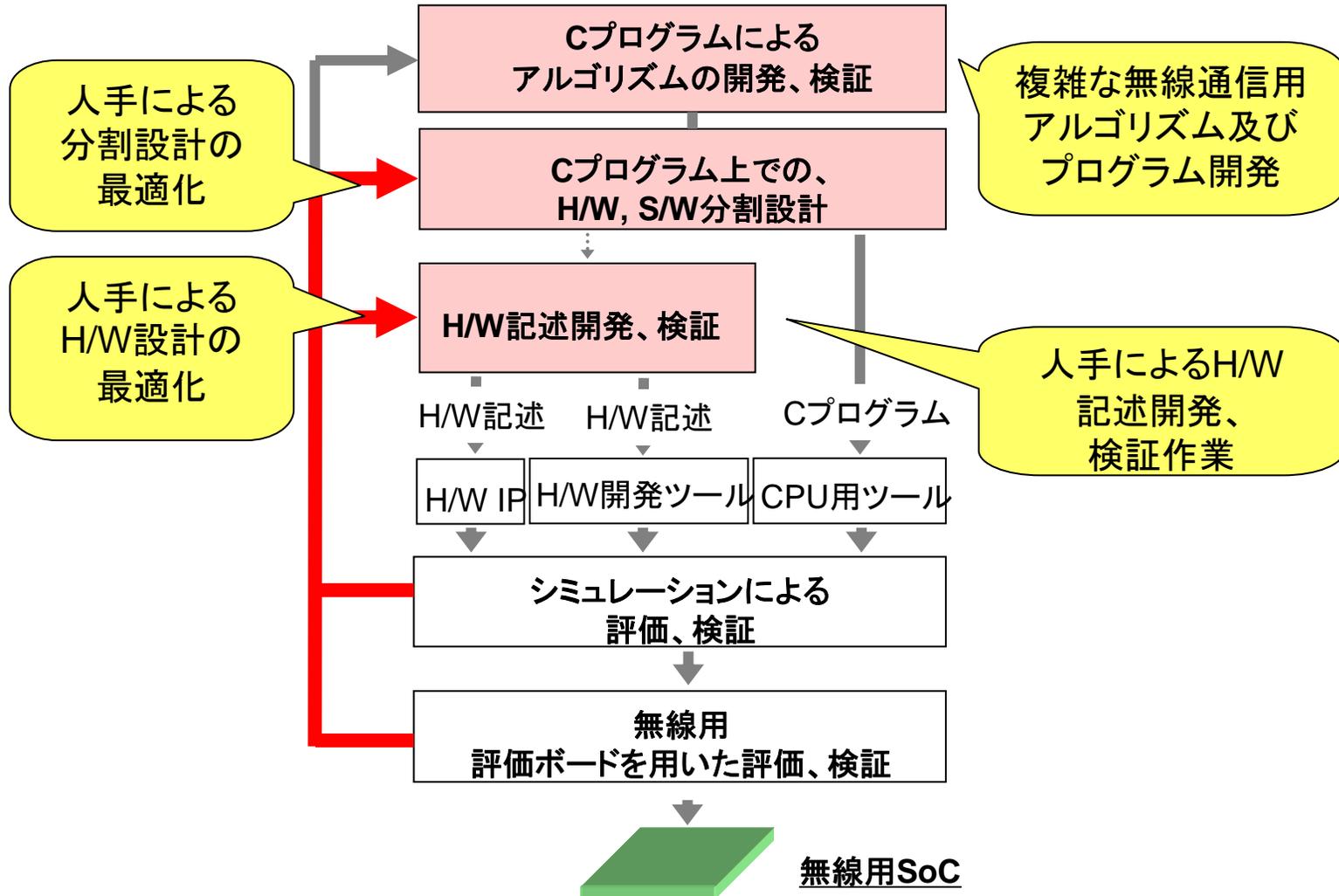
Matlab/Simulinkを用いた無線通信アルゴリズム開発、検証

- ・ 予めフィルタ、線形代数、解析などの無線通信向け関数が用意
- ・ ブロック線図による直感的な開発、結果の可視化
- ・ 構築されたモデルのCソース、ハードウェア記述のコード生成

- ・ 無線通信アルゴリズム開発、検証の効率化
- ・ コード生成

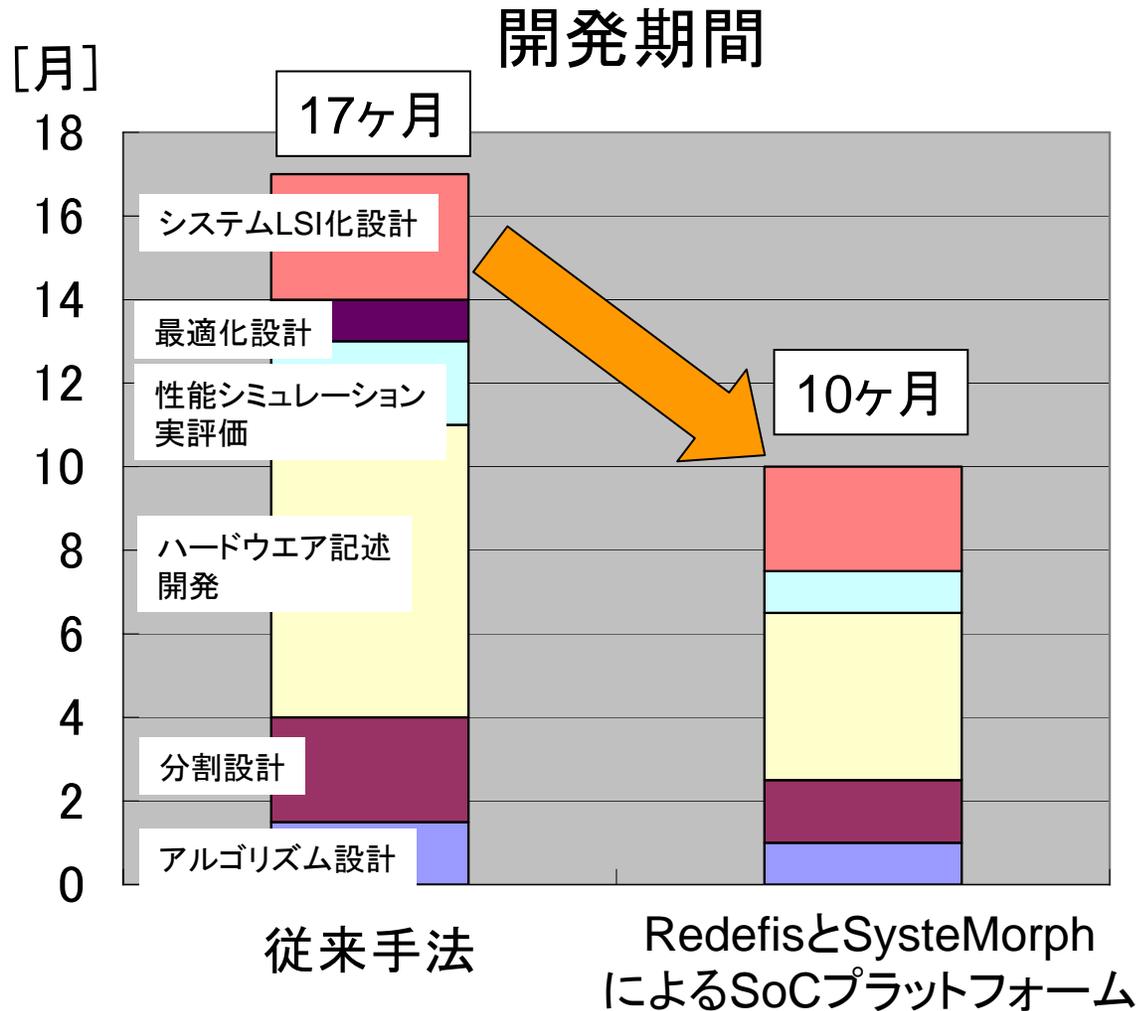


無線用SoCの開発 ～ 従来手法 ～

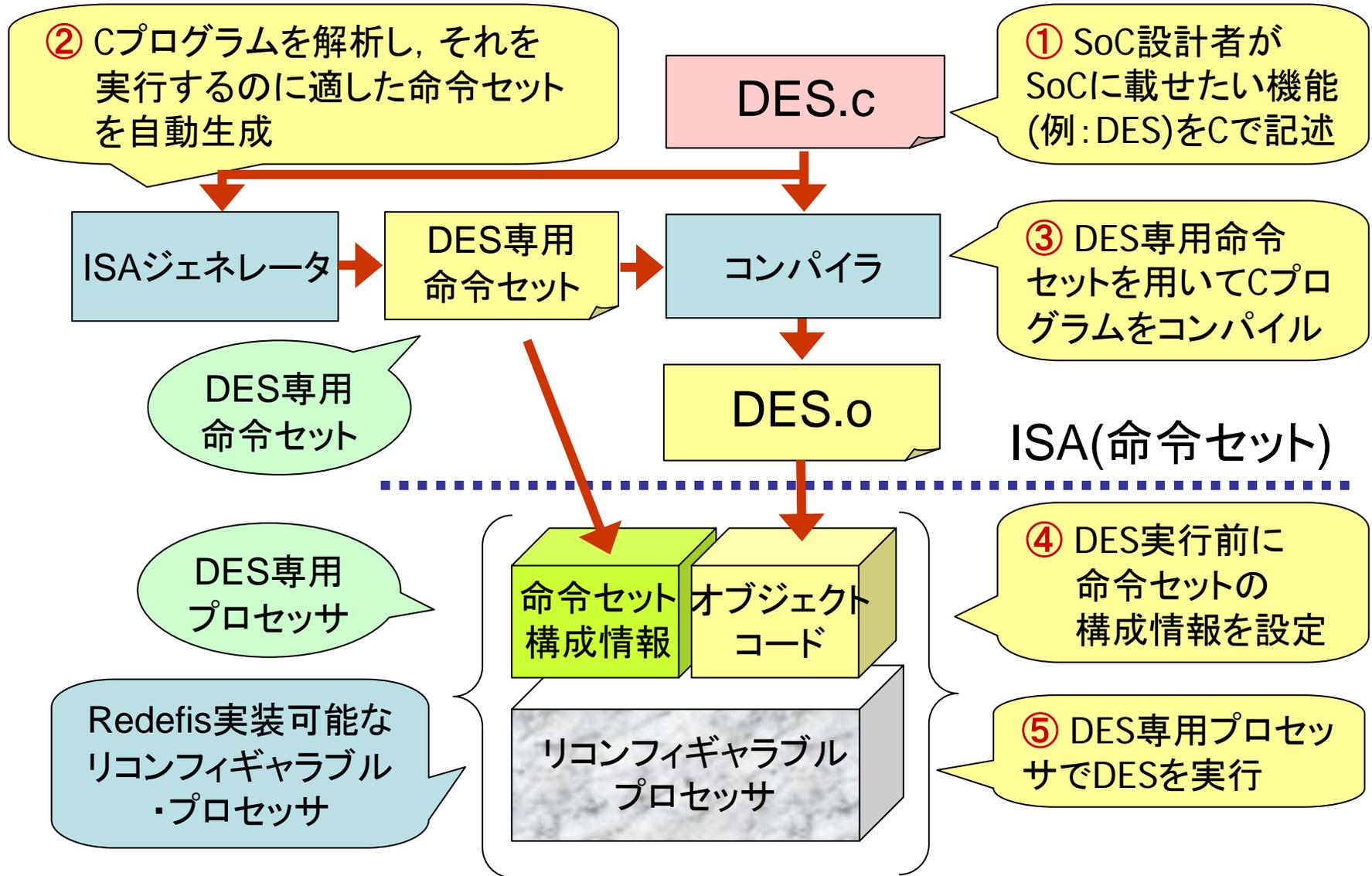


RedefisとSystemMorphによるSoCプラットフォームの応用 ～ 無線用SoCの開発期間 ～

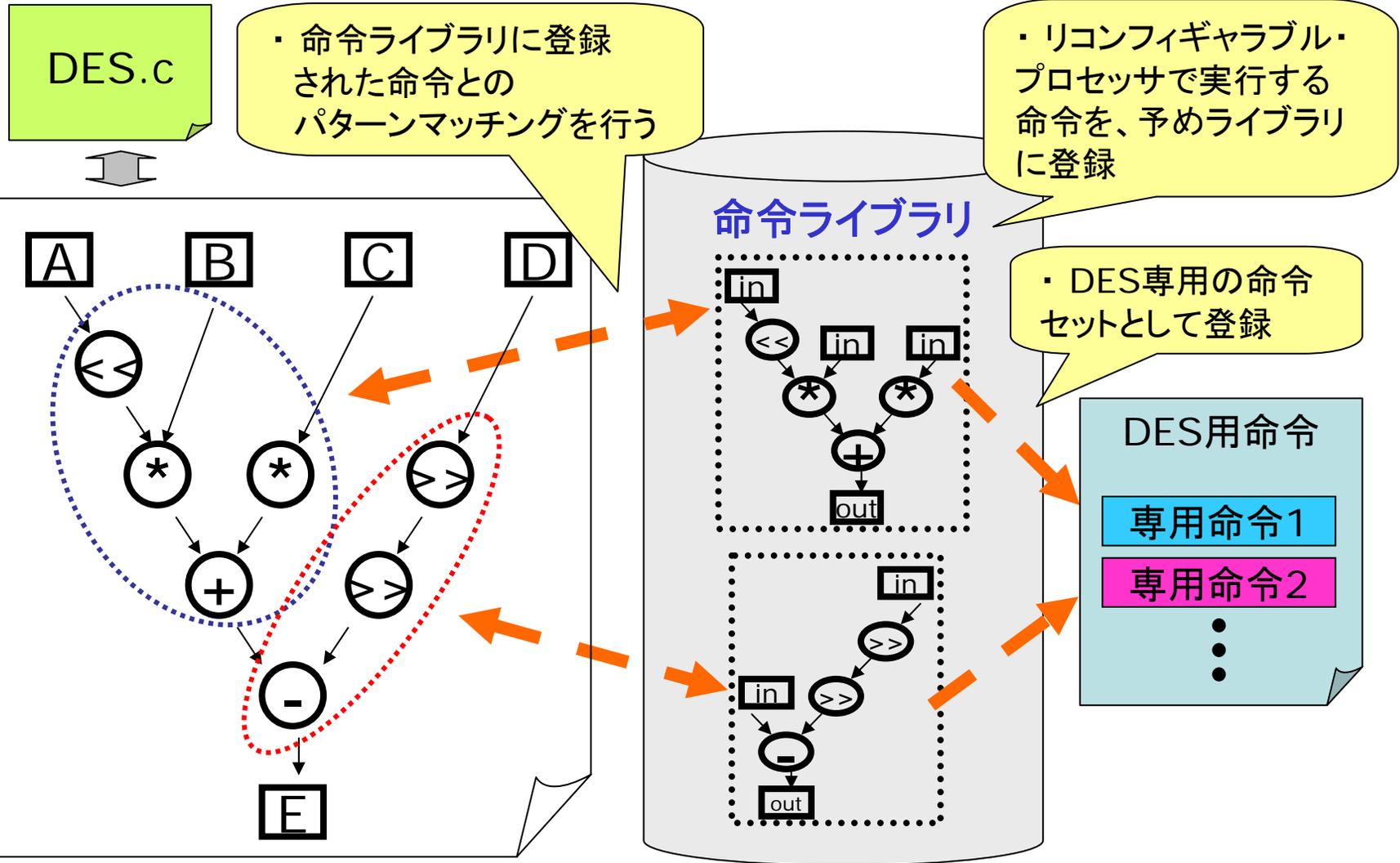
• 開発期間の試算



Redefisとは?

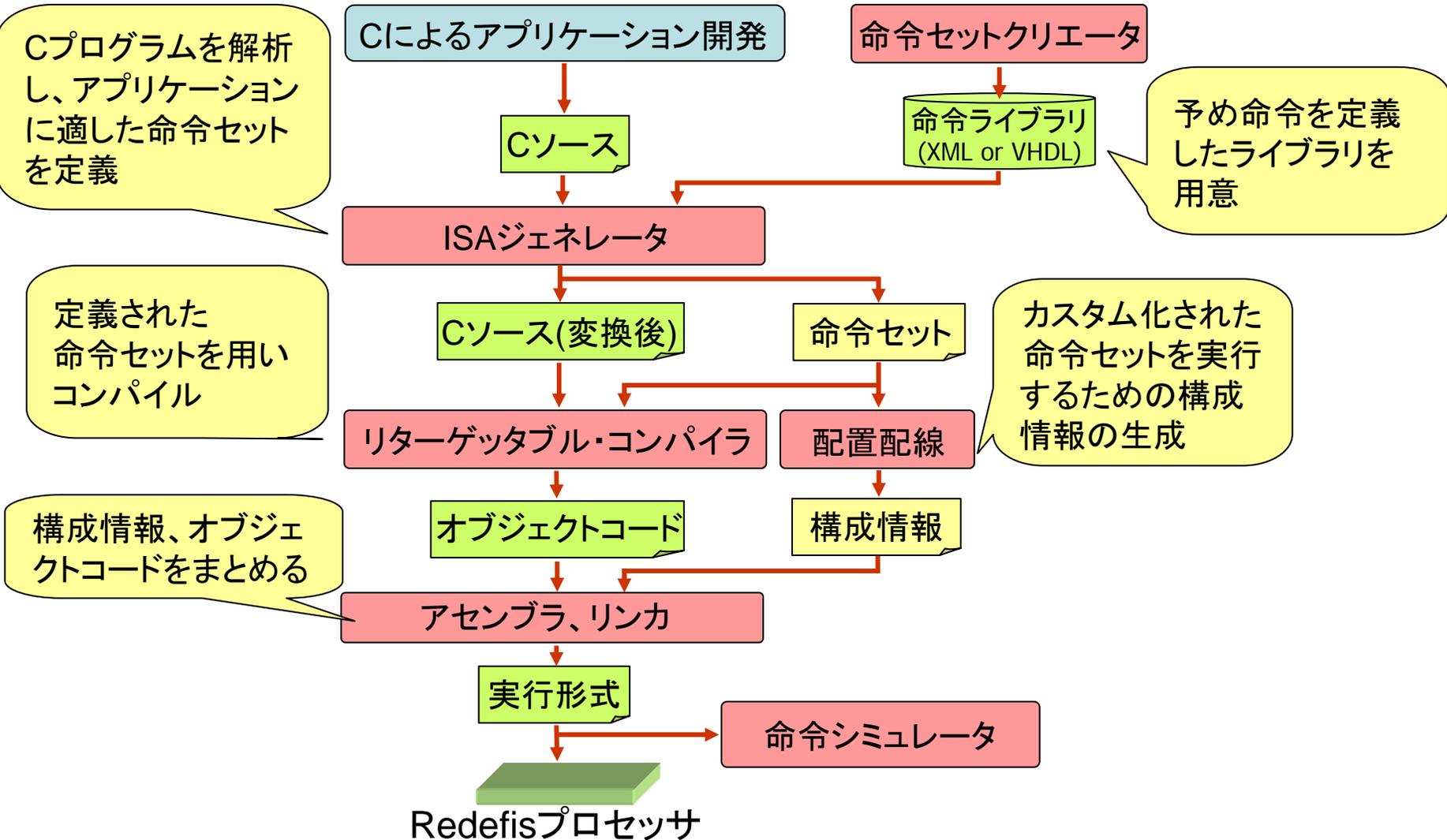


Redefisとは? ~命令セットの定義~



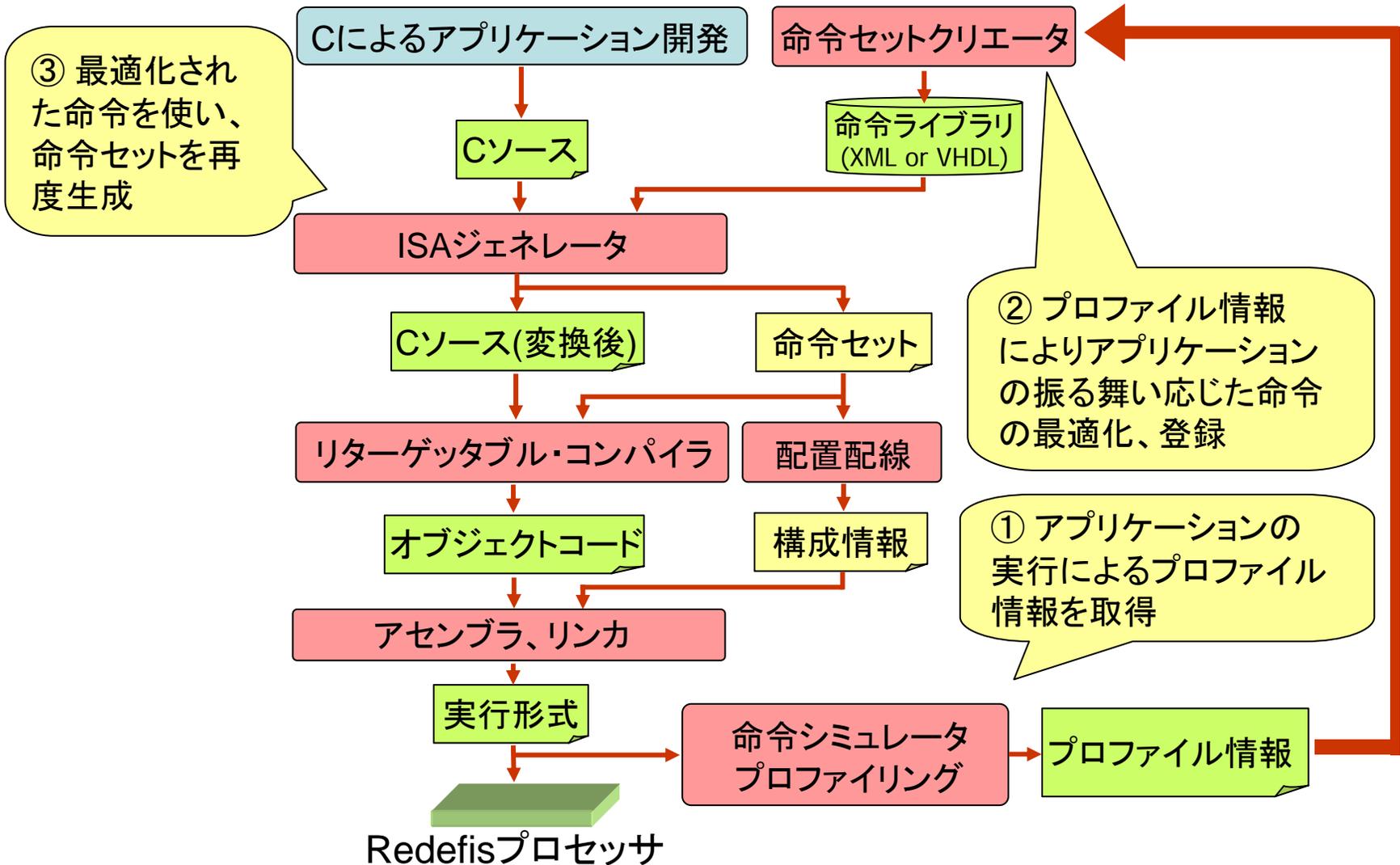
Redefis開発環境

～アプリケーション実装フロー～



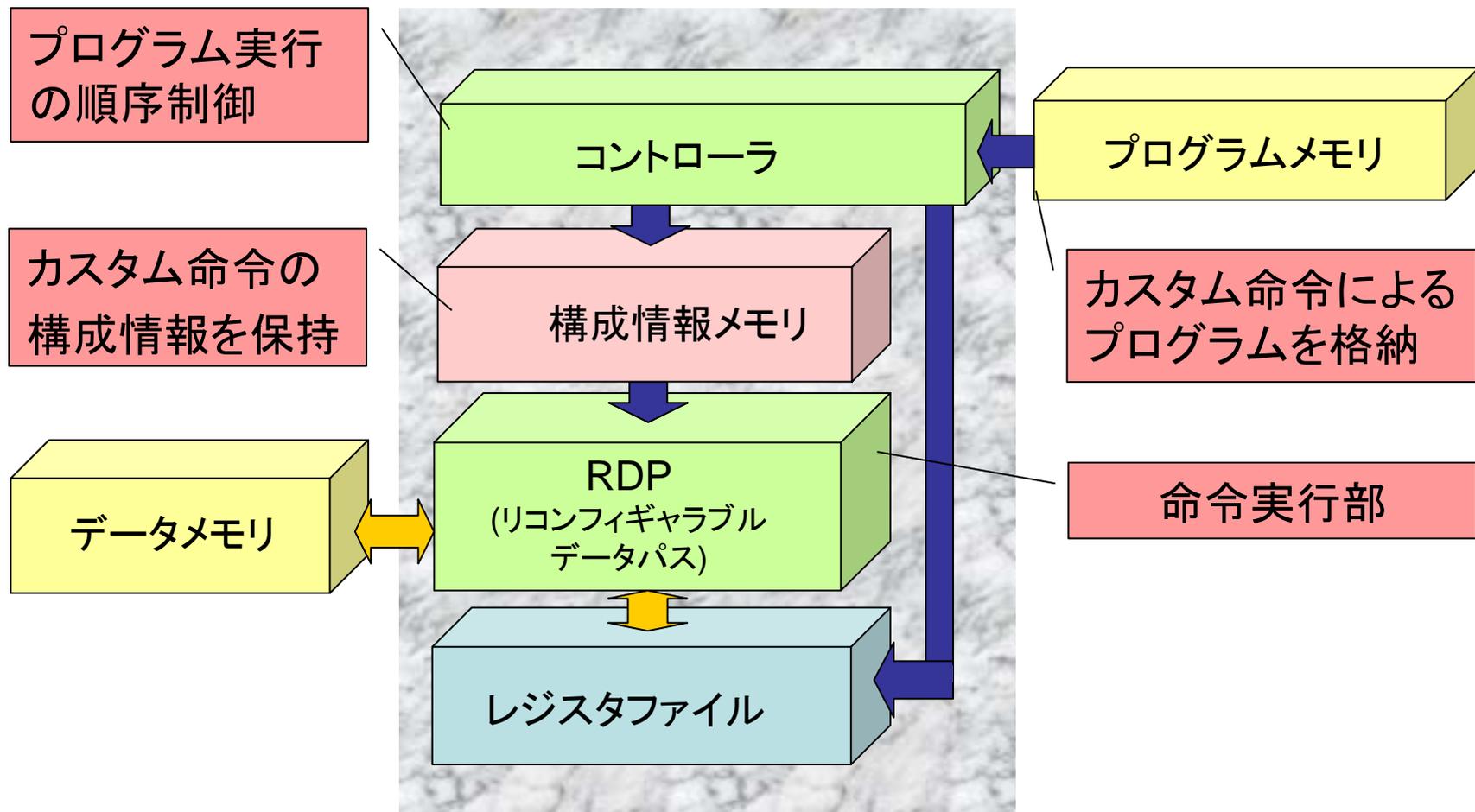
Redefis開発環境

～アプリケーション実装フロー～



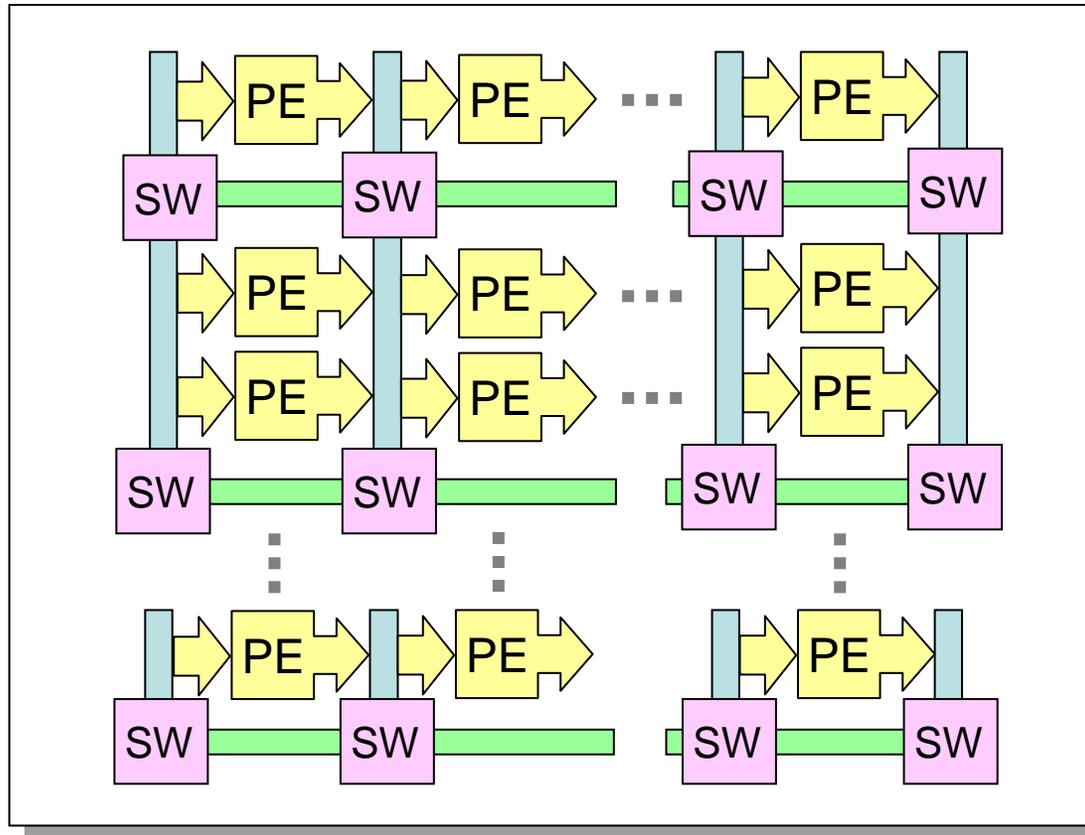
Vulcan

～試作りコンフィギュラブル・プロセッサ～



Vulcan

～RDP(リコンフィギャラブル・データパス)部の構成～



PE : Programmable Element

任意の論理関数を割り当て

SW : Switch

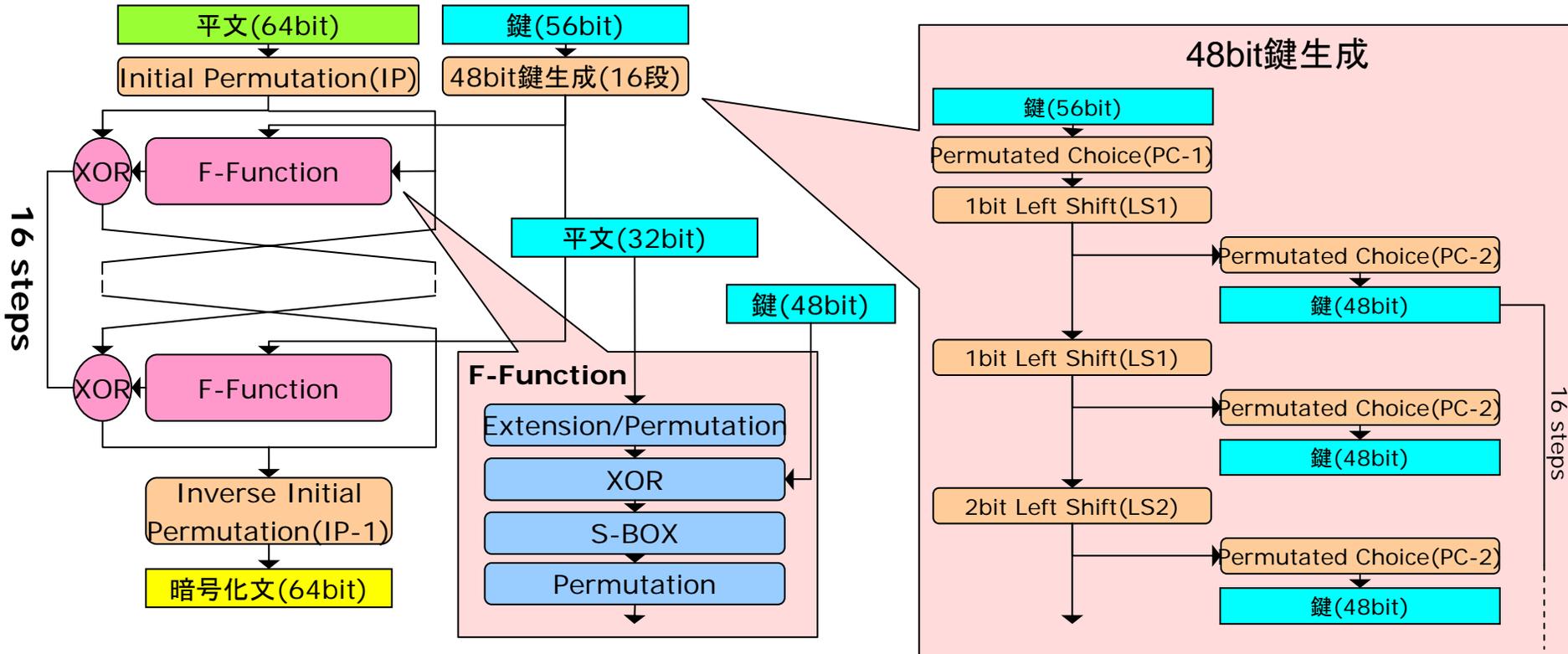
垂直方向バス(64bit)と水平方向バス(64bit)のデータの受け渡しの選択

Vulcan LSI搭載ボード

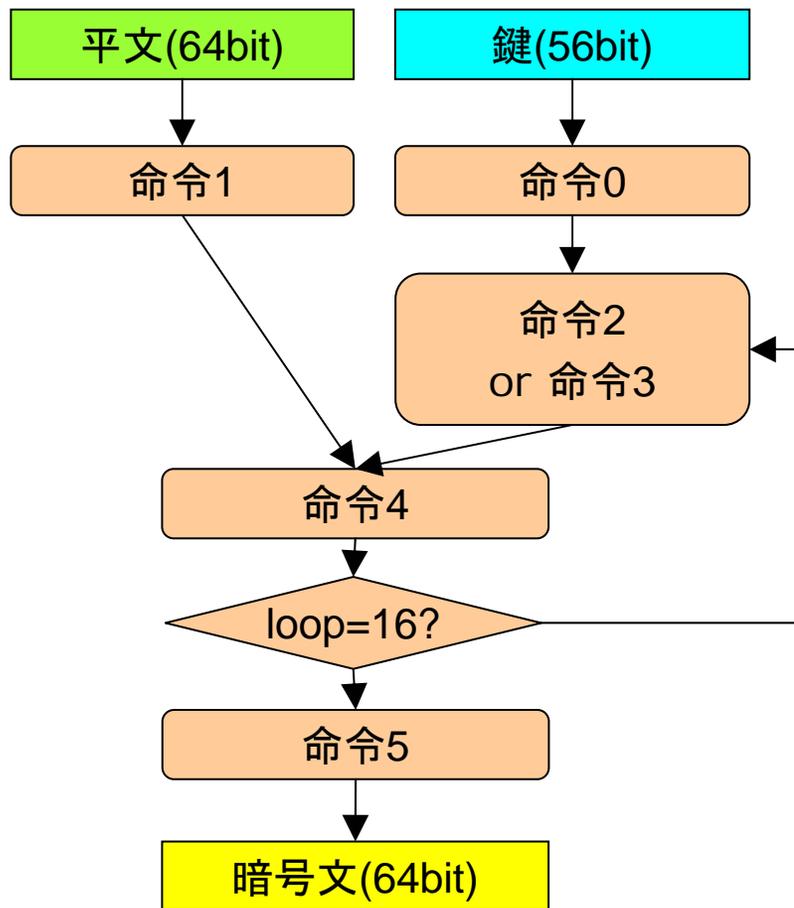


アプリケーション例：DES暗号化

DES(共通鍵を用いた暗号化アルゴリズム)をCプログラムで記述し、Vulcanを用いてDES専用プロセッサを開発



Vulcan上でのDES暗号化処理



命令	内容
0	鍵の読み込みと鍵の転置 (PC-1)
1	平文の読み込みと初期転置 (IP)
2	1bit左循環シフト (LS1)
3	2bit左循環シフト (LS2)
4	縮約転置 (PC-2)とF関数, および, XOR
5	逆初期転置 (IP-1), および, 出力

- オブジェクトコード・サイズ: 24命令
- 実行命令数: 35命令 (1回の暗号化)

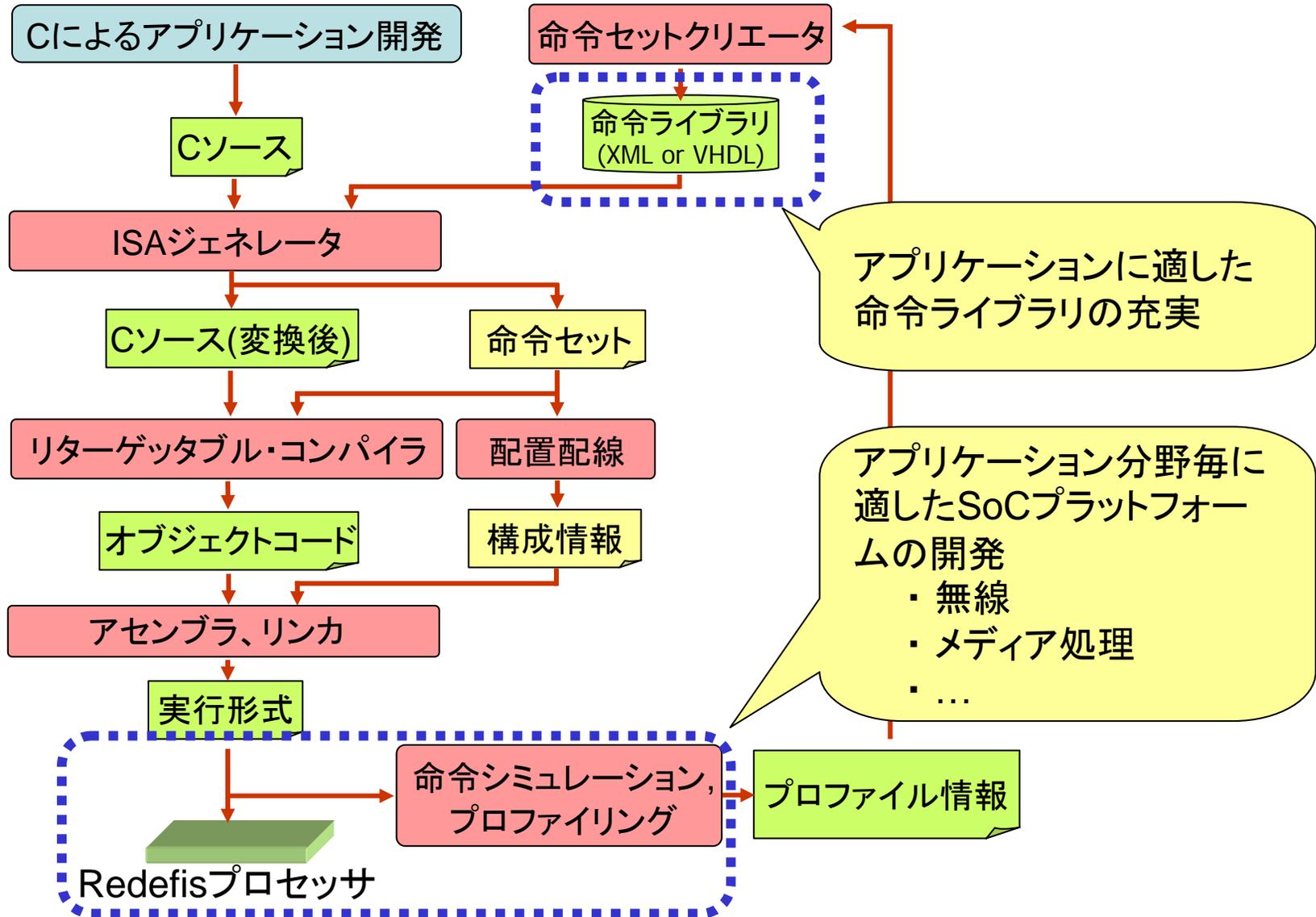
DES暗号化処理の性能比較

プロセッサ	動作周波数	スループット
Vulcan	6.25MHz	570KB/sec
Pentium4	2.4GHz	150KB/sec

- Pentium4でのDES暗号化処理
 - DES暗号化処理をC言語で忠実に記述したプログラムをコンパイルして実行
 - GCCバージョン:2.95.3
 - 最適化オプション:-O2

Redefis開発環境

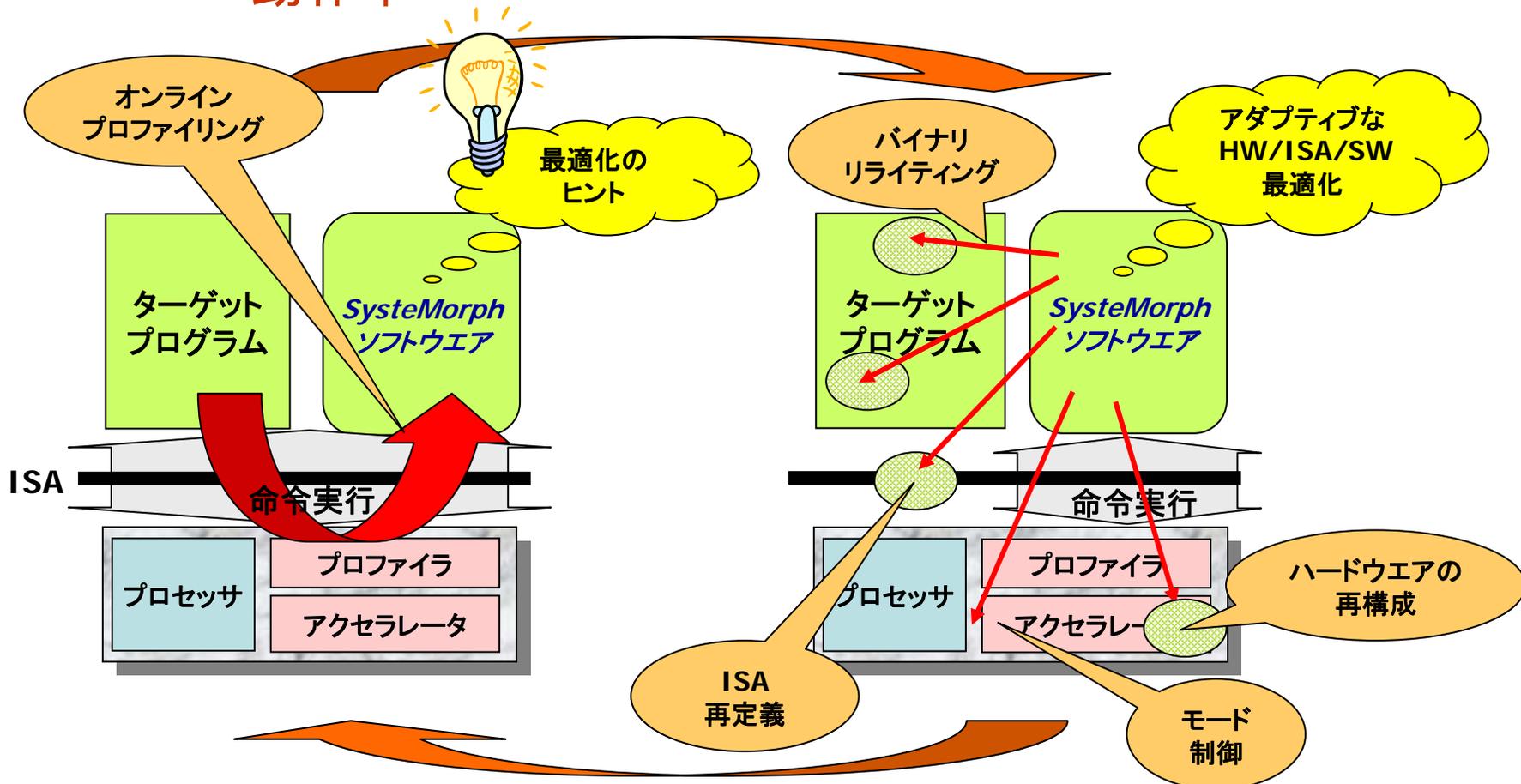
～多様なアプリケーションへの対応～



SystemMorphとは？

アプリケーション・プログラム
動作中...

最適化処理中...

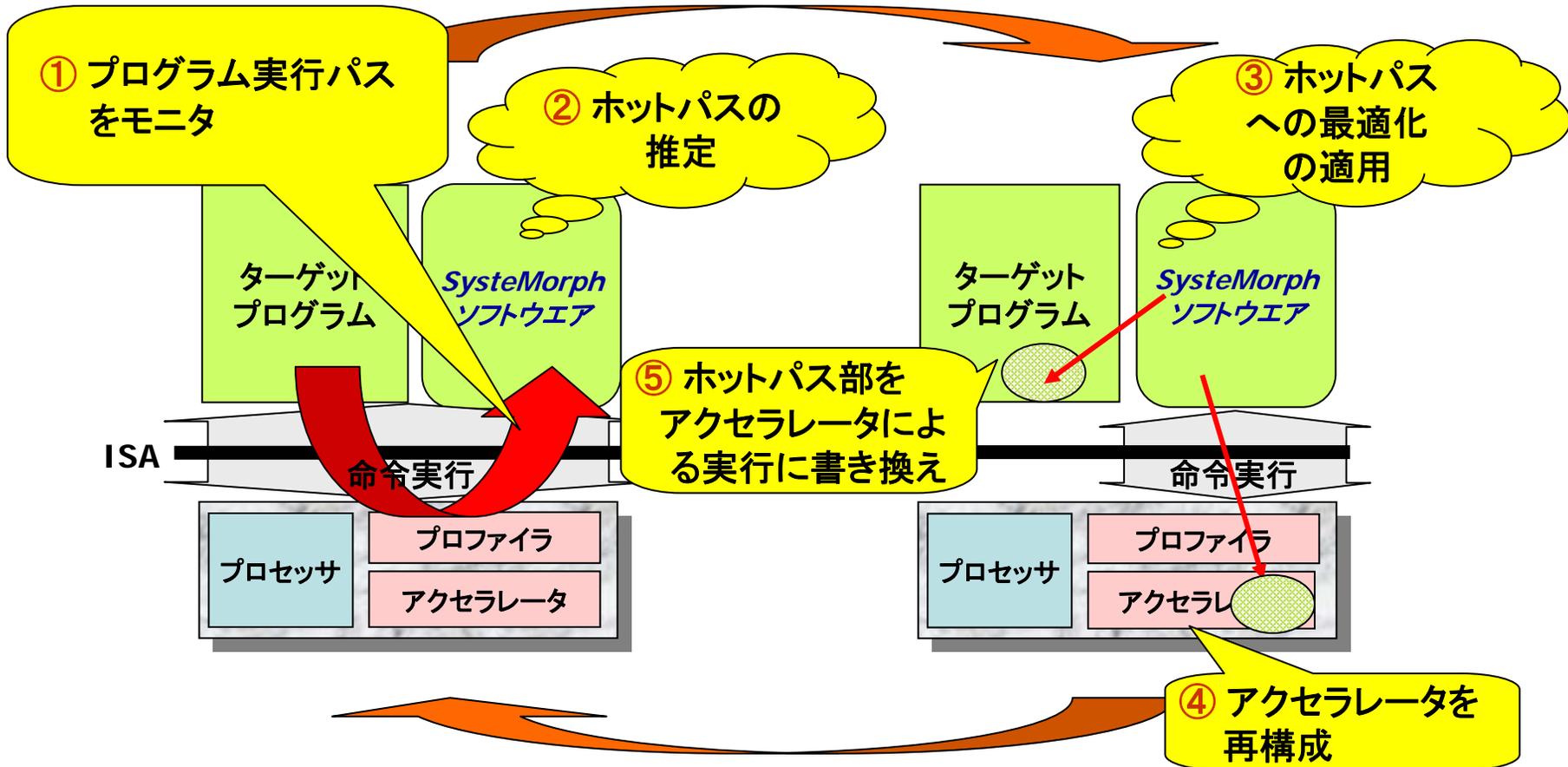


SystemMorphとは

～ 最適化の適用例 ～

アプリケーション・プログラム
動作中...

最適化処理中...



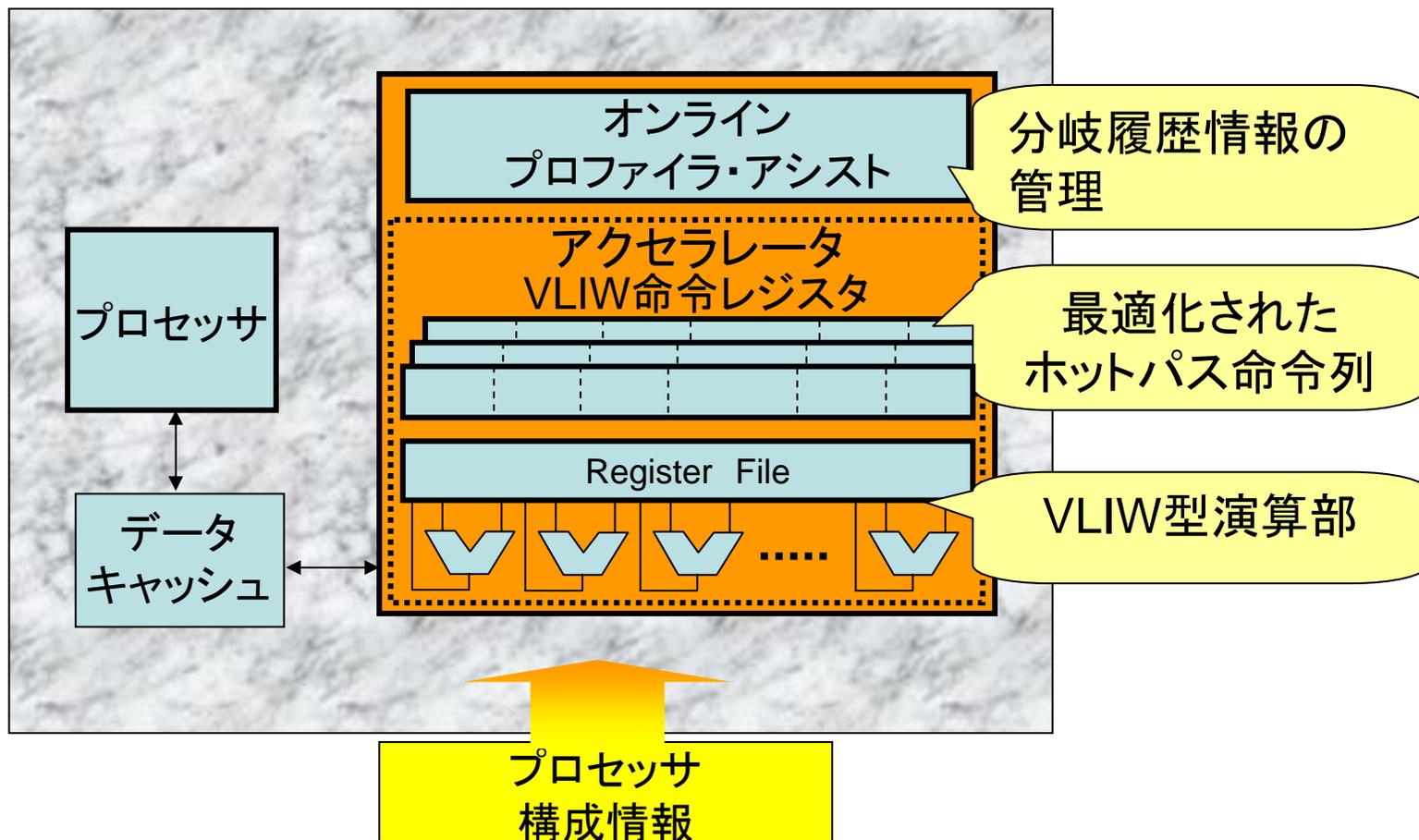
SystemMorphとは?

- SoCの最適化設計の自動化
 - SoCの設計時での最適化設計の自動化
 - Redefis/SystemMorphによるSoCプラットフォーム
 - SoCの出荷後、SoCの使われ方に応じ、SoCが動的に最適化
- プロファイリング情報の利用
 - ユーザ毎の最適化サービス
 - 次のSoC設計へのフィードバック
 - オンライン・セルフチェックングによる対故障化、高信頼性化への応用

SystemMorph

～ 動的最適化可能なSoC ～

VLIW(Very Long Instruction Word)型
アクセラレータによる
SystemMorphプラットフォーム



SystemMorphとは?

～ 最適化のヒントの発見 ～

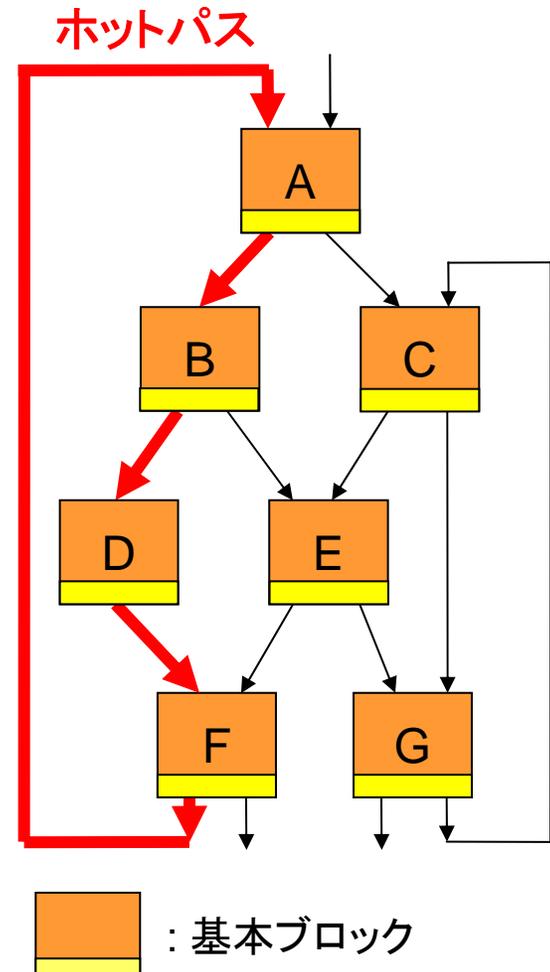
■ オンライン・ホットパス・プロファイリング

• ホットパス

プロセッサによるプログラム実行で
高い頻度で繰り返し実行される命令列

• オンライン・ホットパス・プロファイリング

オンライン(プログラムの実行中)に
プログラムの実行トレースをモニタし、
ホットパスを推定



SystemMorph

～ 動的最適化の流れ ～

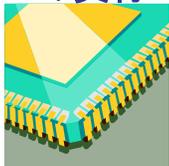
① ホットパスを推定

② ホットパス命令列に対し、アクセラレータによる並列実行に向けた最適化の適用

③ ホットパス部でアクセラレータの呼び出しを行うよう、ターゲットプログラムを書き換え

④ 並列実行により性能向上可能なVLIW型アクセラレータにより、最適化されたホットパス命令列の加速実行

ターゲットプログラムの実行

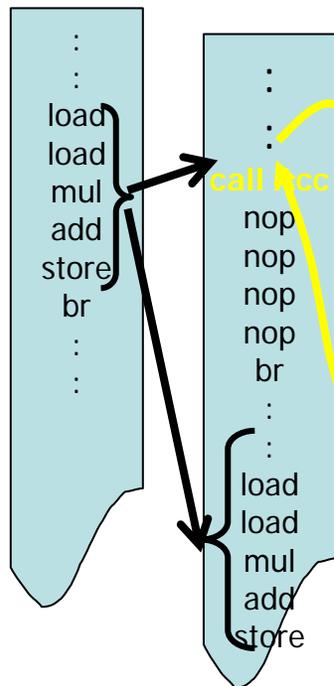


プロセッサ

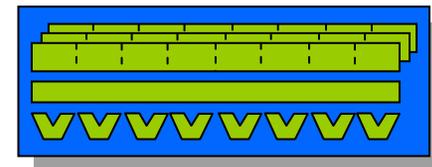
Hot Path 推定

最適化の適用

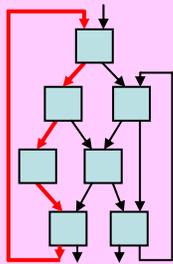
バイナリ書き換え



アクセラレータによるホットパスの加速実行



アクセラレータ



並列性、命令数などからアクセラレータ実行による性能上の得失を判断

次世代SoCプラットフォーム ～ SystemMorphとRedefis ～

■ Redefis

リコンフィギャラブル・プロセッサを用いた、
命令再定義型プロセッサ

■ SystemMorph

SoCの使われ方に応じ、SoC自身による最適化

■ Redefis/SystemMorphによるSoCプラットフォーム

SoCが直面するコスト、短TAT化に応えるアーキテクチャ技術
の開発、SoCの新たな付加価値を作り出す技術の開発

福岡知的クラスター創成事業

文部科学省 知的クラスター創成事業

本研究は「次世代システムLSIアーキテクチャ」の開発として、

九州大学

ISIT(財団法人九州システム情報技術研究所)

共同研究企業

福岡IST(福岡県産業・科学技術振興財団)

の連携で実施されています

