

次世代のSoC最適化技術Systemorph

吉松, 則文
九州システム情報技術研究所

<https://hdl.handle.net/2324/9173>

出版情報 : SLRC プレゼンテーション, 2007-07-25. 九州大学システムLSI研究センター
バージョン :
権利関係 :

次世代のSoC最適化技術 SystemMorph

2007年 7月25日
九州システム情報技術研究所

吉松 則文

プラットフォームベースのSoC開発

SoCの開発コストの増大、開発期間の短期化

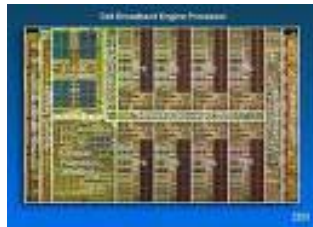
SoC設計の共通の土台(プラットフォーム)を用い、アプリケーション実装に必要な設計にフォーカスし開発期間を短縮

- 高級言語などにより開発が容易なCPU上で汎用的な機能を実現し、性能が満たせない機能をカスタムH/Wとして実装

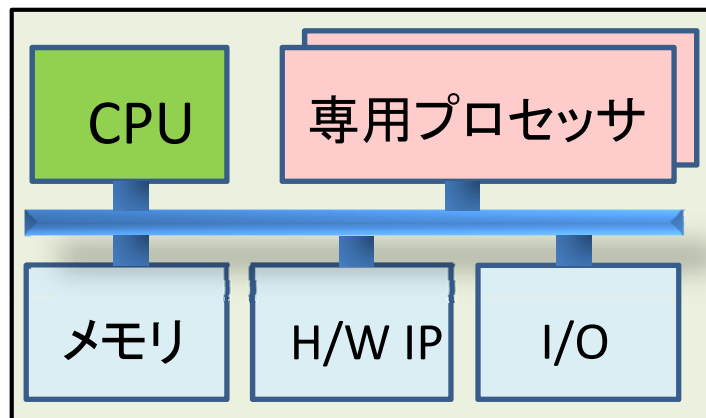
汎用プロセッサと専用プロセッサによる SoCプラットフォーム



OMAP : ARM + DSP (Texas Instruments)



CELL : Power + SPE (ソニー・東芝・IBM)

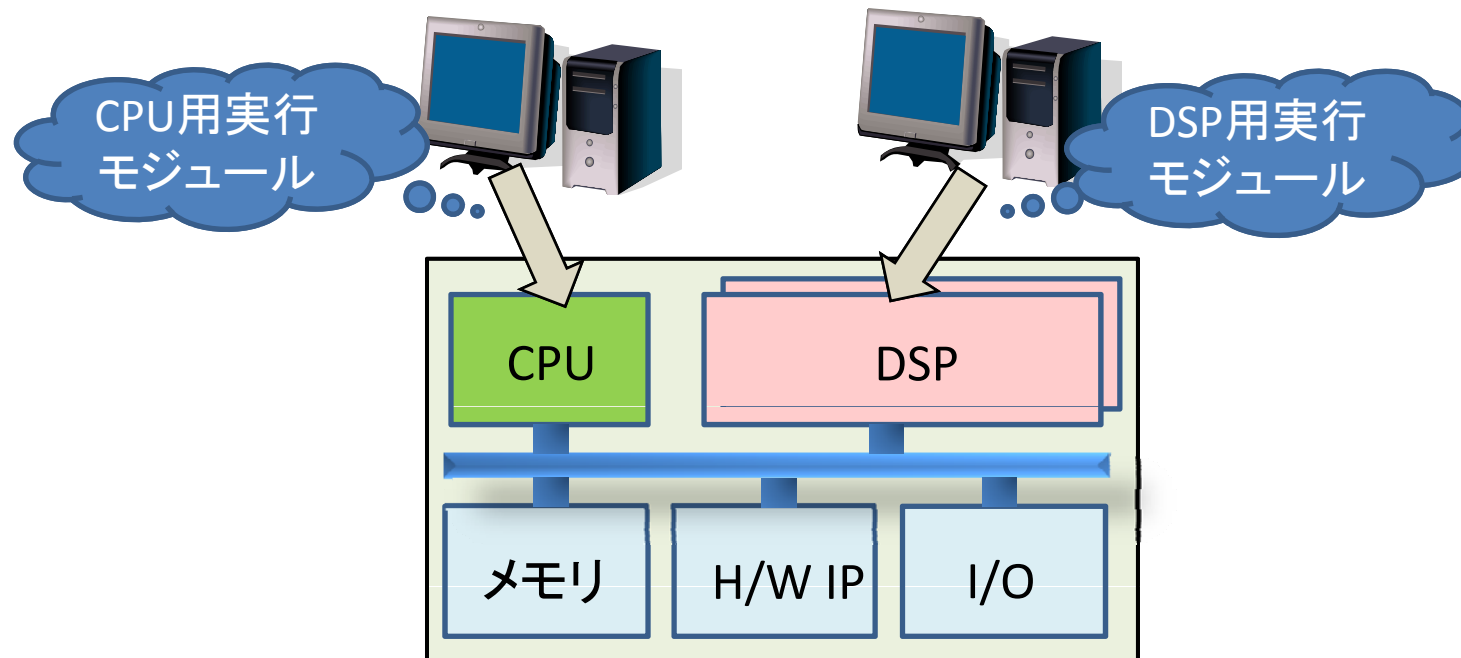


- アプリケーションに向けたカスタム化設計で、
高レベル言語(C/C++等)を用いた開発が可能
⇔ 専用ハードウェア言語による開発

汎用プロセッサと専用プロセッサによる SoCプラットフォーム

性能最適化設計の課題

- アプリケーションの性能クリティカル箇所の特定
- マルチコア用ソフトウェア開発、デバッグの煩雑さ¹⁾
- バイナリレベルのアプリケーションの最適化

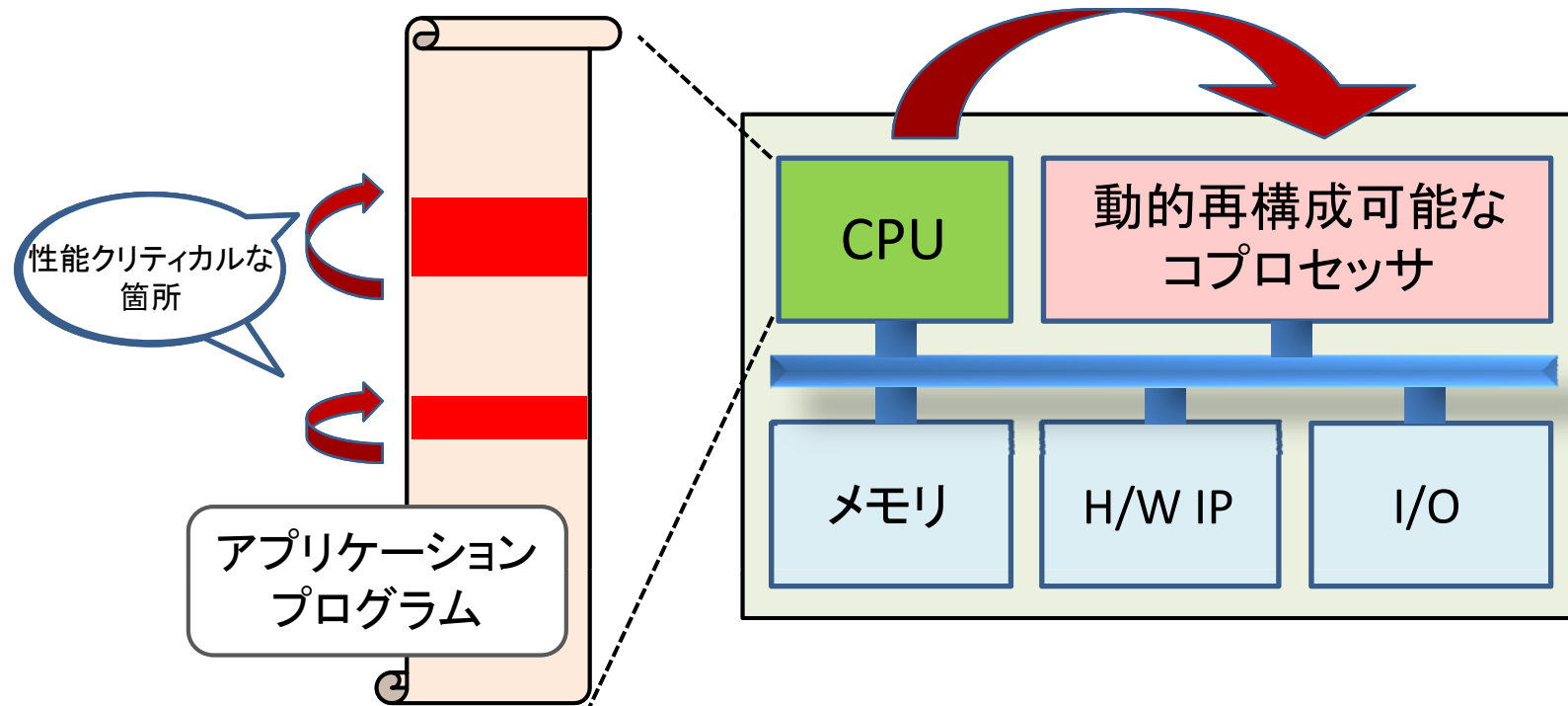


(1)『組み込みシステム向けマルチコア・プロセッサのためのソフトウェア開発支援』,情報処理学会論文紙, [Vol.48, No.SIG_4\(20070315\)](#) pp. 27-47

SystemMorph技術を用いた 性能自動最適化

ソフトウェアの最適化設計の自動化

CPU上で実行されるアプリケーションの実行時の振舞いに応じ、性能クリティカルなプログラム箇所を、再構成可能なコプロセッサに自動的にオフロードし加速実行



最適化タイミング、自動最適化による得失

	開発時	実行時
• 最適化設計期間	×	○
• アプリケーションの振舞いに応じた最適化	×	○
• 性能最適化の効果	○	△
• 最適化設計適用の範囲	○	△

(^.^)
• 最適化設計が不要
• 入力データに応じた振舞いに適応が可能。

(>_<)
タイミングを考慮したコード設計や、時間を要する高度な最適化の効果を実用することは困難。

ホットパスに着目した性能最適化

ホットパス

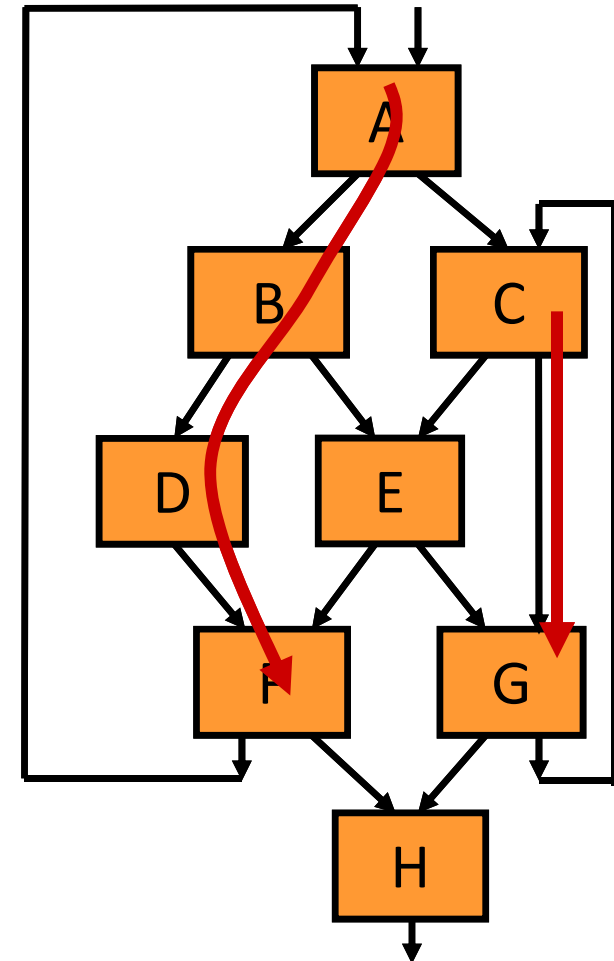
プログラム実行で、高い頻度で実行される命令列に着目



プログラムの実行で性能クリティカルなコード部分と仮定し、性能最適化

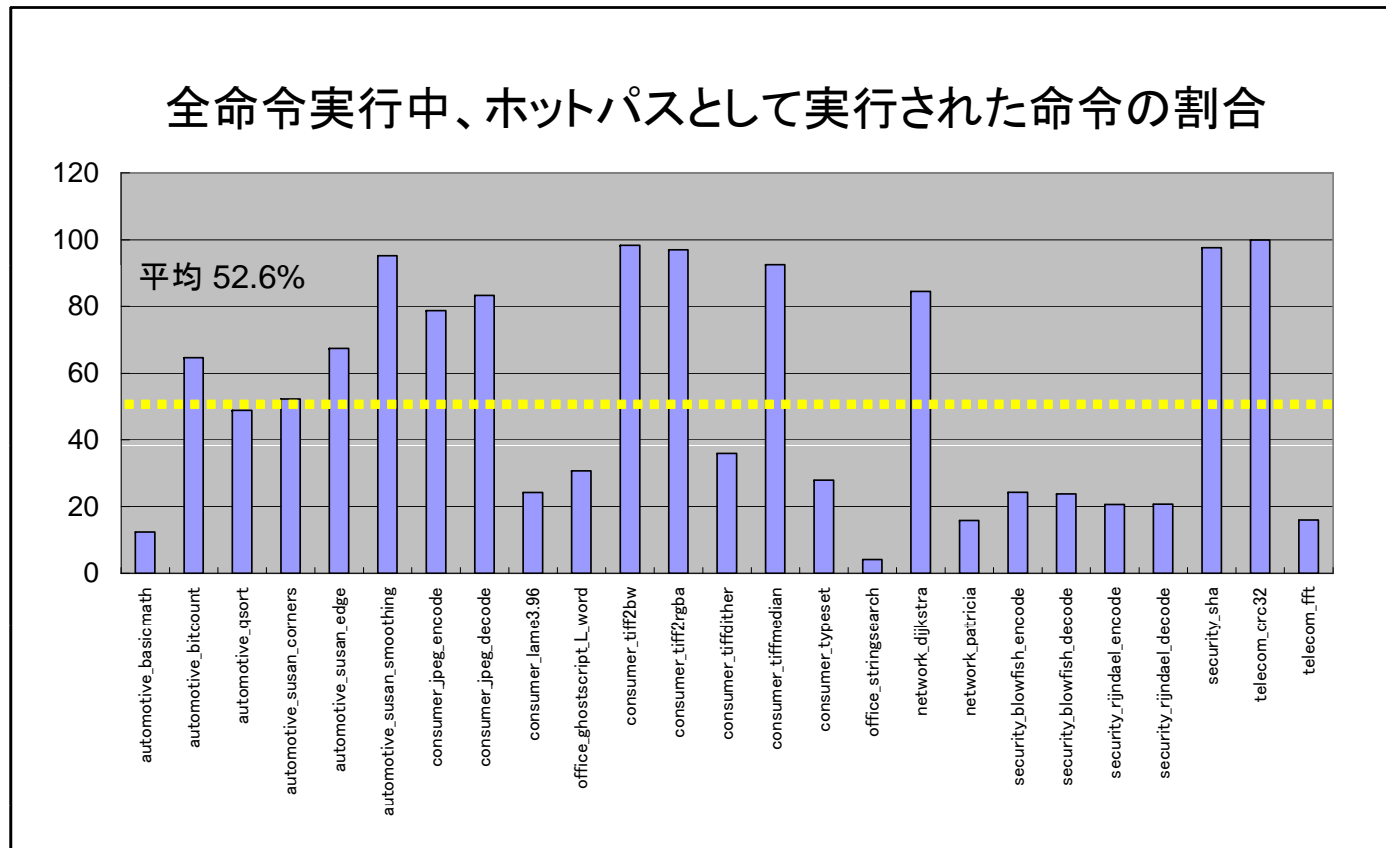
アプリケーションの実行時に適用

- プロファイル適用による性能オーバーヘッド
- ホットパスの推定精度

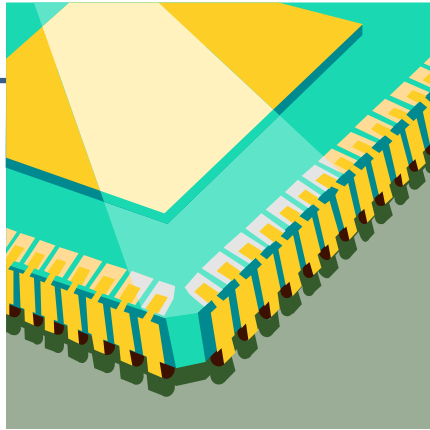


ホットパスに着目した性能最適化

- プログラム実行で、性能クリティカルな箇所(最適化のヒント)を推定して加速実行
- プログラム中の僅かなコード部分のみが最適化の対象



動的ホットパス・アクセラレーションによる 性能最適化



メイン・プロセッサ

アプリケーション
プログラムの実行中

ホットパス・
プロファイリ
ング



最適化の
適用

① 最適化のヒント
の発見
ホットパスの推定

性能
オーバーヘッド

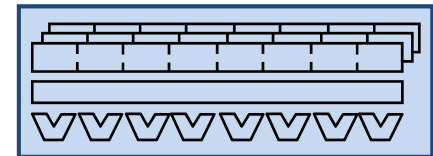
② 最適化の適用
ホットパスの命令列に
対し、コプロセッサ実行
に向けた並列化

③ ホットパス開始アドレスを
コプロセッサの呼出に書き換え

load
load
mul
add
store
br
:

call Acc
nop
nop
nop
nop
br
:
load
load
mul
add
store

コプロセッサによる
ホットパスの実行



コプロセッサ

加速実行

動的ホットパス・アクセラレーションにおける 性能オーバヘッド

実行時の性能最適化

オンラインでのプロファイル、コード最適化による
性能オーバヘッドを小さく

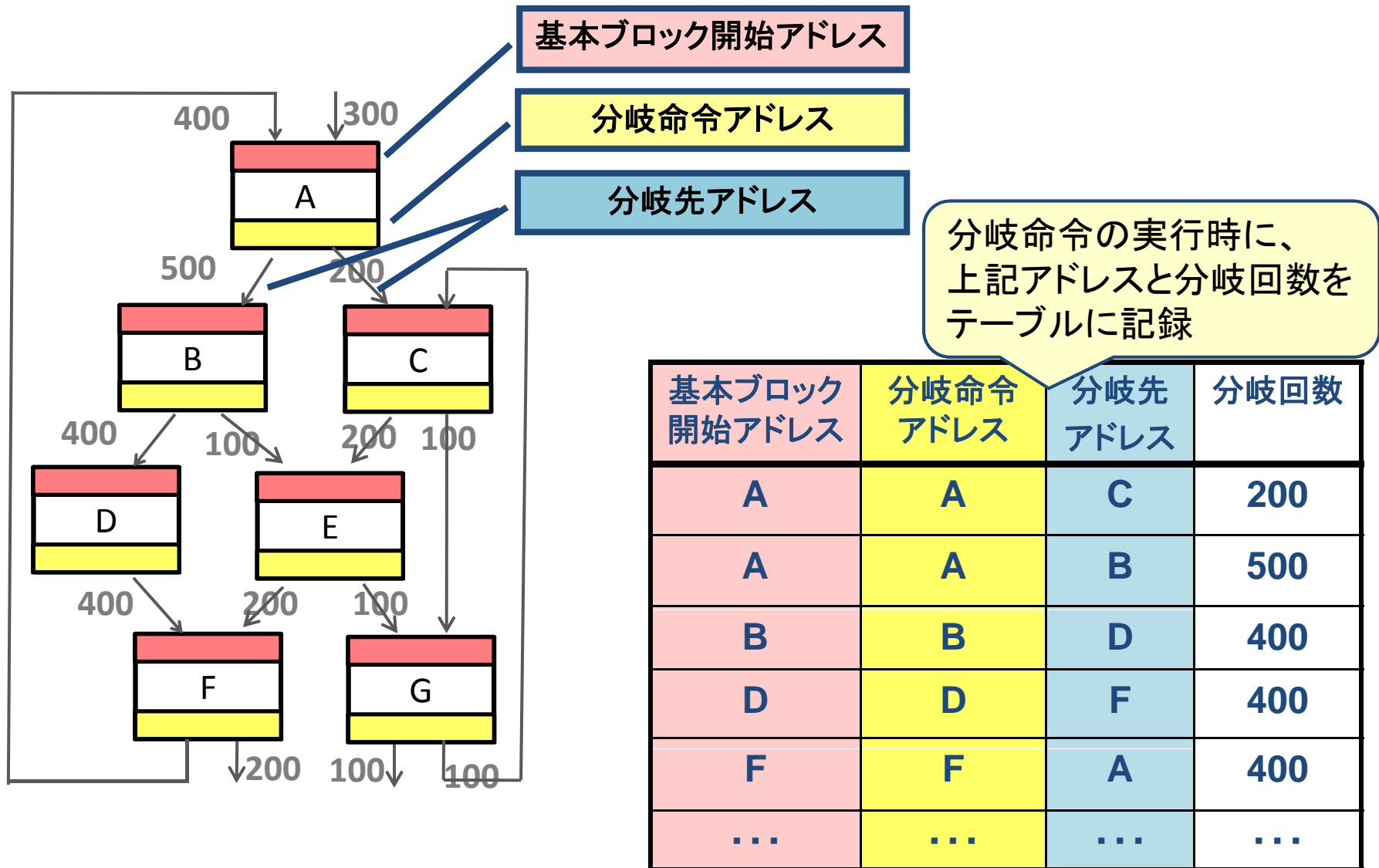
- **ホットパス・プロファイリング**
→ ハードウェアによるホットパスの推定



- **最適化の適用**
→ コプロセッサによる最適化の並列実行

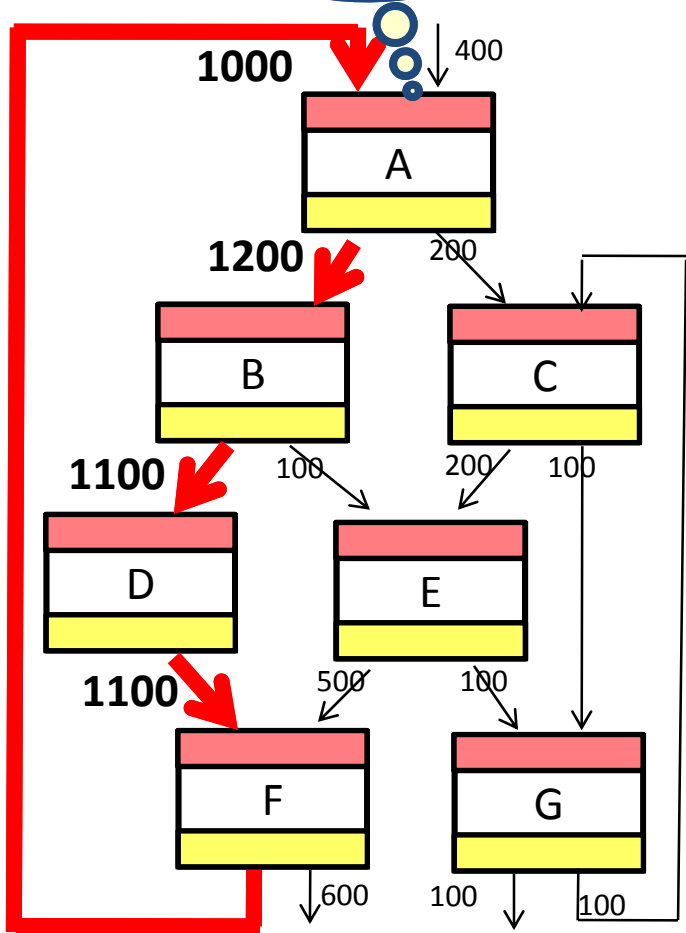


ホットパスの推定方法(1)



ホットパスの推定方法(2)

基本ブロック開始アドレス
の実行回数 > 設定閾値



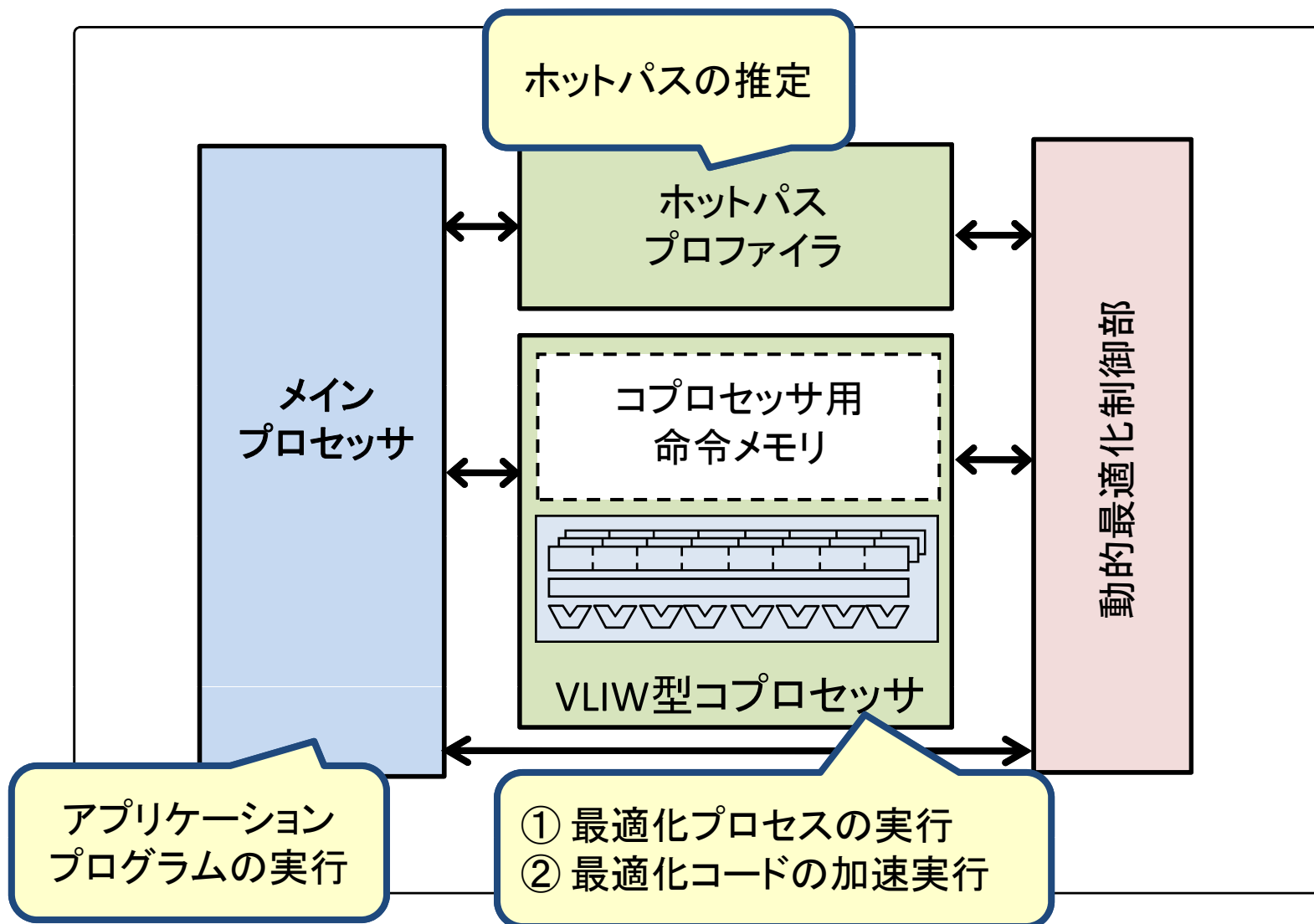
- 分岐履歴を用いた予測精度の向上
 - ホットパス推定処理における容易なテーブル検索
- 簡単な検索用ロジックを備えた
ホットパス推定ハードウェアを設計

先頭アドレスから順にテーブルを検索し、ホットパスを推定

基本ブロック 開始アドレス	分岐命令 アドレス	分岐先 アドレス	分岐回数
A	A	C	200
A	A	B	500
B	B	D	400
D	D	F	400
F	F	A	400
...

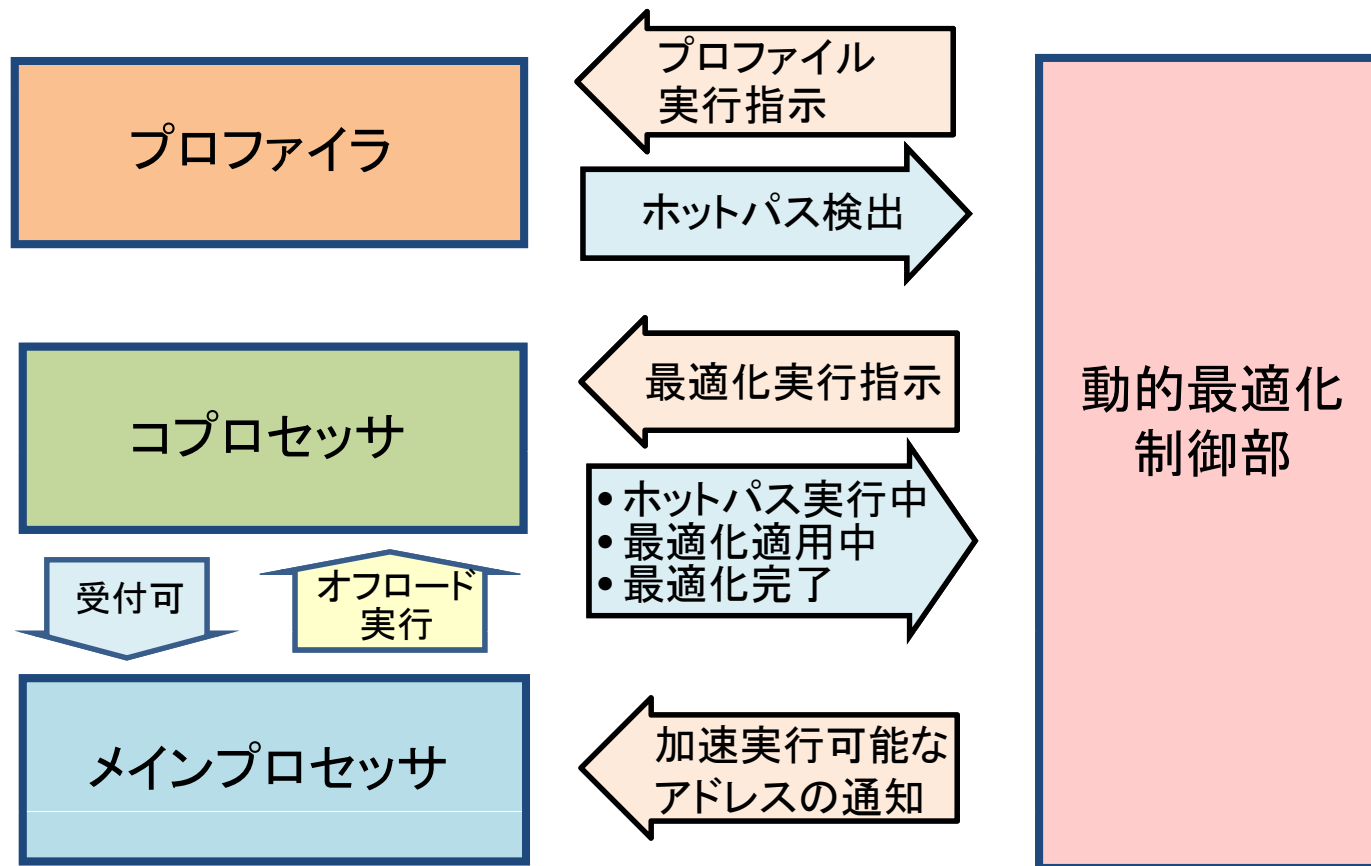
Hot path (ホットパス) is indicated by a vertical label on the right side of the table, pointing to the rows where the branch target is the same as the branch instruction address (A to B, B to D, D to F, F to A).

SystemMorph技術を実装可能なSoC

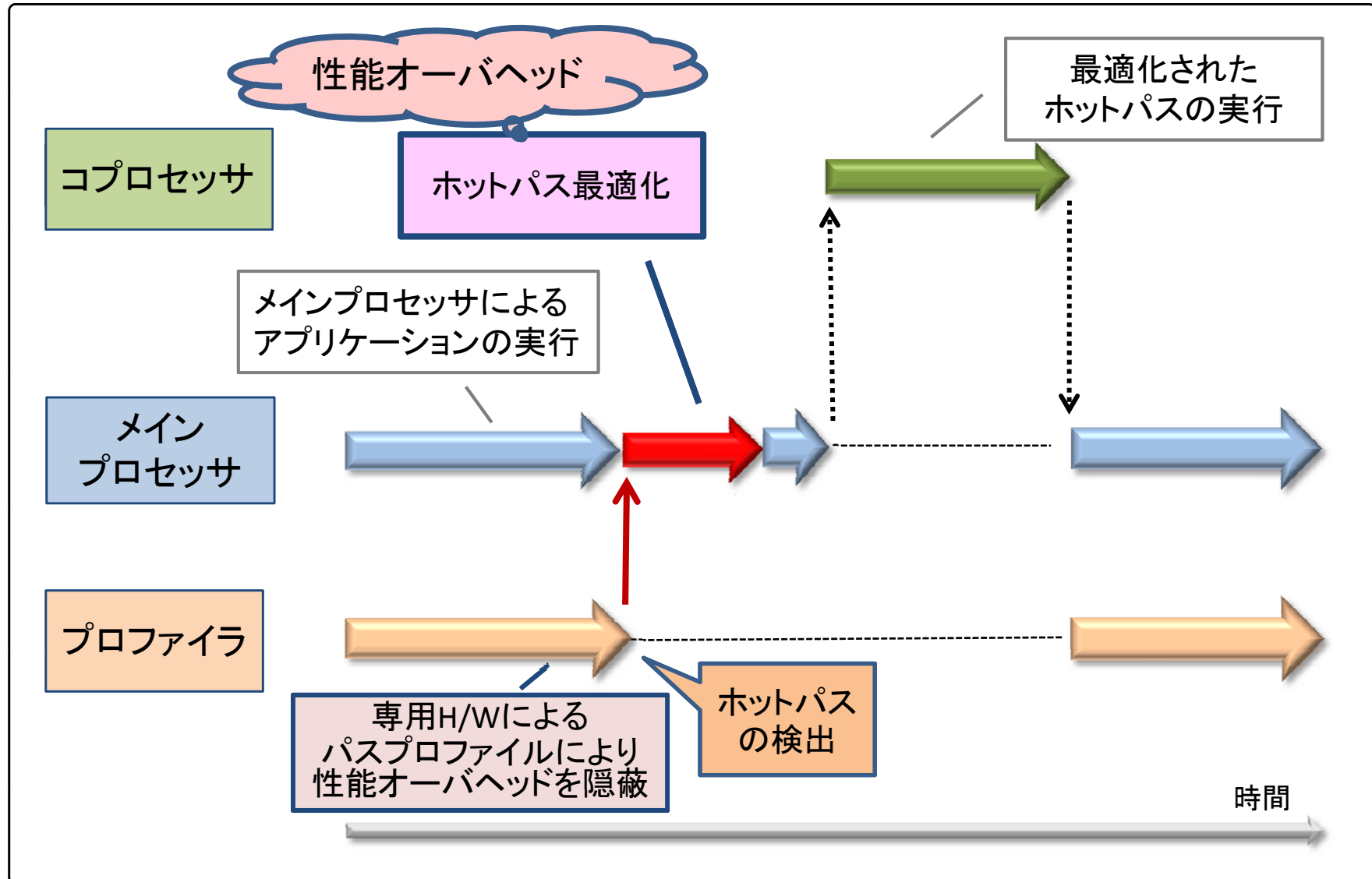


動的最適化の実行制御

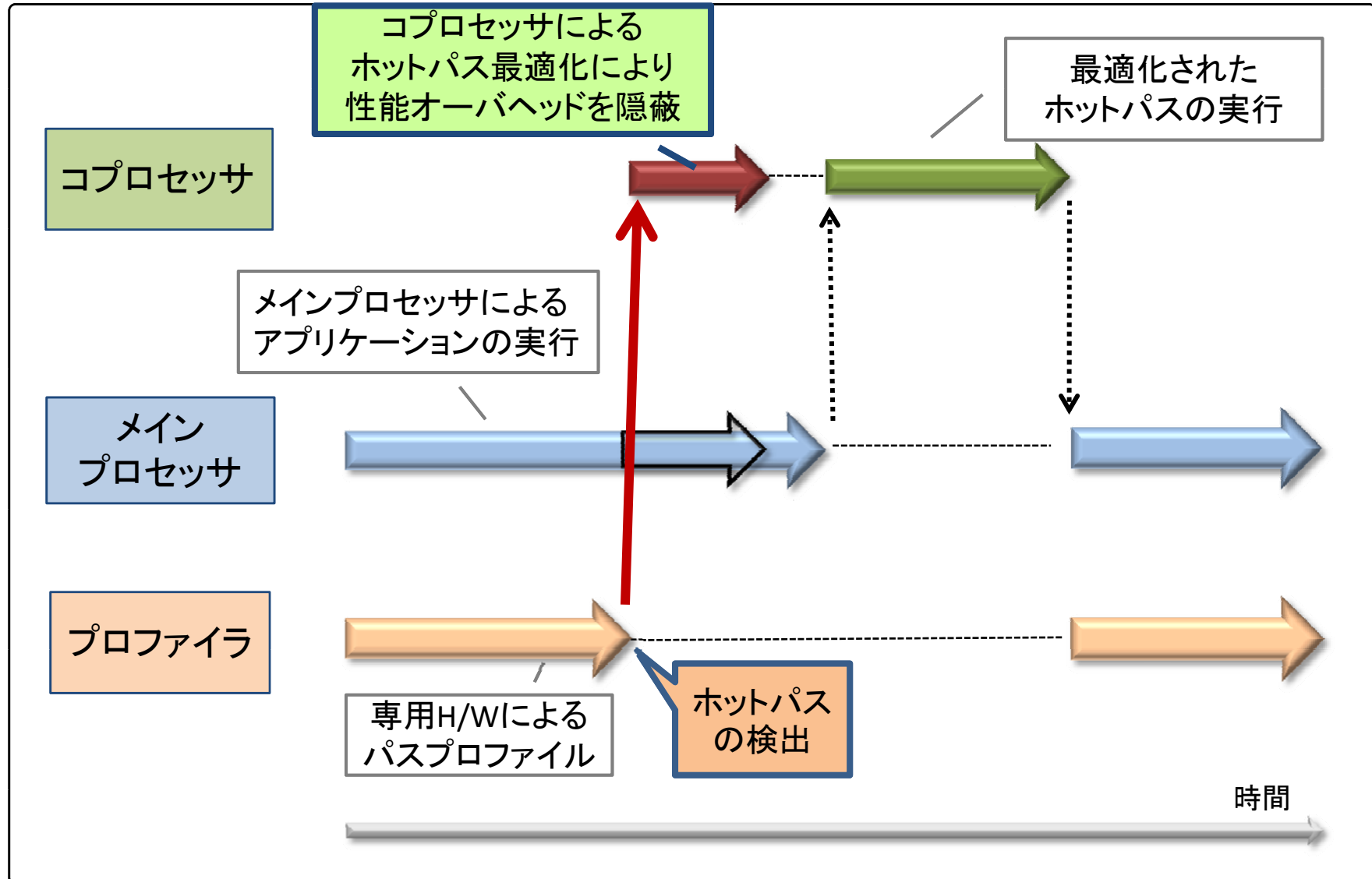
プロファイラ、コプロセッサの状態をモニタしながら、アプリケーション実行中の最適化動作を制御



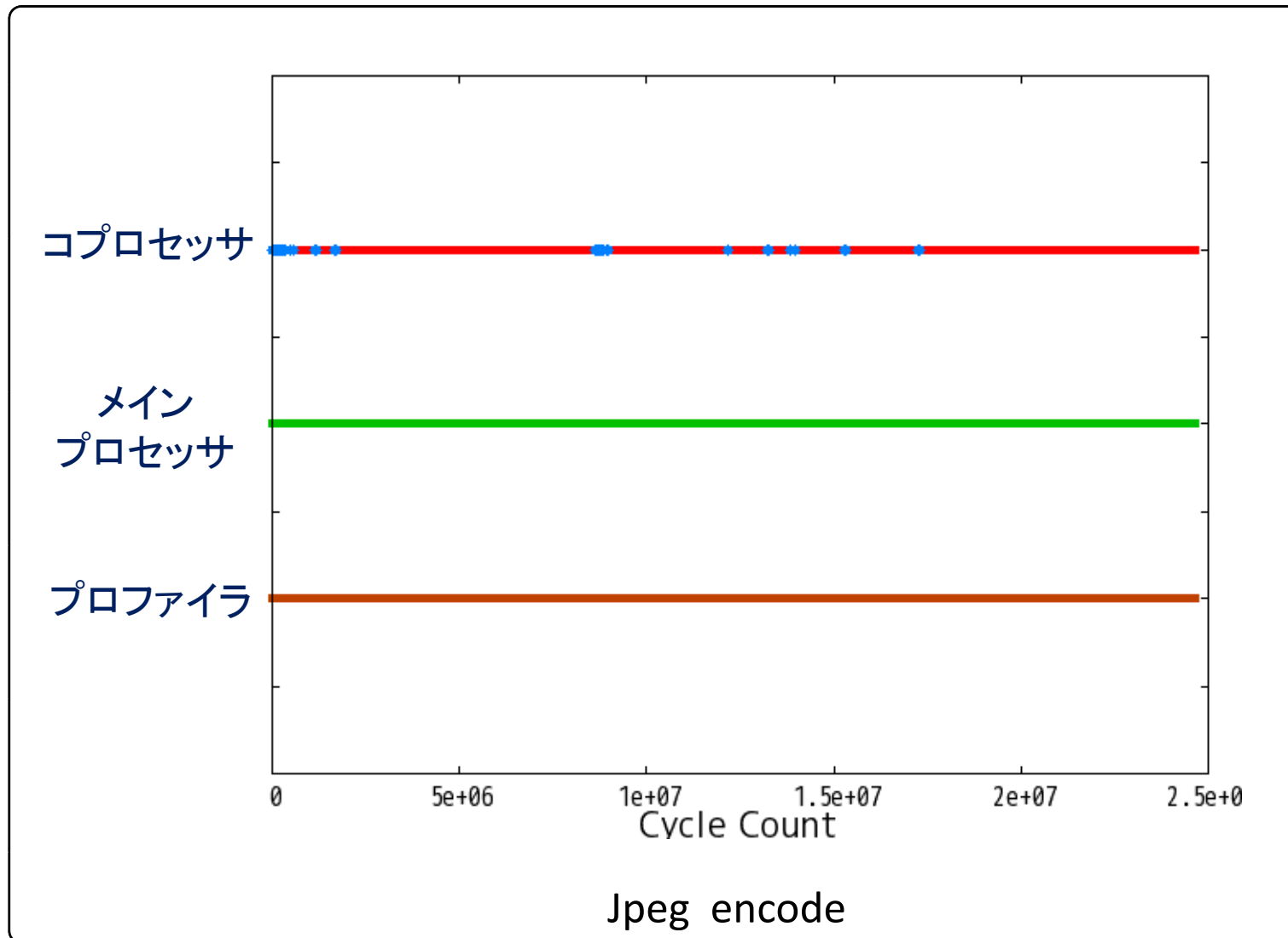
動的最適化の流れ



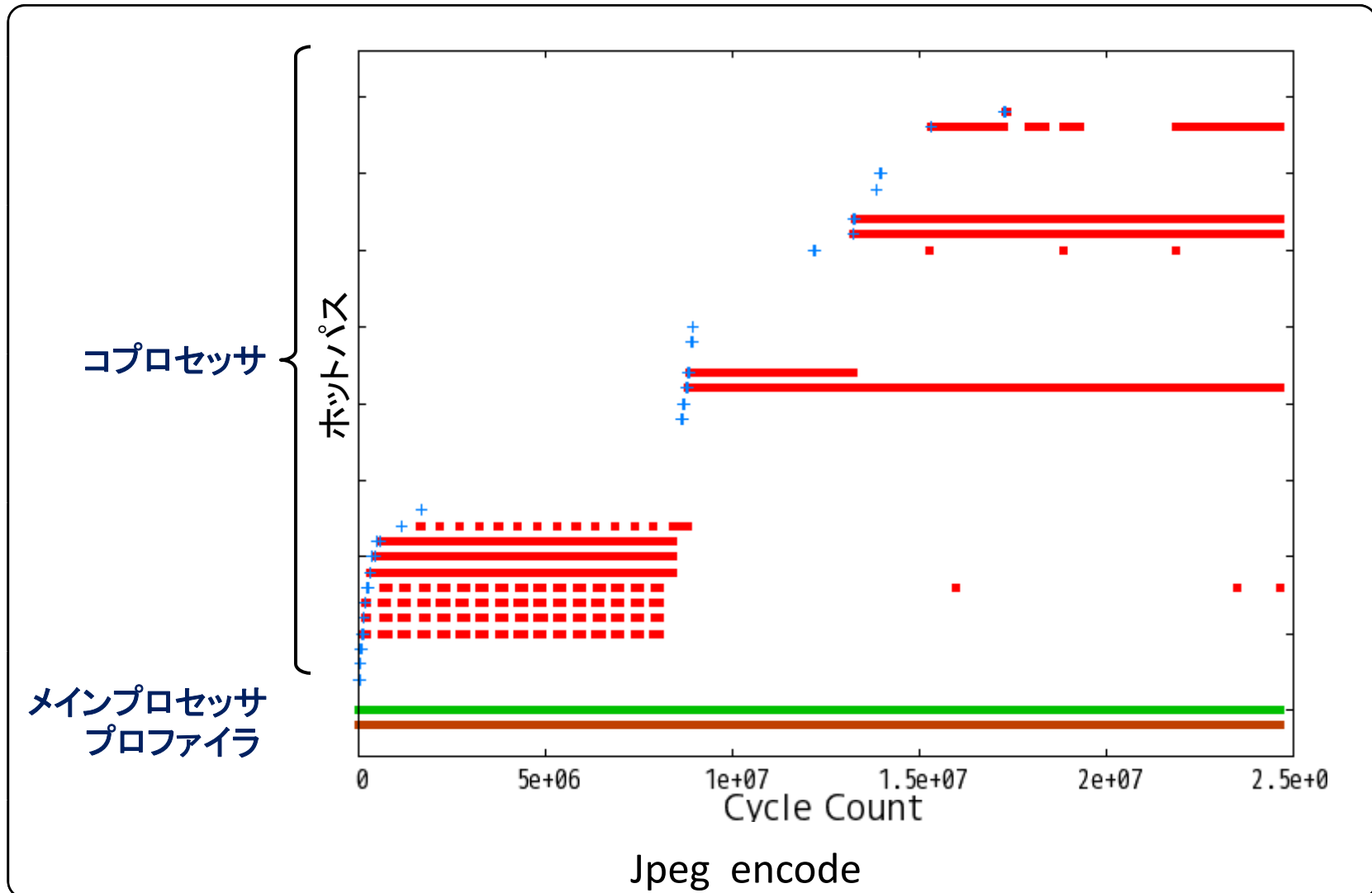
動的最適化の流れ ～コード最適化の並列実行～



ホットパスの検出及び、最適化の適用 ～ シミュレーション ～



ホットパスの検出及び、最適化の適用 ～ シミュレーション (ホットパス別) ～

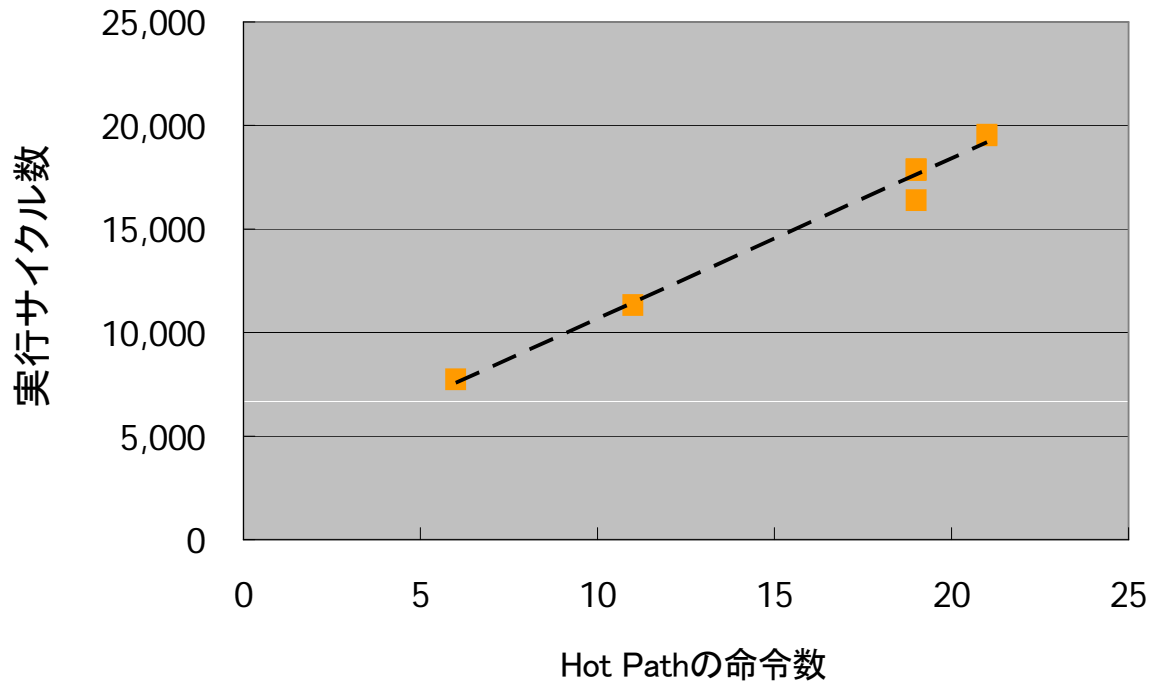


最適化ソフトウェアの実行サイクル

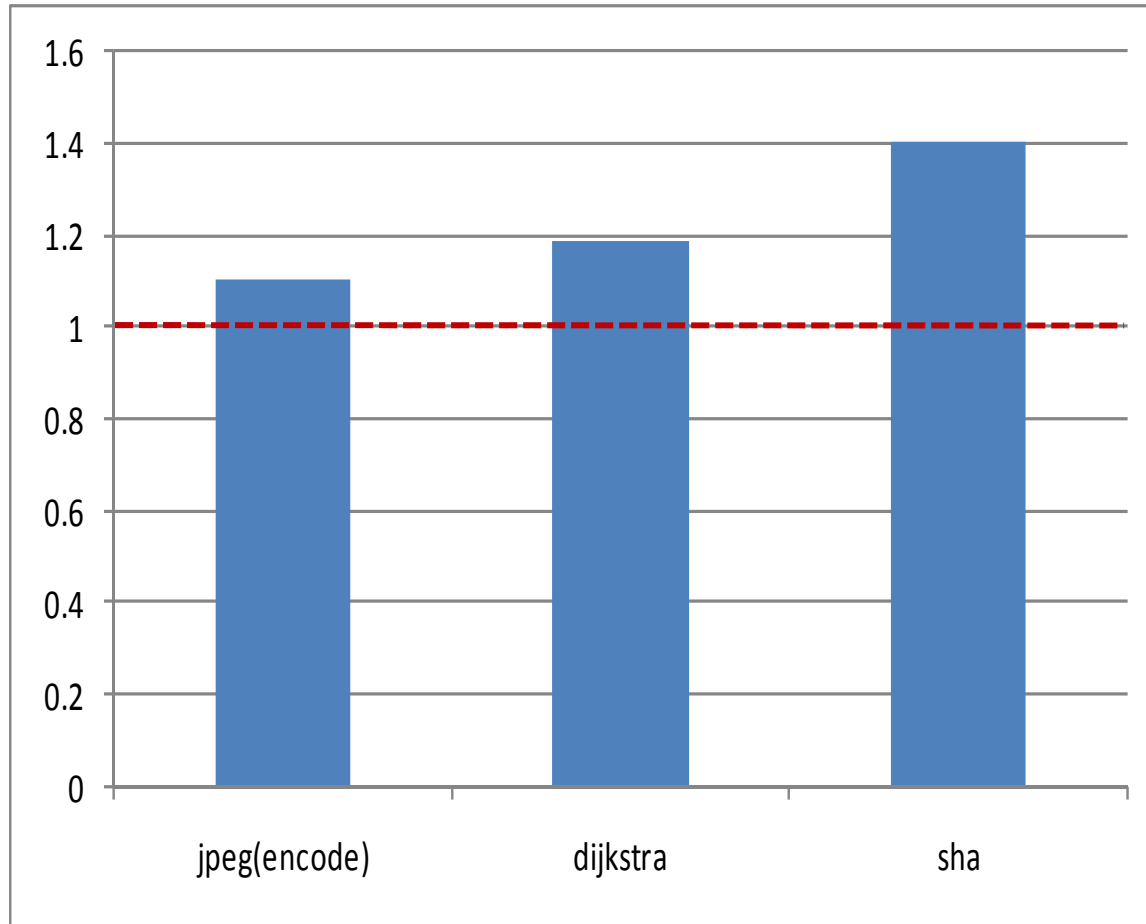
主な処理

- ホットパス部命令列の読み出し
- リストスケジューリングを用いた並列化
- オフロード実行用のコード生成

最適化ソフトウェアの実行サイクル数



最適化適用による性能向上



メインプロセッサ

- 1 issue RISCプロセッサ

コプロセッサ

- 8-way VLIW
(ALUx6, Ld/Stx1, Brx1)

まとめ

- SystemMorph技術によるアプリケーションの実行時の振舞いに応じ、自動最適化の適用が可能なSoCについての評価
 - 実行時の最適化において、性能オーバーヘッドの少ない構成の提案
- 動的ホットパス・アクセラレーションによる性能向上に向けて
 - ホットパスの推定精度の向上
 - コード最適化手法の高度化