モバイル・プラットフォームSoCモデルと設計生産性向上のためのポテンシャル解決策

半導体技術ロードマップ専門委員会
デザイン・ワーキング・グループ（WG1）
04年度・活動報告

WG1:設計　委員構成

- 山本　一郎 (リーダ)
  沖電気工業
- 柏木 治久 (サブリーダ)
  NECエレクトロニクス
- 樋渡 有 (国際対応)
  東芝
- 山田　明宏 (幹事)
  ﾄｯﾊﾟﾝ･ﾃｸﾆｶﾙ･ﾃﾞｻﾞｲﾝｾﾝﾀｰ
- 武田　和彦
  三洋電機
- 豊田　忠雄
  シャープ
- 青木　貫司
  セイコーエプソン
- 柿本　勝
  ソニー
- 古井　芳春
  ソニーセミコンダクタ九州
- 松崎　正己
  富士通
- 隅谷 三喜夫
  松下電器産業
- 塩月 八宏
  ルネサステクノロジ
- 浅井　健史
  ローム
- 小野　信任
  ジーダット･イノベーション
- 今井　正治
  大阪大学
- 小澤　時典
  STARC

目次

- 国際活動･国内活動の概要
- モバイル･プラットフォームSoCモデル
- 設計生産性要求値
- 設計生産性向上のための解決策検討
- まとめ、今後の計画

国際活動　WG1の役割と今年度注力活動

- WG1の役割：　ITRS Design章 + System Drivers章
- 今年度注力活動：Mobile Platform SoC Modelをロードマップ検討の基盤として定義

- Design章
  - 設計技術に対する将来課題とポテンシャル解決策の提示
  - 技術領域：
    ① Design process,　② System-level design, ③ Logical/physical/circuit design
    ④ Design verification, ⑤ Design test,　⑥Design for Manufacturing

- System Drivers章
  - 製造技術および設計技術をドライブするLSI商品を定義
  - ORTC + System Drivers = ITRSの技術要求フレームワーク

•ORTC =Overall Roadmap Technology Characteristic

- 今年度活動：
  - Mobile Platform SoC Modelをロードマップ検討の基盤として作成
    ⇒　05年版ITRSに掲載予定

国内活動　WG1の役割と今年度注力活動

- WG1の役割：
  - 設計技術課題を、時間軸をもって定量評価し、解決策を提案
  - SoCの構造・規模を、時間軸をもって定量化し、
    設計技術および、製造技術のロードマップ検討の基礎として提示

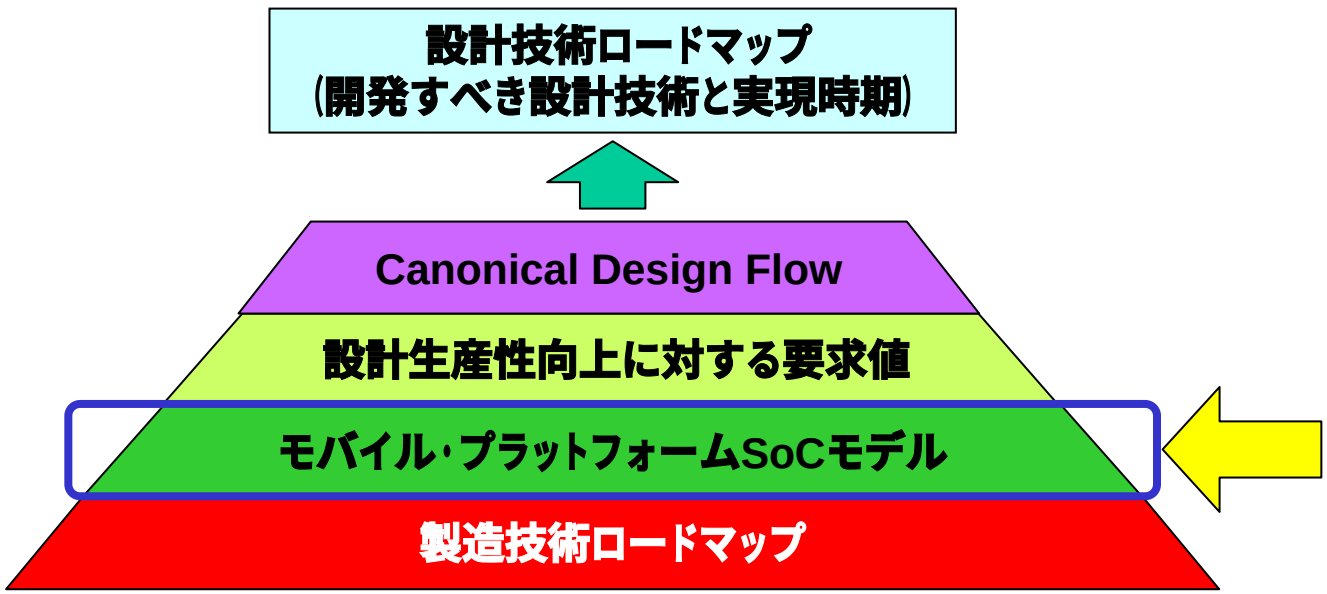
- 今年度注力活動
  - モバイル・プラットフォームSoCモデルの開発
    - 対象：　パーソナル・モバイル機器の中核SoC
    - 成果物：モバイル・プラットフォームSoCの構造・規模を数値表現した将来像
    - 目的：　ロードマップ検討の基礎
  - 設計生産性ロードマップの検討
    - モバイル・プラットフォームSoCにおける設計工数を定量分析
    - 設計生産性向上に向けた課題抽出とポテンシャル解決策検討着手

目次

- 国際化活動・国内活動の概要
- モバイル・プラットフォームSoCモデル
- 設計生産性要求値
- 設計生産性向上のための解決策検討
- まとめ、今後の計画

作成目的

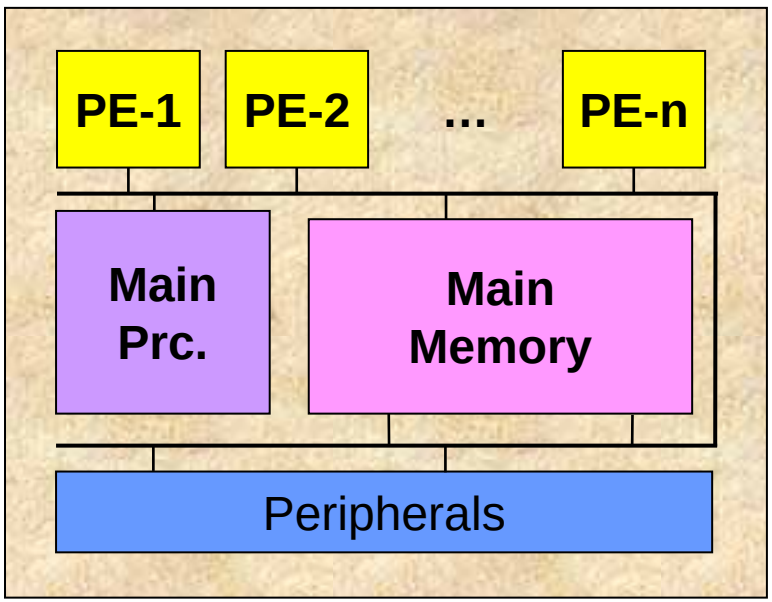
- SoCモデルとは、
  将来のSoC構造・規模を定量分析可能な形式で予測したモデル
- 作成目的
  - 製造技術ロードマップから設計規模を算出し、設計困難度を定量的に把握
    ⇒　設計技術の課題を明確化するとともに、ロードマップ検討の精度を向上
  - 日米欧間の設計技術ロードマップ検討の共通基盤
    ⇒　検討基盤の共有化による国際ロードマップ検討の加速



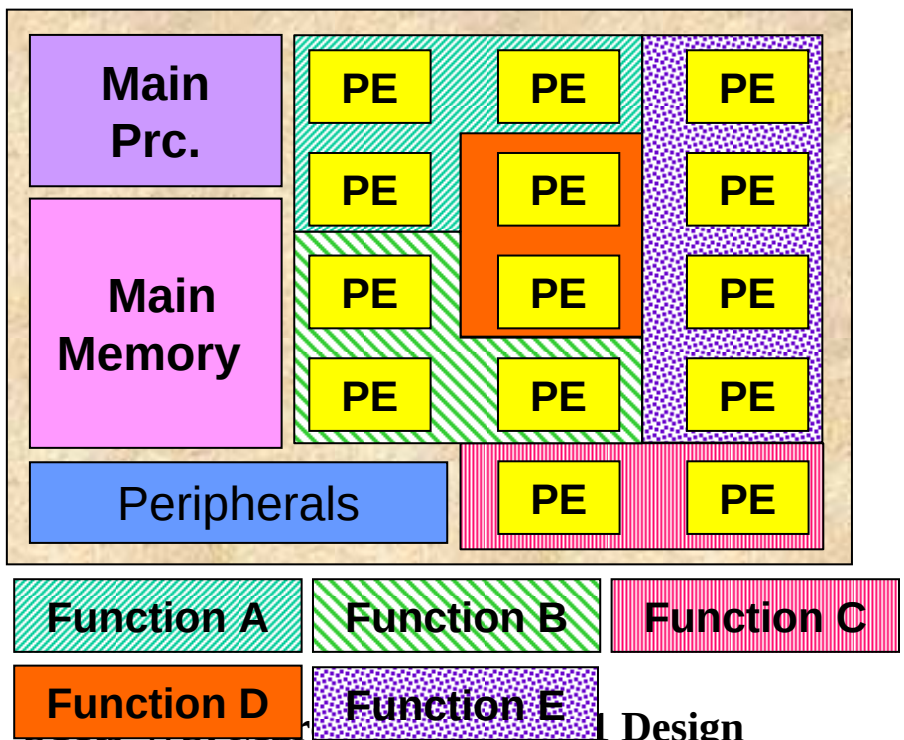
概要

- **対象：パーソナル・モバイル機器の中核**SoC
- **特徴：特定用途向け**Processing Engine (PE) **を複数搭載**
- **構造：** Main Processor + PE + Peripherals + Main Memory
  - Processing Engine (PE)：**特定機能にカスタマイズしたプロセッサ**
    規模の大きい機能は、複数のPE**で実装。単一**PE**のアレイ構造ではない。**

Mobile Platform SoC (構造イメージ)



Mobile Platform SoC (機能イメージ)



構造

- 構造（下記）および数値パラメータを明確に定義し、Microsoft社製Excelに実装

- 基礎パラメータ
  - トランジスタ密度：
    ITRS/ORTC値を使用
  - デバイス性能：
    ITRS/PIDS値を使用
    Low Operating Power Logic

- 面積
  - Die Size：　一定
    - 製造コストを一定とするため
  - 面積オーバーヘッド：　一定
    - オーバヘッド：
      I/O、アナログ、電源系

- Main Processor
  - 全体制御用、汎用プロセッサ
  - 回路規模：　一定
  - 搭載個数：　1個（一定）

- Processing Engine（PE）
  - 特定機能にカスタマイズしたプロセッサ
  - 回路規模：　一定
    - 配線遅延増加の影響を避けるため
  - 搭載個数：
    面積条件が許す最大個数搭載
  - 動作周波数：
    デバイス性能に比例

- Main Memory
  - 容量：　PE数に比例

- Peripherals
  - SoC外とのインタフェース用回路
    - I/O回路はオーバヘッドに含む
  - 回路規模：　一定

数値パラメータ

- 数値パラメータ（下記）は、日米欧のDesign WG間で整合・確認中

Die Size:	49mm2	（一定）
Area Overhead：	28%	（一定）
Main Processor：	1MG Logic	（一定）
	512k bit Memory	（一定）
Processing Engine：	250kG Logic	（一定）
	64k bit Memory	（一定）
Main Memory:	PEあたり、1M bit	（一定）
Peripherals:	1MG Logic	（一定）

導出される未来像

- PE数
  - PEの搭載個数は、面積条件が許す最大個数搭載

$$(\text{PE数}) = \frac{\text{チップ面積} - \text{オーバーヘッド面積} - \text{主プロセッサ面積} - \text{周辺回路面積}}{\text{PE面積} + \text{PEあたりのメモリ面積}}$$

  - 微細化と共に増加：　7個（2004年）　⇒　174個（2016年）
    年率：1.31倍

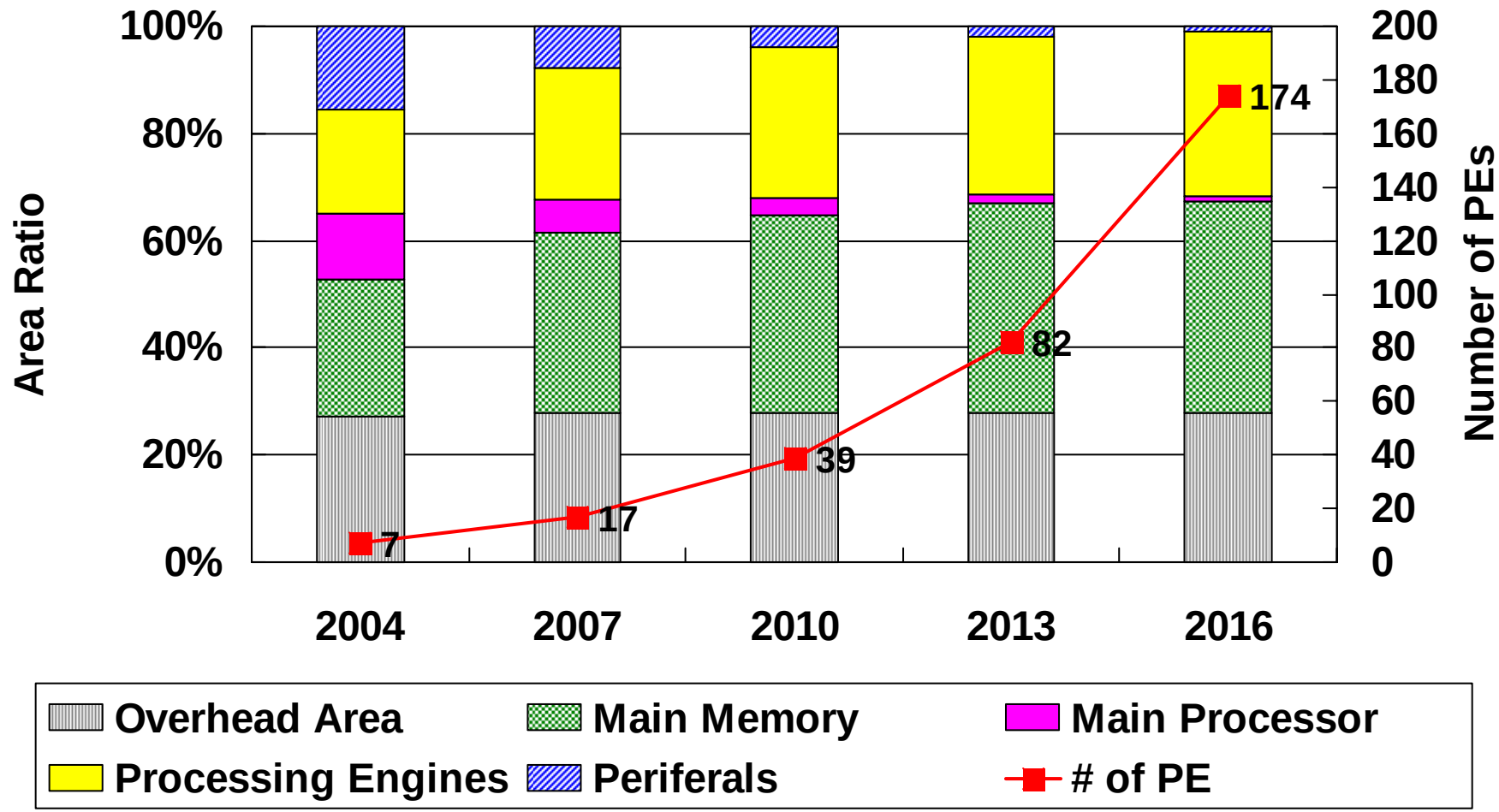
- 総演算能力
  - 全PEを同時に動作させた場合における演算能力

$$(\text{最大演算能力}) \propto (\text{デバイス性能}) * (\text{PE数})$$

  - PE数およびデバイス性能向上に伴い増加
    2004年⇒2016年：168倍、　年率1.53倍
  - 消費電力問題を考えると全PEの同時実行は困難
    ⇒　SoC性能を把握するための1つの指標

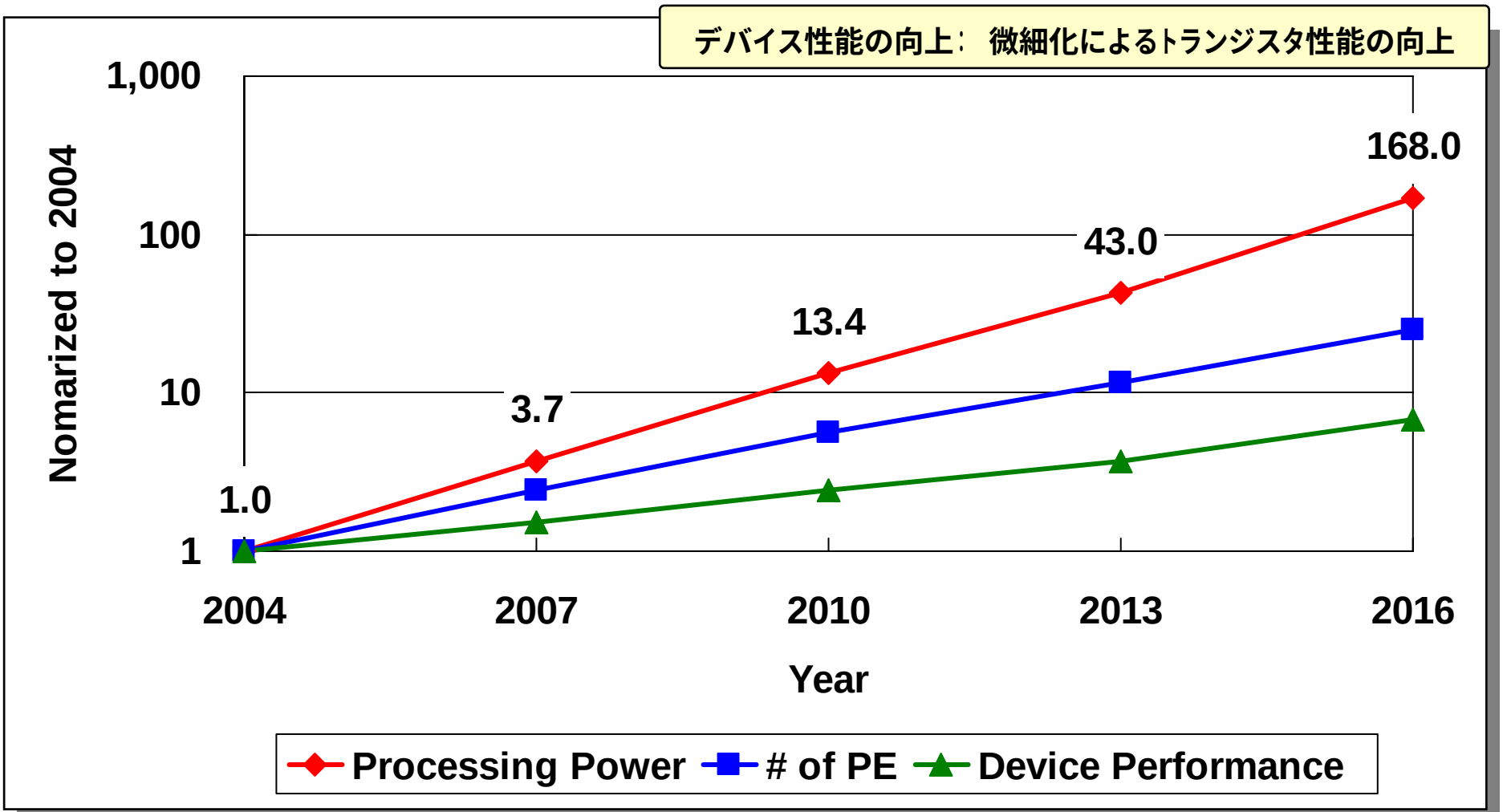
導出される未来像： 面積比率

- 搭載可能PE数は、年率1.31倍で増加
  微細化によるPE面積減および、PE領域の増大の相乗効果



導出される未来像： 総演算能力

- **総演算能力は、年率1.53倍で増加**
  デバイス性能向上および、搭載PE数増加の相乗効果

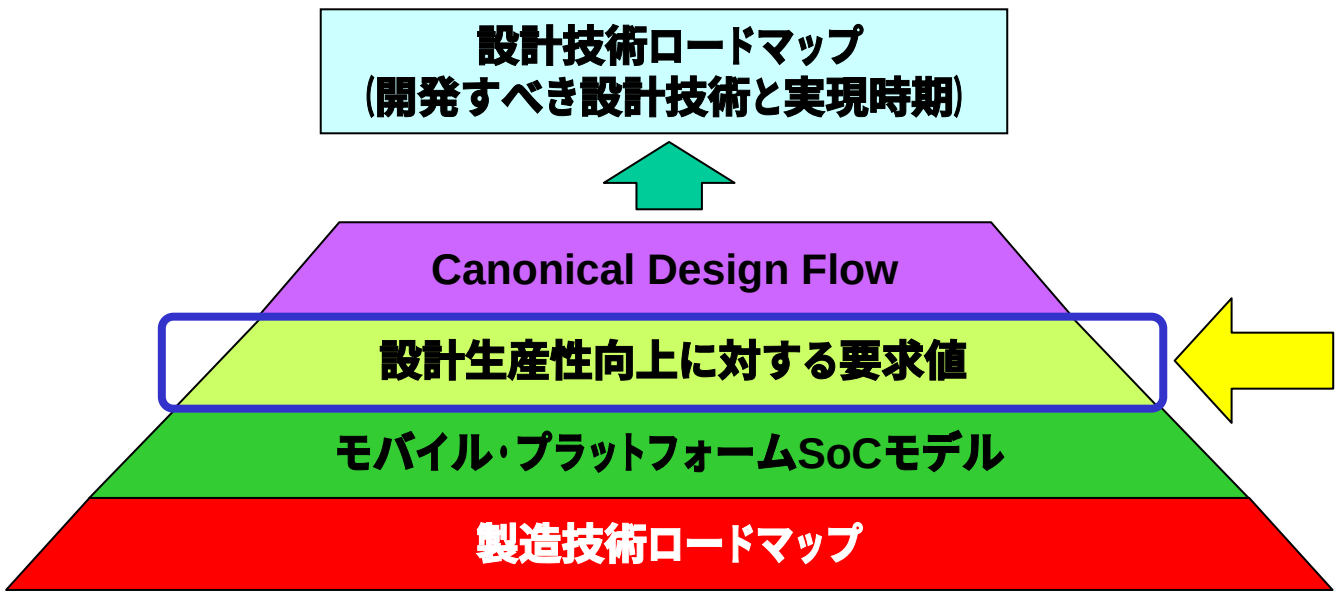


目次

- 国際化活動･国内活動の概要
- モバイル･プラットフォームSoCモデル
- **設計生産性要求値**
- 設計生産性向上のための解決策検討
- まとめ、今後の計画

検討対象、方法、最終目標

- 検討対象：　モバイル・プラットフォームSoCモデル
- 検討方法：　「SoC開発におけるHW設計工数 = 一定」の実現を目標とし
  要求される設計生産性向上率＋IP再利用率を検討
- 最終目標：　設計生産性要求値を設計フローにマッピングし、
  設計技術ロードマップを定量的に分析

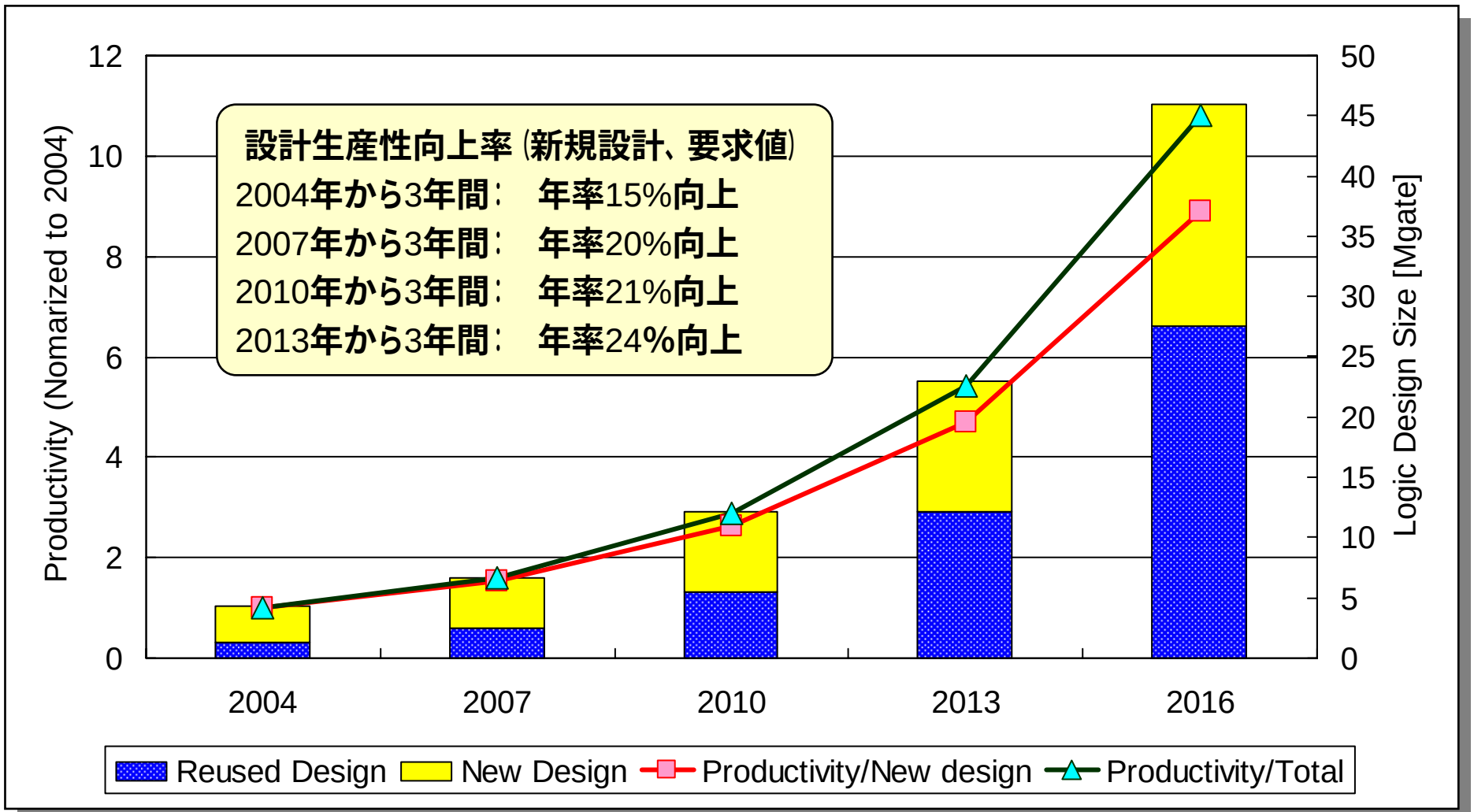


前提・仮定

- SoC開発工数 = 一定
  - SoC開発工数：　HW開発工数が対象
    - HW/SW協調検証、SW開発の工数は対象外
  - 一定：　テクノロジ・ノードに依存せず一定
- 新規設計に要する工数
  - 論理回路規模に比例
    - 前提：　メモリ、アナログ回路などは、IPとして提供
- 設計資産（IP）再利用に必要な工数
  - 同一規模回路の新規設計に要する工数の50%
    - テクノロジ・ポーティング（Soft IPの場合）、検証、評価などの工数
- IP再利用
  - 20%（2004年）　⇒　60%（2016年）　リニアに増加

「SoC開発工数=一定」のための設計生産性向上率

- 2004年から12年間：　8.9倍（新規設計）、11倍（全体：設計再利用効果を含む）
- Technology Nodeに伴い、設計生産性の向上要求が上昇　⇒　IP再利用効果が減少

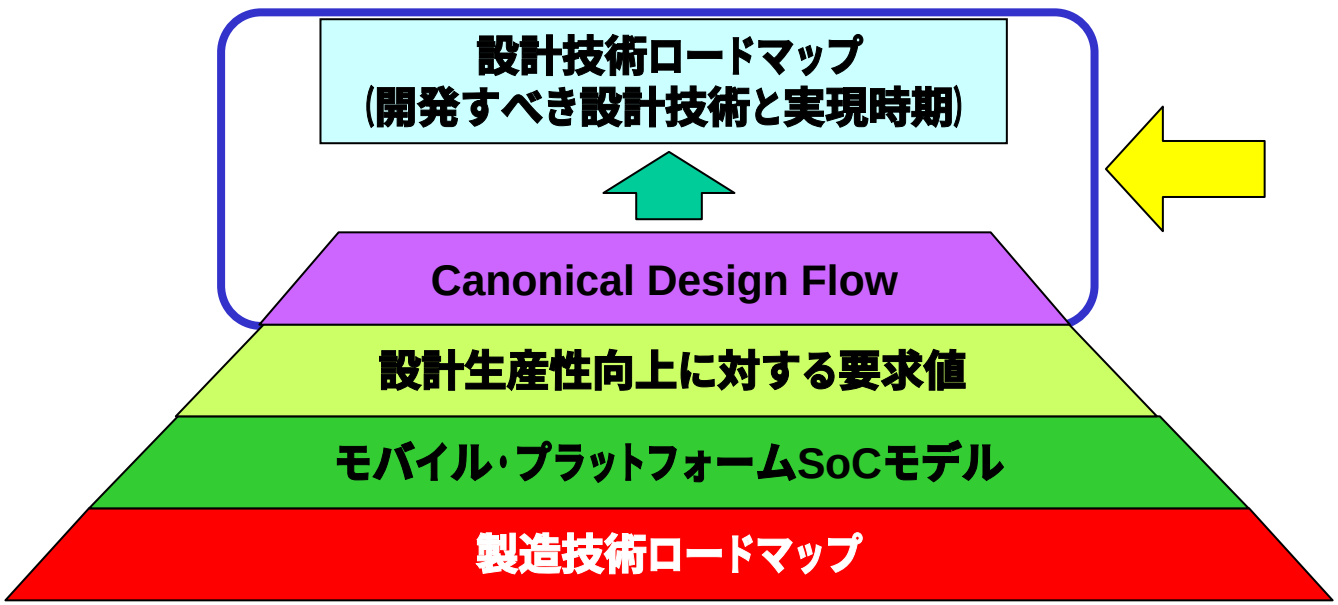


目次

- 国際化活動･国内活動の概要
- モバイル･プラットフォームSoCモデル
- 設計生産性要求値
- 設計生産性向上のための解決策検討
- まとめ、今後の計画

検討対象、方法、最終目標

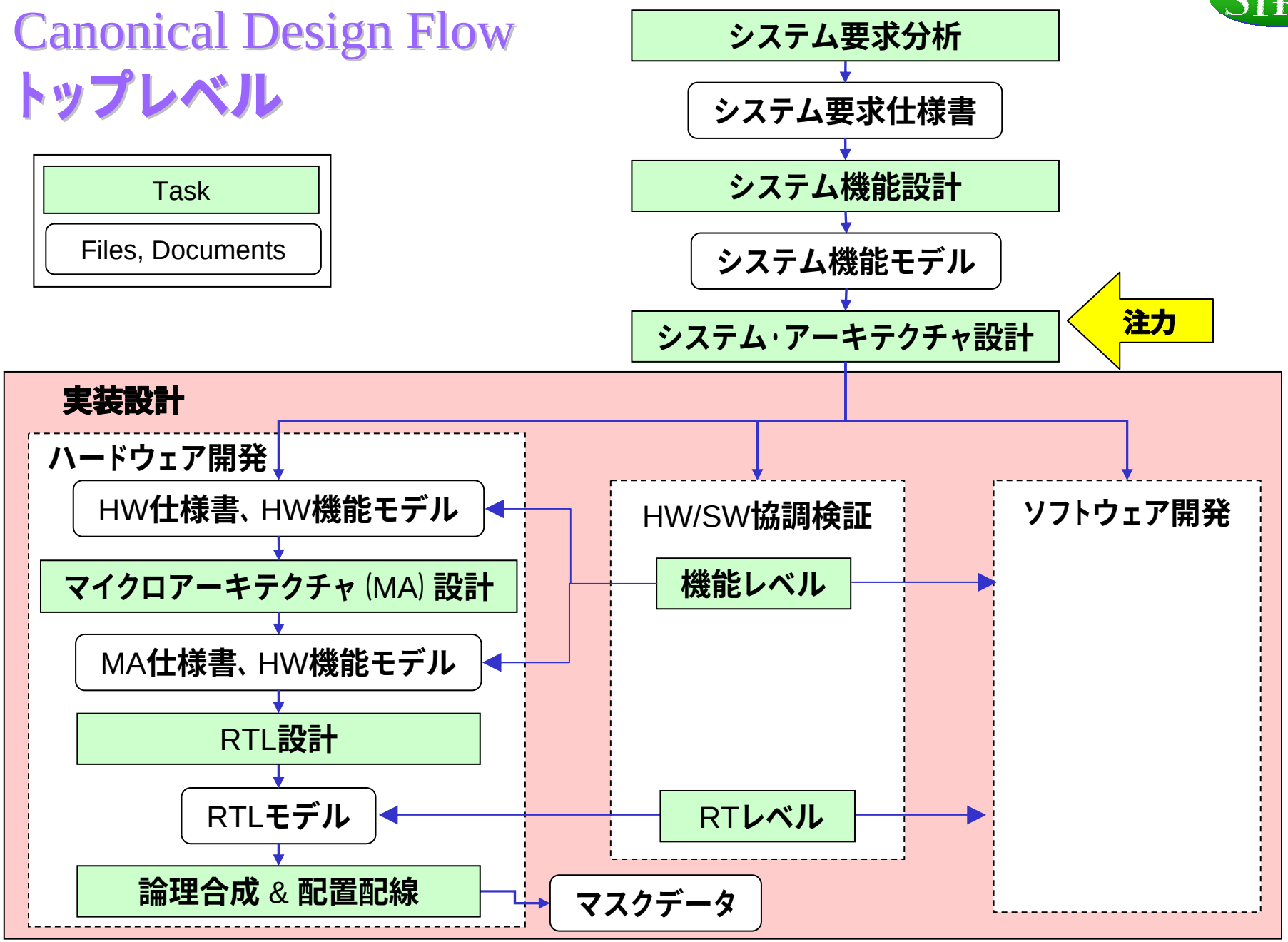
- 検討対象：　システム・アーキテクチャ設計
- 検討方法：　Canonical Design Flowの詳細化（サブ工程への分割）
  各サブ工程の生産性向上のための解決策を検討・定義
- 最終目標：　設計生産性要求値に基づく、設計技術ロードマップの定量分析
  ⇒　今年度：予備的な検討まで完了



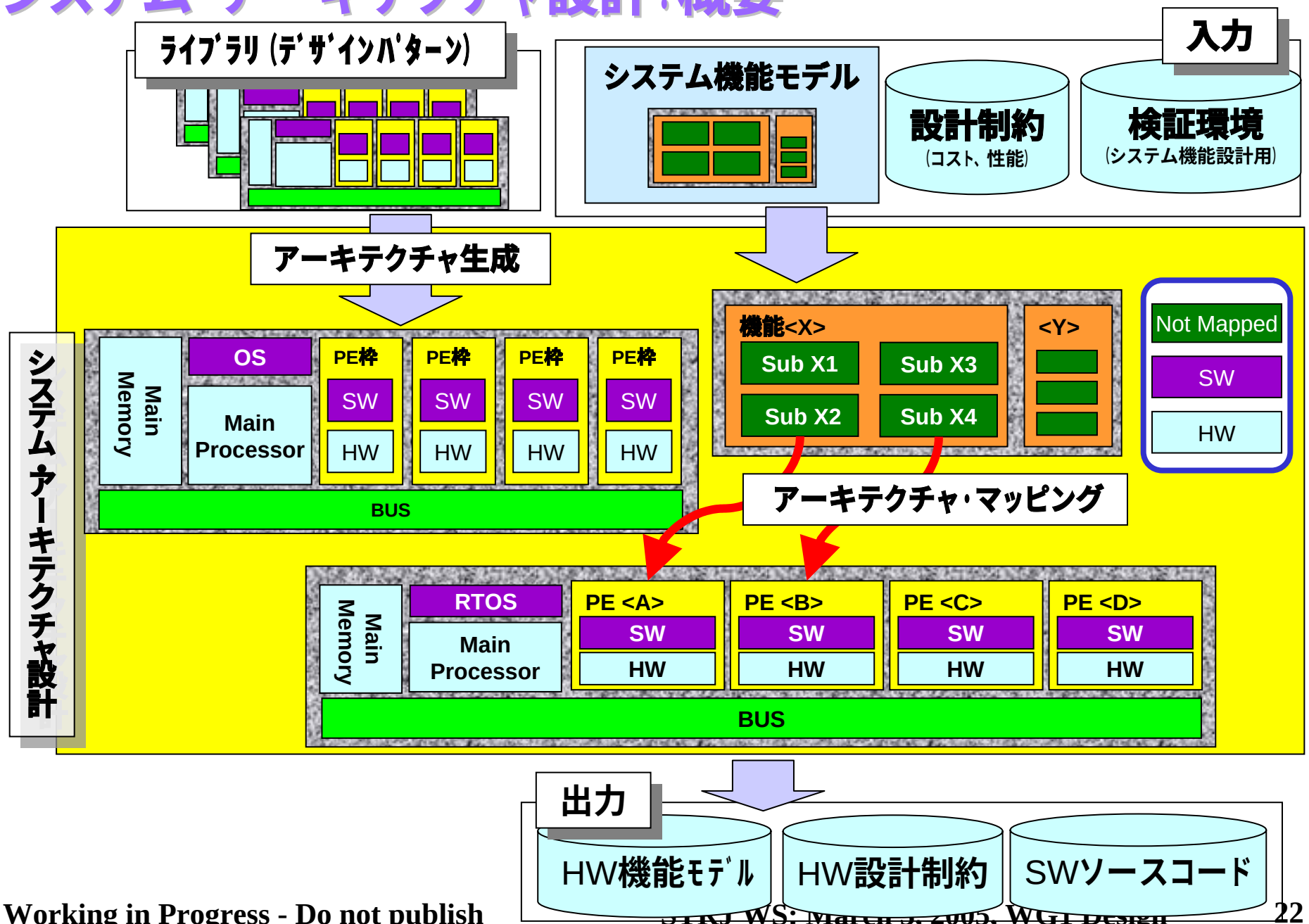
Canonical Design Flow

- 内容
  - ロードマップ検討のために、
    現在及び将来のSoC設計フローを定義･明確化した図及び解説
  - 一般的に理解されている設計フローを、図および文書で明確化
- 目的
  - 設計技術ロードマップの基本背景の提示
    - 設計技術に関する議論の共有化／伝達を容易化
  - 設計コストに関する議論の基盤を提供
- 開発状況
  - トップ･レベル･フロー
    - 日本が作成･提案し、ITRSに2003年版から掲載
  - 04年度検討内容
    - 設計生産性向上策検討の観点から、中間レベル･フローの詳細化に着手

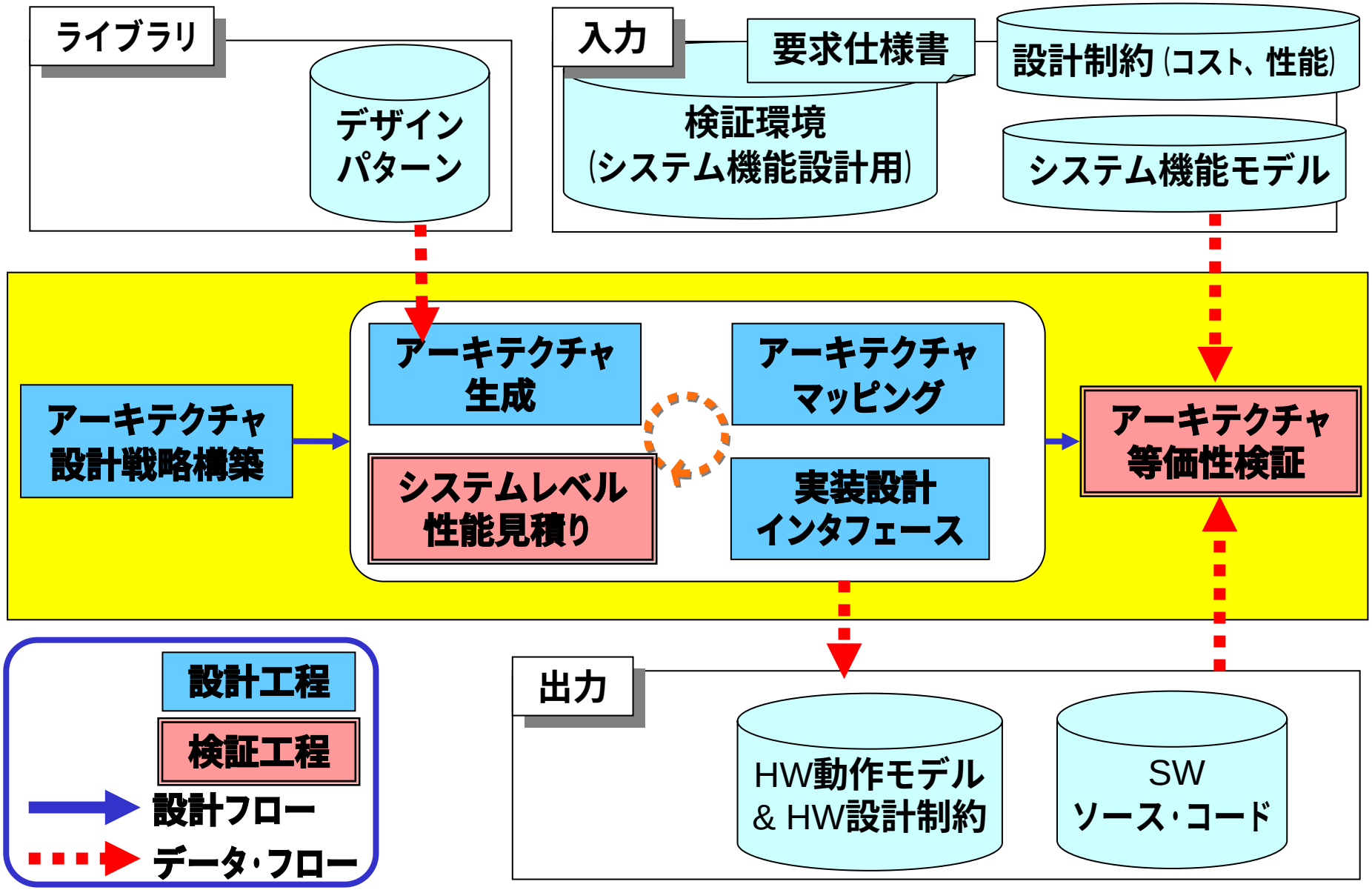
Canonical Design Flow
トップレベル



システム・アーキテクチャ設計：概要



システム・アーキテクチャ設計フロー：中間レベル



サブ工程の定義

- アーキテクチャ設計戦略構築
  - システムレベル性能見積りにおいて、
    評価すべき項目と、評価を実施するための手段（シュミレーション環境）を定義
- アーキテクチャ生成
  - デザイン・パターンから目的に合致するアーキテクチャを選定
- アーキテクチャ・マッピング
  - 機能を分割し、実現手段（HW/SW）を判断し、PEに割り付け
- システムレベル性能見積り
  - 選定した「アーキテクチャ＋マッピング」が、目標性能を満たすかを判定
- 実装設計インタフェース
  - 選定した「アーキテクチャ＋マッピング」を、後工程（実装設計）用に変換
- アーキテクチャ等価性検証
  - 設計結果（実装設計インタフェースのアウトプット）の
    システム機能モデルに対する等価性を検証

設計工程における解決策

- 設計工程に対して、
  アウトプットを定義し、設計生産性向上のための解決策を検討･定義

工程	アウトプット	解決策
アーキテクチャ 設計戦略構築	■ システムレベル性能評価環境 設計結果の性能を見積もる (プロファイリングする) ためのテスト環境	■ 統合検証技術
アーキテクチャ 生成	■ システム･アーキテクチャ アーキテクチャ･ライブラリからの選択結果	＜次年度具体化予定＞
アーキテクチャ マッピング	■ HW/SW分割割り付け結果 機能を分割し、HW/SWとして PEにマッピングした結果	＜次年度具体化予定＞
実装設計 インタフェース	■ HW動作モデル＆HW設計制約 ■ SWソースコード	＜次年度具体化予定＞

検証工程における解決策

- 検証工程に対して、
  検証詳細度を定義し、設計生産性向上のための解決策を検討･定義

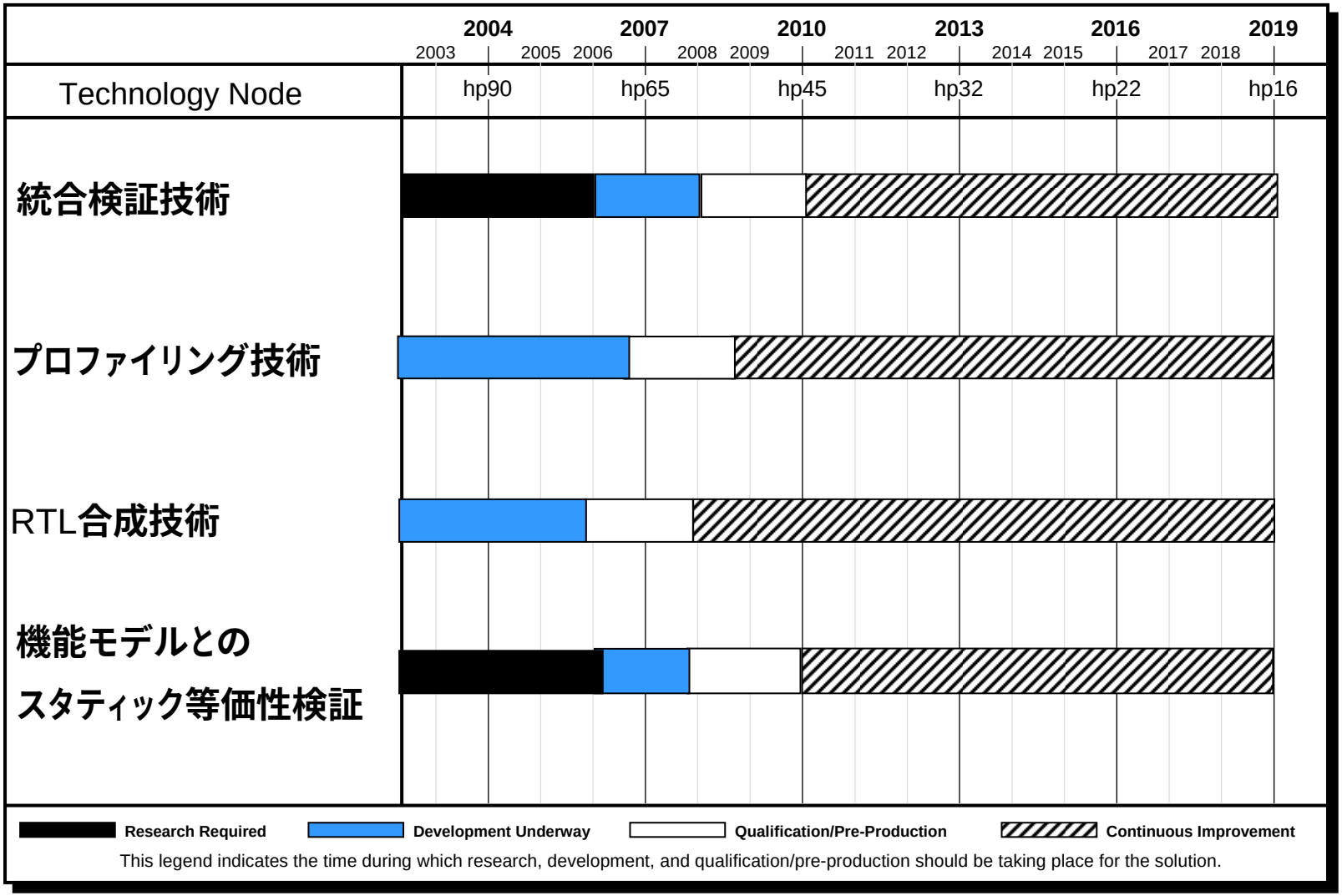
工程	検証詳細度	解決策
システムレベル 性能見積り	■ 性能見積り • シミュレーションに基づく プロファイリングの実施 • シミュレーション量は、 要求見積り精度に合わせて限定 ■ 面積見積り • IPデータ利用、RTL合成の実施 ■ 電力見積り：基礎的な研究が必要	■ プロファイリング技術 ■ RTL合成技術 ■ 電力見積りに関しては、 基礎的な研究が必要
アーキテクチャ 等価性検証	■ 入出力モデル間の等価性 • 入力：機能モデル • 出力：HW動作モデル ＋ SWソースコード	■ 機能モデルとのスタティック 等価性検証 • シミュレーションなしで、数学 的に等価性を証明

解決策の定義

- **統合検証技術**
  - システム～ゲート・レベルの全工程を範囲とする、統合的な検証技術
  - 検証環境を、工程毎の抽象度を越えて再利用する技術
    - 検証環境: テストベンチ、テストパターン（ランダム生成を含む）、レファレンスモデル、アサーション
- **プロファイリング技術**
  - HW間の通信量をシミュレーションで計測し、システム性能を見積もる技術
- **RTL合成技術**
  - 動作記述からRTLを合成する技術
  - HWの「動作周波数＋面積の見積り」の手段として利用
- **機能モデルとのスタティック等価性検証**
  - システム機能モデルに対する設計結果の等価性を検証する技術
    - 入力: システム機能モデル
    - 出力: HW動作モデル＋SWソース・コード
    - スタティック検証: シミュレーションなしで、数学的に等価性を検証

要求時期

- **解決策の要求時期を、ITRSフォーマットで記述。　技術レベルを、4段階**（Research Required、Development Underway、Qualification/Pre-Production、Continuous Improvement）**で表示**



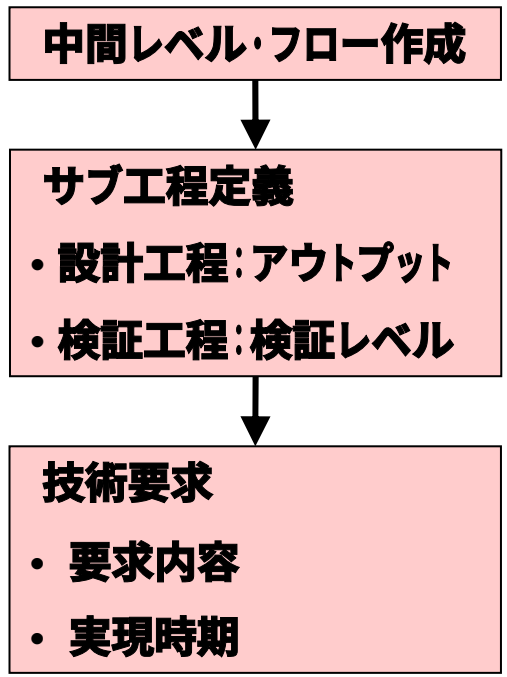
目次

- 国際化活動･国内活動の概要
- モバイル･プラットフォームSoCモデル
- 設計生産性要求値
- 設計生産性向上のための解決策検討
- まとめ、今後の計画

まとめ、今後の計画

＜まとめ＞

- モバイル・プラットフォームSoCモデル
  - 将来のSoC構造・規模を定量分析可能な形式で予測したモデルとして構築
  - ITRS Design ITWGに提示　⇒ 2005年版での掲載に向け賛同・評価を得た
- 設計技術ロードマップ
  - 設計技術向上（設計生産性向上）に対する要求値を明確化
  - システム・アーキテクチャ設計に注力し
    - 中間レベルCanonical Design Flowを作成
    - 各サブ工程における技術要求ロードマップを検討



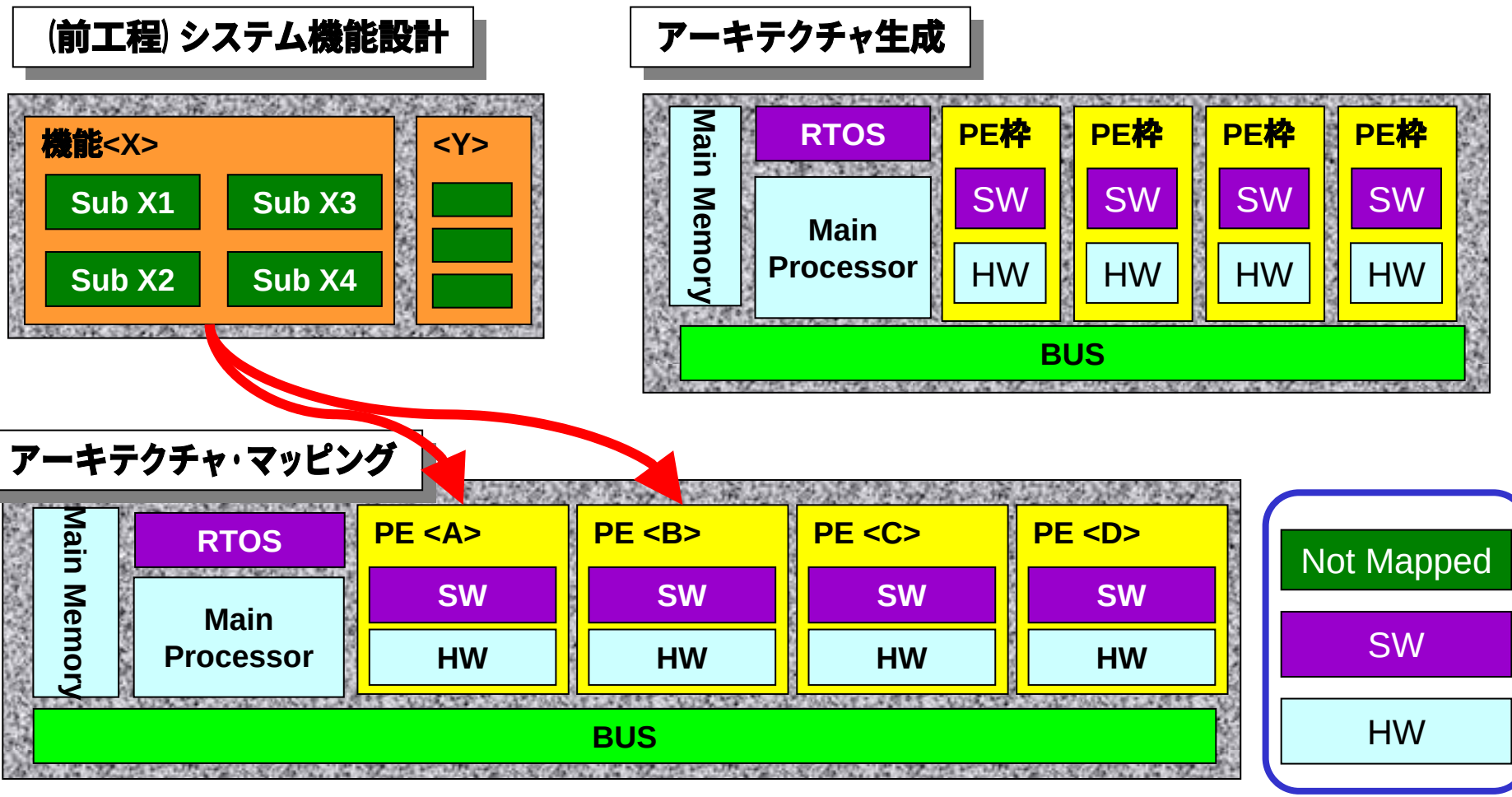
＜今後の計画＞

- モバイル・プラットフォームSoCモデル
  - 2005年版ITRS掲載に向け、ブラッシュアップ
- 設計技術ロードマップ
  - 検討枠組みは完成　⇒　全設計工程に展開
    - 全設計工程における中間レベル・フローの作成
    - 各サブ工程における技術要求と実現時期の定義

補足資料

システム・アーキテクチャ設計：概要

- (前工程) システム機能設計：　システムの機能を定義、SW/HWは未分割
- アーキテクチャ生成：　アーキテクチャを定義 (デザイン・パターンから選択)
- アーキテクチャ・マッピング：　機能を分割し、実現手段 (HW/SW) を判断し、PEに割り付け

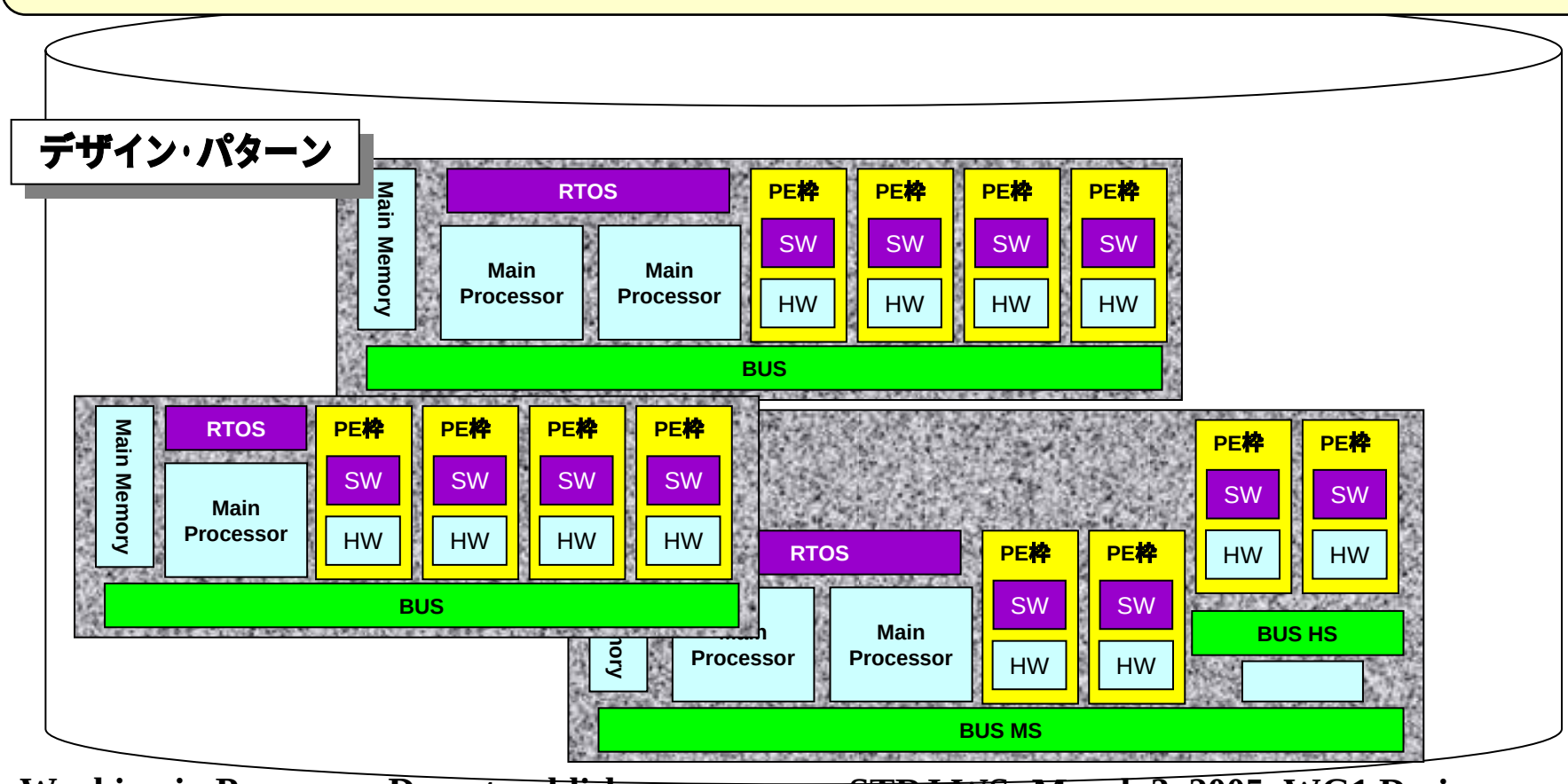


国内活動

年度	活動内容
1999	SoCのハードウェア設計生産性ロードマップ初版作成
2000	配線性クロスカット活動（設計WG、配線WG） ▸ 設計の観点から配線製造の課題を検討
2001	設計生産性をシステムレベルに拡張 ▸ 国内有識者への生産性アンケート実施
2002	デザインクライシスの定量化とソリューション ▸（高位合成、HW/SW協調合成）のロードマップ化
2003	設計遅れ起因の分析と提言 ▸ 設計現場から課題を抽出 ⇒ Time to market短縮への提言
2004	設計生産性ロードマップの策定 ▸ロードマップ策定のためのSoCモデルの設定 （モバイルプラットフォームSoCモデル） ▸課題抽出とポテンシャルソリューションの検討 （システムアーキテクチャ設計を対象）

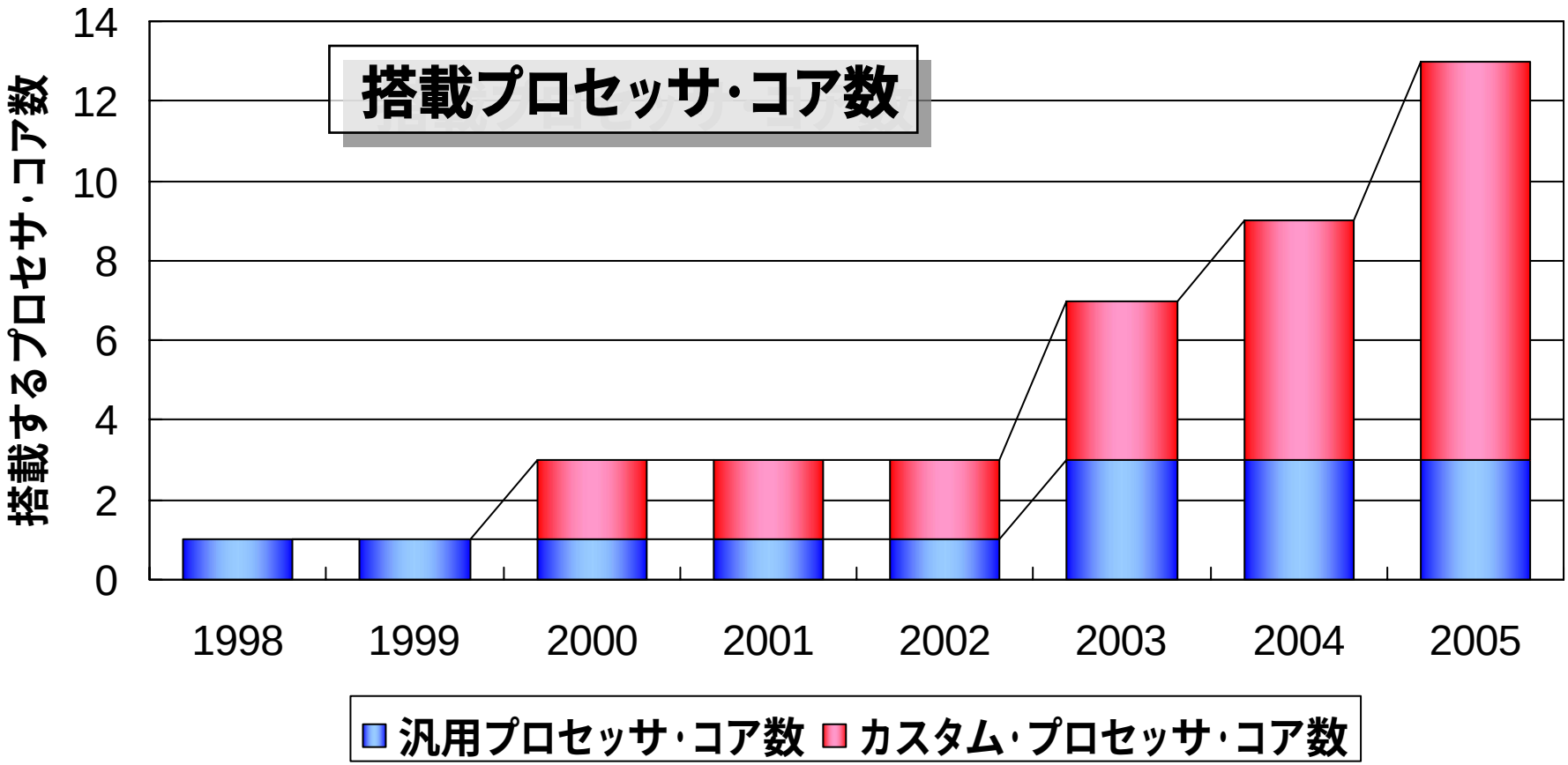
デザイン･パターン

- アーキテクチャ
  - SoCの基本構造
  - Main Processor、RTOS、メモリ、バス、PE（外枠のみ）で構成
- デザイン･パターン
  - 利用可能なアーキテクチャ群を、HW/SW協調設計用に蓄積



SoCモデルの構造：搭載PE数の増加例

- デジタル家電向けマルチメディア処理SoCにおいて、搭載PE数が毎年増加



出典：日経マイクロデバイス, “マイ・プロセサがSoCを強くする”, 2004年8月号

NECエレクトロニクスのデジタル家電向けマルチメディア処理SoCシリーズ