

設計生産性ロードマップ デザイン・クライシスと解決策の定量分析

半導体技術ロードマップ委員会
デザイン・ワーキング・グループ(WG1)
02年度・活動報告

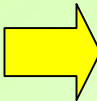
はじめに

- 設計生産性のロードマップ化を検討した背景
 - SoC設計におけるDesign Crisis
 - 微細化による搭載ゲート数の飛躍的増大により、設計工数が爆発的に増加する危機
- 活動経緯
 - 1999年: SoCのハードウェア設計に関する生産性ロードマップ確立
 - 01年版からITRSに掲載
 - 2001年: 検討範囲をシステムレベルにまで拡張
 - 国内有識者に対する設計生産性アンケートを実施
- 02年度活動の目的・目標
 - Design Crisisの定量化
 - Design Crisisに対する解決策のロードマップ化

設計生産性ロードマップ検討手順

- SoC設計生産性の定義
 - アウトプット: 設計生産量(Complexity)の定義
 - インプット: 必要設計工数(Effort)の定義
 - 範囲:
 - システム設計～マスク設計完了
- Design Crisisの定量化
 - 対象商品を「STRJ設計TFのLow Power SoCモデル」とし、SoC設計生産量(Complexity)を定量化
- 解決策の定義とロードマップ化
 - 必要な技術革新の定義
 - 効果と必要時期を検討

設計生産性の定義

$$\text{設計生産性} = \frac{\text{Value}}{\text{Development Cost}}$$


設計生産性を向上させるには、

- ①Valueを増加させ、
- ②Costを削減する。

- Value

- 市場から見た価値

- 機能量、消費電力、処理速度、データ幅(精度)、特性(アナログ)に要素分解
 - 市場ニーズとの合致(市場ニーズの創造)が前提

- 原価低減から見た価値

- 面積、テスト時間

- Development Cost

- 開発に必要な費用

- 工数(人件費)、EDA費、WS費、試作費、評価装置費、技術導入費

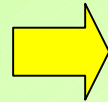
設計生産性の定義：機能量と設計工数

- 設計ロードマップ検討の観点から単純化し、

- 機能量 (Complexity)

- 工数 (Effort)

に着眼して検討



$$\text{設計生産性} = \frac{\text{Value}}{\text{Cost}} \Rightarrow \frac{\text{Complexity}}{\text{Effort}}$$

- 機能量 (Complexity) の定義

- ハードウェア (HW) の機能量 $\propto$ 論理回路規模

- ソフトウェア (SW) の機能量 $\propto$ 混載メモリ規模

- 工数 (Effort) の定義

- HW 設計工数 $\propto$ HW 機能量

← 設計再利用による線形化

- SW 設計工数 $\propto$ SW 機能量

← 設計再利用による線形化

- HW/SW協調設計工数 $\propto N \cdot \log(N)$ 、 $N = (\text{HW 機能量}) + (\text{SW 機能量})$

- HW/SW協調設計:

システムの要求制約を満たしつつ、要求機能を実現するHW/SW分割の決定

SoC Design Crisisの定量化

実現すべきSoCのプロファイル

Low Power SoC Model(Version:2002.12.18)

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
動作周波数	MHz	400	500	600	700	850	1,000	1,200	1,500	1,800
Chip Size	mm ²	64	64	64	64	64	64	64	64	64
プロセッサ部素子数	MTr	13	17	21	27	34	42	84	169	338
ロジックTr数	MTr	4.7	5.4	6.2	7.1	8.2	9.5	14.4	21.8	33.2
キャッシュメモリTr数	MTr	9	11	15	19	25	33	70	147	305
キャッシュメモリ容量	Mbit	1.4	1.9	2.5	3.2	4.2	5.5	11.7	24.5	50.8
専用デジタル部Tr数	MTr	26	33	42	53	67	84	168	335	671
ロジックTr数	MTr	6.8	7.8	9.0	10.3	11.9	13.7	20.8	31.6	48.0
SRAM部Tr数	MTr	20	25	33	42	55	70	147	304	623
SRAM部メモリ容量	Mbit	3.3	4.2	5.5	7.1	9.1	11.7	24.5	50.6	103.8

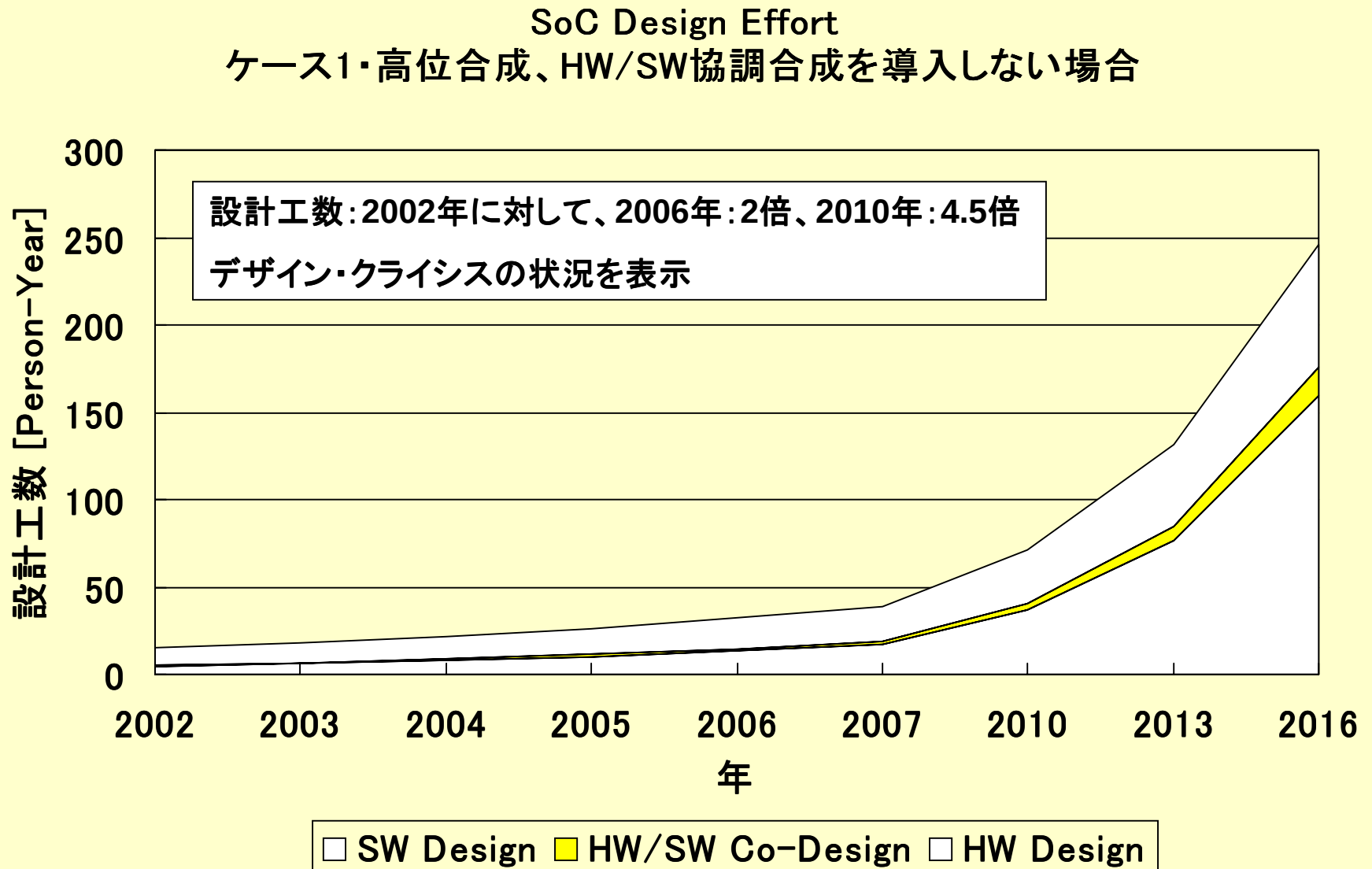
ケース1: 高位合成、HW/SW協調合成なし 技術革新を想定しない場合の設計工数(ケース1)

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
SW Design Effort	person-year	4.5	5.9	7.8	10.2	13.2	17.2	36.7	77.0	159.4
HW/SW Co-Design Effort	person-year	0.5	0.6	0.8	1.1	1.4	1.7	3.6	7.7	16.1
HW Design Effort	person-year	10.0	11.5	13.2	15.2	17.5	20.1	30.6	46.5	70.6
Total Effort	person-year	15.0	18.1	21.8	26.4	32.1	39.0	70.9	131.1	246.2

システム複雑度

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
System Complexity		14.5	17.4	21.0	25.4	30.7	37.3	67.3	123.4	230.1

Case 1: 技術革新を想定しない場合の設計工数(Effort)

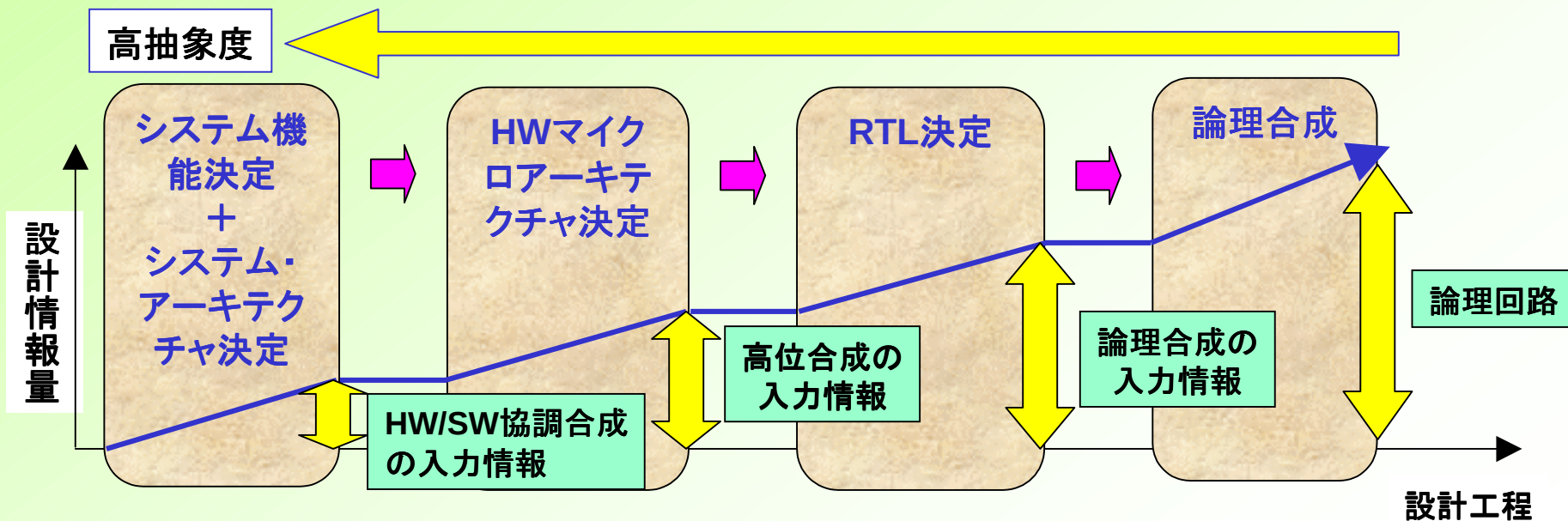


技術革新の定義

- デザイン・クライシスを回避するための技術革新
 - 高位合成、HW/SW協調合成
- 高位合成：
 - ハードウェアの動作を表現する動作記述から、速度、面積等の制約条件を与えることにより、ハードウェアの動作順序およびデータパスの構成を自動生成・最適化し、制御回路とデータパス回路の論理合成可能なRTL記述を出力する設計自動化技術
 - 利用開始時期： 2004年
- HW/SW協調合成：
 - システムの動作記述（ソフトウェアとハードウェアに分割する前）から、システムの性能やコストなどの制約を与えることにより、ハードウェアの高位合成用・動作記述（高位合成の入力）と、ソフトウェアのソース・コードを出力する設計自動化技術
 - 利用開始時期： 2007年

抽象度の高位化

- 技術革新(高位合成、HW/SW協調合成)における設計生産性向上
 - 設計抽象度の高位化により、生産性を向上
 - 設計抽象度の高位化:最終成果物(SoC)の定義に必要な情報量を削減
- 設計情報量の削減方法
 - 上流工程からの情報を受け、
下流工程に渡すべき情報を合成する設計自動化技術の実現
 - 過去: 配置配線、論理合成
 - 将来: 高位合成、HW/SW協調合成



高位合成による設計情報の削減

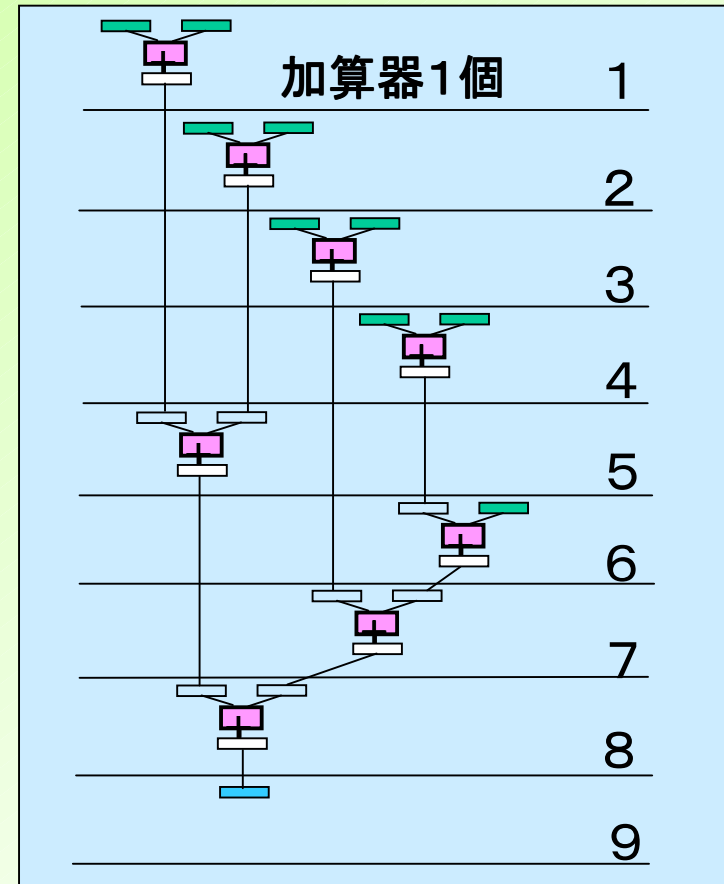
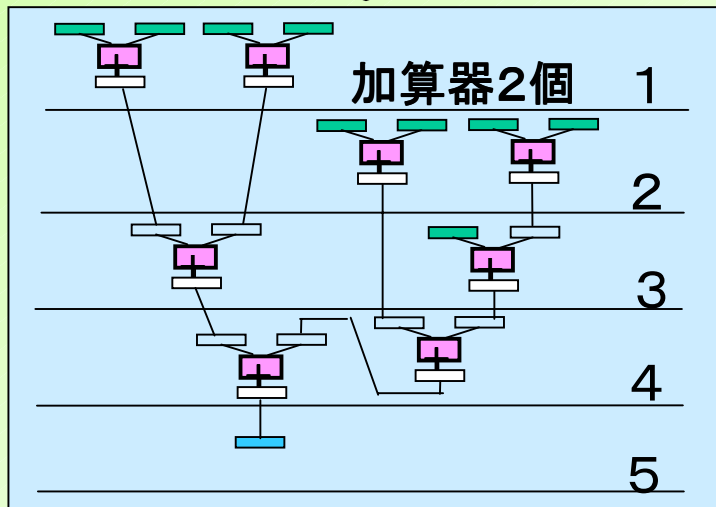
システム機能決定

$x = a + b + c + d + e + f + g + h + i;$

RTL決定

高位合成で自動化

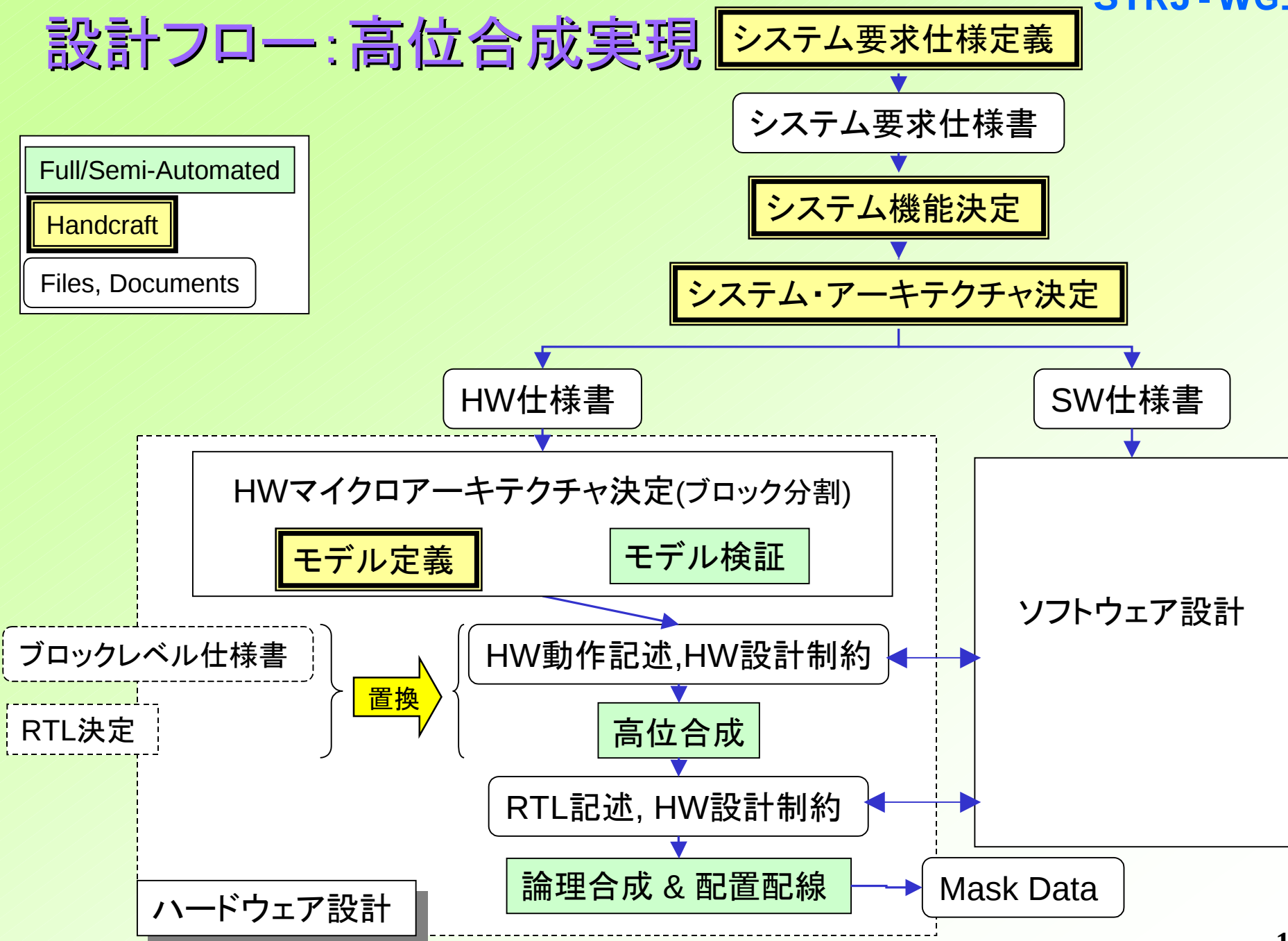
性能・面積間のトレードオフに配慮して
演算器の数、演算順序を決定



性能と面積(加算器数)のトレードオフ

加算器	処理性能
2個	5クロック
1個	9クロック

設計フロー: 高位合成実現



Case 2: 高位合成

- 仮定:
 - 高位合成によるHW設計の設計生産性向上率: 16倍
 - 高位合成の利用開始時期: 2004年
- 高位合成の適用率(HW機能に対する)
 - 2004年: 12.5%
 - 2007年: 100%

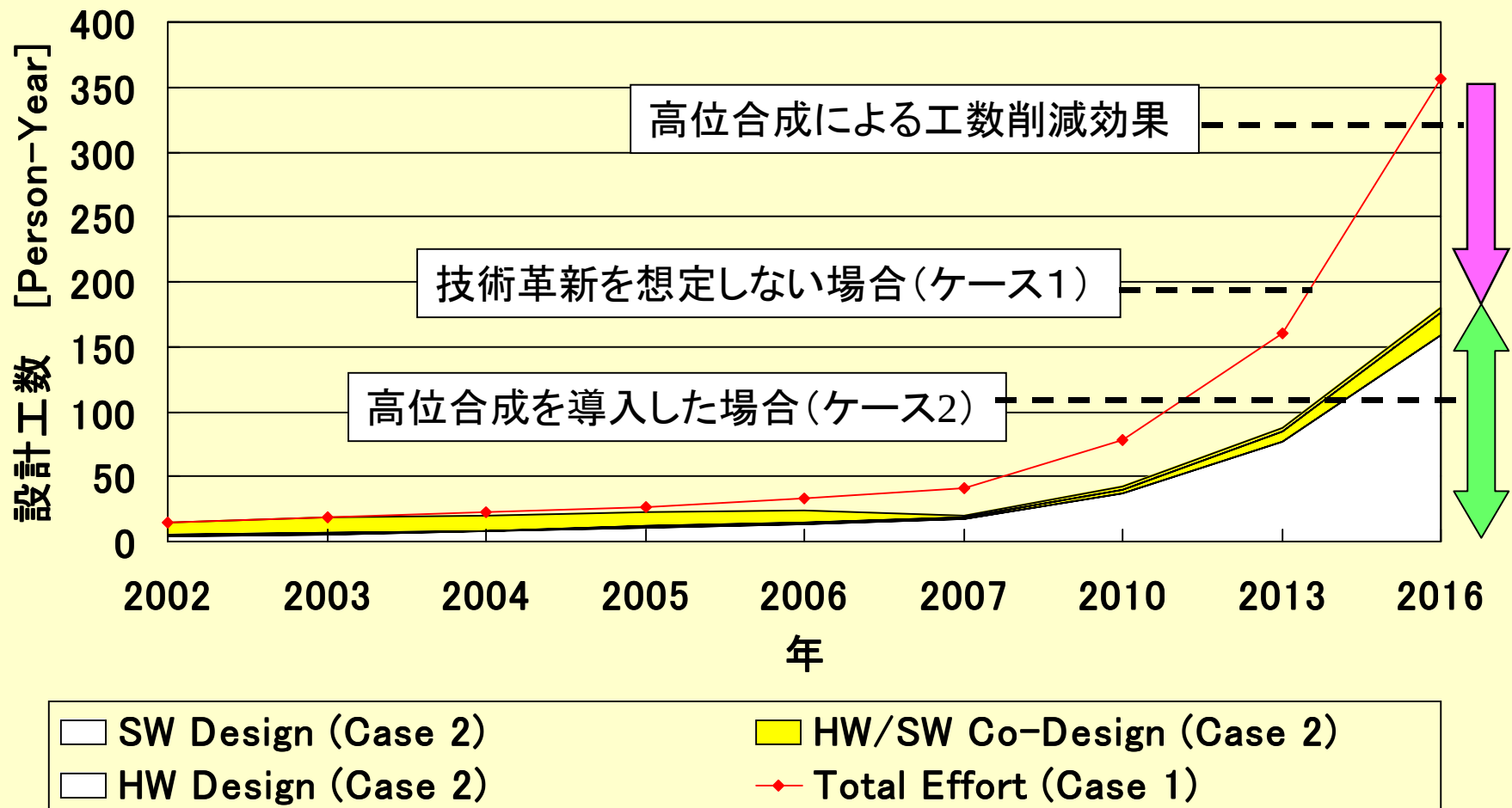
高位合成を実現した場合の設計工数(ケース2)

ケース2: 高位合成が2004年に実現(技術革新1)

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
Abstraction Level of High Level Design		1	1	16	16	16	16	16	16	16
Applicability of High Level Synthesis		0%	0%	12.5%	25.0%	50.0%	100.0%	100.0%	100.0%	100.0%
SW Design Effort	person-year	4.5	5.9	7.8	10.2	13.2	17.2	36.7	77.0	159.4
HW/SW Co-Design Effort	person-year	0.5	0.6	0.8	1.1	1.4	1.7	3.6	7.7	16.1
HW Design Effort	person-year	10.0	11.5	11.7	11.6	9.3	1.3	1.9	2.9	4.4
Total Effort	person-year	15.0	18.1	20.3	22.9	23.9	20.2	42.2	87.5	180.0

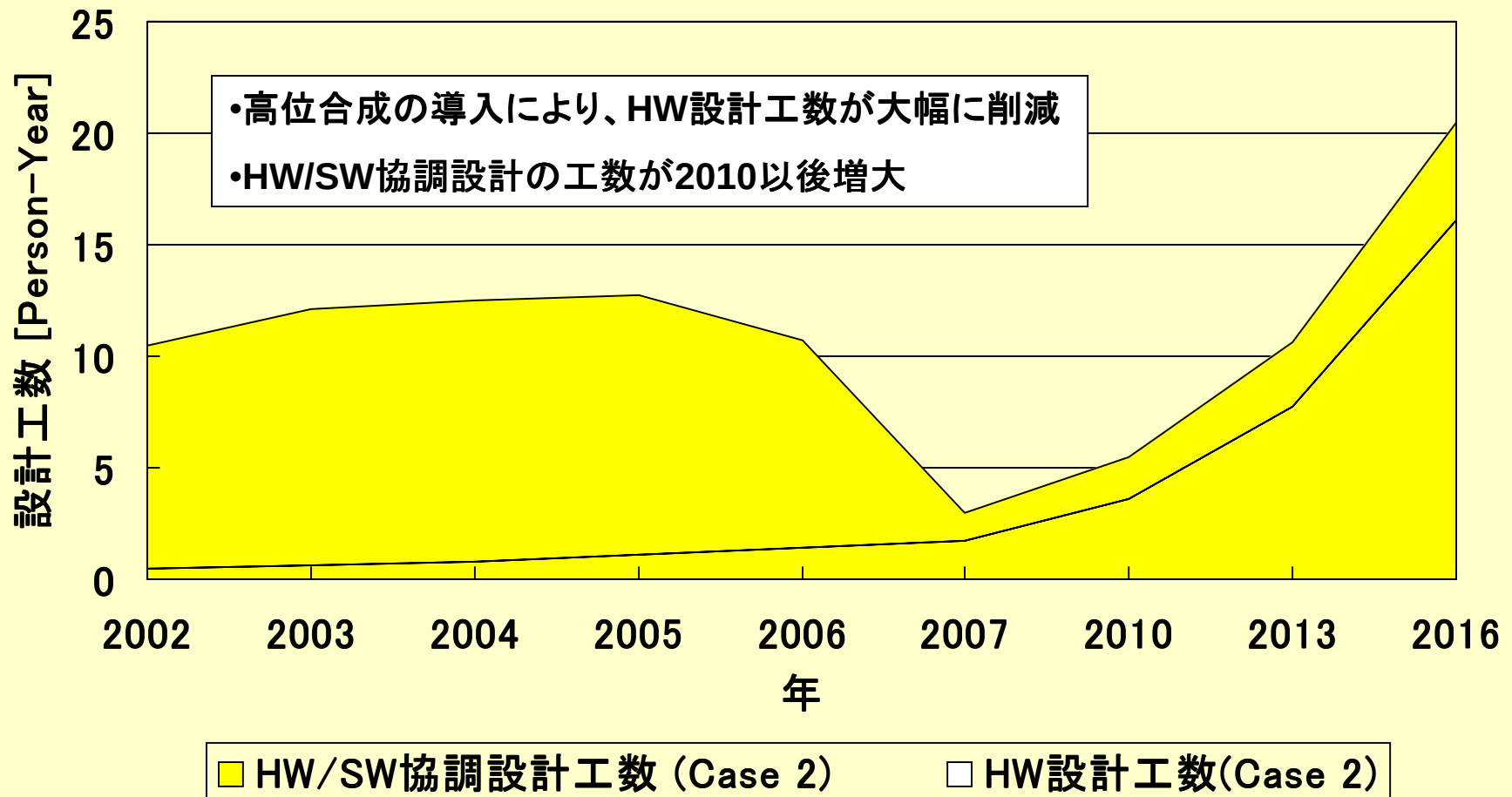
Case 2: 高位合成の導入による設計工数削減

SoC Design Effort
ケース2・高位合成導入を想定した場合

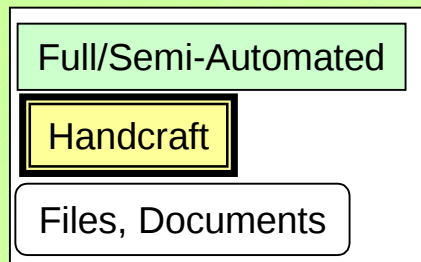


Case 2: 高位合成の導入による設計工数削減

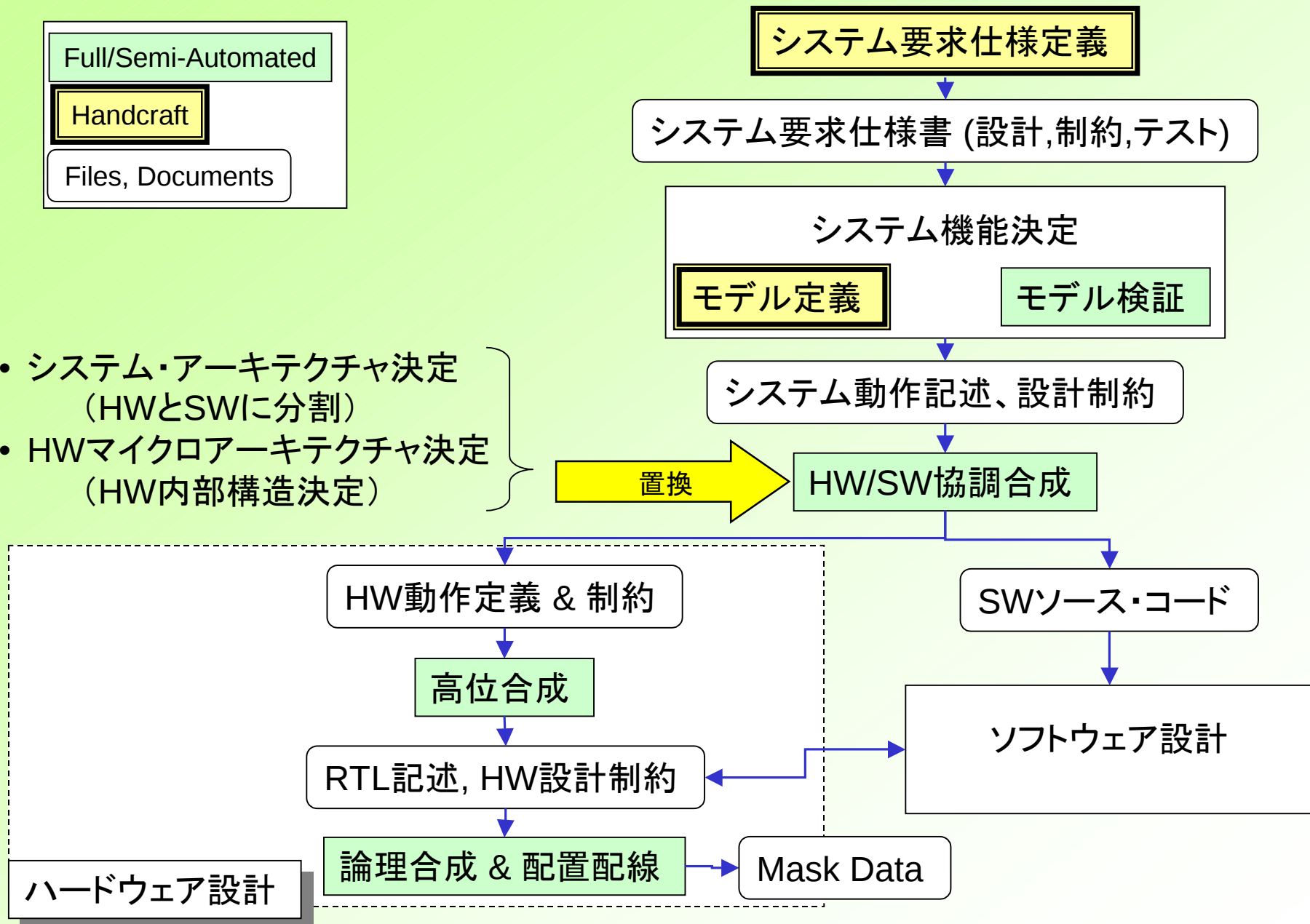
HW/SW協調設計工数 + HW設計工数
ケース2・高位合成導入を想定した場合
(ソフトウェア設計工数を除いたSoC設計工数)



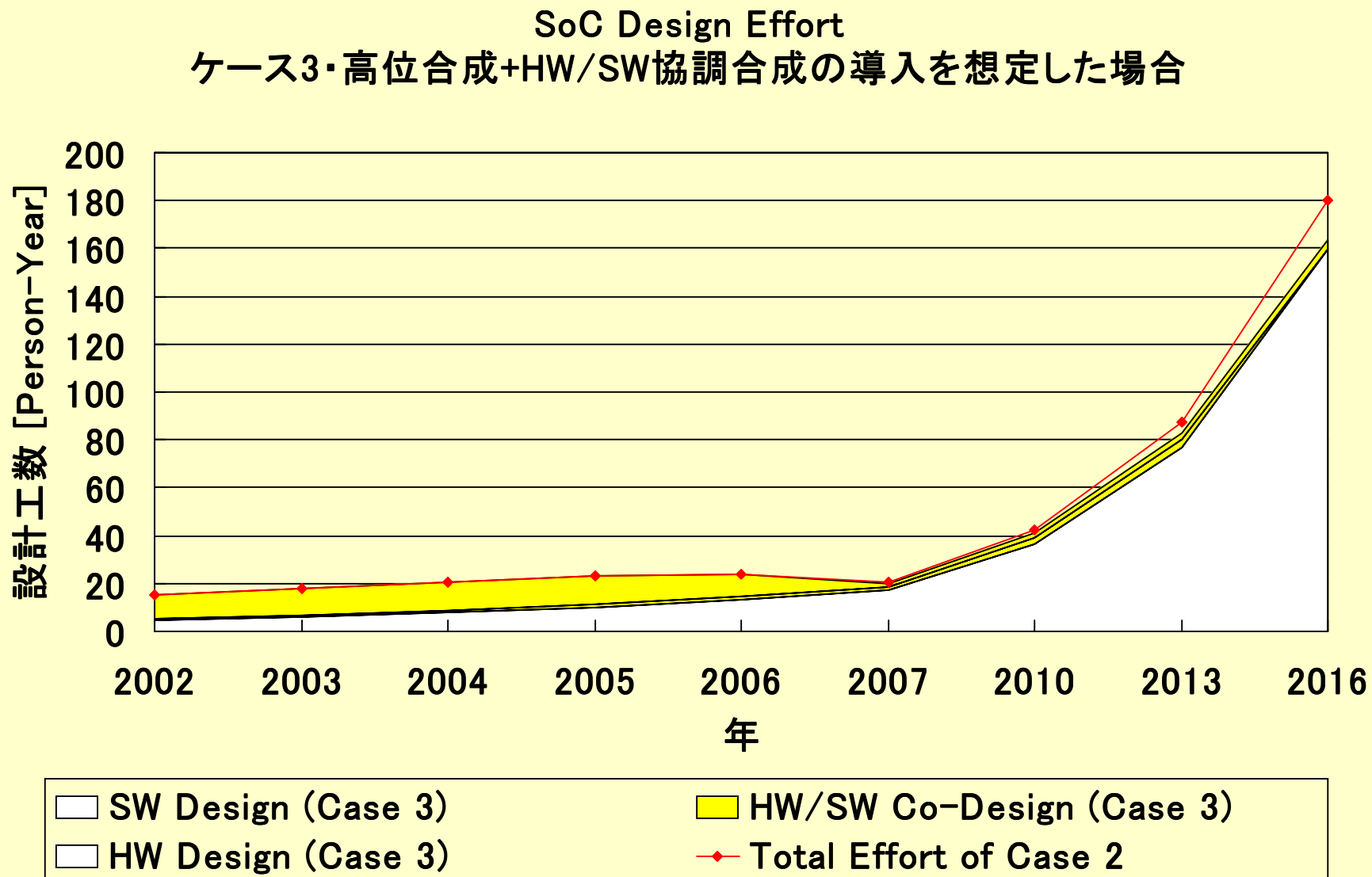
設計フロー: HW/SW協調合成実現



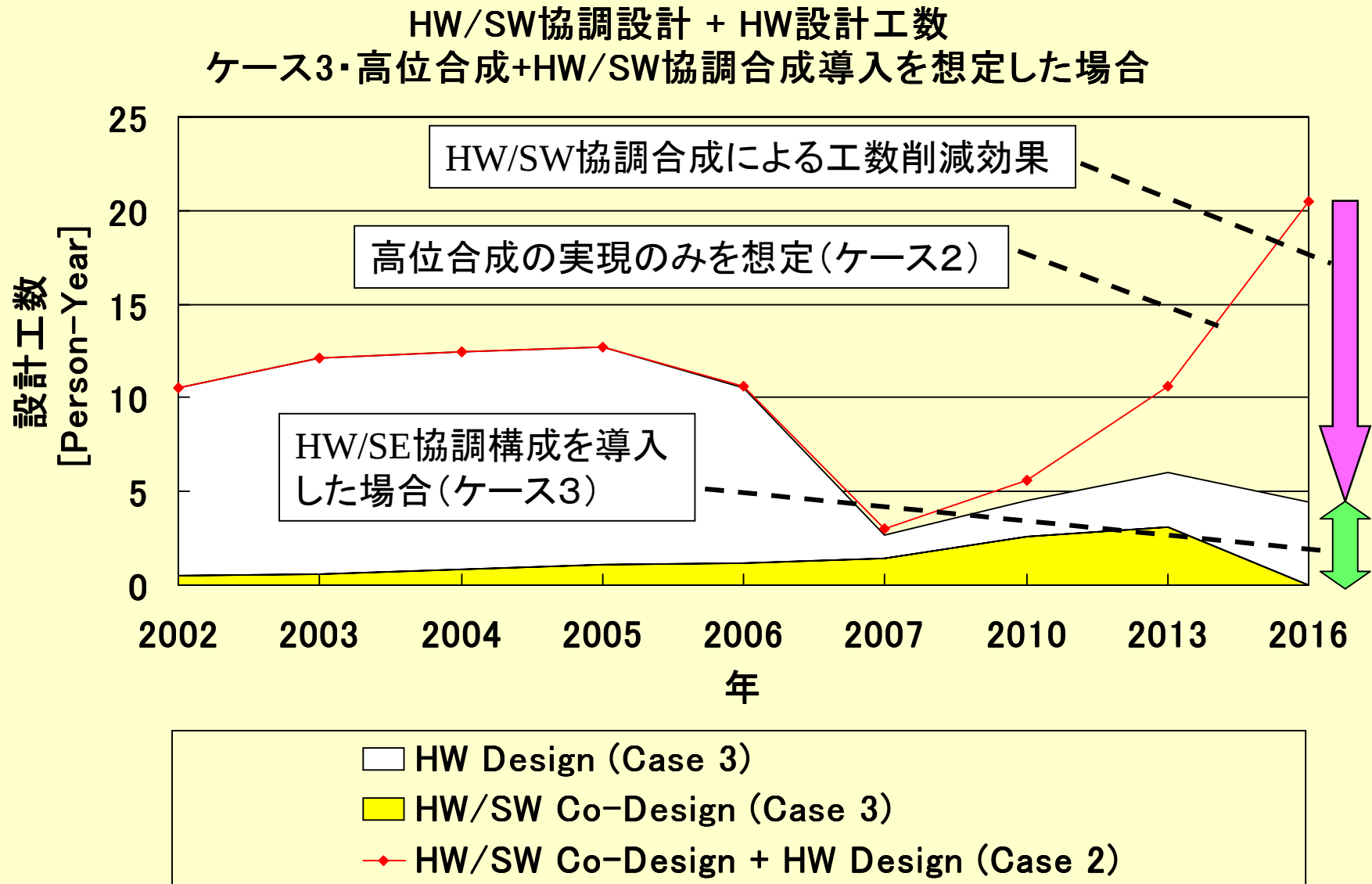
- システム・アーキテクチャ決定 (HWとSWに分割)
- HWマイクロアーキテクチャ決定 (HW内部構造決定)



Case 3: 高位合成+HW/SW協調合成による設計工数削減



Case 3: 高位合成+HW/SW協調合成による設計工数削減



まとめ

- 設計生産性を定義し、Design Crisisを定量的に分析
 - 分析範囲: SoCシステム設計～レイアウト設計
 - 対象商品: STRJ設計TFのLow Power SoCモデル
- Design Crisisの解決に必要な技術を定義し、期待効果を定量的に分析
 - 必要技術の定義: 高位合成、HW／SW協調合成
 - 両技術の導入による設計フローの変遷を明確化
 - 定義した設計フローを、2003年版ITRSに掲載の予定
- 分析結果
 - 高位合成、HW／SW協調合成
 - Design Crisis回避に必須の技術
 - SW Design Crisis
 - 2007年以降は、SoC 設計工数の80%以上を占有
 - 想定した技術革新が実現しても、SW設計に要する工数が大きな脅威

補足資料

補足資料

設計WG(WG1) 委員構成

- 山本 一郎(リーダー)
沖電気工業
- 田口 浩文(サブリーダー)
松下電器産業
- 樋渡 有(国際対応)
東芝
- 山田 明宏(幹事)
トッパン・テクニカル・デザインセンター
- 古市 慎治 三洋電機
- 山内 貴行 シャープ
- 小野 信任 SIIイーディーエー・テクノロジー
- 古井 芳春
ソニーセミコンダクタ九州
- 柿本 勝 ソニー(株)
- 柏木 治久 日本電気
- 斎藤 実 富士通
- 染谷 勤
フューチャーデザインオートメーション
- 小野 眞司 三菱電機
- 上村 卓三 ローム
- 今井 正治 大阪大学
- 小澤 時典 STARC
- 村岡 道明 STARC
- 濱田 英幸 STARC

18名の設計技術に関するエキスパートで構成

目次

- はじめに
- 設計生産性の定義
- 設計生産性の向上方法
- 設計生産性ロードマップ
 - 技術革新の定義
 - ロードマップ・テーブル
- 詳細設計フロー
- 提言、まとめ

はじめに

- SoCのDesign Crisis
 - テクノロジーの微細化による搭載ゲート数の飛躍的増大
⇒設計工数が爆発的に増加して窮地に陥る
- Design Crisisに対処するための技術発展が必要
 - ①IP再利用の促進
 - ②システムレベル設計の効率化(特に、設計抽象度の高度化)
 - ③設計全工程の自動化推進
- 活動経緯
 - 1999年: SoC(ハードウェア)設計の設計生産性ロードマップ確立
 - ITRS2001に掲載
 - 2001年: 検討範囲をシステムレベルに拡張
 - 国内有識者に対する設計生産性アンケートを実施
- 02年度活動の目的・目標
 - SoCのDesign Crisisを定量的に分析
 - Design Crisisに対する解決策のロードマップ化 (システムレベルを含む)

はじめに: SoC設計生産性ロードマップ

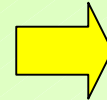
- SoC設計生産性の定義
 - 設計生産量(Complexity)の定義
 - 必要設計工数(Effort)の定義
 - 範囲:
 - システム設計～マスク設計完了 (SoCのシステムレベル設計を含む)
- SoC Design Crisisの定量化
 - SoC設計生産量(Complexity)の増加度
 - 対象商品=「STRJ設計TFのLow Power SoCモデル」
 - 要求される設計生産性
 - 1商品当りのEffort＝一定
- SoC Design Crisisの解決策
 - 必要技術の定義、効果予測、必要時期
 - 2技術革新(高位合成、HW／SW協調合成)を定義し、効果予測、必要時期の検討を実施

目次

- はじめに
- 設計生産性の定義
 - 設計生産性の定義
 - ComplexityとEffort
 - システムと検討範囲
- SoC Design Crisisの解決策
- 設計生産性ロードマップ
- 詳細設計フロー
- 提言、まとめ

設計生産性の定義

$$\text{設計生産性} = \frac{\text{Value}}{\text{Development Cost}}$$



設計生産性を向上させるには、

- ①Valueを増加させ、
- ②Costを削減する。

- Value

- 市場から見た価値
 - 機能量、消費電力、処理速度、データ幅(精度)、特性(アナログ)
 - 市場ニーズに合致した(ニーズを創造する)機能
- 原価低減から見た価値
 - 面積、テスト時間

- Development Cost

- 工数(人件費)、EDA費、WS費、試作費、評価装置費、技術導入費

- STRJ 設計WGは、

Value=機能量(Complexity) および

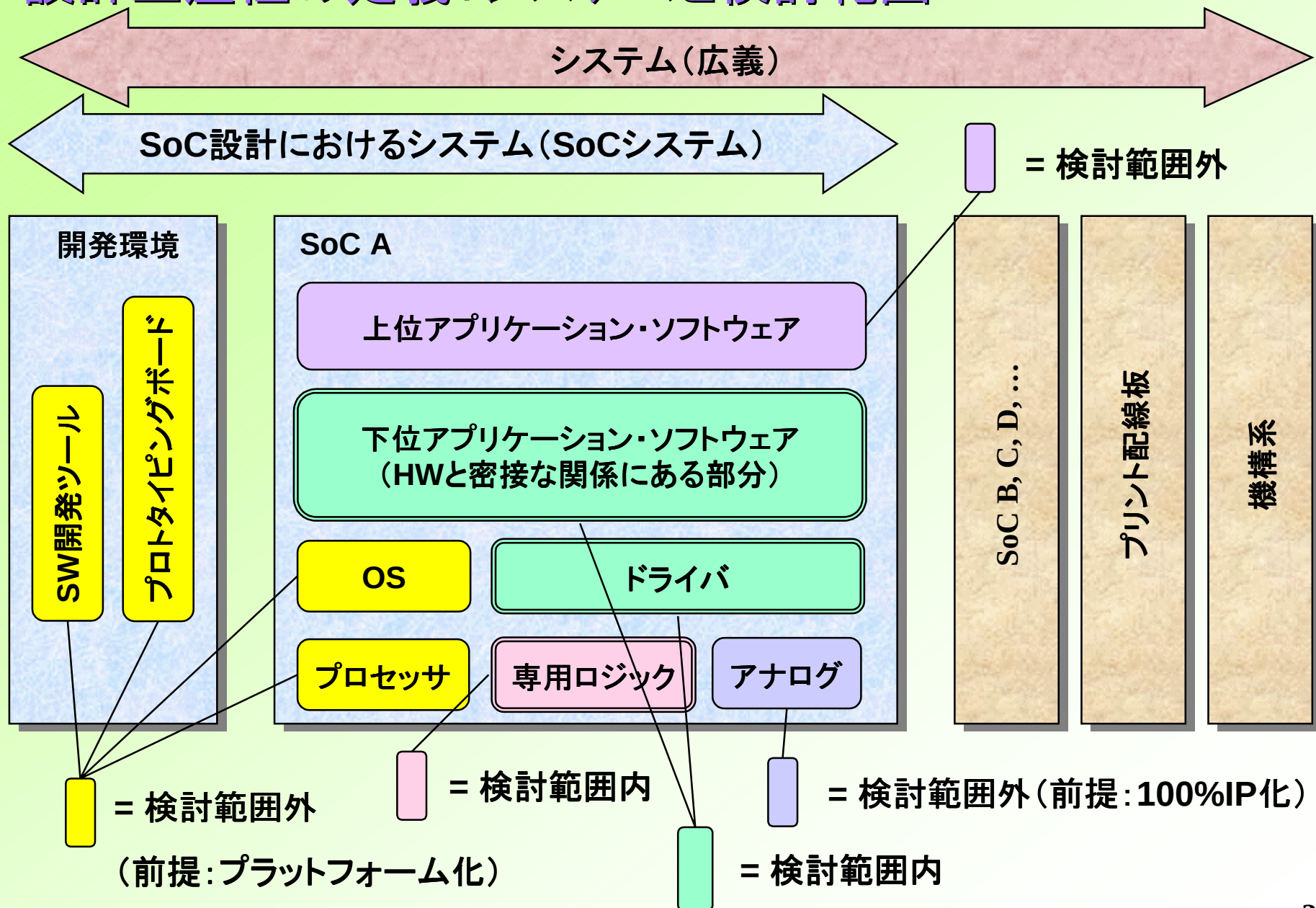
Cost = 工数(Effort)に着眼

設計生産性の定義: ComplexityとEffort

$$\text{設計生産性} = \frac{\text{Complexity}}{\text{Effort}}$$

- 設計生産量 (Complexity) の定義
 - 「STRJ設計TFのPDA用Low Power SoCモデル」を実現すべき商品と設定
 - HWのComplexity $\propto$ 論理回路規模
 - SWのComplexity $\propto$ メモリ規模
- 必要設計工数 (Effort) の定義
 - HW Effort $\propto$ HW Complexity $\leftarrow$ 設計再利用による線形化 (後述)
 - SW Effort $\propto$ SW Complexity $\leftarrow$ 設計再利用による線形化 (後述)
 - HW/SW分割Effort $\propto N^2$
 $N = (\text{HW Complexity}) + (\text{SW Complexity})$

設計生産性の定義：システムと検討範囲



目次

- はじめに
- 設計生産性の定義
- **SoC Design Crisisの定量化**
- SoC Design Crisisの解決策
- 解決策の詳細議論
- 設計生産性ロードマップ
- 詳細設計フロー
- 提言、まとめ

SoC Design Crisisの定量化 (1/2)

- 想定SoCプロファイル:
 - PDA用Low Power SoC(設計TF)
- 仮定: Complexity
 - SW:
 - Processor用Memoryのサイズに比例
 - HW:
 - 専用ロジックの規模に比例
 - HW/SW分割:
 - HW Complexity + SW Complexity
- 仮定: Effort
 - SW: SW Complexityに比例
 - IP利用による粒度を大型化の効果
 - HW: HW Complexityに比例
 - IP利用による粒度を大型化の効果
 - HW/SW分割に関するEffort
 - Complexityの $N \cdot \log(N)$ に比例
- IP利用効果を検証しロードマップ化

SoC Design Crisisの定量化 (2/2)

Low Power SoC Model(Vershion:2002.12.18)

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
CPUブロック素子数	MTr	13	17	21	27	34	42	84	169	338
ロジックTr数	MTr	4.7	5.4	6.2	7.1	8.2	9.5	14.4	21.8	33.2
キャッシュメモリTr数	MTr	9	11	15	19	25	33	70	147	305
キャッシュメモリ容量	Mbit	1.4	1.9	2.5	3.2	4.2	5.5	11.7	24.5	50.8
組み込み論理ブロックTr数	MTr	26	33	42	53	67	84	168	335	671
ロジックTr数	MTr	6.8	7.8	9.0	10.3	11.9	13.7	20.8	31.6	48.0
SRAM部Tr数	MTr	20	25	33	42	55	70	147	304	623
SRAM部メモリ容量	Mbit	3.3	4.2	5.5	7.1	9.1	11.7	24.5	50.6	103.8

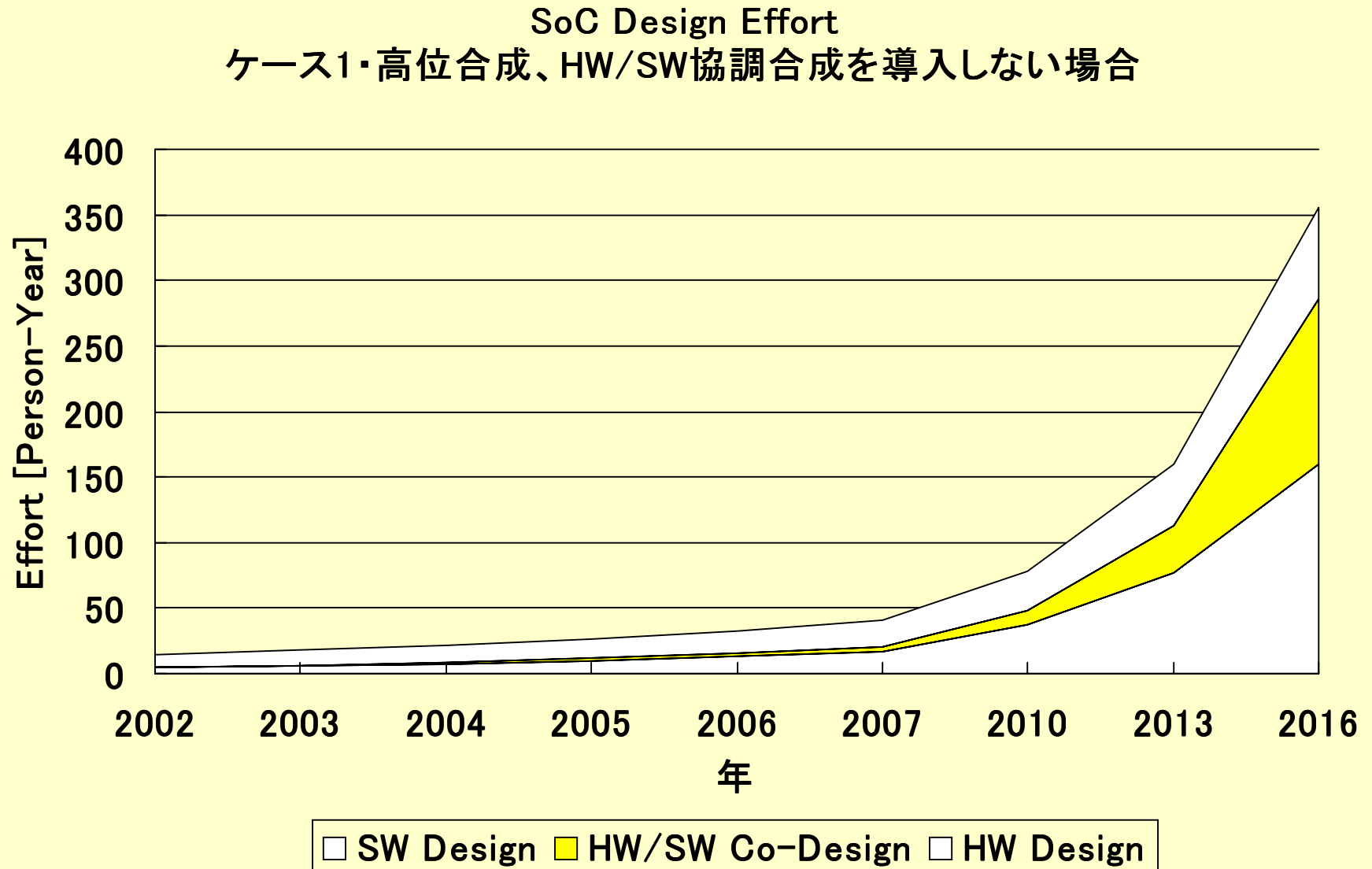
ケース1: 高位合成、HW/SW協調合成なし

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
SW Design Effort	person-year	4.5	5.9	7.8	10.2	13.2	17.2	36.7	77.0	159.4
HW/SW Co-Design Effort	person-year	0.5	0.7	1.1	1.5	2.2	3.3	10.8	36.2	125.9
HW Design Effort	person-year	10.0	11.5	13.2	15.2	17.5	20.1	30.6	46.5	70.6
HW/SW Co-Design + HW Design	person-year	10.5	12.2	14.3	16.7	19.7	23.4	41.3	82.7	196.5
Total Effort	person-year	15.0	18.2	22.1	26.9	33.0	40.6	78.0	159.7	356.0

システム複雑度

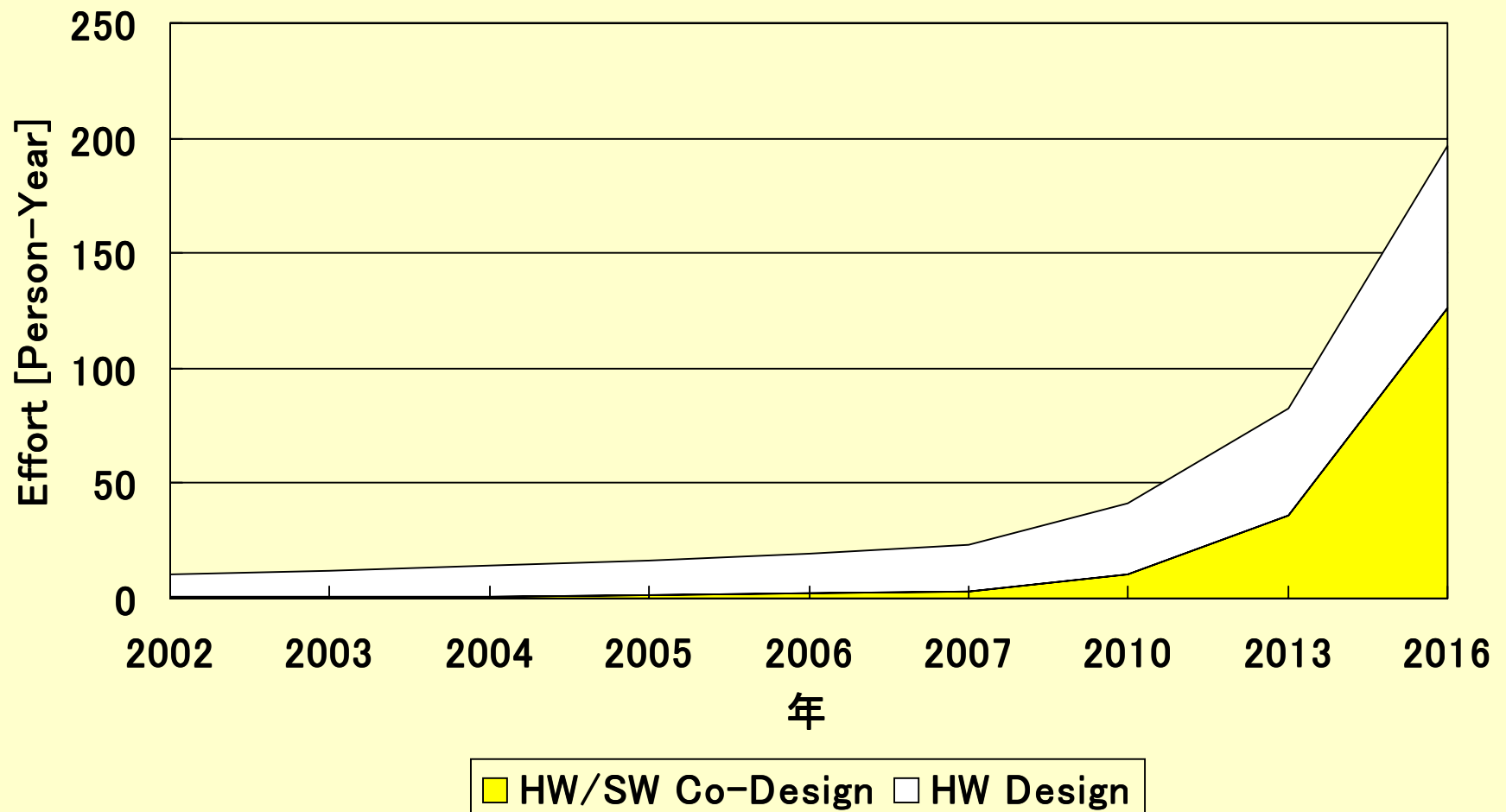
	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
System Complexity		14.5	17.4	21.0	25.4	30.7	37.3	67.3	123.4	230.1

Case 1: 技術革新を想定しない場合のDesign Effort



Case 1: 技術革新を想定しない場合のDesign Effort

HW/SW Co-Design + HW Design Effort
ケース1・高位合成、HW/SW協調合成を導入しない場合



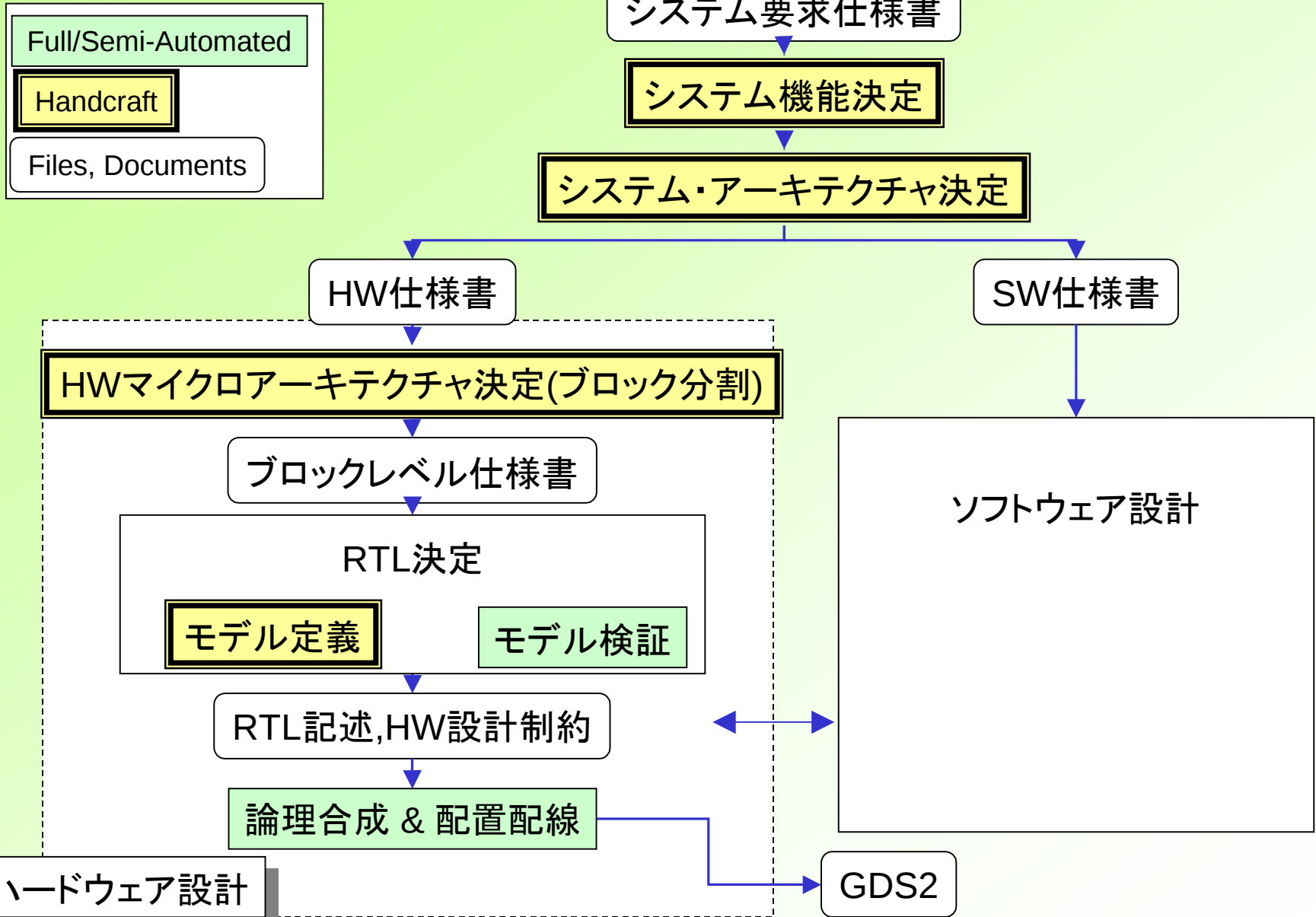
目次

- はじめに
- 設計生産性の定義
- SoC Design Crisisの定量化
- SoC Design Crisisの解決策
 - 技術革新(高位合成、HW/SW協調合成)の定義
 - 設計フロー:現在
 - 設計フロー:高位合成実現
 - 設計フロー:HW/SW協調合成実現
- 解決策の詳細議論
- 設計生産性ロードマップ
- 詳細設計フロー
- 提言、まとめ

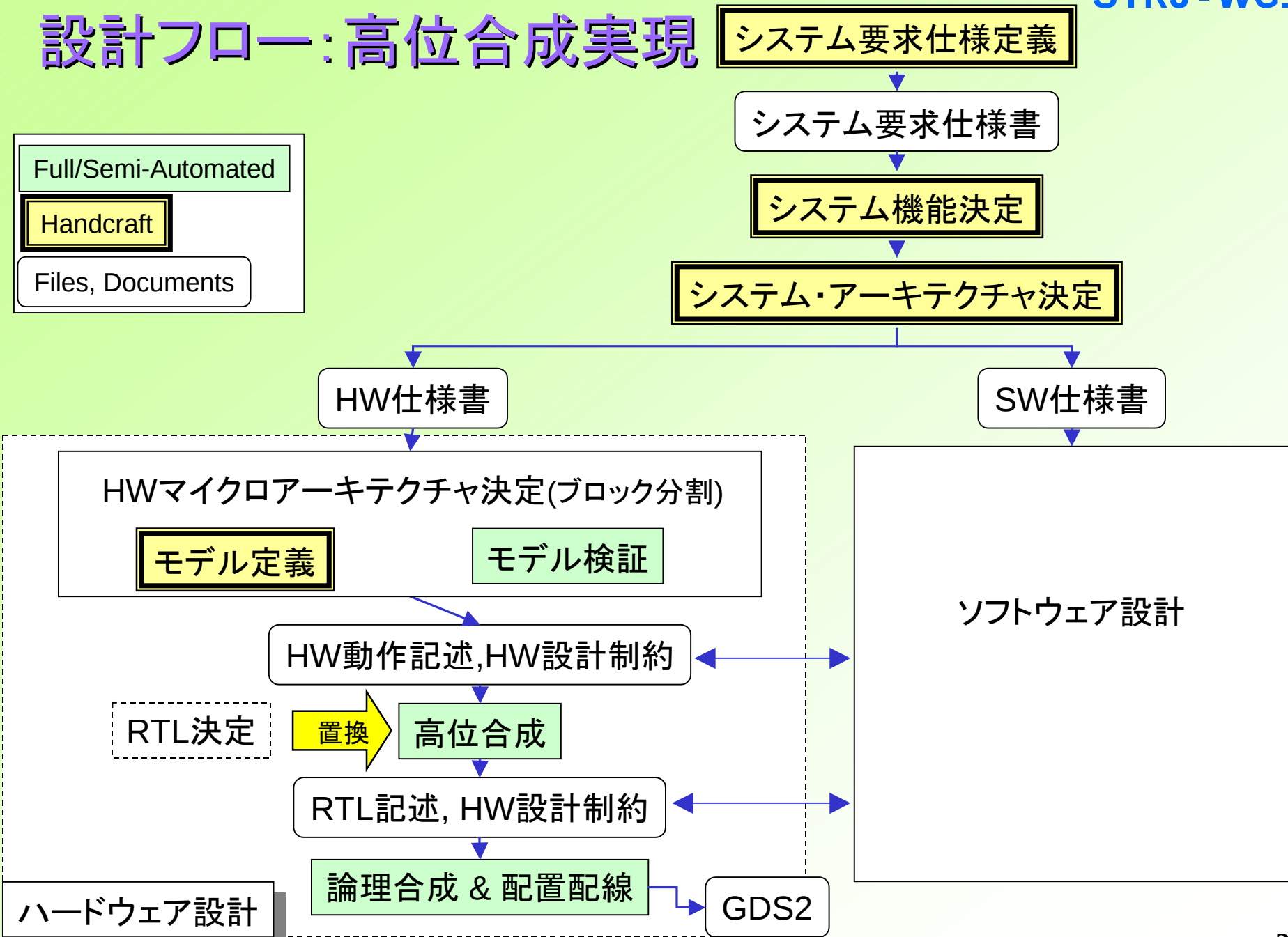
技術革新(高位合成、HW/SW協調合成)の定義

- 技術革新1： 高位合成：
 - 定義：
 - ハードウェアの動作を表現する動作記述から、速度、面積等の制約条件を与えることにより、ハードウェアの動作順序およびデータパスの構成を自動生成・最適化し、制御回路とデータパス回路の論理合成可能なRTL記述を出力する。
 - 利用開始時期： 2004年
- 技術革新2： HW/SW協調合成：
 - 定義：
 - システムの動作記述(ソフトウェアとハードウェアに分割する前)から、システムの性能やコストなどの制約を与えることにより、ハードウェアの高位合成用・動作記述(高位合成の入力)と、ソフトウェアのソース・コードを出力する。
 - 利用開始時期： 2007年

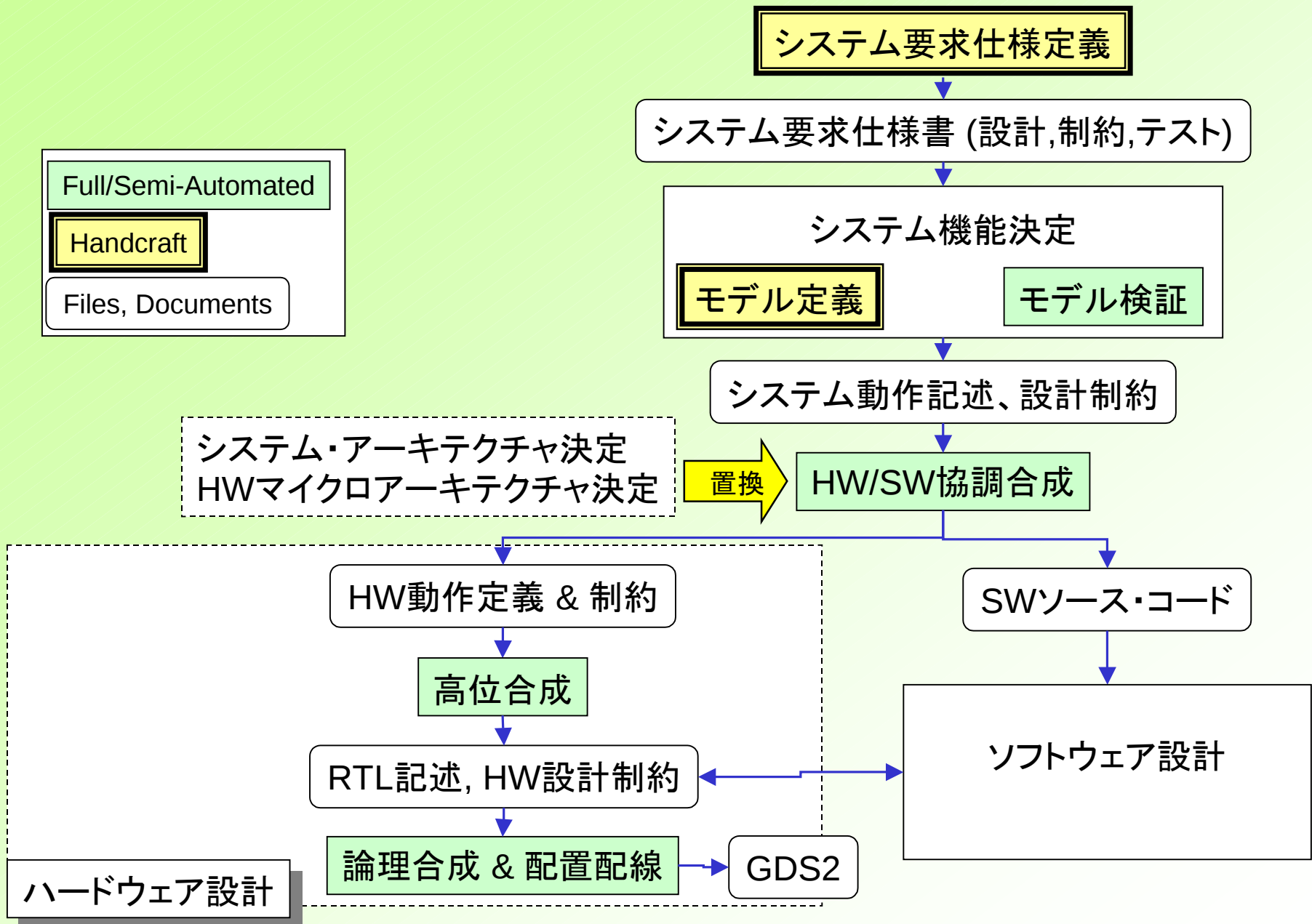
設計フロー: 現在



設計フロー: 高位合成実現



設計フロー: HW/SW協調合成実現

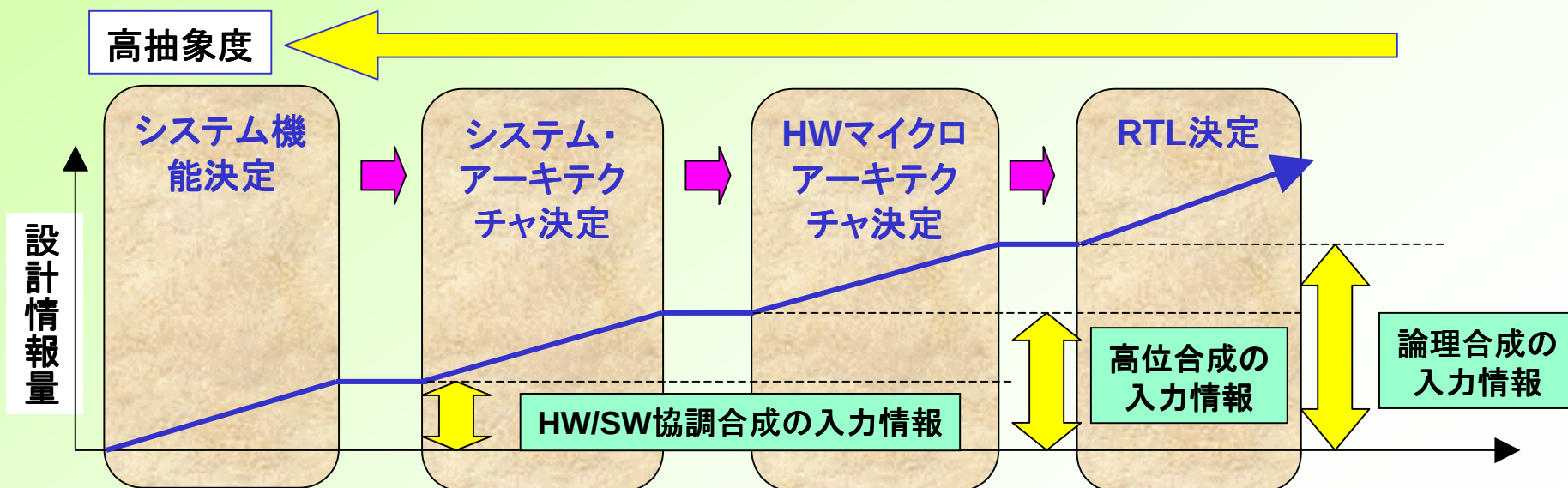


目次

- はじめに
- 設計生産性の定義
- SoC Design Crisisの定量化
- SoC Design Crisisの解決策
- 解決策の詳細議論
 - 抽象度の高位化
 - 粒度の大型化
 - 高位合成ツールの性能向上
 - 論理合成、配置配線
 - 価値の高い仕様の作成
- 設計生産性ロードマップ
- 詳細設計フロー
- 提言、まとめ

抽象度の高位化

- 「設計抽象度高位化」の目的
 - 最終成果物(SoC)の定義に必要な情報量を削減し、生産性を向上する。
- 設計情報量の削減方法
 - 合成プログラムが、上流工程からの情報に基づき自動生成する。
 - HW/SW協調合成、高位合成、論理合成、配置配線
- 究極(最小必要)の設計情報
 - 動作記述＋外部インターフェース条件＋使用環境



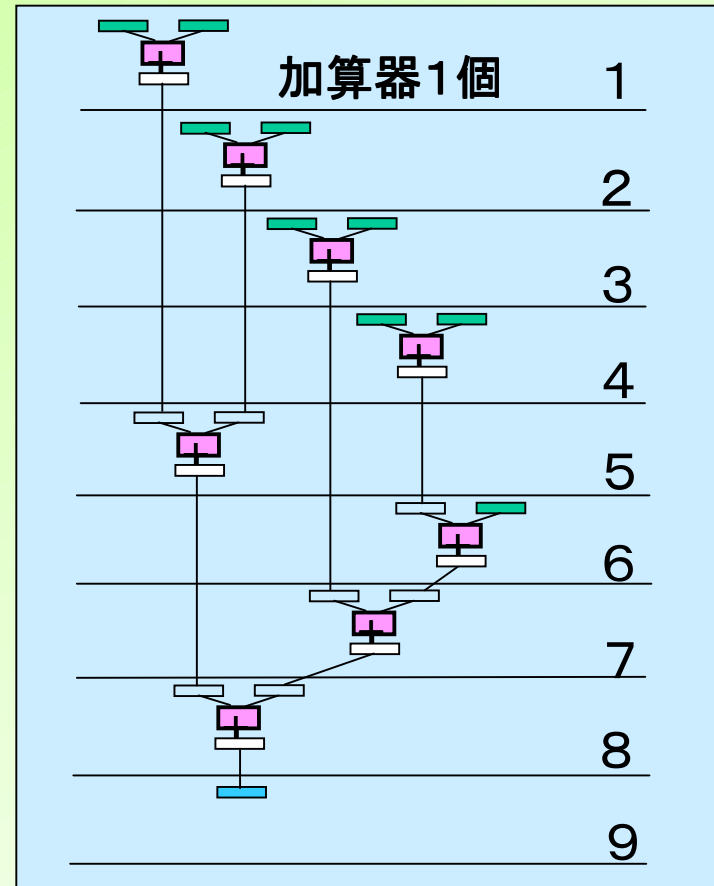
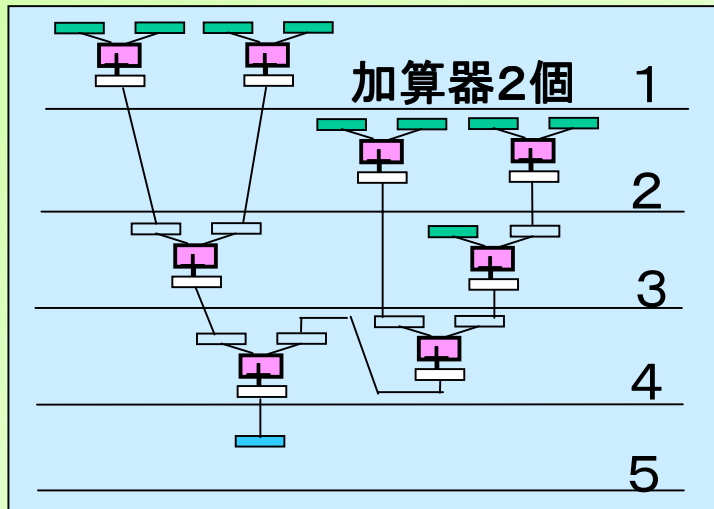
抽象度の高位化：設計情報の削減

システム機能決定

$x = a + b + c + d + e + f + g + h + i;$

RTL決定

演算器の数、演算順序を決定



性能と面積のトレードオフ

加算器	処理性能
2個	5クロック
1個	9クロック

抽象度の高位化:生産性向上

- 最終成果物 (SoC) 定義に必要な情報量を減少させ、

- ① 記述すべき情報量の削減
- ② 一度に考える情報量の削減
(思考可能な規模に抑える効果)
- ③ 仕様変更の容易化
- ④ 既存の設計情報の理解容易化

の実現により、設計生産性を向上させる。

- 設計生産性ロードマップにおける仮定

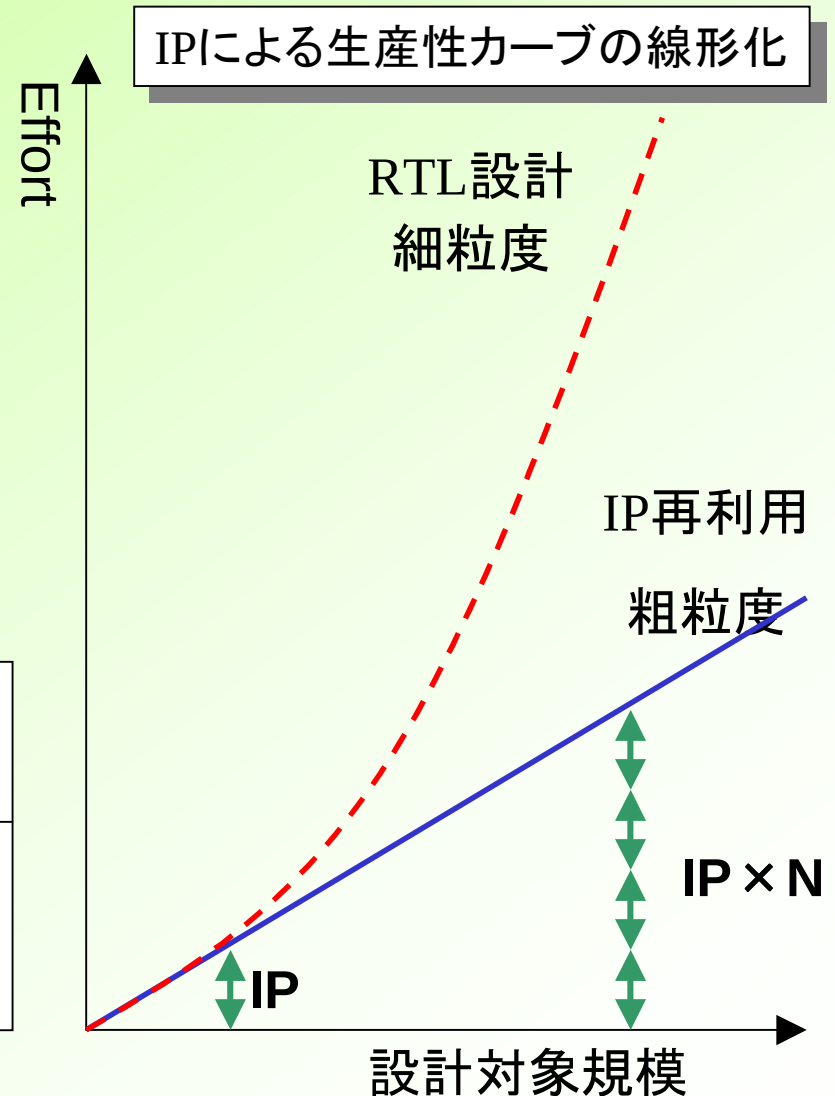
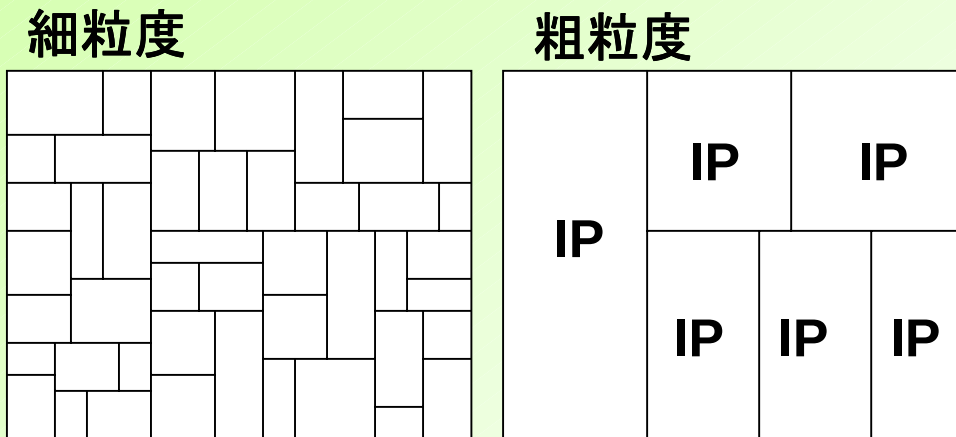
$\text{Effort} \propto \text{記述量 (Complexity)}$ (同一粒度において)

- 記述量 (設計生産性検討の仮定)

– 動作記述: RT記述 = 1 : 16

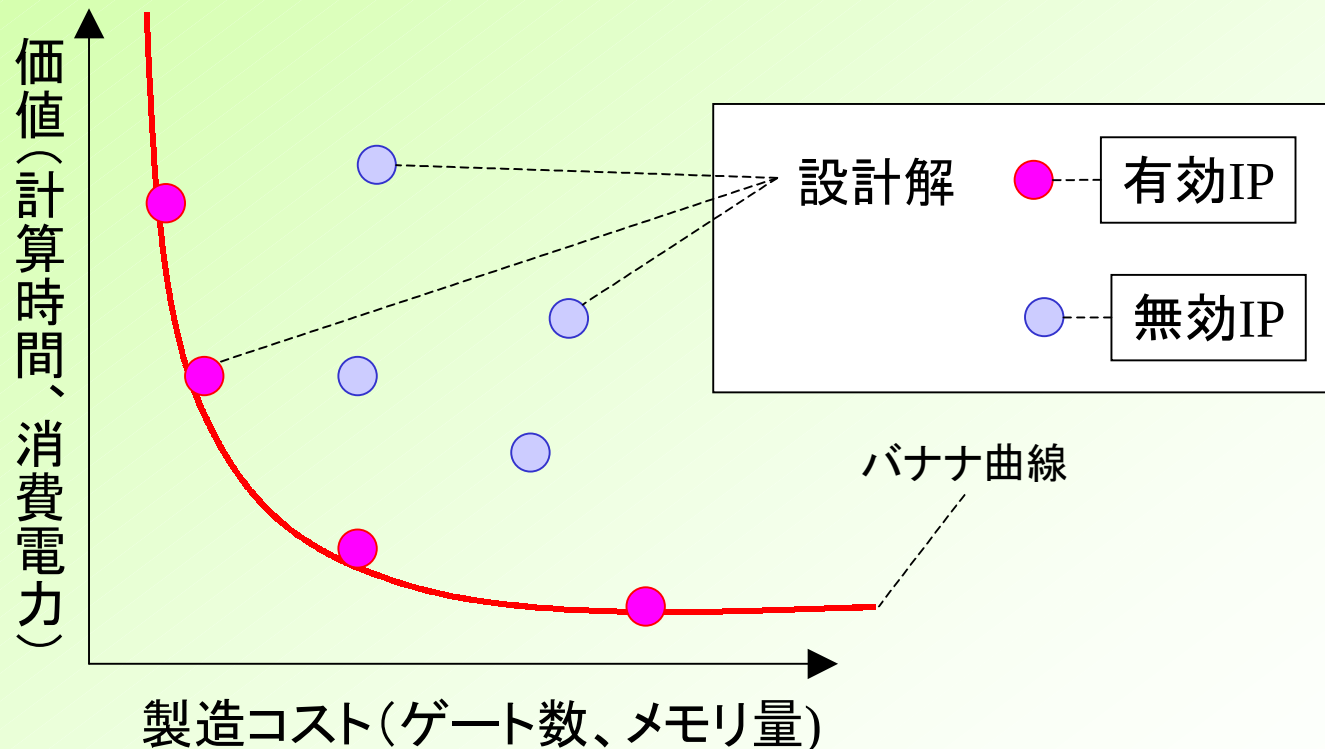
粒度の大型化

- IPは設計粒度を上げ、生産性を向上させる。
- IP流用によるEffortの線形化
 - Effortは、設計対象の規模に対して比例以上の傾きで増大する。
 - IPは設計粒度を上げ、生産性カーブを線形化する。



粒度の大型化: IPが満たすべき条件

- IPが満たすべき条件
 - 1つの機能に対して、複数のIP(設計解)を用意する必要あり。
 - 「製造コストVS価値」のバナナ曲線状にあるIPのみが有効である。
- 「IPを高抽象度で記述⇒高位合成⇒HW化」が現実解となる。



粒度の大型化:生産性向上

- 最終成果物(SoC)の定義に必要な情報量を削減され、
 - ①再利用により、記述すべき情報量が少なくなる。
 - ②一度に考える情報量が少なくなる。
- ⇒設計生産性が向上する。
- 設計生産性ロードマップにおける仮定
Effort $\propto$ 記述量(Complexity) (同一抽象度において)

高位合成ツールの性能

- 高位合成ツール
 - 高位合成（アルゴリズム⇒RTLを合成、HW/SW分割後）
 - HW/SW協調合成（HW／SW分割を含む）
 - 重要である理由
 - 市場から見た価値を決定する
 - 処理速度、消費電力
 - 製造コストを決定する
 - チップサイズ、テスト時間
 - 低レベルの技術者でもSoC設計が可能となる。
- ⇒高位合成ツールを支配する者が、SoC設計産業を支配する。

価値の高い仕様の作成

- 「設計抽象度高位化」は、
開発Effortの低減のみならず、
商品仕様レベルの価値を向上させる。
- 理由
 - 製造開始直前までの仕様変更が可能
 - Virtual Prototype Modelの顧客提供による仕様の最適化
 - 複数アーキテクチャの評価解析による最適アーキテクチャ選択
 - システム検証の早期実施・高速化による仕様ミス低減

目次

- はじめに
- 設計生産性の定義
- SoC Design Crisisの定量化
- SoC Design Crisisの解決策
- 解決策の詳細議論
- 設計生産性ロードマップ
 - Case 2: 高位合成
 - Case 3: 高位合成+HW/SW協調合成
 - 新規設計率
 - 面積比率
- 詳細設計フロー
- 提言、まとめ

Case 2: 高位合成

・ 仮定:

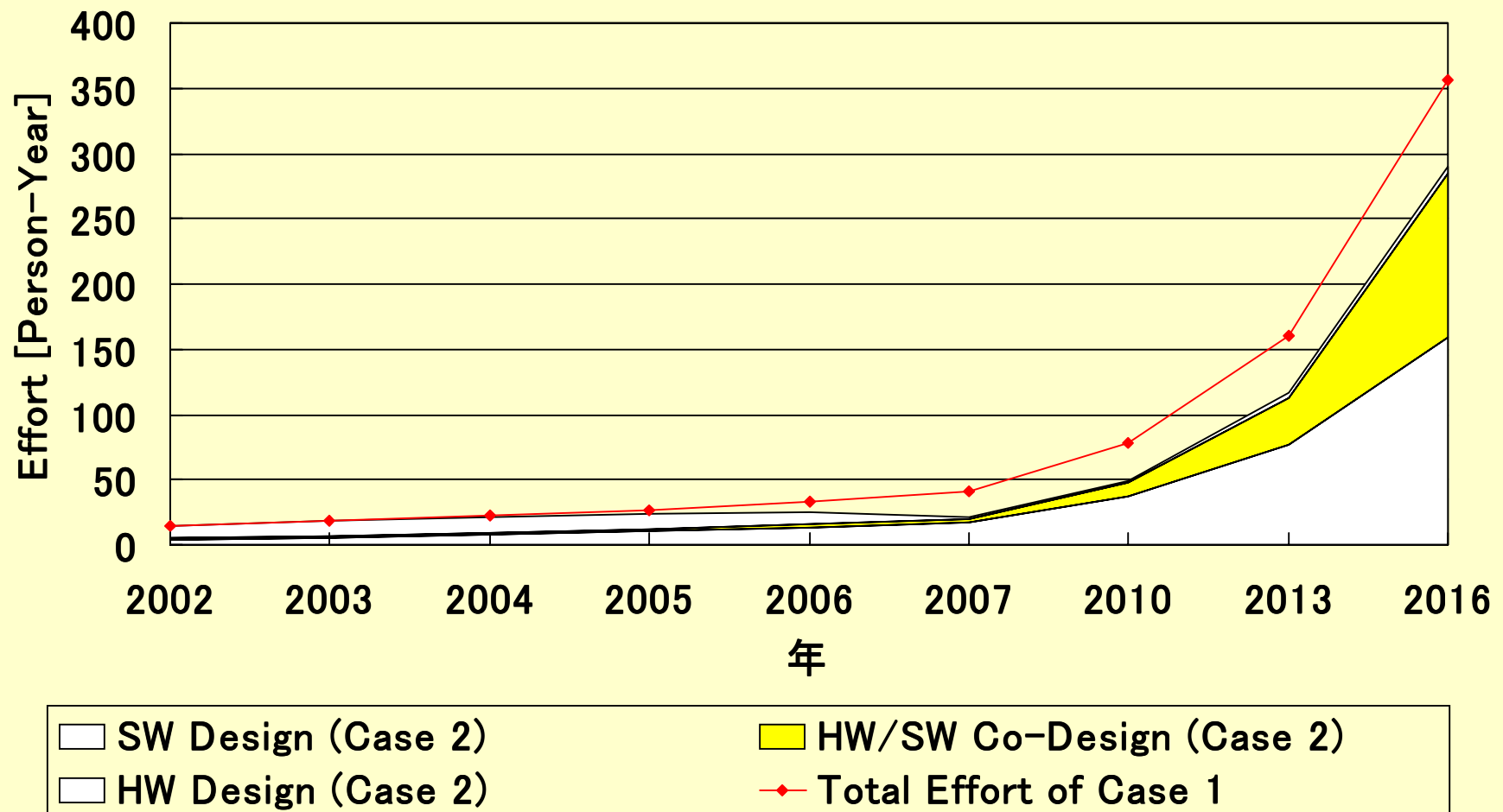
- 高位合成の利用開始時期 = 2004年
- 高位合成により、HW設計の生産性が16倍向上する。(抽象度の高度化効果)
- 2004年に専用HWの12.5%に適用する。
- 高位合成の適用率は年とともに上昇し、2007年に100%に達する。

ケース2: 高位合成が2004年に実現(変節点1)

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
Abstraction Level of High Level Design				16	16	16	16	16	16	16
Applicability of High Level Synthesis				12.5%	25.0%	50.0%	100.0%	100.0%	100.0%	100.0%
SW Design Effort	person-year			7.8	10.2	13.2	17.2	36.7	77.0	159.4
HW/SW Co-Design Effort	person-year			1.1	1.5	2.2	3.3	10.8	36.2	125.9
HW Design Effort	person-year			11.7	11.6	9.3	1.3	1.9	2.9	4.4
Total Effort	person-year			20.5	23.4	24.8	21.7	49.4	116.1	289.7

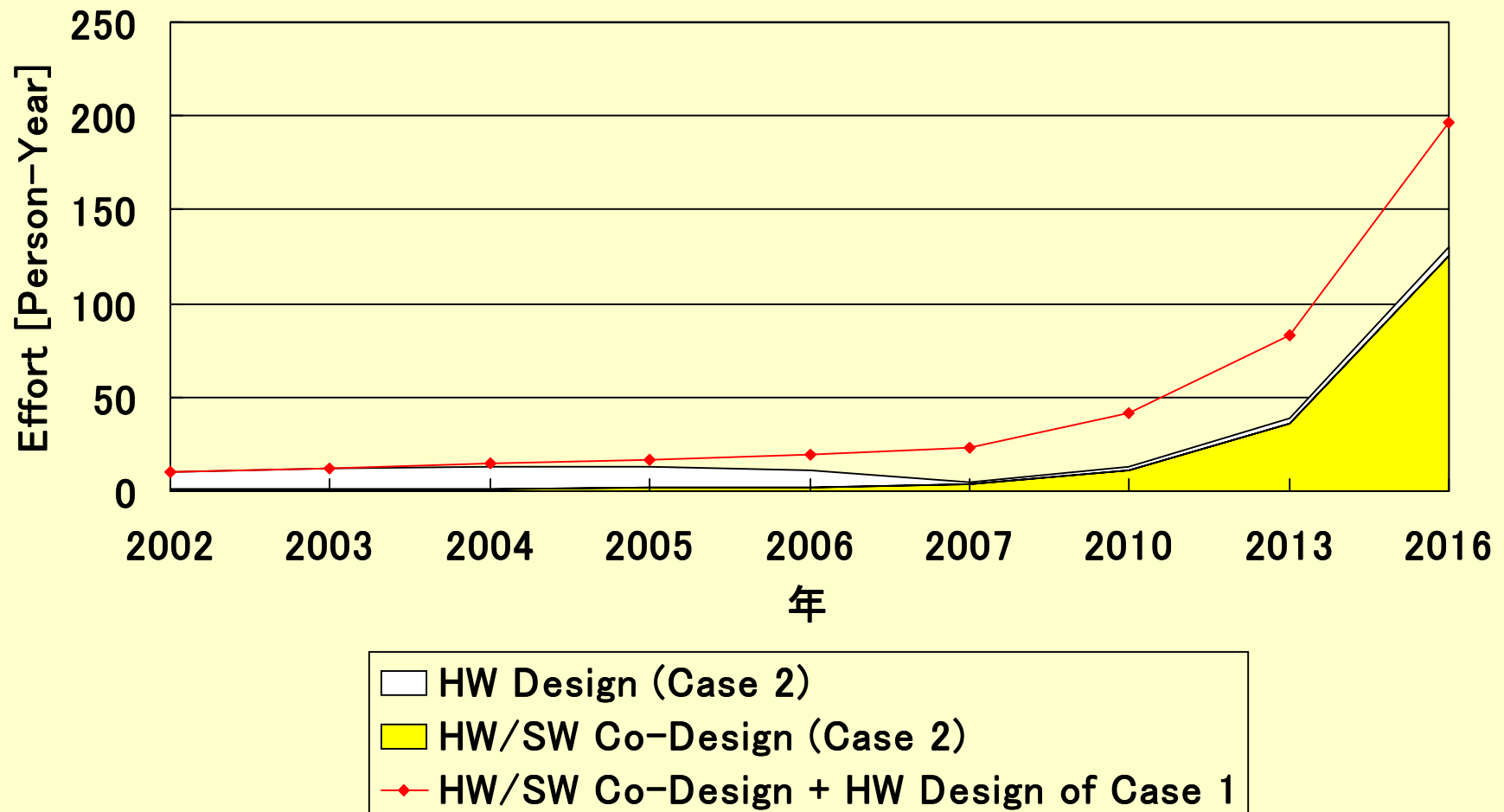
Case 2: 高位合成の導入によるEffort削減

SoC Design Effort
ケース2・高位合成導入を想定した場合



Case 2: 高位合成の導入によるEffort削減

HW/SW Co-Design + HW Design Effort
ケース2・高位合成導入を想定した場合



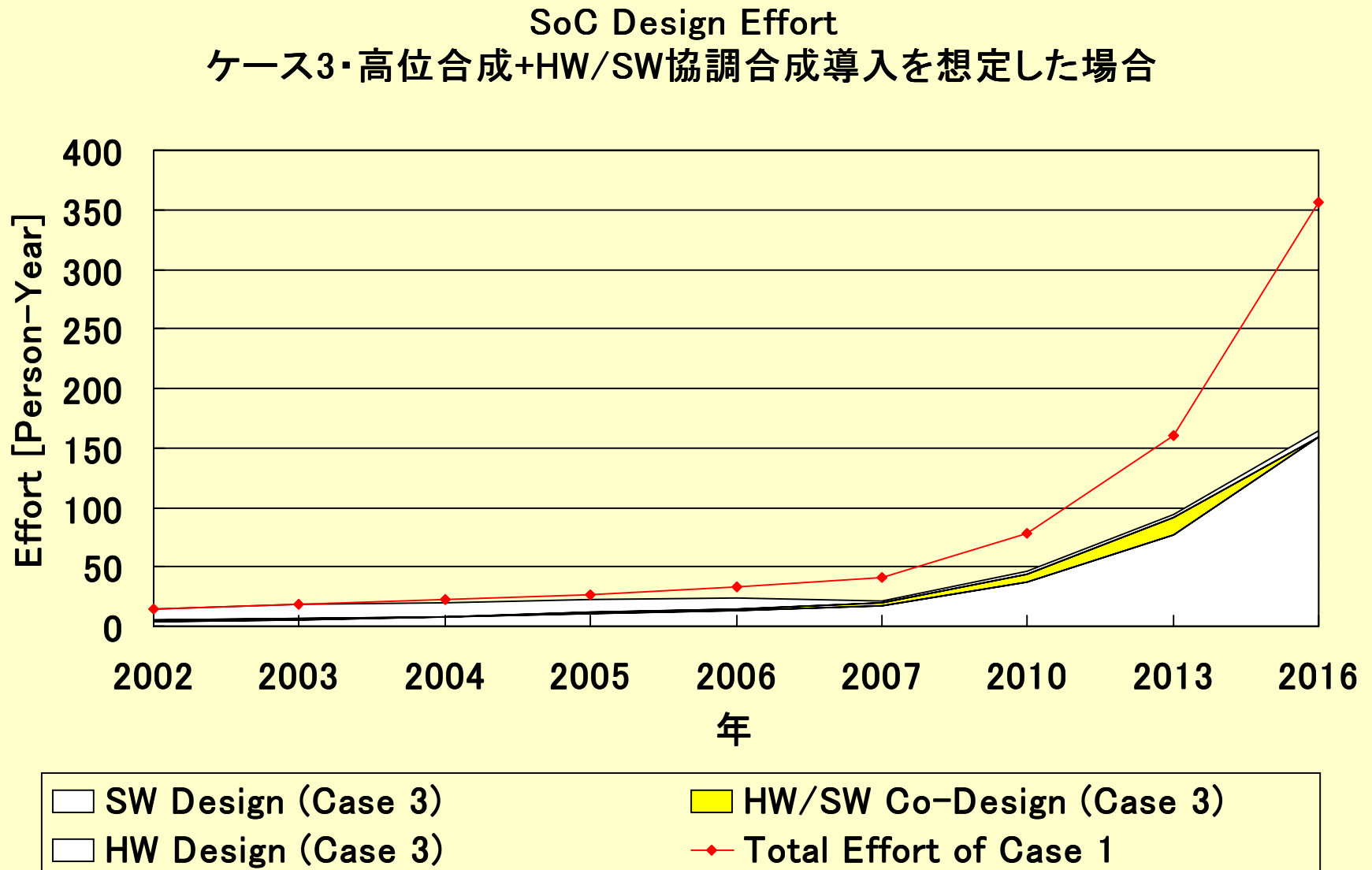
Case 3: 高位合成+HW/SW協調合成

- 仮定:
 - 2006年から導入開始
 - 導入開始時のEffort削減効果は10%
 - 2016年にはHW/SW協調合成の高性能化により、
HW/SW分割に関わるEffortは、総Effortに比べ無視可能な量に低下

ケース3: HW/SW協調合成が2008年に実現(変節点2)

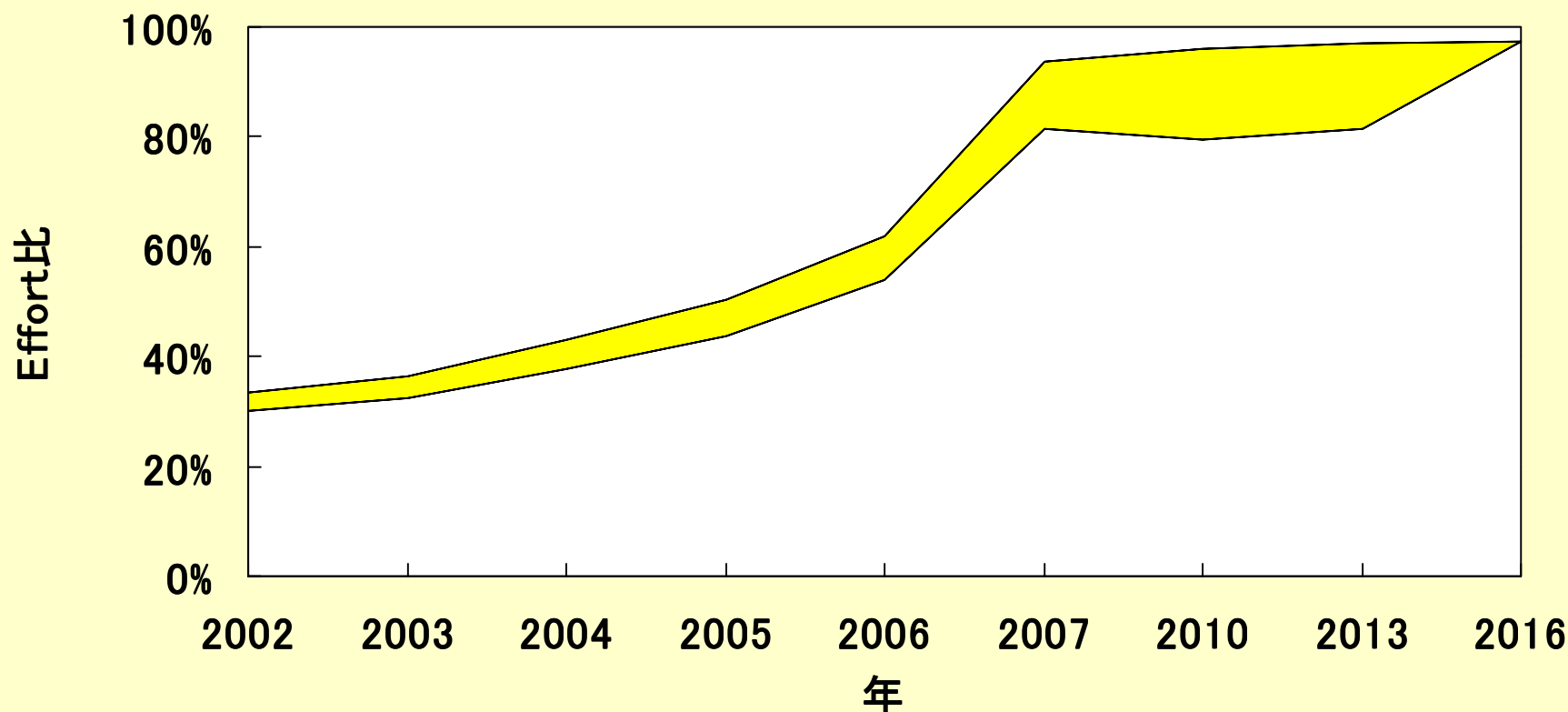
	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
Effort Reduction by HW/SW co-Synthesis						10.0%	20.0%	30.0%	60.0%	100.0%
SW Design Effort	person-year					13.2	17.2	36.7	77.0	159.4
HW/SW Co-Design Effort	person-year					2.0	2.6	7.5	14.5	0.0
HW Design Effort	person-year					9.3	1.3	1.9	2.9	4.4
Total Effort	person-year					24.6	21.1	46.1	94.4	163.9

Case 3: 高位合成+HW/SW協調合成によるEffort削減



Case 3: 高位合成+HW/SW協調合成によるEffort削減

SoC Design Effort
ケース3・高位合成+HW/SW協調合成導入を想定した場合



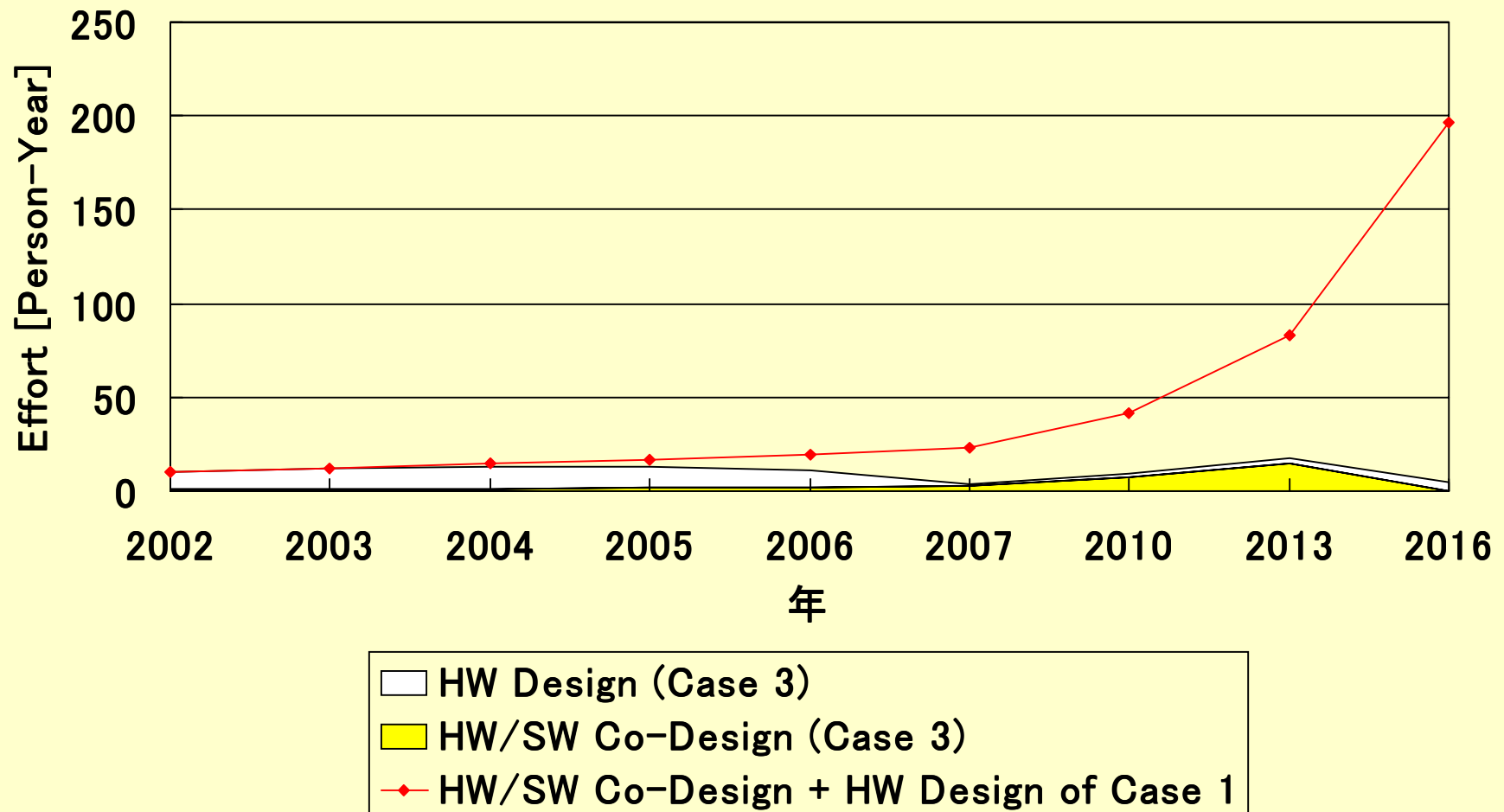
□ SW Design (Case 3)

■ HW/SW Co-Design (Case 3)

□ HW Design (Case 3)

Case 3: 高位合成+HW/SW協調合成によるEffort削減

HW/SW Co-Design + HW Design Effort
ケース3・高位合成+HW/SW協調合成導入を想定した場合

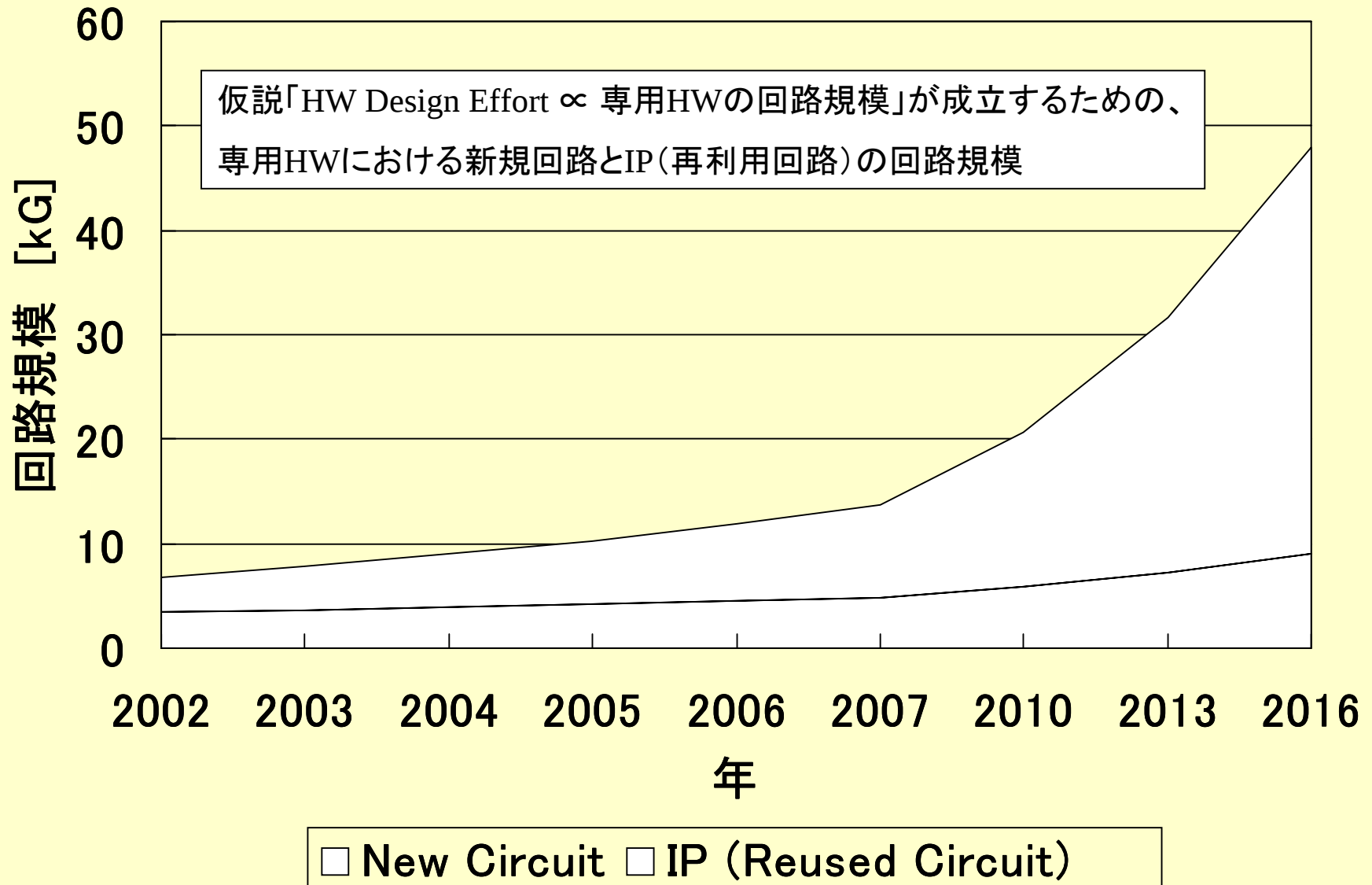


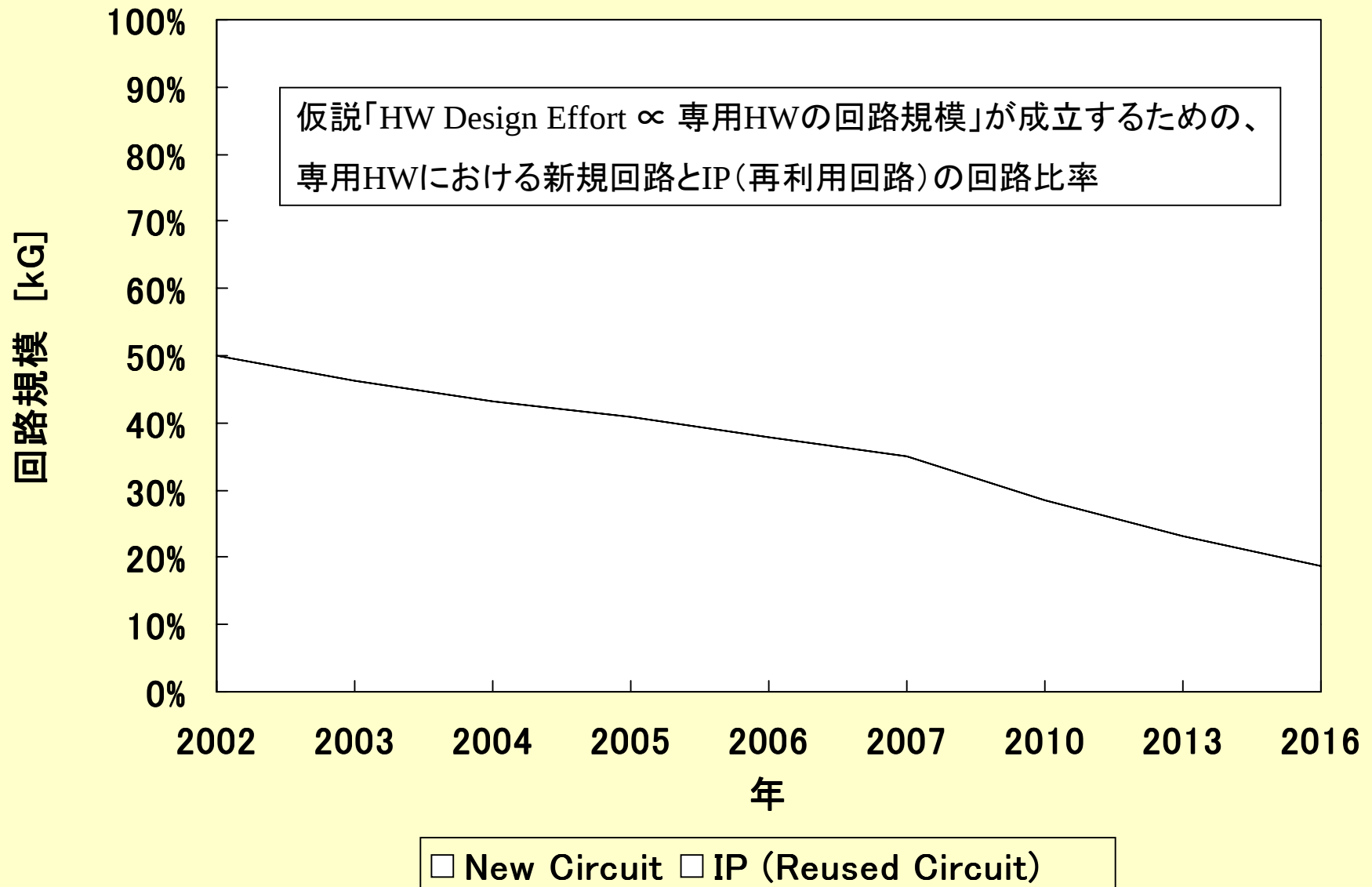
仮説「HW Design Effort $\propto$ 専用HW規模」の確認

- 目的
 - 仮説「HW Design Effort $\propto$ 専用HWの回路規模」の前提は、IP再利用の促進
⇒ 本仮定が成立するIP再利用率を検証
- 算出の仮定：
 - HW設計Effort：
 - 専用ロジック部における新規設計回路の規模に対して2乗で増加
 - 条件：
 - HW設計Effortは、専用ロジック部の回路規模に比例
 - IP再利用に要するEffortは、無視可能

専用ロジック(Special Digital Circuit)のIP化率

	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
Ratio of New Design		50%	47%	43%	41%	38%	35%	29%	23%	19%
Ratio of Reused Design(IP)		50%	53%	57%	59%	62%	65%	71%	77%	81%
Logic Size of New Design	M gate	3.4	3.6	3.9	4.2	4.5	4.8	5.9	7.3	9.0
Logic Size of Reused Design	M gate	3.4	4.2	5.1	6.1	7.4	8.9	14.8	24.3	39.0
HW Design Effort with IP	person-year	10.0	11.5	13.2	15.2	17.5	20.1	30.6	46.5	70.6
HW Design Effort w/o IP	person-year	10.0	13.2	17.5	23.1	30.6	40.5	93.5	215.9	498.9

仮説「HW Design Effort $\propto$ 専用HW規模」の成立条件

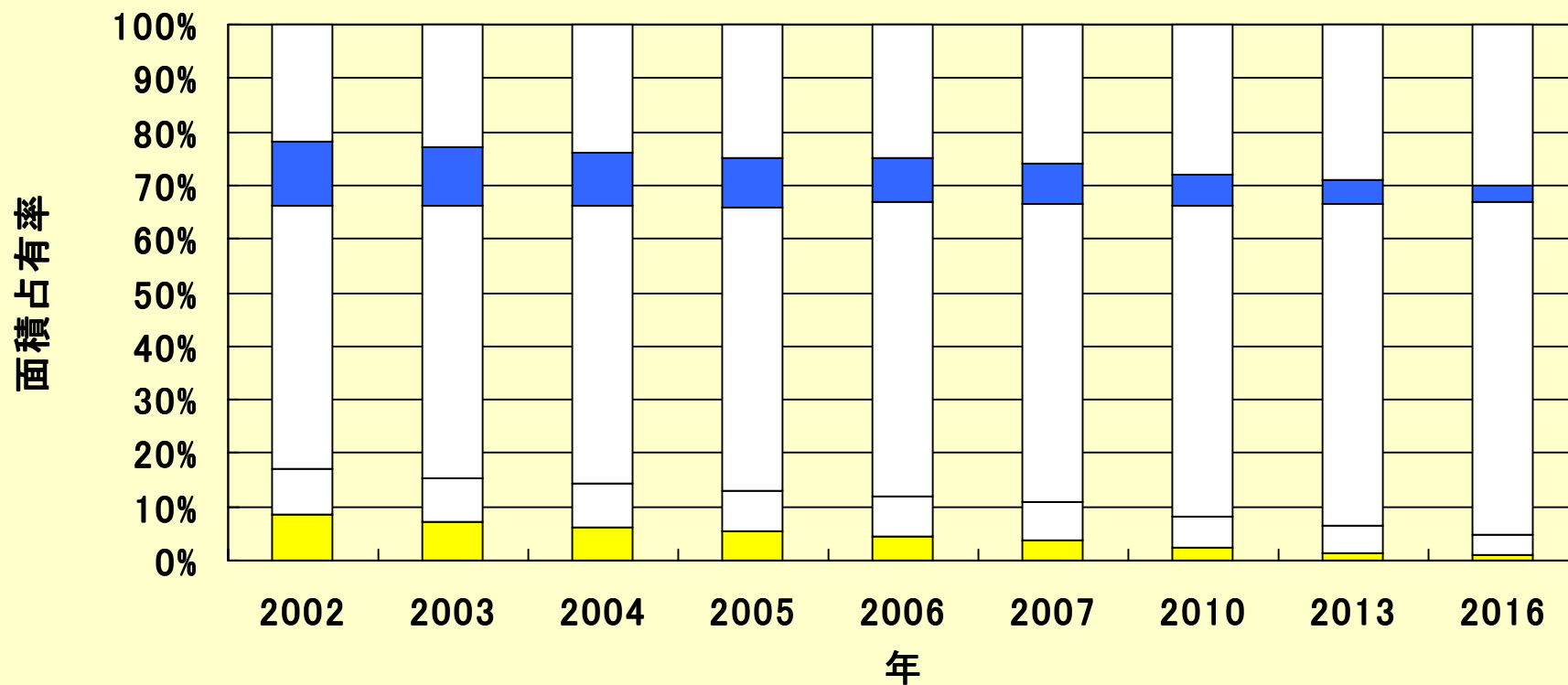
仮説「HW Design Effort $\propto$ 専用HW規模」の成立条件

面積比率

面積比										
	Unit	2002	2003	2004	2005	2006	2007	2010	2013	2016
Technology	nm	130	107	90	80	70	65	45	32	22
CPUブロック	%	34%	34%	34%	34%	34%	34%	34%	34%	34%
ロジック	%	12%	11%	10%	9.0%	8.2%	7.5%	5.7%	4.3%	3.3%
キャッシュメモリ	%	22%	23%	24%	25%	25%	26%	28%	29%	30%
組み込み論理ブロック	%	66%	66%	66%	66%	66%	66%	66%	66%	66%
ロジック	%	17%	16%	14%	13%	12%	11%	8.2%	6.3%	4.8%
ロジック・新規設計	%	8.6%	7.3%	6.2%	5.3%	4.5%	3.8%	2.4%	1.5%	0.9%
ロジック・再利用	%	8.6%	8.3%	8.1%	7.7%	7.4%	7.0%	5.9%	4.8%	3.9%
SRAM	%	49%	51%	52%	53%	55%	56%	58%	60%	62%

- Chip Size
 - Low Power SoCモデルを使用
- ASIC usable M transistors/cm²
 - トランジスタ密度、出典:ITRS2002
- Area Ratio of Special Digital Circuit/Chip
 - 専用ロジック占めるチップ面積率
- Area Ratio of New Design/Chip
 - 新規に設計するロジック回路が占めるチップ面積率
- Area Ratio of Reused Design/Chip
 - 専用ロジック内の再利用回路が占めるチップ面積
- Area Ratio of Processors+Memory
 - Processor(複数)およびProcessor用メモリが占める面積

チップ面積占有率



- 専用デジタルブロック内・新規設計ロジック
- 専用デジタルブロック内・再利用ロジック
- 専用デジタルブロック内SRAM
- プロセッサ領域内ロジック
- プロセッサ領域内キャッシュメモリ

目次

- はじめに
- 設計生産性の定義
- SoC Design Crisisの定量化
- SoC Design Crisisの解決策
- 解決策の詳細議論
- 設計生産性ロードマップ
- **詳細設計フロー**
- 提言、まとめ

詳細設計フロー

- 以降に、
 - 各工程内の詳細フローを示す。

システム要求仕様定義

サブ・タスク

設計費用とその責任範囲

デバイス費用とその責任範囲

顧客要求機能、性能

実現機能、性能の新規性

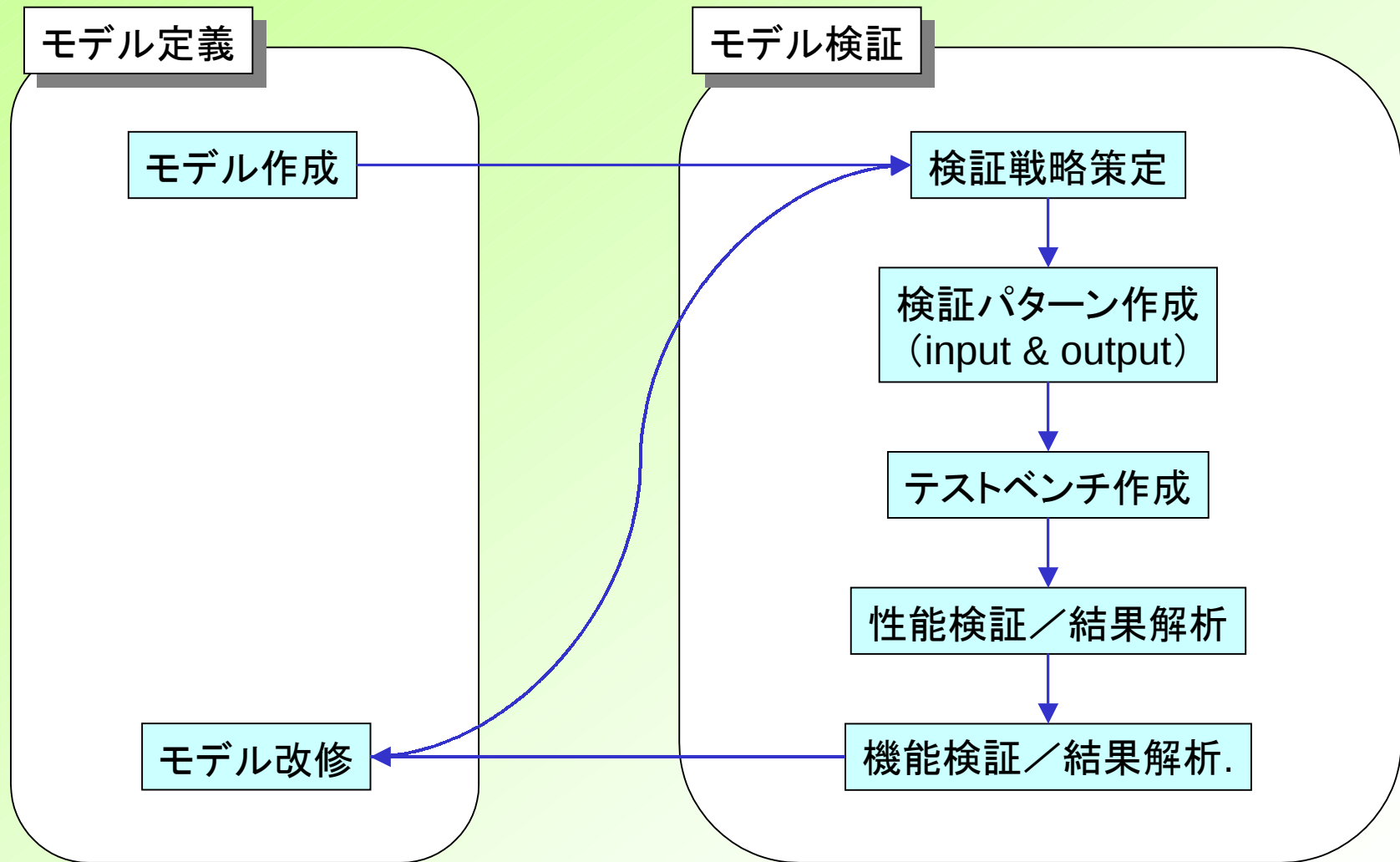
仕様設計は誰がおこなうのか

全設計工程の戦略がねられているか

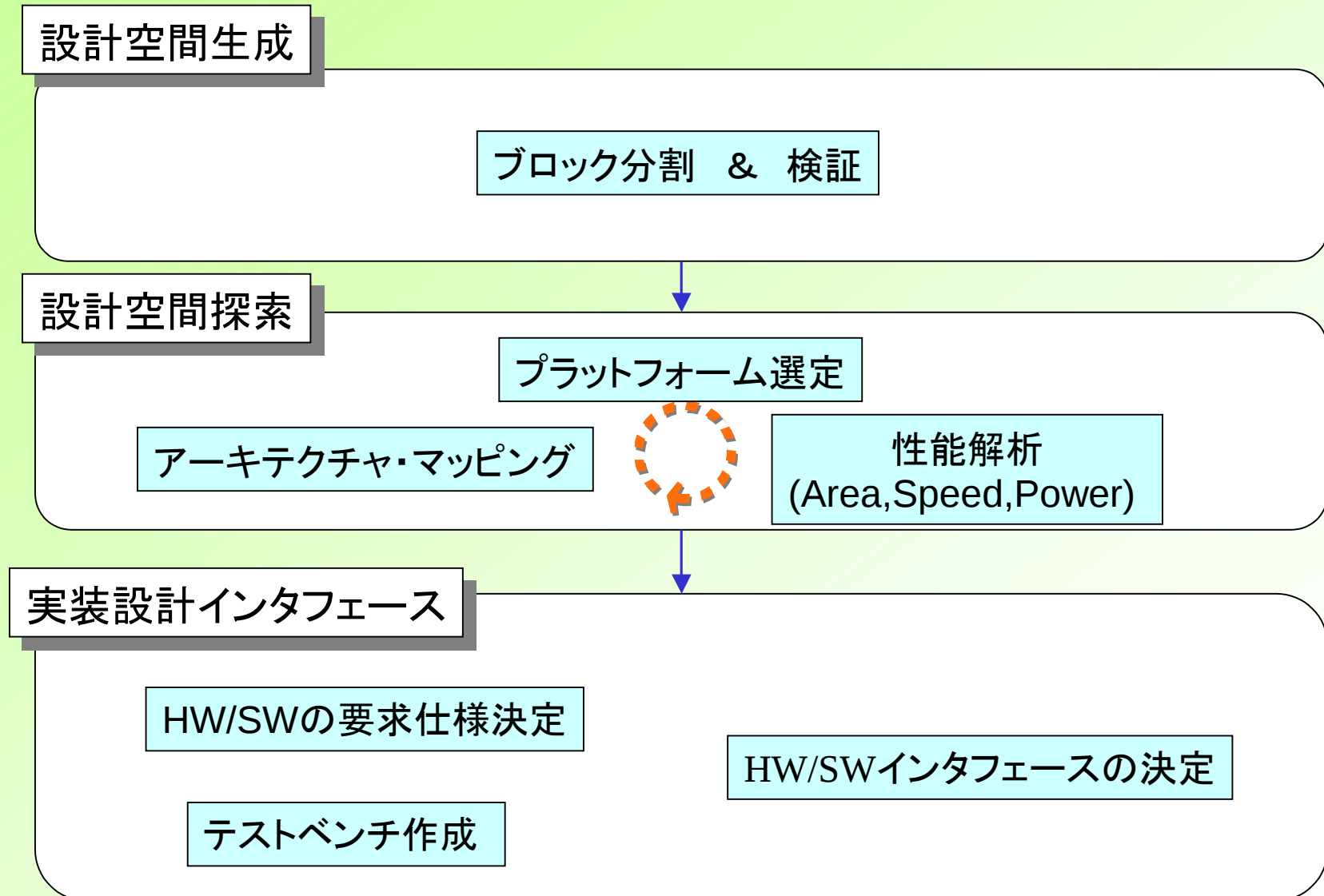
設計リーダーは経験豊富で 信頼されているか

デバイステストの実施者とその責任範囲

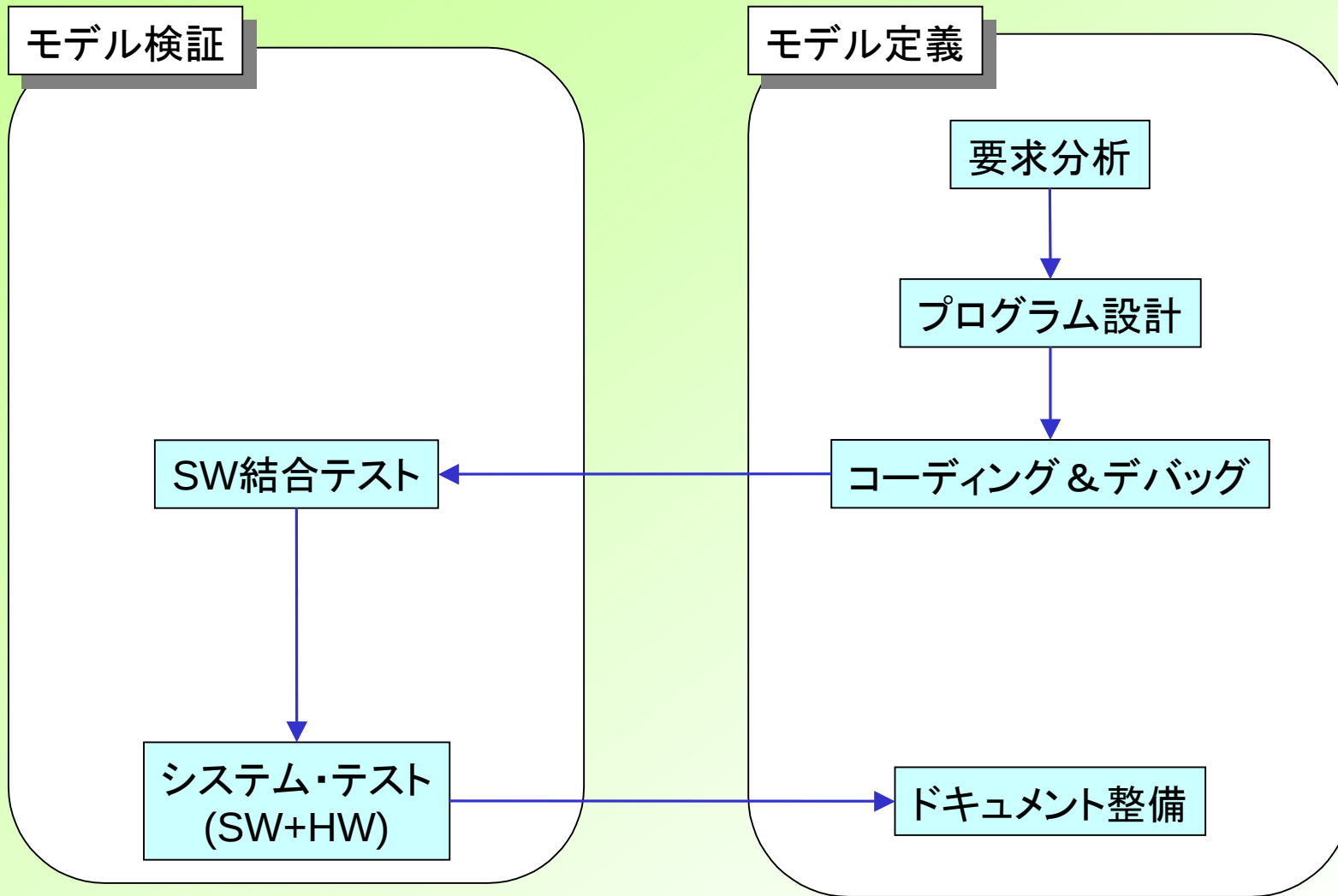
システム機能決定



システム・アーキテクチャ決定



ソフトウェア設計



ハードウェア設計

HWマイクロアーキテクチャ決定

仕様の理解

ブロック分割

ブロック間インタフェース設計

テスト・検証方法決定

アルゴリズム設計

アルゴリズム検証

動作モデル設計

動作モデル検証

RTL決定(高位合成)

RTL決定

RTLモデル検証

面積／消費電力見積り

論理合成 & 配置配線

論理合成

フロアプラン

クロック／電源系設計

配置配線

目次

- はじめに
- 設計生産性の定義
- SoC Design Crisisの定量化
- SoC Design Crisisの解決策
- 解決策の詳細議論
- 設計生産性ロードマップ
- 詳細設計フロー
- 分析結果、まとめ、提言

分析結果

- 高位合成、HW／SW協調合成
 - Design Crisis回避に必須の技術
- SW Design Crisis
 - 2007年以降は、SoC Design Effortの80%以上を占有
 - SW設計Crisisが、今後の最大脅威となる
- プラットフォーム
 - 2007年以降は、チップ面積の70%以上を占有
 - 複数の標準ProcessorsとMemoryで構成
 - Design Crisisの回避に貢献するが、極論すれば「単なる機能のSWシフト」
- IP再利用の推進
 - Design Crisis回避に必須の技術
 - 2010年以後、専用ロジックの70%以上を占める。
 - ただし、2010年における新規ロジック回路の規模は、4. 8Mゲートと巨大

まとめ

- 設計生産性を定義し、Design Crisisを定量的に分析
 - 分析範囲: SoCシステム設計～レイアウト設計
 - 対象商品: STRJ設計TFのPDA用Low Power SoCモデル
- Design Crisisの解決に必要な技術を定義し、期待効果を定量的に評価
 - 必要技術の定義: 高位合成、HW／SW協調合成
 - 両技術の導入による設計フローの変遷を明確化

提言

- 官学産の総力を結集し、
 - 高位合成、HW／SW協調合成技術で、日本が支配的立場を確立
 - 両技術を支配する者が、SoC設計産業を支配
 - Design Crisis回避に必須の技術
 - SoCにおけるソフトウェア設計危機解消の技術開発
 - 次世代プラットフォームで、日本が支配的立場を確立
 - マルチ・プロセッサ型プラットフォーム
 - Processor(複数) + Memoryがチップ面積を占拠
 - ソフトウェアIPの充実: Processorデファクト化の鍵
 - 低消費電力: HW-SW協調による低消費電力を実現
 - IP再利用の推進は
 - Design Crisis回避に必須の技術
 - 2010年以後、新規ロジック回路がチップ面積を占有する比率は、5%以下に減少
 - 高位構成、HW／SW協調合成にマッチし、最適解が得られるIP記述

補足： 物理設計技術について

- 半導体のスケーリングが続く限り、

物理設計は、LSI設計産業の最重要課題の1つである。

本報告は、物理設計技術が期待通りに高度化し、設計危機を生じないとの前提で検討している。

これは、あくまでも仮定であり、物理設計技術の重要性は極めて高いと認識している。

- 物理設計技術 = レイアウト設計、タイミング・クロージャ、Signal Integrity

用語

- 定義 (Modeling)
- 検証 (Verification)
- 決定 (Creation)
- 設計 (Design)

参考文献

- [1]樋渡、田中、デザインウェーブ????
- [2]EDA技術専門委員会・SLD研究会、「システムレベル設計における標準化団体と研究期間に関する動向調査」、EDAアニュアルレポート 2001、(社)電子情報技術産業協会

思案中の資料

はじめに(参考資料)

- SW

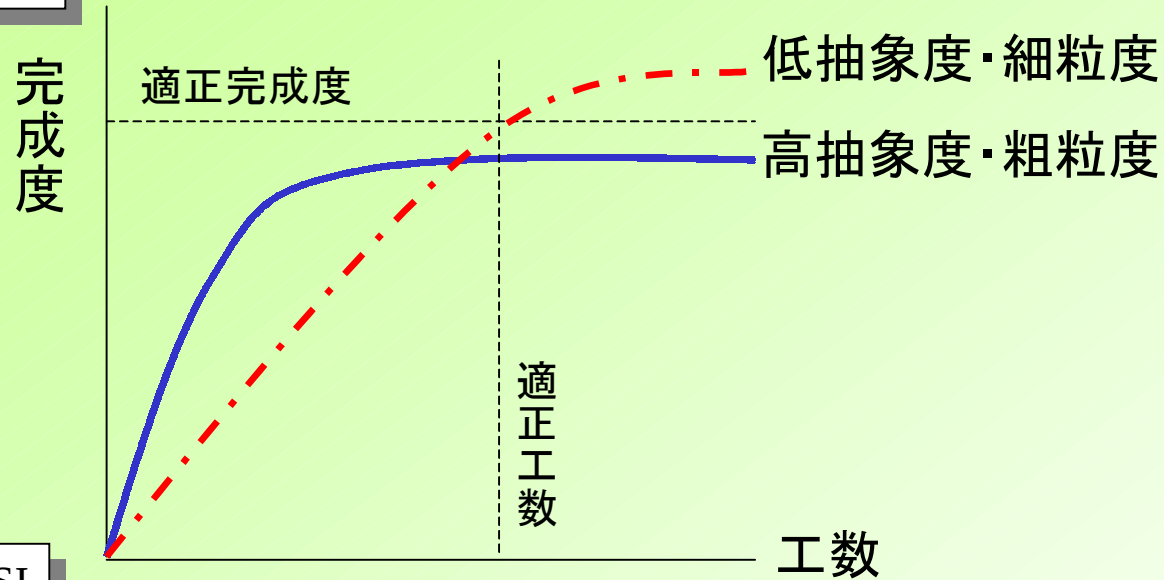
- 太田博之「W-CDMA端末の最新設計手法」日経エレクトロニクス1999.4.5(no.740),p203
- 日経エレクトロニクス編集部「ソフトウェアが携帯端末を情報端末に変える」日経エレクトロニクス1999.7.12(no.747),p163
- 日経エレクトロニクス編集部「特集:ソフト危機デジタル家電開発が水へ分業へ」日経エレクトロニクス1998.3.23(no.712),p141

プラットフォーム(標準アーキテクチャ)

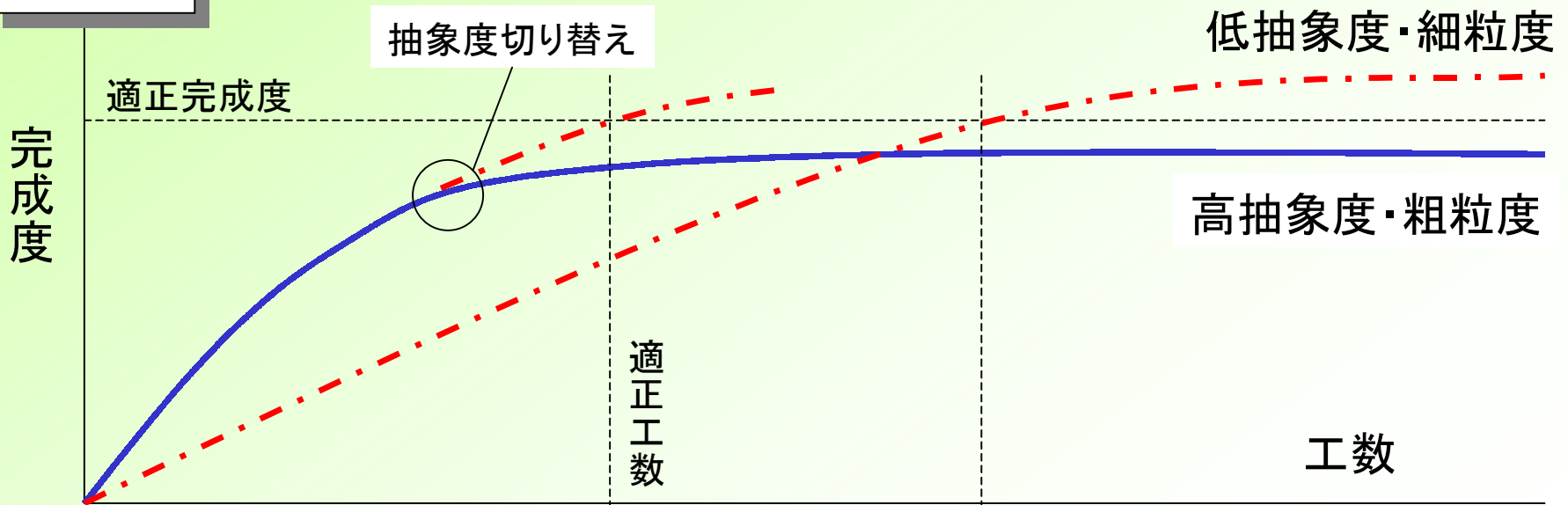
- ①プラットフォーム(標準アーキテクチャ)
 - プラットフォームは、提供される。
 - プラットフォームとは、CPUにキャッシュ・メモリ等を追加した、RTOSを搭載するためのプリミティブな機能である。
 - アプリケーション回路とのインタフェース)は標準化され、プラットフォームとして提供される。
 - 使用するプラットフォーム(標準アーキテクチャ)は、S3(H/S分割・検証)タスクにて選択する。
- ②ドライバおよびミドルウェア
 - プラットフォームの構成要素として提供される。
- ③RTOS
 - プラットフォームに対応して開発済みであるとする。
- ④ソフトウェア開発環境(コンパイラ、デバッガ、...)
 - ソフトウェア開発環境は、標準アーキテクチャ(プラットフォーム)に対応して開発済みであるとする。

メソドロジの移行タイミング

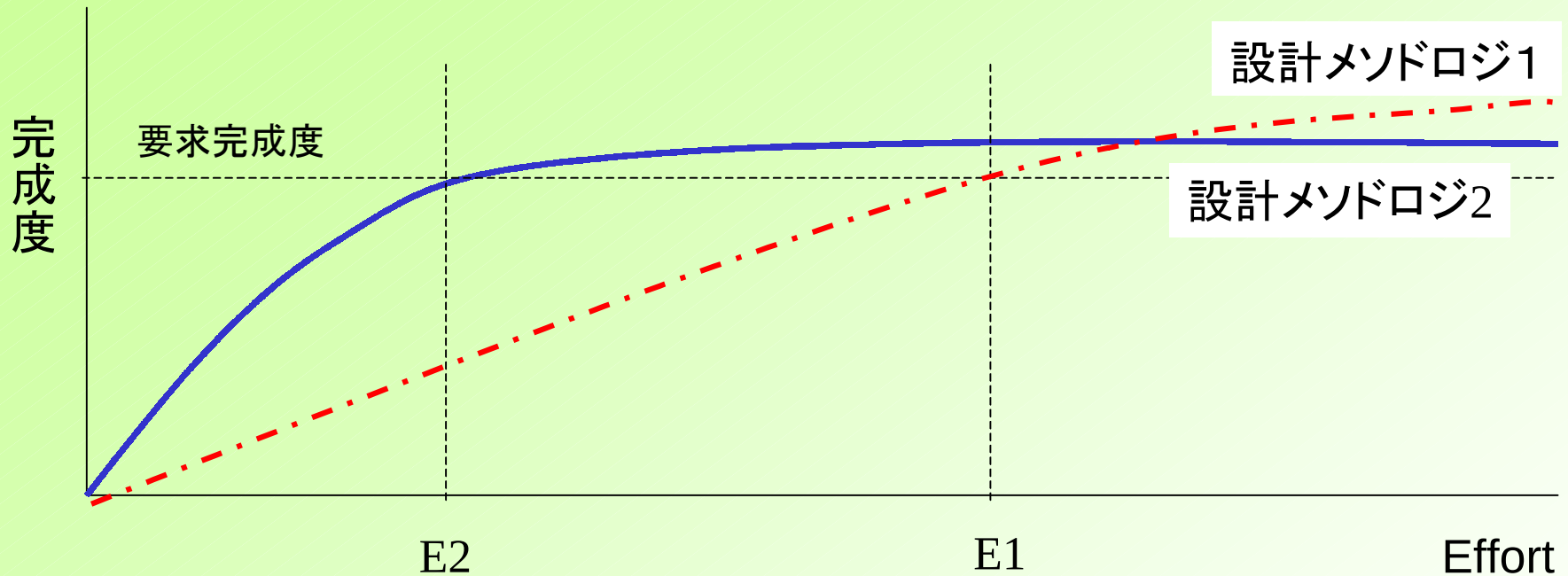
小規模LSI



大規模LSI



設計生産性の向上率



設計生産性は、同一の完成度に対して比較する。
 $E1 / E2$ が、設計生産性の向上率を意味する