

第4回 システムLSI琵琶湖ワークショップ

ニーズ/シーズ調査に基づく
システム設計手法の提案と
システム記述言語の適用化検討

2000年11月29日

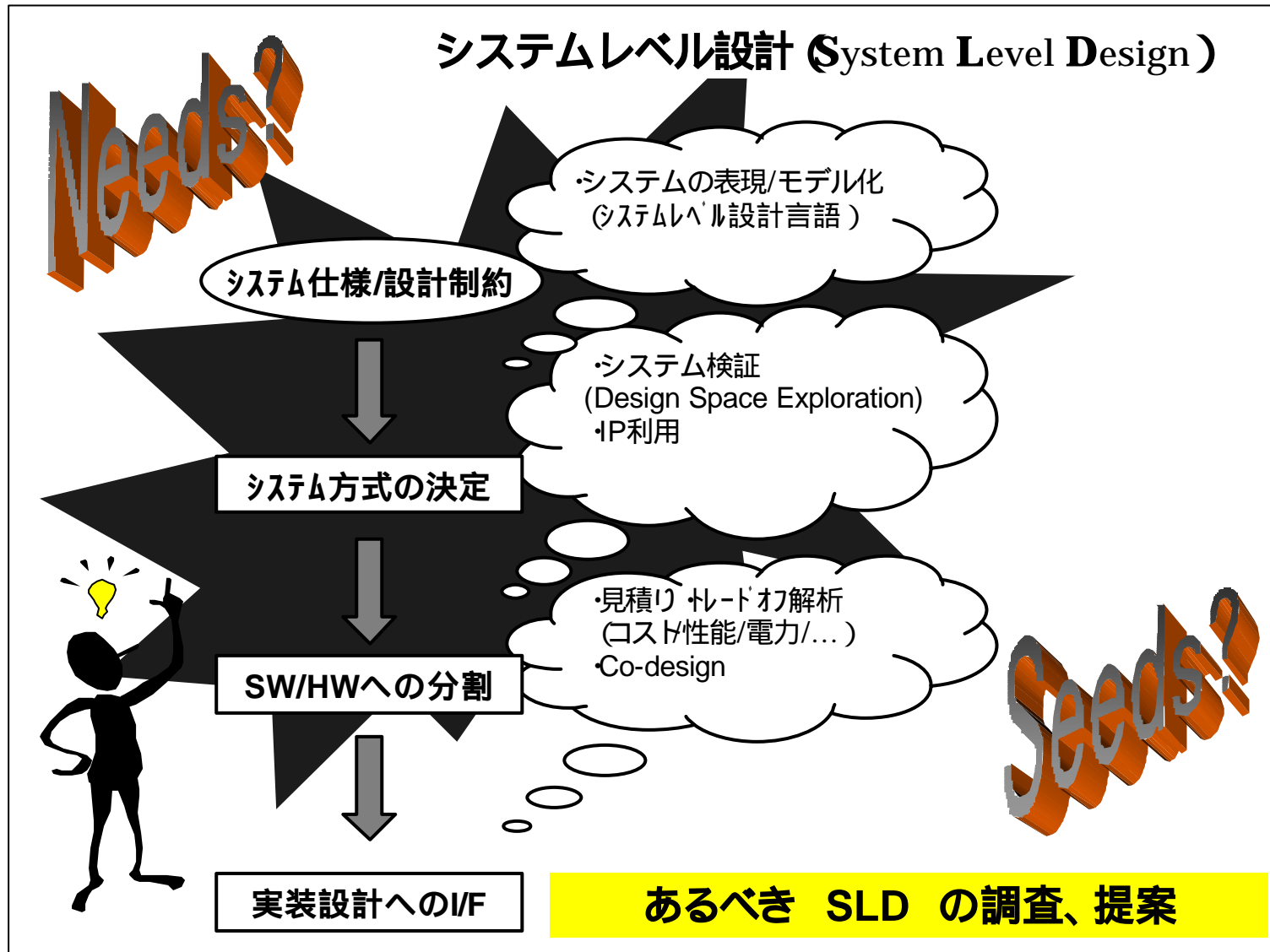
JEITA EDA技術専門委員会 SLD研究会
古知屋正樹 (ソニー株式会社)



アジェンダ

- SLD研究会活動の紹介
- 設計者ニーズの調査
- シーズンとしてのシステム設計技術
- 提案するシステム設計手法
- システム設計言語調査と提案手法への適用
評価
- まとめ

SLD研究会活動の紹介



【発足】
EIAJ EDA
技術委員会
の下部組織
1998/11
現在、21名

【目的】
ニーズ、シーズ
の調査を行ない
あるべきSLDを
提案



SLD研究会メンバー

| | | |
|-----|--------|-------------|
| 主査 | 安田 光宏 | (三菱電機) |
| 副主査 | 黒坂 均 | (NEC) |
| 副主査 | 山口 雅之 | (シャープ) |
| 委員 | 宇部 努 | (イノテック) |
| | 松村 浩二 | (三洋電機) |
| | 古知屋 正樹 | (ソニー) |
| | 荒木 大 | (東芝) |
| | 茂木 浩介 | (凸版印刷) |
| | 栗原 郁夫 | (シノプシス) |
| | 小林 和彦 | (日立) |
| | 山下 智規 | (富士通) |
| | 大塚 正人 | (富士通) |
| | 竹村 和祥 | (松下) |
| | 奥内 康議 | (丸紅ソリューション) |
| | 堺 宏明 | (三菱電機) |
| | 温 兆祺 | (メンター) |
| 客員 | 吉田 紀彦 | (長崎大学) |
| | 橘 昌良 | (高知工科大学) |
| | 今井 正治 | (大阪大学) |
| | 吉田 憲司 | (STARC) |
| | 小澤 時典 | (STARC) |

(計21名)

活動の足跡

| 主な活動内容 | 1998年度 | 1999年度 | 2000年度 |
|------------|---------------|---------------------------------|--------------------------------|
| 標準化動向調査 | | SLDL(Rosetta)調査 システムレベル言語調査 | 標準化言語調査 (SystemC、SpecC、UML) |
| ニーズ調査 | ▲ 設計者技術懇談会 | ▲ 設計者アンケート実施 (14社163件の回答) | ▲ ニーズ分析 |
| シーズ調査 | | 産学のツール調査 | |
| 設計システム像の検討 | | システム提案 | 改善提案 標準化言語の適用評価 |
| | | ▲ SLD会議(ASP-DAC) | ▲ 琵琶湖WS |

EIAJ/SLD研究会URL: <http://eda.ics.es.osaka-u.ac.jp/jeita/eda/english/project/sld/index.html>

出版物: 『EDAアニュアルレポート1998』、『EDAアニュアルレポート1999』EIAJ/EDA技術委員会編

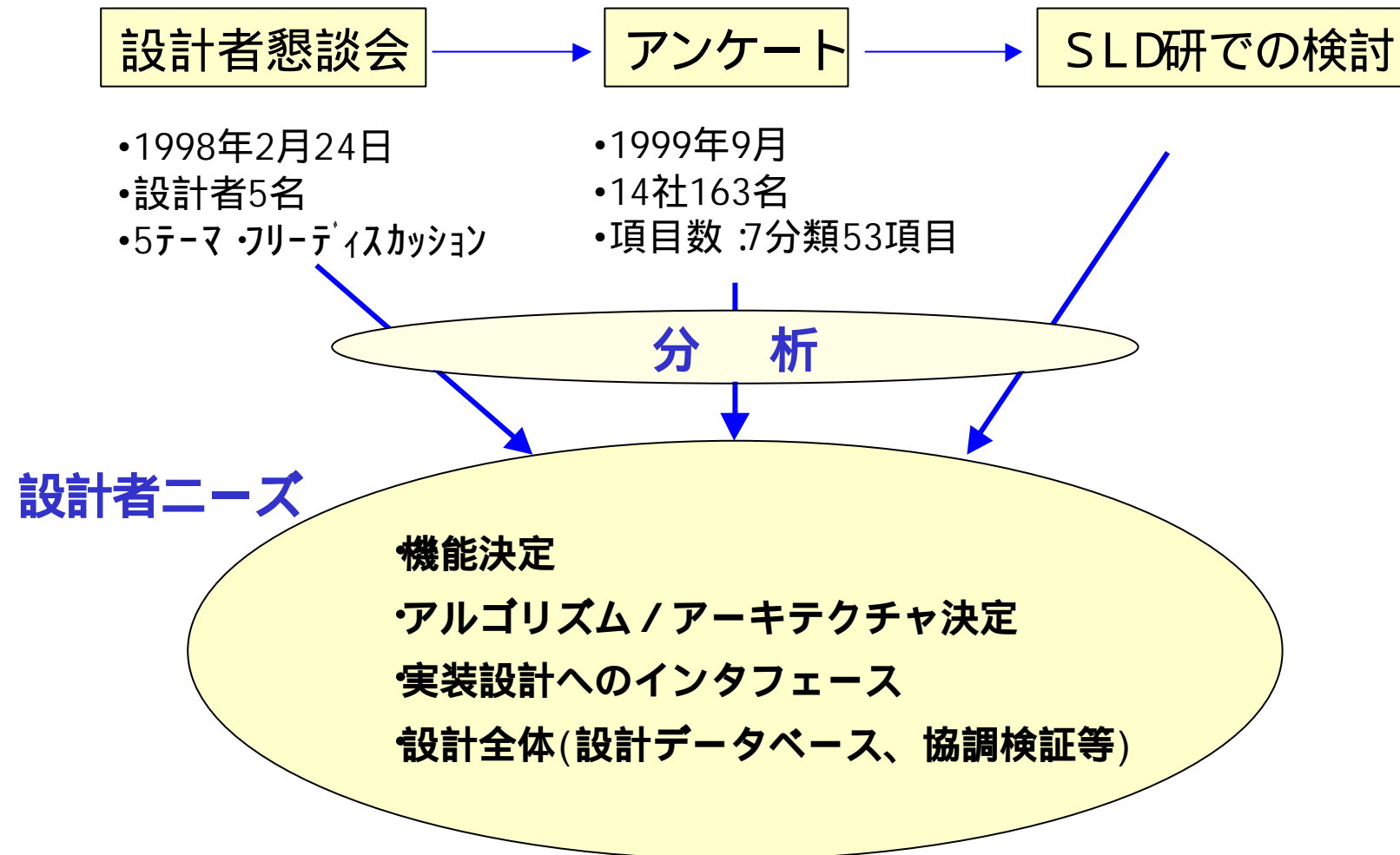


アジェンダ

- SLD研究会活動の紹介
- **設計者ニーズの調査**
- シーズとしてのシステム設計技術
- 提案するシステム設計手法
- システム設計言語調査と提案手法への適用
評価
- まとめ

設計者ニーズの調査

■ ニーズ調査方法

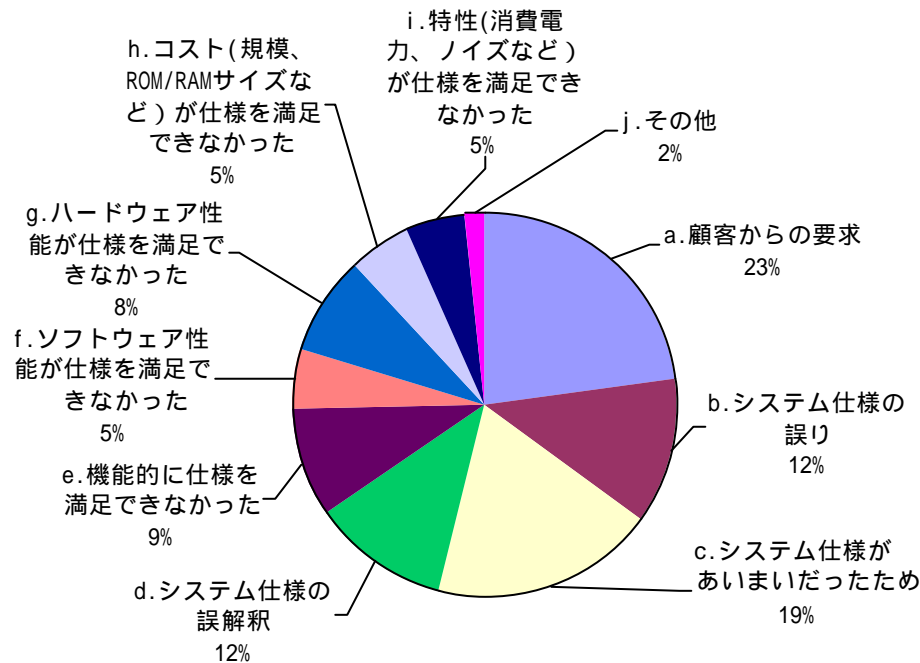


設計者懇談会



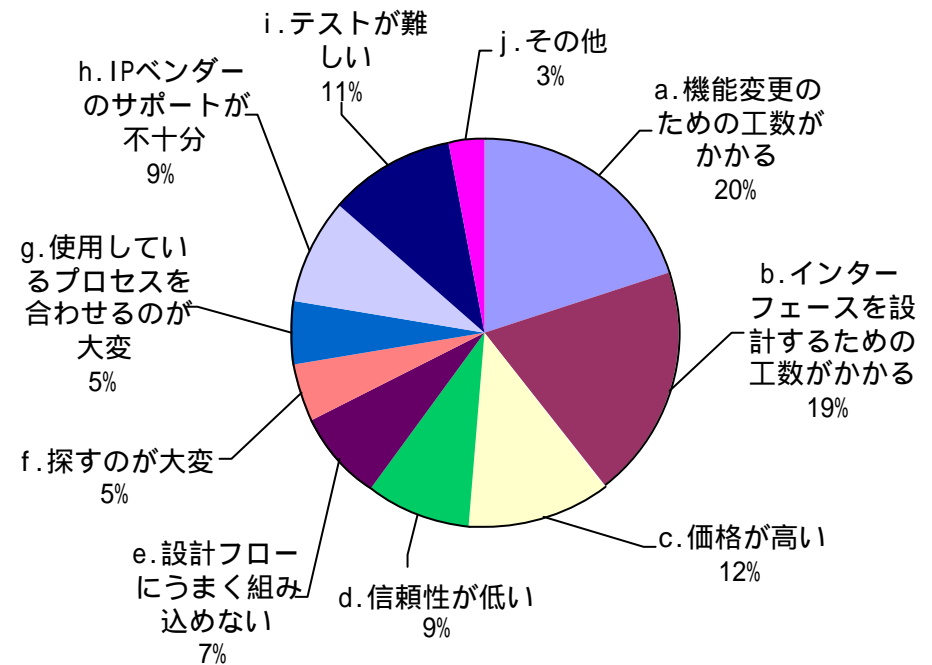
設計者アンケート

システム仕様変更が発生する主な原因



- ・ 仕様の誤り、曖昧さ及び誤解釈 43%
- ・ 仕様を満足できない 32%

IP使用時の問題点



- ・ 機能変更等のカスタマイズが必要 44%
- ・ テストが難しい 11%
- ・ 設計フローへの組み込みが困難 7%

ニーズから抽出した課題

1. 機能決定

- 仕様が曖昧なために、誤解釈し仕様と合わない物を作ってしまう。
- 仕様が未確定な状態で開発を開始しなければならず、確定した時点でリスピンが発生する。
- 機能の検証をリーズナブルな時間でできない。

2. アーキテクチャ決定

- アーキテクチャを机上で決定しており、その妥当性を確かめる手段が無い。
- 見積りが甘く、後工程からのリスピンが起こる。

設計者ニーズ

3. 実装設計へのインターフェース

- 有効なツールがなく、つなげるときの工数が大きい。
- 設計結果が仕様と合わない。

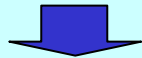
4. 設計全体 (設計データベース、協調検証等)

- IP組込みやカスタマイズに時間がかかる。
- ソフトウェア開発における協調検証シミュレータの性能不足。

解決策

1. 機能決定

仕様の曖昧さ除去 / 未確定部分の許容、及び早期機能検証を可能とするために...



機能仕様を定義する為の方法論と
実行可能なシステムレベル言語の導入

2. アーキテクチャ決定

早期に、性能、コスト等の見積りを可能とするために...



最適アーキテクチャを高速探索する
仕組み (設計フロー / 技術) の実現
・見積りモデル / 過去の設計事例データベースの構築

設計者ニーズ

3. 実装設計へのインターフェース

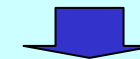
スムーズに実装設計へつなげるために...



HW/SWインターフェース合成技術と
テストベンチによる合致性検証
技術の導入

4. 設計全体 (設計データベース、協調検証等)

IPの容易な利用、高速な協調検証のために...



IPのI/Fの標準化、IP合成技術の導入、機能 ~ 実装までのIPの充実、HWモデルの抽象度アップ



アジェンダ

- SLD研究会活動の紹介
- 設計者ニーズの調査
- シーズとしてのシステム設計技術
- 提案するシステム設計手法
- システム設計言語調査と提案手法への適用
評価
- まとめ



シーズとしてのシステム設計技術

■ 調査対象

■ 『2002年 EDA技術ロードマップ』

EIAJ/EDA技術委員会/EDAビジョン研究会編 (1998年)

■ 大学/研究機関係システム

- POLIS <http://www-cad.eecs.berkeley.edu/Respep/Research/hsc/abstract.html>
"Hardware-Software Co-Design of Embedded Systems, The POLIS Approach"
Kluwer Academic Publishers(1997)
- OCAPI-xl <http://www.imec.be/ocapi>
- IpChinook <http://www.cs.washington.edu/research/chinook/index.html>

■ ベンダ系システム

- CoCentric <http://www.synopsys.com/>
- N2C <http://www.coware.com/>
- VCC <http://www.cadence.com/>

EDA技術の予測

予測では2000年にはほとんどの技術が試行段階になっている
実際には？

| 項目 | 1997年 | 1998年 | 1999年 | 2000年 | 2001年 | 2002年 | 2003年 | 2004年 |
|--|-------|-------|-------|-------|-------|-------|-------|-------|
| [システム設計] | | | | | | | | |
| システムモデルの標準化 | | | | | | | | ■ |
| システム記述言語の標準化(SLDLなど) | | | | | | | | ■ |
| システム記述言語SLDLシミュレーション | | | | | | | ■ | ■ |
| システムレベル性能予測 (アプリケーションソフト、コンパイラ、ハードウェアの性能解析、最適化) | | | | | | | | ■ |
| 高速プロトタイピング | | | | | | | | ■ |
| フォーマルベリフィケーション(仕様-システム) | | | | | | | | ■ |
| システムのHW/SW分割 | | | | | | | | ■ |
| システムレベルライブラリ (IPコア、ミドルウェアなど) | | | | | | | | ■ |
| システムテスト戦略の決定支援 | | | | | | | | ■ |
| アナログ/デジタルのトレードオフ | | | | | | | | ■ |
| [アーキテクチャ設計] | | | | | | | | |
| アーキテクチャモデルの標準化 (分野別モデルを含む) | | | | | | | | ■ |
| アーキテクチャ記述言語の標準化 (Verilog-HDL、VHDLなど) | | | | | | | | ■ |
| アーキテクチャレベルの推定 (面積、遅延、電力、フロアプラン) | | | | | | | | ■ |
| RTL設計制約決定 (面積、タイミング、消費電力、フロアプランのバジェットティング) | | | | | | | | ■ |
| フォーマルベリフィケーション (システム - アーキテクチャ) | | | | | | | | ■ |
| バリデーション/シミュレーション | | | | | | | | ■ |
| アーキテクチャ合成 | | | | | | | | ■ |
| コ・シンセシス | | | | | | | | ■ |
| アーキテクチャテスト方式決定支援 (テスト方式/面積/テスト長) | | | | | | | | ■ |
| アーキテクチャレベルDFT | | | | | | | | ■ |

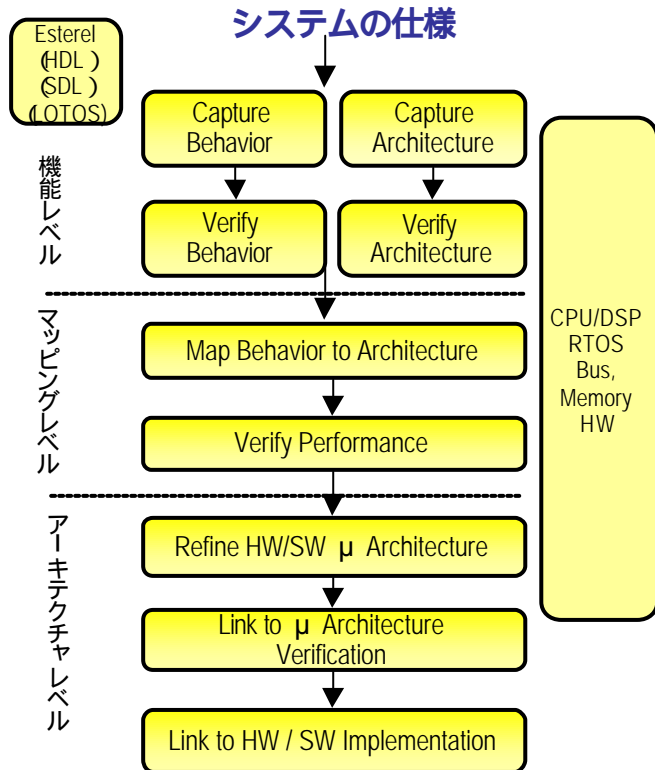
□ : 試行利用 □ : 先行利用 ■ : 実用化

EIAJ/2002年EDA技術ロードマップ(1998)より抜粋

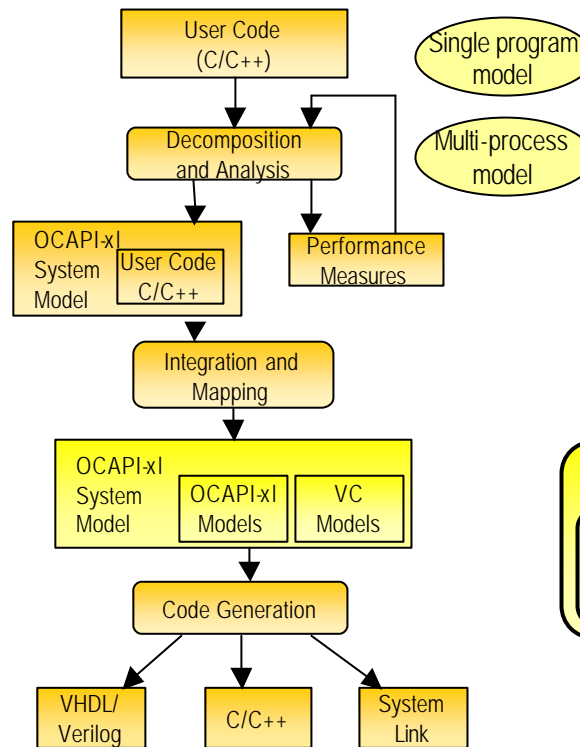
大学・研究機関関係システム

POLIS (UCB, U.S.A.)

- ・マイコン・DSPをベースとした組込みシステム用協調設計環境
- ・機能レベル設計からプロトタイピングまでサポート
- ・ビヘイビアとアーキテクチャの明確な分離による最適アーキテクチャ探索を実現
- ・Fuzzy Instruction Setによる抽象度の高いパフォーマンス評価の実現



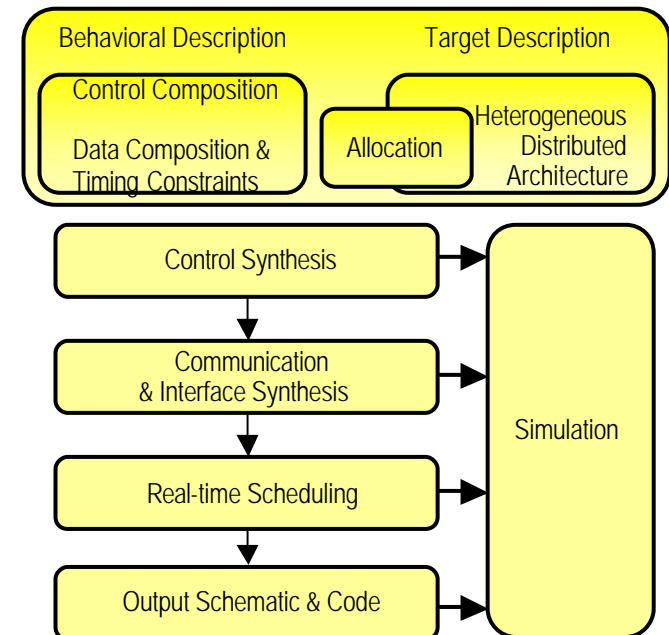
OCAPI-x1 (IMEC, Belgium)



- ・HW、SWを共通に表現できるC++による単一システムモデルのサポート
- ・パフォーマンス解析による最適ブロック分割のサポート
- ・実装へのI/F
VHDL/Verilog及びSoCOS (IMEC)上のCコードを自動生成

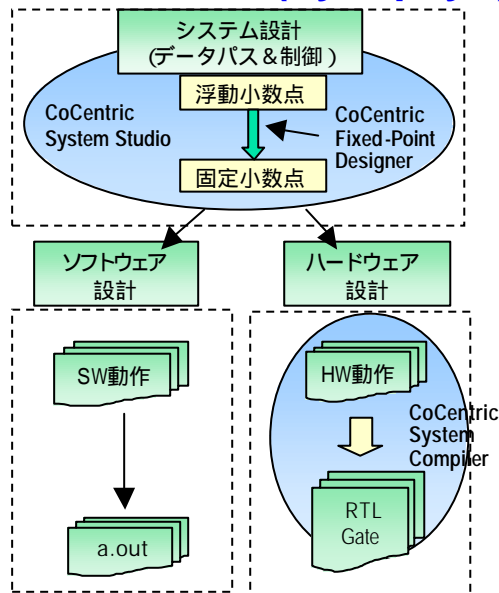
IpChinook (U of Washington, U.S.A.)

- ・IPベース・DesignReuse設計手法をサポート
- ・機能ブロック間の制御やインタフェースを自動生成
- ・豊富な通信プロトコルライブラリーを用意
- ・専用言語をサポート(Java, P, C)



ベンダ系システム

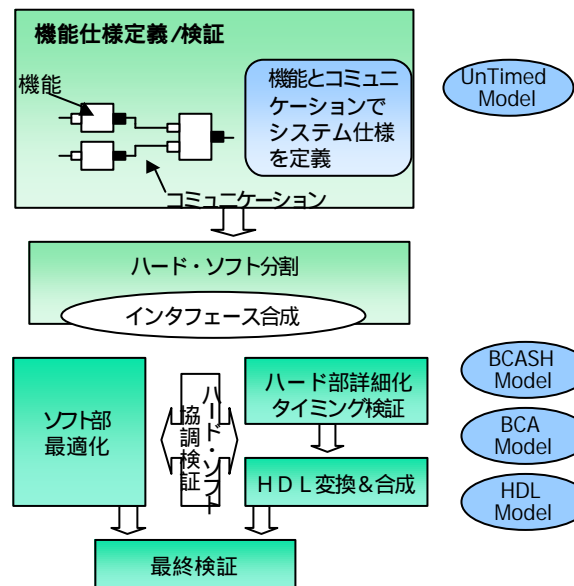
CoCentric (Synopsys)



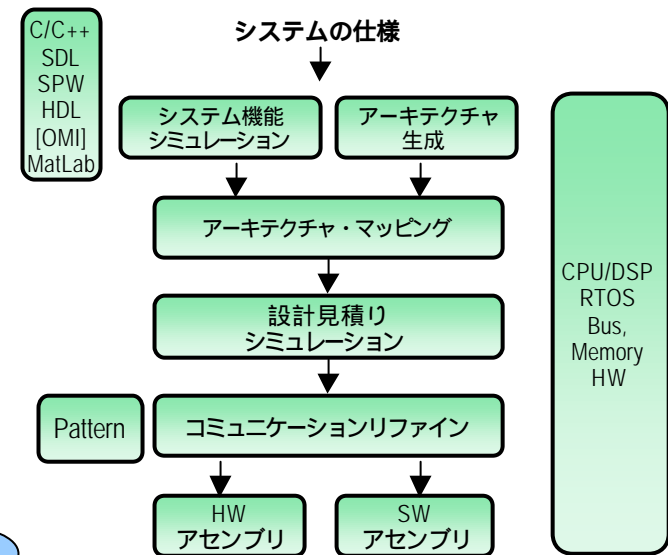
- SOC設計のHW側を主にカバーし、SW側は、協調検証のみをサポート
- 固定小数点化機能
- ピヘイピアレベルからHW合成 (Verilog,VHDL)
- SystemCのサポート

N2C (CoWare)

- 機能レベルから実装までをサポート
- 独自のCベース記述言語(CoWareC)で、Untimed, BCASH, BCAの抽象モデルを定義
- ブロック間コミュニケーションのリファインをサポート
- HW/SWインタフェース合成
- SystemC対応予定



VCC(Cadence)



- 機能とアーキテクチャを分離によるアーキテクチャ探索
- Fuzzy Instruction SetによるSW性能見積り
- IPリユースのサポート
- コミュニケーション合成
- 実装設計へのインタフェース



システム設計技術調査のまとめ

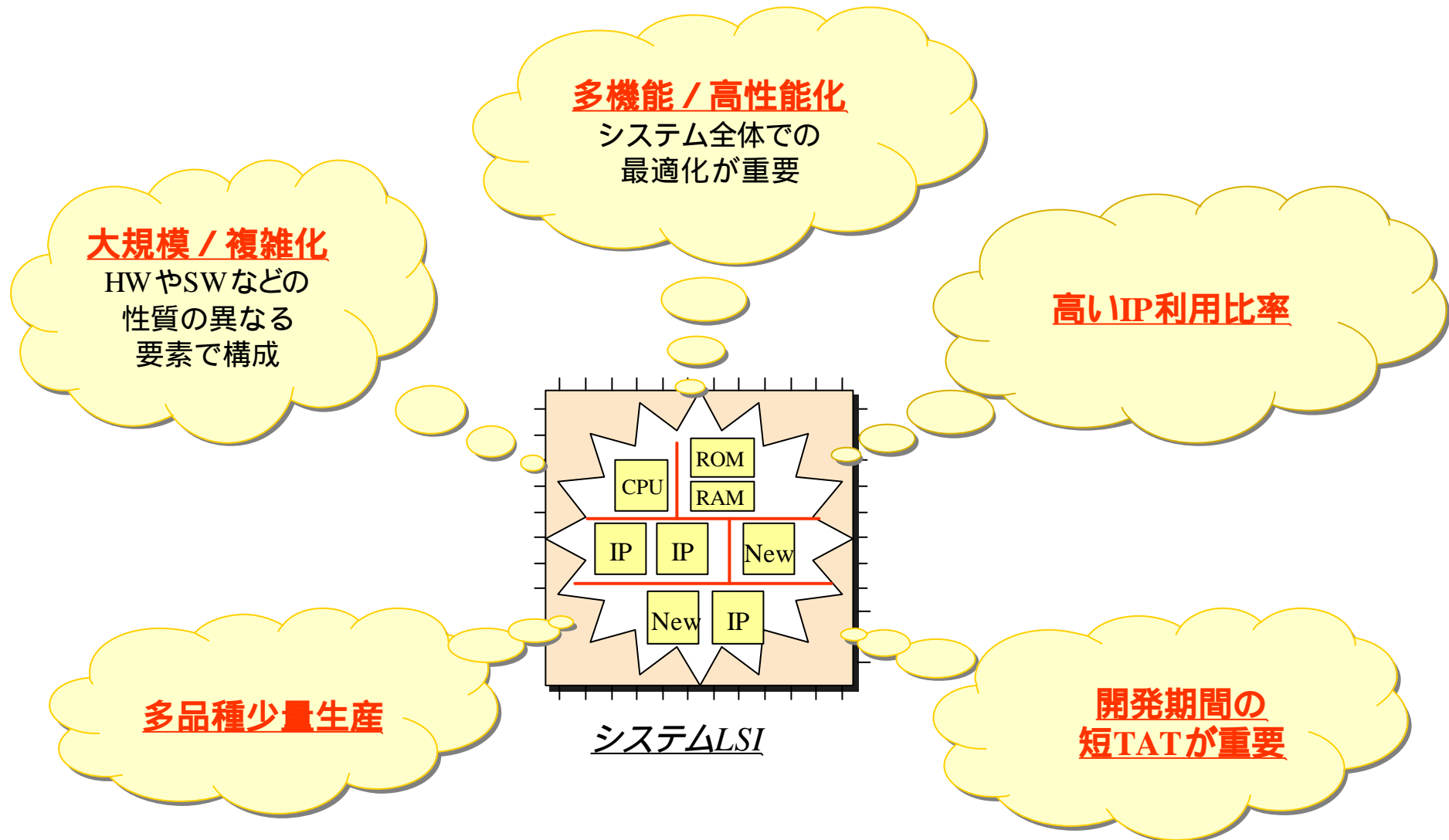
- 実用化されつつある技術
 - 機能検証
 - アーキテクチャ・マッピング
 - 性能見積り
 - インタフェース合成
 - HW/SW 協調検証
- 未解決の課題
 - 機能分割手法
 - 見積りの高速化
 - 面積、電力、コスト見積り
 - システム仕様、設計制約の記述標準化など...



アジェンダ

- SLD研究会活動の紹介
- 設計者ニーズの調査
- シーズとしてのシステム設計技術
- **提案するシステム設計手法**
- システム設計言語調査と提案手法への適用
評価
- まとめ

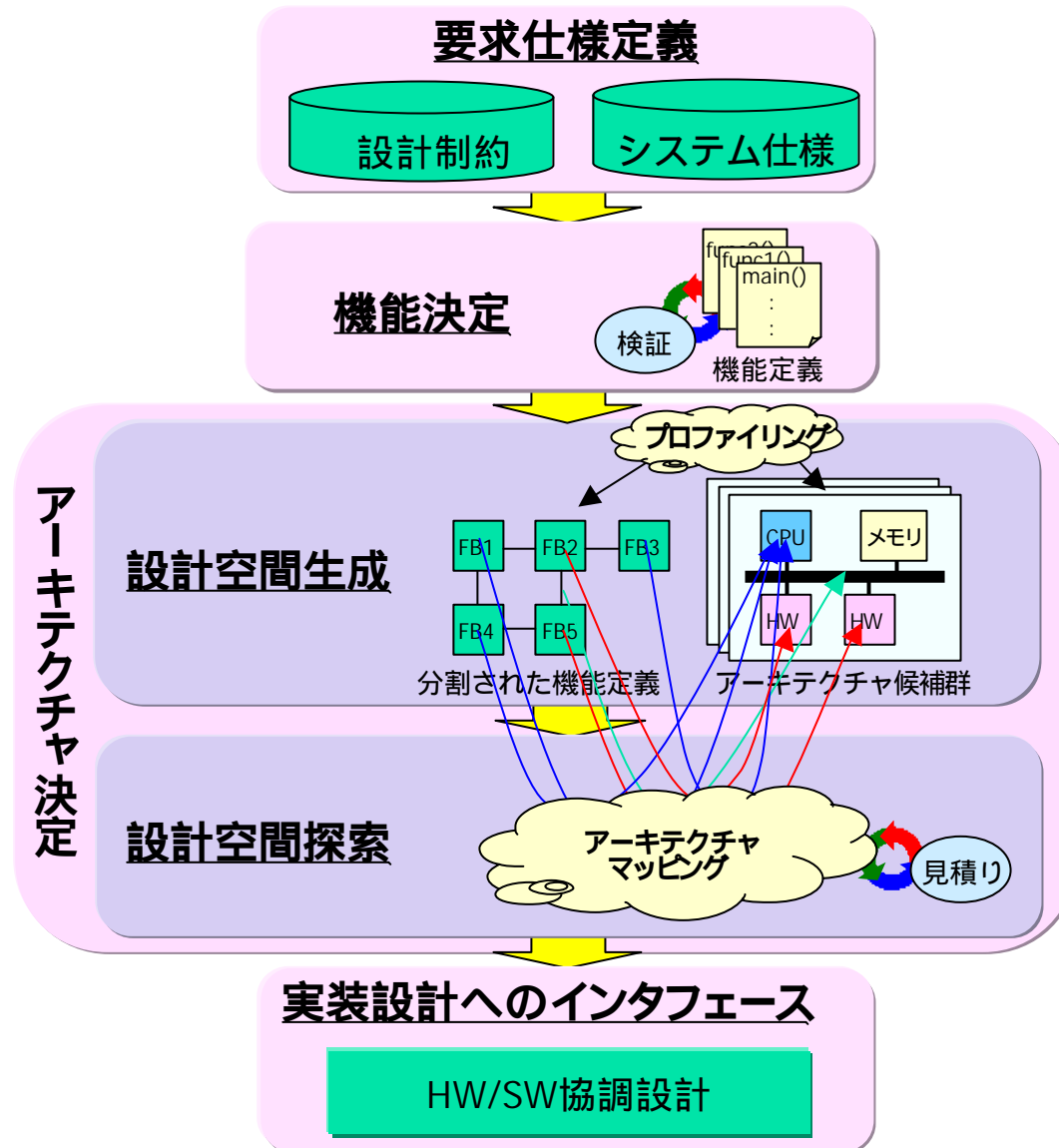
対象とするシステムLSIの特徴



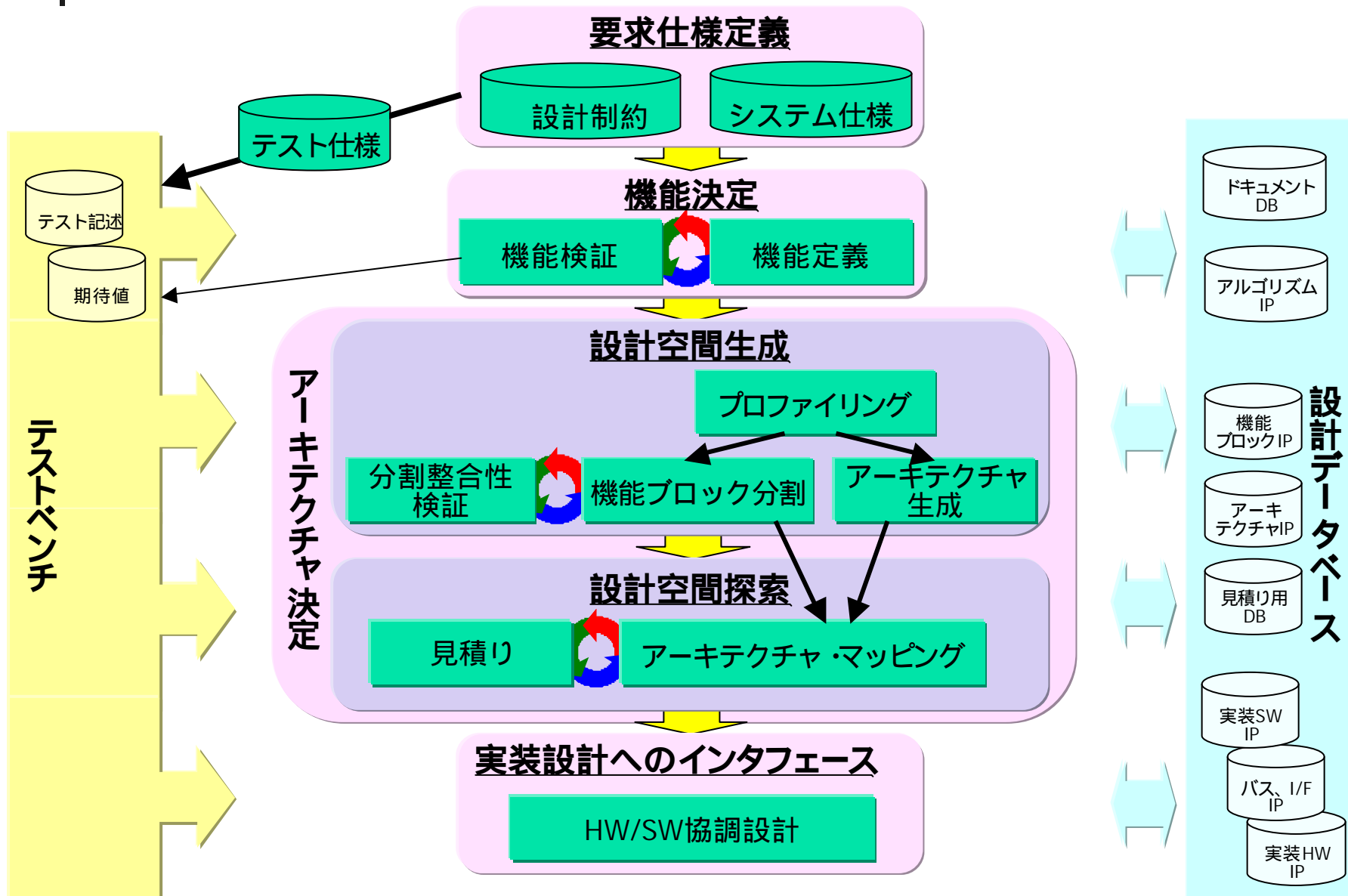
既存のシステム設計フロー



提案するシステム設計フロー

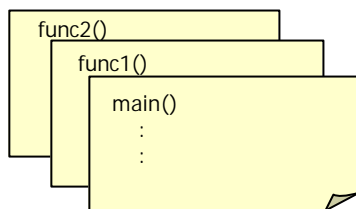


提案するシステム設計フロー



設計空間生成

機能モデルから機能ブロック分割とアーキテクチャ生成を行う



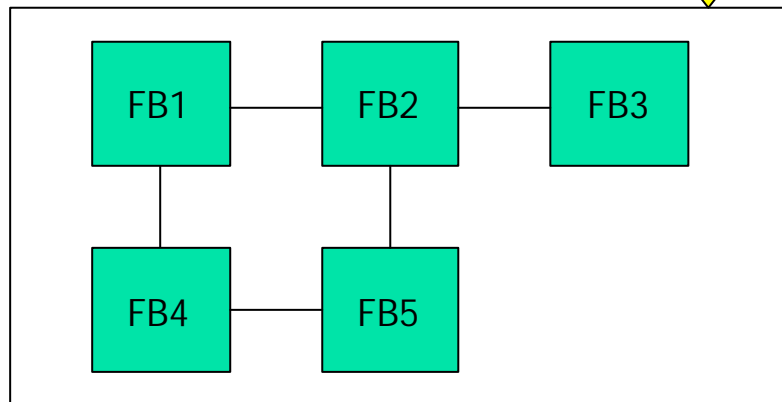
機能モデル

HW・SWを考慮せずにシステムの機能表現した実行可能なモデル
(機能決定フェーズで作成)

設計空間生成

プロファイリング

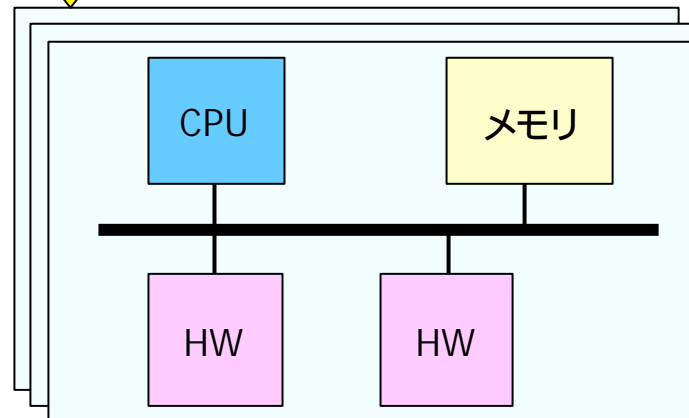
機能ブロック分割



プロセスモデル

機能ブロックとブロック間の通信でシステムを表現したモデル

アーキテクチャ生成



アーキテクチャモデル

システム構成を表現するモデル

プロファイリング

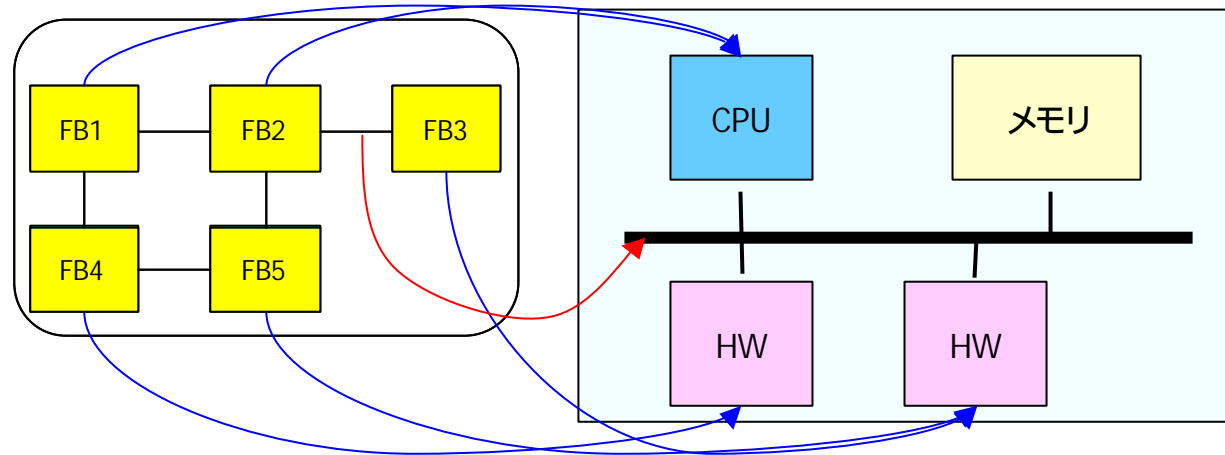
機能モデルを動的または静的に解析し、機能ブロック分割やアーキテクチャ生成を行うための情報を抽出する。

| 情報の分類 | 抽出する情報の例 | 設計空間生成での利用方法 |
|----------|--|--|
| 変数アクセス回数 | <ul style="list-style-type: none">• Read/Writeの回数• アクセスされる箇所• 同一変数のReadとWriteの距離 | <ul style="list-style-type: none">• アルゴリズム変更判断• 分割単位の候補決定• メモリ、レジスタの選択 |
| データ転送量 | <ul style="list-style-type: none">• Read/Writeのデータビット数 * 実行回数• 関数引数のデータ量 | <ul style="list-style-type: none">• 分割単位の候補決定• HW / SW分割の候補決定• 通信プロトコル選択 |
| 演算量 | <ul style="list-style-type: none">• 行単位での演算回数• ブロック単位での演算回数• 関数単位での実行回数 | <ul style="list-style-type: none">• 分割単位の候補決定• HWやSW上での性能見積り |
| 統計解析 | <ul style="list-style-type: none">• 演算 / 制御命令の種類と実行回数 | <ul style="list-style-type: none">• プロセッサの選択• 命令セットの追加 |

設計空間探索

機能ブロック分割した結果をアーキテクチャにマッピングし、見積りを行い設計制約を満足する最適なアーキテクチャを選択する

アーキテクチャ・マッピング



トランザクションモデル

見積り用に処理量のみを表現したモデル

アーキテクチャモデル

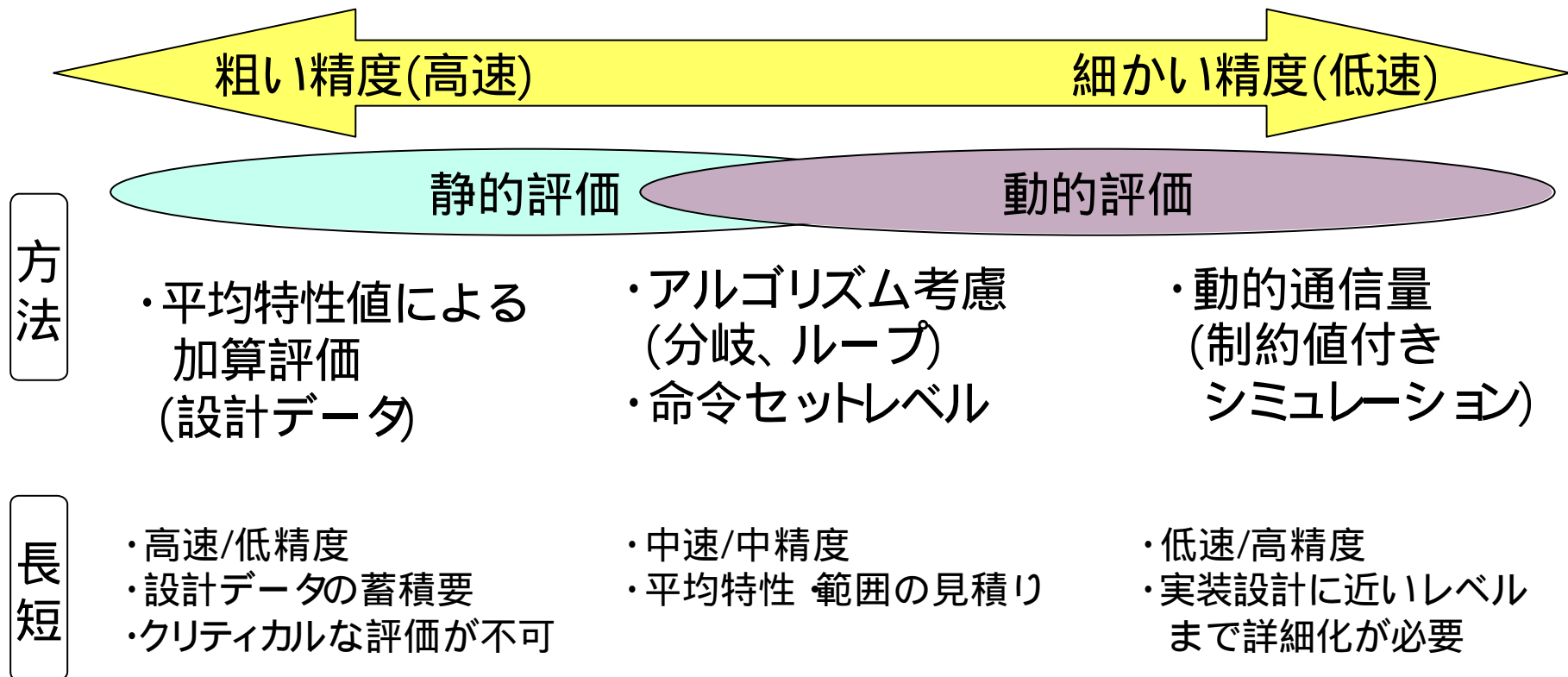
(見積りの例) 処理時間の見積り = (処理量) / (単位時間あたりの処理能力)

トランザクションモデルで表現

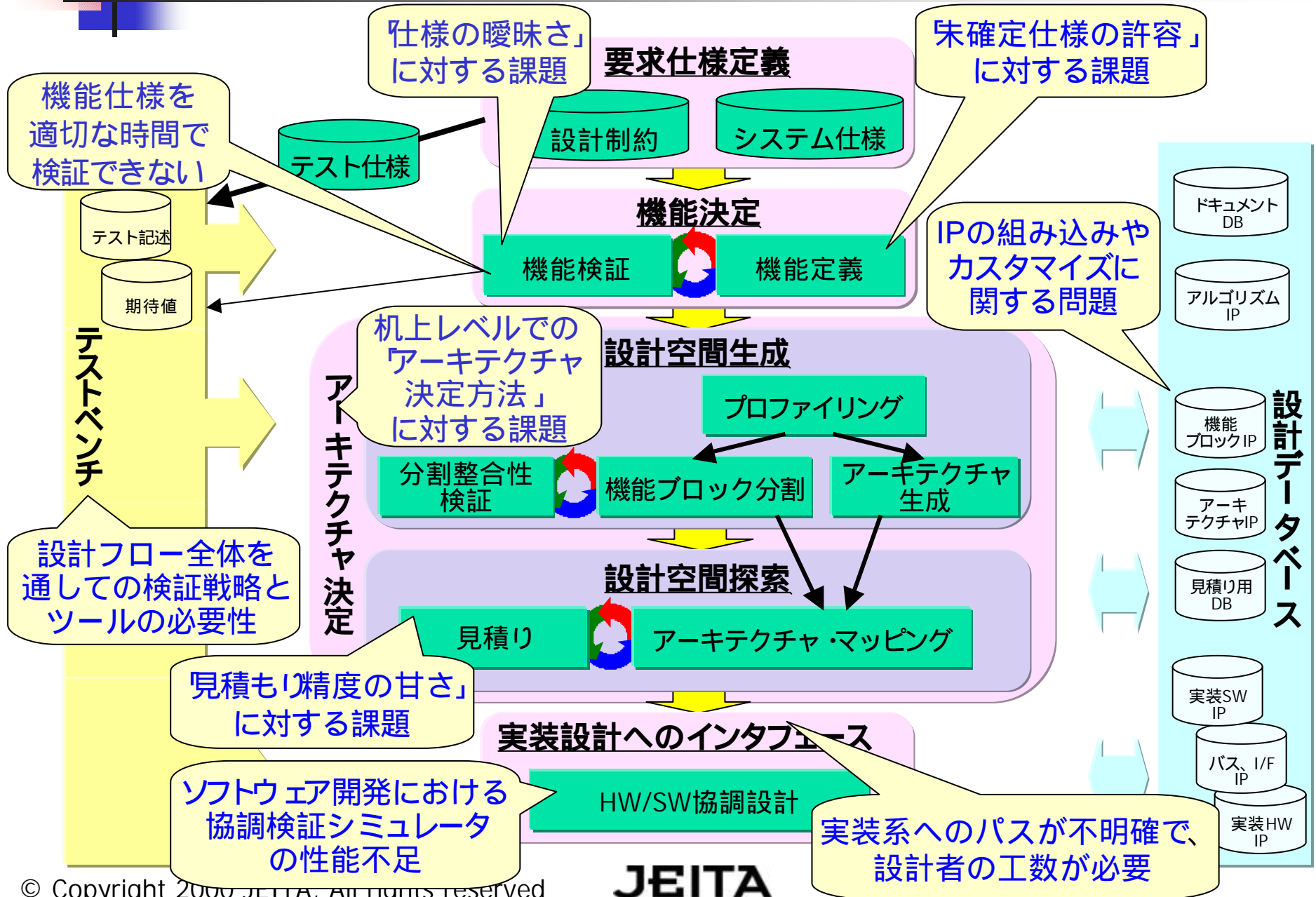
アーキテクチャモデルで表現

トランザクションモデルと精度

見積り精度を変えられるトランザクションモデル



設計者ニーズから抽出した課題の解決



技術 (シーズ) マップ

設計空間生成では、
ツールサポートは、
検証以外はほとんどない

要求仕様定義での
ツールサポートは
ほとんどない

| 設計工程 | | 要素 | ツールA | ツールB | ツールC | ツールD | ツールE | ツールF |
|-------------------|--------|---------------|------|------|------|------|------|------|
| 要求仕様定義 | | システム仕様 | × | × | × | × | × | × |
| | | 設計制約 | × | × | × | × | × | × |
| 機能決定 | | 機能定義 | | | | | | |
| | | 機能検証 | | | | | | |
| アーキテクチャ決定 | 設計空間生成 | プロファイリング | × | × | × | × | | |
| | | 機能ブロック分割 | | | | | | |
| | | アーキテクチャ生成 | | | | | | |
| | 設計空間探索 | 分割整合性検証 | | | | | | |
| | | アーキテクチャ・マッピング | | | | | | |
| | | (性能)見積り | × | | | | | × |
| 実装設計への インタフェース | | インタフェース合成 | × | | | | | |
| | | HW/SW協調検証 | | | × | | | |
| テストベンチ | | テストベンチ作成 | | | | | | |

設計空間探索では、
ツールが整備されてきているが、
見積りの項目 (Power, Cost など) や
速度・精度の点で課題が残されている。

○ : ツールにより自動化が実現されている
設計者を支援する機能がツールにある
設計者を支援する機能がツールにない
× : ツールとして対象としていない

テストベンチ作成を
サポートするツールは
ほとんどない



アジェンダ

- SLD研究会活動の紹介
- 設計者ニーズの調査
- シーズとしてのシステム設計技術
- 提案するシステム設計手法
- **システム設計言語調査と提案手法への適用
評価**
- まとめ



システムレベル設計言語調査

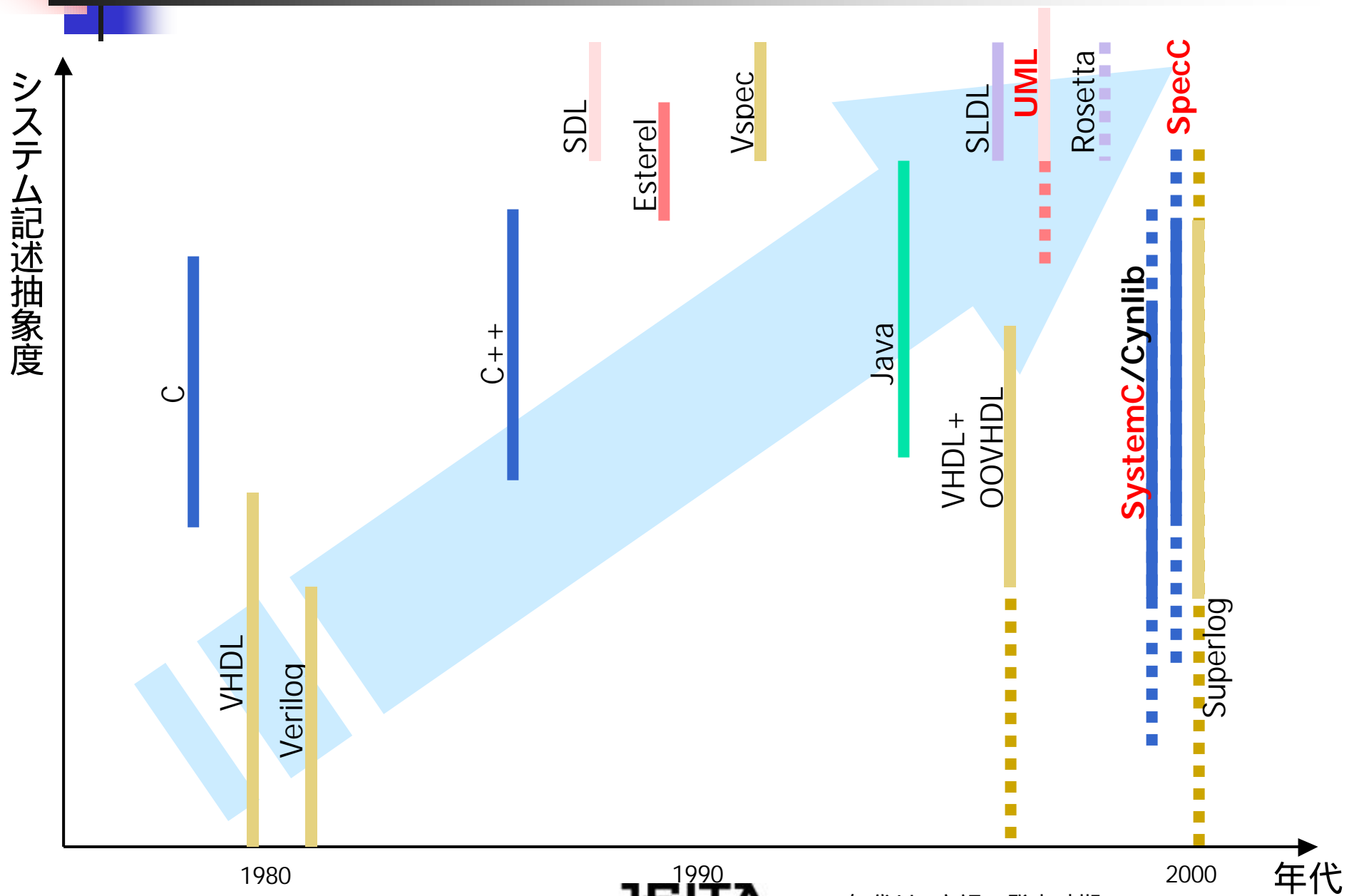
■ 背景

- システム設計フロー実現に対する役割の重要性
- 標準化活動の活発化 (OSCI, STOC, Accellera)

■ 目的

- 標準化動向の把握 (SystemC, SpecC, UMLなど)
- 提案システム設計フローへの適用評価
- 標準化活動へのフィードバック

言語マップ





SystemC概要

■ 背景

- 米Synopsys社、米CoWare社、ベルギーFrontier Design社の技術をベースにシステムレベル設計言語として提案 (1999年9月)

■ 特徴

- **専用クラスの追加によりC++文法を拡張せずにシステムからHW/SWまでを表現**
 - C++の標準的な開発環境がそのまま利用可能
 - モデルの可読性は少し落ちる
- **C++を用いたHW設計から、システムレベル設計へ発展**
 - HDL設計環境からの移行が容易

■ 動向

- Open SystemC Initiative (OSCI)で標準化・普及活動を推進
- EDAベンダ、IPベンダ、半導体メーカーなど70社以上参加 (2000年4月現在)



SpecC言語概要

■ 背景

- カリフォルニア大学アーバイン校のGajski教授等が開発 (1997年)

■ 特徴

- **ANSI-Cの拡張によってHW/SWを同一の言語構文で表現**
 - 高い可読性
 - 専用の開発環境整備が必要 (実装設計へのI/Fなど)
- **システムのモデリングから、HW/SWを考慮したアーキテクチャ決定までカバーする設計方法論が明確**

■ 動向

- SpecCテクノロジ・オープンコンソーシアム (STOC) で標準化・普及活動を推進中
- システムハウス、組込みソフトウェア・ツールベンダの幅広い関心

UML概要

■ 背景

- 複雑、大規模なシステムのモデリング言語として考案
1996年から OMG *1で標準化活動開始
- オブジェクト指向分析と設計の表現方法の統一*2

■ 特徴

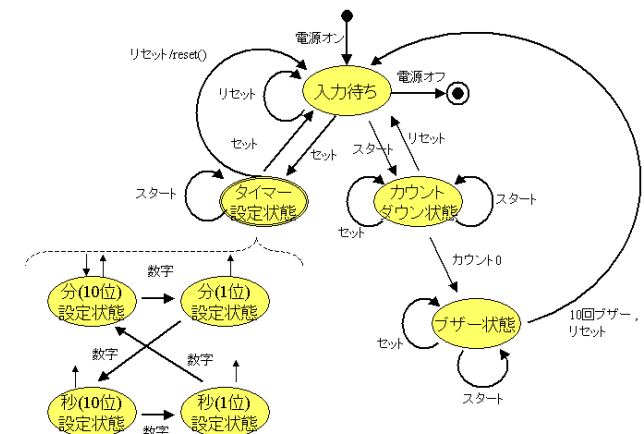
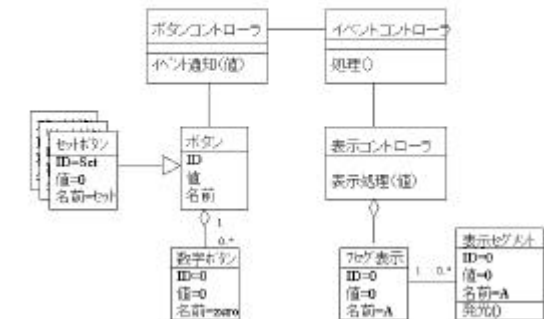
- **高いモデル化能力を持つ図式言語**
 - ビジュアルなマルチビューのシステムモデリングが可能
ユースケース図、クラス図、コンポーネント図、配置図、状態チャート図、シーケンス図、アクティビティ図、コラボレーション図
 - 設計制約の記述可
 - HW設計への適用方法論は未確立

■ 動向

- ビジネスソフトウェア分野で広く適用
- 組み込みソフトウェアへの適用が広まりつつある
ハードウェア設計への適用は模索段階

*1 Object Modeling Group *2 Booch法、OMT法、OOSE法

クラス図例



状態チャート図例

提案フローへの適用検討

対象として機能が
言語の特長
可能
機能不足
× 不可

| 項目 \ 言語 | SystemC (v1.0) | SpecC(v1.0) | UML (v1.3) |
|--|-------------------------------------|--|------------------------------------|
| 要求仕様定義 システム仕様 設計制約 | × × 記述不可 | × × 記述不可 | ビジュアル表現可能 制約記述言語有 |
| 機能決定 機能定義 機能検証 | ○ 可読性おちる 既存インフラあり | 明確な方法論 | SW化が前提 状態チャートのみ可 |
| アーキテクチャ決定 プロファイリング 機能ブロック分割 分割整合性検証 アーキテクチャ生成 アーキテクチャマッピング 見積り | 構造のみ記述可 トランザクションモデル書き難い 方法論不足 | ツール未整備 HW/SW記述の差異なし ○ 構造のみ記述可 トランザクションモデル書き難い 方法論不足 | × 適用を想定せず × × × × × |
| 実装設計へのインターフェース HW/SW協調設計 | RTLまでサポート | RTL以降は既存フロー | × 適用を想定せず |
| テストベンチ作成 | ○ | ○ | テスト仕様まで |
| IP設計 | | | × |



調査言語の特徴のまとめ

- SystemC
 - 専用クラスの追加によりC++文法を拡張せずにシステムからHW/SWまでを表現
 - 設計方法論の明確化が必要
 - 実装面では強いがアーキテクチャ生成から見積もりに関しては検討が必要
- SpecC
 - ANSI-Cの拡張によってHW/SWを同一の言語構文で表現
 - 専用開発環境の整備と充実が普及の鍵
 - システム記述の方法論が充実しているが、アーキテクチャ生成以降が弱い。
- UML
 - ビジュアルモデリングにより仕様 / 制約記述能力が大きい
 - アーキテクチャ設計以降は考慮されていないので、実装につながる設計言語へのリンクを考慮する必要がある。



アジェンダ

- SLD研究会活動の紹介
- 設計者ニーズの調査
- シーズンとしてのシステム設計技術
- 提案するシステム設計手法
- システム設計言語調査と提案手法への適用
評価
- まとめ



まとめ

- システムレベル設計に対するニーズ調査
 - 設計者懇談会
 - 設計者アンケート
- システム設計技術動向 (シーズ) 調査
- システム設計フローの提案と有効性評価
- 提案フローへの標準化言語の適用評価
 - SystemC、SpecC、UML



課題と要望

- 今後の課題
 - 設計フローの実証検証
- 大学・研究機関への要望
 - 高速・高精度見積り技術の開発
 - 機能ブロック自動分割技術の開発
 - プロファイリング技術の開発
- EDAベンダへの要望
 - IP設計データベースのインフラ整備
 - アーキテクチャ決定ツールの充実
 - 仕様からのテストベンチ生成ツールの実現
- 標準化団体への要望
 - システム仕様、設計制約の記述標準化
 - システムモデルの標準化
 - 合成セマンティクスの定義