

1. 設計者ニーズ調査

SLD 研究会では、現在、システム設計においてどのような課題があるか把握するため、設計者ニーズの調査を行った。

最初に、1999 年 2 月 24 日に設計者技術懇談会¹を開催した。設計者技術懇談会は、設計の第一線で活躍している 5 人の設計者に集まっていただき、予め用意しておいた 5 つのテーマに関して、意見を述べてもらい自由に討議するという形で進めた。

続いて、1999 年 9 月に、JEITA (旧 EIAJ) 加盟企業を中心に設計者アンケート²を実施した。設計者アンケートは、システム設計に関してより多くの設計者の現状を知るために、7 分類 53 項目にわたる質問を行った。

そして、これらの結果と研究会メンバーが認識しているニーズを基に分析し、設計者ニーズとして課題を分類、その解決案を検討した。

本章では、それぞれの調査の概要に触れ、分類した課題とその解決案を述べる。

1.1. 設計者技術懇談会

1.1.1. 概要

SLD 研究会において「あるべきシステム設計像」を議論するにあたり、議論の対象が現実の世界とかけ離れないためにも、第一線で活躍中の設計者と意見交換することが必要と考えた。そこで、SLD 研究会のメンバー会社に参加を呼びかけ、結果として 5 名の設計者との懇談会を開催した。

5 名の設計者は、各社において次のようなポジションに位置する。

- 設計支援チームに属し、ASIC の設計を行う。
- ハードウェアの設計部門に属する。
- 所内に試作プロセスを持つ研究所に属する。
- 組込みソフト開発のためのインフラ提供を行う。
- ハードウェア開発を行う。

各設計者に対して、懇談会では次にあげる 5 つのテーマに沿って、意見交換を行った。

¹ 「EDA アニュアルレポート 1998」に、設計者技術懇談会の報告が掲載されている。

² 「EDA アニュアルレポート 1999」に、アンケート実施結果が掲載されている。

- 現在の設計について
- システム仕様の決定について
- 決定された仕様の実現について
- IP に関する考え方
- 期待されるシステム設計とは

以下、それぞれのテーマ毎に、どのような意見が出たかを報告する。

1.1.2. 現在の設計について

現在の設計は、ソフトウェア／ハードウェア分割までの上流設計は有効なツールが無いこともあり、経験と勘に頼って行われている。この時、CPU や OS は決められていることが多い。ソフトウェア／ハードウェアの分割後、それぞれの仕様に基づいて個別に開発を進めており、ハードウェア設計は RTL より行われている。ソフトウェア／ハードウェアそれぞれの開発が終わった後、実機での統合試験を行っている。統合試験において発生した問題は、改変のし易いソフトウェアで対処することが多い。

設計上問題となっているのは、仕様に関して言えば、開発工程の関係で未確定な部分を残したまま設計に入らざるを得ない事や仕様不備である。協調検証に関しては、性能面で満足できないようだ。

1.1.3. システム仕様の決定について

システムの仕様は、客先対応者又は事業部の者と、ソフトウェア／ハードウェアのリーダ等による協議で決定する場合や、システムアーキテクトと呼ばれる専門家が一人で決定する場合がある。机上での合議による仕様決定が多いが、一部シミュレータ等を使い検証を行った後、デザインレビューを経て決める事もある。

決定した仕様の実現性に関しては、システム仕様決定者の経験と勘に頼るところが多い。初期の見積り等も経験と勘がほとんどである。ただし、最終的な見積りには、シミュレータ等のツールを使って評価することもある。

システム全体の機能確認、性能評価はできるだけ早い段階で行いたく、システムレベル検証は望まれている。

システム仕様の決定には、全開発工程の 2～4 割程度を割いており、工数のかかる検討項目は、目的により様々だが、コスト、処理速度、消費電力等が代表的である。

1.1.4. 決定された仕様の実現について

決定された仕様は、仕様に明記されていない部分を勝手な解釈により仕様のない機能を実現してしまったり、設計を下流に進めることで RTL では見えないようなタイミング問題が発生したり、意図通りに実現できないことがある。

このような場合、下流で対応しきれない時に限って仕様変更を伴う。また、開発中にサービスの追加等による仕様変更も生じる。仕様の変更は、ドキュメント化により情報共有をしっかりとさせたうえで、設計変更を実施する。

1.1.5. IP に関する考え方

ハードウェア IP に関しては、必要性は感じているが、使い方が難しく自由に使えるレベルには達してない。問題を発生させないためには、ハード IP³を使うことになるが改修の自由度が少ないため適用できるものは少ない。ソフト IP⁴は自由度があるが、タイミング問題等が発生してしまう。いずれにしても、今後使える IP があれば使っていきたい。

ソフトウェア IP に関しては、OS コール、割込み、別タスクを作る方法により IP 化は可能だが、リアルタイム性の保証は難しい。通常、コード量とパフォーマンスがわかるバイナリ IP を使用する。ソース IP を使用する場合は、コンサルティングと教育無しでは使えない。

1.1.6. 期待されるシステム設計とは

システム設計は、仕様定義から LSI チップ化まで整合性のとれた統合環境で行いたい。記述した仕様に関しては、外界まで含めた全体システムを現実的な時間で検証可能なシミュレータが必要である。ソフトウェア/ハードウェアの分割に関しては、ツールによる自動分割やインタフェースの合成が望まれる。

³ ハード IP : VSIA で定義されているハード VC。VSIA では、ハードウェア VC を 3 つの形式 (ソフト、ファーム、ハード) で定義している。ソフトはコード、ファームはコードと構造、ハードは物理的なものを示す。

⁴ ソフト IP : VSIA で定義されているソフト VC。

1.2. 設計者アンケート

1.2.1. 概要

設計者アンケートは、研究会のメンバー会社を中心に、JEITA のメンバー会社に協力をお願いし、14 社 163 名のシステム LSI 開発者や支援に携わるエンジニアから回答を得ることができた。アンケート対象者の担当業務を図 2.2-1 に示す。

アンケート項目は、設計者技術懇談会でのテーマをもとにさらにブレイクダウンした 7 分類 53 項目について調査を行った。

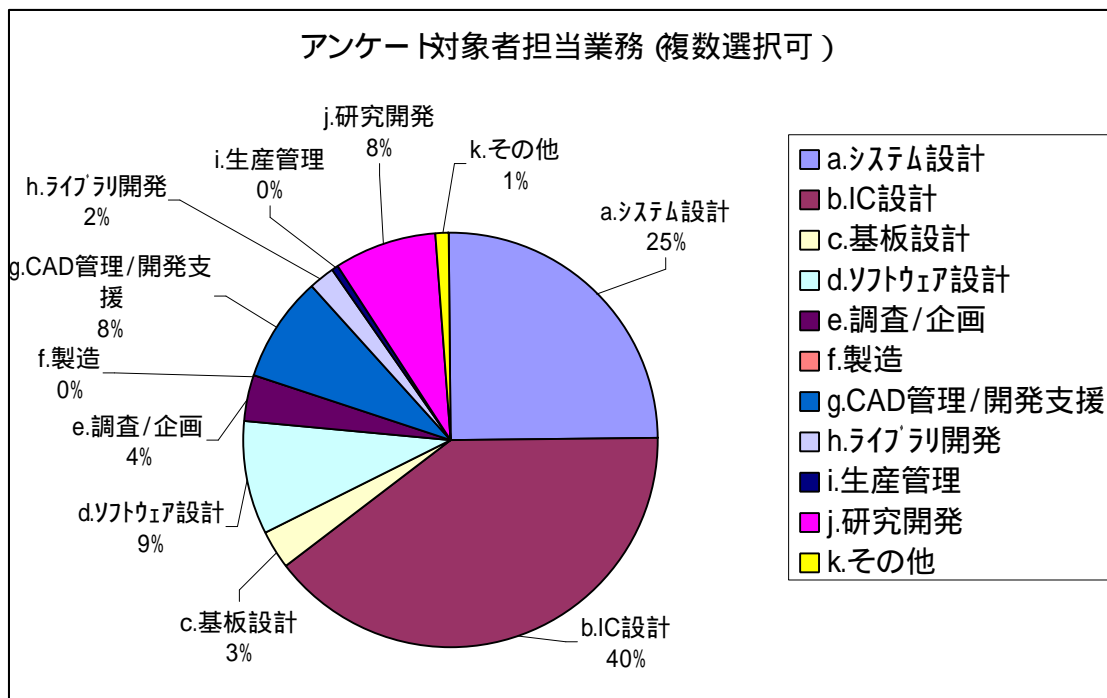


図 2.2-1 アンケート対象者担当業務

1.2.2. アンケート結果

ここでは、アンケート結果の代表的な例として、「現在の設計フローのボトルネック」、「ボトルネックの原因」、「システム仕様変更が発生する原因」及び「IP 使用時の問題点」についての結果を示す。尚、アンケートの詳細は、EDA アニュアルレポート1999 に記載されている。

図2.2-2に現在の設計フローにおけるボトルネックについてのアンケート結果を示す。

システム仕様設計（仕様変更も含む）、システム設計、システム検証、アルゴリズム設計、およびアルゴリズム検証が約 40%の割合を占めており、設計フローの上流段階においてボトルネックがあることがわかる。

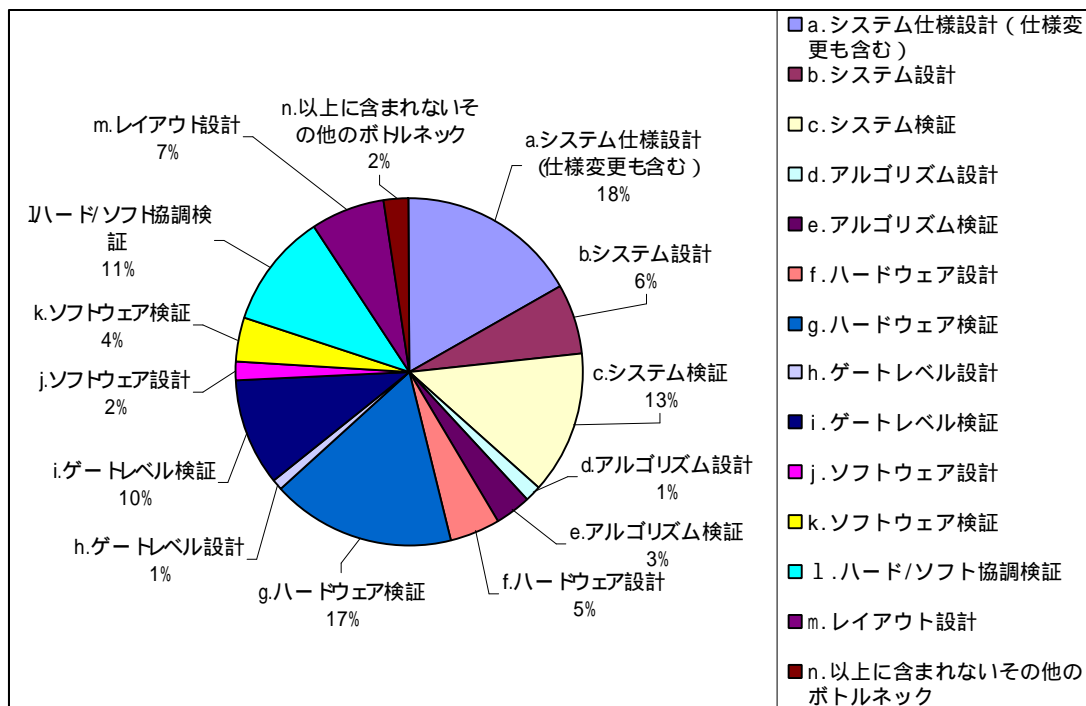


図 2.2-2 現在の設計フローのボトルネック

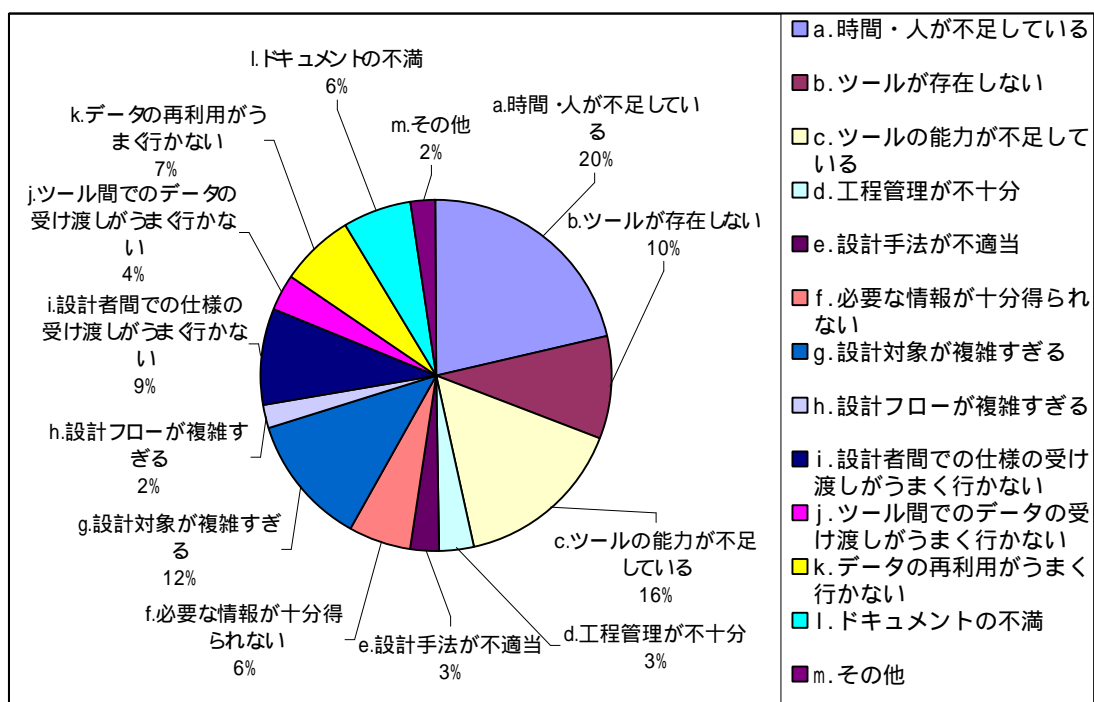


図 2.2-3 ボトルネックの原因

次に、図 2.2-3 にボトルネックの原因についてのアンケート結果を示す。ボトルネックの原因は、人や時間の問題を別とすれば、ツール、設計手法、及び設計再利用が

問題となっていることが分かる。

図 2.2-4 に、システム仕様変更が発生する主な原因についてのアンケート結果を示す。

これをみると、システム仕様の変更が発生する原因としては、仕様の誤りや曖昧さ、誤解釈など、仕様そのものに起因する原因が 43%、機能不足、ハードウェアやソフトウェアの性能不足、コスト・オーバーなど、見積りの甘さに起因するものが 32%と高い割合を示しており、早い段階での仕様の検証や見積り精度の向上などが必要であることが分かる。

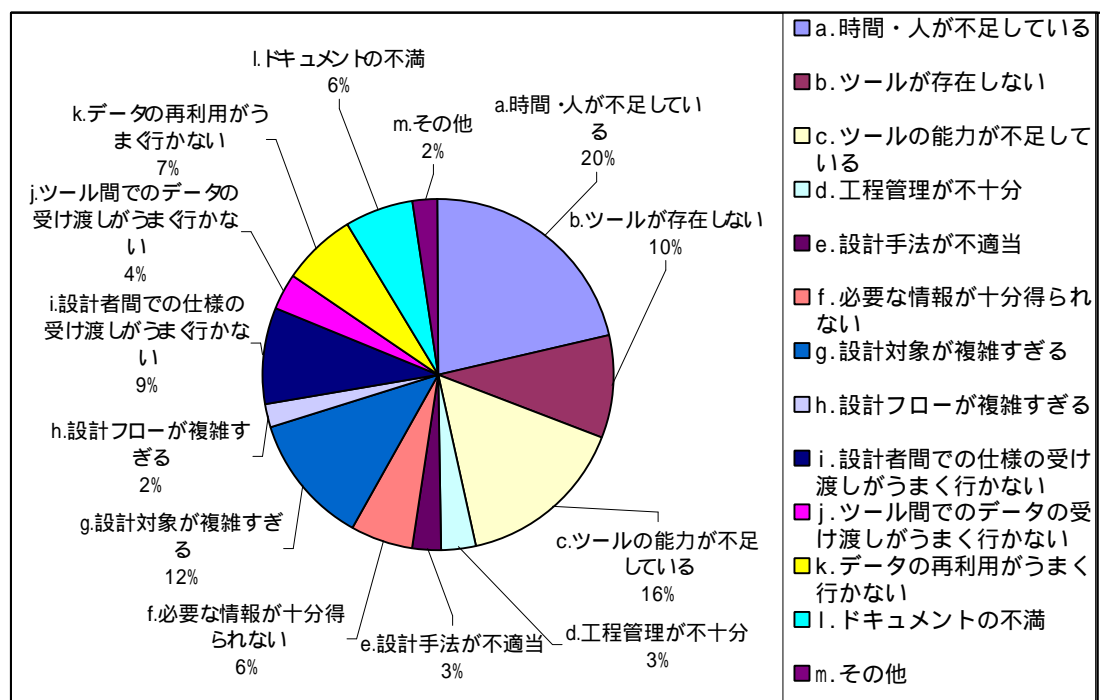


図 2.2-4 システム仕様変更が発生する主な原因

IP 使用時の問題点としては、図 2.2-5 に示されるように半分以上の回答者が、IP を使うためには機能変更等のなんらかのカスタマイズを行うことが必要で、現状では、そのために多くの工数がかかっているということを指摘していることが分かる。

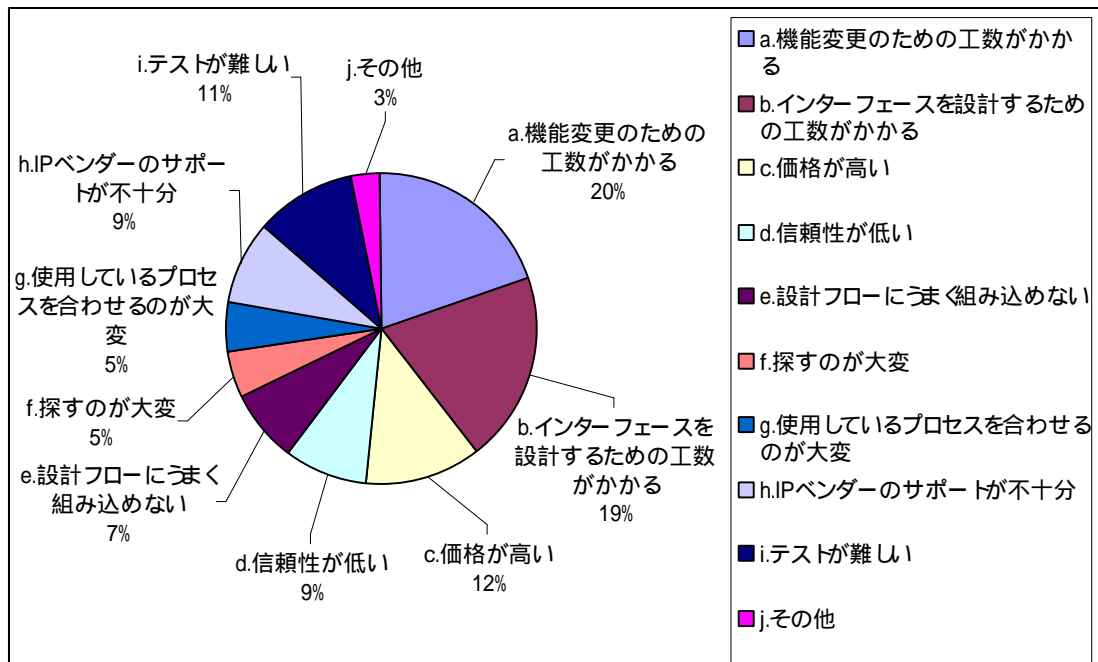


図 2.2-5 IP 使用時の問題点

また、アンケートの自由意見として以下のような回答が得られた。

設計フローに関しては；

- 業界全体の生産性向上のためには、設計データフローのある程度の標準化は避けて通れない
- 仕様書という紙レベルのものではなく、システムレベルで検証された設計データを設計下流に直接展開できるようにしたい
- システムレベル設計の自動化は、下流（LSI チップ化）までを一貫してつながるフローが実現しないと実運用にいたるのは難しい（例えば、ハードウェア/ソフトウェアの分割 ハードウェアへの要求スペックを合成制約条件として自動的に論理合成 タイミングドリブン・レイアウト）

設計ツールに関しては；

- 論理シミュレーション（ハードウェア、ソフトウェアを含む）と基板上でのシミュレーション（伝送線路、ノイズ、EMC 他）を同一環境で行えるツールが必要
- RT レベルより抽象度の高いシステムレベル設計言語の標準化が必要
- ハードウェア/ソフトウェアの協調検証可能な高速検証ツールが必須

IP 利用に関しては；

- IP をうまく使うツールや既存の設計データを IP 化するためのツールがない
- IP の流通の仕組みが確立されることが必要
- 個々の IP が容易に接続可能なプラットフォーム（バス等）がなければ、他社もしくは汎用の IP を使うことは難しい

等が代表的な意見である。

1.2.3. 設計者アンケートのまとめ

アンケート結果より、システム仕様設計に関しては、仕様の曖昧さ・誤解釈を無くすこと、及び仕様設計段階で性能見積りができること、設計ツールに関しては、仕様設計のような上流設計からソフトウェア／ハードウェア記述までが上手くつながること、IP に関しては、カスタマイズなしに容易に接続できることなどが重要であることが分かる。

これらを満たすような設計フロー、ツール及び IP（利用法）が必要となっている。

1.3. 課題

1.3.1. 問題点の分析

前節で述べたニーズ調査結果より、設計者の抱えている問題点は以下の項目に要約することができる。

【1】システム仕様決定に関する問題

システム仕様を決定する際に、現在の設計では適切な表現による有用な決定手法が無い。その為、機能の決定やアーキテクチャなどを決定する場合、まだまだ設計者による勘と経験に頼っている。

【2】IP 利用に関する問題

システム設計を行う場合、多くの設計者は、過去の設計資産等の流用設計を検討している。しかしながら実際には、カスタマイズや性能評価に多くの工数をとられるため、新規設計に切替える場合も少なくない。

【3】設計環境に関する問題

システム設計に関連する設計・検証ツールは、まだまだ多くはない。現状はその不足分を、

人的リソースで補っている。たとえば、実装設計へのインタフェースを行う場合は、設計者が設計データのやりなおしを行う場合も少なくない。

【4】見積りと評価に関する問題

現状、システムの性能や精度は、実装設計に進んだ段階で判断されており、早い段階での見積る仕組みやその精度はまだ低い。

1.3.2. 課題の分類と抽出

問題点を分析してみると次の課題項目毎に分類することができる。

- 機能決定に関する課題
- アーキテクチャ決定に関する課題
- 実装設計へのインタフェースに関する課題
- 設計全般 (IP 利用、協調検証環境等)に関する課題

【1】機能決定に関する課題

機能決定に関する問題点としては、第一に、仕様決定の際に十分な検討がなされなかったことによる仕様内容の曖昧さが挙げられる。また、十分な検討はされていても、自然言語などによる仕様表現の曖昧さからくる、仕様の誤解釈が挙げられる。第二に、システム仕様の中に、業界標準フォーマットなどの採用を考えた場合、仕様が未確定な状態で、開発を開始しなければならないケースも発生する。その場合、それらの仕様が確定した時点で、見直しが必要となる。第三に、システム全体の機能仕様をリーズナブルな時間で検証することが困難な点が挙げられる。

【2】アーキテクチャ決定に関する課題

アーキテクチャ決定に関する問題点としては、第一に、現状のアーキテクチャ決定方法が机上レベルで行われているために、その妥当性を確認し、決定する手段がないことがあげられる。第二に、決定したアーキテクチャに対する見積りの甘さによる、後工程でのやり直しがあげられる。

【3】実装設計へのインタフェースに関する課題

実装設計へのインタフェースに関する問題点としては、第一に実装系へのインタフ

エースが不明確な点が挙げられる。これは、たとえば RTL 設計の実装系へのインタフェースを例にとった場合、RTL から実装までを実証してくれる設計環境が少なく、そのインタフェースを設計者の工数で補っているのが現状である。第二に、機能仕様検証やアーキテクチャ検討結果と、実装設計後の検証結果を合わせることが難しい点が挙げられる。これは、システムレベル設計フロー全体を通じて、全体の検証ストラテジの必要性と、サポートする設計環境がまだ確立されていないことによる。

【4】設計全般での課題

その他システム設計全体を通しての問題としては、第一に IP の組込みやカスタマイズに関する問題が挙げられる。この問題は、IP 利用への要求が高い反面、IP が有効かどうかを簡単に評価できない点と、期待する性能を満足するようにカスタマイズする際多くの工数がかかることによる。第二に、システムのソフトウェア検証を行う場合、現在の協調検証環境では満足のいくパフォーマンスが得られていない点が挙げられる。ソフトウェア検証で求められる検証速度は数 10^5 命令 / 秒 であるのに対し、現状の協調検証シミュレータでは、 $10^3 \sim 10^4$ 命令 / 秒の検証速度しか得られず、およそ 1 桁～2 桁の開きがあることによるものである。

1.3.3. 課題のまとめ

2.3.2 節で抽出した課題をまとめると、表 2.3-1 のようになる。

表 2.3-1 課題のまとめ

課題	内容
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が発生する。 機能の検証をリーズナブルな時間内でできない。
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が起こる。
実装設計への インタフェース	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。
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。

1.4. 解決案

設計者ニーズから抽出した課題に対して、次のような解決方法が考えられる。

1.4.1. 課題解決方法の検討

【1】機能決定に関する課題解決

機能決定の際に課題としてあげられている仕様内容の曖昧さに関しては、機能仕様定義の方法論を検討していく必要がある。また、曖昧な表現による仕様の誤解釈に関しては、仕様表現を実行可能なものにすることにより仕様の早期確認を行い、曖昧な表現を除去することが可能と考えられる。さらに、仕様が確定していない状態で開発を開始した場合の未確定仕様の許容方法に関しては、実行可能で、かつ機能仕様を表現できる設計言語の導入が有効と考えられる。

【2】アーキテクチャ決定に関する課題解決

アーキテクチャ決定の際の課題としてあげられているアーキテクチャの妥当性の確認、及び決定手段がない点に関しては、最適なアーキテクチャを高速に探索する仕組み（技術、設計フロー）を検討していく必要がある。また、アーキテクチャに対する見積りの甘さに関しては、必要な見積りモデルの導入を考えていく必要がある。

【3】実装設計へのインタフェースに関する課題解決

実装設計へのインタフェースに関する課題解決は、実装設計の前段階においてハードウェア/ソフトウェアのインタフェース合成を導入する必要があると考える。この技術の導入により、実装設計の前後に生ずるハードウェアとソフトウェア間のズレ、バランスの調整が可能となり、実装後のリスピンを減らすことにつながる。さらに、システム設計フローを一貫したテストベンチの導入によって、機能仕様検証、アーキテクチャの検討結果、実装設計後の検証結果を統一した環境で検証することが可能となる。

【4】設計全般での課題解決

IP の組み込み、カスタマイズに関する課題解決は、IP のインタフェースの標準化や IP の合成技術の導入、システムレベル設計のフロー全体にわたる様々なフェーズでの IP の充実が必要となる。また、システムの協調検証のパフォーマンス不足に関する課題解決は、ネックとなっているハードウェアモデルの抽象度を上げること、アクセラレ

ータやエミュレータとのリンクをとることによって、パフォーマンスを上げることができると考える。

1.4.2. 解決方法のまとめ

2.4.1 節で検討した課題解決をまとめると、表 2.4-1 のようになる。

表 2.4-1 解決方法のまとめ

課題	解決方法
【機能決定】 仕様が曖昧。 仕様未確定で開発開始。 リーズナブルな時間内で機能検証ができない。	機能仕様を定義する為の方法論の導入。 実行可能なシステムレベル設計言語の導入。
【アーキテクチャ決定】 アーキテクチャの妥当性の確認手段なし。 見積りが甘い。	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。
【実装設計インタフェース】 有効なツールがない。 設計結果が仕様と合わない。	ハードウェア / ソフトウェアインタフェース合成技術の導入。 テストベンチによる合致性検証技術の導入。
【設計全体(設計データベース&協調検証)】 IP 組込みやカスタマイズに時間がかかる。 協調検証シミュレータの性能不足。	IP のインタフェースの標準化。 IP 合成技術の導入。 機能設計から実装設計までの IP の充実。 ハードウェアモデルの抽象度アップ。

これらのニーズから抽出した課題に対する解決案は、本研究会を含め、企業・大学の研究機関、EDA ベンダー等、全体の取組み無くして、実現は困難と考える。