

機能検証の解決策の深耕 -SOC機能検証技術の進展と今後の取り組み-

2011年 3月 4日

JEITA半導体技術ロードマップ専門委員会(STRJ)
設計ワーキンググループ (WG1)

目次

◆はじめに

- ワーキングメンバ、スコープ、ミッションなど

◆国際活動

- ITRS2010アップデートなど

◆国内活動

- SOC機能検証技術の進展と今後の取り組み

◆まとめ

用語集

- RTL: Register Transfer Levelの略。回路をフリップフロップ+組み合わせ論理回路で表現したレベルのこと。現在の論理回路設計はおもにこのレベルの記述を使用する。
- SLD: System Level Designの略
- LCP: Logic/Circuit/Physical Designの略
- DFM: Design For Manufacturabilityの略で、歩留まり等の製造性考慮設計のこと。
- SOC: System On Chipの略
- CP: Consumer Portableの略
- EDA: Electronic Design Automationの略
- OVM: Open Verification Methodologyの略
- VMM: Verification Methodology Manual の略
- UVM: Universal Verification Methodologyの略
- UML: Unified Modeling Languageの略、オブジェクト指向のソフトウェア開発における、プログラム設計図の統一表記法。
- IP: Intellectual Propertyの略で、半導体の設計データやシミュレーションモデルなどの設計資産のこと。シミュレーションモデルを検証IPともいう。
- Virtual Prototype: SW 設計者にHW が出来る前に、仮想HWモデルを使用して、アーキテクチャを検証・デバッグ、性能検証を可能にする。
- アサーション: 論理的に成立すべき関係や条件を記述したもの。それをチェックとして論理検証で活用。
- フォーマル(形式的)検証: 設計と仕様を数学的モデルで表現し数学的推論により正しさを検証する静的検証手法。等価性検証とプロパティ検証がある。
- ランダム検証: ランダムにテストパターンを自動生成し、人手による順序性や規則性を排除し、検証の網羅性を高める検証手法。コーナーケースのバグ検出に効果がある。
- 等価検証: ある回路設計についての2つの表現が同じ振る舞いを表していることを形式的に証明するために用いられる手法。
- 高位合成・動作合成: C/C++などのアルゴリズム記述から、RTLを自動的に合成すること。

目次

◆はじめに

- ワーキングメンバ、スコープ、ミッションなど

◆国際活動

- ITRS2010アップデートなど

◆国内活動

- SOC機能検証技術の進展と今後の取り組み

◆まとめ

設計WG(WG1)メンバー

松崎 正己(リーダ)
富士通セミコンダクター
中山 勝敏(サブリーダ)
ルネサスエレクトロニクス
樋渡 有(国際担当)
STARC
隅谷 三喜夫(幹事、国際担当)
パナソニック
豊田 忠雄
シャープ
斎藤 利忠
東芝
浅田 善己
富士通セミコンダクター
柿本 勝
ソニー
朝重 浩喜
パナソニック

浅井 健史
ローム
田代 雅久
ロームグループ/OKIセミコンダクタ
小野 信任
ジーダット
今井 正治
大阪大学

計13名

設計WGのスコープ

SOC設計全般の広範囲な技術分野を担当

System Level Design

- 仕様から最適なHW/SWに分割し、HWに関してはRTL記述を生成する

Logic / Circuit / Physical Design

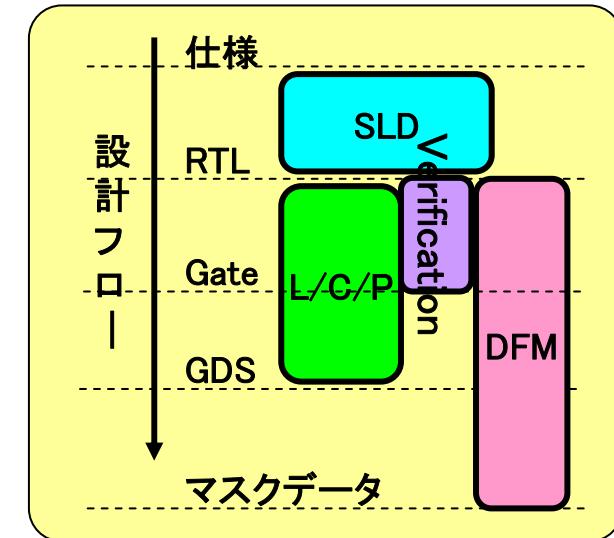
- RTL記述から製造可能な設計品質のレイアウトデータ(GDS II)を生成する

Design Verification

- 機能と性能を仕様に基づき検証する

Design For Manufacturability

- プロセスの物理現象モデルに基づき、製造可能性/歩留まりを検証/最適化する。



設計WGのミッション

◆国際活動 : ITRSのSystem Drivers章とDesign章を担当

- System Drivers章
 - ITRSの全ての技術分野をドライブする製品分野毎の仕様や要求を定義
- Design章
 - 設計技術に対する将来課題と課題解決策の提示

◆国内活動

- SOC構造・規模を時間軸で定量化し、ロードマップ検討の基礎として提示
- 設計技術課題(「**設計生産性**」や「**消費電力**」の観点)を時間軸で定量評価し、解決策を提案(ロードマップ作成)



◆期待される効果

- ITRSロードマップのSOC設計に与える影響を定量化し、発信
- ITRSロードマップ見直しのきっかけをつくる
- 設計技術革新(EDA自動化技術)の加速を支援(EDAベンダーへ)

設計WGの活動内容(2006~2009年度)

	国際活動 (ITRSへの主な貢献)	国内活動	
		設計WG	旧設計TF
2006年度	System Drivers章 Consumer Stationary SOCを提案 Design章 SLDとVerificationの課題の項目値の確認と修正案提示	設計遅れ要因変化の分析と提言 ・設計遅れ要因変化(3年間)の分析と課題解決策提言 DFMのSOC設計への影響考察 ・SOC設計へのばらつきの影響の考察	低電力SOCのロードマップ ・配線分布/抵抗の予測とSOC性能への影響を評価
2007年度	System Drivers章 Consumer Portable &Stationary SOCの数値の見直し Design章 LCPとDFMの課題の確認と修正案	SOC設計技術ロードマップの詳細化/定量化 ・論理検証と物理設計の2分野で「設計生産性向上」の観点でロードマップを詳細化/定量化	配線性能とSOC性能の関係 ・ITRSの配線性能から見たムーアの法則の限界
2008年度	System Drivers章 Consumer Portable &Stationary SOCの数値の見直し Design章 ソフトウェア関係のSLDテーブルへの追加	SOCの低消費電力設計技術の課題と解決策 ・消費電力トレンドを示すロードマップの再構築 モデル(モチーフ)の見直し 消費電力計算式の見直し(パラメータ追加を含む) ・設計生産性に対する低消費電力設計技術の解決策ロードマップの作成	
2009年度	System Drivers章 Consumer Portable SOCモデルの変更 Design章 低消費電力設計の設計工程の貢献度合を新規掲載	SOC大規模化に向けての検証阻害要因分析 ・「検証課題の深耕」をテーマに活動 各社が検証課題を抽出し、検証対象毎に分類・分析を実施。 検証対象(ブロック検証、ブロック間検証、1チップ検証)それぞれの要件、定義を行い、解決すべき課題を明確化。	

目次

◆はじめに

- ワーキングメンバ、スコープ、ミッションなど

◆国際活動

- ITRS2010アップデートなど

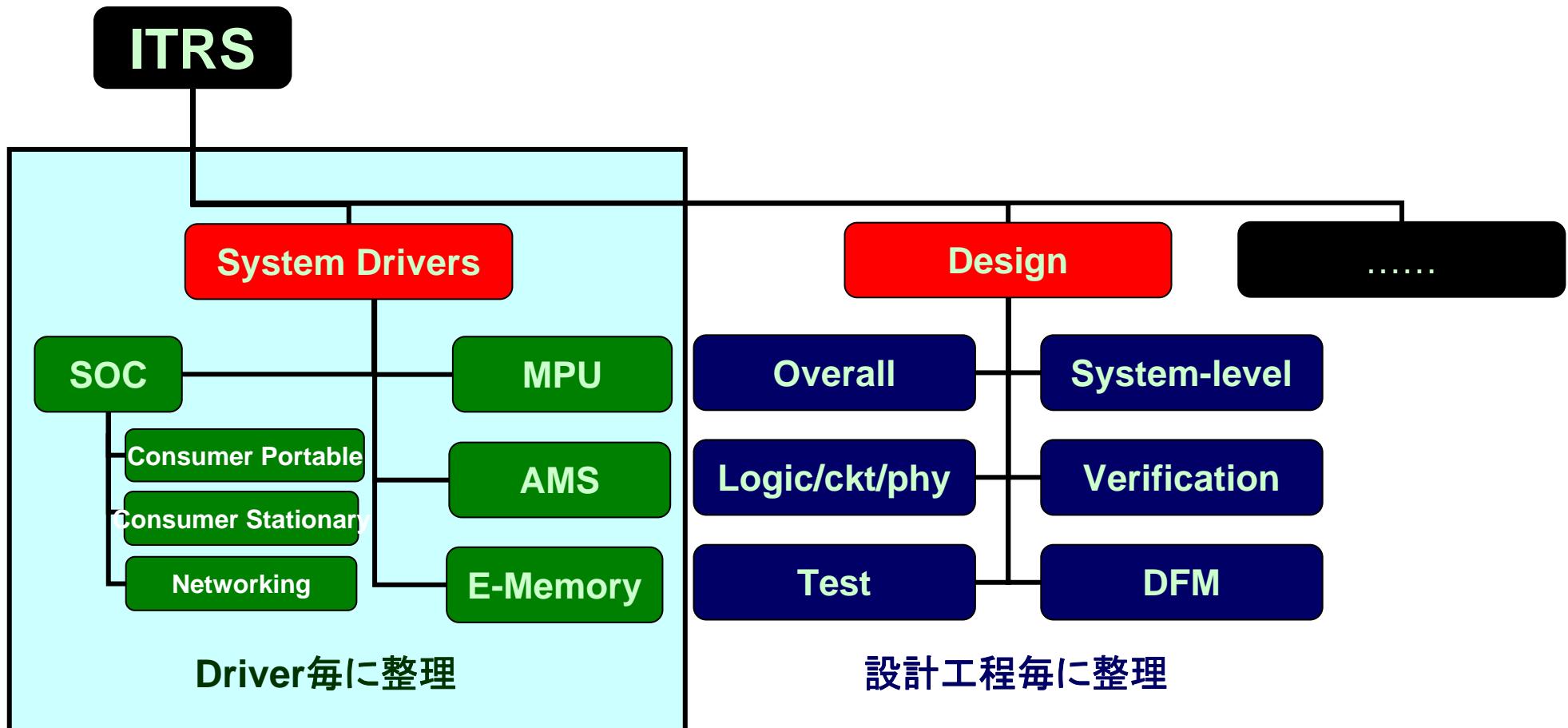
◆国内活動

- SOC機能検証技術の進展と今後の取り組み

◆まとめ

System Drivers章

製造技術および設計技術をドライブする製品分野と
分野毎の仕様や要求を定義



Design章の構成

～設計技術のロードマップ～

General Challenges

シリコン複雑度とシステム複雑度への対応

Productivity

Power

DFM

Interference

Reliability

Key Design Challenges

5つの大きな課題

Mapping

System
design

Logic/circuit
Physical D

Design
verification

Design
Test

DFM

目標を定量化するための枠組み(=設計工程)

ITRS2010 Overview

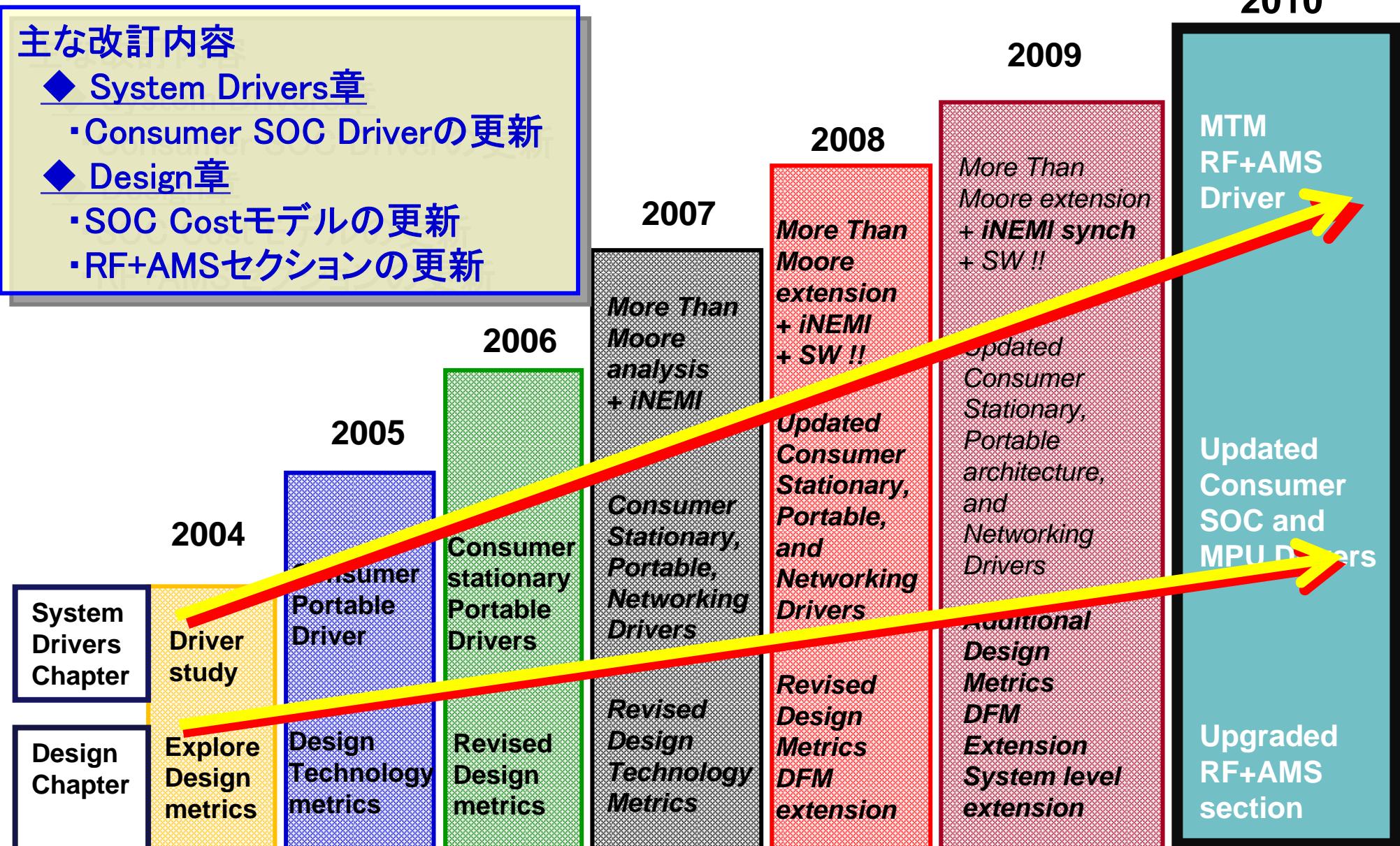
主な改訂内容

◆ System Drivers章

- ・Consumer SOC Driverの更新

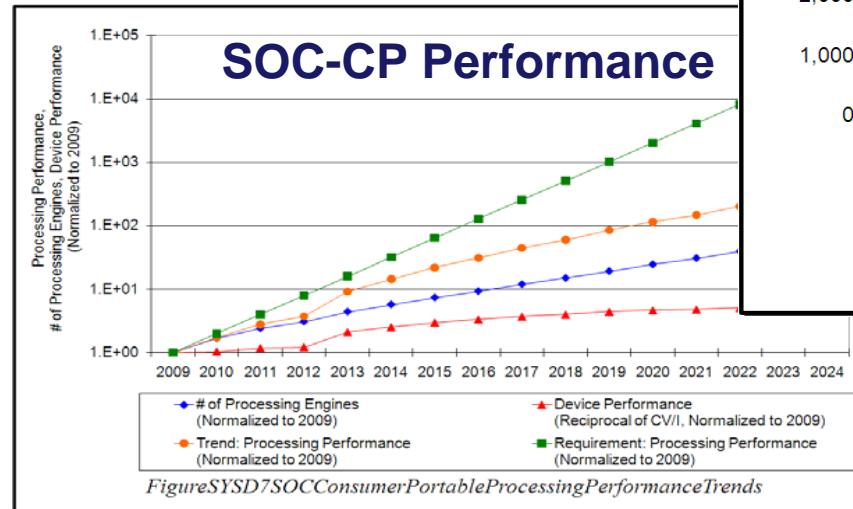
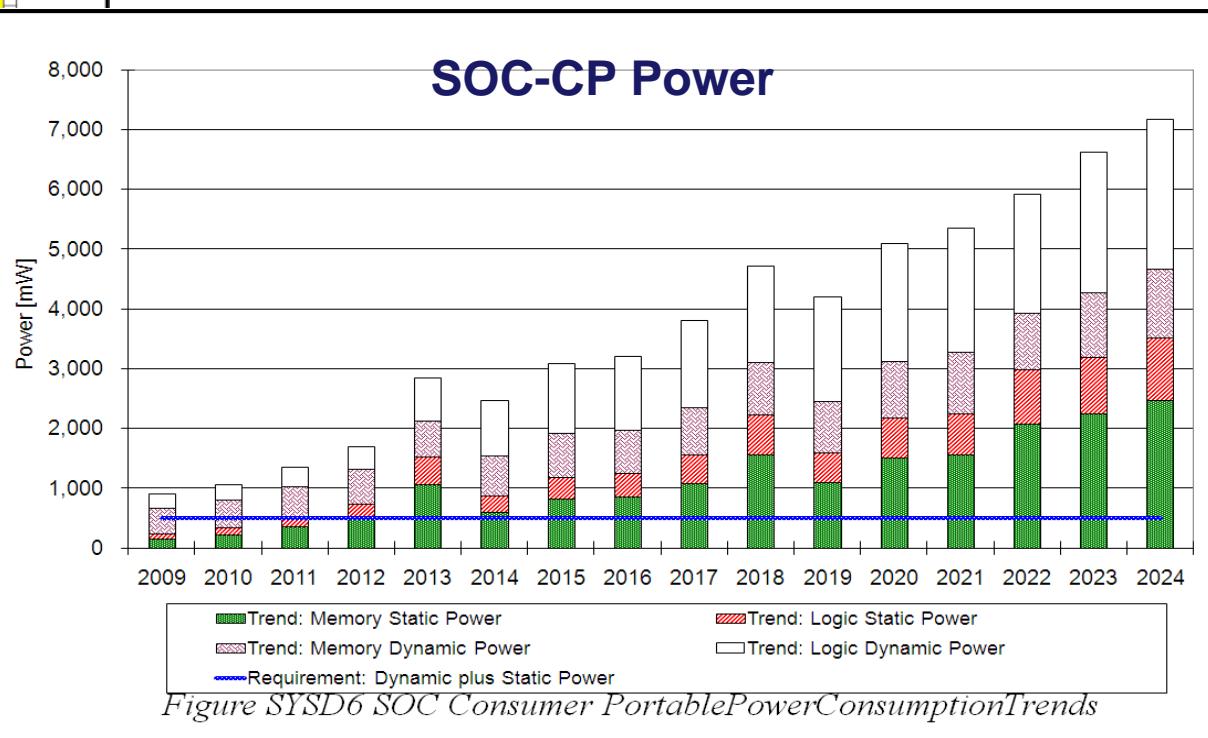
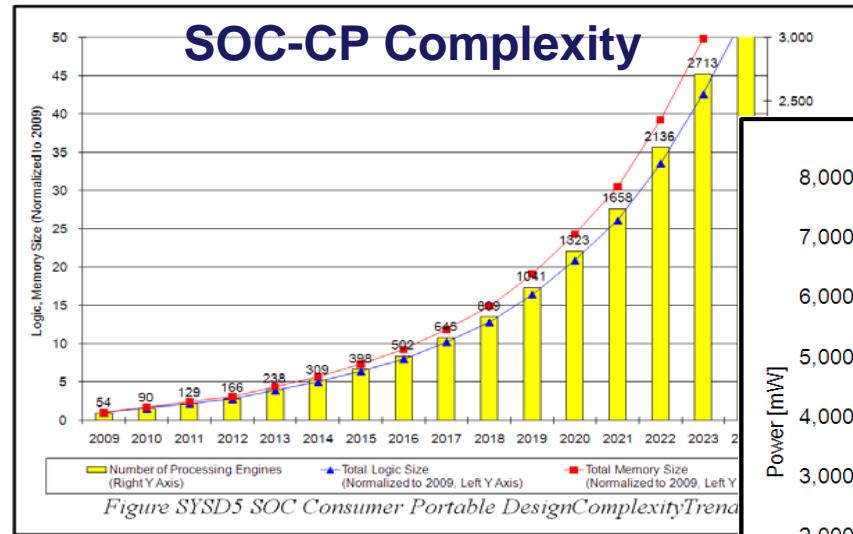
◆ Design章

- ・SOC Costモデルの更新
- ・RF+AMSセクションの更新



System Drivers章トピックス

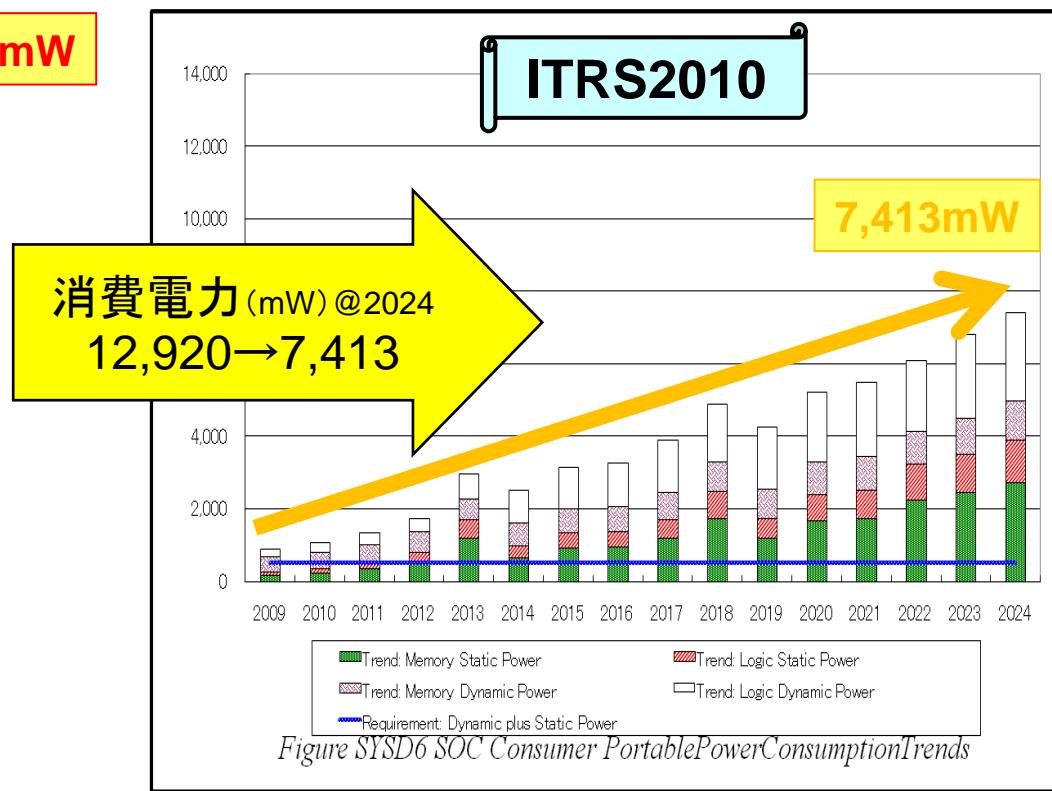
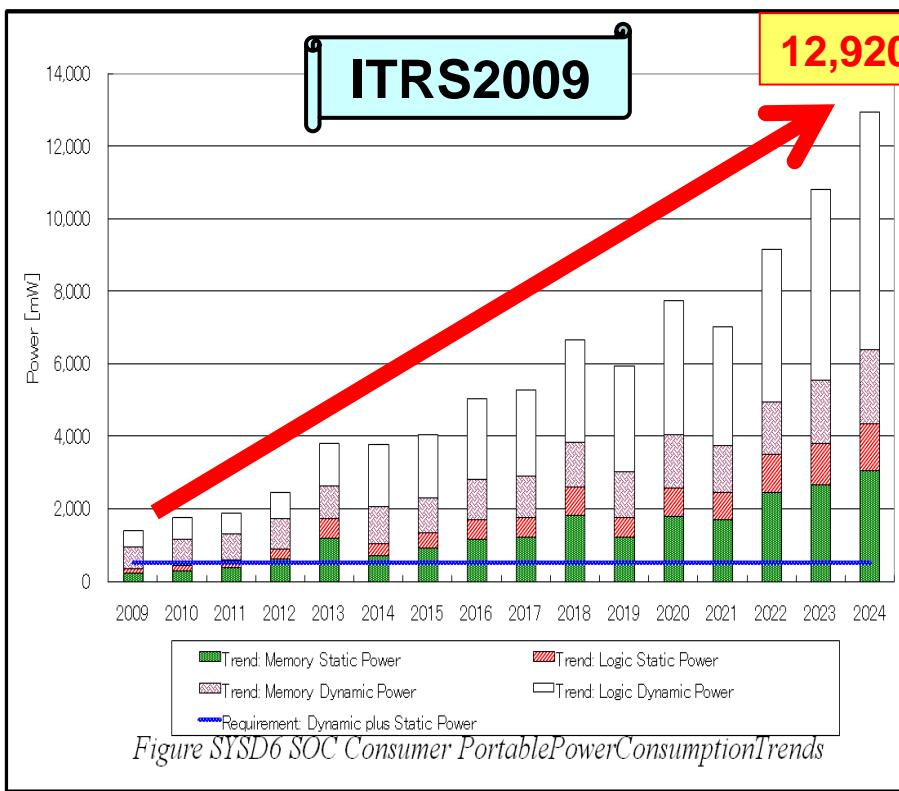
◆Consumer SOC DriverへのPIDS等の変更内容反映



System Drivers 章トピックス

◆消費電力トレンドは、PIDSのLOPのVDDテーブルの変更で、消費電力の増加が抑制

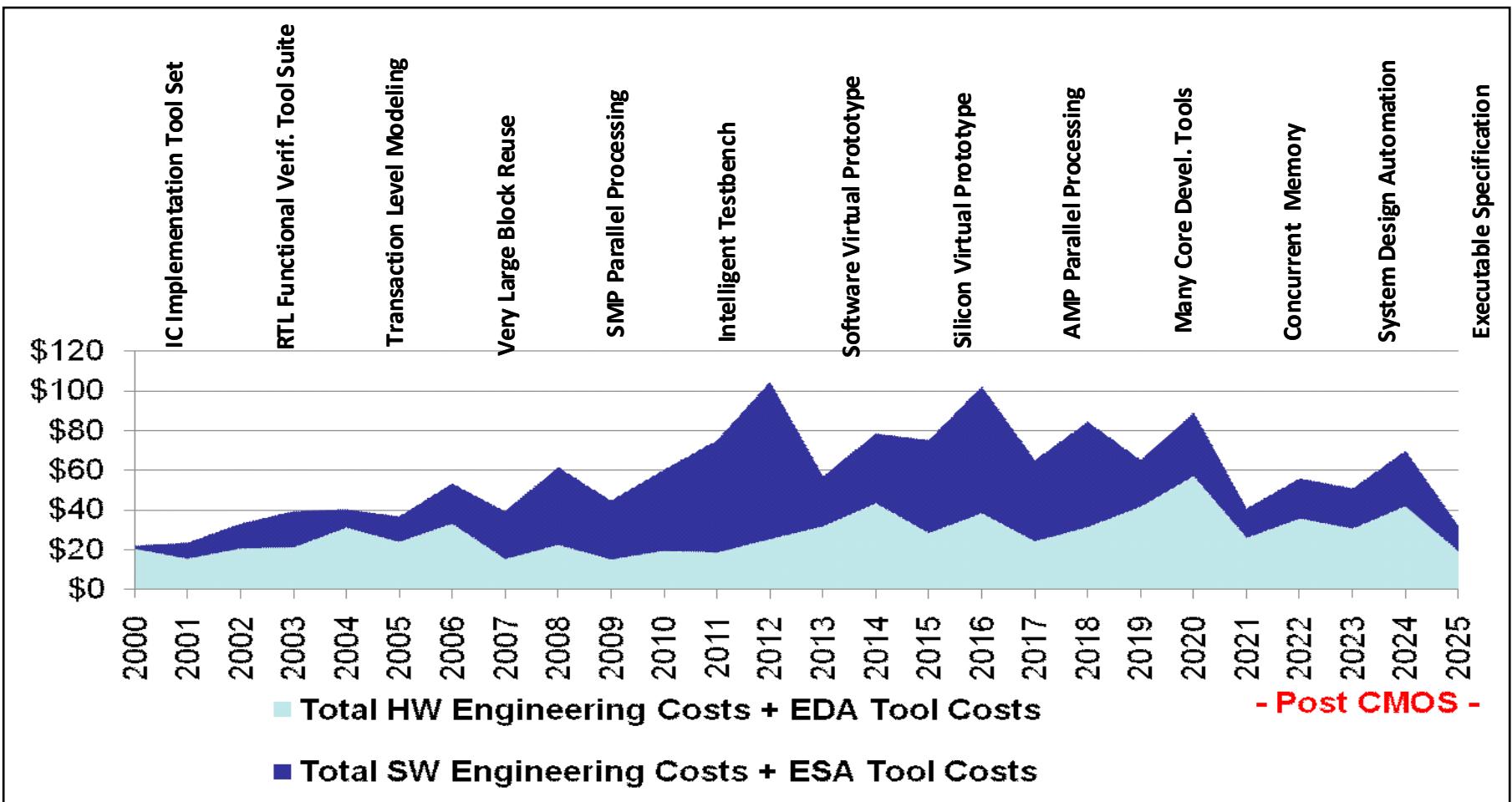
SOC-CP Power



Design章トピックス

◆ Design Costチャートを更新

- ・大きな変更はなく、年代を拡張(2000から2025まで)
- ・システム/SWレベルやPost CMOSの技術開発が重要



ITRS2011の計画

1. 設計章

- ITRS2009掲載の“設計コストチャート”と同様な“消費電力チャート”を検討・作成し、追加する。
- 3D /TSVに関する項目を記載する。
- DFMセクションを改訂する(信頼性保証設計を追加する)
- Verificationセクション、L/C/Pセクションの見直しを行う

2. システムドライバー章

- MPU周波数ロードマップの平坦化(影響度を調査・分析しながら)
- MtMのドライバーとして、RF-AMSを適用したSiP-SOCを新規作成する

3. クロスカット活動、その他

- PIDS: 設計ドリブンな要求定義(項目)を増やす
- 3D/TSV: 設計章で記載する項目の検討と準備
- 議論の継続: A&P, Interconnect, Test

目次

◆はじめに

- ワーキングメンバ、スコープ、ミッションなど

◆国際活動

- ITRS2010アップデートなど

◆国内活動

- SOC機能検証技術の進展と今後の取り組み

◆まとめ

国内活動の概要

◆目標

- SOC機能検証の課題を分析し、実現すべき技術要求と解決案を提示

◆スコープ

- HW/SW分割後の機能仕様からRTLまでのハードウェア機能検証
- 上位レベル検証を検討のスコープに追加して検討。

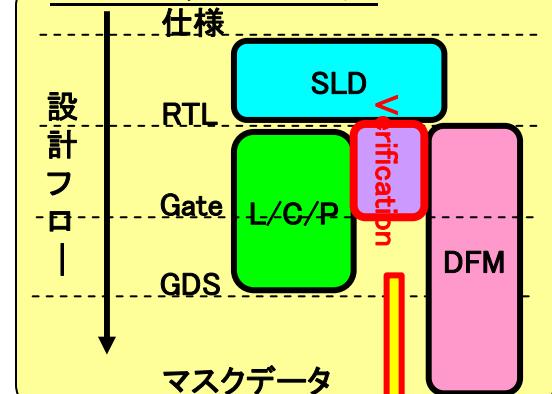
◆検討手順

- 各社からヒアリングした課題を分析
- ”課題に対する現在、および今後の取り組み”を検討
- 2007年度の解決策に対する進捗状況を確認

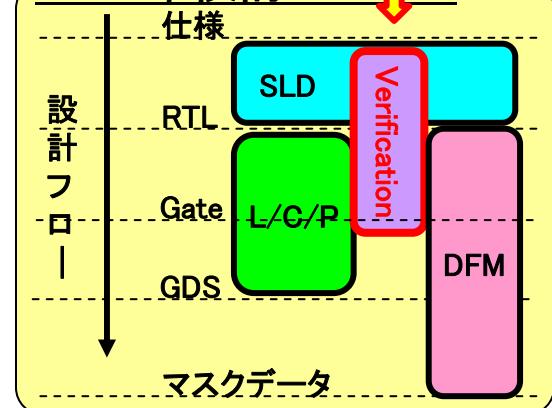
◆提言

- 今後の解決に向けての取り組みを深耕、提言のまとめを実施。

WG1のスコープ



2010年検討スコープ



検証技術の課題変化の分析

- 微細化、複雑化に伴う設計難易度の増加がSOCの機能検証にどのような影響を及ぼしているか？
- 設計現場の現在の状況を抽出し、2007年度の国内活動で検討した内容との変化を分析する。

国内活動	
2007年度	SOC設計技術ロードマップの詳細化/定量化 ・論理検証と物理設計の2分野で「設計生産性向上」の観点でロードマップを詳細化/定量化
2010年度	検証技術の課題変化と分析 ・設計レベルによる課題分類 ・2007年度課題に対する現在、および今後の取り組みを検討

2007年から2010年にかけて、重要課題における進展は？

2007年に検討した技術要求/解決策

設計 レベル	重要課題	2010年の技術要求／解決策
戦略 策定	検証対象の状況に応じた最適な検証戦略の策定	最適な戦略を作るための共通な手順が確立される
	最適な検証のマネジメント	ベストプラクティスに基づいたマネジメント・ガイドラインが策定される
	IPモデルの整備と検証品質	IPモデルの整備と検証品質
	機能検証スキルを持つ人材／専門家の育成	機能検証スキルを持つ人材／専門家の育成
仕様 設計	明確で誤解がない仕様の作成	共通な仕様表現方法が確立される
	網羅的な検証項目の抽出と検証順序の最適化、および仕様変更への対応	仕様と検証項目の対応が確認できるようになる
検証 実行	シミュレーションの高速化	2桁以上の高速化が実現される
	デバッグの効率化	アサーションが自動生成され、プロトタイプの内部状態が容易に観測できる
	設計レベル間検証	等価検証が実用化される
	設計初期段階での性能検証	RTLで性能や消費電力の確認が出来る

2010年に検討した技術要求/解決策

	重要課題	重要課題
戦略 策定	検証対象の状況に応じた最適な検証戦略の策定	検証対象の状況に応じた最適な検証戦略の策定
	最適な検証のマネジメント	
	IPモデルの整備と検証品質	
	機能検証スキルを持つ人材／専門家の育成	
仕様 設計	明確で誤解がない仕様の作成	明確で誤解がない仕様の作成
	網羅的な検証項目の抽出と検証順序の最適化、および仕様変更への対応	網羅的な検証項目の抽出と検証順序の最適化、および仕様変更への対応
検証 実行	シミュレーションの高速化	シミュレーションの高速化
	デバッグの効率化	デバッグの効率化
	設計レベル間検証	設計レベル間検証
	設計初期段階での性能検証	

課題抽出の結果、共通する課題が殆どであり、併せて検討を実施。

スコープをHW/SW分割後の機能仕様からRTLまでのハードウェア機能検証で実施したため対象外。

検証対象の状況に応じた最適な検証戦略の策定

2007年検討時の2010年の技術要求／解決策

最適な戦略を作るための共通な手順が確立される

2007年からの進展

- OVMとVMMの統一(UVM化)による検証環境の標準化
- アサーション、フォーマル検証の利用拡大
- ソフト先行開発のためのVirtual Prototypeの利用拡大

阻害要因

- 客観的な基準がなく、個人のスキルに依存して非体系的
- 検証担当者による検証品質のバラツキで品質保証難

プラットフォームベース設計を推進し、UVMを用いた
検証環境の再利用推進を図る

IPモデルの整備と検証品質

2007年検討時の2010年の技術要求／解決策

IPモデルの整備と検証品質

2007年からの進展

- IPベンダー提供の検証IPラインアップは増加
- OVM/VMMなどの検証環境ベースの検証IPも増加

阻害要因

- IPベンダーから提供されるテストベンチでは不十分
- IPベンダー毎にデリバラブルが異なる
- IPの品質を保証する共通規格がなくIPベンダー依存

IP及び検証IPの品質認定の制度化の仕組みが必要
(共通規格、品質の見える化、デリバラブルの共通化等)

機能検証スキルを持つ人材／専門家の育成

2007年検討時の2010年の技術要求／解決策

個々の機能検証スキルを持つ人材／専門家の育成

2007年からの進展

- 機能検証教育プログラムの整備(国内各社共同開発のガイドライン等)
- ダイナミックアサーション検証の利用は一般的になってきた
- 検証専任者の育成を各社で実施

阻害要因

- 新検証技術の出現により、新たなスキルが必要
- 人材育成の内容は各社各様

検証技術者を育成する体系だった仕組みが必要

明確で誤解がない仕様の作成

2007年検討時の2010年の技術要求／解決策

共通な仕様表現方法が確立される

2007年からの進展

- UMLなどの共通の高位記述フォーマットの活用開始

阻害要因

- 仕様品質は人のスキルに依存しており規定がない
- SOCが大規模化し、検証対象も大きくなっている

記述ルールの策定と詳細仕様の抽象化

網羅的な検証項目の抽出と検証順序の最適化 および仕様変更への対応

2007年検討時の2010年の技術要求／解決策

仕様と検証項目の対応が確認できるようになる

2007年からの進展

- 制約付きランダム検証が普及
- 検証対象にバグを埋め込みチェックする新たな技術が出現

阻害要因

- 网羅度を上げる技術はツール制約が多い
- 検証項目の抽出は検証者のスキルに依存

検証項目の自動抽出環境と網羅度を客観的に確認する仕組みが必要

シミュレーションの高速化

2007年検討時の2010年の技術要求／解決策

2桁以上の高速化が実現される

2007年からの進展

- CPU性能向上、ツール性能向上、分散・並列処理化によるシミュレーション能力向上
- モデルの抽象度向上による高速化

阻害要因

- 分散・並列化による計算機やツールコストの高騰
- 抽象化モデルの準備にコストがかかる
- 既存IPの抽象化モデルの作成が困難

抽象化モデルの整備とスタティック検証の活用

デバッグの効率化

2007年検討時の2010年の技術要求／解決策

アサーションが自動生成され、プロトタイプの内部状態
が容易に観測できる

2007年からの進展

- アサーション、フォーマル検証の利用拡大
- スケマティックビューア系のデバッグツールは成熟

阻害要因

- アサーション検証になれていないために普及が不十分
- 検出エラーの優先付けや対処は個人のスキル依存

検証済みIPの再利用と、ノウハウの共有化

設計レベル間検証

2007年検討時の2010年の技術要求／解決策

等価検証が実用化される

2007年からの進展

- CとRTL間の等価検証技術が使われ始めた

阻害要因

- CとRTL間の等価性検証には、論理の深さや回路の規模など制約が多い
- 等価検証ができない部分は依然、動的シミュレーション実施

等価検証ツールの高性能化、カバー範囲の拡張

国内活動のまとめ

■検証戦略の策定

- プラットフォームベース設計推進し、UVMを用いた検証環境の再利用推進
- IP品質認定の制度化の仕組みが必要
- 検証技術者を育成する体系だった仕組みが必要

■仕様設計

- 記述ルールの策定と詳細仕様の抽象化。
- 検証項目の自動抽出環境と網羅度を客観的に確認する仕組みが必要。

■検証の実行

- 抽象化モデルの整備とスタティック検証の活用
- 検証済みIPの再利用と、ノウハウの共有化
- 等価検証ツールの高性能化、カバー範囲の拡張

EDA技術は日々向上しているものの、
人材育成・抽象化・IP再利用
などの検証環境の整備と体系だった仕組みが重要

目次

◆はじめに

- ワーキングメンバ、スコープ、ミッションなど

◆国際活動

- ITRS2010アップデートなど

◆国内活動

- SOC機能検証技術の進展と今後の取り組み

◆まとめ

まとめ

国際活動

- ITRS2010では主に以下の改定を実施

System Drivers章:

Consumer SOC Driverの更新を実施。

Design章:

SOC Costモデルの更新、RF+AMSセクションの更新

国内活動

- 「SOCの機能検証技術の進展と今後の取り組み」をテーマに活動

- －各社からヒアリングした課題を分析。
- －”課題に対する現在、および今後の取り組み”を検討。
- －2007年度の解決策に対する進捗状況を確認。

今後の解決に向けての取り組みを深耕、提言のまとめを実施。

END

補足資料

検証技術検討の進め方

各社における
情報集約

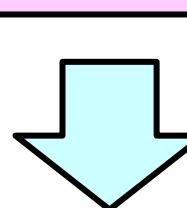
WG1識者による
各項目毎への
まとめ

検証戦略の策定



仕様設計

A社、B社、
C社、D社



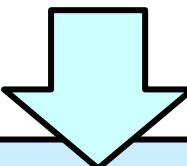
明確で誤解のな
い仕様作成



1.XXXXX
XXXXXX
XXXXXX
XX

検証の実行

A社、B社、
C社、D社



シミュレーション
の高速化



1.XXXXX
XXXXXX
XXXXXX
XX

機能検証スキルを持つ人材／専門家の育成

現在の状況

- UVMの検証環境はあるが、それを使いこなせる人材が少ない。
- ダイナミックアサーションは使われるようになってきたが、フォーマル検証については十分に使いこなせていない。
- 検証専任者の育成の取り組みは行われているが、組織的に行われているわけではない。

現状の問題点

- 今までの検証のやり方と考え方が違うため、新たなスキルセットが必要。
- 検証技術は色々あるが、使いこなせるエンジニアが少ない。

それがなぜ問題なのか

- 人材育成がすすまないことで、適切な人材がアサインできないため、SoCの大規模・複雑化による検証期間の増大と、検証品質が確保できない。

現在の取り組み

- 機能検証教育プログラムの整備を実施(STARCのIP機能検証ガイドラインなど)することで、検証技術者のスキル向上を図っている。

今後何が必要か、必要と予測されるか

- 機能検証技術者を育成する体系だった仕組み(例えばHDL検定のようなスキルを計れる仕組みなど)が必要。

検証対象の状況に応じた最適な検証戦略の策定

現在の状況(2010年)

- 各ブロック・コア、トップ、システム全体の特性(データパス系、制御系など)について、過去の実績や経験に基づいて、それぞれの検証担当者が検証戦略を策定し、それに基づいた検証環境を構築している。
- 検証戦略を実現する手法としての、アサーション(フォーマル含む)やUVMの利用は進んできている。
- システム全体検証を目的に、ソフト先行開発のためのVirtualPrototypeの利用も進んできている。

現状の問題点

- 個人の経験やスキルの依存しており、客観的な規準がなく非体系的である。

それがなぜ問題なのか

- 規準がないため、検証担当者のスキルによる一貫性がなく、検証品質にばらつきがでるため、品質の保証が難しい。

現在の取り組み(Short Term Solution)

- OVM/VMMがUVMに統一されたことにより、標準化を進めている。

今後何が必要か、必要と予測されるか(Long Term Solution)

- QCDを考慮した検証戦略規準の策定。
- プラットフォームベースの設計を推進し、UVMを用いた検証環境を再利用する仕組みを一般化することで、検証品質のバラツキを抑える。

明確で誤解がない仕様の作成

現在の状況(2010年)

- 仕様を書くのをUMLなどの共通の高位記述フォーマットで記述し始めている。
- 仕様が曖昧なままに開発される(仕様のFix遅延)。
- 設計部署により仕様の質にはらつきがある。「仕様」に記載する言葉も、設計部隊毎にまちまち。質を改善しようとしている部署と何も改善されていない部署あり。

現状の問題点

- 「仕様書」と「回路」の不整合が発生している。設計途中での仕様変更が、ドキュメントに反映されていない。安易な仕様の追加/変更も行われる。設計されたものが仕様となってしまう。
- 仕様品質は、人のスキル次第。最低限記述すべき仕様が記述されていない。最低限記述すべき項目の規定もない。解像度(詳細度)が異なり、可読性が悪いものもある。
- SoCの規模が大きくなると全項目の仕様レビューを全担当で行う事が困難。
- 機能記述から機能合成する設計事例が増えており、仕様変更に対して従来のRTL設計より回路が大きく変更され、再検証のための時間が多くかかるようになった。

それがなぜ問題なのか

- アーキテクチャを勘違いし、実装および検証してしまった場合、最悪の場合は、CHIPを作った後に問題発覚し、リファインに直結する
- 「仕様」解釈ミスによる 無駄な作業の発生。例えば あいまいな仕様書があいまいな設計の仕様を生む。また、作成元の意図した仕様と異なった仕様で使われてしまう。
- 仕様が明確にならないため、製品設計自体の仕様/設計も日程が遅れる。

明確で誤解がない仕様の作成(続き)

現在の取り組み(Short Term Solution)

- 各種「仕様書」の再定義(仕様の明確化)。各「仕様書」作成を含んだ設計フロー化(手順の明確化)。
- 「仕様書」のフォーマット整合化。記述方式にローカルルールを作成し、統一性を高めている(仕様書に書くべきことの明確化)。
- 商品企画から詳細仕様の抽象度を上げる。例えばC言語で記述する

今後何が必要か、必要と予測されるか(Long Term Solution)

- 統一的な記述ルールの策定
- 各仕様書と回路、テストベンチ、検証項目(アサーション等)のチェーンデータベース環境の構築(回路、テストベンチ、仕様などが紐づいている仕組み)、仕様変更が与えるインパクトの明確化
- 仕様の妥当性を確認するための環境。動く仕様書としてCモデル作成と確認。UMLなどの仕様記述からSystemCなどの機能記述の自動生成。仕様矛盾の自動チェック。

網羅的な検証項目の抽出

現在の状況(2010年)

- 検証項目の抽出は、設計者と検証者がそれぞれ、仕様書から検証項目を抽出している。
- 網羅度を客観的に確認する手段は十分になく、現状はレビューで項目を確認している。また、加えて、検証専任者が検証項目を追加することで、網羅度を上げている。
- 技術的には、制約付きランダム検証による検証が使われている。スタティックフォーマルで網羅度を上げる技術に関しては、ツールの制約(回路規模、使い勝手)で、まだまだ使えていない。また、検証対象にバグを埋め込み検証項目をチェックするツールも一部出てきている。

現状の問題点

- 検証項目の網羅度を客観的に確認する仕組みや技術がなく、検証者のスキルに依存している。

それがなぜ問題なのか

- 網羅度が過剰だと開発コストの増加になり、一方網羅度が低いと不具合(バグ)を混入させるリスクが増える。

現在の取り組み(Short Term Solution)

- (『現在の状況(2010年)』に記載のとおり) レビューによる項目の確認や検証専任者による項目追加で、網羅度を上げている。

今後何が必要か、必要と予測されるか(Long Term Solution)

- 検証項目の自動抽出環境と網羅度を客観的に確認する仕組みが必要。

IPモデルの整備と検証品質

現在の状況(2010年)

- IPの品質が保証されていない
- IPベンダが用意するテストベンチの内容が貧弱なときがあり、導入後追加する
- ブラックボックスと内部参照可能なIPが流通している

現状の問題点

- IPの品質を保証する仕組みがない
- IPモデルに対する標準化が無く、ベンダによって提供物がまちまち
- ブラックボックスIPの品質検証をどう測定するか不明

それがなぜ問題なのか

- 共通規格がないため、導入資料から品質を判断できない
- 機能モデルや検証ベンチ追加を行うと設計コストや時間がかかる
- 品質に不安が残る

現在の取り組み(Short Term Solution)

- 社内レビューを実施している
- 追加で品質確認作業を実施している
- 未使用機能の明確化を実施している

今後何が必要か、必要と予測されるか(Long Term Solution)

- IP品質認定(共通規格、品質確認結果)の制度化
- IPモデル提供ガイドライン策定(Must/Wantで提供するファイル種類や内容の明確化)

シミュレーションの高速化

現在の状況(2010年)

- システムシミュレーション： モデル抽象度の向上による高速化
- ハードウェアシミュレーション： FPGA/Emulatorなどの活用
- RTLレベル ソフトウェアシミュレーション： CPUパワーの向上、分散・並列処理、ツール性能の向上。

現状の問題点

- 既存IP(RTL)からのCモデル作成が困難。現状のRTL→C変換では性能不足。
- テストベンチを合成可能にしたり、専用トランザクタを作成する必要がある。
- 分散・並列処理化しているが、逐次処理があると性能のネック。高速化は計算機の数と性能に依存。

それがなぜ問題なのか

- 既存IPのCモデル作成に追加コストが発生、また十分な仕様書が存在しない場合がある。
- 環境構築とデバッグに時間が掛かり、立上げが遅れる。
- CPU台数分の性能向上しか得られず高速化が不十分。計算機、EDAツールのコストが高くなる。

現在の取り組み(Short Term Solution)

- RTL→C変換ツールの改良を実施中。
- Cで書かれたテストベンチを高位合成にて変換することで環境構築の時間を短縮している。
- 市販ツールの分散シミュレーションは、順次展開中。ブレードサーバ等の導入による計算機コストの削減、EDAツールのクラウド化。

今後何が必要か、必要と予測されるか(Long Term Solution)

- IPについては、IPの供給に併せてCモデルの供給が必要。そのためには、Cモデルの整備を行い目的毎に使用するCモデルの標準化が必要。(Un-timed, Loosely-timed, Approximately-timedなど)
- シミュレーション環境とFPGA/Emulator環境間の検証環境移植の更なる自動化が望まれる。
- スタティック検証(フォーマル検証等)の活用を広げていく。

デバッグの効率化

現在の状況(2010年)

- アサーション、フォーマル検証を使用。
- スケマティックビューアー系のデバッグツールは成熟している。Lint系ツールでは大量の擬似エラーが発生するが、基本は地道(愚直)にレビュー等による確認作業を実施している。

現状の問題点

- 設計者がAssertion設計になれていないために、普及しない。
- 検出エラーの優先順位付けやエラーへの適切な対処は担当者のスキル依存。

それがなぜ問題なのか

- デバッグの効率化が進まない。
- デバッグの効率および品質が、担当者のスキルに依存してばらつく。

現在の取り組み(Short Term Solution)

- EDAベンダーによるAssertionの自動生成ツールのサポート。ツール使用のノウハウをEDAベンダー、または各社で蓄積されている。
- デバックノウハウの蓄積、技術者の教育を進めている。

今後何が必要か、必要と予測されるか(Long Term Solution)

- IPとしての再利用可能なAssertionの組み込みの手法の確立が必要。蓄積したノウハウをツールで自動化する。
- 検証IP、検証用ベンチの再利用により、検証環境を短期で整備しすることでデバッグを効率化する。

設計レベル間検証

現在の状況(2010年)

- CとRTL間の差異(バグ)の発見には、動的シミュレーションを行っている。

現状の問題点

- CとRTL間の等価性検証には論理の深さや回路の規模などに制約が有る。

それがなぜ問題なのか

- 等価性検証でカバーできない範囲は、動的シミュレーションを実施する必要が出て来る。

現在の取り組み(Short Term Solution)

- 等価性検証と、動的シミュレーションを併用
- コーディングルールの策定と適用

今後何が必要か、必要と予測されるか(Long Term Solution)

- 等価性検証ツールの整備、カバー範囲の拡張。