

# デザイン

## スコープ

設計技術(Design Technology)は、電子機器システムの**構想設計**、**実装設計**および**設計検証**を実現する技術である。設計技術はツール、**ライブラリ**、**プロセス特性のモデリング**および**設計手法**で構成される。電子機器システム設計者の設計構想と設計目標は、設計技術によって製造可能かつ試験可能な形に変換される。設計技術の役割は、製造能力を最大活用したコスト効率の良い生産により、半導体産業の利益および成長を実現することである。2005 年版 ITRS では、設計 ITWG(International Technology Working Group)が、総括ロードマップ技術指標(Overall Roadmap Technologies Characteristics)に基づいて、クロック周波数、レイアウト密度、消費電力などの観点で、**デザイン章**と**システムドライバ章**を作成した。設計技術の課題と設計者からの改善要求は両章に記載されている。また本章の読者には 2003 年版 ITRS デザイン章を参照することを推奨する。2003 年版デザイン章は現在でも有益な内容を多く含む。

2005 年版の主要メッセージ **設計コストは半導体技術ロードマップに対する最も大きな脅威である。**

設計対象を、プログラマブルな汎用プラットフォーム上に実現するか、あるいは個別 IC を新規開発するか、また機能をソフトウェアで実現するか、ハードウェアで実現するかという選択を最適に実行することが、コスト競争力をもたらす。開発費用 NRE(Non-recurring Expenses)はマスク、プローブカード費用で数 100 万ドルに達する。更に設計 NRE として、設計ミスによるチップ再試作のための製造 NRE を含めると数千万ドルに達する。技術革新のスピードアップは製品寿命を短縮し、LSI 顧客にとって time-to-market が重要目標となる。また、製造に要する期間が数週間であることに対して、設計と設計検証期間は不確定であり、数か月または数年を必要とする。半導体生産投資の減価償却と関連産業の投資利潤により、巨額の半導体投資サイクルが運営されている。2003 年版 ITRS には**設計生産性ギャップ(設計クライシス)**が記述された。すなわち 1 チップに実現可能なトランジスタ数の増加が、それらを有効に利用する設計能力の進歩を超えている。にもかかわらず、プロセス技術への投資額は設計技術への投資額に比較してはるかに多大である。

・良いニュース：このような中で設計技術の改善は継続的に進められている。Figure 17 は、**システムドライバ章**に記載された低消費電力 SOC-PE(SOC Power Efficient)の設計コストを示している。2005 年におよそ 1800 万ドルの設計コストが、1993 年から 2005 年までの設計技術の改善が行われなかった場合には、9 億ドルに達していたことが分かる。(分析の詳細は Appendix を参照)

・悪いニュース：ソフトウェア開発は組み込みシステム開発費用の 80%を占める。また LSI 測定コストは製造コストに比べて指数関数的に増大している。マイクロプロセッサ開発においては設計検証エンジニアが設計エンジニアよりも多く必要となる。2005 年の現在、われわれはこのように多数の設計技術課題に直面している。

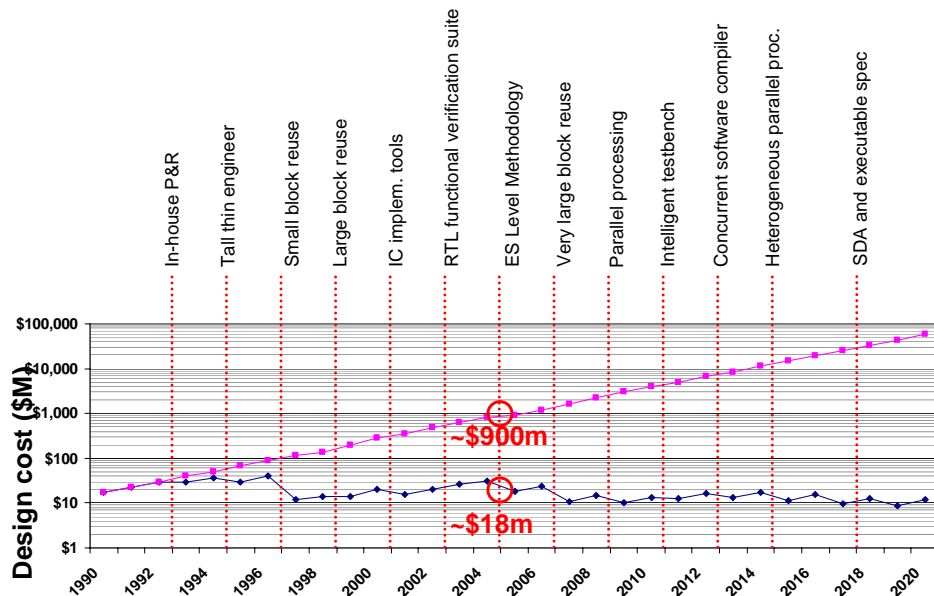


Figure 17 Impact of Design Technology on SOC PE Implementation Cost

本章では、始めに設計技術における複雑度の課題であるシリコン複雑度およびシステム複雑度について考察する。続いて設計技術領域の全体に関わる 5 つの横断的な課題を紹介する。即ち、設計生産性、消費電力、製造容易性、寄生効果による干渉、および故障を起こさない高信頼設計である。設計技術ロードマップとして、初めて定量的に記述した設計技術要求表(Requirements Table)と、その解決技術表(Solutions Table)を示す。またFigure 19aに示すように、設計フローに基づいて設計技術の課題を分析する。即ち、設計工程、システムレベル設計、論理/回路/物理設計、設計検証、テスト設計およびDFM(製造容易化設計)を詳述する。<sup>1</sup> これらの技術課題についてはEDA業界のマネジメント、研究開発部門および大学研究機関といった対応組織ごとに分析する。更にシステムドライバであるMPU、SOC、AMS(Analog Mixed Signal)、メモリの分野別に課題分析する。ここでは現在のEDA技術およびマーケット規模を反映して、MPUとSOCの設計課題が詳述される。最後にAMS設計に特化した設計技術を紹介する。これらを含めた取り組みが特定用途向け、特定システムドライバ向け設計技術の例示となる。

設計技術ロードマップは製造技術ロードマップとは異なる。製造技術ロードマップでは物理的な法則および材料物質によって明確に定義された限界に対して、いかに目標達成するか、目標精度を実現するかを定義する。対照的に設計技術ロードマップは、計算処理時間の限界や未確定な設計目標のもとで、いかに最適化するのが主要テーマとなる。最適化においては、レイアウト密度、動作周波数、消費電力、テストビリティあるいは設計期間のように多数のトレードオフが求められる。このため設計技術の質的な評価は、設計手法や設計対象商品に依存する。更に設計技術ロードマップはITRSの技術世代への関連性は高くない。ITRS技術世代が必要とされる要素技術が実現した時に設定されるのに対して、設計技術では個々技術における革新が設計生産性や設計の質的改善をもたらす。

<sup>1</sup> AMS設計課題についてはシステムドライバ章(AMSドライバ)に記載される。測定装置および測定技術はテスト章で述べられる。テスト章では更に BIST(Built-in self test)を含めたテスト容易化設計技術が詳述される。

## 全体的な課題

設計技術は 2 種類の複雑度に直面する。 - シリコン複雑度およびシステム複雑度 - である。これらは ITRS 製造技術ロードマップに従う。

シリコン複雑度は、プロセススケーリング、もしくは新規の材料、デバイス/配線構造の導入と関連がある。以前は無視できた多くの現象が、現在では設計の正確さと価値に大きな影響を及ぼす。

- \* デバイスの寄生効果および電源/しきい値電圧の理想モデルから外れたスケーリング (リーク、パワーマネジメント、回路/デバイス革新、電流供給)
- \* 高周波デバイスでの配線 (雑音/干渉、シグナルインテグリティ解析とマネジメント、基板とのカップリング、クロス・カップリングによる遅延変動)
- \* 製造ばらつき (統計的なプロセスモデリングと特性抽出、歩留り、リーク電力)
- \* 製造ハンドオフの複雑度 (レチクル改良、マスク描画と検査方法、NRE(Non-Recurring Engineering)コスト)
- \* デバイス性能と関連のある、グローバル配線性能のスケーリング(通信、同期)
- \* 信頼性の低下 (ゲート絶縁体のトンネリングおよびブレイクダウン特性、ジュール熱およびエレクトロマイグレーション、単発的な擾乱、フォールトトレラント)
- \* プロセスばらつき (ライブラリ特性抽出、アナログとデジタル回路性能、エラートレラント設計、レイアウト再利用、信頼性が高く予測可能なインプリメンテーションのための設計基準)。

シリコン複雑度は、長年利用された設計規範を次の点から危うくする。 1) システム全体に渡る同期設計は、消費電力制約、および製造ばらつき対策コストの点から、不可能になる。 2) CMOS トランジスタはその動作上、より大きな統計的ばらつきの影響を受ける。そして、3) トランジスタと配線が 100%動作するチップの製造は、法外に高価になる。利用可能な実装設計の手法(たとえば、一般的な用途においてソフトウェアでプログラム可能なカスタム LSI)では、簡単に 4 桁もの性能低下を招く(たとえば GOps/mW)。また、不適切な設計マージンや設計抽象化、あるいは設計手法の誤った選択では目標特性を達成できない。これらの課題は、より広い設計領域をカバーできる経験豊富な設計者に加え、個別の設計技術間(論理合成とシミュレーション、論理設計とレイアウト設計など)の継続的な統合を要求する。

システム複雑度は、プロセス微細化により可能となるトランジスタ数の指数関数的な増加と関連するが、消費者からの多機能化、低コスト化、および Time to market 短縮要求によって拍車がかかる<sup>2</sup>。多くの課題は設計生産性ギャップと同意語である。これに加えて、設計対象となるシステムの変化や、異種機能の混載などの課題が、システムレベルの SOC 設計に伴い発生する。仕様設計および設計検証は、特に複雑な動作をする場合、大きな課題となる。設計対象の商品価値、品質およびコストのすべてにおいて、トレードオフが生じる。(単純化した例: クロック周波数に対するムーアの法則 (Moore's Law) は、クロック周波数と設計期間(Time to market)の間のトレードオフとして、およそ 1% のクロック周波数改善が 1 週間の設計期間に相当する。)

システム複雑度による設計技術課題には、次のものが含まれる。

- \* 再利用 - 階層設計、特にアナログ/ミックスドシグナルのための混載 SOC 設計(モデリング、シミュレーション、検証、機能ブロックのテスト)の支援
- \* 設計検証とテスト - 仕様設計、検証を考慮した設計、混載 SOC のための検証再利用、システムレベルお

<sup>2</sup> 家電業界では周知である "Law of Observed Functionality" は、トランジスタ数が指数関数的に増加しても、システムの価値は一次関数的にしか増加しないと述べている (T. Claassen, "The Logarithmic Law of Usefulness", Semiconductor International, July 1998 を参照)。同様に System Drivers 章では、MPU の面積増加ほど性能は改善しない (ポラックの法則) と述べている。

#### 4 デザイン

およびソフトウェア検証、アナログ/ミックスドシグナルの検証および新構造デバイスの検証、セルフテスト、高度なノイズ/ディレイ違反テスト、LSI テスタのタイミング制約、テスト再利用)

- \* **コスト重視の設計最適化** - 製造コストのモデリングと分析、品質指標、ダイ-パッケージ-システムレベル同時最適化、フォールトトレラントやテストビリティなど複数のシステム目標に関しての最適化)
- \* **組み込みソフトウェア設計** - 予測可能なプラットフォーム・ベースのシステム設計手法、ハードウェアとネットワークにつながれたシステム環境間での協調設計、ソフトウェア検証と分析)
- \* **信頼性の高い設計のためのプラットフォーム** - 多数の回路方式に対する予測可能なチップ設計、実装設計のための設計上位レベルでのハンドオフ)、そして
- \* **設計プロセス管理** - 設計チームの大きさ、および地理的な分布、データ管理、協力的な設計支援、「システム化された設計」サプライチェーン管理、指標、および継続的なプロセス改良)

Table 12 Overall Design Technology Challenges

Challenges $\geq 32$ nm	Summary of Issues
Design productivity	System level: high level of abstraction (HW/SW) functionality spec, platform based design, multi-processor programmability, system integration, AMS co-design and automation  Verification: executable specification, ESL formal verification, intelligent testbench, coverage-based verification  Logic/circuit/layout: analog circuit synthesis, multi-objective optimization
Power consumption	Logic/circuit/layout: dynamic and static (leakage), system and circuit, power optimization
Manufacturability	Performance/power variability, device parameter variability, lithography limitations impact on design, mask cost, quality of (process) models  ATE interface test (multi-Gb/s), mixed-signal test, delay BIST, test-volume-reducing DFT
Reliability	Logic/circuit/layout: MTTF-aware design, BISR, soft-error correction
Interference	Logic/circuit/layout: signal integrity analysis, EMI analysis, thermal analysis
Challenges < 32 nm	Summary of Issues
Design productivity	Complete formal verification of designs, complete verification code reuse, complete deployment of functional coverage  Tools specific for SOI and non-static-logic, and emerging devices  Cost-driven design flow  Heterogeneous component integration (optical, mechanical, chemical, bio, etc.)
Power consumption	SOI power management
Manufacturability	Uncontrollable threshold voltage variability  Advanced analog/mixed signal DFT (digital, structural, radio), “statistical” and yield-improvement DFT  Thermal BIST, system-level BIST
Reliability	Autonomic computing, robust design, SW reliability
Interference	Interactions between heterogeneous components (optical, mechanical, chemical, bio, etc.)

ATE—automatic test equipment    BISR—built-in self repair    BIST—built-in self test    DFT—design for test  
EMI—electromagnetic interference    ESL—Electronic System-level Design    HW/SW—hardware/software  
MTTF—mean time to failure    SOI—silicon on insulator

同時に、シリコン複雑度とシステム複雑度の課題は、**設計プロセス複雑度の指数関数的な増加**を含む。この複雑度の課題を扱うため、一般に設計技術は、より複雑な設計目標および設計制約の最適化と解析を同時に実現し、設計再利用および設計最適化のための製造コストを付加的に考慮することを認め、そして、組み込みソフトウェア設計や製造インタフェースといった付加的な領域も含めねばならない。シリコン複雑度とシステム複雑度の膨大な広がり、それ自体がまた、設計技術と EDA 業界のロードマップ化への課題である。5 つのクロ



スカット・課題 - 1)生産性、2)消費電力、3)製造力、4)干渉および、5)エラートラント - が与えられるが、これらの解決策候補(Potential Solutions)は設計技術の全分野にまたがり、かつ、この背景には設計コストについての課題が横たわっている。

設計生産性は、システムと設計プロセスの複雑度に密接に関連しており、そして当然ながら設計コストにも影響を与えるが、この設計生産性は短期的にも長期的にも最大に困難な課題であり、そして他の4つの課題からも影響される。

- \* 消費電力は緊急で短期的な課題である、性能ドリブンなダイナミック消費電力の課題から、ばらつきに支配される、リーク電力のより長期的な課題にシフトしつつある。
- \* 製造容易性は、大量のチップを許容できるコストと経済的に実現できるスケジュールで製造できる能力のことである。この製造容易性は、リソグラフィ装置の限界や制約に第一に影響されてきたが、今や様々な形態でのばらつきによって、設計のあらゆる局面の長期的で困難な課題になってきている。

## 課題 1 - 設計生産性

設計コストの指数関数的な増加を避けるため、チップに設計される機能の全体的な生産性を、技術ノードごとに2倍以上にしなければならない。そのため、設計、検証およびテストにおいて、再利用生産性(マイグレーションや AMS(Analog Mixed Signal)RF コアの再利用を含む)を、技術ノードごとに2倍以上にしなければならない。設計生産性の向上における考慮すべき点は、次のとおりである。1) 検証の改善(現状、危機的な状態にあるボトルネックである) 2) より高位レベルでのシステム設計のハンドオフを支援する、信頼できかつ予測可能なシリコン・インプリメンテーションの実現 3) SOC 生産性に対する最も重要な課題として現れた、組込みソフトウェア設計の改善 4) 特に MPU 混載において、巨大かつ分散した組織で様々な供給元からの設計ツールによる作業での、生産性の改善 5) SOC および AMS システムドライバによって要求される、アナログおよびミックス・シグナル(AMS)設計とテストの自動化方法。これらの改善は、正規化された設計品質の測定基準が必要である。これは、設計品質、設計 NRE コスト、製造 NRE コスト、製造により変動するコスト、半導体製品価値の関数であらわされる。安定性、予測性および相互運用のような設計技術品質の測定基準は、開発されかつ改善されなければならない。新しい設計技術の市場投入までの期間は、標準化、相互運用のためのプラットフォーム化、および設計技術の再利用などにより、短縮されなければならない。

## 課題 2 - 消費電力

プレーナ型 CMOS デバイスの非理想的スケーリングは、配線材料やパッケージ技術のロードマップとともに、パワーマネジメントと電流供給に関連した様々な課題を提示する。それらの課題を以下に記す。

- 1) 全体的なロードマップ技術指標およびシステムドライバ章に基づく推定によれば、HP(High Performance) MPUの消費電力は、アセンブリおよびパッケージング章の中で述べられた、高機能単一チップパッケージの電力限界を超過するとともに、許容された電力密度である  $250\text{W}/\text{cm}^2$  をも超過する。SOC-LP(Low Power and low cost) PDAドライバは、ロジック規模および処理能力が指数関数的に増加し続ける時には、平均およびスタンバイ電力の考慮が必要である。設計技術は、システムドライバ章のTable12 で示されるように、結果として生じるパワー・マネジメント・ギャップの対策に取り組まねばならない。
- 2) 電力密度の増加は、信頼性と性能に対する熱の影響を悪化させる。他方で、供給電圧の減少は、スイッチング電流およびノイズを悪化させる。これらの傾向は、チップ上の配線リソース(たとえばアセンブリやパッケージのロードマップに照らして、バンプ数や保護膜の開口サイズを決定し IR ドロップをコントロールする)、ATE(Automatic Test Equipment)能力、およびバーンインの枠組みを圧迫する。
- 3) 高性能、低動作電力、低待機電力を統合したデバイスを実現するには、多くの自由度 - マルチ  $V_t$ 、マルチ  $T_{ox}$ 、マルチ  $V_{dd}$  が単一のコアに共存すること - を同時に活用できる電力最適化が要求される。同時に、アーキテクチャ、オペレーティング・システム、およびアプリケーション・ソフト・レベルにおける電力最適化も必要である。
- 4) リーク電力は、ゲート長や酸化膜厚やしきい電圧など主要なプロセスパラメータにより、指数関数的に変動

する。このことは、スケーリングとプロセス変動の両方が、深刻な課題となることを示す。

### 課題 3 - 製造容易性

「技術の壁(Red bricks)」 - 既知の解決策が存在しない技術必要条件 - は、ITRS の全領域にわたり顕著に見られる。一方で、ITRS の一つの技術領域内での解決が不可能な課題は、設計技術の適切な介入もしくは協調で、(より経済的に)解決できる可能性がある。将来の技術ノードの実現可能性は、このような「技術の壁の共有」に拠ることになる。以下に、いくつかの例を示す。

- 1) テスタ装置のコストおよび処理限界に関する問題は、新規の故障モデル(例えば、クロストークやパス遅延の検証用途)のより速やかな採用により、解決が図られるであろう。本故障モデルには、対応する自動テストパターン生成(ATPG(Automatic Test Pattern Generation))およびビルトイン・セルフテスト(BIST(Built-In Self Test))技術も含まれる。
- 2) システム・インプリメンテーション・コスト、性能検証、および全体的な設計 TAT は、チップ・パッケージ・ボード最適化および解析技術のみならず、システム・イン・パッケージ設計技術により、改善される可能性がある。
- 3) リソグラフィ、PIDS(Process Integration, Devices and Structures)、フロントエンドプロセス(FEP)、および配線技術の、CD コントロール(Critical Dimension Control)に対する要求は、製造ばらつき下における正確さを保証できる新しい設計技術により、緩和される可能性がある(たとえば、ばらつきを考慮した回路設計、規則的なレイアウト、タイミング構造最適化、および静的な性能検証)。
- 4) 製造 NRE コストは、マスク生産と検査フローへの、より知的なインタフェースにより、削減される可能性がある。

### 課題 4 - 干渉

グローバル配線のスケーリングで既に試みられている、リソースを有効活用する通信と同期方式は、ノイズと干渉により、より一層妨げられる。論理設計、回路設計、およびフィジカル設計において広く普及しているシグナル・インテグリティ・メソッドは、明らかに 100nm ノードまでは拡張できるものの、実用的な限界に差しかかりつつある。これらのメソッドは、長い配線へのリピーター挿入や波形鈍りをコントロールするための規則、インダクタンスを考慮した電源/グランド分配設計、などを含む。ミックスドシグナルや RF コンポーネントのスケーリングと SOC 化については、より柔軟かつ強力なメソッドが要求される。今後の検討課題として、ノイズの無歪限界(特にロー・パワーデバイスやダイナミック回路)、多数の容量結合もしくは誘導結合を含む配線、供給電圧の IR ドロップとグランド電圧のはね返り、デバイスオフ電流への熱衝撃、配線抵抗、基板とのカップリングなどがあげられる。基本的な設計技術の課題は、設計のすべてのレベルにおいて、ノイズと干渉の特性抽出、モデル化、解析および見積り方法を改善することである。

### 課題 5 - 信頼性

生産、検証、テストにかかるコストを劇的に削減するには、デバイスや配線に対し 100%の完成度を求めることを緩めなければならない。このようなパラダイム・シフトは、技術スケーリングが進むにつれて、あらゆる場合において加速される。なぜなら、技術スケーリングが進むにつれて、信号や論理値やデバイスや配線に対し、過渡的もしくは永久的な故障が、より生じやすくなるためである。以下、いくつかの深刻な問題を示す。

- 1) 90nm 以降、単発的な擾乱(ソフトエラー)が、フィールドレベルの製品の信頼性に、深刻な影響を与える。組み込みメモリだけでなく、論理回路やラッチも同様に影響を受ける。
- 2) 現在の加速寿命試験(バーンイン)は、供給電圧の減少により実行不可能になる(指数関数的に長いバーンイン時間が必要となる)。更に、バーンインオープンの消費電力要求も増大する。
- 3) “non-catastrophic”なゲート酸化膜のブレイクダウンなど、原子スケールの影響を反映した、新たな「ソフト」欠陥の基準が必要となる。一般に、システムが大きくなりすぎると最終段階での機能テストが不可能になるため、設計のロバスト性を考慮した自動回路挿入が、より重要となる。とりうる対策としては、冗長論理回路の自動挿入、フォールトトレラントを考慮したオンチップ回路再構成、適応回路もしくは自己修復回路の開発、そしてソフ

トウェアに基づくフォールトトレラントが含まれる。

## 設計技術の課題

この章の残りの部分では、まずデザインメソッドロジの概要にふれ、その後に設計技術の5つの伝統的な領域における定量的課題と解決策候補について詳しく述べる。上述したように、ほとんどの課題は、今日におけるEDA技術と市場のセグメンテーションに映し出され、SOCにマッピングされる。

### デザインメソッドロジ

チップの設計と実装の工程は、様々な技法、ツール群、そして設計者の入力が予想どおりに製造可能な製品となるような有効なメソッドロジを必要とする<sup>3</sup>。必要なツールには相当な注意が払われてきたが、同様に重要な題目である設計メソッドロジはしばしば無視されている。各技術世代はより多くの問題を設計者に考えさせる；したがって、新しい現象を解析するための新しい手法やツールが開発され、設計者がクリティカルな設計上の決定をすることを助けるようでなければならない。更に難しいのは、繰り返しを最小限にするために、問題を考え、設計上の決定を行う最適な順序を決めることである。4つの主要なトレンドが未来の最先端のチップ設計工程とそれをサポートする設計システムの構造を左右する。

**動向 1：緊密な連携** 以前はファイルにより動作する一連のバッチツールで構成されていたデザインプロセスは、同時に動作し、メモリ上で設計データを共有するモジュール式のアプリケーションの集まりへと進化している。最新のメソッドロジでは、最適化のループで遅いファイルアクセスを含むことはもはや不可能であり、設計上の多くの課題のために同時に幾つもの基準を最適化しなければならない。この傾向は、今日ではほとんどの市販の設計システムで見られる。それらの設計システムでは、ロジックの最適化が、先進のチップデザインのタイミングを合わせるための配置や全体配線、タイミング解析と一緒に行われるのである。ノイズの問題を避け、電力消費を最小化し、かつ製造容易性を保証するためには更なる進歩が必要である。Figure 17 はこの傾向を表したものである。左の列はハードウェアデザインプロセスを表している。ここでは配線遅延と合成結果に与える配置の影響を扱うために、合成、タイミングとある程度の配置が組み合わされている。この時代のいくつかの配置配線システムは、配線の密集を少なくするために、限られたロジックの変更をいていた。中央の列は今日のデザインプロセスを示している。ここでは一式の解析と最適化のモジュールが、協調して、性能、消費電力、ノイズ、面積の条件を満足しながら、テスト容易性や製造容易性を保ったチップを製造する。右の列は将来、必要となる設計システムを示している。ここでは条件を満たすシステムを実現するためにハードウェアとソフトウェアがともに解析され、ともに最適化される。

**動向 2：製造のための設計** 製造のためのマスクデータの準備は、デザインプロセスのますますクリティカルな部分となってきている。従来は、「データ準備」のアプリケーションが設計データをマスク作成の情報に変換した。今日では、設計を記述する基本的な形状が、プロセスの変位を修正し、転写容易性をあげるために追加された形状とともに、標準のファイルフォーマットを通してマスク作成に引き渡される。しかし、現在のパラダイムでの欠点が製造の NRE コストを指数関数的に増加させている。まず第 1 に、修正形状 (RET (reticle enhancement technology)、メタルフィル、その他) が焼き付けられるウェーハやマスクコストへの影響の完全な理解なしに挿入されている；そのため、デザインは「過剰修正」される。製造プロセスの新しいキャラクタライゼーションとコストのトレードオフが、よりインテリジェントなデータ準備を可能にするためには必要である。第 2 にマスク検査と修正はマスク作成におけるコストと遅延の大きな要素であるにもかかわらず、やはり設計意図への

<sup>3</sup> 設計メソッドロジは設計者と設計技術者の共同で開発される；それは設計プロセスが制約条件を満たしながら確実にデザインを設計目標に「可能な限り近い」ように作る一連のステップである。設計メソッドロジは設計技術とははっきりと異なるものである。設計技術はメソッドロジを含む一連のステップのインプリメンテーションに関するもので、下の各々の設計技術分野の中で議論されている。全て既知の設計メソッドロジは、1) トップダウン的な計画と探索によるシステム仕様と制約条件の遵守と、2) 物理法則、設計と製造技術の限界、システムコストの限界から派生する制約をボトムアップ的に伝播させたものを組み合わせている。

洞察なしに行われている。各形状に対して同一の許容範囲を満たせようとする努力が無駄に払われている。設計データの重要なポイントをマスク作成プロセスに伝え、製造の複雑性を設計プロセスにフィードバックするための標準的なフレームワークが必要である。そのような設計と製造間の双方向の遣り取りにより、チップ設計と製造の先々のコストを抑えることができる。マスクとファンドリのアウトソーシングの伸びを考えれば、この遣り取りは、業界標準のインタフェースを通して行うべきである。最後に、設計サイクルの全局面において設計を最適化するために製造特性を用いる新しい設計・解析ツールは、最小のトータルコストで設計の意 Figure を実現するために、機能強化された製造ソフトウェアと連動しなければならない。Figure17は製造容易性の影響、例えば先の配線ツールは重要部位の歩留りを上げるため配線を離しておくことに有効であったなどということを示している。今日では、イールドドリヴンレイアウト(yield-driven layout)により関心が集まっており、製造容易性は標準的な設計の尺度となっている。65nm 世代以降では、設計と製造のデータは単一のデータベースで統合されなければならない、それにより設計者は設計上のトレードオフをするときにマスクコストへの影響を早い段階で理解することができるし、製造フローの側でもイールドドリヴンあるいはコストドリヴンの最適化を行うときに設計意図を理解することができる。製造容易性は、消費電力、性能、シグナル・インテグリティとともに、将来の他変数設計最適化の第 1 目標に加わることになるだろう。

**動向 3: 抽象度レベルの引き上げ** 今日、重要な部品はまだデバイスレベルで念入りに作られている一方、多くの設計はより設計生産性をあげるためにゲートレベルで行われており、最新の設計フローでは、レジスタ・トランスファ・レベル(RTL)が設計を記述するために使われている。抽象度のレベルを上げるには、抽象原理を形成する正しい基本概念を見出すことと、その抽象度で考慮されるべきトレードオフと次の下位レベルの抽象度へのマッピングを行うツールを開発することという、並外れたイノベーションが必要になる。設計者の設計生産性の継続的改善のためには、RTL よりも十分に上位の、設計のシステムレベル化の台頭が必要である。



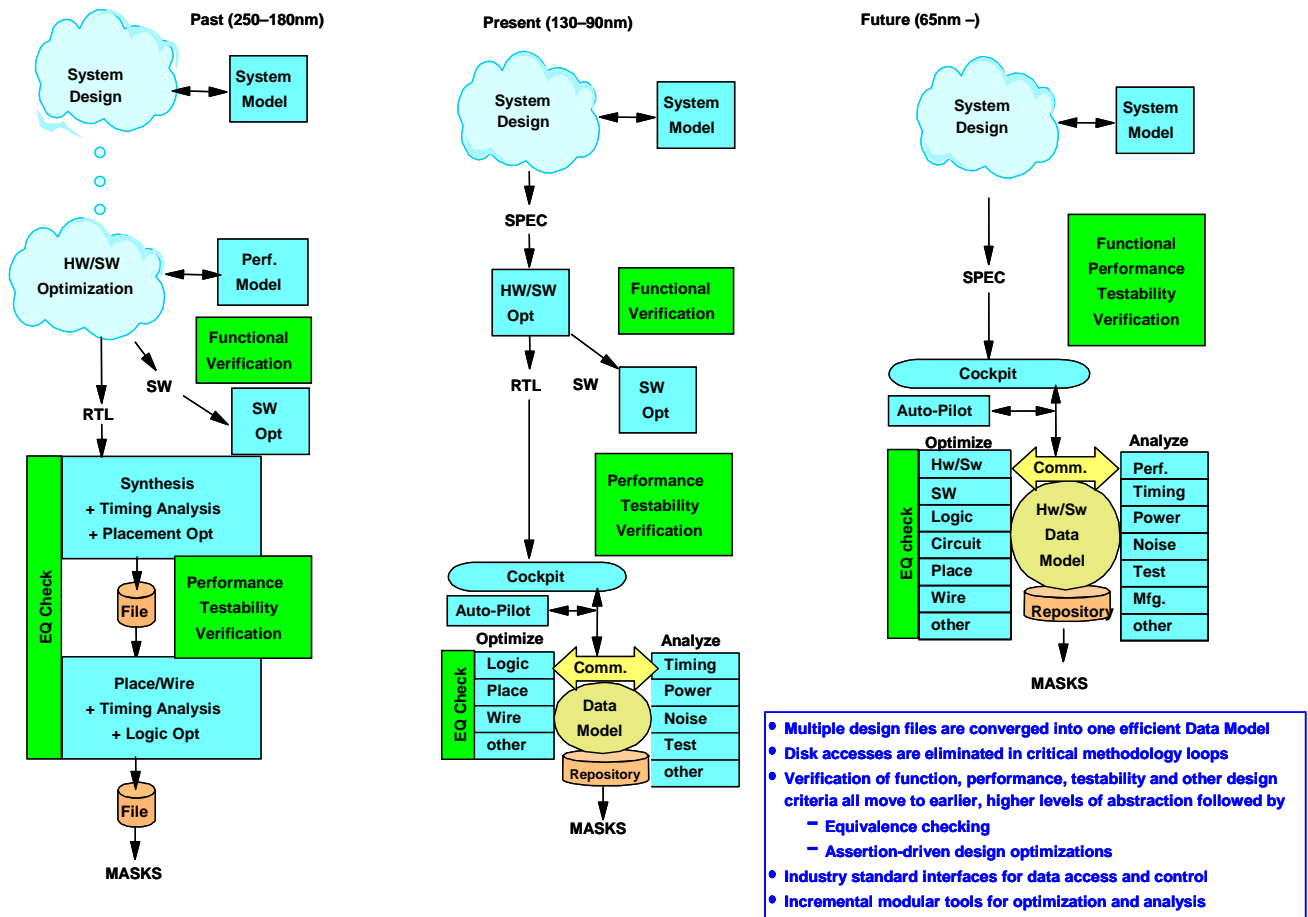


Table 18 Required Evolution of Design System Architecture

抽象度の高度化により、様々な検証方法が可能となり、より早い設計工程での検証ができ、問題を早期に発見することによりマーケット投入までの時間を短縮したり、コストを削減したりすることができるようになる。Table 18 はこの動向（設計技術の進化とともに、緑色の検証の箱が設計工程の上流に推移）を示している。この動向を細かく分類すると次のとおりである。1) 機能検証は、かつてはゲートレベル実装設計がシミュレーション可能になってから開始されていた。完成度や精度の高い早期モデルがまればよかったためである。今日では、機能検証はRTLレベルから開始することができる。後のゲートレベル実装との等価性検証ができたためである。その結果は正確であり、より効果的で、より早期の機能検証を提供する。将来のシステムレベルの抽象度に対しては、“transaction-level”モデリング手法<sup>4</sup>があらわれる可能性が大きい。ローレベルなRTL記述との等価性検証は依然必要とされるであろう。2) 性能とタイミングの検証により早期に製品性能を確認することができるため、再設計の時間ができる。数学的あるいはシミュレーションモデルベースの早期性能検証技術は、早期見積りを提供したが、後のゲートレベルでの確認を必要とした。現在、RTLモデリングはより早期の見積りを提供し、システムレベルモデリングは更に早期のフィードバックを可能にする。（transaction-levelモデルは早期のcycle-accurate性能見積りを提供することができる。）詳細なタイミングの分析は、デバイスモデルが最もよく分かっているゲートやトランジスタのレベルで必ず行われる一方、新たなRTLレベル、システムレベルのタイミング見積りツールは、設計者への早期設計工程での、価値ある早期フィードバックを約束する。設計ツール群の緊密な連携により、設計制約を満たす上での必要な予測が達成でき、設計者にRTL実装設計工程が早期の性能見積りに合うことに対する、より高い信頼を与えることになる。3) テスタビリティ検証は依然として、設計工程での重要なチェックポイントである。歴史的にはゲートレベルでの実装が必要とされてきたが、今日ではより洗練されたツールやself-test手法により、RTLレベルでの設計のテスタビリティに高い信頼を与えるようになった。非常に早期の設計段階でシステムレベルの抽象度での確認を提供するため、同様な進化が必要とされている。

<sup>4</sup> Transaction-levelモデリングでは、データと時間が両方とも抽象化される。不可分な動作が複数のサイクルに対応したり、複雑なデータ転送が単純なリード/ライトコマンドで表されたりする。

**動向 4: 自動化レベルの引き上げ** 歴史的に、新たな抽象度は、より巨大な設計に対する、より効果的なシミュレーションとベリフィケーションを第 1 の目的として、創設されてきた。しかしながら、設計者が設計の意図を新たなモデルによって記述するようになると、合成のような他のツールにとってもチャンスとなる。この動向の 1 つの重要な側面は、後工程でのイタレーションを減らすように、設計ガイドラインが制約ドリブンの最適化に置き換わることである。今日の RTL 設計工程はこのようにして台頭してきた。そして、システムレベルにおいても、同様な進化が必要とされている。この動向も Figure 17 に見ることができる。左側では、より広範囲な性能モデルが、初期のシステム仕様と自動化された RTL 実装設計の間の、形のない間接的な繋がりのところから、生じている。現在の技術世代では、実行可能なシステムレベル仕様が成熟された技術となっているが、将来的にはシステムレベル仕様は、ソフトウェアとハードウェアを包含し、制約ドリブンな実装設計のための支配的な表現とならなければならないであろう。システムレベル設計技術の詳細な要件については、次のシステムレベル設計セクションで、Figure 19b の進化を具体的なフローで示すとともに、述べることにする。

## 設計技術の課題突破

システムレベル設計<sup>5</sup>においては、メソドロジの側面がツールの側面より急速に困難度を増している。シングル・チップ上に巨大な複雑度を持つシステムが実現可能となるが、一方でこれを現実のものとするためには、今日と比べて約 50 倍以上の設計生産性が要求される。この状況、実に我々をひるませる。というのは、異種混交の信頼性の低いデバイスと配線材料を使って、高度に信頼性のある有用なシステムを開発せざるを得ないからであり、これがシリコンの複雑性(ばらつきとか信頼性とかの)の意味するところである。グローバルな同期回路設計は、プロセスばらつきと電力消費の理由によって、非現実的となる。また、チップを横切る信号は、もはや 1 クロックサイクルでは実現できなくなる。したがって、システムレベル設計は、ネットワーキングと分散計算に概念として類似せざるを得ない。例えば、通信システムでは構造設計が先で、次に機能ブロックの設計、そしてそれらが通信バックボーンに統合される。更に、組み込みシステムのソフトウェア割り当てが劇的に増加する、このような HW/SW 協調設計が、現在のシステムレベル設計の大きな技術課題の一つとなっている。

ITRS ロードマップの 2003 年版では、システム・レベル・デザインの技術課題が広く残っており、どれもが非常に複雑なものである。過去 20 年間、設計者はシステムを様々な抽象度のレベルで論証してきた、例えばブロックダイアグラム状態遷移図やプログラムモデルなどである。しかし、そこでの設計自動化ツールの支援はわずかな程度であった。この状況は、設計生産性の進歩が達成されるためには、近い将来に変わらなければならない。HW、SW を含むシステムの仕様、検証、インプリメントを単純化するために、そして、より効率的な設計空間の最適化探索のために、使い慣れた RTL の上位の新しい抽象度のレベルが必要となっている。この新しい抽象度のレベルは、上に述べたトレンドに沿った、以下に列記するような進化や進歩を要求する。

\* **HW と SW の両方の領域における、再利用に基づいた設計** 再利用可能な、高度な機能ブロック(“コア”や intellectual-property (IP) ブロック)は、少なくとも 200% の設計生産性の可能性を提供する。予め検証され、再利用可能なテストが設計の複雑度を低減し、また、再利用可能なソフトウェアのライブラリが組み込みソフトウェアの開発を促進することができる。理想的には、SOC 設計者はブロックダイアグラムを描くと同じくらい容易に、コアを複雑なアプリケーション向きのアーキテクチャに組み立てることができる。実際には、通常、システムにいくつかの新規のコアやソフトウェアが必要になり、やや実

<sup>5</sup> システムレベルでは、シリコンのリソースは抽象的な機能やブロックとして位置付けられている。設計のターゲットは、ソフトウェア(高位のアセンブリ言語での埋め込みコード、コンフィギュレーション・データなど)とハードウェア(コア、ハードウェアされた回路、バス、リコンフィギュラブル・セルなど)から構成される。ハードウェアは、インプリメントされた回路に対応するし、ソフトウェアは、ハードウェアで処理される機能の、論理的な抽象表現に対応している。動作(behavior)とアーキテクチャは設計自由度では独立であり、ソフトウェアとハードウェアはアーキテクチャの構成要素となる。動作の集合体がシステム・ファンクションを定義、一方でアーキテクチャ・ブロックの集合体がシステムプラットフォームを定義する。システム機能からシステム・アーキテクチャ上へのプラットフォーム・マッピングがシステムレベル設計の正に核心であり、また、システムの複雑度の増大や異種混交(アーキテクチャ上や機能上の)によって、困難度を増している。

装設計が遅らせている。例えば、プロセッサコアやマルチメディアコーデックのようなハードウェア IP の再利用は、過去数年においていくらかの進歩を遂げた。それにしてもやはり、再利用可能であることの基準、そして様々なアプリケーションのために特化したカスタマイズ可能なコアは、システム設計工程に広く浸透し、大幅に展開されなければならない。

- \* **プラットフォーム・ベース設計** コアベース設計の拡張は完全なハードウェア“プラットフォーム”を形作る高度に再利用可能なコアのグループを創り出し、更に、SOC 設計工程を容易にする。1つあるいは複数のプログラマブルなプロセッサやリコンフィギュラブルロジックを含む、高度にプログラマブルなプラットフォームにより、新たな SOC を製造せずに、派生品の設計をすることもできる。特定の派生品へのプラットフォームのカスタマイゼーションは、設計空間の探索を行う上での、制約条件になる。すなわち、基本の通信アーキテクチャとプラットフォームのプロセッサの選択は固定され、設計は、確実なカスタマイゼーションのパラメータと、ライブラリからのオプションの IP を選択することだけに限定される。プラットフォーム・ベース設計はまた、HW-SW 分割を必然的に伴っている。HW-SW 分割とは、主要な処理タスクを HW と SW に割り振ることで、システムの性能や、エネルギー消費、チップ内通信バンド幅の消費量、その他システム的な利点に大きな影響を及ぼす。マルチ-プロセッサのシステムでは、“SW-SW”分割と協調設計、すなわち SW のタスクを様々なプロセッサ選択により割り当てることになる。おそらくこれらの決定のうち 80-95% は、プラットフォームや、派生品の SOC が元々固有に持っている独自性により決まり、協調設計による決定は通常、重要な影響を与える少数の機能に対してだけ行われる。プラットフォーム・ベース設計は、ますます設計工程に浸透し続けている。しかしながら、利用可能なプラットフォームの数が増加している中で極めて重要なポイントは、分割決定のサポートや、設計空間の探索や、システムレベルでの異なる目標プラットフォームに自動マッピングするといったことのための、十分なツールサポートが不足しているという点である。
- \* **システムレベル検証** 抽象度の上位化の原則は、システムレベル設計のための、ただ一つの表記法である。C、C++、Java およびその変形による数年にわたる実験の結果として、SystemC が、シミュレーションのためにハードウェアとソフトウェアの相互実行可能なシステムモデルを構築するのに、妥当な形式として近年急速に広まってきた。デザインプロセスのセクションで述べたように、トランザクション・レベルのモデリング手法は高性能なシステム検証を可能にすることを実証した。その裏側では、ソフトウェア言語のハードウェア領域への適用が、エラーや設計欠陥の更なる原因を導いている。そのために、ハードウェアの必要性に重きを置いた、標準化と一般的に認められた機能検証技術が、例えば今、SystemC に対して開発中であるように、もうしばらくの間必要とされる。更に、フォーマル検証手法はこの高い抽象度を活用して、アプリケーションに問題の巨大化を許容する。
- \* **マイクロアーキテクチャ合成** システムレベル仕様のための標準的な形式がシミュレーションや機能検証に採用されるにつれて、その他のツールが出現してきている。システム合成は極めて難しいが、その進展は技術革新の登場の順序から予測が可能である。最初の段階では、マイクロアーキテクチャ仕様の形式でハードウェアを若干高位に表記したもののから、使いものになる RTL 仕様を自動生成するようなものになるだろう。Figure 19a と 19b がこの進化を示している。Figure 19a は典型的な現在の設計フロー (Figure 19a の左部分の実体図) である。ここでは、RTL 以前のマニュアルのステップと、自動化された RTL 実装設計の工程が混在している。Figure 19b は近い将来 ( ~ 2007 ) 必要とされる設計フローを表現している。ここでは、マイクロアーキテクチャ設計結果を RTL にマニュアルでマッピングする工程が、自動化された工程に置き換わっている。

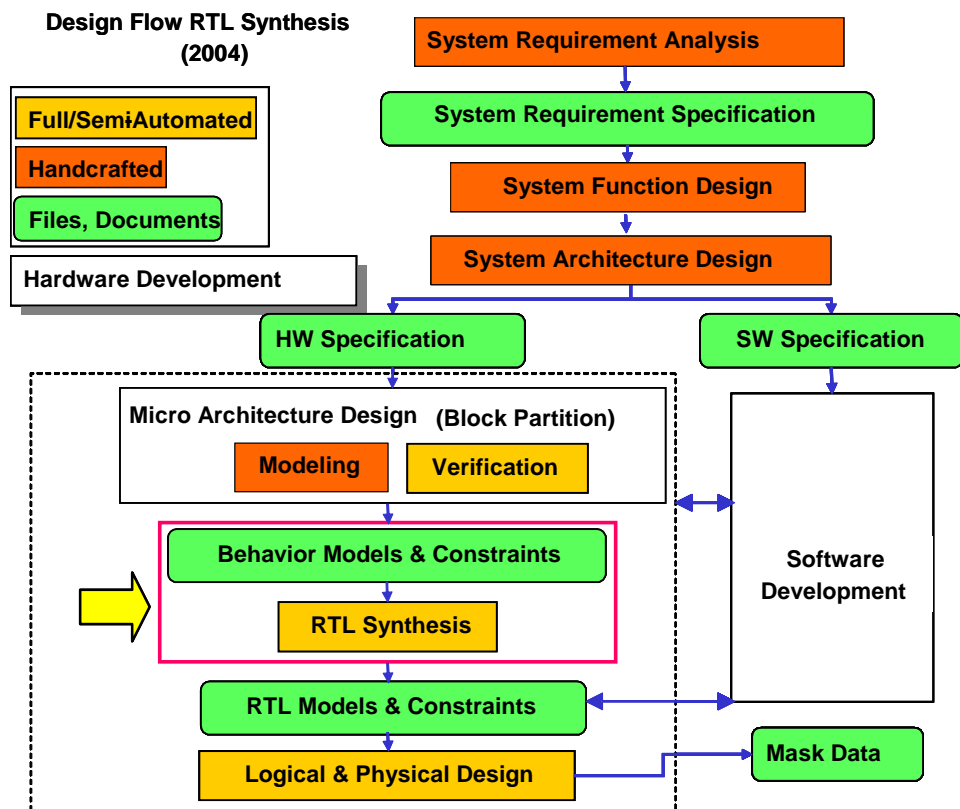


Figure 19a RTL Synthesis for Design Flow in Year 2004

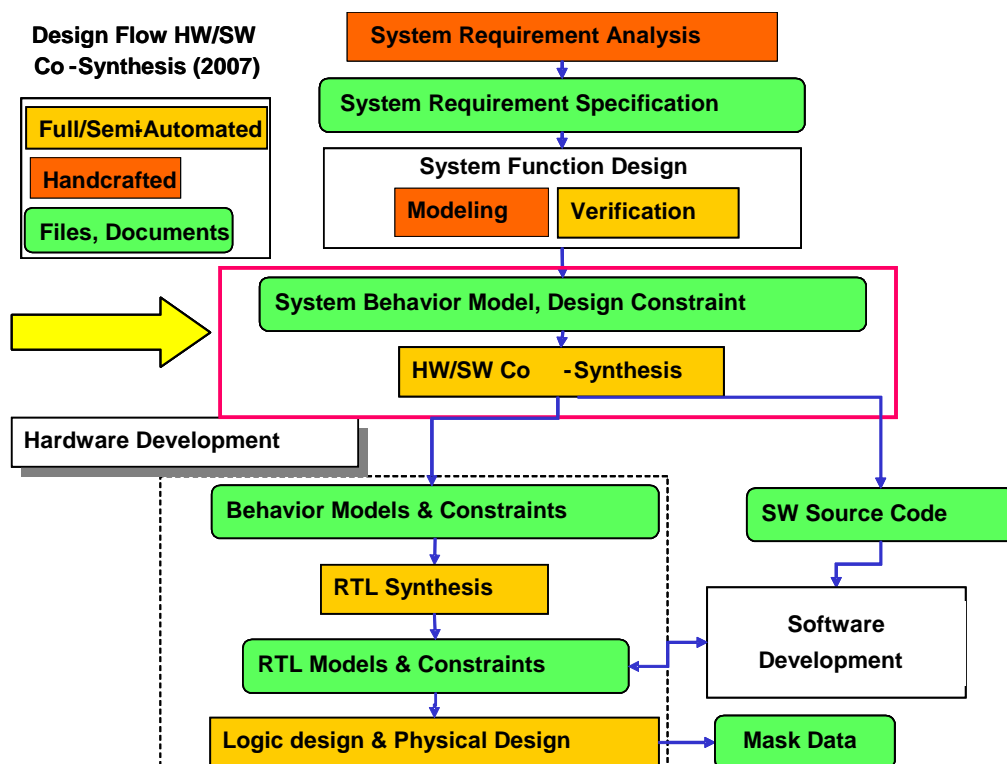


Figure 19b Design Flow in Year 2007



- \* **HW-SW 協調合成** 次に必要とされる進展は、総合的に最適なソリューションを実現するためのハードウェアとソフトウェアの実装を同時に合成する機能である。Figure19 はおよそ 2007 年ごろに必要とされる、この機能に対応する設計フローを示している。ここでは、動作仕様をソフトウェアのプログラムとハードウェアのマイクロアーキテクチャにマニュアルでマッピングする工程が自動化された工程に置き換わっている。HW-SW 協調合成の 1 つの形態は“コプロセッサ合成”である。ここでは、アルゴリズムのソフトウェア記述が解析され、自動的あるいは半自動的に次の 2 つのパーツに分割される。1) 制御構造部分: 標準的な(オンチップの)RISC プロセッサ上で動作する SW で、ほとんどの制御分岐のハンドリングと極めて少ない量の演算計算を行う部分。2) データ処理/データフロー部分: コプロセッサとして HW に実装され、制御 SW を補完する部分。HW 実装により、後者の部分はアプリケーション全体の 90 ~ 95 % まで実行時間を短縮することができる。短縮率は、データフローに対する制御の量と、データ処理をマッピングしたハードウェアの構造の特徴に依存する。コプロセッサ合成は、市販ベースで既に出現し始め、それなりの成功を収めており、ここ数年の HW-SW 協調合成の形態の主流となるだろう。この領域の可能性として重要なことは、暗示的なアプリケーション処理の、統一された(そしてソフトウェアベースの)記述である。これによりマニュアルの過大な努力を要せずに、HW と SW への分割が可能になる。
- \* **高位アナログ記述と合成** より複雑なシステムは、ハードウェアとソフトウェアのコンポーネントだけから構成されるのではなく、アナログをも含めた構成になっていく。それは、ワイヤレスやオートモティブなどのたくさんのアプリケーションにとって、必要不可欠なものである。そのようなシステムを表すことができるにふさわしいシステムレベルの仕様言語や方法論は、ハードウェアとソフトウェアを表現するだけでなく、アナログコンポーネントも含む全体を表現する能力を備えるべきである。アナログコンポーネントの高位記述の自動もしくは半自動合成は、必然的な次のステップの課題ではあるが、必ず成し遂げられなければならないものである。将来、マイクロメカニクのような新しい領域からのコンポーネントを、統合する必要性もまた同様に生じるであろう。

次の表(Table 13)に、次世代以降の技術世代におけるシステムレベル設計に対する定量的な要求を示す。

Table 13a System Level Design Requirements—Near-term Years

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32
<i>Design Reuse</i>									
Design block reuse [1] % to all logic size	32%	33%	35%	36%	38%	40%	41%	42%	44%
<i>Platform Based Design</i>									
Available platforms [2] Normalized to 100% in the start year [3]	96%	88%	83%	83%	75%	67%	60%	55%	50%
Platforms supported [4] % of platforms fully supported by tools [5]	3%	6%	10%	25%	35%	50%	57%	64%	75%
<i>High Level Synthesis</i>									
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	53%	56%	60%	63%	66%	70%	73%	76%	80%
<i>Reconfigurability</i>									
SOC reconfigurability [7] % of SOC functionality reconfigurable	23%	26%	28%	28%	30%	35%	38%	40%	42%
<i>Analog/Mixed Signal</i>									
Analog automation [8] % versus digital automation [9]	12%	14%	17%	17%	24%	24%	27%	30%	32%
Modeling methodology, description languages, and simulation environments [10] % versus digital methodology [11] [12]	53%	55%	58%	60%	62%	65%	67%	70%	73%

Table 13b System Level Design Requirements—Long-term Years

Year of Production	2014	2015	2016	2017	2018	2019	2020
DRAM ½ Pitch (nm) (contacted)	28	25	22	20	18	16	14
<i>Design Reuse</i>							
Design block reuse [1] % to all logic size	46%	48%	49%	51%	52%	54%	55%
<i>Platform Based Design</i>							
Available platforms [2] Normalized to 100% in the start year [3]	46%	43%	42%	39%	36%	33%	32%
Platforms supported [4] % of platforms fully supported by tools [5]	80%	85%	90%	92%	94%	95%	97%
<i>High Level Synthesis</i>							
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	83%	86%	90%	92%	94%	95%	97%
<i>Reconfigurability</i>							
SOC reconfigurability [7] % of SOC functionality reconfigurable	45%	48%	50%	53%	56%	60%	62%
<i>Analog/Mixed Signal</i>							
Analog automation [8] % versus digital automation [9]	35%	38%	40%	43%	46%	50%	52%
Modeling methodology, description languages, and simulation environments [10] % versus digital methodology [11] [12]	76%	78%	80%	83%	86%	90%	92%

Manufacturable solutions exist, and are being optimized  
 Manufacturable solutions are known  
 Interim solutions are known  
 Manufacturable solutions are NOT known

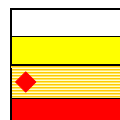


Table13a と b のためのメモ。

[1]この要求はシステム・レベル・デザインに特有でなく、設計一般に対する要求でもある(システムドライバ章で SOC - PE 生産性傾向テーブルを参照)。

定義

新規設計ではなく、既に存在する構成要素を組み合わせて構成する設計部分である。

原理

再利用は設計生産性を支配する主要因の1つであり、システム・レベル・デザインの基本となる重要な概念の1つである。n 年における再利用は、基準とする年から、次の公式により計算できる。

$$\text{reuse}(n) = 1 - (1 - \text{reuse}) \cdot ((1 + \text{pgrowth})^n / (1 + \text{cgrowth})^n)$$

ここに

reuse: 基準とする年における再利用。

pgrowth: 生産性の(予測)平均年率向上率。ただし、再利用による効果は含まない。

cgrowth: 設計複雑度の(予測)平均上昇率

なお、設計者数および設計期間は、一定であると仮定する。

公式の原理は、「製造技術の進歩を完全に利用するためには、生産性の向上(再利用効果を除く)と複雑度の間のギャップを再利用によって達成する」である。

[2] 定義

プラットフォームは、システム構成要素の組み合わせであり、特定のアプリケーション領域の設計容易化を実現する(特定のアプリケーション領域としては、無線電信、自動車、消費者エレクトロニクス/マルチメディア、スモールオフィス・ホームオフィス(SOHO)ネットワークがある。)。システム構成要素は、1個以上のプロセッサ、(リアルタイム)オペレーティング・システム、コミュニケーション・インフラストラクチャー、メモリ、カスタム化可能なアナログおよびデジタル回路、新規論理回路追加のための仮想ソケットである。アプリケーション領域を実現するための基本機能は、組み込み済みの構成要素群により実現できている。システムの差別化は、ハードウェアまたはソフトウェアの新規構成要素の組み込みにより達成する。

原理

プラットフォーム・ベース設計は、再利用を促進するため、設計生産性のための重要なドライバである。加えて、システムレベル仕様設計では、マッピングすべきプラットフォームを規定する必要がある。

[3] 将来には、異なるプラットフォームは、少数に収束すると予測している。製造技術の進歩および集積度向上により、プラットフォームの全体数は減少すると予測するためである。

[4] 定義

特定のプラットフォームに対する(完全な)サポートとは、そのプラットフォーム上での、アーキテクチャ探索、HW/SW 分割、アーキテクチャ/プラットフォーム・マッピング、HW/SW 協調検証、パフォーマンス/エリア/パワー/コストのトレードオフ、HW および SW 合成、HW/SW インタフェース合成を、サポートして自動化する統合設計環境を意味する。

原理

自動化の推進は、システム・レベル・デザインにおける成功の鍵である。

[5] 今日でも、プラットフォーム・ベース・モデリングに対する部分的なソリューションは存在するが、完全に統合できていない。

[6] 定義

見積り結果が、製造した IC における測定結果と一致する度合い。

原理

高位合成技術では、ユーザ定義制約(性能下限、面積上限、など)を満たす高品質の合成結果を得るためには、高精度の見積りが必須である。品質も考慮するならば、更に異なる側面を扱う必要がある。

[7] 定義

SOC、デザインのうち、機能を再構成可能な SW または HW で実現した部分。

原理

システム複雑度は増加し続けるため、将来は、エラーなしでの出荷が不可能になる。したがって、製造後のエラー修復が不可欠となる。加えて、新たなタスクの実現のために既存デバイスのプログラムを修正できるため、リコンフィギュラブル化により再利用が増加する。

[8] 定義

アナログ設計における自動化の達成度。

原理

今日の電子システムの大多数は、アナログ構成要素を組み込んでいる。アナログ/ミックスドシグナルデザインは電子デザインの本質的なそして重要な部分である。したがって、デジタルデザインにおいて、設計生産性の成長を維持または加速するには、アナログ・デザインにおけるより多くの設計レベルでの高い自動化が要求される。

[9] アナログ設計における自動化程度を、デジタル設計における自動化程度に対比した指標。

[10] 定義

アナログ設計におけるメソドロジ、記述言語、シミュレーション環境を、デジタル設計に対比した指標。

原理

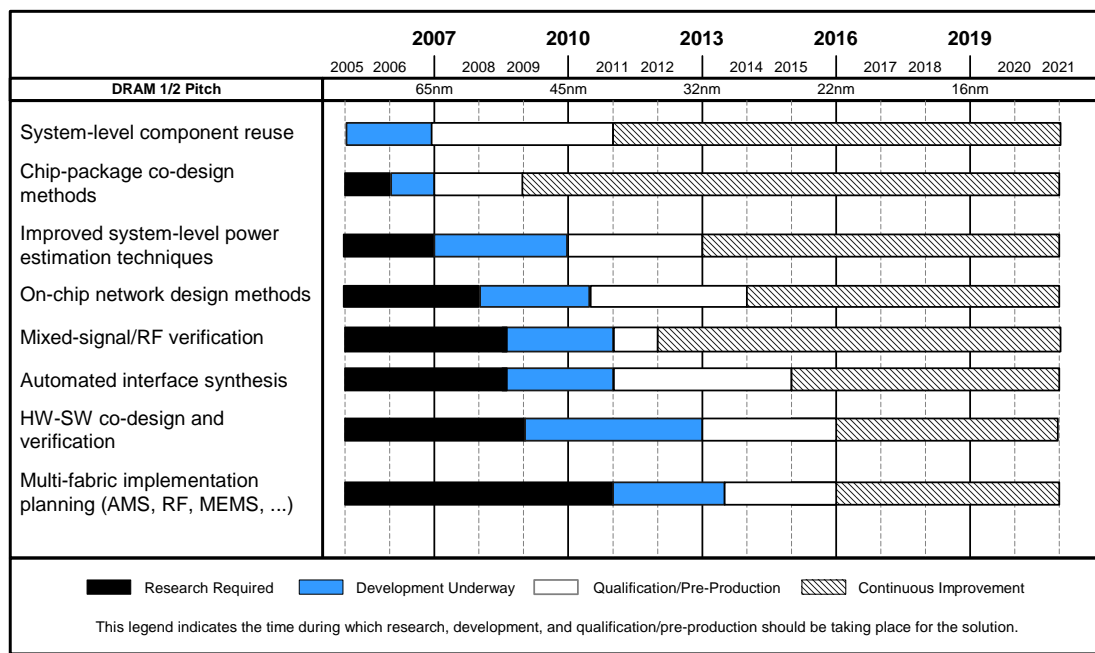
デジタルとアナログ設計はシステムレベルでは等しく重要になるため、システム・レベル・デザインにおける生産性の向上を維持するためには、アナログにおける設計やモデリング方法論は、デジタルのそれらと同等の成熟度が必要となる。

[11] アナログ設計におけるメソドロジ、記述言語およびシミュレーション環境の成熟を、デジタル設計と対比した指標。

[12] 値は 2004 年に実施した調査に基づく。

次の Figure 20 は、対応する期間におけるシステム・レベル・デザインのための解決策を示す。





MEMS—micro-electron mechanical systems

Figure 20 System Level Design Potential Solutions

最後に、次の Table 14 は、システムレベルデザインにおいて、Table 13 に示した要求が Figure 20 に記載された解決策にどのように対応するかを示している。

Table 14 Correspondence between Requirements and Solutions

Design block reuse	System-level component reuse	The larger and more complex the components that can be reused, the larger the overall design reuse expected to be.
	On-chip network design methods	Standardized communication structures and interfaces support reuse, since IPs with standardized interfaces can be easily integrated and exchanged, and also the communication structure itself is reused.
Available platforms	Multi-fabric implementation planning (AMS, RF, MEMS, ...)	Enables to integrate different fabrics on the same die or in the same package (SIP), and, hence, should enable to reduce the absolute number of platforms, since what needed different platforms before could now be integrated on one.
Platforms supported	Automated interface synthesis	Automated interface synthesis is one building block to an integrated synthesis flow for whole platforms.
	Automated HW-SW co-design and verification	An obvious requirement for an integrated, platform-based system development approach
Accuracy of high level estimates	Improved system-level power estimation techniques	While area and performance estimation on higher levels has already been a topic for several years and made some progress, system-level power estimation is a rather innovative but important topic and has to catch up.
	Chip-package co-design methods	Allows taking effects of packaging, for instance impact on the timing, into account for estimations on higher levels.
SOC reconfigurability	On-chip network design methods	On-chip networks are flexible and reconfigurable communication structures.
Analog automation	Multi-fabric implementation planning (AMS, RF, MEMS, ...)	Multi-fabric implementation planning for AMS and RF components seems to be one building block to analog automation.
Modelling methodology, description languages, and simulation environments	Mixed-signal/RF verification	As for digital design, verification is of major importance and verification is increasingly the most important and most time-consuming activity in the design flow.

## 論理、回路、物理設計

IC インプリメンテーションの従来の見方では、論理設計が、システムレベル設計ハンドオフ(現在 RT レベルに)から物理設計への入力にふさわしい、ゲートレベルにマッピングするプロセスである。デバイスの回路設計工程、またプロセスかつ製造上の制約によって規定された電氣的・物理的な特性を達成する部品(スタンダード・セル、十分に設計されたアナログ回路など)を相互に接続させる。チップ・インプリメンテーション(フロアプランニング、配置、配線、検証)の物理設計工程は、デバイスの概略レイアウトに関係があり相互に連携する。物理設計の出力とは、正確さ(設計規則、レイアウト仕様など)および制約(タイミング、消費電力、信頼度など)の検証に加えて、製造するための(現在 GDSII ストリームファイル)にハンドオフ(テープアウト)することである。論理、回路および物理設計は、システムレベル設計を支援する DT のインプリメンテーションを含む。

設計生産性は信頼できて見積り可能なインプリメンテーションファブリックへシステムレベル・サインオフを要求する。しかしながら、シリコン複雑度は、最終的な設計品質(タイミング、消費電力、シグナル・インテグリティ、信頼度、製造容易性など)に対する物理学の影響を評価し抽象化することを困難にする。貧弱な評価により過度にガードバンドすることを回避するために、論理設計、また、システムレベル設計は、物理設計とより緊密にリ

ンクされるようになるに違いない。したがって、しっかりと統合されてインクリメンタルスタチックな(消費電力、タイミング、ノイズ)分析「backplane」に支援された階層的で統制が行き届いた、レイアウトに基づいたインプリメンテーション立案の最近のパラダイムは固執する。将来の成功したインプリメンテーション DT は、複雑度を爆発させてエラートラント、ばらつきおよびコストのような関係に出現することに直面してプロセス・デバイス抽象、設計条件の扱い、分析および最適化を欺くメソッド選択に極度に依存する。

現在のハードウェア設計自動化の実際は、課題とより良い製造プロセスへの機会への対処を発展させなければならない。複雑な設計フローをルーズな連携による管理可能なステップのシーケンスに直列化するような、配慮の分解の法則に約束されたメソッドは、ますます時代遅れになるだろう。このシナリオでは、デザインフローの早期工程で行われた決断は、以降のステージの制約条件を拘束する。このような直列化は、全ての設計側面を同時に考慮するメソッドよりも、最適化されない設計を生成しかねない。しかしながら、このような状況は、全ての設計パラメータを同時に最適化することの実際の困難さにより避け難いものとなり、制約がフィードフォワードで満たされる限り、受容できると考えられている。しかし、これらの制約が満足できなくなると、このメソッドはワークしなくなる。このようなケースでの典型的な対応は、前工程にもどり、問題の原因と疑われる決断を変更するイタレーションである。このようなイタレーションは、特に、論理設計と物理設計の間で必要になっている。それは、レイアウト合成が、タイミング制約すなわちタイミングクロージャーの実現に無能だからである。理想的には、今日のメソッドにおける、時間を浪費する、論理と物理合成間のイタレーションは、回路のレイアウトと同様に論理構造を同時に最適化することにより、これらのステージを融合させることで解決できるかもしれない。

論理、回路、物理設計の課題に対する要求を Table15 に示す。強調されているのは、使われる設計スタイルと、設計パラメータにおける、技術発展の効果に関するものである。このセクションでは更に、Figure21 に示された課題解決策についても述べる。課題解決策は、述べられた要求にアドレスするツールの能力にフォーカスしている。これらの要求と課題解決策への説明を以下に述べる。

## 論理、回路、物理設計における設計要求

1. **非同期グローバルシグナリング** 近代の IC 設計の重要な課題の一つは、可能な限り少ないスキューで集結したクロック信号をチップ全体に配線することである。同様に手ごわいタスクは数個のプレ設計モジュールをチップ上で結合し、各モジュール間に信頼できるコミュニケーションを提供することである。GALS(globally asynchronous locally synchronous)エレクトロニクス設計はこれらの問題にアドレスするメソッドである。GALS ではやや大雑把な機能モジュールが従来の設計技術を用いて設計される。各モジュールはそれぞれのローカル・クロック・ジェネレータと、各モジュールが非同期ハンドシェイク・プロトコルを使って通信することを可能とする self-timed wrapper によって補完される。ハンドシェイク・コンポーネントの数が、この方法で動作するシステムの複雑度を決定する。独立したコンポーネントの数は、例えばクロック分配のような関連オーバーヘッドにより、劇的に増大するとは予測されない。
2. **不確定なパラメータ** EDA ツールは ASIC(application-specific integrated circuit)ベンダが設計チームにデザインを戻さなければいけない回数を減らすように、論理と物理ドメイン間を強くリンクすることによって DSM(data storage management)内部遅延に起因する不確定性を見積もろうと試みている。これが内部ディレイのためにワークしている限り、トランジスタ中の大量の電氣的な不確定性は説明できないままである。結果として静的タイミング解析ツールが必要以上に回路中の動作遅延を見積り、デザインの動作周期への深刻な悲観主義を引き起こしている。Table15 のリストはプロセスばらつきに起因する parametric uncertainty の効果を予測している。増大する不確定性は、論理と回路設計にプロセスばらつきを正確に考慮することが必要であることを強調している。ばらつきへの対応は高性能なデジタルIC設計の側面で重要性が増し、かつ first-time-right ハードウェアとカッティングエッジパフォーマンスに欠かせなくなっている。

3. **同時分析オブジェクト** 現在の VLSI ツールの殆どは、正確な RC 抽出やトランスミッションライン効果やカップリング効果といったテーマを同時に扱うことはできない。ましてや配線性能、面積そしてパワー損失のための伝統的なモデルは、お互いを個別に考慮していた。到来する DSM テクノロジはフローに統合するために多くの対象物を必要とし、それぞれの影響は同時に考慮される。
4. **故障許容製造** 現在のベンダのマーケットシェアとその製品の品質は直接的な相関がある。テクノロジーの出現で、MFFT(manufacturability for fault tolerance)をもつことがますます重要性を増す。表では MFFT 要望は信頼性の要素の観点から測定される。デバイスの信頼性はデザイン温度を低くするというような設計の決定に起因する。表で使用されている測定基準は最先端技術歩留りパラメータに対して正規化した信頼性ファクターを基にしている。
5. **単体設計の回路ファミリー数** 高性能 IC は、より高速で、可能なエリア使用、消費電力、設計効果などを得るために、しばしば回路ファミリーを混在させる。一つのデザイン上に異なる回路ファミリーを扱えることは重要な要望でありつづける。しかしながら、論理ファミリーをミックスすることにより増加する、関連する設計の複雑度や製造の困難さにより、これからの 10 年間はワンチップ上の異なるファミリー数は相対的に安定すると見込まれる。
6. **アナログコンテンツ統合** 表は統合されたアナログ回路量を 1 デザインのパーセンテージという形でまとめている。詳細は[システムドライバ章](#)の AMS セクションを参照のこと。
7. **予測されるプラットフォーム上での設計**
8. **適応型自己修復回路** 使用中のほぼ全ての回路は何らかのダメージで傷つけられると機能しなくなる。製造プロセスばらつきの増加によって製造チップの歩留りは劇的に低下した。多くのケースでシステムサイズと複雑度のトレードオフにより、コンパクトで頑強でない電子部品を実装し、壊れたら置き換える方が安価だということが続いている。更に機能しなくなっている原因を遠隔で探り、修理するためにシステムとコミュニケーションすること高価過ぎてできないケースと同様に、コストが掛かり過ぎてシステムにアクセスできないケースもある。この理由により、これまでになく大きな面積を消費するかもしれないが内部故障に高く適応する回路への要望が増大することを ITRS は予想する。
9. **パワーリーケージ** 消費電力は半導体産業が現在直面している主要な技術的問題である。素子サイズが 0.1 ミクロンより小さくなるにしたがって、待機時電力が低消費設計に新たな課題を引き起こしている。ITRS は時間がたつにつれ、デバイスあたりの動作時電力は減っていくと予測する。しかしながら、2 年ごとにチップ上のデバイスが倍増することがチップ基盤ごとのリーク電流の増加を引き起こすだろう。



Table 15a Logic/Circuit/Physical Design Technology Requirements—Near-term Years

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32
Asynchronous global signaling % of a design driven by handshake clocking	5%	5%	7%	11%	15%	17%	19%	20%	22%
Parameter uncertainty %-effect (on sign-off delay)	5%	6%	6%	8%	10%	11%	11%	12%	14%
Simultaneous analysis objectives # of objectives during optimization	4	4	4	5	6	6	6	6	7
MTTF contribution reliability factor	1	1.1	1.2	1.3	1.4	1.6	1.7	1.8	1.9
Circuit families # of circuit families in a single design	2	2	3	3	4	4	4	4	4
Analog content synthesized % of a design	10%	13%	15%	16%	17%	18%	19%	20%	23%
Leakage # times per device	2	3	4	6	8	9.5	11	12	14

Table 15b Logic/Circuit/Physical Design Technology Requirements—Long-term Years

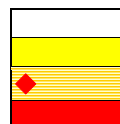
Year of Production	2014	2015	2016	2017	2018	2019	2020
DRAM ½ Pitch (nm)(contacted)	28	25	22	20	18	16	14
Asynchronous global signaling % of a design driven by handshake clocking	20%	25%	30%	30%	30%	35%	40%
Parameter uncertainty %-effect (on sign-off delay)	15%	18%	20%	20%	20%	22%	25%
Simultaneous analysis objectives # of objectives during optimization	8	8	8	8	8	8	8
MTTF contribution reliability factor	2	2.1	2.2	2.3	2.5	2.6	2.7
Circuit families # of circuit families in a single design	4	4	4	4	4	4	4
Analog content synthesized % of a design	25%	28%	30%	35%	40%	45%	50%
Leakage # times per device	16	24	32	32	32	32	32

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known



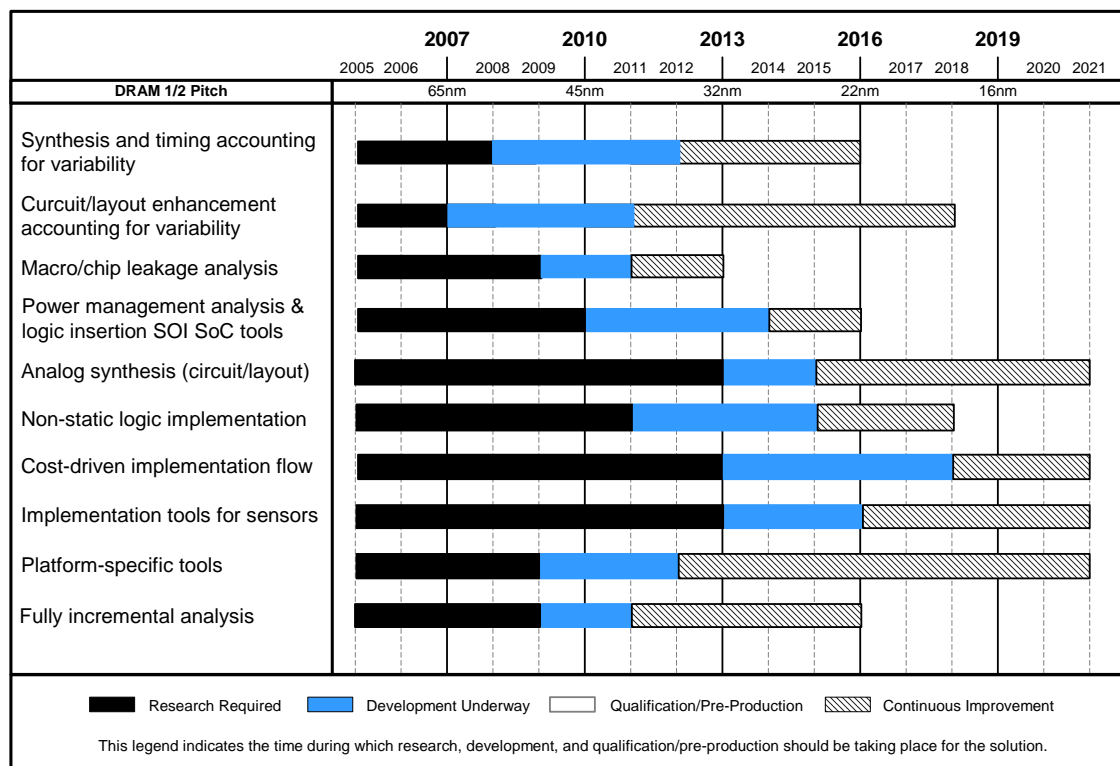


Figure 21 Logical/Circuit/Physical Design Potential Solutions

## 設計検証

設計検証は、与えられた設計が、意図した動作を正確に実行することを確認するタスクである。今日、最新コンピューティングシステムの検証は、潜在的なバグがあるままリリースされることで、ほとんど限定的な成功しかしていない電子システム設計における、コストを支配するようになってきている。実際に、多くの応用分野において、検証作業は、プロジェクトの時間、コスト、工数に関して、支配的な要素となっている。現在の設計プロジェクトにおいては、検証エンジニア数が設計者数を上回っており、最も複雑な設計では、その比率は 2~3 倍に達している。設計構想とインプリメンテーションは、もはや検証という主たる作業の前処理にすぎなくなっている。

この不幸な状況は、以下 2 つの進化に起因している。1 つめは、最近の設計における機能の複雑度が、驚異的な速度で増加していることにある。設計規模は、ムーアの法則により指数関数的に増大している。最悪の場合、検証すべきシステムの独立した構成状態の数によって表すことができる機能複雑度は、ダブル指数関数的に増加する可能性がある。<sup>6</sup> 2 つめは、歴史的に設計工程に注力したことで、論理合成や配置配線、テストパターン生成などの自動化ツールでは大きな進展があったが、検証がボトルネックとして残ってしまったことである。大きなブレークスルーがなければ、検証が半導体産業の今後の進歩に対して、ひときわ目立つ計り知れない障害となるであろう。

<sup>6</sup>この結論に至るには、多くの屈曲がある。例えば、1 チップ上のトランジスタ数が倍増する新しいデザインは、チップ上のラッチの数もおおよそ倍増する。これは、そのデザインの到達可能な状態数がおおよそ二乗になることを意味する。この分析は、正しい動作がシステムの到達可能な状態数の組み合わせを調べるか、もしくは同様な計算方法で検証されることを仮定している。もしも正しい動作の検証が状態シーケンスを超える解析を要求すると、計算処理による解決策はいっそう難しくなる。

これらのブレイクスルーが起こるための全体的な動向は、アドホックな検証手法から、構造化された形式的検証手法への移行にある。電子システム設計の機能検証を行うために、今日、産業界で主流となっている手法は、繰り返しモデルを作成しては、アドホックに選んだ検証ベクタでシミュレーションを行い、たまたま発見されたバグを修正することである。このため、論理シミュレーション技術は産業界において支配的である。なぜなら、それらはシミュレーション・ベクタを非常に効率よく生成できるからである。しかしながら、生成されたテストのカバレッジ(Coverage)は、通常非常に低く、数ヶ月におよぶシミュレーションでも、設計の正しさを十分確認できないという結果や、設計エラーが発見された時のエラー解析フェーズが、非常に時間がかかる複雑なバグトレースにひるんでしまうことになる。更に、これらの従来手法は、重要な品質の特定の設計領域へ検証活動向けるために、エンジニアリングチームに多大な努力を必要としたり、カバレッジが低かったりする。このような状況にもかかわらず、これらの手法は現状の設計プロセスにおいて、非常に大きな慣性を持っているので、代替の設計手法に移行するためのコストは非常に高い。このため、最近使えるようになってきたフォーマルやセミフォーマル技術の利用は、産業界全体としては、まだ一部に限られている。

より構造化された検証への取り組みは、一般に“ゴールデンモデル(Golden Model)<sup>7</sup>”と呼ばれるシステムの動作を表すモデルを作成して、このモデルと設計回路のシミュレーション結果を比較することによって検証を行うことである。更に、検証対象の正しさに対する確信のレベルを評価するために、カバレッジ指標を集める。検証作業を階層的に進めることは一般的で、最初に、チップのコンポーネントについて調べて、次にチップ全体を調べ、更にシステム全体<sup>8</sup>を検証する。これによって、あるひとつのユニットにあるバグを、早期にかつ容易に見つけて、解決することができる。ようやく、フォーマルやセミフォーマル検証手法への興味が高まりつつあり、検証チームは市販ツールの試行を始めて、これらのいくつかの技術で、彼らの主流の検証手法を補うような方法を探し始めている。フォーマル手法は、検証の複数の場面で利用できるもので、検証工程を定式化することから、仕様と検証に対して形式的表記法を適用することや、フォーマル・ソルバやプルーフ技術を使うところまで幅広く適用できる。

複雑なシステム、特にシステムオンチップに対応するためには、それぞれの階層レベルに対応する特定の手法とツールの開発とともに、階層的検証手法が必要であると思われる。これに関連して、設計と実装(インプリメンテーション)のピラミッドは、おおよそ以下のようにまとめることができる。

- 1) **モジュールレベル**:(構造的で、データフローと動作モデルの記述と合成) ここでは、検証ツールは、10K から 500K ロジックゲートのシステムの正しさを検証しなければならない
- 2) **サブシステムレベル**:(ほとんどの部分は、モジュールを構造的に組み立てた結果である) 100K から 10M ロジックゲートのシステムを構成
- 3) **システム/フルチップレベル**:(サブシステムを組み上げることで得られる) 1M から 100M ロジックゲート以上の設計

階層的検証の方式においては、それぞれのレベルごとに異なる検証技術が使われるであろう。モジュールレベルのゴールは、手作りのテストによるシミュレーションと、プロパティとアサーションのフォーマル検証で、高品質なモジュールを作り出すことである。サブシステムレベルでは、検証は、インタフェースのプロパティ検証と制約付きランダムパターン生成によるシミュレーションの組み合わせが必要になる。サブシステムレベルのプロパティ検証は、将来実現できるであろう。例えば、アグレッシブな自動設計回路抽象化技術と

<sup>7</sup> ゴールデンモデルは、一般にSystemCやANSICなどの高位言語で記述される。

<sup>8</sup> システム全体のシミュレーションは、その複雑さのために、ハードウェアエミュレーションがソフトシミュレーションの代わりに利用されることがある。特に大きなマーケットがある場合には、追加コストを吸収することができる。ハードウェアエミュレーションは数 10 倍以上のシミュレーション性能改善があり、検証に換え難い助けとなっている。またシステムインテグレーションやソフトウェア開発を容易にすることができる。しかしながら、エミュレーションシステムは一度に一つのシミュレーションしかできないので、一定倍の改善しか見込めないし、検証問題に対して、スケーラブルで長期的なソリューションにはならない。

抽象モデルの自動詳細化を定義することを通して、検証エンジニアリング工数は最小で、サブシステムの特性を検証することが可能となる。システムレベルでは、カバレッジ測定法とシミュレーション・ベースのアサーションチェックが、検証技術のなかで最も適していると思われる。技術的な進展は、高品質な検証結果を実現する、正確で効率的な手法の開発にかかっている。これらは、適切なカバレッジ測定法、必要な工数に対して非常に効率の良い検証ツール、バグが検出された時に回路デバッグの負荷を軽減する技術の開発によって実現される。

以下に示す Table16 は、次世代のテクノロジー世代における設計複雑度のトレンドをサポートするのに必要な、設計検証への定量的な重要要求項目を示している。この要求項目は、設計の正しさを保証することと、開発プロジェクトで費やされる検証工数の両方について見積もっている。

Table 16a Design Verification Requirements—Near-term

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32
<i>Productivity</i>									
SOC logic size verified Mtx* per engineer-year (based on a ten-engineer team)	4.7	6.1	8.0	10.5	13.8	18.1	23.8	31.2	41.1
<i>Methodology</i>									
% bugs found by formal and semi-formal verification technology (versus simulation)	0.0	2.7	5.3	8.0	10.7	13.3	16.0	18.7	21.3
% verification effort spent on verification of integrated SW / HW / electrical effects	0.0	0.0	0.1	0.1	0.1	0.1	0.2	0.2	0.2
% design specification formalized for verifiability	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.4	0.4
<i>Reuse</i>									
% new verification development (versus IP and reuse)	0.8	0.8	0.7	0.7	0.7	0.6	0.6	0.6	0.5
% verification infrastructure acquired from third parties (verification IP)	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.4	0.4
<i>Functional coverage</i>									
% project verified through functional coverage	40.0	43.7	47.3	51.0	54.7	58.3	62.0	65.7	69.3
Coverage goal density (goals / Mtx* of logic)	667	1022	1378	1733	2089	2444	2800	3156	3511

\* Mtx = millions of transistors

Manufacturable solutions exist, and are being optimized  
 Manufacturable solutions are known  
 Interim solutions are known  
 Manufacturable solutions are NOT known

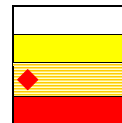


Table 16b Design Verification Requirements—Long-term

Year of Production	2014	2015	2016	2017	2018	2019	2020
DRAM ½ Pitch (nm) (contacted)	28	25	22	20	18	16	14
<i>Productivity</i>							
SOC logic size verified Mtx* per engineer-year (based on a ten-engineer team)	54.1	71.3	94.0	124	163	216	286
<i>Methodology</i>							
% bugs found by formal and semi-formal verification technology (versus simulation)	24.0	26.7	29.3	32.0	34.7	37.3	40.0
% verification effort spent on verification of integrated SW / HW / electrical effects	0.2	0.3	0.3	0.3	0.3	0.4	0.4
% design specification formalized for verifiability	0.5	0.5	0.5	0.6	0.6	0.7	0.7
<i>Reuse</i>							
% new verification development (versus IP and reuse)	0.5	0.5	0.4	0.4	0.4	0.3	0.3
% verification infrastructure acquired from third parties (verification IP)	0.5	0.5	0.5	0.6	0.6	0.7	0.7
<i>Functional coverage</i>							
% project verified through functional coverage	73.0	76.7	80.3	84.0	87.7	91.3	95.0
Coverage goal density (goals / Mtx* of logic)	3867	4222	4578	4933	5289	5644	6000

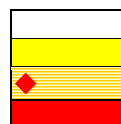
\* Mtx = millions of transistors

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known



表中の数値は、新規設計プロジェクトを想定しているもので、派生設計とは異なる。同様な設計開発期間を維持するためには、設計複雑度の増大に対して、検証効率が比例する関係でなければならない。処理時間で、検証ツールをフォーマル検証とシミュレーションの間で分類すると、特に長期展望では、両者の組み合わせとセミフォーマル検証に変わっていく。ここでの予測は、これらの組み合わせの解決策に基づいており、シミュレーション・ベース手法とフォーマル検証技術の比率とほぼ等しいものである。この設計生産性向上を実現するには、シミュレーション・ベース手法を補うために、更にフォーマルとセミフォーマルエンジンの統合が進むことが必須である。更に、ソフトウェアやミックスドシグナル・コンポーネントの影響の増大と微細化の進展による電氣的影響が重要になってくることから、ヘテロジニアス・コンポーネント間及び、システム全体としての検証は、検証工数全体に占める比率が増大していく。最後に、検証要求に対して更に構造的な手法に移行することで、設計仕様をフォーマルに表現する方向に移行する努力が、検証全体の工程を自動化する方向に向かわせることになる。

検証整備は、次の 3 つの場合に分けられる。1) 検証対象の設計に対して新規に記述する場合、2) サードパーティから入手する場合(検証 IP として知られている)、3) 社内で以前に設計されたものから流用する場合。表では、最初の 2 つの場合を再利用(Reuse)のところで示しており、3 つ目の場合は、これらの値の差として計算できる。表の最後の 2 行は、機能力バレージ(Functional Coverage)の重要性が大きくなることを予測したものである。従来のカバレージは、コードカバレージとステートカバレージが対象であったが、機能力バレージは検証する設計回路機能をより直接的に捕えるものである。しかし、その機能動作に依存した品質となるため、より多くの工数を必要とする。短期的及び長期的に検証の進展を定量化するためには、機能力バレージの採用が増えることが必要である。エンジニアリングチームが、検証環境構築に習熟して、機能力バレージの再利用が増えてきても、この表は将来においても、技術の利用が広がることが必要だということを示している。



## 設計検証の課題

### 短期

検証に対する最も重要な課題の多くは、ほとんどすべてシステムドライバに関連している。短期的には、最も重要な問題は、フォーマルおよびセミフォーマル検証手法がもっと信頼できて、制御できるようにすることを軸としている。特に、検証品質の重要な測定基準と同様、フォーマル検証ツールの処理容量と信頼性に大きな進展があることである。長期的な問題として、抽象レベルを上げること、フォーマル検証の扱える範囲を広げること、重点的に取り組む。長期的な問題は実際に現在に関連するものであるが、短期的な課題は、既に危機的状況にある。一般に、すべての検証課題はSOCにあてはまる。MPUは、その最先端の複雑さと、飛び抜けて大量に生産される途方もなく複雑な設計の、独特な経済性のために、異なった検証課題を提起している。結果として、長期と短期の両方に、異なった分野毎の検証課題と解決策が存在する。

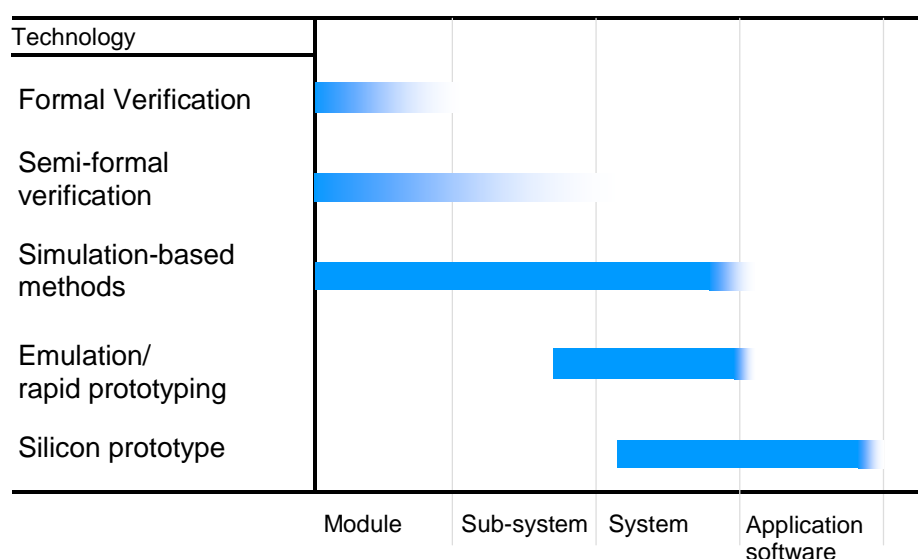


Figure 22 Current Verification Tool Landscape

**能力 (キャパシティ)** Figure 22 は、現在利用可能な検証ツールと設計粒度の関係を示している。横軸は、モジュール、サブモジュール、複数のコンポーネントを接続したある高機能を持つシステム、一つのチップ上に複数の機能を統合したシステムオンチップ、システム上で実行するソフトウェアへと設計複雑度を示している。縦軸は、利用可能な検証技術を示している。

従来のシミュレーション・ベース手法<sup>9</sup>は、広範囲にわたって、モジュールレベルからフルシステムレベルの検証まで、現実的に何にでも使用される。しかしながら、その検証品質は大きく変わる。小さなモジュールに対しては、短期間に多くのテストベクタによるシミュレーションを実行できるので、検証対象の回路のかなり良いカバレッジが得られる。一方、システムレベルでは、設計規模とシミュレーション・ベクタの数に比例して、その中

<sup>9</sup>ここで、“Conventional Simulation(従来のシミュレーション)”とは、一度に一つのシステム動作をシミュレーションすることによる検証技術のことである。“Formal Verification (フォーマル検証)”とは、シンボリック・シミュレーションやシンボリック・トラジェクトリー評価、モデルチェックや、システムの起こり得る動作の網羅的な分析に効果のある理論証明などの、あらゆる技術のことである。場合によって有効な区別は、多大な人間の経験を必要とするコストはかかるが、表現が豊かな解析的能力を提供する論理的証明手法と、さらなる自動化のためのトレードオフの論理的能力につながる別の手法との間に線が引ける。“Semi-formal (セミフォーマル検証)”とは、通常、フォーマル検証とカバレッジ・シミュレーションの技術を混在させることにより、フォーマルカバレッジの完全性を犠牲にすることで、大規模設計への対応を試みる広範囲の技術のことである。

に含まれるアルゴリズムが複雑になるため、シミュレーションスピードは非常に遅くなる。このため、あらゆる複雑なシステムに対して、現実的な時間内では、わずかなシミュレーション・ベクタしか実行できないので、この結果得られるカバレッジは、シミュレーションしたシステムの複雑さを考えると、非常に小さなものになる。フォーマル検証では、システムのいろいろな特性に合わせて広範囲なプロパティが設定されることで、非常に高いカバレッジを得られる。この手法の弱点は、通常モジュールレベルより大きな回路には適用できないというスケラビリティの制約である。セミフォーマル検証は、フォーマル手法とシミュレーション・ベースを混在させようとしている。Figure 22 に示すように、セミフォーマル手法は一般的にフォーマルより大きな回路に適用可能であるが、場合によってはカバレッジが低くなる。ソフトウェア開発が含まれる非常に複雑なシステムでは、ソフトウェアツールは、システムの正しさを確認するには不十分である。このレベルでの解決策には、プログラム可能ゲートアレイ (FPGA) によるラピッド・プロトタイピングや、ハードウェアのシリコン実装がある。課題は、複雑さのすべてのレベルにおいて、高いカバレッジを得られる新しい解決策を作ることである。フォーマル検証とシミュレーションの二つの直行する手法は両方とも重大な弱点を持つ。すなわちフォーマル・ツールは中小規模の回路しか扱うことができないし、シミュレーション・ベースのツールは、ほとんど任意の複雑度の回路をシミュレーションできるが、非常に長時間のシミュレーションを実行しても、無視できるくらいわずかなカバレッジしか得られない。エミュレーションとラピッド・プロトタイピングはソフトウェア・シミュレーションに比べて数 10 倍以上の性能が出るので、高いカバレッジを得ることができる。しかしながら、その改善度は一定であり設計複雑度の大きさに比例するものではない。

**ロバスト性** - Figure 22 には示されていない現在の、検証の解決策の重要な点は、それらの信頼性である。一方では、シミュレーション・ベース手法は、シミュレーション・ベクタ当たりの実行時間は設計複雑度に比例するので、かなり予測性がよい。したがって、設計が最も複雑なデザイン・レベルになるのにしたがって、シミュレーションの性能は直線的に減少する。エミュレーション技術も同じような傾向を持つが、シリコン・プロトタイプは一貫して信頼できる。一方、フォーマル検証手法は、検証問題の複雑さに対応するために、きまぐれな経験則に大きく依存している。どのような設計回路と検証アルゴリズムの組み合わせに対しても、検証アルゴリズムが完了するかどうか判断するのに、専門家ですえ四苦八苦することがある。問題の大きさを測る共通の指標としてのトランジスタやゲート、ラッチの数は、フォーマル検証の複雑度との関係では曖昧なものである。容易に検証できる数千のラッチを含む回路と同じように、知られている検証手法で対応できない 100 未満のラッチを含む回路を容易に見つけることができる。このような予見不能性は、現実の設計環境では受け入れられない。重要な検証課題は、検証工程をもっと確実なものにすることである。確実性は、検証し易い設計に通じるものであれば、検証アルゴリズムの経験則を改善する形でも良いし、与えられた回路の検証困難度のキャラクタライゼーションを改善する形でも良い。

**検証の測定基準(メトリクス)** - 短期的に重要な検証課題として、検証の質を計る必要性があげられる。特に、カバレッジという大事な概念は必須である。カバレッジの測定基準には様々なタイプがあり、それぞれに長所と限界がある。

- コードカバレッジは、シミュレーションの間に設計ソースコードのどれだけがシミュレーションされたかの測定によるものである(例えばコードの行数や数式のカバレッジ)。ラインカバレッジの一種にコード内に書かれた数式の全ての項や Switch 文の全ての条件のカバレッジを測定するものもある。ラインカバレッジの限界は、動的シミュレーション実行を静的なソースコードに対応付けていることにある。つまりシミュレーション実行時には、様々な実行可能なパスがあるために、存在する多くの潜在的な問題を見過ごすことになる。
- 例えば、状態カバレッジや遷移パス・カバレッジなどの構造カバレッジは、設計の有限要素マシンに着目しており、どの状態がシミュレーションで確認されたかを測定する。しかし一般的に設計全体を 1 つの有限要素マシンとして扱うのは不可能であり、カバレッジの際にいくつかの有限要素マシンが対象となるだ

けである。欠点はこのカバレッジは 1 つのマシンに対しては良い結果を与えてくれるが、複数のマシンの組み合わせにより発生する状態の組み合わせが考慮されないことにある。

- 機能カバレッジは設計の各機能を対象にしている。機能は設計ごとに異なるため、明確な定義が設計の仕様に基づいた検証チームによってなされる必要がある。このように結果の質はカバレッジの定義の質にも依存する。

コードと構造カバレッジの測定基準は必ずしも正しいことを示しているわけでないが、設計のある品質を定量化するものである。機能カバレッジは、検証された設計の機能がどの程度測定されたかを計る目的を達している可能性がある。しかし課題は、良い機能カバレッジの定義にある。現状では、統一された環境下で機能カバレッジのための共通測定技術がなく、その必要性がある。設計で、ある一定レベルのカバレッジを得ることは、その設計で使われた特定の機能カバレッジ以上に、価値がある。加えて、機能カバレッジの測定単位を決める汎用的な方式がない。すなわち、開発中にガイドラインとして使える抽象的な基本モデルが求められており、それが開発をサポートする。最終的には検証のためのカバレッジと統一された検証単位の開発をサポートする、機能バグを示す欠陥モデルが必要であり、これはテストにおける欠陥モデルに非常に類似している。

**ソフトウェア** - SOC のような複雑なシステムの検証は、ハードウェア構成要素やハードウェア - ソフトウェア間のインタフェース、システム上で動作しているアプリケーション・ソフトウェアの検証が必須となる。SOC のソフトウェア構成要素は OS 上で動作するアプリケーション・ソフトウェア、もしくは、一般的にハードウェアから独立したレイヤとドライバなどのような低レベルでハードウェアに依存するレイヤに分類される。このようなシステムではソフトウェアレイヤは機能の多くを提供しており、SOC 検証の大きな課題は、ソフトウェアとハードウェア - ソフトウェア間のインタフェースをいかに検証するかである。現在、ソフトウェア開発はデザインレビュー、解析ツール、テスト方法の点で、ハードウェア設計ほど厳格でない。ソフトウェアは本質的に検証が難しく、より複雑で動的なデータと巨大な状態空間を有している。今日使われている検証には「オンチップ検証」があり、ソフトウェア検証のためにハードウェアの製品版の上でソフトウェアを走らせる手法である。この手法はソフトウェアが複雑なために必要とされるものであり、大変高速にシミュレーションができる。欠点は、設計過程の最終段階にならないとソフトウェア検証ができないことにある。ソフトウェア検証の古典的なフォーマルな手法は、SOC に広く適用するには大変な人手を必要とする。それは巨大な状態空間を持つシステムに対して、かなり大幅に抽象化する技術が必要とされるからである。ハードウェア - ソフトウェア間のインタフェース検証はそれ自身が課題である。それは 2 つの領域を同時に検証する必要があり非常に複雑な作業である。この作業をコントロール可能なものにするには、インタフェース動作を適切に抽象化して、抽象化されたドライバレイヤでの正しさの検証と、抽象化されていないレベルでのドライバの正確な動作をアサーションチェックする検証技術が必要となる。短期的な課題は、基本的で低レベルのソフトウェア検証を可能とする技術を開発することである。長期的な課題は、ソフトウェアに適用する検証容易化設計を理解することと、ソフトウェアとハードウェア - ソフトウェア間のインタフェースの確実な検証手法を開発することにある。

**再利用** - あらかじめ設計された IP ブロックは、ごくわずかな時間で新規の複雑な SOC を組み上げることを可能にする。大きな課題は、既設計でかつ検証済みのブロックから組み上げられるシステムを、迅速に検証する検証手法を開発することである。重要な問題は、IP ブロックの抽象的な動作を、いかに厳密で完全に記述するか、IP ブロックが仮定している環境的な制約をいかに記述するか、また検証を単純化するためにいかに階層化を活用するかにある。標準プロトコル検証 IP から、汎用的な IP ブロックの抽象化されたモデル、ブロックの周辺環境からの制約をチェックするプロトコルチェッカ、一連処理の生成まで様々な検証 IP を一緒にして販売され始めた IP コンポーネントもある。しかしこれらはまだ初期的な試みであり、再利用のメソッドを発展させるには、いかなる IP ブロックにも適用できる検証 IP がある程度入手できる必要がある。IP ブロックに関連した検証コンポーネントに加えて、特定のプロトコル向け環境生成プログラムのような、独立した検証 IP も必要である。短期的な進展は、オンチップバスのような標準化された IP の相互結合に対して起こりうる。しかしながら時々問題となる IP ブロック・インタフェースの一般的な問題も最終的には解決されなくてはならない。

**特殊な検証手法** - 設計検証での最大の課題は、現在知られているアルゴリズムによる解法では、今日開発中の設計に対応するには容量が不足することにある。短期的にこの課題を克服する唯一の可能性のある方法は、適切な検証メソッドを用いることにある。これに向けた現状の傾向は、(1)シミュレーション・ベース検証と最近利用が増えてきているセミフォーマル検証の両方に対しての、カバレッジベースの検証、(2)フォーマル記述による仕様のドキュメント、これにより実行時に検証されるフォーマルプロパティの集合が簡単に得られるようになる、(3)カバレッジモデルのテンプレート、があげられる。しかし実用上、確実に完全なメソッドを得るにはまだまだ多くの課題がある。例えば、抽象化の際に設計の重要な特性を落とさないで、設計部品やインタフェースなどに対して整合性の取れた抽象化を行う技術を得るための方法が必要である。重要な特性は、抽象化を行う時の設計の使われ方に依存する。フォーマル仕様記述が開発されているが、大きな課題はこの記述の完全性にある。同一システム内の異なる IP 部品が、完全に独立した設計チームによって開発されている現状では、この課題は避けられないものになっている。新しい検証手法が徐々に受け入れられているが、設計のフォーマルプロパティを作成するには、設計のある特定の使われ方を全体として理解しておく必要があるという課題があり、開発チームに更に工数を必要とする。最後に、検証メソッドは、適切に対処法を用意したとしても、完全さを保障するものではないが、開発リスクを軽減するために発展中の目標である。

**特殊な検証容易化設計** - 検証をより効率的にするために、検証容易化設計に特化した活動が既に見受けられる。例えば、ソフトウェアやソフトウェア・ハードウェアのデバッグの仕組みをシリコンへ組み込むことなど。この方面では、自己検査機能付きプロセッサに関する取り組みが始まっており、小さなウォッチドッグプロセッサはメインプロセッサが正しく実行しているかを検証している。ミックスドシグナル設計では、設計回路のアナログ部をバイパスするループバックモードを挿入して、完全なデジタル回路としてシステムを検証可能にしている。特にソフトウェアレベルでは、シンクロナイザが検証用チェックポイントを設定して探索可能な状態空間を減らすことにより、タスクが独立して同期ポイントを越えて進まないようにしている。MPU 設計では、シンクロナイザにより推論実行システムの検証の複雑さを軽減している。この分野での課題は、いくつかの大きな領域での検証容易化設計の展開とその受け入れである。積み上げ式にシステムを逐次詳細化していく設計手法の開発が行なわれているが、詳細化行程がどの程度自動化できるのかが問題であり、一方、人手の介入は設計エラーの発生源となる可能性がある。

**新しい種類の同時処理** - MPU 設計はより複雑になってきており、新しい種類の同時処理が重要になってきている。既に検証の目を逃れた多くのバグは、キャッシュのコヒーレンシとその他の同時処理による問題に関係している。新しい設計は、チップレベルのマルチプロセッシングとオンチップ・キャッシュコヒーレンス・プロトコル、および同時マルチスレディングのような技術に対して、同時処理数を増加させることによって、検証のプロセスを非常に複雑になっている。将来、内部プロセッサレベルとマルチプロセッサシステムの両方または他のハードウェアとの関連で、新しい種類の同時処理が検証をより難しくさせるであろう。この新しいレベルの複雑度を理解するためには、新しい故障モデルが必要である。その解法として、相互作用の複雑さを軽減してコンカレント・プロトコルを検証可能なものとするために、おそらくハードウェアとソフトウェア技術の混在が必要となるであろう。

## 長期

長期的には、問題は、検証生産性の改良、非常に複雑な設計の検証方法、異種混在システムの検証の必要性が中心になるであろう。最も重要な課題を下記に示す。

**検証容易化設計** - 短期的な技術課題への解決策は、検証するのに何が容易で何が難しいか、どのようにして設計エラーが起こるのかを理解することであり、長期的な技術課題への解決策は、どのようにしてこれらの理解を体系化して、検証容易な設計に変えるかということである。2007 年以降の設計を考えると、検証容易化設計なしでは検証が困難なものになりそうである。設計手法に対する大きな変更が求められ、いくつかのパフォ

ーマンスの低下が発生するであろう。分かりやすいたとえにシーケンシャルテスト容易性があり、計算処理不能なシーケンシャル ATPG は、ほぼ全面採用されているスキャンベーステストになっている。

**高い抽象レベル** - 設計の抽象度が RTL より上に移っていくことに対して、検証も追従しなければならない。技術課題は、高い抽象度での検証方式を開発し採用することで、高位設計により可能となったシステムの複雑度に対処し、高位レベルと低レベルのモデルの等価性を確認する手段を開発することである。この長期的な技術課題は、高位抽象化が検証(すなわち、言語が適切に定義されなかったり、不要に複雑なセマンティックスであったり、RTL モデルとのフォーマルな関係のないシミュレーション向けのモデルに依存した手法であったりした場合)のことを考えずに決められると、もっと難しくなる。

**検証容易化のための仕様表現** - 設計検証に対する継続的な課題は、設計に要求される動作をどのように表現するかである。現在可能な仕様記述は、汎用的な方法でこの問題を解くには力不足である。今まで以上に複雑な設計を表現するために使われる言語の開発には、何が仕様を明確にし、何が不明瞭にするのか、何が変更可能で何が手におえないのかということに対する、深い理解が必要である。例えば、仕様ドキュメントの首尾一貫性を自動的にチェックする方法が必要であり、それにより異なる仕様が競合する要求を出さなくなる。更に設計者がこれらの仕様表記法を使って、常にフォーマルな仕様を作成できるように、トレーニングが必要である。

**デジタル以外の影響の検証** - 現在のところ、設計検証は主としてデジタルシステムの個々の動作に焦点を当てている。シリコンの微細化とシステムの複雑化の両方への挑戦は、将来の検証技術が広範な種類の解析をすることを要求する。シリコンに実装した回路システムの複雑さがVLSI システムの明解なデジタル抽象化をより不確かなものにしていく。電気のアナログ的な振る舞いは、パフォーマンスと、時として機能動作に影響する。これらの影響を解析する既存のシミュレーション技術(SPICE)は遅すぎて、また微細化するデバイスがプロセス変動の影響を受けやすくなるにつれて信頼できなくなる。このため、マイクロプロセッサのアーキテクチャ・シミュレーションが行なわれ始めている。ハイレベル・アーキテクチャ・シミュレータはタイミングや電圧などに基づいて動く低レベルの詳細シミュレータとお互いに連動して、システム全体の評価には最小の性能影響でそのデータがフィードバックされる。長期的には、複合型システム<sup>10</sup>として扱われるデジタルとアナログの境界で、これらの問題を検証するためのフォーマル技術が必要となる。同様に、設計の最上位レベルでは、システムの複雑さにより、将来の検証タスクには(ネットワークプロセッサにおいてサービス品質を保証するように)アナログ的、確率的動作を表現し検証することが求められる。このように複合型システムと確率的検証が課題となろう。

**異種混在システム** - シリコンウェーハ上にデジタル設計に並べて配置される新しい技術開発が新しい課題を起こす。MEMS や電子光学デバイス、電子生物学的デバイスなどがその例である。これらの新しい構成部品のために、デジタル部分とデジタル以外の部分とのインタフェースと、デジタル以外のシステム動作の適切な抽象度の両方のモデル化が、システムのデジタル部の検証には必要となる。

**アナログ/ミックスドシグナル** - 今日のアナログシステムは、連続システムに対する古典的な解析ツールや、周波数領域に対するシステムモデリングと解析を通じて、主に検証される。検証の大半は設計の後処理として、アナログ実験装置にてテストチップ測定で行われる。ミックスドシグナル設計はデジタル部とアナログ部に分けて別々の検証作業が行なわれている。将来、ミックスドシグナル・システムがシリコン開発全体の中でより重要度を増すだろうから、ミックスドシグナルの適切な検証メソッドロジーの開発が必須となる。ここでの課題はシステムのアナログ部とデジタル部の検証を結合させることにある。目標達成のための要望事項の 1 つとしてデジタルシミュレーションとアナログシミュレーションの性能ギャップを埋めることである。

<sup>10</sup> 複合型システムは複雑な連続振る舞い(例えば、微分方程式モデル)と複雑な離散的振る舞い(例えば、有限状態マシン)の両方の振る舞いをする。この分野は古典的コントロール理論と離散的フォーマルベリフィケーションの両方から技術を借用している。



**ソフトウェア** - 長期的にソフトウェアに関するシステムの信頼性情報を提供できる検証技術の開発が必要である。

**冗長性の検証** - 冗長システムの信頼性の定量的な評価、信頼性があるレベルに達するために必要な冗長性のレベルの評価が、安全性重視のアプリケーションでは特に重要である。将来、システム性能と正当性の点で、電氣的影響がますます複雑化しているために、安全性重視以外の設計においても、これらの評価がより重要となっている。

## 設計検証 - 解決策

現在入手可能な、もしくは課題克服に向け開発中の解法が、Figure 23 に示されている。この図は開発チームによる解決策の入手可能時期に加えて、検証危機の克服のために重要な方向性をまとめている。

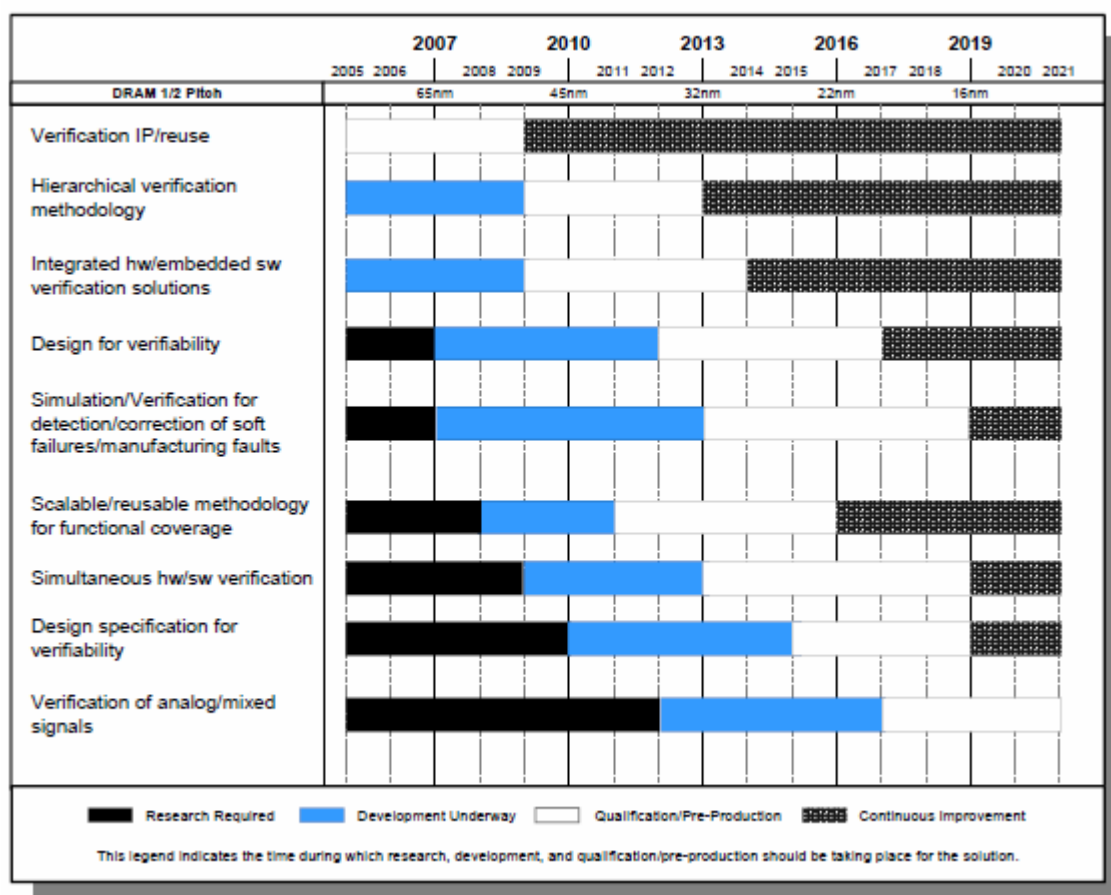


Figure 23 Design Verification Potential Solutions

検証 IP は、現在既に入手可能であり、検証行程の生産性を向上させるために進化しているメソッドのトレンドとなっている。将来、SOC 設計向けサードパーティの IP に積極的に採用されるに伴い、もっと広く普及するであろう。前の節で述べたように、階層化と統合ハードウェア - ソフトウェア検証手法により構築された解決策は、既に研究されており、今後2,3年のうちに入手可能となるであろう。この図で示したその他の解法は、現在、まだ概念段階にある。検証容易設計の開発、ソフトウェアや欠陥を訂正するフォーマルな手法は、前の節で述べたように、MPU 領域で初期の実験段階にある。構造/機能力バレージと仕様に関する技術が本流となるのはまだ先であるが、設計と検証の生産性のギャップを埋めることは重要である。

## テスト設計

ナノメータ・プロセス技術、ますます高くなるクロック周波数、アナログ・デジタル・RF 混在の回路、そして高密度に集積化された SOC、SIP は困難な課題を DFT (Design for Test) に提起している。テスト産業は、コンポーネント・ベース設計のための高位テスト合成から、非常に高性能なアナログや RF システムのためのノイズ、干渉、消費電力にいたる多様な問題に対応して行かねばならない。それらの多くの問題は、設計の初期の段階から最適なテストビリティが考慮され、DFT が組み込まれた場合に限って解決される。更に、メソドロジに上記のことをテスト検証の分析と設計と一体化した要求として取り込む必要がある。論理 - 回路 - 物理設計、そして設計検証における結果(設計フロー、ベクトル、故障モデル、感度)はテスト設計においても再利用される。メモリ、DSP、PE、SOC、アナログミックスドシグナル / RF、そして MPU といった様々な製品セグメントの種類をこえて高度なテストの課題、つまりオンチップ DFT、BIST、テストビリティの拡張、ならびにチップ開発の計画段階での、製造段階でのテスト装置の事前計画の前倒しを要請している。

DFT への技術要求とその解決策候補は、対象としているシステムの特徴によって大きく変化する。そして、それはすべてシステムレベルで統合されなければならない。それによって、SOC も SIP も安い製造コストでかつ性能をほとんど落とさずにテスト可能になる。この節では、システムドライバを共通のフレームワークとして用い、DFT への技術要求と解決策を記載する。

Table 17 に DFT への技術要求を、Figure 24 に DFT への解決策候補を 4 つの主要なシステムドライバに対して記載する。それらは、アナログ、ミックスドシグナル / RF、MPU / PE / DSP、メモリ、そして一般的な SOC / SIP の各システムドライバである。

**1. アナログ / ミックスドシグナル / RF システムドライバ** - アナログ、ミックスドシグナル、RF サブシステムは、コンポーネントの I/O スピードとともに、コア部のクロック周波数やトランジスタやアーキテクチャの性能などのシステム性能として重要になって来ている。今日、業界は 5GHz を超える RF の計測に関して深刻な問題に直面している。ウェーハレベル、またパッケージレベルで 5GHz を超えるテストは非常に高価になる。特別なテストと追加ハードウェアが必要であり、費用とテスト時間がかさむ。10GHz やそれ以上の設計がなされつつあるが、その周波数をテストし得る製品レベルでのテストは存在していない。この問題は将来にわたって更に深刻になると予測される。よって、10-30GHz の高周波数のミックスドシグナル回路を低価格のテストでフルにテストでき、そして複雑なテスト仕様をキャラクタライズできるような新しい手法を是非とも開発せねばならない。同時に、新しい I/O プロトコルを導入して、数 GHz レンジにまで拡張する必要がある。これらの I/O スキームは高速であるのみならず、また非常に複雑である。実際、ソースの同期、変動、同時双方向の動作が、Gbit/s のレートでしかも、Vdd の 1/10 のレベルで電圧が変動する。対照的に、ATE とコンポーネント・テストは MHz のレンジでの通常のクロックベースのテストと I/O 計測で構成されている。したがって、I/O 速度とプロトコルは、設計検証と製造段階でのテストの両方で、ATE 装置、インタフェースハードウェア、テストソケットなどの器具、材料、コストに多大の影響を及ぼす。この変化点は、特に I/O に対して、広範囲の業界の技術開発とチップ上のテスト可能性の応用を要求することになる。

これらのアナログ、ミックスドシグナル、RF サブシステムに対する DFT 技術は活発に研究され、また開発されているが、まだ成功していない。この DFT 技術は、設計工数を削減し、ノイズ除去と同じように頑強性を改善するために、可能な限り多くのデジタル回路を使用すべきである。これらの DFT 手法を使った結果は、それが Pass/Fail インディケータであっても、パラメトリックな測定であっても、信頼性と設計者の受け入れを獲得するために、標準的な仕様ベースのテスト手法との相互関連をとっておかねばならない。この相互関連付けは、今後アナログ、ミックスドシグナル、RF サブシステムの複雑さと性能が、増して、もはや外部の ATE や器具でテストできなくなるようになる以前にきちんとしておかねばならない。

Table 17a Design for Test Technology Requirements—Near-term Years

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32
<i>System Driver: Analog/Mixed-signal/RF</i>									
1. All-digital DFT for analog/mixed-signal/RF circuits and systems % digital circuits in DFT implementations	30	35	40	45	50	55	60	60	60
2. Correlation of DFT results with existing specification-based test methods. % results correlated	30	35	40	45	50	55	60	60	60
3. Availability of fault/defect models for DFT-oriented test methods. % AMS/RF blocks with accepted fault models	20	20	25	30	35	40	45	50	55
<i>System Drivers: MPU/PE/DSP/</i>									
1. DFT coverage of digital blocks or subsystems. % blocks with DFT	60	60	70	70	70	75	75	75	80
2. DFT for delay test of critical paths. % paths covered	50	50	55	55	60	60	60	60	70
3. DFT for fault tolerance in logic blocks. % blocks with fault tolerance	40	40	40	40	45	45	50	50	55
<i>System Drivers: Memories</i>									
1. DFT for yield improvement.	85	85	85	90	90	90	90	95	95
<i>General SOC/SIP requirements</i>									
1. DFT-support for logic and other circuit repair (except memory). % blocks with repair	50	50	50	60	60	60	70	70	70
2. DFT re-use for performance calibration, and measurement purposes. % DFT circuits re-used	30	30	35	35	40	40	40	45	45
3. DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	15	15	15	15	10	10	10	10	10
4. DFT efficacy in test volume reduction. reduction magnitude	2×	2×	5×	5×	5×	10×	10×	10×	20×
5. DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	40	40	45	45	50	50	60	60	70

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known

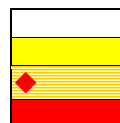


Table 17b Design for Test Technology Requirements—Long-term Years

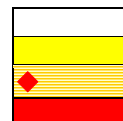
Year of Production	2014	2015	2016	2017	2018	2019	2020
DRAM ½ Pitch (nm)(contacted)	28	25	22	20	18	16	14
<i>System Driver: Analog/Mixed-signal/RF</i>							
1. All-digital DFT for analog/mixed-signal/RF circuits and systems. % digital circuits in DFT implementations	60	80	85	90	90	100	100
2. Correlation of DFT results with existing specification-based test methods. % results correlated	60	80	85	90	90	100	100
3. Availability of fault/defect models for DFT-oriented test methods. % % AMS/RF blocks with accepted fault models	60	65	70	75	80	85	90
<i>System Drivers: MPU/PE/DSP/</i>							
1. DFT coverage of digital blocks or subsystems. % blocks with DFT	80	85	85	90	90	95	95
2. DFT for delay test of critical paths. % paths covered	70	70	80	80	90	90	100
3. DFT for fault tolerance in logic blocks. % blocks with fault tolerance	55	60	65	70	80	90	100
<i>System Drivers: Memory</i>							
1. DFT for yield improvement.	95	95	98	98	98	100	100
<i>General SOC/SIP requirements</i>							
1. DFT-support for logic and other circuit repair (except memory). % blocks with repair	80	80	80	90	90	100	100
2. DFT re-use for performance calibration, and measurement purposes. % DFT circuits re-used	50	50	60	60	70	70	70
3. DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	10	10	5	5	5	5	5
4. DFT efficacy in test volume reduction. reduction magnitude	20×	20×	20×	50×	50×	50×	50×
5. DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	70	75	90	80	90	100	100

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known



アナログ、ミックスドシグナル、RF の信頼性の問題は、誘導時間の過剰や動作環境による、パラメトリックな減衰により更に重要になっている。パラメトリックな問題に対する包括的な理解には、回路に対するうまく作成された故障モデル、特にソフト故障、ノイズに誘導される性能問題(クロストーク、基板ノイズ)、プロセスばらつき、熱効果などが要求される。故障モデルは、それが可能であれば常に、欠陥モデルと物理的に関連付けがなされていなければならない、それによってプロセス品質と制御が、欠陥を削減ないしは消去できるように改善でき、また歩留りを改善できる。アナログテスト手法(アナログの自己テストを含む)のテスト品質を計測しえる意味のあるパラメータを開発することが極めて重要である。多くのアナログ故障は、時間、電圧、フェーズなどの連続量のパラメータが、仕様の範囲からはみ出すことによっており、製造のばらつきや不整合から生じている。効率的で効果的なアナログの故障グレーディングとテスト生成のための故障モデルが必要となる。アナログ、ミックスドシグナル、RF の設計では、故障シミュレーションの計算複雑度を最小化し、同時に高いシミュレーション精

度を維持する EDA ツールが必要である。アナログの故障モデルとプロセスの欠陥モデルを合わせこむことは非常に困難だが、SOC や SIP 上にアナログ、ミックスドシグナル、RF サブシステムを搭載した将来の製品のためには最高に重要なことである。

これらの要求に対する解決策候補は、サブシステムの数だけ多数ある。アナログ、ミックスドシグナル、RF の DFT の問題を解決するために、業界と大学で様々なアプローチが開発されてきた。それらの多くは、例えば PLL、ジッター-BIST、コンバーター-BIST、トランシーバー-DFT などの特定の製品要求にフォーカスするものが多い。DFT の手法は製品種類毎にそのシステム設計とテストを通して最適化するように選択するべきなので、アナログ、ミックスドシグナル、RF の DFT 手法の一本化は期待も切望もされていない。しかしながら、すべての DFT 手法に共通の本質的に重要な機能が、システムレベルの解決策としては絶対必要である。すべてのデジタル DFT 手法は、上記の理由により望ましい解決策になるだろう、つまり、頑丈であり、設計が容易であり、インテグレーションが容易であり、そして CAD ツールの開発やサポートが容易であるからである。現在の DFT 手法では、まだ機能テストとパラメトリック・テストが主流であるが、これらのテストとの良い相関が示されれば、ストラクチャー・ベースの DFT が今後の長期的な解決となる。ストラクチャー・ベース DFT 手法が本質的に有利なポイントは欠陥と故障についてのより深い理解が得られることにあり、これによって品質改善とコスト削減というどのシステムでも最も重要な 2 要件が達成される。一つの特定の DFT 解決策は、チップ上の DFT サブシステムの無線制御と通信のためのラジオ・ラッパーの使用であり、これは現在の設計手法とシームレスに統合されるであろう。オンラインモニタリングが可能になり、外部テスト環境との通信が標準化されたりし、テスト・アクセスの問題は内部的には軽減されたが、一方では、チップ面積、消費電力、ノイズ、回路生成の他の外部負荷などの設計パラメータとの固有のトレードオフが存在する。アナログ、ミックスドシグナル、RF の DFT への複雑な課題を解決するには、創造的な解決策とシステム設計者からテスト・アーキテクトにいたる慣習に捉われない思考が要請される。これらの DFT 解決策は、複雑なシステムインテグレーションにおける、アナログ、ミックスドシグナル、RF のテストがボトルネックを避けるために、時宜を得たタイミングで開発されなければならない。

**2. MPU/PE/DSP システムドライバ** - 近年の GHz のクロック周波数によりデジタル DFT や BIST でカバーされない新たなテスト課題が出てきているとはいえ、これらのシステムドライバは DFT や BIST、また IDDX のような手法については成熟している。ジッターや位相ノイズに対する高周波数クロックでのテストの課題は、先述のミックスドシグナルないし RF の全体的な問題の一部として考慮し得る。論理設計の観点からは、MPU/PE/DSP/メモリ システムドライバに対する要求は良く理解されており、過去 10 年間に多くの課題が DFT と ATE のコミュニティによって成功裡に解決されてきた。将来のことを考えると、より良いデジタル DFT カバレッジを含む主要な要求は、ビルトイン DFT ないし全体的な DFT 手法の一部でなければならない。このビルトイン DFT は、固有のブロックの機能、設計のスタイル、また、故障検出の目標(例えば、ハード故障、ソフト故障、パラメトリック故障)に依存して幅広く変化し得る。カバレッジを著しく改善する必要のある、デジタル・パラメータの二つの例は遅延と消費電力であり、このパラメータのテスト手法を改良するために、DFT と BIST の改善が重要である。故障、欠陥、テストすべきパラメータの広範囲にわたるカバレッジのモニタリングは全体的な DFT や BIST の手法に組み込むことが重要で、特に目標となるパラメトリック減衰や信頼性要求がモニタリングされなければならない。動作環境の中での、オンラインないしはオフラインのカバレッジのモニタリングは、可能性のあるサブシステムの故障を予測したり、分離したりするのに本質的な役割をはたし、それにより故障の修復が適時に実行可能なる。高度に複雑なデジタル・サブシステムのパラメトリック故障による歩留りと製造欠陥を改善するために、フォールト・トレラントで修復機能を有する DFT 手法が要求されている。論理システムのための DFT と BIST フレームワークの統合的なゴールは品質でなければならない。SOC や SIP の様々なレベルの統合(上位レベルで統合されたデジタルブロックからマクロベースの RF やアナログ・ブロックまで)があることで、この解決策はシステムの全体的な階層レベルに拡張可能でなければならないし、個別のサブシステムのテストに関して適切な機能を提供できなければならない。例えば、全体的な歩留りや品質改善のためのフォールトトレランス、キャリブレーション、調整、リペアなどである。



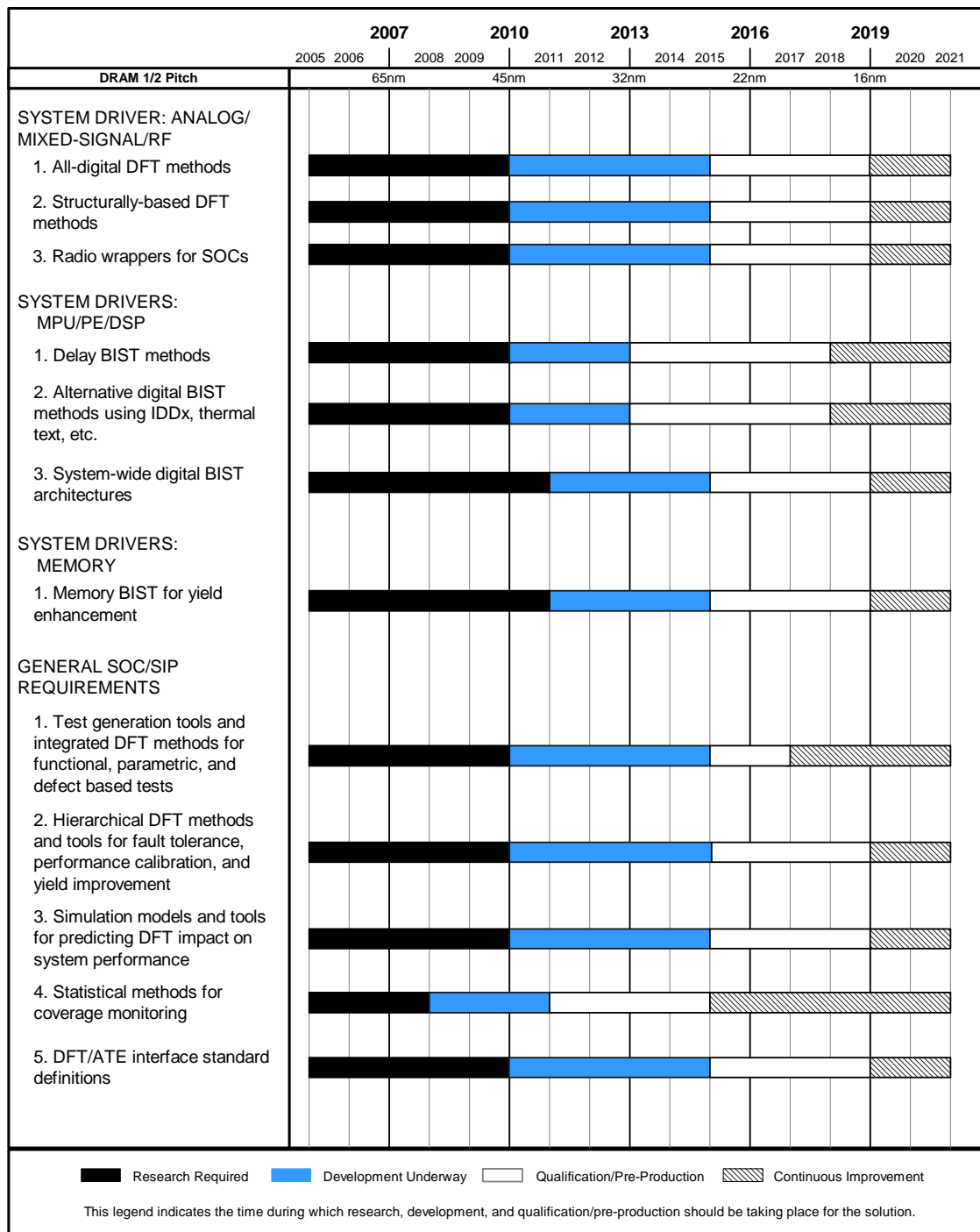


Figure 24 Design for Test Potential Solutions

ここでのドライバに分類される多くの論理システムには成熟した解決策が存在している。残されている課題は、ブロックレベルの解決策よりも、むしろシステムレベルの解決策への要求である。上述で指摘したように、デジタルシステムの性能にかかわる幾つかのクリティカルなパラメータ(遅延、消費電力、ジッター、動作温度など)は、未だに有効な DFT の解決策を求めている。これらのパラメータをキャラクタライゼーション・テストと量産製造テストの両方で決定するために、コスト効率の高い DFT や BIST の手法が非常に望まれる。統合されたキャリブレーションと修復機能を提供するような解決策が、電子システムの設計とテストアーキテクチャの全体に極めて重要なものとして現れるだろう。今日の故障に特化した DFT/BIST 手法と、新しいデジタルのパラメータに特化した DFT/BIST の両方を使って、システムテストの設計者は、様々な手法をそれを適用する局面で最

適に有効に適用することで、結合力のあるシステムワイドな DFT/BIST のアーキテクチャを作り上げなければならない。つぎはき定的なテストアーキテクチャやいい加減なテスト統合は、テスト時間やテストデータ量に関して、ロジスティックスや実行計画に問題を起しがちで、それは DFT/BIST の有利性を減じることになる。デジタル DFT/BIST の多様な手法が与えられることにより、システムテストの設計者は、特定のデジタルシステムに対する一元的に統合されたテストアーキテクチャを作り上げることが可能となる。多様性の中の統一が、どのシステムワイドなデジタルのテストアーキテクチャに最も重要な要件である。

**3. メモリ システムドライバ** - 近年の超大規模メモリブロックが新たなテストの課題を提起しているとはいえ、メモリ システムドライバは DFT、BIST また選択的なテスト手法に関しては、更に成熟している。これらの新たなテスト課題、例えば新しい故障のメカニズム、リード/ライトアクセス時間の不安定性など、は前述のミックスドシグナルのテスト要求の全体的問題の一部として考慮することもできるが、メモリの規模が巨大なので、SOC に統合されたレベルでの分離された問題となっている。メモリ設計の観点からは、テストの一番重要な要求は歩留りの改善である。冗長設計手法が、このテストに関わる問題については非常に大きな貢献をしてきたが、メモリブロックのサイズが増大するにともなう、歩留りの問題は、適切なメモリセルの設計だけでなく、仕様の範囲内でのメモリのタイミング性能を包括するようになっていく。すなわち、歩留り改善のための DFT は、物理的な品質とメモリブロックのタイミング性能の両方の課題を満たすような手法を提供しなければならない。

メモリシステムの物理的な品質の改善に関わる類の問題については十分に成熟した解決策が存在している。上述したように、冗長設計手法と工夫された精緻なレイアウトスタイルに、メモリの故障モデルへの深い理解・把握が結びついて、長期間にわたって歩留り改善を達成してきた。残ったタイミング性能の課題は、DFT/BIST 手法が、特に大規模のメモリやプログラマブルメモリに対して、リード/ライトのアクセス時間を最適化し、信頼性を向上させることである。

**4. 高度に統合された一般的な SOC や SIP** - 既設計ブロックを大規模なデバイス上に統合して行くことで、それらのブロックが単一種類(すべてが論理ブロックとか)であったとしても、線形を超えた複雑度の増大を設計ツール、DFT、そして製造時テストにもたらす。ますます、多様な種類の回路ブロックが統合されつつある。論理ブロック、SRAM そして DRAM は近年では共通して統合されてきたし、今日ではアナログ、ミックスドシグナル、RF、また不揮発フラッシュ回路が論理ブロックや RAN と組み合わせられて来ている。統合されたデバイスについての、シリコン複雑度やコストは比較的予測可能である。しかしながら、搭載されたブロックとデバイス種類が混交であることで、テスト可能性や設計検証、また製造時テストに対して、線形以上で予測不可能なコストの増大をもたらしている。大規模な論理ブロックとして丸々搭載されている ASIC や MPU マクロでは、この問題は既に顕在化しており、実際製造時のテストのコストがシリコンコストを超えている。DFT をとってさえ、そのコストは線形以上になる。この節で述べる DFT/BIST 要求は、先行する三つのシステムドライバでカバーされたブロックレベルの DFT の単なる寄せ集めではないことを、ここで強調しておきたい。システムの範囲に拡大された DFT 要求と、その解決策は、DFT/BIST の全体的なロジスティックスとスケジューリングの解を提供しなければならないし、ブロックレベルではカバーできない、ないしは見過ごされるテスト機能を含んだものでなければならない。

システムのレベルでは、DFT はチップ上のそしてシステム内のテスト生成とテストアプリケーションの手法を提供し、それによってテスト時間と外部 ATE を使ったテストボリュームの負担を軽減しなければならない。テスト生成の専用ハードウェアと組み込みアルゴリズムは、テストアクセスの必要性を削減するし、一方では究極のシステムレベル DFT の要求である、システムのレベルでの BIST への基底環境を提供する。アナログ、ミックスドシグナル、RF サブシステムの DFT 技術の改善にともなう、テスト生成とアプリケーション資源は、システムテストの環境にスムーズに統合されなければならない。DFT によって検出される故障(いくつかの種類のハード故障と多くの種類のパラメトリック故障)は、歩留り改善とタイムツーマーケット時間短縮のために修復されるべきである。したがって DFT とシステム設計メソッドロジは、DFT と協力してこの修復機能を提供しなければならない。更にグローバルで魅力的な DFT の側面は、本来の場所での計測とフィードバックを通してのキャリブレーション

ン機能にある。プロセスばらつきやデバイスの不完全性に対してして、システム性能を調整するために、設計者によって採用されている多くのキャリブレーション手法は、DFT 手法と本質的に等価なものとして同一視できる。システムキャリブレーションのメソッドとして DFT を使用することで、設計者が設計とテストの緊密な連携をとるようになることが促進される。オンラインの測定、フィールドリペア、性能低下の修復などは、包括的な DFT メソッドによって対処が可能になる基本的な性能要求である。

SOC や SIP に搭載されたブロックのテストには、それが単体のときに比べて数桁以上の長いテスト時間が必要になる。SRAM、DRAM、ロジック、フラッシュ、アナログ、ミックスドシグナル、RF の各々に対するテスト手法、ATE、製造インテグレーションは、独自の最適化のために根本的に異なるスタイルをとっており、ゆえに、これらを SOC として統合した場合には、そうしたスタイルが破綻するからである。単体のアナログや RAM では、異なった種類の ATE やテスト装置のコスト、また高度な並列処理によって、本来非常に長くかかるテスト時間を短縮できているが、こうした手法が SOC などでは適用できないのである。再度いうと、SOC としてインテグレートされた場合には、そのブロックの I/O へのアクセスが不可能であったり、妨害されたりするが、単体では可能なわけではこの差による。このため、DFT 手法やブロック間のプロトコルの拡張だけではなく、BIST と組み込みソフトを使った自己テストの使用を拡大することが必要である。SOC に統合されたアナログ、ミックスドシグナル、RF ブロックの DFT や BIST は長いこと研究領域のテーマだったが、SOC への統合の頻度が高まることにより、業界での実現技術となるだろう。

DFT 統合への様々な障害により、チップ面積、テスト目的の I/O 配置、消費電力、バンド幅、信号感度などのシステムインパクトに関して、DFT は金がかかるという認識が広くあった。アナログ、ミックスドシグナル、RF と超高速のデジタルサブシステムを SOC や SIP に統合することで、リーク電流や、ノイズ、ノードの負荷などの DFT によってモニターされるべき難しい課題が生じる。バンド幅の損失や付加的なノイズなどのシステムパフォーマンスへの DFT の影響は、定量化され十分に予測されなければならない。それによって、統合されるべき DFT 手法の早期の利点とコストの競合がスタディできる。DFT 手法は、更に低感度のノードのモニタリングや間接的なテスト結果の見積り(例えば、故障検出やパラメトリック計測)にも注力する必要がある。

DFTの最大の貢献は、テスト量とテストコストの低減にある。このDFTの貢献は、SOC/SIPのレベルで実証されるべきである、というのはこのレベルでテスト量とテストコストがはっきりと、そして容易に計算できるからで表の中のテスト量削減の目標値は、システム内の特定のブロックに対するテスト量やテスト時間ではなく、全体システムをカバーするテスト量とテスト時間である。例えば、2006 に対して 2 倍の削減要求はデジタルサブシステムとしては控えめだが、これがアナログRFを含む全体システムであることを考えれば妥当な要求値である。テスト時間やテストコストは、ATEと非常に強く関連している、というのはDFTやBISTがあっても、なおATEがテストでは重要な役割を持っているからである。様々なDFT手法とATEとの間のインタフェースはきちんと定義しなければならない、それによって固有の製品のテストが要求する異なったDFTとATEの選択と組み合わせの自由度が許容される。テストアクセスポートやその他のテスト関連の標準(IEEE1149.1, 1149.4, 1450, 1500)<sup>111213</sup>が整備されているとはいえ、広範はシステムレイヤ(動作物理的なレイヤ、データ通信のレイヤなど)に拡張された包括的なDFT/ATEインタフェースプロトコルが確立されるべきである、それができればDFT手法の貢献とATEの可能性をフルに活用できるようになる。

SOC/SIP の DFT/BIST の全体的な課題に対する解決策は、過去数年開発されて来ており、継続して非常に速いペースで進んでいる。SOC や SIP の中のテクノロジーや、サブシステムの設計手法の多様性は、テスト生成ツールとDFT/BIST ツールがデジタルテスト手法(故障、欠陥、パラメトリックベース)とアナログ、ミックスドシグナル、RF テスト手法(機能、パラメトリックベース、将来的には故障、欠陥ベース)が統合されることを要求して

<sup>11</sup> IEEE Std 1149.1, IEEE Standard Test Access Port and Boundary-Scan Architecture

<sup>12</sup> IEEE P1450.6, Draft Standard for Standard Test Interface Language(STIL) for Digital Test Vector Data—Core Test Language(CTL)

<sup>13</sup> IEEE Std 1500-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits.

いる。この解決策は、システム複雑度、設計工数、タイムトーマーケット、そして全体のテストコストを低減するために、設計やテストプランニングのツールと伴って開発されなければならない。いかなる階層のレベルでも、効率的なシステム解決策が、設計者にとって使用し得る互換性のある DFT/BIST を提供しなければならないし、設計の進行に応じて設計者の様々な選択と統合されなければならない。全体的な解決策は最終的には、設計とテストの両方の最適化を密に反映した統合でなければならない。

効率的なシミュレーションモデルとツールは、すべての設計とテストのプランニングに内在するものであり、テストソリューションの一部でなければならない。特に、システム性能に関わる DFT/BIST の影響を見積もることにおいてそうである。シミュレーションツールは、設計の工程が様々なレベル(動作、回路、レイアウト)を進行するにしたがって、DFT/BIST の影響の見積りを提供し、それによって、最も適当な DFT/BIST の手法が選択できるようにしなければならない。全体を通じての効果指標は、システムテストとカバレッジ(故障、パラメトリック、機能、ないしはそれらの組み合わせ)である。それは、DFT/BIST の一部として算出されるべきであり、テストの間と動作の間の両方において観測されるべきである。オンラインでカバレッジが観測できる機能は非常に強く求められており、最終的には製造品質の改善を可能にする故障の統計や欠陥の統計に関連づけができるようなカバレッジの統計のアルゴリズムが開発されなければならない。

システムワイドの DFT/BIST 解決策の中で、最も議論的になるのは確実に、チップ上の DFT/BIST 機能と ATE の間のインタフェースである。多くのテストプロトコルが少なくともデジタルシステム用には存在しているし、近年ではミックスドシグナル・システムについても存在している。しかし、SOC/SIP の全体のわたる、インタフェースの標準は現状では欠落している。テストのためのリソースの分割の仕方が業界の中で大きく異なっているため、DFT/BIST と ATE の間のインタフェースは十分に定義されて来なかったが、将来の解決策においては、以下の二つの重要な要件に関して、このインタフェースの定義を持たなければならない。一つは、厳密で明確な定義によって、システムテストにおける様々な ATE の可換性や効果的使用が可能になること、もう一つは、柔軟なインタフェースの定義によって、電子システムの設計とテストアーキテクチャの制約が解消されることである。

## 製造容易化設計(Design For Manufacturability (DFM))

ばらつきの増大に伴い、マスクコストとデータ量の爆発、そしてリソグラフィ装置の限界が集積回路の製造容易性での重大な設計課題を引き起こしている。

1. **アーキテクチャでの課題** - 回路の歩留りを満足することが困難なためにアーキテクチャでの冗長性が必要とされるであろう。この抽象度ではこれ以上のことを行うのは難しい。
2. **論理と回路設計での課題** - デジタル、ミックスドシグナルでの適応型回路の必要性は増すであろう。消費電力、タイミングの収束性を含んだ、統計的な設計が基本となる。ただ、実際に設計を行うには、以下の重要な 2 点を明確にすることが避けられない。
  - 1) 統計的設計ツールへの入力となる、キャラクタライズとモデリング
  - 2) 統計的解析から、更に計算複雑度の高い最適化への発展
 最終的には、統計的解析ツールと統計的最適化手法は、大雑把で精度的に疑問のあるモデルではなく、実際の製造と設計に起因するばらつきを反映することが必須である。統計的解析と最適化の過程での種々のばらつきの合成は、プロセスのキャラクタライズでの統計的なメトリックを通じての初期の分解手法と整合が取れていなければならない。合成と分解の不整合があると、時間のかかる計算結果に対して、不必要な誤差と、疑問となる値を導入してしまう。
3. **レイアウトと物理設計での課題** - まず最初に、デザインルールチェックの複雑度が増大する(グラフを含んで)。ルールは 2 層構造(要求ルールと、推奨ルール)へと展開して、更にはおそらく 3 層構造かあるいは、

層のない構造(合否ルールではなく、設計者がテラアウトでの基準として適応できる、歩留りと面積コストのパレートカーブを提供する)へと進むかもしれない。このチェックは、設計者にその複雑さを意識させることなく行われることが必要である。2 番目に、リソグラフィ装置の解像度の限界のために RET(resolution enhancement techniques: 解像度補正技術)の影響を更に明示的にデザインフローに適応することが要求される。RET ツール、たとえば OPC や、CMP(化学的機械研磨処理)でのフィリングはタイミングや消費電力などの回路的な指標を明示的に意識する必要がある。それにより、全体として、目的に沿ったツールの整合性を保ち、歩留りの向上を可能とし、製造コストの削減と、マスクデータ準備のための時間を改善することに繋がる。このアプローチは必然的に密接な統合フローとなり、回路的な意図を下流へ伝え、個々のツールの勝手な修正による精度劣化を防ぐことになる。結果として、レジスタートランスファーレベルから GDS データ(RTL2GDS)へのフローにおいて、適切な RET や OPC の修正を下流へとつなげるようにプランニングする必要がある。たとえば、グローバル、ローカル配線層の密度は、概略配置の段階で、ダミーフィル挿入の場所を考慮する必要があり、これにより事前の CMP フィリングとその容量の見積りが可能となり、これと同時にクリティカルなネットの情報を最終レイアウトフェーズ、マスクデータ生成(MDP)や OPC レベルへ伝達することができる。

4. **設計課題としての歩留り予測と最適化** - ここでの基本ルールは、もはや”固定値”ではない、なぜならここでのシナリオは、設計改善よりむしろ設計の受け入れを左右するものであるためである。適切な歩留まりの成熟を得て、急峻な歩留りの立上げを可能とするには、有意義な設計ルールの緩和の戦略を用いることが必要となる。これらの”推奨ルール”は、設計レイアウトとウェーハプロセスの要求、例えばアライメントの許容度、光近接補正、RET 向上やその他多くの制約との間の相互の関係から得られる。DFMとしての対策は、歩留りのための面積、消費電力、そして速度のチューニングへの影響を与えるため、設計毎に異なってくる。設計過程において歩留り、面積、そして速度の相互の影響が分析され、かつ商用的な有用性のトレードオフが得られる必要がある。機能的およびパラメトリックな歩留りに対する DFM 対策は、設計の後処理(これは往々にして限定的な結果しか得られず、かつ多くの場合、処理時間がかかる)としてではなく、むしろ設計フローの中で新たな最適化機能として統合されるべきである。この意味するところは、歩留りの予測を、設計のプランニング、合成、配置、配線などのツールへ統合し、歩留り、パフォーマンス、消費電力、信号の完全性、そして(歩留りとの関連で明示的に新たな設計指標としての)面積などのすべての設計目標を考慮することを意味する。つまるところ、歩留りはプロダクト固有の設計属性と、プロセス固有の 故障の可能性の両者の関数である。そのため、ある特定のプロセスの限定的な故障/マージンのパターンを緩和するために最適化された設計は、異なるプロセス条件では事実上歩留りが低い可能性もある。そのため、配置配線ツールにおいて、異なる実装での歩留りコストの正確な評価を可能にするには、実際のターゲットプロセスでの論理ライブラリの正確な歩留りモデルを事前にキャラクタライズしておくことが必要となる。ライブラリの歩留りモデルは、プロセスの立ち上げから成熟段階にわたり頻繁に更新を行うことが必要である。

Table 18 は、上記の主要な課題を、DFM 要求として定量化したものである。Figure 25 は、これらの要求に対する DFM としての時間軸での解決策である。



Table 18a Design-for-Manufacturability—Near-term Years

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013	Driver
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32	
Mask cost (\$m) from publicly available data	1.5	2.2	3.0	4.5	6.0	9.0	12.0	18.0	24.0	SOC
% V <sub>dd</sub> Variability % variability seen at on-chip circuits	10%	10%	10%	10%	10%	10%	10%	10%	10%	SOC
% V <sub>th</sub> variability Doping Variability impact on VTH	24%	29%	31%	35%	40%	40%	40%	58%	58%	SOC
% V <sub>th</sub> variability Includes all sources	26%	29%	33%	37%	42%	42%	42%	58%	58%	SOC
% CD variability CD for now, might add doping later	10%	10%	10%	10%	10%	10%	10%	10%	10%	SOC
% circuit performance variability circuit comprising gates and wires	41%	42%	45%	46%	49%	50%	53%	54%	57%	SOC
% circuit power variability circuit comprising gates and wires	55%	55%	56%	57%	57%	58%	58%	59%	59%	SOC

Table 18b Design-for-Manufacturability—Long-term Years

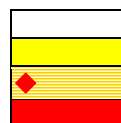
Year of Production	2014	2015	2016	2017	2018	2019	2020	Driver
DRAM ½ Pitch (nm)(contacted)	28	25	22	20	18	16	14	
Mask cost (\$m) from publicly available data	36.0	48.0	72.0	96.0	144.0	192.0	288.0	SOC
% V <sub>dd</sub> Variability % variability seen at on-chip circuits	10%	10%	10%	10%	10%	10%	10%	SOC
% V <sub>th</sub> variability Doping Variability impact on VTH	81%	81%	81%	81%	112%	112%	112%	SOC
% V <sub>th</sub> variability Includes all sources	81%	81%	81%	81%	112%	112%	112%	SOC
% CD variability CD for now, might add doping later	10%	10%	10%	10%	10%	10%	10%	SOC
% circuit performance variability circuit comprising gates and wires	58%	61%	62%	65%	66%	69%	69%	SOC
% circuit power variability circuit comprising gates and wires	59%	60%	60%	61%	61%	62%	62%	SOC

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known



DFMとしての要求は、次に掲げるカテゴリに分類される。

- ・ 経済原則に立脚する要求-このカテゴリはマスクコストを含み、これは数百万ドルのレベルに達し、そのため小さな会社や、新興マーケットの団体から起こるであろうシステムオンチップの技術革新を危険にさらすことになる。
- ・ ばらつきや、リソグラフィ限界からの要求-このカテゴリは、デバイスや配線と直接に関連している下流レベルの抽象度での限界と、設計される回路全体に関連する上流レベルの抽象度での限界を含んでいる。下流レベルでは、定量化された要求値として、(オンチップ回路に給電される)電源電圧のばらつき、(ドピング濃度のばらつきと全体としてのトレンドの両者の影響としての)閾値電圧のばらつき、そしてCD(Critical Dimension)ばらつきの割合がある。上流では回路パフォーマンスのばらつきの割合(回路でのクリティカルパスのように、チップ全体のパフォーマンスを決定する回路速度の不確かさの割合)や、回

路消費電力のばらつきの割合(アクティブと待機状態での両者の電力を含む、消費電力の不確かさの割合)を要求として持つ。

マスクコストなどの経済上の要因は、直接的に制御するのが困難であり、増大していくことが予測される。それに変わる次善策として、マルチプロジェクトウェーハ、コンフィギャラブルロジックやストラクチャード ASIC が提供される。ばらつきに関して、CD コントロールの様な”プロセスレベル”のパラメータや、供給電圧の様な”回路レベル”を含んだ様々なパラメータやその目標値は設計で制御される。(素子の領域を反転するチャンネルドーピングの影響を含む)閾値電圧のばらつきの様な他のパラメータは不可避に増大し、結果として10年後か、それ以前に危機的な状況となる可能性を秘めている。もし根本的な新しい解決策が発見されなければ、プロセスレベルからデバイスレベル、回路レベルへとパラメータを上流へ伝達することによって、非常に大規模で総合的な回路性能や消費電力の変動に対処する必要がある。

DFM の技術的要求こと項に対処するには、Figure25 に記載されている DMF の解決策が必要となる。

- ・ **基本的な経済的制限に対応した解決策** - マスク費用以外を含んだ経済的要因を明らかにするために、主なアルゴリズムやインタフェースにおいて、DFM ツールが必要になる。チームやマネージャは、難しいDFMのトレードオフの経済的価値について、より直接的に評価できるであろう。
- ・ **ばらつきの影響に対する解決策** - 将来DFMはばらつきの様々な特徴を取り扱うようにならなければならない。第1に、性能と消費電力の両方のばらつきに対応しなければならない。その結果、統計的機能解析、有効電力、漏れ電力解析ツールが大変重要となる。第2に、システマティックとランダムという2つの特徴ある統計的な歩留り損失を分類する必要がある。第3に、電源供給、温度、閾値電圧を含んだ、ばらつきを引き起こす環境やプロセスの様々な要因の最適化が必要である。最後にツールベースによる解決策は、十分ではない。
  - 1)回路動作、電源、クロックや入力信号のばらつき要因を理解し、最適化する高性能な適応型回路
  - 2)局所的に非同期な設計、冗長性、誤り訂正符号化(ECC)を含むばらつきに対する基礎的な耐性に関するアーキテクチャを含む、ばらつきを補償する設計技術が必要となる。
- ・ **リソグラフィ制限の影響に対応する解決策** - リソグラフィが果たす重要な役割の為に、ITRS はリソグラフィに関連した DFM 問題や解決策を必要とする分野に取り組んだ。将来の設計フローは、リソグラフィ制約に対応した本質的な取り組みが必要となる。これらの技術は、おそらくルールベース(ツールと設計フローまたはその一方の変更無しに)とモデルベース(直接的にツールと設計フローまたはその一方を変更する)の両方のレイアウト修正を含むことになる。第1に、今日レイアウト後に適用されているRETは、論理合成、タイミング検証、配置配線といった従来の設計工程とますます相互にやり取りすることが必要となる。そして更に明確に性能と消費電力の相関関係の測定基準を盛り込むことが必要となる。このやり取りは、レイアウト検証や論理合成といった”モデルベース”型の設計工程に於いて、直接的、或いは間接的に行われる。第2に、従来の設計基準や設計は、設計の段階で基本的に製造可能となる様に、益々”製造との調和(manufacturing friendly)”が必要となる。製造と調和した設計基準(効果的な製造性の原則に従う”信頼性”のある厳格な規準)、完全に制限された設計基準(斜め配線を使用しない格子状のレイアウトの様な、小さい領域か性能コストで製造性を確実にする簡単な設計基準)、そして自動配置配線ツールと調和したスタンダードセルやコアなど、設計基準は解決策の主要な要素となる。

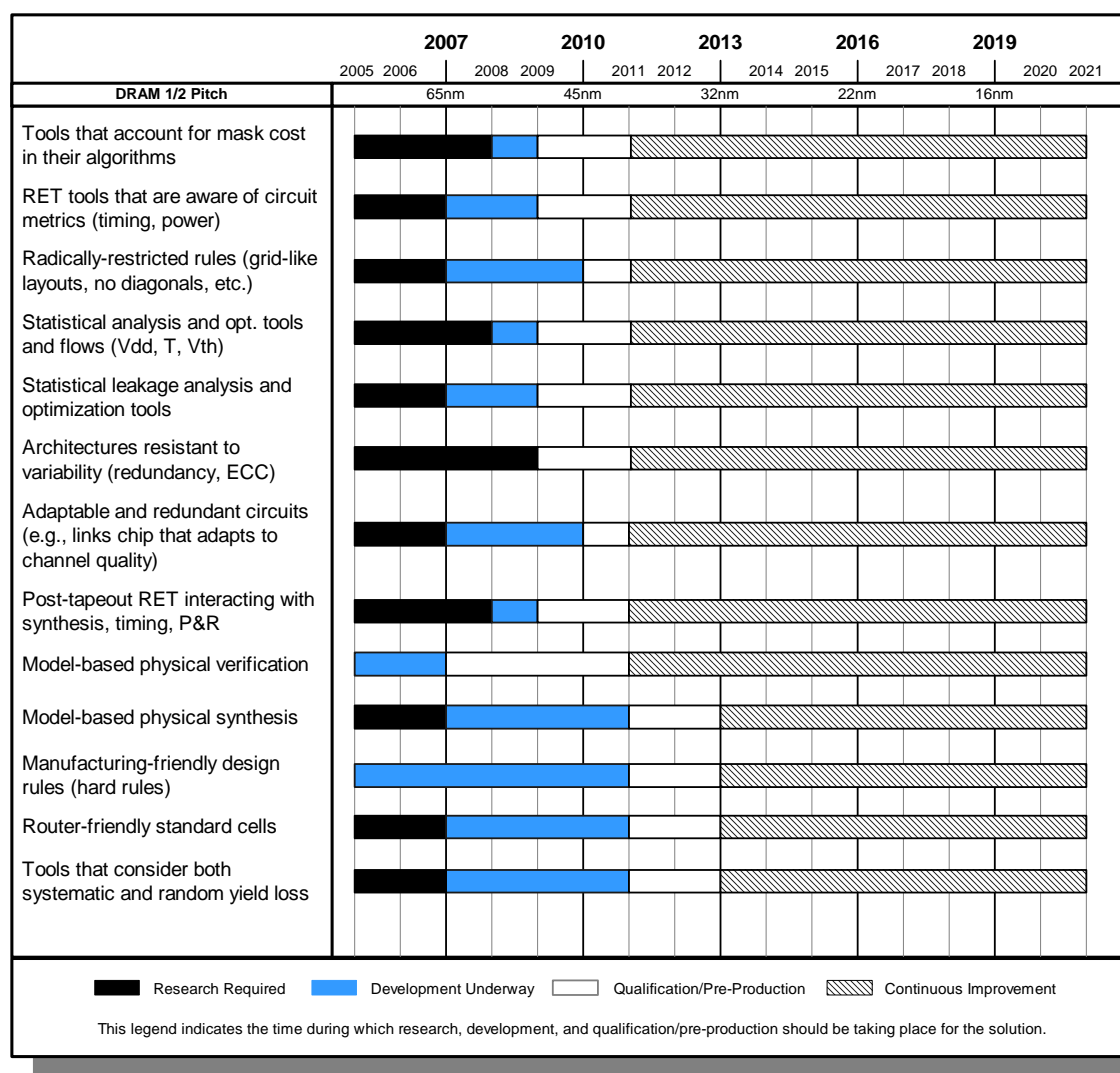


Figure 25 Design for Manufacturability Potential Solutions

初期では、タイミング解析のような直接ばらつきを取り扱う新しい解決策が登場する。5 年以内には(2008 年から 2010 年)統計的な方法論が完全に設計に取り込まれる。当面は、成熟した技術の選択的な応用や、高付加価値品や大量生産品で使用される高度な設計技術の一部が適用される。

リソグラフィを直接的に補完する DFM 技術は一般的となったが、より広範囲な製造モデルや製造データと密接に連動した運用レベルの設計技術となるには更に時間がかかる。次の 10 年では、チップは製造上の課題に満ち溢れている。しかしながらその時までには、設計時に直接製造容易性を改善するのに使用する情報以外に、設計技術は設計者が意識する必要のない運用レベルの DFM 技術によって徹底的に再整備される。

### アナログ、ミックスドシグナル、RF 設計技術のトレンドと課題

アナログ、ミックスドシグナル(AMS)および RF 回路は、デジタル回路とは異なる。これらは、"high" ( $V_{dd} - \text{ノイズ許容値}$ ) と "low" ( $V_{ss} + \text{ノイズ許容値}$ ) で決められた 2 状態の電圧レベルで表現される情報ではない。同様に離散的な時間的期間(例えば、クロック信号が high)で表現される情報ではない。アナログと RF 回路で処理される信号は、非常に高い精度で(または、時間的および振幅的に小さい許容誤差で)、時間的および振幅的に連続している。したがって、回路で使用される各素子の線形性、ノイズ、寄生効果、および電気的な不均一性などの理想的でない特性が、回路内で処理されるアナログおよび RF 信号に直接的に歪みとノイズを

及ぼす。デジタル回路には、各論理ゲートの遷移時における大きな利得により、これらのノイズ源を高いレベルで抑圧する能力が備わっている。しかし、この単純な信号回復の原則は、取り扱う信号のダイナミック・レンジが非常に広い場合、アナログとRFの信号処理では適用することができない。動作速度の問題、また、信号回復回路自身が影響を受けやすい信号を防御する以上に、多くの雑音と歪みを作り出すという事実が、これらの問題をアナログ領域ではかたがたに挑戦的かつ困難な課題としている。

アナログとRFの設計は、素子の理想的でない特性のすべてについての高精度なコンパクトモデリング、キャラクターライズ、特性抽出の必要性を求めてきた。しかしながら、上記の問題はアナログおよびRF回路設計(または、レイアウト設計)において、より高いレベルへの抽象化のためのルール導出を非常に困難にしている。デジタル回路で使用されるツールは、アナログ設計においては適切なものではなく、非常に不正確であった。歴史的に見て、チップ当たりのアナログおよびRF用トランジスタ数の増加はデジタル用トランジスタ数の増加と比較して非常に緩やかであるため、アナログとRF設計者をシステムレベル設計ツールから遠ざけてきた。回路とシステム設計については、この章の後半の「論理、回路、物理設計」節で述べられる。今日、アナログとRF設計において知られていたものと同様の課題が、デジタル設計(IR ドロップ、クロストークなど)においても発生し始めている。SOC デジタル設計では、アナログおよびRF回路とチップ領域を共有する。常に短縮化されるtime-to-marketとより高い生産性へのニーズは、近年において歴史的なパラダイムを変えてきた。そして今日、EDA 業界によるアナログとRFに特化した新しい取り組みが現れている。

**システムドライバ**で述べたように、AMS 設計におけるスケーリングと製造技術移転(migration)に関する多くの課題がある。これらの課題には、供給電圧の低下、相対的なパラメータ変動の増大、チップ当たりのアナログ・トランジスタ数の増大、信号やクロックや素子自体の高速化、SOC としての集積化におけるリーク電流やクロストークの増大、そして設計スキルや自動化の不足への対応が含まれている。個々の課題には、次のものがある：(1) “デジタルでさえアナログとなる” ディープ・サブミクロン効果と高い信号・クロック周波数：これは寄生効果の増大、伝送線路効果、シグナルインテグリティ問題を招く；(2)アナログ設計に費やす労力を少なくするためのアナログ合成ツール；(3)歩留り向上を実現する“製造容易化設計”；(4)移動体通信デバイスにおける信号処理システム(そのほとんどはデジタルである)とアナログ RF フロントエンドの高密度な集積化；そして(5)電子部品と MEMS などの非電子部品の高密度な集積化。これらの課題は、後述する設計技術領域についての解説の中でより詳しく述べられる。

1GHz以上の周波数ではインダクタ、信号配線、伝送路、電子的な同調容量などの分布定数素子の正確な計算が不可欠である。これらの計算は信号の表皮効果と、波形歪につながる周波数依存の位相速度を考慮に入れて、マクロモデルを構築する必要がある。このような計算の重要性といくつかの興味深い結果が、標準的な130nm CMOS プロセスを用いた43GHz VCO(電圧制御発振器)の設計例として2005年の論文で報告されている。この論文では配線幅5 $\mu$ m、配線間隔10 $\mu$ mで、長さが100 $\mu$ mの最上層Cu配線の1巻きでインダクタが実現される。この単純な長方形のレイアウトは83pH、Q値35のインダクタを形成した。2.5D電磁シミュレーションが磁界と表皮効果の正確なモデリングに利用された。キャパシタはおよそ100 $\times$ 100 $\mu$ mのサイズで実現された。

10GHz(真空中の波長 $\lambda$ は100GHzで3mm!)を超える周波数領域ではアンテナ効果が重要である。チップ上に形成されるアンテナは高度な数値計算手法、例えばFDTD(finite difference time domain)法を用いて設計しなければならないが、そのアンテナが信号配線間の寄生効果で形成される場合は、膨大な数となるため計算精度を少し低下させたとしてもより単純化した手法で算出する必要がある。このような高周波領域での設計問題は、定在波・進行波発信回路を用いた1GHz以上のデジタルクロック信号生成のための高性能アナログ回路を使う場合にも発生し、解決されている。

マクロモデル構築には上述の作用のほとんどすべてが影響を与える。信号配線間の容量性および磁気カップリング、パッケージングされたチップの熱特性、アンテナの近距離および遠距離寄生効果、内蔵されたMEMS動作特性などは2Dまたは3Dの電磁界解析手法を用いて、10の4乗から8乗の数の方程式で表現



される。計算精度の大幅な低下を招くことなしに、この方程式は「次数低減法(order reduction methods)」として知られている様々な数学的方法で、次数を 10 の 2 乗から 4 乗まで低減することができる。この領域での解析技術の著しい進歩により、むこう数年間で有効な CAD ツールが実現されると考えられる。このような新しい自動生成されたマクロモデルは、ディジタル回路のタイミング動作やシグナルインテグリティ解析と同様に、アナログ回路設計でも活用されることとなる。

### AMS と RF 用システムレベル設計

システムレベル設計における AMS の主要課題は、アナログ回路の非スケーラビリティと、アナログ・ビヘイビアのモデリングおよび回路合成である。SOC に統合されるディジタル回路やソフトウェアと共に、システム全体の機能とインタフェースの分析を可能とする、設計言語によるモデル化手法にもとづいた、自動化されたアナログ回路合成および回路最適化が必要とされている。課題は次の 4 点である。シミュレーションにおいて、ディジタルでは GHz 単位で 1 秒までの周波数変化に対して、GHz 単位で 100 秒までの周波数変化をとまなうアナログ設計のタイムスケールに対処すること；異種のテストベンチを作成しカバレッジを確保すること；系統的にトップダウンでの制約の伝達を達成すること；そして、機能的および構造的な表現を混在させること。

### AMS と RF 用 論理、物理、回路設計

論理、物理、回路設計では、キーとなる課題はアナログ合成である。スケーラブルな SOC デザインは、アナログ設計のボトルネックを取り除くことが必要である。システムレベル設計と共通の技術ニーズは、再利用可能で、retarget 可能なアナログ IP ジェネレータである。PLL、オペアンプ、パワーアンプなど特定回路に専用化された現在の回路合成は、より一般的な対象回路へ拡張される。レイアウト合成は、高性能なアナログ設計(例えば、ミスマッチに敏感なトランジスタのために交差接続させたレイアウトなど)のニーズに対処ようになる。アナログ・ポストレイアウト・シミュレーションでは、衝突電離や熱的非線形性、ローパス・フィルタとして作用するボディ・コンタクトなどによる歪みと非線形性の増大に対処する。また、合成技術は、アナログ・ディジタル混在における素子のミスマッチ補償など、増え続ける製造ばらつきへの対応を取り扱うことになる。短期的には、光インタフェース回路や Q 値の高いオンチップ・インダクタやチューニング可能な共振器のための新しい合成ツールが必要とされる。長期的な対象回路としては、超低電力なセンシングとセンサー・インタフェース回路、そして、微小光学(ビーム・ステアリング)素子である。

### AMS と RF 用 設計検証

設計検証では、AMS 回路は回路構成に関してよりもむしろ、“仕様に対する”チェックを必要とする。シミュレーションの高速化が歴史的なソリューションである一方で、新しい検証ソリューションは、統計的手法、精度を向上しつつシミュレーションを高速化するコンパクトモデル、そして、新しい合否判定基準を取り込むことになる。また、AMS 設計は、いまだ未成熟である混載システムの検証を短期に解決しなければならない課題とする。したがって当面の課題は、現在のアドホックなアプローチを改善するあらゆる手段によって、より有効なソリューションに向かう経路を見つけることである。MEMS、電子・光学、電子・生物学などの素子が単なる変換器を越えるものとなるにつれ、これらの異種部品を集積したシステムをモデル化し、分析し、検証することが今後の課題となる。カーボン・ナノチューブ・トランジスタ、単電子トランジスタ、共鳴トンネル・ダイオードのような、複雑な物理効果を利用した非従来型デバイスで構成された回路においても同様である。

Table 19 に 2009 年までに予想される短期的な AMS 設計技術のブレークスルーを示す。あわせて、2005 年 3 月版 MEDEA+ Design Automation Roadmap における、AMS 設計技術の要件に関する秀逸な議論も参照されたい。



Table 19 Near-term Breakthroughs in Design Technology for AMS

Field of Breakthrough	2005 State-of-the-Art	2006/07	2008/09
Specification, validation, verification	Established AMS Hardware Description Languages	Multi-language support, AMS extension of HW/SW description languages for full system simulation	Complete specification-driven design flow; some specialized formal verification methods
Architectural design	Algorithm-oriented design (e.g., with Matlab/Simulink)	Language-based performance evaluation; closer coupling of architectural, block, and circuit level	Synthesizeable AMS description; power-aware HW/SW partitioning extended to AMS systems
Physical mixed A/D and RF design	Procedural layout generation, module generators for a few block types	Module generators for often re-used blocks, design centering, performance estimation	Synthesis: behavior to layout (at least for the most important building blocks)
Parasitics extraction, automated modeling, accelerated simulation	Electromagnetic immunity simulation works but is too complicated for broad usage	2D/3D model-based order reduction for interconnect systems and substrate effects on chip, thermal package modeling	New fault-tolerant circuit architectures, robustness against technology parameter variations; order reduction for all kinds of parasitics and antennas

最後に、テスト設計に関しては、アナログ回路はミックスドシグナル SOC の全チップ領域のごく一部であるにもかかわらず、量産テストコストを支配する。アナログテストの変革がなされない限り、ミックスドシグナル製品のコストにおけるアナログテストのコストの比率は増大しつづけるだろう。短期的な要求は、アナログ/ミックスドシグナル用の DFT/BIST であり、特に、ベースバンド用以上に高分解能かつ高周波数への対応である。オンチップ化された 14-16 bit の高分解能 ADC と 1-5GHz の高速 RF 素子のためのテスト技術は、費用対効果に優れているだけでなく、副作用のないものでなければならない。すなわち、それらは内蔵されたアナログ・ブロックの性能を低下させてはならない。高分解能 ADC は通常、直列接続された多段回路もしくはカスケード型変調回路(mash 回路)で構成される。このためひとつの可能性ある方向性は、DFT またはセルフテストの戦略を開発するに当たって、この構造的な知見をうまく利用することである。

[PIDS 章](#)では、アナログ CMOS トランジスタはより高いアナログ電源電圧で動作し、2-3 の技術ノードにおいてスケールアップされないという特質を述べている。しかし、このことは消費電力、プロセス互換性、面積効率、設計複雑度、検証とテストといったクリティカルなコスト問題を解決しない。更に AMS 設計生産性は、新しいミックスドシグナル部の開発のためのキーとなる課題を残している。AMS 設計技術のための短期的なロードマップは、以下の項目の解決に向けたツールおよび、新しい設計記述言語を含んでいる。

- チップとパッケージを考慮した、回路システムの探索
- 回路合成とサイジング
- 回路図の等価性検証
- 製造容易化設計
- アナログ/RF レイアウト合成
- 容量性、誘導性、熱的な寄生効果の抽出、自動モデル生成、およびシミュレーション高速化
- アナログ IP の回路情報要約化と再利用

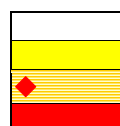
## 設計技術に対する付随的な要求

Table 20 Additional Design Technology Requirements

Year of Production	2005	2006	2007	2008	2009	2012	2015	2018	Driver
DRAM ½ Pitch (nm)	80	70	65	57	50	36	25	18	
SOC new design cycle (months)	12	12	12	12	11	11	10	9	SOC
SOC logic Mtx per designer-year (10-person team)	3.3	4.3	5.4	7.4	10.6	24.6	73.4	113	SOC
SOC dynamic power reduction beyond scaling (X)	0.1	0.2	0.2	0.2	0.2	6	4.7	8.1	SOC
SOC standby power reduction beyond scaling (X)	2.4	3.4	5.1	6.4	8.73	18.8	44.4	232	SOC
% Test covered by BIST	25	30	35	40	45	60	75	90	MPU, SOC

Mtx—Million transistors

Manufacturable solutions exist, and are being optimized  
 Manufacturable solutions are known  
 Interim solutions are known  
 Manufacturable solutions are NOT known



## クロスカット TWG の課題

## モデリングとシミュレーション

ますます縮小していく特徴サイズに関連して、設計に挑戦する重要な問題のひとつは、製造パラメータの変動や、(例えば、チャネルドーピングといった)本質的な原子の性質に起因する設計に関連するパラメータの可変性の増加である。モデリングとシミュレーション技術は、適切な設計パラメータに対して、そのような可変性の量的なインパクトを評価することによって、この問題を緩和するのに役立つはずであり、そうならなければならない: 製造パラメータの変動と同様に統計的な変化は、パラメータ抽出やプロセス及びデバイスシミュレーションといった適切な装置を通して、能動素子や受動素子のサイズやスペーシング、トランジスタ特性、(更に信号遅延やひずみとして表現される)インターコネクトのカップリングといった設計パラメータのその結果生じる分類に変換されなければならない。ますます重要になってくるのは、不純物の原子的な性質である。それは、ある場合は、チャネル領域に平均に現れて、ドーピングと電気デバイスパラメータの莫大な相対的な変動を引き起こす、ほんの 1 つ、または、2、3 の不純物原子に起因することになる。とりわけ重要なものは、プロセス変動を増幅したり、滑らかにしたりするリソグラフィやエッチングといった、異なったものが次々発生するプロセスステップ間の相互作用である。シミュレーションは更に、寄生成分、遅延変化、ノイズ、及び信頼性問題のインパクトの評価に貢献するべきである。その中には動作時の熱問題も含まれている。全体のターゲットは、設計パラメータを使用される技術やデバイスアーキテクチャと更に密接に連結することである。それには、特にそれらのプロセスによって誘発された変化を含んでいなければならない。このことは、設計者が適切な安全係数(レイアウトの中で変化するかもしれない)を選択するのを助けるために必須である。シミュレーションのみが提供し得る付加価値は、比較的少ない時間と小さいコストで、広い変数空間を自動的に調査・探索できることである。

## 付録

### ばらつきのモデリングとロードマップ

ばらつきは、多くの DFM の困難な技術課題の原因と目されているため、ロードマップにするためのシステムティックな方法が求められているし、あるいは、また、ばらつきの望ましいトレンドが設計技術全体のロードマップの重要な部分になる。それは、業界がばらつきを低減すべきか、ないしは、設計生産性を改善するかの提言を集成していくための、設計と製造の“共同(共通)ロードマップ”を可能にして行く。このようなばらつきのフレームワークへの要求は、業界関係者へのばらつき関連の情報への敏感さによってはっきりと示されている。設計のコミュニティでは、ばらつきを設計のパラメータの観点から見て行く必要があるので、ばらつきのフレームワークは、マルチレベルである必要がある。すなわち、設計の複数のアブストラクションのレベルをカバーする必要がある。

ばらつきロードマップのフレームワーク(VRF: Variability Roadmap Framework)は、Figure26 に示されるように設計 TWG が開発しつつある。このフレームワークでは、以下の 3 つのアブストラクションのレベルが考慮されている：

- \* **回路/チップレベル** - これは設計者に最も関連するアブストラクションのレベルである。理想的には、対象回路のタイミングと消費電力のばらつきは、より下位のレベルのパラメータのばらつきに基づいて、このレベルでロードマップ化されるのが望ましい。
- \* **デバイスレベル** - 回路はデバイスで構成されているため、このデバイス - レベルのパラメータがロードマップ化される。例えば、しきい値電圧  $V_{th}$  やオフカレント  $I_{off}$  などがある。
- \* **物理レベル** - これは設計と製造のインタフェースに最も密接なレベルである。このレベルでは、CD や実効デバイス長(Le)、実際のドーピングレベル(Na)などのパラメータがロードマップ化される。これらのパラメータのばらつきは、リソグラフィ装置の解像度の限界、チャンネル内のドーパ量の正確な数値の制御ができないことなどを含む、上述の課題によって生じる。

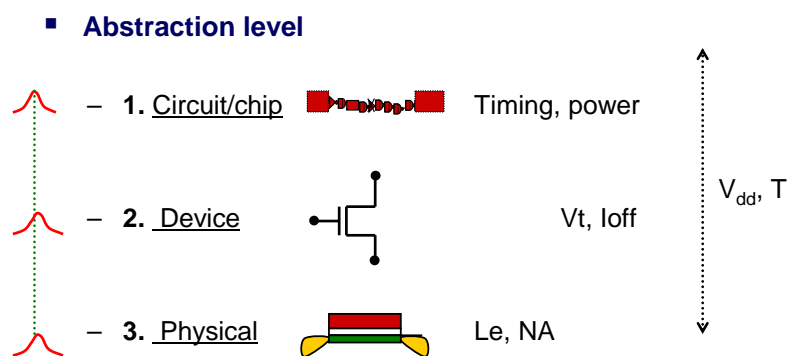


Figure 26 Possible Variability Abstraction Levels

このフレームワークでは、モデリングプロセスは以下の式で表わされる。

$$\Delta \text{ outputs} = \text{model}(\Delta \text{ inputs}) \quad [1]$$

公表されている近似モデルに基づいて、このモデルは単純化されたゲート+配線の回路小片をベースにしている、すなわちパラメータはゲートに関連するものと、配線に関連するものに分解されている。性能は今日のモ

デルではトータルの遅延としてモデル化されているが、これはゲート遅延と配線遅延に分解される。遅延のばらつきは、モンテカルロ式の入力パラメータのシミュレートされた分布に由来する、個々の遅延値の計算された統計的な分布のデルタとしてモデル化される。モデルの入力として、ゲートチャンネル長と幅、酸化膜厚、配線幅、配線長、配線層厚、ILD厚、配線シート抵抗が含まれる。

他の適用の中では、このフレームワークは回路性能のばらつきトレンドにも使用できるだろう。例えば、予測遅延ばらつきは、2つのシナリオで見積もられる：10%の要求CD(実際のチャンネル長)ばらつき、ないしは20%の要求CDばらつきである。この例では、モデルは、回路性能ばらつきがCDトレラント要求に対して著しく変わるようには見えないことを指し示している。つまり、要求が緩和できる可能性を示している。

## DT コストと価値

Figure 27 に示すように、複雑な集積回路を開発して市場に出すためのコストは、多くの異なるファクタによって影響を受ける。各ファクタは固定または変動するコスト要素によって表される。固定費が販売数量に依存しない一方で、変動費は販売数量によって増加する。製品開発は、エレクトロニクス製品のバリューチェーンの基本的な部分であり、一般的に販売数量を横断して広げられる固定費ファクタと見なされる。この議論の目的のために、設計コストは、直接的な製品開発の R&D コストとそれに関連するオーバーヘッド(経費)を加えたものと定義する(Figure 27 を参照)。不幸にも、systems-on-a-chip(SOC)の絶えず増加している複雑さは、設計コストとチップ単価をコントロール不能にしている。ますます競争が激しくなる環境と結合されてコストが上昇する中で、収益の確保が困難になってきている。これは、製品開発費が製品のライフサイクルの中で先行投資の状態になり、一方、売上げのかなりの部分が1年以上遅れで発生するという事実によって悪化させられる(キャッシュフローの悪化)。次の分析は、継続的な設計技術革新のパイプラインがなければ、設計コスト(したがって、製品開発費も)をかけることが急速に不可能な状態になるであろう、さもなければ、設計自体があまり価値のないものにさせられるであろうことを示唆する。

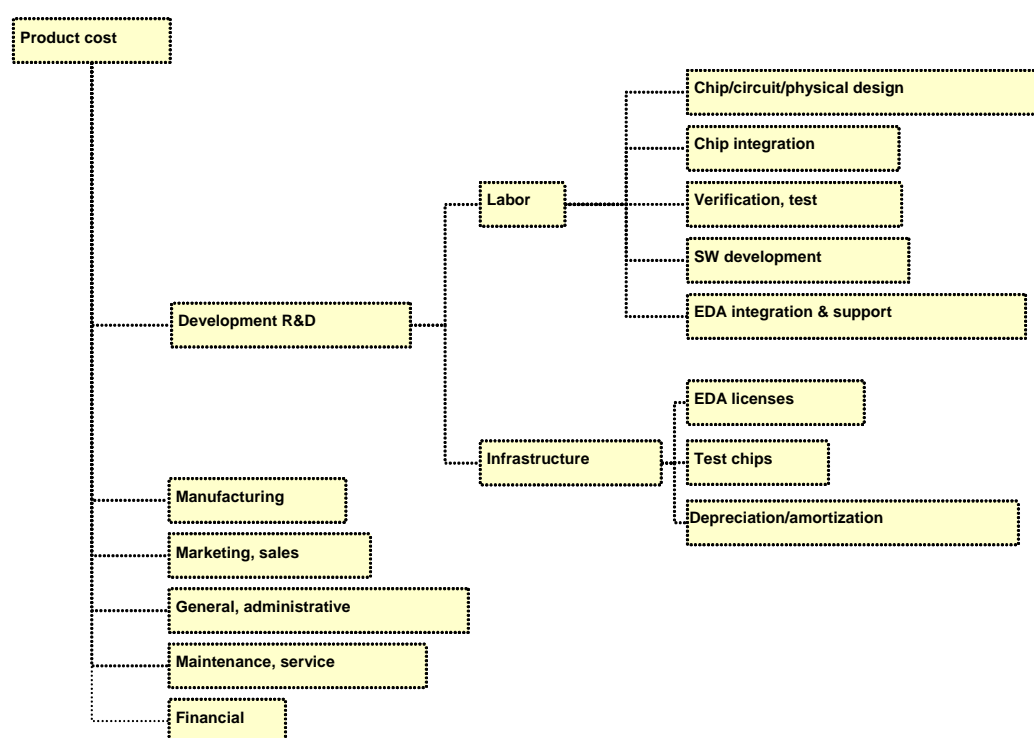


Figure 27 Simplified Electronics Product Development Cost Model

Figure 27 において、ボールド体の項目は設計コストと見なされる(機会コストとロスコストは含まれない)。この図は、製品開発費はおおよそ直接的な労働コストとインフラストラクチャコストに分解されることを示している。労働コストは以下の項目を含む。チップ、回路、及び、レイアウト/物理設計;チップインテグレーション;検証(ベリフィケーション)とテスト;ソフトウェア開発;EDA インテグレーション;ソフトウェアと技術サポート。一方、インフラストラクチャコストは、設計用のソフトウェアライセンス(ソフトウェア開発環境を含む)、テストチップ・インフラストラクチャと減価償却費を含む。これらのコストは、直接費に、(一般経費や管理経費を含む)割当てられた“オーバーヘッド”要素を加えたものとして表される。DT の半導体製品の収益性への極めて重要な貢献は、これらの各々のコスト要素に対する DT 革新のインパクトを列挙して、分析することによって理解される。

## 労働コスト

労働コスト要素は、おおよそ、労働単位コスト(エンジニアの給料(年俸)に関連)と設計の複雑さ(チップ内の各種機能やトランジスタ数に関連)、及び設計生産性(平均的なエンジニアが1年で十分に設計し得るデザインの複雑さに関連)に比例する:

$$DesignLaborCost = \frac{LaborUnitCost \times DesignComplexity}{Designer Productivity}$$

DT革新が設計生産性を増大させたので、それらの最も強い効果は労働コスト要素にある。DT革新の労働コストへのインパクトを測定するために、ITRS Design ITWGはガートナー/データクエストに、設計生産性を測定し、かつ、メジャーなDT革新による生産性の改善状況を計測するように依頼した。1990年(いわゆる「RTLメソッド」が起こった年)の設計生産性は、4Kゲート(=16Kトランジスタ)/年であった。その後の改善項目はTable 21に示されており、その中でのグレーの項目は、現在進行中、もしくは、将来のDT革新項目を表している。その表からわかるように、設計生産性(設計者1人年当たりのロジックゲート数として測定される)は1990年から2005年までの間に、年率で平均39.6%増加したことになる。特に、百万ゲートを設計する設計者数(生産性の逆であるが)は、1990年の250人から2005年には8人に削減された。しかし、労働単位コストは、1990年以来一定の値を維持しなかった。それは、GTXモデルによると、技術者労働コスト(1990年では、給料やその他の経費で181,568ドルと見なされる)が年率5%という歴史的な上昇率で増加したためである。この仮定は、設計のグローバル化の傾向を考慮して、次のロードマップ改訂では再検討されるかもしれない。GTXモデルは、電子設計章の部分として利用できる。

## インフラストラクチャコスト

エンジニアあたりのEDA ツールコストの上昇率は、年率3.9%と見積もることができる(1990年のエンジニアあたりのコストを\$99,301としてスタート)。そのためトータルのインフラストラクチャコストは、EDA ツールコスト×人年で求められる:

$$EDAInfrastructureCost = \frac{EDAUnitCost \times DesignComplexity}{Designer Productivity}$$

このコストは労働コストと関連付けた。他のインフラストラクチャコストは、現在のモデルにおいてはオーバーヘッドとして含まれると推測される。平均労働単位コストがEDAインフラストラクチャコストより速く増加することになったので、製品開発費における労働コストの割合が増加している。



Table 21 Design Technology Improvements and Impact on Designer Productivity

<i>DT Improvement</i>	<i>Year</i>	<i>Productivity Delta</i>	<i>Productivity (Gates/Design-Year)</i>	<i>Cost of Component Affected</i>	<i>Description of Improvement</i>
None	1990		4K		
In-house place and route	1993	+38.9%	5.55K	PD Integration	Automated block placement and routing.
Engineer	1995	+63.6%	9.09K	Chip/circuit/PD Verification	Engineer than can pursue all required tasks to complete a design block, from RTL to GDSII.
Reuse—small blocks	1997	+340%	40K	Circuit/PD Verification	Blocks from 2,500–74,999 gates.
Reuse—large blocks	1999	+38.9%	56K	Chip/circuit/PD Integration Verification	Blocks from 75,000–1M gates.
IC implementation suite	2001	+63.6%	91K	Chip/circuit/PD Integration EDA support	Tightly integrated tool set that goes from RTL synthesis to GDSII through IC place and route.
RTL functional verification tool suite	2003	+37.5%	125K	SW development Verification	RTL verification tool (“cockpit”) that takes an ES-level description and partitions it into verifiable blocks, then executes verification tools on the blocks, while tracking and reporting code coverage.
Electronic system-level (ES-level) methodology	2005	+60%	200K	SW development Verification	Level above RTL, including both HW and SW design. It consists of a behavioral (where the system function has not been partitioned) and an architectural level (where HW and SW are identified and handed off to design teams).
Very large block reuse	2007	+200%	600K	Chip/circuit/PD Verification	Blocks >1M gates; intellectual-property cores
Homogeneous parallel processing	2009	+100–200%	1200K	Chip/circuit/PD Design and Verification	Many identical cores provide specialized processing around a main processor, which allows for performance, power efficiency, and high reuse
Intelligent test bench	2011	37.5%	2400K	Chip/circuit/PD Verification	Like RTL verification tool suite, but also with automation of the Verification Partitioning step.
Concurrent software compiler	2013	60%	3300K	Chip and Electronic System Design and Verification	Enables compilation and SW development in highly parallel processing SOCs
Heterogeneous massive parallel processing	2015	+100–200%	5278K	System Electronic Design and Verification	Each of the specialized cores around the main processor is not identical from the programming and implementation standpoint
System-level DA and executable specification	2017-19	+100–200%	10557K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies.
Total		+264,000%			

## 概算のトータル設計コスト

Figure 16 は、システムドライバ章で定義したドライバであるpower-efficient System-on-Chip ( SOC-PE ) に関し、設計コストにおけるDT革新のインパクトを示している。SOC-PEは2005年で6.5M論理ゲートを搭載していることからSOCの設計コスト(設計者とツールの費用の合計で)は殆ど\$20Mとなる。1993年から2005年の間に実現した6つのメジャーなDT革新なしでは、2003年の同じSOCのための設計コストは、およそ\$900Mとなる。更に、日本の半導体技術ロードマップWG1 ( STRJ-WG1 )によって開発され、2001年度版のITRSのシステムドライバ章で引用されたもう一つの設計生産性指標を用いるならば、このギャップは、更に大きくなる。本指標では、1999年の新規(再利用)ロジックの生産性は360K ( 720K ) ゲート/人年となり、データクエストの見積りの6倍(12倍)である。<sup>14</sup>

---

<sup>14</sup> 差異は、マーケットセグメントの相違と、回路のカスタマイゼーションレベルの違い、データパス回路のパーセンテージ、及び、メソドロジの相違が原因であると考えられている。