

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**を可能にします。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**で構成されます。設計技術は電子機器システム設計者の設計構想と設計目標を、製造可能で試験可能な形式に変換する技術です。設計技術の役割は、製造能力を完全に発揮させるコスト効率の良い設計により、半導体産業の利益および成長を実現することです。

2001年版ITRSでは、設計ITWG(International Technology Working Group)が、Overall Roadmap Technology 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、システムドライバー章を新設しました。設計コストおよび設計生産性の新しい分析が2001年版ITRSの設計技術章に盛り込まれました。設計技術の課題と設計者からの改善要求はシステムドライバー章に記載されました。設計技術章の読者には2000年版ITRS設計技術章を参照することを推奨します。

2001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す。* コストは、設計対象をMPU(Micro Processing Unit),DSP(Digital Signal Processor)などのプログラム可能な商品上で、あるいは新規開発するIC上に、ソフトウェアまたはハードウェアとして実現するための鍵となります。設計開発費用(NRE(Non-recurring Expenses))はマスク、プローブカード費用を含めると100万ドルに達します。さらに設計NREに、設計欠陥によるリスピ(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します。技術革新のスピードアップは製品ライフサイクルを短くし、time-to-marketを重要な目的にします。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数か月または数年を要します。

半導体生産投資の減価償却がなければ、投資は高いリスクを持っていると思われています。確かに「Fill the fab」が設計技術のスローガンでした。また、**設計生産性ギャップ(設計クライシス)**が増大することは、2000年版ITRSで提起されています。1チップで利用可能なトランジスター数は、それらを有効活用する設計能力より速く増大します。プロセス技術への投資は設計技術への投資に比較して圧倒的に多大でした。

(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ます。図15は、システムドライバー章に記載された低消費電力SoC PDA(Personal Digital Assistant)の設計コストを示します。2001年におよそ1500万ドルの設計コストが、1993年と2001年の間に設計技術の改善が行われなかった場合には3億4200万ドルに達していたことが分かります。(分析の詳細は付録を参照してください)

(悪いニュース)ソフトウェアは組み込みシステム開発費用の80%を占めます。またLSIテストコストは製造コストに比べて指数関数的に増大しました。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2倍必要です。*2001年の現在、われわれは上記のような多数の設計技術課題に直面しています。*

本章では、最初に設計技術における**複雑度の課題(シリコン複雑度およびシステム複雑度)**について考察します。その後、特定の**設計手法**が設計技術の今後の設計システムおよび機能にどのように結びつくかを述べます。本章の大部分は設計技術領域における、従来からの観点で詳細に技術課

題を分析します。即ち、設計工程、システムレベル設計、論理/回路/物理設計、設計検証および設計テストです。これらの技術課題は前章のシステムドライバー(MPU、Analog Mixed Signal、SoC)に適用されます。序文では、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し、即ち、設計生産性、消費電力、異なるプロセス技術の混載、寄生効果による干渉(Signal Integrity)、および故障を起こさない高信頼設計です。とりわけこれらの技術課題の詳細については、EDA(Electric Design Automation)業界のマネジメント、開発部門および大学研究機関によって対応可能なレベルにあります。

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ります。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、精度を実現するかを明示します。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、いかに最適化するか、が主要テーマとなります。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テスト時間のような多数のトレードオフが求められます。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します。さらに、設計技術ロードマップはITRSテクノロジーロードへの関連性は高くありません。ITRSテクノロジーロード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を改善します。

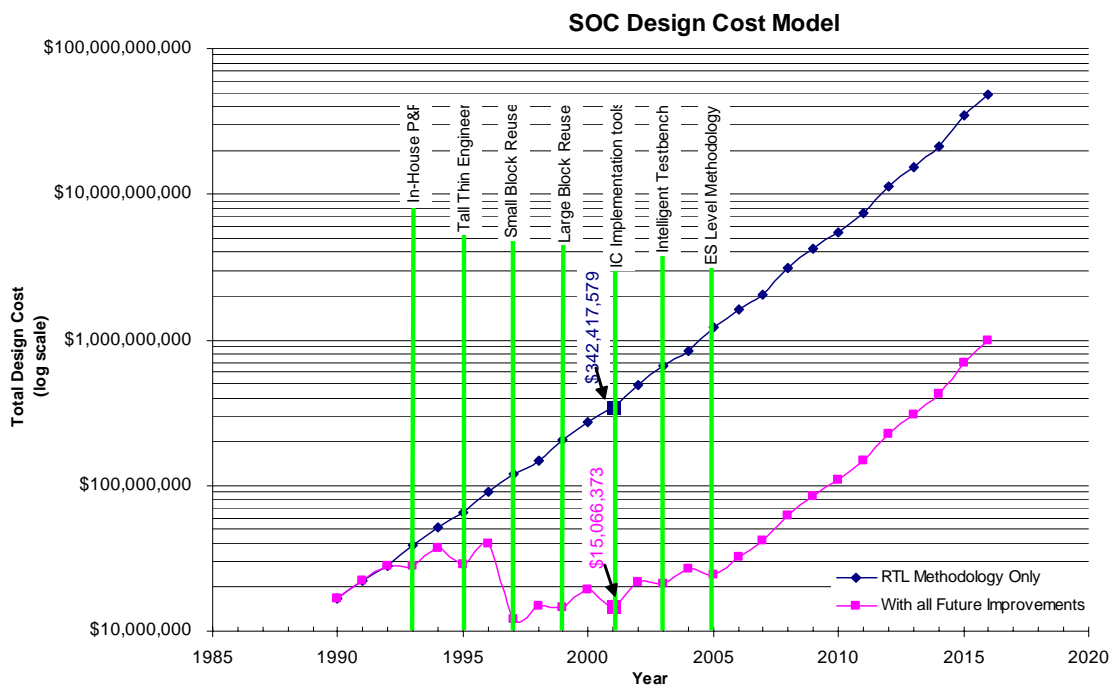


図 15 Impact of Design Technology on System Implementation Cost

複雑度による課題

設計技術は2種類の複雑度に直面します。- シリコン複雑度およびシステム複雑度です。これらはITRS製造技術ロードマップに起因します。

シリコン複雑度は、プロセススケールリングおよび新しい材料、あるいは新たなデバイス構造や配線構造の導入によりもたらされます。以前には無視できた現象が設計の正確さおよび価値に大きな影響

を及ぼします。

- * デバイスの寄生効果および電源電圧/しきい値電圧の理想モデルから外れたスケーリング (電流リーク、パワーマネージメント、回路/デバイス革新、電源供給)
- * 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)
- * 製造設備の性能限界 (統計的プロセスモデリング、ライブラリーキャラクタライズ)
- * デバイス特性を決定付けるグローバル配線のスケーリング (ブロック間通信、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、フォールトトレラント)
- * 製造の複雑度 (レティクルとマスクの描画および検証、NRE コスト)、 さらに
- * プロセスばらつき(ライブラリー特性抽出、アナログとデジタル回路性能、エラートレラントな設計、レイアウト再利用、信頼性が高く見積り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危うくします: (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により実現不可能になります。(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ます。そして、(3)100%正常動作するトランジスタと配線で構成されるチップは法外に高価になります。利用可能な実装設計の手法(たとえばプログラブルなカスタム LSI)では、簡単に 4 桁も性能面での低下を招きます (たとえば GOPS/mW)。また、不適切な設計マージン、設計抽象化あるいは設計手法の誤った選択は目標特性を達成できません。これらのことから、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て、個別の設計技術(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)を統合することが重要となります。

システム複雑度はプロセス微細化によるトランジスタ数の指数関数的な増加と、低コスト化および Time to market 短縮要求によって起こります。この課題は設計生産性ギャップと同意語です。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 SoC 設計に伴って発生します。仕様設計および設計検証は特に大きな課題となります。設計対象の商品価値、品質およびコストのすべてにおけるトレードオフが必要となります。(単純化した例:クロック周波数に対する「ムーアの法則」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します)

システム複雑度による設計技術課題は次のものが含まれます:

- * 再利用(階層設計、特にアナログ/ミックス・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、検証性を考慮した設計、混載 SoC のための検証再利用、システムレベルおよびソフトウェア検証、アナログ/ミックス・シグナルの検証および新構造デバイスの検証、セルフ・テストおよびノイズ/ディレイテスト、LSI テスターのタイミング制約、テスト再利用)
- * コストを重視した設計最適化(製造コストのモデリングと分析、品質指標、チップサイズ-パッケージ-システムレベル全体での最適化に、フォールトトレラント最適化、テストビリティなど)
- * 組込みソフトウェア設計(見積り可能なプラットフォームに基づいた設計手法、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 (多数の回路方式への展開、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(チームサイズおよび地理的な分散、データ管理、協力的な設計支援、「システム化された設計」サプライチェーンマネジメント、指標化された連続的なプロセス改良)

設計手法の教訓

シリコンとシステム複雑度は、*指数関数的に設計プロセスの複雑度を増加させます*。設計技術のための多くの Near-Term 目標を容易にあげることができます。たとえば：(1)より複雑な設計目標および設計制約の同時最適化と分析を実現して、(2)設計再利用および設計最適化のための製造コストを考慮し、(3)組込みソフトウェア設計および製造インターフェースなどです。より一般的な Long-Term ガイドラインを*設計手法の教訓*として紹介します。

設計手法は設計者および設計技術者の協力のもとに開発されています。それは設計制約を満たしながら、設計目標に「できるだけ接近した」設計を行うためのシーケンスです。設計手法は、設計技術(以下にそれぞれの設計技術領域で個別に議論されます)とは異なります。既知の設計手法はすべて、(1)トップダウン手法によりシステム仕様と設計制約を組み合わせ、(2)物理的な法則、設計および製造技術から生じる制約さらにはシステムコスト限界などをボトムアップに統合し決定されます。将来の設計手法および設計ツールは次の教訓を反映したものとなります。

(1) 設計資産の再利用：

設計生産性は以前に設計した回路ブロック(バスコントローラー、CPU、メモリサブシステムなどのような知的財産(IP)コア)を容易に利用できるフレームワークにより決定されます。標準の IP コア間のインタフェース、コミュニケーション・プロトコル合成、検証およびテストなどが再利用には重要です。IP コアの再利用は「Mix-and-match」プラットフォーム、あるいは特有の商品向け LSI の実装設計フローを実現します。

(2) 急速な発展：

設計技術は進化の軌道をたどります。(a)分析は設計検証に発展し、設計検証はテストへ発展します。(b)分析とシミュレーションはモデルおよび設計検証に発展します。そしてそれらは合成と最適化のための設計目標あるいは設計制約へ発展します。同様な傾向は、設計技術のボトムアップ的な共通化と、より高い設計抽象レベルへの研究開発および設計人材の投入を可能にし、標準化と相互動作(interoperability)を実現します。

(3) 反復の回避：

設計レベル間の反復は繰り返された設計変換および他の関連コスト増大を招き、設計プロセスの見積りおよび設計信頼度向上に有益ではありません。このような設計反復は、より広い最適解の探索のためにより高い設計抽象レベルで行われるべきです。

(4) 設計検証を問題防止に：

設計抽象度の低いレベルでの問題(たとえばクロストークによる遅延時間の不確実性)は、より高い設計抽象度レベルでの防止策により低減することが可能です。即ち、リピータ挿入やスルーレート制御、性的タイミング検証技術などがこれにあたります。

(5) 見積り精度の改善：

実行してから判断する方法では十分に設計生産性を改善することができません。実行前に高い精度で見積る技術こそが設計生産性を高めることを可能とします。高精度な設計見積り技術は、より抽象度の高い設計レベルで多数の候補解を考慮することを可能にします。関連した技術として、繰り返し最適化(incremental optimization)、継続的な設計改善(successive approximation)があげられます。

(6) 直交課題の分離：

システムレベルでは、アーキテクチャとビヘイビア、通信処理と演算処理などのように、独立で相互依存性のない課題を分離することが可能です。

(7) 範囲の拡張：

SoC への複数機能の混載と設計余裕の低下によって、設計技術と設計ツールは次のような発展が必要となります。(a) デジタル設計へのソフトウェア、アナログ、ミクストシグナル、RF(Radio Frequency)設計の統合。(b) ユーザインターフェイスの拡張。(c) LSI パッケージや基板、マスクやプロセス技術といった異なる設計階層でのモデリングとシミュレーション。

(8) 統合：

シリコン複雑度は本来異なる両機器の設計技術を結合して使用することを促します。即ち、(a) 合成技術と検証技術 (たとえば、静的タイミング解析バックプレーンを介しての)。(b) 論理/レイアウト/タイミング設計 (たとえば、RTL(Register Transfer Level)から GDS2 迄を同時設計する技術)。(c) 設計とテスト (たとえば、アナログ回路や高速ネットワーク回路用 BIST(Built-in Self-test)、シグナルインテグリティのためのテスト)。設計フロー内での個々プロセス間の連携設計の増加は、設計意図あるいは設計仮定を設計下流工程で利用可能とします。反対に設計予測、設計見積りモデルは上流工程での利用を可能にします。設計ツール間のインタオペラビリティ(相互運用)は統合環境を実現するための重要な技術です。統合設計技術は、従来の設計フローの各工程をそのまま統合することではありません。特に、この統合設計技術により製造プロセスと設計の連結させることが必要です。即ち製造レシピやデバイス構造、さらには製造装置の選択が設計でのトレードオフに大きな影響を与えるからです。プロセス技術の進展とともに、これら製造技術のモデル化は複雑さをますます設計への重要性は増大します。すべての設計ツールに対して、このような製造技術をモデル化する手法が必要とされています。さらに、設計とテスト技術間の統合は従来フローでのギャップ解消では解決できない大きな課題です。

教訓 1 および 2 は、より大きなシステムおよび高付加価値の技術に焦点を絞ることで設計改善が進むことを可能にします。教訓 3~6 は、トップダウン設計、繰り返しのない設計といった「Correct by construction」型のアプローチです。これとは反対に教訓 7 および 8 は「Construct by correction」型のアプローチです。ここでは設計繰り返しは起こりますが、最適化設計のフレームワークにより設計困難度を低減することが可能です。これらの観点を図 16 に示します。設計工程が各設計レベルごとに進行する、従来からのウォーターフォール型の設計フローと、論理、レイアウトおよび他のツールが一体となって機能する設計フローが示されています。全体として、図 16 に示された設計環境の実現が、設計技術コミュニティに求められています。

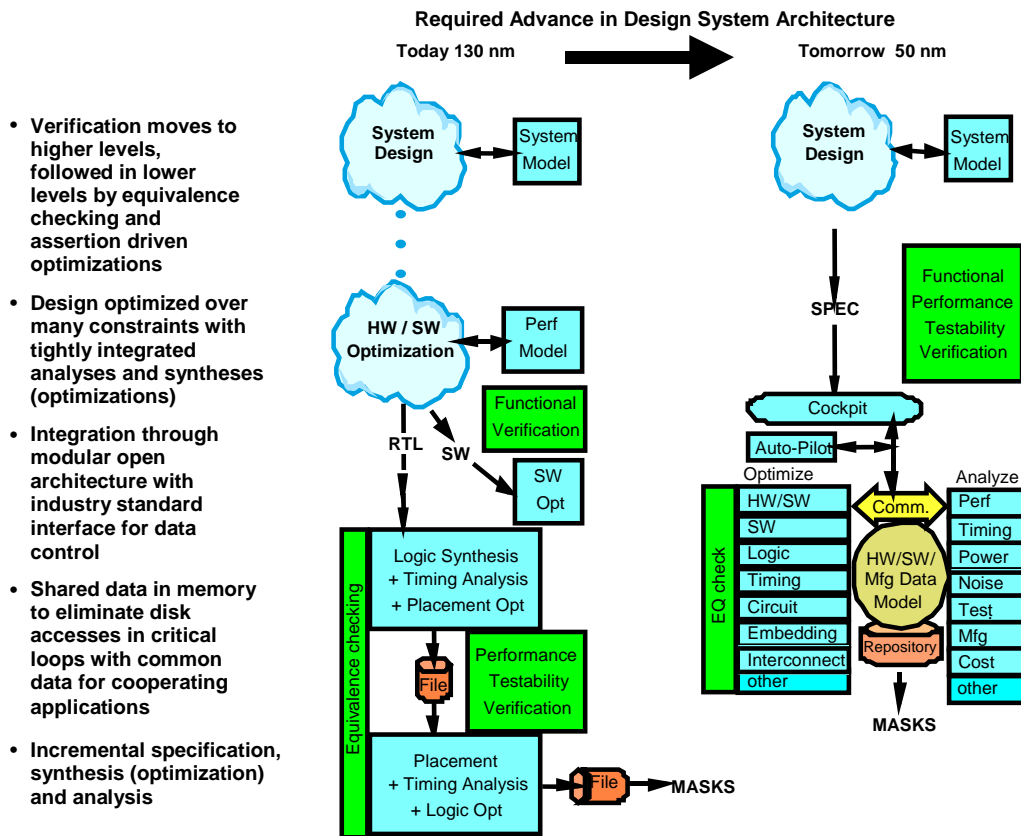


図 16 Required evolution of design system architecture

設計技術の挑戦

本章では、従来の設計技術の 5 つの分野における設計技術の挑戦およびその解決策について述べる。これらの分野は次のとおりである。:

- * デザインプロセス
- * システムレベル設計
- * 論理的、回路、物理設計
- * 設計検証
- * 設計テスト

設計技術の概観を図 17 に示す。詳細な説明に入る前に、クロスカット・チャレンジ(あるいは「モチーフ」) - 生産力、製造力、干渉およびエラートレラント - について述べる。クロスカットによる考慮は全体に渡り必要であり、解決策は設計技術の全分野で横断的に考慮しなければならない。システム複雑度と関連する第 1 のクロスカットの問題は最も重要である。第 2 は第 5 のクロスカットを通じて、必要であるが複雑度問題に比べて考慮の範囲は狭い。

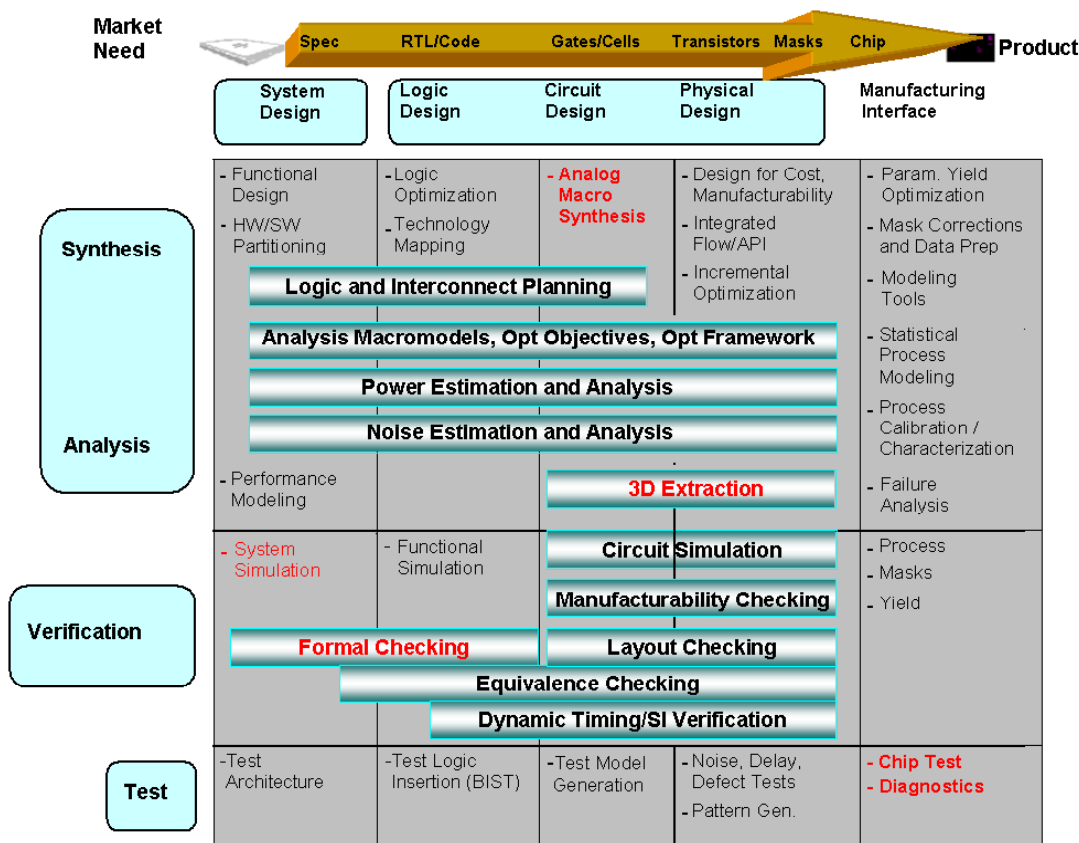


図 17 Landscape of Design Technology

- * **クロスカット・チャレンジ 1—生産力.** 指数関数的に設計コストを増加させることを回避するために、全体的な生産力は計らなければならない。そのため、設計、検証およびテストにおいて、再利用性をさらに図る必要がある。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(現状危機に達したボトルネックである);(2) 高位のシステムデザインを支援する信頼できかつ見積り可能なシリコン・インプリメンテーションの実現;(3) 組み込みソフトウェア設計の改善(SoC 生産力に対する最も重大な挑戦として出現した); (4) 特に MPU 混載については、種々のツール、組織での設計作業での生産力の改善;(5) アナログおよびミックス・シグナル(AMS(Analog Mixed Signal))の合成、検証およびテスト(SoC および AMS システム・ドライバーによって要求される)の自動化。これらの改良は、正規化された設計品質の測定基準が必要である。これらの要素は、設計 NRE コスト、製造 NRE コスト、製造により変動するコスト、半導体製品価値である。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、開発され改善される。新しい設計技術の市場への投入時期は、たとえば、標準化、相互運用および設計技術再利用プラットフォームによってさらに縮小されると考えられる。
- * **クロスカット・チャレンジ 2—パワー管理.** ここでは考慮の難しい CMOS デバイスのスケールアップ則、配線材料およびパッケージ技術を相互に関連させたパワー管理のチャレンジすべきことを示します。(1) Overall Roadmap Technology Characteristics およびシステム・ドライバー章中のロジック/メモリ抽出に基づいた HP(High Performance) MPU 消費電力の単純な外挿法は、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のチップパッケージ・パワー範囲を 25X によって超過する。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、SoC-LP PDA ドライバーは平均およびスタンバイ・パワーの考慮が必要と

なる。設計技術は、システム・ドライバー章の中で示されるパワー・マネージメントの対策に取り組みなければならない。(2) 供給電圧を減少させると、パワー密度は信頼度、変化する電流およびノイズを悪化させる。これらの傾向は、ストレスオンチップ、リソース(たとえばバンプ数および A&P 章の中の passivation 開始サイズ用のロードマップに照らして IR 低下をコントロールすること)、ATE(Automatic Test Equipment)性能および現在のバーン・インパラダイムなど相互に連結させる。(3) 高機能、低パワー、低スタンバイ・パワーデバイスの厳格な統合は、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・レベルで追加のパワー最適化を決定する間に同時に多くの自由度 - 多重 V_t 、および多重 Tox (単一のコアに共存する多重 V_{dd})を含んでいること - を考慮したパワー最適化が必要となる。

- * **クロスカット・チャレンジ 3—設計技術の統合。** 「Red bricks」 - 既知の解決策が存在しない技術必要条件 - は ITRS の全体にわたって考えなければならない。他方では、ITRS の一つの技術領域内で解決することが不可能なチャレンジ項目は、他の設計技術の適切な介在、協調で解決可能かもしれない。将来の技術ノードの実現可能性はこのような「Red bricks の共有」に拠る所が大きい。以下にいくつかの例を示す。(1) 試験デバイス設備コストおよび処理限界は、対応する自動的なテスト・パターン生成(ATPG(Automatic Test Pattern Generation))および内蔵のセルフ・テスト(BIST)技術に加えて、新しい故障モデル(たとえばクロス・トーク(パス遅延))の新たな採用によって解決するかもしれない。(2) システム・インプリメンテーション・コスト、実行検証および全面的な設計 TAT(Turn Around Time)の改善は、システムの設計技術と同様にチップ・パッケージ・ボードの最適化および分析によって実現されるかもしれない。(3) 製造ばらつきの下でのリソグラフィ(PIDS)での CD コントロールの必要性、フロントエンドの処理、配線技術などは新しい設計技術によって緩められるかもしれない(たとえば、ばらつきを考慮した回路設計、規則性を考慮した設計、構造的なタイミング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、より知的なインターフェースにより削減されるかもしれない。
- * **クロスカット・チャレンジ 4—干渉。** 論理、回路、物理設計設計における通常のシグナル・インテグリティメソッドは、100nm のノードでは現実的に解決可能な範囲に達する。これらのメソッドは長い配線へのリピーター挿入やスルー・レイトコントロール、インダクタンス管理のための消費電力/グラウンド分配設計などを含んでいる。ミックス・シグナル、RF コンポーネント、スケールアップおよび SoC インテグレーションはより柔軟でより有効なメソッドが必要である。無歪限界(特にロー・パワーデバイスおよびダイナミック回路では)を含む多数のインターコネクト、供給電圧 IR 低下およびグラウンドはね返り、デバイスオフ電流への熱衝撃、配線抵抗、基板カップリングなどが検討課題になる。設計技術の挑戦は、基本的には設計のすべてのレベルにおけるノイズと干渉のキャラクタライズ、モデリング、分析および評価方法を改善することである。
- * **クロスカット・チャレンジ 5—エラートラント。** 生産、検証、テストのコストを劇的に削減するためにデバイスや接続における 100%の正確さの要求を緩めることが必要である。このようなパラダイム・シフトは、いろいろな技術スケールアップの場合でも考慮することが必要である。以下いくつかの例を示す。(1) 100nm 未満での、厳しいインパクト・フィールドレベル製品信頼性を要求する Single-event upset(ソフトエラー)で、これはメモリだけでなくロジックも同様である。(2) 供給電圧が減少するとともに(指数関数的により長いバーンイン時間が必要)、現在の加速度試験(バーンイン)方法は実行不可能になる。;バーンインオープンの消費電力要求も増大する。(3) 原

子的なスケールでは実現可能なゲート酸化物のために、新しい「ソフトな」欠陥基準が必要となる。一般的にシステムの大規模化の結果、最終段階での機能テストは不可能になる。設計過程でのロバストネスを考慮した自動回路挿入は優先的に考慮すべき事項となる。例を上げると、冗長回路の自動検出、フォールトトレラントを考慮した回路再構成、自己修正、調整、および適用回路の開発、そしてソフトウェアによる方法などである。

デザインプロセス

関連する技術ニーズおよび特定のシステム・ドライバーを含むデザインプロセスの困難なチャレンジを表 13 に示す。これらは設計メソッドロジ、ツールおよび設計システム・アーキテクチャの総括的な必要条件である。

(1) デザインプロセスへの挑戦

1) 設計共有および再利用

今日の設計は、業界標準的なシステムや粗なインターフェースにより統合されたシステムにより実現される。大きな設計プロジェクトでは、設計技術を単一のベンダーに頼るのではなく、利用できるすべてのベンダーにより、要求される設計技術が満たされる。内部的に高度に開発された設計技術はマルチベンダー設計フローの価値を増大させる。大規模な設計は、種々の異なる会社や地理的に離れた設計チームにより開発される可能性があり、設計リソースや設計技術を共有し、再利用する必要が生じる。このため、以下のようにすべての設計技術を横断的に考慮し解決策を検討しなければならない。(a)異種の設計技術や設計間の設計記述と生産を共有する業界標準方法および(b)この方法の産業界での採用が必要である。設計技術データの再利用や相互運用が実現すると、全体の設計コストの 30%が削減できると推測される、(データクエスト、2001 年)。これらは、多数のプラットフォーム(reprogrammable、ASIC(Application Specific Integrated Circuit)など)およびアプリケーション(無線通信、インターネットワーキングなど)を有する分野において、統合コストの削減、高品質のインプリメンテーションを可能にする。

2) システムとシリコン複雑度の増加

システム複雑度に取り組むために、分散プラットフォームあるいは SMP(Symmetric Multiprocessing)プラットフォーム(たとえばチップ/パッケージ/ボードの協調設計、製造容易性、および電力/ノイズ管理の最適化用)上で、データ変換や冗長な処理を最小に抑え、合成の支援とアーキテクチャ分析ツールなどを統合する必要がある。基本のアルゴリズムは定量化することが可能である。その結果、指数関数的に増加するインスタンスサイズはハードウェアを計算する上で有効に解決できる。たとえば、「Construct by correction」、逐次近似法を用いたインクリメンタル分析および最適化は設計量に比例した実行時間で可能である。繰り返しの変更への対応、縮小されたサイズやピッチ、周波数の増加など対応した正確なモデル、ノイズ/干渉、消費電力への対応、信頼度などが製造プロセスに要求される。このようなキャラクタイズは、技術あるいはプロセスの管理の下で標準デバイス技術やプロセスの特別なモデルによって提供される。このモデルは使用する設計技術全般に横断的に必要となる。デザインプロセス全般における補足的な要求は、最適な物を作るための抽象度解析とマクロモデルの定義である。

3) コストドリブン SoC のタイムツーマーケット

急成長している半導体市場では、廉価、比較的low機能、早期の市場投入などの要求が多くを占める。これらの市場の製品は典型的に再利用期待の大きい SoC である。効率的な設計最適化のために抽象度の高いレベルでの再利用可能なコアが必要になる。このため確認/試行を支援するための新ツールが必要になる。再利用ベースの設計解の探索および設計最適化や再利用を保証する設計規則の開発および形式化と、このような規則を支援し実行するツールの開発が必要である。混載のファブリックおよび技術が単一 SoC または SIP(System in Package)に統合されるように、ハードウェア、ソフトウェア、デジタルそして恐らくアナログも統合され、メモリを含めて単一チップ上に実現できる設計フローが出現と考えられる。本章中の設計技術分野であるアナログおよびミックスシグナル設計は、設計生産力のボトルネックとなることは明らかである。従って、コストドリブンな SoC は、デジタル回路で実現した論理合成に匹敵する効率および質を備えたアナログ回路の合成技術が必要となる。長期的なニーズは、全体的にシステムインプリメンテーションコストを最小限にするサプライチェーンマネジメントおよびデザインプロセスと同様に、高位での機能検証、性能検証、製造可能性の確認である。適用される領域および個々の設計でカスタマイズされる設計アーキテクチャが相互運用可能なコンポーネントとなるための特定のプラットフォームが必要であり、そのために EDA のモデルも変わるかもしれない。これは、標準化されたインターフェースを持ちアーキテクチャ再構成可能な設計フローを構築するような EDA 産業が必要となることを意味している。

<i>DIFFICULT CHALLENGES</i> <i>≥ 65 nm / THROUGH 2007</i>	<i>SUMMARY OF ISSUES</i>
Design sharing and reuse	Geographically distributed and multi-company design projects Integration of multi-vendor and internal design technology (MPU, SoC) Standard information model for IC design data, with standard interface (access mechanism) adopted across tools, databases (MPU, SoC) Tool interoperability that minimizes data translation time and redundancy to reduce design cycle times (MPU, SoC) Reduction of integration cost
Increased system and silicon complexities	Device count, scaling, operating frequency, power and noise management (MPU, SoC) Incremental analysis and optimization capability for constraint-dominated design, with runtimes proportional to amount of design changed (MPU, SoC) Scalable design optimization algorithms Concurrent execution of design and analysis tools with appropriate objectives and abstractions (e.g., for power, noise/interference, package-die, and manufacturability optimizations) (MPU, SoC) Common device, wafer recipe and equipment characterizations, controlled by process owner and packaged for “immutable interpretation” by design and analysis tools
Time-to-market for cost-driven SoC	Common information models to support reuse and (cost-driven) design space exploration (SoC) Design rules and information models (e.g., abstracts) that assure reusability; design and validation tools to assure these rules for reusable design IP (SoC) DT integration for hardware/software, digital/analog, MEMS, memory, design tools (AMS, SoC) Synthesis of analog designs comparable to digital RTL-based synthesis (AMS, SoC)
Systematic improvement of design process and design productivity	Standard design process metrics, calibration and benchmarking
<i>ADDITIONAL DIFFICULT CHALLENGES</i> <i>< 65 nm / BEYOND 2007</i>	
Time-to-market for cost-driven SoC	Platform- and application- (and even design-) specific design flows via reusable, interoperable tools (SoC) Synthesis of mixed-technology designs (including analog) comparable to digital RTL-based synthesis (AMS, SoC) System cost minimization tools spanning from standardized process description to supply chain management (SoC) Higher-level verification of function, performance and manufacturability (SoC, MPU)
Systematic improvement of design process and design productivity	Design technology productivity analysis and optimization tools Predictable physical implementation flows, along with predictive models for such flows

表 13 *Design Process Difficult Challenges*

4) デザインプロセスおよび設計生産力の系統的な改良

電子システムの設計は、ある程度、科学ではなくアートである。しかしながら、設計技術が系統的に急速に進む場合、教訓的な「Measure, then improve」は適用されると考えられる。短期技術のニーズでは、継続的なデザインプロセスおよび設計生産性改善を開発するために、設計品質と見積りに関してのデザインプロセスの評価基準が必要になる。設計品質の評価基準は、設計技術により利用可能な製造する能力を評価することをサポートする。見積り評価基準は、ノイズ・フリーなアルゴリズム、ツールおよびフローにおいて、容認された解決基準に関し、見積り可能で安定しているかを、

数量的に評価することを可能にする。関連するニーズとしては、計測とベンチマーキング用インフラストラクチャについてもある。トップダウン設計および迅速な設計空間の探索をサポートするために、ツールスイートスポット(使用する部分)の評価ツール、インスタンス特有のツールランタイムパラメーター、および、ツールおよび設計インスタンスの両方のパラメーター化されたモデルに基づいた設計技術によるソリューションが必要である。長期には、プラットフォームまたはファブリックの見積りが可能なシリコン・インプリメンテーションフローが必要である。最後に補足的に述べると、革新のためには設計技術生産力の測定と改善を必要としている。

システムレベル設計

電子システム設計プロセスは、トップダウンと構造化された方法で、設計者にシリコン資源を割り当てて、開発することができるようにします。システムレベルでは、シリコン資源は抽象的な機能およびブロックで定義されます。設計目標はソフトウェア(高位の組み込みコードとアセンブリ言語、コンフィギュレーションデータなど)およびハードウェア(コア、ハードワイヤード回路、バス、リコンフィギュラブルなセル)を含んでいます。ビヘイビアとアーキテクチャは、アーキテクチャの2つのコンポーネントであるソフトウェアおよびハードウェアの、独立した設計自由度の度合です。アーキテクチャブロックの総計がシステムプラットフォームを定義している一方、ビヘイビアの総計はシステム機能を定義します。システム機能からシステムアーキテクチャへのプラットフォームマッピングは、システムレベル設計の中心であり、増大するシステムの複雑さや異なった成分(アーキテクチャ的か機能的か)によって、より困難になります。大規模なシステム設計を成功させるためには、あらかじめデザインされた IP ブロックおよび機能の広範囲な再利用に頼らざるを得ないと信じられています。しかしながら、再利用はまだシステム設計プロセスに浸透していません。IP ブロックを備えた設計はまだ完全には理解されていません。また、マッピングはほとんど手動のタスクのままです。

設計技術はメソドロジの側面と設計自動化の側面の両方を持っています。前者は、設計解の探索およびモデル改善のような要素をカバーします;後者はツールを定義します。また、アルゴリズムは設計方法をインプリメントする必要がありました。システムレベル設計では、急速にメソドロジの側面がツール側面より困難になってきています。将来のプロセス技術は、非常に大きなシステム複雑度を単一のダイの上で実現することを可能にするでしょう。設計コストを管理する上で、この可能性を利用するには、設計生産性の急激な向上 - 今日の50倍 - を必要とするでしょう。ばらつきと信頼性のようなシリコンの複雑さは、大いに信頼性の高く利用可能なシステムが、全く信頼性の低いデバイスと相互配線コンポーネントから構築されるという状況でこの生産性の改善が達成されなくてはならないことを意味します。主要な結果として生じている傾向は次のとおりです:

- * *HW(Hardware)および SW(Software)の領域の両方で、再利用かつプラットフォームに基づいた設計* 複雑なアプリケーション指向のアーキテクチャへ有機的に結合した、あらかじめ設計され、あらかじめ実証され、あらかじめキャラクタライズされた IP ブロックの再利用は、改善された設計生産性のために最も高い可能性を提示します。
- * *通信と同期のますます抑制するコスト* プロセスばらつきおよび消費電力は、大規模ICに対して、グローバルな同期を非実用的にします。チップを横切る信号の伝達は、1クロックサイクルではもはや達成されません。システム設計スタイルはますますネットワーキングと分散計算隠喩を認めるに違いありません、たとえば、通信では、構造が最初に設計され、それから機能ブロックが通信のバックボーンに統合されました。

- * **異種の統合** 多数の設計テクノロジーでインプリメントされたサブシステムの単一チップへの統合は、単純で信頼できて、見積り可能でなければなりません。
- * **組込み SW** HW プラットフォームが現れて、ますます定着されるにつれて、組込み SW は製品差別化の主要な媒体になるでしょう。その後、組込み SW はシステム設計の中で一番の市民権を得るものになるでしょう。：それは主要な焦点になります。

表 14 はシステムレベル設計のチャレンジおよび課題をリストしています。このセクションの残りで、これらをより詳細に述べます。

<i>DIFFICULT CHALLENGES</i> <i>≥ 65 nm / THROUGH 2007</i>	<i>SUMMARY OF ISSUES</i>
System complexity	Higher-level abstraction and specification (SoC) Dynamism and softness (SoC) System-level reuse (SoC, AMS) Design space exploration and system-level estimation (SoC, AMS, MPU) Efficiency of behavioral synthesis and software compilation (SoC) Automatic interface synthesis (SoC)
System power consumption	Energy-performance-flexibility tradeoffs (MPU, SoC) Novel data transfer and storage techniques (MPU, SoC)
Integration of heterogeneous technologies	Codesign (HW-SW, chip-package-board, fixed-reprogrammable) (SoC, MPU, AMS) Non-scalability of analog circuits; analog behavioral modeling and synthesis (AMS, SoC) Top-down implementation planning with diverse fabrics (digital, AMS, RF, MEMS, EO, SW) (MPU, SoC, AMS)
Embedded software	SW-SW codesign onto highly programmable platforms (SoC) System capture and abstraction (SoC) New automation from high level description to HW-SW implementations, including SW synthesis (SoC) Formal verification for SW (SoC) HW-SW coverification (SoC)
Links to verification, test and culture	Integration-oriented verification and test architectures (SoC) Divergent design practices and cultures (SoC, MPU, AMS)
<i>ADDITIONAL DIFFICULT CHALLENGES</i> <i>< 65 nm / BEYOND 2007</i>	
System complexity	Communication-centric design and network-based communications on chip (SoC, MPU) Design robustness (SoC)
System power consumption	Non-scaling of centrally organized architectures (MPU) Building large systems from heterogeneous SoCs (SoC)
Integration of heterogeneous technologies	Total system integration including new integrated technologies (e.g., MEMS, electro-optical, electro-chemical, electro-biological or organic) (SoC)

表 14 System-Level Design Difficult Challenges

(1) 短期(>65 nm)のシステムレベル設計チャレンジ

1) システム複雑度

システム複雑度および設計生産性へのチャレンジは下記技術ニーズを創ります。

- * **高位の抽象化と仕様** 設計抽象レベルは、HW に対する今日の RT レベルや組込み SW に対する C コードレベル(多くの場合プロセッサ依存)の上まで引き上げなければなりません。操作環境中の意図したシステム機能のモデル、および処理と通信に対するアーキテクチャ資源の両方のモデルは、複雑で多機能なシステムのための設計解の探索に必要です。使用メソッドおよび抽象レベルについて、産業界が一致して、複雑なモデル用の設計言語基盤(たとえば System C やその他 C++ 派生語のような)が現れてくることに、てこ入れしなければなりません。
- * **ダイナミズムと柔軟性** ダイナミズムとは、使用必要条件下(たとえばサービスのマルチメディア品質)にお

いて、実行時間内でシステムを適合させることができるシステム・プロパティです。新しい抽象的概念は、機能とアーキテクチャのそのようなランタイム修正のために必要になります。柔軟性とはシステムを修正したり、再プログラムしたりできるシステム・プロパティです。柔軟性は、多数の抽象レベル - プロセス、ISA、マイクロアーキテクチャおよびインプリメンテーション - に適用されるでしょう。けれども、設計者がチップのプログラム可能な特徴を識別し選択することができるメソッドロジは、まだ存在しません。アーキテクチャ上の革新は、SoC の機会および制約に適合して、かつそのような動的なビヘイビア/アーキテクチャを支援するために必要です。

- * **システムレベル再利用** いくつかの進歩が RT レベルおよびレイアウトレベルの設計再利用を成し遂げましたが、再利用のメソッドロジと関連する設計ツール(たとえば IP 品質および適応性の評価)は、まだ初期段階にあります。設計生産性のインパクトは、プラットフォームに基づいた設計のような方法によって、複雑な HW-SW アーキテクチャの再利用を必要とします。ソフトウェア再利用に対しては、たとえば、「標準」ソフトウェアを実行するが、異なるコストパフォーマンストレードオフあるいは計算上の領域に特定化されるプロセッサに対する設計メソッドロジを要求します。
- * **設計解の探索およびシステムレベル評価** 統合可能なインプリメンテーション製品の範囲の増加は、機能アーキテクチャのマッピングを最適化するための新しいツールを要求します。そのような最適化の基準には電力、面積、処理能力などを含んでいます。評価技術は、基本的なものであるが、まだ、トップダウン設計解の探索における未開発な副産物です。設計プロセスの詳細を省いて、評価結果が設計基準の最終的な値の正確な予測を可能にします。そのような能力は、特定の適用空間の統計モデリングと同様にますます抽象化(たとえば、物理的、RTL とシステムレベルにおいてのタイミング推定)の多数のレベルにまたがるに違いありません。
- * **効率的なビヘイビア合成および SW コンパイル** 最終的な再利用解決策を補完するために、新しい合成方法が必要とされます。機能からアーキテクチャへの自動マッピング - ハードウェアに対するビヘイビア合成とソフトウェアに対するコンパイル - は、合成結果(たとえば、コンパイラによるプロセッサの特徴や並列化の不適当な利用)の低い品質のために今日適用が制限されてきました。ニーズは、混合制御およびデータ・フローに対する受容可能な合成品質、および種々様々の命令セット・アーキテクチャ(コントロール処理および DSP)および設計領域に関する高品質な結果を自動的に達成する、リターゲッタブルな SW コンパイラを含んでいます。
- * **自動的なインターフェース合成** システム制約をよりよく一致させるためには、HW-HW と HW-SW および SW-SW の間のインターフェースは、手設計するか、パラメータ化されたライブラリから取り出す代わりに合成(あるいは標準化)されるに違いありません。これは通信中心の設計アプローチが現れることによる自然な副産物です。

2) システム消費電力

いくつかの現在あるプロセッサ・アーキテクチャ・パラダイム(常にオンしている単体のプロセッサ; 高性能の VLIM および推論的な技術)は、より大規模に混載された並列化によって近いうちに崩れるでしょう。エネルギー、性能および柔軟性の効果的なトレードオフは、システムレベルにおいて予測可能になるに違いないし、動的な電力制御; 柔軟なブロック停止と再起動; 動的な可変オンチップ電圧供給; 標準で制御されるパワーマネジメント; 電力を効率的に使用するリコンフィギュラブルロジックのような技術を要求するでしょう。システムレベルにおいて、短期での消費電力に対する主要な手段は、データ転送とメモリ・アーキテクチャと階層化です。

3) 異種テクノロジーの統合

利用可能な構成技術の範囲にまたがって有効な設計選択を行うことは、異種のモデリングと協調設計への新しいアプローチを要求します。

- * **協調設計** 分割と協調設計のためのDT(Design Technology) - HW - SW、アナログ - デジタル、固定 - 再プログラム、あるいはチップ - パッケージ(- ボード)に関わらず - は、システム開発プロセス全体にわたるコストと危険要因にますます気づくに違いありません。アナログ - デジタル境界間のトレードオフと分割は、単一の実行と分析環境内でのシステムの部分をすべてシミュレートする能力と同様に、互換性をもつ仕様と機能的なモデル化の抽象的概念を要求します。より一般に、スタンドアロンの実体としてではなく、設計最適化は、ますますそのシステム・コンテキスト中にデバイスを考慮しなければなりません。
- * **アナログ回路の非スケラビリティ;アナログのビヘイビアモデリングと合成.** アナログは引き続き回路設計をスケリングできず、設計コストの増加の根拠となっています。自動アナログ回路合成および最適化は、全面的なシステム機能の分析を可能にする言語レベルでモデル化するメソッドと他の統合化されたテクノロジー(デジタル、ソフトウェア)とのインターフェースが必要です。問題はシミュレーションの中で、広大に変わる時間の尺度(数百 GHz までのアナログ周波数対数 GHz までのデジタル周波数)に対処すること;混載のテストベンチの作成およびカバレッジの保証;系統的なトップダウン制約伝播の達成;機能と構造の表現の混合を含んでいます。
- * **種々の製品に対するトップダウンインプリメンテーションプランニング.** より多くの統合可能な製品に対して、システムレベルの目標および制約を製品の詳細(回路位相、インターフェースおよび概略レイアウト)とリンクさせる単一の階層的な混合技術プランニング環境の必要性が増してきます。

4) 組み込みソフトウェア(ESW)

再利用かつ SW 中心の SoC 設計は、(a)ハードウェア中心と、ハードウェアに依存したソフトウェア中心の設計者のチームによるプログラム可能なプラットフォームの生成、および(b)アプリケーションおよびソフトウェア中心の設計者のチームによる派生製品へのプラットフォームの使用へますます分割されます。システムプラットフォーム(これをアプリケーション・プログラマーズ・インターフェース、あるいは API と見なすことができます)は、2 つの設計チーム間に媒介し、アプリケーション領域の要求と根本的なシリコン性能との間の中間媒体です。この背景については、次の重要な問題を含んでいます:

- * **おおいにプログラム可能なプラットフォーム上への SW-SW 協調設計.** おおいにプログラム可能な、アプリケーション領域に特定されるプラットフォームは、派生的な設計に対して、多数のインプリメンテーション製品(恐らく構成可能な制御プロセッサあるいは DSP 上で走る SW、機能ブロックを多重モードでインプリメントされた HW、リコンフィギュラブルロジックなど)およびプログラム可能な相互に接続する資源(SW-SW、HW-HW および HW-SW)を提供します。そのような多数のレベル(ソフトウェア、アセンブリコード、ビットストリームなど)での柔軟性が、「SW-SW の協調設計」 - システムのタイミングおよび電力の制約を満たすプログラム可能な処理と通信資源上への迅速で信頼できて、予測可能な機能のマッピング - には必要です。
- * **システム捕獲および抽象化.** システムは、個々の事象、静的および動的なデータ・フローあるいは連続時間のような様々な計算モデル(MOC)内で表されます。設計者は、各領域あるいは MOC に対して、インプリメンテーションに依存しない実行可能な「自然な」表記法を使って、システムの機能的および通信の要求をモデル化できなければなりません。これらのモデルは、実行可能なシステム仕様書を正確に構成しなければなりません。

SoC 混載の追加的な技術要件は、高位の記述からの HW-SW インプリメンテーション、SW のためのフォーマル

検証および HW-SW 協調検証までの新しい自動化(SW の合成を含んで)を含んでいます。

5) 検証

テストおよび文化へのリンク. ブロック生成から SoC プラットフォーム上でのブロックの再利用へ設計焦点を変えることは、検証、テストおよび DT の「文化的な」面に対して、次の意味合いを持ちます。

- * *統合指向の検証およびテスト・アーキテクチャ.* 設計統合は、階層的なテストベンチの統合とマイグレーションによる検証統合によって並列化されなければなりません。統合化によって、システムレベルのブロックモデルは、合成および検証で使用するができる抽象的な処理レベルの検証モデルの基礎を形成します。形式的に指定された抽象的な仕様記法は、フォーマル HW 検証からのプロパティチェックおよび SW のモデルへの適用を可能にするでしょう;フォーマル SW 検証における研究も進んでいます。統合指向のアーキテクチャは、同様にテストのために必要になります。その結果、全面的なシステムテストへ特定のブロック用の再利用可能なテスト素子を統合することができます。テスト適用(試験デバイス時間、また ATE 速度制限)のコストは、ますます SW ドリブなオンチップ自己テスト技術を動機づけるでしょう;システム設計とテスト・アーキテクチャ設計の間の依存性は初期のシステム概念から計画されなければなりません。
- * *分岐する設計慣習および文化.* システムレベル設計の軌道 - 常により大きい抽象化とアプリケーション特定性に向かって - は、IC インプリメンテーション(それは詳細な物理と製造原価の現実にますます気づくに違いない)の軌道から逸脱します。このインプリメンテーション・ギャップとその解決策はいくつかの面を持っています。(a) 関連づけられた文化的ギャップは、システム設計者から IC インプリメンテーション設計者へ緊密にパスを追跡する設計抽象レベルの注意深くかつ正確な定義によって言及されなければなりません。(b) システム設計者に対して、自動マッピング、予測可能なインプリメンテーション・フローおよび正確なコスト推定を与える IC「シリコン・インプリメンテーション・プラットフォーム」が現れなくてはなりません。そのような接続ポイント(たとえば中央設計環境あるいは「コックピット」)の DT 支援なしでは、システムレベル設計および IC インプリメンテーションの間のリンクは、その場限りの、設計非効率な資源のままでしょう。

(2) 長期(<65 nm)のシステムレベル設計チャレンジ

1) システム複雑度

長期では、システムの複雑さが、計算上および全面的なシステムロバストネスにおいて通信に対する新しい焦点を導きます。

- * *通信中心の設計.* 65nm およびそれ以降では、通信アーキテクチャとオンチップの機能処理ユニットのプロトコルが、今日のアプローチから重要な変更を要求するでしょう。1 クロックサイクル内で、あるいは消費電力の面で実現可能な方法で、大規模チップを横切る信号を動かしたり、同じクロック周波数で制御とデータ・フローの処理を実行することが不可能になると、考え得る結果は非同期(あるいは、全体的に非同期でかつローカルに同期(GALS))設計スタイルへの変更です。そのような形態では、Self-timed な機能性の島がネットワーク指向のプロトコルによって通信します。これは、システムレベル電力最適化案(たとえば最大チップ周波数で大きなクロックネットをスイッチングすることの回避)の要求と一致します。通信中心の展望は、(a)アルゴリズムが、高位の時々のコントロール事象によって制御されたデータ変換を同時に同調した設計に変更されなければならないことおよび(b)機能的に分割されたタスクあるいは機能間の通信が設計プロセスの中心に残るということを暗示します。
- * *設計ロバストネス.* ネットワーク指向のパラダイムに続いて、歩留り問題、非同期設計スタイルおよび通信中心の設計は、SoC 設計が従来の IC 設計の慣行よりも大規模通信網の生成により似てくるといことを暗示します。そのようなネットワークでは、通信が潜在的に損失があると仮定することができます。また、

ノードは通信を失敗するかもしれません。しかし、ネットワークは、正確さとスループットに対して、システム要件を達成しなければなりません。そのようなインプリメンテーション製品の本質的に不完全なセットへの完全に指定された機能のマッピングは、ランタイムでフォールトトレラント処理およびコミュニケーションファブリックを設計し、マッピングし、コントロールするすべてが新しい方法が要求されるでしょう。

2) システムの消費電力

65nm 未満では、MPU 設計は、性能、消費電力、放熱の根本的な壁にあたります。大きさおよび困難さは、早急に高性能なシステムを再考することが必要になるほどです。消費電力は、オンチップ処理の並列化の注意深い適用によってだけ管理することができます。これは、システムレベル設計への異なったアプローチをもたらすでしょう。あらかじめ定められたアーキテクチャに選択された機能をマッピングする代わりに、システムレベル設計の将来のゴールは、最大限に並列化したインプリメンテーションへ最大限に並列化した機能をマッピングすることです。メソドロジ的には、これは、中央処理的なインプリメンテーションに関して、分配されたインプリメンテーションを強調する新しい設計領域を定義します。;これは、先に述べた通信中心の設計の必要性を繰り返しています。よく了解されているアプリケーションに対しては、最適化された HW インプリメンテーションへの直接のマッピングは実行可能かもしれません。;多数の変なスタンダードに依存するアプリケーションについては、リコンフィギュアブルロジックが実現可能かもしれません。;より低いスループットが求められた時には、システムの不必要な部分のすべての電源が落とされるでしょう。そのような傾向の中で、独立型の MPU 設計スタイルは、恐らく、'処理の海'(Sea-of-processing) の要素設計スタイルへと発展するでしょう。

3) 異種テクノロジーの統合

将来の SoC およびシステム・イン・パッケージのテクノロジーは、それらが関連する統合化のキャパシティーに加えて、分離していた学問分野であったアナログおよびミックス・シグナルの設計を終了させるかもしれません。事実上すべての設計領域からのシステム要素が、パッケージあるいは基板レベルのいずれかで統合化のために協調設計されるでしょう。エレクトロ・ケミカル(「チップ上の実験室」)や、エレクトロ・バイオ(センシングや作動、計算が可能な有機的な材料)のような新しいテクノロジーは、新しいモデル化するアプローチ、新しいタイプの生成と統合化のガイドラインおよび規則を要求するでしょう。そしてそのような設計の始まりの数に依存して - すべてが新しいツールセットを促進するでしょう。このエリアにおける初期の研究は、まだ新しい技術自身に注目していて、協調設計、モデリング、統合化およびツールの問題については、まだ注力されていません。

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、ゲートレベル表現まで写像する過程です。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)を相互に接続させます。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します。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)することです。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みます。

設計生産力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 signoff

を要求します。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、**manufacturability** 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します。貧弱な評価による過度に **guardbanding** 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ありません。従って、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て、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でしょう。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レラント、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ロジ選択に極度に依存するでしょう。

論理、回路、物理設計のための困難なチャレンジは表 15 の中で要約されます。短期チャレンジが指摘されたこととしての場合以外は長期へ継続するので、私たちは融合します。重要なポイントは(1)見積りで、(2)モデル化する能力を改善しました。ポイント(1)は、プラットフォームに基づいた設計および信頼できるインプリメンテーション・プラットフォームのような概念、再利用およびプロセスばらつき緩和(たとえばレチクル補正技術)を包含します;これらは、テーブルの中で与えられた最初の 2 つの挑戦に相当します。ポイント(2)は、SOI(Silicon on Insulator)、ミックス・シグナルのインプリメンテーションおよび回路革新の増加した重要性と同様に増加したリーク、**Single-event upset** および原子の規模の結果のような現象を包含します;これらは、テーブル中の次の 2 つの挑戦に相当します。いくつかのアナログ/ミックス・シグナル議論がシステム・ドライバー章のなかに統合されたことに注意してください。

<i>DIFFICULT CHALLENGES ≥ 65 nm / THROUGH 2007</i>	<i>SUMMARY OF ISSUES</i>
Efficient and predictable implementation	Scalable, incremental analyses and optimizations (MPU) Unified implementation/interconnect planning and estimation/prediction (MPU, SoC) Synchronization and global signaling (MPU, SoC) Heterogeneous system composition (SoC, AMS, MPU) Links to verification and test (MPU, SoC)
Variability and design-manufacturing interface	Uncertainty of fundamental chip parameters (timing, skew, matching) due to manufacturing and dynamic variability sources (MPU, SoC, AMS) Process modeling and characterization Cost-effective circuit, layout and reticle enhancement to manage manufacturing variability (MPU, SoC)
Silicon complexity, non-ideal device scaling and power management	Leakage and power management (MPU, SoC) Reliability and fault tolerance (MPU, SoC, AMS) Analysis complexity and consistent analyses / synthesis objectives (MPU, SoC)
Circuit design to fully exploit device technology innovation	Support for new circuit families that address power and performance challenges (MPU) Implementation tools for SOI (MPU, SoC) Analog synthesis (AMS, SoC)
<i>ADDITIONAL DIFFICULT CHALLENGES < 65 nm / BEYOND 2007</i>	
Efficient and predictable implementation	Reliable, predictable fabric- and application-specific silicon implementation platforms (SoC) Cost-driven implementation flows
Variability and design-manufacturing interface	Increasing atomic-scale variability effects (MPU, SoC, AMS)
Silicon complexity, non-ideal device scaling and power management	Recapture of reliability lost in manufacturing test (MPU)
Circuit design to fully exploit device technology innovation	Increasing atomic-scale effects (MPU, AMS) Adaptive and self-repairing circuits (MPU, SoC, AMS) Low-power sensing and sensor interface circuits; micro-optical devices (AMS, SoC)

表 15 Logical, Circuit and Physical Design Difficult Challenges

(1) 論理、回路、物理設計におけるチャレンジ

1) 効率的で見積り可能なインプリメンテーション

- * スケーラブルなインクリメンタル分析および最適化。論理・物理設計ツールが抽象の最低のレベルで動作するので、それらの基本アルゴリズムは少なくとも技術ノードあたり 2X によって成長するインスタンス複雑度に直面します。スケーラビリティは、データを管理する新しい方法が解決スペースおよび分配されたかまたは並列の計算上の資源上への最適化マッピングを横断的に実施することを必要とします。再利用生産性と同様に **Construct-by-correction** メソッドについても、インクリメンタル仕様、合成および分析/検証が、入力に行われた変更の量に比例したランタイムが必要です。さらに、チップ・インプリメンテーションはますます大規模なグローバル最適化および対話し、マルチレベルおよびマルチ目的最適化を要するでしょう。外部要因によるグローバル最適化、資源を有効活用した最適化および部分的プロバプリスティック・設計情報を備えた最適化用技術は必要です。基本アルゴリズム技術におけるそのような改良が、利用可能なインプリメ

ンテーションファブリック(完全にプログラムで可能なことへの十分な検討)の範囲を横断して実施されるべきです。設計プロセス議論で注意されるように、メトリクスは生産性改善を確認するために必要です。

* 一体になっていた、インプリメンテーション/立案および評価/予測を相互に融合させる。今日のインプリメンテーション立案ツールは設計条件のバジェットングおよび概略レイアウトと同時にロジックおよびタイミング構造を作成します。確立しているメカニズムは RT レベルフロアプランニングを組み合わせます、そしてグローバル、リピーター挿入を定義するために立案およびグローバル信号の詳細なレイアウトと同様にパイプラインを敷設することを相互に融合させる;結果は、論理合成および最適化への制約、配置配線として渡されます。関連するシステム・アーキテクチャ最適化はグローバル配線量を縮小することができます;アーキテクチャ最適化を相互に融合させる、与えられたシステムへ配線する数および次元と一致することができます。将来の生産性改善は継続的な論理・物理合成、レイアウトクロックテスト合成などを要求するでしょう、戦略を示すタイミング構造のコ・オプティマイゼーション、ロジック最適化、配置および単一の環境中の配線である、短期ゴールとの統合。十分に検証された設計プロセスが評価と予測の改善により設計解の探索および生産力を改善するだろうということに注意してください。チップ・インプリメンテーションについては、レイアウト評価が(a)により必要です。トポロジーに敏感な、グローバル配線やデバイスのスイッチング時間で計れない遅れを相互に連携させ、(b)デバイス中の容量・誘導性のカップリング相互作用、またそれを相互に連携させる、タイミング/ノイズ確認、および(c)reprogrammable なブロックに支配されたファブリックあるいは IP ファブリック中の実行メトリクスを支配します。DT は、見積り可能な設計プロセスを達成するための様々なメソッドを積極的に追求するに違いありません。システムレベル設計からのハンドオフが RTL 以上、およびより大きなプラットフォームの方へ発展するか、ファブリック特異性、インプリメンテーション・ツールおよびフローが結局プラットフォームに特有のインプリメンテーション・フローの余裕があって、同様に適合する必要があるだろうとともに、長期的です。コスト最適化は別の長期ゴールです。

* 同期およびグローバルシグナリング。ダイサイズが一定のままである一方、高速の設計のためのおよそ 16FO4(16 段のファンアウトが 4 であるインバータ)遅れ(ほぼ 225X、PIDS 章において CV/I 指標)としての最小のクロック期間規模;チップを横断するコミュニケーションは従ってますます多くのクロックサイクルを必要とします。システム・ドライバー章の中で注意されるように、グローバルな MPU は相互に連結します、すでにパイプラインで送られ、クロック周波数のための制限的要因ではありません。しかしながら、純粋な同期設計中のクロック分配は、ダイナミック・パワーの 40%以上を占め、ストレス(たとえば保護する資源必要条件、歩留りロス、ばらつきにより引き起こされた)を増加させることとなります。その結果、同時と非同期設計のより強固で、より効率的な連携への明らかな傾向があります。たとえば、コミュニケーションに基づいた設計パラダイムは、ローカルに同期で高機能ブロックの全体的に非同期カップリングを要します。さらに、グローバルに同期で、ローカルに非同期パラダイムは実行可能です、たとえば、多重コア MPU の中で。チップ上のクロック周波数は 2 つ以上の桁およびマルチ・サイクルによって変わるでしょう、パスを相互に連結させる、普通でしょう。インプリメンテーション DT は、そこでタイミングとパワーの最適化の後に「More paths are critical」タイミング構造と同様にそのような同期パラダイムを支援するに違いありません。後の現象には、プロセスばらつきのために 2 次的に遅れ感度を増加させることが伴います;より大きなマージンの必要性は、結局従来のデレイ-スラッグ最適化上のリターンを

制限します。今までの傾向からグローバルなオンチップ、デジタル PLL(Phase Locked Loop)/DLL 同期にも小振幅差動およびマルチ・フェーズ・クロック技術にも依存する、次にオフチップ化する傾向にあります。これらのことから高められた効率および信頼度、グローバルで、バッファされた、インターコネクト、ブースターのための新しい合成および分析が伴うに違いはない、状態意識および推移意識符号づけ、タイムシェア、資源および他の新しく起こされるパラダイムを相互に融合させます。

- * **混載のシステム構成。** 異成分は多くの形式でインプリメンテーションに影響します。(1) デジタル・ロジックを備えた、アナログ、ミックス・シグナルおよび RF(AMSRF)の混載のシングル・チップ・インテグレーションは、レイアウトを通じて検討することからの新しい挑戦を示します。ツールは、外部制約による定性化(たとえば、レイアウト制約およびレイアウトでの電氣的な規則)に加えて、ノイズと干渉への AMSRF 回路のより大きな感度を扱うに違いありません。長期、MEMS(Micro Electro-mechanical System)の統合あるいは他の技術は斬新な欠点回避およびフォールト・トレラント方法を要求するでしょう。(2) 費用要因(製造原価およびコミュニケーションのコスト)によって駆り立てられて、統合的な選択では、マルチ・チップ(パッケージでのシステム)および積み重ねられたオプションを包含するに違いありません。たとえば、システムレベルの最適化の場合に、パッケージおよびボードまで相互に連携します。アセンブリおよびパッケージング章は、DT と共有される重要な挑戦としてチップ-パッケージ分析およびコ・オプティマイゼーションのツール支援を指定します。(3) コスト考察(マスク NRE、設計 TAT)によって再び駆り立てられて、reprogrammable なブロックは、SoC 設計の中にますます重みをますますでしょう。レイアウトのツールは、再プログラム化可能のより豊富なパレット(電氣的か、金属のみの変更によってのみ)と同様に、より物理的な結果形状および制約を扱うに違いありません。(4) 新しいツールはブロックキャラクタライズおよび抽象を含む困難か、半困難な IP 再利用、制約管理および柔軟な機能的な/物理的な階層管理を支援するために必要です。IP ブロック統合で、インプリメンテーションにより困難度がまします。たとえば、ブロックとセルの「ミクスト・モード」配置は今後も挑戦し続けます。プロセス技術の抽象化はプロセス世代を横断した再利用生産性のために必要でしょう。再利用の別の面はデザイン変更です:長期とインクリメンタル合成は、できるだけ既存の設計の多くを保存する間に既存の設計へ新しい設計特徴を取り入れるべきです。(新しい設計は従ってデザイン変更への参加に従順になるに違いありません。)
- * **検証とテストへのリンク。** 従来のポスト・シリコンをテストするメソッドがあまりにも高価で複雑になるとともに、より系統的なテストおよび検証支援はチップ・インプリメンテーションの間に必要でしょう。将来のロジックおよびレイアウト合成は、タイミング、消費電力、密度、信頼度およびシグナル・インテグリティ制約を尊重する間に自動的にテスト構造を挿入するに違いありません。アプローチは、利用可能なオンチッププロセッサ(たとえば補足のファームウェア・コードによる)のレバレッジへのセルフ・テスト時間などを測るでしょう。動的なことのより大きな融合への傾向としては、ノイズへの非同期と AMS/RF 回路増加の影響、クロス・トークおよびソフト・エラーがあります。これは、恐らく欠陥に基づいた試験に対する信頼の増加に結びつくでしょう:たとえば、物理設計ツールは、新式の欠陥のための欠点位置を抽出する必要があるでしょう。同時に、迅速な生産性向上では、回路に起因する幅広い種類の故障の迅速な分析を支援する、物理的なレベルのツールを要求するでしょう。

2) ばらつきと設計-製造とのインターフェース

プロセスの許容値(process tolerances)を継続的に維持することが不可能となることで(PIDS、配線、またリソグラフィ章を参照) ばらつきに対する DT(設計技術)のチャレンジは深刻さを増しています。Lgate や Tox のような基本のパラメーターの 3-シグマのばらつきや、配線の寸法ばらつきが優に 15%を超えることで、ばらつきに対処すべく、新しい回路トポロジや論理/レイアウトが必要とされます。実に、原子規模での影響が、多くのプロセスの段階(ゲート酸化平坦化、イオン注入など)での統計的ふるまいを支配します。

- * *統計的タイミング解析および性能検証ツールは、分散によってパラメーター化される寄生値、遅延値およびジオメトリを理解する必要があります。一般に、設計の合わせ込み(design centering)は、従来の性能指標のためよりも、むしろ 1 つのウェーハあたりの、パラメーター化された歩留りおよび収獲量(revenue)のために最適化することになります。製造ばらつき(および豊富な新しい材料やプロセス)のために、レイアウトの設計ルールおよびプロセスの抽象性を提供する、より広範囲な設計-製造のインターフェースが要求されます。統計情報を豊富に持つ電氣的/幾何学的な製造ばらつきの原因要素のキャラクタライズが必要です。動的なばらつきの重要なパターンは動作中の熱の効果によります;このばらつきは何十億ものクロックサイクルのタイムスケールの上でのことです。そして、タイミングとノイズに大きく影響します。消費電力密度が上昇し続けるとともに、熱により引き起こされたばらつきに対する単純な保護(guardbanding)をする手法ではコスト高になります。ローカルな発熱のばらつきに対するより正確な分析および制約は過剰品質な設計(overdesign)の量を減らすために必要です。*
- * *レチクル補正技術(RET(Reticule Enhancement Technology)) は多層配線プロセスの平坦化(ダミーパターンの挿入によるレイアウト密度の制御が必要。)とディープ・サブ波長(deep-subwavelength)のオプティカルのリソグラフィ(光近接補正 OPC(Optical Proximity Corrections)と位相シフトマスク PSM(Phase-shifting Mask)のレイアウト)を含みます。RET は製造複雑さの要因の成長度合いを反映し、プロセスロードマップの先行きを困難とします。RET は、レイアウト設計の複雑度、製造へのハンドオフの複雑度、かつ、製造での NRE コストといった観点において、DT に対する負荷を増大させます。(a) OPC と PSM において、レイアウト合成の生産性は、非常に複雑な、状況依存(context-dependent)の設計規則によって困難になります。「ローカルの設計規則」がもはや存在しないという点を、レイアウト検証は受け入れる必要があります。物理検証は正確に理解しモデル化する必要があります。(たとえばポストテープアウトでのレイアウト・データ・ベースに挿入されるダミーメタルの RLC 抽出へのインパクトなど)。(b) 製造へのハンドオフは、RET の広範囲な適用により、データサイズおよびマスク製造/検査コストが激増することで、耐えられなくなります。RET 挿入(またマスク検査)では、単にある重要なデバイスか、あるいはある配線の寸法にのみ注意深い挿入の価値があること、そしてある挿入メカニズムについては実装し検証することのコストが、他のメカニズムよりも高価であることを理解する必要があります。従って、将来の設計フローは、物理検証およびマスク・フローへと、詳細な機能的な意図およびシミュレーション結果を下流へと渡す形になるに違いありません。長期的観点での支援技術は、時代遅れで非能率的な GDSII ストリームおよび MEBES データを置き換えるでしょう:DT と製造の間のより完全に双方向なパイプラインが、恐らく電子システム「サプライ・チェーン」のインフラストラクチャの中で出現するに違いありません。*

3) シリコン複雑度および非理想的なデバイススケール

最下流での抽象度での操作は、多くの困難な物理効果に関して、詳細なモデリング、分析および最適化が要求されます。

- * *リークとパワーマネージメント*。供給電源の低電圧化は、消費電力密度の増加に伴うより多くの電

流により、相対的に電源線での大きな誘導性ノイズを引き起こします。これは、バンプ数 (bump counts) およびピッチでのそれほど積極的でないスケールリングに加えて、同期動作およびパワー・アップ/リセット条件によって悪化します。設計工程の初期段階で、デカップリング容量の挿入のように、IR ドロップや電流のサージを管理するための、電源設計が要求されます。100nm 以下では、注意深く設計された電源配線でも、大きなスイッチング電流によって問題がでるかもしれません。パッケージ側では、信頼性のため、連続的に ~100ms より多くのパッケージ格付け (package rating) を超過しないように最大電力の管理を要求します。パッケージおよびパフォーマンスの信頼度は双方とも、チップ上の温度ばらつきのコントロールを要求します。従って、論理合成およびタイミング/レイアウト最適化を行うツールのアルゴリズムでは、瞬間と平均消費電力に対処できる必要があります。システム・ドライバー章が指摘しているように、MPU および SoC-LP の中のスタンバイ・パワー管理「ギャップ」が、ITRS の最終ノードまでに 25X と 800X に達します。MPU での継続的に増加する消費電力のバジェットは、改良された解析機構 (寄生抽出、ロジック活性度、電流フロー) に加えて、新たなチップ-パッケージでの熱分析を要求します。ライブラリーキャラクタライズ、合成およびレイアウトは多値のスレッシュホールド電圧や多電源の使用からおおよそ 5X の消費電力削減を要求されます。ツールとしては、(Peltier-type thermoelectric cooling) のような長期的観点での技術と同様に、OS を媒介とした動的な周波数、電源管理によって実際の発熱の制御を可能にするような合成の構造が必要になるでしょう。

* **信頼性およびフォールト・トレラント.** 信頼性基準(ホットキャリア効果、エレクトロマイグレーション、ジュール(joule)の self-heating)は、単純で明快な抽象的概念(たとえば、ゲート負荷容量対出力 SLEW の上限値)によって実装工程へ統合されました。現在、そのような「メソドロジ」の抽象的概念は、従来のツールフローに若干の修正で、修正および構築しなおすことが可能です。しかしながら、保護 (guardbanding) (従って 1 つのウェーハあたりの増加する抽出値) を削減するように、抽象性および解析を改善する必要があります。他のチャレンジな点は以下のとおりです。(1) 短期的観点での困難さは、イオン化した (ionizing) 放射線(それは bistable なロジックだけでなくランダムロジックにも同様に影響を与えるであろう)によって引き起こされた、Single-event upset(SEU、あるいはソフト・エラー)に関してであります。縮小スケールリングによって、Qcrit 値は、アルファ粒子からのノイズパルスさえ論理故障を引き起こす値にまで減少します。SEU からの保護のためには、恐らくセル、デバイスタイプおよび配置による統計可能性の特別な解析(個別のキャラクタライズステップとして実行された)を要求するでしょう。その後、自動的な手法では、設計制約を破らずに、SEU を防ぐか管理するために論理、回路、物理設計(たとえばエラー訂正の自動導入、サイジングなどによる)を修正することが必要です。(2) 短期的観点、長期的観点での問題は製造されたデバイスあるいはデバイスおよび配線の動作の 100%の正確さが失われることです-たとえば、Design Test での討議で、従来の加速された耐久試験(「バーンイン」("burn-in"))パラダイムは実現不可能になります。それは (a)供給電圧を減少させることで、指数関数的にバーンイン回数を増加させること(b)より消費電力が増加することで、バーンインのコストを増加させてしまうためです。自動的な論理・物理的な合成は、適切に合成され統合されたオンチップセルフ・テストロジック(設計検証議論での self-checking core の議論参照)の管理の下で作動するフォールト・トレラント技術(冗長ハードウェア、再プログラミング可能な配線、またリコンフィギュラブルなコントロール・ロジックを含みます)を統合するようにますます要求されるでしょう。(3) 新しい欠陥モデルは原子規模の効果により必要です。たとえば、ゲート酸化物は今、非常に薄いので、(あまりにも少数の遷移可能状態(too few trap states)しかありません)破滅的な酸化物のブレイクダウンを起こしえます。もっと正確に言えば、電流は、単純に時間とともに、および酸化物のダメージと共に増加します。

「欠陥」を生成するもののために必要とされる、斬新なスクリーニング基準だけでなく、特殊な回路トポロジでの受け入れ可能な欠陥密度の依存性を理解する必要があります。

- * **モデリングと解析.** 増加するシリコン複雑度および保護 (guardbanding) は、消費電力/電流およびタイミング/ノイズに関してのモデリングと解析を困難にします。主要なチャレンジは、最小の資源および実行コストでカップリングノイズや、遅延の不確実性を緩和するように配線(信号、クロック、パワー・グラウンド)の最適化を支援することです。(1) 電源設計および消費電流の管理を行い、IR ドロップおよびアクティブな消費電力のより正確な分析を支援するためには、キャラクタライズツールおよびより豊富なライブラリー・モデリングの標準化が必要です。速くて詳細な熱解析が要求されるかもしれませんが、たとえば、エレクトロマイグレーションの信頼度のリスクを削減するために、低い誘電率 (low-permittivity) の誘電体の熱伝導性を削減することで、配線の joule heating が問題となりえます。(2) タイミングおよびシグナル・インテグリティの収束は、オンチップおよびパッケージ中の寄生容量およびインダクタンス、配線、クロス・トークノイズ、カップリングに誘起された遅延の不確実性および基板ノイズ注入の正確な解析に依存します。将来の配線最適化のための評価では、信号、供給電源ノイズ、熱の勾配、そして - サブ 100nm ノードの中で - 反射、EMI および基板カップリングなどを含むでしょう。この意味で、今日のタイミング/ノイズライブラリーキャラクタライズでは、ナノメータプロセスでの合成、高精度な解析の観点で、大きな抜け(たとえばプロセスばらつき)の認識があります。(3) より一般的に言って、ナノメーターの実装フローでは、CPU 対精度でのトレードオフのある種の「単調な悲観性」(“monotonically pessimistic”)を要求します。たとえば、より速くそれほど正確でない解析が、論理/レイアウト合成の初期の段階でのフィルタリング、予防に対して十分であるのに対して、既知のクリティカル・ネットはタイミング signoff の中のより時間のかかる、より正確な解析を要求するかもしれません。最終的には、合成での最適化指標として役立ち、効率的に評価され、安定動作をする「マクロ・モデル」によって、解析精度は近似されます。

4) デバイス技術革新を最大限に利用した回路設計

PIDS ロードマップに記述されるように、非理想のスケールリング(たとえば、供給電源のスケールリングがしきい値電圧より進む場合)は、より高いゲートとドレインのリーク電流、Body-Effect(パス・ゲート・ロジックをそれほど魅力的でなくします)およびオーバードライブの損失を引き起こします。パワー・マネージメントのチャレンジに照らせば、従来の処理速度との様々なトレードオフ、即ち、より大きな消費電力、ノイズ感受性、および自動ツールが使えないことなどは、それほど問題ではなくなります。DT は、処理速度/消費電力の性能における全面的な改良を可能とする、スタチックな CMOS に替わる新たなデバイスの展開を可能にするに違いありません。レイアウト自動化および物理検証(たとえば、レイアウトからの最新のアクティブ・パッシブ構造の抽出を自動化すること)の能力は、回路技術の革新と協調して開発されねばなりません。そのような DT の進歩はデバイス技術の進歩と補完的であるに違いありません、たとえば、(a)ゲート絶縁体トンネル効果および破壊の保安全性(それはフローティングのデバイス、ダイナミック回路などに影響する)の緩和は、高誘電率のゲート絶縁体を必要とします、また(b)基板にバイアスをかける(パワー・マネージメント・ギャップに取り組むダイナミックな V_t 調節にとって不可欠)ためには、改善された body contact 技術を要求します。

- * **消費電力と性能の挑戦に取り組む新しい回路ファミリーのサポート.** パワー・マネージメントがアーキテクチャ、OS および適用レベルでの中心となるとともに、いくつかの回路トレンドは DT のサポートを要求します。(a) ローカルに非同期、グローバルに同期のアーキテクチャを使用することで、clock-delayed ドミノあるいは delayed-reset ドミノのような自己充足 (self-sufficient) な回路をさらに、一般的にするでしょう。(スタチックな CMOS は普及したまでであるが)。(b) 非常に低電圧のオペレーションは、デバイス性能要求が並列性

の使用によって緩和されるとともに、より一般的になるでしょう。(c)ボディ・バイアス、クロックおよび電源レベルのようなパラメーターの制御がコンパイラそして、OS ベースになることが増加することで、物理的な応答性(responses)が、アーキテクチャレベル管理をサポートするためにビヘイビアのレベルでモデル化されるに違いないでしょう。

- * *SOI のためのインプリメンテーション・ツール.* シリコン・オン・インシュレーター(SOI)技術はより低い接合容量により、より速いスイッチングを提供し、またデジタル・アナログ基板ノイズを除去します、これはミックス・シグナルの SoC には特に魅力的です。しかしながら、設計者がバルク MOS デバイスから問題なく SOI に移ることができるためには、多くの DT 問題が解決される必要があります。フローティング・ボディ効果は歴史的に自明な解析を要求します:ワーストケースでの保護(guardbanding)が必要であれば、SOI 利点の多くを取り除いてしまうでしょう。これらの解析はタイミング(より複雑なライブラリになるでしょう)および静的消費電力(V_{th} 、それに影響されるリーク、カップリング容量およびインパクトイオン化(impact ionization)に伴う変動などを考慮した)を含んでいます。アルファ粒子 SEU が縮小され、ラッチアップが除去される等、信頼度メカニズムは注意深く解明されるに違いありません。しかし、self-heating および ESD 保護は、埋められた酸化物層の熱抵抗が増加することによりさらに問題になります。SOI により減少したデバイス接合容量はデバイススイッチング速度を上げる代わりに、 $V_{dd} - GND$ のデカップリング容量を削減します;従って、電源完全性(オンチップのデカップリングは、おおよそスイッチング容量の 10 倍になります)の維持は、初期のフロアプランニング段階から、明確にデカップリング容量の管理を要求するかもしれません。完全に-depleted な SOI(FD-SOI)は製造容易性の不安を持っているが、上記の問題の解決策が見つからない場合にはより多くの注目を浴びるかもしれません。
- * *アナログ合成.* スケーラブルな SoC 設計は、アナログ設計でのボトルネックを取り除くことを要求します。システムレベル設計と共有されるのに必要な技術は、再利用可能なリターゲット可能なアナログ IP ジェネレーターにあります。今日の回路(PLL、オペレーションアンプ、パワーアンプなど)などのクラスのための専門的自動的な回路合成が、自動レイアウト合成と同様に、必要となります。アナログ合成は、インパクト電離(impact ionization)、熱の非線形性、バンドパス(bandpass)フィルタなどの役割をするボディコンタクトによる、ひずみと非線形の増加に対処する必要があります。合成は、今後さらに増加する製造ばらつきを支配するに違いありません、たとえば、デバイスのミスマッチに対するハイブリッドなアナログ - デジタルの補償など。近い将来、オプティカルのインターフェース回路や高い Q 値の調整可能な CMOS オンチップのインダクタ/共鳴デバイス用(inductors/resonators)の新しい合成ツールが必要です。長期的な回路タイプの関心は、マイクロオプティカル(たとえば、ビーム制御による)デバイスと同様に非常にロー・パワーのセンシングとセンサーインターフェース回路などがあります。本質的に構造上の問題ではなく「仕様に」則っているか否かという、検証、およびテストが、アナログ回路には必要です。より常に速いシミュレーションが歴史上の課題ではありましたが、一方、新しい検証ソリューションでは、統計的技法や、精度を犠牲にせずに、受理基準を含んでいる範囲でシミュレーション速度を促進するよりよいコンパクトなモデルが必須となるでしょう。BIST/BOST 技術は、ますますアナログ回路(たとえば高速のオプティカルのネットワーキング・インターフェース)の周辺で、合成され置かれるでしょう。そのような技術は侵害的であってはなりません、つまり、それらは、組込みアナログ・ブロックのパフォーマンスを下げてはなりません。

設計検証

設計検証は与えられたデザインが意図した動作を正確に実行することを検証するタスクです。設計検証は設計全体のプロセスに対し比較的小さな脇役しか果たさないということが、大半の半導体産業のごく一般的な見方です。(恐らく、そのような見方はこのドキュメントの全構造に反映されています。) しかしながら、現実には著しく異なります。検証は設計全体のプロセスで支配的なコストになりました。現在のプロジェクトにおいては、最も複雑な設計において、検証技術者の数が、設計者に対し2あるいは3対1の比率で勝っています。設計コンセプトおよびインプリメンテーションは、主要な活動である検証の単なる準備行為になりつつあります。

この不幸な状況は以下の 2 つの結果です。まず最初に、近代、設計の機能の複雑化は息をのむほどの速度で増加しています。設計規模は、Moore's Law で指数関数的に成長しています。最悪の場合、機能の複雑度(検証されなければならない、システムの独立した状態数によって表される)は、設計規模が指数関数的増加することにより、ダブル指数関数的に増加する可能性があります。次に、設計全体のプロセスにおいて、ボトルネックである検証を除いて(残して)、その他方面にて歴史上すばらしく強調すべき、もしくは巨大な発達をもたらしました(たとえば、論理合成、配置配線、テストパターン生成用の自動ツール)ことです。主な突破口なしでは、検証が予測不可能で、半導体産業における進歩を促進する甚大なる障害となるでしょう。

これらの突破口の全体的な傾向は、場あたりの検証方法から、よりフォーマルなものへの移行です。特定のフォーマル検証技術が成功するかどうかでも議論する余地がありますが、しかし、総合的な観点では移行は避けられません。繰返し構築するモデルを用いたり、場あたりのベクターの選択で機能をシミュレーションしたり、さらにたまたま発見されたバグを修正したりすることにより、システム設計の機能検証を試みるべきではありません。- しかし、これは確かに今日用いられている方法です。シミュレーションに基づいた試行錯誤検証方法は、膨大に時間がかかりかつ、予測不能です。なぜなら前述の機能の複雑度はダブル指数的に増加するためです。技術の発達は、設計分析の厳格なまたは効率的な方法に依存します。(フォーマルまたはセミフォーマルの大部分は今日研究されている)。信頼できる厳格なもの、単純化もしくは不必要なもの(“構築の正確さ”の形式を経由し)として予想できる技術など、多くの検証上の挑戦が存在します。表 16 で、設計検証の主な挑戦を要約します。残りのセクションではさらに詳細を記述します。解説のために、内容はシステム・ドライバーに応じて構成されます。

<i>DIFFICULT CHALLENGES ≥ 65 nm / THROUGH 2007</i>	<i>SUMMARY OF ISSUES</i>
Increase verification capacity (SoC, MPU, AMS)	Verification exponential in design size Need high coverage Need to handle large designs Semi-formal techniques
Robust verification tools (SoC, MPU, AMS)	Verification algorithms highly unpredictable Improved heuristics Characterization of problem difficulty
Verification metrics (SoC, MPU, AMS)	Behavior coverage Realistic bug model Algorithms to determine bug coverage
Software verification (SoC)	Software intrinsically more difficult to verify Traditional software verification techniques inapplicable Integrated hardware/software systems Design for verifiability (long-term)
Verification reuse (SoC)	Must allow reuse of verification of IP blocks Specify abstract behavior of IP blocks Specify environmental constraints of IP blocks Hierarchical verification algorithms
MPU verification methodology (MPU, +long-term)	Different cost-benefit trade-off (higher cost acceptable) Need exceptionally high capacity Must be very predictable due to long design cycle and pipelined development teams
MPU design-for-verifiability (MPU, +long-term)	Will be necessary sooner than for other system drivers Specialized techniques likely possible
Greater concurrency (MPU, + long-term)	New processors far more concurrent Greatly increased verification complexity
Anything is progress (AMS)	Extremely primitive state-of-the-art Forces difficult hybrid-systems issues into near term
<i>ADDITIONAL DIFFICULT CHALLENGES < 65 nm / BEYOND 2007</i>	
Design for verifiability (SoC, MPU, AMS)	New methodology needed Characterize and minimize performance and area impact
Higher levels of abstraction (SoC, MPU, AMS)	New algorithms needed Complexity of designs enabled by higher-level design Equivalence checking vs. RTL
Human factors in specification (SoC, MPU, AMS)	Specifications of correctness will become unmanageable Need to understand what kinds of specifications are most understandable Need to consider how to make specifications modular and modifiable
Verification of non-digital systems (SoC, MPU, AMS)	Hybrid systems verification for analog effects Hybrid systems verification for analog properties Verification of probabilistic systems
Heterogeneous systems (AMS, SoC)	How to model, analyze, and verify MEMS, EO devices, and electro-biological devices

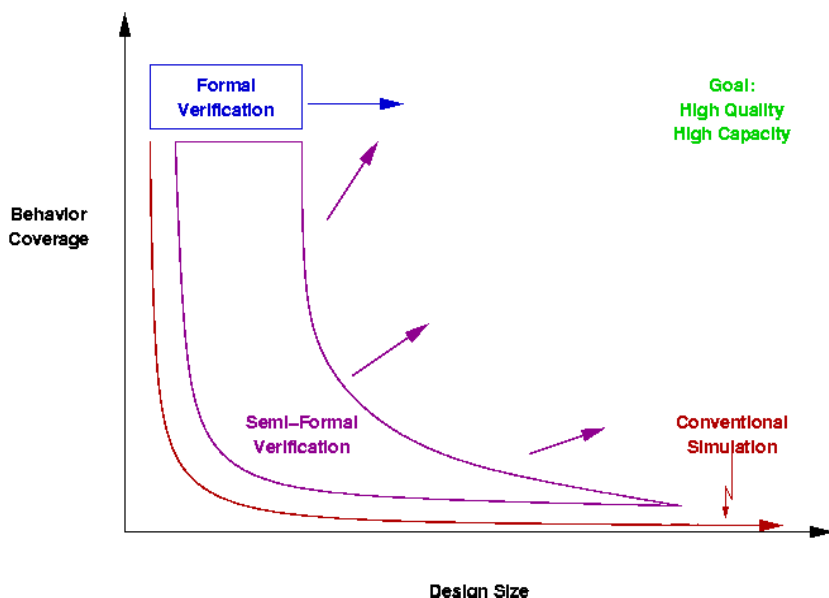
表 16 *Design Verification Difficult Challenges*

(1) 設計検証の挑戦 - 概略

検証のための最も重要な挑戦の多くは、大部分あるいはすべてのシステム・ドライバーに関連があります。短期的には、主要な 3 つの課題(下記(1)~(3))が、より信頼でき制御可能なフォーマルもしくはセミフォーマルな検証手法を開発することに焦点を合せています。特に、優れたキャパシティーやロバストなフォーマル検証ツールが、高品質な検証に意味あるものとして望まれています。長期的には、主要な 4 つの課題(下記(4)~(7))がフォーマルな検証の範囲を広げ、抽象レベルを上げることに焦点を合せています。これらの長期的な問題は現に今日意味のあるものです。しかし短期的な課題への挑戦はすでに危機的状態にあります。

1) キャパシティー

現在のフォーマルな検証ツールは大きく複雑なデザインを信頼して処理できません。シミュレーションに基づいたツールは任意に複雑なデザインをシミュレートすることができますが、シミュレーション時間が急増するにもかかわらず見えないほどの小さなカバレッジしか供給できません。検証のための唯一の最も重要な挑戦は大規模で複雑なデザインに高品質検証カバレッジを供給する方法です。図表は現在のツール概要を示します。従来のシミュレーションは、小さなデザインの高いカバレッジ、または大規模なデザインに対しては高いカバレッジではなく非常に貧弱なカバレッジしか提供することができません。フォーマル検証は、検証しようとするデザインの起こり得る動作の完全なカバレッジを提供しますが、現在大規模な設計を扱うことができません。セミ・フォーマル検証は、キャパシティーを獲得するためにカバレッジを犠牲にして、フォーマルで、シミュレーションに基づいた技術を混合することを試みます。その挑戦は、図形の上部の右コーナーに近づくことです。



図表 18: Current Verification Tool Landscape

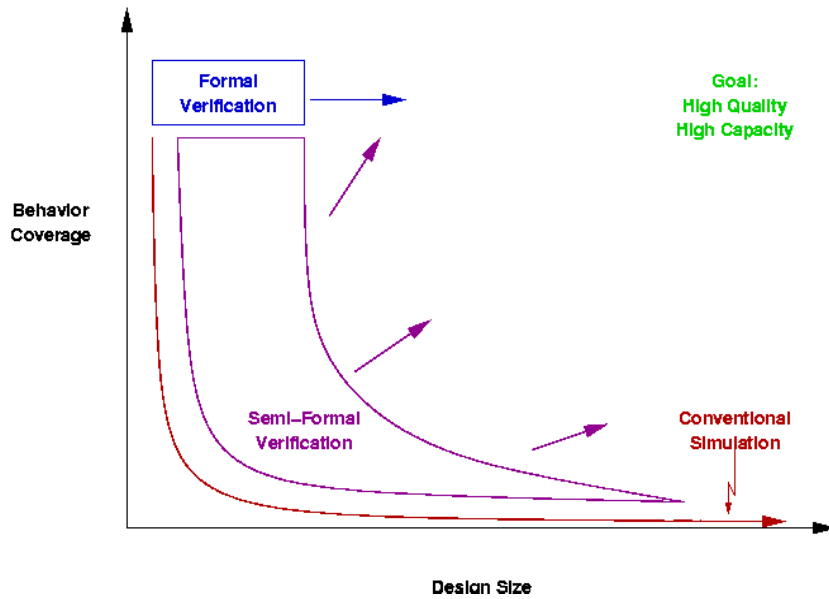


図 18 Current Verification Tool Landscape.

2) ロバストネス

図表 18 は、ある規模のデザインのためにある検証方法がうまくいくという印象を与えます。この印象は誤っています。検証課題の複雑度に対処するために、検証アルゴリズムは高度に気まぐれなヒューリスティクスに依存します。デザインと検証アルゴリズムの所定の組み合わせについては、熟練者さえ検証アルゴリズムが完全かどうか決めるのにひどく苦労するかもしれません。トランジスター、ゲートあるいはラッチ数のように、問題の規模を計る共通的な尺度は、検証複雑度と曖昧にしか関連しません。既知の検証方法をすべて無視する 100 未満のラッチを持つデザインを見つけることは、容易に確認することができるラッチを何千も使用するデザインと同様に簡単です。そのような予想不可能なことは設計現場において受け入れがたいものです。重大な検証の挑戦はもっとロバストな検証の処理をさせることです。これは、検証アルゴリズムのヒューリスティクスを改善するか、あるいは検証容易化な設計メソッドを進め、与えられたデザインに対する検証の困難さの特性を改善することです。

3) 検証メトリクス

短期的な検証の挑戦は、検証エフォートの質を示す要望への対応です。特にカバレッジの有意義な概念は必要です。2 種類のカバレッジ: 動作可能性カバレッジ、可能のあるバグのカバレッジは区別しなければなりません。

- * 動作のカバレッジは十分に定義されていますが、シミュレーションが基本で、セミ・フォーマルな検証にのみ当てはまります。どんなフォーマルな技術も、動作カバレッジを 100% の歩留りで、100% 探求することはできません。さらに、動作カバレッジメトリクスは、一般的に、従来のシミュレーションが設計可能なカバレッジのごくわずかな部分だけを評価できることを示します。實際上、シミュレーションは、これらのメトリクスが示す設計エラーを見つけることよりか、はるかに有効になります。
- * バグカバレッジは検証カバレッジの重要な手段です。しかし、それは現在ほとんど理解されてい

ません。それはテストにおけるフォルトカバレッジと概略で類似しています。しかし不運にも、任意の方法における設計エラーが製造欠陥に似ているという証拠はありません。問題は起こり得るバグ(フォルト)のどの部分が、動作カバレッジのいくつかのコンビネーションに基づいてチェックされ、そして仕様が検証されたかを確定することです。検証された仕様が不完全な場合、設計段階における検証がフォーマル検証によってパスしても、要求された動作に対しては依然問題があります。その挑戦は有意義なバグモデル(欠陥モデルのアナログ、しかし実際上発生する設計エラーについての深い理解を要求します)や与えられた 3 条件(設計、仕様、検証時間)で、バグカバレッジを計算するアルゴリズムを開発することです。

4) 検証容易化設計

短期的な挑戦のソリューションは、検証するのに何が容易もしくは難解か、どのように設計エラーが起こるか、の理解をすることです。より長期的な挑戦のソリューションは、検証容易化設計を生み出す知識を明文化することです。検証容易化設計なしで、検証が 2007 年以降の設計取って、細工しやすくなりそうにありません。設計上の大きなメソッドロジー変化が必要とされるでしょう。そして、幾つかパフォーマンスが低下しそうです。役に立つアナロジーは検査可能性を継続することです。そこでは、シケンシャル ATPG の計算上の扱いにくさは、スキャンテストのほぼ普遍的な適用の帰結です。

5) 抽象度のより高いレベル

設計が RTL を上まわる抽象的なレベルに移るとともに、検証はそのレベルについていかなければならないでしょう。その挑戦は、抽象度の高レベルに対する検証方法を適用・開発することです。より高いレベルの設計によって、増加したシステム複雑度を扱うことが可能になり、高レベルおよびより低レベルのモデル間の同等性をチェックする手段を開発することでしょう。この長期的挑戦は、もし高レベルにおける抽象化の決定が検証を無視して行われる場合、はるかにより困難なものになるでしょう。(たとえばあやふやで不必要に複雑な言語、もしくは RTL モデルとの形式上の関係を持っていない、シミュレーションのみのモデルに依存するメソッドロジー等)

6) 仕様決定における人為的要因

設計検証のための持続的な挑戦は、デザインに要求された動作をいかにして明確化するかです。何が仕様をクリアにするのか不透明にするのか、また変更し易いものにするのかし難しいものにするのかの深い理解は、複雑な設計をするために必要な言語の開発のためには必要とされるでしょう。そしてさらに、設計者は、これらの言語を使用する訓練をし、かつ仕様書を書くことについてももっと訓練されなければならないでしょう。

7) フォーマル検証の適用範囲拡大

現在まで、設計検証はデジタル・システムの独立した動作に焦点を合わせてきました。シリコン複雑度およびシステム複雑度への 2 重の挑戦は、システムのより広いクラスを分析するための、将来の検証努力を強制するでしょう。ロー・レベルのものでは、シリコン複雑度はクリーンであると叙述されており、VLSI システムのデジタル部の抽象度はますます不安定で危機になるでしょう。アナログ電気的効果は、実行や遂には機能に影響を与えるでしょう。これらの結果を分析するための既存のシミュレーションメソッドロジー(たとえばスパイス)は遅すぎ、そして、おそらくプロセスのバラツキに対し、ますます敏感になる小型デバイスに対し、信頼性が低くなるでしょう。長期的なものでは、フォーマル技術は、アナログとデジタルの境界でこれらの問題をハイブリッドシステムとして取り扱い、検証することが必要

とされるでしょう。同様に、設計の最も高い抽象レベルでは、将来の検証タスクを規定するシステムの複雑度が、アナログおよび確立論的動作（たとえばネットワーク・プロセッサ中のサービス保証の質）の規定および検証を要求するでしょう。従って、ハイブリッド・システムおよび **probabilistic** な検証の挑戦が必要でしょう。

(2) 設計検証の挑戦 – SoC-特性

広範囲検証の挑戦はすべて SoC に適用されます。さらに、次の検証の挑戦は、SoC にとって特に重要です。

1) ソフトウェア

SoC の機能性の多くはソフトウェアによって定義されるでしょう。従って、主な SoC 検証の挑戦はソフトウェアおよびハードウェア/ソフトウェア・システムを検証する方法です。伝統的に、ソフトウェア開発は、設計レビュー、分析ツールおよびテストなど行われているが、ハードウェア開発ほど厳格ではありません。その上、ソフトウェアはもっと複雑でダイナミックなデータおよび巨大なスペース（一般に無限モデル化され、はっきりとしない検証を作成した）などにより、検証することは本質的に難しいです。ソフトウェア検証の古典的なフォーマルな技術は、SoC に広く適用するのは、あまりにも労働集約型すぎて適用可能になりません。ロー・レベルのソフトウェア用の新しい検証方法が開発されなければならぬでしょう。短期的な挑戦は、ソフトウェアの正確な基本部分を検証する技術を開発することです。長期的な挑戦は、ソフトウェアに適用されるような検証容易化設計についての理解と同様に、ソフトウェア用のロバスト検証方法を開発することです。

2) 再利用

あらかじめデザインされた IP ブロックは、ごくわずかな時間で先例のない複雑な SoC を組み上げることを可能にすると約束します。主な挑戦はあらかじめ計画的な（そしてあらかじめ検証された）ブロックから組み上げられるシステムを迅速に検証行うことが可能となる検証方法を開発することです。重要な問題は、IP ブロックの抽象的な動作について厳密で完全な記述をする方法、IP ブロックによって仮定された環境上の制約について記述する方法、また検証を単純化する階層を利用する方法です。近々での進展は相互に連結するために標準化された IP（オンチップパスのように）のための解決が最もありそうです。しかし、任意の IP ブロック・インターフェース用の一般的な問題も最終的には解決されなければなりません。

(3) 設計検証の挑戦-MPU-特性

MPU は以下の意味で異なった検証の両方について挑戦が必要です。リーディングエッジである複雑度のため、そして比類なくハイ・ボリュームで使用され、信じられないほどに複雑な独自性の設計のためです。その結果、短期的・長期的、両方とも異なった領域に特有の検証の挑戦および検証機会が存在します。

1) 特別な検証方法

MPU 設計は、常に検証ツールのキャパシティーを超えるでしょう。しかし、成功した MPU は高い経済的利益があるため、より労働集約型の方法が許されます。たとえば、半自動な **theorem-proving** 技術は、恐らく近い将来 **quick-time-to-market** が必要な SoC 設計における広範囲の適用は、遅すぎかつ高価すぎるでしょう。しかし、ハイ・ボリュームの MPU 設計において、特定の困難な検証タスクのために有効に、すでに使用されています。重要な挑戦はキャパシティーと予見性です。予見性は、

他のシステム・ドライバーよりも、MPU 検証にとってより重要です。なぜなら設計サイクルが非常に長く、連続する製品世代に対して多数の設計チームがしばしばパイプライン的に設計するからです。長い設計サイクル、将来に向けて計画され、確保された長い検証プロセス時間、しかしこれは検証プロセスが非論理的に膨らまない場合のみです。MPU 検証の複雑度を扱うために最もよい追加資源は何か？ 変化する検証時間を短縮するために、最も良い追加資源は何なのでしょう？

2) 特別な検証容易化設計

同様に、恐らく他のシステム・ドライバーより早く、MPU 設計の複雑度は検証容易化設計を要求するでしょう。また、MPU 設計の独自の性質は、特定の領域における検証容易化設計が可能であると思われる。たとえば、小さなウォッチドッグプロセッサにある、セルフチェックプロセッサで行われている予備ワークがメインプロセッサの正確な実行を検証できます。MPU 設計にとっての検証容易化設計が挑戦として行われ、それは面積と性能のペナルティを最小にして検証コストを効果的に削減できるでしょう。

3) 新しい種類のコンカレント

MPU 設計がより複雑になるとともに、新しい種類のコンカレント設計が重要になっています。すでに、検証を回避できるバグの多くは、キャッシュ coherence とその他のコンカレントな問題とに関係があります。新しい設計は、チップレベル多重プロセッシング(とオンチップキャッシュ coherence プロトコル) および検証プロセスを非常に複雑にする同時のマルチスレッディングのような技術による、コンカレントなレベルを増加させます。その挑戦としてはコンカレントな新しいフォームを扱うために技術を開発することです。

(4) 設計検証の挑戦 - AMS-特性

AMS 設計は他のシステム・ドライバーと、より広範囲の検証の挑戦を共有します。しかしながら、主要な違いは、それらが長期的課題であるハイブリッド・システム検証を短期的な課題として注力しているということです。不運にも、ハイブリッド・システム検証についての研究はまだその初期の段階です。従って短期的な挑戦は、現在の特別なアプローチを改善するのに可能なあらゆる支援を提供することや、スタンド・アロンシミュレーションの利点を越える、より強力な技術開発に必要とされる基礎研究を開発することです。長期的には、アナログは単にエレクトロニクスだけを指すのではないのかもしれませんが。もし MEMS、エレクトロ・オプティック、エレクトロ・バイオロジカルデバイスが単なる変換器以上になると、主な検証の挑戦はモデル化し、分析し、そのような混載の部分具备了システムを確認する方法になるでしょう。

テスト設計

ナノメーター・プロセス技術、高くなるクロック周波数および SoC インテグレーションは、テスト (DFT(Design for Test)) のために設計するということが必要になります。テスト産業はコンポーネントに基づいた設計のためのハイレベルのテスト合成から、ノイズ干渉および消費電力に及ぶ問題の巨大なスペクトルに対処するに違いありません。高機能(実際にはアナログ)エレクトロニクス。適切なテストバリエーションおよび DFT が設計過程の中で初期に考慮され組込まれる場合のみ、多くの問題は解決することができます。さらに、机上のメソッド教訓、テスト検証の分析への発展、およびテスト設計のより深い統合が必要である。論理 - 回路 - 物理インプリメンテーションでの分析から、また設計検証

からの努力、および結果は(たとえば、流れる、ベクトル、感度)、設計の間に再利用されるに違いありません。メモリ、SoC、AMS および MPU として変えられるような業界を横切った、3 つのハイ・レベルなテストの挑戦が、チップ上の DFT、BIST およびテストビリティの拡張を必要とします。

(1) 高速デバイスインターフェース

構成要素の I/O 速度はシステム実行にとってコア・クロック周波数あるいはトランジスターおよびアーキテクチャ上の実行と同じくらい重要になりました。新しい I/O プロトコルが導入されており、GHz 範囲まで延長されています。これらの I/O スキームは、より速いだけでなくより複雑です。差異でかつ、同時に、Gbit/s で作動する双方向スキーム、差異の電圧で評価する、また供給 Vdd 範囲の 10 分の 1 を揺り動かします。対照的に、ATE および構成要素のテスト遺産は、MHz 範囲にテストして基づいた共通のクロックおよび I/O 測定を含んでいます。従って、I/O 速度およびプロトコルは重要な手段、材料、および ATE 設備、インターフェース・ハードウェア、および設計検証および製造テストの両方のために使用されたテストに対するコスト挑戦を駆り立てます。この変曲点は、特に I/O に対するチップのテストビリティ適応性の開発および適用を要求します。I/O DFT、および loopback のような BIST 方法、ジッタ測定、エッジ検知、偽似乱数のパターン生成などが、検証のために必要になり、これらを製造するために、標準の技術になるでしょう。

(2) 高度に統合された(SoC)設計

ブロックが均質の場合さえ(たとえばすべてのロジック)、より大きな統合デバイスへ向けて、設計ブロックより先に存在する統合は設計ツールの非線形な複雑度を増し、DFT および製造テストを生み出します。ますます、種々様々の回路は統合されています。ロジックおよび SRAM および DRAM は、近年一般に統合されました。しかし、今、アナログ、ミックス・シグナルおよび不揮発性のフラッシュはロジックと RAM と結合しています。LSI の複雑度およびコストは、統合デバイスでは比較の見積りが可能です。しかしながら、組込みブロックおよび混合デバイスタイプは、テストビリティ、設計検証および製造テストコストの高度に非線形と、予測不能を増加します。より大きなロジックデバイス内に全て組込み ASIC または MPU のマクロはすでにこのインパクト(たとえばシリコン・コストを超過する、製造テスト・コスト)を体験しています。DFT でさえ、これらのコストは非線形かもしれません。テスト・データ・ベース再利用を可能にする組込みブロックが、マクロの I/O にアクセスするダイレクト・アクセス DFT(DAT)、しかしさらにテスト・データ・ベース・サイズにより、追加のテスト挿入を要求してもよいでしょう。DAT を備えた組込みブロックの試験はさらに桁違いかもしれませんが、埋め込まれていないバージョンの試験より長いテスト時間。テスト方法、ATE、および SRAM/DRAM、ロジック、フラッシュおよび AMS の LSI のための統合は、ユニークな最適化を備えた根本的に異なる遺産から来ます。従って、組込みフラッシュをテストして、RAM、あるいは統合ロジックデバイス上の AMS は典型的に、これらの最適化を壊します。スタンド・アロンのアナログが RAM デバイスのための、非常に長いテスト時間は、異なる ATE がテスト設備コスト、および SoC あるいは統合デバイスに、利用可能でない並列(SRAM/DRAM)の高い割合によって相殺されます。再び、統合組込み特性は、スタンド・アロンのデバイスで利用可能なブロック I/O へのアクセスを排除するかもしれないし、妨げるかもしれません。拡張した DFT 技術およびプロトコル(たとえば IEEE P1500)だけでなく著しい進歩も、また、BIST の使用および(または)テストのより大きな部分の組込みソフトウェアベースのセルフ・テストが要求されます。

(3) 信頼度スクリーン

テストの製造は歴史を持っています。デバイス実行および機能性を測定しただけでなく、潜在している信頼度欠陥の識別、および分離の必要なビジネス・タスクを実行しました。より明確なこれらは欠陥はデバイスまたは薄膜の信頼度。動的なバーン・イン、Iddq、そしてテストの間の Vdd 電圧ストレスをかけることで、高度な LSI 技術のリーク電流を急速に増加させることにより、すべて厳しく制限された、3 つの本質的なメソッドです。同時に、Vdd を使用する能力は縮小されています。ストレスと Vdd の比率が両方のバーン・インの各技術ノード、加速、

識別および遮る能力とともに、および「ATE の上で」減少するとともに、電圧強調は、非常に制限されたようになります。また非線形のコスト増加をするために、マイクロプロセッサのような高額の商品のためのバーン・イン。同時に、Vdd 電圧ストレス、および Iddq デルタのような高度な Iddq 技術さえも非常に制限されています。近いうちに、著しいコスト増加は過剰殺傷(信号対ノイズ)に起因するかもしれません。私たちの市場信頼度必要条件と歩調を単に合わせるために歩み寄ります。基本的に新しい長期的な解決策が要求され、信頼度測定あるいは特別の信頼度測定用の重要なチップ上のハードウェアを含んでいるかもしれません。結局、チップ上のセルフ・テスト、自己診断、自己修正および自己修復のより広い開発および配備もコストとデバイスの信頼度目標を達成するために要求されるでしょう。ファブレスデバイス産業の継続的な成長は、このための潜在的な挑戦および統合解決策へ別の次元を加えます。たとえばファウンダリが統合テスト戦略に通知し支援することができるという程度まで、彼らは好ましいベンダーになるでしょう。生じる設計テストの困難なチャレンジは、表 17 の中で要約されます。

<i>DIFFICULT CHALLENGES ≥ 65 nm / THROUGH 2007</i>	<i>SUMMARY OF ISSUES</i>
At-speed test with increasing frequencies	Continuation (avoidance) of at-speed functional test with increased clock frequencies (MPU, (ASIC/SoC)) At-speed structure test with increased clock frequencies (MPU, ASIC/SoC) Test and on-chip measurement techniques for multi-gigahertz serial ports (AMS)
Capacity gap between DFT/Test generation/fault grading tools and design complexity	Better EDA tools for advanced (open, delay, etc.) fault models (MPU, ASIC/SoC) DFT to enable low-cost ATE (MPU, ASIC/SoC) Non-intrusive logic BIST (including advanced fault models) (MPU, ASIC/SoC) AMS DFT/BIST, especially at beyond-baseband frequencies (AMS)
Quality and yield impact due to test equipment limits	Power and thermal management during test (MPU, ASIC/SoC) Fault diagnosis and design for diagnosability (MPU, ASIC/SoC) Yield improvement and failure analysis tools and methods (ASIC/SoC)
Signal integrity testability and new fault models	Signal integrity (noise, interference, capacitive/inductive coupling, etc.) testability (MPU, ASIC/SoC) Fault models for analog (parametric) failures (AMS)
SoC test	Integration of SoC test methods onto test equipment platform Integration of multiple fabric-specific test methodologies DFT, BIST and test methods compatible with core-based SoC environment and constraints (AMS) Embedded memory built-in self-diagnosis and self-repair Test reuse
<i>ADDITIONAL DIFFICULT CHALLENGES < 65 nm / BEYOND 2007</i>	
Integrated self-testing for heterogeneous SoCs	Test of multi-gigahertz RF front ends on chip (AMS) Use of on-chip programmable resources for SoC self-test (SoC) Dependence on self-test solutions for SoC with RF, AMS components (SoC) (Analog) signal integrity test issues caused by interference from digital to analog circuitry (AMS) Test methods for heterogeneous SoC including MEMS and EO components (SoC)
Diagnosis and reliability screens	Diagnosis and failure analysis for AMS parts (AMS) Design for efficient and effective burn-in to screen out latent defects (MPU, SoC) Quality and yield impact due to test equipment limits (MPU, ASIC/SoC) New timing-related fault models for defects/noise in nanometer technologies (MPU, ASIC/SoC)
Fault tolerance and on-line testing	DFT and fault tolerant design for logic soft errors (MPU, ASIC/SoC) Logic self-repair using on-chip reconfigurability (SoC) System-level on-line testing (SoC)

表 17 *Design Test Difficult Challenges*

(4) テスト設計における短期的チャレンジ (>65 nm)

1) 増加する周波数を備えたテストで促進してください。

* MPU(ASIC/SoC)のための高速のクロック周波数を備えた *at-Speed* の機能テストの継続(回避)。 MPU については、DFTとBISTが、MPU仕様書未満のデータ割合、および端配置精度のATE上で実行される、*at-Speed* 機能的な試験のために必要です。対照的に、ASIC/SoCは高い人的資源および資本費用

を回避し続けるに違いありません。速度で、追加の故障タイプ、改善された欠点カバレッジ(ツール)、DFT および増加する BIST のコンビネーションによる機能テスト。

- * 増加したクロック周波数を備えた *at-Speed* 構造テスト。DFT は推移およびバス遅延を可能にし続けるに違いありません。スキャンテスト上で、デバイス ATE より、ますます複雑なクロッキングスキームの限界になり、遅くなる。
- * マルチ・ギガヘルツシリアルポートに対する、テストおよびオンチップ測定技術、ギガビット・イーサネットと Firewire のようなプロトコルが採用されます。たとえば略式なシグナリング技術を使用するマルチ GHz の周辺バスを備えた組込みプロセッサ。そのようなアプリケーションは、生成することができる非常に高価なテスト・システムを要求します。また、差異を受け取ることは 100mV ほど低い電圧振動を備えた数 GHz まで信号になります。過度のコストは既存のテスト解決策が大量生産を不可能にします。今日の構成要素の ATE は典型的に、急速にデバイス I/O 速度を発展させて、スキームを示し、大量製造テストには不適当な共通のクロックのマシンです。そのようなシリアル・コミュニケーション・バスおよびコンポーネント用の新しい I/O テストメソッド、DFT およびオンチップ測定技術は必要です。まさに高速のシリアルポートについては、発信機ジッタ、クロス・トーク、信号のロス、レシーバーの PLL が信号流れを自動追跡するという能力、レシーバー・ジッタトレラントなどを含む様々な新しいパラメーターが、テストの挑戦を増加させます。

2) DFT/テスト生成/故障検出率測定ツールと設計複雑度の間のキャパシティー・ギャップ

現在のマイクロプロセッサおよび最大の ASIC の設計データ・ベースは、複雑なパッチ・スキームを強要し、かつ、十分なチップ・フォールト・シミュレーションを分割し、産業 EDA ツールのデータ・サイズ範囲を超過し、ATPG をスキャンします。困難、に、不可能、ちょうどスタック・フォールトのためにさえ、テープアウトに先立って十分なチップ・テスト・スティミュラス/レスポンスデータセットのシミュレーションは、設計フロー中の基本の要求で、修正サイクルと同様に DFT とデバイスのバグ割合の増加を遅くします。代案は実際のシリコン(それは設計反復ループにシリコン処理時間を加える)上のベクトル確認で、劇的にエンジニアリング資源を増加させます。

- * 高度な故障モデル(オープン故障、遅延故障、他)用のよりよい EDA ツール。現実的な欠点のための実利的に使用可能なフォールト・シミュレーションおよび ATPG は複雑なチップをサブ 100nm ノードで設計する場合およびトランジスターレベルで実行をするとともに、必要です。
- * DFT は廉価な ATE を可能にします。DFT は縮小された信号ピン計算、制限のあるピン・エレクトロニクストレラント、縮小されたテスト・データ割合および、より少数の特徴(たとえば APG はない)を備えた ATE 上のすべてのパラメーター面、およびロジックの試験を可能にするために必要です。この要求は I/O、キャッシュおよびロジック DFT を含んでいます。
- * 侵入しないロジック BIST(高度な故障モデルを含んで)。新しい BIST スキームは完全に解読された non に対する、よりよいトレラントおよび偶発事故で必要です。ロジック、制限のある主張の隣接トライステート・バス、また MPU 混載に特有の他の設計要素。開発努力および製造原価の著しい増加のないテストカバレッジおよび内容を拡張するために、ちょうど故障モデリング、フォールト・シミュレーションおよび ATPG カバーするために、ロジック BIST を現実的に拡張しなければなりません。また遷移故障(たとえば、モデル化されて、直接カバーすることは開きません、またブリッジおよび他の故障タイプ)。

ほとんどのミックス・シグナル IC では、アナログ回路類が、小さくまとまっています。しかしながら、生産コストの合計はアナログ・コストによって支配されます。いくつかの情報筋は、もしアナログ試験への変更が行われなければミックス・シグナルの製品原価を合計するアナログ試験コストの比率が増加し続けるだろうと見積ります。アナログおよびミックス・シグナルのコンポーネント用のコスト効率の良い

テストメソッドおよび技術の開発は、従って混載 SoC の試験のための完全な解決を考案する本質的な部分です。

- * **特に高い解像度および(または)高い周波数(ベースバンドの向こうの)のアナログ/ミックス・シグナル DFT/BIST.** オンチップおよび高分解能(たとえば>16 ビット)ADC(Analog-to-Digital Converter)および RF コンポーネントのためにコスト効率が良いテスト技術は必要です。高分解能 ADC が、連続物かマッシュ配置の中で接続している多数の段階で構築されるので、1 つの可能な方向は、DFT かセルフ・テスト戦略を開発する際に、この構造の知識を利用することです。AMS のコンポーネント用の DSP に基づいた DFT および BIST 解決策は、ベースバンドを越えて周波数まで著しく延長されるに違いありません。

3) 質と歩留りはテスト設備性能により衝撃を与えます。

- * **テストの間の消費電力および熱の管理.** エンド・ユーズのシステム・ニーズあるいは不足エンド・ユーズのコントロールおよび管理(熱シンク、気流、フィードバック・ピオス/システム・インテグレーションなど)を統合したことにより、製造テストの消費電力および熱制御はもっと抑制されるかもしれません。コントロールの必要なレベルは、di/dt 緩和用の活発なフィードバックおよび設計のような新しい技術を要求するでしょう。
- * **diagnosability のための故障分析および設計.** ツール、スキームおよび規則はすべての故障タイプには使用可能な自動的な故障分析(ゲートまたは回路に故障する隔離および識別)を可能にするために必要です。
- * **改良と故障の分析ツールおよび方法を歩留りを推定します.** 厳しく売買する短いライフ・タイムおよび Time-to-market と結合した DSM(Deep Sub-micron)プロセスの高いコストは、故障解析プロセスに挑戦します。これらの傾向は、効率的に故障メカニズムを理解し、矯正するアクションを提供し、歩留りを改善するために、プロセス・ランダム異常(干渉/欠陥)が速く探し出され特徴づけられることを要求します。

4) シグナル・インテグリティテストバリエーションおよび新しい故障モデル。

- * **シグナル・インテグリティテストバリエーション(ノイズ/干渉、容量性/誘導のカップリングなど).** シグナル・インテグリティ検証問題がテスト問題(たとえば分配された遅れ変化、クロス・トークに引き起こされた遅れ、ロジック・エラー、過度の電圧、落ちる、または電源配線網上の振動および基板および熱ノイズ)になるので、増加した配線層および配線遅延は新しい故障モデルを要求します。試験はそのようなパラメータの実行故障を目標とするに違いありません。パラメータばらつきによって引き起こされたノイズ欠点の新しいクラスは、電気的であるより高い抽象、回路、およびフォールト・シミュレーション、ATPG および BIST の目的のためのトランジスター・レベルに適切にモデル化されるに違いありません。
- * **アナログ(パラメータ)故障のための故障モデル.** テスト評価、アナログ・ブロックのセルフ・テストテスト設計の解決策および変換器は、引き続き非常に制限されています。ほとんどの既存解決策は、第 1 に機能的な試験に基づき、周波数制限があります。与えられたアナログ・テスト・プログラムを評価するための標準的な指標はありません。アナログセルフ・テストを含むアナログ・テスト技術のテスト品質を評価するための長所の図を開発することは重大です。多くのアナログ故障が、連続変数(時間、電圧、過程など)の中で測定された仕様範囲からのパラメータにより、変化または誤った組合せの製造から発生しています。効率的で有効なアナログ故障検出率測定およびテスト生成のための故障モデルが要求されるでしょう。ミックス・シグナルの設計については、高いシミュレーション精度を維持する間にフォールト・シミュレーションの計算複雑度を最小限にすることができるツールが必要です。

5) SoC テスト.

- * **テスト設備プラットフォーム上への SoC テスト方法の統合.** 多数の挿入、使用、1 つの、より安い、統合ブロック・タイプ(アナログ、RAM、フラッシュ、ロジックなど)について試験デバイスをささげた、一つのプラットフォーム上にすべての測定ハードウェアおよびソフトウェアの統合に対して取り引きすることができ、単一の挿入を使用することができる。プロセスで財産目録とウェーハ上でより長く、より複雑な製造テスト・フロー(多数のプラットフォームおよびステップを備えた)を維持するインパクトも、考慮に入れられるに違いありません。歴史上、追加のハードウェア複雑度の統合は、全面的な ATE 開発および、コストの非線形増加に帰着しました。増加資本コストの償却、1 つの「すべてをします」機械は、異なるブロックの試験が同時に実行されないという点でさらに問題です。従って、他のテストが ATE ハードウェアの他の部分上で実行されるのを待って、多くの場合、大多数の製造に使用する duty cycle に回路特有のテスト・ハードウェアは使用されていません。最小では、組込みブロック(たとえば IEEE P1500 あるいは等価物)のための一貫したブロック・レベル・アクセスが要求されます。結局、非線形の製造テスト・コストは、十分な BIST 解決策に、できるだけ多くのブロックをほかに運用するでしょう。SOI およびシリコン・インテグレーションの経済が失われるでしょう。解決策中の設計技術の焦点は SoC 製品原価縮小では、非常に重要な ROI を提供するだろうが、製品計画の最も初期の段階にアドレスされなければなりません。
- * **多数のファブリックに特有のテストメソッドロジの統合.** 異なる設計/ツールからのブロックの SoC インテグレーションは流れます。また、遺産は厳しく DFT、DFT 統合、ATPG、テスト・データ・ベース検証、およびバック・エンドのベクトル処理ツールおよびフローに挑戦するでしょう。MPU 混載でのように、テープアウトに先立った十分なチップ・テスト・スティミュラス/レスポンスデータセットをシミュレートする能力は、基本の要求です。
- * **コアに基づいた SoC 環境および制約と互換性をもつ DFT、BIST およびテスト方法.** 補足的なコンポーネント用のセルフ・テスト技術はアナログ/RF DFT のために開発されているに違いありません。トランシーバー(RF、オプティカル、ADC/DAC、I/O など)の混載内では、送信器およびレシーバーは互いにテストするためにしばしば使用することができます。たとえば、スティミュラス生成およびレスポンス分析の両方の DSP に基づいた(デジタル)技術は、信号を生成し、かつレスポンスを分析するために組込みプロセッサか DSP コアとともにオンチップ DAC および ADC を使用します。潜在的な利点は低いハードウェア・オーバーヘッド、デジタル ATE の低機能侵入、より高い柔軟性および使用を含んでいます。この技術をより高い周波数でデバイスを扱うために進めなければなりません。
- * **組込みメモリ(DRAM、SDRAM、フラッシュなど)内蔵の自己診断および自己修復.** 組込みメモリに対するこの適応性は、セルフ・テスト適応性、余分の要素(余分なカラム、列あるいはブロック)、余剰分析、および置換ロジックおよびハードウェアを含む著しいチップ上の特徴、およびロジックを要求します。
- * **テスト再利用.** SoC 設計生産力は、テスト・データ・サイズ、および予期された等価なテスト時間に関する ATE 境界条件(制約)内に、統合製品上のテスト IP かテスト・データセット再利用を要求します。たとえば、それらが既存の ATE の上で適合しないので、統合製品のテスト・データセットは追加の挿入を駆り立てるかもしれません。組込みブロックをテストするための DFT モードはそのようなブロックのスタンド・アロンのバージョン用のものより等価か、より速いテスト時間を提供するべきです。

(5) テスト設計の長期的チャレンジ (< 65nm)

1) 混載 SoC のための統合セルフ・テスト

- * チップ上のマルチ・ギガヘルツ RF フロント・エンドのテスト。RF コンポーネントの製造テスト工程には、多大な測定および工数を必要とします。テストの再現性改善、時間短縮、測定精度向上により、問題解析コストを低下させるブレークスルーが必要です。
- * SoC のセルフ・テストのために、チップに搭載したプログラマブル・リソースを使用します。テストのコスト削減には、チップに搭載した MPU、プロセッサ、等のプログラマブル・リソースを用い、STUMPS スタイルのロジック BIST の拡張したセルフ・テスト機能を用います。長期的にはプロセス/パラメータばらつきや、遅延/タイミングによる故障に対するテストが必要となりますが、本セルフ・テスト機能を用いて必要な品質レベルを維持します。
- * (RF およびアナログを含む) SoC のためのセルフ・テスト。ATE 支援によるセルフ・テストまたは BIST が浸透すると考えられます。
- * デジタルからアナログ回路に与える干渉を原因とする(アナログ)シグナル・インテグリティ・テスト。同一チップ上のデジタル回路による干渉が、アナログ回路の動作を混乱させる場合があります。干渉による誤動作を分析するには、デジタル回路を実スピードで実行させ、アナログ回路をテストすることが必要です。2Gbit/秒以上での動作では、クロストークの防止およびテストを実現するメソッドロジが必要となります。
- * MEMS およびエレクトロ・オプティカルコンポーネントを含む混載 SoC のためのテスト方法。新しいタイプの回路を SoC に搭載するには、既存の設計および DFT のフレームワークに適合させる必要があり、セルフ・テスト機能を拡張させる必要があります。短期チャレンジとして、RF、ミックス・シグナル、フラッシュ等の統合に同様なテスト手法が必要であるのと同じ(経済的)理由です。

*

2) 分析と信頼性スクリーニング。

- * アナログ/ミックス・シグナル部分の分析および故障解析。アナログ/ミックス・シグナル部分の分析および故障解析は、現状では大部分を人手で実施しています。アナログ/ミックス・シグナル・コンポーネントのための新しい故障分析および故障解析技術を開発し、デバッグの時間およびコストを削減する必要があります。設計、プロセス、テストの各データに対して、適切な故障モデルおよび評価指標を用いて、相関性を分析するツールおよび技術が必要となります。
- * 潜在欠陥をスクリーン・アウトするための効率的で有効なバーンインを設計で組込む。潜在欠陥を許容可能レベルとするために、バーン・イン容易化設計または新代替スキーム(例: 自己修復)により、バーン・イン時の熱流出を緩和すべきです。
- * テスト設備性能限界による品質と歩留りに対するインパクト。ATE およびテストに使用する計測装置の許容誤差は、パラメータの実測定内にあるべきです。根本的に新しいアプローチ無しでは、信号対ノイズ比および誤検査による歩留り低下が承諾しがたいレベルに達するでしょう。
- * ナノメーター技術において欠陥/ノイズで生じるタイミング関連故障に対する新モデル。パラメータのばらつきや欠陥が引き起こすタイミング・レベルでの故障動作は、ますます顕著になります。タイミング・レベルでの故障動作をモデル化し、フォールト・シミュレーション、ATPG、DFT、BIST ツールおよびメソッドロジに組込むべきです。

3) **フォールト・トレラントおよびオン・ライン試験。**

- * *ロジック内でのソフト・エラーのための DFT およびフォールト・トレラント設計。* ロジック用のソフト・エラー緩和は、現状では制限のある手法しか存在しないため、回路およびアーキテクチャ上の解決策が必要です。
- * *チップに搭載したリ・コンフィグurable回路を使用したロジックの自己修復。* システム・レベルでの信頼度を向上させ、増加するソフト・エラーの影響を緩和緩和するために、セルフ・テスト、自己診断、自己修正および自己修理に開発工数(新方式および既知の技術の統合)を投じる必要があります。
- * *システム・レベルのオンライン・テスト。* SoC のサブブロックが回路固有のセルフ・テスト方法による構造的試験に移るに従い、機能テストは、ATE 上での機能試験から、アプリケーションに対応したテスターによる試験に移行するでしょう。移行を促す要因は、2 つあります。まず第 1 に、機能テスト用に作成するデータが指数関数的に増大するため、開発コストが非現実的になりました。つぎに、システム・プラットフォームがきわめて複雑になるのに伴い、システムの品質および信頼度に対する期待は、SoC のアプリケーション自体を用いたテストを最適解とするためです。即ち、アプリケーションで顧客が使用するシステム自体を用いたテストです。

<i>Year</i>	<i>2001</i>	<i>2002</i>	<i>2003</i>	<i>2004</i>	<i>2005</i>	<i>2006</i>	<i>2007</i>	<i>2010</i>	<i>2013</i>	<i>2016</i>	<i>Driver</i>
<i>MPU ½ Pitch</i>	<i>130</i>	<i>115</i>	<i>100</i>	<i>90</i>	<i>80</i>	<i>70</i>	<i>65</i>	<i>45</i>	<i>32</i>	<i>22</i>	
<i>SoC new design cycle (months)</i>	<i>12</i>	<i>12</i>	<i>12</i>	<i>12</i>	<i>12</i>	<i>12</i>	<i>11</i>	<i>11</i>	<i>10</i>	<i>9</i>	<i>SoC</i>
<i>SoC logic Mtx per designer-year (10-person team)</i>	<i>1.2y</i>			<i>2.6y</i>			<i>5.9r</i>	<i>13.5r</i>	<i>37.4r</i>	<i>117.3r</i>	<i>SoC</i>
<i>SoC dynamic power reduction beyond scaling (X)</i>	<i>0</i>			<i>1.5</i>			<i>2.5y</i>	<i>4y</i>	<i>7r</i>	<i>20r</i>	<i>SoC</i>
<i>SoC standby power reduction beyond scaling (X)</i>	<i>2</i>			<i>6y</i>			<i>15r</i>	<i>30r</i>	<i>150r</i>	<i>800r</i>	<i>SoC</i>
<i>%Test covered by BIST</i>	<i>20</i>			<i>30y</i>			<i>45r</i>	<i>60r</i>	<i>75r</i>	<i>90r</i>	<i>MPU, SoC</i>

表 18 Additional Design Technology Requirements

Solutions Exist Solutions Being Pursued No Known Solutions

Appendix: DT のコストと価値

半導体製品実現における DT の貢献は以下の分析により示せます。2001 年の ITRS Design-ITWG からの依頼により、ガートナー/データクエストが 1990 年(いわゆる「RTL メソドロジ」が起こった年)の設計生産量を測定した所、4K ゲート(=16K トランジスタ)/年でした。さらに、1990 年以降の主要 7 件の DT 改革による設計生産力改良を計測しました。主要 DT 改革とは、次のとおりです:内製の配置配線(1993 年、生産力は+38.9%向上し、5.55K ゲート/年となる)、「tall-thin エンジニア」(1995 年、+63.6%向上、9.09K ゲート)、小規模ブロック(2,500 - 74,999 ゲート)の再利用(1997 年、+38.9%の改良、56 K ゲート)、大規模ブロック (75,000 - 1Mゲート) の再利用(1999 年、+340%向上、40K ゲート)、IC インプリメンテーション・スイート(2001 年、+63.6%向上、91 K ゲート/年)、「インテリジェント・テストベンチ」(2003 年、+37.5%向上、125K ゲート)、「エレクトリックシステム・レベル(ES レベル)メソドロジ」(2005 年、+60 向上、200K ゲート)。DT 改革の定義は以下のとおりです。(1) IC インプリメンテーション・スイートは、RTL 合成から IC 配置配線、GDSII 出力までの IC 設計をカバーする、密統合したツール・セットです。(2) インテリジェント・テスト・ベンチは、RTL 自動検証ツール(「コックピット」)であり、ES レベルの設計記述を入力として検証可能なブロックに分割し、各ブロックに対して、コード・カバレッジを追跡・報告しつつ、各種の検証ツールを実行します。(3) ES レベルは RT レベルの 1 階層上にあり、ハードウェアおよびソフトウェア設計を含んでいます。ES レベルは、ビヘイビア(システム機能が分割されていない場合)と、アーキテクチャ(ハードウェアとソフトウェアに分割され、各設計チームにハンド・オフされている場合)の 2 サブレベルから成ります。

図 1 は、システム・ドライバー章で定義したドライバであるロー・パワー SoC(SoC-LP) PDA 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ます。モデルでは、過去の実績からエンジニア・コス

トは5%/年で上昇するとし(初年である1990年における給与および諸経費は181,568ドルとした)、EDA ツール関連のコストは3.9%/年で上昇するとしました(初年である1990年におけるエンジニア1名あたりのEDA ツールコストは、99,301ドルとした)。1M ロジック・ゲートあたりのデザイナー数は、1990年において250名であり、2001年では11名です(1人年あたりのロジック・ゲート数は、32.8%で向上するとした)。ロー・パワーSoC のPDA モデル(システム・ドライバー章)は、2001年で3Mのロジック・ゲート数であることから、SoC PDA 設計コスト(デザイナー+ツール)は1510万ドルとなる。1993年から2001年の間に実現した5つのDT革新なしでは、2001年の同じSoCのための設計コストがおよそ3億4240万ドルになります。日本の半導体技術ロードマップWG1(STRJ-WG1)が開発しシステム・ドライバー章に引用したもう一つの設計生産性指標を用いるならば、ギャップはより大きくなります。本指標では、1999年の新規(再利用)ロジックの生産力は360k(720k)ゲート/人年となり、データクエストによる評価値の6倍(12倍)です。

私たちは、継続的なDT革新の供給なしでは、設計コストが非現実レベルまで増加し、付加価値のないデザインしか設計できなくなる、と結論を下します。