

デザイン

概要

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。

2003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。本章の読者には2001年版ITRSデザイン章を参照することを推奨する。2001年版デザイン章は現在でも有益な内容を多く含む。

2003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、たとえばMPU(Micro Processing Unit), DSP(Digital Signal Processor)などのプログラム可能な商品上に実現するか、あるいは個別ICを新規開発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100万ドルに達する。さらに設計NREとして、設計ミスによるリスピ(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利益構造により、巨額の半導体投資サイクルが運営されている。2001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。図13は、システムドライバ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ている。2003年におよそ2000万ドルの設計コストが、1993年から2003年までの設計技術の改善が行われなかった場合には、6億3000万ドルに達していたことが分かる。(分析の詳細は付録を参照)

(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多く必要となる。2003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。

2 デザイン

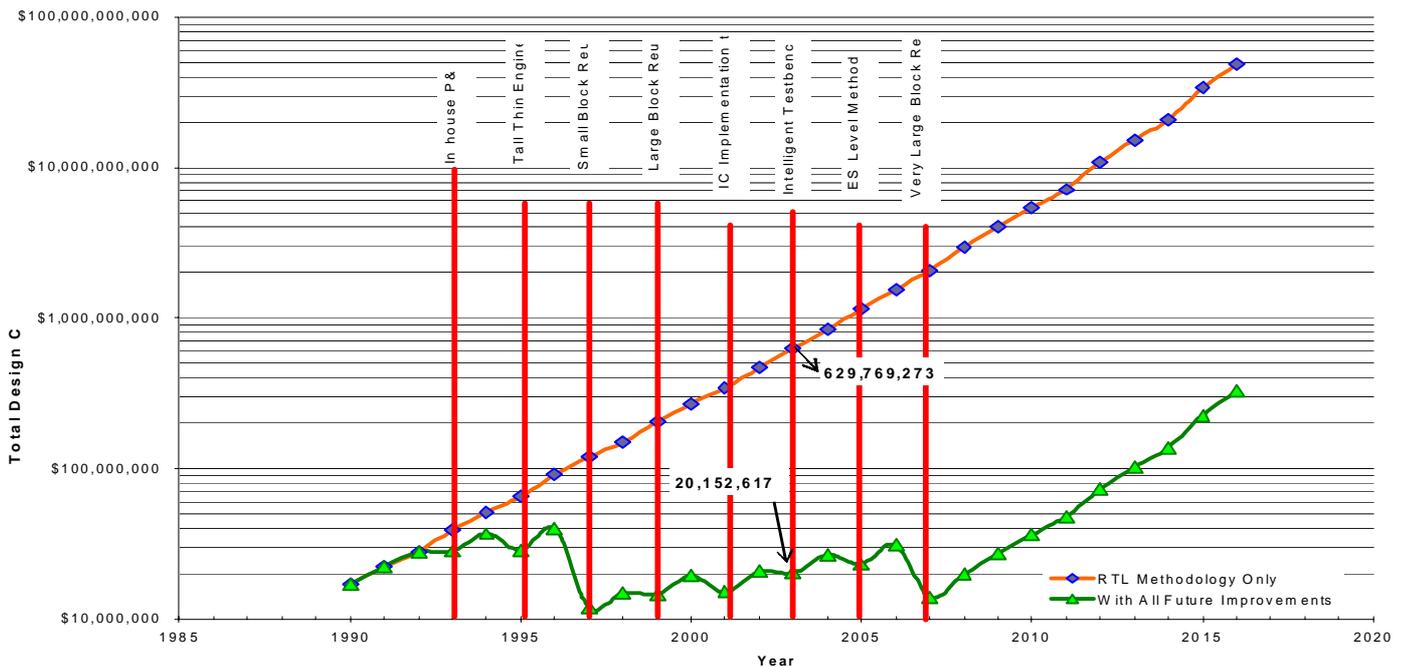


図 13 Impact of Design Technology on SOC LP-PDA Implementation Cost

本章では、始め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、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ある。本章の大部分は従来からの技術分類(図 14)に基づいて技術課題を分析する。即ち、設計工程、システムレベル設計、論理/回路/物理設計、設計検証およびテスト設計を分析する。¹ これらの技術課題については EDA(Electronic Design Automation)業界のマネジメント、開発部門および大学研究機関といった対応組織ごとに述べられる。さらにシステムドライバである MPU、SOC、AMS(Analog Mixed Signal)、メモリと分野別に課題を分析する。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。これは今後さらに重要性が高まる特定用途向け、特定システムドライバ向け設計技術の例示となる。将来の ITRS デザイン章は旧来の EDA 技術分野別ではなく、これを使用する用途向け、システムドライバ向け設計技術に重点を置いた構成となる。

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、いかに最適化するか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な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。さらに設計技術ロードマップは ITRS 技術ノードへの関連性は高くない。ITRS 技術ノ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や設計の質的改善をもたらす。

¹ 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。テスト章ではさらに 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。

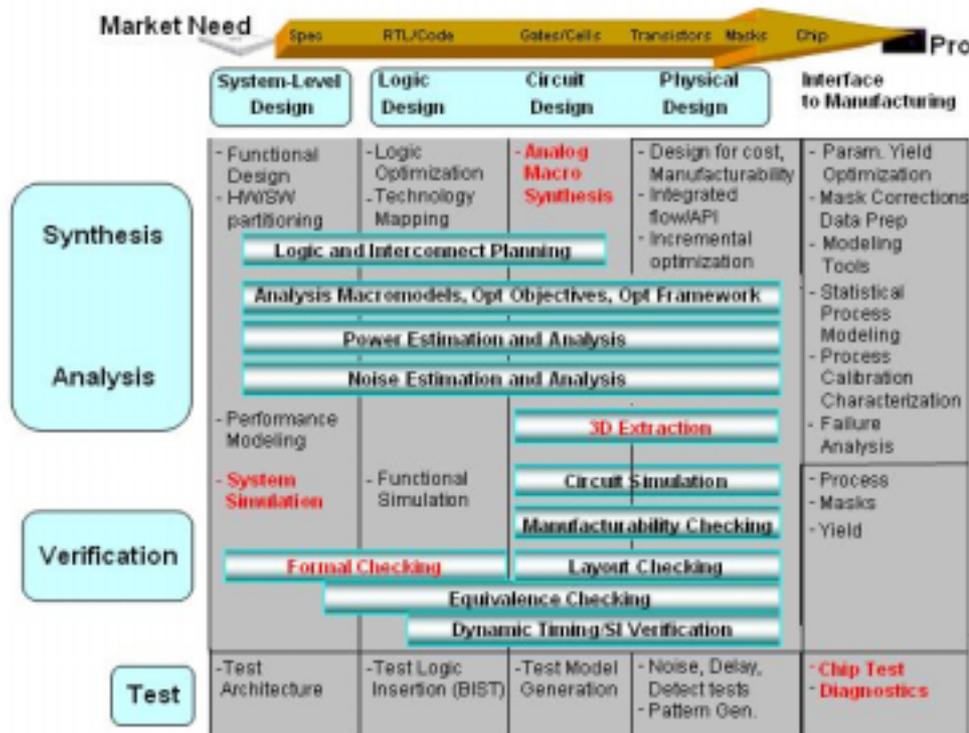


図 14 Landscape of Design Technology

複雑度とクロスカット課題

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、歩留まり、リーク電力)
- * 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)
- * 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)、さらに
- * 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)。

4 デザイン

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²。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、および連続的なプロセス改良)

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、1) 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、2) 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、3) 組込みソフトウェア設計や製造インター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

² 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.Claasen, “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(ボラックの法則)と述べている。

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—生産力、消費電力、製造力、干渉およびエラートレラント—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る。システムと設計プロセス複雑度と密接に関連する、第1のクロスカットの課題は最も大きくかつ重要である。第2から第5の課題は、第1の課題と比べて考慮の範囲は狭く、主にシリコン複雑度の問題に取り組む。

クロスカット・課題 1—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、組み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ド・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。

クロスカット・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示す。1) 総括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表 12 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みなければならない。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。

クロスカット・課題 3—製造技術との融合

「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)、フロントエンドの処理、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、より知的なインターフェースにより、削減される可能性がある。

クロスカット・課題 4—干渉

グローバル配線のスケーリングですでに試みられている、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・メソドロジーは、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。これらのメソドロジーは、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ドロジーが要求される。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。

クロスカット・課題 5—エラートレナント

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)。さらに、バーンインオープン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、設計技術の 5 つの伝統的な領域における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、MPU と SOC にマッピングされる。それにもかかわらず、(1)アプリケーションおよびドライバ特有の設計技術に対するトレンド；(2)設計生産性と品質がスケーリングされていくなかでの AMS(アナログ・ミックスドシグナル)特有の設計技術の限界；(3)システムドライバに従って編集される ITRS のデザイン章において将来考える構成；を強調するため、我々は AMS 特有の設計技術のトレンドと課題を概観することから始める。また、これらの AMS 特有の課題は、要約である表 14 から表 18 の中に含まれている。

アナログ、ミックスドシグナル、RF 特有の設計技術のトレンドと課題

アナログおよび RF 回路は、デジタル回路と比べて異なる。これらは、2 状態(“high”(Vdd – ノイズ許容値)と“low”(Vss + ノイズ許容値))で決められた電圧レベルにより、決められた離散的な時間的期間(例えば、クロック信号が High)で表現されるような、定量化された情報を取り扱わない。アナログと RF 回路で処理された信号は、非常に高い精度で(または、時間的および振幅的により小さい許容誤差で)、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い利得により、これらのノイズ源を高いレベルで抑圧する能力が元々備わっている。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ため、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単なる事実が、これらの問題をアナログ・ドメインではるかに挑戦的かつ簡単ではないものになっている。

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でコンパクトなモデリング、キャラクタライズ、特性抽出の必要性を牽引してきた。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、レイアウトでさえ)においてより高いレベルの抽象化のための単純な規則による特性抽出を非常に困難なものとしている。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チップ当たりのデジタル用トランジスタ数と比較して非常にゆっくりとしているため、これらのすべての境界条件がアナログと 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」節にて主に網羅される。今日、アナログと RF 設計において一般的に知られていたものと同様の問題が、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 産業によるアナログと RF に特化した新しい課題が現れている。

システムドライバ章で述べたように、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

8 デザイン

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、そして、シグナル・インテグリティを含む; (2)限られた人数のアナログ設計者による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)“製造容易化設計”が必要とされるような歩留まり向上; (4)ほとんどすべての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)電子部品と非電子部品(例えば、微小電気機械システム(MEMS)など)の高密度な集積化。これらの課題は、後述する伝統的な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。

システムレベル設計において、クリティカルな AMS の課題は、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と合成である。統合化されるその他の技術(デジタル、ソフトウェア)と共に、システム全体の機能とインターフェースの分析を可能とするような、言語によるモデル化手法に従った自動化されたアナログ回路合成と最適化が必要とされている。問題点は次のものである: シミュレーションにおいて、幅広いタイムスケール(GHz 単位で 100 秒までのアナログ周波数に対して、GHz 単位で 1 秒のデジタル周波数)を変化しながら対処すること; 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。

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を要求している。システムレベル設計と共通の技術的ニーズは、再利用可能で、再目標設定可能なアナログ IP ジェネレータである。特定のクラスの回路(PLL、オペアンプ、パワーアンプなど)用に専用化された現在の自動回路合成は、より一般的な技術によって拡張されることになるだろう。自動レイアウト合成は、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の交差接続させたレイアウトなど)のニーズに対処するようになるだろう。アナログ・ポストレイアウト・シミュレーションでは、インパクト・イオン化や熱的非線形性やローパス・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ことになるだろう。また、合成(技術)は、アナログーデ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のための将来的な対応を取り扱うことになるだろう。短期的には、光インターフェース回路や Q 値の高いオンチップ・インダクター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。長期的に興味深い回路のタイプとしては、超低電力なセンシングとセンサー・インターフェース回路、そして、微小光学(ビーム・ステアリング)素子である。

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。絶え間ない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よりコンパクトなモデル、そして、新しい合否判定基準を取り込むことになるだろう。また、AMS 設計は、混在システム検証という長期的な問題—この分野はいまだ未成熟である—を短期的な問題とする。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可能とするあらゆるサポートを提供して、より強力な技術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。

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なものである。オンチップ化された高分解能(>14-16 bits)ADC と高速(>1-5GHz)RF コンポーネント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の回路もしくは MASH 構造で構成されているため、1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供給電圧で動作し、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が、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。さらに、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、以下の項目のためのツールと新しい記述言語を含んでいる。

- システムの探求
- 回路合成とサイジング
- 回路図の確認
- design for manufacturing
- アナログ/RF レイアウト合成
- 寄生効果の抽出、モデル化、およびシミュレーション
- アナログ IP の回路情報要約化と再利用

次の表は、2007 年までの間に予想される短期的な AMS 設計技術でのブレークスルーをまとめたものである。また、この読者は、2002 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かつ詳細な議論も参照してほしい。

表 13 Near-term Breakthroughs in Design Technology for AMS

Field Of Breakthrough	2003	2004-2005	2006-2007
Specification, validation	Mixed-signal description languages	Multi-language support	Complete specification-driven design flow
Architectural design	Algorithm-oriented design	Language-based performance evaluation	Synthesizable AMS description
Mixed A/D and RF physical design	Procedural layout generation	Design centering, performance estimation	Constraint-driven synthesis: behavior to layout
Parasitic extraction, modeling, simulation	EMI simulation	2D/3D modeling, order reduction	Fault-tolerant circuit architectures, robustness

デザインプロセス

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入力予想通りに製造可能な製品となるような有効なメソッドを必要とする³。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

³ 設計メソッド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」ようにつくる一連のステップである。設計メソッド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。

10 デザイン

設計メソッドはしばしば無視されている。各技術ノード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あればならない。さらに難しいのは、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 1999 年の ITRS でも述べられており、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。図 15 はこの傾向を表したものである。左の列は 1990 年代中ごろの典型的な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行っていた。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、メタルフィル、その他) が焼き付けられるウエ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の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、業界標準のインターフェースを通して行うべきである。最後に、設計サイクルの全ての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図を実現するために、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。図 15 は製造容易性の影響、たとえば先の配線ツールは重要部位の歩留ま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、イールドドリブンレイアウト (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 ノード以降では、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

でもイールドドリブンあるいはコストドリブ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一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。

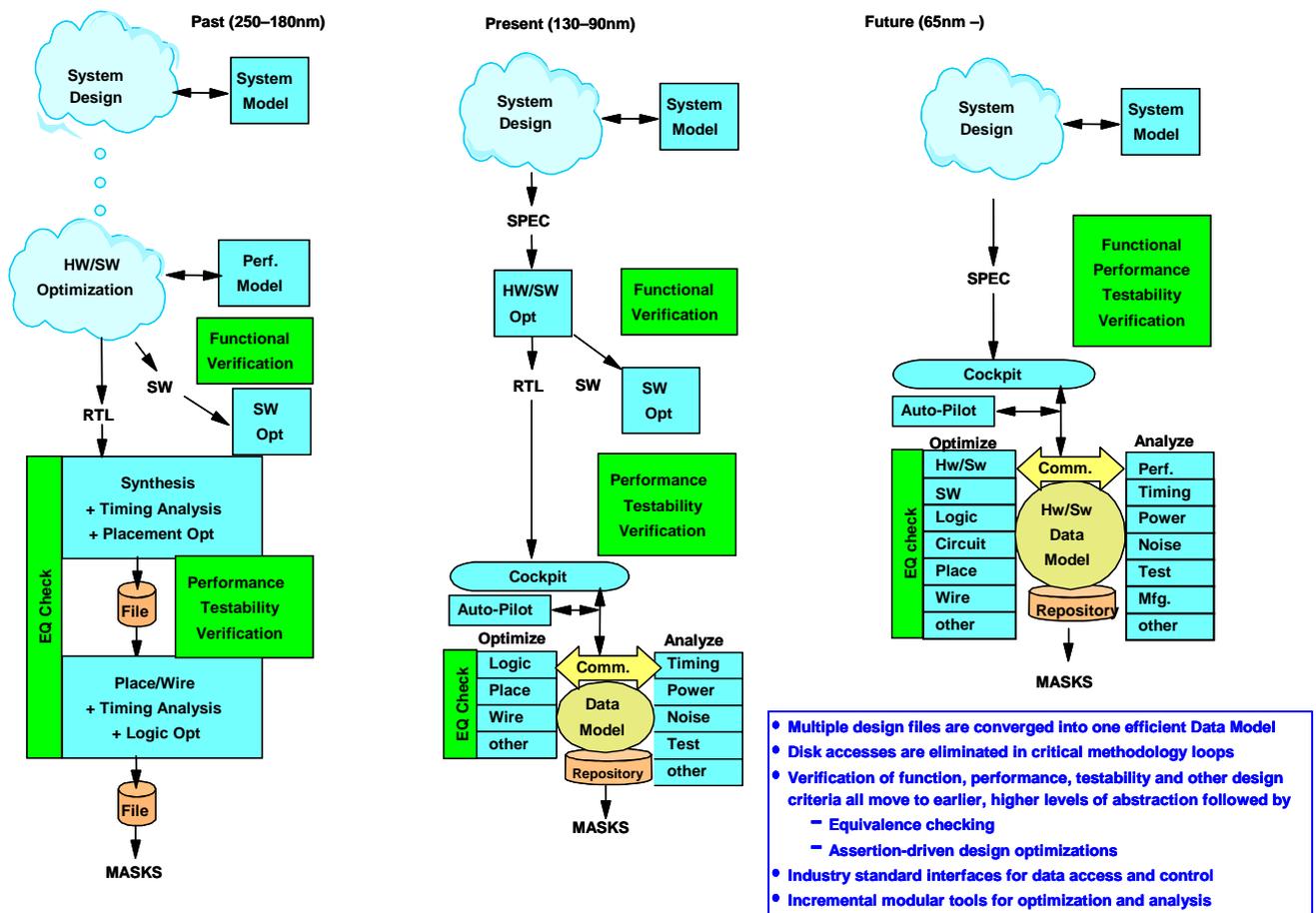


図15 Required Evolution of Design System Architecture

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。図15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RT 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⁴があらわれる可能性が大きいですが、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検証を必要とした。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さら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 RT-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。4) 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、RT レベルでの設計のテスタビリティに高い信頼を与えることになった。非常に早期の設計段階でシステムレベルの抽象度での検証を提供する為、同様な進化が必要とされている。

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一の目的として、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。この動向も図 15 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。現在の技術ノードでは、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、制約ドリブん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、図15の進化を具体的なフローで示すとともに、述べることとする。

⁴ 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。

表 14 Design Process Challenges

Challenges ≥ 50 nm/Through 2009	Summary Of Issues
シリコン複雑度: デバイスと配線	All—リークによる消費電力の指数関数的増大 S, P, A—電力密度と分配 All—テクノロジーとライブラリのキャラクタライズ S, P, A—高周波雑音解析 S, P, A—伝送線路配線 All—eDRAM, eFPGA, SiGe, オプティカル, MEMS
システム複雑度: 状態数, 設計多様性	S, P—急増する状態数を持つシステムの検証 S, P, A—多因子解析と最適化の同時実行 S, P—スケーラブルなアルゴリズム S, A—アナログ・デジタル混載デザインの設計とテスト All—複雑なパッケージ解析 S, A—A/D ツールの統合
設計生産性	S—サードパーティのコンポーネントの統合 S, P—設計ツールの相互運用性 S, P—早期解析検証方法
Time-to-Market	S, A—プラットフォームに基づいた設計 S, P, A—並列プロセッシングの徹底的活用
50 nm 以降の課題 2009 年以降	
製造容易性	All—チップ間変動 All—波長以下のマスク補正 (AltPSM, OPC, RET) All—歩留まりと製造容易性のための設計 All—設計・製造データ共有の標準化
システムレベル設計	S—ハードウェア/ソフトウェア(HW/SW) 共通表現 S—ソフトウェア合成, ハードウェア/ソフトウェア最適化

この表は上記の 4 つの動向が示唆するデザインプロセスの進歩に対する課題をまとめたものである。各課題は最も関連するシステムドライバのリストでラベルをつけられている(S—システムオンチップ、P—マイクロプロセッサ、A—アナログ/ミックスドシグナル、M—メモリ)。

表 14 は上記の 4 つの動向が示唆するデザインプロセスの進歩に対する課題をまとめたものである。各課題は最も関連するシステムドライバのリストでラベルをつけられている(S—SOC、P—MPU、A—AMS、M—メモリ)。このセクションの残りはこれらの解説である。

短期(>50nm)のデザインプロセスの課題

1. シリコン複雑度—これは上記の「シリコン複雑度」の課題、および 2～5 番目の横断的な課題のことである。解決策候補は、デザインルールの制限(レイアウト接地ルールのような)や、解析の継続的な改善(基板カップリング、クロストークがおこすタイミング、IR ドロップ、他)を伴う。

2. システム複雑度—解決策候補のアプローチには、増大するキャパシティ、階層的な手法および、より高位の抽象レベルが必要である。ツールの統合により、非生産的なデータ変換時間やデータの冗長を最小に抑え、分散あるいは SMP プラットフォーム上で同時実行できる合成・分析ツールのアーキテクチャをサポートする必要がある(ダイ・パッケージ・ボードの協調設計、製造容易性の最適化、および電力/ノイズ管理などのため)。基

14 デザイン

本となるアルゴリズムはスケラブルであるべきで、そうすれば指数関数的に増大するインスタンスのサイズであっても現在のコンピュータハードウェア上で効率的に解くことができる。「訂正による構築」型では、繰り返し間で変化する設計量に比例するだけの実行時間での逐次近似法、増分分析、最適化が必要となる。同時に、サイズやピッチの縮小は増加する周波数とともに、遅延、ノイズ/干渉、電力、信頼性の解析モデルにおいて、詳細さと正確性のレベルを上げることが要求される。すなわち、より広範囲の最適化と解析は製造プロセスの特性へのより深い洞察が必要となる。そのような製造プロセスのキャラクタライズは、標準的なデバイス技術や、テクノロジーやプロセスの所有者の管理下にあるプロセス特有のモデルを通して提供されるべきである。このようなモデルの一貫した解釈は、そのモデルを使う設計技術にわたって必要である。最後に、デザインプロセスの全てのレベルにおける相補的な要求事項は、解析の抽象化とデザインプロセスの上流段階での最適化対象としてのマクロ・モデルの特定である。共通のデータモデルとデータベースがデザインフローにおいて可能な限り、機能しなければならない(RTL から製造データ生成まで)。

3. 生産性—設計者の生産性はマルチレベルの自動化、再利用の増加、そして、「選択からの自由」を通して向上するであろう。この方向に設計技術をシステムティックに進化させるためには、設計はサイエンスではなくアートである、という概念は追い払い、「改善のために評価する」という態度をとるべきである。継続的なデザインプロセスと設計生産性の向上のためのメソッドを開発するためには、短期の技術ニーズは品質と予測性に関してデザインプロセスの基準を設けるところから始まる。これらの基準は次には設計品質とデザインプロセスの基準に依存する。品質の基準は設計技術によって可能な製造能力が完全に利用されているかの程度を図るものである⁵。予測性の基準は、ノイズのないアルゴリズム、受容られる解の基準に関して予測可能で安定したツールとフローに向けての定量的な進歩を可能にする⁶。関連するニーズはキャリブレーションとベンチマーキングのためのインフラストラクチャを含む。加えて、トップダウン設計や迅速な設計空間探索には、ツールのスイートスポット(使用できる範囲)、インスタンス特有のツールの実行時間パラメータ、およびツールと設計インスタンス両方のパラメータ化されたモデルに基づいたツールの出力解の品質を推定するものが必要である。結局、アプリケーションまたはドライバ特有のシリコン・インプリメンテーションの予測可能なモデルが必要である。最後に、相補的なイノベーションは設計技術の生産性を測定し、向上させるべきである⁷。

今日では、デザインフローの領域は、業界標準の順次編成ファイルを介して統合される緩やかに結合されたツールから、閉鎖的/独自のインターフェースを備えた緊密な統合システムへと変化している。ハイエンドの設計プロジェクトでは、一つの設計技術ベンダどころか、全ての設計技術ベンダを集めても、必要となる設計技術の全ての範囲を提供することはできない。ゆえに、社内開発された設計技術が、マルチベンダの設計フローを補強しなければならない。ハイエンドの設計には、また、大規模な、幾つもの会社からの地理的にも分散した設計チームが関与することがある。これには設計技術ソリューションにわたってデザインを共有し、再利用することが必要となる。このためのニーズは、1) ばらばらの設計技術間、かつ設計/製造間にわたり、リアルタイムで設計記述を共有するような業界標準の方法、2) すべての設計技術ソリューションにわたってこの方法を業界

⁵ 明らかな基準には、プロセスのジオメトリと供給電圧に関してスピード、消費電力、密度を含む。再利用性、テスト容易性、検証カバレッジ、その他は、設計・製造技術の投資のROIに影響するのであるから、設計品質の基準を決める上で考えなければならない。

⁶ ツールの固有ノイズとは、ツールの入力への機能と関係のない変化(例えば、変数やインスタンスのリネーム、ゲートレベルのネットリストの配線の順序変更など)から生じる解の品質のばらつきである。最近の文献の測定によると、市販の配置配線ツールの固有ノイズは30%にも上るといふ;このようなレベルでは正確な推定能力の開発を阻害するものであり、抑えなくてはならない。

⁷ 幾つかの最近の研究によれば、設計技術の研究開発者は(学術分野を含め)全世界で約6000人である。これらのリソースが注意深く最大のインパクトを与えるように向けられるだけでなく、設計技術の生産性(すなわちテクノロジーの実施)を向上させなくてはならない。この方向への例に挙げた手段は設計生産性への設計技術のインパクトの理解と、設計技術によるIPの広範囲な再利用を含む。

で採用すること、を含む。現在推定(データクエスト、2001)⁸されているところでは全設計コストの 30%に相当するインテグレーションのためのコストを下げることで、そして、それらのコストを素のツールとマシンのコストを超えないようにしながら、シーケンシャルな設計技術データ交換の代わりにリアルタイムでの設計技術の相互運用性を確保しなければならない。インテグレーションコストを減らすとともに、ツールの再利用性と相互運用性を向上させることが、長期的には複合的なプラットフォームとアプリケーション(ワイアレス、インターネットワーキング、など)に対する高品質のインプリメンテーション・フローを可能にするであろう。

4. タイム・トゥー・マーケット商品化までの時間を削減する明白なアプローチには、再プログラマブルで構造化された ASIC 構成の使用、前もって設計されたコアの再利用と「プラットフォーム」アーキテクチャ、そして「広範囲な自動化」があげられる。多くの急成長する半導体市場での最大の要求は、低コスト、どちらかという低性能で、商品化までが早いデザインである。これらの市場での製品の典型は、頻繁に再利用される SOC である。この要求にこたえるには、効率的な設計最適化ができる抽象レベルでの再利用可能なコアの特性を記述する共通の情報モデルが必要である。新しいツールに求められるのは、再利用中心の設計空間探索の「見つけて試す」スタイルと設計最適化をサポートすることである。従えば再利用性を保証するデザインルールの開発と公式化は、そのルールをサポートして強制するような、組み合わせられた設計・解析ソフトウェアを必要とする。異種の構成やテクノロジーがますます単一の SOC や SIP に統合されるにつれ、新しい設計フローは、単一のチップ上に、ハードウェア、ソフトウェア、デジタル、そして多分、アナログ、MEMS、そしてメモリを 1 チップ上に効果的に統合できるものでなくてはならない。すでに述べたように AMS 設計は、デジタル RTL ベースの合成に匹敵する効率と品質でアナログ(そして、長期的には混載技術)設計を合成する能力の改善が求められているボトルネックである。長期的なニーズとしては、さらに高位の機能、性能、製造容易性の検証と同様に、システム実装コストを全体的に最小にするサプライチェーンマネジメントとデザインプロセスがある。EDA のモデルが変わることにより、特定のプラットフォーム、アプリケーション領域、さらには個別のデザインまでもが、相互運用可能なコンポーネンツからなる、カスタマイズされた、デザイン特有の設計フローの構築を必要とするようになるかもしれない。このことは、業界としての EDA の焦点が、完全なフローをつくるのではなく、標準化されたインターフェースを持つ構成部品を作ることに移ることになるであろう⁹。

長期(<50nm)のデザインプロセスの課題

1. 製造容易性—これは製造インテグレーションのクロスカット・課題と製造しやすい設計への動向に関して上記で記した問題を参照のこと。解決策候補のアプローチには、制限されたデザインルール同様、設計と製造容易性の統合的な最適化が含まれる。

2. システムレベル設計—65nm 以下では、システムレベル設計への移行が、HW-SW 設計の統合とともに避けられない。次のセクションでは、システムレベル設計技術の要件を詳細に述べる。

⁸ 全体の設計 NRE コストは(1)設計者と CAD サポートエンジニアの時間当たり給与×設計時間、(2)時間当たりのツールライセンスコスト×設計時間、(3)時間当たりの機械及び他の施設利用費かける設計時間、からなる。「相互運用性コスト」は増大する設計時間(フローの中の余分な変換時間、相互運用性の欠如によって起こされるエラーによる余分な設計の繰り返し)、増大する CAD サポートの給与、増大する施設コスト(データストレージおよびデータマネジメント)に反映される。設計 NRE コストというのは、設計 TAT と時間当たりのリソースコストの関数であるが、デザインの価値、これはタイム・トゥー・マーケット(であるので設計 TAT)と設計品質の関数、と取り違えてはいけない。設計コストとデザイン価値の両方からの知識によってのみ、設計技術の ROI は決定できる。Appendix の設計コスト分析参照。

⁹ プログラマブルなプラットフォームと組み込みソフトウェア中心のシステムレベル設計フロー(下記、システムレベル設計セクション参照)は高度に専門化したツールを必要とするが、あまりに生成するデザインが少ないため、今日の EDA のビジネスモデルを維持できない。そのような仕組みでは、複雑なシステム設計向けのサービス志向のビジネスモデルが生まれ、EDA 会社は(a)設計システムが作れるようなツールとインフラストラクチャの要素を作る、そして(b)システム設計チームとパートナーを組んで、システム設計そのものと並行してデザイン特有の設計システムを作ることになる。

システムレベル設計

システムレベル設計¹⁰においては、メソッド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も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出来なくなる。従って、システムレベル設計は、ネットワーク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。さらに、HW プラットフォームが現われて、ますます定着するにつれて、組み込み SW が製品の差別化の主要なポイントとなっている。

過去 20 年間、設計者はシステムをさまざま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、(付録の Design Cost Model での“非常に大きなブロックの再利用”と比べて)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け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、やや実装設計が遅らせている。再利用は、ITRS の何人かの編集者により必要性を述べられてきたが、システム設計工程においては、まだ十分広まっていない。90nm から 65nm への移行では、この再利用への要求がクリティカルになる。
- * *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、さらに、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

¹⁰ 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・データ等)とハードウェア(コア、ハードワイヤされた回路、バス、リコンフィギュラブル・セル等)から構成される。ハードウェア(HW)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(SW)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。

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必要がある。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。

- * 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。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来ている。システム合成はきわめて難しいが、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から、使いものになる RTL 仕様を自動生成するようになるだろう。図 16 と図 17 がこの進化を示している。図 16 は典型的な現在の設計フロー（図 15 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。図 17 は近い将来（～2004）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。

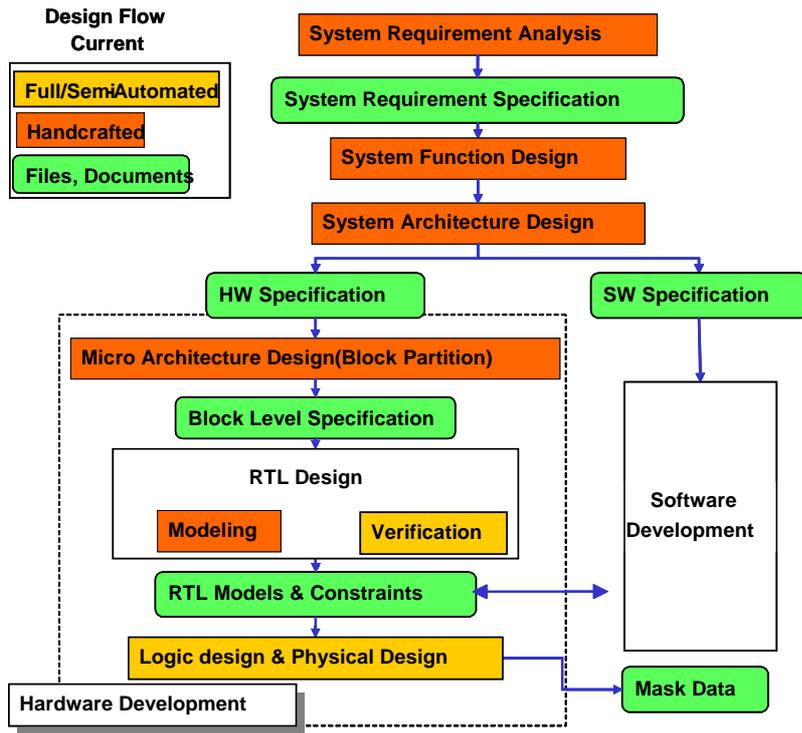


図 16 Current Design Flow

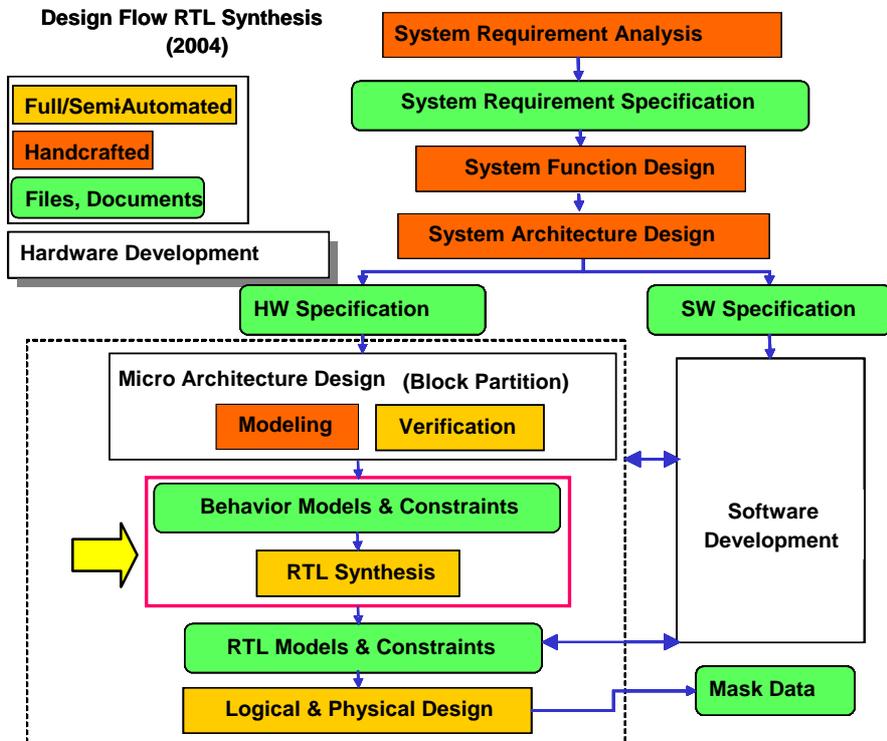


図 17 RTL Synthesis for Design Flow in year 2004

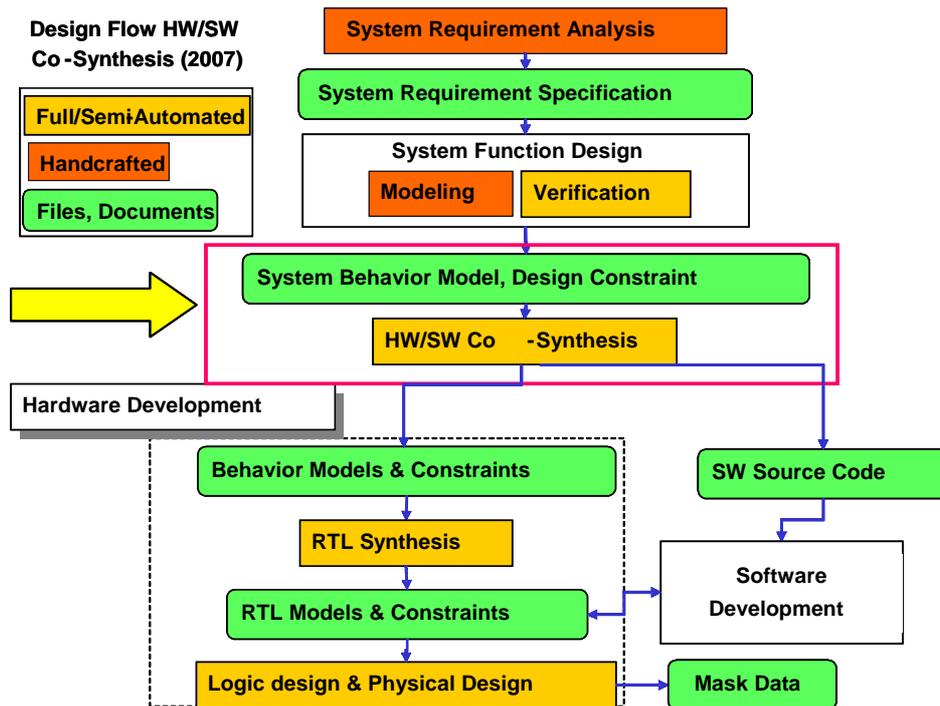


図18 Design Flow in Year 2007

- * **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。図18はおよそ2007年ごろに必要なとされる、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、市販ベースですでに出現し始め、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。

表15 System-level Design Challenges

Challenges ≥ 50 nm/Through 2009	Summary Of Issues
システム複雑度	S—抽象度と仕様の高位化 S—ダイナミズムと柔軟性 S, A—システムレベルの再利用 S, A, P—設計会の探索とシステムレベルでの見積り S—効果的なビヘイビア合成とソフトウェアコンパイル S—インターフェース合成の自動化
システムの消費電力	P, S—エネルギーと性能の柔軟なトレードオフ P, S—新たなデータ転送と保管の技術
異種テクノロジーの統合	S, P, A—協調設計 (HW-SW, chip-package-board, fixed-reprogrammable) A, S—固定化されたアナログ回路、ビヘイビアモデル、合成 P, S, A—様々な構成製品(digital, AMS, RF, MEMS, EO, SW)に対する Top-down implementation planning
組み込みソフトウェア	S—高度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 S—システムキャプチャ及び抽象化 S—SW 合成を含む高位記述からの HW-SW インプリメンテーションへの自動化 S—SW のための形式検証 S—HW-SW 協調検証
検証、テスト、カルチャーとのリンク	S—統合指向の検証およびテスト・アーキテクチャ S, P, A—発散する設計作業およびカルチャー
<i>Additional Challenges <50 nm/Beyond 2009</i>	
システム複雑度	S, P—通信中心の設計とチップ上のネットワークベース通信 All—設計のロバスト性
システムの消費電力	P—スケーリング不能な中央処理型アーキテクチャ S—異種 SOC による巨大なシステムを構築
異種テクノロジーの統合	S—新技術(MEMS, electro-optical, electro-chemical, electro-biological or organic) を含む全体的なシステム統合

この表は、本文に述べた4つのトレンドにともなう設計プロセスへの課題をまとめたものである。各々の課題は最も関連性の高いシステムドライバにラベル付けされている(S-SOC、P-マイクロプロセッサ、A-アナログ、ミックスドシグナル、M-メモリ)

短期(>50 nm)のシステムレベル設計課題

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課題は下記の技術ニーズを創出する。

- * **抽象度と仕様の高位化** 設計抽象度レベルは、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)よりさらに上位に引き上げなければならない。複雑で多機能なシステムの最適な設計空間の探索を可能にするには、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が必要である。使用メソッドおよび抽象度レベルについて、産業界が一致して、複雑なモデル用の設計言語(たとえば C++ から派生した SystemC のような)の基盤づくりを進めなければならない。
- * **ダイナミズムと柔軟性** ダイナミズムとは、システム特性の一つで、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得るものである。新しい抽象度が、機能とアーキテクチャのそのような実行時の修正のために必要になる。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特性の一つである。柔軟性は、複数の抽象度レベル—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。一方で、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は、まだ存在していない。SOC の特徴や制約をみだし、かつ、このようなダイナミックな動作やアーキテクチャを可能にするために、アーキテクチャ上の技術革新が必要となる。
- * **システムレベルの再利用** RT レベルおよびレイアウトレベルの設計再利用では一定の進歩があったが、再利用のメソッドそのもの、および再利用のための設計ツール(たとえば、IP の品質および適応性の評価ツール)は、まだ初期段階にある。¹¹ 設計生産性の著しい向上のためには、プラットフォーム・ベース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が必要となる。ソフトウェアの再利用では、プロセッサの設計メソッドが適用できる。つまり、プロセッサは、「標準」ソフトウェアを実行する一方で、異なるコストパフォーマンストレードオフや計算上のドメインに特化することもある。
- * **設計空間探索およびシステムレベルでの見積り** インプリメンテーション部品が統合できる範囲の増加にともなって、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が要求されている。そのような最適化の基準には電力、面積、スループットなどが含まれる。システムレベルでの見積りは、トップダウンな設計空間探索に付随する、基本的な技術であるが、未だ開発が出来ていない。見積り技術によって、実際の詳細な設計に進む以前に、設計目標基準値が正確に予測できる。こうした見積り機能は、例えば物理レベル、RT レベル、そしてシステム設計レベルでのタイミング見積りのように、より広い抽象度のレベルにまたがることになる。これは、特定のアプリケーション空間に対する統計モデリングと同様の状況である。
- * **効率的なビヘイビア合成および SW コンパイル** 再利用ソリューションを最終的に補完するために、新しい合成方法が必要とされる。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、合成結果の品質の低さゆえに、今日では適用範囲が限定されている。実際、コンパイラによる並列化やプロセッサの特性などでの設計空間探索は、不適切でマッピング結果は品質が悪い。ニーズは、制御とデータフロー混在システムの対する十分な合成品質と、リターゲットブル SW コンパイラを含む。これが可能になれば、様々の命令セット・アーキテクチャ(制御プロセスと DSP)と設計領域に、自動的に高品質な結果を生成することが出来る。

¹¹ IP 再利用を促進するインフラのサポートが必要である。例えば、プロテクションの仕掛けや、検証や認証のサポート等。

- * **インターフェース合成の自動化** 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、手設計したり、パラメータ化されたライブラリから取り出したりするのではなく、合成(あるいは標準化)されるべきである。これはコミュニケーション・セントリックな設計アプローチの出現による自然な結果である。

2. **システムの消費電力** 今日のいくつかのプロセッサ・アーキテクチャー(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、近い将来に破綻し、より大規模な混載並列化にとって替わるだろう。エネルギー、性能および柔軟性の効率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必要がある。それは、例えば、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; 標準的に制御された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・ロジック等である。短期的なレンジでの、消費電力に対するシステムレベルの主要な手段は、データ転送、メモリ・アーキテクチャ、および階層化である。

3. **異種テクノロジーの統合** 適用できる要素技術の範囲にまたがって、より設計の選択肢を効率化するために、異種混交のモデリングと協調設計のための新しいアプローチが求められている。

- * **協調設計** 分割と協調設計のための DT(Design Technology)は、例えば 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)等の何であれ、システム開発プロセス全体にわたるコストとリスク要因にますます大きく関わっている。アナログ - デジタル間のトレードオフと分割では、双方に互換性のある仕様と機能モデリングの抽象化が必要であり、また、単一の実行と分析環境で、システムのすべての部分をシミュレーション出来ることが必要である。より一般的には、設計最適化をスタンドアロンではなく、そのシステム構成全体で考慮しなければならない。
- * **様々な構成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** 統合可能な構成製品の増加により、単一の階層的な異種技術混載のプランニング環境へのニーズが増している。そのプランニングは、構成製品の詳細情報(回路トポロジー、インターフェース、概略レイアウト)に関わるシステムレベルの目標は制約とリンクしている。

4. **組み込みソフトウェア(ESW) - 再利用中心かつ SW 主体の 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内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(これはアプリケーション・プログラマーのインターフェース、あるいは API と見なすことができる)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シリコン性能との間に中間的な媒介と位置付けられる。このシステム・プラットフォームは、次の重要な問題を含んでいる:

- * **高度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** 高度にプログラム可能な、アプリケーション領域に特化した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(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インターフェース・リソース(SW-SW、HW-HW および HW-SW)の適用を可能にする。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」を必要とし、それは、システムのタイミングおよび電力の制約を満たすように、機能を迅速かつ信頼性をもって、プログラム可能な処理とコミュニケーション資源へマッピングする。

- * **システムキャプチャおよび抽象化** システムは、離散的なイベント、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)内で表現される。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、システムの機能上および通信上の要求をモデル化できなければならない。これらのモデルは、正確で実行可能なシステム仕様を構成するものでなければならない。

SOC への追加的な要求技術には、高位記述からの HW-SW インプリメンテーションへの新しい自動化(SW 合成も含む)、SW のための形式検証、および HW-SW 協調検証などが含まれる。

5. **検証、テストとのリンク、そしてカルチャー** - SOC プラットフォーム上での設計の重点が、ブロックの生成から、ブロックの再利用に移ってきており、これは検証、テスト、そして DT(Design Technology)のカルチャー的側面に以下のような関わりあいを与えている。

- * **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らない。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こと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する。形式的に指定された抽象的な仕様記法は、形式的 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。また、形式的 SW 検証の研究も進んでいる。統合指向のアーキテクチャは、テストにおいても同様に必要になっており、その結果、全体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・エレメントを統合することができる。テストコスト(テストの所要時間や、また ATE 速度の制約)の高騰は、ますます SW ドリブなオンチップ自己テスト技術を促進を促すと思われる。これにより、システム設計とテスト・アーキテクチャ設計の間の相互依存性は、システムのコンセプト設計の段階から計画することが必要になる。
- * **発散する設計作業およびカルチャー** システムレベル設計は、ますます抽象化が上がり、かつ、アプリケーションに特化する方向へ突き進んでいる。これは、詳細な物理設計と製造の現実的なコストの問題をますます突きつけられている IC インプリメンテーションの進む方向とは、まるで離反して行く方向である。このインプリメンテーション・ギャップとその解決策は、以下のようないくつかの様相を持っている。(1) このシステムレベル設計とインプリメンテーション設計の間のカルチャーのギャップ¹²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情報と意思疎通を可能にする、設計抽象度レベルの注意深くかつ正確で厳密な定義によって解決されるべきである。(2) 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見積りを与えるような、IC の「シリコン・インプリメンテーション・プラットフォーム」が現実化しなければならない。このような連結ポイント(たとえば、設計環境の中央監視室、ないしは「コックピット」のようなもの)たる DT 無しでは、システムレベル設計と IC インプリメンテーションの間のリンクは、その場しのぎのままであり、設計の効率劣化の原因であり続ける。

¹² システム設計のエンジニアリングの基礎は、IC のインプリメンテーション設計のそれとは違いがあり異なっている。その結果として、インプリメンテーションに熟達した設計者は、システムレベルの設計者の関心事に気づかなくなるし、また、これは逆も真である。すなわち、カルチャー・ギャップである。

長期(<50 nm)のシステムレベル設計課題

1. システム複雑度 長期的には、システムの複雑度によって、計算処理上および全体的なシステムのロバスト性のための通信が新たな課題となる。

* **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大きく異なることを要求する。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したりすることが不可能になるので、考え得る解決策は非同期(あるいは、全体的に非同期でかつ局所的に同期(GALS))設計スタイルへの変更である。そのような形態では、自己時間を持つ機能の群がネットワーク指向のプロトコルによってコミュニケーションする。これは、システムレベルの消費電力最適化手法(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)の要求と合致する。通信中心の設計では、(1)アルゴリズムは、高位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。また、(2)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でなければならない。

* **設計ロバスト性** ネットワーク指向のパラダイムに続いての、歩留り問題、非同期設計スタイルおよび通信中心の設計に変化することにより、SOC 設計が従来の IC 設計のスタイルとは異なって、大規模通信網の生成との類似性が増すと考えられる。そのようなネットワークでは、通信は損失があったり、また、ノードは通信を失敗したりするかも知れないと想定できる。しかし、それでもなお、ネットワークは、正確さとスループットに対するシステム要求を達成しなければならない。そのような、本質的に不完全なインプリメンテーション機能部品セットへ、完全に指定された機能のマッピングを行わなければならないわけで、そのためにランタイムで、フォールトレラント処理と通信機能部品を設計し、マッピングし、そして制御する新しい手法が必要になる。

2. システムの消費電力 65nm 以降では、MPU の設計は、性能、消費電力、放熱の問題の根本的な壁に直面する。その問題の大きさおよび困難さは、高性能なシステムへの抜本的な考え直しをすることが必要になるほどである。消費電力は、オンチップ処理の並列化を細心の注意で適用することのみによって管理することができる。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ことになる。予め準備さ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のではなく、最大限に並列化した機能を最大限に並列化したインプリメンテーションをマッピングすることが、システムレベル設計の将来のゴールとなる。メソッド的には、これは、中央処理型のインプリメンテーションを超えた、分散型のインプリメンテーションを特徴とする設計の新領域を意味する。これは、先述の通信中心の設計の必要性を再現するものである。汎用の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的なマッピングは実行可能であろう。また、多数の変数に依存するアプリケーションについては、リコンフィギュラブルロジックが適用可能であろう。さらに、より低いスループットが求められる時には、システムの不必要な部分のすべての電源を落とすことになるだろう。そうした傾向の中で、スタンドアロンの MPU 設計スタイルは、'処理の海'(Sea-of-processing)の設計スタイルへと発展することになるだろう。

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、それらが統合できる規模や要領の増大につれて、アナログやミックスドシグナルを別の設計分野とする考え方を終らせてしまう可能性がある。事実上すべての設計領域にまたがるシステム・エレメントが、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の対象となるだろう。エレクトロ・ケミカル(「チップ上の実験室」)や、エレクトロ・バイオロジカル(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

いモデル化のアプローチ、新しいタイプの合成と統合化のガイドラインとルールを要求する。そして、そのような新しい設計の数の分だけ、すべてが新しいツールセットを促進する可能性が出てくる。この新しいテクノロジーエリアにおける研究は、まだ始まったばかりで、テクノロジーそれ自身に集中しており、未だ、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着手されていない。

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 signoff 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、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する。貧弱な評価により過度に 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ない。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。

論理、回路、物理設計のための困難な課題は表 16 の中で要約される。特に記述されている以外、短期課題は長期へ継続するので、それぞれの課題を連携させている。重要なポイントは(1)見積り、(2)モデル化する能力を改善したことである。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)を包含する;これらは、テーブルの中で与えられた最初の 2 つの課題に相当する。ポイント(2)は、SOI(Silicon on Insulator) の増加した重要性、ミックスドシグナルのインプリメンテーションおよび回路革新と同様に増加したりーク、ソフトエラーおよび原子の規模の効果のような現象を包含する;これらは、テーブル中の次の 2 つの課題に相当する。いくつかのアナログ / ミックスドシグナル議論がシステムドライバ章のなかに統合されたことに注意する。

表16 Logical, Circuit and Physical Design Difficult Challenges

CHALLENGES ≥ 50 NM/THROUGH 2009	SUMMARY OF ISSUES
効率的で見積り可能な環境構築	P 逐次的な計量分析と最適化 P,S 統一された組みこみと配線のプランと見積り予測 P,S 同期化されたグローバル信号 S,A,P 混載のシステム構成 P,S 検証とテストの連携
柔軟性のある設計と製造のインターフェース	P,S,A 製造のための基本的なチップのパラメータ (timing, skew, matching)とダイナミックな可変基データの不確実性 S,P,A,M プロセスのモデリングと指数化 P,S 費用対効果の高い回路、レイアウト、レチクルの改良
複雑で、理想的でないデバイス微細化とパワーマネジメント	P,S リークとパワーのマネジメント S,P,A,M 信頼性と故障限界、ソフトウェア P,S 解析と複雑さと一貫した解析装置
デバイステクノロジーの進化を十分に取込んだ回路設計	P 新回路構成による消費電力や効率の挑戦へのサポート P,S SOI に関するツールの組みこみ A,S アナログ合成
ADDITIONAL CHALLENGES < 50 nm / BEYOND 2009	
効率的で見積り可能な環境構築	S 信頼性の高い、予測可能な アプリケーション特定のプラットフォームの構築 S コストに影響される構築フロー
柔軟性のある設計と製造のインターフェース	P,S,A ますます増える原子レベルの大きさの変化する影響
複雑で、理想的でないデバイス微細化とパワーマネジメント	P 製造テストの信頼性の回復
デバイステクノロジーの進化を十分に取込んだ回路設計	P,A ますます増える原子レベルの大きさの影響 P,S,A 適応性と自己修繕回路 A,S 低消費電力センサー技術とセンサー・インターフェース回路、超小型光デバイス

この表は上記の4つの動向が示唆するデザインプロセスの進歩に対する課題をまとめたものである。各課題は最も関連するシステムドライバのリストでラベルをつけられている (S システムオンチップ、P - マイクロプロセッサ、A - アナログ/ミックスドシグナル、M - メモリ)。

論理、回路、物理設計における課題

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する。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する。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ある。さらに、チップ・インプリメンテーションはますます多目的のグローバルな最適化を必要とし、それらは 複雑なトレードオフ要因を取りこむソリューションを生み出す必要がある。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な設計情報を備えた最適化技術は必要である。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ある。設計プロセス議論で注意されるように、マトリクスは生産性改善を確認するために必要である。
- * 統一されたインプリメンテーション/相互接続プランニングおよび見積り予測— 今日のインプリメンテーション・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

を作成する。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、そしてグローバルに、リピーター挿入を定義し、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で相互に融合させる；結果は、論理合成および最適化への配置配線制約として渡される。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減少することができる；相互に融合されたアーキテクチャ最適化は、システムに配線層の数と寸法をマッチすることができる。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し、短期ゴールとして 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、同一環境での配置と配線を 必要とする。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性を改善するだろうということに注意する。チップ・インプリメンテーションについては、レイアウト評価が必要である。(1)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、(2)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(3)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する。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ない。¹³ 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・フローの余裕があつて、同様に適合する必要があるだろうとともに、長期である。コスト最適化は別の長期のゴールである。

- * 同期化およびグローバルシグナリングー 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 12FO4(12 段のファンアウトが 4 であるインバータ)遅れ(PIDS 章における CV/I 指標のおよそ 170 倍)としての最小のクロック単位；チップを横切る信号伝達には ますます多くのクロックサイクルを必要とする。システムドライバ章の中で注意されるように、グローバルな MPU の相互接続は、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ない。しかしながら、純粋な同期設計のクロック分配は、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)を増加させることになる。その結果、同期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る。たとえば、信号伝達に基づいた設計パラダイムは、ローカルには同期で、高機能ブロックの全体的には非同期カップリングを必要とする。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ある、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 つの規模およびマルチサイクルによって変わり、パスの相互接続は普通であろう。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ない。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う；より大きなマージンの必要性は、結局従来のディレイスラッグ最適化上のリターンを制限する。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し、次にオフチップ化する傾向にある。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

¹³ 過去のメソッドは 単純な統計上のモデル(例えば "wireload models")と "limited-loops" の繰り返しに頼る。(例えば一つの構築されたパスは レイアウト見積りから始まり 1秒過ぎると 再度 見積りをする) 現在の、そして未来のメソッドは (a) 限定された回路を必要とする、そして改善するか、あるいは 同等にするべきレイアウトスタイル(例えば 2つのレベルの プログラマブルロジックでの構築、あるいは 強固に保護された信号配線)がタイミングとノイズを保証する。(b) 良い予測(例えば constant-delay methodology variants)の欠如による "実施可能な仮定" の使用、そして (c) 第一に予測性の要求項目を削除すること(例えば 複数のクロックサイクルがコンポーネントを分離するにもかかわらず 正しい動作を保証する時間-鈍感な同期化プロトコル)すべてのメソッドと同じように これらは design TAT,area,timing,power など さまざまな規準の中で トレードオフを必要とする

合成および分析が伴うに違いなく、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る。

- * **混載のシステム構成** — 混載は多くの形式でインプリメンテーションに影響する。(1) デジタルロジックを備えた、アナログ、ミックスドシグナルおよび 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課題を示す。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 回路のより大きな制約を扱うに違いない。長期 MEMS(Micro Electro-mechanical System)あるいは他の技術の統合は斬新な欠点回避およびフォールトトレラント方法を必要とする。(2) コスト要因(製造コストおよび信号伝達のコスト)により駆り立てられて、統合的な選択では、マルチチップ(system in package)および積層チップ オプションを包含しなくてはならない。これは、システムレベルの場合に、パッケージおよびボードまでの最適化を必要とする。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パッケージ解析およびコ・オプティマイゼーションのツール支援を指定する。(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、再プログラム可能なブロックは、SOC 設計の中にますます重みをます。レイアウトツールは、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ない。(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表現を含めたハード、あるいはソフトIP再利用、制約条件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ある。IP ブロックを組み入れる事で、インプリメンテーションにより困難度がます。たとえば、ブロックとセルの「ミックスモード」配置は今日も課題のままである。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ある。再利用の別の面はデザイン変更である: 長期 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ある。(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ない。)
 - * **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ロジック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がチップ・インプリメンテーションに必要である。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しなくてはならない。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におよぶ。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る。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: たとえば、物理設計ツールは、新しいタイプの欠陥のための欠点位置を抽出する必要がある。同時に、速い生産性向上では、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。
2. **ばらつきと設計-製造とのインターフェース** — プロセスの許容誤差(process tolerances)を継続的に十分な範囲に保つことが不可能になることで(PIDS、配線、またリソグラフィ章を参照)DT(設計技術)としての、ばらつきに対する課題はさらに深刻になる。Lgate や Tox のような基本のパラメータの 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ーや論理/レイアウトが必要とされる。まさに原子規模での効果が、多くのプロセスの段階(ゲート酸化物平坦化、イオン注入など)での統計的ふるまいを支配する。
- * **統計的タイミング解析(STA)および性能検証ツール** — STA および性能検証ツールは、分散によってパラメータ化される寄生値、遅延値およびジオメトリを理解する必要がある。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 つのウエハあたりの、パラメータ化

された歩留りおよび生産量(revenue)のために最適化することになる。製造ばらつき(および様々な新しい材料やプロセス)は、より広範囲な設計-製造のインターフェースを要求する(そのインターフェースは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。) 製造ばらつきの原因要素のための、十分な統計的および電氣的/幾何学的なキャラクタライズが必要となる。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依存する;このばらつきのタイムスケールは何十億回のクロックサイクルとなる¹⁴。そして、これはタイミングとノイズに大きく影響する。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して、単純な手法によるガードバンド(guardbanding)ではコストが高い。局所的な発熱のばらつきに対するより正確な分析および制約が、過剰品質な設計(overdesign)を減らすために必要である。さらに、統計的なスタティックな消費電力解析に必要なものとして、リーク電流のばらつきがあげられる。リーク電流は、プロセスパラメータ(ゲート長、酸化膜厚と閾値電圧)に対して指数的な依存性があり、今日のハイエンドなものでは、リークでの消費電力では、パフォーマンス依存での35%までの広がりに対して、20倍の広がりを示す。この傾向はさらに増大するため、ばらつきの要因に対する注意深い解析と相関性が非常に重要である。

- * レチクル補正技術(RET(Reticule Enhancement Technology)) – RET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する。)とディープ・サブ波長(deep-subwavelength)のオプティカルリソグラフィ(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)などを含む。RETは製造複雑さ要因が増大してきたことを象徴しており、プロセスロードマップの先行きが困難であることを反映している。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る。1) OPCとPSMのために、設計規則は、非常に複雑で、また周囲の状況に依存したルール(context-dependent)となり、レイアウト合成の生産性が問題となってくる。「ローカルな設計規則」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る。物理検証は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を正確に理解しモデル化する必要がある。2) RETが広範囲に適用され、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、¹⁵ 製造へのハンドオフは、より厳しくなってくる。RET挿入(またマスク検査)では、あるクリティカルなデバイスか、あるいはある配線の寸法だけが注意深い挿入する必要があり、またある種の改善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しなくてはならない。従って、将来の設計フローでは、詳細な機能の意味およびシミュレーション結果を、物理検証およびマスク・フローへと渡すことになるであろう。長期的観点での支援技術は、時代遅れで非効率的なGDSIIストリームおよびMEBESデータを置き換えるであろう: おそらく電子システム「サプライチェーン」のインフラストラクチャの中で、DTと製造の間のより完全な双方向のつながりが出現するに違いない。

3. シリコン複雑度、非理想的なデバイススケールリングおよび、電源制御 – 最下流での抽象度を利用する場合には、多くの困難な物理的効果に関して、詳細なモデリング、分析および最適化が要求される。

- * リークとパワーマネジメント – リーク消費電力は、低消費電力とハイパフォーマンスの両方に対して、非常に重要である、なぜならこれは“不要”な消費電力ということだけでなく、デバイス技術者がトランジスタをスケールリングし続ける際に避けることができない問題でもある。クリティカルなグローバル配線

¹⁴ 他の信頼性に関連するダイナミックなばらつきは、さらに100万倍長いタイムスケールで発生する。(例 ホットキャリアに起因する酸化物の破壊に伴う閾値の変動あるいはエレクトロマイグレーションによる低効率の変動など)

¹⁵ 90nm ノードでは、MEBES ファイルは1つのマスク層で200+GBまで(圧縮前)になると予想される。

に対するバッファ挿入の手法は、大きなリークを引き起こす、なぜならバッファには、スタック効果 (stack effect) がなく、またデバイスの総計のW (チャンネル幅) が大きく、(高周波バンド幅の配線に対しては) 低閾値のデバイスである必要があるためである。歴史的に見て、材料の改善は、現存技術の積極的なスケールアップよりも遅れるため、回路設計者に対するゲートリークは引き続き問題となる。消費電力関連のノイズの課題は、電源供給配線の誘導性ノイズを含む、これは消費電力密度の増大に従い、より多くの電流が、低電圧電源に対して、流れるためである。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す。これは、バンプ (bump) の数およびピッチのスケールアップがそれほど積極でないことに加えて、同期動作およびパワー・アップリセット条件によってさらに悪化する。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る。90nm 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ない。パッケージ側では、信頼性のため、最大電力が、パッケージの基準 (package rating) を連続的に 100ms 以上超過しないように管理される必要がある。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をコントロールを要求する。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る。システムドライバ章が指摘しているように、MPU および SOC-LP 中のダイナミックおよびスタンバイ・パワー管理のギャップが、ITRS の最終ロードまでに 8 倍と 230 倍にそれぞれ達する。MPU では継続的に消費電力のバジェットが増加するために、解析機構 (寄生抽出、ロジック活性度、電流フロー) の改善に加えて、新たなチップ・パッケージでの熱分析が必要となる。ライブラリーキャラクタライズ、合成およびレイアウトでは、多値のスレッショルド電圧や多電源をサポートすることで、おおよそ 5 倍の消費電力削減を実現するように要求される。ツールに対して、(Peltier-type thermoelectric cooling) のような長期的観点での技術や、OS による動的な周波数、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しくみが必要になる。

- * 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 self-heating) は、単純で明快な抽象化 (たとえば、ゲート負荷容量対出力 SLEW の上限値) によって実装フローへ統合されてきた。現在、そのような「メソドロジ的な」抽象化は、従来のツールフローを大きく変更することなく、構築しながら修正することが可能である。しかしながら、さらに抽象化を改善し解析をすすめることで、(1 つのウエハあたりに得られる生産量を増やすための) ガードバンド (guardbanding) を削減する必要がある。1) 短期的および長期的な課題は、製造されたあるいは動作しているデバイスおよび配線が全て 100% 正しいということが困難になることである。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る、それは 低電圧化のために、指数関数的にバーンイン時間の増加が必要になること、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、などの理由からである。論理合成、物理合成は、適切に合成、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) の管理の下で動作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・ロジックを含む) を徐々に統合するようになるであろう。2) 微細化による原子スケール効果のために、新しい欠陥モデルが必要となる。たとえば、ゲート酸化物は、今でも非常に薄いので、(あまりにも少数の遷移可能状態 (too few trap states) しかない) 破壊的な酸化物のブレイクダウンを起こしえる。もっと正確に言えば、電流は、時間とともに、また酸化物が破壊されるに従い増加する。「欠陥」を生成するものに対してのスクリーニング基準だけでなく、特殊な回路トポロジーでは、どれだけの欠陥密度を受け入れられるかという依存性も合わせて理解する必

要がある。

- * **ソフトウェア信頼性** — ソフトエラーは、地球圏外の放射線(宇宙線)か、あるいは物質(α 線による)によって引き起こされるが、これはもはや SRAM のビットセルの逆転(upsets)に限らず、短期的な信頼性駆動のDTに対する課題(discontinuity)である。90nmを超えるノードでは、論理回路とラッチは、Qcrit(critical charge)が小さくなったために、攻撃を受けやすくなっている。この傾向は、記憶素子ノードの容量の低減と、Vdd のスケールリングに従い助長される。直接的な故障によるラッチ回路でのソフトウェア率の増加は、システム信頼性に非常に大きなインパクトをもたらす。さらに、プロセッサのマイクロアーキテクチャがより多くのパイプラインを用いることで、ラッチ間の論理段数は少なくなり、ビット反転(Single-event upset)がラッチに入り込む可能性が増大する。スタティック回路での反転は、ラッチへ伝播しなければエラーとならないが、その他のダイナミックのようなものは、発生すること自身が故障となる。どちらの場合でも、それぞれのセル、デバイスタイプ、配置でのソフトウェア率を正確に予測するような解析ソリューション(別のキャラクタライズ段階を通して。)が要求される。一般的に、自動設計手法では、設計制約に違反せずに、SEU を防ぐか管理するように、論理、回路、物理設計を修正することが必要となる。(たとえばエラー訂正の自動導入、サイジングなどによる)
 - 1) パッケージレベルでは、今後の世代では、システム信頼性とソフトウェアの削減のために、リードフリーの C4(アレイ型 I/O)が必要となる。リード付の C4 では、C4 の外側のクリティカルな回路の配置を工夫することで、 α 線によるSEUを軽減できる。
 - 2) 論理と回路レベルでは、冗長性やパリティなどが、システムの状態を保護するために有効であろう、但しこれはパフォーマンスとのトレードオフとなる。
 - 3) マイクロアーキテクチャのレベルでは、空間的、時間的な冗長性がソフトウェアの削減と、システム信頼性の向上に有効である。
- * **モデリングと解析** — シリコン複雑度の増加およびガードバンド(guardbanding)に従い、消費電力/電流およびタイミング/ノイズに関するモデリングと解析はさらに困難になる。重要な課題は、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ある。
 - 1) 電源設計および消費電流の管理を行うために、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が必要となり、このためには、キャラクタライズツールおよび豊富なライブラリー・モデリングの標準化が必要となる。速くて詳細な熱解析が要求されるかもしれない、たとえば、低い誘電率(low-permittivity)の誘電体を使うことで、熱伝導性が低くなることと、配線の joule heating の問題が組み合わされて、エレクトロマイグレーションの信頼度のリスクが高くなる。¹⁶
 - 2) タイミングおよびシグナル・インテグリティの収束のためには、オンチップおよびパッケージ配線での寄生容量およびインダクタンス、さらに配線のクロストークノイズ、カップリングによる遅延の不確実性および基板ノイズ注入の正確な解析が必要となる。将来、配線最適化のための評価項目では、信号、供給電源ノイズ、熱勾配、そしてサブ 100nm ノードでは一反射、EMI および基板カップリングなどが必要となるであろう。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)がある。
 - 3) より一般的に言って、ナノメータの実装フローでは、CPU(すなわちツールの実行時間)対精度でのトレードオフを行うための「単調な悲観性」(“monotonically pessimistic”)の種々の解析が必要となる。たとえば、論理/レイアウト合成の初期の段階では、フィルタリング、予防のために、高速であるがそれほど正確でない解析で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サインオフの中で時間はかかっても、より正確な解析を要求することがある。最終的には、合成での種々の

¹⁶ SOI プロセスでも同様な熱に関する課題が出てくる、これらはタイミングやノイズ解析に影響を与えるダイナミックなばらつきの原因となり、次の課題として討議されている。

最適化目標値としての「マクロ・モデル」が、合成の中で、効率的に評価されかつ適切にふるまいが定義されるように、近似的に解析される。

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、理想的でないスケールリングの場合(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、ゲートとドレインのリーク電流の増大、ボディ効果(パスゲート・ロジックがそれほど魅力的でなくなる)およびオーバードライブによる損失を引き起こす。パワーマネジメントの観点から見ると、いままでの動作速度に対する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が、あまり意味がなくなってしまう。DT は、処理速度/消費電力の性能における全面的な改善が可能となる、スタティックな CMOS に替わる新たなデバイスの展開を可能にしなくてはならない。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らない。そのような DT の進歩はデバイス技術の進歩と補完的でなければならない、たとえば、1)ゲート絶縁体トンネル効果および破壊の健全性(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)の緩和のために、高誘電率のゲート絶縁体が必要となる、また 2)基板にバイアスをかける(パワーマネジメント・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、body contact 技術の改善が必要となる。¹⁷

* 消費電力と性能の課題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、いくつかの回路のトレンドに従い、DT はそのサポートが必要となる。1) 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)な回路がさらに一般化するであろう。(スタティックな CMOS は普及したままであっても)。2) 非常に低電圧のオペレーションは、さらに一般に多用される。これは並列性の採用によって、デバイス性能要求が緩和されるためである。3)ボディバイアス、クロックおよび電源配線のようなパラメータの制御がコンパイラそして、OS ベースになってくる。そのため、アーキテクチャレベル管理をサポートするために、物理応答のビヘイビアレベルでのモデル化が必要となる。

* 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)技術は低い接合容量により、より速いスイッチングが可能となり、またデジタル・アナログ基板ノイズが回避できる、これはミックスドシグナルの SOC には特に魅力的である。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る。フローティング・ボディ効果や self-heating は歴史的にみてもあきらかに明確にすべき点がある。:ワーストケースでの保護(guardbanding)が必要であれば、SOI 利点の多くは無意味になる。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あろう)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)を含んでいる。¹⁸ アルファ粒子による SEU が削減され、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ない。しかし、self-heating および ESD 保護は、酸化物層の熱抵抗が増加することでさらに問題になる。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、Vdd-GND のデカップリング容量を削減する;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

¹⁷ デバイス技術での向上は PIDS 章で述べられる。ただし、それは万能薬ではないことに注意を要する。ここでの全ての DT に対する要求は依然としてある。

¹⁸ フローティング・ボディ効果は、部分空乏型 SOI(PD-SOI)でのボディ・コンタクトによって、回避できる。ただし、面積の増加や完全空乏型が使えないデメリットはある。(ボディ・コンタクトの有無、やコンタクトの種々の形状との組み合わせでライブラリの種類は激増する可能性がある) ボディ・コンタクト PD-SOI は、MOSFET で 4 端子モデルが必要なため、モデリングやシミュレーションのインフラと密接に関連する。

なります)の維持のために、初期のフロアプランニング段階から、明確なデカップリング容量の管理が必要となるであろう。完全空乏型な 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であろう。

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ある。設計検証は設計全体のプロセスに対し比較的小さな役割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ある。(恐らく、そのような見方はこのドキュメントのほぼ全体にわたって反映されているであろう。)しかしながら、現実には著しく異なる。検証は設計全体のプロセスの支配的なコストとなってきた。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、設計者の数に対して2倍もしくは3倍に達する。設計コンセプトおよび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ほんの前処理にすぎなくなりつつある。

この不幸な状況は、以下2つの進化に起因する。1つめは、ここ近年の設計における機能の複雑化が、息をのむほどの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、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せる)は、設計規模が指数関数的に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る。¹⁹ 2つめは、歴史的に、偉大なる強み、つまり、論理合成や配置配線、テストパターン生成用の自動ツール、これが、ボトルネックである検証を残して、設計の他のプロセスにおいて、桁外れの進歩をとげたことにある。大きな突破口なくして、検証は、半導体産業の今後の進歩において、予測不可能ではかりしれない障害となりうる。

これらの突破口の全体的な傾向として、場当たりの検証手法から、より形式的な検証手法への移行がある。特定の形式的検証技術が成功するかどうかは議論の余地があるが、しかしながら全体的な移行は避けられない。繰り返しモデルを作成しなおしたり、場当たりのベクタの選択を行ってシミュレーションしたり、たまたま発見されたバグを修正したりすることによって、システム設計の機能検証は行うべきものではない。しかしながらこれはまさに今日行われている検証手法である。シミュレーションをベースとした試行錯誤的な検証手法は膨大な時間がかかり、予測不可能である。なぜなら、機能の複雑度は前述したようにダブル指数的に増加するからである。²⁰ 技術の発達は、設計分析のための正確で効果的な手法の開発に依存する。(大半のフォーマルやセミフォーマルな検証は今日研究されている)。また、信頼できる、予測できる技術的な実施は、簡単あるいは、“構造による正確”の形式に対しては)不必要でさえあるかもしれないが、多くの検証への挑戦をもたらす。表17では、設計検証の主な課題をまとめている。本章の残りの部分では、さらに詳細を説明する。

¹⁹ 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また倍増する傾向にある。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セットの検出、もしくは類似した計算方法によって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との連鎖を超えた理論を要求するとすると、コンピューターの挑戦ですらいつそう不利になる。

²⁰ この議論は伝統的なシミュレーション同様、ハードウェア・エミュレーションにもあてはまる。ハードウェア・エミュレーションは“シミュレーション”の速度において、検証に非常に重要な支援を行いつつ、はかりしれないほどの進歩をもたらしている。それは、しばしばシステム・インテグレーションやソフトウェア開発の早期の開始をも許す。しかしながらエミュレーションシステムは、ベクタを1単位時間あたりにひとつずつ処理していくので、検証問題の計測可能な、長期的な解には恐らくなり得ないであろう。

表 17 Design Verification Challenges

<i>Challenges ≥50 nm/Through 2009</i>	<i>Summary Of Issues</i>
増大した検証能力	S, P, A—検証複雑性は設計規模においてダブル指数関数的に増加 S, P, A—大規模設計を処理する能力と同様の高カバレッジの必要性 S, P, A—セミフォーマル技術
確固とした検証ツール	S, P, A—予測不可能な検証アルゴリズムは発見的かつ検証難易性の改善を要求
検証測定基準	S, P, A—動作カバレッジ S, P, A—バグカバレッジを確定するアルゴリズムと現実的なバグモデルの要求
ソフトウェア検証	S—ソフトウェアの本質的な検証困難性の増大 S—伝統的なソフトウェア検証技術の不適切 S—統合されたハードウェア/ソフトウェア・システム S—検証可能な設計
検証再利用	S—IP ブロック検証の再利用 S—IP ブロックの抽象的な動作の指定 S—IP ブロックの環境的制約の指定 S—階層的検証アルゴリズム
MPU 検証手法	P—異なる コスト利益間トレードオフ (高コスト受容) P—例外的なハイキャパシティの必要性 P—長期設計サイクルとバイブライン的な開発チームのための予見性の必要
MPU 検証容易化設計	P—他のシステムドライバよりはるかに必要となる、特殊化された技術
コンカレント設計	P—設計の複雑度を増加させる、新しいプロセッサのコンカレント設計
AMS の促進	A—極度にプリミティブな状態は、短期的には、困難なハイブリッド・システムを強制

この表は 4 つの傾向に示された、設計全体のプロセス進歩の課題をまとめている。各課題は、最も関係したシステムドライバのリスト(S—system on chip, P—microprocessor, A—analog/mixed-signal, M—memory)に分類にされている。

表 17 Design Verification Challenges (continued)

<i>Additional Challenges <50 nm/Beyond 2009</i>	
検証容易化設計	S, P, A—新手法の必要性 S, P, A—特性抽出と性能と領域インパクトの最小化
抽象度の高レベル化	S, P, A—新アルゴリズムの必要性 S, P, A—ハイレベルデザインによって容易化される設計複雑度 S, P, A—Equivalence checking vs. RTL
仕様のヒューマンファクタ	S, P, A—収拾不能な仕様の正確性 S, P, A—どんな仕様を理解しやすいかを理解する必要性 S, P, A—仕様のモジュール化、変更可能な方法の検討
ノン・デジタルシステムの検証	S, P, A—アナログ効果のハイブリッド・システム検証 S, P, A—アナログ特性のハイブリッド・システム検証 S, P, A—確率的システムの検証
異種混合システム	A, S—MEMS, EO デバイスとエレクトロ・バイオロジカル・デバイスのモデル化方と分析方法

この表は 4 つの傾向に示された、設計全体のプロセス進歩の課題をまとめている。各課題は、最も関係したシステムドライバのリスト(S—system on chip, P—microprocessor, A—analog/mixed-signal, M—memory)に分類にされている。

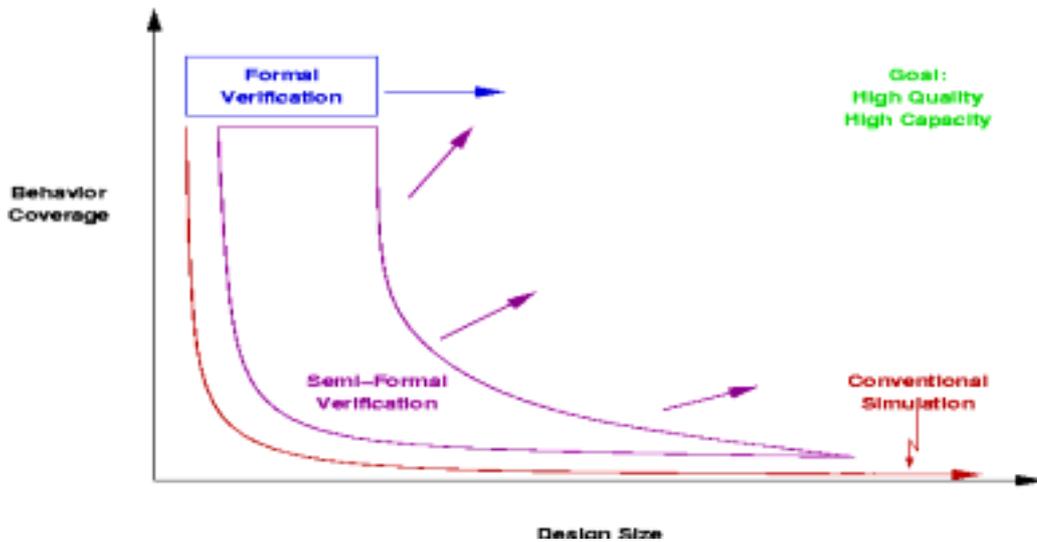
設計検証の課題 —短期 (>50 nm)

検証のための最も重要な課題の多くは、大部分あるいはすべてにおいてシステムドライバに関連する。短期的には、8つの主な課題(以下の1-8)は、フォーマルもしくはセミフォーマルな検証技術を、より信頼性があり、制御可能にすることに集中している。特に、検証能力の進歩とフォーマル検証ツールの確証性は、意味のある検証品質測定基準と同様に必要とされている。長期的には、4つの主な課題(以下9-12)は、抽象レベルをあげることとフォーマル検証の範囲を広げること集中している。これらの長期的な課題は、現に今日意味のあることではあるが、短期的課題はすでに危機的状態にある。一般的に、検証課題のすべては、SOCの課題にあてはまる。加えて、以下の短期的課題はSOCにとって非常に重要である。MPUは明確な検証課題を与える。リーディングエッジの複雑性と途方もなく複雑な設計の固有な経済性の両方から、比較にならない程多くの課題が引き起こされている。

1. 能力 (キャパシティ) — 現在のフォーマル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ない。シミュレーション・ベースのツールはいろいろと複雑なデザインをシミュレートできるが、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しない。検証のための唯一の最も重要な課題は大規模で複雑なデザインに対していかに高品質な検証カバレッジを与えるかにある。図19は現在のツール概要を示す。²¹ 伝統的シミュレーションは、小さなデザインの高カバレッジ、もしくは大規模デザインの非常に貧弱なカバレッジを与えることはできるが、大規模デザインに対して高カバレッジを与えることは出来ない。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供給するが、現在大規模なデザインを扱うことができない。セミフォーマル検証は、キャパシティを獲得するためにカバレッジを犠牲にして、フォーマルとシミュレーション・ベースの技術を混合しようと試みている。この課題は図の上部右コーナー方向に動く。

²¹ ここでは、“Conventional Simulation(伝統的シミュレーション)”とは、1単位時間あたりに1処理しながら、システム動作をシミュレーションすることに基づいた検証技術をあらわす。“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・トラジェクトリーエミュレーション、モデルチェッキング、理論証明のような、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、あらゆる技術をあらわす。“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合するより、フォーマルカバレッジの完全性を犠牲にすることで、大規模デザインを取り扱いを試みる幅広い範囲の技術をあらわす。

図19 Current Verification Tool Landscape



2. ロバスト性 — 図19では、ある規模のデザインにおいては、ある検証手法が有効であるという印象を与える。この印象は間違いである。検証問題の複雑度に対処するために、検証アルゴリズムは非常に気まぐれで、発見的なものに依存している。設計と検証のいかなる組み合わせにおいて、熟練者ですら検証アルゴリズムが完全かどうかを見定めるのには、非常に苦労している。トランジスタやゲート、ラッチの数といった問題の規模を計る共通の尺度は、検証の複雑度とはあいまいにしか関連しない。たやすく検証することのできた数千ラッチを使用したデザインがすぐに見つかるのと同様、既知の検証手法をまったく無視した百以下のラッチを使用したデザインを見つけるのもまた簡単なことである。このような予測不可能なことは、設計現場においては受け入れがたいことである。重大な検証課題は、検証のプロセスをより確証性の高いものにするることである。これは、検証アルゴリズムの発見的な部分を改善するか、あるいは検証容易化設計のための手法を先導し、与えられたデザインの検証難易性を改善するかによって作りあげていくことになる。

3. 検証測定基準 (メトリクス) — 短期的な検証課題として、検証の質を計る必要がある。特に、カバレッジの意味のある概念は必要である。2種類のカバレッジ、つまり、動作可能性カバレッジとバグ可能性カバレッジは区別されなくてはならない。

- 動作カバレッジは、十分に定義されていないが、どんな場合でもシミュレーション・ベースもしくはセミフォーマル検証にのみ当てはまる。(どんなフォーマル技術も、100%の動作カバレッジを与えながらシステムの起こりえる動作の100%を探求する)。動作カバレッジの測定基準は、一般的に伝統的なシミュレーションがデザインの可動動作のごくわずかな部分だけを評価できることを示唆している。実際のところは、シミュレータは、測定基準が示すよりも、設計エラーを見つける上でははるかに有効である。
- バグカバレッジは、検証カバレッジの重要な測定手段である。しかしながら、現状ほとんど理解されていない。バグカバレッジはテストにおけるフォルトカバレッジとやや類似している。しかしながら、不幸なことに、いずれの方法においても設計エラーが製造欠陥に類似しているという証拠はない。問題は、可能なバグ(フォルト)のどの部分が動作カバレッジのいくつかの組み合わせ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にある。検証された仕様が不完全な場合、検証中のデザイン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、以前問題がある。課題は、意味の

あるバグモデル(實際上発生する設計エラーの深い理解を要求することのない、欠陥モデルに類似しているモデル)を作りだすことと、与えられた3つの条件(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にある。

4. ソフトウェア — SOCの機能の大部分はソフトウェアによって定義されるので、主なSOC検証課題はソフトウェアおよびハードウェア/ソフトウェア・システムの検証方法にある。現在のところ、ソフトウェア開発では、設計レビュー、分析ツール、テストなどが行われているが、ハードウェア開発ほど厳密ではない。その上、ソフトウェアは、より複雑でダイナミックなデータと巨大なスペース(通常、検証の決定不可能性からくる無限のモデル化による)により、検証は本質的に難しくなっている。ソフトウェア検証のための古典的フォーマル技術は、SOCに適用するにはあまりにも労働集約型すぎて、適用可能にならない。短期的な課題は、ソフトウェアの基本的で低いレベルの部分ですら検証を許す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の理解と同様、ソフトウェアの確固たる検証手法を開発することである。

5. 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約束する。主な課題は、すでに設計され(かつすでに検証された)ブロックから組み上げられるシステムの迅速な検証に対応した検証手法を開発することである。重要な課題は、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いかに行うか、IPブロックによって仮定された環境的な制約を記述するか、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。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・インター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。

6. 特殊な検証手法 — MPU設計は、常に検証ツールの能力を超えている。しかしながら、MPUの成功は経済的利益が高いため、事情は異なり、労働集約型の手法が許される。例えば、半自動の原理的性能試験技術は、quick-time-to-marketが必要なSOC設計に広範囲に使用するには、当面は遅すぎかつ、高価すぎるままであろう。しかしながら、ハイ・ボリュームのMPU設計における特定の困難な検証タスクにおいては、効果的に使われている。重要な課題は、キャパシティと予見性である。予見性は、他のシステムドライバよりもMPUの検証においてはより重要である。なぜならば、設計サイクルが非常に長い、もしくは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ある。将来に向けて計画された長い設計サイクルは、冗長な検証プロセスをも与える。しかもこれは、検証プロセスが予想もなく膨らまない限りにおいてのみである。MPU検証の複雑度を扱うための最良の追加資源は何か？検証時間の分散を短縮するための最良の追加資源は何なのか？

7. 特殊な検証容易化設計 — 同様に、MPU設計の複雑度は、他のシステムドライバよりも早く、検証容易化設計を要求する。そして、MPU設計の特殊性は、特定の領域における検証容易化設計技術が必要であることを意味する。例えば、小さな監視用のプロセッサはメイン・プロセッサの正確な実行を検証しているが、その中で、セルフチェックプロセッサは予備的な働きを担っている。課題は、MPU設計のための検証容易化設計を開発することにある。この開発は、面積と性能のペナルティを最小にすると同時に、検証コストを効果的に削減する。

8. 新しい種類のコンカレント — MPU設計はより複雑になってきており、新しい種類のコンカレント設計が重要になってきている。すでに、検証の目を逃れた多くのバグは、キャッシュの結合とその他のコンカレントな問題と関係している。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、および同時に起こるマルチスレッディングのような技術に対して、コンカレントなレベル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する。

設計検証に対する技術課題(<50nm)

1. 検証容易化設計 — 短期的な技術課題への解は、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は、どのようにしてこれらの理解を検証容易化設計として体系化す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、いくらかのパフォーマンスの低下が発生するであろう。分かりやすい例えとしては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 はスキャンベーステストがほぼ全面採用されていることがある。
2. 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。
3. 仕様における人間的要因 — 設計検証に対する継続的な課題は、設計の期待する動作をどのように表現するかである。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。さらに設計者がこれらの言語を使うトレーニングを受け、仕様を厳密に記述できるようにならなければならない。
4. フォーマル検証の適用範囲を広げる — 現在のところ、設計検証は主としてデジタルシステムの離散的な動作に焦点を当てている。シリコンの複雑化とシステムの複雑化の両方への挑戦は、将来の検証技術がシステムのより広範なレベルを解析することを要求する。低レベルでのシリコンの複雑さは VLSI システムの明確なデジタル抽象化がより不確かになることを示している。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。長期的には、複合型システム²²として扱われるデジタルとアナログの境界での、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。

テスト設計

ナノメータ・プロセス技術により、ますます高くなるクロック周波数と SOC、SIP インテグレーションは厳しい課題を DFT (Design for Test) に引き起こす。テスト産業はコンポーネントに基づいた設計の高度なテスト合成から、ノイズ干渉及び消費電力におよぶアナログ的な問題の広大な領域に対応してゆくに違いない。最適なテストバリエーションおよび DFT が設計プロセスの中で最初から考慮されて組み込まれる場合のみ、多くの問題が解決され

²² 複合型システムは連続系(例えば微分方程式モデル)と同様、離散系(例えば finite-state machine)の振る舞いを持っている。この技術分野は、古典的な制御理論と離散系形式検証の両方から技術を借りている。

る。さらには、この方法、テスト検証の分析、およびテスト設計のより深い総合力が必要である。そして、論理-回路-物理的インプリメンテーションでの分析結果から、設計検証からの資産(例えば、流れ、ベクトル、感度)は設計の間に再利用される。メモリ、SOC、AMS 及び MPU といった様々な産業セグメントを越えて、3つの高度なテストの課題が、チップ上の DFT、BIST、テストビリティの特長の拡張、及び開発のチッププランニング工程における事前計画的な製造テスト装置へのよりよい統合を必要としている。

1. 高速デバイスインターフェース — コンポーネント I/O 速度はシステム性能のために、コア・クロック周波数あるいはトランジスタおよびアーキテクチャ上性能と同様に重要になってきた。新しい I/O プロトコルが導入されており、数 GHzの領域まで拡張されている。これらの I/O のスキームは速いだけでなく、より複雑となる。(供給同期あるいは差動、さらに供給Vddレンジの10%の振幅で Gbit/s レートでの全二重動作スキームなど)それと対照的に ATEと構成要素のテスト資産は、MHz領域での同期クロックベースのテストと I/O 測定を含んでいる。従って、I/O の速度およびプロトコルは、設計検証と製造テストの両方のために使用される重要な道具、材料、ATE 設備のコスト課題、インターフェース・ハードウェア、およびテスト・ソケットをドライブする。この変曲点は広範な産業的開発と特に I/O への on-die テスタビリティ能力の適用を必要とする。ループバック、ジッタ測定、エッジ検出、擬似ランダム・パターン生成などのような I/O DFT と BIST 方法が、新たな I/O 速度とアーキテクチャの検証と製造テストに必要とされる標準的な技術となるだろう。

2. 高度に統合された SOC と SIP 設計と部品 — ブロックが均質の場合さえ(たとえばすべてがロジックのような)、より大きな統合デバイスへ向けて、すでに存在する設計ブロックの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発生させる。ますます、種々様々の回路が統合されるようになる。ロジック、SRAM および DRAM は、近年一般的に統合されてきた。しかしいまや、アナログ、ミックスドシグナルおよび不揮発性のフラッシュメモリが、ロジックや RAM と統合されようとしている。LSI の複雑度およびコストは、混載デバイスではおおよその見積りが可能である。しかし、組込みブロックおよびミックスド・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と、見積りのむずかしさを増加させる。ASIC や、より大きなロジックデバイス内に組み込まれる MPU マクロは、すでにこのインパクト(製造テストコストがシリコン・コストを超過する)を体験している。DFT でさえ、これらのコストは非線形とも考えられる。組込みブロックやマクロの I/O にアクセスするダイレクト・アクセス DFT(DAT)は、テストのデータベース再利用を可能にするが、テストのデータベース・サイズによっては、追加のテスト挿入の可能性もある。さらに、DAT を備えた組込みブロックのテストは、備えていないバージョンのテストより桁違いの長い時間を引き起こす可能性もある。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 の混載は、統合ロジックデバイス上のテストで一般的に壊される特有の最適化を備えた根本的に異なる資産から発生する。

独立系のアナログか RAM デバイスのための、非常に長いテスト時間は、異なる ATE かテスト設備コスト、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)の割合によって相殺される。統合組込み特性は、独立系のデバイスでは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、妨げるかもしれない。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、BIST の使用、および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る。組み込みミックスドシグナル、アナログ・ブロックのための DFT と BIST は研究分野では切望されており、SOC や混載デバイスでのこれらのタイプの回路の最も有力な用途にドライブされ、産業上の現実となっていくであろう。

多くの場合、異なる回路形式を同一のプロセステクノロジー(特に追加されたリソグラフィ工程を通して)におく為のコスト高は、さいころを積み上げるように、SIP によって、逃げ道を見つけることができる。SIP は SOC よりも広

い範囲のシリコン技術を迅速に、また、より短い工程と製品開発サイクルで統合することができる。また時おり、予定してなかった既存のダイを再利用することもできる。Testing end form factor SIP は、SOCと同様な技術的経済的課題を提供する。すなわち、内部ポートと、適切な部品の品質を達成する為に経済的に妥当なテスト装置と製造フローの統合である。SIPの製品サイクルは一般的により短く、DFTと、テストビリティのニーズに対する製品の“interceptability”の観点から、より限定的である。

3. 信頼度スクリーニング — 製造テストは歴史的にデバイスのパフォーマンスとファンクショナリティの測定だけでなく、潜在している信頼度欠陥の識別、および選別の作業工程を実行している。もっと正確に言えば、これらは、defect driven な初期段階の免れない信頼性故障であり、内在するデバイスまたは薄膜の信頼性ではない。動的なバーンイン、 I_{ddq} 、 V_{dd} 電圧超ストレスは、三種類の本質的なメソドロジであるが、先進的なシリコン技術の急速なリーク電流増加により厳しく限定されている。同時に、above-nominal V_{dd} の有用性も減少している。定格 V_{dd} へのストレス比率がそれぞれの技術ノードにより減少するにつれて、バーンイン、と‘on ATE’電圧ストレスの加速性、同一性の証明、選別能力が極度に制限されるようになる。130nmや 90nmでは、熱暴走が非線形のコスト増加を余儀なくさせ、バーンインの適用をマイクロプロセッサのような高付加価値製品だけに制限する。同時に、 V_{dd} 電圧ストレス、および I_{ddq} デルタのような高度な I_{ddq} 技術さえも非常に制限されている。近いうちに、著しいコスト増加は過剰品質(信号対ノイズのような)に起因する可能性がある。私たちの市場における信頼度必要条件に歩調を合わせるべきである。基本的に新しい長期的な解決策が要求され、信頼度テストあるいは特別の信頼度テスト用の重要なチップ上のハードウェアを含んでいる可能性がある。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範囲の開発および、配備もコストとデバイスの信頼性目標を達成するために要求される。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与える。たとえばファウンダリが統合テスト戦略に支援することができる程度まで、彼らは好ましいベンダになる可能性がある。生じる設計テストの困難な課題は、表 18 の中で要約されている。

表 18 Design Test Challenges

Challenges ≥ 50 nm/Through 2009	Summary Of Issues
マルチギガヘルツ・シリアルI/Oプロトコルと増加するコア周波数での効果的スピードテスト	P, S—増加するクロック周波数での連続スピードテスト P, S—増加するクロック周波数でのスピード構成テスト P, S, A—マルチギガヘルツ・シリアル I/O と非決定インターフェース用の DFT、テスト、チップ上測定技術
DFT/テストジェネレーション/フォルトグレーディングツールとデザインの複雑性における能力ギャップ	P, S—進んだ(オープン、遅延など)フォルトモデル用のより良い EDA ツール P, S—DFT は低コストの自動テスト装置を実現 P, S—やっかいでないロジック BIST (進んだフォルトモデルを含む) A—AMS DFT/BIST、特にベースバンド周波数を越えた時
テストプロセスと分析範囲による品質と歩留まり影響	P, S—テスト中の電源と熱管理 P, S—自己診断用のフォルト分析と設計 S—改善と不良解析ツールと手段 All—歩留まり悪化の原因の追究における困難度の増大
シグナル・インテグリティ・テスト性と新しいフォルトモデル	P, S—シグナル・インテグリティ・テスト性(ノイズ、浮遊容量/インダクタンスの影響等) A—アナログ(パラメトリック)故障用フォルトモデル
SOC と SIP テスト	S—チップレベル DFT に SOC テスト手法を統合 S—経済的製造工程における多様なファブテスト手法の統合 A—コアベース SOC 環境とその範囲内で共用できる DFT、BIST とテスト手法 M—自己診断と自己修復機能内蔵のエンベデッドメモリ(DRAM, SRAM, Flash) All—高集積度合いにおけるテスト再利用性
追加技術課題 < 50nm/2009 年以降	
異種の SOC と SIP 用自己テスト機能内蔵	A—チップ上でのマルチギガヘルツ RF フロントエンドのテスト S—SOC と SIP テスト用のチップ上プログラマブル資源の使用 S,A—SOC 用(RF とアナログを含む)セルフテストの解による A—デジタルからアナログ回路へのインターフェースによっておこる(アナログ)シグナル・インテグリティ・テストの課題 S—MEMS と EO 成分を含む異種の SOC と SIP 用テスト手法
分析、信頼性スクリーニングと歩留まり改善	A—AMS 部分の診断と不良解析 P,S—ゲートレベル以下での自動フォルト分析技術 P,S—欠陥をあらいだす効率の良い効果的なバーンイン用設計 P,S—テスト装置の限界能力による品質と歩留まりへの影響 P,S—欠陥とノイズ用の新しいタイミングリレーテッドフォルトモデル
欠陥範囲とオンラインテスト	P,S—ソフトウェア用 DFT と欠陥範囲内における設計 S—チップ上の再構成を使用したロジック自己修復機能 S—オンラインテスト上システムレベル

この表は上記の 4 つのトレンドによって課題を設計工程に要約している。各々の課題は最も関連のあるシステムドライバのリストで分類されている。(S-system on chip, P-microprocessor, A-analog/mixed-signal, M-memory)

テスト設計における短期的課題(> 50 nm)

1. コア周波数の高速化や GHz レートのシリアル I/O プロトコルの急速な普及に対応できる効率的なスピードテスト

- クロック周波数が高速化する MPU (ASIC/SOC) での *at-speed* 機能テストの継続 (回避)—MPU の *at-speed* 機能テストのためには、ATE のデータレートやエッジプレースメント精度が MPU 仕様より低いいため、DFT 及び BIST が必要である。対照的に、ASIC/SOC においては、BIST 化率の増加と、故障タイプの追加や故障検出率(ツール)の改善等の DFT の統合化により、肥大化する人的工数やコストを抑え続けなければならない。
- 高速化するクロック周波数による *At-speed* ストラクチャルテスト — DFT は、device より遅い ATE で、ますます複雑化するクロック方式の境界条件の中で、遷移遅延およびパス遅延をスキャンテストで可能にし続けなければならない。
- マルチ GHz のシリアル I/O プロトコルおよび非決定的なインターフェースのためのテストおよびオンチップ測定技術— Gb Ethernet, Firewire, S-ATA のようなプロトコルは、計算アーキテクチャの改善よりも

システムパフォーマンス改善が得られるため、ASIC から SOC、MPU にまで利用計画が激増している。そのようなアプリケーションは、100 mV 程度の低い電圧スイングで、数 GHz で動作する異なる信号を生成して、受け取る能力がある非常に高価なテストシステムを必要とする。既存のテスト技術利用では膨大なコストがかかるため量産が不可能であり、クリティカルで高価な ATE を利用したとしても技術的にチャレンジングである。そして、今日のコンポーネント ATE は、過去 20 年間の決定的テストパラダイムに基づいて構築されており、急速に高速化する I/O 速度や、インタラクティブかつパケットベースのプロトコルの大量の量産テストには不向きである。新 I/O テスト手法、DFT およびシリアル通信のためのオンチップ測定技術バスとコンポーネントが必要とされている。テストインターフェースハードウェアは、決定的な ATE と DUT 毎の多数の対話型プロトコルをインターフェースするための広いカスタムシリコンアプリケーションを含むであろう。非常に高速のシリアルポートに対しては、トランスミッタジッタ、クロストーク、信号ロス、ストリームにロックされるためのレシーバの PLL の能力、レシーバジッタ許容差等が、テスト問題を増加させ、オンチップのテスト機能や DFT 戦略の増加することを理解・方向付けねばならない。ループバック、補償およびオンチップ計測技術が必須になる。

2 .DFT/テスト生成 /故障評価ツールと複雑化する設計との能力ギャップ — 現在のマイクロプロセッサ、SOC および ASIC の設計データベースは、市販 EDA ツールのデータサイズ限界を越える。これは、複雑な分割と再結合を強要し、単純なスタック故障ですらフルチップ故障シミュレーションおよびスキャン ATPG を不可能なまでに難しくする。テープアウト前のフルチップシミュレーションは設計フローにおける基本的な要求であり、改訂サイクルと同様に DFT とデバイスバグ率の増加を抑える。これに代るものとして実シリコン上でのベクトル妥当性検査があるが、その方法はシリコン処理時間を設計イタレーションループに加えるため、正しくテストをするのに必要とされるエンジニアリングリソースを非常に増加させる。(更に、変更が必要となるテスト内容の割合は数 10 パーセント増加する)。

- (オープン、遅延等の)先端故障モデルのためのより良い EDA ツール — 100nm 以下の領域を利用するチップのテストに適した故障モデル(縮退故障ではない)に基づいた、実用的な故障シミュレーションと ATPG。
- 低コストの ATE を可能にするための DFT — DFT を利用して、よりシンプルな(少ピン数、限られたピンエレクトロニクス許容差、低テストデータレート、APG 無し等)ATE でパラメータと論理のテストが可能。この要求は、I/O、キャッシュおよび論理 DFT を含む。
- 邪魔をしない論理 BIST (高度な故障モデルを含む) — 不十分なデコード論理、トライステートバス衝突制御、およびその他 MPU 特有論理に対して、より許容性を持つ新たな BIST 手法が必要である。開発・製造コストを著しく上昇させることなく、テストのカバレッジと内容を充実させるために、論理 BIST は現実的な故障タイプをカバーするよう拡張されなくてはならない。ATPG についても、縮退故障と遷移故障だけではなく、オープン・ブリッジその他の故障モデルを直接扱えるように拡張されなくてはならない。

3 .テストプロセスや診断の制限による品質および歩留まりへのインパクト

- テスト時の消費電力と発熱管理 — エンドユースのような統合制御や管理(ヒート・シンク、対流、feedback bios/system integration 等)が使えないことから、製造テスト中の消費電力と発熱制御は、エンドユースのシステムニーズより厳しく制約されるかもしれない。制御レベルは、アクティブフィードバックや di/dt 緩和設計のような新しい技術を要求する。

- **故障診断と診断容易化設計** — ツール、方式および規則は、(スタック故障や遷移故障に限らず)全ての故障タイプに適用できる自動故障診断(欠陥ゲートや回路の特定と識別)を可能にするために必要である。
- **歩留まり改善及び故障解析のためのツールおよび方法** — 高コストの DSM プロセスは短い製品寿命及び短納期と相俟って、故障解析プロセスに厳しい課題を与える。これらの傾向は、故障メカニズムを能率的に理解し、修正処置を行い、そして、歩留まりを向上させるために、プロセスの任意の異常(干渉/欠陥)が迅速に指摘され、特徴付けられることを要求する。
- 各テクノロジノードが光の波長より先に行くことにより故障解析の試行ごとにより深い根本原因に直面するが、各テクノロジノードでの歩留まり改善の習熟度曲線は継続する必要がある。このためには、論理やメモリ回路において、ゲートレベルより低い抽象度(すなわち、故障ノードか結線)までの自動故障特定技術が必要である。

4. シグナル・インテグリティのテスト容易性や新しい故障モデル

- **シグナル・インテグリティのテスト容易性 (ノイズ/干渉, *capacitive/inductive coupling* 等)** — 図形はシユリンクしているものの、欠陥は、比例して収縮していない。シグナル・インテグリティの検証問題がテストの問題(例えば、分散された遅延変化、クロストークによって誘発された遅延や論理エラー、電源ネット上の過度の電圧ドロップやスイング、及び、基板や熱によるノイズ)となるため、配線レベル、及び、ワイヤ遅延の増加により新しい故障モデルが必要となる。テストは、そのようなパラメトリックな性能故障をターゲットにしなければならない。パラメータ変化によって引き起こされたノイズ故障の新しいクラスは、故障シミュレーション、ATPG、及び、BISTのためにも、電気、回路そしてトランジスタレベルよりも高い抽象のレベルに適切にモデル化されなければならない。
- **アナログ(パラメータ)故障のための故障モデル** — アナログ・ブロックやコンバータのテスト評価、DFT、及び、セルフテストによる解決策は、非常に限定的である。現状の大部分の解決策は、主として機能テストに基づいており、そして、周波数制限がある。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がない。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に対する意味のある評価指標を開発することは、きわめて重要である。多くのアナログ故障は、連続型変数(時間、電圧、位相等)で測定された仕様範囲外からのパラメータが原因であり、そして、製造上の変動やミスマッチから発生している。効率的で、効果的なアナログ故障検出、及び、テスト生成のための故障モデルが必要とされるであろう。ミックスドシグナル設計において、ツールは、高いシミュレーション精度を維持しながら故障シミュレーションの計算の複雑さを最小化することが要求される。

5. SOC と SIP テスト

- **SOC テスト手法のチップレベルDFT への統合** — 統合されたブロックか回路タイプ(アナログ、RAM、フラッシュ、論理、MEMS 等)毎の安価で専用のテストによる多数個同測と1つのプラットフォーム上に全ての測定すべきハードウェアとソフトウェアを統合する単一測定方法はトレードオフである。前のスタンドアロンブロック、もしくは、既存の設計を越えた追加の DFT は、組み込みブロックまで、そして、組み込みブロックを横断して、もしくは、特別なブロックや回路タイプにアクセスを行うために、本質的である。在庫や処理中のウエハに対しても、(多種のプラットフォームやステップによって構成される)長くてとても複雑な製造テストフローを維持することのインパクトも考察されなければならない。歴史的に、複雑な追加ハードウェアの統合は、結果として全体の ATE 開発及び投資コストの非線形的な増加をもたらしてきた。”何でもできる”装置の追加投資コストを償却することは、異なるブロックが同時にテスト

されないという点で、更に疑わしい。このように、回路特有のテストハードウェアは、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つており、製造の使用デューティサイクルの大多数の間しばしば使われていない。最低でも、組み込みブロックへの(IEEE P1500 か同等の)一貫したブロックレベルアクセスが必要とされる。結局は、SOC、SIP、及び、シリコン統合の経済性を失うことを回避するために、非線形な製造テストコストは、できる限り多くのブロックを標準化された DFT、及び、更に広い BIST ソリューションへ動かすであろう。この部分に設計エンジニアリングをフォーカスすることは、SOC/SIP 製品コスト削減に関して有意義な ROI を供給するであろう。しかし、製品企画の最も早期(アーキテクチャ検討)のフェーズの間に扱われなければならない。

- 複数の組織固有のテスト方法の統合 — 異なる設計ツールフローや設計資産からのブロックを統合する SOC や SIP は、DFT、DFT 統合、ATPG、テストデータベース検証、及び、バックエンドでのベクタ加工ツールやフローに厳しい課題を与えるであろう。MPU 環境と同様に、テープアウト前のフルチップでのテストパターンのシミュレーションは基本的な要求である。
- コアベースの SOC 環境と制約に適用可能な DFT、BIST、及び、テスト手法 — アナログ/RF DFT のため、相補型コンポーネントのセルフテスト技術が開発されなければならない。トランシーバ(RF、光素子 ADC/DAC、I/O 等)においては、送信側と受信側が相互をテストするためにしばしば使われ得る。例えば、パターン発生および応答解析の両方のための DSP ベースの(デジタル)技術では、信号を発生し且つ応答を解析するために、組み込みプロセッサまたは DSP コアと共にオンチップ DAC と ADC を用いる。潜在的な利点としては、ハードウェアオーバーヘッドの低さ、性能劣化の低さ、柔軟性の高さ、デジタル ATE が利用可能であること、がある。更に高い周波数でデバイスを扱えるように、この技術を進歩させなければならない。
- 組み込みメモリ(DRAM、SDRAM、Flash 等)に内蔵の自己診断と自己修復 — 組み込みメモリに対するこの能力は、セルフテスト適用性、冗長回路(拡張の行、列、及びブロック)、冗長解析、及び、置換論理とハードウェアを含む多くのチップ上の機能と論理を必要とする。
- テスト再利用 — SOC 設計生産性のためには、テストデータ量とテスト時間の ATE 境界条件(制限)の範囲内での、テスト IP、または、テストデータの再利用が必要である。例えば、統合された製品のテストデータセットが既存の ATE にフィットせず、追加のテスト工程が必要となるかもしれない。組み込みブロックの DFT モードは、スタンドアロンモードと同等以上に短時間でのテストを可能にすべきである。

テスト設計における長期的課題(< 50 nm)

1. 異種混載 SOC と SIP のための統合セルフテスト

- チップ上のマルチGHz 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、多大な計測及び作業工数を必要とする。テストの再利用性の改善、テスト時間の短縮、測定精度向上により、不具合対策コストを低下させるブレークスルーが必要である。
- オンチッププログラマブル・リソースの SOC セルフテストでの活用 — テストコスト削減の可能性として、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 スタイルの論理 BIST を拡張したセルフテスト機能が挙げられる。遅延/タイミングの特定の故障モードと同様に、プロセス/パラメータばらつきテストを行う必要性が増加したとき、本セルフテスト機能を用いることが、十分な品質水準を維持する為の長期的な方法である。
- (RF やアナログを含む)SOC のセルフテストソリューションへの依存性 — BIST や ATE に支援されたセルフテストが、普及しなければならない。

- 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構成によって生成された干渉は、敏感なアナログ回路のパフォーマンスを著しく崩壊させ得る。そのような現象を解析するために、デジタル回路構成要素がフルスピードで動いているときにアナログ回路をテストすることが必要である。クロストークを回避する設計手法と、4 Gbit/sec レート以上でのオペレーション中のクロストークテスト手法が開発されなければならない。
- MEMS、及び、電子光部品を含む異種混載SOC のためのテスト方法 — SOC に統合された全ての新しい回路タイプは、現存する設計及び DFT のフレームワークにフィットし、さらにはセルフテスト能力を拡張しなければならない。同じ(たいてい経済的な)理由のために、類似のテスト方法が、統合された RF、ミックスドシグナル、フラッシュ等のために近い将来に必要となる。

2. 診断、信頼性スクリーニング、及び、歩留まり改善

- アナログ/ミックスドシグナル部分の診断および故障解析 — これらは、現状では大部分が人手で実施される。特にアナログ/ミックスドシグナルコンポーネントにおいて、新しい故障診断および故障解析技術を開発し、プロセスデバッグ時間とコストを削減する必要がある。いくつかのソース(設計、プロセス、テストの各データ)からのデータを、適切な故障モデルや測定基準を用いて相関づける新しいツールや技術が必要となる。その診断手法についても、出現する DFT/BIST 方法と互換性があるべきである。
- 潜在的欠陥をたたき出すための効率的で効果的なバーンイン設計 — 潜在的欠陥を許容範囲に制御するために、バーンイン容易化設計または新代替スキーム(自己修復など)によってバーンインを緩和すべきである。
- テスト装置限界による品質と歩留まりインパクト — ATE やテストで使用する計測装置の許容誤差は、パラメータの実測範囲であるべきである。根本的に新しいアプローチなしでは、信号対ノイズ比および誤検査による歩留まり低下が、受容しがたいレベルに達するであろう。
- ナノメータ技術における欠陥/ノイズのための新しいタイミング依存故障モデル — パラメータばらつきや欠陥によるタイミングレベルでの故障動作は、ますます顕著になる。これらをモデル化し、故障シミュレーション、ATPG、DFT、BIST ツール及び手法に組み込まなければならない。

3. フォールトトレラント、及び、オンラインテスト

- 論理ソフトウェアのためのDFTおよびフォールトトレラント設計 — 特に論理におけるソフトウェア緩和は、現状では限られた解しか無く、回路やアーキテクチャでの解決策が必要とされる。
- オンチップのリコンフィギュラブル回路を用いた論理自己修復 — システムレベルの信頼性を向上し、また増加するソフトウェアの影響を緩和するには、チップ上の論理回路に加えて、自己テスト、自己診断、自己修正、および、自己修復に十分な開発工数(新方式と既知技術の統合の両方)を投じる必要がある。
- 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ストラクチャルテストに移るに従い、システム全体としての機能テストは、ATE 上での専用の機能試験ではなく、アプリケーションに対応したテストによる試験に移行するだろう。移行を促す要因は 2 つある。第一に、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ったことである。第二に、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、使用状態での SOC のアプリケーションベースのテスト、即ち、顧客がアプリケーションで使用するのと同じシステムおよび方法を用いたテストがシステムの品質および信頼度に対する期待に最も効果的にマッチすることである。

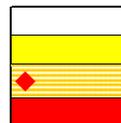
Additional Design Technology Requirements

表19 Additional Design Technology Requirements

Year of Production	2003	2004	2005	2006	2007	2008	2009	2012	2015	2018	Driver
Technology Node		hp90			hp65						
DRAM ½ Pitch (nm)	100	90	80	70	65	57	50	35	25	18	
MPU/ASIC ½ Pitch (nm)	107	90	80	70	65	57	50	35	25	18	
MPU Printed GATE Length (nm)	65	53	45	40	35	32	28	20	15	10	
MPU Physical GATE Length (nm)	45	37	32	28	25	22	20	14	10	7	
SOC new design cycle (months)	12	12	12	12	12	12	11	11	10	9	SOC
SOC logic Mtx per designer-year (10-person team)	1.9	2.5	3.3	4.3	5.4	7.4	10.6	24.6	73.4	113	SOC
SOC dynamic power reduction beyond scaling (X)	0	0.1	0.1	0.2	0.2	0.2	0.2	6	4.7	8.1	SOC
SOC standby power reduction beyond scaling (X)	0.37	1.4	2.4	3.4	5.1	6.4	8.73	18.8	44.4	232	SOC
%Test covered by BIST	20	20	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、チャン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)インターコネクタのカップリングといった設計パラメータ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、チャン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:DT コストと価値

図 20 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(図 20 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(従って、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。

図 20 Simplified Electronics Product Development Cost Model

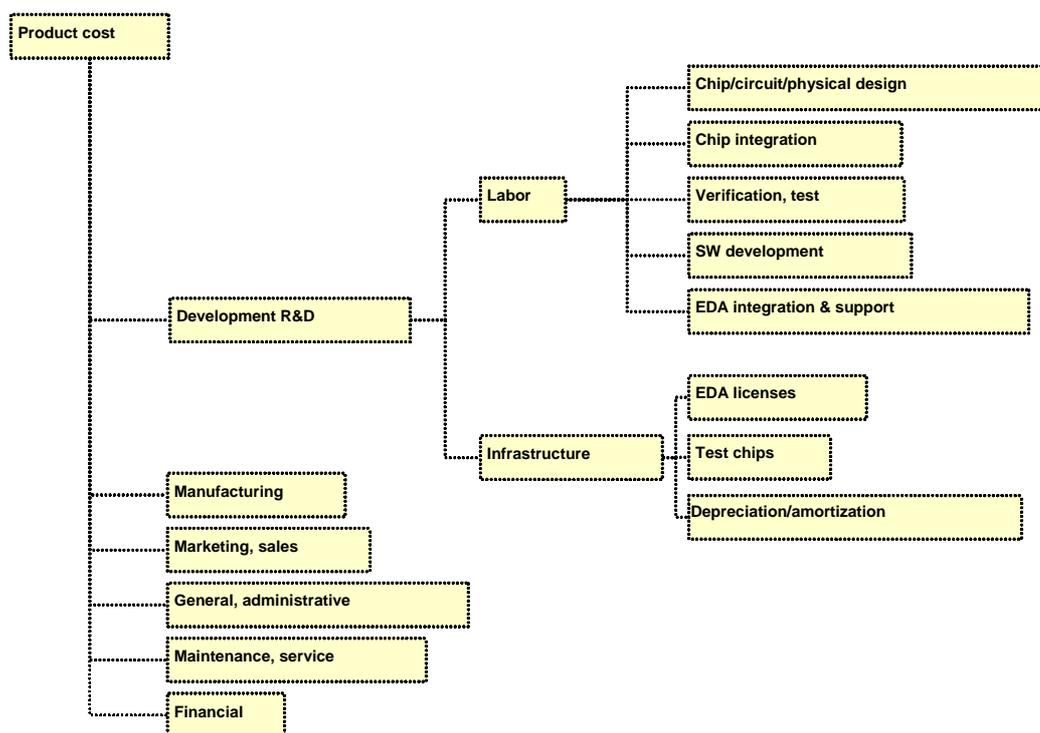


図 20 において、ボールド体の項目は設計コストと見なされる(機会コストとロスコストは含まれない)。この図は、製品開発費はおおよそ直接的な労働コストとインフラストラクチャコストに分解されることを示している。労働コストは以下の項目を含む。チップ、回路、及び、レイアウト/物理設計;チップインテグレーション;検証(バリフィケーション)とテスト;ソフトウェア開発;EDA インテグレーション;ソフトウェアと技術サポート。一方、インフラストラクチャコストは、設計用のソフトウェアライセンス(ソフトウェア開発環境を含む)、テストチップ・インフラストラクチャと

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 メントロジ」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表 20 に示されており、その中でグレーの項目は、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 2003 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)は、1990 年の 250 人から 2003 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。

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。

表 20 Design Technology Improvements and Impact on Design Productivity

<i>DT Improvement</i>	<i>Year</i>	<i>Productivity Delta</i>	<i>Productivity (GATEs/Desn-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 GDS II through IC place and route.
Intelligent testbench	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
TOTAL		+15,000%			

概算のトータル設計コスト

図 13 は、システムドライバ章で定義したドライバであるロー・パワー System-on-Chip (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る。SOC-LP は 2003 年で 4.9M 論理ゲートを搭載していることから SOC PDA の設計コスト(設計者とツールの費用の合計)は \$20.1M となる。1993 年から 2003 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 \$629.8M 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。²³

²³ 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、メンドロジの相違が原因であると考えられている。