

国際半導体技術ロードマップ

2009 年版概要

INTERNATIONAL
TECHNOLOGY ROADMAP
FOR
SEMICONDUCTORS

2009 EDITION

DESIGN

THE ITRS IS DEVISED AND INTENDED FOR TECHNOLOGY ASSESSMENT ONLY AND IS WITHOUT REGARD TO ANY
COMMERCIAL CONSIDERATIONS PERTAINING TO INDIVIDUAL PRODUCTS OR EQUIPMENT.

訳者まえがき

この文書は International Technology Roadmap for Semiconductors 2009 Edition(国際半導体技術ロードマップ 2009 年版)本文の全訳である。

国際半導体技術ロードマップ(以下 ITRS と表記)は、米国、日本、欧州、韓国、台湾の世界5極の専門家によって編集・作成されている。日本では、半導体技術ロードマップ専門委員会(STRJ)が電子情報技術産業協会(JEITA)内に組織され、日本国内で半導体技術ロードマップについての調査活動を行うとともに、ITRS の編集・作成に貢献している。STRJ 内には 14 のワーキンググループ(WG: Working Group)が組織され、半導体集積回路メーカ、半導体製造装置メーカ、材料メーカ、大学、独立行政法人、コンソーシアムなどから専門家が集まり、それぞれの専門分野の調査活動を行っている。

ITRS は改版を重ねるごとにページ数が増え、2009 年版は英文で約 1000 ページの文書となった。このような大部の文書を原文で読み通すことは専門家でも多大な労力を要するし、専門家であっても技術分野が少し異なると ITRS を理解することは必ずしも容易でない。STRJ の専門委員がその専門分野に応じて ITRS を訳出することで、ITRS をより親しみやすいものにすることができるのではないかと考えている。

なお、ITRS 2005 年版(英語の原書)までは、ウェブ公開とともに、印刷された本としても出版していたが、ITRS 2007 年版以降、は印刷コストが大きくなってきたこと、ウェブ上で無料公開されている文書の出版版を本の形で有償頒布しても需要が限られることなどのため、印刷物の形での出版を断念し、ウェブ公開のみとなった。ITRS の読者の皆様にはご不便をおかけするが、ご理解願いたい。ITRS 2009 年版の作成にあたっては、当初から電子媒体で ITRS を公開することを前提に編集を進めた。ITRS の表は原則として、Microsoft Excel のファイルとして作成し、そのまま公開することにした。

ITRS は英語で書かれている。日本語訳の作成は、STRJ 委員が分担してこれにあたり、JEITA の STRJ 担当事務局が全体の取りまとめを行った。訳語については、できる限り統一するように努めたが、なお、統一が取れていないところもある。また、訳者によって、文体が異なるところもある。ITRS の原文自体も多くの専門家による分担執筆であり、そもそも原文の文体も一定していないことも、ご理解いただきたい。誤訳、誤字、脱字などが無いよう、細心の注意をしているが、短期間のうちに訳文を作成しているため、なお間違いが含まれていると思う。また、翻訳の過程で原文のニュアンスが変化してしまうこともある。訳文についてお気づきの点や、ITRS についてのご批判、ご意見などを事務局まで連絡いただけますよう、お願い申し上げます。

今回の訳出にあたっては、ITRS の本文の部分のみとし、ITRS 内の図や表の内部の英文は訳さないでそのまま掲載することとした。Executive Summary の冒頭の謝辞(Acknowledgments)に、ITRS の編集にかかわった方々の氏名が書かれているが、こども訳出せず、原文のままの表記とした。原文中の略語については、できるかぎり、初出の際に、「ITRS(International Technology Roadmap for Semiconductors)」のように()内に原義を示すようにした。英文の略号をそのまま使わないで技術用語を訳出する際、原語を引用したほうが適切と考えられる場合には、「国際半導体技術ロードマップ (ITRS: International Technology Roadmap for Semiconductors、以下 ITRS と表記)」「国際半導体技術ロードマップ (International Technology Roadmap for Semiconductors)」のように和訳の後に()内に原語やそれに対応する略語を表示した。Executive Summary の用語集(Glossary)も参照されたい。原文の括弧()があってそれを訳するために括弧を使った場合もあるが、前後の文脈の関係で判別できると思う。また訳注は「【訳者注:この部分は訳者の注釈であることを示す】」のように【】内に表記した。また[]内の部分は、訳者が原文にない言葉をおぎなった部分であることを示している。訳文は厳密な逐語訳ではなく、日本語として読んで意味が通りやすいように意識している。ITRS のウェブ版ではハイパーリンクが埋め込まれているが、今回の日本語版ではハイパーリンクは原則として削除した。読者の皆様には不便をおかけするが、ご理解いただければ幸いである。

今回の日本語訳全体の編集は全体のページ数が膨大であるため、大変な作業となってしまいました。編集作業を担当いただいた、JEITA 内 STRJ 事務局の古川昇さん、関口美奈さんに大変お世話になりました。

厚くお礼申し上げます。

より多くの方に ITRS をご活用いただきたいとの思いから、今回の翻訳作業を進めました。今後とも ITRS と STRJ へのご理解とご支援をよろしくお願い申し上げます。

2010年5月

訳者一同を代表して

電子情報技術産業協会 (JEITA) 半導体部会 半導体技術ロードマップ専門委員会 (STRJ) 委員長
石内 秀美 (株式会社 東芝)

版權について

ORIGINAL (ENGLISH VERSION) COPYRIGHT © 2009 SEMICONDUCTOR INDUSTRY ASSOCIATION

All rights reserved

ITRS • 2706 Montopolis Drive • Austin, Texas 78741 • 512.356.7687 • <http://public.itrs.net>

Japanese translation by the JEITA, Japan Electronics and Information Technology Industries Association
under the license of the Semiconductor Industry Association

—引用する場合の注意—

原文(英語版)から引用する場合: ITRS 2009 Edition page XX, Figure(Table) YY
この日本語訳から引用する場合: ITRS 2009 Edition (JEITA 訳) XX 頁,図(表)YY
と明記してください。

問合せ先:

社団法人 電子情報技術産業協会
半導体技術ロードマップ専門委員会 事務局
Tel: 03-5218-1068 mailto: roadmap@jeita.or.jp

TABLE OF CONTENTS

スコープ	1
全体的な課題	2
詳細な設計技術の課題	7
デザインメソドロジー	7
システムレベル設計	9
論理、回路、物理設計	13
設計検証	18
テスト設計	26
製造容易化設計 (Design For Manufacturability (DFM))	30
多様化(More than Moore)の分析	37
アナログ、ミックスドシグナル、RF設計技術のトレンドと課題	37
クロスカットTWGの課題	40
モデリングとシミュレーション	40
付録 I: ばらつきのモデリングとロードマップ	41
付録II: DTコストと価値	42

LIST OF FIGURES

図 DESN 1	IMPACT OF DESIGN TECHNOLOGY ON SOC CONSUMER PORTABLE IMPLEMENTATION COST	1
図 DESN 2	THE V-CYCLE FOR DESIGN SYSTEM ARCHITECTURE	1
図 DESN 3	HARDWARE AND SOFTWARE DESIGN GAPS VERSUS TIME	1
図 DESN 4	SYSTEM-LEVEL DESIGN POTENTIAL SOLUTIONS	11
図 DESN 5	EVOLVING ROLE OF DESIGN PHASES IN OVERALL SYSTEM POWER MINIMIZATION	13
図 DESN 6	LOGICAL/CIRCUIT/PHYSICAL DESIGN POTENTIAL SOLUTIONS	17
図 DESN 7	DESIGN VERIFICATION POTENTIAL SOLUTIONS	24
図 DESN 8	VARIABILITY-INDUCED FAILURE RATES FOR THREE CANONICAL CIRCUIT TYPES ..	1
図 DESN 9	DESIGN FOR MANUFACTURABILITY POTENTIAL SOLUTIONS	1
図 DESN 10	MOORE AND NON-MOORE DESIGN TECHNOLOGY IMPROVEMENTS	1
図 DESN 11	POSSIBLE VARIABILITY ABSTRACTION LEVELS	1
図 DESN 12	SIMPLIFIED ELECTRONIC PRODUCT DEVELOPMENT COST MODEL	1

LIST OF TABLES

表DESN 1	OVERALL DESIGN TECHNOLOGY CHALLENGES	5
表DESN 2	SYSTEM-LEVEL DESIGN REQUIREMENTS	10
表DESN 3	CORRESPONDENCE BETWEEN SYSTEM-LEVEL DESIGN REQUIREMENTS AND SOLUTIONS	12
表DESN 4	LOGICAL/CIRCUIT/PHYSICAL DESIGN TECHNOLOGY REQUIREMENTS	14
表DESN 5	CORRESPONDENCE BETWEEN LOGICAL/CIRCUIT/PHYSICAL REQUIREMENTS AND	

SOLUTIONS	17
表DESN 6 DESIGN VERIFICATION REQUIREMENTS	20
表DESN 7 CORRESPONDENCE BETWEEN DESIGN VERIFICATION REQUIREMENTS AND SOLUTIONS	25
表DESN 8 DESIGN FOR TEST TECHNOLOGY REQUIREMENTS	26
表DESN 9 DESIGN FOR MANUFACTURABILITY TECHNOLOGY REQUIREMENTS.....	32
表DESN 10 CORRESPONDENCE BETWEEN DESIGN FOR MANUFACTURABILITY REQUIREMENTS AND SOLUTIONS	36
表DESN 11 DESIGN TECHNOLOGY IMPROVEMENTS AND IMPACT ON DESIGNER PRODUCTIVITY	45

デザイン

スコープ

設計技術(Design Technology)は、電子機器システムの**構想設計**、**実装設計**および**設計検証**を実現する技術である。設計技術はツール、ライブラリ、プロセス特性のモデリングおよび**設計手法**で構成される。電子機器システム設計者の設計構想と設計目標は、設計技術によって製造可能かつ試験可能な形に変換される。設計技術の役割は、製造能力を最大活用したコスト効率の良い生産により、半導体産業の利益および成長を実現することである。2009 年版 ITRS では、設計 ITWG(International Technology Working Group)が、総括ロードマップ技術指標(Overall Roadmap Technologies Characteristics)に基づいて、クロック周波数、レイアウト密度、消費電力などの観点で、デザイン章とシステムドライバ章を作成した。設計技術の課題と設計者からの改善要求は両章に記載されている。また本章の読者には前版の ITRS デザイン章を参照することを推奨する。前版のデザイン章は現在でも有益な内容を多く含む。

2009 年版の主要メッセージ — **設計コストは半導体技術ロードマップに対する最も大きな脅威である。**
設計対象を、プログラマブルな汎用プラットフォーム上に実現するか、あるいは個別 IC を新規開発するか、また機能をソフトウェアで実現するか、ハードウェアで実現するかという選択を最適に実行することが、コスト競争力をもたらす。開発費用 NRE(Non-recurring Expenses)はマスク、プローブカード費用で数 100 万ドルに達する。さらに設計 NRE として、設計ミスによるチップ再試作のための製造 NRE を含めると数千万ドルに達する。技術革新のスピードアップは製品寿命を短縮し、LSI 顧客にとって **time-to-market** が重要目標となる。また、製造に要する期間が数週間であることに対して、設計と設計検証期間は不確定であり、数か月または数年を必要とする。

前版の ITRS より**設計生産性のギャップ**を記載している — 有効に設計できるトランジスタ数に比べて、利用できるトランジスタ数が急激に増加している。製造工場リスク償還(risk foundry amortization)、供給者産業の費用対効果(ROI)、そして実に半導体投資サイクル全体において、このギャップは半導体価格にインパクトを与える。それでも、プロセス技術への投資額は設計技術への投資額に比べて多大である。設計技術ロードマップは、図 DESN1 に示すように設計コストを制御することを可能にする。

1) ハードウェア設計については、検証規模とテストコストが急上昇し続け、製造性考慮設計(DFM)が設計フロー全体に渡って考慮が必要な問題となっている。設計技術の革新により、ハードウェア設計のコストが抑制されている。システムドライバ章で定義されている携帯コンシューマ SoC(SoC-CP)での 2009 年の推定値は 1570 万ドルである一方、1993 年から 2007 年の間に設計技術の革新が行われなかった場合には、約 18 億ドルに達する。

2) ソフトウェア設計については、ホモジニアス、ヘテロジニアス双方によるマルチコア設計の時代に入っており、今や組込みシステム開発コストの 80%以上を占める。2009 年におけるソフトウェア設計全体のコストは、ハードウェア依存ソフトウェア(hardware-related software)の開発コストとあわせて、約 4530 万ドルに達する。ソフトウェア設計については、次の 15 年間に多くの設計技術の革新が必要になる。

設計技術の革新のロードマップが効果的に開発できなかつたり、展開できなかつたりすると、半導体産業の長期的な成長のトレンドを壊すことになる。したがって、我々はこれらの設計技術のギャップを次の 15 年に対処すべき重要な危機だと見ている。

本章では、始めに設計技術における複雑度の課題である**シリコン複雑度**および**システム複雑度**について考察する。続いて設計技術領域の全体に関わる 5 つの**横断的な課題**を紹介する。即ち、設計生産性、消費電力、

2 デザイン

製造容易性、寄生効果による干渉、および故障を起こさない高信頼設計である。設計技術ロードマップとして、**定量的に記述した設計技術要求表(Requirements Table)**と、その解決技術表(Solutions Table)を示す。また、設計フローに基づいて設計技術の課題を分析する。即ち、設計工程、システムレベル設計、論理/回路/物理設計、設計検証、テスト設計およびDFM(製造容易化設計)を詳述する。¹ これらの技術課題については供給側とりわけEDA業界のマネジメント、研究開発部門および大学研究機関といった対応組織ごとに分析する。さらにシステムドライバであるMPU、SoC、AMS(Analog Mixed Signal)、メモリの分野別に課題分析する。ここでは現在のEDA技術およびマーケット規模を反映して、MPUとSoCの設計課題が詳述される。最後にAMS設計に特化した設計技術についても紹介する。本書全体の章構成や考察の加え方は特定用途向け、特定システムドライバ向け設計技術の高まりを映し出したものとなっている。

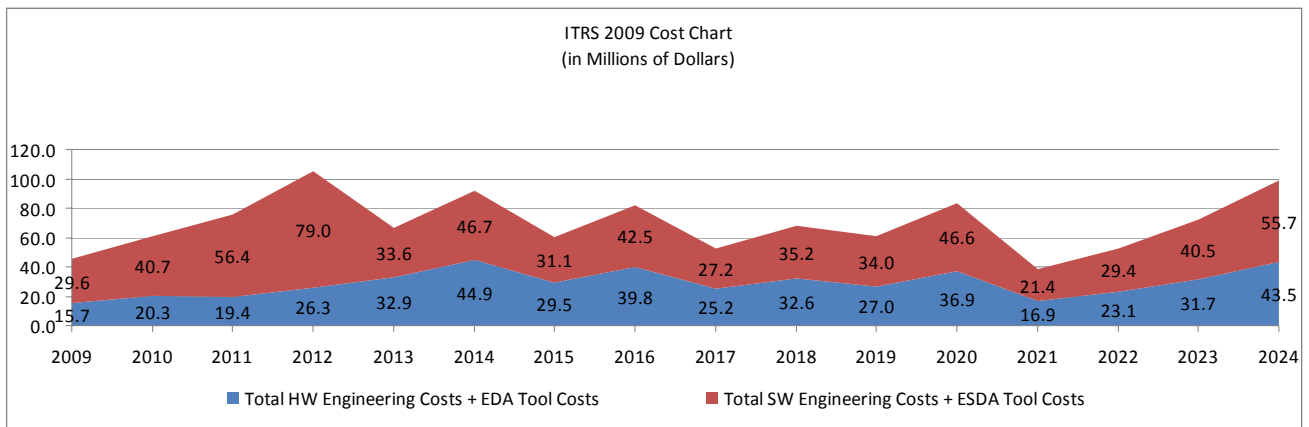


図 DESN 1 Impact of Design Technology on SOC Consumer Portable Implementation Cost

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、潜在的なアプリケーションによる未確定な設計目標、そして、設計最適のための多目的な性質に向き合いながら、市場要求に適合するようにいかに最適化するか为主要テーマとなる。レイアウト密度、動作周波数、消費電力、テストビリティあるいは設計期間のように多数のトレードオフを最適化するのは困難であるため、設計技術には経験則が付きものとなる。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、必要とされるすべての要素技術が整ったときに ITRS 技術世代が設定されるのに対し、設計技術はたとえ単独の要素技術の革新であっても、設計生産性や設計の質的改善をもたらすことが可能であり、また設計技術が開発されたときに展開される。

全体的な課題

設計技術は 2 種類の複雑度に直面する。ー シリコン複雑度およびシステム複雑度ーである。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、ならびに新規の材料、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

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。テスト装置および製造テスト技術はテスト章で述べられる。一方、BIST(Built-in self test)を含めたテスト容易化設計技術はこの章で詳述する。

グ、クロス・カップリングによる遅延変動)

- * 製造ばらつき (統計的なプロセスモデリングと特性抽出、歩留り、リーク電力、ならびに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への影響)
- * 製造ハンドオフの複雑度 (レチクル改良、マスク描画と検査方法、NRE(Non-Recurring Engineering)コスト)
- * デバイス性能と関連のある、グローバル配線性能のスケーリング(通信、同期)
- * 信頼性の低下 (ゲート絶縁体のトンネリングおよびブレイクダウン特性、ジュール熱およびエレクトロマイグレーション、単発的な擾乱、フォールトトレラント)

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(例えば、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(例えば 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、設計技術者に加え、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設計の複雑さは、システムレベルのSOC設計に起因する多様性の形態を示すものである。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。

- * 再利用一階層設計、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
- * 組み込みソフトウェア設計—予測可能なプラットフォーム・ベースのシステム設計手法、ハードウェアとネットワークにつながれたシステム環境間での協調設計、ソフトウェア解析と検証
- * 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
- * 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

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、システム的な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)。

4 デザイン

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

5 つのクロスカット課題—1)設計生産性、2)消費電力管理、3)製造容易性設計、4)干渉および、5)信頼性—が与えられるが、これらの解決策候補(Potential Solutions)は設計技術の全分野にまたがり、かつ、この背景には設計コストについての課題が存在する。この中の最初の 3 つは、ITRS Executive Summary 章で困難な課題として特に取り上げられている。設計生産性(これは“低コスト生産”の課題である)は、システムと設計のプロセスの複雑さに密接に関連し、短期的にも長期的にも最大で最重要な設計技術の課題である。消費電力管理(これは“性能向上”の課題である)は、短期的には性能を左右するダイナミック電流の問題と、ばらつきに左右されるリーク電流の問題の間を行き来している。製造容易性設計(これは“低コスト生産”の課題である)は大量のチップを許容できるコストと期間で生産するための要求であり、かつてはリソグラフィの装置の制約や限界に特化されていたが、これからは多様なばらつきの問題が深刻化しており、歩留まり管理やテスト容易化設計とあわせて、DFM の徹底した統合化が必要になる。表 DESN1 にクロスカット設計課題の主要な項目を要約する。

クロスカット課題1—設計生産性

設計コストの指数関数的な増加を避けるため、チップに設計される機能の全体的な生産性を、技術ノードごとに 2 倍以上にしなければならない。そのため、設計、検証およびテストにおいて、再利用生産性(マイグレーションや AMS(Analog Mixed Signal)RF コアの再利用を含む)を、技術ノードごとに 2 倍以上にしなければならない。設計生産性の向上における考慮すべき点は、次の通りである。(1) 検証の改善(現状、危機的な状態にあるボトルネックである) (2) より高位レベルでのシステム設計のハンドオフを支援する、信頼できかつ予測可能なシリコン・インプリメンテーションの実現 (3) SOC 生産性に対する最も重要な課題として現れた、組込みソフトウェア設計の改善 (4) 特に MPU 混載において、巨大かつ分散した組織で様々な供給元からの設計ツールによる作業での、生産性の改善 (5) SOC および AMS システムドライバによって要求される、アナログ、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が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。

表 DESN 1 Overall Design Technology Challenges

Challenges ≥ 22 nm	Summary of Issues
Design productivity	System level: high level of abstraction (HW/SW) functionality spec, platform based design, multi-processor programmability, system integration, AMS co-design and automation Verification: executable specification, ESL formal verification, intelligent test bench, coverage-based verification Logic/circuit/physical: analog circuit synthesis, multi-objective optimization Logic/circuit/physical: SiP and 3D (TSV-based) planning and implementation flows Heterogeneous component integration (optical, mechanical, chemical, bio, etc.)
Power consumption	Logic/circuit/physical: dynamic and static, system- and circuit-level power optimization
Manufacturability	Performance/power variability, device parameter variability, lithography limitations impact on design, mask cost, quality of (process) models ATE interface test (multi-Gb/s), mixed-signal test, delay BIST, test-volume-reducing DFT
Reliability	Logic/circuit/physical: MTTF-aware design, BISR, soft-error correction
Interference	Logic/circuit/physical: signal integrity analysis, EMI analysis, thermal analysis
Challenges < 22 nm	Summary of Issues
Design productivity	Verification: complete formal verification of designs, complete verification code reuse, complete deployment of functional coverage Tools specific for SOI and non-static logic, and emerging devices Cost-driven design flow
Power consumption	Logic/circuit/physical: SOI power management Logic/circuit/physical : Reliability- and temperature-constrained 3D physical implementation flows
Manufacturability	Uncontrollable threshold voltage variability Advanced analog/mixed signal DFT (digital, structural, radio), “statistical” and yield-improvement DFT Thermal BIST, system-level BIST
Reliability	Autonomic computing, robust design, SW reliability
Interference	Interactions between heterogeneous components (optical, mechanical, chemical, bio, etc.)

ATE—automatic test equipment BISR—built-in self repair BIST—built-in self test DFT—design for test

EMI—electromagnetic interference

ESL—Electronic System-Level

HW/SW—hardware/software

MTTF—mean time to failure

SOI—silicon on insulator

クロスカット課題2—消費電力管理

プレーナ型 CMOS デバイスの非理想的スケーリングは、配線材料やパッケージ技術のロードマップとともに、パワーマネジメントと電流供給に関連した様々な課題を提示する。(1) システムドライバ章における MPU と SoC-CP(Consumer Portable)ドライバとともに、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中で、動作時およびスタンバイ時の電力は増加しないことが求められている。設計技術は、結果として生じるパワーマネジメント・ギャップの対策に取り組まねばならない。(2) 電力密度の増加は、信頼性と性能に対する熱の影響を悪化させる。他方で、供給電圧の減少は、リーク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(例えば、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_{th} 、マルチ Tox 、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－製造容易性設計

「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、設計技術の適切な相乗効果によって、(より経済的に)解決できる可能性がある。製造容易性設計は設計と製造に関する他のすべての分野との相互関係によって登場した。製造容易化設計については、この設計技術ロードマップで一つの節全体を割いて説明している。将来の技術ノードの実現可能性は、このような技術のつながりに拠ることになる。以下に、いくつかの例を示す。(1) テスタ装置のコストおよび処理限界に関する問題は、新規の故障モデル(例えば、クロストークやパス遅延の検証用途)のより速やかな採用により、解決が図られるであろう。本故障モデルには、対応する自動テストパターン生成(ATPG(Automatic Test Pattern Generation))およびビルトイン・セルフテスト(BIST(Built-In Self Test))技術も含まれる。(2) システム・インプリメンテーション・コスト、性能検証、および全体的な設計 TAT は、チップ・パッケージ・ボード最適化および解析技術のみならず、システム・イン・パッケージ設計技術により、改善される可能性がある。(3) リソグラフィ、PIDS(Process Integration, Devices and Structures)、フロントエンドプロセス(FEP)、および配線技術の、CDコントロール(Critical Dimension Control)に対する要求は、製造ばらつき下における正確さを保証できる新しい設計技術により、緩和される可能性がある(例えば、ばらつきを考慮した回路設計、規則的なレイアウト、タイミング構造最適化、および静的な性能検証)。(4) 製造 NRE コストは、マスク生産と検査フローへの、より知的なインタフェースにより、削減される可能性がある。

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既に試みられている、リソースを有効活用する通信と同期方式は、ノイズと干渉により、より一層妨げられる。論理設計、回路設計、およびフィジカル設計において広く普及しているシグナル・インテグリティ・メソッドは、明らかに 100nm ノードまでは拡張できるものの、実用的な限界に差しかかりつつある。これらのメソッドは、長い配線へのリピータ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/グラウンド分配設計、などを含む。ミックスドシグナルや RF コンポーネントのスケーリングと SOC 化については、より柔軟かつ強力なメソッドが要求される。今後の検討課題として、ノイズの無歪限界(特にロー・パワーデバイスやダイナミック回路)、多数の容量結合もしくは誘導結合を含む配線、供給電圧の IR ドロップとグラウ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－信頼性と耐障害性

生産、検証、テストにかかるコストを劇的に削減するには、デバイスや配線に対し 100%の完成度を求めることを緩めなければならない。このようなパラダイム・シフトは、技術スケーリングが進むにつれて、あらゆる場合において加速されるであろう。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) 65nm 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や高抵抗バIAS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レラントが含まれる。

詳細な設計技術の課題

この章の残りの部分では、デザインメソドロジの概要と、5 つの主要な設計技術領域における定量化した課題と解決策の候補について述べる。上述したように、今日の EDA 技術とマーケットセグメントを反映して、ほとんどの課題は SOC にマッピングされる。

デザインメソドロジ

チップを設計し実装する工程には、設計者の適切な入力により製造可能な製品とするために、多くの技術の集積や設計ツールおよび有効な設計手法が必要となっている³。必要となる設計ツールは大きく注目されてきたが、それと同等に重要な設計メソドロジはなおざりにされてきた。テクノロジー世代毎に設計者は新たな問題に取り組むことを求められ、そのため、新たな現象を評価し設計者のクリティカルな設計上の判断を支援する、新しい解析手法とツールが開発されなければならない。さらに大きな課題は、手戻りを最小限にするために、問題を検討する最も効率的な順序を決定して、設計上の決定を行うことである。

“More Moore”、“More Than Moore”、“Beyond COMS”というマイクロエレクトロニクスからナノエレクトロニクスへの変遷によって、シリコン・システムの設計に避けることのできないパラダイム・シフトが生じている。これらは設計プロセスのすべてのレベルに影響を与え、新しいメソドロジやツールに向けた、とてつもなく大きな努力が要求される。設計技術は、ナノエレクトロニクスで与えられる様々な可能性を活用した、高度に複雑で同時にコスト効率の高いシリコン・システムを実現しなければならない。革新的なアプリケーション（詳細はシステムドライバ章を参照）が手頃なものとなるように、困難な EDA の課題を解決しなければならない。シリコンデバイスのシュリンクとそれに関する製造プロセスは、今までは日常的に浸透した強力なシリコン・ソリューションだった。このようなシリコン・ソリューションの展開は、スムーズに仕様データから製造可能なレイアウト・データに変換し、異なる抽象度レベルで必要な検証を行える設計技術と EDA 技術の上に立脚していた（図 DESN2）。しかしながら、こうした設計技術が継続して使えることは、もはや当たり前のことと思ってはならない。

³ 設計メソドロジは設計者と設計技術者が協力して開発される；それは制約条件を満たしながら確実にデザインを設計目標に「可能な限り近い」ように作る設計プロセスの一連のステップである。設計メソドロジは設計技術とは異なるもので、設計技術はメソドロジを含む一連のステップの実装に関するもので、各々の設計技術分野の中で議論されている。すべての既知となっている設計メソドロジは、1) トップダウンの計画と探索によりシステム仕様と制約条件を遵守し、2) 物理法則による制約、設計と製造技術の限界、システムコストの限界をボトムアップに伝播させたものを組み合わせたものである。

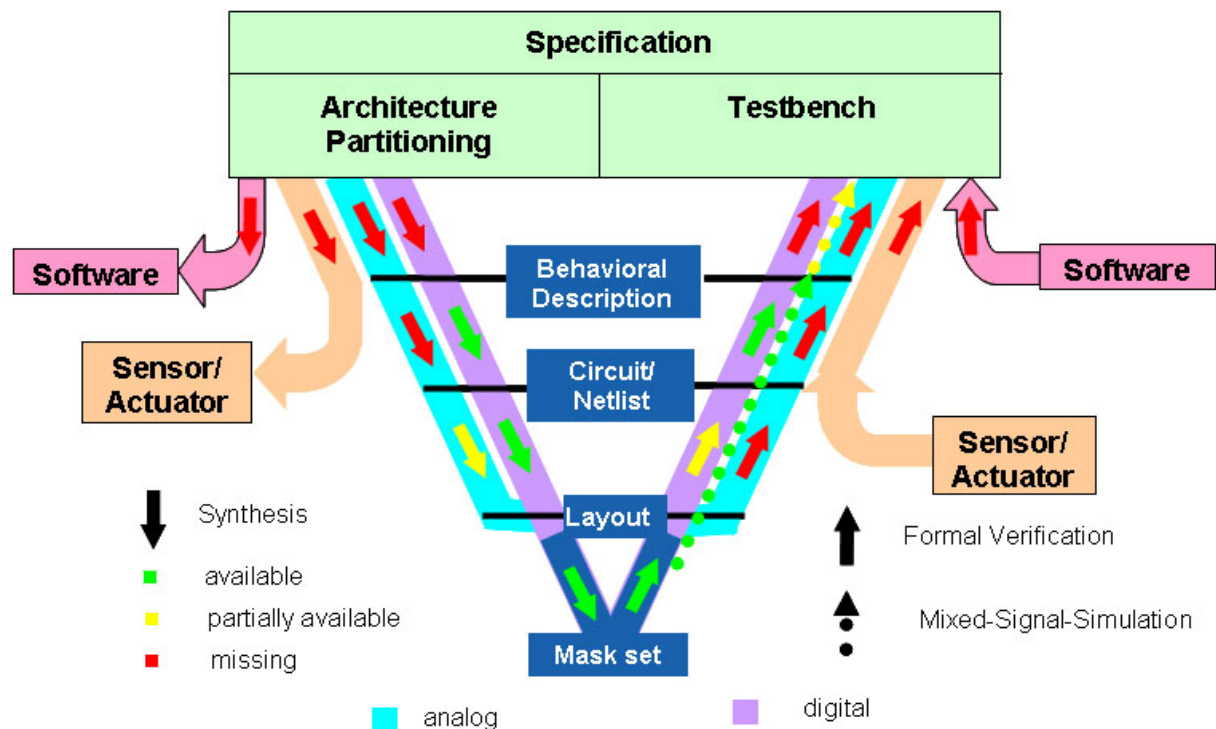


図 DESN 2 The V-Cycle for Design System Architecture

新規 SOC のコストや開発期間を満たすには、インプリメンテーションのパス(左)と検証のパス(右)の二つの主要なパスで構成される複雑な設計プロセス(図 DESN2)の全体にわたる設計技術を必要とする。この図は、”More Moore”と”More than Moore”の両方の設計を統合した設計システム・アーキテクチャの V-サイクルと呼ばれるものである。図中の矢印は、最先端の設計システム環境において、緑色が利用可能、黄色が部分的に利用可能なことを示している。赤い矢印は、EDA に対する将来の要求を示している。

システム仕様はアーキテクチャからマスクまで段階的に詳細化される。SOC 設計とは、まず初期段階でシステムを探索し、システムレベル記述をさらに動作レベル、回路／ネットリストレベル、レイアウトレベル、マスクレベルへと詳細なレベルに変換して、各詳細化ステップの正しさを検証メソッドロジが保証しなければならないものである。SOC の構成部品と動作だけではなく、システム・ソフトウェアやセンサー、アクチュエータなども、将来のインプリメンテーションと検証の工程で扱わなければならない(赤色)。設計システムのデジタル設計パス(図中の紫色)では、動作レベルより高い抽象度(赤)のツールやメソッドロジが必要になる。アナログ設計パス(いくつかの赤の矢印で示す)は、どの抽象度レベルでも完全な解決策がほとんど見つけられていないため、さらに実現が困難である。結果として、アナログ設計工程の自動化は、現在の設計環境では、まだ解のない課題である。まとめると、将来の要求は次の 2 要素となる。1) More than Moore に対しては、現状の SOC 設計フローに含まれていない、ソフトウェアやセンサー/アクチュエータなどを統合することが必要で、これらは赤色にしている。2) 将来に向けて、さらに高抽象度のデジタル/アナログ設計フローをサポートするツールによる、新しい SOC 設計フローが必要である。これは、左側の高位レベル仕様からの動作合成と、これと対応する右側の高抽象レベルの検証ステップを必要とすることになるが、いずれも赤色としている。アナログ、ミックスドシグナルのフローではさらに、すべての設計レベルでのアナログ回路合成とアナログ回路自動検証が必要となる。

⁴ ここでの議論の原点は元々は European Nanoelectronics Initiative Advisory Council (ENIAC) の設計章の中の Strategic Research Agenda (SRA) の最新のアップデートとして提起された。図 DESN2 は自動車の観点からの EDA 設計フローを示したもののだが、十分に将来の EDA 要求を表現している。この図は Peter van Staa (Bosch) による。

ナノテクノロジーへの移行によって、仕様から製造までのすべての設計工程が、大きく相互依存関係を高めるだけでなく、最終製品である IC の歩留まりや信頼性に密接に関係する。その結果として、SOC の設計生産性は、ムーアの法則に基づくナノエレクトロニクス技術の革新のペースに、簡単には追従できない(図 DESN3)。劇的に設計生産性を向上させるために、いわゆるシステムレベルと呼ばれる、さらなる高抽象度のレベルが導入された。次節では、新しいシステムレベルの課題と解決策を記載する。

システムレベル設計

数十年の間、設計者は、ほとんど設計自動化ツールのサポートなしで、システムを様々な抽象レベル(ブロック図、不完全なステート図、プログラミングモデル、等々)で論理検討してきた。設計生産性における必要な進化が達成されるためには、近い将来この状況は変えなければならない。ハードウェアとソフトウェアを含むシステムの仕様、検証、実装を単純化し、効率的な設計空間探索を可能にするためには、今まで親しんできたレジスタトランスファーレベルよりも上位の新しい抽象レベルが必要になる。

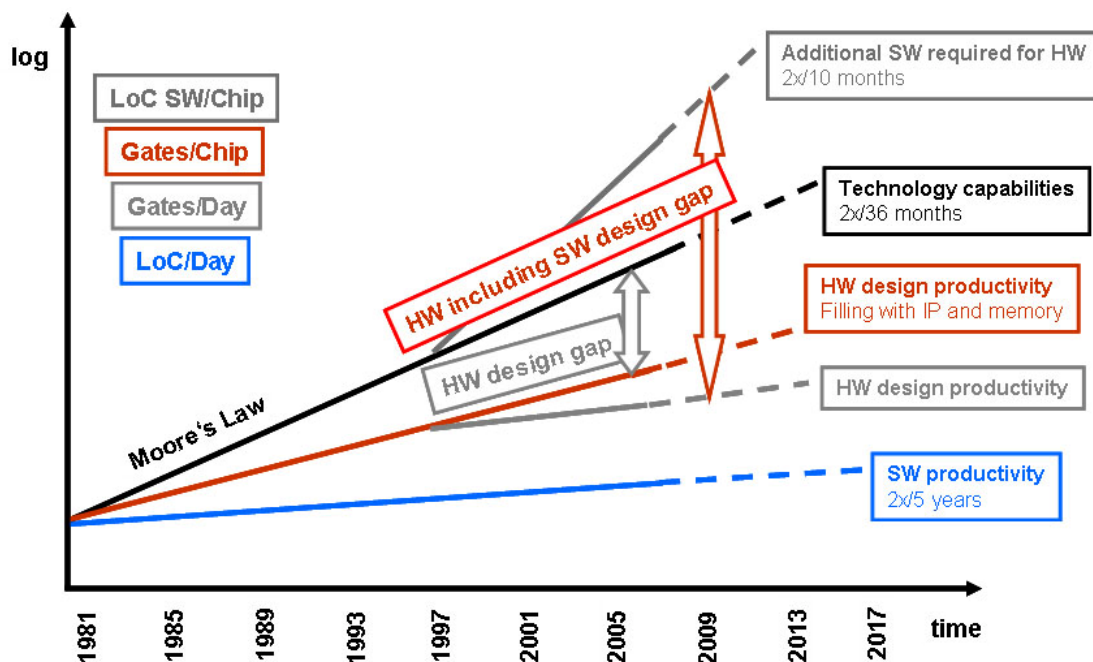


図 DESN 3 Hardware and Software Design Gaps Versus Time

⁵ この図は、10ヶ月毎に2倍になるソフトウェアへの要求と、ハードウェアソフトウェア設計の生産性と同じく36ヶ月毎に2倍になる技術能力を示している。この数年間のハードウェアの設計生産性は、シリコンをマルチコア・コンポーネントとメモリで満たし、新たな機能をソフトウェアのみによって実現することで改善されてきたが、一方で、特にハードウェア依存ソフトウェア(hardware-dependent software)の設計生産性の改善は大きく遅れ、5年毎に2倍になっているにすぎない。赤の矢印がハードウェアとソフトウェアを含む新たな設計ギャップを示している。この版では、初めてこれらの新たなソフトウェアの要求を例示する図を追加した。本資料はITRS2007版で追加された。

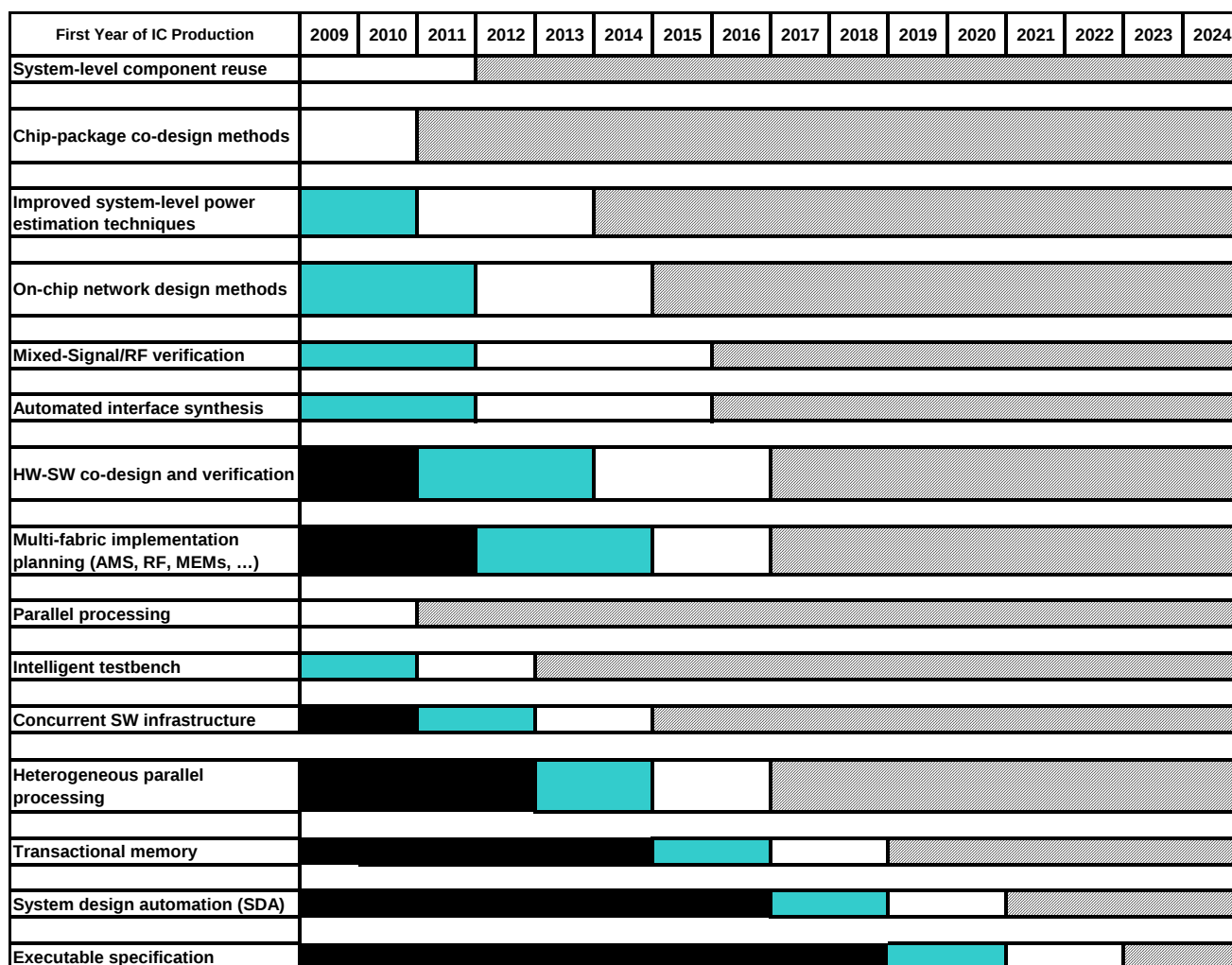
システムレベル設計⁶においては、メソッド面がツール面より急速に困難になってきている。シングル・チップ上に膨大な複雑度を持つシステムが実現可能であるが、これを確実にコスト効率よく開発するには、今日の設計生産性に比べて約 50 倍以上の改善が要求される。シリコン複雑度は、ヘテロジニアスな信頼性の低いデバイスと接続コンポーネントを使って、信頼性のある利用可能なシステムを開発しなければならないことを意味する。グローバルな同期回路設計は、プロセスばらつきと電力消費により非常に高価なものとなり、チップを横断する信号は、もはや 1 クロックサイクルでは到達しない。したがって、システムレベル設計は、機能と接続がパイプライン動作をするかのように、ネットワークと分散処理を取り込まなければならない。例えば、通信構造を最初に設計して、次に機能ブロックを設計し、そしてそれらを通信バックボーンに統合する。組み込みシステムのソフトウェアの量と、マルチコアSOCアーキテクチャの急速な普及によって、システム複雑度は劇的に増大する。図 DESN3 に示すように全体の設計工数をソフトウェアが占めるだけでなく、ハードウェアと密接な組み込みソフトウェアも多くを占めており、実現すべき機能は、今日ハードウェア中心となっているSOC統合検証工程で処理されなければならない。

ITRS ロードマップの 2009 年版では、システムレベル設計の技術課題は、ほとんど前の版と変わっていない。この事実は、これらの技術課題がとてつもなく複雑であるということを表している。たとえば、動作合成はシステムレベル設計にとって必須であるが、10 年以上もの研究課題であり、近年の C 言語や SystemC ベースの合成や、トランザクションレベルモデリング (TLM) 技術の進歩にもかかわらず、実用的な動作合成は現在も実現されていない。また、現在の SOC 設計工程がほとんどベストエフォートベースであり、今後、「自動修復」、「自動構成」、「誤り訂正」等の新たな方向性が予測される。設計技術は、“More Moore” と “More than Moore” の領域で、最終的にコスト効率のよいシステム実装と、社会や市場の要求に基づくアプリケーション要求を結びつける役割を果たすであろう。表 DESN2 には、低レベル設計に対するシステムレベル設計の消費電力削減における相対的な影響度についての新たな技術要求を含めて、将来の技術世代におけるシステムレベル設計に対する定量的な要求を示している。

表 DESN 2 *System-Level Design Requirements*

図 DESN4 は、対応する時間軸におけるシステム・レベル・デザインの解決策を示す。表 DESN3 は、技術課題と解決策の対応を説明している。

⁶ システムレベルでは、シリコンのリソースは抽象ベルの機能とブロックで定義され、設計対象には、高位レベルとアセンブリ言語で書かれた埋め込みコードやコンフィギュレーション・データなどのソフトウェアとコアや、ハード設計回路やバス、リコンギギャブル・セルなどのハードウェアが含まれる。ハードウェアは実装された回路要素で、ソフトウェアはハードウェアで実行される機能の論理的なインスタクションである。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。動作を統合したものがシステム・ファンクションを定義し、アーキテクチャ・ブロックを統合したものがシステムプラットフォームを



This legend indicates the time during which research, development, and qualification/pre-production should be taking place for the solution.

Research Required
Development Underway
Qualification / Pre-Production
Continuous Improvement



図 DESN 4 System-Level Design Potential Solutions

定義する。システム機能のシステム・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とアーキテクチャ及び機能における異種混交によって、困難度が増している。

表 DESN 3 Correspondence Between System-Level Design Requirements and Solutions

<i>Requirement</i>	<i>Solution</i>	<i>Explanation of the Correspondence</i>
<i>Design block reuse</i>	System-level component reuse	The larger and more complex the components that can be reused, the greater the expected overall design reuse
	On-chip network design methods	Standardized communication structures and interfaces support reuse: IPs with standardized interfaces can be easily integrated and exchanged, and communication structures reused
<i>Available platforms</i>	Multi-fabric implementation planning (AMS, RF, MEMS, ...)	Enables integration of different fabrics on same die or in same package (SiP); hence, enables reduced number of platforms
<i>Platforms supported</i>	Automated interface synthesis	Automated interface synthesis is one building block to an integrated synthesis flow for whole platforms.
	Automated HW-SW co-design and verification	Required for integrated, platform-based system development
<i>Accuracy of high level estimates</i>	Improved system-level power estimation techniques	System-level power estimation needs to match progress in high-level area and performance estimation
	Chip-package co-design methods	Packaging effects, e.g., on timing, must be accounted for in higher-level estimations
<i>SOC reconfigurability</i>	On-chip network design methods	To provide flexible, reconfigurable communication structures
<i>Analog automation</i>	Multi-fabric implementation planning (AMS, RF, MEMS, ...)	Multi-fabric implementation planning for AMS and RF components are a building block to analog automation
<i>Modeling methodology, description languages, simulation environments</i>	Mixed-Signal/RF verification	As in digital design, verification is an increasingly critical and time-consuming activity in the design flow
<i>HW offers multi-core systems that have to be exploited by SW</i>	Parallel Processing	Due to thermal and power limitations further performance increases have to be realized with multi-core systems.
<i>Reduce SW verification effort</i>	Intelligent Testbench	SW simulation, formal verification and automated testbenches for SW will reduce the verification effort for embedded software and enhance quality.
<i>Productivity increase required for SW since SW cost >> 50% of total system cost</i>	Concurrent Software Infrastructure	A set of tools that allow concurrent software development and debug
<i>Increase SW execution performance</i>	Heterogeneous Parallel Processing	Parallel processing using different application-specific processors for each of the separate functions in the system
<i>SW Productivity increase required</i>	Transactional Memory	A concurrency control mechanism analogous to database transactions for controlling access to shared memory in concurrent computing. It functions as an alternative to lock-based synchronization
<i>Productivity increase required for HW/SW co-design</i>	System Design Automation (SDA)	True system-level design including electronic hardware and software, mechanical, bio, opto, chemical and fluids domains
<i>Reduce verification effort</i>	Executable Specification	Specifications written in a formal language allow automated verification starting early in the design process and at high abstraction levels, without the need to code multiple verification models. This enables a design flow from specification to completed system that has no manual processes and that can be completely validated at each step.

図 DESN5 は、必要とするシステム消費電力の最小化を実現するためのシステムレベル設計の増大する役割に対するロードマップを示している。この図のパーセントで示す数値は、将来のテクノロジノードにおいて、システム設計の各段階が負担すべき消費電力削減割合をで示している。

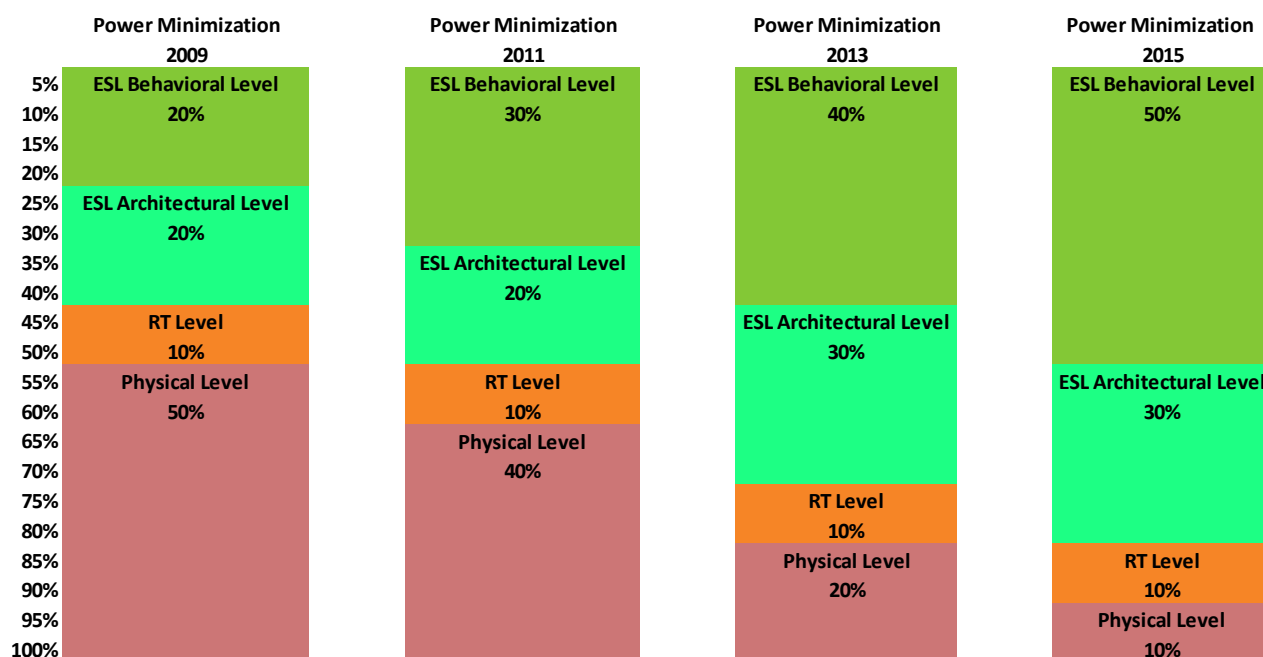


図 DESN 5 Evolving Role of Design Phases in Overall System Power Minimization

論理、回路、物理設計

IC のインプリメンテーションに対する伝統的な理解では、論理設計とは、システムレベル設計のハンド・オフ（現在、レジスタ・トランスファ・レベル）から物理設計への入力に適したゲート・レベル表現へのマッピング・プロセスである。回路設計では、プロセスおよび製造上の制約に基づき、仕様に規定された電氣的・物理的な特性を達成するデバイスと配線トポロジ（スタンダードセル、フル・カスタム、アナログ、など）を創造する。物理設計では、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、抽出、性能解析）を実施し、デバイスと配線を空間的に正しく埋め込む。物理設計のアウトプットは製造（現在、generalized data stream(GDSII) file が中心）へのハンドオフ（テープアウト）であり、正確さ（設計規則、レイアウトと回路の等価性、など）及び制約（タイミング、消費電力、信頼度など）の検証を含む。論理、回路、物理設計は、システムレベル設計を支える DT においてインプリメンテーション層を構成する。

設計生産性は、信頼できて見積もり可能なインプリメンテーション・ファブリックへのシステムレベル・サインオフを要求する。しかし、シリコン複雑度は、最終的な設計品質（タイミング、消費電力、シグナル・インテグリティ、信頼度、製造容易性など）に対する物理的な影響を評価し、抽象化することを困難にする。不正確な見積もりによる過剰なガードバンドの設定を回避するために、論理設計、また将来的にはシステムレベル設計は、物理設計とより緊密にリンクさせる必要がある。このため、最新パラダイムである階層、トップダウン、レイアウトに基づくインプリメンテーション・プランニングは密結合したインクリメンタルなスタティック解析（消費電力、タイミング、ノイズ）をバックプレーンとして存続するであろう。将来のインプリメンテーション DT は、爆発的に増加する複雑度及び、新規懸案事項（欠陥許容、ばらつき、コストなど）に直面するため、その成功はプロセス/デバイスの抽象化、制約操作、分析、及び最適化を調整するメソッドの選択に依存する。

現在のハードウェア設計自動化は、より微細化が進む製造プロセスから与えられるチャレンジと機会に対処するため、より発展させる必要がある。現在のメソッドは、”separation of concerns”（大きな問題を小さな部分問題に分解することができれば解決が容易になるという考え方）を原理としているため、複雑な設計フローを緩やかに結合した管理可能なステップの連続で構成しているが、急速に時代遅れになりつつある。本シナリオにおいては、設計フローにおける上流ステップでの判断が、下流ステップを拘束する制約となる。このような設計

ステップの逐次化は、すべての設計側面を同時に考慮するメソッドに比べて、最適化が劣るデザインを生じる可能性がある。しかし、全設計パラメータの同時最適化の実践は困難であるため、本状況は回避不可能であり、下流に与える制約が達成可能である限りは許容されるであろう。しかし、これらの制約が達成不可能になると、本メソッドは破綻する。この場合の典型的なアクションは、上流設計段階に戻り、問題のある判断を判定して修正する手戻りである。このような手戻りは、レイアウト合成がタイミング制約の達成、すなわちタイミング・クロージャ達成に対する能力不足により、論理設計と物理設計の間で必要になっている。（製造容易性検証（製造頑強性検証）に対するクロージャ達成が、高価な手戻りの新原因として登場している。）理想的には、今日のメソッドにおける時間を浪費する論理合成と物理合成間（あるいは、物理合成と製造容易性検証間）の手戻りは、回路の論理構造とレイアウトの同時最適化により両ステージを融合させれば解決できる可能性がある。

論理、回路、物理設計における設計要求

論理、回路、物理設計における技術要求は、対象テクノロジーからの影響に対する回避策（コントロール要求）、または、与えられた期間とテクノロジーで設計するための要求生産性（オブジェクティブ要求）のいずれかに該当する。各要求に対する定量的な目標を表 DESN4 に示す。

表 DESN 4 Logical/Circuit/Physical Design Technology Requirements

各技術要求に対する説明を以下に示す。

asynchronous global signaling (非同期グローバル・シグナリング) — （オブジェクティブ要求）現在の IC 設計における重要なチャレンジは、基本クロック信号を、許容可能なスキュー以下でチップ全体に分配することである。設計済みモジュールをチップ上で結合させ、独立したモジュール間に信頼できる通信を確保することも、チャレンジである。技術要求の「asynchronous global signaling」とは、同期を送信要求信号と受信了承信号で実現し、大域クロック信号を用いない設計スタイルを意味している。長距離大域信号に対してはリピータ挿入手法を用いているが、リピータ挿入手法における消費電力、配線混雑及び、面積コストの増加につれ、本要求値は増大する。GALS(globally asynchronous locally synchronous)設計では、粗粒度の機能モジュールを従来の設計技術で開発したうえで、ローカル・クロック・ジェネレータと自己タイミング化ラッパを組み込み、各モジュールが非同期のハンドシェイク・プロトコルを使って通信することを可能とする。ハンドシェイク・コンポーネントの個数が、システムの複雑度を決定する。クロック分配と異なり、関連する同期化オーバーヘッドのため、独立コンポーネント数が劇的に増加すると予測するのは妥当ではない。2012 年までには、非同期クロッキングが発展して、市販ツールが利用可能になるであろう。新たな調停機構（ツリー、Fulcrum Microsystems 社のクロスバー、など）が、2014 年にかけて進化し、回路内の遅延を改善するであろう。

Parameter uncertainty(パラメータの不確実性) — （コントロール要求）EDA ツールは、論理設計と物理設計の両ドメイン間のリンクを強めて、配線遅延の不確実性に対処している。この対処方法は配線のみが対象であるため、トランジスタにおける電氣的な不確実性に対しては、影響は大であるが対処できていない。本要求値は、プロセスばらつき及び、それに起因する論理設計・回路設計におけるパラメトリックな不確実性に対する、高精度な対応の必要性を示している。表中の「% - effect」は、縮小するデバイスにおけるパラメトリックなばらつきの関数であり、予測値を示している。

Simultaneous analysis objectives (同時に分析すべき設計目標) — （オブジェクティブ要求）現在のテクノロジーでは、多数の設計目標と物理現象が互いに影響しているが、それらを同時に分析し、最適化する必要がある。ここに、設計目標とは、信号遅延、消費電力（アクティブ、すなわち、ダイナミック）、面積、シグナル・インテグリティ、デルタ遅延、デルタパワー、プリンタビリティ、そして、リライアビリティである。現在開発中の統計的手法により、2008 年までに、歩留まりの最適化が現在の設計フローに組み込まれる。そして、2013 年までに、統

計的手法は設計フローの本流となり、開発期間と歩留まりを最大で 25%改善するであろう。デルタ遅延とデルタパワーの同時分析に、統計的手法が導入されると思われる。演算の信頼性に対する最適化としては、既存のロバスト演算処理(例:University of Michigan の Razor technique)が拡張され、2009 年頃には設計フローに組み込まれるであろう。設計生産性及び設計コストは、2013 年までに評価関数に組み込まれるであろう。設計生産性指標に対する定義は漠然としているため、成熟した最適化手法は知られていない。このため、2013 年の要求は red になる。

Number of circuit families in a single design (一つのデザイン中に混用する回路ファミリ数) — (オブジェクティブ要求) 高性能集積回路では、速度を追求するために、面積、消費電力、設計工数などを犠牲にして、回路ファミリを混用することが多い。回路ファミリが混用される典型的な順序は、スタティック CMOS、マルチ V_t 、マルチ V_{dd} 、そして、ダイナミック CMOS である。非同期 FF (リテンション FF) のような回路形式も、量産チップに進出するであろう。回路ファミリの組み合わせとして、将来の組み合わせの主流を正確に予測するのは困難である。しかし、一つのデザインの中で複数の回路ファミリを扱う能力は、重要な要求であり続ける。同時に、論理ファミリの混用及び設計制約の低次元性により設計複雑度や製造困難度が上昇するため、一つのチップ上にのせる回路ファミリ数が比較的少ない値に抑えられると思われる。

Analog content synthesized (合成されたアナログ・コンテンツ) — (オブジェクティブ要求) デザイン中の合成で作成したアナログ回路の量を、総アナログ回路量に対する割合で予測して表 DESN4 に示した。詳細は、システムドライバ章の AMS 節を参照のこと。アナログ合成が経る軌跡は、20 年前の論理合成の軌跡を思い出させる: 1) 今日、アナログ合成は設計者と置き換わる寸前まで来ている。2) 2013 年までには、アナログ合成の重要性は、1990 年のマイクロプロセッサにおける論理合成の重要性に同等となり、典型的なアナログ回路の 25% に適用されているであろう。3) 2020 年までには、現在のアナログ合成技術は飽和し始めるため、デジタル合成でデータパスのインプリメンテーションなどに期待を寄せているように、新たな技術を用いた自動化が求められる。

Adaptive/self-repairing circuits (適応型/自己修復回路) — (オブジェクティブ要求) 現在、大多数の回路は何らかのダメージを受けると正しく機能しなくなる。製造プロセスにおける変動も、製造チップの信頼性や歩留まりを劇的に減少させる。多くの場合は、システム規模と複雑度のトレードオフにより、壊れたら置き換えるのが安価だという考えで、電子部品をコンパクトだが頑強でない手段でインプリメンテーションするとの判断が続くであろう。しかし、より多くの回路において、システムへのアクセスが極めて高価であるか、または、遠隔的に故障状態を分析して修理するためにシステムと通信するのが極めて高価である。このため、従来ほどは面積が大きくないが、内部故障に完璧に適用できる回路に対する需要が高まる。このようなデザインは、設計検証、製造、およびテストのコストをコントロールするのに役立つ。自己を変更および修復する回路は既に製造されており(例:IBM eFuse 技術では故障部分の電圧及び周波数を調整する)、新しいテクノロジーではさらに普及するであろう。増加し続ける運転コスト及び保守コストに直面するため、自律又は自己制御コンピュータシステムは、不可欠になるであろう

Full-chip leakage power (リーク・パワーのチップ総量) — (コントロール要求) 消費電力は半導体産業が現在直面している主要な技術的問題である。リークパワー(サブスレッショルド、ゲート)は、プロセスの微細技術への移行に伴い指数関数的に増加する。ダイナミック V_t 、クロック・ゲーティング、パワー・ドメイン/電圧アイランド、動的電圧制御、動的周波数スケールリング、複数 V_t トランジスタ、ボディ・バイアス法などの技術により、2012 年までは、リークを緩和するであろう。現在のテクノロジーではサブスレッショルド・リークに対してゲート・リークが支配的であるが、high-k 誘電率の導入によりゲート・リークは制御可能となるためサブスレッショルド・リークが性能の制約となると思われる。リークに対する要求値は 2007 年の値で正規化しており、インバータのリークをテクノロジノードに伴いスケールリングして算出した。

'Native' 3D design technology (ネイティブ 3D 設計技術) — (オブジェクティブ要求) 22nm を過ぎると、消費電力、性能及びコストのボトルネックが著しく拡大するため、産業界は、IC の厚み減少と積層による集積化(3D

IC)に着目するようになる。この 3D IC では、シリコン貫通ビア (through-silicon-via、TSV) が、使用され始めている。Interconnect 章及び Assembly-and-Packaging 章のロードマップが示すように、TSV 技術は、複数ダイ(チップ)上のデジタルブロック及びアナログ/RF ブロックを細粒度レベルで集積するのを可能とする。この集積では、配線、遅延及び消費電力を抑える付随効果がある。しかし、TSV 及び 3D ダイ積層を扱う物理設計ツールが存在しないため、この技術が広く受け入れられるのを遅らせている。現時点で存在する 3D IC 向け物理設計ツールの多くは、「擬似 3D」である。この「擬似 3D」は、2D IC 用の既存ツールの改造に過ぎない。しかし、3D IC 設計からの要求は、より強力なネイティブ 3D 最適化を必要とさせるため、3D モジュール用フロアプラナや、増加する積層(ダイ)間のゲート及び TSV を 3D 配置するツールが求められる。3D 専用の解析ツール(熱機械ストレス、信頼性、など)も必要となる。重要な 3D 設計技術チャレンジには、1) TSV マネジメント(デザイン分割、物理位置、信頼性への影響)、2) 熱機械分析及び最適化、及び 3) 信頼性、電源ノイズ制約及び高度なパワー制御方式に配慮したチップ上のパワー供給が含まれる。

論理、回路、物理設計における解決策候補

論理、回路、物理設計における解決策候補を図 DESN5 に示す。説明コメントは以下の通りである。

1. **Automated handshake logic/circuit tools (ハンドシェーク論理/回路の自動設計ツール)** — チップ内の非同期かつ遅延を抑えた大域通信に必要
2. **Synthesis and timing accounting for variability (ばらつきを考慮した合成およびタイミング)** — テクノロジにおけるパラメトリックばらつきに配慮した合成。過剰に悲観的なクロック周波数を回避する。
3. **Circuit/layout enhancement accounting for variability (ばらつきを考慮した回路/レイアウトの改善)** — パラメトリックばらつきを考慮した消費電力、干渉、欠陥耐性の最適化。
4. **Macro block/chip leakage analysis (ばらつきを考慮した回路/レイアウトの改善)** — 外部環境の変動を考慮したフル・チップの高精度リーク見積もり。
5. **Power management analysis & logic insertion SOI SOC tools (消費電力制御の分析および論理挿入のための SOI/SOC 用ツール)** — 論理レベルでの消費電力制御のための技術(例: パワー・ゲーティング、クロック・ゲーティング)。特に SOI 技術をサポート。
6. **Analog synthesis (circuit/layout) (アナログ合成(回路/レイアウト))** — ASIC 用アナログ物理合成は、構造化合成フローに適したモデル化メソッドロジに向上することが望まれる。マルチ・ファブリック合成ツールも必要である。
7. **Non-static logic implementation (非スタティック論理によるインプリメンテーション)** — ダイナミック回路ファミリ、スタティックダイナミック論理を作成するための論理及び物理合成。
8. **Cost-driven implementation flow (コスト主導のインプリメンテーションフロー)** — 物理領域のコスト(パラメトリック歩留まり、単位原価、リーク・パワー、総パワー)を、設計空間での分析と探索により強くリンクさせる。異なる設計ファブリックから選択するための分析ツール。異なるテクノロジノードを用いた場合のコストを推定するツール。(レイアウト開始前に実施する)製造容易性分析。
9. **3D system design space exploration tools. (3D システム設計空間探索ツール)** — あるアプリケーションが与えられたとき、3D 実装と 2D 実装のどちらを選択すべきかを(コスト、性能、消費電力などの観点で)判定したり、3D 実装における適切な粒度(コア、ブロック、ゲート)を決定するためのツールである。

10. **Native 3D power/thermal analyses, optimizations. (3D ネイティブな消費電力/熱解析、最適化)** — 冷却及び電力供給に対する物理設計は、モジュールの信頼性に重大な影響を与える。性能と消費電力における最適化において、TSV に関連した製造容易性及びコスト問題に取り組むために、分析と最適化が必要とされる。

First Year of IC Production	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Automated handshake logic/circuit tools																
Synthesis and timing accounting for variability																
Circuit/layout enhancement accounting for variability																
Macro/chip leakage analysis																
Power management analysis and logic insertion SOI SoC tools																
Analog synthesis (circuit/layout)																
Non-static logic implementation																
Cost-driven implementation flow																
3D system design space exploration tools																
Native 3D power/thermal analyses, optimizations																

This legend indicates the time during which research, development, and qualification/pre-production should be taking place for the solution.

Research Required
Development Underway
Qualification / Pre-Production
Continuous Improvement



DESIGN 6 Logical/Circuit/Physical Design Potential Solutions

論理、回路、物理設計における要求と解決策候補の関係を表 DESN5 に示す。各要求に対して、定量的ターゲットを実現するようにツール又は手法を構成している。

表 DESN 5 Correspondence between Logical/Circuit/Physical Requirements and Solutions

Requirement	Solution	Explanation of the Correspondence
Asynchronous global signaling % of a design (SoC)	Automated handshake logic/circuit tools	Departure from fully synchronous design paradigm needed for power reduction, latency insensitivity, variation-tolerance
Parameter uncertainty %-effect (on signoff delay)	Synthesis and timing analysis accounting for variability	Tools that account for process uncertainty, and resulting parametric uncertainty, will reduce guardbanding and increase chip yields
Simultaneous analysis objectives	Circuit/layout enhancement accounting for variability	Optimizations which consider parametric uncertainty

<i>Simultaneous analysis objectives</i>	Power management analysis and logic insertion SOI SOC tools	Requires budgeting of area/power/timing constraints
<i>Simultaneous analysis objectives</i>	Cost-driven implementation flow	Cost is an engineering parameter that affects turnaround times. Silicon cost no longer dominant; test and manufacturing costs increase emphasis on adaptive, self-repairing circuits
<i>Circuit families # of circuit families in a single design</i>	Non-static logic implementation	Non-static implementations help improving different chip parameters
<i>Synthesized analog content</i>	Analog synthesis (circuit/layout)	Allows for larger portions of a chip to be analog
<i>Full-chip leakage</i>	Macro block/chip leakage analysis	Enables accurate leakage predictions
<i>% Native 3D design tech in flow</i>	3D design space exploration tools	Enables correct choice of 2D vs. 3D, and 3D granularity
<i>% Native 3D design tech in flow</i>	Native 3D power/thermal analyses, optimizations	Required for co-optimization of reliability and manufacturability along with performance and power

設計検証

機能検証のゴールは、システム実装が完全にその仕様であるデバイスに計画された所望の動作と、一致することを保証することである。残念ながら、半導体設計の複雑度の増大により、機能検証は未だに解決できない課題であり、多くの検証エンジニアや学界の研究活動で費やしている膨大な取り組みでも打破できずにいる。現状の産業界の取り組みは、厳しい設計期間とコストの制約の下で、不完全な検証プロセスが実施されているに過ぎず、設計の部分的な側面を検証の対象として確認し検証しているだけである。これらの検証対象は、設計機能動作の全体から見ればほんの一部分でしかない。この対応不可能な複雑さの原因は、分散処理を実現するために、高度に接続された通信プロトコルで組み合わせられた、高度に統合されたシステム・オン・チップ (SoC) と並列チップ・マルチプロセッサ・システムに対して、検証技術が追従していないことにある。不良チップによる膨大なコストのために、産業界の開発工数の多くの部分を機能の正しさを実現することに費やしている。現在の開発プロジェクトでは、検証エンジニアの数は設計エンジニア数を越えて、最も複雑な設計では、その比率は 2:1 にも達するとの複数の報告がある。このような工数割り当てが行われている理由は、確実な解決策が見あたらず、多くの部分が経験に基づいたアドホックな検証手法のためである。

飛躍的なブレークスルーなしには、設計検証は、半導体産業の将来の成長に対して、複雑度に追従できず注目される障壁となるであろう。アドホックな検証手法から、構造化されたフォーマルなプロセスに移行することで、ブレークスルーが起こる期待がある。システム設計の機能を検証するために、今日使われている検証手法の主流は、繰り返しモデルを作っては、限られたテストベクタでそれらをシミュレーションして、見つかったバグをつぶすというものである。この目的には、非常に高率でシミュレーションベクタを生成する論理シミュレーション技術が主流である。しかしながら、これらのテストによるカバレッジは通常非常に低く、エラーが発見されたときには、問題の原因を特定するためのデバッグ作業は、非常に長くて複雑なデバッグトレースによるマニュアル解析となってしまう。さらに、これらの手法は、品質に影響が大きい部分や低いカバレッジ部分など特定の設計の検証作業となってしまうため、多くの検証工数を必要とする。代替手法への移行には大きなコストがかかるので、従来の検証技術が固定化している。このため、使えるようになってきた新しい代替手段としてのフォーマルやセミフォーマル技術の採用は、なかなか進展しない。

構造化した検証手法では、システムの所望の動作のゴールデンモデルを作って、そのモデルとシミュレーション結果をと比較するというように工数を使うようにしている。カバレッジの計測は、検証対象の設計の正しさはある信頼度で裏付けるものとして集計される。機能検証では、最初に通常一人の設計者が設計する個々のコンポーネントに取り組み、次にチップ、最終的にシステム全体に取り組むように階層的に進めることが一般的である。そうすることにより、ある一つのコンポーネント(構成単位)に含まれているバグに対して、早期にかつ容易に取り組み、解決することができる。(システム全体のシミュレーションは複雑なため、特に追加費用を容易に吸収可能な大きな市場を持つ設計においては、ハードウェア・エミュレーションがしばしばシミュレーションの代わりに使われる。ハードウェア・エミュレーションは、シミュレーションスピードの点では桁違いの性能改善

を手に入れることができ、早期にシステム統合とソフトウェア開発を可能とする。しかしながら、エミュレーションを用いても、わずかに定数倍の改善しか得ることができない。)この手法の最も手間のかかるところは、コンポーネント間の通信インタフェースの検証で、しばしば手遅れで見逃したバグを、複雑で未検証であったコンポーネント間の通信で見つけられることがよくある。ようやく、フォーマルやセミフォーマル検証技術への関心が高まってきている。今日、IC 開発チームは、一握りの商用ツールによるセミフォーマル手法で主流の検証を補完するような、新しい手法を探している。これは急速に発展している領域で、新しい創造的な方法で、多くの場合フォーマル検証技術をシミュレーション・ベース手法と組み合わせる、いくつかの新しい解決策が毎年、研究環境や産業開発の両方で登場している。

ポストシリコン検証(バリデーション)は、テープアウトから顧客出荷までのシリコンデバイスの検証に重点的に取り組むものである。これは、過去10年間に急速に成長した検証領域で、主として、半導体ベース・システムの膨大な複雑さによるものであり、言い換えると、システム・インテグレーション後に明らかとなる多くの潜在的な不具合があることによるものである。ポストシリコン検証のゴールは、それぞれの製造部分で現れる論理や設計バグを検出するところから、通常、コンポーネントの一部にのみ現れる電氣的なプロセス起因の問題や、まれに発生するランダムな製造上の欠陥を見つけ取り除くところまでを含む。この処理には、製造したコンポーネントで完全なプロトタイプシステムを作ることや、組み込みソフトウェアから高レベルのアプリケーションにまで及ぶソフトウェアのテストプログラムを実行することを含む。ポストシリコン検証には2つの重要な特徴がある。1) 高速実行速度により、プリシリコン・シミュレーションに比べて桁違いのサイクル数を評価することができる。例えば、シミュレーションやエミュレーションでは、実行に数ヶ月かかるバリデーション一式を数分で終わることができる。2) 製造したシリコン部品(チップ)の内部の観測性が限られる。実際、内部信号は、プリシリコン・シミュレーション環境で観測できたようには観測できないので、テスト・バリデーションやデバッグはさらに困難な作業になる。これらの検査作業は、通常、ロジックアナライザやオシロスコープのような機器を使って、デバイスの外部信号やレジスタ・インタフェースをアクセスして行われる。

設計検証への要求

表 *DESN6* は、将来のテクノロジードにおける設計複雑度への対応に必要な、設計検証への技術要求項目を示している。技術的評価指標として、設計の正しさを保証する能力と、開発プロジェクトで費やされる検証工数の両方について見積もっている。表中の数値は、派生設計におけるものではなく、新規開発の設計プロジェクトに対するものである。もし検証効率が設計複雑度に比例するならば、検証用ツールのフォーマル検証とシミュレーションの分類は、やがて両者を組み合わせたハイブリッド手法とセミフォーマル検証手法の2つの分類に変わるであろう。表 *DESN6* では、ハイブリッド手法における、フォーマル検証手法に対するシミュレーション・ベース手法の割合を見積もっている。ソフトウェアやミックスドシグナル・コンポーネントによる影響の増大と微細化の進展による電氣的な振る舞いが重要になってくることから、異種のコンポーネント間およびシステム全体としての検証が、全体検証工数に占める割合は増加して行く。最後に、検証要求に対してより構造的な手法に移行することで、設計仕様をフォーマルに表現する努力が、すべての検証工程を自動化する方向に向かわせることになる。

検証環境整備は、次の3つの部分に分けられる。1) 検証する設計対象に対して新規に開発する部分、2) サードパーティから入手する部分(検証 IP として知られている)、3) 社内で以前に設計されたものから流用する部分。表 *DESN6* では、最初の2つの部分を再利用(Reuse)の項目に示しており、3つ目は100%から他の2行の数値を引くことで計算できる。表の最後の2行は、機能カバレッジ(Functional Coverage)の重要性が増すことを予測したものである。従来のカバレッジは、コードカバレッジとステートカバレッジが対象であったが、機能カバレッジは検証すべき設計回路の機能をより直接的に反映している。しかし、それが特定分野に依存するため更なる設計工数を必要とする。短期的および長期的に検証の進展を定量化するためには、機能カバレッジの採用が増えることが必要である。エンジニアリングチームが、機能カバレッジの検証環境構築に習熟して、

その再利用が増えてきても、将来にわたって機能カバレッジの利用が広がる必要があるということを、この表は示している。

表 DESN 6 *Design Verification Requirements*

設計検証の課題

検証に対する最も重要な課題の多くは、ほとんどすべてシステムドライバに関連している。短期的に最も重要な問題は、フォーマルおよびセミフォーマル検証技術が、もっと信頼性が上がり、扱いやすくなることである。特に、検証品質に対する意味のある測定基準と同様に、フォーマル検証ツールの処理容量と信頼性に大きな進展が必要である。長期的には、抽象レベルを上げることと、フォーマル検証の扱える範囲を広げることが、重点的な課題である。これらの長期的な課題は、実は現在も重要であるが、短期的な課題ほど危機的状況にはなっていない。一般に、すべての検証課題は SoC にあてはまる。MPU は、その最先端の複雑さと、飛び抜けて大量に生産される途方もなく複雑な設計の独自の経済性のために、独特な検証課題がある。結果として、長期と短期の両方で、異なる分野ごとの検証課題と解決策が存在する。

処理性能 (キャパシティ) — 今日、検証技術はフォーマル検証とシミュレーション・ベース手法の、2つに大別することができる。両者には致命的な欠陥がある。フォーマル・ツールは中小規模の回路しか扱うことができない。シミュレーション・ベースのツールは、ほとんど制限なく複雑な回路をシミュレーションできるが、非常に長時間のシミュレーションを実行しても、ほんのわずかなカバレッジしか得られない。エミュレーションとラピッド・プロトタイピングはソフトウェア・シミュレーションに比べて桁違いの速度が出るので、高いカバレッジを得ることができる。しかしながら、その改善度は定数倍であり、設計複雑度の増大に比例するものではない。ポストシリコンのバリデーションは、シリコン・ハードウェアとアプリケーション・ソフトウェアからなるシステムの、全体としての機能の正当性に関するものであり、そのために、すべてのコンポーネントが一体として統合され、設計の最上位レベルの複雑さで動作するものである。この領域の課題は、新しい解決策を作り出すことで、おそらく、フォーマル検証とシミュレーション・ベースの両手法のよい面を統合することで、すべての設計階層において高いカバレッジを得ることができる。

頑強性(ロバスト性) — 現在の検証技術で重要な点はその信頼性である。一方では、シミュレーション・ベクタ当たりの実行時間は、設計の複雑度に比例するので、シミュレーション・ベース手法はかなり予測性がよいと言える。したがって、我々が最も複雑なデザイン・レベルに向かうにつれて、シミュレーションの性能は直線的に減少する。エミュレーション技術も同じような傾向を持つが、シリコン・プロトタイプは常に信頼できる。もう一方で、フォーマル検証手法は、検証問題の複雑さに対応するために、ほとんど予測できない経験則に大きく依存している。どのような設計回路と検証アルゴリズムの組み合わせであっても、検証が完了するかどうか判断するのに、専門家ですえ四苦八苦することがある。問題の大きさを測る共通の指標としてのトランジスタやゲート、ラッチの数は、フォーマル検証の複雑度とはほとんど相関がない。容易に検証できる数千のラッチを含む回路が容易に見つけられるのと同じように、既存の検証手法では検証できない 100 未満のラッチを含む回路を容易に見つけることができる。このような予見不能性は、設計現場では受け入れられない。困難な検証課題は、検証工程をもっと確実なものにすることである。これが検証容易化設計(easy-to-verify design)手法となるには、検証アルゴリズムの経験則を改善するか、与えられた回路の検証困難度の特性解析の改良が必要である。

検証の評価指標(メトリックス) — 短期的に重要な検証課題として、検証の質を計る必要性があげられる。特に、カバレッジに対する考察が必要である。カバレッジの評価指標には様々なタイプがあり、それぞれに長所と限界がある。コードカバレッジは、シミュレーションの間に設計ソースコードのどれだけがシミュレーションされたかを測定したものである(例えばコードの行数や数式のカバレッジ)。その主な限界は、動的シミュレーシ

オン実行を静的なソースコードに対応付けていることにある。つまりシミュレーション実行時には、様々な実行可能なパスがあるために、存在する多くの潜在的な問題を見過ごすことにある。状態カバレッジや遷移パス・カバレッジなどの構造カバレッジは、設計の有限要素マシンに着目しており、どの状態がシミュレーションで確認されたかを測定する。しかし一般的に設計全体を 1 つの有限要素マシンとして扱うのは不可能であり、カバレッジの際にいくつかの有限要素マシンが対象となるだけである。このカバレッジは 1 つのマシンに対しては良い結果を与えてくれるが、複数のマシンの組み合わせにより発生する状態の組み合わせが考慮されないことが欠点である。機能カバレッジは設計の各機能を対象にしている。機能は設計毎に異なるため、検証チームによって設計仕様に基づいた明確な定義がされる必要がある。そのため結果の質はカバレッジの定義の質にも依存する。現状では、統一された環境下で機能カバレッジのための共通の評価指標がなく、それが求められているので、設計である一定レベルのカバレッジを得ることは、その設計で使われた特定のカバレッジ以上に、価値がある。加えて、機能カバレッジの測定単位を決める汎用的な方式がないので、設計手法をサポートし、設計時のガイドラインとして使える抽象的な基本モデルが求められている。結局のところ、検証用のカバレッジと統一された測定指標のために、テストに欠陥モデルがあるように、機能バグのモデルが求められている。

ソフトウェア — 複雑な SoC の検証は、ハードウェア・コンポーネントやハードウェアとソフトウェア間のインタフェース、システム上で動作するアプリケーション・ソフトウェアの検証が必要である。SoC のソフトウェア・コンポーネントは 1)アプリケーション・ソフトウェア、2)アプリケーションの実行を制御する OS のようにハードウェアから独立したレイヤ、および、3)ドライバのような低レイヤのハードウェア依存のソフトウェアに分類される。このようなシステムではソフトウェア・レイヤは機能として多くが提供されるため、SoC 検証の主要な課題は、ソフトウェアおよびハードウェア・ソフトウェア間のインタフェースをいかに検証するかにある。現在、ソフトウェア開発はデザインレビュー、解析ツール、テスト方法に関して、ハードウェア設計ほど厳格でない。ソフトウェアは本質的に検証が難しく、より複雑で動的なデータと巨大な状態空間を有している。今日最も一般的に使われているソフトウェア検証手法は「オンチップ検証」で、ハードウェア・コンポーネントの製品版でソフトウェアを走らせている。この手法はソフトウェアの本質的な複雑さから必要とされるもので、非常に高速にシミュレーションを行うことができるが、設計段階の最後にならないとソフトウェア検証が開始できない欠点がある。ソフトウェア検証向けの古典的なフォーマルな手法は、巨大な状態空間を持つシステムで、ソフトウェアアプリケーションをかなり大幅に抽象化したモデルを要求する SoC に広く適用するには、大変な手間を必要とする。ハードウェア・ソフトウェア間のインタフェース検証は、2つの領域を同時に検証する必要があるため、それ自身が課題である。この作業を扱いやすいものにするには、インタフェース動作の適切な抽象化をする方法や、抽象化されたドライバレイヤでの正しさを確認する方法、抽象化されていないレベルでのドライバの正確な動作をアサーションチェックするツールが必要となる。短期的な課題は、基本的に低レベルのソフトウェア検証を可能とする技術を開発することである。長期的な課題は、ソフトウェアに適用する検証容易化設計を理解することと、ソフトウェアとハードウェア・ソフトウェア間のインタフェースの確実な検証手法を開発することである。

再利用 — あらかじめ設計された IP ブロックは、非常に短時間でこれまで以上の複雑な SoC を組み上げることを可能にする。これに対応する検証手法の大きな課題は、既設計でかつ検証済みのブロックから組み上げたシステムの迅速な検証を可能にすることである。重要な問題は、IP ブロックの抽象的な動作を、いかに厳密に、かつ完全に記述するか、IP ブロックが仮定している環境的な制約をいかに記述するか、また検証を単純化するためにいかに階層を活用するかにある。いくつかの IP コンポーネントは、標準プロトコルの検証 IP から、汎用的な IP ブロックの抽象化されたモデル、ブロックの周辺環境からの制約をチェックするプロトコルチェッカ、トランザクション・ジェネレータまで様々な関連する検証 IP と一緒にして販売され始めている。しかしこれらはまだ初期的な試みであり、再利用設計手法を発展させるには、いかなる IP ブロックにも検証 IP が準備され、一貫して利用できることが必要である。IP ブロックに関連した検証コンポーネントに加えて、特定のプロトコル向け環境生成プログラムのような、独立した検証 IP も必要である。短期的な進展は、おそらくオンチップパスのような標準化された接続 IP に対するものからになるが、任意 IP ブロックのインタフェースに対する一般的な問題が、最終的に解決されなければならない。

特殊な検証手法 — 設計検証での最大の課題は、今日開発中の設計に対応するには、現在知られているアルゴリズムによる解法では、能力の限界を超えていることである。短期的にこの課題を克服する唯一の可能性のある方法は、適切な検証メソッドロジを用いることにある。これに向けた現状の動向には、今日使われるようになっている、シミュレーション・ベース検証とセミフォーマル検証の両方に対するカバレッジベース検証や、実装時に検証されるべきフォーマルプロパティ・セットが容易に得られるようになるフォーマル記述による仕様ドキュメント、そしてカバレッジモデルのテンプレート、があげられる。十分に確実で完全なメソッドロジを得るにはまだまだ多くの課題がある。抽象化の際に設計の重要な特性を見落とすことなく、設計部品やインタフェースなどに対して整合性の取れた抽象化を行う技術を得るための方法が求められている。フォーマル仕様記述が開発されているが、重大な限界はこの記述の完全性にあり、同一システム内の異なる IP 部品が、完全に独立した設計チームによって開発されている現状では、避けられない課題になっている。新しい検証手法は少しずつ受け入れられているが、設計内に基本的なフォーマルプロパティを入れることでさえも課題であり、そのためには設計のある特定の特性を全体として理解しておく必要があり、開発チームにさらに工数を必要とする。最後に、検証メソッドロジは、適切に対処法を用意したとしても、完全さを保障するものではないが、開発リスクの軽減を目標にして進展している。

ソフトウェアとトランジスタ故障 — ソフトエラーと永久的なトランジスタ故障の存在によるシステムの信頼性に関する情報を提供できる検証技術の必要性がますます重要となっている。業界の専門家は、過度のトランジスタスケールリングによるシリコン基板の信頼性劣化を繰り返し警告している。この領域では、初期の解決策として、記録素子の SEU 故障に対する耐性を強固にする設計が既に利用可能である。さらに、非常に小さな追加的な面積コストで、(製造時、あるいは使用時において)永久的なトランジスタ故障に対する保護策が幾つかの研究レベルの解決策として提案されている。これらの解決策は、製造部品の高い欠陥率による歩留まり低下の防止にも展開可能である。

検証容易化設計 — 検証をより効率的にする検証容易化設計の分野に特化した活動が既に見受けられる。例えば、ソフトウェアやハードウェアのデバッグの仕組みをシリコンへ組み込むことが可能である。この方面では、自己検査機能付きプロセッサに関する取り組みが始まっており、小さなウォッチドッグプロセッサ、もしくは分配したハードチェッカのネットワークによりメインプロセッサが正しく実行しているかを検証している。ミックスドシグナル設計では、設計回路のアナログ部をバイパスするループバックモードの挿入により、完全なデジタル回路としてシステムを検証可能にしている。マルチスレッド処理のシステムでは、シンクロナイザが検証用チェックポイントを設定して探索可能な状態空間を減らすことにより、タスクが独立して同期ポイントを越えて進まないようにしている。MPU 設計では、シンクロナイザにより投機実行システムの検証の複雑さを軽減している。この分野での課題は、いくつかの重要な領域での検証容易化設計の技術進展とその導入にある。積み上げ式にシステムを逐次詳細化して行く設計手法の開発が行われているが、どのように詳細化行程が自動化できるのかは明らかでなく、一方で、人手の介在は設計エラーの発生源となる可能性がある。長期的には、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。

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

新しい種類の同時処理 — MPU 設計はより複雑になってきており、新しい種類の同時処理が重要になってきている。既に検証の目を逃れた多くのバグは、キャッシュのコヒーレンシとその他の同時処理による問題に関

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

高位抽象レベル — 設計が RTL より高位な抽象レベルへ移行するに従い、検証手法も追従しなければならない。技術課題は、高位抽象度での検証方式を採用し開発すること、高位設計により可能となったシステムの複雑度に対処すること、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）を考慮せずになされると、ずっと難しいものとなる。

デジタル以外の影響の検証 — 現在のところ、設計検証は主としてデジタルシステムの個々の動作に焦点を当てている。シリコン複雑度とシステム複雑化の両方への挑戦は、将来の検証技術に広範な種類の解析を要求する。シリコン上の集積回路システムの複雑さがVLSI システムの明解なデジタル抽象化を次第に不確かなものにしていく。アナログの電氣的効果は性能に影響を与え、最終的には機能にまで影響するであろう。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）はあまりにも遅く、またデバイスの縮小により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恐れがある。このため、マイクロプロセッサのアーキテクチャ・シミュレーションにおいてある予備的研究が行われており、高位アーキテクチャ・シミュレータはタイミングや電圧などの情報を得るために状況に応じた低レベルシミュレータと情報交換を行うことで、性能への影響を最小限に抑えながらシステム全体評価にフィードバックを返している。長期的には、複合型システム⁷として扱うデジタルとアナログの境界における、これらの問題を検証するためのフォーマル技術が必要となる。同様に、設計の最上位レベルでは、システムの複雑さにより、将来の検証タスクには（ネットワークプロセッサにおいてサービス品質を保証するように）アナログ的、確率的動作を表現し検証することが求められる。このように複合型システムと確率的検証が課題となろう。

異種混在システム — シリコンウェーハ上にデジタル設計回路に並べて配置される新しい技術の開発が、まったく新しい課題を提起する。MEMS や電子光学デバイス、電子バイオデバイスなどがその例である。システムのデジタル部の検証を可能とするために、これらの新しい構成部品はデジタル部分と非デジタル部分のインタフェースと非デジタル動作を、適切に抽象化したモデルが必要となる。

アナログ／ミックスドシグナル — 今日のアナログシステムは、連続システムに対する古典的な解析ツールや、周波数領域に対するシステムモデリングと解析によって、ほとんど検証されている。検証の大半は、設計後にアナログ実験装置を用いてテストチップを検証することで行われる。ミックスドシグナル設計は、検証作業をデジタル部とアナログ部に分けて行なう。将来は、ミックスドシグナル・システムがシリコン開発全体の中での比重が増すため、ミックスドシグナルの適切な検証メソッド開発が、この領域において重要となる。ここでの課題はシステムのアナログ部の検証をデジタル部の検証と結合させることにある。目標達成のための要求項目の 1 つは、デジタルシミュレーションとアナログシミュレーションの性能ギャップを埋めることである。

⁷ 複合型システムは複雑な連続振る舞い（例えば、微分方程式モデル）と複雑な離散的振る舞い（例えば、有限状態マシン）の両方の振る舞いをする。この分野は古典的コントロール理論と離散的フォーマルベリフィケーションの両方から技術を借用している。

設計検証の解決策

図 DESN7 はこれまで述べた検討課題への重要な解決策を、その利用可能時期とともにまとめている。

First Year of IC Production	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Verification methodology centered on verification IPs and reuse																
Hierarchical hardware verification methodology																
Solutions for the integrated verification of hardware and embedded software																
Design development and structure, taking into account for verifiability [2]																
Simulation and verification solutions for the detection and correction of soft failures and																
Reusable methodologies for functional coverage development																
Concurrent verification of hardware and software components during development																
Design specification formalized for verifiability [4]																
Verification of analog and mixed signal designs																

This legend indicates the time during which research, development, and qualification/pre-production should be taking place for the solution.

Research Required
Development Underway
Qualification / Pre-Production
Continuous Improvement



図 DESN 7 Design Verification Potential Solutions

図 DESN7 の注釈

[1] これは高位 HW/SW 統合システムの検証技術を必要とする。現在の設計技術はシステムの機能を、ハードウェアで実装されるべき部分とソフトウェアで実装されるべき部分への分割プロセスの自動化に着手している。通常、その結果はシステムコンポーネントの高位記述であり、それらを組み込みソフトウェアとハードウェアのどちらで実現すべきかを示す。しかしながら、高位記述は複雑化しているので、この開発段階でのシステム検証(バリデーション)と、特に高位レベルでの通信プロトコルの正しさの検証ができる解決法が必要である。

[2] 言い換えると、検証に最適なハードウェア開発の技術

[3] 言い換えると、ハードウェア・レイヤと組み込みソフトウェア・レイヤとのインタラクションの正しさを検証する解決法。現状の利用できる解決法は2つのレイヤを別々に検証する：1) ハードウェアレイヤは古典的な有効性検証技術で大半を検証される、2) 組み込みソフトウェアはハードウェア(通常は開発中)のシミュレーションモデル上でソフトウェアを走らせることにより検証される。組み込みソフトウェアアプリケーションの急激な複雑度増加により、特別な解決法を必要とするますます差し迫った問題となっている。

[4] 仕様自身の検証を容易にするような設計仕様を開発のための(おそらくフォーマル)技術

検証 IP は、現在既に入手可能であり、検証工程の生産性向上のために進化しているメソドロジのトレンドの一部となっている。将来、この手法は SOC 設計にサードパーティの IP が積極的に採用されるに伴い、もっと広く普及するであろう。上記で述べたように、ハードウェアソフトウェアの階層化と統合化メソドロジを用いた解決策は、既に研究されており、今後 2,3 年のうちに入手可能となるであろう。表に示したその他の解決策は、現在、まだ概念段階にある。前の節で述べたように、検証容易設計の開発、ソフトウェアや永久的なトランジスタの欠陥を訂正するフォーマルな手法は、MPU 領域で初期の実験段階にある。構造的機能カバレッジと仕様に関する技術が本流の解決策となるのはまだ先であるが、これらは設計と検証の生産性のギャップを埋めるために重要な技術である。表 DESN7 は設計検証への要求と解決策の対応関係を示している。

表 DESN 7 Correspondence Between Design Verification Requirements and Solutions

Requirement	Solution	Explanation of the Correspondence
Productivity of verification tasks	Verification methodology centered on verification IPs and reuse	Verification IPs and reuse reduce the amount of new verification development required in a project
	Hierarchical hardware verification	Structured methodologies improve design team productivity
	Reusable methodologies for functional coverage development	Functional coverage is time-consuming, and specific for each distinct design; development of reusability techniques is critical to boosting productivity
	Concurrent verification of hardware and software components during development	Advancing the verification of hardware in parallel with that of software components can significantly shorten time-to-market of a product, in contrast to methodologies that begin software verification only after the first hardware prototype
Formal and semi-formal verification centered methodology	Hierarchical hardware verification methodology	Enables the decomposition of the system into smaller blocks which are suitable for formal verification
	Design development and structure taking into account verifiability	Design for verifiability organizes a design so as to simplify verification; additional verification-specific hardware structures further simplify design-time verification tasks
Methodologies for system-level verification	Verification methodology centered on verification IPs and reuse	Verification IP components enable an early start on system-level verification
	Integrated verification of hardware and embedded software and their interface	Directly provides solutions for effective system-level verification
Portion of design specification formalized for verifiability	Design specification formalized for verifiability	Formal languages and methodologies to support the formal specification of a design
Escape rate after tapeout	Design structure taking into account verifiability	Development of hardware structures (checker-like) which can be used to detect and correct a system entering an escaped erroneous configuration after customer shipment
System integration bug rate	Analog and mixed-signal verification	Limits the bug rate due to analog effects
	Simulation and verification solutions for the detection and correction of soft failures and manufacturing faults	Manufacturing faults occurring in post-silicon are detected at system-level integration; techniques to detect and correct electrical and transient defects reduce the effort required to expose and correct these problems.
	Hierarchical verification methodology	Supports management of complexity through decomposition
Functional coverage	Reusable methodologies for functional coverage development	Reusable functional coverage solutions leverage the coverage development effort and boost quality of results

テスト設計

ナノメータ・プロセス技術、ますます高くなるクロック周波数、アナログ・デジタル・RF 混在の回路、そして高密度に集積化された SoC、3D ダイ・パッケージ、SiP は困難な課題を DFT (Design for Test) に提起している。テスト産業は、コンポーネント・ベース設計のための高位テスト合成から、非常に高性能なアナログや RF システムのためのノイズ、干渉、消費電力にいたる多様な問題に対応して行かねばならない。それらの多くのテスト問題は、設計の初期の段階から最適なテストバリエーションが考慮され、DFT が組み込まれた場合に限って解決される。さらに、初期解析結果が変わり、後で複数のテスト検証が必要となり、更に設計とテストの一体化が必要となることもある。論理－回路－物理設計、そして設計検証における成果やその結果(設計フロー、ベクトル、故障モデル、感度)はテスト設計においても再利用される。メモリ、DSP、PE、SoC、アナログミックスドシグナル/RF、そして MPU といった様々な製品セグメントの種類をこえて高度なテストの課題、つまりオンチップ DFT、BIST、テストバリエーションの拡張、ならびにチップ開発の計画段階での、製造段階でのテスト装置の事前計画の前倒しを要請している。

DFT への技術要求とその解決策候補は、対象としているサブシステムの特徴によって大きく変化する。それにもかかわらず、SoC も SiP も安い製造コストでかつなるべく性能を落とさずにテスト可能になるようにシステムレベルで統合されなければならない。この節では、システムドライバを共通のフレームワークとして用い、DFT への技術要求と解決策を記載する。

表 DESN8 は詳細設計技術に基づく4つの主要なシステムドライバであるアナログ/ミックスドシグナル/RF ドライバ、MPU/PE/DSP ドライバ、メモリドライバ、そして汎用 SoC/SiP ドライバへの DFT への技術要求を要約している。「微小遅延故障」や「ブリッジ故障」のような新たなデジタル回路の構造テストが超高品質/高信頼性が要求される複雑なデジタルデバイスでは重要視されつつある。

表 DESN 8 *Design for Test Technology Requirements*

アナログ／ミックスドシグナル／RF システムドライバ — アナログ、ミックスドシグナル、RF サブシステムは、コンポーネントの I/O スピードとともに、コア部のクロック周波数やトランジスタやアーキテクチャの性能などのシステム性能として重要になって来ている。今日、業界は 5GHz を超える RF の計測に関して深刻な問題に直面している。コンタクトや治具による信号劣化のためにウェーハレベルで 5GHz を超えるテストを正確に行うのは非常に難しい。パッケージ後のテストでも高周波数はテスト課題となっており、10-300GHz の高周波数のミックスドシグナル回路を低価格の生産テスト装置でフルにテスト及びキャラクタライズできるような手法を是非とも開発せねばならない。同時に、新しい I/O プロトコルを導入して、数 GHz レンジにまで拡張する必要がある。これらの I/O スキームは高速であるのみならず、また非常に複雑である。実際、ソースの同期、変動、同時双方向の動作が、Gbit/s のレートでしかも、Vdd の 1/10 のレベルで電圧が変動する。従来の ATE とコンポーネント・テストツールは GHz 近くのレンジでの共通のクロックベースのテストと I/O 計測で構成されている。したがって、I/O 速度増加とプロトコルは、設計検証と製造段階でのテストの両方で、ATE 装置、インタフェースハードウェア、テストソケットなどの器具、材料、コストに多大の影響を及ぼす。この変化点は、特に I/O に対して、広範囲の業界の技術開発とチップ上のテスト可能性の応用を要求することになる。

これらのアナログ、ミックスドシグナル、RF サブシステムに対する DFT 技術は活発に研究、開発されているテーマであるが成功のためには、この DFT 技術は、設計工数を削減し、ノイズ除去と同じように頑強性を改善するために、可能な限り多くのデジタル回路を使用すべきである。これらの DFT 手法を使った結果は、それが Pass/Fail インディケータであっても、パラメトリックな測定であっても、信頼性と設計者の受け入れを獲得するた

めに、標準的な仕様ベースのテスト手法との相互関連をとっておかねばならない。

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

これらの要求に対する解決策候補は、サブシステムの数だけ多数ある。アナログ、ミックスドシグナル、RF の DFT の問題を解決するために、業界と大学で様々なアプローチが開発されてきた。それらの多くは、例えば PLL、ジッター-BIST、コンバーター-BIST、トランシーバー-DFT などの特定の製品要求にフォーカスするものが多い。DFT の手法は製品種類毎にそのシステム設計とテストが最適化されるように選択するべきなので、アナログ、ミックスドシグナル、RF の DFT 手法の一本化は期待も切望もされていないが、すべての DFT 手法に共通の本質的に重要な機能が、システムレベルの解決策としては絶対必要である。すべてのデジタル DFT 手法は頑丈であり、設計が容易であり、インテグレーションが容易であり、そして CAD ツールの開発やサポートによる開発が容易であることが望ましい解決策になる。現在の DFT 手法では、まだ機能テストとパラメトリック・テストが主流であるが、これらのテストとの良い相関が示されれば、ストラクチャー・ベースの DFT が今後の長期的な解決となる。ストラクチャー・ベース DFT 手法が本質的に有利なポイントは欠陥と故障についてのより深い理解が得られることにあり、これによって品質改善とコスト削減というどのシステムでも最も重要な 2 要件が達成される。

一つの DFT 解決策は、コンピュータと通信をシームレスに統合する現在の設計技術を組み込むかもしれないが、オンチップ DFT サブシステムである制御と通信のための無線部のラッパーを使用することである。オンラインモニタリングが可能になり、外部テスト環境との通信が標準化され、テスト・アクセスの問題は内部的には軽減されるが、一方では、チップ面積、消費電力、ノイズ、回路生成の他の外部負荷などの設計パラメータとの固有のトレードオフが存在する。アナログ、ミックスドシグナル、RF の DFT への複雑な課題を解決するには、創造的な解決策とシステム設計者からテスト・アーキテクトにいたる慣習に捉われない思考が要請される。これらの DFT 解決策は、複雑なシステム・インテグレーションにおける、アナログ、ミックスドシグナル、RF のテストがボトルネックを避けるために、時宜を得たタイミングで開発されなければならない。

MPU/PE/DSP システムドライバ — 近年の GHz のクロック周波数によりデジタル DFT や BIST でカバーされない新たなテスト課題が出てきているとはいえ、これらのシステムドライバは DFT や BIST、また IDD_x のような手法については成熟している。ジッターや位相ノイズに対する高周波数クロックでのテストの課題は、先述のミックスドシグナルないし RF の全体的な問題の一部として考慮し得る。論理設計の観点からは、MPU/PE/DSP/メモリ システムドライバに対する要求は良く理解されており、過去 10 年間に多くの課題が DFT と ATE のコミュニティによって成功裡に解決されてきた。将来のことを考えると、より良いデジタル DFT カバレッジを含む主要な要求は、ビルトイン DFT ないし全体的な DFT 手法の一部でなければならない。このビルトイン DFT は、固有のブロックの機能、設計のスタイル、ハード故障、ソフト故障、パラメトリック故障のような故障検

出のような特定の目標に依存して幅広く変化し得る。カバレッジを著しく改善する必要のある、デジタル・パラメータの二つの例は遅延と消費電力であり、このパラメータのテスト手法を改良するために、DFT と BIST の改善が重要である。故障、欠陥、テストすべきパラメータの広範囲にわたるカバレッジのモニタリングは全体的な DFT や BIST の手法に組み込むことが重要で、特に目標となるパラメトリック減衰や信頼性要求がモニタリングされなければならない。動作環境の中での、オンラインないしはオフラインのカバレッジのモニタリングは、可能性のあるサブシステムの故障を予測したり、分離したりするのに本質的な役割をはたし、それにより故障の修復が適時に実行可能なる。高度に複雑なデジタル・サブシステムのパラメトリック故障による歩留りと製造欠陥を改善するために、フォールトトレラントで修復機能を有する DFT 手法が要求されている。論理システムのための DFT と BIST フレームワークの統合的なゴールは品質でなければならない。

マルチコア MPU デバイスや多数のコアを持つコンシューマ用チップの出現により、テスト時間とコストを削減するために良く似たコアや構造のコンカレント・テストができる DFT が必要になっている。実装時のトレードオフに電力、例えば急激な電流変化を抑えるため複数の位相(スタガード・フェイズ)を使うなども考慮されるであろう。コンカレント・テスト手法にて詳細情報のない単にデータ圧縮の結果としてグローバルな故障結果を示すのではなく、故障解析の結果の正確な位置が示されなければならない。

ここでのドライバに分類される多くの論理システムには成熟した解決策が存在している。残されている課題は、ブロックレベルの解決策よりも、むしろシステムレベルの解決策への要求である。上述で指摘したように、デジタルシステムの性能にかかわる幾つかのクリティカルなパラメータ(遅延、消費電力、ジッター、動作温度など)は、未だに有効な DFT の解決策を求めている。これらのパラメータをキャラクタライゼーション・テストと量産製造テストの両方で決定するために、コスト効率の高い DFT や BIST の手法が非常に望まれる。統合されたキャリブレーションと修復機能を提供するような解決策が、電子システムの設計とテストアーキテクチャの全体に極めて重要なものとして現れるだろう。今日の故障中心の DFT/BIST 手法と、新しいデジタルのパラメータ中心の DFT/BIST の両方を使って、システムテストの設計者は、思い通りに様々な手法を有効に活用することで、システム全体に及ぶ結合した DFT/BIST のアーキテクチャを作り上げなければならない。つぎはぎ的なテストアーキテクチャやいい加減なテスト統合は、テスト時間やテストデータ量に関して、ロジスティックスや実行計画に問題を起しがちで、それは DFT/BIST の有利性を減じることになる。

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

高度に統合された一般的な SoC や SiP 設計 — 既設計ブロックを大規模なデバイス上に統合して行くことで、それらのブロックが単一種類(すべてが論理ブロックとか)であったとしても、線形を超えた複雑度の増大を設計ツール、DFT、そして製造時テストにもたらす。ますます、多様な種類の回路ブロックが統合されつつある。論理ブロック、SRAM そして DRAM は近年では共通して統合されてきたし、今日ではアナログ、ミックスドシグナル、RF、また不揮発フラッシュ回路が論理ブロックや RAN と組み合わせられて来ている。ニューラル・ブロックを用いた人工知能(AI)はニューラルの学習プロセスがあるためテストに更なる課題をもたらしている。統合されたデバイスについての、シリコン複雑度やコストは比較的予測可能である。しかしながら、搭載されたブロックと

デバイス種類が混交であることで、テスト可能性や設計検証、また製造時テストに対して、線形以上で予測不可能なコストの増大をもたらしている。大規模な論理ブロックとして丸々搭載されている ASIC や MPU マクロでは、この問題は既に顕在化しており、実際製造時のテストのコストがシリコンコストを超えている。DFT をとってさえ、そのコストは線形以上になる。この節で述べる DFT/BIST 要求は、先行する三つのシステムドライバでカバーされたブロックレベルの DFT の単なる寄せ集めではないことを、ここで強調しておきたい。システムの範囲に拡大された DFT 要求と、その解決策は、DFT/BIST の全体的なロジスティックスとスケジューリングの解を提供しなければならないし、ブロックレベルではカバーできない、ないしは見過ごされるテスト機能を含んだものでなければならない。

システムのレベルでは、DFT はチップ上のそしてシステム内のテスト生成とテストアプリケーションの手法を提供し、それによってテスト時間と外部 ATE を使ったテストボリュームの負担を軽減しなければならない。テスト生成の専用ハードウェアと組み込みアルゴリズムは、テスト・アクセスの必要性を削減し、DFT 手法の推奨であるシステムのレベルでの BIST への基底環境を提供する。アナログ、ミックスドシグナル、RF サブシステムの DFT 技術の改善にともなって、テスト生成とアプリケーション資源は、システムテストの環境にスムーズに統合されなければならない。DFT によって検出される故障(いくつかの種類のハード故障と多くの種類のパラメトリック故障)は、歩留り改善とタイムツーマーケット時間短縮のために修復または許容範囲内に収められるべきである。したがって DFT とシステム設計メソドロジーは、DFT と協力してこの修復機能を提供しなければならない。さらにグローバルで魅力的な DFT の側面は、本来の場所での計測とフィードバックを通してのキャリブレーション機能にある。プロセスばらつきやデバイスの不完全性に対してして、システム性能を調整するために、設計者によって採用されている多くのキャリブレーション手法は、DFT 手法と本質的に等価なものとして同一視できる。システムキャリブレーションのメソドロジーとして DFT を使用することで、設計者が設計とテストの緊密な連携をとるようになることが促進される。オンラインの測定、フィールドリペア、性能低下の修復などは、包括的な DFT メソドロジーによって対処が可能になる基本的な性能要求である。

組み込みブロックのテストは、組み込まれていないブロックのテストと比べて桁違いに長いテスト時間をかけるべきでない。メモリ、ロジック、アナログ、ミックスドシグナル、RF の各々に対するテスト手法、ATE、製造インテグレーションは、独自の最適化のために根本的に異なるスタイルをとっており、統合してテストを行った場合には、そうしたスタイルが破綻することがある。DFT は統合した場合の全コストが統合しない場合のコストより高くないように、テストコストに関しても考慮すべきである。

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

DFT の最大の貢献は、テスト量とテストコストの低減にある。この DFT の貢献は、SoC/SiP のレベルで実証されるべきである、というのはこのレベルでテスト量とテストコストがはっきりと、そして容易に計算できるからで表の中のテスト量削減の目標値は、システム内の特定のブロックに対するテスト量やテスト時間ではなく、全体システムをカバーするテスト量とテスト時間である。テスト時間やテストコストは、ATE と非常に強く関連している、というのは DFT や BIST があっても、なお ATE がテストでは重要な役割を持っているからである。様々な DFT 手法と ATE との間のインタフェースはきちんと定義しなければならない、それによって固有の製品のテストが要求する異なった DFT と ATE の選択と組み合わせの自由度が許容される。テストアクセスポートやその他のテスト関連の標準である IEEE1149.1(IEEE Standard Test Access Port and Boundary-Scan Architecture), 1149.4, 1450(IEEE P1450.6, Draft Standard for Standard Test Interface Language(STIL) for Digital Test Vector Data—Core Test Language),

1500(IEEE Std 1500-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits.)が整備されているとはいえ、広範はシステムレイヤ(動作物理的なレイヤ、データ通信のレイヤなど)に拡張された包括的な DFT/ATE インタフェースプロトコルが確立されるべきである、それができれば DFT 手法の貢献と ATE の可能性をフルに活用できるようになる。

SoC/SiP の DFT/BIST の全体的な課題に対する解決策は改良され、非常に速いペースで考案され続けている。SoC や SiP 中のテクノロジー、サブシステムの設計手法の多様性は、テスト生成ツールと DFT/BIST ツールがデジタルテスト手法(故障、欠陥、パラメトリックベース)とアナログ、ミックスドシグナル、RF テスト手法(機能、パラメトリックベース、将来的には故障、欠陥ベース)が統合されることを要求している。

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

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

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

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

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

アーキテクチャでの課題 — 回路の歩留りを満足することが困難なためにアーキテクチャでの冗長性が必要とされるであろう。この抽象度ではこれ以上のことを行うのは難しい。

論理と回路での課題 — デジタル、ミックスドシグナルでの適応型回路の必要性は増すであろう。

消費電力、タイミングの収束性を含んだ、統計的な設計が基本となる。ただ、実際に設計を行うには、以下の重要な 2 点を明確にすることが避けられない。

- 1) 統計的設計ツールへの入力となる、キャラクタライズとモデリング
- 2) 統計的解析から、さらに計算複雑度の高い最適化への発展

最終的には、統計的解析ツールと統計的最適化手法は、大雑把で精度的に疑問のあるモデルではなく、実際の製造と設計に起因するばらつきを反映することが必須である。統計的解析と最適化の過程での種々のばらつきの合成は、プロセスのキャラクタライズでの統計的なメトロロジを通じての初期の分解手法と整合が取れていなければならない。合成と分解の不整合があると、不必要な誤差と、疑問となる値に陥り、結果として計算コストが嵩んでしまう。

レイアウトと物理設計での課題 — まず最初に、デザインルールチェックの複雑度が急速に増大する。ルールは 2 層構造(要求ルールと、推奨ルール)へと展開して、さらにはおそらく 3 層構造かあるいは、層のない構造(合否ルールではなく、設計者がテープアウトでの基準として適応できる、歩留りと面積コストのパレートカーブを提供する)へと進むかもしれない。このチェックは、設計者にその複雑さを意識させることなく行われることが必要である。2 番目に、リソグラフィ装置における解像度の限界のために RET(resolution enhancement techniques: 解像度補正技術)の影響をさらに明示的にデザインフローに適応することが要求される。RET ツール、たとえば OPC や、CMP(化学的機械研磨処理)でのフィリングはタイミングや消費電力などの回路的な指標を明示的に意識する必要がある。それにより、全体として、目的に沿ったツールの整合性を保ち、歩留りの向上を可能とし、製造コストの削減と、マスクデータ準備のための時間を改善することに繋がる。このアプローチは必然的に密接な統合フローとなり、回路的な意図を下流へ伝え、個々のツールの勝手な修正による精度劣化を防ぐことになる。結果として、レジスタートランスファーレベルから GDS データ(RTL2GDS)へのフローにおいて、適切な RET や OPC の修正を下流へとつなげるようにプランニングする必要がある。例えば、グローバル、ローカル配線層の密度は、概略配置において、ダミーフィル挿入の場所を決定するために考慮される必要があり、これにより事前の CMP フィリングとその容量の見積りが可能となり、これと同時にクリティカルなネットの情報を最終レイアウトフェーズ、マスクデータ生成(MDP)OPC レベルへ伝達することができる。

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

表 DESN9 は、上記の主要な課題を、DFM 要求として定量化したものである。

DFMとしての要求は、次に掲げる2つのカテゴリに分類される。

経済原則に立脚する要求—このカテゴリはマスクコストを含み、これは数百万ドルのレベルに達し、そのため小さな会社や、新興マーケットの団体から実現されるシステムオンチップの技術革新を危うくすることになる。

ばらつきや、リソグラフィ限界からの要求—このカテゴリは、デバイスや配線と直接に関連している下流レベルの抽象度での限界と、設計される回路全体に関連する上流レベルの抽象度での限界を含んでいる。下流レベルでの定量化された要求値としては、(オンチップ回路に給電される)電源電圧のばらつき、(ドーピング濃度のばらつきと全体トレンドの両者の影響としての)閾値電圧のばらつき、そして CD(Critical Dimension)ばらつきの割合などが含まれる。上流では回路パフォーマンスのばらつきの割合(回路でのクリティカルパスのように、チップ全体のパフォーマンスを決定する回路速度の不確かさの割合)や、回路消費電力のばらつきの割合(アクティブと待機状態での両者の電力を含む、消費電力の不確かさの割合)を要求として持つ。

経済上の主要因であるマスクコストは、以前から検証することが困難であった。マスクコストと TAT の削減のブレークスルーを達成するための研究が必要とされる。そのようなブレークスルーがないため、次善策として、マルチプロジェクトウェーハ、コンフィギュラブルロジックやストラクチャード ASIC が提供される。ばらつきに関して、CD コントロールのような”プロセスレベル”のパラメータや、供給電圧のような”回路レベル”を含んだ様々なパラメータやその目標値は設計で制御される。(素子の領域を反転するチャンネルドーピングの影響を含む)閾値電圧のばらつきのような他のパラメータは不可避に増大し、結果として10年後か、それ以前に危機的な状況となる可能性を秘めている。もし根本的な新しい解決策が発見されなければ、プロセスレベルからデバイスレベル、回路レベルへとパラメータを上流へ伝達することによって、非常に大規模で総合的な回路性能や消費電力の変動に対処する必要がある。

表 DESN9 で予測されるばらつきの増大の結果として、(1)従来からある、製造欠陥による回路の構造の変化による、致命的な不良 と (2)素子や配線のばらつきに関連するパラメトリックな不良との区別が曖昧になっている。例えば、微小な MOSFET の閾値電圧のばらつき増大のため、SARM セルは書き込みテストが失敗することで1つの論理値に固定してしまう現象が発生する。この種の現象は 65nm で既に広く顕在化しているが、さらに顕著になり、他のクリティカルな回路種類であるラッチやレジスタファイルでも起こりえる状況になる。このような不良の原因は下記のように分類される。

- ・ **プロセスばらつき**: チャンネル長、閾値電圧、移動度などのデバイスパラメータの統計的ばらつき。チャンネル長のばらつきはライン端の粗さ(Line Edge Roughness)や他のパターン精度に起因し、閾値電圧や移動度のばらつきは主にドーパントのランダムな不均一性に起因する。
- ・ **寿命のばらつき**: 物理パラメータの変化による回路の動作寿命のばらつき。最も顕著なものは NBTI (negative-bias temperature instability) による V_{th} の変動や、HCI (hot-carrier injection)、TDDB (time-dependent dielectric breakdown) によるゲート電流の変動である。
- ・ **固有ノイズ**: 微細なサイズにおいて顕著となる正常動作に起因するノイズ源、これはショットノイズ、熱ノイズ、テレグラフノイズからなる。

今後のテクノロジーにおけるこれらの現象による影響の予測をするために、米国の設計 TWG は3種類の多用される標準的なCMOSロジックである、SRAMビットセル、ラッチ、インバーターについて研究した。これはアナログミックスドシグナルなどの領域における設計主体の回路の将来傾向予測につながる。この3つの選ばれた回路は、デジタル CMOS 設計での主となる要素を良く代表している。記憶(SRAM ビットセル)、同期(ラッチ)、そして論理関数(インバーター)。現在のハイパフォーマンスのプロセッサでは、数百万の

SRAM ビットや、数百万のラッチそしてインバーターなどが普通に存在する。

この 3 種類の回路の将来的なハイパフォーマンステクノロジーノードでの不良確率は、一般的な論理、SRAM 回路における 12nm までのばらつきモデル予測を含んだ PTM (Predictive Technology Mode) を使うことで、製造プロセスばらつきの影響による振舞いをシミュレーションすることが可能である。

それぞれの不良のカテゴリは：SRAM ビットセルでは read, write, standby の不良であり、ラッチでは D から Q への過剰遅延やホールド不良（クロックサイクル内での）そしてインバーターでは出力でのスイッチング不良である。これらの調査では従来の回路設計の手法（回路トポロジ、P/N 比率、適切なデバイスサイズなど）は変化がないことを前提としている。この手法により、この調査では将来的には回路技術、設計技術も段階的にスケールアップが行われることを予測している。

これらの3種類の代表的な回路の不良率は図 DESN8 に示す。

1. SRAM 不良率は、非常に大量の重複した測定など、既に設計に対して重要な問題を引き起こしているが、これは今後さらに増大する製造ばらつきに対応するために、回路やアーキテクチャでの革新が要求される。
2. ラッチは、SRAM と広く共通した部分があるが、65nm テクノロジーでは SRAM より頑強であるが、22nm 近辺では現在の SRAM の不良率に到達し始め、将来はおそらくその不良率は SRAM を越えるであろう。このため新たな回路とアーキテクチャ技術の両方かどちらかが必要になるであろう。インバーターは CMOS 技術で最も単純でかつ最も頑強な回路であり、今後も頑強であることは正しいが、劣化の傾向を示しており、将来的に問題となる可能性を示している。

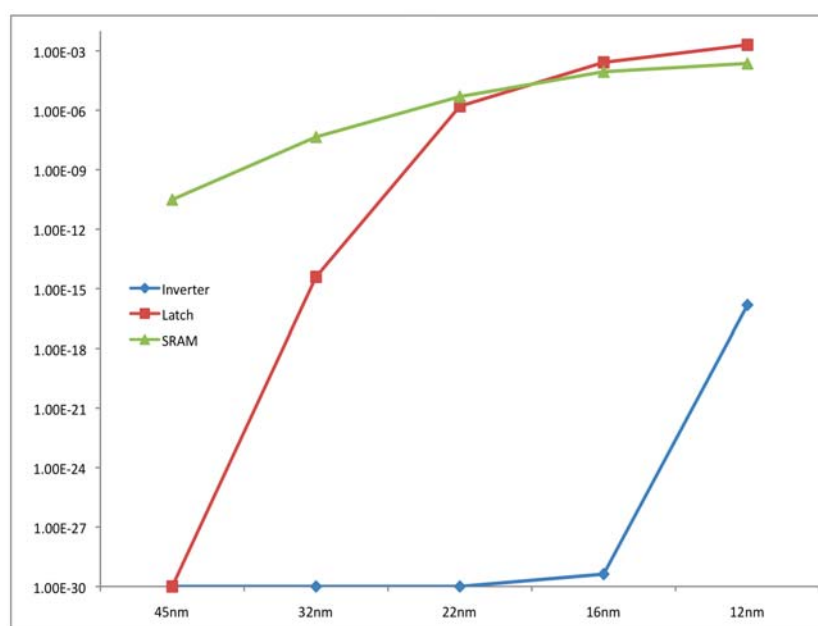


図 DESN 8 Variability-Induced Failure Rates for Three Canonical Circuit Types

図 DESN9 は DFM の解決策候補を示す。

- ・ **基本的な経済的制限に対応した解決策** — マスク費用以外を含んだ経済的要因を明らかにするために、主なアルゴリズムやインタフェースにおいて、DFM ツールが必要になる。チームやマネージャは、難しい DFM のトレードオフの経済的価値について、より直接的に評価できるであろう。
- ・ **ばらつきの影響に対する解決策** — 将来 DFM はばらつきの様々な特徴を取り扱うようにならなければ

ならない。第1に、性能と消費電力の両方のばらつきに対応しなければならない。その結果、統計的機能解析、有効電力、漏れ電力解析ツールが大変重要となる。第2に、システマティックとランダムという2つの特徴ある統計的な歩留り損失を分類する必要がある。第3に、電源供給、温度、閾値電圧を含んだ、ばらつきを引き起こす環境やプロセスの様々な要因の最適化が必要である。最後にツールベースによる解策は、十分ではない。

- 1) 回路動作、電源、クロックや入力信号のばらつき要因を理解し、最適化する高性能な適応型回路
- 2) 局所的に非同期な設計、冗長性、誤り訂正符号化(ECC)を含むばらつきに対する基礎的な耐性に関するアーキテクチャを含む、ばらつきを補償する設計技術が必要となる。

- ・ **リソグラフィ制限の影響に対応する解決策** — リソグラフィが果たす重要な役割のために、ITRS はリソグラフィに関連した DFM 問題や解決策を必要とする分野に取り組んだ。将来の設計フローは、リソグラフィ制約に対応した本質的な取り組みが必要となる。これらの技術は、おそらくルールベース(ツールと設計フローまたはその一方の変更無しに)とモデルベース(直接的にツールと設計フローまたはその一方を変更する)の両方のレイアウト修正を含むことになる。第1に、今日レイアウト後に適用されている RET は、論理合成、タイミング検証、配置配線といった従来の設計工程とますます相互にやり取りすることが必要となる。そしてさらに明確に性能と消費電力の相関関係の測定基準を盛り込むことが必要となる。このやり取りは、レイアウト検証や論理合成といった”モデルベース”型の設計工程において、直接的、或いは間接的に行われる。第2に、従来の設計基準や設計は、設計の段階で基本的に製造可能となるように、益々”製造との調和(manufacturing friendly)”が必要となる。製造と調和した設計基準(効果的な製造性の原則に従う”信頼性”のある厳格な規準)、完全に制限された設計基準(斜め配線を使用しない格子状のレイアウトのような、小さい領域か性能コストで製造性を確実にする簡単な設計基準)、そして自動配置配線ツールと調和したスタンダードセルやコアなど、設計基準は解決策の主要な要素となる。

First Year of IC Production	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Tools that account for mask cost in their algorithms																
RET tools that are aware of circuit metrics (timing, power)																
Radically-restricted rules (grid-like layouts, no diagonals, etc.)																
Statistical analysis and optimization tools and flows (Vdd, T, Vth)																
Statistical leakage analysis and optimization tools																
Architectures resistant to variability (redundancy, ECC)																
Adaptable and redundant circuits																
Post-tapeout RET interacting with synthesis, timing, P&R																
Model-based physical verification																
Model-based physical synthesis																
Manufacturing-friendly design rules, including Radically Restricted Rules																
Litho-friendly (router-friendly) standard cells																
Tools that consider both systematic and random yield loss																

This legend indicates the time during which research, development, and qualification/pre-production should be taking place for the solution.

Research Required

Development Underway

Qualification / Pre-Production

Continuous Improvement



図 DESN 9 Design for Manufacturability Potential Solutions

初期では、タイミング解析のような直接ばらつきを取り扱う新しい解決策が登場する。5 年以内には(2009 年から 2011 年)統計的な方法論が完全に設計に取り込まれる。当面は、成熟した技術の選択的な応用や、高付加価値品や大量生産品で使用される高度な設計技術の一部が適用される。

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

表 DESN10 は DFM の要求とその解決策の間の対応関係を表したものである。

表 DESN 10 Correspondence Between Design for Manufacturability Requirements and Solutions

Requirement	Solution	Explanation of the Correspondence
Mask cost	Tools that account for mask cost in their algorithms	Obvious
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
% V_{dd} variability seen at on-chip circuits	Tools that account for mask cost in their algorithms	Obvious
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
% V_{th} variability (doping variability impact)	Statistical analysis, opt tools and flows (V_{dd} , T , V_{th})	Better estimate of variability impact reduces overdesign
% V_{th} variability Includes all sources	Statistical analysis, opt tools and flows (V_{dd} , T , V_{th})	Better estimate of variability impact reduces overdesign
	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability.
% CD variability	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Leakage power variability will soar. Statistical leakage tools are critical to estimate and control it.
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
Circuit performance variability (gates and wires)		Routing-friendly rules reduce design, mask, and manufacturing complexity
	Router-friendly standard cells	
	Adaptable and redundant circuits	Inherent circuit robustness to variability
Circuit power variability (gates and wires)	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability.
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
		Routing-friendly rules reduce design, mask, and manufacturing complexity
	Router-friendly standard cells	

多様化(More than Moore)の分析

微細化関連の要因と微細化に関わらない要因の相対的な影響度に対する産業界の設計技術に対する要求の傾向を理解するために、本章の 5 つのセクションに記載された各解決策を、‘幾何的なスケーリング’、‘等価なスケーリング’、および‘機能の多様化’の 3 つのカテゴリに分類した。その結果を図 DESN10 に示す。

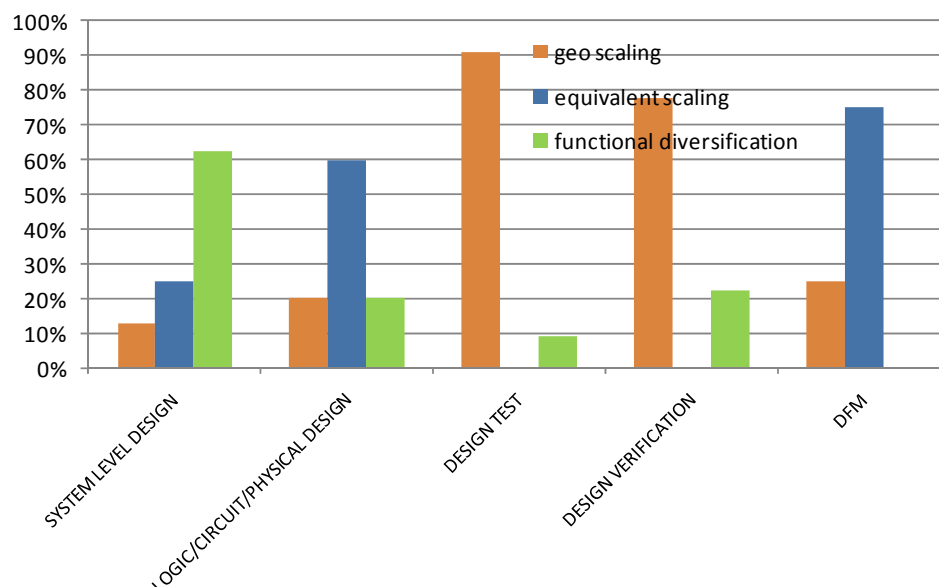


図 DESN 10 Moore and Non-Moore Design Technology Improvements

この図が示すように、設計の早い段階、つまりシステムレベル設計では、機能の多様化が製品要求に対する解決策の必要性に最も大きな影響を与えている。そして、設計が進んで、その抽象度のレベルが低くなるにつれて、等価なスケーリングや特に幾何的なスケーリングが代わって大きな影響を与える。このことは、現在の設計のやり方と整合している。すなわち、1)設計初期のシステムレベルの抽象度の段階では、複数のコア、ファブリティクス、回路方式が混在している。他方、2)物理設計の抽象度の段階では、各ブロックは同じタイプのレイアウトと回路に合成され、はっきりと定義された設計ツールや手法が適用可能になる。

アナログ、ミックスドシグナル、RF 設計技術のトレンドと課題

アナログ、ミックスドシグナル(AMS)および RF 回路は、デジタル回路とは異なる。これらは、"high" (V_{dd} - ノイズ許容値) と "low" (V_{ss} + ノイズ許容値)で決められた 2 状態の電圧レベルで表現される情報ではない。同様に離散的な時間的期間(例えば、クロック信号が high)で表現される情報ではない。アナログと RF 回路で処理される信号は、非常に高い精度で(または、時間的および振幅的に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な利得により、これらのノイズ源を高いレベルで抑圧する能力が備わっている。しかし、この単純な信号回復の原則は、取り扱う信号のダイナミック・レンジが非常に広い場合、アナログと RF の信号処理では適用することができない。動作速度の問題、また、信号回復回路自身が影響を受けやすい信号を防御する以上に、多くの雑音と歪みを作り出すという事実が、これらの問題をアナログ領域ではるかに挑戦的かつ困難な課題としている。

アナログとRFの設計は、素子の理想的でない特性のすべてについての高精度なコンパクトモデリング、キャラクタライズ、特性抽出の必要性を求めてきた。しかしながら、上記の問題はアナログおよびRF回路設計(または、レイアウト設計)において、より高いレベルへの抽象化のためのルール導出を非常に困難にしている。ディジタル回路で使用されるツールは、アナログ設計においては適切なものではなく、非常に不正確であった。歴史的に見て、チップ当たりのアナログおよびRF用トランジスタ数の増加はディジタル用トランジスタ数の増加と比較して非常に緩やかであるため、アナログとRF設計者をシステムレベル設計ツールから遠ざけてきた。回路とシステム設計については、この章の後半の「論理、回路、物理設計」節で述べられる。今日、アナログとRF設計において長く知られていたものと同様の課題が、ディジタル設計(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。より多くの 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。EDA 業界によるアナログと RF に特化した新しい取り組みが特に高次のシステム統合化の為に重要になっている。

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) “ディジタルでさえアナログとなる” ディープ・サブミクロン効果と高い信号・クロック周波数:これは寄生効果の増大、伝送線路効果、シグナルインテグリティ問題を招く; (2)アナログ設計に費やす労力を少なくするためのアナログ合成ツール; (3)歩留り向上を実現する“製造容易化設計”; (4)移動体通信デバイスにおける信号処理システム(そのほとんどはディジタルである)とアナログ RF フロントエンドの高密度な集積化; そして(5)電子部品と MEMS などの非電子部品の高密度な集積化。これらの課題は、後述する設計技術領域についての解説の中でより詳しく述べられる。

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

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

マクロモデル構築には上述の作用のほとんどすべてが影響を与える。信号配線間の容量性および磁気カップリング、パッケージングされたチップの熱特性、アンテナの近距離および遠距離寄生効果、内蔵された MEMS 動作特性などは 2D または 3D の電磁界解析手法を用いて、10 の 4 乗から 8 乗の数の方程式で表現される。計算精度の大幅な低下を招くこと無しに、この方程式は「次数低減法(order reduction methods)」として知られている方法で、次数を 10 の 2 乗から 4 乗まで低減することができる。この領域での解析技術の著しい進歩により、むこう数年間で有効な CAD ツールが実現されると考えられる。このような新しい自動生成されたマクロモデルは、ディジタル回路のタイミング動作やシグナルインテグリティ解析と同様に、アナログ回路設計でも活用されることとなる。

AMS と RF 用システムレベル設計

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

AMS と RF 用 論理、物理、回路設計

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

AMS と RF 用 設計検証

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

AMS と RF 用 テスト設計

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続けるだろう。短期的な要求は、アナログ/ミックスドシグナル用の DFT/BIST であり、特に、ベースバンド用以上に高分解能かつ高周波数への対応である。オンチップ化された 14-16 bit の高分解能 ADC と 1-5GHz の高速 RF 素子のためのテスト技術は、費用対効果に優れているだけでなく、副作用のないものでなければならない。すなわち、それらは内蔵されたアナログ・ブロックの性能を低下させてはならない。高分解能 ADC は通常、直列接続された多段回路もしくは mesh 回路で構成される。このためひとつの可能性ある方向性は、DFT またはセルフテストの戦略を開発するにあたって、この構造的な知見をうまく利用することである。

PIDS 章では、アナログ CMOS トランジスタはより高いアナログ電源電圧で動作し、2-3 の技術ノードにおいてスケールアップされないという特質を述べている。しかし、このことは消費電力、プロセス互換性、面積効率、設計複雑度、検証とテストといったクリティカルなコスト問題を解決しない。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、以下の項目の解決に向けたツールおよび、新しい設計記述言語を含んでいる。

- チップとパッケージを考慮した、回路システムの探索
- 回路合成とサイジング
- 回路図の製造容易化の問題も含めた検証自動化(アナログ的評価)
- アナログ特有な配線ルールを考慮したアナログ/RF レイアウト合成
- アナログ IP の回路情報要約化と再利用

クロスカット TWG の課題

モデリングとシミュレーション

[本稿は Modeling と Simulation ITWG と共有される]ますます縮小して行く特徴サイズに関連して、設計に挑戦する重要な問題の一つは、製造パラメータのバラツキ・変動や、(例えば、チャネルドーピングといった)本質的な原子の性質に起因する設計に関連するパラメータの可変性の増加である。この問題についてはデザイン章の製造容易化設計の部分で特に詳細に議論します。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およ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、(さらに信号遅延やひずみとして表現される) インターコネクットのカップリングといった設計パラメータのその結果生じる分類に変換されなければならない。ますます重要になってくるのは、不純物の原子的な性質である。それは、ある場合は、チャネル領域に平均に現れて、ドーピングと電気デバイスパラメータの莫大な相対的な変動を引き起こす、ほんの 1 つ、または、2、3 の不純物原子に起因することになる。とりわけ重要なものは、プロセス変動を増幅したり、滑らかにしたりするリソグラフィやエッチングといった、異なったものが次々発生するプロセスステップ間の相互作用である。シミュレーションはさらに、寄生成分、遅延変化、ノイズ、およ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。「第二の要求」の取り扱いがアナログ設計において特に重要である(たとえばマッチングは重要な問題である)。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さら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。また、これは、デザインのコーナーモデルの使用によるバラツキに対する過剰補償の発生を避ける事に貢献しなければならない。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。このようにモデリングとシミュレーションは過去のプロセスの合わせこみに対し貢献しなければならない。新しい寛容とそれらの作りこみにたいして現実的な見積りを提供しなければならない。この目標を達成する為には、デバイスやセルレベルの関連情報による設計ツールを用いた顕微的な TCAD シミュレーションへ適切な方法論を開発しなければなりません。(たとえば SPICE パラメータやそれらの統計モデル等)

特にマスク作成に対する短期的なタイムスケールの問題では、例えば、事象をレイアウトからフォトレジストの中に移すのに必要であるアシスト機能、および敷居電圧などの電気データのバラツキの効率的な定義と検討は特に重要である。シミュレーションは、設計者で意図するウェハ城の「理想的な」構

造に十分よく近似し、マスクの修正と適合に貢献するだけでなく、ウェハにデザインを焼き付けるうえでもっともコスト効率の高いマスク構成を選択を助ける等、マスクに対するの高価な過当矯正を避ける助けにならなければならない。この他、長期の挑戦は、特に制御不可能な CD とドーパントの可変性になる。結局、モデリングとシミュレーションは収率を最大にするためにプロセス変動とドーパント変動の影響、および欠陥に対するマスク処理の情報を提供することによって、これらの非理想的な効果の考慮を少なくし設計者がデザインを適合させることにより、収量予測と最適化の設計課題に貢献する。

付録 I: ばらつきのモデリングとロードマップ

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

ばらつきロードマップのフレームワーク(VRF: Variability Roadmap Framework)は、図 DESN11 に示されるように設計 TWG が開発しつつある。このフレームワークでは、以下の 3 つのアブストラクションのレベルが考慮されている:

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

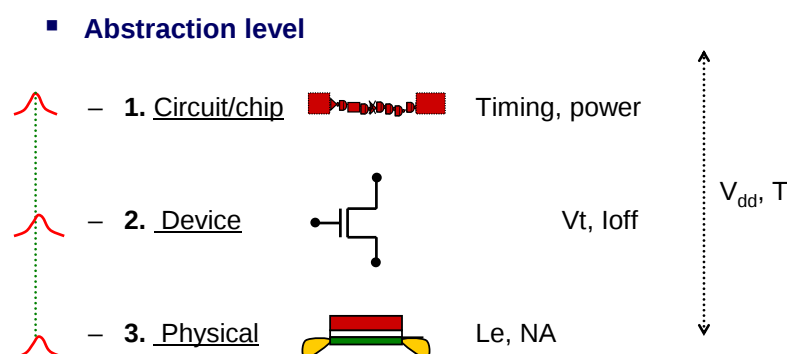


図 DESN 11 Possible Variability Abstraction Levels

このフレームワークでは、モデリングプロセスは以下の式で表わされる。

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

公表されている近似モデルに基づいて、このモデルは単純化されたゲート+配線の回路小片をベースにしている、すなわちパラメータはゲートに関連するものと、配線に関連するものに分解されている。性能は今日のモデルではトータルの遅延としてモデル化されているが、これはゲート遅延と配線遅延に分解される。遅延のばらつきは、モンテカルロ式の入力パラメータのシミュレートされた分布に由来する、個々の遅延値の計算された統計的な分布のデルタとしてモデル化される。モデルの入力として、ゲートチャネル長と幅、酸化膜厚、配線幅、配線長、配線層厚、ILD厚、配線シート抵抗が含まれる。

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

付録 II: DT コストと価値

図 DESN12 は、複雑な集積回路を開発して市場に出すためのコストに対して、どれほど多くの異なるファクタが影響しているかを示している。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。不幸にも、systems-on-a-chip(SOC)の絶えず増加している複雑さは、設計コストとチップ単価をコントロール不能にしている。ますます競争が激しくなる環境と結合されてコストが上昇する中で、収益の確保が困難になってきている。これは、製品開発費が製品のライフサイクルの中で先行投資の状態になり、一方、売上げのかなりの部分が 1 年以上遅れで発生するという事実によって悪化させられる(キャッシュフローの悪化)。次の分析は、継続的な設計技術革新のパイプラインがなければ、設計コスト(したがって、製品開発費も)をかけることが急速に不可能な状態になるであろう、さもないと、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。

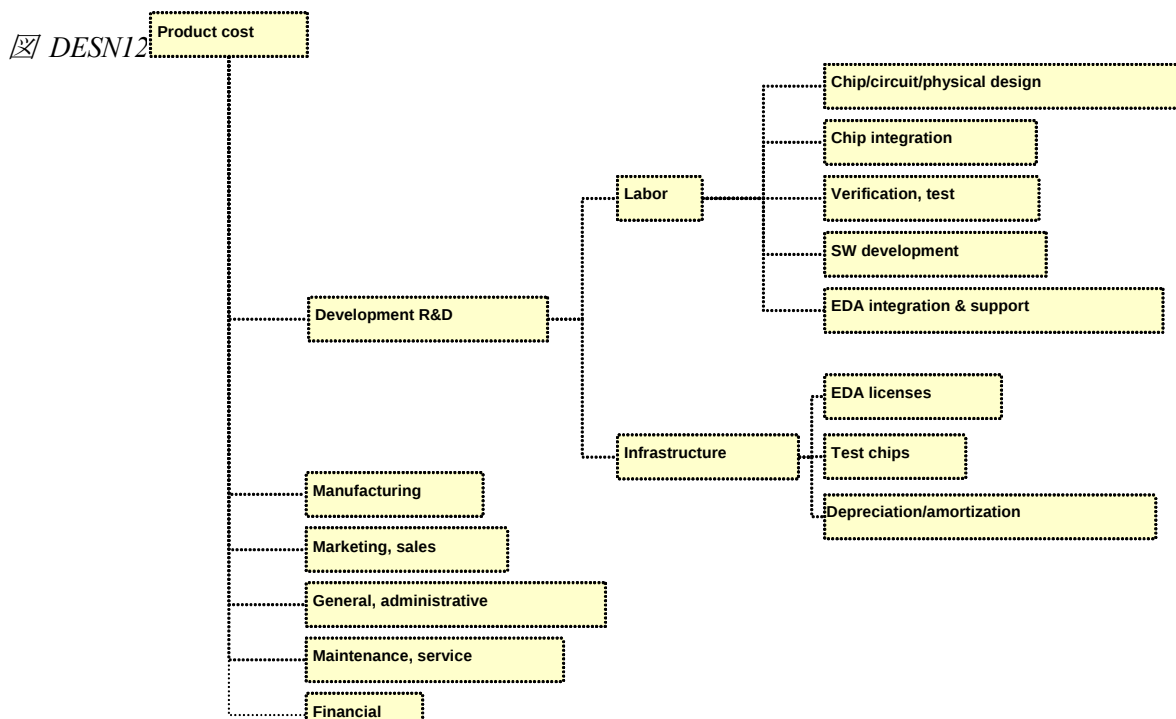


図 DESN 12 Simplified Electronic Product Development Cost Model

図 DESN 12 において、Development R&D 内に示されるほとんどの項目は事実上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、および、レイアウト/物理設計;チップインテグレーション;検証(ベリフィケーション)とテスト;ソフトウェア開発;EDA インテグレーション;ソフトウェアと技術サポート。一方、インフラストラクチャコストは、設計用のソフトウェアライセンス(ソフトウェア開発環境を含む)、テストチップ・インフラストラクチャと減価償却費を含む。これらのコストは、直接費に(一般経費や管理経費を含む)割り当てられた“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)、および設計生産性(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 DESN 11 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。その表から分かるように、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 2009 年までの間に、年率で平均 39.6%増加したことになる。特に、百万ゲートを設計する設計者数(生産性の逆であるが)は、1990 年の 250 人から 2009 年には 1 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。ITRS の設計コストモデルによれば、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。上昇率は 2002 年から 2005 年にかけて年率 2%に下がったが、2007 年と 2008 年に再度年率 5%に回復している。2009 年には世界的な不況により給与の伸びが止まったが、2010 年以降、成長レベルは 5%に立ち直るであろう。中国やインドから報告される IC 設計者の不足により、世界中のハイレベル SOC 設計エンジニアのコストは、設計コストモデルで使われている北アメリカの水準並みに上昇している。。

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)。しかし、その上昇は 2002 年に停止し、そしてここ 8 年間は過去の歴史的な基準を下回っている。今後、EDA ツールの平均売価の上昇率が 3.9%に戻ることが期待されている。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。

トータル設計コスト

この章を一周して元に戻すために、再び図 *DESN1* を振り返りたい。この図では、システムドライバ章で定義された SoC-CP(Consumer Portable)ドライバの設計コストへの設計技術の革新の寄与が定量化されている。SoC-CP の論理ゲート規模は、2009 年で 31.6M ゲートとなり、従って典型的な民生用、ポータブル用 SOC ハードウェア設計コスト（設計者+ツール）は、\$15.7M で、さらにソフトウェア設計コストが\$29.6M かかり、2007 年からソフトウェアの設計コストがハードウェアの設計コストを超えている。1993 年から 2007 年の間に実現した 7 つの主要な設計技術の革新が無ければ、この種の SOC のハードウェア部分だけの設計コストは、2009 年でおおよそ\$1,800M になっていた。このギャップは、Japanese Semiconductor Technology Roadmap Working Group 1 (STRJ-WG1)が作成し、2001 年度 ITRS System Drivers Chapter にも引用された設計生産性の新しい見積りを用いると、より大きな差となる。つまりこの見積りでは、1999 年における新規（リユース）ロジック開発生産性を 360K （720K）ゲート／人・年と見積もった場合、同年のガートナー／データクエストの見積りよりも 6 倍（12 倍）高くなる。

表 DESN 11 Design Technology Improvements and Impact on Designer Productivity

<i>DT Improvement</i>	<i>Year</i>	<i>Productivity Delta</i>	<i>Productivity (Gates/Design- year)</i>	<i>Cost of Component Affected</i>	<i>Description of Improvement</i>
None	1990		4K		
In-house place and route	1993	38.90%	5.55K	PD Integration	Automated block placement and routing, transferred from the semiconductor house to the design team
Engineer	1995	63.60%	9.09K	Chip/circuit/PD Verification	Engineer capable of pursuing 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	38.90%	56K	Chip/circuit/PD Integration Verification	Blocks from 75,000–1M gates
IC implementation suite	2001	63.60%	91K	Chip/circuit/PD Integration EDA Support	Tightly integrated tool set that goes from RTL synthesis to GDSII through IC place and route
RTL functional verification tool suite	2003	37.50%	125K	SW development Verification	Tightly integrated RTL verification tool suite including all simulators and formal tools needed to complete the verification process
Transactional modeling	2005	60%	200K	SW development Verification	Level above RTL, including both HW and SW design and consisting of behavioral (where the system function has not been partitioned) and architectural (where HW and SW are identified and handed off to design teams) levels
Very large block reuse	2007	200%	600K	Chip/circuit/PD Verification	Blocks >1M gates; intellectual-property cores
Homogeneous parallel processing	2009	100% HW	1200K	Chip/circuit/PD Design and Verification	Many identical cores provide specialized processing around a main processor, enabling performance, power efficiency, and high reuse
		100% SW			
Intelligent test bench	2011	37.5%	1650K	Chip/circuit/PD Verification	Analogous to RTL verification tool suite, with automated verification partitioning and taking electronic system-level description as input
Concurrent software compiler	2013	200% SW	1650K	Chip and Electronic System Design and Verification	Enables compilation and SW development in highly parallel processing SoCs
Heterogeneous massive parallel processing	2015	100% HW	3300K	System Electronic Design and Verification	Each of the specialized cores around the main processor is not identical from the programming and implementation standpoint
		100% SW			
Transactional Memory	2017	100% HW	6600K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 1)
		+100% SW			
System-level DA	2019	60% HW	10557K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 2)
		38% SW			
Executable specification	2021	200% HW	31671K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 3)
		+200% SW			
Total		+264,000%			