

INTERNATIONAL
TECHNOLOGY ROADMAP
FOR
SEMICONDUCTORS
2007 年版

デザイン

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 2007 Edition(国際半導体技術ロードマップ 2007年版)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)、2 つのタスクフォース(設計タスクフォースと故障解析タスクフォース)、経済性検討小委員会が組織され、半導体集積回路メーカー、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、2007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、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を本の形で有償頒布しても需要に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。

訳文の作成は、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ればありがたい。

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、ここも訳出せず、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。本書の巻末に用語集(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、恩田豊さん、近藤美智さん、明石理香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。

2008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)

著作権について

ORIGINAL (ENGLISH VERSION) COPYRIGHT © 2007 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: 2007 ITRS page XX, Figure(Table) YY
この和訳から引用する場合: 2007 ITRS JEITA 和訳 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: 03-5275-7258 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.....	6
デザインメソドロジー.....	6
システムレベル設計.....	8
論理、回路、物理設計.....	12
設計検証.....	17
テスト設計.....	26
製造容易化設計 (DESIGN FOR MANUFACTURABILITY (DFM)).....	32
多様化(MORE THAN MOORE)の分析.....	38
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.....	38
クロスカットTWGの課題.....	41
モデリングとシミュレーション.....	41
付録 I: ばらつきモデリングとロードマップ.....	42
付録II: DTコストと価値.....	43

LIST OF FIGURES

FIGURE DESN1	IMPACT OF DESIGN TECHNOLOGY ON SOC CONSUMER PORTABLE IMPLEMENTATION COST.....	2
FIGURE DESN2	THE V-CYCLE FOR DESIGN SYSTEM ARCHITECTURE.....	7
FIGURE DESN3	HARDWARE AND SOFTWARE DESIGN GAPS VERSUS TIME.....	8
FIGURE DESN4	SYSTEM-LEVEL DESIGN POTENTIAL SOLUTIONS.....	11
FIGURE DESN5	LOGICAL/CIRCUIT/PHYSICAL DESIGN POTENTIAL SOLUTIONS.....	16
FIGURE DESN6	DESIGN VERIFICATION POTENTIAL SOLUTIONS.....	24
FIGURE DESN7	DESIGN FOR TEST POTENTIAL SOLUTIONS.....	28
FIGURE DESN8	DESIGN FOR MANUFACTURABILITY POTENTIAL SOLUTIONS.....	36
FIGURE DESN9	MOORE AND NON-MOORE DESIGN TECHNOLOGY IMPROVEMENTS.....	38
FIGURE DESN10	POSSIBLE VARIABILITY ABSTRACTION LEVELS.....	42
FIGURE DESN11	SIMPLIFIED ELECTRONIC PRODUCT DEVELOPMENT COST MODEL....	43

LIST OF TABLES

TABLE DESN1	OVERALL DESIGN TECHNOLOGY CHALLENGES.....	4
TABLE DESN2A	SYSTEM-LEVEL DESIGN REQUIREMENTS—NEAR-TERM YEARS.....	9
TABLE DESN2B	SYSTEM-LEVEL DESIGN REQUIREMENTS—LONG-TERM YEARS.....	10
TABLE DESN3	CORRESPONDENCE BETWEEN SYSTEM-LEVEL DESIGN REQUIREMENTS AND SOLUTIONS.....	12
TABLE DESN4A	LOGICAL/CIRCUIT/PHYSICAL DESIGN TECHNOLOGY REQUIREMENTS—NEAR-TERM YEARS.....	13
TABLE DESN4B	LOGICAL/CIRCUIT/PHYSICAL DESIGN TECHNOLOGY REQUIREMENTS—LONG-TERM YEARS.....	14
TABLE DESN5	CORRESPONDENCE BETWEEN LOGICAL/CIRCUIT/PHYSICAL REQUIREMENTS AND SOLUTIONS.....	16
TABLE DESN6A	DESIGN VERIFICATION REQUIREMENTS—NEAR-TERM.....	18
TABLE DESN6B	DESIGN VERIFICATION REQUIREMENTS—LONG-TERM.....	19
TABLE DESN7	CORRESPONDENCE BETWEEN DESIGN VERIFICATION REQUIREMENTS AND SOLUTIONS.....	25
TABLE DESN8A	DESIGN FOR TEST TECHNOLOGY REQUIREMENTS—NEAR-TERM YEARS.....	27
TABLE DESN8B	DESIGN FOR TEST TECHNOLOGY REQUIREMENTS—LONG-TERM YEARS.....	27
TABLE DESN9A	DESIGN FOR MANUFACTURABILITY TECHNOLOGY REQUIREMENTS—NEAR-TERM YEARS.....	33
TABLE DESN9B	DESIGN FOR MANUFACTURABILITY TECHNOLOGY REQUIREMENTS—LONG-TERM YEARS.....	33
TABLE DESN10	CORRESPONDENCE BETWEEN DESIGN FOR MANUFACTURABILITY REQUIREMENTS AND SOLUTIONS.....	37
TABLE DESN11	NEAR-TERM BREAKTHROUGHS IN DESIGN TECHNOLOGY FOR AMS.....	40
TABLE DESN12	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。2007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。

2007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。

2007年版は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。

- 設計のハードウェア局面では、検証規模とテストコストは急上昇し続け、製造性考慮設計(DFM)の問題は設計フローに浸透している。設計技術の革新はハードウェア設計コストを抑制している。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 2007 年の推定値は 1500 万ドルであり、一方 1993 年から 2007 年までに設計技術の革新が行われなかった場合には、9 億ドルに達していたことが分かる。
- 設計のソフトウェア局面では、ホモジニアス/ヘテロジニアス双方によるマルチコアの時代に入ったとすると、組込みシステム開発コストの 80%以上を IC 設計が占めるとされている。2007 年では、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、全体の設計コストは約 4000 万ドルに達する。設計のソフトウェア局面では、次の 15 年の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 年に存在する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、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 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

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。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。

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を紹介する。これらを含めた取り組みが特定用途向け、特定システムドライバ向け設計技術の例示となる。

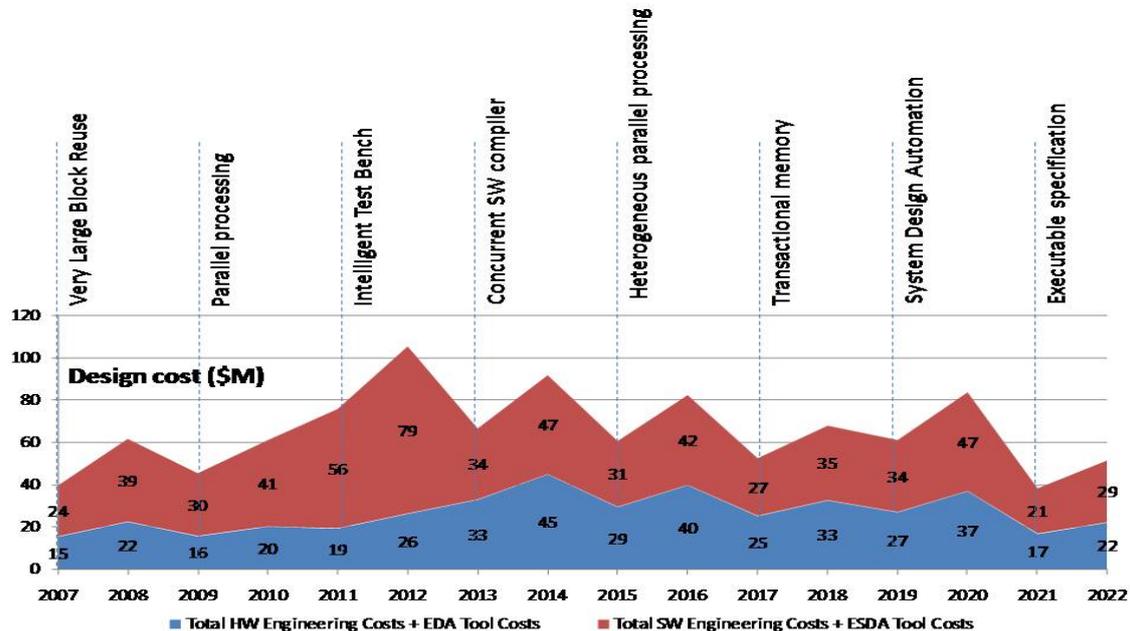


Figure DESN1 Impact of Design Technology on SOC Consumer Portable Implementation Cost

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が主要テーマとなる。最適化においては、レイアウト密度、動作周波数、消費電力、テストビリティあるいは設計期間のように多数のトレードオフが求められる。このため設計技術の質的な評価は、設計手法や設計対象商品に依存する。さらに ITRS 技術世代が必要とされる要素技術が実現したとき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。

全体的な課題

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

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

- * デバイスの寄生効果および電源/しきい値電圧の理想モデルから外れたスケーリング (リーク、パワーマネジメント、回路/デバイス革新、電流供給)
- * 高周波デバイスでの配線 (雑音/干渉、シグナルインテグリティ解析とマネジメント、基板とのカップリング、クロス・カップリングによる遅延変動)
- * 製造ばらつき (統計的なプロセスモデリングと特性抽出、歩留り、リーク電力)
- * 製造ハンドオフの複雑度 (レチクル改良、マスク描画と検査方法、NRE(Non-Recurring Engineering) コスト)
- * デバイス性能と関連のある、グローバル配線性能のスケーリング(通信、同期)
- * 信頼性の低下 (ゲート絶縁体のトンネリングおよびブレイクダウン特性、ジュール熱およびエレクトロ

マイグレーション、単発的な擾乱、フォールトトレラント)

- * プロセスばらつき (ライブラリ特性抽出、アナログとデジタル回路性能、エラートレラント設計、レイアウト再利用、信頼性が高く予測可能なインプリメンテーションのための設計基準)。

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

システム複雑度は、プロセス微細化により可能となるトランジスタ数の指数関数的な増加と関連するが、消費者からの多機能化、低コスト化、および Time to market 短縮要求によって拍車がかかる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
- * 設計検証とテスト一仕様設計、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一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一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 一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
- * 設計プロセス管理一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、より複雑な設計目標および設計制約の最適化と解析を同時に実現し、設計再利用および設計最適化のための製造コストを付加的に考慮することを認め、そして、組み込みソフトウェア設計や製造インタフェースといった付加的な領域も含めねばならない。シリコン複雑度とシステム複雑度の膨大な広がり、それ自体がまた、設計技術と EDA 業界のロードマップ化への課題である。

5つのクロスカット課題一1)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)信頼性一が与えられるが、これらの解決策候補(Potential Solutions)は設計技術の全分野にまたがり、かつ、この背景には設計コストについての課題が横たわっている。この中の最初の3つは、ITRS Executive Summary

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)。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。

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に密接に関連し、従って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。

Table DESN1 Overall Design Technology Challenges

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

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

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

ばならない。

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ジングは、配線材料やパッケージ技術のロードマップとともに、パワーマネジメントと電流供給に関連した様々な課題を提示する。それらの課題を以下に記す。

- 1) 全体的なロードマップ技術指標およびシステムドライバ章に基づく推定によれば、HP(High Performance) MPU の消費電力は、アセンブリおよびパッケージング章の中で述べられた、高機能単一チップパッケージの電力限界を超過するとともに、許容された電力密度である $250\text{W}/\text{cm}^2$ をも超過する。SOC-LP(Low Power and low cost) PDA ドライバは、ロジック規模および処理能力が指数関数的に増加し続けるときには、平均およびスタンバイ電力の考慮が必要である。設計技術は、システムドライバ章の Table12 で示されるように、結果として生じるパワーマネジメント・ギャップの対策に取り組まねばならない。
- 2) 電力密度の増加は、信頼性と性能に対する熱の影響を悪化させる。他方で、供給電圧の減少は、スイッチング電流およびノイズを悪化させる。これらの傾向は、チップ上の配線リソース(たとえば、アセンブリやパッケージのロードマップに照らして、バンプ数や保護膜の開口サイズを決定し IR ドロップをコントロールする)、ATE(Automatic Test Equipment)能力、およびバーンインの枠組みを圧迫する。
- 3) 高性能、低動作電力、低待機電力を統合したデバイスを実現するには、多くの自由度 – マルチ V_t 、マルチ 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、スケージ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、設計技術の適切な介入もしくは協調で、(より経済的に)解決できる可能性がある。将来の技術ノードの実現可能性は、このような「技術の壁の共有」に拠ることになる。以下に、いくつかの例を示す。

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

クロスカット課題4－干渉

グローバル配線のスケージ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ジングと SOC 化については、より柔軟かつ強力なメソッドが要求される。今後の検討課題として、ノイズの無歪限界(特にロー・パワーデバイスやダイナミック回路)、多数の容量結合もしくは誘導結合を含む配線、供給電圧の IR ドロップとグランド電圧のはね返り、デバイスオフ電流への熱衝撃、配線抵抗、基板とのカップリングなどがあげられる。基本的な設計技術の課題は、設計のすべてのレベルにおいて、ノイ

ズと干渉の特性抽出、モデル化、解析および見積り方法を改善することである。

クロスカット課題5－信頼性

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

- 1) 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の消費電力要求も増大する。
- 3) “non-catastrophic”なゲート酸化膜のブレイクダウンなど、原子スケールの影響を反映した、新たな「ソフト」欠陥の基準が必要となる。一般に、システムが大きくなりすぎると最終段階での機能テストが不可能になるため、設計のロバスト性を考慮した自動回路挿入が、より重要となる。とりうる対策としては、冗長論理回路の自動挿入、フォールトトレラントを考慮したオンチップ回路再構成、適応回路もしくは自己修復回路の開発、そしてソフトウェアに基づくフォールトトレラントが含まれる。

詳細な設計技術の課題

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

デザインメソドロジー

チップの設計と実装の工程は、様々な技法、ツール群、そして設計者の入力が予想どおりに製造可能な製品となるような有効なメソドロジーを必要とする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設計メソドロジーはしばしば無視されている。各技術世代はより多くの問題を設計者に考えさせる；したがって、新しい現象を解析するための新しい手法やツールが開発され、設計者がクリティカルな設計上の決定をすることを助けるようであればならない。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。

“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、新しいメソドロジーやツールに向けた、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 EDA に関して突きつけてくるとてつもない課題も解決しなければならない。シリコンデバイスと製造プロセスのシュリンクは、今までは日常的に行渡った強力なシリコン・ソリューションの基盤だった。これは、仕様データからスムーズにレイアウト・データに変換し、そして異なる抽象度の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、こうした設計技術は、もはや当然のことでは有り得ない。

³ 設計メソドロジー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。設計メソドロジーは設計技術とははっきりと異なるものである。設計技術はメソドロジー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。すべて既知の設計メソドロジーは、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。

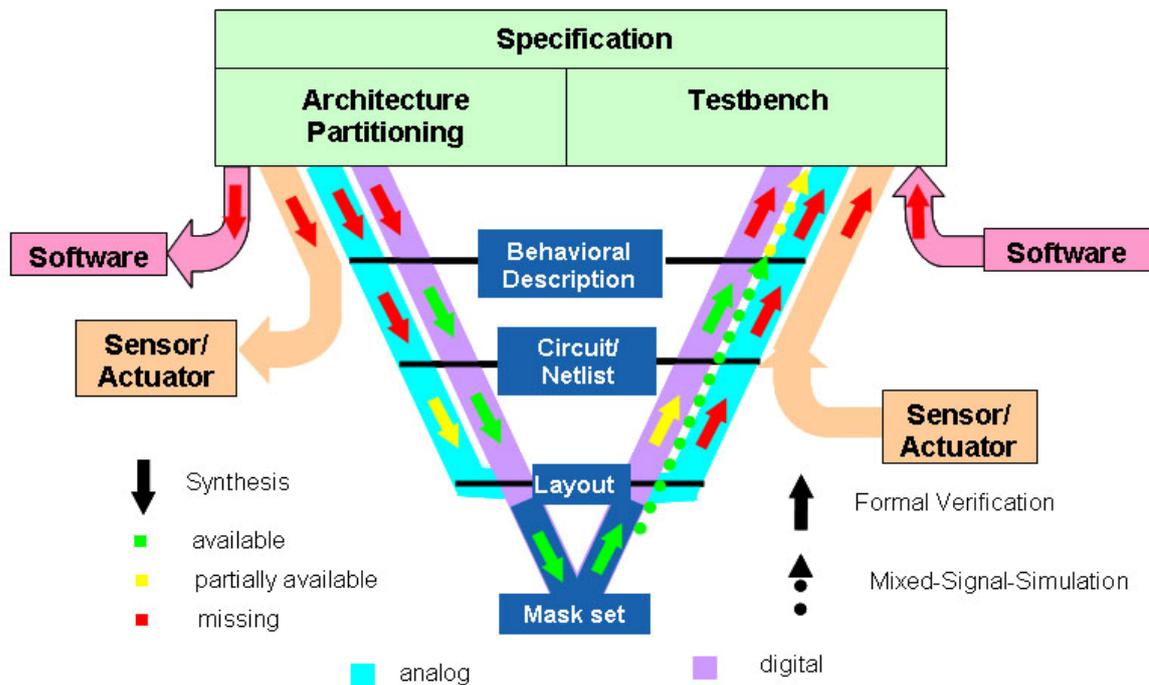


Figure DESN2 The V-Cycle for Design System Architecture⁴

新しい SOC のコストや time-to-market が設計技術に課する要求は、複雑な設計プロセスのあらゆる部分に広まり、それは二つの主要なパス（インプリメンテーションのパスと検証のパス）から構成されている（図 DESN2）。この図は、V-サイクルと呼ばれる設計システム・アーキテクチャを表しており、“More Moore”と“More than Moore”の両方の設計の様相を統合している。図中の矢印は、開発済みの設計システム環境で可能なもの（緑色）、部分的に可能なもの（黄色）を示し、将来の EDA に対する要求は赤色の矢印で示す。

システム仕様はアーキテクチャからマスクまで段階的なリファインを受ける。こうした SOC 設計のアイデアの骨子は、まず初期段階でシステムを様々な可能性を探索し、同時にシステムレベルの記述を詳細なレベル（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）に変換して行く。そして、各リファインのステップの正しさを検証テクノロジーが保証しなければならない。SOC の構成部品や動作を超えて、将来のインプリメンテーションと検証のプロセスでは、システム・ソフトウェアやセンサー、アクチュエータが取り扱われなければならない。設計システムのデジタル設計パス（図中の紫色）では、動作レベルの抽象度より上のレベルのツールやメソッドが必要になる。すべての抽象度のレベルで完備したソリューションがほとんど見つけられないため、アナログ設計パスはより一層困難である。結果として、アナログ設計の自動化されたプロセスは、現在の設計環境では、まだ未解決である。まとめると、将来の要求は次の 2 要素となる。1) More than Moore の方向が、現今の SOC フローの外側にある諸相を要求する。とりわけ、片やソフトウェア、もう片やセンサー、アクチュエータであり、現状は欠落している。2) 将来にわたって、SOC 設計フローのための新しい要求が出てくる。とりわけ、デジタルとアナログの両方を高い抽象度のレベルでサポートするツール。これは、必然的に高位レベルの仕様からの動作合成と、対応する抽象度の検証を必然的にもなうが、図 DESN2 に示すように現状未達成である。アナログ、ミックスドシグナルのフローではさらに、各設計レベルでのアナログ回路合成とアナログ回路検証のツールが必要となる。

ナノスケール技術への移行によって、仕様から製造までのすべての設計プロセスが、相互依存性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わってくる。その結果として、SOC の設計生産性は、Moore の法則で定式化されるナノエレクトロニクス技術の革新のスピードに追いつくことができない（図

⁴ Elements of this discussion were initially developed as part of a recent update of the Strategic Research Agenda (SRA) within the European Nanoelectronics Initiative Advisory Council (ENIAC) chapter on Design Automation. Figure DESN2 shows status of the EDA design flow from an automotive point of view, but is general enough to demonstrate the future EDA requirements. The figure was developed by Peter van Staa (Bosch).

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

何十年もの期間設計者は、ほとんど設計自動化ツールのサポート無しで、システムを様々な抽象レベル(ブロック図、不完全な状態図、プログラミングモデル、等々)で考えてきた。設計生産性における必要な進化が達成されるためには、この状況は近い将来変わらなければならない。ハードウェア、ソフトウェアを含んだシステムの仕様、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。

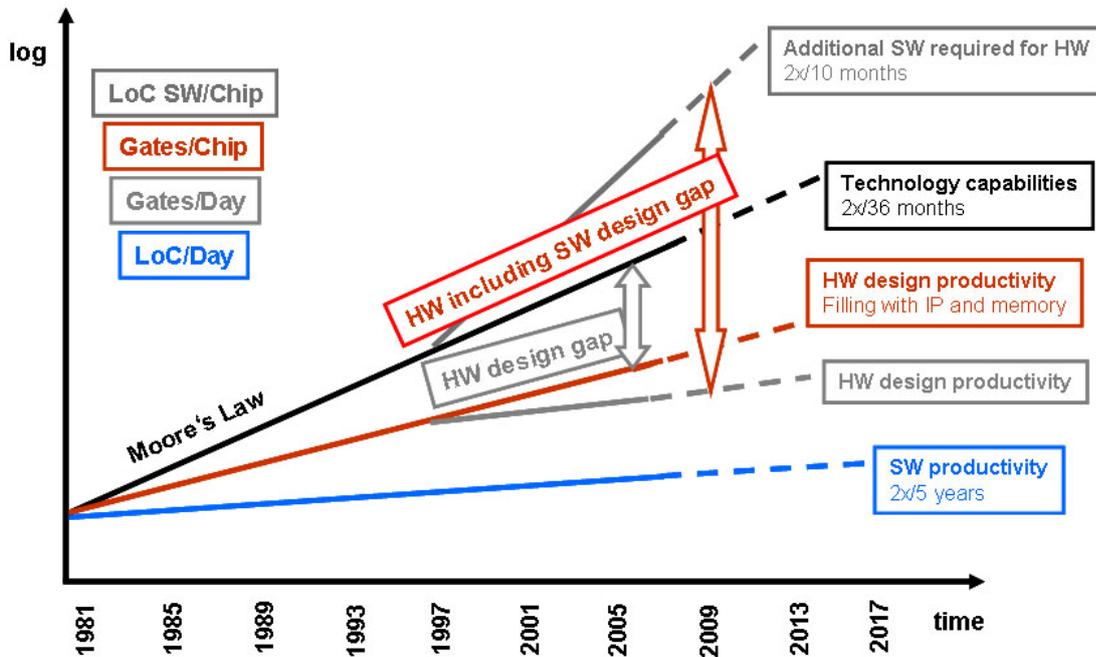


Figure DESN3 Hardware and Software Design Gaps Versus Time⁵

システムレベル設計⁶においては、メソドロジーの側面がツールの側面より急速に困難度を増している。シングル・チップ上に巨大な複雑度を持つシステムが実現可能となるが、一方でこれを現実のものとするためには、今日と比べて約 50 倍以上の設計生産性が要求される。シリコン複雑度は、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することを意味する。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の理由によって、非現実的となる。また、チップを横切る信号は、もはや 1 クロックサイクルでは実現できなくなる。したがって、システムレベル設計は、ネットワークキングと

⁵ This figure shows the demand for software which is currently doubling every 10 months, the capability of technology which is currently doubling every 36 months all well as the hardware and software design productivity. Whereas the hardware design productivity improved over the last couple of years by filling the silicon with multi-core components and memory providing functionality only with additional software, the productivity especially for hardware-dependent software is far behind and only doubling every 5 years. This ITRS version includes therefore for the first time additional figures for illustrating these new software requirements. This material is based on ITRS data on one hand and on additional input from Infineon and STMicroelectronics and has been presented by Wolfgang Ecker from Infineon at DATE 2007. The red arrow summarizes the new design gap including both hardware and software.

⁶ システムレベルでは、シリコンのリソースは抽象的な機能やブロックとして位置付けられている。設計のターゲットは、ソフトウェア(高位のアセンブリ言語での埋め込みコード、コンフィギュレーション・データなど)とハードウェア(コア、ハードワイアされた回路、バス、リコンギギャラブル・セルなど)から構成される。ハードウェアは、インプリメントされた回路に対応するし、ソフトウェアは、ハードウェアで処理される機能の、論理的な抽象表現に対応している。動作(behavior)とアーキテクチャは設計自由度では独立であり、ソフトウェアとハードウェアはアーキテクチャの構成要素となる。動作の集合体がシステム・ファンクションを定義、一方でアーキテクチャ・ブロックの集合体がシステムプラットフォームを定義する。システム機能からシステム・アーキテクチャ上へのプラットフォーム・マッピングがシステムレベル設計の正に核心であり、また、システムの複雑度の増大や異種混交(アーキテクチャ上や機能上)によって、困難度を増している。

分散処理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。組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。ハードウェア中心である今日では、図 DESN3 に見られる設計全体のエフォートを支配しているソフトウェアだけでなく、ハードウェアと連携して機能を実現するハードウェア依存ソフトウェア (hardware-dependent software) まで、結局、SOC インテグレーションや、検証の工程で取り扱われなければならない。

* ITRS ロードマップの 2007 年版では、システムレベル設計の技術課題は、大部分が前版と変わっていない。それは、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 年以上も研究課題であり続け、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、効果的な動作合成は現在も実現されてない。また、現在の SOC 設計工程が主にベストエフォートによって駆動されているところでは、今後、「自己治癒」、「自己構成」、「誤り補正」等の新たな方向性は避けようがない。設計技術は、“More Moore” と “More than Moore” の領域で、最終的に費用効果がよいシステム実装と、社会的要求と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のに、効果的な役割を演じて行くであろう。表 DESN2 に、次世代以降の技術世代におけるシステムレベル設計に対する定量的な要求を示す。

Table DESN2a System-Level Design Requirements—Near-term Years

Year of Production	2007	2008	2009	2010	2011	2012	2013	2014	2015
DRAM ½ Pitch (nm) (contacted)	65	57	50	45	40	36	32	28	25
Design Reuse									
Design block reuse [1] % of all logic	35%	36%	38%	40%	41%	42%	44%	46%	48%
Platform Based Design									
Available platforms [2] Normalized to 100% in the start year [3]	87%	83%	75%	70%	60%	55%	52%	48%	45%
Platforms supported [4] % of platforms fully supported by tools [5]	10%	25%	35%	50%	57%	64%	75%	80%	85%
High Level Synthesis									
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	60%	63%	66%	70%	73%	76%	80%	83%	86%
Reconfigurability									
SOC reconfigurability [7] % of SOC functionality that is reconfigurable	28%	28%	30%	35%	38%	40%	42%	45%	48%
Analog/Mixed Signal									
Analog automation [8] % versus digital automation [9]	17%	17%	24%	24%	27%	30%	32%	35%	38%
Modeling methodology, description languages, simulation environments [10] % vs. digital methodology	55%	58%	60%	62%	65%	67%	70%	76%	78%

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known

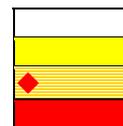


Table DESN2b System-Level Design Requirements—Long-term Years

Year of Production	2016	2017	2018	2019	2020	2021	2022
DRAM ½ Pitch (nm) (contacted)	22	20	18	16	14	12	10
<i>Design Reuse</i>							
Design block reuse [1] % of all logic	49%	51%	52%	54%	55%	57%	58%
<i>Platform Based Design</i>							
Available platforms [2] Normalized to 100% in the start year [3]	43%	40%	37%	35%	32%	29%	27%
Platforms supported [4] % of platforms fully supported by tools [5]	90%	92%	94%	95%	97%	99%	100%
<i>High Level Synthesis</i>							
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	90%	92%	94%	95%	97%	99%	100%
<i>Reconfigurability</i>							
SOC reconfigurability [7] % of SOC functionality that is reconfigurable	50%	53%	56%	60%	62%	65%	68%
<i>Analog/Mixed Signal</i>							
Analog automation [8] % versus digital automation [9]	40%	43%	46%	50%	52%	55%	58%
Modeling methodology, description languages, and simulation environments [10] % versus digital methodology	80%	83%	86%	90%	92%	95%	98%

表 DESN2a と b のためのメモ。

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

定義

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

原理

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

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

ここに

reuse: 基準とする年における再利用。

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

cgrowth: 設計複雑度の(予測)平均上昇率

なお、設計者数および設計期間は、一定であると仮定する。

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

[2] 定義

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

原理

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

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

[4] 定義

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

原理

自動化の推進は、システム・レベル・デザインにおける成功の鍵である。

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

[6] 定義

見積り結果が、製造した IC における測定結果と一致する度合い。

原理

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

[7] 定義

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

原理

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

[8] 定義

アナログ設計における自動化の度合い

原理

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

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

[10] 定義

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

原理

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

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

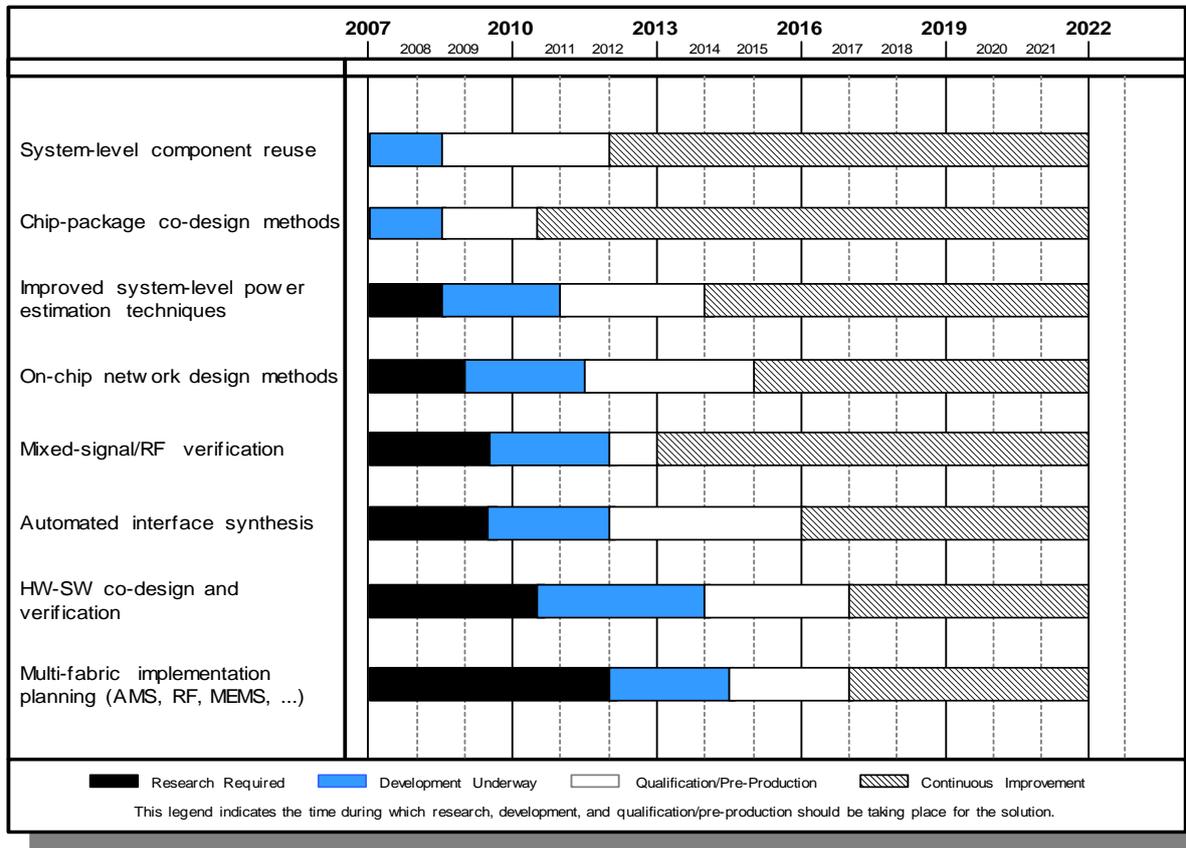


Figure DESN4 System-Level Design Potential Solutions

Table DESN3 Correspondence Between System-Level Design Requirements and Solutions

Requirement	Solution	Explanation of the Correspondence
Design block reuse	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
Available platforms	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
Platforms supported	Automated interface synthesis	Automated interface synthesis is one building block to an integrated synthesis flow for whole platforms.
	Automated HW-SW co-design and verification	Required for integrated, platform-based system development
Accuracy of high level estimates	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
SOC reconfigurability	On-chip network design methods	To provide flexible, reconfigurable communication structures
Analog automation	Multi-fabric implementation planning (AMS, RF, MEMS, ...)	Multi-fabric implementation planning for AMS and RF components are a building block to analog automation
Modeling methodology, description languages, simulation environments	Mixed-Signal/RF verification	As in digital design, verification is an increasingly critical and time-consuming activity in the design flow

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）および制約（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および、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、およ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管理可能なステップの連続で構成しているが、ステップ間の連携は疎であり、急速に時代遅れにな

るであろう。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、下流に送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き、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イタレーションである。このようなイタレーションは、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。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のイタレーションは、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。

Table DESN4a Logical/Circuit/Physical Design Technology Requirements—Near-term Years

Year of Production	2007	2008	2009	2010	2011	2012	2013	2014	2015
Asynchronous global signaling: % of a design driven by handshake clocking	7%	11%	15%	17%	19%	20%	22%	23%	25%
Parameter uncertainty: %-effect (on signoff delay)	6%	8%	10%	11%	11%	12%	14%	15%	18%
Simultaneous analysis objectives: # of objectives during optimization	4	5	6	6	6	6	7	8	8
Circuit families: # of families in a single design	3	3	4	4	4	4	4	4	4
Synthesized analog content: % of total design analog content	15%	16%	17%	18%	19%	20%	23%	25%	28%
Full-chip leakage (normalized to full-chip leakage power dissipation in 2007)	1	1.5	2	2.5	2.75	3	3.5	4	6

Table DESN4b Logical/Circuit/Physical Design Technology Requirements—Long-term Years

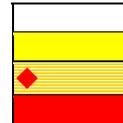
Year of Production	2016	2017	2018	2019	2020	2021	2022
Asynchronous global signaling: % of a design driven by handshake clocking	30%	30%	30%	35%	40%	43%	45%
Parameter uncertainty: %-effect (on signoff delay)	20%	20%	20%	22%	25%	26%	28%
Simultaneous analysis objectives: # of objectives during optimization	8	8	8	8	8	8	8
Circuit families: # of families in a single design	4	4	4	4	4	4	4
Synthesized analog content: % of total design analog content	30%	35%	40%	45%	50%	55%	60%
Full-chip leakage (normalized to full-chip leakage power dissipation in 2007)	8	8	8	8	8	8	8

Manufacturable solutions exist, and are being optimized

Manufacturable solutions are known

Interim solutions are known

Manufacturable solutions are NOT known



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、および、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上で、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ーを組み込み、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。

1. **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、および、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。
2. **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。設計生産性および設計コストは、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 Vt、マルチ Vdd、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およ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、ディ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およ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および保守コストに直面するため、自律また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 Vt、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 Vt 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ーノードに伴いスケールリングして算出した。

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。

3. **Automated handshake logic/circuit tools (ハンドシェイク論理/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
4. **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)** — テクノロジー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。
5. **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。
6. **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。

7. **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。
8. **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。
9. **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および物理合成。
10. **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。

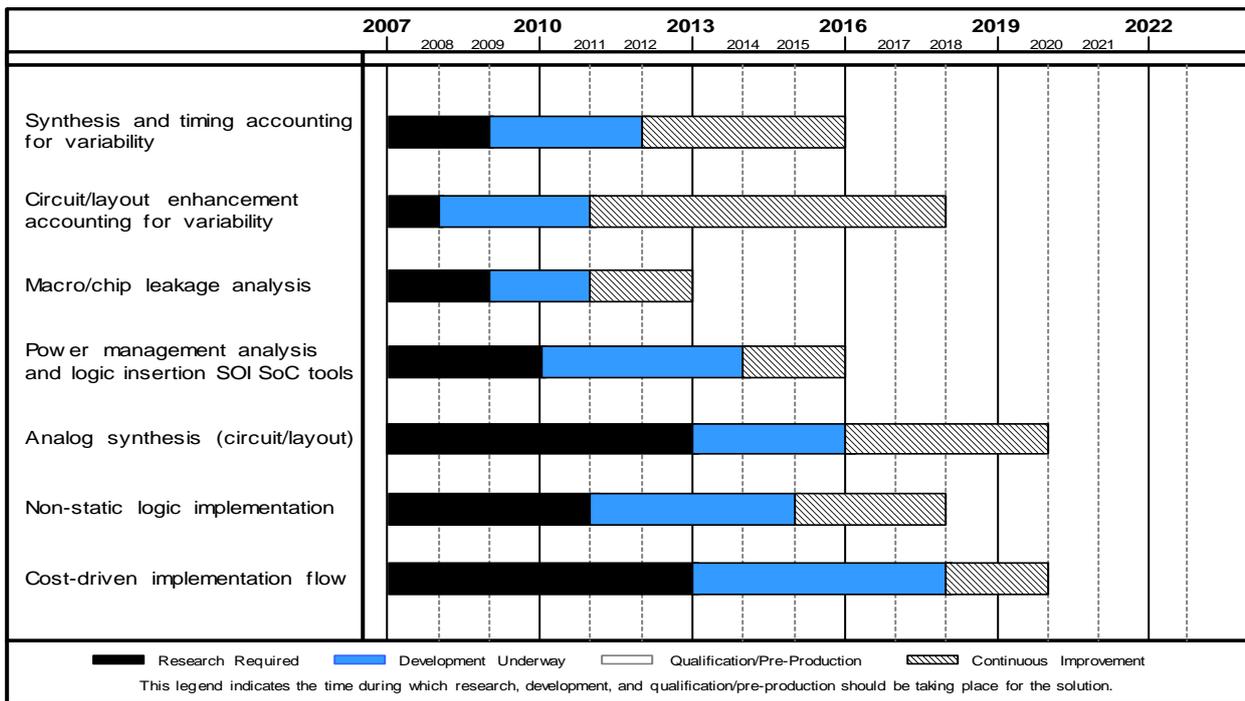


Figure DESN5 Logical/Circuit/Physical Design Potential Solutions

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または手法を構成している。

Table DESN5 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
Simultaneous analysis objectives	Power management analysis and logic insertion SOI SOC tools	Requires budgeting of area/power/timing constraints
Simultaneous analysis objectives	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</i> # of circuit families in a single design	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

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にも達すると複数の報告がある。このような工数割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の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み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で数倍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、エミュレーションは、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、コンポーネント間の複雑な未検証の通信で見つける。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間に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数十倍から数百倍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、将来のテクノロジノ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しなければならないとすれば、検証用ツールのフォーマル検証とシミュレーションへの分類は、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つに進展していくであろう。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補足するものである。しかし、利用される領域(domain)を考慮する必要があるため、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ことを、この表は示している。

Table DESN6a Design Verification Requirements—Near-term

Year of Production	2007	2008	2009	2010	2011	2012	2013	2014	2015
Productivity									
Design size verifiable by 1 engineer-year (in millions of transistors - based on an SOC design and a 10-person engineering team) [1]	7.9	10.3	13.5	17.6	23.1	30.3	39.8	52.3	69.6
Methodology									
Design errors exposed using formal or semi-formal verification (% , versus simulation)	4.7	7.1	9.4	11.8	14.1	16.5	18.8	21.2	23.5
Effort spent on system-level verification: software, hardware and electrical effects (%)	11.6	13.1	14.7	16.3	17.8	19.4	20.9	22.5	24.1
Portion of the design specification formalized for verifiability (%)	13.8	17.5	21.3	25.0	28.8	32.5	36.3	40.0	43.8
Bugs									
Escape rate: bugs found after first tapeout (per each 100K lines of design code)	8	7	7	7	6	6	6	6	5

Bugs found after system integration until tapeout (per each 100K lines of design code)	62	68	74	79	85	91	97	103	109
Reuse									
Portion of the verification infrastructure (e.g., test beds, coverage, checkers) which is newly developed (versus reused components and acquired IP) (%) [2]	73.9	70.8	67.8	64.7	61.6	58.6	55.5	52.5	49.4
Portion of the verification infrastructure which is acquired from third parties (i.e., verification IP) (%) [2]	15.5	18.3	21.1	23.8	26.6	29.4	32.1	34.9	37.6
Functional coverage									
Portion of design for which verification quality is evaluated through functional coverage (%)	46.5	49.7	52.9	56.2	59.4	62.6	65.9	69.1	72.4
Coverage goal density (expressed as number of coverage goals for each million transistors of the design) [3]	1294	1608	1922	2235	2549	2863	3176	3490	3804

Table DESN6b Design Verification Requirements—Long-term

Year of Production	2016	2017	2018	2019	2020	2021	2022
Productivity							
Design size which can be verified by 1 engineer-year (in millions of transistors - based on an SOC design and a 10-person engineering team) [1]	91.8	121.0	159.7	210.9	278.6	368.5	487.6
Methodology							
Design errors exposed using formal or semi-formal verification (% , versus simulation)	25.9	28.2	30.6	32.9	35.3	37.6	40.0
Effort spent on system-level verification: software, hardware and electrical effects (%)	25.6	27.2	28.8	30.3	31.9	33.4	35.0
Portion of the design specification formalized for verifiability (%)	47.5	51.3	55.0	58.8	62.5	66.25	70
Bugs							
Escape rate: bugs found after first tapeout (per each 100K lines of design code)	5	5	4	4	4	3	3
Bugs found after system integration until tapeout (per each 100K lines of design code)	115	121	126	132	138	144	150
Reuse							
Portion of the verification infrastructure (e.g. test beds, coverage, checkers) which is newly developed (versus reused components and acquired IP) (%) [2]	46.4	43.3	40.2	37.2	34.1	31.1	28.0
Portion of the verification infrastructure which is acquired from third parties (i.e., verification IP) (%) [2]	40.4	43.2	45.9	48.7	51.5	54.2	57.0
Functional coverage							
Portion of design for which verification quality is evaluated through functional coverage (%)	75.6	78.8	82.1	85.3	88.5	91.8	95.0
Coverage goal density (expressed as number of coverage goals for each million transistors of the design) [3]	4118	4431	4745	5059	5373	5686	6000

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

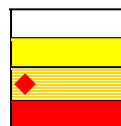


表 DESN6a と b の注釈

1 行目 (productivity)を除き、2007年の初期値は業界調査データから引用して、最終データ(2022年)を予測値として、直線的に増加もしくは減少する傾向としている。

[1] 設計生産性要求値は、システムドライバ章の SOC design productivity requirement から引用し、1年で検証できるロジック数を求めるため、設計生産性要求値を検証で費やす設計工数比率で割っている。検証で費やす設計工数の割合は、2007年の70%から2022年の50%まで線形に変化するとした。

[2] 検証再利用の考え方は、新規検証環境 + 再利用 + サードパーティの検証 IP = 100%とした。表は、新規検証環境構築とサードパーティ検証 IP についての値を示している。

[3] 言い換えると、それぞれの設計が合成された場合に100万トランジスタに相当する論理となる設計もしくは設計のある機能側面に対して挿入されるアサーションやチェッカの数。

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あることである。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、無視できるくらいわずかなカバレッジしか得られない。エミュレーションとラピッド・プロトタイプングはソフトウェア・シミュレーションに比べて数 10 倍以上の速度が出るので、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が判断するのに、専門家でさ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。検証し易い設計の手法につながるのであれば、検証アルゴリズムの経験則を改善することでも良いし、与えられた回路の検証困難度の特性解析を容易にすることでも良い。

検証の評価指標(メトリクス) — 短期的に重要な検証課題として、検証の質を計る必要性があげられる。特に、カバレッジという重要な概念は必須である。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、動的シミュレーション実行を静的なソースコードに対応付けていることにある。つまりシミュレーション実行時には、様々な実行可能なパスがあるために、存在する多くの潜在的な問題を見逃すことにある。例えば、状態カバレッジや遷移パス・カバレッジなどの構造カバレッジは、設計の有限要素マシンに着目しており、どの状態がシミュレーションで確認されたかを測定する。しかし一般的に設計全体を 1 つの有限要素マシンとして扱うのは不可能であり、カバレッジの際にいくつかの有限要素マシンが対象となるだけである。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 — SOC(Systems-on-a-Chip)のような複雑なシステムの検証は、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・レイヤで機能の多くを提供するため、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、基本的で低レベルのソフトウェア検証を可能とする技術を開発することである。長期的な課題は、ソフトウェアに適用する検証容易化設計を理解することと、ソフトウェアとハードウェア—ソフトウェア間のインタフェースの確実な検証手法を開発することである。

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バ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きている、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 — ソフトウェア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。アナログの電気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の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バ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

現在入手可能な、もしくは上記課題克服に向け開発中の解法を示す。図 DESN6 は開発チームに期待する解決策の利用可能時期とともに、検証危機に取り組むべき重要な方向性をまとめている。

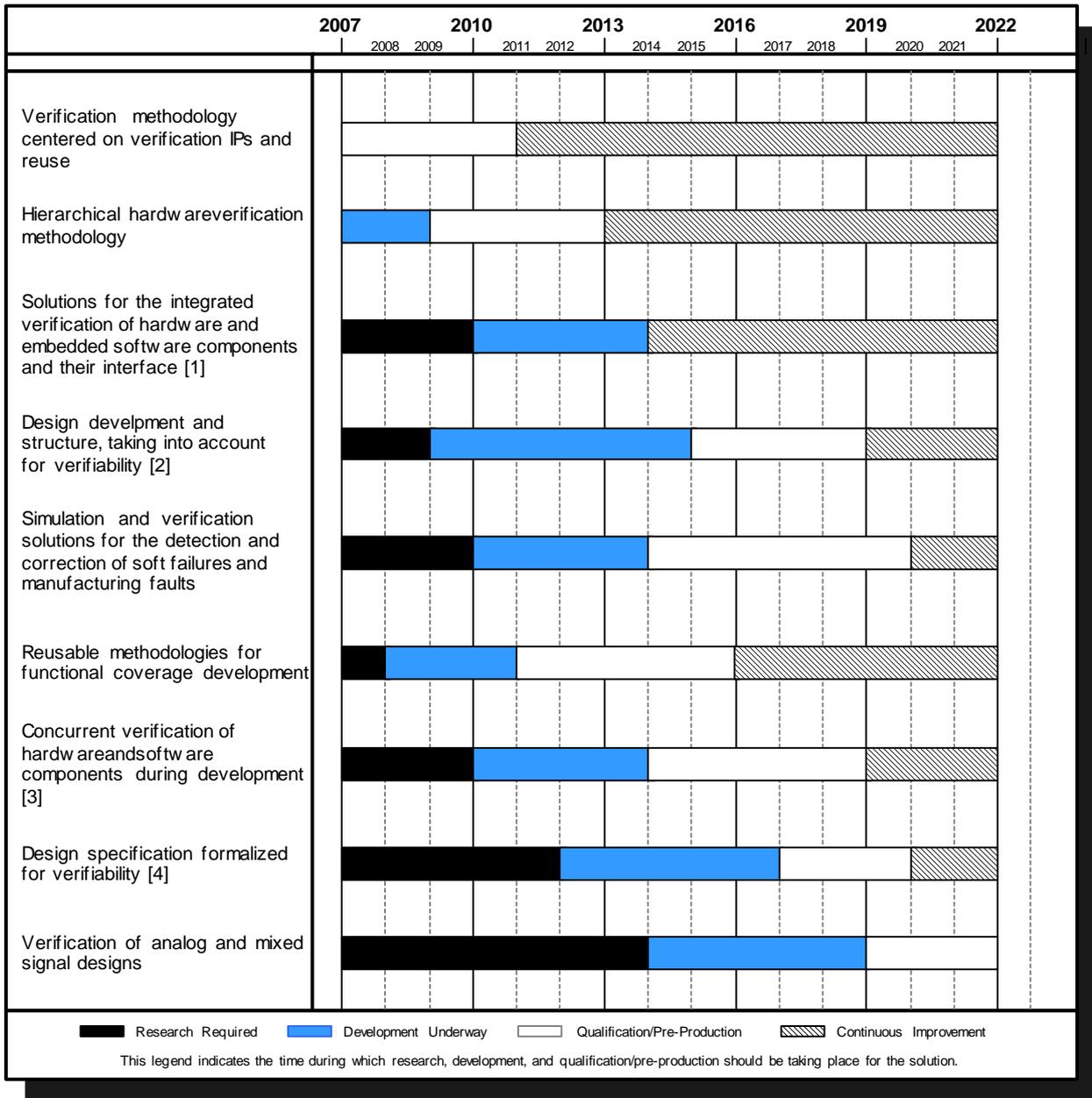


Figure DESN6 Design Verification Potential Solutions

図 DESN6 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、ソフテラー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。

Table DESN7 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、SIP は困難な課題を DFT (Design for Test) に提起している。テスト産業は、コンポーネント・ベース設計のための高位テスト合成から、非常に高性能なアナログや RF システムのためのノイズ、干渉、消費電力にいたる多様な問題に対応して行かねばならない。それらの多くの問題は、設計の初期の段階から最適なテストビリティが考慮され、DFT が組み込まれた場合に限り解決される。さらに、メソロジーに上記のことをテスト検証の分析と設計と一体化した要求として取り込む必要がある。論理回路—物理設計、そして設計検証における結果(設計フロー、ベクトル、故障モデル、感度)はテスト設計においても再利用される。メモリ、DSP、PE、SOC、アナログミックスドシグナル/RF、そして MPU といった様々な製品セグメントの種類をこえて高度なテストの課題、つまりオンチップ DFT、BIST、テストビリティの拡張、ならびにチップ開発の計画段階での、製造段階でのテスト装置の事前計画の前倒しを要請している。

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

表 DESN8 に DFT への技術要求を、図 DESN7 に DFT への解決策候補を 4 つの主要なシステムドライバに対して記載する。それらは、アナログ、ミックスドシグナル/RF、MPU/PE/DSP、メモリ、そして一般的な SOC/SIP の各システムドライバである。

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

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

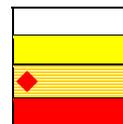
Table DESN8a Design for Test Technology Requirements—Near-term Years

Year of Production	2007	2008	2009	2010	2011	2012	2013	2014	2015
<i>System Driver: Analog/Mixed-signal/RF</i>									
All-digital DFT for analog/mixed-signal/RF circuits and systems. % digital circuits in DFT implementations	40	45	50	55	60	60	60	60	80
Correlation of DFT results with existing specification-based test methods. % results correlated	40	45	50	55	60	60	60	60	80
Availability of fault/defect models for DFT-oriented test methods. % AMS/RF blocks with accepted fault models	25	30	35	40	45	50	55	60	65
<i>System Drivers: MPU/PE/DSP</i>									
DFT coverage of digital blocks or subsystems. % blocks with DFT	70	70	70	75	75	75	80	80	85
DFT for delay test of critical paths. % paths covered	55	55	60	60	60	60	70	70	70
DFT for fault tolerance in logic blocks. % blocks with fault tolerance	40	40	45	45	50	50	55	55	60
<i>System Drivers: Memories</i>									
DFT for yield improvement.	85	90	90	90	90	95	95	95	95
<i>General SOC/SIP requirements</i>									
DFT support for logic and other non-memory circuit repair. % blocks with repair	50	60	60	60	70	70	70	80	80
DFT reuse for performance calibration, and measurement purposes. % DFT circuits reused	35	35	40	40	40	45	45	50	50
DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	15	15	10	10	10	10	10	10	10
DFT efficacy in test volume reduction. Reduction factor	5x	5x	5x	10x	10x	10x	20x	20x	20x
DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	45	45	50	50	60	60	70	70	75

Table DESN8b Design for Test Technology Requirements—Long-term Years

Year of Production	2016	2017	2018	2019	2020	2021	2022
<i>System Driver: Analog/Mixed-signal/RF</i>							
All-digital DFT for analog/mixed-signal/RF circuits and systems. % digital circuits in DFT implementations	85	90	90	100	100	100	100
Correlation of DFT results with existing specification-based test methods. % results correlated	85	90	90	100	100	100	100
Availability of fault/defect models for DFT-oriented test methods. % AMS/RF blocks with accepted fault models	70	75	80	85	90	95	100
<i>System Drivers: MPU/PE/DSP</i>							
DFT coverage of digital blocks or subsystems. % blocks with DFT	85	90	90	95	95	97.5	100
DFT for delay test of critical paths. % paths covered	80	80	90	90	100	100	100
DFT for fault tolerance in logic blocks. % blocks with fault tolerance	65	70	80	90	100	100	100
<i>System Drivers: Memories</i>							
DFT for yield improvement.	98	98	98	100	100	100	100
<i>General SOC/SIP requirements</i>							
DFT support for logic and other non-memory circuit repair. % blocks with repair	80	90	90	100	100	105	110
DFT reuse for performance calibration, and measurement purposes. % DFT circuits reused	60	60	70	70	70	72.5	75
DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	5	5	5	5	5	5	5
DFT efficacy in test volume reduction. Reduction factor	20x	50x	50x	50x	50x	50x	50x
DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	90	80	90	100	100	100	100

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



アナログ、ミックスドシグナル、RF の信頼性の問題は、誘導時間の過剰や動作環境による、パラメトリックな減衰により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

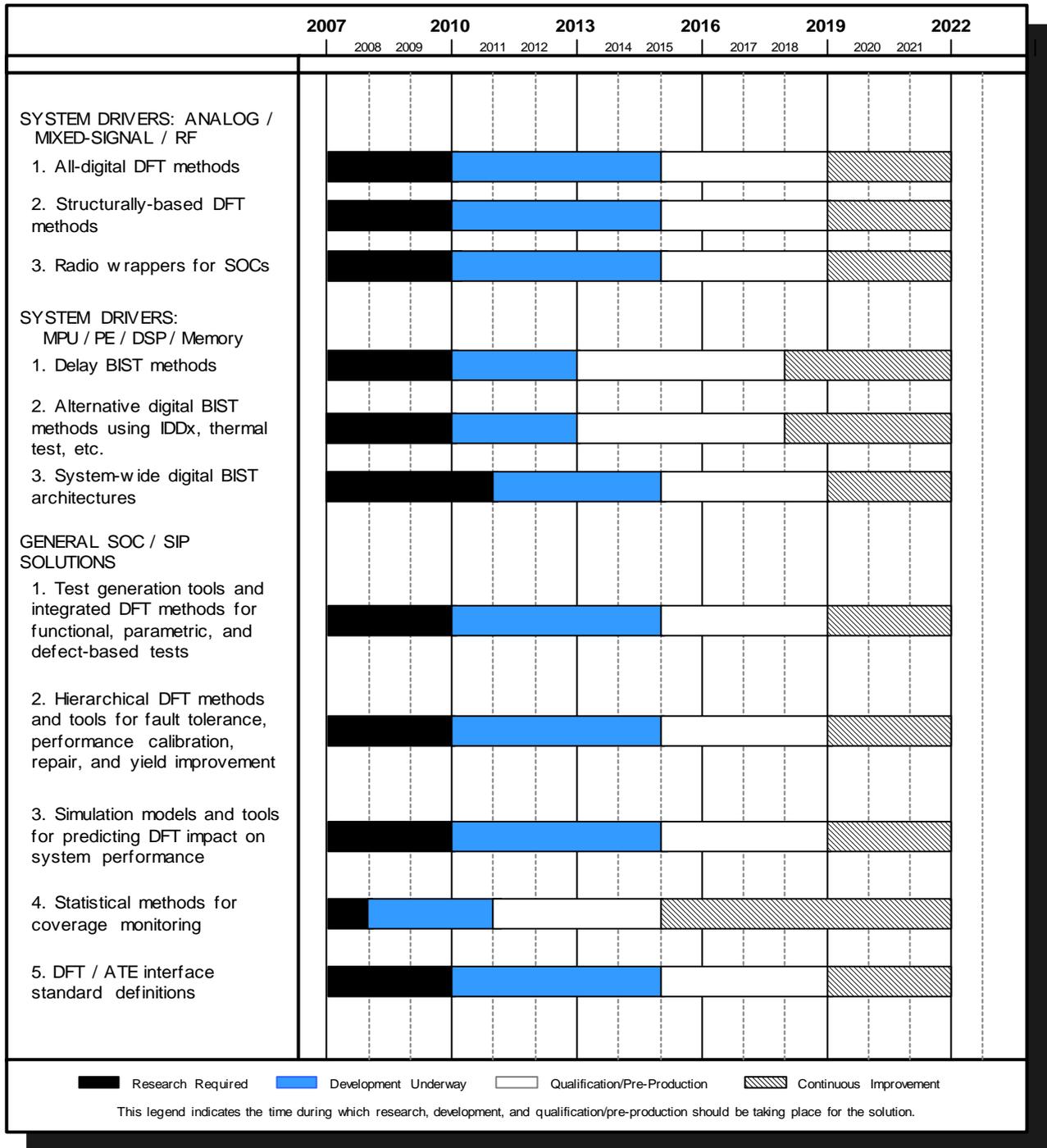


Figure DESN7 Design for Test Potential Solutions

あるパラメータを開発することが極めて重要である。多くのアナログ故障は、時間、電圧、フェーズなどの連続量のパラメータが、仕様の範囲からはみ出すことによっており、製造のばらつきや不整合から生じてい

る。効率的で効果的なアナログの故障グレーディングとテスト生成のための故障モデルが必要となる。アナログ、ミックスドシグナル、RF の設計では、故障シミュレーションの計算複雑度を最小化し、同時に高いシミュレーション精度を維持する EDA ツールが必要である。アナログの故障モデルとプロセスの欠陥モデルを合わせこむことは非常に困難だが、SOC や SIP 上にアナログ、ミックスドシグナル、RF サブシステムを搭載した将来の製品のためには最高に重要なことである。

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

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

ここでのドライバに分類される多くの論理システムには成熟した解決策が存在している。残されている課

題は、ブロックレベルの解決策よりも、むしろシステムレベルの解決策への要求である。上述で指摘したように、デジタルシステムの性能にかかわる幾つかのクリティカルなパラメータ(遅延、消費電力、ジッター、動作温度など)は、未だに有効な DFT の解決策を求めている。これらのパラメータをキャラクタライゼーション・テストと量産製造テストの両方で決定するために、コスト効率の高い DFT や BIST の手法が非常に望まれる。統合されたキャリブレーションと修復機能を提供するような解決策が、電子システムの設計とテストアーキテクチャの全体に極めて重要なものとして現れるだろう。今日の故障に特化した DFT/BIST 手法と、新しいデジタルのパラメータに特化した DFT/BIST の両方を使って、システムテストの設計者は、様々な手法をそれを適用する局面で最適に有効に適用することで、結合力のあるシステムワイドな DFT/BIST のアーキテクチャを作り上げなければならない。つぎはぎちなテストアーキテクチャやいい加減なテスト統合は、テスト時間やテストデータ量に関して、ロジスティックスや実行計画に問題を起しがちで、それは DFT/BIST の有利性を減じることになる。デジタル DFT/BIST の多様な手法が与えられることにより、システムテストの設計者は、特定のデジタルシステムに対する一元的に統合されたテストアーキテクチャを作り上げることが可能となる。多様性の中の統一が、どのシステムワイドなデジタルのテストアーキテクチャに最も重要な要件である。

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きている。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。

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

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

システムのレベルでは、DFT はチップ上のそしてシステム内のテスト生成とテストアプリケーションの手法を提供し、それによってテスト時間と外部 ATE を使ったテストボリュームの負担を軽減しなければならない。テスト生成の専用ハードウェアと組み込みアルゴリズムは、テスト・アクセスの必要性を削減するし、一方では究極のシステムレベル DFT の要求である、システムのレベルでの BIST への基底環境を提供する。アナログ、ミックスドシグナル、RF サブシステムの DFT 技術の改善にともなって、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トゥマーケット時間短縮のために修復されるべきである。したがって DFT とシステム設計メソドロジーは、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。

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

DFT 統合への様々な障害により、チップ面積、テスト目的の I/O 配置、消費電力、バンド幅、信号感度などのシステムインパクトに関して、DFT は金がかかるという認識が広くあった。アナログ、ミックスドシグナル、RF と超高速のデジタル・サブシステムを SOC や SIP に統合することで、リーク電流や、ノイズ、ノードの負荷などの DFT によってモニターされるべき難しい課題が生じる。バンド幅の損失や付加的なノイズなどのシステムパフォーマンスへの DFT の影響は、定量化され十分に予測されなければならない。それによって、統合されるべき DFT 手法の早期の利点とコストの競合がスタディできる。DFT 手法は、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。例えば、2006 に対して 2 倍の削減要求はデジタル・サブシステムとしては控えめだが、これがアナログ RF を含む全体システムであることを考えれば妥当な要求値である。テスト時間やテストコストは、ATE と非常に強く関連している、というのは DFT や BIST があっても、なお ATE がテストでは重要な役割を持っているからである。様々な DFT 手法と ATE との間のインタフェースはきちんと定義しなければならない、それによって固有の製品のテストが要求する異なった DFT と ATE の選択と組み合わせの自由度が許容される。テストアクセスポートやその他のテスト関連の標準である 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、過去数年開発されて来ており、継続して非常に速いペースで進んでいる。SOC や SIP 中のテクノロジーや、サブシステムの設計手法の多様性は、テスト生成ツールと DFT/BIST ツールがデジタルテスト手法(故障、欠陥、パラメトリックベース)とアナログ、ミックスドシグナル、RF テスト手法(機能、パラメトリックベース、将来的には故障、欠陥ベース)が統合されることを要求している。この解決策は、システム複雑度、設計工数、タイムトゥマーケット、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分でなければならない。特に、システム性能に関わる DFT/BIST の影響を見積もることにおいてそうである。シミュレーションツールは、設計の工程が様々なレベル(動作、回路、レイアウト)を進行するにしたがって、DFT/BIST の影響の見積りを提供し、それによって、最も適当な DFT/BIST の手法が選択できるようにしなければならない。全体を通じての効果指標は、システムテストとカバレッジ(故障、パラメトリック、機能、ないしはそれらの組み合わせ)である。それは、DFT/BIST の一部として算出されるべきであり、テストの間と動作の間の両方において観測されるべきである。オンラインでカバレッジが観測できる機能は非常に強く求められており、最終的には製造品質の改善を可能にする故障の統計や欠陥の統計に関連づけができるようなカバレッジの統計のアルゴリズムが開発されなければならない。

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

製造容易化設計 (DESIGN FOR MANUFACTURABILITY (DFM))

ばらつきが増大にともない、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。

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

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

- 1) 統計的設計ツールへの入力となる、キャラクタライズとモデリング
- 2) 統計的解析から、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、時間のかかる計算結果に対して、不必要な誤差と、疑問となる値を導入してしまう。

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增大する(グラフを含んで)。ルールは 2 層構造(要求ルールと、推奨ルール)へと展開して、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、リソグラフィ装置の解像度の限界のために RET(resolution enhancement techniques: 解像度補正技術)の影響を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)でのフィリングなどはタイミングや消費電力などの回路的な指標を明示的に意識する必要がある。それにより、全体として、目的に沿ったツールの整合性を保ち、歩留りの向上を可能とし、製造コストの削減と、マスクデータ準備のための時間を改善することに繋がる。このアプローチは必然的に密接な統合フローとなり、回路的な意図を下流へ伝え、個々のツールの勝手な修正による精度劣化を防ぐことになる。結果として、レ

ジスタートランスファーレベルから GDS データ(RTL2GDS)へのフローにおいて、適切な RET や OPC の修正を下流へとつなげるようにプランニングする必要がある。たとえば、グローバル、ローカル配線層の密度は、概略配置の段階で、ダミーフィル挿入の場所を考慮する必要があり、これにより事前の CMP フィリングとその容量の見積りが可能となり、これと同時にクリティカルなネットの情報を最終レイアウトフェーズ、マスクデータ生成 (MDP)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 — ここでの基本ルールは、もはや”固定値”ではない、なぜならここでのシナリオは、設計改善よりむしろ設計の受け入れを左右するものであるためである。適切な歩留まりの成熟を得て、急峻な歩留りの立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ーハプロセスの要求、例えばアライメントの許容度、光近接補正、RET 向上やその他多くの制約との間の相互の関係から得られる。DFM としての対策は、歩留りのための面積、消費電力、そして速度のチューニングへの影響を与えるため、設計毎に異なってくる。設計過程において歩留り、面積、そして速度の相互の影響が分析され、かつ商用的な有用性のトレードオフが得られる必要がある。機能的およびパラメトリックな歩留りに対する DFM 対策は、設計の後処理(これは往々にして限定的な結果しか得られず、かつ多くの場合、処理時間がかかる)としてではなく、むしろ設計フローの中で新たな最適化機能として統合されるべきである。この意味するところは、歩留りの予測を、設計のプランニング、合成、配置、配線などのツールへ統合し、歩留り、パフォーマンス、消費電力、信号の完全性、そして(歩留りとの関連で明示的に新たな設計指標としての)面積などのすべての設計目標を考慮することを意味する。つまるところ、歩留りはプロダクト固有の設計属性と、プロセス固有の 故障の可能性の両者の関数である。そのため、ある特定のプロセスの限定的な故障/マージンのパターンを緩和するために最適化された設計は、異なるプロセス条件では事実上歩留りが低い可能性もある。そのため、配置配線ツールにおいて、異なる実装での歩留りコストの正確な評価を可能にするには、実際のターゲットプロセスでの論理ライブラリの正確な歩留りモデルを事前にキャラクタライズしておくことが必要となる。ライブラリの歩留りモデルは、プロセスの立ち上げから成熟段階にわたり頻繁に更新を行うことが必要である。

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。図 DESN8 は、これらの要求に対する DFM としての時間軸での解決策である。

Table DESN9a Design for Manufacturability Technology Requirements—Near-term Years

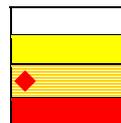
Year of Production	2007	2008	2009	2010	2011	2012	2013	2014	2015
Normalized mask cost from public and IDM data	1.0	1.3	1.7	2.3	3.0	3.9	5.1	6.6	8.7
% V _{dd} variability: % variability seen in on-chip circuits	10%	10%	10%	10%	10%	10%	10%	10%	10%
% V _{th} variability: doping variability impact on V _{th} , (minimum size devices, memory)	31%	35%	40%	40%	40%	58%	58%	81%	81%
% V _{th} variability: includes all sources	33%	37%	42%	42%	42%	58%	58%	81%	81%
% V _{th} variability: typical size logic devices, all sources	16%	18%	20%	20%	20%	26%	26%	36%	36%
% CD variability	12%	12%	12%	12%	12%	12%	12%	12%	12%
% circuit performance variability circuit comprising gates and wires	46%	48%	49%	51%	60%	63%	63%	63%	63%
% circuit total power variability circuit comprising gates and wires	56%	57%	63%	68%	72%	76%	80%	84%	88%
% circuit leakage power variability circuit comprising gates and wires	124%	143%	186%	229%	255%	281%	287%	294%	331%

Table DESN9b Design for Manufacturability Technology Requirements—Long-term Years

Year of Production	2016	2017	2018	2019	2020	2021	2022
Normalized mask cost from public and IDM data	11.4	14.9	19.6	25.6	33.6	44.2	57.7
% V _{dd} variability: % variability seen in on-chip circuits	10%	10%	10%	10%	10%	10%	10%
% V _{th} variability: doping variability impact on V _{th} , (minimum size devices, memory)	81%	81%	112%	112%	112%	112%	112%
% V _{th} variability: includes all sources	81%	81%	112%	112%	112%	112%	112%
% V _{th} variability: typical size logic devices, all sources	36%	36%	50%	50%	50%	50%	50%
% CD variability	12%	12%	12%	12%	12%	12%	12%
% circuit performance variability circuit comprising gates and wires	63%	65%	66%	69%	69%	71%	73%

% circuit total power variability circuit comprising gates and wires	92%	96%	102%	110%	121%	130%	140%
% circuit leakage power variability circuit comprising gates and wires	368%	381%	395%	360%	325%	477%	628%

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



DFM としての要求は、次に掲げるカテゴリに分類される。

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

マスクコストなどの経済上の要因は、直接的に制御するのが困難であり、増大して行くことが予測される。それに変わる次善策として、マルチプロジェクトウェーハ、コンフィギャラブルロジックやストラクチャード ASIC が提供される。ばらつきに関して、CD コントロールのような”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。

DFM の技術的要求こと項に対処するには、図 DESN8 に記載されている DFM の解決策が必要となる。

- ・ **基本的な経済的制限に対応した解決策** — マスク費用以外を含んだ経済的要因を明らかにするために、主なアルゴリズムやインタフェースにおいて、DFM ツールが必要になる。チームやマネージャは、難しい DFM のトレードオフの経済的価値について、より直接的に評価できるであろう。
- ・ **ばらつきの影響に対する解決策** — 将来 DFM はばらつきの様々な特徴を取り扱うようにならなければならない。第 1 に、性能と消費電力の両方のばらつきに対応しなければならない。その結果、統計的機能解析、有効電力、漏れ電力解析ツールが大変重要となる。第 2 に、システム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。

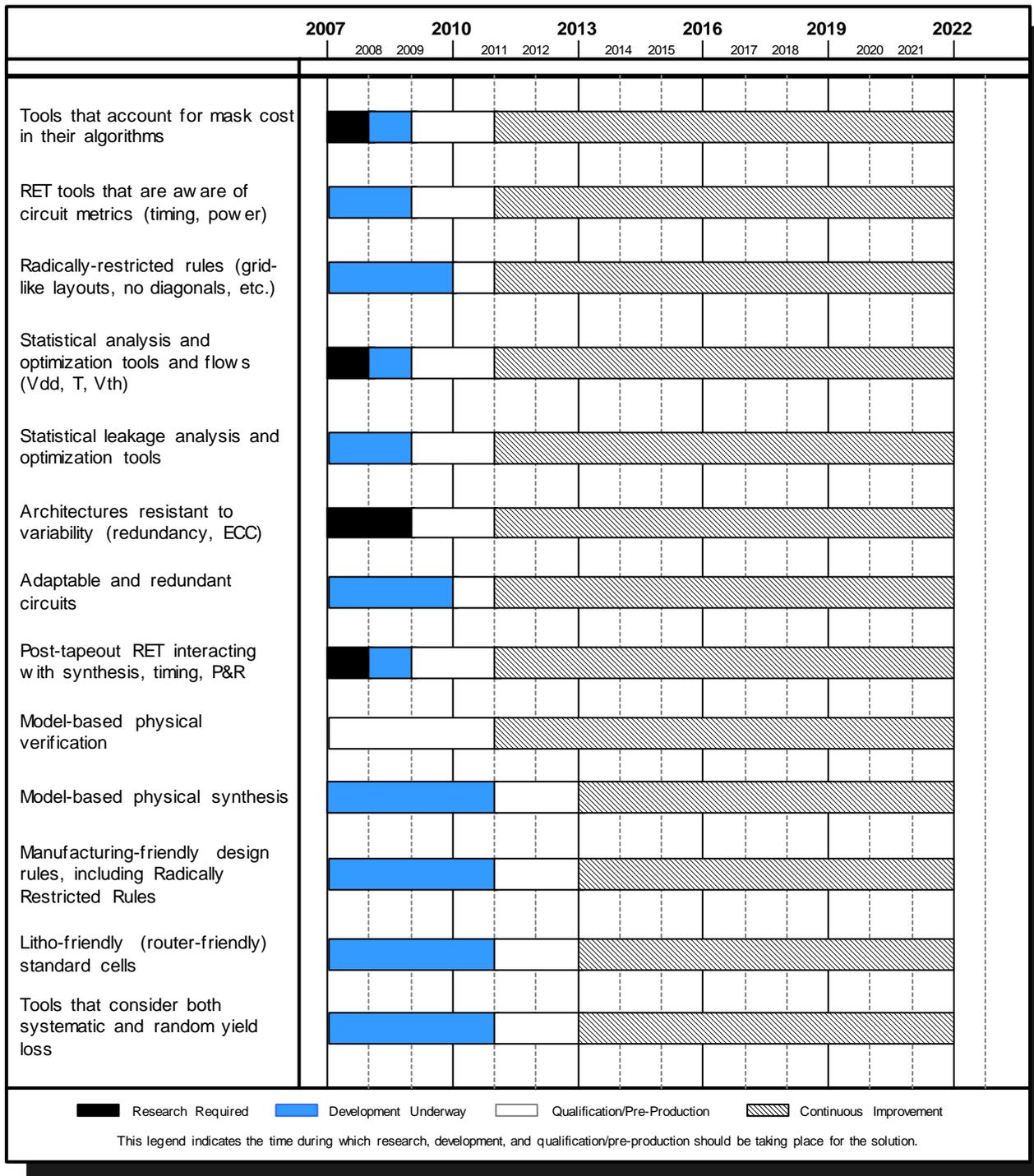


Figure DESN8 Design for Manufacturability Potential Solutions

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、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DFMの解決策の間の対応関係を表したものである。

Table DESN10 Correspondence Between Design for Manufacturability Requirements and Solutions

<i>Requirement</i>	<i>Solution</i>	<i>Explanation of the Correspondence</i>
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)	Router-friendly standard cells	Routing-friendly rules reduce design, mask, and manufacturing complexity
	Adaptable and redundant circuits	Inherent circuit robustness to variability
	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

多様化(MORE THAN MOORE)の分析

設計技術の改善に対する、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 DESN9 に示す。

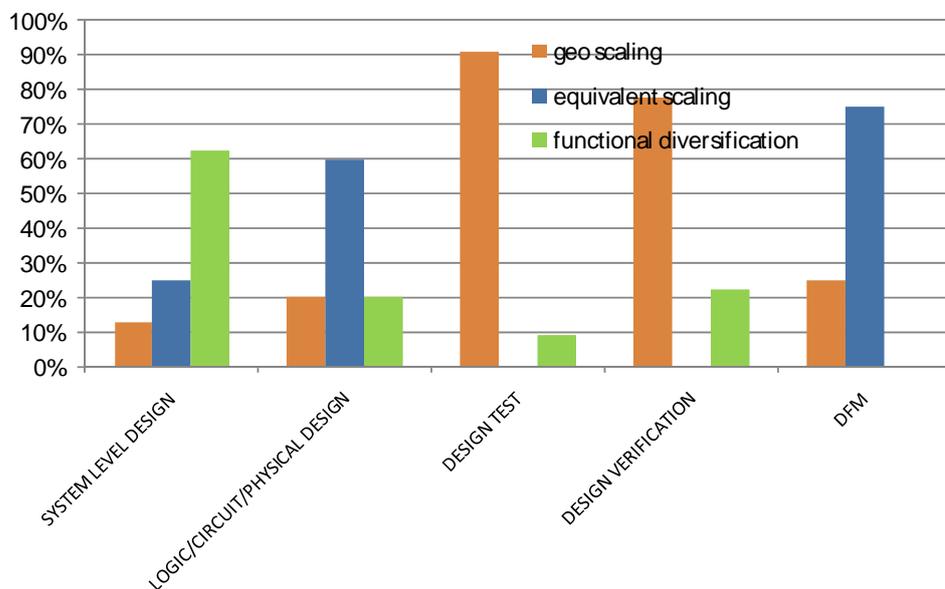


Figure DESN9 Moore and Non-Moore Design Technology Improvements

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と、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。デジタル回路で使用されるツールは、アナログ設計においては適切なものではなく、非常に不正確であった。歴史的に見て、チップ当たりのアナログおよびRF用トランジスタ数の増加はデジタル用ト

ランジスタ数の増加と比較して非常に緩やかであるため、アナログと RF 設計者をシステムレベル設計ツールから遠ざけてきた。回路とシステム設計については、この章の後半の「論理、回路、物理設計」節で述べられる。今日、アナログと RF 設計において知られていたものと同様の課題が、デジタル設計(IRドロップ、クロストークなど)においても発生し始めている。SOC デジタル設計では、アナログおよび RF 回路とチップ領域を共有する。常に短縮化される time-to-market とより高い生産性へのニーズは、近年において歴史的なパラダイムを変えてきた。そして今日、EDA 業界によるアナログと RF に特化した新しい取り組みが現れている。

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

1GHz以上の周波数ではインダクタ、信号配線、伝送路、電子的な同調容量などの分布定数素子の正確な計算が不可欠である。これらの計算は信号の表皮効果と、波形歪につながる周波数依存の位相速度を考慮に入れて、マクロモデルを構築する必要がある。このような計算の重要性といくつかの興味深い結果が、標準的な 130nm CMOS プロセスを用いた 43GHz VCO(電圧制御発振器)の設計例として 2005 年の論文で報告されている。この論文では配線幅 5 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 FDTD (finite difference time domain) 法を用いて設計しなければならないが、そのアンテナが信号配線間の寄生効果で形成される場合は、膨大な数となるため計算精度を少し低下させたとしてもより単純化した手法で算出する必要がある。このような高周波領域での設計問題は、定在波・進行波発信回路を用いた 1GHz以上のデジタルクロック信号生成のための高性能アナログ回路を使う場合にも発生し、解決されている。

マクロモデル構築には上述の作用のほとんどすべてが影響を与える。信号配線間の容量性および磁気カップリング、パッケージングされたチップの熱特性、アンテナの近距離および遠距離寄生効果、内蔵された MEMS 動作特性などは 2D または 3D の電磁界解析手法を用いて、10 の 4 乗から 8 乗の数の方程式で表現される。計算精度の大幅な低下を招くこと無しに、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)のニーズに対処するようになる。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-光学、電子-生物学などの素子が単なる変換器を越えるものとなるにつれ、これらの異種部品を集積したシステムをモデル化し、分析し、検証することが今後の課題となる。カーボン・ナノチューブ・トランジスタ、単電子トランジスタ、共鳴トンネル・ダイオードのような、複雑な物理効果を利用した非従来型デバイスで構成された回路においても同様である。

表 DESIGN11 に 2011 年までに予想される短期的な AMS 設計技術のブレイクスルーを示す。あわせて、MEDEA+Design Automation Roadmap における、AMS 設計技術の要件に関する秀逸な議論も参照されたい。

Table DESN11 Near-term Breakthroughs in Design Technology for AMS

Field of Breakthrough	2007 State-of-the-Art	2008/09	2010/11
Specification, validation, verification	Established AMS Hardware Description Languages	Multi-language support, AMS extension of HW/SW description languages for full system simulation	Complete specification-driven design flow; some specialized formal verification methods
Architectural design	Algorithm-oriented design (e.g., with Matlab/Simulink)	Language-based performance evaluation; closer coupling of architectural, block, and circuit level	Synthesizeable AMS description; power-aware HW/SW partitioning extended to AMS systems
Physical mixed A/D and RF design	Procedural layout generation, module generators for a few block types	Module generators for often re-used blocks, design centering, performance estimation	Synthesis: behavior to layout (at least for the most important building blocks)
Parasitics extraction, automated modeling, accelerated simulation	Electromagnetic immunity simulation works but is too complicated for broad usage	2D/3D model-based order reduction for interconnect systems and substrate effects on chip, thermal package modeling	New fault-tolerant circuit architectures, robustness against technology parameter variations; order reduction for all kinds of parasitics and antennas

最後に、テスト設計に関しては、アナログ回路はミックスドシグナル SOC の全チップ領域のごく一部であるにもかかわらず、量産テストコストを支配する。アナログテストの変革がなされない限り、ミックスドシグナル製品のテ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カスケード型変調回路(mash 回路)で構成される。このためひとつの可能性ある方向性は、DFT またはセルフテストの戦略を開発するにあたって、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
- 回路図の等価性検証
- 製造容易化設計
- アナログ/RF レイアウト合成
- 容量性、誘導性、熱的な寄生効果の抽出、自動モデル生成、およびシミュレーション高速化
- アナログ IP の回路情報要約化と再利用

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。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。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 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マティックな方法が求められているし、あるいは、また、ばらつきの望ましいトレンドが設計技術全体のロードマップの重要な部分になる。それは、業界がばらつきを低減すべきか、ないしは、設計生産性を改善するかの提言を集成していくための、設計と製造の“共同(共通)ロードマップ”を可能にして行く。このようなばらつきのフレームワークへの要求は、業界関係者へのばらつき関連の情報への敏感さによってはっきりと示されている。設計のコミュニティでは、ばらつきを設計のパラメータの観点から見て行く必要があるので、ばらつきのフレームワークは、マルチレベルである必要がある。すなわち、設計の複数の抽象化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 26 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 つの抽象化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抽象化のレベルである。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ーフェースに最も密接なレベルである。このレベルでは、CD や実効デバイス長(Le)、実際のドーピングレベル(Na)などのパラメータがロードマップ化される。これらのパラメータのばらつきは、リソグラフィ装置の解像度の限界、チャンネル内のドーパ量の正確な数値の制御ができないことなどを含む、上述の課題によって生じる。

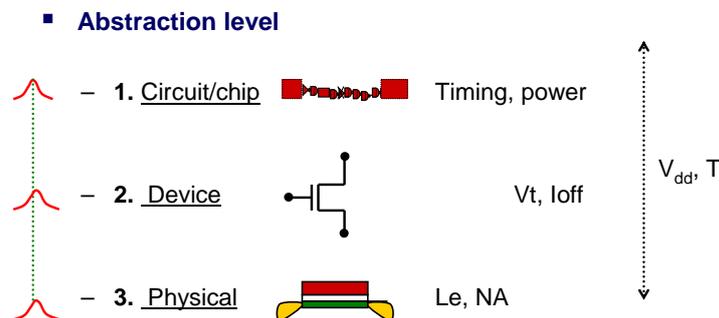


Figure DESN10 Possible Variability Abstraction Levels

このフレームワークでは、モデリングプロセスは以下の式で表わされる。

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

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

他の適用の中では、このフレームワークは回路性能のばらつきトレンドにも使用できるだろう。例えば、予測遅延ばらつきは、2 つのシナリオで見積もられる: 10%の要求 CD(実際のチャンネル長)ばらつき、ないしは 20%の要求 CD ばらつきである。この例では、モデルは、回路性能ばらつきが CDトレラント要求に対して著

しく変わるようには見えないことを指し示している。つまり、要求が緩和できる可能性を示している。

付録 II: DT コストと価値

図 DESN11 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。不幸にも、systems-on-a-chip(SOC)の絶えず増加している複雑さは、設計コストとチップ単価をコントロール不能にしている。ますます競争が激しくなる環境と結合されてコストが上昇する中で、収益の確保が困難になってきている。これは、製品開発費が製品のライフサイクルの中で先行投資の状態になり、一方、売上げのかなりの部分が 1 年以上遅れで発生するという事実によって悪化させられる(キャッシュフローの悪化)。次の分析は、継続的な設計技術革新のパイプラインがなければ、設計コスト(したがって、製品開発費も)をかけることが急速に不可能な状態になるであろう、さもなければ、設計自体があまり価値のないものにさせられるであろうことを示唆する。

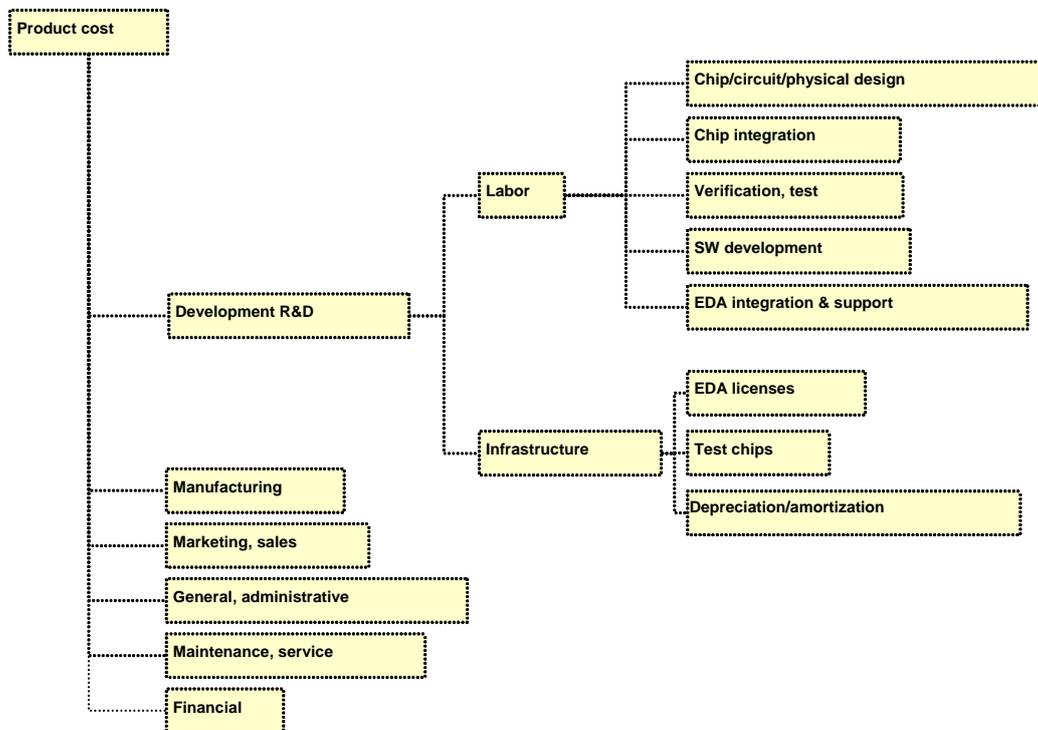


Figure DESN11 Simplified Electronic Product Development Cost Model

図 DESIGN 11 において、ボー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、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(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 メソロジー」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 12 に示されており、その中でのグレーの項目は、現在進行中、もしくは、将来の DT 革新項目を表している。その表から分かるように、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2007年までの間に、年率で平均39.6%増加したことになる。特に、百万ゲートを設計する設計者数(生産性の逆であるが)は、1990年の250人から2005年には8人に削減された。しかし、労働単位コストは、1990年以来一定の値を維持しなかった。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。この仮定は、設計のグローバル化の傾向を考慮して、次のロードマップ改訂では再検討されるかもしれない。上昇率は2002年から2005年にかけて年率2%に下がったが、中国やインドからできえ報告されるIC設計者の不足により、再度年率5%に戻っている。

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6年間は過去の歴史的な主準を下回っている。業界は今このスランプから回復しつつあり、2008年には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×年人で求められる:

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

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 DESM1 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、2007年で17M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、\$15.5Mで、さらにソフトウェア設計コストが\$24.0Mかかる。2007年で初めて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る。1993年から2005年の間に実現した6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、およそ\$900Mになっていた。

Table DESN12 Design Technology Improvements and Impact on Designer Productivity

<i>DT Improvement</i>	<i>Year</i>	<i>Productivity Delta</i>	<i>Productivity (Gates/Design-Year)</i>	<i>Cost of Component Affected</i>	<i>Description of Improvement</i>
None	1990		4K		
In-house place and route	1993	+38.9%	5.55K	PD Integration	Automated block placement and routing
Engineer	1995	+63.6%	9.09K	Chip/circuit/PD Verification	Engineer can pursue all required tasks to complete a design block, from RTL to GDSII
Reuse—small blocks	1997	+340%	40K	Circuit/PD Verification	Blocks from 2,500–74,999 gates
Reuse—large blocks	1999	+38.9%	56K	Chip/circuit/PD Integration Verification	Blocks from 75,000–1M gates
IC implementation suite	2001	+63.6%	91K	Chip/circuit/PD Integration EDA support	Tightly integrated tool set that goes from RTL synthesis to GDSII through IC place and route
RTL functional verification tool suite	2003	+37.5%	125K	SW development Verification	RTL verification tool (“cockpit”) that takes an ES-level description and partitions it into verifiable blocks, then executes verification tools on the blocks, while tracking and reporting code coverage
Transactional Modeling	2005	+60%	200K	SW development Verification	Level above RTL, including both HW and SW design; it consists of a behavioral (where the system function has not been partitioned) and an architectural level (where HW and SW are identified and handed off to design teams)
Very large block reuse	2007	+200%	600K	Chip/circuit/PD Verification	Blocks >1M gates; intellectual-property cores
Homogeneous parallel processing	2009	+100% HW +100% SW	1200K	Chip/circuit/PD Design and Verification	Many identical cores provide specialized processing around a main processor, which allows for performance, power efficiency, and high reuse
Intelligent test bench	2011	37.5%	1650K	Chip/circuit/PD Verification	Like RTL verification tool suite, but also with automation of the Verification Partitioning step
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 +100% SW	3300K	System Electronic Design and Verification	Each of the specialized cores around the main processor is not identical from the programming and implementation standpoint
Transactional Memory	2017	+100% HW +100% SW	6600K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 1)
System-level DA	2019	60% HW 38% SW	10557K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 2)
Executable specification	2021	200% HW +200% SW	31671K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 3)
Total		+264,000%			