

# 21世紀のシリコンプラットホーム アーキテクチャ設計技術

村上 道郎 富沢 方之 灘岡 満  
宇都宮 雅彦 御手洗 睦

LSIプロセス技術の微細化に連動する、LSI集積度の向上は留まるところを知らず、継続的に年率58%も向上している。LSI設計技術も設計生産性を大幅に向上させているが、これまでのところ年率21%程度に留まっている<sup>1)</sup>。

当社は80年代はじめからこの状況を‘Design Crisis’と標榜し、設計生産性向上を推進してきた。80年代のエンジニアリングワークステーションによる回路図作成の機械化、市販高速論理シミュレータによるTAT短縮とユーザ設計環境との一体化、90年代の論理合成によるゲート回路作成の自動化、高精度タイミング検証技術によるファーストシリコンでの正常動作等である。

しかし、このままではLSI集積度と設計生産性の乖離が甚だしくなり、2005年には10億トランジスタがチップに搭載可能となるが、その設計には400人年を要するようになる<sup>1)</sup>。本稿では、この設計の危機を乗り越えるために現

在開発中の、また今後開発を予定している、21世紀に利用可能な設計技術についてまとめた。まず、システムレベル設計技術について述べ、次に大幅な設計期間短縮を可能にするプラットホーム設計技術について述べる。さらに、これらを支えするIP (Intellectual Property) ベース設計技術、LSI化設計技術、低消費電力化技術について述べる。

## システムレベル設計技術

LSI集積度の向上により、一昔前のシステムが1チップに収まるようになってきた。LSIの設計は単なる部分回路の設計ではなく、システムの設計そのものになりつつあり、システムレベルの設計/検証手段が必要となってきた。このような状況を解決するために、設計の抽象度を上げ、システムレベルで設計する必要がある。シス

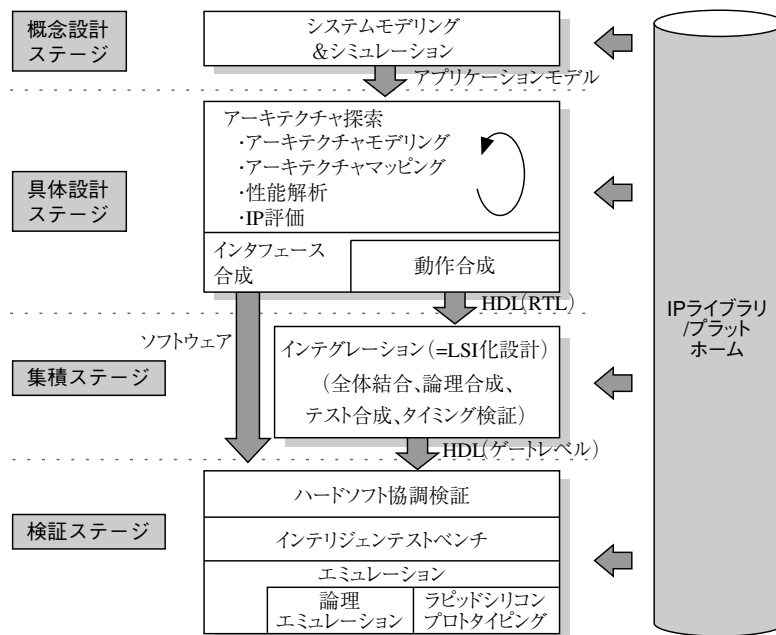


図1 システムレベル設計手法

テムレベル設計技術は、図1に示すように、大きく4つの設計ステージから構成される。以下に各ステージの概要を説明する。

#### (1) 概念設計ステージ

ターゲットとするアプリケーション・システムの機能をモデリングし、シミュレーションにより機能を検証するとともに、システムのボトルネック解析を行う。モデリングには、Cベースのシステム記述言語に加え、ブロック図、状態遷移図等、多彩なデザイン・エントリ手段を用いる。本ステージで構築されたアプリケーションモデルは、次工程の具体設計ステージでも使われる。

#### (2) 具体設計ステージ

要求機能を実現するためのインプリメンテーションの検討と決定を行うことを目的としたステージ。概念設計ステージで確定した機能を実現するためのアーキテクチャモデルを構築し、アプリケーションモデルをアーキテクチャモデルにマッピングし、性能解析で裏付けを取りながら、コスト・設計期間・柔軟性などの面で最適なハード・ソフト構成を検討・決定する。このように、従来設計者の経験と勘で行われてきたアーキテクチャ検討が、シミュレーションにより定量的に評価しながら、可能となるので、設計生産性のみならず、設計品質の向上も期待できる。また、後工程のためにインタフェース合成や動作合成を行い、HDL (Hardware Description Language) モデル、Cソースプログラムを出力する。

#### (3) 集積ステージ

具体設計ステージで決定された構造を基に、LSIとして実現することを目的としたステージ。HDLベースの記述

から詳細な論理の検証を行い、性能・サイズを満たすゲートレベル回路データを得ることがゴールとなる。プラットフォームホームベース設計の場合、プラットフォームを核としてインテグレーションをある程度自動化することが考えられ、IPコンポーネント結合、シミュレーションモデル作成、論理合成、テスト合成などの作業をサポートする支援ツールを使用する。

#### (4) 検証ステージ

ゲートレベルで論理の正当性を確認するが、単に正常動作することのみならず、異常入力時の動作の確保や極めてまれなケースの動作の保証もしなければならない。また、上位ステージに比べ大幅にシミュレーション時間が増加するにも関わらず、論理の正当性を確認するために、現実的な時間の中で検証作業を完了させる手段が必須となる。さらに、ソフトウェア開発およびデバッグの比重も増大する。これらの課題を解決するために、インテリジェントテストベンチ、ハード・ソフト協調検証、エミュレーションなどの手法を適切に使用する。

### プラットフォームベース設計技術

設計生産性向上のための手段として、既存設計資産を流用するアプローチがある。次節で詳述される、設計済みの機能ブロックをIPとして流用する、IPベース設計である。さらに流用度を高め、システムLSIのアーキテクチャ(複数IPの接続形態)まで流用するのがプラットフォームベース設計である。標準となるアーキテクチャを事前に定義・検証しておき、これに製品独自機能を追加/修正することにより、派生品LSIの開発期間を大幅に短縮する。

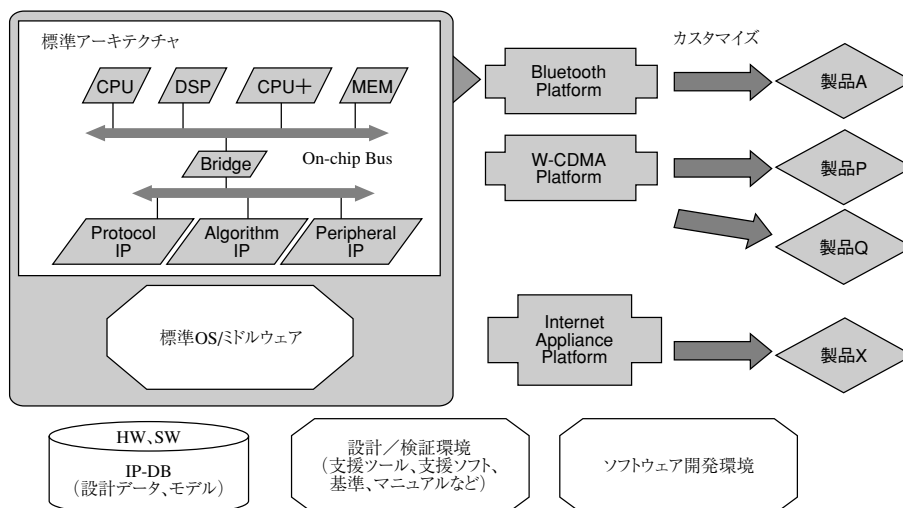


図2 プラットフォームベース設計

プラットフォームの構成要素として、システムLSIの標準アーキテクチャの他、OS (Operating System)、ミドルウェア、設計検証環境も含み、ソフトウェアや開発環境の流用・活用も行う (図2)。

当社におけるプラットフォームベース設計技術の基本概念がSPA (Silicon Platform Architecture) である。SPAは、後述する $\mu$ PLATを基盤に置き、SoC (System on Chip) の開発期間の大幅な短縮を可能にする。

#### (1) $\mu$ PLAT

$\mu$  PLATは、ARMプロセッサ、および基本的な周辺IPからなる標準アーキテクチャであり、DSM (Deep SubMicron) における動作周波数を確保するために、ハードIPとして提供される<sup>2)</sup>。さらに、標準アーキテクチャ上で動作するリアルタイムOS等の基本ソフト、ユーザIPやソフト開発用のプロトタイピング基板、ユーザIP開発用テストベンチ、コンパイラ・デバッカ等のソフトウェア開発環境も提供される<sup>3)</sup>。

#### (2) アプリケーションスペシフィックプラットフォーム

$\mu$  PLATは、このように様々なアプリケーションで共通に使う基本IPを集積した汎用プラットフォームである。さらに回路の流用範囲を広め、特定アプリケーション分野向けのIPまで標準アーキテクチャに取り込めば、派生品LSIの設計生産性が一層向上する。また、このようなアプリケーション・スペシフィックなプラットフォームは、システムレベル設計技術と結びつけば、理想的な設計環境を構築できる。例えば、システムレベル設計の概念設計ステージでは、ターゲットとなるアプリケーション・システムの機能をモデル化するが、アプリケーションに

必要なアルゴリズムやプロトコル等を一から準備するのは大変である。必要なアルゴリズム、プロトコル、およびそれらを組み込んだ典型的なアプリケーションシステムの機能モデルをプラットフォームとして準備しておけば、容易にその派生システムの機能モデルを構築できる。また、具体設計ステージ以降の各ステージについても、各ステージに応じた標準アーキテクチャやオプションIPなどのモデルを事前に準備しておくことにより、システムのモデリングの手間を大幅に省くことができ、設計生産性を大きく向上させることができる。

### IPベース設計技術

前述のシステムレベル設計技術とプラットフォームベース設計技術を支える、IPベース設計技術とIP関連技術について述べる。

#### (1) IPベース設計技術

IPベース設計技術とは、LSIの構成要素として再利用容易化を行った設計・検証済みIPの各種モデルを用意し、そのIPを組み合わせて設計する事により、設計期間短縮、設計生産性向上を実現する技術である。図3に開発ステージごとに提供されるIPモデルを示す。

今後システムレベル設計環境の整備計画に同期し、必要とされるIPモデルを順次取り揃えていく予定である。

#### (2) IP関連技術

IP関連技術を、IP再利用容易化技術、IP認証技術の順に説明する。

IP再利用容易化技術の要素技術は、設計ガイドライン、IPパッケージ技術、テクノロジー変換技術である。設計ガ

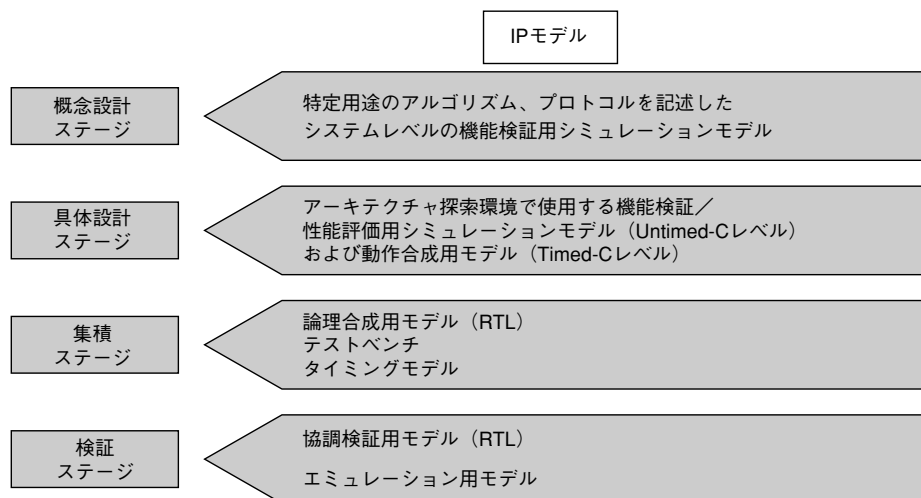


図3 開発ステージごとに提供されるIPモデル

イドラインは、IP開発者が従うべき設計規約を制定しており、次のルール群で構成される。命名ルール、回路設計ルール、バスインタフェースルール、RTL (Register Transfer Level) 記述ルール、論理合成ルール、テスト容易化ルール、レイアウトルール、配布物ルール、およびドキュメントルールである。今後、システムレベル設計用ルールを追加していくほか、業界で認知された標準ルールが制定され次第、順次取り込み、所有IPの社外発信および社外IP導入の効率化を図っていく。

IPパッケージ技術は、ソフトIPユーザが論理合成した際に発生するタイミング等のエラーをなくすよう、合成制約等もIPデータの一部として盛り込む技術である。IPデータの暗号化技術も含む。

テクノロジ変換技術は、カスタム設計されたハードIPを異なるテクノロジ向けに自動変換する技術である。より微細なプロセステクノロジへの変換が代表例である。変換後のパフォーマンスや消費電力の最適化をいかに効果的に行えるかがキーポイントである。

IP認証技術は、IPの品質を第三者が客観的・数値的に評価可能にする技術である。数値化されたIP品質は、IPユーザがIPを選択する際の重要な目安となる。IP品質の客観的評価のため、IPデータが従うべきルールをIP認証基準として制定し、その準拠度を測定する。IP認証基準のルール項目はIP設計ガイドラインと同等であるが、IPデータのみからその準拠度が判定できるルール項目に絞り込んである。例えば、命名ルールやRTL記述ルールである。準拠度の測定は、レビュー形式でSpread Sheetに判定結果を記入し、数値化される。RTLスタイルチェッカなどによる自動化がキー技術である。

## LSI化設計技術

21世紀になるとSoCの回路規模は10メガゲートを超え、動作速度は500メガヘルツを超えることが予測される。このようなSoCを短期間で確実に商品化するためには、高度なEDAツールによる設計・検証技術の確立が必須となる。特に、後戻りの無い設計技術と、設計データの完全性を追求した検証技術がキーとなる。図4にSoC設計検証フローを示し、以下に解説する。

### (1) 設計技術

SoCの回路は複数の回路ブロックで構成される。ハイレベルフロアプランは、タイミング収束を短期間で完了させるために、ブロック間の配線を制御しながら、チップのタイミング仕様に基づいて、各ブロックのタイミング仕様を作成する。ここで、ブロックの配置、ブロック間の配線が決定される。チップレベルのタイミング仕様を元に、タイミング、エリア、レイアウト、パワー、テストを考慮した論理合成を行い、各ブロックのゲート回路が生成される。この時点で、チップレベルのタイミング制約を考慮して、ブロック内のゲート等素子の配置が決定される。

従来、ゲート回路生成とレイアウト設計は独立した設計工程であったが、微細化によって配線のタイミングに与える影響が増大しているため、回路設計段階でその影響を考慮しておくことが必要となる。即ち、フロアプランでのブロック配置の決定や、論理合成時のブロック内のゲート等素子の配置の決定がそれぞれである。したがって、タイミングドリブンレイアウトでは、ブロックレベルのタイミング仕様に合せて、主にブロック内の配線のみを実行すれば良く、チップに与えられたタイミング制約を

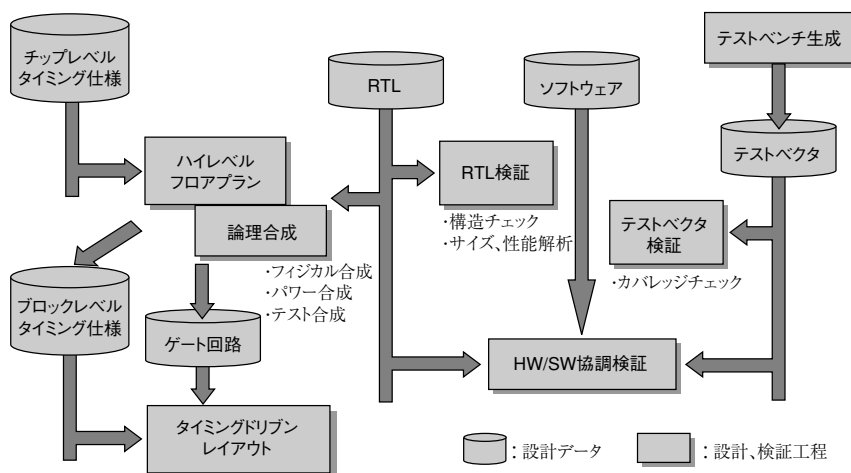


図4 SoC設計検証フロー

守ったLSIが容易に実現できる<sup>4)</sup>。

プロセス技術の微細化は、レイアウト工程にも重要な機能を要求している。配線抵抗の増加は、低電圧化でより顕著となる電圧降下に起因する誤動作への考慮と、タイミングへの影響の考慮を必要とし、また隣接配線容量の増加は、カップリングノイズによるタイミングへの影響の考慮を必要とする。

## (2) 検証技術

回路機能定義データであるRTLは、RTL検証ツールによってその構造の確かさがチェックされるのみでなく、サイズ、性能（タイミング、パワー）の予測までもが行われる。構造チェックは、業界標準をベースに数百項目も実施され、RTLの品質を飛躍的に向上させる事ができ、再利用性の高い回路設計としても重要となる。さらに、従来、論理合成を待たなければ出来なかったサイズ、性能の予測が、早期にしかも高速に行われるため、設計TATの削減が図れる。

## 低消費電力化技術

プロセス技術の進展によりLSIの高集積化が進み、携帯電話等バッテリー電源を持つ携帯機器に使われるLSIの低消費電力化が課題となっている。これらのLSIは高速かつ低消費電力が要求される。一般的にCMOSデジタル回路では、動作時消費電力は負荷容量、電源電圧の2乗、動作周波数の積に比例し、待機時消費電力はリーク電流が支配項となる。回路の遅延時間は、トランジスタにかかる電源電圧に比例し、かつ実効的なゲート電圧に反比例する。消費電力を小さくするためには電源電圧を下げればよいが、それにより実効的なゲート電圧が小さくなり遅延時間は大きくなる。遅延時間を維持するためにはしきい値電圧を低くし、実効的なゲート電圧を大きくする必要がある。したがって、高速性と低消費電力を両立させるためには、電源電圧を下げ、かつしきい値電圧を下げればよい。しかし、リーク電流は、しきい値電圧を低くすると増加してしまう。そこで、リーク電流を動的に管理する、つまり、待機時はリーク電流を抑え、動作時は高速性を優先する方法が考えられる。そのような回路技術として、MT-CMOS (Multi-Threshold-voltage CMOS)、VTCMOS (Variable Threshold-voltage CMOS) 等がある。MT-CMOSは、低しきい値トランジスタで構成した回路と電源線間に低リーク電流の高しきい値トランジスタで構成したトランジスタ（スイッチ）を挿入した回路形式である。待機時にはこのスイッチを遮断してリーク電流を抑える。VTCMOSは基板バイアス効果でしきい値電圧を制御する回路形式である。VTCMOS

はLSIのテスト方法のひとつであるIDDQテストが可能などメリットがあるが、しきい値電圧制御回路が複雑で設計が難しい面がある。

当社ではSOI (Silicon On Insulator) デバイスの開発を進めている<sup>5)</sup>。SOIデバイスはトランジスタが絶縁層で囲まれており、基板バイアス制御が困難であること、通常のバルクCMOSにも適用できることを考慮し、回路形式が簡単なMT-CMOSを採用した。

今回MT-CMOS回路の特性を確認するために、0.2 $\mu$ mSOIデバイスを用いてTEG (Test Element Group) の試作を行い、その評価結果を得たので紹介する。

設計した回路は遅延評価TEG、およびSRAMである。遅延評価TEGには、高しきい値トランジスタで構成したインバータチェーン、低しきい値トランジスタで構成し

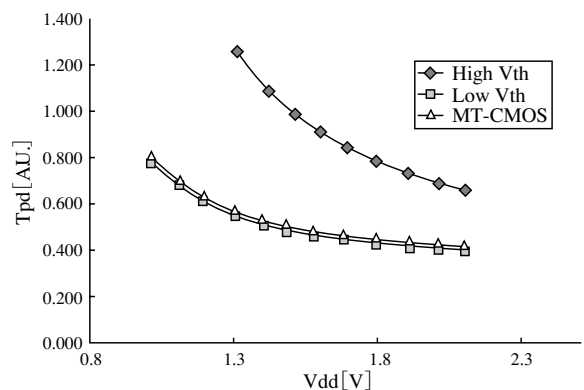


図5 遅延時間

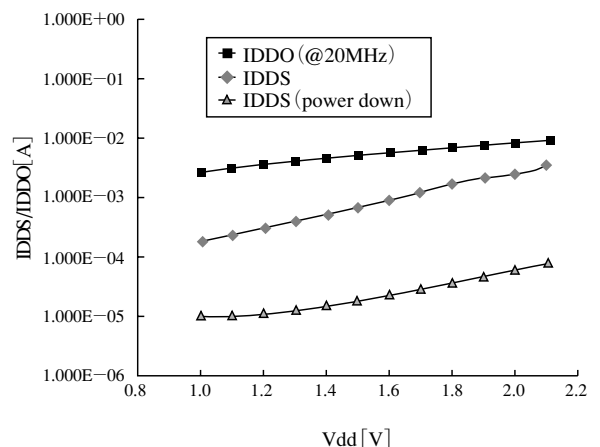


図6 SRAM消費電流

たインバータチェーン，MT-CMOS構成にしたインバータチェーンを搭載した。図5に遅延時間の電源電圧依存性を示す。

低しきい値トランジスタを採用することによりVdd=1.5Vで約2倍高速になる。MT-CMOS構成にした場合，遅延時間は低しきい値トランジスタで構成したインバータとほぼ同じである。

SRAMは8Kword×32bit構成とし，メモリセルは高しきい値トランジスタを用い，周辺回路である行デコーダ，カラムデコーダ，入出力回路はMT-CMOS回路構成にした。メモリセルはデータを保持するため待機時でも電源から切断できない。図6に消費電流の電源電圧依存性のグラフを示す。MT-CMOS構成での待機時の電源電流は，パワーダウンモードにすることにより約40分の1に抑えられた。

しきい値電圧を低く設定し高速性を維持しつつ，電源電圧を低くすることにより消費電力を低減し，かつ待機時のリーク電流についても低減できることを示した。また，他の低電力化の方法に負荷容量の低減がある。一般的には微細化技術，多層配線技術などプロセス技術によるところが大きい，設計面ではフロアプランの最適化による配線長の短縮，トランジスタサイズの最適化，ゲーティッドクロックなどのクロック分配技術がある。システムレベルでのパワーマネジメント方法，特にソフトウェアによる動的な管理も重要となる。低消費電力技術はSoCを実現するための総合的な技術であり，各設計ステージで展開が期待される。

## ま と め

21世紀に利用可能になる設計技術を紹介した。最近その兆しが見えているように，SoCはIPを多用したプラットフォーム型が主流になり，Siベンダ側の設計開始も従来のRTLから，システムレベルへと移行する。このような急激な技術革新に沿って，最先端の設計技術を開発・提供し，短設計期間，低消費電力等の，お客様のニーズに答えていきたい。◆◆

## 参考文献

- 1) 日経エレクトロニクス，No. 701，1997
- 2) 来住他：μPLATのハードウェア開発，沖研究開発 第184号，Vol.67，No.3，2000
- 3) 高田他：ソフトウェアSPA，沖研究開発 第184号，Vol.67，No.3，2000
- 4) 野村他：メタルプログラマブルSPA設計環境，沖研究開発 第184号，Vol.67，No.3，2000

- 5) 松橋他：エレベーターソース・ドレインを用いた0.15μmゲート長SOI CMOSトランジスタの開発，沖研究開発 第184号，Vol.67，No.3，2000

## ● 筆者紹介

- 村上道郎：Michio Murakami.シリコンソリューションカンパニー LSI事業部 設計システム部  
 富沢方之：Masayuki Tomisawa.シリコンソリューションカンパニー LSI事業部 プラットホーム開発部  
 灘岡 満：Mitsuru Nadaoka.シリコンソリューションカンパニー LSI事業部 設計システム部  
 宇都宮雅彦：Masahiko Utsunomiya.シリコンソリューションカンパニー LSI事業部 設計システム部  
 御手洗睦：Mutsumi Mitarashi.シリコンソリューションカンパニー LSI事業部 先端商品開発第一部