

# 第 1 章

## ミックスド・シグナルLSI と その回路設計

大規模システムLSI技術の急速な進歩によって、従来、個別のLSIで構成されていたシステムが、一つのシステムLSIで実現できるようになってきました。システムLSIとは、数百万ゲート以上の規模をもつ大規模なLSIのことで、演算回路、論理回路、CPUなどのデジタル回路だけでなく、プログラムROMやRAMなどのメモリ、場合によっては、A-D変換、D-A変換、OPアンプなどのアナログ回路を1チップ上に搭載しています。市場からの小型軽量化や高性能化の要求が高まるにしたがって、デジタルだけでなくアナログ回路を混載した「ミックスド・シグナルLSI」の必要性が高まっています。

ミックスド・シグナルLSIの設計には、アナログ設計とデジタル設計が混在した「ミックスド・シグナル設計(Mixed Signal Design)」が必要となります。従来、このような設計には、アナログ回路とデジタル回路をそれぞれ個別に設計し、最後にレイアウト上で統合するような設計手法が一般的でした。しかし、アナログ回路とデジタル回路の接続に起因した機能やタイミングの検証が不十分である場合、両者を接続した際に不具合を生じ、LSIが動作しない危険性があります。そのため、設計初期段階からミックスド・シグナル設計環境を用いて、アナログ回路とデジタル回路の混在による機能を検証し、設計品質を向上させることが重要になっています。

このような状況の中、アナログ回路およびミックスド・シグナル回路の記述と検証、および設計生産性の向上を目指して、“Verilog-AMS(Analog and Mixed Signal)”ハードウェア記述言語が標準化されました。Verilog-AMSは、IEEE1364で標準化されている“Verilog-HDL”を基本とし、さらに、アナログ記述言語“Verilog-A(Analog)”と、アナログ回路とデジタル回路の容易な接続と混在を可能とする拡張構文で構成された言語です。Verilog-AMSを用いると、上位レベルでのアナログ回路の検証、デジタル部とアナログ部の混在検証が実行でき、その結果、設計品質の向上と設計期間の短縮が期待できます。

本章では、ミックスド・シグナルLSIの構成と設計手法、ハードウェア記述言語Verilog-AMSの概要や歴史的な背景、さらにVerilog-AMSを用いたアナログ回路とミックスド・シグナル回路のモデリングや設計によって得られる意義やメリットを紹介します。

見本

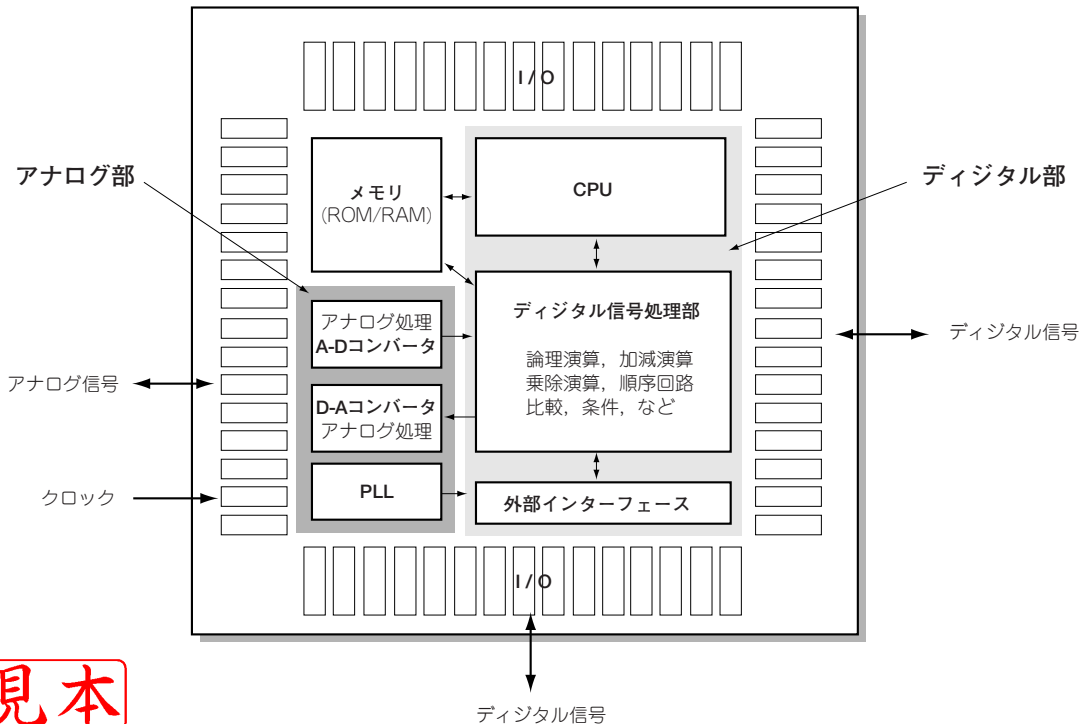
## 1.1 ミックスド・シグナルLSI

アナログとデジタルの混載したシステムLSI(ミックスド・シグナルLSI)は、たとえば図1.1のような構造をもちます。センサやマイクなどから入力されたアナログ信号は、増幅やフィルタなどのアナログ処理を施した後、アナログ信号を適切な時間的間隔でサンプリングし、A-Dコンバータを通してデジタル信号に変換します。

A-D変換されたデジタル信号、または他のデジタル・システムからのデジタル信号はデジタル信号処理部に入力されます。デジタル信号処理部はシステムの中核にあたります。ここでは、論理演算、加減乗除演算、比較、符号化、条件判断、選択などの各種のデジタル演算処理を施します。また、演算処理した結果のデータはメモリ(ROM/RAM)に書き込んだり、読み出したりします。

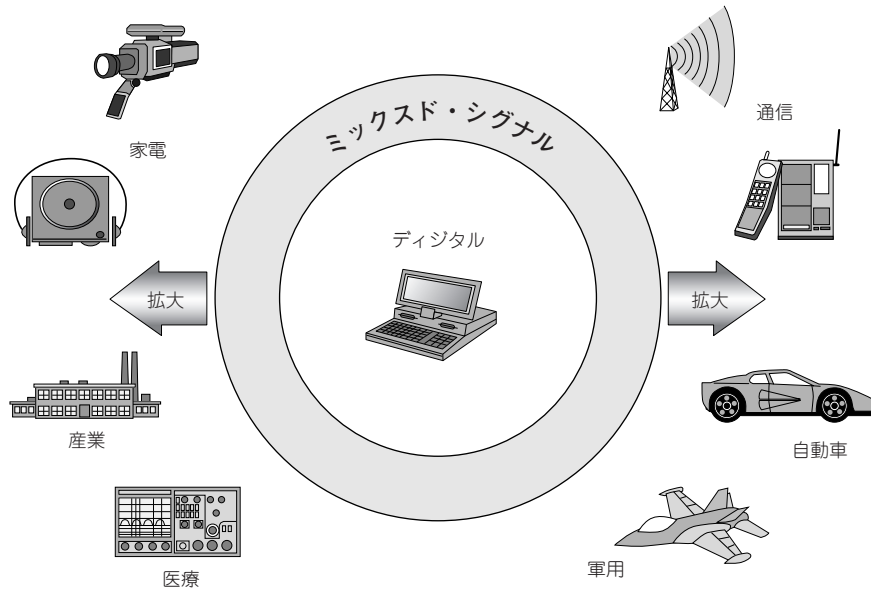
デジタル信号処理部の実現には、CPUとプログラムを用いたソフトウェアによる実現と、演算処理を論理ゲート回路として実現したハードウェアによる実現方法があります。ソフトウェアによる実現では、CPUを用い、実行手順をソフトウェアとしてプログラムします。そのため、処理速度はそれほど速くはありませんが、実行手順の変更が容易です。ハードウェアによる実現は、論理回路を開発するコストは必要ですが、高速処理が可能です。システムLSIでは、通常、CPUによるソフトウェア処理と論理回路によるハードウェア処理(デジタル信号処理部)が混在します。

〔図1.1〕 ミックスド・シグナルLSIの構造



見本

〔図1.2〕 ミックスド・シグナルLSIの適用分野



デジタル信号処理部からの出力は、表示や外部制御、外部システムへのインターフェースなどを行います。表示には7セグメント・ディスプレイや液晶画面、LEDなどがあり、モータやスイッチ、ネットワークなどを制御する場合があります。システムLSIからアナログ信号を出力する場合には、デジタル信号をD-Aコンバータによってアナログ信号に変換しなければなりません。

システムLSIでは、おもな処理はデジタルで行いますが、信号の増幅、フィルタ、アナログ信号とデジタル信号の変換、クロック信号の生成や信号の同調などを行うPLL(Phase Locked Loop)は、アナログ回路として実現されます。このように、多くのシステムLSIでは、アナログ回路とデジタル回路が混在します。

#### ◆ミックスド・シグナルLSIの適用分野

ミックスド・シグナルLSIは、多くの分野で必要とされています(図1.2)。

たとえば、ロボット、工作機械、計測機器などの産業機器をはじめ、家電、医療、自動車、軍用などの多くの電子機器で、アナログ信号の処理や、デジタル信号に変換するためのA-DコンバータやD-Aコンバータを必要とします。携帯電話やBluetoothなどのような無線通信分野でも、復調や変調などにアナログ技術が必要です。また、他の電子機器の制御のためにアナログ信号出力が必要な場合もあります。

さらに、従来のデジタルLSIでも、動作周波数の高速化によって、内部配線や入出力インターフェースなどにアナログ的な設計が必要とされてきています。たとえば、高速バスやネットワークなどのドライバ回路では、出力はデジタル信号ですが、高速であるためにアナログ的な見地からの設計が必要

見本

ミックスド・シグナルLSIの需要は今後も増えることが予想されます。

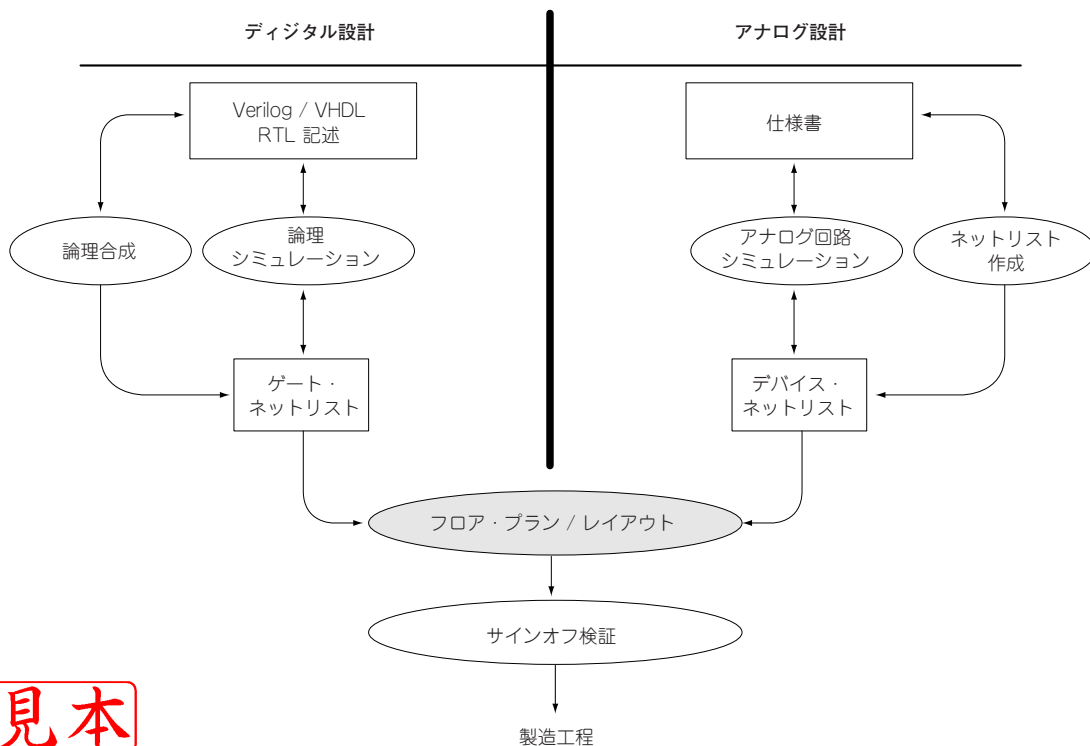
## 1.2 デジタル設計とアナログ設計の設計自動化の進展

現在、多くのミックスド・シグナルLSIの設計は、アナログ回路とデジタル回路を別々に設計し、それぞれの回路が完成した後にレイアウト上で統合しています(図 1.3)。しかし、このような場合、LSI全体の検証が可能なのは設計工程の最終段階です。アナログ回路とデジタル回路を接続した際に機能やタイミングで問題が発生しても、問題点の解析や設計の上流工程に戻るのに多大な労力と時間が必要です。

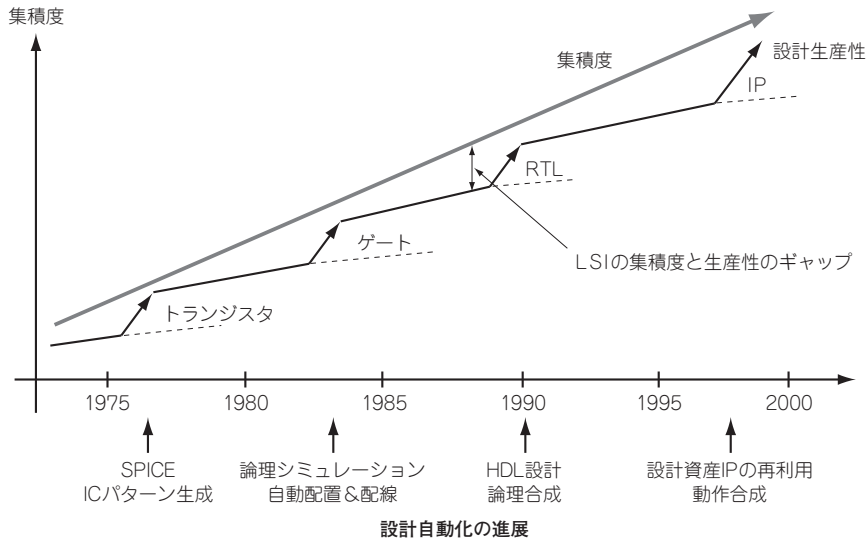
ミックスド・シグナルLSIの検証には、現在、大きく分けて2種類の方法があります。一つ目は、デジタル回路を論理ゲートではなく、すべてトランジスタに置き換え、アナログ回路とともにSPICEシミュレータで検証する方法です。二つ目は、別々なデジタル・シミュレータとアナログ・シミュレータを用い、通信プロトコルを用いて双方のシミュレータを接続して検証する方法です。しかし、これらの方法は、シミュレーション時間が長かったり、実行のための操作性が悪かったり、あまり効率的ではありません。また、アナログ回路とデジタル回路の分離や、個別なシミュレータの起動方法の設定、信号間の接続の設定等に手間がかかる場合があります。

そのため、単一のシミュレーション環境で、容易にミックスド・シグナルの検証を実行できる環境が

〔図 1.3〕 現状のミックスド・シグナル設計手法



〔図1.4〕 デジタル設計での設計生産性の向上



望まれていました。これを可能にするのがVerilog-AMSといえます。

ここでは、Verilog-AMSの解説に入る前に、コンピュータを用いたデジタル設計とアナログ設計の設計自動化の進展を振り返ってみたいと思います。

### ◆デジタル設計手法の進展

では、デジタル設計のパラダイムの変遷をまとめてみます。

#### (1) 論理回路図から論理シミュレーションによる機能検証とタイミング検証の導入

デジタル設計は、初期のトランジスタ・レベルでの設計を経て、論理式、カルノー図やブール代数を用いた論理設計、および、ブレッドボード作成による論理検証によって設計されてきました。しかし、コンピュータの発展とともに、コンピュータ上に論理回路図を入力し、論理シミュレーションによって機能とタイミングを検証する設計手法に移行しました。また、レイアウト設計でもコンピュータによるセルの自動配置や配線が行われ、その結果、大幅に設計期間を短縮することができました(図1.4)。

#### (2) ハードウェア記述言語と論理合成ツールの導入

LSIがさらに大規模、複雑化するにしたがい、さらなる設計生産性の向上が必要になりました。そのため、ハードウェア記述言語(HDL: Hardware Description Language)と論理合成ツールを用いた上位レベル設計へ移行しました。HDLを用いた設計では、回路機能を論理合成可能なHDL記述スタイルで記述し、HDLシミュレータを用いて回路機能が仕様を満足しているかどうかを検証します。

検証したHDL記述は、論理合成ツールによって、HDLに等価な論理式に論理変換された後、実現するテクノロジー(半導体プロセス技術)にしたがって、タイミング、面積、消費電力などの制約条件を満足するように最適化し、論理回路が生成されます。生成されたゲート回路は、論理シミュレータもしくはスタティック・タイミング解析ツールを用いて論理およびタイミングの検証が行われます。

見本

〔図1.5〕 HDLによる設計記述レベルとその内容

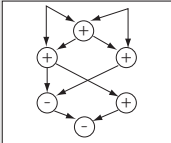
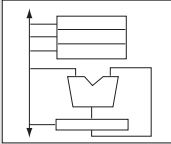
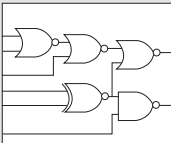
設計記述レベル	記述内容
動作 	仕様や動作などクロックを考慮しない記述スタイル  <div style="text-align: right;">↓ 動作合成</div>
RTL 	レジスタと組み合わせ回路を明確に示した記述スタイル  <div style="text-align: right;">↓ 論理合成</div>
論理 	論理演算子またはゲート・レベルのネットリストによる記述スタイル

図1.5にHDLによる記述レベルとその内容を示します。論理レベルでは、論理式または論理ゲートを用いて論理構造を記述します。RTL(Register Transfer Level)では、順序素子(レジスタ)を明確に記述し、さらに、レジスタ間の組み合わせ回路を記述します。RTLからの合成では、記述した回路構造に従って順序素子と組み合わせ回路が自動生成されます。また、さらに上位の動作レベルでは、クロックに依存しない形で仕様や動作を直接的に記述します。動作合成では、演算処理に必要なクロック数を制約条件として与えてRTL回路を自動生成します。生成される回路の順序素子や演算器の個数はクロック処理数によって大幅に異なってきます。

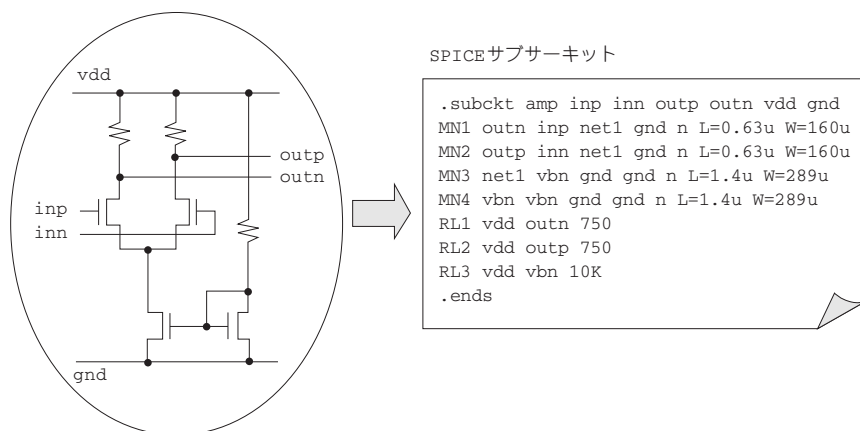
### (3) IP (設計資産) の活用とハードウェア・ソフトウェア協調設計の導入

合成ツールの出現によって、デジタル設計者は、論理レベル設計から、より上位のアーキテクチャ設計に集中できるようになりました。現在、多くの回路がIP(Intellectual Property: 知的財産の略。ここでは設計資産)として流通しています。IPには、システム・シミュレーションを実現するための動作モデル(ソフトIP)や、実際にLSIを設計するために必要な合成可能なモデル(ファームIP)があります<sup>注1</sup>。既存回路をIP化して再利用したり、標準的な機能をもつIPを外部から購入したりして、大規模なシス

**見本**

注1) さらにレイアウト・データによるIPをハードIPという。

[図 1.6] アナログ回路(OP アンプ)と対応する SPICE ネットリスト



テムLSIを短時間で設計することが可能になりました。最近では、C言語でシステムを記述、検証し、C言語からハードウェアを生成する動作合成ツールや、ソフトウェアとハードウェアの協調設計環境が整いつつあります。

このように、デジタル設計では、LSIの大規模化に伴い、設計の生産性を向上させるためのコンピュータによる設計自動化が進展しています。

#### ◆アナログ設計手法の進展

アナログ設計も、かつては手作業による回路設計とプリントボードの作成による検証が行われてきました。しかし、デジタル設計と同様に、コンピュータの発展によって、コンピュータ上にトランジスタ回路図を入力し、シミュレーションを行う検証に移行しました。アナログ回路の動作検証には回路シミュレータである**SPICE**(Simulation Program with Integrated Circuit Emphasisの略、スパイス)シミュレータが使われます(図 1.6)。

SPICEシミュレータは、米国のU.C.Berkeley大学が1970年代後半に開発したシミュレータで、キルヒホッフの法則から回路方程式を作成し、対応した微分方程式の計算から、回路内の各ノードの電圧と電流を求めるシミュレータです。SPICEシミュレータは、半導体プロセスのパラメータ、たとえば、単位面積当たりの容量や抵抗率、拡散濃度、酸化膜の誘電率、配線の伝導率などからトランジスタやキャパシタなどの半導体デバイスの等価モデルを作成し、AC解析、DC解析、過渡解析などによってアナログ回路のもつ機能や特性を解析します。SPICEシミュレータのソース・コードが公開されたことにより、多くのベンダから独自に拡張されたSPICEシミュレータが販売されています。

アナログ回路は非線型動作であり、回路ノードの電圧や電流の計算には、時間ステップごとに多くの微分方程式を解かなければなりません。アナログ回路も大規模化しているため、SPICEシミュレーションには長い処理時間が必要です。そのため、過去、デジタル設計と同様にアナログHDLが提案されてきました。しかし、標準化されていないHDLは、設計ツール固有なHDLであり、設計資産としての再利用性や汎用性に問題があります。また、デジタル設計のような論理合成ツールに対応するツールがな

見本

ければ、HDLで検証した回路を再度トランジスタ・レベルで設計しなければならず、残念ながらアナログHDLによる設計はあまり普及していませんでした。

### 1.3 ハードウェア記述言語 Verilog-AMS とは

Verilog-AMSは、IEEE1364-1995 Verilog-HDL仕様に基づき、デジタル設計だけでなくアナログ設計を含めたミックスド・シグナル設計を可能とする言語仕様をもちます。Verilog-AMSによって、デジタル回路だけでなく、電流や電圧に基づいたアナログ回路動作の記述が可能です。Verilog-AMSを用いることで、ミックスド・シグナル設計の抽象度の高い上位レベルの検証から、論理ゲートやトランジスタを含んだ下位レベルの検証まで幅広い範囲での検証を可能とします。

ここではまず、ハードウェア記述言語の基礎概念、およびVerilog-AMSの標準化の歴史とその言語の特徴について紹介します。

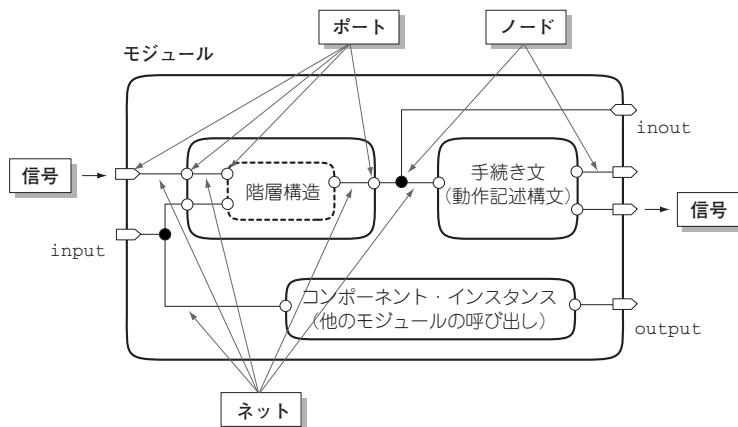
#### ◆ハードウェア記述言語とは

回路のハードウェア動作を言語レベルで記述可能なのがハードウェア記述言語（HDL）です。HDLは、C言語などのプログラミング言語の構文に似ていますが、大きな違いは、プログラミング言語の処理は手続き構文に基づいた順序的な処理であるのに対し、並行的に動作するハードウェア動作を記述できることです(図1.7)。

HDLの大きな特徴に「信号」や「ネット（接続）」、および「時間」の概念をもつことがあげられます。信号は、デジタル信号では1と0の値をもち、アナログ信号では電圧と電流をもちます。ネットは、ポートを介してコンポーネント同士を接続し、ドライバによってドライブされ、指定した遅延時間を経た後、他のコンポーネントの入力に信号値を伝えます。

これらの概念は、実際にハードウェアとして物理的に接続されたコンポーネントが、それぞれの入力信号に従って独立的に動作するのと同一です。

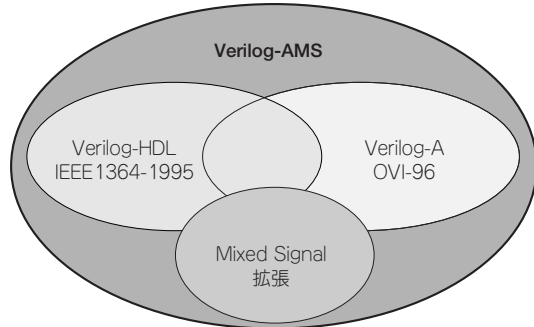
〔図1.7〕 ハードウェア記述言語の概念と構造



見本



〔図1.8〕  
Verilog-AMSの言語仕様



接続可能なコンポーネントには、論理ゲートなどのプリミティブ、同時動作する手続き文、他の回路の呼び出し(インスタンス)、サブプログラムなどがあります。手続き文には、論理演算、算術演算、条件文、イベント文、代入文、遅延制御文などを記述することができます。それぞれの手続き文は並行的に実行されますが、手続き文内の構文はプログラミング言語のように順序的に実行されます。

代表的なHDLには、Verilog-HDLとVHDLがありますが、ハードウェアを記述する目的と基本的な概念はほぼ同じです。しかし、Verilog-HDLは、LSI設計の記述性を重視した言語であり、記述が容易で、ゲート・レベル・シミュレーションの機能も充実しているという特徴があります。一方、VHDLは設計仕様のドキュメント言語として開発されたため、読解性に優れ、システム仕様のあいまいさを排除した厳格な言語だという特徴があります。Verilog-HDLはIEEE1364-1995、VHDLはIEEE1076-1987、IEEE1164-1992にて標準化されています。

HDLは、Verilog-HDLまたはVHDL言語専用のシミュレータ、もしくは両方のHDLをサポートするシミュレータを用い回路動作を検証します。VHDLとVerilog-HDLは、将来どちらかのHDLが優先的に使用されるようになるわけではなく、それぞれの特徴を生かした使い分けが進むと考えられています<sup>注2</sup>。

#### ◆Verilog-AMS標準化の歴史

Verilog-HDLは、1980年代にゲートウェイ・デザイン・オートメーション社(Gateway Design Automation)が販売していたVerilog-XLシミュレータ専用のHDLとして開発されました。Verilog-HDLはC言語を基にし、設計や記述のしやすさを重視した言語です。1989年にゲートウェイ社はケイデンス・デザイン・システムズ社(Cadence Design Systems)に買収されましたが、その後、Verilog-HDLの言語仕様が公開され、第三者でもVerilog-HDLに対応したツールの開発が可能になりました。そして、標準化作業を経て1995年にIEEE1364-1995 Verilog-HDLとして標準化されました。Verilog-HDL構文規則は、LRM(Language Reference Manual)にて定義されています。

一方、Verilog-AMSの前身であるアナログ設計記述言語Verilog-Aは、ケイデンス・デザイン・システムズ社のアナログ・シミュレータ用のアナログHDLを基本とし、それにVerilog-HDLの構文を取

**見本**

注2: ただし、言語仕様の改定によって、Verilog-HDLとVHDLの差が少なくなってきたり、ひょっとしたら、将来は一つのHDLとして統合されるかもしれない。

り込んだ言語仕様をもちます。Verilog-Aは、アナログ回路を対象としたHDLで、Verilog-HDLに関連した団体であるOVI(Open Verilog International)にて標準化されました。

Verilog-AMSは、IEEE1364で標準化されているVerilog-HDL(以後、IEEE1364 Verilog-HDLを区別するためにVerilog-Dと呼びます)を基本とし、さらに、アナログ設計言語Verilog-Aと、アナログ回路とデジタル回路の容易な接続と混在を可能とする構文を追加した言語です(図1.8)。Verilog-AMSによって、デジタル回路だけでなく、電流や電圧に基づいたアナログ回路動作の記述が可能であり、抵抗、容量、インダクタといったアナログ回路素子を含めた検証ができます。

このように、Verilog-AMSを用いることで、アナログ回路の上位レベルでの検証、デジタル部とアナログ部の混在検証が容易に実行でき、その結果、アナログ設計およびミックスド・シグナル設計の設計品質の向上と設計期間の短縮が期待できます。

## 1.4 Verilog-AMSを用いた設計メリット

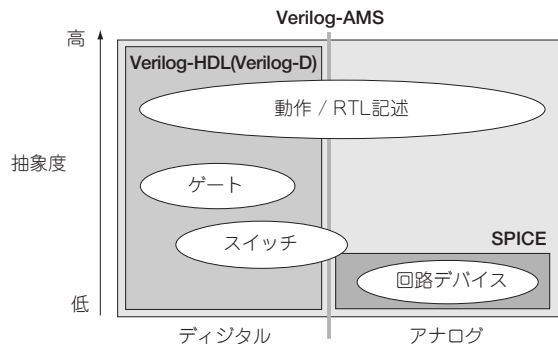
アナログ設計では、デジタル設計のようにHDLから論理回路を生成するような合成ツールはまだ実用化されていません<sup>注3</sup>。そのため、アナログ設計ではHDLを合成ツールの入力言語として使用するような使用法は現在のところ考えられません。そのため、Verilog-AMSの使用目的は、デジタル回路とは若干異なっています。

これまでの説明の繰り返しにもなりますが、アナログ設計およびミックスド・シグナル設計に対してVerilog-AMSを使用することは、まとめると以下のようなメリットが考えられます。

- (1) ミックスド・シグナル検証の実現による設計品質の向上
- (2) 抽象度の高い上位レベルでの仕様検討と動作原理の理解が容易
- (3) ミックスド・シグナル・シミュレーションの高速化
- (4) バックアノテーションやノイズを含めた詳細シミュレーション
- (5) IP化による回路の知的保護と再利用

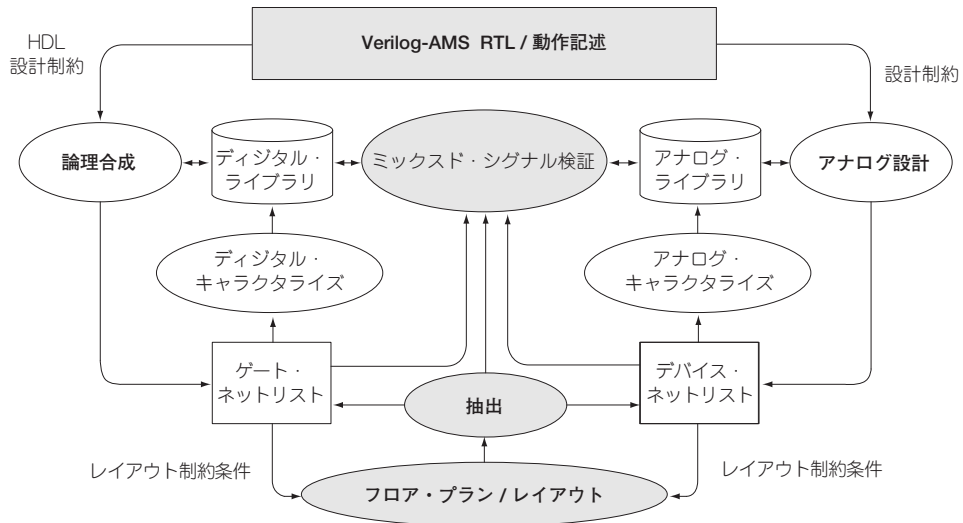
以下にそれぞれの項目について解説します。

〔図1.9〕  
Verilog-AMSによる記述可能な抽象度レベル



注3: 本書Appendix「A.5 アナログ合成」でアナログ合成について解説している。

〔図 1.10〕 Verilog-AMS を用いたミックスド・シグナル設計フロー



#### ◆ミックスド・シグナル検証の実現による設計品質の向上

Verilog-AMSで記述可能な抽象度のレベルを図1.9に示します。Verilog-AMSを用いることで、抽象度の高いレベルでのアナログとデジタルが混在したミックスド・シグナル検証から、論理ゲートやトランジスタ・レベルでの設計検証、さらにレイアウト結果から容量や抵抗値、遅延値をバックアノテーションした最終的なサインオフ検証までが同一の環境で実行できます。

これらの上位から下位までの一連の検証は、アナログ回路とデジタル回路の混在だけでなく、個別な回路に対する上位レベルから下位レベルへの設計レベルの混在も可能です。Verilog-AMSを用いたミックスド・シグナル設計フローの例を図1.10に示します。Verilog-AMSを用いることで、アナログとデジタルとの接続による機能やタイミングの検証が実現でき、設計する回路の品質が向上することが期待できます。

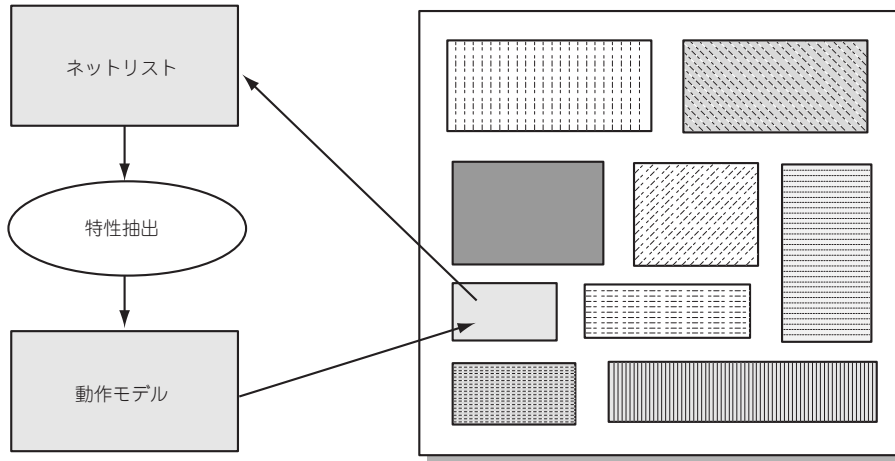
#### ◆抽象度の高い上位レベルでの仕様検討と動作原理の理解が容易

抽象度の高い上位レベルの仕様検討では、アナログやデジタルの動作構文を用いて、仕様に基づいて回路動作を記述、検証します。Verilog-AMSでは、三角関数、算術関数などの関数、ラプラス変換や $z$ 変換などの演算子がサポートされ、Verilog-HDLより抽象度の高い記述を可能とします。

上位レベル設計では、既存の動作モデル、または新規に開発した動作モデルを用いて、各回路ブロックに対応するモデルを組み合わせて回路全体を構成します。その後、モデル内に定義された特性パラメータ値を全体の仕様に基づいて最適化します。通常、アナログ・モデル内のパラメータには、機能、遅延、周波数特性、ゲイン、入出力インピーダンスなどのパラメータが定義されています。全体の回路仕様を満たさせる各ブロックの特性を決めた後、各ブロックのトランジスタ・レベルでの詳細設計を行い

見本

〔図1.11〕既存回路の動作モデルへの入れ替え



ます。

多くのアナログ回路は、デジタル信号処理に基づいたデジタル回路に置き換わっています。しかし、その基本動作の理解には、アナログ処理のほうが理解しやすい場合が多くあります。Verilog-AMSを用いることで、動作原理の理解が容易になり、学習するための手助けになることが期待されます。

#### ◆ミックスド・シミュレーションの高速化

すでに設計が終了し、詳細な動作検証によって特性が確認された回路に対応した動作モデルを作成し、より大きな回路を含めた検証を実現します。たとえば、図1.11のように、すでに回路動作を検証した回路に対して動作モデルを作成し、検証済みの回路を順次動作モデルに入れ替えながら全体シミュレーションの高速化をはかります。動作モデルのシミュレーションはSPICEシミュレーションに比べて数十倍高速です。

既存回路の特性を考慮しない理想的な数式に基づいた動作モデルでは、検証精度が上がらない恐れがあります。全体の検証精度を維持するためには、動作モデルでモデル化する特性の選定と、回路レベルから特性を抽出するためのテストベンチの作成が重要です(第11章「11.1 モデル化フローとモデル作成のガイドライン」を参照)。

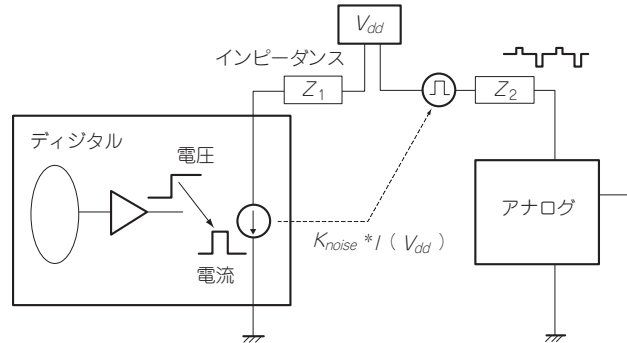
これによって、従来、実用的な時間では終了しなかったミックスド・シグナルLSIの検証が可能になります。

#### ◆バックアノテーションやノイズを含めた詳細シミュレーション

ミックスド・シグナルLSIでは、両者を統合する半導体プロセスを用います。単独のアナログ回路にはバイポーラ・プロセスが適していますが、ミックスド・シグナルLSIでは、CMOSプロセスを用いてアナログ回路を設計するか、またはCMOSプロセスにバイポーラ・プロセスを付加したBi-CMOSプロセスを用います。

**見本**

〔図1.12〕  
アナログ回路に対するデジタル  
回路からのノイズの影響を検証す  
る例



ミックスド・シグナルLSIでは、アナログ回路とデジタル回路の相互の影響を最小限に抑えるような注意深い設計が必要です。微弱な信号を扱うアナログ回路は、ノイズの除去が難しく、ノイズの混入によって特性が悪化するため、ノイズの混入は防がなければなりません。そのため、通常、アナログ回路とデジタル回路では、電源やグラウンド配線を別にし、さらに、デジタル回路のスイッチングで発生するノイズの影響がアナログ回路に回り込まないように、チップ内のレイアウトに注意します。アナログとデジタル間のノイズの影響を含めた検証では、たとえば、デジタル回路のスイッチングによる電源電流の変化をモデル化し、さらにアナログ電源にノイズを発生するモデルを入れ、アナログ回路への影響を検証します。また、必要があれば基板からの回り込みの影響も考慮します。Verilog-AMSでは、多くのランダム発生システム・タスクが提供され、ノイズを含めたミックスド・シグナル検証も可能です。

また、デジタルLSIでは、レイアウト後のバックアノテーションによるサインオフ検証で、セル間の配線による遅延値を論理ゲート回路に付加して検証を行います。この場合、電源やグラウンドは電圧変動のない理想的な電源として扱われます。しかし、電源やグラウンドに流れる電流による電圧降下の影響、内部配線同士の容量による影響、LSIに接続する外部負荷や容量を含めた検証など、高速で微細なデジタルLSIでは、検証精度を向上させるために、アナログ的な解析手法やシミュレーションが必要になります。

Verilog-AMSでは、論理的な精度で十分なデジタル回路はそのまま論理ゲートとしてシミュレーションを実行し、アナログ回路や詳細な検証が必要なデジタル回路はアナログ回路レベルとして実行することで、検証精度を向上させることができます(図 1.12)。

#### ◆IP化による回路の知的保護と再利用

半導体製造ベンダでは、自社のプロセスに対応したシミュレーション・ライブラリを設計する顧客に提供します。シミュレーションのためのライブラリを外部に提供する場合、ネットリストによる提供では回路構造が明確になってしまいます。Verilog-AMSを用いた動作モデル・ライブラリがあれば、詳細な回路構造を必要とせず、シミュレーションが可能です。しかも、トランジスタ・レベルより高速なシミュレーションを行うことができます。さらに、シミュレーション実行時に、異なる特性が必要な場合でも、ライブラリ内のパラメータを変更するだけでよく、上位レベルでの検証も容易です。

見本

Verilog-AMSはIEEEで標準化されたHDLです。そのため、Verilog-AMSをサポートするシミュレータであれば、異なるシミュレータ環境に依存しない、汎用的なライブラリとして提供することができます。

## 1.5 第1章のまとめ

ミックスド・シグナル設計は今後もますます重要になってくると考えられます。Verilog-AMSを用いることで、抽象度の高い上位レベルでの仕様検証から、動作モデルを用いたミックスド・シグナル・シミュレーションの高速化、バックアノテーションを含めたシミュレーション、IP化による回路の知的保護などが可能です。

Verilog-AMSを用い、アナログ設計とデジタル設計を統合した環境で設計することで、アナログ回路とデジタル回路間の接続や分割に関する問題をあらかじめ検証することができます。

# 第 2 章

## Verilog-AMS の基本構造

Verilog-AMSは、Verilog-D<sup>注4</sup>と同様に、モジュール(module)文によって設計回路の機能を定義します。モジュールのインターフェース部では、回路名や入出力ポート名、入出力属性などの回路のインターフェース情報を定義し、モジュール本体には、構造記述および動作記述によって回路のもつ機能やタイミングを定義します。

**構造記述**は、論理ゲートやプリミティブ、他のモジュールの呼び出しを行い、それらを接続することで回路構造を記述します。**動作記述**は、手続き文を用いてアナログまたはデジタル動作を記述します。Verilog-AMSでは、同一モジュール内に「アナログ動作」と「デジタル動作」を混在させることができます。

本章では、モジュール構文、モジュール内に定義可能な構文の概要や構成を最初に紹介し、次にVerilog-AMSの字句に関する規約を紹介します。それぞれの構文の詳細は、第3章以降に紹介します。

### 2.1 モジュール

設計回路を定義するのがモジュールです。モジュールは、回路名、入出力ポート宣言、データ・タイプ宣言などのインターフェース部分と、回路機能を構造記述や動作記述によって記述したモジュール本体で構成されます。Verilog-AMSでは、アナログ構文とデジタル構文を混在させることができます。動作記述では、代入文、条件文、演算子などの手続き構文を用いて回路動作を記述します。構造記述では、他のモジュールやプリミティブをインスタンス化して接続を指定します(図2.1)。

ここでは、モジュール構文とモジュール内に含まれる各種の宣言や構文の概要を紹介します。

#### ◆モジュール構文

モジュール定義は、`module`文で始まり、`endmodule`文で終わります。

`module`文の次に**モジュール名**と**ポートリスト**を指定します。ポートリストはオプションで、モジュールへの入力、出力または入出力のポート名をリストして指定します。一つの設計ファイルは、一つまた

見本

<sup>注4</sup> 本書では、IEEE1364 Verilog-HDLをVerilog-AMSと区別するためにVerilog-Dと呼ぶことにする。