特集

仕様レベル設計検証システム：Supervise

Design and Verification System for Specification Levels: Supervise

あらまし

現在，ハードウェアの設計規模増大に伴い，装置開発期間のより一層の短縮が強く望まれている。このため，富士通はICL社と共同で仕様レベル設計・検証システム「Supervise」を開発し，仕様から詳細設計に至るまでシームレスな設計・検証環境を実現した。Superviseはハードウェア設計言語VHDLを拡張した*VHDL+*をベースとし，データおよびタイミングの抽象記述を可能とすることにより，仕様設計段階からの早期シミュレーションを実現した。これにより検証期間を従来比で20～30%短縮することが期待できる。

本稿では，まずSuperviseの主要な機能を解説し，Superviseがシステム設計，設計資産(IP)再利用，テストベンチ作成，およびプロトコル検証など幅広い分野で適用可能なことを示す。つぎに富士通のネットワーク装置の設計に適用した実例を紹介し，最後にIEEE標準化，高位レベル合成，ハード/ソフト協調設計など将来の展望について述べる。

Abstract

Hardware systems continue to grow in scale and complexity, and there is a strong demand for even shorter development periods. To meet this requirement, Fujitsu and ICL have developed a design and verification system for specification levels called“ Supervise, ”and have realized a seamless design and verification environment by providing integrated support for the development stages from specification to detail design. Supervise uses VHDL+, which is an extended hardware design language of VHDL that enables abstract descriptions of data and timing and thus enables simulation at the early stages of specification design. We expect that Supervise can reduce the verification period by 20 to 30%.

This paper first describes the key functions of Supervise and points out its wide range of applications, which include system design, IP reuse, test bench generation, and protocol verification. Then, several examples of applying Supervise to the design of devices for Fujitsu networks are introduced. Finally, the future prospects of the system, including efforts that have been taken for IEEE standardization, synthesis at higher levels, and co-design of hardware and software, are described.

斉藤　実（さいとう　みのる）

1981年東北大学工学部電子工学科卒。
同年富士通入社。以来CADシステムの開発に従事。
CAD開発統括部第一CAD開発部

ま え が き

ハードウェアの設計規模が増大し，回路が複雑化する一方で，製品の開発期間は一層の短縮化が求められている。この要求を満たすためには，デザインサイクルの早期段階における設計効率の向上と，早期検証を実現し，設計の後半での手戻りを極力排除する必要がある。

富士通とICLは，設計の上流段階での早期検証を実現するため，仕様レベル設計・検証システム：Superviseを開発した。[(1)-(3)]

Supervise の機能

Superviseでは，仕様および論理記述手段として，ハードウェア設計言語VHDL[(注1)]を拡張し，より抽象的な記述を可能とした設計言語*VHDL+*を定義する。Superviseは，*VHDL+*言語のコンパイラ，シミュレータ，およびブラウザから構成される仕様レベル設計・検証システムである(図-1)。以下にSuperviseの特長を述べる。

● 抽象的記述による早期モデリング

一般的設計手法では，仕様設計書は文書として作成し，機能設計の段階で初めてハードウェア記述言語によりCADエントリし，シミュレーションを開始する。このため仕様段階と機能設計段階の間にデザインギャップが存在する。

Superviseは，より抽象的な記述を可能とした設計言語*VHDL+*を採用することにより，仕様段階における早期モデリングと早期シミュレーションで仕様のサインオフ[(注2)]を可能とし，デザインギャップを排除するものである。この結果，仕様から詳細設計に至るまでシームレスな設計・検証環境を実現できる(図-2)。

以下に*VHDL+*がサポートする抽象的記述例を説明する。

(1) データの抽象化

概略仕様段階から，*VHDL+*による記述およびシミュレーションを可能とするため，抽象的レベルでのデータ定義を可能とした。

(2) タイミングの抽象化

事象間のタイミングを，並列または順次実行として定義することを可能とした。また物理的時間(ns，psなど)，あるいはクロックサイクルによるタイミング定義において，範囲を持たせることを可能とした。

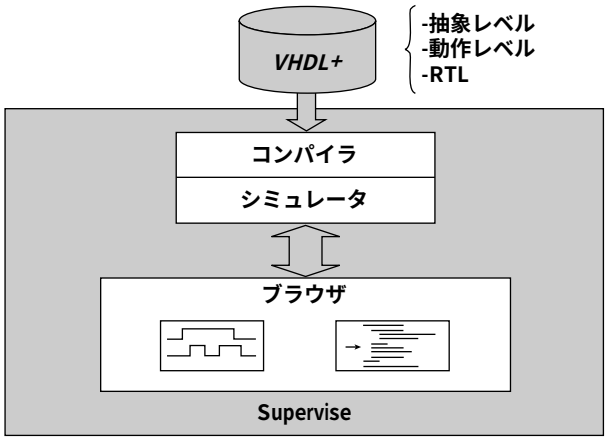

図-1 Supervise構成図
Fig.1-Configuration of Supervise.

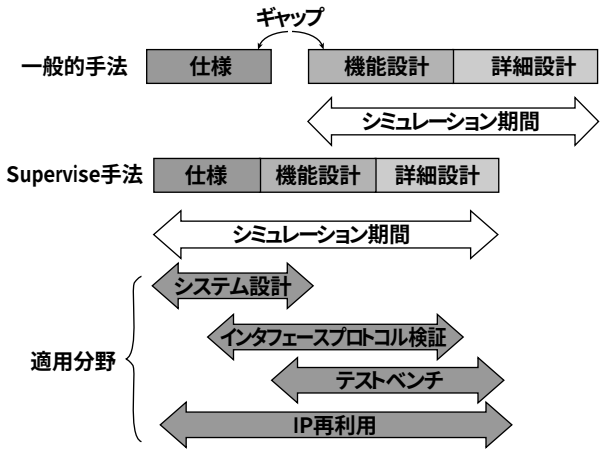

図-2 Superviseによる設計手法
Fig.2-Design method by using Supervise.

(3) 確率的事象の表現

指定確率に従って，複数の動作から一つの動作をランダムに選択し，実行することを可能とした。

● 独立したインタフェース記述

装置が複数のユニットから構成されている場合，*VHDL+*はユニット間の通信プロトコルを，独立したインタフェース仕様として記述できるようにした。またこの仕様はシミュレーション可能であるため，仕様設計段階からシミュレーションによりユニット間のインタフェースプロトコルをチェックすることができる。これにより，インタフェース仕様の明確化と，確実なコンカレント設計を保証する。

インタフェースは以下の機能によって階層的に記述できる。

(1) Transaction

複数のMessageから構成される複合的通信手段を定義

(注1) VHSIC (Very High Speed Integrated Circuit) Hardware Description Languageの略語で，IEEE標準のハードウェア記述言語。

(注2) 現段階の設計を完了し，下位工程への設計データリリースを承認すること。

する。多岐にわたる通信を，混在して記述することが可能である。主に抽象度の高いシステムレベル記述において使用される。

(2) Message

特定の通信手段を定義する。他のMessageに分解して階層的な記述をすることが可能である。上位レベルMessageから伝達される情報を，Signalにマッピングすることにより，*VHDL+*とVHDLとのインタフェースを確立する。主に抽象度の中間レベルからRTL(注3)までのレベルで使用される。

● ミックスレベルシミュレーション

Superviseは，階層的インタフェース記述により抽象レベルからRTLまでの広範囲の記述を含む，ミックスレベルシミュレーションを可能とする。図-3の例では異なるレベル(抽象レベル，動作レベル，RTL)で記述されたユニットをインタフェース記述によって結合している。インタフェース記述では，Transaction，Message，およびSignal間の相互変換が自動的に行われる。これにより，仕様から詳細設計までのシームレスな設計検証が可能となる。

● 抽象記述による高速シミュレーション

抽象度の高いレベルでは詳細記述に比べ記述量が1/10～1/100程度となるため，高速シミュレーションが可能となる。周辺回路を抽象的記述とし，ターゲットのみを詳細記述することによって，全体の記述量を削減し，シミュレーションを高速化することが可能である。

Supervise の適用分野

Superviseは，以下に述べるように広い分野において適用可能である(図-2)。

● システム設計

システム全体をトップダウンに設計するための使用法であり，本来の*VHDL+*機能を最大限に利用する適用分野である。まずシステム全体をいくつかのユニットに分割し，各ユニット間のインタフェースプロトコルを，*VHDL+*のインタフェース記述で行う。設計の初期段階では，ユニットの機能は*VHDL+*の抽象レベルで記述する。設計の進捗に伴い，ユニットを分割するとともに，記述レベルを詳細化し，最終的にはRTLまでブレークダウンする。

本手法は，ICL社における大型コンピュータの開発,(1) Ericsson社における交換機開発(3)などで実績がある。

● IP再利用

*VHDL+*で記述することにより，既存の設計資産(IP)の再利用を効果的に行うことが可能である(図-4)。IPは，抽象的記述モデルあるいは合成可能なRTLモデルなど，任意の記述レベルで提供可能である。ユーザ設計ユニットとのインタフェースの抽象度を上げることにより，設計者はIPのインプリメントに依存せず，機能仕様のみに着目し仕様設計を行うことが可能である。

● インタフェースプロトコル検証

Superviseはシミュレーション実行中，各ユニット間の通信が，インタフェースプロトコルを遵守しているかを常にチェックする(図-3)。Transactionとして記述される抽象度の高いレベルから，信号レベルのタイミングチャートで表現されるレベルまで，広範囲のプロトコルチェックが可能である。

● テストベンチ(注4)機能

ハイレベルで記述したテストベンチを，階層的インタフェース記述により，任意のレベルで記述された設計データと結合し，シミュレーションすることが可能である。抽象度の高いレベルでテストベンチを記述することにより，記述量を削減することができる。また発生確立

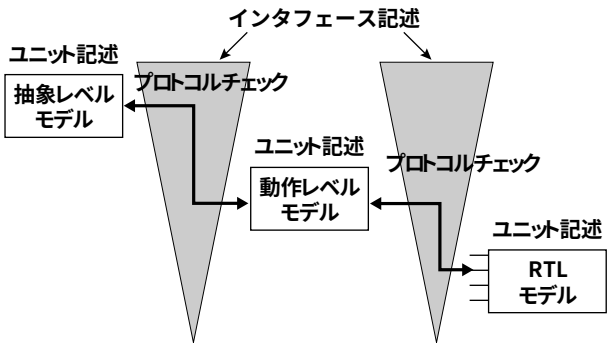

図-3 ミックスレベルシミュレーション
Fig.3-Mixed level simulation.

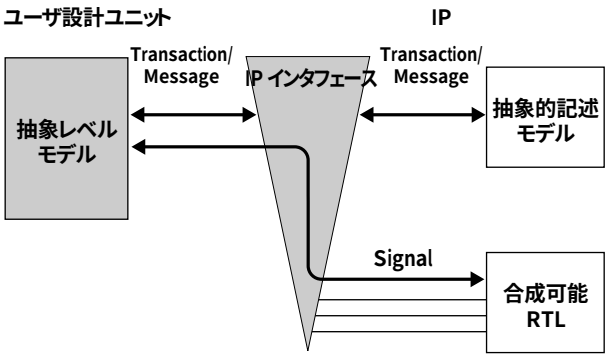

図-4 IP再利用(仕様レベル)
Fig.4-IP reuse(Specification level).

(注3) Register Transfer Levelの略語で，レジスタ間の信号のやりとりで表現する内部構造を意識したハードウェア記述レベル。

(注4) シミュレーションのためのテストデータを生成したり，シミュレーション結果をチェックするための記述。

を指定した，疑似ランダム動作が可能となる。これにより，データの内容だけでなく，シーケンスやタイミングをランダムに変化させたストレス試験が容易に実現できる。

Superviseの適用例

ここでは富士通において，ネートワークルータの一部にSuperviseを適用した実例を紹介する。とくに，ネットワーク装置用DMAC（DMA Controller）チップの設計のためテストベンチ機能，およびプロトコル検証機能で顕著な効果を得た。

DMACシミュレーションの概略構成およびデータフローを図-5に示す。シミュレーションモデルはSparcLITEモデル，RAMモデル，被シミュレーション対象であるDMACモデル，二つのインタフェース記述，およびテストベンチから構成される。

テストベンチはネットワークをモデル化したものであり，フレームと呼ばれるまとまった転送単位の送受信動作を，*VHDL+*の抽象的レベルで記述している。その機能は送信したデータと，返信されたデータを抽象レベルで比較し，ネットワーク仕様の妥当性をチェックするものである。

二つのインタフェース記述，CS_RXとCS_TXは同一のインタフェースプロトコルで規定されるため，ここでは一つのインタフェース記述を２か所で使用し，記述量を削減している。

● インタフェース仕様記述

CS-RX/TXインタフェース仕様は，フレームの構成を階層的に定義する最上位レベルのメッセージと，その構成要素である下位レベルメッセージの集合として記述される。フレーム転送のタイミングチャートとプロトコルとの関係を図-６に示す。フレームはその先頭を表す“ dt_tr_start ”，中間を表す“ dt_tr_middle ”，および末尾を表す“ dt_tr_eof ”の各下位レベルメッセージから構成される。シミュレータは，これらのメッセージがプロトコルを遵守して通信しているかを常に監視し，違反を検出するとプロトコルエラーとしてユーザに通知する（プロトコルチェック）。

● テストベンチ記述

上述のインタフェース仕様として，フレーム転送の詳細なシーケンスを記述しているため，テストベンチは簡潔な抽象記述でコンパクトに実現できる。またフレームデータに，指定した確率でパリティエラーをランダムに挿入したり，誤ったフレーム転送シーケンスを発生させたりし，DMACのエラー検出回路が正常に動作するかどうかを自動的にチェックする。このように，データの内容や，シーケンス，さらにエラー状態をランダムに生成し，コーナーケースを効果的に検証するストレステストを実現できる。

● 適用効果

*VHDL+*を適用することにより，一般的なテストベンチに比較して，以下に示すような効果を得た。

(1) フレームデータをランダムに生成することにより，テストデータを含めたテストベンチ記述量を大幅に削減（従来の1/10以下）。

(2) 正常系動作における境界条件（フレーム長，フレーム間隔，フレームシーケンスなど）を考慮したランダム生成により，コーナーケース検証を可能とし，カバレッジを向上（フレーム長ランダムテストにおいて実

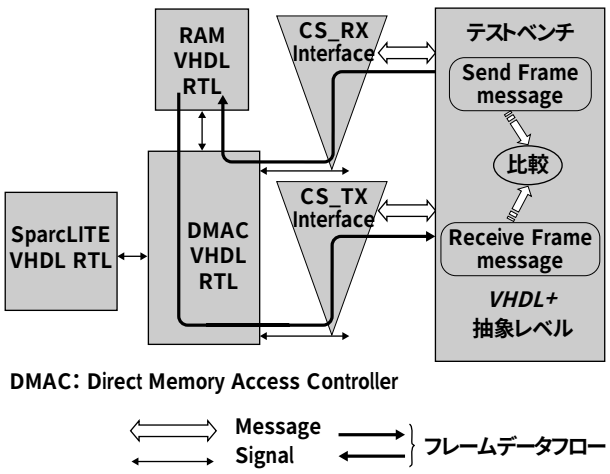

図-5 DMACシミュレーションモデル構成
Fig.5-DMAC simulation model.

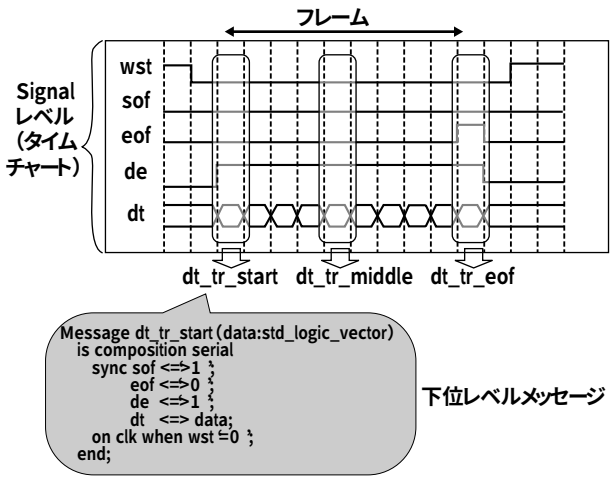

図-6 タイミングチャートとプロトコルの関係
Fig.6-Relation between timing diagram and protocol.

際にバクを検出)。

(3) 異常系動作(パリティエラー，フレームシーケンスエラーなど)のランダム生成，およびエラーチェック機能の自動モニタにより，異常系の検証を実現し，装置の信頼度を向上。

将 来 展 望

● IEEE標準化

現在*VHDL+* のIEEE標準化を目的としてSID(System and Interface based Design)ワーキンググループが活動を行っている。2000年初頭までに*VHDL+* 正式版LRM(Language Reference Manual)草稿を作成し，2000年下期に*VHDL+*標準化決定を目標としている。

● 高位レベル合成

現状では*VHDL+*のハイレベル記述からRTLへの設計はマニュアルで行う必要があるが，*VHDL+*の抽象的記述から合成可能なRTLへの変換(高位レベル合成)に対する強い要求がある。この高位レベル合成を実現することにより，IPの詳細なインタフェースプロトコルを隠蔽した設計手法(オブジェクト指向設計)が可能となる。

● ハード/ソフト協調設計

今後，ハード/ソフト協調設計に対する要求がますます強くなることが予測される。現状の*VHDL+*は，ハードウェア仕様の記述性は強力であるが，今後ハード/ソフト協調設計・検証に対応するためには，C/C++などのソフトウェア言語とのインタフェースを強化する必要がある。

む す び

富士通では，ネットワーク装置設計にSuperviseを適用し，RTLで設計された回路の論理シミュレーション用テストベンチ記述，およびプロトコル検証に効果的に利用可能なことを実証した。

今後は上記設計で開発した*VHDL+*資産をIPとして広範囲に適用することにより，システム検証期間の更なる短縮を図る。またIEEE標準化に向けて活動を行うとともに，高位レベル合成，他言語との結合など広範囲の機能拡張を目指す。

参 考 文 献

(1) A. Jebson, C. Jones and H. Vosper：CHISLE：an Engineer s tool for hardware system design. *ICL Technical Journal*, 8, 3, pp.506-519(1993).

(2) M. M. K. Hashmi and A. C. Bruce：Design and Use of a System-Level Specification and Verification Methodology. IEEE European Design Automation Conference, 1995.

(3) D. Wilkes and M. M. K. Hashm：Application of High Level Interface-based Design to Telecommunications system Hardware. Proceedings of 36th Design Automation Conference, pp.778-783(1999).