

## R4400搭載ボードレベルシミュレーション事例

成田宏樹 山銅俊雄 木村高博 長谷川拓己 高橋悦男  
日本電気株式会社

MIPS社のCPUチップであるR4400のフルファンクションのシミュレーションモデルを使用した、ボードレベルシミュレーション事例を紹介する。R4400を含めたボードレベルシミュレーションは、LSI単体では不可能なR4400と複数のLSI間のインターフェースの論理検証を可能にした。実際には、ハードウェアアクセラレータを使用した高速なゲートレベルシミュレーションと、PCIバスVHDLモデルを統合したより実物のボードに近いモデルのRTLでのVHDLシミュレーションが実現できた。本ボードレベルシミュレーションは、我々の開発製品であるUP4800/680に適用され、製品の高品質化に役立った。

### Board-level simulation with R4400 model

Hiroki Narita, Toshio Sando, Takahiro Kimura, Takumi Hasegawa, Etsuo Takahashi  
NEC Corporation

This paper deals with a board-level simulation using a full-functional MIPS R4400 simulation model. Unlike a LSI-level simulation, the board-level simulation can verify the interface logic between R4400s and multiple LSIs. The simulation is carried out at both gate-level and RTL VHDL-level. A hardware accelerator is used at the gate-level simulation, which brings about a satisfactory simulation speed. By integrating a PCI-bus VHDL model into the simulation model, the VHDL simulation can verify almost all of the board logic. This board-level simulation was used in the development of the NEC UP4800/680 UNIX server and contributed to its superior design quality.

## 1 はじめに

我々は現在サーバ、EWS等を開発している。これらの装置の多くにはMIPS社のCPUチップや、高速拡張バスの実質標準規格であるPCIバスを採用している。CPU、PCIバスの仕様は複雑であるため、CPU、PCIバスに接続するLSIはCPU、PCIバスと複雑なデータのやり取りを行うことになる。このような状況下、設計したLSIの品質を上げるためには、CPU、PCIバス、LSIを接続した実際の装置に近いボードレベルでの検証が必要不可欠である。また、ITF(Initial Test Firmware)においても、CPUとLSI全体の複雑なシステムのコントロールをしなければならない。このようなITFの品質を上げるためには、LSIと同様ボードレベルでの検証が必要である。

ボードレベルでの検証を行うためには、実物か、あるいは実物と同等機能を持つCPUのシミュレーションモデルが必要不可欠である。従来は、CPUと同等機能を持つシミュレーションモデルが入手困難であった。このため、ボードレベルの検証は、主に、疑似的なCPUソフトウェアモデルとLSIを接続したシミュレーションや、実チップを使用して製作した評価用装置により行われていた。しかし、疑似的なCPUソフトウェアモデルでは、シミュレーションの網羅率に限界があり、また、ITFの検証は困難である。また、評価用装置での検証では、評価用の装置を組み立てた後、すなわち装置開発工程の後半になるまで検証結果が分からず、論理バグが検出された場合は、開発工程に大幅な後戻りを発生させることになる。これ以外の検証方法として、実チップとエミュレータのリンクによるエミュレーション、実チップとシミュレータのリンクによるシミュレーション等もある。しかしこれらの方法は、費用面等で大きな問題があり実用は困難であった。

今回、MIPS社のCPUチップの1つであるR4400のフルファンクションのシミュレーションモデルとPCIバスのVHDLシミュレーションモデルを入手し

た。これらのモデルを使用したボードレベルシミュレーションを行ったので、その事例を紹介する。

R4400とPCIバスのシミュレーションモデルを使用したボードレベルのシミュレーションを行ったことにより、以下のような効果をあげた。

(1)R4400、PCIバスとのインターフェースの検証  
R4400、PCIバスの仕様を誤って理解していると致命的な論理バグを埋め込んでしまうことになる。R4400モデル、PCIバスシミュレーションモデルを使用してボードレベルシミュレーションを行うことにより、仕様の確認を行うことができる。

(2)設計初期段階での論理検証

ネットリスト等の論理情報が揃った段階で検証が可能である。このため、論理バグが検出された場合でも、設計初期段階で論理修正が可能となり、設計期間へのインパクトを小さくできる。

(3)ITFの検証

ITFがROMに書き込まれている場合、評価用装置でITFのバグが発覚すると、ROMへのITFの焼き付け直しが必要となる。しかし、シミュレーションであれば、ROMへの書き込み情報ファイルを変更するだけでITFの検証が可能となる。

本稿では、まず、ボードレベルシミュレーションの実現のための手法(シミュレータ、R4400モデル、PCIバスモデル)の調査結果を述べる。その後、ボードレベルシミュレーションの初期段階として運用した、ゲートレベルシミュレーションの事例、および、ハードウェア記述言語VHDLを使用したボードレベルシミュレーションの事例について述べる。最後に、ボードレベルシミュレーション結果データを提示し、その結果について考察を加える。

## 2 シミュレーションの実現

1章で述べたように、R4400、PCIバスを含めたボードレベルでの検証を行うためには、R4400、

PCIバスモデルが必要であり、そしてこれらのモデルが使用できるシミュレータが必要である。本章では、市販されているR4400モデル、PCIバスモデル及びシミュレータに関する比較、検討内容と、その結果として我々が選定した、ZYCAD社ハードウェアアクセラレータXPと自社R4400モデルバンク、VANTAGE社VHDLシミュレータOptium、synopsys社PCIバスVHDLモデルについて説明する。

## 2.1 従来手法調査

ボードレベルシミュレーションを実現するための、R4400、PCIバスモデル、シミュレータを選定するために、従来手法の問題点を洗いだした。その結果、以下を主要目的として選定した。

- (1)フルファンクションのR4400モデルの使用
- (2)大規模対応/高速なシミュレータの使用
- (3)設計上流工程でのシミュレーション

従来のボードレベルの検証方法の特徴を表1に示す。

表1. 従来手法の特徴

	評価用 装置	疑似モデル +シミュレータ	実チップ +シミュレータ	実チップ +シミュレータ
検証範囲	○	×	△	○
規模	○	×	×	△
設計の後戻り	×	○	○	△
ITFの検証	○	×	○	○

## 2.2 比較/検討

R4400モデル、PCIバスモデル、シミュレータの実際の選定にあたっては、価格、品質等も考慮しなければならない。これらの条件を含めて、ハードウェアアクセラレータ、ソフトウェアシミュレータ、エミュレータについて行った比較を表2に示す。

表2. シミュレータ比較

シミュレータ	HWアクセラレータ		ソフトウェア		エミュレータ
	フルファンクション	疑似	フル	フルファンクション	
R4400モデル	有	有	有	有	
PCIバスモデル	無(※)	有	有	有	
大規模	○	○	○	△	
高速性	○	×	×	◎	
チップ入手時期	◎	△	○	○	
品質	○	△	○	○	
価格	○	◎	△	×	
総合	◎	○	×	△	

(※): ただし、ソフトウェアシミュレータとリンクすることにより使用可能。

以上のような比較結果により、ボードレベルシミュレーションは、シミュレータは、ZYCAD社ハードウェアアクセラレータXPとVANTAGE社VHDLシミュレータOptiumをリンクして使用し、R4400モデルはXP上で動作するR4400モデルバンク(2.3参照)、PCIバスモデルは、Optium上で動作するsynopsys社PCIバスVHDLモデルを使用することとした。各モデル、シミュレータの特徴を表3に示す。

表3. 選定したモデル、シミュレータの特徴

XP	<ul style="list-style-type: none"> <li>・ゲートレベルハードウェアアクセラレータ</li> <li>・高速</li> <li>・デバッグ機能は少ない</li> </ul>
Optium	<ul style="list-style-type: none"> <li>・VHDLソフトウェアシミュレータ</li> <li>・豊富なデバッグ機能</li> <li>・VHDLの標準的なシミュレータ</li> </ul>
R4400 モデル バンク	<ul style="list-style-type: none"> <li>・フルファンクションで極めて実チップに近い動作</li> <li>・XP上で動作</li> <li>・直接MIPS社からネットリストを入手して作成しているため高品質</li> </ul>
PCIバス VHDLモデル	<ul style="list-style-type: none"> <li>・VHDLソースモデル</li> <li>・Optium上で動作</li> <li>・動作は使用者が定義</li> </ul>

## 2.3 R4400モデルバンクについて

モデルバンクは、ZYCAD社が販売している、汎用プロセッサモデルライブラリである。R4400はその中の1つのモデルである。モデルバンクの特徴を以下に示す。

- (1)モデルバンクは半導体メーカーから提供され

たネットリストから作成される。R4400はMIPS社からネットリストが提供されている。

(2)モデルバンクは、暗号化されたものがユーザーに配布される。暗号化されたモデルバンクは、XPにロードすることにより、動作可能なモデルになる。

(3)MIPS社以外にも、AMD社、TI社といったメーカーのモデルも既に揃っている。

以上のように、ボードレベルシミュレーションのために、XPとOptiumをリンクして使用すること、R4400モデルバンクとPCIバスVHDLモデルを使用することを決定した。しかし、XPとOptiumのリンクシミュレーションは、その品質に未知な部分があるため、ボードレベルシミュレーションの初期段階では、XP単体とR4400モデルバンクを使用したゲートレベルシミュレーションを行い、ゲートレベルでのシミュレーションが安定した時点で、XPとOptiumをリンクし、R4400モデルバンクとPCIバスVHDLモデルを使用したシミュレーションに移行することとした。

### 3 ゲートレベルシミュレーション

本章では、ゲートレベルシミュレーションの以下のステップについて説明する。

- (1)ボードモデル作成
- (2)シミュレーションパターン作成
- (3)シミュレーション結果検証

このゲートレベルシミュレーションでは、ハードウェアアクセラレータを使用したことによる高速なボードレベルシミュレーションを実現できた。

#### 3.1 モデル構築

ボードモデルの構成を図1に、モデル作成フローを図2に示す。

図1. モデル構成

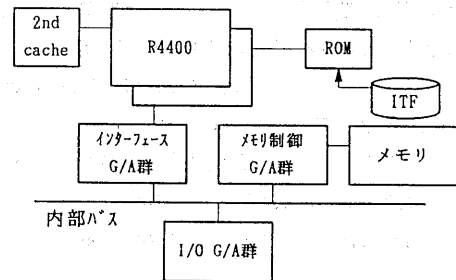
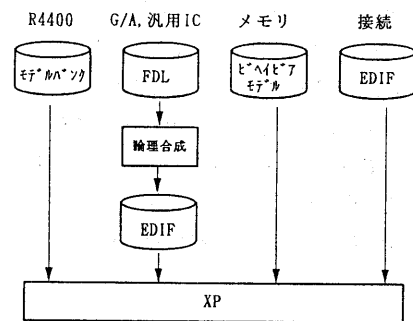


図2. モデル作成



#### (1)G/A

G/Aは、元々社内製ハードウェア記述言語FDL(Functional Description Language)で設計されている。このFDLを論理合成することにより、EDIFネットリストを作成する。

#### (2)メモリ

XP独自の形式であるビヘイビアモデルで作成する。C言語に類似した形式の記述である。

#### (3)R4400-G/A-メモリ間の接続

EDIFネットリストを人手作成する。

#### (4)汎用IC

FDLで機能を記述し、論理合成することによりEDIFネットリストを作成する。

#### 3.2 パターン作成

ボードレベルシミュレーション実行のための入力パターンは2種類ある。

- (1)クロック、バス初期化、R4400初期化用パター

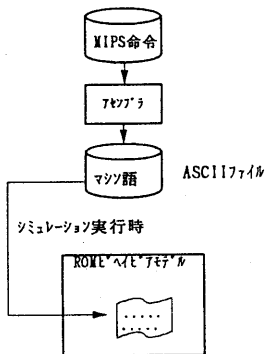
ン

XP独自の形式で作成する。時間列と"0"、"1"の羅列である。

### (2)ITF

ROMに書き込まれるITF。図3に示すように、MIPS命令をアセンブルしてマシン語にし、それをASCIIファイルに格納する。このASCIIファイルは、シミュレーション実行時にROMのビヘイビアモデルに読み込まれる。

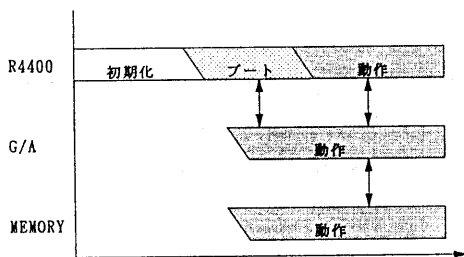
図3. ITF



### 3.3 シミュレーション実行

図4にシミュレーションの流れの概略を示す。

図4. シミュレーションの流れ



#### (1)R4400初期化

R4400の基本動作モードを設定する。実物ではROMの内容を読み込むことにより行うが、シミュレーションではパターンで入力する。

#### (2)ブート

R4400の初期化が終了した後、R4400にリセット信号を入力する。パターンで入力する。

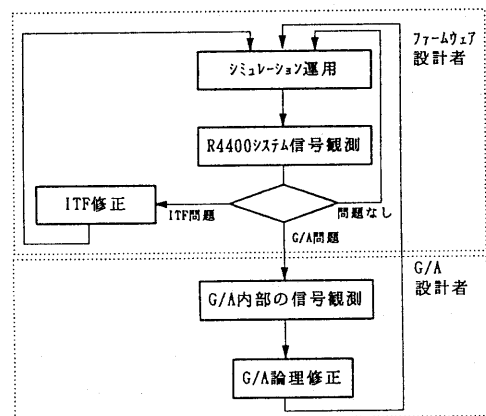
#### (3)動作

R4400にリセット信号が入力されると、R4400がシステム信号をG/Aに出力する。その後、各G/A、メモリが動作を開始する。シミュレーション中、R4400はROMからITFを読み込み、その命令に従って動作する。

### 3.4 シミュレーション結果の検証

3.3で示したシミュレーションの実行は、ボードレベルのシステムの動作のよく分かる者、すなわちファームウェア設計者が行った。図5に、シミュレーションの運用とシミュレーション結果の検証の流れについて示す。

図5. シミュレーションの運用と検証



#### (1)ファームウェア設計者

ファームウェア設計者は、シミュレーションの運用と結果検証を行う。この結果検証は主にR4400のシステム信号を見ることにより行われる。問題が発生した場合は、ITFの問題なのか、G/Aの問題なのかの切り分けを行う。ITFの問題であればITFを修正し、G/Aの問題であればG/A設計者に内容を伝える。

#### (2)G/A設計者

ファームウェア設計者が、検証の結果G/Aの問題であると判断した場合、G/A設計者は、G/A内部の信号のシミュレーション結果を検証する。検証の結果、G/Aのバグであると判断した場合はG/Aの論理修正を行う。

以上のように、ゲートレベルシミュレーションにおいては、ハードウェアアクセラレータXPの効果により、高速なシミュレーションを実現できた。次のステップとして、XPとOptiumをリンクし、R4400モデルバンクとPCIバスVHDLを使用したVHDLシミュレーションを実施した。

## 4 VHDLでのシミュレーション

本章では、VHDLを用いたボードレベルシミュレーションの以下のステップについて述べる。

- (1) ボードモデル作成
- (2) シミュレーションパターン作成
- (3) シミュレーション結果検証

VHDLによるボードレベルシミュレーションでは、PCIバスを含めた検証が可能になったため、より検証の網羅率を上げることができた。

### 4.1 モデル構築

ゲートレベルシミュレーションでは、結果検証をネットリストレベルで行うことになる。ネットリストとFDLの対応をとることは容易ではなく、論理バグが検出された場合の論理修正、すなわちFDLの修正も容易ではない。一方、VHDLシミュレーションでは、結果検証をVHDLレベルで行う。そして、論理バグが検出された場合は、VHDLとFDLを対比して行うことになる。FDLとFDLを変換して作成したVHDLの対応は容易にとることができるため、モデル修正も容易である。

ボードモデルの構成を図6に、モデル作成フローを図7に示す。

図6. モデル構成

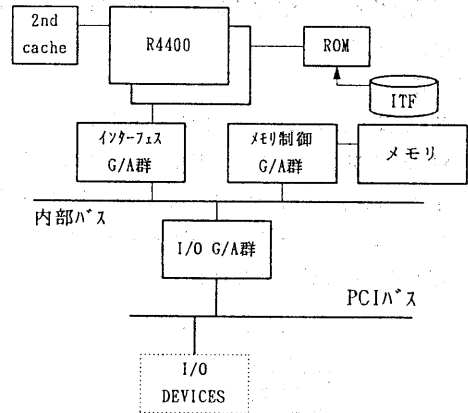
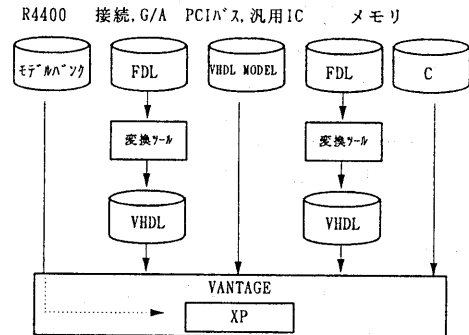


図7. VHDLモデル作成



#### (1) G/A

FDLで記述されたG/Aモデルを変換ツールでVHDLに変換する。変換ツールは、RTLでの変換を行うので、変換前のFDLと変換後のVHDLの対応は容易にとることができる。

#### (2) メモリ

G/Aと同様に、FDLで記述しVHDLに変換するか、C言語で作成し、OptiumのC言語インターフェースを使用する。

#### (3) R4400-G/A-メモリ、PCIバス間の接続

G/Aと同様に、FDLで記述しVHDLに変換する。

#### (4) PCIバス

PCIのメモリリードやメモリライトといった動作は、設計者が定義する。PCIバスモデル独自の形式で記述する。

#### (5)汎用IC

PCIバスモデルと同様にsynopsys社が販売しているVHDL用汎用ICモデルライブラリを使用する。

### 4.2 パターン作成

ゲートレベルシミュレーションと同じ入力パターンを作成する。

#### (1)クロック、バス初期化、R4400初期化パターン

VHDL記述で作成するか、あるいはOptium独自の入力形式で作成する。Optium独自の形式では、時間列と"0"、"1"の羅列で記述する。

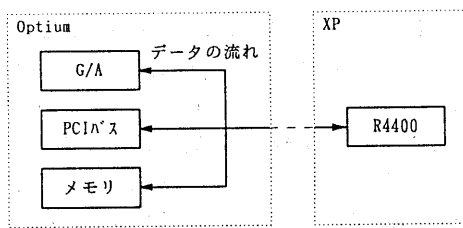
#### (2)ITF

ゲートレベルの時と全く同様に作成する。

### 4.3 シミュレーション実行

シミュレーション実行手順は、ゲートレベルの場合と全く同様である。ただし、図8に示すように、R4400はXP上で動作し、G/A、PCIバス等のVHDLモデルはOptium上で動作するため、XPとOptium間でのシミュレーションデータのやり取りが多く発生する。

図8. シミュレーションデータの流れ



---: ワークステーション上で動作するOptiumと、ワークステーションとバスケーブルで接続されるXPとのデータのやり取り

### 4.4 シミュレーション結果の検証

シミュレーション結果の検証は、ゲートレベルシミュレーションの時と全く同様に行われる。

以上が、VHDLシミュレーションの説明である。VHDLシミュレーションでは、PCIバスモデルを新たに統合することができたため、よりボードモデ

ルとしての完成度が高く、検証の網羅率を上げることができた。また、論理合成前のVHDLで検証を行うことにより、設計上流過程での検証ができるため、論理バグが検出された場合でも論理修正へのフィードバックは小さく済んだ。しかし、4.3で述べたように、XPとOptiumをリンクしたシミュレーションでは、XPとOptium間でのシミュレーションデータのやり取りが多数発生し、このデータのやり取りに処理時間が費やされるため、満足のいくシミュレーション処理速度とはならなかった。

## 5 評価

シミュレーションを実設計に適用した結果について以下に示す。

3、4章で述べた、シミュレーション実行に関する各種データを表4、5に示す。なお、ゲートレベルシミュレーションとVHDLシミュレーションでは、異なるシミュレーションモデルで運用している。

### 5.1 バグ検出状況

ゲートレベルシミュレーション、VHDLシミュレーション共に、LSIのバグを検出している。ITFのバグの検出もできている。従来の手法では、ITFのバグ検出は、シミュレーションレベルでは困難であり、評価用装置あるいは実機による検証で行っていた。ボードレベルでのシミュレーションを実行することにより、LSIの検証と同時にITFの検証も可能となった。

### 5.2 シミュレーション速度

ゲートレベルシミュレーションでは、ハードウェアアクセラレータが効果を発揮し、高速なシミュレーションを実現できた。一方、VHDLシミュレーションでは、XPとOptium2つのシミュレータをリンクして実行するため、シミュレータ間のデータのやり取りが多く発生し、処理速度が低下した。この問題を解決することが処理速度向上の

ための1つの方法であると考える。

表4. ゲートレベルシミュレーション

モデル規模	約1Mゲート(R4400、メモリは除く)
検出バグ	HWバグ(論理バグ) : 1件 FWバグ : 4件 HWバグ(クランプ漏れ) : 2件
シミュレーション速度	10~50 [clk/s](平均15 clk/s)

表5. VHDLシミュレーション

モデル規模	VHDLライン数: 約25万行 (実チップ規模: 約1.2Mゲート。 R4400、メモリ、PCIバスは除く)
検出バグ	HWバグ(論理バグ) : 3件 FWバグ : 4件
シミュレーション速度	5 [clk/s]

## 6 おわりに

今回、フルファンクションのR4400モデルを導入したことにより、従来より一歩進んだボードレベルの検証を可能にした。このシミュレーションは、UNIXサーバUP4800/680等の開発に適用され、装置の高品質化に役立った。また、その後の各種製品開発にも利用されている。

ボードレベルシミュレーションの今後の課題を以下に示す。

### (1)速度改善

現在のシミュレータの能力でのボードレベルシミュレーションでは、システムの初期化程度までしか検証できない。今後は、OS立ち上げまでの検証を目標としたい。そのためには、現在の処理速度を2桁以上改善する必要がある。処理速度向上のためには、シミュレータの改善が必要だが、リンクシミュレーションの場合はシミュレータ間のデータのやりとりを最小限に抑さえるモデルを構築する等、シミュレーション運用側も工夫をするべきであると考える。

### (2)規模対策

シミュレーションモデルの規模増加は、シミュレータの速度以外にも、あらゆる面で影響がある。モデルの規模が増加すれば、その分シミュレーション結果も増加する。膨大なシミュレーション結果を効率的に検証する手段が必要である。

### (3)シミュレーションモデルの入手

実物に近いボードレベルでのシミュレーションを実現するためには、CPU、汎用IC、メモリ等の高精度なシミュレーションモデルのタイムリーな入手が必要不可欠である。我々にとっては、特に、ハードウェア記述言語レベルでリンクできるシミュレーションモデルが必要である。現在、シミュレーションモデルはCADベンダーから各種市販されている。しかし、その種類には限界があるため、これだけに頼るわけにはいかない。CPUに関しては、ハードウェア記述言語による設計方法を採用している半導体メーカーが多いため、ハードウェア記述言語レベルでのモデル入手が可能ではないかと考えている。汎用IC、メモリに関しては、ハードウェア記述言語にリンクしたモデル作成ツールの導入が現実的であると考えている。

今回のボードレベルシミュレーションは、シミュレータをZYCAD社、VANTAGE社の2社から導入し、シミュレーションモデルをsynopsys社から導入して行った。先端装置の開発では、CADツールが設計者の思惑通りに動作しないことが多々起こる。今回も導入当初は各種の問題が発生したが、各CADベンダーと密接にコンタクトを取りながら問題解決にあたった。それが今回のボードレベルシミュレーションの成功原因の1つと言える。CADツールを使用していく上では、ツール導入後のCADベンダーとの協力体制が重要である。