

1チップCPUプロセッサの設計検証(3)
シミュレータ上でのシステムレベル検証

5N-9

三部健¹、近江谷廉人¹、橋詰雅樹²、井関秀行³、原島忠雄¹、山口一良¹

¹三菱電機(株)、²Horizon Research, Inc.、³三菱電機エンジニアリング(株)

1. はじめに

最近の計算機システムに対する要求は高機能化・高性能化が求められ、そのために設計はますます複雑化し、論理設計量は増加している。増加した論理量を一定の物理サイズに格納するためにLSIも集積度を増し、我々が開発した170万Tr. 1チップCPUに代表されるような大規模LSIが登場してきている。そのようなLSIを複数個用いる計算機システムの設計はそれぞれのLSIの単体シミュレーション同様に十分なシステムレベルシミュレーションが事前に必要となり、また最近それが可能となっている。

我々は、複数のLSIで構成されたCPUボード、I/O、サーバプロセッサまでを含んだ計算機システム全体のシステムレベルシミュレータを構築した。

本稿では、構築したシステムレベルシミュレータの概要と構成、実際に行なったシミュレーション方法、適用事例について報告する。

また今回のシステムレベルシミュレーションを行うことにより、ファーストシリコンで、実機上の機能診断プログラムから、システム診断プログラムテストを経て、OSまで動作する事を目標とし、これを実現したことを付け加えておく。

2. システムシミュレータの概要と構成

今回構築したシステムレベルのシミュレータの構成を図1に示す。シミュレータの構成は対象のハードウェア(CPUボードとチャンネル、サーバプロセッサ)とテストハンドラと大きく2つに分かれる。対象のCPUボードは最大構成で170万Tr. の1チップCPU2石、400kGSOG×5石、384MBメモリ

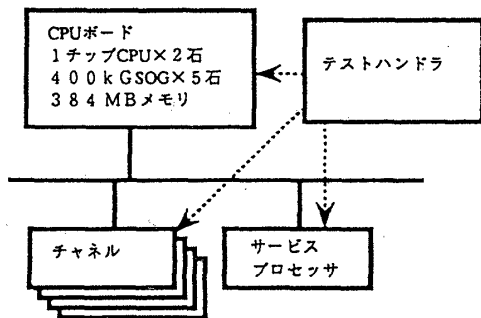


図1 システムレベルシミュレータの構成

Gのゲートアレイを用いた周辺LSI5石、384MBのメインメモリで構成されている。各LSIはモデル開発ステップに応じて、擬似レベル、機能レベル、RTL、ゲートレベルを用意した。また各レベルのモデルのインターフェースを同一にすることにより容易にミックスレベルのシミュレーションが可能となっている。

テストハンドラは次の機能を保持している。

1) 計算機システムの構成のセット

テストハンドラ内のパラメータを変更することにより対象の計算機システムの構成、制御情報を自由に変更することが可能である。

2) デバッグ機能

各LSI内にあるレジスタ、外部バス、内部バス、キャッシュ、TLB等を表示させる機能を持たせることによりデバッグ機能を強化している。会話的コマンドを持たせることにより、手軽にレジスタの内容、メインメモリの内容等を変更することができ、条件を一時的に変え、テストを実行することが可能である。

3) 実行管理機能

テストは図2に示すようなフローを用いて実行される。最初に必要に応じてレジスタ、キャッシュ、TLB等に

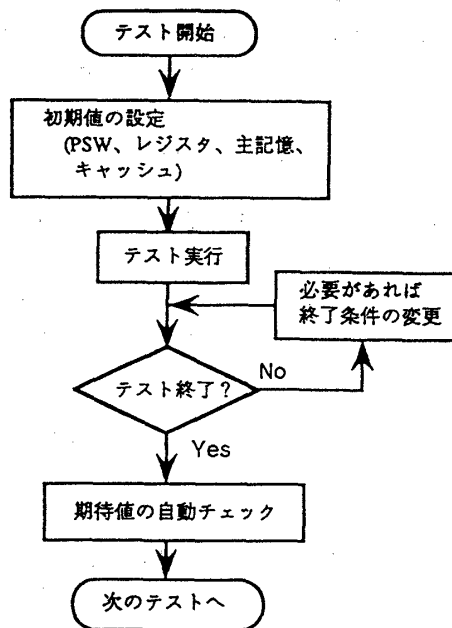


図2 テストケース実行フロー

Design Verification of One-chip CPU Processor (3)
System-level Verification using Logic Simulator
Ken SAMBU¹, Yasuhito OMIYA¹, Masaki HASHIZUME², Hideyuki ISEKI³,
Tadao HARASHIMA¹, Kazuyoshi YAMAGUCHI¹
¹Mitsubishi Electric Corp., ²Horizon Research Inc., ³Mitsubishi Engineering Co.Ltd.

初期値をセットする。次にPSWのセットをし、テストをスタートさせる。テストケースに記述されているテストの終了条件をモニタリングし、テストの終了を判定する。テスト実行にはバッチ機能を持たせ、複数のテストケースを連続実行できるようにした。種々のハングアップを検出する機能を持たせることにより、ハングアップしたときにはテストを終了させ、次のテストケースを実行させる機能を持っている。

4) 解析機能

内部バス、外部バス、アーキテクチャレベルのレジスタをトレースする機能や、キャッシュ、TLBの内容をダンプする機能を持ち、解析するときの情報を出力する。

5) 期待値との比較

テストが終了すると期待値とモデルでの実測値との比較を行い、テストがパスしたか否かを判定する。もし、テストがパスしなかった場合には、バグ解析に必要な情報を自動的にファイルにダンプする。

3. シミュレーションの方法、適用事例

3.1 シミュレーションの方法

今回行ったシミュレーションはトップダウン手法を用い、次のように行った。

1) 擬似レベル

まずLSIの擬似モデルを各LSIの仕様書から作成し、CPUボード上での各LSI間のインターフェースの確認を行った。また、この上でテストハンドラの開発を行った。

2) 機能レベル

次に、機能モデルを作成し、シミュレーションによりテストケースの作成とLSIの詳細仕様の確認を行った。テストケース言語は命令列とテストハンドラに対する指示をマクロアセンブラで記述した。

3) RTL/ゲートレベル

RTL、ゲートモデルは各LSIの設計者から単体テスト終了後にリリースを受ける。そして、リリースされたLSIに関するテストケース、診断プログラムのような客観的テストの順でゲートモデルを用いて検証を行った。この時点でバグが存在しなければ、LSIのレイアウト設計に入る。

バグの解析はRTLを用いて行った。RTLは論理合成ツール、ライブラリに依存せず、解析が行ない易いからである。

従来プロト機デバッグに持ち越されていた、プログラムデバッグ機能、例外処理、エラー処理を完全に検証するテストケースを用いたことによりシステムレベルシミュレーションのカバレッジを向上させている。

最後に全てのLSIのゲートモデルが整ったときには、

確認の意味で全てゲートモデルを用いてシミュレーションを行った。但し、一部のテストケースについては必要性が無いことから機能モデルとの組合せにとどめた。

4) 実機レベル

実機デバッグ時には、実機上のデバッグの環境とシミュレータ上でのデバッグの環境の合わせ込みを行い、システムレベルシミュレータを実機の横に置いた。実機上で不具合が生じた場合には再現するテストケースをシステムレベルシミュレータ上で作成し、シミュレータ上で実機では観測できない内部信号の解析を行なった。

3.2 適用事例と効果

ファーストシリコンで目標のカバレッジを得るために必要なテストケースの数、クロック数の見積り、シミュレーションに必要なマシンパワーの算定を行った。

今回はLSIリリース時に5日で全システムテストが終了するようにシミュレーション専用ワークステーション10台(トータル500MIPS)用意した。ワークステーション上のプロセスのメモリサイズはモデル構成により100~400MB要した。今回用意したテストケースの種類、テストケースの数、クロック数を表1に示す。

トップダウン設計手法にあった検証環境を構築することによりLSIリリース時に集中しがちな検証作業を分散させることができた。目視等による結果のチェックを排除できるように工夫して人手作業を最低限にして、設計不具合の修正時に全テストを流し直すことを徹底し、その結果検証品質を維持できた。

表1 用意したテストケース

テストケースの種類	件数	クロック数
手作り	7500	311万clk
診断プログラム	3500	1500万clk
ランダムテスト(4種)	---	500万clk
例外処理、エラー処理	400	260万clk

4. 終わりに

本稿では170万Tr.を集積した1チップCPUに代表されるような大規模LSIを用いて構成されたCPUボード、I/O、サービスプロセッサまでを含んだ計算機システム全体のシステムレベルシミュレーションの概要と構成、また実際に行なったシミュレーション方法、適用結果について報告した。

システムレベルシミュレーションを事前に十分行なうことで実機上では機能診断プログラムからOSまでファーストシリコンで動作することが確認できた。

テストカバレッジの一層の強化、システム診断プログラムへの取り組み等が今後の課題である。