設 計 自 動 化 60－23
（１９９１．１２．１３）

ＣＰＵコアを内蔵するシステムＬＳＩの論理シミュレーション方法

山田　靖憲*　　天野　亘孝*　　小畑　誠**　　小島　智**

*　（株）日立製作所　システム開発研究所
**（株）日立製作所　半導体設計開発センタ

CPUコアを内蔵するシステムLSIの大規模化と品種数増加に伴い、その設計期間・工数の削減が大きな課題になっている。この課題を解決するには、設計の初期段階で論理シミュレーションを徹底的に実施し、論理不良を取り尽くすことが必須である。本稿では、システムLSIの論理シミュレーションの高速化を目的として、それに内蔵されるCPUコアの論理動作モデル化方法を提案する。本方法により開発した８ビットマイクロプロセッサの論理動作モデルは、ゲートモデルと比べて、モデルの精度が同等で、論理シミュレーションの処理速度が約３５倍高速であった。

A Logic Simulation Method for System LSI's Including a CPU Core.

Yasunori YAMADA*　Nobutaka AMANO*　Makoto OBATA**　Satoshi KOJIMA**

*　Systems Development Laboratory, Hitachi, Ltd.
**　Semiconductor Design and Development Center, Hitachi, Ltd.

Expanding the scale of one system LSI including a CPU core and increasing the number of system LSI series cause the serious increase of design term and human resources. To avoid these problems, it is necessary to remove logic errors as much as possible by carrying out sufficient logic simulations at an early stage of the design. This paper presents a method of creating a behavioral model of a CPU core included in system LSI's in order to accelerate logic simulations. Using this method, we have developed a behavioral model of an 8-bit micro processor. The logic simulation speed of the behavioral model was 35 times faster than that of the original gate model under the same accuracy.

1．はじめに

近年、一つのチップ内にCPUコアやメモリを内蔵して特定用途に使用されるシステムLSIの論理規模と品種数が増大するにつれて、その設計期間・工数の削減が大きな課題になっている。この課題を解決するには、設計の初期段階で論理シミュレーションを徹底的に実施し、論理不良を取り尽くすことが必須である。

ところで、CPUコアは高集積度が要求されるため、従来はゲートレベルで設計されていた。そのため、CPUコアの論理シミュレーションを高速化するために、CPUコアのゲートモデルを論理動作モデルで置換することは重要な課題の一つである。

本稿では、この課題を解決するためのCPUコアの論理動作モデル化方法を中心に述べる。まず、第2章で使用したミックストレベル論理シミュレータの概要を述べ、次に、第3章で8ビットマイクロプロセッサの論理動作モデル化方法を述べ、最後に、第4章で開発した論理動作モデルの評価結果を述べる。

2．ミックストレベル論理シミュレータの概要

使用したミックストレベル論理シミュレータHUSL-MX（Hierarchical and Universal Simulation for Logic systems at MiXed abstraction levels）は、ゲートレベル論理シミュレータHUSL[1),2)]をミックストレベルに拡張したイベントドリブン型のソフトウェア論理シミュレータである。HUSL-MXは論理動作モデル、機能論理モデル、ゲートモデルが混在した論理モデルを受け付け、各論理モデルに応じたディレイ精度で論理シミュレーションを行い、その結果を波形表示で出力する（図1）。

各論理モデルの概要を以下に述べる。

（1）論理動作モデル

論理動作モデルは論理回路の動作を記述したもので、以下の三要素から構成される（図2）。

（a）プライマリ信号線

プライマリ信号線は論理動作モデル間を接続する信号線または論理動作モデルと論理シミュレータ本体との間で信号伝達を行うための信号線であり、入力信号線、出力信号線、入出力信号線の3種類がある。

（b）並列動作ブロック

並列動作ブロックは論理動作モデル内の並列実行単位の論理ブロックであり、論理動作モデルが複数個の並列動作ブロックで構成されている場合、各並列動作ブロックは並列に動作する。各並列動作ブロックの内部は逐次処理または状態遷移で表現され、各状態の内部も逐次処理または状態遷移で表現され、状態遷移の内部の階層表現が可能である。

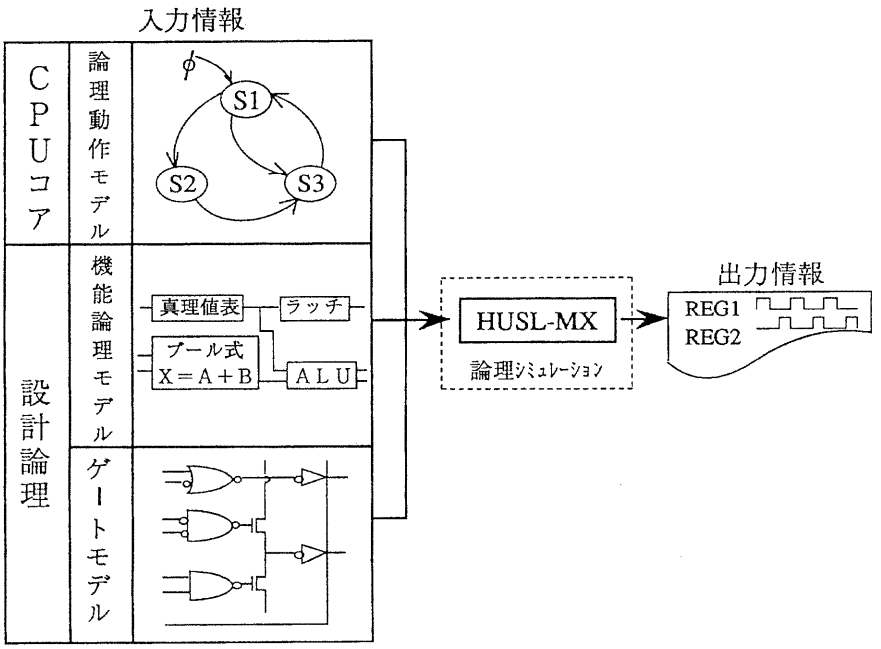

図1　ミックストレベル論理シミュレータＨＵＳＬ－ＭＸの概要

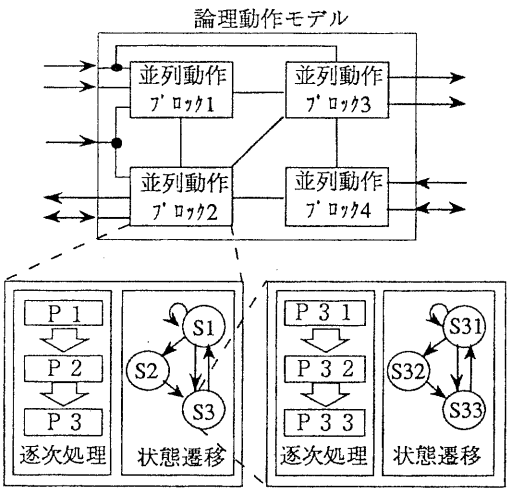

図２　論理動作モデルの構成

（ｃ）並列動作ブロック間信号線

並列動作ブロック間信号線は並列動作ブロック間を接続する信号線である。

この論理動作モデルはＣ言語を拡張したハードウェア記述言語で記述される。すなわち、演算、データ転送、条件判定等の一般動作はＣ言語の仕様で記述され、状態遷移、並列動作ブロック、可変ディレイ等のハードウェア固有の動作は拡張仕様で記述される。この論理動作モデルが取り扱う論理値は２値（０，１）である。

（２）機能論理モデル

機能論理モデルは論理回路の機能を真理値表、ブール式、マクロで記述したものである。ここで、各マクロはその機能を上述のハードウェア記述言語で記述したものが使用される。この機能論理モデルが取り扱う論理値は３値（０，１，Ｘ）である。

（３）ゲートモデル

ゲートモデルは論理回路をゲートやトランジスタ等のプリミティブとそれらの接続関係で記述したものである。このゲートモデルが取り扱う論理値は３２値であり、パスディレイの評価が可能である。

３．論理動作モデル化方法

８ビットマイクロプロセッサの論理動作モデル化方法を以下に述べる（図３）。

（１）論理動作解析

設計データを基に論理動作を解析し、論理動作表を作成する。ここで、使用する設計データはマイクロプログラム、タイミングチャート、論理シミュレーション結果の３種類であり、マイクロプログラムはCPUコアの各命令の実行プロセスを手続きの記述に展開するために使用し、タイミングチャートはバスの制御タイミングや各外部信号線の動作タイミング等を解析するために使用し、論理シミュレーション結果はマイクロプログラムとタイミングチャートの確認や例外処理等の情報補填のために使用する。

論理動作表は、各命令の実行サイクルが進むにつれて、CPUコアの外部入出力端子と内部レジスタの値がどのように変化するかをまとめ、動作シーケンスが同一であるものをグループ化した表である（図４）。この論理動作表は、論理動作モデルをトップダウン設計する場合と異なり、論理シミュレーション結果が元のゲートモデルと波形レベルで一致する高精度の論理動作モデルを設計するために必須である。

（２）状態／状態遷移の定義

論理動作表と外部仕様の状態遷移図を基に状態／状態遷移を定義し、内部仕様の状態遷移図を作成する。外部仕様の状態遷移図はCPUコアのマニュアルに記載されているもので、CPUコアの各動作（状態）の変化（遷移）を外部仕様から捉えたものである。これに対して、内部仕様の状態遷移図は内部動作のタイミングに共通部分が多いものを一つの状態として定義し、それらの遷移関係を表現したものである。本ケースでは、外部仕様の７状態が内部仕様の６状態に減少した（図５）。

（３）モジュール分割

論理動作表と内部仕様の状態遷移図を基に論理動作モデルのモジュール分割を行い、モジュール構成図を作成する。本ケースでは、次の三つの並列動作ブロックと次の二つの関数群に分割した（図６）。

（ａ）CPU制御ブロック

図５（ｂ）の６状態の遷移条件となる各信号線の値を監視し、状態遷移の制御を行う。また、各状態の内部動作の処理を起動する。

（ｂ）命令実行ブロック

命令実行状態において、データバスからの入力信号値に依存して、命令のオペランドとオペコードのデコードを行い、該当するタイミング制御関数を呼び出す。

（ｃ）バス制御ブロック

命令実行や例外処理に依存してバスを制御し、データの読み書きを行う。同時に、リー

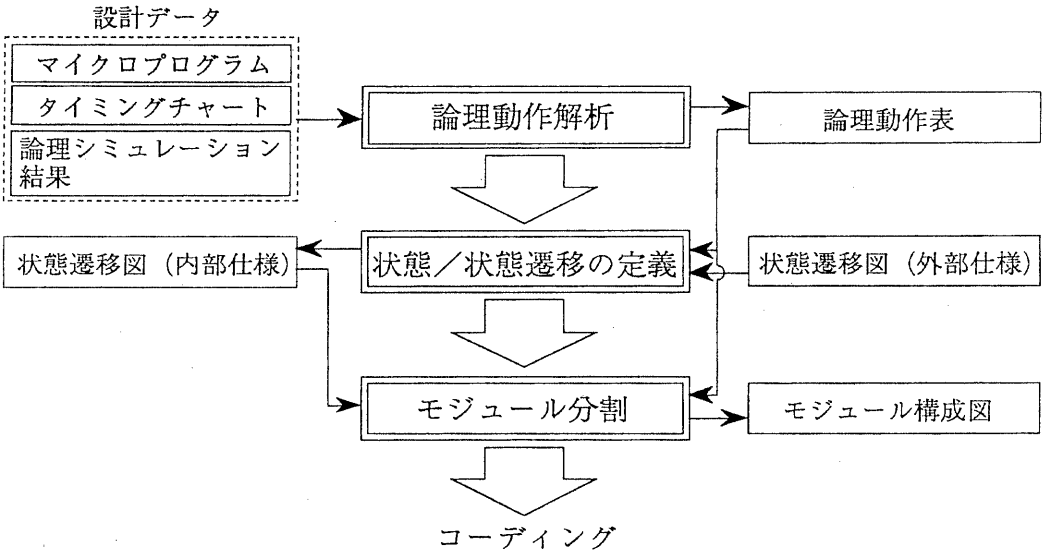

図３　論理動作モデル化方法

実効アドレス計算部

命令ニモニック	サイクル	#	アドレスバス値	データバス値	バスフェーズ	PFP	PC	出力信号 IFRZ	出力信号 WRCN
@(DISP16,Rn)	3	1	PFP値	P.F値	IF	+1	—	0	1
@ABS16		2	PFP値	P.F値	IF	+1	+3	0	1
#IMM16		3	PFP値	P.F値	IF	+1	+1	0	1
@(DISP8,Rn)	2	1	PFP値	P.F値	IF	+1	—	0	1
@ABS8		2	PFP値	P.F値	IF	+1	+3	0	1
#IMM8									
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

（注）
PFP値：プリフェッチポインタの値
P.F値：プリフェッチポインタの示すアドレスのデータ
IF：Instruction Fetch
PC：Program Counter

図４　論理動作表の例

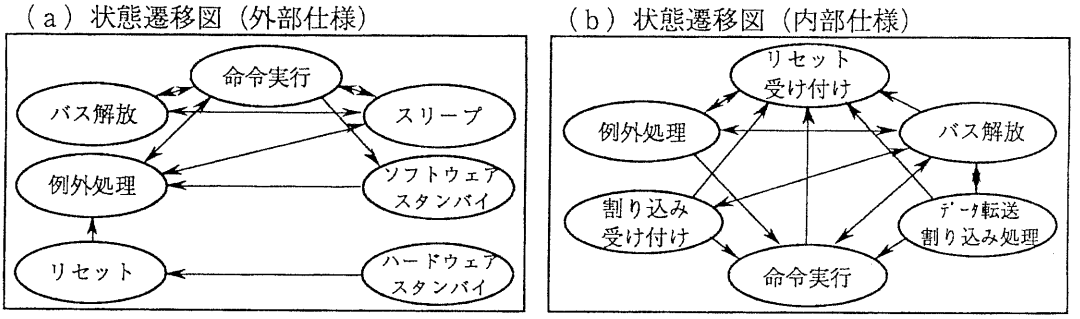

図５　状態遷移図

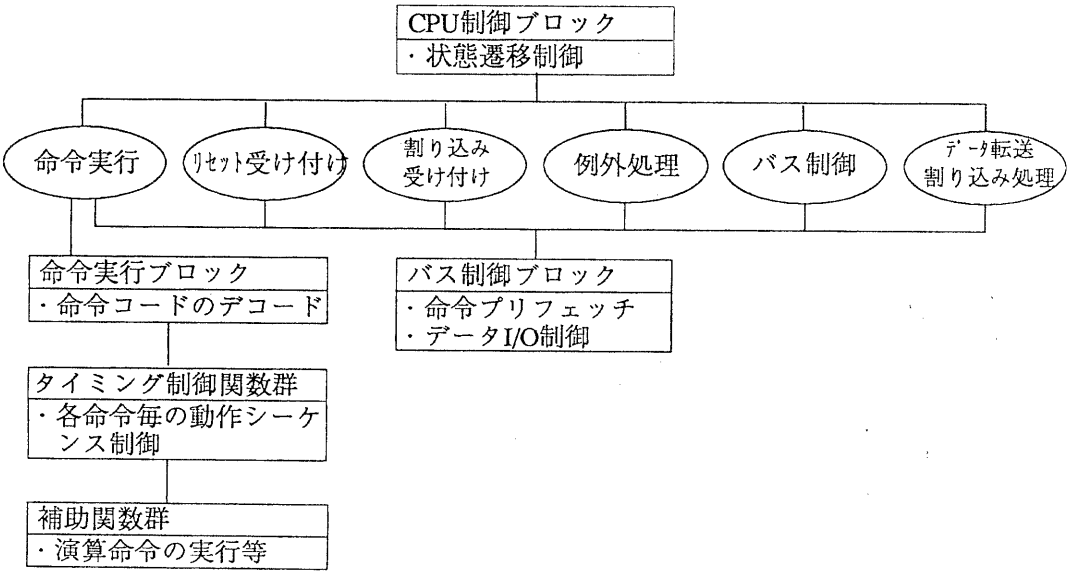

図６　モジュール構成図

ド、ライト等のストローブ信号も制御する。

（d）タイミング制御関数群

タイミング制御関数は論理動作表でまとめた動作シーケンスを各々関数化したものである。各関数は呼び出し時のクロック順に従って各レジスタ値の更新や出力信号の制御を行う。

（e）補助関数群

補助関数は演算命令の演算処理用の関数やモジュール内で共通に使用される関数である。

4．論理動作モデルの評価結果

上述の論理動作モデル化方法に基づき、８ビットマイクロプロセッサの論理動作モデルを開発した。その評価結果は以下のとおりである。

（1）記述量

記述量はゲートモデルの７．５ＫＳに対して論理動作モデルが５．５ＫＳであった。ランダム論理部の記述量は約１／３に減少したが、マイクロプログラム部分の記述量はほぼ同等であった（表１）。

（2）精度

論理動作モデルは論理シミュレーション結果が元のゲートモデルと波形レベルで一致した。それゆえ、論理動作モデルの精度はゲートモデルと同等である。

（3）処理速度

論理動作モデルの論理シミュレーションの処理速度は約３５倍（対ゲートモデル比）であった。

5．おわりに

表1　記述量

モデル区分	記述量		
	ランダム論理	マイクロプログラム	合計
ゲートモデル	2.8KS	4.7KS	7.5KS
論理動作モデル	0.9KS	4.6KS	5.5KS

システムLSIの論理シミュレーションの高速化を目的として、それに内蔵されるCPUコアの論理動作モデル化方法を提案した。本方法により開発した8ビットマイクロプロセッサの論理動作モデルは、ゲートモデルと比べて、モデルの精度が同等で、論理シミュレーションの処理速度が約35倍高速であった。

今後は、VHDL[3),4)]、UDL/I[5)]等の標準化が推進されているハードウェア記述言語を使用したCPUコアの論理動作モデルの開発を進める予定である。

謝辞

本研究の機会を与えていただいたシステム開発研究所の堂免信義所長と久保隆重部長（当時）ならびに半導体設計開発センタの小澤時典部長、また、本研究内容について有益なご意見・ご討論をいただいたシステム開発研究所の新舎隆夫主任研究員と半導体設計開発センタの浅野弘道主任技師に深謝いたします。

参考文献

1) Takahashi, T., Kojima, S., Yamashiro, O., Eguchi, K. and Fukuda, H. : An MOS Digital Network Model on a Modified Thevenin Equivalent for Logic Simulation, Proc. of 21st DA Conf., pp. 549-555 (1984).
2) Eguchi, K., Yamashiro, O., Kojima, S., Takahashi, T. and Fukuda, H. : A Logic Simulation Technique for Gate/Transistor Level Circuits with Precise Delay Estimation, ICCAD-84 Digest of Technical Papers, pp. 200-202 (1984).
3) Moe Shahdad : An Overview of VHDL Language and Technology, Proc. of 23rd DA Conf., pp. 320-326 (1986).
4) James R.Armstrong : CHIP-LEVEL MODELING with VHDL, Prentice-Hall Inc. (1989).
5) 唐津他：論理合成時代のハードウェア設計用標準言語：UDL/I, 信学論, Vol J74-A, No. 2, pp. 170-178 (1991).