

ハードウェアアーキテクチャシミュレータ
PALSIMの構成と実現

6Q-3

芳井 晶* 高橋 隆一**

* 日本電気技術情報システム開発株式会社

** 日本電気株式会社 C&Cシステム研究所 応用システム研究部

1. はじめに

半導体技術の進歩により、機能/論理設計初期段階のCAD化が重要な課題となっている⁽¹⁾。ハードウェア記述言語を用いることは文書化、設計自動化等の点においても多くの利点を持っている。シミュレータPALSIMはハードウェア記述言語PALSEQ⁽²⁾によって表現されたフローチャートや状態機械の動作を追跡するための会話型ハードウェアアーキテクチャシミュレータである⁽³⁾。

本稿ではPALSIMの構成と実現について述べる。

2. シミュレータPALSIM

ハードウェア記述言語PALSEQでは、状態機械をオートマトン記述、フローチャートをRTシーケンス記述、全体のモジュール構成をシステム記述で表現する。

PALSIMは36個のコマンドを有し、ファシリティ、状態機械の状態のトレース、条件付きトレース、状態、制御の分布状況参照、レジスタの各ビット毎の更新回数表示、参照回数表示、値の参照と書き換え、初期値の設定(レジスタの値が0であるか未定義であるか)、記述に用いられた論理式の評価、

ダンプ、シミュレータの内部状態のセーブ等を行うことができる。また、PALSIMはシミュレーションを行わずに静的にPALSEQによる記述のエラーチェックを行うこともできる。静的にエラーチェックされる項目は、ファシリティの未定義、ワード、フィールドの指定の誤り、演算子のオペランドのビット巾の違い、状態遷移先の未定義、制御の飛び先の未定義等である。

本システムはOS・UNIX上で開発され、その上で稼働する。

3. PALSIMの構成

PALSIMは、「模擬系」と「検証系」の2つの系で構成されている。「模擬系」は、必要最小限のエラーチェックを行いシミュレーションのためのロードモジュールを作成する。このロードモジュールを実行することによりシミュレーションが行われる。これに対して「検証系」は、ロードモジュールの作成を行わずに高速にPALSEQとして正しい記述か静的に調べる。

図1にPALSIMの構成を示す。

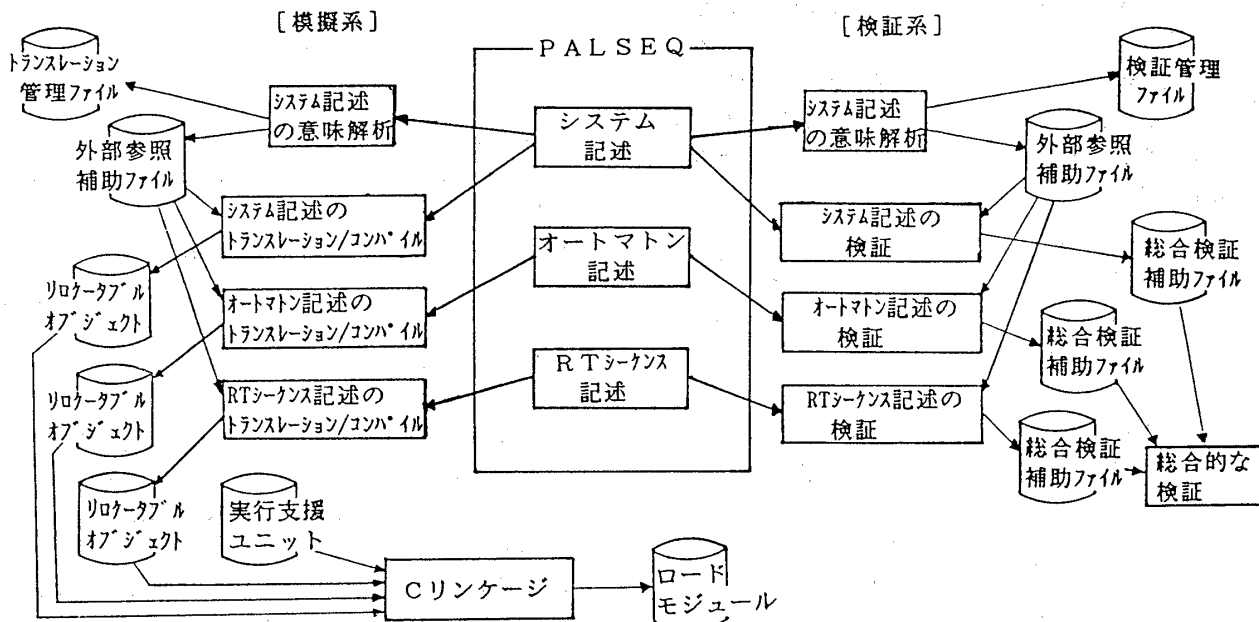


図1. PALSIMの構成

Structure and implementation of hardware architecture simulator PALSIM

Akira YOSHII* Ryuichi TAKAHASHI**

* NEC Scientific Information System development Ltd.

** C&C Systems Research Laboratories NEC Corporation.

模擬系は、オートマトン記述を状態機械の状態単位に、またRTシーケンス記述をフローチャートを構成する一連の処理単位にプログラミング言語Cの関数に翻訳し、実行支援ユニットとリンクして一つのロードモジュールを作成する。実行支援ユニットとはシミュレーション用の関数や演算用の関数のリロケータブルオブジェクトである。この過程においてはCのコンパイルに時間がかかるため、一度ロードモジュールが作成された後に記述に修正があった場合に、その記述だけのトランスレーション/コンパイルが行われるようにしている。図1のトランスレーション管理ファイルはこの管理を行う。このファイルは、UNIXのファイル管理コマンドmakeが参照するmakefileである。外部参照補助ファイルは、各記述をCの関数に翻訳する過程でシステム記述との宣言の矛盾を検出するために使用される。

検証系は、各記述毎に静的なエラーチェックを行いながら総合検証補助ファイルを出力する。総合検証補助ファイルは、PALSEQによる記述が全て揃わないと検出できない種類のエラーチェックを行うために使用される。外部参照補助ファイルおよび検証管理ファイルは模擬系の場合と同様の目的で使用される。

4. PALSIMの実現

PALSIMの開発では、トランスレータ部分の作成の信頼性の向上と工数の削減を目的としてUNIX上のコンパイラ・コンパイラyaccと字句解析プログラムジェネレータlexを用いた。ハードウェア記述言語PALSEQは、予約語が68個、文法としてシステム記述、RTシーケンス記述、オートマトン記述それぞれ39、64、57個の生成規則を用いたLALR(1)で定義される(ただし、同一の非終端記号に対する生成規則は一つに数えた)。図2にトランスレータの作成方法を示す。

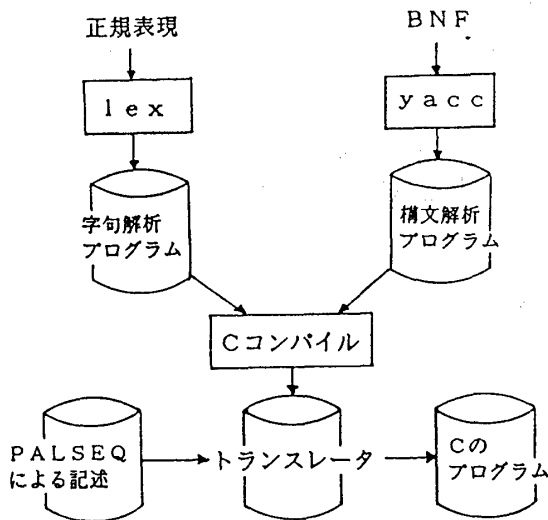


図2. トランスレータの作成方法

PALSIMではシミュレーションのコマンド解析、模擬系起動のコマンドの解析、検証系起動のコマンドの解析等、構文解析、字句解析を必要とするプログラムモジュールにyaccとlexを広範に使用したためコマンドの修正も容易だった。

表1に、PALSIMを構成するプログラムモジュールのライン数(有効部)を示す。本システムの開発工数はおよそ2.5人年であった。シミュレータの速度としては1クロックあたり1モジュールにつき約4ミリ秒である。

表1. 各モジュールの大きさ

	モジュールの機能	ライン数(K)
模 擬 系	システム記述の意味解析	1.2
	システム記述のトランスレーション	4.2
	オートマトン記述のトランスレーション	4.7
	RTシーケンス記述のトランスレーション	5.1
	各トランスレータの共通関数	2.1
	実行支援ユニット	15.0
	小 計	32.3
検 証 系	システム記述の意味解析	1.2
	システム記述の検証	3.8
	オートマトン記述の検証	5.0
	RTシーケンス記述の検証	5.5
	総合的な検証	0.5
	各検証の共通関数	1.0
	小 計	17.0
	合 計	49.3

5. おわりに

ハードウェアアーキテクチャシミュレータPALSIMの構成と実現について述べた。

PALSIMは、CCITT ADPCMのハードウェアアーキテクチャ評価に試用された。

参考文献

[1] Takahashi, R, Yoshimura, T, Goto, S: "A VLSI Architecture Evaluation System" International Conference on Computer Design (ICCD'86) October (1986)

[2] 高橋: 「VLSIハードウェアアーキテクチャ設計言語: PALSEQ」 信学技報 CAS84-133 (1984)

[3] 高橋: 「ハードウェアアーキテクチャシミュレータ: PALSIM」 信学技報 SSD85-79 (1985)