

構造化分析法による LSI 上位仕様からの性能評価法の検討*

5B-6

長沼 次郎 小倉 武 星野 民夫†

NTT LSI 研究所‡

1 はじめに

近年、LSI の大規模化、複雑化に伴い、設計 TAT も増大している。設計 TAT 増大の一因は、LSI 設計の流れの最上流に位置する仕様記述、理解の不完全性、困難性にある。このため、我々は LSI 設計の短 TAT 化を図るため、構造化分析法に基づく LSI 上位仕様の記述、検証法の検討を進めてきた [1]。特に、論理型言語のアルゴリズムックデバッグ法を適用した効率的な上位仕様記述の検証（誤り発見）手法を提案した [2]。

本稿では、構造化分析法による LSI 上位仕様レベルでの性能評価を行うため、検証用の記述（論理型言語）を拡張 DCG によるプログラム変換により、性能評価用の記述へ自動変換し、これにより、実行ステップ数、実行時間、実行時の動的資源量等の性能評価項目を算出することを検討した。

2 システム概要

本仕様記述・検証・評価システムの全体構成を図 1 に示す。本システムは、構造化分析法 [3] を支援する上流 CASE ツール (Soft DA/SA ツール [4]) に、検証用ツール [2]、性能評価用ツールを付加した構成となっている。性能評価用ツールでは、検証用の記述を性能評価用の記述へ自動変換し、各種性能評価項目の算出を行う。

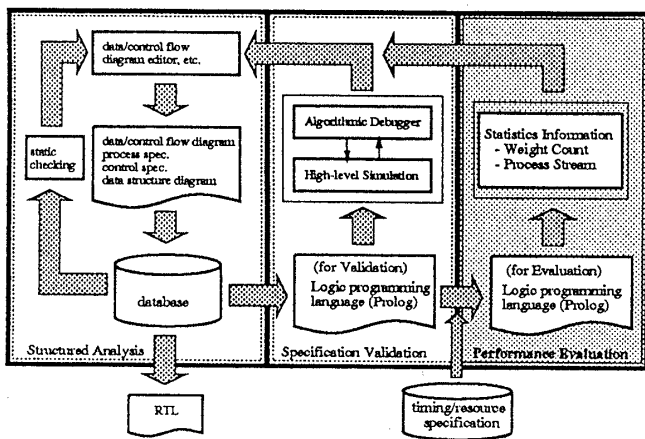


図 1: 仕様記述・検証・評価システムの全体構成

*Performance Evaluation in High Level LSI Design Specification Based on Structured Analysis Method

†Jiro Naganuma, Takeshi Ogura, and Tamio Hoshino

‡NTT LSI Laboratories

3 性能評価法

3.1 概要

構造化分析法では、対象システムの振舞い（動作）を、図的なシンボル（データフロー図等）を用いて、階層的な逐次処理、並列処理として表現する。仕様記述から変換された検証用の記述 [2] を用いることにより、仕様の持つ対象システムの動作がシミュレーションできる。仕様の階層と検証用の記述 (Prolog[5]) の階層的な実行との関係を図 2 に示す。

仕様の各階層でのプロセスと Prolog の階層的な実行（述語呼出し）は、“1 対 1” に対応付けられており、これにより、Prolog 実行時の述語呼出しを観測することにより、本来の仕様に基づく対象システムの実行時のプロセスの起動シーケンスが得られる。得られる起動シーケンスは、対象システムの逐次、並列動作が含まれており、これから各種性能評価項目が算出できる。

3.2 性能評価の項目

構造化分析法による LSI 上位仕様の性能評価の項目として、実行ステップ数、実行時間、実行時の動的資源量等を算出する。このため、各プロセス実行時の起動シーケンスの抽出、実行時間、資源量等の重みの加算を実現する。このとき、構造化分析法では、階層的に記述され

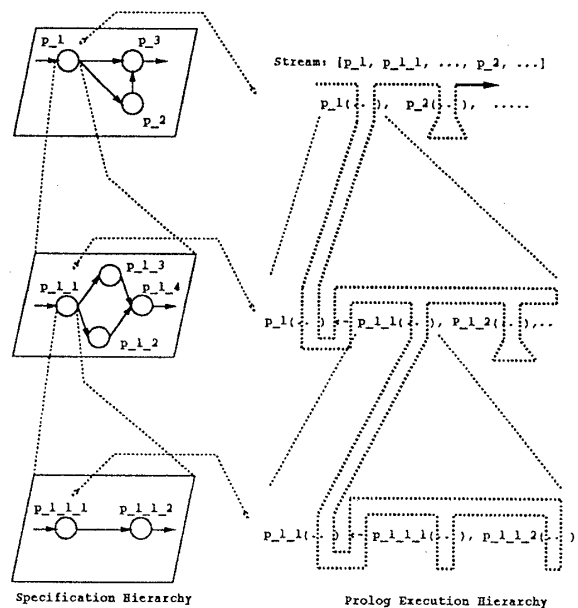


図 2: 仕様の階層と Prolog の階層的な実行

たプロセスの逐次処理、並列処理を表現しており、複数プロセス実行の重なりも考慮して、起動シーケンス抽出、重み付けによる性能評価を行うことができる。

3.3 性能評価ツール

起動シーケンスの抽出、重みの加算を実現するための性能評価用の記述を得るため、検証用の記述をベースとして、起動シーケンスの抽出、重みの加算用の記述要素を付加する。これらの記述要素を付加するため、拡張 DCG [6] と呼ばれる Prolog のマクロ記法を用いて、検証用の記述から性能評価用の記述への変換を自動化した。

記述要素を付加するための拡張 DCG 記述は以下のものである。

(1) 重み (weight) の加算

```
acc_info(weight,I,In,Out,(Out is In+1),0, _).
           重み変数      加算      初期値 最終値
```

(2) 起動シーケンス (stream) の抽出

```
acc_info(stream,I,Out,In,Out=[I|In], _, []).
           シーケンス変数 挿入      初期値 最終値
```

(3) 参照述語定義

```
pred_info(process_name, 4, [weight, stream]).
...      プロセス名 引数の数 対応する変数リスト
```

(4) 一般形定義

```
head ->> [W]:weight, [F/A/N]:stream, body.
...      重み追加 (時間等) シーケンス追加 (プロセス ID)
```

検証用記述から性能評価用記述への変換過程を図 3 に示す。検証用記述は、まず上記の性能評価のための拡張 DCG に変換し、次に拡張 DCG 用プリプロセッサ [6] により、目的の性能評価用の記述に変換する。性能評価のための拡張 DCG への変換過程で、各プロセスの実行時間、資源量等の重みを参照する。拡張 DCG への変換プログラムは、Prolog で約 200 行で記述されている。

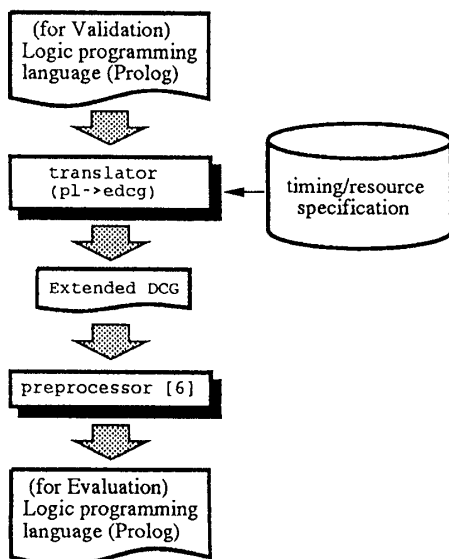


図 3: プログラムの変換過程

4 適用実験

仕様記述・検証の実験の対象として、(1) 通信制御プロセッサ (USART) [7]、(2) 簡単な 8 ビットプロセッサ (CPU8) [8]、(3) Prolog マシン (ASCA) [9] を取り上げた。仕様を Soft DA/SA ツールで記述し、検証用記述 (Prolog) に変換し、さらに、評価用記述に変換した。仕様の記述量、変換された検証用の記述量を表 1 に示す。評価用の記述量は、引数の数、ボディゴールの数は異なるが、節の数は検証用記述と同量である。

変換された評価用記述を実行することにより、対象仕様の実行時における起動シーケンスの抽出、重みの加算が行われ、これらにより、実行ステップ数、実行時間、実行時の動的資源量が算出できた。なお、評価用記述のプログラムの動的な振舞いは、検証用記述のそれとまったく同一である。

表 1: 対象の仕様記述量

	number of hierarchical processes	number of bottom processes	number of transformed Prolog clauses
USART	6	21	109
CPU8	29	28	160
ASCA	8	17	191

5 おわりに

構造化分析法による LSI 上位仕様からの性能評価法として、検証用の記述 (論理型言語) を性能評価用の記述へ自動変換し、これにより、実行ステップ数、実行時間、実行時の動的資源量等の性能評価項目が算出できることを示した。今後、同様な手法を用いて各種統計情報の算出を試みるとともに、構造化分析法を中心としたハード/ソフトコデザイン環境を構築していく。

参考文献

- [1] 長沼他, “構造化分析法を用いた LSI 仕様記述・検証法の検討,” 情処学会 DA シンポジウム '92, Aug. 1992.
- [2] J. Naganuma et al., “High-Level Design Validation Using Algorithmic Debugging,” ED&TC'94, Mar. 1994.
- [3] D.J.Hatley and I.A.Pirbhai, “リアルタイムシステムの構造化分析,” 日経 BP 社, 立田監訳, 1989.
- [4] 磯田他, “設計情報とコードの一体管理方式に基づくソフトウェア開発支援システム (Soft DA/SA),” NTT R&D, Vol.38, No.11, 1989.
- [5] W.F.Clocksinn and C.S.Mellish, “Programming in Prolog,” Springer-Verlag, 1981.
- [6] P. Van Roy, “Can Logic Programming Execute as Fast as Imperative Programming?,” doctoral dissertation, Tech. Rep. UCB/CSD 90/600, UCB, 1990.
- [7] “Microcommunications Handbook,” Intel, 1988.
- [8] “論理設計 CAD に関する調査報告書,” 日本電子工業振興協会, 1986.
- [9] J. Naganuma et al., “High-Speed CAM Based Architecture for a Prolog Machine (ASCA),” IEEE Trans. Comput. vol. 37, no. 11, pp. 1375-1383, Nov. 1988.