

## 動作記述からのテストデータ自動生成について

## 4M-4

蛭田 規之 来山 康治 島田 章  
(沖電気工業株式会社)

## 1. はじめに

現在我々は、動作記述を元にそのすべてを動作させるテストデータを生成するプログラムを開発中である。現在プロトタイプ作成により実験中であるが、このプログラムの概要を紹介する。

## 2. 開発の動機

ハードウェアの設計においては、動作記述を元にアーキテクチャの設計、機能設計などを行うが、これらの設計結果が元となる動作を網羅しているかを調べるためのテストデータは人手により作成しているため作成工数が大きくさらに品質が安定しない。

そこで、本プログラムでは、動作記述を元に、その動作のすべてのパスを活性化するテストデータを自動生成し、動作テストの品質の確保を目指している。

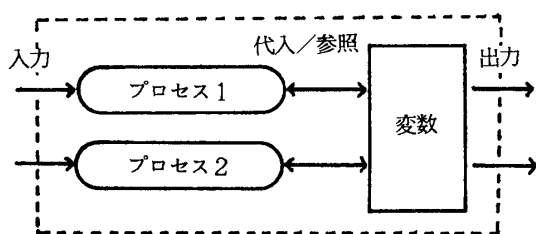
## 3. 機能の概要

## 3.1 入力

主な入力は以下のとおりである。

## a. 動作記述

入力となる動作記述は、繰り返し実行される手続きによるハードウェアプロセスの表現とその手続きで使用する変数(レジスタ)の宣言等からなる。(メモリは扱えない)概念を図に示す。



< 図1. 動作記述の概念図 >

動作のタイミングは、唯一つのクロックに同期するものとする。それぞれのクロックでは前半に条件判定、後半に代入操作を行う。複数プロセスの記述が可能である。

## b. 制約記述

モジュールの入力端子に対して、入力データの値の設定または、範囲の指定ができる。

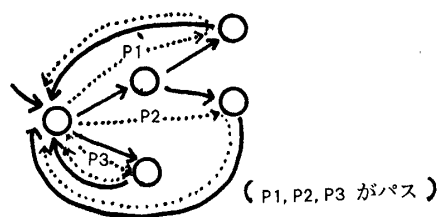
## c. テストデータ

人手作成したテストデータ。プログラムは、不足分のテストデータを付加する。

## 3.2 パスの定義

今回は、パスを以下のように定義する。

”パスとは、動作記述中最初の記述から各分岐を経て、終了点に達するかまたはループで最初に戻るかのどちらかまでとする。途中のループについては、必要なだけ繰り返しを行うが、入力待ちのループは、1回行えば十分とする。”



< 図2. パスの定義 >

## 3.3 機能

## a. 動作記述のチェック

シンタックスのチェック及びデッドロック等の簡単なチェックを行う。

## b. テストデータの生成

生成されるテストデータは、初期状態から始まり、すべてのパスを実行するまでの動作を行うテストデータであるが、次の2つの機能を付加してある。

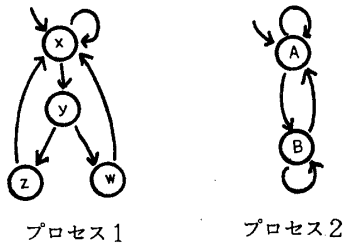
イ: オペレータに対する試験系列が指示されていれば、そのオペレータに試験系列を印加できる(最小の)ループを探し出して、繰り返し実行する。

ロ： 指定段数（指定があれば）のパスの組み合わせを網羅する系列を生成する。

### 3.4 処理概要

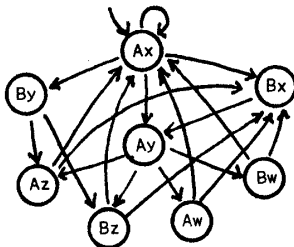
処理は以下のように行っている。

- a. シンタックスチェック
- b. 内部データ作成  
動作記述の条件分岐等を表現した、有効グラフの形式である。  
ハードウェアプロセス毎に生成する。



< 図3. 内部データの生成 >

- c. 複数のグラフのマージ  
複数あるグラフを組み合わせて全体のシーケンスを表現する1つのグラフを作成する。  
この全体のグラフを作成する際には、静的にわかる範囲で各ハードウェアプロセスの実行の組み合わせのチェックを行っている。



< 図4. 図3をマージした例（最悪例） >

- d. デッドロック等の簡単なチェック  
3. で作られたグラフの形態を用いて起こりうる異常状態をチェックする。
- e. テストデータの生成  
全体のグラフに対してパスを定義して、テストデータの生成を行う。生成は動的な変化を考慮し、インタプリティブに行っている。

### 4. 結果

実験結果の一例を示す。  
入力は、2つのプロセスが入力ポートと内部

変数を共有している例題で、共有変数への書き込みは相互排除の処理がされている。それぞれのプロセスの分岐の最大の深さは7段、分岐の最大の幅は6である。生成されたデータは、125サイクルとなった。

```

caab c D E <-ピン名
l12
k

(コメント) 1111 i o o <-入力/出力
↓
00000 c177ffff xx x
1 c172ffff xx x
2 c174ffff xx 1
3 c173ffff xx 1
4 c171ffff xx 1
5 c170ffff xx 2
6 c177ffff xx 2
7
    
```

< 図5. 出力例 >

### 5. 考察

例に示した2つのプロセスからなる記述に対して、全パス活性のデータを人手で作成したところ、87サイクルで生成できた。分岐の選択が、現在は、記述に現われた順であるためこの様な結果になったと思われる。

### 6. 今後の課題

今回は、全パス活性の立場からデータを生成したが、動作記述の検証を行うには動的な面を考慮しなければならないだろうと思われるが、この点については今後の課題としたい。  
さらに、現状では検証データとしては不完全なため、生成データの確認の為のマンマシンインタフェースについて検討中である。

### 参考文献

- [1] 柳沢：“プログラムテストに用いるパスジェネレータの作成法”、情報処理学会研究会報告、ソフトウェア工学63-4、1988
- [2] 小林、若林、脇村：“方式DAの機能試験用テストデータ自動生成法”、NTT研究実用化報告、Vol. 36, No. 10, 1987