

# データフローマシン "TOPSTAR-II" による論理シミュレーション

2J-8

秋元 勲 (富士通) 深沢友雄 栗原 謙  
 鈴木達郎 田中 寛彦 元 嗣 達 (東大)

## §1. はじめに

従来のコントロールフロー方式の計算機による論理回路のシミュレーション作業は、対象となる論理回路のネットワークを一次元化したプログラムで記述せざるを得ないため、回路そのものを自然な形で表現し実行することが、難かしかった。

データフロー制御方式では、プログラムそのものを論理回路のネットワークで記述できるため自然な形での回路表現が可能であり、複雑な論理回路のプログラムも容易に作成することができる。

又、データフロー方式の大きな特徴である並列処理性を十分に生かし、高速にシミュレーションを実行することができ、可能性もある。もともと論理回路の動作を並列処理することは自然な方法であり、この点からも論理シミュレーションはデータフローで処理するのに適していると言える。

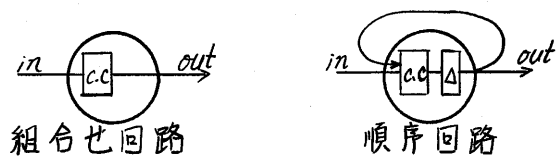
## §2. システムの概要

現在、TOPSTAR-IIで稼働しているシミュレーション用システムプログラムによる制御方式の特徴をあげると以下のようなになる。(参考文献(1), (2))

- (1) バッファ制御を行ない論理回路のループ構造を認める。
- (2) クロックはシステムが管理する。
- (3) 任意の大きさの回路要素を1モジュールとして1ノードに割り当てる。

- (4) 転送回数軽減のため変化しなかったデータは送らない。
- (5) ノードの出力を自分自身に返すことでノードに記憶をもたせる。

言語Cを用いて対象となる論理回路を記述し、11くつかのノードに分けて実行する。各ノードは図1に示すように時間遅れのない組合せ回路、又は時間遅れのある順序回路で構成される。



c.c: Combinational circuit  
 Δ: time delay  
 図1 ノードの構成

## §3. コンパイラ

これまでは、マクロアセンブラを用いて各Cモジュールの制御テーブルを作成するだけで十分であったが、扱う回路規模が大きくなるに従い、

- ・データのイニシャライズに手間がかかる。
- ・ネットワーク相互のチェックが、必要になる。

などの問題が発生してきた。

これらの問題を解決するために、制御テーブルを自動生成するコンパイラを作成した(言語Cを用いて記述)。

ユーザーはコンパイラへのソースとして、ノード間の接続情報や、ノードのCモジュールへの配置情報、記憶回路の初期値などを記述するだけで、コンパイラがこれをチェックして、必要

