

2J-9

# データフローマシン "TOPSTAR" による論理回路シミュレーション

深沢友雄 栗原 謙 鈴木達郎  
田中英房 元園 達 (東大 工学部)

## §1 データフローと論理シミュレーション

従来の制御フロー計算機で論理シミュレーションを行なう場合、

- (1) 回路のネットワークを一次元で展開してプログラムする。
- (2) コンパイラによって高級言語(例えばPL/I)から成るオブジェクトを作成する。
- (3) 高級言語をコンパイルして実行する。

といふ手順が必要となる。

一方データフローマシンでは、プログラムをネットワークで記述できる為回路のネットワークと、データフロープログラムとの自然な対応により、シミュレーションの対象となる回路を記述できる。従って、データフロー言語で回路を容易に表現することができ、それをコンパイル、実行すれば論理シミュレーションが行なえる。

又、論理回路においては、各回路要素が各時刻において同時に動いているという意味で、全ての要素が並列に動作している。従って、データフローマシンの特徴の一つである並列処理性を十分に生かして高速に論理シミュレーションを行なうことができる。

## §2 論理シミュレーション用システムプログラム

論理シミュレーション固有の問題点の解決の為に、既存のシステムプログラムVIを改訂して新たなシステムプログラムを作成した。

すなわち、

- (1) ループ構造を認める。その為、バッファ数は複数となるが、データの順序を保存する様なバッファ制御を行なう。(逆越し禁止)
- (2) シミュレーションの対象とする回路のクロックは、システムが管理する。
- (3) 任意の規模の回路要素を一つのモジュールとしてデータフローの一つのノードに割り当てることが出来る。
- (4) 変化のないデータは送らない。

等を考慮した。

又、ノードに記憶を持たせる必要も生じるが、(例えばレジスタなど)これは、ノードの出力を自分自身に戻すループによって解決した。

又、シミュレータの出力制御の為、或る程度の集中制御が必要となるが、これは、全ノードの出力データを入力とする出力専用のノードを設けることにより解決した。

## §3 シミュレーション例

シミュレーションの対象とする回路は、ゲートレベルより上のレベルで設計された同期回路とする。

プログラムの記述は、ノードの相互関係等をCM内のテーブルに記述することによって行なう。

例として、簡単な、設計途中段階の計算機のシミュレーションを行なった。その計算機の状態遷移図を図1に、回路構成を図2に示す。(参考文献(1))

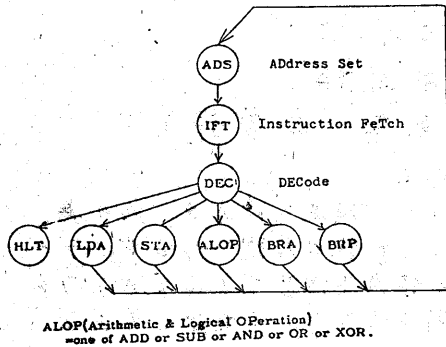


図1 状態遷移図

§4 評価・検討

シミュレーションの場合、各ノードは全て並列に実行可能となるので、PM数（ノード数）を増やすまでは並列性を取出して計算できる。図1、図2の例から、PM数を1〜3台にした時の1クロック当りのシミュレーション時間の実測値を図3に示す。

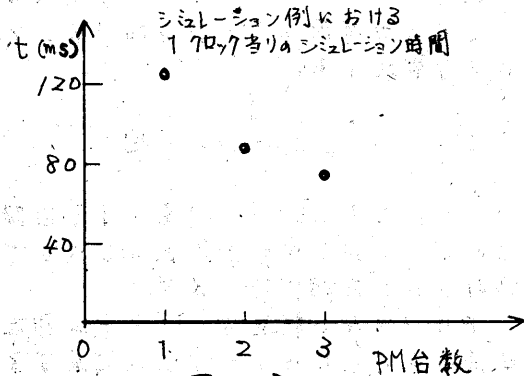


図3

本シミュレーション例では、対象とする回路が簡単な為、PMの負荷が軽すぎ、オーバーヘッドが目につくが、複雑なネットワークになる程、並列性を十分に生かしたシミュレーションが可

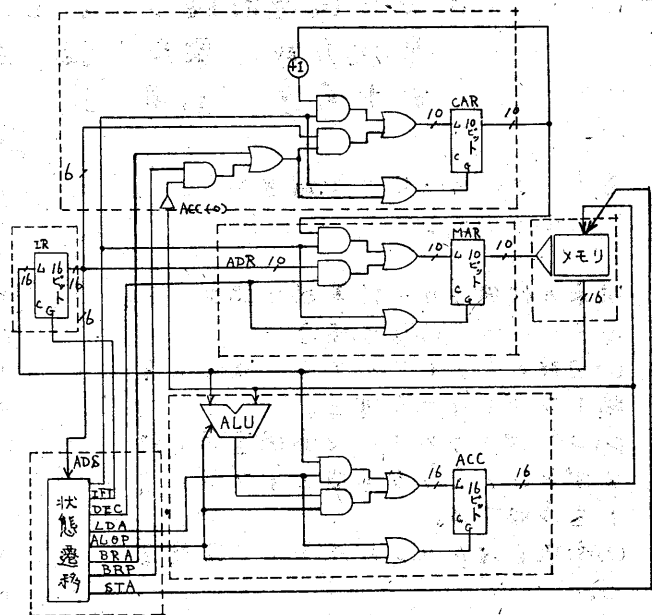


図2 シミュレーションの対象とした計算機の回路構成

能になると思われる。

§5 結論

以上より、データフローマシンの特徴を生かして、容易にかつ高速に論理シミュレーションを行なえる見通しがあった。更に本システムは、信号処理等にも応用でき、又非同期回路のシミュレーションへの拡張も可能である。

一方、今後の課題として、共有記憶の問題等、データフロー本来の問題点もある。又プログラミング言語及びそのコンパイラの開発等の課題も残っている。

<参考文献>

- 上原, 川平, 斎藤, 丸山 「大型コンピュータなどの論理設計に適用できるCADシステム」 日経エレクトロニクス '79, 11, 12, pp104~pp130.