

2J-8

# データフローマシン“TOPSTAR-II”

## ヒートン論理システム・ショーン

## 然元動（富士通）

鈴木達郎 田中英彦

深沢友雄 粟原 謙

元園達（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ことで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ミングも容易に作成することができます。

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て、高速にシミュレーションを実行することができる可能性がある。たとえば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 で稼動し  
ているシミュレーション用システムブ  
ログラムによる制御方式の特徴をあげ  
ると以下のようになる。（参考文献  
(1), (2)）

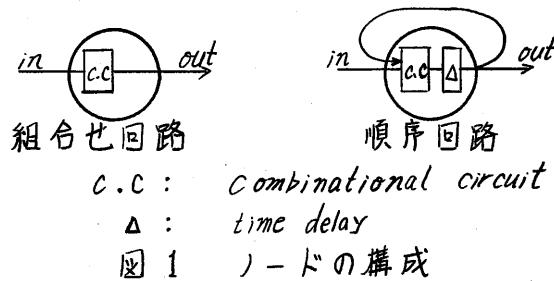
(1) バッファ制御を行ない論理回路のループ構造を認める。

(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が  
フ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ごノードに記憶をもたせる。

言語 C を用ひ乙対象となる論理回路を記述し、い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。



### §3. コンパイラ

これまで、マクロアセンブラーを用いて各モジュールの制御テーブルを作成するだけであつたが、次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を用ひ乙記述)。

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、コンパイラがこれをチェックして、必要

な制御テーブル情報をオブジェクトとして生成するものである。

又、組合せ回路も扱えるように従来一律だったクロックの遅延時間をユーザーが自由に設定できるようにした。

図2にコンパイラのソースの一例を示す。

```
; YE-REGISTER
NODE YEREG(PYEREG),DT1
INFROM ABUS(ABS),YEREG(YER),YREG(YR,YESLI), &
CTL(RISHT,YELST,CLEAR,S2,SFHT),EXIN(CDR2)
OUTTO YREG(YER),EBUS(YER),YEREG(YER),OUT(YER)
TMN YER(L2) /2*0/
ENDNODE

; STAT-REGISTER
NODE STAT(PSTAT),DT1
INFROM STAT(STAU,STA1),ABUS(ABS),ALU(DGCAL),AL
CTL(STSE1,STSE2,STSE3,STSE4,CLEAR,MCK,STRS
OUTTO BBUS(STAO,STA1),ALU(STAO),ALUCD(STAO), &
STAT(STAO,STA1),OUT(STAO,STA1)
TMN STAU(L1) /5/, STA1(L1) /0/
ENDNODE
```

図2 コンパイラのソース例

#### §4. PPS-1のシミュレーション

現在、当研究室にある複合計算機PPS-1の1CPU部分にシリアル機能レベルCのシミュレーションを行なっている。

図3はPPS-1のCPU部分のブロック図を、図4はその中のYEレジスタを言語Cで記述した例を示す。

#### §5. 結論

データフローの特徴を生かした一般的な論理回路のシミュレーションが可能であることがわかった。

今後の課題としては、コンパイラを改良してノードのCモジュールへの自動最適割付けを可能にしたり又、複数クロックを持つ回路のシミュレーションに取り組む、考慮する必要があると思われる。

[参考文献] (1)深沢他「データフローマシンTOPSTARによる論理シミュレーション」昭55情報処理全国大会講演論文集 PP89-90 (2)深沢他「データフローアルゴリズムによる論理シミュレータ」電子装置技術 6-3 1980-10

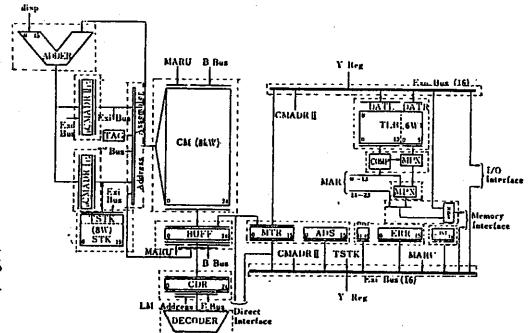
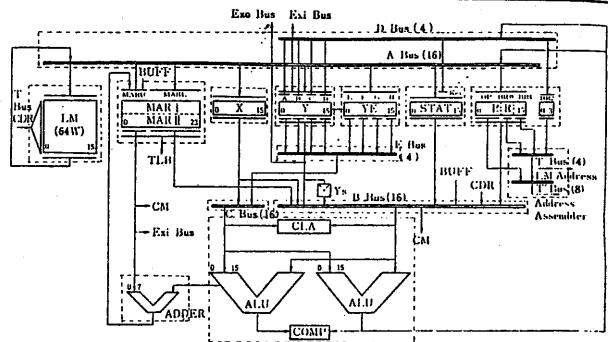


図3 PPS-1のブロック図

```
*****  
* Procedure #5 : ye-register(yeres) [2byte]  
*  
* inset: abs0,abs1,oldyer0,oldyer1,oldyres1,yesli  
* outset: newyer0,newyer1  
*  
*****  
main()  
{  
/*in*/ int abs0,abs1,oldyer0,oldyer1,oldyres1,yesli;  
/*in*/ int risht,yelst,clear,s2,sfht,cdrl;  
/*out*/ int newyer0,newyer1;  
int yemd1,yemd2,cdrl2,rsft,lsft;  
  
/* control signal */  
cdrl2= cdrl & 02;  
rsft= s2 && sfht && cdrl2;  
lsft= s2 && sfht && !cdrl2;  
yemd1= yelst || (rsft && risht);  
yemd2= yelst || (lsft && risht);  
  
/* shift & output operation */  
if (clear)  
    newyer0=newyer1=0;  
else if (!yemd1 && !yemd2){  
    newyer0=oldyer0;  
    newyer1=oldyer1;  
}  
else if (yemd1 && yemd2){  
    newyer0=abs0;  
    newyer1=abs1;  
}  
else if (!yemd1 && yemd2){  
    newyer0=(oldyres1<<7) | (oldyer0>>1);  
    newyer1=(oldyer0<<7) | (oldyer1>>1);  
}  
else if (yemd1 && !yemd2){  
    newyer0=(oldyer0<<1) | (oldyer1>>7);  
    newyer1=(oldyer1<<1) | (yesli & 01);  
}  
else {  
    newyer0=oldyer0;  
    newyer1=oldyer1;  
}
```

図4 YEレジスタを言語Cで記述した例