

UDC 681.3.06:
681.352.6

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-391

伝ぱん遅延時間をもつ素子で構成される
論理回路のシミュレーション

磯部俊夫

1974年10月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

伝ぱん遅延時間をもつ素子で構成される 論理回路のシミュレーション*

磯部 俊夫**

Simulation of Digital Systems Consisting of Elements with Propagation Delay Time

By Toshio ISOBE

This report describes the algorithm and computer program for the simulation of digital systems in which all elements have propagation delay time.

This program consists of FORTRAN-type subroutines and has two important functions. The first is the ability to simulate strictly the influence of the elements propagation delay. The second is element definition with a certain format subroutine which a user can program with his arbitrary properties. An input routine is designed so that input of data is simplified as much as possible.

The results are presented for simulating both a half adder and a very small scale computer to demonstrate the efficiency and accuracy of this simulation program.

1. ま え が き

電子計算機を利用した設計、いわゆるCAD(Computer Aided Design)において電子計算機の主なる役割の一つは、設計者の設計したものが現実に設計者の意図する通りに動くかどうかをシミュレーションすることである。シミュレーションの結果は設計者により評価され、満足な結果が得られない場合は、設計変更が行なわれ再びシミュレーションされる。このような操作が繰返され設計が行なわれる。

電子計算機で代表されるディジタル機械は、一般に数多くの論理素子から構成されており、論理設計の段階での誤りが起りやすい。論理設計での誤まりはハード・ウェアを製作する以前に見つけることが望ましい。ディジタル論理回路はアンド回路、オア回路、ノット回路の基本回路で表現することができ、入力と出力の関係はブール代数を使用して書き表わせる。このような理由によりかなり以前から計算機を使った論理回路シミュレータの研究がなされてきた。初期のシミュレータ^{1~5)}の多くは同期回路用のものであり、クロック・パルスごとに各論理素子の状態を計算するという形式を取っている。論理回路を構成する論理素子もアンド回路、オア回路、フリップ・フロップ等、高々数十種のものに限定されている。一方最近のディジタル

機器はIC, LSI化されアンド回路、フリップ・フロップ等の基本的な論理素子から数ビット・フルアダ、各種レジスタにいたるまでICで用意されている。また1ゲートあたりの伝ぱん遅延時間が1.5ナノ秒という高速なものが使用されるようになり、リード線の伝ぱん遅れ時間が問題になるほどである。このように高速・高機能のIC, LSIで構成された論理回路のシミュレーションは、従来の論理回路シミュレータではシミュレーションしにくくなってきた。すなわち従来のシミュレータは、

- (1) 非同期回路のシミュレーションができない。
- (2) 論理素子の遅延時間が厳密にシミュレーションできない。
- (3) 使用素子がアンド回路、オア回路等の基本論理素子に限定されたものが多く、新しい機能をもつ論理素子をシミュレータに組込むのが困難。

の欠点がある。

クロック・パルスの代りにステップとフェーズという概念を導入し、非同期回路を扱えるようにした論理回路シミュレータ⁶⁾も発表されているが、上記(2), (3)の解決には成っていない。

これらの欠点を除くための論理回路シミュレータを開発した。このシミュレータでは、論理回路を構成する論理素子はフォートラン・サブルーチンという形でシミュレータに組み込まれる。例えばNANDとい

* 昭和48年11月22日受付

** 計算センター (元計測部)

り論理素子はサブルーチン NAND で定義される。したがって新しい機能の論理素子を含む系のシミュレーションが必要になったときは、その機能に相当するサブルーチンを作ればよい。ここで定義される論理素子はすべて伝ばん遅延時間が考慮されてある。また従来の論理シミュレータの扱う信号状態は0あるいは1の二状態すなわち1ビットで表現されていたものを32ビット(2³²の状態を表現できる)に拡張した。このためここで扱えるものは論理素子のみでなく、自動制御系で使用されるブロック図で表わせるようなものも含むことができる。

このシミュレータはフォートラン形式のサブルーチンの集りという形で作った。したがって他のフォートラン・プログラムと混用して使用することができる。

2. 論理回路のモデル化

2.1 論理回路のモデル化

n 個の論理素子から構成されている論理回路を考える。論理素子 i ($i=1, 2, \dots, n$) の任意の時刻 t での出力状態を $f_i(t)$ とかき、 $f_i(t)$ は次のような条件を満すものとする。

$C_{min} \leq f_i(t) \leq C_{max}$ C_{min}, C_{max} は一定

$$f_i(t) - f_i(t + \epsilon) = 0 \text{ あるいは } > C_0$$

ϵ は任意, C_0 は一定の正の値

$f_i(t)$ の集合を Y とし, Y は有限個の点で構成されるものとする。このような条件を満す関数は図1のような階段状になる。

$f_i(t)$ を次のような関数で表現できるものとする。

$$f_i(t) = f_i(t, \tau_{i,1} f_1(t - \tau_{i,1}), \tau_{i,2} f_2(t - \tau_{i,2}), \dots, \tau_{i,n} f_n(t - \tau_{i,n}), C_i(t))$$

ここでは R は各論理素子の結合関係を示す行列であり、

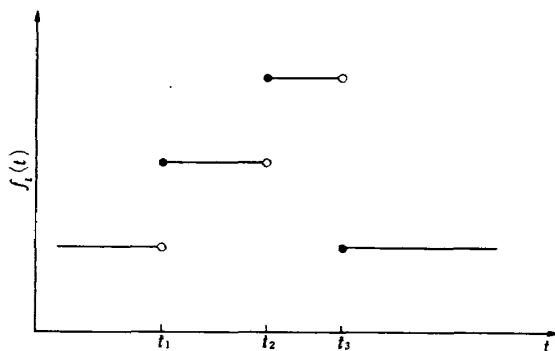


図1 論理素子の出力を表わす関数の一例

$$R = \begin{pmatrix} 0, & r_{1,2}, & r_{1,3}, & \dots & r_{1,n} \\ r_{2,1}, & 0, & r_{2,3}, & \dots & r_{2,n} \\ \dots & \dots & \dots & \dots & \dots \\ r_{n,1}, & r_{n,2}, & \dots & \dots & 0 \end{pmatrix}$$

$$r_{i,k} = \begin{cases} 0 & (i=1, 2, \dots, n, k=1, 2, \dots, n) \\ 1 & \end{cases}$$

とかける。 $r_{i,k}$ の意味は、論理素子 k の出力が論理素子 i に入力されているかどうかを示すもので、1のときは入力されていることを、0のときは入力されていないことを示す。

τ は伝ばん遅延時間と呼ばれるもので

$$\tau = \begin{pmatrix} \tau_{1,1}, & \tau_{1,2}, & \dots & \tau_{1,n} \\ \tau_{2,1}, & \tau_{2,2}, & \dots & \tau_{2,n} \\ \dots & \dots & \dots & \dots \\ \tau_{n,1}, & \tau_{n,2}, & \dots & \tau_{n,n} \end{pmatrix}$$

とかける。 $\tau_{i,k}$ は f_k の変動による f_i への影響が生ずるまでの遅れ時間を表わし、 $\tau_{i,k}$ は論理素子 i に固有のものである。

C_i は論理素子の内部状態を表わすもので、 C_i のとりうる値は集合 Y に属し、 f_i が変動する時刻 t のみ変動可能である。

論理素子 i の出力状態 f_i に変動を与える原因は、 i 以外の論理素子 k の出力状態 f_k の変動と素子自身の時間の変動による。

このような性質を持った論理素子で構成した論理回路において、任意の時間内の回路の状態が有限個の時間列 $T(t_1, t_2, \dots, t_m)$ でのみ変化するものと仮定すると、回路の状態は、 T とその時の各論理素子の出力値で表現できる。時刻 t_l ($1 \leq l \leq m$) で何らかの原因により f_j の値が変動し、回路の状態が変動したとする。次に回路の状態が変動するのは時刻 t_{l+1} である。 $t_l < t < t_{l+1}$ を満す t では、各論理素子の出力状態は一定で変動しない。いま f_i は、 f_j のみの変動により影響を受けるものとする。このときの伝ばん遅延時間を $\tau_{i,j}$ とする。時刻 t_l 後 f_i が変動するのは、時刻 $t_l + \tau_{i,j}$ である。

ここで、 $t_l + \tau_{i,j} \geq t_{l+1}$ 、 $t_l + \tau_{i,j} \in T$ が成り立っている。 $t_l + \tau_{i,j}$ を次の2つの場合に別けて考える。

(1) $t_l + \tau_{i,j} = t_{l+1}$ のとき

時刻 t_{l+1} に変動するのは、 f_j の変動の影響を受けた f_i である。このときの f_i の値は、 $t_l < t' < t_{l+1}$ を満す t' における回路の各論理素子の出力状態、 $C_i(t')$ および t_{l+1} で決定される値とする。

(2) $t_k + \tau_{i,j} > t_{k+1}$ のとき

$t_k = t_k + \tau_{i,j}$ とおく。時刻 t_{k+1} , t_{k+2} , \dots, t_{k-1} で変動する論理素子を q とする。

(i) すべての q において $\tau_{i,q} = 0$ であるとき

$t_{k+1} \leq t \leq t_{k-1}$ で f_i は論理素子 q からの影響を受けない。したがって t_{k+1} , t_{k+2} , \dots, t_{k-1} での f_j による回路の状態の変動は、 f_i に何らの影響をも与えない。よってこの場合は(1)と同様になる。

(ii) $\tau_{i,q} \neq 0$ のとき

t_{k+1} , t_{k+2} , \dots, t_{k-1} において f_q が最初に変動する時刻を t_p とする。 f_q による f_i への伝ばん遅延時間を $\tau_{i,q}$ とし、また $t_k \leq t' < t_{k+1}$, $t_p \leq t'' < t_{p+1}$ なる時刻をそれぞれ t' , t'' とする。

(a) $t_k < t_p + \tau_{i,q}$ のとき

$t = t_k$ で、 f_i の値は t' における各論理素子の出力状態、 $C_i(t')$ および t_k で決定される値とする。
 $t = t_p + \tau_{i,q}$ で、 f_i の値は t'' における各論理素子の出力状態、 $C_i(t'')$ および $t_p + \tau_{i,q}$ で決定される値とする。ただし、この場合 $t_k < t < t_p + \tau_{i,q}$ において変動する q 以外の論理素子が f_i に影響を与えるときは、その影響を受ける。

(b) $t_k \geq t_p + \tau_{i,q}$ のとき

$t = t_p + \tau_{i,q}$ で、 f_i の値は t'' における各論理素子の出力状態、 $C_i(t'')$ および $t_p + \tau_{i,q}$ で決定される値とする。このとき f_i は、 f_j からの影響を無視する。

以上のようにモデル化された系では、系を構成する各論理素子の出力状態を表わす関数 f, r, τ および各論理素子の $t=0$ における出力状態が与えられれば、シミュレーションができる。

論理素子 i は、図 2-(a) のような N 入力端子を持つブロックで表現する。一般にブロックの出力は一つとは限らず図 2-(b) のように多出力を持ったブロックとする。

2.2 論理素子

論理素子は関数 f に相当する。 f の値を決定するパラメータは、時間 t , 入力端子に接続されている他の素子の出力状態 r , 内部状態 C である。 r の変動によ

り f が変動するときの遅れすなわち伝ばん遅延時間 τ を t とは独立なパラメータとしたとき、時間のパラメータ t を含む論理素子と含まない論理素子に分ける。時間のパラメータを含む論理素子を仮に発振器と呼ぶことにする。発振器は次の 2 つの種類がある。

$$f_{G_1} = f(t, c)$$

$$f_{G_2} = f(t, r, c, \tau)$$

f_{G_1} に属する発振器は、入力端子 r を持たず外部の状態に一切左右されず時間と内部状態だけの関数となる。電源を入れれば直ちに発振する発振器は f_{G_1} に属する発振器である。 f_{G_2} に属する発振器は、入力端子 r を持った発振器で r の状態により発振をしたり、しなかったりするような発振器である。モノステーブルマルチバイブレタはこの f_{G_2} に属する発振器に入れる。

時間のパラメータを含まない素子において、その出力の値が変動するためには入力条件 r が変動することが必要条件となる。普通論理素子といわれているものは、大体この範ちゅうに入る。一般に内部状態 C のパラメータを含まないものを組合わせ回路、含むものを順序回路とよんでいる。

入力端子を持つ論理素子では、入力条件の変動した時刻から出力の変動する時刻までの間を過度状態、それ以外を定常状態という。すなわち、

入力状態の変化 \rightarrow 過度状態 \rightarrow 定常状態
 という経過をたどる。

これらの論理素子で構成される系は次のような条件を満たすものとする。

(1) 論理素子の出力端子はいくつかの他の論理素子の入力端子に結合されてもよいが、自分の入力端子あるいは他の出力端子と結合されてはならない。

(2) 系の内には少なくとも一つは発振器が存在しなければならない。

3. シミュレーション・プログラム

2.での論理回路のモデル化に従ってシミュレーション・プログラムを作る。プログラムは、各論理素子の結合関係 R を入力データから読み込みテーブルにその関係を書き込む部分、 R を参照し各論理素子の出力

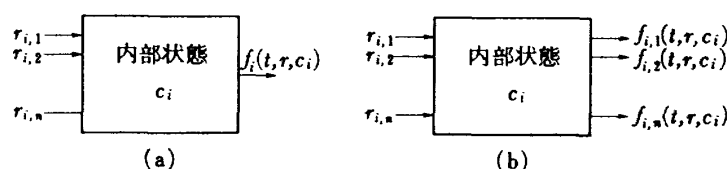


図 2 論理素子のブロックによる表現

態から時間を進める部分および各論理素子の出力状
を計算する部分から成っている。

プログラムは共通データ領域を中心に構成し、入出
ルーチン、シミュレーション・ルーチン等を共通デ
ータ領域のデータを処理するサブルーチンという形で
設計した。したがって各サブルーチンは、共通デー
タ領域以外は互に独立に取扱える。形式的には、一つ
のまとまったプログラムという形ではなく、それぞれ独
した機能を持つサブルーチンの集合という形になっ
ている。シミュレーションの方法は、離散変化モデル
の汎用システム・シミュレータSIMSCRIPT⁷⁾
似たものとなった。

図3はプログラムの構造を示したものである。

* メイン・プログラム、出力ルーチンの例は付録
を参照。

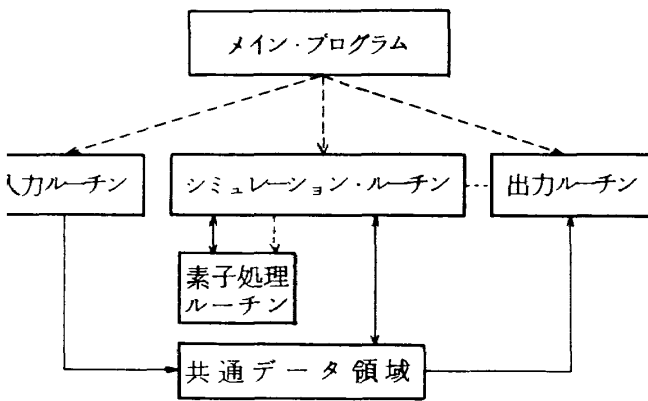


図3 プログラムの構造

3.1 データ

共通データ領域には、各論理素子間の関係、現時間
の出力端子の状態および次の時間での出力の状態に関
するデータの領域が確保されている。シミュレーショ
ンすべき系内の論理素子の出力端子、入力端子にはそ
れぞれ1から始まる整数で通し番号を付けておく。各
論理素子の外部構造はテーブルIEDTに、処理ルー
チンで参照するコンスタント・データはITADに、
出力端子の現時間の状態および各端子間の結合関係は
テーブルIÖUTに、入力端子に関するデータはテー
ブルINPT、過度状態にある出力端子の状態はテー
ブルNPTTに書き込まれる。IÖUT, INPT,
NPTTの各テーブルは次に示すような構造になっ
ている。

IÖUTテーブル

$I\bar{O}, K, G, IN_1, IN_2, \dots, IN_n$

$I\bar{O}$; 現在の出力端子の出力値

K ; $K=1$ なら現在過渡状態である。

$K=0$ なら現在定常状態である。

G ; 発振器の出力であるなら発振器名、発振
器の出力でないときは $G=0$

IN ; 出力端子に接続されている入力端子の情
報が書かれているINPUTテーブルの
場所

INPUTテーブル

$E, \bar{O}T$

IEDT

理ルーチン名	ITADT での場所	NINPT での場所	入力数	IÖUT での場所	出力数
BLÖCK 1	N_{c1}	N_{i1}	p_1	N_{o1}	q_1

BLÖCK N	N_c	N_i	p	N_o	q

IÖUT

$I\bar{O}$	K	G	IN_1	IN_2	---	---	IN_n
$I\bar{O}_1$	K_1	G_1	IN_{11}	IN_{21}			
$I\bar{O}_{N_0}$	K_{N_0}	G_{N_0}	IN_{1N_0}	IN_{2N_0}			

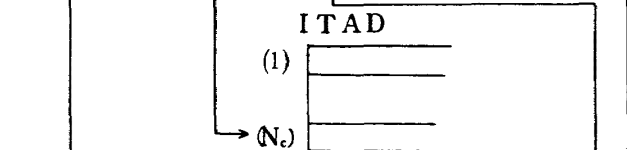


図4 素子に関するデータ テーブル間の関係

E;論理素子名

$\bar{O}T$;接続されている出力端子の情報が書かれている $\bar{I}\bar{O}UT$ テーブルの場所

p 入力 q 出力を有する n 論理素子に関するデータは、各テーブルに図 4 に示すような関係で書かれる。

NPTT テーブル

NT, N, $\bar{N}O$

NT; 過渡状態が終了する時刻

N; 過渡状態にある出力端子の $\bar{I}\bar{O}UT$ テーブルの場所

$\bar{N}O$; 過渡状態後の出力端子の値

3.2 シミュレーション・ルーチン

3.2.1 シミュレーション・ルーチン

シミュレーションはインタプリテブな方法で行なう。現在の時刻を $\bar{N}OWT$ とする。シミュレーションは $\bar{N}OWT=0$ から始まる。このときテーブル NPTT には何も書かれていない。まず全論理素子に対してそれぞれの処理ルーチンが CALL され、各論理素子に与えられた入力条件がその論理素子の出力値を満足しているかどうか調べられる。満足していない場合は、その論理素子の伝ばん遅延時間後に出力値が変動するのであるから、その時刻、出力値、出力端子番号を NPTT にストアする。発振器ルーチンは時刻 0 で CALL されたとき必ず次の状態が存在するように作られていなくてはならない。そうでなければ $\bar{N}OWT=0$ での操作が終わったとき、テーブル NPTT には少なくとも一つは情報がストアされている。そして以下のようにシミュレーションを実行する。NPTT にストアされたデータの中で $\bar{N}OWT$ に最も近い時刻に出力値が変動するデータを取り出す(一つ以上あってもよい)。そのときデータが、

$NT', N', \bar{N}O'$

であったとする。ここで新たに NT' を $\bar{N}OWT$ とし時間を進め、テーブル $\bar{I}\bar{O}UT$ の第 N' 番目の K, $\bar{I}\bar{O}$ を次のように書き直し、P, IN 部を参照し処理する。

- (1) K を 0 にする
- (2) $\bar{I}\bar{O}$ を $\bar{N}O'$ に書きかえる
- (3) G 部が 0 でないときは G 部に書かれている発振器処理ルーチンを CALL する
- (4) IN 部から INPT テーブルを参照し、今書きかえた出力端子と接続されている論理素子の処理ルーチンを CALL する

処理ルーチンは CALL されると、現在時刻 $\bar{N}OWT$ と入力条件から出力値 y を計算し、もし現在の出力

値が y と異なるときは、出力値が y になる時刻を求め、処理ルーチンの結果は $\bar{I}\bar{O}UT$ の現在値と比較される。もし異なれば NPTT にストアされ、過渡状態を示す K を 1 にする。このときすでに過渡状態であり、その終了時間が現時刻で起きた過渡状態の終了時間より遅いときは、前に生じた過渡状態は NPTT テーブルから取り除かれる。そうでないときは、現時刻での過渡状態もそのままストアする。

NPTT に情報がストアされていない状態になれば、系内の全論理素子が定常状態になったのでありシミュレーションは終了する。

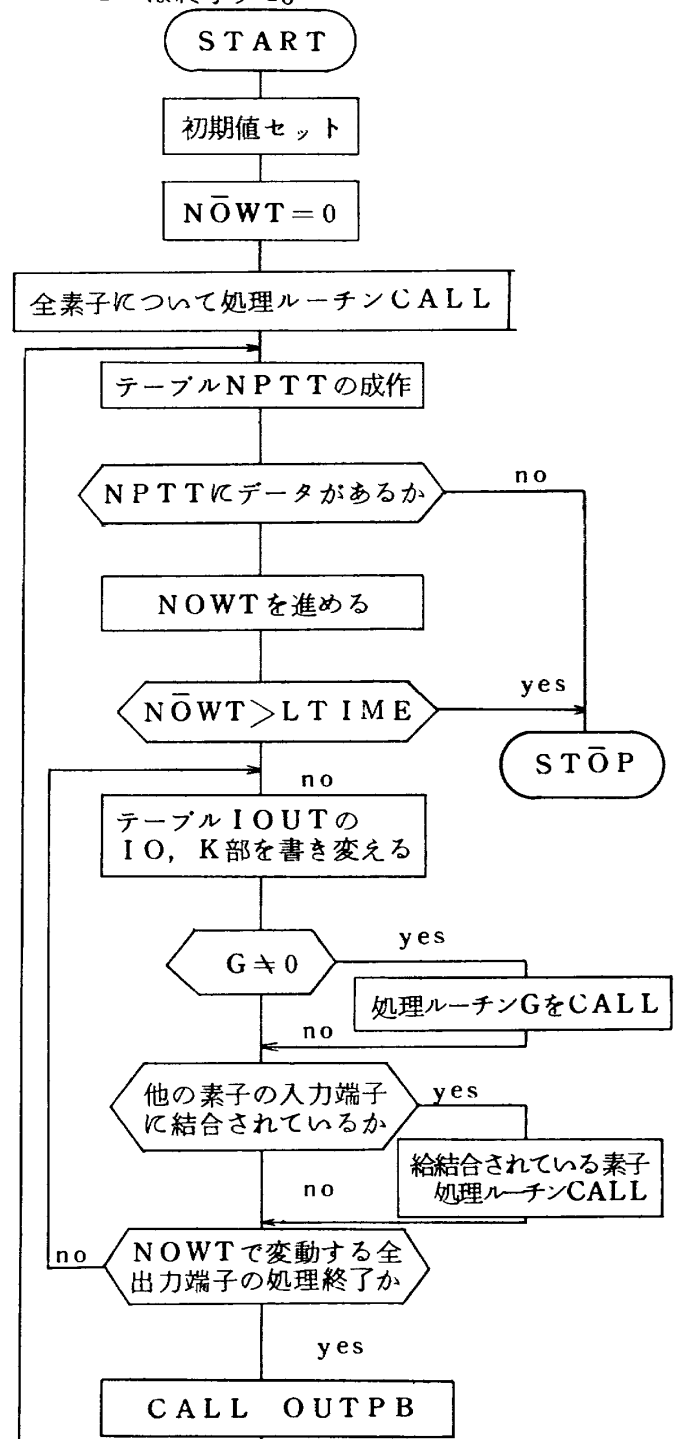


図 5 シミュレーション・ルーチンのフロー・チャート

図5はシミュレーション・ルーチンのフロー・チャートである。

3.2.2 素子処理ルーチン

論理素子の処理は素子処理ルーチンで行なわれる。シミュレーション・ルーチンから素子処理ルーチンへ送られるデータは、現在時間 \overline{NOWT} 、 \overline{NOWT} における入力端子の値、出力端子の値、現出力値になった時刻、およびコンスタント・データである。素子処理ルーチンからの出力は、入力条件と \overline{NOWT} における論理素子の状態で決まる値およびその値が現在の出力値と異なるときは出力端子がその値に変わる時刻(この時刻は \overline{NOWT} より大きくなければならない)である。素子処理ルーチンは、入力と出力の形式が整っていることを要求するが、その内部でのデータのあつかい方、処理等は一切規定しない。素子処理ルーチンは発振素子と非発振素子に分けられる。これらはシミュレーション・ルーチンから素子処理ルーチンをCALLする条件が異なることによる分類である。発振素子は、その出力端子の状態が変動したときにCALLされる素子であり、非発振素子は、その入力条件が変動したときにCALLされる素子である。発振素子でも入力端子をもつ素子は、入力条件の変動でもCALLされる。

3.3 入力ルーチン

入力ルーチンは、シミュレーションすべき系を記述してある入力媒体からデータを読み込み、シミュレーション・ルーチンで処理可能なデータ形式に直し、共通データ領域にファイルするためのルーチンである。入力媒体としては、カード、磁気テープ、グラフィックス端末等があり、必要に応じてそれぞれの入力ルーチンを作ればよい。ここではカード入力による入力ルーチンDSDCRDについて述べる。

DSDCRDが読むデータには、素子定義データ、定数表定義データ、処理ルーチン定義データ、出力指定データ、ENDデータである。データはエレメントと仕切り記号で構成し、エレメントとエレメント間には仕切り記号を入れる。仕切り記号は、ブランク、 \prime 、 \prime 、 $($ 、 $)$ である。

(1) 素子定義データ

素子番号、処理ルーチン名、 \overline{OT} データ、 NI データ、 \overline{NO} データ、 T データ

の形をしている。

素子番号 5桁以内の数、他の素子との識別をするためのものと同じ番号を他の素子に付けてはいけない。

処理ルーチン名 論理素子が処理されるルーチン名

\overline{OT} データ \overline{OT} /出力端子番号 n (初期値)

{ n と接続されている入力端子}

q 個の出力を持った論理素子では q 個の \overline{OT} データが必要、1出力のときは $\prime n$ は必要ない。初期値が0のときは初期値を略しても良い。入力端子は

素子番号/端子番号 n

という形で書く。入力が1つしかない論理素子、あるいは p 入力でも全てが対等な場合は $\prime n$ は省略しても良い。

NI データ NI 入力端子数

\overline{NO} データ \overline{NO} 出力端子数

T データ T 定数表識別番号

NI 、 \overline{NO} 、 T の各データは処理ルーチン定義データで定義されたものを使うなら省略しても良い。また \overline{OT} 、 NI 、 \overline{NO} 、 T の各データは任意の順序で並んでいても良い。

(2) 定数表定義データ

T 定数表識別番号 { 定数 }

(3) 処理ルーチン定義データ

素子定義データから素子番号、 \overline{OT} データを除いたもの、素子定義データで定義してあれば処理ルーチン定義データは必要ない。

(4) 出力指定データ

\overline{OUTPUT} { 素子番号/出力端子番号 }

(5) ENDデータ

END シミュレーション開始時刻、終了時刻

データの終りを示す。このカードを読むとDSDCRDは必要な処理をしてRETURNする。

データが一枚のカードに書ききれないときはCを書くこと次のカードに続いて書くことができる。

(1)~(4)までのデータは順序不同で良い。なお上記で{ }の記号は列を表わしている。列内の仕切り記号はブランクあるいは \prime である。

A	B		S	C
0	0		0	0
1	0		1	0
0	1		1	0
1	1		0	1

図6 半加算器の真理値表

4. 使用例

4.1 半加算器のシミュレーション

DTLタイプのICで半加算器を設計する。半加算器の真理値表は図6の通りである。これは5個のNANDゲートを使用して図7の回路で実現できる。図7は次のようにデータ化できる。

1. NAND OT 2 3 5
2. NAND OT 4
3. NAND OT 4
4. NAND
5. NAND

シミュレーションを実行するためにはサブルーチンNANDを用意する必要がある。そこで実際のNAND回路ICがどのような特性を持つか調べる。ここで例に上げるICは三菱電機製のM5946Pである。カタログによる特性は、論理値0をL, 1をHとすると温度25℃で出力がLからHになる伝ばん遅延時間 t_{pdH} は最小25ns(印加電源電圧 $V_{cc}=4.5V$)最大80ns($V_{cc}=5.5V$), HからLになる伝ばん遅延時間 t_{pdL} は最小10ns($V_{cc}=4.5V$)最大30ns($V_{cc}=5.5V$)であり、 t_{pdH} , t_{pdL} の定義は図8のようになされている。図9に実際のICでの入力、出力の波形のオシロスコープでの観測したものを示す。

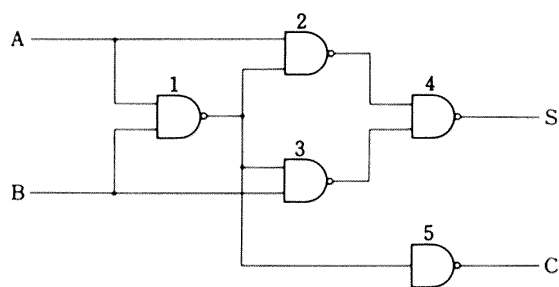
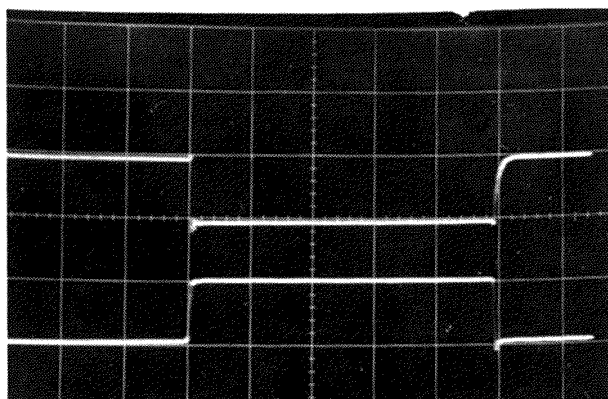
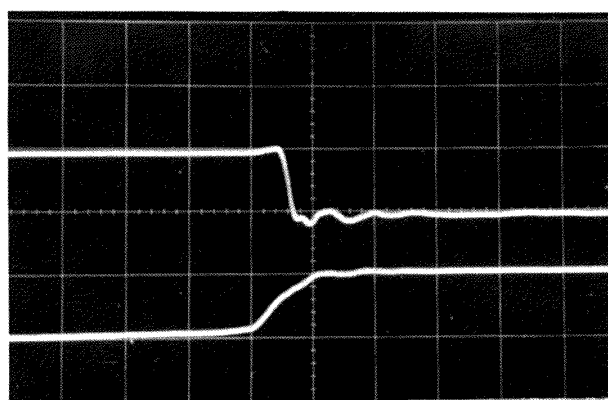


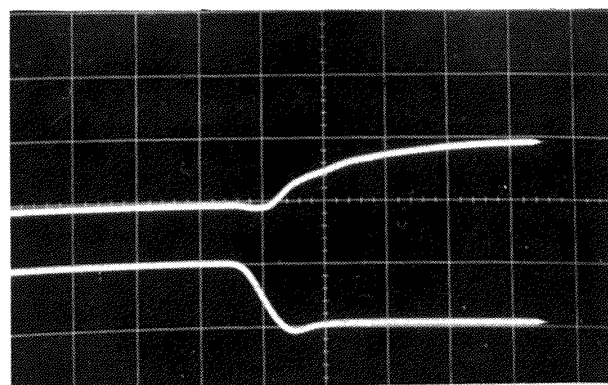
図7 半加算器の一例



(a) 時間軸 2 μ sec/div



(b) 時間軸 100 n sec/div



(c) 時間軸 100 n sec/div

図9 IC(M5946P)の入力入出力波形
電圧軸5V/div

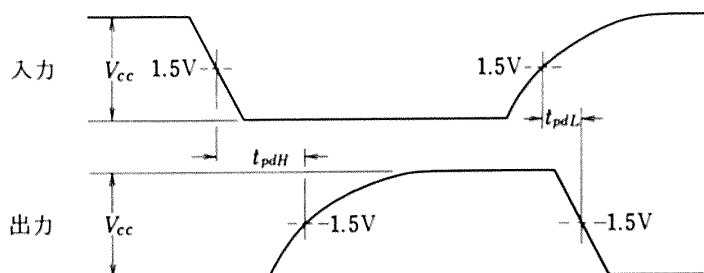


図8 伝ばん遅延時間の定義

t_{pdH} というのは図8から分るように電圧が0Vから1.5Vになるまでの時間で V_{cc} になるまでの時間ではない。すなわちHの状態というのは1.5V $\sim V_{cc}$ の範囲である。したがってHからLに変わるときHの状態で図10に示すようにLになるまでの時間も異なってくる。以上のことを考慮してサブルーチンNANDを作った。このルーチンで必要とする定数表は t_{pdH} , t_{pdL} の順に書かれる。パルスはサブルーチンPULS8で作る。これは9出力端子をもつ発振器で、9番目の端子はタイミング・パルスが、1 \sim 8番目の端子には定数表のデータで指定した値に従ってパルスが発生する。例えば1の値に対しては端子1に、2に対しては端子2に、3に対しては端子1と2にというふうにパルスが発生する。PULS8の定数表はスタート時間、終了時間、パルス巾、データ1、データ2……の順に並べる。データの値が128以上になると再びデータ1に戻る。

NANDの $t_{pdH}=50ns$, $t_{pdL}=20ns$, パルス周期1,000ns, パルス巾500nsでシミュレーションを実行する。図11は入力データである。シミュレーション結果を図12に示す。図13は図12をプロッタで描かしたものである。この結果によると時刻4020で素子2, 3の出力に素子1の伝はん遅延によるハザードが発生している。図14はこのときの実際の回路での観測波形である。このハザードは素子4の出力には伝はんされていない。図15はこのときの実際の回路での観測波形である。以上のように本シミュレータの結果と実際の半加算器(図7)の各部の出力状態が一致した。従来のシミュレータにおいては、素子2, 3に現われるハザード、またこれらの影響が素子4の出力に現われないという現象をシミュレーションするのは困難であった。

4.2 小型計算機のシミュレーション

1語8ビット, 記憶容量32語の8命令を持つ計算機を設計し, 設計どりに動作するかどうかをシミュレーションしてみる。

この計算機の命令形式は図16のようになっている。命令は次の8つである。

- (1) 無条件ジャンプ(J)
- (2) アキュムレータ(ACC)が0ならジャンプ(JZ)
- (3) アドレスで指定された番地の内容とACCの内容を加算してACCに入れる。(A)
- (4) ACCの内容からアドレスで指定された番地の内容を引きACCに入れる。(S)
- (5) ACCの内容をアドレスで指定された番地に格納する。(ST)
- (6) アドレスで指定された番地の内容をACCに入れる。(L)
- (7) アドレス部の内容だけシフトする。3ビット目が0のときは左へ, 1のときは右へシフトする。(SL)
- (8) 入出力命令

図18が設計例(入出力命令に関する部分は省略)である。ここで使用されている論理素子は次の11種

```

1 NAND OT 2 3 5
2 NAND OT 4
3 NAND OT 4
4 NAND
5 NAND NI 1
7 PULS8 OT/1 1 2 OT/2 1 3 NO 9 NI C 1 3
CUTPLT 7/9 7/1 7/2 1 2 3 4 5
T 1 50 20
T 3 1000 500 500 4 1 2 3 128
NAND NO 1 NI 2 T 1
END 0 5000

```

図11 半加算器の入力データ

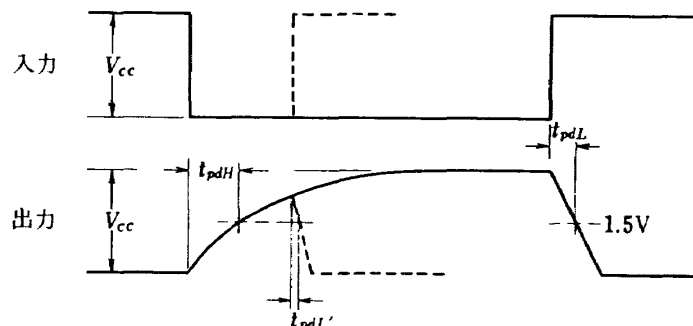


図10 Hの状態による t_{pdL} の変動

TIME	T	A	B	1	2	3	S	C
0	0	0	0	0	0	0	0	0
50	0	0	0	1	1	1	1	1
51	0	0	0	1	1	1	0	0
1000	1	0	0	1	1	1	0	0
1500	0	0	0	1	1	1	0	0
2000	1	1	0	1	1	1	0	0
2020	1	1	0	1	0	1	0	0
2070	1	1	0	1	0	1	1	0
2500	0	0	0	1	0	1	1	0
2550	0	0	0	1	1	1	1	0
2570	0	0	0	1	1	1	0	0
3000	1	0	1	1	1	1	0	0
3020	1	0	1	1	1	0	0	0
3070	1	0	1	1	1	0	1	0
3500	0	0	0	1	1	0	1	0
3550	0	0	0	1	1	1	1	0
3570	0	0	0	1	1	1	0	0
4000	1	1	1	1	1	1	0	0
4020	1	1	1	0	0	0	0	0
4021	1	1	1	0	1	1	0	0
4070	1	1	1	0	1	1	0	1
4500	0	0	0	0	1	1	0	1
4550	0	0	0	1	1	1	0	1
4570	0	0	0	1	1	1	0	0
5000	1	0	0	1	1	1	0	0

図12 半加算機のシミュレーション結果

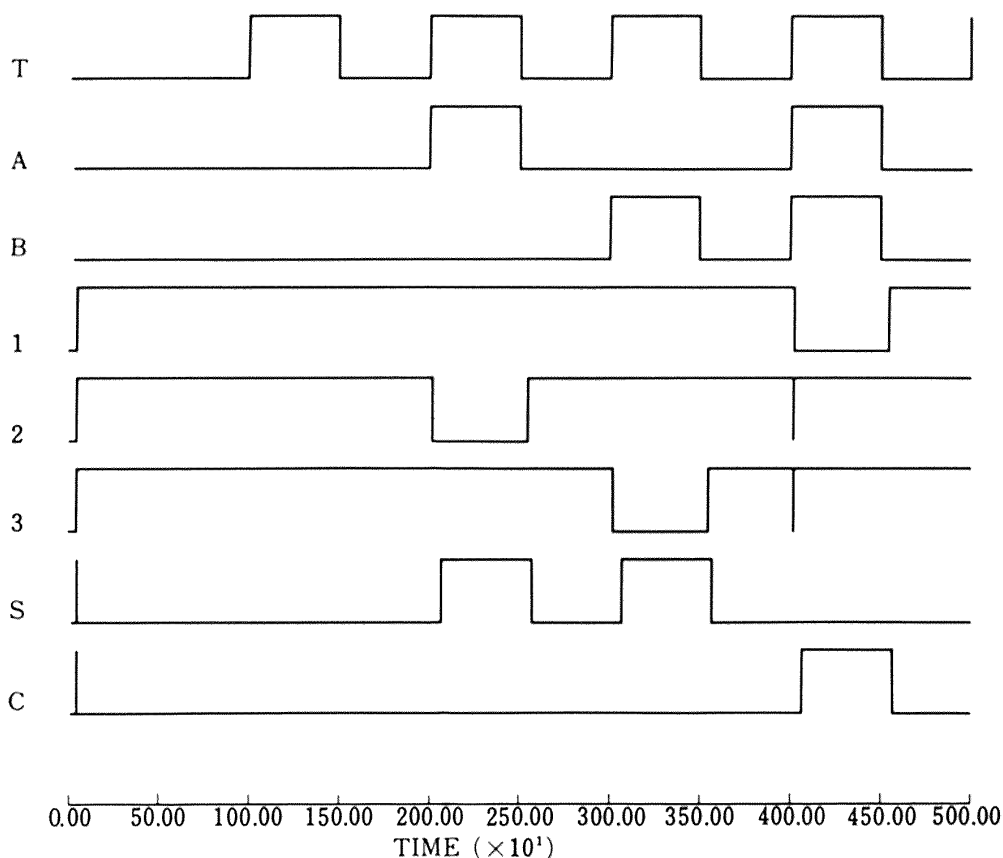
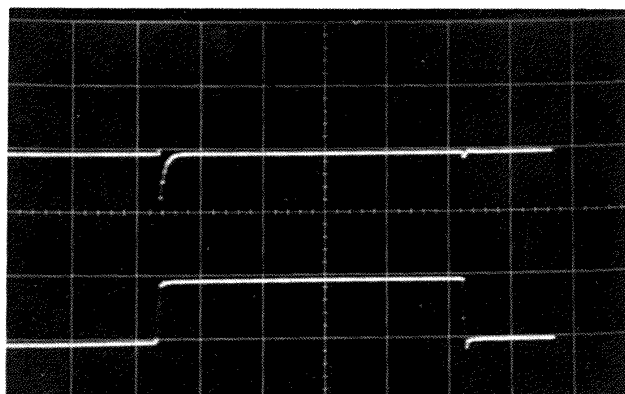
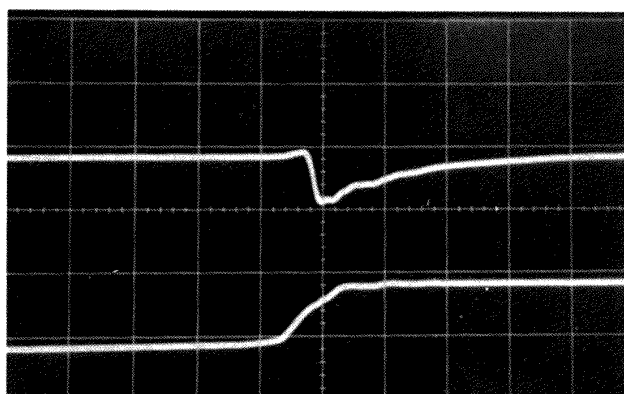


図13 プロッタによる半加算器のシミュレーション結果



(a) 時間軸 2 μsec/div



(b) 時間軸 100 n sec/div

図14 素子1の伝ばん遅延による素子2(3)に生ずるハザード 電圧軸5V/div

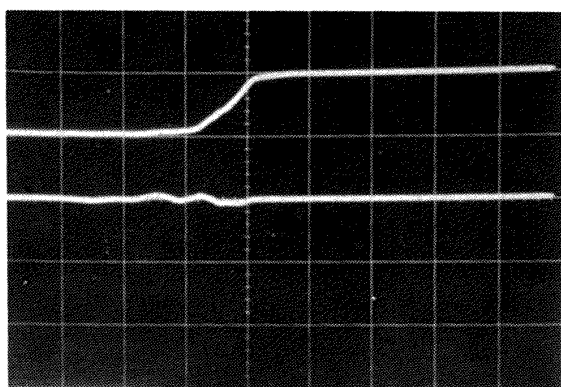


図15 素子2, 3に図14のハザードが発生したときの素子4の出力波形 電圧軸5V/div 時間軸100n sec/div

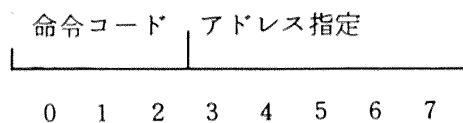


図16 命令形式

命令コード	記号	命令
0	J	Jump
1	JZ	Jump on Zero
2	A	Add
3	S	Subtract
4	ST	Store
5	L	Load
6	SL	Shift Logical
7	IOC	Input Output Control

図17 命令コード表

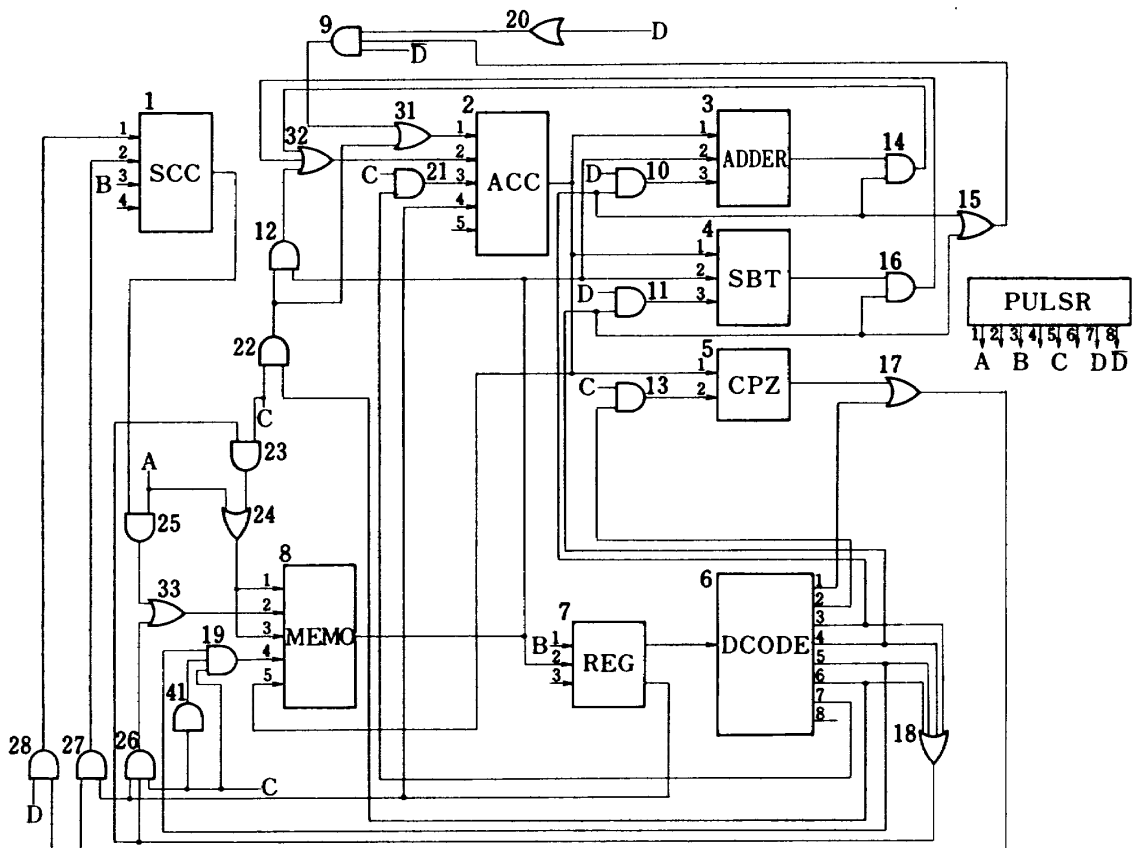


図18 小型計算機的设计例

類である。

(1) SCC 8

入力端子1がオンのとき入力端子2(5ビット)のデータが出力レジスタにストアされる。入力端子3がオンのとき出力レジスタの内容に1を加える。入力端子4がオンのとき出力レジスタをクリアする。

(2) ACC 8

入力端子5がオンのとき出力レジスタをクリアする。入力端子1がオンのとき入力端子2(8ビット)を出力レジスタへストアする。入力端子3がオンのとき入力端子4(8ビット)の内容だけシフトする。0ビット目が1なら右へ, 0なら左へシフトする。

(3) ADDER 8

入力端子3がオンのとき入出端子1と2(それぞれ8ビット)の内容を加算して出力する。

(4) SBT 8

入力端子3がオンのとき入力端子1(8ビット)の内容から入力端子2(8ビット)の内容を減算して出力する。

(5) CPZ

入力端子2がオンのとき入力端子1の内容が0なら出力を1にする。それ以外は出力0

(6) DCODE

入力端子(3ビット)の内容に1を加えた番号の出力端子を1にする。

(7) REG 1

入力端子3がオンのとき出力レジスタ1, 2がクリアされる。入力端子1がオンのとき入力端子2(8ビット)の上位3ビットが出力レジスタ1へ, 下位5ビットが出力レジスタ2へストアされる。

(8) MEMO 32

8ビット32語のメモリ, 入力端子1がオンのとき入力端子2(5ビット)の内容がアドレス・レジスタにストアされる。入力端子3がオンのとき入力端子4(8ビット)の内容がアドレス・レジスタの示すメモリ番地へ書き込まれる。入力端子5がオンのときアドレス・レジスタの示すメモリ番地の内容が出力レジスタに出力される。

(9) AND 8

8ビット, パラレル・アンド・ゲート

(10) OR 8

8ビット・パラレル・オア・ゲート

(11) PULSR

パルス発振器, 出力端子1~8の発生パルスが図19のようなもの, 出力端子1, 3, 5, 7でのパルスをA, B, C, Dと呼び次のような役目をする。

Aでは命令読み出し

Bでは命令解説, およびSCCの内容を+1する。

Cではオペランド読み出し, またはJ, JZ, SL命令では実行。

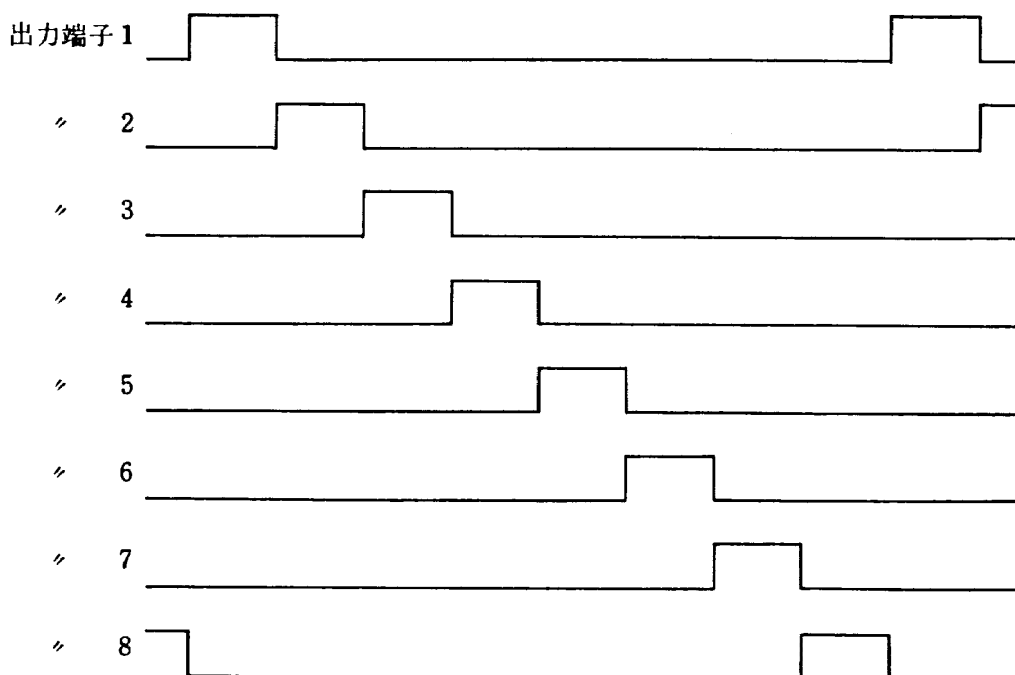


図19 PULSRの出力波形

Dでは命令実行

以上11種類の機能に対する処理ルーチンを用意し図18の回路を図20のようなデータに書きシミュレーションを行なう。各論理素子の伝ぱん遅延時間は適当に決めた。

メモリの初期値を次のようにしSCCを0にセットする。

0番地	1	(J, 1)
1番地	168	(L, 8)
2番地	73	(A, 9)
3番地	138	(ST, 8)
4番地	193	(SL, 1)
5番地	32	(JZ, 0)
6番地	105	(S, 9)
7番地	6	(J, 5)
8番地	1	
9番地	2	
10番地	0	
)		
32番地	0	

このようなメモリの初期値は、次のようなことを意

味する。まず制御は0番地から始まるが、0番地では無条件に1番地へジャンプするだけである。1番地の意味は8番地の内容をACCへロードせよということで、8番地には1が入っているのでACCに1が置かれる。2番地は、ACCの内容と9番地の内容を加えACCに置けということで、9番地の内容2と1が加えられた3がACCに置かれる。3番地ではACCの3が8番地にストアされ、4番地ではACCの内容が左へ1ビットシフトされ6になる。5番地では、ACCが0なら0番地へジャンプせよという命令である。ACCは、いま6が入っているので6番地へ制御が移る。6番地では、ACCの内容から9番地の内容2が引かれACCは4になる。7番地は5番地への無条件ジャンプである。ACCが0になるまで5.6.7の間をループする。ACCが0になると制御は0番地へ移り以下同様な動きをする。図21はこのシミュレーションの結果である。図21で例えば制御が1番地へ移った時刻3,000以降を見る。時刻3,000でパルス発生器PULSRからA相パルスが発生する。パルスAにより素子25のゲートが開きSCCの出力は、素子25, 33を通り時間10の遅れで素子8のMEMO

```

1 SCC8 OT(C) 25 NO 1 NI 4 T 1
2 ACC8 OT(C) 3/1,4/1 5/1 8/5 NC 1 NI 5 T 2
3 ADDR8 CT 14 NO 1 NI 3 T 3
4 SET8 OT 16 NC 1 NI 3 T 3
5 CF2 CT 17 NC 1 NI 2 T 7
6 DCCDE CT/1 17 OT/2 13 CT/3 18,10,14,15 OT/4 18,11,16,15 OT/5 18,19 C
   CT/6 18,22 OT/7 21 NO 8 NI 1 T 2
7 REG1 CT/1 6 CT/2 2/4,26,27 NC 2 NI 3 T 2
8 MEMO32 CT 7/2,12,4/2,3/2 NO 1 NI 5 T 4
9 AND8 CT 31 NC 1 NI 3 T 1
10 AND8 CT 3/3 NO 1 NI 2 T 1
11 AND8 CT 4/3 NO 1 NI 2 T 1
12 AND8 CT 22 NC 1 NI 2 T 1
13 AND8 CT 5/2 NO 1 NI 2 T 1
14 AND8 CT 22 NO 1 NI 2 T 1
15 OR8 OT 9 NO 1 NI 2 T 1
16 AND8 OT 22 NC 1 NI 2 T 1
17 OR8 OT 27,28 NO 1 NI 2 T 1
18 OR8 OT 26,25 NO 1 NI 4 T 1
19 AND8 OT 2/4 NO 1 NI 3 T 1
20 OR8 CT 9 NO 1 NI 1 T 6
21 AND8 OT 2/3 NO 1 NI 2 T 1
22 AND8 CT 12,31 NO 1 NI 2 T 1
23 AND8 CT 24 NC 1 NI 2 T 1
24 OR8 OT 2/1,3/3 NO 1 NI 2 T 1
25 AND8 CT 23 NC 1 NI 2 T 1
26 AND8 OT 23 NC 1 NI 3 T 1
27 AND8 OT 1/2 NO 1 NI 2 T 1
28 AND8 OT 1/1 NO 1 NI 2 T 1
31 OR8 OT 2/1 NO 1 NI 2 T 1
32 OR8 CT 2/2 NO 1 NI 3 T 1
33 OR8 OT 2/2 NO 1 NI 2 T 1
40 PULSR CT/1 25,24 OT/3 1/3,7/1 OT/5 13,19,41,21,22,23,26 OT/7 10,11,28,20 C
   OT/8 9 NO 8 NI 1 T 5
   OT 19 NC 1 NI 1 T 2
41 OR8
T 1 5
T 2 50
T 3 100
T 4 150
T 5 1000,250, 100000
T 6 15
T 7 300
    
```

図20 小型計算機の入力データ

の入力端子2に達する。またパルスAは素子24を通り時間5の遅れで素子8の入力1, 3をオンにし、メモリの読み出しを開始する。SCCの出力が素子8の入力2へ達する時刻は3010であり、メモリ読み出しに必要な時間が150であるから、SCCの出力値1の番地の内容が読み出されるのは時刻3160であ

り、168という値がメモリの出力レジスタに得られる。B相パルスは時刻3500に発生する。これは素子1(SCC)の入力3に加えられ、SCCの内容に時間5の後、1を加えられる。またBパルスは素子7の入力1に加えられ、メモリの出力値を命令部とアドレス部に分ける。それぞれの値5, 8は時刻3550

TIME	A	B	C	D	*D	SCC	ACC	MEMORY	F	M
1000	1	0	0	0	0	0	0	0	0	0
1155	1	0	0	0	0	0	0	0	0	0
1250	1	0	0	0	0	0	0	0	0	0
1500	1	0	0	0	0	0	0	1	0	0
1505	1	0	0	0	0	1	0	1	0	0
1550	1	0	0	0	0	1	0	1	0	1
1750	1	0	0	0	0	1	0	1	0	1
2000	1	0	1	0	0	1	0	1	0	1
2250	1	0	0	0	0	1	0	1	0	1
2500	1	0	0	1	0	1	0	1	0	1
2750	1	0	0	0	1	1	0	1	0	1
3000	1	0	0	0	0	1	0	1	0	1
3160	1	0	0	0	0	1	0	168	0	1
3250	1	0	0	0	0	1	0	168	0	1
3500	1	0	0	0	0	1	0	168	0	1
3505	1	0	0	0	0	2	0	168	0	1
3550	1	0	0	0	0	2	0	168	5	8
3750	1	0	0	0	0	2	0	168	5	8
4000	1	0	1	0	0	2	0	168	5	8
4065	1	0	1	0	0	2	168	168	5	8
4160	1	0	1	0	0	2	168	1	5	8
4220	1	0	1	0	0	2	1	1	5	8
4250	1	0	0	0	0	2	1	1	5	8
4500	1	0	0	1	0	2	1	1	5	8
4750	1	0	0	0	1	2	1	1	5	8
5000	1	0	0	0	0	2	1	1	5	8
5160	1	0	0	0	0	2	1	73	5	8
5250	1	0	0	0	0	2	1	73	5	8
5500	1	0	0	0	0	2	1	73	5	8
5505	1	0	0	0	0	3	1	73	5	8
5550	1	0	0	0	0	3	1	73	2	9
5750	1	0	0	0	0	3	1	73	2	9
6000	1	0	1	0	0	3	1	73	2	9
6160	1	0	1	0	0	3	1	2	2	9
6250	1	0	0	0	0	3	1	2	2	9
6500	1	0	0	1	0	3	1	2	2	9
6750	1	0	0	0	1	3	1	2	2	9
6810	1	0	0	1	1	3	3	2	2	9
7000	1	0	0	0	0	3	3	2	2	9
7155	1	0	0	0	0	3	3	1	2	9
7160	1	0	0	0	0	3	3	136	2	9
7250	1	0	0	0	0	3	3	136	2	9
7500	1	0	0	0	0	3	3	136	2	9
7505	1	0	0	0	0	4	3	136	2	9
7550	1	0	0	0	0	4	3	136	4	8
7750	1	0	0	0	0	4	3	136	4	8
8000	1	0	1	0	0	4	3	136	4	8
8160	1	0	1	0	0	4	3	1	4	8
8250	1	0	0	0	0	4	3	1	4	8
8405	1	0	0	0	0	4	3	3	4	8
8500	1	0	0	1	0	4	3	3	4	8
8750	1	0	0	0	1	4	3	3	4	8
9000	1	0	0	0	0	4	3	3	4	8
9155	1	0	0	0	0	4	3	1	4	8
9160	1	0	0	0	0	4	3	193	4	8
9250	1	0	0	0	0	4	3	193	4	8
9500	1	0	0	0	0	4	3	193	4	8
9505	1	0	0	0	0	5	3	193	4	8
9550	1	0	0	0	0	5	3	193	6	1
9750	1	0	0	0	0	5	3	193	6	1
10000	1	0	1	0	0	5	3	193	6	1
10105	1	0	1	0	0	5	6	193	6	1
10250	1	0	0	0	0	5	6	193	6	1

図 21-1

に得られる。命令部の5は、素子6の入力になり出力端子6をオンにする。時刻4000でC相パルスが発生し、素子26のゲートを開く。素子7の出力2のアドレス部の8が素子26、33を通り素子8の入力2に加えられ、素子8の出力に8番地の内容1が時刻4160に読み出される。Cパルスは、また素子22のゲートを開き、素子22の出力は、素子31を通り素子1(ACC)の入力1をオンにすると同時に素子

12のゲートを開きMEMOの出力値を素子32を通してACCの入力2へ導びく。ACCは、時刻4160にMEMOの古い値168が一度セットされるが、時刻4220に新しい値1がセットされ、1番地に書かれている命令の実行を終了する。以下各ステップも同じようにシミュレーションされており、図18の計算機は意図した通り動作していることを確かめた。

10500	C	0	0	1	0	5	6	193	6	1	
10750	C	0	0	0	1	5	6	193	6	1	SL 1
11000	C	0	0	0	0	5	6	193	6	1	
11155	C	0	0	0	0	5	6		6	1	
11160	C	0	0	0	0	5	6	32	6	1	
11250	C	0	0	0	0	5	6	32	6	1	
11500	C	1	0	0	0	5	6	32	6	1	
11505	C	1	0	0	0	6	6	32	6	1	
11550	C	1	0	0	0	6	6	32	1	0	
11750	C	0	0	0	0	6	6	32	1	0	
12000	C	0	1	0	0	6	6	32	1	0	
12250	C	0	0	0	0	6	6	32	1	0	
12500	C	0	0	1	0	6	6	32	1	0	
12750	C	0	0	0	1	6	6	32	1	0	JZ 0
13000	C	0	0	0	0	6	6	32	1	0	
13155	C	0	0	0	0	6	6	1	1	0	
13160	C	0	0	0	0	6	6	105	1	0	
13250	C	0	0	0	0	6	6	105	1	0	
13500	C	1	0	0	0	6	6	105	1	0	
13505	C	1	0	0	0	7	6	105	1	0	
13550	C	1	0	0	0	7	6	105	3	9	
13750	C	0	0	0	0	7	6	105	3	9	
14000	C	0	1	0	0	7	6	105	3	9	
14160	C	0	1	0	0	7	6	2	3	9	
14250	C	0	0	0	0	7	6	2	3	9	
14500	C	0	0	1	0	7	6	2	3	9	
14750	C	0	0	0	1	7	6	2	3	9	
14810	C	0	0	0	1	7	4	2	3	9	S 9
15000	C	0	0	0	0	7	4	2	3	9	
15155	C	0	0	0	0	7	4	1	3	9	
15160	C	0	0	0	0	7	4	5	3	9	
15250	C	0	0	0	0	7	4	5	3	9	
15500	C	1	0	0	0	7	4	5	3	9	
15505	C	1	0	0	0	8	4	5	3	9	
15550	C	1	0	0	0	8	4	5	0	5	
15750	C	0	0	0	0	8	4	5	0	5	
16000	C	0	1	0	0	8	4	5	0	5	
16250	C	0	0	0	0	8	4	5	0	5	
16500	C	0	0	1	0	8	4	5	0	5	
16510	C	0	0	1	0	5	4	5	0	5	
16750	C	0	0	0	1	5	4	5	0	5	J 5
17000	C	0	0	0	0	5	4	5	0	5	
17155	C	0	0	0	0	5	4	1	0	5	
17160	C	0	0	0	0	5	4	32	0	5	
17250	C	0	0	0	0	5	4	32	0	5	
17500	C	1	0	0	0	5	4	32	0	5	
17505	C	1	0	0	0	6	4	32	0	5	
17550	C	1	0	0	0	6	4	32	1	0	
17750	C	0	0	0	0	6	4	32	1	0	
18000	C	0	1	0	0	6	4	32	1	0	
18250	C	0	0	0	0	6	4	32	1	0	
18500	C	0	0	1	0	6	4	32	1	0	
18750	C	0	0	0	1	6	4	32	1	0	JZ 0
19000	C	0	0	0	0	6	4	32	1	0	
19155	C	0	0	0	0	6	4	1	1	0	
19160	C	0	0	0	0	6	4	105	1	0	
19250	C	0	0	0	0	6	4	105	1	0	
19500	C	1	0	0	0	6	4	105	1	0	
19505	C	1	0	0	0	7	4	105	1	0	
19550	C	1	0	0	0	7	4	105	3	9	
19750	C	0	0	0	0	7	4	105	3	9	
20000	C	0	1	0	0	7	4	105	3	9	
20160	C	0	1	0	0	7	4	2	3	9	
20250	C	0	0	0	0	7	4	2	3	9	
20500	C	0	0	1	0	7	4	2	3	9	
20750	C	0	0	0	1	7	4	2	3	9	

図 21-2

2081C	C	0	0	0	1	7	2	2	3	9	S	9
2100C	1	0	0	0	0	7	2	2	3	9		
21155	1	0	0	0	0	7	2	1	3	9		
2116C	1	0	0	0	C	7	2	5	3	9		
2125C	C	0	0	0	C	7	2	5	3	9		
2150C	C	1	0	0	0	7	2	5	3	9		
21505	C	1	0	0	0	8	2	5	3	9		
2155C	C	1	0	0	C	8	2	5	3	5		
2175C	C	0	0	0	0	8	2	5	3	5		
2200C	C	0	1	0	0	8	2	5	3	5		
2225C	C	0	0	0	C	8	2	5	3	5		
2250C	C	0	0	1	C	8	2	5	3	5		
2251C	C	0	0	1	C	5	2	5	3	5		
2275C	C	0	0	0	1	5	2	5	3	5	J	5
2300C	1	0	0	0	0	5	2	5	3	5		
23155	1	0	0	0	0	5	2	1	3	5		
2316C	1	0	0	0	C	5	2	32	3	5		
2325C	C	0	0	0	C	5	2	32	3	5		
2350C	C	1	0	0	C	5	2	32	3	5		
23505	C	1	0	0	0	6	2	32	3	5		
2355C	C	1	0	0	0	6	2	32	1	0		
2375C	C	0	0	0	C	6	2	32	1	0		
2400C	C	0	1	0	0	6	2	32	1	0		
2425C	C	0	0	0	C	6	2	32	1	0		
2450C	C	0	0	1	0	6	2	32	1	0		
2475C	C	0	0	1	1	6	2	32	1	0	JZ	0
2500C	1	0	0	0	C	6	2	32	1	0		
25155	1	0	0	0	0	6	2	1	1	0		
2516C	1	0	0	0	0	6	2	105	1	0		
2525C	C	0	0	0	0	6	2	105	1	0		
2550C	C	1	0	0	C	6	2	105	1	0		
25505	C	1	0	0	0	7	2	105	1	0		
2555C	C	1	0	0	0	7	2	105	3	9		
2575C	C	0	0	0	C	7	2	105	3	9		
2600C	C	0	1	0	C	7	2	105	3	9		
2616C	C	0	1	0	C	7	2	2	3	9		
2625C	C	0	0	0	0	7	2	2	3	9		
2650C	C	0	0	1	C	7	2	2	3	9		
2675C	C	0	0	1	1	7	2	2	3	9		
2681C	C	0	0	1	1	7	0	2	3	9	S	9
2700C	1	0	0	0	0	7	0	2	3	9		
27155	1	0	0	0	0	7	0	1	3	9		
2716C	1	0	0	0	0	7	0	5	3	9		
2725C	C	0	0	0	0	7	0	5	3	9		
2750C	C	1	0	0	0	7	0	5	3	9		
27505	C	1	0	0	0	8	0	5	3	9		
2755C	C	1	0	0	C	8	0	5	3	5		
2775C	C	0	0	0	C	8	0	5	3	5		
2800C	C	0	1	0	0	8	0	5	3	5		
2825C	C	0	0	0	0	8	0	5	3	5		
2850C	C	0	0	1	C	8	0	5	3	5		
2851C	C	0	0	1	0	5	0	5	3	5		
2875C	C	0	0	0	1	5	0	5	3	5	J	5
2900C	1	0	0	0	0	5	0	5	3	5		
29155	1	0	0	0	0	5	0	1	3	5		
2916C	1	0	0	0	0	5	0	32	3	5		
2925C	C	0	0	0	C	5	0	32	3	5		
2950C	C	1	0	0	0	5	0	32	3	5		
29505	C	1	0	0	0	6	0	32	3	5		
2955C	C	1	0	0	0	6	0	32	1	0		
2975C	C	0	0	0	0	6	0	32	1	0		
3000C	C	0	1	0	0	6	0	32	1	0		
3025C	C	0	0	0	0	6	0	32	1	0		
3050C	C	0	0	1	0	6	0	32	1	0		
3051C	C	0	0	1	0	0	0	32	1	0		
3075C	C	0	0	0	1	0	0	32	1	0	JZ	0

図 21-3

31000	1	0	0	0	0	0	0	32	1	0
31155	1	0	0	0	0	0	0	1	1	0
31250	C	0	0	0	0	0	0	1	1	0
31500	C	1	0	0	0	0	0	1	1	0
31505	C	1	0	0	0	1	0	1	1	0
31550	C	1	0	0	0	1	0	1	0	1
31750	C	0	0	0	0	1	0	1	0	1
32000	C	0	1	0	0	1	0	1	0	1
32250	C	0	0	0	0	1	0	1	0	1
32500	C	0	0	1	0	1	0	1	0	1
32750	C	0	0	0	1	1	0	1	0	1
33000	1	0	0	0	0	1	0	1	0	1
33160	1	0	0	0	0	1	0	168	1	1
33250	C	0	0	0	0	1	0	168	0	1
33500	C	1	0	0	0	1	0	168	0	1
33505	C	1	0	0	0	2	0	168	0	1
33550	C	1	0	0	0	2	0	168	5	8
33750	C	0	0	0	0	2	0	168	5	8
34000	C	0	1	0	0	2	0	168	5	8
34065	C	0	1	0	0	2	168	168	5	8
34160	C	0	1	0	0	2	168	3	5	8
34220	C	0	1	0	0	2	3	3	5	8
34250	C	0	0	0	0	2	3	3	5	8
34500	C	0	0	1	0	2	3	3	5	8
34750	C	0	0	0	1	2	3	3	5	8
35000	1	0	0	0	0	2	3	3	5	8
35155	1	0	0	0	0	2	3	1	5	8
35160	1	0	0	0	0	2	3	73	5	8
35250	C	0	0	0	0	2	3	73	5	8
35500	C	1	0	0	0	2	3	73	5	8
35505	C	1	0	0	0	3	3	73	5	8
35550	C	1	0	0	0	3	3	73	2	9
35750	C	0	0	0	0	3	3	73	2	9
36000	C	0	1	0	0	3	3	73	2	9
36160	C	0	1	0	0	3	3	2	2	9
36250	C	0	0	0	0	3	3	2	2	9
36500	C	0	0	1	0	3	3	2	2	9
36750	C	0	0	0	1	3	3	2	2	9
36810	C	0	0	0	1	3	5	2	2	9
37000	1	0	0	0	0	3	5	2	2	9
37155	1	0	0	0	0	3	5	1	2	9
37160	1	0	0	0	0	3	5	136	2	9
37250	C	0	0	0	0	3	5	136	2	9
37500	C	1	0	0	0	3	5	136	2	9
37505	C	1	0	0	0	4	5	136	2	9
37550	C	1	0	0	0	4	5	136	4	8
37750	C	0	0	0	0	4	5	136	4	8
38000	C	0	1	0	0	4	5	136	4	8
38160	C	0	1	0	0	4	5	3	4	8
38250	C	0	0	0	0	4	5	3	4	8
38405	C	0	0	0	0	4	5	5	4	8
38500	C	0	0	1	0	4	5	5	4	8
38750	C	0	0	0	1	4	5	5	4	8
39000	1	0	0	0	0	4	5	5	4	8
39155	1	0	0	0	0	4	5	1	4	8
39160	1	0	0	0	0	4	5	193	4	8
39250	C	0	0	0	0	4	5	193	4	8
39500	C	1	0	0	0	4	5	193	4	8
39505	C	1	0	0	0	5	5	193	4	8
39550	C	1	0	0	0	5	5	193	6	1
39750	C	0	0	0	0	5	5	193	6	1
40000	C	0	1	0	0	5	5	193	6	1
40105	C	0	1	0	0	5	10	193	6	1
40250	C	0	0	0	0	5	10	193	6	1
40500	C	0	0	1	0	5	10	193	6	1
40750	C	0	0	0	1	5	10	193	6	1

J 1

L 8

5. むすび

伝ばん遅延時間をもつ素子で構成された論理回路のシミュレーションについて考察し、シミュレーションプログラムを製作した。このプログラムを使用して、半加算加と小型計算機のシミュレーションを行なってみた。このシミュレータでは、系の構成論理素子が

- (1) 必ず伝ばん遅延時間をもつ
- (2) 論理素子の出力値は、時間とその論理素子への入力状態からフォートラン・プログラムにより計算できる。

という条件をもっていることを必要とする。また(1)、(2)の条件を満たす論理素子で構成されている系であれば、論理回路以上の系でもシミュレーションできるということである。

以上、本シミュレータについて次のような結果を得た。

- (1) IC等の論理素子で組まれる論理回路の設計、評価において、このシミュレータは非常に有効な道具となりうる。
- (2) 時間遅れをもつ論理素子で構成されている系を正確にシミュレーションできる。
- (3) 論理素子の定義を独立したサブルーチンで行なうという方式では、サブルーチンを作らなければならないという煩わしさはあるが、新しい機能をもつ論理素子を自由にシミュレータに組み入れることができるため、シミュレーションの範囲を大幅に広げることができる。

なお、本シミュレータは、バッチ処理での使用のために製作したが、将来図形入出力装置によるオンラインでの利用が可能になるよう拡張する予定である。

参 考 文 献

- 1) McClure, R. M.; A Programming Language for Simulating Digital Systems. J.ACM, pp.14~22, Jan., 1965
- 2) 高島, 津田, 加藤, 戸田, 中村, 高山; 論理構成のシミュレーション・プログラム, 情報処理, pp.64~72, March, 1963
- 3) 加藤, 戸田, 中村, 山田; 計算機を用いた計算機論理のデバッグについて, 情報処理, pp.73~82, March, 1963
- 4) 高島, 加藤, 高村, 新井, 菅野, 今出; 並列処理を用いた大容量高速論理シミュレータ, 情報処理, pp.263~272, Sept. 1966
- 5) Gwensbolyn G.H.; Computer-Aided Design: Simulation of Digital Design Logic, IEEE Trans., C-18, Jan., 1969
- 6) 荒牧, 柴田; 論理回路シミュレータGPLS-IIとその応用, 情報処理, pp.198~208, April 1970
- 7) 中西, 伊藤; コンピュータシミュレーション, SIMSCRIPTによるシミュレーション, 電子通信学会誌, pp.37~42, Jan. 1972

付 録

1 処理サブルーチンの構造

処理サブルーチンは次のような形式で書く。

SUBROUTINE, ルーチン名(NOWT, INPUT, II, IOU, IOUTT, NO, ITAD, NTTM, NTST, IST)

ここで,

NOWT; 整数型, 現在時間

INPUT; 整数型1次元配列名, 入力端子の状態

II; INPUTの大きさ

IOU; 整数型1次元配列名, 現在の出力端子の状態

IOUTT; 整数型1次元配列名, IOUに書かれている状態に成った時刻

NO; IOU IOUTTの大きさ

ITAD; 整数型1次元配列名, 定数テーブル, 定数表定義データによって定義されたものが書かれている。

NTTM; 整数型1次元配列名, 出力用テーブル, 出力端子の次の状態をこのテーブルに書き込む。大きさはNOになる。

NTST; 整数型1次元配列 NTTM に書き込んだ状態に成る時刻, ここに書かれる時刻は NOWT より大きくないと無効になる。大きさはNO

IST; 整数型, バッファ・エリア, 自由に使ってよい。

なおNOWTを現在時間より小さくしてRETURN

させると何もしなかったと解釈される。

図22に半加算器のシミュレーションに使ったサブルーチンNANDを示す。

2 共通データ領域

共通データはCOMMON エリアに書かれ次に示すようになっている。

COMMON/DCSCOM/NOWT, IEDT(401), NPTT(1501), IOU(601), INPT(501), ITAD(201), IWOR(602), NOTD, MOTD(200), IOSTIME, LTIME

出力指定データでのデータはMOTDに書かれる。MOTDに書かれた個数はNOTDに書かれる。MOTD(I)に書かれた出力端子の状態はIOU(MOTD(I)+1)に書かれている。

IOSTIME; ENDデータにおけるシミュレーション開始時刻が書かれている。

LTIME; ENDデータにおけるシミュレーション終了時刻が書かれている。

3 出力用ルーチンOUTPB

NOWTが更新される前にシミュレーション・ルーチンは出力用ルーチンOUTPBをCALLする。ルーチンOUTPBは必要に応じて適切なものを作る。シミュレーション・ルーチンからのデータはCOMMON/DCSCOM/で受け取る。図23に半加算器のシミュレーションに使った出力用ルーチンOUTPBを示す。

```

C NAND      HARP
            SUBROUTINE NAND
            (NOWT, INPUT, II, IOU, IOUTT, NO, ITAD, NTTM, NTST, IST)
            DIMENSION INPUT(1), IOU(1), ITAD(1), NTTM(1), NTST(1), IOUTT(1)
            ISH=ITAD(1)
            ISL=ITAD(2)
            IS1=NOWT-IOUTT(1)
            DO 600 I=1, II
            IF (INPUT(I), EQ, 0) GO TO 1
            CONTINUE
            IA=0
            IS=ISL
            IF (IS1.GE. ISH) GO TO 2
            IS=IS1*ISL/ISH
            GO TO 2
            IA=1
            IS=ISH
            IF (IS1.GE. ISL) GO TO 2
            IS=IS1*ISL/ISL
            IF (IS.LE. 0) IS=1
            NTTM(1)=NOWT+IS
            NTST(1)=IA
            RETURN
            END
    
```

図22 サブルーチンNANDのリスト

4 使用法

必要とする処理ルーチン、OUTPB をコーディングし、メイン・プログラムを作る。図24に半加算器のシミュレーションに使ったメイン・プログラムのリストを示す。サブルーチンDSDCRD はデータをカ

ード・リーダーからリードするルーチンで、パラメータ IERRR はデータにエラーがあるとき正の整数となる。サブルーチンDCSFL はシミュレーション・ルーチンの名前である。

```

COUTPB      HARP
             SUBROUTINE OUTPB
             COMMON/DCSCOM/NOVT ,IE,IEDT(100),IEDT1(100),IEDT2(100)
             C      ,IEDTT(100),NP,NPTT(100),NPTTD(500)
             C      ,IOJ,IOUT(200),ICUT1(200),IOUTT(200),IN,INPT(500)
             C      ,IT,ITAD(200),INP,NPTT1(400),INP1,NPTD1(200)
             C      ,NOTD,MOTD(200),ISIME,LTIME
             DIMENSION IUT(200),IUTX(200)
             DATA IUTX(1)/-1/
             500   FORMAT(1H0,I10,13(1X,I7))
             501   FORMAT(1H0,10X,13(1X,I7))
             IF(NOTD.LE.0) GO TO 50
             DO 600 I=1,NOTD
             J=MOTD(I)
             IUT(I)=IOUT1(J)
             600   CONTINUE
             IND=0
             DO 601 I=1,NOTD
             IF(IUT(I).EQ.IUTX(I)) GO TO 601
             IND=1
             IUTX(I)=IUT(I)
             601   CONTINUE
             IF(IND.EQ.0) GO TO 123
             IF(NOTD.GT.13)GO TO 11
             IL=NOTD
             GO TO 12
             11     IL=13
             IND=0
             12     WRITE(6,500) NOVT,(IUT(I),I=1,IL)
             IF(IND.NE.0) GO TO 123
             WRITE(6,501) (IUT(I),I=14,NOTD)
             GO TO 123
             50     CONTINUE
             123   RETURN
             END

```

図23 サブルーチンOUTPBの例

```

CMAIN       HARP
             DIMENSION K(3)
             DATA K/1,2,3/
             CALL DSDCRD(IERRR)
             IF(IERRR.NE.0) GO TO 123
             WRITE(6,500)(K(J),J=1,3)
             500   FORMAT(1H1,6X,4HTIME,7X,1HT,7X,1HA,7X,1HB,3(7X,I1),7X,1HS,
             C      7X,1HC)
             CALL OUTPB
             CALL DCSFL
             123   STOP
             END

```

図24 メインプログラムの例

航空宇宙技術研究所報告 391 号

昭和 49 年 10 月 発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町 1880
電話武蔵野三鷹(0422)47-5911(大代表) ●182

印刷所 株式会社 東京プレス
東京都板橋区桜川 2-27-12
