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　（航空宇宙技術研究所）

1　はじめに

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・フロップ等の基本論理素子から数ビット・フルアダ，各種レジスタさらには，数ビット並列処理ＣＰＵまで一つのチップで用意されるようになった。またゲートあたりの伝ぱん遅延時間が数ナノ秒という高速の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。このような高速・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困難である。
(2) 素子の遅延時間を考慮したシミュレーションが困難である。
(3) 論理回路を構成する論理素子が，アンド回路，オア回路，フリップ・フロップ等の基本素子に限定されていて，新らしい機能をもつ素子を新たにシミュレータに組込むのが困難である。

の欠点がある。これらの欠点を除いた論理回路シミュレーション・プログラムを成作した。このシミュレータでは，論理回路を構成する素子はＦＯＲＴＲＡＮ　サブルーチンという形でシミュレータに組込まれる。例えばＮＡＮＤという素子は，サブルーチンＮＡＮＤで定義される。したがって新らしい機能の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シミュレータに組込まれる。ここで定義される素子は必ず伝ぱん遅延時間をもっていることを必要とする。また従来の論理シミュレータの多くは，その扱う信号状態が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（2^{32}の状態を表現できる）に拡張した。このためこのシミュレータで扱える素子は論理素子のみでなく，自動制御系で使用されるようなブロック図で表わせるものをも含むことができる。

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の任意時刻tでの出力の状態を$f_i(t)$とかき，$f_i(t)$は次のような条件を満すものとする。

$$y_{min} \leqq f_i(t) \leqq y_{max}$$

$$|f_i(t) - f_i(t+\varepsilon)| = 0 \quad \text{あるいは} \quad y$$

εは任意，またyの集合Yは有限個の点で構成された集合とする。さらに$f_i(t)$を次のような関数で表現できるものとする。

$$f_i(t) = f\left(t, f_1(t-\tau_{i,1}), f_2(t-\tau_{i,2}), \cdots f_{i-1}(t-\tau_{i,i-1}), f_{i+1}(t-\tau_{i,i+1}), \cdots f_n(t-\tau_{i,n}), C_i(t)\right)$$

ここでCは$C \in Y$であり，f_iが変動する時刻にのみ変動可能，また$\tau > 0$

このような条件を満す関数fは，階段状になる。論理素子iの出力状態f_iに変動をよえる原因は，i以外の論理素子の出力状態の変動，あるいは時間の変動による。f_kの変動によるf_iへの影響は，f_kの変動より時間$\tau_{i,k}$後に生ずる。このτを伝ぱん遅延時間と呼ぶ。

以上のような性質をもつ論理素子で構成した論理回路では，任意の時刻t_0か

らtmにおける回路の状態は，有限個の時刻列T（t0, t1, t2 …… tm）とそれぞれの時刻における各論理素子の出力値で表現できる．時刻tℓ（1≦ℓ≦m）で何らかの原因によりfiの値が変動したとする．次に回路の状態が変動するのは，時刻tℓ+1である．tℓ<t<tℓ+1を満すtでは，各素子の出力状態は一定で変動しない．いまfjは，fiの変動により変動を受けるものとする．このときの伝ぱん遅延時間をτj,iとする．時刻tℓ後fjが変動するのは，時刻tℓ+τj,iである．

ここで時刻tℓ+τj,iは

$$t_\ell + \tau_{j,i} \geqq t_{\ell+1}, \qquad t_\ell + \tau_{j,i} \in T$$

が成り立っている．tℓ+τj,iを次の二つの場合に分けて考える．

(1) $t_\ell + \tau_{j,i} = t_{\ell+1}$ のとき

このときtℓ+1の時刻にfjは変動する．このときのfjの値は，tℓ<t'<tℓ+1を満すt'における回路の各素子の出力状態，Cj(t')およびtℓ+1で決定できるものとする．

(2) $t_\ell + \tau_{j,i} > t_{\ell+1}$ のとき

$t_k = t_\ell + \tau_{j,i}$ とおく．時刻tℓ+1, tℓ+2, …… tk-1で変動する素子の一つをfgとする．

(i) fgがfjに直接影響を与えないとき

tℓ+1≦t≦tk-1でfjは，他の素子からの影響を受けない．したがってtℓ+1, tℓ+2 …… tk-1での回路の状態の変動は無視できる．よってこの場合は(1)に帰着できる．

(ii) fgがfjに影響を与えるとき

tℓ+1, tℓ+2, …… tk-1においてfjに最初に影響を与える時刻をtpとする．fgによるfjの伝ぱん遅延時間をτj,g，tℓ≦t'<tℓ+1, tp≦t''<tp+1なる時刻をt', t''とする．

(a) $t_k < t_p + \tau_{j,g}$ のとき

t=tkでfjの値をt'における各素子の出力状態，Cj(t')およびtkで決定される値とする．t=tp+τj,gで，fjの値をt''における各素子の出力状態，Cj(t'')およびtp+τj,gで決定される値とする．ただし，tk<t<tp+τj,gにおいて変動するfがfjに影響を与えるときは，その影響を受ける．

(b) $t_k \geqq t_p + \tau_{j,g}$ のとき

t=tp+τj,gで，fjの値をt''における各素子の出力状態，Cj(t'')およびtp+τj,gで決定される値とする．このときfjはfiからの影響を無視する．

このようにモデル化された系では，系を構成するf，τおよびfの初期値が与えられればシミュレーションができる．

2.2 素子

素子の出力はfに相当する．素子の出力を決定するパラメータは，時間t，入力状態I（これは他の素子の出力値となっている．），内部状態cである．Iの変動によりfが変動するときの遅れ，すなわち伝ぱん遅延時間τをtとは独立なパラメータとしたとき，時間のパラメータを含む素子を発振器と呼ぶことにする．

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Iが変動することが必要条件となる．入力端子をもった素子では，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（τに相当）を過渡状態，それ以外を定

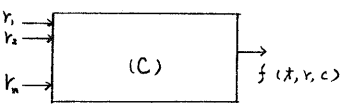

図1 素子のブロック表現

常状態という。すなわち素子は

入力状態の変化 ⟶ 過渡状態 ⟶ 定常状態

という経過でその出力が変動する．素子は図1のようなN入力端子をもつブロックで表現する．これらの素子で構成される系は次の条件を必要とする。

(1) 素子の出力端子は，いくつかの他の素子の入力端子に結合しても良いが，自分自身の入力端子あるいは他の出力端子と結合してはいけない。

(2) 系の内には，少なくともーつは発振器あるいは発振器に相当する回路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

プログラムは共通データ領域を中心に構成し，入出力ルーチン，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．

3.1. 共通データ領域

共通データ領域には，各素子間の関係，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が確保されている．シミュレーションすべき系内の素子の出力端子，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・データはテーブル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IOUTに，入力端子に関するデータはテーブルINPTに，過渡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。IOUT，INPT，NPTTの各テーブルは次に示すような構造になっている。

IOUTテーブル (IC, K, G, IN_1, IN_2, ……, IN_n)

- IO ; 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，そうでないときは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, OT)

- E ; 素子名
- OT ; 接続されている出力端子の情報が書かれているIOUTテーブルの場所

NPTTテーブル (NT, N, NO)

- 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IOUTテーブルの場所
- NO ; 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

現在の時刻をNOWTとする．シミュレーションはNOWT=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。まず全素子に対してそれぞれの処理ルーチンがCALLされ，各素子に与えられた入力条件がその素子の出力値を満足しているかどうかが調べられる．満足していない場合は，その素子の伝ぱ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．発振器ルーチンのうちの少なくてもーつは，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ければならない。そうであれば，NOWT=0での操作が終ったとき，テーブルNPTTには，

少なくとも一つの情報がストアされている。このような状態にし，以下のようにシミュレーションを実行する。PNTTにストアされたデータの中でNOWT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）．そのときのデータが NT′，N′，NO′ であったとする．ここで新たにNT′をNOWTとし時間を進め，テーブルIOUT の中N′番目のK，IO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) IO をNo′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NOWT 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になる時刻を求める。処理ルーチンの結果は，IOUT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．

3.3. 素子処理ルーチン

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，現時間NOWT，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NOWT における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は，出力端子がその値に変る時刻（この時刻は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，発振素子ルーチン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，入力条件の変動によってもCALL される．

3.4. データ入力ルーチン

データ入力ルーチンは，シミュレーションすべき系を記述したデータを読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，共通データ領域に書込むためのルーチンである。このルーチンであつかうデータは，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，出力指定データそれにENDデータの五種類である。各データはエレメントと仕切記号で構成し，エレメントとエレメントの間には仕切記号を入れる．仕切記号は，ブランク，，，／，（，）である．

(1) 素子定義データ

素子番号　処理ルーチン名　OTデータ　NIデータ　NOデータ　Tデータ
の形をしている．

素子番号　；5桁以内の整数．他の素子との識別をするためのもので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；素子が処理されるルーチン名

OTデータ；OT／出力端子番号n（初期値）　{nと接続されている入力端子}

2個の出力をもつ素子では2個のOTデータが必要．1出力のときは／nは必要ない．初期値が0のときは初期値を指定しなくてもよい．入力端子は

素子番号／端子番号n　　という形で書く．入力端子が1つしかない素子，あるいはP入力の端子があってもP個全てが対等な場合は　／n　は省略してもよい．

NIデータ；NI　入力端子数，NIデータのない素子は発振素子とみなされる．また入力端子数の次に　G　を付けたものも発振素子となる．

NOデータ；NO　出力端子数

Tデータ　；T　定数表識別番号

NI，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OT，NI，NO，Tの各データは任意の順序で並んでいてもよい．

(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，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

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今迄読んだデータを解読し，共通データ領域にデータを書込む．

データが一枚のカードに書きされないときは，最後にCを書く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よい．なお上記での{　}の記号は列を表わしている．列内の仕切記号はブランクあるいは，である．

4. 使用例

4.1. NANDゲートICによる半加算器のシミュレーション

ICで半加算器を設計する．半加算器の真理値表は図2の通りである．これは5個のNANDゲートを使用して図3の回路で実現できる．この回路をシミュレーションしてみる．ここでのNANDゲートは，負い論理

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

図2　半加算器の真理値表

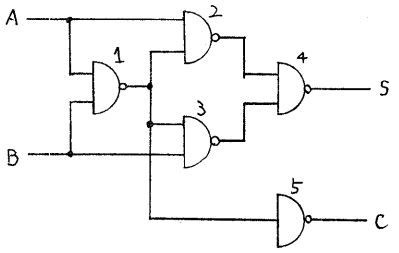

図3　半加算器の一例

```
      SUBROUTINE NAND(NOWT,INPUT,II,IOUT,IOUTT,NOO,ITAD,NTTM,NTST,IST)
      DIMENSION INPUT(1),IOUT(1),ITAD(1),NTTM(1),NTST(1),IOUTT(1)
      DO 10 I=1,II
      IF (INPUT(I).EQ.0) GO TO 1
   10 CONTINUE
      IA=0
      IS=ITAD(2)
      GO TO 2
    1 IA=1
      IS=ITAD(1)
    2 NTTM(1)=NOWT+IS
      NTST(1)=IA
      RETURN
      END
```

図 6

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
70	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
4070	1	1	1	0	1	1	1	1
4090	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

```
      SUBROUTINE NAND
      ISH=ITAD(1)
      ISL=ITAD(2)
      ISl=NOWT-IOUTT(1)
      DO 10 I=1,II
      IF(INPUT(I).EQ.0) GO TO 1
   10 CONTINUE
      IA=0
      IS=ISL
      IF(ISl.GE.ISH) GO TO 2
      IS=ISl*ISL/ISH
      GO TO 2
    1 IA=1
      IS=ISH
      IF(ISl.GE.ISL) GO TO 2
      IS=ISl*ISH/ISL
    2 IF(IS.LE.0) IS=1
      NTTM(1)=NOWT+IS
      NTST(1)=IA
      RETURN
      END
```

図 9

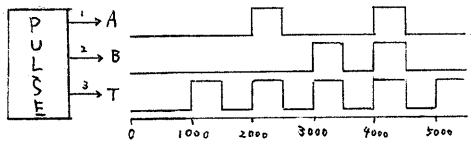

図4　パルス発生器の出力波形

```
1 NAND OT 2 3 5
2 NAND OT 4
3 NAND OT 4
4 NAND
5 NAND NI 1
6 PULSE OT/1 1 2 OT/2 1 3 NO 9 T 3
OUTPUT 6/3 6/1 6/2 1 2 3 4 5
T 1 50 20
T 3 1000 5000 500 4 1 2 3 128
NAND NO 1 NI 2 T 1
END 0 5000
```

図5　半加算器入力データ

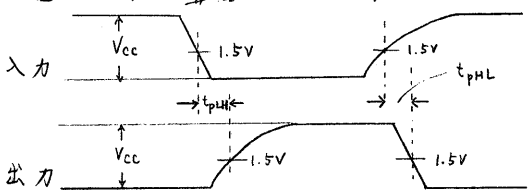

図8　ICの伝ぱん遅延時間の定義

図 7

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 10

値Lから高い論理値Hになる伝ぱん遅延時間t_{pLH}を50ｎ秒，逆の場合の伝ぱん遅延時間t_{pHL}を20ｎ秒であるとする．半加算器のA，Bに加えるパルスは，パルス発生器PULSEで発生させる．パルス中500ｎ秒の図4に示すパルスが発生する．入力データは，図5のように書ける．ここでNANDルーチンを図6に示すものを使用するとシミュレーション結果は，図7に示すようになる．この結果では，時間4020に伝ぱん遅延時間によるハザードがNAND2，3に発生し，これがNAND4に伝ぱんされている．しかし実際にICで図3の回路を組み，実験してみるとNAND2，3にはハザードが生ずるが，NAND4にはそれが伝ぱんされない．実際のIC(三菱M5946P)でのt_{pLH}，t_{pHL}の定義は図8のようになされている．Lの状態は0～1.5V，Hは1.5V～V_{cc}を示しt_{pLH}は0から1.5Vになるまでの時間を，t_{pHL}は，V_{cc}から1.5Vになるまでの時間で定義される．したがって，t_{pLH}は0の状態でないLからHに変るときは，カタログで定義されたものより短かくなる．t_{pHL}も同様である．このことを考慮してNANDルーチンを作ったのが図9で，それによるシミュレーション結果が図10である．

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命令をもつ計算機を設計し，設計どうり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，命令部に3ビット，アドレス指定に5ビットを当てたものとなっている．命令は次の8つである．

(1) J；無条件ジャンプ
(2) JZ；アキュムレータ(AC)が0ならジャンプ
(3) A；アドレスで指定された番地の内容とACの内容を加算しACに入れる．
(4) S；ACの内容からアドレスで指定された番地の内容を引きACに入れる．
(5) ST；ACの内容をアドレスで指定された番地に格納する．
(6) L；アドレスで指定された番地の内容をACに入れる．
(7) SL；アドレス部の内容だけシフトする．
(8) 入出力命令

図11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素子は次の11種類である．

(1) SCC8；入力端子1がオンのとき入力端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) ACC8；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)のデータ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(5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) ADDER8；入力端子3がオンのとき入力端子1と2(それぞれ8ビット)の内容を加算して出力する．
(4) SBT8；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なら出力をオンにする．
(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オンにする．

(7) REG1 ； 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) MEMO32 ； 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・レジスタのネすメモリ番地に書込まれる．入力端子5がオンのときアドレス・レジスタのネすメモリ番地の内容が出力レジスタに出力される．

(9) AND8 ； 8ビット・パラレル・アンド・ゲート．

(10) OR8 ； 8ビット・パラレル・オア・ゲート

(11) PULSR ； パルス発振器．出力端子を8つもち，1，3，5，7の出力パルスをそれぞれA，B，C，Dパルスと呼ぶ．各パルスは次のような動作を行なう．A；命令読出し，B；命令解読，SCCの内容に1を加える．C；オペランド読出し，またはJ，JZ SL命令では実行．D；命令実行．

メモリの初期値を右のようにし，SCCに0をセットする．図13はこのような初期値でシミュレーションを実行させた結果である．

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

メモリの初期値

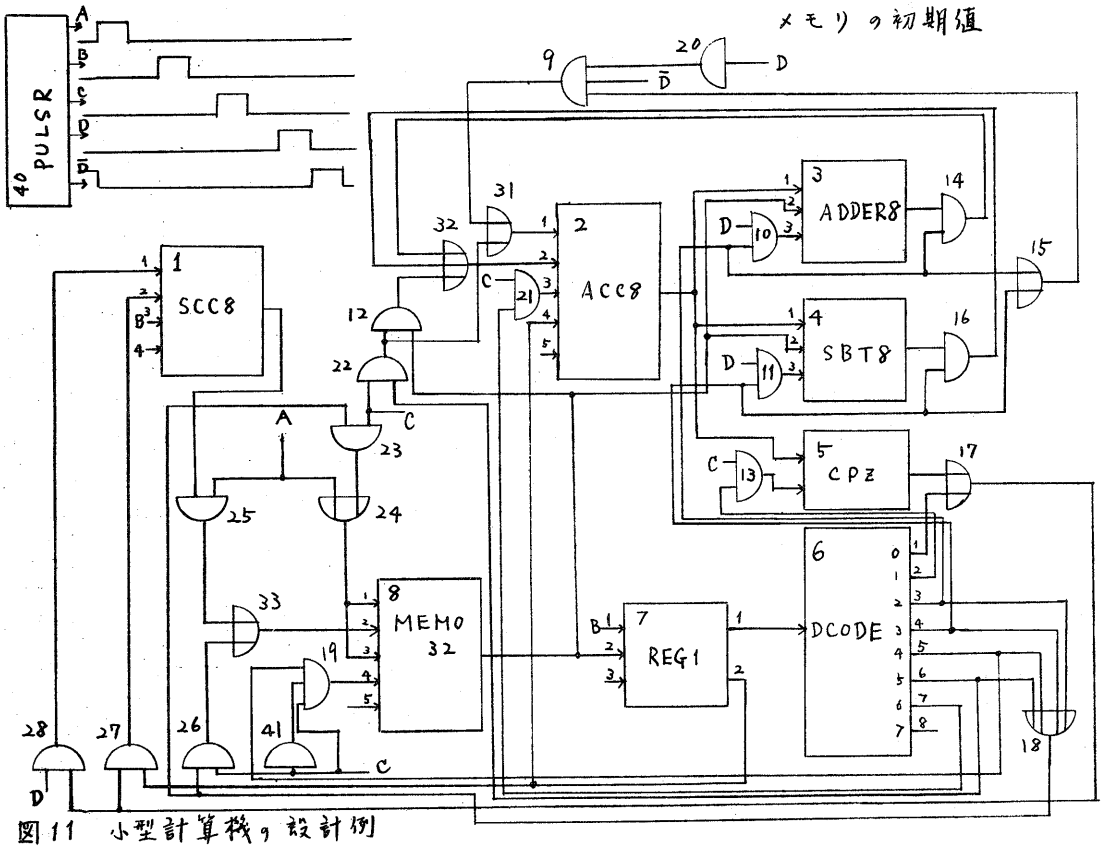

図11 小型計算機の設計例

```
 1 SCC8   OT(0) 25 NO 1 NI 4 T 1
 2 ACC8   OT(0) 3/1 4/1 5/1 8/5 NO 1 NI 5 T 2
 3 ADDER8 OT 14 NO 1 NI 3 T 6
 4 SBT8   OT 16 NO 1 NI 3  T 6
 5 CPZ    OT 17 NO 1 NI 2  T 7
 6 DCODE  OT/1 17     OT/2 13 OT/3 18,10,14,15 OT/4 18,11,16,15 OT/5 18,19  C
          OT/6 18,22 OT/7 21 NO 8 NI 1 T 2
 7 REG1   OT/1 6 OT/2 2/4,26,27 NO 2 NI 3 T 2
 8 MEMO32 OT 7/2,12,4/2,3/2 NO 1 NI 5 T 4
 9 AND8   OT 31 NO 1 NI 3 T 1
10 AND8   OT 3/3 NO 1 NI 2 T 1
11 AND8   OT 4/3 NO 1 NI 2 T 1
12 AND8   OT 32 NO 1 NI 2 T 1
13 AND8   OT 5/2 NO 1 NI 2 T 1
14 AND8   OT 32    NO 1 NI 2 T 1
15 OR8    OT 9 NO 1 NI 2 T 1
16 AND8   OT 32 NO 1 NI 2 T 1
17 OR8    OT 27,28 NO 1 NI 2 T 1
18 OR8    OT 26,23 NO 1 NI 4 T 1
19 AND8   OT 8/4 NO 1 NI 3 T 1
20 OR8 OT 9 NO 1 NI 1 T 6
21 AND8   OT 2/3 NO 1 NI 2 T 1
22 AND8   OT 12,31 NO 1 NI 2 T 1
23 AND8   OT 24 NO 1 NI 2 T 1
24 OR8    OT 8/1,8/3 NO 1 NI 2 T 1
25 AND8   OT 33 NO 1 NI 2 T 1
26 AND8   OT 33 NO 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    OT 2/2 NO 1 NI 3 T 1
33 OR8    OT 8/2 NO 1 NI 2 T 1
40 PULSR OT/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 0 T 5
41 OR8    OT 19 NO 1 NI 1 T 2
T 1  5
T 2  50
T 3  100
T 4  150
T 5  1000,250, 1000000
T 6  15
T 7  300
```

図12　図11のカード

TIME	A	B	C	D	*D	SCC	ACC	MEMORY	O	L	
0	0	0	0	0	0	0	0	0	0	0	
1000	1	0	0	0	0	0	0	0	0	0	
1155	1	0	0	0	0	0	0	1	0	0	
1250	0	0	0	0	0	0	0	1	0	0	
1500	0	1	0	0	0	0	0	1	0	0	
1505	0	1	0	0	0	1	0	1	0	0	
1550	0	1	0	0	0	1	0	1	0	1	
1750	0	0	0	0	0	1	0	1	0	1	
2000	0	0	1	0	0	1	0	1	0	1	
2250	0	0	0	0	0	1	0	1	0	1	
2500	0	0	0	1	0	1	0	1	0	1	
2750	0	0	0	0	1	1	0	1	0	1	J 1
3000	1	0	0	0	0	1	0	1	0	1	
3160	1	0	0	0	0	1	0	168	0	1	
3250	0	0	0	0	0	1	0	168	0	1	
3500	0	1	0	0	0	1	0	168	0	1	
3505	0	1	0	0	0	2	0	168	0	1	
3550	0	1	0	0	0	2	0	168	5	8	
3750	0	0	0	0	0	2	0	168	5	8	
4000	0	0	1	0	0	2	0	168	5	8	
4065	0	0	1	0	0	2	168	168	5	8	
4160	0	0	1	0	0	2	168	1	5	8	
4220	0	0	1	0	0	2	1	1	5	8	
4250	0	0	0	0	0	2	1	1	5	8	
4500	0	0	0	1	0	2	1	1	5	8	
4750	0	0	0	0	1	2	1	1	5	8	L 8

5000	1	0	0	0	0	2	1	1	5	8	
5160	1	0	0	0	0	2	1	73	5	8	
5250	0	0	0	0	0	2	1	73	5	8	
5500	0	1	0	0	0	2	1	73	5	8	
5505	0	1	0	0	0	3	1	73	5	8	
5550	0	1	0	0	0	3	1	73	2	9	
5750	0	0	0	0	0	3	1	73	2	9	
6000	0	0	1	0	0	3	1	73	2	9	
6160	0	0	1	0	0	3	1	2	2	9	
6250	0	0	0	0	0	3	1	2	2	9	
6500	0	0	0	1	0	3	1	2	2	9	
6750	0	0	0	0	1	3	1	2	2	9	
6810	0	0	0	0	1	3	3	2	2	9	A 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	0	0	0	0	0	3	3	136	2	9	
7500	0	1	0	0	0	3	3	136	2	9	
7505	0	1	0	0	0	4	3	136	2	9	
7550	0	1	0	0	0	4	3	136	4	8	
7750	0	0	0	0	0	4	3	136	4	8	
8000	0	0	1	0	0	4	3	136	4	8	
8160	0	0	1	0	0	4	3	1	4	8	
8250	0	0	0	0	0	4	3	1	4	8	
8405	0	0	0	0	0	4	3	3	4	8	
8500	0	0	0	1	0	4	3	3	4	8	
8750	0	0	0	0	1	4	3	3	4	8	ST 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	0	0	0	0	0	4	3	193	4	8	
9500	0	1	0	0	0	4	3	193	4	8	
9505	0	1	0	0	0	5	3	193	4	8	
9550	0	1	0	0	0	5	3	193	6	1	
9750	0	0	0	0	0	5	3	193	6	1	
10000	0	0	1	0	0	5	3	193	6	1	
10105	0	0	1	0	0	5	6	193	6	1	
10250	0	0	0	0	0	5	6	193	6	1	
10500	0	0	0	1	0	5	6	193	6	1	
10750	0	0	0	0	1	5	6	193	6	1	SL 1
11000	1	0	0	0	0	5	6	193	6	1	
11155	1	0	0	0	0	5	6	1	6	1	
11160	1	0	0	0	0	5	6	32	6	1	
11250	0	0	0	0	0	5	6	32	6	1	
11500	0	1	0	0	0	5	6	32	6	1	
11505	0	1	0	0	0	6	6	32	6	1	
11550	0	1	0	0	0	6	6	32	1	0	
11750	0	0	0	0	0	6	6	32	1	0	
12000	0	0	1	0	0	6	6	32	1	0	
12250	0	0	0	0	0	6	6	32	1	0	
12500	0	0	0	1	0	6	6	32	1	0	
12750	0	0	0	0	1	6	6	32	1	0	JZ 0
13000	1	0	0	0	0	6	6	32	1	0	
13155	1	0	0	0	0	6	6	1	1	0	
13160	1	0	0	0	0	6	6	105	1	0	
13250	0	0	0	0	0	6	6	105	1	0	
13500	0	1	0	0	0	6	6	105	1	0	
13505	0	1	0	0	0	7	6	105	1	0	
13550	0	1	0	0	0	7	6	105	3	9	
13750	0	0	0	0	0	7	6	105	3	9	
14000	0	0	1	0	0	7	6	105	3	9	
14160	0	0	1	0	0	7	6	2	3	9	
14250	0	0	0	0	0	7	6	2	3	9	
14500	0	0	0	1	0	7	6	2	3	9	
14750	0	0	0	0	1	7	6	2	3	9	
14810	0	0	0	0	1	7	4	2	3	9	S 9
15000	1	0	0	0	0	7	4	2	3	9	
15155	1	0	0	0	0	7	4	1	3	9	
15160	1	0	0	0	0	7	4	5	3	9	
15250	0	0	0	0	0	7	4	5	3	9	

図 13　小型計算機のシミュレーション結果