U.D.C. 72.011

清水建設研究所報
第4号昭和39年10月

直交図面の記号法と電算処理（2）

清水　達雄

目　次

2.　最小寸法図の自動図示

§ 9.　問題．起・終線の決定

さて，当面の課題は，正則な（＝十字交差のない）長方形分割図Ⓠに対して，その

正則用記号　→　最小寸法図

を，電子計算機に実行させること．さらにしぼって，自動図示の直接的制御に充分な，体系的情報として，Ⓠの

線 I の，**方向・起線・終線・階数**の一覧表

を算出する，算法をのべること．

与えられる，正則用記号とは，2進数字列

$$DA\langle Ⓠ\rangle=\cdots\cdots DA\langle I\rangle\cdots\cdots$$

ただし

$$DA\langle I\rangle=D\langle I\rangle A\langle I\rangle$$

$$D\langle I\rangle=\begin{cases}0 & I\text{が横線}\\ 1 & I\text{が縦線}\end{cases}$$

$$A\langle I\rangle=*\cdots\cdots*1$$

$*\cdots\cdots*$　は，**枝配列**で

$$*=\begin{cases}0 & \text{枝が後枝}\\ 1 & \text{枝が前枝}\end{cases}$$

こういうものを，内部分割線

$I=2,\ \cdots$

に対して作り，**続け書き**したもの．

$DA\langle 2\rangle,\ DA\langle 3\rangle,\ \cdots$

と，わかち書きされているわけではない．

定義33　この2進記号列の，左からM番目の数字を

$Q[M]$

と，略記する．

つまり，与えられているのは，見掛け上は単に

$Q[1]$，$Q[2]$，…

のような列で，これを解読してゆく．基本の操作は

一字一字の，拾い読み

なのだけれども，そこに，段落が，もうけられる．

枝終止符 1，の識別で一段落

その間に，幾何学的にいえば，§3．でのべた

中間図の復原

が，一段階進められる．それと対応して，算法上では，まず，つぎにあげる「線関数」が，拡張・更新される．

定義34　つぎの関数記号を導入する．

$D[J]$　　方向

$B[J]$　　起線番号

$E[J]$　　終線番号

$S[J]$　　席数

変数のJは線番号で，第I段落では，変域が

$J=0$，1，…，I

また，上記のうち

DとBは，初設定のまま変らないが，

EとSは，段落によって値が更新される．

これだけでは，定義になってないが，まず

定義35　第1段落として，前辺に対し

$D[0]=0$　　上辺0は横線

$D[1]=1$　　左辺1は縦線

$B[0]=1$　　上辺の起線は左辺

$B[1]=0$　　左辺の起線は上辺

$\left.\begin{array}{l}E[0]=0\\E[1]=0\end{array}\right\}$　前辺の終線は未定（後辺），定義13を参照

$\left.\begin{array}{l}S[0]=\infty\\S[1]=\infty\end{array}\right\}$　ただし∞は充分大きい数の略記，定理2系の証明を参照

これを，準備段階として，記号列

$Q[1]$，$Q[2]$，…

の解読にとりかかる．……　第2段落おわり．

つぎに，おなじく，……　第3段落おわり．

……………………………………………………

いま，第$I-1$段落までおわったものとしよう．

定義36　第$I-1$段落までおわったとき，記号列が

$Q[1]$，…，$Q[M-1]$

まで解読されているものとする．記号列の残部

$Q[M]\cdots\cdots=T[I]$

を，こんどの，**記号列尾部**とよぶ．T：Tail.

それが，空な列なら，解読は終り．さもなければ

定義37　記号列尾部の，先頭にある

$Q[M]=D[I]$

で，Iの方向Dを定める．

つぎに，Iの起線は，§3．の補題1にしたがい

定義38　Iと方向反対で，席数が正のJ，つまり

$D[J]\neq D[I]$，　$S[J]>0$

で，番号最大のもの，いいかえれば

$J=I-1$，$I-2$，…

と番号をへらしながら探し，最初に出会うのをBとし

$B[I]=B$

これにともない，

$S[B]-1$　を　$S[B]$

の新しい値とする．

それから，おなじく補題2にしたがい

定義39　見つかった起線Bから，こんどは

$J=B+1$，$B+2$，…

と番号をましながら

$D[J]\neq D[I]$，$E[J]=0$

のようなJたちを探し，出会ったら，改めて

$E[J]=I$

そのようなJの箇数を，Fとすれば，定義からいって

$F=I$の前枝の数

このFから，§6．によれば，枝終止符が定められた．

枝終止符

$=A\langle I\rangle\cdots\cdots$のような列中の$(F+1)$番目の1

定義40　記号列尾部の2番目以下

$Q[M+1]\cdots\cdots$

を，$(F+1)$番目の1，で区切り，その部分の

0の数$=S[I]$

§ 10. その ALGŌL 表示

以上で，帰納的な定義の，輪がとじた．定義37〜40をALGŌL 風に書き表わしてみよう．略式だけれども

```
        D:=D[I]:=Q[M];
        J:=I-1;  go to  L2;
L 1:    J:=J-1;
L 2:    if  D[J]=D  then go to  L1;
        S:=S[J]-1;
        if  S<0  then go to  L1;
        S[J]:=S;
        B[I]:=J;
```

これで起線がきまり，その席数が改新される．つぎは

```
        F:=0;
L 3:    J:=J+1;
        if  J=I  then go to  L4;
        if  D[J]=D  then go to  L3;
        if  E[J]>0  then go to  L3;
        E[J]:=I;
        F:=F+1;  go to  L3;
```

これが終線の指定と，前枝数の算出で，つぎに

```
L 4:    S:=0;  go to  L6;
L 5:    S:=S+1;
L 6:    M:=M+1;
        if  Q[M]=0  then go to  L5;
        F:=F-1;
        if  F≥0  then go to  L6;
        S[I]:=S;
```

これで，全部すんだ．

ただし実は，Fの計算は不要なので，それには，中の部分と後の部分との，順序を逆にすればよい．

```
        S:=0;  go to  L4;
L 3:    S:=S+1;
L 4:    M:=M+1;
        if  Q[M]=0  then go to  L3;
L 5:    J:=J+1;
        if  J=I  then go to  L6;
        if  D[J]=D  then go to  L5;
        if  E[J]>0  then go to  L5;
        E[J]:=I;  go to  L4;
L 6:    S[I]:=S;
```

こうしておくと，F番目までの前枝，$Q=1$ に対しては終線Eが指定され，$F+1$番目の $Q=1$，枝終止符ではJが見当たらず，$=I$となって，L6 に出る．

なお以上に，首・尾をつける必要がある．まず首部，

```
        D[0]:=B[1]:=
        E[0]:=E[1]:=0;
        D[1]:=B[0]:=1;
        S[0]:=S[1]:=∞;
        I:=2;  M:=1;
L 0:    if  T[M]=0  then go to  L7;
```

記号列尾部Tは，空でない限り，枝終止符1で終る．だから，記号列を2進小数と見なし，記号のつきた末尾に

0 0……

がついているものとした場合に，

$T=0$ が，空な列に相当

他方また，前記の末尾，L6 に続けて

```
        I:=I+1;  M:=M+1;
        go to  L0;
```

をおき，帰納的定義を進める．最後に記号列がつきての

```
L 7:    G:=0;
        D[I]:=1;  B[I]:=0;
L 8:    J:=0;  go to  L10;
L 9:    J:=J+1;
        if  J=I  then go to  L11;
L10:    if  D[J]=D  then go to  L9;
        if  E[J]>0  then go to  L9;
        E[J]:=I;  go to  L9;
L11:    if  G<0  then go to  L12;
        G:=G-1;  I:=I+1;
        D[I]:=0;  B[I]:=1;
        E[I]:=I-1;  go to  L8;
L12:    S[0]:=S[1]:=0;
```

ここで，Gは門番のようなもの．はじめ右辺Iの

方向Dは縦，起線Bは0の上辺

とし，$J=0$から，L 5以下と似た方式で，終線指定．ついでL11へ出て，Gを1へらしてから，下辺の

方向は横，起線は左辺，終線は右辺

とし，$J=0$から，ふたたび終線指定．こんどはGが負だからL12へ出る．つまり，L 8以下の終線指定を，ちょうど2回行うための，小細工がG．この種の小細工を使えば，L 9以下とL 5以下との，一本化もできる．なおL12，前辺の席数を0，は直接必要でないけど，

すべての $S[I]=0$

となるから，Sの記憶場所の転用に便利．

ともかく以上で，2進記号列から，すべての線の

方向・起線・終線

の定められることが，わかった．

§ 11. 区 間 席 数

さて，階数の決定のためには，すこしく準備がいる．前掲のプログラムを，まず精密化しなければならない．あれでは，せっかく

　　枝配列

が与えられているのに，単に

　　後枝数

が与えられたのと，おなじ結論しかえられてない．というのは，いま，枝配列から，前枝記号相当の1をすて，

　　後枝数だけの0，および枝終止符1

を残したもの，で考えてみよう．

§9. の算法にしたがえば，それでも

　　前枝への終線指定

は，正しく実行される．それで前枝数Fもわかるが，

　　$F+1$　でなく，単に　1

番目の1が枝終止符，ということで，わかち書きができ

　　その部分の0の数＝後枝数＝$S[I]$

これで，席数の初回設定ができる．それで，

　　方向・起線・終線

が，正しく定められる．ここまでは，前枝記号なしでも行けること，すでに §3. でのべられている．

では，前枝記号がある場合に，どういう情報が，なおえられるのだろうか．§10. で，算法本体の

　　中部と後部との順序変更

を行った．その改良プログラムを見てみよう．

L 3:　$S:=S+1$

で，席数の積算がされている．それは

　　$Q[M]\neq 0$，したがって $=1$，つまり前枝

の出現ごとに，中断されている．だから実は

　　隣接する前枝間の，後枝数

が，算出されている．のに，それをただ合算している．

定義41　線Iのすべての前枝を，順に

　　$J_1/J_2/\cdots/J_n$

とし，これになお，起・終線

　　$B[I]=J_0$,　　$E[I]=J_{n+1}$

を加えると，Iのかってな後枝Xに対し

　　$J_{i-1}/X/J_i$

のような J_i が，確定する．このとき，後枝

　　Xは，**区間** $J_{i-1}\sim J_i$ に属する

という．また，そのようなXの箇数を，この区間での，**区間後枝数**とよぶ．

定義42　区間は，一般には，その後端線で代表させ，

　　区間 $J_{i-1}\sim J_i$　を　J_i **担当区間**

などとよぶ．ただし，

　　$J_i=E[I]$

の場合には，とくに，I担当区間，のようによぶ．

定義43　中間図で，分割線Iをふくむものについて，Iの前枝Jの担当区間の，もとの図での区間後枝数と，その図でとの，差を，J担当の**区間席数**とよんで，つぎのように表わす．

　　$SS[J]$　　むしろ単に　$S[J]$

このように記号を略しても，特別の困難は起らない．もともと，J担当の区間席数がいわれるのは，

　　$E[J]=I$

が定められているときで，そのとき，ふつうの席数

　　$S[J]=0$

こうきまっているから，そのいわば空き家を，利用してかまわない．ただし，前記のプログラムで

L 2:　if　$D[J]=D$　then go to　L1;

のあとに

　　　if　$E[J]>0$　then go to　L1;

を，おぎなっておく．一般にいって

　　$E[J]=0$　かつ　$S[J]>0$

が，元来のいみでの，$S>0$に一致する．

なお，I自身の担当区間については，前枝Jの担当区間が，すべて「満席」つまり

　　$S[J]=0$

となったときに，本来のいみの席数

　　$S[I]$

が，問題の区間の区間席数に当たるものになる．

定義44　区間席数0のは度外視して，もっとも前方にある区間を，待機区間，その担当線を**待機線**とよんで

　　$W[I]$

で表わす．後枝数0のときは，考えない．W: Wait.

定義45　待機線の前方，もっとも近くに，その中間図で実際についている線を，**副待機線**とよんで

　　$V[I]$

で表わす．V: Vice.

そういう線は，実際にある．ただし，とくべつな場合として，起線$B[I]$をふくめていう．

§ 12. その ALGŌL 表示

以上にのべたところを，ALGŌL 風に記述してみよう．まず，区間席数の初設定だけれども，SSをふやし

```
         S:=0;
  K 3:   SS:=0; go to L4;
  L 3:   SS:=SS+1;
  L 4:   M:=M+1;
         if Q[M]=0 then go to L3;
  L 5:   J:=J+1;
         if J=I then go to L6;        同一
         if D[J]=D then go to L5;
         if E[J]>0 then go to L5;
         E[J]:=I;
         S[J]:=SS; S:=S+SS;
         go to K3;
  L 6:   S[I]:=S;
```

これで一見よさそうだが，これだと，最後の区間のSSが，Sに加算されないでしまう．go to L 6でなく，

```
  K 6:   S:=S+SS;
```

へ．しかしただこれでは，ぐるぐる廻りになる．そこで

```
         JJ:=0;
```

を，はじめにおき，go to は J6，そうしてたとえば

```
         S[J]:=SS; go to K6;
  J 6:   JJ:=1;
  K 6:   S:=S+SS;
         if JJ=0 then go to K3;
         S[I]:=S;
```

つぎに，待機線・副待機線だけれども，はじめには，後枝はついていないから，副待機線は，前枝ないし起線で，その起線からはじめる．

```
         V:=V[I]:=B[I];
```

それから，L 5以下で拾いだしているJから，Wを条件

$$S[J]>0$$

によって求める．条件に合わなければ，そのJをV．

```
         if V<0 then go to N6;
         if SS>0 then go to M6;
         V:=J; go to N6;
  M 6:   W[I]:=J; V[I]:=V;
         V:=-1;
  N 6:   if JJ=0 then go to K3;
```

ここで，Vを負にしたりするのは，一度WとVが定まったら，以後にまた直されるのをふせぐため．さきのJJといい，こういう，小細工は，どうにでも作られる．

さて，まえにもどり，Iがその起線に，後枝としてついた場合の，区間席数や待機線・副待機線の変化を，しらべよう．Iは当然に，待機線Wの担当区間につく．

```
         W:=W[J];
         if W=J then go to Q2;
         S[W]:=S[W]-1;
```

その結果として，満席になると，Wは待機線としての資格を失う．そこで，新しいWを探すのだけれど，資格は

$$E[K]=J,\quad S[K]>0$$

そういうのがなければ，J自身をWにする．そこで

```
         if S[W]>0 then go to Q2;
         K:=W;
  M 2:   V:=K; (副待機線の候補として)
  N 2:   K:=K+1;
         if K=J then go to P2;
         if E[K]≠J then go to N2;
         if S[K]=0 then go to M2;
  P 2:   W[J]:=K; V[J]:=V;
         go to J3;
```

ところで，満席でない場合にも，副待機線のほうは，当然に変る．それは，定義からいってIで

```
  Q 2:   V[J]:=I;
  J 3:   .........
```

以上を，まえのプログラムに，つけ加える．

なお，一番のはじめに，前辺に対する

```
         W[0]:=V[1]:=1;
         W[1]:=V[0]:=0;
```

がいる．それから，後辺のところで，右辺Iに対する

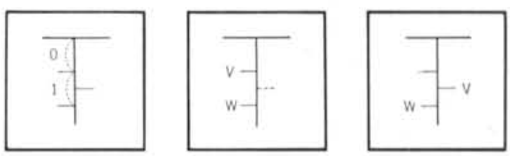
```
         V[I]:=0; (たとえば)
```

がいる．これは$=1$でもよい．一般にいって

$$V[I]=V<I$$

となっていさえすればよい．

元来，WとVは，後枝のとりつく位置，正しくは順序関係を，示すためのもので，後枝数0のIについては，一般には，必要でない．しかし，つぎの §13. でのべる算法では，$I\leq V$のようなVを，Iの後枝と判定する．その誤判を防ぐため，後枝数0でも，Vに適当な値を与えるよう，前ページの箇所でも，注意を払ってある．



§ 13. 前 後 関 係

さて，いま定めた

待機線W，副待機線V

の概念を使って，定義17にいう，線の前後関係

X/Y

を定めてゆく，算法を考えよう．目標は，階数の決定にあるのだから，この関係は，隣接するXとYに限ってよい．また，この関係を定めてゆく立場からすると，

XとYがともに前枝ないし起・終線

の場合も，除いてよいことになる．それを見るために，残りの場合を，分類列挙してみよう．

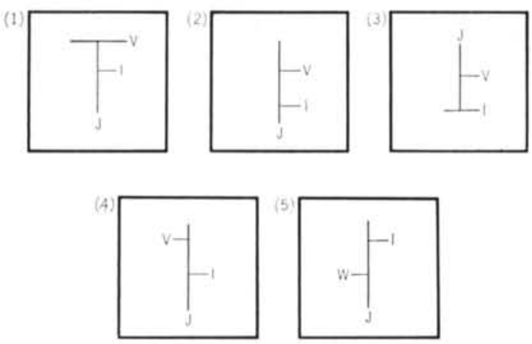
(1) 起線／後枝

(2) 後枝／後枝

(3) 後枝／終線

(4) 前枝／後枝

(5) 後枝／前枝



図で，線の名前は，つぎの規則で定めてある．

J　関係の基礎となる線

I　番号最大の線

V　それと平行で，V/I

W　　〃　　I/W

そうすると，このIを復原する過程での

$V[J]=V$

$W[J]=W$

となっていて，しかもJとIとは

$B[I]=J$　　(1)，(2)，(4)，(5)

$E[J]=I$　　(3)

の関係にある．さらにいえば，

(5)では，Wが満席の場合，また$W\neq J$

(3)では，Vは後枝

まとめていうと

$B[I]=J$　が定まったとき

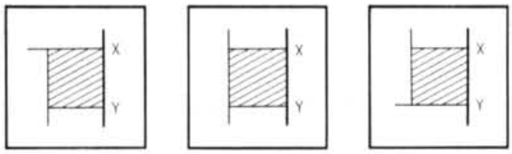
$V[J]=V$　に対し　　　　　　V/I

$W[J]=W$　がそれで満席となれば　I/W

$E[J]=I$　を定めたとき

$V[I]=V$　が後枝ならば　　　　V/I

この定め方で充分なこと，これが，まえに除外した場合をふくむことは，XとYのはさむ長方形の「第4の辺」上で考えれば，よくわかる．



それでは，いまの規則を，ALGŌL 風に表わそう．ただし，関係の記号／は使えないから，その代用として

定義46　上記のいみで，I/J　のことを

$\bar{O}[I,\ J]=1$

その他の場合は

$\bar{O}[I,\ J]=0$

そうするとこれは，まず，すべての

$\bar{O}[X,\ Y]=0$

とした上で，進めるべきだから，

for　$X:=0$　step　1　until　∞　do

for　$Y:=0$　step　1　until　∞　do

$\bar{O}[X,\ Y]:=0;$

前辺に対しては，それで自然に

$\bar{O}[X,\ 0]=\bar{O}[X,\ 1]=0$

それの以外のIに対しては，

$B[I]=J$　がきまったとき

$\bar{O}[V[J],\ I]:=1;$

……………………………

if　$S[W]>0$　then　…　のつぎに

$\bar{O}[I,\ W]:=1;$

$E[J]=I$　としたときに

$V:=V[J];$

if　$V>J$　then　$\bar{O}[V,\ I]:=1;$

以上の定め方で，おなじXとYの組合わせについて

$\bar{O}[X,\ Y]:=1;$

が，2度でることのないことも，注意しておこう．

§ 14. プログラム第1部

これで，ようやく一段落がついた．だんだんにのべてきた，ALGŌL 風プログラムの断片を，ここに一括しておこう．型の宣言などがないという点で，完全ではない．また例の begin……end だの，procedure だのを使わず，label にたよっていて，おさないけれど，前号にのせた，機械語によるプログラムへの，手引きのつもりでいる．対象を正則な図に限り，しかも第1部でしかないけれど，最も本質的なものは，ほぼふくまれている．

```
      for  X: =0  step 1  until  ∞  do
      for  Y: =0  step 1  until  ∞  do
      Ō[X, Y]: =0;
      S[0]: =S[1]: =∞;
      D[0]: =B[1]: =E[0]: =E[1]: =
      W[1]: =V[0]: =0;
      D[1]: =B[0]: =
      W[0]: =V[1]: 1;
      I: =2;  M: =1;

L 0:  if  T[M]=0  then go to  L7;
      D: =D[I]: =Q[M];
      J: =I-1;  go to  L2;
L 1:  J: =J-1;
L 2:  if  D[J]=D  then go to  L1;
      if  E[J]>0  then go to  L1;
      S: =S[J]-1;
      if  S<0  then go to  L1;
      S[J]: =S;  B[I]: =J;
      Ō[V[J], I]: =1;
      W: =W[J];
      if  W=J  then go to  Q2;
      S[W]: =S[W]-1;
      if  S[W]>0  then go to  Q2;
      Ō[I, W]: =1;
      K: =W;
M 2:  V: =K;
N 2:  K: =K+1;
      if  K=J  then go to  P2;
      if  E[K]≠J  then go to  N2;
      if  S[K]=0  then go to  M2;
      W[J]: =K;  V[J]: =V;
      go to  J3;
Q 2:  V[J]: =I;

J 3:  V: =V[I]: =B[I];
      S: =0;
K 3:  SS: =0;  go to  L4;
L 3:  SS: =SS+1;
L 4:  M: =M+1;
      if  Q[M]=0  then go to  L3;
L 5:  J: =J+1;
      if  J=I  then go to  J6;
      if  D[J]=D  then go to  L5;
      if  E[J]>0  then go to  L5;
      E[J]: =I;
      V: =V[J];
      if  V>J  then  Ō[V, I]: =1;
      S[J]: =SS;  go to  K6;
J 6:  JJ: =1;
K 6:  S: =S+SS;
      if  V<0  then go to  N6;
      if  SS>0  then go to  M6;
      V: =J;  go to  N6;
M 6:  W[I]: =J;  V[I]: =V;
      V: =-1;
N 6:  if  JJ=0  then go to  K3;
      S[I]: =S;
      I: =I+1;  M: =M+1;
      go to  L0;

L 7:  G: =0;
      D[I]: =1;  B[I]: =V[I]: =0;
L 8:  J: =0;  go to  L10;
L 9:  J: =J+1;
      if  J=I  then go to  L11;
L10:  if  D[J]=D  then go to  L9;
      if  E[J]>0  then go to  L9;
      E[J]: =I;  go to  L9;
L11:  if  G<0  then go to  L12;
      G: =G-1;  I: =I+1;
      D[I]: =0;  B[I]: =1;
      E[I]: =I-1;  go to  L8;
L12:  S[0]: =S[1]: =0;
```

なお，つぎにのせる「機種の解説」は，先廻りして春に書いたもの．この号でもまだ先廻りだけれど，上記と前号のとの，対照の手助けともなろうかと思った．

3. 機種の解説

§ 15. 記憶の内容と容量

前号でのべたプログラムは，アメリカの

Royal Mc Bee Corporation 社，
Royal Precision Electronic Computer
"LGP-30"

に対している．当社設計部計算課に，設置の機種．小型に属しよう．インデックス・レジスターもない．命令は16種だけ．実用的には，きゅうくつだろうけれど，簡明なところは好ましい．これに対するプログラムは，他の機種に対するものに，たやすく翻訳されるだろう．

まず，記憶装置は，**磁気ドラム**で，**1語の長さ**は

$2^5=32$　ビット

に基づいている．ただし

最後のビット＝スペーサー（空白）ビット

で，これは使われない．また

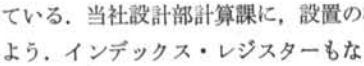
最初のビット＝符号ビット

$$=\begin{cases} 0 & \text{が　非負} \\ 1 & \text{が　負} \end{cases}$$

けっきょく中間の

30　ビット

が，（正数の）数値表示に使われる．



先頭の符号ビット　を　第0ビット
以下のビット　を順に　第1～30ビット

とよぶ．乗除では，

小数点は，第0と第1の間にある

ようにふるまう．つまり，表示する非負数xは

$0 \leq x < 1$

負数xは，この**補数**として表示され，範囲は

$-1 \leq x < 0$

たとえば，正数

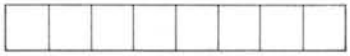
.000001（あとは0）

の符号を変えたものは

1111111（あとは0）

となる．先頭の1が符号ビット．

ところで，2進表示そのままでは，出し入れに不便なので，この機種では，**16進法**を使用する．これははじめにのべた，32ビットの，4桁切りに相当する．



32ビット＝4ビット×8

その4ビットを単位の，$2^4=16$箇の数字として

0，1，…，9，F，G，J，K，Q，W

を使用する．9までは，ふつう通りで，そのあと

10　を　F
11　を　G
12　を　J
13　を　K
14　を　Q
15　を　W

こうしたローマ字のあてはめ方については，後述する．

16進の8桁

＊＊＊＊＊＊＊＊

その8桁目の最終ビットは「空白」の0だから，8桁目にはつねに偶数がおかれる．たとえば

$7WWWWWWQ$，$Q=14$で偶数

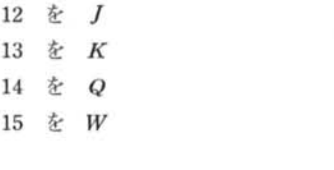
これは，2進表示にすると，＋の

.1111111111111111111111111111111(0)

またたとえば

$7WWWJ082$，2は偶数

＝.1111111111111111110000001000001(0)



さて，記憶装置の**容量**は

4096＝64×64　語

で，ドラム1周上に64語，その64列を軸方向に並べる．

周上の位置をいうのに　**セクター**
列の順番をいうのに　**トラック**

の語を使う．それらの番号としては，10進で

00～63

をあてる．そして

トラック番号，セクター番号

の対を，つづけ書きして，記憶**番地**をいう．たとえば

0963＝09トラックの63セクター

番地は実は全部が1本につながっていて，上記のつぎが

0963＋1＝1000

つまり64番セクター＝つぎのトラックの0番セクターとして，つながれる．

§ 16. 命令の形式と種類

プログラムは，もちろん，内蔵される．個々の命令は

機能部＋番地部

の形の，1番地方式で，1語があてがわれる．2進数として記憶されるわけだけれども，プログラミングは10進でする．まず，機能部は16種類あって，ローマ字

Z，B，Y，R，I，D，N，M，

P，E，U，T，H，C，A，S．

であらわされる．これらは，この順で

0～15，2進4桁数　0000～1111

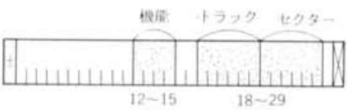
として，第12～15ビットにおかれる．つぎに番地部は

トラック００～６３　2進6桁の000000～111111

が，第18～23ビットにおかれ，つづく第24～5に

セクター００～６３　2進6桁の000000～111111

がおかれる．まとめて示せば



たとえば，記憶装置内におかれた，正の2進数

.000000000001101001111111111110

は，命令として

機能部C，番地部６３６３

こういうのを，つぎのようにかく．

C６３６３

トラックやセクターが10進1桁以下でも0をおぎなう．

C０１００

こうした形式の命令の，効果を，つぎに説明しよう．

A，S，D，M，N，E，B，C，

H，Y，R，U，T，Z，P，I．

の順でのべる．効果の記述のため

積算器（アキューミュレータ）の内容を，*acc*.

命令の番地部を *Add*, *Add* 番地の内容を，*add*.

その命令のおかれている番地を *Loc*.

であらわそう．16箇の前半8箇は，*acc* を変えるが，

$acc = a$

だったものを，たとえば

$a + add = acc$

に変えることを，ALGŌL の流儀で

$acc := acc + add$

と示す．右辺を計算した結果を左辺とする，標語的に

$acc + add \rightarrow acc$

(1) 演算命令

4則，つまり加減乗除，を基本とする．

A　(Add)加法．効果は，　　$acc := acc + add$

S　(Subtract) 減法．　　$acc := acc - add$

D　(Divide) 除法．　　$acc := acc \div add$

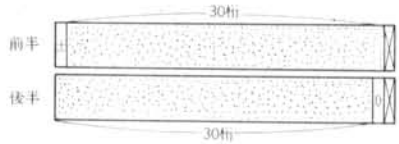
以上の3命令では，結果 x が

$-1 \leq x < +1$

の範囲からはみだすと，計算機は停止する．いわゆる

オーバー・フロー

乗法では，一般には絶対値が小さくなり，オーバー・フローは起らない．2進30桁どうしの積，60桁をわけて



のようにおいたとして

M　(Multiply) 乗法．　　$acc :=$ この前半

N　(はMのつぎ)　　$acc :=$ この後半

Mでは，第0ビットが符号を示すけれども，Nの場合に第0ビットは，符号でなく，積の第31位を示す．

特殊な，しかし重要な使い方として，乗数

$add = 2^{-q}$　（第 q ビットのみ1）

の場合，*acc* 内容は桁ずらし，**シフト**される．そうして

M　の効果は，右シフト q

N　の効果は，左シフト $(31 - q)$

演算命令として，もうひとつ

E　(Extract) 抽出．桁ごとの乗法ともいわれる．

$acc(q) := acc(q) \times add(q)$

ただし $acc(q) = acc$ の第 q ビット

$add(q) = add$ の第 q ビット

(2) 置数と格納など

演算にさきだち，積算器に数をまず置くこと，演算のあとで，結果を一時記憶に格納する，などのため，まず

B　(Bring) 置数．　$acc := add$

もとの *acc* は，払われるわけだけれども，いまかりに，払うだけの命令，C（番地部欠）があったとすると，

B＊＊＊＊≡C，A＊＊＊＊

（御破算で願いましては，……なり）．実際には

C　(Clear)　格納し破算．$add := acc$, $acc := 0$

がある．格納命令としては，もうひとつ

H　(Hold)　格納し保存．　$add := acc$, *acc* 不変．

もっとも，一方だけで間に合うことは合う．

H＊＊＊＊≡C＊＊＊＊，A＊＊＊＊

C＊＊＊＊≡H＊＊＊＊，S＊＊＊＊

格納命令の変種として，なお2種がある．これらは，飛躍命令と組合わせられて，効果的な働きをする．

Y （原語では Store Address）番地格納．

add の第18〜29ビット，*add*（18〜29）：＝

acc の第18〜29ビット，*acc*（18〜29）

ほかの部分は不変，*acc* も不変．

これで，命令の番地部だけの変更ができる．とくにサブルーチンの出口の，飛躍命令の行先番地設定のため

R （Return）帰還．*Loc*＋2をYする，つまり

add（18〜29）：＝*Loc*＋2．なお *acc* 不変．

たとえば，つぎにのべる飛躍命令Uと組合わせて

Loc　Rサブ出口

＋1　Uサブ入口　―飛躍して→　サブ入口

＋2　……………　←飛躍して―　サブ出口U

(3) 飛躍

プログラムを構成する命令は，ふつう格納位置の順に実行される．しかし

U （Unconditional Transfer）無条件飛躍．

Add にある命令に飛躍．

それから，

T （Test） 負飛躍．つまり

acc＜0　なら，Uと同等，*Add* へ．

acc≧0　なら，無効命令，*Loc*＋1へ．

なお，特殊な使い方として，このT命令の第0ビットに1を入れておくと，計算機の

TC（Transfer Control）ボタン

の状態に関係して動作する．

押してないと，ふつうどおり．

押してあると，非負でも飛躍，Uと同等．

(4) 停止

Z （＜Zero，Stop）停止．

番地部は必要でないが，一般には，0000をおく．

つぎにのべるプログラムでは使用しないが，番地部の

トラック≠00，01，02，03

とすると，条件つきの停止となる．計算機には，4箇の

BP（Break Point）ボタン

32，16，08，04

がついている．そうして，たとえば，Z命令の

トラック＝20＝16＋04

のとき，このBPボタン

16　および　04

の双方が押されていると無効化，その他では停止する．

(5) 入出力

タイプライタと連絡する命令が2種あり，まず

P （Print）印刷．番地部のトラックの数値により．

数字・記号・文字の印刷その他の動作をさせる．

上下	数値	上下	数値	上下	数値
）0	02	Aa	57	Ss	61
L 1	06	Bb	05	Tt	45
＊2	10	Cc	53	Uu	41
''3	14	Dd	21	Vv	31
△4	18	Ee	37	Ww	62
%5	22	Ff	42	Xx	39
$6	26	Gg	46	Yy	09
π7	30	Hh	49	Zz	01
Σ8	34	Ii	17	動作	数値
（9	38	Jj	50	*LC*	04
_ ―	07	Kk	54	*UC*	08
＝＋	11	Mm	29	*CS*	12
：；	15	Nn	25	*CR*	16
？/	19	Oo	35	*SP*	03
］．	23	Pp	33	*BS*	20
［，	27	Qq	58	*TB*	24
''	32	Rr	13	*Del.*	63

ローマ字*L*の小文字は，数字の1と共用．動作の部，

LC　Lower Case

UC　Upper Case

CS　Color Shift

CR　Carriage Return Line Feed

SP　Space

BS　Back Space

TB　Tab （人手でセットしておく）

Del.　Delete で削除，印刷からのぞかれる．

最後のは，計算機からでなくテープからの印刷のため．もともと，タイプでテープ・パンチのとき，たとえば

数字1　→　06＝2進の000110　対応の穴列．

（逆に，その穴列からその字を印刷）．ミスしたとき，*Del.* で63＝2進の111111，全部穴にして，削除．

印刷は時間をとるので，PのあとZで**一時停止**させ，印刷後タイプからのスタート指令で計算を再開させる．

なお，最初期入力，ブート・ストラップなどでは

I （Input）入力．番地部は0000

が，つぎの対の形で使われる．

P0000，I0000

プログラム解説には関係がないから，立ち入らない．