BELL TELEPHONE LABORATORIES
INCORPORATED

SUBJECT: A Movie Language for Phasors and Signals

DATE: May 19, 1965

FROM: W. H. Huggins

MEMORANDUM FOR FILE

1. Introduction

To simplify the making of animated movies using rotating phasor diagrams to illustrate sinusoidal signals, sidebands, and modulation phenomena, a set of macros have been written for the ROLL-mode microfilm subroutines. These macros may also be used in conjunction with the modified BEFLIX program, thus providing the features of both the vector-drawing and character-typing modes of output needed for making titles, etc.

The macros described in this memo provide simple instructions for defining data for plotting points, drawing lines and arrows, and for generating waveforms of oscillatory signals having arbitrary frequency and rates of growth or decay.

Before discussing the various features of these instructions, it will be helpful to describe briefly the arrangement of data tables containing the coordinates of points and lines.

The coordinates of points and lines and other parameters are entered as real numbers (i.e., with decimal points) and are stored in floating-point form. (The conversion of the floating-point form to integers is performed automatically by the graphical output routines.)

A data table containing the coordinates of 4 points may be visualized most easily as the array of number pairs shown in Fig. 1. The coordinates of the first point are the

A =

X1	Y1
X2	Y2
X3	Y3
X4	Y4

Fig. 1. A Data Array

number-pair (X1,Y1), the coordinates of the second point (X2,Y2), etc.

In defining this table, the symbol A has been identified as the name of the first location in the table, the content of which is X1. Then A+1 denotes the next location in the table, the content of which is Y1. Introducing the notation C(LOC) to denote the content of a location LOC, we see that

C(A) = X1
C(A+1) = Y1
C(A+2) = X2
C(A+3) = Y2
C(A+4) = X3
etc.

Although it may be helpful to visualize the X and Y coordinates as separate columns, one must recognize that the data are stored in the computer in a single column with X and Y interleaved as illustrated here. (That is, double subscripts, as used in FORTRAN, are not used in these programs.)

Data tables are used by the graphical output routines. For instance, the statement

```
PLTPTS   3,A,CIRCLE
```

will cause three small circular points to be plotted at the coordinates (X1,Y1), (X2,Y2) and (X3,Y3). (The symbol CIRCLE is defined by the octal digits 460000000000.) Similarly the instruction

```
PLTPTS   2,A+4,CIRCLE
```

will plot two circular points at the coordinates (X3,Y3) and (X4,Y4). Observe that the instruction

```
PLTPTS   1,A+1,CIRCLE
```

will plot a single circle with interchanged coordinates (Y1,X2).

Data tables are used to describe lines, such as the axes of a graph or phasor arrows. A line is described by specifying the coordinates of its two end points. Thus a single line is specified by an array of 4 consecutive numbers.

A =

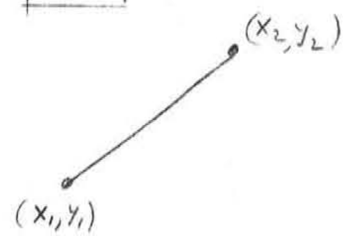
X1	Y1
X2	Y2
X3	Y3
X4	Y4

$\Longrightarrow$ A =

X1
Y1
X2
Y2
etc.

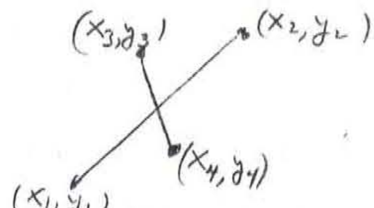
The instruction

DRWLNS   1,A



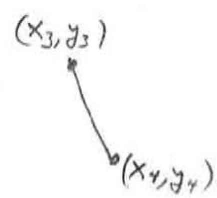
would draw a single line between the end points described by Table A.

DRWLNS   2,A



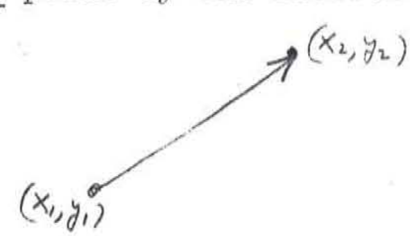
would draw two lines, the first as shown above and the second between the points (X3,Y3) and (X4,Y4) specified in Data Table A. This second line only may be drawn by the instruction

DRWLNS   1,A+4



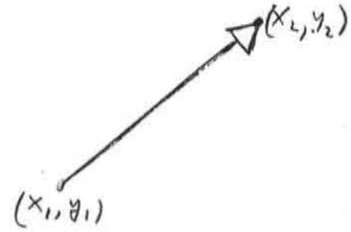
The order of the pair of points describing a line is important if the lines are directed. Then, an arrow head may be automatically drawn at the <u>second</u> point by the instruction

DRWARO   1,A,HD,OPEN



or

DRWARO 1,A,HD,CLOSED.



The symbol HD is either the symbolic address of the numerical value of the length of the arrow head when projected onto the line (using 1024.0 as the maximum size of the SC4020 plotting surface) or the (floating-point) numerical value itself. The arrow head may be either OPEN or CLOSED, as illustrated above.

2. Location-Defining Macros

Definition of symbols and arrays may be accomplished by two macros which name and allocate blocks of storage located immediately after the end of the main program and prior to the literal table. These two macros use the remote (RMT) feature of the system, thus removing them from the main-program sequence. This useful feature allows the user to define his symbols where they are needed rather than in a consolidated list at the beginning of the program. If a symbol has already been defined, the largest dimension will be used. (A conditional test is incorporated to prevent multiple definitions.) The maximum size of the array is determined by the largest dimension specified.

All of the data-setting (SETXXX) macros described in the next section will also automatically define an output array if it has not already been defined. Furthermore, the size of this array will automatically become the largest specified size on any of the calls. Thus, the array dimensioning is automatic.

For instance, the statements

```
A   DIM   4,2      or      A   DIM   8
```

will reserve a block of (4)(2) = 8 storage locations at the end of the program, the first location being denoted by the symbol A and the last location being denoted by A+7. If, however, the symbol A appears as the output array of another instruction, say

```
SETVAL   26,B,A
```

then the dimension of A would automatically be increased from 8 to 26.

The statement

```
DEF   (B,NT,ARRAY)
```

will reserve three consecutive locations with the first denoted symbolically by B, the second by NT, and the last by ARRAY. (Observe that B+1 specifies the same location as NT or ARRAY-1.)

It should be noted that the two integers M and N in the statement "A DIM M,N" are provided merely as a convenience to the user so that he can think of rectangular matrix-like arrays. However, double-indices are not used in these macros and all arrays must be indexed as a single column of storage locations.

In general, all symbols appearing in the following statements denote symbolic addresses whose contents are the specified quantity, with the exception of the single symbols M, N, L, the number-of-frames argument NF used in SETVEL and MØVEXY, and the NTIMES argument in DØ NTIMES(SUBLIST), which denote integers (rather than addresses of integers). As a rule, N will denote the number of data items, M the number of points (or XY pairs), and L will denote the number of lines (or pairs of points).

3. Data-Setting Macros

The next four data-defining macros also use the remote (RMT) system feature and may therefore be incorporated at any point in the same program without disrupting it at run-time. However, the main program must terminate in a transfer (usually to ENDJØB) to protect these data areas.

Definition of arrays of decimal data may be accomplished by the statement

```
LØC    SETNUM    (N1,N2,...)
```

where LØC is the symbolic address of the first location in the array, and the list N1,N2,... contains either successive decimal integers (no decimal point); decimal numbers (with decimal point); or defined symbolic locations of such numbers, or any mixture thereof. For instance,

```
AB    SETNUM    (36.15,42,AB)
```

will establish three storage locations whose contents are

```
  C(AB) = 36.15
C(AB+1) = 42
C(AB+2) = 36.15
```

Arrays of octal numbers may be defined by the statement

```
LØC    ØDATA    (N1,N2,...)
```

where the octal integers will be right justified in each 36-bit word. Thus,

```
MASK    ØDATA    (520000000000,42)
```

will establish two storage locations whose octal contents are

```
C(MASK)   = 520000000000
C(MASK+1) = 000000000042
```

Arrays of Hollerith data may be defined by the statement

```
LØC    HDATA    N(W1,...,WN)
```

where the N, six-character words in the list will be stored in binary-coded form in successive locations beginning at LØC. For instance,

ASTRIX HDATA 1(*)

will establish a single location whose contents is the alphanumeric octal code for the asterix symbol followed by the code for 5 blanks, i.e., 546060606060.

To place an integer or its symbolic equivalent into a location denoted by the symbol LØC, one may write the statement

LØC SETEQU INTGER

Several instructions are available for transferring data from one array to another. Thus, data may be transferred from one array to another by

SETVAL N,A,C,

which sets CK = AK for K = 0,1,...,N-1 To set the first N elements of any array A to zero, use the statement

SETVAL N,O,A

Thus the instruction

SETVAL 8,O,A

will fill the Table A previously defined with zeros.

To transfer the first M, X coordinates from array A to array C, one may use the instruction

```
SETXS   M,A,C,
```

For instance,

```
SETXS   4,A,A+2
```

will set X2,X3, and X4 all equal to X1, without affecting the Y values. Similarly, to transfer the first M, Y coordinates from array A to array C, use the instruction

```
SETYS   M,A,C
```

This information leaves the X values unchanged. Here also, if the symbol A is a zero, zero values will be transferred to the output array.

To translate M points given in the table XY by some offset (XØ,YØ) and place the new coordinates into another table XYØ, one may use

```
SETØFF   M,XY,XYØ,ØFFSET
```

Here, the offset argument is optional. If it is omitted (viz., SETØFF   M,XY,XYØ), this instruction is equivalent to (SETVAL   N,XY,XYØ) with N = 2M, except that transfer begins with the last pair of points and proceeds to the first pair. Thus,

```
SETVAL   6,XY,XY+2
```

will set all 4 points in the table to the same coordinates as the first point, whereas

```
SETØFF   3,XY+2,XY
```

will set all 4 points equal to the coordinates of the last point.

Two macros are provided for floating and unfloating data arrays. Thus,

```
SETINT   N,A,K
```

will unfloat the first N items of A and place the resulting integers in K, whereas

```
SETFLT   N,K,A
```

transfers and floats the first N items of K into A.

The sums and differences of two arrays may be obtained by the instructions

```
SETSUM   N,A,B,C
```

which sets the first N elements of the C array equal to the sum of the corresponding elements in the A and B arrays. Similarly,

```
SETDIF   N,A,B,C
```

sets CK = AK - BK for K = 0,1,...,N - 1.

In the instructions SETSUM, SETDIF, SETVAL, SETXS, SETYS and SETINT, the transfer begins at the first entry of the tables and proceeds toward the last entry. This fact is important when transferring data within the same table, as illustrated by the SETXS example given above. There, the single coordinate X1 is propagated through all X locations. In contrast,

```
SETXS   3,A+2,A
```

will replace X1 with X2, X2 with X3, and X3 with X4.

A time base may be generated by the statement

```
SETIME   NT,UNIT,TIME
```

which computes

TIME from (NT)(UNIT).

The 2M parameters which control the rate of rotation and rate of growth or decay of each of the M phasors produced by SIGGEN are set into an array PAR by the statement

```
SETPAR   M,ØPTIØN,DATA,NF,PAR
```

The data specified in the array, DATA = (A0,B0,A1,B1,A2,B2,...,AM,BM) are interpreted in accord with two options:

(1) For option = AMPDEG, A(K) is the amplitude-growth factor per NF frames and B(K) is the frequency of the Kth signal expressed in degrees per NF frames of the movie.

(2) For option = DT, A(K) is the real part of the Kth pole and B(K) is the imaginary part (in nepers and radians, respectively). DT is the symbolic address of the time interval between successive frames. (Any symbol may be used for DT except AMPDEG.)

To generate a set of harmonically related sinusoidal signals, as required by a Fourier series, the parameter array PAR may be set by the single instruction

```
SETFRQ   M,FUND,PAR.
```

Using this PAR in SIGGEN will produce a constant signal (of zero frequency) and the first M-1 harmonics of FUND, where FUND is the fundamental frequency (in degrees per frame).

A line is described by two data-pairs which contain the coordinates of the ends of the line. The data array for a collection of L lines is assembled from arrays XY1 and XY2, containing respectively the coordinates on the first and second end-points, by the instruction,

```
SETLNS   L,XY1,XY2,LNA
```

To define a collection of lines with their first ends at the origin, the symbol XY1 is replaced by "0". For instance, since each signal is described by a point, whereas a phasor is described by a <u>line</u>, twice as many numbers are needed to specify a phasor data table.

$$XY = \begin{array}{|c|c|} \hline X1 & Y1 \\ \hline X2 & Y2 \\ \hline \end{array} \quad \Rightarrow \quad PHA = \begin{array}{|c|c|} \hline 0 & 0 \\ \hline X1 & Y1 \\ \hline 0 & 0 \\ \hline X2 & Y2 \\ \hline \end{array}$$

Transfer of the data for M points from XY to PHA at the locations indicated in the preceding diagram is accomplished by the instruction

SETLNS 2,0,XY,PHA

Similarly, the symbol XY2 may be replaced by "0" if the second end of each line is to be at the origin.

A table of M, XY pairs may be thought of as M complex numbers. It is possible to multiply these numbers by a single complex scalar A = (A1,A2) and place the results in another table AXY by the single instruction

SETSCA M,A,XY,AXY

By setting A2 = 0, this operation may be used to scale all data in the table XY by the same real constant, A1. Similarly, by letting A = (cos θ, sin θ), the points in the array XY will be rotated counterclockwise about the origin through an angle θ.

To move a set of points from XY1 to XY2 in NF frames, the points must be moved by amounts per frame given by the table XYVEL which may be computed by the simple instruction

SETVEL M,XY1,XY2,NF,XYVEL

Here, NF is either an integer or a symbol equivalent to an integer.

4. Utility Macros

Two macros useful for saving and restoring any of the seven index registers of the IBM-7094 are

```
XRSAVE LØC (ARG$)
```

which saves the contents of the index registers listed as ARGS in successive locations beginning at LØC, and

```
LOC    XRREST  (ARGS)
```

which restores the index registers listed as ARGS from the values stored in successive locations beginning at LØC. Thus to save registers 1 and 4 at the beginning of a subroutine, one may write

```
       XRSAVE SUBXIT(1,4)
```

and at the end of the subroutine

```
SUBXIT XRREST(1,4)
       TRA    (return address)
```

To provide N blank frames of film for leader, etc., simply write

```
       LEADER N
```

where N is a positive integer.

An array RA of point data given in polar form, (R1,A1,R2,A2,...,RM,AM), where RK is the magnitude and AK is the angle (in degrees) of the Kth point, may be converted to rectangular form by the instruction

```
RA2XY   M,RA,XY
```

Similarly, conversion of XY data to polar form, with the angle between 0 and 360 degrees, may be accomplished by the instruction

```
XY2RA   M,XY,RA
```

Finally, the instruction

```
RA2LØG   M,RA,LØGRA
```

places the natural logarithms of the complex data-pairs RA into the array LØGRA.

A list of subroutines (each terminating in TRA 1,4) may be executed N times by the instruction

```
DØ   NTIMES,(SUBLST)
```

Graphical output instructions may use the RØLL mode subroutines DVR, DVR1, DVR2, DVR3, TSP, and TSP1. However, these subroutines require different data structures than those just described. Furthermore, the data must be integers. Therefore the following macros are provided to facilitate use of the data arrays described above.

DRWLNS   L,XYENDS,ØFFSET

This statement will draw L lines whose ends are specified in the data array XYENDS, with offset specified by a two-element array (XØ,YØ) having the symbolic name ØFFSET. If the ØFFSET is zero, the last argument is omitted (DRWLNS L,XYENDS). Four data elements must be specified for each line to be drawn.

As mentioned in the Introduction, points are specified by their (X,Y) coordinates and may be plotted by the instruction

PLTPTS   M,XYPTS,CHAR,ØFFSET

If no offset is desired, the last argument is omitted.

A clock with face marked at 5-minute intervals and minute and hour hands is generated by the statement

CLØCK   RADIUS,ØFFSET

where RADIUS is the symbolic address of the clock radius and ØFFSET contains the XY coordinates of the clock center. The clock is set to have a minute-hand PERIØD (in frames) by the statement

SETCLK   PERIØD

The clock is plotted by including the subroutine CLØCK in an appropriate (SUBLIST). It is made to run (i.e., to advance with each call to CLØCK) by the statement

RUNCLK

It may be stopped by the statement

STPCLK.

The instructions (or their SUBLIST equivalents),

```
TSX   ZRØCLK,4
TSX   ADVCLK,4
TSX   SHØCLK,4
```

may also be used to set the clock to zero, to advance the clock by one-frame, and to plot the clock.

Dynamic movements may be accomplished by the statement

```
MØVEXY   M,XY1,XY2,NFRMS,XY(SUBLST)
```

which moves M points XY from the initial positions specified in XY1 to the final positions specified in XY2. The motion is linear and occurs in NFRAMES. At each frame the list of subroutines (SUBLST) is performed.

5. Signal-Processing Macros

The single statement

```
SIGGEN   M,PAR,AMP,XYØLD,XY,NGT,(SUBGT),NHT,(SUBHT)
```

creates a generator for M complex signal phasors, whose frequencies and growth rates are determined by the 2M parameters in PAR. The initial values of each phasor are specified by the 2M elements in AMP. The 2M elements comprising the array XY contain the (X,Y) coordinates of the M phasors at the present frame. The 2M elements of XYØLD contain the (X,Y) coordinates at the previous frame. (Thus, continuous traces may be defined by the instruction SETLNS M,XYØLD,XY, LINES.) The integer NGT is the present sample number. NHT is the historical sample number. This macro is usually included within a subroutine or loop so that SIGGEN is used repeatedly to generate the data for successive frames of the movie.

The present integer time of the generator is NGT. Each call to SIGGEN increments NGT by 1 and performs the list of subroutines (SUBGT), using the signal values at NGT. The historical time NHT is incremented from zero to NGT to obtain each new value of NGT while performing the list of subroutines (SUBHT) for each NHT value between 0 and NGT. In this way the historical trace is recomputed for each frame of the movie, thus minimizing storage requirements (at the expense of repeated computation).

Usually, the last subroutine in (SUBGT) will contain a (TSX RØLL,4) instruction to advance the film of the microfilm plotter. RØLL may simply be given as the last subroutine in the list, viz. (SUB1,SUB2,RØLL).

On each execution of SIGGEN, the content of NGT is increased by 1. The historical subroutines (SUBHT) are performed only if NHT $<$ NGT. Hence, by setting NGT to a negative integer value initially, the signal generator will remain inoperative until a sufficient number of calls to SIGGEN have been made to increase NGT to a positive integer value. This feature may be used to suspend the phasor animation for a brief initial period while executing other changes, such as slow movement of the graph axes, etc., under control of the CHANGE STATE and STATE ATTIME subroutines.

The phasors stored in array XY may be added vectorially and their partial sums placed into an array SXY by the instruction

```
SIGSUM   M,XY,SXY
```

then

```
C(SXY)   = X1
C(SXY+2) = X1 + X2
C(SXY+4) = X1 + X2 + X3
C(SXY+5) = Y1 + Y2 + Y3
         etc.
```

To change the tuning of the signal generator and any other parameters at specified instants during the run, it is possible to specify a list of subroutines which will be executed at specified instants by the statement,

```
STATE   ATTIME   ((NT1,LØC1)(NT2,LØC2)...).
```

Then subroutine LØC1 will be executed when NT = NT1, subroutine LØC2 will be executed when NT = NT2, etc.

To call this subroutine, one writes the single statement

```
CHANGE STATE,NT
```

where NT is the address of the integer time measure to be compared with the ordered integer values NT1, NT2, etc., specified in the "STATE ATTIME" subroutine.

6. Availability

This macro package is available on semipermanent disk, and may be opened by control card:

```
WHHMVE MYDISK  PMACRØ
```

The macro package is liberally annotated and the interested user may obtain a listing by using the program

```
       JOB CARD
WHHMVE MYDISK  PMACRØ
       FAP
XXX    CRUNCH PMACRØ
(Insert here users program)
       END
```

Instructions for suppressing the listing of the macros are given in the listing. A summary of the macros appears in the attached appendix.

MH-1371-WHH-JG

W. H. HUGGINS

Attached
appendix

Appendix

```
  SYMBØL AND DATA-DEFINING MACRØS --  USE RMT LISTING
  ====================================================
      DEF      (A,B,C,...)           LISTS ØF SINGLE SYMBØLS
LØC   DIM      N  ØR  M,N            ASSIGNS N ØR MXN SPACES AT LØC
LØC   HDATA    N(W1,W2,...)          BCI DATA ENTRY BEGINNING AT LØC
LØC   ØDATA    (N1,N2,...)           ØCTAL DATA ENTRY BEGINNING AT LØC
LØC   SETEQU   INTGER                PUTS EQUIV ØF INTGER AT LØC
LØC   SETNUM   (ARGS)                STØRES ARGS BEGINNING AT LØC WHERE
                                     ARGS ARE LØC SYMBØLS ØR DEC. NUMBERS
      SETVAL   N,A,C                 TRANSFERS FIRST N VALUES FRØM A TØ C
      SETXS    M,A,C                 TRANSFERS FIRST M X VALUES FRM A TØ C
      SETYS    M,A,C                 TRANSFERS FIRST M Y VALUES FRM A TØ C
      SETØFF   M,XY,XYØ,ØFFSET       TRANSFERS M PØINTS ØF XY INTØ XYØ
                                     WITH ØFFSET (ØFFSET ØPTIØNAL).
      SETINT   N,A,K                 UNFLØAT  N  ITEMS ØF  A  INTØ  K
      SETFLT   N,K,A                 FLØATS N ITEMS ØF K INTØ A
      SETSUM   N,A,B,C               A+B INTØ C ARRAY FØR FIRST N ITEMS
      SETDIF   N,A,B,C               A-B INTØ C ARRAY FØR FIRST N ITEMS
      SETIME   NT,UNIT,TIME          CØMPUTE TIME FRØM NT*UNIT
      SETFRQ   M,FUND,FREQH          PUTS M-1 HARMØNICS ØF FUND INTØ FREQH
      SETPAR   M,ØPT,DATA,NF,PAR     CØMPUTES SIGGEN PARAMETERS FRØM DATA
                                     ØPT=AMPDEG FØR NF=NUMBER ØF FRAMES
                                     ØPT=DT     FØR DT=TIME BETWEEN FRAMES
      SETLNS   L,XY1,XY2,LNA         SETS DATA INTØ LNA ARRAY FØR L LINES
      SETSCA   M,A,B,C               PUTS M CØMPLEX PRØDUCTS ØF (A1,A2)
                                     WITH B-PAIRS INTØ C-PAIRS.
      SETVEL   M,XY1,XY2,NF,XYVEL    SETS XYVEL TØ MØVE M PØINTS FRØM
                                     XY1 TØ XY2 IN NF FRAMES.
  UTILITY MACRØS
  ==============
      LEADER   N                     PUT N-FRAME LEADER ØN FILM
      XRSAVE   LØC,(XRLIST)          SAVE XR LIST BEGINNING IN LØC
LØC   XRREST   (XRLIST)              RESTØRE XR LIST FRØM LØC
      XY2RA    M,XY,RA               CØNVERTS M PØINTS FRM RECT TØ PØLAR
      RA2XY    M,RA,XY               CØNVERT M PØINTS FRØM PØLAR TØ RECT.
      RA2LØG   M,RA,LØGRA            CØNVERTS M PØINTS FRØM PØLAR TØ LØG
      DØ       NTIMES,(SUBLST)       REPEATS LIST ØF SUBRØUTINES NTIMES.
      DRWLNS   L,XYENDS,ØFFSET       DRAWS  L  LINES,  (ØFFSET ØPTIØNAL)
      DRWARØ   L,LNSA,LENGTH,TYPE,ØFFSET  DRAWS  L  ARRØWS WITH HEAD
                                     LENGTH (FLTNG-PT) AND TYPE IS (ØPEN
                                     ØR CLØSED).  (ØFFSET ØPTIØNAL).
      PLTPTS   M,XYPTS,CHAR,ØFFSET PLØTS M PØINTS. (ØFFSET ØPTIØNAL).
      CLØCK    RADIUS,ØFFSET         DEFINES SETCLK PERIØD, RUNCLK, STPCLK
                                     ALSØ SUBS,  ZRØCLK, ADVCLK, SHØCLK
      MØVEXY   M,XY1,XY2,NFRMS,XY(SUBLST)  MØVES XY FRM XY1 TØ XY2  IN
                                     NFRMS DØING SUBLIST AT EACH FRAME
  SIGNAL-PRØCESSING MACRØS
  ========================
      SIGGEN   M,PAR,AMP,XYØLD,XY,NGT,(SUBGT),NHT,(SUBHT)
      SIGSUM   M,XY,SXY              SUMS SUCCESIVE XY DATA INTØ SXY
STATE ATTIME   ((NT1,LØC1)(NT2,LØC2)...)  SPECIFIES THAT THE SUBRØUTINE
                                     LØC1 WILL BE IMPLEMENTED WHEN NT=NT1,
                                     ETC. (SEE CHANGE-STATE MACRØ BELØW)
      CHANGE   STATE,NT              CALLS STATE ATTIME SUBRØUTINE
```