

**stichting
mathematisch
centrum**



AFDELING INFORMATICA

IW 14/74

FEBRUARY

A.E. BROUWER & C.L. PIPPEL
PDP8-PROGRAMMING 1
THE KV8/1 STORAGE SCOPE AND H306 JOYSTICK

2e boerhaavestraat 49 amsterdam

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

PDP8-PROGRAMMING 1

The KV8/I storage scope and H306 joystick

by

A.E. Brouwer & C.L. Pippel

Preface

This report is the first in a series entitled "PDP8-PROGRAMMING". This series will contain reports on PDP8 software and programming aspects of PDP8 hardware.

Contents

| | |
|---|----|
| Summary | 1 |
| 0. Introduction | 1 |
| 1. The screen | 1 |
| 2. The flags | 2 |
| 3. The registers | 2 |
| 3.1. XSH and YSH (X resp. Y sample and hold register) | 3 |
| 3.2. XINTH, YINTH (X resp. Y integrate and hold register) | 3 |
| 3.3. D-register | 3 |
| 3.4. AB, PT flip-flops | 3 |
| 3.5. L/S, VC flip-flops | 4 |
| 4. The A/D conversion | 4 |
| 4.1. Instructions for A/D conversion | 4 |
| 4.1.1. SAC | 4 |
| 4.1.2. LDA | 5 |
| 4.1.3. SDA | 6 |
| 4.2. A/D conversion routine | 6 |
| 5. The instructions | 7 |
| 5.1. device number 05 | 7 |
| 5.2. device number 06 | 7 |
| 5.3. device number 07 | 11 |
| 6. Examples | 11 |
| 7. PRFIG | 15 |
| 8. Character generators | 18 |
| 8.1. A two-page character generator | 18 |
| 8.2. A three-page character generator | 24 |
| 8.3. A character generator using dots | 31 |

Summary

This description includes all information about the KV8/I scope and H 306 joystick available to us, but only insofar as relevant to the programmer. It includes the results of several experiments, conducted by us. For a description of the hardware we refer to the appropriate publications of DEC and Tektronix.

0. Introduction

The KV8/I storage scope - interfaced with the PDP8/I computer - is a data storage and display instrument. It has a screen of 21×16.2 cm on which figures composed of points, straight lines and circular arcs can be drawn.

Normally the figure remains visible until it is erased (under program control or by pushing the ERASE button). However, by manually adjusting the intensity to 'write thru', one can inhibit the storage of a figure as it is being displayed. [Something like this can be done also under program control by drawing long vectors with short vector timing.]

The H 306 joystick is a device consisting of an interrupt bar and a stick that controls the motion of a small ellipse, called cursor. The ellipse can be made visible on the screen. The position of the cursor can be determined by the program.

1. The screen

The screen is addressed by two coordinates, X and Y, where $-512_{10} \leq X, Y \leq 511_{10}$ (i.e. X and Y are ten bit two's complement values). However, only points with $|X| \leq \text{ca. } 280_{10}$ and $|Y| \leq \text{ca. } 360_{10}$ correspond to a spot on the screen. The point (0,0) is the center of the screen. [Since the corners of the screen are rounded and the focus along the border is not too well, we advise not to use points with $|X| > 270_{10}$ or $|Y| > 350_{10}$. These numbers may vary from installation to installation, but if you find significant discrepancies, yours is probably badly regulated.]

To draw a picture on the screen one generally performs the following

actions:

- select the appropriate mode
- specify coordinates
- execute function (e.g. draw line, circle, point, etc.).

The precise action of the instructions involved is dependent on the values of several registers and flip-flops in the display-logic. Therefore, before describing the instructions we first consider these registers and flip-flops.

2. The flags

Two flags are connected to the skip and interrupt bus: the ready flag of the scope (RF) and the cursor flag. The cursor flag becomes 1 when the interrupt bar is hit; it is cleared by the program and by pushing START.

Instructions:

- SNC = 6051 skip (the next sequential instruction) when the cursor flag is not up.
- CCF = 6052 clear the cursor flag.

The ready flag becomes 1 after the completion of certain actions by the scope (as will be described later), and is cleared by the program - explicitly or by initiating certain actions - any by pushing START.

Instructions:

- SRF skip (the next sequential instruction) when the RF is up.
- CRF clear the RF.

[The instructions LDX, LDY, EXC, EXL clear the RF and set it again when the scope is ready. Pushing the ERASE button does not clear the RF, but after the erasure of the display the flag is set.]

3. The registers

The logic of the scope contains a.o. four analog registers, a 10-bit digital register and a number of flip-flops.

Analog registers:

3.1. XSH and YSH (X resp. Y sample and hold register)

In point mode they hold the (X,Y) coordinates of the point. In linear vector mode they hold ΔX and ΔY . In circular vector mode they hold the (X,Y) coordinates of the centre of the circle. They are not disturbed in cursor mode, but their drift is specified as being not more than 30 positions/sec.

3.2. XINTH, YINTH (X resp. Y integrate and hold register).

They hold the (X,Y) coordinates in vector mode and are not disturbed in point or cursor mode. Their drift is specified as being not more than 6 positions/sec.

Digital registers and flip-flops:

3.3. D-register

The D-register is a 10-bit register, that is used with each transfer of information from the CPU to the scope-logic. It is loaded from AC2-11 by the instructions (binary): "110 000 11x 1xx", i.e. by LDX, LDY, EXC, EXL and LDA.

3.4. AB, PT flip-flops

The AB and PT flip-flops determine the mode of operation of the scope:

| PT | AB | |
|----|----|---------------------------|
| 0 | 0 | : incremental vector mode |
| 0 | 1 | : absolute vector mode |
| 1 | 0 | : point mode |
| 1 | 1 | : cursor mode |

The PT flip-flop is loaded from AC2 by the instructions (binary): "110 000 110 x1x", i.e. by SAC, LDF, EXC and EXL. The AB flip-flop is loaded from AC3 by the same instructions.

3.5. L/S, VC flip-flops

The L/S and VC flip-flops determine the timing of a vector (short vector 250 μ s, long vector 4.05 ms, or vector continue). The L/S flip-flop is loaded by the binary instructions: "110 000 110 xly" with the value: (\neg AC4) & (y \vee AC6). The VC flip-flop is loaded from AC7 by EXC and EXL. It is cleared by SAC and LDF.

4. The A/D conversion

To convert an analog value into a digital one, we have to compare repeatedly that analog value against a variable digital trial value; a skip-type instruction tells which one was the largest. The analog values that can be read out are: XINTH, YINTH, XSH, YSH, XCUR (the X coordinate of the cursor) and YCUR.

4.1. Instructions for A/D conversion

To compare an analog value with a digital one the following three actions are required:

- select analog source, i.e. specify which analog source value is to be converted,
- specify digital value,
- perform test.

These actions are performed resp. by the instructions SAC, LDA and SDA described below.

4.1.1. SAC

SAC (=6062) selects the analog source to be tested according to bits 2,3 and 6 of the AC.

| AC2 | AC3 | AC6 | source |
|-----|-----|-----|--------|
| 0 | 0 | 0 | XINTH |
| 0 | 0 | 1 | YINTH |
| 0 | 1 | 0 | XINTH |
| 0 | 1 | 1 | YINTH |
| 1 | 0 | 0 | XSH |

| | | | |
|---|---|---|------|
| 1 | 0 | 1 | YSH |
| 1 | 1 | 0 | XCUR |
| 1 | 1 | 1 | YCUR |

Also SAC performs a LDF, i.e. loads PT and AB with AC2 resp. AC3.

Remark: In addition to SAC the analog source is (re)selected by the instructions LDF, EXC and EXL, i.e. by "110 000 110 xly" according to the bits 2,3 and 6 of the AC:

```
register pair:  if AC2 = 0
                then INTH
                else if AC3 = 0
                    then SH
                    else CUR;
```

```
coordinate:    if y = 0  $\wedge$  AC6 = 0
                then X
                else Y;
```

In other words: EXC acts like SAC, but LDF and EXL always select the Y-source.

4.1.2. LDA

LDA (=6074) loads the D register from AC2 - 11; this value will be converted to a voltage and tested against the selected source. Since this conversion takes some time, you have to wait some time, e.g. 50 μ s, before executing the SDA instruction, but there is no flag indicating that the conversion has been completed.

Remarks:

The following sequence of instructions has the same effect (as LDA):

```
LDX          / load X
SRF          / wait for flag
JMP .-1
```

In this case we wait about twice as long as necessary, however. - DEC specifies a delay of 50 μ s after LDA, before executing SDA. We found, however, that a delay of 18 μ s is sufficient on each of the KV8/I's available to us (on one of them even 1.5 μ s was

enough). In the routine of 4.2 this means that the instruction CLA CLL CML can be replaced by CLA CLL IAC RTL, reducing the execution time to about 0.5 ms.

4.1.3. SDA

Finally the instruction SDA (=6073) actually performs the test. It skips when the digital trial value was too large. Note that AC2 is a sign bit, i.e. $1000 < 1777 < 0000 < 0777$.

4.2. A/D conversion routine

```

/DEC-08-FISA-PA(L)
/KV ANALOG TO DIGITAL CONVERSION SUBROUTINE
/CALL AS JMS ADCONV WITH SOURCE SELECTION CODE IN AC:
/0000 OR 0400 FOR X INTEGRATOR
/0040 OR 0440 FOR Y INTEGRATOR
/1000 FOR X SAMPLE AND HOLD REGISTER
/1040 FOR Y SAMPLE AND HOLD REGISTER
/1400 FOR X CURSOR
/1440 FOR Y CURSOR
/EXITS WITH FULL 2'S COMPLEMENT ANS IN AC
/REQUIRES 25 CONSECUTIVE IN PAGE REGISTERS (DECIMAL)
/APPROXIMATE 900 US TO COMPLETE THE CONVERSION WORST CASE

ADCONV,      0
      SAC
      CLA CLL CML RTR      /2000 IN AC
      DCA TEST      /INITIALIZE TRIAL BIT REGISTER
      TAD K7000      /INITIALIZE PARTIAL CONVERSION REGISTER
RC1,      DCA TEMP      /SAVE PARTIAL CONVERSION
      TAD TEST
      CLL RAR
      DCA TEST      /ADVANCE TRIAL BIT
      TAD TEMP
      SZL
      JMP I ADCONV      /EXIT WHEN TRIAL BIT OVERFLOWS LINK
      TAD TEST
      LDA
      CLA CLL CML
      RAR
      SNL
      JMP .-2      /WAIT 54 US
K7000,      NOP
      TAD TEMP
      SDA
      TAD TEST      /SKIPPED IF TRIAL TOO LARGE
      JMP RC1      /CONTINUE

TEMP,      7000      /PARTIAL CONVERSION
TEST,      2000      /TRIAL BIT
$

```

5. The instructions

| octal value | mnemonic | description | | | | | | | | | | | | | | | |
|------------------------------|----------|--|-----|-----|--|---|---|-------------------------|---|---|----------------------|---|---|------------|---|---|-------------|
| 5.1. <u>device number 05</u> | | | | | | | | | | | | | | | | | |
| 6050 | - | No operation, i.e. delay of 4.25 μ s | | | | | | | | | | | | | | | |
| 6051 | SNC | Skip when cursor flag is low | | | | | | | | | | | | | | | |
| 6052 | CCF | Clear cursor flag | | | | | | | | | | | | | | | |
| 6053 | - | No operation | | | | | | | | | | | | | | | |
| 6064 | - | No operation | | | | | | | | | | | | | | | |
| 6055 | - | Identical with 6051 | | | | | | | | | | | | | | | |
| 6056 | - | Identical with 6052 | | | | | | | | | | | | | | | |
| 6057 | - | No operation | | | | | | | | | | | | | | | |
| 5.2. <u>device number 06</u> | | | | | | | | | | | | | | | | | |
| 6060 | - | No operation | | | | | | | | | | | | | | | |
| 6061 | - | No operation | | | | | | | | | | | | | | | |
| 6062 | SAC | Select analog comparator Select analog source according to AC2, AC3 and AC6. (4.1.1). Select mode of operation according to AC2 and AC3 (see LDF). Clear the Vector Continue (VC) flip-flop. | | | | | | | | | | | | | | | |
| 6063 | LDF | Load Format Select the mode of operation (3.4) according to AC2 and AC3: <table border="1"> <thead> <tr> <th>AC2</th> <th>AC3</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>incremental vector mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>absolute vector mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>point mode</td> </tr> <tr> <td>1</td> <td>1</td> <td>cursor mode</td> </tr> </tbody> </table> also the analog source is selected (see SAC 4.1.1). Clear the Vector Continue flip-flop. | AC2 | AC3 | | 0 | 0 | incremental vector mode | 0 | 1 | absolute vector mode | 1 | 0 | point mode | 1 | 1 | cursor mode |
| AC2 | AC3 | | | | | | | | | | | | | | | | |
| 0 | 0 | incremental vector mode | | | | | | | | | | | | | | | |
| 0 | 1 | absolute vector mode | | | | | | | | | | | | | | | |
| 1 | 0 | point mode | | | | | | | | | | | | | | | |
| 1 | 1 | cursor mode | | | | | | | | | | | | | | | |

If cursor mode is selected the cursor is displayed.

6064 LDX

Load X

6065 LDY

Load Y

effect:

- the ready flag is cleared
- AC2-11 is loaded into the D-register. This number is converted into a voltage V. Then XSH (resp. YSH) is loaded with:

if AB = 0 then V else V-XINTH (resp. YINTH), provided that the mode is different from cursor mode.
- The ready flag is raised after $100 \pm 30 \mu\text{s}$ (according to DEC) or $164 \mu\text{s}$ (on our installation).

6066 EXC

Execute

6067 EXL

Execute long

effect:

- The ready flag is cleared
- AC2-11 is loaded into the D-register. These bits determine the function to be executed:

AC2 & AC3 select the mode of operation: (see LDF and 3.4) and AC2, AC3 and AC6 select the analog source (see SAC 4.1.1).

AC4 & AC6 select the value for the L/S flip-flop:

EXC: L/S is cleared unless AC4 = 0 and AC6 = 1 and set otherwise.

EXL: L/S is set unless AC4 = 1 and cleared otherwise.

AC5 has no significance.

AC7 is loaded into VC. When VC = 1 the stroke timer (ST) is disabled and the function is executed continually.

if AC9 = 1 the display is erased. The RF is set after 0.5 ± 0.1 sec.

if AC10 = 1 the analog registers XSH, YSH, XINTH and YINTH are reset to zero. The reset holds on for the duration of the ST or VC.

If in cursor mode, the cursor is displayed. (visible as a small non-stored ellipse, independent of AC11).

If in point mode, a point is drawn at the (XSH,YSH) position; AC11 determines its visibility. XSH, YSH, XINTH and YINTH are not changed. {The use of invisible points is not very clear.}

If in vector mode, a vector is drawn beginning at (XINTH,YINTH).

If AC11 = 1 it will be visible else invisible.

If AC8 = 1 it will be a circular vector with centre (XSH,YSH) else it will be a linear vector going to (XINTH+XSH,YINTH+YSH).

If L/S = 1 it will be long, otherwise short. The timing is set by ST (Stroke Timer) and is 250 ± 5 μ s for short vectors and 4.05 ms for long vectors. (Long vector timing lasts exactly 2700 machine

cycles of $1.5 \mu\text{s}$ each; short vector timing is 1/16th of this.)

- The ready flag is set after:

$0.5 \pm 0.1 \text{ sec}$ if $\text{AC9} = 1$ (erase).

$25 \pm 5 \mu\text{s}$ if $\text{AC2} + \text{AC7} \geq 1$ (point mode, cursor mode or continuous mode).

$275 \pm 5 \mu\text{s}$ if $\text{L/S} = 0$ (short vector).

4.08 ms if $\text{L/S} = 1$ (long vector).

(First the Execute Delay Timer (XDT, $24 \pm 5 \mu\text{s}$) is activated. Then if in vector mode, the ST is activated unless it is disabled by VC.)

Remarks:

- The ERASE - button.

Pushing the ERASE-button on the panel at the right-hand side of the scope has the following effect:

- The display is erased, provided that the previous time it was erased (either by hand or by the program) was at least ca. 0.4 sec. earlier.
- When ready, the RF is set.

Note:

RF is not cleared by pushing the ERASE-button. If the scope was executing another operation while the ERASE-button was pushed, or initiated a new operation thereafter, RF will be set when the erase is completed (not sooner or later).

- EXC and EXL never change XSH and YSH unless $\text{AC10} = 1$ (reset). ERASE (both manual and by program) does not change XSH, YSH, XINTH and YINTH.

During the drawing of a vector XINTH and YINTH always point to the position where the beam is writing.

- A long vector should be drawn if the coordinate distance is more than $40g = 32$ positions; when short vectors are drawn over long distances they do not remain visible, i.e. they are only visible at the moment they are drawn. Also long vectors should not be longer than $1000g = 512$

positions, and not too short, since in the latter case the line becomes awfully thick. However, if the vector is drawn invisible, then it is advisable to use two short vectors with the same (!) destination instead of one long vector. [One vector (short or long) will generally not reach its destination; but we found that two short vectors suffice. Moreover, two short vectors take less time to draw than one long vector].

- When a circular vector is drawn, the angular velocity is 1 degree/45 μ s.

5.3. device number 07

| | | |
|------|-----|---|
| 6070 | - | No operation |
| 6071 | SRF | Skip on ready flag high |
| 6072 | CRF | Clear ready flag |
| 6073 | SDA | Skip if digital value larger than analog source |
| 6074 | LDA | Load digital to analog converter (see 4.1.2) |
| 6075 | - | SRF LDA |
| 6076 | - | CRF LDA |
| 6077 | - | SDA LDA |

6. Examples

6.1. The program

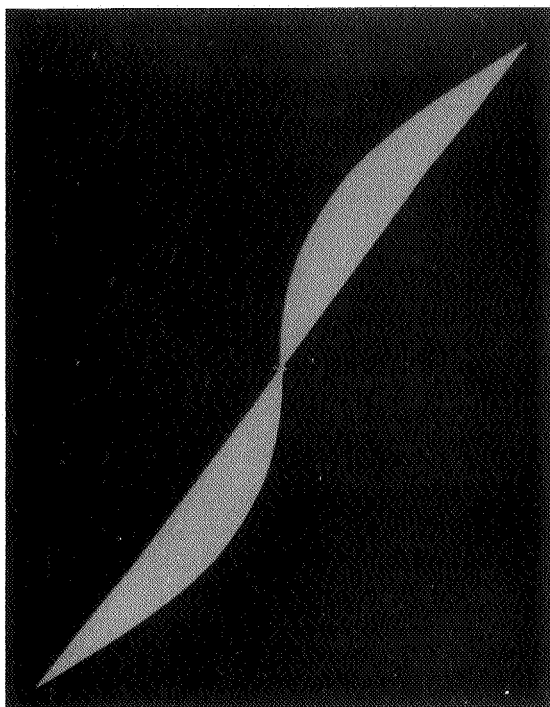
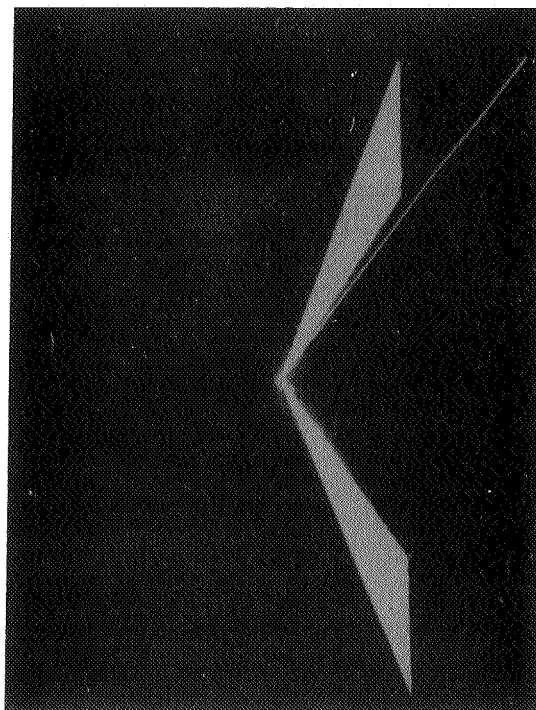
```

      TAD      (400
      LDF          /ABSOLUTE
AGAIN,  CLA
      LDX          /XSH:=-XINTH
      WAIT
      LDY          /YSH:=-YINTH (="+OR- YO)
      WAIT
      TAD      (441  /LONG VISIBLE ABSOLUTE STRAIGHT LINE
      EXC          /TO THE ORIGIN
      WAIT
I1,    TAD      (200
I2,    LDX
I3,    WAIT
      TAD      (441
      EXC          /TO (200 ,+OR-YO)
      WAIT
      JMP      AGAIN
WAIT=JMS .
0;SRF;JMP .-1;CLA;JMP I .-4
$

```

will draw the following figure:

i.e. a pair of lines symmetric w.r.t. the X-axis, approaching each other (because of the drift of the YSH).



Likewise, when the instructions labeled I1, I2, I3 are removed the figure at the left is drawn.

6.2. The program

| | | | |
|--------|---------------------|---------------------------------|-------------|
| START, | CLA CLL CML IAC RTL | TAD | CY |
| | EXC | TAD | (400 |
| | WAITSC | LDY | |
| | TAD (-1000 | WAITSC | |
| | DCA CX | TAD | (1001 |
| LOOP0, | TAD (-1000 | EXC | |
| | DCA CY | WAITSC | |
| | ISZ CX | JMP | LOOP |
| | SKP | RDY, | CLA |
| | JMP RDY | | JMP I (7600 |
| LOOP, | TAD CX | CX, | 0 |
| | TAD (400 | CY, | 0 |
| | LDX | | |
| | WAITSC | WAITSC=JMS I | [. |
| | ISZ CY | 0;SRF;JMP .-1;CRF;CLA;JMP I .-5 | |
| | SKP | \$ | |
| | JMP LOOP0 | | |

fills a large square with points. The reader is invited to execute this program and examine its results closely. We found the lines $X = 0$ and $Y = 0$ missing, this phenomenon being more clearly visible on some installations than on others. (When you see a massive square, retry with lower intensity.)

6.3. The execution of one of the instructions LDX, LDY and LDA while a Vector Continue operation is being done, or the execution of a scope IOT while the RF cleared by the previous operation is still down, can give all kinds of peculiar effects and should therefore be avoided in general.

However, one useful application is the sequence

```

TAD X
LDX          / load X
CLA STL
RAR
SNA
JMP .-2      / wait a short time
TAD Y
LDY

```

```

SRF
JMP .-1      / wait the usual time
TAD FORMAT
EXC

```

This loads X and Y correctly and reduces the execution time with about 0.1 ms, that is, with up to 30%.

6.4. The angular velocity of a circular vector

Independent of the radius, a circular vector is drawn at a rate of 1° per $45 \mu\text{s}$. This implies:

1. To draw an arc of N degrees, set CNT to $-10_{10} * N$;
start the arc with

```
TAD [31; EXC; SRF; JMP .-1;
```

and wait with

```
ISZ CNT; JMP .-1; LDF; CLA.
```

(Remember that LDF stops Vector Continue.)

2. TAD [51; EXC; SRF; JMP .-1
gives an arc of exactly 90° since long vector timing is $90 * 45 \mu\text{s} = 4.05 \text{ ms}$.
3. TAD [11; EXC; SRF; JMP .-1
gives an arc of $5\frac{5}{8}$ degrees.

6.5. List of useful codes

For some codes COD the effect of TAD COD; EXC is listed.

| <u>code</u> | <u>effect</u> |
|-------------|------------------------------------|
| 0 | invisible short incremental vector |
| 1 | visible short incremental vector |
| 401 | visible short absolute vector |
| 441 | visible long absolute vector |
| 2 | reset |
| 4 | erase |
| 6 | erase and reset |
| 26 | erase; continuous reset |
| 42 | reset for 4 ms |

1001 visible point
 11 $5\frac{5}{8}^\circ$ arc
 51 90° arc
 31 start circular arc (clockwise)
 1400 display cursor
 1422 reset; display cursor

7. PRFIG

```

2      /PRFIG
3      /PRFIG DISPLAYS A FIGURE COMPOSED OF (IN)VISIBLE INCREMENTAL
4      /VECTORS. EACH VECTOR CAN BE DRAWN IN ONE OF THE FOLLOWING DIRECTIONS:
5      /0:      0,DX
6      /1:      DX,DX
7      /2:      DX,0
8      /3:      DX,-DX
9      /4:      0,-DX
10     /5:      -DX,-DX
11     /6:      -DX,0
12     /7:      -DX,DX
13     /CALLING SEQUENCE:
14     /      JMS PRFIG
15     /      ARG1
16     /      ARG2
17     /      .
18     /      .
19     /      ARGN
20     /ARG1 CONTAINS THE NEGATIVE VALUE OF THE NUMBER OF VECTORS TO BE DRAWN,
21     /THE OTHER CONTAINS THE CODE OF THE VECTORS.
22     /THREE VECTORS ARE CODED IN ONE ARGUMENT, ARG[3:5],ARG[6:8] AND
23     /ARG[9:11] DETERMINE THE DIRECTION OF THE VECTORS. STROKE 1 ('LE''LE'
24     /2) IS VISIBLE IFF ARG[1]=1.
25
26     0200 0000 PRFIG, 0
27     0201 7440          SZA
28     0202 3310          DCA      DX
29     0203 6214          RDF
30     0204 1377          TAD      (CDF CIF
31     0205 3235          DCA      EXFIG /PREPARE A PROPER RETURN
32     0206 1600          TAD I    PRFIG
33     0207 3240          DCA      CTR      /COUNTS VECTORS
34     0210 7346 LPFIG,  CLA CLL  CMA RTL /AC:=-3
35     0211 3241          DCA      CTR1   /COUNTS THE STROKES
36     0212 2200          ISZ      PRFIG  /SELECT NEXT PARAMETER
37     0213 1600          TAD I    PRFIG
38     0214 0376          AND      (7000
39     0215 3242          DCA      VIS      /VISIBILITY BITS
40     0216 1600          TAD I    PRFIG
41     0217 3243          DCA      ARG      /TAKE ARGUMENT
42     0220 1243 AGFIG,  TAD      ARG
43     0221 7104          CLL RAL      /SELECT
44     0222 7006          RTL      /NEXT STROKE
45     0223 3243          DCA      ARG
46     0224 1243          TAD      ARG
47     0225 0376          AND      (7000 /ISOLATE DIRECTION CODE
48     0226 4244          JMS      STROKE /DRAW THE VECTOR
49     0227 2240          ISZ      CTR      /ALL VECTORS DONE?

```

```

50 0230 7410 SKP
51 0231 5235 JMP EXFIG /YES
52 0232 2241 ISZ CTR1 /SELECT NEXT ARGUMENT?
53 0233 5220 JMP AGFIG /NO
54 0234 5210 JMP LPFIG /YES
55 0235 6203 EXFIG, CDF CIF
56 0236 2200 ISZ PRFIG
57 0237 5600 JMP I PRFIG
58
59 /COUNTERS
60 0240 0000 CTR, 0
61 0241 0000 CTR1, 0
62
63 0242 0000 VIS, 0
64 0243 0000 ARG, 0
65
66 /THE ROUTINE STROKE COMPUTES THE DESTINATION COORDINATES AND DISPLAYS
67 /THE VECTOR. THE VECTOR IS DRAWN VISIBLE IFF VIS[0] IS 1.
68 / AC[0:2] DETERMINE THE DIRECTION OF THE STROKE.
69
70 0244 0000 STROKE, 0
71 0245 3275 DCA A /DIRECTION CODE
72 0246 4311 JMS WAITSC /WAIT FOR FLAG
73 0247 6063 LDF
74 0250 7332 CLA CLL CML RTR /2000
75 0251 4300 JMS DECOD /AC BECOMES 0,-DX OR DX
76 0252 3277 DCA Y0 /Y-COORDINATE OF THE DESTINATION POINT
77 0253 4300 JMS DECOD /AC BECOMES 0,-DX OR DX
78 0254 3276 DCA X0 /X-COORDINATE OF THE DESTINATION POINT
79 0255 1276 TAD X0
80 0256 6064 LDX /LOAD X COORDINATE
81 0257 7320 CLA CLL CML
82 0260 7004 RAL
83 0261 7420 SNL
84 0262 5260 JMP ,-2 /WAIT A MOMENT
85 0263 1277 TAD Y0
86 0264 6065 LDY /LOAD Y COORDINATE
87 0265 4311 JMS WAITSC /WAIT FOR FLAG
88 0266 1242 TAD VIS
89 0267 7104 CLL RAL
90 0270 3242 DCA VIS /UPDATE VIS
91 0271 7004 RAL /ROTATE BIT INTO THE AC
92 0272 6066 EXC /EXECUTE SHORT INCREMENTAL VECTOR
93 0273 7300 CLA CLL
94 0274 5644 JMP I STROKE
95
96 0275 0000 A, 0
97 0276 0000 X0, 0
98 0277 0000 Y0, 0
99
100 /DECOD IS USED TO COMPUTE THE PROPER RELATIVE COORDINATES FROM
101 /THE DIRECTION CODE OF STROKE(WHICH IS CONTAINED IN A).
102 /INPUT-OUTPUT BEHAVIOUR:
103 /AC[0:2]+A AC
104 /0000 0
105 /1000 DX
106 /2000 DX
107 /3000 DX
108 /4000 0
109 /5000 -DX
110 /6000 -DX
111 /7000 -DX
112
113 0300 0000 DECOD, 0
114 0301 1275 TAD A
115 0302 7104 CLL RAL
116 0303 7640 SZA CLA
117 0304 1310 TAD DX
118 0305 7430 SZL
119 0306 7041 CIA

```

```

120 0307 5700          JMP I  DECOD
121
122                /PARAMETER DECOD,STROKE,PRFIG
123 0310 0000 DX,      0
124
125 0311 0000 WAITSC, 0
126 0312 7200          CLA
127 0313 6071          SRF
128 0314 5313          JMP      ,-1
129                /CRF
130 0315 5711          JMP I  WAITSC
131
132                /SETPNT DEFINES THE X,Y INTH REGISTERS BY DRAWING A SHORT INVISIBLE
133                /ABSOLUTE VECTOR TO THE POINT (C0,C1). WHEN THE LENGTH OF THE
134                /VECTOR IS GREATER THAN 40 THE ROUTINE SHOULD BE CALLED TWICE.
135                /THIS IS FASTER AND MORE ACCURATE THAN DRAWING ONE INVISIBLE
136                /LONG VECTOR.
137
138 0316 0000 SETPNT, 0
139 0317 4311          JMS      WAITSC /WAIT FOR FLAG
140 0320 1375          TAD      (400 /ABSOLUTE
141 0321 6063          LDF
142 0322 7200          CLA
143 0323 1337          TAD      C0
144 0324 6064          LDX                /LOAD X COORDINATE
145 0325 7201          CLA  IAC
146 0326 7004          RAL
147 0327 7420          SNL
148 0330 5326          JMP      ,-2 /WAIT A MOMENT
149 0331 7104          CLL  RAL /CLEAR AC,BUT SAVE THE LINK
150 0332 1340          TAD      C1
151 0333 6065          LDY                /LOAD Y COORDINATE
152 0334 4311          JMS      WAITSC /WAIT FOR FLAG
153 0335 6066          EXC                /EXECUTE
154 0336 5716          JMP I  SETPNT
155
156                /PARAMETERS SETPNT
157 0337 0000 C0,      0
158 0340 0000 C1,      0
159 0375 0400
160 0376 7000
161 0377 6203
162                $$
163
A 0275 AGFIG 0220 ARG 0243 CTR 0240 CTR1 0241
C0 0337 C1 0340 DECOD 0300 DX 0310 EXFIG 0235
LPFIG 0210 PRFIG 0200 SETPNT 0316 STROKE 0244 VIS 0242
WAITSC 0311 X0 0276 Y0 0277

```

| | | | | | | | | |
|-------|-----|------|------|-----|-----|----|----|--|
| A | 71 | 96# | 114 | | | | | |
| AGFIG | 42# | 53 | | | | | | |
| ARG | 41 | 42 | 45 | 46 | 64# | | | |
| CTR | 33 | 49 | 60# | | | | | |
| CTR1 | 35 | 52 | 61# | | | | | |
| C0 | 143 | 157# | | | | | | |
| C1 | 150 | 158# | | | | | | |
| DECOD | 75 | 77 | 113# | 120 | | | | |
| DX | 28 | 117 | 123# | | | | | |
| EXC | 92 | 153 | | | | | | |
| EXFIG | 31 | 51 | 55# | | | | | |
| LDF | 73 | 141 | | | | | | |
| LDX | 80 | 144 | | | | | | |
| LDY | 86 | 151 | | | | | | |
| LPFIG | 34# | 54 | | | | | | |
| PRFIG | 26# | 32 | 36 | 37 | 40 | 56 | 57 | |

| | | | | | |
|--------|------|-----|------|-----|---------|
| SETPNT | 138# | 154 | | | |
| SRF | 127 | | | | |
| STROKE | 48 | 70# | 94 | | |
| VIS | 39 | 63# | 88 | 90 | |
| WAITSC | 72 | 87 | 125# | 130 | 139 152 |
| X0 | 78 | 79 | 97# | | |
| Y0 | 76 | 85 | 98# | | |
| °L0375 | 140 | | | | |
| °L0376 | 38 | 47 | | | |
| °L0377 | 30 | | | | |

8. Character generators

8.1. A two-page character generator

```

2          /PS/8 HANDLER FOR KV8/I COMPUTER DISPLAY.
3          /USES TWO PAGES OF CORE AND DISPLAYS THE COMPLETE
4          /ASCII CHARACTER SET. (SORRY THAT BACK ARROW, ↑ AND
5          /SQUARE BRACKETS ARE BAD).
6          /WRITTEN BY FLOOR ANTHONI, BIOMEDICAL LABS OF THE
7          /NATIONAL RESEARCH ORGANISATION, TNO, RYSWYK, HOLLAND.
8          /MODIFIED BY AEB, MC, AMSTERDAM
9
10         *5600
11         /THIS PAGE CONSISTS OF A NORMAL PS8 HANDLER ENTRY
12
13         5600 0000 DPL,      0          /ENTRY POINT
14         5601 7200          CLA          /JUST TO BE SURE
15         5602 5245          JMP DPLTEM   /BECOMES 0 AFTER THE FIRST TIME
16         5603 6214          RDF          /PICK UP THE RETURN FIELD.
17         5604 1256          TAD DPLCIF   /+6203
18         5605 3242          DCA DPLXIT   /CREATE RETURN
19         5606 1600          TAD I DPL    /FIELD OF BUFFER
20         5607 0276          AND C70
21         5610 1256          TAD DPLCIF   /+6203
22         5611 3254          DCA DPLCDF
23         5612 7330          CLA CLL CML RAR /AC=4000
24         5613 1600          TAD I DPL    /CHECK FOR READ OR WRITE
25         5614 2200          ISZ DPL
26         5615 0320          AND C7700   /NO. OF PAGES TO WRITE
27         5616 7550          SPA SNA      /READ IS BAD, NUMB OF PAGES
28         5617 5240          JMP DPLERR  /MUST BE >0
29         5620 7041          CIA
30         5621 3250          DCA DPLWC    /WORDCOUNT =DOUBLE PAGES
31         5622 1600          TAD I DPL    /GET BUFFER ADDRESS
32         5623 2200          ISZ DPL      /FOR CORRECT RETURN
33         5624 3247          DCA DPLCA    /AND USE AS ADDRESS POINTER
34
35
36         /HERE COMES THE PROGRAM MAIN FRAME:UNPACK THE CHARS
37         /FROM THE BUFFER AND DISPLAY THEM.
38
39         5625 4253 DPLGET, JMS GETBUF    /PICK UP A CHAR
40         5626 3246          DCA DPLTM
41         5627 4253          JMS GETBUF    /NEXT CHARACTER
42         5630 7112          CLL RTR
43         5631 7012          RTR
44         5632 1246          TAD DPLTM    /+4 M.S.BITS
45         5633 7012          RTR
46         5634 7012          RTR
47         5635 4644          JMS I DISP   /AND DISPLAY
48         5636 2250          ISZ DPLWC    /WORDCOUNT OVERFLOW?
49         5637 5225          JMP DPLGET  /NOT YET

```



```

50 5640 2200 DPLERR, ISZ DPL          /IN CASE OF ERROR ACC0=HARD ERR,
51 5641 2200          ISZ DPL
52 5642 7402 DPLXIT, HLT            /CDF CIF TO RETURN FIELD
53 5643 5600          JMP I DPL
54          /IN THE FOLLOWING LOCATIONS A ONCE ONLY CODE FIXES
55          /THE INDIRECT ADDRESSES USED IN THIS PROGRAM,
56          /FOR RELOCATABILITY.
57
58 5644 0132 DISP, DPLDIS=DPLTEM-1 /BECOMES POINTER TO DPLDIS
59 5645 4245 DPLTEM, JMS .
60 5646 1244 DPLTM, TAD DISP
61 5647 1245 DPLCA, TAD DPLTEM
62 5650 3244 DPLWC, DCA DISP
63 5651 0002          2          /DISPLAYWORD FOR TAB(=SPACE)
64 5652 3202          DCA DPL+2    /CLEAR THE JUMP
65 5653 4644 GETBUF, JMS I DISP    /WAIT FOR ERASE THE FIRST TIME
66 5654 5203 DPLCDF, JMP DPL+3    /POINTS TO FIELD OF BUF
67 5655 1647          TAD I DPLCA
68 5656 6203 DPLCIF, CDF CIF 0    /HANDLERS ALWAYS IN FIELD 0
69 5657 2247          ISZ DPLCA
70 5660 3245          DCA DPLTEM
71 5661 1245          TAD DPLTEM
72 5662 4644          JMS I DISP    /INTERRUPTS WERE INHIBITED FOR 19.5 US
73 5663 1245 GET3, TAD DPLTEM
74 5664 0357          AND C7400
75 5665 6031          KSF
76 5666 5653          JMP I GETBUF
77 5667 7200          CLA
78          /+N MEANS: SK.P THIS BUFFER
79          /ANY OTHER CHAR: SKIP ALL REMAINING BUFFERS
80 5670 6034          KRS
81 5671 1275          TAD DPLGO
82 5672 7650          SNA CLA
83 5673 6032          KCC
84 5674 5240          JMP DPLERR    /NORMAL RETURN
85
86 5675 7562 DPLGO, -216
87 5676 0070 C70, 70
88          *DPL+100
89          /HERE COMES THE LIST WITH DISPLAYWORDS
90          /ONE FOR EACH CHARACTER, THE TWO LS.BITS CODE FOR
91          /THE MASK TO TAKE.
92          LISTC=,
93 5700 0002          0002          /SPACE
94 5701 2403          2403          /°
95 5702 0123          0123          /"
96 5703 5071          5071          /#
97 5704 5355          5355          /$
98 5705 6145          6145          /!
99 5706 3171          3171          /^
100 5707 0023         0023          /'
101 5710 0110         0110          /{
102 5711 0204         0204          /)
103 5712 0374         0374          /*
104 5713 5001         5001          /+
105 5714 6003         6003          /,
106 5715 1001         1001          /-
107 5716 2003         2003          /.
108 5717 0300         0300          //
109 5720 7700         C7700, 7700    /0, ALSO USEFUL CONSTANT
110 5721 6011         6011          /1
111 5722 1731         1731          /2
112 5723 1715         1715          /3
113 5724 3405         3405          /4
114 5725 3215         3215          /5
115 5726 1375         1375          /6

```

| | | | | |
|-----|------|------|---------------------|---|
| 116 | 5727 | 2300 | 2300 | /7 |
| 117 | 5730 | 1775 | 1775 | /8 |
| 118 | 5731 | 1755 | 1755 | /9 |
| 119 | 5732 | 2203 | 2203 | /: |
| 120 | 5733 | 4203 | 4203 | /; |
| 121 | 5734 | 0700 | 0700 | /<< |
| 122 | 5735 | 1011 | 1011 | /= |
| 123 | 5736 | 0414 | 0414 | /> |
| 124 | 5737 | 1563 | 1563 | /? |
| 125 | 5740 | 3771 | 3771 | /° |
| 126 | 5741 | 3625 | 3625 | /A |
| 127 | 5742 | 3175 | 3175 | /B |
| 128 | 5743 | 0371 | 0371 | /C |
| 129 | 5744 | 7602 | 7602 | /D |
| 130 | 5745 | 1371 | 1371 | /E |
| 131 | 5746 | 1361 | 1361 | /F |
| 132 | 5747 | 1762 | 1762 | /G |
| 133 | 5750 | 1465 | 1465 | /H |
| 134 | 5751 | 4311 | 4311 | /I |
| 135 | 5752 | 7202 | 7202 | /J |
| 136 | 5753 | 4060 | 4060 | /K |
| 137 | 5754 | 0071 | 0071 | /L |
| 138 | 5755 | 5104 | 5104 | /M |
| 139 | 5756 | 5014 | 5014 | /N |
| 140 | 5757 | 7400 | 7400 | /O, ALSO USEFUL CONSTANT |
| 141 | 5760 | 1761 | 1761 | /P |
| 142 | 5761 | 7410 | 7410 | /Q |
| 143 | 5762 | 3436 | 3436 | /R |
| 144 | 5763 | 1355 | 1355 | /S |
| 145 | 5764 | 4301 | 4301 | /T |
| 146 | 5765 | 0475 | 0475 | /U |
| 147 | 5766 | 4300 | 4300 | /V |
| 148 | 5767 | 5210 | 5210 | /W |
| 149 | 5770 | 0314 | 0314 | /X |
| 150 | 5771 | 0304 | 0304 | /Y |
| 151 | 5772 | 2700 | 2700 | /Z |
| 152 | 5773 | 0161 | 0161 | /[|
| 153 | 5774 | 0014 | 0014 | /BACKSLASH |
| 154 | 5775 | 0605 | 0605 | /] |
| 155 | 5776 | 1467 | 1467 | /↑ |
| 156 | 5777 | 0170 | 0170 | /BACK ARROW |
| 157 | | | | |
| 158 | | | | /THIS IS THE ACTUAL DISPLAY ROUTINE WHICH CHECKS FOR |
| 159 | | | | /SPECIAL CHARACTERS, IGNORES NOTHING FOR HONEST DISPLAY |
| 160 | | | | /(CONTROL CHARS WILL BE VISUALISED), CREATES AN INTEN- |
| 161 | | | | /SIFY WORD AND SEARCHES THE MASKS, IT THEN HAPPENS |
| 162 | | | | /TO DISPLAY THE CHARACTER. |
| 163 | 6000 | 0000 | DPLDIS, 0 | /ENTER WITH CHAR IN AC, 8 BIT |
| 164 | 6001 | 5301 | JMP SAVE1 | /FIRST TIME ONLY CODE, BECOMES AND 377 |
| 165 | 6002 | 7450 | SNA | /IGNORE BLANKS |
| 166 | 6003 | 5600 | JMP I DPLDIS | |
| 167 | 6004 | 1347 | TAD M215 | |
| 168 | 6005 | 7450 | SNA | /CR? |
| 169 | 6006 | 5324 | JMP CR | /YES |
| 170 | 6007 | 7001 | IAC | /CTRL FORM? |
| 171 | 6010 | 7450 | SNA | |
| 172 | 6011 | 5336 | JMP FORM | /YES |
| 173 | 6012 | 1237 | TAD C2 | /LINEFEED? |
| 174 | 6013 | 7450 | SNA | |
| 175 | 6014 | 5327 | JMP LF | /YES |
| 176 | 6015 | 1350 | TAD BASE1 | /CREATE POINTER |
| 177 | 6016 | 3302 | DCA SAVE2 | /SAVE2 POINTS TO DISPLWORD OF THIS |
| 178 | 6017 | 1702 | TAD I SAVE2 | /MAKE 1 BLANK START VECTOR BIT |
| 179 | 6020 | 7110 | CLL RAR | |
| 180 | 6021 | 3301 | DPL6, DCA SAVE1 | /SAVE CONTAINS 11 INTENSIFY BITS AND |
| 181 | 6022 | 7325 | CLA CLL CML IAC RAL | /AC=3 MASK |
| 182 | 6023 | 0702 | AND I SAVE2 | /CHAR, PICK IT UP |
| 183 | 6024 | 7106 | RTL CLL | /MULTIPLY BY 8 |

```

184 6025 7004          RAL
185 6026 1367          TAD BASE2      /CREATE POINTER TO MASKWORD
186 6027 3240          DCA DPLINS     /POINTS TO MASKWORD
187
188
189
190 6030 1357  DPL1,   TAD M13        /COUNTS 5 AND A HALF MASK WORDS
191 6031 3303          DCA COUNT
192 6032 1303  DPL2,   TAD COUNT      /WHICH HALF?
193 6033 7010          RAR           /ODD=RIGHT;EVEN=LEFT
194 6034 7620          SNL CLA
195 6035 5274          JMP DPL4      /LEFT HALF
196 6036 7410          SKP
197 6037 0002  C2,     2              /DISPLAYWORD FOR RUBOUT (=SPACE)
198 6040 1341  DPLINS, TAD MASK0     /INDEXED INSTRUCTION
199 6041 2240          ISZ   DPLINS
200 6042 3302  DPL3,   DCA SAVE2     /CONTAINS Y1,X1,Y0,X0
201          /TAD   SAVE1
202          /RAL
203          /SPA SZL
204          /JMP   ,+3
205          /DCA   SAVE1
206          /JMP   DPLNOT
207 6043 1302          TAD SAVE2      /GET DELTA=X
208 6044 0360          AND C7
209 6045 1304          TAD X
210 6046 6064          LDX;WAIT
211 6047 4306
212 6050 1302          TAD SAVE2     /GET DELTA=Y
213 6051 7010          RAR
214 6052 7012          RTR
215 6053 0360          AND C7
216 6054 1305          TAD Y
217 6055 6065          LDY;WAIT
218 6056 4306
219 6057 1301          TAD SAVE1     /GET THE INTENSIFYWORD
220 6060 7104          CLL RAL      /SEPARATE NEXT BIT
221 6061 3301          DCA SAVE1
222 6062 7004          RAL           /AC=0 OR 1 (1=INTENS.BIT)
223 6063 1355          TAD C400      /SHORT VECTOR CODE
224 6064 6066          EXC;WAIT
225 6065 4306
226 6066 2303  DPLNOT, ISZ COUNT     /ALL ELEVEN VECTORS DONE?
227 6067 5232          JMP DPL2      /NOT YET
228 6070 1360          TAD C7        /YES, MOVE X RIGHT ONE POSITION
229 6071 1304  DPL7,   TAD X
230 6072 3304          DCA X
231 6073 5600          JMP I DPLDIS /RETURN
232
233 6074 1302  DPL4,   TAD SAVE2     /FOR LEFT HALF
234 6075 7012          RTR
235 6076 7012          RTR
236 6077 7012          RTR
237 6100 5242          JMP DPL3
238          /ONCE ONLY CODE FOR RELOCATABILITY
239
240 6101 1200  SAVE1,   TAD DPLDIS
241 6102 1350  SAVE2,   TAD BASE1
242 6103 3350  COUNT,  DCA BASE1
243 6104 1377  X,      TAD C377
244 6105 3201  Y,      DCA DPLDIS+1
245 6106 5317          JMP FORM2      /LOAD X,LOAD Y, DO VECTOR
246
247          WAIT=JMS ,.-1
248 6107 6071          SRP
249 6110 5307          JMP   ,.-1

```

```

250 6111 6072 CRF
251 6112 7200 CLA
252 6113 5706 JMP I , -5
253
254 /SPECIAL CHARACTERS HANDLER
255
256 6114 6031 KBTST, KSF
257 6115 5336 JMP FORM
258 /KRS
259 /TAD MCTN
260 /SZA CLA
261 6116 5600 JMP I DPLDIS
262 /KCC
263 6117 7327 FORM2, CLA CLL CML IAC RTL /AC=6=ERASE CODE
264 6120 6066 EXC;WAIT /ERASE
265 6121 4306
266 6122 1356 NEWP, TAD TOP /MOVE Y TO TOP
267 6123 3305 DCA Y
268 6124 1346 CR, TAD LEFT /MOVE X TO LEFT
269 6125 3304 DCA X
270 6126 5221 JMP DPL6 /DO A DUMMY SPACE BY CLEARING THE
271 /INTENSIFY WORD AND DOING A WILD SET
272 /OF MASKS.
273 6127 1357 LF, TAD M13 /MOVE Y 1 LINE DOWN
274 6130 1305 TAD Y
275 6131 3305 DCA Y
276 6132 1305 TAD Y /PAGE OVERFLOW?
277 6133 1356 TAD TOP /SYMMETRIC PAGE
278 6134 7700 SMA CLA
279 6135 5600 JMP I DPLDIS
280 6136 6071 FORM, SRF
281 6137 5314 JMP KBTST
282 6140 5322 JMP NEWP
283
284 6141 6000 MASK0, 6000
285 6142 0464 0464
286 6143 3200 3200
287 6144 3064 3064
288 6145 3204 3204
289 6146 7360 LEFT, 7360
290 6147 7763 M215, -15
291 6150 7776 BASE1, LISTC-DPLWC-32 /WILL POINT TO LISTC-26
292 6151 6202 MASK1, 6202
293 6152 3430 3430
294 6153 6264 6264
295 6154 3060 3060
296 6155 0400 C400, 0400 /ALSO USEFUL CONST.
297 6156 0534 TOP, 0534
298 6157 7765 M13, -13
299 6160 0007 C7, 7
300 6161 3402 MASK2, 3402
301 6162 6064 C6064, 6064
302 6163 0200 0200
303 6164 3404 3404
304 6165 3032 3032
305 6166 0003 C3, 0003
306 6167 1341 BASE2, TAD MASK0
307 6170 0177 C177, 177
308 6171 0213 MASK3, 0213
309 6172 2202 2202
310 6173 4242 4242
311 6174 6264 6264
312 6175 4440 SPACE, 4440
313 6176 7562 MCTN, 7562
314
315 6177 0370 C377, AND C177
316 $

```


| | | | | | | | | |
|-------|------|------|------|------|------|-----|-----|------|
| NEWP | 266# | 282 | | | | | | |
| SAVE1 | 164 | 180 | 219 | 221 | 240# | | | |
| SAVE2 | 177 | 178 | 182 | 200 | 207 | 212 | 233 | 241# |
| SPACE | 312# | | | | | | | |
| SRF | 248 | 280 | | | | | | |
| TOP | 266 | 277 | 297# | | | | | |
| WAIT | 211 | 218 | 225 | 247# | 265 | | | |
| X | 209 | 229 | 230 | 243# | 269 | | | |
| Y | 216 | 244# | 267 | 274 | 275 | 276 | | |

8.2. A three-page character generator

```

1          /VSCG
2          /VARIABLE STROKE CHARACTER GENERATOR.
3          /FROM: EDGRIN (MURRAY RUBEN, 1969)
4          /PB, MC, 310572
5
6          LOCATE=1000      /ARBITRARY
7
8          *LOCATE
9
10         1000 0000 DSPY, 0          /CALL WITH ASCII CHAR IN AC
11         1001 4766 JMS I SRCHI      /SEARCH CHAR ?
12         1002 1765 TAD I SAVE2      /CHECK FOR CONTR CHAR
13         1003 1354 TAD M240
14         1004 7710 SPA CLA
15         1005 5600 DSPYI, JMP I DSPY  /EXIT
16         1006 1765 TAD I SAVE2
17         1007 0361 OUTCHM, AND C77 /MASK CHARACTER TO 6 BITS
18         1010 7104 CLL RAL /*2
19         1011 1370 TAD SAR1        /ADD DISPATCH
20         1012 3347 DCA A
21         1013 1747 TAD I A /THIS IS MASK WORD
22         1014 0357 AND C7 /MASK OFF MASK BITS
23         1015 7104 CLL RAL /*2
24         1016 3350 DCA B
25         1017 1350 TAD B
26         1020 7104 CLL RAL /*4
27         1021 1350 TAD B /*6 NOW
28         1022 1371 TAD MASKS      /6 TIMES MASK + MASK HEAD POINTER
29         1023 3351 DCA C /PTR FOR MASK ADDRESSES
30         1024 1747 TAD I A
31         1025 7110 CLL RAR
32         1026 3350 DCA B /SETS INTENSIFY BLANKING BITS, FIRST ALWAYS BLANKED
33         1027 2347 ISZ A
34         1030 1747 TAD I A
35         1031 3347 DCA A /RETRIEVE CONTROL WORD
36         1032 3352 DCA H /SETS LEFT HALF
37         1033 1352 VA, TAD H
38         1034 7041 CIA
39         1035 3352 DCA H /RESET HALFWORD SWITCH
40         1036 2352 ISZ H
41         1037 2351 ISZ C /ADVANCE PTR ON ZERO H
42         1040 1347 TAD A
43         1041 7440 SZA /TEST NEXT CONTROL BIT
44         1042 5250 JMP .+6
45         1043 1360 TAD C10
46         1044 4767 JMS I SCALE
47         1045 1345 TAD X0
48         1046 3345 DCA X0 /CHAR ADVANCE
49         1047 5600 JMP I DSPY /EXIT

```

| | | | | |
|-----|------|------|---------|---|
| 50 | 1050 | 7104 | | CLL RAL |
| 51 | 1051 | 3347 | | DCA A /NEXT CONTROL BIT IN LINK |
| 52 | 1052 | 7420 | | SNL |
| 53 | 1053 | 5233 | | JMP VA /NOT AN EXECUTION |
| 54 | 1054 | 1364 | | TAD C6064 /INITIALIZE VOUT |
| 55 | 1055 | 3315 | | DCA VSTATE |
| 56 | 1056 | 1352 | | TAD H /0 IF RIGHT HALF, 1 IF LEFT HALF |
| 57 | 1057 | 7110 | | CLL RAR /INTO LINK |
| 58 | 1060 | 1751 | | TAD I C /GET MASK ADDRESS |
| 59 | 1061 | 7420 | | SNL |
| 60 | 1062 | 5266 | | JMP ,+4 /R.H. |
| 61 | 1063 | 7112 | | CLL RTR |
| 62 | 1064 | 7012 | | RTR |
| 63 | 1065 | 7012 | | RTR /L.H. |
| 64 | 1066 | 3765 | | DCA I SAVE2 |
| 65 | 1067 | 1765 | | TAD I SAVE2 |
| 66 | 1070 | 7012 | | RTR |
| 67 | 1071 | 7010 | | RAR |
| 68 | 1072 | 0357 | | AND C7 |
| 69 | 1073 | 4767 | | JMS I SCALE |
| 70 | 1074 | 1345 | | TAD X0 /ADD X MASK TO CPR |
| 71 | 1075 | 4312 | | JMS VOUT /LOAD X ABSOLUTE |
| 72 | 1076 | 1765 | | TAD I SAVE2 |
| 73 | 1077 | 0357 | | AND C7 |
| 74 | 1100 | 4767 | | JMS I SCALE |
| 75 | 1101 | 1346 | | TAD Y0 /ADD Y MASK TO CPR |
| 76 | 1102 | 4312 | | JMS VOUT /LOAD Y ABSOLUTE |
| 77 | 1103 | 1350 | | TAD B |
| 78 | 1104 | 7104 | | CLL RAL |
| 79 | 1105 | 3350 | | DCA B /RETRIEVE NEXT BLANKING BIT INTO LINK |
| 80 | 1106 | 7004 | | RAL /AND THEN INTO BIT 11 |
| 81 | 1107 | 1362 | | TAD CEX /EXECUTE ABSOLUTE (VISIBLE) VECTOR |
| 82 | 1110 | 4312 | | JMS VOUT |
| 83 | 1111 | 5233 | | JMP VA /BACK FOR MORE VECTORS |
| 84 | | | | |
| 85 | 1112 | 0000 | VOUT, | 0 |
| 86 | 1113 | 6071 | | 6071 /WAIT FOR READY FLAG |
| 87 | 1114 | 5313 | | JMP .-1 |
| 88 | 1115 | 6064 | VSTATE, | 6064 /AUTO SEQUENCED INSTRUCTION |
| 89 | 1116 | 2315 | | ISZ VSTATE /SEQUENCE |
| 90 | 1117 | 7200 | | CLA |
| 91 | 1120 | 5712 | | JMP I VOUT |
| 92 | 1121 | 1363 | RESET, | TAD TOP /ERASE AND RESET INT ^ CPR |
| 93 | 1122 | 6066 | | 6066 /EXECUTE |
| 94 | 1123 | 3346 | | DCA Y0 /SET Y TO TOP |
| 95 | 1124 | 1355 | CRLF, | TAD M14 |
| 96 | 1125 | 4767 | | JMS I SCALE |
| 97 | 1126 | 1346 | | TAD Y0 |
| 98 | 1127 | 3346 | | DCA Y0 |
| 99 | 1130 | 1353 | CR, | TAD MARGIN |
| 100 | 1131 | 3345 | | DCA X0 |
| 101 | 1132 | 1356 | SYNC, | TAD M10 /"SYNCHRONIZE" |
| 102 | 1133 | 4767 | | JMS I SCALE |
| 103 | 1134 | 1345 | | TAD X0 |
| 104 | 1135 | 3345 | XCUTE, | DCA X0 |
| 105 | 1136 | 1354 | | TAD M240 |
| 106 | 1137 | 5207 | | JMP OUTCHM |
| 107 | 1140 | 1355 | LF, | TAD M14 |
| 108 | 1141 | 4767 | | JMS I SCALE |
| 109 | 1142 | 1346 | | TAD Y0 |
| 110 | 1143 | 3346 | | DCA Y0 |
| 111 | 1144 | 5332 | | JMP SYNC |
| 112 | | | | |
| 113 | 1145 | 0000 | X0, | 0 /X CHARACTER POSITION |
| 114 | 1146 | 0000 | Y0, | 0 /Y CHARACTER POSITION |

```

115 1147 0000 A, 0 /CONTROL WORD
116 1150 0000 B, 0 /MASK WORD
117 1151 0000 C, 0 /MASK POINTER
118 1152 0000 H, 0 /HALFWORD SWITCH
119
120 1153 7360 MARGIN, -420
121 1154 7540 M240, -240
122 1155 7764 M14, -14
123 1156 7770 M10, -10
124 1157 0007 C7, 7
125 1160 0010 C10, 10
126 1161 0077 C77, 77
127 1162 0400 CEX, 400
128 1163 0506 TOP, 506
129 1164 6064 C6064, 6064
130 1165 1230 SAVE2, SAVE1
131 1166 1231 SRCH1, SEARCH
132 1167 1251 SCALE, VSCALE
133 1170 1356 SAR1, SAR
134 1171 1303 MASKS, MASK0-1
135
136 DLIST=. /TABLE OF ACTIVE CONTROL CHARACTERS
137 1172 0212 212
138 1173 1140 LF /LINE FEED
139 1174 0215 215
140 1175 1130 CR /CARRIAGE RETURN
141 1176 0216 216
142 1177 1124 CRLF /NLCR (CTRL/N)
143 1200 0213 213
144 1201 1121 RESET /ERASE (CTRL/K)
145 1202 0377 377
146 1203 1005 DSPY1 /RUBOUT (IGNORED)
147 1204 0376 376
148 1205 1005 DSPY1 /ALTMODE (IGNORED)
149 1206 0211 211
150 1207 1265 TAB /TAB
151 1210 0237 237
152 1211 1132 SYNC /STABILIZE (CTRL/SHIFT/Q)
153 1212 0221 221
154 1213 1263 SCALE1 /ENLARGE (CTRL/Q)
155 1214 0222 222
156 1215 1262 SCALE0 /RESET FORMAT (CTRL/R)
157
158 1216 0000 0 /ZERO ENDS THE LIST
159 1217 7360 M420, -420
160 1220 7770 CM10, -10
161 1221 0100 C100, 100
162 1222 1171 SRCH3, DLIST-1
163 1223 1005 DSP, DSPY1
164 1224 1145 VX0, X0
165 1225 1135 EXECUTE, XCUTE
166
167 1226 0001 SCF, 1
168 1227 0000 SCT, 0
169 1230 0000 SAVE1, 0
170
171 1231 0000 SEARCH, 0 /SEARCH ROUTINE.
172 1232 3230 DCA SAVE1 /SAVE THE INPUT ASCII CHARACTER
173 1233 1222 TAD SRCH3
174 1234 3341 DCA WSP
175 1235 2341 SRCH2, ISZ WSP /FOLLOW ALONG DOWN THE CHARACTER TABLE.
176 1236 1741 TAD I WSP /BRING IN A LIST ELEMENT
177 1237 7450 SNA
178 1240 5631 JMP I SEARCH /END OF TABLE FOUND AND NO MATCH*
179 1241 7041 CIA /COMPLEMENT TO TEST.
180 1242 2341 ISZ WSP
181 1243 1230 TAD SAVE1 /NOW TEST AGAINST THE "CHAR",
182 1244 7640 SZA CLA
183 1245 5235 JMP SRCH2 /NO MATCH FOUND SO TRY AGAIN*
184 1246 1741 TAD I WSP /"MATCH FOUND" **

```



```

185 1247 3231 DCA SEARCH /DO DOUBLE INDIRECT JUMP
186 1250 5631 JMP I SEARCH /FROM DLIST POINTER
187
188 1251 0000 VSCALE, 0
189 1252 3227 DCA SCT
190 1253 1226 TAD SCF
191 1254 7041 CIA
192 1255 3341 DCA WSP
193 1256 1227 TAD SCT
194 1257 2341 ISZ WSP
195 1260 5256 JMP .-2
196 1261 5651 JMP I VSCALE
197
198 1262 3226 SCALED, DCA SCF
199 1263 2226 SCALE1, ISZ SCF
200 1264 5623 JMP I DSP
201
202 1265 1220 TAB, TAD CM10
203 1266 4251 JMS VSCALE
204 1267 1217 TAD M420
205 1270 3230 DCA SAVE1
206 1271 1221 TAB1, TAD C100
207 1272 4251 JMS VSCALE
208 1273 1230 TAD SAVE1
209 1274 3230 DCA SAVE1
210 1275 1624 TAD I VX0
211 1276 7041 CIA
212 1277 1230 TAD SAVE1
213 1300 7710 SPA CLA
214 1301 5271 JMP TAB1
215 1302 1230 TAD SAVE1
216 1303 5625 JMP I EXECUTE
217
218 /THE FOLLOWING PARTS ARE THE CHARACTER TABLES FOR THE CHARACTER
219 /GENERATOR .
220
221 /CHARACTER MASK COORDINATES
222
223
224 1304 0301 MASK0, 0301 /D,J,5,^,!,}
225 1305 2143 2143
226 1306 4145 4145
227 1307 2705 2705
228 1310 0747 0747
229 1311 0301 0301
230 1312 0701 MASK1, 0701 /L,U,V,W,X,Y,I,T,N,M,0,1,(,),}
231 1313 0706 0706
232 1314 2447 2447
233 1315 2724 2724
234 1316 2141 2141
235 1317 0147 0147
236 1320 0141 MASK2, 0141 /2,S,4,9,7,Z,$,/,\,<,>
237 1321 0747 0747
238 1322 4404 4404
239 1323 0747 0747
240 1324 0141 0141
241 1325 2127 2127
242 1326 0424 MASK3, 0424 /A,C,E,F,G,H,K,0,Q,R,P,3,6,8,L,8
243 1327 3444 3444
244 1330 4147 4147
245 1331 0701 0701
246 1332 4144 4144
247 1333 2404 2404
248 1334 2622 MASK4, 2622 /+,B,A,,-,+,*
249 1335 4305 4305

```

| | | | | | |
|-----|------|------|--------|------|---------------------------|
| 250 | 1336 | 4503 | 4503 | | |
| 251 | 1337 | 0426 | 0426 | | |
| 252 | 1340 | 4404 | 4404 | | |
| 253 | | | | | |
| 254 | 1341 | 0000 | WSP, | 0 | |
| 255 | | | | | |
| 256 | 1342 | 4525 | MASK5, | 4525 | /°,#,,= |
| 257 | 1343 | 0503 | 0503 | | |
| 258 | 1344 | 2343 | 2343 | | |
| 259 | 1345 | 4721 | 4721 | | |
| 260 | 1346 | 2707 | 2707 | | |
| 261 | 1347 | 0141 | 0141 | | |
| 262 | 1350 | 0627 | MASK6, | 0627 | /"',,!,!,?,°,,, |
| 263 | 1351 | 2525 | 2525 | | |
| 264 | 1352 | 4524 | 4524 | | |
| 265 | 1353 | 2323 | 2323 | | |
| 266 | 1354 | 1121 | 1121 | | |
| 267 | 1355 | 2147 | 2147 | | |
| 268 | | | SAR=. | | /CHARACTER DISPATCH TABLE |
| 269 | 1356 | 7745 | 7745 | | /° |
| 270 | 1357 | 6347 | 6347 | | |
| 271 | 1360 | 5603 | 5603 | | /A |
| 272 | 1361 | 4760 | 4760 | | |
| 273 | 1362 | 7703 | 7703 | | /B |
| 274 | 1363 | 2175 | 2175 | | |
| 275 | 1364 | 7003 | 7003 | | /C |
| 276 | 1365 | 0170 | 0170 | | |
| 277 | 1366 | 7700 | 7700 | | /D |
| 278 | 1367 | 3551 | 3551 | | |
| 279 | 1370 | 5603 | 5603 | | /E |
| 280 | 1371 | 5170 | 5170 | | |
| 281 | 1372 | 5403 | 5403 | | /F |
| 282 | 1373 | 5160 | 5160 | | |
| 283 | 1374 | 7603 | 7603 | | /G |
| 284 | 1375 | 0176 | 0176 | | |
| 285 | 1376 | 5203 | 5203 | | /H |
| 286 | 1377 | 4760 | 4760 | | |
| 287 | 1400 | 5201 | 5201 | | /I |
| 288 | 1401 | 1156 | 1156 | | |
| 289 | 1402 | 7000 | 7000 | | /J |
| 290 | 1403 | 5404 | 5404 | | |
| 291 | 1404 | 6503 | 6503 | | /K |
| 292 | 1405 | 6172 | 6172 | | |
| 293 | 1406 | 6003 | 6003 | | /L |
| 294 | 1407 | 0070 | 0070 | | |
| 295 | 1410 | 7401 | 7401 | | /M |
| 296 | 1411 | 3304 | 3304 | | |
| 297 | 1412 | 7001 | 7001 | | /N |
| 298 | 1413 | 3005 | 3005 | | |
| 299 | 1414 | 7403 | 7403 | | /O |
| 300 | 1415 | 0370 | 0370 | | |
| 301 | 1416 | 7403 | 7403 | | /P |
| 302 | 1417 | 4560 | 4560 | | |
| 303 | 1420 | 7603 | 7603 | | /Q |
| 304 | 1421 | 2370 | 2370 | | |
| 305 | 1422 | 7503 | 7503 | | /R |
| 306 | 1423 | 4572 | 4572 | | |
| 307 | 1424 | 7602 | 7602 | | /S |
| 308 | 1425 | 6360 | 6360 | | |
| 309 | 1426 | 5001 | 5001 | | /T |
| 310 | 1427 | 1150 | 1150 | | |
| 311 | 1430 | 7001 | 7001 | | /U |
| 312 | 1431 | 6005 | 6005 | | |
| 313 | 1432 | 6001 | 6001 | | /V |
| 314 | 1433 | 4011 | 4011 | | |
| 315 | 1434 | 7401 | 7401 | | /W |
| 316 | 1435 | 6205 | 6205 | | |
| 317 | 1436 | 5001 | 5001 | | /X |
| 318 | 1437 | 1007 | 1007 | | |
| 319 | 1440 | 6401 | 6401 | | /Y |
| 320 | 1441 | 1330 | 1330 | | |

| | | | | |
|-----|------|------|------|--|
| 321 | 1442 | 7002 | 7002 | /Z |
| 322 | 1443 | 0074 | 0074 | |
| 323 | 1444 | 7001 | 7001 | /[|
| 324 | 1445 | 0154 | 0154 | |
| 325 | 1446 | 4002 | 4002 | /\ |
| 326 | 1447 | 0044 | 0044 | |
| 327 | 1450 | 7000 | 7000 | /] |
| 328 | 1451 | 3050 | 3050 | |
| 329 | 1452 | 5404 | 5404 | /† |
| 330 | 1453 | 6070 | 6070 | |
| 331 | 1454 | 6404 | 6404 | /B.A. |
| 332 | 1455 | 2074 | 2074 | |
| 333 | | | | |
| 334 | | | | |
| 335 | 1456 | 0000 | 0000 | /SP |
| 336 | 1457 | 7000 | 7000 | /EXECUTE 3 "INVISIBLE" VECTOR STROKES. |
| 337 | 1460 | 5006 | 5006 | /° |
| 338 | 1461 | 2046 | 2046 | |
| 339 | 1462 | 5006 | 5006 | /" |
| 340 | 1463 | 3201 | 3201 | |
| 341 | 1464 | 5245 | 5245 | /# |
| 342 | 1465 | 5572 | 5572 | |
| 343 | 1466 | 7642 | 7642 | /\$ |
| 344 | 1467 | 6363 | 6363 | |
| 345 | 1470 | 6740 | 6740 | /† |
| 346 | 1471 | 1675 | 1675 | |
| 347 | 1472 | 6760 | 6760 | /^ |
| 348 | 1473 | 3637 | 3637 | |
| 349 | 1474 | 4006 | 4006 | /' |
| 350 | 1475 | 3000 | 3000 | |
| 351 | 1476 | 6001 | 6001 | /(|
| 352 | 1477 | 0124 | 0124 | |
| 353 | 1500 | 6001 | 6001 | /) |
| 354 | 1501 | 1202 | 1202 | |
| 355 | 1502 | 5204 | 5204 | /* |
| 356 | 1503 | 7700 | 7700 | |
| 357 | 1504 | 5004 | 5004 | /+ |
| 358 | 1505 | 6014 | 6014 | |
| 359 | 1506 | 4006 | 4006 | /, |
| 360 | 1507 | 0030 | 0030 | |
| 361 | 1510 | 4004 | 4004 | /- |
| 362 | 1511 | 0014 | 0014 | |
| 363 | 1512 | 4006 | 4006 | ./ |
| 364 | 1513 | 0060 | 0060 | |
| 365 | 1514 | 4002 | 4002 | // |
| 366 | 1515 | 4400 | 4400 | |
| 367 | 1516 | 7601 | 7601 | /0 |
| 368 | 1517 | 3107 | 3107 | |
| 369 | 1520 | 6401 | 6401 | /1 |
| 370 | 1521 | 0456 | 0456 | |
| 371 | 1522 | 7602 | 7602 | /2 |
| 372 | 1523 | 1714 | 1714 | |
| 373 | 1524 | 5503 | 5503 | /3 |
| 374 | 1525 | 4770 | 4770 | |
| 375 | 1526 | 6402 | 6402 | /4 |
| 376 | 1527 | 0364 | 0364 | |
| 377 | 1530 | 7600 | 7600 | /5 |
| 378 | 1531 | 3434 | 3434 | |
| 379 | 1532 | 7603 | 7603 | /6 |
| 380 | 1533 | 0175 | 0175 | |
| 381 | 1534 | 6002 | 6002 | /7 |
| 382 | 1535 | 0070 | 0070 | |
| 383 | 1536 | 5703 | 5703 | /8 |
| 384 | 1537 | 4770 | 4770 | |
| 385 | 1540 | 7402 | 7402 | /9 |
| 386 | 1541 | 0364 | 0364 | |
| 387 | 1542 | 5006 | 5006 | /: |
| 388 | 1543 | 1460 | 1460 | |
| 389 | 1544 | 5006 | 5006 | /; |

```

390 1545 1430 1430
391 1546 6002 6002 /<
392 1547 2120 2120
393 1550 5005 5005 /=
394 1551 5500 5500
395 1552 6002 6002 />
396 1553 4240 4240
397 1554 7506 7506 /?
398 1555 6346 6346
399 /THIS IS THE LAST OF THE CHARACTER DISPATCH TABLE.....
400
401 /IT ALSO FINISHES THE CHARACTER GENERATOR PROGRAM.....
402
403 $
404

```

| | | | | | | | | | |
|--------|------|--------|------|-------|------|--------|------|--------|------|
| A | 1147 | B | 1150 | C | 1151 | CEX | 1162 | CM10 | 1220 |
| CR | 1130 | CRLF | 1124 | C10 | 1160 | C100 | 1221 | C6064 | 1164 |
| C7 | 1157 | C77 | 1161 | DLIST | 1172 | DSP | 1223 | DSPY | 1000 |
| DSPYI | 1005 | EXCUTE | 1225 | H | 1152 | LF | 1140 | LOCATE | 1000 |
| MARGIN | 1153 | MASKS | 1171 | MASK0 | 1304 | MASK1 | 1312 | MASK2 | 1320 |
| MASK3 | 1326 | MASK4 | 1334 | MASK5 | 1342 | MASK6 | 1350 | M10 | 1156 |
| M14 | 1155 | M240 | 1154 | M420 | 1217 | OUTCHM | 1007 | RESET | 1121 |
| SAR | 1356 | SAR1 | 1170 | SAVE1 | 1230 | SAVE2 | 1165 | SCALE | 1167 |
| SCALED | 1262 | SCALE1 | 1263 | SCF | 1226 | SCT | 1227 | SEARCH | 1231 |
| SRCHI | 1166 | SRCH2 | 1235 | SRCH3 | 1222 | SYNC | 1132 | TAB | 1265 |
| TAB1 | 1271 | TOP | 1163 | VA | 1033 | VOUT | 1112 | VSCALE | 1251 |
| VSTATE | 1115 | VX0 | 1224 | WSP | 1341 | XCUTE | 1135 | X0 | 1145 |
| Y0 | 1146 | | | | | | | | |

| | | | | | | | | | |
|--------|------|------|------|------|----|------|------|----|------|
| A | 20 | 21 | 30 | 33 | 34 | 35 | 42 | 51 | 115# |
| B | 24 | 25 | 27 | 32 | 77 | 79 | 116# | | |
| C | 29 | 41 | 58 | 117# | | | | | |
| CEX | 81 | 127# | | | | | | | |
| CM10 | 160# | 202 | | | | | | | |
| CR | 99# | 140 | | | | | | | |
| CRLF | 95# | 142 | | | | | | | |
| C10 | 45 | 125# | | | | | | | |
| C100 | 161# | 206 | | | | | | | |
| C6064 | 54 | 129# | | | | | | | |
| C7 | 22 | 68 | 73 | 124# | | | | | |
| C77 | 17 | 126# | | | | | | | |
| DLIST | 136# | 162 | | | | | | | |
| DSP | 163# | 200 | | | | | | | |
| DSPY | 10# | 15 | 49 | | | | | | |
| DSPYI | 15# | 146 | 148 | 163 | | | | | |
| EXCUTE | 165# | 216 | | | | | | | |
| H | 36 | 37 | 39 | 40 | 56 | 118# | | | |
| LF | 107# | 138 | | | | | | | |
| LOCATE | 6# | 8 | | | | | | | |
| MARGIN | 99 | 120# | | | | | | | |
| MASKS | 28 | 134# | | | | | | | |
| MASK0 | 134 | 224# | | | | | | | |
| MASK1 | 230# | | | | | | | | |
| MASK2 | 236# | | | | | | | | |
| MASK3 | 242# | | | | | | | | |
| MASK4 | 248# | | | | | | | | |
| MASK5 | 256# | | | | | | | | |
| MASK6 | 262# | | | | | | | | |
| M10 | 101 | 123# | | | | | | | |
| M14 | 95 | 107 | 122# | | | | | | |
| M240 | 13 | 105 | 121# | | | | | | |
| M420 | 159# | 204 | | | | | | | |
| OUTCHM | 17# | 106 | | | | | | | |
| RESET | 92# | 144 | | | | | | | |

| | | | | | | | | | |
|--------|------|------|-----|-----|-----|------|------|------|-----|
| SAR | 133 | 268# | | | | | | | |
| SAR1 | 19 | 133# | | | | | | | |
| SAVE1 | 130 | 169# | 172 | 181 | 205 | 208 | 209 | 212 | 215 |
| SAVE2 | 12 | 16 | 64 | 65 | 72 | 130# | | | |
| SCALE | 46 | 69 | 74 | 96 | 102 | 108 | 132# | | |
| SCALE0 | 156 | 198# | | | | | | | |
| SCALE1 | 154 | 199# | | | | | | | |
| SCF | 167# | 190 | 198 | 199 | | | | | |
| SCT | 168# | 189 | 193 | | | | | | |
| SEARCH | 131 | 171# | 178 | 185 | 186 | | | | |
| SRCH1 | 11 | 131# | | | | | | | |
| SRCH2 | 175# | 183 | | | | | | | |
| SRCH3 | 162# | 173 | | | | | | | |
| SYNC | 101# | 111 | 152 | | | | | | |
| TAB | 150 | 202# | | | | | | | |
| TAB1 | 206# | 214 | | | | | | | |
| TOP | 92 | 128# | | | | | | | |
| VA | 37# | 53 | 83 | | | | | | |
| VOUT | 71 | 76 | 82 | 85# | 91 | | | | |
| VSCALE | 132 | 188# | 196 | 203 | 207 | | | | |
| VSTATE | 55 | 88# | 89 | | | | | | |
| VX0 | 164# | 210 | | | | | | | |
| WSP | 174 | 175 | 176 | 180 | 184 | 192 | 194 | 254# | |
| XCUTE | 104# | 165 | | | | | | | |
| X0 | 47 | 48 | 70 | 100 | 103 | 104 | 113# | 164 | |
| Y0 | 75 | 94 | 97 | 98 | 109 | 110 | 114# | | |

8.3. A character generator using dots

```

1          XLIST /OPDEF
2          /DOTS
3          /CHARACTER GENERATOR WITH 5*7 GRID
4          /A,E,BROUWER, MC, A'DAM, 040673
5
6          BR=5
7          HO=7
8          DOTS, 0
9          0200 0000 SNA
10         0201 7450 JMP I DOTS /SKIP BLANKS
11         0202 5600 TAD (-215
12         0203 1377 SNA
13         0204 7450 JMP CR /CARRET
14         0205 5350 IAC
15         0206 7001 SNA
16         0207 7450 JMP FF /FORM FEED
17         0210 5344 TAD (2
18         0211 1376 SNA
19         0212 7450 JMP LF /LINE FEED
20         0213 5353 IAC
21         0214 7001 SNA
22         0215 7450 JMP TAB /TABULATION
23         0216 5357 TAD (211-340
24         0217 1375 CLL
25         0220 7100 TAD C100
26         0221 1270 SNL
27         0222 7420 JMP ,+3 /IGNORE ALL OTHER CONTROL CHARS
28         0223 5226 TAD C240 /RESTORE CHAR
29         0224 1277 JMS DSPLY /DISPLAY IT
30         0225 4230 CLA
31         0226 7200 JMP I DOTS
32         0227 5600
33         0230 0000 DSPLY, 0 /IN WITH 8-BIT ASCII
34         0231 0374 AND (77 /MASK 6 BITS

```

```

35 0232 3336      DCA    PTR
36 0233 1336      TAD    PTR
37 0234 7104      CLL   RAL
38 0235 1336      TAD    PTR      /3*PTR
39 0236 1373      TAD    (TABLE
40 0237 3336      DCA    PTR
41 0240 1736      TAD    PTR      /GET THE CONTROL CODE
42 0241 3341      DCA    WORD1    /INTO THE TRIPLE LENGTH WORD
43 0242 2336      ISZ   PTR
44 0243 1736      TAD    PTR
45 0244 3342      DCA    WORD2
46 0245 2336      ISZ   PTR
47 0246 1736      TAD    PTR
48 0247 3343      DCA    WORD3
49 0250 1372      TAD    (-BR
50 0251 3337      DCA    CTR1     /COUNTS COLUMNS
51 0252 4261      JMS   COL      /DISPLAY A COLUMN
52 0253 3337      ISZ   CTR1     /ALL COLUMNS DONE?
53 0254 5252      JMP   , -2     /NO
54 0255 7305      CLA   CLL IAC RAL /AC4
55 0256 1334      TAD    X0
56 0257 3334      DCA    X0      /INCREMENT X-COORDINATE
57 0260 5630      JMP    DSPLY   /EXIT
58
59 0261 0000      COL,   0
60 0262 1371      TAD    (-HO
61 0263 3340      DCA    CTR2     /COUNTS ROWS
62 0264 4321      COL1,  JMS    ROTBIT /ROTATE TRIPLE LENGTH WORD ONE LEFT
63 0265 7430      SZL
64 0266 4301      JMS   DOT     /YES, DISPLAY A POINT
65 0267 2335      ISZ   Y0      /INCREMENT Y-COORDINATE
66 0270 0100      C100, 100     /PREVENT SKIPPING
67 0271 2340      ISZ   CTR2     /ALL ROWS DONE?
68 0272 5264      JMP   COL1     /NO
69 0273 1371      TAD    (-HO
70 0274 1335      TAD    Y0
71 0275 3335      DCA    Y0      /RESET Y-COORDINATE
72 0276 2334      ISZ   X0      /INCREMENT X-COORDINATE
73 0277 0240      C240, 240     /PREVENT SKIPPING
74 0300 5661      JMP    COL
75
76 0301 0000      DOT,   0      /DISPLAY A POINT
77 0302 1334      TAD    X0
78 0303 6064      LDY
79 0304 4314      WAIT
80 0305 1335      TAD    Y0
81 0306 6065      LDY
82 0307 4314      WAIT
83 0310 1370      TAD    (1001  /VISIBLE POINT
84 0311 6066      EXC
85 0312 4314      WAIT
86 0313 5701      JMP    DOT
87
88      WAIT=JMS .
89 0314 0000      0;SRF;JMP , -1;CLA;JMP I , -4
90 0315 6071
91 0316 5315
92 0317 7200
93 0320 5714
94
95 0321 0000      ROTBIT, 0
96 0322 1343      TAD    WORD3
97 0323 7104      RAL   CLL
98 0324 3343      DCA    WORD3
99 0325 1342      TAD    WORD2
100 0326 7004      RAL
101 0327 3342      DCA    WORD2
102 0330 1341      TAD    WORD1
103 0331 7004      RAL
104 0332 3341      DCA    WORD1

```

```

105 0333 5721          JMP I   ROTBIT
106 0334 7400  X0,    -400
107 0335 0500  Y0,    500
108 0336 0000  PTR,    0
109 0337 0000  CTR1,   0
110 0340 0000  CTR2,   0
111 0341 0000  WORD1,  0
112 0342 0000  WORD2,  0
113 0343 0000  WORD3,  0
114
115 0344 1367  FF,     TAD     (500
116 0345 3335          DCA     Y0      /RESET COORDINATES
117 0346 6072          CRF          /CLEAR READY FLAG
118 0347 4314          WAIT        /AND WAIT FOR ERASE
119 0350 1366  CR,     TAD     (-400
120 0351 3334          DCA     X0      /RESET X-COORDINATE
121 0352 5600          JMP I   DOTS
122 0353 1365  LF,     TAD     (-12
123 0354 1335          TAD     Y0
124 0355 3335          DCA     Y0      /DECREMENT Y-COORDINATE
125 0356 5600          JMP I   DOTS
126 0357 1334  TAB,    TAD     X0
127 0360 0364          AND     (7700
128 0361 1270          TAD     C100   /RATHER LARGE TAB
129 0362 5351          JMP     CR+1
130
131 0364 7700
132 0365 7766
133 0366 7400
134 0367 0500
135 0370 1001
136 0371 7771
137 0372 7773
138 0373 0400
139 0374 0077
140 0375 7651
141 0376 0002
142 0377 7563
143          PAGE
144
145          TABLE=.
146 0400 3122 3122;3714;0574;/°
147 0401 3714
148 0402 0574
149 0403 7604 7604;4211;1370;/A
150 0404 4211
151 0405 1370
152 0406 4077 4077;7114;4554;/B
153 0407 7114
154 0410 4554
155 0411 3720 3720;3014;0504;/C
156 0412 3014
157 0413 0504
158 0414 4077 4077;7014;0574;/D
159 0415 7014
160 0416 0574
161 0417 7762 7762;3114;0602;/E
162 0420 3114
163 0421 0602
164 0422 7742 7742;2110;0402;/F
165 0423 2110
166 0424 0402

```

| | | | |
|-----|------|------|-------------------|
| 167 | 0425 | 3720 | 3720;3014;4762;/G |
| 168 | 0426 | 3014 | |
| 169 | 0427 | 4762 | |
| 170 | 0430 | 7742 | 7742;0100;4376;/H |
| 171 | 0431 | 0100 | |
| 172 | 0432 | 4376 | |
| 173 | 0433 | 0020 | 0020;3774;0400;/I |
| 174 | 0434 | 3774 | |
| 175 | 0435 | 0400 | |
| 176 | 0436 | 2020 | 2020;1004;0176;/J |
| 177 | 0437 | 1004 | |
| 178 | 0440 | 0176 | |
| 179 | 0441 | 7742 | 7742;0242;1202;/K |
| 180 | 0442 | 0242 | |
| 181 | 0443 | 1202 | |
| 182 | 0444 | 7760 | 7760;1004;0200;/L |
| 183 | 0445 | 1004 | |
| 184 | 0446 | 0200 | |
| 185 | 0447 | 7740 | 7740;4140;1376;/M |
| 186 | 0450 | 4140 | |
| 187 | 0451 | 1376 | |
| 188 | 0452 | 7740 | 7740;4040;4376;/N |
| 189 | 0453 | 4040 | |
| 190 | 0454 | 4376 | |
| 191 | 0455 | 7760 | 7760;3014;0776;/O |
| 192 | 0456 | 3014 | |
| 193 | 0457 | 0776 | |
| 194 | 0460 | 7742 | 7742;2110;4414;/P |
| 195 | 0461 | 2110 | |
| 196 | 0462 | 4414 | |
| 197 | 0463 | 3720 | 3720;3212;0674;/Q |
| 198 | 0464 | 3212 | |
| 199 | 0465 | 0674 | |
| 200 | 0466 | 7742 | 7742;2312;4614;/R |
| 201 | 0467 | 2312 | |
| 202 | 0470 | 4614 | |
| 203 | 0471 | 2121 | 2121;3115;0504;/S |
| 204 | 0472 | 3115 | |
| 205 | 0473 | 0504 | |
| 206 | 0474 | 0040 | 0040;3770;0402;/T |
| 207 | 0475 | 3770 | |
| 208 | 0476 | 0402 | |
| 209 | 0477 | 3760 | 3760;1004;0176;/U |
| 210 | 0500 | 1004 | |
| 211 | 0501 | 0176 | |
| 212 | 0502 | 0346 | 0346;1401;4016;/V |
| 213 | 0503 | 1401 | |
| 214 | 0504 | 4016 | |
| 215 | 0505 | 7750 | 7750;0202;0376;/W |
| 216 | 0506 | 0202 | |
| 217 | 0507 | 0376 | |
| 218 | 0510 | 6145 | 6145;0101;2306;/X |
| 219 | 0511 | 0101 | |
| 220 | 0512 | 2306 | |
| 221 | 0513 | 0141 | 0141;1700;2006;/Y |
| 222 | 0514 | 1700 | |
| 223 | 0515 | 2006 | |
| 224 | 0516 | 6064 | 6064;3114;2606;/Z |
| 225 | 0517 | 3114 | |
| 226 | 0520 | 2606 | |
| 227 | 0521 | 0037 | 0037;7014;0400;/[|
| 228 | 0522 | 7014 | |
| 229 | 0523 | 0400 | |
| 230 | 0524 | 0101 | 0101;0101;0100;/\ |
| 231 | 0525 | 0101 | |
| 232 | 0526 | 0100 | |
| 233 | 0527 | 0020 | 0020;3017;7400;/] |
| 234 | 0530 | 3017 | |

| | | | |
|-----|------|------|-----------------------|
| 235 | 0531 | 7400 | |
| 236 | 0532 | 0200 | 0200;5770;1010;/† |
| 237 | 0533 | 5770 | |
| 238 | 0534 | 1010 | |
| 239 | 0535 | 0407 | 0407;0100;4020;/° |
| 240 | 0536 | 0100 | |
| 241 | 0537 | 4020 | |
| 242 | 0540 | 0000 | 0000;0000;0000;/SPACE |
| 243 | 0541 | 0000 | |
| 244 | 0542 | 0000 | |
| 245 | 0543 | 0000 | 0000;1370;0000;/° |
| 246 | 0544 | 1370 | |
| 247 | 0545 | 0000 | |
| 248 | 0546 | 0001 | 0001;6000;3400;/" |
| 249 | 0547 | 6000 | |
| 250 | 0550 | 3400 | |
| 251 | 0551 | 1235 | 1235;6007;3450;/# |
| 252 | 0552 | 6007 | |
| 253 | 0553 | 3450 | |
| 254 | 0554 | 2212 | 2212;5532;5044;/§ |
| 255 | 0555 | 5532 | |
| 256 | 0556 | 5044 | |
| 257 | 0557 | 6144 | 6144;6106;2306;/‡ |
| 258 | 0560 | 6106 | |
| 259 | 0561 | 2306 | |
| 260 | 0562 | 3322 | 3322;3262;0040;/^ |
| 261 | 0563 | 3262 | |
| 262 | 0564 | 0040 | |
| 263 | 0565 | 0000 | 0000;0070;0000;/' |
| 264 | 0566 | 0070 | |
| 265 | 0567 | 0000 | |
| 266 | 0570 | 0007 | 0007;0424;0400;/((|
| 267 | 0571 | 0424 | |
| 268 | 0572 | 0400 | |
| 269 | 0573 | 0020 | 0020;2421;6000;/) |
| 270 | 0574 | 2421 | |
| 271 | 0575 | 6000 | |
| 272 | 0576 | 2507 | 2507;0761;6124;/* |
| 273 | 0577 | 0761 | |
| 274 | 0600 | 6124 | |
| 275 | 0601 | 0402 | 0402;0760;4020;/+ |
| 276 | 0602 | 0760 | |
| 277 | 0603 | 4020 | |
| 278 | 0604 | 0026 | 0026;0700;0000;/, |
| 279 | 0605 | 0700 | |
| 280 | 0606 | 0000 | |
| 281 | 0607 | 0402 | 0402;0100;4020;/- |
| 282 | 0610 | 0100 | |
| 283 | 0611 | 4020 | |
| 284 | 0612 | 0030 | 0030;1400;0000;/. |
| 285 | 0613 | 1400 | |
| 286 | 0614 | 0000 | |
| 287 | 0615 | 2004 | 2004;0100;2004;/// |
| 288 | 0616 | 0100 | |
| 289 | 0617 | 2004 | |
| 290 | 0620 | 3724 | 3724;3114;2574;/0 |
| 291 | 0621 | 3114 | |
| 292 | 0622 | 2574 | |
| 293 | 0623 | 0020 | 0020;5774;0000;/1 |
| 294 | 0624 | 5774 | |
| 295 | 0625 | 0000 | |
| 296 | 0626 | 7122 | 7122;3114;4614;/2 |
| 297 | 0627 | 3114 | |
| 298 | 0630 | 4614 | |
| 299 | 0631 | 2120 | 2120;3114;4554;/3 |
| 300 | 0632 | 3114 | |
| 301 | 0633 | 4554 | |
| 302 | 0634 | 1405 | 1405;0227;7440;/4 |
| 303 | 0635 | 0227 | |
| 304 | 0636 | 7440 | |

| | | | | | | | | |
|--------|-----|-----|-----|------|------|-----|-----|-----|
| WAIT | 79 | 82 | 85 | 88# | 118 | | | |
| WORD1 | 42 | 102 | 104 | 111# | | | | |
| WORD2 | 45 | 99 | 101 | 112# | | | | |
| WORD3 | 48 | 96 | 98 | 113# | | | | |
| X0 | 55 | 56 | 72 | 77 | 106# | 120 | 126 | |
| Y0 | 65 | 70 | 71 | 80 | 107# | 116 | 123 | 124 |
| °L0364 | 127 | | | | | | | |
| °L0365 | 122 | | | | | | | |
| °L0366 | 119 | | | | | | | |
| °L0367 | 115 | | | | | | | |
| °L0370 | 83 | | | | | | | |
| °L0371 | 60 | 69 | | | | | | |
| °L0372 | 49 | | | | | | | |
| °L0373 | 39 | | | | | | | |
| °L0374 | 34 | | | | | | | |
| °L0375 | 23 | | | | | | | |
| °L0376 | 17 | | | | | | | |
| °L0377 | 11 | | | | | | | |

The three character generators from sections 8.1 through 8.3 can be compared on the photograph below.

