HIGH-RESOLUTION
COLOR GRAPHICS
ON THE
APPLE-II COMPUTER

S. Wozniak (WOZ)
APPLE COMPUTER, INC.
November 30, 1977
APPLE-II HI-RES GRAPHICS SUBROUTINES

The APPLE-II computer comes with a high-resolution (hereafter 'HI-RES') color graphics display mode of 280 horizontal by 192 vertical resolution. Because 8K bytes of RAM are dedicated solely to maintaining the HI-RES display, a minimum 12K byte system (configured for HI-RES) is required to use this mode. For practical reasons, 16K bytes is the strongly recommended minimum. A 6502 machine language subroutine package has been developed to simplify efficient use of the APPLE-II HI-RES display for assembly language and BASIC programmers. The routines for initializing the HI-RES display, plotting points, drawing lines, and drawing shapes are described herein.
USING THE HI-RES SUBROUTINES

Despite the fact that HI-RES graphics commands are not built into APPLE-II BASIC, a convenient scheme for referencing the subroutines and their parameters by name has been devised, as illustrated below.

<table>
<thead>
<tr>
<th>TRADITIONAL METHOD</th>
<th>IMPROVED METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRADITIONAL METHOD</td>
<td>IMPROVED METHOD</td>
</tr>
<tr>
<td>OF CALLING</td>
<td></td>
</tr>
<tr>
<td>MACHINE LANGUAGE SUBROUTINES</td>
<td></td>
</tr>
<tr>
<td>&gt; POKE 800, X MOD 256</td>
<td>&gt; X0 = X</td>
</tr>
<tr>
<td>&gt; POKE 801, X / 256</td>
<td>&gt; Y0 = Y</td>
</tr>
<tr>
<td>&gt; POKE 802, Y</td>
<td>&gt; COLR = C</td>
</tr>
<tr>
<td>&gt; POKE 812, C (color)</td>
<td>&gt; CALL PLOT</td>
</tr>
<tr>
<td>&gt; CALL 2834</td>
<td></td>
</tr>
</tbody>
</table>

The first statement of a program using the HI-RES subroutines should be as follows

```
0   X0 = Y0 = COLR = SHAPE = ROT = SCALE
```

The purposes of this statement are to define a line number 0 (necessary when later appending the HI-RES PREFIX program) and to enter the first 6 BASIC variable names in the symbol table in a fixed sequence. When executed, each of the 6 parameters will be assigned storage at fixed locations relative to the address contained in the BASIC 'start of variables' pointer, LOMEM, making them readily accessible by the HI-RES subroutines.
Different parameter names may be used provided that they retain the same number of characters. This is necessary to insure that the storage locations for each relative to LOMEM do not change. For example, the name XX could be used in place of X0 but XCOORD could not.

The parameters SHAPE, ROT, and SCALE are used only by the HI-RES shape draw subroutines and may be ommitted from programs using only the PLOT and LINE features. Ommitting unnecessary variable definitions is one method of enhancing the overall performance of some BASIC programs on the APPLE-II and is thus desirable.

FIRST LINE OF PROGRAMS NOT USING
THE SHAPE DRAW SUBROUTINES

0 X0 = Y0 = COLR

After the parameter names have been defined, the HI-RES subroutine names themselves may be defined and assigned corresponding subroutine entry addresses as values. Calling subroutines by name is preferable to calling them by entry address because the entry addresses may vary in future versions of the HI-RES subroutines, and names are better self documenting.
<table>
<thead>
<tr>
<th>Absolute CALL</th>
<th>CALL by name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 INIT = 2048</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 CALL 2048</td>
</tr>
<tr>
<td></td>
<td>100 CALL INIT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 CALL 2048</td>
</tr>
<tr>
<td></td>
<td>200 CALL INIT</td>
</tr>
</tbody>
</table>

In the above CALL by name example, should the INIT subroutine entry address change to -12288, only line 5 need be changed. In the absolute CALL example, lines 100 and 200 (and any others referencing the INIT subroutine) will have to be changed. The self documenting advantage of the CALL by name example should be apparent.

The following statement lists all HI-RES subroutine entry initializations available to BASIC programs. Other names may be used at the programmer's discretion.

```
5 INIT = 2048 : CLEAR = 2062 : BKGNL = 2865 :
POSN = 2809 : PLOT = 2830 : LINE = 2836 :
DRAW = 2871 : DRAW1 = 2874 : XDRAW = 2884 :
XDRAW1 = 2887 : FIND = 2556
```
The allowable color specification values may also be referenced by name, if the initialization statement below is included in your program. Note that 'GREEN' is preceeded by 'LET' to avoid a syntax error due to confusion with the GR command.

7  BLACK = 0 : LET GREEN = 42 : VIOLET = 85 : WHITE = 127

If your APPLE-II has been modified for additional HI-RES colors, the following assignments are also valid.

8  ORANGE = 170 : BLUE = 213 : BLACK2 = 128 : WHITE2 = 255

Unnecessary variable definitions should be avoided as they will slow some programs. Therefore, a program should not define VIOLET = 85 unless it uses the color VIOLET. The example below illustrates condensed initialization statements for a program using only the INIT, PLOT, and DRAW subroutines, and the colors GREEN and WHITE.

0  XO = Y0 = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : PLOT = 2830 : DRAW = 2871
7  LET GREEN = 42 : WHITE = 127
In extreme cases any of the following techniques will further enhance program performance.

(1) Omit the color and subroutine name initializations. Refer to colors and subroutines by value, not name. This does not apply to the parameter references.

(2) Define the most frequently used program variable names prior to the subroutine name and color name initializations (lines 5 and 7 in the prior examples). The example below will speed up programs extensively referencing variables I, J, and K.

0 X0 = Y0 = COLR = SHAPE = ROT = SCALE
2 I = J = K
5 INIT = 2048 : CLEAR = 2062 : BKGND = 2865 :
   POSN = 2809 ........... etc.
7 BLACK = 0 : LET GREEN = 42 : ........... etc.

(3) Use the parameter names as program variables when possible. Because they are defined first, the parameters are the most quickly accessed BASIC variables.
INITIALIZATION SUBROUTINES

The normal HI-RES display consists of a 280 horizontal by 160 vertical grid above 4 lines of text and is initiated with the BASIC command below.

> CALL INIT

The INIT subroutine also clears the HI-RES display and initializes other HI-RES subroutines. After calling INIT the programmer may eliminate the 4 line text display, extending the HI-RES display to a 192 vertical resolution, with the following command:

> POKE -16302,0

The 4-line text display may be restored at any time as follows:

> POKE -16301,0

Valid X-coordinates vary from 0 (leftmost) to 279 (rightmost) Valid Y-coordinates vary from 0 (topmost) to 159 or 191 (bottommost) depending on whether or not the 4 line text display is enabled.

At any time after INIT has been called, the entire HI-RES display may be cleared with the CLEAR subroutine as shown below.

> CALL CLEAR
The HI-RES display may be quickly set to any background color with the BKGND subroutine. BKGND expects a color specification in the BASIC variable COLR. The example below turns the entire HI-RES display green.

```
0  X0 = Y0 = COLR
5  INIT = 2048 : BKGND = 2865 :
       LET GREEN = 42
10  CALL INIT
20  COLR = GREEN
30  CALL BKGND
40  END
```

Only the colors previously mentioned (BLACK, GREEN, VIOLET, and WHITE) may be specified in COLR. Do not make up your own. For example, COLR = YELLOW is not allowed.

If COLR is greater than 255 when BKGND is called then a range error will occur. The message '(beep) *** RANGE ERR' will be displayed and the program will halt.
POINTS AND LINES

The PLOT subroutine is used to plot a single point of the HI-RES display in a specified color. COLR must be less than 255, X0 must be 0 to 279, and Y0 must be 0 to 191 when PLOT is called or a range error will result and the program will halt. The program below plots one white dot at X-coordinate 35, Y-coordinate 55 (35,55) and one at (85,90).

0  X0 = Y0 = COLR
5  INIT = 2048 : PLOT = 2380 : WHITE = 127
10  CALL INIT
20  COLR = WHITE
30  X0 = 35 : Y0 = 55 : CALL PLOT
40  X0 = 85 : Y0 = 90 : CALL PLOT
50  END

Connecting any two coordinates with a straight line is almost as easy as plotting points. After plotting one endpoint as shown in the example above, the other endpoint is specified in X0 and Y0 and the the LINE subroutine is called. As with the PLOT subroutine, COLR must be less than 256, X0 must be 0 to 279, and Y0 must be 0 to 191 or a range error will result and the program will halt. The following example draws a white line from (35,40) to (170,100), a green line from (270,10) to (5,145), and a violet line from (20,70) to (190,110).
0 \ X0 = Y0 = COLR
5 \ INIT = 2048 : PLOT = 2830 : LINE = 2836 :
   LET GREEN = 42 : VIOLET = 85 : WHITE = 127
10 CALL INIT
20 COLR = WHITE : X0 = 35 : Y0 = 40 : CALL PLOT
25 X0 = 170 : Y0 = 100 : CALL LINE
30 COLR = GREEN : X0 = 270 : Y0 = 10 : CALL PLOT
35 X0 = 5 : Y0 = 145 : CALL LINE
40 COLR = VIOLET : X0 = 20 : Y0 = 70 : CALL PLOT
45 X0 = 190 : Y0 = 110 : CALL LINE
50 END

The following example illustrates that the parameter variables may be used as FOR loop indices. Horizontal violet lines are drawn on a green background at every tenth vertical coordinate.

0 \ X0 = Y0 = COLR
5 \ INIT = 2048 : BKGD = 2865 : PLOT = 2830 :
   LINE = 2836 : LET GREEN = 42 : VIOLET = 85
10 CALL INIT
20 COLR = GREEN : CALL BKGD
30 COLR = VIOLET
40 FOR Y0 = 5 TO 155 STEP 10
50 X0 = 10 : CALL PLOT : X0 = 270 : CALL LINE
60 NEXT Y0 : END
Multiple lines which are connected endpoint to endpoint may be drawn without intervening PLOT calls. In the example below, a white line connects (10,20) to (250,70), and green line connects (250,70) to (20,150), and a violet line connects (20,150) to (260,30).

```
0  XO = YO = COLR
5  INIT = 2048 : PLOT = 2830 : LINE = 2836 :
   LET GREEN = 42 : VIOLET = 85 : WHITE = 127
10 CALL INIT
20  COLR = WHITE : XO = 10 : YO = 20 : CALL PLOT
30  XO = 250 : YO = 70 : CALL LINE
40  XO = 20 : YO = 150 : COLR = GREEN : CALL LINE
50  XO = 260 : YO = 30 : COLR = VIOLET : CALL LINE
60  END
```

CAUTION

Do not attempt to draw a line prior to the first PLOT. Because the first endpoint has not been defined, the line may be drawn in random memory locations, not necessarily restricted to the screen memory.
DRAWING SHAPES

Up to 255 different shapes may be defined, edited, and saved on a single tape.
After loading the HI-RES subroutines such a 'shape tape' (containing a 'shape table') may be read as follows.

1. Position shape tape in recorder.
2. Load shape tape with the BASIC command:
   > CALL 3001
3. Start recorder (PLAY).
The above command immediately begins reading tape.
4. Wait for two beeps.

Shape tables always load beginning at address $C000 with the HI-RES subroutines in locations $800-$BFF. Upon loading a shape table, the BASIC 'start of variables' pointer LOMEM is set to contain the address of the location immediately following the last shape table byte.

If not enough free memory is available to contain the shape table then the message '(beep) *** MEM FULL ERR' will be displayed.
If no beep is heard when loading a shape tape then something is probably wrong with the tape connection and you will have to hit RESET and type CC (Control-C) to reenter BASIC. If you hear a single beep and then the system hangs it means your shape tape is probably bad and after hitting RESET and typing CC you may have to restore the LOMEM setting to $C000 ($3072) as follows.

> LOMEM : 3072
The DRAW subroutine is used to display any of the predefined shapes included in the current shape table. The origin or 'beginning point' of the shape is specified in X0 and Y0 and the color is specified in COLR as with PLOT. The shape number desired is specified in SHAPE. For example, SHAPE = 3 specifies that the third shape of the current shape table is to be drawn. A scale factor is specified in the variable SCALE and a rotation in ROT. A scale factor of 4 implies a shape 4 times the defined size. A scale factor of 0 is always interpreted as 256.

Rotations

<table>
<thead>
<tr>
<th>ROT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROT=0</td>
<td>(no rotation)</td>
</tr>
<tr>
<td>ROT=48</td>
<td>(270 deg. CW)</td>
</tr>
<tr>
<td>ROT=16</td>
<td>(90 deg. CW)</td>
</tr>
<tr>
<td>ROT=32</td>
<td>(180 deg. CW)</td>
</tr>
</tbody>
</table>

COLR must be 0 to 255, X0 must be 0 to 279, Y0 must be 0 to 191, ROT must be 0 to 255 (due to MOD 64 arithmetic, ROT=64 is equivalent to ROT=0), SCALE must be 0 to 255, and SHAPE must be greater than zero and less than or equal to the current number of shape table shapes or else a range error will result when DRAW is called and the program will halt. In other words, the programmer will always be notified if HI-RES subroutines are called with any invalid parameters.
The following program example draws shape number 3 in white at a 90 degree clockwise rotation and scale factor of 2. The origin is at (140,80). It is assumed that a shape table with at least 3 shape definitions has been loaded.

```
0  XO = YO = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : DRAW = 2871
7  WHITE = 127
10 CALL INIT
20 XO = 140 : YO = 80 : COLR = WHITE
30 SHAPE = 3 : ROT = 16 : SCALE = 2
40 CALL DRAW
50 END
```

The XDRAW subroutine is identical in operation to the DRAW subroutine except that the defined shape is exclusive-ored (EX-OR'd) onto the screen. The EX-OR operation complements all screen memory bits of the shape, 0's become 1's and vice-versa. No color need be specified. A unique property of XDRAW is that 2 successive calls with identical parameters will first cause a shape to be drawn (in white) and then erased. The following program example causes the rotation of shape number 3 to track paddle 0. XDRAW is used for both the draw and erase operations. Although the color is not optional, the variable COLR may not be ommitted from the parameter declarations (line 0) or the SHAPE, ROT, and SCALE parameters will not be assigned storage in their standard locations relative to LOMEM.
0 \( X_0 = Y_0 = \text{COLR} = \text{SHAPE} = \text{ROT} = \text{SCALE} \)
5 \( \text{INIT} = 2048 : \text{XDRAW} = 2884 \)
10 CALL INIT
20 \( X_0 = 140 : Y_0 = 80 : \text{SHAPE} = 3 : \text{SCALE} = 2 \)
30 \( R = 0 : \text{GOTO 60} : \text{REM DRAW FIRST SHAPE} \)
40 \( R = \text{PDL}(0) : \text{IF R = ROT THEN GOTO 30} \)
50 CALL XDRAW : \text{REM ERASE AT OLD ROT} \)
60 \( \text{ROT} = R : \text{CALL XDRAW} : \text{REM DRAW AT NEW ROT} \)
70 \( \text{GOTO 40} : \text{REM CHECK FOR ROT CHANGE} \)
80 END

DRAW1 and XDRAW1 are identical to DRAW and XDRAW respectively except that the most recently plotted (or drawn) point serves as the shape origin and the current color is not updated. Thus \( X_0, Y_0, \) and COLR are not specified.

If you draw a shape and then wish to draw a line from the final plot position of that shape to a fixed coordinate, you may do so. After drawing the shape, however, you must call FIND prior to calling LINE. The FIND subroutine determines the X-Y coordinates of the final shape plot position (or current plot position if used after other subroutines) and uses it as the beginning endpoint of the following call to LINE. The following program example draws a shape and then a violet line from the final plot position of the shape to \((10,25)\).
COLLISIONS

Overlapping shapes define points of 'collision'. The DRAW and XDRAW subroutines return a collision count in the absolute location $32A$ (810 decimal). The collision count will be constant for a fixed shape, rotation, scale, and background, provided that no collisions with other shapes are detected. The difference between the 'standard' collision value and the encountered value (while drawing a shape) is a true collision indicator.

100 CALL DRAW

110 COUNT = PEEK (810)
APPENDING THE HI-RES PREFIX

The HI-RES PREFIX program may be permanently appended to any BASIC programs you write, making a 2-step LOAD unnecessary. If you have the APPLE-II RENUMBER/APPEND program then treat the user-written program as the one with greater line numbers (despite the fact that it begins with line 0) and the HI-RES PREFIX program as the one with smaller line numbers. If you don't have the RENUMBER/APPEND program then the APPEND may done manually as follows:

1. > LOAD (user program)
2. > POKE 0, PEEK (76)
   > POKE 1, PEEK (77)
   > POKE 76, PEEK (202)
   > POKE 77, PEEK (203)
   (user program is now hidden)
3. > LOAD (HI-RES PREFIX program)
4. > POKE 76, PEEK (0)
   > POKE 77, PEEK (1)
5. > SAVE (combined program)
## SUMMARY

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Calling address</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td>2062</td>
<td></td>
</tr>
<tr>
<td>BKGND</td>
<td>2865</td>
<td>COLR</td>
</tr>
<tr>
<td>POSN</td>
<td>2809</td>
<td>XO, YO, COLR</td>
</tr>
<tr>
<td>PLOT</td>
<td>2830</td>
<td>XO, YO, COLR</td>
</tr>
<tr>
<td>LINE</td>
<td>2836</td>
<td>XO, YO, COLR</td>
</tr>
<tr>
<td>DRAW</td>
<td>2871</td>
<td>XO, YO, COLR, SHAPE, ROT, SCALE</td>
</tr>
<tr>
<td>DRAW1</td>
<td>2874</td>
<td>SHAPE, ROT, SCALE</td>
</tr>
<tr>
<td>XDRAW</td>
<td>2884</td>
<td>XO, YO, COLR, SHAPE, ROT, SCALE</td>
</tr>
<tr>
<td>XDRAW1</td>
<td>2887</td>
<td>SHAPE, ROT, SCALE</td>
</tr>
<tr>
<td>FIND</td>
<td>2556</td>
<td></td>
</tr>
<tr>
<td>SHAPE LOAD</td>
<td>3001</td>
<td></td>
</tr>
</tbody>
</table>

For NO TEXT display  

>POKE -16302,0

For mixed GRAPHICS/TEXT  

>POKE -16301,0

Select secondary screen display  

>POKE -16299,0

Select primary screen display  

>POKE -16300,0

Select secondary screen plotting  

>POKE 806,64

Select primary screen plotting  

>POKE 806,32

(Defaults are GRAPHICS/TEXT, primary screen display, and primary screen plotting)

Collision detect (shape draw only)  

PEEK (S10)
HI-RES PREFIX LOAD

AFTER LOAD

PV, LOMEM (normally $800)

PP (start of program)

HI-MEM (end of program)

$800

HI-RES SUBR'S

PV, LOMEM ($CO0)

$BFF

USER AREA

BASIC (1 page)

HI-RES SUBR'S (4 pages)

BASIC (1 page)

USER PROG (if any)

USER AREA

PP

USER PROG (execution initiated at line 0)

HI-MEM

Note: A 'page' is 256 bytes.

APPLE-II BASIC POINTERS

LOMEM (in $4A, $4B)---------Contains 'start of BASIC variables' address.

PV (SCC, SCD)--------------End of BASIC variables. Equal to LOMEM if no active variables.

PP (SCA, SCB)--------------Start of BASIC program. Equal to HI-MEM if no program.

HI-MEM ($4C, $4D)----------End of BASIC pv.
HI-RES PARAMETER LOCATIONS
(beyond LOMEM)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Locations beyond LOMEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>XO</td>
<td>$05, $06</td>
</tr>
<tr>
<td>YO</td>
<td>$0C, $0D</td>
</tr>
<tr>
<td>COLR</td>
<td>$15, $16</td>
</tr>
<tr>
<td>SHAPE</td>
<td>$1F, $20</td>
</tr>
<tr>
<td>ROT</td>
<td>$27, $28</td>
</tr>
<tr>
<td>SCALE</td>
<td>$31, $32</td>
</tr>
</tbody>
</table>

Note: Each parameter is two bytes in length. The low-order byte is stored in the lesser of the two locations assigned.

HI-RES SUBROUTINES SEGMENT MAP

- CODE $800-$9E8
- DATA $9E9-$9FB
- CODE $9FC-$BFF
SHAPE TAPE

Record #1 ------- Contains length of record #2. Two bytes long, low-order first.

Record Gap ------- Minimum of .7 seconds.

Record #2 ------- Shape table (see below).

```
START
(in $328, $329)

START is set to $C00 by the SHAPE LOAD subr.

0-255
Unused

n = number of shapes

index to shape definition #1 relative to START

low
high

.
.
.

low
high

SHAPE DEF #1

.
.
.

index to shape definition #n relative to START

SHAPE DEF #n

LOMEM

BASIC VARIABLES
```
### SHAPE DEFINITIONS

<table>
<thead>
<tr>
<th>first shape byte</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>final shape byte</td>
<td>$00</td>
</tr>
</tbody>
</table>

'end of shape' mark

### SHAPE BYTE

**MSB** | **LSB**
--- | ---

\[
\begin{array}{cccc}
X & Y & P & X & Y & P & X & Y \\
\end{array}
\]

Third vector  
First vector  
Second vector

### X Y Vector

- **0 0**: +
- **0 1**: +  \( P = 0 \) Move without plot
- **1 0**: +  \( P = 1 \) Plot, then move
- **1 1**: +  Third vector is move without plot

**ZEROES ARE IGNORED**—If the remaining one or two vectors of a shape byte are zeroes then they are ignored.
SAMPLE SHAPE BYTES

(plot-prior-move vectors are circled)

00 100 100  $24
00 101 101  $2D
00 110 110  $36
00 111 111  $3F

00 000 100  $04
00 000 001  $01
01 101 101  $6D
11 111 111  $FF

10 101 100  $AC
01 100 110  $66
01 101 001  $69
10 010 010  $92

INVALID SHAPE BYTES

 ignored ignored ignored ignored

ignored ignored ignored ignored ignored

too many

 ignored
 ignored ignored ignored ignored ignored

 ignored
 ignored ignored ignored ignored ignored

 ignored
 ignored ignored ignored ignored ignored

 ignored
 ignored ignored ignored ignored ignored

 ignored
1. HPAG (in location $326) contains the high-order byte of the
starting address of the current HI-RES display memory in
which plotting is being done.

   Primary screen memory plotting ------ HPAG = $20
   ($2000-$3FFF)

   Secondary screen memory plotting ---- HPAG = $40
   ($4000-$5FFF)

2. HBSAL and HBASH (in locations $26 and $27) contain the BASE
ADDRESS corresponding to the current Y-coordinate. The
BASE ADDRESS is the address of the leftmost display byte of
the current line. HBSAL and HBASH will track all plotting
and drawing 'on-the-fly'.

   Current HPAG
   \[
   \begin{array}{cccccccc}
   \text{P} & \text{Q} & \text{R} & 0 & 0 & 0 & 0 & 0 \\
   \text{MSB} & \text{LSB} \\
   \end{array}
   \]

   Current Y-Coordinate
   \[
   \begin{array}{cccccccc}
   \text{A} & \text{B} & \text{C} & \text{D} & \text{E} & \text{F} & \text{G} & \text{H} \\
   \text{MSB} & \text{LSB} \\
   \end{array}
   \]

   HBASH
   \[
   \begin{array}{cccc}
   \text{P} & \text{Q} & \text{R} & \text{F} \\
   \text{MSB} & \text{LSB} \\
   \end{array}
   \]

   HBSAL
   \[
   \begin{array}{cccc}
   \text{E} & \text{A} & \text{B} & \text{A} \\
   \text{MSB} & \text{LSB} \\
   \end{array}
   \]

3. HNDX (in location $325) contains the byte index from the
BASE ADDRESS to the current plot byte and is a function of
the current X-coordinate.

   \[\text{HNDX} = \frac{X}{7} \text{ (integer divide with truncate)}\]
4. HMASK (in location $30$) contains a bit mask corresponding to the current bit position within the current plot byte and is a function of the current X-coordinate. The high-order bit is always set.

<table>
<thead>
<tr>
<th>X MOD 7</th>
<th>HMASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (leftmost)</td>
<td>$81$</td>
</tr>
<tr>
<td>1</td>
<td>$82$</td>
</tr>
<tr>
<td>2</td>
<td>$84$</td>
</tr>
<tr>
<td>3</td>
<td>$88$</td>
</tr>
<tr>
<td>4</td>
<td>$90$</td>
</tr>
<tr>
<td>5</td>
<td>$A0$</td>
</tr>
<tr>
<td>6 (rightmost)</td>
<td>$C0$</td>
</tr>
</tbody>
</table>

5. HCOLOR (in location $1C$) is the HI-RES 'on-the-fly' color mask. The low-order seven bits are rotated one bit position for odd values of HNDX. The high-order bit selects one of two color sets on systems modified for extra HI-RES colors.

<table>
<thead>
<tr>
<th>COLOR</th>
<th>EVEN HNDX</th>
<th>ODD HNDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLACK</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>GREEN</td>
<td>0 0 1 0 1 0 1 0 1 0</td>
<td>0 1 0 1 0 1 0 1 0 1</td>
</tr>
<tr>
<td>VIOLET</td>
<td>0 1 0 1 0 1 0 1 0 1</td>
<td>0 0 1 0 1 0 1 0 1 0</td>
</tr>
<tr>
<td>WHITE</td>
<td>0 1 1 1 1 1 1 1 1 1</td>
<td>0 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>BLACK2</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>ORANGE</td>
<td>1 0 1 0 1 0 1 0 1 0</td>
<td>1 1 0 1 0 1 0 1 0 1</td>
</tr>
<tr>
<td>BLUE</td>
<td>1 1 0 1 0 1 0 1 0 1</td>
<td>1 0 1 0 1 0 1 0 1 0</td>
</tr>
<tr>
<td>WHITE2</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>
HI-RES INTERNAL VARIABLES

SHAPEL, SHAPEH ($1A, $1B) On-the-fly shape pointer.
HCOLOR1 ($1C) On-the-fly color byte.
COUNTH ($1D) High-order byte of step count for LINE.
HBASL, HBASH ($26, $27) On-the-fly BASE ADDRESS
HMASK ($30) On-the-fly BIT MASK.
QDRNT ($53) 2 LSB's are rotation quadrant for DRAW.
XOL, XOH ($320, $321) Most recent X-coordinate. Used for
initial endpoint of LINE. Updated
by PLOT, LINE, and FIND, not DRAW.
Y0 ($322) Most recent Y-coordinate (see XOL, XOH).
BXSAV ($323) Saves 6502 X-Register during HI-RES calls
from BASIC.
HCOLOR ($324) Color specification to PLOT, POSN.
HNDX ($325) On-the-fly byte index from BASE ADDRESS.
HPAG ($326) Starting page of plot memory. Normally
$20 for plotting in primary HI-RES
display memory ($2000-$3FFF).
SCALE ($327) On-the-fly scale factor for DRAW.
SHAPXL, SHAPXH ($328, $329) Start of shape table pointer.
COLLSN ($32A) Collision count from DRAW, XDRAW.
<table>
<thead>
<tr>
<th>Line</th>
<th>Label</th>
<th>Type</th>
<th>$xx</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>SHAPEL</td>
<td>EPZ</td>
<td>$1A</td>
<td>Pointer to SHAPE List</td>
</tr>
<tr>
<td>15</td>
<td>SHAPEH</td>
<td>EPZ</td>
<td>$1B</td>
<td>Running color mask</td>
</tr>
<tr>
<td>16</td>
<td>HCOLORI</td>
<td>EPZ</td>
<td>$1C</td>
<td>Base addr for current Hi-res plot line</td>
</tr>
<tr>
<td>17</td>
<td>COUNTR</td>
<td>EPZ</td>
<td>$1D</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>HBASL</td>
<td>EPZ</td>
<td>$26</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>HBASH</td>
<td>EPZ</td>
<td>$27</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>HMASK</td>
<td>EPZ</td>
<td>$30</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>AIL</td>
<td>EPZ</td>
<td>$3C</td>
<td>Monitor A1.</td>
</tr>
<tr>
<td>22</td>
<td>AIR</td>
<td>EPZ</td>
<td>$3D</td>
<td>Monitor A2.</td>
</tr>
<tr>
<td>23</td>
<td>A2L</td>
<td>EPZ</td>
<td>$3E</td>
<td>Basic 'Start of Vars'</td>
</tr>
<tr>
<td>24</td>
<td>A2K</td>
<td>EPZ</td>
<td>$3F</td>
<td>Delta-X for HLIN, SHAPE</td>
</tr>
<tr>
<td>25</td>
<td>LOMEML</td>
<td>EPZ</td>
<td>$4A</td>
<td>Shape Temp.</td>
</tr>
<tr>
<td>26</td>
<td>LOMEMH</td>
<td>EPZ</td>
<td>$4B</td>
<td>Delta-Y for HLIN, SHAPE</td>
</tr>
<tr>
<td>27</td>
<td>DXL</td>
<td>EPZ</td>
<td>$50</td>
<td>Rot Quadrant (SHAPE). Error for HLIN.</td>
</tr>
<tr>
<td>28</td>
<td>DXH</td>
<td>EPZ</td>
<td>$51</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>SHAPEX</td>
<td>EPZ</td>
<td>$51</td>
<td>Basic start of prog. pr</td>
</tr>
<tr>
<td>30</td>
<td>D Y</td>
<td>EPZ</td>
<td>$52</td>
<td>Basic end of vars ptr.</td>
</tr>
<tr>
<td>31</td>
<td>QDRNT</td>
<td>EPZ</td>
<td>$53</td>
<td>Basic acc.</td>
</tr>
<tr>
<td>32</td>
<td>EL</td>
<td>EPZ</td>
<td>$54</td>
<td>Prior X-coord save</td>
</tr>
<tr>
<td>33</td>
<td>EH</td>
<td>EPZ</td>
<td>$55</td>
<td>After HLIN or Hplot.</td>
</tr>
<tr>
<td>34</td>
<td>PPL</td>
<td>EPZ</td>
<td>$56</td>
<td>HLIN, Hplot y-coord save</td>
</tr>
<tr>
<td>35</td>
<td>PPH</td>
<td>EPZ</td>
<td>$57</td>
<td>X-reg save for basic</td>
</tr>
<tr>
<td>36</td>
<td>PVL</td>
<td>EPZ</td>
<td>$58</td>
<td>Color for Hplot, Hposw</td>
</tr>
<tr>
<td>37</td>
<td>PVH</td>
<td>EPZ</td>
<td>$59</td>
<td>Horiz offset save</td>
</tr>
<tr>
<td>38</td>
<td>ACL</td>
<td>EPZ</td>
<td>$60</td>
<td>Hi-res page ($20 normal)</td>
</tr>
<tr>
<td>39</td>
<td>ACH</td>
<td>EPZ</td>
<td>$61</td>
<td>Scale for SHAPE, MOVE</td>
</tr>
<tr>
<td>40</td>
<td>XOL</td>
<td>EQU</td>
<td>$320</td>
<td>Start of</td>
</tr>
<tr>
<td>41</td>
<td>XOH</td>
<td>EQU</td>
<td>$321</td>
<td>SHAPE TABLE.</td>
</tr>
<tr>
<td>42</td>
<td>YO</td>
<td>EQU</td>
<td>$322</td>
<td>Collision count.</td>
</tr>
<tr>
<td>43</td>
<td>BXSAV</td>
<td>EQU</td>
<td>$323</td>
<td>Start of SHAPE TABLE.</td>
</tr>
<tr>
<td>44</td>
<td>HCOLOR</td>
<td>EQU</td>
<td>$324</td>
<td>Switch to Hi-res video</td>
</tr>
<tr>
<td>45</td>
<td>HNDX</td>
<td>EQU</td>
<td>$325</td>
<td>Select text/graphics</td>
</tr>
<tr>
<td>46</td>
<td>HPAG</td>
<td>EQU</td>
<td>$326</td>
<td>Select graphics mode</td>
</tr>
<tr>
<td>47</td>
<td>SCALE</td>
<td>EQU</td>
<td>$327</td>
<td>Basic mem full error</td>
</tr>
<tr>
<td>48</td>
<td>SHAPXL</td>
<td>EQU</td>
<td>$328</td>
<td>Basic range error</td>
</tr>
<tr>
<td>49</td>
<td>SHAPXH</td>
<td>EQU</td>
<td>$329</td>
<td>2-byte tape read setup</td>
</tr>
<tr>
<td>50</td>
<td>COLLSN</td>
<td>EQU</td>
<td>$32A</td>
<td>TWO-EDGE TAPE SENSE</td>
</tr>
<tr>
<td>51</td>
<td>SHSTRT</td>
<td>EQU</td>
<td>$C00</td>
<td>Tape read (A1, A2).</td>
</tr>
<tr>
<td>52</td>
<td>HIRES</td>
<td>EQU</td>
<td>$C057</td>
<td>Read without header</td>
</tr>
<tr>
<td>53</td>
<td>MIXSET</td>
<td>EQU</td>
<td>$C053</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>TXTCLR</td>
<td>EQU</td>
<td>$C050</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>MEMFULL</td>
<td>EQU</td>
<td>$E36B</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>RANGERR</td>
<td>EQU</td>
<td>$E368</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>ACADR</td>
<td>EQU</td>
<td>$F11E</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>RD2BIT</td>
<td>EQU</td>
<td>$FCFA</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>READ</td>
<td>EQU</td>
<td>$FEFD</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>READX1</td>
<td>EQU</td>
<td>$FF02</td>
<td></td>
</tr>
</tbody>
</table>
63 * RAM VERSION $800 TO $BFF
64
65 *
66
\800: A9 20 67 SETHRL LDA $20
\802: 8D 26 03 68 STA HPG
\805: AD 57 C0 69 LDA HIRES
\808: AD 53 C0 70 LDA MIXSER
\808: AD 50 CD 71 LDA TXTCLR
\80E: A9 00 72 HCLR LDA $50
\810: 85 1C 73 BKGDOS STA HCOLOR1
\812: AD 26 03 74 BKGD LDA HPG
\815: 85 1B 75 STA SHAPEH
\817: A0 00 76 LDY $50
\819: 84 1A 77 STY SHAPEL
\81B: A5 1C 78 BKGD1 LDA HCOLOR1
\81D: 91 1A 79 STA SHAPEL,Y
\81F: 20 A2 08 80 JSR CSHFT2
\822: CB 81 INY
\823: D0 F6 82 BNE BKGD1
\825: E6 1B 83 INC SHAPEH
\827: A5 1B 84 LDA SHAPEH
\829: 29 1F 85 AND $5F
\82B: D0 EE 86 BNE BKGD1
\82D: 60 87 RTS

INIT FOR $2000-3FFF
HI-RES SCREEN MEMORY.
SET HIRES DISPLAY MODE
WITH TEXT AT BOTTOM.
SET GRAPHICS DISPLAY
SET FOR BLACK BKGD.
INIT HI-RES SCREEN MEM
FOR CURRENT PAGE, NORMAL
$2000-3FFF OR $4000-5F
(SHAPEL,H) WILL SPECIFY
32 SEPARATE PAGES
THROUGHOUT THE INIT.
TEST FOR DONE.
HI-RES GRAPHICS POSITION AND PLOT SUBRS

4:50 P.M., 12/2/1977

0620:  8D 22 03 90  HPOSN
0631:  8E 20 03 91
0634:  8C 21 03 92
0537:  48                     YO
0638:  29 C0 03 94
063A:  85 26 95
063C:  4A                     XOL
063D:  4A                     XOH
063E:  05 26 98
0640:  85 26 99
0422:  68                     PHA
0431:  85 27 101
0434:  0A                     ASL A
0438:  0A                     ASL A
043B:  0A                     ASL A
0442:  26 27 105
0446:  0A                     ROL HBASH
044A:  0A                     ASL A
044E:  66 26 109
0550:  A5 27 110
0572:  29 1F 111
0574:  OD 26 03 112
0577:  85 27 113
0579:  8A                     TXA
057A:  C0 00 115
057C:  F0 05 116
057E:  A0 23 117
0680:  69 04 118
0682:  CB 119  HPOSN1
0683:  E9 07 120  HPOSN2
0685:  B0 FB 121
0687:  8C 25 03 122
068A:  AA                     HPOSN
068B:  AD EA 06 124
068E:  85 30 125
0690:  98                     TYA
0691:  4A                     LSR A
0692:  AD 24 03 128
0695:  85 1C 129  HPOSN3
0577:  B0 29 130
0579:  60                     RTS
057A:  20 2E 06 132  HPLST
057D:  A5 1C 133  HPLS1
067F:  51 26 134
0651:  25 30 135
0653:  51 26 136
0625:  91 26 137
0557:  60                     139 *

STA YO
STX XOL
STY XOH
PHA
AND #SCO
STA HBSAL
LSR A
LSR A
ORA HBSAL
STA HBSAL
PLA
STA HBASH
ASL A
ASL A
ASL A
ROL HBASH
ASL A
ROL HBASH
ASL A
ORA HBSAL
LDA HBASH
AND #51F
ORA HPG
STA HBASH
TXA
CPY #50
BEQ HPOSN2
LDY #523
ADC #54
INY
SBC #57
BCC HPOSN1
STY HNDX
TAX
LDA MSKTBLS-5F9,X
STA HMASK
TYA
LSR A
LDA HCOLOR
STA HCOLOR1
BCC CSHFT2
RTS
JSR HPOSN
LDA HCOLOR1
EOR (HBSAL),Y
STA HCOLOR1
EOR (HBSAL),Y
STA (HBSAL),Y
RTS

ENTER WITH Y IN A-REG,
XL IN X-REG,
AND XH IN Y-REG.

FOR Y-COORD = OADBCEF
CALCULATES BASE ADDR
IN HBSAL, HBASH FOR
ACCESSING SCREEN MEM
VIA (HBSAL), Y ADDR
MODE.

CALCULATES
HBASH = PPPFGHCD,
HBASL = EABAB000

WHERE PPP=001 FOR $20B
SCREEN MEM RANGE AND
PPP=010 FOR $4000-7F
(GIVEN Y-COORD=ADEF)

DIVIDE XO BY 7 FOR
INDEX FROM BASE ADDR
(QUOTIENT AND BIT
WITHIN SCREEN MEM 35
(MASK SPEC'D BY REME)

SUBTRACT OUT SEvens.

WORKS FOR XO FROM
0 TO 279, LOW-ORDER
BYTE IN X-REG,
HIGH IN Y-REG ON ENT

IF ON ODD BYTE (CARRY)
THEN ROTATE HCOLOR 5

BIT FOR 180 DEGREES
PRIOR TO COPYING TO

SUBSTITUTE CORRESPONG

BIT OF HCOLOR1.
USE SIGN FOR LFT/RT SE

SHIFT LOW-ORDER
7 BITS OF HMASK
ONE BIT TO LSB.

DECR HORIZ INDEX.

WRAP AROUND SCREEN.
NEW HMASK, RIGHTMOST
DOT OF BYTE.

UPDATE HORIZ INDEX.

ROTATE LOW-ORDER
7 BITS OF HCOLORI
ONE BIT POSN.

ZXYXYXYX -> ZXYXYXYX

XYZ

SHIFT LOW-ORDER
7 BITS OF HMASK
ONE BIT TO MSB.

NEXT BYTE.

WRAP AROUND SCREEN IF
ALWAYS TAKEN.
4:50 P.M., 12/2/1977

NO 90 DEG ROT (X-OR).
IF B2=0 THEN NO PLOT.
FOR EX-OR INTO SCREEN.
SCREEN BIT SET?
ALWAYS TAKEN.
NO 90 DEG ROT.
IF B2=0 THEN NO PLOT.
SET HI-RES SCREEN BIT.
TO CORRESPONDING HCB.
IF BIT OF SCREEN CHANGES.
THEN INCR COLS. 
ADD QDRNT TO.
SPECIFIED VECTOR.
AND MOVE LFT, RT.
UP, OR DWN BASED.
ON SIGN AND CARRY.
SIGN FOR UP/DWN SELECT.
CALC BASE ADDRESS.
(ADR OF LEFTMOST BIT.
FOR NEXT LINE UP.
IN (HBASL,HBASH).
WITH 192-LINE WRAPAR.
**** BIT MAP ****
FOR ROW = ABCDEFGH,

WHERE PPP=001 FOR PRIE.
HI-RES PAGE ($2000-...
4:50 P.M., 12/2/1977

0927: 85 27 229 UPDN1 STA HBASL
0929: 60 230 RTS
092A: 18 231 DOWN CLC
092B: A5 27 232 DOWN4 LDA HBASL
092D: 69 04 233 ADC $34
092E: 234 EQA EQU $-1
092F: 2C EA 09 235 BIT EQ1C
0931: D0 F3 236 BNE UPDN1
0934: 06 26 237 ASL HBASL
0936: 90 19 238 BCC DOWN1
0938: 69 E0 239 ADC $5E0
093A: 18 240 CLC
093B: 2C 2E 09 241 BIT EQ4
093E: F0 13 242 BEQ DOWN2
0940: A5 26 243 LDA HBASL
0942: 69 50 244 ADC $550
0944: 49 F0 245 EOR $3F0
0946: F0 02 246 BEQ DOWN3
0948: 49 F0 247 EOR $3F0
094A: 85 26 248 DOWN3 STA HBASL
094C: AD 26 03 249 LDA HPAG
094E: 90 02 250 BCC DOWN2
0951: 69 E0 251 DOWN1 ADC $5E0
0953: 66 26 252 DOWN2 ROR HBASL
0955: 90 D0 253 BCC UPDN1

CALC BASE ADR FOR NEXT DOWN TO (HBASL, HBASL)
WITH 192-LINE WRAPAR
HI-RES GRAPHICS LINE DRAW SUBRS

PAGE 8

4:50 P.M., 12/2/1977

0957: 48  256  HLINRL  PHA
0958: A9  00  257  LDA $50  SET (XOL,XOH) AND
095A: 8D  20  03  258  STA XOL  YO TO ZERO FOR
095D: 8D  21  03  259  STA XOH  REL LINE DRAW
0960: 8D  22  03  260  STA YO  (DX, DY).
0963: 68  261  PLA  ON ENTRY
0964: 48  262  HLIN  PHA
0965: 38  263  SEC  XL: A-REG
0966: ED  20  03  264  SBC XO  XH: X-REG
0969: 48  265  PHA  Y: Y-REG
096A: 8A  266  TXA  SBC XOH
096B: ED  21  03  267  STA QDRNT  CALC ABS(X-XO)
096E: 85  53  268  BCS HLIN2  IN (DXL,DXH)
0970: BO  0A  269  PLA  X DIR TO SIGN BIT
0972: 68  270  PLA  OF QDRNT.
0973: 49  FF  271  ADC $1  0=RIGHT (DX POS)
0975: 69  01  272  PHA  1=LEFT (DX NEG)
0977: 48  273  LDA $50  SBC QDRNT
0978: A9  00  274  STA DXH  CALC ABS(X-XO)
097A: E5  53  275  STA EH  IN (EL, EH) TO
097C: 85  51  276  HLIN2  ABS(X-XO)
097E: 85  55  277  STA DXL
0980: 68  278  STA EL
0981: 85  50  279  STA DXL
0983: 85  54  280  STA EL
0985: 68  281  PLA
0986: ED  20  03  282  STA XO  SBC YO
0989: 8E  21  03  283  STA XO
098C: 98  284  TYA  CALC -ABS(Y-YO)-1
098D: 18  285  CLC  IN DY.
098E: ED  22  03  286  SBC YO  ROTATE Y DIR INTO
0991: 90  04  287  BCC HLIN3  QDRNT SIGN BIT
0993: 49  FF  288  EOR $FF  (0=UP, 1=DOWN)
0995: 69  FE  289  ADC $FE  ROR QDRNT
0997: 85  52  290  HLIN3  STA DY
0999: 8C  22  03  291  STY YO  INIT (COUNTL,COUNTH),
099C: 66  53  292  SEC  TO -(DELTX+DELTY+1)
099E: 38  293  SEC
099F: E5  50  294  SBC DXL  INIT (COUNTL,COUNTH),
09A1: AA  295  TAX  TO -(DELTX+DELTY+1)
09A2: A9  FF  296  LDA $FF
09A4: E5  51  297  SBC DXH  HORIZ INDEX
09A5: 85  1D  298  STA COUNTH  ALWAYS TAKEN.
09A5: AC  25  03  299  LDY HNDX
09A8: BO  05  300  BCS MOVEX2  MOVE IN X-DIR. USE
09AD: 0A  301  MOVEX  ASSUME CARRY SET.
09AE: 20  88  08  302  ASL A  (EL, EH)-DELTY TO (EL, EH)
09B1: 38  303  SEC  NOTE: DY IS (-DELTY)-1
09B2: A5  54  304  MOVEX2  CARRY CLR IF (EL, EH).
09B4: 65  52  305  LDA EL  GOES NEG.
09B5: 65  54  306  ADC DY
09B8: A5  55  307  STA EL
09BA: EZ  00  308  LDA EH
09BE: E5  55  309  HCOUNT  SBC $50
09BC: 20  88  08  302  JSR LFTRT  QDRNT 96 FOR LFT/RTT
09BF: 38  303  SEC
09C2: A5  54  304  MOVEX2  LDA EL
09C4: 65  52  305  ADC DY
09C5: 65  54  306  STA EL
09CE: A5  55  307  LDA EH
09DA: E9  00  308  SBC $50
09DF: E5  55  309  HCOUNT  STA EH
<table>
<thead>
<tr>
<th>Time</th>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0930:1</td>
<td>B1 26</td>
<td>310</td>
<td>LDA (HBASL), Y</td>
</tr>
<tr>
<td>0930:1</td>
<td>45 1C</td>
<td>311</td>
<td>EOR HCOLOR1</td>
</tr>
<tr>
<td>0932:1</td>
<td>25 30</td>
<td>312</td>
<td>AND HMASK</td>
</tr>
<tr>
<td>7C4:1</td>
<td>51 26</td>
<td>313</td>
<td>EOR (HBASL), Y</td>
</tr>
<tr>
<td>0936:1</td>
<td>91 26</td>
<td>314</td>
<td>STA (HBASL), Y</td>
</tr>
<tr>
<td>0938:1</td>
<td>E8</td>
<td>315</td>
<td>INX</td>
</tr>
<tr>
<td>0939:1</td>
<td>D0 04</td>
<td>316</td>
<td>BNE HLIN4</td>
</tr>
<tr>
<td>09CB:1</td>
<td>E6 1D</td>
<td>317</td>
<td>INC COUNTH</td>
</tr>
<tr>
<td>09CD:1</td>
<td>F0 6B</td>
<td>318</td>
<td>BEQ RTS2</td>
</tr>
<tr>
<td>09CF:1</td>
<td>A5 53</td>
<td>319</td>
<td>HLIN4</td>
</tr>
<tr>
<td>09D1:1</td>
<td>B0 DA</td>
<td>320</td>
<td>LDA QDRNT</td>
</tr>
<tr>
<td>09D2:1</td>
<td>20 F9 08 321</td>
<td></td>
<td>BCS MOVEX</td>
</tr>
<tr>
<td>09D3:1</td>
<td>18</td>
<td>322</td>
<td>CLC</td>
</tr>
<tr>
<td>09D6:1</td>
<td>A5 54</td>
<td>323</td>
<td>LDA EL</td>
</tr>
<tr>
<td>09D9:1</td>
<td>65 50</td>
<td>324</td>
<td>ADC DXL</td>
</tr>
<tr>
<td>09DB:1</td>
<td>85 54</td>
<td>325</td>
<td>STA EL</td>
</tr>
<tr>
<td>09DD:1</td>
<td>A5 55</td>
<td>326</td>
<td>LDA EH</td>
</tr>
<tr>
<td>09DF:1</td>
<td>65 51</td>
<td>327</td>
<td>ADC DXL</td>
</tr>
<tr>
<td>09E1:1</td>
<td>5Q D9</td>
<td>328</td>
<td>MSKTLB</td>
</tr>
<tr>
<td>09E3:1</td>
<td>81</td>
<td>329</td>
<td>DBT $8.1</td>
</tr>
<tr>
<td>09E4:1</td>
<td>82 84 88 330</td>
<td></td>
<td>DBT $82, $84, $88</td>
</tr>
<tr>
<td>09E7:1</td>
<td>90 A0</td>
<td>331</td>
<td>DBT $90, $A0</td>
</tr>
<tr>
<td>09E9:1</td>
<td>CO</td>
<td>332</td>
<td>DBT $CO</td>
</tr>
<tr>
<td>09EA:1</td>
<td>1C</td>
<td>333</td>
<td>MSKTLB</td>
</tr>
<tr>
<td>09EB:1</td>
<td>FF FE FA</td>
<td></td>
<td>DBT $FF, $FE, $FA, $</td>
</tr>
<tr>
<td>09EE:1</td>
<td>FA EC E1</td>
<td></td>
<td>DBT $A1, $8D, $78, $</td>
</tr>
<tr>
<td>09F1:1</td>
<td>D4 C5 B4 334</td>
<td></td>
<td>DBT $FF, $FE, $FA, $</td>
</tr>
<tr>
<td>09F4:1</td>
<td>A1 8D 78</td>
<td></td>
<td>DBT $FF, $FE, $FA, $</td>
</tr>
<tr>
<td>09F7:1</td>
<td>61 49 31</td>
<td></td>
<td>DBT $FF, $FE, $FA, $</td>
</tr>
<tr>
<td>09FA:1</td>
<td>18 FF 335</td>
<td></td>
<td>DBT $FF, $FE, $FA, $</td>
</tr>
</tbody>
</table>
HI-RES GRAPHICS COORDINATE RESTORE SUBR.

4:50 P.M., 12/2/1977

PAGE: 10

09FC:  A5 26 338  HFIND  LDA  HBASL
09FE:  0A  339  ASL  A
09FF:  A5 27 340  LDA  HBASH
A001:  29 03 341  AND  #$3
A002:  2A  342  ROL  A
A003:  05 26 343  ORA  HBASL
A004:  0A  344  ASL  A
A005:  0A  345  ASL  A
A006:  0A  346  ASL  A
A007:  0A  347  ASL  A
A008:  8D 22 03 348  STA  YO
A009:  A5 27 349  LDA  HBASH
A00A:  4A  350  LSR  A
A00B:  4A  351  LSR  A
A00C:  6D 22 03 352  STY  YO
A00D:  AD 25 03 353  STA  YO
A00E:  4A  354  LDA  HNDX
A00F:  355  ASL  A
A010:  6D 25 03 356  ADC  HNDX
A011:  0A  357  ASL  A
A012:  AA  358  TAX
A013:  CA  359  DEX
A014:  A5 30 360  LDA  HMASK
A015:  29 7F 361  AND  #$7F
A016:  28 362  HFIND1  INX
A017:  4A  363  LSR  A
A018:  D0 FC 364  BNE  HFIND1
A019:  8D 21 03 365  STA  XOH
A01A:  8A  366  TXA
A01B:  18  367  GLC
A01C:  6D 25 03 368  ADC  HNDX
A01D:  90 03 369  BCC  HFIND2
A01E:  52 21 03 370  INC  XOH
A01F:  8D 20 03 371  HFIND2  STA  XOL
A020:  60  372  RTS2  RTS

CONVERTS BASE ADR TO Y-COORD.

FOR HBASL = EABAB000

HBASH = PPPF0GHC

GENERATE Y-COORD = ABCDEFGH

(PPP=SCREEN PAGE,

NORMALLY 001 FOR $2000-$3FFF

HI-RES SCREEN)

CONVERTS HNDX (INDEX FROM BASE ADR)

AND HMASK (BIT MASK) TO X-COORD

IN (XOL,XOH)

(RANGE $0-$133)

CALC HNDX*7 +

LOG (BASE 2) HMASK.
HI-RES GRAPHICS SHAPE DRAW SUBR

375 *
376 * SHAPE DRAW
377 * R = 0 TO 63
378 * SCALE FACTOR USED (1=NORMAL)
379 *

0A3B: 86 1A 380 DRAW STX SHAPEL DRAW DEFINITION
0A3D: 84 1B 381 STY SHAPEH POINTER.
0A3F: AA 382 DRAW1 TAX
0A40: 4A 383 LSR A ROT (SO-S3F)
0A41: 4A 384 LSR A QDRNT 0=UP, 1=RT
0A42: 4A 385 LSR A 2=DRAW, 3=LFT.
0A43: 4A 386 LSR A
0A44: 85 53 387 STA QDRNT
0A46: 8A 388 TAX
0A47: 29 0F 389 AND #$FF
0A49: AA 390 TAX
0A4A: BA EB 09 391 LDY COS,X SAVE COS AND SIN
0A4D: 84 50 392 STY DXL VALS IN DXL AND DY.
0A4F: 49 0F 393 EOR #$FF
0A51: AA 394 TAX
0A52: BA EC 09 395 LDY COS+1,X
0A55: C8 396 INY
0A56: 84 52 397 STY DY
0A58: AC 25 03 398 DRAW2 LDY MNDX BYTE INDEX FROM
0A5B: A2 00 399 LDX #$50 HI-RES BASE ADR.
0A5D: 8E 2A 03 400 STX COLLED CLEAR COLLISION COUNT.
0A60: A1 1A 401 LDA (SHAPEL,X) 1ST SHAPE DEF BYTE.
0A62: 85 51 402 DRAW3 STA SHAPEX
0A64: A2 80 403 LDX #$80 EL, EH FOR FRACTIONAL
0A66: 86 54 404 STX EL L,R,U,D VECTORS.
0A68: 86 55 405 STX EH SCALE FACTOR.
0A6A: AE 27 03 406 LDX SCALE
0A6D: A5 54 407 DRAW4 LDA EL IF FRACTION
0A6F: 38 408 SEC
0A70: 65 50 409 ADC DXL THEN MOVE IN
0A72: 85 54 410 STA EL SPECIFIED VECTOR
0A74: 90 04 411 BCC DRAWS DIRECTION.
0A76: 20 D8 08 412 JSR LRUD1
0A79: 18 413 CLC
0A7A: A5 55 414 DRAW5 LDA EH IF FRACTION
0A7C: 65 52 415 ADC DY THEN MOVE IN
0A7E: 85 55 416 STA EH SPECIFIED VECTOR
0A80: 90 03 417 BCC DRAW6 DIRECTION +90 DEG.
0A82: 20 D9 08 418 JSR LRUD2 LOOP ON SCALE
0A85: CA 419 DRAW6 DEX FACTOR.
0A86: D0 E5 420 BNE DRAWS NEXT 3-BIT VECTOR
0A88: A5 51 421 LDA SHAPEX OF SHAPE DEF.
0A8A: 4A 422 LSR A NOT DONE THIS BYTE.
0A8B: 4A 423 LSR A
0A8C: 4A 424 LSR A
0A8D: D0 D3 425 BNE DRAWS SHAPE DEFINITION.
0A8F: E6 1A 426 INC SHAPEL
0A91: D0 02 427 BNE DRAWS
0A93: E6 1B 428 INC SHAPEH

4:50 P.M., 12/2/1977
0A95: A1 1A 429 DRAWS LDA (SHAPEL,X) DONE IF ZERO.
0A97: D0 C9 430 BNE DRAWS
0A99: 60 431 RTS
434 *
435 * EX-OR SHAPE INTO SCREEN.
436 *
437 * ROT = 0 TO 3 (QUADRANT ONLY)
438 * SCALE IS USED
439 *

OA9A:  86 1A  440  XDRAW   STX  SHAPEL
OA9C:  84 1B  441  STY  SHAPEX  SHAPE DEFINITION
OA9E:  AA    442  XDRAW1  SHAPEL  POINTER.
OA9F:  4A    443  LSR  A  ROT (50-53F)
OAAD:  4A    444  LSR  A  QDRNT  0=UP, 1=RT,
OAAL:  4A    445  LSR  A  2=DWN, 3=LFT.
OAA2:  4A    446  LSR  A
OAA3:  85 53  447  STA  QDRNT
OAA5:  8A    448  TXA
OAA6:  29 0F  449  AND  #$F
OAA8:  AA    450  TAX
OAA9:  BC  EB 09 451  LDY  COS,X  SAVE COS AND SIN
OAAC:  84 50  452  STY  DXL  VALS IN DXL AND DY.
OAAE:  49 0F  453  EOR  #$F
OAB0:  AA    454  TAX
OAB1:  BC  EC 09 455  LDY  COS+1,X
OAB4:  C8    456  INY
OAB5:  84 52  457  STY  DY
OAB7:  AC  25 03 458  XDRAW2  INDEX FROM HI-RES
OABA:  A2  00  459  LDX  #$0  BASE ADR.
OABC:  8E  2A 03 460  STX  COLLSH  CLEAR COLLISION DETECT
OABF:  A1 1A  461  LDA (SHAPEL,X)  1ST SHAPE DEF BYTE.
OAC1:  85 51  462  XDRAW3  STA  SHAPEX
OAC3:  A2  80  463  LDX  #$80
OAC5:  86 54  464  STX  EL  EL,EH FOR FRACTIONAL
OAC7:  86 55  465  STX  EH  L,R,U,D VECTORS.
OAC9:  AE  27 03 466  LDX  SCALE  SCALE FACTOR.
OACC:  A5 54  467  XDRAW4  LDA  EL
OACE:  38    468  SEC  IF FRAC COS OVFL
OACF:  65 50  469  ADC  DXL  THEN MOVE IN
OADF:  85 54  470  STA  EL  SPECIFIED VECTOR
OADD:  20 C0  08 472  BCC  XDRAW5  DIRECTION
OAE5:  18    473  CLC
OAE9:  A5 55  474  XDRAW5  LDA  EH  IF FRAC SIN OVFL
OADB:  65 52  475  ADC  DY  THEN MOVE IN
OADD:  85 55  476  STA  EH  SPECIFIED VECTOR
OAEF:  90 00  477  BCC  XDRAW6  DIRECTION +90 DEG.
OAE1:  20 D9  08 478  JSR  LRJDX1
OAE2:  C0    479  XDRAW6  JSR  LRJDX2
OAE5:  D0  E5  480  LDA  SHAPEX  LOOP ON SCALE
OAE7:  A5  51  481  LSR  A  FACTOR.
OAE9:  44    482  LSR  A  NEXT 3-BIT VECTOR
OAEA:  4A    483  LSR  A  OF SHAPE DEF.
OAE1:  4A    484  LSR  A
OAE2:  DO  D3  485  BNE  XDRAW3
OAE5:  E6 1A  486  INC  SHAPEL  DONE IF ZERO.
OAF0:  DO  02  487  BNE  XDRAW7
OAF2:  E6 1B  488  INC  SHAPEL  SHAPE DEF.
OAF4:  A1 1A  489  XDRAW7  LDA (SHAPEL,X)
OAF5:  DO  C9  490  BNE  XDRAW3  DONE IF ZERO.
OAF8:  60    491  RTS
EN TRY POINTS FROM APPLE-II BASIC

4:50 P.M., 12/2/1977

009: 20 90 DB 494 BPOSN JSR PCOLR POSN CALL, COLR FROM

00C: 8D 24 03 495 STA HCOLOR

0AF: 20 AF DB 496 JSR GETYO YO FROM BASIC.

202: 4B 497 PHA XO FROM BASIC.

203: 20 9A OB 498 JSR GETXO

006: 68 499 PLA

007: 20 2E 08 500 JSR HPOSN

00A: AE 23 03 501 LDX BXSAV

00D: 60 502 RTS

00E: 20 F9 0A 503 BLOT JSR BPOSN PLOT CALL (BASIC).

011: 4C 7D 06 504 JMP HPLT1

014: AD 25 03 505 BLIN1 LDA HNDX

017: 4A 506 LSR A SET HCOLOR FROM

018: 20 90 DB 507 JSR PCOLR BASIC VAR COLR.

01B: 20 75 08 508 JSR HPOSN3

01E: 20 9A OB 509 BLINE JSR GETXO LINE CALL, GET XO FROM

021: 8A 510 TXA

022: 48 511 PHA

023: 98 512 TYA

024: AA 513 TAX

025: 20 AF DB 514 JSR GETYO YO FROM BASIC

025: A8 515 TAY

029: 68 516 PLA

02A: 20 64 09 517 JSR HLIN

02D: AE 23 03 518 LDX BXSAV

030: 60 519 RTS

031: 20 90 OB 520 BGNB JSR PCOLR BACKGROUND CALL

334: 4C 10 08 521 JMP BKGD0
DRAW CALL FROM BASIC.

EX-OR DRAW
FROM BASIC.

SAVE FOR BASIC.

SCALE FROM BASIC.

ROT FROM BASIC.
SAVE ON STACK.

START OF SHAPE TABLE.

SHAPE FROM BASIC.

SHAPE NO. = 2.

ADD 2-BYTE INDEX
TO SHAPE TABLE

START ADR
(X LOW, Y HI).

ROT FROM STACK.
BASIC PARAM FETCH SUBR'S

0901: A0 16 568 PCOLR  LDY #$16
0902: B1 4A 569 PBTE  LDA (LOMEML),Y
0904: D0 16 57Q  BNE RERR1  GET BASIC PARAM.
0906: 88 571  DEX  (ERR IF >255)
0907: B1 4A 572  LDA (LOMEML),Y
0909: 60 573  RTS  SAVE FOR BASIC.
090A: 8E 23 03 574  STX BXSAV  X0 LOW-ORDER BYTE.
090D: A0 05 575  LDY #$35
090F: B1 4A 576  LDA (LOMEML),Y
0911: AA 577  TAX  HI-ORDER BYTE.
0912: C8 578  INY
0913: B1 4A 579  LDA (LOMEML),Y
0915: A8 580  TAY
0916: E0 18 581  CPX #$18
0917: E9 01 582  SBC #$1  RANGE ERR IF >279.
0919: 90 ED 583  BCC RTSB
091A: 4C 68 EE 584 RERR1  JMP RANGERR
091B: A0 0D 585  GETYO  OFFSET TO YO FROM LOMM
091C: 20 92 OB 586  JSR PBYTE  GET BASIC PARAM YO.
091D: C9 CO 587  CMP #$CO  (ERR IF >191)
091E: B0 F4 588  BCS RERR1
091F: 60 589  RTS

SHAPE TAPE LOAD SUBROUTINE

PAGE: 18

PAGE: 18

0B91: 8E 23 03 592 SHLOAD  STX BXSAV  SAVE FOR BASIC.
0B9C: 20 1E F1 593  JSR ACADR  READ 2-BYTE LENGTH INTO
0B9F: 20 FD FE 594  JSR READ  BASIC ACC ($CE,CF).

* WARNING: OPERAND OVERFLOW IN LINE 595

0BC2: A9 00 595  LDA #$SHSTRT
0BC4: 85 3C 596  STA A1L
0BC6: 8D 28 03 597  STA SHAPXL
0BC9: 18 598  CLC
0BCA: 65 CE 599  ADC ACL
0BCC: A8 600  TAY
0BCD: A9 0C 601  LDA #$SHSTRT/256
0BCF: 85 3D 602  STA A1R
0BD1: 8D 29 03 603  STA SHAPXH
0BD4: 65 CF 604  ADC ACH
0BD6: B0 25 605  BCS MFULL1  NOT ENOUGH MEMORY.
0BD8: C4 CA 606  CPY PPL
0BDA: 48 607  PHA
0DB3: E5 CB 608  SBC PPH
0BDD: 68 609  PLA
0BDE: B0 1D 610  BCS MFULL1
0BEC: 84 3E 611  STY A2L
0BEE: 85 3F 612  STA A2H
0BE4: C8 613  INY
0BE5: D0 02 614  BNE SHLOD1
0BE7: 69 01 615  ADC #$1
0BEC: 84 4A 616  SHLOD1  STY LOMEML
0BEB: 85 4B 617  STA LOMEMH
0BFD: 84 CC 618  STY PVL
0BFE: 85 CD 619  STA PVH
0BF1: 20 FA FC 620  JSR RD2BIT
0BF4: A9 03 621  LDA #$53  5 SECOND HEADER.
0BF6: 20 02 FF 622  JSR READX1
0BF9: AE 23 03 623  LDX BXSAV
0BFC: 60 624  RTS