



# HI-RES TRICKS

## SECOND FEATURE

**T**hese powerful and flexible routines can be used easily from BASIC or machine language, and will give you a large palette of special effects. Spice up your programs with Hi-Res Tricks!

**H**i-Res Tricks is a package of 15 machine language routines that allow you to manipulate images on Apple's high-resolution graphics screen. You can create inverse screens, horizontal and vertical scrolling, mirror images (see Figure 1), upside down screens and more. Hi-Res Tricks can give your programs some extra visual punch, and you don't have to know machine language to use these routines.

### THE ROUTINES

Let's look at the 15 routines, one at a time.

**Negative Create.** This routine inverts the entire Hi-Res screen. All pixels (dots) that were on are turned off, and all pixels that were off are turned on.

**Left Shift.** This routine scrolls the Hi-Res display seven pixels (one byte) to the left. The screen wraps around: whatever scrolls off the screen on the left reappears on the right.

**Right Shift.** This routine scrolls the Hi-Res display seven pixels (one byte) to the right. The screen wraps around: whatever scrolls off the screen on the right reappears on the left.

**Up Shift.** The Hi-Res display is scrolled up one pixel vertically. The screen wraps around: whatever scrolls off the screen on the top reappears on the bottom.

**Down Shift.** This routine scrolls the Hi-Res display down one pixel vertically. The screen wraps around: whatever scrolls off the screen on the bottom reappears on the top.

**Left-Right Flip.** By swapping the left and right sides of the screen, the image is reversed horizontally.

**Top-Bottom Flip.** By swapping the top and bottom sides of the screen, the image is reversed vertically.

**Page 1 Plus Page 2.** This routine takes the contents of Hi-Res page 2 and adds it to whatever is on page 1. The effect is like a transparent overlay.

**Page 1 Minus Page 2.** This routine subtracts the contents of Hi-Res page 2 from page 1. This, in effect, erases some parts of page 1, depending on what is on page 2.

**Page 1 Transferred From Page 2.** The contents of page 2 are copied to page 1, erasing whatever was on page 1 in the process.

TABLE 1: Routine Calling Addresses

Routine	Starting Location	
	Hex	Decima <sup>1</sup>
Negative Create	\$6000	24576
Left Shift	\$601C	24604
Right Shift	\$606A	24682
Up Shift	\$60B8	24760
Down Shift	\$6110	24848
Left-Right Flip	\$6168	24936
Top-Bottom Flip	\$61F6	25078
Page 1 + Page 2	\$629E	25246
Page 1 - Page 2	\$62C2	25282
Page 1 → Page 2	\$62E8	25320
Page 1 ← Page 2	\$630A	25354
Two Page Shift Left	\$632C	25388
Two Page Shift Right	\$63A8	25512
Two Page Shift Up	\$6424	25636
Two Page Shift Down	\$64D8	25816
Page 1 - Page 2 Special	\$658C	25996

Note: These locations are valid only if you enter the programs as listed.

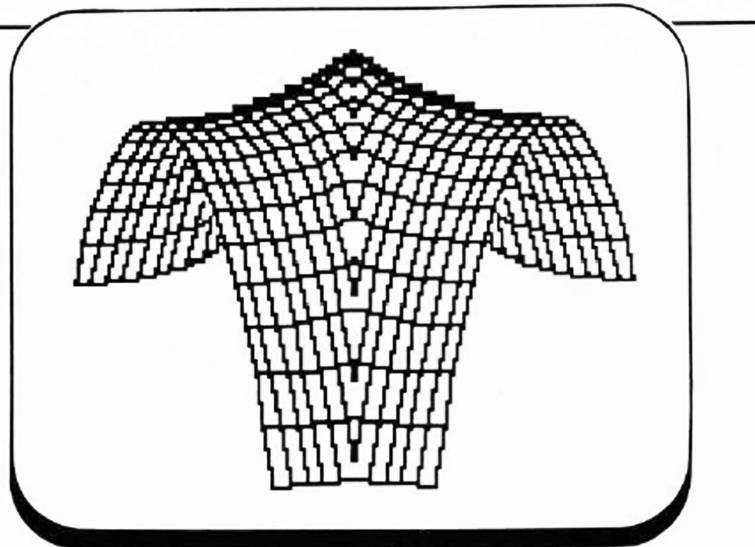


FIGURE 1: Left-Right Mirror

**Page 1 Transferred to Page 2.** This routine copies the contents of page 1 to page 2, erasing whatever was on page 2 in the process.

**Page 1 Transferred From Page 2 (Special).** This routine is special because only parts of page 2 are transferred onto page 1. This makes it possible to have as many variations to a display as you can fit onto one page of Hi-Res graphics. Page 1 can be set up for a general background, then parts of page 2 may be repeatedly transferred onto page 1 for different displays.

**Two-Page Left Shift.** Both Hi-Res pages are scrolled to the left as if they were one unit. This looks like a double-wide screen being scrolled.

**Two-Page Right Shift.** This routine scrolls both Hi-Res pages to the right.

**Two-Page Up Shift.** Both Hi-Res pages are scrolled up as if they were one unit. The effect is like a double-height screen being scrolled.

**Two-Page Down Shift.** This routine scrolls both Hi-Res pages down.

## USING HI-RES TRICKS

Each of the routines in Hi-Res Tricks can be used separately, and each can be loaded anywhere in memory. This makes them very flexible.

To use a routine from BASIC, simply BLOAD it into an unused section of memory and CALL it, using the address at which you loaded it. HIRES.TRICKS (Listing 1) contains all 15 routines as one large file, but you can split it up to suit your needs. HIRES.DEMO (Listing 2) loads all the routines into memory and CALLS the appropriate one for each demonstration. Table 1 shows the addresses to CALL if you load all the routines as listed.

To use the routines from machine language, just JSR to the location where you have stored the routine. You must save the X and Y Registers if they contain important data; the routines will change the values stored there.

As presented in Listing 1, the first seven routines (Negative Create through Top-Bottom Flip) are set up to act on Hi-Res page 1. They can be modified to act on Hi-Res page 2; see the Modifications section of this article.

The routine called Page 1 Transferred From Page 2 (Special) requires some extra explanation. Since only parts of page 2 are transferred onto page 1, you must first set up three memory locations before CALLing the routine.

1. \$658D (decimal 25997) sets the horizontal column of page 2 where the program starts the transfer.
2. \$6605 (decimal 26117) sets the horizontal column of page 2 where the program stops the transfer.
3. \$6599 (decimal 26009) sets the horizontal column of page 1 where the transfer begins.

These locations can be set from a machine language program by the statement:

```
LDA #SXX, STA SYYYY
```

or by POKEing them with values from BASIC. Forty columns are available, from 0 to 39. If the number you POKE is not in this range, the routine won't work properly and may overwrite parts of memory.

There is no horizontal stop for page 1. Whether you are going to transfer 2 or 5 or 35 columns from page 2, the exact same space must be available on page 1. The only information you have to supply for page 1 is where to start.

To run the demonstration program (Listing 2), you will need two Hi-Res pictures of your own on the disk. These can be any Hi-Res pictures you choose, as long as they

have been BSAVED from page 1 (\$2000). To save a picture in the correct format, use the command:

```
BSAVE picture,A$2000,L$2000
```

When you run the demonstration by typing RUN HIRES.DEMO, you are prompted for the file names of the two pictures you want to use. Enter their names and the program will manipulate them using many of the Hi-Res Tricks routines.

## ENTERING THE PROGRAMS

To enter the Hi-Res Tricks routines, either type the assembly source code in Listing 1 into your assembler and assemble the program, or enter the Monitor with CALL -151, type in the hexadecimal codes and save it with:

```
BSAVE HIRES.TRICKS,A$6000,L$609
```

Be sure you have saved Listing 1 before you enter and run Listing 2. To enter the demonstration program, type in the Applesoft program in Listing 2, and save it with:

```
SAVE HIRES.DEMO
```

For help with entering Nibble programs, see the directions in the Program Listings Section of this magazine.

## HOW HI-RES TRICKS WORKS

All of the programs make use of the zero page for temporary storage. This is done for two reasons: First, this allows relocatability; second, the zero page is faster and much easier to use than other areas. In all programs, the Y-Register is used to indirectly index the base address stored in the zero page so that the effective address is indexed horizontally across the Hi-Res page. The X-Register, however, is used to index vertically down the page.

### Negative Create

Lines 22-36 of Listing 1 are the Negative Create routine. After storing the first address in zero page and setting the index in the Y-Register to zero, the routine gets the value of a location in the Hi-Res area and then "complements" it. That is, if a bit was off, it is turned on; if it was on, it is turned off. This is done by the exclusive OR (EOR #SFF) in line 28.

The complemented value is then returned to its original location, the index is incremented and the next location is complemented. This process is repeated until the index goes through all the values from S00-SFF.

Next, the high byte of the address is incremented, and the process is repeated until the end of the page is reached. At the end of the page, the high byte has a value of S40 or more. This test is used in all the routines to determine when each routine should end. The routine then returns to the calling program.

## Left Shift and Right Shift

The Left Shift routine appears in lines 44-86 of Listing 1, and the Right Shift routine appears in lines 94-137.

Left Shift begins by saving the first byte in a horizontal line of the screen. Then every byte is shifted one position to the left. Finally, the saved first byte is put in the last byte. This creates the effect of scrolling the screen to the left.

To index the screen horizontally, the Y-register is incremented, which indirectly indexes the base address in zero page locations SFA and SFB. After moving across a line, the program proceeds to the next line. To get to the next line, the value \$04 is added to the high byte of the base address. The high byte is then checked to see if the end of the Hi-Res area has been reached; if not, the routine shifts that line.

Next, the value \$1F80 is subtracted from the base address to get us back into the Hi-Res page. This value is used because the Hi-Res page is not stored sequentially in memory but in eight interlaced sections. The X-Register keeps track of which section of the Hi-Res page is being acted upon. For example, after completing the first section, the base address in SFA and SFB is \$4000. Subtracting \$1F80 gives a base address of \$2080, which is the address of the next section.

After completing the first section, the base address in SFA and SFB is now \$4380. Subtracting \$1F80 again gives \$2400, but now the value of the X-Register is zero so the program subtracts an additional \$03D8, giving \$2028. This is the address of the start of the second section. A similar process brings the routine to the third section. After the third section is complete and the subtractions have been made, the value of the low byte of the base address is \$78. Since this is the only time this value is reached, it is used to determine the end of the routine.

The procedures in the Right Shift routine are similar to those in the Left Shift routine but in reverse order. The additions and subtractions of \$04 are the same, still moving down the screen. The difference here is that instead of beginning at \$00 and going to \$27, the Y-Register now goes from \$27 to \$00. The value of the last byte is now put into the first location to create the scrolling effect.

## Up Shift and Down Shift

Lines 145-191 of Listing 1 are the Up Shift routines, and lines 200-245 are the Down Shift routine.

Though these routines are similar to the side shift routines, they need an additional base address and the Y-register is set first (before the X-register). This allows the routine to index the addresses first vertically, then horizontally.

The top byte in a vertical column is temporarily saved. Then every byte is shifted

up one line, and the saved word is put into the bottom of the column.

The line addresses are calculated as in the side shifts, but they are recalculated completely for each horizontal move, which slows the program down somewhat. Because each execution moves the screen only one pixel, shifting up or down is slower than shifting sideways. However, it is possible to speed up the program if you are more concerned with speed than with relocatability.

## Left-Right Flip

This routine appears in lines 253-328 of Listing 1. The address calculations in the Left-Right Flip routine are the same as in the Left Shift routine because the program progresses down the screen. Again, the Y-Register is the horizontal index; however, the program takes a value from one address on a line and puts it in another some distance away, so two indexes are needed. They are stored in the zero page and loaded into the Y-Register as needed. One index goes from \$00 to \$14, and the other goes from \$27 to \$15. As the program proceeds, the first index is incremented and the second is decremented.

The bytes in the horizontal columns \$00 and \$27 are swapped first, then those in \$01 and \$26, and so on. This puts the first byte into the last location and the last into the first. Then the second goes into the next-to-last, the next-to-last into the second, and so on until \$14 and \$15 are swapped. After completing the first horizontal line, the routine moves down the screen to the next line.

Simply moving a byte from one screen location to another will not reverse the entire image. The bit order within the byte must be reversed also. The bits are reversed with little difficulty: loading the Accumulator with a byte, rotating the Accumulator right (ROR) and then rotating a memory location left (ROL).

By RORing the Accumulator, the rightmost bit goes into the Carry bit, and the rest of the bits are shifted one position to the right. When the memory location is ROLed, all the bits are shifted one position to the left and the Carry bit goes into the first location. As long as the memory location is zeroed first, the bit sequence in memory will be reversed from what was originally in the Accumulator.

One additional step is required. Since the display is created by the seven rightmost bits in a word (the picture bits), the RORing and ROLing is done only seven times. The eighth bit is the color bit and it must be preserved; otherwise, orange turns to green, and blue to violet.

The Accumulator is RORed one more time to get the last bit into the Carry bit; however, instead of ROLing the memory location this time, the Carry bit is only tested. If it is set, indicating that the eighth bit was on, then the value in memory is

ORed with \$80, setting the eighth bit again. If the Carry is clear, then that part of the program is skipped, leaving the eighth bit off.

## Top-Bottom Flip

Like the Up Shift and Down Shift programs, this routine (lines 335-425 of Listing 1) progresses up and down the page simultaneously. Like the Left-Right Flip, it swaps the values in two corresponding locations, but it does not need to reverse the bit order of the words. This routine first swaps the words in the top and bottom thirds of the page, then it swaps the words in the middle section. Also, since it works across the page horizontally, separate index locations are not needed. But additional zero page locations are needed to store more base addresses.

## Page 1 Plus Page 2

This routine appears in lines 442-460 of Listing 1. There are two methods to perform this operation depending on the background color of page 2. Since most graphs and charts have a black background, that's what you'll see when you run HIRESDemo (Listing 2). It can be modified quite easily for pictures and charts to a white or colored background.

In order to combine the two pages, the bits which make up the pages must be combined. If a bit is off in a byte in page 1, then we want to keep it off. If it is on, then we want to keep it on. If all of the bits in one of the bytes are zero, ANDing it with another byte will turn off all of the bits in that byte too. To get a black background, all of the bits in the background must be off. Therefore, in order to combine bits that are off in page 2 with bits that are on in page 1, the bytes must be ORed. This turns on a bit if it is on in either or both bytes.

For a white or colored background, the effect is reversed. If you assume a white background in page 2, then all of the bits would be on. ORing the byte from page 2 with a byte from page 1 would turn on all the bits including any bits that should be off. In this case, the bytes must be ANDed so that if a bit is off in page 1, the same bit will be off in the combined byte. In the AND operation, a bit is on only if the same bit is on in the first word and the second.

As shown, this routine works only if the background color of page 2 is black. To change it to a white or colored background, change the byte in location \$62B0 to \$31. This changes that step from an OR to an AND.

## Page 1 Minus Page 2

This routine (lines 468-487 of Listing 1) reverses the process of the Page 1 Plus Page 2 routine. Line 477 (location \$62D4) holds an exclusive OR (EOR). By EORing the

byte from page 2, the bits are complemented or reversed; if a bit is on, it's turned off while if it's off, it's turned on. The complemented byte is then combined with the equivalent word in page 1.

By complementing the byte, we have effectively reversed the color. If it was originally black (all bits off), it will now be white (all bits on). So, following the same logic as above, the bytes must be ANDed.

To change this program to handle a white or colored background in page 2, the instruction in location \$62D6 must be changed to an OR (\$11) because the white background will be effectively changed to black by the EOR process.

## MODIFICATIONS

The routines in Hi-Res Tricks can be modified to create even more Hi-Res effects. For example, you can shift the display left,

**A**ll of the programs make use of the zero page for temporary storage.

right, up and down without a scroll effect. You can also create mirror images of your displays. Other easy modifications will make the routines work on page 2. More involved modifications will allow the routines to operate on only specific sections of the pages.

The hex addresses and POKE instructions that follow are only valid if you have entered the programs exactly as listed. If you relocated them for any reason, you'll have to adjust the addresses.

Before you attempt any of these modifications, make sure you have a copy of the original version available as a backup.

### No Scroll

Shifting the Hi-Res page left, right, up or down without scrolling erases the display once it has gone past the edge of the page. The portion of the display that scrolls off the screen is lost forever; there is no way to bring it back. So be sure that you have saved your Hi-Res graphic safely to disk if you ever want to use it again!

To modify the Left Shift routine so that it doesn't scroll, change locations \$6039 and \$603A to NOPs (SEA) from the Monitor or by issuing the statement POKE 24633,234 : POKE 24634,234 from BASIC. However, whatever is in the last vertical column on the right will then be duplicated each time the program is executed.

Similar modifications can be made to perform right, up and down shifts without scrolling.

### Making Mirrors

Changing the contents of locations \$61B6 and \$61B7 to SEA (POKE 25014,234: POKE 25015,234) will duplicate the left side of the page onto the right side, but it will not transfer the right to the left. The left side is unchanged, producing a mirror effect in which the left side is mirrored in the right.

Similarly, changing the contents of locations \$61B0 and \$61B1 to SEA (POKE 25008,234 : POKE 25009,234) will cause the right side to be mirrored by the left.

### Page Two Operations

To operate the programs on page 2, just change the base address locations and the test for the end of page as follows:

**Negative Create.** Change location \$6005 to \$40 and change \$6018 to \$60 (using the statement POKE 24581,64 : POKE 24600,96).

**Left Shift.** Change location \$6021 to \$40 and change \$6043 to \$60 (using the statement POKE 24609,64 : POKE 24643,96).

**Right Shift.** Change location \$606F to \$40 and change \$6091 TO \$60 (using the statement POKE 24687,64 : POKE 24721,96).

**Up Shift.** Change location \$60BF to \$44, change \$60C7 to \$40 and change \$60E2 to \$60 (using the statement POKE 24767,68 : POKE 24775,64 : POKE 24802,96).

**Down Shift.** Change location \$6117 to \$5B, change \$611F to \$5F and change \$613A to \$40 (using the statement POKE 24855,91 : POKE 24863,95 : POKE 24890,64).

**Left-Right Flip.** Change location \$616D to \$40 and change \$61CA to \$60 (using the statement POKE 24941,64 : POKE 25034,96).

**Top-Bottom Flip.** Change location \$61FD to \$40, change \$6205 to \$5F, change \$620D to \$40, change \$6215 to \$5F, change \$6235 to \$60 and change \$627A to \$60. (POKE 25085,64 : POKE 25093,95 : POKE 25101,64 : POKE 25109,95 : POKE 25141,96 : POKE 25210,96)

To make these changes "on the fly" from BASIC, just POKE as indicated. If you plan to return to the original function within the same program, be sure to save the original values by PEEKing them first.

Many other effects can be created with careful program modifications. These include operation on part of a Hi-Res screen and use of a phantom Hi-Res page 3. Other modifications can improve the speed of the Hi-Res Tricks routines at the expense of memory and relocatability.

## Hi-Res Tricks

### Listing 1 for Hi-Res Tricks HIRES.TRICKS

```

1 .....
2 .....
3 .....
4 * HIRES TRICKS .....
5 * BY JOHN R. MAJKA .....
6 * COPYRIGHT (C) 1987 .....
7 * BY MICROSPARC, INC. ....
8 * CONCORD, MA 01742 .....
9 .....
10 * PAGE 1 GRAPHICS ROUTINES .....
11 * MERLIN PRO ASSEMBLER .....
12 .....
13 .....
14 ORG $6000 .....
15 .....
16 .....
17 .....
18 * NEGATIVE CREATE * .....
19 .....
20 .....
21 .....
22 LDA #$00 .....
23 STA $FA .....
24 LDA #$20 .....
25 STA $FB .....
26 NEG1 LDY #$08 .....
27 NEG2 LDA ($FA),Y .....
28 EOR $FF .....
29 STA ($FA),Y .....
30 INY .....
31 BNE NEG2 .....
32 INC $FB .....
33 LDA $FB .....
34 CMP #$40 .....
35 BNE NEG1 .....
36 RTS .....
37 .....
38 .....
39 .....
40 * LEFT SHIFT * .....
41 .....
42 .....
43 .....
44 LDA #$00 .....
45 STA $FA .....
46 LDA #$20 .....
47 STA $FB .....
48 LEFT1 LDY #$08 .....
49 LEFT2 LDY #$00 .....
50 LDA ($FA),Y .....
51 STA $FE .....
52 LEFT3 INY .....
53 LDA ($FA),Y .....
54 DEY .....
55 STA ($FA),Y .....
56 INY .....
57 CPY #$27 .....
58 BNE LEFT3 .....
59 LDA $FE .....
60 STA ($FA),Y .....
61 LDA $FB .....

```

6030: 18	62	CLC		60F5: A5 FA	177	LDA	SFA	6180: 91 FA	292	STA	(SFA) .Y
603E: 89 04	63	ADC	#504	60F7: 38 FA	178	SEC		6182: A4 FC	293	LDY	SFC
6040: 85 FB	64	STA	SFB	60F8: E9 D8	179	SBC	#108	6184: A5 EE	294	LDA	SEE
6042: C9 E0	65	CMP	#540	60FA: 85 FA	180	STA	SFA	6186: 91 FC	295	STA	(SFA) .Y
6044: 30 F0	66	BMI	LEFT2	60FC: A5 FB	181	LDA	SFB	6188: E6 FC	296	INC	SFC
6046: A5 FA	67	LDA	SFA	6100: 85 FB	182	SBC	#103	618A: C6 FD	297	DEC	SFC
6048: 38	68	SEC		6102: A5 FA	184	STA	SFB	618C: A5 FC	298	LDA	SFC
6049: E9 80	69	SBC	#580	6104: C9 78	185	LDA	SFA	618E: C9 14	299	CMP	#114
604B: 85 FA	70	STA	SFA	6106: D0 C6	186	CMP	#178	61C0: D0 BA	300	BNE	LR3
604D: A5 FB	71	LDA	SFB	6108: A5 FE	187	BNE	UP2	61C2: A5 FB	301	LDA	SFB
604F: E9 1F	72	SBC	#51F	610A: 91 FC	188	LDA	SFE	61C4: 18	302	CLC	
6051: 85 FB	73	STA	SFB	610C: 88	189	STA	(SFC) .Y	61C5: 69 04	303	ADC	#504
6053: CA	74	DEX		610E: 10 AB	190	DEY		61C7: 85 FB	304	STA	SFB
6054: D0 D0	75	BNE	LEFT2	6110: 60	191	BPL	UP1	61C9: C9 A0	305	CMP	#540
6056: A5 FA	76	LDA	SFA	610F: 10	192	RTS		61CB: 30 A7	306	BMI	LR2
6058: 38	77	SEC			193			61CD: A5 FA	307	LDA	SFA
6059: E9 D8	78	SBC	#5D8		194			61CF: 38	308	SEC	
605B: 85 FA	79	STA	SFA		195			61D0: E9 80	309	SBC	#580
605D: A5 FB	80	LDA	SFB		196			61D2: 85 FA	310	STA	SFA
605F: E9 03	81	SBC	#503		197			61D4: A5 FB	311	LDA	SFB
6061: 85 FB	82	STA	SFB		198			61D6: E9 1F	312	SBC	#51F
6063: A5 FA	83	LDA	SFA		199			61D8: 85 FB	313	STA	SFB
6065: C9 78	84	CMP	#578		200			61DA: C6 FE	314	DEC	SFE
6067: D0 D0	85	BNE	LEFT1		201			61DC: D0 96	315	BNE	LR2
6069: 00	86	RTS			202			61DE: FB 02	316	LDR	SFC
	87				203			61E0: D0 8E	317	LRB	BNE LR1
	88				204			61E2: A5 FA	318	LR9	LDA SFA
	89				205			61E4: 38	319	SEC	
	90				206			61E5: E9 D8	320	SBC	#5D8
	91				207			61E7: 85 FA	321	STA	SFA
	92				208			61E9: A5 FB	322	LDA	SFB
	93				209			61EB: E9 03	323	SBC	#503
606A: A9 00	94	LDA	#500	6122: 81 FC	208	LDA	(SFC) .Y	61ED: 85 FB	324	STA	SFB
606C: 85 FA	95	STA	SFA	6124: 85 FE	209	STA	SFE	61EF: A5 FA	325	LDA	SFA
606E: A9 20	96	LDA	#520	6126: A2 08	210	LDX	#508	61F1: C9 78	326	CMP	#578
6070: A5 FB	97	STA	SFB	6128: 01 FA	211	LDA	(SFA) .Y	61F3: D0 EB	327	BNE	LR8
6072: A2 08	98	RIGHT1	LDX #508	612A: 91 FC	212	STA	(SFC) .Y	61F5: 60	328	RTS	
6074: A0 27	99	RIGHT2	LDY #527	612C: A5 FA	213	LDA	SFA		329		
6076: B1 FA	100	RIGHT3	LDA (SFA) .Y	612E: 85 FC	214	STA	SFB		330		
6078: 85 FE	101		STA SFE	6130: A5 FB	215	LDA	SFB		331		
607A: 88	102		DEY	6132: 85 FD	216	STA	SFD		332		
607D: B1 FA	103		LDA (SFA) .Y	6134: 38	217	SEC			333		
607E: C8	104		STA (SFA) .Y	6135: E9 04	218	SBC	#584		334		
607F: 91 FA	105		STA (SFA) .Y	6137: 85 FB	219	STA	SFB		335		
6080: 88	106		DEY	6139: C9 20	220	CMP	#520		336		
6081: EA	107		NOP	613B: 10 EB	221	BPL	DOWN3		337		
6082: EA	108		NOP	613D: A5 FA	222	LDA	SFA		338		
6083: D0 F5	109		BNE RIGHT3	613F: 18	223	CLC			339		
6085: A5 FE	110		LDA SFE	6140: 69 80	224	ADC	#580		340		
6087: 91 FA	111		STA (SFA) .Y	6142: 85 FA	225	STA	SFA		341		
6089: A5 FB	112		LDA SFB	6144: A5 FB	226	LDA	SFB		342		
608B: 18	113		CLC	6146: 69 1F	227	ADC	#51F		343		
608C: 69 04	114		ADC #504	6148: 85 FB	228	STA	SFB		344		
608E: 85 FB	115		STA SFB	614A: CA	229	DEX			345		
6090: C9 40	116		CMP #540	614C: D0 0B	230	BNE	DOWN3		346		
6092: 30 E0	117		BMI RIGHT2	614D: A5 FA	231	LDA	SFA		347		
6094: A5 FA	118		LDA SFA	614F: 18	232	CLC			348		
6096: 38	119		SEC	6150: 69 D8	233	ADC	#5D8		349		
6097: E9 80	120		SBC #580	6152: 85 FA	234	STA	SFA		350		
6099: 85 FA	121		STA SFA	6154: A5 FB	235	LDA	SFB		351		
609B: A5 FB	122		LDA SFB	6156: 69 03	236	ADC	#503		352		
609D: E9 1F	123		SBC #51F	6158: 85 FB	237	STA	SFB		353		
609F: 85 FB	124		STA SFR	615A: A5 FA	238	LDA	SFA		354		
60A1: CA	125		DEX	615C: C9 58	239	CMP	#558		355		
60A2: D0 D0	126		BNE RIGHT2	615E: D0 C6	240	BNE	DOWN2		356		
60A4: A5 FA	127		LDA SFA	6160: A5 FE	241	LDA	SFE		357		
60A6: 38	128		SEC	6162: 91 FC	242	STA	(SFC) .Y		358		
60A7: E9 D8	129		SBC #5D8	6164: 88	243	DEY			359		
60A9: 85 FA	130		STA SFA	6166: 10 AB	244	BPL	DOWN1		360		
60AB: A5 FB	131		LDA SFB	6168: 60	245	RTS			361		
60AD: E9 03	132		SBC #503		246				362		
60AF: 85 FB	133		STA SFB		247				363		
60B1: A5 FA	134		LDA SFA		248				364		
60B3: C9 78	135		CMP #578		249				365		
60B5: D0 D0	136		BNE RIGHT1		250				366		
60B7: 60	137		RTS		251				367		
	138				252				368		
	139				253				369		
	140				254				370		
	141				255				371		
	142				256				372		
	143				257				373		
	144				258				374		
60B8: A0 27	145		LDY #527		259				375		
60BA: A9 00	146	UP1	LDA #500		260				376		
60BC: 85 FA	147		STA SFA		261				377		
60BE: A9 24	148		LDA #524		262				378		
60C0: 85 FB	149		STA SFB		263				379		
60C2: A9 00	150		LDA #500		264				380		
60C4: 85 FC	151		STA SFC		265				381		
60C6: A9 20	152		LDA #520		266				382		
60C8: 85 FD	153		STA SFD		267				383		
60CA: 81 FC	154		LDA (SFC) .Y		268				384		
60CC: 85 FE	155		STA SFE		269				385		
60CE: A2 08	156	UP2	LDX #508		270				386		
60D0: B1 FA	157	UP3	LDA (SFA) .Y		271				387		
60D2: 91 FC	158		STA (SFC) .Y		272				388		
60D4: A5 FA	159		LDA SFA		273				389		
60D6: 85 FC	160		STA SFC		274				390		
60D8: A5 FD	161		LDA SFD		275				391		
60DA: 85 FD	162		STA SFD		276				392		
60DC: 18	163		CLC		277				393		
60DD: 69 04	164		ADC #504		278				394		
60DF: 85 FB	165		STA SFB		279				395		
60E1: C9 40	166		CMP #540		280				396		
60E3: 30 EB	167		BMI UP3		281				397		
60E5: A5 FA	168		LDA SFA		282				398		
60E7: 38	169		SEC		283				399		
60E9: E9 80	170		SBC #580		284				400		
60EA: 85 FA	171		STA SFA		285				401		
60EC: A5 FB	172		LDA SFB		286				402		
60EE: E9 1F	173		SBC #51F		287				403		
60F0: 85 FB	174		STA SFB		288				404		
60F2: CA	175		DEX		289						
60F3: D0 D0	176		BNE UP3		290						
					291						
					292						
					293						
					294						
					295						
					296						
					297						
					298						
					299						
					300						
					301						
					302						
					303						
					304						
					305						
					306						
					307						
					308						
					309						
					310						
					311						
					312						
					313						
					314						
					315						
					316						
					317						
					318						
					319						
					320						
					321						
					322						
					323						
					324						
					325						
					326						
					327						
					328						



```

647C: D0 04 745 BNE UP2E 64FF: 85 FB 821 STA $FB 897
647E: F0 02 746 BEQ UP2E 6501: C9 40 822 CMP #540 898
6480: 10 A4 747 UP2D BPL UP2A 6503: 10 EB 823 BPL DN2C 899
6482: A9 00 748 UP2E LDA #500 6505: A5 FA 824 LDA $FA 900
6484: 85 FA 749 STA $FA 6507: 18 825 CLC 901
6486: A9 44 750 LDA #544 6508: 69 80 826 ADC #580 902
6488: 85 FB 751 STA $FB 650A: 85 FA 827 STA $FA 903
648A: A9 00 752 LDA #500 650C: A5 FB 828 LDA $FB 904
648C: 85 FC 753 STA $FC 650E: 69 1F 829 ADC #51F 905
648E: A9 40 754 LDA #540 6510: 85 FB 830 STA $FB 906
6490: 85 FD 755 STA $FD 6512: CA 831 DEX 907
6492: B1 FC 756 LDA (SFC),Y 6513: D0 DB 832 BNE DN2C 908
6494: 91 FE 757 STA (SFL),Y 6515: A5 FA 833 LDA $FA 909
6496: A2 08 758 UP2F LDX #508 6517: 18 834 CLC 910
6498: B1 FA 759 UP2G LDA (SFA),Y 6518: 69 DB 835 ADC #508 911
649A: 91 FC 760 STA (SFC),Y 651A: 85 FA 836 STA $FA 912
649C: A5 FA 761 LDA $FA 651C: A5 FB 837 LDA $FB 913
649E: 85 FC 762 STA $FC 651E: 69 03 838 ADC #583 914
64A0: A5 FB 763 LDA $FB 6520: 85 FB 839 STA $FB 915
64A2: 85 FD 764 STA $FD 6522: A5 FA 840 LDA $FA 916
64A4: 18 765 CLC 6524: C9 58 841 CMP #558 917
64A6: 69 04 766 ADC #504 6526: D0 C6 842 BNE DN2B 918
64A8: 85 FB 767 STA $FB 6528: A5 FC 843 LDA $FC 919
64AA: C9 08 768 CMP #560 652A: 85 FE 844 STA $FE 920
64AC: 30 EB 769 BMI UP2G 652C: A5 FD 845 LDA $FD 921
64AD: A5 FA 770 LDA $FA 652E: 85 04 846 STA $FF 922
64AF: 38 771 SEC 6530: D0 0F 847 BNE DN2E 923
64B0: E9 80 772 SBC #580 6532: F0 02 848 BEQ DN2E 924
64B2: 85 FA 773 STA $FA 6534: 10 A4 849 DN2D BPL DN2A 925
64B4: A5 FB 774 LDA $FB 6536: A9 D0 850 DN2E LDA #500 926
64B6: E0 1F 775 SBC #51F 6538: 85 FA 851 STA $FA 927
64B8: 85 FB 776 STA $FB 653A: A9 3B 852 LDA #53B 928
64BA: CA 777 DEX 653C: 85 FB 853 STA $FB 929
64BB: D0 DB 778 BNE UP2G 653E: A9 D0 854 LDA #500 930
64BD: A5 FA 779 LDA $FA 6540: 85 FC 855 STA $FC 931
64BF: 38 780 SEC 6542: A9 3F 856 LDA #53F 932
64C0: E9 D8 781 SBC #508 6544: 85 FD 857 STA $FD 933
64C2: 85 FA 782 STA $FA 6546: B1 FC 858 LDA (SFC),Y 934
64C4: A5 FB 783 LDA $FB 6548: 91 FE 859 STA (SFE),Y 935
64C6: E9 03 784 SBC #503 654A: A2 08 860 DN2F LDX #508 936
64C8: 85 FB 785 STA $FB 654C: B1 FA 861 DN2G LDA (SFA),Y 937
64CA: A5 FA 786 LDA $FA 654E: 91 FC 862 STA (SFC),Y 938
64CC: C9 78 787 CMP #578 6550: A5 FA 863 LDA $FA 939
64CE: D0 C6 788 BNE UP2F 6552: 85 FC 864 STA $FC 940
64D0: A5 FC 789 LDA $FC 6554: A5 FB 865 LDA $FB 941
64D2: 91 FC 790 STA (SFC),Y 6556: 85 FD 866 STA $FD 942
64D4: 88 791 DEY 6558: 38 867 SEC 943
64D6: 10 A9 792 BPL UP2D 6559: E9 04 868 SBC #504 944
64D8: 60 793 RTS 655B: 85 FB 869 STA $FB 945
64D9: 794 655D: C9 20 870 CMP #520 946
64DA: 795 655F: 10 EB 871 BPL DN2G 947
64DB: 796 6561: A5 FA 872 LDA $FA 948
64DC: 797 - 2 PAGE SHIFT DOWN- 6563: 18 873 CLC 949
64DE: 798 6564: 69 80 874 ADC #580 950
64DF: 799 6566: 85 FA 875 STA $FA 951
64E0: 800 6568: A5 FB 876 LDA $FB 952
64E1: A0 27 801 LDY #527 656A: 69 1F 877 ADC #51F 953
64E2: A9 D0 802 DN2A LDA #500 656C: 85 FB 878 STA $FB 954
64E3: 85 FA 803 STA $FA 656E: CA 879 DEX 955
64E4: A9 5B 804 LDA #55B 656F: D0 DB 880 BNE DN2G 956
64E5: 85 FB 805 STA $FB 6571: A5 FA 881 LDA $FA 957
64E6: A9 D0 806 LDA #500 6573: 18 882 CLC 958
64E7: A5 FC 807 STA $FC 6574: 69 DB 883 ALU #508 959
64E8: A9 5F 808 LDA #55F 6576: 85 FA 884 STA $FA 960
64E9: 85 FD 809 STA $FD 6578: A5 FB 885 LDA $FB 961
64EA: B1 FC 810 LDA (SFC),Y 657A: 69 03 886 ADC #503 962
64EB: 85 EF 811 STA $EF 657C: 85 FB 887 STA $FB 963
64EC: A2 08 812 DN2B LDX #508 657E: A5 FA 888 LDA $FA 964
64ED: B1 FA 813 DN2C LDA (SFA),Y 6580: C9 58 889 CMP #558 965
64EE: 91 FC 814 STA (SFC),Y 6582: D0 C6 890 BNE DN2F 966
64EF: A5 FA 815 LDA $FA 6584: A5 EF 891 LDA $EF 967
64F0: 85 FC 816 STA $FC 6586: 91 FC 892 STA (SFC),Y 968
64F1: A5 FB 817 LDA $FB 6588: 88 893 DEX 969
64F2: 85 FD 818 STA $FD 6589: 10 A9 894 BPL DN2D 970
64F3: 38 819 SEC 658B: 60 895 RTS 971
64F4: E9 04 820 SBC #504 658D: 60 896

```

--End assembly, 1545 bytes, Errors: 0

END OF LISTING 1

## Listing 2 for Hi-Res Tricks HIRES.DEMO

```

KEY PERFECT 5 0
RUN ON
HIRES TRICKS
=====
CODE-5.0  ADDR# - ADDR#  CODE-4.0
-----
8EA793EE  6000 - 604F  2A37
57D8E41C  6050 - 609F  273A
A0C0FAE  60A0 - 60EF  25D2
FE525392  60F0 - 613F  2ADD
71907C81  6140 - 618F  284C
9D835E37  6190 - 61DF  26F2
92040108  61E0 - 622F  2947
D77F2E7E  6230 - 627F  28C7
335FD30B  6280 - 62CF  299E
397CBA55  62D0 - 631F  2B25
1F4956C0  6320 - 636F  298B
A2126BDB  6370 - 63BF  283C
F0DC36CB  63C0 - 640F  2429
B08119C6  6410 - 645F  2BAA
1B78B30C  6460 - 64AF  2961
92796AC2  6480 - 64FF  2648
A6D67D62  6500 - 654F  298D
57DF2505  6550 - 659F  24A4
55D7D367  65A0 - 65EF  2953
99F87B78  65F0 - 6608  80CD
93CFD590 = PROGRAM TOTAL = 0609

```

```

10 REM *****
20 REM * HIRES DEMO *
30 REM * BY JOHN MAJKA *
40 REM * COPYRIGHT (C) 1987 *
50 REM * BY MICROSPARC, INC. *
60 REM * CONCORD, MA 01742 *
70 REM *****
80 TEXT : HOME : PRINT "HI-RES DEMO": PRINT
  "BY JOHN MAJKA": PRINT "COPYRIGHT 1987 B
  Y MICROSPARC, INC."
90 EF = 1: ONERR GOTO 580
100 VTAB 10: HTAB 1: CALL - 958: PRINT "NAM
  E OF FIRST PICTURE (? FOR CAT): "; INPUT
  "": PC$ = ON PC$ = "" GOTO 100: ON LEFT$
  (PC$,1) = "?" GOTO 620: PRINT CHR$(4)"
  VERIFY"PC$
110 EF = 2: ONERR GOTO 580
120 VTAB 14: HTAB 1: CALL - 958: PRINT "NAM
  E OF SECOND PICTURE (? FOR CAT): "; INPUT
  "": PD$: ON PD$ = "" GOTO 120: ON LEFT$
  (PD$,1) = "?" GOTO 620: PRINT CHR$(4)"
  VERIFY"PD$

```



```

130 EF = 3: ONERR GOTO 580
140 HGR :D$ = CHR$(4): POKE 34,20: HOME : E
    $ = CHR$(7)
150 PRINT D$;"BLOOD HIRES.TRICKS": PRINT D$:
    "BLOOD":PC$
160 PRINT "HIRES TRICKS-SINGLE PAGE": POKE 3
    4,21
170 PRINT "NEGATIVE CREATE":E$
180 FOR I = 1 TO 10: CALL 24576: FOR J = 1 TO
    500: NEXT J: NEXT I
190 HOME : PRINT "LEFT SHIFT":E$
200 FOR I = 1 TO 40: CALL 24604: NEXT
210 HOME : PRINT "RIGHT SHIFT":E$
220 FOR I = 1 TO 40: CALL 24682: NEXT
230 HOME : PRINT "UP SHIFT":E$
240 FOR I = 1 TO 96: CALL 24760: NEXT
250 HOME : PRINT "DOWN SHIFT":E$
260 FOR I = 1 TO 96: CALL 24848: NEXT
270 HOME : PRINT "LEFT-RIGHT FLIP":E$
280 FOR I = 1 TO 10: CALL 24936: FOR J = 1 TO
    1000: NEXT J: NEXT I
290 HOME : PRINT "TOP-BOTTOM FLIP":E$
300 FOR I = 1 TO 10: CALL 25078: FOR J = 1 TO
    1300: NEXT J: NEXT I
310 HOME : PRINT "RIGHT SIDE MIRROR OF LEFT"
    :E$
320 POKE 25014,234: POKE 25015,234: CALL 249
    36
330 FOR I = 1 TO 1500: NEXT
340 HOME : PRINT "BOTTOM MIRROR OF TOP"
350 PRINT D$;"BLOOD"PC$: POKE 25120,234: POKE
    25121,234: POKE 25193,234: POKE 25194,23
    4: CALL 25078
360 HOME : PRINT "BOTTOM MIRROR OF TOP":E$
370 FOR I = 1 TO 2000: NEXT
380 PRINT D$;"BLOOD"PC$;A$2000"
390 POKE 34,20: HOME : PRINT "HIRES TRICKS-T
    WO PAGES": POKE 34,21
400 PRINT "CURRENT PAGE 1":E$
410 FOR I = 1 TO 1500: NEXT
420 HOME : PRINT "CURRENT PAGE 2": PRINT D$:
    "BLOOD"PD$;A$4000": POKE 49234,1: POKE
    49237,1

```

```

430 FOR I = 1 TO 2000: NEXT
440 HOME : POKE 49236,1: POKE 49235,1: PRINT
    "2 OVERLAYED ONTO 1":E$
450 CALL 25246
460 FOR I = 1 TO 3000: NEXT
470 HOME : PRINT "1 MINUS 2": CALL 25282
480 FOR I = 1 TO 3000: NEXT
490 HOME : PRINT "PAGES 1-2 SCROLL LEFT":E$
500 FOR I = 1 TO 20: CALL 25388: NEXT
510 HOME : PRINT "PAGES 1-2 SCROLL RIGHT":E$
520 FOR I = 1 TO 20: CALL 25512: NEXT
530 HOME : PRINT "PAGES 1-2 SCROLL DOWN":E$
540 FOR I = 1 TO 96: CALL 25816: NEXT
550 HOME : PRINT "PAGES 1-2 SCROLL UP":E$
560 FOR I = 1 TO 96: CALL 25636: NEXT
570 TEXT : HOME : END
580 IF EF = 3 THEN TEXT : HOME
590 POKE 216,0: PRINT : PRINT "TROUBLE LOCAT
    ING FILE "PAS
600 VTAB 22: HTAB 1: PRINT "PRESS RETURN TO
    RE-ENTER NAMES": GET Z$: PRINT
610 ON EF GOTO 90,110,90
620 HOME : PRINT CHR$(4) LEFT$( "CATALOG",
    7 - 4 * ( PEEK (48896) = 76)): PRINT "PR
    ESS RETURN TO CONTINUE": GET Z$: PRINT
    : HOME : GOTO 100

```

END OF LISTING 2

KEY PERFECT 5.0  
 RUN ON  
 HIRES DEMO

```

=====
CODE-5.0  LINE# - LINE#  CODE-4.0
-----
33E34911  10 - 100  A4A5
1944E163  110 - 200  9F68
4BA40F41  210 - 300  772C
010839B4  310 - 400  8BF4
F39A5E73  410 - 500  7308
0AF25C95  510 - 600  7A3D
180BC777  610 - 620  2DE8
95791B85 = PROGRAM TOTAL = 0737

```