SUPERSET

FOR HEATH/ZENITH H/Z19 TERMINALS

AND H89/Z90 COMPUTERS

The TMSI "SUPERSET" adds a new dimension to your Heath/Zenith H/Z19 terminal or H/Z89 series computer. Benefits include a vastly increased character set, improved graphics via the NORCOM GT-PROM*, faster operation, rock stable "flicker-free" display, and reduced power consumption to run cooler.

The SUPERSET has all modes and escape sequences of a standard H/Z19 or H/Z89, so no software changes are needed. It also includes most features of the HUG/Watzman, Super-19, and Ultra-ROM including on-screen clock, new baud rates, and enhanced keyboard and display modes.

FEATURES

- 4 standard text fonts: H19, VT-100, GT-PROM*, and Greek. Optional SUPERFONT adds H29, IBM PC, superscripts, subscripts, and double-wide characters.
- 4 graphic sets: enhanced H19, DEC special graphics, GT-PROM*, and math; Optional SUPERFONT adds IBM PC, fineline, and 100x160 dot graphics.
- Magic Menus, for instant help and setup.
- True reverse screen (black characters on white screen).
- New attributes: blink, reverse video, or alternate character set.
- 4-mode function keys (normal/shift/ctl/ctl-shift).
- Transmit speed limiting for improved screen dumps.
- Improved interlace mode, with reduced flicker.
- Screen saver blanks screen after 15 min inactivity.
- Improved Hold Screen mode with SCROLL key.
- Superfast operation for quick response (up to 38K baud).
- On-screen digital clock/calendar. Optional SUPERCLOCK adds battery backup.
- Save/Restore/Swap terminal mode commands for rapid task switching.
- Native and transparent modes for special applications.
- It even does Windows! With SUPERCLOCK, save/restore/swap any part of screen with off-screen memory.

The kit contains a set of low-power CMOS ICs to replace the program, keyboard, and character generator ROMs; display RAMs; and a few other miscellaneous ICs. The new parts are significantly faster and reduce loading on the +8 volt supply by about 300mA. Installation takes under an hour, and soldering is required.

Since this kit operates in a manner not authorized by Heath/Zenith, it voids the warranty and may not work in all machines. But we have installed it successfully on many machines, and use it ourselves daily.

WARRANTY

This product carries a ninety (90) day limited warranty. If you have any questions or problems with this product or our service, please call or write us at the address below, and we'll do our best to help. Sorry, we can't accept collect calls.

For a period of ninety (90) days after delivery, we will repair or replace free of charge any part or product found defective in either materials or workmanship. We further warrant that the product will meet our published specifications for a period of ninety (90) days when assembled and used according to our published instructions. If a defective part or a design error on our part causes a part or product to fail during the warranty period, we will repair or replace it free of charge.

Our warranty does not cover, and we are not responsible for damage caused by abuse, incorrect assembly, defective tools, or unauthorized modifications. Our warranty does not include reimbursement for inconvenience, loss of use, or customer labor.

This warranty includes only parts and products sold by us, and does not cover equipment to which our parts or products may be connected. We are not responsible for incidental or consequential damages. Some states do not allow exclusion of incidental or consequential damages, so this may not apply to you.

TMSI c/o Lee A. Hart
28612 Middle Crossing Road
Dowagiac, MI 49007

(616) 782-3980

GT-PROM is included under license from NORCOM, 9630 Hayes, Overland Park KS.

TABLE OF CONTENTS

PARTS SUPPLIED

Check to be sure you have all the following parts. Don't remove the ICs (integrated circuits) from their protective packaging until ready for use.

Qty	Part no.	Description
1	101.402	SUPERSET Program ROM
1	101.412	SUPERSET Character ROM
	(or 101.430)	(SUPERFONT Character ROM)
1	101.422	SUPERSET Keyboard ROM
(1)	(MK48T02)	(SUPERCLOCK)
4	2114AL-2	1Kx4 fast static RAM
1	74F138	1 of 8 decoder, Fast TTL
1	74F00	quad 2-input NAND gate, Fast TTL
1	701.402	24-to-28 pin socket adapter
1	701.412	24-to-28 pin socket adapter
	(or 701.430)	(24-to-28 pin socket adapter for SUPERFONT)
1	330 pf	disk ceramic capacitor
1	4.7uF 25vdc	tantalum capacitor
1	P3.1	3-pin .025" header
1	P6.1	6-pin .025" header
1	12" piece	hook-up wire
1	3" piece	braided solder-wick

INTRODUCTION

The SUPERSET is an enhanced set of parts for the H/Z-19 terminal and the H/Z-89/90 computers. It includes ALL features of the standard H/Z19 terminal (Heath, VT-52, and ANSI modes) plus the following:

- Selectable character sets: A standard H19 has one 128-character font. The SUPERSET expands this to 4 fonts of 128 characters each (512 total).

  Font 1 is the original Heath set, with minor improvements such as 1/4 dot and 3/4 dot area fills, missing 2x2 graphic character, and improved appearance when using text characters for graphic applications (drawing boxes with :|-/ etc).

  Font 2 is the DEC VT-100 with Special Graphics set, useful for thin-line drawings and for displaying control codes.

  Font 3 is the NORCOM GT-PROM which adds a boldface text font, and graphics characters to plot dots in a 2x5 matrix within a character cell (160x125 dot resolution graphics).

  Font 4 replaces the graphics characters with a set of special characters including math symbols and Greek alphabet.

The optional SUPERFONT adds 4 additional fonts (for a total of 8). Additionally, each font has 256 characters instead of 128, for a total of 2048 displayable characters.

Font 5 is the Heath H/Z29 terminal character set. It has the standard H19 set plus superscripts, subscripts, Greek letters, and fine-line graphics.

Font 6 is the VT-100, but with both single- and double-width character sets (40 or 80 characters per line). In transparent mode or with graphics enabled, it displays mnemonics for all control codes (ᴸꜰ for Line Feed, etc.).

Font 7 is APA-graphics (All-Points-Addressable). Each character cell is broken into 8 parts, 2 dots wide by 4 dots high. Thus pictures can be drawn to a resolution of 100 pixels high by 160 pixels wide.

Font 8 is the IBM PC monochrome display adapter character set. Characters are formed in a 7x9 dot matrix (rather than 5x7), and include a variety of special symbols unique to the PC.

- Cuts heat and power consumption. The SUPERSET reduces TLB power consumption by 300 mA for a cooler and more reliable machine.

- Faster operation. Supports up to 38400 baud without handshaking, and a 3 MHZ TLB clock. Previously slow routines like cursor positioning, character insert and delete, and line and screen erase are over 3:1 faster. Screen updates up to 4 times quicker!

- Flicker-free display. The SUPERSET includes our flicker-free kit; no "black dashes" when characters are scrolled or written rapidly on the screen.

- Digital clock/calendar display on the 25th line. The 24-hour clock may be set, read, or reset to 00:00:00. The calendar can be set or read to display the month, date, and year. Time and/or date can be sent to your computer with a single keystroke. The clock/calendar display may be turned on or off, and normal 25th line operation is not affected. The clock is accurate to 10 sec/day, and unaffected by reset; the optional SUPERCLOCK improves accuracy and adds battery backup.

- New display options. True reverse screen mode (white screen with black characters). Improved interlace mode with reduced flicker. Screen saver automatically blanks screen after 15 minutes without activity, and re-enables it if any key is struck or data is received.

- Magic Menus. Control-ESCape calls a series of full-screen menus that outline terminal operation and ESC sequences. Select the desired operation from the menu. When finished, the screen is restored to its original state.

- Local function keys. CONTROL+function keys perform immediate local functions; position cursor, clear screen, select fonts, set modes, display menus, send date or time, etc.

- More attributes. Instead of just reverse video, the attribute (8th bit) can select reverse video, blinking, or with SUPERFONT, a character from the upper half of an extended 256-char set.

- New cursor options. Select any combination of blinking- or non-blinking; and block, underline, or no cursor.

* Enhanced transmit functions. In addition to transmit page and 25th line, the SUPERSET adds transmit line, character, clock, and calendar. Transmit speed limiting holds speed to 60 chars/sec to allow screen dumps to slow devices like printers or BASIC. A cursor shows characters as they are sent.

* Improved terminal operation. Options for Half and Full duplex, 7- or 8-bit word length. and hardware and software handshaking. New baud rates include 75, 134.5, and 38,400 baud.

* Native mode keyboard. Each key produces a unique 8-bit code instead of multi-character escape sequences. Function keys have unique codes for normal, shifted, control, and control-shift combinations.

* Transparent mode. Each 8-bit code from 0 to 255 produces a unique visible character, so ESC sequences aren't needed for graphics and reverse video. With SUPERFONT, a single code can display any of 256 unique characters.

* Save and Restore Modes. ESC sequences to save and restore current terminal state; normal/shift/alternate keypad, normal/reverse video, text/graphics, cursor type, etc. A "soft" ESC z option resets the terminal to the last saved state, rather than to the DIP switch settings.

* Improved ANSI mode. Much faster, more VT-100 ESC sequences.

SPECIAL CASES

These instructions assume you are installing the SUPERSET in a standard, unmodified H/Z19 terminal or H/Z89 computer. This kit affects ONLY the Terminal Logic Board (TLB); don't worry about any changes you have made elsewhere in the computer.

If you have changed your TLB clock speed, it should be put back to the original 2MHz. Though the SUPERSET will change this to 3MHz, it is best not to mix up two different ways of doing the same thing.

Since the SUPERSET replaces your existing ROMs, you cannot use it with the HUG/Watzman, Ultra-ROM, Super-19, or Font-19 ROMs. But most of the features of these parts are already provided. The SUPERSET also will not work with the Cleveland Codonics Imaginator or Northwest Digital Graphics-Plus boards.

The SUPERSET is derived from the standard H19 ROM and HUG/Watzman ROM, so it includes all of the features and ESC sequences of these parts. As a result, it is 99% compatible with software that works with either of these parts. The remaining 1% includes programs that send previously illegal ESC sequences, or take advantage of bugs in the earlier ROMs.

In addition, the SUPERSET includes most of the features of the Ultra-ROM and Super-19. But it is not software compatible with these ROMs; it is impossible to choose a set of ESC sequences compatible with all these parts. When choosing an ESC sequence for a given function, the SUPERSET's order of preference was:

1st: use the same ESC sequence as a stock Heath H/Z-19
2nd: same as HUG/Watzman
3rd: Heath H/Z-29
4th: Super-19
5th: Ultra-ROM
last: invent a new ESC sequence

INSTALLATION

Installation is easiest if you have an H/Z19 terminal. Just follow the numbers in each illustration. Work carefully, and mark off each step as you complete it. Check your work, or better yet, have someone else check it for you -- mistakes suddenly get very expensive after you've turned on the power!

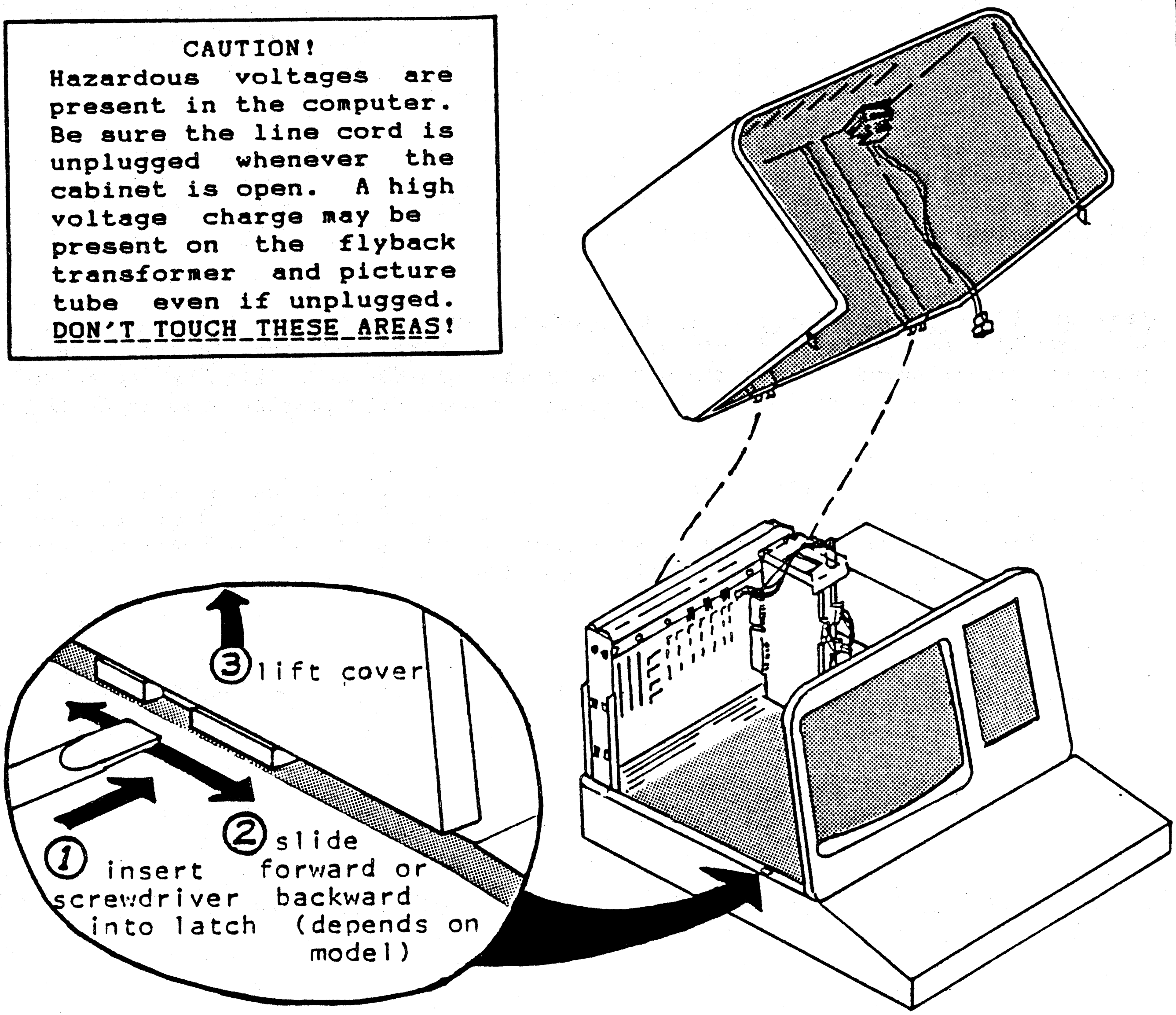
If you have an H/Z89 series computer, things are more involved. But there is an H19 terminal hiding under all that clutter. There are so many variations between computers that we can only give general instructions to remove and install the CPU and accessory boards. Call us if you need help, or refer to the appropriate manuals.

This is NOT a project for a beginner. Installation requires skill in soldering, counting pins on integrated circuits, and locating and cutting traces on a printed circuit board. If you did not build your H19/H89 as a kit or have no experience as an electronics technician, please find someone else to do the installation.

TMSI can install your SUPERSET for you. Please call first to get an RMA (Return Material Authorization) number. This lets us plan ahead to avoid long delays. Installation is $25 if you send us your working TLB; or $50 (plus parts) if you tried to do it yourself, broke the TLB, and THEN send it in.

OPENING THE CABINET

CAUTION!
Hazardous voltages are present in the computer. Be sure the line cord is unplugged whenever the cabinet is open. A high voltage charge may be present on the flyback transformer and picture tube even if unplugged. DON'T TOUCH THESE AREAS!



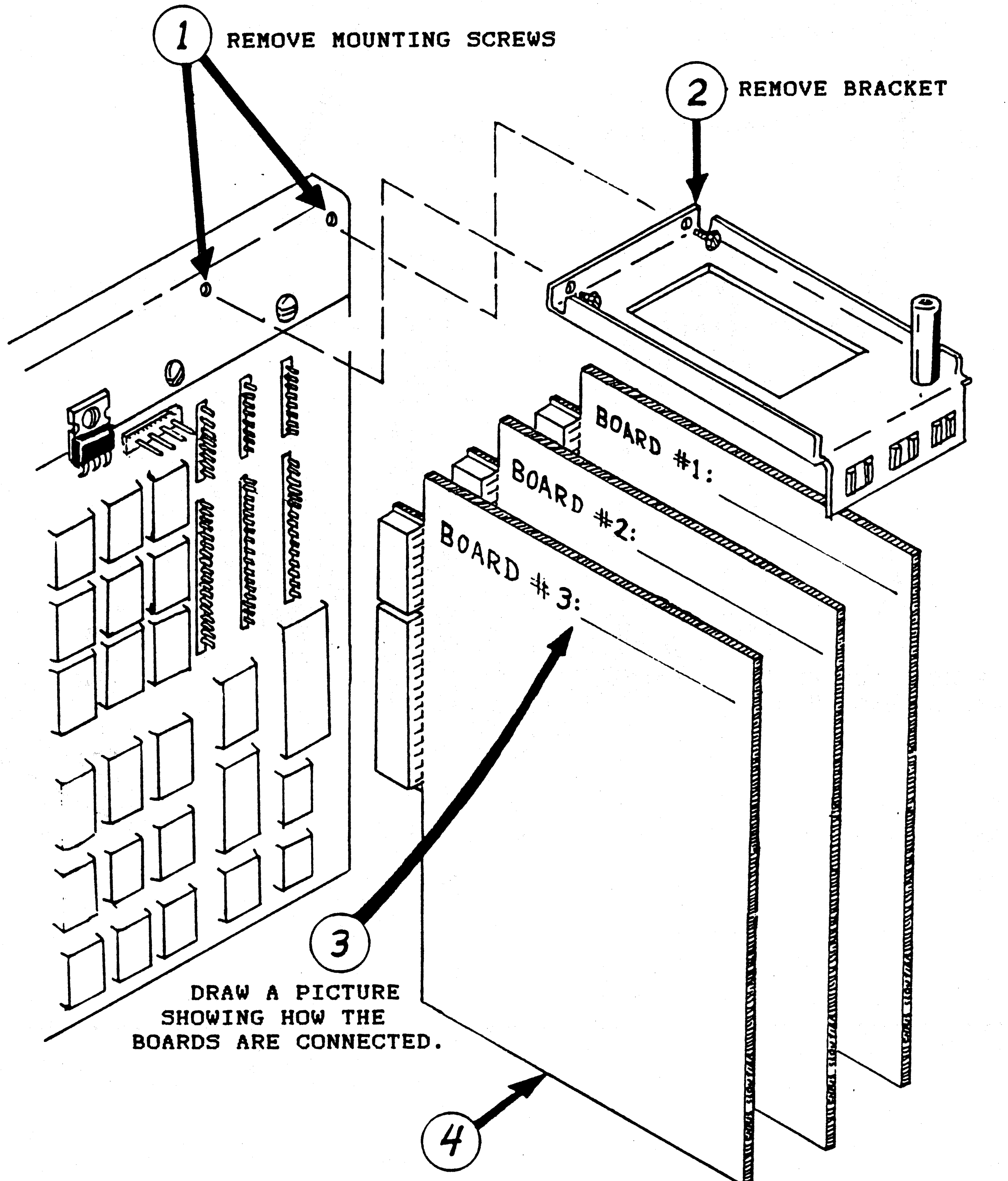
1. Turn the machine off and unplug the power cord.

2. Find the two latches on the cover (one on each side). Open as shown above.

3. Tilt the cover back, and unplug the fan's power connector.

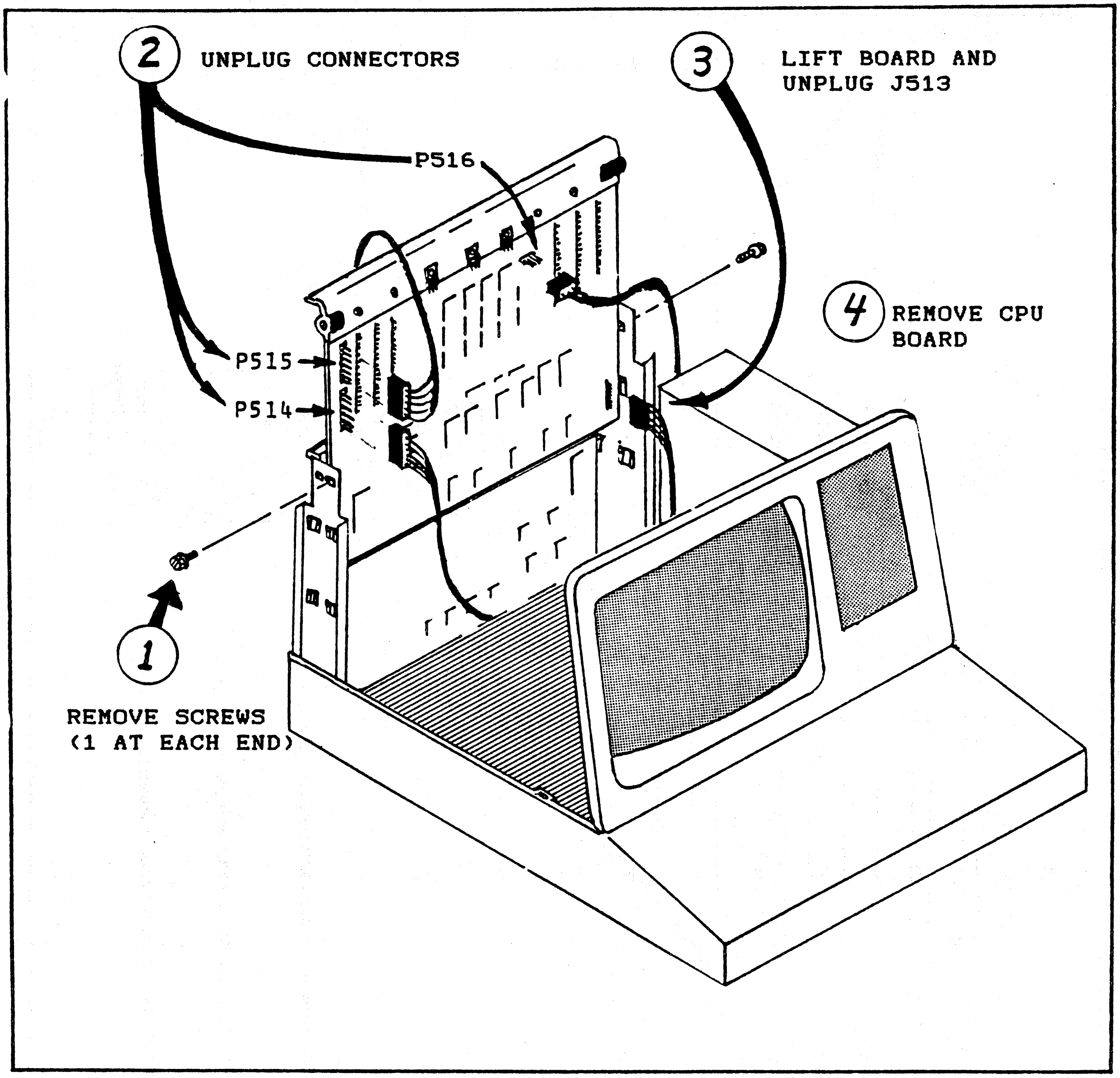
4. Slide the cover off the machine, and set it aside.

UNPLUG I/O BOARDS (COMPUTERS ONLY)



UNPLUG BOARDS FROM THE CPU BOARD, INCLUDING ANY CABLES TO THE CPU BOARD. BOARDS NEED NOT BE REMOVED FROM MACHINE.

REMOVE CPU BOARD (COMPUTERS ONLY)



Notes on the SUPERFONT

1. INSTALLATION - To install the Superfont, replace pages 10 and 18 of the Superset manual with the enclosed pages 10 and 18.

2. OPERATION - The Superfont adds four new fonts (for a total of 8). The new fonts can be selected with local function keys ctl-shift-f1 thru ctl-shift-f4, or by the ESC sequences shown in the Superset manual.

3. MAGNOLIA CP/M - If you are running Magnolia CP/M, the system may "hang" when the new fonts are selected. This is because the Superset uses the DTR line to select the extra fonts. Heath CP/M ignores it, but Magnolia CP/M expects the DTR line to always be high, and thinks the terminal got turned off if DTR ever goes low.

   To fix this, bend out pin 4 or 5 of the 1488 IC on the TLB (U453 on newer machines, U452 on older machines). Connect a wire between this pin and ground to force the DTR signal on P404 pin 14 (the violet wire) high.

   Magnolia CP/M implements hardware handshaking on the RTS line, so the Superset's software handshaking can be disabled.

4. AUTO-BAUD RATE SELECTION - Because of the Superset's additional baud rates and new ESC sequences, software that tries to detect the terminal's baud rate by arcane means may not work. In particular, Dean Gibson's boot code for the Quikstor Winchester, and the monitor ROM he wrote for KRES do not work. A special version of the Superset program ROM is available if you have this problem. It contains BOTH the Superset plus the stock H19 code, and allows an external switch to select between them. Contact Lee Hart for details.

5. MEMORY PROBLEMS - The Superset was designed to use CMOS memory chips (identified by a "C" in their part number). They are faster, and take a tiny fraction of the power used by the original NMOS parts.

   Unfortunately, they are also made in Japan. Thanks to our government's restrictions on Japanese imports, I have been forced to substitute older NMOS parts. As a result, the 300mA power savings mentioned in the manual is more like 100mA.

INTEGRATED CIRCUIT HANDLING

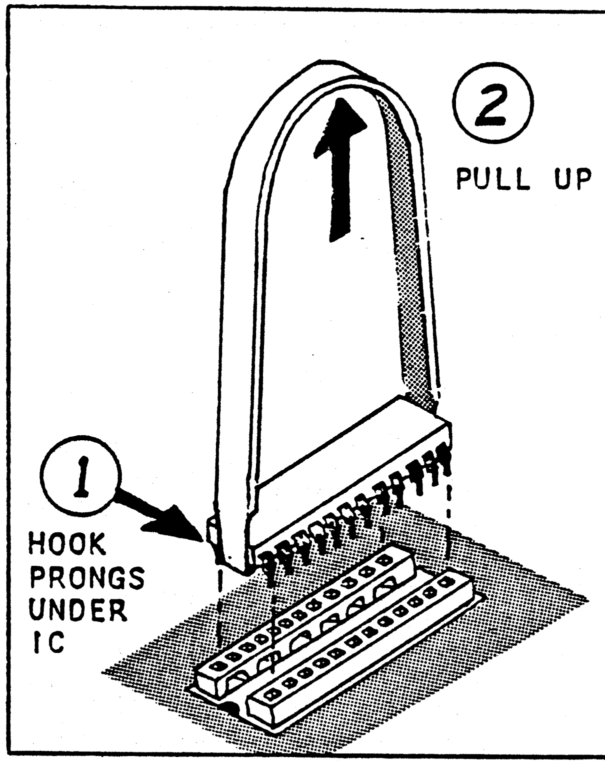
Integrated circuits (ICs) are rugged and reliable components, but they can be easily damaged by static electricity or improper handling. Use the following proceedures when installing or removing ICs:

1. Remove the IC with an IC extraction tool (or a small screwdriver). Hook the tool over the ends of the IC and pull straight up, rocking slightly from end to end.

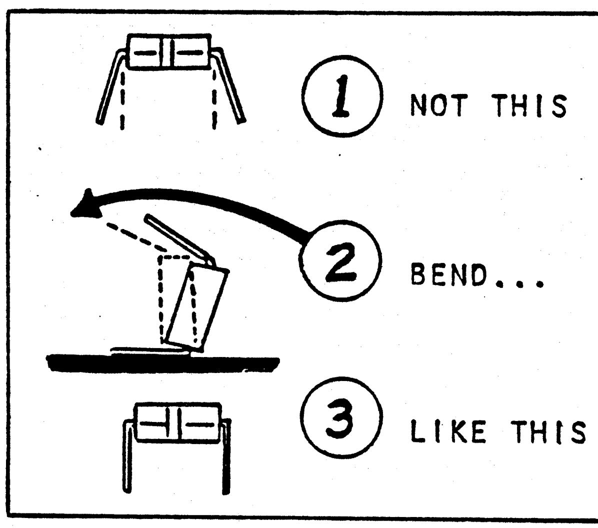
2. Hold on to the IC until it is either replaced on a board or stored safely. ICs can be safely stored in special IC tubes, black conductive foam, or wrapped in aluminum foil.

3. Straighten any bent pins on the IC. The pins should be parallel to each other and at right angles to the case. Some ICs may have their pins spread out slightly (see illustration). If so, align the pins by gently pressing against a table top and bending as shown.

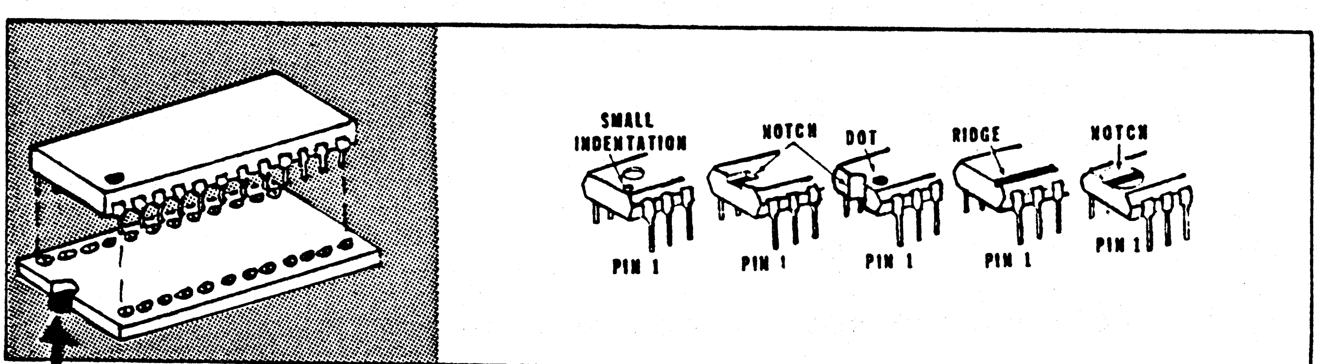
4. When installing the IC, align the notch and/or dot with the index mark on the board (see below). Be sure all the pins enter the holes of the socket, and then press the IC into its socket.



REMOVE THE IC

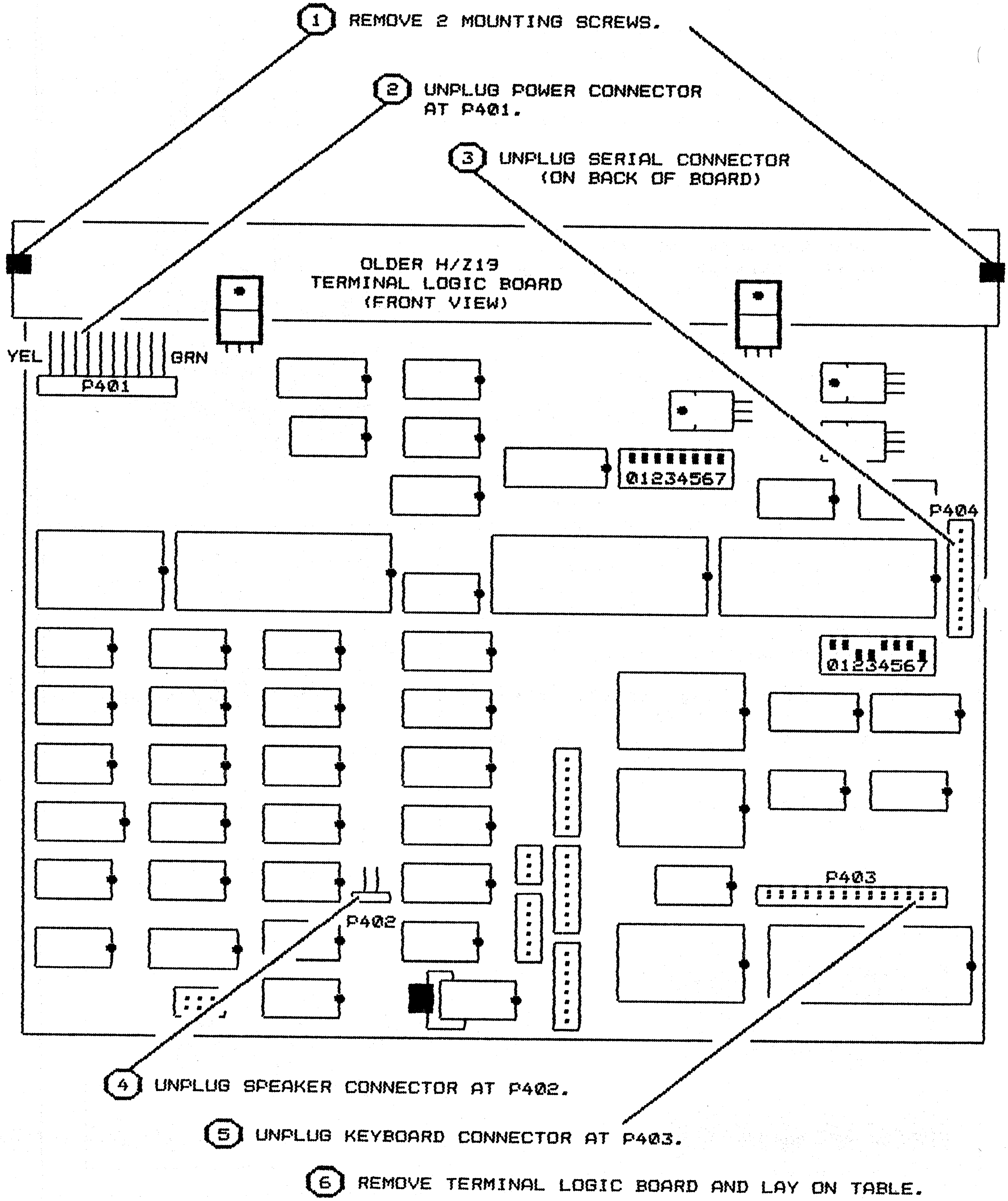


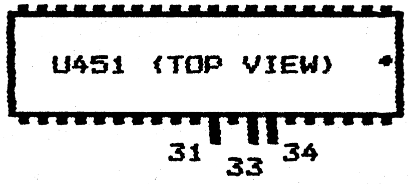
STRAIGHTEN THE LEADS



INSTALL IC WITH PIN 1 END AT INDEX MARK

USE THESE INSTRUCTIONS FOR AN OLDER H/Z19 WITH A 10-HOLE CONNECTOR AT P401.
FOR AN H/Z19A WITH AN 11-HOLE CONNECTOR AT P401, GO TO PAGE 17.

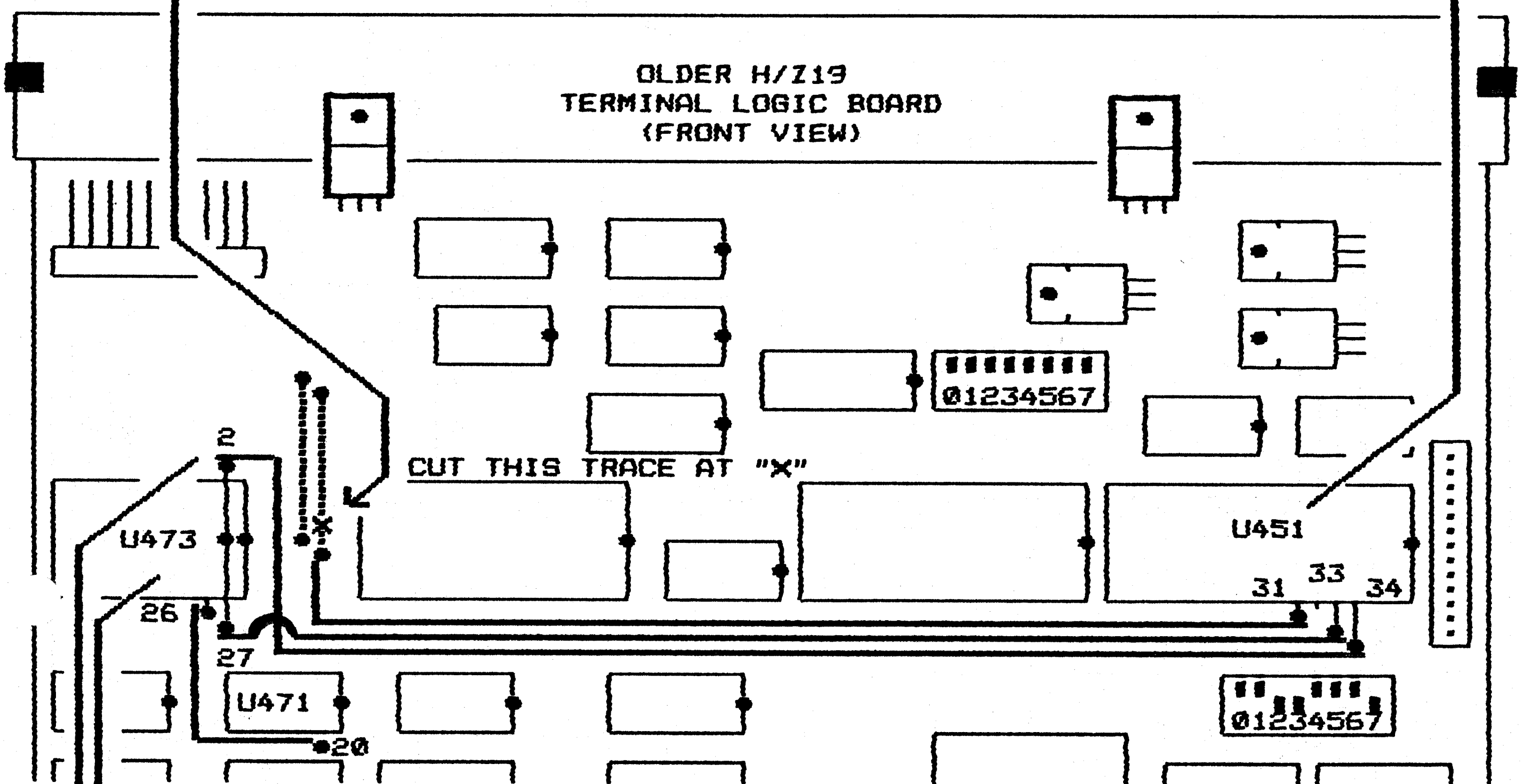




1. REMOVE THE IC AT U451.
BEND OUT PINS 31, 33, AND 34.
REPLACE U451 SO THESE PINS
DO NOT ENTER THE SOCKET.

2. A. FIND THE TWO FOIL TRACES JUST TO THE RIGHT OF U473.
CUT THE <u>RIGHT</u> TRACE NEAR THE BOTTOM HOLE AT THE "X".

B. SOLDER A WIRE FROM U451 PIN 31 TO THE HOLE AT THE
LOWER END OF THE TRACE YOU JUST CUT.

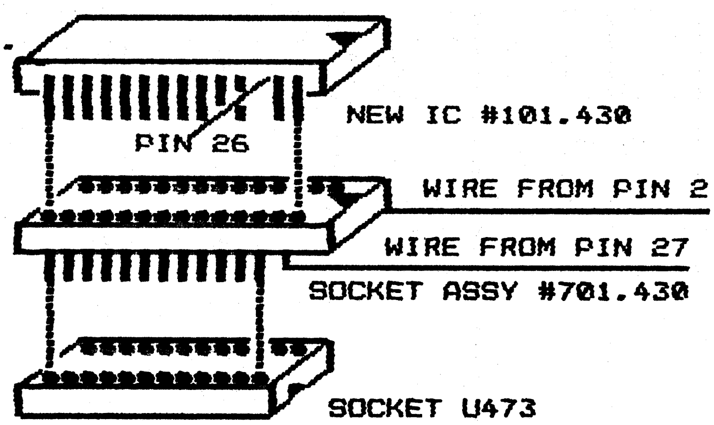


3. A. REMOVE IC #444-29 FROM U473.

B. INSTALL SOCKET #701.430
AT U473. THE SIDE WITH THE
WIRES AND EXTRA PINS GOES
ON THE <u>RIGHT</u> END AS SHOWN.

C. BEND OUT PIN 26 OF THE
"SUPERFONT" IC #101.430

D. INSTALL "SUPERFONT" IC
AT U473 SO PIN 26 DOES
NOT ENTER THE SOCKET.



4. SOLDER THE WIRE FROM PIN 2 OF SOCKET ASSY #701.430 TO U451 PIN 34.

5. SOLDER THE WIRE FROM PIN 27 OF SOCKET ASSY #701.430 TO U451 PIN 33.

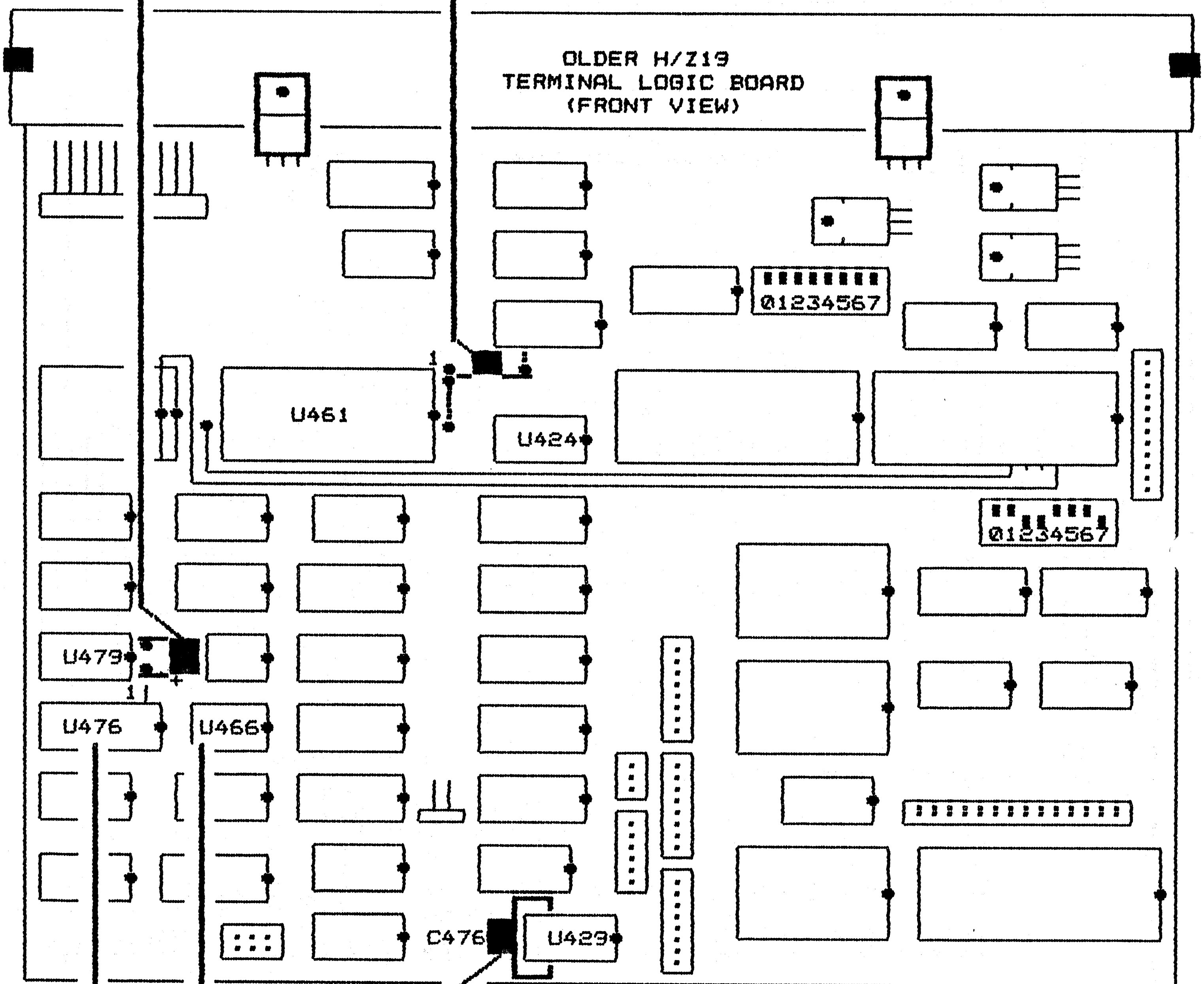
6. SOLDER ONE END OF A WIRE TO THE BENT-OUT PIN 26 OF IC #101.430.
SOLDER THE OTHER END TO THE PLATED-THRU HOLE JUST LEFT OF THE
PRINTED "20" JUST BELOW U471.

(1) REPLACE THE CAPACITOR NEAR U479 WITH THE 10UF CAPACITOR SUPPLIED. THE END MARKED "+" GOES IN THE BOTTOM HOLE, TOWARD U476 AS SHOWN.

(2) SOLDER A 330PF CAPACITOR INTO THE HOLES SHOWN:

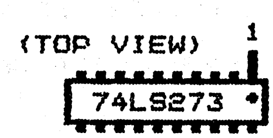
A. THE RIGHT HOLE IS JUST ABOVE THE "U" IN THE SILKSCREENED LEGEND "U424".

B. THE LEFT HOLE (GROUND) IS JUST RIGHT OF THE SILKSCREENED "1" FOR PIN 1 OF U461.



(3) LOOK FOR C476 (470 PF; MAY BE ON BACK OF BOARD). IF FOUND, REMOVE IT.

(4) REMOVE THE 74LS00 AT U466. REPLACE IT WITH THE 74F00 SUPPLIED.

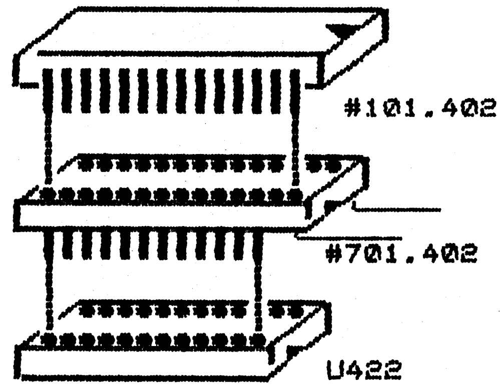


(5) REMOVE 74LS273 AT U476. BEND OUT PIN 1 AS SHOWN, AND REPLACE IT SO PIN 1 DOES NOT ENTER SOCKET.

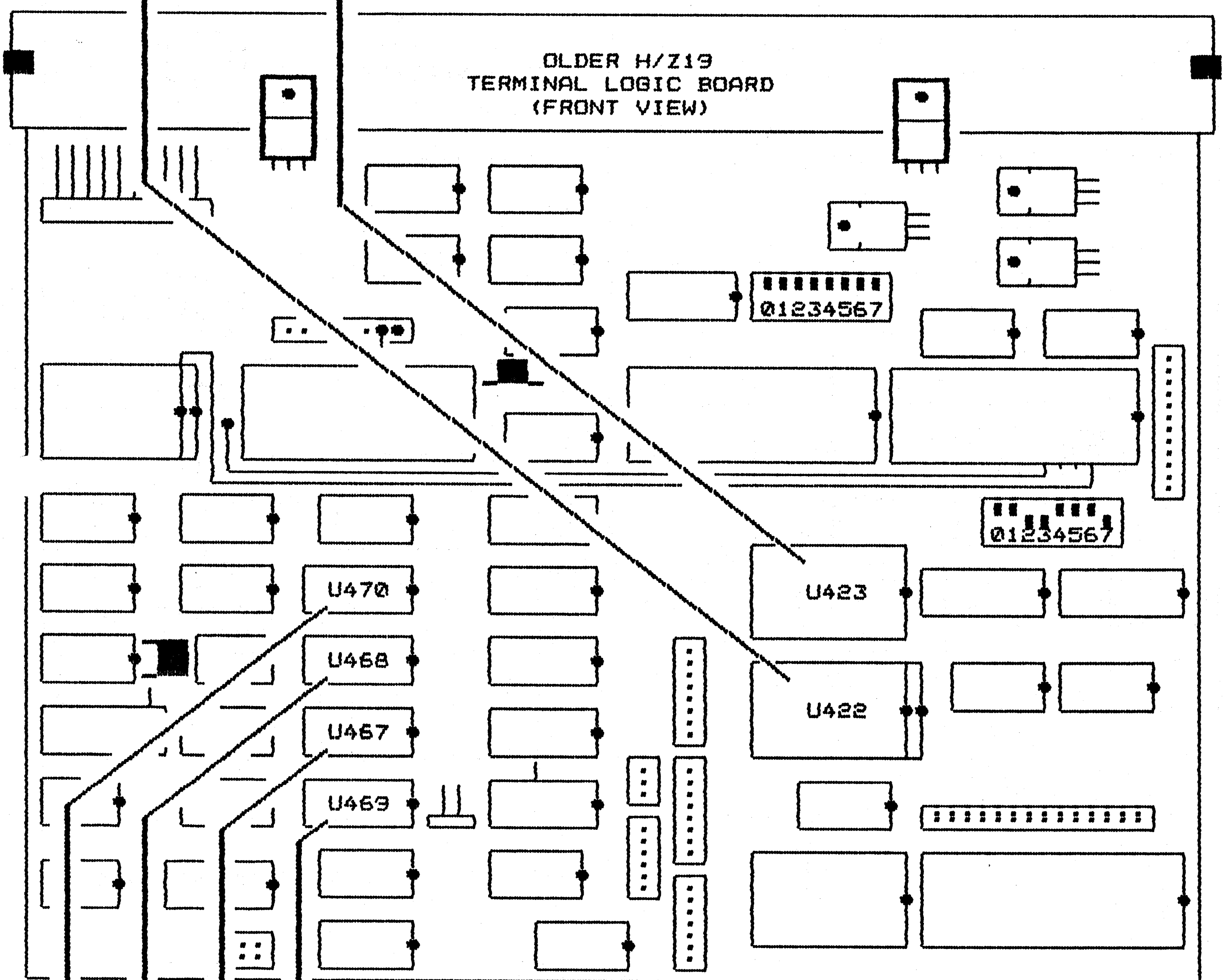
(1) A. REMOVE 444-46 (OR 444-53) AT U422.

B. INSTALL SOCKET #701.402 AT U422. THE END WITH THE EXTRA WIRES GOES ON THE RIGHT SIDE AS SHOWN.

C. INSTALL "PROGRAM" IC #101.402 AT U422.



(2) REMOVE ANY IC AT U423.



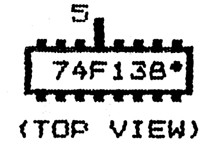
(3) REMOVE 2114 AT U469. REPLACE WITH 2114AL-2.

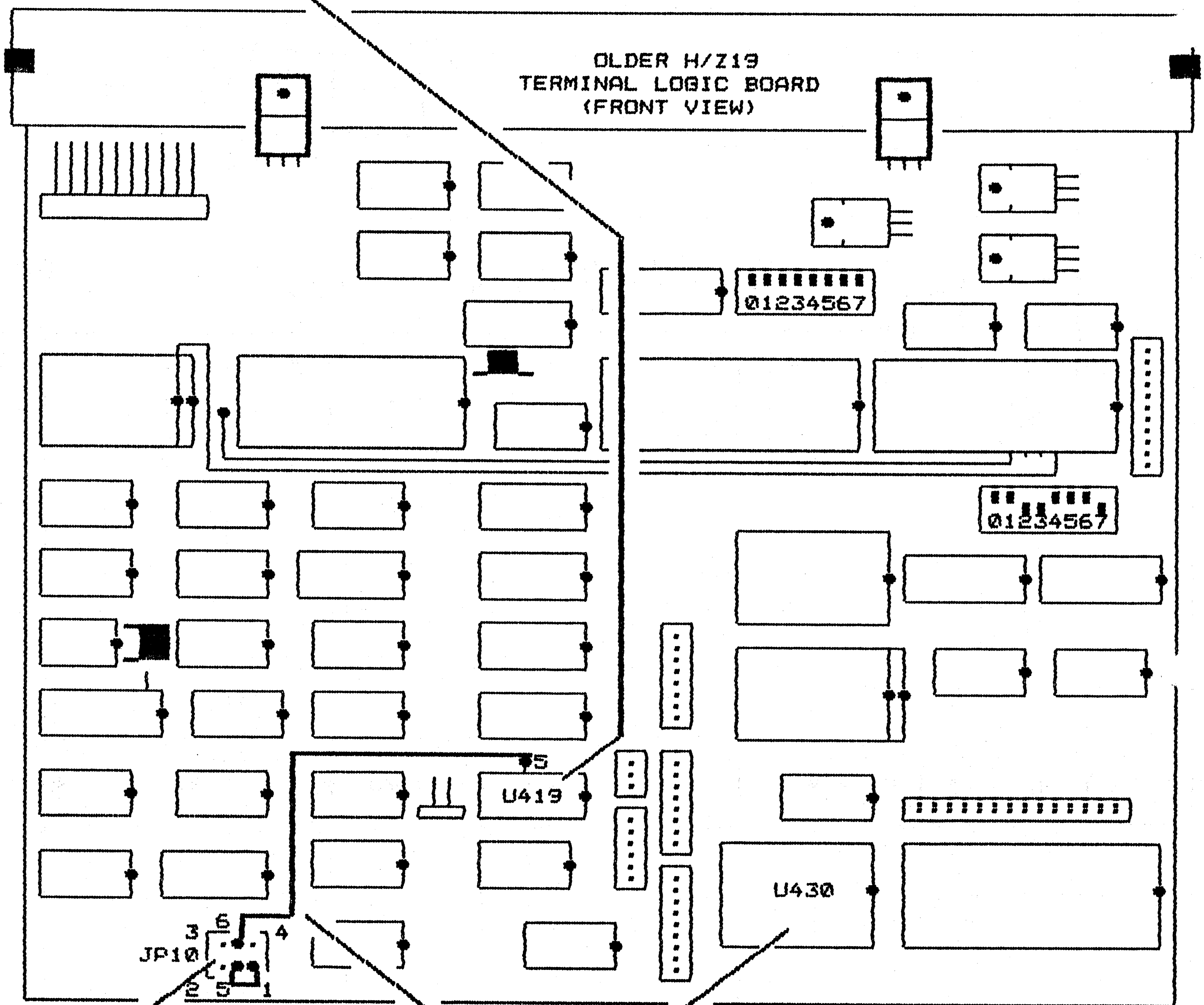
(4) REMOVE 2114 AT U467. REPLACE WITH 2114AL-2.

(5) REMOVE 2114 AT U468. REPLACE WITH 2114AL-2.

(6) REMOVE 2114 AT U470. REPLACE WITH 2114AL-2.

(1) REMOVE 74LS138 AT U419. FIND THE NEW 74F138, AND BEND PIN 5 OUT AS SHOWN. INSTALL IT AT U419 SO PIN 5 DOES NOT ENTER THE SOCKET.





(2) REMOVE 444-37 AT U430. REPLACE IT WITH "KEYBOARD" IC #101.422.

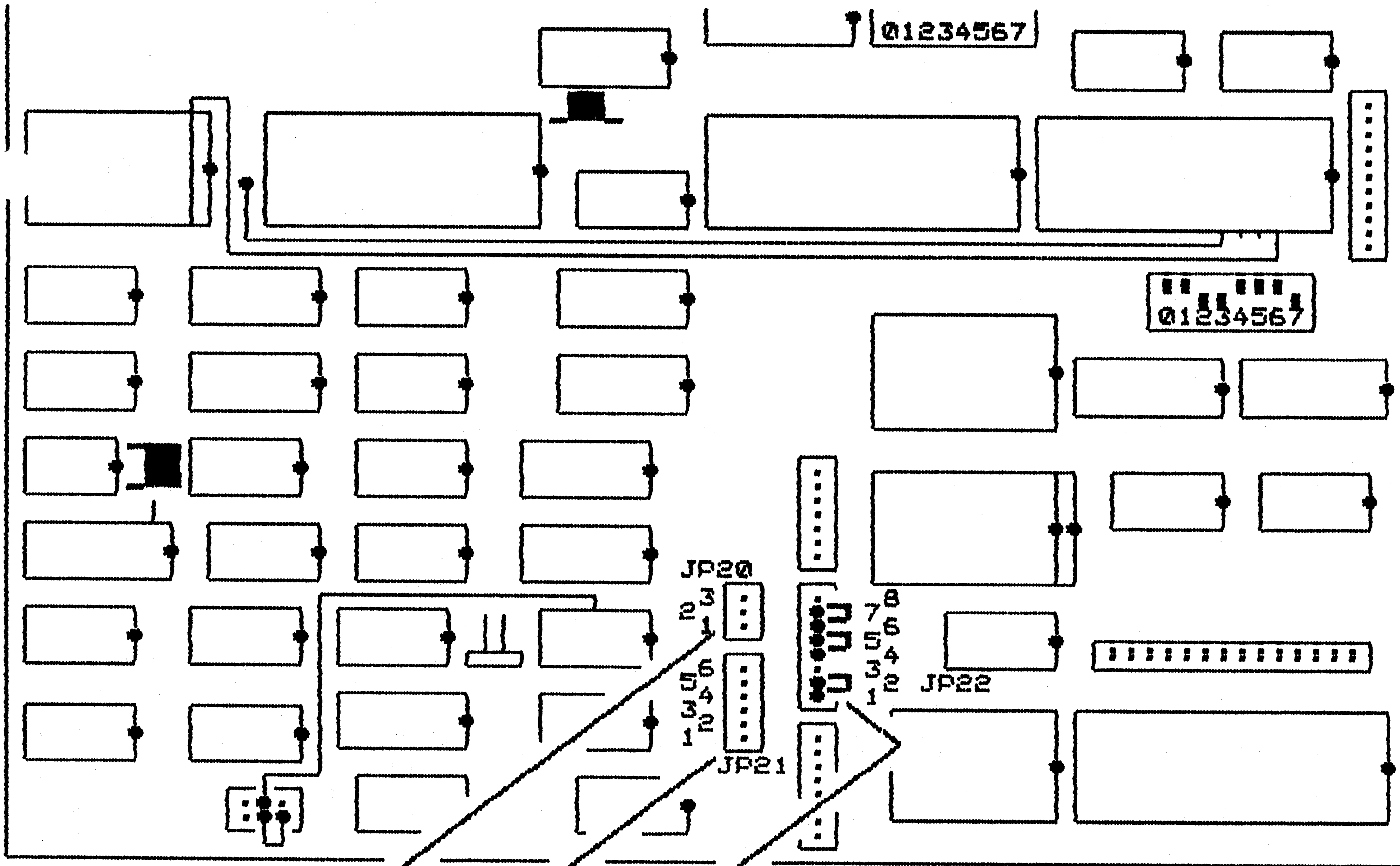
(3) A. SOLDER ONE END OF A 5" PIECE OF WIRE TO PIN 6 OF JP10.

B. SOLDER THE OTHER END OF THIS WIRE TO THE BENT-OUT PIN OF U419 (74F138 PIN 5).

C. POSITION THIS WIRE FLAT AGAINST THE BOARD.

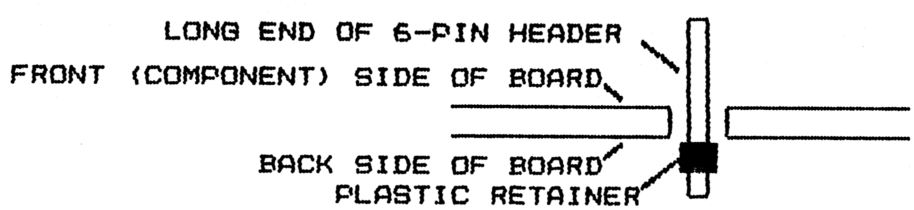
(4) A. CUT AND REMOVE THE CONNECTION BETWEEN JP10 PINS 2-5. THERE MAY BE A PIECE OF WIRE, FOIL ON THE BOARD, OR BOTH.

B. SOLDER A JUMPER WIRE BETWEEN JP10 PINS 1-5.



1. A. CAREFULLY CUT AND REMOVE ANY JUMPERS BETWEEN JP22 PINS 2-3, 3-4, AND 7-8. THERE MAY BE A PIECE OF WIRE, FOIL ON THE BOARD, OR BOTH.
   B. SOLDER A JUMPER BETWEEN JP22 PINS 1-2.
   C. " " " JP22 PINS 4-5.
   D. " " " JP22 PINS 6-7.

2. A. CUT AND REMOVE ALL JUMPERS BETWEEN ADJACENT PINS AT JP21.
   B. CLEAR SOLDER FROM THE HOLES WITH THE SOLDER WICK SUPPLIED.
   C. INSTALL A 6-PIN HEADER IN THE HOLES AT JP21 (SEE BELOW). THE PLASTIC BODY GOES ON THE BACK SIDE OF THE BOARD.
   D. SOLDER THE PINS TO THE BOARD FROM THE FRONT SIDE.



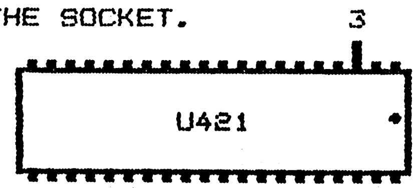
3. LIKEWISE, REMOVE ANY JUMPERS AT JP20 BETWEEN PINS 1-2 OR 2-3. SOLDER A 3-PIN HEADER INTO THE HOLES AT JP20.

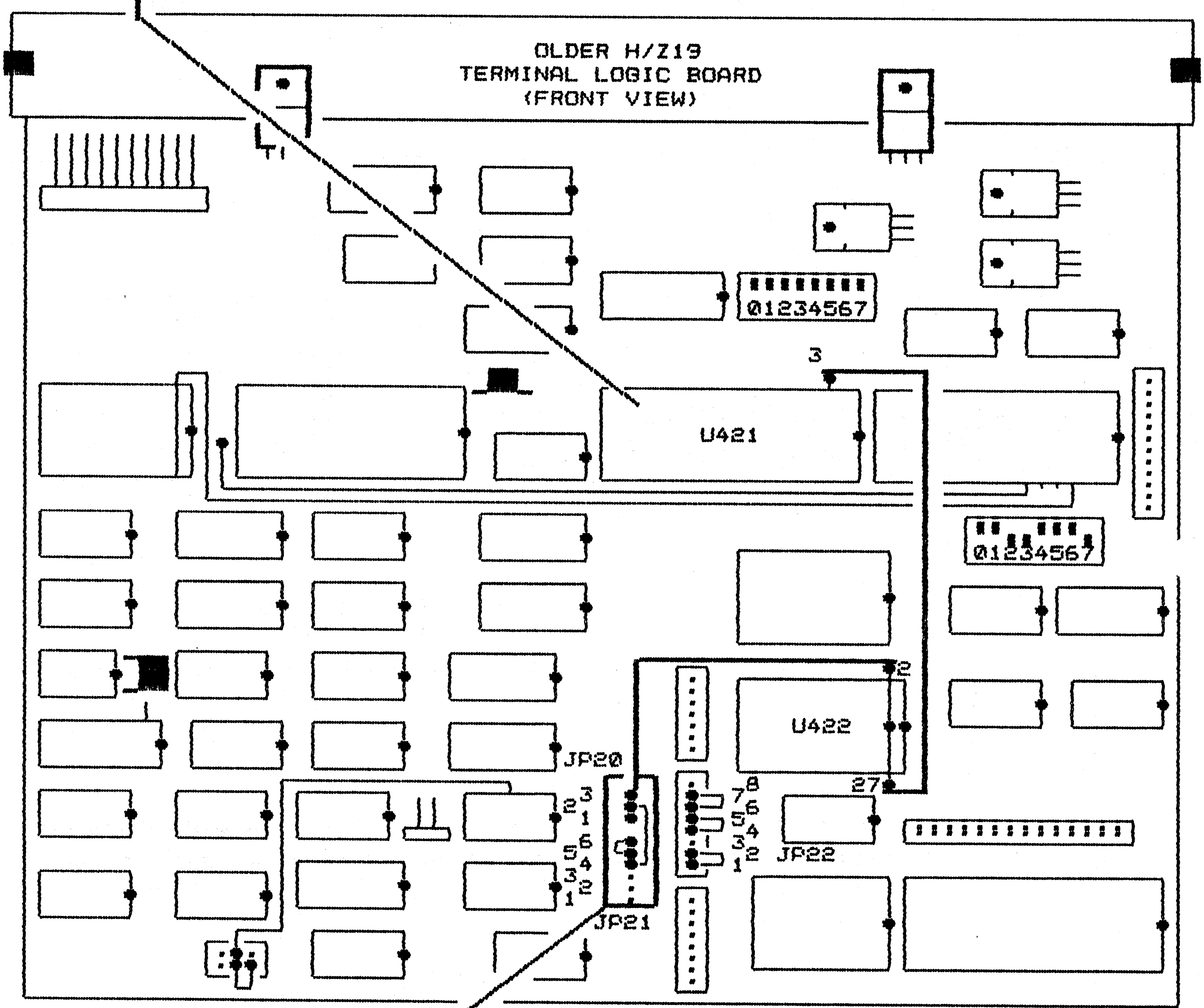
4. IF YOU HAVE AN OHMMETER, CHECK TO BE ABSOLUTELY SURE THERE ARE NO SHORTS BETWEEN ADJACENT PINS EXCEPT AS SHOWN ABOVE.

(1) A. REMOVE THE Z80 IC AT U421, AND BEND OUT PIN 3 AS SHOWN.

B. INSTALL U421 SO PIN 3 DOES NOT ENTER THE SOCKET.

C. SOLDER THE WIRE FROM SOCKET U422 TO U421 PIN 3.



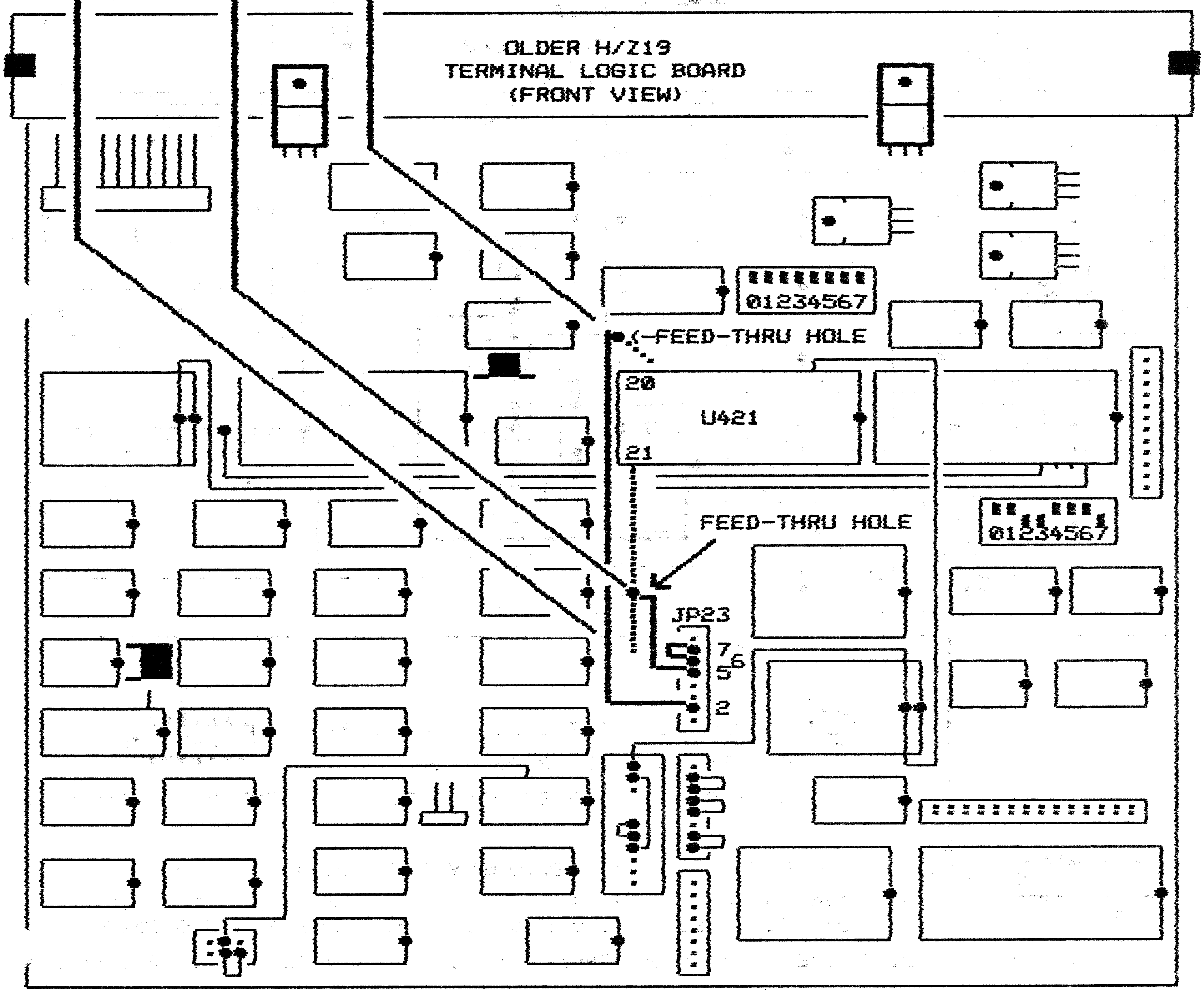


(2) PLUG THE 10-PIN CONNECTOR FROM SOCKET U422 ONTO JP20 AND JP21 AS SHOWN.

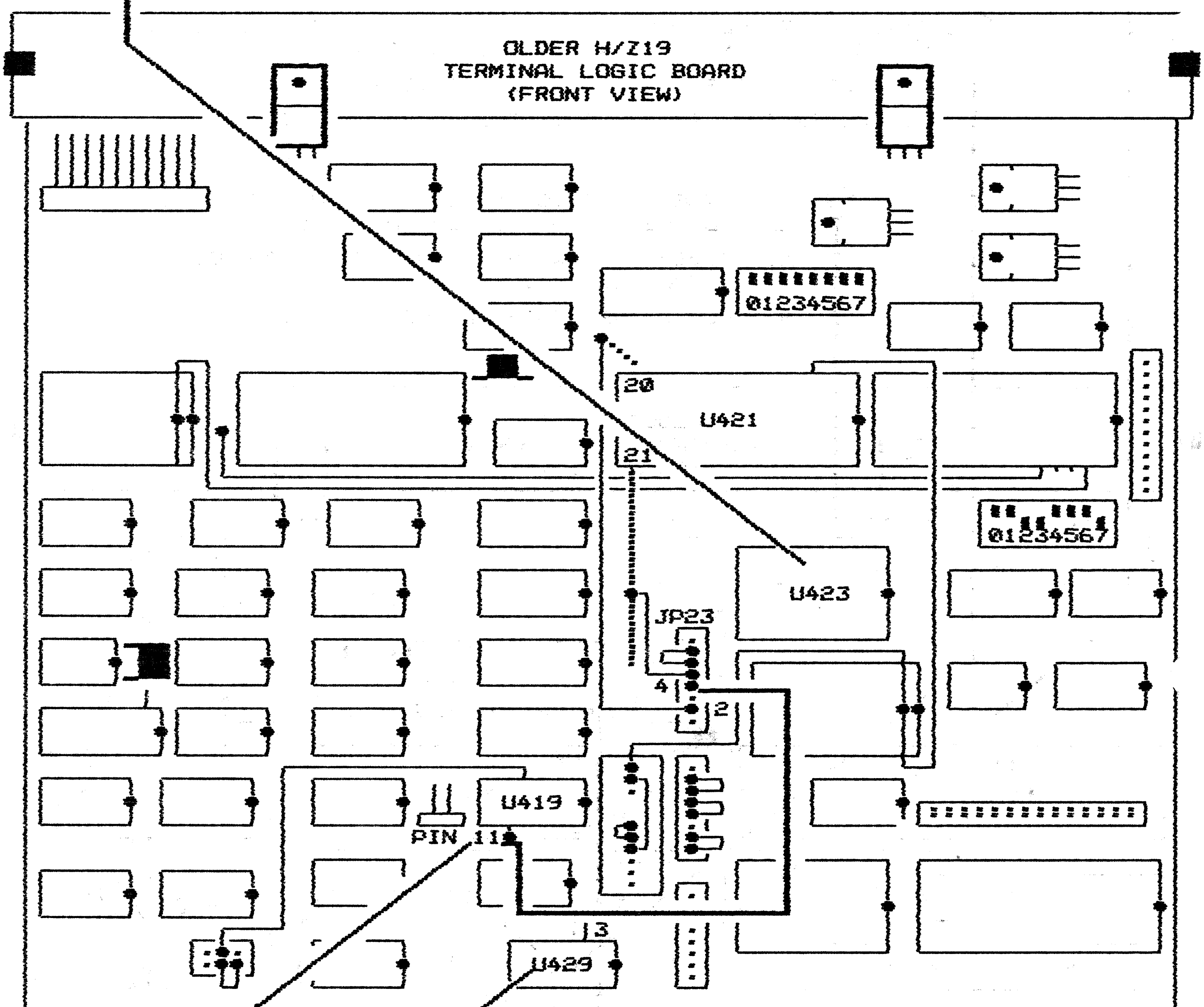
1. A. AT JP23, REMOVE ANY JUMPERS (FOIL OR WIRES) BETWEEN ADJACENT PINS.
   B. SOLDER A JUMPER WIRE BETWEEN JP23 PINS 6-7.

2. FIND THE TRACE FROM U421 PIN 21 (LOWER LEFT CORNER). FOLLOW IT DOWN 2" TO A FEED-THRU HOLE.
   A. SOLDER ONE END OF A 2" WIRE INTO THIS HOLE.
   B. SOLDER THE OTHER END OF THIS WIRE TO JP23 PIN 5.

3. FIND THE TRACE COMING OUT FROM UNDER THE UPPER LEFT CORNER OF U421, NEAR PIN 20.
   A. SOLDER A 5" WIRE TO THE FEED-THRU HOLE AT THE END OF THIS TRACE.
   B. SOLDER THE OTHER END OF THIS WIRE TO JP23 PIN 2.



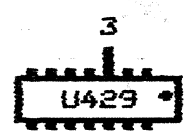
(1) INSTALL THE MK48T02 SUPERCLOCK IC AT U423. BE SURE THAT PIN 1 (MARKED WITH A RED DOT) IS IN THE UPPER RIGHT CORNER.

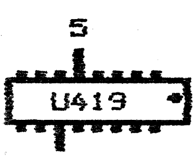


(2) A. REMOVE THE 74LS32 IC AT U429.

B. BEND OUT PIN 3 AS SHOWN.

C. REPLACE U429 SO PIN 3 DOES NOT ENTER THE SOCKET.





(TOP VIEW)

(3) A. REMOVE THE 74S138 PREVIOUSLY INSTALLED AT U419.

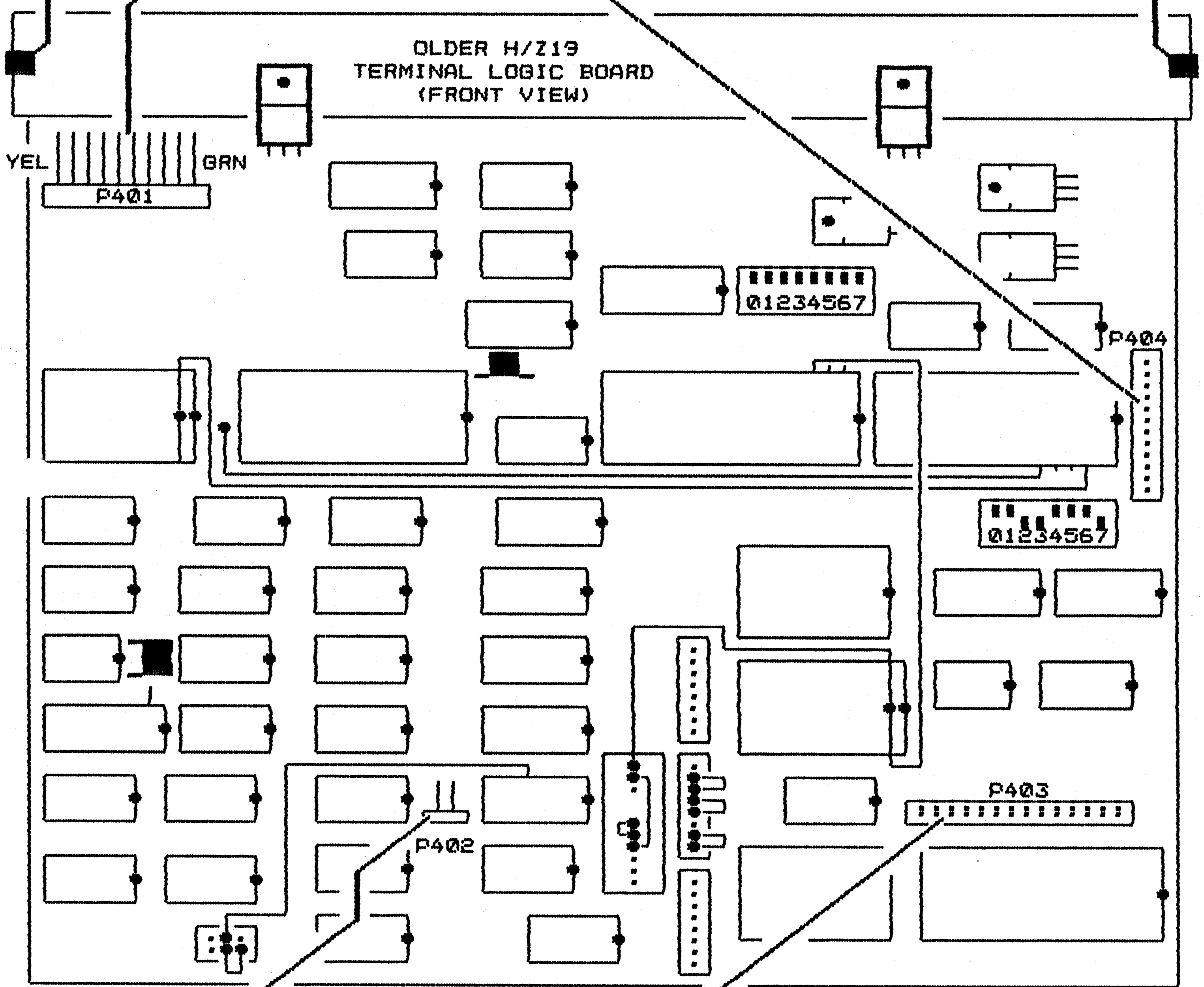
B. BEND OUT PIN 11 AS SHOWN (PIN 5 IS ALREADY BENT OUT). REPLACE IT SO PINS 5 AND 11 DO NOT ENTER THE SOCKET.

C. SOLDER A 3" WIRE FROM PIN 11 OF U419 TO PIN 4 OF JP23.

(1) PUT TERMINAL LOGIC BOARD BACK IN MACHINE. REPLACE THE MOUNTING SCREWS (1 AT EACH END).

(2) PLUG POWER CONNECTOR ONTO P401. BE SURE IT IS NOT OFFSET BY A PIN.

(3) PLUG IN SERIAL CONNECTOR P404 (ON BACK). BE SURE IT IS NOT OFFSET BY A PIN, AND THE BROWN WIRE IS ON TOP.



(4) PLUG IN THE SPEAKER CONNECTOR AT P402.

(5) PLUG IN KEYBOARD CABLE AT P403. BE SURE IT IS NOT OFFSET BY A PIN.

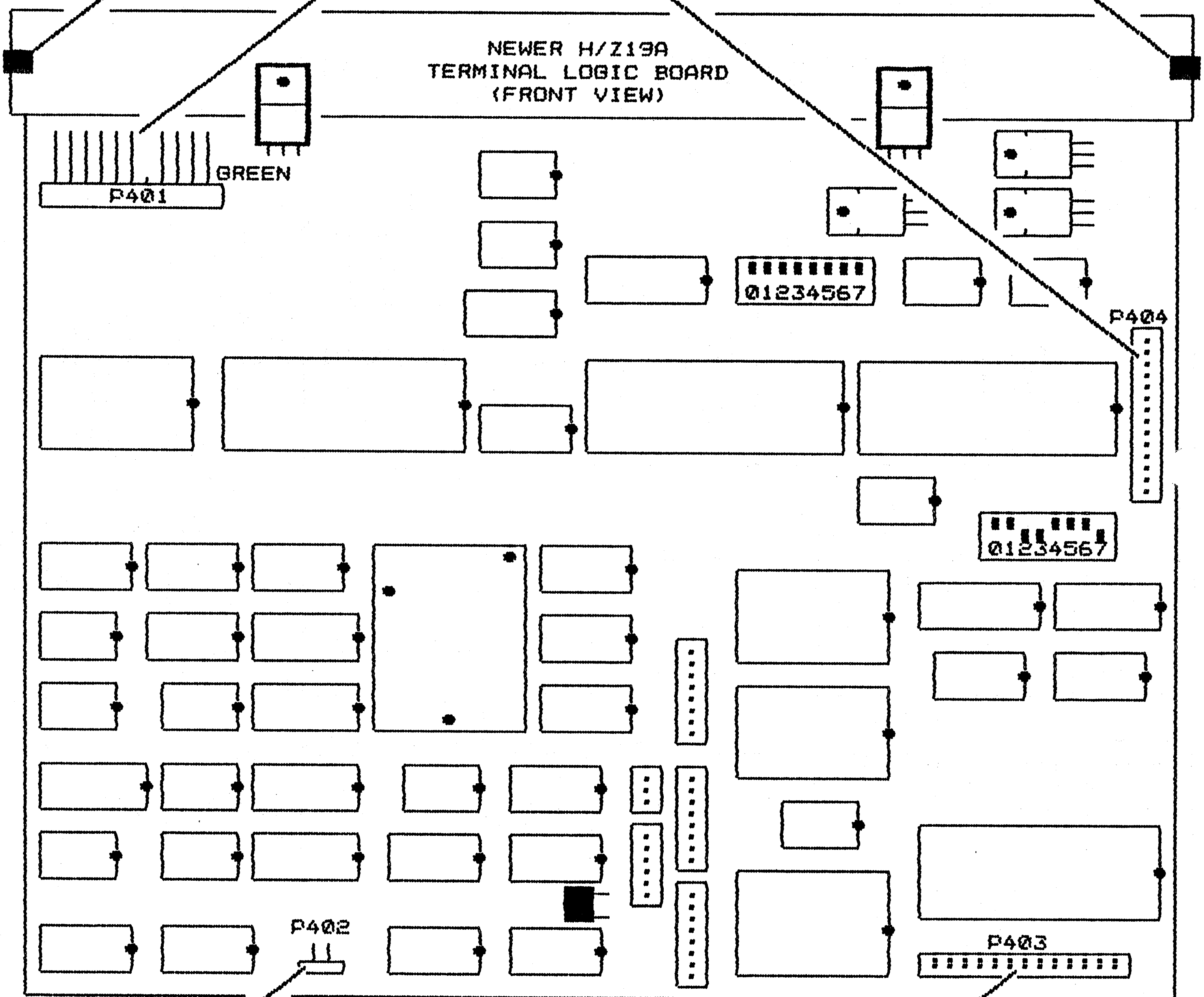
THIS COMPLETES INSTALLATION OF THE "SUPERSET".
CHECK YOUR WORK CAREFULLY; THEN GO TO "TESTING", PAGE 24.

USE THESE INSTRUCTIONS FOR A NEWER H/Z19A WITH AN 11-HOLE CONNECTOR AT P401.
FOR AN H/Z19 WITH A 10-HOLE CONNECTOR AT P401, GO TO PAGE 9.

1. REMOVE 2 MOUNTING SCREWS (ONE AT EACH END), AND 6 SCREWS ON THE BACK (IF NECESSARY).

2. UNPLUG POWER CONNECTOR AT P401.

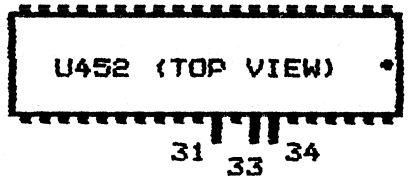
3. UNPLUG SERIAL CONNECTOR (ON BACK OF BOARD).



4. UNPLUG SPEAKER CONNECTOR AT P402.

5. UNPLUG KEYBOARD CONNECTOR AT P403.

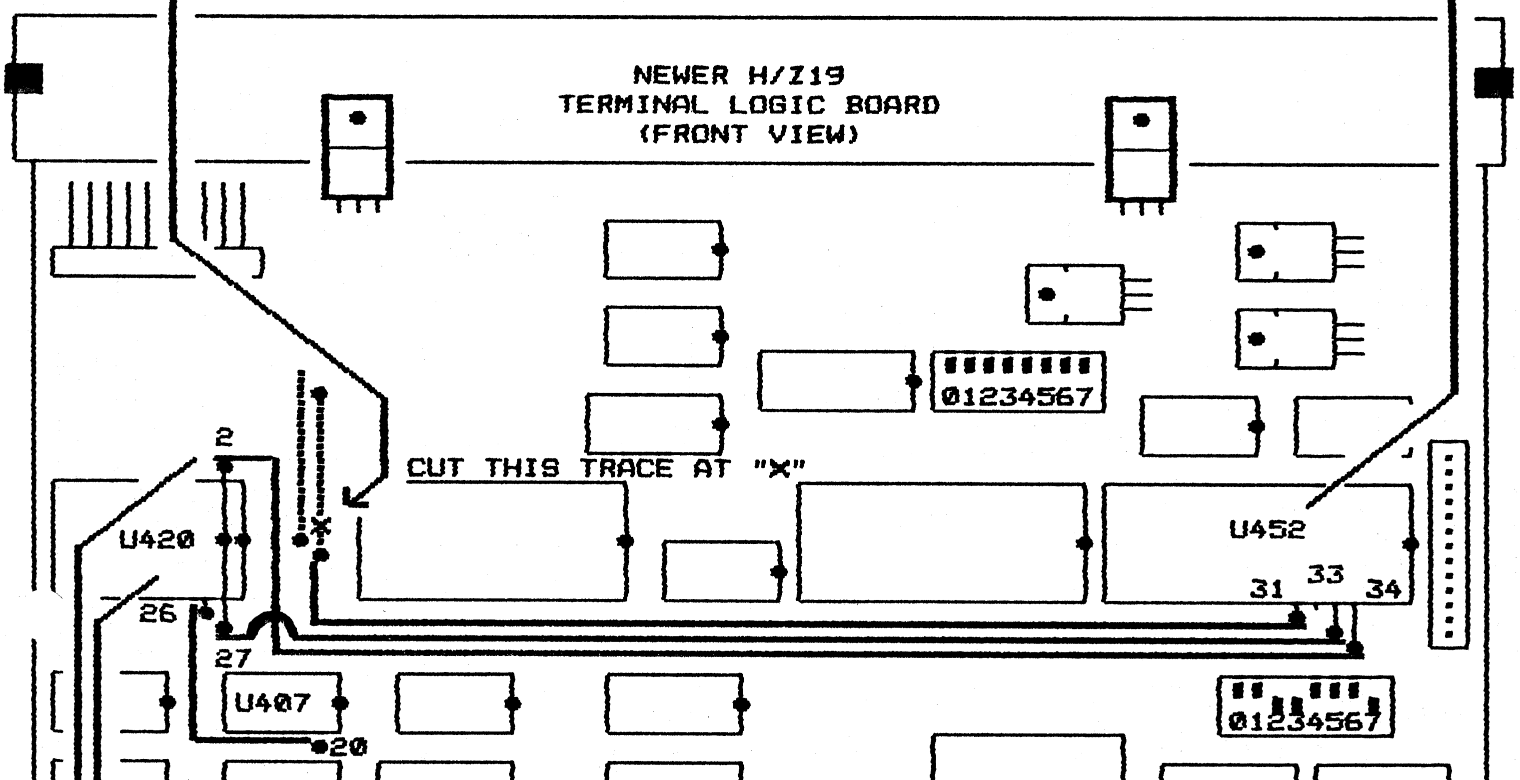
6. REMOVE TERMINAL LOGIC BOARD AND LAY ON TABLE.



1. REMOVE THE IC AT U452.
BEND OUT PINS 31, 33, AND 34.
REPLACE U452 SO THESE PINS
DO NOT ENTER THE SOCKET.

2. A. FIND THE TWO FOIL TRACES JUST TO THE RIGHT OF U420.
CUT THE <u>RIGHT</u> TRACE NEAR THE BOTTOM HOLE AT THE "X".

B. SOLDER A WIRE FROM U452 PIN 31 TO THE HOLE AT THE
LOWER END OF THE TRACE YOU JUST CUT.

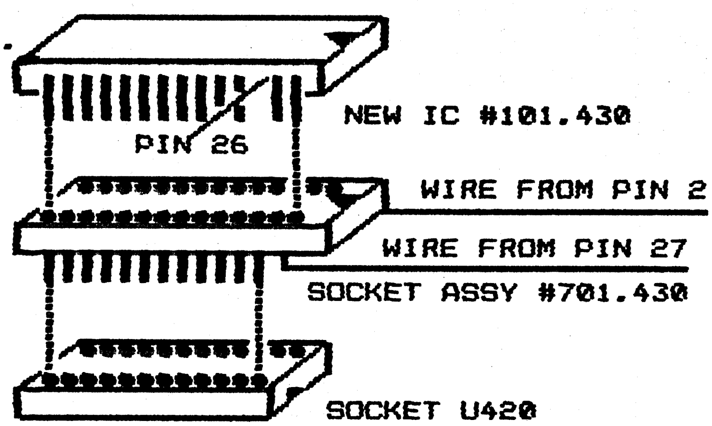


3. A. REMOVE IC #444-29 FROM U420.

B. INSTALL SOCKET #701.430
AT U420. THE SIDE WITH THE
WIRES AND EXTRA PINS GOES
ON THE <u>RIGHT</u> END AS SHOWN.

C. BEND OUT PIN 26 OF THE
"SUPERFONT" IC #101.430

D. INSTALL "SUPERFONT" IC
AT U420 SO PIN 26 DOES
NOT ENTER THE SOCKET.



4. SOLDER THE WIRE FROM PIN 2 OF SOCKET ASSY #701.430 TO U420 PIN 34.

5. SOLDER THE WIRE FROM PIN 27 OF SOCKET ASSY #701.430 TO U452 PIN 33.

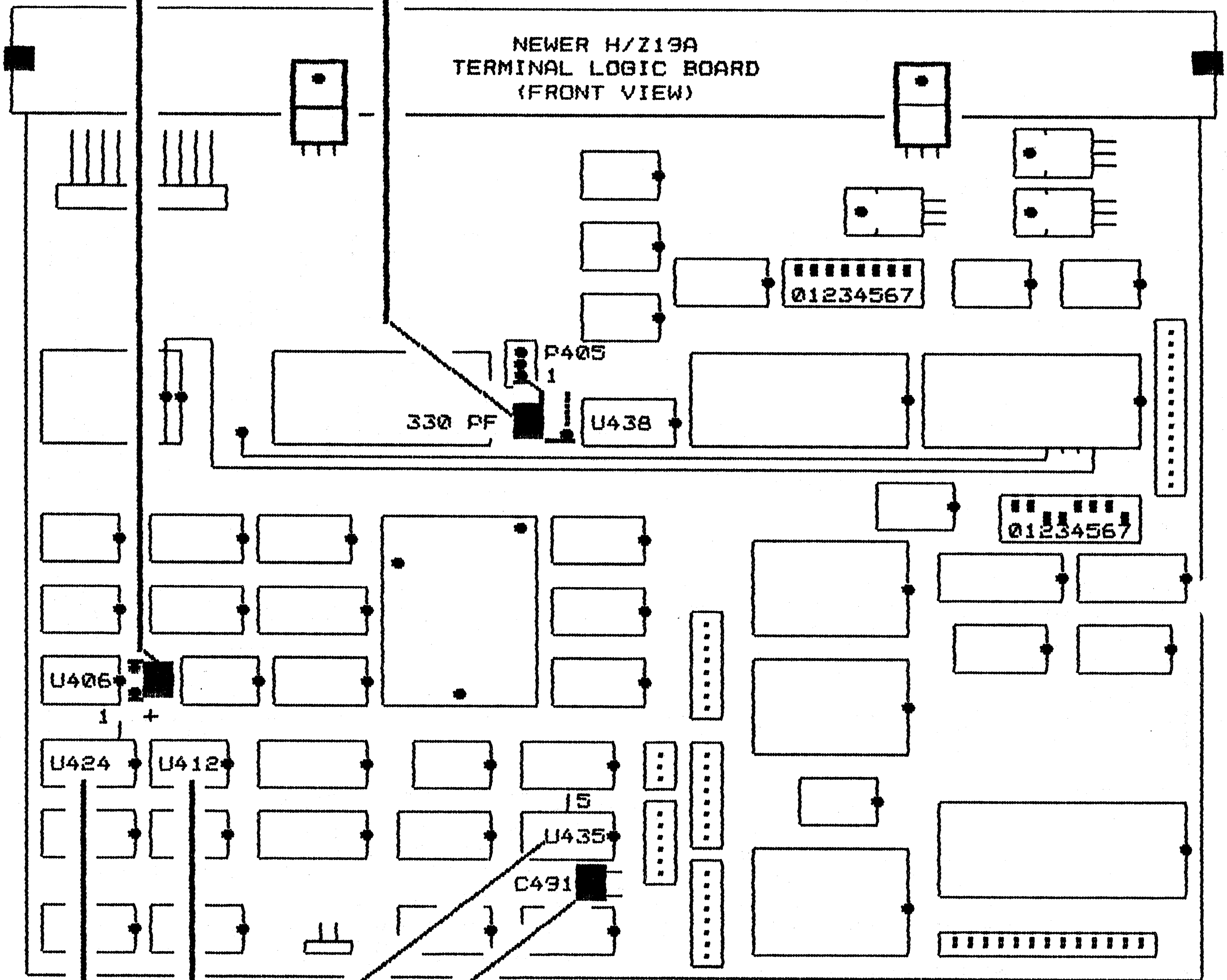
6. SOLDER ONE END OF A WIRE TO THE BENT-OUT PIN 26 OF IC #101.430.
SOLDER THE OTHER END TO THE PLATED-THRU HOLE JUST LEFT OF THE
PRINTED "20" JUST BELOW U407.

(1) REPLACE THE CAPACITOR CLOSEST TO U406 WITH THE 10 UF PART SUPPLIED. THE END MARKED "+" GOES IN THE LOWER HOLE, TOWARD U424 AS SHOWN.

(2) SOLDER A 330 PF CAPACITOR INTO THE HOLES SHOWN:

A. THE RIGHT END GOES INTO HOLE 1 OF P405.

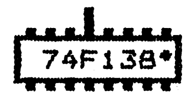
B. THE OTHER END GOES TO THE HOLE JUST LEFT OF THE SILKSCREENED LEGEND "9" NEAR U438.



(1) REMOVE THE 470 PF CAPACITOR AT C491.

(2) REMOVE 74LS138 AT U435. FIND THE NEW 74F138, AND BEND OUT PIN 5 AS SHOWN. INSTALL IT AT U435 SO PIN 5 DOES NOT ENTER SOCKET.

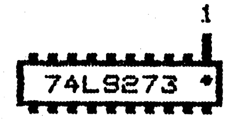
5



(TOP VIEW)

(3) REMOVE THE 74LS00 AT U412. REPLACE IT WITH THE 74F00 SUPPLIED.

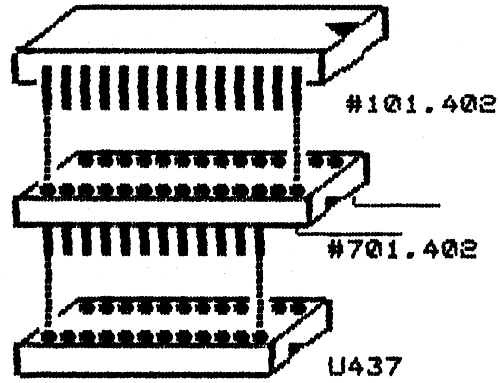
1



(TOP VIEW)

(4) REMOVE THE 74LS273 AT U424. BEND OUT PIN 1 AS SHOWN, AMD REPLACE IT SO PIN 1 DOES NOT ENTER THE SOCKET.

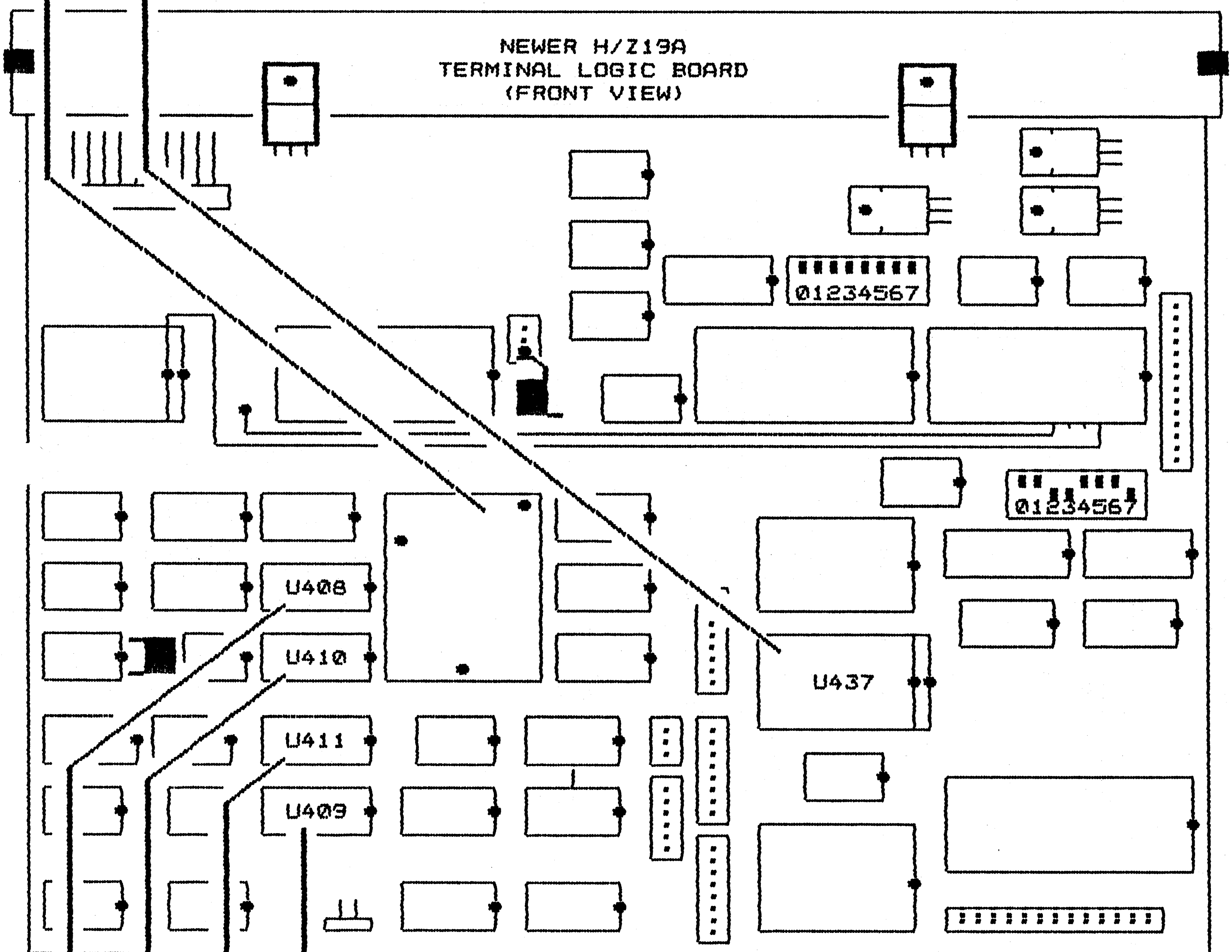
(1) REMOVE THE 3 SCREWS, AND THE METAL SHIELD OVER ICS U426 AND U427.



(2) A. REMOVE IC #444-46 AT U437.

B. INSTALL SOCKET #701.402 AT U437. THE END WITH THE EXTRA WIRES GOES ON THE RIGHT SIDE AS SHOWN.

C. INSTALL "PROGRAM" IC #101.402 AT U437.



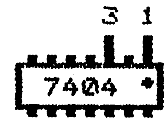
(3) REMOVE 2114 AT U409. REPLACE WITH 2114AL-2.

(4) REMOVE 2114 AT U411. REPLACE WITH 2114AL-2.

(5) REMOVE 2114 AT U410. REPLACE WITH 2114AL-2.

(6) REMOVE 2114 AT U408. REPLACE WITH 2114AL-2.

(1) REMOVE THE 7404 AT U426. BEND OUT PINS 1 AND 3 AS SHOWN. REPLACE THE 7404 SO PINS 1 AND 3 DO NOT ENTER THE SOCKET.

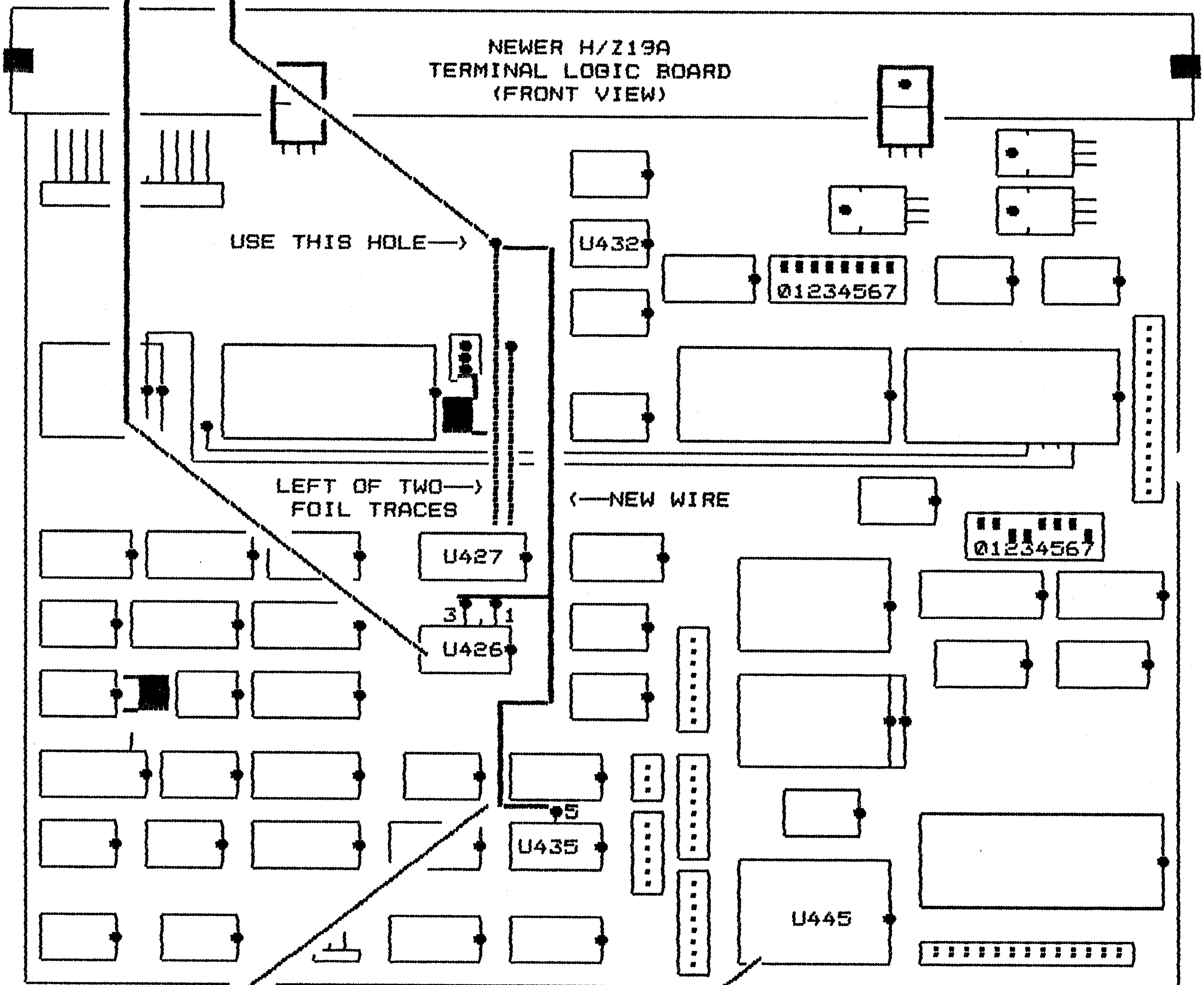


(TOP VIEW)

(2) FIND THE TWO FOIL TRACES COMING OUT FROM UNDER U427. FOLLOW THE <u>LEFT</u> TRACE UPWARD UNTIL IT ENDS.

A. SOLDER ONE END OF A 4" PIECE OF WIRE INTO THIS HOLE.

B. SOLDER THE OTHER END OF THIS WIRE TO PINS 1 AND 3 OF U426.

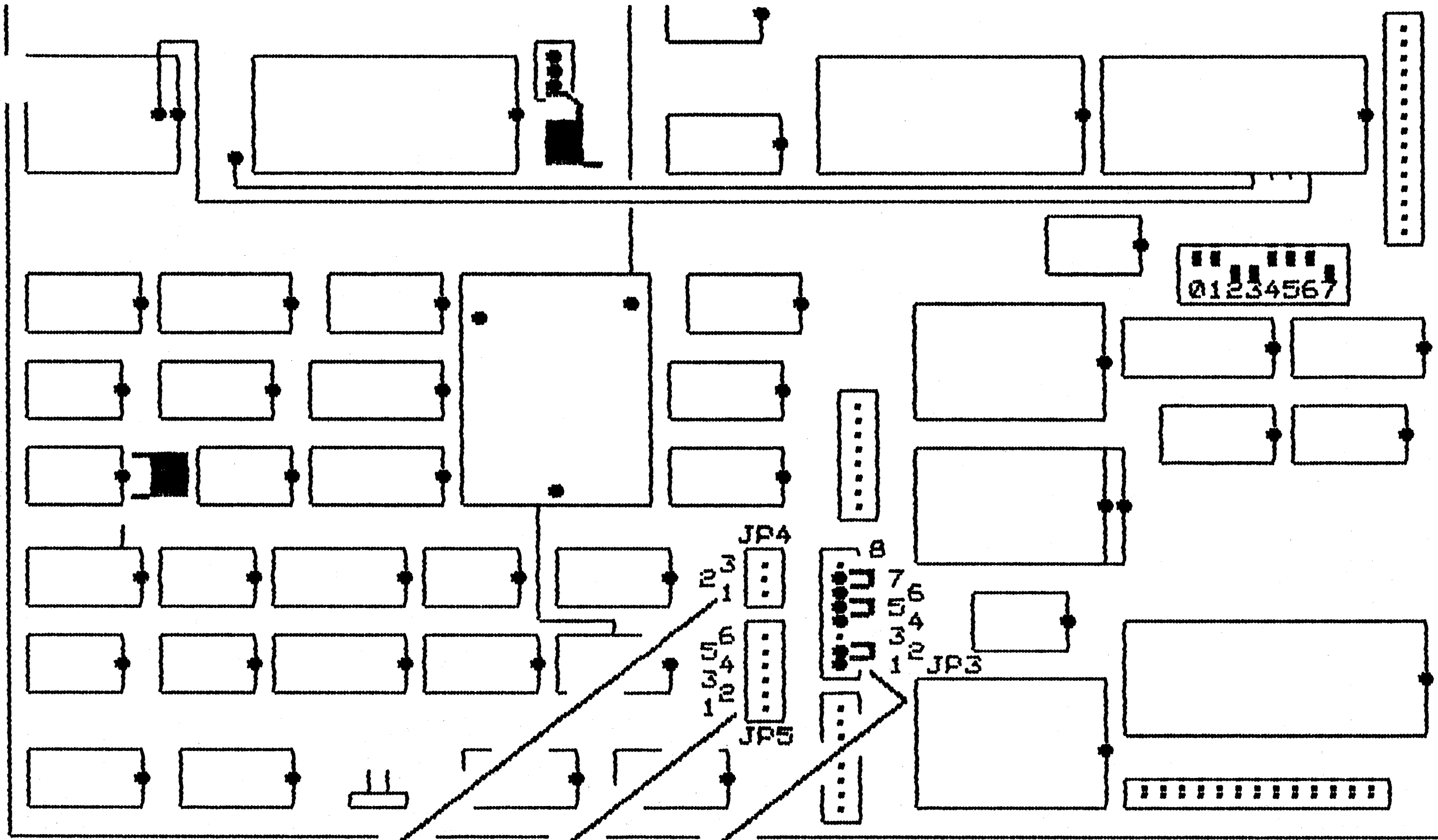


(3) REMOVE 444-37 AT U445. REPLACE IT WITH "KEYBOARD" IC #101.422.

(4) A. SOLDER ONE END OF A 4" PIECE OF WIRE TO PIN 5 OF U435.

B. SOLDER THE OTHER END OF THIS WIRE TO PINS 1 AND 3 OF U426.

(1) IF THERE IS A METAL PLATE ON THE BACK OF THE BOARD, REMOVE IT. IT IS HELD IN PLACE BY ABOUT A DOZEN SMALL SCREWS.



(2) CHECK THE JUMPERS AT JP3. IF NOT AS SHOWN:

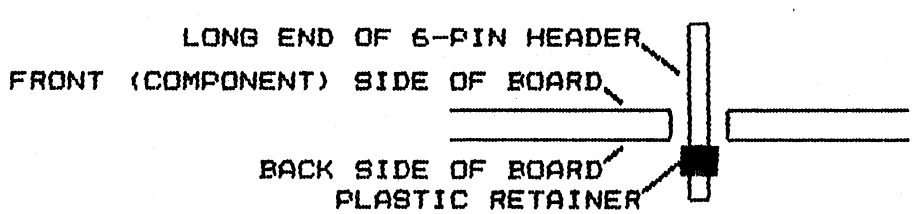
A. CUT AND REMOVE ALL JUMPERS AT JP3.
B. SOLDER A JUMPER BETWEEN JP3 PINS 1-2.
C. SOLDER A JUMPER BETWEEN JP3 PINS 4-5.
D. SOLDER A JUMPER BETWEEN JP3 PINS 6-7.

(3) A. CUT AND REMOVE ALL JUMPERS AT JP5 BETWEEN ADJACENT PINS.

B. REMOVE SOLDER FROM THE HOLES WITH THE SOLDER WICK SUPPLIED.

C. INSTALL A 6-PIN HEADER IN THE HOLES AT JP5 (SEE BELOW). THE PLASTIC BODY GOES ON THE <u>BACK</u> SIDE OF THE BOARD.

D. SOLDER THE PINS TO THE BOARD FROM THE FRONT SIDE.



(4) LIKEWISE, REMOVE ANY JUMPERS AT JP4 BETWEEN PINS 1-2 OR 2-3. SOLDER A 3-PIN HEADER INTO THE HOLES AT JP4 AS SHOWN ABOVE.

(5) IF YOU HAVE AN OHMMETER, CHECK TO BE <u>ABSOLUTELY</u> SURE THERE ARE NO SHORTS BETWEEN ADJACENT PINS EXCEPT AS SHOWN ABOVE.

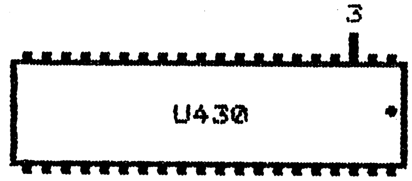
(1) IF YOU REMOVED A METAL PLATE FROM THE BACK OF THE BOARD, CUT OFF THE EXCESS PIN LENGTH AT JP4 AND JP5 ON THE BACK OF THE BOARD, SO THEY CANNOT TOUCH THE METAL PLATE. NOW RE-MOUNT THE METAL PLATE ON THE BACK OF THE BOARD.

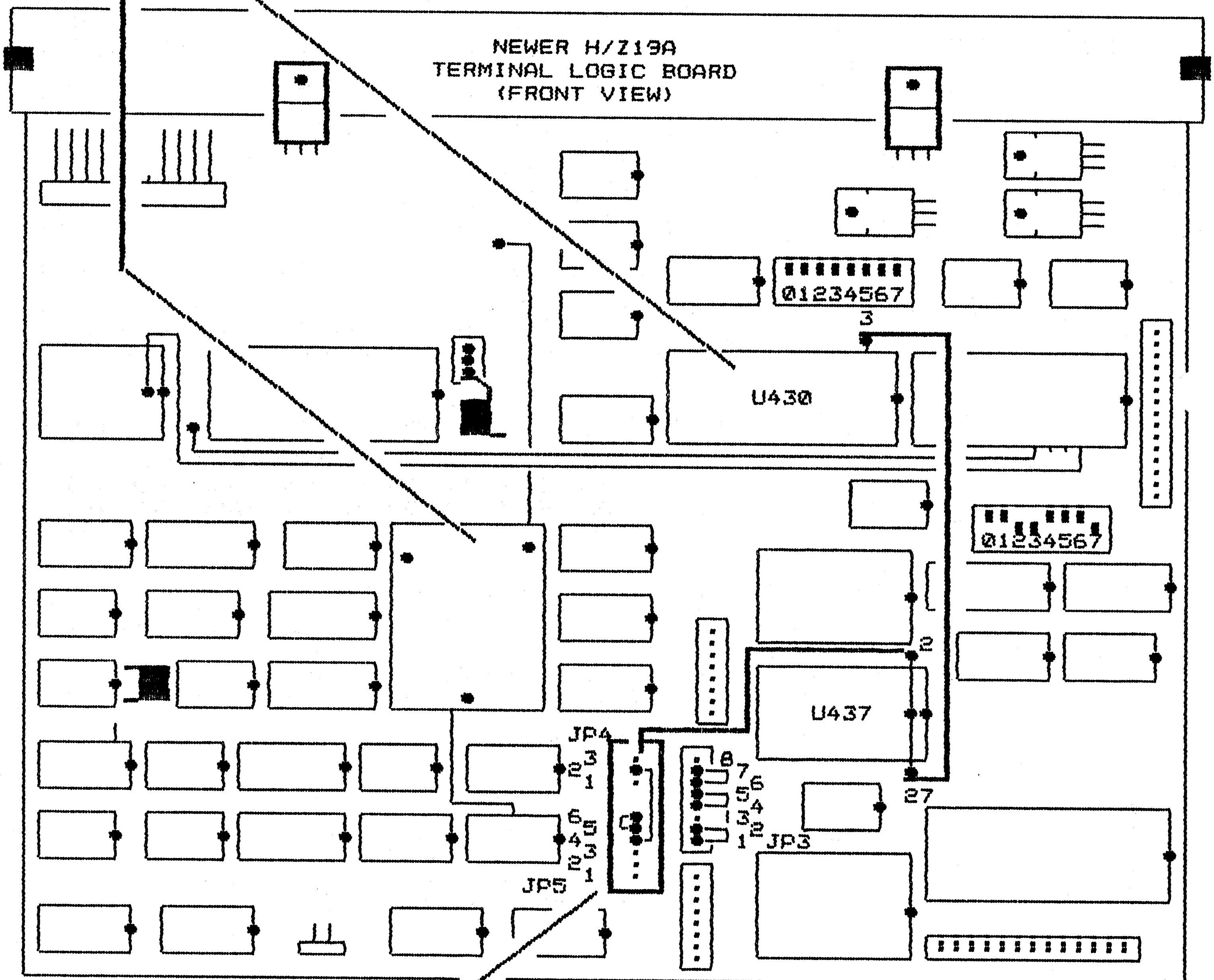
(2) REPLACE THE SHIELD AND ITS MOUNTING SCREWS OVER U426 AND U427. BE SURE YOUR NEW WIRES DO NOT TOUCH THE SHIELD.

(3) A. REMOVE THE Z80 IC AT U430, AND BEND OUT PIN 3 AS SHOWN.

B. INSTALL U430 SO PIN 3 DOES NOT ENTER THE SOCKET.

C. SOLDER THE WIRE FROM SOCKET U437 PIN 27 TO U430 PIN 3.





(4) PLUG THE 10-PIN CONNECTOR FROM SOCKET U437 ONTO JP4 AND JP5 AS SHOWN.

(1) A. AT JP2, REMOVE ANY JUMPERS (FOIL OR WIRES) BETWEEN ADJACENT PINS.

B. SOLDER A JUMPER BETWEEN JP2 PINS 6-7.

(2) LOCATE THE 4 PLATED-THRU HOLES JUST BELOW U438.

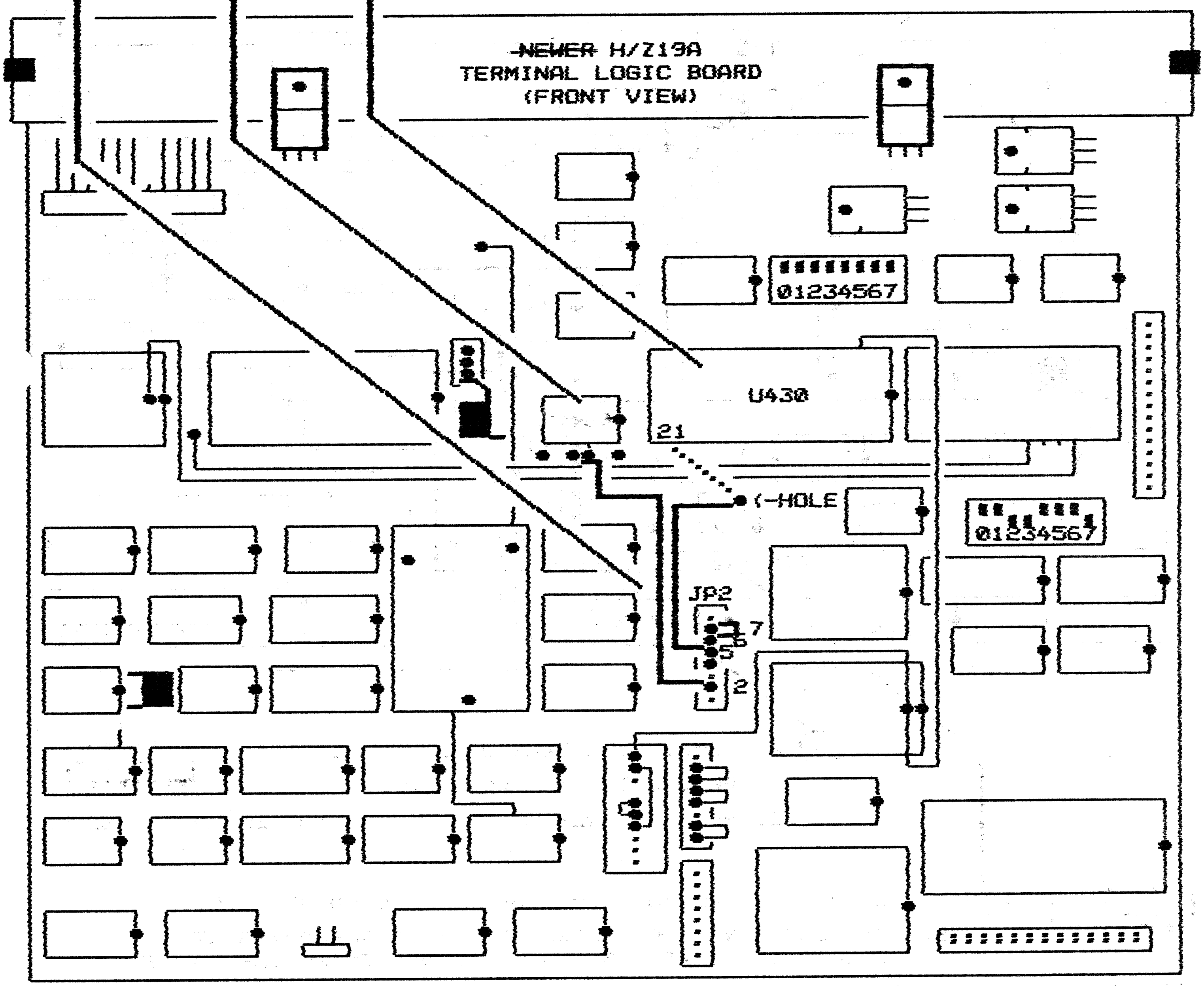
A. SOLDER ONE END OF A 4" WIRE TO THE 3RD HOLE (U438 PIN 14).

B. SOLDER THE OTHER END OF THIS WIRE TO JP2 PIN 2.

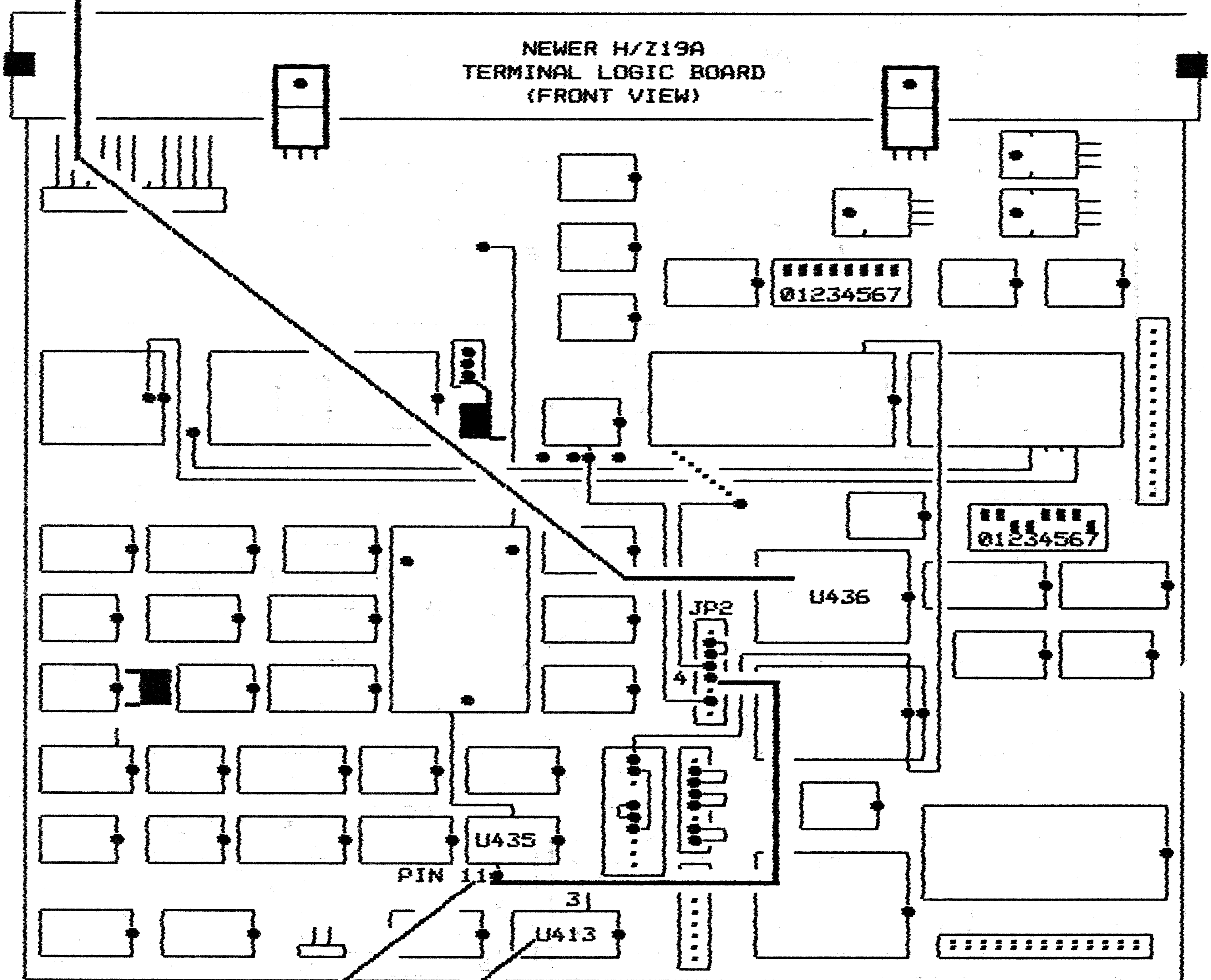
(3) FIND THE TRACE FROM U430 PIN 21 (LOWER LEFT CORNER). IT RUNS DIAGONALLY ABOUT 1/2" TO A FEED-THRU HOLE.

A. SOLDER ONE END OF A 3" WIRE INTO THIS HOLE.

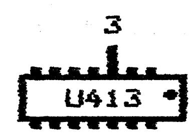
B. SOLDER THE OTHER END OF THIS WIRE TO JP2 PIN 5.

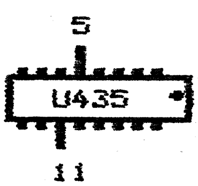


1. INSTALL THE MK48T02 SUPERCLOCK IC AT U436. BE SURE THE PIN 1 END (MARKED WITH A RED DOT) GOES IN THE UPPER RIGHT CORNER.



4. A. REMOVE THE 74LS32 IC AT U413.
   B. BEND OUT PIN 3 AS SHOWN.
   C. REPLACE U429 SO PIN 3 DOES NOT ENTER THE SOCKET.





(TOP VIEW)

5. A. REMOVE THE 74S138 PREVIOUSLY INSTALLED AT U435.
   B. BEND OUT PIN 11 AS SHOWN (PIN 5 IS ALREADY BENT OUT). REPLACE IT SO PINS 5 AND 11 DO NOT ENTER THE SOCKET.
   C. SOLDER A 2" WIRE FROM PIN 11 OF U435 TO PIN 4 OF JP2.

Notes on the SUPERCLOCK

1. Insert the following pages into your Superset manual to install and use the Superclock.

2. INITIAL TIME AND DATE - The Superclock is normally turned OFF before shipment, so it may not indicate the correct time and date upon installation. It is turned on automatically the first time you set the time.

3. FAST/SLOW SET - When you set the clock, the time display is stopped until you type the last key (after the 1's of seconds). This key also starts the Superclock oscillator, and adjusts it to run faster or slower in 5 second-per-month increments.

   Initially, use the SPACE bar (decimal 32) to start the clock; this sets the clock to its nominal speed. After 1 month, see how far off the clock is. Larger-valued keys ("!" thru "?") make it run faster; smaller values (which are control codes) make it run slower.

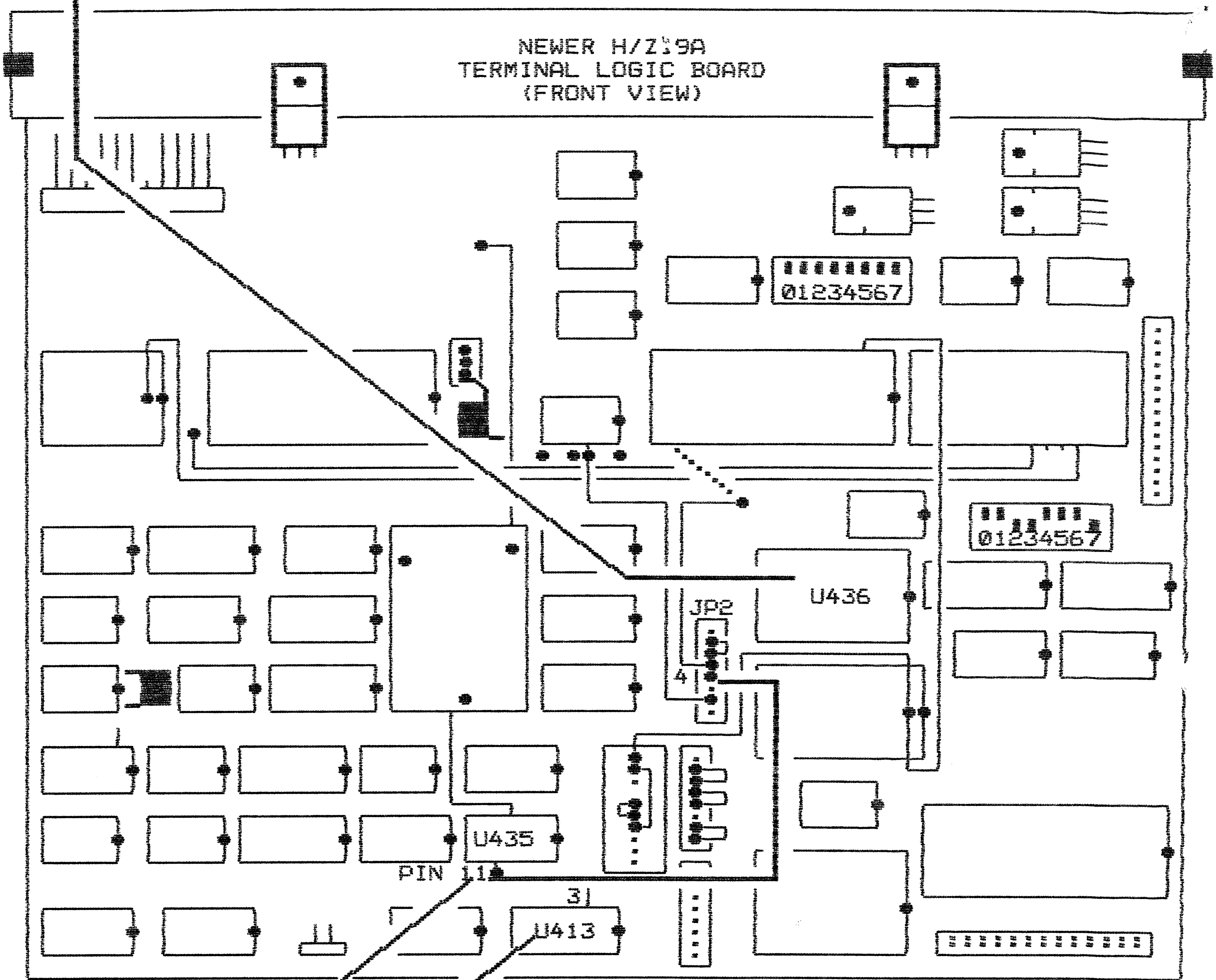
   For example, the clock is 60 seconds slow after 1 month. That's 12 steps of 5 sec/month. The final key to start the clock should be the 12th ASCII key after SPACE; 32+12=44 which is the comma (,) key.

   Or suppose the clock is 30 seconds fast per 1 month. That's 6 steps, so the final key is the 6th ASCII code before SPACE; 32-6=26, which is control-Z.

4. POWER SUPPLY VOLTAGE - The Superclock write-protects itself when its power supply voltage is out of range. If everything works with the power on, but it forgets the time when the power is off, check the voltage between pins 12-24 of the Superclock chip. It must be between 4.75 to 5.25 volts with the board powered up and working. If not, setting the clock or calendar only affects the screen, and doesn't get saved in the Superclock.

   This is especially likely with newer H19A boards. Heath added inductors L403-L407 to reduce EMI emissions, but they also cause a substantial voltage drop in the +5v supply. If the voltage regulator outputs are OK, but the IC voltages are out of spec, replace the inductors with a piece of wire.

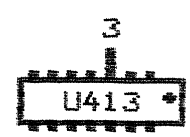
(1) INSTALL THE MK48T02 SUPERCLOCK IC AT U436. BE SURE THE PIN 1 END (MARKED WITH A RED DOT) GOES IN THE UPPER RIGHT CORNER.

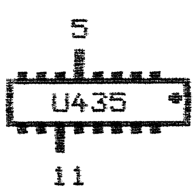


(4) A. REMOVE THE 74LS32 IC AT U413.

B. BEND OUT PIN 3 AS SHOWN.

C. REPLACE U413 SO PIN 3 DOES NOT ENTER THE SOCKET.





(TOP VIEW)

(5) A. REMOVE THE 74F138 PREVIOUSLY INSTALLED AT U435.

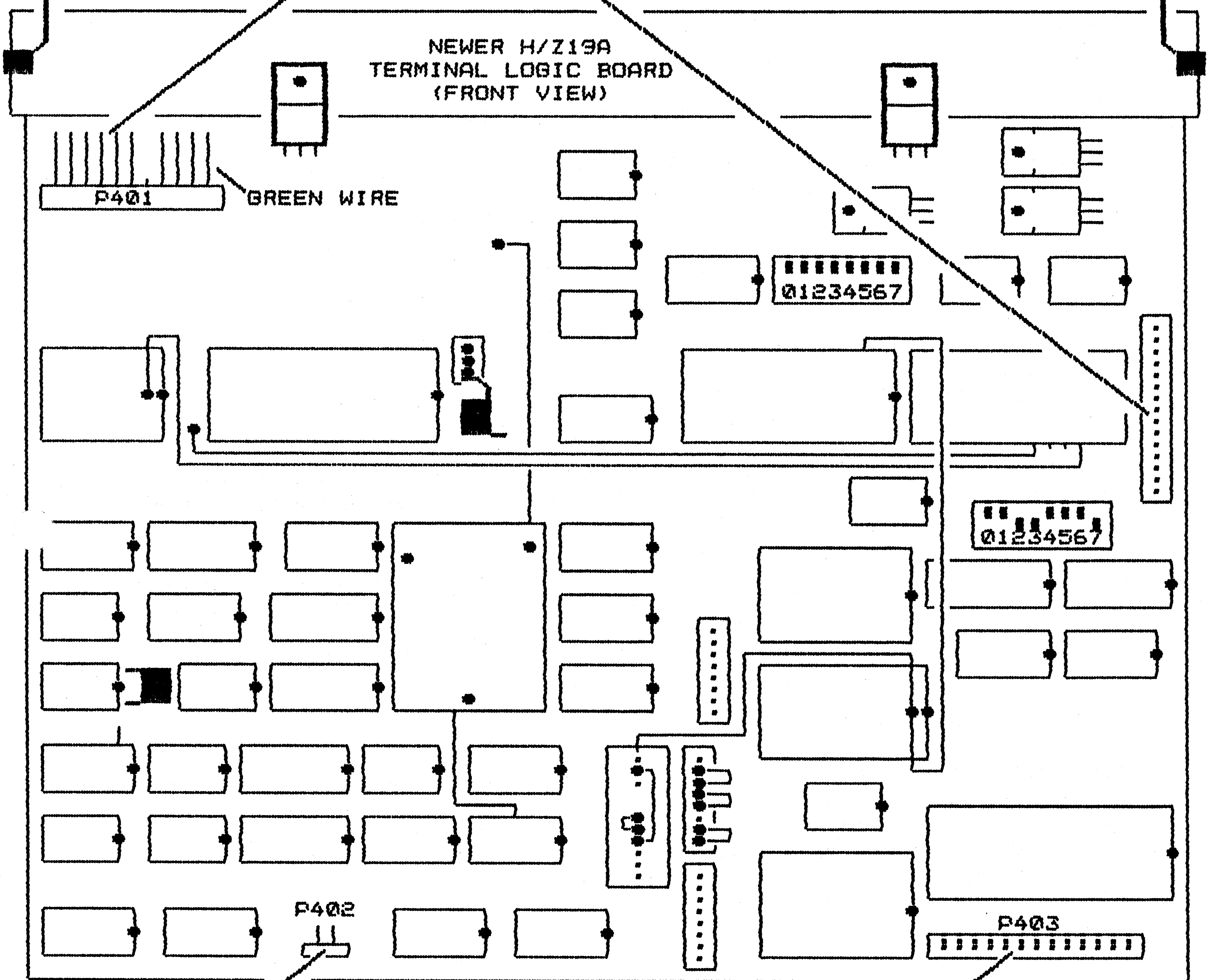
B. BEND OUT PIN 11 AS SHOWN (PIN 5 IS ALREADY BENT OUT). REPLACE IT SO PINS 5 AND 11 DO NOT ENTER THE SOCKET.

C. SOLDER A 2" WIRE FROM PIN 11 OF U435 TO PIN 4 OF JP2.

(1) PUT THE BOARD BACK IN THE MACHINE, AND REPLACE THE MOUNTING SCREWS.

(2) PLUG THE POWER CONNECTOR ONTO P401.

(3) PLUG IN SERIAL CONNECTOR P404 (ON BACK).
BE SURE IT IS NOT OFFSET BY A PIN,
AND THE BROWN WIRE IS ON TOP.



(4) PLUG IN THE SPEAKER CONNECTOR AT P402.

(5) PLUG IN THE KEYBOARD CONNECTOR AT P403.
BE SURE IT IS NOT OFFSET BY A PIN.

THIS COMPLETES INSTALLATION OF THE "SUPERSET".
CHECK YOUR WORK CAREFULLY. THEN GO TO "TESTING", PAGE 25.

TESTING AND REPAIRS

To test your SUPERSET in an H19 or Z19 terminal, assemble everything normally, but leave the top cover open.

If you have an H89 or Z90 computer, you can test your installation before putting the CPU board back. This is done by plugging cable connector J514 (that originally went to P514 on the CPU board) onto P401 of the TLB board instead. There are several possibilities:

Older H89 or Z90: - P514 and P401 are both 10-pin connectors.
- There is a yellow wire in pin 10 of J513.

THEN: - Plug J513 directly onto P401.

Newer H89-A or Z90-A: - P514 and P401 are both 11-pin connectors with one pin removed or blocked.
- There is no wire in pin 11 of J513.

THEN: - Remove the shielded wire from pin 11 of the 6" cable that connected P401 to P513, and plug this wire into pin 11 of J513.
- Plug J513 directly onto P401.

All other cases: - P514 and P401 have different numbers of pins, or special CPU or graphics boards are installed.

THEN: - Install the CPU board normally. You could rearrange the wires at J513 for testing, but it's more trouble than it's worth.

- It is OK to leave the I/O and accessory cards unplugged while testing the terminal. If you have a Heath/Zenith (or Magnolia) soft-sector disk controller, you won't get the "H:" (or "MMS:") prompt with it removed.

QUICK TESTS

Turn on the terminal (computer). The CRT filament should warm up and glow orange, and you should get a cursor in the upper left corner of the screen. A digital clock/calendar will appear in the lower right corner, being updated every second. The rest of the screen should be blank. Push the "OFF-LINE" key so it latches down, and type on the keyboard. It should act like a typewriter; whatever you type appears on the screen.

Type enough characters to fill most of the screen. The "REPEAT" key will help do this quickly. The screen should be rock-stable, and centered vertically and horizontally. If not, adjust the vertical height, vertical linearity, horizontal width, and horizontal centering controls on the video board as necessary (see your Heath/Zenith manuals for details).

Type the alphabet in uppercase letters, and then again in lowercase letters right under it. Now type "ESC F" and re-type the lowercase letters again. They should display as the standard Heath graphics characters. There are two exceptions: The graphic "j" (which used to be ÷) is now % like the Z29 and Z100; and the graphic "g" (which was the ± sign) is now ░ the 1/4-dot fill symbol.

Now type "ctl-ESC" (short for hold down the CTRL key and then hit ESC). A menu should appear on the right side of the screen. The screen should be stable, with no trace of flicker or instability. You can page through the menus by hitting any key on the numeric keypad.

Select an item in the menu by typing the appropriate key (upper/lower case matters). For instance, step to "MAIN MENU 1". Type a lowercase "h" for the "set options". The main menu disappears, and the "SET OPTIONS" menu appears. Now type a "5" to select a "white screen" (typewriter "5", not the keypad "5"). The menu disappears, and the screen now displays dark characters on a white background.

You will also see local function keys (in parentheses) in the menus. A local function key is one that takes effect immediately, without requiring the off-line key or menus. Type "ctl-ESC" again to display the menu. Type the "SPACE" bar to select the "character set" menu. Note that you can select the GT-PROM character set with either "L" or local function key "ctl-f3".

Type "ctl-f3" (hold down CTRL and then hit the "f3" key). This selects the GT-PROM character set, which has bolder text characters, and different graphics characters. Now try "ctl-f2" for the VT-100 set. Here you will find thin-line graphics plus special symbols for CR, LF, VT, etc. Now hit "ctl-f4" to select the special character set, which contains Greek letters and math symbols. Last, type "ctl-f1" to return to the standard Heath set.

Hold down the "LINE FEED" and "REPEAT" keys. The text should scroll quickly off the top of the screen, leaving it blank. Type some more characters, and then hit the right "SHIFT" and "RESET" keys. The screen should be instantly erased, but the clock and calendar stay valid.

If no problems are encountered, your installation was a success! If you have any problems, turn off the power and recheck your work carefully. Look for ICs installed backwards or missing, ICs that get "hot", connectors installed wrong, etc. If you can't find the problem, check the list of possible problems below.

IN CASE OF DIFFICULTY

1. Random characters on screen when first turned on, or while right-shift-RESET are depressed, but OK after reset.
   - Occurs in very old H19s; looks odd, but not a problem.

2. Everything works, but the black "dashes" are still present.
   - Pin 1 of the 74LS273 is still in its socket. Bend it out so it doesn't touch anything. In rare cases, you may have to add a wire between pin 1 and +5 volts.
   - Pin 5 of the 74F138 not connected correctly.

3. Everything works, but I get white "dashes" as I type.
   - The wiring to pin 5 of the 74F138 is wrong. Check, and fix.

4. Menus don't appear when I type ctrl-ESC.
   - Z80 not running at 3 MHz. Recheck wiring at U426 (newer H19-A) or JP10 (older H19).
   - 50/60Hz switch (S402 #7) in wrong position.
   - new keyboard decoder #101.422 not installed.

5. Menus appear, but flicker badly.
   - 330pF capacitor not installed correctly.
   - wiring to pin 5 of the 74F138 is wrong.

6. Completely dead; CRT does not light up, no clicks from keyboard.
   - Connectors P401, P514, or P515 installed wrong.
   - 101.401 Program ROM installed wrong.
   - Program ROM too slow (used original part instead of new one).
   - Jumpers for Program ROM #101.401 wrong. Be sure you found all the old jumpers; sometimes there is both a piece of wire AND foil on the board. Look on BOTH sides of the board. Remove only jumpers between ADJACENT pins, not wiring between non-adjacent pins.

7. As I type, random characters appear anywhere on screen.
   - 470pF capacitor not removed.
   - 74F00 not installed, or in wrong place.
   - 74F138 not installed, or in wrong place.
   - slow or bad 2114AL-2 video RAMs (or using old 2114s).

8. Random characters appear only on upper (or lower) half of screen.
   - Slow or bad video RAMs. Locate the bad IC by swapping pairs of ICs and see if the problem moves to different regions of the screen.
   - 74F00 not installed correctly.
   - 470pF capacitor not removed.

9. Screen all white; but keys "click" and cursor moves normally.
   - character generator ROM installed wrong.

10. Can't get all 4 character sets, or get them with the wrong ESC sequences.
   - wires from CharGen ROM #101.412 to 8250 (U451 or U452) connected wrong.

11. System OK, then "crashes" (no key clicks, screen freezes). Operation is restored by right-shift-RESET.
   - Loose connections, bad solder joints, and ICs not in their sockets. Flex the board slightly while operating to see if it promotes failures.
   - Slow Z80 (U421 in older H19, U430 in newer H19A). Replace with 4MHz Z80A.
   - Slow 6845 (U461 in H19, U417 in H19A). Replace it with a 68A45 or 68B45.
   - Slow 2112s (U424-U425 in H19, U438-U439 in H19A). Install a faster part, or modify the chip select logic at U419 (H19) or U435 (H19A) as follows:
     a. Get the old 74LS138; cut off pin 9 close to the case.
     b. Remove the new 74F138; bend out pins 5,7,9,10,11,12,13,14, and 15.
     c. Mount 74F138 on top of 74LS138; solder pins 1,2,3,4,6,8, and 16.
     d. Plug the "piggybacked" pair into socket U419 (H19) or U435 (H19A).
     e. Solder one end of a short piece of wire to pin 9 of the 74F138.
     f. Follow the foil trace on the BACK side of the board from pin 9 of U419 (H19) or U435 (H19A) about 2" left to a plated-thru hole. Solder the other end of the wire to this hole.
   - If all else fails, you can run the TLB at 1.5MHz instead. On an older H19, connect JP10 pins 3-6 (instead of JP10 1-5); On newer H/Z19-As, don't connect the wire to the plated-thru hole above U427; instead, solder it to U27, pin 12. At 1.5MHz, you will have to stay at 9600 baud, and the menus will not be displayed.

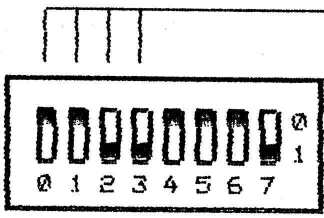
12. Vertical "shake" or "shimmy; clock runs fast or slow.
   - DIP switch SW402 #7 (50/60 Hz select) in wrong position.
   - 330 pF capacitor not installed correctly.

13. Slight vertical "jitter" in interlace mode.
   - Interlace mode always does this. It is most noticeable with white tubes, fluorescent room lighting, high brightness, and large amounts of reverse video. It's best with a green or amber tube, and reduced brightness levels. The only sure cure is to replace the CRT tube with a longer persistence phosphor.

DIP SWITCH SETTINGS

The functions of switch S401 on the terminal logic board have been altered to accept new baud rates and word lengths. Switch S402 has not been altered, but is included here for completeness. The switches are shown in their normal default positions.

SWITCH S401 - Primary Power-up Configuration



SWITCH# Ø	1	2	3		BAUD RATE
Ø	Ø	Ø	Ø	-	75
1	Ø	Ø	Ø	-	110
1	1	1	1	-	134.5
Ø	1	Ø	Ø	-	150
1	1	Ø	Ø	-	300
Ø	Ø	1	Ø	-	600
1	Ø	1	Ø	-	1200
Ø	1	1	Ø	-	1800
1	1	1	Ø	-	2000
Ø	Ø	Ø	1	-	2400
1	Ø	Ø	1	-	3600
Ø	1	Ø	1	-	4800
1	1	Ø	1	-	7200
Ø	Ø	1	1	-	9600
1	Ø	1	1	-	19200
Ø	1	1	1	-	38400

Ø = Parity enabled
1 = Parity disabled

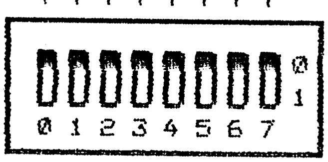
Ø = Odd parity
1 = Even parity

Ø = 8-bit word length
1 = 7-bit word length

Ø = Full duplex
1 = Half duplex

SWITCH S402 - Secondary Power-up Configuration

Ø = Cursor is blinking underline
Ø = Key click enabled
Ø = Discard characters past end of line
Ø = no LF on CR
Ø = no CR on LF
Ø = Heath/Zenith ESCape sequences
Ø = Unshifted keypad
Ø = 60 Hz AC line



1 = 50 Hz AC line
1 = Shifted keypad
1 = ANSI mode ESCape sequences
1 = Auto CR on LF
1 = Auto LF on CR
1 = Wrap characters past end of line
to start of next line
1 = Key click disabled
1 = Cursor is non-blinking box

OPERATION

This chapter describes the SUPERSET from a user's point of view. I'll assume you're already familiar with normal H/Z-19 operation, and just cover the changes caused by the SUPERSET. If this is a rash assumption, then you should be reading that great American classic, the "owner's manual".

BACKGROUND

The H/Z-19 is an ASCII terminal; it sends and receives data using the American Standard Code for Information Interchange (pronounced "ASS'key"). ASCII defines 128 characters. 95 are "printable"; when received, they print something on the screen and advance the cursor. The other 33 are "control" codes; they aren't displayable, but do cause something to happen. RETURN, LINE FEED, BACKSPACE, DELETE, ESC, and TAB are control codes important enough to have their own keys.

The rest of the control codes are sent by holding down the CTRL key and then pressing A-Z @ [\ ] ^ or _. Like the SHIFT keys, CTRL "shifts" the function of other keys. For example, ETX is "End of TeXt", better known to CP/M fans as control-C. To save space, I'll abbreviate this as "ctl-C".

THE KEYBOARD

Every key on the H19 already has an assigned function, so how do you add more? The trick is the CTRL key! Holding down the CTRL key tells the SUPERSET you want to perform a "local function", i.e. send a command directly to the SUPESET. Here are the local functions:

Local Function	Description
ctl-@, ctl-A thru ctl-Z, ctl-[, ctl-\, ctl-], ctl-^, and ctl-_	send control codes to computer (same as H19)
ctl-ESC	display "Magic Menu" of available functions
ctl-shift-ESC	unlock keyboard
ctl-DELETE	"hard" reset terminal to power-up state
ctl-SCROLL	disables Hold Screen mode
ctl-shift-SCROLL	enables Hold Screen mode
ctl-f1	select H19 character set
ctl-f2	" VT-100 " "
ctl-f3	" GT-PROM " "
ctl-f4	" Special " "
ctl-shift-f1	select H29 character set (with SUPERFONT)
ctl-shift-f2	" VT-100 extended " "
ctl-shift-f3	" APA-Graphics " "
ctl-shift-f4	" IBM PC monochrome " "
ctl-f5	toggle interlace on/off
ctl-shift-f5	" white screen on/off
ctl-BLUE	" transparent screen on/off
ctl-shift-BLUE	" native keyboard on/off
ctl-RED	restore modes (and screen with SUPERCLOCK)
ctl-shift-RED	swap screen with Page 2 (with SUPERCLOCK)
ctl-WHITE	send time to computer
ctl-shift-WHITE	send date to computer

Local function keys ALWAYS work, regardless of the terminal's status or modes. They are not affected by the OFF-LINE key or any of the possible modes (ANSI mode, keyboard status, transparent screen, etc.). They do not send any characters to your computer, except when you specifically ask for it, like "send control code" or "send time".

MAGIC MENUS

Local function ctl-ESC displays a series of on-screen "help" menus. The menus list every mode and command that your SUPERSET recognizes. Additionally, you can enter most commands directly from the menus, avoiding the need to go "off-line". This virtually eliminates the need for the plastic "cheat sheet" when trying to use the terminal.

The first ctl-ESC shows Main Menu 1: Modes of Operation. Select any mode in the menu by simply typing its corresponding key. Type "F" to enable graphics, for example. Note that upper/lower case matters. The graphics mode is enabled, the menu disappears, and the screen is restored to its previous state.

If you don't want to change anything, just hit RETURN. If the function you want isn't shown, any key on the keypad will advance you to the next page in the main menu. After the last page, the menus start over with the first.

Some functions will call up sub-menus. For example, "r" (set baud rate) displays the baud rate menu. It shows that "L" will select 9600 baud. Type the desired key, or just hit RETURN to exit without changes.

The keys shown in the menus are also the ESC sequences needed to perform the commands. Thus "ESC r L" is also the correct ESC sequence to transmit to the terminal to select 9600 baud.

The "set time", "set date", and "position cursor" commands appear in the menus, but can't be selected from them (you must go off-line, or have them sent by the computer). The ANSI mode can be selected, but ANSI ESC sequences are not shown.

If a command has a local function key, it is shown in parentheses. Local function keys work immediately even while displaying the menus; thus you should not use the ctl-RED "swap screen" or it may save a piece of the menu as well!

I call them "magic" menus because they work instantly, and can save/restore the screen without additional RAM. I CHALLENGE you to figure out how I did it!

LOCAL EDITING KEYS

Under most circumstances, you work with the OFF-LINE key up. Keys you type are sent to your computer, but have no direct effect on the screen. The screen only changes if your computer sends characters back to change it. If the program running is pretty smart, that's OK. You can enter data, correct it, and move the cursor around because the program understands how to do it.

But dumb programs don't know how to move the cursor, or do any of the other fancy things a smart terminal can do. This makes such programs harder to use.

Have you ever typed a complex PIP command and hit return, only to find that you left out a period? Have you ever cursed Microsoft BASIC's clumsy line editor? Have you ever wondered why you can't just move the cursor to the error, fix it, and then re-execute the command? Well, that's what local editing lets you do!

If you hold down the CTRL key and hit any key on the numeric keypad, no code will be sent to your computer. Instead, the terminal moves the cursor, or inserts or deletes characters or lines just as if you were using a text editor.

Local Edit Keys	Description
ctl-ERASE	erase screen from cursor position to end
ctl-shift-ERASE	erase entire screen, and move cursor to home
ctl-↑ (keypad 8)	cursor up
ctl-← (keypad 4)	cursor left
ctl-→ (keypad 6)	cursor right
ctl-↓ (keypad 2)	cursor down
ctl-DC (keypd 9)	delete character at cursor
ctl-DL (keypd 3)	delete line at cursor
ctl-IC (keypd 7)	toggle "insert character" mode on/off
ctl-IL (keypd 1)	insert a blank line at cursor
ctl-0 (keypad 0)	type a "0" on-screen
ctl-. (keypad .)	type a "." on-screen
ctl-HOME (kpd 5)	home cursor to upper left corner
ctl-ENTER	move cursor to left end of line
ctl-RETURN	send line from cursor to end, w/o trailing spaces

Suppose you typed the following PIP command and hit return; PIP would respond with the error message:

```
A>pip existing.doc=existingbak<ret>

CANNOT READ: EXISTINGB

A>
```

Nuts: You forgot the period between "existing" and "bak". Here's a way to fix it without retyping the command:

1. Hold the CTRL key down for all the following steps.
2. Move the cursor to the "b" in "existingbak" with the keypad arrow keys.
3. Hit IC (keypad 7) to toggle the "insert character" mode ON.
4. Hit the keypad "." key to insert a period.
5. Hit IC again to toggle the "insert" mode back OFF.
6. Move the cursor back to the 1st "P" in "PIP".
7. Hit RETURN.
8. Now release the CTRL key.

The final ctl-RETURN sends every character from the cursor to the end of the line to your computer, just as if you had typed them all yourself. Trailing spaces are not sent, and a RETURN is sent at the end of the line. PIP will see a perfectly valid command this time, and execute it just fine.

The same holds true for editing a BASIC program on-screen. Suppose you want to copy line 1000 to 2000:

1. Type "LIST 1000" to display the line.
2. Hold down the CTRL key, and move the cursor to the "1" in "1000".
3. Push the OFF-LINE key so it latches down.
4. Type "2" to change the line to "2000".
5. Push the OFF-LINE key so it pops up.
6. Hold down the CTRL, and move the cursor back to the "2" in "2000".
7. Hit ctl-RETURN to send the line to BASIC.

If you list your program, there will be a new line 2000, which is exactly the same as line 1000. Notice you had to go off-line to type the "2". That's because keys in the main keyboard would just send control codes with the CTRL key down. You could just as well LIST a whole subroutine, go off-line and edit it to your heart's content, and then transmit your changes selectively back into BASIC line-by-line. It's complicated to describe, but straightforward to actually do.

While the line is being sent by a ctl-RETURN, a 2nd cursor shows the character being sent (though not usually visible at high baud rates). Not all programs can accept data this fast; if you get a long "beeeeeeep" when you transmit a line, it means somebody can't keep up. Try enabling transmit speed limiting. If that doesn't help, then handshaking is required.

The SCROLL Key

The SCROLL key was pretty useless on a standard H/Z-19. Most of the time it did nothing. If the "Hold Screen" mode was enabled, the SCROLL key was required to allow the screen to scroll.

The SUPERSET fixes this. With Hold Screen mode off (i.e. normal) the SCROLL key alternately sends a ctl-S, then a ctl-Q (XOFF and XON respectively). In CP/M and HDOS, this alternately starts and stops the display. You can thus "type filename" and use the SCROLL key to start and stop typing text as you read it.

The SCROLL key can get out of sync with the operating system (sends a ctl-Q when CP/M was expecting a ctl-S, and vice versa). If so, just hit SCROLL a second time. In CP/M for instance, you stop the display with the SCROLL key, then start it again with the SPACE bar. The next SCROLL key sends its ctl-Q, which is ignored by CP/M because the display was already started. The next SCROLL key again sends ctl-S to stop the display.

The Hold Screen mode has also been improved. It is used to prevent received characters from causing the screen to scroll up. Hold Screen mode is enabled by local function key ctl-shift-SCROLL.

Hold Screen counts line feeds; if a line feed would cause text to scroll off the top of the screen, the SUPERSET holds the display. It uses handshaking to stop the computer from sending more data. If the clock/calendar is being displayed, the message "SCROLL" appears on the 25th line.

The SCROLL key will allow one more line feed to be received; shift-SCROLL will allow 24 more line feeds. Ctl-SCROLL will disable the Hold Screen mode entirely, so scrolling takes place normally.

Hold Screen mode won't work unless your computer and/or software supports hardware or software handshaking. If it does not, the terminal may appear to "hang up", or sound a continuous "beeeeeeep".

CHARACTER SETS

The character sets are pretty much self-documenting, but here are some hints. The stock Heath H19 character set has a few improvements. The graphic "j" is now the missing 2x2 checkerboard pattern (like the Z29 and Z100). Graphic "g" is now a 1/4-dot fill (3/4-dot fill in reverse video). This is useful for bar charts and area fills, used by software like Spectre Technologies "Rembrandt".

The SUPERFONT's extended VT-100 set has two novel features. First, the graphics characters actually spell out the names of the ASCII control codes. This is very useful for software debugging when used with transparent mode.

Second, the reverse-video attribute selects double-wide characters; <space> thru <underline> which includes 0-9, A-Z, and most punctuation. Each character has a left- and a right-half. The character itself displays its own left half. The right half is either the lower-case version of the same letter (if ASCII character 64-95, add 32); or a graphic version (if ASCII 32-63, add 64 and enable graphics). Example: "Aa" displays "A"; "5u" with graphics enabled is "5".

The IBM PC set emulates the monochrome display adapter (not the more common color/graphics adapter). It combines text, Greek letters, and line graphics all in the same set. It lets you run the H19 as a "slave" on a multi-user PC, or on an H-1000 to improve PC compatibility. It forms nice looking characters in a 7x9 dot matrix instead of 5x7, and is "bolder" because it uses dot pairs.

The APA graphics set has no text characters, so it looks bizzare without special software. But resolution is adequate for a 40-character x 16-line text display (or 60 characters with a proportional font). I would dearly love to get someone to write a text/graphics editor to use it.

The bits are mapped in a 4-high by 2-wide matrix. Look at the hex value of the ASCII character received; the high digit is the left 4 bits, and the low digit is the right 4. For example, ASCII "A" is 41 hex; the "4" is 0100 binary which defines the left 4 bits. Top bit is off, 2nd from top is on, 3rd and 4th off. The "1" in "41" is 0001, so on the right half of the character only the lower bit is on. ASCII is a 7-bit code, so use reverse video to turn the top-left bit on (which also reverses the rest of the bits, to make life interesting).

SERIAL DATA FORMAT

ASCII is a 7-bit code, while the H8, H89, Z90 etc. are 8-bit computers. The usual solution is to ignore the 8th bit. The computer sets it to zero on characters it sends, and ignores it on those it receives.

The H/Z-19 terminal is normally set for 8-bit data, even though it is not used. On the SUPERSET, the 8th bit can be put to good use. 8-bit data is enabled by setting DIP switch S401 #5 "up" (see "DIP switch settings").

RECEIVE: Characters with their 8th bit set are displayed with the current attribute; in reverse video, blinking, or as a character in an extended set (with SUPERFONT). Control codes with the 8th bit set are displayed as their corresponding graphics character. You can thus display reverse video and graphics without preceeding them with an ESC sequence. You can also "type filename.txt" to see if a file (like those created by Wordstar) has characters with the 8th bit set.

In transparent mode, all normal control code processing is suspended, so every 8-bit character received displays a unique character on-screen (1 of 256).

TRANSMIT: The keyboard normally sends only 7-bit data. If the native keyboard mode is enabled, the keyboard sends function keys as a single 8-bit code (see "Native Keyboard Mode" for a list of the actual codes).

Occasionally a program may send characters with the high bit set by accident (or more likely by sloppy design). Such characters will appear in reverse video or graphics. For example, my copy of ZCPR displays the "?" in its error messages in reverse video ("type no-such" displays "NO-SUCH?").

If this is troublesome, set S401 #5 "down" to select 7-bit words. This will force the 8th bit to 0 on all characters received or transmitted.

ESCAPE CODES

This chapter describes the SUPERSET from a programmer's viewpoint. It assumes that you are already familiar with normal H/Z-19 operation, and have a copy of its manuals. Therefore, only new or modified ESCape sequences are described.

CHARACTER SET

Heath: ESC <space> <Code>

ANSI: not available

The H/Z19 has 128 characters (95 alphanumeric and 33 graphic), displayable in normal and reverse video. The SUPERSET expands this to four 128-character sets. Changing sets alters every character on the screen, and all future characters received. The sets only change what you see on the screen; they have no effect on screen dumps or any other terminal operation. The sets are independent of each other and cannot be mixed; for instance, GT-PROM text with H19 graphics.

The H-19 set is selected at power-up or shift-RESET. No other command (including ESC z) alters the selected character set. The ESC sequence is the same as the Super-19 and works with Textpro and other Super-19 compatible software.

Heath: ESC <space> <Code>

Code	Title	Description
D	H-19	Standard H19 text and graphics character set. The : ; . - + and ! are moved slightly to improve appearance when drawing boxes. The ÷ (graphic "j") is replaced by % (same as Z-100), and the ± sign (graphic "g") is replaced by ░ (1/4-dot fill).
L	GT-PROM	NORCOM "T-PROM" enhanced text character set, which provides a BOLDER display font; and the "G-PROM" enhanced graphic set, which displays dots in a 5x2 cell (though not all combinations).
H	VT-100	DEC VT-100 text characters, with VT-100 "special graphics set". This is a fine-line equivalent to the H19's double-weight line graphics.
P	MATH/GREEK	Modified H/Z text characters, with the Greek alphabet and math symbols as the graphics set.

The character sets are shown in Appendix A. The optional SUPERFONT expands the number of character sets from 4 to 8. Additionally, each of the four new sets has 256 characters instead of 128. The four additional sets are:

C	H29	Heath H29 terminal, which adds super- and subscripts.
G	VT-100+	double-wide characters, displayable control characters.
K	APA-graphics	160w x 100h resolution all-point-addressable graphics.
O	IBM-PC	PC monochrome display adapter character set.

Character set size is set automatically when it is selected; it's not necessary to use an "ESC a" or "ESC s" command (see NORMAL/EXTENDED CHARSET below). When an extended set is selected, the additional 128 characters are accessed by the attribute bit normally used for reverse video (see ATTRIBUTE CONTROL below). "ESC p" causes subsequent characters to be displayed from the upper half of the character set; "ESC q" causes subsequent characters to come from the lower half.

Example:

The following BASIC program writes all of the displayable characters in 1) non-graphic, normal video; 2) graphic, normal video; 3) graphic, reverse video; 4) non-graphic, reverse video. After momentarily pausing, it changes the character set from SPECIAL, to VT-100, to GT-PROM, and exits in the H-19 set.

Notice that the entire screen changes character sets at once; no new characters need to be written. The program ends back in the H19 set.

```
 10 E$ = Chr$(27):      REM E$ = <ESC>
 15 Print E$;"v";:      REM auto-wrap mode active
 20 Print E$;" P";:     REM switch to SPECIAL set
 30 Gosub 200:          REM write characters out
 40 Print E$;"F";:      REM enter graphics mode
 50 Gosub 200
 60 Print E$;"p";:      REM enter reverse video mode
 70 Gosub 200
 80 Print E$;"G";:      REM exit graphics mode
 90 Gosub 200
100 Print E$;"q";:      REM exit reverse video mode
110 Gosub 300:          REM pause
120 Print E$;" H";:     REM switch to VT-100
130 Gosub 300
140 Print E$;" L";:     REM switch to GT-PROM
150 Gosub 300
160 Print E$;" D";:     REM return to H19
170 Stop:               REM stop program
180 REM
190 REM Print all displayable characters
200 REM
210 For X = 33 to 128:  REM print all displayable
220   Print Chr$(X);:   REM characters
230   Next X
240 Print: Print:       REM move to next line
250 Return
260 REM
270 REM Pause for a moment
300 REM
310 For X = 1 to 10000
320   Next X:           REM dummy loop for pause
930 Return
940 End
```

NORMAL/EXTENDED CHARSET

```
Heath: normal 128-char set       ESC a
       extended 256-char set     ESC s

 ANSI: not available
```

Use these commands only with the optional SUPERFONT. Normal sets have 128 characters (H19, VT-100, GT-PROM, and Special). The SUPERFONT adds extended 256-character versions (H29, VT-100+, APA-graphics, and IBM-PC respectively).

The command "ESC a" is the default, and selects a 128-character set with the reverse video and blink attribute working normally. "ESC s" selects the corresponding 256-character set. If you were using the H19 set for example, "ESC s" switches to the H29 set, and "ESC a" switches back to the H19.

The upper 128 characters are selected by the attribute bit normally used for reverse video. "ESC p" displays subsequent characters in the top half of the set; "ESC q" displays the lower half. For example, an H19 reverse-video "P" becomes the H29's superscript "P"; a reverse-video "p" becomes a subscript "P".

These commands are Super-19 compatible, and work with software designed for it. If you do not have the SUPERFONT and are running Magnolia CP/M, don't select the extended character set. It will set the DTR line low, and "hang" the system.

ATTRIBUTE CONTROL

```
Heath: enable attribute     ESC p
       disable attribute    ESC q
       attribute=blink      ESC x <
       attribute=reverse    ESC y <
       attributes on        ESC x :
       attributes off       ESC y :

 ANSI: enable attribute     ESC [ 7 m
       disable attribute    ESC [ m    or   ESC [ 0 m
```

An "attribute" affects how a character is displayed. The standard H19 has only one attribute (reverse video), which is always on (available). "ESC p" enables the attribute, so all subsequent characters will be displayed in reverse video. "ESC q" disables the attribute; subsequent characters display in normal video.

The SUPERSET has two attributes; reverse video, and blink. Reverse video is the default attribute, selected by "ESC y <". The blink attribute "ESC x <" causes all characters received between the "ESC p" and "ESC q" to blink between normal and reverse video. This can display a flashing "WARNING" message, for example.

A third attribute is available with the SUPERFONT. The 256-character sets use the attribute to select between the upper and lower halves of the set. "ESC p" displays subsequent characters in the upper 128; "ESC q" displays the lower 128.

The H19 has only one bit of memory per character available for attributes. Thus only one attribute at a time can be active; reverse video, blink, or extended character set. If an extended set is selected with attributes on, characters in the upper half will also be reversed or blinking. The command "ESC y :" turns the reverse video and blink attributes off; "ESC x :" turns them on.

WHITE SCREEN

Heath:	white screen	ESC h 5
	normal screen	ESC i 5

ANSI: not available

A normal H19 displays bright characters on a black background. Researchers claim that a bright screen with dark characters is more readable and causes less eyestrain. Now you can find out for yourself.

"ESC i 5" is the default, dark background with bright characters. "ESC h 5" selects a bright screen with dark characters. Operation is otherwise unaffected; it has no effect on reverse video or transmit character/line/screen functions.

Note: A properly adjusted H19 looks VERY good. But white screen operation will make any deficiencies in your CRT plainly visible; geometric distortion, power supply ripple, phosphor burns, bad grounding, ringing, etc.

INTERLACE MODE

Heath:	interlace	ESC h 9
	normal	ESC i 9

ANSI: not available

The H19 normally displays its data using 250 horizontal scan lines, repeated 60 times per second. This is coarse enough that you can see the individual lines making up each character.

Interlace mode doubles this to 499 scan lines, repeated 30 times per second. This improves appearance by "filling in" the gap between scan lines, and is preferred for photographing the screen. The Super-19 command "ESC h 9" enables interlace mode, and "ESC i 9" disables it.

Since interlace mode repeats each screen half as often, a noticeable "flicker" can be observed, particularly with white tubes and high brightness. The flicker is greatly reduced at low brightness levels, and by using a longer-persistence (green or amber) tube.

The SUPERSET's interlace mode is improved over the Super-19 and Ultra-ROM by more accurate timing control. Clock accuracy is unimpaired by interlace mode.

SCREEN-SAVER

Heath:	screen-saver on	ESC h :
	screen-saver off	ESC i :

ANSI: not available

The screen-saver automatically blanks the screen after 15 minutes of inactivity. This prolongs the life of your display, especially long-persistence phosphors. The screen is restored by receiving any character from the computer, or by hitting any key on the keyboard. The CTRL, left SHIFT, and REPEAT keys are good choices because they don't otherwise do anything by themselves.

ON-SCREEN CLOCK

Heath:	set time	ESC X <hh> <mm> <ss> <CR>	
		ESC : <hh> <mm> <ss> <CR>	
	read time	ESC e	responds with "<hh><mm><ss>"
		ESC ;	responds with "ESC:<hh>:<mm>:<ss>"
	display clock	ESC c	
	no clock display	ESC d	

ANSI: not available

A clock is displayed in the lower right-hand corner on the 25th line. The clock keeps 24-hour or 'military' time: midnight is 00:00:00, 3:15 am is 03:15:00, and 7:45 pm is 19:45:00. The clock can be set from the keyboard as follows:

1. press the off-line key so it stays down
2. press the ESC (escape) key
3. press the upper-case X key (shift-X)
4. enter 2 digits for the hour (00-23)
5. enter 2 digits for the minutes (00-59)
6. enter 2 digits for the seconds (00-59)
7. hit the SPACE bar to start the clock
8. press the off-line key so it pops up

The clock stops while the time is entered. The display is updated as you type the hour, minutes and seconds. Any displayable non-digit is ignored, so you can type a colon between the hour, minutes, and seconds; "ESC X 10:30:00" is a valid entry. Any control key aborts the command. Any key after the 6th digit ends the sequence and starts the clock. The SPACE bar was used for convenience.

The commands for setting the clock are shown in the ctl-ESC menus for reference only; you cannot set the time from the menu. However, the clock can be set by the computer. This BASIC line sets the clock to 18:34:00, or 6:34 pm.

```
PRINT CHR$(27);"X183400"
```

The clock is accurate to ±10 sec/day, and keeps time as long as the terminal is turned on (even through RESET). The optional SUPERCLOCK improves accuracy to ±1 min/month, and has a battery so the clock stays accurate continuously.

The clock/calendar display can be disabled (for a blank 25th line) by "ESC d", and enabled by "ESC c". The clock need not be disabled to use the 25th line; it is automatically removed when the 25th line is enabled. If the clock is enabled, the clock display will reappear when the 25th line is disabled.

The time can be read regardless of whether it is displayed or not. "ESC e" sends the time in HUG/Watzman format as a 6-digit number; 13:38:47 is sent as "133847". "ESC ;" sends the time in Super-19 format as an ESC sequence with colons; thus 13:38:47 is sent as "ESC:13:38:47". Note that if the computer echoes this sequence to the terminal, it can reset the clock back a second.

You can send the time via local function key "ctl-WHITE". The time is sent with colons, i.e. 13:38:47 is sent as "13:38:47". This lets you insert the time into any text being edited with a single keystroke.

There are no ANSI codes for any of the clock functions.

ON-SCREEN CALENDAR

Heath:	set date	ESC | <mm> <dd> <yy> <CR>	
	send date	ESC ~	responds with "ESC | <mm>:<dd>:<yy>"
	display calendar	ESC c	
	no calendar display	ESC d	

ANSI: not available

The calendar is displayed in the lower right-hand corner on the 25th line, just left of the clock. The commands are the same as the Super-19. The date can be set from the keyboard as follows:

1. press the off-line key so it stays down
2. press the ESC (escape) key
3. press the | key (vertical bar, or shift-backslash)
4. enter 2 digits for the month (01-12)
5. enter 2 digits for the date (01-31)
6. enter 2 digits for the year (00-99)
7. press the off-line key until it pops up

The display is updated as you type. Until the last digit, any displayable non-digit is ignored, so a slash may be typed between the month, day, and year; thus "ESC | 12/25/87 <ret>" is a valid entry. Any control key, or the 6th digit typed ends the command.

The command for setting the calendar is displayed by the "ctl-ESC" menus for reference only; you cannot set the date from the menus. However, the date can be set by the computer with this sequence. This BASIC line sets it to 12/25/87:

```
PRINT CHR$(27);"|122587"
```

The calendar displays dates in "American" format; month/day/year. At midnight the day is automatically incremented, although no carry is generated at the end of the month; 2/28/87 increments to 2/29/87, and 12/31/87 goes to 12/32/87. The date is maintained as long as the terminal is turned on (even through RESET).

The optional SUPERCLOCK provides a battery backup so the clock/calendar remains accurate even with the power off. It also keeps a proper calendar, and handles carrys for 28, 29 (leap year), 30, and 31-day months, and years to 1999.

The clock/calendar display can be disabled (for a blank 25th line) by "ESC d", and enabled by "ESC c". The calendar need not be disabled to use the 25th line; it is automatically removed when the 25th line is enabled. If the clock is enabled, the calendar will reappear when the 25th line is disabled.

The calendar can be read two ways, regardless of whether it is currently being displayed or not. "ESC ~" sends the date in Super-19 format, as an ESC sequence with colons. For example, 12:25:87 would be sent as "ESC|12:25:87".

Second, you can send the date via local function key "ctl-shift-WHITE". The date is transmitted with slashes; 12/25/87 is sent as "12/25/87". This lets you insert the date into any text being edited with a single keystroke.

There are no ANSI codes for any of the calendar functions.

HARD/SOFT RESET

```
Heath: reset terminal       ESC z
       set "hard" ESC z     ESC i ;
       set "soft" ESC z     ESC h ;

 ANSI: hard reset terminal  ESC [ z
```

The "ESC z" command resets the terminal to either of two states. A "hard" ESC z resets the terminal to its power-on configuration, as set by the DIP switches. The screen is cleared, the cursor homed, and any characters received during the reset (immediately after the ESC z) are lost. Any baud rate changes are lost. All modes which have no DIP switch setting (reverse video, graphics, interlace, white screen, transmit speed limiting, etc.) are reset. This "hard" ESC z is the default, or is activated by an "ESC i ;" command.

The SUPERSET provides a second set of "software" DIP switches. The configuration can be saved in them at any time by an "ESC R <Pn>" command (see SAVE/RESTORE MODES). The modes saved in these switches are:

```
 reverse video: on/off
      graphics: on/off
type of cursor: on/off/steady/blinking/block/underline
   insert mode: on/off
      keyboard: shifted/unshifted/alternate/native
   hold screen: on/off
     key click: on/off
at end of line: wrap/discard
 auto LF on CR: on/off
 auto CR on LF: on/off
 ESC sequences: Heath/ANSI
```

A "soft" ESC z is activated by an "ESC h ;" command. Any subsequent ESC z resets the terminal to the last saved configuration. The screen is cleared and the cursor homed. Characters received during the reset are handled normally. Any modes NOT listed above are left unchanged; baud rate, interlace mode, white screen, transmit speed limiting, character set, attributes, etc.

This is useful for programs like PIE or Textpro that send an "ESC z" when they exit. Setup the SUPERSET in the desired mode; for example white screen, no key click, and a new baud rate. Save this configuration with "ESC R 4", and enable a soft reset with "ESC h ;". Now when the program exits, the SUPERSET will be restored to the desired configuration.

SAVE/RESTORE/SWAP MODES

Heath:	restore modes and screen	ESC R 1	(ctl-RED)
	save modes and screen	ESC R 2	
	restore modes	ESC R 3	
	save modes	ESC R 4	
	restore screen	ESC R 5	
	save screen	ESC R 6	
	swap modes and screen	ESC R 7	
	swap screen	ESC R 8	(ctl-shift-RED)
	swap modes	ESC R 9	

ANSI: not available

Most H19 modes can't be read by a remote computer (graphics on/off, keyboard shifted/unshifted, etc.). A program thus can't make full use of the terminal without destroying its previous configuration. For example, a subprogram may need to interrupt a main program, use the terminal, and then somehow restore the terminal screen and modes to their original state.

The SUPERSET maintains two sets of mode flags. The "ESC R <Pn>" commands allow the two sets of flags to be saved, restored, or swapped internally. <Pn> represents an ASCII digit 1-9 which defines the parameters acted upon. These commands are similar to the Ultra-ROM. See SOFT/HARD RESET for a description of the modes saved and restored.

SAVE/RESTORE SCREEN

Heath:	restore modes and screen	ESC R 1	(ctl-RED)
	save modes and screen	ESC R 2	
	restore modes	ESC R 3	
	save modes	ESC R 4	
	restore screen	ESC R 5	
	save screen	ESC R 6	
	swap modes and screen	ESC R 7	
	swap screen	ESC R 8	(ctl-shift-RED)
	swap modes	ESC R 9	

ANSI: not available

If the SUPERCLOCK is installed, a second page of screen memory is available. The entire screen (including the 25th line, if enabled) can be saved, restored, or swapped with the 2nd page of memory. If the SUPERCLOCK is not installed, the "screen" portion of the command is ignored. The "modes" portion of the command functions as described above.

Local function keys are provided for two of the frequently used options. The ctl-RED key performs a "soft" reset to a previously saved state. Ctl-shift-RED povides a quick way to peek at previously saved data (a "help" screen, disk directory, etc.). The first ctl-shift-RED displays the saved data; a second ctl-shift-RED then restores the original screen.

WINDOWS

Heath: ESC m <topline><leftcol><#lines><#cols><page2line><page2col>

ANSI: not available

If the SUPERCLOCK is installed, a 2nd page of screen memory is available. The "ESC m" command uses it to copy user-defined "windows" between on-screen and off-screen memory. This lets you create your own windows that "pop-up" instantly and restore the screen when done, like the help menus.

"ESC m" defines a window to be swapped between on-screen and off-screen memory. Windows can be anything from a single character to the entire screen. Both on- and off-screen windows are the same size, but need not be in the same positions.

The "ESC m" command is followed by 0 to 6 parameters. A parameter is a printable ASCII character, <space> to <o>, representing a number 1 thru 80 respectively. The same method is used in the "ESC Y <line><column>" cursor position command. All parameters are optional; a <RETURN> or other out-of-range character will end the command early and execute it with defaults for the missing parameters.

CANCEL (ctl-X) aborts without action. If no parameters are supplied ("ESC m <RET>"), the entire screen is swapped with page 2 (including the 25th line, if enabled). This is in effect the same as a swap screen command, "ESC R 8".

The first four parameter define a rectangular region of the main screen. <topline> defines the top line of the window; the default is line 1 (the top line). <leftcol> is the leftmost column of the window; the default is column 1. <#lines> is the height of the window in lines (space=1, !=2, etc.); the default is from <topline> to end of screen. <#cols> is the width in columns (space=1, etc.); the default is from <leftcol> to column 80. If a window is specified past the edge of the screen, the defaults are used.

The last two parameters define the top right corner of an identically-sized region of page-2 memory. <page2line> is the top line; the default is the same line specified by <topline>. <page2col> is the leftmost column; the default is the same as <leftcol>. If the page 2 parameters define a window past the edge of the screen, the defaults are used.

Example 1: Pull-Up Menus

This program displays labels for the function keys on the 25th line. When a function key is hit, a "pull-up" menu appears above the chosen key with additional information.

First, set up a menu screen. The top line of each menu item will become the 25th line, and the text below it will be the pull-up menu. The menus are saved on page 2. To display a menu item, swap it to the screen with an "ESC m ..." command. To restore the screen, the same "ESC m ..." command swaps it back.

Example 2: Moving Objects

"ESC m" commands are normally used in pairs; first to display part of page 2, and second to save it and restore the screen. This is a powerful technique; a "sprite" can be defined on page 2, and moved rapidly across a stationary background. Display it; pause; save it; then repeat with the sprite in a new location. The second example uses this method to show a bouncing "smile" face.

```
10 REM Program to demonstrate pop-up "windows" with the SUPERCLOCK
20 WIDTH 255: E$=CHR$(27): PRINT E$;"p"
30 PRINT E$;"Y1(";"  f1-FONT      f2-STYLE      f3-MARGINS   ": REM our menu
40 PRINT E$;"Y2(";" Select FONT: Select STYLE: Set MARGINS: "
50 PRINT E$;"Y3(";"   1-H19        1-normal      1-left      "
60 PRINT E$;"Y4(";"   2-VT-100     2-reverse     2-right     "
70 PRINT E$;"Y5(";"   3-GT-PROM    3-interlace   3-top       "
80 PRINT E$;"Y6(";"   4-IBM PC                   4-bottom    "
90 REM
100 PRINT E$;"R6";E$;"E";:            REM save it on page 2, and clear screen
110 PRINT E$;"y1";E$;"x1";:           REM enable and clear 25th line
120 PRINT E$;"m8# H1(";:REM window to 25th line="8" column 4="#" one line=" "
130 REM                   40 cols wide="H" from page2 line 18="1" col 9="("
140 REM
150 PRINT E$;"q";"Hit f1, f2, f3, or control-C to end"
160 A$=INKEY$:IF A$<>CHR$(27) THEN 160:REM wait for an ESC
170 A$=INKEY$:IF LEN(A$)=0 THEN 170: REM then get next key
180 REM
190 ON ABS(ASC(A$)-82) GOTO 220,270,320: REM is it f1,f2, or f3?
200 GOTO 160
210 REM                               display f1 menu
220 PRINT E$;"m4#$-2(";:              REM show 5-line 14-char FONT menu
230 A$=INKEY$:IF LEN(A$)=0 THEN 230:  REM wait for a key
240 PRINT E$;"m4#$-2(";:              REM save FONT menu
250 GOTO 160
260 REM                               display f2 menu
270 PRINT E$;"m50#.25";:              REM show 4-line 15-char STYLE menu
280 A$=INKEY$:IF LEN(A$)=0 THEN 280:  REM wait for a key
290 PRINT E$;"m50#.25";:              REM save STYLE menu
300 GOTO 160
310 REM                               display f3 menu
320 PRINT E$;"m4>$-2C";:              REM show 5-line 14-char MARGINS menu
330 A$=INKEY$:IF LEN(A$)=0 THEN 330:  REM wait for a key
340 PRINT E$;"m4>$-2C";:              REM save MARGINS menu
350 GOTO 160
```

```
10 REM display a bouncing "smile" face
20 REM
30 WIDTH 255: E$=CHR$(27): PRINT E$;"j";E$;"H";E$;"F";E$;"p";E$;"R6";
40 PRINT "r      _"
50 PRINT "  ^  ^ ¯";                     REM display smile face
60 PRINT " eaaaad "
70 PRINT "_      ";E$;"qr";CHR$(13);"_";E$;"G";E$;"k";
80 REM
90 X=32: Y=32: STEPX=4: STEPY=2
100 PRINT E$;"m";CHR$(Y);CHR$(X);"#'  ";: REM save face
110 IF LEN(INKEY$)>0 THEN END
120 X=X+STEPX: Y=Y+STEPY:                REM move it
130 PRINT E$;"m";CHR$(Y);CHR$(X);"#'  ";: REM display in new spot
140 IF X>103 OR X<33 THEN STEPX=-STEPX: PRINT CHR$(7);
150 IF Y>51 OR Y<33 THEN STEPY=-STEPY: PRINT CHR$(7);
160 GOTO 100:                            REM "bounce" off edges  loop
```

The window command takes approximately 30 milliseconds per 2000 characters, allowing 30 complete screen swaps per second. Speed is proportional to the size of the window being swapped, so fast animation is quite practical.

TRANSMIT SPEED LIMITING

Heath: limit speed ESC h 3
full speed ESC i 3

ANSI: not available

The SUPERSET normally sends data as fast as the current baud rate allows. When transmitting long strings of characters, this may be too fast for the host computer to keep up. This usually shows up with the Transmit Current Line, Transmit 25th line, Transmit Edited Line, and Transmit Page functions.

The Super-19 command "ESC h 3" limits transmission speed for these commands to 60 characters/sec. Characters are still sent at the full baud rate; idle time is simply inserted between the characters. Transmit speed limiting is disabled (the default) by "ESC i 3". Also see HANDSHAKING to limit speed via hardware handshaking.

TRANSMIT PAGE

Heath: ESC #

ANSI: ESC [p

Transmits lines 1-24 to the computer. A moving cursor indicates the character being sent. If reverse video or graphic characters are encountered, the are preceeded by the appropriate Heath or ANSI ESC sequences. A single <return> is sent at the end of the page, and the bell sounds.

Most programs cannot accept such a large block of data at full speed. Transmit page sends at least 1921 characters, and can send up to 19,201 (ANSI mode, with every other character reverse video graphics). If transmit speed limiting is on, transmission is limited to 60 characters/sec. Otherwise the page is sent at the full baud rate, limited only by hardware handshaking.

The following commands will "capture" the screen in a disk file under CP/M. PIP creates the file SCREEN.PIC and then puts everything into it that comes from the console until the final control-Z (end of file).

1. ctl-ESC h 3	enable transmit speed limiting
2. PIP SCREEN.PIC=CON:	tells PIP to create
3. ctl-ESC #	tell Superset to transmit screen
4. (wait for final "beep")	screen sent to PIP
5. ctl-Z	tells PIP to end the file

TRANSMIT 25TH LINE

Heath: ESC]

ANSI: ESC [q

Transmits the 25th line to the computer. A moving cursor indicates the character being sent. Reverse video and graphic characters are preceeded by appropriate ESC sequence. A <return> is sent at the end of the line. If the 25th line is disabled, only the carriage return is sent.

If transmit speed limiting is on, transmission is limited to 60 chars/sec. Otherwise the line is sent at the full baud rate, limited only by hardware handshaking.

TRANSMIT EDITED LINE

Heath: ESC ? (local function ctl-RETURN)

ANSI: not available

Transmits the characters from the cursor position to the end of the line. Trailing spaces are not sent. A moving cursor indicates the character being sent. Reverse video, blinking, or characters in the upper half on an extended set are sent with their 8th bit high. Graphic characters are sent as their equivalent control code. The line ends with a <return>.

If transmit speed limiting is on, transmission is limited to 60 chars/sec. Otherwise the line is sent at the full baud rate, limited only by hardware handshaking.

This command is useful for performing local editing. For instance, a CP/M command can be re-executed by holding down the control key and moving the cursor back to the first character of the command. Then type ctl-RETURN to transmit this line to CP/M.

The Transmit Edited Line is also useful when prompting an operator for input. Simply echo every key he types back to the terminal, until a <return>. Then your program moves the cursor to the start of the command and sends "ESC ?" to read the line. This method lets the user edit his input line with not only Backspace, but insert, delete, and cursor controls as well.

TRANSMIT CURRENT LINE

Heath: ESC ^ (last character is the caret, or shift-6)

ANSI: ESC [1 p

The entire line the cursor is currently on is transmitted to the computer. A moving cursor indicates the character being sent. If there are graphics or reverse video on the line, the appropriate escape sequences are also sent. All trailing spaces are sent. The line is ended with a carriage return.

The following are examples of a screen dump program in BASIC. It prints a copy of the screen on your printer (text only; not reverse video or graphics).

for CP/M:

```
100 REM Print Screen using SUPERSET transmit line
110 E$ = CHR$(27);
120 PRINT E$;"j";:            REM save cursor position
130 PRINT E$;"H";:            REM home cursor to top line
140 FOR L = 1 TO 24:          REM loop for 24 lines
150   PRINT E$;"^";:          REM transmit line
160   L$=INPUT$(80):          REM get sent line
170   LPRINT L$:              REM print line out to printer
180   IF I<24 THEN PRINT:     REM move cursor to next line
190   NEXT I:                 REM continue with loop
200 PRINT E$;"k";:            REM put cursor in original position
```

for HDOS:

```
100 REM Print Screen using SUPERSET transmit line
105 OPEN "O",1,"LP:"          REM output file #1 is LP.DVD
110 E$ = CHR$(27);
120 PRINT E$;"j";:            REM save cursor position
130 PRINT E$;"H";:            REM home cursor
140 FOR L = 1 TO 24:          REM loop all 24 lines
150   PRINT E$;"^";:          REM request transmit line
160   L$=INPUT$(80):          REM get sent line
170   PRINT #1,L$:            REM print line out to printer
180   IF I<24 THEN PRINT:     REM move cursor to next line
190   NEXT I:                 REM continue with loop
200 PRINT E$;"k";:            REM put cursor in original position
205 CLOSE #1
```

To use these programs as subroutines in your own programs:

1. add '210 RETURN' at the end of the routine.
2. increase the line numbers to a number after the end statement of your program.
3. add 'GOSUB nnnn' in your program where you want to dump the screen. The nnnn represents the first line number of the screen dump routine.

The line is sent at the current baud rate,. which may be so fast that BASIC misses characters. If so, the speed can be slowed down by enabling transmit speed limiting (ESC h 3). The line will then be sent at 60 characters/sec.

TRANSMIT CURRENT CHARACTER

Heath: ESC _   (last character is the underline, shift-dash)

ANSI: ESC [2 p

The character at the current cursor position is transmitted to the computer. If a graphic character, it is transmitted as the corresponding control code. For example a Heath graphic "a" (—) is transmitted as a control-A. If a reverse video character, the character is sent with its 8th bit set.

Examples: The character "A" is transmitted as an ASCII "A" (101 octal, 41 hex). The graphics 'i' (gray box in the H19 character set) is transmitted as a control-I (011 octal, 09 hex). A reverse video "space" (white box) is transmitted as 240 octal, A0 hex.

If the terminal is set to communicate with 7-bit words, it will not be possible to detect reverse video. Also, HDOS and CP/M both remove the 8th bit on received characters. You must either use CP/M's CONIO BDOS function, or read the console UART directly to detect reverse video.

The following BASIC example uses the transmit character function to get a character, in reverse-video or graphics, by-passing the operating system. When it is running, it prints a test line containing graphic characters. With the arrow keys, move the cursor to any location on the screen. Pressing the period key twice prints the character where you left the cursor.

```
 10 PRINT "Transmit character at cursor demonatration."
 20 PRINT: PRINT "Use the arrow keys to move the cursor"
 30 PRINT "to the character you wish to transmit, and"
 40 PRINT "press the period ('.') key twice."
 41 REM
 42 REM shift the keypad
 43 REM
 50 E$ = CHR$(27);: PRINT E$;"x6";
 60 PRINT: PRINT "Test line "E$"Fih~r"E$"G"E$"pX"E$"c";
 70 C$ = INPUT$(2): IF C$ = ".." THEN 90
 80 PRINT C$;: GOTO 60
 90 REM
 91 OUT &0351,0:                       REM for H89 or H8 with H8-4
 92 REM OUT &0373,$025:                REM for H8 with H8-5
100 PRINT E$CHR$(95);:                 REM transmit character
110 REM
111 C=INP(&0350):                      REM for H89 or H8 with H8-4
112 C=INP(&0372):                      REM for H8 with H8-5
120 C$=CHR$(C)
130 PRINT E$"E": PRINT "The Character was ";
140 IF C>127 THEN PRINT E$"p";
150 IF (C AND 127)<32 THEN PRINT E$"F";:C=C+96:C$=CHR$(C);
160 PRINT C$E$"q"E$"G"E$"y6":          REM clean up
170 OUT &0351,1:                       REM for H89 or H8 with H8-4
171 REM OUT &0373,&027):               REM for H8 with H8-5
180 END
```

SET BAUD RATE

Heath: ESC r <parameter>

ANSI: ESC [<parameter> r

The SUPERSET adds 3 new baud rates to the terminal: 75, 134.5, and 38,400 baud. Note that using a higher baud rate won't necessarily speed up your screen response; a CP/M computer running at 2 MHz only sends about 1,100 characters/sec. The greatest benefit is achieved when the computer is also speeded up.

Heath <parameter>		Baud Rate		ANSI mode <parameter>
@		75		0
A		110		1
B		150		2
C		300		3
D		600		4
E		1200		5
F		1800		6
G		2000		7
H		2400		8
I		3600		9
J		4800		10
K		7200		11
L	*	9600	*	12
M		19,200		13
N	#	38,400	#	14
O	#	134.5	#	15

key: # represents a new baud rate
* represents the standard baud rate

FULL/HALF DUPLEX

Heath: half duplex ESC x >
full duplex ESC y >

ANSI: half duplex ESC [> 14 h
full duplex ESC [> 14 l (last character is lowercase 'L')

This function selects full duplex or half duplex operation. Full duplex is the standard mode used by Heath/Zenith software. In full duplex, keys typed on the keyboard are simply sent to the computer; they do not appear on the screen unless the computer "echoes" them back to the terminal.

Half duplex is sometimes useful when communicating with another system using a modem. In half duplex, each key typed on the keyboard appears on the screen AND is sent to the computer. If the computer echoes the key, it will therefore appear on the screen twice.

HANDSHAKING

Heath:	disable XON/XOFF	ESC x =	
	enable XON/XOFF	ESC y =	
ANSI:	disable XON/XOFF	ESC [> 13 h	
	enable XON/XOFF	ESC [> 13 l	(last is lowercase 'L')

"Handshaking" lets the terminal and computer warn each other when they aren't ready to receive. The H/Z-19 had XON/XOFF receive handshaking, but it was buggy and seldom used. There was no transmit handshake (no limit on how fast data was sent to the computer), so transmit screen sent data too fast to be useful.

The SUPERSET only needs handshaking to avoid missing received characters while the menus are displayed; for slow commands like save/restore/swap screen; or at 38400 baud with super-fast computers (8MHz H-1000 etc). There is a 128-character input buffer, for data received but not yet processed. If the buffer reaches 80 characters, the SUPERSET tells the computer to stop sending. When it gets below 32, the SUPERSET says to resume sending. If your computer ignores handshaking and the buffer exceeds 128, a long "beeeeep" warns that characters were lost.

If software XON/XOFF handshaking is enabled (the default), an "XOFF" is sent (ASCII DC3, ctl-S, hex 13) by the SUPERSET to stop sending, and an "XON" (ASCII DC1, ctl-Q, hex 11) to start again. This is the method used by CP/M and HDOS.

Many programs don't support XON/XOFF; e.g. in Wordstar, ctl-S moves the cursor left. For such programs XON/XOFF handshaking can be disabled with "ESC x =". The SCROLL key and hold screen mode still function with XON/XOFF disabled.

signal name	pin	direction	description
DTR Data Terminal Ready	20	output: high	= terminal is on
		low	= terminal is off or disconnected
RTS Request To Send	4	output: high	= terminal is ready to receive
		low	= terminal not ready to receive
DSR Data Set Ready	6	input: high	= enables CTS sensing
		low	= disable CTS (always OK to send)
CTS Clear To Send	5	input: high	= enables terminal to send data
		low	= disables transmission of data

Hardware handshaking is always enabled, but has no effect if not implemented by your computer. The RTS output tells the computer when it can send data. The CTS input tells the terminal when it can send data. To add hardware handshaking to the Heath CP/M BIOS, you must edit BIOS.ASM and create a new BIOS. At the label:

```
CRTOS1: CALL    UOS
        JZ      CRTOSB
        MVI     A,6     (*)     add the lines with the asterisks.
        CALL    PINX    (*)
        ANI     10H     (*)
        JZ      CRTOSB  (*)
```

Rather than add CTS handshaking, you can expand the type-ahead buffer. This lets slow languages like BASIC use "transmit line" without missing keys. Near the end of BIOS.ASM, find the label CRTBUF: and change the value from 40 to 80:

```
CRTBUF: DB      80      (*)     was 40, change it to 80.
```

NATIVE KEYBOARD

Heath: enter ESC x ? (local function ctl-shift-BLUE)
exit ESC y ? (local function ctl-shift-BLUE)

ANSI: enter ESC [> 15 h
exit ESC [> 15 l (last is lowercase 'L')

In this mode each key sends a unique code. The keypad sends control codes with the 8th bit set. Function keys send the code they normally send after ESC, but with the 8th bit set. Switch SW401 #6 must be =0 for the 8th bit to be sent.

	Key Pressed	Transmitted Code Hex	Octal	Decimal	Explanation
function key:	ERASE	CA	312	202	"J" + 8th bit
	BLUE	D0	320	208	"P" + 8th bit
	RED	D1	321	209	"Q" + 8th bit
	WHITE	D2	322	210	"R" + 8th bit
	f1	D3	323	211	"S" + 8th bit
	f2	D4	324	212	"T" + 8th bit
	f3	D5	325	213	"U" + 8th bit
	f4	D6	326	214	"V" + 8th bit
	f5	D7	327	215	"W" + 8th bit
function key:	shift-ERASE	C5	305	197	"E" + 8th bit
	shift-BLUE	A0	244	164	"$" + 8th bit
	shift-RED	A1	245	165	"%" + 8th bit
	shift-WHITE	A2	246	166	"&" + 8th bit
	shift-f1	A3	247	167	"'" + 8th bit
	shift-f2	A4	250	168	"(" + 8th bit
	shift-f3	A5	251	169	")" + 8th bit
	shift-f4	A6	252	170	"*" + 8th bit
	shift-f5	A7	253	171	"+" + 8th bit
keypad:	0	80	200	128	Control-A + 8th bit
	1	81	201	129	Control-B + 8th bit
	2	82	202	130	Control-C + 8th bit
	3	83	203	131	Control-D + 8th bit
	4	84	204	132	Control-E + 8th bit
	5	85	205	133	Control-F + 8th bit
	6	86	206	134	Control-G + 8th bit
	7	87	207	135	Control-H + 8th bit
	8	88	210	136	Control-I + 8th bit
	9	89	211	137	Control-J + 8th bit
	.	8A	212	138	Control-K + 8th bit
	ENTER	8B	213	139	Control-L + 8th bit
keypad:	shift-0	90	220	144	Control-P + 8th bit
	shift-1	91	221	145	Control-Q + 8th bit
	shift-2	92	222	146	Control-R + 8th bit
	shift-3	93	223	147	Control-S + 8th bit
	shift-4	94	224	148	Control-T + 8th bit
	shift-5	95	225	149	Control-U + 8th bit
	shift-6	96	226	150	Control-V + 8th bit
	shift-7	97	227	151	Control-W + 8th bit
	shift-8	98	230	152	Control-X + 8th bit
	shift-9	99	231	153	Control-Y + 8th bit
	shift-.	9A	232	154	Control-Z + 8th bit
	shift-ENTER	9B	233	155	ESCAPE + 8th bit

TRANSPARENT MODE

Heath:	enable	ESC x @	(local function ctl-BLUE)
	disable	not available	(local function ctl-BLUE)
ANSI:	enable	ESC [> 16 h	
	disable	not available	

In transparent mode, every character received produces a unique character on the screen. If a control code, it is displayed as a graphic symbol. If the high bit is set, the character is displayed in reverse video.

This is useful for diagnostic purposes, because it lets you see EVERY code being received by the terminal. Since RETURN and LINE FEED are just another displayable character, you should set the terminal to auto-wrap (ESC v) before entering the transparent mode. Otherwise any characters after the 80th will simply "pile up" at the end of the line.

The symbols displayed for control codes will vary with the character set (see Appendix A). The optional SUPERFONT VT-100+ character set is particularly easy to interpret because it has special graphic symbols for all 33 control codes.

This mode disables ALL (without exception) escape code processing. Since every possible code is displayed, there is no way for the computer to command an exit from this mode. It can only be exited by local function key ctl-BLUE, terminal reset ctl-DELETE, or system reset right-shift-RESET.

If 7-bit words are selected at SW401 #6, the 8th bit will be ignored. CP/M and HDOS also zero the 8th bit before sending a character. So they too will not send reverse video without special software. In CP/M this means using the direct console I/O (CONIO) BDOS call, or sending characters directly to the 8250 UART.

If you enable both transparent screen and native keyboard modes, each key will display a unique character. The f1 key for example sends an "S" with the 8th bit set (B3 hex), which displays a reverse-video "S" if 8-bit words are enabled.

CURSOR TYPES

Heath:	cursor off	ESC y 5	
	cursor on	ESC x 5	
	underline	ESC y 4	
	block	ESC x 4	
	blink	ESC y ;	
	non-blink	ESC x ;	
ANSI:	cursor on	ESC [> 5 l	(last is lowercase "L")
	cursor off	ESC [> 5 h	
	underline	ESC [> 4 l	(last is lowercase "L")
	block	ESC [> 4 h	
	blink	ESC [> 11 l	(digit "1", digit "1" lowercase "L")
	non-blink	ESC [> 11 h	

The standard H19 cursor options include cursor on/off, and block/underline cursor. The SUPERSET adds a blinking/non-blinking option. This provides five possibilities: no cursor, non-blinking block, blinking block, non-blinking underscore, and blinking underscore cursor.

WINDOWS

Heath: ESC m toplineleftcol#lines#colspage2linepage2col
       ESC m topline+32...   same, but sync to vertical retrace

ANSI: not available

If the SUPERCLOCK is installed, a 2nd page of screen memory is available. The "ESC m" command uses it to copy user-defined "windows" between on-screen and off-screen memory. This lets you create your own windows that "pop-up" instantly and restore the screen when done, like the help menus.

"ESC m" defines a window to be swapped between on-screen and off-screen memory. Windows can be anything from a single character to the entire screen. Both on- and off-screen windows are the same size, but need not be in the same positions.

The "ESC m" command is followed by 0 to 6 parameters. A parameter is a printable ASCII character, space to o, representing a number 1 thru 80 respectively. The same method is used in the "ESC Y linecolumn" cursor position command. All parameters are optional; a RETURN or other out-of-range character will end the command early and execute it with defaults for the missing parameters.

CANCEL (ctl-X) aborts without action. If no parameters are supplied ("ESC m RET"), the entire screen is swapped with page 2 (including the 25th line, if enabled). This is in effect the same as a swap screen command, "ESC R 8".

The first four parameter define a rectangular region of the main screen. topline defines the top line of the window; the default is line 1 (the top line). leftcol is the leftmost column of the window; the default is column 1. #lines is the height of the window in lines (space=1, !=2, etc.); the default is from topline to end of screen. #cols is the width in columns (space=1, etc.); the default is from leftcol to column 80. If a window is specified past the edge of the screen, the defaults are used.

The last two parameters define the top right corner of an identically-sized region of page-2 memory. page2line is the top line; the default is the same line specified by topline. page2col is the leftmost column; the default is the same as leftcol. If the page 2 parameters define a window past the edge of the screen, the defaults are used.

The window command takes 30 milliseconds per 2000 characters, allowing 30 complete screen swaps per second. Speed is proportional to the size of the window being swapped, so fast animation is quite practical. When doing repetitve window commands, the object may "flicker" or appear transparent due to the fast swaps. To fix this, add 32 (decimal) to the topline parameter. This delays the NEXT window command until after the next vertical retrace.

Example 1: Pull-Up Menus

This program displays labels for the function keys on the 25th line. When a function key is hit, a "pull-up" menu appears above the chosen key with additional information.

First, set up a menu screen. The top line of each menu item will become the 25th line, and the text below it will be the pull-up menu. The menus are saved on page 2. To display a menu item, swap it to the screen with an "ESC m ..." command. To restore the screen, the same "ESC m ..." command swaps it back.

WINDOWS

Heath ESC x topline[leftcol[#lines[#cols[page2line[page2col]

If the SUPERCLOCK is installed, a 2nd page of screen memory is available. The "ESC x" command uses it to copy user-defined 'windows' between on-screen and

"ESC x" defines a window to be swapped between on-screen and off-screen memory. Windows can be anything from a single character to the entire screen. Both on- and off-screen windows are the same size, but need not be in the same position.

same method is used in the "ESC Y linecolumn" cursor position command. All parameters are optional; a RETURN or other out-of-range character will end the command early and execute it with defaults for the missing parameters.

enabled). This is in effect the same as a swap screen command, "ESC x @".

The first four parameters define a rectangular region of the main screen. Topline

topline to end of screen. #cols is the width in columns (space+1, etc.); the default is from leftcol to column 80. If a window is specified past the edge of the screen, the defaults are used.

specified by topline. page2col is the leftmost column; the default is the same as leftcol. If the page 2 parameters define a window past the edge of the screen, the defaults are used.

window being swapped, so fast animation is quite practical. When doing repetitive window commands, the object may "flicker" or appear transparent due to the fast swaps. To fix this, add 32 (decimal) to the topline parameter. This delays the

This program displays labels for the function keys on the 25th line. When a function key is hit, a "pull-up" menu appears above the chosen key with

line, and the text below it will be the pull-up menu. The menus are saved on page 2. To display a menu, swap it to the screen with an "ESC x ..."; the same "ESC x ..." command swaps it back.

```
10 REM Program to demonstrate pop-up "windows" with the SUPERCLOCK
20 WIDTH 255: E$=CHR$(27): PRINT E$;"p"
30 PRINT E$;"Y1(";"  f1-FONT      f2-STYLE      f3-MARGINS  ": REM our menu
40 PRINT E$;"Y2(";" Select FONT: Select STYLE: Set MARGINS: "
50 PRINT E$;"Y3(";"   1-H19        1-normal      1-left     "
60 PRINT E$;"Y4(";"   2-VT-100     2-reverse     2-right    "
70 PRINT E$;"Y5(";"   3-GT-PROM    3-interlace   3-top      "
80 PRINT E$;"Y6(";"   4-IBM PC                   4-bottom   "
90 REM
100 PRINT E$;"R6";E$;"E";:             REM save it on page 2, and clear screen
110 PRINT E$;"y1";E$;"x1";:            REM enable and clear 25th line
120 PRINT E$;"m8# H1(";:REM window to 25th line="8" column 4="#" one line=" "
130 REM                   40 cols wide="H" from page2 line 18="1" col 9="("
140 REM
150 PRINT E$;"q";"Hit f1, f2, f3, or control-C to end"
160 A$=INKEY$:IF A$CHR$(27) THEN 160:REM wait for an ESC
170 A$=INKEY$:IF LEN(A$)=0 THEN 170: REM then get next key
180 REM
190 ON ABS(ASC(A$)-82) GOTO 220,270,320: REM is it f1,f2, or f3?
200 GOTO 160
210 REM                                display f1 menu
220 PRINT E$;"m4#$-2(";:               REM show 5-line 14-char FONT menu
230 A$=INKEY$:IF LEN(A$)=0 THEN 230: REM wait for a key
240 PRINT E$;"m4#$-2(";:               REM save FONT menu
250 GOTO 160
260 REM                                display f2 menu
270 PRINT E$;"m50#.25";:               REM show 4-line 15-char STYLE menu
280 A$=INKEY$:IF LEN(A$)=0 THEN 280: REM wait for a key
290 PRINT E$;"m50#.25";:               REM save STYLE menu
300 GOTO 160
310 REM                                display f3 menu
320 PRINT E$;"m4$-2C";:               REM show 5-line 14-char MARGINS menu
330 A$=INKEY$:IF LEN(A$)=0 THEN 330: REM wait for a key
340 PRINT E$;"m4$-2C";:               REM save MARGINS menu
350 GOTO 160
```

Example 2: Moving Objects

"ESC m" commands are normally used in pairs; first to display part of page 2, and second to save it and restore the screen. This is a powerful technique; a "sprite" can be defined on page 2, and moved rapidly across a stationary background. Display it; pause; save it; then repeat with the sprite in a new location. This example will display a bouncing "smile" face.

```
 10 REM display a bouncing "smile" face (with Superclock)
 20 WIDTH 255
 30 E$=CHR$(27): M$=E$+"m" :PRINT E$;"j";E$;"H";E$;"F";E$;"p";E$;"R6";
 40 PRINT "r      _"
 50 PRINT "  ^  ^  ":                         REM define smile face
 60 PRINT " eaaaad "
 70 PRINT "_      ";E$;"qr";CHR$(13);"_";E$;"G";E$;"k";
 80 REM
 90 X=32: Y=32: STEPX=2: STEPY=1
100 P$=M$+CHR$(Y+32)+CHR$(X)+"#'  "+M$+CHR$(Y+STEPY)+CHR$(X+STEPX)+"#'  "
110 PRINT P$;:                                REM display it
120 X=X+STEPX: Y=Y+STEPY:                     REM move it
130 IF X103 OR X=32 THEN STEPX=-STEPX: PRINT CHR$(7);
```

```
10 REM Program to demonstrate pop-up "windows" with the SUPERCLOCK
20 WIDTH 255: E$=CHR$(27): PRINT E$;"p"
30 PRINT E$;"Y1C";"  1-FONT     2-STYLE     3-MARGINS   ";: REM out menu
[illegible]
70 PRINT E$;"Y3C";"  3-GT-PROM    3-interlace   3-top    "
80 PRINT E$;"Y4C";"  4-IBM PC              4-bottom   "
90 REM
[illegible]
130 REM              40 cols wide-"H" from page2 line 16-"1" col 9-"N"
140 REM
150 PRINT E$;"n";"Hit f1, f2, f3, or control-C to end"
[illegible]
190 ON ASC(A$)-80 GOTO 220,270,320: REM is it f1,f2, or f3?
200 GOTO 160
210 REM              display f1 menu
[illegible]
250 GOTO 160
260 REM              display f2 menu
270 PRINT E$;"=0h,2C";:         REM show 4-line 15-char STYLE menu
[illegible]
310 REM              display f3 menu
320 PRINT E$;"=Q-2C";:          REM show 5-line 14-char MARGINS menu
330 A$=INKEY$: IF LEN(A$)=0 THEN 330: REM wait for a key
[illegible]
```

Example 2: Moving Objects

"ESC =x" commands are normally used in pairs: first to display part of page 2, [illegible]

location. This example will display a bouncing "smile" face.

```
10 REM display a bouncing "smile" face (with Superclock)
[illegible]
50 PRINT "  ^ ^  ";             REM define smile face
60 PRINT " ommmo "
70 PRINT "     ";E$;"=r";CHR$(13);" ";E$;"S";E$;"X";
[illegible]
110 PRINT E$;                   REM display it
120 X=X+STEPX: Y=Y+STEPY        REM move it
130 IF X<35 OR X>72 THEN STEPX=-STEPX: PRINT CHR$(7);
```

```
140 IF Y=52 OR Y=32 THEN STEPY=-STEPY: PRINT CHR$(7);
150 IF LEN(INKEY$)=0 THEN GOTO 100        REM "bounce" off edges and loop
160 END                                   REM or type any key to end
```

Note the pair of window commands in line 100. The first one uses topline+32 so the face is displayed at least 16 mSec (one vertical frame) before the 2nd window command removes it. Without the "+32", the face would be displayed and removed at about a 50% duty cycle, making it flicker and appear transparent.

Since BASIC is so slow, the two commands are first assembled in the variable P$, so they can be output quickly by the PRINT statement.

DISPLAY 8TH BIT

Heath: display 8th bit ESC h
ignore 8th bit ESC i

ANSI: not available

The H19 terminal is normally set to send/receive 8-bit characters (by DIP switch SW401 #6). However, it actually ignored the 8th bit on characters received, and always set the 8th bit to 0 on characters transmitted.

The SUPERSET allows the 8th bit to be put to good use. If SW401 #6 is set for 8-bit characters, then ESC h causes characters with the 8th bit set to be displayed with the current attribute active; reverse video, blinking, or an extended character set. If a control character is received with its 8th bit set, it displays as the corresponding graphics character.

This greatly reduces the need for ESC sequences to display reverse video and graphics (ESC p, ESC q, ESC F, and ESC G). Just add 128 to any character you want displayed in reverse video, or add 32 to any lowercase letter to display the corresponding graphic character.

It also lets you to detect characters with the high bit set. For example, Wordstar sets the 8th bit high on many characters, which causes problems for other programs. Typing a Wordstar file (TYPE FILENAME at the CP/M prompt) will reveal these characters in reverse video.

Note that your operating system may strip the 8th bit on characters sent to the screen. If so, you will have to either use direct console I/O (CP/M's CONIO function) or output characters directly to the console UART.

Some programs send characters with the high bit set either by mistake or poor design. This can cause appearance problems on the screen. ESC i will cause the 8th bit to be ignored (the default), like a stock H19. Note that this also causes the transparent mode to ignore the 8th bit.

USER FUNCTION KEYS

Heath:	user function keys	ESC h =
	H19 function keys	ESC i =
	user shift-function keys	ESC h >
	Super19 shifted function keys	ESC i >
	select BASIC user keys	ESC f 1
	select CP/M user keys	ESC f 2
	select HDOS user keys	ESC f 3
	select user-defined keys	ESC f 4 (with SUPERCLOCK)
	program user keys	ESC f 5 (with SUPERCLOCK)
ANSI:	not available	

The top-row function keys (f1-f5, BLUE, RED, and WHITE) normally send stock H19 ESC codes. The shifted top-row function keys (shift-f1, shift-BLUE, etc.) send a different code, compatible with the Super-19. Many programs put these keys to good use, frequently labelling their functions with a message on the 25th line.

However, programs like CP/M, HDOS and BASIC make no use of these keys. The SUPERSET provides a way to put these unused keys to work, and thus save typing for frequently-used commands. These are called "user function keys".

ESC h = enables the top-row function keys to send user codes instead of stock H19 codes. ESC i = returns to stock H19 codes (the default). Shifted function keys are not affected.

ESC h > enables all shifted top-row function keys to send user codes instead of Super-19 codes. ESC i > returns to stock Super-19 codes (the default). Unshifted function keys are not affected.

ESC f number selects the user code set to be used; BASIC (the default), CP/M, or HDOS. The same set is used for both shifted and unshifted function keys.

	USER KEYS DISABLED		USER KEYS ENABLED		
function	unshifted	shifted	unshifted and/or shifted		
key	H19	Super-19	BASIC	CP/M	HDOS
f1	ESC S	ESC '	load "	dir	cat
f2	ESC T	ESC (	save "	ren	sy1:
f3	ESC U	ESC)	list	user	sy2:
f4	ESC V	ESC *	goto	list	mount
f5	ESC W	ESC +	gosub	stat	dismount
BLUE	ESC P	ESC $	return	era	reset
RED	ESC Q	ESC %	for	pip	copy
WHITE	ESC R	ESC &	next	type	type

If the SUPERCLOCK is installed, you can define your own user keys. ESC f 5 programs the user codes to match the labels (in reverse video) on the 25th line. ESC f 4 then selects your user codes. Once programmed, user keys are non-volatile and will be retained even with the power off.

It's simplest to program the user codes off-line. Push the OFF-LINE key so it locks down. Type ESC x 1 to enable the 25th line, and ESC Y 8 space to position the cursor there. Type ESC t to enable the keypad, and ESC p to enable

USER FUNCTION KEYS

Heath: user function keys ESC x 4

select BASIC user keys ESC { 1
select CP/M user keys ESC { 2

ANSI: not available

good use, frequently labelling their functions with a message on the 25th line.

However, programs like CP/M, HDOS and BASIC make no use of these keys. The

ESC x 4 enables the top-row function keys to send user codes instead of stock H19 codes. ESC y 4 returns to stock H19 codes (the default). Shifted function keys are not affected.

function keys are not affected.

ESC { number selects the user code set to be used, BASIC (the default), CP/M, or

Function key	unshifted H19	shifted Super-19	unshifted and/or shifted BASIC	CP/M	HDOS
f4	ESC V	ESC *	goto	list	mount
f5	ESC W	ESC +	gosub	stat	dismount
BLUE	ESC P	ESC @	return	era	reset

If the SUPERCLOCK is installed, you can define your own user keys. ESC { 0 programs the user codes to match the labels (in reverse video) on the 25th line. ESC { 4 then selects your user codes. Once programmed, user keys are

locks down. Type ESC x 1 to enable the 25th line, and ESC Y 8 space to position the cursor there. Type ESC ! to enable the keypad, and ESC p to enable reverse video. Now type the string for the f1 key; it will be displayed in reverse video. Move right with the right-arrow key (keypad 6) to leave at least one normal-video character between labels. Now type the string for f2. Continue for all 8 function keys. When done, ESC f 5 programs them into the user area.

Any ASCII character string can be used including control codes or ESC sequences. Graphic characters define control codes. For example, RETURN is a ctl-M, which would be entered as a graphic-m (ESC F m). Similarly ESC (ctl-[) is entered as a graphic-[(ESC F [). The maximum length for any one key is 16 characters, but the total length for all 8 function keys cannot exceed 40 characters.

There are many ways to put the user keys to work. Since most H89 software does not use shifted function keys, you can enable them as user keys with no effect on existing software. They are then ready at the BASIC or operating system level to reduce typing (shift-RED is short for "pip", etc.).

Programmable user keys are much more versatile. Any frequently-used sequence of keys can be programmed. Suppose you're writing a document that frequently uses the name "Przyblynski". Rather than type it every time, program a user key to enter it with a single keystroke.

They can also add functions to existing programs. For example, my word processor requires a 5-key sequence for a block move (ESC to the command menu; BM <return to do a block move; and <return to go back to the edit screen. So I programmed shift-f1 to send this sequence automatically.

LABEL 25TH LINE

Heath:	label user keys on clock line	ESC h ?
	no labels on clock line	ESC i ?
ANSI:	not available	

If the 25th line is not in use by an application program, it is available for displaying the time and date. In this case, the rest of the line is blank.

If the clock line is being displayed, and user function keys are enabled (for shifted or unshifted top-row keys), then ESC h ? (the default) will label the user key functions on the clock line. The labels appear in reverse video above each key (with a gap above ERASE, since it is not a user key).

ESC i ? will disable the user code display. This is useful when the user codes have little meaning when displayed in ASCII (arcane ESC sequences, for example), or if you just prefer a "clean" line.

The clock line (including labels) automatically goes away when a program enables the 25th line. When the program exits and disables the 25th line, the clock line and labels will return. This makes it easy to remember what the user keys do.

Suppose you set the shifted user keys for CP/M. In your application programs, the function keys work normally, along with any 25th line labels they provide. But when they disable the 25th line and return to CP/M, the user labels appear automatically. Just remember that if the clock is displayed, the labels are shifted-functions (shift-f1 is DIR, shift-RED is PIP, etc.).

reverse video. Now type the string for the f1 key; it will be displayed in reverse video. Move right with the right-arrow key (keypad 6) to leave at least one normal-video character between labels. Now type the string for f2. Continue

[illegible]

Graphic characters define control codes. For example, RETURN is a ctl-M, which would be entered as a graphic-m (ESC F m). Similarly ESC (ctl-[) is entered as a graphic-[(ESC F [). The maximum length for any one key is 16 characters, but

[illegible]

not use shifted function keys, you can enable them as user keys with no effect on existing software. They are then ready at the BASIC or operating system level to reduce typing (shift-RED is short for "pip", etc.).

[illegible]

the name "Przybylpski". Rather than type it every time, program a user key to enter it with a single keystroke.

[illegible]

shift-f1 to send this sequence automatically.

[illegible]

no labels on clock line ESC 1 ?

ASCII not avail**able**

[illegible]

If the clock line is being displayed, and user function keys are enabled (for shifted or unshifted top-row keys), then ESC X 7 (the default) will label the

[illegible]

ESC 1 ? will disable the user code display. This is useful when the user codes have **little** meaning when displayed in ASCII (arcane ESC sequences, for example), or if **you just** prefer a "clean" line.

[illegible]

and labels will return. This makes it easy to remember what the user keys do.

Suppose you set the shifted user keys for CP/M. In your application programs,

[illegible]

shifted-functions (shift-f1 is DIR, shift-RED is PIP, etc.).

ESCAPE CODE SUMMARY

The following lists all ESCape commands recognized by the SUPERSET. The left column is for Heath/Zenith (VT-52) mode; the right column is for ANSI mode. The symbol "*" identifies new or improved commands.

The ASCII ESCape character (27 decimal, 1B hex, 033 octal) must preceed the sequence of characters shown. An ESCape sequence can be aborted by the ASCII CANcel (24 decimal, 18 hex, 030 octal). <parameter> means one of several codes is possible. For ANSI codes, "n" is an optional repetition parameter; it tells how many times the command is to be repeated.

Heath/Zenith		Function		ANSI
<space><set>	*	Select character set (Super-19)		
<space>D	*	Heath H19 with Z100 extensions		
<space>H	*	DEC VT-100 with special graphics		
<space>L	*	NORCOM GT-PROM		
<space>P	*	special math and Greek		
<space>C	*	Heath H29 (with SUPERFONT)		
<space>G	*	VT-100 graphics (with SUPERFONT)		
<space>K	*	APA graphics (with SUPERFONT)		
<space>O	*	IBM PC (with SUPERFONT)		
#		Transmit page		[p or [Op
$	*	shifted BLUE key (Super-19)	*	[$
%	*	shifted RED key (Super-19)	*	[%
&	*	shifted WHITE key (Super-19)	*	[&
'	*	shifted f1 key (Super-19)	*	['
(	*	shifted f2 key (Super-19)	*	[(
)	*	shifted f3 key (Super-19)	*	[)
*	*	shifted f4 key (Super-19)	*	[*
+	*	shifted f5 key (Super-19)	*	[+
/K		Response to ESC Z (identify as VT52)		
:hh:mm:ss	*	Set clock (Super-19)		
:hh:mm:ss	*	response to Super-19 Read clock (ESC ;)		
:mm:dd:yy	*	response to Super-19 Read date (ESC ~)		
;	*	Read clock (Super-19)		
<		Enter ANSI mode		
		Enter Heath/Zenith mode		[?2h
=		Enter alternate keypad mode		=
>		Exit alternate keypad mode		>
?	*	Transmit edited line		
?<parameter>		Output from keypad in alternate mode		O<parameter>
?M		ENTER key		OM
?n		. key		On
?p		0 key		Op
?q		1 key		Oq
?r		2 key		Or
?s		3 key		Os
?t		4 key		Ot
?u		5 key		Ou
?v		6 key		Ov
?w		7 key		Ow
?x		8 key		Ox
?y		9 key		Oy

Heath/Zenith	Function	ANSI
@	Enter insert character mode	[4h
A	Cursor up (n times)	[nA
B	Cursor down (n times)	[nB
C	Cursor right (n times)	[nC
D	Cursor left (n times)	[nD
E	Clear screen (except 25th line)	[2J
E	SHIFT-ERASE key	[J
F	Enter graphics mode	[10m
G	Exit graphics mode	[11m
H	Move cursor to home position	[H or [f
I	Reverse index (reverse linefeed)	M
J	Erase from cursor to end of screen	[0J
J	ERASE key (unshifted)	[J
K	Erase from cursor to end of line	[K or [0K
L	Insert line at cursor pos. (n times)	[nL
M	Delete line at cursor pos. (n times)	[nM
N	Delete character at cursor (n times)	[nP
O	Exit insert character mode	[4l
R<parameter>	* Save/restore/swap modes (and screen with SUPERCLOCK)	
R1	* Restore modes (and screen)	
R2	* Save modes (and screen)	
R3	* Restore modes	
R4	* Save modes	
R5	* (restore screen)	
R6	* (save screen)	
R7	* swap modes (and screen)	
R8	* (swap screen)	
R9	* swap modes	
P	BLUE key	[0P
Q	RED key	[0Q
R	WHITE key	[0R
S	f1 key (DEC f1 key)	[0S
T	f2 key (DEC f2 key)	[0T
U	f3 key (DEC f3 key)	[0U
V	f4 key (DEC f4 key)	[0V
W	f5 key	[0W
Xhhmmss<RET>	* Set clock. hhmmss=hours,minutes,seconds	
Y<line><col>	Position cursor Line = <space> thru <8> Col=<space> thru <o>	[<line>;<col>f or [<line>;<col>H
Z	Identify as VT52; response is ESC / K	
[	Enter hold screen mode	[3h
\	Exit hold screen mode	
]	Transmit 25th line	* [q
^	* Transmit current (cursor) line	[1p
_	* Transmit character at cursor	[2p
a	* Extended 256-character set	
b	Erase from beginning to cursor	[1J
c	* Enable clock display	
d	* Disable clock display	
e	* Send time to host	

Heath/Zenith		Function		ANSI
h<parameter>	*	Super-19 set modes		
h1	*	full duplex		
h2	*	hardware handshaking		
h3	*	limit transmit speed		
h5	*	white screen		
h7	*	wrap long lines		
h9	*	interlace on		
h:	*	screen saver on		
h;	*	enable soft reset (ESC z)		
i<parameter>	*	Super-19 reset modes		
i1	*	half duplex		
i2	*	software handshaking		
i3	*	full transmit speed		
i5	*	black screen		
i7	*	discard long lines		
i9	*	interlace off		
i:	*	screen saver off		
i;	*	enable hard reset (ESC z)		
j		Save current cursor position		[s
k		Restore current cursor position		[u
l		Erase entire line		[2K
m<p1><p2><p3><p4><p5><p6>		Swap Window (SUPERCLOCK)		
	*	0-6 parameters, like ESC Y<Pn1><Pn2>		
		p1 - top line on screen		
		p2 - left column on screen		
		p3 - height in lines		
		p4 - width in columns		
		p5 - top line on Page 2		
		p6 - left column on Page 2		
n		Cursor position report		[6n
		Heath response is ESC Y<line><col>		
		ANSI response is ESC[<line>;<col>R		
o		Erase from beginning of line to cursor		[1K
p		Enter reverse video mode		[7m
q		Exit reverse video mode		[0m
r<parameter>		Set baud rate		
r@	*	75 baud	*	[0r
rA		110 baud		[1r
rB		150 baud		[2r
rC		300 baud		[3r
rD		600 baud		[4r
rE		1200 baud		[5r
rF		1800 baud		[6r
rG		2000 baud		[7r
rH		2400 baud		[8r
rI		3600 baud		[9r
rJ		4800 baud		[10r
rK		7200 baud		[11r
rL		9600 baud		[12r
rM		19200 baud		[13r
rN	*	38400 baud		[14r
rO	*	134.5 baud		[15r

Heath/Zenith		Function	ANSI
s	*	Normal 128-character set	
t		Enter keypad shifted mode	
u		Exit keypad shifted mode	
v		Wrap characters past 80 to next line	[?7h
w		Discard characters past 80th column	[?7l
x<parameter>		Set modes	
x1		Enable 25th line	[>1h
x2		Disable keyboad click	[>2h
x3		Enter hold screen mode	[>3h
x4		Set block cursor	[>4h
x5		Disable cursor	[>5h
x6		Enter keypad shifted mode	[>6h
x7		Enter keypad alternate mode	[>7h
x8		Enable auto LF on CR	[>8h
x9		Enable auto CR on LF	[>9h or [20h
x;	*	Set non-blinking cursor	[>11h
x=	*	Set hardware handshaking	[>13h
x>	*	Enable half duplex	[>14h
x?	*	Enable native keyboard mode	[>15h
x@	*	Enable transparent mode	[>16h
y<parameter>		Reset modes	
y1		Disable 25th line	[>1l
y2		Enable keyboard click	[>2l
y3		Exit hold screen mode	[>3l
y4		Set underline cursor	[>4l
y5		Enable cursor	[>5l
y6		Exit keypad shifted mode	[>6l
y7		Exit keypad alternate mode	[>7l
y8		Disable auto LF on CR	[>8l
y9		Disable auto CR on LF	[>9l or [20l
y;	*	Set blinking cursor	[>11l
y=	*	Set software handshaking	[>13l
y>	*	Disable half duplex	[>14l
y?	*	Disable native keyboard mode	[>15l
z		Reset to power up configuration	[z
{		Enable keyboard	[2l
|	*	Set Date (Super-19)	
}		Disable keyboard	[2h
~	*	Read Date (Super-19)	

ESCAPE CODE SUMMARY

The following lists all ESCape commands recognized by the SUPERSET. The left column is for Heath/Zenith (VT-52) mode; the right column is for ANSI mode. The symbol "*" identifies new or improved commands.

The ASCII ESCape character (27 decimal, 1B hex, 033 octal) must preceed the sequence of characters shown. An ESCape sequence can be aborted by the ASCII CANcel (24 decimal, 18 hex, 030 octal). <parameter> means one of several codes is possible. For ANSI codes, "n" is an optional repetition parameter; it tells how many times the command is to be repeated.

Heath/Zenith		Function		ANSI
<space><set>	*	Select character set (Super-19)		
<space>D	*	Heath H19 with Z100 extensions		
<space>H	*	DEC VT-100 with special graphics		
<space>L	*	NORCOM GT-PROM		
<space>P	*	special math and Greek		
<space>C	*	Heath H29 (with SUPERFONT)		
<space>G	*	VT-100 graphics (with SUPERFONT)		
<space>K	*	APA graphics (with SUPERFONT)		
<space>O	*	IBM PC (with SUPERFONT)		
#		Transmit page		[p or [Op
$	*	shifted BLUE key (Super-19)	*	[$
%	*	shifted RED key (Super-19)	*	[%
&	*	shifted WHITE key (Super-19)	*	[&
'	*	shifted f1 key (Super-19)	*	['
(	*	shifted f2 key (Super-19)	*	[(
)	*	shifted f3 key (Super-19)	*	[)
*	*	shifted f4 key (Super-19)	*	[*
+	*	shifted f5 key (Super-19)	*	[+
/K		Response to ESC Z (identify as VT52)		
:hh:mm:ss	*	Set clock (Super-19)		
:hh:mm:ss	*	response to Super-19 Read clock (ESC ;)		
:mm:dd:yy	*	response to Super-19 Read date (ESC ~)		
;	*	Read clock (Super-19)		
<		Enter ANSI mode		
		Enter Heath/Zenith mode		[?2h
=		Enter alternate keypad mode		=
>		Exit alternate keypad mode		>
?	*	Transmit edited line		
?<parameter>		Output from keypad in alternate mode		O<parameter>
?M		ENTER key		OM
?n		. key		On
?p		0 key		Op
?q		1 key		Oq
?r		2 key		Or
?s		3 key		Os
?t		4 key		Ot
?u		5 key		Ou
?v		6 key		Ov
?w		7 key		Ow
?x		8 key		Ox
?y		9 key		Oy

Heath/Zenith		Function	ANSI
@		Enter insert character mode	[4h
A		Cursor up (n times)	[nA
B		Cursor down (n times)	[nB
C		Cursor right (n times)	[nC
D		Cursor left (n times)	[nD
E		Clear screen (except 25th line)	[2J
E		SHIFT-ERASE key	[J
F		Enter graphics mode	[10m
G		Exit graphics mode	[11m
H		Move cursor to home position	[H or [f
I		Reverse index (reverse linefeed)	M
J		Erase from cursor to end of screen	[0J
J		ERASE key (unshifted)	[J
K		Erase from cursor to end of line	[K or [0K
L		Insert line at cursor pos. (n times)	[nL
M		Delete line at cursor pos. (n times)	[nM
N		Delete character at cursor (n times)	[nP
O		Exit insert character mode	[4l
R<parameter>	*	Save/restore/swap modes (and screen with SUPERCLOCK)	
R1	*	Restore modes (and screen)	
R2	*	Save modes (and screen)	
R3	*	Restore modes	
R4	*	Save modes	
R5	*	(restore screen)	
R6	*	(save screen)	
R7	*	swap modes (and screen)	
R8	*	(swap screen)	
R9	*	swap modes	
P		BLUE key	[0P
Q		RED key	[0Q
R		WHITE key	[0R
S		f1 key (DEC f1 key)	[0S
T		f2 key (DEC f2 key)	[0T
U		f3 key (DEC f3 key)	[0U
V		f4 key (DEC f4 key)	[0V
W		f5 key	[0W
Xhhmmss<RET>	*	Set clock. hhmmss=hours,minutes,seconds	
Y<line><col>		Position cursor Line = <space> thru <8> Col=<space> thru <o>	[<line>;<col>f or [<line>;<col>H
Z		Identify as VT52; response is ESC / K	
[		Enter hold screen mode	[3h
\		Exit hold screen mode	
]		Transmit 25th line	* [q
^	*	Transmit current (cursor) line	[1p
_	*	Transmit character at cursor	[2p
a	*	Extended 256-character set	
b		Erase from beginning to cursor	[1J
c	*	Enable clock display	
d	*	Disable clock display	
e	*	Send time to host	

Heath/Zenith	Function	ANSI
f<parameter>	* set top-row function key codes	
f1	* BASIC user function keys	
f2	* CP/M " " "	
f3	* HDOS " " "	
f4	* user-defined " "	
f5	* program user keys to match 25th line	
h<parameter>	* Super-19 set modes	
h1	* full duplex	
h2	* hardware handshaking	
h3	* limit transmit speed	
h5	* white screen	
h7	* wrap long lines	
h9	* interlace on	
h:	* screen saver on	
h;	* enable soft reset (ESC z)	
h<	* display 8th bit on receive	
h=	* user codes for unshifted function keys	
h>	* user codes for shifted function keys	
h?	* label user function keys on clock line	
i<parameter>	* Super-19 reset modes	
i1	* half duplex	
i2	* software handshaking	
i3	* full transmit speed	
i5	* black screen	
i7	* discard long lines	
i9	* interlace off	
i:	* screen saver off	
i;	* enable hard reset (ESC z)	
i<	* ignore 8th bit on receive	
i=	* H19 codes for unshifted function keys	
i>	* Super-19 codes for shifted function keys	
i?	* no user function key labels on clock line	
j	Save current cursor position	[s
k	Restore current cursor position	[u
l	Erase entire line	[2K
m<p1><p2><p3><p4><p5><p6>	Swap Window (SUPERCLOCK)	
	* 0-6 parameters, like ESC Y<Pn1><Pn2>	
	p1 - top line on screen	
	p1+32 to delay for next Vsync	
	p2 - left column on screen	
	p3 - height in lines	
	p4 - width in columns	
	p5 - top line on Page 2	
	p6 - left column on Page 2	
n	Cursor position report	[6n
	Heath response is ESC Y<line><col>	
	ANSI response is ESC[<line>;<col>R	
o	Erase from beginning of line to cursor	[1K
p	Enter reverse video mode	[7m
q	Exit reverse video mode	[0m

Heath/Zenith		Function	ANSI
r<parameter>		Set baud rate	
r@	*	75 baud	* [0r
rA		110 baud	[1r
rB		150 baud	[2r
rC		300 baud	[3r
rD		600 baud	[4r
rE		1200 baud	[5r
rF		1800 baud	[6r
rG		2000 baud	[7r
rH		2400 baud	[8r
rI		3600 baud	[9r
rJ		4800 baud	[10r
rK		7200 baud	[11r
rL		9600 baud	[12r
rM		19200 baud	[13r
rN	*	38400 baud	[14r
rO	*	134.5 baud	[15r
s	*	Normal 128-character set	
t		Enter keypad shifted mode	
u		Exit keypad shifted mode	
v		Wrap characters past 80 to next line	[?7h
w		Discard characters past 80th column	[?7l
x<parameter>		Set modes	
x1		Enable 25th line	[>1h
x2		Disable keyboad click	[>2h
x3		Enter hold screen mode	[>3h
x4		Set block cursor	[>4h
x5		Disable cursor	[>5h
x6		Enter keypad shifted mode	[>6h
x7		Enter keypad alternate mode	[>7h
x8		Enable auto LF on CR	[>8h
x9		Enable auto CR on LF	[>9h or [20h
x;	*	Set non-blinking cursor	[>11h
x=	*	Set hardware handshaking	[>13h
x>	*	Enable half duplex	[>14h
x?	*	Enable native keyboard mode	[>15h
x@	*	Enable transparent mode	[>16h
y<parameter>		Reset modes	
y1		Disable 25th line	[>1l
y2		Enable keyboard click	[>2l
y3		Exit hold screen mode	[>3l
y4		Set underline cursor	[>4l
y5		Enable cursor	[>5l
y6		Exit keypad shifted mode	[>6l
y7		Exit keypad alternate mode	[>7l
y8		Disable auto LF on CR	[>8l
y9		Disable auto CR on LF	[>9l or [20l
y;	*	Set blinking cursor	[>11l
y=	*	Set software handshaking	[>13l
y>	*	Disable half duplex	[>14l
y?	*	Disable native keyboard mode	[>15l
z		Reset to power up configuration	[z
{		Enable keyboard	[2l
|	*	Set Date (Super-19)	
}		Disable keyboard	[2h
~	*	Read Date (Super-19)	

QUESTIONS AND ANSWERS

Here are some of the common questions asked about the SUPERSET. If you have a question (or an answer) that is not listed, let us know.

Q: How does the SUPERSET reduce power consumption?

A: The standard H/Z-19 TLB draws about 1150mA from the +8 volt supply. Heath used NMOS memories, which are cheap but take a lot of power. Nowdays, there are CMOS equivalents which take virtually no power. The CMOS parts supplied save about 300mA, which is about as much power as removing an I/O board.

Q: What is "baud rate"?

A: The baud rate is the number of bits per second sent or received by the terminal. The standard baud rate is 9600. There are 10 bits per character, so this is 960 characters per second. There are 2000 characters on the screen, so it takes about 2 seconds to change them all. 9600 baud is as fast as a standard H/Z-19 can go without missing characters.

Q: Can I increase the baud rate to speed up my terminal?

A: Yes. 19200 baud is twice as fast, and 38400 baud is four times faster. You can select 19200 temporarily by typing "ctl-ESC r M"; or permanently by setting DIP switch SW401 section #0. For 38400, type "ctl-ESC r N" or set SW401 section #1. See the section on DIP SWITCH SETTINGS for details.

Q: How do I select 19200 baud on an H/Z89 computer?

A: This is more complicated. You'll have to change the terminal, the computer, and the software all at once. For CP/M, run Heath's CONFIGUR program, change the Console baud rate from 9600 to 19200, and save it on disk. Upon exiting CONFIGUR, the baud rate will change and you will get garbage on the screen. Now change the terminal baud rate temporarily using the instructions above. Your CP/M prompt will come back and work again.

For HDOS, insert a non-bootable disk (one just INITialized for instance), and try to boot it at the "H:" prompt. Of course, it won't boot. When the drive light goes out, temporarily set the baud rate to 19200 with " ctl-ESC r M ". Now replace the disk with a bootable disk, go back ON-LINE and hit the "space" bar until the system responds. This disk is now set for 19200 baud.

Run a few programs, and you will notice that the screen responds faster. When you are satisfied with the results, you can permanently change the baud rate on the TLB as described above. Then change the computer permanently by setting DIP switch SW501 on the CPU board (set section #6 to 1=ON=left).

Q: Once I set the DIP switches for 19200 baud, I get "fxff xfgxsf" when I boot my disks. What happened?

A: Once you have permanently set SW401 and SW501 for 19200 baud, you can only boot disks CONFIGURed for a 19200 baud Console. If you forget, the disk will boot OK but print garbage instead of a sign-on message. You can temporarily fix this by typing "ctl-ESC r L" to go back to 9600 baud. Then run CONFIGUR to set the disk to boot at 19200 baud.

Q: At 19200 baud my screen is faster, but not twice as fast. Why?

A: The screen update rate is set by the software, the computer clock speed, or the baud rate; whichever is slowest. At 9600 baud, the baud rate is the weak link. At 19200, the 2MHz Z80 on your computer board is the weak link. If you also speed up your CPU board to 4MHz, then you'll see the 2:1 improvement.

Q: How do I get to 38400 baud?

A: This is awkward, because the CPU board's MTR-90 doesn't have a DIP switch setting for 38400 baud (though I have a "patched" version that does for $18). CP/M can CONFIGUR for 38400 baud, but you must boot at a lower rate and change manually to 38400 baud after boot-up. However, it should be possible to auto-execute a file that switches the terminal to 38400 baud.

Now for HDOS. Make a new system disk and boot it. Before you type spaces to set the baud rate, switch the SUPERSET to the new baud rate (i.e. ctl-ESC r N). Now type spaces and continue the boot process. HDOS should automatically set the baud rate to 38400.

HDOS 2.0 system disks previously set at 9600 must be patched to 38400. Use a disk utility program like DUMP (HUG #885-1062) for hard sector disks or UDUMP (HUG #885-8004) for others. Examine address 05 on track 0, sector 0. The value 0C specifies 9600 baud. Patch it to 06 for 19200, or 03 for 38400 baud.

Q: What causes the "flicker"?

A: The screen memory is shared between the Z80 microprocessor and 6845 CRT controller. Normally, the CRT takes precedence. But when the Z80 needs to read or write a character to the screen, it "steals" the memory away from the CRT. The result is a black dash on the screen. The effect is most severe with large amounts of reverse video or at high baud rates. Speeding up the Z80's clock shortens the dashes, but they are still there.

Q: How did you get rid of it?

A: The SUPERSET solves the problem by using very fast screen memory chips. The memory is switched back and forth between Z80 and CRT controller at a 3MHz rate (by connecting the Z80 clock to the 74S138 multiplexer). When the clock is high, the screen memory is used by the Z80, which latches its data at the falling edge. When the clock is low, the screen memory is used by the CRT controller, which latches data on the rising edge.

Q: How about custom character sets?

A: The SUPERFONT doubles the number of character sets to 8. In fact, it really contains 16 128-character sets, or 2048 total characters! It is quite easy to design your own fonts for foreign languages, special graphics, underlining, boldface, double-width characters, etc. If you would like a disk with the present character sets and instructions for editing, send us $10 and specify the desired disk format. We can then program a SUPERFONT with your own fonts.

Q: Can I use my Ultra-ROM or Super-19 ROM?

A: No. The SUPERSET program ROM replaces them. But since it already has most features of the Super-19 and Ultra-ROM, you shouldn't miss them. If there is a feature of these ROMs that you'd like added to the SUPERSET, let us know.

Q: Can I use the Cleveland Codonics "Imaginator"?

A: No. The Imaginator uses subroutines in the standard Heath ROM that have moved, or are no longer there. We'd like to fix this, but don't have access to the Imaginator's source code.

Q: Can I use the Northwest Digital "Graphics-Plus?"

A: No. The Graphics-Plus installation removes all the chips used by the SUPERSET kit, and all video is generated by the graphics board itself.

Q: How about a Sigmasoft IGC (Integrated Graphics Controller)?

A: Yes, with a few modifications. Sigmasoft moves the character generator ROM to their IGC board, so that's where the Superset's character ROM must go. Locate the 2K character ROM socket on the IGC board, and cut all traces to pin 21. Connect one end of a jumper wire to pin 21 of this socket.

Locate the 34-pin ribbon cable connector nearby, in the back left corner. Connect the other end of this wire to the center pin in the row closest to the edge of the board.

Install the Superset's character ROM and 24-to-28 pin socket adapter in the 2K socket on the IGC. You may need to lengthen the wires from the socket adapter to reach the 8250 UART on the terminal board. That's it!

All functions of the Superset and IGC work together normally, except that the Superset replaces the IGC's 2-character set capabilities.

Q: Why didn't you make it VT-100 compatible instead of just VT-52?

A: Frankly, because I don't have sufficient documentation to know how a VT-100 works, and don't have any software to test it even if I did. If you can help in this area, please contact me.

Q: Why do the new ROMs have more pins than the original parts?

A: The original ROM had 4K bytes of memory; either one 4K part, or two 2K parts. The SUPERSET and SUPERFONT use 32K parts instead. They have extra address pins that must be connected. The socket adapters make this as painless as possible. With the SUPERSET, SUPERFONT, and SUPERCLOCK all installed, your H/Z-19 board has over 70K of memory; more than the H89 itself!

Q: How was this manual printed?

A: This manual was prepared on an H89, using the "Magic Wand" word processor. It was printed on an Okidata u92 printer using its "correspondence mode" for text. The pictures were created with "Magic Wand" by manually turning on the H19 graphics mode while editing, and printed by initializing the printer's downloadable font with H19 graphics.

Q: What about speed?

A: The SUPERSET is over twice as fast as a standard H/Z-19, even considering the 50% faster clock. Critical routines in particular (insert/delete character, erase line/page) are dramatically faster. Programmers, I CHALLENGE you to figure out how I clear the screen twice as fast as even an LDIR instruction!

The speed of the SUPERSET is in contrast to the slowness of your author in getting this manual out. It's been an interminable project, taking longer to write than the program itself. I only hope you find the end result worth it!

Q: What else can I do to improve my H/Z-19?

A: The venerable H/Z19 stacks up well against modern terminals thanks to its advanced Z80. Just try to find terminals that can run at 9600 baud without missing characters. Other things in the works: more fonts (up to 32!); 7x9 dot fonts; 640x250 dot resolution graphics WITHOUT a graphics board; programmable sound generator; still faster Z80 clock speeds; new attributes (bright, dim, and italics); downloadable character sets in non-volatile RAM. We're just beginning!

The physical modifications are easy; the difficulty lies in overcoming the air of apathetic hopelessness that traps so many 8-bit computer users. One of the foundations of computer science is that ALL computers are equivalent; there is no program that runs on a MacIntosh or IBM PC that can't be written to run just as well on an 8-bit machine. H89 software lacks the flash and glamour of a PC only because it uses less memory.

Today's technology has eliminated the memory shortage. You can have megabytes of memory in an H89 as easily as the newer computers. It only remains for YOU to decide what to do with it!

USING THE GT-PROM

The GT-PROM is a new character set which improves the graphics capability and appearance of text on Heath H19/H19A terminals and H89/H89A computers. It displays 128 symbols, including 95 ASCII characters and 33 graphics symbols.

All 95 of the ASCII characters have been enhanced to produce a pleasing and stylish display of text on the CRT screen. Twenty-four of the original graphic symbols have been modified, to sharpen vertical resolution and to improve the match between vertical and horizontal line widths. To make room for the new symbols, five "non-graphic" symbols from the old set were deleted (paragraph, plus/minus, division, right-arrow, and down-arrow). Also, five diagonal graphic symbols were replaced with symbols using a uniform square pixel shape. Graphic symbols that were unchanged or only slightly modified were left in their original locations, to minimize the effect on existing graphic programs.

The documentation includes a listing of a screen plot program in BASIC to demonstrate the value of the improved graphics resolution.

OPERATION

The correct appearance of all characters, and the relationships of displayed characters to key symbols are listed in the "character set" section of this manual. Remember, reverse video provides the inverse of each character. The format is as follows.

```
                  +--------+
                  !        !
                  !   █    !
                  !  █ █   !
                  ! █   █  !  character display map
                  ! █   █  !
                  ! █████  !
                  ! █   █  !
                  ! █   █  !
                  !        !
                  +--------+
ASCII key symbol  ! A  65  !  character number in decimal
                  !    41h !  character number in hexadecimal
                  +--------+
```

The GT-PROM can produce a square "dot" in any of ten positions within a character cell (2 wide by 5 high). With an 80x25 character screen, this gives an effective graphics resolution of 160 horizontal by 125 vertical. The codes to produce each of the ten individual dots are shown in their respective boxes:

```
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
 |  j  |  k  |      | 106 | 107 |      | 6Ah | 6Bh |      | 152 | 153 |
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
 |  }  |  ~  |      | 125 | 126 |      | 7Dh | 7Eh |      | 175 | 176 |
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
 |  |  |  ^  |      | 124 |  94 |      | 7Ch | 5Eh |      | 174 | 136 |
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
 |  r  |  _  |      | 114 |  95 |      | 72h | 5Fh |      | 162 | 137 |
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
 |  h  |  g  |      | 104 | 103 |      | 68h | 67h |      | 150 | 147 |
 +-----+-----+      +-----+-----+      +-----+-----+      +-----+-----+
     ASCII             Decimal          Hexadecimal           Octal
```

DEMONSTRATION PROGRAM

On the next page is a listing of a demonstration program for the GT-PROM to give you an idea of what the extra resolution can do. It is written in Microsoft BASIC, and runs under CP/M 2.2. You should be able to get it to run without much difficulty using any similar BASIC interpreter.

The program displays a damped sinewave within a rectangular border. Notice the relative smoothness of the sinewave contour, and the uniform thickness of the horizontal and vertical borders.

You can easily modify the program to display other functions by changing lines 220-230 and 280-310. For example,

```
220 A=(I/K)*3.14159                            280 XN=-1.5
222 R=SIN(3*A)                                 290 XX=1.5
224 X(I)=R*COS(A)                              300 YN=-1
230 Y(I)=R*SIN(A)                              310 YX=1
```

displays a function called a "trifolium" or a "3-leaved rose".
And by entering,

```
                                               280 XN=0
220 X(I)=(I+1)/(K+1)                           290 XX=1
230 Y(I)=(X(I)^-5)/(EXP(1/X(I))-1)             300 YN=0
                                               310 XY=22
```

you can display Plank's radiation function (Jahnke and Emde, Tables of Functions, Addendum VII).

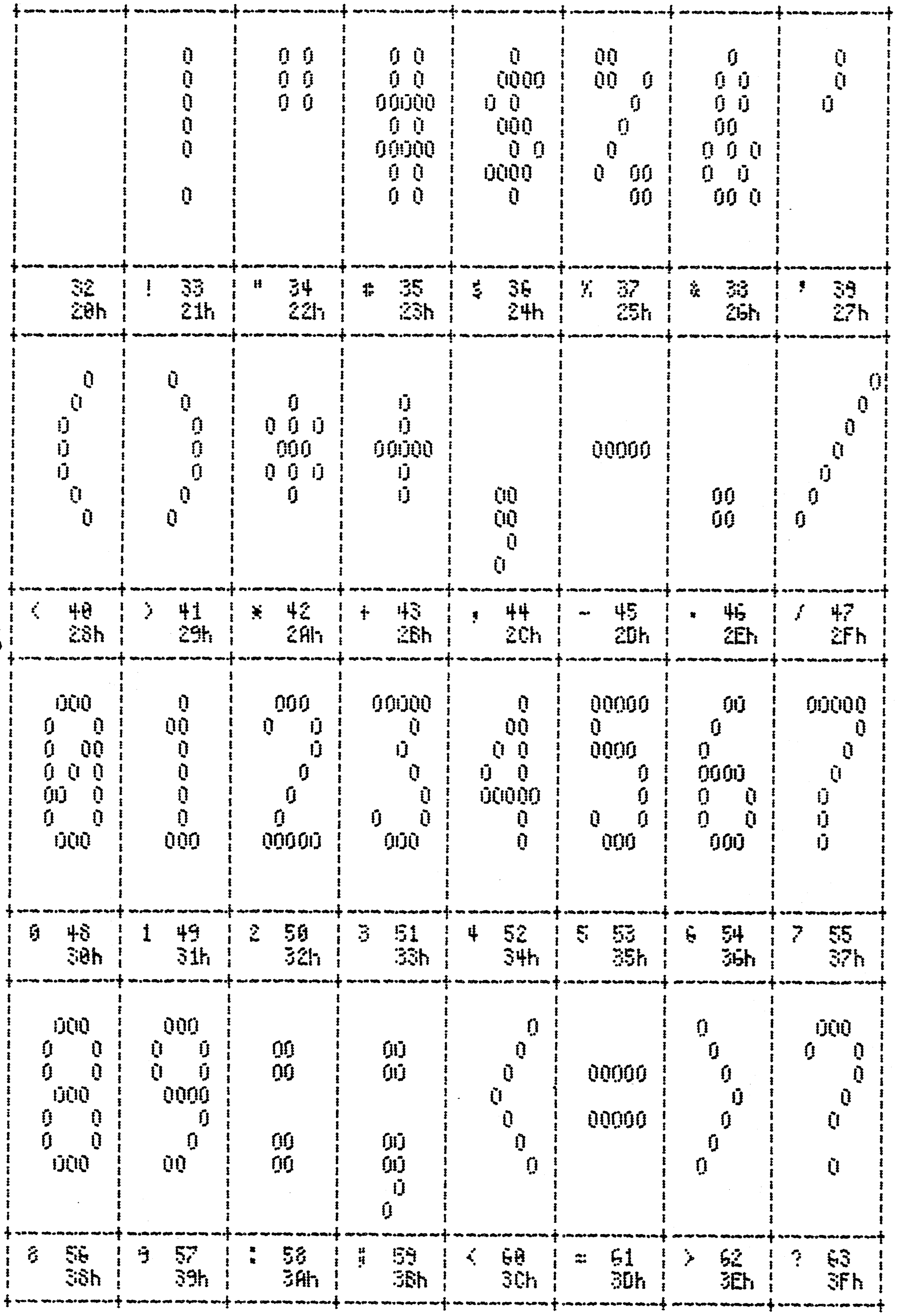
The GT-PROM can also display the old low-resolution block graphics symbols. This makes it possible to compare the new and old resolutions by changing lines 1110 and 1120 as follows:

```
1110 P$(0)=CHR$(109): P$(1)=CHR$109): P$(2)=CHR$(110)
1120 P$(3)=CHR$(110): P$(4)=CHR$(110)
```

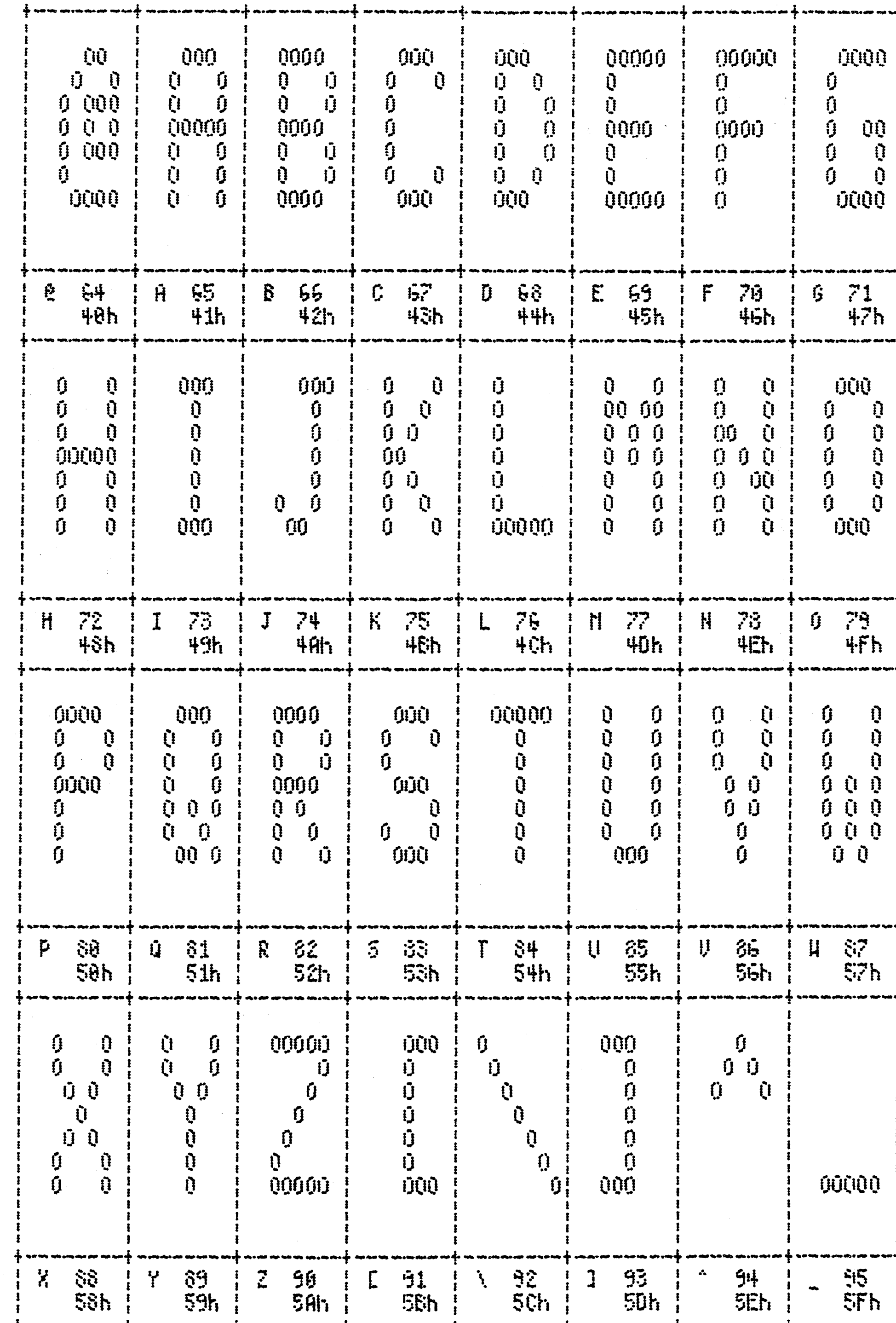
Now the program will use the original LOW resolution graphic symbols. Change lines 1110 and 1120 back to their original form to go back to the HIGHER resolution of the GT-PROM.

```
100 REM         Demonstration program for H18 or H89 with GT-PROM
110 REM         character generator.  Runs on MBASIC under CP/M 2.2.
120 REM
130 REM
140 REM         Roger Lembke
150 REM         NORCOM, 9630 Hayes, Overland Park, KS 66212
160 REM
170 DIM X(100),Y(100)
180 REM         Set up X and Y arrays to contain the function to
190 REM         be displayed; in this case, a damped sinewave.
200 K=68
210 FOR I=0 TO K
220 X(I)=I*.3
230 Y(I)=EXP(-.13*X(I))*SIN(X(I))
240 NEXT I
250 REM         Specify min and max limits of X and Y to define
260 REM         the range of the display.
270 REM
280 XN=0:       REM X min
290 XX=20.4:    REM X max
300 YN=-.6:     REM Y min
310 YX=.9:      REM Y max
1000 REM
1010 REM        The rest of this program can be used as a subroutine
1020 REM        to plot any function stored in the X and Y arrays,
1030 REM        as shown in lines 210-240. The X and Y limits for
1040 REM        the plot must also be provided, as in lines 280-310.
1050 REM
1060 DIM N$(70,20),P$(4)
1070 G$=CHR$(27)+CHR$(70): A$=CHR$(27)+CHR$(71)
1080 N1=69: N2=95: XR=XX-XN: YR=YX-YN
1090 L1$=CHR$(96): L2$=CHR$(97)
1100 C1$=CHR$(101): C2$=CHR$(100): C3$=CHR$(102): C4$=CHR$(99)
1110 P$(0)=CHR$(104): P$(1)=CHR$(114): P$(2)=CHR$(124)
1120 P$(3)=CHR$(125): P$(4)=CHR$(106)
1130 FOR I=0 TO 70: FOR J=0 TO 20: N$(I,J)=" ": NEXT J,I
1140 FOR J=0 TO 20: N$(0,J)=L1$: N$(70,J)=L1$: NEXT J
1150 FOR I=0 TO 70: N$(I,0)=L2$: N$(I,20)=L2$: NEXT I
1160 N$(0,0)=C1$: N$(70,0)=C2$: N$(0,20)=C3$: N$(70,20)=C4$
1170 FOR I=0 TO K
1180 I1=INT(((X(I)-XN)/XR)*(N1-1)+1.5)
1190 I2=INT(((Y(I)-YN)/YR)*(N2-1)+1.5)
1200 I3=I2-5*INT(I2/5)-1: IF I3=-1 THEN I3=4 ELSE GOTO 1210
1210 I4=1+INT((I2-I3)/5): IF I4>1 THEN GOTO 1220 ELSE I4=1: I3=0
1220 N$(I1,I4)=P$(I3)
1230 NEXT I
1240 PRINT G$
1250 FOR J=20 TO 0 STEP -1
1260 FOR I=0 TO 70: PRINT N$(I,J);: NEXT I
1270 PRINT: NEXT J
1280 PRINT A$;: END
```

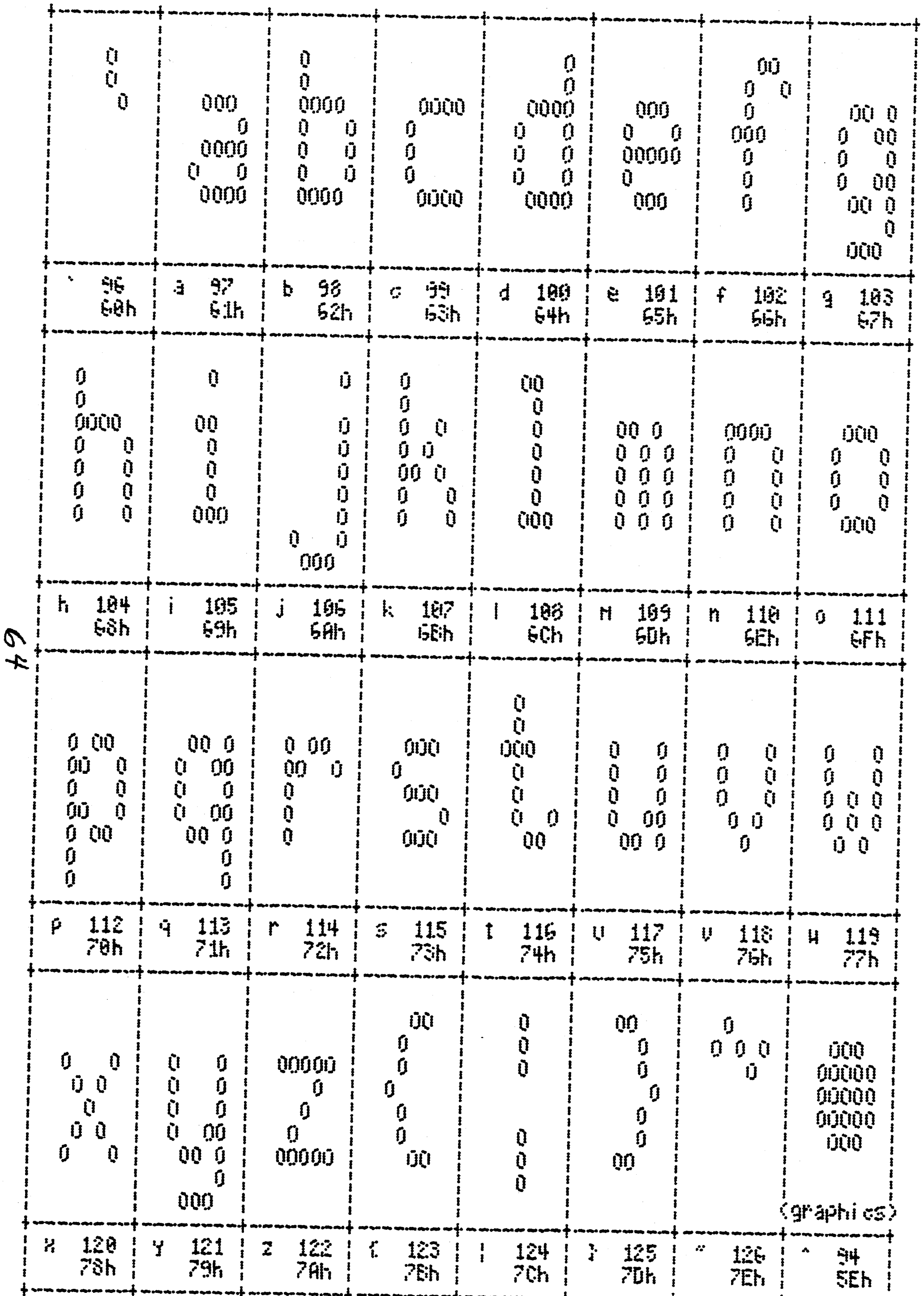
STANDARD Character Set 0



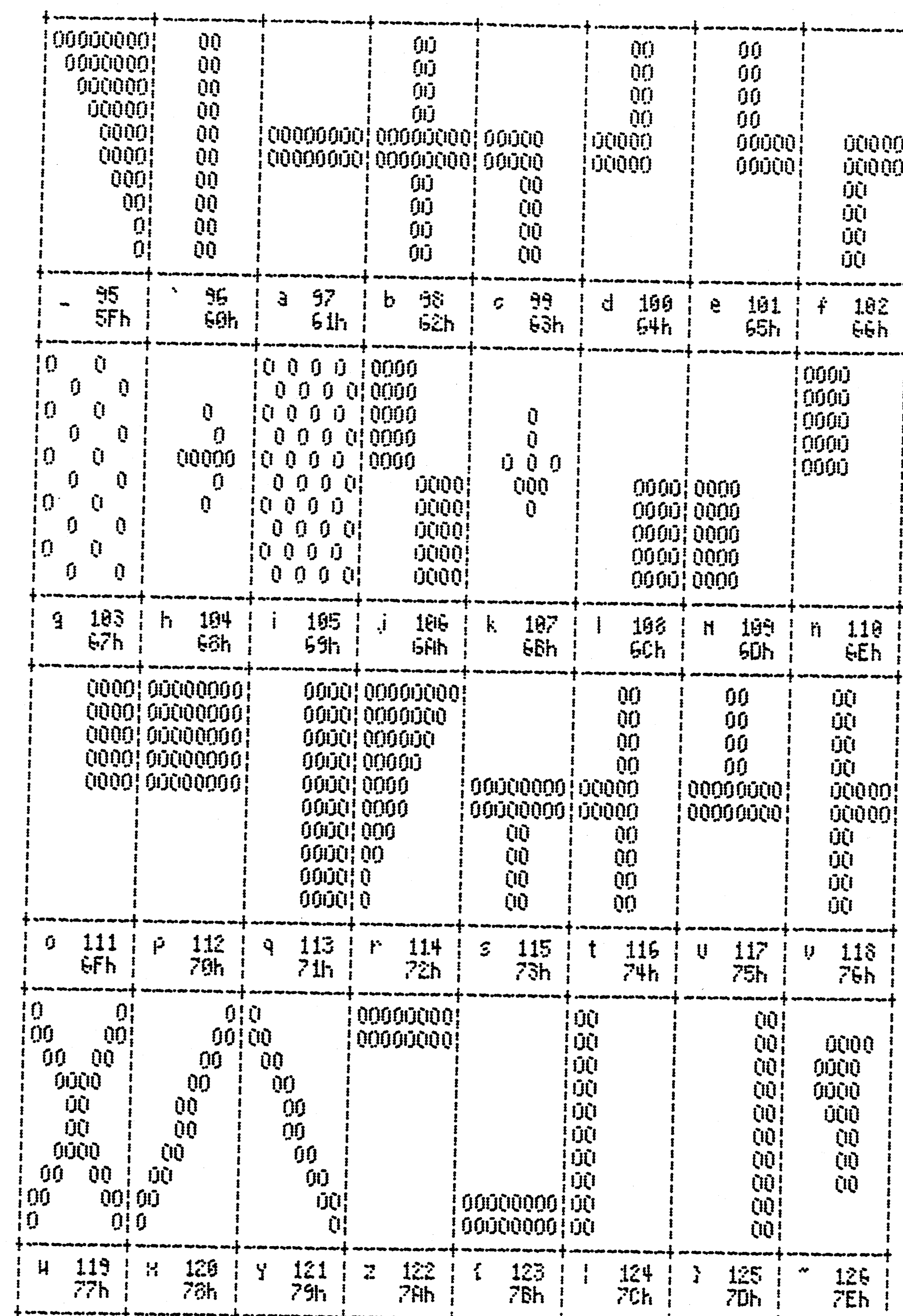
HEATH H19 text mode



STANDARD Character Set



HEATH H19 graphics mode



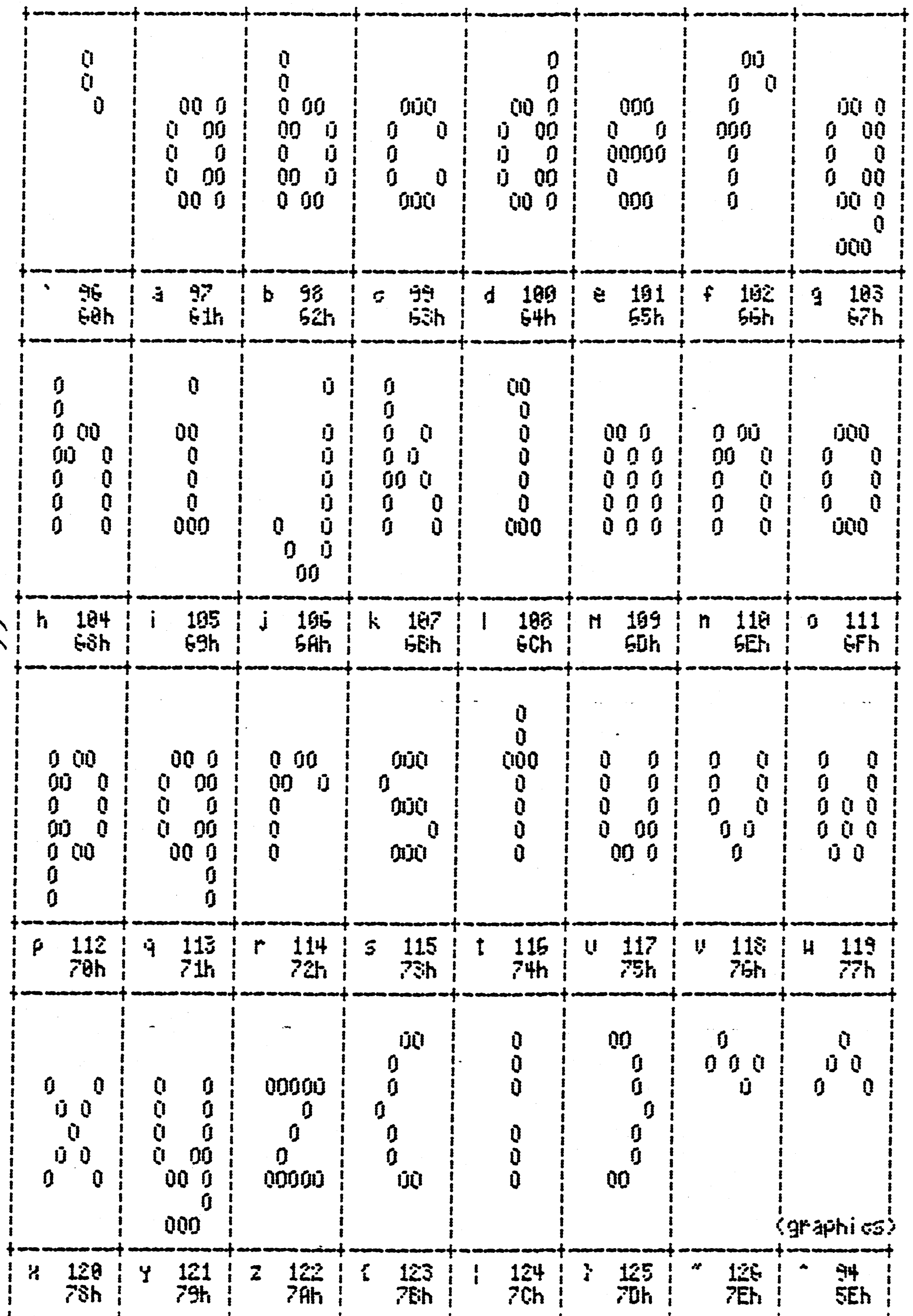
STANDARD Character Set A

32 20h	! 33 21h	" 34 22h	# 35 23h	$ 36 24h	% 37 25h	& 38 26h	' 39 27h
(40 28h	) 41 29h	* 42 2Ah	+ 43 2Bh	, 44 2Ch	- 45 2Dh	. 46 2Eh	/ 47 2Fh
0 48 30h	1 49 31h	2 50 32h	3 51 33h	4 52 34h	5 53 35h	6 54 36h	7 55 37h
8 56 38h	9 57 39h	: 58 3Ah	; 59 3Bh	< 60 3Ch	= 61 3Dh	> 62 3Eh	? 63 3Fh

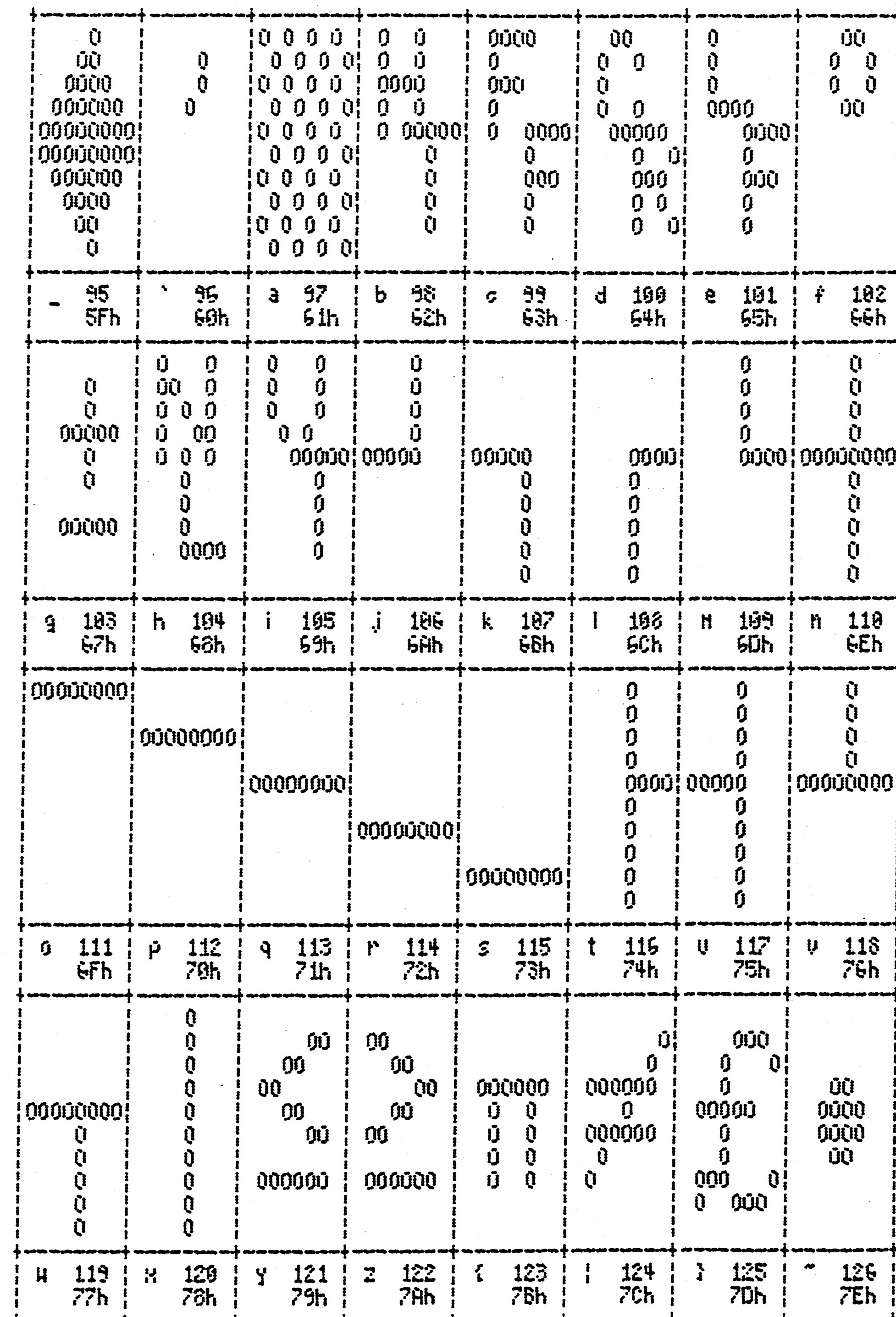
DEC VT-100 text mode

@ 64 40h	A 65 41h	B 66 42h	C 67 43h	D 68 44h	E 69 45h	F 70 46h	G 71 47h
H 72 48h	I 73 49h	J 74 4Ah	K 75 4Bh	L 76 4Ch	M 77 4Dh	N 78 4Eh	O 79 4Fh
P 80 50h	Q 81 51h	R 82 52h	S 83 53h	T 84 54h	U 85 55h	V 86 56h	W 87 57h
X 88 58h	Y 89 59h	Z 90 5Ah	[91 5Bh	\ 92 5Ch	] 93 5Dh	^ 94 5Eh	_ 95 5Fh

STANDARD Character Set



DEC VT-100 graphics mode



STANDARD Character Set L

NORCOM GT-PROM Text Mode

```
       |   00   | 00 00  |  00 0  |     0  | 00     |   00   |   00
       |   00   | 00 00  |  00 0  |  00000 | 00  00 |  00 0  |   00
       |   00   | 00 00  | 000000 | 00 0   |    00  |  000   |  00
       |   00   |        |  00 0  |  0000  |   00   |  00    |
       |   00   |        | 000000 |   0 00 |  00    | 00 0 0 |
       |        |        |  00 0  | 00000  | 00  00 | 00  0  |
       |   00   |        |  00 0  |  0     |     00 |  000 0 |
-------+--------+--------+--------+--------+--------+--------+-------
    32 | !   33 | "   34 | #   35 | $   36 | %   37 | &   38 | '   39
   20h |    21h |    22h |    23h |    24h |    25h |    26h |    27h
```

```
    00 |  00    |        |        |        |        |        |     00
   00  |   00   |   00   |   00   |        |        |        |    00
  00   |    00  | 0 00 0 |   00   |        |        |        |    00
  00   |    00  |  0000  | 000000 |        | 000000 |        |   00
  00   |    00  | 0 00 0 |   00   |        |        |        |   00
   00  |   00   |   00   |   00   |   000  |        |  000   |  00
    00 |  00    |        |        |   000  |        |  000   | 00
       |        |        |        |    00  |        |        |
       |        |        |        |   00   |        |        |
-------+--------+--------+--------+--------+--------+--------+-------
(   40 | )   41 | *   42 | +   43 | ,   44 | -   45 | .   46 | /   47
   28h |    29h |    2Ah |    2Bh |    2Ch |    2Dh |    2Eh |    2Fh
```

```
 0000  |   00   |  0000  |  0000  |    000 | 000000 |  0000  | 000000
00   0 |  0000  | 0   00 | 0  000 |   0000 | 0      | 000  0 |     00
00   0 |   000  |     00 |    000 |  0 000 | 00000  | 000    |    00
00 0 0 |   000  |    000 |   00   | 0  000 |    000 | 00000  |   00
00   0 |   000  |   000  |    000 | 000000 |    000 | 000  0 |  000
00   0 |   000  |  000   | 0  000 |    000 | 0  000 | 000  0 | 000
 0000  |   000  | 000000 |  0000  |    000 |  0000  |  0000  | 000
-------+--------+--------+--------+--------+--------+--------+-------
0   48 | 1   49 | 2   50 | 3   51 | 4   52 | 5   53 | 6   54 | 7   55
   30h |    31h |    32h |    33h |    34h |    35h |    36h |    37h
```

```
 0000  |  0000  |        |        |     00 |        | 00     |  0000
00   0 | 0  000 |  000   |  000   |    00  |        |  00    | 0   00
000  0 | 0  000 |  000   |  000   |   00   | 000000 |   00   |     00
 0000  |  00000 |        |        |  00    |        |    00  |     0
0  000 |    000 |  000   |  000   |   00   | 000000 |   00   |    00
0   00 | 0  000 |  000   |  000   |    00  |        |  00    |
 0000  |  0000  |        |   00   |     00 |        | 00     |    00
       |        |        |  00    |        |        |        |
-------+--------+--------+--------+--------+--------+--------+-------
8   56 | 9   57 | :   58 | ;   59 | <   60 | =   61 | >   62 | ?   63
   38h |    39h |    3Ah |    3Bh |    3Ch |    3Dh |    3Eh |    3Fh
```

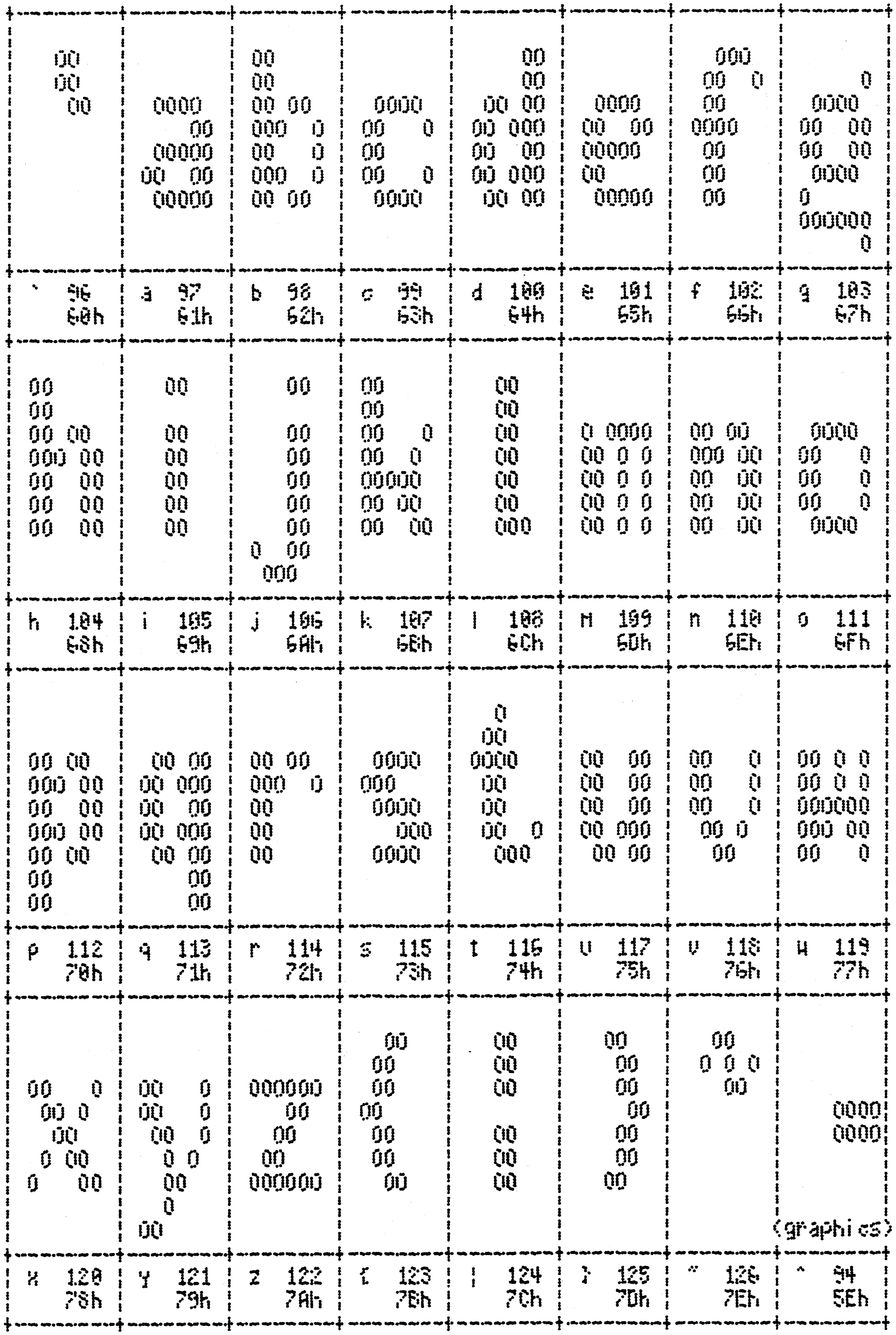
```
 000   |   0    | 00000  |  0000  | 0000   | 000000 | 000000 |  0000
0   0  |  000   | 000  0 | 000  0 | 000 0  | 000    | 000    | 000  0
00 000 | 0 000  | 000  0 | 000    | 000  0 | 000000 | 000000 | 000
00 0 0 | 0  000 | 000 0  | 000    | 000  0 | 000    | 000    | 000 00
00 000 | 0  000 | 000  0 | 000  0 | 000  0 | 000    | 000    | 000  0
0      | 000000 | 000  0 | 000  0 | 000  0 | 000    | 000    | 000  0
 0000  | 0  000 | 00000  |  0000  | 00000  | 000000 | 000    |  0000
-------+--------+--------+--------+--------+--------+--------+-------
@   64 | A   65 | B   66 | C   67 | D   68 | E   69 | F   70 | G   71
   40h |    41h |    42h |    43h |    44h |    45h |    46h |    47h
```

```
000  0 |  000   |    000 | 000  0 | 000    | 0    0 | 00   0 |  0000
000  0 |  000   |    000 | 000 0  | 000    | 00  00 | 000  0 | 000  0
000000 |  000   |    000 | 0000   | 000    | 00 000 | 0000 0 | 000  0
000  0 |  000   |    000 | 000 0  | 000    | 0 0 00 | 0 00 0 | 000  0
000  0 |  000   |    000 | 000  0 | 000    | 0 0 00 | 0 0000 | 000  0
000  0 |  000   | 0  000 | 000  0 | 000    | 0   00 | 0  000 | 000  0
000  0 |  000   |  0000  | 000  0 | 000000 | 0   00 | 0   00 |  0000
-------+--------+--------+--------+--------+--------+--------+-------
H   72 | I   73 | J   74 | K   75 | L   76 | M   77 | N   78 | O   79
   48h |    49h |    4Ah |    4Bh |    4Ch |    4Dh |    4Eh |    4Fh
```

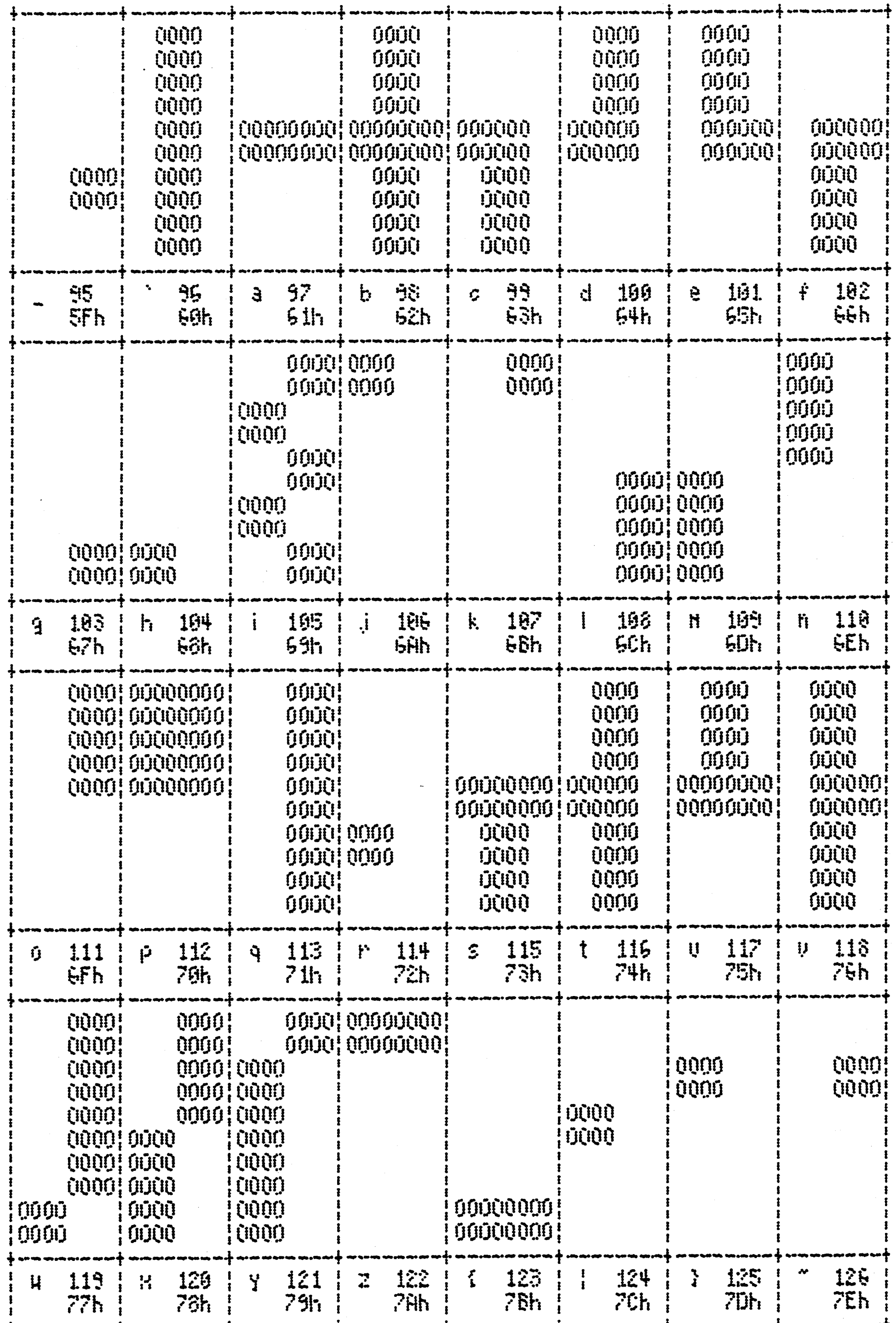
```
00000  |  0000  | 00000  |  0000  | 000000 | 000  0 | 000  0 | 00   0
000  0 | 00   0 | 000  0 | 000  0 |  000   | 000  0 | 000  0 | 00   0
000  0 | 00   0 | 000  0 |  000   |  000   | 000  0 | 000  0 | 00 0 0
000  0 | 00   0 | 000  0 |   000  |  000   | 000  0 | 000  0 | 00 0 0
00000  | 00 0 0 | 00000  |    000 |  000   | 000  0 |  00  0 | 000 00
000    | 00  0  | 000 0  | 0  000 |  000   | 000  0 |   0 0  | 00  00
000    |  000 0 | 000  0 |  0000  |  000   |  0000  |    0   | 0    0
-------+--------+--------+--------+--------+--------+--------+-------
P   80 | Q   81 | R   82 | S   83 | T   84 | U   85 | V   86 | W   87
   50h |    51h |    52h |    53h |    54h |    55h |    56h |    57h
```

```
000  0 | 000  0 | 000000 | 0000   | 00     | 0000   |   00   |
000  0 | 000  0 |     00 | 00     |  00    |   00   |  00 0  |
 00 0  | 000  0 |    000 | 00     |   00   |   00   | 00   0 |
  00   |  00 0  |   000  | 00     |    00  |   00   |        |
 0 00  |   00   |  000   | 00     |     00 |   00   |        |
0  000 |  00    | 000    | 00     |      00|   00   |        |
0  000 | 00     | 000000 | 0000   |       00| 0000  |        | 000000
-------+--------+--------+--------+--------+--------+--------+-------
X   88 | Y   89 | Z   90 | [   91 | \   92 | ]   93 | ^   94 | _   95
   58h |    59h |    5Ah |    5Bh |    5Ch |    5Dh |    5Eh |    5Fh
```

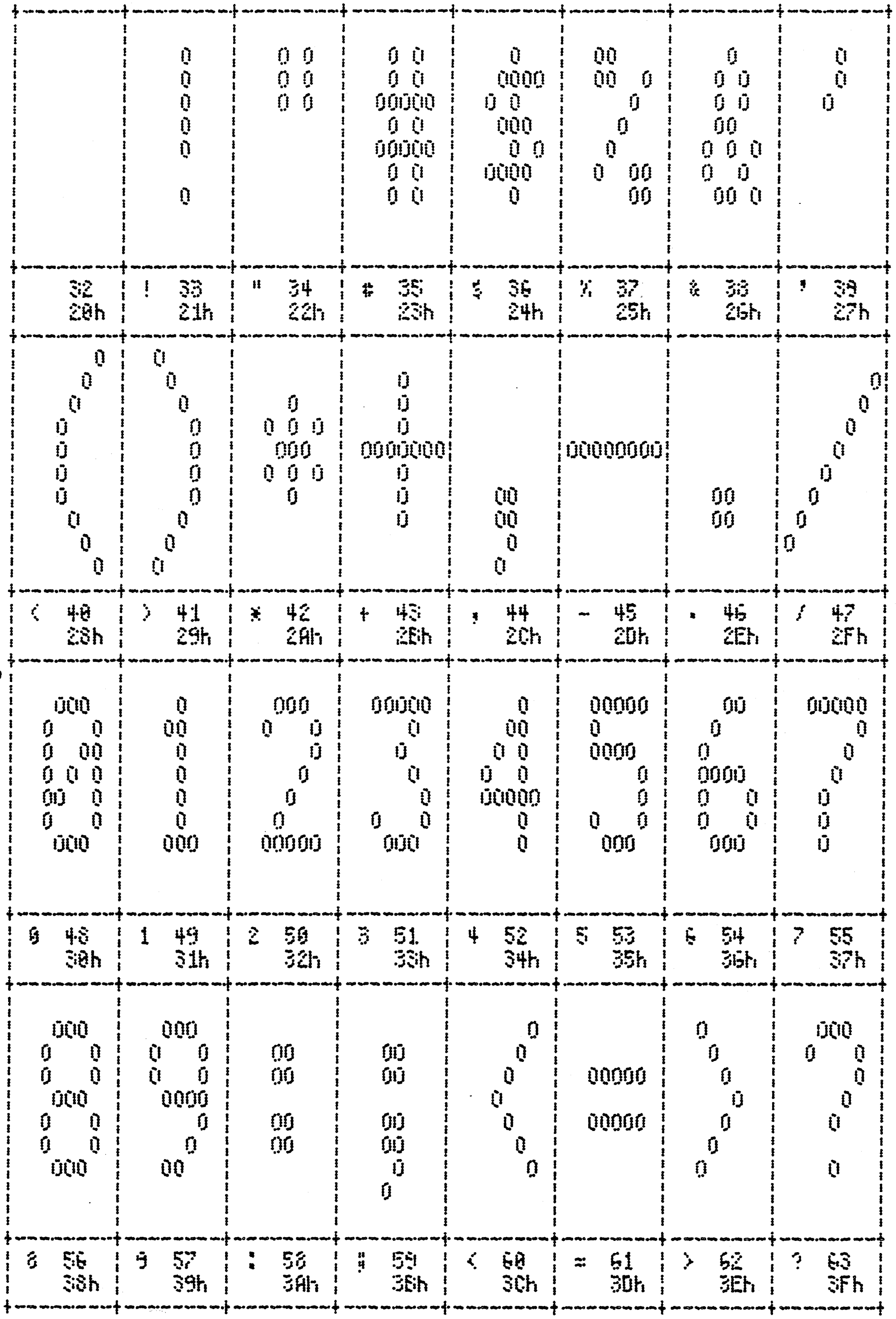
STANDARD Character Set L



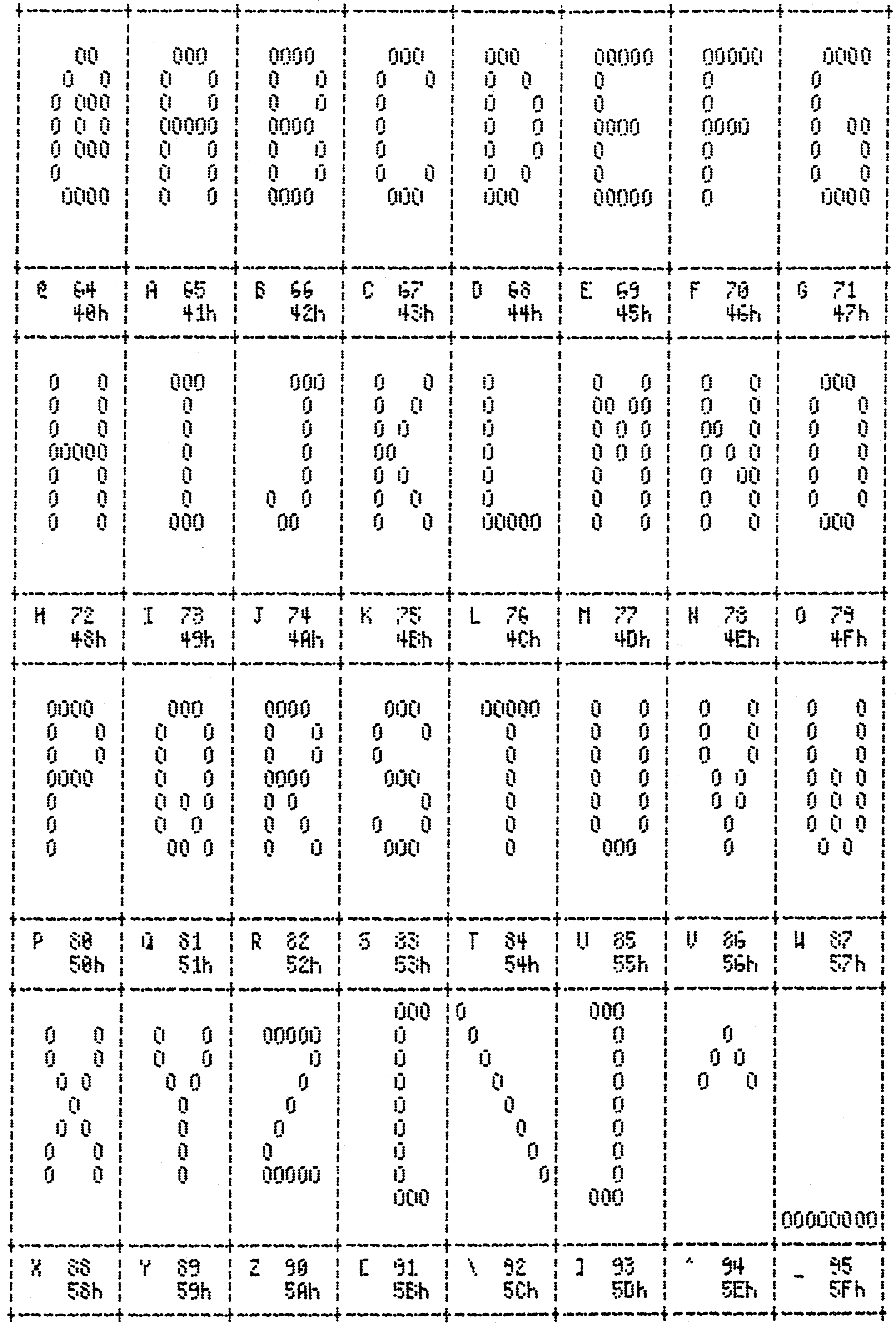
NORCOM GT-PROM graphics mode



STANDARD Character Set P



HEATH + GREEK text mode



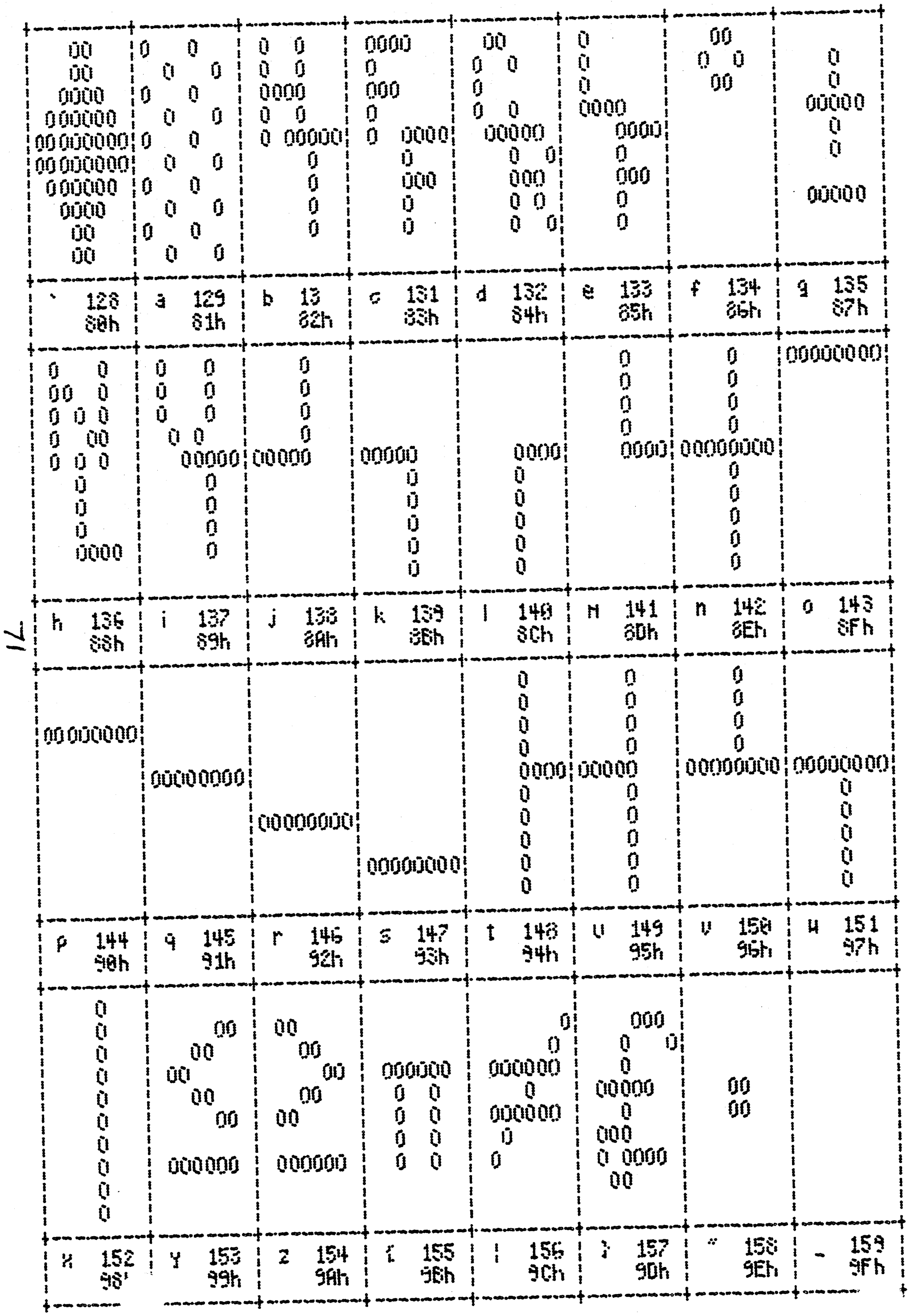
STANDARD Character Set

` 96 60h	a 97 61h	b 98 62h	c 99 63h	d 100 64h	e 101 65h	f 102 66h	g 103 67h
h 104 68h	i 105 69h	j 106 6Ah	k 107 6Bh	l 108 6Ch	m 109 6Dh	n 110 6Eh	o 111 6Fh
p 112 70h	q 113 71h	r 114 72h	s 115 73h	t 116 74h	u 117 75h	v 118 76h	w 119 77h
x 120 78h	y 121 79h	z 122 7Ah	{ 123 7Bh	| 124 7Ch	} 125 7Dh	~ 126 7Eh	(graphics) ^ 94 5Eh

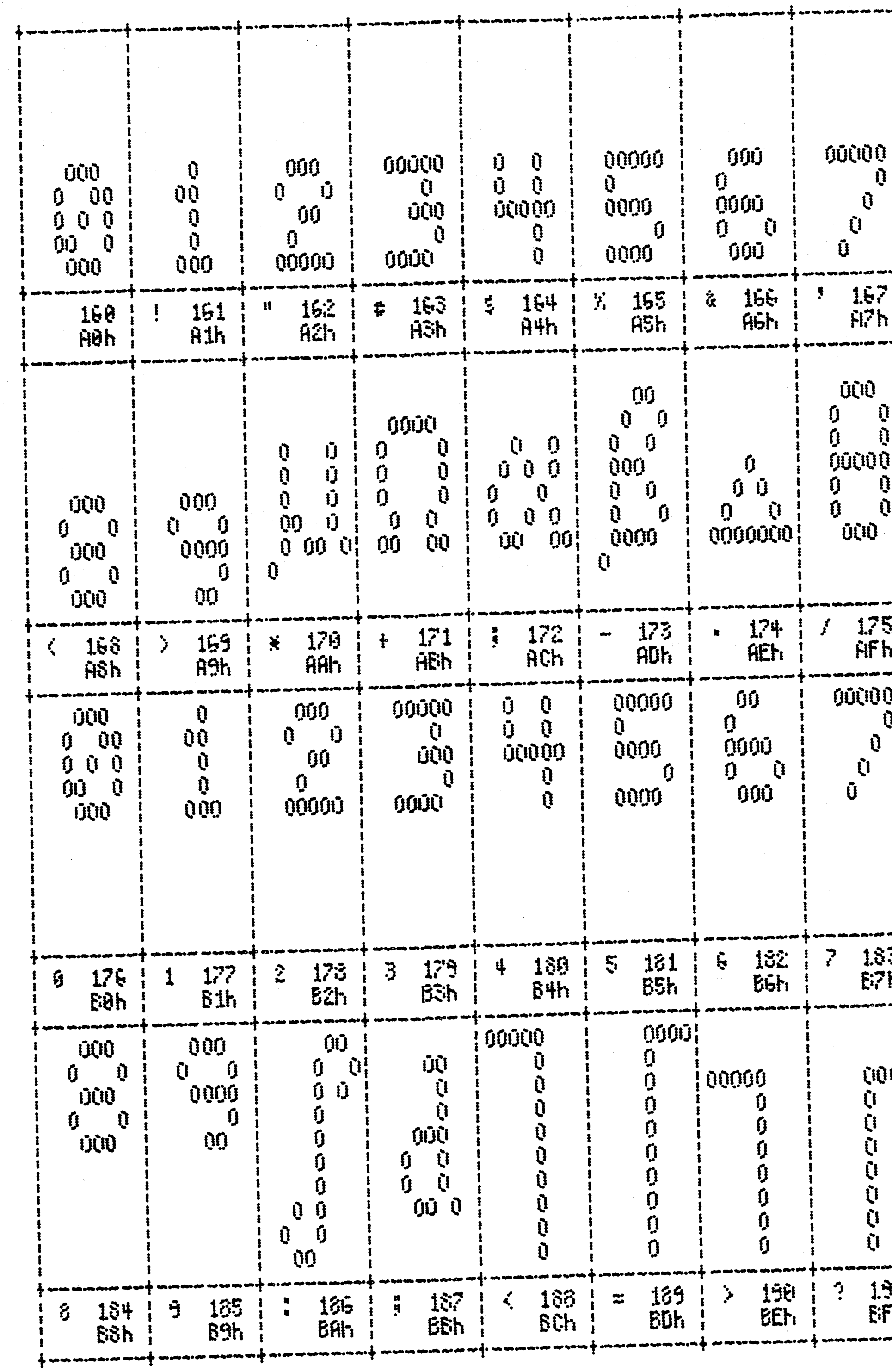
HEATH + GREEK graphics mode

_ 95 5Fh	` 96 60h	a 97 61h	b 98 62h	c 99 63h	d 100 64h	e 101 65h	f 102 66h
g 103 67h	h 104 68h	i 105 69h	j 106 6Ah	k 107 6Bh	l 108 6Ch	m 109 6Dh	n 110 6Eh
o 111 6Fh	p 112 70h	q 113 71h	r 114 72h	s 115 73h	t 116 74h	u 117 75h	v 118 76h
w 119 77h	x 120 78h	y 121 79h	z 122 7Ah	{ 123 7Bh	| 124 7Ch	} 125 7Dh	~ 126 7Eh

SUPERFONT Character Set C



Heath H29 reverse video, text mode



SUPERFONT Character Set C

@ 192 C0h	A 193 C1h	B 194 C2h	C 195 C3h	D 196 C4h	E 197 C5h	F 198 C6h	G 199 C7h
H 200 C8h	I 201 C9h	J 202 CAh	K 203 CBh	L 204 CCh	M 205 CDh	N 206 CEh	O 207 CFh
P 208 D0h	Q 209 D1h	R 210 D2h	S 211 D3h	T 212 D4h	U 213 D5h	V 214 D6h	W 215 D7h
X 216 D8h	Y 217 D9h	Z 218 DAh	[219 DBh	\ 220 DCh	] 221 DDh	^ 222 DEh	_ 223 DFh

Heath H29 reverse video, text mode

` 224 E0h	a 225 E1h	b 226 E2h	c 227 E3h	d 228 E4h	e 229 E5h	f 230 E6h	g 231 E7h
h 232 E8h	i 233 E9h	j 234 EAh	k 235 EBh	l 236 ECh	m 237 EDh	n 238 EEh	o 239 EFh
p 240 F0h	q 241 F1h	r 242 F2h	s 243 F3h	t 244 F4h	u 245 F5h	v 246 F6h	w 247 F7h
x 248 F8h	y 249 F9h	z 250 FAh	{ 251 FBh		252 FCh	} 253 FDh	~ 254 FEh	^ 255 FFh

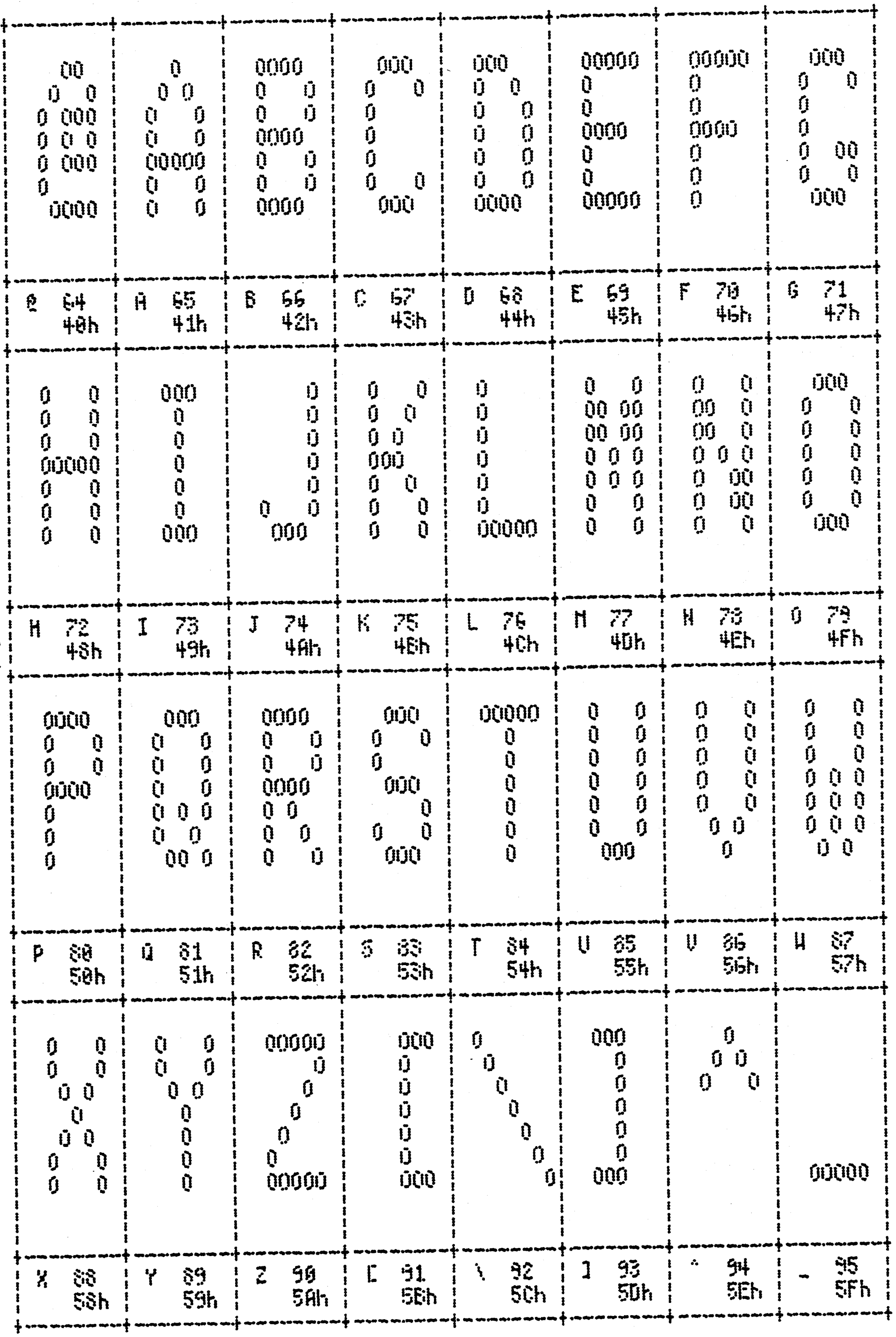
SUPERFONT Character Set 0 - Normal Video, Graphics Mode

` 0 00h	a 1 01h	b 2 02h	c 3 03h	d 4 04h	e 5 05h	f 6 06h	g 7 07h
h 8 08h	i 9 09h	j 10 0Ah	k 11 0Bh	l 12 0Ch	m 13 0Dh	n 14 0Eh	o 15 0Fh
p 16 10h	q 17 11h	r 18 12h	s 19 13h	t 20 14h	u 21 15h	v 22 16h	w 23 17h
x 24 18h	y 25 19h	z 26 1Ah	{ 27 1Bh	| 28 1Ch	} 29 1Dh	~ 30 1Eh	_ 31 1Fh

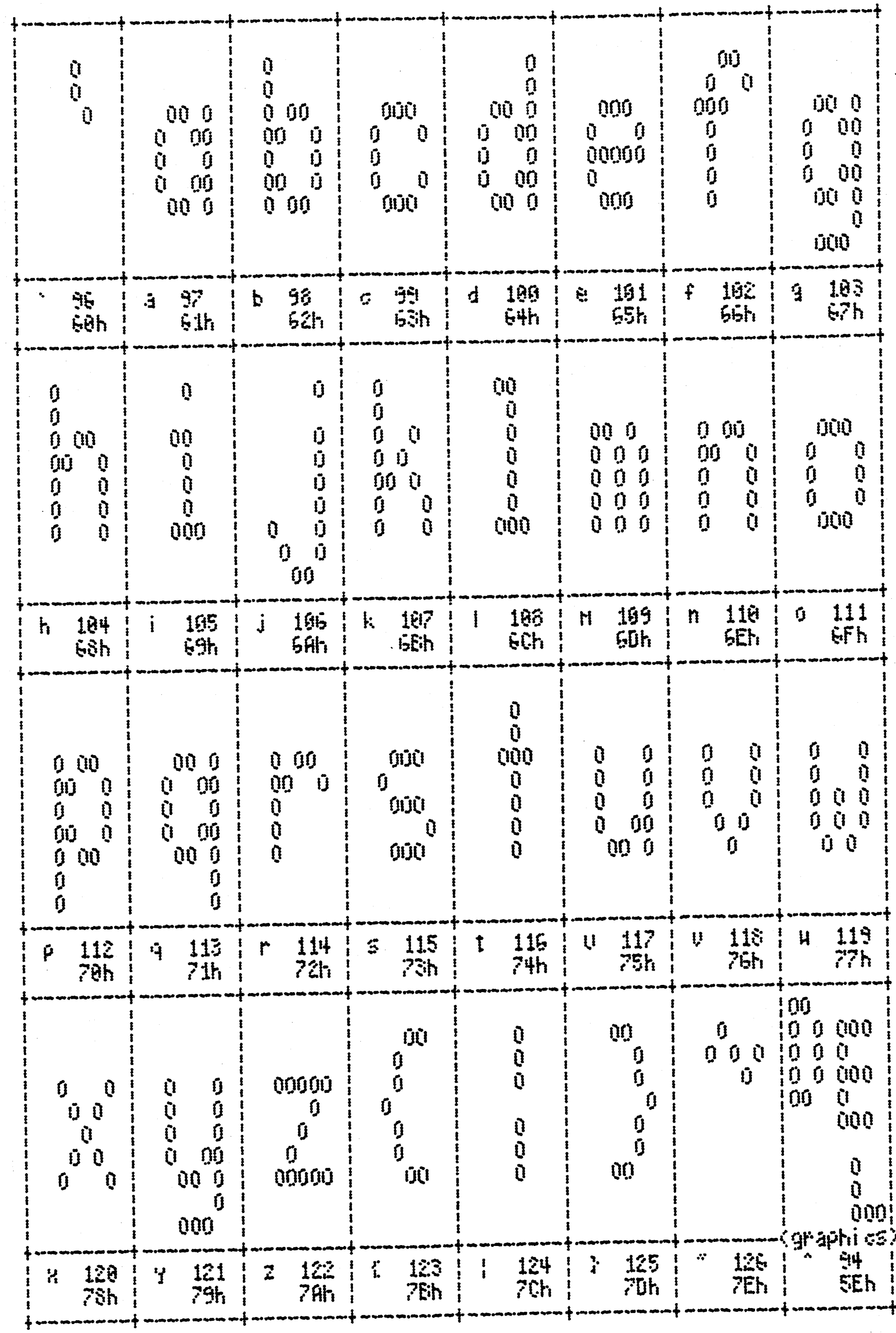
SUPERFONT Character Set 0 - Normal Video, Text Mode

32 20h	! 33 21h	¨ 34 22h	# 35 23h	$ 36 24h	% 37 25h	& 38 26h	´ 39 27h
(40 28h	) 41 29h	* 42 2Ah	+ 43 2Bh	, 44 2Ch	- 45 2Dh	. 46 2Eh	/ 47 2Fh
0 48 30h	1 49 31h	2 50 32h	3 51 33h	4 52 34h	5 53 35h	6 54 36h	7 55 37h
8 56 38h	9 57 39h	: 58 3Ah	; 59 3Bh	< 60 3Ch	= 61 3Dh	> 62 3Eh	? 63 3Fh

SUPERFONT Character Set G



enhanced VT-100, normal video, text mode



SUPERFONT Character Set 6

160 A0h	` 224 grph E0h	! 161 A1h	a 225 grph E1h	" 162 A2h	b 226 grph E2h	‡ 163 A3h	c 227 grph E3h
$ 164 A4h	d 228 grph E4h	% 165 A5h	e 229 grph E5h	& 166 A6h	f 230 grph E6h	' 167 A7h	g 231 grph E7h
(168 A8h	h 232 grph E8h	) 169 A9h	i 233 grph E9h	* 170 AAh	j 234 grph EAh	+ 171 ABh	k 235 grph EBh
, 172 ACh	l 236 grph ECh	- 173 ADh	m 237 grph EDh	. 174 AEh	n 238 grph EEh	/ 175 AFh	o 239 grph EFh

VT100, reverse video (double-wide)

0 176 B0h	p 240 grph F0h	1 177 B1h	q 241 grph F1h	2 178 B2h	r 242 grph F2h	3 179 B3h	s 243 grph F3h
4 180 B4h	t 244 grph F4h	5 181 B5h	u 245 grph F5h	6 182 B6h	v 246 grph F6h	7 183 B7h	w 247 grph F7h
8 184 B8h	x 248 grph F8h	9 185 B9h	y 249 grph F9h	: 186 BAh	z 250 grph FAh	; 187 BBh	{ 251 grph FBh
< 188 BCh	| 252 grph FCh	= 189 BDh	} 253 grph FDh	> 190 BEh	~ 254 grph FEh	? 191 BFh	_ 255 grph FFh

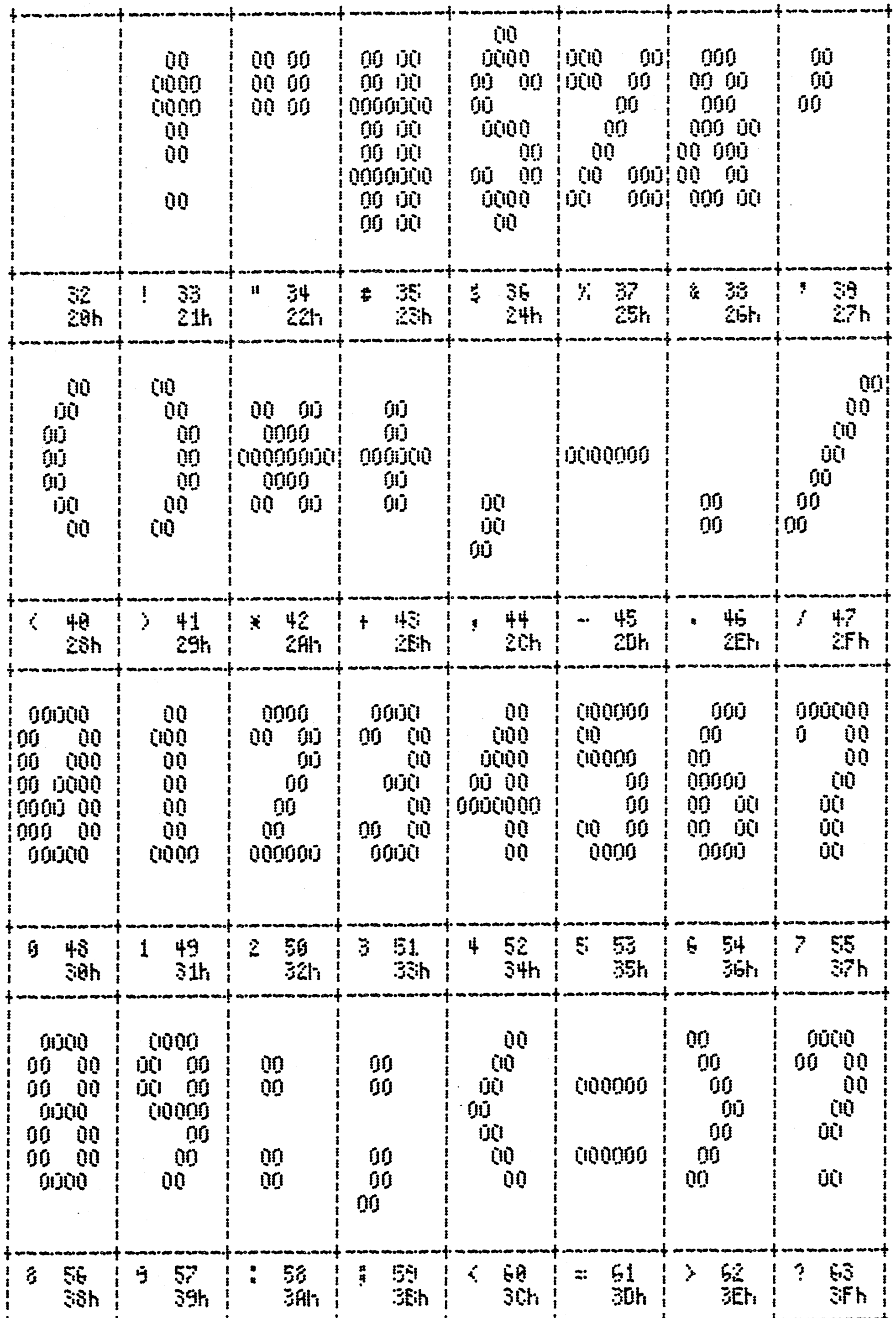
SUPERFONT Character Set

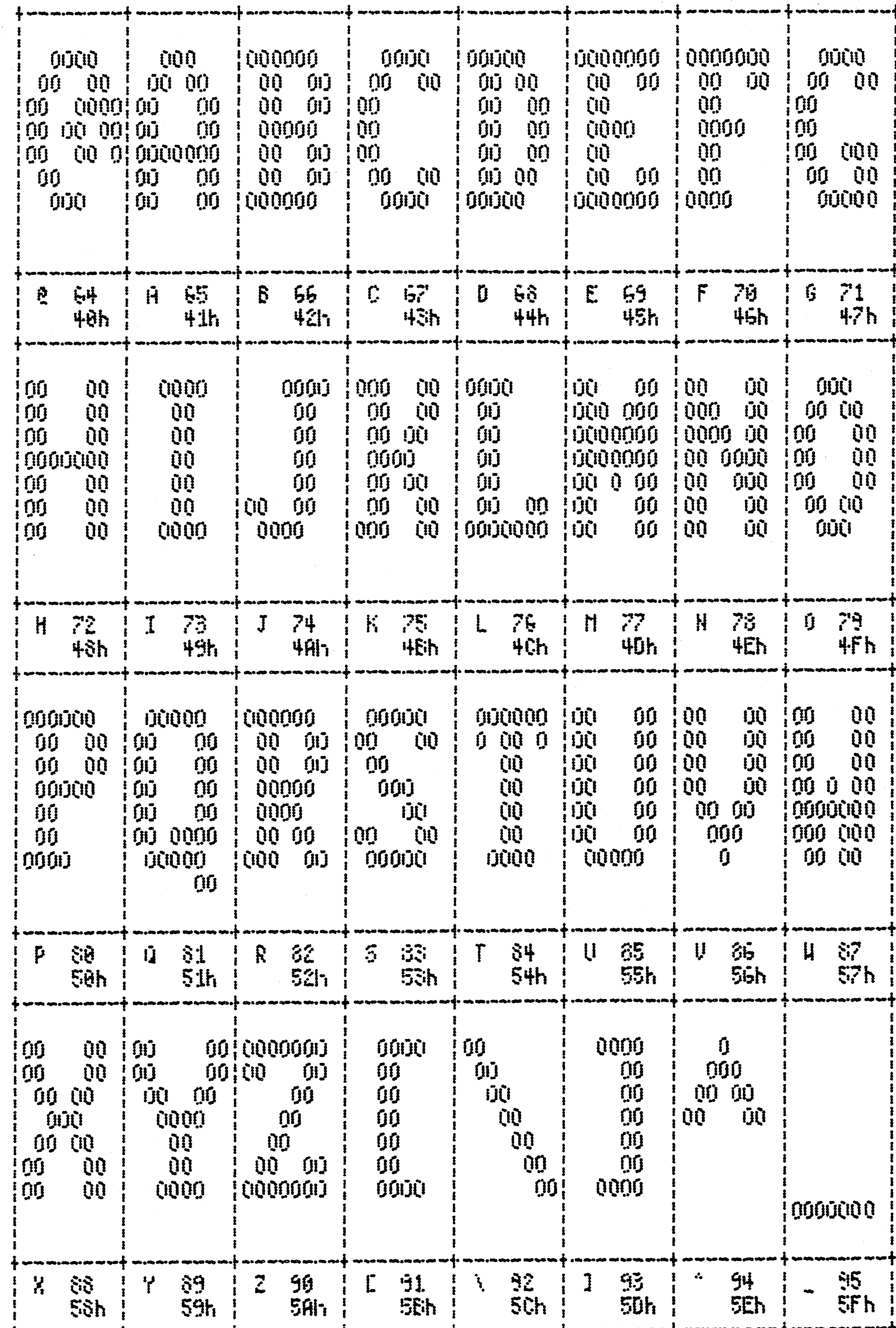
@ 192 C0h	` 224 E0h	A 193 C1h	a 225 E1h	B 194 C2h	b 226 E2h	C 195 C3h	c 227 E3h
D 196 C4H	d 228 E4h	E 197 C5h	e 229 E5h	F 198 C6h	f 230 E6h	G 199 C7h	g 231 E7h
H 200 C8h	h 232 E8h	I 201 C9h	i 233 E9h	J 202 CAh	j 234 EAh	K 203 CBh	k 235 EBh
L 204 CCh	l 236 ECh	M 205 CDh	m 237 EDh	N 206 CEh	n 238 EEh	O 207 CFh	o 239 EFh

VT100, reverse video (double-wide)

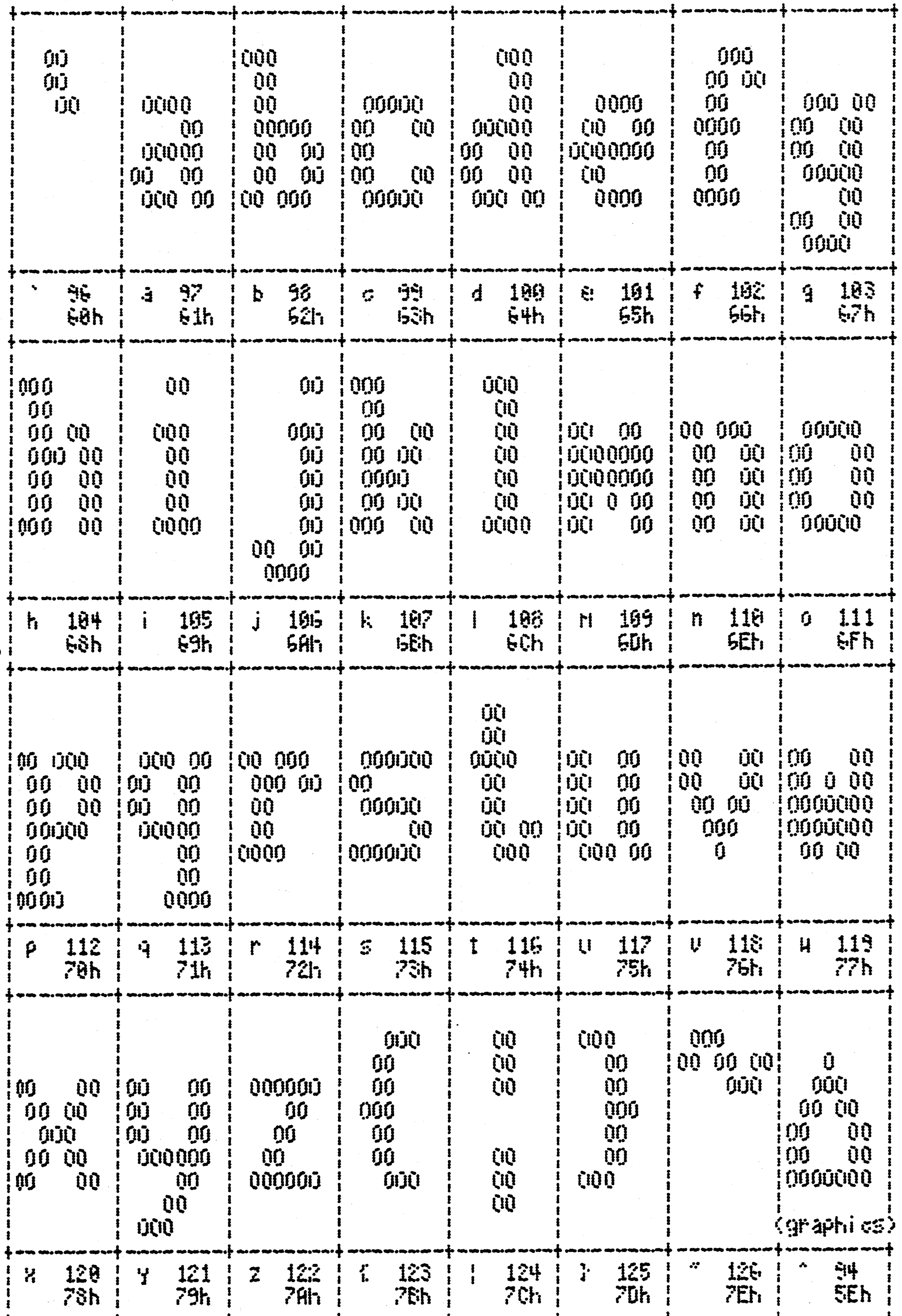
P 208 D0h	p 240 F0h	Q 209 D1h	q 241 F1h	R 210 D2h	r 242 F2h	S 211 D3h	s 243 F3h
T 212 D4h	t 244 F4h	U 213 D5h	u 245 F5h	V 214 D6h	v 246 F6h	W 215 D7h	w 247 F7h
X 216 D8h	x 248 F8h	Y 217 D9h	y 249 F9h	Z 218 DAh	z 250 FAh	[219 DBh	{ 251 FBh
\ 220 DCh	| 252 FCh	] 221 DDh	} 253 FDh	^ 222 DEh	~ 254 FEh	_ 223 DFh	^ 222 graph DEh

SUPERFONT Character Set 0 IBM PC, normal video, text mode

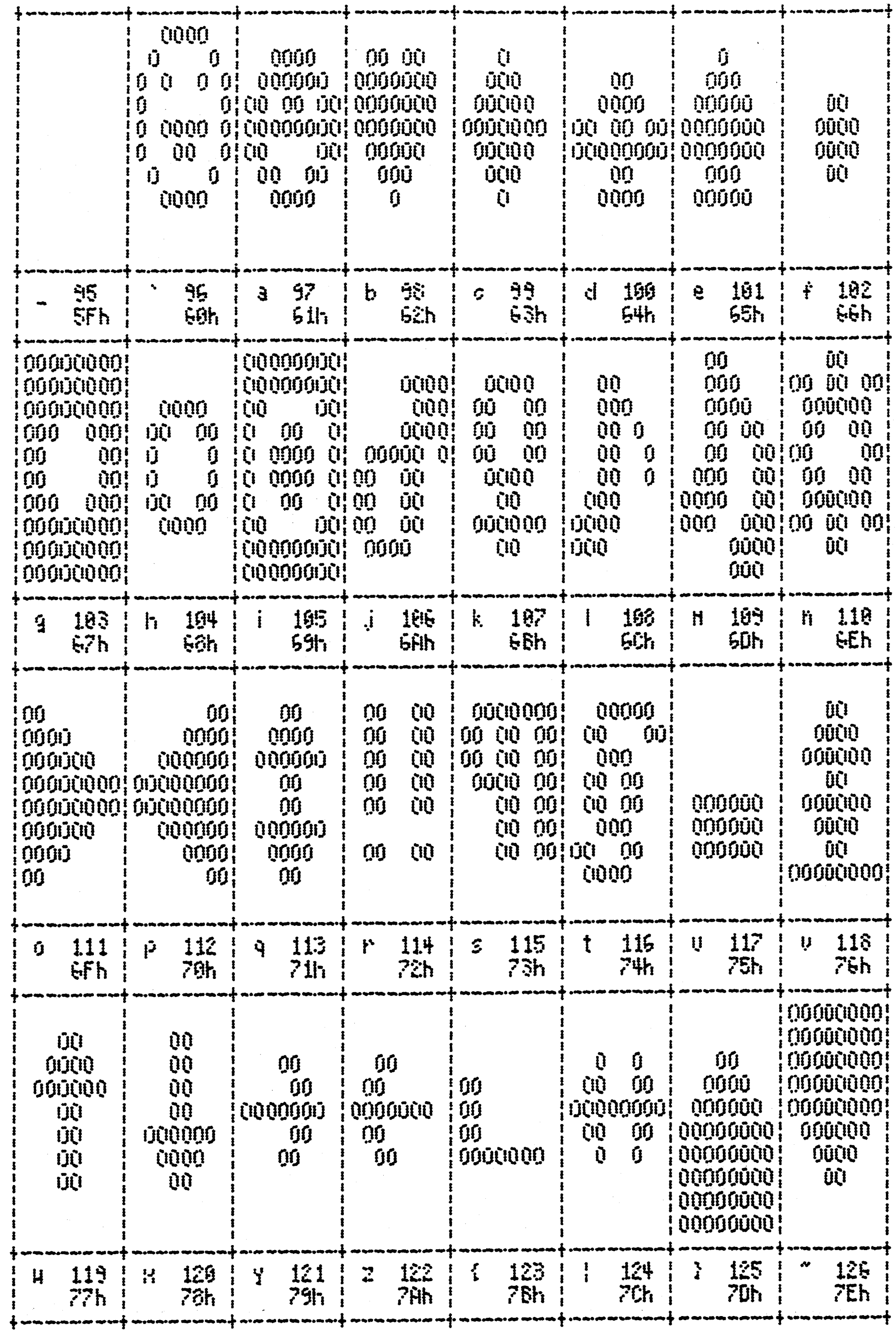




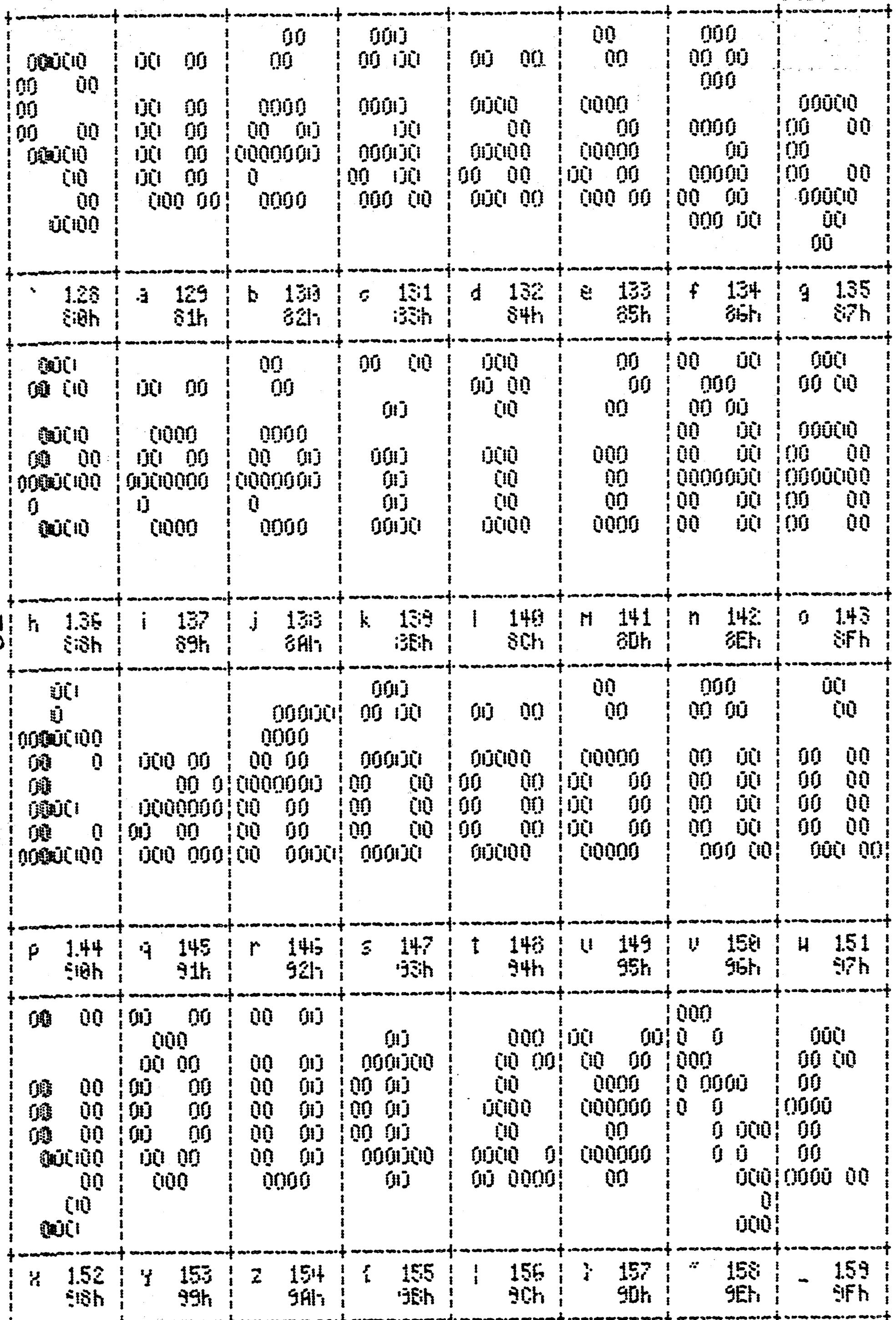
SUP ONT Character Set 0 - IBM PC, normal video, text mode



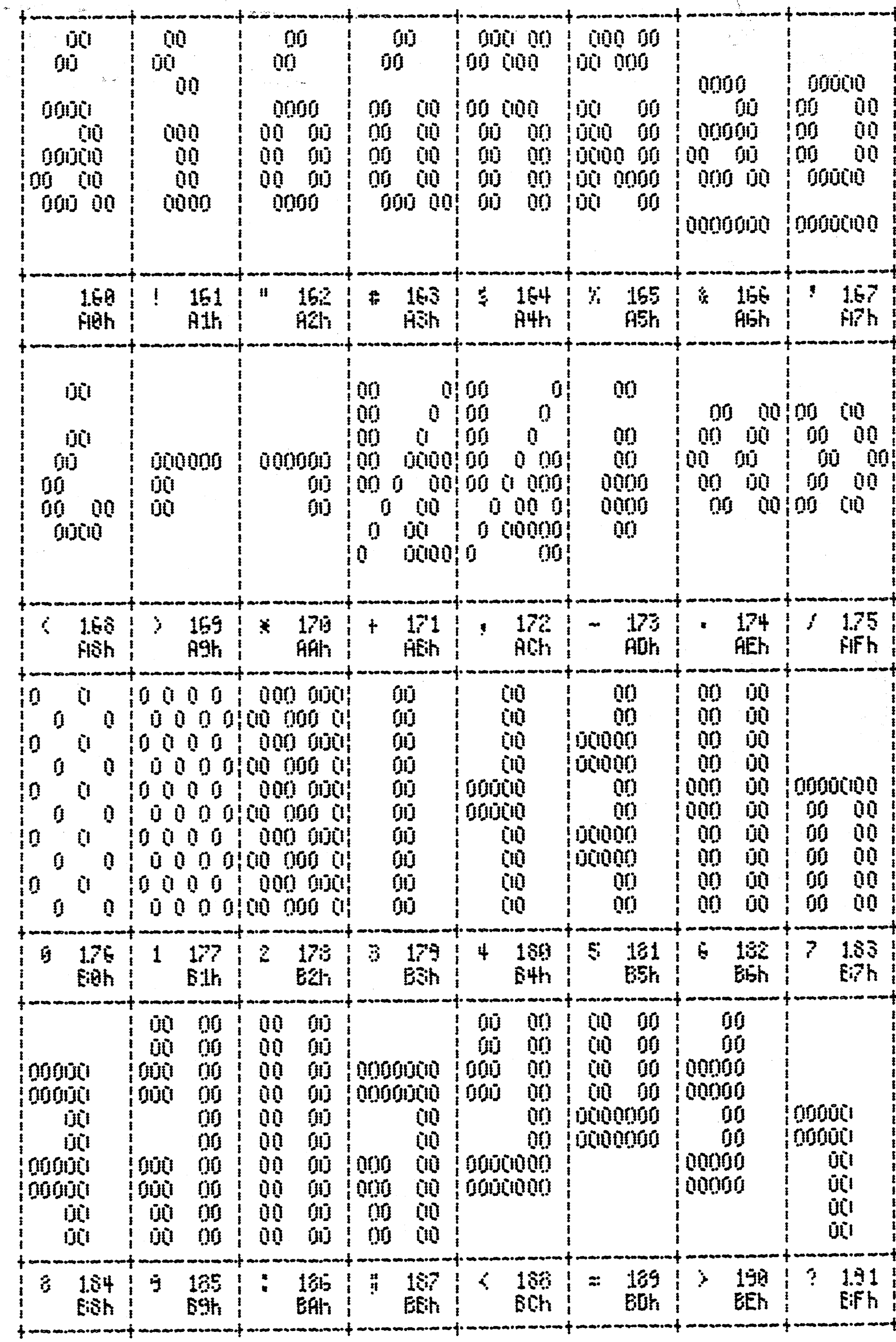
SUPERFONT Character Set 0 - IBM PC normal video, phics Mode



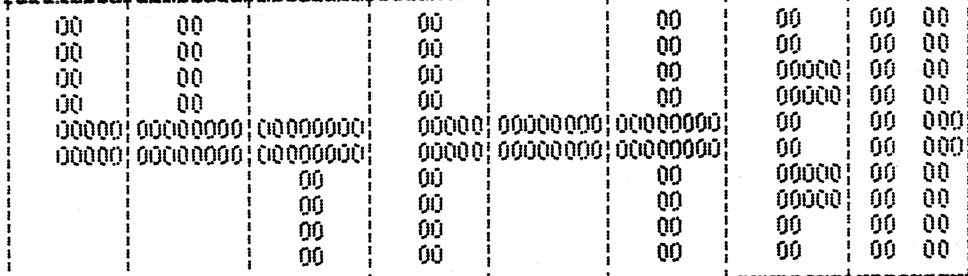
SUPERFONT Character Set 0 - IBM PC, reverse video, graphics mode



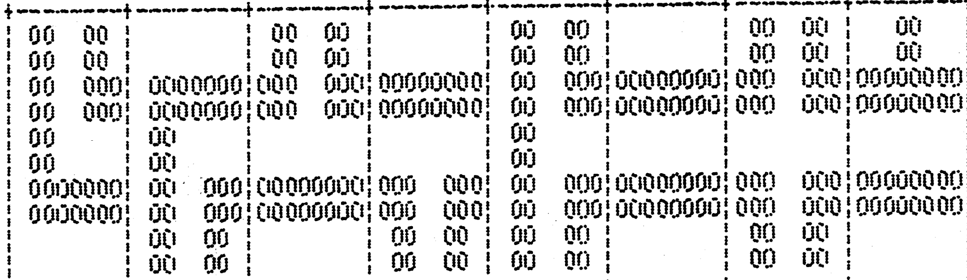
SUPERFONT Character Set 0 - IBM PC, reverse video, text mode



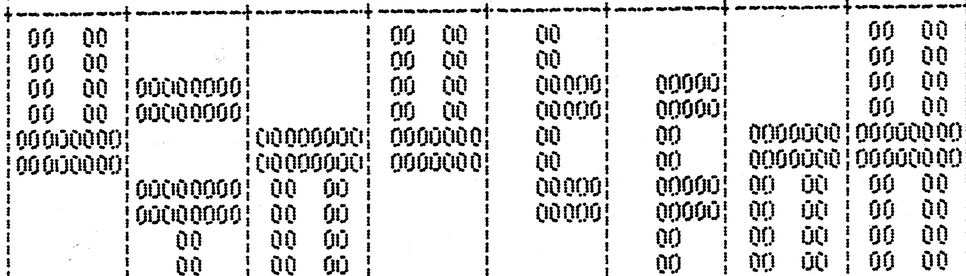
SUPERFONT Character Set 0



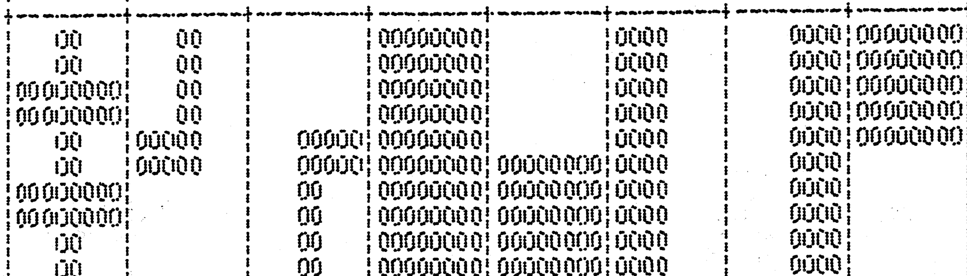
@ 192 C0h	A 193 C1h	B 194 C2h	C 195 C3h	D 196 C4h	E 197 C5h	F 198 C6h	G 199 C7h



H 200 C8h	I 201 C9h	J 202 CAh	K 203 CBh	L 204 CCh	M 205 CDh	N 206 CEh	O 207 CFh

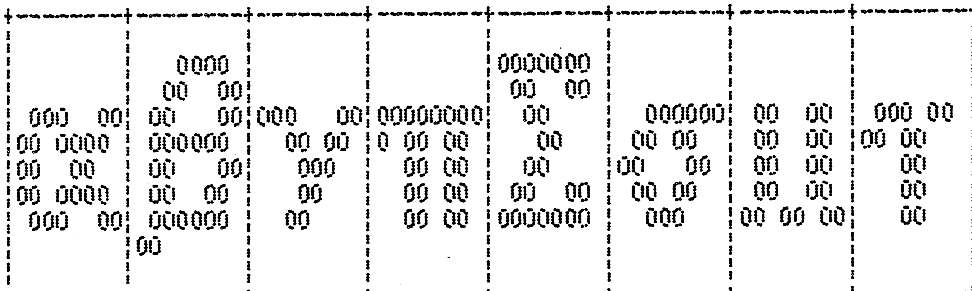


P 208 D0h	Q 209 D1h	R 210 D2h	S 211 D3h	T 212 D4h	U 213 D5h	V 214 D6h	W 215 D7h

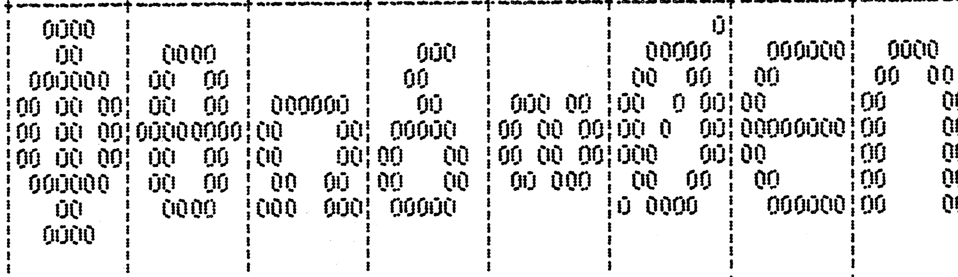


X 216 D8h	Y 217 D9h	Z 218 DAh	[219 DBh	\ 220 DCh	] 221 DDh	^ 222 DEh	_ 223 DFh

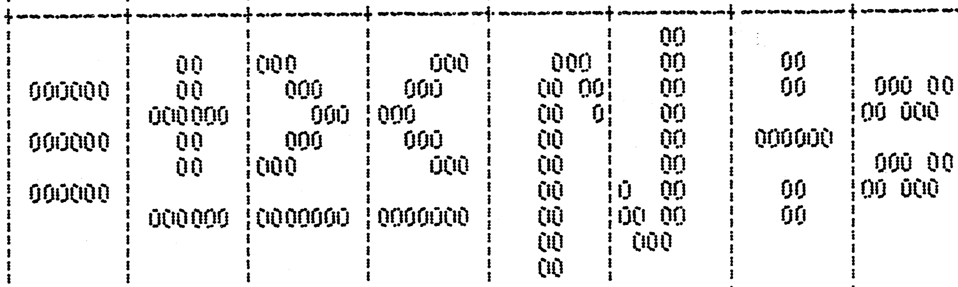
IBM PC, reverse video, text mode



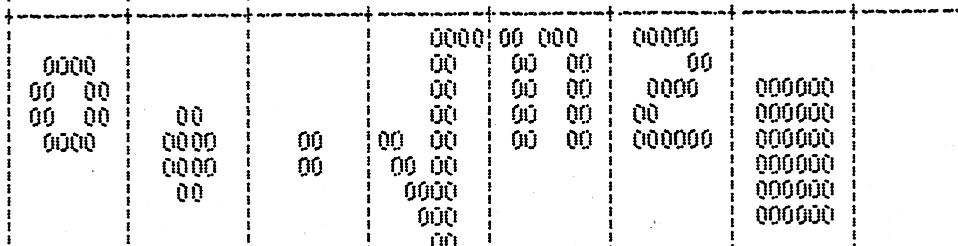
` 224 E0h	a 225 E1h	b 226 E2h	c 227 E3h	d 228 E4h	e 229 E5h	f 230 E6h	g 231 E7h



h 232 E8h	i 233 E9h	j 234 EAh	k 235 EBh	l 236 ECh	m 237 EDh	n 238 EEh	o 239 EFh



p 240 F0h	q 241 F1h	r 242 F2h	s 243 F3h	t 244 F4h	u 245 F5h	v 246 F6h	w 247 F7h



x 248 F8h	y 249 F9h	z 250 FAh	{ 251 FBh	| 252 FCh	} 253 FDh	~ 254 FEh	^ 255 FFh

SUPERFONT Character Set K - APA Graphics, normal video

character=	(`)	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
decimal =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
character=	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)	(x)	(y)	(z)	({)	(¦)	(})	(~)	(_)
decimal =	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
character=		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
decimal =	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
character=	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
decimal =	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
character=	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
decimal =	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
character=	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
decimal =	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
character=	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
decimal =	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
character=	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	(^)
decimal =	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

APA Graphics, reverse video

character=	(`)	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
decimal =	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
character=	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)	(x)	(y)	(z)	({)	(¦)	(})	(~)	(_)
decimal =	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
character=		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
decimal =	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
character=	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
decimal =	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
character=	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
decimal =	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
character=	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
decimal =	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
character=	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
decimal =	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
character=	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	(^)
decimal =	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255