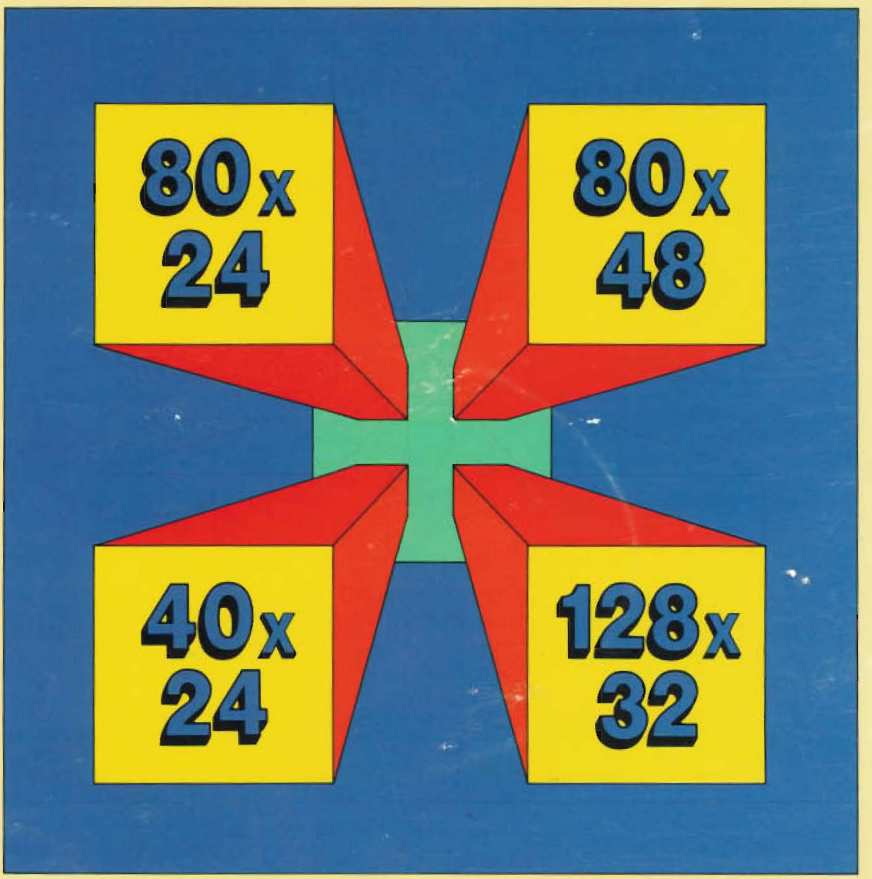
Videx INC ™

UltraTerm™ Video Display Card

Installation and Operation Manual



UltraTerm™

A Multi-Mode Video Display Peripheral

SECOND EDITION
June 1983

COPYRIGHT NOTICE

897 NW Grant Avenue
Corvallis, Oregon 97330

NOTICE:

Videx, Inc. reserves the right to make improvements or changes in the product described in this manual at any time without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

Videx, Inc. makes no warranties, neither express nor implied except as explicitly set forth in the Limited Warranty below, with respect to this manual nor with respect to the product described in this manual, its quality, performance, merchantability or fitness for any purpose. Videx, Inc. software is sold or licensed "as is." The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not Videx, Inc., its distributors, or its retailers) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Videx, Inc. be liable for direct, indirect, incidental, or consequential damages resulting from any defect of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

LIMITED WARRANTY:

Videx, Inc. warrants this product to be free from defects in material and workmanship for a period of ninety (90) days from the date of original purchase. Videx, Inc. agrees to repair or, at our option, replace any defective unit without charge. Videx, Inc. assumes no responsibility for any special or consequential damages. No other warranty, neither express nor implied, is authorized by Videx, Inc. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Table of Contents

Contents

Expanded Table of Contents

Contents

Chapter Five OPERATION

Chapter Six SOFTWARE ENVIRONMENTS

Chapter Seven SOME SPECIFIC SOFTWARE

Chapter Eight THE PROGRAMMER'S GUIDE

Contents

CHAPTER ONE

Introduction

Introduction

In this chapter we will give you a brief description of your UltraTerm and some directions on how you can best use this manual. Please take a few minutes to read this chapter, then decide which other chapters of the manual you will need to examine. A few moments spent with this manual will make the installation of your UltraTerm an easier, faster, and more satisfying procedure.

Section 1.a HOW TO USE THIS MANUAL

We have written this manual with four goals in mind. They are:

1. To provide orientation and installation instructions for all users.
2. To provide a tutorial presentation for new users.
3. To provide complete and concise reference information for advanced users.
4. To provide complete technical specifications for programmers who will incorporate UltraTerm features into their products.

You should need no other documents to completely understand and successfully use your UltraTerm. Please let us know if you find any area of our documentation inadequate.

Section 1.a.1 MANUAL ORGANIZATION

We have designed this manual to be useful to beginners, advanced users and hardware/software wizards. To help us organize our presentation more effectively, we have broken this (and all our other manuals) into five major parts:

Part I (Installation and troubleshooting) [Ch. 1-3] This section will help you learn the basic principles of the UltraTerm, install the card in your Apple][, and use the new features this card adds to your personal computer system.

Part II (Product Definition and Use) [Ch. 4-5] This section will help you to learn to use the full power of your UltraTerm. The complete command structure and all the user options are described in this section. Both beginners and advanced users will find this section useful.

Part III (The Software Interface) [Ch. 6-8] In this section we will describe the interaction of the UltraTerm with the most popular operating systems used with the Apple][. We will also describe the ways in which the card can be used with certain

specific software systems such as word processors, data communication systems and data base management programs.

Part IV (The Hardware Interface) [Ch. 9] The requirements for the video display monitor to be used with the UltraTerm are detailed in this section. The possible interactions between the UltraTerm and other peripheral cards installed in your Apple][are also examined in this section.

Part V (Technical Notes) [Appendices] We have included a complete schematic diagram, listings of the UltraTerm firmware and complete technical specifications on the video display controller circuit in this section. This section will be useful primarily to advanced users and hardware and software wizards.

We have, of course, included an Index and a Glossary to make this manual easier to use. There is also a tear-out reference card which contains a concise list of the UltraTerm commands.

Section 1.a.2 NOTATION

When we are referring to characters or keys on your Apple's keyboard, we will enclose them in single quotation marks. For example:

> **Striking the 'A' key will cause the letter 'A' to appear at the position of the cursor on the video display.**

When we refer to control characters and shifted characters, we will specify the keys which need to be pressed, separated by a hyphen, inside the single quotation marks. When we are referring to single control characters, we will use the '^{C}' to indicate the 'CTRL' key. For example:

> **Pressing 'G^{C}' while in BASIC will cause the Apple's speaker to produce a beep. The 'SHIFT-1' key will cause an exclamation mark to be displayed.**

If we are referring to a keyboard entry which does not produce a visible character, or if we refer to a character by a name or abbreviation, we will enclose the character in triangular brackets. For example:

> **Striking 'I^{C}' will generate the <TAB> character.**

Whenever we refer to character sequences using the control and shift keys, you must hold down the first key while you strike the second.

Section 1.b PRODUCT OVERVIEW

Your UltraTerm card is a sophisticated video display peripheral for the Apple][computer. It allows you to display text generated by your computer in 7 different display modes, including 160 characters by 24

lines and 80 characters by 48 lines. The default start-up mode is 80 characters by 24 lines—which emulates our Videoterm display. Thus, you can use all the special software packages which have been developed for the 80-column Videoterm over the last three years.

Section 1.b.1 THE PRINTED CIRCUIT BOARD

The UltraTerm system consists of a printed circuit (PC) board and a cable to connect the card to your display monitor. The functional areas of the PC board are shown in figure 1 1 and described in the following paragraphs.

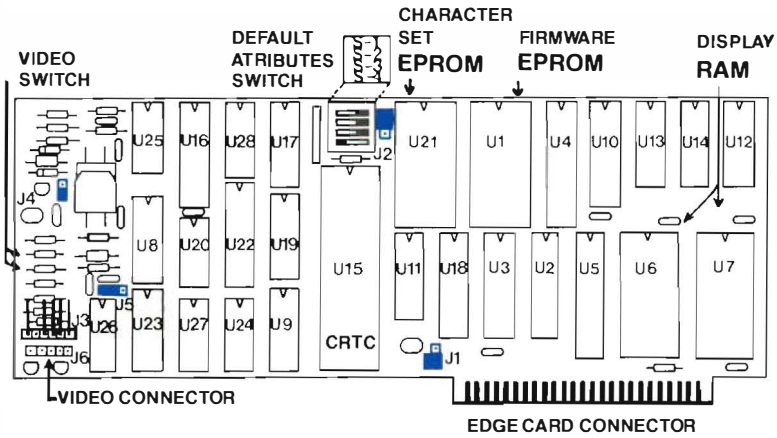


FIGURE 1.1 Block Diagram of the UltraTerm

Video Connectors and Video Switch These connectors handle the input from the Apple's 40-column or graphic output and the 80 or 132-column output from the UltraTerm. The video switching circuitry allows software selection of the signal to be sent to the video display monitor.

Edge-Card Connector These gold-plated fingers plug into one of the expansion slots at the back of your Apple. The connector allows transfer of the signals necessary to control the UltraTerm between your Apple and the UltraTerm circuitry.

Firmware EPROM This is a Read-Only-Memory chip which contains a program that is used by your Apple to control the operation of the UltraTerm card.

Display RAM These memory circuits contain 4096 bytes of random-access memory that are used to store the characters which will appear on the video monitor when the UltraTerm is in use.

CRTC This integrated circuit is the heart of your UltraTerm. CRTC is an acronym for Cathode Ray Tube Controller. (The display on your video monitor is generated on the face of a cathode ray tube. A TV picture tube is also a cathode ray tube.) The CRTC is actually a specially-programmed

microprocessor which continuously converts the character information in the display RAM to video signals to drive your display monitor.

Character set EPROM This integrated circuit contains the information which the CRTC uses to generate a matrix of dots on your display which represent a particular character in the display RAM. The EPROM normally provided with the UltraTerm contains a standard character set and a high-quality character set.

Section 1.b.2 VIDEO SIGNAL CONNECTION

Video signals are routed to the UltraTerm card through the video signal cable as shown in figure 1.2. The video cable has three connectors to allow you to connect your display monitor, the UltraTerm card, and the normal Apple][video signals. At one end of this cable is a male RCA phono plug. This plug is inserted in the video output jack on the back of your Apple. In the middle of the cable is the keyed 5-connector socket which is connected to the 5-pin connector on the UltraTerm card. On the other end of the cable is a female RCA phono socket into which you must plug the male connector on your monitor cable.

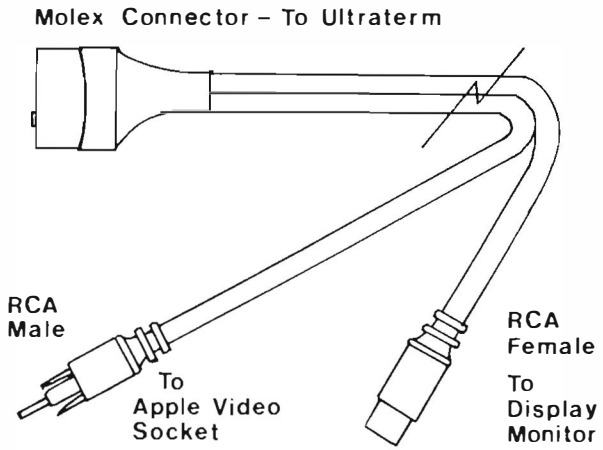


FIGURE 1.2 Video Connector Cable

Section 1.b.3 SOFTWARE FEATURES

The firmware EPROM on your UltraTerm provides your Apple][with an operating program having the following features:

- Compatibility with these Operating Systems:

  Apple DOS
  Apple Pascal and the SoftTech P-System
  CPM (When using a Z-80 processor card)

- Applesoft BASIC and Integer BASIC are supported.
- Many Word-Processing systems are supported.
- Upper and lower-case characters may be entered from an unmodified keyboard, using 'A^{C}' to change cases.
- Operating commands may be generated by program control.
- A standard set of escape sequences and control characters can be used for cursor movement and display editing.
- Escape sequences can be used to change video display modes from the keyboard in BASIC.
- Display output can be halted and resumed under keyboard control.

Section 1.b.4 HARDWARE FEATURES

The state-of-the-art design of the UltraTerm circuitry provides you with many features not found on other video display cards:

- 24-line by 80-column display.
- 24-line by 96-column display.
- 32-line by 128-column display.
- 24-line by 132-column display.
- 48-line by 80-column display—great for editing and word-processing users.
- Several more modes are available with a high-quality character set using interlaced display mode.
- Emulation of Videoterm 24-line by 80-column operation.
- Character-by-character selection of one of two sets of special character attributes. These attributes may include combinations of the following:

  Normal/High-resolution character set
  Normal/Inverse video
  Highlight/lowlight characters.
- Stable, flicker-free display with fast hardware scrolling.
- Display of all 96 ASCII characters with true descenders on lower-case characters.
- 15-character line-drawing set as part of standard character set.
- 7-character block graphics font as part of standard character set.
- Highly readable 7 by 9 dot character font.
- Operates in any peripheral slot except slot 0 without modification.
- Complete theory of operation is provided in this manual.
- Can be used in Apple //e with extended memory card.

Section 1.c HARDWARE REQUIREMENTS

There are two fundamental hardware requirements for successful operation of your UltraTerm card. First of all, you must plug the card into an Apple][, Apple][Plus, Apple //e or Apple /// computer. Secondly, the video output signal must be routed to a high-resolution video display monitor.

WARNING

Since the UltraTerm in the 132-column mode will send dot information to the display almost 60% faster than an 80-column device, not all display monitors are suitable. See chapter 9 for a more complete discussion of this subject.

Your display monitor should have a bandwidth of at least 15 megaHertz to provide a sharp display in the 132-column mode. We have tested the following display monitors and found them to be suitable when properly adjusted:

- Apple Monitor /// (Our recommendation for the UltraTerm)
- NEC JB-902M (Has some flicker in interlace mode)
- NEC JB-1201M (Larger display, but still flickers)
- AMDEK 300A (Amber screen, works with all modes)

Section 1.d PRODUCT REGISTRATION

We have attached a Product Registration Form at the front of this manual. We would like you to fill out this form and mail it to us. This information is NOT necessary to validate the warranty on your UltraTerm, but it will help us to better understand the needs and background of our customers.

Please answer all questions as completely as you can. If you can't answer a question, leave the space blank. If you have any additional comments, please use the comment card at the back of the manual or write us a letter. These comments will be routed directly to our technical staff. Comments on the back of the product registration form are sometimes overlooked.

CHAPTER TWO

Installation and Checkout

Installation and Checkout

Section 2.a COMPLETE INSTALLATION INSTRUCTIONS

Your UltraTerm card will function properly in any of your Apple's expansion slots except slot Ø. However, many operating systems, such as Apple Pascal, make certain assumptions about the use of the expansion slots. These assumptions are:

Slot	Device
Ø	Language Card (not used in //e)
1	Printer Interface
2	MODEM
3	External Console (UltraTerm)
4	Additional Disk Drives
5	Additional Disk Drives
6	First Disk Drive controller
7	Rigid disk controller or other interface

Since the Pascal operating system will treat your UltraTerm as an external console, it must be plugged into slot #3 to work properly with that operating system. In deference to the users of the Pascal system, we will present the rest of these installation instructions as if the UltraTerm card is installed in slot #3. If you are going to use a different slot for your card, you should have no problem with the installation and testing—just remember to change the slot number when it is referred to in the instructions and test programs.

Section 2.a.1 PRE-INSTALLATION CHECKOUT

We carefully inspect and test each UltraTerm card before shipping it to your dealer. Each card is carefully packed to prevent damage during shipping. In spite of these precautions, you should inspect your card to be sure that no obvious damage has occurred in transit.

- UltraTerm card in good condition—no missing or damaged components.
- Apple][computer is operating properly. If your computer is at all 'flaky' or will not run Applesoft BASIC properly, it will be very difficult to verify proper operation of your UltraTerm card.

Section 2.a.2 BOARD INSTALLATION

- Turn off your Apple][.
- Disconnect the power cord from the back of the computer.
- If you are using an extended memory card in the Auxiliary slot of an Apple //e, you must install the blue jumper plug over the upper two pins of J-1. The locations of the jumper pins are shown in figure Y.1
- Remove the lid from your Apple. Do this by placing the heels of your hands on the back corners of the case and pulling straight up on the back edge of the lid with your fingers until it pops loose. Lift the back edge about one inch then slide the lid to the rear until the front edge is clear of the case. You can then lift the lid free and set it aside. (If you lift the rear edge too far, the front edge of the lid will hit and possibly damage components attached to your keyboard.)
- Locate expansion slot #3. With the keyboard nearest to you, the slots are numbered from 0 to 7 with slot #0 at the left, next to the metal case of the power supply. (If you have an Apple //e, there is no slot #0.) There are slot numbers printed on the main PC board between the slots and the back edge of the board.
- Press the UltraTerm card straight down into the expansion slot connector. When it is properly seated, the top of the card will be level and parallel to your Apple's main circuit board.

Section 2.a.3 VIDEO CABLE CONNECTION

- Locate the end of the cable that came with your UltraTerm card which has a male RCA Phono plug. Insert this plug into the video output socket on the back of your Apple][.
- Lead the cable into your computer through one of the slots in the back panel. If you have an Apple //e, you will have to remove the plastic cover from one of the openings in the back panel. Plug the molex connector in the middle of the cable onto the pins on the UltraTerm card. The connector is keyed so that it cannot be connected backwards.
- Lead the remaining end of the cable out of the computer. This end has a female RCA phono jack. Plug the male RCA connector from your video display monitor into this connector. If the cable to your display monitor doesn't have a male RCA phono plug, you will have to purchase an adapter from your local computer store.

Section 2.a.4 FINAL INSTRUCTIONS

- Put the cover back on your Apple][. When you insert the front edge of the cover under the top of the keyboard, be careful not to disturb the keyboard connectors. Press firmly on the back edges of the cover and it will snap into place.
- Re-connect the power cable to your computer.

Section 2.b SHORT FORM INSTALLATION CHECKLIST

- Turn off power and remove cover from Apple][.
- Install jumper over upper two pins of J1 on the UltraTerm if you have an Apple //e with an Apple 80-column card or extended memory card.
- Install UltraTerm card in Slot #3.
- Plug male RCA connector into Apple Video output.
- Attach Molex connector to UltraTerm connector.
- Connect display monitor to female RCA socket.
- Replace Apple Cover and connect power cord.

Section 2.c CHECKOUT

This section contains some simple tests which will help you make sure that your UltraTerm card is working properly. The test programs are written in Applesoft BASIC. If you have an Apple][with Integer BASIC in ROM, you will have to load and run Applesoft.

If you must load Applesoft from disk:

- Put your DOS System Master Disk in your boot drive.
- Turn on your Apple. Your disk drive should start and its 'IN USE' light should come on. The Hello program on the system master disk should load Applesoft into RAM for you. If your disk drive does not come on, immediately turn off the power to your Apple and refer to Chapter 3.
- Type the 'FP' command to switch to Applesoft BASIC.

If you have an Apple][Plus:

- Open all disk drive doors. This will make sure that no auto-start files will be loaded.
- Turn on your Apple. The 'IN USE' light on your boot drive should come on and stay on. If this does not happen and the power light on your keyboard does not come on, immediately turn off the power switch and refer to Chapter 3.
- Press <CTRL-RESET>.

For all users:

- Your video display should show the Applesoft prompt (']') followed by a flashing cursor block.
- Enter 'PR#3' followed by <RETURN>. You are now in the 80-column mode of the UltraTerm card. The 'PR#3' should be gone and you should see only the Applesoft prompt and a flashing cursor.
- Type some random lines of text and look at the display on your monitor. If the characters do not appear as you type them, your display monitor may not be adjusted properly, or your video cables may be improperly connected. If you have a problem, do the next test before going to Chapter 3.

- Press 'G^{C}'. You should hear a short beep. This beep should be a little lower in pitch than the beep Applesoft uses to tell you that you have made a mistake. If you hear this beep, it means that the firmware EPROM on your UltraTerm card is working properly. Now you can go to Chapter 3 if your display is not working properly.
- Enter the following program. End each line with a <RETURN>.

```
10 HOME: FOR N=1 TO 100
20 PRINT "The quick brown fox jumps over lazy dogs.";
30 NEXT N
40 PRINT "PRESS ANY KEY TO CONTINUE...";
50 GET A$: PRINT CHR$(22);"0": HGR2:HCOLOR=3
60 HPLOT 0,0 TO 279,191
70 GET A$: PR#3: PRINT:HOME
80 LIST: END
```

- Type 'RUN' <RETURN>. Your video display should look like figure 2.1. We picked this display to allow you to adjust your video display monitor if necessary. A discussion of display monitors and their adjustment is included in Chapters three and nine.
- Press the space bar on your keyboard. You should see a diagonal line from the upper left-hand to the lower right-hand corner of your display monitor. This test checks the video switch which automatically selects the HI-RES graphics or text display.
- Press the space bar again. You should see the 80-column text screen. The program should be finished and the Applesoft prompt followed by a flashing cursor should be at the bottom of the screen.
- Type <ESC>'3' to set the 160-column by 24-line mode. RUN the program again. If the characters on the left hand margin are off the edge of the screen, your monitor has too much overscan to use this mode. See chapter 9.
- When the program ends, type <ESC>'6' to set the 80-column by 48-line mode. RUN the checkout program (the screen will be only half full). If the display appears to shimmer, your monitor has a low-persistence phosphor. You will see this shimmer whenever you use the modes which utilize the interlace mode with the high-quality character set.
- When the program ends, type <ESC>'1' to return to the 80 × 24 mode.
- Your initial check out is complete. You may proceed to Chapter Four for more complete operating instructions.

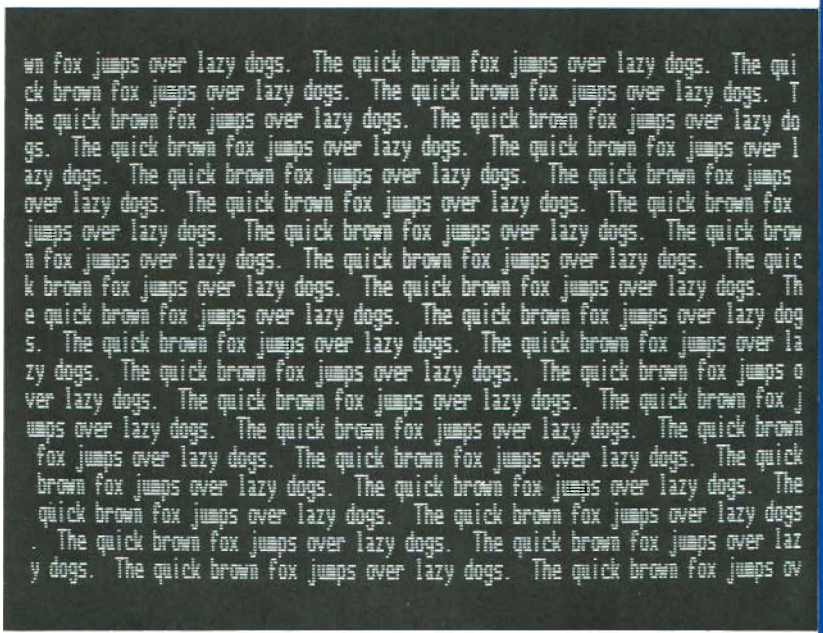


Figure 2.1 Sample Screen Produced By Test Program

CHAPTER THREE

Troubleshooting

Chapter 3

Our many thousands of hours of troubleshooting experience with the Videoterm card have shown us that most problems are a result of easily-corrected installation errors, video monitor adjustments, or misinterpretation of our operating instructions. In this chapter we will help you diagnose problems with your UltraTerm. If you can trace the problem to installation, video connections or monitor adjustments, we will show you how to get your UltraTerm working. If you find that your UltraTerm is not working properly, we suggest you get help from your dealer. Your UltraTerm uses a multi-layer printed circuit board which allows us to put a lot of circuitry in a very small space. This makes servicing the board a task which should be undertaken only by a qualified technician.

If you have a problem that you cannot diagnose with help from this chapter, please feel free to call one of our service technicians at (5Ø3) 758-Ø521. They are available to help you from 8AM to Noon and from 1PM to 4:3ØPM (Pacific time) Monday through Friday (they do get holidays off).

We will start with the assumption that your Apple and video monitor worked well together before you installed your UltraTerm. If you are not sure of this, turn off your Apple, remove your UltraTerm and check out your Apple and video monitor. If they work properly, turn off your Apple and install your UltraTerm (following the instructions in Chapter two).

The next item to check is the internal video switch on your UltraTerm. Do you get a 4Ø-column Apple video display when you first turn on your Apple? If you do, the UltraTerm video switch is properly selecting the Apple video signal. This means your monitor, cables and connections are working properly.

Next, use the 'PR#3'
If you now get a prompt sign, and the video display works properly, your second try at installation has solved your missing video problem. If your screen goes blank when you use the 'PR#3'
is working. Type a few 'G^{C}' keys. You should hear a beep, about an octave lower than the Applesoft beep, from the Apple speaker. If you hear the beep, the firmware on your UltraTerm card is working. If there is no beep, the firmware on your UltraTerm is not executing its code properly. In either case, it's time to check with your dealer or call our service technicians.

Section 3.b UNACCEPTABLE VIDEO DISPLAY

If the display quality of your UltraTerm is not what you expect it to be, there are several areas you can check. The two most common problems are: 1) choosing a display mode unsuited to your monitor; and 2) display monitor improperly adjusted.

Section 3.b.1 DISPLAY MODE AND MONITOR MISMATCHED

Chapter Nine will tell you in detail which monitors are suitable for the different video modes you can use with your UltraTerm. If you are using a display mode which is not suited to your monitor, you will probably experience one of the two following problems:

1. Some characters are missing on the edges of the screen. You will probably experience this problem if you use a display such as the Apple Monitor /// in the 160-character mode. This mode uses more of the horizontal scan time than is displayed by the monitor. As a result, some of the characters are displayed before the CRT electron beam reaches the left edge of the screen, and others are displayed after the beam leaves the right edge. If you have a monitor with a width adjustment, you can shrink the width of the display until all the characters are visible. You will also experience this problem when you use the 96-character display mode with the Monitor ///. Since the Monitor /// and many other displays do not have an external width adjustment, we suggest you use the 80, 128 or 132-column modes with these displays.
2. If your display appears to shimmer or flicker when you select a display mode which uses the high-density character set or displays more than 24 lines, you probably have a monitor which does not have a long-persistence phosphor. In the interlace mode, your UltraTerm writes each scan line only one half as often as it does in the non-interlace mode. As a result, if your monitor does not retain the bright dots on the screen until the next scan, the display appears to flicker or shimmer. If you feel you must use an interlace mode, you can minimize the shimmer by careful adjustment of the contrast and brightness controls. In the interest of avoiding eyestrain, we suggest you use the non-interlace modes unless you have a monitor with a long-persistence phosphor.

Section 3.b.2 DISPLAY MONITOR OUT OF ADJUSTMENT

If your display monitor is out of adjustment, may want to try adjusting its controls to improve the display with your UltraTerm. You will generally find that if your monitor is adjusted for the best display with the Apple video

signal, you will not need to make any large adjustments for the best display with your UltraTerm. The following adjustments may improve the display when you use your UltraTerm:

1. Adjust brightness and contrast to provide adequate character brightness with a completely black background. The brightness level depends on the highlight/lowlight mode. On some displays, the brightness level may vary depending on the number of characters on the screen. Try filling the screen with characters and adjusting the brightness for the best display.
2. Adjust the horizontal and vertical hold controls until your UltraTerm is stable and doesn't show any tearing of the first characters at the top of the screen. On most monitors, you should be able to adjust the vertical hold to make the display roll both upwards and downwards. Adjust this control to a point midway between the upward and downward rolls. If your display continues to roll in spite of your adjustments, switch to the Apple video signal. If your display still rolls, there may be a problem with your internal video switch. Make sure that the problem disappears when you connect your monitor directly to the output jack on the back of your Apple, then visit your dealer or call our service technicians.
3. Characters which are of uneven height from the top to the bottom of the display can be corrected by adjusting the vertical linearity control. Unfortunately, on many displays such as the Monitor ///, this control is inside the cabinet. Since there are high voltages present inside the cabinets of video display monitors, internal adjustments should be carried out only by qualified service technicians.
4. Fuzzy or indistinct characters can have two possible causes. First, your monitor may have too little bandwidth to display the number of characters your UltraTerm can produce. You should have a monitor with a bandwidth of at least 15 MHz. This is particularly important if you are using the 128, 132, or 160-character modes. If your monitor has adequate bandwidth, but the display is still fuzzy, the electron beam may not be properly focused on the screen. Some monitors have an external adjustment which you may set for the best display. Other monitors, such as the Monitor / //, do not have an external focus control. You should take the monitor to your dealer for adjustment unless it is out of warranty and you are comfortable working with exposed high-voltage circuits.

NOTE: Moving Jumper J-4 may improve the display on some monitors (see Appendix Y.8).

Section 3.c WARRANTY AND NON-WARRANTY REPAIR

When you have a problem with a Videx product, your first step should be to contact your dealer. If the dealer is unable to solve the problem, please give our service department a call at (503) 758-0521. Our service technicians can often diagnose the problem and send you a part which will repair your board. In this way, they can often save you the time and expense of sending your board in for repair.

Before you call us, please prepare a brief summary of your problem. If you can, please have your computer nearby and running. Our service technicians may be able to suggest tests which can diagnose your problem more completely.

If you must return your UltraTerm to us for service, the service technician will give you an RMA number. (RMA stands for Return Merchandise Authorization). You must have an RMA number for any merchandise you send us—whether it is still in warranty or not! You should clearly mark the RMA number on the outside of your package, as well as on a brief note included with the defective board. We use the RMA number and our in-house computer to monitor the progress of your board through our service department and to ensure the fastest possible turnaround time.

We have included a tear-out RMA form at the back of this manual for your convenience. Please fill out this form and include it with your UltraTerm when you return it to us. This form will help you to be sure you don't forget any vital information—like your return address!

CHAPTER FOUR

The Beginner's Guide

The UltraTerm firmware also checks the output characters which are printed by programs for special control characters. For instance, when a program sends the <CR> or Carriage Return character, the UltraTerm moves the cursor to the beginning of the next line. The <CR> does not normally appear on the screen—even though the card can display control characters, they are normally swallowed by the firmware.

The important thing to remember is that there are two ways to send commands to the UltraTerm card:

1. By entering special command characters from the keyboard. (Input Commands)
2. By having a program print command characters as part of the program output. (Output Commands)

You should note that certain characters are used as both input and output commands. Some operating systems may send all keyboard input directly to the output device. This is known as 'keyboard echo.' When you use a system like this, a command which would normally work only as an output command may also work as an input command. In addition, some operating systems or peripheral card firmware may intercept commands from the keyboard and change them before sending them to the output device. This is done by the UltraTerm firmware. When you type the <K[C]> key, the UltraTerm will intercept the character and change it to a left bracket '[' before sending it to the screen. Other systems and application programs may make their own special changes to the input characters before sending them to the output device. Since we don't know what special programs you may be using, we can't always guarantee that a certain key will always produce the same result. We will tell you what will happen if you use the UltraTerm commands in the most common operating systems in the next chapter. In Chapter 7, we will tell you what to expect if you use some of the more popular applications programs and word processing systems.

Section 4.b.1 INPUT COMMANDS

You can send your UltraTerm an input command any time that you are typing characters at the keyboard. The UltraTerm will execute the command, but may not pass the command characters on to the program that was waiting for input. Since the program may not know that you have entered the command, you should be cautious about modifying the screen display. Otherwise, you may erase some necessary information without telling the program what you have done! If you have a program that must receive one of the UltraTerm input commands, you will have to use a special input routine which bypasses the UltraTerm firmware.

Please note that many input commands function only with BASIC and DOS. Many of these commands cannot be used with Pascal and CP/M.

Section 4.b.2 OUTPUT COMMANDS

Your software can send commands to the UltraTerm at any time when it is sending characters to the screen. The command will be executed, but the command characters may not appear on the screen. Some programs (such as Apple-Writer][when modified with our Pre-Boot disk) can cause command characters to appear on the screen. For example, the Apple Writer][program can display all the Carriage Return characters in a block of text by using the CR abbreviation character which is part of the UltraTerm character set. The program does this by storing the abbreviation character directly into the UltraTerm display RAM and bypassing the card's firmware. The firmware will normally execute the command and swallow the command character.

Section 4.b.3 OPERATING SYSTEM COMMANDS

The firmware on your UltraTerm card recognizes certain operating system commands which change the video display without sending control characters to the card. For example, the HOME, VTAB and INVERSE commands of Applesoft BASIC will work properly even though they do not cause any characters to be sent to the UltraTerm.

Section 4.b.4 AUTOMATIC VIDEO SWITCHING

Your UltraTerm contains an electronic switch that will automatically select the proper video signal when you change display modes. When you turn on or reset your computer, this switch sends the Apple's 40-column video signal to your display monitor. The 'PR#3' command causes the electronic switch to send the output of your UltraTerm to the display. We will show you a simple software command you can use to switch to the Apple video signal so that you can view the graphics display. The 'PR#3' command will switch you back to the video mode you were using before you switched to the graphics mode. The equivalent commands in other operating systems such as Pascal will also control the electronic switch on the UltraTerm in a similar manner.

Section 4.c A SUMMARY OF THE CHAPTER

In this chapter we introduced you to the following features of your UltraTerm:

- You can use your UltraTerm without having to learn any new commands or operating methods.
- You can configure your operating system to automatically use your UltraTerm, or you can configure it by hand at any time.

- The UltraTerm responds to command characters typed at the keyboard (input commands).
- The UltraTerm responds to command characters printed by programs (output commands).
- Command characters are normally 'swallowed' by the UltraTerm.
- Certain operating systems commands such as HOME will be acted upon by the UltraTerm.
- The electronic switch on the UltraTerm will automatically select the Apple video signal for 40-column output or graphics output.

The Beginner's Guide

This chapter will provide a simple set of instructions to help you get started with your UltraTerm. We also describe the way your UltraTerm works with the most-used operating systems for the Apple.

Section 4.a WHEN YOU FIRST TURN THE POWER ON

Since your Apple][or Apple][Plus can normally display only 40 columns of text, most of the programs and operating systems for your computer are written to use only a 40-column display. When you first turn the power on, your operating system (Apple DOS, Pascal or CPM) must be informed that you want to use the UltraTerm card. When you tell the operating system that you want to use a special feature, we say you are 'configuring' the system. There are two ways to configure an operating system or program:

1. By hand—each time you start up your computer or run a new program, you type in a command that tells the system to use the UltraTerm card. This is what we did when we used the 'PR#3' command during checkout.
2. Automatically—you run a utility program just once which writes a special data file or modifies some software on your boot disk. The new information on the disk will automatically select the UltraTerm card when you boot the disk. This method is used by the Pascal and CPM operating systems.

Chapter 7 contains complete instructions for configuring the most common operating systems for the Apple. Chapter 7 will help you use your UltraTerm with some specific programs which can use the card. Most of the examples in the next chapter will use Applesoft BASIC and Apple DOS.

Section 4.b NORMAL USAGE

In normal usage, your UltraTerm can be used just like your standard Apple video display. Your card does have many additional features for which you will have to learn new commands. However, we will save these for later.

Many new users think of the UltraTerm card as a display device only. Actually, the firmware program on your card works with both the keyboard input and the video display. When you are entering characters from the keyboard, the UltraTerm firmware checks for special command characters. If command characters are found, the UltraTerm executes the proper command, then 'swallows' the command sequence. Thus, when you enter 'A^C', the UltraTerm toggles the upper-lower case mode to allow you to enter lower-case characters. The 'A^C' is not sent to the screen or the program which is waiting for input.

The UltraTerm firmware also checks the output characters which are printed by programs for special control characters. For instance, when a program sends the <CR> or Carriage Return character, the UltraTerm moves the cursor to the beginning of the next line. The <CR> does not normally appear on the screen—even though the card can display control characters, they are normally swallowed by the firmware.

The important thing to remember is that there are two ways to send commands to the UltraTerm card:

1. By entering special command characters from the keyboard. (Input Commands)
2. By having a program print command characters as part of the program output. (Output Commands)

You should note that certain characters are used as both input and output commands. Some operating systems may send all keyboard input directly to the output device. This is known as 'keyboard echo.' When you use a system like this, a command which would normally work only as an output command may also work as an input command. In addition, some operating systems or peripheral card firmware may intercept commands from the keyboard and change them before sending them to the output device. This is done by the UltraTerm firmware. When you type the <K[C]> key, the UltraTerm will intercept the character and change it to a left bracket '[' before sending it to the screen. Other systems and application programs may make their own special changes to the input characters before sending them to the output device. Since we don't know what special programs you may be using, we can't always guarantee that a certain key will always produce the same result. We will tell you what will happen if you use the UltraTerm commands in the most common operating systems in the next chapter. In Chapter 7, we will tell you what to expect if you use some of the more popular applications programs and word processing systems.

Section 4.b.1 INPUT COMMANDS

You can send your UltraTerm an input command any time that you are typing characters at the keyboard. The UltraTerm will execute the command, but may not pass the command characters on to the program that was waiting for input. Since the program may not know that you have entered the command, you should be cautious about modifying the screen display. Otherwise, you may erase some necessary information without telling the program what you have done! If you have a program that must receive one of the UltraTerm input commands, you will have to use a special input routine which bypasses the UltraTerm firmware.

Please note that many input commands function only with BASIC and DOS. Many of these commands cannot be used with Pascal and CP/M.

CHAPTER FIVE

Operation

Chapter 5

In this chapter we will describe the operation of your UltraTerm and show you how it will respond to commands sent to it. The chapter is divided into sections covering input commands, output commands and special operating-system commands. We will also show you how to set the DIP switches which select the attribute sets your UltraTerm will use when you first turn on your computer.

Input commands are commands which you send to your UltraTerm from the keyboard. The UltraTerm will not pass on the characters in the command to your program. The command is executed by the UltraTerm firmware and the command characters are 'swallowed.' All other characters are passed on to your program as usual. Here are the input commands your UltraTerm will accept:

A^{C}: Uppercase/lowercase toggle. This command will switch you from upper case input to lower case input or vice-versa. Only the A through Z keys on your keyboard are affected. (If you have a Keyboard Enhancer or Enhancer][, you do not need this command and it will work properly only if your Apple is in the standard Apple keyboard mode. See your Enhancer manual for details.) This command is not needed or available on the Apple //e.

K^{C}: This command will generate the '[' character. This character cannot be generated on a standard Apple keyboard without special software (like that in the UltraTerm firmware). This command is also not available if you have an Apple //e. The '[' is available on the keyboard of the Apple //e.

S^{C}: This is the pause command. This command causes output to the UltraTerm card to halt so that you can examine the display. Another 'S^{C}' or any other character will cause the UltraTerm to continue.

The firmware on your UltraTerm also allows you to type BASIC programs using lower-case letters (if you have an Apple //e or an Apple][with an Enhancer][). The firmware automatically translates all the input characters to upper case unless they are enclosed in quotation marks. When you LIST the program you will see the BASIC commands in upper-case letters, but strings enclosed in quotation marks will remain exactly as you typed them.

Section 5.b OUTPUT COMMANDS

An output command is a special character or characters which is sent to the UltraTerm. Instead of displaying the command, the UltraTerm will recognize the command and take some special action. The command characters are swallowed by the UltraTerm. The commands may be sent to the UltraTerm by your program or by the operating system. If your operating system (like Apple DOS) echoes input characters to the screen, you can perform some of these commands from the keyboard as well. Some of the commands contain special characters which cannot be generated on a standard Apple keyboard. You will need to use a CHR$(X) function from BASIC or the CHR(N) function in Pascal to generate the commands. Here are the output commands to which your UltraTerm will respond:

G^C: This is the ASCII bell character. It will cause a short beep from the Apple's speaker. The beep produced by the UltraTerm will be a little lower in pitch than the beep produced by the Apple alone.

H^C: This command will move the cursor back one space. It is also generated by the left-arrow key. When you enter this command from the keyboard, the operating system usually deletes the character preceding the cursor from the input buffer.

J^C: This command character is the ASCII Line Feed. It will move the cursor down one line. If the cursor is already at the bottom of the screen, the whole screen will move up one line, and the cursor will stay on the bottom line.

K^C: This command will clear the display from the cursor position to the end of the screen. The character under the cursor will disappear, but the cursor will not move.

L^C: Sending this character to the UltraTerm will clear the whole screen and move the cursor to the uper left-hand corner. This is the ASCII Form Feed character.

M^C: This character, the Carriage Return, will move the cursor to the beginning of the current display line. If it is sent from BASIC, a line feed will also be sent.

N^C: This command selects the standard attribute set for display. All characters sent after this command will be displayed at with the default attributes (normal video and lowlight intensity, unless you have changed the attributes). This command does not function in DOS and BASIC and you should use the 'NORMAL' command instead.

O^C: This is the alternate attributes command. It will select the alternate display attributes for all characters sent after the command. The alternate attributes normally display inverse video. This command does not function in DOS or BASIC and you should use the 'INVERSE' command instead.

R^C: We call this the Raw Mode command. It is used to disable most of the special commands of the UltraTerm. After you send this command, the UltraTerm will respond only to G^C, H^C, J^C, and M^C. This command can be cancelled only with the 'PR#3' command.

U^C: This command sets the Apple 40-column mode. The video switch is set to select the Apple video signal.

V^C: This command sets the video format for your UltraTerm card. The V^C character is followed by a single digit between 0 and 8 which determines the command as follows:

0 Set the Apple 40-column mode. The video switch is set to select the Apple video signal.

1 Set the 80 × 24 video mode. This is the mode which emulates the earlier Videoterm cards. In this and the following modes, the video signal from the UltraTerm card is selected.

2 Set the 96-column by 24-line display mode. This mode won't show all the characters on a Monitor ///, but will work with the NEC JB-1201 monitor.

3 Set the 160-column by 24-line mode. This mode won't show all the characters on a Monitor ///, but will work with the NEC monitor.

4 This command sets an 80-column by 24-line mode with the High-Quality interlaced character set. This and the next four modes will show some flicker unless your display monitor has a long-persistence display tube like that on the Apple Monitor ///. With interlace on, the vertical elements of your characters will more completely connected.

5 Set the 80 by 32 mode with interlace operation.

6 Set the 80 by 48 mode with interlace on. (This mode does not use the high-quality character set.)

7 Set the 132 by 24 mode with interlace on.

8 Set the 128 by 32 mode with interlace on.

W^C: This is the lead-in character for the command to set the video attributes. The W^C is followed by two digits, each of which may range from 0 to 7. The first digit sets the attributes which will be used when the high bit of the character in the display RAM is clear. The second character sets the attributes to be used when the high bit of the byte in the character RAM is set. See Section 8.d.3 for a discussion of video attribute programming and the display characteristics for each of the digits.

Nibble Value	Resulting Display Characteristics		
7	**Alternate char. set**	**inverse video**	**highlight**
6	**Alternate char. set**	**inverse video**	**lowlight**
5	**Alternate char. set**	**normal video**	**highlight**
4	**Alternate char. set**	**normal video**	**lowlight**
3	**Standard char. set**	**inverse video**	**highlight**
2	**Standard char. set**	**inverse video**	**lowlight**
1	**Standard char. set**	**normal video**	**highlight**
0	**Standard char. set**	**normal video**	**lowlight**

Note: These nibble values are used with the W^C command to set the display attributes.

Y^C: This command will move the cursor to the upper left-hand corner of the display. The display will not be cleared.

Z^C: This is the lead-in for the Control-Character Display Command. It is followed by a single character to select the command. The results produced by different command characters are as follows:

@^C, A^C to G^C Display the appropriate block graphic character.

H^C to P^C Display symbols for ASCII control codes if using standard character set. If the high-quality character set is enabled, additional block graphics characters are displayed.

Q^C to _^C Display the appropriate line drawing character.

Z^C1 This command switches you to the Apple 40-column video display. It is included for compatability with the Videoterm.

The following commands cannot be entered from the standard Apple keyboard. Each command is followed by the appropriate CHR$(N) command as you would use it in a BASIC program.

\^C(CHR$(28)): This command will move the cursor forward one space.

]^C(CHR$(29)): This is the Clear to End of Line command. All the characters from the cursor to the end of the current line will be cleared, including the one under the cursor.

^^C(CHR$(30)): This is the GOTOXY lead-in command. The two characters following the lead-in will determine the new position of the cursor. The first following character will determine the new horizontal position. The second will specify the vertical position. The position specifiers are offset by 31, so the sequence CHR$(30), CHR$(64), CHR$(48) would move the cursor to column 33 of row 17. The value of x may range from 32 to 112, and the value of y may range from 32 to 56. To move to a location determined by variables X and Y,

you would use:

PRINT CHR$(30);CHR$(X+31);CHR$(Y+31);

_^{C}(CHR$(31)): This is the reverse line feed command. It will cause the cursor to move up one line. If the cursor is at the top of the screen, nothing will change.

The following short demonstration program will display the complete character set on the screen. It also demonstrates the use of the 'Z^{C}' output command to display the line-drawing and block graphics characters.

```
10 HOME  J = 1
20 FOR I = 0 TO 127
30 IF I < 32 THEN PRINT CHR$(26); :REM CONTROL-Z
40 POKE 36, J * 6: PRINT CHR$(I);"-";I;
50 J = J + 1. IF J > 11 THEN J = 1. PRINT: PRINT
60 NEXT I
```

These commands are different from input or output commands in that they may not actually send characters to the UltraTerm. Or, they may require some additional action from the operating system to operate as expected. These commands function properly only in the BASIC or DOS operating systems.

U^{C}: We call this command the Copy-forward. It will cause the cursor to move one space to the right. In addition, the character which was under the cursor before the move will be picked up from the screen and sent to the computer as if it had been typed on the keyboard.

<ESC>: This is the lead-in command for the screen editing mode. The command is followed by one or more characters which determine the editing command. The valid editing commands are:

0	Set Apple 40-column display
1	Set 80 × 24 display
2	Set 96 × 24 display
3	Set 160 × 24 display
4	Set 80 × 24 display with interlace
5	Set 80 × 32 with interlace
6	Set 80 × 48 with interlace
7	Set 132 × 24 with interlace
8	Set 128 × 32 with interlace
@	Clear the screen.
A	Cursor Right

B Cursor Left
C Cursor Down
D Cursor Up
E Clear From Cursor to End of Line
F Clear From Cursor to End of Screen
I Cursor Up
J Cursor Left
K Cursor Right
M Cursor Down

The I, J, K, and M command characters may be repeated without entering another <ESC> for multiple cursor moves. The command will end with the first character which is not an I, J, K, or M. The cursor move keys are slightly different on the Apple //e, which has up and down arrow keys. These keys will not move the UltraTerm cursor.

HOME This command is available only in Applesoft BASIC. It clears the 40-column screen. The UltraTerm firmware is able to detect this command and will also clear the UltraTerm display and move the cursor to the upper left-hand corner of the screen.

INVERSE The UltraTerm will display all following characters in inverse video when this command is used in BASIC.

NORMAL All following characters will be displayed in normal (white on black) video. (Please note that INVERSE and NORMAL function in this manner only if you have not changed the Video Attributes Register. If you change the register, you can alter or even disable these commands.)

HTAB This command will work properly only if you HTAB to a column between 1 and 40. HTABs past column 40 are not supported. We suggest you use the POKE commands described in Chapter 8.

VTAB The VTAB command will work just as it does in BASIC, except that you cannot VTAB lower than line 24. The cursor will move to the line whose number follows the command. The horizontal position of the cursor will remain the same.

FLASH This command will produce uncertain results when used with the UltraTerm. You should remove it from your BASIC programs before you use them with your UltraTerm.

Section 5.d DEFAULT ATTRIBUTE SWITCHES

The video attributes that your UltraTerm uses when your computer is turned on or reset are selected by four DIP switches. The first two switches select the attributes used when the high bit of the character is zero, and the second two select the attributes when the high bit is one. We call these the standard and alternate attribute sets. In each of these two groups of

switches, one selects either highlight or lowlight intensity and the other selects normal or inverse video. The switches are arranged as follows:

SWITCH POSITION				
LEFT		**RIGHT**		
Highlight	-1-	Lowlight	**Intensity**	
				Standard Attributes
Inverse	-2-	Normal	**Video**	
Highlight	-3-	Lowlight	**Intensity**	
				Alternate Attributes
Inverse	-4-	Normal	**Video**	

When we shipped your UltraTerm, switches 1, 2 and 3 were set in the RIGHT position and switch 4 was set in the LEFT position. This results in lowlight normal video when the standard attributes are selected and lowlight inverse video when the alternate attributes are selected. This will give you the expected normal and inverse displays when you use the appropriate commands in BASIC.

CHAPTER SIX

Software Environments

reconnects the DOS I/O hooks. Here is a short 'Hello' program that will turn on the UltraTerm when the disk is bootstrapped. To use this as a 'Hello' program, you would type the program into the computer, then use the 'INIT HELLO' DOS command to initialize a blank disk. Be sure to clear any old programs from memory with a 'NEW' command before you type in the program.

```
10 PR#3: REM TURN ON THE UltraTerm
20 CALL 1002: REM RECONNECT THE DOS
30 HOME: REM CLEAR SCREEN
40 PRINT:" UltraTerm IN 80-COLUMN MODE"
50 PRINT:PRINT
60 END
```

If your UltraTerm is running and you want to restart Apple DOS or bootstrap the system, you should first set your Apple back to the 40-column mode. If you don't do this, the new DOS will not send the output to the UltraTerm and you won't see anything on the screen unless your hello program turns the UltraTerm on again. You can return to the 40-column mode by using the keyboard command 'CTRL-RESET'

Section 6.a.2 NORMAL USE

Normal use of your UltraTerm doesn't demand further action on your part once you have used the 'PR#3' command to turn on the card. You can use your Apple much like you would with the 40-column display. Your programs may take advantage of the features of the UltraTerm by sending output commands to the card. However, you will probably find that the input commands are used most often. The following commands can be sent from the keyboard at any time. Most of them work in just the same fashion that they do on the 40-column screen.

- A^C: Uppercase/lowercase toggle. This command will switch you from upper case input to lower case input or vice-versa. Only the A through Z keys on your keyboard are affected.
- K^C: This command will generate the '[' character. This character cannot be generated on a standard Apple keyboard without special software (like that in the UltraTerm firmware).
- S^C: This is the pause command. This command causes output to the UltraTerm card to halt so that you can examine the display.

NOTE: The A^C and K^C commands will be disabled if the UltraTerm ever receives a lower-case letter from the keyboard. In this case the firmware assumes that all the characters can be generated by the keyboard and these commands are not needed.

Some other commands, which are actually output commands, will be echoed to the UltraTerm by the DOS. Thus, you can use these commands from the keyboard as if they were input commands. The ones you will use most often are:

H^C: The Back-Space. It is also generated by the left-arrow key. When you enter this command from the keyboard, the DOS deletes the character preceding the cursor from the input buffer.

M^C: This character, the Carriage Return, will move the cursor to the beginning of the current display line. A line feed will be issued automatically.

Many of the display control and editing commands built into DOS and BASIC are also valid when used with the UltraTerm.

U^C: The Copy-Forward will cause the cursor to move one space to the right. In addition, the character which was under the cursor before the move will be picked up from the screen and sent to the computer as if it had been typed on the keyboard.

<ESC>: This is the lead-in command for the screen editing mode. The valid editing commands are:

0-8	Set the display mode (#lines and columns)
@	Clear the screen.
A	Cursor Right
B	Cursor Left
C	Cursor Down
D	Cursor Up
E	Clear to End of Line
F	Clear to End of Screen
I	Cursor Up
J	Cursor Left
K	Cursor Right
M	Cursor Down

The I, J, K, and M command characters may be repeated without entering another <ESC> for multiple cursor moves. The command will end with the first character which is not an I, J, K, or M.

HOME This command will clear the UltraTerm display and move the cursor to the upper left-hand corner of the screen.

INVERSE The UltraTerm will display all following characters in inverse video.

NORMAL All following characters will be displayed in normal (white on black) video.

Section 6.b PASCAL

The Apple Pascal operating system will automatically enable and use an UltraTerm card if the card is in slot #3. If the card is in some other slot, it cannot be used as the console device for Pascal. This is the reason that we have used slot #3 in all our examples in this manual. The Pascal system will enable the UltraTerm when it is bootstrapped. You will not need to execute any special commands. The SofTech P-System (An upgraded Pascal system offered by SofTech Inc.) will also automatically use an UltraTerm card in slot #3.

Section 6.b.1 CONFIGURATION

There is a program called 'SETUP' on the 'APPLE3:' disk of your Pascal system. You should execute this program, and when it asks if you have lower case, you should answer 'Yes' When it asks for the number of columns, you should answer '80' When you execute this program you will create a file called 'NEW.MISCINFO'. After the program is finished, you should use the filer to delete the old 'SYSTEM.MISCINFO' file and rename 'NEW.MISCINFO' to 'SYSTEM.MISCINFO' The new parameters will be used the next time you bootstrap the system. Some of the system messages will now appear in upper and lower case letters, and the prompt line at the top of the screen will be expanded. The 'SETUP' program is completely described in Chapter 8 of the Pascal Operating System Reference Manual. You will need to run the program only once. You can then transfer the new 'SYSTEM.MISCINFO' file to any other Pascal Boot disks you are using.

Section 6.b.2 NORMAL USE

Once you have configured the 'SYSTEM.MISCINFO' file, no further changes to the Pascal system are needed. You will be able to use the 80-column display of the UltraTerm just as you did the 40-column display, except that you will not have to bother with horizontal scrolling. In fact, the commands which would normally be used for horizontal scrolling are no longer defined. Since the system will display both upper and lower case letters, the Pascal Editor can now be used for word processing much more easily.

The Pascal system is much more selective about which control characters it will echo to the screen. Thus, many of the output commands which could be entered from the keyboard in BASIC are not available in Pascal. If you try to enter a control character which Pascal does not allow, the system will generally echo a '?' and ignore the command. The following commands are strictly input commands and are available in Pascal:

A^C: Uppercase/lowercase toggle. This command will switch you from upper case input to lower case input or vice-versa. Only the A through Z keys on your keyboard are affected.

K^C: This command will generate the '[' character. This character is used much more often in Pascal than in BASIC, as it is the character used to delimit array subscripts.

S^C: This is the pause command. This command causes output to the UltraTerm card to halt so that you can examine the display.

NOTE: The A^C and K^C commands will be disabled if the UltraTerm ever receives a lower-case letter from the keyboard. In this case the firmware assumes that all the characters can be generated by the keyboard and these commands are not needed. If you have an Apple with an Enhancer II, the '[' is generated with the <CTRL-','> sequence.

The Pascal editor will accept a number of other control characters. This editor is covered in more detail in the next chapter. Your own application programs can accept and use any control characters you want, as long as they are passed on by the system. The input command characters shown above cannot be used because they will be swallowed by the UltraTerm.

The output commands listed in Chapter five can be used with Pascal just as they are with BASIC. The Pascal Editor will not allow you to directly embed the control characters in strings to be printed. Therefore, you will have to use the CHR(NN) function to print the control characters. The following Pascal statement would select the alternate character attribute set:

WRITE(CHR(15)); {set alternate attributes
—usually inverse video}

Some Run-time Pascal programs such as VisiSchedule, the Wizardry game and early versions of PFS, will force the system to use the 40-column screen. Unfortunately, these programs also initialize the UltraTerm card. This causes the UltraTerm card to set the video switch to the 80-column mode. As a result, you may not see any of the output from the program. At this time we do not have any software patch to solve this problem. The newer versions of these programs are generally written to avoid this problem. You should contact your software supplier if your UltraTerm does not work properly with any of these programs. While you wait for updated software, all we can suggest is that you manually move the output connector from your UltraTerm card to the normal video output on the back of your Apple.

Section 6.c CP/M

The CP/M operating system, when used with the Microsoft Softcard or other Z-80 cards, will automatically use the UltraTerm card for output. The UltraTerm card must be in slot #3 to be used automatically.

You can experiment with the video format and the character attributes directly from the CP/M command mode. This can be done because CP/M will echo the UltraTerm command characters to the screen, followed by a question mark. The question mark appears because the UltraTerm commands are not valid CP/M commands. If you change the video format with the <V^C> command, you won't see the question mark, since the screen will be cleared immediately. If you change the character attributes while experimenting, you can return to the default parameters by selecting a video format with the <V^C> command. The firmware will select the default attributes when the new video format is enabled.

Some of the output commands of your UltraTerm will not work properly when used directly from the CP/M command mode. This is because the operating system intercepts them and changes them before they are echoed to the terminal. The <L^C> (home cursor and clear screen) command is a good example. The command character is intercepted by CP/M and changed to the Cursor Right character. The translation of command characters is handled by using two tables in the CP/M I/O configuration block. Your CP/M system comes with a utility program, CONFIGIO, which allows you to modify these tables.

Some particular CP/M programs which can use UltraTerm features, such as Wordstar, are discussed in the next chapter.

CHAPTER SEVEN

Some Specific Software

Section 7.b WORDSTAR

Wordstar comes with a configuration program called 'INSTALL' You will need to run INSTALL to create the proper editing environment for your UltraTerm.

Section 7.b.1 CONFIGURATION

The INSTALL program is described in the Wordstar Installation Manual. Appendix B of that manual has some details specific to the Apple][. There are three main categories of information that are needed to INSTALL Wordstar:

1. display and keyboard information
2. printer information
3. custom Wordstar program changes

When asked about the display and keyboard, you will notice that you are given choices that include Videx cards. If your version of Wordstar does not include the UltraTerm as an option, then you should respond as if you had our Videoterm card. If you have no special keyboard hardware, you should select the option which specifies the UltraTerm with software U/L conversion.

If you have a keyboard enhancer such as the Videx Enhancer][or the Videx Keyboard and Display Enhancer, then you should specify a UltraTerm with hardware U/L conversion. If you have modified your shift key as described in the Wordstar Installation Manual, then UltraTerm with shift mod option is the proper response. If you are used to using your <esc> key as a shift, then it is acceptable to specify the UltraTerm with software U/L conversion.

The information you provide about your printer will not be affected by the use of the UltraTerm. Specify this information as you normally would.

Near the end of the installation process, you will be asked 'ARE THE MODIFICATIONS TO WORDSTAR NOW COMPLETE?' The usual response is 'N'. This will invoke the 'patcher' The patcher is described in the Wordstar Installation Manual. The patcher will ask for addresses of data which need to be changed, and for new data to put in those addresses. The following table contains the patches you will have to make to use the 80 × 48 mode:

Address	Data	
248	30	sets 48-line mode
284	2	
285	1B	
286	28	
28B	2	
28C	1B	
28D	29	
0	(Entering a zero tells the patcher that you are finished.)	

After you are done with the patcher, INSTALL will go to its normal confirmation and termination messages.

If, in the future, you wish to turn off alternate characters, then simply change address 284 and 28B to both be Ø.

Section 7.b.2 NORMAL USE

Wordstar for the Apple][was written with 8Ø-column output in mind. This means that most of the "normal use" information in the Wordstar manual is valid. one feature which does need to be remembered is the proper way to toggle between uppercase and lowercase. 'A^C' will not work with Wordstar. Wordstar has its own mechanism for changing case using the <ESC> key. There is also a shift key modification described in the Wordstar documentation. You may also use a Videx Enhancer][for true typewriter-like operation.

NOTE: If you configure Wordstar for a non-standard operating mode, such as 8Ø × 48, you must first set that same format in the CP/M command mode with a <V^C6>.

Section 7.c APPLEWRITER][PREBOOT

We will be offering a preboot diskette for the Applewriter][word processing program. At press time for this manual, the exact specifications for this program have not been finished. We expect that the 256-byte addressing mode and enhanced display quality of your UltraTerm will combine to make the combination of Applewriter][, Preboot and UltraTerm a very attractive word processing package.

The Applewriter preboot disk will allow you to select one of three modes using 24, 32 or 48 lines by 80 columns. The Preboot will work either with Applewriter][or Applewriter //e. On the Apple //e you may also use an extended memory card if you have one installed. Please note that using the extended memory will slow the response of the program to keyboard input. If you find that the keyboard response is too slow with the 32 or 48-line formats, you should use the 24-line format.

CHAPTER EIGHT

The Programmer's Guide

The Programmer's Guide

This chapter will help you to write and modify programs to take advantage of the features of your UltraTerm. The programming techniques which you can use with each language will be explained and we will show you examples of some of the more important ones.

Section 8.a GENERAL CONSIDERATIONS

When you want to use the special features of your UltraTerm, you should normally activate them with the output commands described in Chapter 5. In some special cases, you may want to use one of the other two methods of controlling your UltraTerm. These two methods are language-specific commands and 'PEEK' and 'POKE' statements. Some languages, such as Pascal, may place very strict limitations on how you can use these latter two methods. Most general-purpose computer languages and many applications programs will allow you to control your UltraTerm by sending it output commands. The next three sections of this chapter will show you how to use the features of your UltraTerm in the three most common programming languages on the Apple: BASIC, Pascal and Assembly Language. Since the Apple FORTRAN language uses the Pascal operating system, the methods we will describe for Pascal can be used with FORTRAN. Of course, you will have to use FORTRAN output statements within your programs, but the other elements of the operating system, such as the Editor and Filer program, are the same as those used with Pascal.

Section 8.b APPLESOFT AND INTEGER BASIC

In this section we will examine the features of your UltraTerm that can be used with the two versions of BASIC that are available on your Apple. There are some differences in the command structures of the two versions of BASIC. Where these differences will affect your use of your UltraTerm, we will explain the differences. The largest difference is that there are more commands available in Applesoft. So, quite often, we will have to tell you that a command won't work with Integer BASIC. Where possible, we will give you another way to get the same result.

Section 8.b.1 ACTIVATING YOUR ULTRATERM

The simplest way to activate your UltraTerm is to use the 'PR#3' before you start running your program. If your Apple is displaying the BASIC prompt

(either the ']' in Applesoft or the '>' in Integer BASIC) you can simply enter the 'PR#3' command and not have to worry about re-connecting DOS.

If you want to have the 'HELLO' program on your boot disk activate your UltraTerm, you will have to use a slightly different procedure. When the 'PR#3' command is used within a program, it will disconnect DOS. If you later use a DOS command without re-connecting DOS, you will get a 'SYNTAX ERROR' message. The following line will activate your UltraTerm, then re-connect DOS:

---

10 PR#3 : CALL 1002 : REM ACTIVATE UltraTerm AND RECONNECT DOS

Of course, you do not need to add the REM statement to make the program line work, but it will certainly make your programs easier to understand.

When you use the 'PR#3' command, your UltraTerm firmware will also simulate an 'IN#3' command. This allows you to use the input commands such as the S^C, (the output pause command) from your keyboard. You will never need to use the 'IN#3' command in your programs.

Section 8.b.2 ULTRATERM OUTPUT COMMANDS

All of the output commands we described in Chapter Five work with both versions of BASIC. You can include these commands in your programs using PRINT statements to send the commands to your UltraTerm.

Section 8.b.3 TEXT MODE COMMANDS

Both Applesoft and Integer BASIC have several built-in commands you can use to control the text display. We have listed these commands along with any new information you will need to use them with your UltraTerm.

FLASH This command will not work with your UltraTerm. If you use it by accident, you will get an unreadable display because BASIC will change the ASCII codes which are sent to the UltraTerm.

HOME You will not have to change this command in your BASIC programs. It will work just as it does with the 40-column display: the cursor will move to the upper left-hand corner and the display will be cleared. This command is not available in Integer BASIC, but you can simulate it with a 'CALL -936'.

HTAB There are some limitations to the way you can use this command with your UltraTerm. You can HTAB only in the forward direction and you cannot HTAB past column 40. We

recommend you use the 'POKE 36, HT' command where HT is the column number to which you want to move the cursor. This alternative command is also limited to movement in the forward direction.

INVERSE This command will function properly with your UltraTerm.

NORMAL This command will switch back to normal (white on black) video if you have used the INVERSE command.

POS This command does not work with the UltraTerm. You should use the 'PEEK' command as described in section 8.b.5.

PRINT The use of commas and semicolons for print formatting is fully supported by your UltraTerm. The 'PRINT TAB(HT);X1' will not work with your UltraTerm.

SPC This command works properly with your UltraTerm.

VTAB This command will work properly as long as you VTAB to a line number less than 25. If you are using the 48-line mode, you should use the 'POKE' commands described in section 8.b.5 to move to lines on the lower half of the screen.

These Applesoft commands provide a convenient way for you to experiment with the character attributes available with your UltraTerm. You can use the INVERSE, NORMAL and <CTRL>'V' commands directly from the keyboard to change the attributes of displayed characters. For a more complete discussion of the programming of the character attribute register see section 8.d.3.

The Apple BASIC manual mentions several monitor ROM routines which can be used to control the screen display by using CALL statements. Most of these routines (except the CALL -936 mentioned above) will not work with your UltraTerm. Indeed, some of these routines may cause unexpected results with your UltraTerm. You can use the UltraTerm output commands to get the same results that are produced by these 'CALL' commands.

Section 8.b.4 GRAPHICS MODE COMMANDS

Before we start describing the Apple graphics mode commands, we would like you to note that we are not going to be describing the line drawing and block graphics characters included in the UltraTerm character set. These special characters are discussed in Appendix B.

The Apple graphics mode commands will not automatically switch the video display to the Apple video signal. To display the graphics screens you must use the 'V^{C}-Ø' command to turn the Apple video back on. You can then select the appropriate graphic mode with the 'HGR', HGR2' or 'GR' command. Once you have selected the Apple video signal, you can use the commands listed in your Apple manuals to set the appropriate graphics mode. To return to the UltraTerm text mode you must use the 'PR#3' command. This command will switch the video signal back to your

UltraTerm. The cursor may not be where you left it, but any special display modes will still be selected. In particular, if the 48-line mode was selected and the cursor was positioned below line 24, the cursor will be moved up to line 24 when you switch from graphics to text mode.

Section 8.b.5 SCREEN CONTROL WITH 'PEEK' AND 'POKE'

The 'PEEK' and 'POKE' commands in BASIC can be used to control some of the operations of your UltraTerm. The Apple 40-column screen will allow you to set text windows by poking the window values into locations 32 through 35. These locations are not used by your UltraTerm firmware. In addition, since your UltraTerm uses a fast hardware scrolling method, you cannot set text windows on the UltraTerm display.

Your UltraTerm will support some of the methods of determining and altering the cursor position. However, your UltraTerm uses its own special memory locations to store the cursor horizontal and vertical positions. When you want to determine the cursor position, we recommend that you examine these locations. For a UltraTerm in slot #3, these locations are:

Cursor Horizontal—PEEK(1395)
Cursor Vertical—PEEK(1523)

Examining the normal cursor horizontal and vertical locations (36 and 37) may not give you the right result if the cursor is outside the standard Apple text window.

You can use 'POKE 36,CH' and 'POKE 37,CV' commands to move the UltraTerm cursor anywhere on the display screen. A 'POKE 36, CH' will move the cursor to column CH. Using 'POKE 37, CV' moves the cursor to line CV. You should note that the cursor displayed on your screen will not move until you actually print a character. If you want to move the cursor without displaying anything on the screen, you can print an ASCII <NULL> command (CHR$(0)) after you have POKED the new values into the cursor locations. This character will not change any of your UltraTerm settings and will not show up on the screen.

Section 8.b.6 DIRECT KEYBOARD INPUT

You may find that there are times when you want to accept input directly from the keyboard with your programs. You can monitor the keyboard directly from BASIC with the following routine:

```
10 REM * SUBROUTINE TO GET A CHARACTER *
20 LET KEY = PEEK(-16384): REM LOOK AT KEYBOARD
30 IF KEY < 128 THEN 20: REM REPEAT UNTIL KEY PRESSED
40 POKE -16368,0: REM CLEAR KEYBOARD STROBE
50 LET GC$ = CHR$(KEY-128): REM GC$ = INPUT CHARACTER
60 RETURN
```

Section 8.c PASCAL

You can use all of the features of your UltraTerm with the Apple Pascal operating system. However, due to the more structured nature of this system, using some of the card's features requires more advance planning and careful programming. In this section we will show you how to configure the operating system to take advantage of the special features of your UltraTerm and how to use the UltraTerm commands within your programs.

We would like you to note that many of the examples that we will give are not complete programs, but only code segments that you can insert into your own program. These code segments cannot be compiled and executed by themselves. We will have to assume that you are familiar with the Pascal system and can use the Editor and other operating system programs to add these code segments to your own programs.

Section 8.c.1 INITIALIZING THE ULTRATERM

Your UltraTerm card will be automatically activated by the Pascal operating system if it is installed in slot #3. This is the reason we have used slot #3 for all of our programming examples. Once the card has been activated, the operating system will no longer allow you to switch to the Apple's 40-column text display. Thus, you should not use the 'V^{C}-0' output command to disable your UltraTerm. If you do use this command, you will have to re-boot your Pascal system. When the Pascal system enables your UltraTerm, it will automatically set the electronic switch to select the video signal from the UltraTerm.

Section 8.c.2 DISPLAY MODES AND SYSTEM.MISCINFO

When your Pascal system is initialized, it reads the characteristics of the system console (the keyboard and video display) from a file called SYSTEM.MISCINFO on the boot disk. The data in this file when you first get your Pascal system is configured to operate the UltraTerm in the 80-column by 24-line mode. There is a program on your APPLE3 disk which will allow you to change the information in the SYSTEM.MISCINFO file. This program is called SETUP.

If you want to use your UltraTerm in one of the other display modes (80 × 48, 132 × 24, or 160 × 24), you must do three things:

1. You must change the information in the SYSTEM.MISCINFO file to match the new mode. The parameters you must change are the Screen Width and the Screen Height. This is done by X)ecuting the SETUP program.
2. You must re-initialize the Pascal system so that the new values for screen height and width will be read from the SYSTEM.MISCINFO file. This can be done by using the I)nitialize command of the operating system.
3. Next, you should select the new display mode by transmitting the appropriate output command ('V^C-4' etc.) to the UltraTerm with a simple program. You cannot use the input commands ('ESC-4' etc.) because the Pascal system does not allow the use of these commands.

Section 8.c.3 ULTRATERM OUTPUT COMMANDS

Output commands can be sent directly to your UltraTerm with the 'WRITE' and 'WRITELN' commands. The 'CHR' function is used to send control characters in the same way that the 'CHR$' command is used in BASIC. All of the output commands which can be used in BASIC are also available in Pascal. However, many of the commands will seldom be used directly because Pascal has its own built-in functions to accomplish the same tasks. An example is the 'GOTOXY' procedure which can be used to move the cursor instead of the UltraTerm 'CHR(30)' command. You should note that Pascal will not allow you to enter the control codes for commands directly into the strings used in 'WRITE' commands. The control characters will not be accepted by the Pascal Editor when you write the program. This forces you to use the 'CHR' function when you write your programs. Here is a sample program which can be used to select one of the alternate video modes:

```
Program Setmode;
{ This is a program to select one of the alternate video modes on the
UltraTerm card. It does not alter the SYSTEM.MISCINFO file. }
Var   Selection:char;
Begin
  Repeat
    Page(Output); {Pascal equivalent of BASIC 'HOME'}
    Gotoxy(10,5);
    Writeln('***  UltraTerm Alternate Display Mode Selection  ***');
    Writeln;
```

```
    Writeln('You may select one of the following modes: ');
    Writeln('   1  80 × 24, non-interlace (normal mode)');
    Writeln('   2: 96 × 24, non-interlace');
    Writeln('   3: 160 × 24, non-interlace');
    Writeln('   4: 80 × 24, interlaced');
    Writeln('   5: 80 × 32, interlaced');
    Writeln('   6: 80 × 48, interlaced');
    Writeln('   7: 132 × 24, interlaced');
    Writeln('   8: 128 × 32, interlaced');
    Writeln;
    Write('Enter your selection by number ');
    Write(' or enter "E" to exit: ');
    Repeat
        Read(Keyboard, Selection);
    Until Selection in ['1'..'8','E','e'];
    { Now send control characters to select mode—just send
      Control-V followed by the Selection character!}
    If Selection in ['1'..'8'] then Write(Chr(22),Selection);
  Until Selection in ['E','e'];
End.
```

Section 8.c.4 PASCAL-SPECIFIC COMMANDS

The Apple Pascal language has several built-in commands that allow you to use the features of your UltraTerm without special programming techniques. Here is a list of those commands and their operation with your UltraTerm:

Page(Output); This command will clear the video display and move the cursor to the upper left-hand corner.

GotoXY(XX,YY); This will move the cursor to column XX and line YY. XX and YY must be integers and must be within the screen width and height values set in SYSTEM.MISCINFO.

GRAFMODE; This command is part of the Turtlegraphics unit in the System Library. This will reserve the memory space for the hi-res graphics display. With an Apple 40-column display, it would also switch to the graphics video output. With your UltraTerm you will also have to use the statement

Write(Chr(22),'0');

to switch the video output to the Apple video signal.

TEXTMODE; This command, also part of the Turtlegraphics unit, normally returns you to the text display. This command does not work properly with the UltraTerm, since it uses an electronic switch which your UltraTerm does not use. With your UltraTerm you can return to the text display at any time by simply writing any character to the display. The statement

Write(CHR(0));

would switch you back to the UltraTerm text display.

Section 8.d ASSEMBLY LANGUAGE

This section will give you an introduction to the techniques you can use to program your UltraTerm in assembly language. For more detailed information on the theory of operation, memory usage and CRTC register usage, you should see the appendices.

Section 8.d.1 INITIALIZING THE ULTRATERM

You can use the following routine to switch from the Apple 40-column display to the default 24-line by 80-character display:

```
LDA #$00        ASCII NULL CHARACTER
JSR $C300       UltraTerm INITIALIZATION
JMP $03EA       RE-CONNECT DOS, THEN
                RETURN
```

We strongly suggest that you use this routine to initialize your UltraTerm, rather than directly programming the CRTC registers. We have spent a lot of time determining the proper values for the registers in each mode. We would like to keep you from duplicating this effort needlessly. After you have called this subroutine and returned to your own program, the UltraTerm will be initialized, the video signal switched to the UltraTerm and the screen will be cleared. The DOS I/O hooks will be set up and all DOS files will be closed.

Section 8.d.2 SIMPLE INPUT AND OUTPUT

The easiest way to get a keyboard entry is to call the 'RDKEY' routine in the monitor ROM. This routine is located at $FD0C. This routine will allow the 'CTRL-A' input routine for switching between upper and lower-case input to work properly. The ASCII code for the key pressed will be returned

in the accumulator. If you wish, you can write your own input routines which directly manipulate the Apple keyboard I/O locations. If you do this, you will have to write your own routines to simulate the UltraTerm input commands.

To send a character to the UltraTerm, place the ASCII code in the accumulator and call the 'COUT' routine in the Apple monitor ROM. This routine is located at $FDED. The following routine shows how you could use this routine to set the 24-line by 132-character display mode:

```
SET 132   LDA #22        LOAD CTRL-V
          JSR $FDED      OUTPUT VIA COUT
          LDA #'2        MODE 2 FOR 24 × 132
          JMP $FDED      OUTPUT VIA COUT AND
                         RETURN
```

Section 8.d.3 MEMORY USAGE AND CRTC PROGRAMMING

Your UltraTerm uses eight slot-dependent storage locations in the 40-column screen memory area. These locations are used to store variables used in the firmware routines. You can examine these locations in your assembly-language programs to determine the status of your UltraTerm. These storage locations (for card in slot #3) are used as follows:

Address	Name	Usage
$047B	**BASEL**	Low byte of screen base address
$04FB	**BASEH**	High byte of screen base address
$057B	**CHORZ**	Cursor horizontal position
$05FB	**CVERT**	Cursor vertical position
$067B	**BYTE**	I/O Byte for Pascal entries
$06FB	**START**	(Screen start address)/16
$077B	**POFF**	Power-Off flag and Lead-in counter
$07FB	**FLAGS**	General-purpose flags register

Your UltraTerm also uses the sixteen addresses beginning at $C0B0 to control the operation of the card. Some of these addresses are write-only locations, others may also be read, however, the data byte that you read has no significance. It is the reading of a particular address that will set a specific operating mode. The following table defines the control registers for a card in slot #3:

Address	Read	Write
$C0B0	Select character RAM Page 0 (512-byte mode)	Select UltraTerm video, Select CRTC register #
$C0B1	No Effect	CRTC data written to selected register
$C0B2	No Effect	Mode Control Port
$C0B3	No Effect	Video Attribute Register
$C0B4	Select Character RAM Page 1 (512-byte mode)	No Effect
$C0B8	Select Character RAM Page 2 (512-byte mode)	No Effect
$C0BC	Select Character RAM Page 3 (512-byte mode)	No Effect

Reading or writing to addresses marked 'No Effect' will have no predictable effect on the operation of your UltraTerm. However, it may have unpredictable effects! We recommend that you read and write only to the device control addresses as we have defined them. If you mis-use them or use addresses not defined in the table, you may get puzzling or frustrating results.

The Mode Control Port ($C0B2) is used to set the operating mode of your UltraTerm. Setting and clearing the bits in this port control the operation of the card as defined in the following table:

Bit	Function
7	Firmware Page Select
6	Video Signal Select 1 = UltraTerm
5	Clock Frequency 1 = 28.7595, 0 = 17.430 MHz
4	Character Address Format 1 = 256-Byte Pages, 0 = 512-Byte Blocks
3	Character RAM Address bit 11 (256-byte mode)
2	Character RAM Address bit 10 (256-byte mode)
1	Character RAM Address bit 9 (256-byte mode)
0	Character RAM Address bit 8 (256-byte mode)

The Character Attribute Register ($C0B3) is used to set the display attributes for the characters stored in the display RAM. Each character may be displayed on the screen with one of two sets of attributes. One set will be selected if the high bit of the character in the RAM is set, the other if the high bit is clear. A set of attributes is selected by combining the following characteristics:

Bit 2—Standard or Alternate Character Set
Bit 1—Inverse or Normal Video
Bit 0—Highlight or Lowlight Dot Intensity

When you write a byte into the Attribute Register, the high nibble (bits 4–7) sets the attributes for characters with the high bit set. The lower nibble sets the attributes for characters with the high bit clear. Only the lower three bits of each nibble are significant, as there are only three possible attributes for each character. The following table shows the attributes you will get for a particular nibble written to the attribute port:

Nibble Value	Resulting Display Characteristics		
7	Alternate char. set	inverse video	highlight
6	Alternate char. set	inverse video	lowlight
5	Alternate char. set	normal video	highlight
4	Alternate char. set	normal video	lowlight
3	Standard char. set	inverse video	highlight
2	Standard char. set	inverse video	lowlight
1	Standard char. set	normal video	highlight
0	Standard char. set	normal video	lowlight

NOTE: These NIBBLE values are used with the W^C command to set the display attributes.

For a more complete description of the way the device control locations, mode control port, and video attributes function, you should consult the appendices.

CHAPTER NINE

The Hardware Interface

The Hardware Interface

In this chapter we will describe the requirements for the video display hardware you will need to take full advantage of your UltraTerm card. We will also discuss the compatibility of your card with other peripherals you may have plugged into your Apple. While we have tested the UltraTerm with many of the cards and programs available for the Apple, we cannot guarantee that we have tested the particular combination of peripherals in your computer. If you discover any problems or unusual interaction between your UltraTerm and other cards in your Apple, please contact our customer support department.

Section 9.a VIDEO DISPLAY MONITORS

If you are going to be completely satisfied with your UltraTerm, you must use it with a compatible video display monitor. There are two primary requirements you need to consider when selecting a video display. They are the video bandwidth or resolution of the display and the persistence of the phosphor used on the display screen.

Your UltraTerm requires a monitor with a minimum bandwidth of 15 mHz to produce a sharp display in the 128, 132 or 160-character per line modes. When you are using these modes, your UltraTerm is using a 28-mHz clock to send the display dots to your monitor. This frequency is about 1.6 times greater than the clock frequency used in the 80-column mode. As a result, many of the display monitors which provide reasonable results in the 80-column mode may not work well in the wider display modes. We have done all we can to minimize the requirements for the video monitor, but there is simply no way we can make the 160-column display work on some monitors.

The interlaced display mode used to display 32 or 48 lines of text on your display writes the characters to the display only thirty times per second. This is half the scan rate used in the 24-line modes. As a result, if you use a monitor with a low-persistence phosphor, you may notice a shimmer or flicker of the image on the screen. This flicker can be eliminated by using a display screen with a phosphor which continues to emit light for several milliseconds after it has been scanned. Phosphors of this type are called 'long-persistence' The phosphor on the Apple Monitor /// has sufficient persistence to eliminate flicker in the 32 and 48 line modes of your UltraTerm.

Many video display monitors actually sweep the electron beam which lights up the phosphor dots past both edges of the display screen. This extended sweep is called 'overscan' If your display monitor has excessive overscan, it may not show all the characters at the beginning or end of a display line. Your display will look as if your screen is providing a window into another display several inches wider. In particular, we have noticed

that the overscan on the Apple Monitor /// makes it impossible to display all the characters in the 160-character per line mode. This is in spite of the fact that the monitor has more than adequate bandwidth. In defense of the Monitor /// and other monitors which have some overscan, we should note that a reasonable amount of overscan is very helpful in reducing distortion at the edges of the screen.

On some display monitors, you may reduce the overscan by reducing the width of the display. You will have to make your own decision about any tradeoffs between increased display width and increased distortion.

While testing various display monitors with the UltraTerm we have arrived at the following conclusions:

Apple Monitor /// An excellent overall display, it will allow you to use all the display modes except the 24-line by 160-character mode and the 24 × 96 mode. The Monitor /// is our choice as the best monitor to use with the UltraTerm.

NEC JB-902M This 9-inch display has adequate bandwidth to display all the video modes of the UltraTerm. The monitor has minimal overscan and can display a full 160-character line. Many users will find the display too small for comfortable use with either the wider displays or the 48-line mode. The monitor also has a short-persistence phosphor which results in a noticeable shimmer with the interlaced display modes.

NEC JB-1201M This 12-inch monitor has the same characteristics as the 9-inch JB-902M.

Leedex Video 100 This was one of the first display monitors available at a reasonable price. While it may be adequate for the 80-column modes, the resolution is only just acceptable for the 132 and 160-character modes. The phosphor has a short persistence and is not suitable for the interlaced display modes.

Amdek 300A This 12-inch monitor has a long-persistence amber phosphor. The scan limits are set up so that you can use any of the display modes of your UltraTerm. The characters displayed are sharp and clear. An excellent monitor.

Section 9.b MODEMS AND COMMUNICATIONS PROGRAMS

Your UltraTerm is compatible with all the modems and communications programs we have tested. The DC Hayes Micromodem][firmware will work with your card, but it will not provide nearly as much control and operating convenience as a good data communications program. We particularly recommend ASCII Express, PRO version by Southwestern Data Systems. Your UltraTerm should have no problems with other cards which follow Apple's peripheral card protocols.

Section 9.c PRINTERS AND PRINTER INTERFACES

Your UltraTerm should co-exist peacefully with your printer interface. Some printer interfaces may not correctly format data sent to the screen as well as the printer—particularly when print formats more than 40 columns wide are used. The VIDEX Serial/Parallel card is one interface which will allow you to use the full display width of your UltraTerm while echoing printed characters to the screen. Furthermore, the 132-character display mode of your UltraTerm will simplify the design and previewing of forms which will be printed on 15-inch paper or with compressed print on 8-1/2 inch paper.

Character Code

THE ASCII CHARACTER CODE CHART (7 & 8 BITS)

Decimal:		0	16	32	48	64	80	96	112
or:		128	144	160	176	192	208	224	240
	Hex:	$00	$10	$20	$30	$40	$50	$60	$70
	or:	$80	$90	$A0	$B0	$C0	$D0	$E0	$F0
0	$0	@ᶜ Nul	Pᶜ Dle		0	@	P	`	p
1	$1	Aᶜ Soh	Qᶜ Dc1	!	1	A	Q	a	q
2	$2	Bᶜ Stx	Rᶜ Dc2	"	2	B	R	b	r
3	$3	Cᶜ Etx	Sᶜ Dc3	#	3	C	S	c	s
4	$4	Dᶜ Eot	Tᶜ Dc4	$	4	D	T	d	t
5	$5	Eᶜ Enq	Uᶜ Nak	%	5	E	U	e	u
6	$6	Fᶜ Ack	Vᶜ Syn	&	6	F	V	f	v
7	$7	Gᶜ Bel	Wᶜ Etb		7	G	W	g	w
8	$8	Hᶜ Bs	Xᶜ Can	(	8	H	X	h	x
9	$9	Iᶜ Ht	Yᶜ Em	)	9	I	Y	i	y
10	$A	Jᶜ Lf	Zᶜ Sub	*	:	J	Z	j	z
11	$B	Kᶜ Vt	[ᶜ Esc	+	;	K	[	k	{
12	$C	Lᶜ Ff	\ᶜ Fs	,	<	L	\	l	|
13	$D	Mᶜ Cr	]ᶜ Gs	-	=	M	]	m	}
14	$E	Nᶜ So	^ᶜ RS			N	^	n	~
15	$F	Oᶜ Si	_ᶜ Us	/	?	O	_	o	rub

HOW TO READ THE ASCII CHARACTER CODE CHART

The ASCII value of any character in the chart may be determined by adding the value at the top of its column with the value to the left of its row. The table may be used to find values in either decimal (base ten) or hexadecimal. The first two columns of characters are the control characters. They are followed by their ASCII names.

Example: A control G is represented by: Gᶜ Bel. "Bel" is a short hand notation for "bell", meaning the bell character. Its ASCII value is $7 or $87 (hexadecimal) or 7 or 135 (decimal).

Character Sets

This Appendix shows the two character sets that come with your UltraTerm. The figures were originally printed by dumping an image of the High-Res graphics screen to an Epson MX-80 printer. The High-Res screen displays were produced by the font editor we use to design character sets. Since the proportions of the printout may not match the proportions of your screen, you may find that the characters on your screen look somewhat different.

Section B.1 THE STANDARD CHARACTER SET

Figure B.1 shows the standard character set. This character set does not require interlace except when 48 lines are displayed. The hexadecimal value for the character code can be determined by combining the value along the left side of the figure with the value over the character. For example, the code for the capital 'P' is hexadecimal 50. Note that the values from $10 to $1F contain eight block graphic characters and eight ASCII symbols. This character set uses a dot matrix which is 9 dots wide and 12 dots high.

NOTE: In both character sets, the character with code 0 (ASCII NUL) must not have any bits set. This is a required to maintain proper video levels during the video blanking interval.

Figure B.1 Standard Character Set

Section B.2 THE HIGH-QUALITY CHARACTER SET

The High-Quality Character Set, which uses a 9 × 16 dot matrix, is shown in Figure B.2. Please note that this character set includes sixteen block graphics characters and sixteen line-drawing characters. The ASCII symbols are not part of this character set.

Figure B.2 High-Quality Character Set

Section B.3 EUROPEAN CHARACTER SETS

Your UltraTerm can be equipped with special character set EPROMS to allow you to display characters used in many European languages. These characters sets are an option which you must purchase either from your dealer or directly from us. The languages supported and the characters which are different from the normal ASCII character set are shown in figure B.3.

Standard Character Set

HEXADECIMAL	23	40	5B	5C	5D	60	7B	7C	7D	7E
ENGLISH (UK)	£	@	[	\	]	`	{	¦	}	~
GERMAN	#	§	Ä	Ö	Ü	`	ä	ö	ü	ß
FRENCH	£	à	°	ç	§	`	é	ù	è	¨
ITALIAN	£	§	°	ç	é	ù	à	ò	è	ì
SWEDISH	#	@	Ä	Ö	Å	`	ä	ö	å	~
SPANISH	£	§	¡	Ñ	¿	`	°	ñ	ç	~

High Quality Character Set

HEXADECIMAL	23	40	5B	5C	5D	60	7B	7C	7D	7E
ENGLISH (UK)	£	@	[	\	]	`	{	¦	}	~
GERMAN	#	§	Ä	Ö	Ü	`	ä	ö	ü	ß
FRENCH	£	à	°	ç	§	`	é	ù	è	¨
ITALIAN	£	§	°	ç	é	ù	à	ò	è	ì
SWEDISH	#	@	Ä	Ö	Å	`	ä	ö	å	~
SPANISH	£	§	¡	Ñ	¿	`	°	ñ	ç	~

Figure B.3 European Character Sets. The alternate characters for each language are shown under the hexadecimal value for the character.

CRTC REGISTERS

This appendix will describe how to communicate with the CRTC. The CRTC has two memory addresses allocated to it. Data written to the first address will control which one of eighteen internal CRTC registers will respond to the second address. The two addresses used are:

$C0B0 You select a CRTC register by writing the register number to this address.

$C0B1 You write data to be stored in the CRTC register to this address. Some of the CRTC register can also be read. Others will return garbage data.

Section C.1 REGISTER SUMMARY

The default values for the 80-column videoterm emulation mode are summarized in table C.1

TABLE C.1

Register Number	Register Description	Access Type	Power-on Value
R0	horizontal total	write	$82
R1	horizontal displayed	write	$50
R2	horizontal sync position	write	$64
R3	horizontal sync width	write	$29
R4	vertical total	write	$1B
R5	vertical adjust	write	$08
R6	vertical displayed	write	$18
R7	vertical sync position	write	$1A
R8	interlace mode	write	$00
R9	max. scan line address	write	$08
R10	cursor start	write	$E0
R11	cursor end	write	$08
R12	start address (high)	write	$00
R13	start address (low)	write	$00
R14	cursor address (high)	read/write	$00
R15	cursor address (low)	read/write	$00
R16	light pen (high)	read	
R17	light pen (low)	read	

R0 **Horizontal total** This is an 8 bit write-only register that determines the horizontal scan frequency. The count which is stored here is in character time units. Use the number of displayed characters plus the number of non-displayed character times, minus 1

R1 **Horizontal displayed** This is an 8 bit write-only register that determines the size of the horizontal display area. The count which is stored here is the number of displayable characters per line.

R2 **Horizontal sync position** This is an 8 bit write-only register that determines where in a horizontal scan the sync pulse will occur. The data is in character time units.

R3 **Horizontal sync width** This is a 4 bit write-only register that determines the width of the horizontal sync pulse. The data is in character time units.

R4 **Vertical total register** This is a 7 bit write-only register that, along with R5, determines the vertical refresh rate. The number stored here is the number of displayed lines plus the number of non-displayed lines that allow for 50 or 60 Hz refresh rates, minus 1. The number will usually come out with a fractional part. Just the integer part should be used here.

R5 **Vertical adjust** This is a 5 bit write-only register that contains the fraction needed to augment the integer value described for R4.

R6 **Vertical displayed** This is a 7 bit write-only register that determines the size of the vertical display area. The count which is stored here is the number of character display lines.

R7 **Vertical sync position** This is a 7 bit write-only register that determines the position of the vertical sync pulse.

R8 **Interlace mode** This is a 2 bit write-only register that specifies whether or not to interlace scan, and if so, what type of interlace. Bits 0 and 1 determine the interlace mode as follows:

Bit 0 clear, bit 1 set or clear normal sync mode. In this mode there is no interlace.

Bit 0 set, bit 1 clear interlace sync mode. Each scan line is output twice. This doubles the number of scan lines, without doubling the screen memory or font EPROM sizes. The scan lines themselves will only have half the normal spacing.

Bit 0 set, bit 1 set interlace sync with video mode. In this mode there will be twice as many unique scan lines output. The scan lines will only have half the space between them. A monitor with a long persistence phosphor is required.

R9 **Maximum scan line address** This is a 5 bit write-only register that determines the height of the character font. The value stored here should be the number of scan lines for a character (including any blank space above or below) minus 1

R10 **Cursor start** This is a 7 bit write-only register that determines cursor type, and top of cursor within a character cell. Bits 5 and 6 determine cursor type as follows:

Bit 6 clear, bit 5 clear a non-blinking cursor is displayed.

Bit 6 clear, bit 5 set no cursor is displayed.

Bit 6 set, bit 5 clear the cursor will blink at 1/16th of the field rate.

Bit 6 set, bit 5 set the cursor will blink at 1/32nd of the field rate.

Bits 0 through 4 define the top of the cursor within the character cell. Valid numbers to specify in bits 0 through 4 are 0 through 11 (decimal).

R11 **Cursor end** This is a 5 bit write-only register that determines the bottom of the cursor within a character cell. The number stored here must be smaller than, or equal to, the value used for the top of cursor (bits 0 through 4 of R10).

R12 **Start address (high)** When combined with R13, this 6-bit write-only register will specify which byte of screen memory will be displayed in the upper left corner of the screen. These 6 bits are the more significant bits of the start address.

R13 **Start address (low)** This is an 8-bit write-only register that forms the low-order byte of the start address.

R14 **Cursor address (high)** When combined with R15, this 6-bit read/write register will specify which byte of screen memory will have a cursor associated with it. These 6 bits are the more significant bits of the cursor address.

R15 **Cursor address (low)** This is an 8 bit read/write register that forms the least significant byte of the cursor address.

R16 **Light pen (high)** When combined with R17, this 6-bit read only register will provide a screen memory address. This address will represent a position on the screen that is associated with a light pen, or some other pointing device. The address in R16 and R17 is updated each time the light pen strobe goes from low to high. R16 represents the more significant part of the screen memory address.

R17 **Light pen (low)** This is an 8-bit read only register that provides the low order byte of the light pen address.

Firmware Listing

Section F.1 INTERFACE FIRMWARE

```
2    **********************************
3    *                                *
4    *  Ultraterm interface firmware  *
5    *  V. 1.0   2 May. 1983   12:00  *
6    *                                *
7    *        Written by D. A.        *
8    *     (C) 1983  Videx, Inc.      *
9    *                                *
10   **********************************
11   *
12   LINEHZ   EQU  60
13   C0       EQU  $C0
14   C000     EQU  C0*$100
15   *
16   *
17   * ZERO PAGE EQUATES
18   *
19   WNDWDTH  EQU  $21
20   CH       EQU  $24
21   CV       EQU  $25
22   BASL     EQU  $28
23   INVFLG   EQU  $32
24   PROMPT   EQU  $33
25   XSAVE    EQU  $35
26   CSWL     EQU  $36
27   CSWH     EQU  $37
28   KSWL     EQU  $38
29   KSWH     EQU  $39
30   A1L      EQU  $3C
31   A1H      EQU  $3D
32   A2L      EQU  $3E
33   A2H      EQU  $3F
34   A4L      EQU  $42
35   A4H      EQU  $43
36   RNDL     EQU  $4E
37   RNDH     EQU  $4F
38   *
39   * MISC EQUATES
40   *
41   STACK    EQU  $100
42   IN       EQU  $200
43   VIDWAIT  EQU  $FB78
44   APVTAB   EQU  $FC22
45   SETKBD   EQU  $FE89
46   SETVID   EQU  $FE93
47   IORTS    EQU  $FFCB
```

```
* TEMPORARIES
*
MODE     EQU  $478       ; MODE MASK FOR MODE CONTROL PORT
HEIGHT   EQU  $4F8       ; SCREEN HEIGHT
SWDTH    EQU  $578       ; SCREEN WIDTH
PWDTH    EQU  $5F8       ; PRINTED SCREEN WIDTH
OLDCHAR  EQU  $678       ; PREVIOUS CHARACTER FROM GETLN
NO       EQU  $6F8       ; SLOT * $10
TEMPX    EQU  $778       ; GENERAL TEMPORARY USAGE
MSLOT    EQU  $7F8       ; SLOT + $C0
*
* SLOT N PERMANENTS
*
BASEL    EQU  $478-C0    ; SCREEN BASE ADDRESS LOW
BASEH    EQU  $4F8-C0    ; SCREEN BASE ADDDRESS HIGH
CHORZ    EQU  $578-C0    ; CURSOR HORIZONTAL POSITION
CVERT    EQU  $5F8-C0    ; CURSOR VERTICAL POSITION
BYTE     EQU  $678-C0    ; I/O BYTE
START    EQU  $6F8-C0    ; SCREEN START ADDRESS
POFF     EQU  $778-C0    ; POWER OFF AND STATE CODE
*
BSTATE   EQU  %00000111  ; STATE CODE MASK
BPOFF    EQU  %11111000  ; POWER OFF MASK
*
FLAGS    EQU  $7F8-C0
*
BFORMT   EQU  %00000111  ; DISPLAY FORMAT MASK
BGETLN   EQU  %00010000  ; INPUT CAME FROM GETLN
BKEYBD   EQU  %00100000  ; LOWERCASE KEYBOARD AVAILABLE
BLCCON   EQU  %01000000  ; U. CASE TO L. CASE CONVERT FLAG
BINV     EQU  %10000000  ; PASCAL INVERSE FLAG
```

```
81   * APPLE IO DEVICES
82   *
83   KBD       EQU  $C000
84   KBDSTRB   EQU  $C010
85   SPKR      EQU  $C030
86   *
87   * ULTRATERM IO PORTS
88   *
89   REGSEL    EQU  $C080       ; CRTC REGISTER SELECTION PORT
90   DATA      EQU  $C081       ; CRTC REGISTER DATA PORT
91   MCREG     EQU  $C082       ; MODE CONTROL REGISTER
92   *
93   MCPBITS   EQU  %11110000   ; SELECTION FOR 0      SELECTION FOR 1
94   MPBANK    EQU  %10000000   ; READ SCREEN          ROM PAGE TWO
95   MPVIDEO   EQU  %01000000   ; APPLE VIDEO          ULTRATERM VIDEO
96   MPCLOCK   EQU  %00100000   ; 80 COLUMN CLOCK      132 COLUMN CLOCK
97   MPADDR    EQU  %00010000   ; BLOCK ADDRESS MODE  PAGE ADDRESS MODE
98   *
99   MB256     EQU  MPBANK.MPVIDEO.MPADDR
100  MB132     EQU  MPBANK.MPVIDEO.MPCLOCK
101  MBSVS     EQU  MPBANK.MPVIDEO
102  MBANK     EQU  MPBANK
103  *
104  ATTREG    EQU  $C083       ; CHARACTER ATTRIBUTE REGISTER
105  *
106  ATDFLT    EQU  %10000000   ; SELECT DEFAULT HILITE, INVERSE
107  ATINV1    EQU  %00100000   ; SELECT INVERSE FOR D7 = 1
108  ATHI1     EQU  %00010000   ; SELECT HILIGHT FOR D7 = 1
109  ATCHR     EQU  %00000100   ; SELECT LOW DENSITY CHARACTER SET
110  ATINV0    EQU  %00000010   ; SELECT INVERSE FOR D7 = 0
111  ATHI0     EQU  %00000001   ; SELECT HILIGHT FOR D7 = 0
112  *
113  ATLRG     EQU  ATDFLT
114  ATSML     EQU  ATDFLT.ATCHR
115  *
116  * ULTRATERM SCREEN MEMORY
117  *
118  DISP0     EQU  $CC00       ; PRIMARY SCREEN PAGE
119  DISP1     EQU  $CD00       ; SECONDARY SCREEN PAGE FOR BLOCK MODE
```

```
                    121           OBJ  $8000
                    122           ORG  C000
                    123  *
                    437  *
                    438           DS   C000+$100-*
                    439  *
                    440           >>>  CN00
                    440           DO   C000+$300-*
                    440           <<<
                    441           >>>  CN00
                    441           DO   C000+$300-*
                    441           <<<
                    442           >>>  CN00
                    442           DO   C000+$300-*
                    442           LST  OFF
                    442           FIN
                    442  *
                    442  * BASIC INITIAL I/O ENTRY POINT
                    442  *
                    442  ENTER
C300: 2C CB FF      442           BIT  IORTS      ; SET VFLAG IN INITIAL ENTRY
C303: 70 39         442           BVS  ENTR
                    442  INFAKE
C305: 38            442           SEC             ; FAKE INPUT ENTRY C=1
C306: 90            442           HEX  90
                    442  OUTENTR
C307: 18            442           CLC
C308: B8            442           CLV
C309: 50 33         442           BVC  ENTR
                    442  *
                    442  *
C30B: 01 87         442           HEX  0187       ; ULTRATERM IDENTITY WORD
                    442  *
                    442  * PASCAL I/O ADDRESSES AND ROUTINES
                    442  *
C30D: 17            442           DFB  INIT
C30E: 1D            442           DFB  READ
C30F: 24            442           DFB  WRITE
C310: 2A            442           DFB  STATUS
                    442  *
C311: 4C 63 C3      442           JMP  MOVE
C314: 4C B0 C3      442           JMP  XFER
                    442  *
                    442  INIT
C317: 20 00 C8      442           JSR  PINIT
C31A: A2 00         442           LDX  #$00
C31C: 60            442           RTS
                    442  *
                    442  READ
C31D: 20 42 CB      442           JSR  PREAD
C320: 29 7F         442           AND  #$7F
C322: 10 13         442           BPL  CLRX
                    442  *
                    442  WRITE
C324: 20 04 CB      442           JSR  PWRITE
C327: A2 00         442           LDX  #$00
C329: 60            442           RTS
```

```
                    442  *
                    442  STATUS
C32A: C9 00         442          CMP  #$00
C32C: F0 09         442          BEQ  CLRX
C32E: AD 00 C0      442          LDA  KBD
C331: 0A            442          ASL
C332: 90 03         442          BCC  CLRX
C334: 20 9C CA      442          JSR  KEYSTAT
                    442  CLRX
C337: A2 00         442          LDX  #$00
C339: 60            442          RTS
                    442  *
                    442  * BASIC INPUT ENTRY POINT
                    442  *
                    442  INENTR
C33A: 91 28         442          STA  (BASL),Y   ; REPLACE FLASHING CURSOR
C33C: 38            442          SEC
C33D: B8            442          CLV
                    442  ENTR
C33E: 8D F8 05      442          STA  PWDTH      ; SAVE CHARACTER
C341: 86 35         442          STX  XSAVE      ; SAVE INPUT BUFFER INDEX
C343: 48            442          PHA             ; SAVE REGISTERS ON STACK
C344: 8A            442          TXA
C345: 48            442          PHA
C346: 98            442          TYA
C347: 48            442          PHA
C348: AD F8 05      442          LDA  PWDTH      ; RETRIEVE CHARACTER
C34B: 48            442          PHA             ; PUSH IT ON STACK
C34C: AD FF CF      442          LDA  $CFFF      ; TURN OFF CO-RESIDENT MEMORY
C34F: A2 C3         442          LDX  #>ENTER    ; ESTABLISH INDEX VALUES
C351: A0 30         442          LDY  #>ENTER*$10
C353: 50 03         442          BVC  IO         ; DO I/O IF NOT INITIAL ENTRY
C355: 4C EB C3      442          JMP  BINIT
                    442  *
                    442  IO
C358: B0 03         442          BCS  INPUT      ; DO INPUT IF CARRY SET
C35A: 4C FA C3      442          JMP  OUTPUT
                    442  *
                    442  INPUT
C35D: 20 1A C8      442          JSR  BSTART     ; SETUP TEMPORARIES
C360: 4C AD C9      442          JMP  BASINP     ; DO INPUT
```

```
                    442  * APPLE //e MOVE ROUTINE
                    442  *
                    442  MOVE
C363: 48            442          PHA
C364: 98            442          TYA
C365: 48            442          PHA
C366: AD 13 C0      442          LDA  $C013
C369: 48            442          PHA
C36A: AD 14 C0      442          LDA  $C014
C36D: 48            442          PHA
                    442  *
C36E: 90 08         442          BCC  MOVEC2M
C370: 8D 02 C0      442          STA  $C002
C373: 8D 05 C0      442          STA  $C005
C376: B0 06         442          BCS  MOVESTRT
                    442  *
                    442  MOVEC2M
C378: 8D 04 C0      442          STA  $C004
C37B: 8D 03 C0      442          STA  $C003
                    442  *
                    442  MOVESTRT
C37E: A0 00         442          LDY  #$00
                    442  *
                    442  MOVELOOP
C380: B1 3C         442          LDA  (A1L),Y
C382: 91 42         442          STA  (A4L),Y
C384: E6 42         442          INC  A4L
C386: D0 02         442          BNE  NXTA1
C388: E6 43         442          INC  A4H
                    442  NXTA1
C38A: A5 3C         442          LDA  A1L
C38C: C5 3E         442          CMP  A2L
C38E: A5 3D         442          LDA  A1H
C390: E5 3F         442          SBC  A2H
C392: E6 3C         442          INC  A1L
C394: D0 02         442          BNE  C01
C396: E6 3D         442          INC  A1H
                    442  C01
C398: 90 E6         442          BCC  MOVELOOP
                    442  *
C39A: 8D 04 C0      442          STA  $C004
C39D: 68            442          PLA
C39E: 10 03         442          BPL  C03
C3A0: 8D 05 C0      442          STA  $C005
                    442  C03
C3A3: 8D 02 C0      442          STA  $C002
C3A6: 68            442          PLA
C3A7: 10 03         442          BPL  MOVERET
C3A9: 8D 03 C0      442          STA  $C003
                    442  MOVERET
C3AC: 68            442          PLA
C3AD: A8            442          TAY
C3AE: 68            442          PLA
C3AF: 60            442          RTS
```

```
                    442  * APPLE //e XFER ROUTINE
                    442  *
                    442  XFER
C3B0: 48            442          PHA
C3B1: AD ED 03      442          LDA  $03ED
C3B4: 48            442          PHA
C3B5: AD EE 03      442          LDA  $03EE
C3B8: 48            442          PHA
                    442  *
C3B9: 90 0A         442          BCC  XFERC2M
C3BB: 8D 03 C0      442          STA  $C003
C3BE: 8D 05 C0      442          STA  $C005
C3C1: 50 19         442          BVC  XFERSZP
C3C3: 70 08         442          BVS  XFERAZP
                    442  XFERC2M
C3C5: 8D 02 C0      442          STA  $C002
C3C8: 8D 04 C0      442          STA  $C004
C3CB: 50 0F         442          BVC  XFERSZP
                    442  *
                    442  XFERAZP
C3CD: 68            442          PLA
C3CE: 8D EE 03      442          STA  $03EE
C3D1: 68            442          PLA
C3D2: 8D ED 03      442          STA  $03ED
C3D5: 68            442          PLA
C3D6: 8D 09 C0      442          STA  $C009
C3D9: 6C ED 03      442          JMP  ($03ED)
                    442  *
                    442  XFERSZP
C3DC: 68            442          PLA
C3DD: 8D EE 03      442          STA  $03EE
C3E0: 68            442          PLA
C3E1: 8D ED 03      442          STA  $03ED
C3E4: 68            442          PLA
C3E5: 8D 08 C0      442          STA  $C008
C3E8: 6C ED 03      442          JMP  ($03ED)
```

```
                    442  * BASIC INITIALIZE
                    442  *
                    442  BINIT
C3EB: A9 3A         442           LDA  #INENTR     ; INIT INPUT ENTRY POINT
C3ED: 85 38         442           STA  KSWL
C3EF: 86 39         442           STX  KSWH
C3F1: A9 07         442           LDA  #OUTENTR    ; INIT OUTPUT ENTRY POINT
C3F3: 85 36         442           STA  CSWL
C3F5: 86 37         442           STX  CSWH
C3F7: 20 00 C8      442           JSR  PINIT       ; INIT PERMANENTS AND CRTC
                    442  *
                    442  *
                    442  OUTPUT
C3FA: 20 1A C8      442           JSR  BSTART      ; SETUP TEMPORARIES
C3FD: 4C 15 CA      442           JMP  BASOUT      ; OUTPUT CHARACTER
                    442           <<<
                    443           >>>  CN00
                    443           DO   C000+$300-*
                    443           <<<
                    444           >>>  CN00
                    444           DO   C000+$300-*
                    444           <<<
                    445           >>>  CN00
                    445           DO   C000+$300-*
                    445           <<<
                    446           >>>  CN00
                    446           DO   C000+$300-*
                    446           <<<
```

```
                    448  * CO-RESIDENT ROM CODE
                    449  *
                    450  *
                    451  PINIT
C800: 38            452          SEC
C801: 90            453          HEX  90
                    454  FINIT
C802: 18            455          CLC
C803: 2C 58 C8      456          BIT  RTSO
C806: 20 1C C8      457          JSR  IENTER1
                    458  EXIT
C809: AD 78 04      459          LDA  MODE
C80C: 29 7F         460          AND  #MBANK!$FF
C80E: AC F8 06      461          LDY  NO
C811: 99 82 C0      462          STA  MCREG,Y
C814: 8D 78 04      463          STA  MODE
                    464  IEXIT
C817: 60            465          RTS
                    466  *
                    467  PSTART
C818: 38            468          SEC
C819: 90            469          HEX  90
                    470  *
                    471  BSTART
C81A: 18            472          CLC
C81B: B8            473          CLV
                    474  *
                    475  IENTER1
C81C: 8C F8 06      476          STY  NO          ; ESTABLISH NO
C81F: 8E F8 07      477          STX  MSLOT       ; ESTABLISH MSLOT
C822: 70 03         478          BVS  SETUP       ; OVERFLOW SET UPON INITIALIZE
C824: 4C CF C8      479          JMP  CSTART      ; GO TO COMMON START ROUTINE
                    480  *
                    481  SETUP
C827: B0 08         482          BCS  SETUP1      ; CARRY CLEAR IF FORMAT INIT
C829: A9 30         483          LDA  #$30        ; SET POWER OFF BYTE
C82B: 9D B8 06      484          STA  POFF,X
C82E: 4C 6F C8      485          JMP  NEWFMT      ; SET FORMAT
                    486  *
                    487  SETUP1
C831: BD B8 06      488          LDA  POFF,X      ; GET POWER OFF FLAG
C834: 29 F8         489          AND  #BPOFF      ; STRIP OFF STATE CODE
C836: 49 30         490          EOR  #$30        ; HAS POWER BEEN TURNED OFF?
C838: C9 01         491          CMP  #$01
C83A: B8            492          CLV
C83B: 70            493          HEX  70
                    494  RESTART
C83C: 38            495          SEC
C83D: A9 30         496          LDA  #$30        ; CLEAR POWER OFF BYTE
C83F: 9D B8 06      497          STA  POFF,X
C842: 90 03         498          BCC  IEXIT       ; CARRY SET IF INIT IS NEEDED
```

```
                    500  * CHECK FOR APPLE //e
                    501  *
C844: A0 20         502          LDY  #BKEYBD
C846: A9 A5         503          LDA  #$A5        ; PICK AN UNLIKELY VALUE
C848: 48            504          PHA              ; PUSH IT ON THE STACK
C849: 4C 50 C8      505          JMP  RDSKIP
                    506  *
                    507          DS   C000+$84D-*
C84D: 4C 42 CB      508          JMP  PREAD
                    509  *
                    510  RDSKIP
C850: 8D 09 C0      511          STA  $C009       ; SWITCH TO STACK 2 (APPLE //e)
C853: 68            512          PLA              ; RETRIEVE VALUE
C854: 8D 08 C0      513          STA  $C008       ; RETURN TO STACK 1 (APPLE //e)
C857: C9 A5         514          CMP  #$A5        ; IF DIFFERENT THEN IT
C859: D0 10         515          BNE  A2E         ; IS AN APPLE //e.
C85B: A9 5A         516          LDA  #$5A        ; PICK ANOTHER UNLIKELY VALUE
C85D: 48            517          PHA              ; PUSH IT ON THE STACK
C85E: 8D 09 C0      518          STA  $C009       ; SWITCH TO STACK 2 (APPLE //e)
C861: 68            519          PLA              ; RETRIEVE VALUE
C862: 8D 08 C0      520          STA  $C008       ; RETURN TO STACK 1 (APPLE //e)
C865: C9 5A         521          CMP  #$5A        ; IF THE SAME THEN ASSUME AN
C867: D0 02         522          BNE  A2E         ; APPLE ][ OR APPLE ][+
C869: A0 00         523          LDY  #$00        ; DON'T SET LOWERCASE KEYBOARD
                    524  *
                    525  A2E
C86B: 98            526          TYA              ; SAVE KEYBOARD MODE
C86C: 9D 38 07      527          STA  FLAGS,X
                    528  NEWFMT
C86F: 20 DA C8      529          JSR  NEWFMT2
C872: A9 00         530          LDA  #$00        ; CLEAR PERMANENTS
C874: 9D B8 03      531          STA  BASEL,X
C877: 9D 38 04      532          STA  BASEH,X
C87A: 9D B8 04      533          STA  CHORZ,X
C87D: 9D 38 05      534          STA  CVERT,X
C880: 9D 38 06      535          STA  START,X
```

```
                    537  * CLEAR SCREEN MEMORY
                    538  *
C883: A2 0F         539          LDX  #$0F        ; CLEAR $10 PAGES
C885: A0 00         540          LDY  #$00
                    541  CLOOP1
C887: 98            542          TYA
C888: 48            543          PHA
C889: 8A            544          TXA              ; PUT PAGE NUMBER IN A
C88A: 0D 78 04      545          ORA  MODE        ; OR IN MODE MASK
C88D: 09 D0         546          ORA  #MB256      ; USE PAGE ADDRESSING
C88F: AC F8 06      547          LDY  NO
C892: 99 82 C0      548          STA  MCREG,Y     ; SELECT PAGE
C895: 68            549          PLA
C896: A8            550          TAY
C897: A9 20         551          LDA  #$20        ; USE NORMAL SPACE
                    552  CSLOOP1
C899: 99 00 CC      553          STA  $CC00,Y     ; CLEAR ENTIRE PAGE
C89C: C8            554          INY
C89D: D0 FA         555          BNE  CSLOOP1
C89F: CA            556          DEX
C8A0: 10 E5         557          BPL  CLOOP1      ; NEXT PAGE
C8A2: AE F8 07      558          LDX  MSLOT       ; RESTORE X
C8A5: AC F8 06      559          LDY  NO          ; GET DEVICE INDEX
C8A8: AD 78 04      560          LDA  MODE        ; RESTORE ADDRESS MODE
C8AB: 99 82 C0      561          STA  MCREG,Y
C8AE: BD 38 07      562          LDA  FLAGS,X     ; COMPUTE FORMAT TABLE INDEX
C8B1: 29 07         563          AND  #BFORMT
C8B3: 0A            564          ASL
C8B4: 0A            565          ASL
C8B5: 0A            566          ASL
C8B6: 0A            567          ASL
C8B7: AA            568          TAX
C8B8: 29 0F         569          AND  #$0F        ; USE LOWER FOUR BITS
                    570  LOOP
C8BA: 99 80 C0      571          STA  REGSEL,Y    ; FOR THE CRTC ADDRESS
C8BD: BD BA CE      572          LDA  TABLE,X     ; GET THE PARAMETER
C8C0: 99 81 C0      573          STA  DATA,Y      ; STORE INTO THE CRTC
C8C3: E8            574          INX
C8C4: 8A            575          TXA
C8C5: 29 0F         576          AND  #$0F
C8C7: D0 F1         577          BNE  LOOP        ; LOOP UNTIL DONE
C8C9: AE F8 07      578          LDX  MSLOT       ; RESTORE X REGISTER
C8CC: 4C AC CB      579          JMP  HOME        ; HOME CURSOR
```

```
                        581  CSTART
C8CF: 90 41             582          BCC   BSTART1     ; DO BASIC HOME TEST
C8D1: BD 38 07          583          LDA   FLAGS,X     ; SET INVERSE FLAG FOR PASCAL
C8D4: 49 80             584          EOR   #$80
C8D6: 85 32             585          STA   INVFLG
                        586  TINIT
C8D8: 38                587          SEC
C8D9: 90                588          HEX   90
                        589  NEWFMT2
C8DA: 18                590          CLC
C8DB: AC F8 06          591          LDY   NO          ; GET FORMAT NUMBER FOR INDEX
C8DE: BD 38 07          592          LDA   FLAGS,X
C8E1: 29 07             593          AND   #BFORMT
C8E3: AA                594          TAX
C8E4: BD 24 C9          595          LDA   MODTBL,X    ; SET MODE MASK
C8E7: 8D 78 04          596          STA   MODE
C8EA: 99 82 C0          597          STA   MCREG,Y     ; ASSURE PROPER MODE
C8ED: BD 2C C9          598          LDA   HGTBL,X     ; SET HEIGHT
C8F0: 8D F8 04          599          STA   HEIGHT
C8F3: BD 34 C9          600          LDA   SWDTBL,X    ; SET SCREEN WIDTH
C8F6: 8D 78 05          601          STA   SWDTH
C8F9: BD 3C C9          602          LDA   PWDTBL,X    ; SET PRINTED WIDTH
C8FC: 8D F8 05          603          STA   PWDTH
C8FF: A9 29             604          LDA   #41         ; ADJUST WINDOW FOR HOME DETECT
C901: 85 21             605          STA   WNDWDTH
C903: B0 09             606          BCS   NFSKIP      ; EXIT IF START SETUP
C905: BD 44 C9          607          LDA   ATRTBL,X    ; INITIALIZE ATTRIBUTES
C908: 99 83 C0          608          STA   ATTREG,Y
C90B: AD 78 04          609          LDA   MODE
                        610  NFSKIP
C90E: AE F8 07          611          LDX   MSLOT       ; RECOVER X
C911: 60                612          RTS
                        613  BSTART1
C912: A9 A0             614          LDA   #$A0        ; IF TWO TEMPORARIES ARE SPACES
C914: CD F8 04          615          CMP   HEIGHT      ; THEN HOME HAS OCCURED
C917: D0 BF             616          BNE   TINIT
C919: CD 78 05          617          CMP   SWDTH
C91C: D0 BA             618          BNE   TINIT
C91E: 20 D8 C8          619          JSR   TINIT
C921: 4C AA CB          620          JMP   CLSCRN      ; CLEAR THE SCREEN
```

```
                622  MODTBL
C924: C0        623          DFB  M8SVS
C925: D0        624          DFB  M8256
C926: F0        625          DFB  M8256,MB132
C927: D0        626          DFB  M8256
C928: D0        627          DFB  M8256
C929: D0        628          DFB  M8256
C92A: F0        629          DFB  M8256,MB132
C92B: F0        630          DFB  M8256,MB132
                631  *
                632  HCTBL
C92C: 18        633          DFB  24
C92D: 18        634          DFB  24
C92E: 18        635          DFB  24
C92F: 18        636          DFB  24
C930: 20        637          DFB  32
C931: 30        638          DFB  48
C932: 18        639          DFB  24
C933: 20        640          DFB  32
                641  *
                642  SWDTBL
C934: 50        643          DFB  80
C935: 60        644          DFB  96
C936: A0        645          DFB  160
C937: 50        646          DFB  80
C938: 50        647          DFB  80
C939: 50        648          DFB  80
C93A: A0        649          DFB  160
C93B: 80        650          DFB  128
                651  *
                652  PWDTBL
C93C: 50        653          DFB  80
C93D: 60        654          DFB  96
C93E: A0        655          DFB  160
C93F: 50        656          DFB  80
C940: 50        657          DFB  80
C941: 50        658          DFB  80
C942: 84        659          DFB  132
C943: 80        660          DFB  128
                661  *
                662  ATRTBL
C944: 84        663          DFB  ATSML
C945: 84        664          DFB  ATSML
C946: 84        665          DFB  ATSML
C947: 80        666          DFB  ATLRC
C948: 80        667          DFB  ATLRC
C949: 84        668          DFB  ATSML
C94A: 80        669          DFB  ATLRG
C94B: 80        670          DFB  ATLRG
```

```
                        672  RDSCRN
C94C: AD 78 04          673          LDA  MODE        ; SET MODE TO SELECT READ SCREEN
C94F: 29 7F             674          AND  #$7F
C951: 8D 78 04          675          STA  MODE
C954: BC B8 04          676          LDY  CHORZ,X     ; COMPUTE SCREEN ADDRESS
C957: 20 78 C9          677          JSR  PAGSEL
C95A: BD 00 CC          678          LDA  DISP0,X     ; READ SCREEN
C95D: 90 03             679          BCC  RSKIP1
C95F: BD 00 CD          680          LDA  DISP1,X
                        681  RSKIP1
C962: AA                682          TAX
C963: AD 78 04          683          LDA  MODE        ; FIX MODE
C966: 09 80             684          ORA  #MBANK
C968: 8D 78 04          685          STA  MODE
C96B: AC F8 06          686          LDY  NO          ; FIX MODE CONTROL PORT
C96E: 99 82 C0          687          STA  MCREG,Y
C971: 8A                688          TXA
C972: AE F8 07          689          LDX  MSLOT       ; RESTORE X REGISTER
C975: 09 80             690          ORA  #$80
C977: 60                691          RTS
                        692  *
                        693  PAGSEL
C978: 18                694          CLC
C979: 98                695          TYA              ; ADD CHORZ TO BASE ADDRESS
C97A: 48                696          PHA
C97B: 7D B8 03          697          ADC  BASEL,X
C97E: 48                698          PHA              ; SAVE SCREEN ADDRESS LOW
C97F: A9 00             699          LDA  #$00
C981: 7D 38 04          700          ADC  BASEH,X
C984: 29 0F             701          AND  #MCPBITS!$FF ; SELECT SCREEN PAGE
C986: 0D 78 04          702          ORA  MODE        ; FOR PAGE ADDRESSING MODE
C989: AC F8 06          703          LDY  NO
C98C: 99 82 C0          704          STA  MCREG,Y
C98F: 48                705          PHA
C990: 0A                706          ASL
C991: 29 0C             707          AND  #$0C        ; SELECT SCREEN BLOCK
C993: 0D F8 06          708          ORA  NO          ; FOR BLOCK ADDRESSING MODE
C996: AA                709          TAX
C997: BD 80 C0          710          LDA  REGSEL,X
C99A: 68                711          PLA
C99B: 4A                712          LSR              ; PUT BIT 8 OF ADDRESS IN CARRY
C99C: 68                713          PLA
C99D: AA                714          TAX              ; PUT ADDRESS LOW IN X
C99E: 68                715          PLA
C99F: A8                716          TAY              ; RESTORE Y
C9A0: 60                717          RTS
                        718  *
                        719  *
                        720          DS   C000+$9AA-*
C9AA: 4C 07 CB          721          JMP  PWRITE1
```

```
                        723  BASINP
C9AD: 68                724          PLA             ; POP STACK
C9AE: 20 17 CE          725          JSR  FIXCSR     ; ADJUST CURSOR FOR GET STATEMENTS
C9B1: A4 35             726          LDY  XSAVE      ; GET INPUT BUFFER INDEX
C9B3: F0 2B             727          BEQ  GETLN      ; IF ZERO ASSUME GETLN
C9B5: 88                728          DEY
C9B6: AD 78 06          729          LDA  OLDCHAR    ; GET LAST CHARACTER FROM GETLN
C9B9: C9 88             730          CMP  #$88       ; IF BS ASSUME GETLN
C9BB: F0 23             731          BEQ  GETLN
C9BD: D9 00 02          732          CMP  IN,Y       ; IF SAME AS CHARACTER IN INPUT
C9C0: F0 1E             733          BEQ  GETLN      ; BUFFER THEN ASSUME GETLN
C9C2: 20 79 CB          734          JSR  CAPSLK     ; CHECK AS UPPERCASE ALSO
C9C5: D9 00 02          735          CMP  IN,Y
C9C8: D0 35             736          BNE  NTGETLN
C9CA: AD 78 06          737          LDA  OLDCHAR    ; GET LAST CHARACTER FROM GETLN
C9CD: 99 00 02          738          STA  IN,Y       ; FIX INPUT BUFFER
C9D0: B0 0E             739          BGE  GETLN      ; GO TO GETLN
                        740  ESC
C9D2: 20 B9 C0          741          JSR  ESCNEW     ; PERFORM ESCAPE FUNCTION
C9D5: A9 C0             742          LDA  #MBSVS     ; WAS IT AN EXIT COMMAND?
C9D7: 2D 78 04          743          AND  MODE
C9DA: D0 04             744          BNE  GETLN      ; NO, CONTINUE READING CHARACTERS
C9DC: A9 98             745          LDA  #$98       ; YES, RETURN A CONTROL X
C9DE: D0 1C             746          BNE  NOTPICK
                        747  GETLN
C9E0: A9 10             748          LDA  #BGETLN    ; SET GETLN FLAG
C9E2: 20 98 CB          749          JSR  FLGSET
C9E5: 20 59 CB          750          JSR  RDKEY      ; GET CHARACTER FROM KEYBOARD
C9E8: C9 9B             751          CMP  #$9B       ; CHECK FOR ESCAPE
C9EA: F0 E6             752          BEQ  ESC
C9EC: C9 8D             753          CMP  #$8D       ; CHECK FOR CR
C9EE: D0 05             754          BNE  NOTCR
C9F0: 48                755          PHA             ; FIX INPUT BUFFER FOR MIXED
C9F1: 20 80 CC          756          JSR  FIXBUF     ; UPPERCASE AND LOWERCASE
C9F4: 68                757          PLA
                        758  NOTCR
C9F5: C9 95             759          CMP  #$95       ; CHECK FOR PICK
C9F7: D0 03             760          BNE  NOTPICK
C9F9: 20 4C C9          761          JSR  RDSCRN     ; READ THE SCREEN
                        762  NOTPICK
C9FC: A8                763          TAY             ; SAVE CHARACTER IN OLDCHAR
C9FD: D0 05             764          BNE  SAVOLD
                        765  NTGETLN
C9FF: 20 59 CB          766          JSR  RDKEY      ; GET CHARACTER FROM KEYBOARD
CA02: A0 00             767          LDY  #$00       ; CLEAR OLDCHAR
                        768  SAVOLD
CA04: 8C 78 06          769          STY  OLDCHAR
CA07: BA                770          TSX             ; PUT CHARACTER INTO STACK
CA08: E8                771          INX
CA09: E8                772          INX
CA0A: E8                773          INX
CA0B: 9D 00 01          774          STA  $100,X
CA0E: AE F8 07          775          LDX  MSLOT      ; RECOVER X
CA11: A0 00             776          LDY  #$00       ; SET CH = 0
CA13: F0 58             777          BEQ  SETCH
```

```
                        779  BASOUT
CA15: 8D 38 07          780          LDA  FLAGS,X     ; CHECK GETLN FLAG
CA18: 29 10             781          AND  #$GETLN
CA1A: C9 10             782          CMP  #$GETLN
CA1C: 68                783          PLA
CA1D: 90 08             784          BLT  BOUT        ; IF CLEAR THEN SKIP
CA1F: AC 78 06          785          LDY  OLDCHAR     ; GET LAST INPUT CHARACTER
CA22: C0 E0             786          CPY  #$E0        ; IF IT IS LOWERCASE THEN USE IT
CA24: 90 01             787          BLT  BOUT
CA26: 98                788          TYA
                        789  BOUT
CA27: 9D B8 05          790          STA  BYTE,X      ; SAVE CHARACTER IN BYTE
CA2A: 20 17 CE          791          JSR  FIXCSR      ; ADJUST CURSOR POSITION
CA2D: 20 06 CE          792          JSR  OUTPT1      ; OUTPUT CHARACTER
CA30: E4 39             793          CPX  KSWH        ; IF INPUT HOOK ISN'T CONNECTED
CA32: F0 06             794          BEQ  NOCSR       ; THEN PUT A CURSOR ON THE SCREEN
CA34: 20 10 CB          795          JSR  CSRMOV
CA37: 20 50 CE          796          JSR  FIXWDTH
                        797  NOCSR
CA3A: A9 EF             798          LDA  #$GETLN!$FF ; CLEAR THE GETLN FLAG
CA3C: 20 A3 CB          799          JSR  FLGCLR
CA3F: BD B8 05          800          LDA  BYTE,X
CA42: C9 8D             801          CMP  #$8D        ; WAS IT A CR?
CA44: D0 18             802          BNE  LSTFIX      ; NO, DO NOT STOP LISTING
CA46: AC 00 C0          803          LDY  KBD         ; HAS CONTROL S BEEN STRUCK?
CA49: 10 13             804          BPL  LSTFIX
CA4B: C0 93             805          CPY  #$93
CA4D: D0 0F             806          BNE  LSTFIX      ; NO, DO NOT STOP LISTING
CA4F: 2C 10 C0          807          BIT  KBDSTRB     ; CLEAR KEYBOARD STROBE
                        808  KBDWAIT
CA52: AC 00 C0          809          LDY  KBD
CA55: 10 FB             810          BPL  KBDWAIT     ; WAIT UNTIL NEXT KEY TO RESUME
CA57: C0 83             811          CPY  #$83        ; IS IT CONTROL C?
CA59: F0 03             812          BEQ  LSTFIX      ; IF SO, THEN RETURN WITH IT
CA5B: 2C 10 C0          813          BIT  KBDSTRB     ; IF NOT, CLEAR KEYBOARD STROBE
                        814  LSTFIX
CA5E: A0 00             815          LDY  #$00        ; START WITH ZERO
CA60: BD B8 04          816          LDA  CHORZ,X     ; GET CURSOR HORIZONTAL
CA63: ED F8 05          817          SBC  PWDTH       ; WITHIN 8 CHARACTERS OF RIGHT?
CA66: C9 F8             818          CMP  #$F8
CA68: 90 03             819          BCC  SETCH       ; NO, PUT ZERO IN CH
CA6A: 69 27             820          ADC  #$27        ; YES, ADJUST CH FOR LISTINGS
CA6C: A8                821          TAY
                        822  SETCH
CA6D: 84 24             823          STY  CH          ; SAVE NEW CH
CA6F: A9 C0             824          LDA  #M8SVS
CA71: 2D 78 04          825          AND  MODE        ; SWITCH TO 40 COLUMNS?
CA74: D0 15             826          BNE  NORMOUT     ; NO, SKIP TO EXIT
CA76: A9 17             827          LDA  #23         ; FIX CURSOR VERTICAL
CA78: C5 25             828          CMP  CV
CA7A: B0 03             829          BCS  VSKIP
CA7C: 9D 38 05          830          STA  CVERT,X
                        831  VSKIP
CA7F: 20 93 FE          832          JSR  SETVID      ; PR#0
CA82: 20 89 FE          833          JSR  SETKBD      ; IN#0
CA85: 20 22 FC          834          JSR  APVTAB      ; VTAB
CA88: 20 50 CE          835          JSR  FIXWDTH     ; FIX WINDOW WIDTH
```

```
                        837  NORMOUT
CA8B: 20 09 C8          838          JSR  EXIT        ; SWITCH TO 40 COLUMNS
CA8E: AE F8 07          839          LDX  MSLOT       ; RECOVER X
CA91: BD 38 05          840          LDA  CVERT,X     ; SET CV = CVERT
CA94: 85 25             841          STA  CV
CA96: 68                842          PLA              ; RECOVER REGISTERS
CA97: A8                843          TAY
CA98: 68                844          PLA
CA99: AA                845          TAX
CA9A: 68                846          PLA
CA9B: 60                847          RTS
```

```
                        849  KEYSTAT
CA9C: 48                850          PHA             ; SAVE KEY
CA9D: C9 E0             851          CMP  #$E0       ; IF LOWERCASE THEN SET
CA9F: 90 08             852          BLT  NOTLWR     ; THE LOWERCASE KEYBOARD FLAG
CAA1: A9 20             853          LDA  #BKEYBD
CAA3: 1D 38 07          854          ORA  FLAGS,X
CAA6: 9D 38 07          855          STA  FLAGS,X
                        856  NOTLWR
CAA9: BD 38 07          857          LDA  FLAGS,X    ; IF LOWERKEYBOARD FLAG IS SET
CAAC: 29 20             858          AND  #BKEYBD    ; THEN ACCEPT KEY UNMODIFIED
CAAE: C9 20             859          CMP  #BKEYBD
CAB0: 68                860          PLA             ; RECOVER KEY
CAB1: B0 3F             861          BCS  INDONE
CAB3: C9 8B             862          CMP  #$8B       ; CHECK FOR CONTROL K
CAB5: D0 02             863          BNE  NOTK
CAB7: A9 DB             864          LDA  #$DB       ; SUBSTITUTE A RIGHT BRACKET
                        865  NOTK
CAB9: C9 81             866          CMP  #$81       ; CHECK FOR CONTROL A
CABB: D0 0D             867          BNE  NTSHFT
CABD: BD 38 07          868          LDA  FLAGS,X    ; TOGGLE UPR/LWR CONVERT FLAG
CAC0: 49 40             869          EOR  #BLCCON
CAC2: 9D 38 07          870          STA  FLAGS,X
CAC5: 2C 10 C0          871          BIT  KBDSTRB    ; CLEAR KEYBOARD STROBE
CAC8: 18                872          CLC             ; REJECT KEY
CAC9: 60                873          RTS
                        874  NTSHFT
CACA: 48                875          PHA             ; SAVE KEY
CACB: BD 38 07          876          LDA  FLAGS,X    ; CHECK UPR/LWR CONVERT FLAG
CACE: 0A                877          ASL
CACF: 0A                878          ASL
CAD0: 68                879          PLA             ; RESTORE CHARACTER
CAD1: 90 1F             880          BCC  INDONE     ; DON'T CONVERT IF FLAG CLEAR
CAD3: C9 B0             881          CMP  #$B0
CAD5: 90 1B             882          BLT  INDONE     ; CONVERT ONLY ALPHA KEYS
CAD7: 2C 63 C0          883          BIT  $C063
CADA: 30 14             884          BMI  NOSHIFT    ; SHIFT KEY UP, SEND AS LOWERCASE
CADC: C9 B0             885          CMP  #"0"       ; ZERO BECOMES "}"
CADE: F0 0E             886          BEQ  ZERO
CAE0: C9 C0             887          CMP  #"@"       ; @ BECOMES "P"
CAE2: D0 02             888          BNE  NOT@
CAE4: A9 D0             889          LDA  #"P"
                        890  NOT@
CAE6: C9 DB             891          CMP  #"["       ; [ \ ] ^ _ BECOME
CAE8: 90 08             892          BLT  INDONE
CAEA: 29 CF             893          AND  #$CF       ; K L M N O
CAEC: D0 04             894          BNE  INDONE
                        895  ZERO
CAEE: A9 DD             896          LDA  #"]"
                        897  NOSHIFT
CAF0: 09 20             898          ORA  #$20       ; CONVERT TO LOWERCASE
                        899  INDONE
CAF2: 48                900          PHA             ; DUPLICATE KEY
CAF3: 29 7F             901          AND  #$7F       ; STRIP OFF HIGH BIT
CAF5: 9D B8 05          902          STA  BYTE,X     ; SAVE FOR PASCAL
CAF8: 68                903          PLA             ; RECOVER FOR BASIC
CAF9: 38                904          SEC             ; ACCEPT KEY
```

```
                        906  RTS9
CAFA: 60                907          RTS
                        908  *
                        909  *
                        910          DS    C000+$B00-*
CB00: 2C CB FF
CB03: 70                911          HEX   2CCBFF70
                        912  *
                        913  *
                        914  PWRITE
CB04: 9D B8 05          915          STA   BYTE,X
                        916  PWRITE1
CB07: 20 18 C8          917          JSR   PSTART      ; CALCULATE TEMPORARIES
CB0A: 20 06 CE          918          JSR   OUTPT1
CB0D: 20 09 C8          919          JSR   EXIT
                        920  CSRMOV
CB10: AC F8 06          921          LDY   NO          ; GET DEVICE INDEX
CB13: A9 0F             922          LDA   #$0F        ; SELECT          FOR
CB15: 99 80 C0          923          STA   REGSEL,Y    ; CURSOR ADDRESS LOW
CB18: BD B8 04          924          LDA   CHORZ,X     ; CALCULATE ADDRESS
CB1B: 4C 27 CB          925          JMP   SSKIP
                        926  *
                        927          DS    C000+$B1E-*
                        928  SHUTUP
CB1E: A9 FF             929          LDA   #$FF
CB20: 8D FF CF          930          STA   $CFFF
CB23: 60                931          RTS
CB24: 4C 1E CB          932          JMP   SHUTUP
                        933  *
                        934  SSKIP
CB27: CD F8 05          935          CMP   PWDTH
CB2A: B0 15             936          BCS   RTS6
CB2C: 7D B8 03          937          ADC   BASEL,X
CB2F: 99 81 C0          938          STA   DATA,Y      ; SAVE ADDRESS
CB32: A9 0E             939          LDA   #$0E        ; SELECT REGISTER FOR
CB34: 99 80 C0          940          STA   REGSEL,Y    ; CURSOR ADDRESS HIGH
CB37: A9 00             941          LDA   #$00
CB39: 7D 38 04          942          ADC   BASEH,X
CB3C: 29 1F             943          AND   #$1F
CB3E: 99 81 C0          944          STA   DATA,Y      ; SAVE ADDRESS
                        945  RTS6
CB41: 60                946          RTS
                        947  *
                        948  *
                        949  PREAD
CB42: 20 18 C8          950          JSR   PSTART      ; CALCULATE TEMPORARIES
                        951  KEYIN
CB45: E6 4E             952          INC   RNDL        ; UPDATE BASIC RANDOM NUMBER
CB47: D0 02             953          BNE   KEYIN2
CB49: E6 4F             954          INC   RNDH
                        955  KEYIN2
CB4B: AD 00 C0          956          LDA   KBD         ; POLL KEYBOARD
CB4E: 10 F5             957          BPL   KEYIN       ; LOOP UNTIL KEY IS STRUCK
CB50: 20 9C CA          958          JSR   KEYSTAT     ; CHECK STATUS AND CONVERT KEY
CB53: 90 F0             959          BCC   KEYIN       ; REJECTED, TRY AGAIN
CB55: 2C 10 C0          960          BIT   KBDSTRB     ; CLEAR KEYBOARD STROBE
                        961  RTS0
```

```
CB58: 60          962             RTS
                  963  RDKEY
CB59: 20 10 CB    964             JSR  CSRMOV      ; PUT CURSOR ON SCREEN
CB5C: 20 45 CB    965             JSR  KEYIN       ; GET KEY FROM KEYBOARD
CB5F: 48          966             PHA
CB60: A9 0E       967             LDA  #$0E        ; REMOVE CURSOR
CB62: AC F8 06    968             LDY  NO
CB65: 99 80 C0    969             STA  REGSEL,Y
CB68: A9 FF       970             LDA  #$FF
CB6A: 99 81 C0    971             STA  DATA,Y
CB6D: 68          972             PLA
CB6E: 60          973             RTS
                  974  *
                  975  CHRPUT
CB6F: B0 04       976             BCS  WSKIP       ; WRITE TO SECOND PAGE IF C=1
CB71: 9D 00 CC    977             STA  DISP0,X     ; PUT CHARACTER IN SCREEN MEMORY
CB74: 60          978             RTS
                  979  WSKIP
CB75: 9D 00 CD    980             STA  DISP1,X     ; PUT CHARACTER IN SCREEN MEMORY
CB78: 60          981             RTS
                  982  *
                  983  CAPSLK
CB79: C9 E0       984             CMP  #$E0        ; IF LOWER CASE CHARACTER
CB7B: 90 02       985             BLT  RTS5
CB7D: 29 DF       986             AND  #$DF        ; CONVERRT IT TO UPPERCASE
                  987  RTS5
CB7F: 60          988             RTS
```

```
                        990  *********************
                        991  *                   *
                        992  *  GO TO XY         *
                        993  *  OLD LEAD IN      *
                        994  *  RAW MODE         *
                        995  *  VIDEO FORMAT     *
                        996  *  CHARACTER ATTR.  *
                        997  *                   *
                        998  *********************
                        999  *
                       1000 GOXY
CB80: A9 31             1001          LDA  #$31
CB82: 2C                1002          HEX  2C
                       1003 LEADIN
CB83: A9 33             1004          LDA  #$33
CB85: 2C                1005          HEX  2C
                       1006 RAW
CB86: A9 34             1007          LDA  #$34
CB88: 2C                1008          HEX  2C
                       1009 VIDEO
CB89: A9 35             1010          LDA  #$35
CB8B: 2C                1011          HEX  2C
                       1012 SETATR
CB8C: A9 36             1013          LDA  #$36
CB8E: 9D B8 06          1014          STA  POFF,X
CB91: 60                1015          RTS
                       1016 *
                       1017 *********************
                       1018 *                   *
                       1019 *  HIGHLIGHT        *
                       1020 *  LOWLIGHT         *
                       1021 *  FLAG SET         *
                       1022 *  FLAG CLEAR       *
                       1023 *                   *
                       1024 *********************
                       1025 *
                       1026 HILITE
CB92: A9 3F            1027          LDA  #$3F
CB94: 85 32            1028          STA  INVFLG
CB96: A9 80            1029          LDA  #BINV       ; SET INVERSE FLAG BIT
                       1030 FLGSET
CB98: 1D 38 07         1031          ORA  FLAGS,X     ; SET FLAG BIT
CB9B: D0 09            1032          BNE  FLGSAV
                       1033 *
                       1034 LOLITE
CB9D: A9 FF            1035          LDA  #$FF
CB9F: 85 32            1036          STA  INVFLG
CBA1: A9 7F            1037          LDA  #BINV!$FF   ; CLEAR INVERSE FLAG BIT
                       1038 FLGCLR
CBA3: 3D 38 07         1039          AND  FLAGS,X     ; CLEAR FLAG BIT
                       1040 FLGSAV
CBA6: 9D 38 07         1041          STA  FLAGS,X     ; SAVE FLAG BIT
CBA9: 60               1042          RTS
```

```
                    1044 **********************
                    1045 *                      *
                    1046 *  CLEAR SCREEN        *
                    1047 *  HOME CURSOR         *
                    1048 *  CLEAR TO EOP        *
                    1049 *                      *
                    1050 **********************
                    1051 *
                    1052 CLSCRN
CBAA: 38            1053            SEC
CBAB: 90            1054            HEX  90
                    1055 HOME
CBAC: 18            1056            CLC
CBAD: 20 F9 CB      1057            JSR  CR1        ; SET CHORZ = 0
CBB0: 9D 38 05      1058            STA  CVERT,X    ; SET CVERT = 0
CBB3: 90 1C         1059            BCC  JVTAB      ; VTAB, EXIT IF C = 0
CBB5: 20 E6 CC      1060            JSR  VTAB
                    1061 *
                    1062 *
                    1063 CLREOP
CBB8: 20 AC CC      1064            JSR  CLREOL     ; CLEAR TO END OF CURRENT LINE
CBBB: BD 38 05      1065            LDA  CVERT,X
                    1066 CLEOP1
CBBF: 69 00         1067            ADC  #$00       ; NEXT LINE
CBC1: CD F8 04      1068            CMP  HEIGHT     ; DONE?
CBC3: B0 0C         1069            BGE  JVTAB      ; YES, EXIT TO VTAB
CBC5: 48            1070            PHA             ; SAVE LINE NUM.
CBC6: 20 E9 CC      1071            JSR  VTABZ      ; VTAB
CBC9: A0 00         1072            LDY  #$00       ; START AT BEGINNING OF LINE
CBCB: 20 BE CC      1073            JSR  CLEOLZ     ; CLEAR TO END OF LINE
CBCE: 68            1074            PLA
CBCF: B0 ED         1075            BCS  CLEOP1     ; CARRY IS ALWAYS SET
                    1076 JVTAB
CBD1: 4C E6 CC      1077            JMP  VTAB
                    1078 *
                    1079 **********************
                    1080 *                      *
                    1081 *  BELL                *
                    1082 *                      *
                    1083 **********************
                    1084 *
                    1085 BELL
CBD4: A0 60         1086            LDY  #$60
                    1087 BELL1
CBD6: A2 80         1088            LDX  #$80
                    1089 BELL2
CBD8: CA            1090            DEX
CBD9: D0 FD         1091            BNE  BELL2
CBDB: AD 30 C0      1092            LDA  SPKR
CBDE: 88            1093            DEY
CBDF: D0 F5         1094            BNE  BELL1
                    1095 GETX
CBE1: AE F8 07      1096            LDX  MSLOT
CBE4: 60            1097            RTS
```

```
                    1099 **********************
                    1100 *                     *
                    1101 *  ADVANCE            *
                    1102 *  STORE ADVANCE      *
                    1103 *  CARRIAGE RETURN    *
                    1104 *  LINEFEED           *
                    1105 *  SCROLL             *
                    1106 *  CLEAR TO EOL       *
                    1107 *                     *
                    1108 **********************
                    1109 *
                    1110 ADVANCE
CBE5: 20 EB CB      1111          JSR  PREADV     ; DO PREADVANCE FOR PASCAL
CBE8: 4C 16 CC      1112          JMP  ADVNCE     ; FINISH ADVANCING
                    1113 *
                    1114 PREADV
CBEB: BC B8 04      1115          LDY  CHORZ,X    ; IF BEYOND SCREEN WIDTH
CBEE: CC F8 05      1116          CPY  PWDTH      ; THEN DO A CR
CBF1: B0 33         1117          BCS  CRLF
                    1118 RTS8
CBF3: 60            1119          RTS
                    1120 *
                    1121 CR
CBF4: BD B8 05      1122          LDA  BYTE,X     ; IF FROM BASIC
CBF7: 30 2D         1123          BMI  CRLF       ; DO LINEFEED AFTER CR
                    1124 CR1
CBF9: A9 00         1125          LDA  #$00       ; SET CHORZ = 0
CBFB: 9D B8 04      1126          STA  CHORZ,X
CBFE: 60            1127          RTS
                    1128 *
                    1129 STOADV
CBFF: 85 35         1130          STA  XSAVE      ; SAVE CHARACTER
CC01: 20 EB CB      1131          JSR  PREADV     ; DO PREADVANCE FOR PASCAL
CC04: BC B8 04      1132          LDY  CHORZ,X
CC07: 20 78 C9      1133          JSR  PAGSEL     ; COMPUTE SCREEN ADDRESS
CC0A: A5 32         1134          LDA  INVFLG     ; COMBINE INVFLG WITH CHARACTER
CC0C: 29 80         1135          AND  #$80
CC0E: 45 35         1136          EOR  XSAVE
CC10: 20 6F CB      1137          JSR  CHRPUT     ; PUT CHARACTER ON SCREEN
CC13: AE F8 07      1138          LDX  MSLOT      ; RESTORE X
                    1139 ADVNCE
CC16: FE B8 04      1140          INC  CHORZ,X    ; ADVANCE CHORZ
CC19: BD B8 05      1141          LDA  BYTE,X     ; IF PASCAL, EXIT
CC1C: 10 D5         1142          BPL  RTS8
CC1E: BD B8 04      1143          LDA  CHORZ,X    ; IF CHORZ > SCREEN WIDTH THEN
CC21: CD F8 05      1144          CMP  PWDTH      ; DO A CRLF
CC24: 90 CD         1145          BCC  RTS8
```

```
                     1147 CRLF
CC26: 20 F9 CB       1148          JSR  CR1        ; DO A CR
                     1149 LF
CC29: FE 38 05       1150          INC  CVERT,X    ; NEXT LINE
CC2C: BD 38 05       1151          LDA  CVERT,X    ; BOTTOM?
CC2F: CD F8 04       1152          CMP  HEIGHT
CC32: 90 9D          1153          BCC  JVTAB      ; NO, EXIT
CC34: DE 38 05       1154          DEC  CVERT,X    ; FIX CVERT AND SCROLL
                     1155 *
CC37: BD 38 07       1156          LDA  FLAGS,X    ; INCREMENT START ADDRESS
CC3A: 29 07          1157          AND  #BFORMT    ; BY VALUE BASED ON SCREEN FORMAT
CC3C: A8             1158          TAY
CC3D: B9 3A CF       1159          LDA  SCLTBL,Y
CC40: 18             1160          CLC
CC41: 7D 38 06       1161          ADC  START,X
CC44: AC 78 04       1162          LDY  MODE
CC47: C0 C0          1163          CPY  #MBSVS     ; 512 ADDRESSING MODE HAS LESS
CC49: D0 02          1164          BNE  N512       ; SCREEN AREA
CC4B: 29 7F          1165          AND  #$7F
                     1166 N512
CC4D: 9D 38 06       1167          STA  START,X    ; SAVE NEW START VALUE
CC50: 18             1168          CLC             ; CALCULATE THE START ADDRESS
CC51: 20 04 CD       1169          JSR  BASCALC
CC54: AC F8 06       1170          LDY  NO
CC57: A9 0D          1171          LDA  #$0D       ; SELECT START ADDRESS LOW REG.
CC59: 99 80 C0       1172          STA  REGSEL,Y
CC5C: BD B8 03       1173          LDA  BASEL,X
CC5F: 29 F0          1174          AND  #$F0       ; SAVE START ADDRESS LOW
CC61: 99 81 C0       1175          STA  DATA,Y
CC64: A9 0C          1176          LDA  #$0C       ; SELECT START ADDRESS HIGH REG.
CC66: 99 80 C0       1177          STA  REGSEL,Y
CC69: BD 38 04       1178          LDA  BASEH,X
CC6C: 29 0F          1179          AND  #$0F
CC6E: 99 81 C0       1180          STA  DATA,Y     ; SAVE START ADDRESS HIGH
CC71: AC F8 04       1181          LDY  HEIGHT     ; PUT HEIGHT-1 INTO A
CC74: 88             1182          DEY
CC75: 98             1183          TYA
CC76: 20 E9 CC       1184          JSR  VTABZ      ; VTAB
CC79: A0 00          1185          LDY  #$00
CC7B: 20 BE CC       1186          JSR  CLEOLZ     ; CLEAR BOTTOM LINE
CC7E: B0 66          1187          BCS  VTAB
```

```
                    1189 FIXBUF
CC80: A5 33         1190        LDA  PROMPT     ; FIX INPUT BUFFER IF PROMPT
CC82: C9 DD         1191        CMP  #"]"       ; IS |, >, OR *
CC84: F0 08         1192        BEQ  FIXIT
CC86: C9 BE         1193        CMP  #">"
CC88: F0 04         1194        BEQ  FIXIT
CC8A: C9 AA         1195        CMP  #"*"
CC8C: D0 1E         1196        BNE  CLREOL
                    1197 FIXIT
CC8E: A2 00         1198        LDX  #$00       ; START AT BEGINNING
CC90: A0 00         1199        LDY  #$00       ; TURN QUOTE FLAG OFF
                    1200 FLOOP1
CC92: BD 00 02      1201        LDA  IN,X       ; GET CHARACTER
CC95: 48            1202        PHA
CC96: C9 A2         1203        CMP  #$A2       ; IS IT A QUOTE?
CC98: D0 01         1204        BNE  NTQTE      ; NO, SKIP
CC9A: C8            1205        INY             ; TOGGLE QUOTE FLAG
                    1206 NTQTE
CC9B: 98            1207        TYA             ; PUT QUOTE FLAG IN CARRY
CC9C: 4A            1208        LSR
CC9D: 68            1209        PLA             ; GET CHARACTER
CC9E: B0 06         1210        BCS  NXTIN      ; CONVERT TO UPPERCASE IF
CCA0: 20 79 CB      1211        JSR  CAPSLK     ; QUOTE FLAG IS OFF
CCA3: 9D 00 02      1212        STA  IN,X       ; SAVE CHARACTER
                    1213 NXTIN
CCA6: E8            1214        INX             ; NEXT BUFFER POSITION
CCA7: D0 E9         1215        BNE  FLOOP1     ; CONTINUE UNTIL DONE
CCA9: AE F8 07      1216        LDX  MSLOT      ; RECOVER X
                    1217 *
                    1218 *
                    1219 CLREOL
CCAC: BC B8 04      1220        LDY  CHORZ,X    ; START AT CHORZ
CCAF: 4C BE CC      1221        JMP  CLEOLZ
                    1222 *
                    1223 CLEOL2
CCB2: C8            1224        INY             ; ADVANCE POSITION
CCB3: 4A            1225        LSR             ; RECOVER CARRY
CCB4: 20 6F CB      1226        JSR  CHRPUT     ; PUT SPACE ON SCREEN
CCB7: 2A            1227        ROL             ; SAVE CARRY
CCB8: E8            1228        INX             ; NEXT PAGE INDEX
CCB9: D0 09         1229        BNE  CLSKIP     ; IF 0, SELECT NEW PAGE
CCBB: AE F8 07      1230        LDX  MSLOT
                    1231 CLEOLZ
CCBE: 20 78 C9      1232        JSR  PAGSEL     ; CALCULATE SCREEN ADDRESS
CCC1: A9 20         1233        LDA  #$20       ; USE A SPACE
CCC3: 2A            1234        ROL             ; SAVE CARRY
                    1235 CLSKIP
CCC4: CC 78 05      1236        CPY  SWDTH      ; DONE?
CCC7: 90 E9         1237        BLT  CLEOL2     ; NO, CONTINUE LOOP
CCC9: 4C E1 CB      1238        JMP  GETX       ; RECOVER X
```

```
                    1240 **********************
                    1241 *                     *
                    1242 *  BACKSPACE          *
                    1243 *  REVERSE LINEFEED   *
                    1244 *  VERTICAL TAB       *
                    1245 *                     *
                    1246 **********************
                    1247 *
                    1248 BS
CCCC: BD B8 04      1249            LDA  CHORZ,X     ; IF CHORZ NOT = 0 THEN DECREMENT
CCCF: F0 04         1250            BEQ  ENDUP
CCD1: DE B8 04      1251            DEC  CHORZ,X
CCD4: 60            1252            RTS
                    1253 *
                    1254 ENDUP
CCD5: AD F8 05      1255            LDA  PWDTH       ; GO TO END OF LINE
CCD8: 9D B8 04      1256            STA  CHORZ,X
CCDB: DE B8 04      1257            DEC  CHORZ,X
                    1258 *
                    1259 * MOVE CURSOR UP
                    1260 *
                    1261 UP
CCDE: BD 38 05      1262            LDA  CVERT,X     ; REVERSE LINEFEED IF NOT AT TOP
CCE1: F0 50         1263            BEQ  RTS1
CCE3: DE 38 05      1264            DEC  CVERT,X
                    1265 *
                    1266 * CALCULATE NEW BASE ADDRESSES FOR CURRENT LINE
                    1267 *
                    1268 VTAB
CCE6: BD 38 05      1269            LDA  CVERT,X     ; GET VERTICAL POSITION
                    1270 VTABZ
CCE9: 8D 78 07      1271            STA  TEMPX       ; SAVE IT
CCEC: BD 38 07      1272            LDA  FLAGS,X     ; GET FORMAT NUMBER
CCEF: 29 07         1273            AND  #$FORMT
CCF1: A8            1274            TAY
CCF2: B9 42 CF      1275            LDA  MULTBL,Y
CCF5: AB            1276            TAY
CCF6: AD 78 07      1277            LDA  TEMPX       ; MULTIPLY VERTICAL POSITION
CCF9: 0A            1278            ASL              ; BY FOUR
CCFA: 0A            1279            ASL
                    1280 VTLOOP
CCFB: 6D 78 07      1281            ADC  TEMPX       ; MULTIPLY BY 5, 6, 8, OR 10
CCFE: 88            1282            DEY
CCFF: D0 FA         1283            BNE  VTLOOP
CD01: 7D 38 06      1284            ADC  START,X     ; ADD IN START
                    1285 BASCALC
CD04: AC 78 04      1286            LDY  MODE        ; MULTIPLY BY 16
CD07: 48            1287            PHA
CD08: 6A            1288            ROR
CD09: C0 C0         1289            CPY  #MBSVS      ; CAN'T USE HIGH BIT IN
CD0B: D0 02         1290            BNE  NT512       ; 512 BYTE BLOCK ADDRESSING MODE
CD0D: 29 7F         1291            AND  #$7F
                    1292 NT512
CD0F: 4A            1293            LSR
CD10: 4A            1294            LSR
CD11: 4A            1295            LSR
CD12: 09 20         1296            ORA  #$20        ; FOOL THE APPLE //e
```

```
CD14: 9D 38 04  1297          STA  BASEH,X
CD17: 68        1298          PLA
CD18: 0A        1299          ASL
CD19: 0A        1300          ASL
CD1A: 0A        1301          ASL
CD1B: 0A        1302          ASL
CD1C: 9D B8 03  1303          STA  BASEL,X
CD1F: AD F8 05  1304          LDA  PWDTH
CD22: C9 84     1305          CMP  #132       ; IF PRINTED SCREEN WIDTH = 132
CD24: D0 0D     1306          BNE  RTS1
CD26: A9 0E     1307          LDA  #$0E       ; THEN ADD 14 TO THE BASE ADDRESS
CD28: 7D B8 03  1308          ADC  BASEL,X
CD2B: 9D B8 03  1309          STA  BASEL,X
CD2E: 90 03     1310          BCC  RTS1
CD30: FE 38 04  1311          INC  BASEH,X
                1312 RTS1
CD33: 60        1313          RTS
```

```
                    1315 FORMAT
CD34: 29 0F         1316          AND  #$0F       ; IF FORMAT 0 THEN
CD36: D0 0F         1317          BNE  FMT1       ; EXIT TO 40 COLUMNS
                    1318 PRO
CD38: AD 58 C0      1319          LDA  $C058      ; OLD SOFT VIDEO SWITCH
CD3B: AC F8 06      1320          LDY  NO
CD3E: A9 3F         1321          LDA  #MBSVS!$FF ; SET MODE TO DISPLAY
CD40: 2D 78 04      1322          AND  MODE       ; 40 COLUMN VIDEO
CD43: 8D 78 04      1323          STA  MODE
CD46: 60            1324          RTS
                    1325 FMT1
CD47: A8            1326          TAY             ; PUT FORMAT - 1 IN FLAGS
CD48: A9 F8         1327          LDA  #BFORMT!$FF
CD4A: 20 A3 CB      1328          JSR  FLGCLR
CD4D: 88            1329          DEY
CD4E: 98            1330          TYA
CD4F: 29 07         1331          AND  #BFORMT
CD51: 20 98 CB      1332          JSR  FLGSET
CD54: 4C 6F C8      1333          JMP  NEWFMT     ; INITIALIZE NEW FORMAT
                    1334 *
                    1335 GOTOX
CD57: 38            1336          SEC             ; SAVE X POSITION - $20
CD58: E9 20         1337          SBC  #$20
CD5A: 29 7F         1338          AND  #$7F
CD5C: 8D 78 07      1339          STA  TEMPX
CD5F: 60            1340          RTS
                    1341 *
                    1342 GOTOY
CD60: 38            1343          SEC             ; SUBTRACT $20 FROM Y POSITION
CD61: E9 20         1344          SBC  #$20
CD63: 29 7F         1345          AND  #$7F       ; ESTABLISH CVERT
CD65: CD F8 04      1346          CMP  HEIGHT
CD68: B0 03         1347          BGE  BADY
CD6A: 9D 38 05      1348          STA  CVERT,X
                    1349 BADY
CD6D: AD 78 07      1350          LDA  TEMPX      ; ESTABLISH CHORZ
CD70: CD F8 05      1351          CMP  PWDTH
CD73: B0 03         1352          BGE  BADX
CD75: 9D B8 04      1353          STA  CHORZ,X
                    1354 BADX
CD78: 4C E6 CC      1355          JMP  VTAB       ; GO TO NEW POSITION
                    1356 *
                    1357 LEAD
CD7B: 48            1358          PHA             ; IF 0, 1, 2, OR 3 THEN DO
CD7C: 49 B0         1359          EOR  #$B0       ; VIDEOTERM LEAD FUNCTIONS
CD7E: C9 04         1360          CMP  #$04
CD80: 68            1361          PLA
CD81: 90 03         1362          BLT  DOVTZ
                    1363 JSTADV
CD83: 4C FF CB      1364          JMP  STOADV     ; PUT CHARACTER ON SCREEN
                    1365 DOVTZ
CD86: 49 98         1366          EOR  #$98
CD88: 4C F6 CD      1367          JMP  CONTROL
                    1368 *
                    1369 RAWVID
CD8B: C9 A0         1370          CMP  #$A0       ; ALLOW CHARACTERS >= SPACE
CD8D: B0 F4         1371          BGE  JSTADV
```

```
CD8F: C9 8D      1372          CMP  #$8D       ; ALLOW CR
CD91: F0 5B      1373          BEQ  VIDOUT
CD93: C9 8B      1374          CMP  #$8B       ; ALLOW CHARACTERS <= LF
CD95: 90 57      1375          BLT  VIDOUT
CD97: 60         1376          RTS
                 1377 *
                 1378 LOATR
CD98: 48         1379          PHA
CD99: BD 38 07   1380          LDA  FLAGS,X    ; COMPUTE CHARACTER SET
CD9C: 29 07      1381          AND  #BFORMT
CD9E: A8         1382          TAY
CD9F: 68         1383          PLA             ; COMBINE ATTRIBUTE WITH
CDA0: 59 44 C9   1384          EOR  ATRTBL,Y   ; CHARACTER SET
CDA3: 29 07      1385          AND  #$07
CDA5: 8D 78 07   1386          STA  TEMPX      ; SAVE LOW ATTRIBUTE
CDA8: 60         1387          RTS
                 1388 *
                 1389 HIATR
CDA9: 29 03      1390          AND  #$03       ; PUT ATTRIBUTE IN HIGH NYBBLE
CDAB: 0A         1391          ASL
CDAC: 0A         1392          ASL
CDAD: 0A         1393          ASL
CDAE: 0A         1394          ASL
CDAF: 0D 78 07   1395          ORA  TEMPX      ; COMBINE WITH LOW ATTRIBUTE
CDB2: AC F8 06   1396          LDY  NO
CDB5: 99 83 C0   1397          STA  ATTREG,Y   ; SET NEW ATTRIBUTES
CDB8: 60         1398          RTS
```

```
                    1400 * PERFORM ESCAPE FUNCTIONS
                    1401 *
                    1402 ESCNEW
CDB9: 20 E2 CD      1403          JSR  ESCRD       ; READ ESCAPE KEY
CDBC: C9 09         1404          CMP  #$09        ; IS IT 0 - 8?
CDBE: B0 03         1405          BCE  ESC2        ; NO, TRY OTHERS
CDC0: 4C 34 CD      1406          JMP  FORMAT      ; YES, SELECT NEW FORMAT
                    1407 ESC2
CDC3: 49 70         1408          EOR  #$70        ; IS IT A - F?
CDC5: C9 08         1409          CMP  #$08
CDC7: 90 2D         1410          BLT  CONTROL     ; YES, PERFORM FUNCTION
                    1411 ESCNOW
CDC9: C9 0E         1412          CMP  #$0E        ; IS IT >= N?
CDCB: B0 20         1413          BGE  RTS3        ; YES, EXIT
CDCD: C9 09         1414          CMP  #$09        ; IS IT < I?
CDCF: 90 1C         1415          BLT  RTS3        ; YES, EXIT
CDD1: C9 0C         1416          CMP  #$0C        ; IS IT L?
CDD3: F0 18         1417          BEQ  RTS3        ; YES, EXIT
CDD5: A8            1418          TAY
CDD6: B9 4C CE      1419          LDA  XLTBL-$09,Y ; CONVERT TO A, B, C, OR D
CDD9: 20 F6 CD      1420          JSR  CONTROL     ; PERFORM FUNCTION
CDDC: 20 E2 CD      1421          JSR  ESCRD       ; READ NEW ESCAPE KEY
CDDF: 4C C3 CD      1422          JMP  ESC2        ; PROCESS KEY
                    1423 *
                    1424 ESCRD
CDE2: 20 59 CB      1425          JSR  RDKEY
CDE5: 20 79 CB      1426          JSR  CAPSLK
CDE8: 9D B8 05      1427          STA  BYTE,X
CDEB: 49 B0         1428          EOR  #$B0
                    1429 RTS3
CDED: 60            1430          RTS
                    1431 *
                    1432 VIDOUT
CDEE: C9 A0         1433          CMP  #$A0        ; OUTPUT CHARACTERS >= SPACE
CDF0: B0 91         1434          BGE  JSTADV
CDF2: C9 87         1435          CMP  #$87        ; PERFORM FUNCTIONS ON
CDF4: 90 0F         1436          BLT  RTS4        ; CONTROL CHARACTERS > CTRL F
                    1437 CONTROL
CDF6: 0A            1438          ASL
CDF7: A8            1439          TAY
CDF8: B9 5B CE      1440          LDA  CTLTBL+1,Y  ; GET FUNCTION ADDRESSES
CDFB: 48            1441          PHA              ; AND PUSH THEM ON THE STACK
CDFC: B9 5A CE      1442          LDA  CTLTBL,Y
CDFF: 48            1443          PHA
CE00: BD B8 05      1444          LDA  BYTE,X
CE03: 09 80         1445          ORA  #$80
                    1446 RTS4
CE05: 60            1447          RTS              ; DISPATCH TO FUNCTION
```

```
                    1449 * GENERAL OUTPUT ROUTINE
                    1450 *
                    1451 OUTPT1
CE06: AD 59 C0      1452            LDA  $C059       ; TURN OLD SOFT VIDEO SWITCH ON
CE09: BC B8 06      1453            LDY  POFF,X      ; FETCH CURRENT STATE
CE0C: B9 82 CE      1454            LDA  STATE-$30,Y ; ESTABLISH NEW STATE
CE0F: 9D B8 06      1455            STA  POFF,X
CE12: 98            1456            TYA              ; PERFORM CURRENT STATE FUNCTION
CE13: 29 27         1457            AND  #$27
CE15: D0 DF         1458            BNE  CONTROL
                    1459 *
                    1460 *
                    1461 FIXCSR
CE17: A5 25         1462            LDA  CV          ; PERFORM VTAB
CE19: DD 38 05      1463            CMP  CVERT,X
CE1C: F0 06         1464            BEQ  CVOK
CE1E: 9D 38 05      1465            STA  CVERT,X
CE21: 20 E6 CC      1466            JSR  VTAB
                    1467 CVOK
CE24: A5 24         1468            LDA  CH          ; PERFORM COMMA TAB
CE26: DD B8 04      1469            CMP  CHORZ,X
CE29: B0 1C         1470            BCS  NCOMMA
CE2B: C9 11         1471            CMP  #$11
CE2D: B0 18         1472            BCS  NCOMMA
CE2F: 09 F0         1473            ORA  #$F0
CE31: 3D B8 04      1474            AND  CHORZ,X
CE34: 65 24         1475            ADC  CH
CE36: CD F8 05      1476            CMP  PWDTH
CE39: 90 0A         1477            BCC  NTEOL
CE3B: 20 26 CC      1478            JSR  CRLF
CE3E: BD 38 05      1479            LDA  CVERT,X
CE41: 85 25         1480            STA  CV
CE43: A9 00         1481            LDA  #$00
                    1482 NTEOL
CE45: 85 24         1483            STA  CH
                    1484 NCOMMA
CE47: DD B8 04      1485            CMP  CHORZ,X     ; PERFORM HTAB
CE4A: 90 03         1486            BCC  RTS2
CE4C: 9D B8 04      1487            STA  CHORZ,X
                    1488 RTS2
CE4F: 60            1489            RTS
```

```
                    1491 *
                    1492 FIXWDTH
CE50: A9 28         1493          LDA  #40
CE52: 85 21         1494          STA  WNDWDTH
CE54: 60            1495          RTS
                    1496 XLTBL
CE55: 04 02 01
CE58: FF 03         1497          HEX  040201FF03
                    1498 *
                    1499 ************************
                    1500 *  ESCAPE CODES        *
                    1501 ************************
                    1502 *
                    1503 CTLTBL
CE5A: A9 CB         1504 ESC@     DA   CLSCRN-1
CE5C: E4 CB         1505 ESCA     DA   ADVANCE-1
CE5E: CB CC         1506 ESCB     DA   BS-1
CE60: 28 CC         1507 ESCC     DA   LF-1
CE62: DD CC         1508 ESCD     DA   UP-1
CE64: AB CC         1509 ESCE     DA   CLREOL-1
CE66: B7 CB         1510 ESCF     DA   CLREOP-1
                    1511 *
                    1512 ************************
                    1513 *  CONTROL CODES      *
                    1514 ************************
                    1515 *
CE68: D3 CB         1516 CTLG     DA   BELL-1
CE6A: CB CC         1517 CTLH     DA   BS-1
CE6C: 57 CB         1518 CTLI     DA   RTSO-1
CE6E: 28 CC         1519 CTLJ     DA   LF-1
CE70: B7 CB         1520 CTLK     DA   CLREOP-1
CE72: A9 CB         1521 CTLL     DA   CLSCRN-1
CE74: F3 CB         1522 CTLM     DA   CR-1
CE76: 9C CB         1523 CTLN     DA   LOLITE-1
CE78: 91 CB         1524 CTLO     DA   HILITE-1
CE7A: 57 CB         1525 CTLP     DA   RTSO-1
CE7C: 57 CB         1526 CTLQ     DA   RTSO-1
CE7E: 85 CB         1527 CTLR     DA   RAW-1
CE80: 57 CB         1528 CTLS     DA   RTSO-1
CE82: 57 CB         1529 CTLT     DA   RTSO-1
CE84: 37 CD         1530 CTLU     DA   PRO-1
CE86: 88 CB         1531 CTLV     DA   VIDEO-1
CE88: 8B CB         1532 CTLW     DA   SETATR-1
CE8A: 57 CB         1533 CTLX     DA   RTSO-1
CE8C: AB CB         1534 CTLY     DA   HOME-1
CE8E: 82 CB         1535 CTLZ     DA   LEADIN-1
CE90: 57 CB         1536 CTL[     DA   RTSO-1
CE92: E4 CB         1537 CTL\     DA   ADVANCE-1
CE94: AB CC         1538 CTL]     DA   CLREOL-1
CE96: 7F CB         1539 CTL^     DA   COXY-1
CE98: DD CC         1540 CTL_     DA   UP-1
                    1541 *
                    1542 ************************
                    1543 *  DISPATCH TABLE     *
                    1544 ************************
                    1545 *
                    1546 DSPTBL
```

```
CE9A: ED CD      1547           DA   VIDOUT-1
CE9C: 56 CD      1548           DA   GOTOX-1
CE9E: 5F CD      1549           DA   GOTOY-1
CEA0: 7A CD      1550           DA   LEAD-1
CEA2: 8A CD      1551           DA   RAWVID-1
CEA4: 33 CD      1552           DA   FORMAT-1
CEA6: 97 CD      1553           DA   LOATR-1
CEA8: A8 CD      1554           DA   HIATR-1
                 1555 *
                 1556 LEADTBL
CEAA: 3B C8      1557 CTLZ0     DA   RESTART-1
CEAC: 37 CD      1558 CTLZ1     DA   PRO-1
CEAE: 9C CB      1559 CTLZ2     DA   LOLITE-1
CEB0: 91 CB      1560 CTLZ3     DA   HILITE-1
                 1561 *
                 1562 *
                 1563 STATE
CEB2: 30         1564           DFB  $30
CEB3: 32         1565           DFB  $32
CEB4: 30         1566           DFB  $30
CEB5: 30         1567           DFB  $30
CEB6: 34         1568           DFB  $34
CEB7: 30         1569           DFB  $30
CEB8: 37         1570           DFB  $37
CEB9: 30         1571           DFB  $30
                 1572 *
                 1573 TABLE
                 1574 * 80 X 24
                 1575           >>>  T80
                 1575           >>>  HT1
CEBA: 7A         1575           HEX  7A
                 1575           <<<
CEBB: 50 5D      1575           HEX  505D
                 1575           >>>  SW1
CEBD: 38         1575           HEX  38
                 1575           <<<
                 1575           <<<
                 1576           >>>  X24
                 1576           DO   LINEHZ!60
                 1576           HEX  2107181E
                 1576           ELSE
CEBE: 1C 00 18
CEC1: 1A         1576           HEX  1C00181A
                 1576           FIN
CEC2: 00 08 60
CEC5: 08         1576           HEX  00086008
                 1576           >>>  COMMON
CEC6: 00 00 FF
CEC9: 00         1576           HEX  0000FF00
                 1576           <<<
                 1576           <<<
                 1577 * 96 X 24
                 1578           >>>  T96
                 1578           >>>  HT1
CECA: 7A         1578           HEX  7A
                 1578           <<<
CECB: 60 68      1578           HEX  6068
```

Appendix F

```
                    1578            >>>  SW1
CECD: 38            1578            HEX  38
                    1578            <<<
                    1578            <<<
                    1579            >>>  X24
                    1579            DO   LINEHZ!60
                    1579            HEX  2107181E
                    1579            ELSE
CECE: 1C 00 18
CED1: 1A            1579            HEX  1C00181A
                    1579            FIN
CED2: 00 08 60
CED5: 08            1579            HEX  00086008
                    1579            >>>  COMMON
CED6: 00 00 FF
CED9: 00            1579            HEX  0000FF00
                    1579            <<<
                    1579            <<<
                    1580 * 160 X 24
                    1581            >>>  T160
                    1581            >>>  HT2
CEDA: CA            1581            HEX  CA
                    1581            <<<
CEDB: A0 AB         1581            HEX  A0AB
                    1581            >>>  SW2
CEDD: 4D            1581            HEX  4D
                    1581            <<<
                    1581            <<<
                    1582            >>>  X24
                    1582            DO   LINEHZ!60
                    1582            HEX  2107181E
                    1582            ELSE
CEDF: 1C 00 18
CEE1: 1A            1582            HEX  1C00181A
                    1582            FIN
CEE2: 00 08 60
CEE5: 08            1582            HEX  00086008
                    1582            >>>  COMMON
CEE6: 00 00 FF
CEE9: 00            1582            HEX  0000FF00
                    1582            <<<
                    1582            <<<
                    1583 * 80 X 24
                    1584            >>>  T80
                    1584            >>>  HT1
CEEA: 7A            1584            HEX  7A
                    1584            <<<
CEEB: 50 5D         1584            HEX  505D
                    1584            >>>  SW1
CEED: 38            1584            HEX  38
                    1584            <<<
                    1584            <<<
                    1585            >>>  X241
                    1585            DO   LINEHZ!60
                    1585            HEX  26011821
                    1585            ELSE
CEEE: 1F 05 18
```

```
CEF1: 1C          1585          HEX  1F05181C
                  1585          FIN
CEF2: 03 0E 60
CEF5: 0C          1585          HEX  030E600C
                  1585          >>>  COMMON
CEF6: 00 00 FF
CEF9: 00          1585          HEX  0000FF00
                  1585          <<<
                  1585          <<<
                  1586 * 80 X 32
                  1587          >>>  T80
                  1587          >>>  HT1
CEFA: 7A          1587          HEX  7A
                  1587          <<<
CEFB: 50 5D       1587          HEX  505D
                  1587          >>>  SW1
CEFD: 38          1587          HEX  38
                  1587          <<<
                  1587          <<<
                  1588          >>>  X32
                  1588          DO   LINEHZ!60
                  1588          HEX  2F012029
                  1588          ELSE
CEFE: 27 01 20
CF01: 24          1588          HEX  27012024
                  1588          FIN
CF02: 03 0B 60
CF05: 0C          1588          HEX  030B600C
                  1588          >>>  COMMON
CF06: 00 00 FF
CF09: 00          1588          HEX  0000FF00
                  1588          <<<
                  1588          <<<
                  1589 * 80 X 48
                  1590          >>>  T80
                  1590          >>>  HT1
CF0A: 7A          1590          HEX  7A
                  1590          <<<
CF0B: 50 5D       1590          HEX  505D
                  1590          >>>  SW1
CF0D: 38          1590          HEX  38
                  1590          <<<
                  1590          <<<
                  1591          >>>  X48
                  1591          DO   LINEHZ!60
                  1591          HEX  4403303B
                  1591          ELSE
CF0E: 39 00 30
CF11: 35          1591          HEX  39003035
                  1591          FIN
CF12: 03 07 60
CF15: 08          1591          HEX  03076008
                  1591          >>>  COMMON
CF16: 00 00 FF
CF19: 00          1591          HEX  0000FF00
                  1591          <<<
                  1591          <<<
```

```
                    1592 * 132 X 24
                    1593          >>>  T132
                    1593          >>>  HT2
CF1A: CA            1593          HEX  CA
                    1593          <<<
CF1B: A0 A6         1593          HEX  A0A6
                    1593          >>>  SW2
CF1D: 4D            1593          HEX  4D
                    1593          <<<
                    1593          <<<
                    1594          >>>  X24I
                    1594          DO   LINEHZ!60
                    1594          HEX  26011821
                    1594          ELSE
CF1E: 1F 05 18
CF21: 1C            1594          HEX  1F05181C
                    1594          FIN
CF22: 03 0E 60
CF25: 0C            1594          HEX  030E600C
                    1594          >>>  COMMON
CF26: 00 00 FF
CF29: 00            1594          HEX  0000FF00
                    1594          <<<
                    1594          <<<
                    1595 * 128 X 32
                    1596          >>>  T128
                    1596          >>>  HT2
CF2A: CA            1596          HEX  CA
                    1596          <<<
CF2B: 80 96         1596          HEX  8096
                    1596          >>>  SW2
CF2D: 4D            1596          HEX  4D
                    1596          <<<
                    1596          <<<
                    1597          >>>  X32
                    1597          DO   LINEHZ!60
                    1597          HEX  2F012029
                    1597          ELSE
CF2E: 27 01 20
CF31: 24            1597          HEX  27012024
                    1597          FIN
CF32: 03 0B 60
CF35: 0C            1597          HEX  030B600C
                    1597          >>>  COMMON
CF36: 00 00 FF
CF39: 00            1597          HEX  0000FF00
                    1597          <<<
                    1597          <<<
                    1598 *
                    1599 SCLTBL
CF3A: 05            1600          DFB  $05
CF3B: 06            1601          DFB  $06
CF3C: 0A            1602          DFB  $0A
CF3D: 05            1603          DFB  $05
CF3E: 05            1604          DFB  $05
CF3F: 05            1605          DFB  $05
CF40: 0A            1606          DFB  $0A
```

```
CF41: 08        1607            DFB  $08
                1608 MULTBL
CF42: 01        1609            DFB  $01
CF43: 02        1610            DFB  $02
CF44: 06        1611            DFB  $06
CF45: 01        1612            DFB  $01
CF46: 01        1613            DFB  $01
CF47: 01        1614            DFB  $01
CF48: 06        1615            DFB  $06
CF49: 04        1616            DFB  $04
```

```
--End assembly--

3914 bytes

Errors: 0

Symbol table - alphabetical order:

    A1H     =$3D        A1L     =$3C        A2E     =$C86B      A2H     =$3F
    A2L     =$3E        A4H     =$43        A4L     =$42        ADVANCE =$CBE5
    ADVNCE  =$CC16      APVTAB  =$FC22      ATCHR   =$04        ATDFLT  =$80
?   ATHIO   =$01    ?   ATHI1   =$10    ?   ATINVO  =$02    ?   ATINV1  =$20
    ATLRG   =$80        ATRTBL  =$C944      ATSML   =$84        ATTREG  =$C083
    BADX    =$CD78      BADY    =$CD6D      BASCALC =$CD04      BASEH   =$0438
    BASEL   =$03B8      BASINP  =$C9AD      BASL    =$28        BASOUT  =$CA15
    BELL    =$CBD4      BELL1   =$CBD6      BELL2   =$CBD8      BFORMT  =$07
    BGETLN  =$10    M   BINIT   =$C7EB      BINV    =$80        BKEYBD  =$20
    BLCCON  =$40        BOUT    =$CA27      BPOFF   =$F8        BS      =$CCCC
    BSTART  =$C81A      BSTART1 =$C912  ?   BSTATE  =$07        BYTE    =$05B8
    C0      =$C0        C000    =$C000  M   C01     =$C798  M   C03     =$C7A3
    CAPSLK  =$CB79      CH      =$24        CHORZ   =$04B8      CHRPUT  =$CB6F
    CLEOL2  =$CCB2      CLEOLZ  =$CCBE      CLEOP1  =$CBBE      CLOOP1  =$C887
    CLREOL  =$CCAC      CLREOP  =$CBB8  M   CLRX    =$C737      CLSCRN  =$CBAA
    CLSKIP  =$CCC4  MD  CN00    =$C000  MD  COMMON  =$C000      CONTROL =$CDF6
    CR      =$CBF4      CR1     =$CBF9      CRLF    =$CC26      CSLOOP1 =$C899
    CSRMOV  =$CB10      CSTART  =$C8CF      CSWH    =$37        CSWL    =$36
?   CTLG    =$CE68  ?   CTLH    =$CE6A  ?   CTLI    =$CE6C  ?   CTLJ    =$CE6E
?   CTLK    =$CE70  ?   CTLL    =$CE72  ?   CTLM    =$CE74  ?   CTLN    =$CE76
?   CTLO    =$CE78  ?   CTLP    =$CE7A  ?   CTLQ    =$CE7C  ?   CTLR    =$CE7E
?   CTLS    =$CE80  ?   CTLT    =$CE82      CTLTBL  =$CE5A  ?   CTLU    =$CE84
?   CTLV    =$CE86  ?   CTLW    =$CE88  ?   CTLX    =$CE8A  ?   CTLY    =$CE8C
?   CTLZ    =$CE8E  ?   CTLZ0   =$CEAA  ?   CTLZ1   =$CEAC  ?   CTLZ2   =$CEAE
?   CTLZ3   =$CEB0  ?   CTL[    =$CE90  ?   CTL\    =$CE92  ?   CTL]    =$CE94
?   CTL^    =$CE96  ?   CTL_    =$CE98      CV      =$25        CVERT   =$0538
    CVOK    =$CE24      DATA    =$C081      DISP0   =$CC00      DISP1   =$CD00
    DOVTZ   =$CD86  ?   DSPTBL  =$CE9A      ENDUP   =$CCD5  M   ENTER   =$C700
M   ENTR    =$C73E      ESC     =$C9D2      ESC2    =$CDC3  ?   ESC@    =$CE5A
?   ESCA    =$CE5C  ?   ESCB    =$CE5E  ?   ESCC    =$CE60  ?   ESCD    =$CE62
?   ESCE    =$CE64  ?   ESCF    =$CE66      ESCNEW  =$CDB9  ?   ESCNOW  =$CDC9
    ESCRD   =$CDE2      EXIT    =$C809  ?   FINIT   =$C802      FIXBUF  =$CC80
    FIXCSR  =$CE17      FIXIT   =$CC8E      FIXWDTH =$CE50      FLACS   =$0738
    FLGCLR  =$CBA3      FLGSAV  =$CBA6      FLGSET  =$CB98      FLOOP1  =$CC92
    FMT1    =$CD47      FORMAT  =$CD34      GETLN   =$C9E0      GETX    =$CBE1
    GOTOX   =$CD57      GOTOY   =$CD60      GOXY    =$CB80      HEIGHT  =$04F8
    HGTBL   =$C92C      HIATR   =$CDA9      HILITE  =$CB92      HOME    =$CBAC
MD  HT1     =$C000  MD  HT2     =$C000      IENTER1 =$C81C      IEXIT   =$C817
    IN      =$0200      INDONE  =$CAF2  M   INENTR  =$C73A  M?  INFAKE  =$C705
M   INIT    =$C717  M   INPUT   =$C75D      INVFLG  =$32    M   IO      =$C758
    IORTS   =$FFCB      JSTADV  =$CD83      JVTAB   =$CBD1      KBD     =$C000
    KBDSTRB =$C010      KBDWAIT =$CA52      KEYIN   =$CB45      KEYIN2  =$CB4B
    KEYSTAT =$CA9C      KSWH    =$39        KSWL    =$38        LEAD    =$CD7B
    LEADIN  =$CB83  ?   LEADTBL =$CEAA      LF      =$CC29      LINEHZ  =$3C
    LOATR   =$CD98      LOLITE  =$CB9D      LOOP    =$C8BA      LSTFIX  =$CA5E
    MB132   =$E0        MB256   =$D0        MBANK   =$80        MBSVS   =$C0
```

```
    MCPBITS =$F0        MCREG   =$C082      MODE    =$0478      MODTBL  =$C924
M   MOVE    =$C763  M   MOVEC2M =$C778  M   MOVELOOP=$C780  M   MOVERET =$C7AC
M   MOVESTRT=$C77E      MPADDR  =$10        MPBANK  =$80        MPCLOCK =$20
    MPVIDEO =$40        MSLOT   =$07F8      MULTBL  =$CF42      NO      =$06F8
    N512    =$CC4D      NCOMMA  =$CE47      NEWFMT  =$C86F      NEWFMT2 =$C8DA
    NFSKIP  =$C90E      NOCSR   =$CA3A      NORMOUT =$CA8B      NOSHIFT =$CAF0
    NOT@    =$CAE6      NOTCR   =$C9F5      NOTK    =$CAB9      NOTLWR  =$CAA9
    NOTPICK =$C9FC      NT512   =$CD0F      NTEOL   =$CE45      NTGETLN =$C9FF
    NTQTE   =$CC9B      NTSHFT  =$CACA  M   NXTA1   =$C78A      NXTIN   =$CCA6
    OLDCHAR =$0678  M   OUTENTR =$C707      OUTPT1  =$CE06  M   OUTPUT  =$C7FA
    PAGSEL  =$C978      PINIT   =$C800      POFF    =$06B8      PRO     =$CD38
    PREAD   =$CB42      PREADV  =$CBEB      PROMPT  =$33        PSTART  =$C818
    PWDTBL  =$C93C      PWDTH   =$05F8      PWRITE  =$CB04      PWRITE1 =$CB07
    RAW     =$CB86      RAWVID  =$CD8B      RDKEY   =$CB59      RDSCRN  =$C94C
    RDSKIP  =$C850  M   READ    =$C71D      REGSEL  =$C080      RESTART =$C83C
    RNDH    =$4F        RNDL    =$4E        RSKIP1  =$C962      RTS0    =$CB58
    RTS1    =$CD33      RTS2    =$CE4F      RTS3    =$CDED      RTS4    =$CE05
    RTS5    =$CB7F      RTS6    =$CB41      RTS8    =$CBF3  ?   RTS9    =$CAFA
    SAVOLD  =$CA04      SCLTBL  =$CF3A      SETATR  =$CB8C      SETCH   =$CA6D
    SETKBD  =$FE89      SETUP   =$C827      SETUP1  =$C831      SETVID  =$FE93
    SHUTUP  =$CB1E      SPKR    =$C030      SSKIP   =$CB27  ?   STACK   =$0100
    START   =$0638      STATE   =$CEB2  M   STATUS  =$C72A      STOADV  =$CBFF
MD  SW1     =$C000  MD  SW2     =$C000      SWDTBL  =$C934      SWDTH   =$0578
MD  T128    =$C000  MD  T132    =$C000  MD  T160    =$C000  MD  T80     =$C000
MD  T96     =$C000      TABLE   =$CEBA      TEMPX   =$0778      TINIT   =$C8D8
    UP      =$CCDE      VIDEO   =$CB89      VIDOUT  =$CDEE  ?   VIDWAIT =$FB78
    VSKIP   =$CA7F      VTAB    =$CCE6      VTABZ   =$CCE9      VTLOOP  =$CCFB
    WNDWDTH =$21    M   WRITE   =$C724      WSKIP   =$CB75  MD  X24     =$C000
MD  X24I    =$C000  MD  X32     =$C000  MD  X48     =$C000  M   XFER    =$C7B0
M   XFERAZP =$C7CD  M   XFERC2M =$C7C5  M   XFERSZP =$C7DC      XLTBL   =$CE55
    XSAVE   =$35        ZERO    =$CAEE
```

Symbol table - numerical order:

```
?   ATH10   =$01    ?   ATINV0  =$02        ATCHR   =$04    ?   BSTATE  =$07
    BFORMT  =$07        BGETLN  =$10        MPADDR  =$10    ?   ATH11   =$10
    BKEYBD  =$20        MPCLOCK =$20    ?   ATINV1  =$20        WNDWDTH =$21
    CH      =$24        CV      =$25        BASL    =$28        INVFLG  =$32
    PROMPT  =$33        XSAVE   =$35        CSWL    =$36        CSWH    =$37
    KSWL    =$38        KSWH    =$39        LINEHZ  =$3C        A1L     =$3C
    A1H     =$3D        A2L     =$3E        A2H     =$3F        BLCCON  =$40
    MPVIDEO =$40        A4L     =$42        A4H     =$43        RNDL    =$4E
    RNDH    =$4F        BINV    =$80        MPBANK  =$80        MBANK   =$80
    ATDFLT  =$80        ATLRG   =$80        ATSML   =$84        CO      =$C0
    MBSVS   =$C0        MB256   =$D0        MB132   =$E0        MCPBITS =$F0
    BPOFF   =$F8    ?   STACK   =$0100      1N      =$0200      BASEL   =$03B8
    BASEH   =$0438      MODE    =$0478      CHORZ   =$04B8      HEIGHT  =$04F8
    CVERT   =$0538      SWDTH   =$0578      BYTE    =$05B8      PWDTH   =$05F8
    START   =$0638      OLDCHAR =$0678      POFF    =$06B8      NO      =$06F8
    FLAGS   =$0738      TEMPX   =$0778      MSLOT   =$07F8      C000    =$C000
    KBD     =$C000  MD  HT1     =$C000  MD  HT2     =$C000  MD  SW1     =$C000
MD  SW2     =$C000  MD  T80     =$C000  MD  T96     =$C000  MD  T128    =$C000
MD  T132    =$C000  MD  T160    =$C000  MD  X24     =$C000  MD  X24I    =$C000
MD  X32     =$C000  MD  X48     =$C000  MD  COMMON  =$C000  MD  CN00    =$C000
    KBDSTRB =$C010      SPKR    =$C030      REGSEL  =$C080      DATA    =$C081
    MCREG   =$C082      ATTREG  =$C083  M   ENTER   =$C700  M?  INFAKE  =$C705
```

```
M  OUTENTR =$C707
M  STATUS  =$C72A
M  IO      =$C758
M  MOVESTRT=$C77E
M  CO3     =$C7A3
M  XFERAZP =$C7CD
   PINIT   =$C800
   PSTART  =$C818
   SETUP1  =$C831
   NEWFMT  =$C86F
   CSTART  =$C8CF
   BSTART1 =$C912
   PWDTBL  =$C93C
   PAGSEL  =$C978
   NOTCR   =$C9F5
   BASOUT  =$CA15
   LSTFIX  =$CA5E
   KEYSTAT =$CA9C
   NOT@    =$CAE6
?  RTS9    =$CAFA
   SHUTUP  =$CB1E
   KEYIN   =$CB45
   CHRPUT  =$CB6F
   GOXY    =$CB80
   SETATR  =$CB8C
   FLGCLR  =$CBA3
   CLREOP  =$CBB8
   BELL1   =$CBD6
   PREADV  =$CBEB
   STOADV  =$CBFF
   LF      =$CC29
   FLOOP1  =$CC92
   CLEOL2  =$CCB2
   ENDUP   =$CCD5
   VTLOOP  =$CCFB
   RTS1    =$CD33
   GOTOX   =$CD57
   LEAD    =$CD7B
   LOATR   =$CD98
?  ESCNOW  =$CDC9
   CONTROL =$CDF6
   CVOK    =$CE24
   FIXWDTH =$CE50
?  ESCA    =$CE5C
?  ESCE    =$CE64
?  CTLI    =$CE6C
?  CTLM    =$CE74
?  CTLQ    =$CE7C
?  CTLU    =$CE84
?  CTLY    =$CE8C
?  CTL]    =$CE94
?  LEADTBL =$CEAA
?  CTLZ3   =$CEB0
   MULTBL  =$CF42
   SETVID  =$FE93

M  INIT    =$C717
M  CLRX    =$C737
M  INPUT   =$C75D
M  MOVELOOP=$C780
M  MOVERET =$C7AC
M  XFERSZP =$C7DC
?  FINIT   =$C802
   BSTART  =$C81A
   RESTART =$C83C
   CLOOP1  =$C887
   TINIT   =$C8D8
   MODTBL  =$C924
   ATRTBL  =$C944
   BASINP  =$C9AD
   NOTPICK =$C9FC
   BOUT    =$CA27
   SETCH   =$CA6D
   NOTLWR  =$CAA9
   ZERO    =$CAEE
   PWRITE  =$CB04
   SSKIP   =$CB27
   KEYIN2  =$CB4B
   WSKIP   =$CB75
   LEADIN  =$CB83
   HILITE  =$CB92
   FLGSAV  =$CBA6
   CLEOP1  =$CBBE
   BELL2   =$CBD8
   RTS8    =$CBF3
   DISP0   =$CC00
   N512    =$CC4D
   NTQTE   =$CC9B
   CLEOLZ  =$CCBE
   UP      =$CCDE
   DISP1   =$CD00
   FORMAT  =$CD34
   GOTOY   =$CD60
   JSTADV  =$CD83
   HIATR   =$CDA9
   ESCRD   =$CDE2
   RTS4    =$CE05
   NTEOL   =$CE45
   XLTBL   =$CE55
?  ESCB    =$CE5E
?  ESCF    =$CE66
?  CTLJ    =$CE6E
?  CTLN    =$CE76
?  CTLR    =$CE7E
?  CTLV    =$CE86
?  CTLZ    =$CE8E
?  CTL^    =$CE96
?  CTLZ0   =$CEAA
   STATE   =$CEB2
?  VIDWAIT =$FB78
   IORTS   =$FFCB

M  READ    =$C71D
M  INENTR  =$C73A
M  MOVE    =$C763
M  NXTA1   =$C78A
M  XFER    =$C7B0
M  BINIT   =$C7EB
   EXIT    =$C809
   IENTER1 =$C81C
   RDSKIP  =$C850
   CSLOOP1 =$C899
   NEWFMT2 =$C8DA
   HGTBL   =$C92C
   RDSCRN  =$C94C
   ESC     =$C9D2
   NTGETLN =$C9FF
   NOCSR   =$CA3A
   VSKIP   =$CA7F
   NOTK    =$CAB9
   NOSHIFT =$CAF0
   PWRITE1 =$CB07
   RTS6    =$CB41
   RTSO    =$CB58
   CAPSLK  =$CB79
   RAW     =$CB86
   FLGSET  =$CB98
   CLSCRN  =$CBAA
   JVTAB   =$CBD1
   GETX    =$CBE1
   CR      =$CBF4
   ADVNCE  =$CC16
   FIXBUF  =$CC80
   NXTIN   =$CCA6
   CLSKIP  =$CCC4
   VTAB    =$CCE6
   BASCALC =$CD04
   PRO     =$CD38
   BADY    =$CD6D
   DOVTZ   =$CD86
   ESCNEW  =$CDB9
   RTS3    =$CDED
   OUTPT1  =$CE06
   NCOMMA  =$CE47
   CTLTBL  =$CE5A
?  ESCC    =$CE60
?  CTLG    =$CE68
?  CTLK    =$CE70
?  CTLO    =$CE78
?  CTLS    =$CE80
?  CTLW    =$CE88
?  CTL[    =$CE90
?  CTL_    =$CE98
?  CTLZ1   =$CEAC
   TABLE   =$CEBA
   APVTAB  =$FC22

M  WRITE   =$C724
M  ENTR    =$C73E
M  MOVECZM =$C778
M  CO1     =$C798
M  XFERCZM =$C7C5
M  OUTPUT  =$C7FA
   IEXIT   =$C817
   SETUP   =$C827
   A2E     =$C86B
   LOOP    =$C8BA
   NFSKIP  =$C90E
   SWDTBL  =$C934
   RSKIP1  =$C962
   GETLN   =$C9E0
   SAVOLD  =$CA04
   KBDWAIT =$CA52
   NORMOUT =$CA8B
   NTSHFT  =$CACA
   INDONE  =$CAF2
   CSRMOV  =$CB10
   PREAD   =$CB42
   RDKEY   =$CB59
   RTS5    =$CB7F
   VIDEO   =$CB89
   LOLITE  =$CB9D
   HOME    =$CBAC
   BELL    =$CBD4
   ADVANCE =$CBE5
   CR1     =$CBF9
   CRLF    =$CC26
   FIXIT   =$CC8E
   CLREOL  =$CCAC
   BS      =$CCCC
   VTABZ   =$CCE9
   NT512   =$CD0F
   FMT1    =$CD47
   BADX    =$CD78
   RAWVID  =$CD8B
   ESC2    =$CDC3
   VIDOUT  =$CDEE
   FIXCSR  =$CE17
   RTS2    =$CE4F
?  ESC@    =$CE5A
?  ESCD    =$CE62
?  CTLH    =$CE6A
?  CTLL    =$CE72
?  CTLP    =$CE7A
?  CTLT    =$CE82
?  CTLX    =$CE8A
?  CTL\    =$CE92
?  DSPTBL  =$CE9A
?  CTLZ2   =$CEAE
   SCLTBL  =$CF3A
   SETKBD  =$FE89
```

Section F.2 ASSEMBLY CROSS REFERENCE FOR UltraTerm

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

DEF	LABEL	REFERENCES
31	A1H	335 339
30	A1L	327 333 337
525	A2E	515 522
33	A2H	336
32	A2L	334
35	A4H	331
34	A4L	328 329
1110	ADVANCE	1505 1537
1139	ADVNCE	1112
44	APVTAB	834
109	ATCHR	114
106	ATDFLT	113 114
111	ATHI0	
108	ATHI1	
110	ATINV0	
107	ATINV1	
113	ATLRG	666 667 669 670
662	ATRTBL	607 1384
114	ATSML	663 664 665 668
104	ATTREG	608 1397
1354	BADX	1352
1349	BADY	1347
1285	BASCALC	1169
63	BASEH	532 700 942 1178 1297 1311
62	BASEL	531 697 937 1173 1303 1308 1309
723	BASINP	301
22	BASL	276
779	BASOUT	409
1085	BELL	1516
1087	BELL1	1094
1089	BELL2	1091
75	BFORMT	563 593 1157 1273 1327 1331 1381
76	BGETLN	748 781 782 798
397	BINIT	293
79	BINV	1029 1037
77	BKEYBD	502 853 858 859
78	BLCCON	869
789	BOUT	784 787
71	BPOFF	489
1248	BS	1506 1517
471	BSTART	300 408
613	BSTART1	582
70	BSTATE	
66	BYTE	790 800 902 915 1122 1141 1427 1444
13	C0	14 62 63 64 65 66 67 68 73
14	C000	122 217 432 438 507 720 910 927
340	C01	338
347	C03	345
983	CAPSLK	734 1211 1426
20	CH	823 1468 1475 1483
64	CHORZ	533 676 816 924 1115 1126 1132 1140 1143 1220 1249 1251 1256 1257 1353 1469 1474 1485 1487
975	CHRPUT	1137 1226

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

```
DEF   LABEL     REFERENCES
1223 CLEOL2     1237
1231 CLEOL2     1073  1186  1221
1066 CLEOP1     1075
541  CLOOP1     557
1219 CLREOL     1064  1196  1509  1538
1063 CLREOP     1510  1520
269  CLRX       255  264  267
1052 CLSCRN     620  1504  1521
1235 CLSKIP     1229
216  CNOO       440  441  442  443  444  445  446
212  COMMON     179  189  199  209
1437 CONTROL    1367  1410  1420  1458
1121 CR         1522
1124 CR1        1057  1148
1147 CRLF       1117  1123  1478
552  CSLOOP1    555
920  CSRMOV     795  964
581  CSTART     479
27   CSWH       403
26   CSWL       402
1516 CTLG
1517 CTLH
1518 CTLI
1519 CTLJ
1520 CTLK
1521 CTLL
1522 CTLM
1523 CTLN
1524 CTLO
1525 CTLP
1526 CTLQ
1527 CTLR
1528 CTLS
1529 CTLT
1503 CTLTBL     1440  1442
1530 CTLU
1531 CTLV
1532 CTLW
1533 CTLX
1534 CTLY
1535 CTLZ
1557 CTLZO
1558 CTLZ1
1559 CTLZ2
1560 CTLZ3
1536 CTL[
1537 CTL\
1538 CTL]
1539 CTL^
1540 CTL_
21   CV         828  841  1462  1480
65   CVERT      534  830  840  1058  1065  1150  1151  1154  1262  1264  1269
                1348  1463  1465  1479
```

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

```
DEF   LABEL     REFERENCES
1467  CVOK      1464
90    DATA      573  938  944  971  1175  1180
118   DISPO     678  977
119   DISP1     680  980
1365  DOVTZ     1362
1546  DSPTBL
1254  ENDUP     1250
223   ENTER     290  291
279   ENTR      225  232
740   ESC       752
1407  ESC2      1405  1422
1504  ESC@
1505  ESCA
1506  ESCB
1507  ESCC
1508  ESCD
1509  ESCE
1510  ESCF
1402  ESCNEW    741
1411  ESCNOW
1424  ESCRD     1403  1421
458   EXIT      838  919
454   FINIT
1189  FIXBUF    756
1461  FIXCSR    725  791
1197  FIXIT     1192  1194
1492  FIXWDTH   796  835
73    FLAGS     527  562  583  592  780  854  855  857  868  870  876  1031
                1039  1041  1156  1272  1380
1038  FLGCLR    799  1328
1040  FLGSAV    1032
1030  FLGSET    749  1332
1200  FLOOP1    1215
1325  FMT1      1317
1315  FORMAT    1406  1552
747   GETLN     727  731  733  739  744
1095  GETX      1238
1335  GOTOX     1548
1342  GOTOY     1549
1000  GOXY      1539
52    HEIGHT    599  615  1068  1152  1181  1346
632   HGTBL     598
1389  HIATR     1554
1026  HILITE    1524  1560
1055  HOME      579  1534
126   HT1       143  149
130   HT2       155  161  167
475   IENTER1   457
464   IEXIT     498
42    IN        732  735  738  1201  1212
899   INDONE    861  880  882  892  894
275   INENTR    398
226   INFAKE
```

Appendix F

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

DEF	LABEL	REFERENCES
897	NOSHIFT	884
890	NOT@	888
758	NOTCR	754
865	NOTK	863
856	NOTLWR	852
762	NOTPICK	746 760
1292	NT512	1290
1482	NTEOL	1477
765	NTGETLN	736
1206	NTQTE	1204
874	NTSHFT	867
332	NXTA1	330
1213	NXTIN	1210
55	OLDCHAR	729 737 769 785
229	OUTENTR	401
1451	OUTPT1	792 918
407	OUTPUT	297
693	PAGSEL	677 1133 1232
451	PINIT	248 404
68	POFF	484 488 497 1014 1453 1455
1318	PRO	1530 1558
949	PREAD	253 508
1114	PREADV	1111 1131
24	PROMPT	1190
467	PSTART	917 950
652	PWDTBL	602
54	PWDTH	280 287 603 817 935 1116 1144 1255 1304 1351 1476
914	PWRITE	258
916	PWRITE1	721
1006	RAW	1527
1369	RAWVID	1551
963	RDKEY	750 766 1425
672	RDSCRN	761
510	RDSKIP	505
252	READ	240
89	REGSEL	571 710 923 940 969 1172 1177
494	RESTART	1557
37	RNDH	954
36	RNDL	952
681	RSKIP1	679
961	RTS0	456 1518 1525 1526 1528 1529 1533 1536
1312	RTS1	1263 1306 1310
1488	RTS2	1486
1429	RTS3	1413 1415 1417
1446	RTS4	1436
987	RTS5	985
945	RTS6	936
1118	RTS8	1142 1145
906	RTS9	
768	SAVOLD	764
1599	SCLTBL	1159
1012	SETATR	1532
822	SETCH	777 819

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

DEF	LABEL	REFERENCES
247	INIT	239
299	INPUT	296
23	INVFLG	585 1028 1036 1134
295	IO	292
47	IORTS	224
1363	JSTADV	1371 1434
1076	JVTAB	1059 1069 1153
83	KBD	265 803 809 956
84	KBDSTRB	807 813 871 960
808	KBDWAIT	810
951	KEYIN	957 959 965
955	KEYIN2	953
849	KEYSTAT	268 958
29	KSWH	400 793
28	KSWL	399
1357	LEAD	1550
1003	LEADIN	1535
1556	LEADTBL	
1149	LF	1507 1519
12	LINEHZ	173 183 193 203
1378	LOATR	1553
1034	LOLITE	1523 1559
570	LOOP	577
814	LSTFIX	802 804 806 812
100	MB132	625 629 630
99	MB256	546 624 625 626 627 628 629 630
102	MBANK	460 684
101	MBSVS	623 742 824 1163 1289 1321
93	MCPBITS	701
91	MCREG	462 548 561 597 687 704
51	MODE	459 463 545 560 596 609 673 675 683 685 702 743 825 1162 1286 1322 1323
622	MODTBL	595
305	MOVE	244
319	MOVEC2M	314
326	MOVELOOP	341
352	MOVERET	350
323	MOVESTRT	317
97	MPADDR	99
94	MPBANK	99 100 101 102
96	MPCLOCK	100
95	MPVIDEO	99 100 101
58	MSLOT	477 558 578 611 689 775 839 1096 1138 1216 1230
1608	MULTBL	1275
56	NO	461 476 547 559 591 686 703 708 921 968 1170 1320 1396
1166	N512	1164
1484	NCOMMA	1470 1472
528	NEWFMT	485 1333
589	NEWFMT2	529
610	NFSKIP	606
797	NOCSR	794
837	NORMOUT	826

ASSEMBLY CROSS REFERENCE FOR ULTRATERM

DEF	LABEL	REFERENCES
45	SETKBD	833
481	SETUP	478
487	SETUP1	482
46	SETVID	832
928	SHUTUP	932
85	SPKR	1092
934	SSKIP	925
41	STACK	
67	START	535 1161 1167 1284
1563	STATE	1454
262	STATUS	242
1129	STOADV	1364
134	SW1	145 151
138	SW2	157 163 169
642	SWDTBL	600
53	SWDTH	601 617 1236
154	T128	1596
160	T132	1593
166	T160	1581
142	T80	1575 1584 1587 1590
148	T96	1578
1573	TABLE	572
57	TEMPX	1271 1277 1281 1339 1350 1386 1395
586	TINIT	616 618 619
1261	UP	1508 1540
1009	VIDEO	1531
1432	VIDOUT	1373 1375 1547
43	VIDWAIT	
831	VSKIP	829
1268	VTAB	1060 1077 1187 1355 1466
1270	VTABZ	1071 1184
1280	VTLOOP	1283
19	WNDWDTH	605 1494
257	WRITE	241
979	WSKIP	976
*	X	774
172	X24	1576 1579 1582
182	X241	1585 1594
192	X32	1588 1597
202	X48	1591
360	XFER	245
377	XFERAZP	371
372	XFERC2M	367
386	XFERSZP	370 375
1496	XLTBL	1419
25	XSAVE	281 726 1130 1136
*	Y	553 1419 1454
895	ZERO	886

```
2    * This listing has been provided to assist programmers in
3    * developing software for the Ultraterm. If further information
4    * is needed, consult the Ultraterm software guidelines document.
5    *
6    * These screen drivers consist of the following routines
7    *
8    * INIT          initializes the Ultraterm in a given format
9    * GOTOXY        Calculates an X Y position on the screen
10   * SCROLL        Scrolls the screen up by one line
11   * STOADV        Stores a character on the screen and advances
12   * CURSOR        Puts the cursor on the screen
13   * CSROFF        Removes the cursor from the screen
14   *
15   * The Ultraterm has 8 different screen formats; the table below,
16   * lists these formats and their associated number
17   *
18   * Format #        Description
19   *    0             80 x 24 non-interlaced
20   *    1             96 x 24 non-interlaced
21   *    2            160 x 24 non-interlaced
22   *    3             80 x 24 interlaced
23   *    4             80 x 32 interlaced
24   *    5             80 x 48 interlaced
25   *    6            160 x 24 interlaced (used for 132 x 24)
26   *    7            128 x 32 interlaced
27   *
28   *
29   CH        EQU   $01          ; NOTE: CH AND CV SHOULD BE MAINTAINED
30   CV        EQU   $02          ;       BY THE DRIVING PROGRAM
31   PAGE      EQU   $03
32   NO        EQU   $04
33   MSLOT     EQU   NO+$01
34   MODEMASK  EQU   $06
35   FORMAT    EQU   $07
36   YSAVE     EQU   $08
37   *
38   MODE      EQU   $478
39   *
40   START     EQU   $6F8-$C0
41   FLAGS     EQU   $7F8-$C0
42   *
43   DEV0      EQU   $C080
44   DEV1      EQU   $C081
45   DEV2      EQU   $C082
```

```
                    47  * INITIALIZATION
                    48  *
                    49  * ENTER WITH SLOT IN A AND FORMAT IN Y
                    50  *
                    51  INIT
8000: 84 07         52          STY  FORMAT     ; SAVE FORMAT
8002: 09 C0         53          ORA  #$C0       ; MAKE $CN
8004: AA            54          TAX
8005: 86 05         55          STX  MSLOT
8007: 0A            56          ASL             ; MAKE $N0
8008: 0A            57          ASL
8009: 0A            58          ASL
800A: 0A            59          ASL
800B: A8            60          TAY
800C: 84 04         61          STY  N0
800E: AD FF CF      62          LDA  $CFFF      ; BANK OFF ROMS
8011: B1 04         63          LDA  (N0),Y     ; SELECT CN00
8013: A5 07         64          LDA  FORMAT     ; INIT WITH FORMAT
8015: 9D 38 07      65          STA  FLAGS,X
8018: 20 02 C8      66          JSR  $C802
801B: AD 78 04      67          LDA  MODE       ; CREATE MODE MASK
801E: 09 10         68          ORA  #$10
8020: 85 06         69          STA  MODEMASK
8022: 60            70          RTS
                    71  *
                    72  * SCROLL ROUTINE
                    73  *
                    74  * SCROLLS SCREEN AND RECALCULATES START
                    75  *
                    76  SCROLL
8023: A6 05         77          LDX  MSLOT      ; GET VARIABLE INDEX
8025: 18            78          CLC
8026: A4 07         79          LDY  FORMAT     ; ADD SCREEN WIDTH / $10
8028: B9 BF 80      80          LDA  MTBL,Y     ; TO START,X
802B: 7D 38 06      81          ADC  START,X
802E: 9D 38 06      82          STA  START,X
8031: 20 B2 80      83          JSR  MULTIPLY   ; MULTIPLY BY $10
8034: 48            84          PHA
8035: A9 0C         85          LDA  #$0C       ; SELECT START HIGH REGISTER
8037: A4 04         86          LDY  N0
8039: 99 80 C0      87          STA  DEV0,Y
803C: A5 03         88          LDA  PAGE       ; SET START HIGH
803E: 29 0F         89          AND  #$0F
8040: 99 81 C0      90          STA  DEV1,Y
8043: A9 0D         91          LDA  #$0D       ; SELECT START LOW REGISTER
8045: 99 80 C0      92          STA  DEV0,Y
8048: 68            93          PLA             ; SET START LOW
8049: 99 81 C0      94          STA  DEV1,Y
```

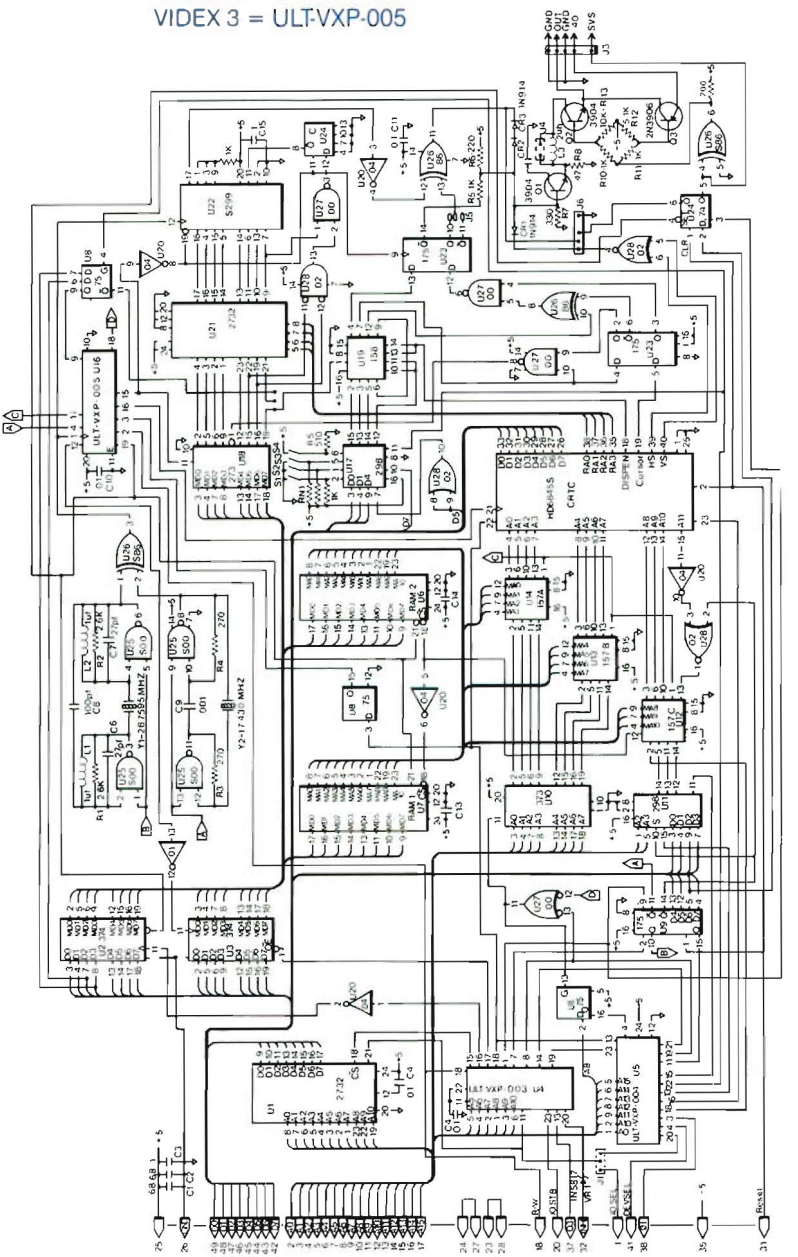
```
                        96  * GOTO X, Y
                        97  *
                        98  * CREATES NEW PAGE AND X WITH CH, CV, AND START,X
                        99  *
                       100  GOTOXY
804C: A6 05            101          LDX  MSLOT       ; GET VARIABLE INDEX
804E: A4 07            102          LDY  FORMAT
8050: B9 BF 80         103          LDA  MTBL,Y
8053: A8               104          TAY
8054: A9 00            105          LDA  #$00
8056: 18               106          CLC
                       107  MLOOP                    ; MULTIPLY BY WIDTH / $10
8057: 65 02            108          ADC  CV
8059: 88               109          DEY
805A: D0 FB            110          BNE  MLOOP
                       111  *
805C: 7D 38 06         112          ADC  START,X     ; ADD START OFFSET
805F: 20 B2 80         113          JSR  MULTIPLY    ; MULTIPLY BY $10
8062: 18               114          CLC
8063: 65 01            115          ADC  CH          ; ADD CH OFFSET
8065: AA               116          TAX
8066: 90 0D            117          BCC  PAGSEL      ; SELECT SCREEN PAGE
8068: E6 03            118          INC  PAGE
806A: 4C 75 80         119          JMP  PAGSEL
                       120  *
                       121  * STORE AND ADVANCE
                       122  *
                       123  * STORE CHARACTER IN A AT CURRENT POSITION
                       124  * INCREMENT X AND PAGE
                       125  *
                       126  STOADV
806D: 9D 00 CC         127          STA  $CC00,X     ; PUT CHARACTER IN A ON SCREEN
8070: E8               128          INX              ; ADVANCE SCREEN POSITION
8071: D0 11            129          BNE  EXIT
8073: E6 03            130          INC  PAGE        ; SELECT NEXT PAGE
                       131  PAGSEL
8075: 84 08            132          STY  YSAVE       ; SAVE Y
8077: A5 03            133          LDA  PAGE        ; PUT PAGE NUM INTO MCP
8079: 29 0F            134          AND  #$0F
807B: 05 06            135          ORA  MODEMASK
807D: A4 04            136          LDY  NO
807F: 99 82 C0         137          STA  DEV2,Y
8082: A4 08            138          LDY  YSAVE       ; RECOVER Y
                       139  EXIT
8084: 60               140          RTS
```

```
                    142  * MOVE THE CURSOR AND TURN IT ON
                    143  *
                    144  CURSOR
8085: 84 08         145           STY  YSAVE       ; SAVE Y
8087: A4 04         146           LDY  NO
8089: A9 0E         147           LDA  #$0E        ; SELECT CURSOR HIGH REGISTER
808B: 99 80 C0      148           STA  DEV0,Y
808E: A5 03         149           LDA  PAGE        ; SAVE CURSOR HIGH
8090: 29 1F         150           AND  #$1F
8092: 99 81 C0      151           STA  DEV1,Y
8095: A9 0F         152           LDA  #$0F        ; SELECT CURSOR LOW REGISTER
8097: 99 80 C0      153           STA  DEV0,Y
809A: 8A            154           TXA              ; SAVE CURSOR LOW
809B: 99 81 C0      155           STA  DEV1,Y
809E: A4 08         156           LDY  YSAVE       ; RECOVER Y
80A0: 60            157           RTS
                    158  *
                    159  *
                    160  * TURN CURSOR OFF
                    161  *
                    162  *
                    163  CSROFF
80A1: 84 08         164           STY  YSAVE       ; SAVE Y
80A3: A4 04         165           LDY  NO
80A5: A9 0E         166           LDA  #$0E        ; SELECT CURSOR HIGH REGISTER
80A7: 99 80 C0      167           STA  DEV0,Y
80AA: A9 FF         168           LDA  #$FF        ; PUT CURSOR OFF OF SCREEN
80AC: 99 81 C0      169           STA  DEV1,Y
80AF: A4 08         170           LDY  YSAVE       ; RECOVER Y
80B1: 60            171           RTS
                    172  *
                    173  *
                    174  MULTIPLY                 ; MULTIPLY BY $10
80B2: 48            175           PHA
80B3: 6A            176           ROR
80B4: 4A            177           LSR
80B5: 4A            178           LSR
80B6: 4A            179           LSR
80B7: 85 03         180           STA  PAGE        ; SAVE PAGE NUMBER
80B9: 68            181           PLA
80BA: 0A            182           ASL
80BB: 0A            183           ASL
80BC: 0A            184           ASL
80BD: 0A            185           ASL
80BE: 60            186           RTS
                    187  *
                    188  MTBL
80BF: 05            189           DFB  80/$10
80C0: 06            190           DFB  96/$10
80C1: 0A            191           DFB  160/$10
80C2: 05            192           DFB  80/$10
80C3: 05            193           DFB  80/$10
80C4: 05            194           DFB  80/$10
80C5: 0A            195           DFB  160/$10
80C6: 08            196           DFB  128/$10
```

```
--End assembly--

199 bytes

Errors: 0

Symbol table - alphabetical order:

   CH       =$01     ?  CSROFF   =$80A1   ?  CURSOR   =$8085      CV       =$02
   DEV0     =$C080      DEV1     =$C081      DEV2     =$C082      EXIT     =$8084
   FLAGS    =$0738      FORMAT   =$07     ?  GOTOXY   =$804C   ?  INIT     =$8000
   MLOOP    =$8057      MODE     =$0478      MODEMASK =$06        MSLOT    =$05
   MTBL     =$80BF      MULTIPLY =$8082      NO       =$04        PAGE     =$03
   PAGSEL   =$8075   ?  SCROLL   =$8023      START    =$0638   ?  STOADV   =$806D
   YSAVE    =$08

Symbol table - numerical order:

   CH       =$01        CV       =$02        PAGE     =$03        NO       =$04
   MSLOT    =$05        MODEMASK =$06        FORMAT   =$07        YSAVE    =$08
   MODE     =$0478      START    =$0638      FLAGS    =$0738   ?  INIT     =$8000
?  SCROLL   =$8023   ?  GOTOXY   =$804C      MLOOP    =$8057   ?  STOADV   =$806D
   PAGSEL   =$8075      EXIT     =$8084   ?  CURSOR   =$8085   ?  CSROFF   =$80A1
   MULTIPLY =$8082      MTBL     =$80BF      DEV0     =$C080      DEV1     =$C081
   DEV2     =$C082
```

Schematic Diagram

Note: VIDEX 1 = ULT-VXP-003
VIDEX 2 = ULT-VXP-004
VIDEX 3 = ULT-VXP-005



Theory of Operation

Your UltraTerm is the most sophisticated product we have produced to date. Since this appendix will describe its operation in some detail, you will find this to be the most technically complex text in this manual. To understand the operation of the card you will have to refer to the schematic diagram as you read this material. We suggest you make a photocopy of the schematic before you start. This backup copy will save a tremendous amount of wear and tear on the binding of your manual.

We will describe the operation of your UltraTerm by taking you on a guided tour from the Apple bus to the video output connector. Along the way we will point out the major building blocks of your card and describe their operation in some detail. In this appendix, as in the rest of the manual, we will refer to memory and I/O addresses as if your card is plugged into slot #3.

There is one aspect of the UltraTerm design which deserves some preliminary explanation. This is the concept of "pipelining". We use this term to refer to the temporary storage and delay of certain signals within the UltraTerm. Some elements of the UltraTerm, particularly the character generator EPROM, cannot produce valid outputs quickly enough to be useful during a single video character time (about 296 nanoseconds). For this reason, their outputs are stored and delayed until the next character time. (Even with pipelining, we still need to use EPROMs with access times less than 300 nanoseconds for the character generator.) Other signals, such as the video attributes, can be produced more quickly. These signals are also stored until they can be shifted out in synchronization with the character information. For example, the video serializer receives eight parallel data bits and begins shifting out the dots for one character while the information for the next character is being produced by the character generator EPROM.

Section Y.1 APPLE BUS INTERFACE

There are three programmable array logic devices, (PALS), shown on the schematic as Videx1, Videx2 and Videx3, on your UltraTerm. These devices decode the addresses on the Apple bus, generate device selection signals for the circuits on your UltraTerm, and provide timing and synchronizations signals. They also detect addresses in the range $CF00 to $CFFF and disable your UltraTerm's firmware as required when $CFFF is addressed. Videx1 generates strobe signals which enable input and output latches U2 and U3. These devices transfer data to and from the video refresh memory. Quad latch U9 stores the data for the mode control port. The CRTC and the video refresh memory are selected by strobe outputs from Videx1 and Videx2 and receive their data directly from the Apple bus. The device selected is determined by the two low-order

address bits from the Apple.

The eight low-order addresses for the video refresh memory are buffered by U10. The four high-order bits are generated by the mode control port (MCP), U9, and U11. If your UltraTerm is in the 512-byte addressing mode in order to emulate a Videoterm, the twelfth bit is set to zero and only the lower 2048 bytes of the video refresh memory are used.

The 2732A EPROM which contains the firmware has its outputs gated directly onto the Apple bus. Its outputs are enabled by a signal from Videxl.

Section Y.2 FIRMWARE INTERFACE

The firmware to operate your UltraTerm is contained in a 4K-byte 2732A EPROM, U6. The lower half of this IC contains seven versions of the code which appears at $CN00 to $CNFF, one for each slot. The segment of code which appears at each page is selected by address bits A8, A9 and A10.

There are 2K bytes of address space available for use in the co-resident memory space at $C800. However, the upper 1K bytes of this space is used by the video refresh memory. For this reason the firmware is split into two banks. These banks are selected with bit seven of the MCP. When the second bank of firmware is selected it overlays the Video Refresh Memory (VRM) at addresses from $CC00 to $CFE0. The first bank of the firmware always occupies the region from $C800 to $CBFF.

Section Y.3 CRT CONTROLLER

The CRTC, U15, is the central element of your UltraTerm. It is responsible for sequencing the addresses to the video refresh memory, displaying the cursor, generating synch pulses and controlling the display format. The programmability of this circuit is the key to generating the many different display modes you can use with your UltraTerm.

The CRTC appears as a pair of memory locations ($C0B0 and $C0B1) to your Apple. The data stored at the first address selects the CRTC register. The data stored at the second address will be transferred to the selected CRTC register. A complete description of the functions of the CRTC registers is included in appendix C.

Section Y.4 VIDEO REFRESH MEMORY

The VRM stores the ASCII codes which your UltraTerm converts to video signals. The memory is made up of two high-speed (100 nS) static RAM chips, U6 and U7. This memory must be made available both to the Apple, which stores the ASCII data, and to the CRTC, which reads the ASCII data and converts it to video signals. In order for the video display to

continue without interruptions, the CRTC must have priority in addressing the VRM. Otherwise, the video display would show black dashes as the video logic was denied access to the refresh memory. This problem is prevented on your UltraTerm by latching the addresses and data from the Apple and transferring them to the VRM when the CRTC is carrying out internal operations.

The addresses sent to the VRM are selected by multiplexer chips U12, U13, and U14. The timing PAL, Videx3 generates the signal which causes the multiplexers to select either the CRTC addresses or the Apple bus addresses.

If your UltraTerm is in the Videoterm emulation mode and is using the 512-byte addressing mode, the lower 9 bits of the VRM address are taken from the Apple address bus. The upper two bits are latched from Apple addresses A3 and A2 when the slot-dependent I/O locations are read.

When your UltraTerm is in the 256-byte mode, the upper four VRM address bits are taken from the lower four bits of the MCP, bits are set by simply storing the proper high-order address data (combined with clock and page select bits in the high nybble) into the MCP This method avoids much of the address-manipulation arithmetic required in the 512-byte mode.

RAM address arbitration (ensuring that the CRTC has priority) is accomplished by the timing PAL, Videx3. Latches U2 and U3 store the Apple's address and data while the CRTC is using the VRM for video display refreshing.

Section Y.5 CHARACTER GENERATION

The dot patterns which make up a displayed character are generated by combining the ASCII value for the character with the row address bits from the CRTC. The resulting address is used to fetch the dot pattern for one row of the character from the character generator EPROM, U21 The Standard/Alternate character set bit from the Video Attributes Register (VAR) is used to select either the upper or lower half of the 2732 EPROM. Latch U18 pipelines the address information to the character generator, delayed by one character clock time. The high-order bit of this latch determines which of the two sets of attributes will be used with the character. The state of this bit selects either one or the other of the two bit patterns in the VAR, U17. The functions of the different attribute bits are explained in chapter 8. The default attributes, set by the DIP switches S1-S4, are loaded into the attribute register whenever the Apple RESET line is pulled low.

The eight output bits of the character generator EPROM make up the first 8 dots of a 9-cell wide matrix for the displayed character. The ninth dot is normally off, resulting in a space between characters. Sections of U27 and U28 and U34 are used to select certain characters (the line drawing and graphics characters) which will have the eighth dot duplicated into the ninth dot position. This allows us to display graphics characters which are completely connected from character to character.

Section Y.6 TIMING GENERATION LOGIC

Your UltraTerm has two crystal-controlled clock oscillators. One generates a clock with a frequency of 28.7595 MHz, the other a signal with a frequency of 17.430 MHz. These oscillators consist of sections of U25 and their associated crystals and resistors. The 28.7595 MHz clock is used when you are in the 128, 132, or 160-column modes. The 17.430 MHz clock is used in the 80 and 96-column modes. These clock frequencies are the fundamental dot writing rate for the transfer of video dots to your video monitor. A character time consists of nine cycles of the dot clock. The clock which will be used is selected by bit 5 of the MCP. A high value for this bit selects the 28.7595 MHz clock. This dot clock is used to shift the bits from the character generator through the video serializer, U22.

The clock signals which synchronize the operations of your UltraTerm are generated by a nine-state counter which is part of PAL Videx3. Outputs from this chip control the reading and writing of data for the VRM and the selection of data from the Video Attributes Register.

Section Y.7 Video Combiner and Internal Video Switch (IVS)

The video signal from U22 and the composite inverse video attribute are combined in gate U26. The Highlight/Lowlight attribute and the sync signals are combined with the video signal by a diode mixer.

Transistors Q1 and Q2 form the internal video switch which selects either the Apple video or the UltraTerm video, depending on the status of IVS output, part of U26. These transistors and their controlling logic form the soft video switch which is controlled by bit 6 of the MCP. When this bit is low, the video from the Apple is selected.

The digital video, composite sync, and IVS outputs are sent to Molex connector J6 where they may be used by external devices.

Section Y.8 CONNECTORS AND JUMPER BLOCKS

Your UltraTerm has several connectors and jumper blocks which allow you to set default operating conditions. When you receive your board these jumpers are set to allow your UltraTerm to work properly with an Apple][. The functions of the jumpers and connectors is explained in the following section.

J1 This jumper plug is used to select either the standard IOSEL signal from the Apple or an internal IOSEL which always responds to addresses in the $C3XX memory region. When the internal IOSEL signal is used, the card will always work, even if the Apple IOSEL is inhibited. This is the case if your UltraTerm is used in an Apple //e with an 80-column or

extended memory card in the auxiliary slot. When the internal IOSEL is active, the INHIBIT line of the Apple //e is activated and the Apple 80-column firmware will be disabled.

J2 This jumper allows the selection of the standard or alternate character set by the high bit (bit 7) of the output character. It is normally set so that the character set is determined by bit 2 of the Video Attributes Register.

J3 This is the Video output connector. The cable to your monitor and to the Apple video output is connected here.

J4 The video waveform is controlled by this jumper. When the jumper is installed the video output pulses are square waves. Without the jumper plug, the pulses become triangular waves. The Apple Monitor /// and many other monitors will produce sharper characters with a triangular video waveform.

J5 Switching this jumper will invert the video output signal. This jumper is used in conjunction with J6 for special applications.

J6 This connector provides the composite Sync, Video and UltraTerm select signals. It is designed to be used with special video processing boards.

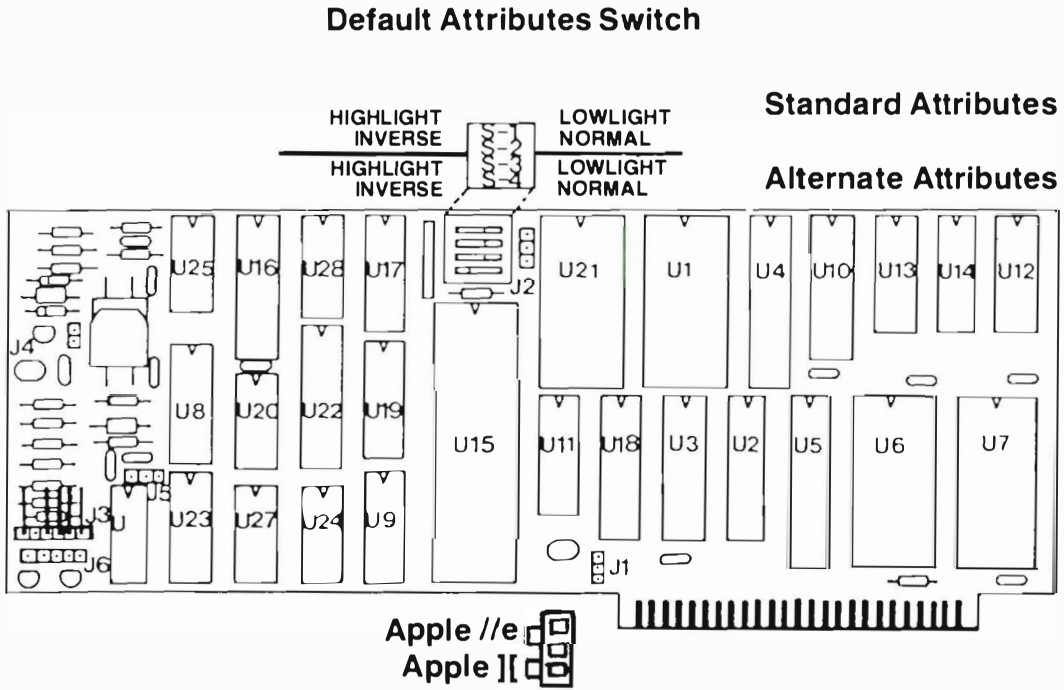


Figure Y.1 UltraTerm P.C. Board Layout

Glossary

APPLICATION PROGRAM An application program is a program which is written to accomplish a specific task. The program may be written in BASIC, Pascal or Machine Language. A payroll program or an accounting package are examples of applications programs.

ASCII This is an acronym for American Standard Code for Information Interchange. This standard defines the way the alphabet, numbers and control characters are encoded by your computer. The ASCII codes use only seven of the eight bits in a byte, so we use the last bit to set special attributes for a displayed character.

BANDWIDTH This is a measure of the range of frequencies which an electronic device can faithfully reproduce. If the bandwidth of a device such as a display monitor is too small, it cannot preserve all the information contained in the video signal.

BASIC This is an acronym for Beginner's All-purpose Symbolic Instructional Code. It is the computer language, originally developed at Dartmouth University, which is used on most personal computers. APPLESOFT is a version of BASIC.

BOOTSTRAP If you have an Apple with a disk system, when you first turn the power on, the computer will try to read the disk operating system (DOS) from the diskette. This procedure of reading a program into memory, then executing that program is called BOOTSTRAPPING.

CONFIGURE When you change certain variables or routines in a program to take advantage of special hardware features, such as your UltraTerm card, you are CONFIGURING a program.

CONSOLE This is the term the Pascal operating system uses to refer to the main text input and output system. On your Apple][, console input comes from the keyboard and the console output appears on your video display.

CRTC This is an acronym for Cathode Ray Tube Controller. This is an integrated circuit which automatically scans a block of memory, then converts the ASCII data in the memory to video signals. The CRTC also provides horizontal and vertical synchronization signals, and a cursor.

CURSOR The cursor is the solid or flashing block on your display which indicates where the next character that you enter will appear on the video display. Different programs may alter the appearance of the cursor by changing certain registers inside the CRTC.

EPROM This is an acronym for Electrically Programmable Read-Only Memory. An EPROM can be programmed in a special interface card, then used as a ROM by your UltraTerm card. While it is installed in your UltraTerm, the data in the EPROM cannot be altered. EPROMs can be erased by exposure to ultraviolet light, then re-programmed.

ELECTRONIC SWITCH An electronic switch can select one of two signals, depending on the logic level on its control input. On your UltraTerm an electronic switch selects either the UltraTerm video or the Apple video signal. Electronic switches have no moving parts to wear out and can easily be controlled by your computer.

FIRMWARE A program which is stored in an EPROM is firmware. It is called this because it is somewhere between hardware and software. Hardware cannot be changed, at least not without great difficulty and a knowledge of electrical engineering. Software is easily changed—if you know how to program your computer. Firmware can be changed, but it requires a special programming device. The firmware programs on your UltraTerm card control the operation of the card and its interaction with your computer.

HARDWARE Your Apple][and UltraTerm are hardware. The electronic circuits which go together to make a particular device are called hardware.

INTERLACE When your UltraTerm is in interlace mode, every other vertical scan is delayed by one-half of a horizontal scan time. Thus, every other complete screen will be one-half line lower than the previous field. The result is that there are twice as many horizontal scans on the screen. However, each scan will be refreshed only 30 times per second instead of 60 times. You may consider that your UltraTerm is writing all the even-numbered horizontal lines in one thirtieth of a second and all the odd-numbered lines in the next thirtieth of a second.

INVERSE VIDEO When your video display shows black characters on a light background, they are being displayed in inverse video.

KEYBOARD ECHO When your computer sends all characters that you enter on the keyboard to the output device (the video display), your system is using keyboard echo. This allows you to use output commands by simply typing them on the keyboard.

OPERATING SYSTEM This is the supervisor program that controls the use of the resources of your computer. The Pascal Operating System is responsible for processing input and output and executing commands which allow you to run utility or application programs. Apple DOS 3.3 is an operating system which allows floppy disks to be used with BASIC.

OVERSCAN A video display monitor uses a beam of electrons to excite the phosphor which produces the lighted dots on the screen. When the electron beam starts scanning off the left edge of the screen and continues scanning past the right edge of the screen, this is called overscan. Since the scan must occur in a fixed interval, overscan reduces the time available to display characters on the screen.

PERIPHERAL A peripheral is a separate piece of hardware which is connected to your computer to allow it to accomplish a specific task. A disk drive is a peripheral which allows you to store information on floppy disks.

PHOSPHOR The inside of the display screen of your video monitor is coated with a chemical compound which emits light when it is struck by an electron beam. This chemical compound often contains the element Phosphorus, and is called a phosphor for this reason.

PROMPT A special character or word that your computer displays when it is waiting for input is called a prompt. The prompt for Applesoft is the ']' character.

REGISTER This is simply a storage location whose contents affect the operation of a device inside or connected to your computer. The X and Y registers inside your Apple can control data storage operations. The registers in the CRTC used in your UltraTerm control the format of your video display.

RESOLUTION This is a measure of the smallest dot which a video monitor can display. If the resolution of a display is poor, the dots which make up a character will appear to merge together and the characters will be fuzzy. Resolution is often limited by the bandwidth of the display electronics.

UTILITY PROGRAM A general-purpose program which is designed to handle disk files or other types of computer data is called a Utility. Utilities generally do not care about the special significance to the data they handle—they simply move bytes around or set up data for other programs to use. The FID program provided with Apple DOS 3.3 is a utility program.

VIDEO DISPLAY This is the television-like device used to change the electrical signal generated by your UltraTerm into a visual display.

WORD PROCESSOR A program which allows you to enter, edit, store and display text is a word processor. Most word processors allow you to specify such details of the printed document as the margins, right, center and left justification and page length.

Index

RMA Form for UltraTerm

RMA #_____

Call (503-758-0521) for RMA#

Serial #_____

Previous Service RMA_____

Name_____

Organization_____

_____

Addr._____

_____

_____

Shipping Address:

Name_____

Addr._____

_____

_____

Shipping Instr._____

Phone # (days)_____

Phone # (evenings)_____

Date purchased_____

Received_____

System Configuration:

- ☐ Autostart
- ☐ Apple][plus
- ☐ Apple //e
- ☐ Old Monitor ROM
- ☐ Apple][
- ☐ Apple ///

Resident Language:

- ☐ AppleSoft
- ☐ Pascal
- ☐ Integer
- ☐ CP/M

Number of disc drives:_____

List on the back of the page all products installed in the Apple at the time the failure occurred, and any software that was in use.

For problems that occurred during installation, did you get a:

- ☐ power light?
- ☐ power-up beep?
- ☐ Display?

Were there any installation errors?

Does the problem occur with:

- ☐ 80-Col. Modes
- ☐ Wider Modes (128–160 Col.)

Describe, in detail, the nature of the problem.

Describe, in detail, the circumstances under which the problem occurred.

Does the problem occur only several minutes after powerup?
☐ Yes ☐ No

COMMENTS:

Corvallis, Oregon 97330
503/758-0521