OWNER'S MANUAL

Model 7710

Asynchronous Serial Interface

California Computer Systems

CCS MODEL 7710

APPLE II ASYNCHRONOUS SERIAL INTERFACE

OWNER'S MANUAL

Rev B
42000036-01

California Computer Systems
740 S. Milpitas Blvd.
Milpitas, CA 95035

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The CCS Model 7710 interfaces your Apple computer to such peripheral devices as printers, modems, video terminals, and other computers. The 7710 is available in two versions, the 7710-01 and the 7710-02, which differ only in the driver program that they contain in the two programmable read only memory integrated circuits (PROMs) on the card. The 7710-01 firmware (a program in PROM), is useful primarily as a printer driver. The 7710-02, which is designed for use with modems, contains firmware that enables the Apple to act as a dumb terminal. A third firmware version (7710-03) is essentially a patched version of the 7710-02 version, and allows for compatibility with the Apple IIe. 7710-02s shipped after July, 1983 contain the patched firmware. They can be recognized by the labels 7710-03 L and 7710-03 H on the PROM chips of the card. The 7710-01 firmware is compatible with the IIe without modification. Each of these firmware packages is available separately, allowing you to add capabilities with a minimum of new hardware.

If you plan to use the 7710 with a printer, the 7710-01 is usually the best choice. If, however, you want to use the 7710 to turn the Apple into a terminal or communicate with another computer through a modem, you will need the 7710-02. Many commercially-available communications programs having file transfer abilities and and other computer-to-computer communication capabilities are compatible with the 7710. In most cases these programs can be used with both the 7710-01 and the 7710-02. For a more complete description of these programs and their use with the 7710, refer to Chapter 6 of this manual. Remember, if you do not have one of these communications programs, you need 7710-02 if you are going to use the Apple as a terminal to another computer.

The 7710 features switch-selectable baud rates for compatibility with a wide variety of peripheral devices at optimum data rates. Both RTS/CTS and DTR/DSR handshaking are implemented. The 7710-01 is also capable of XON-XOFF software handshaking.

1.2 SPECIFICATIONS

SYSTEM INTERFACE

Internal:	Apple II Peripheral Slot 1-7
External:	EIA RS-232-C (1969) Configurations A through E Primary Circuits Only DCE Failsafe Input Circuits (Shorts and Opens)
TRANSFER MODE	Asynchronous Serial 7 or 8 Data Bits Odd, Even, or No Parity Bits 1 or 2 Stop Bits
DATA RATES	50 baud 75 baud 110 baud 134.5 baud 150 baud 200 baud 300 baud 600 baud 1200 baud 1800 baud 2400 baud 4800 baud 9600 baud 19200 baud External
PROGRAM MEMORY	256 Bytes PROM, Auto Power Down Replaceable with 2112 RAMs
REQUIRED POWER	+5VDC +12VDC -12VDC
FEATURES	Supports Interrupt Daisy Chain with On-Board Arbitration Logic DMA Daisy Chain Pass-Through Glass Epoxy (FR-4) PC Board Gold Plated Connector Fingers Solder Mask Both Sides of Board Component Silkscreen

1.3 USING THIS MANUAL

For users not familiar with the principles of serial interfaces in general and the RS-232-C standards in particular, the remainder of this chapter is a useful introductions to these concepts. All users of the 7710 are encouraged to read carefully Chapters 2,3,6, and 7 regarding the configuration, installation, and operation of the 7710. The remainder of the manual contains more technical information which may be ignored by the majority of 7710 users.

Chapter 4 contains information required only by users who plan to write their own assembly language drivers for the 7710, or those who wish to alter the existing drivers. Chapter 5, a discussion of the hardware design for the

7710, is intended for the curious and for those who need to trouble-shoot or modify the 7710. Chapter 6 is concerned with the use of the 7710 with the Apple Pascal operating system and the CP/M operating system. Chapter 7 should be read by those using the Apple IIe or those contemplating the use of a modem with the 7710. Appendix A contains miscellaneous technical information, including a schematic/logic diagram and a list of user-replaceable parts. Listings for the 7710-01 and 7710-02 firmware drivers are provided in Appendix B. Appendix C contains the cable connection requirements for a large number of serial devices. PROM, RAM, and interface check-out procedures are given in Appendix D. Appendix E is a guide for what to do if your interface does not work. It attempts to anticipate any problems you might encounter.

Throughout this manual, low-active signals are indicated by a minus before the signal name/mnemonic (e.g., -DEV SEL). Control characters are indicated by a ^ before the character (e.g., ^P). The symbol <cr> represents a carriage return.

1.4 SERIAL INTERFACES: BASIC THEORY

A computer is an expensive do-nothing unless you can give it data, instruct it what to do with the data, and then have it present the results. To help do this, peripheral devices are designed. But because computers can communicate with peripherals in a number of formats, another kind of device is necessary: interfaces. Interfaces translate between a computer, which inputs and outputs in one format, and a peripheral, which inputs and/or outputs in another format.

There are two major types of computer data transfer: parallel and serial. Parallel interfaces transfer words of data simultaneously on parallel data lines. In parallel communications, all data bits must be transmitted and received simultaneously. Simultaneous reception is assured only if the distance of transmission remains below a certain limit, typically about ten feet for a computer-to-peripheral cable.

The simultaneous reception problem is not present in serial data transfer; the transmitter disassembles words and sends them a bit at a time over a single wire and the receiver reassembles the arriving stream of bits into words. For this scheme to work, the receiver and transmitter must make identical assumptions about which part of the data stream represents which bit of which word. Two ways have been devised to do this: synchronous and asynchronous modes of transmission. During synchronous transmission, a pre-defined pattern of syncronization bits is sent out. When the receiver detects this pattern, the subsequent data is divided into pre-defined word-length groups. Highly precise synchronization of transmitter and receiver timing is thus required.

Asynchronous interfaces handle the problem differently. They add a start bit to the front of each word, plus one or more stop bits to the end. The total group of bits is then transmitted one bit at a time. Because the receiver resynchronizes at the beginning of each word, it can identify the data bits even if it is slightly out of sync with the transmitter.

In both synchronous and asynchronous modes, the receiver must know how fast the data is sent. Generally, as long as the receiver expects data at the rate the sender is sending it, the actual rate of transfer does not matter. Common usage defines many standard signal rates. Usually, they are an even multiple of 75 baud (bits per second, including overhead bits such as start and stop bits). A few other rates, 50, 110, and 134.5, are used by the industry giants. The RS-232-C standard for serial communication does not specify any standard signal rates, but suggests a practical upper limit of 20 kilobaud and indirectly establishes a theoretical upper limit of 50 kilobaud.

It is not enough for computers and peripherals to exchange data; each must also tell when the other is ready to transmit or receive. This is done with "handshaking" signals. Because a wide variety of handshaking schemes are possible, the Electronics Industry of America (EIA) created the RS-232-C interface specifications to let manufacturers know what to expect. Two "sides" were defined. Because one side of the interface is usually connected to some type of computer terminal, equipment at that end is called Data Terminal Equipment (DTE). Equipment at the other end is called Data Communications Equipment (DCE), because to transmit serial data over long distances via telephone wires a Modulator/Demodulator, or Modem, is needed. For short distances (less than several hundred feet), modems and telephone wires are not needed, but one side of the interface must be made to think it's DCE.

RS-232-C defines the necessary protocol between the DCE and the DTE for many possible configurations. (see Sections A.4 and A.5) The first five (A, B, C, D, and E) are most commonly used. Configuration A defines a one-way, transmit-only interface, which might be used by a simple serial keyboard. Configuration B is also a one-way, transmit-only interface, but it has more handshaking. A paper tape reader might use this type of interface. Configuration C is also one-way, but receives only, and might be used by a serial printer.

With Configuration D, we start getting into two-way traffic. Configuration D is HALF DUPLEX; although it can carry traffic both ways, it can only carry on way at a time. When a modem is used in this configuration, it is often called a "two-wire modem" because the telephone line is connected with only two wires. This type of interface is not often used. Configuration E is a FULL DUPLEX link, meaning the link simultaneously supports traffic in both directions. CRT terminals often use this type of link. A modem used in this configuration is called a "four-wire modem." The 7710 supports all five configurations. Additionally, a Data Terminal Ready (DTR) handshake line has been provided, and is normally used only on synchronous interfaces and some asynchronous printers.

TABLE 2.1 BAUD RATE SELECTION

Baud Rate	1	2	3	4	Baud Rate	1	2	3	4
50 Baud	ON	OFF	ON	ON	75 Baud	OFF	OFF	ON	ON
110 Baud	OFF	OFF	OFF	OFF	134.5 Baud	ON	ON	OFF	ON
150 Baud	ON	OFF	OFF	OFF	200 Baud	OFF	ON	OFF	ON
300 Baud	OFF	ON	OFF	OFF	600 Baud	ON	OFF	OFF	ON
1200 Baud	OFF	OFF	ON	OFF	1800 Baud	ON	OFF	ON	OFF
2400 Baud	ON	ON	OFF	OFF	2400 Baud	OFF	OFF	OFF	ON
4800 Baud	OFF	ON	ON	OFF	9600 Baud	ON	ON	ON	OFF
19200 Baud	OFF	ON	ON	ON	External	ON	ON	ON	ON

CHAPTER 2

SET-UP AND INSTALLATION

2.1 BOARD CONFIGURATION

Before you install and operate your 7710 Asynchronous Interface, you need to configure it and your peripheral to ensure that the 7710, your peripheral, and your Apple meet each other's expectations. The configuration, described in detail in this chapter, may be as simple as setting the baud rate switches and connecting the 7710 to your peripheral with a standard RS-232-C cable, or it may involve more complicated manipulations, such as constructing a cable or adaptor, replacing the PROM chips with RAM chips, or more rarely changing the data format that the 7710 uses.

2.1.1 Setting the Baud Rate

Your peripheral device manual should tell you at which baud rate or rates the peripheral operates. If the peripheral can handle several baud rates, choose the highest one that is also available on the 7710. Set your peripheral device to that baud rate, following its instructions. Then set the rocker switches on the 7710 for that rate, as shown in Table 2-1 (o means to push that side of the switch down).

*** NOTE ***

If your peripheral's baud rate signal is available as an output and your cable makes the signal available to the 7710 at RS-232-C connector pin 24, you can set the baud rate of the 7710 to the peripheral's baud rate by setting the baud rate switches to the code for External. The incoming frequency must be 16 time the baud rate; e.g., 300 baud requires a 4800 Hz external clock signal.

2.1.2 RAM JUMPER

If you plan to write your own driver program or you wish to alter the existing PROM driver, you must install RAM chips (2112) in place of the two PROM chips at positions U5 and U6 on the 7710 (U5 and U6 are also labelled HI BYTE and LOW BYTE). In addition, a jumper wire must be soldered between the pads marked RAM located just above the card connector fingers at the right side of the board. The jumper prevents the RAM contents from being continually destroyed by the PROM power-down feature. Section 4.3 describes in more detail the procedures for using RAMs in the 7710.

2.2 PERIPHERAL CONFIGURATION

Unless otherwise noted, the following instructions apply to both the 7710-01 and 7710-02.

2.2.1 PERIPHERAL HALF/FULL DUPLEX (7710-01 only)

If you are interfacing the 7710-01 to a video terminal that allows you to select between half and full duplex operation, select FULL DUPLEX. Your computer's firmware expects a full duplex keyboard/display. This means that after receiving a character from the video terminal's keyboard, the computer sends that character back out to the terminal's display, allowing a quick check that the computer received what you wanted it to receive. When a terminal is in the half duplex mode, it displays the character as it is typed in and also displayed the character again when it receives the echo from the computer, which results in two copies of that character being displayed on the terminal. Thus, typing in RUN while in half duplex mode results in RRUUNN being displayed on the video terminal.

2.2.2 PERIPHERAL LINE FEED

The 7710-01 firmware generates a Line Feed character each time a Carriage Return is sent. If your terminal or printer has an Auto Line Feed option (usually selected with a dip switch), make sure it is turned OFF. If it is turned ON your texts will be double spaced. The 7710-02 firmware does not generate a line feed after each carriage return. Therefore, your printer <u>must</u> have an Auto Line Feed option if you want to use it with the 7710-02 firmware.

2.2.3 PERIPHERAL UPPER CASE

Unless you are using an Apple IIe or an Apple II or II+ with a lower case adaptor, your computer expects all alphabetic characters to be upper case. If your terminal has an Upper Case Only option, enable it. (While the standard firmware converts lower case to upper case, it is a good idea to select Upper Case Only if your terminal has that option.

2.2.4 Peripheral Parity

On asynchronous links, data transfer is usually so reliable that no parity generation or checking is needed. The driver routine therefore defaults to using no parity. This situation can be altered, however, as described in Section 3.1.3. Whichever parity option you choose (even, odd, or none), the terminal must be set to match the command.

2.3 THE PERIPHERAL CABLE

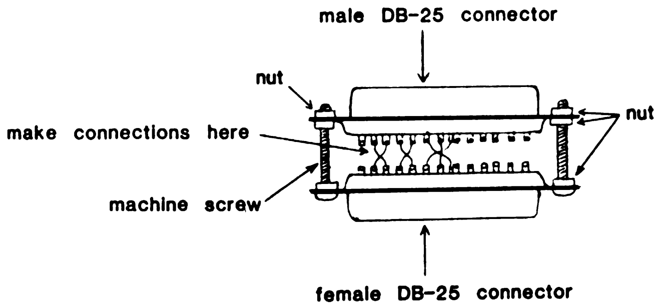
The 7710 is shipped with a short cable (approx. 20 cm) which can reach just a little beyond the back of your computer. You need to obtain a cable to connect your peripheral to the female DB-25 connector on the end of this cable. In some cases a standard RS-232-C cable is required, but it is often the case that you cannot simply plug your device into the 7710 with a standard RS-232 cable and expect it to work. This is due to the different ways in which various manufacturers implement the RS-232-C standard. For instance, most modems are wired in the DCE configuration (see Section 2.4), in which data is transmitted from the modem on pin 3 of the cable and data received on pin 2, while most printers are configured in the opposite orientation (DTE), with data coming into the printer on pin 3 and going out of the printer on pin 2. There are often other differences from one device to the next, and these differences must be taken into account when attempting to connect the 7710 to a serial device. Since CCS has intended the 7710 to be flexible enough to interface to almost any serial device, we do not attempt to supply any specific cable because of the large number of possible cable configurations that would be possible. However, in most cases a minimum of effort is required on the part of the user to make the proper connections; usually either a standard RS-232-C cable or a standard cable in conjuction with a commercially-available adaptor works. (Appendix C details the cable requirements for a large number of serial devices.) The 7710 handshakes its output data (e.g., to a printer) on pin 4 (RTS) and handshakes it input data (e.g., from a keyboard) on pin 20 (DTR).

2.4 DCE-TO DTE CONVERSION

The RS-232-C standard for serial data communication defines the characteristics of the interchange signals between Data Terminal Equipment (DTE) and Data Communications Equipment (DCE), such as a Modem. Each equipment type has its own set of handshaking signals assigned to specified pins of a DB-25 connector. You can think of a DTE device as piece of equipment that would "receive" data (e.g., a printer or video terminal), and a DCE device as something that is the "originator" of data (e.g., a computer). Since the 7710 is designed to interface with DTE equipment--terminals, printers, etc.-- its connector configuration is that of a DCE device. Most users of the 7710-02 firmware, however, will want to interface the 7710 to a modem or computer so the Apple can be used as a terminal, or DTE device. In this application, the 7710's connector usually needs to be rewired for DTE configuration or an adaptor must be used. (Some computers' RS-232-C connectors may take the DTE role, allowing the 7710 connector to be used as is. Consult you computer's manual.)

A DCE-to-DTE adaptor (often called a null modem) can be ordered from various electronics stores and computer stores, or can be built for much less cost by the user. This adaptor can be expensive enough that some users may opt to build it themselves. To make the DCE-to-DTE adaptor, you need one male and one female DB-25 connectors (cost is about $2 to $4), two small screws (with a few nuts), some short pieces of of wire and solder, and a soldering iron. Refer to the drawing below while building this adaptor. The DB-25 connectors have two sides, one containing the pins that are plugged into other DB-25 connectors, and a side with pins for making solder connections. Also, the connectors have screw holes on either side to facilitate fastening the two connectors together. To fasten with one and one-half inch screws, place the solder connections facing each other and the connector pins facing outward; solder wires between the connectors, making the pin-to-pin connections required. For a null modem these connections are as follows:

PIN 2 to PIN 3
PIN 3 to PIN 2
PIN 4 to PIN 5
PIN 5 to PIN 4
PIN 6 to PIN 20
PIN 20 to PIN 6
PIN 7 to PIN 7



Earlier versions of the 7710 manual were missing two of the the necessary connections: the 4 to 5 and the 20 to 6 connection.

If you will be using the External Baud Rate option on the 7710, you must also wire the 7710's External Clock input to the DTE External Clock line:
PIN 24 to PIN 15 (External Clock from DCE).

2.5 CARD INSTALLATION

****
* CAUTION *
****

Please read the following section carefully. Plugging the cable onto the card in the wrong orientation may result in damage to the 7710.

Before installing the interface in your computer, align pin 1 of the I/O cable connector (J2) with pin 1 of the mating connector on the 7710. Pin 1 of the cable connector is identified by the outside stripe (red or blue) on the cable and by a triangular mark on the connector. Pin 1 of the card connector is identified by a 1 on the silkscreen of the card. When all pins are properly aligned, push down firmly on the connector until you can no longer see the metal pins. Gently fold the ribbon cable on the diagonal so that the back panel connector is to the right of the board. Crease the fold only slightly; too much crease could fatigue and break the wires. This provides slack in the cable needed for strain relief.

The card is now ready to be inserted into the computer. Before installing the card, turn OFF the power to the Apple. Installing or removing any peripheral card with the power on can damage the Apple, your peripheral card, or both. After removing the cover from the Apple, you have access to eight 50 pin connectors. They are numbered 0 through 7 from left to right. Place the 7710 in any of these connectors, with the exception of slot #0, the leftmost. In the Apple II and Apple II+ slot 0 is reserved for PROM and RAM cards. In the Apple IIe, where slot 0 is in a different position on the motherboard than in earlier versions of the Apple II, it is reserved for 80 column video boards.

Although the usual position for the 7710 is slot 1, this can vary depending on your system's configuration. (See Chapter 5 for a discussion of proper slot positioning of the 7710 under Apple Pascal and CP/M) Plug the 7710 into the appropriate slot with the component side of the 7710 (side with the black integrated circuits) to the right. Run the I/O cable through one of the slots in the back of the Apple so the RS-232-C connector is outside the computer. Replace the computer cover, and you are ready to try out the board.

CHAPTER 3

OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter covers the procedures for using the 7710-01 and the 7710-02. As you pursue the procedure, remember that the two versions of the card are designed for different purposes; these are outlined in Chapter 1. The 7710-01 contains a general purpose input/output driver program most often used to interface the 7710 with a printer. Because the 7710-01 is not equiped with the terminal mode (found in the 7710-02), additional communications software (See Chapter 7, paragraph 7.2.2) is required to use it with a modem or in any other manner to make the Apple a terminal. On the other hand, the 7710-02 has communications capability, and is best used with a modem or in a configuration using the Apple as the terminal for another computer. The 7710-02 can be used with a printer, but lack some of the flexibility of the 7710-01.

Although there has been some confusion in the past about how one determines which version of the 7710 one has purchased, there are some simple ways to tell the difference. The 7710-01 is shipped in a box labelled 7710A or as 7710-01, while similar boxes containing 7710-02s will be labelled 7710-D or 7710-02. 7710 cards shipped after July, 1983 have paper labels on the PROM chips which identify the card; 7710-01 cards have PROMs labelled 7710-01 L and 7710-01 H, while 7710-02 cards are labelled 7710-02 L and 7710-02 H or 7710-03 H and 7710-03 L. If you bought your 7710 used, you should be aware that the previous owner might have exchanged the two PROM chips that contain the driver program. Since the only difference between these two versions is the program they contain in their PROMS, your card may be labelled 7710A but contain the 7710-02 (or 7710-03) PROMS, making it in reality a 7710-02. This brings us to the most certain way to tell which version of the card you own, which is to use the Apple's system monitor to list the contents of the PROMs on the video monitor and then compare that listing with the listings in Appendix B of this manual. Instructions for doing this are given in Appendix D of this manual.

The 7710-01 is relatively simple to use. One command from either BASIC or the system Monitor causes the Apple to send output to or expect input from the peripheral interfaced to the 7710. Simple procedures also allow you to alter the characters per line and the data format parameters.

The 7710-02 in the non-terminal mode operates similarly to the 7710-01, though it does not support the same parameters. Operation of the 7710-02 in the terminal mode is described in Section 3.2.

3.2 7710-01 OPERATION

If you are using the 7710 as a printer interface, you will need to tell the Apple when to stop sending output to the screen and where to send it so that the printer receives it. To establish a printer or other peripheral interfaced through the 7710 as the destination of output from the Apple, enter one of the following commands (n equals the 7710's slot number) from the keyboard.

```
PR#n     (from Integer or Applesoft Basic)
n^P      (from Monitor)
```

All subsequent output from the computer are now handled by the firmware's output routines. Anything typed at the keyboard thus goes to the printer (or other peripheral connected to the 7710), and does not appear on the screen. To restore the screen as the output destination, type in PR#0 from BASIC of 0^P from the monitor.

The preceeding paragraph tells how to use the 7710 from immediate mode (i.e., not within a program) using the Apple DOS operating systems. Many users of the 7710 need to send output to a printer under program control. The following BASIC program illustrates the technique for using the 7710 from within a BASIC program. Note that the same commands are used as when output destination is altered from the keyboard, although both the IN#n and PR#n commands are preceeded by a control D character, which is necessary to prevent the IN# and PR# commands from disconnecting the Disk Operating System (DOS).

```
10 D$=CHR$(4):  REM  SETS D$ EQUAL TO CONTROL D
20 INPUT "IN WHICH SLOT IS THE 7710 INSTALLED? (1-7) ";SLOT
30 PRINT D$;"PR#";SLOT : REM TELLS APPLE TO SEND ALL SUBSEQUENT OUTPUT TO SLOT
   SELECTED BY LINE 20
40 PRINT "THIS SHOULD BE SENT TO THE PRINTER"
50 PRINT D$;"PR#0" : REM SEND SUBSEQUENT OUTPUT TO THE SCREEN AGAIN
60 END
```

3.2.2 INPUT

If you are using the 7710 to interface a keyboard, paper tape reader, or other input device to the Apple, you need to tell the Apple when to stop expecting input from the main keyboard and where to go for future input. To establish the peripheral interfaced through the 7710 as the source of input to the Apple, enter one of the following commands (n = the 7710's slot number) from the main keyboard.

```
IN#n    (from either Integer or Applesoft Basic)
n^K     (from Monitor)
```

Either of these commands causes your computer to go to the 7710 firmware's input routines for all subsequent input. The Apple keyboard is now virtually "dead". The Apple will respond only to Reset and control C entered from the keyboard. Therefore, you should enter the PR# command before the IN# command if you wish to both input and output through the 7710. To restore the Apple keyboard as the input device, enter IN#0 or 0^K from the new input device (if it has a keyboard) or under program control, or enter RESET or ^C from the Apple keyboard.

The following program illustrates the use of the IN#n command to take input from an external device under the control of a BASIC program.

```
10 D$=CHR$(4) : REM SET D$ EQUAL TO CONTROL D
20 INPUT "IN WHICH SLOT IS THE 7710 INSTALLED? ";SLOT
30 PRINT D$;"IN#";SLOT
40 REM AT THIS POINT THE APPLE RESPONDS TO COMMANDS TYPED ON THE EXTERNAL
   DEVICE
50 INPUT A$: REM BASIC HALTS HERE AND WAITS FOR INPUT, WHICH MUST NOW COME
   FROM THE EXTERNAL DEVICE
60 PRINT D$;"IN#0" : REM RETURN INPUT CONTROL TO APPLE'S KEYBOARD
70 PRINT A$: REM PRINT THE STRING OBTAINED FROM THE EXTERNAL DEVICE
80 END
```

3.2.3 Changing 7710-01 Parameters

There are two default parameters in the standard driver (7710-01) driver, not the 7710-02 driver) which can be changed AFTER an IN#n/n^K or PR#n/n^P command has been executed and at least one character has been sent or received by the 7710.

3.2.3.1 Data Parameters

The default value for the ACIA operating mode command is $11 (decimal 17), which specifies an 8 bit word, 2 stop bits, no parity, and no interrupts. (Refer to paragraph 4.1 for an explanation of the ACIA and its operation.) Although most serial devices can be set to conform to this format, you can alter this data format. To change the operating mode (data format), POKE the desired command (derived from table 4.1) into location 49280 + 16*n (hex $C080 + $n0 or -16256 + 16*n in integer BASIC), where n is the slot number. Do so only after one character has been sent or received.

The desired command is determined from table 4-1 in the following manner. The number you need to poke into the ACIA command register is the decimal equivalent of an 8 bit binary number. Each of the 8 bits are set according to your requirements. Bit 8 of this command byte determines whether Receive Interrupts are enabled or disabled. Bits 5 and 6 determine whether transmit interrupts are enabled and also whether the RS-232-C Clear To Send line is set high or low. Bits 2, 3, and 4 determine the data format. For instance if bits 2,3, and 4 are all zeros, this selects 7 data bits, even parity, and 2 stop bits. Bits 0 and 1 determine whether an ACIA master reset is done and also set the clock frequency. Unless you are writing your own driver, bits 5, 6, and 7 should be set to 0, bit 0 should be 1 and bit 1 should be 0. When you change the data format you should set bits 2,3, and 4 to the desired pattern. For instance, if you want 8 data bits, no parity, and 1 stop bit, the total bit pattern is as follows:

```
bit number--->   7  6  5  4  3  2  1  0
                 ----------------------
bit value --->   0  0  0  1  0  1  0  1 = decimal 21
```

The data format you poke into the appropriate address remains in effect until another IN#n or PR#n command is encountered, at which time the default value of $11 is reinstated. This method of changing the data format works only with the 7710-01 firmware driver; the procedure for changing the data format when using the 7710-02 firmware is different (See Paragraph 3.2.3, Changing 7710-02 Parameters).

Application software that does not provide you with the option to change data format may not work with the 7710 if you are attempting to communicate with a serial device that does not accept the default 8 data, 2 stop, no parity setting of the 7710.

3.2.3.2 Characters per Line

After 255 characters have been printed on one line, the 7710-01 firmware will automatically initiate a carriage return/line feed sequence. If you want some other number of characters per line, simply POKE the desired number into locate 1528 + the slot number. This number must be greater than 0 and less than 255. From BASIC, the keyboard command sequence to do this is as follows:

```
PR#n
PRINT CHR$(0);
POKE  1528+n,c     (n=slot number and c=characters/line)
```

(Some printers do not print characters until a complete line ending with a carriage return is received.)

The following example shows how to change the number of character per line under program control. Line 50 prints a null character without a carriage return, initializing the interface before the new value is POKEd to the character-per-line location. Any PRINT statement could be used in this line, but non-null characters would be printed, which would be undesirable in most cases.

```
10 D$=CHR$(4) : REM SET D$ EQUAL TO CONTROL D
20 INPUT "NUMBER OF CHARACTERS PER LINE?";C
30 INPUT "SLOT NUMBER?";S
40 PRINT D$;"PR#";S
50 PRINT CHR$(0);
60 POKE 1528+S,C : REM CHANGE CHARACTERS PER LINE
70 REM  REPLACE THIS LINE WITH LINES FOR YOUR OWN  PROGRAM
80 PRINT D$;"PR#0"
90 END
```

3.2.4 Freezing Terminal Display (^S)

If your serial device is capable of sending characters to the 7710, you can temporarily stop the 7710 from sending out data by having your serial device send a ^S to the 7710. Upon receiving the ^S, the 7710 halts until it receives any other character, at which time it resumes transmitting.

3.3 7710-02 OPERATION

THE FOLLOWING INSTRUCTIONS APPLY ONLY TO THE 7710-02. THEY ARE NOT APPLICABLE TO THE 7710-01.

The 7710 Asynchronous Serial Interface with 7710-02 firmware allows the Apple to function as an ordinary computer terminal when connected to either a modem or another computer. In the terminal mode of operation, the firmware supports both full- and half-duplex modes and the BREAK signal. The user can switch from a non-terminal mode to the terminal mode with a few commands entered from the Apple keyboard or from an external device. In the non-terminal mode, the 7710-02 firmware allows the Apple to interface to such peripherals as printers and terminals. Input to the Apple can be taken from the 7710 instead of the Apple's keyboard and output from the Apple can be sent to the 7710 as well as the Apple's display.

3.3.1 Normal I/O Mode

In the non-terminal mode, the 7710-02 responds to the PR# and IN# commands, as well as the Monitor equivalents, as described in Paragraph 3.1.

3.3.2 TERMINAL MODE

After the IN# and PR# commands, the Apple still functions as a computer; i.e., its "brain" is still connected. In the Terminal Mode, the Apple simulates a computer terminal, its brain, in effect, disconnected. The Terminal Mode can be entered and exited by a number of commands, either entered on the Apple's keyboard or sent by the external device.

3.3.2.1 Keyboard Commands

The terminal Commands entered from the keyboard are always prefixed by ^A. They can be used only after an IN#n command and are the only keyboard commands, besides Reset and ^C, that the Apple recognizes after IN#n has been entered. Do <u>not</u> use the PR#n command if you want to enter terminal mode.

Full-Duplex Command (^A^F)

The Full-Duplex command allows your Apple to operate as a full-duplex terminal. In the full-duplex mode, any character typed on the Apple keyboard is sent to the external device but is not directly sent to the Apple's screen. If the external device sends that character back to the 7710 (echoes), it is displayed on the Apple's screen. The full duplex mode is often preferred since it confirms the communication's reliability. The receiving device, however, must be capable of echoing the character.

To enter full duplex mode, type:

IN#n (where n is the slot number)
^A^F

Half-Duplex Command (^A^H)

The Half-Duplex command causes your Apple to mimic a half-duplex terminal. In the half-duplex mode, a character typed on the keyboard is sent to both the Apple's screen and the external device. Thus, the user sees exactly what he has entered. If you use the half-duplex mode with a receiving device designed for full-duplex, every character on the screen is ddoouubblleedd, since the receiving device echos the characters it receives. To enter half-duplex mode, type IN#n followed by carriage return and ^A^H.

Break Command (^A^S)

While many terminals in time-sharing systems have a BREAK key, the Apple does not. This command allows the Apple to send out the BREAK signal. It is needed only when communicating with an external device that requires the BREAK signal. Turn ON the BREAK signal by entering ^A^S. Turn it OFF by pressing any key other than Control or Shift.

Exit Command (^A^X)

To exit from the terminal mode, enter ^A^X, which causes a backslash to appear on the screen. If a PR# command is in effect, it does not change. Input, however, is now taken from the Apple's keyboard.

Reset and ^C would also return the Apple to normal operations; however, any program running would be terminated. After Reset, both PR# and IN# are set to their default values, IN#O and PR#O.

3.3.2.2. Commands from External Devices

Remote Mode (^R)

When the Apple is operating in the terminal mode, an external device can connect and disconnect the Apple's brain. If the Apple receives ^R while in the terminal mode, it operates in a Remote Mode to the external device. Any input coming from the external device goes to the Apple's brain. Thus, the external device can control the Apple as if its instructions come from the Apple II's keyboard. When the PR#n command is in effect (where n = the 7710's slot number) the Apple echos what it receives from the external device.

Terminal Mode (^T)

After ^R has been issued, ^T returns the Apple to the terminal mode. It essentially cancels the Remote Mode command and reinstates the Apple as a half-duplex or full-duplex terminal. If the 7710 has been accessed by an IN#n command, but the terminal mode has not yet been invoked, ^T sent by an external device turns the Apple into a half-duplex terminal.

3.3.3 CHANGING 7710-02 PARAMETERS

Some of the 7710-02 firmware parameters (e.g., serial data format) can be changed by BASIC POKE commands or monitor commands. To change the parameters, you must first initialize the 7710 board by using the IN#n or PR#n commands or their monitor equivalents, n^P or n^K. Any subsequent invoking of the IN#n or PR#n commands or their monitor equivalents reinitializes the 7710, causing the parameters to return to their default values.

Note that the 7710-02 firmware does NOT support the standard 7710-01 parameter for characters-per-line. This means that a line feed is not appended to each carriage return sent. If you are using the 7710-02 with a printer, you will get all of your printing on one line unless whatever software you are using provides the line feed or your printer supplies the line feed. Most printers have a DIP switch somewhere which can be set to enable the appending of line feeds to carriage returns.

3.3.3.1 Serial Data Format

As with the 7710-01, the 7710-02 default is 8 data bits, no parity, and 2 stop bits. This can be changed by POKEing the appropriate value from Table 4-1 into memory location $7F8 + slot number (decimal 2040 + slot). This procedure is different from the procedure used with the 7710-01 driver. However, for both firmware versions this poke must be issued after at least one character has been sent or received. Refer to paragraph 4.1 for details about the ACIA registers.

As in the case of the 7710-01, application software that will not allow user specification of data format may not work if you are using a serial device that will not accept the default format.

CHAPTER 4

PROGRAMMING INFORMATION

This chapter contains the information you need if you plan to write your own assembly language driver for the 771Ø interface, or if you plan to make modifications in the code of the existing driver. Included are discussions of the 685Ø Asynchronous Communications Interface Adapter (ACIA) registers, register addressing, and general driver requirements. Instructions are also provided for saving the driver on disk and loading it into RAM. Refer to Chapter 2, paragraph 2.1.2 for instructions on replacing the PROMs with RAMs.

The information in this chapter is highly technical and is intended for experienced Apple users, programmers, and those with good reason to alter the default operating parameters of the 771Ø driver routines.

4.1 ACIA REGISTERS

Each peripheral slot of the Apple has been allocated 16 memory addresses for the purpose of memory-mapped input and output, and 256 addresses for PROM/ROM routines. The I/O addresses are located at $CØny, where n = 8 + slot number and y = $1-F. On the 771Ø the I/O addresses are occupied by the ACIA registers as follows:

```
$CØnØ = ACIA command register  (write operation)
$CØnØ = ACIA status register   (read operation)
$CØn1 = ACIA transmit register (write operation)
$CØn1 = ACIA receive register  (read operation)
```

These addresses are different under the Microsoft CP/M operating system, in accordance with the Z-8Ø to 65Ø2 address translation discussed in the Softcard Manual. Use $EØnØ and $EØn1 instead of the address above.

NOTE

> The last address digit is not decoded beyond even or odd. This means that data can be passed through any odd address within the range, while the ACIA's command/status registers can be accessed at any even address. For example, CØnØ, CØn2, CØn4, CØn6, and so on up to CØnF are all equivalent.

4.1.1 Acia Command Register

ACIA operation is controlled by one-byte commands written to the Command Register. The commands are defined in Table 4-1. Because the baud rate generator outputs a clock 16 times the baud rate, bits 1 and Ø should be set to Ø1, respectively.

To set the ACIA up to transmit and receive data, the ACIA must be initialized with a Master Reset. To do so, write a 1 into bits Ø and 1 (decimal 3) of the ACIA command register. Before data can be sent, the data format must be written to the command register. The pattern of of zeros and ones in bits 2, 3, and 4 determines the number of data bits, the number of stop bits, whether parity is used, and if so, whether it is even or odd. Bits 5, 6, and 7 similarly determine the state of the various handshaking lines, and enable or disable interrupts. Bits Ø and 1, in addition to being used to cause a Master Reset, control the clock frequency, which can be 1, 16, or 64 times the baud rate. The normal setting is 16 times the baud rate, but this value can be changed to effect a crude software control of baud rate. For instance, if the hardware DIP switches are set for a baud rate of 12ØØ and the normal value of 16 x baud rate is changed to 64 x baud rate, the effective baud rate is divided by 4, or 3ØØ baud.

4.1.2 Acia Status Register

The status register bits, when set (equal to 1), have the meaning given in Table 4.2. For instance, if bit Ø at address CØnØ is 1, a character has been received by the 771Ø.

TABLE 4.1. ACIA COMMANDS

7654 3210	
xxxx xx00	The clock is 1x the baud rate
xxxx xx01	The clock is 16x the baud rate
xxxx xx10	The clock is 64x the baud rate
xxxx xx11	ACIA Master Reset
xxx0 00xx	7 data + Even Parity + 2 stop bits
xxx0 01xx	7 data + Odd éarity + 2 stop bits
xxx0 10xx	7 data + Even Parity + 1 stop bit
xxx0 11xx	7 data + Odd Parity + 1 stop bit
xxx1 00xx	8 data + No Parity + 2 stop bits
xxx1 01xx	8 data + No Parity + 1 stop bit
xxx1 10xx	8 data + Even Parity + 1 stop bit
xxx1 11xx	8 data + Odd parity + 1 stop bit
x00x xxxx	Set RS232-C CTS Disable transmit interrupts
x01x xxxx	Set RS-242-C CTS Enable transmit interrupts
x10x xxxx	Clear RS-232-C CTS Disable transmit interrupts
x11x xxxx	Set RS-232-C CTS Transmit break on transmit data Disable transmit interrupts
0xxx xxxx	Disable Receive Interrupts
1xxx xxxx	Enable Receive Interrupts when: Receiver data register full Receive data overrun DTR signal inactive

TABLE 4.2. ACIA STATUS

Bit 0	Received data is ready for the computer.
Bit 1	The transmit register can accept data.
Bit 2	RS-232-C DTR inactive; don't send.
Bit 3	RS-232-C RTS inactive; don't send.
Bit 4	Received data improperly framed.
Bit 5	Data received before previous byte read.
Bit 6	Parity Error in received data.
Bit 7	ACIA-generated transmit or receive interrupt.

4.2 INPUT AND OUTPUT ROUTINES

Your computer looks at two page 0 locations to determine where the current keyboard handler and console output driver programs are located. The addresses are:

$36 - $37: console output handler
$38 - $39: keyboard input handler

When you type in the BASIC command IN#n, the firmware writes a $00 in location $38 and a $Cn in location $39; the equivalent monitor command, n^K, does the same thing. This creates an effective address of $Cn00 for the keyboard handler initialization program. The next time keyboard input is wanted, the initialization routine gets called. It must set everything up, then pass control to the input routine to actually do the input. One of the initializer's tasks is to change location $38 to identify the input handler's correct entry point. Then, the next time input is wanted, we can go straight to the input routine. Likewise, when PR#n is typed in, location $36 is set to $00 and $37 is set to $Cn. On the first output, control is passed to $Cn00 for output initialization. Location $36 should then be set to match the output handler's entry point for all subsequent console output.

If any peripheral control options are allowed in the program, invoking an IN#n or PR#n command will cause the default options to be selected for both input and output unless the options are initialized after the input-or-output initialization decision is made.

Input and output by your computer are handled on a byte-by-byte basis. As a result, data can be passed between the I/O routines and the computer in the Accumulator (A Register). Input data should be left in the Accumulator when control is returned to the caller. The output routine can find its data in the Accumulator.

The input and output routines should be called as subroutines. Control can be returned to the caller with a simple RTS (Return from Subroutine) instruction. All register contents should be saved on entry to subroutines, then restored to their original contents just before leaving (except for parameter-passing registers). This practice saves headaches and program-debugging time later on.

Several scratchpad memory locations are available. The video-display-refresh memory locations (addresses $400-$7FF) use only the first 120 of every 128 locations for the display data. The remaining 24 addresses can be used for other purposes. Two sets of available locations are $6Fn and $77n, where n = 8 + slot number. For most programs, these locations should be sufficient. But one other scratch location merits identification. Address $7F8 is often used to hold the page address of the currently-active peripheral. The page address is $Cn, where n is the slot number.

4.2.1 Interface Driver Program

You now have enough information to program a simple remote console interface driver. Your program should consist of three parts: initialization, input, and output.

a. For initialization, you must:

- Save the registers
- Reset ACIA
- Initialize ACIA with default command word
- Establish input and/or output entry points
- Allocate and/or initialize any special pointers, counters, etc
- Reset ACIA for appropriate I/O routine.
- Restore registers and return

b. For input, you must:

- Save the registers
- Check ACIA status and wait until input data is ready
- Read input data
- Do any special data conversion as needed-- e.g., lower-to-upper case
- Restore registers
- Return to caller

c. For output, you must:

- Save the registers
- Do any special preprint control (tabs, form feeds, etc.)
- Wait until ACIA can take more data
- Write data to ACIA
- Do any postprint control (line or page control, insert a line feed after a carriage return, etc.)
- Restore registers
- Return to caller

4.2.2 Driver Tasks

Several tasks are common to all of the routines. To stretch 256 bytes of space as far as possible, you should make as much code as possible common. Unless your driver is slot-dependent you can't predict what absolute addresses contain the common code, so you can't do subroutine calls, but must use relative code throughout. Unused status flags can be used to indicate the entry point. The standard driver uses the V (Overflow) flag to indicate initializing or not, and C (Carry) flag to indicate input or output. This allows you to use task-unique code according to the state of the relevant flag. After the flags have served their purposes, they can be reused to indicate a tab in progress or other function.

The program listing for the 7710 firmware in Appendix B provides examples of the handling of specific driver tasks. You are encouraged to use as much of this code as you need for your application.

4.3 LOADING YOUR DRIVER INTO RAM

If you have installed RAM, it must be loaded every time you turn your computer on and want to use the interface. The following procedure was devised for floppy disk systems; if you use some other storage medium, you'll need to devise your own scheme.

4.3.1 Saving the Driver On Disk

Most users who write their own firmware use an assembler, which can be used to assemble the source code directly into the RAM on the 7710. Once loaded into the RAM on the 7710, it can be saved on a disk with the following command:

```
BSAVE filename,A$Cn00,L$FF     (where n is the slot number)
```

4.3.2 Power-on Loading of the Driver

The driver routine can be loaded from disk into the RAM of the 7710 either directly (in immediate mode) by typing:

```
BLOAD filename,A$Cn00  (n= slot number of 7710)
```

or under program control as illustrated in the following program:

```
10 INPUT "ASYNC. INTERFACE SLOT IS: ",S
20 IF S<1 OR S>7 THEN GOTO 10
30 PRINT CHR$(4);"BLOAD DRIVER,A$C";S;"00"
40 REM REST OF PROGRAM GOES HERE
```

These statements can be incorporated into your HELLO program so the driver routine is BLOADed into the 7710's RAM every time you boot the disk.

CHAPTER 5

HARDWARE DESIGN

5.1 INTRODUCTION

The 7710 hardware can be divided conceptually into four sections:

a. Transmitter/Receiver
b. Baud rate generator
c. Control
d. Program memory

Each section is discussed separately in subsequent paragraphs.

5.2 TRANSMITTER/RECEIVER

The transmitter/receiver section's major component is the 6850 Asynchronous Communications Interface Adapter (ACIA). The ACIA performs the functions listed in (a) through (f).

a. Parallel to serial conversion
b. Serial to parallel conversion
c. When transmitting, adds start and stop bits
d. When receiving, removes start and stop bits
e. Makes status information available
f. Controls handshaking with peripherals

A programmable control register allows specification of word length, number of stop bits, parity type or inhibition, and clock division ratios. Additionally, the control register initiates transmitter/ receiver cycles and enables or disables interrupts. As presented previously in Table 4-2, a status register provides operation and status bits.

5.2.1 Data Carrier Detect

The ACIA provides three handshake lines on the serial side for peripherals/modems. The -Data Carrier Detect line (-DCD) permits the peripheral to control the receiver section and initiate an interrupt when the peripheral is not ready. This allows the peripheral to tell the computer to slow down. The -DCD input is tied to the inverted Data Terminal Ready (DTR) signal (pin 20) of the RS-232-C connector. Therefore, when DTR goes low, the ACIA stops transmission; when DTR goes high, transmission resumes.

5.2.2 Request to Send and Clear to Send

The other two ACIA handshake signals are -Request to Send (-RTS) and -Clear to Send (-CTS). These signals correspond to the primary RS-232-C handshake lines of the same names. However, the ACIA signals were named for the ACIA used as a DTE device; on the 7710 it is used as a DCE device. Thus ACIA output -RTS is inverted to control RS-232-C line CTS, which when high tells the peripheral that the ACIA is ready to transmit, while RS-232-C line RTS inverted controls ACIA input -CTS; RTS active indicates that the peripheral is ready to receive data from the ACIA.

5.2.3 Line Receiver/Line Driver

A 75154 and a 75150 perform the line receiver and line driver functions respectively, translating the standard Transistor-Transistor Logic (TTL) signals of the ACIA to the signal levels required by RS-232-C. They also provide for fail-safe operation should your interface cable short or disconnect accidentally.

5.3 BAUD RATE GENERATOR

The ACIA requires a clock for signal transmission and sampling. Different serial devices require data to be clocked at different rates. The 7710 employs a 4702 baud rate generator to supply standard baud rates from 50 to 19200. This chip contains an oscillator for the on-board quartz crystal and a controllable frequency divider circuit. The oscillator generates a highly stable 2.4576 MHz square wave signal, which is counted down by a divisor determined by the four switches on the card. The output of the 4702, which controls the ACIA Transmitter and Receiver Clock pins, is 16 times the actual selected baud rate. Thus the ACIA must be programmed to expect a x16 clock

5.3.1 Select Inputs

The select inputs to the 4702 provide 16 four-digit codes. By itself the 4702 generates 13 different baud rates from 50 to 9600, which account for 13 of the codes. One of the three remaining codes is a second code to select 2400 baud. Another is used to select 19,200 baud.

5.3.2 Prescaler

Normally, the 4702 prescales the 2.4576 MHz signal by dividing it first by 16 to yield a 9600 (x16) signal. The prescaler, however, allows connection to the signal after it has been divided by 8. Thus a 19,200 baud signal is available. The last code is used on the 7710 to select a signal from pin 24 of the RS-232-C connector, allowing the peripheral (or another source) to generate the baud rate.

5.4 CONTROL SECTION

The control section of the interface consists primarily of a buffer between the ACIA and the Apple data bus, ACIA control logic, and interrupt arbitration logic. Since the ACIA has only limited ability to drive the data lines of the computer, an 8304B bi-directional line buffer between the computer and the ACIA ensures successful data transfer. A power-down feature has been included to reduce power consumption by this device. When -DEVICE SELECT goes low, a transistor turns on power to the 8304B, allowing data transfer. When -DEVICE SELECT goes high, the transistor turns off power to the 8304B. R/-W determines the direction of data transfer through the 8304B.

5.4.1 ACIA Control Logic

The ACIA control logic is relatively straight-forward. When -DEV SEL is low, the ACIA is selected. Address bit A0 determines whether a data or command/status register is accessed, while R/-W determines whether a read (receiver or status) register or a write (transmitter or command) register is accessed.

5.4.2 Interrupt Arbitration

The interrupt arbitration logic is one link in the interrupt daisy chain, which prioritizes peripheral-generated interrupts to ensure that only one device interrupts at a time. If no higher-priority interrupt is in progress (pin 28, INT IN, high), a low from ACIA output -IRQ will force the INT OUT line low, telling the lower-priority devices that an interrupt is pending, forcing them to wait, and will force pin 30, -IRQ, low. After being serviced, the ACIA removes its interrupt request, and the arbitration logic allows INT OUT to go high again.

Remember that the device in the leftmost slot in the group 1 through 7 has the highest interrupt priority; slot 0 does not support the daisy chain. Also keep in mind that empty slots between cards break the daisy chain, and only cards to the left of the empty slot(s) will be part of the chain.

5.5 PROGRAM MEMORY

Your computer dedicates 256 bytes of memory space to each peripheral connector, addressable at $Cn00-CnFF where n is the slot number. The high byte of the address activates the -IO SEL line for the appropriate slot; the low byte selects the actual location on the on-board memory. The 7710 includes a driver loaded on two 256 x 4 PROMs. The driver is enabled when -IO SEL is low and R/-W is high. Because of the relatively high power consumption of the PROMs, a power-down circuit has been included to remove power from the PROMs when -IO SEL is inactive.

5.5.1 Modification of the Driver Software

Should you desire to develop your own software, you may substitute 2112 static RAMs for the PROMs. Using RAMs allows full development and testing of a program before it is committed to PROMs. If RAMs are installed, a jumper wire must be soldered between the pads labeled RAM to disable the power-down feature, which would otherwise destroy the RAM contents as fast as they could be written in. Pin 14 of the PROM/RAM is controlled by R/-W, providing the necessary RAM input as well as deselecting the PROM during write operations.

CHAPTER 6

USE OF THE 7710 WITH PASCAL AND CP/M

6.1 USE OF THE 7710 WITH THE MICROSOFT SOFTCARD

The 7710 is fully compatible with CP/M as implemented with the Microsoft Softcard. As stated in the Installation Chapter of the Softcard manual, the 7710 can be installed in slot 1, 2, or 3, although the 7710 has different functions in the different slots. The BIOS of the Softcard CP/M contains the necessary driver routines to access the 7710, and thus does not use the PROM routines supplies with the 7710 except to identify the presence of the 7710 in the Apple. This means that under Softcard CP/M both the 7710-01 and 7710-02 operate identically. For more details, refer to the Softcard Installation and Operation Manual.

Some other Z-80 cards on the market do use the PROMs on the 7710 for input and output as well as identification, and this can cause problems in the use of some applications such as Wordstar. CCS does not promise complete compatibility with Z-80 cards other than the Microsoft Softcard, although in most cases any problems you encounter are likely to be minor.

6.2 USE OF THE 7710 WITH APPLE PASCAL

The 7710 is fully compatible with the Apple Pascal Operating System. Refer to the Apple Pascal Manuals for information about which slots can be used and the necessary commands for accessing the 7710.

CHAPTER 7

USE OF THE 7710 IN THE APPLE IIe

7.1 THE 7710-01 IN THE APPLE IIE

The 7710-01 operates in the Apple IIe exactly as it does in the Apple II and Apple II+.

7.2 THE 7710-02 IN THE APPLE IIE

The 7710-02 can be used in the Apple IIe for communication with other computers, but only if one of the following four conditions is met:

a) Your 7710 was shipped from CCS after July, 1983. If so, its PROMs will have the labels 7710-03 L and 7710-03 H.

b) You are using one of several commercially available application programs such as Ascii Express Professional, Data Capture 4.0, or Softerm.

c) You install RAM chips in place of the PROMs and load a patched version of the 7710-02 driver into these RAMs (7710-03).

d) You have a copy of the old Apple II+ system master disk containing Integer Basic.

These four situations are discussed in more detail in the following paragraphs.

7.2.1 IIe COMPATIBLE VERSION OF 7710-02 (7710-03 PROMS)

If your PROMS are labelled 7710-03 L and 7710-03 H, then you have the IIe compatible PROMs. If you are still in doubt, use the Apple System monitor to examine the contents of memory location Cn65, where n is the number of the slot in which you have installed the 7710. This is the procedure for doing so:

```
From  Integer or Applesoft Basic type:
      CALL -151 (carriage return)
      *Cn65 (carriage return)
```

The monitor will then display the contents of location Cn65. If this location contains the value 1B, then your card will work in the IIe in terminal mode (40 columns only).

7.2.2 COMMERCIAL COMMUNICATIONS SOFTWARE

Many of the commercial communications programs work with the 7710 in the IIe and most will work with either the 7710-01 or the 7710-02. The following is a partial list of these programs:

Program	Manufacturer
Ascii Express Pro. Z-Term (CP/M) P-Term* (Pascal)	Southwestern Data Systems P.O. Box 582 Santee, CA 92071 (619) 562-3670
Crosstalk (CP/M)	Microstuff, Inc. 1900 Leland Drive, Suite 12 Marietta, GA 30067 (404) 952-0267
Data Capture 4.0	Link Systems 1640 19th Street Santa Monica, CA 90404 (504) 246-8438
Softerm	Softronics 6626 Prince Edward Memphis, TN 38119 (901) 755-5006
Transcend 1 Transcend 2 Transcend 3	SSM Microcomputer Products, Inc 2190 Paragon Drive San Jose, CA 95131 (408) 946-3616
Visiterm (7710-01)	Visicorp 2895 Zanker Road San Jose, CA 95134 (408) 946-9000

Unless otherwise noted, the above programs run under the Apple DOS operating system.

Microterm from Microcom of Massachusetts is not compatible with any version of the 7710 at this time (6/10/83).

Visiterm differs from the other program in that it actually requires the 7710-01 and does not work with the 7710-02. Visiterm recognizes the presence of the CCS 7710 card by looking at the first 10 bytes of the driver routines present in the various peripheral slots. If it finds a match to the 7710-01 firmware it sets up to talk to the 7710, but it completely bypasses the 7710 firmware and controls the interface itself. Visicorp is now aware that their program cannot recognize the card with 7710-02 PROMS installed, and may change this in future releases of Visiterm.

7.2.3 IIe PATCH FOR THE 7710-02 FIRMWARE

The reason for the 7710-02 firmware incompatibility with the IIe is known and there is a patch to alleviate the problem. The 7710-02 driver makes a call to a location in the Apple monitor that has been changed in the new IIe system monitor. Changing a single byte of the 7710-02 driver routine fixes the problem, however this patch still does not allow use of the 80 column mode. Eighty column capability is a feature of several of the communications programs mentioned above.

7.2.3.1 Forty Column Patched Version

Those who wish to use the patched version can either purchase a new PROM set from Customer Service at CCS ($30) or purchase RAMS and replace the PROMS on their 7710s with these RAMS, into which the patched driver can be loaded from disk. The 7710-02 driver can be moved from the peripheral card memory space into a lower part of the Apple's memory, altered, and then BSAVEd to disk. Each time the user powers up the Apple, this program must be BLOADed into the card's RAM. The RAMs used are two 2112 chips, which can be bought retail for about $3.00.

The procedure for altering the driver and saving it on disk for use with 2112 RAMs is as follows:

a) Boot the Apple with a DOS 3.3 diskette.
b) From Applesoft BASIC type:

```
BSAVE DRIVER,A$Cn00,L$FF (carriage return)
```

where n is the number of the slot in which you have placed the 7710. This saves a copy of the 7710-02 driver onto your diskette.

c) Type:

```
BLOAD DRIVER,A$6000 (carriage return)
```

Thisloads a copy of the driver into memory starting at location hex 6000 (decimal 24576).

d) Type:

```
POKE 24677,27 (carriage return)
```

This makes the necessary patch.

e) Type:

```
BSAVE DRIVER,A$6000,L$FF (carriage return)
```

This saves the patched driver to disk.

f) The driver can now be loaded into RAM on the 7710 by the following statement:

```
BLOAD DRIVER,A$Cn00  (where n is the slot number)
```

This statement can be incorporated into your HELLO program so that every time you reboot, the driver is automatically loaded into the RAM on the 7710. As long as your hello program executes the following two statements, the driver will be loaded:

```
D$=CHR$(4)
PRINT D$;"BLOAD DRIVER,A$Cn00"
```

Remember, this patched driver does not work in 80 column mode.

7.2.4 Using the Terminal Firmware with the Apple II+ monitor.

Since the cause of the 7710-02 firmware incompatibility with the Apple IIe is its interaction with the new Apple IIe monitor, we can correct the incompatibility by loading a copy of the old Apple II+ monitor into the language card area of the IIe. To do so, you must obtain a copy of the Apple II+ system master. By loading Integer BASIC from this diskette, you can have a copy of the old monitor in memory. To activate the old monitor, type:

INT (carriage return)

You can then enter terminal mode as follows

```
IN#n     (n=slot number of 7710)
^A^F     (or ^A^H for half duplex)
```

APPENDIX A

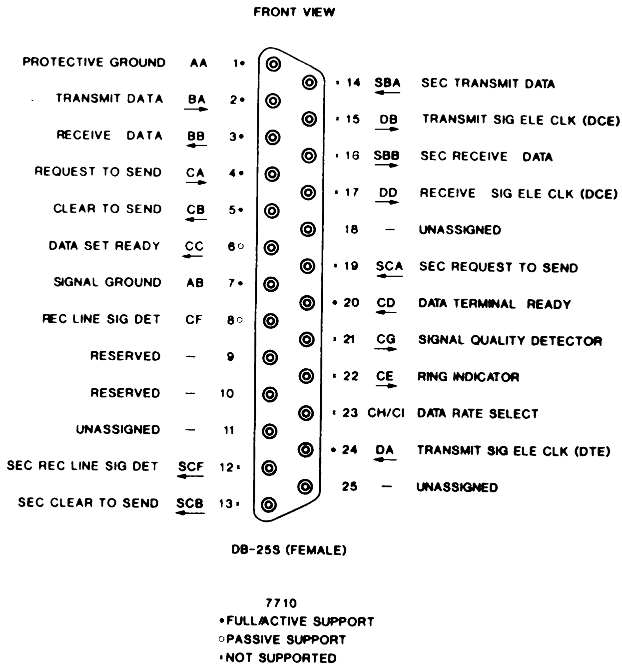
TECHNICAL INFORMATION

A.1 USER-REPLACEABLE PARTS

QTY	REF	CCS P/N	DESCRIPTION
5	C1-4,7	42034-21046	CAPACITOR, MONOLITHIC .luf, 50vdc
2	C5,6	42215-55605	CAPACITOR, MICA 56pf, 500vdc, 10%
1	J2	56004-02013	HEADER, DUAL 13 PIN, STRAIGHT ENTRY
2	Q1,2	36100-02907	TRANSISTOR, SI; PN2907 GENERAL PURPOSE
1	R1	40002-01005	RESISTOR, FIXED, COMP 10 ohm, 1/4W, 10%
1	R2	40002-06815	RESISTOR, FIXED, COMP 680 ohm, 1/4W, 10%
1	R3	40003-01515	RESISTOR, FIXED, COMP 100 ohm, 1/2W, 10%
4	R4-7	40002-02215	RESISTOR, FIXED, COMP 220 ohm, 1/4W, 10%
1	R8	40002-02725	RESISTOR, FIXED, COMP 2.7K, 1/4W, 10%
1	R9	40002-01055	RESISTOR, FIXED, COMP 1M, 1/4W, 10%
1	U1	30300-00150	IC, INTERFACE; 75150 DUAL RS-232-C DRIVER
1	U2	31100-06850	IC, DIGITAL, MOS; 6850 ACIA
1	U3	30300-00154	IC, INTERFACE; 75154 QUAD RS-232 RCVR
1	U4	31000-04702	IC, DIGITAL, CMOS; 4702 BAUD RATE GENERATOR
1	U7	30900-08304	IC, DIGITAL, TTL; 8304B OCTAL BUS DRVR/RCVR
1	U8	30000-00009	IC, DIGITAL, TTL; 74LS09 QUAD 2-IN AND, OC
1	U9	30000-00136	IC, DIGITAL,TTL;74LS136 QUAD 2-IN EX-OR, OC
1	U10	30000-00003	IC, DIGITAL, TTL; 74LS03 QUAD 2-IN NAND, OC
1	Y1	48132-45762	XTAL, QUARTZ 2.4576MHz, HC-6
1	XU1	58102-00080	SOCKET, IC; LOW PROFILE 8 PIN DIP
3	XU8-10	58102-00140	SOCKET, IC; LOW PROFILE 14 PIN DIP
2	XU3-6	58102-00160	SOCKET, IC; LOW PROFILE 16 PIN DIP
1	XU7	58102-00200	SOCKET, IC; LOW PROFILE 20 PIN DIP

1	XU2	58102-00240	SOCKET, IC; LOW PROFILE 24 PIN DIP
1		50026-01051	CABLE, 9" I/O, RS-232-C CONNECTOR
1	S1	27111-41010	SWITCH, DIP, 4PST

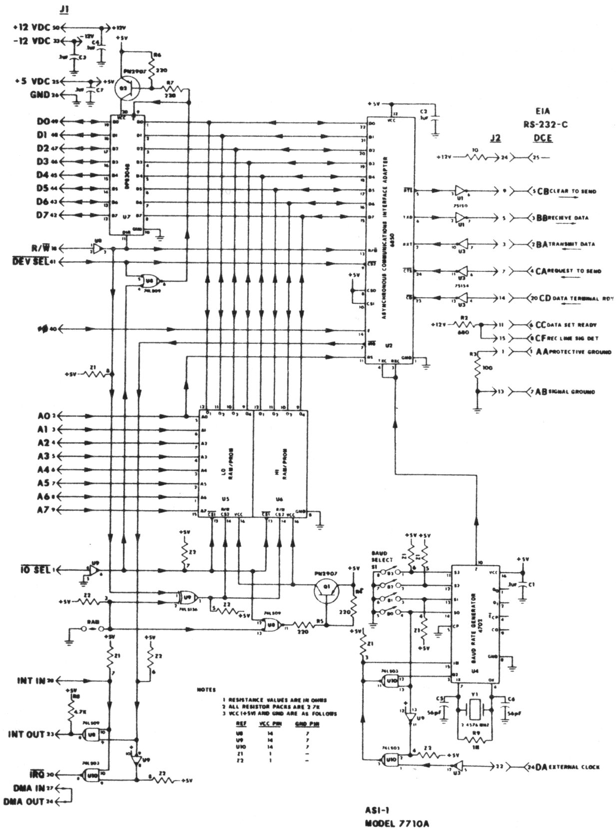
A.2 RS-232-C CONNECTOR PINOUTS



EIA RS-232C
DCE TYPE CONNECTOR PIN ASSIGNMENT

Arrows pointing away from the connector indicate outgoing signals and arrows pointing toward the connector indicate signals going to the 771Ø.

A.3 SCHEMATIC/LOGIC DIAGRAM



ASI-1
MODEL 7710A
ASYNCHRONOUS SERIAL INTERFACE

Power Consumption: 89 milliamps when not being used and 231 when both PR# and IN# are active.

A.4 DEFINITION OF RS-232-C CONFIGURATIONS

A	Transmit Only
B	Transmit Only*
C	Receive Only
D	Half Duplex; or Duplex*
E	Full Duplex
F	Primary Channel Transmit Only* /Secondary Channel Receive Only
G	Primary Channel Receive Only / Secondary Channel Transmit Only*
H	Primary Channel Transmit Only / Secondary Channel Receive Only
I	Primary Channel Receive Only / Secondary Channel Transmit Only
J	Primary Channel Transmit Only* / Half Duplex Secondary Channel
K	Primary Channel Receive Only / Half Duplex Secondary Channel
L	Half Duplex Primary Channel / Half Duplex Secondary Channel; or
L	Duplex Primary Channel* / Duplex Secondary Channel*
M	Duplex Primary Channel / Duplex Secondary Channel
Z	Special (Circuits specified by supplier)

*Note the inclusion of Request to Send in a Transmit Function, where it would not ordinarily be expected, but could indicate a non-transmit mode to the data communications equipment (DCE) to permit it to remove a line signal or to send synchronizing or framing signals as required.

A.5 RS-232-C STANDARD CONFIGURATIONS

	Interchange Circuit	A	B	C	D	E	F	G	H	I	J	K	L	M	Z
AA	Protective Ground	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AB	Signal Ground / Common Return	X	X	X	X	X	X	X	X	X	X	X	X	X	X
BA	Transmitted Data	X	X		X	X	X		X		X		X	X	O
BB	Received Data			X	X	X		X		X		X	X	X	O
CA	Request to Send		X		X		X				X		X		O
CB	Clear to Send	X	X		X	X	X		X		X		X	X	O
CC	Data Set Ready	X	X	X	X	X	X	X	X	X	X	X	X	X	O
CD	Data Terminal Ready	S	S	S	S	S	S	S	S	S	S	S	S	S	O
CE	Ring Indicator	s	s	s	s	s	s	s	s	s	s	s	s	s	o
CF	Received Line Signal Detector			X	X	X		X		X		X	X	X	O
CG	Signal Quality Detector														o
CH/CI	Data Signalling Rate Selector (DTE/DCE)														o
DA/DB	Transmit Signal Element Timing (DTE/DCE)	T	T		T	T	T		T		T	T	T	T	O
DD	Receiver Signal Element Timing (DCE)			t	t	t		t		t		t	t	t	o
SBA	Secondary Transmitted Data							x		x	x	x	x	x	o
SBB	Secondary Received Data						x		x		x	x	x	x	o
SCA	Secondary Request to Send							x			x	x	x		o
SCB	Secondary Clear to Send							x		x	x	x	x	x	o
SCF	Secondary Received Line Signal Detector						x		x		x	x	x	x	o

Notes: Upper case indicates a line supported by the CCS 7710.
Lower case indicates a line not supported by the CCS 7710.
X = Basic interchange circuits, all systems
T = Additional interchange circuits required for synchronous channel
S = Additional interchange circuits required for switched service
O = Specified by supplier as required
- = Optional; supported by the CCS 7710

APPENDIX B

FIRMWARE LISTINGS

```
0000:          *
0000:          *
0000:          *
0000:          * DRIVER PROGRAM FOR THE
0000:          * CCS 7710-01 ASYNCHRONOUS SERIAL INTERFACE CARD
0000:          *
0000:          * COPYRIGHT 1979 CALIFORNIA COMPUTER SYSTEMS
0000:          *
0000:          *THIS PROGRAM CONTAINS THE NECESSARY CODE TO ALLOW DIRECT
0000:          *LOGICAL REPLACEMENT OF YOUR COMPUTER'S KEYBOARD AND TV
0000:          *OUTPUT. OUTPUT DEFAULTS TO 255 CHARACTERS PER LINE AND
0000:          *NO PAGING. THREE ENTRY POINTS ARE DEFINED: INITIALIZATION,
0000:          *INPUT, AND OUTPUT. WHEN THE BASIC COMMANDS IN#N OR PR#N
0000:          *ARE ISSUED, THE COMPUTER SETS UP A JUMP VECTOR ADDRESS TO
0000:          *THE INITIALIZATION ENTRY POINT. THE INITIALIZATION ROUTINE
0000:          *WILL ADJUST THE INPUT OR OUTPUT ENTRY POINT VECTOR TO THE
0000:          *CORRECT ADDRESS WHEN THE FIRST INPUT OR OUTPUT OCCURS.
0000:          *
0000:          * SYSTEM EQUATES
----- NEXT OBJECT FILE NAME IS CCS7610C SOURCE.OBJ0
C200:                  ORG  $C200
00FF:          MAXCHR  EQU  $FF        255 CHARACTERS PER LINE DEFAULT
0087:          BKSP    EQU  $87        ASCII BACKSPACE - 1 (FOR CARRY)
008A:          LNFD    EQU  $8A        ASCII LINE FEED
008C:          FF      EQU  $8C        ASCII FORM FEED
008D:          CARRET  EQU  $8D        ASCII CARRIAGE RETURN
0095:          FS      EQU  $95        FORWARD SPACE
00A0:          SPACE   EQU  $A0
0024:          CH      EQU  $24        TAB COLUMN POINTER
0036:          CSWL    EQU  $36        LOCATION OF OUTPUT DRIVER VECTOR
0038:          KSWL    EQU  $38        LOCATION OF INPUT DRIVER VECTOR
0039:          KSWH    EQU  $39
0538:          CPL     EQU  $5F8-$C0   MAXCHR SAVE LOCATION
0638:          LOCASE  EQU  $6F8-$C0   LC CONVERSION MASK HOLD
0738:          LNCNT   EQU  $7F8-$C0   LINE COUNTER
06B8:          CHCNT   EQU  $778-$C0   CHARACTER COUNTER
C080:          CMD     EQU  $C080      + $N0 FOR THE ACIA COMMAND PORT
```

```
C080:                STATUS  EQU  CMD        ACIA STATUS PORT
C081:                DATA    EQU  CMD+1      ACIA DATA PORT
FCA8:                WAIT    EQU  $FCA8      ADDRESS OF WAIT ROUTINE
FFCB:                RETURN  EQU  $FFCB      USED TO FIND SLOT ADDRESS
C200:                *
C200:                * THE COMMON CODE
C200:                *
C200:2C CB FF        INIT    BIT  RETURN     SET V = 1
C203:70 04                   BVS  COM
C205:18              OUTEP   CLC             CLEAR THE CARRY FOR OUTPUT
C206:B0                      DFB  $B0        MAKE A BCS TO SKIP NEXT INSTRUCTION
C207:38              INEP    SEC             SET CARRY FOR INPUT
C208:B8                      CLV             CLEAR V FOR I/O
C209:48              COM     PHA             SAVE REGISTERS AND STATUS
C20A:8A                      TXA
C20B:48                      PHA
C20C:98                      TYA
C20D:48                      PHA
C20E:08                      PHP
C20F:78                      SEI             DISABLE INTERRUPTS
C210:20 CB FF                JSR  RETURN     PUT SLOT ADDRESS ON THE STACK
C213:BA                      TSX             PUT THE SLOT PAGE
C214:BC 00 01                LDY  $100,X     NUMBER INTO Y
C217:68                      PLA             RECOVER THE OUTPUT DATA (IF ANY)
C218:68                      PLA
C219:68                      PLA
C21A:68                      PLA
C21B:9A                      TXS             RESTORE STACK POINTER
C21C:48                      PHA             SAVE THE DATA ON STACK TOP
C21D:98                      TYA             GET THE SLOT PAGE NUMBER
C21E:AA                      TAX             ESTABLISH X INDEX
C21F:0A                      ASL  A          MULTIPLY BY 16 TO GET THE
C220:0A                      ASL  A          $N0 INDEX FOR ACIA ACCESS
C221:0A                      ASL  A
C222:0A                      ASL  A
C223:A8                      TAY             ESTABLISH Y INDEX
C224:68              COMA    PLA             GET THE SAVED STATUS CODES
C225:28                      PLP
C226:48                      PHA
C227:50 1B                   BVC  IO         ROUTINE I/O BRANCH
C229:                *
C229:                * INITIALIZATION ROUTINE
C229:                *
C229:                *THIS CODE HANDLES THE FIRST INPUT OR THE FIRST OUTPUT REQUEST
C229:                *AFTER INVOKING THE IN#N OR THE PR#N COMMANDS. IT INITIALIZES
C229:                *THE ACIA, SETS UP THE LOWER CASE CONVERSION MASK, THE DESIRED
C229:                *ENTRY POINT VECTOR, AND FINALLY GOES TO THE APPROPRIATE INPUT
C229:                *OR OUTPUT ROUTINE.
C229:                *
C229:A9 23                   LDA  #$23       RESET THE ACIA
C22B:99 80 C0                STA  CMD,Y
C22E:A9 11                   LDA  #$11       SET FOR 8+2, NO PARITY, NO INT
C230:99 80 C0                STA  CMD,Y
C233:B8                      CLV             CLEAR THE INIT FLAG
C234:A5 38                   LDA  KSWL       SEE IF INPUT WANTED
```

```
C236:DØ 29             BNE  OINIT      NO, BRANCH FOR OUTPUT INIT
C238:E4 39             CPX  KSWH       MAYBE, NAKE SURE
C23A:DØ 25             BNE  OINIT      NO, BRANCH FOR OUTPUT
C23C:9D 38 Ø6          STA  LOCASE,X   SET NO LC CONV MASK
C23F:A9 Ø7             LDA  #7         SET NORMAL INPUT ENTRY POINT
C241:85 38             STA  KSWL       VECTOR
C243:38                SEC             FALL THROUGH NEXT BRANCH
C244:9Ø 29       IO    BCC  OUT        GO TO NORMAL OUTPUT
C246:            *
C246:            * INPUT ROUTINE
C246:            *
C246:            *THIS ROUTINE EXPECTS NO PARAMETERS FROM CALLING ROUTINES.
C246:            *IT WAITS UNTIL DATA HAS BEEN TYPED IN FROM THE KEYBOARD, AND
C246:            *THEN RETURNS THAT DATA IN THE ACCUMULATOR TO THE CALLER. IT
C246:            *IGNORES ALL LINE FEEDS AND CONVERTS ALL LOWER CASE LETTERS TO
C246:            *UPPER CASE (IF THE LOCASE MASK = #$2Ø.)
C246:            *
C246:B9 8Ø CØ    INPUT LDA  STATUS,Y   GET ACIA STATUS
C249:4A                LSR  A          ISOLATE RXRDY BIT
C24A:9Ø FA             BCC  INPUT      LOOP UNTIL DATA IS READY
C24C:68                PLA             GET RID OF DATA ON STACK TOP
C24D:B9 81 CØ          LDA  DATA,Y     READ THE NEW DATA
C25Ø:Ø9 8Ø             ORA  #$8Ø       SET BIT 7 FOR NORMAL VIDEO
C252:48                PHA             SAVE DATA ON STACK
C253:C9 8A             CMP  #LNFD      IGNORE LINE FEEDS
C255:FØ EF             BEQ  INPUT
C257:68                PLA
C258:C9 EØ             CMP  #$EØ       SEE IF IT IS LOWER CASE
C25A:9Ø 4E             BCC  DONA       GO FINISH UP IF NOT
C25C:5D 38 Ø6          EOR  LOCASE,X   CONVERT TO LOWER CASE IF MASK = #$2Ø
C25F:BØ 49             BCS  DONA       NOW, GO FINISH UP
C261:            *
C261:            * OUTPUT INITIALIZATION
C261:            *
C261:A9 Ø5       OINIT LDA  #5         SET NORMAL OUTPUT ENTRY POINT VECTOR
C263:85 36             STA  CSWL
C265:A9 FF             LDA  #MAXCHR    SET DEFAULTS
C267:9D 38 Ø5          STA  CPL,X
C26A:A9 ØØ             LDA  #Ø
C26C:9D B8 Ø6          STA  CHCNT,X    ZERO CHARACTER COUNTER
C26F:            *
C26F:            * OUTPUT ROUTINE
C26F:            *
C26F:            *THIS ROUTINE DOES THE ACTUAL OUTPUT OF THE DATA. IT EXPECTS
C26F:            *TO FIND THE DATA FOR OUTPUT ON THE TOP OF THE STACK (WHERE
C26F:            *THE COMMON CODE PUT IT.)
C26F:            *
C26F:BD B8 Ø6    OUT   LDA  CHCNT,X    SEE IF TAB WANTED
C272:C5 24             CMP  CH
C274:BØ Ø3             BCS  OUTA       BRANCH IF NO TAB
C276:A9 AØ             LDA  #SPACE     SPACE OUT TO COLUMN
C278:48                PHA             SAVE ON STACK
C279:68          OUTA  PLA             RETRIEVE CHARACTER
C27A:48          OUTD  PHA             RESAVE IT
C27B:Ø8                PHP             SAVE TAB STATUS
```

```
C27C:C9 95         CRTLU   CMP  #FS        SEE IF CONTROL U
C27E:F0 04                 BEQ  CHCTI      BRANCH IF SO
C280:29 60                 AND  #$60       TEST FOR OTHER CNTRL CHARACTER
C282:F0 03                 BEQ  OUTB       SKIP COUNTER INCREMENT
C284:FE B8 06      CHCTI   INC  CHCNT,X    BUMP COUNTER
C287:28            OUTB    PLP             REGET TAB FLAG
C288:B9 80 C0      OUTC    LDA  STATUS,Y   GET ACIA STATUS
C28B:29 03                 AND  #3         ISOLATE TX AND RX BUFFER BITS
C28D:F0 F9                 BEQ  OUTC       WAIT IF NOT READY
C28F:29 01                 AND  #1         CHECK FOR INPUT CHARACTER
C291:D0 46                 BNE  INCHR      GET CHARACTER IF THERE IS ONE
C293:68                    PLA             GET DATA FOR OUTPUT
C294:99 81 C0              STA  DATA,Y     OUTPUT IT
C297:90 D6                 BCC  OUT        BRANCH IF TAB
C299:70 01                 BVS  LF         BRANCH IF SELF GEN CHAR
C29B:48                    PHA             RESAVE CHARACTER
C29C:C9 8A         LF      CMP  #LNFD      SEE IF LINE FEED
C29E:F0 09                 BEQ  DONE       YES, BRANCH
C2A0:E9 87         BS      SBC  #BKSP      SEE IF BACK SPACE
C2A2:D0 13                 BNE  CR         NO, BRANCH
C2A4:DE B8 06              DEC  CHCNT,X    YES, ADJUST COUNTER
C2A7:30 12                 BMI  CRA        DISALLOW NEGATIVE COUNT
C2A9:              *
C2A9:              * FINAL COMMON CODE
C2A9:              *
C2A9:              *THIS PART OF THE CODE RESTORES THE REGISTERS AND RETURNS TO
C2A9:              *THE CALLER. IT IS USED BY ALL OF THESE ROUTINES.
C2A9:              *
C2A9:68            DONE    PLA             SET THE STACK STRAIGHT
C2AA:BA            DONA    TSX             MODIFY A REGISTER VALUE IN
C2AB:E8                    INX             STACK TO INSURE IT IS
C2AC:E8                    INX             RESTORED TO THE RIGHT VALUE
C2AD:E8                    INX
C2AE:9D 00 01              STA  $100,X
C2B1:68                    PLA             RESTORE REGISTERS
C2B2:A8                    TAY
C2B3:68                    PLA
C2B4:AA                    TAX
C2B5:68                    PLA
C2B6:60                    RTS             DONE!
C2B7:              *
C2B7:              * REST OF OUTPUT CODE
C2B7:              *
C2B7:E9 06         CR      SBC  #6         SEE IF A CARRIAGE RETURN
C2B9:D0 12                 BNE  AUTOCR     NO, BRANCH
C2BB:9D B8 06      CRA     STA  CHCNT,X    ZERO OUT COUNTER
C2BE:85 24                 STA  CH         AND TAB POINTER
C2C0:A9 C0                 LDA  #$C0       WAIT FOR PRINT HEAD TO RETURN
C2C2:20 A8 FC              JSR  WAIT
C2C5:A9 8A         MAKELF  LDA  #LNFD      MAKE A LINE FEED
C2C7:2C CB FF      SELF    BIT  RETURN     V=1 FOR SELF GENERATED CHAR;
C2CA:38                    SEC             C=1 FOR NO TAB
C2CB:B0 AD                 BCS  OUTD       ALWAYS BRANCH
C2CD:BD B8 06      AUTOCR  LDA  CHCNT,X    END OF LINE YET?
C2D0:DD 38 05              CMP  CPL,X
```

```
C2D3:9Ø D4           BCC  DONE       NO, DONE
C2D5:A9 8D           LDA  #CARRET    YES, GET A CARRIAGE RETURN
C2D7:DØ EE           BNE  SELF       PROCESS IT
C2D9:          *
C2D9:          * CONTROL S INTERRUPT CODE
C2D9:          *
C2D9:          *THIS ROUTINE PROCESSES THE CONTROL S (DC3) TO INTERRUPT
C2D9:          *SENDING. ANY CHARACTER WILL CAUSE SENDING TO RESUME.
C2D9:          *
C2D9:Ø8        INCHR PHP             SAVE PSW ON STACK
C2DA:B9 81 CØ        LDA  DATA,Y     GET INPUT CHARACTER
C2DD:29 7F           AND  #$7F       STRIP MSB
C2DF:C9 13           CMP  #$13       CHECK FOR CONTROL S
C2E1:FØ Ø3           BEQ  WAIT1      IF SO, WAIT FOR NEXT CHARACTER
C2E3:28              PLP             IF NOT, CLEAR STACK AND
C2E4:          * GO SEND CHARACTER
C2E4:DØ A2           BNE  OUTC
C2E6:B9 8Ø CØ  WAIT1 LDA  STATUS,Y   WAIT FOR INPUT CHARACTER
C2E9:29 Ø1           AND  #1         GET INPUT BUFFER STATUS
C2EB:FØ F9           BEQ  WAIT1
C2ED:B9 81 CØ        LDA  DATA,Y     READ DATA TO CLEAR FLAG
C2FØ:28              PLP             GET BACK OLD PSW
C2F1:DØ 95           BNE  OUTC       GO SEND NEXT CHARACTER
```

```
ØØØØ:          ****************************************************************
ØØØØ:          *                                                              *
ØØØØ:          * 771Ø-Ø2 FIRMWARE DRIVER FOR THE CCS 771Ø ASYNCHRONOUS        *
ØØØØ:          *                  SERIAL INTERFACE                            *
ØØØØ:          *                                                              *
ØØØØ:          *              WRITTEN BY FRED VILLES
ØØØØ:          *
ØØØØ:          *
ØØØØ:          *
ØØØØ:          *
ØØØØ:          *
ØØØØ:          * ACIA PORT ADDRESSES
CØ8Ø:          STATUS EQU $CØ8Ø
CØ8Ø:          CONTRL EQU $CØ8Ø
CØ81:          DATA   EQU $CØ81
ØØØØ:          *  APPLE EQUATES
ØØ4E:          RNDL   EQU $4E        RANDOM # SEED
ØØ4F:          RNDH   EQU $4F
ØØ28:          BASL   EQU $28        SCREEN LINE POINTER
ØØ24:          CH     EQU $24        SCREEN COLUMN POINTER
Ø1ØØ:          STACK  EQU $1ØØ
FF58:          IORTS  EQU $FF58      ADDRESS OF RTS INSTRUCTION
FE89:          SETKBD EQU $FE89      ENTRY TO DO IN#Ø
FD1B:          KEYIN  EQU $FD1B      ENTRY TO READ APPLE KEYBOARD
FDFØ:          COUT1  EQU $FDFØ      ENTRY SENDS CHAR TO SCREEN
FDED:          COUT   EQU $FDED      CHAR TO SCREEN AND CARD
FD26:          GETKEY EQU $FD26      ENTRY GETS KEY AND RESETS FLASH
CØØØ:          KEY    EQU $CØØØ      APPLE KEYBOARD PORT
```

```
ØØØØ:                  *
Ø7F8:                  CN       EQU  $7F8         HOLD $CN WHERE N=SLOT #
Ø6F8:                  PRNT     EQU  $6F8         B7=Ø IF PRINT IS TRUE
Ø578:                  ASAVE    EQU  $578         SAVE A REGISTER
Ø638:                  CASE     EQU  $638         UC/LC CONVERSION BYTE
Ø738:                  RAMCTL   EQU  $738         HOLD CURRENT ACIA CONTROL
ØØØØ:                  *
----- NEXT OBJECT FILE NAME IS CCS761ØD SOURCE.OBJØ
C2ØØ:                           ORG  $C2ØØ
ØØØØ:                           OBJ  $C2ØØ
C2ØØ:                  *
C2ØØ:2C 58 FF          HINIT    BIT  IORTS        SETS V FLAG
C2Ø3:                  * IORTS IS THE ADDRESS OF A RETURN INSTRUCTION
C2Ø3:7Ø Ø4                      BVS  COMMON       SKIP SOFT ENTRIES
C2Ø5:                  *
C2Ø5:18                PRNTRY   CLC
C2Ø6:BØ FE                      BCS  *
C2Ø7:                           ORG  *-1
ØØØØ:                           OBJ  *
C2Ø7:                  *
C2Ø7:38                INNTRY   SEC               INPUT ENTRY
C2Ø8:B8                         CLV               FLAG SOFT ENTRY
C2Ø9:                  *                          COMMON ENTRY CODE
C2Ø9:8D 78 Ø5          COMMON   STA  ASAVE        SAVE A REGISTER
C2ØC:98                         TYA
C2ØD:48                         PHA               SAVE Y REGISTER
C2ØE:8A                         TXA
C2ØF:48                         PHA               SAVE X REGISTER
C21Ø:Ø8                         PHP               SAVE STATUS
C211:78                         SEI               DISABLE INTERRUPTS
C212:2Ø 58 FF                   JSR  IORTS        WHILE PLAYING WITH
C215:BA                         TSX               THE STACK TO GET
C216:BC ØØ Ø1                   LDY  STACK,X      $CN
C219:8C F8 Ø7                   STY  CN
C21C:98                         TYA
C21D:ØA                         ASL  A
C21E:ØA                         ASL  A
C21F:ØA                         ASL  A
C22Ø:ØA                         ASL  A
C221:4E F8 Ø6                   LSR  PRNT         SET PRINT FLAG TRUE
C224:28                         PLP               ENABLE INTERRUPTS ASAP
C225:AA                         TAX               $NØ->X
C226:5Ø 21                      BVC  SOFT         SKIP INITS IF SOFT ENTRY
C228:                  *
C228:A9 2Ø                      LDA  #$2Ø
C22A:99 38 Ø6                   STA  CASE,Y       CASE=$2Ø FOR LC->UC
C22D:A9 A3                      LDA  #$A3
C22F:9D 8Ø CØ                   STA  CONTRL,X     ACIA MASTER RESET
C232:4A                         LSR  A            SET FULL HANDSHAKE
C233:99 38 Ø7                   STA  RAMCTL,Y     WITH 8 DATA, 2 STOP BITS
C236:A5 36                      LDA  $36          TEST FOR PR#N OR IN#N
C238:DØ ØB                      BNE  SETIN        HARD ENTRY
C23A:C4 37                      CPY  $37
C23C:DØ Ø7                      BNE  SETIN
C23E:A9 Ø5                      LDA  #$5          PR#N--SET SOFT ENTRY
```

```
C240:85 36            STA  $36          ADDRESS IN VECTOR
C242:18               CLC
C243:90 04            BCC  SOFT
C245:A9 07    SETIN   LDA  #$07         IN#N--SET SOFT ENTRY
C247:85 38            STA  $38          ADDRESS IN VECTOR
C249:         *
C249:         *
C249:B8       SOFT    CLV
C24A:         *
C24A:AC F8 07 RSTCTL  LDY  CN           LOAD $C+SLOT NUMBER
C24D:B9 38 07         LDA  RAMCTL,Y     LOADS FROM $738
C250:29 BF            AND  #$BF         SET RTS SIGNAL
C252:9D 80 C0 SETCTL  STA  CONTRL,X     CONTRL=$C080
C255:90 7D            BCC  OUT2         BRANCH IF PR# ACTIVE?
C257:A4 24    IN      LDY  CH           CH=$24, SCREEN COLUMN POINTER
C259:B1 28            LDA  (BASL),Y     BASL=$28 SCREEN LINE POINTER
C25B:48               PHA               SAVE SCREEN CHARACTER
C25C:AD 00 C0 IN1     LDA  KEY          CHECK KEYBOARD
C25F:0A               ASL  A            SHIFT BIT 7 INTO CARRY FLAG
C260:68               PLA               GET SCREEN CHAR
C261:90 0F            BCC  IN3          BRANCH IF NO KEY
C263:08               PHP
C264:20 26 FD         JSR  GETKEY       RESET SCREEN AND GET KEY
```

FOR THE 7710-03 DRIVER, THE PREVIOUS INSTRUCTION IS JSR KEYIN, MAKING LOCATION C265 EQUAL TO $1B. ALL OTHER LOCATIONS SAME IN THE TWO DRIVERS

```
C267:28               PLP
C268:C9 81            CMP  #$81         CONTROL-A
C26A:F0 47            BEQ  COMAND       YES--GO PROCESS COMMAND
C26C:70 63            BVS  OUT1         SEND CHAR IF TERM MODE
C26E:A4 24    IN2     LDY  CH
C270:B1 28            LDA  (BASL),Y     SAVE SCREEN CHAR
C272:48       IN3     PHA               AND SET SCREEN
C273:29 3F            AND  #$3F         TO FLASHING
C275:09 40            ORA  #$40
C277:91 28            STA  (BASL),Y
C279:         *
C279:E6 4E    RBUMP   INC  RNDL
C27B:D0 02            BNE  CHKINP
C27D:E6 4F            INC  RNDH
C27F:BD 80 C0 CHKINP  LDA  STATUS,X     CHECK ACIA INPUT
C282:4A               LSR  A            STATUS FOR CHAR
C283:90 D7            BCC  IN1          NO, GO LOOP
C285:68               PLA               YES, RESET SCREEN
C286:91 28            STA  (BASL),Y
C288:AC F8 07         LDY  CN           UPDATE THE ACIA
C28B:B9 38 07         LDA  RAMCTL,Y     CONTROL ON EXIT
C28E:9D 80 C0         STA  CONTRL,X     TO TURN RTS OFF
C291:BD 81 C0         LDA  DATA,X       GET ACIA CHAR
C294:09 80            ORA  #$80         SET HIGH BIT
C296:C9 92            CMP  #$92         CONTROL-R?
C298:F0 AF            BEQ  SOFT
C29A:C9 94            CMP  #$94         CONTROL-T?
C29C:F0 18            BEQ  SETTRM
```

```
C29E:C9 8A                 CMP  #$8A      LF?
C2A0:F0 A8                 BEQ  RSTCTL    YES--IGNORE
C2A2:50 4C                 BVC  RETRN1    RETURN CHAR IF NOT TERM
C2A4:C9 E0                 CMP  #$E0      LC?
C2A6:90 03                 BCC  NOTLC     NO
C2A8:59 38 06              EOR  CASE,Y    YES--CONVERT
C2AB:08           NOTLC    PHP
C2AC:20 ED FD              JSR  COUT      PRINT CHAR
C2AF:28                    PLP
C2B0:38           RCLNK    SEC
C2B1:70 97                 BVS  RSTCTL
C2B3:             *
C2B3:             *
C2B3:20 1B FD     COMAND   JSR  KEYIN     GET CHAR
C2B6:2C 58 FF     SETTRM   BIT  IORTS     SET TERMINAL MODE
C2B9:C9 98                 CMP  #$98      IF CONTROL-X, GO DO IN#0
C2BB:F0 33                 BEQ  RETRN1    AND RETURN WITH CHAR
C2BD:49 E3                 EOR  #$E3
C2BF:C9 70                 CMP  #$70      IF CONTROL-S, GOT TURN ON
C2C1:F0 8F                 BEQ  SETCTL    BREAK AND LOOP
C2C3:49 6B                 EOR  #$6B      CONTROL-H?
C2C5:F0 04                 BEQ  SETPRT    YES
C2C7:C9 0E                 CMP  #$0E      CONTROL-F?
C2C9:D0 E5                 BNE  RCLNK     NO--RESET ACIA
C2CB:6E F8 06     SETPRT   ROR  PRNT      YES--SET PRINT FLAG
C2CE:70 E0                 BVS  RCLNK     AND LOOP
C2D0:             *
C2D0:             *
C2D0:68           OUT      PLA
C2D1:8D 78 05     OUT1     STA  ASAVE
C2D4:BD 80 C0     OUT2     LDA  STATUS,X  LOOP UNTIL THE
C2D7:29 02                 AND  #$02      ACIA TRANSMIT
C2D9:F0 F9                 BEQ  OUT2      BUFFER IS EMPTY
C2DB:AD 78 05              LDA  ASAVE
C2DE:9D 81 C0              STA  DATA,X    OUTPUT CHARACTER
C2E1:AC F8 06              LDY  PRNT      GET FLAG AND SKIP
C2E4:30 05                 BMI  OUT3      COUT1 IF FULL DUPLEX
C2E6:08                    PHP
C2E7:20 F0 FD              JSR  COUT1
C2EA:28                    PLP
C2EB:70 81        OUT3     BVS  IN2       LOOP IF TERMINAL MODE
C2ED:             *
C2ED:0E F8 06              ASL  PRNT      RESTORE PRINT FLAG
C2F0:             *
C2F0:8D 78 05     RETRN1   STA  ASAVE
C2F3:50 03                 BVC  RETRN2
C2F5:20 89 FE              JSR  SETKBD
C2F8:68           RETRN2   PLA
C2F9:AA                    TAX
C2FA:68                    PLA
C2FB:A8                    TAY
C2FC:AD 78 05              LDA  ASAVE
C2FF:60                    RTS
```

APPENDIX C

INTERFACING THE 7710 TO VARIOUS RS-232-C DEVICES

Despite the RS-232-C standards, handshaking for serial devices is far from standardized. If you plan to use the 7710 as a serial interface, you must make sure that the 7710 finds the proper signals on the interface pins, altering the cable or adding jumper wires as required. Interface wiring for many of the common serial devices is given below. If your device is not mentioned here, you will need to compare the 7710 handshaking with the device's handshaking as described in the printer manual, then construct or alter your cable accordingly.

In the following listing, straight through means that a standard RS-232-C cable that makes all 25 pin connections will work. However, if you intend to make your own cable, you need not connect all 25 pins; only pins 1-8 and 20 are used by the 7710. The standard null modem connections indicated for many of the devices listed below are as follows:

7710 pin	2	3	4	5	6	20	7
to							
other device	3	2	5	4	20	6	7

These data are based both on our own experiences and on feedback from our customers and dealers. Due to changes that other manufacturers make in their equipment and to inaccurate feedback, in some cases the information may not be entirely correct. The configurations about which we are uncertain are marked "unconfirmed".

IMPORTANT: Printers often have a bewildering array of switches that control such operating parameters as data format, baud rate, print style, and others. You must have these switches set correctly or your printer will not work properly (if at all). In some cases we have tried to help you in sorting out the various switch settings, but there are some general guidelines that might help in difficult situations. In general you will have to consult your printer manual for information about these switches.

Number of stop bits: If your serial device gives you an option for 1 or 2 stop bits, set it to 2 stop bits.

Parity: Set your serial device to no parity (if possible).

Number of data bits: This can be 7 or 8, depending on your serial device. Try 8 first and then 7 if that does not work. Some printers print italics, Greek letters, or graphics if the printer is set to 8 data bits.

Handshaking: The 771Ø firmware looks at the state of pin 4 (Clear to Send) to see if it should send data. If the voltage applied to pin 4 by your serial device is high (+12 volts typically), the 771Ø will send data. If not, it waits until the voltage goes back high. Printers use either pin 4, 11, 19, or 2Ø to tell the 771Ø whether they are ready to receive data or not. Therefore, you should connect one of these to pin 4 of the 771Ø. If your printer (or other serial device) is not in the following list, you must determine which pin (4, 11, 19, or 2Ø) provides this handshake signal and connect that pin to pin 4 on the 771Ø. Some printers have the option to use hardware handshaking such as this or to use software handshaking (XON-XOFF). If so, they usually have an internal DIP switch that selects whether you want to use software or hardware handshaking. You should set this switch to the hardware handshaking position (often called PRINTER READY/BUSY).

CONNECTIONS FOR SPECIFIC SERIAL DEVICE (IN ALPHABETICAL ORDER BY MANUFACTURER)

Anadex Printers (all Anadex printers use same interface)

```
771Ø    2  3  4  7  6-2Ø
 to
Anadex  2  3 19  7
```

You may also need to connect 2Ø - 2Ø and make sure RV. channel signal enabled on printer. There are 3 DIP switch blocks. On the third make sure 7 and 8 are on, and 9 and 1Ø are off. This selects 8 data bits, 2 stop bits, and no parity.

Anchor Signalman Mark I Modem (Anchor Automation, Inc.)
Standard null modem connections

Anderson Jacobson Modem (1235)

```
771Ø    7  2  3  4--6--2Ø
 to
A.J.    7  3  2
```

Anderson Jacobson 242A Modem
Standard null modem connections

Anderson Jacobson Printers

```
771Ø  2  3  4  7  2Ø
 to
 AJ   2  3  2Ø 7  4
```

Refer to the installation section of your printer manual for instructions on enabling the DTR signal. This signal is disabled when the printer is shipped, but must be enabled when the printer is used with the 771Ø. Also, check the data format; AJ may want 7 data bits.

Apple Imagewriter Printer

```
771Ø    2  3  7  4  2Ø
 to
Apple   2  3  7  2Ø 4
```

Set DIP switch 2-3 to the DTR position, not the XON-XOFF position

Apple Letter Quality Printer (Qume)

```
    7710      2  3  4  20  6  7
     to
    Apple     2  3  20  4  6  7    (Be sure DTR is enabled with DIP switch)
```

Apple Plotter

standard null modem connections

Axiom IMP-20 Printer

```
    7710    3  7  4  8--20
     to
    Axiom   3  7  20
```

Bascom Turner Digital Recorder Model 4120

```
    7710   2  3  4  5  7  8--20
     to
    4120   2  3  4  5  7
```

Bell 212 Modem (or 212A) and Bell Dataset Model 103J-L1-2 Modem

```
    7710      2  3  4  8  6  20  7
     to
    Bell      3  2  8  4  20  6  7
```

BMC PB101 printer (emulates the Brother HR1

Brother HR1

```
    7710   2  3  4   5  6  7  8--20  (unconfirmed)
     to
    HR1    2  3  20  5  6  7  8
```

Canon Electronic Typewriter (AP91 or AP92 interface)

straight through

Centronics Serial Interface Printers

```
    7710  3   4   6  7  8  20
     to
    Cent. 3  11*  6  7  8  20
```

*This Centronics printer signal is of the wrong polarity for use with the 7710 and must be inverted if the printer is to be operated at a high baud rate. To do this take the cover off the printer and locate switch pad #2. On this switch pad flip DIP switch 8 to the ON position. If this fails, call Centronics.

If operation at 300 baud is acceptable, connect pin 4 of the 7710 interface to pin 4 of the printer. Also on 704 set switches to enable Rev. ch.

On the 739 you must make sure that you have the printer selected to 8 data bits for operation under CP/M. Do so by switching switches 2 and 3 off on switch pak #1.

C-ITOH Printers

```
    7710     3   4  7  6--20
     to
    C-ITOH  3  20  7
```

Comr

ex CR1 Printer
straight through with 4—20

This printer has 2 rows of DIP switches, an upper row and lower row, having the following functions:

Upper Row
- 1-6 set the baud rate
- 7 selects no of data bits, put in open position for 8 data bits
- 8 parity, open=even, closed=odd

Lower Row
- 1-4 hammer impact intensity, set as desired
- 5 paper media select, set as desired
- 6 format control, set as desired
- 7 auto line feed, set to off
- 8 serial/parallel, do not change

Computrend Acoustic Modem
Standard null modem connections

Daisywriter
7710 3 7 4—6—20 (unconfirmed)
to
D.W. 3 7

DTC 380Z Printer (Data Terminals & Communications)
A straight through connection can be used, with the following caution. The printer has a 37 pin connector and is in some cases supplied with a cable which has the 37 pin connector on one end and a normal DB-25 connector on the other. This cable must be wired correctly for a straight through connection to work. If not, the following description of the signals at the 37 pin connector should help in configuring this or any other cable available:

pin No.	Signal
5	protective ground
12	transmit data (from printer)
35	receive data (to printer)
31	RTS (from printer)
16	CTS (to printer)
34	DSR (to printer)
6	signal ground
13	DTR (from printer)

There are a large number of switches to set on the three blocks of DIP switches.

Switch Block 1 (upper switch block)
- 1- off parallel ready/busy polarity (not applicable here)
- 2- off busy/strobe stuff (not applicable here)
- 3- off auto line feed
- 4- off debug mode
- 5- off debug mode select
- 6- off off selects 6 lines/inch, on selects 8 lines/inch
- 8- ?

Switch Block 2 (middle block)
- 1- off select serial or parallel interface
- 2- off selects 2 stop bits (on selects 1 stop bit)
- 3- off or on (off=even parity, on = odd parity)

4- off selects 8 data no parity (on = 7 data + parity)
5-8 baud rate switches (usually set to 9600)

Switch Block 3 (bottom block)

1- off demonstration
2- off continuous demonstration
3- off self test mode
4-8 reserved for future use

Data I/O System 19 PROM/ROM Programmer
straight through

Data South printer

```
7710   3  4  7  8  20
 to
 DS    3 20  7  8  4
```

Datec 212 Modem
standard null modem connections

DECwriter (LA34 or LA 38 confirmed)

```
7710  2  3  7  6  20--4
 to
DEC   2  3  7  20  6
```

DEC LA 50

```
7710    3  7  4  8--20
 to
LA 50   3  7  11
```

(Set the XON-XOFF/Printer Ready busy switch to the Ready/Busy position and the polarity switch to the busy low/ready high position)

DECwriter LA 120

```
7710  2  3  7  4--8--20
 to
DEC   2  3  7
```

Diablo 630

```
7710  2  3  4  5  6  7  8  20
 to
630   2  3  20 5  6  7  8  4
```

Install a jumper plug connecting pins 5 and 6 of A60 on HPR05 to enable the DTR signal.
If your 630 has the API (All Purpose Interface), then the printer buffer signal is already on pin 20. You must order a cable from Diablo than will run from the 50 pin connector on the Diablo to a 25 pin DB-25 connector that you can plug into the 7710. Also, switch 7 on the right bank of DIP switches must be up. In the down position the DC1-DC3 software protocol is selected. In up position, pin 20 is toggled upon printer buffer full.

Diablo 620
same as 630 (put switch 4 on the printer to the ready/busy position

Diablo 1600 and Xerox 1700 Series
Same configuration as Diablo 630. Remove printer cover. Unplug connector at HPR04 module. Use a paper clip to push out the wire (may be a black wire) on pin 3, switch it to pin 2. If there is a grey wire on pin 2, move it to pin 3. If you have problems, call a Diablo

or Xerox service center and tell them you want to activate the DTR signal.

Digitec Data Logger
Standard null modem connections

Digicom 213B Modem
Standard null modem connections

Dip Printer
```
7710   3  4  7  6--20   (unconfirmed, but based on data from DIP-81 manual)
 to
DIP    3  5  7  5--20
```

Epson CP-20 Modem
Straight through

Epson MX Series-- with 8141 interface
```
7710   3   4  6  7  20
 to
MX     3  11  6  7  20
```

Epson with 8145 or 8148 interface
```
7710   2  3  6  7  4    8--20
 to
Epson  2  3  6  7  20
```

Epson with 8155 interface (Golden Eagle interface)
```
7710    3  4  7  6--20     Be sure DTR is selected by jumpering 2-3 on
 to                        the 8155 (shipped that way).  Also switch S2
Epson   3  20 7            should be on and switch S1 off.
```

Gandalf Modem LDM 414
```
7710     2  3  4  5  7  6--20
 to
 ndalf   3  2  5  4  7
```

3404 Printer
```
7710    2  3  4  5  7  20   (factory dip switch settings work OK)
 to
 GE     2  3  11 5  7  20
```

Hays Smartmodem (300 or 1200)
Standard null modem connections

also confirmed to work:
```
7710  2  3  7  8  20--6--4  (no handshaking will take place)
 to
Hays  3  2  7  8  20
```

HEI Card Reader Model 121-4
```
7710   2  4  5  7   8--20
 to
HEI    2  4  5  7
```

HP plotter 7470A or 7475

```
    7710     2  3  7  4     (buffer full signal is on pin 20)
     to
     HP      2  3  7  20

HP A2905B Printer (a reconfigured Epson MX-80)
    7710     3  4  7  20--6      (same as IDS configuration)
     to
     HP      3  20 7

HP 2635A
     7710  2  3  7  4--8--20     (uses XON/XOFF software handshake)
      to
     HP    2  3  7

Houston Instruments Plotter (confirmed)
    You can select either software or hardware handshaking with the Houston
    Instruments Plotters.  If you select software use these connnections:

    7710       2  3  7  4--6--20
     to
    plotter    2  3  7
    If you use hardware handshaking use these connections:
    7710       3  4  7     (straight through OK also)
     to
    plotter    3  4  7
    Pin 6 is on some of the Houston Instruments is used to select baud rate.
    Wiring pin 6 to one of the pins from 14-19 selects baud rate.

IBM Selectric  ESCON Interface
    7710     2  3  4  7  20--6
     to
    IBM      3  2  20 7

IDS Paper Tiger and  C-ITOH (also IDS Prism, unconfirmed)
    7710     3   4  7  20--6
     to
    IDS      3  20  7

Intertec Superbrain
    7710             2  3   4---5   6  7  20
     to
    superbrain       2  3       5   6  7  20

Kaypro Computer
    It appears (unconfirmed) that the Kaypro has 2 serial ports, one DCE and
    one DTE.  If using the DCE port, use the following:
    7710     2  3  7  6  4--8--20
     to
    Kaypro  3  2  7  20
    If you are using the DTE port, do the following:
    7710     2  3  7  20  4--8
     to
    Kaypro  2  3  7  20
```

Lear Siegler ADM-5

```
    7710     2  3  4  5  7  8  6--20  (unconfirmed, but based on data from
     to                                ADM-5 manual)
    ADM-5    2  3  4  5  7  8
```

LEX-11 Modem (Lexicon Corp. of Miami, FL)

```
    7710     2  3  4  7  6--20
     to
    LEX-11   3  2  5  7
```

LEX-12

Use standard null modem connections

MPI Printmate 99

```
    7710        3  4  7  20
     to
    Printmate   3  11 7  20
```

NEC 3510

```
    7710     3  4  7  2  5  6  8  20
     to
    NEC      3 19  7  2  5  6  8  20
```

Pin 19 carries the buffer full signal, which must be set to high active by setting the Reverse Channel dip switch in the OFF position position. This switch is located either behind the form length thumbwheels on operator control panel or on the back left side on a vertical PC board on some of the newer spinwriters.

NEC 5515 (xx15 series Spinwriters emulate the Diablo 1600 series)

```
    7710   2  3  5  6  7  8  4   20   Set switches on NEC to enable XON-XOFF
     to
    NEC    2  3  5  6  7  8  20  4
```

Novation Cat

Standard null modem connections

Novation D-Cat (unconfirmed, but based on data from Novation, 4/15/83)

```
    7710     2  3  4  7  6--20
     to
    D-Cat    3  2  5  7
```

Okidata Microline

```
    7710     3  4  7  6-20
     to
    Oki      3  11 7  6-20
```

May have to set to 7 data bits. There are 8 DIP switches on the control panel and 2 banks of 8 on the serial interface itself. Set them as follows:

Main control panel settings

1-on

2 through 8-off

On serial interface

1-off

2 through 8-off

9 through 12-on
13 through 16-off

Okidata with Superspeed Serial Option
Straight through
Set the dip switches on the Okidata such that the buffer full signal is on pin 4. Also, it is necessary to set the Okidata to use 7 data bits. One pattern of dip switch settings that a customer said worked with his printer is as follows:
On- 1,2,5,6,7,10,11,12,14 (These settings are suspect)
Off- 3,4,8,9,13,15,16

Olivetti Typewriter (VDS TP121S interface)
7710 3 7 20—4
to
Olivetti RD(white) SG(black) HK (red)

Olivetti Typewriter ET121 with NDF interface
7710 2 3 4 6 7 8—20 (select printer/busy handshaking, not ETX-ACK or XON-XOFF)
to
Olivetti 2 3 20 6 7

Olympia Printer or Typewriter
7710 3 7 4 6—20
to
Olympia 3 7 20

Olympia ES-100
7710 3 4 7 8—20
to
Olympia 3 4 7

Pioneer Universal External Interface
straight through

Precision Microproducts EPROM Programmer Model EPUP
7710 2 3 7 4—6—20
to
EPUP 3 2 7

Qume Printers
With the printer's Modem/No Modem switch in the Modem position, a Qume printer can be connected to the 7710 with a standard RS-232-C cable. The necessary connections are:
7710 2 3 4 5 6 7 8 20—4
to
Qume 2 3 4 5 6 7 8 20
May have to use 1 stop bit. To do so, change the ACIA to decimal 21 ($15). Set parity switch to no parity and enable DTR (pin 20).

Racal Vadic DA 212PA modem
standard null modem connections

Radio Shack Modem I
standard null modem connections (confirmed)

```
Robert Shaw Control Processor
    7710    2  3  7  4-6-20   (unconfirmed)
     to
    C.P.    2  3  7  4-5  6-8-20

Smith Corona TPl  (Preliminary)
    7710     3  4  7  20   (Change ACIA command byte to $19 or decimal 25.
     to                       This sets 8 data bits, 1 stop, and even parity)
    TPI     3  4  7  20

SSM Transmodem 1200
     standard null modem connections

Swintec Printer  (emulates Olympia ES101)
     7710     3  4  7  8--20
      to
     Swintec 2  5  7

Star Micronics Gemini 10 Printer
    7710     2  3  4  5  6  7  8--20
     to
    Gemini  2  3  20 5  6  7  8

Talley 1612
    7710    2  3   4  5  7  20
     to
    1612    2  3  19  5  7  20

Texas Instruments TI800 Series
    7710 pin   3   4  6  7  8  20-6
      to
    TI pin     3  20  6  7  8

Texas Instruments TI 700
    straight through  (unconfirmed)

Tokyo Electric (TEC) F10-40 daisy wheel printer
    straight through

Toshiba P1350 (letter quality dot matrix printer)
    straight through (emulates Qume Sprint 5, uses ETX-ACK)
    Set the jumpers on the printer as follows:
    1- there are 3 options, which are connect C.D. to +5, to ground, or to
       cable.  Set it to connect to cable.
    2- two options, DSR to ground or to cable.  Set it to connect to cable.
    3- select the option that connects CTS to cable
    4- S.D., cannot be changed
    5- don't change
    6- DTR, don't change
    You must select 7 data bits on the printer or it will print only graphics.
```

Transtar Printer 130S

7710	2	3	4	20	5	6	7	8	20
to									
130S	2	3	20	4	5	6	7	8	20

Trendstar Printer

7710	3	7	4	6—20
to				
Trendstar	3	7	20	

UDS modem model 202LP (1200 baud)
straight through (modem is DTE)

UDS modem model 212 LP

7710	2	3	4	7	20
to					
UDS	3	2	5	7	6

Universal Data Systems 212 Modem
straight through

US Robotics Modem
standard null modem connections (unconfirmed)

Votrax Type 'N Talk

7710	2	3	4	5	7	8 20
to						
Votrax	2	3	5	4	7	8—20

Votrax Personal Speech System
straight through

Xerox 620 Memorywriter

7710	2	3	5	6	7	8	4	8—20	(unconfirmed, but based on data from
to									Xerox 620 manual)
620	2	3	5	6	7	8	20		

APPENDIX D

CHECKOUT

Your 771Ø has been fully tested, but you may wish to test it yourself. The simple tests described in this section check most of the circuitry of the 771Ø.

Please note that all 771Ø cards are shipped with PROMs. Unless you remove these ROMs and substitute RAMs, you should not run Test 2. Test 1 is valid for substitue PROMs, but you must, of course, compare the screen display with a correct program listing (see Appendix B for PROM listings).

D.1 TEST 1: PROM TEST

This test displays the contents of the PROMs on the video screen, verifying that the PROMs can be read by the computer, and allowing you to compare the contents with the program listings provided in Appendix B.

a. From BASIC type CALL -151 <cr> to enter the monitor.

b. Type CnØØL <cr>, where n is the slot number of the 771Ø.

c. Compare the appropriate listing from Appendix B with the listing you see on the screen.

d. To list more of the program, type L <cr>.

e. Repeat steps c and d until you have viewed the entire PROM.

You are looking for differences between the hexadecimal code on the screen and what is given in Appendix B. There are a few locations that may be different for reasons given in the next paragraph, but in general a bad PROM is obvious by the appearance of question marks or BRK instructions in the positions where legitimate 65Ø2 instructions should be.

NOTE

The disassembler in the Apple system monitor cannot recreate assembler pseudo-operation codes such as ORG or DFB. Occasionally, use of the ORG instruction could hide an instruction from the disassembler. For instance, the code:

```
BCS  *
ORG  *-1
SEC
```

will disassemble as

```
BCS  *+$38
```

Watch for this programming trick when comparing the listings. It is a valid code, but may make you think you have bad PROMs. Programming tricks such as this are used to conserve memory in tight situation. This trick is used once in the 771Ø-Ø2 firmware, and in general a bad PROM will show numerous differences from the correct listing.

D.2 TEST 2: RAM TEST

This test verifies that you can read from and write to all locations of the program RAM, and is to be used <u>only</u> if you have replaced the PROMs with RAMs. A 256 bytes segment of your computer's firmware is copied into the RAMs, then the copy is compared to the original. Errors are displayed on the screen.

a. From BASIC type CALL -151 <cr>.

b. Type CnØØ<FØØØ.FØFFM <cr> (n=slot number of 771Ø)

c. Type CnØØ<FØØØ.FØFFV <cr>

d. A * should appears almost immediately on the screen. If not, rerun the test to determine if you made a mistake. If you still have problems see your CCS dealer.

D.3 TEST 3: SERIAL DATA LOOP TEST

This test checks the ACIA, the clock, and the line drivers by transmitting a known byte of data, looping it back to the receiver, reading the data, and comparing the result. For this test, you need a "loop-back" test fixture. To make one, obtain a standard male DB-25 connector and wire pins 2 and 3 together, pins 4 and 5 together, and pins 8 and 2Ø together. This fixture allows transmitted data to be looped back into the ACIA receiver. An alternative, if less reliable, way to make these connections is to simply push pieces of wire into the end of the female connector supplied with the 771Ø. One piece of wire goes into the hole for pin 2 and the other end of this piece of wire goes into pin 3 and so on.

a. Disconnect any cable going to your peripheral device and connect the loop back test fixture in its place.

b. Verify that the External baud rate is NOT selected.

c. From BASIC type CALL -151 <cr>

d. Type: CØnØ:3<cr> to reset the ACIA. (n=8+slot number)

CØnØ:11<cr> to initialize the ACIA

CØn1:55<cr> to write an alternating bit pattern.

CØn1<cr> to read the received data.

f. Compare the displayed value with the value transmitted.

g. Repeat step e using different baud rates, ACIA commands, and data patterns until you are satisfied that the interface works properly.

APPENDIX E

TROUBLESHOOTING GUIDE

E.1 INTRODUCTION

This appendix is a step-by-set guide for what steps to take if you have not been able to get the 771Ø to communicate correctly with your particular serial device. Although a careful reading of the previous sections of this manual should normally be sufficient to enable you to accomplish your interfacing needs, this section attempts to anticipate any problems you might have. This section is a compilation of solutions to problems encountered with numerous 771Ø-serial device interfacing situations in the past. Potential problems are listed in order of their frequency of occurrence.

E.2 BAUD RATE

The baud rate must be set to the same value on both the 771Ø and your serial device. Directions for setting the baud rate of the 771Ø are given in Section 2.1.1 of this manual. Consult the manual supplied with your serial device for directions on how to set its baud rate.

E.3 ORIENTATION OF THE SUPPLIED CABLE

The short cable supplied with the 771Ø should be connected to the 771Ø such that pin 1 of its RS-232-C connector goes to pin 1 on the J2 connector (26 pin) on the card. Pin 1 of the J2 connector is marked and is located on the side of the connector that faces the front of the Apple (keyboard side). The pin 1 side of the cable is marked with a red (sometimes blue) wire. The cable should be plugged onto J2 such that the red wire of the cable goes to pin 1 of J2. Thus, the red wire should be facing the front of the computer.

A few of the short cables supplied with the 771Ø were assembled incorrectly, such that pin 1 of J2 got connected with pin 25 of the RS-232-C connector. To determine is you received such a cable, refer to the drawing below. The DB-25 connector is arranged such that pin 1 and pin 25 are on opposite sides of the connector. If the red wire of your cable runs from the front side of J2 to the pin 25 side of the DB-25 connector on the cable, then you have an incorrectly assembled cable. See your dealer for a replacement.

Even if you have a good cable, it is possible that you plugged it onto the 771Ø in the wrong orientation. Because pin 25 is connected to +12 volts and R1 is a 1Ø ohm resistor, a connection made in the wrong orientation may direct current flow to ground through R1. The resulting current flow will burn up this resistor. However, even if this happened, the 771Ø may still be functional if R1 did not get hot enough to damage components near it, because most serial devices do not use pin 25. Reorient the cable and try the card again.

E.4 THE PROMS

If you have not already done so, perform the PROM test detailed in Appendix A, paragraph D.1. perform the PROM test detailed there. If this test fails, you have bad PROMS, which must be replaced. (The card may function even with bad PROMs with some communications programs, which do not use the PROMs)

E.5 CABLING PROBLEMS

Cabling between the 771Ø and your RS-232-C serial device is often the cause of problems. Because the RS-232-C "standard" can be applied differently by different manufacturers, do not assume that you can use a standard 25 pin cable between the 771Ø and all serial devices. Although this may be true for interfacing the 771Ø to many video terminals, it is usually not the case with printers and modems. Still, there are certain conventions that are followed by almost every manufacturer, such as using pin 7 as the signal ground and pin 1 as the protective ground (also called chassis ground). Table E-1 gives the most common configuration found, one that comes closest to being the current "standard":

TABLE E-1 A TYPICAL RS-232-C PIN CONFIGURATION

Connector Pin No.	Pin Name	Pin Function
1	Protective Ground	Protective ground
2	Transmit Data	Transmits or receives data
3	Receive Data	Transmits or receives data
4	Request to Send	Handshake line
5	Clear to Send	Handshake line
6	Data Set Ready	Handshake line
7	Signal ground	Signal ground (must be connected)
8	Carrier Detect	Signals a connection to a distant computer (e.g., by a modem)
2Ø	Data Terminal Ready	Handshake line

E.5.1 HANDSHAKE LINES

Pins 4, 5, 6, and 2Ø are handshake lines, which carry signals that can take on one of two logical states, either "high" or "low" (high is some voltage above +3 volts and low is some voltage below -3 volts. These signals are used by one serial device to tell the other to transmit data or not to transmit data. For instance, pin 4 of the CCS 771Ø serial interface is the Request to Send line, which must be logical high in order for the 771Ø to send or receive data. Similarly, pin 2Ø (Data Terminal Ready) of the 771Ø must not be low or data will not be transmitted or received by the 771Ø.

E.5.2 TRANSMIT AND RECIEVE DATA LINES

Notice that there is an ambiguity indicated for pins 2 and 3, the transmit data and receive data lines. The line with the name Transmit Data can sometimes actually be the receiving line and the line with the name Receive data can actually be the transmitting line. This is because a serial interface can take on one of two roles, either the role of a transmitting device (called the DCE configuration) or the role of a receiving device (the DTE configuration). A device with the DCE configuration sends data out on pin 3 and receives data on pin 2, in apparent conflict with the names for these two pins. In the DCE configuration the handshake lines 4 and 2Ø are input signals and the handshake lines 5 and 6 are output signals.
In the DTE configurations these roles are reversed: 2, 4, and 2Ø are outputs signals and 3, 5, and 6 are input signals. For this reason, manuals that give only the pin names that they use on their interface do not supply you with the complete information that you need to connect their device to the 771Ø (or any other serial interface). You must also know whether the device is configured as DTE or DCE. Some manuals go one step better and tell which direction each signal is going. For instance, in this manual, the arrows in figure A.2 tell in which direction each signal goes relative to the 771Ø.

E.5.3 NULL MODEM ADAPTOR (DCE-TO-DTE ADAPTOR)

Now it should be obvious that connecting two serial devices together with a standard RS-232-C cable that makes all 25 connections will not work if both devices have the same configuration. Two devices with the same configuration (DTE or DCE) both attempt to send data on the same pin, which cannot work: the send data pin of one device must be connected to the receive data pin of the other device. Similarly, the sending side of the handshake lines must be routed into the receive side of handshake lines. This is the situation you will usually face when you want to use the 771Ø with a modem. In this case a DCE to DTE adaptor is called for. Such an adaptor, which is often called a null modem, makes the following pin connections:

771Ø pin	other serial device pin
1	1
2	3
3	2
4	5
5	4
6	2Ø
2Ø	6
7	7

This null modem adaptor is available commercially from computer stores and electronics stores but can often be expensive. It can be made with about $1Ø worth of parts according to instructions given in Section 2.4 of this manual. This adaptor is inserted between the short cable supplied with your 771Ø and the cable leading from your serial device. This arrangement has the advantage of keeping all your expensive cables intact and also it is easier to rearrange than a ribbon cable.

Appendix C of this manual contains the connections that are necessary to make a large number of serial devices work with the 771Ø. If your device is listed there, simply make the connections given. If not, you will have to decide what connections to make. Here are the things that you will have to do in order to accomplish this:

a) Wire pin 7 of the 771Ø (function: signal ground) to pin 7 of your device. This is almost always necessary. It is not necessary to wire pin 1 of the 771Ø to your device.
b) Wire pin 2 of the 771Ø (function: receives data) to the pin of your device that transmits data (either pin 2 or 3, depending on whether the device is configured DTE or DCE).
c) Wire pin 3 of the 771Ø (function: transmits data) to the pin of your device that receives data (either pin 2 or pin 3, depending on whether the device is configured DTE or DCE).
d) Pin 4 of the 771Ø (function: handshake line that must be logic high for the 771Ø to transmit) should be wired to pin 4 of your device if the device is DTE or pin 5 if the device is DCE. If your serial device does not support an outgoing handshake on pins 4 or 5, check to see if it supports an outgoing handshake on pin 2Ø or some other pin. If so, wire that pin of your serial device to pin 4 of the 771Ø. On some printers this might be pin 11 (as for Epson printer) or pin 19 (as on the NEC Spinwriter). On some printers, a dip switch inside the printer will have to be set in a certain position in order to enable the signal on pin 19 or 2Ø to tell the 771Ø that the printer buffer is full.
e) Pin 5 of the 771Ø (function: outgoing handshake line) should be wired to pin 5 of your serial device if the device is DTE or pin 4 if it is DCE. If your device does not support an incoming handshake on pin 4 or 5, then you do not need to connect pin 5.
f) Pin 6 of the 771Ø is always set to logic high. If your serial device does not support any handshaking (on pins 4,5,6, or 2Ø), then wire pin 6 of the 771Ø to pins 4 and 2Ø of the 771Ø. In this situation, only pins 2, 3, and 7 go between the 771Ø and your serial device. If your device does support an incoming handshake on either 6 or 2Ø, wire 771Ø pin 6 to your device's pin 6 if it is DTE or pin 2Ø if it is DCE.
g) Pin 2Ø of the 771Ø (function: incoming handshake line that must logic high for the 771Ø to transmit) should be wired to pin 2Ø if your device supports pin 2Ø and is DTE or pin 6 if your device supports pin 6 and is DCE.
h) Your serial device might support handshaking on pins 4 and 5 but not on 6 and 2Ø, in which case you should make pin 2Ø of the 771Ø high by connecting the 771Ø pins 6 and 2Ø together.

In many cases the manual that comes with your particular serial device does not tell whether the device is configured as DTE or DCE and does not tell the signal direction of the various pins. This makes things difficult for everyone concerned. If your device is not listed in Appendix C of this manual, then you may want to call the Customer Service or Technical Support Department of the company that manufactures your serial device in order to obtain this information.

E.6 SERIAL DATA LOOP TEST

You can test the majority of the circuitry on the 771Ø by performing the Serial Data Loop Test described in section D.3 of Appendix D. If the 771Ø fails this test, and assuming you performed the test correctly, it is probably bad. Return it to your dealer for servicing.

E.7 7710 VERSIONS

As explained in Chapter 1 of this manual, the 7710 comes in two versions, the 7710-01 and the 7710-02, which differ only by the machine language driver program they contain in the two Programmable Read Only Memory (PROM) chips at positions U5 and U6 on the board. You may be trying to use commands that exist only in the version you do not have. For instance, attempting to invoke terminal mode when you have the 7710-01 version is an exercise in futility, since only the 7710-02 has the capability to go into terminal mode. Chapter 3 of the this manual details the capabilities of the two firmware versions.

E.8 COMMERCIAL SOFTWARE COMPATIBILITY

Some commercially-available software packages may not, for a number of reasons, work with the 7710-01. For instance, at this time the popular communications program Visiterm (from Visicorp of San Jose) will work with the 7710-01 but not the 7710-02. Programs that make use of the peripheral card RAM space (specific locations in the area between $478 and $7FF) may be incompatible with the 7710, since the 7710 driver firmware uses some of these locations for temporary data storage. If you suspect a software incompatibility with the 7710, contact the vendor of the software for more information.

E.9 DATA FORMAT PROBLEMS

The 7710 defaults to using 8 data bits, 2 stop bits and no parity. If possible, your peripheral device should be set to match this configuration. Such operating parameters are usually set by putting various dip switches on the serial device in question in various positions. If your device cannot be changed to match the 7710's default values, you will be forced to change the 7710 to match your serial device's data format. Instructions for doing so are given in Chapter 4 of this manual.

E.10 DEFECTIVE CARD

If you have gone through all of the considerations above and still cannot get your 7710 to function, then it is possible that you have a defective card. Assuming the card has passed the serial loopback test, the next step is to perform the PROM test described in Appendix D. Note that the card can pass the serial loopback test even with bad PROMS. If your card fails the PROM test then the PROMS must be replaced.

E.11 COMMUNICATIONS PROTOCOLS

Many users will want to use serial devices that for one reason or another use software communications protocols to accomplish handshaking with the 7710. The most commonly-used such protocol is XON-XOFF (also called control-S, control Q or DC1-DC3), in which the serial device tells the 7710 that it cannot accept any more data by sending a Control S character to the 7710, which must have software to recognize that a Control S has been received. After sending the Control S to halt the 7710, the host device signals that it can again accept data by sending a Control Q to the 7710, which responds by resuming its data transmission. Other software handshake protocols are similar, but use different ASCII characters. The 7710-01 does contain code for responding to Control S, so handshaking with this protocol is possible, while the 7710-02 firmware cannot respond to this protocol. Neither version of the 7710 has provision for responding to other software handshaking protocols, but those writing their own driver routines can implement these schemes.

E.12 DEFECTIVE TELEPHONE OR MODEM

If you are using a modem with a telephone (acoustic coupler), you might consider the possibility that either your modem or your telephone is defective.

E.13 RETURNING THE 7710 TO CCS FOR REPAIR

If you are convinced that your 7710 is defective and that you would like to return it to CCS for repair, you must first call Customer Service at CCS for a Return Material Authorization number. If this number is not clearly visible on your shipping container, our receiving department will refuse to accept the card. Send the card to:

Customer Service
California Computer Systems
250 Caribbean Drive
Sunnyvale, CA 94086

Do not include your manual or the warranty card, but do include proof of purchase date if the card is still under warranty.