

HEWLETT-PACKARD

HP 82941A

# BCD Interface

OWNER'S MANUAL

SERIES 80





# **HP82941A BCD Interface**

## **Owner's Manual**

### **Series 80**

**April 1982**

Reorder Number  
82941-90011

## General Information

### Introduction

The HP 82941A BCD Interface provides the HP Series 80 Personal Computers with the capability of interfacing up to two peripheral instruments that handle numeric information in BCD (binary coded decimal) form. These instruments fall into a variety of classifications including voltmeters, multimeters, counters, and any peripheral that outputs and/or inputs BCD information.

This manual provides general information about the interface, how to install it and a guide to programming to provide the best use of the interface features.

### I/O ROM

To operate the interface, the I/O ROM (P/N 00085-15003 for the HP-85 and HP-83 or P/N 00087-15003 for the HP-87) is required. This ROM fits into a ROM drawer which plugs into any of the four I/O ports in the back of the computer.

### Specifications

Pertinent specifications are given in this section.

#### Power Requirements

The computer supplies all power for the interface.

#### Operating Temperature

0°C to 55°C (32°F to 131°F)

#### Logic Requirements

Positive true or negative true logic may be used. The logic configuration may be set by switches on the interface circuit board or by software.

#### Level Requirements

Input low: .8V @ 1mA.

Output low: .45V @ 4.5 mA.

Input high: 2.0V.

Output high: 2.4V @ 240  $\mu$ A.

#### I/O Data Lines

44 bi-directional lines via eleven 4-bit ports (P0 through P10.)

#### Sign Bits

Four sign bits (S1 through S4) via port P11.

**Handshake Lines**

Four output lines (I/OA, I/OB, CTLA, CTLB).

Two input lines (FLGA, FLGB).

**Standard Data Format**

The interface can recognize sixteen different BCD characters and two signs from the BCD bus. These are listed in table 1-1.

**Table 1-1. BCD Characters**

BCD Code		Representation	Signs		Representation
Positive True Logic	Negative True Logic		Positive True Logic	Negative True Logic	
0000	1111	0	0	1	+
0001	1110	1	1	0	-
0010	1101	2			
0011	1100	3			
0100	1011	4			
0101	1010	5			
0110	1001	6			
0111	1000	7			
1000	0111	8			
1001	0110	9			
1010	0101	:(colon)			
1011	0100	;(semicolon)			
1100	0011	< (less than)			
1101	0010	= (equals)			
1110	0001	> (greater than)			
1111	0000	? (question mark)			

During an output, the interface accepts an ASCII code from the computer and outputs the lower four bits of this code to the BCD bus. However, ASCII codes 0 through 32, and code 44 are ignored by the interface.

**Cable**

The interface is supplied with a 31-conductor shielded cable connected to the interface on both ends. The cable is approximately 4 meters long. The cable structure is discussed in detail in section 2.

**Dimensions**

Approximately  $16.7 \times 12.7 \times 1.5$  cm ( $6.59 \times 5 \times .59$  in).



## Installation

### Unpacking and Inspection

If the shipping carton is damaged, ask the carrier's agent to be present when the interface is unpacked. If the interface is damaged or fails to meet electrical specifications, immediately notify the carrier and the nearest HP sales and service office. Retain the shipping carton for the carrier's inspection. The sales and service office will arrange for the repair or replacement of your interface without waiting for the claim against the carrier to be settled.

### Installation Considerations

Standard terms for BCD lines have not been adopted by different manufactures of BCD devices. In some cases you may need to know the purpose a line serves to know where to terminate it on a peripheral. The following installation options should be read and understood before installation is attempted.

The select code and default configuration switches may be set to meet the needs of your installation, or, many of the interface modes can be configured by software.

The switches are preset at the factory as follows:

- Select Code 3.
- Standard Format (1 channel).
- Trailing Edge Handshake.
- Positive True Logic On  
All Data Lines,  
Port 10,  
Port 11 (Sign Bits).
- Outputs Disabled.

The *select code* and *output disabled* are the two switch configurations that cannot be re-configured by software. These two items must be configured by the switches. Software can re-configure the interface and ignore the remaining switch settings.

If you wish to verify or change any of the above switch settings, refer to the following disassembly procedure and disassemble the interface. Then refer to the discussion of the switch settings. If this isn't necessary, refer to the discussion of the interface cable on page 00.

## Disassembly

Refer to figure 2-1 to see how the interface parts fit together. Place the interface on a flat surface with the side having the seven screws facing upward and the cable coming out to the left. Then use the following steps to disassemble the interface:

1. Use a Pozidriv® screwdriver and remove the screws.
2. Hold the interface parts together and turn the interface over so that the seven screw holes are facing downward and the cable is coming out to the left.
3. Hold the cable strain reliefs in place and remove the top half of the interface housing.

If you have followed the above steps, the switches should be oriented as shown in the following figures. If they are not, orient the switches as shown before identifying the switch segments.

After the switches are set to meet your needs, refer to the discussion on cables before you reassemble the interface since it may be necessary to remove one end of the cable from the interface.

To reassemble the interface, reverse the above procedure, making sure the ground contact is in place. The ground contact should be under the circuit board when the component side is up.

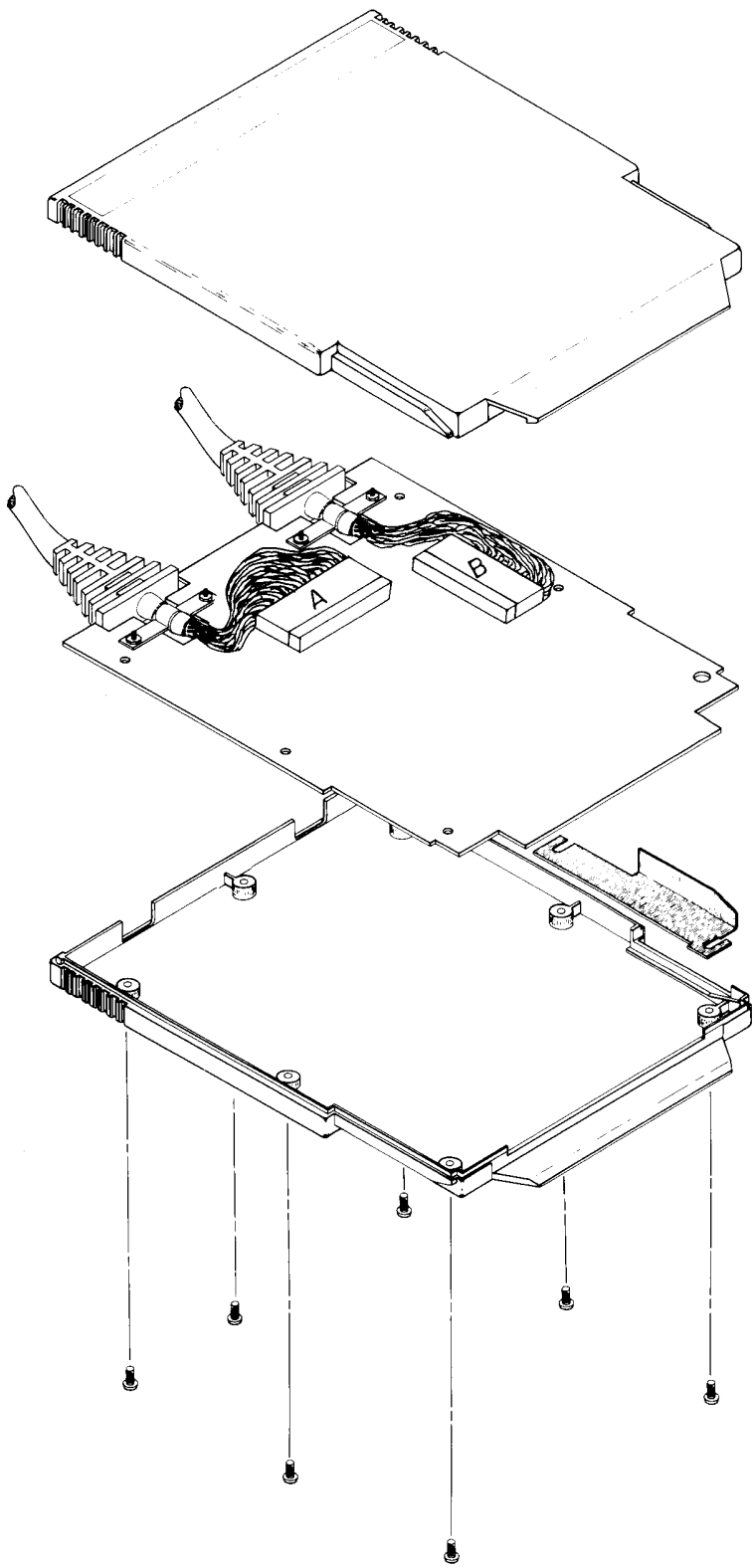


Figure 2-1. Disassembly

## Switch Settings

As previously noted, the select code and default configuration switches are preset at the factory. Once the interface is disassembled, any of the factory settings may be changed or verified by referring to the following discussions of these functions.

The interface may be equipped with either slide or rocker type switches. Both types are illustrated in the factory preset positions.

**Note:** If you change any of the factory settings, make sure that you change the proper switch segments. Do not disturb the settings of adjoining switches. The small tip of a pencil or similar object is recommended for this purpose.

## Select Code Switches

Switch segments 2, 3 and 4 of switch S1 are preset at the factory for select code 3 as follows:

- Switch segment 2 set to "0".
- Switch segment 3 set to "0".
- Switch segment 4 set to "0".
- Switch segment 1 is not used and may be in either position.

The "0" and "1" positions are labeled on the circuit board.

Select codes 3 through 10 may be set with these switch segments. To change or verify the factory settings, orient the circuit board as shown in figure 2-2 and locate switch segments 2, 3 and 4 of switch S1. Then identify the "0" and "1" switch positions on the circuit board. You may verify that select code 3 is properly set by comparing the actual positions of switch segments 2, 3 and 4 with those illustrated. They should be the same.

To change the select code, refer to the table of figure 2-2 and set switch segments 2, 3 and 4 as required for the select code chosen. For example, if select code 10 is to be set, the three switch segments must all be set to "1". In this case, if the switches are the slide type, slides 2, 3 and 4 must all be set to the "1" position; if they are the rocker type, all three rockers must be pressed down toward the "1" position.

**Note:** Select codes 1 and 2 are reserved for the CRT and (internal) printer, respectively.



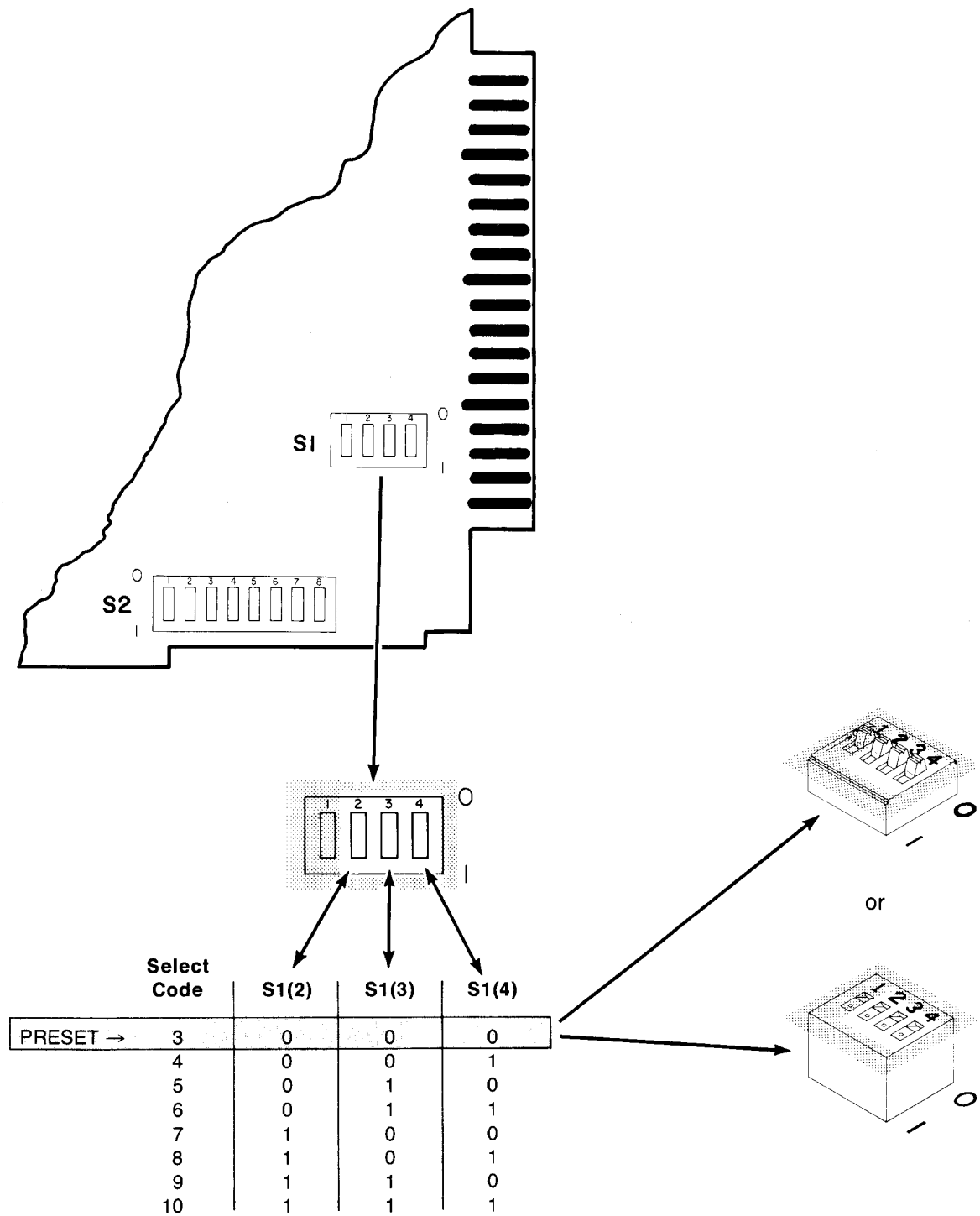


Figure 2-2. Select Code Switches

## Default Configuration Switches

Switch segments 1 through 8 of switch S2 are all preset to “0” at the factory to implement the following:

- Standard Format (1 channel)
- Trailing-edge Handshake
- Positive True Logic for
  - Data Lines
  - Sign Bits
  - Port 10
  - Control Lines
  - Flag Lines
- Output Disabled (software cannot re-configure this function)

The “0” and “1” switch positions are labeled on the circuit board.

To verify any of the factory settings, orient the circuit board as shown in figure 2-3 and locate switch S2. Then identify the “0” and “1” switch positions on the circuit board. The factory settings should have switch segments 1 through 8 in the “0” positions as illustrated.

To change any of the factory settings, refer to the table of figure 2-3 and identify the switch segments you wish to change. For example, if you want to enable outputs, you must set switch segment 8 to the “1” position. Any of the configuration switches may be changed to meet your needs. However, keep in mind that software can ignore any of the switch settings except switch segment 8 and the select code switches.

Switch Segment	Assignment	Set to "0"	Set to "1"
S2(1)	Format	Standard Format (1 channel)	Option Format (2 channels)
S2(2)	Handshake Lines	Trailing Edge	Leading Edge
S2(3)	Data Sense	Positive True Logic	Negative True Logic
S2(4)	Signs Sense	Positive True Logic	Negative True Logic
S2(5)	Port 10 Sense	Positive True Logic	Negative True Logic
S2(6)	Control Sense	Positive True Logic	Negative True Logic
S2(7)	Flag Sense	Positive True Logic	Negative True Logic
S2(8)	Output Enable	Outputs Disabled	Outputs Enabled

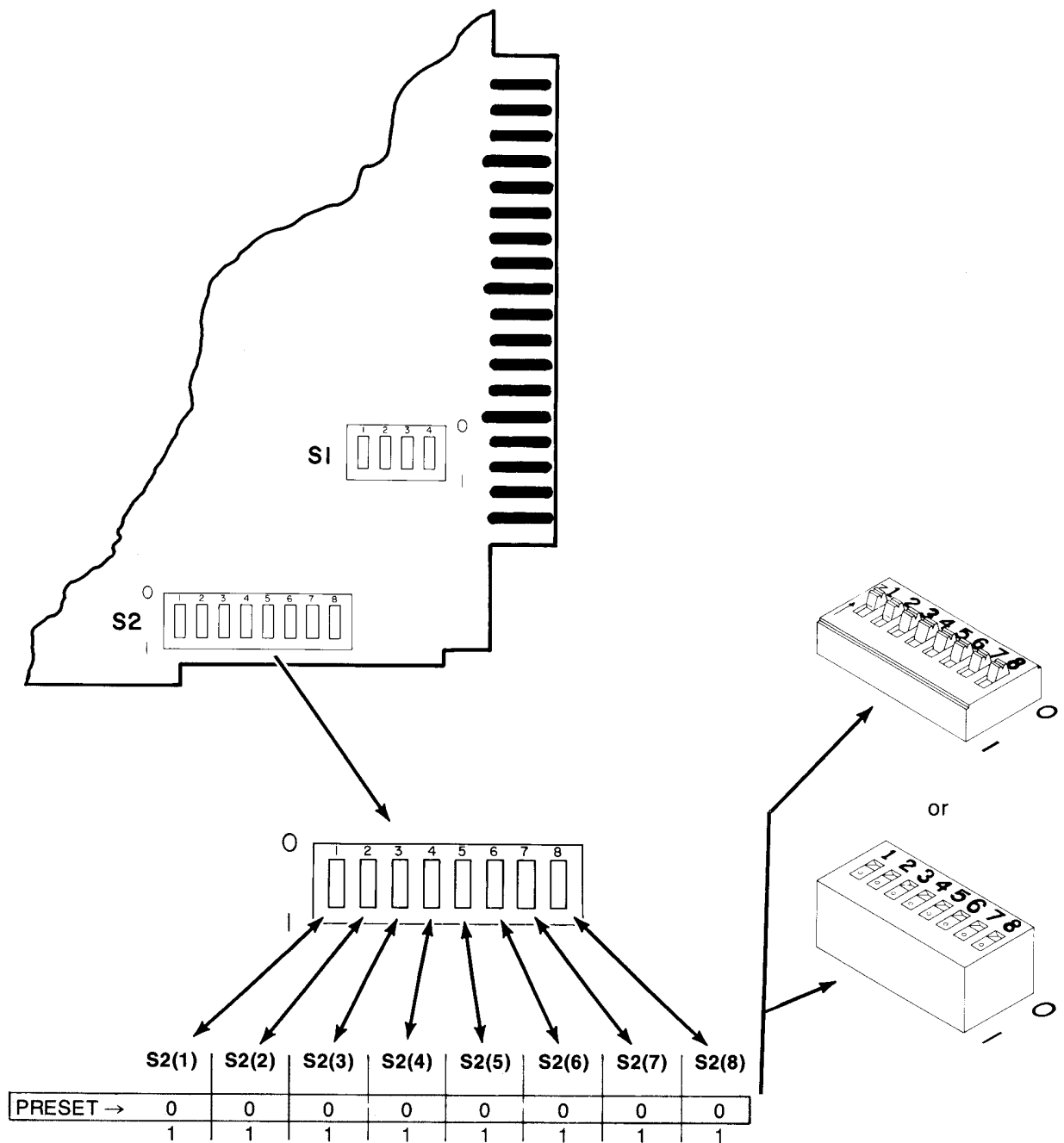
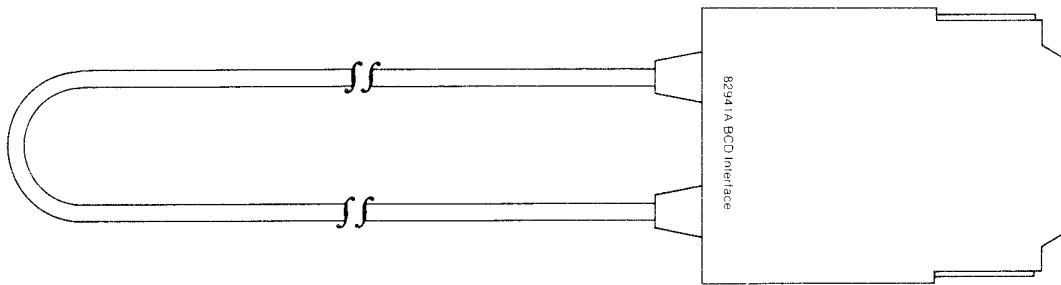


Figure 2-3. Default Switches

## Interface Cable

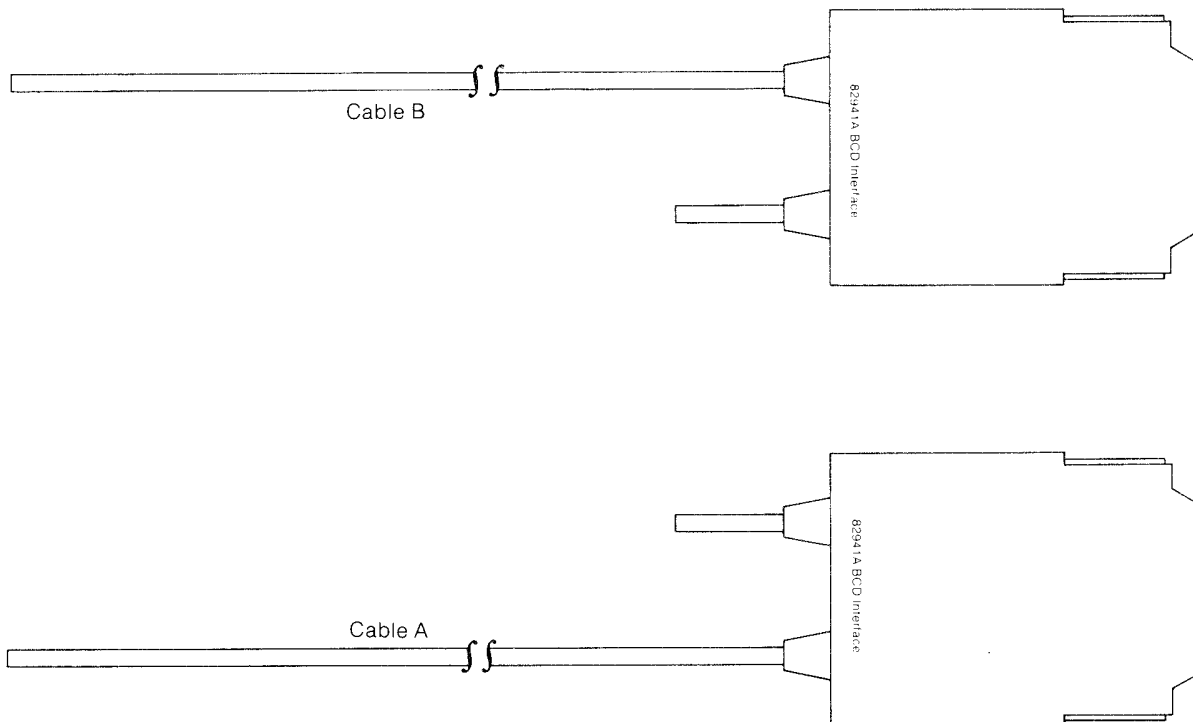
This discussion will assist you in cutting the interface cable to the proper length, identifying the cable contents and preparing the cable ends.

The interface is delivered with a single 31-conductor shielded cable, 4 meters long. Each end of the cable is terminated to the interface circuit board by plug-in connectors as illustrated in figure 2-4.



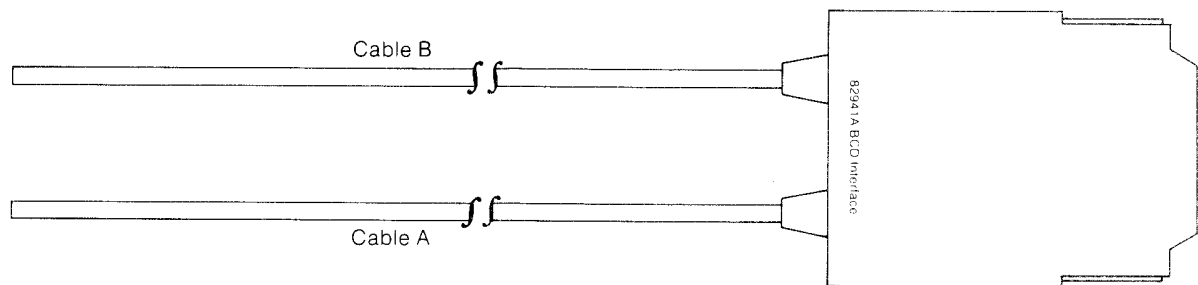
**Figure 2-4. Interface Cable Before Cutting**

The cable must be cut to interface a peripheral. If it is cut very near one end, you will have one cable approximately 4 meters long. This single cable could be cable A or cable B, depending upon which end is cut as illustrated in figure 2-5. Generally, a single cable interfaces one peripheral. However, it is possible to interface two peripherals with one cable or one peripheral with both cables.



**Figure 2-5. Single Cable Configurations**

If the cable is cut in the center, you will have two cables, each 2 meters long as illustrated in figure 2-6. In a 2-cable configuration, various lengths for each cable may be obtained by cutting the supplied cable at intermediate points between one end and its center. Two cables can interface two peripherals or one peripheral with extended digit capability.



**Figure 2-6. Two Cable Configuration**

If you require two cables and the length of each must be more than 2 meters, order another cable by specifying HP part number 8120-3192. Then, remove either end of the supplied cable by removing the cable clamp hardware and unplugging the cable connector from the circuit board. Plug the new cable onto the circuit board connector and secure with the cable clamp hardware. This will provide you with two cables that can be cut to a length of up to 4 meters each. Cables longer than 4 meters should not be used.

If two peripherals are to be interfaced and one requires more digit capability than one cable can provide, it is permissible to split one cable between two devices.

As you can see, there are several considerations that need to be made before you cut the cable. Once you decide where to cut it, use a pair of wire cutters large enough to accommodate the cable diameter.

#### **CAUTION**

If the cable is cut so that there is only one cable, the unused cable stub should be removed from the circuit board. If the cable stub is left connected and any of the wires become shorted together or come in contact with a conductive surface during I/O operations, equipment damage may result. After the interface is reassembled, use a piece of electrical tape to cover the hole resulting from the removal of the unused cable stub.

## Cable Structure/Default Formats

Let's now examine the contents of each cable. Wire colors of the various lines are given later.

Cable A contains the following BCD lines:

BCD Ports P0 — P4 (4 bits each)  
 BCD Port P10 (4 bits)  
 Sign Bits S1 and S2 ( $\frac{1}{2}$  of P11)  
 I/OA (1 bit)  
 CTLA (1 bit)  
 FLGA (1 bit)  
 GND (2 lines)  
 A and B Shield

Cable B contains the following BCD lines:

BCD Ports P5 — P9 (4 bits each)  
 BCD Port P10 (4 bits)  
 Sign Bits S3 and S4 ( $\frac{1}{2}$  of P11)  
 I/OB (1 bit)  
 CTLB (1 bit)  
 FLGB (1 bit)  
 GND  
 A and B Shield

Notice that ports P0 through P4 are in cable A while cable B continues with ports P5 through P9. Also notice that port P10, plus the GND and shield wires for both channels are in both cables. Let's look at this two-cable port structure more closely.

Ports P0 through P10 may be used to transfer mantissa, exponent or function digits that were listed in table 1-1. One or two peripherals may be connected to these ports. The number of ports used by a peripheral is determined by its functional capability and/or the amount of resolution needed. Thus, one peripheral may use one port or, by using both cables, all of the ports.

Switch segment 1 of switch S2 selects one of two default formats or the interface can be configured by software. If a switch settable default format is used, the various ports have dedicated assignments as shown in tables 2-1 and 2-2. However, by writing to the appropriate registers given in section 3, the port assignments can be changed in a variety of ways and thereby ignore the switch settings.

**Table 2-1. Standard Default Format (one peripheral)**

Number Input/Output Ports	Ports/ Bits Used	Digit Representation	Handshake Lines Used
8	P0 — P7	Mantissa	Channel A
1	P8	Exponent	Channel A
1	P9	Function	Channel A
$\frac{1}{2}$	P11 (S1, S2)	Signs	Channel A



**Note:** Channel A handshake lines include I/OA, CTLA and FLGA.

**Table 2-2. Option Default Format (two peripherals)**

Number of Input/Output Ports	Ports/Bits Used	Digit Representation	Handshake Lines Used
4	P0 — P3	Mantissa	Channel A
1	P4	Function	Channel A
4	P5 — P8	Mantissa	Channel B
1	P9	Function	Channel B
½	P11 (S1, S2)	Signs	Channel A
½	P11 (S3, S4)	Signs	Channel B

**Note:** Channel A handshake lines include I/OA, CTLA and FLGA; channel B handshake lines include I/OB, CTLB and FLGB.

Notice that the exponent field is not used in the option default format. However, it may be used if the interface is configured by software. In fact, you can specify as many as three exponent digits per channel with a software configuration. Also notice that neither of the switch settable default formats use port P10 directly. This port is also accessed by software and is useful as a control port.

Notice that the ports are assigned in consecutive order from the mantissa, exponent and function fields. For example, the standard default format uses ports P0 through P7 for the mantissa digits with P0 representing the most significant; port P8 is used for the exponent; port P9 is used for the function. If this format is used, all of these ports will have these dedicated assignments and be used in a transfer, even if nothing has been wired to a port. Therefore, if you don't need eight digit mantissa resolution or, perhaps exponent and/or function digits aren't needed at all, you will most likely want to re-configure the interface by software.

The option default format also uses dedicated port assignments as outlined in table 2-2. Notice that exponent digits are not used in this format. If this format is used, one peripheral is terminated to cable A and one peripheral is terminated to cable B. If you are going to interface two peripherals and this format will not fulfill your needs, again the interface can be configured by software.

Since the switch settable formats will not satisfy all installation requirements, let's briefly consider a software configuration. This will be discussed in more detail later. For now, let's consider the restrictions. Any configuration is legal if the following restrictions are observed:

1. The sum of all ports used must not exceed eleven.
2. The number of exponent digits per channel must not exceed three.
3. If the direction of a transfer is different between channels A and B, channel B may have only function digits.

If any of these restrictions are violated, an error message will be generated. These error messages are listed in Section 3.

**Note:** The ports are always assigned in consecutive order starting with channel A mantissa (P0) exponent A, function A, mantissa B, exponent B and function B. If a field isn't needed, it will be ignored if it isn't specified.

Now consider port P10, a special port different from the others. If port P10 is selected by software to be a part of a channel, it may be used to represent mantissa, exponent or function digits for either channel. If port P10 isn't selected to be a part of either channel, it may be used as an output-only control port independent of both channels.

Port P11 contains the sign bits which have dedicated assignments and cannot be changed by software. The assignments are as follows:

- S1 — Mantissa Sign, Channel A
- S2 — Exponent Sign, Channel A
- S3 — Mantissa Sign, Channel B
- S4 — Exponent Sign, Channel B

### Preparing Cable Ends/Wire Color Codes

At this point in the installation procedure the interface cable should have already been cut to meet the needs of your installation. This discussion will assist you in preparing the ends of the interface cable. The cable ends should be prepared before the interface is installed in the computer.

Since there are numerous installation considerations and a variety of BCD peripherals available, there cannot be a universal method for preparing the cable ends for all peripherals. There are, however, a few general procedures you can follow.

First, determine how much cable wires need to fan out from the main body of the cable(s) in order to connect your peripheral(s). This can vary considerably from one device to the next because of the spacing of the terminating locations on the peripheral. You should also keep in mind that you may be terminating one cable to two devices. Once you decide how much the cable wires need to fan out, refer to figure 2-7 and cut off the outer insulation of the cable to accommodate this. Then cut the shield off even with the outer insulation and place heat shrink tubing tape around the cable(s) so that none of the shield is exposed.

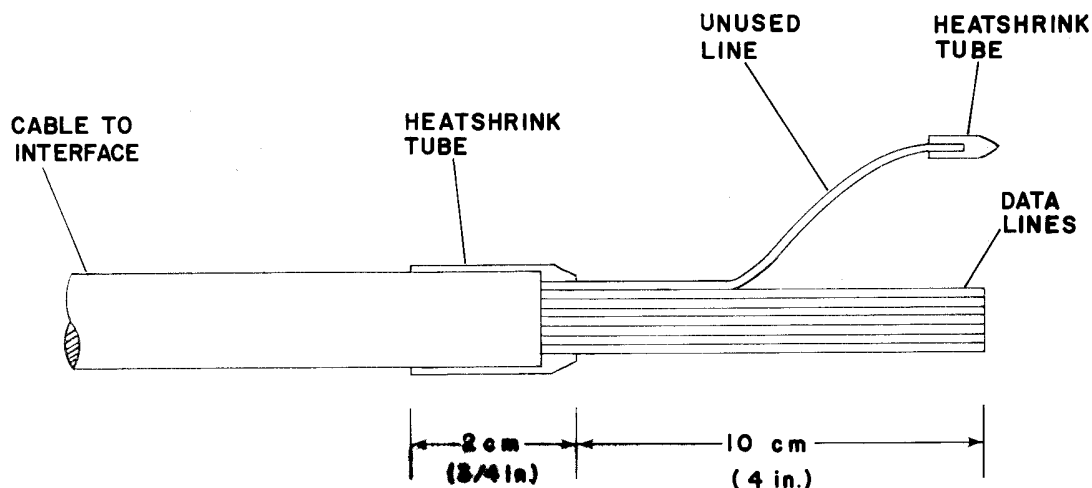
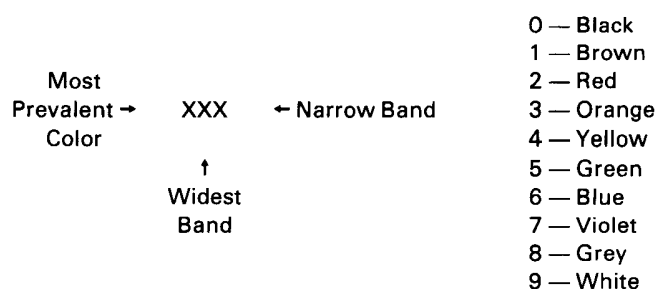


Figure 2-7. Cable Preparation

The color code for the cable wires is the same as the standard resistor color code. Digits have the following significance:



If only one number is given for the color code, the wire will be a solid color (i.e., a “1” specifies a solid brown wire). If two numbers are given, the wire will have two colors (i.e., “94” specifies the most prevalent color is white and has a yellow band). Three numbers indicate the wire has three colors (i.e., “902” specifies the most prevalent color is white, the widest band is black and the narrow band is red).

Refer to tables 2-3 and 2-4 and determine which BCD lines you intend to connect to your peripherals(s). It may also be helpful to fold out the schematics located in the back of this manual as you identify the cable wires needed for your installation. Remove only enough insulation from each wire you intend to use to accomplish your installation. Do not remove an excessive amount of insulation from the wires as they may short together or to other conductive surfaces. Some BCD peripherals may require that the cable wires be soldered to a circuit board. If this is the case, make sure not to remove so much insulation that the wires could protrude through the circuit board holes and short together or with other conductive surfaces.

After the insulation is removed from the cable wires you intend to use, proceed with installing the interface and connecting peripherals. **Do not connect any of the cable wires to a peripheral without the interface installed in an HP Series 80 Personal Computer port.**

Table 2-3. Cable A Wire Colors

## Handshake Lines and Sign Bits

Mnemonic	Wire Color
I/OA	912
CTLA	913
FLGA	914
S1 (Mantissa Sign, Channel A)	905
S2 (Exponent Sign, Channel A)	906
A and B GND	907, 915
A and B Shield (these should not be connected to the peripheral)	—

## Digit Ports

## Significance

## Wire Color Code

Port	Standard Default Format	Option Default Format	Wire Color Code			
			MSB (8)	(4)	(2)	LSB (1)
P0	Mantissa Digit 1 (Channel A)	Mantissa Digit 1 (Channel A)	4	3	2	1
P1	Mantissa Digit 2 (Channel A)	Mantissa Digit 2 (Channel A)	8	7	6	5
P2	Mantissa Digit 3 (Channel A)	Mantissa Digit 3 (Channel A)	92	91	90	9
P3	Mantissa Digit 4 (Channel A)	Mantissa Digit 4 (Channel A)	96	95	94	93
P4	Mantissa Digit 5 (Channel A)	Function Digit (Channel A)	0	908	98	97
P10	Special Control Port	Special Control Port	904	903	902	901

**Table 2-4. Cable B Wire Colors****Handshake Lines and Sign Bits**

Mnemonic	Wire Color
I/OB	912
CTLB	913
FLGB	914
S3 (Mantissa Sign, Channel B)	905
S4 (Exponent Sign, Channel B)	906
A and B GND	907, 915
A and B Shield (these should not be connected to the peripheral)	—

**Digit Ports**

Significance			Wire Color Code			
Port	Standard Default Format	Option Default Format	MSB (8)	(4)	(2)	LSB (1)
P5	Mantissa Digit 6 (Channel A)	Mantissa Digit 1 (Channel B)	4	3	2	1
P6	Mantissa Digit 7 (Channel A)	Mantissa Digit 2 (Channel B)	8	7	6	5
P7	Mantissa Digit 8 (Channel A)	Mantissa Digit 3 (Channel B)	92	91	90	9
P8	Exponent Digit (Channel A)	Mantissa Digit (Channel B)	96	95	94	93
P9	Function Digit (Channel A)	Function Digit (Channel B)	0	908	98	97
P10	Special Control Port	Special Control Port	904	903	902	901

**Handshake Line Mnemonic Variations**

Before proceeding with installing the interface or connecting a peripheral, let's look at where the I/O, CTL and FLG lines should be terminated on a BCD device. There is no simple answer to this since BCD instruments made by different manufacturers (and often different instruments made by the same manufacturer) give various names to these lines.

Most BCD devices manufactured by Hewlett-Packard call the CTL line *External Trigger* and the FLG response line *Print Command*. Other common names for the CTL line are:

- Trigger
- External Encode
- Sample

The FLG response line might be called:

- Print
- Print Enable
- Ready
- Data Flag

Often, the only way to know where to terminate these lines on the BCD instrument is to read the description of the lines in the operating manual for that instrument. In section 3 a handshake sequence is given where the CTL, FLG and I/O lines are discussed in detail. In this discussion, a brief description is adequate.

The CTL line initiates a transfer when it is true. So, as you examine the description of the lines in the BCD instrument's operating manual, look for key words that mean the same thing. For example, the key words in the description might typically be something like *initiates a measurement*. The description of the line may also tell you which logic sense and handshake configuration (trailing or leading edge) you need.

The FLG response line is a ready/busy indicator from the peripheral. It goes true at the beginning of a reading and returns false when the reading is complete. The description for the FLG line would most likely reference something similar to this and it may also reference a print command like *HIGH to LOW transition constitutes a print command*.

The I/O line is used by the interface to tell the BCD peripheral whether the interface is specifying an input or an output. When it is true, an output is specified; when it is false, an input is specified. Therefore, if the BCD instrument can't do both, the I/O line will not be required. If the peripheral can receive and send data, look for something in the line description that references I/O operations.

## Installing the Interface and Connecting Peripherals

This procedure should be read and understood before you install the interface or connect a peripheral to it.

### Safety Precautions

Manufacturers of peripheral devices often use different grounding techniques. In some instances, logic ground is allowed to float with respect to earth ground in an effort to reduce ground return interference with digital signals. This may cause a voltage level between the two grounds to be high enough to be hazardous. Therefore, care should be taken when you are installing the interface or when peripherals are being connected to or disconnected from the interface.

On the BCD Interface, the ground contact connects to the cable shield, and, when the interface is installed on the computer, the ground contact connects to earth ground and to logic ground. On the peripheral end of the cable, the shield is not terminated to any ground. By not connecting the cable shield on the peripheral end, two things are accomplished:

- The possibility of ground loop problems are greatly reduced.
- If the peripheral's ground is floating or defective, touching the ground contact when the interface is removed from an I/O slot cannot result in a shock hazard.

You should keep in mind, however, if the interface is removed from an I/O slot while it is connected to a peripheral, logic ground does appear on the edge connector of the interface. Unless you know the voltage level of logic ground with reference to earth ground, never touch the edge connector while the interface is terminated to a peripheral.

#### WARNING

To avoid personal injury and equipment damage, read and understand the preceding safety precautions and do not deviate from the order of the following steps to install the interface and peripheral(s). Before proceeding, make sure the cable shield has been insulated with heat shrink tubing or electrical tape.

1. Turn the computer power switch to the OFF position. However, make sure the computer is plugged into a grounded (3-wire) ac outlet.
2. Remove the protective covers from two of the I/O ports to accommodate the interface and ROM Drawer. Any remaining unused I/O ports should be kept covered.
3. Refer to figure 2-8 to install the interface. Notice that the interface is keyed to prevent it from being inserted upside-down. In a like manner, install the ROM Drawer into an I/O port making sure that it contains the I/O ROM module.



**Figure 2-8. Installing the Interface**

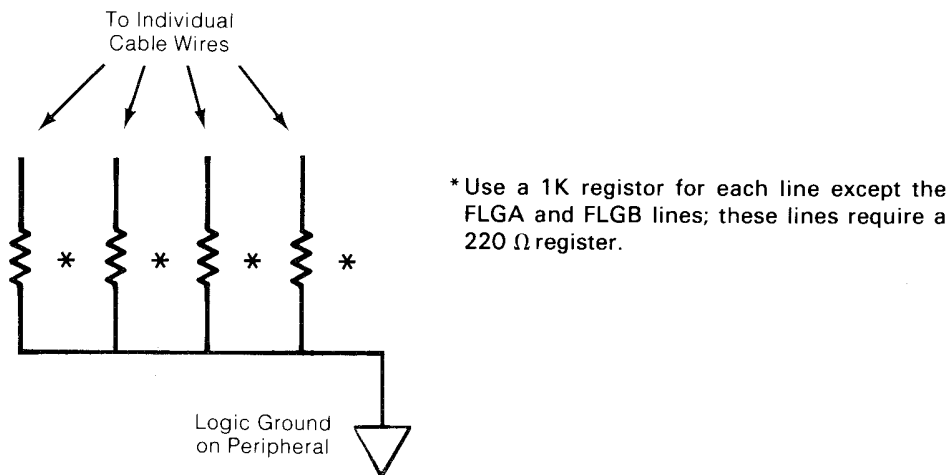
4. Turn the power switch of all peripherals to be connected to the OFF position.
5. Refer back to tables 2-3 and 2-4 and connect the appropriate cable wires to your peripheral(s). If you are using a two-cable configuration, make sure to identify the A and B cables, as the same wire colors appear in both cables.
6. If there are unused wires after your peripherals are connected, you may use either of two options:
  - Leave each wire unterminated, in which case the line will float high.
  - Terminate each wire individually through a 1k $\Omega$  resistor to logic ground on the peripheral. This method is useful for a partially used port where unused bits must be tied low.

Let's consider each option individually.

In most instances, leaving unused lines floating is acceptable. If you choose to do this, each unused wire should be individually sleeved with heat shrink tubing to prevent them from coming in contact with each other or another conductive surface. The major factor determining whether or not you should use this option is the software used by your system. Keep in mind that if you leave a port floating high, it will be interpreted as a question mark or as a "0", depending upon which logic sense is configured (see table 1-1). Normally, however, an unused port would never be read or written to. If an unterminated port is inadvertently written to, it will have no adverse affect on the interface.



Terminating each wire individually through a separate resistor to the peripheral's logic ground is considerably more cumbersome. For example, if you have one port to terminate to logic ground, you will need four resistors, one for each wire. This is illustrated in figure 2-8. If you use this option, each wire terminated to logic ground, regardless of how many there are, must have a separate resistor. Never place any of the wires together and then connect them to the peripheral's logic ground through a common resistor. Again, software is the major determining factor for using this option. If this option is used, care should be taken to never write to a port terminated to ground. This could cause equipment damage. It is also important that you properly insulate the side of each resistor connected to a wire to prevent a short from occurring.



**Figure 2-9. Terminating Unused Lines to Logic Ground**

7. Turn the power switches of the computer and peripheral(s) to the ON position.

## Removing Peripherals/Disconnecting the Interface

Use the following steps in the order given to disconnect peripherals from the interface or to disconnect the interface from the computer.

### CAUTION

Do not remove the interface from an I/O port with the computer power switch on. Doing this will cause damage to either the interface, the computer or both.

1. Turn all peripheral power switches off.
2. Turn the computer power switch off.
3. Remove the interface cable wires from the peripheral(s) you are disconnecting from the interface. If there are two peripherals and only one is to be disconnected, each wire removed must be individually sleeved or terminated to the peripheral's logic ground before power is returned to the computer. This was explained in step 6 of the installation procedure. If you intend to remove the interface from the I/O slot, proceed with step 4.
4. If the interface is removed from the I/O port while it is terminated to a peripheral, care should be taken not to touch the edge connector of the interface. Logic ground of the peripheral appears on this connector. If this ground is floating or defective, touching the edge connector may expose you to an electrical shock hazard.

## Using Your BCD Interface

### Introduction

The HP 82941A BCD Interface enables your HP Series 80 Personal Computer to communicate with a variety of instruments that present data in Binary Coded Decimal (BCD) format. This section explains how to program the interface.

### Binary Coded Decimal

*Binary Coded Decimal* is, as the name implies, a method of encoding the decimal digits (0 through 9) in a four bit binary format. The following table shows the binary coding of the ten decimal digits and six additional ASCII characters that are allowed. The interface can be programmed through default switches or program statements to recognize either positive-true or negative-true logic. The table shows the allowable ASCII characters and the associated BCD codes for each logic sense.

**Table 3-1. BCD Coding**

ASCII	Positive True	Negative True
0	0000	1111
1	0001	1110
2	0010	1101
3	0011	1100
4	0100	1011
5	0101	1010
6	0110	1001
7	0111	1000
8	1000	0111
9	1001	0110
:	1010	0101
;	1011	0100
<	1100	0011
=	1101	0010
>	1110	0001
?	1111	0000

### Data and Handshake Lines

Data and handshake lines are provided as follows:

- Data - 44 bi-directional data lines organized as 11 four-bit ports (P0 through P10).
- Sign Bits - four sign bits (S1 through S4) through port P11.
- Handshake - four output handshake lines (I/OA, I/OB, CTLA, CTLB).

## Operating Modes

The operating mode refers to selecting the fields as shown in the next table.

**Table 3-2. Operating Mode**

Field	Range
Number of channels	A, B, or both.
Channel Direction	Input or output.
Number of mantissa digits in each channel*	0 through 11.
Number of exponent digits for each channel*	0 through 3.
Number of function digits in each channel*	0 through 11.
Decimal point placement for each mantissa	May be placed before any mantissa digit.
Logic sense of all signals	Positive or negative true.
Handshake triggering	Leading or trailing edge.

## Default Formats

The BCD interface provides two standard default formats: single and dual channel. These formats are selected by changing the reset switch (S2) switch 1 or by program control. The default formats and their associated port assignments are shown in the next tables.

**Table 3-3. Single Channel Format (S2 switch 1 = "0" position)**

Number of Ports	Ports (or bits) Used	Data	Handshake
8	P0 through P7	Mantissa	Channel A
1	P8	Exponent	Channel A
1	P9	Function	Channel A
1/2	P11 (bits S1 and S2)	Sign Bits	Channel A

**Table 3-4. Dual Channel Format (S2 switch 1 = "1" position)**

Number of Ports	Ports (or bits) Used	Data	Handshake
4	P0 through P3	Mantissa	Channel A
1	P4	Function	Channel A
4	P5 through P8	Mantissa	Channel B
1	P9	Function	Channel B
1/2	P11 (bits S1 and S2)	Sign Bits	Channel A
1/2	P11 (bits S3 and S4)	Sign Bits	Channel B

Examples are provided following the discussion of the interface registers to show how to select these and other formats with program statements.

\* The combined sum of mantissa, function, and exponent digits specified cannot exceed 11.

## Data Rates

The data transfer rate depends upon the direction of the transfer, the type of transfer (normal, fast handshake, or interrupt), concurrent operations taking place and peripheral timing. The maximum data rates that the interface can handle are as follows:

**Table 3-5. Transfer Rates**

Transfer Type	Input	Output
Fast Handshake	20k bytes/sec	22 bytes/sec
Normal ENTER	7.75k bytes/sec	4.75k bytes/sec
Interrupt	400 bytes/sec	400 bytes/sec

The data rates shown for the fast handshake mode are achieved by placing the following restrictions on the fast handshake mode:

1. Standard Format
  - Channel A only (one peripheral).
  - Eight mantissa digits only.
  - One exponent digit only.
  - Function digit may be selected but is ignored.
  - Mantissa and exponent signs will be transferred.
2. Byte count must be specified on an input.
3. Positive true logic on data lines.
4. Trailing edge handshake only.
5. Decimal points may be selected but will not be transferred.
6. ASCII codes 0 through 32 and 44 (comma) cannot be used for output.

If the peripheral fails to complete a handshake during the fast handshake mode, the CPU in the computer halts and the system hangs up. To recover, the FLGB line, unused in the fast handshake mode format, may be used to indicate errors by the peripheral. If FLGB goes true while the interface is waiting for a handshake, the fast handshake mode is terminated and Error 115 is displayed.

## Program Statements

The following program statements are implemented by the BCD interface card. See the syntax reference in the *I/O ROM Owner's Manual* for a complete description of these statements.

ABORTIO*	ENABLE INTR	REMOTE	SEND TALK
ASSERT*	ENTER	RESET	SEND LISTEN
HALT*	OUTPUT	SEND CMD	STATUS*
CONTROL		SEND DATA	TRANSFER

## Using the Interface

This discussion assumes that you have read the *I/O ROM Owner's Manual*.

### Programming With Default Formats

This discussion shows how to program the interface with the single or dual channel default formats that are provided. Factory settings are assumed for the interface select code and the default switches. The factory settings are:

- Interface Select Code (S1) = 3 (all set to the "0" position).
- Reset Default Switches (S2) = all set to the "0" position.

#### Example 1

Run the following program:

```
10 RESET 3 ! RESET INTERFACE
20 STATUS 3,0 ; R0,R1,R2,C3,C4,C5,C6,C7,C8,C9,C0
30 IMAGE K,X,00, / ! PRINT FORMAT
40 PRINT USING 30 ; "READ REG 0=";R0,"READ REG 1=";R1
50 PRINT USING 30 ; "READ REG 2=";R2
60 PRINT USING 30 ; "REG 3=";C3,"REG 4=";C4,"REG 5=";C5
70 PRINT USING 30 ; "REG 6=";C6,"REG 7=";C7,"REG 8=";C8
80 PRINT USING 30 ; "REG 9=";C9,"REG 10=";C0
90 END
```

Here is the printout:

READ REG 0= 3	—Interface I.D.
READ REG 1= 0	—Interrupt Mask
READ REG 2= 0	—Handshake Lines
REG 3= 8	—Mantissa Digits Specified
REG 4= 1	—Exponent Digits Specified
REG 5= 1	—Function Digits Specified
REG 6= 0	—Decimal Point Location
REG 7= 0	—Handshake Logic Sense
REG 8= 0	—Data Logic Sense
REG 9= 0	—Function Logic Sense
REG 10= 0	—Signs and Port 10 Logic Sense

Note that Read Register 0 and Registers 3, 4, and 5 are not zero. Read Register 0 identifies the interface as the BCD interface. Register 3 indicates the number of mantissa digits to be input (in this case eight). Register 4 indicates the number of exponent digits (one) to be input and Register 5 indicates the number of function digits (one) to be input. Registers 7 through 10 indicate the logic sense of the various data and handshake lines (the value of 0 indicates positive true logic sense). Read registers 1 and 2 and Register 6 are not used for this example and are discussed later in this section.

---

\* Executing this statement interrupts any I/O operation that may be in progress.



**Example 2**

Assume that you are connected to a digital multimeter. The multimeter outputs eight significant data digits, a one-digit exponent, and a one-digit function value. All logic senses (data, handshake, etc.) are defined as positive true.

The following program reads the data from the multimeter and prints the results:

```
10 RESET 3 ! RESET INTERFACE
20 ENTER 3 ; D1,F1 ! GET CHANNEL A DATA
30 PRINT "READING=";D1
40 PRINT "FUNCTION=";F1
50 STOP
```

Here is a typical printout:

```
READING= 1250.524
FUNCTION= 1
```

**Example 3**

The function value is used to indicate the position of the function switch (ohms, volts, amps, etc.) located on the multimeter. Let the function values be defined as:

```
1 = Ohms
2 = Volts AC
3 = Amps
4 = Volts DC
```

Run the following program:

```
10 RESET 3 ! RESET INTERFACE
20 ENTER 3 ; D1,F1 ! GET CHANNEL A DATA
30 ON F1 GOSUB 70,90,110,130 ! CHANNEL A BRANCHES
40 PRINT "READING=";D1;Z$
50 PRINT "FUNCTION=";F1
60 STOP
70 Z$="OHMS" ! FUNCTION =1
80 RETURN
90 Z$="VOLTS-AC" ! FUNCTION =2
100 RETURN
110 Z$="AMPS" ! FUNCTION =3
120 RETURN
130 Z$="VOLTS-DC" ! FUNCTION =4
140 RETURN
```

Here are typical results:

```
READING= 1250.524 OHMS
FUNCTION= 1
```

**Example 4**

The BCD interface can be connected to two BCD devices simultaneously. Set the reset default switch (S2, switch 1) to the "1" position. This changes the default mode of the interface to two channel operation. Run the following program:

```

10 RESET 3 ! RESET INTERFACE
20 STATUS 3,0 ; R0,R1,R2,C3,C4,C5,C6,C7,C8,C9,C0
30 IMAGE K,X,DD,/ ! PRINT FORMAT
40 PRINT USING 30 ; "READ REG 0=";R0,"READ REG 1=";R1
50 PRINT USING 30 ; "READ REG 2=";R2
60 PRINT USING 30 ; "REG 3=";C3,"REG 4=";C4,"REG 5=";C5
70 PRINT USING 30 ; "REG 6=";C6,"REG 7=";C7,"REG 8=";C8
80 PRINT USING 30 ; "REG 9=";C9,"REG 10=";C0
90 END

```

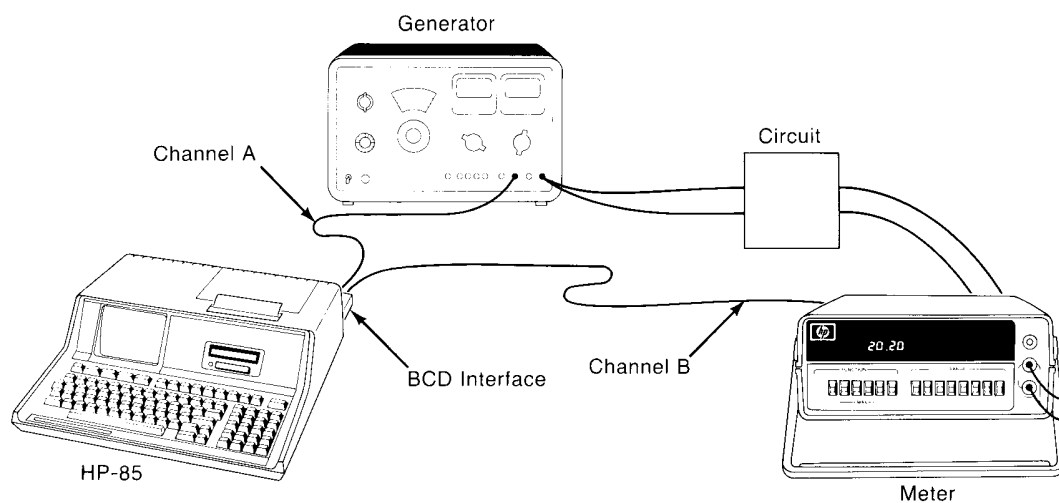
Here are the results:

READ REG 0= 3	—Interface I.D.
READ REG 1= 0	—Interrupt Mask
READ REG 2= 0	—Handshake Lines
REG 3= 68	—Mantissa Digits Specified
REG 4= 0	—Exponent Digits Specified
REG 5= 17	—Function Digits Specified
REG 6= 0	—Decimal Point Location
REG 7= 0	—Handshake Logic Sense
REG 8= 0	—Data Logic Sense
REG 9= 0	—Function Logic Sense
REG 10= 0	—Signs and Port 10 Logic Sense

Note that registers 3, 4, and 5 values are different than those obtained for example 1.

The value (68) for Register 3 indicates that four mantissa digits are reserved for channel A input and four mantissa digits are reserved for channel B input. The value (0) for Register 4 indicates that neither channel is using an exponent digit. The value (17) for Register 5 indicates that each channel has one digit reserved for input function data.

For example, the interface might be connected to a signal generator and a digital voltmeter:



**Figure 3-1. Typical Test System Connection**

**Example 5**

The signal generator is driving a circuit under test for frequency response. The voltmeter takes readings of the circuit output. By comparing the voltage vs. frequency readings, the bandpass of the circuit is determined.

Here is an example program. Typical frequency and voltage parameters are used.

```

10 ENTER 3 ; A1,F1,B2,F2 ! GET CHANNEL A AND
   B DATA
20 DISP A1;F1;B2;F2 ! DISPLAY DATA
30 ON F1 GOSUB 80,120,160 ! CHANNEL A
   FUNCTION BRANCHES
40 ON F2 GOSUB 100,140,180 ! CHANNEL B
   FUNCTION BRANCHES
50 PRINT A1;Z$
60 PRINT B2;Y$
70 STOP
80 Z$="Hz" ! A FUNCTION =1
90 RETURN
100 Y$="Microvolts" ! B FUNCTION =1
110 RETURN
120 Z$="KHz" ! A FUNCTION =2
130 RETURN
140 Y$="Millivolts" ! B FUNCTION =2
150 RETURN
160 Z$="MHz" ! A FUNCTION =3
170 RETURN
180 Y$="Volts" ! B FUNCTION =3
190 RETURN

```

Run the program. Here is a typical result:

```

1000 Hz
2020 Microvolts

```

**Partial Fields**

The examples shown thus far enter the input data from the interface in the following order: mantissa A, function A, mantissa B, function B. You cannot enter a portion (or field) of data (e.g., mantissa B) without first entering all preceding data. Partial field addressing is provided to enable you to access only the data that you desire. Partial field addressing is accomplished by specifying primary addresses 00 through 06. The partial field specifiers and the data that they access are shown in the next table. Interface select code 3 is assumed.

**Table 3-6. Partial Field Specification**

Device Selector	Data Entered
300	All data (Default)
301	Channel A mantissa, exponent, and function.
302	Channel B mantissa, exponent, and function.
303	Channel A mantissa and exponent.
304	Channel B mantissa and exponent.
305	Channel A function.
306	Channel B function.

**Example 6**

Run the previous program (example 5) and observe the results. Execute the following statements and observe the results shown here.

```

ENTER 300;A1,F1,B2,F2      1000  1  2020  1
PRINT A1;F1;B2;F2

ENTER 301;A1,F1            1000  1
PRINT A1;F1

ENTER 302;B2,F2            2020  1
PRINT B2;F2

ENTER 303;A1                1000
PRINT A1

ENTER 304;B2                2020
PRINT B2

ENTER 305;F1                1
PRINT F1

ENTER 306;F2                1
PRINT F2

```

Once a partial field has been specified, that field remains in effect until a new field is selected or the interface is reset.

The REMOTE statement can also be used to set a partial field specifier provided the address ranges from 00 to 06. Thus the sequence—

```

REMOTE 306
ENTER 3; F2
PRINT F2

```

could also be used to enter and print the channel B function data. This would be especially useful when documenting or debugging a program since the REMOTE statements stand out amongst the ENTER and OUTPUT statements.

Partial fields are addressed by ENTER, OUTPUT, SEND TALK, SEND LISTEN and SEND CMD statements.

## Port 10

The previous examples show how to take readings from external devices. The operator is required to manually set the instrument functions (frequency, function, etc.). The BCD interface provides a special port to allow control of external instruments from the program. This discussion explains how to use port 10 for output.

In the two channel example (example 5), the BCD interface ports are dedicated as follows:

- Ports P0 through P3 - Signal generator frequency reading.
- Port P4 - Signal generator function (range switch).
- Ports P5 through P8 - Voltmeter reading.
- Port P9 - Voltmeter function (voltage range switch).
- Port 10 - Unused.

Port 10 is a special purpose port that can be used as an input or output port. Port 10 is the only port that can be used as an output port without setting the reset default switch S2(8) to the output enable position. Port 10 is accessed by performing a control operation to the four least significant bits of Write Register 2. For example: `CONTROL 3,2;15` sets all four bits of port 10. The next example assumes the same instrument configuration and function values as those used for example 5.

The function values returned are the same as defined for that example.

The following assumptions about the signal generator are made:

- The generator sweep values range from 1 through 1000.
- A bit can be set to begin the sweep at 1.
- The generator has three ranges.
- Each range can be externally selected.

Here is the bit mask for the generator:

Bit 3	Bit 2	Bit 1	Bit 0
1 = MHz Range	1 = KHz Range	1 = Hz Range	1 = Start Sweep
Value = 8	Value = 4	Value = 2	Value = 1

### Example 7

The program initially sets the generator to the 1 Hz range and starts the sweep. When the sweep reaches 1000 Hz, the program switches the generator to the KHz range and restarts the sweep. When the sweep reaches 1000 KHz, the program switches the generator to the MHz range and again restarts the sweep. When the sweep reaches 1000 MHz, the entire program is restarted. Here is the program:

```

10 CONTROL 3,2 ; 3 ! SET PORT 10 OUTPUT VALUE
   TO 3
20 ENTER 3 ; A1,F1,B2,F2 ! ENTER CHANNEL A
   AND B DATA
30 IF A>999 THEN GOTO 230 ! CHECK TO CHANGE
   FREQ
40 DISP A1;F1;B2;F2
50 ON F1 GOSUB 110,150,190 ! CHANNEL A
   FUNCTION BRANCHES
60 ON F2 GOSUB 130,170,210 ! CHANNEL B
   FUNCTION BRANCHES
70 DISP A1;Z$
80 DISP B2;Y$
90 GOTO 20
100 END
110 Z$="Hz" ! A FUNCTION =1
120 RETURN
130 Y$="Microvolts" ! B FUNCTION =1
140 RETURN
150 Z$="KHz" ! A FUNCTION =2
160 RETURN
170 Y$="Millivolts" ! B FUNCTION =2
180 RETURN
190 Z$="MHz" ! A FUNCTION =3
200 RETURN
210 Y$="Volts" ! B FUNCTION =3
220 RETURN
230 IF F1<3 THEN 250 ! CHECK FOR MAX FREQ RANGE
240 GOTO 10
250 CONTROL 3,2 ; 2^(F1+1)+1 ! CHANGE GENERATOR FUNCTION
260 GOTO 20

```

## Interrupts

The BCD interface provides program interrupt. Interrupts are generated only from the most-significant digit of the function. If you are only inputting one digit of function information (see the previous example), then this digit generates the interrupt. You enable the interrupt mask to Write Register 1 with an ENABLE INTR and specify a BASIC service routine with an ON INTR statement.

### Example 8

Let's change the previous example to generate an interrupt if the voltmeter detects an overrange condition. Assume that the voltmeter returns a function value of 8 when an overrange is detected. Add the following program lines to the program used for the previous example:

```

1 ON INTR 3 GOSUB 300
2 ENABLE INTR 3;128 ! INTERRUPT FOR BIT 3 OF FUNCTION B

```

### Existing Program

```

300 STATUS 3,1 ; S1
310 IF S1<128 THEN ENABLE INTR 3;128 @ RETURN ! INT.CAUSE
320 DISP "METER OVERRANGE"
330 STOP

```



Line 1 - defines the program branch.

Line 2 - defines the bit in the function digit that can cause an interrupt (see Write Register 1) and enables the interrupt.

Line 300 - enters the value of read register 1 into variable S1. The status of Read Register 1 must ALWAYS be checked following an interrupt. Subsequent interrupts are prevented until this interrupt is serviced.

Line 310 - checks to see which bit of the function digit caused the interrupt. This check is used to prioritize the interrupts when more than one bit generates the interrupt (see multiple interrupts). This line also re-enables the interrupt with a new control word (128) to Write Register 1 if the function A value returned is less than 128 (bit 7 not set). Notice that the RETURN statement is placed on the same line as the ENABLE INTR statement.

### Multiple Interrupts

More than 1 bit of the function value can be specified to generate an interrupt. This feature allows you to define the priorities of the interrupts. Here is an example routine to service multiple interrupts. Note that the bit 3 routine is serviced prior to the bit 0 routine if they both occur.

#### Example 9

```
10 ON INTR 3 GOSUB 200
20 ENABLE INTR 3;9 ! INTERRUPT BITS
```

#### Main Program

```
200 STATUS 3,1 ; S1
210 IF BIT(S1,3) THEN 240
220 PRINT "BIT 0 DETECTED"
230 ENABLE INTR 3;9 @ RETURN
240 PRINT "BIT 3 DETECTED"
250 IF BIT(S1,0) THEN 220
260 ENABLE INTR 3;9 @ RETURN
```

Line 20 - specifies bits 0 and 3 to generate an interrupt.

Line 200 - enters the status byte from Read Register 1 (interrupt cause).

Line 210 - checks for a bit 3 interrupt (first priority). If bit 3 is true (set), the program branches to the bit 3 service routine (lines 240 through 260).

Line 220 - is the bit 0 service routine.

Line 230 - re-enables the interrupt and returns to the main program.

Line 240 - is the bit 3 service routine.

Line 250 - checks for a bit 0 interrupt (second priority). A program branch to the bit 0 routine (line 220) occurs only if bit 0 is true (set).

Line 260 - re-enables the interrupt and returns to the main program.

You determine the priorities for the interrupts. The bit 0 interrupt can be defined as having the higher priority by exchanging the arguments of the bit functions (lines 210 and 250).

## Registers

The interface contains 13 registers that can be accessed from the HP Series 80 Personal Computers. These registers are divided into three groups: read registers, write registers, and bi-directional registers. Read registers are accessed with the `STATUS` statement. Write registers are accessed with the `CONTROL` statement. The bi-directional registers are accessed with either the `STATUS` (input) statement or the `CONTROL` (output) statement. Other statements (`ASSERT`, `ENABLE INTR`) access specific registers only and are explained with the register description. This discussion explains each register and shows the access procedure.

**Note:** The default values shown for all registers assume that the reset default switch (S2) is set to the factory setting (all "0"s). Positive-true logic sense is assumed for all descriptions.

### Read Register 0

**Interface ID**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Always 0	Always 0	Always 0	Always 0	Always 0	Always 0	Always 1	Always 1
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,0;S0
```

The value returned (preset to 3) from this register is entered into variable S0. The value 3 indicates that this is the BCD interface.

### Read Register 1

**Interrupt Cause**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Generated from Function B (Most Significant Digit)				Generated from Function A (Most Significant Digit)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,1;S1
```

The value returned indicates the bit (or bits) that generated the interrupt. See Write Register 1 for a complete discussion about the BCD card interrupts.

Default value: 0

## Read Register 2

### Control Line Messages

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I/O A 0 = Input 1 = Output	I/O B 0 = Input 1 = Output	Cntrl A 0 = Ready 1 = Busy	Cntrl B 0 = Ready 1 = Busy	Flag A 0 = Ready 1 = Busy	Flag B 0 = Ready 1 = Busy	0	0
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

```
STATUS 3,2;S2
```

The value returned indicates the status of the various handshake signals.

Default value: 0

The meanings of the various bits are described next.

- Bit 7 set (1) indicates that channel A ports are enabled for output operation. Bit 7 reset (0) indicates that channel A ports are set to the input mode.
- Bit 6 set (1) indicates that channel B ports are enabled for output operation. Bit 6 reset (0) indicates that channel B ports are set to the input mode.
- Bit 5 set (1) indicates that the channel A "Control" handshake line is active (busy). Bit 5 reset (0) indicates that the channel A "Control" handshake line is ready.
- Bit 4 set (1) indicates that the channel B "Control" handshake line is active (busy). Bit 4 reset (0) indicates that the channel B "Control" handshake line is ready.
- Bit 3 set (1) indicates that the channel A "Flag" handshake line is busy. The peripheral device connected to channel A uses the "Flag" line to indicate its status to the computer. Bit 3 reset (0) indicates that the channel A peripheral is ready.
- Bit 2 set (1) indicates that the channel B "Flag" handshake line is busy. The peripheral device connected to channel B uses the "Flag" line to indicate its status to the computer. Bit 2 reset (0) indicates that the channel B peripheral is ready.
- Bits 1 and 0 are not used.

## Write Register 0

Write Register 0 is not implemented. Attempting to access Write Register 0 generates error 111.

## Write Register 1

**Interrupt Mask**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Function B (Most Significant Digit)				Function A (Most Significant Digit)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
ENABLE INTR 3; <mask value>
CONTROL 3, 1; <mask value>
```

Write Register 1 contains the interrupt mask for channel A and channel B. The interrupts are generated from the most significant digit of the function value of each channel. For example, assume that channel A is configured for two digit function values from an external device. These values range from 00 through 99. You can specify an interrupt only for values of the tens digit. The units digit cannot generate an interrupt. If a single digit function value is specified (default) then that single digit is the most significant digit and can generate an interrupt (see example 8).

## Write Register 2

**Handshake Lines and Port 10**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I/O Channel A	I/O Channel B	CTL Channel A	CTL Channel B	Port 10 Output (when available)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
CONTROL 3, 2; <value>
ASSERT 3; <value>
```

Write Register 2 enables you to control the interface handshake lines from a program. The output data for port 10 (if port 10 is available for output) is also written to this register.

The meanings of the various bits are described next.

- Bit 7 set (1) enables channel A ports for output. The reset default switch (S2) switch 8 must be set to the “1” position. Attempting to set bit 7 with S2 switch 8 in the “0” position (default) generates error 113. Bit 7 clear (0) enables channel A ports for input.
- Bit 6 set (1) enables channel B ports for output. The reset default switch (S2) switch 8 must be set to the “1” position. Attempting to set bit 6 with S2 switch 8 in the “0” position (default) generates error 113. Bit 6 clear (0) enables channel B ports for input.
- Bit 5 enables you to control the CTL handshake line for channel A. This line remains active until the interface is reset or bit 5 is cleared (set to 0). If this bit is not reset, subsequent ENTER or OUTPUT operations generate error 118.
- Bit 4 enables you to control the CTL handshake line for channel B. This line remains active until the interface is reset or bit 4 is cleared (set to 0). If this bit is not reset, subsequent ENTER or OUTPUT operations generate error 118.
- Bits 3 through 0 are port 10. When port 10 is used as an output port, the data is written to these four bits (see example 7). The input/output switch (S2) switch 8 does not affect port 10. When port 10 is enabled as an input port or configured as part of a channel, any attempt to output to port 10 generates error 114.

### Register 3

**Mantissa Digits**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Number of Digits Assigned for Channel B Mantissa (0 — 11)				Number of Digits Assigned for Channel A Mantissa (0 — 11)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,3;S3
CONTROL 3,3;<value>
```

The value contained in this register indicates how many mantissa digits are specified for each channel. To change this specification, execute the appropriate CONTROL statement to this register. The new specification remains in this register until changed by another CONTROL statement or until the interface is reset.

Reset default: 8 (Channel A only)

**Note:** The reset default value for two channel operation is 68. This indicates that four mantissa digits are specified for each channel (see example 4).

## Register 4

**Exponent Digits**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Number of Digits Assigned for Channel B Exponent (0 — 3)				Number of Digits Assigned for Channel A Exponent (0 — 3)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,4;S4
CONTROL 3,4; <value>
```

The value contained in this register indicates how many exponent digits are specified for each channel. To change this specification, execute the appropriate `CONTROL` statement to this register. The new specification remains in this register until changed by another `CONTROL` statement or until the interface is reset.

Reset default: 1 (Channel A only)

**Note:** The reset default value for two channel operation is 0. This indicates that no exponent digits are specified for either channel.

## Register 5

**Function Digits**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Number of Digits Assigned for Channel B Function (0 — 11)				Number of Digits Assigned for Channel A Function (0 — 11)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,5;S5
CONTROL 3,5; <value>
```

The value contained in this register indicates how many function digits are specified for each channel. To change this specification, execute the appropriate `CONTROL` statement to this register. The new specification remains in this register until changed by another `CONTROL` statement or until the interface is reset.

Reset default: 1 (Channel A only)

**Note:** The reset default value for two channel operation is 17. This indicates that one function digit is specified for each channel.

## Register 6

**Decimal Point Placement**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Number of Mantissa Digits Assigned to the Right of the Decimal Point. (Channel B)				Number of Mantissa Digits Assigned to the Right of the Decimal Point. (Channel A)			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,6;S6
CONTROL 3,6; <value>
```

The value contained in this register indicates how many mantissa digits to the right of the decimal point are specified. To change this specification, execute the appropriate **CONTROL** statement to this register. The new specification remains in this register until changed by another **CONTROL** statement or until the interface is reset. Note that the specification cannot exceed the number of mantissa digits specified for that channel. For example, if eight mantissa digits are specified for channel A, then a maximum of eight digits can be specified to the right of the decimal point for this channel. Decimal point placement can only be specified for input operations.

Reset default: 0

## Register 7

**Control Sense and Handshake Mode**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Logic Sense I/O A	Logic Sense I/O B	Logic Sense CTL A	Logic Sense CTL B	Logic Sense Flag A	Logic Sense Flag B	Handshake Mode A	Handshake Mode B
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,7;S7
CONTROL 3,7; <value>
```

The value contained in this register indicates the logic sense of the control and handshake lines. To change this specification, execute the appropriate `CONTROL` statement to this register. The new specification remains in this register until changed by another `CONTROL` statement or until the interface is reset.

Reset default: 0

The meanings of the various bits are described next.

- Bit 7 set (1) indicates that the logic sense for the channel A I/O control line is negative-true. Bit 7 reset (0) indicates positive-true logic sense.
- Bit 6 set (1) indicates that the logic sense for the channel B I/O control line is negative-true. Bit 6 reset (0) indicates positive-true logic sense.
- Bit 5 set (1) indicates that the logic sense for the channel A CTL handshake line is negative-true. Bit 5 reset (0) indicates positive-true logic sense.
- Bit 4 set (1) indicates that the logic sense for the channel A CTL handshake line is negative-true. Bit 4 reset (0) indicates positive-true sense.
- Bit 3 set (1) indicates that the logic sense for the channel A Flag handshake line is negative-true. Bit 3 reset (0) indicates positive-true sense.
- Bit 2 set (1) indicates that the logic sense for the channel B Flag handshake line is negative-true. Bit 3 reset (0) indicates positive-true logic sense.
- Bit 1 set (1) indicates leading-edge trigger sense for the interface CTL A handshake line. This specifies that the CTL line goes false on the leading edge of the channel A Flag line. Bit 1 reset (0) indicates trailing-edge trigger sense for this handshake sequence.
- Bit 0 set (1) indicates leading-edge trigger sense for the interface CTL B handshake line. This specifies that the CTL line goes false on the leading edge of the channel B Flag line. Bit 0 reset (0) indicates trailing-edge trigger sense for this handshake sense for this handshake sequence.

## Register 8

**Data Sense**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Logic Sense for Channel B Input Data.				Logic Sense for Channel A Input Data.			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1



To access this register execute:

```
STATUS 3,8;S8
CONTROL 3,8;<value>
```

The value contained in this register indicates the logic sense of the control and handshake lines. To change this specification, execute the appropriate **CONTROL** statement to this register. The new specification remains in this register until changed by another **CONTROL** statement or until the interface is reset.

Reset default: 0

The meanings of the various bits are described next.

- Bit 7 (1) set indicates that the logic sense of the most significant bit (bit 3) of channel B data is negative-true. Positive-true logic sense is indicated by bit 7 clear (0).
- Bit 6 set (1) indicates that the logic sense of bit 2 of channel B data is negative-true. Positive-true logic sense is indicated by bit 6 clear (0).
- Bit 5 set (1) indicates that the logic sense of bit 1 channel B data is negative-true. Positive-true logic sense is indicated by bit 5 clear (0).
- Bit 4 set (1) indicates that the logic sense of the least significant bit (bit 0) of channel B data is negative-true. Positive-true logic sense is indicated by bit 4 clear (0).
- Bit 3 (1) set indicates that the logic sense of the most significant bit (bit 3) of channel A data is negative-true. Positive-true logic sense is indicated by bit 3 clear (0).
- Bit 2 set (1) indicates that the logic sense of bit channel A data is negative-true. Positive true-logic sense is indicated by bit 2 clear (0).
- Bit 1 set (1) indicates that the logic sense of bit 1 of channel A data is negative-true. Positive-true logic sense is indicated by bit 1 clear (0).
- Bit 0 set (0) indicates that the logic sense of the least significant bit (bit 0) of channel A data is negative-true. Positive-true logic sense is indicated by bit 0 clear (0).

## Register 9

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Logic Sense for Channel B Function Data.				Logic Sense for Channel A Function Data.			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,9;S9
CONTROL 3,9;<value>
```

The value contained in this register indicates the logic sense of the control and handshake lines. To change this specification, execute the appropriate `CONTROL` statement to this register. The new specification remains in this register until changed by another `CONTROL` statement or until the interface is reset.

Reset default: 0

The meanings of the various bits are described next.

- Bit 7 (0) set indicates that the logic sense of the most significant bit (bit 3) of channel B function is negative true. Bit 7 clear (0) indicates positive-true logic sense.
- Bit 6 set (1) indicates that the logic sense of bit 2 of channel B function is negative-true. Positive-true logic sense is indicated by bit 6 clear (0).
- Bit 5 set (1) indicates that the logic sense of bit 1 of channel B function is negative-true. Positive-true logic sense is indicated by bit 5 clear (0).
- Bit 4 set (1) indicates that the logic sense of the least significant bit (bit 0) of channel B function is negative true. Bit 4 clear (0) indicates positive-true logic sense.
- Bit 3 (1) set indicates that the logic sense of the most significant bit (bit 3) of channel A function is negative true. Bit 3 clear (0) indicates positive-true logic sense.
- Bit 2 set (1) indicates that the logic sense of bit 2 of channel A function is negative-true. Positive-true logic sense is indicated by bit 2 clear (0).
- Bit 1 set (1) indicates that the logic sense of bit 1 of channel A function is negative-true. Positive-true logic sense is indicated by bit 1 clear (0).
- Bit 0 set (1) indicates that the logic sense of the least significant bit (bit 0) of channel A function is negative true. Bit 0 clear (0) indicates positive-true logic sense.

## Register 10

**Handshake Lines and Port 10**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Logic Sense Exponent B	Logic Sense Mantissa B	Logic Sense Exponent A	Logic Sense Mantissa A	Logic Sense for Port 10 Data.			
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

To access this register execute:

```
STATUS 3,10;S0
CONTROL 3,10; <value>
```

The value contained in this register indicates the logic sense of the control and handshake lines. To change this specification, execute the appropriate `CONTROL` statement to this register. The new specification remains in this register until changed by another `CONTROL` statement or until the interface is reset.

Reset default: 0

The meanings of the various bits are described next.

- Bit 7 set (1) indicates negative-true logic sense for the sign bit of the channel B exponent. Bit 7 clear (0) indicates positive-true logic sense for this bit.
- Bit 6 set (1) indicates negative-true logic sense for the sign bit of the channel B mantissa. Bit 6 clear (0) indicates positive-true logic sense for this bit.
- Bit 5 set (1) indicates negative-true logic sense for the sign bit of the channel A exponent. Bit 5 clear (0) indicates positive-true logic sense for this bit.
- Bit 4 set (1) indicates negative-true logic sense for the sign bit of the channel A mantissa. Bit 4 clear (0) indicates positive-true logic sense for this bit.
- Bit 3 (1) set indicates that the logic sense of the most significant bit (bit 3) of port 10 data is negative-true. Bit 3 clear (0) indicates positive-true logic sense.
- Bit 2 set (1) indicates that the logic sense of bit 2 of port 10 data is negative-true. Positive-true logic sense is indicated by bit 2 clear (0).
- Bit 1 set (1) indicates that the logic sense of bit 1 of port 10 data is negative-true. Positive-true logic sense is indicated by bit 1 clear (0).
- Bit 0 set (1) indicates that the logic sense of the least significant bit (bit 0) of port 10 data is negative-true. Bit clear (0) indicates positive-true logic sense.

## Nonstandard Formatting

You can specify formats other than the single or dual channel default formats that are provided. This discussion explains how to select formats by executing `CONTROL` statements to the appropriate registers.

### Single Channel Formatting

Assume that you want to specify the following single channel format for channel A:

**Table 3-7. Format Selection**

Format Desired	Register Access	Value
5 mantissa digits	Register 3	5
2 exponent digits	Register 4	2
2 function digits	Register 5	2
4 digits to right of decimal point	Register 6	4
Negative true data	Register 8	15

**Example 10**

Run the following program and observe the results as shown:

```

10 RESET 3 ! RESET INTERFACE
20 STATUS 3,0 ; R0,R1,R2,C3,C4,C5,C6,C7,C8,C9,C0
30 IMAGE K,X,DDD, /
40 PRINT USING 30 ; "READ REG 0=";R0,"READ REG 1=";R1
50 PRINT USING 30 ; "READ REG 2=";R2
60 PRINT USING 30 ; REG 3=";C3,"REG 4=";C4,"REG 5=";C5
70 PRINT USING 30 ; "REG 6=";C6,"REG 7=";C7,"REG 8=";C8
80 PRINT USING 30 ; "REG 9=";C9,"REG 10=";C0
90 DISP "PAUSE"
100 PAUSE
110 CONTROL 3,3 ; 5,2,2,4@ CONTROL 3,8 ; 15 ! SET FORMAT
120 STATUS 3,0 ; R0,R1,R2,C3,C4,C5,C6,C7,C8,C9,C0
130 PRINT USING 30 ; "READ REG 0=";R0,"READ REG 1=";R1
140 PRINT USING 30 ; "READ REG 2=";R2
150 PRINT USING 30 ; REG 3=";C3,"REG 4=";C4,"REG 5=";C5
160 PRINT USING 30 ; "REG 6=";C6,"REG 7=";C7,"REG 8=";C8
170 PRINT USING 30 ; "REG 9=";C9,"REG 10=";C0
180 DISP "PAUSE2"
190 PAUSE

```

READ REG 0= 3	—Interface I.D.
READ REG 1= 0	—Interrupt Mask
READ REG 2= 0	—Handshake Lines
REG 3= 68	—Mantissa Digits Specified
REG 4= 0	—Exponent Digits Specified
REG 5= 17	—Function Digits Specified
REG 6= 0	—Decimal Point Location
REG 7= 0	—Handshake Logic Sense
REG 8= 0	—Data Logic Sense
REG 9= 0	—Function Logic Sense
REG 10= 0	—Signs and Port 10 Logic Sense

When PAUSE appears on the display, press **CONT** and observe the results as shown.

READ REG 0= 3	—Interface I.D.
READ REG 1= 0	—Interrupt Mask
READ REG 2= 0	—Handshake Lines
REG 3= 5	—Mantissa Digits Specified
REG 4= 2	—Exponent Digits Specified
REG 5= 2	—Function Digits Specified
REG 6= 4	—Decimal Point Location
REG 7= 0	—Handshake Logic Sense
REG 8= 15	—Data Logic Sense
REG 9= 0	—Function Logic Sense
REG 10= 0	—Signs and Port 10 Logic Sense

The interface is now configured to the desired format. This configuration remains until it is changed or until the interface is reset.

## Dual Channel Formatting

Assume that you want to specify the following dual channel format:

Table 3-8. Format Selection

Channel A		Channel B		Register Access
Format Desired	Value	Format Desired	Value	
2 mantissa digits	2	2 mantissa digits	32	Register 3
1 exponent digit	1	2 exponent digits	32	Register 4
1 function digit	1	1 function digit	16	Register 5
No decimal point	0	No decimal point	0	Register 6
Positive true data	0	Negative true data	240	Register 8

### Example 11

Add the following lines to the previous program:

```

175 ! SET 2 CHANNEL FORMAT
180 CONTROL 3,3 ; 34,33,17,0 @ CONTROL 3,8 ; 240 ! SET FMT
190 STATUS 3,0 ; R0,R1,R2,C3,C4,C5,C6,C7,C8,C9,C0
200 PRINT USING 30 ; "READ REG 0=";R0,"READ REG 1=";R1
210 PRINT USING 30 ; "READ REG 2=";R2
220 PRINT USING 30 ; "REG 3=";C3,"REG 4=";C4,"REG 5=";C5
230 PRINT USING 30 ; "REG 6=";C6,"REG 7=";C7,"REG 8=";C8
240 STOP

```

When PAUSE is displayed, press **CONT** and observe the results as shown here. The interface is now configured for the dual channel format as specified. This configuration remains until it is changed or until the interface is reset.

```

READ REG 0= 3      —Interface I.D.
READ REG 1= 0      —Interrupt Mask
READ REG 2= 0      —Handshake Lines
REG 3= 34          —Mantissa Digits Specified
REG 4= 33          —Exponent Digits Specified
REG 5= 17          —Function Digits Specified
REG 6= 0           —Decimal Point Location
REG 7= 0           —Handshake Logic Sense
REG 8= 240         —Data Logic Sense

```

## Data Output

The interface can output data via the I/O ports. The reset default switch (S2) switch 1 must be placed in the “1” position.

### Example 12

The next program assumes single channel default format. Here is the program:

```
10 RESET 3 ! RESET INTERFACE
20 CONTROL 3,2 ; 128 ! SET CHANNEL A TO OUTPUT MODE
30 STATUS 3,2 ; S2
40 DISP S2
50 IMAGE S8ZE,Z
60 OUTPUT 3 USING 50 ; 12.345678,2
70 STOP
```

The following data is output to the peripheral:

```
12345678E-6
2
```

The output image specification must correspond exactly with the mode of the interface card. The following image statements can be used for single and dual channel default formats:

```
D8ZE,Z - Single channel default format.
4Z,Z,M4Z,Z - Dual channel default format.
```

The interface can be configured for non-standard output formats in the identical manner as shown for non-standard inputs (see Non Standard Formats in the *I/O ROM Owner's Manual*).

## Transfers

When entering data with the TRANSFER statement, you can calculate the buffer sizes and count parameters as follows:

1. Count one byte for each mantissa and function digit.
2. Add one byte for mantissa sign (if applicable).
3. Count one byte for each exponent digit plus two bytes for the “E” and the exponent sign.

## BCD I/O Statements

Statement	Description
ABORTIO	Aborts any transfer in progress, sets all control lines false.
ASSERT	Immediately writes a value to Register 2, placing I/OA, I/OB, CTLA, CTLB, and port 10 lines in the specified state.
CONTROL	Writes values to the interface control registers.
ENABLE INTR	Writes enable mask to Control Register 1. Used to select event interrupts.
ENTER	Enters data from the interface into the BASIC program.
HALT	Aborts any transfer in progress. Does not change external lines and does not reset the interface.
OUTPUT	Outputs data from the BASIC program to the interface.
REMOTE	Sets the partial field specifier if the listen address ranges from 00 to 06, otherwise ignored.
RESET	Resets the interface to the default switch settings, sets I/O lines to the input state, and sets all handshake lines false.
SEND	CMD - Commands 32 through 38, and 64 through 70 are valid. DATA -transfers data to the peripheral. Values 0 through 32, and 44 are ignored. TALK and LISTEN - Addresses whose lower 5 bits evaluate from 0 to 6 set the partial field.
STATUS	Inputs values from the interface status registers.
TRANSFER	Moves data from a port to a buffer, or from a buffer to a port. INT and FHS transfers are allowed.

## Maintenance, Service, and Warranty

### Maintenance

There are no customer serviceable parts inside the HP 82941A BCD Interface. It should not be necessary to clean the interface module or cable contacts. The action of installing the module in the port or plugging the cable into a peripheral is normally sufficient to clean contamination from the contacts.

### Service

If at any time you suspect that the interface may be malfunctioning, do the following:

1. Turn off the computer and all peripherals. After disconnecting all plug-in devices from the ports, turn on the computer. If the cursor appears and no error message is displayed, the computer is functioning properly.
2. Turn off the computer. After installing the interface module in question in any port, turn on the computer.
  - If `ERROR 110 : I/O CARD` appears, the interface module requires service.
  - If the cursor does not appear, the system is not operating properly. To help determine if the interface module is interfering with proper operation, repeat this step with the module installed in a different port.
3. If improper operation is indicated in either the interface module or the computer, repair service is required.

### Warranty and Repair Service Information

The warranty statement and procedures for obtaining repair service are contained on the Warranty and Service Information sheet shipped with your HP 82941A BCD Interface. If you need additional information, please contact your authorized HP dealer or the nearest Hewlett-Packard sales and service facility.

If you have any questions concerning the warranty, please contact:

**In the U.S.:**

One of the six Field  
Repair Centers listed on the Service  
Information Sheet packaged with  
your owners's documentation.

**In Europe:**

Hewlett-Packard S.A.  
7, rue du Bois-du-lan  
P.O. Box  
CH-1217 Meyrin 2  
Geneva  
Switzerland  
Tel. (22) 82 70 00



**Other Countries:** Hewlett-Packard Intercontinental  
 3495 Deer Creek Road  
 Palo Alto, California 94304  
 U.S.A.  
 Tel. (415) 857-1501

If your system malfunctions and repair is required, you can help assure efficient service by providing the following items with your unit(s):

1. A description of the configuration of the HP Series 80 Personal Computer system, exactly as it was at the time of malfunction, including ROMs, interfaces, and other peripherals.
2. A brief yet specific description of the malfunction symptoms for service personnel.
3. Printouts or any other materials that illustrate the problem area.
4. A copy of the sales slip or other proof of purchase to establish the warranty coverage period.

Each computer and peripheral carries an individual serial number. It is recommended that you keep a separate record of this number. Should your unit be stolen or lost, the serial number is often necessary for tracing and recovery, as well as any insurance claims. Hewlett-Packard does not maintain records of individual owner's names and unit serial numbers.

## General Shipping Instructions

Should you ever need to ship any portion of your computer system, be sure it is packed in a protective package (use the original case), to avoid in-transit damage. Hewlett-Packard suggests that the customer always insure shipments.

If you happen to be outside of the country where you bought your computer or peripheral, contact the nearest authorized HP-83/85 dealer or the local Hewlett-Packard office. All customs and duties are your responsibility.

## Potential for Radio Frequency Interference

The HP 82941A BCD Interface uses radio frequency energy and may cause interference to radio and television reception. The interface has been type-tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J or Part 15 of the FCC Rules. These specifications provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If the interface does cause interference to radio or television, which can be determined by turning the computer on and off with the interface installed and the interface removed, you can try to eliminate the interference problem by doing one or more of the following:

- Reorient the receiving antenna.
- Change the position of the computer with respect to the receiver.
- Change the position of the peripheral(s) and interface cables with respect to the receiver.
- Move the computer away from the receiver.
- Plug the computer (and peripherals) into a different outlet so that the computer and the receiver are on different branch circuits.

If necessary, consult an authorized HP dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet, prepared by the Federal Communications Commission, helpful: *How to Identify and Resolve Radio-TV Interference Problems*. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

## Functional Description

### Introduction

This section contains a description of the interface circuit operations. This includes a schematic diagram description of various interface components and signal lines. Although this information is provided, component level repair is not recommended due to the microcomputer based organization of the interface.

### Translator IC Description

The interface uses an 8049 microcomputer ( $\mu$ C) which requires the +5 TTL logic level. The microprocessor in the HP Series 80 Personal Computer (CPU) uses a +6V logic level. A special IC, known as a Translator, permits communication between the two devices by providing level translation. The +5V and +6V power supplies and two +12V clock signals ( $\Phi$ 1 and  $\Phi$ 2) required by the translator are located on the mainframe. They are brought out to the interface via the I/O backplane when the interface is inserted into one of the four I/O ports.

The translator supports the following operations:

- Handshaking of data and command information from the CPU to the  $\mu$ C.
- Handshaking of data from the  $\mu$ C to the CPU.
- Interrupts issued to the CPU by the  $\mu$ C.
- Interrupts issued  $\mu$ C by the CPU.
- Fast handshake operation where the translator halts the computer with each data byte transfer to synchronize the flow of data.

It may be helpful to refer to the interface schematic diagram (figure 4-5) while reading the theory discussed in this section.

### Select Codes

Interfaces are memory mapped. When the computer addresses an interface, it provides an address over the address and data bus (B0 through B7). The Translator demultiplexes the address and then compares address bits (A1, A2, A3) with select code bits (SC0, SC1, SC2). If they match, the interface is addressed. When addressed, another bit (A0) is sent out by the Translator's Address Demultiplexer. If A0 is low, the CPU writes to the control register (CR) and reads the status register (SR). When A0 is high, the CPU writes to the output buffer (OB) and reads the input buffer (IB). These four registers are discussed next.

### Translator I/O Registers

The computer sends data to the microcomputer via the Output Buffer Register (OB) and defines that data by setting appropriate bits in the Control Register (CR). Both registers are physically located within the translator IC. The OB is write-only by the CUP and read-only by the microcomputer. Except for bit 0 and bit 7, the CR is also write-only by the CPU and read-only by the microcomputer. Different status bits are read by the microcomputer for bit 0 and bit 7 of the CR than those written as bits 0 and 7 by the computer.

**CPU Write to CR**

7	6	5	4	3	2	1	0
RESET	0	0	0	0	CED	COM	CINT

 **$\mu$ C Read CR**

7	6	5	4	3	2	1	0
OBF	0	0	0	0	CED	COM	IBF

CINT	Interrupt. This bit is routed to the T1 input of the microcomputer ( $\mu$ C) which is used to interrupt.
COM	Command.
CED	End Data. Used by CPU to terminate a data transfer to the $\mu$ C.
RESET	Resets the $\mu$ C. This bit is routed directly to the reset input of the $\mu$ C.
IBF	Input buffer full.
OBF	Output buffer full.

Two other registers within the translator are used when the CPU receives data from the  $\mu$ C. These registers are the input buffer register (IB) used to handle the data and the Status Register (SR) which contains status bits to implement communication protocol. The IB is write-only by the  $\mu$ C and read-only by the CPU. Except for bit 0 and bit 7, the SR is also write-only by the  $\mu$ C and read-only by CPU. Different status bits are read for bits 0 and 7 by the CPU than those written as bits 0 and 7 by the  $\mu$ C.

 **$\mu$ C Write to SR**

7	6	5	4	3	2	1	0
HLTEN	0	0	0	PACK	PED	BUSY	SERVICE

**CPU Read SR**

7	6	5	4	3	2	1	0
OBF	0	0	0	PACK	PED	BUSY	IBF

SERVICE	The $\mu$ C sets this bit to initiate an interrupt to the CPU.
BUSY	Informs the CPU of the state of the $\mu$ C. When low, $\mu$ C is monitoring OBF to determine when the next command or data byte is available from the CPU. When OBF = 1, the $\mu$ C sets BUSY to 1, reads the OB and CR, performs the necessary operation then returns BUSY to low.
PED	Processor ( $\mu$ C) End Data. Set high by the $\mu$ C upon detection of a programmed termination sequence (byte count).

PACK	Processor ( $\mu$ C) Interrupt Acknowledge. The $\mu$ C's interrupt service routine sets this bit after being interrupted by the computer.
HLTEN	Used by the $\mu$ C to halt the CPU based on the status of OBF and IBF during fast handshake data transfers.
IBF	Input Buffer Full.
OBF	Output Buffer Full.

## Data Transfers

The 48 bi-directional data lines consist of twelve 4-bit ports, P0 through P11. P0 through P10 may be used to transfer mantissa, exponent and/or function digits; P11 is dedicated to transfer mantissa and exponent sign bits for both channels. This port structure was outlined in section 2.

Let's look at how a data transfer takes place. To simplify the discussion, the handshake timing will be discussed later.

### Inputs

The 16 BCD characters that can be input from a peripheral were listed in table 1-1. Each character requires one port (4 lines) and may be transferred via ports P0 through P10. For an input, the peripheral places the data to be transferred on the ports it is configured to. The interface inputs one port at a time, starting with the port containing the most significant digit of the transfer. As a port is read, the BCD data is converted to one of the 16 ASCII characters of table 1-1 and sent to the CPU. Then the next port is read, and so on. If a mantissa or exponent field is included in the transfer, the appropriate sign bits from P11 are also input. A "+" sign is obtained when a sign bit line is false; a "-" sign is obtained when it is true. For example, if positive true logic is configured for the sign bits, a low level is false (+) and a high level is true (-). For negative true logic, a low level is true (-) and a high level is false (+). P11 is not read for function digits.

### Outputs

For an output, the computer sends an 8-bit ASCII character to the interface. The  $\mu$ C on the interface accepts the ASCII character and places the lower four bits of the code on the lines of the appropriate port. When all the data to be output is on the lines, the peripheral accepts the data. This is true for all ASCII characters except ASCII characters 0 through 32 plus 44, which are ignored. An ASCII "+" or "-" may precede the mantissa or exponent to be output correctly to P11. If no ASCII sign character is sent, P11 will default to the "+" (false state).

Mantissa, exponent and function field data is output in the above manner. This means that non-digit data (BCD codes above 1001 (9)) may be output to a peripheral without causing an error message to be generated, depending upon the peripheral. The manner in which non-digit data is interpreted by a peripheral is entirely device dependent; they do not have to be the same as the non-digit characters listed in table 1-1.

## Handshake Lines and Timing

Let's now examine the handshake lines and timing that permit the above transfers to take place.

Each channel has three handshake lines: I/O, CTL, and FLG. Trailing edge or leading edge handshake modes may be configured. Trailing edge refers to the CTL line going false either when the FLG line goes false or on the trailing edge of the FLG line going false. Leading edge refers to the CTL line going false either when FLG goes true or on the leading edge of FLG going true. Refer to figures A-1 and A-2 as you read the following sequences for both an input and output. The manner in which a particular port is addressed is discussed later.

For an input from a peripheral, the interface drives the I/O line false. To initiate the transfer, the interface sets the CTL line true. The peripheral responds by setting FLG true, takes a reading and gets the data ready. If a leading edge handshake is configured, the interface will place CTL false when it detects FLG going true. When the peripheral has the data on the lines it sets FLG false. If a trailing edge handshake is configured, the interface will then drive CTL false. Regardless of the handshake mode configured, the interface will input the data when it detects FLG going false and send the data to the CPU. The peripheral must hold the data valid on the lines until the interface requests the next reading by setting CTL true. This permits the interface to be interrupted by the computer after both CTL and FLG are returned false but before the data lines are read and resume reading the lines after the interrupt is serviced.

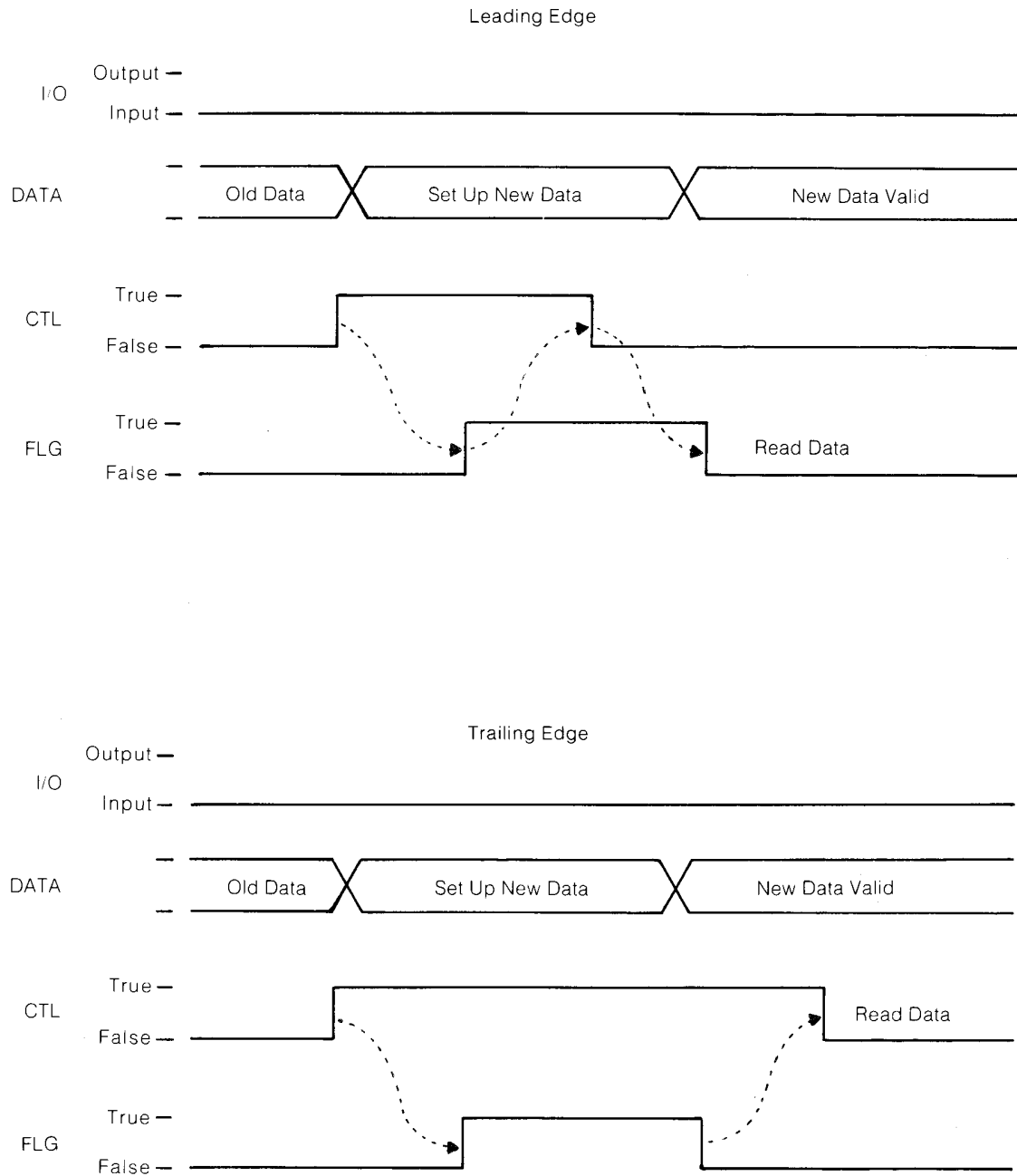
For an output to a peripheral, the interface drives the I/O line true. The computer places the data on the lines of the port(s) prior to the handshake. When all of the data is ready, the interface sets CTL true. The peripheral responds by setting FLG true and accepting the data. If a leading edge handshake is configured, the interface will drive CTL false when it detects FLG going true. When the data is transferred, the peripheral sets FLG false. If a trailing edge handshake is configured, CTL is driven false when it detects FLG going false. Regardless of the configuration, new output data may be placed on the data lines after FLG is returned false.

There is no maximum timing requirement for the FLG line. However, it must be held true for at least  $15\mu\text{s}$  to ensure the interface doesn't miss it and thereby not complete a handshake and hang. Also, an interrupting command should not be issued to the interface during a handshake sequence unless the FLG line is held true for at least  $15\mu\text{s}$  after the interrupt is serviced. If this was allowed to happen, the interface might initiate a transfer, get interrupted before the handshake is completed and perhaps miss the FLG pulse because it was in an interrupt service routine. This would also cause the interface to hang.

When channels A and B are both being used, the channel A handshake and data transfer is completed before channel B. The two do not occur simultaneously. However, simultaneous triggering of two peripherals may be accomplished by terminating CTLA to both peripherals with both response lines from the peripherals ANDed to FLGA. This would require external logic circuitry in some cases.

**Figure A-1. Input Handshake Timing**Flag Duration: 15 $\mu$ s Minimum

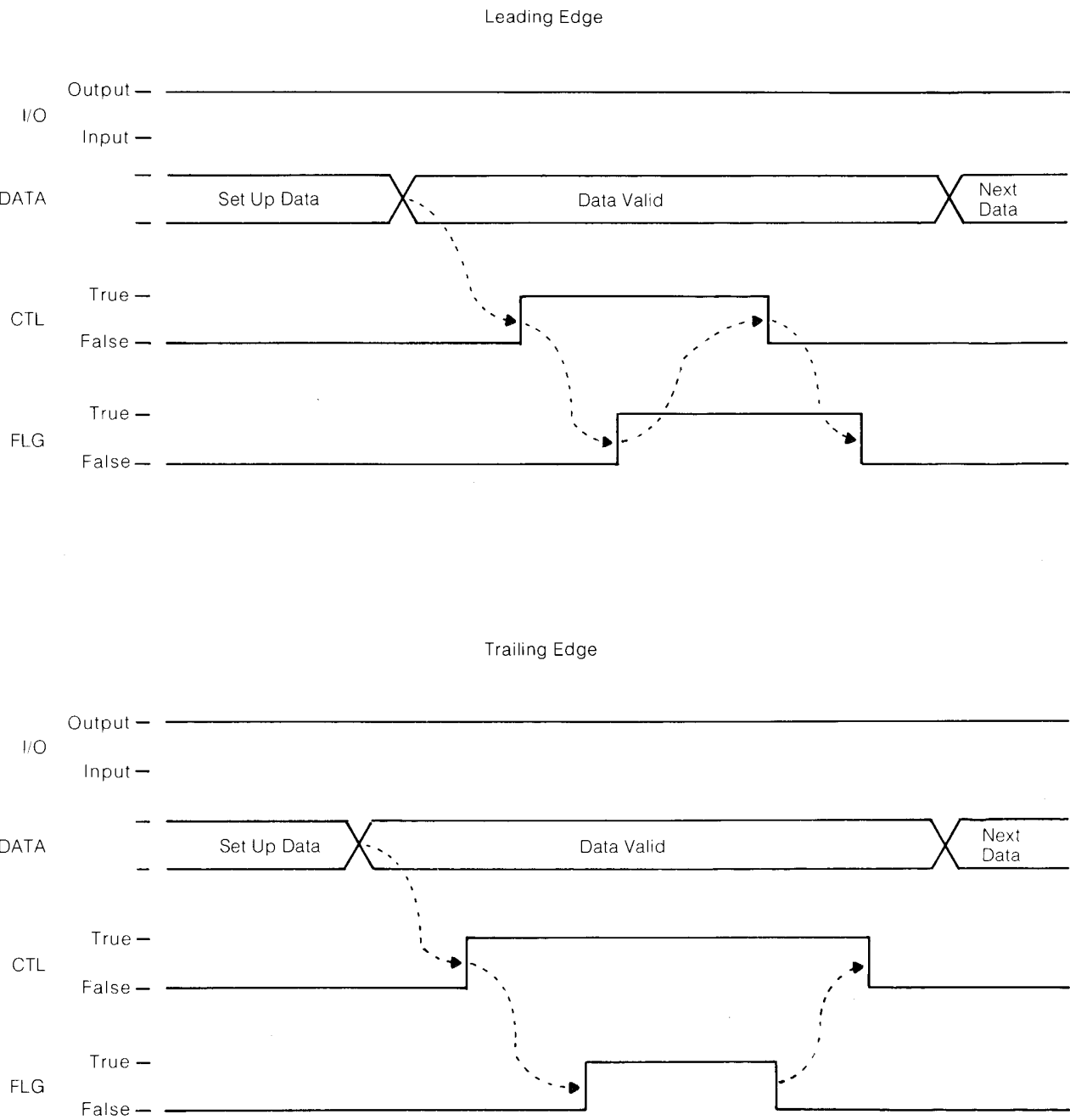
Timings Are Not Valid If Card Is Interrupted During Handshake

**Figure 3-4. Input Handshake Timing**

Flag Duration: 15 $\mu$ s Minimum

Data Valid 40 $\mu$ s Minimum Before Control  
(15 $\mu$ s Minimum On Fast Handshake)

Timings Are Not Valid If Card Is Interrupted During Handshake



**Figure A-2. Output Handshake Timing**



## Switch Buffer

The switch buffer isolates default switch S2 (segments 3 through 8) from D0 through D5 of the interface address-data bus. The default configurations were discussed in section 2.

U3 is enabled only during initialization or when the 8049  $\mu\text{C}$  is reset allowing the switch settings to be read by the  $\mu\text{C}$ . Once the switches are read, it isn't necessary to do so again unless the  $\mu\text{C}$  is reset.

To read the switches, the  $\mu\text{C}$  sends an address to the Translator read/write logic and latches it with the ALE line. The Translator responds by placing ADR3 low. The  $\mu\text{C}$  then places RD low. During the period when ADR3 and RD are low, U3 is enabled and the switch settings are read by the  $\mu\text{C}$ .

## Handshake Buffer

This hex inverter buffer has open collector outputs. It provides the drive capability for the handshake lines of both channels and buffers these lines between the  $\mu\text{C}$  and BCD peripherals. The 2.2 k $\Omega$  pullup resistors used on these lines decrease noise susceptibility and encourage a fast signal rise time. The peripheral response lines use 10k $\Omega$  pullup resistors on the  $\mu\text{C}$  side of the buffer.

## I/O Expanders

Each I/O expander has four 4-bit ports on the peripheral side and one 4-bit port on the 8049  $\mu\text{C}$  side. The ports on the  $\mu\text{C}$  side are all connected in a parallel and pass data to and from the  $\mu\text{C}$  via the ports on the peripheral side. Thus, as the name implies, the I/O expanders extend the I/O capability of the  $\mu\text{C}$ .

The following table gives a description of the lines and pinouts of the three I/O expanders.

**Table A-1. I/O Expander Pinout**

Symbol	Pin No.	Function
PROG	7	Clock Input. A high-to-low transition signifies that address and control information is on P20 — P23; a low-to-high transition signifies that data is on P20 — P23.
CS	6	Chip Select Input. When high, any change on an output is inhibited.
P20 — P23	11 — 8	When PROG has a high-to-low transition, P20 — P23 contains address and control information. When PROG has a low-to-high transition, P20 — P23 has data for a selected output port or data from a selected port before the low-to-high transition occurs.
P40 — P43	2 — 5	4-bit bi-directional I/O port.
P50 — P53	1, 23 — 21	4-bit bi-directional I/O port.
P60 — P63	20 — 17	4-bit bi-directional I/O port.
P70 — P73	13 — 16	4-bit bi-directional I/O port.
V <sub>cc</sub>	24	+5 Vdc.
GND	12	Circuit Ground.

Each I/O expander is identical. Therefore, the scheme used to address the various I/O ports having the same pin numbers from one IC to the next is also identical. For example, the same address code is used to specify ports P0, P4 or P8 because they have the same pinout but on different IC's. Once the appropriate port is specified, the IC containing the port is activated via its CS line. Thus, to select port P0, U5 is activated when its CS line goes low. Similarly, P4 is selected by activating U6, and P8 is selected by activating U7.

P20 and P21 address the I/O port on the peripheral side while P22 and P23 specify the operation to take place as listed in the following table.

**Table A-2. I/O Port Selection and Operation**

P21	P20	Ports Selected	P23	P22	Operation Specified
0	0	P0, P4, P8	0	0	Read
0	1	P1, P5, P9	0	1	Write
1	0	P2, P6, P10	1	0	ORLD
1	1	P3, P7, P11	1	1	ANLD

The ORLD operation takes new data and performs a logical OR operation with it and the old data and then writes the result to the selected port.

The ANLD operation takes new data and performs a logical AND operation with it and the old data and then writes the result to the selected port.

## 8049 Microcomputer

Many of the activities carried out by the  $\mu$ C have already been discussed in this section. This discussion will concentrate more on data transfers, give the pin assignments and provide information about the  $\mu$ C registers. It may be helpful to fold out the schematic diagram as you read the following discussion.

The  $\mu$ C is the intermediary between the computer and BCD peripherals. It implements interface protocol with its own self-contained ROM. The  $\mu$ C responds to instructions from the host computer, accepts BCD data from peripherals, converts it to one of the ASCII codes of table 1-1 and forwards it to the CPU. If the interface outputs to a peripheral, the  $\mu$ C accepts any standard ASCII code from the computer and sends the lower four bits of the code to the peripheral. This is true for all ASCII codes except the control codes (0 — 31), space (32) and comma (44). These characters are ignored.

Input and output data between the interface and peripherals is routed via the lower four bits (P20 — P23) of an 8-bit bi-directional port. The next three bits of this port (P24 — P26) are used to select the individual I/O expanders via their CS inputs. P27 of this port is used only when a special test connector is installed on the interface and causes the  $\mu$ C to perform an extended self-test at power-on or reset.

Let's now consider an input from a peripheral using the standard switch settable format where mantissa, exponent and function digits are transferred by their respective ports. This is outlined in Table A-3. Keep in mind that a transfer does not require all three fields, nor is a particular field constrained to the port assignments shown if a software configuration is made.

Table A-3 Standard Format

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+ -	X	X	X	X	X	X	X	X	E	+ -	X	,	X	LF
S1	P0	P1	P2	P3	P4	P5	P6	P7	S2	P8		P9		
	MSD			Mantissa				LSD		Exp			Fun	

The  $\mu$ C first inputs port P11 to obtain the mantissa sign bit (S1), converts it to an ASCII “+” or “-” and forwards it to the CPU. Then, starting with port P0, the  $\mu$ C inputs the most significant mantissa digit, converts it to ASCII code and sends it to the CPU. The remaining mantissa ports (P1 — P7) are transferred one at a time in the same manner. When the exponent field is selected, the  $\mu$ C generates the ASCII code for “E” and sends it to the CPU. Then port P11 is input again to obtain the exponent sign bit (S2). After sign bit S2 is transferred, port P8 is input and transferred for the exponent digit. The  $\mu$ C next generates and sends the ASCII code for “,” which separates the exponent field from the function field and then inputs and transfers port P9 for the function digit. The transfer is then terminated when the  $\mu$ C generates and sends the ASCII code for “LF” to the CPU.

Notice when mantissa or exponent fields are selected, port P11 is first input to obtain the appropriate sign bit. Therefore, the sign bit lines should always be connected so that proper signs are reflected. Also notice that ASCII codes “E”, “,” and “LF” were generated by the  $\mu$ C, not the peripheral. The  $\mu$ C can also generate decimal points, which is discussed later.

Obviously, the above configuration will not satisfy all needs. Suppose your installation requires five mantissa digits, two exponent digits and no function digits. Furthermore, suppose a decimal point is needed in the mantissa so there will be three mantissa digits to the right of it. Let’s see how software can accomplish such a configuration.

Some of the  $\mu$ C registers, which are discussed later, reflect the default switch settings unless changed by software. Most of these registers can be changed to meet the needs of installation. Assuming the select code is set to 3, the following statement will configure the interface as outlined in the previous paragraph:

```
CONTROL 3,3;5,2,0,3
```

To understand the above statement, refer to the register tables on page 00. Notice that the mantissa, exponent, function and decimal point registers are in sequential order (3 — 6). In our statement, we have addressed select code 3 and the first register which we want to change the contents of (Register 3) and then placed the number “5” into that register. When the contents of Register 3 were changed, the register pointer automatically incremented to register 4 and its contents changed to “2”. Likewise, Register 5 was changed to “0”, and Register 6 to “3”.

Now that the interface has been configured differently, let’s look at the transfer sequence again.

As before, the  $\mu$ C first inputs port P11 to obtain the mantissa sign bit (S1). The most significant mantissa digit is then input and transferred via port P0. Port P1 then transfers the next mantissa digit. The  $\mu$ C then generates the decimal point and forwards it to the CPU. The next mantissa digit is then input and transferred via port P2. After port P4 is transferred, the mantissa is complete and the  $\mu$ C generates and sends the ASCII code for “E” to the CPU and then inputs port 11 again to obtain the exponent sign bit (S2). The first exponent digit is next input and transferred except that it is now transferred via port P5,

whereas before it was transferred via port P8. The second exponent digit is then input and transferred via port P6. Since function digits aren't specified, the  $\mu$ C generates and sends the ASCII code for "LF" to the CPU after the last exponent digit is transferred which terminates the transfer.

Let's now look at an input using the option switch settable format. The digits are assigned to the ports as outlined in table A-4. The sequence is similar to the standard format except that exponent fields are not used and there are now two channels.

**Table A-4. Option Format (Channels A & B)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
+ -	X	X	X	X	,	X	LF	+ -	X	X	X	X	,	X	LF
S1	P0	P1	P2	P3		P4		S3	P5	P6	P7	P8		P9	
	Mantissa A								Mantissa B					Fun B	
	MSD			LSD					MSD			LSD			

In the option format, the interface first reads port P11 to obtain the channel A mantissa sign bit (S1) and transfers it. The channel A mantissa is then transferred one digit at a time starting with port P0. After P3 is transferred, the  $\mu$ C generates the ASCII code for "," and then transfers the channel A function digit. This completes the channel A transfer and it is terminated by the  $\mu$ C generating the ASCII code for "LF". When channel B is transferred, port P11 is again input to obtain the channel B mantissa sign bit (S3). The channel B mantissa is transferred one digit at a time starting with port P5 and ending with port P8. The  $\mu$ C then generates the ASCII code for "," and transfers the function digit via port P9. The transfer is terminated when the  $\mu$ C generates the "LF" and sends it to the CPU.

In the option format, the number of mantissa and function digits may be specified with software as they were in the standard format. Thus, channel A is not constrained to ports P0 through P4. It may use more of the ports than channel B, it may use the same number as shown in table A-4, or it may use less. If you recall in Section 2, it was stated that both cables could be terminated to one peripheral or one of the cables could be split up between two peripherals. Also, notice that in the option format, port P11 is input twice to obtain the mantissa sign bits for both channels.

For an output, the number of digits in all fields selected must match the way the interface is configured (switches or software) or a data format error will occur. Sign bits for mantissa or exponent fields are optional on outputs in normal and interrupt transfer modes (but not in the fast handshake mode) and, if they are not included, "+" sign is assumed. If an exponent field is selected, the capital "E" must be used, not the lowercase "e".

The following  $\mu$ C registers are accessible with the **CONTROL** and **STATUS** statements.

**Read Register 0 (interface ID)**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

Write to register 0 will cause an error.

**Read Register 1 (Interrupt Cause)**

7	6	5	4	3	2	1	0
Function B MSD				Function A MSD			

**Write Register 1 (Interrupt Mask)**

7	6	5	4	3	2	1	0
Function B MSD				Function A MSD			

0 = Disabled                      1 = Enabled

If any bit set in the Interrupt Mask Register matches a bit set on the proper function port, a corresponding bit is set in the Interrupt Cause Register by the  $\mu$ C. This causes the interface to generate an interrupt to the CPU.

**Read Register 2 (Control Line Messages)**

7	6	5	4	3	2	1	0
I/OA	I/OB	CTLA	CTLB	FLGA	FLGB	0	0

**Write Register 2 (Handshake Lines and Port 10)**

7	6	5	4	3	2	1	0
I/OA	I/OB	CTLA	CTLB	Port 10			

**I/O Bits**

0 = Input

1 = Output

**CTL Bits**

0 = Ready

1 = Busy

**FLG Bits**

0 = Ready

1 = Busy

**Read/Write Register 3 (Mantissa)**

7	6	5	4	3	2	1	0
# digits in channel B				# digits in channel A			

The total number of mantissa digits for both channels cannot exceed 11.

The sum of digits specified by registers 3, 4 and 5 cannot exceed 11 for both channels simultaneously.

**Read/Write Register 4 (Exponent)**

7	6	5	4	3	2	1	0
# digits in channel B				# digits in channel A			

Up to three exponent digits may be specified for both channels simultaneously.

**Read/Write Register 5 (Function)**

7	6	5	4	3	2	1	0
# digits in channel B				# digits in channel A			

The total number of function digits for both channels cannot exceed 11.

**Read/Write Register 6 (Decimal Point)**

7	6	5	4	3	2	1	0
# digits to right of decimal point in channel B				# digits to right of decimal point in channel A			

The number specified for either channel cannot be larger than the number of mantissa digits specified for that channel.

**Read/Write Register 7 (Handshake Sense)**

7	6	5	4	3	2	1	0
I/OA	I/OB	CTLA	CTLB	FLGA	FLBG	Handshake A	Handshake B

**I/O, CTL and FLG Bits**

0 = Positive true logic

1 = Negative true logic

**Handshake Bits**

0 = Trailing edge

1 = Leading edge

**Read/Write Register 8 (Data Sense)**

7	6	5	4	3	2	1	0
Channel B Data				Channel A Data			

0 = Positive true logic

1 = Negative true logic

**Read/Write Register 9 (Function Sense)**

7	6	5	4	3	2	1	0
Channel B Function				Channel A Function			

0 = Positive true logic

1 = Negative true logic

**Read/Write Register 10 (Signs and Port 10 Sense)**

7	6	5	4	3	2	1	0
Exponent Sign Channel B	Mantissa Sign Channel B	Exponent Sign Channel A	Mantissa Sign Channel A	Port 10			

0 = Positive true logic

1 = Negative true logic

**Table A-5. BCD Interface Replaceable Parts List**

Reference Designator	CD	HP Part No.	TQ	Description
A1	2	82941-60901	1	PC Assembly
C1, C2	6	0180-0228	2	C-F: 22 $\mu$ F, 15V
C3	4	0160-4767	1	C-F: 20pF, 200V
C4 — C13	5	0160-4571	10	C-F: 0.1 $\mu$ F, 50V
J1, J2	0	1251-6492	2	Connector: 32-Pin
R1	4	1810-0278	1	Resistor Network: 3.3k $\Omega$ , 2%, 1.25W
R2	2	1810-0367	1	Resistor Network: 4.7k $\Omega$ , 2%, 1.25W
R3 — R9	8	1810-0280	7	Resistor Network: 10k $\Omega$ , 2%, 1.25W
R10	9	1810-0231	1	Resistor Network: 2.2k $\Omega$ , 2%, 1.25W
S1	8	3101-2533	1	Switch: 4-Segment, SPST
S2	1	3101-2534	1	Switch: 8-Segment, SPST
U1	3	1MB5-0101	1	IC: Translator
U2	4	1820-2439	1	IC: 8049 $\mu$ C
U3	9	1820-1402	1	IC: 80C95, Hex Buffer
U4	1	1820-0175	1	IC: 7405, Hex Inverter
U5 — U7	7	1820-2177	3	IC: 8243, I/O Expander
U8	-	1820-1112	1	IC: 7474, Dual D Flip-Flop
Y1	1	0410-1222	1	11 MHz Crystal
	9	0363-0174	1	Ground Contact
	0	2200-0143	7	Screws: 4-40 Machine
	8	1400-1063	2	Top Cable Clamp
	9	1400-1064	2	Bottom Cable Clamp
	9	0590-0199	4	Hex Nut With Lockwasher
	3	82941-60902	1	Case, Top and Bottom
		8120-3192	1	Interface Cable

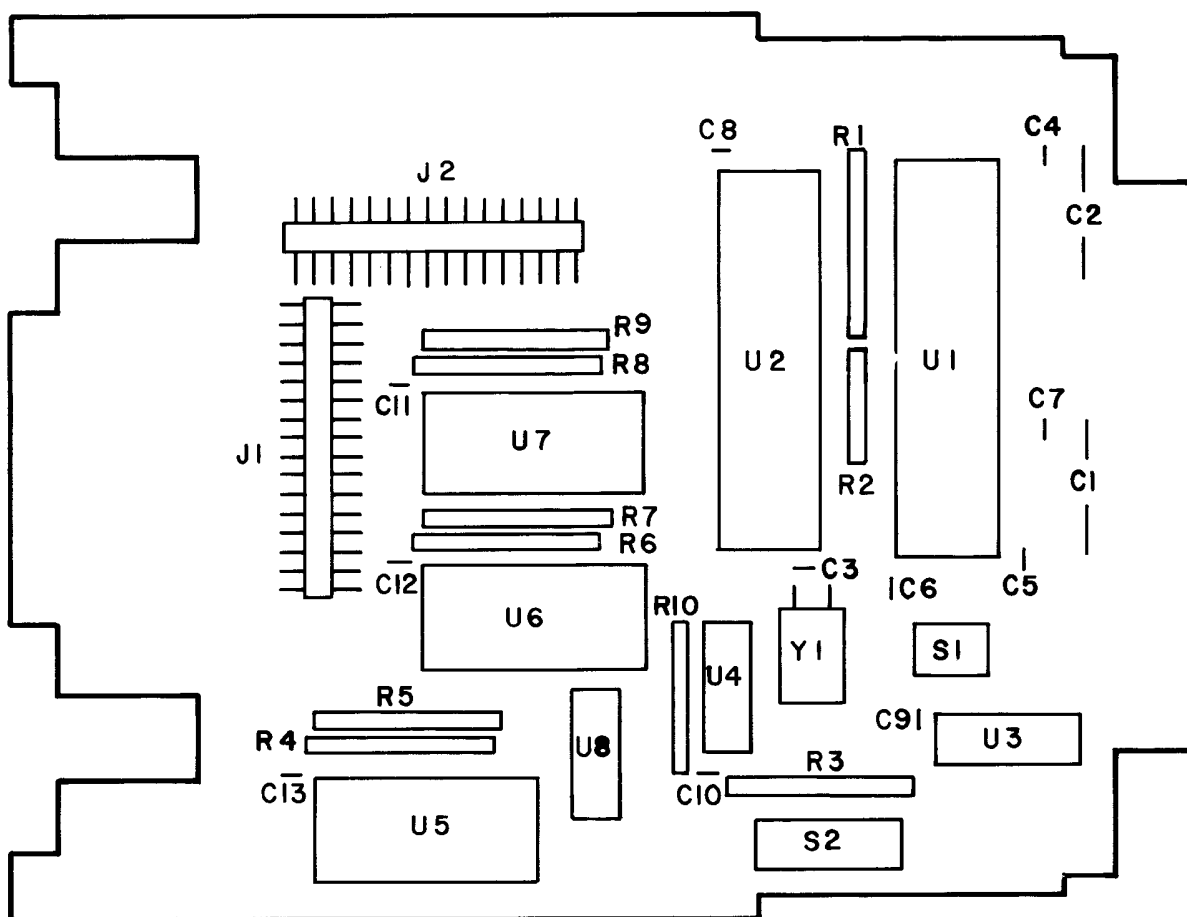
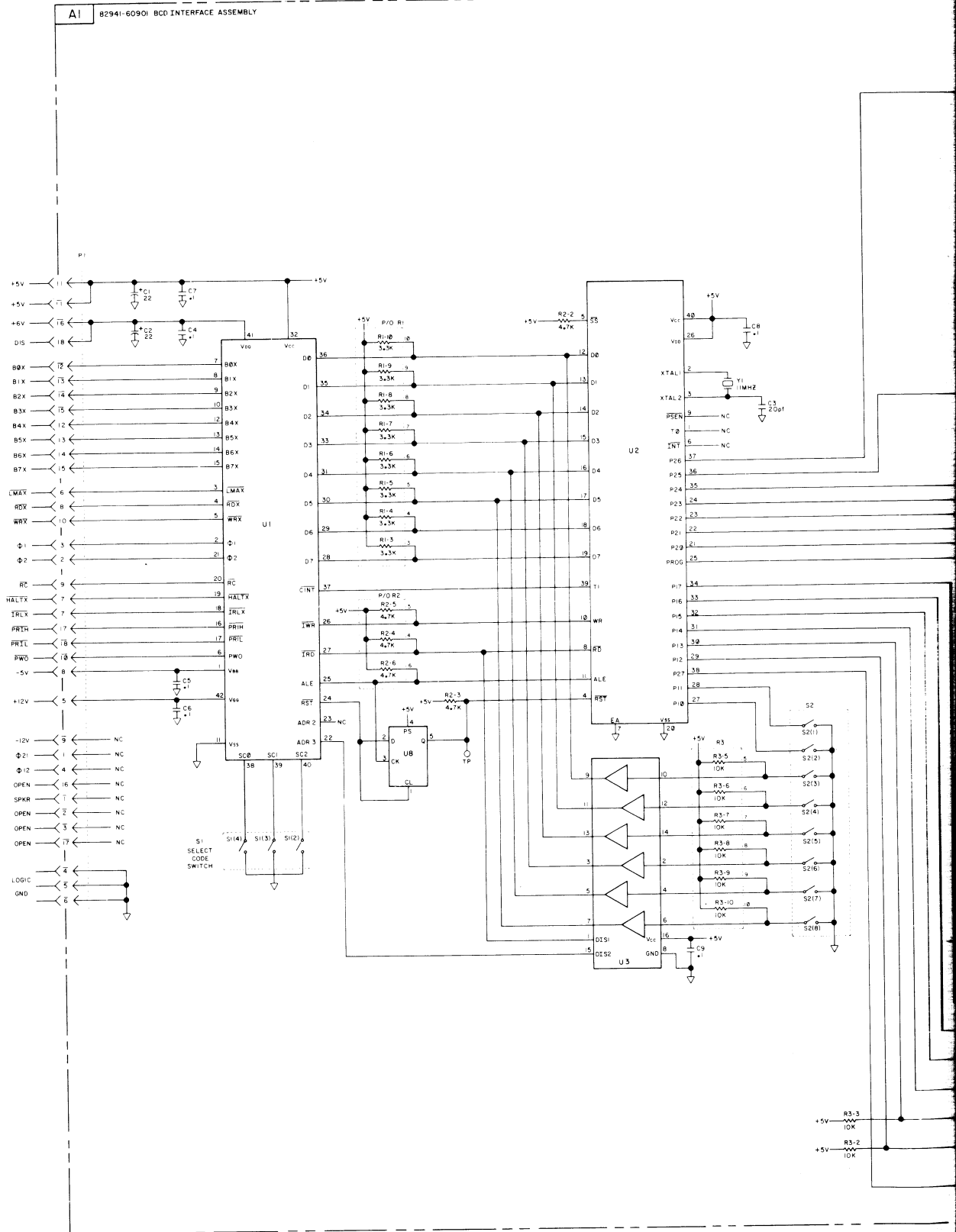


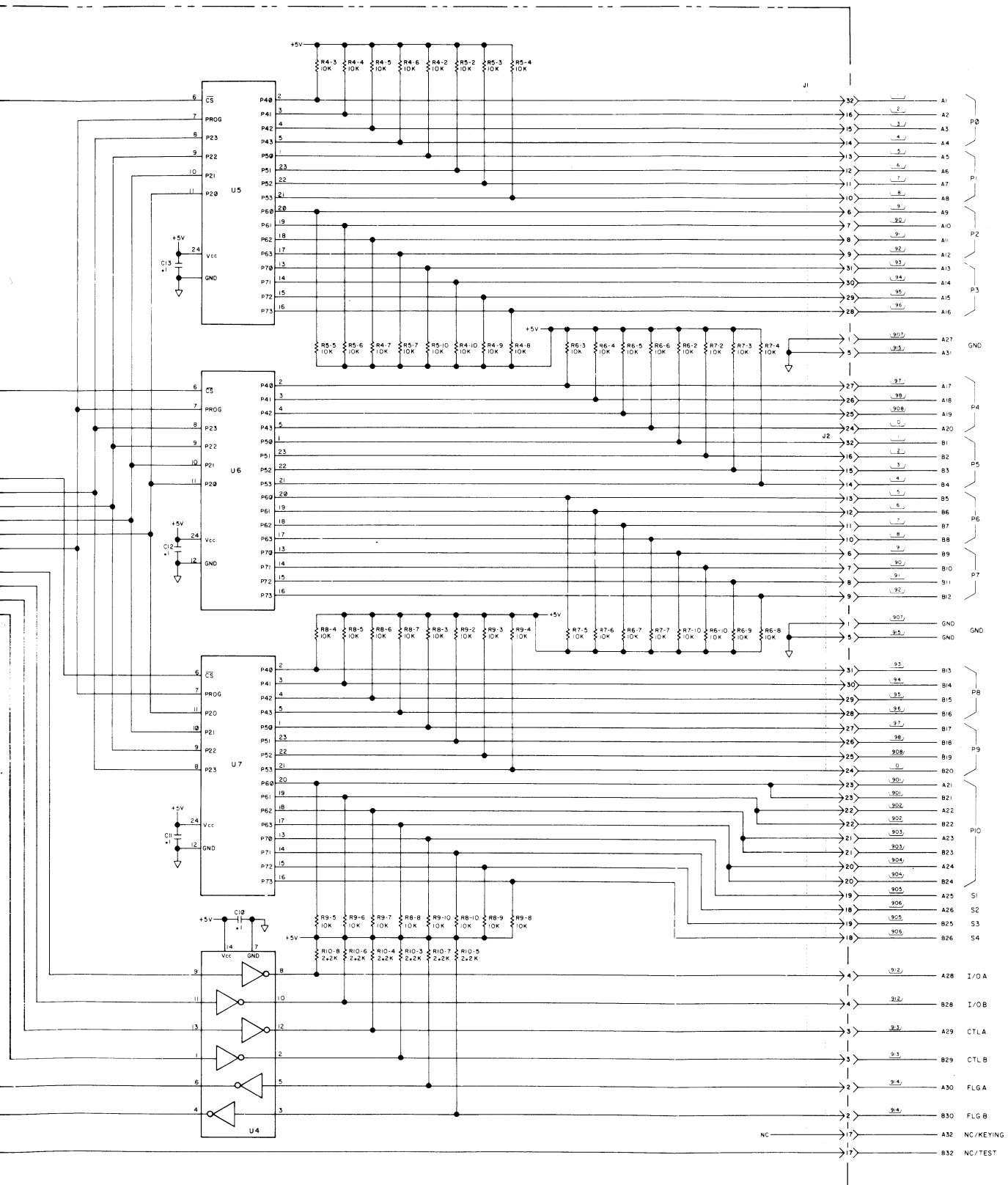
Figure A-4. BCD Interface Component Locator





**Figure A-5. BCD Interface Schematic Diagram**





## BCD Interface Errors

### BCD Interface Errors

Error No.	Meaning	Possible Cause
110	Self test failure in I/O card.	Hardware failure.
111	Illegal I/O operation.	Instruction not recognized by the interface. Illegal parameter in instruction. Exceeded the available number of registers.
113	Illegal mode of card.	Exceeded 11 digits total. More than 3 exponent digits per channel. Channel B has mantissa or exponent digits when direction of channel B does not equal the direction of A. Not standard format for FHS. Tried to set channel to output when the output enable switch was open.
114	Port 10 not available.	Tried to write data to port 10 when it was selected by a channel.
115	Fast handshake transfer aborted by the peripheral.	Flag B went true during fast handshake.
116	Direction mismatch.	Tried to OUTPUT to an input channel or ENTER from an output channel.
117	Instruction directed to nonexistent field.	Field was allocated 0 digits. Partial field specifies a field of 0 digits.
118	Handshake not ready.	Control was true at start of handshake.
119	Data format error.	Output data did not match card's mode in number of digits.



Personal Computer Division  
1010 N.E. Circle Blvd., Corvallis, OR 97330 U.S.A.

Reorder Number  
82941-90011

Printed in U.S.A. 5/82  
82941-90015