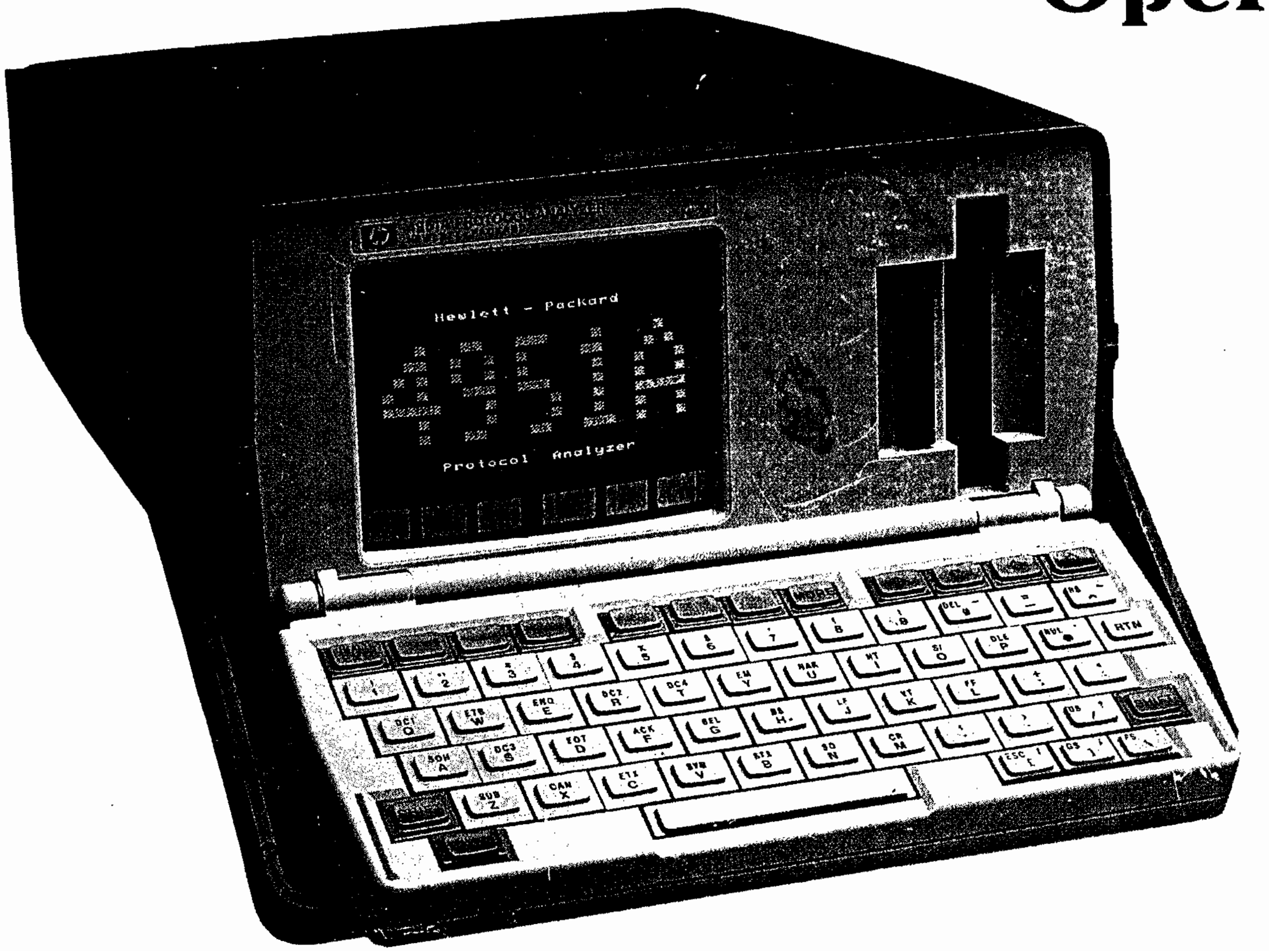
4951A Protocol Analyzer

Operating Manual



HEWLETT
PACKARD

4951A
PROTOCOL ANALYZER

Operating Manual

DC100 TAPE CARTR.

CAT. No

7045-7N-997-7746

SERIAL NUMBERS

This manual applies directly to instruments
with serial numbers prefixed *2347A.*

Store 35 356 AE

Manual Part No: 04951-90003
Microfiche Part No: 04951-90004

Printed April 1984

TABLE OF CONTENTS

SAFETY SUMMARY

The following general safety precautions must be observed during all phases of operation; service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of this instrument. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

GROUND THE INSTRUMENT.
To minimize shock hazard, the instrument chassis and cabinet must be connected to an electrical ground. The instrument is equipped with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet international Electrotechnical Commission (EC) safety standards.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.
Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS.
Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE.
Do not attempt internal service or adjust unless another person, capable of rendering first aid and resuscitation, is present.

USE CAUTION WHEN EXPOSING OR HANDLING THE CRT.
Breakage of the Cathode-ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, avoid rough handling or jarring of the instrument. Handling of the CRT shall be done only by qualified maintenance personnel using approved safety mask and gloves.

DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT.
Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the instrument. Return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

PRINTING HISTORY

Each new edition of this manual incorporates all material updated since the previous edition. Manual change sheets are issued between editions, allowing you to correct or insert information in the current edition.

The part number on the back cover changes only when each new edition is published. Minor corrections or additions may be made as the manual is reprinted between editions. A vertical bar on the edge of a page indicates a change from the previous edition.

First Printing..........................April 1984

PART I
GETTING STARTED

CHAPTER I
LEARNING THE CONTROLS

Power Up and Installation
Front Panel Controls
Top Level Softkeys

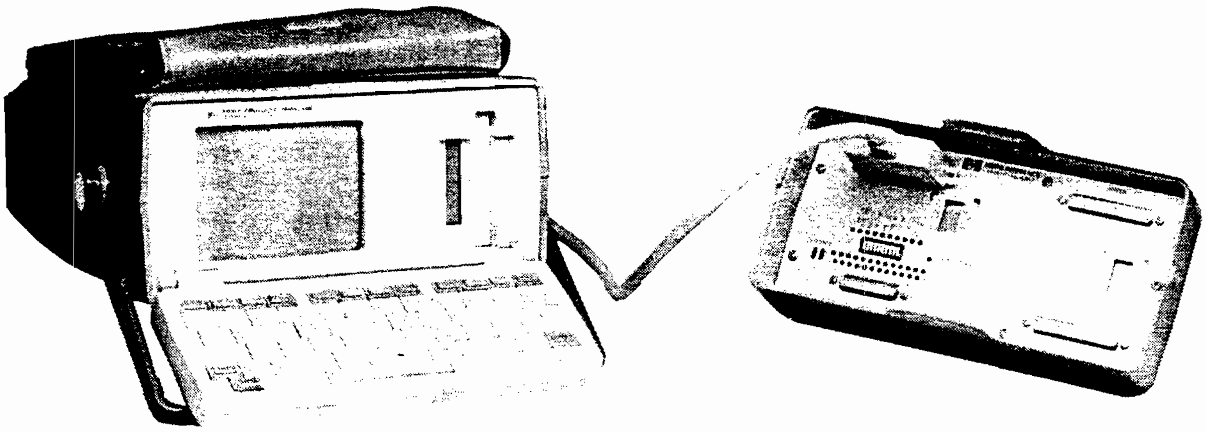


Figure 1-1. The 4951A with Interface Pod

Introduction

The 4951A is a portable, user-friendly protocol analyzer. Some unique features are:

* **Auto Configure.** Automatically determine line parameters and begin monitoring.

* **Post-Processing.** Repeatedly make new measurements on captured data.

* **Softkey guided measurements.** Simplifies setup and programming.

* **63 triggers.** Trap on characters, error conditions, timers, lead changes.

* **5 timers and counters.** Count events and measure time intervals between triggers.

* **Full ASCII keyboard.** Enter all control or hex characters.

* **Nonvolatile memory.** 32 Kbytes for storing data. 8 Kbytes for storing menus and programs.

* **Five display formats.** DTE Only, DCE Only, Two Line (DTE and DCE), Data & State (DTE and DCE with lead transitions), Frame & Packet (decoding of level 2 and 3).

* **Remote.** Transfer data, setups, monitor and simulate menus, timers and counters over a data link.

* **BERT.** Measure bit errors, block errors, and percent error free seconds.

* **Tape Storage (option).** Mass storage of data, setups, programs, and measurements.

HOW TO USE THIS MANUAL

This manual is in two parts. **Part I** tells how to operate the 4951A: Use this part on the first day. **Part II** describes the menus in detail; Use this part after the first day.

PART I GETTING STARTED

Chapter 1 **Learning the Controls --** Describes the 4951A and its controls: Power Up, Front Panel, and the Top Level Menu.

Chapter 2 **The Three Instrument Functions --** Shows you how to perform the three 4951A functions: Monitoring, Simulation, and Bit Error Rate Tests.

Chapter 3 **A Self Demonstration --** You learn by actual operation.

PART II THE MENUS

Chapters 4 through 12 explain each menu in detail by following the order of the **Top Level** softkeys. The Top Level Menu (page 1-9) accesses all instrument functions. To access the Top Level Menu at any time, press EXIT.

MANUAL CONVENTIONS

Softkeys are enclosed by <>. Hardkeys are capitalized.

BEFORE YOU GET STARTED

Initial Inspection. Inspect the analyzer and accessories for any physical damage sustained in transit. Ensure that you have received all the items that should accompany the analyzer (refer to the table of Accessories Supplied). If accessories are missing or if the unit is received in a damaged condition, notify the nearest HP Sales and Support Office and file a claim with the carrier.

Line Voltage Selection. Before connecting operating power, ensure the line voltage selection on the rear panel above the power jack is correct for your area. To change the line voltage selection, pry off the fuse cover from the top. While pressing inward, turn the plastic cam to the correct voltage. Always use a slow-blow fuse of the type indicated on the rear panel. Refer to the Service Manual for further information.

Grounding Requirements. The 4951A is equipped with a three-conductor power cable which, when connected to an appropriate power outlet, grounds the analyzer. To preserve this protection, do not operate the analyzer from a line power outlet that has no ground protection.

Power Cord. The cord packaged with each analyzer depends on its destination. The Service Manual contains a chart of power cord plugs matched to different areas. If the analyzer has the wrong plug for the area, contact your HP Sales and Support Office.

Shipment. Refer to the Service Manual for packaging information. If your analyzer is being returned for service, contact the nearest HP Field Repair Center or Sales and Support office for complete shipping instructions.

POWER UP AND INSTALLATION

TURNING THE 4951A ON

CAUTION: DO NOT PLUG IN THE INSTRUMENT UNTIL YOU ARE SURE THE LINE VOLTAGE SELECTION IS CORRECT.

CONNECTING THE POD

CAUTION: ALWAYS TURN THE INSTRUMENT OFF BEFORE CONNECTING OR DISCONNECTING THE POD.

TURNING THE 4951A OFF

CAUTION: ALWAYS GO TO THE TOP LEVEL MENU BEFORE TURNING THE INSTRUMENT OFF.

The 4951A contains a battery for maintaining current data and menu setups after turn off. However, if you turn off the analyzer at certain times -- eg, during a run -- data or setups may be destroyed. A message to this effect then appears, and the analyzer resets itself automatically. To ensure that menus and setups are saved after turn off, always go to the Top Level Menu before turning the analyzer off. If you do not wish to save the menus and data, press <Reset> in the Top Level Menu to clear the memory and return to default settings. See Chapter 12.

HOOKUP

Hookup directions for monitoring, simulating and BERT are given in Chapter 2 (pages 2-5, 2-11, and 2-16 respectively). BERT hookup is also shown on pages 9-9, 9-10.

FRONT PANEL CONTROLS

KEYBOARD

The 4951A has a full ASCII keyboard. The following keys have special functions:

SOFTKEYS — The six function keys directly under the CRT. The label of each key, displayed at the bottom of the screen, changes for each menu and field. Except when entering keyboard characters, use the softkeys to enter all parameter selections.

CURSOR KEYS — The four arrow keys which move the cursor.

EXIT — Accesses the Top Level Menu. During a run, EXIT is a halt key. In some menus, eg BERT and Mass Store, you must press EXIT twice.

MORE — Accesses additional softkeys whenever more are available.

SHIFT — Selects lower-case characters when pressed with another key.

CNTL — Selects a control character when pressed with another key.

RETURN — Moves the cursor onto the next lower field (same as cursor down).

DISPLAY

The 5" CRT displays 16 lines of 32 characters. **Softkey labels** occupy the bottom two lines.

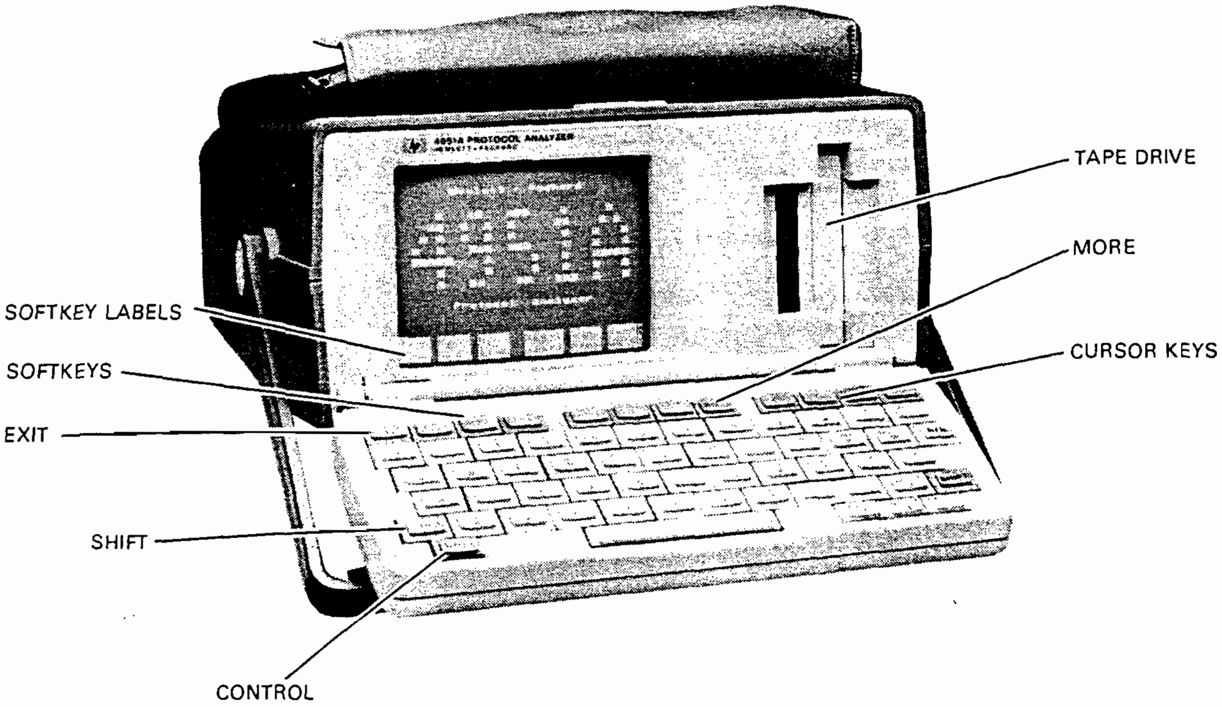


Figure 1-2. Front Panel Controls

THE TOP LEVEL MENU

The Top Level Menu (Figure 1-3) accesses all instrument functions. The Top Level Menu comes up whenever the instrument is turned on. Press EXIT to access the Top Level Menu. Press MORE to see all Top Level softkeys. Top Level softkeys are described below.

AUTO CONF Automatically configure to line parameters (when monitoring on-line).

SETUP Manually configure to line parameters.

MONITOR Select monitoring measurements and triggers.

SIMULATE Select simulation measurements and triggers.

RUN MENU Execute monitoring, simulation, or BERT.

EXAMINE DATA Display the buffer or tape after run-time.

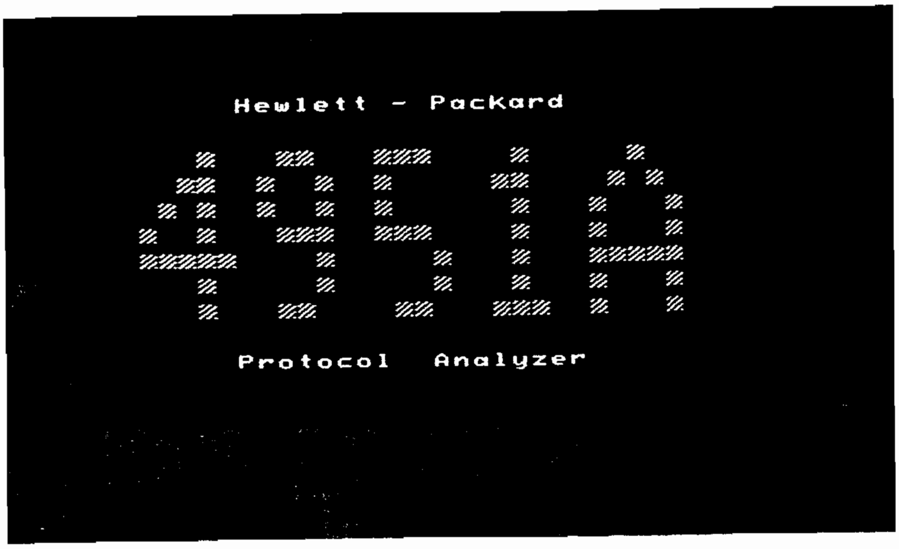
BERT Configure Bit Error Rate Tests.

MASS STORE Control tape functions.

REMOTE Transmit and receive menus and data to another 4951A or a 4955A

RESET Reset all menus to their default conditions and clear the buffer.

SELF TEST Perform self-test procedures.


Hewlett - Packard

4951A

Protocol Analyzer

Auto- Config	Setup	Monitor	Simulate	Run	Examine Data

[MORE]

BERT	Remote	Mass Store		Reset	Self Test

Figure 1-3. The Top Level Menu

CHAPTER 2

THE THREE INSTRUMENT FUNCTIONS

Monitoring
Simulation
Bit Error Rate Tests

Introduction

WHAT'S IN THIS CHAPTER

This chapter describes the three basic functions of the instrument: **Monitoring**, **Simulating**, and **Bit Error Rate Tests**. For more information on these topics, or on any Top Level softkey, go to the chapter by that title.

After looking over this chapter, you may want to go right on to Chapter 3, which leads you through monitoring and simulating exercises with the instrument.

THINGS TO REMEMBER

The Top Level Menu (page 1-8) accesses all other menus. EXIT will access the Top Level Menu at any time.

EXIT acts like a halt key during program execution. EXIT will stop execution and access the Top Level softkeys.

Press MORE to see any additional softkeys in any menu. A small vertical "more" at the lower right of the CRT prompts you whenever there is another softkey set in any menu.

Always go to the Top Level Menu before turning the analyzer off. This will ensure that setups, data, and programs are saved.

Always turn the analyzer off before connecting and disconnecting the pod.

THE THREE INSTRUMENT FUNCTIONS

Monitoring

On-Line. Using Auto Configure or manual setup, you can bridge into most data lines and begin observing the activity.

From Buffer. After having monitored the line for a few moments, you will have captured data in the buffer memory. Once you have data in the buffer, you can repeatedly run any monitoring measurements, just as if you were monitoring on-line. You can also load the buffer from tape. "Post-processing" enables you to try many different measurements on the same data.

Simulation

You can substitute the 4951A for any DTE or DCE on the line, performing measurements while you transmit and receive strings of data characters.

Bit Error Rate Tests (BERT)

This function enables you to determine the data link quality. You can find bit and block errors, error seconds, and percent error free seconds.

MONITORING

The 4951A is a window through which you can observe the activity on a data link. The 4951A lets you monitor either on-line, or from the buffer memory. With the latter capability, you can repeatedly post-process data after a run.

SUMMARY OF MONITORING STEPS (Described on the following pages)

1. Hookup — Bridge the 4951A into line to be monitored.

2. Setup — Using either Auto Configure or the Setup Menus, configure the 4951A to line.

3. Programming — You need make no entries in the Monitor Menu. The 4951A will begin non-selectively displaying line data as soon as you enter the Run Mode. However, you can use the Monitor Menu to analyze a line in depth.

4. Running — Begin monitoring by accessing the Run Menu.

STEP 1 -- HOOKUP

Bridge the 4951A into the line, using the correct pod (eg, RS-232C/V.24) or RS-449) and cables, as shown below. If you already have data in the buffer from a previous run or via tape, you can monitor from buffer; and no pod is necessary. Always turn off the analyzer before connecting or disconnecting the interface pod.

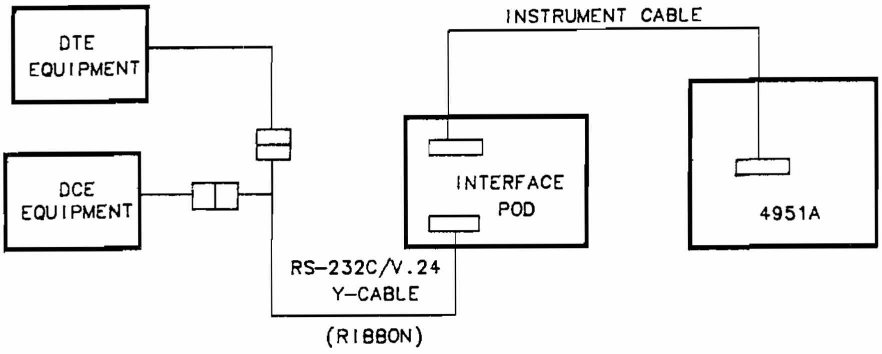


Figure 2-1. Hookup for Line Monitoring

STEP 2 -- SETUP

Prior to monitoring, you must configure the 4951A for the line. You should do this first, because setup selections determine the available choices in other menus.

You can use either Auto Configure or the Setup Menus. Using Auto Configure, the 4951A automatically configures to the line and begins montoring. You may also use the Setup Menus and then go to the Run Menu.

Using Auto Configure for Setup

Merely press <Auto Configure>. The 4951 will identify line parameters and begin monitoring.

To change the **display format**, press EXIT, and go to the Setup Menus. To modify any setup parameters, halt the run by pressing EXIT, and go to the Setup Menus.

In some nonstandard protocols, the 4951A may not be able to identify all the parameters. In that case, halt the run and modify the setup in the Setup Menus. See Chapters 4, 5 and 8.

Using the Setup Menus for Setup

To manually configure, or to modify Auto Configure results, press <Setup> on the Top Level Menu.

1. Select the appropriate protocol.

2. Change any parameter by moving the cursor to that field and pressing the desired softkey.

3. To begin monitoring, press EXIT and then <Run Menu>. Then press <Monitor Line>. The 4951A will begin filling the buffer and displaying data.

What to Do When the Protocol is Nonstandard

If you have trouble configuring the analyzer because of a nonstandard protocol, or because of a defective line, use the following procedure. See chapters 5 and 8.

1. Use Auto Configure to give you a starting point. See chapter 4 for limitations.

2. Change the data code, or other parameters in the appropriate Setup Menu.

3. If some of the line data is still not meaningful, use the Char Async/Sync Setup Menu, as described in chapters 5 and 8.

STEP 3 -- PROGRAMMING THE MONITOR MENU

Optional Entry

You need make no entries in the Monitor Menu. You can go right to the Run Menu and begin monitoring.

Measurements in the Monitor Menu

You need not use the Monitor Menu; but if you wish to analyze the line in depth, the Monitor Menu enables you to use the instrument with great power. See chapter 6 for more information. To access the Monitor Menu, press <Monitor> on the Top Level Menu.

Here's a summary of what the Monitor Menu will do:

Triggering	Define triggers with the <When> statement, enabling you to "look for" up to 63 events simultaneously. The 4951A will branch to any other action upon finding a trigger.
Timing	Five timers measure intervals between triggers with 1 millisecond resolution.
Counting	The 4951A's five counters will each count up to 9999 events.
Conditional Actions	The <If> statement performs actions conditionally, depending on the status of counter, or of a lead at the time of the last trigger.

STEP 4 -- RUNNING THE TEST

In Auto Configure, the 4951A automatically goes into the run mode and begins monitoring.

If you are not using Auto Configure, press <Run Menu> on the Top Level Menu. To monitor on-line, then press <Monitor Line>. If data is already in the buffer from a previous run or via tape, press <Monitor Buffer> to do post processing. Monitoring on-line and monitoring from buffer are essentially the same processes:

1. All counters and timers are reset to zero.
2. Programs in the Monitor Menu -- if any -- begin executing.
3. Buffer data is displayed starting at data block 1.

Halting the Run

Press EXIT to stop the test. The most recent data will be displayed on the screen. (Without halting the test, you can freeze the display by pressing <Stop Display>.)

Changing Display Formats

To choose a different display format, halt the run and change the display format field in the Setup Menu. In HDLC, SDLC, and X.25, five formats are available: DTE, DCE, Two-Line, Data & State, and Frame & Packet. In the BSC and Char Async/Sync Menus, only the first four formats are available. See Chapters 7, and 8.

SIMULATING

The 4951A can take the place of either a DTE or DCE, supplying clocks, data, and error checks in the selected data code and protocol.

SUMMARY OF SIMULATING STEPS (Described on the following pages)

1. Hookup — Substitute the 4951A for either the DTE or DCE.

2. Setup — Use the Setup Menus for simulation.

3. Programming — Unlike Monitoring, you must make some entries in the Simulation Menu:

   a. Select the device to be simulated: DTE or DCE.

   b. Program the interface, using the <Set Lead> softkey.

   c. Transmit the desired characters, using the <Send> softkey.

   d. Use triggers, timers, etc, as in the monitor menu.

4. Running — To begin simulating enter the Run Menu. The ETC clock is automatically provided when simulating a DTE. The TC and RC clocks are automatically provided when simulating a DCE.

STEP 1 -- HOOKUP

Disconnect the line and -- using the correct pod (RS-232C/V.24 or RS-449) and cables as shown below -- substitute the 4951A for the device (DTE or DCE) being simulated. Always turn off the analyzer before connecting or disconnecting the interface pod.

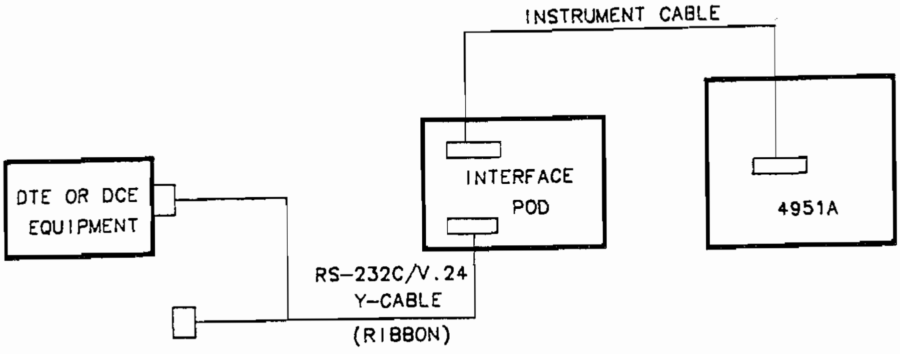


Figure 2-2. Simulation Hookup

STEP 2 -- SETUP

Use the Setup Menus for simulation. Configure the 4951A to your system parameters.

What If You Don't Know the Setup ?

You must know your system in order to simulate. However, you can find some parameters by observing the data line during normal operation.

1. Hook up the 4951A for monitoring.

2. Use Auto Configure to find system parameters.

3. After monitoring for a few moments, go to the Examine Data Menu and look at the buffer data. To see the setup, press the <Timer & Counter> softkey.

4. Auto Configure selects SDLC for bit oriented protocols, and Char Async/Sync for character oriented protocols. You must observe the data in the buffer to find the exact Level 2 and Level 3 protocol on your line.

A Setup Checklist

1. If DTE clock, Bits/sec, and Sync/Async (Char Async/Sync Menu) selections are incorrect, no data can be received or displayed.

2. Protocol, data code, and error checking must be correct to ensure response by the device at the other end.

STEP 3 -- PROGRAMMING THE SIMULATE MENU

Unlike Monitoring, you must make the following three entries in the Simulate Menu.

1. **Select DTE or DCE**

   Determine whether the 4951A is to be a DTE or a DCE.

2. **Handshaking**

   Determine the handshaking requirements on the RS-232C/V.24 or RS-449 leads. Use the <Set Lead> softkey to turn the leads on or off at the desired time. The 4951A normally sets all leads "off" before a test. The device at the other end may not respond if the appropriate control leads are not turned on or off at the proper times.

3. **Transmitting Data**

   Use the <Send> softkey to enter the characters to be transmitted. Otherwise, the 4951A will send only idles. You must know the protocol and polling sequences being used on your line to ensure correct responses.

Other Entries

After satisfying these requirements, you can write further instructions in the Simulate Menu. As in the Monitor Menu, you can set triggers, count events, measure time intervals, etc. See Chapter 6.

STEP 4 -- RUNNING THE TEST

To execute a simulation program, use the Run Menu: Press <Run> on the Top Level Menu and select the test to be run -- in this case <Simulate>. The following will happen:

1. All counters and timers are reset to zero.

2. The 4951A turns on or off the leads you have directed. Observe the pod LCDs for lead activity, or use the Data & State display format.

3. The 4951A, acting like a DTE or DCE, sends out the specified data. Observe the pod LCDs and the display.

4. Line data and lead activity, both from the 4951A and the other transmitting devices, is stored in memory.

5. The display shows the data as it is stored in memory.

Press EXIT to stop the test. The last data loaded into memory will be displayed on the screen. To execute the program again, press <Run Menu> on the Top Level Menu, and then <Simulate>.

BIT ERROR RATE TESTS

Bit error rate tests measure the amount of digital noise on a line: how often are "highs" changed to "lows", and vice versa.

BERT STEPS

1. **Hookup** (see Chapter 9)

   a. End-to-End. Substitute a 4951A for the DTE at both ends of the line. (Figure 2-3).

   b. Loopback. Substitute a 4951A for only one DTE and "loopback" the modem or terminal at the other end of the line. (Figure 2-4).

2. **Setup**

   Press <BERT> on the Top Level Menu and make the appropriate selections.

3. **Running**

   Press <Run Menu> on the Top Level Menu. In the Run Menu, press <BERT>. The 4951A will begin transmitting and receiving; and a data screen will show test status. Press EXIT to halt the test.

"QUICK BROWN FOX" AND STARTUP TESTS

Use the Simulation Menu to perform these tests. See the examples in Appendix F.

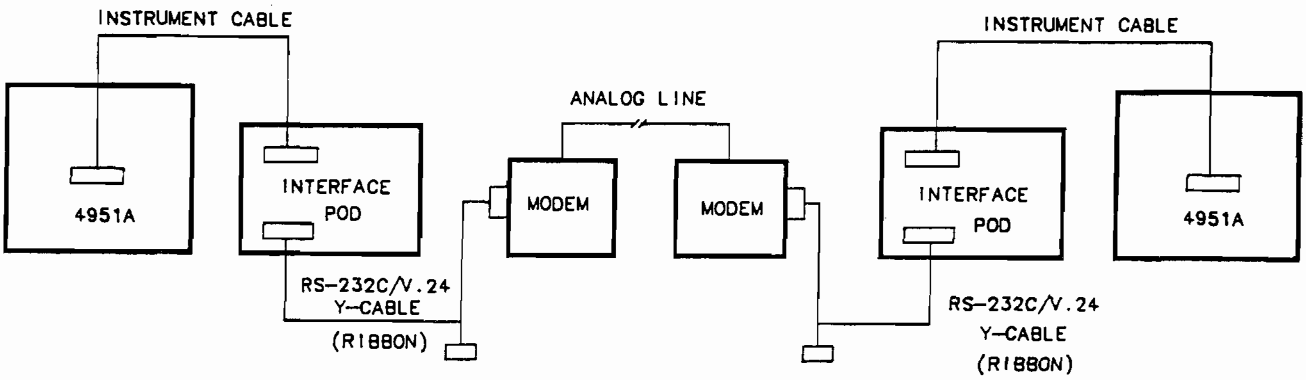


Figure 2-3. End-to-End BERT Hookup

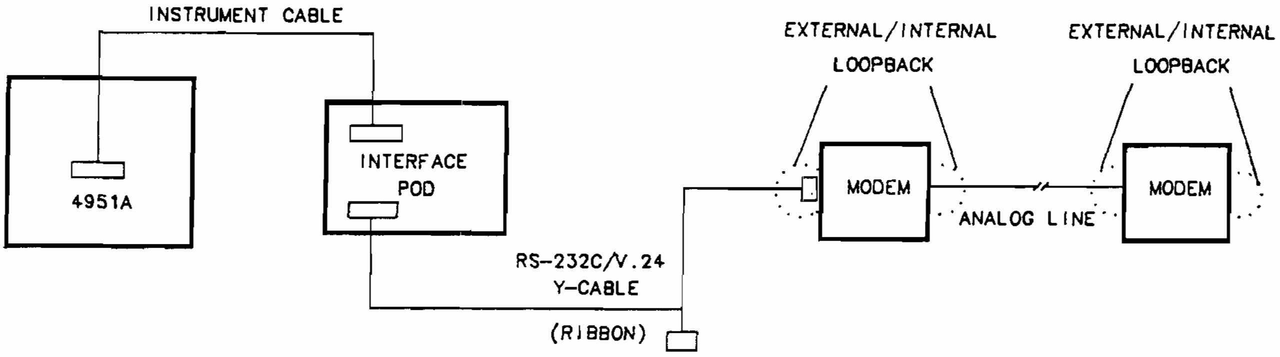


Figure 2-4. Loopback BERT Hookup

CHAPTER 3

A SELF DEMONSTRATION

Introduction

This chapter is for those who learn best by turning knobs and pressing keys. In this chapter you will use the Setup, Monitor, Simulate, and Examine Data Menus.

This chapter is optional. The 4951A with Auto Configure is incredibly easy to operate: Just hook it up to the line, press Auto Configure and begin monitoring. To make any measurement or change any setup, merely press a softkey. The softkey labels prompt you with the next choice. You can't make a mistake.

When the 4951A is connected to the interface pod, store and display all data being sent during simulation. This enables you to fill the buffer even without being connected to a line.

SUMMARY OF STEPS

1. Connect the 4951A to the Interface Pod.
2. Set up the 4951A for protocol, data code, and bit rate.
3. Simulate a DCE: Control the interface leads and transmit strings.
4. Run the simulate program.
5. Observe the captured, looped-back data in the buffer.
6. Run a monitor program from buffer.

STEP 1: HOOK UP TO THE POD

Turn off the 4951A. Connect the analyzer to the interface pod, as shown in figure 3-1 below. Always turn off the analyzer before connecting and disconnecting the pod.

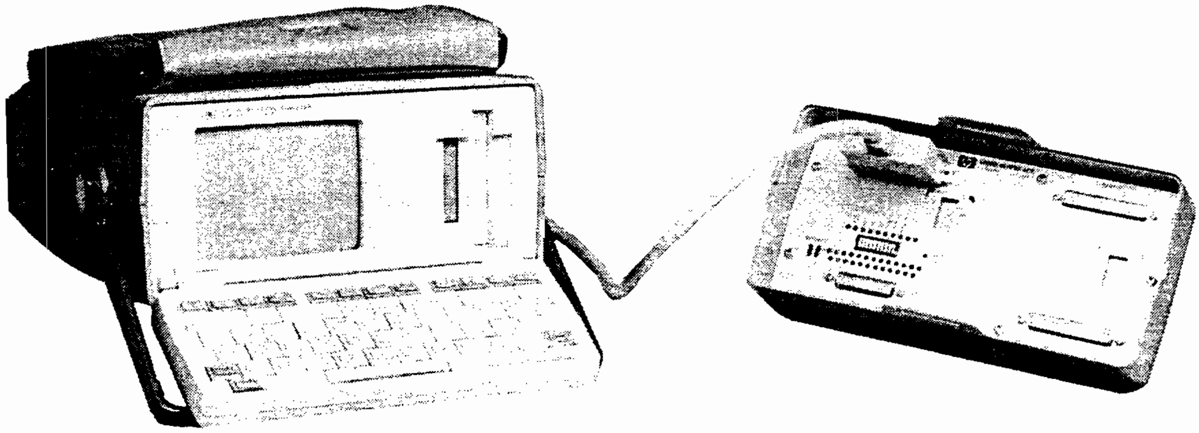


Figure 3-1. Hooking Up the Interface Pod

STEP 2: SETUP

Press <Reset> in the Top Level Menu. This sets all menus to their default parameters. You do not normally need to do this. Most of the time, you will want to take advantage of the 4951A's nonvolatile memory, which saves setups and buffer data after turn-off. We use <Reset> here to ensure that we all start out with the same setup.

Press <Setup> in the Top Level Menu. Select **Char Async/Sync** as the Setup Menu (we could use any protocol for this exercise). Use the cursor and return keys to change fields. Make sure your setup has the following parameter entries:

```
Code:  ASCII 8                    Bits/sec:  1200

Mode:  Synchronous                Display mode:  Data & State

DTE clock:  DCE                   Suppress:  None
```

STEP 3: PROGRAM THE SIMULATE MENU

In this step we program the analyzer to transmit the desired characters and turn on interface leads at the right time. In normal simulation, the correct leads must be set on and off, or the terminal on the other end of the line might not accept the message.

Press <Simulate> in the Top Level Menu. Use the cursor and return keys to change fields. Select **DCE** as the device you will simulate. Enter the following program.

```
Block 1:   Set Lead CD  On
             and then
           Send SY SY SX ABCDE EX SY SY
             and then
           Set Lead CD  Off
             and then
           Wait 100
             and then
           Goto Block 1
```

This program causes the 4951A to repeatedly turn on lead CD and send the character string ABCDE. The 100 millisecond delay makes it easier to see CD transitions.

NOTE: In character oriented protocols, you must explicitly enter the sync characters, such as SY SY, to ensure that the receiving device will accept the message.

STEP 4: RUN THE SIMULATE PROGRAM

Press <Run Menu> in the Top Level Menu, then <Simulate>. The 4951A begins transmitting the character string, internally looping the data back in the pod, and then storing it in the buffer. The data appears as if it were coming from the line. Note the following features:

LCD Indicators

Four interface pod LCD indicators should be blinking: TC, RC, RD, and CD. If we were simulating a DTE (with DTE as clock source), the ETC, TD, and RTS leads would blink if programmed.

DCE and DTE Displays

The Data & State display format shows both DTE and DCE data, as well as timing relationships on four interface leads. DCE data appears in inverse video. DTE data would appear in regular video. Try changing display formats in the Setup Menu.

The <Summary> and <Stop Display> Softkeys

Press <Summary>. Without stopping the run, you can at any time review the setup and observe timer and counter activity. Press <Stop Display>. This freezes the display, but does not halt the run.

Block Numbers

Block numbers increase to 16 and start over at 1. A "block" of memory holds 2 Kbytes of information (data, timing information, and lead status).

STEP 5: OBSERVE THE BUFFER

During a run, data is constantly being loaded into the buffer. Press EXIT to halt the run. Press <Examine Data> in the Top Level Menu to observe the buffer. Note the following features:

Display Format

Go to the Setup Menu at any time to change the display format.

Character Decoding

Move the cursor through the characters. Each character is decoded in binary, hex, and octal, and its parity bit is displayed.

Bit Shifting

Note "shift = 0" at the top of the display. If this were a character-oriented protocol, another softkey, <Bit Shift> would appear. You could then shift bits up to one less than size of the data code (eg, six places in ASCII 7) while observing the change in the characters. This is useful in finding the correct character framing in unknown protocols.

Timer and Counter Display

Press <Timer & Counter>. This shows you the setup and the state of the timers and counters at the end of the run.

STEP 6: RUNNING A MONITOR PROGRAM

Now that you have data in the buffer, you can repeatedly run monitor programs from buffer. Press <Monitor> in the Top Level Menu. Enter the following program:

```
Block 1:    When DCE_A
             then goto Block 2
Block 2:    Start timer 1
            When DCE_E
             then goto Block 3
Block 3:    Stop timer 1
```

This program measures the time interval between the start of a data string and the end.

Note that each timer statement is tied to the <When> trigger statement preceding it. This is the correct way to measure time. Time measurements must be referenced to a specific event with a preceding <When> trigger statement.

Go to the Run Menu and press <Monitor Buffer>. Data will now be displayed, just as if you were running on-line.

In the Examine Data Menu press <Timers & Counters>. Timer 1 should show 26 msec.

OBSERVING THE DTE CHANNEL

Jumpering Channels on the Pod

Up to now you have been able to observe what you are sending on the DCE channel because the 4951A always displays what it is sending.

To observe both channels, you can loop the DCE channel to the DTE channel. Use one of the small jumper wires supplied with the instrument to connect pin TD on the interface pod to pin RD. Press <Run Menu> and then <Simulate>. You should now see DTE data (regular video) mixed with DCE data (inverse video).

The DTE Clock Selection

In the Setup Menu, change the **DCE clock** selection to DTE. Now you will see only DCE data (inverse video). Why? When the DCE supplies the DTE clock, it appears on TC. However, when the DTE supplies its own clock, it appears on ETC. Thus, when you changed the clock selection, the DTE could not find a clock on ETC. Jumper the RC pin to the ETC pin and again run the program.

Remove the RC-ETC jumper. Go back to the Simulate Menu and **Simulate** DTE. In the Setup Menu make sure the DTE clock source is still DTE. Change the lead in the Simulate Menu to RTS and run the program again. The display should now show only the DTE channel. With RD and TD still jumpered, why can't we see the DCE channel? Because there is no DCE clock on RC. Jumper ETC to RC and run the program again.

PART II
THE MENUS

CHAPTER 4

AUTO CONFIGURE

Using Auto Configure

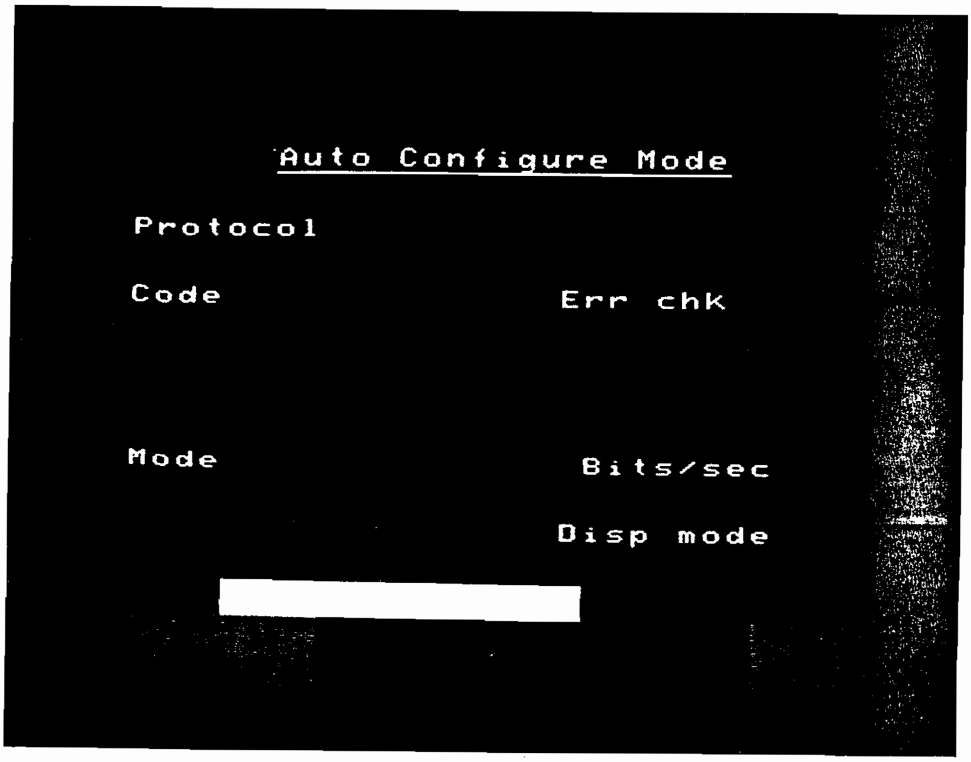
Hook up the analyzer to the line for monitoring (page 2-5). Press the <Auto Configure> softkey on the Top Level Menu. It's as simple as that!

The 4951A will briefly display its parameter selections in either the SDLC or the Char Asyn/Syn Setup Menu. Then it will automatically go into the Run Mode and begin monitoring. You may at anytime press the <Summary> softkey to review the Setup results. To change the **display format**, or any other setup parameter, halt the run by pressing EXIT, and then go back and modify the setup in the Setup Menu. Go to the Run Menu to again start the run.

NOTE: Auto Configure will alter the Setup Menu and the buffer data; so if you need the present setup and buffer data, save them on tape.

Auto Configure as a Starting Point

Auto Configure will work on most lines, with most protocols and data codes. Sometimes, however, there will be nonstandard protocols where Auto Configure is unable to find all the parameters. The Setup Summary that appears before monitoring in Auto Configure will tell you the missing parameters. You can then go to the appropriate Setup Menu and select the correct parameters, using the procedure on page 8-9. If Auto Configure has found all the parameters, but the data does not make sense, try another data code of the same size (eg, substitute ASCII 8 for EBCDIC). Even in the case of nonstandard protocols, Auto Configure will give you a starting point to capture data. See pages 5-20 and 8-9.



Auto Configure Mode

Protocol

Code Err chk

Mode Bits/sec

Disp mode

Figure 4-1. Auto Configure Display (before monitoring)

Auto Configure and Bit Oriented Protocols

The 4951A always selects SDLC for bit oriented protocols. Monitoring will always be correct except in some cases of X.25, or HDLC with extended address and control. In these cases, the selected data code may be incorrect. Follow the procedures below.

Extended Address and Control in HDLC

To observe extended address and control on HDLC lines, go to the setup menu, change the protocol to HDLC, turn on Extended Address and/or Extended Control, and change the display format to Frame & Packet. If the data does not make sense, try another data code.

Decoding Packets in X.25

To decode packet information on X.25 lines, monitor the line for a few moments to capture data in the buffer; or load the data from tape (chapter 11). Then go to the Setup Menu.

Change the protocol to X.25. Change the display format to Frame & Packet. Go to the Examine Data Menu and observe the buffer data. If the data does not make sense, try another data code.

Packet information is automatically decoded in the Examine Data Menu using the Frame & Packet display format. See page 8-7.

Auto Configure and Character Oriented Protocols

Whenever Auto Configure sees a character oriented protocol, whether synchronous or asynchronous, it will select Char Async/Sync Setup. Unlike BSC, Char Async/Sync allows full duplex operation. The user can scan the the Char Async/Sync Setup and try BSC if it looks correct and if he is sure the line is half duplex. The 4951A will find the sync characters, data code, bit rate, etc. of most character oriented protocols.

IPARS -- When Bit Order and Bit Sense are Changed

On standard IPARS protocols, the bit sense is inverted (1's are changed to 0's and vice versa), and the bit order is reversed (MSB is sent first). When Auto Configure recognizes an IPARS protocol, it automatically inverts the bit sense and reverses the bit order before storing the data in buffer memory. Thus, the data can be easily read when it appears on the display.

In some IPARS protocols the data has a different bit order and bit sense than the sync characters. Auto Configure always sets the bit sense and order so the sync characters are 3F 3E on the screen. Thus, Auto Configure will correctly capture and frame the data, but the displayed data may not make sense. Go to the Char Async/Sync Setup Menu and change the bit sense and/or bit order. Then look at the data again.

If one or more of the requirements on page 4-6 are violated, Auto Configure may select IPARS as the data code. If you know that your line does not use IPARS, check that your line data satisfies these requirements.

Auto Configure Requirements

If any of the following assumptions are violated, Auto Configure may not select the parameters.

1. Both data and idle conditions must be present. Asynchronous protocols must have a minimum of two idle characters between messages.

2. A transmit (TC or ETC) clock (x1) must be present for synchronous data. In synchronous NRZI mode, the clock must be encoded with the data.

3. Synchronous character oriented protocols must have the sync character pattern present at least once in a 50-100 character sequence; and the sync pattern must be preceded by two idle characters.

4. Auto Configure requires a variety of alphanumeric, control, and binary characters in the data. There must be sufficient non-repetitive data of different types for Auto Configure to make an identification. For example, if only lower case ASCII characters are sent, EBCD code might be selected.

5. There must be at least one "0" bit preceded and followed by a "1" bit, and one "1" bit preceded and followed by a "0" bit, in a 50-100 character sequence.

6. In bit oriented protocols, there must be at least one good FCS.

7. In bit oriented protocols, at least one frame must be less than 255 characters in length.

Error Messages

No data present: There is no line data. Both data and idle conditions must be present.

No Idles: There are insufficient idles on the line. Both data and idles must be present. Asynchronous protocols must have minimum of two idle characters between messages.

No pod attached: The pod is not attached.

No Sync Characters: Could not find any of the sync characters listed on page 4-9.

Nonstandard Baud Rate: The baud rate is not within 5% of those listed on page 4-9.

Baud rate > 19200 bps: Auto Configure may work at higher rates.

Framing error: Could not find a "1" stop bit in an asynchronous protocol. This error may occur because a transmit clock (TC or ETC) is missing in a synchronous protocol. The 4951A assumes an asynchronous protocol, but cannot then find the stop bit.

Auto Configure Specifications

	BIT ORIENTED (SDLC)	CHARACTER (Synchronous)	CHARACTER (Asynchronous)
Mode	Sync, NRZI		Async (1 stop bit needed)
Code	ASCII 8, EBCDIC	ASCII 7, ASCII 8, EBCDIC, Hex,6,7,8; IPARS (0 idle), IPARS (1 idle), Transcode	ASCII 7, ASCII 8, EBCDIC, EBCD, Baudot
Parity		None, Odd, Even, Ignore	None, Odd, Even, Ignore
Err Chk	CRC-CCITT	None, CRC-6, CRC-12, CRC-16, LRC, (IPARS: CRC-6 only) (Hex: no error checking)	None, CRC-6, CRC-12, CRC-16, LRC
DTE Clock Source	DTE, DCE	DTE, DCE	

Auto Configure Specifications (cont)

	BIT ORIENTED (SDLC)	CHARACTER (Synchronous)	CHARACTER (Asynchronous)
Speed	(Within +/- 5 % , NRZI within +/- 0.5%)		
	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, *2000, 2400, 3200, 3600, 4800, 7200, 9600, *12k, 14.4k, *16k, 19.2k (* not NRZI)	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3200, 3600, 4800, 7200, 9600, 12k, 14.4k, 16k, 19.2k	50, 75, 110, 134.5, 150, 300 600, 1200, 1800 2000, 2400, 3200 3600, 4800, 7200, 9600, 19.2k
Sync Chars	Flags (7E)	EBCDIC: 32 32 ASCII: 16 16 IPARS: 3F 3E Transcode: 3A 3A Hex: LSB of sync char must = 0 and both sync chars must be the same	
Tranparent Text		EBCDIC: DL (10) ASCII: DL (10) Transcode: DL (1F) None	(Same as Synchronous)

Auto Configure Specifications (cont)

	BIT ORIENTED (SDLC)	CHARACTER (Synchronous)	CHARACTER (Asynchronous)
Start BCC		EBCDIC: SX (02) or SH (01) ASCII: SX (02) or SH (01) Transcode: SX (0A) or SH (00)	(Same as Synchronous)
Stop BCC		EBCDIC: EX (03) or EB (26) ASCII: EX (03) or EB (17) Transcode: EX (2E) or EB (0F) Will not support ITB	(Same as Synchronous)
Bit Order	LSB 1st	LSB 1st, IPARS: MSB 1st	LSB 1st
Bit Sense	Normal	Normal, IPARS: Inverted	Normal
Idle Char	7E	FF, IPARS: FF or 00	FF

CHAPTER 5

THE SETUP MENUS

Introduction

HOW SETUP CONTROLS OTHER MENU SELECTIONS

Setup, whether performed manually or via Auto Configure, determines some choices in the other menus. For example, error checking is performed during monitoring according to the current setup. And the appropriate error checking characters are automatically appended to Send strings. Data is displayed in the Examine Data or Run Menus according to the current setup. See Chapter 6.

THE FIVE SETUP MENUS

Press <Setup> on the Top Level Menu to access the Setup Menus. Move the cursor to the Protocol field and select one of the following:

HDLC (bit oriented) Allows extended address and control fields.

SDLC (bit oriented) Allows NRZI synchronizing. This setup is always selected when **Auto Configure** recognizes a bit oriented protocol.

X.25 (bit oriented) Packet information is decoded in the Examine Data Menu

BSC (character oriented) Supports half duplex, character oriented BSC.

CHAR ASYNC/SYNC May be used to configure to most protocols, whether bit or character oriented. This setup is always selected when **Auto Configure** recognizes a character oriented protocol.

WHEN TO USE THE SETUP MENUS

When monitoring on-line, Auto Configure automatically configures the 4951A to most lines. You may, of course, use the Setup Menus to manually configure. Generally, however, use the Setup Menus for the following:

1. MONITORING FROM BUFFER. For post-processing, use the setup menus. You can of course, use an "auto configured" setup from a previous run. Setups remain even after power off, unless you press <Reset>.

2. CHANGING DISPLAY FORMATS. Auto Configure always uses the display format currently selected in the Setup Menu. Use the Setup Menus to change formats.

3. SUPPLEMENTING AUTO CONFIGURE. Use the Setup Menus to modify any parameters after initial synchronizing with Auto Configure.

4. SIMULATING.

SAVING SETUPS

Nonvolatile Memory. If you turn off the power only while in the Top Level Menu, menu setups and buffer data are saved. Always turn the power off in the Top Level Menu. Otherwise, some settings may be destroyed.

Tape. You can store Menus, or both Menus & Data, to tape. All menus except BERT are saved. See chapter 11.

THE BIT ORIENTED MENUS

The three Bit Oriented Setup Menus are HDLC, SDLC, and X.25. Press <Setup> on the Top Level Menu. Move the cursor to the protocol field and select HDLC, SDLC, or X.25.

To decode frames in bit oriented protocols, use Frame & Packet display mode. To decode X.25 packets, use the Examine Data Menu in frame & packet format.

The three Bit Oriented Setup Menus, with their softkey options, are shown on page 5-6. Asterisks indicate differences between the three menus. The default parameter selections, which appear after <Reset>, are listed in chapter 12.

In Bit Oriented Setup, the 4951A performs automatic zero bit insertion/extraction.

HDLC and SDLC

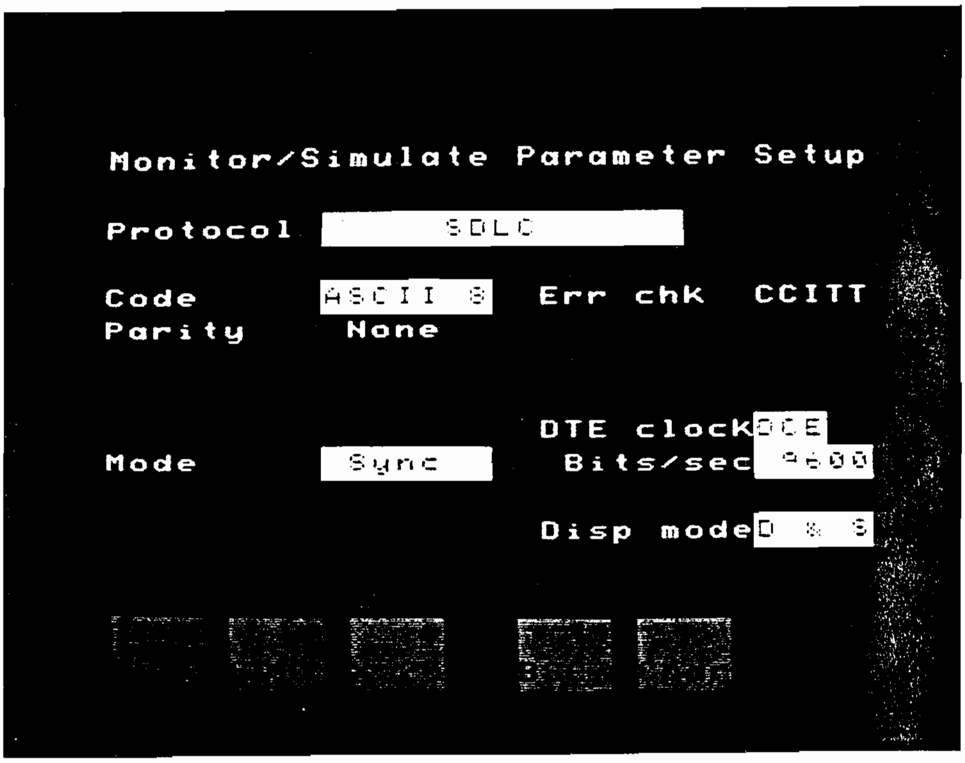
Except for the following differences, HDLC and SDLC have the same format.

1. HDLC allows Extended Address and Control fields.

2. SDLC allows either normal Sync Mode or NRZI Sync.

X.25

X.25 is the same as HDLC except for allowing ISO Level 3 (network) data to be placed in the information field of Information Frames. Use the X.25 menu when monitoring or simulating X.25 lines. The Examine Data Menu decodes packets in Frame & Packet display format. See Chapter 8.



```
Monitor/Simulate Parameter Setup

Protocol          SDLC

Code        ASCII 8     Err chk   CCITT
Parity       None


                        DTE clockDCE
Mode         Sync        Bits/sec 9600

                        Disp modeD & S
```

Figure 5-1. SDLC Setup Menu

BIT ORIENTED PROTOCOL SETUP MENUS

(* used only in HDLC)
(** used only in SDLC)

```
 Protocol  [ HDLC ]
           [ SDLC ]
           [ X.25 ]

* Ext Addr [ Off ]                     * Ext Ctrl  [ Off ]
           [ On  ]                                 [ On  ]

 Code  [ ASCII 8 ]                       Err chk     CCITT
       [ Hex 8   ]
       [ EBCDIC  ]
                                         DTE clock [ DCE ]
 Parity      None                                  [ DTE ]

 Mode        Sync                        Bits/sec  [ 19200 ]  [  3600 ]  [    150 ]  [ 16000 ]
      ** [ Sync NRZI ]                             [  9600 ]  [  3200 ]  [  134.5 ]  [ 14400 ]
                                                   [  7200 ]  [  2000 ]  [    110 ]  [ 12000 ]
                                                   [  4800 ]  [  1800 ]  [     75 ]
                                                   [  2400 ]  [   600 ]  [     50 ]
                                                   [  1200 ]  [   300 ]  [Teletext]

                                         Disp mode [ Two Line ]  [   Data & State ]
                                                   [ DTE only ]  [ Frame & Packet ]
                                                   [ DCE only ]
```

BIT ORIENTED MENU DEFINITIONS

EXT ADDR (HDLC) HDLC allows an extended address field. When an additional address octet is to follow, the first or least significant bit of the address octet is set to 0. The last address octet in a series has the LSB set to 1.

EXT CTRL (HDLC) HDLC allows a 16-bit control field to handle larger N(s) and N(r) counts.

CODE The bit oriented menus allow ASCII 8, EBCDIC or, using Hex 8, any 8-bit code.

MODE All bit-oriented protocols are synchronous: the data is transmitted with a clock. In **NRZI** (SDLC only) the clock is encoded within the data.

DTE CLOCK DTE data can be synchronized to either a DCE or DTE clock. If this selection is incorrect, only DCE data will be displayed.

DISP MODE All five display formats are available for the bit oriented menus. The Frame & Packet format decodes all control field bits. In addition, when viewing the buffer in this format, packet information is also decoded. See Chapters 7 and 8 for examples of the different types of displays.

BITS/SEC Except for NRZI, all the selections shown on page 5-6 are supported. NRZI may not work at 16000, 12000, and 2000 bps. In **Teletext**, the DTE sends at 75 bps, and the DCE sends at 1200 bps.

HINTS FOR SETTING UP BIT ORIENTED LINES

OBSERVE THE POD

The DTE and DCE lines on the pod LCDs should be flashing. Except in the case of NRZI sync, there should also be clock activity.

USE AUTO CONFIGURE

Use Auto Configure for initial synchronizing. If the data is bit oriented, Auto Configure will always select SDLC as the protocol. You will need to change protocols in the following two cases.

1. **HDLC with Extended Address or Control.** Change the protocol to **HDLC** with the following setup:

```
Ext Addr and/or Ext Ctrl:  On                    Disp mode:  Frame & Packet
```

2. **X.25 Packets.** If the protocol is X.25, change the setup to the following. After capturing data, use the Examine Data Menu to observe packet decoding.

```
Protocol:  X.25                                  Disp mode:  Frame & Packet
```

CHOOSING DISPLAY FORMATS

In the bit oriented menus, you can use any of the five display formats. For frame decoding, use Frame & Packet, as described on the following pages.

DECODING FRAMES WITH FRAME AND PACKET DISPLAY FORMAT

During run-time, the Frame & Packet display format decodes Level 2 frame information in HDLC, SDLC, or X.25. The frame information described below is decoded. See figure 5-2.

After run-time, for HDLC or SDLC, the Frame & Packet display looks the same when observing the buffer in the Examine Data Menu, except that up to 57 data characters can be shown at the top of the display. See figure 5-3.

ADDRESS Hex address of the secondary channel. (Extended addresses can be seen when HDLC with extended address is being used).

TYPE Identifies the type of frame from the Control Field.

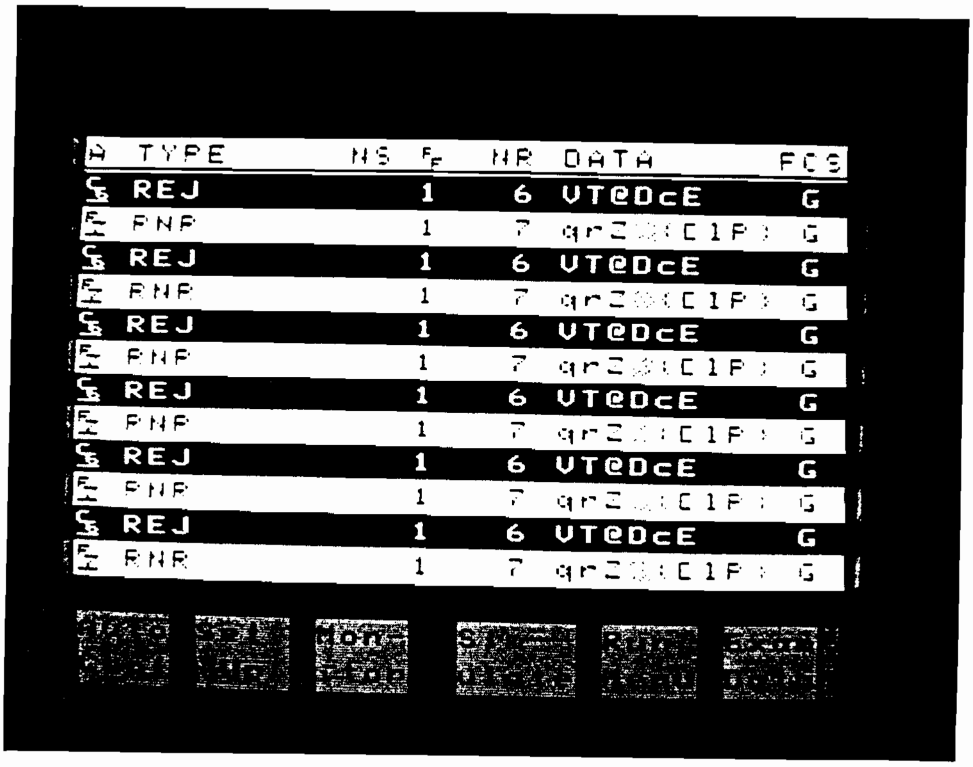
N(S) Send Sequence Number of the frame. (Normally modulo 8; but becomes modulo 128 when HDLC with extended control is being used).

P/F Poll/Final Bit. Poll = 1, Final = 0.

N(R) Receive Sequence Number of the frame. (Normally modulo 8; but becomes modulo 128 when HDLC with extended control is being used).

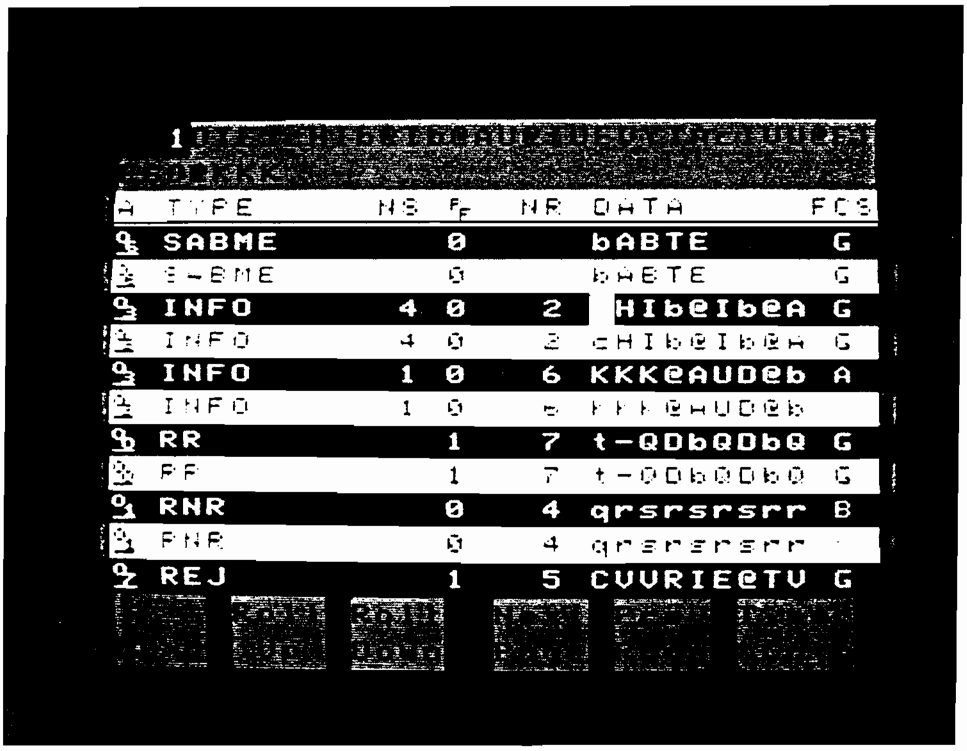
Data Displays the first nine characters of the information field.

FCS Indicates the status of the Frame Check Sequence (CRC-CCITT) as either good (GG), bad (BB), or indicates an aborted frame (AA).



```
A TYPE          NS  FF   NR  DATA          FCS
  REJ                1    6  UT@DcE         G
  PNP                1    7  qrZ :[1P:      G
  REJ                1    6  UT@DcE         G
  RNR                1    7  qrZ :[1P:      G
  REJ                1    6  UT@DcE         G
  RNP                1    7  qrZ :[1P:      G
  REJ                1    6  UT@DcE         G
  PNP                1    7  qrZ :[1P:      G
  REJ                1    6  UT@DcE         G
  PNR                1    7  qrZ :[1P:      G
  REJ                1    6  UT@DcE         G
  RNR                1    7  qrZ :[1P:      G
```

Figure 5-2. HDLC in Frame & Packet Display Format (run-time)



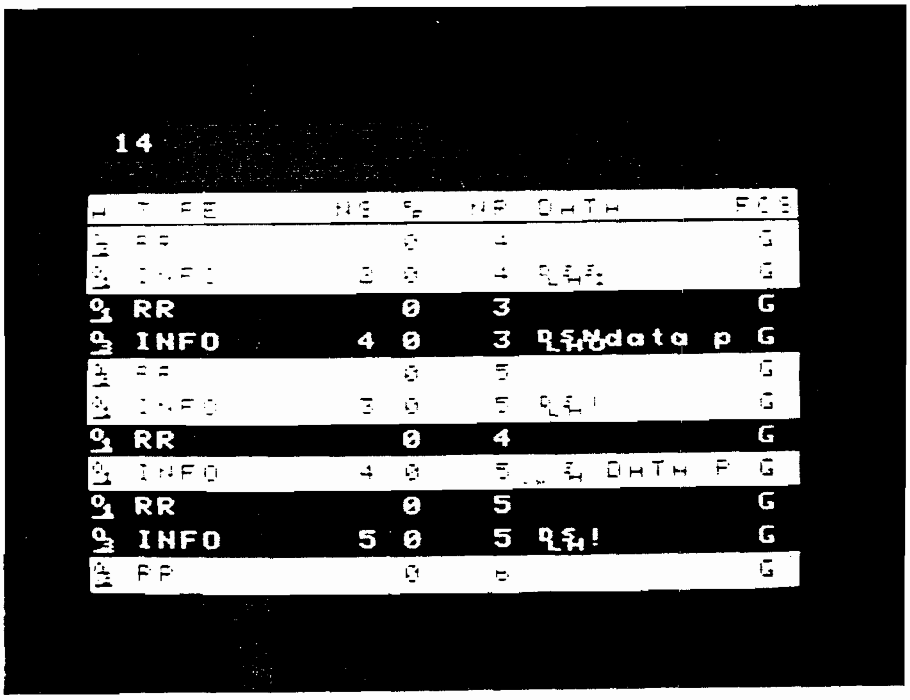
```
1
A TYPE          NS  FF   NR  DATA          FCS
  SABME              0       bABTE          G
  S-BME              0       bABTE          G
  INFO           4   0    2   HIb@Ib@A      G
  INFO           4   0    2  cHIb@Ib@A      G
  INFO           1   0    6  KKK@AUD@b      A
  INFO           1   0    6  KKK@AUD@b
  RR                 1    7  t-QDbQDbQ      G
  FF                 1    7  t-QDbQDbQ      G
  RNR                0    4  qrsrsrsrr      B
  PNR                0    4  qrsrsrsrr
  REJ                1    5  CUURIE@TU      G
```

Figure 5-3. HDLC in Frame & Packet Display Format (Examine Data Menu)

DECODING X.25 PACKETS IN THE EXAMINE DATA MENU

X.25 looks the same as HDLC or SDLC when running in the frame and packet display format. After capturing data, however, the Examine Data Menu shows both frame decoding and packet decoding. As shown on the next page, packet information at the cursor location is decoded at the top of the display. The DTE and DCE columns contain the following information.

Q-Bit	Qualifier Bit.
D-Bit	Delivery Confirmation Bit.
MOD	Modulo 8 or 128.
LCN	Logical Channel Number.
TYPE	Type of packet.
P(S)	Packet Send Sequence Number.
M-Bit	More Data Mark.
P(R)	Packet Receive Sequence Number.
Data	Displays the first five characters of the data field.



```
14
                          NS  PF  NR  DATA          FCS
     RR                        0   4                  G
     INFO                  3   0   4                  G
01   RR                        0   3                  G
03   INFO                  4   0   3      data  p     G
     RR                        0   5                  G
     INFO                  3   0   5                  G
01   RR                        0   4                  G
     INFO                  4   0   5        DATA  P   G
01   RR                        0   5                  G
03   INFO                  5   0   5   !              G
     PP                        0                      G
```

Figure 5-4. X.25 in Frame & Packet Display Format (Examine Data Menu)

THE BSC MENU

BSC MENU DEFINITIONS

The BSC Menu and available softkey selections are shown on the following page.

PARITY — Except for ASCII 7, which uses odd parity, there is no parity check. In simulate mode, the 4951A will transmit BSC with the correct parity; and in monitor mode, the 4951A will expect to see BSC data with the correct parity.

MODE — BSC is synchronous: the data is transmitted with a clock.

SYNC ON — The 4951A automatically chooses the correct sync characters for each data code. The sync characters are: 32 32 (EBCDIC), 16 16 (ASCII), or 3A 3A (Transcode). The 4951A requires at least two sync characters for proper framing.

ERR CHK — Select LRC or CRC-16 for ASCII or EBCDIC, and LRC or CRC-12 for SixBit Transcode.

BITS/SEC — The bit rates for BSC are the same as the bit oriented protocols.

DISP MODE — Frame & Packet display format is not used in BSC.

SUPPRESS — The BSC Menu allows you to suppress almost any combination of text and control characters from the display. Suppressed characters are not deleted from the buffer. Note that idle characters are assumed to be FF in BSC.

BSC SETUP MENU

```
Protocol  [ BSC ]

Code   [ ASCII 7   ]                        Err chk     [ LRC      ]
       [ Transcode ]                                    [ CRC 12 ]
       [ EBCDIC    ]                                    [ CRC 16 ]

Parity    Odd  (ASCII 7)                    DTE clock   [ DCE ]
          None (Transcode)                              [ DTE ]
          None (EBCDIC)
                                            Bits/sec    [ 19200 ]  [ 3600 ]  [    150 ]  [ 16000 ]
                                                        [  9600 ]  [ 3200 ]  [  134.5 ]  [ 14400 ]
                                                        [  7200 ]  [ 2000 ]  [    110 ]  [ 12000 ]
Sync on    16  16  (ASCII 7)                            [  4800 ]  [ 1800 ]  [     75 ]
           3A  3A  (Transcode)                          [  2400 ]  [  600 ]  [     50 ]
           32  32  (EBCDIC)                             [  1200 ]  [  300 ]  [Teletext]

                                            Disp mode   [ Two Line ]  [      DCE Only ]
                                                        [ DTE Only ]  [ Data & State ]

                                            Suppress    [ None         ]  [ Idle & Ctl    ]
                                                        [ Idle         ]  [ Idle & Txt    ]
                                                        [ Null         ]  [ Null & Ctl    ]
                                                        [ Control      ]  [ Null & Txt    ]
                                                        [ Text         ]  [ Id & Nu & Ctl ]
                                                        [ Idles & Null ]  [ Id & Nu & Txt ]
```

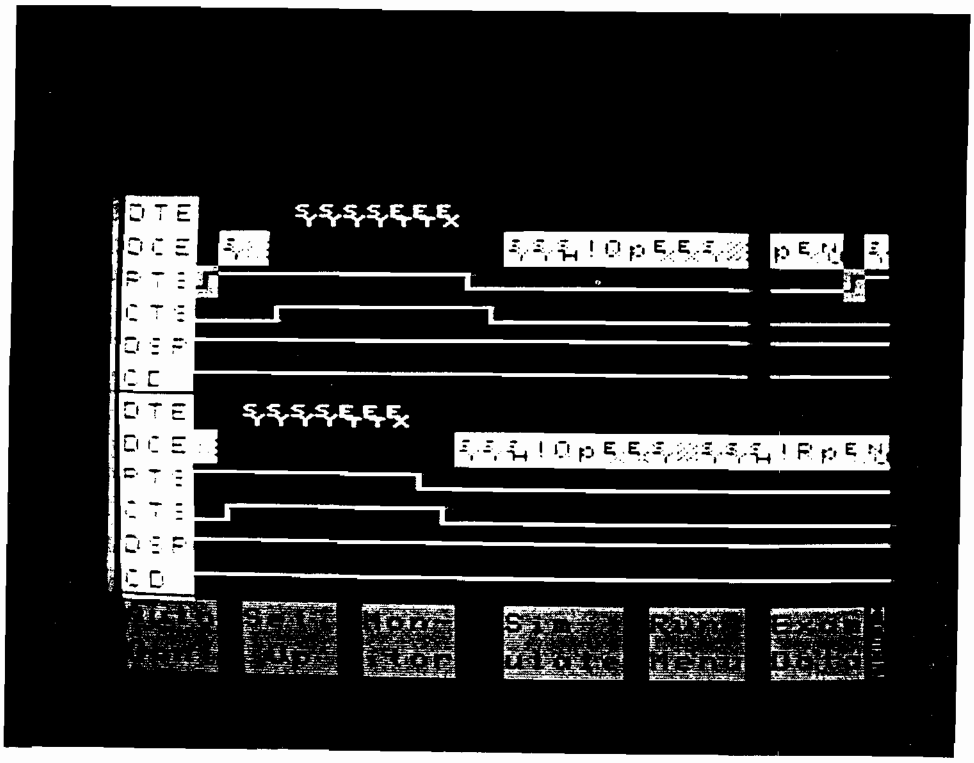


Figure 5-5. BSC in Data & State Display Format

THE CHAR ASYNC/SYNC MENU

Shown on the next page is the Char Async/Sync Menu and its softkey selections. This Menu is the most versatile of all the setup menus. You can use it to capture most protocols: synchronous or asynchronous, character or bit oriented.

TAILORING THE 4951A TO ANY DATA CODE

Note the large number of codes available with this menu. In the Char Async/Sync Menu, you select all the parameters to go with your data code. This menu enables you to tailor the analyzer to many different codes with different characteristics.

Note: The 4951A does not perform zero bit insertion or extraction for bit oriented protocols when in Char Async/Sync setup.

WHEN TO USE THE CHAR ASYNC/SYNC MENU

1. For asynchronous lines.

2. For character oriented protocols other than half-duplex BSC.

3. To see all line activity, including line idles. (See page 5-20).

4. For nonstandard protocols.

5. For most protocols: character or bit oriented, synchronous or asynchronous.

CHAR ASYNC/SYNC SETUP MENU (* indicates synchronous mode only)

```
Bit order  [ LSB first ] [ MSB first ]              Bit sense  [ Normal ] [ Inverted ]

Code  [ ASCII 8 ] [ Hex 6     ] [ IPARS idle 0]     Err    [ None ] [ CRC 6  ] [CRC 16 ]
      [ Hex 8   ] [ EBCDIC    ] [ IPARS idle 1]      chk   [  LRC ] [ CRC 12 ]
      [ ASCII 7 ] [ Transcode ] [ Baudot   ]
      [ Hex 7   ] [ Hex 5     ] [ EBCD ]            * Start on/Stop on  [ Use keyboard ]

Parity [ None ] [ Even ] [ Odd ] [ Ignore ]         * DTE clock  [ DCE ] [ DTE ]

Transparent  [         None ]                       Bits/sec  [ 19200 ] [ 3600 ] [   150 ] [ 16000 ]
  text char  [ Use keyboard ]                                 [  9600 ] [ 3200 ] [ 134.5 ] [ 14400 ]
                                                              [  7200 ] [ 2000 ] [   110 ] [ 12000 ]
Mode     [ Asyn 1   ]  [ Asyn 2 ]                             [  4800 ] [ 1800 ] [    75 ]
         [ Asyn 1.5 ]  [ Sync   ]                             [  2400 ] [  600 ] [    50 ]
                                                              [  1200 ] [  300 ] [Teletxt]

* Sync on  [         Idles ]                      Disp mode  [ Two Line ]  [     DCE Only ]
           [ Use keyboard ]                                  [ DTE Only ]  [ Data & State ]

* Drop  sync  [ Use keyboard ]  chrs              Suppress   [ None          ]  [ Idle & Ctl    ]
   after  [ Use keyboard ] [ None ]                          [ Idle          ]  [ Idle & Txt    ]
                                                             [ Null          ]  [ Null & Ctl    ]
                                                             [ Control       ]  [ Null & Txt    ]
                                                             [ Text          ]  [ Id & Nu & Ctl ]
                                                             [ Idles & Null  ]  [ Id & Nu & Txt ]
```

CHAR ASYNC/SYNC DEFINITIONS

BIT ORDER/SENSE — Normally, the LSB is sent first, and data is not inverted. Some protocols (eg, IPARS) may be different. These selections affect only incoming and outgoing run-time data. Incoming data is changed at the input interface before processing. When simulating, data is changed at the output interface. Buffer data is not changed.

START ON/STOP ON — Determines error checking bounds. Error checking starts on the character after the **Start On** character; however, the **Stop On** character is included in the BCC. This selection does not appear if **Error chk** is None.

TRANSPARENT TEXT — This character delimits the boundaries of a field, outside of which all control characters are to be treated as data.

MODE — Synchronous, or Asynchronous (1, 1.5, 2 stop bits). Stop bit selection is not necessary when monitoring asynchronous protocols, since the 4951A needs only one stop bit, even if more are present.

SYNC ON — Synchronous mode only. Selects the sync characters for proper framing. The 4951A requires at least two sync characters.

DROP SYNC AFTER — Synchronous mode only. Tells the analyzer to "drop" sync (stop bringing in data) and start looking for sync characters again.

DTE CLOCK — Synchronous mode only. Specifies the DTE transmit clock source.

Hints For Using the Char Async/Sync Menu

FINDING THE CORRECT SYNC CHARACTERS (Synchronous mode only)

Selecting the "Sync on" Characters

Sync on determines character synchronization: the analyzer searches incoming data for the two specified sync characters. Unless the analyzer can find the sync characters, it will not load line data.

To load line data for study when you do not know the sync character, select **Sync on** Idles. NOTE: The 4951A assumes that all character oriented protocols idle in FF. If your line uses some other condition, you must **Sync on** that condition.

To store all data, including idles, enter **Drop sync** 0 **chrs after** None. This causes the analyzer to never drop sync, and to bring in all data including idles (see page 5-21).

NOTE: Normally, the 4951A does not store idles. This is to prevent the buffer from being filled with nothing but idles.

STORING THE DATA

If you do not know the data code, try an 8-bit code first. Select no parity and no error checking.

After making the above selections in the Char Async/Sync Menu, go to the Run Menu and <Monitor from Line> for a few moments to fill the buffer with data for study. Then go to the Examine Data Menu to view the data in buffer.

FINDING THE CORRECT FRAMING

Bit Shifting (BSC and Char Async/Sync only)

Even if you do succeed in bringing in data by synchronizing on idles, the displayed information will probably be meaningless because of incorrect framing. To make the data meaningful, go to the Examine Data Menu and <Bit Shift> the captured data.

NOTE: The 4951A does not shift through the parity bit. Unless you use a code with no parity (see figure 5-4), you must use trial and error to find the correct framing.

If at least part of the data still does not become meaningful while bit shifting, change the data code to another without parity. When the data becomes meaningful, you will be able to determine the correct sync characters. Change the **Sync on** selection to these characters.

ELIMINATING SUPERFLUOUS DATA

Dropping Sync (Synchronous mode only)

Once you find the correct framing through the above procedure, you can eliminate idles. Otherwise the buffer will be mostly filled with idles. To eliminate idles, enter **Drop sync** 0 **chars after** FF. If the line idles in a character other than FF, enter that character instead.

Drop sync after determines where the analyzer "drops sync" and again begins looking for the **Sync on** condition. **Drop sync** 5 **chrs after** 3F tells the analyzer to drop synchronization 5 characters after it finds 3F. Once the analyzer drops sync, it cannot load data until it again finds the specified sync characters.

SELECTING "DROP SYNC" CHARACTERS

"Within Text" and "Outside Text" Characters

You may specify up to seven "ORed" hex characters for the **after** selection. For example, you can enter: **Drop sync** 5 **chrs after** FF 3F 7F 16 32 FF 37. The first selection specifies a "within text" character; the next six selections are for characters "outside text".

"Within text" characters are those within the error checking bounds defined by the **Start on** and **Stop on** selectons. Error checking starts on the character after the **Start on** character; however, the specified **Stop on** character is included in the error checking. For example, assume the data is SY SY SX AB BC EX SY SY. If the **Start on** character is SX and the **Stop on** character is EX, error checking will start on AB and end on EX.

NOTE: "Within text" is defined only when error checking (**Err chk**) is selected. **Start on** and **Stop on**, which then appear, delimit error checking bounds. When **Err chk** is None, all text is "outside".

How to Select "Drop sync" Characters.

For example, assume the data is: SY SY SX AB 3C 4D 13 EX 3A 4B. Assume also that error checking is to **Start on** SX and to **Stop on** EX. If you specified: **Drop sync** 0 **chrs after** AB 3A FF FF FF FF FF, the analyzer would drop sync on character AB.

Using the same data and error checking selection, if you specified: **Drop sync** 1 **chrs after** 3A BC CD CD BF AF 4F, the analyzer would never drop sync because 3A does not occur "within text".

Data Code	No Parity	Even or Odd Parity	Ignore Parity
Hex 5 Baudot	5 bits (no parity bit)	6 bits (including parity bit)	6 bits * (parity bit = 0)
Hex 6 EBCD IPARS Transcode	6 bits (no parity bit)	7 bits (including parity bit)	7 bits * (parity bit = 0)
Hex 7 ASCII 7	7 bits (no parity bit)	8 bits (including parity bit)	8 bits * (parity bit = 0)
Hex 8 ASCII 8 EBCDIC	8 bits (no parity bit)	9 bits (including parity bit)	9 bits * (parity bit = odd)

Figure 5-4. Character Frame Sizes vs Data Code (* these settings are forced in Simulate)

CHAPTER 6

THE MONITOR AND SIMULATE MENUS

Introduction

WHAT'S IN THIS CHAPTER

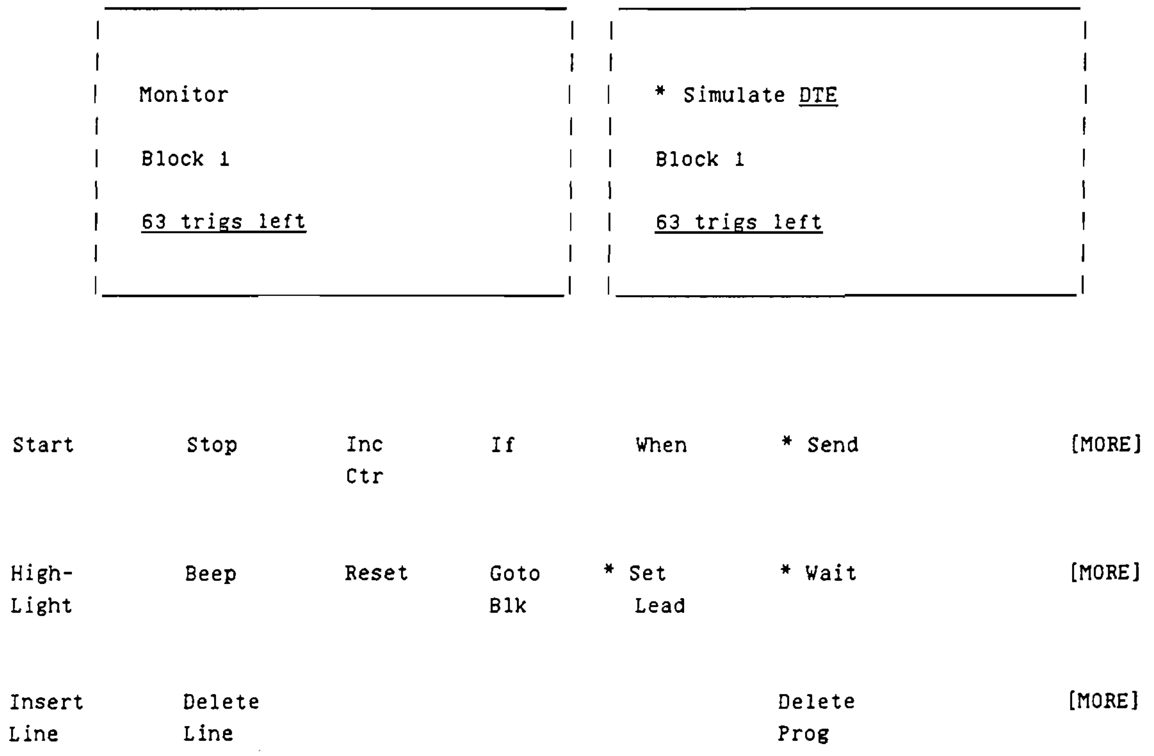
This chapter shows you how to make measurements using the Monitor/Simulate Menus. Since the 4951A uses triggering to perform all measurements, this chapter tells you how to tie your programs to trigger statements.

DIFFERENCES BETWEEN MONITOR AND SIMULATE MENUS

Monitoring has no effect on the line: it is passive and non-interactive. Simulation is active: the 4951A takes the place of a DTE or DCE on the data line. This basic distinction, active vs passive, is reflected in the difference between the two menus.

There are five differences between the Monitor and Simulate Menus:

1. In Simulate, you must specify either DTE or DCE simulation.
2. In Simulate Menu, you can transmit characters with the <Send> softkey.
3. In Simulate, you must program the interface with the <Set Lead> softkey.
4. In Simulate, you can delay output (Send, Set Lead) with the <Wait> statement.
5. In Simulate, clocks are automatically provided on the interface: ETC is provided when simulating a DTE; TC and RC are provided when simulating a DCE.



```
Monitor                         * Simulate DTE

Block 1                         Block 1

63 trigs left                   63 trigs left


Start    Stop     Inc    If     When     * Send     [MORE]
                  Ctr

High-    Beep     Reset  Goto   * Set    * Wait     [MORE]
Light                    Blk      Lead

Insert   Delete                          Delete     [MORE]
Line     Line                            Prog
```

F7igure 6-1. The Monitor and Simulate Menus (* Indicates Simulate only)

PROGRAMMING

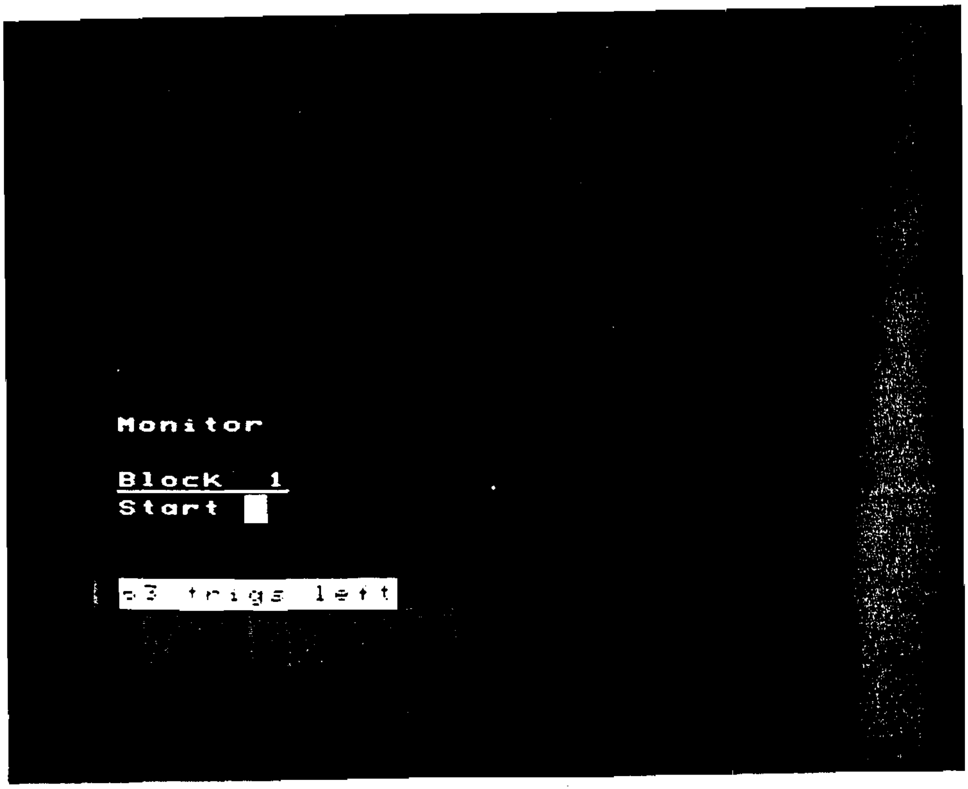
Always Do Setup First. If you change the Setup Menu after you've already developed a program, or change DTE to DCE (or vice versa) within a program, the program may have **blinking entry fields**, indicating those entries are inappropriate for the Setup in effect. Your program may fail unless you change either the setup or the program. If you change the **data code** or **protocol** after entering a character string, you will need to retype the string (see pages 6-9, 6-30).

Softkey Programming. The softkeys display only appropriate choices. Press one of the softkeys in the Monitor or Simulate Menu. Other choices will appear, leading you through the program. For example, pressing <Start> causes the new softkey choices <Display>, <Tape>, and <Timer> to appear. See Figures 6-2 and 6-3.

Block Structure. Programs are organized in blocks. A maximum of 31 blocks is allowed. Blocks provide "reference spots" for looping back or jumping ahead.

Editing Programs. Use the third set of softkeys on the Monitor or Simulate Menu as shown on page 6-3, and the cursor keys. The third set of softkeys can be accessed by the MORE key when you are at the beginning of a line. Blocks cannot be inserted or deleted. It's a good idea to leave empty blocks between used blocks for future editing.

Running Programs. After the program is developed, press <Run Menu> on the Top Level Menu. In the Run Menu choose either <Monitor> or <Simulate>, depending on whether your program is in the Monitor or the Simulate Menu. Notice that you may monitor from the "line", or from the "buffer". The 4951A lets you run programs over and over on the data in its nonvolatile buffer.

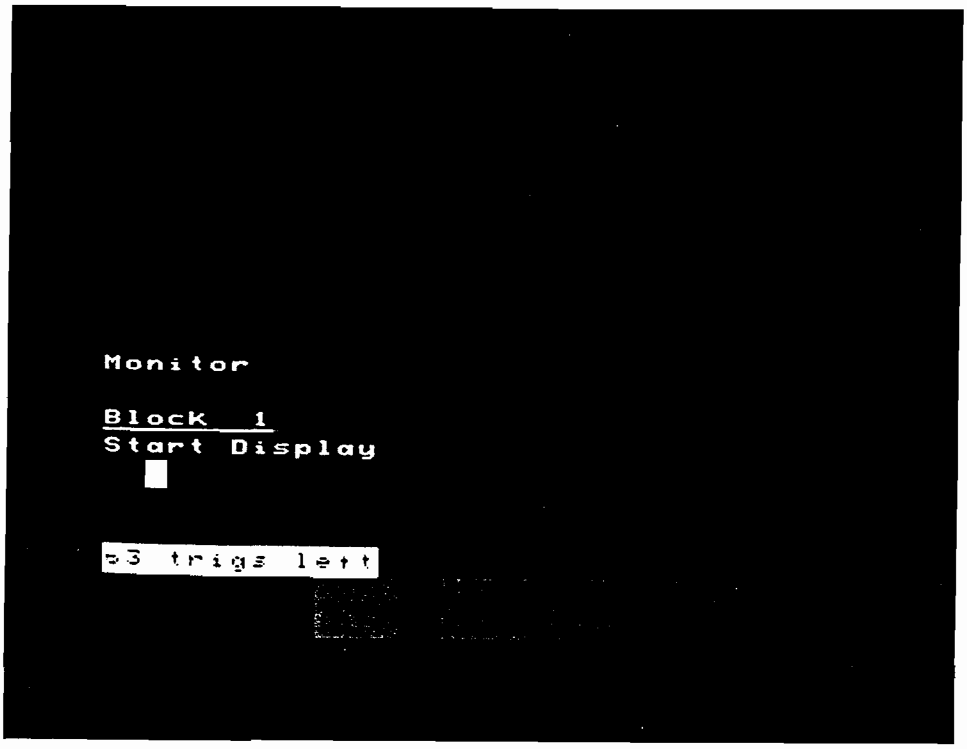


```
Monitor

Block  1
Start █

23 trigs left
```

Figure 6-2. Softkey Programming



```
Monitor

Block  1
Start Display
 █

63 trigs left
```

Figure 6-3. Softkey Programming

TRIGGERING

The 4951A indiscriminately stores line data in its buffer. You must define the events which interest you. You can tell the analyzer to look for any line event by telling it to "trigger" on that event. Most monitor/simulate measurements must be tied to a preceding trigger statement.

WHAT IS A TRIGGER?

Triggers are line events that you tell the analyzer to look for. Triggers can be characters, errors, lead changes, or the timeout of a timer.

THE IMPORTANCE OF TRIGGERING

1. The 4951A can "look for" up to 63 trigger events simultaneously.

2. The 4951A can branch to any action as a result of a trigger.

3. Most monitor/simulate measurements reference preceding trigger events.

HOW TO SPECIFY TRIGGERS

<When> is the only statement that can define a trigger. Each character specified in a <When> statement constitutes one trigger. For example, "When DTE abcd" uses four triggers. A trigger "counter" in the display shows how many triggers are left.

Triggering on Characters

```
                         | Text
                         | Hex               (* indicates bit oriented
                         | Binary               protocols only)
                         | Don't Care
          | DTE          | Not
When ---- |      ------- |      [MORE]
          | DCE          |                  |*Good FCS
                         |*End Frame ------- |*Bad FCS
                         |                  |*Abort
                         |*Start Flag
                         | Delete
                         | Insert
```

SELECTING CHARACTERS

Use the <Text> softkey for keyboard characters. The SHIFT key accesses lower-case characters; and the CNTL key accesses control characters. You can see the binary or hex value by positioning the cursor over that character and pressing <Hex> or <Binary>.

Editing Character Strings

Use the cursor keys, or the <Delete> and <Insert> softkeys to edit a string. Press MORE to access these softkeys when the cursor is positioned in the string.

SELECTING CHARACTERS (continued)

Changing the Setup after Typing a String

If you change the data code or protocol in the Setup Menu after typing a character string, you must retype the string to avoid sending or triggering on the wrong characters. Characters in one code may not have the same meaning in another code. When you move the cursor to that character, the 4951A will show "?" if it cannot find the hex or text equivalent in the new code. You can always see the binary value.

When a Character is not on the Keyboard

EBCDIC and some other data codes have control characters which are not on the keyboard. Go to the table of data codes in the appendix and find the hexadecimal equivalent. Press the <Hex> or <Binary> softkey and enter that character from the keyboard.

Binary and Hex Characters

Use the <Hex> or <Binary> softkeys to enter hexadecimal characters or binary strings. Two hex numbers occupy each character position, requiring two keyboard entries. Hex characters are underlined to differentiate them from text control characters with the same abreviation. When you press <Binary>, eight binary bits are displayed, allowing you to enter a 1 or 0 in any bit position from the sofkeys. Once you move the cursor out of the binary string, it collapses to its hex equivalent; but it is underlined to indicate it was entered in binary.

If the data code selected in the Setup Menu is less than eight bits (eg, Baudot or Transcode), the appropriate number of higher order bits are disregarded.

SELECTING CHARCTERS (continued)

Masking out Characters

Use <Don't Care> to mask out string characters or bits of no interest. "Don't Care" characters are denoted by a boxed "X". If any bit in a binary string is designated as "don't care", the compressed character is denoted by "?". See figure 6-4.

Excluding Characters

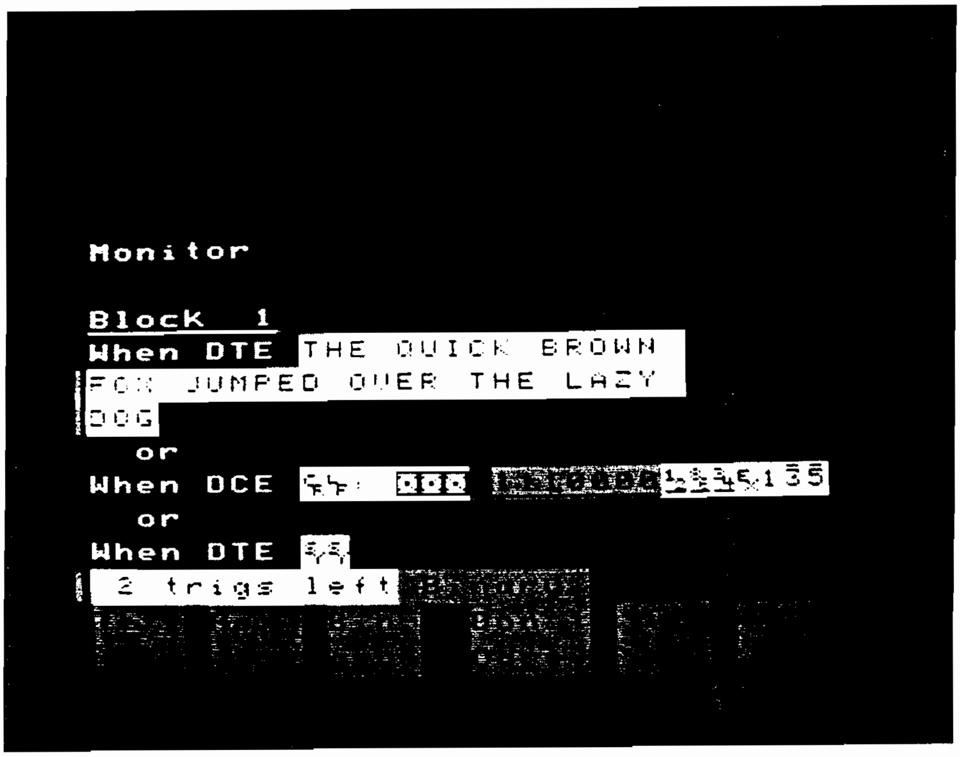
To trigger on "anything but" a character, use <Not>. "Not" characters are overlined. See figure 6-4.

Flags and Frame Check Characters

Unlike <Send> strings, flags and frame check characters are not automatically appended for <When> strings. You can enter these characters using the MORE key. The MORE key accesses the "End Frame" characters (the FCS characters and the last flag). End Frame characters may be useful if you wish to trigger on Bad FCS or Abort Characters.

Parity

When triggering on a character, the 4951A ignores the parity bit. You can see this by expanding the specified trigger character in binary when the setup is ASCII 7. The most significant (left) bit will be designated "don't care" by a boxed "X". You can explicitly define this character by entering a 1 or 0 in binary. This will override the Setup Menu.



Monitor

Block 1
When DTE THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG
or
When DCE
or
When DTE
2 trigs left

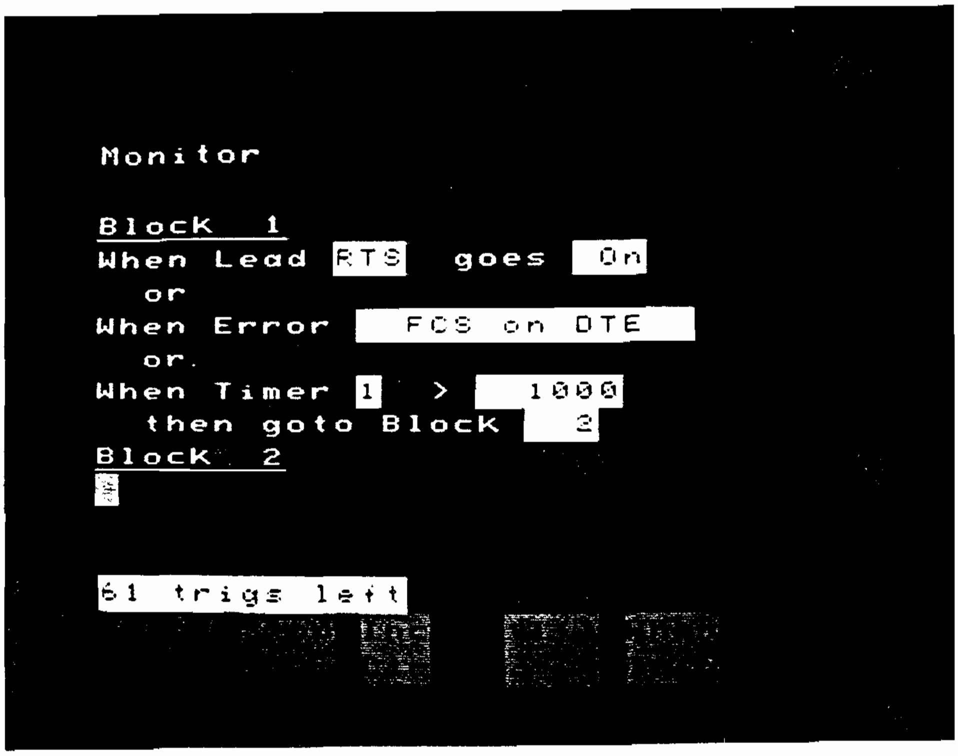
Figure 6-4. Triggering on Characters

Triggering on Errors, Leads, Timers

```
                                                   | RTS'
                                                   | CTS              On
                   | Lead ------------------------ | DSR      goes ---
                   |                               | DTR              Off
                   |                               | CD
                   |
                   |
When --------- |                               | Parity DTE (* FCS DTE) (**Framing Error DTE)
                   | Error ------------------------ | Parity DCE (* FCS DCE) (**Framing Error DCE)
                   |                               | BCC DTE (* Abort DTE)
                   |                               | BCC DCE (* Abort DCE)
                   |
                   |                                (* bit oriented setups)
                   | Timer [1,2,3,4,5]   >   [  ]     (** asynchronous setup)
```

Leads. Only the RS-232C/V.24 leads are shown above. If a different pod, such as RS-449, is connected, those leads will appear as softkey choices.

Types of Errors. BCC (Block Check Characters) and Parity errors are used only with character protocols. FCS (Frame Check Sequence) is used only with bit protocols. Framing Errors appear only in asynchronous setup (Char Async/Sync Menu). The correct error softkeys appear automatically, according to the current Setup.



```
Monitor

Block   1
When  Lead  RTS   goes  On
   or
When  Error   FCS  on  DTE
   or.
When  Timer  1   >   1000
   then  goto  Block   2
Block   2


61  trigs  left
```

Figure 6-5. Triggering on Errors, Leads, Timers

Combining Triggers

COMBINING <WHEN> STATEMENTS

<When> statements within the same block are ORed: the analyzer will look for them all simultaneously. If two are satisfied simultaneously, the first one listed will take priority. To sequence <When> statements, put them in subsequent blocks.

```
Block 1:    When DTE  abcd
              then goto Block 2
            When Error  Parity on DTE
              or
            When Error  Parity on DCE
              then goto Block 3
            When Lead  RTS goes On
              then goto Block 4
```

HOW TO "OR" TRIGGERS

The <When> statements in this example are ORed: The analyzer looks for all four simultaneously. Once a trigger is found, all other triggers in that block are disabled. If two <When> statements are satisfied simultaneously, only the first one in the block is recognized.

```
Block 1:    When DTE  abcd
              then goto Block 2
Block 2:    When DTE  efgh
            then goto Block 5
```

HOW TO "SEQUENCE" TRIGGERS

The 4951A must find string "abcd" before it can look for string "efgh". To get to Block 5, the analyzer must find both strings in order.

OVERLAPPING TRIGGERS

For overlapping triggers, the one found first disables the other triggers.

```
Block 1:    When DTE  abc
              then goto Block 2
            When DTE  ab
              then goto Block 3
```

In this example 'ab' will always be found first and will then disable the first <When> statement.

```
Block 1:    When DTE  abc
              then goto Block 2
            When DTE  bc
              then goto Block 3
            When DTE  c
              then goto Block 4
```

If the data is 'yabc', only the first <When> is satisfied. If the data is 'ybc', only the second <When> is satisfied. If the data is 'yc', only the third <When> is satisfied. The first <When> to be satisfied disables the others.

```
Block 1:    When DTE  c
              then goto Block 2
            When DTE  bc
              then goto Block 3
```

If the data is 'ybc', only the trigger 'c' is found.

MEASURING TIME BETWEEN TRIGGERS

The 4951A has five timers which can each measure time intervals in milliseconds up to 65,535 ms.

As shown in the following examples, statements using timers or leads relate to the status of the line at the time of the last trigger. You should always make sure statements relating to line status are tied to a preceding trigger statement.

```
Block 1:   When Lead RTS goes  On
              then goto Block  2
Block 2:   Start Timer  1
Block 3:   When Lead CTS goes  On
              then goto Block  4
Block 4:   Stop Timer  1
```

CORRECT WAY TO MEASURE TIME
This example shows the correct way to measure the time interval between two trigger conditions. The starting and stopping of the timer is entirely dependent upon the occurrence of the two trigger conditions.

```
Block 1:   Start Timer  1
           When RTS goes  On
              then goto Block  2
Block 2:   Stop Timer  1
           When CTS goes  On
              then goto Block ....
```

INCORRECT WAY TO MEASURE TIME
Timer 1 will now start when the run begins, rather than when RTS goes on. Timer 1 will stop when RTS goes off. You are not measuring the time between trigger events.

USING TIMERS IN SIMULATE

These simulate examples illustrate the same principles described above.

Simulate DTE

```
Block 1:   Set Lead RTS  On
           When Lead RTS goes On
              then goto Block  2

Block 2:   Start Timer  1
           When Lead CTS goes On
              then goto Block  3

Block 3:   Stop Timer  1
```

This example is correct. Timer 1 does not start or stop *until* the preceding <When> statement is satisfied.

Simulate DTE

```
Block 1:   Set Lead RTS  On

Block 2:   Start Timer_1
           When Lead CTS goes On
              Then goto Block  3

Block 3:   Stop Timer  1
```

This example is incorrect. It is not known when Timer 1 will start.

COUNTING EVENTS

The <Inc Ctr> statement (increment counter) is used for counting events.

```
Inc Ctr  [1,2,3,4,5]
```

Types of Events Counted. The 4951A has five counters, allowing you to count five different events simultaneously. "Events" may be characters or character strings occurring on the line, lead changes, timer changes, counter changes, or program loops -- almost any action the analyzer performs can be counted.

Maximum Count. Each counter will count to 10,000 and then start over from zero. By having one counter increment whenever a second counter overflows, you can count up to nearly 10,000 times 10,000. You can cascade all five counters this way.

Reset. Counters and timers are always reset to zero at the beginning of a run -- ie, when you press <Run>. Counters or timers may also be reset under program control with the <Reset> statement. When they are reset during a program, they go to zero and do not restart unless you start them again.

Examples. The first example below counts the number of parity errors on the DTE line. The second example counts the number of times RTS goes on.

Block 1:	When Error Parity on DTE then goto Block 2	Block 1:	When Lead RTS goes On then goto Block 2
Block 2:	Increment Counter 1 and then goto Block 1	Block 2:	Increment Counter 2 and then goto Block 1

TESTING CURRENT INTERFACE STATUS

The <If> statement tests current counter or lead status. For leads,"current" means at the time of the last trigger. Counters are independent of line status.

```
                    |Counter  [1,2,3,4,5]    >     [  ]
                    |
                    |
If ---------------- |
                    |
                    |               | RTS
                    |               | CTS          On
                    | Lead -------- | DSR    is ---
                    |               | DTR          Off
                    |               | CD
```

HOW <IF> AND <WHEN> ARE DIFFERENT

Only <When> defines a trigger. <When> causes the analyzer to look for future events or transitions. <If> is concerned only with "current" states. Unlike <If>, program flow stops until <When> is satisfied.

COMBINING <IF> STATEMENTS

Just as with <When> statements, <If> statements within the same block are "ORed": the first one satisfied controls the branch.

USING <IF> WITH COUNTERS

Counters run independently of line status. Therefore, an <If> statement testing counter status need not be preceded by a <When> trigger statement.

```
Block 1:    When RTS goes On
               then goto Block 2
Block 2:    Increment Counter 1
            If Counter_1 > 99
               then goto Block 4
Block 3:    Goto Block 1
Block 4:    Stop Tests
```

This example counts the number of of times RTS goes on. When RTS goes on 100 times, it stops the test.

USING <IF> WITH LEADS

Line status can only be checked by a <When> trigger statement. Therefore, an <If> statement testing a lead condition always refers to the line status at the time of the last trigger.

```
Block 1:    When Lead RTS goes On
               then goto Block 2
Block 2:    If Lead CTS is On
               then goto Block 4
```

Block 2 will test CTS at the time the <When> statement in Block 1 is satisfied.

FILTERING DATA

MARKING TRIGGER EVENTS -- START, STOP, BEEP, HIGHLIGHT

By using these four commands, you can have the 4951A notify you when it has found some event. And in the 4951A, events are defined by triggers. As we discussed previously in this chapter, timers and lead status must be tied to a preceding <When> statement. The same is true of the above four commands. For example, whenever you "Start" an action, always provide a reference to some line event with a preceding <When> statement. **NOTE:** the <Wait> statement cannot be used with any of these commands. Use <Wait> only with <Send> and <Set Lead> to delay output.

START AND STOP

The <Start> and <Stop> statements can be used to filter events of interest: you let the 4951A do the watching for you. Define an event of interest in a preceding <When> statement, and then "start" or "stop" the display, tape, or timers when that event occurs.

Stop Display

The <Stop Display> statement freezes the display after the occurence of some trigger event. That trigger event, and the immediately preceding data, are displayed on the screen. Note that the run is not stopped: the buffer is continually being filled with new data. To stop the run after the event, use <Stop Tests>.

Start & Stop Tape

The tape can be started and stopped only once during a program.

Start & Stop Timer

Timers measure intervals between trigger events. Always precede <Start> and<Stop> timer statements with a <When> statement defining the event. Otherwise, your time measurements may not be accurate. See page 6-14

Stop Tests

The <Stop Tests> statement causes the analyzer to halt. No new data will be loaded into the buffer or displayed on the screen, the tape will stop, and any active timers will stop. You can use this statement within a program to have the analyzer immediately stop upon finding some event.

Examples of Start and Stop

The first example stops the run if there is a Negative Acknowledgement on the DTE line. Note that you enter the "NAK" by pressing the CNTL and "U" keys at the same time. The second example freezes the display if there is a Frame Check Sequence error on the the DTE line.

```
Block 1:    When DTE NAK                  Block 1:    When Error FCS on DTE
              then goto Block 2                         then goto Block 2
Block 2:    Stop Tests                    Block 2:    Stop Display
```

BEEP AND HIGLIGHT

Beep

The <Beep> statement provides an audible sound for some specified condition. You can have the analyzer beep anytime, and as often as desired.

Highlight

Use <Highlight> after a <When> statement to mark trigger events in memory: characters, errors, lead or timer transitions. Highlighted characters appear in half-bright video both during run-time, and when looking at the buffer in <Examine Data> mode. Lead and timer transitions appear in the DCE line. The 4951A "remembers" the only the last 64 highlights in the buffer. Only the last character of string is highlighted.

Highlight examples are shown in Figure 6-6. Note that the clock timeout highlight is denoted by a small clock.

Examples of Highlight and Beep

The first example highlights "xyz" in memory whenever it occurs on the DCE line. The second example causes a continuous beep whenever the string "abc" occcurs on the DTE line.

```
Block 1:    When DCE xyz                  Block 1:    When DTE abc
              then goto Block 2                         then goto Block 2
Block 2:    Highlight                     Block 2:    Beep
              then goto Block 1                         and then goto Block 2
```

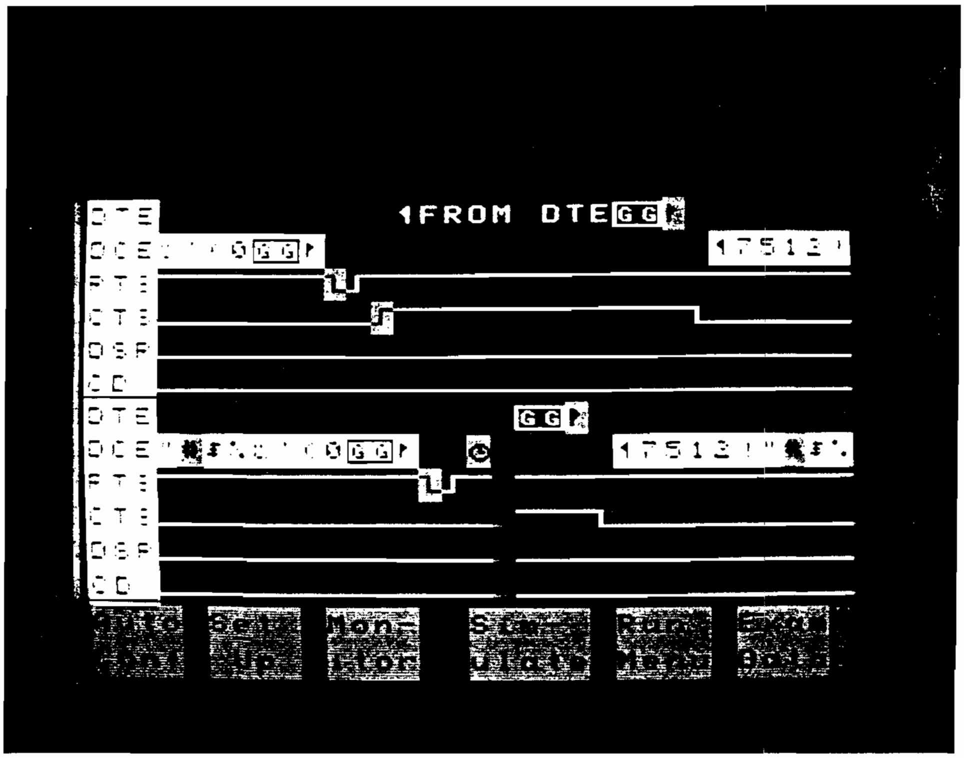
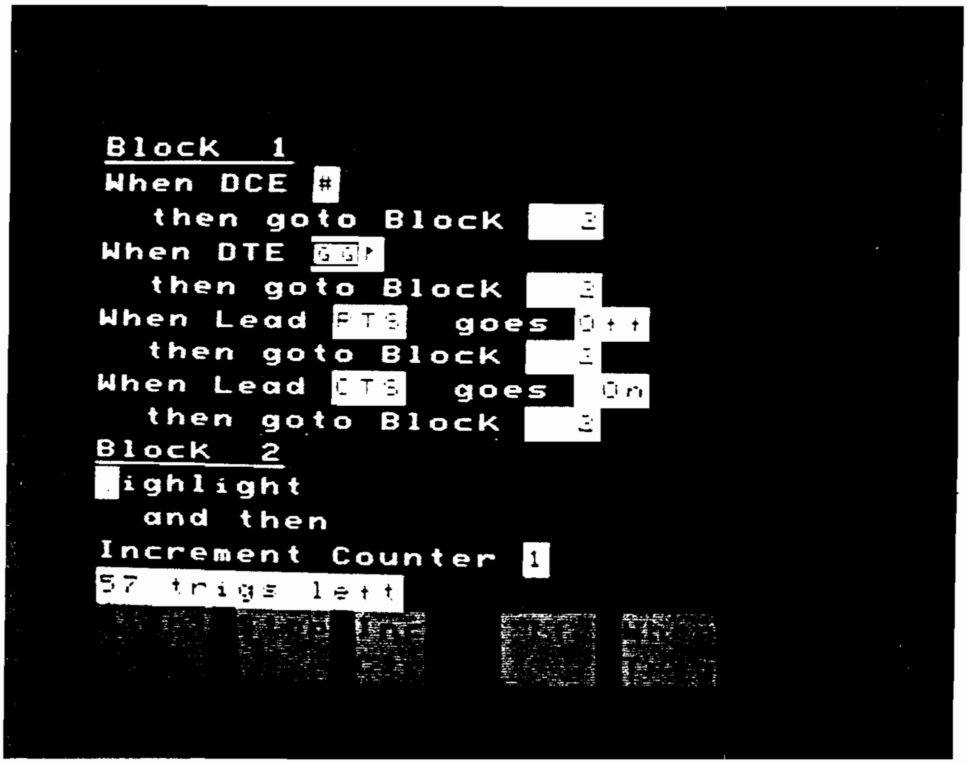


Figure 6-6. Highlights in the Buffer (Examine Data Menu)



```
Block  1
When DCE #
  then goto Block   2
When DTE GG
  then goto Block   2
When Lead RTS  goes Off
  then goto Block   2
When Lead CTS  goes  On
  then goto Block   2
Block  2
Highlight
  and then
Increment Counter 1
57 trigs left
```

Figure 6-7. Program Producing the Display in Figure 6-6.

TRANSMITTING CHARACTERS -- SEND

(Simulate only)

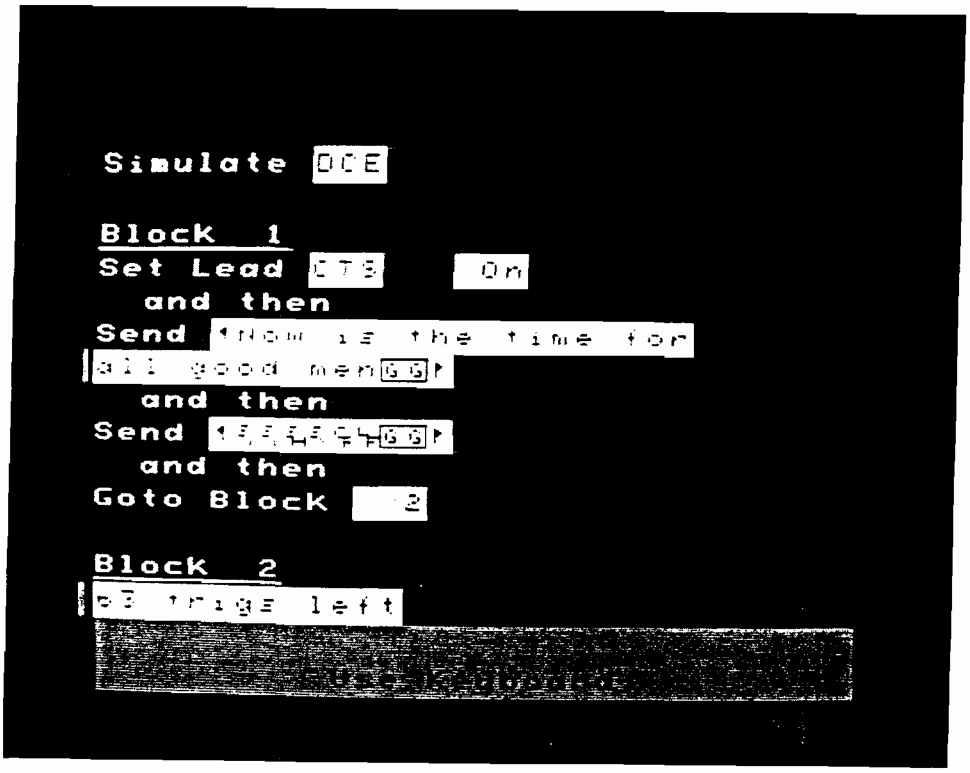
Using <Send>, you can simulate a DTE or DCE by sending any bit or character sequence in any of the codes supported by the 4951A. Maximum length for each string is 251 characters.

```
                  | Text
                  | Hex
                  | Binary
Send ------------ |              -------------- [input from the keyboard]
                  |    [MORE]
                  |
                  | Delete
                  | Insert
```

HANDSHAKING REQUIREMENTS ON THE INTERFACE

NOTE: DETERMINE WHICH INTERFACE LEADS MUST BE SET ON OR OFF BEFORE SENDING DATA. OTHERWISE, THE RECEIVING EQUIPMENT MAY NOT ACCEPT THE DATA.

The 4951A does not need to set control leads before sending data. However, the receiving equipment may require control signals before accepting the data you are sending. See page 6-32 for discussion of the <Set Lead> statement.



```
Simulate DCE

Block 1
Set Lead CTS     On
  and then
Send ‹Now is the time for
all good men›
  and then
Send ‹›
  and then
Goto Block 2

Block 2
63 trigs left
          Use keyboard
```

Figure 6-8. Sending Characters in Simulate

SELECTING SEND CHARACTERS

Use the <Text> softkey for keyboard characters. The SHIFT key accesses lower-case characters; and the CNTL key accesses control characters. You can see the binary or hex value by positioning the cursor over that character and pressing <Hex> or <Binary>.

NOTE: IN CHARACTER ORIENTED PROTOCOLS, YOU MUST EXPLICITLY ENTER SYNC CHARACTERS, SUCH AS SY SY. OTHERWISE, THE RECEIVING DEVICE WILL NOT RECOGNIZE THE MESSAGE (see page 6-31).

When a Character is not on the Keyboard

EBCDIC and some other data codes have control characters which are not on the keyboard. Go to the table of data codes in the appendix and find the hexadecimal equivalent. Press the <Hex> or <Binary> softkey and enter that character from the keyboard.

Binary and Hex Characters

Use the <Hex> or <Binary> softkeys to enter hexadecimal characters or binary strings. Two hex numbers occupy each character position, requiring two keyboard entries. When you press <Binary>, eight binary bits are displayed, allowing you to enter a 1 or 0 in any BIT position from the softkeys. Once you move the cursor out of the binary string, it collapses to its hex equivalent.

Editing Strings

Use the <Delete> and <Insert> softkeys to edit a string. Press MORE to access these softkeys when the cursor is positioned in the string.

Sending Idles

During simulation, the 4951A continuously sends idles when not sending data. This is also true when using <Wait> to delay output. You can explicity enter idles when simulating; but otherwise they are not stored in the buffer during normal monitoring or simulating. To store idles, see page 5-20.

Block Check Characters (BCC)

In character oriented protocols, the 4951A automatically appends the correct block check characters to <Send> strings. You can see these characters at run-time, or in the buffer after a run. In Char Async/Sync setup, you can select the characters on which error checking is to start and stop. **Start on** character starts error checking on the character following the designated character. **Stop on** character includes the designated character in the error check.

Flags and Frame Check Characters

Flags and frame check sequence (FCS) characters are automatically added whenever a bit oriented protocol -- HDLC, SDLC, X.25 -- is selected in the Setup Menu. The 4951A does not show you the actual frame check character. For received data, **GG**, **BB**, or **AA** are displayed to indicate "good FCS", "bad FCS", or "abort". For Send strings, good FCS characters (**GG**) are automatically selected; but you may choose Bad FCS (**BB**) characters or Abort (**AA**) characters, either by moving the cursor to the frame check characters, or by pressing the <End Frame> softkey. Flags and frame check characters disappear if you change the Setup to a character oriented protocol and again move the cursor into the string.

Parity Bits

In <Text> mode, the 4951A automatically selects the parity bit according to the current setup.

Using <Hex> or <Binary> mode, you can explicitly change the parity bit of any Send character, regardless of the setup. For example, assume the setup is ASCII 7 with odd parity. In the <Text> mode, if you enter an "E" in the send string, the transmitted binary code will be 01000101. The parity is 0 (left-most bit). To change the parity bit to 1, use <Binary> or <Hex> and enter 11000101 or C5.

NOTE: Although your Send string now shows a hex C5 -- which you are actually transmitting -- the run-time and Examine Data displays ignore the parity bit, and will still show an "E".

Zero Bit Insertion

In bit oriented protocols, the 4951A automatically inserts a 0 after five consequtive 1's before transmitting non-flag characters (invisible to the user). When receiving, it automatically removes any 0 bits inserted by the transmitter. This is not true in Char Async/Sync setup.

Changing the Setup After Typing the String

If you change the data code or protocol in the Setup Menu after typing a character string, you must retype the string to avoid sending or triggering on the wrong characters. Characters in one code may not have the same meaning in another code. When you move the cursor to that character, the 4951A will show "?" if it cannot find the hex or text equivalent in the new code. You can always see the binary value.

USING TIMERS WITH <SEND>

As always, timers measure intervals between trigger events. Each line event is "time stamped" as it is placed in the buffer. Timers are always referenced to the last preceding <When> trigger statement.

NOTE: As shown in this example, sync characters must be explicitly entered in character oriented protocols. Otherwise, the receiver will not accept the message.

```
Block 1:    Send  SY SY SX abcdefghijk EX SY SY
                 and then goto Block 2
Block 2:    When  DTE  a
                 then goto Block 3
Block 3:    Start Timer 1
            When  DTE  k
                 then goto Block 4
Block 4:    Stop Timer 1
```

This is the correct way to measure the time it takes to send the string. The timer is activated by preceding <When> statements.

```
Block 1:    Start Timer 1
Block 2:    Send  SY SY SX abcdefghijk EX SY SY
                 and then
            Stop Timer 1
```

This is incorrect because the timer is not tied to a <When> trigger statement. You will not measure the time it takes to send the string

CONTROLLING INTERFACE LEADS

(Simulate only)

In Simulate Mode, <Set Lead> turns on or off one of the RS-232C/V.24 or RS-449 leads. The 4951A always knows which pod is attached and displays the correct softkeys. With a RS-232C/V.24 interface, a lead is "on" when the voltage is high; it is "off" when the voltage is low. When simulating a DTE, you cannot control DCE leads, and vice versa: Only the appropriate softkeys are displayed, as shown below. (* indicates RS 449-leads). See Appendix C for more information.

	DTE		DCE
Set Lead	RTS (*RS) DTR (*TR)	Set Lead	CTS (*CS) DSR (*DM) CD (*RR)

LEAD STATUS DURING SIMULATION

NOTE: DETERMINE WHICH INTERFACE LEADS MUST BE SET ON OR OFF BEFORE SENDING DATA. OTHERWISE, THE RECEIVING EQUIPMENT MAY NOT ACCEPT THE DATA.

The 4951A must be programmed to control the leads in the simulate mode (this is the only time the 4951A controls the interface leads). At the begining of a simulation run, the 4951A sets all the above interface leads off. You must actually turn these leads on with the <Set Lead> statement in order to do handshaking with a receiving device.

LEAD STATUS WHEN NOT SIMULATING

Lead status is independent of the 4951A except when it is simulating. Remember this when you use <If Lead> statement in a monitor program.

SET LEAD EXAMPLES

Because the 4951A always sets all five leads (DTR, RTS, DSR, CTS, CD) off at the beginning of the simulation run, <Set Lead> statements are needed to turn the appropriate leads back on before sending data. If this is not done, the receiving device might not accept data from the 4951A. You must know the handshaking requirements on your system in order to simulate correctly.

Simulate DTE

```
Block 1:    Set Lead DTR On
                and then goto Block 2
Block 2:    Wait 1000
                and then
            Set Lead RTS On
Block 3:    When Lead CTS goes On
                then goto Block 4
Block 4:    Send abcd
                and then
            Set Lead RTS Off
```

Simulate DCE

```
Block 1:    When Lead RTS goes On
                then goto Block 2
Block 2:    Wait 100
                and then
            Set Lead CTS On
                and then
            Set Lead CD On
                and then
            Send abcd
```

DELAYING OUTPUT -- WAIT

(Simulate only)

WAIT CONTROLS OUTPUT

NOTE: THE <WAIT> STATEMENT CONTROLS OUTPUT ONLY.

Use <Wait> only with <Send> and <Set Lead> statements. <Wait> has no effect on program flow or timers.

DELAYING STRINGS OR LEADS

The <Wait> command can be set in 1 ms increments to cause delays of up to 65,535 ms. In combination with counters, very long delays can be set up. The following example repeatedly sends a string of numbers and then waits 50 ms.

```
Block 1:     Send 1234567
               and then
             Wait-50 msec
               and then goto Block 1
```

ERROR MESSAGES

Max Length. This message appears if you attempt to specify more than 255 characters in a single string.

Max Strings. Appears if the Monitor and Simulate Menus combined contain more than 1400 strings.

Menu Full. Appears if the Monitor and Simulate Menus combined contain more than 143 steps.

Invalid Mon/Sim Menu. This may occur if you enter "When DTE/DCE" without completing the trigger condition.

Status Messages (Current Mode of Entering Data)

Text. Enter a single keyboard character.

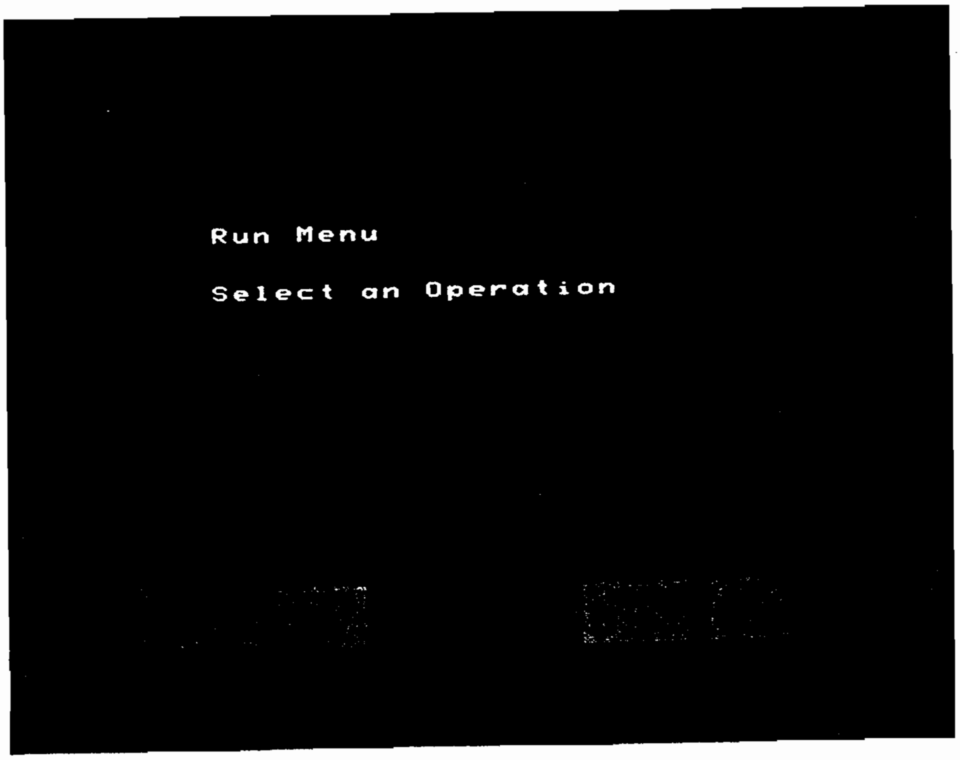
Hex. Enter two hex digits for a character.

Binary. Enter eight bits from softkeys. If the Setup data code is less than eight bits, the most significant bits are ignored.

End Frame. Enter the FCS character (good, bad, abort, don't care).

CHAPTER 7

THE RUN MENU



Run Menu

Select an Operation

Figure 7-1. The Run Menu

The Run Menu

Except in Auto Configure, where the 4951A automatically goes into the run mode, use the **Run Menu** to execute all tests.

MONITORING ON-LINE

For a detailed description of monitoring on-line, see chapters 2 and 6.

1. **Hookup**

   Connect the 4951A to the line to be monitored. See chapter 2.

2. **Setup**

   Use Auto Configure, or the Setup Menus. See chapters 4 and 5.

3. **Program the Monitor Menu**

   This step is optional. Go to the Monitor Menu and program any measurements you want the analyzer to make. See chapter 6.

4. **Run Menu**

   In the Run Menu press <Monitor Line>. The 4951A will display the line data. Use the **Setup** Menus to change the display format.

MONITORING FROM BUFFER

Running from Buffer is almost the same as running on-line.

1. **Hookup**

   The 4951A need not be connected to the pod to monitor from buffer.

2. **Load the Buffer**

   Load the buffer with data, either from the tape, or by running on-line. With the nonvolatile memory, previously loaded data can be used.

3. **Setup**

   Use the Setup Menus. With the nonvolatile memory, previous setups are saved and can be used.

4. **Program the Monitor Menu**

   Again, this step is optional: you may go right to the Run Menu. But one of the advantages of monitoring from buffer is that you can program the 4951A to run measurements over and over on the data in the nonvolatile buffer.

5. **Run Menu**

   In the Run Menu press <Monitor Buffer>. The 4951A will begin displaying buffer data and running any measurements you may have setup in the Monitor Menu.

RUNNING SIMULATION

1. **Hookup**

   Substitute the 4951A for the DTE or DCE. See chapter 2.

2. **Setup**

   Use the Setup Menus. See chapter 5.

3. **Program the Simulate Menu**

   In the Simulate Menu, select either DTE or DCE. Using the softkeys, select the operations (eg, Sending or Setting Leads) you want the 4951A to simulate. See chapter 6.

4. **Run Menu**

   In the Run Menu press <Simulate>. To change the display format, go back to the Setup Menus.

RUNNING BERT

Hook up the 4951A as a DTE. After the appropriate selections in the BERT menu, perform the BERT test by going to the Run Menu and pressing <BERT>. See chapters 2 and 9.

Run-Time Softkeys

Softkeys and messages shown at the bottom of the display during run-time are:

```
Hex        Stop          Block = n         Summary
           Disp
```

Hex/Text. Pressing <Hex> converts all subsequent displayed data to hex format. The softkey label then changes to <Text> for changing the display back to the current data code.

Stop Display/Start Display. The <Start Display> softkey alternates with <Stop Display>. The <Stop Display> softkey freezes the display, and <Start Display> causes the most recent incoming data to be displayed. These do not affect the run, but the continuity of the run-time display may be lost.

Block = n. Message indicating which 2K bytes of memory (1 to 16) is being displayed. When memory wraparound occurs, the next 2K block to be loaded becomes Block 1. (When viewing the buffer after run-time with Examine Data, the oldest data becomes Block 1. In Examine Data, block numbers may go as high as 128 if the buffer data has been loaded from tape.)

Summary/Data Display. These alternate to show either data or the Setup Summary. Press <Summary> at anytime, without affecting the run, to review the current setup and observe the counters and timers (see Figure 7-2). The summary tracks the current Setup Menu. Timers are updated whenever a trigger is found. Counters are updated every 1/2 second. Counters automatically roll over at 9999 to 0. Press <Data> to return to the data display.

Run-Time Messages

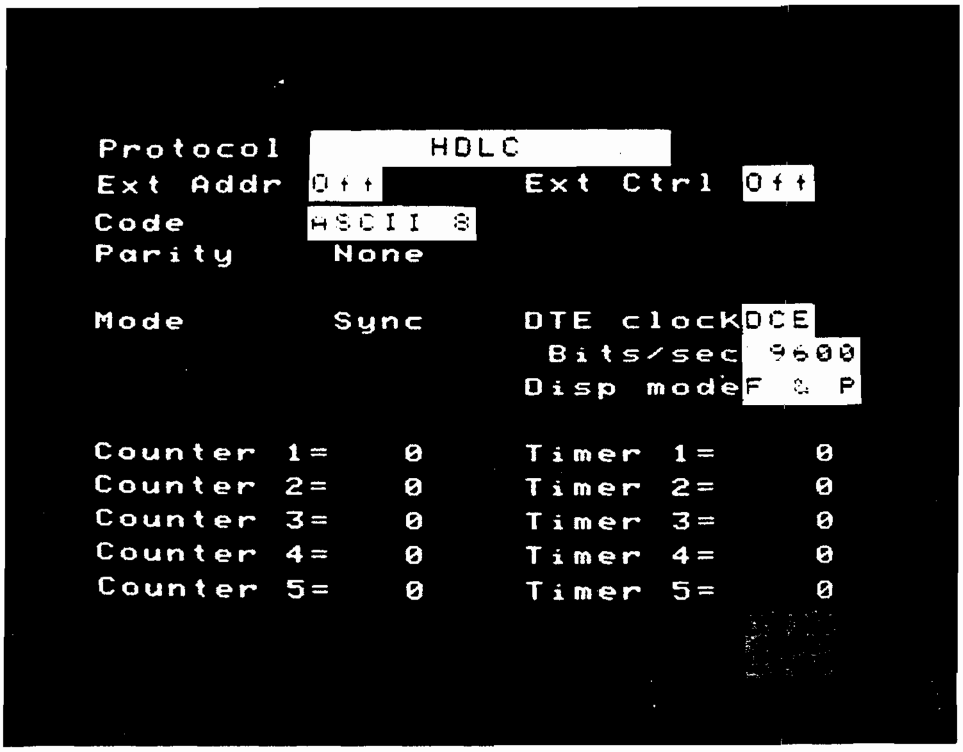
Running. Message indicating data is being processed.

No Pod Attached. An interface pod must be attached in order to run BERT, Auto Configure, Simulation, and Monitor on-line. No pod is necessary to Monitor from Buffer.

Buffer Overflow. Data has filled the buffer and will begin to overwrite data that has not yet been processed. This can occur when storing data directly from the line to tape, or when incoming speed is higher than specified.

Receiver Overrun. The hardware capability to process serial input is being exceeded. Typically, this may occur at line speeds greater than 30 kbps in character oriented protocols, and speeds greater than 64 kbs in bit oriented protocols.

Invalid Monitor/Simulate Menu. This will occur because of incomplete <When> or <If> statements. For example, if you do not finish the statement "When DTE".



```
Protocol          HDLC
Ext Addr  Off            Ext Ctrl  Off
Code      ASCII 8
Parity      None

Mode        Sync         DTE clockDCE
                          Bits/sec 9600
                         Disp modeF & P

Counter 1=    0          Timer 1=     0
Counter 2=    0          Timer 2=     0
Counter 3=    0          Timer 3=     0
Counter 4=    0          Timer 4=     0
Counter 5=    0          Timer 5=     0
```

Figure 7-2. The Setup Summary Display

Displaying Data

Five display formats are available. Use the Setup Menus to change the display format. Figures 7-3 to 7-7 show examples of each format.

DTE — DTE data only. Displayed in regular video.

DCE — DCE data only, Displayed in inverse video.

Two Line — DTE over DCE. DCE data is displayed in inverse video.

Data & State — DTE over DCE data, and timing diagrams of four interface leads.

Frame & Packet — Decodes bit oriented frames. Decodes X.25 packets in the Examine Data Menu (see chapter 8 for a definition of terms).

FULL DUPLEX AND HALF DUPLEX DATA

See figures 7-8 and 7-9 for examples of full duplex and half duplex data. On half duplex data, the 4951A displays complete DTE messages alternating with complete DCE messages. On full duplex data, the 4951A displays the individual characters according to the timing order in which they are received.

BLINKING CHARACTERS

Blinking characters indicate failed error checks: BCC, FCS, parity, or framing errors resulting from incorrect setup or loss of synchronization. See chapter 8.

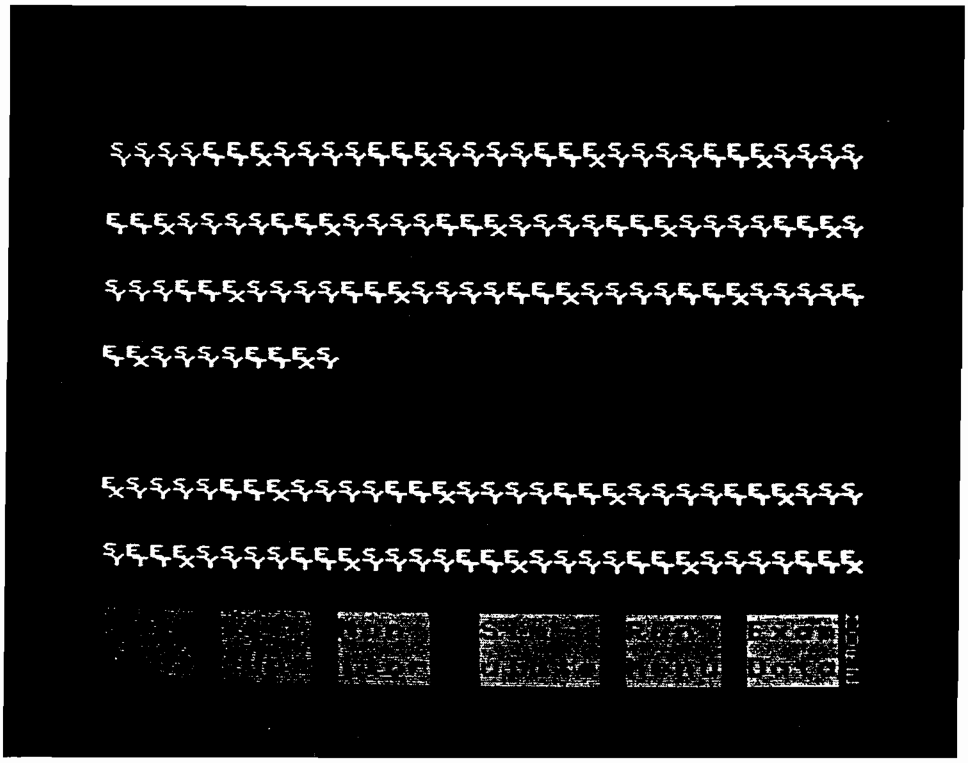


Figure 7-3. DTE Display Format

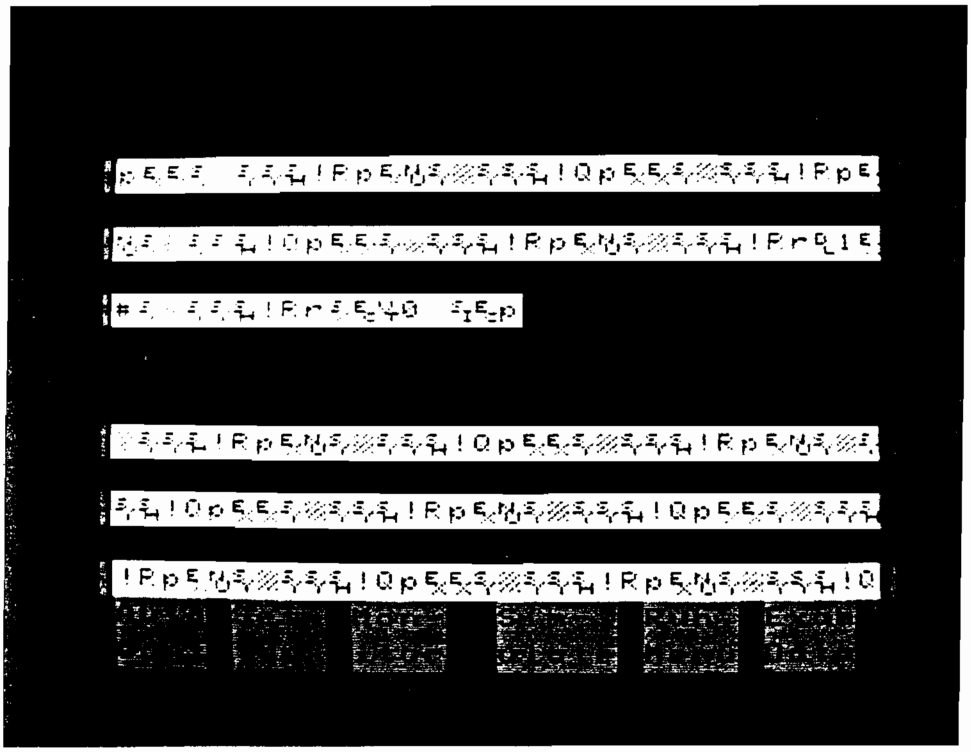


Figure 7-4. DCE Display Format

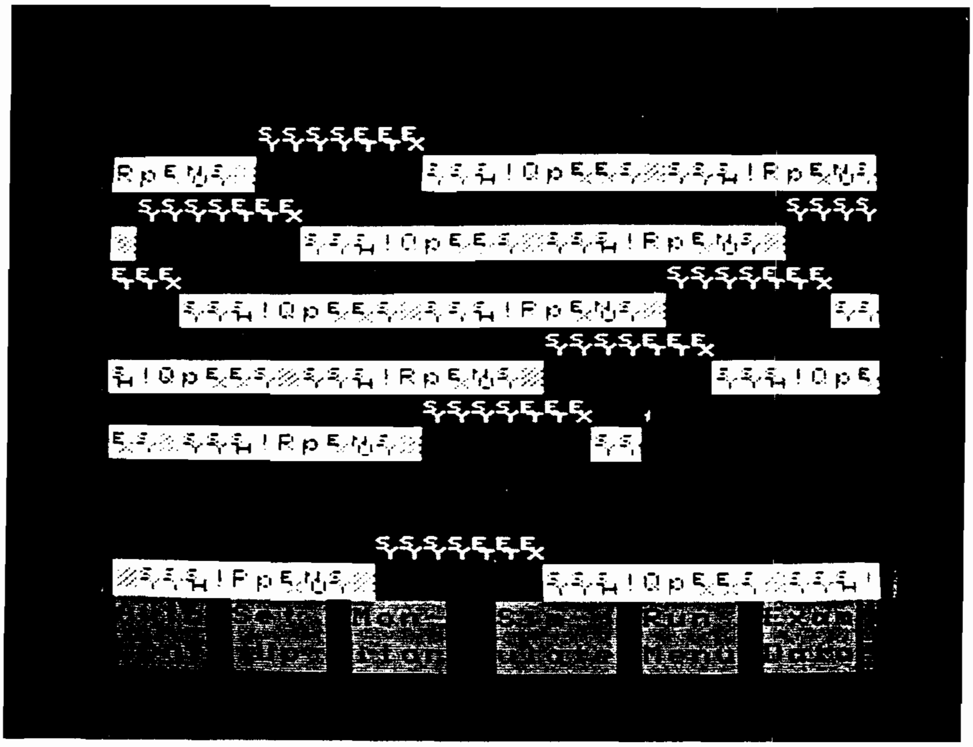


Figure 7-5. Two Line Display Format

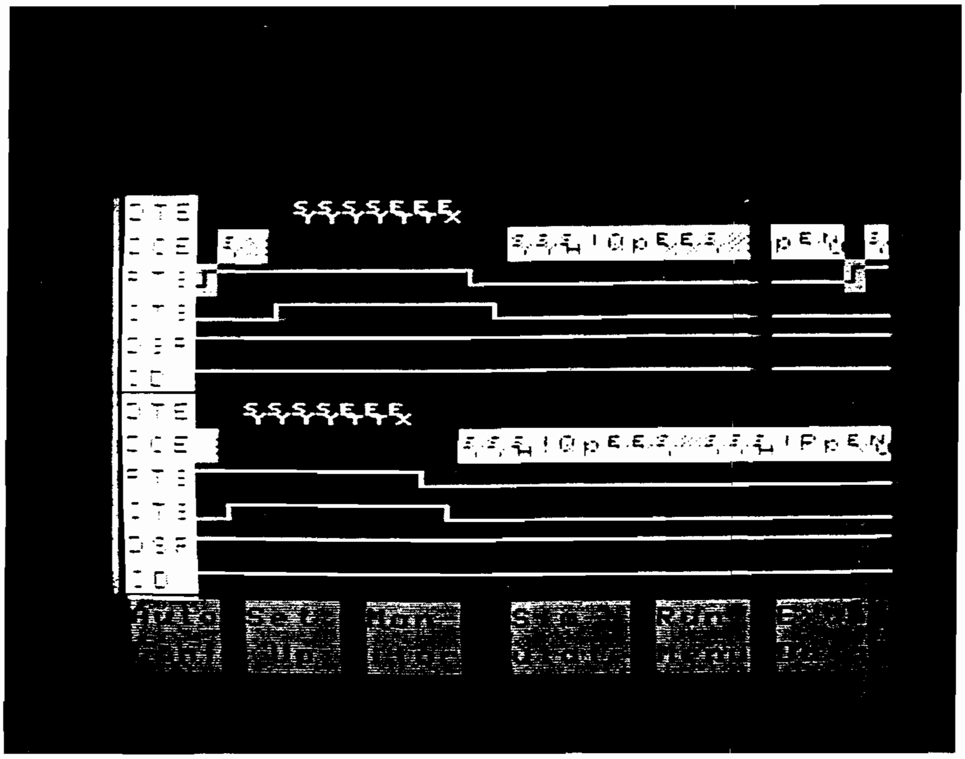
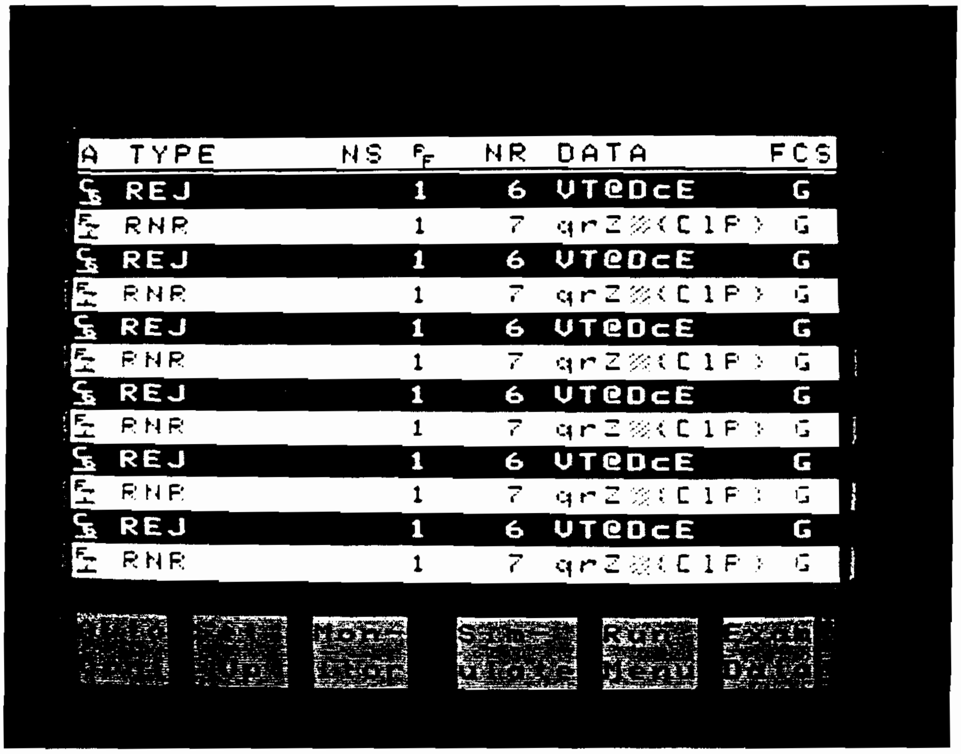
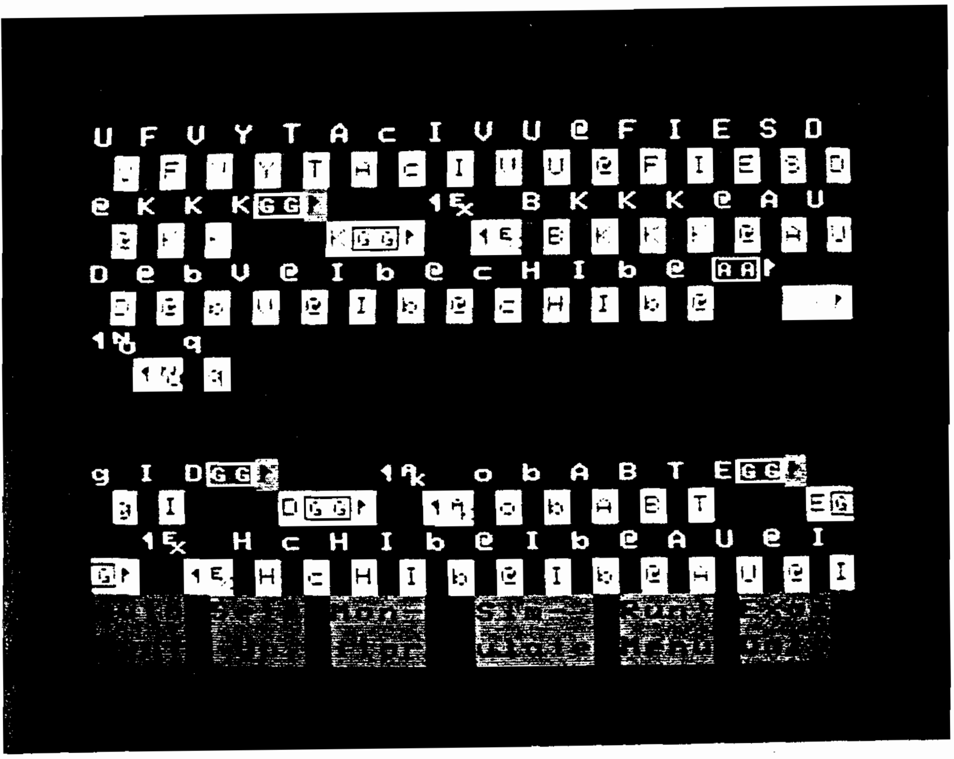


Figure 7-6. Data & State Display Format



A	TYPE	NS	PF	NR	DATA	FCS
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G
	REJ		1	6	UT@DcE	G
	RNR		1	7	qrZ▨(C1P)	G

Figure 7-7. Frame & Packet Display Format (See chapters 5 and 8 for definitions)



```
U F U Y T A c I U U @ F I E S D
@ K K K           B K K K @ A U
D @ b U @ I b @ c H I b @
  q
g I D             o b A B T E
    H c H I b @ I b @ A U @ I
```

Figure 7-8. How Full Duplex Data Looks (See page 7-8)

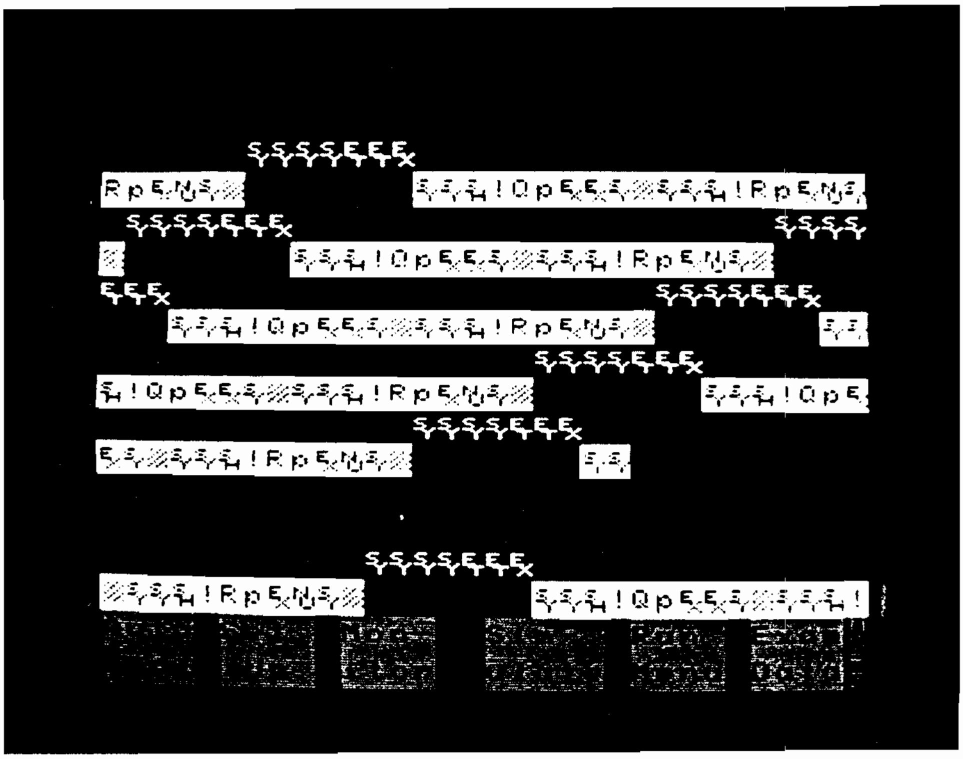


Figure 7-9. How Half Duplex Data Looks (See page 7-9)

CHAPTER 8

THE EXAMINE DATA MENU

Viewing the Buffer
Softkeys
Packet Decoding
Finding Unknown Protocols
Frame Sizes vs Data Codes

Viewing the Buffer

Press <Exam Data> on the Top Level Menu to look at the buffer after run-time. Note how this differs from Monitoring On-line, Monitoring From Buffer, or Simulating: In all these, you are looking at the buffer during run-time -- you can stop the display, but you cannot go backward. The Examine Data Menu lets you scroll through the entire 32 Kbyte buffer.

WHAT IS STORED IN THE BUFFER

Most line activity is stored in the buffer. This is what makes it possible for the 4951A to post-process data (monitor from buffer). The following are stored:

1. DTE and DCE characters.

2. Lead changes on the five interface leads. Select Data & State display format or use the highlight feature in the Monitor and Simulate Menus.

3. Errors, such as parity, BCC, and FCS.

4. Frame markers and packet markers.

5. Time marks and lead status.

HOW TO LOAD THE BUFFER

The buffer is always being continually loaded with data when monitoring on-line or simulating. The buffer can also be loaded from tape.

Uses for the Examine Data Menu

Viewing Timers and Counters after a run. The Examine Data Menu lets you look at the final state of the timers and counters after a run. The timers and counters will only be reset if (1) Another run is started; (2) <Reset> is pressed; or (3) Before load operations from tape.

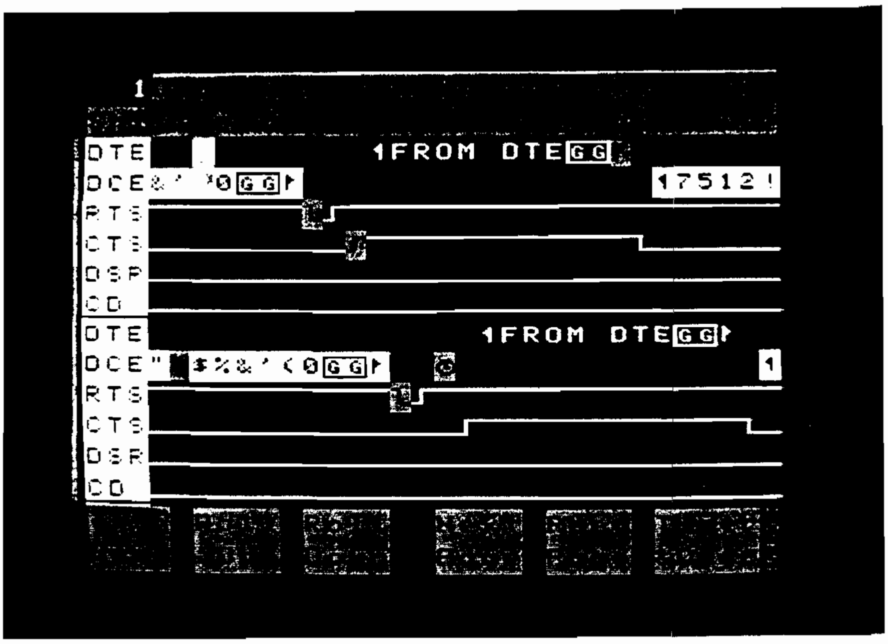
Viewing the Entire Buffer. During run-time you can stop the display; but you can't go back and look at what you've missed. The Examine Data Menu enables you to go back after a run and scroll through the buffer.

Bit Shifting. If the framing is off because the sync characters are unknown, use the bit shift softkey in the Examine Data Menu to realign the bits until the data becomes meaningful. See page 8-10.

Decoding Characters. Move the cursor to any character and observe the binary, hex, and octal equivalents at the top of the display. You can also see the parity bit for any character.

Decoding Packets. In X.25 setup and Frame & Packet display format, the 4951A will decode packet headers simultaneously with control field information in the Examine Data Menu. See page 8-7 for an illustration and description.

Decoding High Level Protocols. If the data contains other higher level protocol information (eg ISO levels 4-7, or SNA), the relevant fields can be read from the hex/octal/binary decoding at the top of the display.



	Hex	Roll Up	Roll Down	Next Page	Prev Page	Timer &Cntr
MORE						
	Spec. Block	Next Hilit				Bit Shift

Figure 8-1. Buffer Display in Examine Data Menu (Data & State Display Format)

Softkeys

Hex/Text. Displays buffer data in either the code selected in the Setup Menu, or in hexadecimal.

Roll Up/Roll Down. For scrolling through the buffer.

Next Page/Prev Page. For moving from one screen-full of data to another. A page is one full screen of information.

Timers & Counters. You can at any time look at a summary of the Setup parameters, as well as the status of the timers and counters at the end of the last run.

Specify Block. For specifying any 2K byte block. The block number indicates the first character's position in the buffer. Some buffer information, like time marks, is not displayed, so <Next Page> may cause the block number to jump by several numbers. Buffer data loaded from tape may have block numbers as high as 128.

Next Highlight. The <Highlight> softkey in the Monitor or Simulate Menus lets you mark trigger events. This softkey lets you move to the next higlighted event.

Next Segmt/Prev Segmt. With this feature you can examine the tape like the buffer. These softkeys will load either the next or the previous 16K bytes from tape into the buffer for observation. Appears only when you have loaded a tape file.

Bit Shift. Shifts framing of the displayed characters one bit at a time. Use this softkey to find the correct framing of unknown protocols (see page 8-10). The parity bit is not shifted. Appears only in character oriented setups.

Displaying Data

The same five formats available during run-time are available. See figures 7-3 to 7-7.

DTE	DTE data only. Displayed in regular video.
DCE	DCE data only, Displayed in inverse video.
Two Line	DTE over DCE. DCE data is displayed in inverse video.
Data & State	DTE over DCE data, and timing diagrams of four interface leads.
Frame & Packet	Decodes bit oriented frames. In the Examine Data Menu only, decodes X.25 packets. See page 8-7.

HOW SETUP AFFECTS DISPLAY

In some display formats you may not be able to observe the buffer data. For example, with frame & packet format, you cannot see BSC data. Data & State format will always show any data in the buffer, even when it consists only of lead transitions.

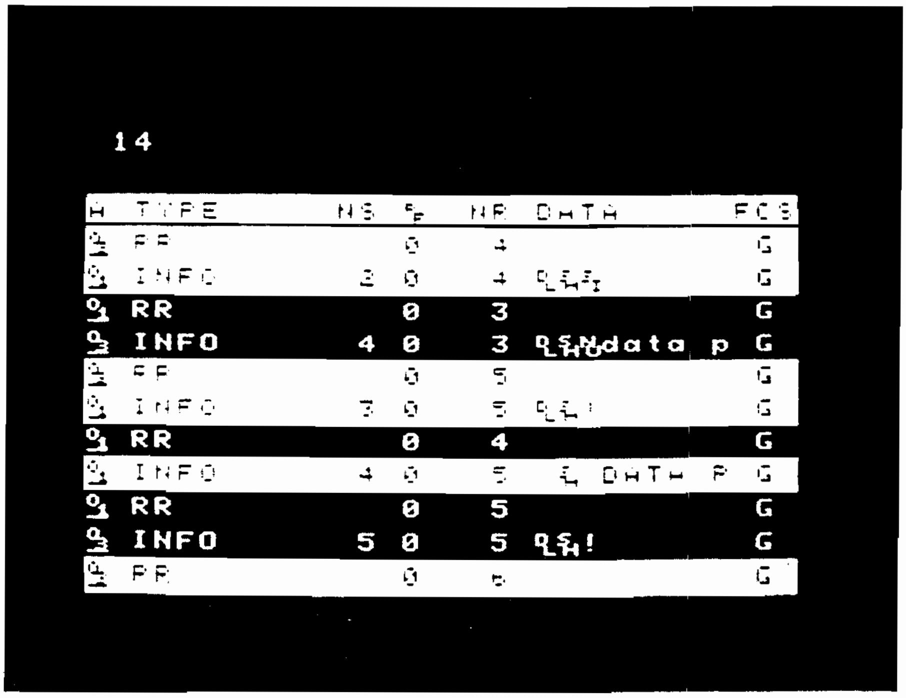
BLINKING CHARACTERS

Blinking characters indicate failed error checks: BCC, FCS, parity, or framing errors resulting from incorrect setup or loss of synchronization. See page 8-9.

X.25 In Frame & Packet Display Format

X.25 looks the same as HDLC or SDLC when running in the frame and packet display format. After capturing data, however, the Examine Data Menu shows both frame decoding and packet decoding. As shown on the next page, packet information at the cursor location is decoded at the top of the display. The DTE and DCE columns contain the following information. See Appendix E for more details.

Q-Bit	Qualifier Bit.
D-Bit	Delivery Confirmation Bit.
MOD	Modulo 8 or 128.
LCN	Logical Channel Number.
TYPE	Type of packet.
P(S)	Packet Send Sequence Number.
M-Bit	More Data Mark.
P(R)	Packet Receive Sequence Number.
Data	Displays the first five characters of the data field.



```
14

H  TYPE          NS  PF  NR  DATA           FCS
01 RR                 0   4                   G
03 INFO           2   0   4                   G
01 RR                 0   3                   G
03 INFO           4   0   3      data    p    G
01 RR                 0   5                   G
03 INFO           3   0   5      !            G
01 RR                 0   4                   G
03 INFO           4   0   5       DATA   P    G
01 RR                 0   5                   G
03 INFO           5   0   5      !            G
01 RR                 0   6                   G
```

Figure 8-2. Decoding X.25 Packets with Frame & Packet Format

Finding Unknown Protocols

Use the Examine Data Menu in conjunction with the Char Asyn/Syn Menu to determine the parameters of unknown protocols. See also page 5-20.

Use Auto Configure as a Starting Point

Use Auto Configure to find at least some of the parameters and give you a starting point.

Set up the Char Async/Sync Menu

Set up the Char Asyn/Syn Menu to capture all the data on the line, including idles.

1. If you know the data code, **Sync on** idles. Otherwise, sync on FF or 00. Most character oriented protocols idle in FF. Some IPARS circuits idle in 00.

2. **Drop sync** 0 **chrs after** none. Now you will never drop sync and will thus take in all the data.

3. If you do not know the data code, initially use a data code with no parity of the same character frame size as was found by Auto Configure. See page 8-11.

Use the Examine Data Menu

Monitor the line to capture some data in the buffer.

1. In the Examine Data Menu, try bit shifting. If the data still does not make sense, go back to the Setup Menu and try another data code with no parity of the same character frame size. Because the 4951A does not shift through the parity bit, select a data code with no parity (see page 8-11).

2. Try data codes of a different size.

3. When you are able to identify the idles, change the **Sync on** to the two sync characters immediately following the idles. Change **Drop sync after** to the idle character.

4. If **bit sense** is inverted or **bit order** reversed (eg, IPARS), you may need to go back and capture some new data with these two parameters changed.

5. Parity, block check, and frame check errors are indicated by blinking characters. Character frame length is affected both by the data code and the error checking. For example, ASCII 7 with odd parity uses an 8-bit character frame; whereas ASCII 8 with odd parity uses a 9-bit frame. See page 8-11.

Data Code	No Parity	Even or Odd Parity	Ignore Parity
Hex 5 Baudot	5 bits (no parity bit)	6 bits (includig parity bit)	6 bits (parity bit = 0)
Hex 6 EBCD IPARS Transcode	6 bits (no parity bit)	7 bits (including parity bit)	7 bits (parity bit = 0)
Hex 7 ASCII 7	7 bits (no parity bit)	8 bits (including parity bit)	8 bits (parity bit = 0)
Hex 8 ASCII 8 EBCDIC	8 bits (no parity bit)	9 bits (including parity bit)	9 bits (parity bit = odd)

Figure 8-3. Frame Sizes vs Data Codes (*indicates forced selections for Simulate)

Error Messages

No data in buffer -- Use EXIT key to exit. This occurs if the buffer is empty when you go to the Examine Data Menu. Monitor On-Line, or load from the tape to fill the buffer.

No displayable data in buffer for the selected display format. This indicates that the buffer contains non-displayable data, such as lead transitions. Use Data & State display format to see the lead transitions.

Tape removed during a Read operation. When you remove the tape during a load operation.

Tape read error: buffer data invalid. This may be caused by a broken tape controller, or by a worn out tape.

End of valid data. When you scroll to the end of buffer data.

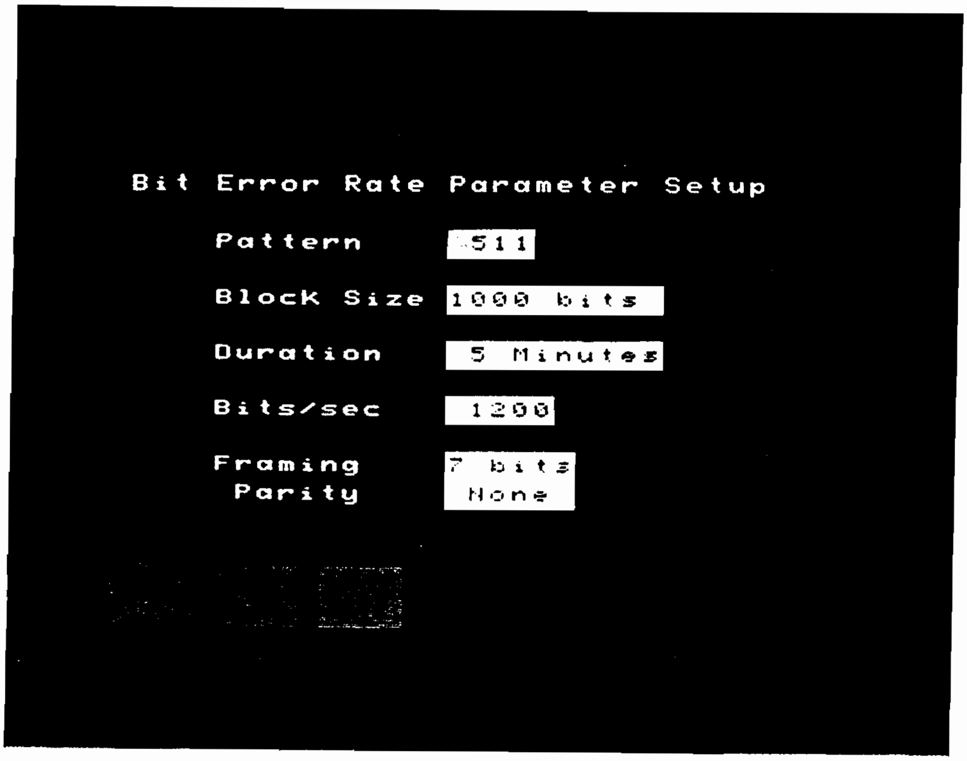
Start of valid data. When you scroll to the beginning of buffer data.

No more highlights. When you press the <Next Hilit> key and there are no more highlights.

End of tape file. When you specify a block beyond the last block on tape.

CHAPTER 9

BIT ERROR RATE TESTS (BERT)



```
Bit Error Rate Parameter Setup

    Pattern        511
    Block Size     1000 bits
    Duration       5 Minutes
    Bits/sec       1200
    Framing        7 bits
     Parity        None
```

Figure 9-1. The BERT Menu

Definitions

Bit Error Rate Tests (BERT) measure digital noise: how often "highs" are changed to "lows", and vice versa

PRBS (Pseudo Random Bit Sequence). A BERT tester generates pseudo random bit sequences from a shift register of length L, where the sequence length is N = (2**L - 1) bits. A PRBS may be of any length, but certain pattern lengths have become standard. The 4951A uses PRBS lengths of 63, 511, or 2047.

Bit Error Rate. The number of bit errors divided by the number of bits received.

Blocks. Bit error rate does not give any idea of error distribution. For example, if most errors occur within a few moments of each other, it might indicate that the line was okay, but had perhaps been affected by a lightning hit or path switch. For this reason, bits are grouped in blocks for measuring block error rate.

NOTE: BERT "blocks" are not to be confused with the blocks used in other 4951A menus.

Block Error Rate. Block error rate is the number of block errors divided by the number of blocks received. Whether there is one error, or ten errors in a block, it is still counted as one block error.

Block Sizes. The Bell system uses a block size of 1000 bits. CCITT, the world-wide standard, uses a block size equal to the pattern size. For example, if the PRBS pattern is 511 bits, then the block size would also be 511 bits.

BERT Menu Softkeys

BERT Menu selections are shown on page 9-5. (Press BERT on the Top Level Menu).

Pattern. Three PRBS pattern lengths are available: 2047, 511, and 63 bits.

Block Size. Two selections are available: 1000 bits and CCITT specification. The 1000-bit block size is used in the US; and CCITT is used in other countries. When CCITT is selected, block size is always the same as pattern size.

Duration. You can select the length of the test either as a time interval, or as the number of bits sent. For later comparison, test durations must be the same.

Bits/Sec. Notice the Bits/Sec selections are different from the other menus.

Framing. Framing means that you will send standard asynchronous characters with one start bit and two stop bits. Thus, the frame size is equal to the start and stop bits, plus an optional parity bit, plus the selected character size. To select framing, choose the size of the data character (5, 6, 7, or 8 bits). An optional parity bit may be added immediately after the data character, before the two stop bits. Each frame alternates with an idle (high) time which is the same length as the frame. If you do not wish framing, press <None>.

Parity. If you select framing, three new softkey choices will appear. You can select odd or even parity, or have no parity bit at all.

BERT Menu Selections

```
Pattern      [ 2047   ]
             [  511   ]
             [   63   ]

Block Size   [  1000 bits    ]                      Framing    [ None     ]
             [ CCITT spec    ]                                 [ 5 bits   ]
                                                               [ 6 bits   ]
Duration     [ 10^4 ]  [    10^9        ]                      [ 7 bits   ]
             [ 10^5 ]  [   5 min        ]                      [ 8 bits   ]
             [ 10^6 ]  [  10 min        ]
             [ 10^7 ]  [  15 min        ]           Parity     [ None     ]
             [ 10^8 ]  [    Cont        ]                      [ Odd      ]
                                                               [ Even     ]

Bits/sec     [ 19200 ]    [ 3600  ]  [   150   ]
             [  9600 ]    [ 3200  ]  [ 134.5   ]
             [  7200 ]    [*2000  ]  [   110   ]    * 2k works only with framing
             [  4800 ]    [ 1800  ]  [    75   ]
             [  2400 ]    [  600  ]  [    50   ]
             [  1200 ]    [  300  ]  [   EXT   ]
```

Running a BERT Test

Run Menu. After you have entered the test parameters in the BERT Menu, press <Run Menu>. In the Run Menu press <BERT>.

Data Screen. When you press <BERT> in the Run Menu, a run-time data screen continuously displays test progress. The data screen shows elapsed seconds since synchronization, number of bits and blocks sent, number of errors found, and the number of errored seconds.

Completion of a Test. When a receiving BERT tester receives all the bits required for the test -- or you press EXIT -- the receiver stops the test. The transmitter continues to transmit, ensuring that the other receiver gets all needed test bits.

% Error-free Seconds. When the receiver is finished -- or you press EXIT -- % error-free seconds is computed.

Exit Key. EXIT halts reception. Press EXIT again to return to the Top Level.

Setup Summary. During a test, press <Summary> to look at the setup parameters without stopping the test. To change any of the setup parameters, stop the test by pressing EXIT twice and re-entering the BERT Menu.

Data Screen Definitions

Elapsed Seconds. Elapsed time since receiver synchronization.

Errored Seconds. Tells how many of the elapsed seconds had error occurrences.

% Error-Free Seconds. Errored Seconds divided by Elapsed Seconds. Displayed at the end of the test.

Block Count. Tells how many blocks have been sent thus far in the test.

Block Errors. Tells how many blocks had at least one error. Divide block errors by block count to get Block Error Rate.

Bit Count. The number of actual data bits sent since synchronization (excluding framing, start, stop, and parity bits).

Bit Errors. Divide bit errors by bit count to get Bit Error Rate.

Inject Error. Press <Inject Error> at any time during the test. The receiver at the other end should indicate one bit error. This function can be used at the beginning of the test to check for proper hookup.

Inject 10 Errors. This is a way of sending a burst of errors. The receiver at the other end should have counted ten bit errors, one or two block errors, and one or two errored seconds.

Requirements

SYNCHRONIZATION

Unless the BERT receiver is synchronized to the transmitter at the other end, the receiver has no way of knowing whether the next bit in the received PRBS pattern is correct. You should use BERT testers equivalent to the 4925A which have the following characteristics:

(1) For unframed patterns, the speed of the clock generating the transmitter pattern must be within 1% of the clock generating the receiver pattern. (In framing, data characters are enclosed in start and stop bits, and alternate with idle characters.)

(2) With framing, the clocks should be within 5% of each other.

HANDSHAKING

For BERT testing the 4951A simulates a DTE. At the beginning of the test, the 4951A sets the RTS and DTR interface leads "on" (For-RS 449 interfaces, it sets RS and DS on).

Error Messages

There are two possible error messages (both faults are automatically recoverable).

Out of lock -- data fault: The tester couldn't synchronize at the beginning of the test because of a wrong pattern, or the absence of data.

Out of lock -- sync loss: The tester lost synchronization during the test.

Examples

EXAMPLE 1: End-to-End Testing

Two BERT testers are connected to opposite ends of the line. Each BERT tester contains both a transmitter and a receiver, making it possible to check both send and receive channels simultaneously. The transmitter at each end is essentially a PRBS generator; the receivers are pattern checkers.

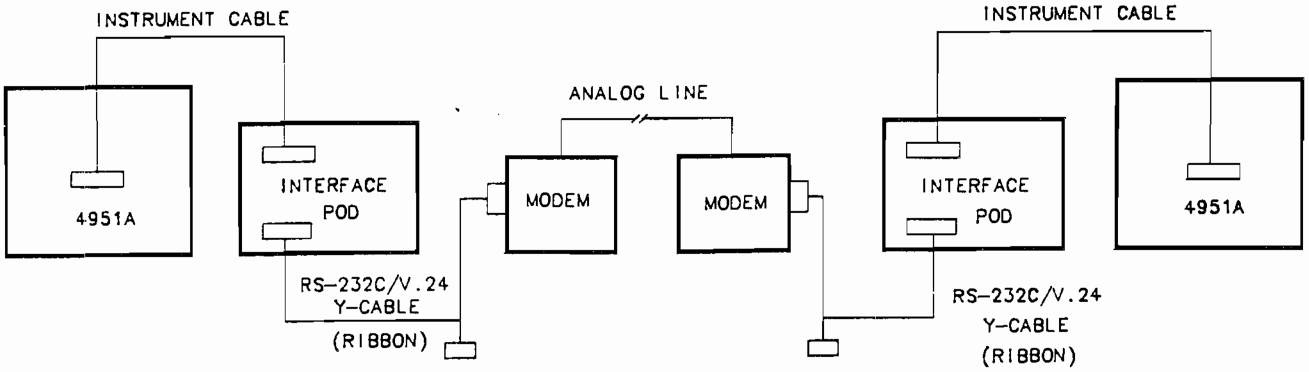


Figure 9-2. End-to-End Testing

EXAMPLE 2: Loopback Testing

If we have only one BERT tester, we can loop back at the other end. The BERT tester will send on one channel, and receive its own transmission on the other channel. Of course, if we loop back, we will be adding together the errors on both the send and receive channels: one channel may contain many more errors than the other channel.

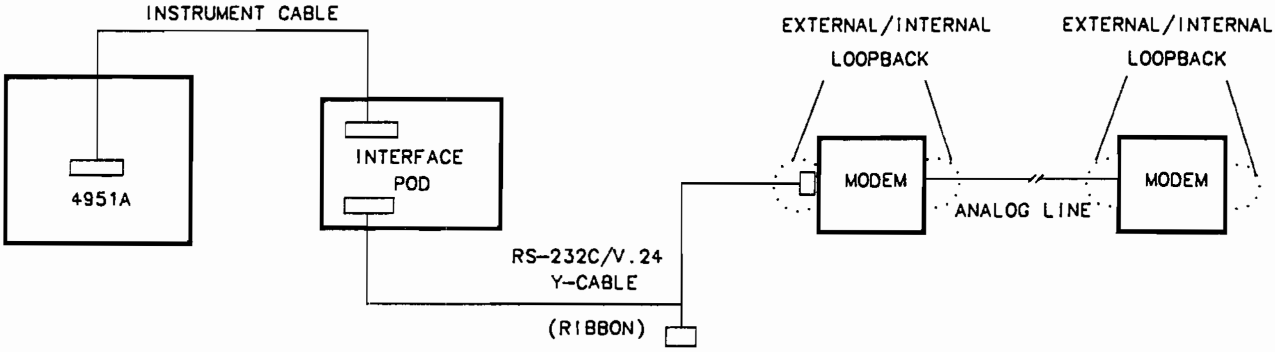


Figure 9-3. Loopback Testing

OTHER TESTS

Some BERT testers (such as the HP 4925A) perform the following character error checking besides BERT tests.

Quick Brown Fox Tests

The "Quick Brown Fox" message (or any message) tests the ability of terminals to receive messages. Use the Simulate Menu to run this test. The 4951A will, of course, also check parity errors. See Appendix F.

Startup Tests

The 4951A will do many types of start-up tests, such as RTS - CTS delay. Use the Simulate Menu. See Appendix F.

CHAPTER 10

THE REMOTE MENU

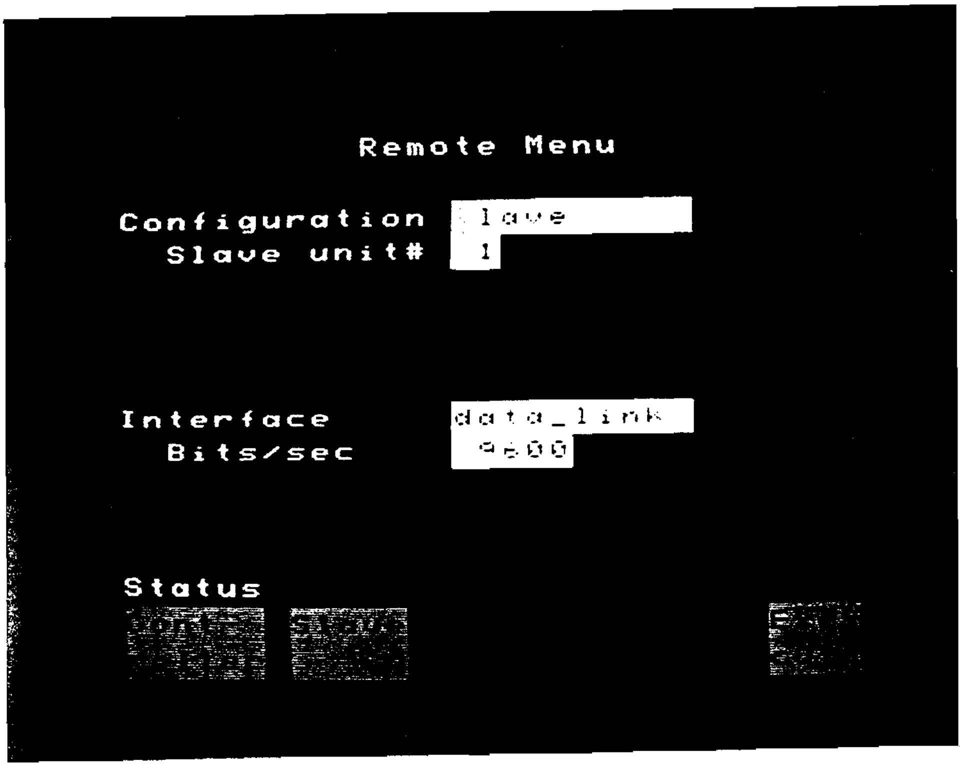
Remote Operations With the 4951A

The 4951A can transmit and receive menus and buffer memory to another 4951A or a 4955A via the data link interface. Application programs can also be transferred between two 4951A's.

For remote operations, the 4951A must be in the Remote Menu. While in the remote menu, the 4951A cannot use any other menus.

If a service person at a remote site is having trouble, the following procedure can be used to send the data and menu setups to the datacom center for study:

1. Using the interface pods, connect the 4951A (4955A) at each end of the line to a modem.
2. On both 4951A's select <Remote> in the Top Level Menu.
3. At the datacom center select <Controller> configuration, <Slave's ID> operation, and the correct bit rate. Press <Execute> to raise DTR to the controller modem.
4. At the remote site, select <Slave> configuration, the address, and bit rate. Press <Execute> to raise DTR to the remote modem.
5. Estabish phone communication between the remote site and the datacom center. Connect both modems to the phone line.
6. At the datacom center, again <Execute> the <Slave's ID> operation. This is necessary to synchronize remote transfers.
7. Transfer operations can now be performed.



Remote Menu

Configuration lave
Slave unit# 1

Interface data_link
Bits/sec 9600

Status

Figure 10-1. Remote Menu -- Slave Configuration

Using the 4951A as a Controller

CONTROLLER OPERATIONS

As a Controller, the 4951A downloads menus and data to the Slave. The 4951A can also receive uploaded information from the Slave.

Upload Menus. Receives setup, monitor, simulate, and run menus.

Upload Data. Receives buffer data. The Controller can specify memory block numbers: Start Block n1, End Block n2.

Upload Appl. Receives an application program from the slave's Application Memory.

Download Menus. Transmits setup, monitor, simulate, and run menus.

Download Data. Transmits buffer data. The Controller can specify memory block numbers: Start Block n1, End Block n2.

Download Appl. Transmits an application program from its Application Memory to that of the slave.

Slave Status. Requests the slave to transmit its current status.

Slave ID. This controller command is necessary to synchrononize remote transfers after initial hook-up. The slave will transmit "HP 4951A".

REMOTE MENU -- CONTROLLER CONFIGURATION

```
Configuration      Controller

Operation          [ Upload Menus     ]    Start Blk  [   ]   End Blk  [   ]
                   [ Upload Data      ]
                   [ Upload Appl      ]
                   [ Slave Status     ]
                   [ Slave ID         ]
                   [ Download Menus   ]
                   [ Download Data    ]
                   [ Download Appl    ]

Interface          Data link

Bits/sec           [ 19200  ]
                   [ 9600   ]
                   [ 4800   ]
                   [ 2400   ]
                   [ 1200   ]
                   [ 600    ]
                   [ 300    ]

Status             Operation executing
                   Operation successful
                   Slave rejected operation
                   Slave not responding
```

Using the 4951A as a Slave

4951A SLAVE RESPONSES

A slave 4951A will respond to any of the previously listed commands from a controller 4951A. When in the remote menu, a slave 4951A will transmit or receive menus, data, or application programs to the controller. Also, any error condition which occurred during the transfer can be obtained by the controller **Slave Status** controller command.

4955A CONTROLLER COMMANDS

Application programs cannot be transferred between a 4955A and a 4951A. Other than this exception, a 4951A will respond to all the above commands from a Controller 4955A.

The following operations are different in operations between a 4955A Controller and a 4951A Slave.

Upload Timers & Counters

Upon receiving this command from a 4955A controller, a slave 4951A will upload the status of its timers and counters.

Address

The 4955A specifies an address in all controller commands. The slave address must be selected to be the same. The address does not matter in operations between two 4951A's.

REMOTE MENU -- SLAVE CONFIGURATION

Configuration	Slave	
Slave Addr #	[Use Keyboard]	(Only 4955A's can request an address)
Interface	Data link	
Bits/sec	[19200] [9600] [4800] [2400] [1200] [600] [300]	
Status	Operation executing Operation successful Slave rejected operation Slave not responding	

Ending Remote Operations

To stop execution of any remote operation, press EXIT. If your press EXIT again the 4951A will display the following message:

To Disconnect the Remote Link,
press the HANG UP softkey,
otherwise press EXIT

Pressing EXIT will return you to the Top Level Menu. You can then go back to the Remote Menu at any time and perform any operation.

Pressing <Hang Up> will turn off DTR.

If you press <Hang Up>, you must re-enter the Remote Menu and again press <Execute> at both ends of the line to raise DTR.

Handshaking Requirements

OPERATIONS WITH A MODEM

In remote operations, the 4951A is configured as a DTE. The following handshaking convention is used. (See also page 10-2).

1. At both ends of the line, press <Execute> before establishing a call with the modem. This turns on DTR, which stays on until pressing EXIT twice and selecting <Hangup>.

2. After establishing phone communication and connecting the modems to the line, the Controller must <Execute> the <Slave's ID> operation to synchronize remote transfers.

3. At the beginning of a Remote operation, the 4951A sets RTS on.

4. The 4951A will not send data to the modem until CTS goes on.

5. The 4951A will not look for data from the modem until CD goes on.

6. To turn off DTR, press EXIT twice, and then <Hang Up>.

OPERATIONS WITHOUT MODEMS

The 4951A is configured as a DTE for remote operations. If remote operations are carried out via modem, there is no problem. However, if two units are connected directly, they cannot talk to each other: one must be a DCE. For applications with no modem, use a modem eliminator cable. The RS-232C/V.24 printer cable M/M (HP part no. 13242G) can be used for this application.

Slave Error Messages

Buffer Size Too Small. The controller is trying to download too much.

Start block# must = first. The controller has not specified the first block in the slave buffer. Note that the first block may not be "1" if the buffer data has been loaded from tape.

No data in requested blks. The controller has requested data from empty blocks.

Buffer empty. The slave buffer is empty.

Conversion error: menus reset. This might occur if the menus being transferred are invalid.

Menus incompatible with 4951A. This might occur for certain menus created by a 4955A.

Modem handshake fails. The controller RTS, CTS handshaking has failed

Invalid Mon/Sim Menu. This can occur if you say "When DTE/DCE" and then do not specify a trigger.

Operation not valid for 4951A. The operation is one that only a 4955A can perform.

Issue ID request to enable slave. You must always <Execute> this operation immediately after establishing phone communication in order to sychronize remote transfers.

CHAPTER 11
THE MASS STORE MENU

Introduction

The tape drive is optional (Option 001): you may buy the 4951A with or without a tape drive. Of course, the tape drive can be later retrofitted in the field.

WHY USE THE TAPE?

Because of the versatile triggering capability of the 4951A, you will usually be able to find a problem before using all the memory. However, the tape drive has several advantages:

Store data directly from the line onto tape; this increases your buffer memory 16 times (see page 11-9).

Save all the menus and the buffer data for future reference. In the BERT Menu only the setup (not the results) is saved.

Use the tape like a large buffer. The 4951A nonvolatile buffer memory holds 32K bytes of data. A tape cartridge holds up to 512 Kbytes. The tape increases the virtual buffer size 16 times. In the Examine Data Menu, you can use the <Next Tape Segment> and <Previous Tape Segment> commands to view tape data.

Transfer menus and data between a 4951A and a 4955A. 4951A tapes are compatible with 4955A tapes.

Load future application programs using the tape.

HOW TO USE THE TAPE DRIVE

Type of Cartridges

Specify HP 98200A to order a set of five certified blank cartridges.

Care of Cartridges

If the cartridge is placed near a a strong magnetic field, some data on the tape may be lost. If many Time and Sync errors occur during tape load and store operations, use the <Tension> softkey in the Mass Store Menu; also check the tape system for dirty heads.

Inserting Cartridges

Insert the tape cartridge with the RECORD slide located to the top right.

The RECORD slide is used to protect previously recorded tapes from being overwritten. To store data to tape, push the slide in the direction of the arrow. You may load (read) from the tape with the slide in either position.

Compatibility

Data and Menu files developed on the 4955A are compatible with the 4951A.

The Mass Store Menu

The Mass Store Menu is used for tape operations. You cannot use this menu if you do not have the optional tape drive installed. The following softkeys appear when you press <Mass Store> in the Top Level Menu:

Tension Initialize Catalog Load Store

TENSION

Rewinds the tape to the correct tension. The tape is first rewound, then fast-forwarded, then rewound. This may be used if the tape is noticeably slack or if a large number of read errors are occurring (See page 11-11 for tape error messages). You can do other operations, such as menu setup, etc, while the tape is tensioning -- no need to wait.

INITIALIZE

Erases the tape directory and formats the tape with a new directory. This must always be used for blank tapes. Do not initialize any tape you want to keep. Initializing erases the tape directory and you will irrevocably lose any information on that tape.

CATALOG

The catalog operation displays the tape directory giving File Name, File Type, and a Comment field. Five file types are possible:

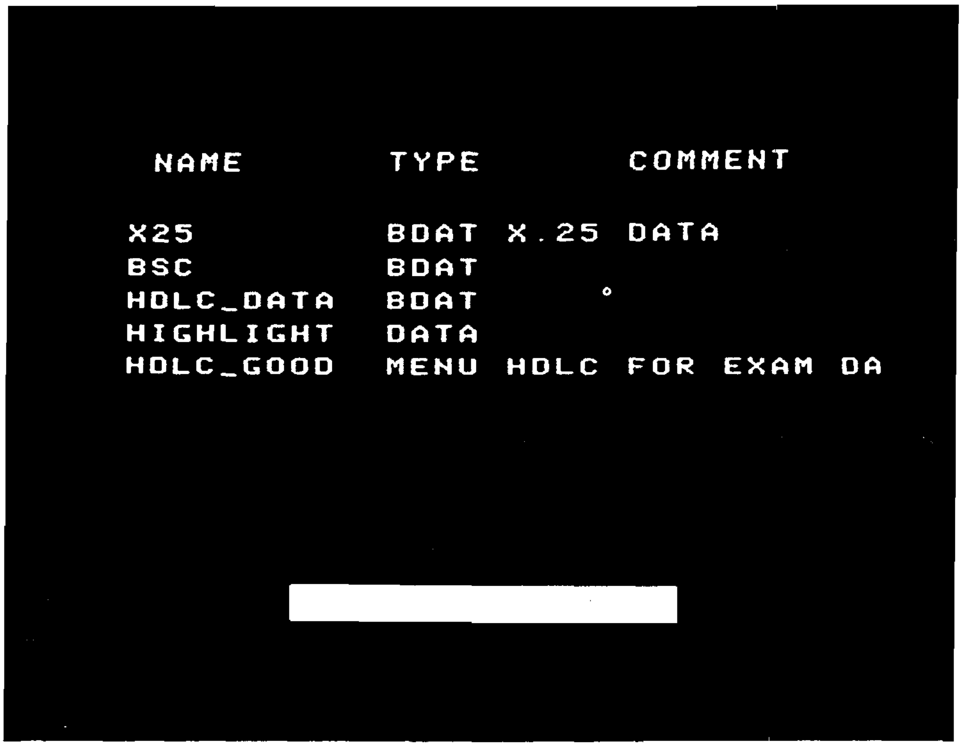
DATA	Both buffer data and menus (setup, monitor, simulate, BERT).
MENU	Consists of menus only (setup, monitor, simulate, BERT).
BDAT	Buffer data only. This 4955A file type may be loaded into a 4951A.
APPL	Only 4951A application files can be loaded.
BASI	This 4955A file type may not be loaded into the 4951A.

Notice that only two types of files are generated by the 4951A: DATA and MENU. Data type files always contain both data and menus. Menu type files contain only menus. The 4951A will load BDATA files which have been generated by the 4955A.

The Comment field is 32 characters long, but only the first 17 are displayed. All 32 are displayed on the 4955A.

Group File Names

The 4955A allows Group file names, but the 4951A ignores them. If two files generated by a 4955A have the same file name (but belong to different groups), the 4951A will list both files in the catalog. Only the first file listed can be loaded.



```
 NAME        TYPE       COMMENT

X25          BDAT  X.25 DATA
BSC          BDAT
HDLC_DATA    BDAT
HIGHLIGHT    DATA
HDLC_GOOD    MENU  HDLC FOR EXAM DA
```

Figure 11-1. A Catalog Listing

Loading From Tape

HOW TO LOAD DATA INTO THE BUFFER

Insert the tape cartridge with the file to be loaded into the tape slot. Go to the Mass Store Menu and press <Catalog> to see whether the file is on the tape (optional). Now press <Load> and type in the file name as it is listed in the catalog. Press <Execute> to load the file into the memory. The 4951A will accept all 4955A file types except Basic and 4955A Application Programs.

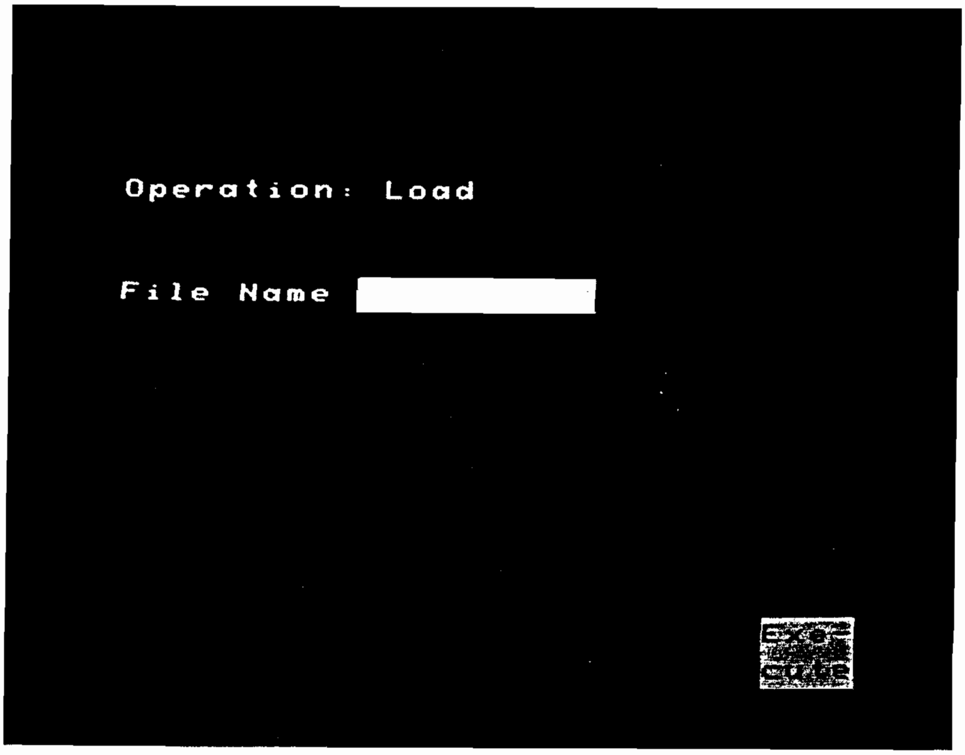
Menu Changes

CAUTION: DO NOT PERFORM THE LOAD OPERATION IF YOU WANT TO SAVE PRESENT MENU SETUPS.

Unless the file is of type BDAT (generated by a 4955A), the 4951A menu setups will be changed by the load operation. The Setup, Monitor, Simulate, and BERT setups will all be modified to the new values. You must first store these menus to another tape if you want to save them.

OPERATIONS WITH LARGE FILES

When loading a tape file that is too large for the buffer, you can scroll through the rest the file by using the <Next Segment> or <Previous Segment> softkeys in the Examine Data Menu. These softkeys scroll through the file in 16 Kbyte segments (1/2 the buffer size).



Operation: Load

File Name

Figure 11-2. The Load Operation

Storing to Tape

WHAT YOU CAN STORE ON TAPE

You can store both "Menus and Data", or "Menus" only. The menus saved are: Setup, Monitor, Simulate, and BERT. BERT setups only are saved; results are not stored.

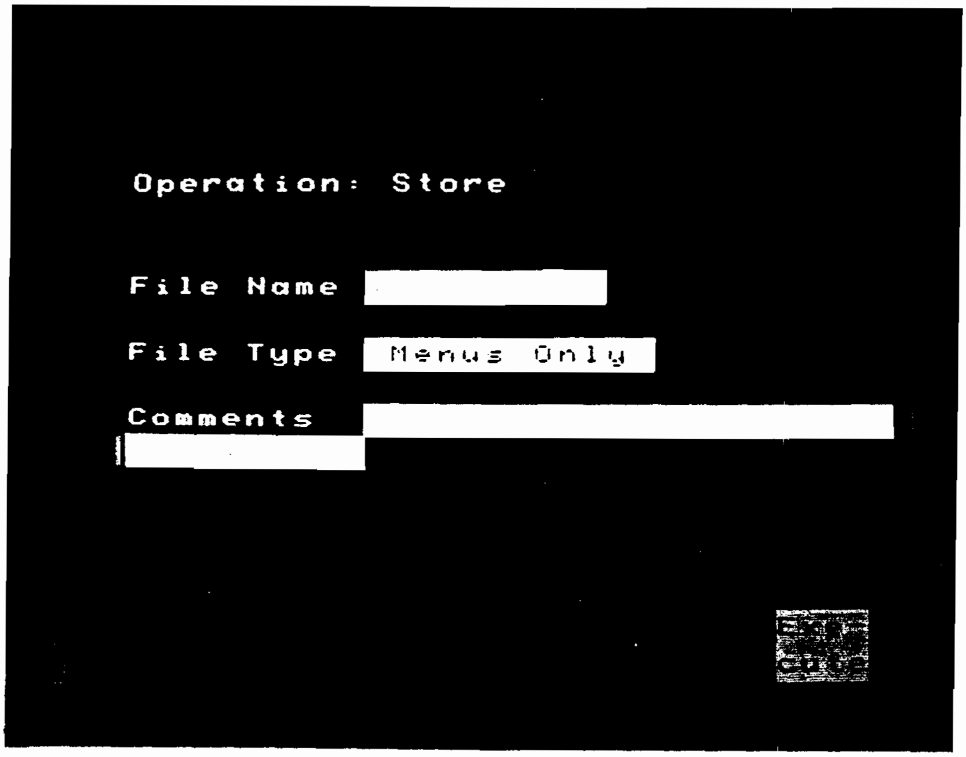
HOW TO STORE TO TAPE

Insert a tape cartridge into the tape slot. Press <Store>, enter the file name, the file type, and an optional comment; then press <Execute>. If the tape cartridge has insufficient room for the file being stored, "EOT Error" (end of tape) will be displayed. If the tape cartridge is blank, it must first be initialized to create a tape directory. This is similar to formatting a disk.

STORING LINE DATA DIRECTLY

Data can be directly stored to tape as you run from the line. In the monitor or simulate menus, use <Start tape> and <Stop tape> in your program instructions to to start or stop the tape at anytime under program control. NOTE: You can only start and stop the tape once by program control.

A tape cartridge contains two tracks, each of which can hold up to 256K bytes of data. The tape can keep up at line bit rates of 9600 bps full duplex and 19.2K bps half duplex. Switching tracks takes about 20 seconds, however, and you may "overflow" the buffer" at this time. The tape is able to keep up at higher line bit rates if line utilization (percentage of data to idles) is low.



Operation: Store

File Name

File Type Menus Only

Comments

Figure 11-3. The Store Operation

Mass Store Error Messages

Invalid File Name. The first character of a file name must always be an upper case letter. After the first character, use any combination of upper and lower case letters, numbers, or underscores (__).

Tape Out. The tape is out at the beginning of the operation, or if the tape is taken out during the operation.

File Not Found. The 4951A cannot find the specified file during a load operation.

Time Error and Sync Error. These errors indicate either a hardware or tape problem. Tension the tape or use a different tape. Otherwise, see the Service Manual.

Protected. This indicates the tape is "write protected". Slide the Record tab on the tape cartridge in the direction of the arrow in order to store to tape.

Despool. Tape either has run off the spool, or is about to. Wind the tape onto an empty spool. The tape drive needs service if it re-occurs

Checksum. Bit errors have occurred. Use the same procedure as for Time Errors.

Format. The tape is not in the proper format.

Servo. Motor speed is out of tolerance due to faulty motor electronics, or a jammed tape.

EOT. End of tape. The tape is full.

CHAPTER 12
RESET AND SELF TEST

The Reset Softkey

PURPOSE

The <Reset> softkey enables you to clear the memory and go back to default entries in the Setup, Monitor, and Simulate Menus.

WHEN SETUPS AND BUFFER ARE NOT SAVED

Because the 4951A has a battery backed up memory, menu setups and buffer memory data are saved after turn off. The menus saved are: Setup, Monitor, and Simulate. However, setups and buffer data cannot be guaranteed in the following two cases:

1. The instrument was not turned off when it was in the Top Level Menu.

2. The battery has completely run down. This should never happen unless the instrument has been stored for more than a week in a very hot environment (or six months at room temperature).

In these cases the following message appears when you go to the monitor, simulate, or setup menus.

MENUS CORRUPT; MENUS HAVE BEEN
RESET TO THE DEFAULT CONDITION

The analyzer has been reset automatically: the buffer has been cleared, and setups return to their default values. Always press EXIT and go to the Top Level Menu before turning the instrument off.

Setup Menu Defaults

The following tables list default entries for the five Setup Menus: HDLC, SDLC, X.25, BSC, and Char Async/Sync. Whenever you press <Reset> on the Top Level Menu, these entries will appear in each menu. Otherwise, the entries will be whatever you had selected before you turned the power off.

HDLC PROTOCOL	SDLC PROTOCOL	X.25 PROTOCOL
External Address Off		
External Control Off		
Code ASCII 8	Code ASCII 8	Code ASCII 8
	Mode Sync	
DTE clock DCE	DTE clock DCE	DTE clock DCE
Bits/sec 9600	Bits/sec 9600	Bits/sec 9600
Display mode Data & State	Display mode Data & State	Display mode Data & State

Setup Menu Defaults (cont)

```
BSC PROTOCOL                       CHAR ASYNC SYNC

                                   Bit order  LSB 1st    Bit sense  Norm

Code  ASCII 7                      Code  ASCII 7

Error check  LRC                   Error check  LRC
                                     Start on  SOH STX
                                     Stop on  & ETX NUL NUL

                                   Parity  Odd

                                   Transparent text char  None

                                   Mode  Sync
Sync on  16 16                       Sync on  32 32
                                     Drop sync  10  characters
                                       after  2D 2D 37 3D 70 7F FF

DTE clock  DCE                     DTE clock  DCE
  Bits/sec  9600                     Bits/sec  9600

Display mode  Data & State         Display mode  Data & State

Suppress  None                     Suppress  None
```

The Self Test Menu

Whenever you turn on the 4951A it will first go through a self-test. After approximately six seconds, it will then display the Top Level Menu. You can run the self-test at any time by pressing the <Self Test> softkey in the Top Level Menu. If you then press <Loop>, the analyzer will continue in the self test mode until you press EXIT. See the Service Manual for more information on the Self Test Menu.

APPENDICES

APPENDIX A
SPECIFICATIONS

Operating Characteristics

Protocols

X.25, HDLC, SDLC (NRZI), BSC, and most character asynchronous or synchronous protocols.

Data Transfer Rates (bps)

50, 75, 100, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3200, 3600, 4800, 7200, 9600, 12000, 14400, 16000, 19200, teletext 1200/75, and EXTERNAL up to 19200 full duplex for all monitoring, simulation, triggering, and BERT tests.

The HP 4951A can capture a complete buffer full of data at line speeds up to 64 kbps. (Bit oriented protocols only).

Clock Accuracy: 0.005%

Data Codes

ASCII, EBCDIC, Baudot, Six Bit Transcode, IPARS, and EBCD.

Mass Storage Memory

32 Kbytes of RAM stores data characters, timing, and lead status information.

Optional tape drive: Up to 512 Kbytes for storing data, timing information, menu configurations, and application programs. Write to tape: 9600 bps full duplex, 19200 bps half duplex.

Lead Status

The status of five control leads are stored for each interface. They are RTS, CTS, DTR, DSR, and CD for RS-232C/V.24, and CS, RS, RR, TR, and DM for RS-449.

Character Framing

5, 6, 7, or 8 information bits, plus parity. For asynchronous systems select 1, 1.5, or 2 stop bits per character.

Error Checking

CRC-CCITT, CRC-16, CRC-12, CRC-6, LRC, and parity.

Triggers

63 triggers consisting of characters, errors, interface lead transitions, or timer values. All be simultaneously active up to 19200 bps.

Timers

Five timers, each of which has a maximum count of 65565 msec. Resolution 1 msec.

Counters

Five counters, each of which can be incremented up to 9999.

Keyboard

Full ASCII keyboard with six softkeys and cursor control.

Display

12.7 cm (5 in.) diagonal with 16 lines and 32 characters per line.

Display Formats

Five: DTE only, DCE only, DTE over DCE, Data and State, and Frame and Packet.

Send Strings

255 characters per string maximum, 1750 characters total.

Remote Capability

Over the RS-232C/V.24 link: transfer data, setups, and programs.

Self Test

Extensive self test and verification routines for isolating failures to a functional component group. Built-in signature analysis permits fault isolation to the component level.

Bit Error Rate Testing

Simultaneously measure bit errors, block errors, errored seconds, and percent error free seconds.

Block Size: 63, 511, 1000, or 2047 bits.

Patterns: 63, 511, or 2047 bit pseudo random sequence.

Character Framing: Select 5, 6, 7, 8 bits per character and parity, or none (continuous). Select odd or even parity with character framing, or none with no framing.

Inject Errors: Inject single errors or bursts of errors.

Additional Characteristics

Auto-configuration of all setup parameters.

Battery maintained RAM for all setups, data, and menus.

Select bit order as LSB or MSB first and select bit sense as inverted or normal.

INTERFACE ACCESSORIES

HP 18173A, HP 18174A, and HP 18180A. Each interface is supplied with the appropriate 1.5 meter "Y" cable.

HP 18173A (RS-232C/V.24) Interface

Ten switches for line isolation. 25 test points for monitoring, forcing, or cross-patching. One nondedicated MARK/SPACE monitor for user patching to any line. Nine hard-wired activity indicators: TD - Transmit Data, RD - Receive Data, TC - Transmit Clock, RC - Receive Clock, DTR - Data Terminal Ready, DSR - Data Set Ready, RTS - Request to Send, CTS - Clear to Send, and CD - Carrier Detect.

Mark/Space Monitor: Indicates voltages greater than +3.0 volts, and voltages less than -3.0 volts.

Interface Activity Indicators: These turn on at voltages greater than +2.75 volts, and off at voltages less than +0.25 volts.

Voltage Source: +/- 12 volts.

Weight: 0.6 kg (1.3 lb).

HP 18174A (RS-449) Interface

Nine dedicated activity indicators: SD - Send Data, RD - Receive Data, ST - Send Timing, RT - Receive Timing, RS - Request to Send, CS - Clear to Send, TR - Terminal Ready, DM - Data Mode, and RR - Receiver Ready.

Interface Activity Indicators: These turn on when the differential voltage is greater than 0.2 volts.

Weight: 0.6 kg (1.3 lb).

HP 18180A (Combination RS-232C/V.24 and RS-449 Interfaces).

Weight: 0.7 kg (1.5 lb).

Specifications

Weight

Net: 5.7 kg. (12.6 lbs.)
Shipping: 9.5 kg. (21 lbs.)

Size

Height: 11.2 cm, width 25.5 cm, depth 28.6 cm. (4.4 x 10.2 x 11.3 in.)

Temperature

Operating: 0 C to +55 C (+32 F to +131 F) **
Storage: -40 C to +75 C (-40 F to +167 F)

** The tape drive should only be operated from +5 C to +40 C (+41 F to +104 F).

Altitude

Operating: 4600 m (15000 ft).
Storage: 15300 m (50000 ft).

Power Requirements

110, 220 Vac, -15% to +15%; 48 to 66 Hz single phase.
Typical less than 15 VA, maximum less than 30 VA.

APPENDIX B

ACCESSORIES

ACCESSORIES

ACCESSORIES SUPPLIED

Power Cord (see the service manual for power cord part numbers)	
Pod-Instrument Cable (for all interface pods)	HP 04951-61604
Operating Manual	HP 04951-90003
Service Manual	HP 04951-90002

Interface pods, listed below, are not supplied and must be ordered separately.

ACCESSORIES AVAILABLE

18173A	RS-232C/V.24/MIL-188C Interface Pod
18174A	RS-449 Interface Pod
18180A	Combination RS-232C/V.24 and RS-449 Interface Pod
18172A	Soft vinyl carrying case for extra pods
98200A	Certified blank tape cartridges (set of five)

OPTIONS

Option 001	Integral Tape Unit
Option 100	Adds accessory 18173A
Option 101	Adds accessory 18174A
Option 102	Adds accessory 18180A
Option 910	Extra operating and service manuals
Option 916	Extra operating manual

APPENDIX C

THE INTERFACE

The RS-232C/V.24 Interface
The RS-449/422A/423A Interface

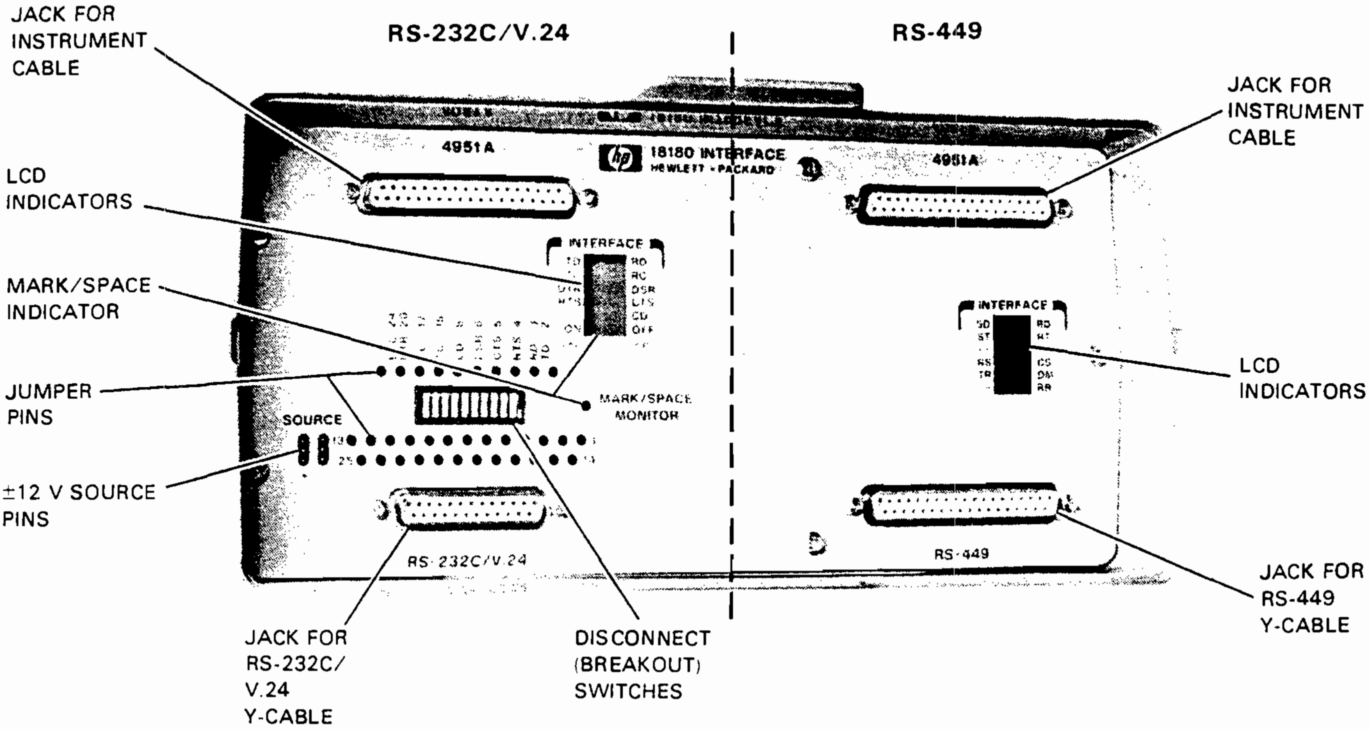


Figure C-1. The HP 18180A (Combination RS-232C/V.24 and RS-449 Interface)

RS-232C/V.24 Interface (HP 18173A or HP 18180A)

JACKS

The top jack, labeled 4951A, connects the interface pod to the 4951A via the Pod-Instrument cable supplied with the instrument. The bottom jack, labeled RS-232C/V.24 connects the Interface to the line for monitoring or simulation (see chapter 2 for Hookup).

JUMPER PINS

All 25 pins of the bottom jack are brought out for jumpering. If your network cable has different pin assignments from the interface standard, you can use the supplied jumper wires to connect the interface lines to the desired pin on your cable.

Pins 2, 3, 4, 5, 8, 15, 20, and 24 are also brought out on the other side of the breakout switches for jumpering.

SOURCE PINS

The six Source Pins supply +12 volts and -12 volts. You may set any signal line on or off by jumpering that line to the Source Pins.

DISCONNECT (BREAKOUT) SWITCHES

Pins 2, 3, 4, 5, 6, 8, 15, 17, 20, and 24 may be individually disconnected from the data link by means of switches. This lets you isolate non-driven interface lines from the 4951A. Non-driven lines may develop cross talk noise which will be mistaken by the analyzer for transitions.

LCD INDICATORS

The LCD indicator for a signal line is dark when that line is On or Spacing. The LCD indicator is blank when a line is Off, Marking, or tri-stated. For the indicator to be dark the voltage on that line must be greater than +2.75 volts. Once the indicator is dark, it will not go blank until the voltage becomes less than +0.25 volts. Therefore, the LCD for individual lines do not distinguish Marking and tri-state. Use the Mark/Space Monitor to do this.

LCD Indicator	Interface Line
Dark	Logical "0" (Spacing, ON, positive voltage)
Blank	Logical "1" (Marking, OFF, negative voltage, or tri-state)

MARK/SPACE MONITOR

The Mark/Space Monitor Pin enables you to check the level of any signal line. Jumper this pin to any RS-232C/V.24 signal pin and observe at the ON/OFF LCD indicators. If the line is greater than +3 volts, the On indicator will be darkened. If the line is less than -3 volts, the Off indicator will be darkened. The other LCD indicators do not distinguish between Marking and tri-state conditions (they are blank below +0.25 volts). The Mark/Space Monitor lets you check these lines, or any other signal lines for a mark/space levels.

18173 INTERFACE (RS-232C/V.24)

Pin	Circuit	Function	EIA	CCITT	Source
1	GND	Protective Ground	AA	101	...
2	TD	Transmitted Data	BA	103	DTE
3	RD	Received Data	BB	104	DCE
4	RTS	Request To Send	CA	105	DTE
5	CTS	Clear To Send	CD	106	DCE
6	DSR	Data Set Ready	CC	107	DCE
7	GND	Ground Signal	AB	102	...
8	CD	Carrier Detect	CF	109	DCE
9-11	...	unassigned	...	...	...
12	SCD	Sec Carrier Detect	SCF	122	DCE
13	SCS	Sec Clear To Send	SCB	121	DCE
14	STX	Sec Transmitted Data	SBA	118	DTE
15	TC	Transmit Clock	DB	114	DCE
16	SRD	Sec Received Data	SBB	119	DCE
17	RC	Received Clock	DD	115	DCE
18	...	unassigned	...	...	...
19	SRS	Sec Request to Send	SCA	120	DTE
20	DTR	Data Terminal Ready	CD	108.2	DTE
21	SQ	Signal Quality	CG	110	DCE
22	RI	Ring Indicator	CE	125	DCE
23	DRS	Data Rate Selector	CH	111	DTE
			CI	112	DCE
24	ETC	Ext Transmit Clock	DA	113	DTE
25	...	unassigned	...	...	...

The RS-449 Interface (HP 18174A or HP 18180A)

The HP 18174A (or combination HP 18180A) follows EIA RS-449/422A/423A electrical, mechanical, functional and procedural specifications. RS-449 was intended by the Standards Committees as a replacement and enhancement for the RS-232C/V.24 interface and can be used for both low and high-speed applications. RS-449 is made up of two electrical standards, RS-423A and RS-422A.

RS-422A uses a balanced signal lead configuration for data and clocks to enable high speed operation. RS-423A uses an unbalanced signal lead configuration. Because the 4951A implements RS-422A electrical standards for all category I circuits, it can also support RS-423A circuits.

As you can see from Figure 13-1, the 18174A (or combination 18180A) interface does not have an integral breakout box for disconnecting and jumpering lines. Selected pins are, however, monitored by LCD indicators. For the LCD indicators to transition, the unbalanced or differential A-B voltage must be greater than 0.2 volts.

18174 INTERFACE (RS-449)

Pin	Circuit	Function	Pin	Circuit	Function
1	SHIELD		20	RC	Receive Common
2	SI	Signaling Rate Indicator	21	Spare	
3	Spare		22	SDb	Send Data (b)
4	SDa	Send Data (a)	23	STb	Send Timing (b)
5	STa	Send Timing (a)	24	RDb	Receive Data (b)
6	RDa	Receive Data (a)	25	RSb	Request Send (b)
7	RSa	Request to Send (a)	26	RTb	Receive Timing
8	RTa	Receive Timing (a)	27	CSb	Clear to Send
9	CSa	Clear to Send (a)	28	IS	Terminal in Service
10	LL	Local Loopback	29	DMb	Data Mode (b)
11	DMa	Data Mode (a)	30	TRb	Terminal Ready (b)
12	TRa	Terminal Ready (a)	31	RRb	Receiver Ready (b)
13	RRa	Receiver Ready (a)	32	SS	Select Standby
14	RL	Remote Loopback	33	SQ	Signal Quality
15	IC	Incoming Call	34	NS	New Signal
16	SF/SR	Select Frequency/rate	35	TT	Terminal Timing
17	TTa	Terminal Timing (a)	36	SB	Standby Indicator
18	TM	Test Mode	37	SC	Send Common
19	SG	Signal Ground			

APPENDIX D
DATA CODES

ASCII Character Conversion Table

Dec Value	Binary	Hex	Displayed Character	Keyboard Mnemonic	Description	Dec Value	Binary	Hex	Displayed Character	Keyboard Mnemonic	Description
0	000 0000	00	NU	NUL	Null	30	001 1110	1E	RS	RS	Record Separator
1	000 0001	01	SH	SOH	Start of Header	31	001 1111	1F	US	US	Unit Separator
2	000 0010	02	SX	STX	Start of Text	32	010 0000	20	(space)		
3	000 0011	03	EX	ETX	End of Text	33	010 0001	21	!		
4	000 0100	04	ET	EOT	End of Transmission	34	010 0010	22	"		
5	000 0101	05	EQ	ENQ	Enquiry	35	010 0011	23	#		
6	000 0110	06	AK	ACK	Positive Acknowledge	36	010 0100	24	$		
7	000 0111	07	BL	BEL	Bell	37	010 0101	25	%		
8	000 1000	08	BS	BS	Back Space	38	010 0110	26	&		
9	000 1001	09	HT	HT	Horizontal Tab	39	010 0111	27	'		
10	000 1010	0A	LF	LF	Line Feed	40	010 1000	28	(		
11	000 1011	0B	VT	VT	Vertical Tab	41	010 1001	29	)		
12	000 1100	0C	FF	FF	Form Feed	42	010 1010	2A	*		
13	000 1101	0D	CR	CR	Carriage Return	43	010 1011	2B	+		
14	000 1110	0E	SO	SO	Shift Out	44	010 1100	2C	,		
15	000 1111	0F	SI	SI	Shift In	45	010 1101	2D	-		
16	001 0000	10	DL	DLE	Data Link Escape	46	010 1110	2E	.		
17	001 0001	11	D1	DC1	Device Control 1	47	010 1111	2F	/		
18	001 0010	12	D2	DC2	Device Control 2	48	011 0000	30	0		
19	001 0011	13	D3	DC3	Device Control 3	49	011 0001	31	1		
20	001 0100	14	D4	DC4	Device Control 4	50	011 0010	32	2		
21	001 0101	15	NK	NAK	Negative Acknowledge	51	011 0011	33	3		
22	001 0110	16	SY	SYN	Synchronous Idle	52	011 0100	34	4		
23	001 0111	17	EB	ETB	End of Transmission Block	53	011 0101	35	5		
24	001 1000	18	CN	CAN	Cancel	54	011 0110	36	6		
25	001 1001	19	EM	EM	End of Medium	55	011 0111	37	7		
26	001 1010	1A	SB	SUB	Substitute	56	011 1000	38	8		
27	001 1011	1B	EC	ESC	Escape	57	011 1001	39	9		
28	001 1100	1C	FS	FS	File Separator	58	011 1010	3A	:		
29	001 1101	1D	GS	GS	Group Separator	59	011 1011	3B	;		

ASCII Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Keyboard Mnemonic	Description	Dec Value	Binary	Hex	Displayed Character	Keyboard Mnemonic	Description
60	011 1100	3C	<			94	101 1110	5E	^		
61	011 1101	3D	=			95	101 1111	5F	–		
62	011 1110	3E	>			96	110 0000	60	'		
63	011 1111	3F	?			97	110 0001	61	a		
64	100 0000	40	@			98	110 0010	62	b		
65	100 0001	41	A			99	110 0011	63	c		
66	100 0010	42	B			100	110 0100	64	d		
67	100 0011	43	C			101	110 0101	65	e		
68	100 0100	44	D			102	110 0110	66	f		
69	100 0101	45	E			103	110 0111	67	g		
70	100 0110	46	F			104	110 1000	68	h		
71	100 0111	47	G			105	110 1001	69	i		
72	100 1000	48	H			106	110 1010	6A	j		
73	100 1001	49	I			107	110 1011	6B	k		
74	100 1010	4A	J			108	110 1100	6C	l		
75	100 1011	4B	K			109	110 1101	6D	m		
76	100 1100	4C	L			110	110 1110	6E	n		
77	100 1101	4D	M			111	110 1111	6F	o		
78	100 1110	4E	N			112	111 0000	70	p		
79	100 1111	4F	O			113	111 0001	71	q		
80	101 0000	50	P			114	111 0010	72	r		
81	101 0001	51	Q			115	111 0011	73	s		
82	101 0010	52	R			116	111 0100	74	t		
83	101 0011	53	S			117	111 0101	75	u		
84	101 0100	54	T			118	111 0110	76	v		
85	101 0101	55	U			119	111 0111	77	w		
86	101 0110	56	V			120	111 1000	78	x		
87	101 0111	57	W			121	111 1001	79	y		
88	101 1000	58	X			122	111 1010	7A	z		
89	101 1001	59	Y			123	111 1011	7B	{		
90	101 1010	5A	Z			124	111 1100	7C	¦		
91	101 1011	5B	[			125	111 1101	7D	}		
92	101 1100	5C	\			126	111 1110	7E	~		
93	101 1101	5D	]			127	111 1111	7F	▨	DEL	Delete

EBCDIC Character Conversion Table

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
0	0000 0000	00	NU	NUL		Null
1	0000 0001	01	SH	SOH		Start of Header
2	0000 0000	02	SX	STX		Start of Text
3	0000 0011	03	EX	ETX		End of Text
4	0000 0100	04	PF	PF	(hex)	Punch Off
5	0000 0101	05	HT	HT		Horizontal Tab
6	0000 0110	06	LC	LC	(hex)	Lower Case
7	0000 0111	07	▨	DEL		Delete
8	0000 1000	08	hex)			
9	0000 1001	09	RF	RLF	(hex)	
10	0000 1010	0A	SM	SMM	(hex)	Start Manual Message
11	0000 1011	0B	VT	VT		Vertical Tab
12	0000 1100	0C	FF	FF		Form Feed
13	0000 1101	0D	CR	CR		Carriage Return
14	0000 1110	0E	SO	SO		Shift Out
15	0000 1111	0F	SI	SI		Shift In
16	0001 0000	10	DL	DLE		Data Link Escape
17	0001 0001	11	D1	DC1		Device Control 1 –
18	0001 0010	12	D2	DC2		Device Control 2
19	0001 0011	13	D3	DC3		Device Control 3
20	0001 0100	14	RE	RES	(hex)	Restore
21	0001 0101	15	NL	NL	(hex)	New Line
22	0001 0110	16	BS	BS		Back Space
23	0001 0111	17	IL	IL	(hex)	Idle
24	0001 1000	18	CN	CAN		Cancel
25	0001 1001	19	EM	EM		End of Medium
26	0001 1010	1A	CC	CC	(hex)	Cursor Control
27	0001 1011	1B	C1	CU1	(hex)	

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
28	0001 1100	1C	FS	IFS	FS	Information File Separator
29	0001 1101	1D	GS	IGS	GS	Information Group Separator
30	0001 1110	1E	RS	IRS	RS	Information Record Separator
31	0001 1111	1F	US	IUS	US	Information Unit Separator
32	0010 0000	20	DS	DS	(hex)	Digit Select
33	0010 0001	21	SS	SOS	(hex)	Start of Significance
34	0010 0010	22	FS	FS	(hex)	Field Separator
35	0010 0011	23	(hex)			
36	0010 0100	24	BP	BYP	(hex)	Bypass
37	0010 0101	25	LF	LF		Line Feed
38	0010 0110	26	EB	ETB		End of Transmission Block
39	0010 0111	27	EC	ESC		Escape
40	0010 1000	28	(hex)			
41	0010 1001	29	(hex)			
42	0010 1010	2A	SM	SM	(hex)	Set Mode
43	0010 1011	2B	C2	CU2	(hex)	
44	0010 1100	2C	(hex)			
45	0010 1101	2D	EQ	ENQ		Enquiry
46	0010 1110	2E	AK	ACK		Positive Acknowledge
47	0010 1111	2F	BL	BEL		Bell
48	0011 0000	30	(hex)			
49	0011 0001	31	(hex)			
50	0011 0010	32	SY	SYN		Synchronous Idle
51	0011 0011	33	(hex)			
52	0011 0100	34	PN	PN	(hex)	Punch On
53	0011 0101	35	RS	RS	(hex)	Reader Stop
54	0011 0110	36	UC	UC	(hex)	Upper Case
55	0011 0111	37	ET	EOT		End of Transmission
56	0011 1000	38	(hex)			
57	0011 1001	39	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
58	0011 1010	3A	(hex)			
59	0011 1011	3B	C3	CU3	(hex)	
60	0011 1100	3C	D4	DC4		Device Control 4
61	0011 1101	3D	NK	NAK		Negative Acknowledge
62	0011 1110	3E	(hex)			
63	0011 1111	3F	SB	SUB		Substitute
64	0100 0000	40	(space)			
65	0100 0001	41	(hex)			
66	0100 0010	42	(hex)			
67	0100 0011	43	(hex)			
68	0100 0100	44	(hex)			
69	0100 0101	45	(hex)			
70	0100 0110	46	(hex)			
71	0100 0111	47	(hex)			
72	0100 1000	48	(hex)			
73	0100 1001	49	(hex)			
74	0100 1010	4A	¢	(hex)		
75	0100 1011	4B	.			
76	0100 1100	4C	<			
77	0100 1101	4D	(			
78	0100 1110	4E	+			
79	0100 1111	4F	!	(hex)		
80	0101 0000	50	&			
81	0101 0001	51	(hex)			
82	0101 0010	52	(hex)			
83	0101 0011	53	(hex)			
84	0101 0100	54	(hex)			
85	0101 0101	55	(hex)			
86	0101 0110	56	(hex)			
87	0101 0111	57	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
88	0101 1000	58	(hex)			
89	0101 1001	59	(hex)			
90	0101 1010	5A	!			
91	0101 1011	5B	$			
92	0101 1100	5C	*			
93	0101 1101	5D	)			
94	0101 1110	5E	;			
95	0101 1111	5F	¬			
96	0110 0000	60	-			
97	0110 0001	61	/			
98	0110 0010	62	(hex)			
99	0110 0011	63	(hex)			
100	0110 0100	64	(hex)			
101	0110 0101	65	(hex)			
102	0110 0110	66	(hex)			
103	0110 0111	67	(hex)			
104	0110 1000	68	(hex)			
105	0110 1001	69	(hex)			
106	0110 1010	6A	|			
107	0110 1011	6B	,			
108	0110 1100	6C	%			
109	0110 1101	6D	_			
110	0110 1110	6E	>			
111	0110 1111	6F	?			
112	0111 0000	70	(hex)			
113	0111 0001	71	(hex)			
114	0111 0010	72	(hex)			
115	0111 0011	73	(hex)			
116	0111 0100	74	(hex)			
117	0111 0101	75	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
118	0111 0110	76	(hex)			
119	0111 0111	77	(hex)			
120	0111 1000	78	(hex)			
121	0111 1001	79	`			
122	0111 1010	7A	:			
123	0111 1011	7B	#			
124	0111 1100	7C	@			
125	0111 1101	7D	'			
126	0111 1110	7E	=			
127	0111 1111	7F	"			
128	1000 0000	80	(hex)			
129	1000 0001	81	a			
130	1000 0010	82	b			
131	1000 0011	83	c			
132	1000 0100	84	d			
133	1000 0101	85	e			
134	1000 0110	86	f			
135	1000 0111	87	g			
136	1000 1000	88	h			
137	1000 1001	89	i			
138	1000 1010	8A	(hex)			
139	1000 1011	8B	(hex)			
140	1000 1100	8C	(hex)			
141	1000 1101	8D	(hex)			
142	1000 1110	8E	(hex)			
143	1000 1111	8F	(hex)			
144	1001 0000	90	(hex)			
145	1001 0001	91	j			
146	1001 0010	92	k			
147	1001 0011	93	l			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
148	1001 0100	94	m			
149	1001 0101	95	n			
150	1001 0110	96	o			
151	1001 0111	97	p			
152	1001 1000	98	q			
153	1001 1001	99	r			
154	1001 1010	9A	(hex)			
155	1001 1011	9B	(hex)			
156	1001 1100	9C	(hex)			
157	1001 1101	9D	(hex)			
158	1001 1110	9E	(hex)			
159	1001 1111	9F	(hex)			
160	1010 0000	A0	(hex)			
161	1010 0001	A1	~			
162	1010 0010	A2	s			
163	1010 0011	A3	t			
164	1010 0100	A4	u			
165	1010 0101	A5	v			
166	1010 0110	A6	w			
167	1010 0111	A7	x			
168	1010 1000	A8	y			
169	1010 1001	A9	z			
170	1010 1010	AA	(hex)			
171	1010 1011	AB	(hex)			
172	1010 1100	AC	(hex)			
173	1010 1101	AD	(hex)			
174	1010 1110	AE	(hex)			
175	1010 1111	AF	(hex)			
176	1011 0000	B0	(hex)			
177	1011 0001	B1	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
178	1011 0010	B2	(hex)			
179	1011 0101	B3	(hex)			
180	1011 0100	B4	(hex)			
181	1011 0101	B5	(hex)			
182	1011 0110	B6	(hex)			
183	1011 0111	B7	(hex)			
184	1011 1000	B8	(hex)			
185	1011 1001	B9	(hex)			
186	1011 1010	BA	(hex)			
187	1011 1011	BB	(hex)			
188	1011 1100	BC	(hex)			
189	1011 1101	BD	(hex)			
190	1011 1110	BE	(hex)			
191	1011 1111	BF	(hex)			
192	1100 0000	C0	{			
193	1100 0001	C1	A			
194	1100 0010	C2	B			
195	1100 0011	C3	C			
196	1100 0100	C4	D			
197	1100 0101	C5	E			
198	1100 0110	C6	F			
199	1100 0111	C7	G			
200	1100 1000	C8	H			
201	1100 1001	C9	I			
202	1100 1010	CA	(hex)			
203	1100 1011	CB	(hex)			
204	1100 1100	CC	⎍	(hex)		
205	1100 1101	CD	(hex)			
206	1100 1110	CE	⑂	(hex)		
207	1100 1111	CF	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
208	1101 0000	D0	}			
209	1101 0001	D1	J			
210	1101 0010	D2	K			
211	1101 0011	D3	L			
212	1101 0100	D4	M			
213	1101 0101	D5	N			
214	1101 0100	D6	O			
215	1101 0111	D7	P			
216	1101 1000	D8	Q			
217	1101 1001	D9	R			
218	1101 1010	DA	(hex)			
219	1101 1011	DB	(hex)			
220	1101 1100	DC	(hex)			
221	1101 1101	DD	(hex)			
222	1101 1110	DE	(hex)			
223	1101 1111	DF	(hex)			
224	1110 0000	E0	\			
225	1110 0001	E1	(hex)			
226	1110 0010	E2	S			
227	1110 0011	E3	T			
228	1110 0100	E4	U			
229	1110 0101	E5	V			
230	1110 0110	E6	W			
231	1110 0111	E7	X			
232	1110 1000	E8	Y			
233	1110 1001	E9	Z			
234	1110 1010	EA	(hex)			
235	1110 1011	EB	(hex)			
236	1110 1100	EC	⑂	(hex)		
237	1110 1101	ED	(hex)			

EBCDIC Character Conversion Table (Cont'd)

Dec Value	Binary	Hex	Displayed Character	Standard Mnemonic	Keyboard Entry If Other Than Standard	Description
238	1110 1110	EE	(hex)			
239	1110 1111	EF	(hex)			
240	1111 0000	F0	0			
241	1111 0001	F1	1			
242	1111 0010	F2	2			
243	1111 0011	F3	3			
244	1111 0100	F4	4			
245	1111 0101	F5	5			
246	1111 0110	F6	6			
247	1111 0111	F7	7			
248	1111 1000	F8	8			
249	1111 1001	F9	9			
250	1111 1010	FA	|	(hex)		
251	1111 1011	FB	(hex)			
252	1111 1100	FC	(hex)			
253	1111 1101	FD	(hex)			
254	1111 1110	FE	(hex)			
255	1111 1111	FF	(hex)			

Baudot Character Conversion Table

Dec Value	Binary	Hex	Unshifted Characters (letters)	Shifted Characters (figures)
0	0 0000	00	NU	NU
1	0 0001	01	E	3
2	0 0010	02	LF	LF
3	0 0011	03	A	-
4	0 0100	04	(space)	(space)
5	0 0101	05	S	'
6	0 0110	06	I	8
7	0 0111	07	U	7
8	0 0100	08	CR	CR
9	0 1001	09	D	$
10	0 1010	0A	R	4
11	0 1011	0B	J	BL
12	0 1100	0C	M	,
13	0 1101	0D	F	!
14	0 1110	0E	C	:
15	0 1111	0F	K	(
16	1 0000	10	T	5
17	1 0001	11	Z	"
18	1 0010	12	L	)
19	1 0011	13	W	2
20	1 0100	14	H	#
21	1 0101	15	Y	6
22	1 0110	16	P	0
23	1 0111	17	Q	1
24	1 1000	18	O	9
25	1 1001	19	B	?
26	1 1010	1A	G	&
27	1 1011	1B (figs)	SO (shift out)	SO (shift out)
28	1 1100	1C	M	.
29	1 1101	1D	X	/
30	1 1110	1E	V	;
31	1 1111	1F (LTRS)	SI (shift in)	SI (shift in)

EBCD Character Conversion Table

Dec Value	Binary	Hex	Unshifted Characters (letters)	Shifted Characters (figures)
0	00 0000	00	(space)	(space)
1	00 0001	01	-	—
2	00 0010	02	@	(hex) (¢)
3	00 0011	03	&	+
4	00 0100	04	8	*
5	00 0101	05	q	Q
6	00 0110	06	y	Y
7	00 0111	07	h	H
8	00 1000	08	4	:
9	00 1001	09	m	M
10	00 1010	0A	u	U
11	00 1011	0B	d	D
12	00 1100	0C	(hex) (PN)	(hex) (PN)
13	00 1101	0D	(hex) (RES)	(hex) (RES)
14	00 1110	0E	(hex) (BYP)	(hex) (BYP)
15	00 1111	0F	(hex) (PF)	(hex) (PF)
16	01 0000	10	2	<
17	01 0001	11	k	K
18	01 0010	12	s	S
19	01 0011	13	b	B
20	01 0100	14	O	)
21	01 0101	15	VT	VT
22	01 0110	16	FF	FF
23	01 0111	17	(hex)	(hex)
24	01 1000	18	6	,
25	01 1001	19	o	O
26	01 1010	1A	w	W
27	01 1011	1B	f	F
28	01 1100	1C (UC)	SO (shift out)	SO (shift out)
29	01 1101	1D	BS	BS
30	01 1110	1E	EB	EB
31	01 1111	1F (LC)	SI (shift in)	SI (shift in)
32	10 0000	20	1	=
33	10 0001	21	j	J
34	10 0010	22	/	?
35	10 0011	23	a	A
36	10 0100	24	9	(
37	10 0101	25	r	R
38	10 0110	26	z	Z
39	10 0111	27	i	I
40	10 1000	28	5	%
41	10 1001	29	n	N
42	10 1010	2A	v	V
43	10 1011	2B	e	E
44	10 1100	2C	RS	RS
45	10 1101	2D	CR	CR
46	10 1110	2E	LF	LF
47	10 1111	2F	HT	HT
48	11 0000	30	3	;
49	11 0001	31	l	L
50	11 0010	32	t	T
51	11 0011	33	c	C
52	11 0100	34	#	"
53	11 0101	35	$	!
54	11 0110	36	,	,
55	11 0111	37	.	.
56	11 1000	38	7	>
57	11 1001	39	p	P
58	11 1010	3A	x	X
59	11 1011	3B	g	G
60	11 1100	3C	ET	ET
61	11 1101	3D	(hex) (IL)	(hex) (IL)
62	11 1110	3E	ESC	ESC
63	11 1111	3F	▨ (DEL)	▨ (DEL)

IPARS Character Conversion Table

Dec Value	Binary	Hex	Displayed Character	Keyboard Function	Dec Value	Binary	Hex	Displayed Character	Keyboard Function
0	00 0000	00	(hex)		32	10 0000	20	@	
1	00 0001	01	1		33	10 0001	21	J	
2	00 0010	02	2		34	10 0010	22	K	
3	00 0011	03	3		35	10 0011	23	L	
4	00 0100	04	4		36	10 0100	24	M	
5	00 0101	05	5		37	10 0101	25	N	
6	00 0110	06	6		38	10 0110	26	O	
7	00 0111	07	7		39	10 0111	27	P	
8	00 1000	08	8		40	10 1000	28	Q	
9	00 1001	09	9		41	10 1001	29	R	
10	00 1010	0A	0		42	10 1010	2A	:	UMSG
11	00 1011	0B	•		43	10 1011	2B	<	
12	00 1100	0C	CR	Return	44	10 1100	2C	+	
13	00 1101	0D	EI	End of Medium, Incomplete	45	10 1101	2D	EU	End of Medium, Unsolicited
14	00 1110	0E	=	Sense	46	10 1110	2E	)	
15	00 1111	0F	(hex)	Go-Ahead	47	10 1111	2F	(	
16	01 0000	10	(hex)	Write	48	11 0000	30	$	
17	01 0001	11	/	Erase/Write	49	11 0001	31	A	
18	01 0010	12	S		50	11 0010	32	B	
19	01 0011	13	T		51	11 0011	33	C	
20	01 0100	14	U		52	11 0100	34	D	
21	01 0101	15	V		53	11 0101	35	E	
22	01 0110	16	W		54	11 0110	36	F	
23	01 0111	17	X		55	11 0111	37	G	
24	01 1000	18	Y		56	11 1000	38	H	
25	01 1001	19	Z		57	11 1001	39	I	
26	01 1010	1A	-		58	11 1010	3A	?	
27	01 1011	1B	#		59	11 1011	38	.	
28	01 1100	1C	(space)		60	11 1100	3C	%	
29	01 1101	1D	EC	End of Medium, Complete	61	11 1101	3D	EP	End of Medium, Push Button
30	01 1110	1E	[	Start	62	11 1110	3E	S2	Sync 2
31	01 1111	1F	,		63	11 1111	3F	S1	Sync 1, Reset

Note: The SYN keycap maps to the Sync 2, hex 3E character.

Transcode Character Conversion Table

Dec Value	Binary	Hex	Displayed Characters	Keyboard Mnemonic	Dec Value	Binary	Hex	Displayed Characters	Keyboard Mnemonic
0	00 0000	00	SH	SOH	32	10 0000	20	-	
1	00 0001	01	A		33	10 0001	21	/	
2	00 0010	02	B		34	10 0010	22	S	
3	00 0011	03	C		35	10 0011	23	T	
4	00 0100	04	D		36	10 0100	24	U	
5	00 0101	05	E		37	10 0101	25	V	
6	00 0110	06	F		38	10 0110	26	W	
7	00 0111	07	G		39	10 0111	27	X	
8	00 1000	08	H		40	10 1000	28	Y	
9	00 1001	09	I		41	10 1001	29	Z	
10	00 1010	0A	SX	STX	42	10 1010	2A	EC	ESC
11	00 1011	0B	.		43	10 1011	2B	,	
12	00 1100	0C	<		44	10 1100	2C	%	
13	00 1101	0D	BL	BEL	45	10 1101	2D	EQ	ENQ
14	00 1110	0E	SB	SUB	46	10 1110	2E	EX	ETX
15	00 1111	0F	EB	ETB	47	10 1111	2F	HT	HT
16	01 0000	10	&		48	11 0000	30	0	
17	01 0001	11	J		49	11 0001	31	1	
18	01 0010	12	K		50	11 0010	32	2	
19	01 0011	13	L		51	11 0011	33	3	
20	01 0100	14	M		52	11 0100	34	4	
21	01 0101	15	N		53	11 0101	35	5	
22	01 0110	16	O		54	11 0110	36	6	
23	01 0111	17	P		55	11 0111	37	7	
24	01 1000	18	Q		56	11 1000	38	8	
25	01 1001	19	R		57	11 1001	39	9	
26	01 1010	1A	(space)		58	11 1010	3A	SY	SYN
27	01 1011	1B	$		59	11 1011	3B	#	
28	01 1100	1C	•		60	11 1100	3C	@	
29	01 1101	1D	US	US	61	11 1101	3D	NK	NAK
30	01 1110	1E	ET	EOT	62	11 1110	3E	EM	EM
31	01 1111	1F	DL	DLE	63	11 1111	3F	▨	DEL

APPENDIX E
OSI LEVEL 2 AND 3 TABLES

LEVEL 2: THE DATA LINK INTERFACE

CHARACTER ORIENTED PROTOCOLS: BSC

```
                                                   ITB
| SYN | SYN | SOH | Header | STX | Text | ETX | BCC | BCC |
                                                   ETB
```

BIT ORIENTED PROTOCOLS

```
| Flag | Address | Control | Data | FCS | FCS | Flag |
```

Flags: Flags (7E) act as frame delimiters

Address Field: Command frames contain receiving stations's address. Response frames contain sending station's address.

Control Field: Identifies function and purpose of the frame. Contains commands, responses, and sequence numbers.

Information Field: Any number of bits, typically in multiples of 8 (octets).

FCS: Frame Checking Sequence for Error Detection.

TYPES OF BOP FRAMES (Indicated by the following control fields)

1. Information (I) Frames: For transferring information.

```
| N(R) | P/F | N(S) | 0 |
```

2. Supervisory (S) Frames: To acknowledge I frames, request retransmission of I frames, and to communicate status (busy, ready).

```
Receive Ready (RR)              | N(R) | P/F | 0 0 | 0 1 |

Reject (REJ)                    | N(R) | P/F | 0 1 | 0 1 |

Receive Not Ready (RNR)         | N(R) | P/F | 1 0 | 0 1 |

Selective Reject (SREJ)         | N(R) | P/F | 1 1 | 0 1 |
```

3. Unnumbered (U) Frames: To issue commands and responses.

```
| Type | P/F | Type | 1 1 |
```

UNNUMBERED FORMAT COMMANDS

Control Field Bits	Mnemonic	Name
1 0 0 P 0 0 1 1	SNRM	Set Normal Response Mode
0 0 0 P 0 0 1 1	SARM	Set Asynchronous Response Mode
0 0 1 P 1 1 1 1	SABM	Set Asynchronous Balanced Mode
1 1 0 P 1 1 1 1	SNRME	Set Normal Response Mode Extended
0 1 0 P 1 1 1 1	SARME	Set Asynchronous Response Mode Extended
0 1 1 P 1 1 1 1	SABME	Set Asynchronous Balanced Mode Extended
0 0 0 P 0 1 1 1	SIM	Set Initialization Mode
0 1 0 P 0 0 1 1	DISC	Disconnect
0 0 0 P 0 0 1 1	UI	Unnumbered Information
0 0 1 P 0 0 1 1	UP	Unnumbered Poll
1 0 0 P 1 1 1 1	RSET	Reset
1 0 1 P 1 1 1 1	XID	Exchange Identification

UNNUMBERED FORMAT RESPONSES

0 1 1 F 0 0 1 1	UA	Unnumbered Acknowledgement
0 0 0 F 1 1 1 1	DM	Disconnected Mode
0 0 0 F 0 1 1 1	RIM	Request Initialization Mode
0 0 0 F 0 0 1 1	UI	Unnumbered Information
1 0 0 F 0 1 1 1	FRMR	Frame Reject
1 0 1 F 1 1 1 1	XID	Exchange Identification
0 1 0 F 0 0 1 1	RD	Request Disconnect

SAMPLE MONITOR MENU TRIGGERS

When DTE	Trigger on data from the DTE
When DTE (flag) $^{0}_{\underline{1}}$	Address (second byte)
When DTE (flag) $^{0}_{\underline{1}}$ xxxxxxxx	Control Field, don't cares (3rd byte)
When DTE (flag) $^{0}_{\underline{1}}$ xxxxxx11	U-Frame
When DTE (flag) $^{0}_{\underline{1}}$ 100x0011	Type of U-Frame = SNRM
When DTE (flag) $^{0}_{\underline{1}}$ 10010011	Poll bit set to 1
When DTE (flag) $^{0}_{\underline{3}}$ xxxx0001	S-Frame
When DTE (flag) $^{0}_{\underline{3}}$ 10110001	S-Frame, Type=RR, N(R)=5, P/F=1
When DTE (flag) $^{0}_{\underline{3}}$ 00100010 THIS IS AN I-FRAME	I-Frame, N(R)=1, N(S)=1, P/F=0

LEVEL 3

PACKET CONSTRUCTION (MODULO 8)

```
 _________________
|         |        |
|  GFI    |  LCGN  |
|         |        |    Octet 1
| Q D 0 1 |        |
|_________|________|
|                  |
|       LCN        |    Octet 2
|__________________|
|             |    |
| PACKET TYPE | C  |    Octet 3
|_____________|____|
```

Packet Types

Data	RRRMSSS0
Call Request	00001011
Call Accepted	00001111
Clear Request	00010011
Clear Confirmation	00010111
Interrupt	00100011
Interrupt Confirmation	00100111
Receive Ready	RRR00001
Receive Not Ready	RRR00101
Reject	RRR01001
Reset Request	00011011
Reset Confirmation	00011111
Restart Request	11111011
Restart Confirmation	11111111
Diagnostic	11110001

DATA PACKETS

```
        MODULO 8                                          MODULO 128
 ___________________                                  ___________________
|         |         |                                |         |         |
|  GFI    |  LCGN   |                                |  GFI    |  LCGN   |
|         |         |        Octet 1                 |         |         |
| Q D 0 1 |         |                                | Q D 0 1 |         |
|_________|_________|                                |_________|_________|
|                   |                                |                   |
|        LCN        |        Octet 2                 |        LCN        |
|___________________|                                |___________________|
|      |   |      | |                                |             |     |
| P(R) | M | P(S) |0|        Octet 3                 | P(S)        |  0  |
|______|___|______|_|                                |_____________|_____|
|                   |                                |             |     |
|     USER DATA     |              Octet 4           | P(R)        |  M  |
|___________________|                                |_____________|_____|
                                                     |                   |
                                                     |     USER DATA     |
                                                     |___________________|
```

CALL REQUEST/INCOMING CALL PACKET

```
 _____________________
|  GFI    |   LCGN    |      Octet 1
|_________|___________|
|        LCN          |      Octet 2
|_____________________|
|  0 0 0 0 1 0 1 1    |      Octet 3
|_____________________|
| Calling  | Called   |
|   DTE    |  DTE     |
| Address  | Address  |      Octet 4
| Length   | Length   |
|__________|__________|
| Called DTE Address  |      Octet 5
|_____________________|
| Calling DTE Address|       Octet 6
|_____________________|
|      | Facility     |      Octet 7
| 0 0  | Field Length |
|______|______________|
|   Facility Codes    |      Octet 8
|   and Parameters    |
|_____________________|
|   Call User Data    |
|                     |
```

CALL ACCEPTED/CALL CONNECTED PACKET

```
 _____________________
|  GFI    |   LCGN    |
|_________|___________|
|        LCN          |
|_____________________|
|  0 0 0 0 1 1 1 1    |
|_____________________|
| Calling  | Called   |
|   DTE    |  DTE     |
| Address  | Address  |
| Length   | Length   |
|__________|__________|
| Called DTE Address  |
|_____________________|
| Calling DTE Address|
|_____________________|
|      | Facility     |
| 0 0  | Field Length |
|______|______________|
|                     |
|     Facilities      |
|_____________________|
```

APPENDIX F
EXAMPLES

EXAMPLE 1 MEASURING A SINGLE RTS-CTS DELAY

This test measures the time from when RTS goes on until CTS goes on.

To view the timers and counters, press <Summary> during run-time, or <Timer & Cntr> in the Examine Data Menu after run-time.

Note that timer measurements must be referenced to a preceding trigger for accurate measurements.

```
Block 1:     When Lead RTS goes On
               then goto Block 2

Block 2:     Start Timer 1                        Timer 1 indicates RTS-CTS delay.

             When Lead CTS goes On                Note that Start and Stop statements
               then goto Block 3                  must be preceded by When statements
                                                  for accurate timing.
Block 3:     Stop Timer 1
```

EXAMPLE 2 MONITORING A DCE

In this example, you monitor a DCE by simulating the DTE. The simulate mode allows you to turn on the proper leads and supply a clock to a terminal. (When simulating a DTE, the 4951A automatically supplies the ETC clock). Upon receiving the proper clocks and lead commands, the DCE begins sending data, which the 4951A automatically stores and displays while in the simulate mode.

```
Simulate DTE

Block 1:     Set Lead DTR  On
               and then
             Set Lead RTS  On
```

EXAMPLE 3 MONITORING A DTE

In this example, you monitor a DTE by simulating a DCE. The simulate mode allows you to turn on the proper leads and supply a clock to the DCE. (When simulating a DCE, the 4951A automatically supplies both the TC and RC clocks). Upon receiving the proper clocks and lead commands, the DTE begins sending data, which the 4951A automatically stores and displays while in the simulate mode.

```
Simulate DCE

Block 1:     Set Lead DSR  On
               and then
             Set Lead CD  On
               and then
             Set Lead CTS  On
```

EXAMPLE 4 FOX MESSAGE

This test checks the ability of asynchronous terminals and printers to receive and display data. The "FOX" message is transmitted to the terminal, and then echoed from the terminal is checked for parity errors.

```
Simulate DCE

Block 1:     Send THE QUICK BROWN FOX JUMPS
             OVER A LAZY DOG 0123456789.

Block 2:     When DCE .
               then goto Block 1

             When Error Parity on DTE
               then goto Block 3

Block 3:     Increment Counter 1
               and then
             Goto Block 2
```

EXAMPLE 5 COUNTING PARITY ERRORS

This program counts the number of parity errors on both the DTE and DCE lines and keeps track of the number of minutes of the test.

```
Block 1:     When DTE x                        Timer 5 starts when any character
               or                              is sent on the DTE or DCE line.
             When DCE x                        ("x" = don't care)
               then goto Block 2

Block 2:     Start Timer 5                     Timer 5 counts milliseconds up to
                                               one minute.
Block 3:     When Error Parity on DTE
               then goto Block 4

             When Error Parity on DCE
               then goto Block 5

             When Timer 5 is > 59999
               then goto Block 6

Block 4:     Increment Counter 1               Counter 1 indicates DTE errors.
               and then
             Goto Block 3                      Counter 2 indicates DCE errors.

Block 5:     Increment Counter 2
               and then
             Goto Block 3

Block 6:     Increment Counter 5               Counter 5 keeps track of the
               and then                        number of minutes into the test.
             Reset Timer 5
               and then
             Goto Block 2
```

EXAMPLE 6 MEASURING MORE THAN ONE RTS-CTS DELAY

The preceding test measured only one RTS-CTS delay. This test continues to measure RTS-CTS delays until stopped.

Timer 1 and Timer 2 measure alternate delays. If only one timer were used, you would not have had enough time to see the timer before it was reset.

To view the timers and counters press <Summary> in the Run Menu during run-time. After run-time press <Timer & Counter> in the Examine Data Menu.

```
Block 1:     When Lead RTS goes On
               then goto Block 2

Block 2:     Reset Timer 1
               and then
             Start Timer 1                        Timer 1 measures the first RTS-CTS
                                                  delay.
             When Lead CTS goes On
               then goto Block 3

             When Lead RTS goes Off               The two "When" statements are ORed.
               then goto Block 6                  An alarm "beep" occurs if RTS goes
                                                  off before CTS goes on.
Block 3:     Stop Timer 1

             When Lead RTS goes On                You can now view timer 1 while the
               then goto Block 4                  analyzer finds the next delay.
```

```
Block 4:     Reset Timer 2                        Blocks 3-5 duplicate blocks 1-2.
               and then
             Start Timer 2                        Timer 2 now measures the next RTS-
                                                  CTS delay.  Thus, the user has time
             When Lead CTS goes On                to view timer 1 before it is reset.
               then goto Block 5

             When Lead RTS goes Off               The two "When" statements are ORed.
               then goto Block 6                  If RTS goes off before CTS goes on
                                                  an alarm "beep" occurs in block 6.
Block 5:     Stop Timer 2
               and then
             Goto Block 1

Block 6:     Reset Timer 1
               and then
             Reset Timer 2
               and then
             Beep
               and then
             Goto Block 1
```

EXAMPLE 7 SIMULATING RTS-CTS DELAY

In this test, you substitute the 4951A for the DTE. Thus, you can test the modem in isolation.

Timer 1 measures the time it takes for the modem to respond with CTS on.

```
Simulate DTE

Block 1:      Set Lead RTS On

              When Lead RTS goes On
                then goto Block 2

Block 2:      Reset Timer 1                       Timer 1 is reset because the
                and then                          program later loops back to this
              Start Timer 1                       block.
                and then
              Start Timer 5                       Timer 1 shows CTS response time.
              When Lead CTS goes On
                then goto Block 3

              When Timer 5 > 2000                 Timer 5 causes the instrument to
                then goto Block 4                 beep if CTS does not go on within
                                                  2 seconds.
```

```
Block 3:      Reset Timer 5
                and then
              Stop Timer 1
                and then
              Set Lead RTS Off                    RTS is now turned off and the test
                and then                          begun again after 30 seconds.
              Wait 29999                          (You can change this delay).
                and then
              Goto Block 1

Block 4:      Reset Timer 5                       Block 4 is the "error block".  If
                and then                          CTS does not go on two seconds
              Reset Timer 1                       after RTS goes on, the analyzer
                and then                          beeps and restarts the test.
              Beep
                and then
              Set Lead RTS Off
                and then
              Wait 250
                and then
              Goto Block 1
```

EXAMPLE 8 LOOPBACK

In this test, the local modem is looped back. The 4951A is substituted for the DTE and sends the "Quick Brown Fox" message 100 times. The modem is checked for proper handshaking an echo response.

```
Simulate DTE

Block 1:     Set Lead RTS  On
               and then
             Start Timer 5

             When Lead CTS goes On                    The modem is checked for correct
               then goto Block 2                      handshaking response.

             When Timer 5 is > 2000 msec              Timer 5 indicates whether the modem
               then goto Block 8                      responds within 2 seconds.

Block 2:     Reset Timer 5                            Timer 5 is reset for the next loop.
               and then
             Send THE QUICK BROWN FOX JUMPS           The message is sent to the modem.
             OVER A LAZY DOG 0123456789.
               and then
             Set lead RTS  Off
```

```
Block 3:     When DCE THE QUICK BROWN FOX JUMPS       The modem is checked to see whether
             OVER A LAZY DOG 0123456789.              it echoes back each character.
               then goto Block 5
                                                      Because the two "when" statements
             When Lead CTS goes Off                   are ORed, every charecter must be
               then goto Block 4                      received before CTS goes off.
                                                      Counter 2 indicates the number of
Block 4:     Increment Counter 2                      times this does not happen.

Block 5:     Increment Counter 1                      Counter 1 shows the total number
                                                      of transactions up to 100.
             If Counter 1 is > 99
               then goto Block 7

Block 6:     Goto Block 1                             The test starts over.

Block 7:     Stop Tests

Block 8:     Reset Timer 5
               and then
             Stop Timer 5
               and then
             Beep                                     An alarm "beep" indicates lack of
               and then                               modem response.
             Goto Block 1
```

EXAMPLE 9 END-TO-END: TRANSMIT FIRST (HP 4925A Compatible)

The End-to-End test consists of the two programs described in Examples 9 and 10.

In the End-to-End Test, two 4951A's (or a 4951A and a 4925A) are substituted for the DTE's at both ends of a line. Handshaking and messages are performed and checked 100 times. Except for the fact that one DTE transmits first, and the other DTE receives first, both programs are identical. There are two sections to this program: In blocks 5-6 this DTE is transmitting; in blocks 1-4 this DTE is receiving. Counter 1 indicates how many times the test failed. Counter 2 indicates the total number of transactions.

NOTE: The "receive first" unit must be started first.

The proper setup is necessary for this test. Use the Char Async/Sync Menu with all the default selections (chapter 12) except the following:

Data Code Hex 8 Drop sync 4 chrs after $^{1}_{8}$ $^{1}_{8}$ $^{1}_{8}$ $^{1}_{8}$ $^{1}_{8}$ $^{1}_{8}$ $^{1}_{8}$

Sync on $^{F}_{F}$ $^{9}_{3}$

Simulate DTE

Block 1:	Goto Block 6	The program immediately jumps to the transmit section.
Block 2:	Set Lead DTR On	
	If Lead CD is On then goto Block 3	The <If> and <When> statements are Ored.
	When Lead CD goes On then goto Block 3	

Block 3:	When Lead CD goes Off then goto Block 4	The two <When> statemets are ORed: If CD goes off before the message is received, then the error counter is incremented.
	When DCE $^{7}_{6}$ $^{B}_{3}$ $^{F}_{A}$ $^{4}_{1}$ then goto Block 5	The DCE characters are the same as those sent by the 4925A
Block 4:	Beep and then Increment Counter 1	
Block 5:	Increment Counter 2	Counter 2 tells total transactions.
Block 6:	Wait 100 msec and then	The transmit section of the program begins.
	Set Lead RTS On	
	If Lead CTS is On then goto Block 7	
	When Lead CTS goes On then goto Block 7	
Block 7:	Send $^{F}_{F}$ $^{9}_{3}$ $^{7}_{6}$ $^{B}_{3}$ $^{F}_{A}$ $^{4}_{1}$ $^{1}_{8}$	This is the same message sent by a 4925A
	When DTE $^{1}_{8}$ then goto Block 8	
Block 8:	Set Lead RTS Off	
	If Counter 2 > 99 then goto Block 10	When Counter 2 reaches 100, the test is ended.
Block 9:	Goto Block 2	
Block 10:	Stop Tests	

EXAMPLE 10 END-TO-END: RECEIVE FIRST (HP 4925A Compatible)

This is the part of the END-TO-END TEST for the DTE which receives first. There are two sections to the program: In blocks 1-4 this DTE is transmitting; in blocks 5-6 this DTE is receiving. Counter 2 tells how many times the test failed. Counter 1 keeps track of the total number of transactions.

NOTE: The "receive first" unit must be started first.

Use the Char Async/Sync Menu for the setup. Use all the default selections (see Chapter 12) except the following:

Data Code Hex 8 Drop sync 4 chrs after $^{1}{}_{8}$ $^{1}{}_{8}$ $^{1}{}_{8}$ $^{1}{}_{8}$ $^{1}{}_{8}$ $^{1}{}_{8}$ $^{1}{}_{8}$

Sync on $^{F}{}_{F}$ $^{9}{}_{3}$

Simulate DTE

Block 1: Set Lead DTR On

This is the Receive portion of the End-to-End test.

If Lead CD is On
then goto Block 2

When Lead CD goes On
then goto Block 2

Block 2: When Lead CD goes Off
then goto Block 3

These two <When> statements are Ored. If CD goes off before this DTE has received the message, Counter 1 will indicate another failure.

When Lead $^{7}{}_{6}$ $^{B}{}_{3}$ $^{F}{}_{A}$ $^{4}{}_{1}$
then goto Block 4

Block 3: Beep
and then

Increment Counter 1

Block 4: Increment Counter 2

Counter 2 indicates another transaction.

Block 5: Wait 100 msec
and then

Set Lead RTS On

The transmit section of the program begins.

If Lead CTS is On
then goto Block 6

When Lead CTS goes On
then goto Block 6

Block 6: Send $^{F}{}_{F}$ $^{9}{}_{3}$ $^{7}{}_{6}$ $^{B}{}_{3}$ $^{F}{}_{A}$ $^{4}{}_{1}$ $^{1}{}_{8}$

This is the same message as that sent by a 4925A

When DTE $^{1}{}_{8}$

then goto Block 7

Block 7: Set Lead RTS Off

If Counter 2 > 99
then goto Block 9

When the total transactions = 100 the test is ended.

Block 8: Goto Block 1

Block 9: Stop Tests