ATLANTIC RESEARCH CORPORATION
TELEPRODUCTS DIVISION SPRINGFIELD, VIRGINIA

04/8169

MEMORANDUM

TO: Recipients of Version 5.00 INTERVIEW 7X00 Software

FROM: Lynn Taylor LWT

DATE: Dec 23, 1988

SUBJECT: INTERVIEW 7X00 Series Software Release Information

There has been a lot of new features added and many product improvements in this software release. Functionality has been improved in Run Mode, the function keys have been organized so they behave more consistently with the rest of the unit, and some new Run Mode display functions have been added. The following is a list of features that have been added since the last release:

1. Program Load/Save times have been improved.
2. The Run Mode Function keys have been organized for better operation. The DONE key returns the run time function keys to the top level function key rack.
3. There has been several new Print features added for Freeze Mode. They are as follows:
   1. SNA Trace print
   2. Layer 3 print for Q.931 and SS7
   3. BERT, T1, and T1 BERT Statistics can be printed
   4. Display Window print
   5. User Trace print
   6. Raw Mode print
   7. Run Mode printf statements can be directed to a file instead of the printer
4. Automon: This function will make a determination on the data and setup the Interview 7X00 to monitor the data.
5. Display window: This feature directs all 'display' routines to write to the 'Display Window' screen. When a 'display' routine is used, the output will only write to the Display Window screen. The last 16 lines that have been displayed are retained when changing screens and during Freeze mode. The display routines are as follows:
   displayc(c): Display the character 'c'.
   displays(s): Display the string 's'.
   displayf(const char *format, ...): Display a formatted string to the Display Window screen.
   display_prompt(s): Display the string 's' starting at 0,0.
   unsigned int pos_cursor(r, c): Position cursor at row 'r', column 'c'. Returns the previous cursor position in the form of unsigned int 'rc'.
   restore_cursor(unsigned int rc): Positions the cursor to the previous position.

A new feature has been added to the existing displayf routine. It is a modifier feature which allows the user to set the color, enhance, and display modifier. The function uses an unsigned long variable and is invoked by using '%m'. The following gives the modifier bit values:

```
31:     B/W Strike Thru
30:     B/W Blink
29:     B/W Low Intensity
28:     Color, B/W Hex
27:     B/W Reverse video
26:     Color, B/W Underline: Same color as Fore ground
25:     Color Display, B/W No Display
24:     Color, B/W Overline: Same color as Fore ground
23:     Color Strike Thru: Same color as Fore ground
22:     Color Blink
19-21:  Fore ground
        000: Black
        001: Dark Blue
        010: Green
        011: Light Blue
        100: Red
        101: Purple
        110: Yellow
        111: White
16-18:  Back ground
        000: Black
        001: Dark Blue
        010: Green
        011: Light Blue
        100: Red
        101: Purple
        110: Brown
        111: White
8-15:   Not used for '%m', character to be displayed when
        used in table form.
7:      Display Modifier Bit
        0: This display_character may be forced into the
           hex attribute format
        1: This display_character may not be forced into
           the hex attribute format
3-6:    Reserved, always set to zero
0-2:    Font
        000: ASCII_FONT       Use ASCII character set
        001: SPECIAL_FONT     Use special character set
        010: PRIMARY_FONT     Use Primary System codes
        011: ALTERNATE_FONT   Use Alternate System codes
        111: HEX_FONT         Hexadecimal character set
```

6. User variables

   extern unsigned short current_line: Contains the current line number

   extern unsigned short current_col: Contains the current column number

extern unsigned short window_color: Contains the color & enhancements
extern unsigned char window_modifier: Contains the window modifier

7. DDCMP Layer 1 Monitor: This feature calculates the BCC and passes date to Layer 2 in an IL_MESSAGE buffer.
   Event variables
   ```
   extern fast_event evar_gd_bcc2_rd;
   extern fast_event evar_gd_bcc2_td;
   extern fast_event evar_bd_bcc2_rd;
   extern fast_event evar_bd_bcc2_td;
   ```
8. X.21 Monitor and emulation
9. X.21 "C" functions
   1. User routines
      1. enter_call_phase(): Sets the unit for ASCII, 7, ODD, SYNC.
      2. enter_data_phase(): Sets the unit up to the Line Setup Parameters.
      3. bool add_event_to_buff(unsigned short event): Returns TRUE if the event(s) were accepted. Returns FALSE if no event(s) were accepted.
      4. bool add_array_to_buff(&events[0], count):
         unsigned short events[n];
         unsigned short count; /*Set to n*/
         Returns TRUE if the event(s) were accepted. Returns FALSE if no event(s) were accepted.
   2. User variables
      1. extern unsigned char td_modifier
      2. extern unsigned char rd_modifier
10. User Trace: There are 8 new display modes available during run mode. They include a Program Trace and 7 Layer Traces. These trace modes use a set of routines callable by the user's program. The routines are:

```
extern struct trace_buf l3_trbuf;
extern int tracef
  (struct trace_buf *, const char *, ...);
extern void traces (struct trace_buf *, const char *);
extern void tracec (struct trace_buf *, char);
extern void stracef
  (unsigned long *, const char *, ...);
```

The Display Setup portion of the Line Setup screen has some additional selections for these display modes. If Layer Trace is selected as the default display, the user enters the layer number(1-7 on Interview 7X00) of the trace he desires. If Program Trace is selected, three additional fields appear. Two fields allow the user to selectively enable program trace for only a particular layer and/or a particular test. If the fields are left blank, they indicate all layers and tests. The third field is an option field which allows the user to turn on a state trace for the selected test. If this is turned on, the program will put a message into the Program Trace buffer each time the selected test changes states. Note: the Program Trace layer & test fields on the display setup screen affect ONLY the

Protocol Spreadsheet trace functions. They do not prevent other layers and tests from accessing the trace buffer using AR"C" code.

An example of how to use the standard trace routines are shown below. Note that the examples use the layer 3 trace buffer. The program trace buffer is named 'prog_trbuf', and the other layers trace buffers have similar names to layer 3.

```
#include <trace_buf.h>
extern struct trace_buf l3_trbuf;
extern int tracef (struct trace_buf *, const char *,...);
extern void traces (struct trace_buf *, const char *);
extern void tracec (struct trace_buf *, char);
extern void stracef (unsigned long *, const char *,...);
static long temporary_buffer [100];
tracef (&l3_trbuf, "Tracef is very similar to %s.\n",
   "fprintf");
traces (&l3_trbuf, "Traces is very similar to fputs.");
tracec (&l3_trbuf, '\n');
stracef (temporary_buffer, "%mThis blinks.",0x40500000L);
tracef (&l3_trbuf, "%mThis is steady. %b  This is steady.",
   0x400000L, temporary_buffer);
```

11. Interview 20 Plus Data transfer: Data can be transferred from an Interview 20 Plus to an Interview 7X00. Data cannot be transferred from an Interview 7X00 to an Interview 20 Plus unit.
12. TIM Packages: Routines that are only valid when a TIM is installed have been organized to allow future expansion.
13. User Codes: The method for displaying the available Code Sets has changed. The Codes available for the unit are now obtained from the '/sys/codes' directory. Each file in the directory is read and displayed in the Function Key part of the CODES line in the Line Setup screen. Due to the change in operation, programs that have been saved with XS-3 will not load correctly. All programs that were saved with XS 3 will have to be loaded, the CODES entry on the Line Setup screen will have to be changed to XS_3, and the program will have to be saved again. This will insure proper load operations when using this code set. Two new files have been added to display the Katakana Code Set. They are:
    1. JIS 7
    2. JIS 8
14. Program Chaining: This will allow the user to link programs together while the unit is in Run Mode.

1. load_program(const char *filename): The filename must be an absolute pathname, prefixed with the device name. The valid device names are "FD1", "FD2", and "HRD". A typical example would be "HRD/usr/tempfile".15.

Misc items in Protocol Spreadsheet

1. The sequence /*CTRL+L*/ will send a Form Feed to the printer when printing Protocol Spreadsheet.

2. extern fast_event keyboard_new_any_key: Reads all keys and stores the value in 'unsigned short keyboard_any_key'. ASCII keys still work with 'keyboard_new_key'.
3. There have been tick values added for each layer. The time is the value that is passed up by the il_message buffer.

```
extern unsigned long l2_tick_count
extern unsigned long l3_tick_count
extern unsigned long l4_tick_count
extern unsigned long l5_tick_count
extern unsigned long l6_tick_count
extern unsigned long l7_tick_count
```

4. The PAD character is saved and loaded in raw form.
5. The FEB stops receiving data when an overflow occurs and freezes the real time clock in the display.
6. extern fast_event display_screen_changed: This event comes true when a Run Mode screen changes. Two user variables, 'unsigned short crnt_display_screen' and 'unsigned short prev_display_screen', are updated when the screen changes. A screen change occurs when the display is changed or when the Freeze key is pressed. The following values are used to indicate the current screen:

NO_DISPLAY	0x0000
SINGLE	0x0001
DUAL	0x0002
SINGLE+LEADS	0x0003
DUAL+LEADS	0x0004
STATS_TABULAR	0x0011
STATS_GRAPHIC	0x0012
DISP_WINDOW	0x0021
PROG_TRACE	0x0031
PROTOCOL_TRACE	0x0040
L2_PROTOCOL_TRACE	0x0042
L3_PROTOCOL_TRACE	0x0043
USER_TRACE	0x0050
L1 USER	0x0051
L2 USER	0x0052
L3 USER	0x0053
L4 USER	0x0054
L5 USER	0x0055
L6 USER	0x0056
L7 USER	0x0057
TIM_PKG_STD_STATS	0x0061
TIM_PKG_AUX1	0x0062
TIM_PKG_STD_BERT	0x0068
TIM_PKG_BROADBAND_BERT	0x0069
FREEZE MODE	0x0100

7. The users program can now access the Hardware and the Software configuration of the unit from the following structures:

```
struct mpm_info {
  char              rev_num;
  char              present;
  unsigned long     lo_addr;
  unsigned long     hi_addr;
};

struct unit_config {
  char              floppy_exists_mask;
  char              hard_disk;
  char              test_board;
  char              mux;
  char              modem;
  char              num_mpms;
  struct mpm_info   mpm[4];
  char              cpm_rev;
  char              gbm_rev;
  char              pcm_rev;
  char              modem_rev;
  char              mux_rev;
  char              tim_type;
  unsigned long     last_ram_cpm;
  unsigned long     self_test_errors;
  unsigned long     version;
  unsigned long     model_number;
  char              feb_type;
  char              spare1;
  unsigned short    spare1_5;
  unsigned long     spare2;
  unsigned long     spare3;
  unsigned long     spare4;
  unsigned long     spare5;
  unsigned long     spare6;
  unsigned long     spare7;
  unsigned long     spare8;
  unsigned long     spare9;
  unsigned long     sparea;
  unsigned long     spareb;
  unsigned long     sparec;
  unsigned long     spared;
  unsigned long     sw_version;
  unsigned long     fw_version;
};

struct unit_setup {
  unsigned long     speed_dce;
  unsigned long     speed_dte;
  unsigned long     usec_per_tick;
  char              bit_order_polarity;
  char              bits_per_byte;
```

```
    char            clocking_type;
    char            data_source;
    char            format;
    char            mode;
    char            parity;
    char            code_name[13];
};
```

The following values correspond to the fields in the structures:

UNIT_CNFG_FLOPPY1_MASK	0x01
UNIT_CNFG_FLOPPY2_MASK	0x02
ORIGINAL_FEB	0x00
NEWER_FASTER_FEB	0x01
FAST_RECORD_FIFO_FEB	0x02
TIM_TYPE_RS232	0xF0
TIM_TYPE_X21	0xF1
TIM_TYPE_V35	0xF2
TIM_TYPE_RS449	0xF3
TIM_TYPE_EXPAND_ADAPT	0xF4
TIM_TYPE_RC8245	0xF5
TIM_TYPE_UNDEF1	0xF6
TIM_TYPE_UNDEF2	0xF7
TIM_TYPE_UNDEF3	0xF8
TIM_TYPE_UNDEF4	0xF9
TIM_TYPE_UNDEF5	0xFA
TIM_TYPE_UNDEF6	0xFB
TIM_TYPE_ISDN	0xFC
TIM_TYPE_G703	0xFD
TIM_TYPE_T1	0xFE
TIM_TYPE_NONE	0xFF
DATA_SOURCE_DISK	0
DATA_SOURCE_LINE	1
MODE_AUTOMON	0
MODE_MON	1
MODE_EMDCE	2
MODE_EMDTE	3
PARITY_NONE	0
PARITY_EVEN	1
PARITY_ODD	2
PARITY_MARK	3
PARITY_SPACE	4
CLOCKING_TYPE_INTERNAL	0
CLOCKING_TYPE_EXTERNAL	1
CLOCKING_TYPE_SPLIT	2
FORMAT_SYNC	0
FORMAT_BOP	1

FORMAT_ASYNC	2
FORMAT_ISOC	3
BIT_ORDER_POLARITY_NORMAL	0
BIT_ORDER_POLARITY_NORMAL_INVERSE	1
BIT_ORDER_POLARITY_REVERSE_NORMAL	2
BIT_ORDER_POLARITY_REVERSE_INVERSE	3

ERRATA
TO
INTERVIEW 7000 SERIES TECHNICAL MANUAL

ATLC-197-951-100 and ATLC-107-951-108, Issue 5, July 1990

As Issue 5 of the *INTERVIEW® 7000 Series Technical Manual* went to press, additional features were added to the software. Explanations for these changes follow.

- A new C routine has been added to significantly increase the speed of byte-by-byte copy:

bcopy

Synopsis

```
extern void bcopy(source_ptr, destination_ptr,number);
unsigned char * source_ptr;
unsigned char * destination_ptr;
unsigned short number;
```

Description

This routine executes a very fast byte-byte-byte copy of a specified number of bytes from a source variable to a destination variable. You may want to use it to perform fast, real-time copying of received data.

> NOTE: If you have your own version of *bcopy*, we still recommend that you use the faster AR version. Otherwise, your version may be interpreted as an attempt to redefine a reserved name.

Inputs

The first parameter is a pointer the source variable.

The second parameter is a pointer to the destination variable.

The third parameter specifies the number of bytes to be copied from the source to the destination.

Example

As data is received from the line, it temporarily stored in IL buffers as it is passed up the layers. The IL buffers are then reused. In the following example, received data at Layer 2 is quickly copied from the IL buffer so that it is not lost when the buffer is reallocated.

```
{
 struct il_buffer
  {
   unsigned short lock;
   unsigned short maintain_bits;
   unsigned short buffer_size;
```

```
    unsigned short transmit_tag;
    unsigned short receive_tag;
    unsigned long char_buff_frame_start;
    unsigned long char_buff_frame_end;
    unsigned short tick_count_high;
    unsigned short tick_count_mid;
    unsigned short tick_count_low;
    unsigned short available_space_offset;
    unsigned short bytes_remaining;
    unsigned long bcc_indicator;
    unsigned char data [4064];
  };
 struct il_buffer * il_buffer_ptr;
 extern event m_lo_ph_prmtv;
 extern volatile unsigned short m_lo_ph_il_buff;
 extern volatile unsigned short m_lo_ph_sdu_size;
 char data_string[256];
}
LAYER: 2
     STATE:
        CONDITIONS:
        {
         m_lo_ph_prmtv
        }
        ACTIONS:
        {
         il_buffer_ptr = ((void *)((long)m_lo_ph_il_buff << 16));
         bcopy(&il_buffer_ptr->data[0], &data_string[0],m_lo_ph_sdu_size);
        }
```

- Two new C event variables check for received flags. There is one variable for each side of the line.

Type	Variable	Meaning
extern fast_event	fevar_flag_td	True when a BOP flag is detected on the TD side of the line. Line Setup configured for emulate or monitor mode.
extern fast_event	fevar_flag_rd	True when a BOP flag is detected on the RD side of the line. Line Setup configured for emulate or monitor mode.

- A new array of C structures stores information relating to multiple transmit windows for up to sixteen addresses in SNA/SDLC multi-drop operation.

Type	Variable	Value (hex/decimal)	Meaning
Structure Name:	**l2_path_table[]**		Array of structures for monitoring the transmit windows of multiple addresses in SDLC multi-drop operation. Declared as type *extern volatile struct*. Declared automatically for softkey-entered window conditions. Each array element (structure) is associated with a particular address. In point-to-point operation, only the first position in the array is used. Reference a structure variable as follows: l2_path_table*[0]*.current_window_edge. Line Setup configured for emulate mode only.
unsigned char	flags		reserved for future use
unsigned char	number	*0-f/0-15*	Associated drop number for address listed in the ADDR table on the SDLC Frame Level Setup screen. In point-to-point operation, this drop number defaults to zero.
unsigned int	c_addr	*0-fff0-255*	One of the (sixteen possible) controller addresses listed in the ADDR table on the SDLC Frame Level Setup screen.
unsigned int	window_start		reserved
unsigned int	window_end		reserved
unsigned int	lower_window_edge		see current_window_edge
unsigned int	upper_window_edge		see current_window_edge
unsigned int	current_window_edge		When equal to upper edge, window is full; when equal to lower edge, window is empty; when not equal to upper edge, window is not full; and when not equal to lower edge, window is not empty.
unsigned int	resend_edge		When resend edge is not equal to lower window edge, there is more to resend; when resend edge is equal to lower window edge, there is no more to resend.
unsigned char	last_rcvd_nr	*0-7 (MOD 8)*	N(R) of the last info or supervisory frame received from the specified address (see c_addr above).
unsigned char	last_rcvd_ns	*0-7 (MOD 8)*	N(S) of the last info frame received from the specified address (see c_addr above).
unsigned char	last_sent_nr	*0-7 (MOD 8)*	N(R) of the last info or supervisory frame sent to the specified address (see c_addr above).
unsigned char	last_sent_ns	*0-7 (MOD 8)*	N(S) of the last info frame sent to the specified address (see c_addr above).
unsigned char	vr		reserved for future use
unsigned char	vs		reserved for future use
unsigned char	unack_frame_start		reserved for future use
unsigned char	spare		reserved for future use

- A point about the File Maintenance Save command for Linkable Program (LPGM) files needs clarification. When you Save a program as an LPGM file, your spreadsheet program *excluding* C *#pragmas* is saved as compiled linkable-object code. To retain *#pragmas* in your program, reference them as the hook text in a *#pragma hook 0* before you save the program. Although the *#pragma hook 0* is not saved, the hook text is.

  There are two exceptions to this rule: *#pragma object* and *#pragma hook 0.* Enter these directives on the spreadsheet following the format explained in Section 59, C Basics. They will be saved intact in LPGM files.

  To include other *#pragmas*, *#pragma il_buffers 128* for example, in an LPGM file, enter them in a *#pragma hook 0* as follows:

  ```
  #pragma hook 0 "#pragma il_buffers 128"
  ```

- Modifications to some Easy View menus necessitated the replacement of Sections 4 and 19 of the manual. Section 19 was also enhanced with additional examples.

INTERVIEW® 7000 Series
TECHNICAL MANUAL

Issue 5, July 1990

DIVISION OF TELENEX
7401 Boston Boulevard
Springfield, Virginia 22153

Notice

This is Issue 5 of the INTERVIEW® 7000 Series Technical Manual, July 1990.

Issue 5 is written to specifications for the INTERVIEW 7000 Series—the 7000, 7200 *TURBO*, 7500, and 7700 *TURBO*—software revision 8.00. In most instances, further software revisions will be accompanied by an addendum to this issue. In cases where new software does not affect the accuracy of the manual, however, an addendum may not be produced.

Software revision 8.00 adds support for a new Easy View™ feature for the 7200 *TURBO* and the 7700 *TURBO*. Use the menus in Easy View to quickly load and execute programs or to access help information or tutorials about INTERVIEW screens, menu selections, and protocols. Section 4 provides an overview of the Easy View system. It describes the appearance of screens and the keys used to interact with those screens. Section 18 presents the Easy View Setup screen, used to configure the parameters that control Easy View operation. Section 19 describes the Easy View Maintenance program. Use this program to install application programs into the Easy View system or to create multiple versions of programs.

In addition to support for Easy View, software revision 8.00 adds the following features to the INTERVIEW:

- *TURBO* units can now fully utilize 2 Mbytes of RAM on each *TURBO* MPM. Previously, only 1 Mbyte was available.
- Access to Line Setup and Miscellaneous Utilities menus has been improved. Now you can enter these menus with fewer keystrokes, and once in the menu, you are ready to make a selection for the field highlighted by the blinking cursor.
- Cursor timing is a new feature of Freeze-mode data display that allows you to mark an event in the buffer and then view the elapsed time between that marked event and the cursor. The time is displayed at the top of the screen. (See Section 6.)
- A new file type, Linkable Program, has been added to the File Maintenance system. It contains all of the setup menus of a program (PRGM) file except for Triggers and source-code Spreadsheet. In addition to the menus, a linkable-program file contains the linkable-object code compilation of the Protocol Spreadsheet. (See Section 14.)
- Now you can save the configuration of the Printer Setup screen to a file that will be searched for and accessed during boot-up. The parameters saved in this file are automatically loaded in to the Printer Setup screen. (See Section 15.) A C routine has also been added for resetting the printer-output page to one. (See Section 67.)
- The data-plus-leads display can be printed from Freeze mode. (See Section 15.)
- The number and size of interlayer message buffers has been made flexible. Use either the IL_BUFFERS programming block on the Protocol Spreadsheet (see Section 27) or two new C *#pragmas* (see Section 66) to configure the size and number of IL buffers.

- A CALL_SETUP_SEND_IDLE action is new to the X.21 layer personality package. It guarantees that a specified change in the idle character actually occurs *during* a string transmission. (See Section 35.) The equivalent C routine has also been added. (See Section 73.)
- SNA/SDLC multi-drop capability has been added to the SDLC and SNA/SDLC layer personality packages. Choose from point-to-point or multi-drop operation on the Frame Level Setup screen. If you select multi-drop, specify up to sixteen controller addresses. A loopback address selection now appears for the RESEND, RESET_NR, and RESET_NS actions. (See Section 38 and 39.) Three new C routines that perform these same functions have also been provided. See Section 76.
- The signalling channel number is now selectable for CCS mode in G.703. (See Section 53.)
- Two C routines, *disable_dce* and *disable_dte*, may be used to completely disable the monitoring of the DCE or DTE side of the line. This reduction (by half) in the receive load enables the INTERVIEW to achieve better speeds for user-implemented BERT. (See Section 62.)
- Use new C routines for to assign labels to the function keys in the Display Window, User Trace, and Protocol Trace screens. (See Section 64.)
- The meaning of the value in the "wait" parameter of several remote port I/O routines has been updated to incorporate a timeout value. (See Section 70.)
- Another new C routine, *surrender_cpu*, allows one task to surrender the CPU so that another task can run. (See Section 72.)
- Software support has been added for a for future hardware enhancement—the TIM expansion shelf. At this time, the only change is that the Layer Setup screen looks slightly different. Its operation, however, remains the same.
- During boot-up, the system checks the hardware configuration of *TURBO* units for compatibility with Easy View. If a 2-Mbyte CPM is found, a warning message is displayed. Although the unit may need an upgrade to a 4-Mbyte CPM, its general operation is unaffected.

An addendum to Issue 4 of the manual, dated May 1990, has been incorporated into Issue 5. In addition to the sections already described above, Issue 5 also incorporates information in the following areas:

- The discussion in Section 2 about accessing menus has been updated. The RETURN key now causes the system to behave as if the highlighted function key had been pressed. This is similar to the way selections are made from the Easy View menus.
- Section 3 incorporates the new Freeze-mode use of the MARK key for cursor timing. It also introduces the EZ VU key used to toggle between Easy View and Program mode.
- Section 15 explains how to save the Printer Setup and includes the translation of character representations from data-plus-leads displays to printed output (in Table 15-2).
- Appendix A2 provides status and error messages you may see during Easy View operation.

The AR Division of Telenex Corporation reserves the right to improve this manual or the equipment it describes without prior notice.

Any duplication of the material in this manual in any form without written permission from Telenex Corporation is strictly forbidden.

For technical information, programming assistance, error decoding, and repairs, contact Customer Service. Customers within the Washington D.C. Greater Metropolitan Area call 644-9190. Those outside of the local area call 1-800-368-3261 (in Virginia, call 1-703-644-9190).

Address questions and comments about this manual and other AR Division technical publications to Technical Writing. Customers within the Washington D.C. Greater Metropolitan Area call 644-9078. Those outside of the local area call 1-800-368-3261, extension 9078 (in Virginia, call 1-703-644-9078).

Contents

Part I: Basic Operation

I Overview

1 Hardware 1-1

1.1 Physical Dimensions 1-3
1.2 Keyboard 1-3
1.3 Front Panel 1-3
1.4 Disk Drives 1-5
1.5 Back Panel 1-7
1.6 Storage Capacity 1-12
1.7 Clock 1-13
1.8 Operating Environment 1-14
1.9 Operating Positions 1-14
1.10 Power Up 1-14

2 General Operation 2-1

2.1 Power Up 2-1
2.2 Initializing System Software (Booting Up) 2-3
2.3 Installing New System Software on Hard Disk 2-7
2.4 Backing Up the Hard Disk 2-8
2.5 The Menus 2-9
2.6 Running a Test Program 2-15
2.7 Data Flow 2-18
2.8 How to Correct Common Problems 2-20
2.9 Optimizing High-Speed Operation 2-23

3 Keyboard 3-1

3.1 Hard Keys and Softkeys 3-3
3.2 Programming Keys 3-3
3.3 Real-Time Keys 3-10
3.4 Freeze-Mode Keys 3-13

4 Easy View 4-1

4.1 Booting Up 4-3
4.2 Entering and Exiting Easy View 4-3
4.3 Easy View Menus 4-4
4.4 Using Easy View 4-7

II Setup and Display

5 Line Setup 5-1

5.1 Mode 5-3
5.2 Source 5-4
5.3 Code 5-5
5.4 Bits 5-6
5.5 Parity 5-6
5.6 Format 5-6
5.7 Clock Source 5-10
5.8 Bit-Order/Polarity 5-11
5.9 NRZI 5-12
5.10 MIL 5-12
5.11 Sample Line Setups 5-12

6 Run-Mode Display 6-1

6.1 Auto Configure 6-3
6.2 Entering Run Mode 6-5
6.3 Selecting Character-Data Display 6-7
6.4 Special Features of Data Display 6-13
6.5 Protocol Trace 6-17
6.6 Program Trace 6-19
6.7 Statistics Display 6-22
6.8 Display Window 6-23
6.9 User Trace 6-25
6.10 Display Correlation in Freeze Mode 6-27
6.11 No Display 6-28

7 Record Setup 7-1

7.1 Format of Recorded Data 7-3
7.2 Recording Medium 7-3
7.3 The Screen Buffer 7-4
7.4 The Record Setup Screen 7-5
7.5 Trigger Control of Capture 7-10
7.6 Spreadsheet Control of Recording 7-10
7.7 Manual Control of Recording 7-11

8 Layer Setup 8-1

8.1 Personality Packages 8-3
8.2 Selecting and Loading Protocols 8-3
8.3 The Protocol Configuration Screen 8-4
8.4 Saving the Layer Setup Screen 8-5

9 FEB Setup 9-1
9.1 Buffering Idle, Control Leads, and Ticks 9-3

III BCC and BERT

10 Block Checking 10-1
10.1 BCC Symbols 10-3
10.2 BCC Conditions 10-5
10.3 Transmitted BCC 10-5
10.4 Standard BCC Parameters 10-6
10.5 BCC Setup Menu Fields 10-10
10.6 BISYNC vs. Selectable CRC Mode 10-13

11 Bit Error Rate Testing 11-1
11.1 Pseudorandom Bit Patterns 11-3
11.2 Test Configurations 11-4
11.3 BERT Operation: Full Duplex 11-4
11.4 BERT Operation: Half Duplex 11-6
11.5 BERT Setup Screen 11-7
11.6 Transmission Format: Line Setup Menu 11-15
11.7 Run Mode: Keyboard Control 11-17
11.8 Run Mode: Status Line 11-18
11.9 Run Mode: Statistical Display 11-19
11.10 Loopback at the Transmitting INTERVIEW 11-21
11.11 T1 BERT 11-23
11.12 G.703 BERT 11-35

IV Standard Interface

12 Standard Interface: RS-232 12-1
12.1 Connectors 12-3
12.2 Green-Red LEDs 12-4
12.3 Breakout Box 12-5
12.4 Screen Display of Lead Status 12-9
12.5 Program Control of Lead Activity 12-9
12.6 Interface Control Menu Screen 12-10

V File System and Miscellaneous Utilities

13 Disk Maintenance 13-1
13.1 The Disks 13-3
13.2 Allocating Disk Space 13-3
13.3 Data Acquisition Tracks vs. the Filing System 13-3
13.4 The Disk Maintenance Screen 13-4

14 File Management 14-1
14.1 Loading and Saving Files 14-3
14.2 Orienting Yourself in the Filing System 14-4
14.3 The File Maintenance Screen 14-9

15 Printer Control 15-1
15.1 The Connector 15-3
15.2 Configuring the Printer Setup Screen 15-3
15.3 Saving the Printer Setup 15-7
15.4 How to Print Static Displays 15-8
15.5 How to Print Data 15-10
15.6 Spreadsheet Control of Printing 15-17
15.7 Printing Disk Files 15-17

16 The Time-of-Day Clock 16-1

17 Color Display 17-1
17.1 Connectors for External Monitors 17-3
17.2 Color Control from the Miscellaneous Utilities Screen 17-3
17.3 Black and White Data Enhancements 17-6

18 Easy View Setup 18-1
18.1 Enable Easy View 18-3
18.2 Enter Easy View After Power-Up 18-4
18.3 Keep Easy View Menu Information in Memory 18-4
18.4 Display Program Warning Messages 18-4

19 Easy View Maintenance 19-1
19.1 The Easy View Menu File 19-1
19.2 Accessing the Maintenance Program 19-2
19.3 Automatic Installation 19-3
19.4 Customizing Programs 19-3
19.5 Script File Format 19-11
19.6 Creating a Text (Help) File 19-18
19.7 Easy View Directory Structure 19-19

VI Statistics

20 Tabular Statistics 20-1
20.1 Counters and Timers 20-3
20.2 Preparing the Tabular Statistics Screen 20-4

20.3 Sampling Current Values 20-6
20.4 Accumulators 20-7
20.5 Keeping a Statistical Log 20-9
20.6 The Sampling Action as Divisor 20-11

21 Graphic Statistics 21-1

21.1 Enabling the Graphic Display 21-3
21.2 Cursor Movement on Graphical Statistics Menu 21-4
21.3 Menu Fields 21-4

VII Programming Concepts: Tiers and Layers

22 Three-Tiered Programming 22-1

22.1 Trigger Setup Screens 22-2
22.2 The Protocol Spreadsheet 22-3
22.3 C Programming Language 22-4
22.4 Integrating Programming Methods 22-5

23 The Layered Program Model 23-1

23.1 States 23-3
23.2 Tests 23-3
23.3 Layers and the OSI Model 23-5
23.4 Personality Packages 23-7
23.5 Primitives 23-8
23.6 Constants 23-8

VIII Trigger Menus

24 Trigger Conditions 24-1

24.1 Active Triggers 24-4
24.2 Combining Conditions on the Same Trigger Setup Screen 24-4
24.3 Receiver 24-5
24.4 EIA 24-9
24.5 Timeout 24-10
24.6 Transmission Complete 24-10
24.7 Internal Flag Bits 24-10
24.8 Buffer Full 24-11
24.9 Counter 24-11
24.10 Keyboard 24-12

25 Trigger Actions 25-1

25.1 Displaying a Prompt 25-3
25.2 Transmitting 25-4

25.3 Internal Flags 25-6
25.4 Enhancing the Display 25-7
25.5 Controlling Timeouts 25-8
25.6 Counters 25-9
25.7 Timers 25-10
25.8 Alarm 25-11
25.9 Capture of Data in the Screen Buffer 25-11

26 The Trigger Summary Screen 26-1

IX Protocol Spreadsheet

27 Programming Blocks 27-1

27.1 Before You Begin a Spreadsheet Program 27-3
27.2 Creating a Spreadsheet Program 27-3
27.3 Program Structure 27-6
27.4 Compiled Spreadsheet 27-10
27.5 Configuring the Size/Number of IL Buffers 27-13
27.6 Comments in a Spreadsheet Program 27-14

28 Constants 28-1

28.1 Definition of Constants 28-3
28.2 Constant Names 28-4
28.3 Scope 28-4
28.4 Referencing Constants 28-5
28.5 Nested Constants 28-6
28.6 Precedence 28-6
28.7 Expansion 28-7

29 Editor 29-1

29.1 Basic Editing Functions 29-3
29.2 Editing Function Keys 29-5

30 Layer-Independent Conditions and Actions 30-1

30.1 Naming Requirements 30-1
30.2 Rules for Combining Conditions 30-2
30.3 Layer-Independent Conditions 30-3
30.4 Layer-Independent Actions 30-8

31 Layer 1 Conditions and Actions 31-1

31.1 Layer 1 Conditions 31-1
31.2 Layer 1 Actions 31-6

32 Strings 32-1
32.1 Strings To Be Matched Against Line Data 32-3
32.2 Strings To Be Transmitted 32-3

X OSI Primitives

33 OSI Primitives on the Protocol Spreadsheet 33-1
33.1 Softkey Selections 33-4
33.2 Sample Primitives: CONNECT INDs and CONNECT REQs 33-7
33.3 Sample Primitives: DATA INDs and DATA REQs 33-9

34 Automatic OSI Primitives 34-1

XI Protocol Packages

35 X.21 Layer 1 35-1
35.1 X.21 bis 35-4
35.2 Transmitter/Receiver Phases 35-4
35.3 Sending From Layer 2 35-5
35.4 X.21 Conditions 35-5
35.5 X.21 Actions 35-7

36 X.25 Layer 2 36-1
36.1 Frame-Level Setup 36-3
36.2 Protocol Trace 36-5
36.3 Monitor Conditions 36-11
36.4 Emulate-Mode Conditions 36-15
36.5 Emulate Actions 36-21
36.6 Display Actions 36-32
36.7 Automatic Primitives 36-33
36.8 Programming Example: Converting Protocol Bytes to Hexadecimal .. 36-35
36.9 Programming Example: A Simple "Automatic" Layer 2 X.25 Test .. 36-36

37 X.25 Layer 3 37-1
37.1 Packet-Level Setup 37-3
37.2 Protocol Trace 37-8
37.3 Monitor Conditions 37-15
37.4 Emulate-Mode Conditions 37-24
37.5 Emulate Actions 37-29
37.6 Display Actions 37-42
37.7 Automatic Primitives 37-44
37.8 Programming Example: Forcing Data Packets Out on the Line 37-45

38 SDLC 38-1

38.1 Frame-Level Setup 38-3
38.2 Protocol Trace 38-6
38.3 Monitor Conditions 38-11
38.4 Emulate-Mode Conditions 38-16
38.5 Emulate Actions 38-21
38.6 Display Actions 38-31
38.7 Automatic Primitives 38-33

39 SNA 39-1

39.1 Frame-Level Setup 39-3
39.2 SDLC Conditions and Actions 39-4
39.3 Protocol Trace 39-4

40 DDCMP Layer 1 40-1

40.1 Outsync 40-1
40.2 Block Checking 40-2

41 ISDN D Channel 41-1

42 LAPD 42-1

42.1 Frame-Level Setup 42-3
42.2 Protocol Trace 42-5
42.3 Monitor Conditions 42-9
42.4 Emulate-Mode Conditions 42-14
42.5 Emulate Actions 42-19
42.6 Display Actions 42-29
42.7 Automatic Primitives 42-31

43 Q.931 43-1

43.1 Protocol Trace 43-3
43.2 Monitor Conditions 43-9
43.3 Display Actions 43-12

44 SS#7 Layer 1 44-1

44.1 SS#7 Run-Time Displays 44-3
44.2 Setup for SS#7 44-3
44.3 SS#7 Compression at Layer 1 44-4

45 SS#7 Layer 2 45-1

45.1 Set Up for SS#7 45-3
45.2 Protocol Trace 45-3
45.3 Monitor Conditions 45-6
45.4 Display Actions 45-10
45.5 SS#7 Emulation 45-11
45.6 SS#7 Frame Structures and Values 45-11

46 SS#7 Layer 3 46-1

46.1 Packet-Level Setup 46-3
46.2 Protocol Trace 46-4
46.3 Monitor Conditions 46-7
46.4 Display Actions 46-10
46.5 Structure of SS#7 Message Signal Units 46-12

XII Optional Interfaces

47 V.35 47-1

47.1 Connectors 47-3
47.2 Green-Red LEDs 47-4
47.3 Breakout Box 47-4
47.4 Special Input and Output Pins 47-5
47.5 Screen Display of Lead Status 47-7
47.6 Monitoring V.35 Control Leads 47-7
47.7 Control of Lead Activity in Emulate Modes 47-7

48 RS-449 48-1

48.1 Connectors 48-3
48.2 Green-Red LEDs 48-4
48.3 Breakout Box 48-4
48.4 Special Input and Output Pins 48-5
48.5 Screen Display of Lead Status 48-7
48.6 Monitoring RS-449 Control Leads 48-8
48.7 Control of Lead Activity in Emulate Modes 48-8

49 X.21 49-1

49.1 Connectors 49-3
49.2 Green-Red LEDs 49-4
49.3 Breakout Box 49-4

49.4 Special Input and Output Pins 49–5
49.5 Screen Display of Lead Status 49–6
49.6 Monitoring X.21 Control Leads 49–8
49.7 Control of Lead Activity in Emulate Modes 49–8
49.8 Interface Control Menu Screen 49–8

50 RS–485 50–1

50.1 RS–485 Test Interface Module 50–3
50.2 Program Setup 50–6

51 ISDN 51–1

51.1 Overview of ISDN Testing Capabilities 51–3
51.2 Test Interface Module 51–6
51.3 Testing Configurations 51–6
51.4 Setting Up Menus for ISDN Testing 51–9
51.5 ISDN Interface Setup Menu 51–9
51.6 ISDN Runtime Statistics Screen 51–11

52 T1 52–1

52.1 Overview of T1 Testing Capabilities 52–3
52.2 The Test Interface Module 52–6
52.3 Testing Configurations 52–11
52.4 Setting Up Menus for T1 Testing 52–18
52.5 Interface Control Screen 52–19
52.6 Interpreting the T1 Statistics Display 52–27
52.7 D4 and Extended SuperFrame Structures 52–32

53 G.703 53–1

53.1 Overview of G.703 Testing Capabilities 53–3
53.2 The Test Interface Module 53–8
53.3 Testing Configurations 53–11
53.4 Setting Up Menus for G.703 Testing 53–16
53.5 Interface Control Screen 53–17
53.6 Interpreting the G.703 Statistics Display 53–26
53.7 Multiframe Structures 53–31

Part II: Advanced Programming

XIII Theory of Operation

54 Data Flow 54-1

54.1 Two Types of CPU 54-3
54.2 Front-End Buffer 54-5

55 Program Main 55-1

55.1 Translating a Simple Test into C 55-1
55.2 A Minimum of One Softkey State 55-2
55.3 Writing the Test Entirely in C 55-3

56 Regions in Spreadsheet 56-1

56.1 Layer and Test 56-1
56.2 State, Enter State, and Next State 56-2
56.3 Conditions and Actions 56-7
56.4 Example of Complete C Program 56-13
56.5 Summary of C Regions 56-14

57 Events 57-1

57.1 Example of Event: *fevar_time_of_day* 57-1
57.2 Various Origins of waitfor Events 57-2
57.3 Programming Considerations 57-3

58 Receiving and Transmitting Data 58-1

58.1 Locating Data in an IL Buffer 58-1
58.2 Monitor Path vs. Receive Path 58-3
58.3 Passing a Buffer Upwards 58-3
58.4 Layer 1 Transmit 58-4
58.5 Passing a Buffer Between Tasks 58-7
58.6 Sample Transmit Program: Sync or Async Echo 58-9
58.7 Sample Transmit Program: BOP Echo 58-10

XIV C Language

59 C Basics 59-1

59.1 Notable Variations in C 59-3
59.2 Editing a C Program 59-4

59.3 Error Reporting in C 59-4
59.4 Preprocessor Directives 59-5
59.5 Data Types 59-13
59.6 Operator Precedence 59-17
59.7 Strings 59-19
59.8 Recommended Sources 59-22

60 Variables 60-1

60.1 Creating or Accessing C Variables 60-1
60.2 Declaring Variables 60-2
60.3 Comparing a Variable to a Value 60-3
60.4 Checking a Variable in a Waitfor Clause 60-4
60.5 Checking and Displaying Equivalent Values of a Variable 60-7
60.6 Isolating Bits from a Variable Value 60-8
60.7 Pointing to an Address 60-10
60.8 Creating a Character Pointer 60-11
60.9 Pointing with Subscripts 60-12
60.10 Creating a String 60-12
60.11 Comparing Strings 60-14
60.12 Accessing a Variable Inside of a Structure 60-15
60.13 Creating a Structure Pointer 60-16

61 Routines 61-1

61.1 Declarations 61-1
61.2 Arguments 61-1
61.3 Returns 61-2
61.4 User-Defined Routines 61-3
61.5 Example Routines 61-4

XV C Library

62 Monitor/Transmit Line Data 62-1

62.1 Structures 62-1
62.2 Variables 62-2
62.3 Routines 62-7

63 EIA 63-1

63.1 Variables 63-1
63.2 Routines 63-3

64 Display Window and Trace 64-1

64.1 Current Display Mode 64-1
64.2 Prompt Line 64-3

64.3 Display Window 64-4
64.4 Program and User Traces 64-28
64.5 Attributes 64-39
64.6 Protocol Trace Buffers 64-45

65 Counters, Timers, and Accumulators 65-1

65.1 Counters 65-1
65.2 Timers 65-8
65.3 Accumulators 65-12
65.4 Routines 65-14

66 OSI 66-1

66.1 Structures 66-3
66.2 Variables 66-13
66.3 Routines 66-31

67 Print 67-1

67.1 Structures 67-1
67.2 Variables 67-3
67.3 Routines 67-4

68 Disk I/O 68-1

68.1 Streams 68-1
68.2 Routines 68-3

69 Status 69-1

69.1 Unit Configuration 69-1
69.2 Current Display Mode 69-1

70 Remote Port I/O 70-1

70.1 Structures 70-1
70.2 Variables 70-2
70.3 Routines 70-4

71 AUX Port I/O 71-1

71.1 Variables 71-3
71.2 Routines 71-5

72 Other Library Tools 72-1

72.1 Structures 72-1
72.2 Variables 72-3
72.3 Routines 72-8

XVI Protocol Library

73 X.21 Library 73-1

73.1 Structures 73-1
73.2 Variables 73-2
73.3 Routines 73-4

74 X.25 Layer 2 Library 74-1

74.1 Structures 74-1
74.2 Variables 74-1
74.3 Routines 74-9

75 X.25 Layer 3 Library 75-1

75.1 Structures 75-1
75.2 Variables 75-1
75.3 Routines 75-10

76 SDLC Library 76-1

76.1 Structures 76-1
76.2 Variables 76-1
76.3 Routines 76-9

77 SNA Library 77-1

77.1 Structures 77-1
77.2 Variables 77-1
77.3 Routines 77-5

78 DDCMP Library 78-1

78.1 Structures 78-1
78.2 Variables 78-1
78.3 Routines 78-1

79 ISDN D Channel Library 79-1

79.1 Structures 79-1
79.2 Variables 79-2
79.3 Routines 79-2

80 LAPD Library 80-1

80.1 Structures 80-1
80.2 Variables 80-1
80.3 Routines 80-9

81 Q.931 Library 81-1

81.1 Structures 81-1
81.2 Variables 81-1
81.3 Routines 81-5

82 SS#7 Layer 2 Library 82-1

82.1 Structures 82-1
82.2 Variables 82-1
82.3 Routines 82-5

83 SS#7 Layer 3 Library 83-1

83.1 Structures 83-1
83.2 Variables 83-1
83.3 Routines 83-8

Appendixes

Appendix A: Operator Messages A-1
Appendix A1: Interactive Messages A1-1
Appendix A2: Easy View Messages A2-1
Appendix A3: Error Messages Issued by C Translator A3-1
Appendix A4: Error Messages Issued by C Compiler A4-1

Appendix B: Glossary of Acronyms, Abbreviations and Mnemonics B-1

Appendix C: Selectable Data Speeds C-1

Appendix D: Code Charts D-1
Appendix D1: Keyboard-to-Hex Translation D1-1
Appendix D2: Hex-to-Display Translation D2-1
Appendix D3: User-Defined Codes D3-1

Appendix E: Communications with the AR Division Factory E-1

Appendix F: Packing and Shipping Instructions F-1

Appendix G: Rack Mount (OPT-951-98-1) G-1

Appendix H: Optional Codes JIS7/JIS8 ..... H-1

Appendix I: Interface Specifications I-1

Appendix J: Field Service on the INTERVIEW 7000 Series J-1
Appendix J1: Eliminating Static Electricity J1-1
Appendix J2: Removing Logic Boards J2-1
Appendix J3: Installing Logic Boards J3-1
Appendix J4: Installing Multiplexer Board J4-1
Appendix J5: Replacing Firmware on the CPM Board J5-1
Appendix J6: Installing the Hard Disk Drive (OPT-951-01-1) J6-1
Appendix J7: Servicing Other Components J7-1

Appendix K: C Language Summary K-1

Index

Index A: Technical Manual Part I and Part II

Index B: C Structures, Variables, and Routines

Table D3-1	Code-Set Characters	D3-4
Table I-1	Remote Connector	I-2
Table I-2	Printer Connector	I-3
Table I-3	Auxiliary Connector	I-4
Table I-4	RGB Monitor	I-5
Table I-5	RS-232 Test Interface Module	I-7
Table I-6	V.35 Test Interface Module	I-9
Table I-7	X.21 Test Interface Module	I-11
Table I-8	RS-485 Test Interface Module	I-13
Table I-9	RS-449/V.36/V.37 Test Interface Module	I-15
Table I-10	T-1 Test Interface Specifications	I-17
Table I-11	G.703 Test Interface Specifications	I-19
Table I-12	ISDN Test Interface Specifications	I-21
Table J3-1	S1-Switch Settings for MPM Boards in the INTERVIEW 7000 Series	J3-15

Table 66-8 Layer 7 OSI Variables 66–30
Table 67-1 Print Structures 67–2
Table 69-1 Status Structures 69–2
Table 70-1 Remote Port I/O Variables 70–3
Table 71-1 AUX Port I/O Variables 71–4
Table 72-1 Structure Fields—Other Library Tools 72–2
Table 72-2 Other Library Variables 72–4
Table 73-1 X.21 Structures 73–1
Table 73-2 X.21 Variables 73–3
Table 74-1 X.25 Layer 2 Structures 74–2
Table 74-2 X.25 Layer 2 Variables 74–3
Table 75-1 X.25 Layer 3 Structures 75–2
Table 75-2 X.25 Layer 3 Variables 75–3
Table 76-1 SDLC Structures 76–2
Table 76-2 SDLC Variables 76–3
Table 77-1 SNA Variables 77–2
Table 78-1 DDCMP Variables 78–2
Table 79-1 ISDN Structures 79–1
Table 79-2 ISDN Variables 79–2
Table 80-1 LAPD Structures 80–2
Table 80-2 LAPD Variables 80–3
Table 81-1 Q.931 Variables 81–2
Table 82-1 SS#7 Layer 2 Variables 82–2
Table 82-2 SS#7 Layer 1 Variables 82–5
Table 83-1 SS#7 Layer 3 Variables 83–2
Table A4-1 Numbered Error Messages Returned for C Coding A4–1
Table D1-1 Keyboard-to-EBCDIC D1–2
Table D1-2 Keyboard-to-ASCII D1–3
Table D1-3 Keyboard-to-EBCD D1–4
Table D1-4 Keyboard-to-XS-3 (SYN=35; EOM=55) D1–5
Table D1-5 Keyboard-to-IPARS D1–6
Table D1-6 Keyboard-to-REVERSE EBCD D1–7
Table D1-7 Keyboard-to-SELECTRIC D1–8
Table D1-8 Keyboard-to-BAUDOT D1–9
Table D1-9 Keyboard-to-JIS7 ..... D1–10
Table D1-10 Keyboard-to-JIS8 (space parity) D1–12
Table D1-11 Keyboard-to-JIS8 (mark parity) D1–13
Table D2-1 Hex-to-Display Translation D2–3

Table 46-3	SCCP Message Headers (Octet 9 for CCITT International or National Format; Octet 12 for US-ANSI Format)	46-19
Table 46-4	Telephone User Part (TUP) Message Headers	46-20
Table 46-5	ISDN Message Headers	46-22
Table 48-1	RS-449/RS-232 Conversion for Display and Triggering	48-7
Table 52-1	D4 Frame with In-Band Signalling	52-34
Table 52-2	ESF Frame with In-Band Signalling	52-35
Table 53-1	CAS Multiframe Structure — Signalling with Channel 16	53-32
Table 53-2	CCS/CAS Signalling with CRC-4	53-34
Table 59-1	Data Types: Ranges of Values Displayed and Printed	59-16
Table 59-2	Operator Precedence	59-18
Table 59-3	C String Non-Literals	59-21
Table 62-1	Transmit Structures	62-1
Table 62-2	Monitor/Transmit Variables	62-3
Table 62-3	Character Buffer 16-Bit Word	62-17
Table 63-1	EIA Variables	63-2
Table 64-1	Current Display Variables	64-2
Table 64-2	Display Window Variables	64-5
Table 64-3	Display Window Buffer Structures	64-8
Table 64-4	Display Window/Trace Buffer 32-Bit Data Word	64-14
Table 64-5	Special Graphic Character Set	64-17
Table 64-6	Trace Buffer Structures	64-30
Table 64-7	Conversion Specifiers	64-44
Table 64-8	Protocol Trace Buffer Variables	64-46
Table 64-9	Protocol Trace Buffer Structures	64-48
Table 65-1	Counter Structures	65-2
Table 65-2	Counter Variables	65-3
Table 65-3	Counter, Timer, and Accumulator Structures	65-5
Table 65-4	Timer Structures	65-9
Table 65-5	Timer Variables	65-10
Table 65-6	Accumulator Structures	65-14
Table 66-1	OSI Structures	66-9
Table 66-2	Layer 1 OSI Variables	66-14
Table 66-3	Layer 2 OSI Variables	66-15
Table 66-4	Layer 3 OSI Variables	66-18
Table 66-5	Layer 4 OSI Variables	66-21
Table 66-6	Layer 5 OSI Variables	66-24
Table 66-7	Layer 6 OSI Variables	66-27

List of Tables

Table 2-1	Fields That Can Be Changed Without Causing Recompile	2–16
Table 3-1	Frequently Used Programming Keys	3–4
Table 5-1	Synchronization Parameters for Standard Codes	5–8
Table 6-1	Special Display Symbols	6–14
Table 7-1	RAM Size (bytes) for Normal and High-Speed Recording in the INTERVIEW 7000 and the 7200 *TURBO*	7–8
Table 7-2	RAM Size (bytes) for Normal and High-Speed Recording in the INTERVIEW 7500 and the 7700 *TURBO*	7–8
Table 10-1	Standard Block-Check Parameters for Sync or Async/Isoc Formats	10–7
Table 10-2	Block Check Polynomials	10–7
Table 10-3	BCC Setup Menu Fields	10–11
Table 11-1	Full- or Half-Duplex BERT (RS-232/V.24 Interface Control screen)	11–7
Table 11-2	BERT Pattern (BERT Setup screen)	11–8
Table 11-3	Half-Duplex BERT: Default Synchronization Points	11–13
Table 11-4	Sync or Async BERT (Line Setup screen)	11–17
Table 12-1	Significance of On/Switch Selections on Interface Control Menu	12–13
Table 12-2	Standard Switched-Lead Configurations	12–15
Table 14-1	File Types	14–12
Table 15-1	Character Representations on Menu Screen Printouts	15–7
Table 15-2	Character Representations on Data Printouts	15–10
Table 26-1	Abbreviations, Trigger Summary Conditions	26–4
Table 26-2	Abbreviations, Trigger Summary Actions	26–5
Table 32-1	Valid Entries in Receive Strings	32–2
Table 32-2	Valid Entries in Transmit Strings	32–6
Table 33-1	Primitive Prefixes and Associated Layers	33–4
Table 33-2	OSI Service Primitives	33–11
Table 36-1	Default Parameters in SEND Actions	36–23
Table 36-2	Automatic Primitives Generated at X.25 Layer 2	36–34
Table 37-1	Automatic Primitives Generated at X.25 Layer 3	37–44
Table 38-1	Automatic Primitives Generated at Layer 2 SDLC	38–34
Table 39-1	Fields in SNA Trace Display	39–7
Table 42-1	Automatic Primitives Generated in LAPD	42–31
Table 45-1	LSU Status Field(Bits 2–0 of first Status Field Octet)	45–5
Table 46-1	MSU Service Indicators (Bits 3–0 of the SIO)	46–17
Table 46-2	Network Management Headers (Octet 9 for CCITT Format; Octet 12 for US-ANSI Format)	46–18