PERKIN ELMER

OS/32
ASYNCHRONOUS COMMUNICATIONS

Reference Manual

48-047 F00 R00

The Perkin-Elmer Corporation, Computer Systems Division 2 Crescent Place, Oceanport, New Jersey 07757

TABLE OF CONTENTS

CHAPTERS (Continued)

APPENDIXES

FIGURES

FIGURES (Continued)

TABLES

PREFACE

This manual is intended for users whose installations are configured with the following asynchronous devices:

- Model 550 video display unit (VDU)
- Model 1100 VDU
- Carousel 300
- Model 1200 VDU
- Models 1250/1251 VDUs
- Sigma 10 terminal
- Remote line printer
- Graphic display terminals

The reader should be familiar with the OS/32 basic data communications subsystem.

Chapter 1 defines asynchronous device support on a device independent level and a device dependent level. Chapter 2 discusses the hardware, software, and configuration procedures required for the terminal managers and the asynchronous line driver. Chapter 3 describes the asynchronous line driver that interfaces the communications adapter with the user program and allows the use of protocols not supported by a terminal manager, special buffering techniques, and data or command chaining. Chapter 4 describes the Teletype (TTY)/VDU terminal manager. Chapter 5 describes the features of the Models 1200/1250/1251 Editing VDU Terminal Managers, including multidrop, light pen, and downline load support.

This manual replaces S29-542 and provides device statements for the Sigma 10, Models 1250/1251 VDUs and the Perkin-Elmer remote line printer. It also adds information on the support of the current loop communications multiplexor (CLCM) and outlines the additional features supported by the terminal manager for use on the Models 1250/1251 VDUs. This manual pertains to the OS/32 6.0 software release and higher.

The following publications can be used in conjunction with this manual:

MANUAL TITLE	PUBLICATION NUMBER
M47-102 Programmable Asynchronous Single Line Adapter (PASLA) Maintenance Manual	29-301
Perkin-Elmer Carousel 300 Programming Manual	29-462
OS/32 Basic Data Communications Reference Manual	29-541
Model 1200 Terminal Installation and Programming Manual	29-631
Current Loop Communications Multiplexor Programming Manual	29-732
M47-100/101 Programmable Asynchronous Line System (PALS) Maintenance Manual	29-276
OS/32 System Macro Library Reference Manual	48-006
Models 1250/1251 Visual Display Units (VDU) Terminal Configuration User Guide	48-022
OS/32 System Generation (Sysgen) Reference Manual	48-037
OS/32 Supervisor Call (SVC) Reference Manual	48-038
OS/32 Application Level Programmer Reference Manual	48-039
OS/32 System Level Programmer Reference Manual	48-040
32-Bit Systems User Documentation Summary	50-003
Models 1250/1251 VDUs User's Manual	59-300-0048

For further information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1
GENERAL INFORMATION

1.1 INTRODUCTION

This reference manual describes asynchronous support of remote data terminals or computers via the OS/32 communications subsystem. The communications subsystem supports asynchronous devices on two levels:

- Device independent (or device transparent) level
- Device dependent (or device sensitive) level

Device independent level of access is achieved by issuing supervisor call 1 (SVC 1) to a terminal manager. This can be done by a user READ or WRITE macro. A terminal manager contains the logic to initiate, maintain, and terminate transmissions to a logical device called a terminal. The terminal manager calls the asynchronous line driver, which controls the data adapter interface and transfers data over a communication line. Two terminal managers are described in this manual:

- Teletype (TTY)/Video Display Unit (VDU) Terminal Manager
- Perkin-Elmer Models 1200/1250/1251 Editing VDU Terminal Managers

Device dependent level of access is achieved by a communication user task (u-task) directly accessing the same asynchronous line driver via SVC 15. This line driver is described in Chapter 3 of this manual.

CHAPTER 2
HARDWARE AND SOFTWARE CONFIGURATION PROCEDURES

2.1 HARDWARE

This section presents the asynchronous devices supported by the data communications subsystem. Figure 2-1 shows the interrelationships of the software and hardware associated with asynchronous communications.

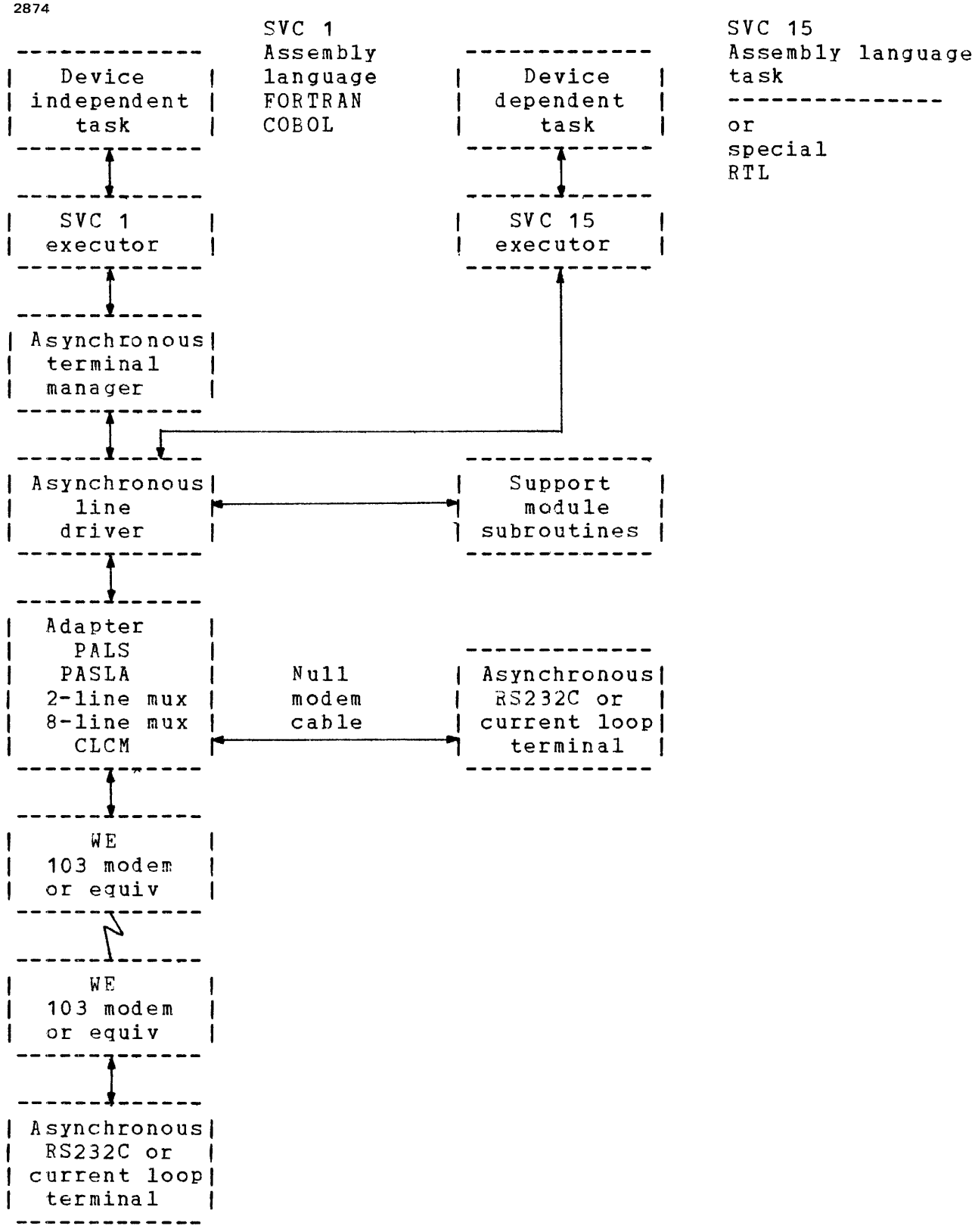


Figure 2-1 Functional Relationships of Device Dependent and Device Independent Asynchronous Support

2.1.1 Data Set Adapters

As shown in Figure 2-1, the first hardware device to respond to an SVC 1 or SVC 15 coming through the asynchronous line driver is one of the five data set adapters listed in Table 2-1.

TABLE 2-1 ASYNCHRONOUS ADAPTERS

DEVICE	PRODUCT NUMBER	DEVICE CODE
Programmable asynchronous line system (PALS)	M47-102	144
Programmable asynchronous single line adapter (PASLA)	M47-100	144
Two-line communications multiplexor (2-Line mux)	M47-104	144
Eight-line communications multiplexor (8-Line mux)	M47-105	144
Current loop communications multiplexor (CLCM)	M47-110	144

NOTE

PALS cannot be used with the Perkin-Elmer 3200 Series processors, only with the Model 7/32 or 8/32 processor.

2.1.2 Strapping Options

Strapping options must be taken into account when performing system generation (sysgen). The PASLA and the PALS provide the following strapping options:

- Baud Rates:
  - PASLA: 75 to 9600
  - PALS: 75, 110, 134.49, 150, 300, 600, 1200, 1800 (others available on special request)

PASLA has two clocks, and PALS has four clocks. The user selects the rate at which data is transmitted to and from the terminal. For details on how to strap baud rates to the clocks, see the Programmable Asynchronous Line System (PALS) Maintenance Manual or the Programmable Asynchronous Single Line Adapter (PASLA) Maintenance Manual.

- Data set ready
- Clear-to-send
- Carrier

These can be strapped so that they appear in a constant ready state to the CPU. When connecting a terminal to a modem via a null modem cable, these options are normally disabled.

- Full duplex (4-Wire). This allows the user to transmit messages in two directions simultaneously. Full-duplex transmission also allows echoplex operation in dumb terminals. Most local applications (cable connected) and all Western Electric 103 modem applications should be strapped 4-wire, and the sysgen statement for that device must indicate 4-wire.

- Half duplex (2-Wire). This is used for alternate send and receive mode when echoplexing is not required.

- Hardware address. This provides the program with the number of the communications line leading to a given hardware device. When strapped 4-wire, the sysgen statement for that device must specify the even hardware address.

The 2-line and 8-line multiplexors provide the following strapping options:

- Baud rates: 50 to 19,200 baud with these strap/clock combinations:

	SET1	SET2	SET3	SET4
CLOCK A	50	75	150	300
CLOCK B	110	134.5	600	1200
CLOCK C	1800	2000	4800	7200
CLOCK D	2400	3600	9600	19200

- The strapping options for data-set-ready, clear-to-send, carrier, full duplex, half duplex, hardware address, and ring are similar to PALS/PASLA.

The CLCM board is normally strapped for 150/600/4,800/9,600 baud rate operation. By modifying the board from the standard factory design, five alternative baud groupings are possible.

Each of the eight lines of the CLCM can be operated at one of 16 possible baud rates as follows: 50, 75, 110, 134.5, 150, 300, 600, 1,200, 1,800, 2,000, 2,400, 3,600, 4,800, 7,200, 9,600, or 19,200. A 4-bit code selects the baud rate for each line. Two bits of the code are programmable, allowing the selection of 1 of 4 baud rates. The remaining two bits are switch selectable, providing four subsets of 4 baud rates each.

2.1.3 Terminals

RS232C or CLCM compatible devices supported are:

TERMINAL	DEVICE CODE
Nonediting video display unit (VDU) (Models 550 and 1100)	147
Teletype (TTY) Keyboard/Printer (Carosuel 300)	147
Model 1200 Editing VDU	156
Models 1250/1251 Point-to-Point or Multidrop	157
Graphic Display Terminals	158
Sigma 10 Terminal	146
Remote Line Printer	145

2.1.4 Modems

The only modems that should be used to interface the asynchronous devices to the software described in this manual are:

Western Electric 103A, 103J, 113D, or equivalent modem (4-wire switched or leased)

Null Modem, M46-106 (cable for direct connection)

The following options should be selected when a 103J or 113D is used:

- Receive space disconnect NO
- Send space disconnect NO
- Loss-of-carrier disconnect NO
- CC indication EARLY
- CB and CF indications SEPARATE
- CC indication on analog loop ON

- Failsafe state of CN circuit OFF
- Automatic answer YES
- Common grounds YES

2.1.5 Cables

A cable is supplied with the PASLA to connect it to the convenience panel of a Perkin-Elmer system cabinet. This cable is Perkin-Elmer Part Number 17-197.

An optional null modem cable, Perkin-Elmer Part Number 17-197, is used to connect the convenience panel with the PASLA 7-inch interface and the following video display units:

- Low-end VDU (115V/60Hz)
- Low-end VDU (240V/50Hz)
- Alphanumeric VDU

The typical length of the null modem cable is 50 feet. It has some unusual characteristics; e.g., some wires loop back, which sometimes require that it be modified for terminals other than those cited previously.

A 2-line mux has one 17-463 ribbon cable connected between the connector at the edge of the board and the cable entry panel. An 8-line mux can have up to four 17-463 ribbon cables connected between the four connectors at the edge of the board and the cable entry panel.

An optional cable, Perkin-Elmer Part Number 17-050F02R02 (Product Number M10-054), is used to connect one communications line from the I/O convenience panel to a modem.

Up to two 17-522 ribbon cables can be used to connect the CLCM to the processor entry panel. To connect the processor entry panel to the current loop adapter, select from the list below one cable that meets the required length.

PART NUMBER	LENGTH
17-535 F01	30.48m (100')
17-535 F02	76.20m (250')
17-535 F03	152.40m (500')
17-535 F04	304.80m (1,000')
17-535 F05	762.00m (2,500')
17-535 F06	1,524.00m (5,000')

When using the Models 1250/1251 Terminal Managers to support the Models 1250/1251 VDUs in a multidrop environment, Perkin-Elmer recommends that the Spectron MIS-3400 modem splitter be used to connect the 1250 terminals to the software. Figure 2-2 designates which segments of the MIS-3400 must be strapped to provide compatibility with the terminal manager.

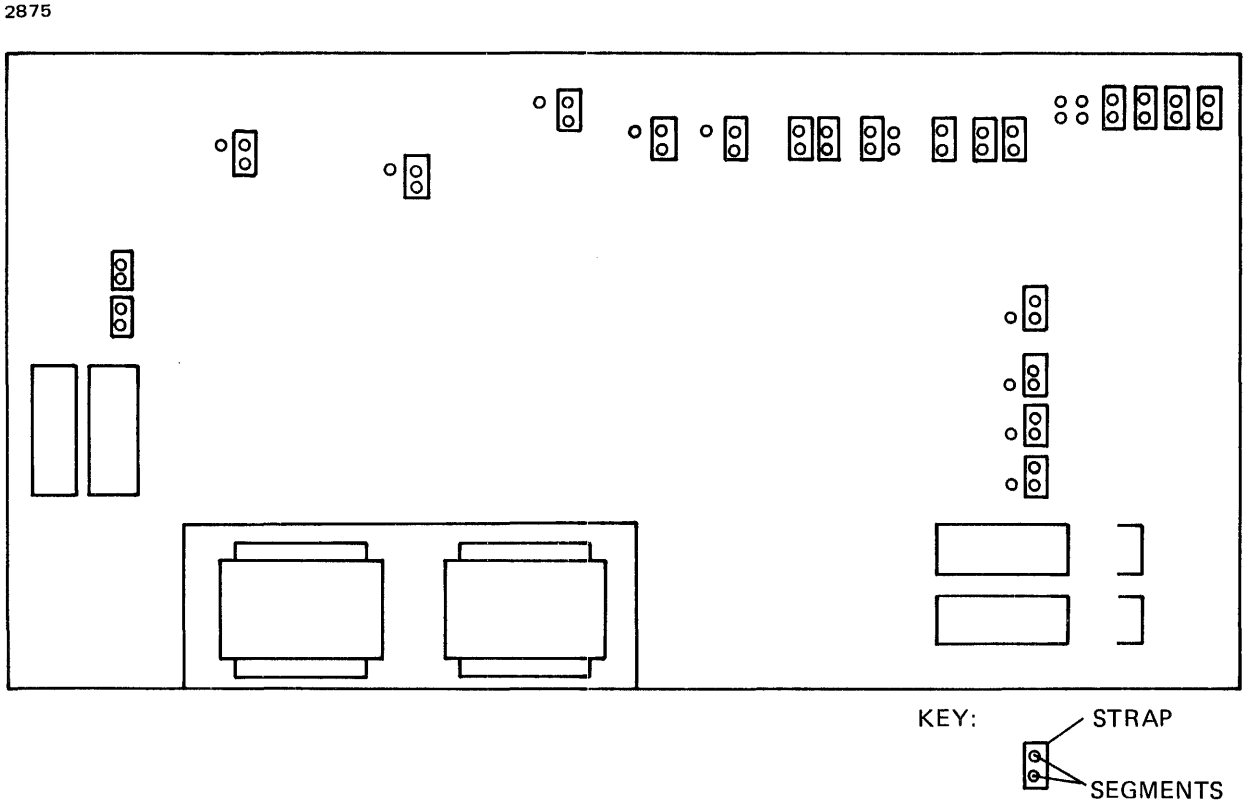


Figure 2-2 Spectron MIS-3400 Strapping for Models 1250/1251 Video Display Units (VDU) Multidrop Configuration

2.1.6 DMA I/O Subsystem (DIOS)

The Perkin-Elmer DMA I/O Subsystem can be used with an asynchronous system. This provides a significant gain for processor bandwidth. For Models 1200 and 1250/1251 VDUs, DIOS microcode contains subroutines to process special read/write characters.

2.2 SOFTWARE

The recommended sysgen procedure for an OS/32 configured with the communications subsystem is found in the OS/32 System Generation (SYSGEN) Reference Manual.

The device statements for asynchronous devices are given on the following pages. They must be specified as presented; no default is provided for the extended basic data communications options.

2.2.1 Conversational VDU/TTY

The VDU/TTY sysgen device statement format is:

```
1 dnem: adr,147,,Xxdcod,lrecl,Xrdcl,Xwtcl,pdseql
```

Where:

dnem	dnem is the device mnemonic.
adr	is the device address.
dcod	specifies the device code; 147 is the device code for conversational TTY/VDU.
Xxdcod	specifies the basic data communications extended device code. This value can be specified in either decimal or hexadecimal. If using hexadecimal, precede the numeric value with an X. See Table 2-2 for a description of options.
lrecl	specifies terminal logical record length. Normally this length is the number of characters that can be printed on each line. See Table 2-3 for a list of common terminal logical record lengths.
Xrdcl	specifies activated read control characters. Changing this value permits the user to specify which special characters may be used to terminate a line, as a line delete, or as a backspace. This value may be specified in either decimal or hexadecimal. If using hexadecimal, precede the number with an X. See Table 2-4 for a description of options.
Xwtcl	specifies activated write control characters. Changing this value permits the user to halt write on certain special characters, activate the carousel buffer overload protocol, or use an ESC character similar to a break. Normally, no halt characters are activated. Specify the carousel buffer overload protocol whenever a Carousel 300 series terminal is used. This value can be specified in either decimal or hexadecimal. See Table 2-4 for a description of options.

pdseql — specifies the length of the pad sequence appended to all conversational reads and writes. This value can range from 0 to 15. Normally, it should be set to 2 (LF, CR) for a VDU, or 3 (LF, CR, CR) for a TTY. For certain TTY devices, specify a larger pad count to allow adequate time for carriage return.

TABLE 2-2 EXTENDED DEVICE CODES

BIT	HEX MASK (DECIMAL VALUE)	MEANING
0	8000	Master/slave bit (processor-to-processor link only)
	8000 (32768)	Indicates that this end of processor-to-processor link is master
	0000 (0)	Indicates that this end of processor-to-processor link is slave
1-3	7000	Reserved - must be zero
4-5	0C00	Line configuration bits
	0800 (2048)	Automatic dial-in or manual dial-out
	0400 (1024)	Leased line
	0000 (0)	Directly connected (null modem cable)
6-7	0300	Line protocol bits
	0300 (768)	Half duplex 2-wire
	0200 (512)	Simplex write *
	0100 (256)	Simplex read *
	0000 (0)	Half duplex 4-wire *
8	0080	Explicit Connect Request bit
	0000	Indicates system will do an automatic connect if an SVC1 read/write request is issued

TABLE 2-2 EXTENDED DEVICE CODES (Continued)

BIT	HEX MASK (DECIMAL VALUE)	MEANING
		to a line that is not connected. Status returned is 8225. If the line is disconnected during read/write request, A0XX status is returned. Next read/write issued will cause system to automatically connect the line.
	0080	Indicates system will return error A018 if SVC1 read/write request is issued to a line that is not connected.
9	0040	Reserved - must be zero
10-11	0030	Clock bits (PALS/PASLA only)
	0030 (48)	Clock D
	0020 (32)	Clock C
	0010 (16)	Clock B
	0000 (0)	Clock A
12-15	000F	Default option index for Models 1200 and 1250/1251. Must be zero for all other devices.

* Requires adapter-strapped full duplex

TABLE 2-3 COMMON TERMINAL LOGICAL RECORD LENGTHS

TERMINAL	LOGICAL RECORD LENGTH
Model 550/550B	80
Model 1100 VDU	80
Model 1200 VDU	80
Models 1250/1251 VDUs	80
Carousel	128
M33 TTY	72
M35 TTY	80
Sigma 10 terminal	73
Remote line printer	132

TABLE 2-4 SPECIAL ASYNCHRONOUS CHARACTERS

TYPE	CHARACTER	ASCII	MEANING	READ MASK	WRITE MASK
Termination characters	CR	X'0D'	Carriage Return	X'8000'	X'8000'
	ETX	X'03'	End of Text (CTRL C)	X'4000'	X'4000'
	EOT	X'04'	End of Transmission (CTRL D)	X'2000'	X'2000'
	User-defined		Terminate Read/ Write	X'0200'	X'0200'
	Any enabled line delete character		Terminate Read on Line Delete	X'0100'	

TABLE 2-4 SPECIAL ASYNCHRONOUS CHARACTERS (Continued)

TYPE	CHARACTER	ASCII	MEANING	READ MASK	WRITE MASK
Backspace	BS	X'08'	Backspace (CTRL H)	X'0080'	
	<--	X'5F'	Back Arrow or Underscore (Shift 0)	X'0040'	
	User Defined	---		X'0010'	
Line Delete	#	23	Number sign	X'0108'	
	User Defined			X'0102'	
	NAK or	15	NAK (CTRL-U)	X'0101'	
	CAN	18	CANCEL (CTRL-X)	X'0101'	
Control	DC1, DC2	11,12	Carousel/Printer (START)		X'0400'
	DC3, DC4	13,14	Buffer Controls (STOP)		X'0400'
Break	BREAK		Break fulfills prepare		X'0001'
	ESC	X'1B'	Allow Escape to break write		X'0008'

2.2.2 Model 1200 Editing VDU

The Model 1200 Editing VDU sysgen device statement format is:

```
1 dnem: adr,156,,Xxdcod,lrecl,,,pdseql
```

See Section 2.2.1 for descriptions of the variables in this device statement.

156 specifies device code 156, Model 1200 Editing VDU.

xdcod — specifies the same communications extended device code as a conversational VDU. See Table 5-2 for additional Model 1200 VDU options.

2.2.3 Models 1250/1251 Point-to-Point VDUs

The Models 1250-1251 Point-to-Point VDUs sysgen device statement format is:

```
1 dnem: adr,157,,Xxdcod,lrecl,,,dseql
```

See Section 2.2.1 for descriptions of the variables in this device statement.

157 specifies device code 157, Models 1250/1251 Point-to-Point VDUs.

Xxdcod — specifies the same communications extended device code as a conversational VDU. See Table 5-2 for additional Models 1250/1251 VDU options.

2.2.4 Models 1250/1251 Multidrop VDU

The Models 1250/1251 Multidrop VDU sysgen device statement format is:

```
1 dnem: adr,158,,Xxdcod,lrecl,,,pdseql
```

See Section 2.2.1 for descriptions of the variables in this device statement.

158 specifies device code 158, Models 1250/1251 Editing VDUs.

Xxdcod — specifies the same communications extended device code as a conversational VDU. See Table 5-2 for additional Models 1250/1251 VDU options.

2.2.5 Supervisor Call 15 (SVC 15) Only Asynchronous Device

The SVC 15 only asynchronous device sysgen device statement format is:

```
1 dnem: adr,144,,Xxdcod,,Xrdcl,Xwtcl
```

See Section 2.2.1 for descriptions of the variables in this device statement.

144 specifies device code 144, SVC 15 only asynchronous device.

2.2.6 Sigma 10 Terminal

The Sigma 10 Terminal sysgen device statement format is:

```
1 dnem: adr,146,,Xxdcod,lrecl,Xrdcl,Xwtcl,pdseql
```

See Section 2.2.1 for descriptions of the variables in this device statement. 146 specifies device code 146, Sigma 10 Terminal.

NOTE

For local operation of the Sigma 10 Terminal use the following format:

```
1 dnem: adr,146,,,73
```

2.2.7 Remote Line Printer - DCOD 145

Format:

```
1 dnem: adr,145,,Xxdcod,lrecl,Xrdcl,Xwtcl,pdseql
```

See Section 2.2.1 for descriptions of the variables in this device statement. 145 specifies device code 145, Remote Line Printer.

NOTE

For local operation of the Remote Line Printer, use the following format:

```
1 dnem: adr,145,,,132
```

2.2.8 Module Specification

An additional option is available for the module statements. This option can be used to request inclusion of a user-assembled communications subsystem module. Acceptable formats for the module statement now include:

```
ITAM.xxx
```

where xxx is a unique identifier. Currently supported options are ITAM.M01, the standard communications module, and ITAM.U00 for a user-assembled module.

CHAPTER 3
ASYNCHRONOUS LINE DRIVER

3.1 INTRODUCTION

This chapter discusses the asynchronous line driver that allows users to:

- access terminals using protocols or codes not supported by a terminal manager,
- use special buffering techniques, and
- use data or command chaining to achieve a higher throughput rate.

The reader should be familiar with the basic data communications concepts, particularly supervisor call 15 (SVC 15), described in the OS/32 Basic Communications Reference Manual.

The asynchronous line driver provides an interface between the user program and the PALS, PASLA, 2-line mux, 8-line mux, or current loop communications multiplexor (CLCM) communications adapter which, in turn, support the terminals listed in Chapter 2. This line driver allows the user to specify the control sequences and data necessary to complete a transmission over a communications line.

To transmit or receive data over a communications line with RS232C interface leads, the asynchronous line driver communicates with PALS, PASLA, 2-line mux, or 8-line mux. To transmit or receive data over a communications line with current loop interface leads, the asynchronous line driver communicates with CLCM. The driver does not have to be aware of what devices are downstream from these adapters. The terminal manager written to support the driver, however, must be given the device attributes.

3.2 ASYNCHRONOUS LINE DRIVER AND SVC 15

The asynchronous line driver is an SVC 15 line driver; i.e., it can be accessed by a user SVC 15 or by the terminal manager. SVC 15 access to the line driver provides greater control over adapters, devices, and formats than terminal manager access. A user task (u-task) makes an I/O call with the SVC 15 instruction and parameter block. See the OS/32 Basic Data Communications Reference Manual for a detailed description of this parameter

block. Figure 3-1 illustrates the SVC 15 parameter block structure.

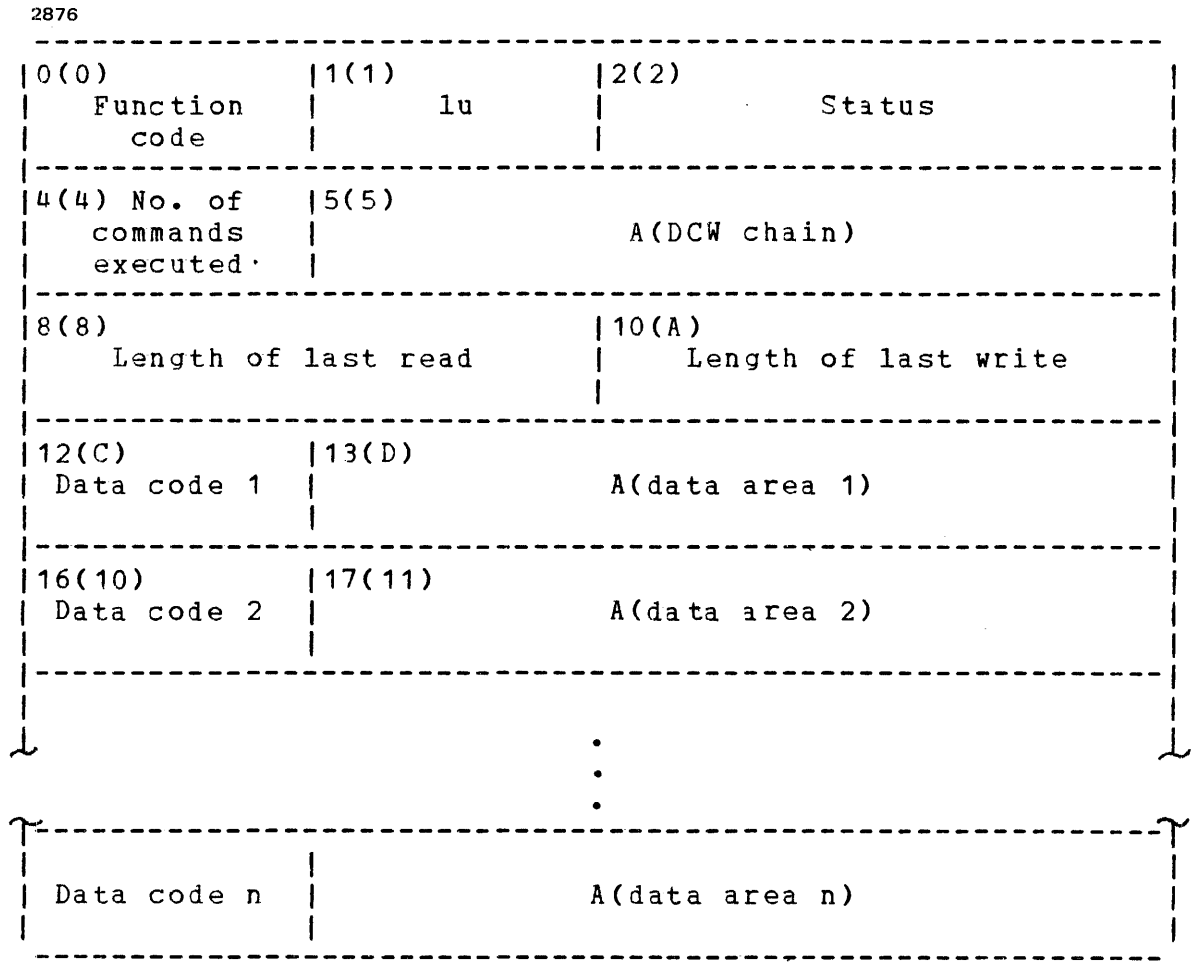


Figure 3-1 Supervisor Call 15 (SVC 15) Parameter Block

The first byte in the parameter block is a function code provided by the u-task. This function code specifies certain options applicable to the entire SVC 15. Figure 3-2 illustrates these options.

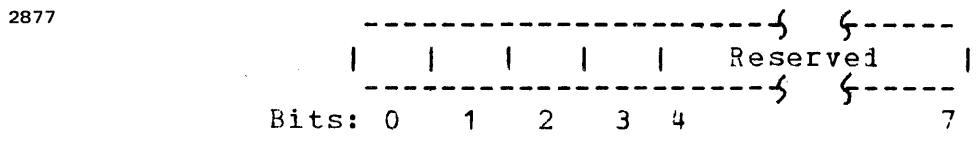


Figure 3-2 Function Code Format

Fields:

0 HALT I/O
An SVC 15 call with this bit set specifies that the u-task is requesting to halt an I/O operation that is in progress. The program status word (PSW) condition code indicates the results of the halt I/O call as follows:

cc=0 The halt I/O has been accepted and the original call can be considered to end with status indicating halt I/O. If an error occurred before the HALT I/O call, an error status will be returned. If the original call specified termination traps, a trap is generated for the original parameter block when the halt goes to completion.

cc=1 The halt I/O was not accepted because the driver was not performing any SVC 15 I/O to the logical unit (lu) specified for the task at the time of the call. The status field is not changed. I/O may have terminated normally just prior to the halt I/O call.

1 COMMAND QUEUE ENTRY ENABLE
This bit must be set, along with the corresponding bit in the driver command word (DCW) and the enable SVC 15 queue entry bit in the task status word (TSW), to allow a trap at the start of each DCW execution.

2 BUFFER QUEUE ENTRY ENABLE
This bit must be set, along with the corresponding bit in the DCW and the enable SVC 15 queue entry bit in the TSW, to allow a trap at the start of each buffer use associated with the DCW.

3 TERMINATION QUEUE ENTRY ENABLE
This bit must be set, along with the enable SVC 15 queue entry bit of the TSW, to allow a trap on normal or abnormal completion of the SVC 15. A halt I/O call does not generate a trap; however, the call being halted does if it originally specified termination queue entry enable.

4-7 Reserved for future use.

The second byte is the lu, also provided by the u-task. This byte specifies the device assigned for SVC 15 access.

The next halfword is for the communications subsystem status returned from the driver to the u-task. Figure 3-3 shows the SVC 15 status halfword.

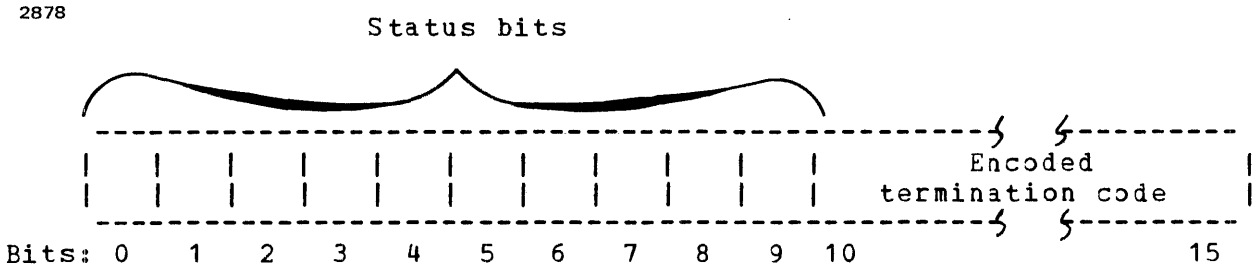


Figure 3-3 Supervisor Call 15 (SVC 15) Status Halfword

STATUS BIT (HEX)	MEANING	DESCRIPTION
0 (X'8000')	Error	Set for all error conditions and any termination code greater than 2.
1 (X'4000')	Busy	Driver is still busy with SVC 15 call. Can be cancelled via halt I/O.
2 (X'2000')	Transfer not begun	This bit is reset after the first character is received or sent.
3 (X'1000')	Timeout	Set for I/O timeout.
4 (X'0800')	Parity	Set on detection of parity errors. If the encoded error is also parity, the driver terminated on the error.
6 (X'0200')	Back space error	Attempt to back space over buffer limits using chained or queued buffers.
8 (X'0080')	Line delete detect	Line delete character detected in input data.

Table 3-1 lists the termination codes resulting from a terminated SVC 15. These codes occur independently of the status bits defined for Figure 3-3.

TABLE 3-1 ENCODED ERRORS AND DEFINITIONS

ENCODED (HEX)	STATUS	MEANING
00	No errors	No errors
02	Line delete	Line delete caused termination during read
03	Break on write	Break detected during write
04	Break on read	Break detected during read
05	Data check	Terminated by data error (see bits 4 and 5)
08	Framing error	Framing or stop-bit error
09	Reverse channel	Reverse channel error
0A	Loss of carrier	Lost carrier on read
0B	CL2S error	Lost clear-to-send on write
0C	Data set not ready	Data set not ready
0D	Device unavailable	Adapter not present
0E	Overflow	Character overflow
0F	Ring	Ring status detected during data transfer
10	Buffer overrun-1	Busy and/or done bits in chained buffers bad; may indicate priority too low
11	NCE overflow	Number of commands executed greater than 255
12	Task queue error	Task queue full, invalid, or nonexistent
13	Buffer overrun-2	Buffer-management-routine error; may indicate priority too low
14	Timeout	Timeout
15	Halt I/O	Halt I/O request aborted I/O
18	Illegal command	Command or modifier not valid; e.g., switched line not connected
19	Memory fault-1	Memory fault in referencing data
1A	Memory fault-2	Memory fault in referencing buffer
1B	Illegal lu	logical unit illegal (not SVC 15 or not assigned)
1C	Illogical status	Device status not valid; might be hardware problem
1D	Power fail	Power failure occurred during I/O
1E	Illegal condition	Illegal software condition detected; might be caused by user-written drivers
1F	Illegal translation table	Attempt to use a nonexistent translation table
23	Queue empty	Queued-buffer list empty
24	Queue overflow	Queued-buffer list overflow
27	DIOS error	DMA I/O system (DIOS) hardware error

Byte 4 indicates the number of commands executed. This information, returned by the driver to the u-task, consists of the number of DCWs fetched and executed (not necessarily error free).

The next three bytes, provided by the u-task, must contain the address of the first halfword of the user DCW chain. A DCW chain consists of consecutive driver command halfwords with their respective chain option bits set. DCWs are halfwords, each of which specify to the driver a particular functional operation to be performed, such as read or write. They also specify certain options applicable for the duration of the command. Figure 3-4 shows the format of the DCW.

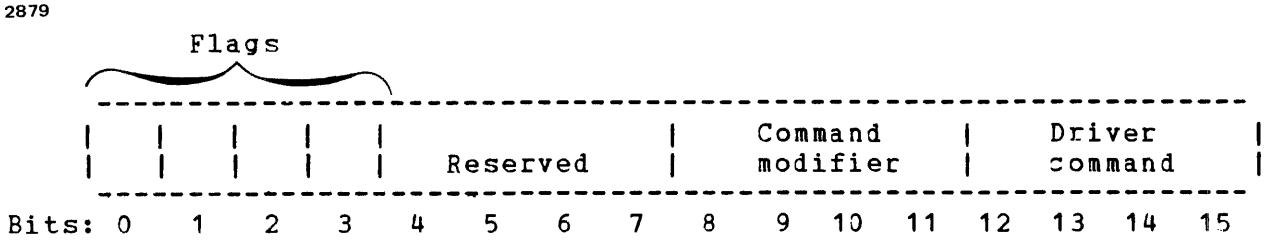


Figure 3-4 Driver Command Word Format

Table 3-2 shows the options enabled by setting the appropriate bits in the flags field of the DCW.

TABLE 3-2 DRIVER COMMAND OPTIONS

FLAG	DESCRIPTOR	DESCRIPTION
Bit 0	Chain command	After this command is executed, if this bit is set, the next command in sequence is executed. Otherwise, the driver terminates.
Bit 1	Command trap	If set and enabled by the function code, a trap indicating command trap is generated to the calling task before this command is executed.

TABLE 3-2 DRIVER COMMAND OPTIONS (Continued)

FLAG	DESCRIPTOR	DESCRIPTION
Bit 2	Buffer trap	If set and enabled by the function code, a buffer trap is generated after each buffer is transmitted or filled.
Bit 3	Timeout	If set, an error timer is initialized before the command is executed. If the timer expires before the command goes to completion, the entire SVC 15 is aborted with a timeout status. If not set, the timer is stopped and the command does not timeout. There are separate error time values for read and write.

The next fullword in the parameter block returned by the driver to the u-task consists of two halfwords. They indicate, respectively, the length of the last read and the length of the last write performed by this I/O call.

The remainder of the parameter block consists of data fields required by the DCW chain. Each data field consists of a 1-byte code and a 3-byte data address. Data codes indicate the type of buffering desired. The data address always points to the data required by the corresponding DCW. (This could be no data or one or more data fields, depending on the particular command.) See Appendix A for the DCW halfword binary format. See the OS/32 Basic Data Communications Reference Manual for a detailed description of data codes and buffering techniques. Buffers are described later in this chapter.

3.3 COMMANDS SUPPORTED

The asynchronous line driver supports the following commands:

- Null type:
  - XFER
  - CXFER
  - WAIT
  - NOP

- Control type:

  - EXAMINE
  - RING WAIT
  - ANSWER
  - DISCONNECT

- Read type; using standard data communications subsystem buffer management:

  - READ
  - READ1
  - READ2

- Prepare type; for one single character:

  - PREPARE

- Write type; using standard data communications subsystem buffer management:

  - WRITE
  - WRITE1
  - WRITE2

- Hold type:

  - HOLD space (transmit break)

- Mode type; used to modify certain programmable adapter options, allowing one asynchronous line driver to communicate with different types of asynchronous terminals:

  - TOUT
  - CMD2
  - RCMD
  - WCMD
  - RDISABLE
  - WDISABLE
  - DISC
  - TRANSL
  - SPCHAR

All mode commands receive an address from a data field. This address points to a byte, halfword, or fullword field containing an output instruction or an error timeout value. The contents of this field are stored in the DCB for use by the asynchronous line driver.

If the default values specified in the DCB are acceptable, no mode commands need be executed. Once a value is changed by a mode command, the only means of restoring the default condition is by a mode command specifying the correct value. It is necessary to coordinate such modification if access is being shared by more than one program.

The chain command and command trap bits of the DCW can be used in conjunction with the mode commands.

These commands are described in detail below:

COMMAND	MODIFIER/ COMMAND BYTE (HEX)	MEANING
XFER	XX10	This command obtains one data field, which specifies the address of the next DCW. This command must be chained.
CXFER	XX18	This command obtains two data fields; the first specifies the address of two consecutive halfwords. The first halfword is a mask that is ANDed with the present status of the communications subsystem. The second halfword is compared with the result from above. If equal, the next command address is specified by the second data field. If unequal, the next command in sequence is executed. This command must be chained.
WAIT	XX08	This command obtains one data field that specifies the address of a halfword containing a timeout count in units of 100 milliseconds. This delay is performed before continuing with command word chain processing.
NOP	XX00	This command obtains one data field, which is ignored. It is useful for reserving space in both the command chain and the DCW chain. The data field of this command must specify a valid address.
EXAMINE	XX01	This command obtains one data field. The value obtained specifies the address of a writable byte into which the status of the device is stored. The last known device status is fetched from a byte in

		memory that is maintained by the driver during I/O activity. If the byte is nonzero, its contents are returned to the user and it is reset to 0. If the byte is 0, a sense status is performed on the device and the device status is returned to the user.
RING WAIT	XX09	This command fetches no data fields. Interrupts from the adapter are enabled; however, the data terminal ready lead to the modem is not. The command terminates when an interrupt is received with ring status set. If chain command is set, execution continues with the next command; otherwise, the driver terminates. If timeout is set, the command only waits as long as the value specified in the write error time value. If this interval expires, timeout error status is set. If timeout is not set, the command waits indefinitely.
ANSWER	XX11	For nonswitched lines and switched lines already connected, this command immediately terminates. For dial-in lines that are unconnected, the data terminal ready lead to the modem is enabled, causing the modem to answer when the data set indicates it is ready for I/O. Timeout and chain commands are handled as described in the RING WAIT command.
DISCONNECT	XX19	This command disables the data terminal ready lead to the modem, causing a disconnect on a switched line. The command waits for one second, then continues to the next command (if chain command is set) or terminates (if reset).
READ	XX02	This command obtains either one or two data fields, depending on which one of the three standard data communications buffer management techniques is specified in the data code of the first data field obtained. For indirect and chained buffers, one data field is obtained; for direct and queued buffers, two fields are obtained. All buffers must be in the same logical segment of the task as the address contained in the first data field of the parameter block. The command terminates normally when all buffers are

exhausted or a termination character is received. If timeout is requested and the I/O does not terminate normally before the timer expires, then I/O is aborted and timeout status is returned. The special characters recognized during read and their enable mask values are shown in Table 2-4.

READ1	XX0A	This command obtains one data field that specifies the address of a writable byte into which a character is read. The command terminates after reading one character. If timeout is requested in the command, the read error time value (changeable by a MODE command) is used.
READ2	XX12	This command is similar to the above command except that two bytes are read and stored.
PREPARE	XX03	This command obtains one data field that specifies the address of a 1-byte data area. Characters are read from the adapter, and the command terminates when a character is read equal to the contents of the data area. The characters read are not stored and are lost. Receipt of break can optionally fulfill the requirements of the PREPARE command (controlled by the special character write enable bit X'0001'). When the prepare criterion is fulfilled (correct character received, or break if enabled) the command is considered executed and the next command, if chained, is executed. If timeout is specified in the command, then the read time value is used. The entire SVC 15 terminates with timeout status if the PREPARE command is not satisfied within the allowed time.

NOTE

The special case of PREPARE chained to a READ results in a lookahead to set up the read buffer. In this way, an input stream can be searched for a special key character and then the text that follows can be read.

WRITE	XX04	This command obtains either one or two data fields based on the same criterion as in READ. The command terminates normally when all buffers are exhausted or a termination character is detected in the data being transmitted. The asynchronous line driver performs special character recognition during write operations in an analogous fashion to the read. Each character is enabled via a halfword bit mask changed via the mode SPCHR command. Backspace and line delete characters are of no practical value in a write situation, and none are defined. The ending characters for write and their enable masks are shown in Table 2-4.
WRITE1	XX0C	This command obtains one data field that specifies the address of a byte of data transmitted to the adapter. The command terminates after the character is transmitted.
WRITE2	XX14	Same as WRITE1, except two characters are transmitted. If detected in a data stream being transmitted, the ending character is transmitted and the write is terminated.

NOTE

If echoplex is specified for READ, an extra pad character (X'FF') is sent after the last character is written for all WRITEs. (The pad is not sent if the image translation table is specified.)

HOLD	XX05	Chain command and command trap are the valid flag bits. This command obtains one data field that specifies the address of a halfword containing a timeout count in units of 100 milliseconds. The driver transmits a continuous space (line break or a character equal to zero) for the specified interval, after which the command terminates. Clock resolution is +0, -1 clock unit of 100.

TOUT XX06

This command obtains the address of two halfwords that specify error timeout values in seconds. If any command specifies timeout, this time value is placed into the DCB.TOUT field of the DCB and is decremented every second by the system clock. If the particular command is not completed within the allotted time, the entire SVC 15 is aborted and the timeout status bit is set. If no other encoded errors are indicated in the status field, the timeout code is also placed in the encoded portion of the status.

If the timeout status bit is set and the encoded error is not timeout, the encoded error occurred first and might, in fact, be the reason for the timeout.

There are separate time values for read and write. The data field of the MODE TOUT command specifies a fullword. The first halfword obtained by mode TOUT is the time value for all READ commands and the second halfword is the time value for all WRITE commands. Zero is not a valid time value. Precision range is +0, -1 second.

2880

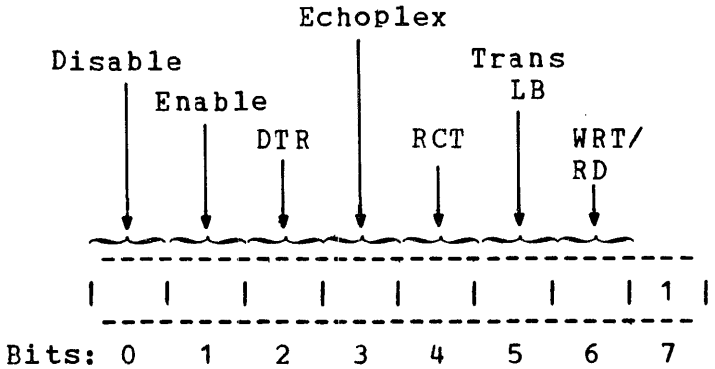


Figure 3-5 Adapter Command 1 (CMD1)

RCMD XX16

This command obtains the address of the byte specifying the asynchronous driver adapter command used for read operations. This command is then stored in the DCB.MOCR field. The asynchronous driver communicates with the adapter through this command. For example, whenever an adapter is to perform a read operation, the driver issues the read command stored in the DCB.MOCR field by RCMD. (See Figure 3-5).

For read operations the adapter command should normally specify:

ENABLE+
DATA TERMINAL READY + (DTR)
The proper combination for read and (optionally) echoplex. See the OS/32 Basic Data Communications Reference Manual.

This must be consistent with the adapter type, its particular strapping, and the modem. If the echoplex is used (normally only on 4-wire) it should be specified only in the READ command. See Figure 3-5.

RDISABLE XX26

This command obtains the address of the byte containing the adapter command that is stored in the DCB.DOCR field for use by the driver. This adapter command disables interrupts from the read side of the adapter after each completed DCW READ and WRITE command and after any error conditions. It should normally leave the adapter and modem in the read mode and specify:

DISABLE +
DATA TERMINAL READY+
Proper combination for READ. See Figure 3-5. See the OS/32 Basic Data Communications Reference Manual.

WCMD or WDIS XX1E

This command obtains the address of the write adapter commands that are stored in the DCB.MOCW and DCB.DOCW fields for use by the asynchronous driver.

The write commands WCMD and WDIS are used similarly to the READ commands and should normally specify:

WCMD: ENABLE+
DATA TERMINAL READY+
Proper combination for WRITE

See the OS/32 Basic Data Communications Reference Manual.

WDIS: DISABLE+
DATA TERMINAL READY+
Proper combination for WRITE

See Figure 3-5 and the OS/32 Basic Data Communications Reference Manual.

The write commands leave (or disable) the line in the write state, with data terminal ready; while the read commands disable the line in the read state, with data terminal ready. This allows the user to string several write or read commands together so the disable at the end of each command does not result in dropping and/or subsequent raising of request to send (RQ2S), unless the command does indicate a change of state.

CMD2 XX0E

The format of the I/O command byte 2 obtained by the mode CMD2 command is shown in Figure 3-6. Because programmable adapter options can be set via the mode CMD2 command, the bit pattern for the CMD2 is determined by the user's particular installation. Setting bit 7 to 0 allows the user to select the following clock adapter options: number of bits, number of stops, and parity (odd, even, or none).

2881

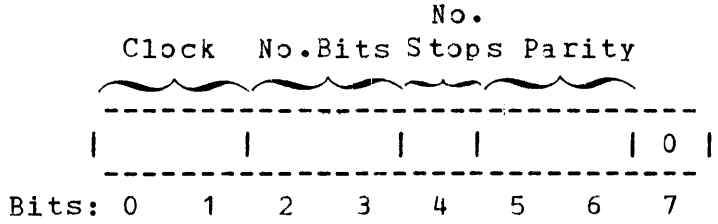


Figure 3-6 Adapter Command 2 (CMD2)

TRNSL XX00

This command allows the user to modify the default translation table. It obtains the address of a one-byte field specifying the type of translation table to be used by the driver. This byte can contain the following values:

X'00' Normal translation (7-bit ASCII, strip off and ignore parity)

X'01' 8-bit image translation

The address of the translation table to be used is stored in the DCB.XLT field.

The only ending characters recognized by the image translation table are CR (X'0D' or X'8D') and ETX (X'03' or X'83'). The characters must be enabled to become effective.

SPCHR XX4E This command obtains the address of a fullword field specifying the special character enable mask halfwords for read and write operations. SPCHR stores the halfwords in the DCB.SPCR and DCB.SPCW fields for use by the asynchronous line driver. The first halfword is the read enable mask, and the second halfword is the write enable mask. The format of each mask is shown in Table 2-4.

3.4 SPECIAL CHARACTER PROCESSING

The special characters that can be used during read or write operations are divided into four major categories:

- Ending or Terminating Characters - These characters, when encountered in the data stream, are stored in the buffer or sent down the communications line and result in termination of that particular READ or WRITE command. If the command is chained, the next command is executed. See Table 2-4.

- Line Delete Characters - When these characters are detected in the input data stream, the line delete special character bit is set in the status halfword. If the line delete is also enabled as a terminating character, the entire SVC 15 is aborted with the encoded status indicating line delete. See Table 2-4.

- Backspace Characters - These characters, when encountered in the input stream, result in the equivalent of a backspace; i.e., the previous character is backspaced over. See Table 2-4.

- Miscellaneous Other Characters - Miscellaneous characters have variable meanings, depending on the type of terminal involved. These characters can be used to turn off or on a specific operating character for a predefined terminal. See Table 2-4.

Since several terminal or line procedures use different characters for the previous purposes, the asynchronous line driver recognizes several characters in each category. Special character recognition for each character can be separately

enabled or disabled by a bit mask. The mask consists of two halfwords that indicate the characters this particular device allows for a given purpose on read and write. These two halfwords are modified by the mode SPCHR command explained in the description of the mode commands.

When using chained or queued buffers, it is not possible to backspace over buffer limits. (The data communications subsystem allows the user to manipulate buffers other than those currently in use by the driver.) If a backspace that would back over a buffer limit (except the first buffer) is attempted, the backspace function is disabled and the backspace over buffer limit status is set. The result is that this and all future backspace characters are stored in the buffer. Reissue the read or attempt to perform user backspace handling if this status occurs.

ESC

Setting the escape bit allows receipt of ESC during a write to be treated as if a break were received. Escape is ignored if this bit is reset.

BUFFER CONTROL

The Carousel control codes (DC2 and DC4 in Table 2-4) are sent by the Carousel to pause (DC4) and continue (DC2) the computer output as the Carousel's input buffer fills or empties past predetermined limits. The write enable bit for the Carousel must be set to allow the asynchronous line driver to react to these control characters during output. If the Carousel control code bit is reset, receipt of these characters during write is ignored.

NOTE

The control characters DC1 and DC3 used by the remote printer are handled in the same manner as DC2 and DC3, respectively.

3.5 DEFAULT VALUES

The default values, assembled in DCB144, DCB145, DCB146, DCB147, DCB148, DCB149 and DCB150 for the previous mode commands are shown below:

```
TOUT        DC  H'30',H'30'  30 seconds for read, 30 seconds
                              for write
CMD2        DB  X'38'        8-bit data characters, no parity,
                              2 stop bits
RCMD        DB  X'71'        ENABLE, DTR, ECHOPLEX, READ
WCMD        DB  X'63'        ENABLE, DTR, WRITE
RDIS        DB  X'A1'        DISABLE, DTR, READ
WDIS        DB  X'A3'        DISABLE, DTR, WRITE
DISC        DB  X'B1'        DISABLE, READ
TRNSL       DB  0            7-bit ASCII, strip off parity
SPECCHAR    DCX 81E8,0000    All characters enabled
```

The special character format for DCB144 is:

```
SPECCHAR      DCX F9ED,6004
```

Ending sequence count for DCB149 and 150 is:

```
SYCT          DB X'2'          LF, CR as ending sequence
```

3.6 BUFFERS AND TRAPS

The following information on buffers appears in Chapter 3 of the OS/32 Basic Data Communications Reference Manual.

SVC 15 specifies the first entry in each of the following two related chains used to define the request:

1. The DCW chain, which specifies the sequence of operations to be performed by the driver; i.e., READ, WRITE, etc

2. The data field chain, which specifies the arguments required by each driver command in the DCW chain

SVC 15 activates the line driver which fetches and executes the first DCW in the DCW chain. Once autonomous driver execution is started, control is returned to the user task with the condition code indicating the result of the call. If no error is encountered in initiating the operation specified by the first DCW, the status field of the SVC 15 parameter block is set to indicate that the line driver is active with the request. For the remainder of the I/O request, as each command operation is

completed, the next operation specified in the DCW chain is fetched and executed by the line driver at the priority of the calling task. This sequence of fetch and execute is repeated until the entire DCW chain is interpreted or an error condition is encountered.

To monitor the progress of SVC 15 and provide facilities for buffer management, the task can specify that a trap is to be generated at the start of the driver command execution, at the time the line driver starts to use a buffer, and/or at termination of SVC 15. These traps allow the task to synchronize execution with the concurrent processing of the SVC 15 request. When traps are enabled, and a trap-causing event occurs, the task trap handling routine is given control before any subsequent task level instruction can be executed. Remember that the trap-handling routines are operating at a lower priority than the line driver; thus, several entries can be made to the task queue before the trap handling routine completes processing a single entry.

Both the DCW and data field chains are usually interpreted by fetching the next required entry from the memory location immediately following the last entry processed. Special entries allow each chain to be contained in nonadjacent areas in memory. There is a DCW chain entry specifying that the data field points to the next DCW chain entry. Similarly, there is a data field chain entry pointing to the next entry in the data field chain instead of containing the address of a data area. The only restriction is that all buffers specified in one SVC 15 request must be contained in the same logical program segment. See the OS/32 Application Level Programmer Reference Manual for a discussion of program segments.

The user uses the SVC 15 parameter block to specify a control sequence to be performed by pointing to a DCW chain. Through the SVC 15 parameter block, the user also specifies the data areas associated with each driver command by pointing to a chain of data fields. A data field, illustrated in Figure 3-7, consists of a 1-byte code indicating the data field type and a 3-byte pointer to the data described by the data field.

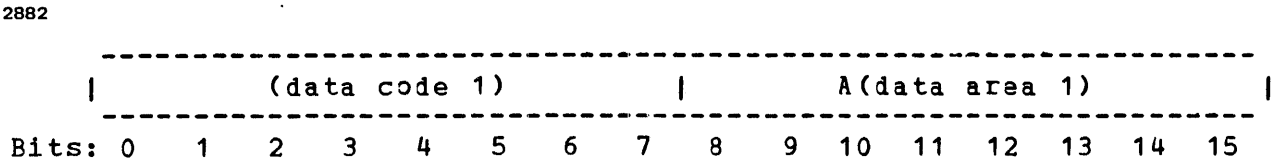


Figure 3-7 Supervisor Call 15 (SVC 15) Data Field

DATA CODE (HEX)	CONTENTS
00	Address of direct buffer
04	Address of indirect buffer
08	Address of chained buffer
0A	Address of queued buffer list

3.6.1 Buffer Types

There are three buffer types used by the asynchronous line driver:

- Direct
- Indirect
- Chained or Queued

These buffer types and their respective data codes are described in the following sections.

3.6.2 Direct Buffers (Data Code X'00')

A direct buffer requires two data fields in the data field chain. These data fields contain the start and end addresses of the buffer. A direct buffer is similar to an SVC 1 data buffer. The start address points to the first data character; the end address points to the last data character. A 1-character buffer has a start address equal to the end address. Direct buffers can begin on any byte boundary. See Figure 3-8.

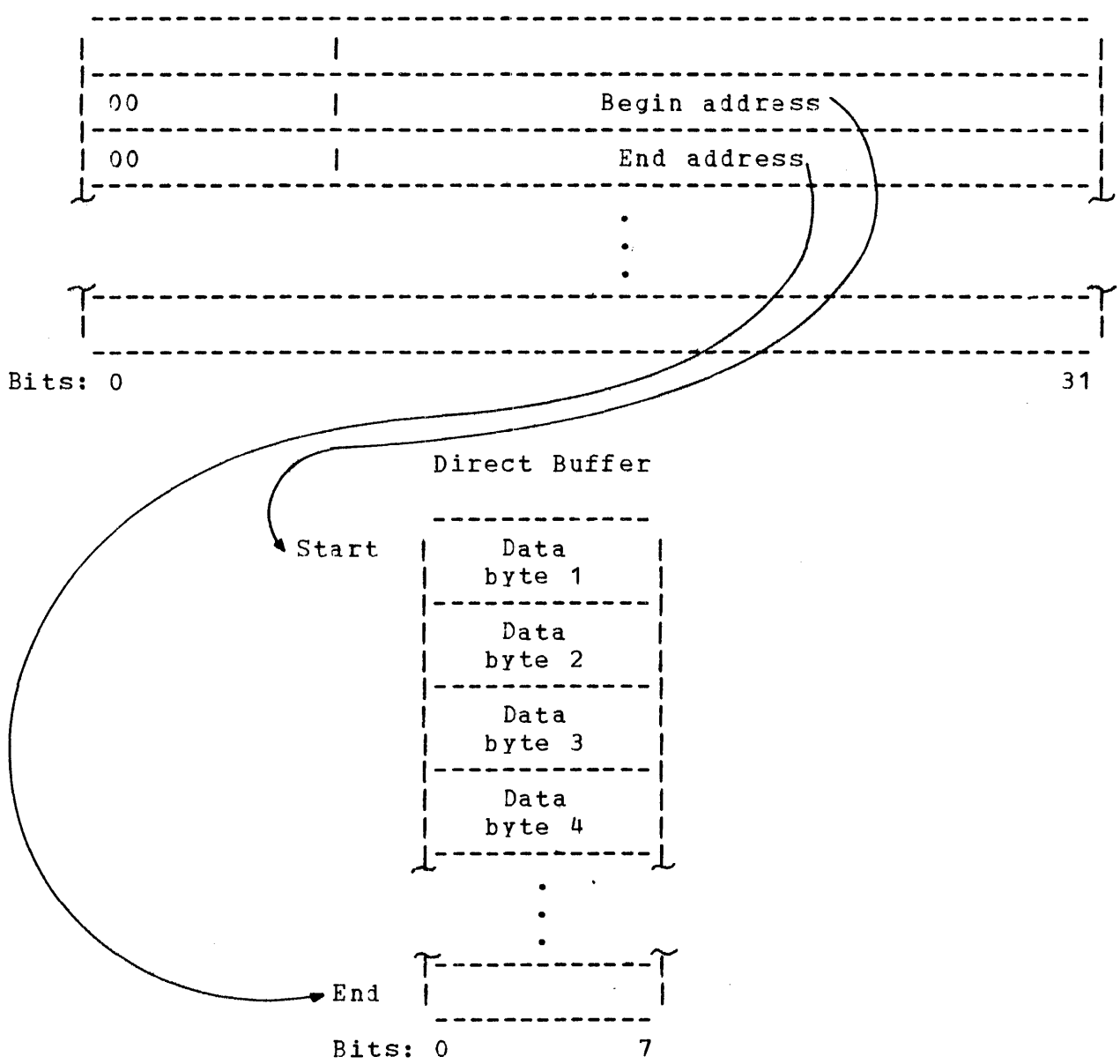


Figure 3-8 Direct Buffer

3.6.3 Indirect Buffers (Data Code X'04')

An indirect buffer is specified by one data field containing its start address. The buffer itself contains all required size information. The first halfword indicates the number of bytes available in the buffer. The second halfword of the buffer is updated by the driver; it indicates how many bytes of data were actually transferred by the I/O operation. An indirect buffer can be aligned on a halfword boundary. See Figure 3-9.

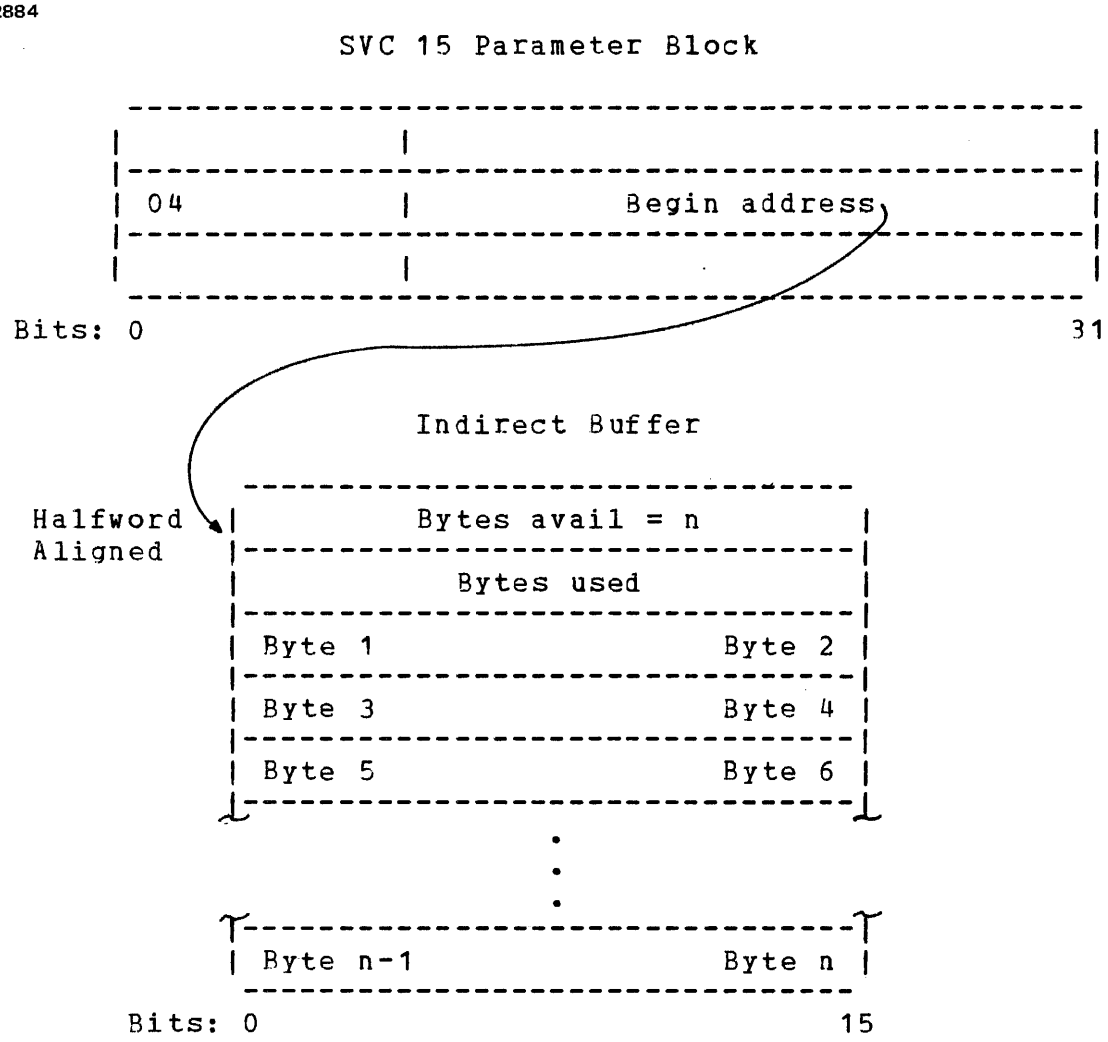


Figure 3-9 Indirect Buffer

3.6.4 Chained Buffers (Data Code X'08')

Chained buffers are specified by one data field containing the address of the first buffer in the chain. Chained buffers are like indirect text buffers but have an additional fullword at the beginning. This fullword is called the link word and can contain the address of another chained buffer. Thus, two or more buffers can be linked together into a chain. The last buffer in a chain of linked buffers contains a zero link word indicating the end of the chain. Chained buffers can also be configured into a closed chain (a ring) if the last buffer links back to the first buffer. See Figure 3-10.

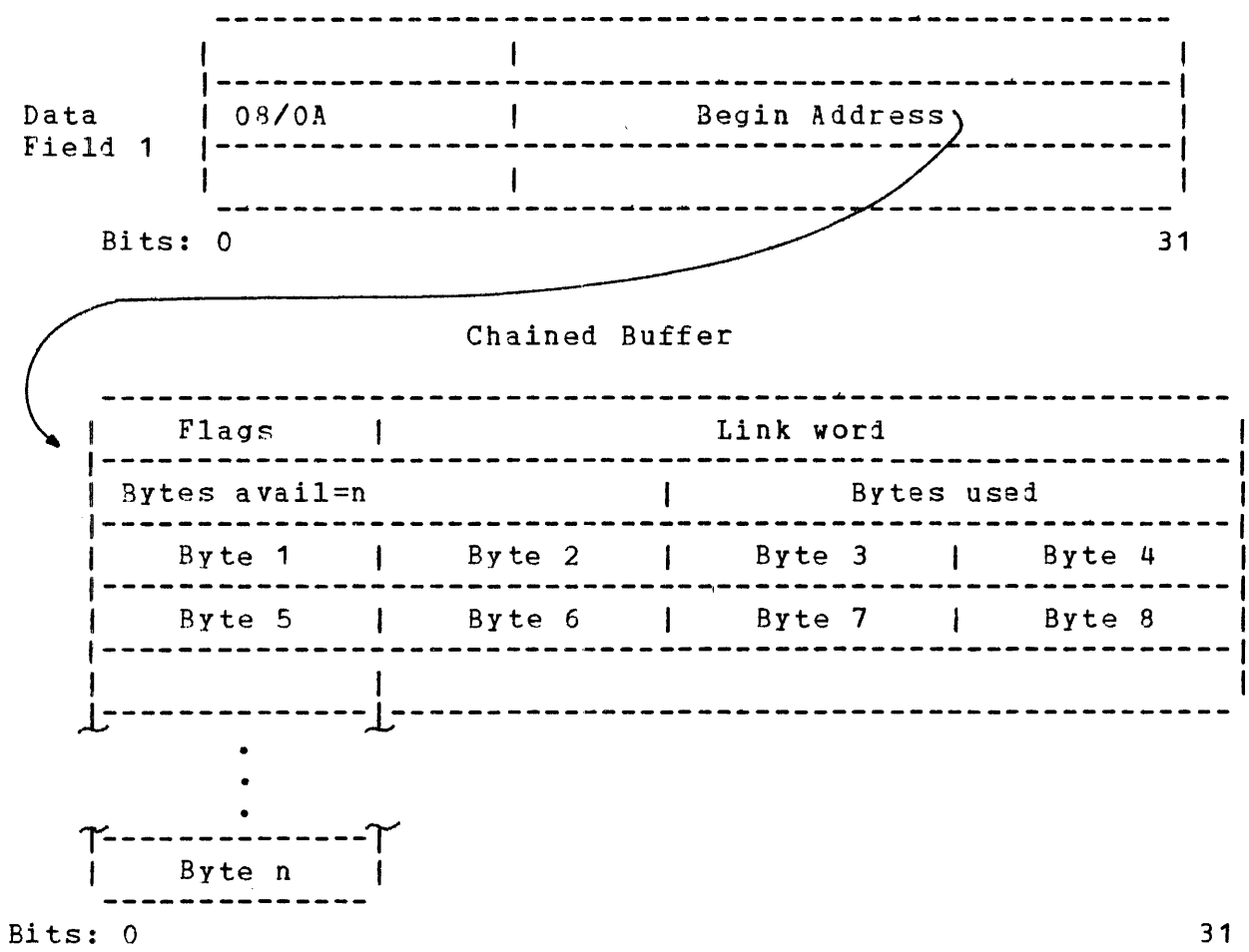


Figure 3-10 Chained/Queued Buffer Format

The first byte of the link word is used for certain flags indicating conditions or options within the buffer. Chained buffers must be aligned on a fullword boundary. See Figure 3-11.

A task can manipulate the links and data of chained buffers while I/O activity is in progress. Bits 0-7 of the link word (the flag byte) are used to coordinate driver and u-tasks as shown in Figure 3-11.

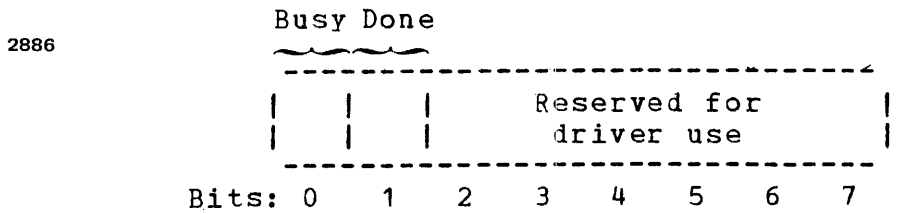


Figure 3-11 Chained/Queued Buffer Link Word Flag Byte

BIT 0	BIT 1	
0	0	A buffer is available for use by the driver. The link word contains a valid address or zero.
1	0	The driver is currently using this buffer for I/O. The user task must not change: - Data - Size values - Link word - Flags
1	1	The driver has finished using the buffer. The driver will not begin to use this buffer again should it reoccur in the chain. A u-task can change any value, and the bytes used field reflects actual transfer.
0	1	Invalid setting. The driver treats it as if both busy and done bits were set to one.

3.6.5 Queued Buffers (Data Code X'0A')

Queued buffers are specified by two data fields. Each data field contains the address of a standard Perkin-Elmer circular list. The first list (list 1) specifies a queue of buffers from which the basic communications subsystem removes buffers for input/output operations. The second list (list 2) specifies a queue of buffers that the basic communications subsystem returns to the applications program following I/O activity. List 1 may coincide with list 2. Figure 3-14 illustrates the standard Perkin-Elmer circular list. The basic communications subsystem removes buffers from the top of list 1 by executing a remove from top of list (RTL) instruction and returns buffers to the bottom of list 2 by executing an add to bottom of list (ABL) instruction.

The format of each individual queued buffer, whose address is contained in the list, is identical to the format of a chained buffer. As with other buffer types, the circular list definition, and all buffers included within the list, must be in the same logical segment. Restrictions on modification of the buffer control fields during I/O apply equally to queued buffers and chained buffers.

When an I/O buffer is removed from list 1 by the basic communications subsystem, the link address field is cleared to prevent error verification ambiguity, and the address of the

buffer is maintained solely within driver control storage. The buffer is, in effect, not available to the applications program during I/O.

The busy and done bits within the flag byte are used analogous to chained buffers. When I/O is complete, the buffer is returned to the bottom of list 2. Simultaneously with I/O operation, the applications task can add new I/O buffers to the bottom of list 1 or remove completed buffers from the top of list 2. Only list processing instructions (RTL, RBL, ATL, ABL) can be used by the applicatons task to modify a circular list. Any other attempt to modify circular list control fields could result in a loss of control.

Should the communications subsystem attempt to return a buffer to list 2 and not be able to complete the action because the list is full, a queue overflow (X'24') error termination results. The addresses of any buffers currently being used for I/O are then chained to the bottom buffer in list 2 to return them to the applications task. As the list address field is initialized to zero at the start of I/O, a nonzero link field should be checked by the applications task to detect buffers returned due to a queue overflow error condition.

The buffer trap mechanism is available for queued buffers. To conserve processor time, this mechanism is generated only when a buffer is added to a previously empty list 2, indicated by the status returned by the last RTL or RBL. This technique requires an application program to process all buffers in list 2 whenever a trap interrupt occurs. See Figure 3-12.

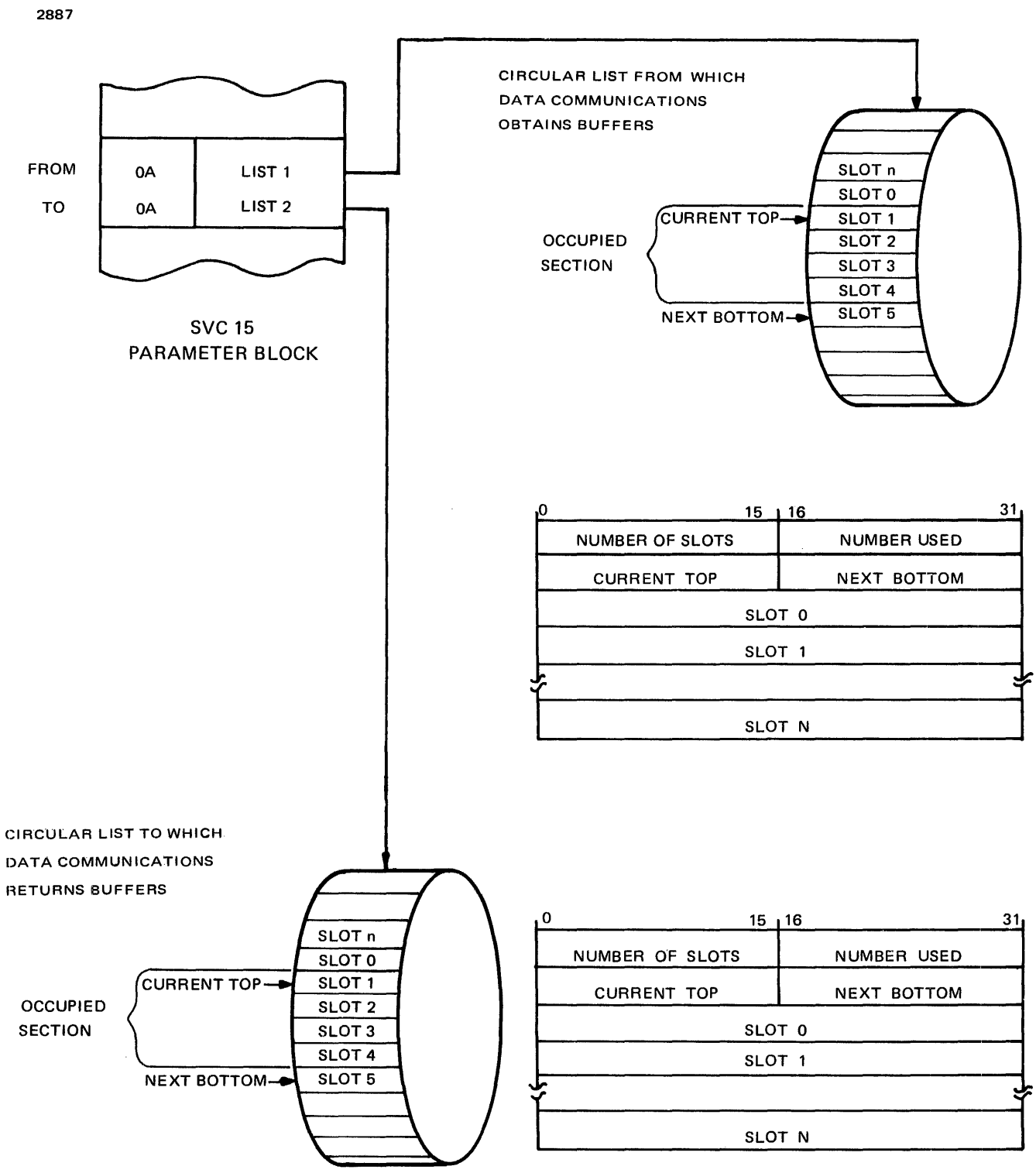


Figure 3-12 Conceptual Circular List and Format

The first two fullwords contain the list parameters. Immediately following the parameter block is the list itself. The first fullword in the list is designated slot 0. The remaining slots are designated 1, 2, 3, etc., up to a maximum slot number equal to the number in the list minus one. An absolute maximum of 65,535 fullword slots can be specified. (Maximum slot designation is equal to X'FFFE'.)

The first parameter halfword indicates the number of slots (fullwords) in the entire list. The second parameter halfword indicates the current number of slots being used. When this halfword equals zero, the list is empty. When this halfword equals the number of slots in the list, the list is full. Once initialized, this halfword is automatically maintained. It is incremented when elements are added to the list and decremented when elements are removed.

The third and fourth halfwords of the list parameter block specify the current top-of-the-list and the next bottom-of-the-list, respectively. These pointers are also automatically updated.

3.6.6 Traps

Two things are required to perform a trap when using the task queue service method:

1. In the TSW of the user-dedicated locations (UDL), the SVC queue entry enable bit must be set.

2. In the SVC 15 function code and in the DCW, the appropriate bits must be enabled for the particular trap desired. See Figure 3-2 and Figure 3-4.

When a condition causes a trap, the current TSW (status and location) is saved in the appropriate area of the UDL. A new TSW (status location counter) is loaded from the appropriate area in the UDL. The new TSW controls the traps or task queue entries to be allowed during the execution of the trap service routine. The trap routine must save general and floating point registers as necessary prior to servicing the trap. An SVC 9 (load TSW) is used to load the saved (old) TSW, returning control to the normal execution sequence.

The SVC 15 parameter block with the trap bits enabled can be added to the user task queue, causing a trap. All traps are transmitted to the calling task before execution of the command. The format of the items added to the task queue is shown in Figure 3-13.

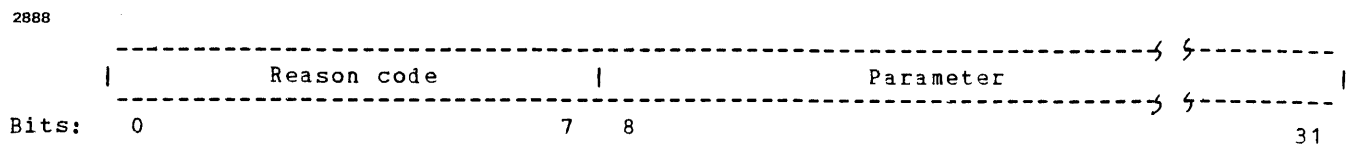


Figure 3-13 Task Trap Format

The code indicates the reason the entry was placed on the queue. The SVC 15 reason code for each of the four kinds of traps is shown in Table 3-3. The parameter entered on the task queue is the address of the SVC 15 parameter block in all cases.

TABLE 3-3 SUPERVISOR CALL 15 (SVC 15) TRAPS

SVC 15 TRAPS	REASON CODE	MEANING
Command trap	X'0A'	This trap is set by a bit in the DCW and a bit in the function code. It is generated to the calling task before execution of the command.
Buffer trap	X'0B'	This trap is set by a bit in the DCW and enabled by the function code. It is generated when the first character is transferred and after each buffer is transmitted or filled.
Termination trap	X'0C'	This trap must have a bit set in the function code and in the SVC 15 queue entry enable in the PSW. It is generated when the driver terminates.
Halt I/O trap	X'0D'	This trap is generated in the place of the X'0C' trap if I/O is terminated as a result of a HALT I/O call.

3.7 HOW TO USE THE ASYNCHRONOUS LINE DRIVER

The asychronous line driver can be used to build a data file by working interactively with an editing VDU. The user begins by providing the information specified by the format appearing on the screen. When the user enters the data required (refer to Appendix C for a sample printout), the interactive process begins. The system continues presenting formats on the screen for user response. This interactive process continues until it is terminated by the user.

CHAPTER 4
TELETYPE (TTY)/VIDEO DISPLAY UNIT (VDU) TERMINAL MANAGER

4.1 INTRODUCTION

This chapter describes the OS/32 Basic Data Communications TTY/VDU terminal manager (INITMASY), which requires the asynchronous line driver.

The TTY/VDU terminal manager is a nonbuffered terminal manager that supports remote asynchronous teletypes, non-editing VDUs, graphic displays, or Carousel 300 terminals in a user program compatible with the local TTY/VDU driver. It provides a device independent supervisor call 1 (SVC 1) access capability featuring data optimization on output and automatic special-character processing on input. It processes the extended options format, connects and disconnects, and is supported by high-level languages.

The TTY/VDU terminal manager occupies approximately 950 bytes; and the DCB/CCB, 400 bytes.

In addition, the terminal manager requires support from the asynchronous SVC 15 line driver and the system support module. As no data buffering is performed, a line control block (LCB) is not required.

A terminal manager has the logic to initiate, maintain, and terminate transmission to a logical device called a terminal. This level of support allows:

- A user program to access local or remote peripherals without recompilation
- User access to remote facilities without regard to the line protocols,codes, and functions of those facilities.
- A user application to be supported by a standard terminal manager without implementing special purpose software

4.2 TERMINALS

The TTY/VDU terminal manager and hardware interfaces support the following remote terminals and any other terminal that meets EIA RS232C interface specifications:

- ASR-33 TTY keyboard/printer
- ASR-35 TTY keyboard/printer
- Nonediting VDU
- Graphic display terminal
- Carousel 300 terminal
- Carousel 300 terminal with electronic format control
- Models 550 and 550/B nonediting terminals
- Model 1100 nonediting terminal

This is a nonbuffered terminal manager that supports the ASCII character set only.

The extended device code (DCB.XDCD) must be initialized at system generation (sysgen) time. A description of the communications subsystem extended device code is shown in Figure 4-1.

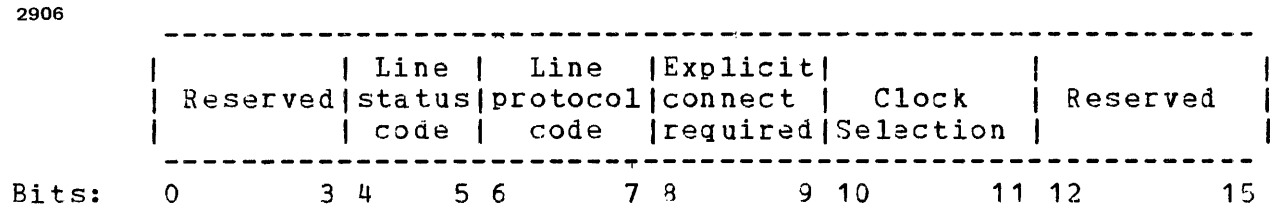


Figure 4-1 Extended Device Code Halfword

LINE CODE	DECIMAL VALUE
Line Status	
Direct connection	0
Leased line	1024
Dial in/manual dial	2048

LINE CODE	DECIMAL VALUE
Line Protocol	
Half duplex 4-wire	0
Half duplex 2-wire	768
Simplex read	256
Simplex write	512
Explicit connect required	128
Clock Selection	
Clock A	0
Clock B	16
Clock C	32
Clock D	48

4.3 SUPPORTED ATTRIBUTES

This terminal manager supports read, write, halt I/O, wait, unconditional proceed, image, variable record lengths up to 72 characters (M33), 74 characters (graphic display terminal), 80 characters (M35 or nonediting VDU), and 132 characters (Carousel 300). All other I/O transfer requests are returned as errors. All command requests are ignored.

4.4 SUPPORTED FUNCTIONS

The OS/32 Basic Data Communications TTY/VDU Terminal Manager supports the functions of read ASCII, write ASCII, and read or write image. For more information of device independent functions (SVC 1), see the OS/32 Supervisor Call (SVC) Programmer Reference Manual.

Read ASCII	Data read is masked to 7-bit ASCII. Data is read until the buffer is full or a carriage return is found, whichever occurs first. Upon termination, a carriage return/line feed sequence is sent to the printer. Typing the character # causes the line input to be ignored, an LF/CR/CR sequence to be output, and the read operation to be restarted. Typing the character (shift-O), BS (ASCII backspace), or cursor left causes the previous character entered to be ignored.
Write ASCII	The buffer is scanned to eliminate trailing blanks. Data is output until the buffer is exhausted or until a carriage return is found in the data stream. A line feed is automatically appended to the deleted CR. If no CR is detected, an LF/CR/CR sequence is output.
Read or Write Image	None of the previous formatting actions occur. The amount of data requested is typed out or read in, without masking, to 7-bit

ASCII, eliminating trailing blanks, checking for # characters, or detecting or appending carriage returns or line feeds. On image read, however, an ASCII CR is detected as an end of line terminating control character.

4.5 THE PERKIN-ELMER CAROUSEL 300 TERMINAL

The TTY/VDU terminal manager supports the Perkin-Elmer Carousel 300 terminal with or without electronic format control. The Carousel 300 is a versatile, high quality, 30-character per second computer terminal designed for timesharing, data communications, and special applications requiring the availability of a local or remote terminal. The terminal consists of the carousel serial impact printing mechanism, a 76-key alphanumeric keyboard with a 10-key numeric pad, a control panel, a power supply, and a molded cover case. Interfacing to the communications line or local processor is accomplished via the standard RS232C interface. Refer to the Perkin-Elmer Carousel 300 Programming Manual. Other features include:

- 132-character print line at 10 characters per inch
- 15-inch forms width capability
- 128-character line buffer
- 32-character keyboard buffer to increase system throughput
- Independently addressable horizontal and vertical print positions in increments of 1/10 inch and 1/48 inch, respectively
- A peak speed of 40 cps

The TTY/VDU terminal manager passes through to the terminal any of the following escape character sequences requested by SVC 1:

ESC 0 — Set left margin - Causes the current print position to become the new left margin. Valid for print positions 0 to 127. A margin can be redefined by backspacing, spacing, or performing an addressable horizontal tab to the desired new margin position, and then performing an ESC 0. Addressable horizontal tabulation codes to the left of the margin are valid and do not affect margin definition.

ESC 1 — Horizontal tab set - Electronically sets the current print position as a tab stop

ESC 2	Horizontal tab clear - Electronically clears the current print position as a tab stop
ESC 3	Clear all tabs - Clears all previously set tabs
FSC 4	Ribbon up - Raises ribbon to printing position with a single-color ribbon or restores ribbon to black printing position from red when using a two-color ribbon
ESC 5	Ribbon down - Lowers ribbon to stencil position with a single-color ribbon or to red ribbon position when using a two-color ribbon
ESC 7	Reverse line feed - Moves paper 1/6 inch to the previous line
ESC P	Inhibit escapement - Causes print carriage to remain at last printed column position
ESC P	Restore escapement - Restores printing pitch to 10 characters per inch
CONTROL L	Top of form - The control logic monitors the passage of a standard 11-inch, 66-line form through the printer. The form can be skipped in whole or in part when the code is received. Movement stops at the end of the 11-inch form and corresponds to the top of the next form. The initial setting of the top-of-form position is automatic and only requires the first manual positioning of the form with the platen knob. If power is turned off, the top of the form must be redefined when power is restored.
Addressable horizontal	The control logic allows direct forward and reverse tabulation of the print carriage to any of the 132 printing positions. Each tabulation position is reached by inputting a 3-code sequence from the data source. See Table 4-1 for coding.

NOTE

The firmware presets every eighth column as a tab stop when power is first turned on.

Addressable vertical tabulation	The control logic allows the direct addressing of 127 vertical tabulation positions. Forward, up and/or reverse, down, movement of the paper or form can be achieved by entering the appropriate codes as shown in Table 4-2.

TABLE 4-1 ADDRESSABLE HORIZONTAL TABULATION CODING CHART

2907

ESC									
L or l				M or m				N or n	
Sp 0	0 16	@ 32	P 48	Sp 64	0 80	@ 96	P 112	Sp 128	
! 1	1 17	A 33	Q 49	! 65	1 81	A 97	Q 113	! 129	
" 2	2 18	B 34	R 50	" 66	2 82	B 98	R 114	" 130	
# 3	3 19	C 35	S 51	# 67	3 83	C 99	S 115	# 131	
$ 4	4 20	D 36	T 52	$ 68	4 84	D 100	T 116	$ 132	
% 5	5 21	E 37	U 53	% 69	5 85	E 101	U 117		
& 6	6 22	F 38	V 54	& 70	6 86	F 102	V 118		
/ 7	7 23	G 39	W 55	/ 71	7 87	G 103	W 119		
(8	8 24	H 40	X 56	(72	8 88	H 104	X 120		
) 9	9 25	I 41	Y 57	) 73	9 89	I 105	Y 121		
* 10	: 26	J 42	Z 58	* 74	: 90	J 106	Z 122		
+ 11	; 27	K 43	[59	+ 75	; 91	K 107	[123		
, 12	< 28	L 44	\ 60	, 76	< 92	L 108	\ 124		
— 13	= 29	M 45	] 61	— 77	= 93	M 109	] 125		
. 14	> 30	N 46	^ 62	. 78	> 94	N 110	^ 126		
/ 15	? 31	O 47	— 63	/ 79	? 95	O 111	— 127		

CODE SEQUENCE:

ESC + L,l or M,m or N,n +

note: upper or lower case may be used

ASCII CODE / PRINT POSITION

TABLE 4-2 ADDRESSABLE VERTICAL TABULATION CODING CHART

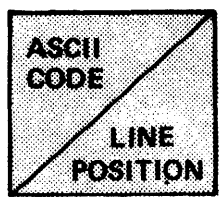
2908

ESC							
! = FWD., # = REV.				" = FWD., $ = REV.			
Sp 0	0 16	@ 32	P 48	Sp 64	0 80	@ 96	P 112
! 1	1 17	A 33	Q 49	! 65	1 81	A 97	Q 113
" 2	2 18	B 34	R 50	" 66	2 82	B 98	R 114
# 3	3 19	C 35	S 51	# 67	3 83	C 99	S 115
$ 4	4 20	D 36	T 52	$ 68	4 84	D 100	T 116
% 5	5 21	E 37	U 53	% 69	5 85	E 101	U 117
& 6	6 22	F 38	V 54	& 70	6 86	F 102	V 118
/ 7	7 23	G 39	W 55	/ 71	7 87	G 103	W 119
(8	8 24	H 40	X 56	(72	8 88	H 104	X 120
) 9	9 25	I 41	Y 57	) 73	9 89	I 105	Y 121
* 10	: 26	J 42	Z 58	* 74	: 90	J 106	Z 122
+ 11	; 27	K 43	[59	+ 75	; 91	K 107	[123
' 12	< 28	L 44	\ 60	' 76	< 92	L 108	\ 124
— 13	= 29	M 45	] 61	— 77	= 93	M 109	] 125
. 14	> 30	N 46	^ 62	. 78	> 94	N 110	^ 126
/ 15	? 31	O 47	— 63	/ 79	? 95	O 111	— 127

CODE SEQUENCE:

ESC + ! or " (for forward)

ESC + # or $ (for reverse)



Vertical line increments

The following codes are used to move the platen in increments of 1/48 inch:

FWD (UP)	REV (DOWN)	NO. INCREMENTS (IN 1/48")
ESC @		0
ESC A	a	1
ESC B	b	2
ESC C	c	3
ESC D-super-script	d-subscript	4
ESC E	e	5
ESC F	f	6
ESC G	g	7

Form length codes

Control logic is set, on application of power, to a value equivalent to 11 inches on application of power, to a value equivalent to 11 inches for top of form. This value can be altered for any form length up to 127 lines by keyboard or u-task input of the codes shown in Table 4-3.

NOTE

The 11-inch top-of-form value is restored only on power down/power up or by entering the proper codes.

TABLE 4-3 FORM LENGTH CODING CHART

2909

ESC							
k				K			
Sp 0	0 16	@ 32	P 48	Sp 64	0 80	@ 96	P 112
! 1	1 17	A 33	Q 49	! 65	1 81	A 97	Q 113
" 2	2 18	B 34	R 50	" 66	2 82	B 98	R 114
# 3	3 19	C 35	S 51	# 67	3 83	C 99	S 115
$ 4	4 20	D 36	T 52	$ 63	4 84	D 100	T 116
% 5	5 21	E 37	U 53	% 69	5 85	E 101	U 117
& 6	6 22	F 38	V 54	& 70	6 86	F 102	V 118
/ 7	7 23	G 39	W 55	/ 71	7 87	G 103	W 119
(8	8 24	H 40	X 56	(72	8 88	H 104	X 120
) 9	9 25	I 41	Y 57	) 73	9 89	I 105	Y 121
* 10	: 26	J 42	Z 58	* 74	: 90	J 106	Z 123
+ 11	; 27	K 43	[59	+ 75	; 91	K 107	[123
' 12	< 28	L 44	\ 60	' 76	< 92	L 108	\ 124
– 13	= 29	M 45	] 61	– 77	= 93	M 109	] 125
. 14	> 30	N 46	^ 62	. 78	> 94	N 110	^ 126
/ 15	? 31	O 47	— 63	79	? 95	O 111	— 127

CODE SEQUENCE:

ESC + k or K + ASCII CODE / NO. OF LINES

4.5.1 System Characteristics

The Carousel 300 terminal responds to the input of serial ASCII data. When operating at 150 or 300 baud, each character is composed of a 1-unit start bit, which is always space, followed by seven units of information bits, a 1-unit parity bit, and a 1-unit stop bit, which is always mark.

4.5.1.1 Character Structure

The total character structure consists of 10 units (see Figure 4-2). At 110 baud, the character structure consists of 11 units, the extra unit being assigned to a second stop bit. Characters are transmitted with the low order bit first and the eighth bit (parity bit) last; i.e., serial, bit-by-bit.

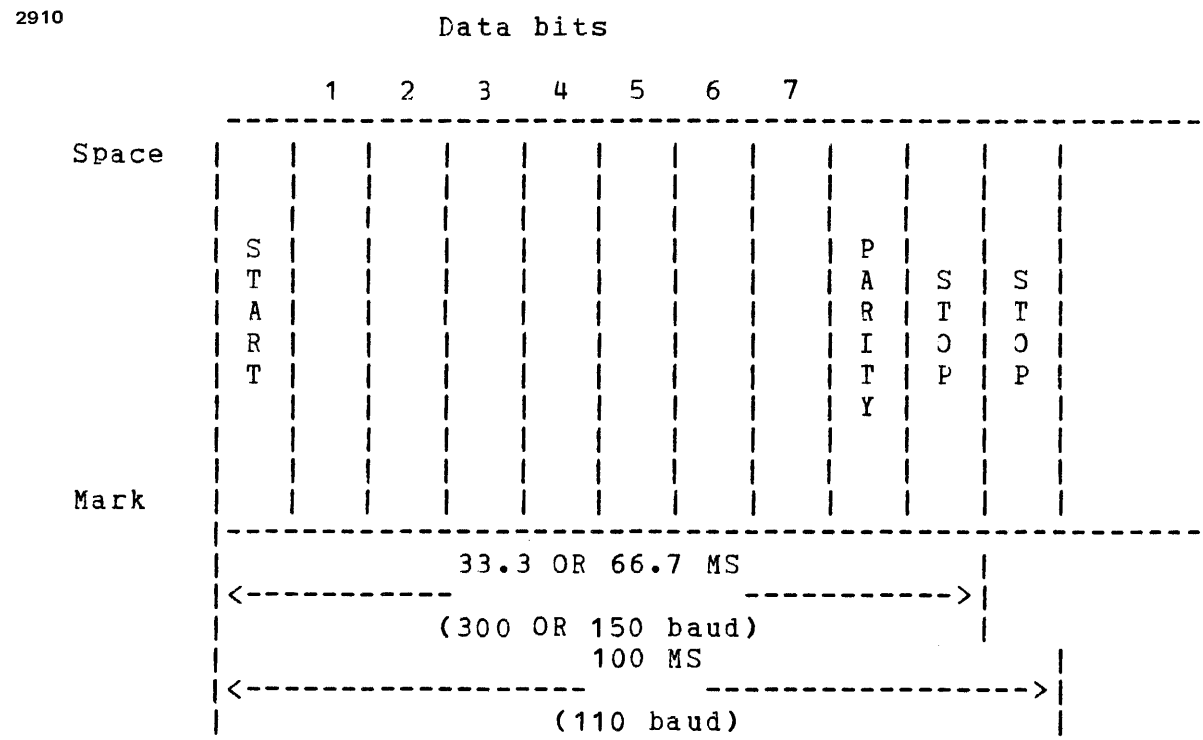


Figure 4-2 Character Structure

4.5.1.2 Modem Connection

External modems, supplied by either the common carrier or the terminal user, are connected to the Carousel 300 internal RS232C interface via an extender cable. This 8-foot cable attaches to the rear of the cover case and terminates in a standard 25-pin male EIA connector.

4.6 PERKIN-ELMER MODEL 1100 VDU

Another device supported by the terminal manager is the Perkin-Elmer Model 1100 VDU. It is upward compatible with the ASR-33 TTY and the Carousel 50, 30, and 35.

The basic unit operates on 115 V, 60 Hz. Available options are 230 V, 50 Hz.

The hardware for the VDU product family is assembled from basic building blocks designed around the Motorola 6800 Micro-Processor and the various peripheral and communication interface chips.

4.6.1 Operational Characteristics

General specifications for the Model 1100 VDU are:

- Screen capacity, characters 1,920
- Characters per line 80
- Number of lines 24
- Tube size (diagonal) 12 inches
- Phosphor P4
- Displayable characters (upper/lower case, numbers, punctuation, control) 128
- Character matrix 9 x 12
- Character generation 7 x 9
- Number of scans 12
- Refresh rate (noninterlaced) 60 hz (50)
- Character code: ASCII (Expanded through multicode sequences)
- Keyboard layout: ASCII (Bit pairing)
- Repeat key rate: 15 CPS (60 Hz) or 12.5 CPS (50 Hz)
- Cursor, blinking-reverse video block cursor with blinking disable strap.
- Single key cursor controls: space, backspace, return, line feed, tab, back-tab.
- Multicode cursor control sequences: up, down, left, right, home, direct cursor addressing.

- Scroll mode (switch selectable): See section 4.6.3.2
- Tab: Tab stops are set every eighth position on initial power-on. Tab stops can be set for positions 1 to 80 on all lines.
- Transparent mode: All data received is stored in display memory and displayed without an initiating control function.
- Audible signal: 2 KHz, 166.7 ms. duration (60 Hz) or 200 ms. duration (50 Hz)

The Model 1100 VDU communications interface features:

- Communication options that are read only during the power-up sequence. Changing switch settings while the unit is powered-on is not recognized by the terminal until the next power-up cycle, or the clear-all key is depressed.
- Asynchronous serial interface conforming to RS232C and CCITT V.24.
- Stop bits - switch selectable: one to two stop bits.
- Baud rates - switch selectable: 75 - 110 - 200 - 300 - 600 - 1200 - 1800 - 2400 - 4800 - 9600 baud.
- Transmit/receive mode: half or full duplex, switch selectable.
- Switch-selectable parity: odd, even, always marking or always spacing. Characters received with parity errors are displayed as an alternate dotted rectangle.

4.6.2 Controls

The Model 1100 VDU supports the following operator control functions:

- Clear all
- Power on/off
- Intensity
- Stop bit select (one/two)
- Baud rate selection
- Mode selection (full/half duplex)

- Parity selection: odd-even-zero-one
- Local (A)
- Auto/LF (A)
- U/C (A)
- Break (M)
- Scroll enable (A)
- New line enable (A)
- Here is (M)

A = Alternate action switch

M = Momentary switch

The Model 1100 VDU supports the following internal straps and switches (on the VDU controller board):

- Disable blinking block cursor
- Selection of display mode through switch on PC board: normal or reverse video
- Multicode character selection by means of 7 straps: default condition is ESC
- Permanent or switched DTR: if strap is installed, the data terminal ready lead to the modem is disabled when terminal is in local mode

4.6.3 Options (All Factory Installed)

The following options are available with the Model 1100 VDU:

- Current loop interface (20 ma), active or passive, strap selectable (for use with CLCM when using OS/32 Basic Data Communications).
- Here-is ROM with a maximum 32-character here-is message.
- Auxiliary serial interface
- Additional RS232C compatible serial interface with baud rate and stop-bit strapping selection.
- Current loop option (20 ma), active or passive, for additional RS232C interface.
- Optional 230 V + 10% 50 Hz.

- Optional numeric pad with keys 0: 9 and . (period) and , (comma).
- Anti-glare screen
- VDU-to-modem cable: connects visual display unit to RS232C compatible modem
- Optional underline cursor (block standard).

4.6.4 Principles of Operation

The Model 1100 VDU is controlled by data received over the communications line and from the keyboard.

In FDX or HDX mode, data characters received over the line are stored in display memory and displayed on the screen. Control characters (X'00' - X'1F') and DEL (X'7F') are not stored unless the display is in transparent mode or the character is preceded by a multicode character. ASCII control characters and data initiated at the keyboard are transmitted over the line in HDX or FDX mode. In HDX or local mode, the same actions take place when the key is depressed as occur when the character is received over the line. In FDX mode, the other end device must retransmit characters received from the terminal (echoplex) if they are to affect the display.

In transparent mode, all control characters stored in memory are displayed.

In normal mode, control characters are not displayed. Single- and multi-character control sequences cause appropriate action to take place, except when the display is in transparent mode. Local control keys do not transmit data, but cause the appropriate actions to take place when depressed. They also enable a condition for alternate action type keys.

4.6.5 Local Control Keys

Here Is	Transmits contents of the answer-back ROM, maximum 32-characters.
Upper Case Only	Lower case alphabetic characters are converted to upper case, plus the following punctuation marks are converted: @ to ', [to {, : to *,] to }, and ^ to ~. Conversion takes place when a character enters from the keyboard or line; lower case characters already in memory are not converted.

AUTO/LF	A CR automatically advances the cursor to the first position of the next line.
New Line Enable	Characters beyond position 80 of any line continue to be displayed, starting at the first position of the next line.
Scroll Enable	When LF characters are entered with the cursor in line 24, the cursor remains in line 24. Line 1 is cleared, and the contents of lines 2 through 24 are moved to lines 1 through 23.
Local	Only data entered via the keyboard is displayed. Data cannot be received or transmitted via the communications line.
Repeat	Each character on the keyboard is repeated at the rate of 15 characters per second (60 Hz) or 12.5 characters per second (50 Hz).
Clear All	Simulates a powerup sequence: display memory is cleared to nulls, cursor is moved to home, tabs are set to every eighth position, communication options are read, and all interfaces initialized.
Print	Initiates a local printout. Data on the screen is transmitted to the auxiliary serial interface (if present) from the current cursor position to the end of the screen.
Back Tab	Moves cursor to the first preceding tab stop. If the cursor is to the left of the first tab position, the cursor moves to column 1. If no tabs are set, the back tab moves the cursor to position 1.

4.6.6 Single-Character ASCII Functions

Line Feed	Moves the cursor down one line, except when the VDU is in the scroll enable mode and the cursor is in line 24. The display rolls up one line. If the terminal is not in scroll enable mode, and the cursor is in line 24, the cursor wraps around to line 1.
Return	Moves the cursor to position 1 of the current line. If auto/LF is enabled, a line feed function is performed in addition to the return function.

Space and All Other Displayable Characters	Writes a character at the current position and moves the cursor one position to the right. The cursor remains at position 80 unless new line enable is selected.
Backspace (BS)	Moves the cursor to the left one position. The cursor remains at position 1 for any additional BS unless new line enable is selected. Then BS wraps around to position 80 of the previous line. The cursor does not move beyond Home (first character, first line).
Tab (HT)	Moves the cursor to the next tab position. If new line enable is not set and the cursor is past the last tab position, tab moves the cursor to position 80. If new line enable is selected and the cursor is past the last tab position, tab moves the cursor to the first tab position on the next line. If there are no tabs set, tab moves the cursor to position 80; if new line enable is also set, the cursor moves to line 24.
Break	Causes the communications line to go to a space (break) for as long as the key is depressed.
Multicode	Initiates a special function sequence (multicode character defined by straps). Default is ESC.
Bell	Causes the audible alarm to sound for 166.7 ms (60 Hz) or 200 ms (50 Hz).
Form Feed	If scroll is enabled, form feed has the same effect as line feed. If scroll is not enabled, form feed erases the screen and moves the cursor to home.
Enquiry (ENQ)	Causes the contents of the answer-back ROM to be transmitted.
Data Link Escape (DLE)	Ignored unless followed by STX, in which case it causes the display to enter transparent mode. In transparent mode, all characters including control are stored in memory and displayed. New line is enabled; scroll is disabled. If DLE is received in transparent mode, the

next character is examined. If ETX is received, the display is returned to normal mode; otherwise a received character is stored and displayed.

4.6.7 Multicode Sequences

Multicode-A — Cursor Up (↑)

Moves the cursor up one line. If in line 1, the cursor wraps around to line 24

Multicode-B — Cursor Down (↓)

Moves the cursor down one line. If in line 24, the cursor wraps around to line 1

Multicode-C — Cursor Right (→)

Moves the cursor one position to the right. When the cursor is moved beyond position 80, the cursor wraps around to position 1 of the following line if the terminal is in new-line-enable mode. If new line is not enabled, the cursor remains in column 80 (line 1 follows line 24).

Multicode-D — Cursor Left (←)

Moves the cursor one position to the left. When the cursor is moved beyond position 1, the cursor wraps around to position 80 on the previous line if the terminal is in new-line-enable mode. If new line is not enabled, the cursor remains in column 1.

Multicode-H — Cursor Home (H)

Moves the cursor to position 1, line 1 (home)

Multicode-X — Direct Cursor Address - Line Position

Moves the cursor vertically to any line as specified by the character following X, as shown in Table 4-4

Multicode-Y	Direct Cursor Address - Character Position Moves the cursor horizontally to any position on a line. The character following Y specifies the horizontal character position, as specified in Table 4-4.
Multicode-Z	Read Cursor Address Causes the terminal to transmit the line and character position of the cursor, as specified in Table 4-4
Multicode-1	Set Tab A tab stop is set at the cursor position. Tab stops can be set in any of the 24 lines of the display and are effective for all lines.
Multicode-2	Clear Tab The tab stop at the cursor location is cleared.
Multicode-3	Clear All Tabs All tab stops are cleared.
Multicode-K	Clear All Clears the display memory to nulls. All tab stops are cleared.
Multicode-I	Clear Line Clears line (reset to nulls) starting with the present cursor position to the end of line
Multicode-J	Clear Display Clears the display (reset to nulls). Except for tab stops, the entire display is cleared from the present cursor location to the end of the page.

TABLE 4-4 CURSOR ADDRESSING

ASCII	HEX	LINE/ COLUMN	ASCII	HEX	LINE/ COLUMN
(SP)	20	1	H	48	41
!	21	2	I	49	42
"	22	3	J	4A	43
#	23	4	K	4B	44
$	24	5	L	4C	45
%	25	6	M	4D	46
&	26	7	N	4E	47
'	27	8	O	4F	48
(	28	9	P	50	49
)	29	10	Q	51	50
*	2A	11	R	52	51
+	2B	12	S	53	52
`	2C	13	T	54	53
-	2D	14	U	55	54
.	2E	15	V	56	55
/	2F	16	W	57	56
0	30	17	X	58	57
1	31	18	Y	59	58
2	32	19	Z	5A	59
3	33	20	[	5B	60
4	34	21	/	5C	61
5	35	22	]	5D	62
6	36	23	^	5E	63
7	37	24	-	5F	64
8	38	25	`	60	65
9	39	26	a	61	66
:	3A	27	b	62	67
;	3B	28	c	63	68
<	3C	29	d	64	69
=	3D	30	e	65	70
>	3E	31	f	66	71
?	3F	32	g	67	72
@	40	33	h	68	73
A	41	34	i	69	74
B	42	35	j	6A	75
C	43	36	k	6B	76
D	44	37	l	6C	77
E	45	38	m	6D	78
F	46	39	n	6E	79
G	47	40	o	6F	80

4.7 ENCODED ERROR MESSAGES

Encoded messages are displayed indicating the status of an I/O operation occurring under the control of the TTY/VDU terminal manager. Table 4-5 lists these messages.

TABLE 4-5 ENCODED ERRORS AND DEFINITIONS FOR TELETYPE (TTY) TERMINAL MANAGER

STATUS CODE (HEX)	MEANING
0000	No errors
8402	Line delete caused termination during read
8203	Break detected during write
8204	Break detected during read
8205	Terminated by data error (see parity bits)
8208	Framing or stop-bit error
8409	Reverse channel error
200A	Lost carrier on read
200B	Lost clear-to-send on write
200C	Data set not ready
840D	Device unavailable; adapter not present
820E	Character overflow
840F	Ring status detected during data transfer
8410	Busy and/or done bits in chained buffers bad; may indicate priority too low
8411	Number of commands executed greater than 255
8412	Task queue full, invalid or nonexistent
8413	Buffer-management-routine error; may indicate priority too low
8282	Timeout
8281	Halt I/O request aborted I/O
8418	Invalid command or modifier
8419	Memory fault in referencing data
841A	Memory fault in referencing buffer
811B	Logical unit illegal
841C	Illogical device status
A01D	Power failure
841E	Illegal software condition
841F	Illegal translation table
8225	Timeout during connect sequence
8426	ESC, R not received on RQS

The first byte of each status code listed in Table 4-5 refers to the device independent status of the error. These codes are defined as follows:

CODE	DEFINITION
C0	Illegal function
AC	Device unavailable; sign off user. If switched line, reissue call.
90	End of medium
88	End of file
84	Unrecoverable error; report to operator
81	Illegal or unassigned lu
82	Parity or recoverable error; reissue the call
20	Device unavailable or parity error sent to recovery routine

CHAPTER 5
MODELS 1200/1250/1251 EDITING VIDEO DISPLAY UNIT (VDU) TERMINAL MANAGERS

5.1 INTRODUCTION

This chapter describes the functions of the Models 1200/1250/1251 Editing VDU Terminal Manager. This terminal manager operates within OS/32 (R06 software release and higher). The Models 1200/1250/1251 terminal managers provide the user with the full range of editing features and programmable operations that are available with these models. These features include the multidrop and light pen capabilities of the Models 1250/1251.

User programs can communicate with these VDUs via supervisor call 1 (SVC 1). The terminal manager supports local and remote point-to-point communications as well as standard utility programs.

Because the Models 1200/1250/1251 terminal managers support all features of INITMASY, the terminal manager can replace INITMASY when Models 1200 or 1250/1251 VDUs are being added to existing teletype (TTY) configured systems.

5.2 FUNCTIONAL DESCRIPTION

5.2.1 Device Assignment

The Models 1200/1250/1251 Editing VDUs can be assigned to a user task (u-task) logical unit (lu) by one of two methods:

1. The operator ASSIGN command, or
2. The SVC 7 assign function.

Refer to the OS/32 Supervisor Call (SVC) Reference Manual and the OS/32 Basic Data Communications Reference Manual for detailed assignment procedures.

To assign the Models 1250/1251 VDUs to an lu in a multidrop environment, use the GENERATE command or macro in the user program. The GENERATE command or macro generates system structures associated with the terminal name; the ELIMINATE command or macro is used to eliminate them.

5.2.1.1 GENERATE Command

The GENERATE command builds system structures within system space for the Models 1250/1251 VDUs.

Format:

```
GENERATE tn,cnum
```

Parameters:

tn	is a file descriptor (fd) specifying the name of the terminal. This standard fd, defined by the designer/operator, has the format of voln:filename.ext, where voln is the sysgened device mnemonic of the communications line, and filename.ext is the name of the terminal.
cnum	is a decimal number from 0 through 127 (except for codes listed in Appendix B) specifying the terminal address that corresponds to a poll/select address for the terminal. A system operator obtains a value for cnum from the system designer.

Example:

```
LOAD EGU.TSK
TA EGU
ST ,COM=CON:,LOG=PRT2:,ERR=GENERR:EGU
GENERATE BSC0:DISPLAY.001,0
GENERATE BSC0:DISPLAY.002,1
GENERATE BSC1:DISPLAY.001,2
GENERATE BSC1:PRINTER.002,3
END
```

5.2.1.2 GENERATE Macro

The GENERATE macro, issued from within a task, builds system structures within system space for the Models 1250/1251 VDUs.

Format:

NAME	OPERATION	OPERAND
symbol	GENFRATE	xadr ,XPCB={xadrx / reg} ,FD='fd' ,CNUM={absexp / reg}

Parameters:

xadr — is the symbolic or indexed address of a previously constructed parameter control block (PCB). If this parameter is omitted, the parameter control block is automatically built, and its address is placed in register 14.

XPCB= — xadrx is the symbolic or indexed address of an extended parameter control block. If this parameter and the xadr parameter are omitted, the extended PCB is automatically built and linked to the previously constructed PCB.

(reg) is a register expression specifying a decimal value from 0 through 15 that indicates the register containing the address of an extended PCB.

FD= — 'fd' is the unpacked file descriptor of the terminal for which the PCB is being generated. This fd corresponds to the terminal name (tn) in the GENERATE command. If this parameter is omitted, the fd must have been previously specified in the FMPCB macro.

CNUM= — absexp is an absolute byte expression specifying the decimal value from 0 through 127 that designates the terminal poll/select address. If this parameter is omitted, it must have been previously specified in the XFMPCB macro. See Section 5.2.1.3.

Functional Details:

Before execution of the GENERATE macro, the PCB must contain a value for FD, and the extended portion of this PCB built by the XFMPCB macro must contain a value for CNUM. If the PCB does not contain these values, they must be specified with the GENERATE macro.

Example:

```
GENERATE FM.PCB,CNUM=(R2)
```

5.2.1.3 XFMPCB Macro

The XFMPCB macro constructs an extended portion for the file management PCB.

Format:

NAME	OPERATION	OPERAND
symbol	XFMPCB	,CNUM={absexp / (reg)}

Parameter:

CNUM= absexp is an absolute byte expression specifying the decimal value from 0 through 255 that corresponds to the ASCII character code for the terminal poll/select address. If this parameter is omitted, it must have been previously specified in the GENERATE macro.

(reg) is a register expression specifying a decimal value from 0 through 15 that indicates the register containing the poll/select address.

Functional Details:

If symbol is specified, it becomes the symbolic address. Whether or not symbol is specified, the macro places the extended PCB address into register 14.

The extended PCB must contain a value for CNUM before the GENERATE macro can be expanded. If omitted from the XFMPCB macro, CNUM must be included with the GENERATE macro.

5.2.1.4 ELIMINATE Command

The ELIMINATE command eliminates system structures from system space that were previously built for the Models 1250/1251 VDUs.

Format:

```
ELIMINATE tn
```

Parameter:

tn	is an fd specifying the name of the terminal to be eliminated. See the GENERATE command for a detailed description of this parameter.

Example:

```
LOAD EGU.TAK
TA EGU
ST ,LOG=PRT2:,ERR=GENERR.EGU,COM=CON:
ELIMINATE BSC2:DISPLAY.001
END
```

5.2.1.5 ELIMINATE Macro

The ELIMINATE macro eliminates system structures within system space previously generated for the Models 1250/1251 VDUs.

Format:

NAME	OPERATION	OPERAND
symbol	ELIMINATE	xadr
		,FD='fd'

Operands:

xadr	is the symbolic or indexed address of a previously constructed PCB. If this parameter is omitted, the PCB is automatically built, and its address is placed in register 14.
FD=	'fd' is the unpacked file descriptor of the terminal for which the PCB is being generated. This fd corresponds to the terminal name (tn) in the GENERATE command. If this parameter is omitted, the fd must have been previously specified in the FMPCB macro.

Functional Details:

Before execution of the ELIMINATE macro, the PCB must contain a value for FD, and the logical unit (lu) to which the file descriptor is assigned must be closed.

Example:

```
ELIMINATE    FM.PCB1,FD-MDCB.DISPLAY.002
```

5.2.2 SVC 1 Interface

SVC 1 is used to initiate I/O for both local and remote devices. See the OS/32 Supervisor Call (SVC) Reference Manual for a detailed description of the SVC 1 parameter block.

5.2.3 SVC 1 Extended Options

The extended options field of the SVC 1 parameter block specifies to the OS/32 data communications subsystem the I/O functions to be supported for the Models 1200/1250/1251 VDUs. When this field is used, bit 7 of the SVC 1 function code and bit 6 of the task option word located in the task control block (TCB) are set to 1.

Figure 5-1 illustrates the fullword format of the extended options field of the SVC 1 parameter block.

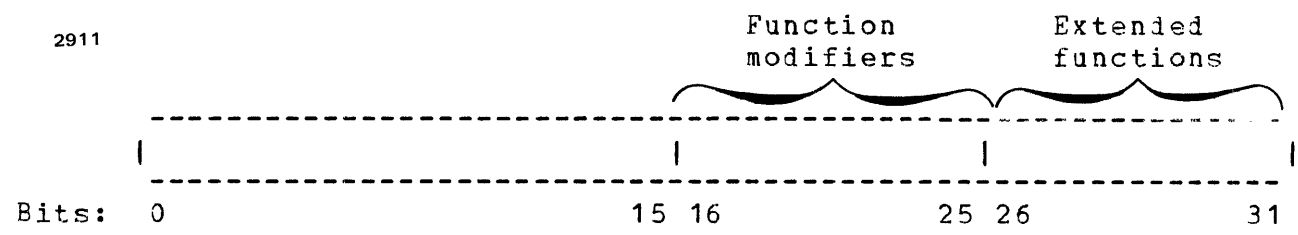


Figure 5-1 Extended Options Fullword Format

Bits 0 through 15 are for general use in both local and remote communications. Bits 16 through 25 are used as function modifiers, while bits 26 through 31 can be used to specify up to 64 device dependent I/O functions, 41 of which are used in Models 1200 and 1250/1251 communications. Available extended functions and function modifiers for the SVC 1 extended options fullword are listed and described in Table 5-1.

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS

BIT SETTING (HEX)	OPTION	MEANING
Y'8000 0000'	Connect (CON)	Terminal manager answers a telephone ring on a dial-in line during a read or write line initialization sequence.
Y'4000 0000'	Disconnect (DCT)	Terminal manager disconnects from a switched line following final data transfer.
Y'0000 0000'	Image (IMG)	Data being transmitted is in image mode and is not formatted.
Y'2000 0000'	Format (FMT)	Terminal manager performs normal record buffering, inserts or deletes line control characters, and recognizes appropriate data format control characters on transmitted data.
Y'0080 0000'	Vertical forms control (VFC)	Requests VFC option for an ASCII I/O operation.
Y'0000 8000'	Nontermination on CR (NTC)	Read will terminate only on detection of end of text.
Y'0000 4000'	Enable uppercase only (EUO)	Translation of lower to upper case is enabled.
Y'0000 2000'	Clear screen (CLS)	Clears screen before I/O
Y'0000 1000'	Unlock keyboard (UKB)	Unlocks the keyboard after I/O

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS (Continued)

BIT SETTING (HEX)	OPTION	MEANING
Y'0000 0800'	Lock keyboard (LKB)	Locks the keyboard after I/O
Y'0000 0400'	Write blinking characters (WBL)	Writes blinking characters
Y'0000 0200'	Write protected characters (WPC)	Writes protected characters
Y'0000 0100'	Write reverse characters (WRV)	Writes reverse video characters
Y'0000 0080'	Write low-intensity characters (WLI)	Writes low-intensity characters
Y'0000 0000'	Conversational mode (CNV)	Executes the read/write as conversational
Y'0000 0001'	Request to send (RQS)	Issues the RQS sequence from a terminal and places an attention identifier character (AID) in the user buffer, usually followed by read immediate
Y'0000 0002'	Read immediate unprotected (RIU)	Reads sequence used in conjunction with the RQS sequence to read unprotected fields from a formatted screen
Y'0000 0003'	Read immediate modified (RIM)	Reads sequence used in conjunction with RQS sequence to read modified data fields from a formatted screen only
Y'0000 0004'	Read immediate all (RIA)	Reads sequence used in conjunction with the RQS to read an entire screen. Data transfer is based on setting of full/partial screen mode.

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS (Continued)

BIT SETTING (HEX)	OPTION	MEANING
Y'0000 0005'	Send immediate unprotected (SIU)	Reads unprotected data fields in send immediate mode; i.e., whenever the send key is pressed
Y'0000 0006'	Send immediate modified (SIM)	Reads modified data fields from a formatted screen whenever the send key is pressed
Y'0000 0007'	Send immediate all (SIA)	Reads the entire screen of data whenever the send key is pressed
Y'0000 0008'	Write buffer edit (WBE)	Indicates the call is a write in edit mode; i.e., depending on which function modifier is set, WBE can write blinking characters, protected characters, etc.
Y'0000 0009'	Read cursor address (RCA)	Places cursor address in the user buffer exactly as it is received from Perkin-Elmer Models 1200 or 1250/1251 VDUs
Y'0000 000A'	Read options (ROP)	Reads two option bytes through Models 1200 or 1250/1251 VDUs into the user buffer
Y'0000 000B'	Read status immediate (RSI)	Terminal immediately transmits status byte to host
Y'0000 000C'	Read status when ready (RSR)	Reads status byte through Models 1200 or 1250/1251 VDUs only when print or insert/delete is completed
Y'0000 000D'	Clear screen (CLS)	Clears unprotected areas of the screen only
Y'0000 000E'	Clear memory (CLM)	Clears all memory including protected area

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS (Continued)

BIT SETTING (HEX)	OPTION	MEANING
Y'0000 000F'	Enter transparent mode (ETM)	All subsequent characters, including control codes, are stored in memory and displayed. No control action takes place. A new line is enabled and scroll is disabled, regardless of the local control key setting.
Y'0000 0010'	Enter normal mode (ENM)	Exits from transparent mode. Control characters are neither stored nor displayed.
Y'0000 0011'	Home cursor (HME)	Places cursor in home position
Y'0000 0012'	Set cursor position (SCP)	Cursor position line hexadecimal numbers 00 through 17 and column hexadecimal numbers 00 through 4F are provided in the user buffer.
Y'0000 0013'	Clear field (CLF)	Clears the current field to which the cursor is pointing
Y'0000 0014'	Set tab stops (STS)	Sets tab stop positions in the user buffer. Tabs are effective for all lines.
Y'0000 0015'	Clear all tab stops (CAT)	Clears all tab stops
Y'0000 0016'	Modify data communications default extended options (MOD)	Modifies the encoded value in bits 12-15 of the data communications extended device code DCB.XDCD
Y'0000 0017'	Unlock keyboard (UKB)	Unlocks the keyboard.
Y'0000 0018'	Lock keyboard (LKB)	Locks the keyboard
Y'0000 0019'	Clear modified data tags (CDT)	Clears modified data tags only

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS (Continued)

BIT SETTING (HEX)	OPTION	MEANING
Y'0000 001A'	Stop print option (SPO)	Stops printing after printout currently in progress is finished
Y'0000 001B'	Print screen from home (PHM)	Prints entire screen
Y'0000 001C'	Print screen from cursor (PCR)	Prints screen from current cursor position to the end of screen
Y'0000 001D'	Simulprint form feed control (PFF)	Activates the form feed control for the print page
Y'0000 001E'	Continuous simulprint (PCO)	Simultaneously outputs characters to the Models 1200 or 1250/1251 VDUs and the remote printer
Y'0000 001F'	Page pause (PGP)	Directs Models 1200/1250/1251 VDU terminal managers to send RQS sequence and wait for the user to press the VDU function keys
Y'0000 0020'	Write edit with attribute character generated (WAG)	Sends any character with parity bit set as an attribute character to Models 1200 or 1250/1251 VDUs following an ESC! generated by the terminal manager
Y'0000 0021'	Read all with attribute character truncated (RAT)	Optimizes the data read from Models 1200 or 1250/1251 VDUs during read all by disregarding ESC!
Y'0000 0022'	Send all with attribute character truncated (SAT)	Optimizes the data read from Models 1200 or 1250/1251 VDUs during send all by disregarding ESC!

TABLE 5-1 SVC 1 FUNCTION/FUNCTION MODIFIER OPTIONS (Continued)

BIT SETTING (HEX)	OPTION	MEANING
Y'0000 0023'	Send key override (SKO)	Overrides the previous send key pressed during RQS. This applies to only one data transmission.
Y'0000 0024'	Write status line user area (WSL)	Writes a maximum of 30 characters to the 30-byte user area of the Models 1250/1251 status line
Y'0000 0025'	Send status line user area (SSL)	Reads a 33-byte user area of the status line whenever the Models 1250/1251 SEND key is pressed
Y'0000 0026'	Write user-supplied terminal configuration to EAROM (WUP)	Writes a user-supplied terminal configuration to Models 1250/1251 VDU EAROMS (permanent)
Y'0000 0027'	Write user-supplied terminal configuration to RAM (WUT)	Writes a user-supplied terminal configuration to Models 1250/1251 VDU RAMs (temporary)
Y'0000 0028'	Read terminal configuration (RTC)	Reads 902 configuration bytes from Models 1250/1251 VDUs into the user buffer

NOTE

Extended option bits 3 through 7, 9 through 15, and 25 are reserved and must be set to 0.

SVC 1 extended functions are mutually exclusive; however, an I/O with multiple requests or operations can be performed. The function modifiers are used to expand a function's capability. For example, the write edit function can be expanded to write blinking, write protected, write reverse video, or write low-intensity by a function modifier. However, not all function modifiers apply to each extended function. Table 5-2 lists the possible function/function modifier combinations.

TABLE 5-2 FUNCTION/FUNCTION MODIFIER COMBINATIONS

2912

EXTENDED FUNCTIONS FIELD DEC VALUE	MNEMONICS	SVC 1		COMMUNICATIONS OPTIONS			EXTENDED FUNCTION MODIFIERS								
		RED	WRT	CON	DCT	IMG/ FMT	NTC	EUO	CLS	UKB	LKB	WBL	WPR	WRV	WLI
0	CNV	*	*	*	*	FMT	*	*	*	*	*				
1	ROS	*	X	*	*	FMT				*	*				
2	RIU	*	X	*	*	FMT	*	*		*	*				
3	RIM	*	X	*	*	FMT	*	*		*	*				
4	RIA	*	X	*	*	FMT	*	*		*	*				
5	SIU	*	X	*	*	FMT	*	*		*	*				
6	SIM	*	X	*	*	FMT	*	*		*	*				
7	SIA	*	X	*	*	FMT		*		*	*				
8	WBE	X	*	*	*	FMT	*	*	*	*	*	*	*	*	*
9	RCA	*	X	*	*	FMT				*	*				
10	ROP	*	X	*	*	FMT				*	*				
11	RSI	*	X	*	*	FMT				*	*				
12	RSR	*	X	*	*	FMT				*	*				
13	CLS	X	*	*	*	FMT				*	*				
14	CLM	X	*	*	*	FMT				*	*				
15	ETR	X	*	*	*	FMT				*	*				
16	ENM	X	*	*	*	FMT				*	*				
17	HME	X	*	*	*	FMT			*	*	*				
18	SOP	X	*	*	*	FMT				*	*				
19	CLF	X	*	*	*	FMT				*	*				
20	STS	X	*	*	*	FMT			*	*	*				
21	CAT	X	*	*	*	FMT			*	*	*				
22	MOD	X	*	*	*	FMT									
23	UKB	X	*	*	*	FMT									
24	LKB	X	*	*	*	FMT									
25	CDT	X	*	*	*	FMT				*	*				
26	SPO	X	*	*	*	FMT									
27	PHM	X	*	*	*	FMT			*	*	*				
28	PCR	X	*	*	*	FMT			*	*	*				
29	PFF	X	*	*	*	FMT									
30	PCO	X	*	*	*	FMT									
31	PGP	X	*	*	*	FMT									
32	WAG	X	*	*	*	FMT	*			*	*				
33	RAT	*	X	*	*	FMT	*			*	*				
34	SAT	*	X	*	*	FMT	*			*	*				
35	SKO	X	*	*	*	FMT				*	*				
36	WSL	X	*	*	*	FMT			*	*	*				
37	SSL	*	X	*	*	FMT				*	*				
38	WUP	X	*	*	*	FMT			*	*	*				
39	WUT	X	*	*	*	FMT			*	*	*				
40	RTC	*	X	*	*	FMT				*	*				

* An asterisk indicates a valid function function/modifier combination. X indicates an invalid function function/modifier combination. A blank box indicates the function modifier does not apply to that particular function situation.

5.3 USING THE MODELS 1200/1250/1251 VDU TERMINAL MANAGERS

Programmers can encounter various levels of complexity in using the Models 1200/1250/1251 VDU terminal managers. If the Models 1200 and 1250/1251 VDUs are used as replacements for TTY terminals in an interactive environment, the terminal manager is interfaced with the standard Basic Data Communications TTY/VDU Terminal Manager. However, if the terminals are used to take advantage of all their features, the programmer should be aware of both the hardware features and the logical interface procedures used to access the terminal manager for each model.

The following sections describe the hardware capabilities, the operation modes, software and hardware environment requirements, and general system generation (sysgen) procedures for the Models 1200 and 1250/1251 VDUs.

5.3.1 Models 1250/1251 Terminal Configuration

The Models 1250/1251 VDUs must be configured through a terminal configuration procedure to operate in a multi-terminal monitor (MTM) or integrated transaction controller (ITC) environment. Configuration can be performed in two ways:

1. at the terminal end (user configuration), and

2. at the host end (downline load configuration).

If the Models 1250/1251 VDUs are configured at the terminal end, the user selects and enters desired options from sets of functions called menus. If the Models 1250/1251 VDUs are configured at the host end, the host performs downline loading by sending the appropriate multicode sequences to the terminal.

For detailed information concerning user and downline load configurations, refer to the Models 1250/1251 Visual Display Units (VDU) Terminal Configuration User Guide and the Models 1250/1251 Visual Display Units (VDU) User's Manual.

5.3.2 Terminal Features and Special Character Format

The following sections describe special character sequences interpreted by the Models 1200 and 1250/1251. Multicode, attribute, and certain line character sequences can also be included within the user buffer to activate specific terminal features. The description of the input AID character, status bytes, and option bytes can be used to interpret the special sequences received from the terminals through terminal manager extended functions.

5.3.2.1 Multicode Sequence

Multicode sequences are used to implement such terminal operations as cursor movement, lock-out, setting tabs, field or screen clearing, editing and setting attribute bytes and print options.

The standard method for executing multicode operations from the host computer involves transmission of a multicode character to the terminal, immediately followed by a character designated unique to the function being performed. See Table 5-3 for a list of some common multicode sequence characters. For example, by transmitting the sequence ESC A (where ESC has been defined as the multicode character), the host commands the terminal to move the cursor up one line. Most multicode sequences on the Model 1200 can also be initiated from the keyboard by depressing the multicode key and then depressing the designated character key.

The ESC character is normally used as the multicode character for Models 1200/1250/1251 terminal configurations. However, the Models 1250/1251 terminal configuration allows the programmer to designate any ASCII character as the multicode character in the event that the ESC character is dedicated to another system function. In order to store a multicode character as itself, it must be entered through the keyboard or transmitted over the communications line twice.

TABLE 5-3 MULTICODE SEQUENCES

HEX CODE	ASCII DISPLAY	DESCRIPTION
21	!	Set attribute
28	(	Lock keyboard
29	)	Unlock keyboard
31	1	Set tab
32	2	Clear tab
33	3	Clear all tabs
3B	;	Set print options
3C	<	Send key override
41	A	Cursor up
42	B	Cursor down
43	C	Cursor right
44	D	Cursor left
45	E	Set full screen
46	F	Set partial screen
47	G	Set conversational
48	H	Home cursor
49	I	Clear line/field
4A	J	Clear unprotected*
4B	K	Clear display memory*
4C	L	Insert line*

TABLE 5-3 MULTICODE SEQUENCES (Continued)

HEX CODE	ASCII DISPLAY	DESCRIPTION
4D	M	Delete line*
4E	N	Insert character*
4F	O	Delete character*
50	P	Poll/select
51	Q	Reset modified data tags
52	R	Request to send
53	S	Set buffer address
54	T	Insert cursor
58	X	Set cursor pos-line
59	Y	Set cursor pos-character

* These multicode sequences require time delays in the transmission stream. The user should have NULL fill characters following the multicode sequence in the transmission buffer. For further information, refer to the Model 1200 and Models 1250/1251 User Manuals.

5.3.2.2 Attribute Characters

Attribute characters are used to define the start of a field and the mode in which the field is displayed. The end of a field is defined by a second attribute entry that defines the start of the next field.

Programmers can send attribute characters to the VDU by sending ESC ! (Hex 1B, 21) followed by the attribute character, and issuing a write edit with the attribute character generated. Also, programmers can issue a write-edit function call with the function modifiers such as write blink, write protect, write reverse video, and write low intensity. Figure 5-2 shows the format of the attribute character.

These Models 1200/1250/1251 commands are specified by an escape character (Hex 1B) plus the ASCII values shown in the column to the left of each command. See Appendix B for ASCII codes.

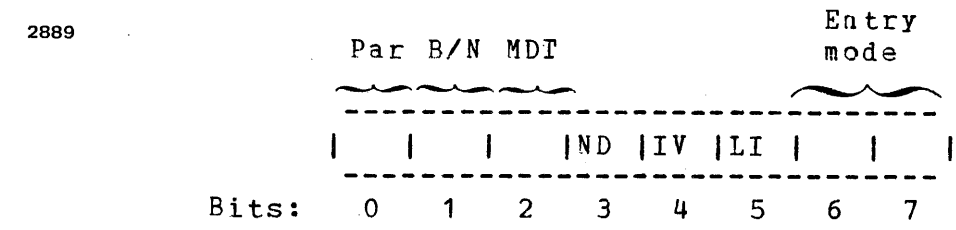


Figure 5-2 Attribute Characters

BIT	MEANING
0	Parity
1	0=normal field 1=blinking field
2	Modified data tag identifiers (modified field for read/send modified transmissions) 0=not modified 1=modified
3	0=display field 1=nondisplay field (security)
4	0=normal video 1=inverse video
5	0=normal intensity 1=low intensity
6-7	00=alphanumeric 01=numeric entry only (1-9., + -$) 10=protected field 11=graphics (bits 1-5=graphic characters)

5.3.2.3 Line Drawing Characters

User programs can cause the terminal to enter the graphic mode by sending a shift out character (SO, Hex '0E'). The characters following are then interpreted as line drawing characters. Refer to Table 5-4. User programs can exit the graphic mode by sending a shift in character (SI, Hex '0F').

User programs can also cause the terminal to enter the graphic mode by sending an attribute character (with bits 6 and 7 set to one) to be interpreted as a line drawing character. The graphic mode is exited immediately after outputting this attribute character.

TABLE 5-4 LINE DRAWING CHARACTERS

BITS 4 5 6 7	HEX VALUE	BIT1 = X BIT2 = X BIT3 = 0		BIT1 = X BIT2 = X BIT3 = 1	
0 0 0 0	0	@	-	P	└
0 0 0 1	1	A	¦	Q	┌
0 0 1 0	2	B	+	R	┐
0 0 1 1	3	C	┴	S	┘
0 1 0 0	4	D	┬	T	├
0 1 0 1	5	E	└	U	┤
0 1 1 0	6	F	┌	V	┤
0 1 1 1	7	G	┐	W	£
1 0 0 0	8	H	┘	X	÷
1 0 0 1	9	I	├	Y	>
1 0 1 0	A	J	┤	Z	<
1 0 1 1	B	K	─	[	→
1 1 0 0	C	L	≥	\	←
1 1 0 1	D	M	+	]	↑
1 1 1 0	E	N	┴	^	↓
1 1 1 1	F	O	┬	-	

5.3.2.4 Status and Option Bytes

The Models 1200 and 1250/1251 allow the user to read the status byte and two option bytes. The status byte is shown in Figure 5-3. The option bytes for the 1200 are shown in Figures 5-4 and 5-5. The option bytes returned from the Models 1250/1251 are always null.

2890

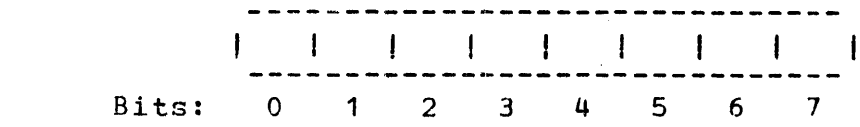


Figure 5-3 Status Byte Format

0-Parity bit (set accordingly)
1-Overrun=1
2-Parity error=1
3-Printer error=1
4-Printer busy=1
5-Keyboard locked=1
6-Command error=1
7-Background busy=1

2891

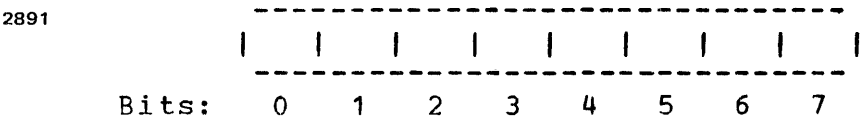


Figure 5-4 Option Byte 1 Format

0-Parity bit (set accordingly)
1-Send page terminator, ETX=0, EOT=1
2-CR line terminator enabled=1
3-Uppercase only (U/C)=1
4-AUTO LF enabled=1
5-Scroll enable=1
6-Full/partial screen, full=1
7-Conversational mode=1

2892

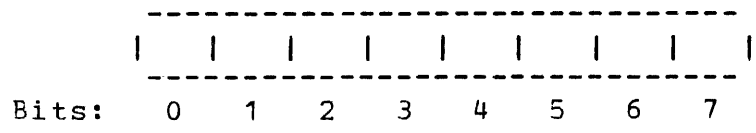


Figure 5-5 Option Byte 2 Format

0-Parity bit (set accordingly)
1-Transmission mode (nonconversational)
2-Send immediate all=00
  Send immediate unprotected=01
  Send immediate modified=10
  Request to send=11
3-Parity option-space=00
4-Mark=01, even=10, odd=11
5-Null suppress all=1
6-Send line terminator enable=1
7-Send line terminator EOT=1, ETX=0

5.3.2.5 Function Keys and Attention Identifier (AID) Characters

As an option, the Model 1200 offers 16 user-defined function control keys. The Models 1250/1251 offer 12 user-defined keys. The number of control codes generated can be expanded to 32 and 24, respectively, by depressing shift. These keys generate a multicode sequence followed by an AID character which is returned to the user buffer following an RQS function. Table 5-6 lists the function control keys and their associated AID characters for the Model 1200. Only keys 1-12 apply to the Models 1250/1251.

The three send functions listed in Table 5-5 are available on three separate keys on the Model 1200 keyboard. The Models 1250/1251 have one SEND key which can be designated as Send Line, Send Message, or Send Page.

TABLE 5-5 ATTENTION IDENTIFIER (AID) CHARACTERS

FUNCTION CONTROL KEY	AID CHARACTERS TRANSMITTED (HEX)	
	UNSHIFTED	SHIFTED
1	A (41)	a (61)
2	B (42)	b (62)
3	C (43)	c (63)
4	D (44)	d (64)
5	E (45)	e (65)
6	F (46)	f (66)
7	G (47)	g (67)

TABLE 5-5 ATTENTION IDENTIFIER (AID) CHARACTERS (Continued)

FUNCTION CONTROL KEY	AID CHARACTERS TRANSMITTED (HEX)	
	UNSHIFTED	SHIFTED
8	H (48)	h (68)
9	I (49)	i (69)
10	J (4A)	j (6A)
11	K (4B)	k (6B)
12	L (4C)	l (6C)
13	M (4D)	m (6D)
14	N (4E)	n (6E)
15	O (4F)	o (6F)
16	P (50)	p (70)
SEND PAGE	1 (31)	1 (31)
SEND LINE	2 (32)	2 (32)
SEND MSG	3 (33)	3 (33)

5.3.3 Modes of Operation

5.3.3.1 Conversational Mode

Conversational mode enables the Model 1200 and 1250/1251 VDUs to transmit and receive data on a character-by-character basis. Conversational mode supports read ASCII and write ASCII.

In conversational mode, the terminal manager provides the same logical capability as the TTY/VDU Terminal Manager.

During read ASCII all data is read until the buffer is full or a carriage return (CR) is found, whichever occurs first. When the read terminates, a CR and an LF sequence are sent to the terminal. When the character # is entered at the keyboard, the current line is ignored. When a backspace character is entered at the keyboard, the last character entered is ignored.

During write ASCII, data is output to the terminal until the buffer is empty or a CR is found. An LF is automatically appended to the detected CR. If no CR is found, an LF CR sequence is output to the terminal.

5.3.3.2 Image Mode

When operating in image mode, the Models 1200/1250/1251 terminal managers perform no special character recognition. Data is read or written until the buffer is empty. User tasks (u-task) must ensure that the I/O device is under control. To perform an image I/O, the SVC 1 function code must have the standard/extended options bit set and the extended option fullword format bit reset.

5.3.3.3 Editing Mode

The editing mode allows the applications programmer to use the editing capabilities of the Model 1200 and Models 1250/1251 VDUs. In the editing mode, the terminal can function as follows:

1. The application program outputs to the screen providing a fill-in-the-blanks form or data to be edited.

2. The terminal operator enters data to the terminal while the terminal is in local mode and indicates screen-update completion by depressing a send or function key.

3. The CPU receives screen data in a single buffered read.

Execution of data transmission from the terminal varies depending on whether the read is performed in read-immediate mode or send immediate mode, and on which of the three types of read formats (read all, read modified, or read unprotected) is requested. In addition, certain special output functions (clear screen, insert tabs, move cursor) permit modification of the screen image; certain special input functions (read cursor position, read status) permit determination of current screen state. The concepts behind these functions are detailed in the following section.

5.3.3.4 Concept of Request to Send (RQS) and Immediate

In immediate mode, data is entered from the keyboard and stored in terminal memory until the operator depresses a send key. The stored data is then transmitted to the host CPU. This mode is subdivided into three programmable transmission modes:

- Send immediate all
- Send unprotected
- Send modified

These modes permit, respectively, transmission of all data in terminal memory, only data in unprotected fields, or only data in fields modified by the operator. The operator, in turn, can select transmission of a single line, a message starting at the termination of the last message, or an entire page.

The RQS mode allows the system designer to fully optimize the allocation of host CPU memory. As in immediate mode, no data is transmitted while the operator is editing a page to be entered. However, when a send key is depressed, the terminal transmits a RQS sequence only. This sequence informs the program of which send key was depressed; e.g., send line, send message, or any of the 16 special function keys. The program then identifies the depressed key, allocates the necessary buffer storage, and issues the required read multicode sequence to initiate transmission. The RQS mode permits all of the preceding at the discretion of the host CPU. In addition, the programmer can use a send-key-override command to permit the program to select the correct send mode if the operator's choice of keys is unacceptable.

5.3.3.5 Types of Data Read Formats

Four types of data read formats are possible: Read All, Read All With Attribute Character Truncated, Read Modified, and Read Unprotected. The description of each of these data read formats follows:

1. Read All. Based on the setting of the full/partial screen selector, data is transferred starting from home or the current cursor position. If the null-suppress-all-option is enabled, nulls are suppressed, and the end of a line is indicated by transmission of a CR character (X'0D'). The character LF (X'0A') is transmitted following a CR if AUTO/LF is enabled. There is no specific format for the data read in the buffer. All data displayed on the VDU is filled into the user program buffer. The data format is illustrated in Figure 5-6.

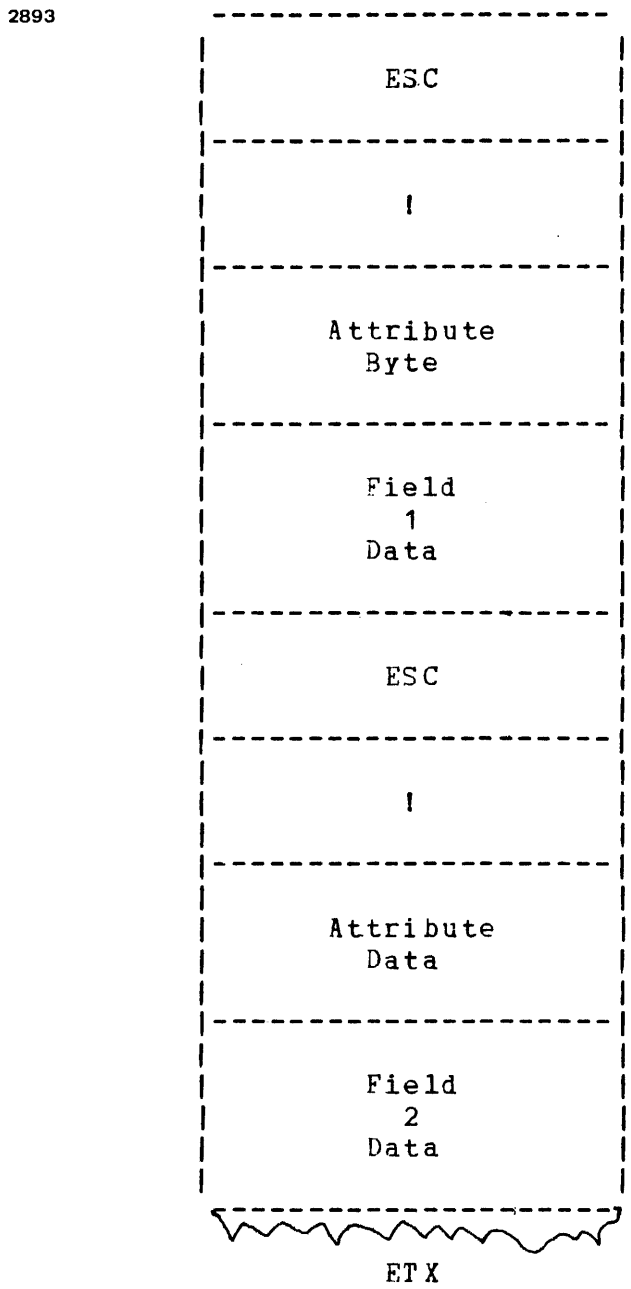


Figure 5-6 Read All with Format Screen

2. Read All with Attribute Character Truncated. The terminal manager throws away the ESC ! (X'1B',X'21') received and sends the following character (by setting the parity bit) as an attribute character. There is no specific format for the data in the user buffer. All characters except ESC ! are sent into the user buffer. The data format is illustrated in Figure 5-7.

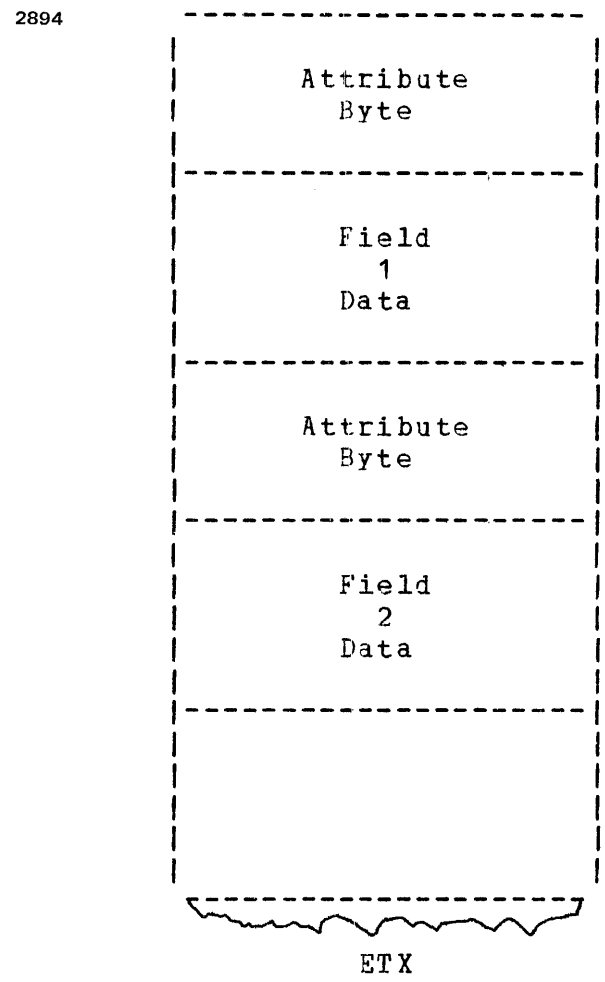


Figure 5-7 Read All with Attribute Character Truncated

3. Read Unprotected. If the screen is unformatted; (i.e., contains no fields) this operation has the same effect as read/send all. If the screen is formatted, only unprotected fields are transmitted; nulls are suppressed. A field containing all nulls is indicated by a single group separator (GS) on the Model 1200 or a single Formatted-Send Modified, Send Unprotected-Field Terminator on the Models 1250/1251. The data formats are illustrated in Figure 5-8 and 5-9.

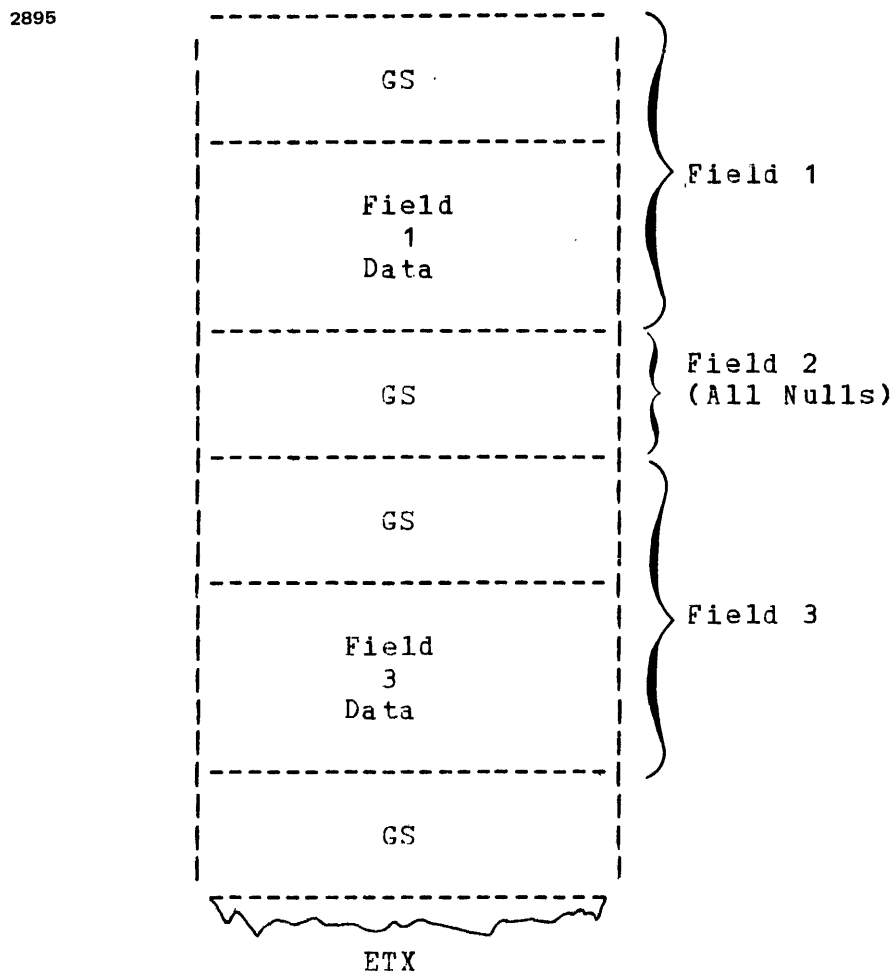


Figure 5-8 Read Unprotected Format (Model 1200 VDU)

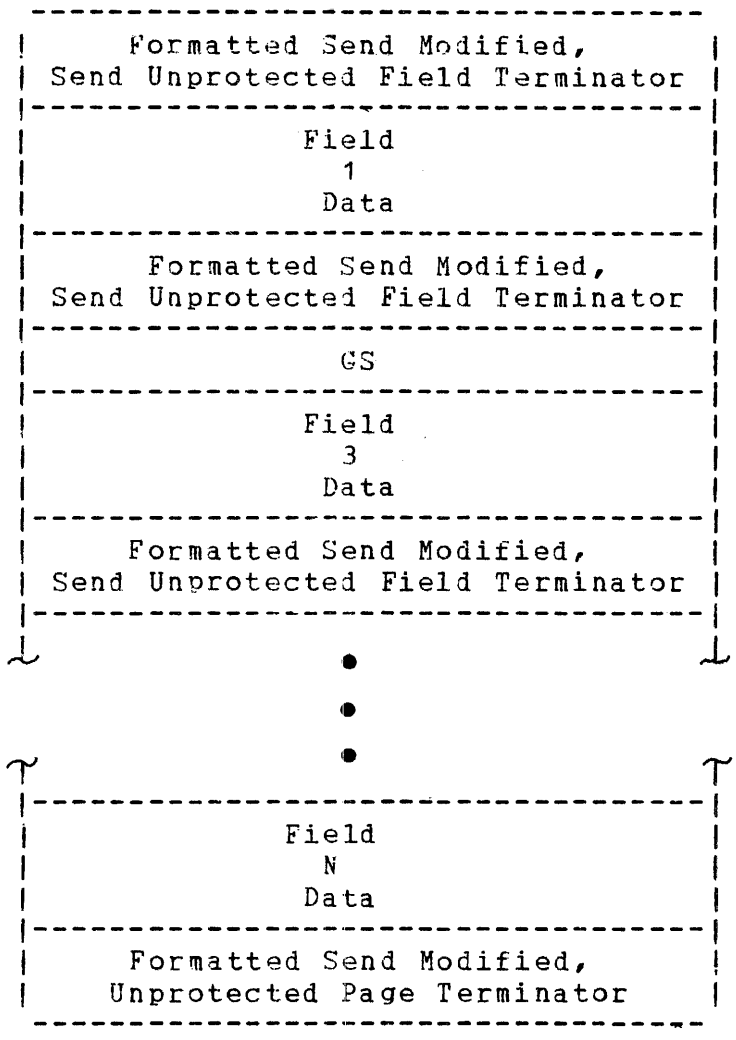


Figure 5-9 Read Unprotected Format (Models 1250/1251 VDUs)

Execution of send line with a formatted screen transmits only the first field of the current cursor line following the current cursor position.

4. Read Modified. This operation is legal only on a formatted screen. Read modified transmits only those fields for which the modified data tag is set in the field attribute character. A modified data tag is set when data is entered in a non-light pen field or if the designator character of a light pen field has been altered by light pen detection. Transmission starts with an SOH, followed by a 2-character cursor address sequence, and then the modified fields. Each field is preceded by a GS character and the buffer address of the first data character in the field for the Model 1200. For the Models 1250/1251, the Formatted-Send Modified, Send Unprotected-Field Terminator preceeds each field. Nulls are suppressed. Model 1200 terminates transmission with an ETX character; Models 1250/1251 use the Send Modified, Send Unprotected Page Terminator. These formats are illustrated in Figures 5-10 and 5-11 where it is assumed that fields 1 and 3 have been modified, and field 2 has not. Figure 5-11 also assumes that there are no light-pen fields. Light pen fields transmit only the address of the field rather than the address and the data.

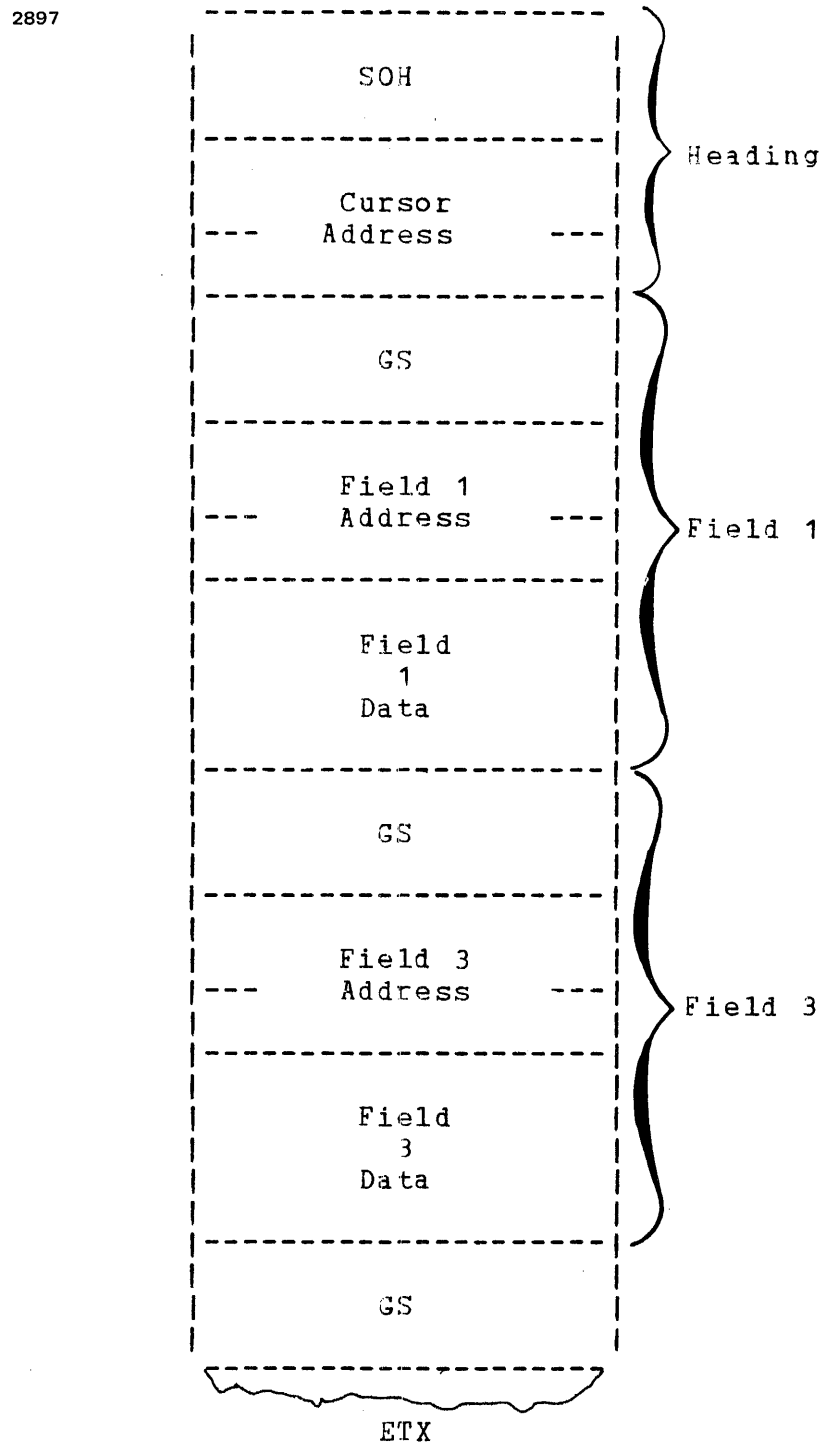


Figure 5-10 Read Modified Format (Model 1200 VDU)

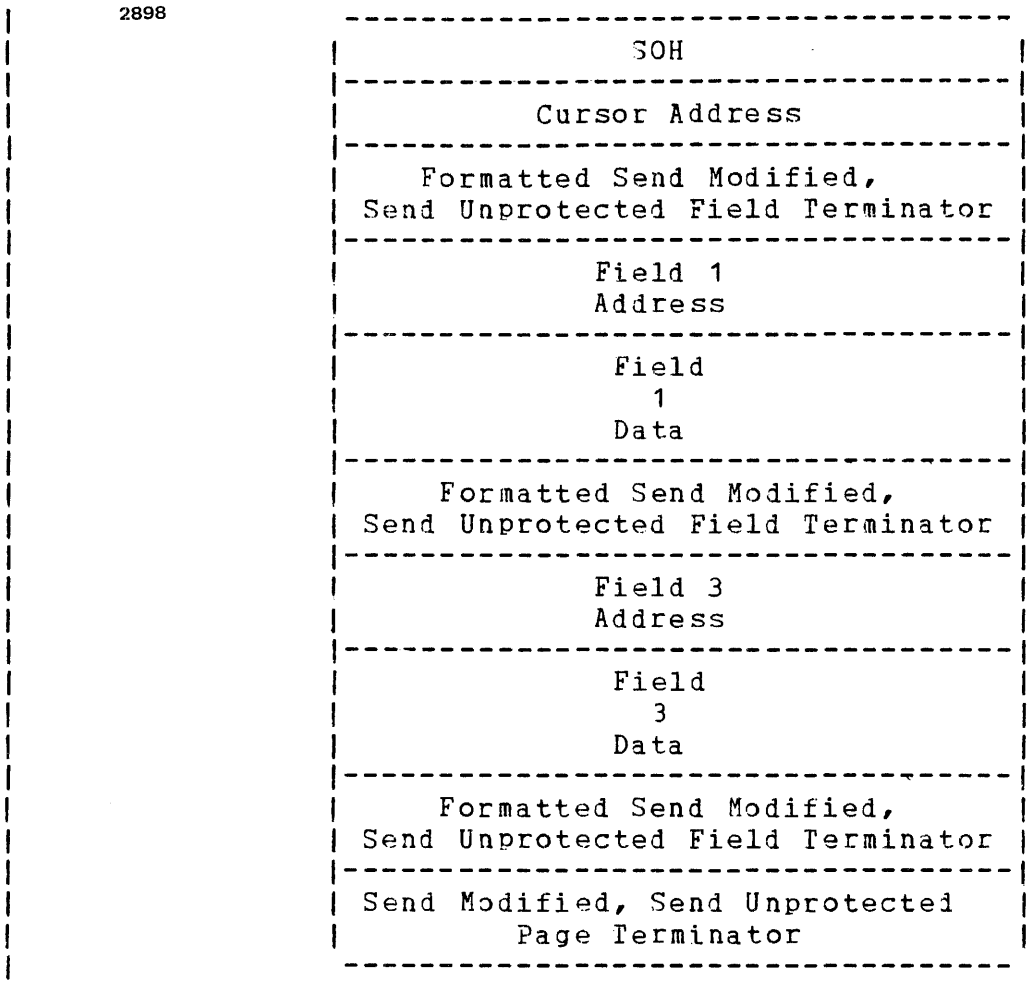


Figure 5-11 Read Modified Format (Models 1250/1251 VDUs)

5.3.3.6 Types of Data Write Formats

Two types of edit data write formats are provided: edit write and write edit with attribute character generated.

- Edit Write. The terminal manager sets the Models 1200 or 1250/1251 in normal mode and disables echoplex. The data in the user buffer is sent to the VDU.

- Write Edit with Attribute Character Generated (WAG). When the terminal manager receives a character with parity bit set, it generates an ESC ! multicode sequence to set the attribute character. The character received is considered as an attribute character to be sent to the VDU.

5.3.3.7 Cursor Addressing and Random Addressing

The terminal manager enables user programs to read cursor address (line 00-23/column 00-79), to home cursor, and to set cursor positions provided as line/column in the user buffer.

The terminal manager also handles random I/O by setting the cursor position first (for read case) or the VDU display buffer (for write case) before issuing I/O. The line/column is provided in the random field (first two bytes) of the SVC 1 parameter block.

The user can also imbed the control sequence in the data to be sent to the terminal to achieve cursor addressing and random address. See Table 5-6 for multicode sequence. The line/column format of the cursor or display buffer in the embedded data is (xx,xx). Refer to Table 4-4.

5.3.3.8 Horizontal Tabulation

Figure 5-12 provides 80 bytes for the user task to specify tab-stop location. Each byte contains a hexadecimal tab-stop position (0-4F) corresponding to the column (1-80) of the VDU which is being specified as a tab-stop location. Each byte is sequentially scanned until an FF is encountered, which ends the setting of tab stops. Otherwise, since the screen has only 80 columns, 80 is the maximum number to scan.

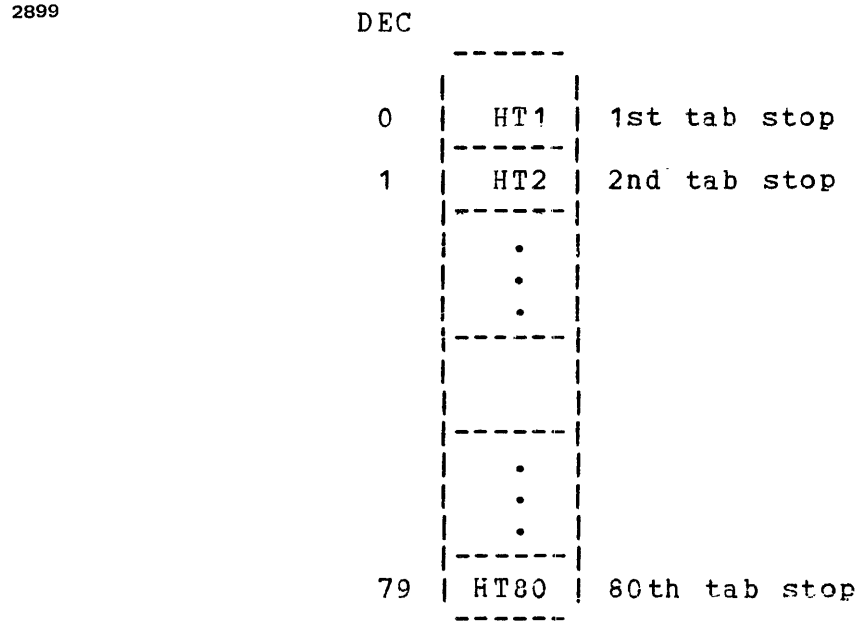


Figure 5-12 User Buffer Format Table for Horizontal Tabs

5.3.3.9 Printer Options

The character following multicode is an ASCII digit from 1 to 5 (X'31' to X'35') that determines the operation of the auxiliary serial interface, if enabled. On all printouts, the attribute characters, line characters, and fields specified as nondisplay are printed as spaces.

- No printout. The printout currently in progress is allowed to finish, and then printing stops.
- Print screen from home. The entire screen is printed, up to the end of the screen.
- Print screen from cursor. The screen is printed from the current cursor position to the end of the screen.
- Simulprint (FF control). On receipt of a form-feed character, the contents of the screen are printed. If read-status-when-ready is in effect, a status byte is transmitted when printing is complete. The normal effect of the form feed (clear unprotected and home cursor) takes place until printing is complete.
- Simulprint (continuous). In this mode, characters are simultaneously printed and displayed as they are received from the line. If the printer is running too slowly for the line, it transmits a DC4 control character to request that the host pause transmission. A DC2 control character is transmitted to indicate that the host can resume transmission.

5.3.4 Sysgen and Environment

Models 1200/1250/1251 terminal managers support is obtained by specifying Model 1200 or Models 1250/1251 at sysgen. The terminal is configured just as any local device such as a line printer or card reader.

The only restriction on the required order of modules in the library is that the DCBs for all devices supported by a particular driver or terminal manager must precede the driver or terminal manager.

The system software required to support the terminal manager includes:

- OS/32 R06 and higher
- System support module
- Asynchronous line driver

5.3.4.1 Special Parity Requirement

The terminal manager uses even parity to output all data and control sequences to the Model 1200 or Models 1250/1251 VDUs. The terminal manager also expects all input data and control sequences received from the terminal to have even parity. These procedures require that the switch on the Model 1200 be set in the even-parity position or the parity be dynamically defined for the Models 1250/1251. Failure to do so results in I/O parity errors.

5.3.4.2 Extended Device Code Specification

The method of system generation used for the Model 1200 uses the default option index (bits 12 to 15 of the extended device code halfword). The format of the extended device code halfword is shown in Figure 5-13. The extended device code is initialized at sysgen time by using CUP/32. Consult the OS/32 System Planning and Configuration Guide for operation of this program.

Devices other than the Models 1200 and 1250/1251 VDUs that can be used with this terminal manager require the master bit (bit 0) of the extended device code to be set in the configuration utility program input deck. Setting this bit directs the terminal manager not to send the ESC G sequence to the terminal on conversational calls.

When bit 0 of the device code halfword is not set, an I/O request cannot be halted while certain 2-character control sequences are being output, such as Enter Conversation or Edit Mode. If the Models 1200 or 1250/1251 terminals are set for conversational mode during power-up, the Halt I/O feature can be initiated by setting bit 0. See the Model 1200 Terminal Installation and Programming Manual for power-up and mode-setting procedures.

2900

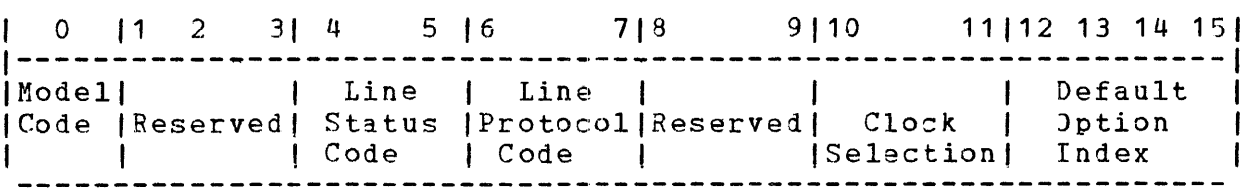


Figure 5-13 OS/32 Basic Data Communications Extended Device Code Halfword

	DECIMAL VALUE (MASK)	HEXADECIMAL VALUE
Model Code		
Terminal is Model 1200 or Models 1250/1251	0	0000
Terminal is not Model 1200 or Models 1250/1251	32768	8000
Line Status Code		
Direct connection	0	0000
Leased line	1024	0400
Dial-in manual dial	2048	0800
Line Protocol Code		
Half-duplex 4-wire	0	0000
Half-duplex 2-wire (Model 1200 only)	768	0300
Simplex read (Model 1200 only)	256	0100
Simplex write (Model 1200 only)	512	0200
Clock Selection		
Clock A	0	0000
Clock B	16	0010
Clock C	32	0020
Clock D	48	0030
Default Option Index	Refer to Table 5-7	

5.3.4.3 Default Extended Functions/Options

If the extended option bit (bit 7) is set in the SVC 1 function code, the terminal manager further examines the extended functions/options the user provides. Otherwise, a table of default-extended functions/options is provided. See Table 5-6. The particular entry of this table is selected according to the encoded value of bits 12-15 of the extended device code (DCB.XDCD).

TABLE 5-6 DEFAULT EXTENDED OPTIONS

EXTENDED DEVICE CODE		DEFAULT EXTENDED OPTIONS SELECTED
OPTION	HEX VALUE	DESCRIPTION
0	0	Conversational, formatted I/O, unlock keyboard after I/O
1	1	Same as encoded value 0 case, plus enable upper case only
2	2	Same as encoded value 0 case, plus nontermination on CR
3	3	Conversational, formatted I/O, lock keyboard after I/O
4	4	Same as encoded value 3 case, plus enable upper case only
5	5	Same as encoded value 3 case, plus nontermination on CR
6	6	Combination of encoded value 1, 2
7	7	Combination of encoded value 4, 5
8	8	Unused, can be source SYSGENed by user
9	9	Unused, can be source SYSGENed by user
10	A	Unused, can be source SYSGENed by user
11	B	Unused, can be source SYSGENed by user
12	C	Unused, can be source SYSGENed by user
13	D	Unused, can be source SYSGENed by user
14	E	Unused, can be source SYSGENed by user
15	F	Unused, can be source SYSGENed by user

5.3.4.4 Terminal Switch and Strap Settings

It is imperative that certain Model 1200 strap and switch settings be in a specific position, while others can be set in various positions according to system needs. These settings are dynamically defined for the Models 1250/1251. A set of required/optional terminal switch settings, strap settings, and dynamic definitions follows:

- Multicode character selection = "ESC" - Required
- Send line terminator = "ETX" - Required
- Send page or send message terminator = "ETX" - Required
- Send line terminator enable, ETX follows CR at end of send Line (unformatted) - Required
- Strap to enable transmission of CR at end of every line in unformatted send page or send message - Optional
- Disable blinking block cursor - Optional
- Parity selection - must be in even position - Required

5.4 INTERNAL TERMINAL MANAGER DESIGN

The Models 1200/1250/1251 terminal manager enables user programs to communicate with the VDU via SVC 1 supervisor calls (with extended functions/options). The Models 1200/1250/1251 terminal managers support read, write, wait, proceed, unconditional proceed, image I/O, random addressing, and user-extended functions/options. Four operational modes of the Models 1200/1250/1251 terminal managers are supported: conversational mode, image mode, data-dependent editing mode, and data-independent editing mode.

5.4.1 Relationship to Operating System and Asynchronous Line Driver

The device-independent support of the Model 1200 and Models 1250/1251 VDUs in a point-to-point environment is provided by the Models 1200/1250/1251 terminal managers. The terminal manager calls the asynchronous driver to perform user-desired I/O. However, a user SVC 1 enters the terminal manager only via the SVC 1 executor of the OS. Refer to Figure 5-14 for the relationships to OS and the Asynchronous Line Driver.

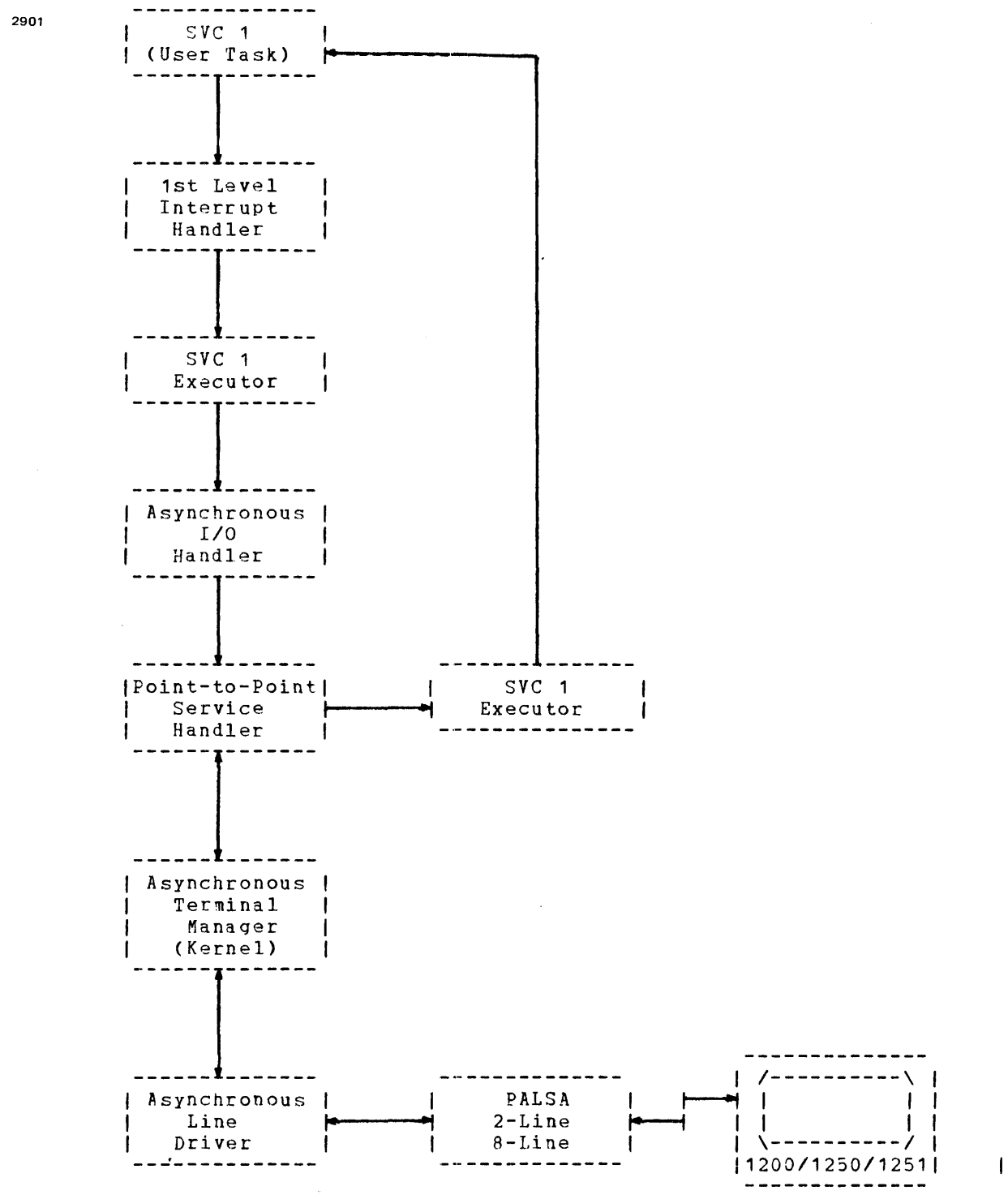


Figure 5-14 Models 1200/1250/1251 Terminal Manager/Asynchronous Line Driver Relationship (Point-to-Point)

The terminal manager is initially entered from the SVC 1 executor which validates the SVC 1 function call, sets up the OS environment, and sets up the OS/32 Basic Data Communications environment. The asynchronous line driver provides the interface between the terminal manager and an asynchronous communications adapter connected to the VDU. Figure 5-15 lists the DCB Fields For the Model 1200 and Models 1250/1251 VDUs.

2902

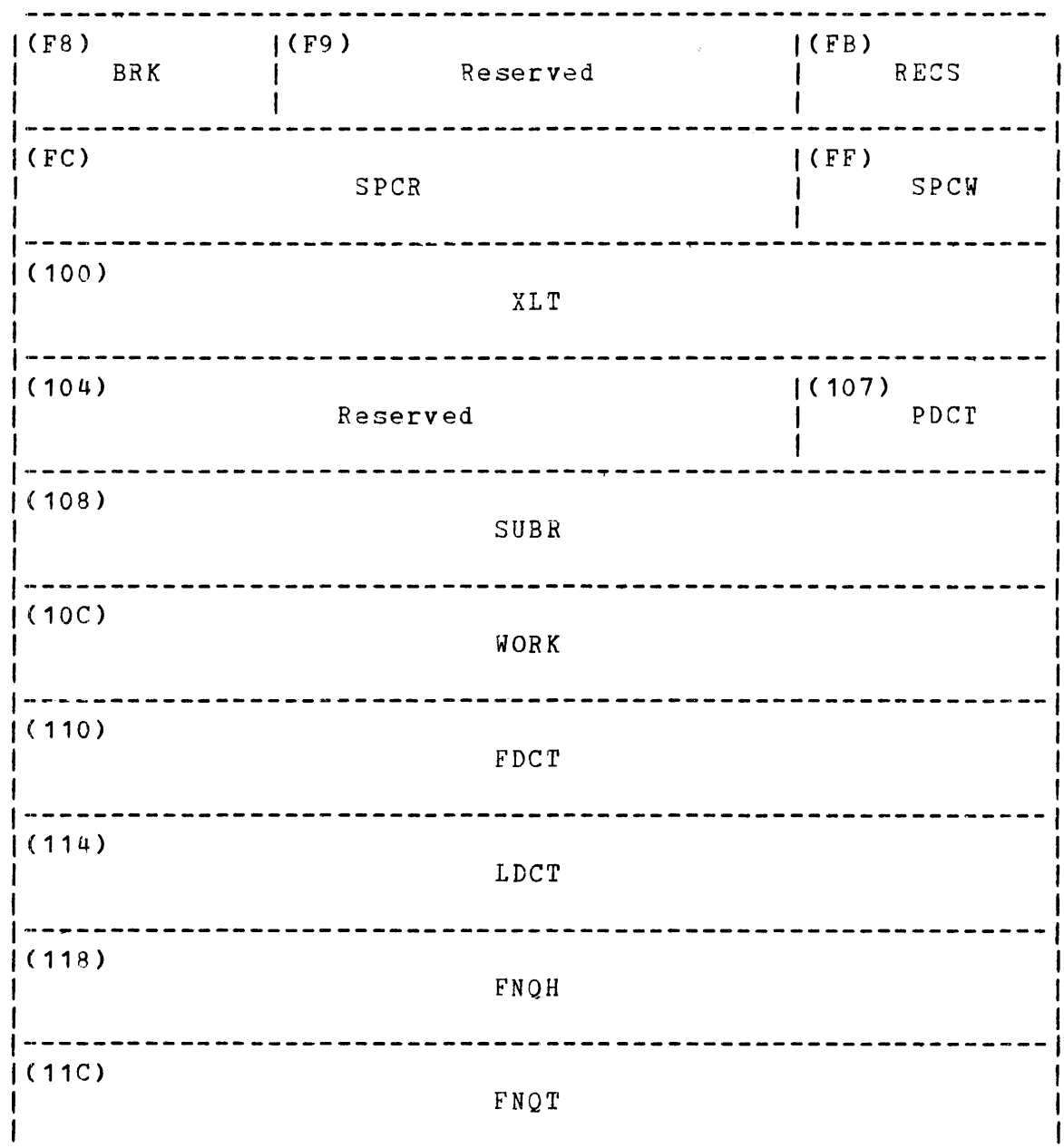


Figure 5-15 Device Control Block (DCB) Fields for Models 1200 and 1250/1251 VDUs

(120)	TMLH
(124)	BFPT
(128)	CDAT
(12C)	CMDM
(134)	EXIT
(138)	LINK
(13C)	INDX
(140)	SVFR
(144) PTMV	(146) Reserved

Figure 5-15 Device Control Block (DCB) Fields for Models 1200 and 1250/1251 VDUs (Continued)

These fields must follow the BASIC DCB and the communications subsystem DCB fields described in the OS/32 Basic Data Communications Reference Manual.

Following is a description of the Models 1200/1250/1251 related DCB fields:

DCB.BRK	Output command for break
DCB.RECS	Transparent record size
DCB.SPCR	Special character mask for read

DCB.SPCW	Special character mask for write
DCB.XLT	Translate table address
DCB.PDCT	Count of trailing (pad) characters
DCB.SUBR	Subroutine return address save area
DCB.WORK	Working storage used by terminal manager
DCB.FDCT	First device control table
DCB.LDCT	Last device control table
DCB.FNQH	First device control table in function queue
DCB.FNQT	Last device control table in function queue
DCB.TMLH	Logical timerchain head PTR
DCB.BFPT	Pointer to buffer in system space
DCB.CDAT	Current date
DCB.CMDM	Bit mask of invalid extended function
DCB.EXIT	Return address initialized
DCB.LINK	Link address initialized
DCB.INDX	Extended option index initialized
DCB.SVFR	Return address for MSVP WTB/DO.SVFI
DCB.PTMV	Timer value for poll cycle

5.4.2 Internal Structure

Following is a functional description of the three major components of the Models 1200/1250/1251 terminal managers. See the flow diagram, Figure 5-16.

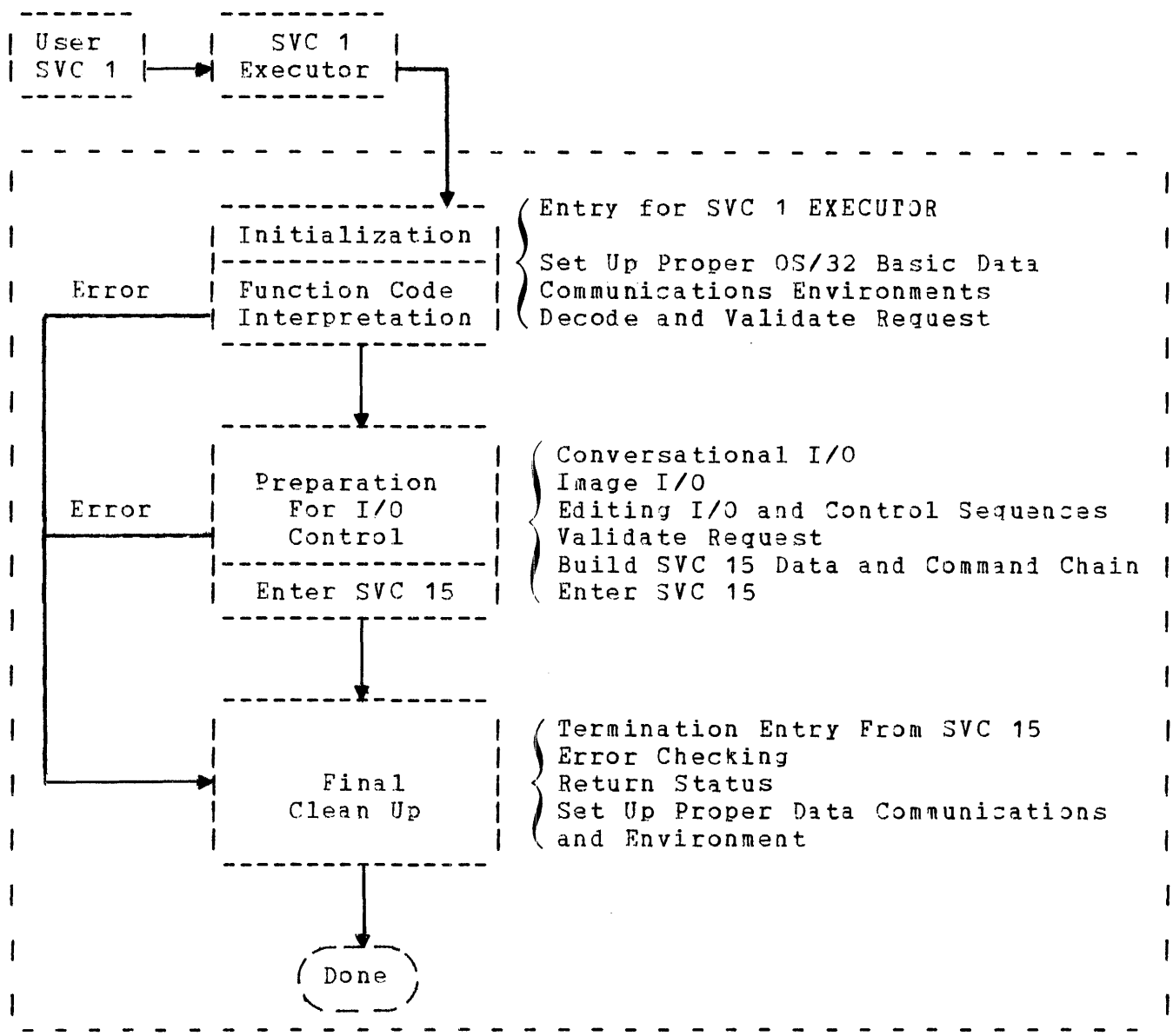


Figure 5-16 Kernel Terminal Manager Flow Diagram

5.4.2.1 Initialization and Function Code Interpretation

This component provides the entry for SVC 1 execution, sets up data communications environments, decodes the SVC 1 function code with extended functions/options, sets the default extensions as required, validates extended functions/options, and jumps to various routines to handle various I/O preparations.

5.4.2.2 Preparation for I/O, Control, and Enter SVC 15

From the initialization decoding process, various routines are provided in this component. Validation of extended functions/options in the individual case is also performed. After all preparations and the SVC 15 data/command chain are built, actual entering of SVC 15 is initiated. This component also provides control sequences.

5.4.2.3 Final Clean Up

This component provides termination return of SVC 15, handles errors, returns status to the user, and sets up or restores data communications control.

5.5 TYPICAL PROBLEM CHECKLIST

The user should exercise caution in using the Model 1200 and the Models 1250/1251 point-to-point VDUs. Users can run stand-alone diagnostics or call customer service if the following error checklist entries are not responded to in the correct manner:

1. Model 1200 switch is not in even parity position or even parity is not defined for Models 1250/1251.
2. Read immediate does not follow request to send.
3. Failure to provide random address (X,Y coordinates) in user SVC 1 parameter block for random read/write
4. Failure to provide tab-stop positions in the user buffer
5. Failure to provide cursor position (00-17, for line #, 00-4F for column #) for setting cursor position in the user buffer
6. Failure to supply a buffer. A valid buffer address is required for all calls.
7. Incorrect PALS/PASLA cable
8. Terminal is not online.
9. Incorrect speed match between terminal switch settings, PALS/PASLA clock strapping and CUP OS/32 Basic Data Communications extended device code
10. Terminal switch was set without performing either a powerup/down or clear all.
11. Auto line feed, new line, or scroll enable switch definitions are inconsistent with data output.

12. Failure to provide null characters following certain embedded escape sequences

13. Failure to activate CPU clock prior to telephone answer sequences

14. Pin 25 is not cut on PALS/PASLA cable.

15. Incorrect Bell modem options were specified.

5.6 MULTIDROP FEATURE

The terminal manager supports the mutlidrop feature only on the Models 1250/1251 VDUs. This feature allows multiple Models 1250/1251 VDUs to share a single communications line by using the poll and select technique. Since only one terminal can transmit data at any one time in a multidrop environment, terminal response is delayed longer than in a point-to-point environment.

Multidrop support includes:

- dynamic generation of necessary data structures,
- transparent access to multidrop devices at the device independent I/O level,
- support of all multidrop functions limited to block mode, and
- remote connections support via Bell* 212A, 103J, or 113D series modems.

5.7 LIGHT PEN

The terminal manager also supports the use of the light pen on the Models 1250/1251. The light pen is a light-sensitive pen connected to the terminal that detects variations in light emitted by the data characters on the screen. Refer to the Models 1250/1251 VDUs User's Manual for a detailed description of the light pen feature.

5.8 ENCODED ERROR MESSAGES

Encoded messages are displayed as the result of an error occurring when a READ or WRITE macro is issued. Table 5-7 lists the messages that can be generated.

TABLE 5-7 ENCODED ERRORS AND DEFINITIONS FOR MODELS 1200/1250/1251 TERMINAL MANAGERS

STATUS CODE (HEX)	MEANING
0000	No errors
8402	Line delete caused termination during read
8203	Break detected during write
8204	Break detected during read
8205	Terminated by data error (see parity bits)
8208	Framing or stop-bit error
8409	Reverse channel error
200A	Lost carrier on read
200B	Lost clear-to-send on write
200C	Data set not ready
840D	Device unavailable; adapter not present
200E	Character overflow
840F	Ring status detected during data transfer
8410	Busy and/or done bits in chained buffers bad; may indicate priority to low
8411	Number of commands executed greater than 255
8412	Task queue full, invalid, or nonexistent.
8413	Buffer-management-routine error; may indicate priority too low
8282	Timeout
8281	Halt I/O request aborted I/O
8481	Illegal command or modifier
8419	Memory fault in referencing data
841A	Memory fault in referencing buffer
811B	Lu illegal
821C	Illogical device status
A01D	Power failure
841E	Illegal software condition
841F	Illegal translation table
8225	Timeout during connect sequence

TABLE 5-7 ENCODED ERRORS AND DEFINITIONS FOR MODELS 1200/1250/1251 TERMINAL MANAGERS (Continued)

STATUS CODE (HEX)	MEANING
8426	ESC, R not received on RQS
8227	DMA I/O System (DIOS) hardware error
*C028	Attempted a conversational I/O to a polled terminal
*A029	No response from terminal to poll

* Applies to Models 1250/1251 only

The first byte of each status code listed in Table 5-1 refers to the device independent status of the error. These codes are defined as follows:

CODE	DEFINITION
C0	Illegal function
AC	Device unavailable, signoff user. If switched line, reissue call.
20	Device unavailable or parity error sent to recovery routine
90	End of medium
88	End of file
84	Unrecoverable error; report to operator
82	Parity or recoverable error; reissue the call.
81	Illegal or unassigned lu

APPENDIX A
DRIVER COMMAND WORD (DCW) FORMAT

2519

	COMMAND	MODIFIER/ COMMAND BYTE HEX	VALID COMMAND HALFWORDS	NO. DATA FIELDS	DATA FIELD SPECIFIES
NULL	NOP	XX00	CC CT X X XXXX 00000 000	1	Any valid address
	WAIT	XX08	CC CT X 0 XXXX 00001 000	1	Halfword
	XFER	XX10	CC CT X X XXXX 00010 000	1	Halfword
	CXFER	XX18	CC CT X X XXXX 00011 000	2	2 Halfwords Valid Address
CONTROL	EXAMINE	XX01	CC CT X TO XXXX 00000 001	1	Byte
	RING WAIT	XX09	CC CT X TO XXXX 00001 001		None
	ANSWER	XX11	CC CT X TO XXXX 00010 001		None
	DISCONNECT	XX19	CC CT X TO XXXX 00011 001		None
READ	READ BUFFER	XX02	CC CT BT TO XXXX 00000 010	1 or 2	Buffers
	READ1	XX0A	CC CT BT TO XXXX 00001 010	1	Byte
	READ2	XX12	CC CT BT TO XXXX 00010 010	2	Bytes
PREPARE	PREP	XX03	CC CT X TO XXXX 00000 011	1	Byte
WRITE	WRITE BUFFER	XX04	CC CT BT TO XXXX 00000 100	1 or 2	Buffers
	WRITE1	XX0C	CC CT BT TO XXXX 00001 100	1	Byte
	WRITE2	XX14	CC CT BT TO XXXX 00010 100	2	Bytes
HOLD	BREAK	XX05	CC CT X TO XXXX 00000 101	1	Halfword
MODE	TOUT	XX06	CC CT X X XXXX 00000 110	1	Fullword
	CMD2	XX0E	CC CT X X XXXX 00001 110	1	Byte
	RCMD	XX16	CC CT X X XXXX 00010 110	1	Byte
	WCMD	XX1E	CC CT X X XXXX 00011 110	1	Byte
	RDIS	XX26	CC CT X X XXXX 00100 110	1	Byte
	WDIS	XX2E	CC CT X X XXXX 00101 110	1	Byte
	DISC	XX36	CC CT X X XXXX 00110 110	1	Byte
	TRNSL	XX46	CC CT X X XXXX 01000 110	1	Byte
	SPEC CHAR	XX4E	CC CT X X XXXX 01001 110	1	Fullword

APPENDIX B
STANDARD ASCII CODE

2520

Bits b7 b6 b5					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
				COLUMN →								
b4	b3	b2	b1	ROW ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	¦
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

* Standard 96-character ASCII Set

** 64-character ASCII Set displayed when U/C Switch is enabled. (DEL is as legal character in this mode and is displayed as a quadrangle with alternate dots.)

*** In transparent mode, all control codes are displayed as shown in sample display. In normal display mode, control characters are not displayed.

APPENDIX C
SAMPLE PRINTOUT

```
2521

         SCRAT
         WIDTH 120
         TARGT 32
EXAMPLE1 PROG                  EXAMPLE1   ITAM/32
*
*
*        THIS EXAMPLE ILLUSTRATES EDITING MODE INPUT WITH EDITING CRT
*        USING THE ITAM/32 SVC 15 SUPERVISOR CALL
*        THE   PROGRAM WRITES A FORM TO THE CRT
*        THE   FORMAT  IS:    NAME:
*                             STREET ADDRESS:
*                             CITY:
*                             STATE,ZIP CODE:
*                             ACCT.NO:
*
*        THE   USER RESPONDS BY FILLING IN THE FORM
*          AND TRANSMITTING THE INFORMATION TO THE COMPUTER
*        IN EDITING OR NON-EDITING MODE.
*
*        LU 1- SVC15 I/O DEVICE EDITING CRT
*        LU 3-LIST DEVICE
*        LU 7-ERROR MSG DEVICE
*   THE USER MUST PERFORM THE FOLLOWING ASSIGN COMMAND FOR LOGICAL
*       UNIT 1:
*                ASSIGN 1,CRT1:,,,SVC15        ASSIGN LU1 FOR SVC15 I/O
*
*
*
* ***************************************************************************
         EJECT
START    EQU   *
         B     EXAMPLE1
* ***************************************************************************
SVC15.   STRUC
SVC15.FN DS    0                        FUNCTION CODE
SVC15.FC DS    1                        FUNCTION CODE
SVC15.LU DS    1                        LOGICAL UNIT
SVC15.ST DS    2                        STATUS
SVC15.CN DS    1                        COMMAND NUMBER
SVC15.CM DS    3                        POINTER TO DCW CHAIN
SVC15.LR DS    2                        LENGTH OF LAST READ
SVC15.LW DS    2                        LENGTH OF LAST WRITE
SVC15.DA DAS   0                        DATA WORD
SVC15.DC DS    1                        DATA CODE
SVC15.DF DS    3                        DATA FIELD ADDR
         ENDS
         NLIST
SVC1.    STRUC
SVC1.FUN DS    0                       FUNCTION CODE
SVC1.FC  DS    1                       FUNCTION CODE
SVC1.LU  DS    1                       LOGICAL UNIT
SVC1.STA DS    1                       STATUS
SVC1.DN  DS    1                       DEVICE NUMBER
SVC1.SAD DAS   1                       STARTING ADDR
SVC1.EAD DAS   1                       ENDING ADDR
SVC1.RAD DAS   1                       RANDOM ADDR
SVC1.LXF DAS   1                       LENGTH OF LAST XFER
         ENDS
```

```
REGS     STRUC
U0       DS    1
U1       DS    1
U2       DS    1
U3       DS    1
U4       DS    1
U5       DS    1
U6       DS    1
U7       DS    1
U8       DS    1
U9       DS    1
U10      DS    1
U11      DS    1
U12      DS    1
U13      DS    1
U14      DS    1
U15      DS    1
         ENDS
         LIST
* ************************************************************************
         EJECT
ERROR    EQU   X'8000'
EDITERR  EQU   X'FFDF'                    ERROR MASK EDITING CRT
F        EQU   15
READSPC  EQU   X'F9ED'                    DEFAULT READ SPCHAR
WRTSPC   EQU   X'8004'                    DEFAULT WRITE SPCHAR
NXSPCR   EQU   0                        INDEX FOR SPCHAR READ
NXSPCW   EQU   2                        INDEX FOR SPCHAR WRITE
NOECHO   EQU   X'81'                      ENABLE+DTR+READ
LK       EQU   X'7D'                      LOCK KEYBOARD EDITING CRT
PC       EQU   X'69'                      POSITION CURSOR (HOME CURSOR)
XMRD     EQU   X'6B'                      XMIT CODE OUTPUT
CM       EQU   X'76'                    CLEAR MEMORY
CHARS    EQU   60                         NUMBER OF CHARS PER LINE
MAXLINES EQU   10                         MAX NUM LINES ALLOWED
UNLK     EQU   X'70'                      UNLOCK/RELEASE KEYBOARD
ECHO     EQU   X'71'                      ENABLE+DTR+READ+ECHOPLEX
LF       EQU   X'0A'
CR       EQU   X'0D'
SPS      EQU   X'79'                      START PROTECT SEQUENCE
EPS      EQU   X'7A'                      END PROTECT SEQUENCE
SBS      EQU   X'7B'                      START BLINK SEQUENCE
EBS      EQU   X'7C'                      END BLINK SEQUENCE
XMWR     EQU   X'6A'
XMSTAT   EQU   X'0040'                    XMIT STATUS BIT
TRPWSV15 EQU   Y'88000300'                TRAP WAIT, TASK QUEUE SERVICE
*                                         TRAP ENABLE,QUEUE ENTRY TIMEOUT
*                                         COMPLETION,SVC 15 & ENTRY ENABL
TERMTRAP EQU   Y'0C000000'
POLKMASK EQU   Y'00FFFFFF'
REASMASK EQU   Y'FF000000'
TSWNEW   EQU   Y'300'
GSIZE    EQU   04
RTNADDR  EQU   4
ETX      EQU   X'03'
PARAM    EQU   Y'01000000'          DATA CODE = PTR TO DCW PARM
DIRECT   EQU   Y'0'                 DIRECT TEXT DATA CODE
INDIRECT EQU   Y'04000000'          INDIRECT TEXT DATA CODE
LINE     EQU   19                       NO. OF INPUT CHARS -1
FORMLEN  EQU   5                    NO. OF LINES IN FORM
SPACE    EQU   X'20'
* ************************************************************************
         NLIST
         COPY  UDL
         LIST
         EJECT
* ************************************************************************
```

```
EXAMPLE1 EQU   *
* ***************************************************************************
*
* ZERO TASK QUEUE
*
* ***************************************************************************
INITQ    RTL   U10,TSKQ                    CLEAR TASK Q
         BO    INITUDL               TASK Q IS EMPTY
         B     INITQ
* ***************************************************************************
INITUDL  EQU   *
* ***************************************************************************
*
*  INITIALIZE UDL FOR SVC 15 TRAP
*
* ***************************************************************************
         LA    U10,TSKQ                    TASK QUEUE ADDR
         ST    U10,UDL.TSKQ
         LI    U10,TSWNEW                  SVC 15 'Q' ENTRY ENABLE,TASK Q
         ST    U10,UDL.TSKN                TASK Q SERVICE,NEW TSW
         LA    U10,TRAPH             TRAP HANDLER
         ST    U10,UDL.TSKN+RTNADDR        RETURN ADDR FROM TRAPWAIT
         SVC   9,INITTSW             INITIALIZE NEW TSW
* ***************************************************************************
*
* INITIALIZE DCB FOR SPECIAL CHARACTER RECOGNITION
*
* ***************************************************************************
BEGIN    EQU   *
* ***************************************************************************
         SVC   15,SPCCRT
         BTC   F,DONE                      CONDITION CODE ERROR
         LH    U15,SPCCRT+SVC15,ST
         THI   U15,ERROR                   CHECK FOR ERROR
         BNZ   ERRTN
         EJECT
*                  EXAMPLE1  SVC15  OUTPUT EXECUTOR
* ***************************************************************************
*
* EXECUTE SVC 15 WRITE TO DISPLAY FORM
*
* ***************************************************************************
FORMWRT  EQU   *
* ***************************************************************************
         LA    U1,CRTWRITE                 PBLK ADDR
         BAL   U2,EXECIO
         LH    U0,SVC15,ST(U1)
         THI   U0,ERROR              ERROR STATUS?
         BNZ   ERRIO                 YES
         SPACE 5
*                  EXAMPLE1  SVC15  INPUT EXECUTOR
* ***************************************************************************
*
* EXECUTE SVC 15 READ
*
* ***************************************************************************
FORMRD1  EQU   *
* ***************************************************************************
         LA    U1,READ1                    PBLK ADDR
         BAL   U2,EXECIO
         EJECT
*                  EXAMPLE1  INPUT  PROCESSOR
*
* CHECK FOR XMIT STATUS FOLLOWING READ
*
         LH    U0,SVC15,ST(U1)
```

```
         THI   U0,ERROR
         BNZ   ERRIO   ERROR STATUS
         THI   U0,XMSTAT           XMIT STAUTS?
         BNZ   FORMRD2                  XMIT STATUS FOUND CONTINUE
*
*  XMIT  STATUS NOT RECEIVED**READ INPUT FROM CRT***
*        NON-EDITING INPUT MODE
*
*
*  NON-EDITING INPUT  FROM  EDITING CRT
*        USER  INPUTS INFORMATION AND TERMINATES ALL INPUT WITH CTLC
*
*
*
*
         LA    U1,READ1                 READ INPUT IN NON-EDIT MODE
         BAL   U2,EXECIO
         LH    U0,SVC15.ST(U1)
         THI   U0,ERROR                 ERROR?
         BNZ   ERRIO                    YES
         B     DONE
* ***********************************************************************
FORMRD2  EQU   *
* ***********************************************************************
         LA    U1,CRTREAD               PBLK ADDR
         BAL   U2,EXECIO
         LH    U0,SVC15.ST(U1)
         THI   U0,ERROR
         BNZ   ERRIO               ERROR STATUS
*
*        PRINT INFORMATION READ
*       WRITE INFORMATION TO LIST DEVICE LU3
*
         LA    U7,BUF3
         BAL   U2,PRN.FORM
         B     FORMRD1                  READ NEXT FORM
         EJECT
*                 EXAMPLE1  TRAP  HANDLER
* ***********************************************************************
* THIS CODE PROCESSES  SVC 15 TRAPS
*
* ***********************************************************************
TRAPH    EQU   *
* ***********************************************************************
         RTL   U10,TSKQ                 REMOVE QUEUE ENTRY
         BO    TRAPERR             TASK Q IS EMPTY,ERROR
         LR    U11,U10
         NI    U10,PBLKMASK
         CI    U11,REASMASK        REASON CODE
         CLI   U11,TERMTRAP             TERM TRAP?
         BNE   ERRIO               NO,ERROR
         CR    U1,U10              VALID PBLK ADDR?
         BER   U2                       RETURN TO MAIN PROGRAM
         B     ERRIO                    ERROR, PRINT MESSAGE
         SPACE 5
*          EXAMPLE1  ERROR MESSAGE EXECUTOR
* ***********************************************************************
*
* I/O ERROR PRINT ERROR MESSAGE: IO ERRXXYY  XX=I/O ERROR CODE YY=LU#
*
* ***********************************************************************
ERRIO    EQU   *
* ***********************************************************************
         SVC   2,UNPACK                 UNPACK STAT#,LU#
         SVC   1,MSGIO                  I/O ERROR MSG
         B     ERRTN
```

```
* ***************************************************************
*
* ERROR STATUS - PAUSE
*
* ***************************************************************
ERRTN    EQU   *
* ***************************************************************
         SVC   2,PAUSE
         B     EXAMPLE1          RESTART TASK
* ***************************************************************
*
* TASK Q IS EMPTY  'ERROR'
*
* ***************************************************************
TRAPERR  EQU   *
* ***************************************************************
         SVC   1,TSKQEMP
         B     ERRTN
         SPACE 5
*              EXAMPLE1  TERMINATE TASK
* ***************************************************************
*
DONE     EQU   *
*
* RESET DCB TO SPCHAR DEFAULT
*
* ***************************************************************
         SVC   15,SPCINIT               SET SPECIAL CHAR TO DEFAULT
         SVC   3,0                      END OF JOB
* ***************************************************************
         EJECT
         EJECT
*                SUBROUTINES
*  SUBROUTINE  TO  EXECUTE  I/O
*  REGS: U1=PRLK ADDR    U2=RTN ADDR
*
*
EXECIO   EQU   *
         SVC   15,0(U1)                 EXECUTE I/O
         BTC   F,ERRTN                  ERROR
         SVC   9,TRAPWAIT          WAIT FOR I/O TERMINATION
         B     ERRIO                    SHOULD NEVER RTN HERE,ERROR
*
* THIS SUBROUTINE PRINTS THE FORM ON LU3
*  THE FORM IS INPUT IN EDITNG MODE
*
*  ENTRY:                                   RETURN: U2=RETURN ADDR
*        U7=ADDR OF INPUT BUFFER
*
*    U11,U12,U13,U14,U15 ARE DESTROYED
*
PRN.FORM EQU   *
         LI    U15,SPACE
PRN.CONT EQU   *
         LI    U14,CHARS              #NO.OF CHARS PER LINE
         XR    U11,U11
         XR    U13,U13
         LA    U6,PRNST
CLEAR    EQU   *
         STB   U15,0(U6,U11)          INIT PRINT BUFFER TO SPACES
         AIS   U11,1
         SIS   U14,1
         BNZ   CLEAR
PRN.LOOP EQU   *
         LB    U12,0(U7,U13)
         STB   U12,0(U6,U13)          MOVE CHAR TO PRINT BUFFER
```

```
         CLHI  U12,CR                    IS CHAR CR?
         BE    PRN.LINE                  YES, PRINT LINE
         CLHI  U12,ETX                   IS CHAR ETX?
         BE    PRN.LINE                  YES, PRINT LAST LINE
         AIS   U13,1                     INCREM BYTE INDEX
         B     PRN.LOOP
PRN.LINE EQU   *
         SVC   1,SVC1PRNT
         LH    U0,SVC1PRNT+SVC1.STA
         BNZ   ERRIO
         AIS   U13,1
         AR    U7,U13
         CLHI  U12,ETX                   LAST LINE?
         BNE   PRN.CONT
         BR    U2                        YES, RETURN
         EJECT
         EJECT
*                PARAMATER  BLOCKS
* ************************************************************
         ALIGN ADC
INITTSW  DC    TSWNEW
         DAC   BEGIN                     RTN ADDR FROM SVC 9
*
*
*  SVC 15  SPECIAL CHARACTER  PARAMETER BLOCK
*
         ALIGN ADC
SPCCRT   EQU   *
         DB    0,1                       FC,LU
         DB    0,0                       STATUS
         DAC   SPCHRCMD
         DCX   0,0                       LLR,LLW
         DAC   BUF0+PARAM
         ALIGN ADC
SPCHRCMD EQU   *
         DCX   4E                        SPECIAL CHARACTER COMMAND
         ALIGN ADC
BUF0     EQU   *
         DCX   4004,4004           READ SPCHAR:ETX,XMIT
*                                  WRITE SPCHAR:ETX,XMIT
* ************************************************************
*
*  SVC 15  EDITING CRT WRITE PARAMETER BLOCK
*
         SPACE 2
         ALIGN ADC
CRTWRITE EQU   *
         DB    X'10',1             FC,LU
         DCX   0                         STATUS
         DAC   DCWWRT                    CMD NUMBER,DCW ADDR
         DCX   0                         LLR
         DCX   0                         LLW
         DAC   BUF6+INDIRECT
         DAC   BUF7+INDIRECT
         SPACE 2
         ALIGN ADC
DCWWRT   EQU   *
         DCX   8004                      WRITE BUFFER-BLINKING MSG
         DCX   0004                      WRITE BUFFER-FORM
         ALIGN ADC
BUF6     EQU   *
  DC     2(BUF6END-BUF6DATA),X'0'    AVAIL BYTES,BYTES USED
BUF6DATA EQU   *
         DB    PC,CM,LK,PC,SPS,SBS
         DB    C'FILL IN THE FORM AND XMIT TO PROCESSOR'
         DB    EBS,ETX
```

```
BUF6END  EQU   *-1
         ALIGN ADC
BUF7     EQU   *                          FORMS BUFFER=BUF7
         DC    Z(BUF7END-BUF7DATA),X'0'  AVAIL BYTES,BYTES USED
BUF7DATA EQU   *
         DB    LF,LF,CR
         DB    C'NAME: '
         DB    EPS
         NLIST
         DO    20
         DB    X'20'
         LIST
         DB    SPS
         DB    LF,CR
         DB    C'STREET ADDRESS: '
         DB    EPS
         NLIST
         DO    20
         DB    X'20'
         LIST
         DB    SPS
         DB    LF,CR
         DB    C'CITY: '
         DB    EPS
         NLIST
         DO    20
         DB    X'20'
         LIST
         DB    SPS
         DB    LF,CR
         DB    C'STATE,ZIP CODE: '
         DB    EPS
         NLIST
         DO    20
         DB    X'20'
         LIST
         DB    SPS
         DB    LF,CR
         DB    C'ACCT.NO: '
         DB    EPS,UNLK,PC
BUF7END  EQU   *-1
* ***************************************************************************
         ALIGN ADC
READ1    EQU   *
         DB    X'10',1              FC,LU
         DCX   0                          STATUS
         DAC   DCWRD1                     CMD NO,RD1 CMD ADDR
         DCX   0                          LLR
         DCX   0                          LLW
         DAC   BUF8+DIRECT
         DAC   BUF8END+DIRECT
         SPACE 2
         ALIGN ADC
DCWRD1   EQU   *
         DCX   0002                 READ BUFFER CMD
         ALIGN ADC
*
BUF8     EQU   *
         NLIST
         DO    CHARS*MAXLINES
         DB    X'20'
         LIST
BUF8END  EQU   *-1
* ***************************************************************************
*
*
```

```
*  SVC 15 EDITING CRT    READ PARAMETER BLOCK
*
          SPACE 2
          ALIGN ADC
CRTREAD   EQU   *
          DB    X'10',1              FC,LU
          DCX   0                          STATUS
          DAC   DCWRD                      CMD NUMBER,DCW ADDR
          DCX   0                          LLR
          DCX   0                          LLW
          DAC   BUF1+PARAM
          DAC   BUF2+INDIRECT
          DAC   BUF3+DIRECT
          DAC   BUF3END+DIRECT
          DAC   BUF4+PARAM
          DAC   BUF5+PARAM
*
*  SVC 15  DRIVER COMMAND WORD(DCW) READ CMD CHAIN PARAMETER BLOCK
*
          ALIGN ADC
DCWRD     EQU   *
          DCX   8016                       MODE-RCMD-CHAINED
          DCX   8004                       WRITE BUFFER COMMAND-CHAINED
          DCX   8002                       READ BUFFER COMMAND-CHAINED
          DCX   8014                  WRITE2 COMMAND CHAINED
          DCX   0016                       MODE-RCMD-UNCHAINED
*
*
*
          ALIGN ADC
BUF1      EQU   *
          DB    NOECHO                     NO ECHOPLEX
          ALIGN ADC
BUF2      EQU   *
          DCX   3,0                 BYTES AVAIL,BYTES USED
          DB    LK,PC,XMWR                 LOCK KEYBOARD,POS CURSOR,XMIT
          ALIGN ADC
*
*  INIT BUFFER TO CHARS*MAXLINES BLANKS
*
BUF3      EQU   *
          NLIST
          DO    CHARS*MAXLINES
          DB    X'20'
          LIST
BUF3END   EQU   *-1
          ALIGN ADC
BUF4      EQU   *
          DB    PC,UNLK             POS CURSOR,UNLOCK KEYBOARD
          ALIGN ADC
BUF5      EQU   *
          DB    ECHO                       ECHOPLEX
          ALIGN ADC
UNPACK    EQU   *                   U0=ERROR STATUS
          DB    X'00'+4,6
          DAC   BFUNPK
*
*  PBLK  TO PRINT I/O ERROR MESSAGE
* ON LU7
          ALIGN ADC
SPCINIT   EQU   *
          DB    0,1                        FC,LU
          DB    0,0                        STATUS
          DAC   SPCCMD                     COMMAND
          DCX   0,0                        LLR,LLW
          DAC   BUF9+PARAM
```

```
          ALIGN ADC
SPCCMD    EQU   *
          DCX   4E                        SPECIAL CHARACTER COMMAND
          ALIGN ADC
BUF9      EQU   *
          DCX   F9ED,6004                 READ/WRITE SPECIAL CHARACTER
*                                         DEFAULT
          ALIGN ADC
MSGIO     EQU   *
          DC    X'2807'                   WRITE TO LU#7
          DS    2
          DAC   MSGSTART
          DAC   MSGEND
          DS    4                         RANDOM ADDR
          DS    4                         LENGTH OF DATA TRANSFER
          ALIGN ADC
MSGSTART  DC    C'IO ERR  '
BFUNPK    DS    4
MSGEND    EQU   *-1
TRAPWAIT  DC    TRPWSV15                  TSW FOR SVC 9 TRAPWAIT
          DCF   0                         RTN LOCATION, SVC 9
          ALIGN ADC
TSKQ      DLIST QSIZE
          ALIGN ADC
PAUSE     DC    X'0001'             SVC2 PAUSE
          ALIGN ADC
TSKQEMP   DC    X'2807'
          DC    2
          DAC   TQSTRT
          DAC   TQEND
          DS    4
          DS    4
TQSTRT    DC    C'SVC15 TASK Q EMPTY   ERROR'
TQEND     EQU   *-1
          ALIGN ADC
SVC1PRNT  EQU   *
          DB    X'28',3                   WRITE LU3
          DB    0,0                       STATUS
          DAC   PRNST                     BUFFER START
          DAC   PRNEND                    BUFFER END
          DAS   1                         RANDOM ADDR
          DAS   1                         LENGTH OF XFER
SVC1END   EQU   *-1
          ALIGN ADC
* INIT BUFFER TO BLANKS(80 CHARS)
PRNST     EQU   *
          DLIST
          DO    CHARS
          DB    X'20'
          LIST
PRNEND    EQU   *-1
          END
```

APPENDIX D
SAMPLE MODEL 1200 PROGRAM

This appendix contains a sample program which uses the Model 1200, plus illustrations of screen formats output to a print device connected to the Model 1200, and hex dumps of data read from the Model 1200. A description of these items follows:

- Sample Program - See the comments included within the program for detailed description. All screen format printers were called by this program, and all hex dumps show data received by the program.

- Read Template (Figure D-1) - This is a printout of the fill-in-the-blanks form received by the Model 1200.

- Read After Initial Input (Figure D-2) - This is a printout of the screen after data was input by the operator.

- Read After Modification (Figure D-3) - This is a printout of the screen after data was modified by the operator. The only change is the middle initial of the input name.

- Read All (Figure D-4) - This is a hex dump of data received by a read-all of the screen image shown in Figure D-2.

- Read Unprotected (Figure D-5) - This is a hex dump of data received by a read-unprotected of the screen image shown in Figure D-2.

- Read Modified (Figure D-6) - This is a hex dump of data received by a read-modified of the screen image shown in Figure D-3.

PROG= DEMOCRT ASSEMBLED BY CAL 03-066R04-01 (32-BIT)

```
                                3  *          THE PURPOSE OF THIS PROGRAM IS
                                4  *       TO PROVIDE A SAMPLE OF SOME SIMPLE
                                5  *       OPERATIONS WHICH MAKE USE OF THE
                                6  *       ITAM PET1200 TERMINAL MANAGER. A
                                7  *       FORMAT IMAGE IS WRITTEN TO THE SCREE
                                8  *       AND READ BACK IN READ ALL, READ
                                9  *       UNPROTECTED AND READ MODIFIED FORMAT
                               10  *
0000001I  4300 4000 0D4CI       11          B       START
                               12  *
```


EQUATES + BUFFERS

```
                                15  *          COPY  REGS          (LINE NOT LISTED)
0000061I                       111          LIST

                               113  *          SPECIAL CHARACTER EQUATES

          0000 001B            115  ESC       EQU   X'1B'        ESCAPE CHARACTER
          0000 0021            116  ATRB      EQU   X'21'        ATTRIBUTE CHARACTER FOLLOWS
          0000 0003            117  ETX       EQU   X'03'        END OF TEXT SENTINAL

          0000 0000            119  NORMAL    EQU   X'00'        NORMAL TEXT
          0000 0002            120  PROTECT   EQU   X'02'        PROTECTED FIELD
          0000 0008            121  REVIDEO   EQU   X'08'        REVERSE VIDEO DISPLAY
          0000 0004            122  LIGHTINT  EQU   X'04'        LIGHT INTENSITY
          0000 0040            123  BLINK     EQU   X'40'        BLINKING

                               125  *          OUTPUT BUFFER

0000081I                       127          ALIGN  ADC
          0000 0081I           128  OUTBUF    EQU   *
0000081I  1B210A               129  NAME      DB    ESC,ATRB,PROTECT+REVIDEO
00000BI   504C4541             130          DB    C'PLEASE TYPE NAME'
          53452054
          59504520
```

EQUATES + BUFFERS

```
                4E414D45
00001BI 1B2100          131             DB      ESC,ATRB,NORMAL
                        132     *       DO      240+NAME-*+4            (LINE NOT LISTED)
                        133     *       DB      C' '                    (LINE NOT LISTED)
0000FCI                 137             LIST
0000FCI 1B210A          138     ADDRESS DB      ESC,ATRB,PROTECT+REVIDEO
0000FFI 504C4541        139             DB      C'PLEASE TYPE ADDRESS'
        53452054
        59504520
        41444452
        455353
000112I 1B2100          140             DB      ESC,ATRB,NORMAL
                        141     *       DO      400+ADDRESS-*+4     (LINE NOT LISTED)
                        142     *       DB      C' '                    (LINE NOT LISTED)
0002901                 146             LIST
0002901 1B210E03        147     FILLER  DB      ESC,ATRB,PROTECT+REVIDEO+LIGHTINT,ETX
        0000 0293I      148     OUTBUFE EQU     *-1
```

INPUT BUFFERS

```
0002941                 152             ALIGN   ADC
0002941                 153     INBUF   DS      2500            INPUT BUFFER TO RECEIVE FROM SCREEN
        0000 0C57I      154     INBUFE  EQU     *-1
000C581                 155             ALIGN   ADC
000C581                 156     DUMBUF  DS      4               DUMMY BUFFER
        0000 0C5BI      157     DUMBUFE EQU     *-1
```

PARAMATER BLOCKS

```
000C5CI                     160            ALIGN  ADC
          0000 0C5CI        161  CLEARMEM  EQU    *            CLEAR SCREEN MEMORY P-BLOCK
000C5CI   2901              162            DCX    2901
000C5EI   0000              163            DCX    0
000C60I   0000 0C58I        164            DAC    DUMBUF
000C64I   0000 0C58I        165            DAC    DUMBUF
000C68I   0000 0000         166            DAC    0
000C6CI   0000 0000         167            DAC    0
000C70I   2000 000E         168            DC     Y'2000000E'


000C74I                     170            ALIGN  ADC
          0000 0C74I        171  CLEARTAB  EQU    *            CLEAR TABS P-BLOCK
000C74I   2901              172            DCX    2901
000C76I   0000              173            DCX    0
000C78I   0000 0C58I        174            DAC    DUMBUF
000C7CI   0000 0C58I        175            DAC    DUMBUF
000C80I   0000 0000         176            DAC    0
000C84I   0000 0000         177            DAC    0
000C88I   2000 0015         178            DC     Y'20000015'


000C8CI                     180            ALIGN  ADC
          0000 0C8CI        181  WRITESCN  EQU    *            WRITE OUTPUT BUFFER TO SCREEN
000C8CI   2901              182            DCX    2901
000C8EI   0000              183            DCX    0
000C90I   0000 0008I        184            DAC    OUTBUF
000C94I   0000 0293I        185            DAC    OUTBUFE
000C98I   0000 0000         186            DAC    0
000C9CI   0000 0000         187            DAC    0
000CA0I   2000 0008         188            DC     Y'20000008'


000CA4I                     190            ALIGN  ADC
          0000 0CA4I        191  CLEARMDT  EQU    *            CLEAR MODIFIED DATA FLAGS
000CA4I   2901              192            DCX    2901
000CA6I   0000              193            DCX    0
000CA8I   0000 0C58I        194            DAC    DUMBUF
000CACI   0000 0C58I        195            DAC    DUMBUF
000CB0I   0000 0000         196            DAC    0
```

PARAMATER BLOCKS

```
000CB41  0000 0000            197            DAC     0
000CB8I  2000 0019            198            DC      Y'20000019'



000CBCI                       200            ALIGN   ADC
         0000 0CBCI           201  REQST2SD  EQU     *               REQUEST TO SEND
000CBCI  4901                 202            DCX     4901
000CBEI  0000                 203            DCX     0
000CC0I  0000 0C58I           204            DAC     DUMBUF
000CC4I  0000 0C58I           205            DAC     DUMBUF
000CC8I  0000 0000            206            DAC     0
000CCCI  0000 0000            207            DAC     0
000CD0I  2000 0001            208            DC      Y'20000001'



000CD4I                       210            ALIGN   ADC
         0000 0CD4I           211  READALL   EQU     *               READ ALL
000CD4I  4901                 212            DCX     4901
000CD6I  0000                 213            DCX     0
000CD8I  0000 0294I           214            DAC     INBUF
000CDCI  0000 0C57I           215            DAC     INBUFE
000CE0I  0000 0000            216            DAC     0
000CE4I  0000 0000            217            DAC     0
000CE8I  2000 0004            218            DC      Y'20000004'



000CECI                       220            ALIGN   ADC
         0000 0CECI           221  READMOD   EQU     *               READ MODIFIED FIELDS
000CECI  4901                 222            DCX     4901
000CEEI  0000                 223            DCX     0
000CF0I  0000 0294I           224            DAC     INBUF
000CF4I  0000 0C57I           225            DAC     INBUFE
000CF8I  0000 0000            226            DAC     0
000CFCI  0000 0000            227            DAC     0
0000001  2000 0003            228            DC      Y'20000003'



0000D4I                       230            ALIGN   ADC
         0000 0D04I           231  READUNP   EQU     *               READ UNPROTECTED FIELDS
0000D4I  4901                 232            DCX     4901
0000D6I  0000                 233            DCX     0
0000D8I  0000 0294I           234            DAC     INBUF
```

PARAMATER BLOCKS

```
000D0C1  0000 0C57I      235          DAC     INBUFE
000D101  0000 0000       236          DAC     0
000D141  0000 0000       237          DAC     0
000D181  2000 0002       238          DC      Y'20000002'



000D1C1                  240          ALIGN   ADC
         0000 0D1C1      241  PRINT   EQU     *              PRINT SCREEN
000D1C1  2901            242          DCx     2901
000D1E1  0000            243          DCx     0
000D201  0000 0C581      244          DAC     DUMBUF
000D241  0000 0C581      245          DAC     DUMBUF
000D281  0000 0000       246          DAC     0
000D2C1  0000 0000       247          DAC     0
000D301  2000 001B       248          DC      Y'2000001B'



000D341                  250          ALIGN   ADC
         0000 0D341      251  SAVE    EQU     *              SAVE SCREEN INPUT ON DISC OR TAPE
000D341  2802            252          DCx     2802
000D361  0000            253          DCx     0
000D381  0000 0294I      254          DAC     INBUF
000D3C1  0000 0C57I      255          DAC     INBUFE
000D401  0000 0000       256          DAC     0
000D441  0000 0000       257          DAC     0
000D481  0000 0000       258          DAC     0
```

MAINLINE PROGRAM

```
        0000 0D4CI         START    EQU    *                 START OF SAMPLE PROGRAM CODE
                           *
                           *                        CLEAR SCREEN MEMORY, CLEAR TABS,
                           *                            WRITE FORM TO SCREEN,
                           *                            AND PRINT SCREEN...
                           *
000D4CI  E110 FF0C                  SVC    1,CLEARMEM
000D50I  E110 FF20                  SVC    1,CLEARTAB
000D54I  E110 FF34                  SVC    1,WRITESCN
000D58I  E110 FFC0                  SVC    1,PRINT
                           *
                           *                        PERFORM REQUEST TO SEND TO
                           *                             SEE WHEN OPERATOR INPUTS
                           *                             DATA, READ ENTIRE SCREEN,
                           *                             PRINT SCREEN & SAVE INPUT...
                           *
000D5CI  4180 4000 0DA0I            BAL    R8,CLEARBUF
000D62I  E110 FF56                  SVC    1,REQST2SD
000D66I  E110 FF6A                  SVC    1,READALL
000D6AI  E110 FFAE                  SVC    1,PRINT
000D6EI  E110 FFC2                  SVC    1,SAVE
                           *
                           *                        READ UNPROTECTED AREAS
                           *                             ON SCREEN & SAVE
                           *                             INPUT...
                           *
000D72I  4180 4000 0DA0I            BAL    R8,CLEARBUF
000D78I  E110 FF88                  SVC    1,READUNP
000D7CI  E110 FFB4                  SVC    1,SAVE
                           *                        CLEAR MODIFIED DATA TAGS,
                           *                             PERFORM REQUEST TO
                           *                             SEND, READ MODIFIED
                           *                             FIELDS ON SCREEN, PRINT
                           *                             SCREEN AND SAVE INPUT...
                           *
000D80I  E110 FF20                  SVC    1,CLEARMDT
000D84I  4180 4000 0DA0I            BAL    R8,CLEARBUF
000D8AI  E110 FF2E                  SVC    1,REQST2SD
000D8EI  E110 FF5A                  SVC    1,READMOD
000D92I  E110 FF86                  SVC    1,PRINT
000D96I  E110 FF9A                  SVC    1,SAVE
                           *
                           *                        GO TO END OF JOB
                           *
000D9AI  E130 0000                  SVC    3,0



000DA0I                             ALIGN  ADC
         0000 0DA0I        CLEARBUF EQU    *                 CLEAR INPUT BUFFER SUBROUTINE
000DA0I  F8F0 0000 0000             LI     R15,0
000DA6I  E6E0 F4EA                  LA     R14,INBUF
000DAAI  50FE 0000         CLEARLP  ST     R15,0(R14)
000DAEI  FAE0 0000 0004             AI     R14,ADC
000DB4I  F9E0 0000 0C57I            CI     R14,INBUFE
000DBAI  4280 FFEC                  BL     CLEARLP
000DBEI  0308                       BR     R8
000DC0I                             END
```

2906

```
PLEASE TYPE NAME
PLEASE TYPE ADDRESS
```

Figure D-1 Read Template

```
PLEASE TYPE NAME JOHN P. TESTER
PLEASE TYPE ADDRESS 2 INTERDATA PLACE
                    TINTON FALLS, NEW JERSEY
```

Figure D-2 Read After Initial Input

```
PLEASE TYPE NAME JOHN A. TESTER
PLEASE TYPE ADDRESS 2 INTERDATA PLACE
                    TINTON FALLS, NEW JERSEY
```

Figure D-3 Read Modified Screen Image

2536

```
1B21 0A50 4C45 4153 4520 5459 5045 204E 414D 451B 2120 4A4F 484E 2050 2E20 5445
5354 4552 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 1B21 0A50 4C45 4153 4520 5459
5045 2041 4444 5245 5353 1B21 2032 2049 4E54 4552 4441 5441 2050 4C41 4345 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2054 494E
544F 4E20 4641 4C4C 532C 204E 4557 204A 4552 5345 5920 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 1B21 0E00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 03
```

Figure D-4 Read All

2537

```
1D4A 4F48 4E20 502E 2054 4553 5445 5220 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 201D
3220 494E 5445 5244 4154 4120 504C 4143 4520 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 5449 4E54 4F4E 2046 414C 4C53 2C20 4E45
5720 4A45 5253 4559 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2003
```

Figure D-5 Read Unprotected

2538

```
0120 391D 2032 4A4F 484E 2041 2E20 5445 5354 4552 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020
2020 2020 03
```

Figure D-6 Read Modified

INDEX

L

M N O

P

Q

R

S

T U V W

X Y Z

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

No postage necessary if mailed in U.S.A.

6434

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE