Rainbow™

Technical Documentation

digital

QV069–GZ

Rainbow™ 100+/100B

Technical Documentation

digital equipment corporation

First Printing, June 1984
Second Printing, April 1985

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

CP/M and CP/M–86 are registered trademarks of Digital Research Inc.
CP/M–80 is a trademark of Digital Research Inc.

MS is a trademark of Microsoft Corporation.

Z80 is a registered trademark of Zilog, Inc.

8088 is a registered trademark of Intel Corporation.

The following are trademarks of Digital Equipment Corporation:

digital™
DEC
DECmate
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
RSX
UNIBUS
VAX
VMS
VT
Work Processor

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

Printed in U.S.A.

Rainbow™ 100+/100B Technical Documentation

Recommended Documents

Contains a list of additional reading materials.

Rainbow 100+/100B System Specification

This specification describes the hardware and the firmware for the Rainbow 100+/100B computers. Discussions of each of the standard and optional components of the Rainbow 100B system unit are included. The system motherboard description discusses the dual-processor architecture, local bus structures, standard video and communications interfaces, memory and I/O maps, and register bit-formats. The VT102 terminal emulation firmware description includes the variations from the standard VT102. Operating system support firmware is also discussed.

Rainbow 100+/100B Terminal Emulation Manual

This document discusses each terminal key character, processing of received characters, and use of control functions. The appendixes list character codes and control functions along with the ANSI code extension techniques for escape and control sequences. There is a description of the Rainbow computer and VT102 terminal differences and a series of international keyboard illustrations.

Rainbow 100 Technical Manual

This manual describes how the Rainbow computer operates. It includes explanations of the features, capabilities, system architecture, and technical characteristics of the Rainbow computer.

Rainbow Color/Graphics Option Programmer's Reference Guide

This guide is written for the experienced systems programmer. The information contained in this guide is operating system independent; however, it is specific to 8088-based software and hardware. It is divided into the following parts:

Operating Principles Gives an overview of the Graphics Option hardware and software.

Programming Guidelines Discusses such aspects of programming as initialization, various write and read operations, notes, and timing considerations.

Reference Materials Describes the option's registers, buffers, masks, and maps, and the GDC register contents and supported GDC commands.

The two appendixes contain a specifications summary and a block diagram of the option.

INTEL Specifications

The *8274 Multi-protocol Serial Controller (MPSC)* and the *8237A/8237A–5 High Performance Programmable DMA Controller* are intended for communications programmers or engineers responsible for integrating the controllers into hardware designs. They describe the controllers' functions and general operations, and provide detailed programming specifications and hardware interface information.

The synchronous and asynchronous communication documents are an aid for the less experienced programmer who is implementing applications that use either synchronous or asynchronous communication techniques. They provide information on how to use MPSC in either synchronous or asynchronous modes. They also include a number of examples and sample programs to aid the programmer in developing communications applications.

HOW TO ORDER ADDITIONAL DOCUMENTATION

If you want to order additional documentation by phone:

And you live in:	Call:	Between the hours of:
New Hampshire, Alaska or Hawaii	603-884-6660	8:30 AM and 6:00 PM Eastern Time
Continental USA or Puerto Rico	1-800-258-1710	8:30 AM and 6:00PM Eastern Time
Canada (Ottawa-Hull)	613-234-7726	8:00 AM and 5:00 PM Eastern Time
Canada (British Columbia)	1-800-267-6146	8:00 AM and 5:00 PM Eastern Time
Canada (all other)	112-800-267-6146	8:00 AM and 5:00 PM Eastern Time

If you want to order additional documentation by direct mail:

And you live in: **Write to:**

USA or Puerto Rico

DIGITAL EQUIPMENT CORPORATION
ATTN: Peripherals and Supplies Group
P.O. Box CS2008
Nashua, NH 03061

NOTE: Prepaid orders from Puerto Rico must be placed with the local DIGITAL subsidiary (Phone 809-754-7575)

Canada

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: P&SG Business Manager

Other than USA, Puerto Rico or Canada

DIGITAL EQUIPMENT CORPORATION
Peripherals and Supplies Group
P&SG Business Manager
c/o Digital's local subsidiary or approved distributor

TO ORDER MANUALS WITH EK PART NUMBERS WRITE OR CALL

P&CS PUBLICATIONS
Circulation Services
10 Forbes Road
NR03/W3
Northboro, Massachusetts 01532
(617)351-4325

Rainbow™ Recommended Documents

Other Technical Documentation Kits

1. Rainbow CP/M–86/80 V2.0 Technical Documentation (QV067–GZ)
2. Rainbow MS–DOS V2.05 Technical Documentation (QV068–GZ)

Additional Documents

1. Letterprinter 100 User Documentation Package (EK–LP100–UG)

   Includes:

   Letterprinter 100 Operator Guide
   LA100-Series Programmer Reference Manual
   Letterprinter 100 Installation Guide
   Letterprinter 100 Operator and Programmer Reference Card

2. Letterwriter 100 User Documentation Package (EK–LW100–UG)

   Includes:

   Letterwriter 100 Operator Guide
   Letterwriter 100 Installation Guide
   LA100-Series Programmer Reference Manual

3. Installing and Using the LQP02 Printer (AA–L662B–TK)
4. Installing and Using the LA50 Printer (EK–0LA50–UG)

   Includes:

   LA50 Printer Programmer Reference Manual

5. Rainbow 100 Extended Communications Option Programmer's Reference Guide (AA–V172A–TV)
6. PC100 Rainbow 100B System Unit IPB (EK–SB100–IP)
7. VT102 Video Terminal User Guide (EK–VT102–UG)
8. CP/M Operating System Manual (AA–X637A–TV)

Rainbow™ 100+/100B

System Specification

digital equipment corporation

First Printing, June 1984

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

CP/M and CP/M–86 are registered trademarks of Digital Research Inc.
CP/M–80 is a trademark of Digital Research Inc.

Z80 is a registered trademark of Zilog, Inc.

8088 is a registered trademark of Intel Corporation.

The following are trademarks of Digital Equipment Corporation:

digital™

DEC
DECmate
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
RSX
UNIBUS
VAX
VMS
VT
Work Processor

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

Printed in U.S.A.

TABLE OF CONTENTS/REVISION STATUS

Subhead	Title	Page
	Title Page	1
	Table of Contents/Revision Status	2
1	SYSTEM OVERVIEW	4
2	BASE SYSTEM	4
3	SYSTEM MODULES	5
3.1	MOTHER BOARD	5
3.2	OPTION MODULES	38
4	RX5Ø DRIVE	45
4.1	GENERAL DESCRIPTION	45
4.2	DRIVE CHARACTERISTICS	45
4.3	TRACK FORMAT	45
4.4	HEADER FORMAT	46
5	PC1ØØ-B FIRMWARE	47
5.1	PRODUCT GOALS	47
5.1.1	Functional Anomalies	48
5.2	PERFORMANCE	51
5.3	NON-GOALS	51
5.4	GENERAL	51
5.4.1	Text Strings	53
5.4.2	Character Sets	53
5.5	START-UP/SHUT-DOWN/RESET	61
5.5.1	Power-Up Initialization	61
5.5.2	Selection of Keyboard/Language	61
5.5.3	Power-Off	62
5.5.4	Hardware Resetting	62
5.5.5	RAM Parity Error	62
5.6	MEMORY-MAPPED VIDEO ACCESS SERVICE	63
5.7	KEY ACCESS SERVICE	64
6	FUNCTIONAL DEFINITION	65
6.1	OPERATIONAL DESCRIPTION	65
6.2	CLUSTERING OF LANGUAGES	65
6.3	TRANSMITTED CHARACTERS	66
6.4	RECEIVED CHARACTER PROCESSING	78
6.5	VT1Ø2 CONTROL OF ATTACHED DEVICES	79
6.6	OTHER DIFFERENCES - TERMINAL VS CONSOLE MODES	80
7	EXTERNAL INTERACTIONS	84
7.1	OPERATING SYSTEM	84
7.2	STACK OVERHEAD	84
7.3	STACK USE BY APPLICATIONS	84
8	VT1Ø2	84
8.1	SET-UP	84
8.2	RESET TECHNIQUE	90

TABLE OF CONTENTS/REVISION STATUS (Continued)

Subhead	Title	Page
9	INTERFACE LAYER	90
9.1	SOFTWARE INTERRUPT TYPE 4Ø (DECIMAL)	90
1Ø	IMAGE OF Z8Ø RAM SPACE TO BE LOADED	99
11	BOOT LOADER TO READ TRACK Ø, SECTOR 1 OF FLOPPY	99
12	MENU SELECTION PROCESS	102
13	SYSTEM PARAMETER INFORMATION	103
14	DIAGNOSTIC AND MANUFACTURING SUPPORT	104
15	POWER SUPPLY	106
16	CONNECTOR OUTPUTS	108
16.1	VIDEO INTERFACE CONNECTOR	108
16.2	COMMUNICATIONS INTERFACE CONNECTOR	109
16.3	PRINTER PORT INTERFACE CONNECTOR	111
16.4	FLOPPY INTERFACE CONNECTOR	112
17	OPTIONS	113
18	CABLES	114
19	ENVIRONMENTAL	114
2Ø	RELIABILITY	114
21	PHYSICAL PACKAGING	114
22	VIDEO CHARACTER SET	115
23	APPLICABLE STANDARDS AND REGULATIONS	115

1 SYSTEM OVERVIEW

The PC1ØØ-B system is a low-cost, user-installable personal business computer used to run applications in the Fortune 1ØØØ marketplace. The PC1ØØ-B provides hardware and software support for both stand-alone processing and as a terminal emulation for DEC and other computer manufacturer's systems.

The PC1ØØ-B is designed to be used by users with no prior computer experience. Applications software include packages designed by third party software houses.

2 BASE SYSTEM

The base system consists of the following components:

2.1 BA25-C

The BA25-C is the system nucleus. The multi-box is the primary housing for the system, and encloses the following components:

a. PC1ØØ-B System Module - The basic intelligence of the system and provides the means for interconnection of all options. This module includes the following features:

   o 8Ø88 CPU
   o Z8ØA CPU
   o 64KB unshared dynamic memory
   o 64KB shared dynamic memory
   o 32 to 64KB ROM
   o 256 x 4 NVM
   o VT1ØØ compatible DCØ11, DCØ12 video electronics
   o Async/Bisync communications port
   o LA5Ø, LA1ØØ, LA12 Printer Port
   o LK2Ø1 Keyboard interface
   o RX5Ø Floppy controller
   o Option expansion capability
   o Extended communications
   o Color graphics
   o Extended memory (64 to 768KB)

b. RX5Ø Mini-Floppy Drive - A dual platter mini-floppy drive system which is the main storage area for the system.

c. H7842-D Power Supply - A 14Ø W switching regulator power supply with a switch selectable 1ØØ-12Ø/22Ø-24Ø V primary circuit.

2.2 VR2Ø1-A MONITOR

A 12-inch diagonal composite monitor that supplies video information to the user, as well as an interconnect means for the keyboard.

2.3 LK2Ø1-AA KEYBOARD

An ergonomic keyboard,supporting 1Ø5 keys. This is interconnected via a coiled cord terminated in a four-conductor telephone plug.

3 SYSTEM MODULES

3.1 MOTHER BOARD

The PC1ØØ-B includes a two-processor architecture based on the simultaneous operation of an 8Ø88 and a Z8ØA CPU. These CPUs operate from and transfer data through a shared block of 62KB of RAM. In addition to this block of shared memory, each processor has its own memory and peripheral circuitry.

Note

> Because the floppy controller module is an integral part of the system, it is included in the mother board section.

In addition to running application/user software, each processor supports a portion of the needed functions of the computer. The Z8ØA processor performs the functions required to read/write the floppy disks. The 8Ø88 handles the video output, keyboard I/O, printer port and the communications as well as any other options.

3.1.1 Mother Board Block Diagram

The Mother Board Block Diagram is shown in Figure 1.

3.1.2 8Ø88 System

The 8Ø88 microprocessor on the module controls nearly everything except the floppy disk. The 8Ø88 runs from a clock of 4.815 MHz and controls the following:

- Video
- Keyboard
- Printer
- Communication line
- Optional graphics board
- Optional extended communications board

```
    +----------------------INTERRUPT---------------------------------------+
    |                                                     +-------------+  |
+---------+     +------+     +----------+                 |  EXT. COMMS.|  |
|         |     |      |     |  SHARED  |-+----DMA--------|      or     |  |
|  Z80A   |-----|      |-----|  62 KB   | |  +------+     | WINCHESTER  |  |
|         |     |      |     |   RAM    |----|      |     |    DISK     |  |
|         |     |  8   |     +----------+ |  |      |     |   OPTION    |  |
+---------+     |      |     |   2 KB   | |  |      |     +-------------+  |
                |  B   |     | UNSHARED |----|  8   |                      |
                |  I   |     |   RAM    | |  |      |     +---------+      |
+---------+     |  T   |     +----------+ |  |  B   |     |         |      |
|  2 KB   |     |      |     |  64 KB   |-+  |  I   |-----|  8088   |------+
|UNSHARED |-----|  D   |     | UNSHARED |    |  T   |     |         |
|  RAM    |     |  A   |     |   RAM    |----|      |     |         |
+---------+     |  T   |     +----------+    |  D   |     +---------+
                |  A   |     |  32 to   |    |  A   |
                |      |     |  64 KB   |----|  T   |
+---------+     |  B   |     |   ROM    |    |  A   |     +---------+
|  RX50   |     |  U   |     +----------+    |      |     |         |
|  DISK   |-----|  S   |                     |  B   |-----|KEYBOARD |
|  DRIVE  |     |      |     +----------+    |  U   |     |   I/F   |
|  I/F    |     |      |     | OPTIONAL |    |  S   |     |         |
+---------+     |      |     |   RAM    |----|      |     +---------+
                |      |     |(64 KB to |    |      |
                +------+     | 768 KB)  |    |      |
                             +----------+    |      |     +---------+
                                             |      |     | ASYNC/  |
                             +----------+    |      |-----| BISYNC  |
                             | DC 11/12 |    |      |     | COMMS   |
                             |  VIDEO   |----|      |     +---------+
                             |CONTROLLER|    |      |
                             +----------+    |      |
                               |     |       |      |     +---------+
                               | +------+    |      |     |         |
                               | | 4 KB |    |      |-----| PRINTER |
                               | |SCREEN|----|      |     |  PORT   |
                               | | RAM  |    |      |     +---------+
                             +---+------+    |      |
                             |  4 KB    |    |      |
                             |ATTRIBUTE |----|      |     +---------+
                             |   RAM    |    |      |     |         |
                             +----------+    |      |-----|  COLOR  |
                                             |      |     |GRAPHICS |
                                             |      |     |         |
                                             +------+     +---------+
```

Figure 1. Mother Board Block Diagram

The 8088 also controls the Z80A's RESET line, as it can start/stop the Z80A at will. The clock time on the 8088 is approximately 208 nanoseconds. Contention from either the Z80A or refresh can cause wait states.

3.1.3 8088 Memory

The 8088 has several different types of memory available for its use:

a. 128KB dynamic memory (62KB shared)
b. 32 to 64KB ROM
c. 4KB video screen memory (static)
d. 4KB video attribute memory (static)
e. 256X4 NVM with shadow RAM
f. 64KB to 768KB optional unshared dynamic memory

3.1.3.1 Standard Memory - The 128 KB of standard memory is partitioned into a low order 64 KB bank and a high order 64 KB bank. 62 KB of the low order 64 KB bank is addressable by, and therefore accessible to, the Z80A processor. The Z80A is unable to address (and therefore can't modify) the first 2KB portion of this bank. Therefore, the 8088 keeps its interrupt vectors and some other information safe from being molested by a Z80A application.

The high order 64 KB bank of standard memory is accessible to the 8088 via the same memory bus as the lower 64 KB of standard memory. It is not addressable by the Z80A, and thus is not shared, but Z80A accesses to the low order 64K bank utilize the standard memory bus, causing the 8088 to incur wait states when 8088 access to the high order 64 KB bank of standard memory coincide with Z80A accesses to the low order 64 KB bank.

If there is no contention for the standard memory bus at the time of an 8088 access, no wait states are required for the cycle. If the bus is busy due to a refresh cycle, DMA cycle, or a Z80A memory cycle, which was initiated prior to the 8088's request, wait states will occur until the request can be filled.

Refresh has the highest priority for memory cycles. DMA has the second highest priority for memory cycles. The processors have the lowest priority. The 8088 has approximately equal priority with the Z80A.

No parity generaton/checking is implemented for the standard 128 KB memory.

3.1.3.2 ROM - There is 32 to 64KB of ROM (two sockets) on the module which is addressable by the 8088. The ROM contains Z80A code and 8088 code for diagnostics, bootstrap, and VT102 emulation. The code for the Z80A must be moved into shared memory by the 8088 in order to be executed by the Z80A. No wait states are required when the 8088 accesses this memory; however, because the circuitry assumes that all memory is dynamic RAM, wait states will be executed whenever refresh cycles are in progress. Supported ROMs are of the 27128/27256 pinout variety, with access times <= 450 nsec. Each ROM socket has a jumper associated with it, to select the 27128 (128 K bit) pinout or the 27256 (256 K bit) pinout. The default condition is with the jumper absent, selecting the 27128 type pinout.

3.1.3.3 4KB Screen Memory and 4KB Attribute Memory - There is screen and attribute memory available to the 8088 which allows it to control what is on the CRT display. This memory is available to the 8088 90 percent of the time. In the remaining ten percent, the DC011 and DC012 have access to this memory and prohibit the 8088 from access. Wait states to the 8088 occurs during refresh cycles and while the DC011 and DC012 are using the memory. The worst case time in which the 8Ø88 can be held in a wait state due to contention with the DCØ11 and DCØ12 is approximately 12Ø microseconds.

3.1.3.4 256 X 4 NVM With Shadow RAM - The PC1ØØ-B mother board contains 1Ø24 bits of non-volatile storage that is organized 256 x 4. The NVM, as it is called, is located on the 8Ø88 CPU bus at address ØEDØØØH through ØEDØFFH and the data path to the device is through data bits Ø, 1, 2, and 3. Phantom images of the NVM exist from address ØED1ØØH through ØEDFFFH.

The device contains a 256 x 4 bit static RAM that performs as any other static memory. The device also contains a 256 x 4 bit non-volatile memory that is overlaid with the 256 x 4 bit static memory. On initialization, the 8Ø88 does a RECALL of the NVM which places that data into the static memory. At this time, any read or write to the memory occurs to the static memory. The RECALL is done via a bit in the Diagnostic Write register. On power-up, this bit is set to a Ø, and must be set to a 1 by firmware before data from the NVM RAM is available.

To perform a RECALL, the bit is set to a Ø and then set back to 1. The minimum width for this pulse is 45Ø ns. The data is available immediately after the RECALL bit is reset. The data that is in the static memory portion can be stored in the NVM by the 8Ø88 CPU via the PROGRAM NVM bit also located in the Diagnostic Write register. This bit is also set to a Ø on power-up. To perform a PROGRAM NVM operation, the bit is set to a 1 and then back to a Ø. This pulse has a minimum width of 1ØØ ns. Once the PROGRAM NVM bit has met the minimum pulse width it can be removed, however, the device cannot be accessed by the CPU for 1Ø ms. At this time, the device is in the process of storing the data into the NVM. There is no indication to the CPU that the device is done other than 1Ø ms has passed. If another operation is done on the device during those 1Ø ms, it will be ignored. Once the operation is started, it cannot be terminated unless the power is turned off. In this case, data in the device is not valid.

3.1.3.5 Optional (Unshared) Dynamic Memory - The module can optionally be expanded with 64KB to 768KB of memory for use by the 8Ø88. If installed, this memory is always available and never requires wait states (except when the memory cycle contends with a refresh cycle).

3.1.3.6 8088 I/O Map - The 8088 I/O map follows:

PORT H	FUNCTION	
00H	Interrupts Z80A Flop (Write)	
00H	Clears 8088 Interrupt Flop (Read)	
02H	Communications and LED Register	WO
02H	General Communications Status	RO
04H	DC011 Write Register	WO
06H	Communications Bit Rate Register	WO
08H	Option Present Status Register	RO
0AH	Maintenance Port	WO
0AH	Maintenance Port	RO
0CH	DC012 Write Register	WO
0EH	Printer Bit Rate Register	WO
10H	Keyboard Data Register(8251A)	RO/WO
11H	Keyboard Control/Status Register(8251A)	RO/WO
20H-2FH	Ext. Comm. Option/Option Select 1	
30H-3FH	Ext. Comm. Option/Option Select 3	
40H	Comm Data Reg. (7201)	RO/WO
41H	Printer Data Reg. (7201)	RO/WO
42H	Comm Control/Status Reg. (7201)	RO/WO
43H	Printer Control/Status Reg. (7201)	RO/WO
50H-5FH	Graphics Option Select	R/W
60H-6FH	Ext. Comm. Option/Option Select 2	
70H-7FH	Ext. Comm. Option/Option Select 4	

3.1.3.7 8088 Memory Map - The 8088 memory map is shown in Figure 2.

```
 ______________________________   FFFFF
|               | ROM 1 (BOOT) |
| ROM 1 (BOOT)  | (128 K bit)  |
|  (256 K bit)  |______________|  FC000
|  (optional)   | ROM 1 (BOOT) |
|               |  (aliased)   |
|_______________|______________|  F8000
|               |    ROM 0     |
|     ROM 0     | (128 k bit)  |
|  (256 K bit)  |______________|  F4000
|  (optional)   |    ROM 0     |
|               |  (aliased)   |
|---------------+--------------|  F0000
|                              |
|        ATTRIBUTE RAM         |
|______________________________|  EF000
|                              |
|        SCREEN RAM            |
|______________________________|  EE000
|                              |
|        NVM (aliased)         |
|______________________________|  ED0FF
|        NVM                   |
|______________________________|  ED000
|                              |
|        NVM (aliased)         |
|______________________________|  EC000

 ______________________________   DFFFF
|                              |
|        OPTION RAM            |
|                              |
|------------------------------|  20000
|                              |
|  STANDARD RAM (UNSHARED)     |
|------------------------------|  10000
|                              |
|  STANDARD RAM (SHARED)       |
|______________________________|  0
```

Figure 2. 8088 Memory Map

3.1.3.8 8Ø88 Interrupts - The following table lists the 8Ø88 interrupts. Values given are in hexadecimal. The values are listed for VECTOR SEL = 1 (default condition), and for VECTOR SEL = Ø (relocated vectors).

Priority	Interrupt Source	Interrupt Type		Vector Address	
		VECTOR SEL		VECTOR SEL	
		1	(Ø)	1	(Ø)
Highest	Memory Parity Error Interrupt (NMI)	Ø2	(Ø2)	(Ø8)	(Ø8)
	Vertical Frequency Interrupt	2Ø	(AØ)	8Ø	(28Ø)
	Extended Comms Interrupt 1 (optional)	21	(A1)	84	(284)
	Graphics	22	(A2)	88	(288)
	DMA Controller Interrupt (from Optional Extended Comm. Board)	23	(A3)	8C	(28C)
	Comm./Printer (72Ø1) Interrupt	24	(A4)	9Ø	(29Ø)
	Extended Comms Interrupt Ø (optional)	25	(A5)	94	(294)
	Keyboard (8251A) Interrupt	26	(A6)	98	(298)
Lowest	Interrupt from Z8ØA	27	(A7)	9C	(29C)

The VECTOR SEL control bit is used to relocate the hardware interrupt vector space to accommodate the requirements of different operating systems. The VECTOR SEL bit is implemented as VECTOR SEL L, using the DTR L output of the 8251A UART (keyboard serial port). Refer to section 3.1.3.1Ø for a more detailed description.

3.1.3.9 Video Subsystem: 8Ø88 - The video subsystem resides on the mother board and is controlled by the 8Ø88. The subsystem provides fully VT1ØØ-compatible video features.

3.1.3.9.1 General Video Features - The video subsystem supports the following features:

a. 24 line x 83 column display
b. 24 line x 137 column display
c. Smooth scrolling (full screen and split screen)
d. Double height lines
e. Double width lines
f. Reverse video
g. Bold
h. Blinking
i. Underline
j. RS17Ø "like" composite video output
k. 255-character set

The software on the 8Ø88 is able to vary the speed of the smooth scrolling, (for example, 3, 6, 12, or 18 lines/sec). The double height and double width attributes may be selected on a line by line basis. The other attributes (reverse, bold, blink, and underline) may be selected on a character-by-character basis.

3.1.3.9.2 Video Memory - The video subsystem has 4KB of screen RAM and 4KB of attribute RAM. Only the four LSBs of the attribute RAM are actually looked at by the video subsystem.

3.1.3.9.3 Video Processor (DCØ11 And DCØ12) - When accessing the screen RAM, the video processor generates the 12-bit address for a particular byte in the lower 4KB bank (character RAM). The corresponding byte in the upper 4KB bank (attribute RAM) is selected also. The two bytes are passed to the video processor in parallel.

The video processor uses the character code to index into a character generator and uses the attribute information to modify the video data.

The contents of the screen RAM directly control the display of the lines and characters. This region of memory contains the displayable characters, their attributes the line attributes, and the addresses that link one line to the next. The microprocessor modifies and updates this information in the intervals between the video processor's DMAs.

The video processor begins reading the screen RAM at the start of RAM (location ØEEØØØH) following each vertical reset. Three bytes of control data are located at the end of each line of characters. The first byte, called the terminator, is FF hex and is a unique character that the video processor recognizes as the end of the line. The next two bytes form an address (low byte followed by high byte) which points to the first character of the next line to be displayed. The byte of attributes that corresponds to the low byte of the address contains three bits of line attributes which are applied to the line being pointed to.

Attribute RAM	Attribute Data	No Attribute	Line Attribute No Attribute
Character RAM	Character Data	Terminator	Address of Next Line

The bits are assigned in the following manner:

	D7	D6	D5	D4	D3	D2	D1	DØ
Char. Attrib.	Unused	Unused	Unused	Unused	Not Under Line	Not Blink	Not Bold	Rev. Video
Char. Data	Alt. Char Set	Code for Character						
Line Attrib.	Unused	Unused	Unused	Unused	Unused	Double Width	Double Height	Scroll Region

(Smooth) scroll region - if set, this line scrolls; if not set, it doesn't.

Double Height	Double Width	Result
Ø	Ø	bottom half double height
Ø	1	top half double height
1	Ø	double width
1	1	normal height, normal width

3.1.3.9.4 DCØ11 **Programming Information** - The DCØ11 video-timing chip can be accessed by the 8Ø88 (WRITE-ONLY) at I/O address 4. The DCØ11 must be programmed with the desired refresh rate and column mode on power-up and after any mode changes. To program the DCØ11, write two of the following four codes:

Code	Configuration
ØØ	8Ø column mode sets
1Ø	132 column mode interlaced mode
2Ø	6Ø Hz mode resets
3Ø	5Ø Hz mode interlaced mode

Interlaced/non-interlaced mode is determined by the order in which 8Ø/132 column and 5Ø/6Ø Hz are set. Every time the DCØ11 is programmed, its internal timing chain is reset. Since this causes the screen to jump, the DCØ11 should be programmed only if absolutely necessary. For example, the following two instructions set the DCØ11 to 8Ø-column, 6Ø Hz, no interlace:

```
MOV AX,2ØØØH
OUT  DCØ11,AX
```

Note

When 8Ø column mode is selected, the video processor is actually capable of displaying 83 columns in single width mode or 41 columns in double width/height mode. When 132-column mode is selected, 137 columns can be displayed in single width mode or 68 columns in double width/height mode.

3.1.3.9.5 DCØ12 Programming Information - The DCØ12 video control chip can be accessed by the 8Ø88 (WRITE-ONLY) at I/O address ØCH. The following codes are defined for the DCØ12:

Code	Result
ØØ	Set scroll latch LSB's to ØØ
Ø1	Set scroll latch LSB's to Ø1
Ø2	Set scroll latch LSB's to 1Ø
Ø3	Set scroll latch LSB's to 11
Ø4	Set scroll latch MSB's to ØØ
Ø5	Set scroll latch MSB's to Ø1
Ø6	Set scroll latch MSB's to 1Ø
Ø7	Set scroll latch MSB's to 11
Ø8	Toggle blink flip flop
Ø9	Clear vertical frequency interrupt
ØA	Set reverse field on
ØB	Set reverse field off
ØC	Not supported
ØD	Set basic attribute to reverse video with 24 lines and set blink flip flop off
ØE	Not supported
ØF	Set basic attribute to reverse video w/48 lines and set blink flip flop off

On power-up, the DCØ12 can be programmed to bring it to a known state. Typically, codes ØØ, Ø4, Ø9, ØB, and ØD will be programmed at power-up time.

The value to which the scroll latch is set determines which scan row the first line of a scrolling region starts on. Likewise, it determines the last scan row displayed for the last line in a scrolling region.

For example, when the latch is set to zero (the degenerate case), the first line of the scroll region starts at scan row zero (so the line is completely visible). The last line of the scrolling region terminates at scan row 9 (so this line is also completely visible).

When the scroll latch is non-zero, for example 5, the first line of the scrolling region starts with scan row 5 (so only the bottom half of the line is visible). The last line of the scrolling region terminates at scan row 4 (so only the top half of the line is visible).

If the scroll latch is incremented from Ø through 9 and back to Ø again once each frame, the screen appears to smooth scroll from bottom to top (assuming that line linkages and line attributes are properly handled). On the other hand, if the scroll latch is decremented from Ø to 9 then down through Ø, the screen appears to smooth scroll from top to bottom (again assuming that all line linkages and line attributes are properly handled).

A scrolling region is defined as a group of lines with their scrolling attributes set, surrounded by lines whose scrolling attribute is not set. Note that the scrolling attribute for a line resides in the line pointer information at the end of the previous line. .Also, the first line on the screen (the one at RAM location Ø), has its scrolling attribute reset by definition. Also note that the definition of a scrolling region does not preclude the definition of more than one scrolling region per screen, although that is of dubious value.

Whenever the scroll latch is non-zero, each scrolling region on the screen requires an extra (scrolling) line to be linked in. For example, if the scrolling region is 1Ø lines long, when the scroll latch is set non-zero there will have to be an eleventh line linked in. If scrolling up (incrementing the scroll latch), the line must be linked in at the bottom. When the scroll latch is incremented back to Ø again, the top line of the scrolling region must be unlinked. When scrolling down (decrementing the scroll latch), new lines must be linked in at the top of the scroll region and unlinked down at the bottom. All line linking/unlinking should be done during the vertical blanking interval (after the vertical frequency interrupt is rung). In 6Ø-Hz mode, there are two blanked lines at the beginning of the screen (the line at RAM location Ø, and the line that it points to).

The first line (at location Ø) is guaranteed to have been read by the time that the interrupt service routine is entered; any changes to this line will not affect the screen until the next frame time. However, the second line will not be read for over 5ØØ microseconds after asserting the interrupt.

If it is to be changed, it must be changed very soon after entering the interrupt service routine in order to guarantee that the change will be visible in the current frame.

Therefore, if the first visible line on the screen is involved in the scroll region and is being either linked in or unlinked, then the vertical interrupt routine must guarantee that its pointer (which resides in the second invisible line) is changed within approximately 5ØØ usec after the ringing of the interrupt.

The modification of the scroll latch is much less time critical than this. Because the scroll latch is loaded by the DCØ12 by the vertical reset at the beginning of each frame, the only requirement is that the scroll latch be modified before the next frame begins. Note that the scroll latch value is the value that will be used during the next frame rather than the current frame.

3.1.3.1Ø Keyboard Interface - The interface to the keyboard is a RS423 full-duplex connection. The interface runs at 48ØØ bits per second asynchronous, with an 8-bit no parity character format. The UART used on the mother board is an 8251A. It must be set up in asynchronous mode with a 16 times clock and 8-bits no parity. The 8251A contains a write-only command register that is used to configure the operating mode of the UART. The first byte written to the command register after a hard or soft reset of the UART is interpreted as a Mode Instruction. For the asynchronous mode of operation, all successive bytes written to the command register are interpreted as Command Instructions.

The Mode Instruction Write Format of the 8215A (output to port 11 hex) is as follows: (shown for asynchronous mode)

D7	D6	D5	D4	D3	D2	D1	DØ
ØØ = Invalid Ø1 - 1 1Ø = 1.5 11 = 2	 Stop Bits	Ø = Odd Parity 1 = Even Parity	Ø = No Parity 1 = Parity	ØØ = 5 Bit Char. Ø1 = 6 Bit Char. 1Ø = 7 Bit Char. 11 = 8 Bit Char.		ØØ = Synchronous Ø1 - 1X Clock 1Ø = 16X Clock 11 = 64X Clock	

The Command Instruction Write Format of the 8215A (output to port 11 hex) is as follows: (shown for asynchronous mode)

D7	D6	D5	D4	D3	D2	D1	DØ
Not Used	1 = Soft Reset	1 = RTS Active	1 = Error Reset	1 = Send Forced break	1 = Enable Recv'r	1 = DTR Active	1 = Enable Xmit'r

DTR (Data Terminal Ready) - This output is normally used to signal the DCE (Data Communication Equipment) that the DTE (Data Terminal Equipment) is ready to communicate. In the PC1ØØ-B, DTR is used as a general purpose, latched output. When DTR L is active low (a "1" in the UART command register), the hardware interrupt vectors appear in their normal locations. When DTR L is inactive high, the vectors are relocated, as defined in section 3.1.3.8.
The recommended setup procedure is to output the following sequence to port 11 (hex):

(all values in hex)

Ø,Ø,Ø,4Ø,4E,17

DURING THE INITIALIZATION, CONSECUTIVE WRITES MUST NOT BE SPACED ANY CLOSER THAN 3 MICROSECONDS.

The Status Read Format of the 8251A (input from Port 11 Hex) follows:

D7	D6	D5	D4	D3	D2	D1	DØ
DSR	SYNDET/ BRKDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY

Note 1

SAME DEFINITIONS AS I/O PINS

PARITY ERROR - The PE flag is set when a parity error is detected. It is reset by the ER bit of the Command Instruction. PE does not inhibit operation of the 8251A.

OVERRUN ERROR - The OE flag is set when the CPU does not read a character before the next one becomes available. It is reset by the ER bit of the Command Instruction. OE does not inhibit operation of the 8251A, however, the previously overrun character is lost.

FRAMING ERROR (Async only) - The FE flag is set when a valid Stop bit is not detected at the end of every character. It is reset by the ER bit of the Command Instruction. FE does not inhibit the operation of the 8251A. A framing error will be generated when the keyboard cable is not properly connected, or if certain hardware malfunctions occur in the keyboard.

DATA SET READY - Indicates that the DSR is at a zero level. Used to read a manufacturing jumper. TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by -CTS and TxEN; the latter is conditioned by both -CTS and TxEN. i.e., TxRDY status bit = DB Buffer Empty TxRDY pin out = DB Buffer Empty (-CTS=0) (TxEN=1)

Note 2

TxE (Transmitter Empty) - When the 8251A has no characters to send, the TxEMPTY output will go "high". It resets upon receiving a character from the CPU if the transmitter is enabled. TxEMPTY remains low when the transmitter is disabled if it is actually empty.

RxRDY (Receiver Ready) - This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY is connected to the interrupt structure of the CPU. For polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

TxRDY (Transmitter Ready) - This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin is used as an interrupt to the system, since it is masked by TxEnable. For Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of -WR when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is not masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

Parity errors should not occur. A hardware or software problem exists if parity errors do occur. The keyboard generates an interrupt to the 8088 when either the TxRDY pin or the RxRDY pin is asserted. The interrupt is a type 26 Hex.

3.1.3.11 Printer Port Interface - This is a general purpose printer port which provides an RS423 interface compatible with DEC printers. EIA signals supported are:

- o Transmit Data
- o Receive Data
- o Data Terminal Ready
- o Data Set Ready asserted

Software programmable bit rates supported are:

75	12ØØ
15Ø	24ØØ
3ØØ	48ØØ
6ØØ	96ØØ

Printer bit rates are selected by writing the following to 8Ø88 port ØEH:

Data Bit Ø-2	**Bit Rate**
Ø	75
1	15Ø
2	3ØØ
3	6ØØ
4	12ØØ
5	24ØØ
6	48ØØ
7	96ØØ

Note

Bit 3 controls the communications port clock as follows:

Source	**Value of Bit 3**
Internal	Ø
External	1

The Printer Port is programmed to use a 16X baud rate clock input.

Software-programmable character formats supported are 5-8 bits/character with 1, 1-1/2, or 2 stop bits/character. Parity may be selected as odd, even, or none. Software should support XON/XOFF restraint protocol, and DTR restraint protocol for this port. The D-type 25-pin female EIA printer connector physically resides on the mother board in the normal printer port location and attaches directly to a printer. DSR and CTS are always asserted.

3.1.4 MPSC Overview

The 72Ø1 Multi-Protocol Serial Controller is a microcomputer peripheral device that supports Asynchronous (Start/Stop), Byte Synchronous (Monosync, IBM Bisync), and Bit Synchronous (ISO's HDLC, IBM's SDLC) protocols. This controller's flexible architecture allows implementation of many variations of these three protocols with low software and hardware overhead.

The Multi-Protocol Serial Controller (MPSC) implements two independent serial receiver/transmitter channels. The printer port uses one channel and the communications port uses the other channel.
As implemented on the mother board, the MPSC supports two microprocessor interface options: Polled and Interrupt.

3.1.4.1 Asynchronous Operations, General - For operation in the asynchronous mode, the MPSC must be initialized with the following information:

- o character length (WR3; D7, D6 and WR5; D6, D5)
- o clock rate (WR4; D7, D6)
- o number of stop bits (WR4; D3, D2)
- o odd, even or no parity (WR4; D1, DØ)
- o interrupt mode (WR1, WR2)
- o receiver·(WR3; DØ) or transmitter (WR5; D3) enable

When loading these parameters into the MPSC, WR4 information must be written before the WR1, WR3, WR5 parameters/commands.

For transmission via a modem or RS423 interface, the Request To Send (RTS) (WR5; D1) and Data Terminal Ready (DTR) (WR5; D7) bits must be set along with the Transmit Enable bit (WR5; D3). Setting the Auto Enables (WR3; D5) bit allows the programmer to send the first character of the message without waiting for a clear to send (CTS).

Both the Framing Error and Receive Overrun Error flags are latched and cause an interrupt.

If the External/Status Interrupt bit (WR1; DØ) is enabled, Break Detect (RRØ; D7) and Carrier Detect (RRØ; D3) will cause an interrupt. Reset External/Status Interrupts (WRØ; D5, D4, D3) will clear Break Detect and Carrier Detect bits if they are set.

A status read after a data read will include error status for the next word in the buffer. If the Interrupt on First Character (WR1; D4, D3) is selected, then data and error status are held until an Error Reset command (WRØ; D5, D4, D3) is given.

If the Interrupt on Every Character Mode bit (WR1; D4, D3) is selected, the interrupt vector is different if there is an error status in RR1. When the character is read, the error status bit is set and the Special Receive Condition vector is returned if Status Affects vector (WR1B; D2) is selected.

In a polled environment, the Receive Character Available bit (RRØ; DØ) must be monitored so that the CPU can determine when data is available. The bit is reset automatically when the data is read. If the X1 clock mode is selected, the bit synchronization must be accomplished externally. Refer to Figure 3.

	D7	D6	D5	D4	D3	D2	D1	DØ
WR3	ØØ RX 5 B/CHAR Ø1 RX 7 B/CHAR 1Ø RX 6 B/CHAR 11 RX 8 B/CHAR		AUTO ENABLES	Ø	Ø	Ø	Ø	RX ENABLE
WR4	OO X1 CLOCK Ø1 X16 CLOCK 1Ø X32 CLOCK 11 X64 CLOCK		Ø	Ø	ØØ ENABLE SYNC MODES Ø1 1 STOP BIT 1Ø 1.5 STOP BIT 11 2 STOP BITS		EVEN/ ODD PARITY	PARITY ENABLE
WR5	DTR Note 1	ØØ TX 5 b/CHAR Ø1 TX 7 b/CHAR 1Ø TX 6 b/CHAR 11 TX 8 b/CHAR		SEND BREAK	TX ENABLE	Ø	RTS Note 1	Ø

Figure 3. Asynchronous Mode Register Setup

Note 1

These bits in MPSC register WR5 not used. Refer to subhead 3.1.4.5.1 Communications Control Register.

3.1.4.2 Communications Port - This port is used to communicate to another computer. It has full modem support and supports the same signals as the VT1Ø2. U.S. and European full- and half-duplex modems can be supported by this port. The port has ASYNC as well as BISYNC modes with a RS423 (V.24/V.28) physical interface conforming to CCITT V.21, V.22 and V.23.

Break detection by this port is supported. Bit rates supported are:

5Ø
75
11Ø
134.5
15Ø
2ØØ
3ØØ
6ØØ
12ØØ
18ØØ
2ØØØ
24ØØ
36ØØ
48ØØ
96ØØ
192ØØ

Communications bit rates are selected by writing the following to 8088 port 06H:

Nibble Data	Bit Rate	Percent Error
0 H	50	0
1 H	75	0
2 H	110	0
3 H	134.5	0
4 H	150	0
5 H	200	0
6 H	300	0
7 H	600	0
8 H	1200	+.14
9 H	1800	0
A H	2000	0
B H	2400	-.17
C H	3600	+.46
D H	4800	+.46
E H	9600	+.46
F H	19200	-2.04

The low nibble of the data written to port 06H sets the transmit clock while the high nibble sets the receive clock. The Communications Port is programmed to use a 16X baud rate clock input.

For example: Data 0AH written to 8088 port 06H would set the receive bit rate to 50 and the transmit bit rate to 2000.

Bit 3 on port 0EH selects the comm port clocks (RxC, TxC). External when set; internal when reset.

Note

> Bit 0-2 on port 0EH controls the printer port bit rates.

All bit rates are software selectable. Transmit and receive bit rates may be selected independently from the available bit rates. The ROM code supports VT102 emulation on this port.

Signals supported are:

1. Receive Data
2. Transmit Data
3. Secondary Transmit Data
4. Request to Send
5. Secondary Request to Send
6. Clear to Send
7. Secondary Clear to Send
8. Receive Line Signal Detect
9. Secondary Receive Line Signal Detect/Speed Indicator (Bell 212A)
10. Ring Indicator
11. Data Set Ready
12. Speed Select

3.1.4.3 Synchronous Operation - Mono Sync, Bi Sync, General - The MPSC must be initialized with the following parameters:

- Odd or even parity (WR4; D1, DØ)
- X1 clock mode (WR4; D7, D6)
- 8- or 16-bit sync character (WR4; D5, D4)
- CRC polynomial (WR5; D2)
- Transmitter Enable (WR5; D3)
- Interrupt modes (WR1, WR2)
- Transmit character length (WR5; D6, D5)
- Receive character length (WR3; D7, D6

WR4 parameters must be written before WR1, WR3, WR5, WR6 and WR7. The data is transmitted on the falling edge of the Transmit Clock (TxC) and is received on the rising edge of Receive Clock (RxC). The X1 clock is used for both transmit and receive operations for all three sync modes: Mono, Bi and External.

	D7	D6	D5	D4	D3	D2	D1	DØ
WR3	ØØ RX 5 Ø1 RX 7 1Ø RX 6 11 RX 8	B/CHAR B/CHAR B/CHAR B/CHAR	AUTO ENABLES	ENTER HUNT MODE	RX CRC ENABLE	Ø	SYNC CHAR LOAD INHIBIT	RX ENABLE
WR4	Ø	Ø	ØØ 8 BIT SYNC Ø1 16 BIT SYNC 11 EXT SYNC		Ø	Ø	EVEN/ ODD PARITY	PARITY ENABLE
WR5	DTR Note 1	ØØ TX 5 Ø1 TX 7 1Ø TX 6 11 TX 8	b/CHAR b/CHAR b/CHAR b/CHAR	SEND BREAK	TX ENABLE	1 (SELECT CRC-16)	RTS Note 1	TX CRC ENABLE

Figure 4. Synchronous Mode Register Setup -- Monosync, Bisync

Note 1

These bits in MPSC register WR5 not used. Refer to subhead 3.1.4.5.1 Communications Control Register.

3.1.4.4 Synchronous Operation, SDLC, General - Like the other synchronous operations, the SDLC mode must be initialized with the following parameters:

- SDLC mode (WR4; D5, D4)
- SDLC polynomial (WR5; D2)
- Request to Send, Data Terminal Ready, transmit character length (WR5; D6, D5)
- Interrupt modes (WR1; WR2)
- Transmit enable (WR5; D3)
- Receive enable (WR3; DØ)
- Auto enable (WR3; D5)
- External/status interrupt (WR1; DØ)

WR4 parameters must be written before WR1, WR3, WR5, WR6 and WR7.

	D7	D6	D5	D4	D3	D2	D1	DØ
WR3	ØØ RX 5 B/CHAR Ø1 RX 7 B/CHAR 1Ø RX 6 b/CHAR 11 RX 8 b/CHAR		AUTO ENABLES	ENTER HUNT MODE	RX CRC ENABLE	ADDRESS SEARCH MODE	Ø	RX ENABLE
WR4	Ø	Ø	Ø 1 (SELECTS SDLC/ HDLC MODE)		Ø	Ø	Ø	Ø
WR5	DTR Note 1	ØØ TX< 6b/CHAR Ø1 TX 7 b/CHAR 1Ø TX 6 b/CHAR 11 TX 8 b/CHAR		Ø	TX ENABLE	Ø (SELECT SDLC CRC)	RTS Note 1	TX CRC ENABLE

Figure 5. Synchronous Mode Register Setup -- SDLC/HDLC

Note 1

These bits in MPSC register WR5 not used. Refer to subhead 3.1.4.5.1 Communication Control Register.

3.1.4.5 Modem Control Lines - Several modem control signals are not implemented using the 72Ø1 Multi-Protocol Serial Controller chip. These signals are implemented with the Communications Control Register and The Communications Status Register.

3.1.4.5.1 Communications Control Register - The communications control register is an 8-bit write only register that controls the modem lines on the communications port. It also controls diagnostic error codes displayed by the four 8Ø88 LEDs. This register is accessed by performing a write to address Ø2H. The register bit format is shown in Figure 6 and the bits are described in Table 1.

```
7   6   5   4   3   2   1   Ø   ADDRESS Ø2H (WO)
|   |   |   |   |   |   |   |__ COMM SPD SEL H
|   |   |   |   |   |   |______ COMM SRTS H
|   |   |   |   |   |__________ COMM DTR L
|   |   |   |   |______________ COMM RTS
|   |   |   |__________________ LED D6 (LSB)
|   |   |______________________ LED D3
|   |__________________________ LED D4
|______________________________ LED D5 (MSB)
```

Figure 6. Communications Control Register (8Ø88) Format

Table 1. Communications Control Register (8Ø88) Bit Description

Bit	Name	Description
Ø	COMM SPD SEL H	This bit controls the Speed Select line of the communications port.
1	COMM SRTS H	This bit controls the Secondary Request To Send line of the communications port.
2	COMM DRT L	This bit controls the Data Terminal Ready line of the communications port.
3	COMM RTS	This bit controls the Request To Send line of the communications port.
4	LED (D6)	This bit displays the least significant bit of the diagnostic error message code. When written with a Ø, the LED lights.
5	LED (D3)	This bit displays the second bit of the diagnostic error message code. When written with a Ø, the LED lights.
6	LED (D4)	This bit displays the third bit of the diagnostic error message code. When written with a Ø, the LED lights.
7	LED (D5)	This bit displays the most significant bit of the diagnostic error message code. When written with a Ø, the LED lights.

3.1.4.5.2 Communications Status Register - The Communications Status Register is an 8-bit read only register that holds the status of the modem control lines for the communications port. It also serves as a status register for the special interprocessor interrupt lines and the status of the hardware failure detect enable signal. This register is accessed by performing a read to address Ø2H. The register bit format is shown in Figure 7 and the bits are described in Table 2.

```
7  6  5  4  3  2  1  Ø   ADDRESS Ø2 H (RO)
|  |  |  |  |  |  |  |__ COMM RI
|  |  |  |  |  |  |_____ COMM SI/SCF
|  |  |  |  |  |________ COMM DSR
|  |  |  |  |___________ COMM CTS
|  |  |  |______________ COMM RLSD
|  |  |_________________ HDWRE FAILURE DETECT ENABL
|  |____________________ INT 88 L
|_______________________ INT Z8Ø L
```

Figure 7. Communications Status Register (8Ø88 Format)

Table 2. Communications Status Register (8Ø88) Bit Description

Bit	Name	Description
Ø	COMM RI	This bit reflects the status of the Ring Indicator line of the communications port.
1	COMM SI/SCF	This bit reflects the status of the Speed Ondicator line or the Secondary Receive Line Signal Detect of the communications port.
2	COMM DSR	This bit reflects the status of the Data Set Ready line of the communications port.
3	COMM CTS	This bit reflects the status of the Clear To Send line of the communications port.
4	COMM RLSD	This bit reflects the status of the Feceive Line Signal Detect of the communications port.
5	HFD ENB L	This bit reflects the status of Hardware Failure Detect Enable L.
6	INT 88 L	This bit reflects the status of the INT 88 L bit that is asserted by the Z8ØA to interrupt the 8Ø88, for interprocessor communications.
7	INT Z8Ø L	This bit reflects the status of the INT Z8ØA L bit that is asserted by the 8Ø88 to interrupt the Z8ØA for interprocessor communications.

3.1.5 Z8ØA System

The following describes the section of the system controlled directly by the Z8ØA.

3.1.5.1 Z8ØA CPU - The module includes one Z8ØA microprocessor, which runs from a clock of 4.Ø12 MHz. The Z8ØA alone has access to the floppy disk interface and thus is responsible for controlling the floppy (via programmed I/O) for all applications.

3.1.5.2 Z8ØA Shared Memory - The Z8ØA has available to it a 64KB RAM that is divided into 62KB shared and 2KB unshared memory. Accesses to the shared portion of memory select the corresponding address in the standard bank of 64KB RAMS. Accesses to the unshared memory select a private 2K x 8 byte-wide static RAM.

If the shared RAM is "busy" at the time of a Z8ØA access, the Z8ØA will execute wait states until the RAM is free. The RAM is considered "busy" when an 8Ø88 cycle or a refresh cycle is in progress or is pending.
In addition to wait cycles due to contention, all M1 cycles from the shared RAM have one extra wait cycle due to the timing for this sort of machine cycle.

In any case, the Z8ØA is held in a wait state for no longer than approximately two microseconds. If both processors are executing out of the shared memory, the Z8ØA cannot reliably access the floppy disk (for example, lost data errors will often result).

3.1.5.3 Z8ØA Private RAM - The 2 KB of unshared RAM may be accessed by the Z8ØA at any time without any wait states.

3.1.5.4 Z8ØA I/O Map - The following is the Z8ØA I/O map.

PORT	FUNCTION	
ØØH	Clear Interrupt to Z8ØA (Read)	RO
ØØH	Interrupts 8Ø88 (Write)	WO
2ØH	Set ZFLIP	WO (See Note)
21H	Disk Diagnostic Read Register	RO
21H	Disk Diagnostic Write Register	WO (See Note)
4ØH	Disk Control Read Register	RO
4ØH	Disk Control Write Register	WO
6ØH	FDC Status Register	RO
6ØH	FDC Control Register	WO
61H	FDC Track Register	R/W
62H	FDC Sector Register	R/W
63H	FDC Data Register	R/W

Note

The above Z8ØA I/O ports have a great number of alias addresses throughout the Z8ØA's 256 I/O port address space. Prudent programming practice precludes using any Z8ØA I/O port address that is not defined above.

Writing Diagnostic Write Register at address 21H will reset ZFLIP. Writing the Diagnostic Write Register at address 2ØH will set ZFLIP. Inadvertent use of these registers will likely cause program problems.

3.1.5.5 Z8ØA Memory Map - The Z8ØA memory map is shown in Figure 8.

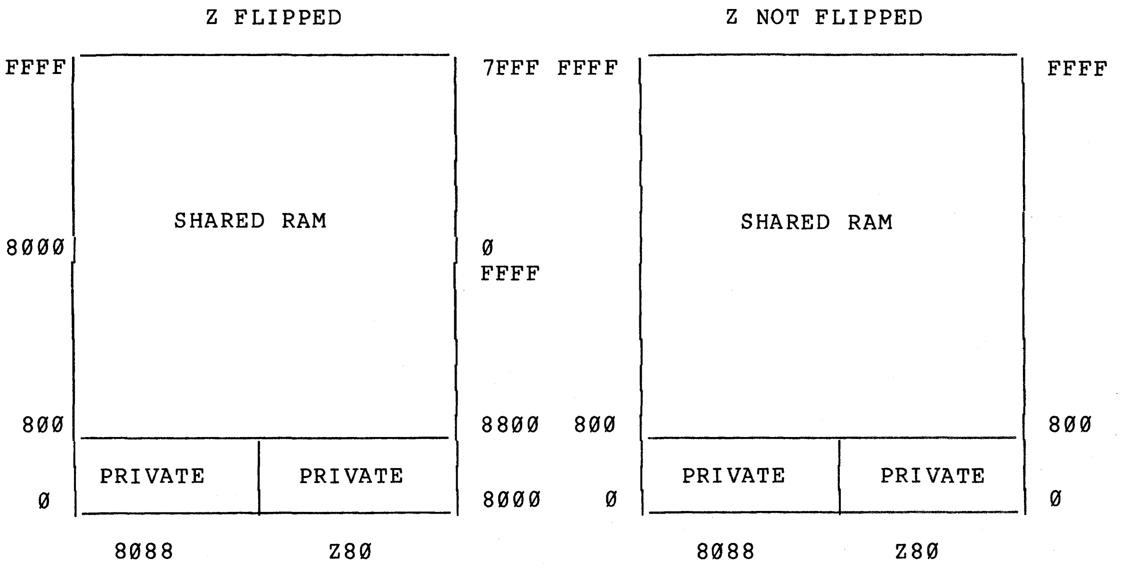


Figure 8. Z80A Memory Map

3.1.5.6 Z80A Cycle Time - The clock time on the Z80A is approximately 250 ns. Unshared memory accesses have no wait states. Shared memory accesses have wait states on M1 cycles and for cycles in which there is contention between devices accessing the shared RAM. Contention exists because of refresh cycles and 8088 cycles.

3.1.5.7 Z80A Interrupts - The only interrupts are interprocessor interrupts from the 8088 CPU. The vector placed on the bus is F7 (hex) which causes a RST 30 instruction to be executed in interrupt mode 0.

3.1.5.8 Floppy Controller Module - The floppy controller module is not optional. It is a separate module that connects to the mother board via J7. The interface is designed to control up to four 5-1/4 inch platters with one or two surfaces. The controller supports soft-sectored double-density diskettes using a PLL circuit. Single- or double-sided drives are supported. The interface adheres to drive capability and signal definition of the ANSI standard interface for mini-floppy drives.

The floppy controller block diagram is shown in Figure 9.

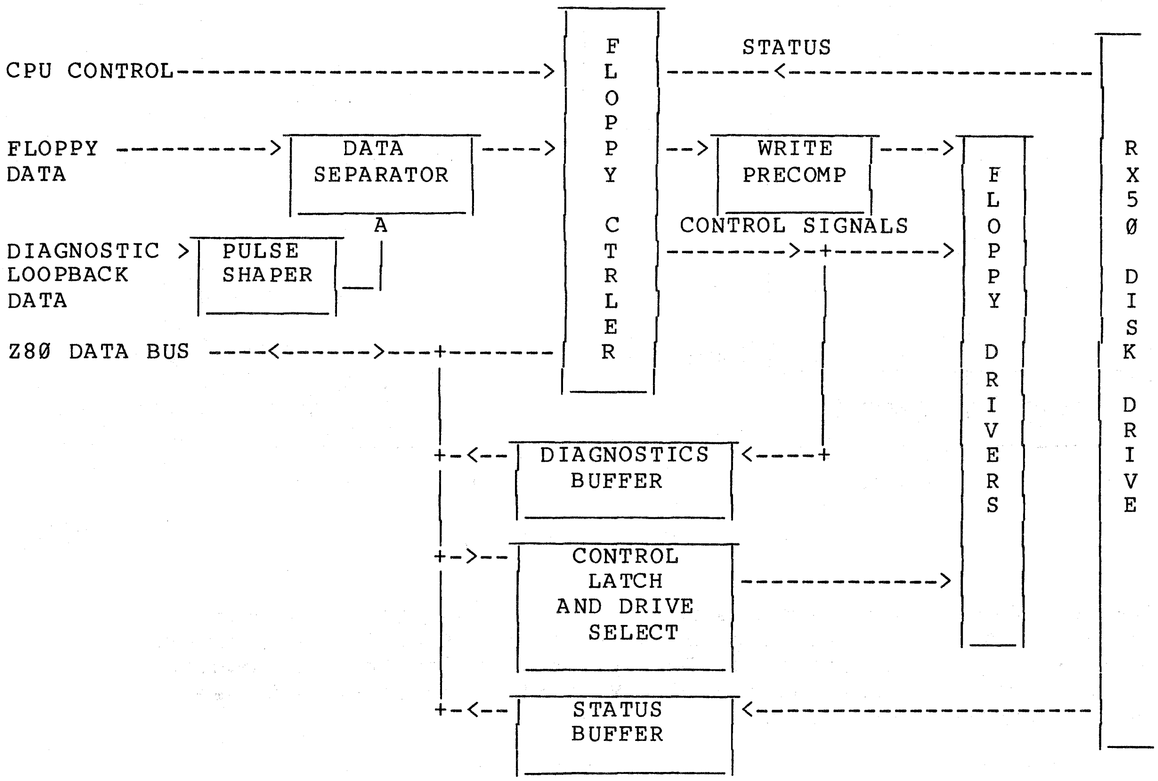


Figure 9. Floppy Controller Block Diagram

3.1.5.8.1 Floppy Disk Controller Registers: Z80A

1. **Command Register (Write Only) (Port Address 60 Hex)**

   This 8-bit write-only register is loaded by the program with the command that is to be executed by the drive. A command summary follows:

Table 3. Command Summary

								Bits	
Type	Command	7	6	5	4	3	2	1	Ø
I	Restore	Ø	Ø	Ø	Ø	h	v	r(1)	r(Ø)
I	Seek	Ø	Ø	Ø	1	h	v	r(1)	r(Ø)
I	Step	Ø	Ø	1	u	h	v	r(1)	r(Ø)
I	Step In	Ø	1	Ø	u	h	v	r(1)	r(Ø)
I	Step Out	Ø	1	1	u	h	v	r(1)	r(Ø)
II	Read Sector	1	Ø	Ø	m	Ø	e	Ø	Ø
II	Write Sector	1	Ø	1	m	Ø	e	Ø	a(Ø)
III	Read Address	1	1	Ø	Ø	Ø	e	Ø	Ø
IV	Force Interrupt	1	1	Ø	1	I(3)	I(2)	I(1)	I(Ø)

Note

Read Track and Write Track are not supported.

Flag Summary

Type I Commands

h = Head Load Flag (Bit 3)
v = Verify Flag (Bit 2)
rl,rØ = Stepping Motor Rate (Bits 1-Ø)
u = Update Flag (Bit 4)

Type II and III Commands

m = Multiple Record Flag (Bit 4)
aØ = Data Address Mark (Bit Ø)
e = 3Ø msec delay

Type IV Commands

Ii = Interrupt Condition Flags
For more detailed information on the meaning and purpose of these bits, refer to the System Module Functional Specification.

Floppy Command Summary

This module accepts nine commands for floppy disk control. See Table 3 for a command summary. Commands should only be loaded into the command register when the Busy status bit is off. The one exception is the Force Interrupt command. The Busy status bit is set when a command is executed.

Type I Commands

Type I commands are for head positioning. The stepping rate of these commands are dictated by the drive. Rl = Ø and RØ = Ø (6 ms) is the recommended stepping rate for the RX5Ø drive.

The head load flag determines if the head is loaded at the beginning of the command. Otherwise, the head is loaded at the end of a command.

The verification flag allows a verification operation to take place on the destination track. The verification consists of reading the first encountered ID field off of the disk.

The track address of the ID field is compared to the Track Register. If there is a match and a valid ID CRC, the verification is complete. If not valid, the Seek error status bit in the FDC is set.

The Step, Stepin, and Stepout commands contain an update flag for updating the track register when this bit is set after the step has been completed.

Type II Commands

The Type II commands are to read and write sectors to the disk. Prior to loading the Type II command into the Command register, the Sector register must be loaded with the desired sector number.

Upon receipt of the command, the Busy status bit is set. If the e flag is set (normal case), the head is loaded and the HLT signal is sampled after 3Ø ms; otherwise, no delay is incurred after a command.

The HLT does not become active until 5ØØ ms after the head is loaded to allow the spindle motor to have time to accelerate. The FDC then attempts to find the ID field with the specified track and sector.

If the desired field is not found within five revolutions of the disk, the Record Not Found status bit is set. Otherwise, the command is executed by the FDC generating Data Requests (DRQS) for servicing the data register. Each of the Type II commands contain an m flag which determines if multiple sectors are to be read or written, depending on the command. When set, multiple sectors are read or written with the sector register internally updated for address verification on the next track.

The FDC continues to do the transfers until the sector register exceeds the number of sectors on the track or until a force interrupt command is loaded into the command register.

Note

> If the command is not terminated by software, the 1793 continues looking for five index pulses after the last sector on the disk has been read or written.

If the Sector register exceeds the number of sectors on the track, the Record Not Found ISTER EXCEEDS THE NUMBER OF SECTORS ON THE TRACK, THE Record Not Found Status bit is set. When the head is loaded, the Busy status bit is set, and when an ID field is encountered that has the correct track, sector, side numbers and correct CRC, the data field is presented to the computer (read) or presented by the computer (write).

At the end of the Read operation, the type of Data Address Mark encountered is recorded in the Status register (Bit 5). On a Write operation, the aØ flag (Bit Ø) determines the type of Data Address Mark to be written onto the disk. If set, a deleted data mark is written else a data mark is written.

Type III Commands

The Read Address command is to read in the six bytes of the ID field (track number, side number, sector address, sector length, and two bytes of CRC).

Type IV Command

This command is to terminate an operation upon the specified condition in Bits 3-Ø. See Table 1, Command Summary, for descriptions of the termination conditions.

Type I Command Bit Description

1. **Bit Ø,1 - Stepping Rate Bits** - These bits control the rate at which the stepping pulses are sent to the drive. Check the specifications for the drive in use to determine the drive's proper step rate. See the table above for stepping rate breakdown.
2. **Bit 2 - Track Verify Bit** - This bit determines if there is a verification operation to take place on the destination track. During verification, the head is loaded and after a 3Ø-ms delay, the HLT input is sampled. After a 5ØØ-ms motor start up time, the HLT input becomes active. When HLT is true, the first ID field is read off the disk. The track address of the ID field is compared to the track register. If there is a match and a valid ID CRC, the verification is complete and an interrupt is generated. If not valid, the seek error status is set.

3. **Bit 3 - Head Load Flag** - This bit determines if the head is to be loaded at the beginning of a command. If the head is loaded then the head remains loaded until either the FDC receives a command that specifically disengages the head or 15 revolutions of the disk have passed with the busy bit = Ø.

4. **Bit 4 - Update Bit (Step Commands)** - When set, the track register is updated by one for each step; otherwise the track register is not affected.

5. **Bits 5-7** - Determine the command to be executed.

Type II Commands Bit Description

1. Bit Ø - Data Address Mark Bit - When set upon a write sector command, this bit defines a Data Mark (ØFBH) to be written on the disk. If the bit is not set then a Deleted Data Mark (ØF8H) is written onto the disk. When writing valid data on the disk this bit should be set.

2. Bit 1 - Always Ø.

3. Bit 2 - 3Ø Millisecond Delay Bit - When set during a command, there is a 3Ø-ms delay before reading begins. For maximum controller throughput, this bit should be Ø. It should be set if the last command was a seek or new drive select.

4. Bit 3 - Always Ø.

5. Bit 4 - Multiple Sector Bit - When set, this bit allows multiple sectors to be transferred.

6. Bits 5-7 - Determine the command to be executed.

Type III Command Bit Description

1. Bits Ø,1 - Always set to Ø.

2. Bit 2 - Same as Bit 2 for Type II commands.

3. Bits 3-7 - Determine the command to be executed.

Type IV Command Bit Description

This command can be loaded into the register at any time. If there is a current command under execution, the command is terminated. See Table 3 for a description of conditions upon which the command is terminated.

2. Status Register (Read Only) (Port Address 6Ø Hex)

This read only register also resides at the same address as the command register. It contains the 8-bit status resulting from the completion of a command. A description of the status bits follows.

Type I Status Bit Description

```
7   6   5   4   3   2   1   Ø
|   |   |   |   |   |   |   |__ Busy
|   |   |   |   |   |   |______ Index
|   |   |   |   |   |__________ Track Ø
|   |   |   |   |______________ ID Field CRC Error
|   |   |   |__________________ Seek Error
|   |   |______________________ Head Loaded
|   |__________________________ Write Protect
|______________________________ Not Ready
```

1. **Bit Ø - Busy Bit** - When this bit is true (1), the FDC is currently executing a command. Only a Type IV command can be issued when this condition exists.

2. **Bit 1 - Index Bit** - When this bit is true (1), the index pulse is currently occurring.

3. **Bit 2 - Track Ø Bit** - When this bit is true (1), the read/write head is currently positioned at track Ø.

4. **Bit 3 - ID Field CRC Error Bit** - When true, this means that there was a CRC error of the ID field.

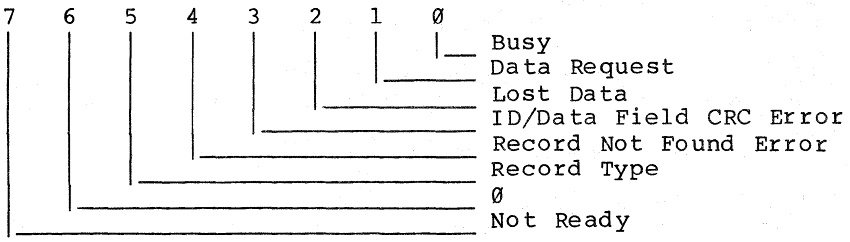
5. **Bit 4 - Seek Error Bit** - When true, a seek error was encountered meaning that the destination track address was not found.

6. **Bit 5 - Head Loaded Bit** - This bit reflects the current status of the head. When set, the head is loaded and the HLT input is asserted.

7. **Bit 6 - Write Protect Bit** - When set, the bit means that the current disk is write protected. An attempt to write a sector generates an interrupt if the device interrupt enable bit is set.

8. **Bit 7 - Not Ready Bit** - When set, the bit indicates that the drive is not ready. This could mean that the drive is not up to speed, the disk is in upside down, or the door is open. This bit must be clear before any commands are issued to the FDC.

Type II Read Sector Status Bit Description



1. **Bit Ø - Busy Bit** - Same as Type I status.

2. **Bit 1 - Data Request Bit** - This bit means that the data register is full and it is waiting for the CPU to read the register.

3. **Bit 2 - Lost Data Bit** - When set, it means that the data register had not been serviced within 27.Ø microseconds and the data in the data register is not valid.

4. **Bit 3 - ID/Data Field CRC Error Bit** - When set, an error is found in one or more ID fields or the data field. This bit is reset when updated.

5. **Bit 4 - Record Not Found Error Bit** - When equal to one, this bit means that a Data Address Mark was not found within 43 bytes of the last ID field CRC byte or it can indicate that the desired track, sector or side was not found.

6. **Bit 5 - Record Type Bit** - This bit reflects the type of Data Mark that was encountered during the read. When set a Deleted Data Mark was found. If clear, a Data Mark was encountered.

7. **Bit 6** - Always set to Ø.

8. **Bit 7 - Not Ready Bit** - Same as Type I Not Ready Status Bit.

Type II Write Sector Status Bit Description

```
7   6   5   4   3   2   1   Ø
|   |   |   |   |   |   |   |__ Busy
|   |   |   |   |   |   |______ Data Request
|   |   |   |   |   |__________ Lost Data
|   |   |   |   |______________ CRC Error
|   |   |   |__________________ Record Not Found Error
|   |   |______________________ Write Fault
|   |__________________________ Write Protect
|______________________________ Not Ready
```

1. **Bit Ø - Busy Bit** - Same as Type I status Busy Bit.

2. **Bit 1 - Data Request Bit** - This bit means that the data register is empty and it is waiting for the CPU to write the register.

3. **Bit 2 - Lost Data Bit** - When set, it means that the data register had not been written within 23.Ø microseconds and the data on the disk is not valid (zero bytes are substituted for data lost).

4. **Bit 3 - CRC Error Bit** - When set, this bit indicates an error in one or more ID fields. This bit is reset when updated.

5. **Bit 4 - Record Not Found Error Bit** - When equal to one, this bit indicates that the desired track, sector or side was not found.

6._ **Bit 5 - Write Fault Bit** - Not implemented; should always be Ø.

7. **Bit 6 - Write Protect Bit** - When this bit is set after a write command, then an attempt was made to write on a write protected disk.

8. **Bit 7 - Not Ready Bit** - Same as Type I Not Ready Status Bit

3. Track Register (Port Address 61 Hex)

This R/W 8-bit register holds the updated address of the current read/write head. It is incremented by one every time the head is stepped toward the spindle and decremented by one every time the head is stepped away from the spindle. The contents of the register are compared with the recorded track number in the ID field during disk read, write and verify operations.

4. Sector Register (Port Address 62 Hex)

This read/write 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk read and write operations.

5. **Data Register (Port Address 63 Hex)**

For a seek operation, this 8-bit read/write register holds the desired track position. During data transfers, this register is the data buffer for the disk.

3.1.5.8.2 General Control and Status Register Bit Description: Z8ØA -

This 8-bit register holds various control information for the drive as well as the module. The lowest four bits are read/write while the upper four bits are read only.

3.1.5.9 General Floppy Control Register: Z8ØA - The following write-only register (Port Address 4Ø Hex) holds control lines used to select drives and write delay pre-comp values for the floppies.

1. **Bits Ø-1** - These bits control the selection of floppy drives. The binary values written to them (Ø - 3) selects drive Ø through 3. Only 1 drive can be selected at a given time.

2. **Bit 2 - Diagnostic READY override bit** - When set, this bit asserts DRIVE READY to the 1793.

3. **Bit 3** - This bit controls the MOTOR Ø ON bit. Turns on the motor on in the first drive unit.

4. **Bit 4** - This bit controls the MOTOR 1 ON bit. Turns on the motor on in the second drive unit.

5. **Bit 5** - This bit selects the SIDE of the disk to be accessed. For single-sided drives, this bit is always set to a Ø for side Ø.

6. **Bits 6-7** - These binary bits are used to control the write delay pre-comp values. The following table lists the values for all tracks:

(TG43)	PC1	PCØ	TRACK
Ø	Ø	Ø	Ø - 9
Ø	Ø	Ø	1Ø - 19
Ø	Ø	Ø	2Ø - 29
Ø	Ø	Ø	3Ø - 39
1	Ø	Ø	4Ø - 49
1	Ø	Ø	5Ø - 6Ø
1	Ø	1	61 - 69
1	Ø	1	7Ø - 79

3.1.5.9.1 Drive Select Light Operation - The drive select logic is set up so that none of the drives are enabled on power-up. When a disk is installed, the door is closed, the drive is selected and either HEAD LOAD or MOTOR ON is asserted. Then the drive active indicator light illuminates, the head loads, and the motor turns on. Only one drive can be selected at a time.

The drive motors, on the other hand, are not gated with any signals. Each motor on signal can be activated independent of any other condition. The software never turns on both motors simultaneously. It is necessary to delay the start of the second selected motor for 5ØØ ms after the start of the first motor.

CAUTION

> When both drive motors are off, a MOTOR ON override must not be generated for the unselected drive. Due to a hardware idiosyncrasy, this causes both drive motors to turn on simultaneously.

3.1.5.1Ø **General Floppy Status Register: Z8ØA** - The following read-only register (Port Address 4Ø Hex) holds the status of the RX5Ø drive lines coming from the 1793 FDC and going to the floppy drive.

1. **Bits Ø-1** - These bits read back the status of Bit Ø and 1 from the general floppy control register. They indicate which drives have been selected.

2. **Bit 2** - This bit reflects the status of the TRACK GREATER THAN 43 signal from the 1793 going to the floppy.

3. **Bit 3** - This bit reflects the status of MOTOR ON Ø line at the floppy connector. The signal, when read as Ø, indicates that the MOTOR ON Ø bit is set.

4. **Bit 4** - This bit reflects the status of MOTOR ON 1 line at the floppy connector. The signal, when read as Ø, indicates that the MOTOR ON 1 bit is set.

5. **Bit 5** - This bit reflects the status of the side select signal at the floppy connector.

6. **Bit 6** - This bit reflects the status of the INTERRUPT REQUEST signal coming from the 1793. This is used to indicate that a status bit has changed.

7. **Bit 7** - This bit reflects the status of the DATA REQUEST signal from the 1793. Used to indicate that the 1793 has read data to be transferred or requires new write data.

3.1.5.1Ø.1 Floppy Disk Controller Required Delays - The following list describes required delays from one operation to the next operation:

Operation	Next Operation	Delays Required (microseconds)
Write to Command Register	Read Busy Bit (status bit Ø)	12
Write to Command Register	Read Status Bits 1-7	28
Write to Any Register	Read from Different Register	Ø
Write to Track, Sector or Same Register	Read from Data Register	8
Write to Any Register	Write to Another Register	14
Interrupt Request	Read Status Register	4

3.1.5.1Ø.2 Floppy Disk Motor Speedup Detection - While writing to a sector on a disk, the opening or closing of the OTHER DRIVE'S door will cause a transient in the floppy spindle motor speed. This transient can make any sector being written at the time of the opening or closing to be not readable in all situations. The following procedure is recommended and is implemented in the BIOS of CP/M-86/8Ø:

Just prior to writing a sector upon the disk, the drive being written to is de-selected. The other drive IN THE SAME RX5Ø disk assembly is then selected and the condition of the Ready bit is sampled. The disk being written to is then selected again, and the write operation is performed.

After the sector has been completed, the same operation is performed; the disk drive being written to is de-selected and the sister drive is selected. Ready is again sampled. If the condition of this bit had changed from the previous sample taken, then the sector must be re-written.

3.1.5.1Ø.3 Floppy Controller Head Load Timer Activation - The Head Load Timer can be fired only in the following circumstance: The Head Load Timer must not be already timing. This means that neither MOTOR ON bits (MØ, M1) are active, nor is the HEAD LOAD bit (HLD). Upon the next occurence of any of these three bits, and the state of the drive is READY, then the Head Load Timer will be actuated. This timer puts a 5ØØ-ms delay prior to HLT going true.

3.1.6 Mother Board Physical Dimensions

The mother board is a modified quad module with the following connectors:

J1	Communications Connector	25-pin	D-male
J2	Printer Connector	25-pin	D-female
J3	Video/Keyboard Connector	15-pin	D-male
J4	Extended Comms Connector	40-pin	HEADER
J5	Extended Comms Connector	40-pin	HEADER
J6	Memory Option Connector	52-pin	HEADER
J7	Graphics Option Connector	40-pin	HEADER
J8	Power Connector	13-pin	
J9	Floppy Controller Pin	40-pin	
J10	A/B Floppy Board	34-pin	
J11	C/D Floppy Board	34-pin	

3.2 OPTION MODULES

The following option modules will be supported by the Rainbow system:

3.2.1 Memory Option Description/Features

The memory option for the PC100-B allows the user to upgrade the system with an additional 64K to 768K bytes of memory. In addition to the 128K bytes of standard memory on the system module, a total of 896K bytes of available memory for the PC100-B is possible.

There are two basic types of memory options for the PC100-B. Each type has two different variants.

The 64KB and 192KB variants of the memory option use the same 50-class etch. The 128KB and 256KB variants use the same 50-class etch. The 64KB and 192KB boards are not user upgradeable. The 128KB and 256KB boards are user upgradeable to a maximum of 768KB, using upgrade kits supplied by DEC.

If installed, this memory is always available and never requires wait states, except when the memory cycle contends with a refresh cycle.
The option is equipped with parity generation and a parity error detect circuit to notify the 8088 CPU in the event of a memory error. If such an error occurs, the memory option interrupts the 8088 CPU through a non-maskable interrupt. At this point the firmware takes the proper action to notify the user.

3.2.2 Extended Communications Option - Description/Features

The extended communications option card is a major component of the PC100-B system and is connected to the PC100-B main module via standoffs. The purpose of the option is to add a second communications port to the PC100-B with bit and byte synchronous capability. It also gives the PC100-B a separate high-speed serial communications port to support clustering and the addition of a high-speed file server. It has two 40-pin connectors through which it plugs into the system.

The extended communications option functions in the following ways within the PC1ØØ:

1. By means of the 8237 DMA Controller, block transfers data bi-directionally between memory and the high-speed communications link (72Ø1 shared MPSC) while maintaining full interrupt support. The transfer to memory is into the PC1ØØ's shared RAM only and not into optional memory.

2. Distinguishes bit protocols at a clock rate of 722 kHz by means of the 72Ø1 MPSC.

3. Provides an optional bisync port (72Ø1 MPSC) that is a subset of the PC1ØØ's communication port.

4. Provides two complete serial communications controllers in a single 72Ø1 MPSC package to:

   a. Convert parallel data (from the processor) to serial data, as required by various protocols.

   b. Convert serial data streams of the protocols back to parallel data for the processor.

   c. Buffer incoming and outgoing data, allowing the processor time to respond.

   d. Insert and delete framing bits and characters.

   e. Calculate and check parity and check CRC error.

   f. Inform CPU what actions need to be taken and when.
   g. Interface with outside world over discrete modem control lines.

5. Uses a 72Ø1 Bus Interface Controller to provide:

   a. Bus Control Logic (BCL), which determines the internal source or destination of data and control transfers between the MPSC and the processor bus.

   b. Interrupt Control Logic (ICL), which prioritizes internal input requests and places information on the data bus during an Interrupt Acknowledge cycle (provided the MPSC vectored interrupt feature has been enabled)

   c. DMA Control Logic (DMACL), which enables the MPSC to make a data transfer without interrupting the processor. DMACL accepts service requests (if they are prioritized) and, like ICL (in b above), places information on the data bus at appropriate times. DMACL also accepts information from the data bus. When enabling the MPSC, DMACL activates an external controller to move data directly from the MPSC to memory or vice versa.

d. Clock and Reset Logic (C&RL), which controls timing states in the MPSC and is (usually) connected to the processor clock. The extended communications option consists of the following main components mounted on a printed circuit board:

1. A 5 MHz 8237 Direct Memory Access Controller (DMAC).
2. A 7201 Multi-Protocol Serial Controller (MPSC) with the following features:

    a. A high-speed synchronous serial communications port with external clocks and RS422 differential drive capability.

    b. A general-purpose synchronous serial communications port, with RS423 drive capability, capable of supporting bisync modes.

Refer to the Extended Communications Option Functional Specification for further information.

3.2.2.1 Reset Sequence For Extended Communications - The firmware will perform the following RESET function on the Extended Communications option upon power up, and any time that it has to handle an interrupt from the Extended Communications option: A write to 8088 port 27H will reset the option.

3.2.3 Graphics Option - Description/Features

3.2.3.1 Overview - The Graphics option is a bit mapped color graphics option which resides on a daughter board inside the Rainbow system box, and attaches to the Rainbow system board via a 40-pin connector, J7. The Graphics option will emulate VT240 functionality in both graphics and text handling. This includes, but is not limited to, the funcionality of the VT100, VT102, and VT125.

3.2.3.2 Graphics System Configurations - Three distinct system configurations are possible, dependent on the selection of the monitor(s) and cable(s) that are chosen. These are as follows:

a. Monochrome Graphics Configuration (VR201 only)

b. Color Graphics Configuration (VR241 only)

c. Monochrome and Color Graphics Configuration (VR201 and VR241)

Refer to the Rainbow Graphics Module Specification, A-SP-5415688-0-DBP, for programming information.

3.2.3.2.1 Monochrome Graphics Configuration - The monochrome graphics configuration uses the VR201 monitor and a BCC02 cable. These items are shipped as standard equipment with the base Rainbow System. In this configuration graphical output on the monitor is provided by selecting the "grey" bit map output from the graphics option module. This selection is made by a multiplexer that selects between the "grey" bit map output of the graphics option module and the output of the DCØ11 and DCØ12 character cell video display controller subsystem on the motherboard.

The default selection, upon power-up, is the DCØ11 and DCØ12 circuit. The graphics option module may be selected by setting bit 2 of 8Ø88 I/O port ØAH. This should be done after the graphics option module has been programmed for proper screen format. To reselect the character cell video display controller, bit 2 of I/O port ØAH should be reset to zero.

3.2.3.2.2 Color Graphics Configuration - In the color monitor only graphics configuration, a VR241 color monitor and a BCC17 cable are used. These items are ordered separately from the base Rainbow System.

In this configuration, graphical output on the color monitor is provided by the red, grey, and blue bit maps of the graphics option. The green bit map is not used, and must be filled with the value ØFFH. The "green" information is placed in the grey bit map. The green input of the color monitor is driven by the monochrome video output of the motherboard.
The function of the monochrome output, and its ability to be multiplexed between the graphics option module and the DCØ11 and DCØ12 character cell video display controller, have been described in the previous section. The selection of the character cell video display controller, as described in the previous section, causes the textual information from the DCØ11 and DCØ12 to be displayed in green on the color monitor. When in the text mode, the red and blue bit maps of the graphics option must be disabled.

3.2.3.2.3 Monochrome and Color Graphics Configuration - The monochrome and color graphics configuration uses a VR2Ø1, a VR241, and a "Y" cable (part number to be assigned). The VR2Ø1 monochrome monitor is driven by the monochrome video output from the motherboard. This is the only one of the two monitors that can display textual information from the DCØ11 and DCØ12 character cell video display controller. Graphical output to the monochrome monitor, from the grey bit map of the graphics option module is not supported in this configuration.

The color monitor is driven by the red, green, and blue outputs of the graphics option module. Graphical information is placed in the red, green, and blue bit maps, respectively, of the option module.

3.2.3.3 Features - The Graphics option for Rainbow will support the following features:

a. Medium resolution mode - 24Ø X 38Ø pixels X 4 planes

b. High resolution mode - 24Ø X 8ØØ pixels X 2 planes

c. 16 simultaneous colors from a pallet of 4Ø96

d. 96ØØ baud character throughput (hardware only)

e. Smooth and jump split screen scrolling

3.2.3.4 Differences From PC1ØØ-A to PC1ØØ-B Graphics Operation - The PC1ØØ-B motherboard circuitry supports 16 shades (levels) of grey scale from the monochrome video output. The PC1ØØ-A motherboard circuitry supports 4 shades of grey scale. The difference is only detectable in the medium resolution mode; the performance is the same in high resolution mode.

3.2.4 Winchester Disk Storage Option

3.2.4.1 General Description - The RCD51-BA subsystem for the PC1ØØ-B consists of two assemblies: an RD51-A 1Ø MByte Winchester Disk Drive, and a controller module.
The RD51 Winchester drive is a low cost, random access, rotating memory device which stores 1Ø Mb of data in fixed length blocks on 5-1/4 inch (13Ø mm) rigid disk media, utilizing standard Winchester technology. The storage media is contained in the drive in a fixed non-operator removable configuration.

The RD51 controller is a highly integrated module occupying the Extended Communication option slot and has the capability of controlling one ST5Ø6 interface compatible Winchester drive. The controller architecture allows for subsystem extensibility by having sufficient track address and head select bits to support higher capacity drives when available, assuming interface and transfer rate remain unchanged. Connection between drive and controller is by a molded cable which interfaces 34-pin control and 2Ø-pin data connectors.

3.2.4.2 Drive Characteristics

1. Performance Specifications

Formatted capacity	
Per drive	1Ø MB
Per surface	2.5 MB
Per track	8192 bytes
Per sector	512 bytes
Sectors per track	16
Transfer rate	5M bits/sec
Access time	
Track to track	3 msec
Average seek	85 msec, including settle
Maximum seek	2Ø5 msec,including settle
Head Settle	15 msec
Rotational latency	8.33 msec average 16.7 msec maximum

2. Functional Specifications

Rotational speed	3600 rpm ± 1%
Recording density	9074 bpi
Track density	345 tpi
Cylinders	305
Tracks	1220
Disks	2
Physical size	5.75 inch wide x 3.25 inch high x 8.05 inch deep
Weight	5.0 lbs.
Power	
5 Vdc + 5%	50 mv peak-to-peak ripple maximum
0.7 ampere typical 1.0 ampere maximum	
12 Vdc + 5%	75 mv peak-to-peak ripple maximum
1.8 ampere typical 3.5 amperes for 20 sec maximum	
Heat dissipation	25 watts typical 29 watts maximum
Environment	
Temperature	50 deg.F to 122 deg.F
Humidity	20% to 80% relative humidity
Temp. Gradient	20 deg.F/hr.

3.2.4.3 Controller Characteristics

1. Functional Specification

Mechanical	3.9 inch x 12.8 inch module compatible with aft PC100 option slot.
Power	DC 5 Vdc + 5%, 50 mv ripple 1.5 ampere typical, 2.0 amperes maximum +12 Vdc + 5% 75 mv ripple .032 ampere typical, .05 ampere maximum
Environment	DEC STD 102 Class B
Data Transfer	Programmed transfer Full sector buffer
Drives per controller	Single drive

Features	Buffered seek Field formatting capability ST5Ø6 compatible interface Track position status Fixed retries at 8 Field diagnostic circuitry

3.2.4.4 Subsystem Product Specifications

1. Performance Specifications

Error rates

Soft Read Errors	1 per $1Ø^{1Ø}$ bits read
Hard Read Errors	1 per $1Ø^{12}$ bits read
Seek Errors	1 per $1Ø^{6}$ seeks

2. Reliability Specifications

MTBF	9K POH @ 5Ø% duty cycle (11K POH Drv., 15ØK POH Cont.)
MTTR	Less than .5 hours
Fault isolation	.95 probability of isolating drive and controller

4 RX5Ø DRIVE

4.1 GENERAL DESCRIPTION

The RX5Ø subsystem is a 5-1/4-inch flexible diskette drive and a single board controller which enables the PC1ØØ-B to store or retrieve information on one side of each front-loaded diskette. Each diskette can contain up to 4Ø9,6ØØ 8-bit bytes (formatted), allowing a total of 819,2ØØ bytes of storage per device.

4.2 DRIVE CHARACTERISTICS

No. of recorded surfaces	2
No. of diskettes/drive	2
No. of tracks/surface	8Ø
No. of sectors/track	1Ø
No. of bytes/sector	512
No. of bits/byte	8
Capacity (formatted)	
per drive	819,2ØØ bytes
per surface	4Ø9,6ØØ bytes
per track	5,12Ø bytes
Access Time, track to track	6 ms, one track
head load time, including settle time	3Ø ms. max
rotational latency	1ØØ ms typical, 2ØØ ms max.
random access	29Ø ms average
drive motor start	5ØØ ms max.
Transfer rate	25ØK bytes/sec (average)
Disk rotation	3ØØ RPM + 1% , -
Size	5.75 inch wide x 3.25 inch high x 8.5 inch deep
Weight	3.8 pounds

4.3 TRACK FORMAT

Each of the tracks is formatted as described below. Each data field is made up of 512 8-bit bytes, with a total of 1Ø data fields or sectors, numbered Ø1 through ØA (hex) on each track. The following is a description of the track fields.

Description	**No. of Bytes**	**Contents (HEX)**
Pre ID gap	47	4E
ID Fields		
Sync	8	ØØ
Mark	3	A1**
Header IDAM	1	FE
Track Address	1	Track no. (ØØ-4F)
Side Number	1	ØØ
Sector Address	1	Sector n. (Ø1-ØA)
Bytes/sector code	1	Ø2
CRC	1	Calculated header CRC code
Pose ID gap	22	4E
Data Fields		
Sync	12	ØØ
Mark	3	A1**
Data DAM	1	FB
Data	512	2ØH
CRC	2	Calculated data CRC code
Post amble	1	ØØ
Pre-index gap	*7Ø	4E

* This field is written once per track until an index field is encountered.

** The clock bit is missing between bits 4 and 5.

Fields modified by a WRITE operation are:

1. The DATA SYNC field
2. The DATA MARK field
3. The DATA field
4. The DATA CRC field
5. The POST AMBLE field

4.4 HEADER FORMAT

The diskettes are pre-formatted with header data. The header data fields cannot be modified or re-written by the system. The header field is made up of seven 8-bit bytes as follows:

Byte 1:ID Address Mark (IDAM), FE (hex). This byte coupled with the ID SYNC FIELD and MARK field is decoded by the controller to identify the start of a header.

Byte 2:Track Address. This is the absolute binary track address (ØØ to 4F hex). Each sector contains track address information to identify its radial position on 1 of 8Ø separate tracks.

Byte 3:Zeros.

Byte 4:Sector Address. This is the absolute binary sector address (Ø1 to ØA hex). Each sector contains address information to identify its circumferential position on a track. There is no sector ØØ.

Byte 5:Sector Length 02 hex. This byte specifies the number of bytes contained in one sector. The RX50 drive is formatted with 512 bytes per sector.

Byte 6,7: These two bytes represent the cyclical redundancy check characters that are calculated from the first five header bytes.

5 PC100-B FIRMWARE

Note

> References to the 7201 dual channel USART should be considered the same as an 8274 since the two IC parts are equivalent and used interchangeably.

The PC100-B firmware includes two variations of VT102 emulation: "terminal" mode and "console" mode. "Terminal" mode enables PC100-B to act like a VT102 connected to a host computer via the communications port. The "console" mode enables PC100-B to act like a VT102 (without printer port and using FDX data leads only as a protocol) when running programs on the PC100-B.

The firmware provides services to a "user" for console-out, console-in, console-in-status, enable/disable cursor, return version number, change interrupt vector map, ring the keyboard bell, line-at-a-time screen data transfers, initialize interrupt vectors, return clock rate, 16-bit "key data", and keyboard LED control.

Communications and printer port drivers are supplied by the operating system in console mode.

The firmware also provides self-test diagnostics and a minimal bootstrap loader for floppy disks or winchester hard disk option.

5.1 PRODUCT GOALS

The PC100-B VT102 emulation runs a firmware program using the 8088 processor and looks to the user like a VT102. It provides subfunctions in modules usable to other programs. These other programs need to be able to execute similar functions. The VT102 emulation processes incoming character strings in the same manner as a VT102. The VT102 emulation also returns characters to the host in a manner similar to that of VT102 given the same SET-UP environment.

Differences between VT102 and PC100-B emulation of VT102 are listed below.

VT52 emulation within the VT102 emulator performs as a VT102 (for example, VT102 emulation of VT52 includes most VT102 functions such as 132 columns, auto-wrap, split screen, double high, double wide, etc.). The basis for VT102 functionality is the VT102 engineering specification REV A (A-SP-VT102-0-02 A) dated 01-Aug-1981.

5.1.1 Functional Anomalies

The following is a list of deviations from VT1Ø2 functionality, variances with Terminals Interface Architecture (TIA) and other features of the firmware.

1. When printing from the screen in terminal mode and encountering a "blob" character, the VT1Ø2 sends ASCII "SUB" to the printer. The PC1ØØ-B sends the VT1ØØ line-drawing graphics character "blob" bracketed by the appropriate character set selection escape sequences, if required. Also PC1ØØ-B assumes the printer is capable of properly receiving 8-bit DEC STD 169 characters.

2. All 'break key' functions work with keyboard locked but they also cause the keyboard to unlock.

3. At the completion of a 'print cursor line' operation, PC1ØØ-B sends the escape string to restore the printers GØ char set in between the terminating carriage return and line feed. VT1Ø2 sends it after the line feed.

4. Locking the keyboard does stop an auto-repeat but unlocking the keyboard does not restart auto-repeat unless the original key is still the one held down. Any new key must be pressed after the keyboard is unlocked in order to have it auto-repeat.

5. When the SET-UP key is pressed to enter SET-UP mode in the PC1ØØ-B, the key-holding buffer is cleared which causes any unserviced keys to be lost and SET-UP is immediately honored.

6. The printer port baud rate selection and the communications port external clock selection both reside in the same write-only 8Ø88 port. Selecting communications port external clocks can make the printer port baud rate incorrect and selecting printer port baud rates in SET-UP will de-select external clocks for communication. Also the break control bit for the 72Ø1 is in the same write-only register as the number-of-data-bits. The firmware can read the NVM and set this properly for use with terminal mode. An application cannot do this.

7. PC1ØØ-B maintains wrap-pending flag unconditionally and tests it conditionally. VT1Ø2 maintains the flag conditionally and tests it unconditionally. This affects where the next character goes when the auto-wrap mode is changed while the cursor is in the 'line-filled' position.

8. In PC1ØØ-B, the escape sequences to select alternate ROM and alternate ROM special graphics are parsed but ignored.

9. NVM defaults are not the same as VT1Ø2 for the printer port.

1Ø. PC1ØØ-B executes C1 control codes for index, next line, horizontal tab set, reverse index, single shift 2, single shift 3, control sequence introducer. Reception of any C1 control code will abort an escape sequence in process (CSI restarts an escape sequence). 8-bit graphic chars will be treated as if the 8th bit were Ø if received during an escape sequence.

11. PC1ØØ-B always sets insertion/replacement mode to replacement before saving into NVM.

12. The PC1ØØ-B accepts and acts on 8-bit character codes, the VT1Ø2 always strips the 8th bit. If 8-bit codes are received in VT52 mode they will be handled the same as in ANSI mode.

13. Shift out (CTRL/N) and shift in (CTRL/O) in VT52 mode will abort VT52 'graphics' operation if the char set selected is not the 'graphics' set.

14. Terminal mode print functions are implemented via the 'print screen' key on the PC1ØØ-B. VT1Ø2 uses the keypad 'enter' key. PC1ØØ-B 'print screen' is equivalent to VT1Ø2 <shift/enter> and PC1ØØ-B <control/print screen> is equivalent to VT1Ø2 <control/enter>.

15. When hold-screen is in effect, all attempts to 'receive' a character will hang until hold-screen is removed. This includes selections from the opening menu, console out requests, data moves to screen display, terminal mode character reception. In terminal mode the receive buffer will continue to fill, but will not be emptied. When 'full', it will automatically send XOFF if enabled in SET-UP, otherwise, data will be lost if the host does not stop sending. The diagnostic routines in ROM have been given a separate entry to the display process that bypasses the 'hold screen' test.

16. The PC1ØØ-B will parse but ignore the escape sequences to set GØ and G1 to the alternate ROM and alternate ROM special graphics (ESC (1 , ESC (2 , ESC) 1 , ESC) 2). It will also parse but ignore the escape sequences to run self tests (ESC [2 ; Pn y) and the LED control (ESC [Pn q). Also the device status report request (ESC [5 n) will always cause the ready, no malfunctions reply (ESC [Ø n).

17. Serial line SET-UP selections of 7-bit mark and space actually use the 72Ø1 in 8-bit no-parity mode. The mark/space aspect is handled by the firmware drivers in terminal mode. In console mode, the operating system drivers do not make this distinction and set-ups 7M and 7S are the same as 8N in console mode.

18. The HOLD SCREEN key on the PC1ØØ-B does not work the same as the NO SCROLL key on a VT1Ø2. On a VT1Ø2, it sends an XOFF/XON as it toggles back and forth and CTRL/S and CTRL/Q typed from the keyboard can be used to get the same effect. In PC1ØØ-B, setting HOLD SCREEN does not necessarily cause an XOFF to be sent. It sets an internal flag that causes the 'receive character' process to loop until the flag is cleared. This effectively 'hangs' any console output (normal or direct) in console mode. In terminal mode this 'hang' causes the comm receive buffer to fill up until it reaches the high water mark at which point it will send an XOFF if enabled by SET-UP. After the HOLD SCREEN is removed, characters are removed from the receive buffer until the low water mark is reached which causes XON to be sent if enabled. As a result of this method of implementation, PC1ØØ-B honors HOLD SCREEN even in 'local', VT1Ø2 does not.

19. PC100-B resets CAPS LOCK to 'lower case' any time the 'S' (self test) selection is made from the opening menu.

20. The following keys generate escape sequences that end in characters which cause valid selections at opening menu time, cursor arrow keys will select drives to boot from and PF4 in the keypad will select self-test.

21. The PC100-B in VT52 mode honors the origin mode setting, VT102 in VT52 mode does not.

22. In PC100-B, ESC c (reset to initial state) does not reset keypad and cursor keys to their normal modes.

23. In PC100-B, print screen while screen is 'held' is deferred until after 'hold' is removed and char being 'held' is processed.

24. In PC100-B terminal mode, after using 'hold-screen' on incoming data, the last char for display is being 'held'. Entering setup, switching to local, and exiting from setup does not clear the 'hold' state or the char. When 'hold' is finally removed, the char originally being 'held' is displayed before any locally generated characters.

25. In PC100-B any noise on the printer port DTR line can cause an interrupt that will set a flag indicating a printer was once available.

26. In PC100-B, cursor key mode and keypad mode are independent. This agrees with the TIA spec but not the VT102. In the VT102 the cursor keys only send application codes if both cursor and keypad modes are set to 'application'.

27. In PC100-B the TAB character always clears the wrap-pending flag. This agrees with the TIA but not the VT102. As a result auto-wrap will not be the same if TAB is the 81st char in an 80 char line. Char 82 will not wrap but char 83 will. In a VT102, char 82 will wrap.

28. In PC100-B terminal mode, the second XOFF is sent at 'buffer-full'. In VT102 the second XOFF is sent 12 char before 'buffer-full'. Also the PC100-B buffer is 255 char in size, VT102 is 128.

29. PC100-B allows a tab stop in the first column, VT102 does not.

30. Function keys are not ignored when entering the answerback message and produce unpredictable results.

31. PC100-B aborts escape sequence parsing when it finds an intermediate char causing all following characters to be displayed. VT102 aborts the sequence but continues parsing until it finds a final char so the intervening part of the escape sequence does not display.

32. Due to differences in implementation and timing, the PC100-B and VT102 can have different transient appearances where the cursor is concerned. For example, the cursor may appear momentarily and/or in different locations when the same data is sent to both for display.

33. In terminal mode, local, printer controller mode does not send keyboard characters to the printer.

34. Any printer related escape sequence (ANSI or VT52) or keyboard entry is ignored if the printer DTR signal is not asserted at the time.

35. Switching auto-xon/xoff after establishing contact can cause the 'terminal' to hang under the right conditions. Typing an xon (CTRL/Q) in these cases should clear the hung state.

5.2 PERFORMANCE

The performance of the VT1Ø2 emulation is at least equal to that of the actual VT1Ø2. Using pure text for data in jump scroll mode, the VT1Ø2 emulation operates at 96ØØ baud unrestrained as a terminal. It is a goal for it to operate at 38.4K baud as a console.

5.3 NON-GOALS

The non-goals for this program are:

1. Emulation of bugs in the VT1Ø2 software.
2. SET-UP mode identical to that of the VT1Ø2.
3. The VT1Ø2 firmware excludes all VT131 hooks. There is no support for editing, block mode transmit, protected fields, option ROM linkage, etc.
4. All printer baud rates of VT1Ø2 supported.

5.4 GENERAL

The firmware of the PC1ØØ-B provides the following services:

1. Power-up initialization of hardware

2. Self-test diagnostics

3. VT1Ø2 emulation - available in "terminal" and "console" modes

4. Image of Z8ØA RAM space to be loaded

5. Boot loader to read track Ø, sector 1 of floppy or winchester disk

6. Opening menu selection process

7. Automatic shut-off of screen display after 3Ø minutes of non-use, and restoration of display on first activity (any keyboard key or received char).

8. Support of the 15 keyboards supported by the Rainbow 1ØØ (PC1ØØ-A) product.

9. National language power-up and selftest system messages,

10. National language Boot Menu,

11. Implementation of the compose algorithm,

12. National language Set-Up.

13. Choice of DEC 8-bit codes or national replacement characters

The firmware is organized such that the VT102 emulation primitives form the "console" functionality for use by "applications" through the interface layer. When in "terminal" mode a background loop is entered which calls on the "console" primitives and adds the necessary functionality to provide full "terminal" mode.

An interface layer is placed over the "console" primitives to provide an "application" with means of accessing those primitives.

Note

> In "console" mode there is no support provided for the printer or the communication ports. This hardware (7201) must be controlled directly by the operating system.

For "applications" that need more immediate control of the hardware, services are provided to obtain "low level" key information, enable/disable cursor, and transfer data directly to screen RAM.

The interface between the "application" and the firmware is implemented using a software interrupt, with arguments passed and returned in CPU registers.

This leads to a layered structure as diagrammed below. From the firmware view point, the operating system in this example is an "application". It can actually be anything, including another firmware routine.

All entries to firmware routines from external processes are via a software interrupt. This makes the interface release-independent because ROM code loads the proper vectors during initialization.

```
             8088 side
             |--------|---------|---------------|
             |        | NORMAL  |               |
             |   SU   | USER    |               |
             |     |----------------|    SU     |
             |     |   OPERATING    |           |
             |     |    SYSTEM      |           |
*************************************           |
*   VT102    *    INTERFACE         *           |
* 'TERMINAL' *      LAYER     X     *           |
*************************************           |
*     VT102 'CONSOLE' PRIMITIVES    *           |
*************************************--- |------|
| VIDEO | KBD | PRINTER |     COMM       | OPT  |
| HDWR  |     |  PORT   |     PORT       | HDWR |
|-----------------------------------------------|
```

SU indicates sophisticated user

***** indicates located in ROM

5.4.1 Text Strings

All text strings are located in a single section of the code space so they may be changed with no affect on ROM code (foreign languages). The text strings are accessed by a table of pointers which remains in a fixed location so routines do not need to know the exact text locations. This table and its associated text strings are in one ROM to minimize changes required for other language versions. The keyboard key-to-code mapping tables are also in this same ROM.

The total amount of ROM space allotted to text strings cannot be increased. There is no restriction on individual string sizes, only total bytes used and order of messages.

5.4.2 Character Sets

As with the VT1Ø2, the VT1Ø2 emulation supports the following character sets: UK, USASCII, and Special Graphics.

The character generator ROM also contains the displayable right half of the DEC multinational character set, GR (shown as the DEC supplemental graphic set in DEC STD 169). These characters are accessed by direct writing of data into the screen via interrupt 4Ø calls or by reception of the corresponding 8-bit code. The character generator ROM also contains space for 31 additional displayable characters reserved for future use.

The "console" VT1Ø2 accepts 8-bit character codes to display the alternate characters. The "terminal" VT1Ø2 accepts 7- or 8-bit codes (depending on comm port parameters) and displays characters based on character set mapping through escape sequences. The "console" also works with escape sequence character set mapping and 7-bit characters.

It also accepts 8-bit C1 control codes for index, next line, horizontal tab set, reverse index, single shift 2, single shift 3, control sequence introducer. Any C1 control code will abort an escape sequence in process and CSI will restart it.

Table 4 is a table of the characters and corresponding codes available in the PC1ØØB for display. The codes are actually a part of the address for the bit-map of that character in the character generator ROM. They are the upper 8 bits of the address; the lower 4 bits select the proper scan line within the character. There are 6 unused "scan lines" at the end of each character (uses 1Ø out of 16).

Note

All undefined and reserved characters are indicated by a reversed question mark.

Table 4. Displayable Characters and Corresponding Codes

CHAR CODE IN RAM 8 BITS (HEX)	CHAR CODE RCVD 7 BITS	CHAR SET (DEFINES RULES FOR RCVD TO RAM TRANSLATE)	NAME OF CHARACTER
ØØ	ØØ	ALL	NULL, IGNORED ON RCV, DISPLAYS A BLANK
	5F	SPECL GRAPHICS	BLANK
Ø1	6Ø	SPECL GRAPHICS	DIAMOND
Ø2	61	SPECL GRAPHICS	CHECKERBOARD (BLOB)
Ø3	62	SPECL GRAPHICS	HT (HORIZONTAL TAB)
Ø4	63	SPECL GRAPHICS	FF (FORM FEED)
Ø5	64	SPECL GRAPHICS	CR (CARRIAGE RETURN)
Ø6	65	SPECL GRAPHICS	LF (LINE FEED)
Ø7	66	SPECL GRAPHICS	DEGREE SYMBOL
Ø8	67	SPECL GRAPHICS	PLUS/MINUS SIGN
Ø9	68	SPECL GRAPHICS	NL (NEW LINE)
ØA	69	SPECL GRAPHICS	VT (VERTICAL TAB)
ØB	6A	SPECL GRAPHICS	LOWER RIGHT CORNER
ØC	6B	SPECL GRAPHICS	UPPER RIGHT CORNER
ØD	6C	SPECL GRAPHICS	UPPER LEFT CORNER
ØE	6D	SPECL GRAPHICS	LOWER LEFT CORNER
ØF	6E	SPECL GRAPHICS	CROSSING LINES
1Ø	6F	SPECL GRAPHICS	HORIZONTAL LINE, SCAN 1
11	7Ø	SPECL GRAPHICS	HORIZONTAL LINE, SCAN 3
12	71	SPECL GRAPHICS	HORIZONTAL LINE, SCAN 5
13	72	SPECL GRAPHICS	HORIZONTAL LINE, SCAN 7
14	73	SPECL GRAPHICS	HORIZONTAL LINE, SCAN 9
15	74	SPECL GRAPHICS	LEFT 'T'
16	75	SPECL GRAPHICS	RIGHT 'T'
17	76	SPECL GRAPHICS	BOTTOM 'T'
18	77	SPECL GRAPHICS	TOP 'T'
19	78	SPECL GRAPHICS	VERTICAL BAR
1A	79	SPECL GRAPHICS	LESS-THAN OR EQUAL
1B	7A	SPECL GRAPHICS	GREATER-THAN OR EQUAL
1C	7B	SPECL GRAPHICS	PI SYMBOL
1D	7C	SPECL GRAPHICS	NOT EQUAL SIGN
1E	7D	SPECL GRAPHICS	U.K. POUND STERLING SIGN
	23	UK	
1F	7E	SPECL GRAPHICS	CENTERED DOT
2Ø	2Ø	UK/USASCII SPECL GRAPHICS	SPACE
21	21	UK/USASCII SPECL GRAPHICS	EXCLAMATION POINT
22	22	UK/USASCII SPECL GRAPHICS	DOUBLE QUOTES
23	23	USASCII SPECL GRAPHICS	NUMBER SIGN (POUND SIGN)
24	24	UK/USASCII SPECL GRAPHICS	DOLLAR SIGN
25	25	UK/USASCII SPECL GRAPHICS	PER-CENT SIGN

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR CODE IN RAM 8 BITS (HEX)	CHAR CODE RCVD 7 BITS	CHAR SET (DEFINES RULES FOR RCVD TO RAM TRANSLATE)	NAME OF CHARACTER
26	26	UK/USASCII SPECL GRAPHICS	AMPERSAND SIGN
27	27	UK/USASCII SPECL GRAPHICS	SINGLE QUOTE
28	28	UK/USASCII SPECL GRAPHICS	LEFT PARENTHESES
29	29	UK/USASCII SPECL GRAPHICS	RIGHT PARENTHESES
2A	2A	UK/USASCII SPECL GRAPHICS	ASTERISK SIGN
2B	2B	UK/USASCII SPECL GRAPHICS	PLUS SIGN
2C	2C	UK/USASCII SPECL GRAPHICS	COMMA
2D	2D	UK/USASCII SPECL GRAPHICS	DASH (MINUS SIGN)
2E	2E	UK/USASCII SPECL GRAPHICS	PERIOD
2F	2F	UK/USASCII SPECL GRAPHICS	SLASH (FRACTION BAR)
3Ø	3Ø	UK/USASCII SPECL GRAPHICS	NUMERAL Ø
31	31	UK/USASCII SPECL GRAPHICS	NUMERAL 1
32	32	UK/USASCII SPECL GRAPHICS	NUMERAL 2
33	33	UK/USASCII SPECL GRAPHICS	NUMERAL 3
34	34	UK/USASCII SPECL GRAPHICS	NUMERAL 4
35	35	UK/USASCII SPECL GRAPHICS	NUMERAL 5
36	36	UK/USASCII SPECL GRAPHICS	NUMERAL 6
37	37	UK/USASCII SPECL GRAPHICS	NUMERAL 7
38	38	UK/USASCII SPECL GRAPHICS	NUMERAL 8
39	39	UK/USASCII SPECL GRAPHICS	NUMERAL 9
3A	3A	UK/USASCII SPECL GRAPHICS	COLON
3B	3B	UK/USASCII SPECL GRAPHICS	SEMI-COLON
3C	3C	UK/USASCII SPECL GRAPHICS	LEFT ANGLE BRACKET
3D	3D	UK/USASCII SPECL GRAPHICS	EQUALS SIGN

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR CODE IN RAM 8 BITS (HEX)	CHAR CODE RCVD 7 BITS	CHAR SET (DEFINES RULES FOR RCVD TO RAM TRANSLATE)	NAME OF CHARACTER
3E	3E	UK/USASCII SPECL GRAPHICS	RIGHT ANGLE BRACKET
3F	3F	UK/USASCII SPECL GRAPHICS	QUESTION MARK
4Ø	4Ø	UK/USASCII SPECL GRAPHICS	AT SIGN
41	41	UK/USASCII SPECL GRAPHICS	CAPITAL A
42	42	UK/USASCII SPECL GRAPHICS	CAPITAL B
43	43	UK/USASCII SPECL GRAPHICS	CAPITAL C
44	44	UK/USASCII SPECL GRAPHICS	CAPITAL D
45	45	UK/USASCII SPECL GRAPHICS	CAPITAL E
46	46	UK/USASCII SPECL GRAPHICS	CAPITAL F
47	47	UK/USASCII SPECL GRAPHICS	CAPITAL G
48	48	UK/USASCII SPECL GRAPHICS	CAPITAL H
49	49	UK/USASCII SPECL GRAPHICS	CAPITAL I
4A	4A	UK/USASCII SPECL GRAPHICS	CAPITAL J
4B	4B	UK/USASCII SPECL GRAPHICS	CAPITAL K
4C	4C	UK/USASCII SPECL GRAPHICS	CAPITAL L
4D	4D	UK/USASCII SPECL GRAPHICS	CAPITAL M
4E	4E	UK/USASCII SPECL GRAPHICS	CAPITAL N
4F	4F	UK/USASCII SPECL GRAPHICS	CAPITAL O
5Ø	5Ø	UK/USASCII SPECL GRAPHICS	CAPITAL P
51	51	UK/USASCII SPECL GRAPHICS	CAPITAL Q
52	52	UK/USASCII SPECL GRAPHICS	CAPITAL R
53	53	UK/USASCII SPECL GRAPHICS	CAPITAL S
54	54	UK/USASCII SPECL GRAPHICS	CAPITAL T
55	55	UK/USASCII SPECL GRAPHICS	CAPITAL U

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR CODE IN RAM 8 BITS (HEX)	CHAR CODE RCVD 7 BITS	CHAR SET (DEFINES RULES FOR RCVD TO RAM TRANSLATE)	NAME OF CHARACTER
56	56	UK/USASCII SPECL GRAPHICS	CAPITAL V
57	57	UK/USASCII SPECL GRAPHICS	CAPITAL W
58	58	UK/USASCII SPECL GRAPHICS	CAPITAL X
59	59	UK/USASCII SPECL GRAPHICS	CAPITAL Y
5A	5A	UK/USASCII SPECL GRAPHICS	CAPITAL Z
5B	5B	UK/USASCII SPECL GRAPHICS	LEFT SQUARE BRACKETS
5C	5C	UK/USASCII SPECL GRAPHICS	BACK-SLASH
5D	5D	UK/USASCII SPECL GRAPHICS	RIGHT SQUARE BRACKETS
5E	5E	UK/USASCII SPECL GRAPHICS	CIRCUMFLEX
5F	5F	UK/USASCII SPECL GRAPHICS	UNDERLINE
6Ø	6Ø	UK/USASCII	ACCENT GRAVE
61	61	UK/USASCII	LOWER CASE A
62	62	UK/USASCII	LOWER CASE B
63	63	UK/USASCII	LOWER CASE C
64	64	UK/USASCII	LOWER CASE D
65	65	UK/USASCII	LOWER CASE E
66	66	UK/USASCII	LOWER CASE F
67	67	UK/USASCII	LOWER CASE G
68	68	UK/USASCII	LOWER CASE H
69	69	UK/USASCII	LOWER CASE I
6A	6A	UK/USASCII	LOWER CASE J
6B	6B	UK/USASCII	LOWER CASE K
6C	6C	UK/USASCII	LOWER CASE L
6D	6D	UK/USASCII	LOWER CASE M
6E	6E	UK/USASCII	LOWER CASE N
6F	6F	UK/USASCII	LOWER CASE O
7Ø	7Ø	UK/USASCII	LOWER CASE P
71	71	UK/USASCII	LOWER CASE Q
72	72	UK/USASCII	LOWER CASE R
73	73	UK/USASCII	LOWER CASE S
74	74	UK/USASCII	LOWER CASE T
75	75	UK/USASCII	LOWER CASE U
76	76	UK/USASCII	LOWER CASE V
77	77	UK/USASCII	LOWER CASE W
78	78	UK/USASCII	LOWER CASE X
79	79	UK/USASCII	LOWER CASE Y
7A	7A	UK/USASCII	LOWER CASE Z
7B	7B	UK/USASCII	LEFT BRACES
7C	7C	UK/USASCII	VERTICAL LINE (BROKEN)

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR	CHAR	CHAR SET	NAME OF CHARACTER
CODE	CODE	(DEFINES RULES FOR	
IN	RCVD	RCVD TO RAM	
RAM	7 BITS	TRANSLATE)	
8 BITS			
(HEX)			
7D	7D	UK/USASCII	RIGHT BRACES
7E	7E	UK/USASCII	TILDE
7F	7F	ALL	RESERVED FOR CHARACTER GENERATOR ROM CHECKSUM DISPLAYS JUNK, SHOULD NOT BE USED
8Ø			UNFILLED RECTANGLE FOR 'AUTO-BLANKED CURSOR'
81			RESERVED FOR FUTURE USE
82			RESERVED FOR FUTURE USE
83			RESERVED FOR FUTURE USE
84			RESERVED FOR FUTURE USE
85			RESERVED FOR FUTURE USE
86			RESERVED FOR FUTURE USE
87			RESERVED FOR FUTURE USE
88			RESERVED FOR FUTURE USE
89			RESERVED FOR FUTURE USE
8A			RESERVED FOR FUTURE USE
8B			RESERVED FOR FUTURE USE
8C			RESERVED FOR FUTURE USE
8D			RESERVED FOR FUTURE USE
8E			RESERVED FOR FUTURE USE
8F			RESERVED FOR FUTURE USE
9Ø			RESERVED FOR FUTURE USE
91			RESERVED FOR FUTURE USE
92			RESERVED FOR FUTURE USE
93			RESERVED FOR FUTURE USE
94			RESERVED FOR FUTURE USE
95			RESERVED FOR FUTURE USE
96			RESERVED FOR FUTURE USE
97			RESERVED FOR FUTURE USE
98			RESERVED FOR FUTURE USE
99			RESERVED FOR FUTURE USE
9A			RESERVED FOR FUTURE USE
9B			RESERVED FOR FUTURE USE
9C			RESERVED FOR FUTURE USE
9D			RESERVED FOR FUTURE USE
9E			RESERVED FOR FUTURE USE
9F			RESERVED FOR FUTURE USE
AØ			NOT USED
A1			INVERTED EXCLAMATION POINT
A2			CENT SIGN
A3			U.K. POUND STERLING SIGN
A4			RESERVED (DEC STD 169)
A5			YEN SIGN
A6			RESERVED (DEC STD 169)
A7			SECTION SIGN
A8			GENERAL CURRENCY SIGN
A9			COPYRIGHT SIGN

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR	CHAR	CHAR SET	NAME OF CHARACTER
CODE	CODE	(DEFINES RULES FOR	
IN	RCVD	RCVD TO RAM	
RAM	7 BITS	TRANSLATE)	
8 BITS			
(HEX)			
AA			FEMININE ORDINAL INDICATOR
AB			LEFT ANGLE QUOTATION MARKS
AC			RESERVED (DEC STD 169)
AD			RESERVED (DEC STD 169)
AE			RESERVED (DEC STD 169)
AF			RESERVED (DEC STD 169)
BØ			DEGREE SIGN
B1			PLUS/MINUS SIGN
B2			SUPERSCRIPT 2
B3			SUPERSCRIPT 3
B4			RESERVED (DEC STD 169)
B5			MICRO SIGN
B6			PARAGRAPH SIGN, PILCROW
B7			MIDDLE DOT
B8			RESERVED (DEC STD 169)
B9			SUPERSCRIPT 1
BA			MASCULINE ORDINAL INDICATOR
BB			RIGHT ANGLE QUOTATION MARK
BC			FRACTION 1/4
BD			FRACTION 1/2
BE			RESERVED (DEC STD 169)
BF			INVERTED QUESTION MARK
CØ			CAPITAL A WITH GRAVE ACCENT
C1			CAPITAL A WITH ACUTE ACCENT
C2			CAPITAL A WITH CIRCUMFLEX ACCENT
C3			CAPITAL A WITH TILDE
C4			CAPITAL A WITH UMLAUT
C5			CAPITAL A WITH RING
C6			CAPITAL AE DIPTHONG
C7			CAPITAL C WITH CEDILLA
C8			CAPITAL E WITH GRAVE ACCENT
C9			CAPITAL E WITH ACUTE ACCENT
CA			CAPITAL E WITH CIRCUMFLEX ACCENT
CB			CAPITAL E WITH UMLAUT
CC			CAPITAL I WITH GRAVE ACCENT
CD			CAPITAL I WITH ACUTE ACCENT
CE			CAPITAL I WITH CIRCUMFLEX ACCENT
CF			CAPITAL I WITH UMLAUT
DØ			RESERVED (DEC STD 169)
D1			CAPITAL N WITH TILDE
D2			CAPITAL O WITH GRAVE ACCENT
D3			CAPITAL O WITH ACUTE ACCENT
D4			CAPITAL O WITH CIRCUMFLEX ACCENT
D5			CAPITAL O WITH TILDE
D6			CAPITAL O WITH UMLAUT

Table 4. Displayable Characters and Corresponding Codes (Continued)

CHAR CODE IN RAM 8 BITS (HEX)	CHAR CODE RCVD 7 BITS	CHAR SET (DEFINES RULES FOR RCVD TO RAM TRANSLATE)	NAME OF CHARACTER
D7			CAPITAL OE DIPTHONG
D8			CAPITAL O WITH SLASH
D9			CAPITAL U WITH GRAVE ACCENT
DA			CAPITAL U WITH ACUTE ACCENT
DB			CAPITAL U WITH CIRCUMFLEX ACCENT
DC			CAPITAL U WITH UMLAUT
DD			CAPITAL Y WITH UMLAUT
DE			RESERVED (DEC STD 169)
DF			GERMAN SMALL SHARP S
EØ			LOWER CASE A WITH GRAVE ACCENT
E1			LOWER CASE A WITH ACUTE ACCENT
E2			LOWER CASE A WITH CIRCUMFLEX ACCENT
E3			LOWER CASE A WITH TILDE
E4			LOWER CASE A WITH UMLAUT
E5			LOWER CASE A WITH RING
E6			LOWER CASE AE DIPTHONG
E7			LOWER CASE C WITH CEDILLA
E8			LOWER CASE E WITH GRAVE ACCENT
E9			LOWER CASE E WITH ACUTE ACCENT
EA			LOWER CASE E WITH CIRCUMFLEX ACCENT
EB			LOWER CASE E WITH UMLAUT
EC			LOWER CASE I WITH GRAVE ACCENT
ED			LOWER CASE I WITH ACUTE ACCENT
EE			LOWER CASE I WITH CIRCUMFLEX ACCENT
EF			LOWER CASE I WITH UMLAUT
FØ			RESERVED (DEC STD 169)
F1			LOWER CASE N WITH TILDE
F2			LOWER CASE O WITH GRAVE ACCENT
F3			LOWER CASE O WITH ACUTE ACCENT
F4			LOWER CASE O WITH CIRCUMFLEX ACCENT
F5			LOWER CASE O WITH TILDE
F6			LOWER CASE O WITH UMLAUT
F7			LOWER CASE OE DIPTHONG
F8			LOWER CASE O WITH SLASH
F9			LOWER CASE U WITH GRAVE ACCENT
FA			LOWER CASE U WITH ACUTE ACCENT
FB			LOWER CASE U WITH CIRCUMFLEX ACCENT
FC			LOWER CASE U WITH UMLAUT
FD			LOWER CASE Y WITH UMLAUT
FE			RESERVED (DEC STD 169)
FF			NOT ALLOWED, THIS IS 'TERMINATION' CODE

5.5 START-UP/SHUT DOWN/RESET

5.5.1 Power-Up Initialization

This process initializes all the hardware, including any indicated EXPANSION RAM options and all the flags, pointers, etc. Power-up must also read in the contents of the NVM and configure itself accordingly. The NVM contains information on memory configuration for use by self-test diagnostics. It shows which 64K memory blocks are installed.

Note

> NVM contents refers to the data stored in non-volatile memory, normally by means of the Set-up process, that affects the system operation as defined by the various parameters.

The NVM consists of two elements, the non-volatile storage part and a volatile shadow RAM part. A recall operation transfers the contents of the non-volatile storage part to the shadow RAM part. A store operation transfers the contents of the shadow RAM part into the non-volatile storage part, destroying any previous contents. Data can only be transferred to/from the CPU from the shadow RAM part, and only when the NVM is not either in the recall or store modes of operation. A CRC is calculated and stored along with the shadow RAM data. This CRC is verified after any recall operation. If the CRC does not verify, the recovery procedure is:

1. A second recall is done.
2. If second recall is OK, continue as normal.
3. If second recall is also bad, it could be due to two reasons:
   a. first time ever used, contains random data
   b. bad NVM
4. Put defaults into shadow RAM, store into NVM and display NVM INITIALIZED TO DEFAULTS message and continue as normal.

5.5.2 Selection of Keyboard/Language

The NVM maintains a binary valued variable that indicates whether or not a keyboard has been selected. This variable is tested just before the opening menu is displayed. If a keyboard has been selected, then everything proceeds as normal using the keyboard and the implied language. If a keyboard has not been selected, a screen is displayed that allows the operator to make a keyboard selection. The operator may choose to skip the selection process which then uses the current language of the cluster. The selection process uses keys that are the same in all languages, arrow keys to make choice and 'select' to make it happen. To change a keyboard selection after the initial selection is made the operator must enter Set-up, set the keyboard to 'unselected', save in NVM, and then reset the system to get the selection screen. Any error message that occurs when a keyboard selection has not been made will appear in the default language of the current cluster. Error messages normally appear in the selected language. In Set-up, the choice of keyboard 'selected' or 'not selected' is the only selection the operator can make. Following a 'not selected' choice, the operator must save the set-ups into NVM and then reset the system to get the keyboard selection choices in order to select a new keyboard or remain with 'not selected'. The default is 'not selected'.

5.5.3 Power-Off

No attempt to do anything special is made on power-off.

5.5.4 Hardware Resetting

The system resets similar to a VT1Ø2. Enter Set-Up mode and press the Ctrl key and the Set-Up key simultaneously. The resetting is accomplished by jumping to a separate location at the start of the self-test diagnostics. Entry at this point distinguishes it from a power-up start. This assumes the system is still capable of entering Set-Up. If not, the only recovery is to cycle the power off and on.

CAUTION

> Leaving interrupts disabled for 1ØØ ms or more in the 8Ø88 CPU causes the hardware failure detect circuitry to be activated. It is possible to disable the hardware failure detect circuitry, if it is mandatory to leave interrupts masked for a longer period of time. This procedure may adversely affect the video display and any real-time dependent system operations; use with care and discretion. The following is the correct procedure: 1.) disable interrupts using a CLI instruction, 2.) disable the hardware failure detect circuit by writing a ØØH to 8Ø88 I/O port 1ØC (hex). The hardware failure detect circuit will be re-enabled automatically, after the 8Ø88 interrupt mask has been re-enabled, using the STI instruction.

5.5.5 RAM Parity Error

When the expansion RAM is installed, a parity error activates the NMI input. The NMI causes the ram option failure message to be displayed on the screen and causes the bell to beep. No more options are allowed except to enter Set-Up and reset the system.

If an operating system needs to handle parity errors itself, it takes over the NMI interrupt vector.

5.6 MEMORY-MAPPED VIDEO ACCESS SERVICE

A "sophisticated user" accesses the screen/attribute RAM directly for fast data transfers.

Note

> The character stored in the screen RAM by this process is actually a code (not necessarily ASCII). This code is bits 4-11 of the address in the character generator ROM for the first scan line of the bit pattern of that character.

The screen display is a linked list and there are several related tables, flags, and pointers that must be retained. It is imperative, then, that the "sophisticated user" follow certain restrictions when directly accessing the screen/attribute RAM.

1. In order to guarantee a known starting condition and remove all effects of scrolling, double height, double width line, top and bottom margins, origin mode, and so on, the user must send the escape sequence to set the desired screen width. These also place the cursor at the top, left screen position and clear the screen.
   for 80 columns escape [? 3 l
   for 132 columns escape [? 3 h

   (Note that a lower case L is used here.)

2. The standard escape sequences to position the cursor and set double height and width lines can be used. The user is responsible for keeping track of what lines have been modified so no attempt is made to put more characters on a line than it can hold.

CAUTION

> Each line ends with a termination code and pointer to the next line. Video display hardware uses these in its operation. Destroying these values in either screen display or attribute space causes unpredictable results on the display.

3. Each character (data) screen position has a related attribute. When the screen width escape sequence initializes the screen, these attributes are all set to the "off" condition.

Bit assignments for character attributes are:

Bit Ø = Reverse Video
Ø = normal
1 = reverse video

Bit 1 = Bold
Ø = bold
1 = not bold

Bit 2 = Blink
Ø = blink
1 = not blink

Bit 3 = Underscore

Ø = underscore
1 = not underscore

4. Contents of character locations can be changed at any time. However due to the way the cursor is implemented, attributes at the cursor position cannot be changed at will. When the user wishes to change the attributes of the character at the cursor position, he must use the DISABLE CURSOR function. This removes all cursor-related attribute affects. After the attributes have been changed as desired, the user must use the ENABLE CURSOR function to restore the cursor to operation. See subheads 6.1.5 and 6.1.6 for these functions.

5.7 KEY ACCESS SERVICES

This allows a "sophisticated user" to obtain low level key data (for example, a code for left arrow instead of an escape sequence) to simplify the process of detecting special keys. If no key is waiting for detection a "no data" status is returned.

Two different layers of access are provided. The lowest level (Level 1) gives a unique 16-bit code for any key in combination with any or all of the Shift, Caps Lock, or Ctrl keys. It also identifies the function keys with a unique code (reference section 6.1.4). The highest level (Level 2) is the same as the VT1Ø2 generates, plus additional 8-bit codes for certain keys and foreign keyboards.

Certain keys are trapped out for special processing and are never seen in the buffer. The following keys cannot be remapped:

KEY	POSITION
Hold Screen	G99
Set-Up	GØ1
Control	C99
Caps Lock	CØØ
Shift Keys	B99,B11
Compose Character	A99

Any other keys may be remapped by a "console" mode user.

6 FUNCTIONAL DEFINITION

6.1 OPERATIONAL DESCRIPTION

The VT102 emulation is always resident in the PC100-B's ROM, and can be used even without working floppy disks.

The VT102 "console" emulator must be completely interrupt driven. The "terminal" VT102 uses a "background" routine to add the additional functions of printer port and modem protocols. The VT102 HDX modem protocols are not supported by the firmware.

The VT102 emulator must operate in two distinct modes. Terminal mode provides VT102 capabilities. Console mode also has VT102 capabilities with these exceptions: no printer port, no local echo, and modem protocol equivalent to full duplex data leads only.

6.2 CLUSTERING OF LANGUAGES

The amount of character storage space required to include all language variations in a single set of 128K bit ROMs is too great. Therefore, the languages are 'clustered' in 5 sets of 3 languages. English is part of all sets. The first set consists of English, French, and German. This set covers keyboards for U.S., UK, Ireland, English and French Canada, French Belgium, Germany, Austria, German and French, Switzerland, and France. This accounts for approximately 75% of the European market. The second set consists of English, Dutch, and French. The third set consists of English, Swedish and Finnish. The fourth set consists of English, Norwegian, and Danish. The fifth set consists of English, Italian, and Spanish. The keyboard selection ID is stored in NVM. Two nibbles are assigned to the ID to allow for expansion beyond 16 keyboard variations. The keyboard IDs and their associated languages are assigned as shown in Table 5.

Table 5. Keyboard ID Codes and Associated Languages

KEYBOARD ID	LANGUAGE	SET 1	SET 2	SET 3	SET 4	SET 5
0 = US	English	X	X	X	X	X
1 = CANADIAN FR	French	X	X			
2 = BRITISH	English	X	X	X	X	X
3 = FINNISH	Finnish			X		
4 = SWEDISH	Swedish			X		
5 = NORWEGIAN	Norwegian				X	
6 = DANISH	Danish				X	
7 = SPANISH	Spanish					X
8 = GERMAN/AUSTRIAN	German	X				
9 = SWISS FR	French	X	X			
10 = SWISS GER	German	X				
11 = DUTCH	Dutch		X			
12 = FLEMISH	Dutch		X			
13 = FRENCH	French	X	X			
14 = ITALIAN	Italian					X

Table 5 ID codes were chosen for best code efficiency in the keyboard translation process. A table is associated with each cluster ROM that defines the allowable keyboards and the associated relative languages for those keyboards. This enables the keyboard selection routine to allow all the choices with default languages assigned to keyboards whose natural language is not in the current cluster. Table 5 contains one entry for each of the 15 (or more) keyboards. Each entry contains a number which defines the 'relative language' for that keyboard. For example, in cluster 1, English is relative language Ø, French is 1, German is 2. These relative language numbers are used as an index into a group of text strings that are for the same message. The text strings are broken into two types; those that are the same in all languages, and those that are different as a function of language. The 'fixed' text strings are accessed in the same manner as currently used, pointer to 'tag name' in diagnostics and 'terminator count' in Set-Up.

If the string is different by language, it is accessed in diagnostics by counting terminators according to the 'relative language' in use after first getting the start of the group of strings by the usual method. The two groups of text types are kept together with the 'fixed' type first in order. The type of algorithm to use is then determined by comparing the string location pointer with the start address of the first of the 'different' strings.

Set-Up uses a method of counting string terminators to access a message. The 2-group approach works in Set-Up by multiplying the message number by the number of languages supported and then counting that number of terminators to get to the block of text strings for that message. Then the language index value advances that many more terminators to get the actual text string desired. This technique is independent of number of languages, size of strings, and allows for conservation of data space by extracting any common elements.

6.3 TRANSMITTED CHARACTERS

The LK2Ø1 keyboard generates a code for each key which identifies the physical location of that key on the keyboard. These position codes are converted to character codes by means of ROM-resident language tables. The ROM must be mapped by language to the keyboard (different keycap legends). Two different types of codes are passed to routines requesting keyboard data. These types are designated as Level 1 and Level 2. Level 2 is the standard 7 bit codes transmitted by a VT1Ø2 with the addition of the DEC STD 169 multi-national character codes being sent in the context of the current language option. At Level 2 (terminal mode) all keys not defined in a normal VT1Ø2 send no code and cause the bell to beep. In console mode, level 2, the function keys not defined in a normal VT1Ø2 send escape sequences. Level 2 supports all the VT1Ø2 key-generated escape sequences (cursor and keypad keys) in the current key mode context (keypad numeric/application, cursor key normal/application). In addition foreign keyboard support includes a correspondence/data processing mode that allows up to four different character codes per key. This mode is a Set-Up parameter and allows a foreign keyboard user to access key codes normally lost because of special character requirements.

6.3.1 Multi-National 8-Bit/National 7-Bit Character Selection

The invoking of the 8-bit/7-bit character translation is provided as part of the binary parameter field. The default is 8-bit operation. The display for this Set-Up parameter will be the 27th location in the major field marked 'PARAM SET'.

The minor field display will be:

CHAR		CODES
Ø	=	DEC-8
1	=	7-bit

6.3.2 National 7-Bit Character Codes

A shell is placed over the keyboard output function and the display input function that conditionally translates 8-bit/7-bit character codes. The translation is dependent on the state of an NVM binary valued variable to select either 8-bit multinational or 7-bit national language operation. The translation process only applies to the 'normal' console/terminal I/O paths; it does not apply to the extended console or 16-bit keyboard paths. Choice of the 7-bit National Replacement Characters (NRC) brings in a shell that performs the translation (if required). There are two parts to the shell; keyboard input and video display. NRC character sets are keyboard related; there is one and only one NRC set available to a particular keyboard. The current mappings of keyboard to NRC are as follows:

Keyboard	NRC table to be used
American	No table
French/Canada	French/Canada
UK	UK
Finnish	Finnish
Swedish	Swedish
Norwegian	Norwegian
Danish	Danish
Spanish	Spanish
German	German
Swiss/French	Swiss
Swiss/German	Swiss
Dutch	Dutch
Flemish	French
French	French
Italian	Italian

If there is no table, then no replacement will be attempted.

6.3.2.1 Keyboard Input Algorithm - No replacement is attempted if we are in DEC 8-bit mode. No replacement is attempted if there is no NRC table associated with a keyboard. No replacement is attempted in escape or control sequences. If a 7-bit code is encountered, it is looked up in the NRC for the current keyboard. If it is found then no code is returned and the keyboard bell is rung. If it is not found the 7-bit code is sent as normal. If an 8-bit code is encountered, it is looked up in the NRC for the current keyboard. If it is found it is replaced by the appropriate

7-bit code, also found in the NRC table, and it is this 7-bit code which is returned as the character entered. If it is not found then no code is returned and the bell is rung as above.

6.3.2.2 Video Display Algorithm - No replacement is attempted if we are in DEC 8-bit mode. No replacement is attempted if there is no NRC table associated with a keyboard. No replacement is attempted if the character set is other than USASCII. If a replacement 7-bit code is encountered, it is looked up in the NRC for the current keyboard. If it is found, it is replaced by the appropriate 8-bit code, also found in the NRC table and it is this 8-bit code that is displayed. If it is not found, it is the 7-bit character which is displayed. If an 8-bit code is encountered, it is displayed as normal.

NRC **TABLES**

1 **French/Canadian.**

8-bit character	7-bit character
lower case a with grave	@
lower case a with circumflex	[
lower case c with cedilla	\
lower case e with grave	}
lower case e with acute	{
lower case e with circumflex	]
lower case i with circumflex	^
lower case o with circumflex	`
lower case u with grave	|
lower case u with circumflex	~

2 **Finnish.**

8-bit character	7-bit character
upper A with umlaut	[
upper A with ring	]
upper O with umlaut	\
upper U with umlaut	^
lower a with umlaut	{
lower a with ring	}
lower e with acute	`
lower o with umlaut	|
lower u with umlaut	~

3 **Swedish.**

8-bit character	7-bit character
upper E with acute	@
upper A with umlaut	[
upper A with ring	]
upper O with umlaut	\
upper U with umlaut	^
lower a with umlaut	{
lower a with ring	}
lower e with acute	`
lower o with umlaut	|

3 **Swedish** (continued)

8-bit character	7-bit character
lower u with umlaut	~

4 **Norwegian/Danish.**

8-bit character	7-bit character
upper A with umlaut	@
upper A with ring	]
upper AE dipthong	[
upper O with slash	\
upper U with umlaut	^
lower a with umlaut	`
lower a with ring	}
lower ae dipthong	{
lower o with slash	|
lower u with umlaut	~

5 **Spanish.**

8-bit character	7-bit character
inverted exclamation mark	[
british pound	#
section sign	@
degree sign	{
inverted question mark	]
upper N with tilda	\
lower c with cedilla	}
lower n with tilda	|

6 **German.**

8-bit character	7-bit character
section sign	@
upper A with umlaut	[
upper O with umlaut	\
upper U with umlaut	]
sharp SS	~
lower a with umlaut	{
lower o with umlaut	|
lower u with umlaut	}

7 **French.**

8-bit character	7-bit character
british pound	#
section sign	]
degree sign	[
lower a with grave	@
lower c with cedilla	\
lower e with grave	}
lower e with acute	{

7 **French** (continued)

8-bit character	7-bit character
lower u with grave	|
Umlaut sign	~

8 **Italian.**

8-bit character	7-bit character
british pound	#
section sign	@
degree sign	[
lower i with grave	~
lower a with grave	{
lower c with cedilla	\
lower e with grave	}
lower e with acute	]
lower o with grave	|
lower u with grave	

9 **UK.**

8-bit character	7-bit character
British Pound sign	#

1Ø **Swiss.**

8-bit character	7-bit character
small u with grave	number sign, #
small a with grave	commercial at, @
small e with acute	left square bracket, [
small c with cedilla	backslash, \
small e with circumflex	right square bracket,]
small i with circumflex	circumflex, ^
small e with grave	underline, _
small o with circumflex	grave accent, `
small a with umlaut	left curly bracket, {
small o with umlaut	vertical line, |
small u with umlaut	right curly bracket, }
small u with circumflex	tilde, ~

11 **Dutch.**

8-bit character	7-bit character
british pound	#
3/4 sign *	@
ij sign **	[
1/2 sign	\
vertical bar	]
umlaut sign ***	{
florin sign ****	|
1/4 sign	}
acute accent	~

* The 3/4 sign is displayed as a reverse question mark.
** The ij sign is displayed as lower y with umlaut.
*** The umlaut sign is displayed as double quotes.
**** The florin sign is displayed as lower f.

Note

> The 3/4, ij and florin signs are not in DEC STD 169 nor engraved on the Dutch keyboard and thus cannot be entered directly.

Because the Dutch set replaces some 7-bit characters as well as 8-bit characters, the algorithm is slightly different. If the Dutch set is in use then the 8-bit table is searched. If the character is found, it is replaced; else the 7-bit part is searched.

6.3.3 Support For TIA Control Code Generation Using Number Keys

The foreign keyboards have several symbols missing that are used on the domestic keyboard, in conjunction with <ctrl> to generate control codes. The TIA defined an alternate method for generating these control codes. The keyboard routine has been modified to use these keys (the number keys 2 through 8) in combination with the control key as specified by the TIA and shown below. This is implemented on all keyboards, including US, as per the TIA. Also, the shift/dependency to generate the control codes in the 'normal' way has been removed. For example, it is no longer necessary to type shift/tilde with control to generate the RS, only type the key containing tilde (as the shifted character).

Key	ASCII	Control Char
2	NULL	ØØH
3	ESC	1BH
4	FS	1CH
5	GS	1DH
6	RS	1EH
7	US	1FH
8	DEL	7FH

6.3.4 Auto-Repeat Control Codes Generated By the Keyboard

The keyboard algorithms are modified to allow control keys to auto-repeat. All keys will auto-repeat, if it is enabled, including those keys that generate control codes. The only time auto-repeating is not allowed is during compose sequences.

Level 1 is only available to a sophisticated user in console mode. This provides a unique 16-bit code for any key in combination with the CAPS LOCK, SHIFT, and CONTROL keys with the following exceptions:

Hold Screen - not available
SET-UP - not available
Compose character - not available

The keyboard may be broken into several functional key groups. Each has its own general characteristics.

Note

> The following refers to the keyboard layout shown in the figure at the end of this section. This layout differs from a VT102 keyboard.

6.3.5 Unseen Fixed Function Keys - Hold Screen, Set-up

These keys,always provide the same function regardless of console or terminal mode and are never provided to any level of output request.

6.3.5.1 HOLD SCREEN - POSITION G99 - Freezes the screen display or unfreezes it (toggle mode).

Any attempt to output any character is blocked until Hold Screen is "off." May cause an XOFF to be sent in terminal mode if receive buffer reaches high water mark and auto XON/XOFF is enabled. This is equal to the NO SCROLL key on a VT102.

Note

> If a program does not want to be 'hung up' by a display routine that is being blocked due to a 'hold screen' in effect, the user should test the state (available in SYSPAR) and put off "console out" until the user unblocks the display process by turning the Hold Screen off.

6.3.5.2 SET-UP - POSITION G01 - This causes entry to and exit from Set-Up mode. A system reset occurs when the CONTROL key is pressed in combination with the Set-Up key while in SET-UP mode.

6.3.5.3 COMPOSE CHARACTER - POSITION A99 - The compose algorithm, as described in DEC STD 169, will be implemented. However, the keyclick and bell specifications found in the standard may not be completely adhered to.

6.3.6 Fixed Function Keys - ESCAPE, LINE FEED, BACKSPACE, SHIFT, CONTROL, LOCK, TAB, RETURN, DELETE

These keys always provide the same function regardless of console or terminal mode and are provided to any level of output request.

6.3.6.1 ESCAPE - POSITION G11 - Escape at all times generates the ASCII escape character code, 1B (hex). It is not affected by SHIFT, CONTROL, or CAPS LOCK keys.

6.3.6.2 LINE FEED - POSITION G13 - Line feed at all times generates the ASCII line feed character code, OA (hex). It is not affected by SHIFT, CONTROL, or CAPS LOCK keys.

6.3.6.3 BACKSPACE - POSITION G12 - Backspace at all times generates the ASCII backspace character code, 08 (hex). It is not affected by SHIFT, CONTROL, or CAPS LOCK keys.

6.3.6.4 SHIFT (2 Keys) -POSITION B99, B11 - Shift at all levels causes a modification of the codes being generated by the alpha, numeric, and symbol keys. For alpha keys it sends the upper case code. The numeric and symbol keys send the code for the upper character shown on the keycap. If there are multiple upper and/or lower characters shown on the keycap, then the upper character/case is sent according to the correspondence/data processing mode in effect at the time.

6.3.6.5 CONTROL - POSITION C99 - Control at all levels causes a modification of the codes being generated by the alpha keys and some of the symbol keys and the space bar. Some of the symbol keys are only accessible with some foreign keyboards by using the data processing mode. The control codes remain associated with the keycap legend. If for example the alpha key for C is moved, a <Ctrl/C> still generates the ETX code.

6.3.6.6 LOCK - POSITION C00 - Lock at all levels causes selection of upper case for all alpha keys when "on." "On" state is indicated when the LED marked "Lock" is lit.

6.3.6.6.1 CAPS-SHIFT LOCK FUNCTION - The handling of the shift/caps lock mode is determined by an NVM parameter. NVM contains a binary valued variable that defines whether the 'lock' key functions as a shift or caps lock. If shift lock, then all keys that have a shifted representation will generate the code for the shifted character. If caps lock, then only alphabetic keys will generate their shifted character code. The 25th bit in the PARAMETERS field of Set-up will be used as the shift/caps lock selector. When selected, this field will display the appropriate text and allow the operator to modify the state. The lock mode choice of shift/caps lock is provided as part of the binary parameter field. The default is 'caps lock' mode. The display for this Set-up parameter will be the 25th location in the major field marked 'PARAM SET'. The minor field display will be :

LOCK	MODE
0 =	CAPS
1 =	SHIFT

6.3.6.7 TAB - POSITION D00 - Tab at all times generates the ASCII horizontal tab code, 09 (hex). It is not affected by SHIFT, CONTROL, or CAPS LOCK keys (except in SET-UP mode).

6.3.6.8 RETURN - POSITION C13 - Return at all times generates the ASCII carriage return code, ØD (hex). It is not affected by SHIFT,CONTROL, or CAPS LOCK keys.

Note

> If NEW-LINE mode is selected, this key will generate a CR LF combination.

6.3.6.9 DELETE - POSITION E13 - Delete at all times generates the ASCII delete character code, 7F (hex). It is not affected by SHIFT, CONTROL, or CAPS LOCK keys.

6.3.7 Alpha and Symbol Keys - POSITIONS EØØ-E12, DØ1-D12, CØ1-C12,

These are the Standard Keys affected by the SHIFT, BØØ-B1Ø, CONTROL, and CAPS LOCK keys, as well as the correspondence/data processing mode. They are mapped to match the keycap legends according to the language being used.

Note

> This requires use of the control key for those 'symbols' that generate control codes. Some foreign language keyboards may also require use of the 'data processing' keyboard mode (see SETUP) in order to select the desired 'symbol'.

6.3.8 Keypad Keys - POSITION E2Ø-E23, D2Ø-D23, C2Ø-C23, B2Ø-B22, A21-A23

These keys act the same at Level 2 as in a VT1Ø2 except for the Enter key which is not used for print functions. Either character codes or escape sequences are generated depending on keypad numeric/application mode and ANSI/VT52 mode. At Level 1 these keys are considered as function keys, and control/shift/caps lock flags are included in the 16-bit code.

6.3.9 Cursor Arrow Keys - POSITION C17, B16-B18

These keys act the same at Level 2 as in a VT1Ø2. Escape sequences are generated depending on cursor key normal/application mode and ANSI/VT52 mode. At Level 1 these keys are considered as function keys and the control/shift/caps lock flags are included in the 16-bit code.

6.3.1Ø Special Function Keys - PRINT SCREEN, BREAK

These keys have a defined function in terminal mode.

6.3.1Ø.1 Print Screen - POSITION GØØ - In terminal mode Print Screen causes the contents of the screen to be sent to the attached printer. Pressing <Ctrl/Print Screen> causes the terminal to toggle back and forth between auto print "on" and "off". This key is used in place of the ENTER key on a VT1Ø2 for printer functions.

6.3.1Ø.2 FLAG FOR PRINT SCREEN KEY - In order to provide MS-DOS and others a means of detecting the 'print screen' key without using the 16-bit keyboard interface, a flag bit in location SYSPAR has been defined. The ROM code will set this flag when a 'print screen' key is detected in the process of extracting key data from the key buffer. The O/S or application is responsible for clearing this bit after it is detected and when the O/S or application is first started. The location of SYSPAR is at address EFØØ:FFE and the flag bit is bit 7.

6.3.1Ø.3 PRINT SCREEN ESCAPE SEQUENCE - The escape sequence ESC [1 2 ~ has been assigned to the 'print screen' key for use in console mode allowing an application to see when this key has been depressed. It is the application's responsibility to do the printing, the firmware only indicates the key has been activated.

6.3.1Ø.4 BREAK KEY - POSITION GØ3 - In terminal mode the Break key acts just like in a VT1Ø2 including Shift-Break for a long break disconnect and Control-Break for a transmitted answerback message.

In console mode, this key is considered a function key and is available at Level 1 or as an escape string at level 2. The control/shift/caps lock flags are included in the 16-bit code. The pressing of this key also sets a flag bit in the SYSPAR location.

6.3.11 Function Keys - POSITION GØ2, GØ5-GØ9, G14-G16, G2Ø-G23, E16-E18,- D16-D18

These keys are only available in console mode at either Level 1 or Level 2 (as escape sequences). The control/shift/caps lock flags are included in the 16-bit level 1 code.

Note

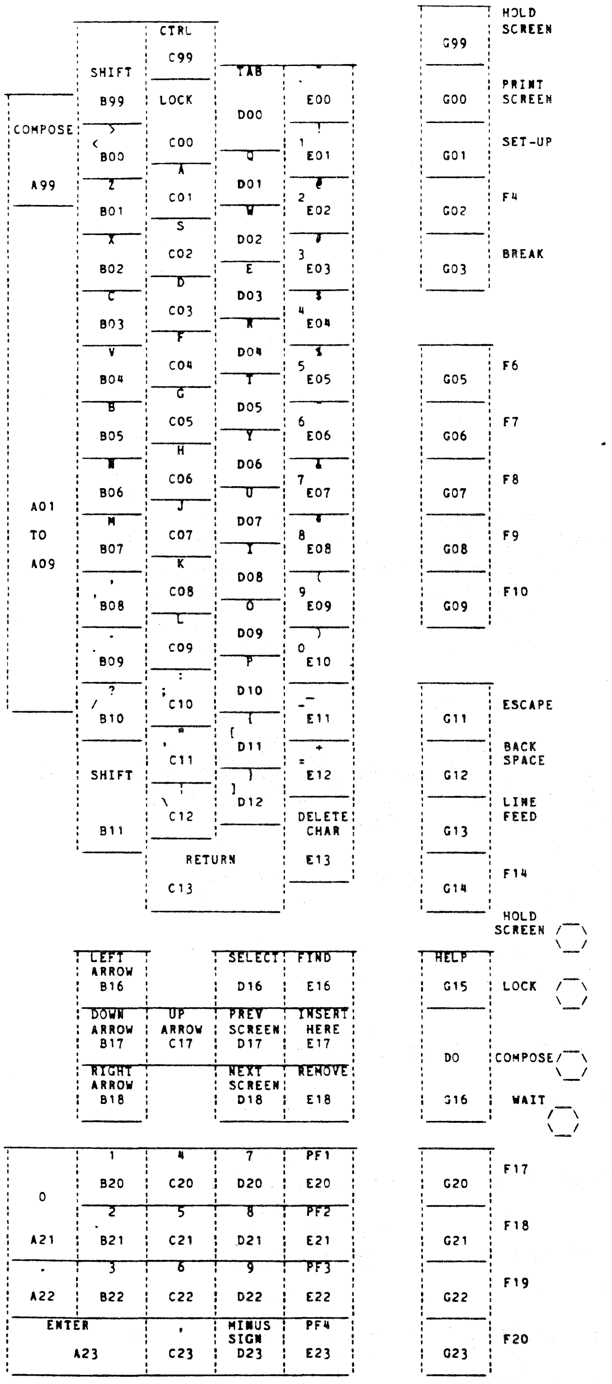
Some of these keys have functions within Set-Up mode:

Help - position G15
Next Screen - position D18
Previous Screen - position D17

ESCAPE SEQUENCES GENERATED BY FUNCTION KEYS (all final characters are 'tilde').

Name	Position	ESC Sequence
Print Screen	G00	ESC [1 2 ~
F4	G02	ESC [1 4 ~
F6	G05	ESC [1 7 ~
F7	G06	ESC [1 8 ~
F8	G07	ESC [1 9 ~
F9	G08	ESC [2 0 ~
F10	G09	ESC [2 1 ~
F14	G14	ESC [2 6 ~
HELP	G15	ESC [2 8 ~
DO	G16	ESC [2 9 ~
F17	G20	ESC [3 1 ~
F18	G21	ESC [3 2 ~
F19	G22	ESC [3 3 ~
F20	G23	ESC [3 4 ~
FIND	E16	ESC [1 ~
INSERT HERE	E17	ESC [2 ~
REMOVE	E18	ESC [3 ~
SELECT	D16	ESC [4 ~
PREV SCREEN	D17	ESC [5 ~
NEXT SCREEN	D18	ESC [6 ~

The keyboard layout of the PC100-B is shown below.


COMPOSE
A99
A01
TO
A09
SHIFT
B99
>
<
B00
Z
B01
X
B02
C
B03
V
B04
B
B05
N
B06
M
B07
,
,
B08
.
.
B09
?
/
B10
SHIFT
B11
CTRL
C99
LOCK
C00
A
C01
S
C02
D
C03
F
C04
G
C05
H
C06
J
C07
K
C08
L
C09
:
;
C10
"
'
C11
|
\
C12
RETURN
C13
TAB
D00
Q
D01
W
D02
E
D03
R
D04
T
D05
Y
D06
U
D07
I
D08
O
D09
P
D10
{
[
D11
}
]
D12
~
`
E00
!
1
E01
@
2
E02
#
3
E03
$
4
E04
%
5
E05
^
6
E06
&
7
E07
*
8
E08
(
9
E09
)
0
E10
_
-
E11
+
=
E12
DELETE
CHAR
E13
G99
HOLD
SCREEN
G00
PRINT
SCREEN
G01
SET-UP
G02
F4
G03
BREAK
G05
F6
G06
F7
G07
F8
G08
F9
G09
F10
G11
ESCAPE
G12
BACK
SPACE
G13
LINE
FEED
G14
F14
HOLD
SCREEN
LEFT
ARROW
B16
SELECT
D16
FIND
E16
HELP
G15
LOCK
DOWN
ARROW
B17
UP
ARROW
C17
PREV
SCREEN
D17
INSERT
HERE
E17
DO
COMPOSE
RIGHT
ARROW
B18
NEXT
SCREEN
D18
REMOVE
E18
G16
WAIT
1
B20
4
C20
7
D20
PF1
E20
G20
F17
0
A21
2
B21
5
C21
8
D21
PF2
E21
G21
F18
.
A22
3
B22
6
C22
9
D22
PF3
E22
G22
F19
ENTER
A23
,
C23
MINUS
SIGN
D23
PF4
E23
G23
F20

6.4 RECEIVED CHARACTER PROCESSING

6.4.1 Received Character Processing

VT1Ø2 emulation responds to different characters and control sequences so as to duplicate the response of the VT1Ø2. The full description is found in section 8 (Terminal Control Functions) of the VT1Ø2 Engineering Specification. A summary follows.

Reception of 8-bit control codes cause the same effect as if the 2-character, 7-bit escape-FE equivalent had been received.

Note

Reception of the 'RESET TO INITIAL STATE' escape sequence (ESC c) will also reset the 72Ø1 serial line controller effectively disabling its interrupt structure. This escape sequence should not be used from within an application without restoring the interrupt structure. Normally the operating system is in control of the interrupts of the 72Ø1 and an application will not know how to restructure the interrupts.

Note

Some operational capabilities are common to both terminal and console modes; others are not. These are noted below in the following format:

1. Those common to both are marked BOTH.

2. Differences are marked as either TERMINAL or CONSOLE.

6.4.2 ANSII Mode Control Functions

BOTH These functions give the terminal its intelligence. Examples in this group include:

1. Cursor functions (Movement, positioning, position reporting, etc.)

2. Mode setting and resetting
3. Line and character modes (Blink character, underline character, etc.)

4. Terminal editing (Insert and delete line and character, etc.)

5. Terminal identify, test, and status

6. Terminal characteristics (Key autorepeat,linefeed/newline mode, etc.)

CONSOLE MODE No local echo, modem, or printer functions in "console" mode.

6.4.3 Operational Variations

Some of the sequences listed in the previous section affect the terminal's operational mode. Examples include VT52 or ANSI mode, smooth or jump scroll and the like.

The escape sequence which normally causes a VT1Ø2 to execute self-tests are ignored and the status report always returns a "no problem" status.

6.4.4 Terminal Reports

BOTH The host computer tells the VT1Ø2 emulator to report its current cursor position, status, and device attributes.

TERMINAL MODE Printer status reports occur only in Terminal Mode.

6.4.5 Terminal Reset

BOTH The terminal also responds to a command from the computer which causes it to reset to its saved state. This is not the same as a keyboard entered system reset which returns to the opening menu. This recalls the NVM set-ups, clears the screen, and homes the cursor only.

6.5 VT1Ø2 CONTROL OF ATTACHED DEVICES

TERMINAL MODE VT1Ø2 emulation firmware has the same control over the printer and EIA modem as the VT1Ø2.

CONSOLE MODE VT1Ø2 emulation firmware does not have control over the printer and comm ports or EIA modem lines.

6.5.1 Modem Control

TERMINAL MODE Terminal Mode has full duplex capabilities:

1. data leads only

2. full modem

3. asymmetrical (requires special cable)

Terminal Mode does not have half duplex capabilities.

CONSOLE MODE In console mode the "application" must control the comm port hardware directly.

6.5.2 Printer Control

BOTH The baud rates available for the printer port are more limited than VT1Ø2 baud rates. The available baud rates are: 75, 15Ø, 3ØØ, 6ØØ, 12ØØ, 24ØØ, 48ØØ, 96ØØ.

TERMINAL MODE VT1Ø2 firmware contains code to control a serially connected printer in this mode only. The emulation supports the print screen and print cursor line commands, auto print mode, and printer control mode.

CONSOLE MODE In Console Mode the printer is only accessible to an "application" through direct control of the hardware.

6.6 OTHER DIFFERENCES TERMINAL VS. CONSOLE MODES

TERMINAL MODE This mode emulates a VT1Ø2, where keyboard characters go to the communication line, and communication line characters go to the display, plus all the printer support and Set-Up and modem control, etc.

Differences from a real VT1Ø2 are due to differences in the keyboard (keys in different places), different numbers of and different labels for LEDs and more extensive Set-Up information required (option ID's, volumes of bell and click, etc.). Also, no support of HDX modem protocols is provided.

CONSOLE MODE This mode acts like a VT1Ø2 console (without modem control, local echo, or printer port) to an application, where keyboard characters go to the application, application characters go to the display, and the printer is under control of the application; the communication port is under control of the application. However, the VT1Ø2 can still be put into Set-Up mode and have its characteristics changed like a real console.
Because there is no XON/XOFF between the "console VT1Ø2" and an application, both are running on the same CPU and/or in a single threaded environment. As a result the service routine hangs waiting for buffer space to become available (emptied by interrupt process) before returning to the calling routine.

There are two accesss methods:

1. The "application" accesses the VT1Ø2 "console" through the interface as though it were a serial line controller communicating over a high speed comm line.

2. A "sophisticated application" may access the video display RAM through indirect write of the display/attribute memory. This bypasses normal VT1Ø2 rcvd char processing. A "sophisticated user" also obtains 16-bit coded (level 1) key data to bypass escape sequence encoding/decoding needs.

Note

Level 1 and level 2 'character-available' status are not interchangeable, a level 2 character-available does not imply a level 1 character available.

Within the VT102 emulation there are routines to pass status and data from/to the interface layer.

Within the VT102 emulation, interrupts must not be disabled any longer than 450 microsec. This requires cautions on re-entrancy of routines that are shared and potential problems of not completing an interrupt handler that re-enables interrupts and then does not complete before it is called again.

DATA FLOW DIAGRAMS FOR THE DIFFERENT MODES OF OPERATION

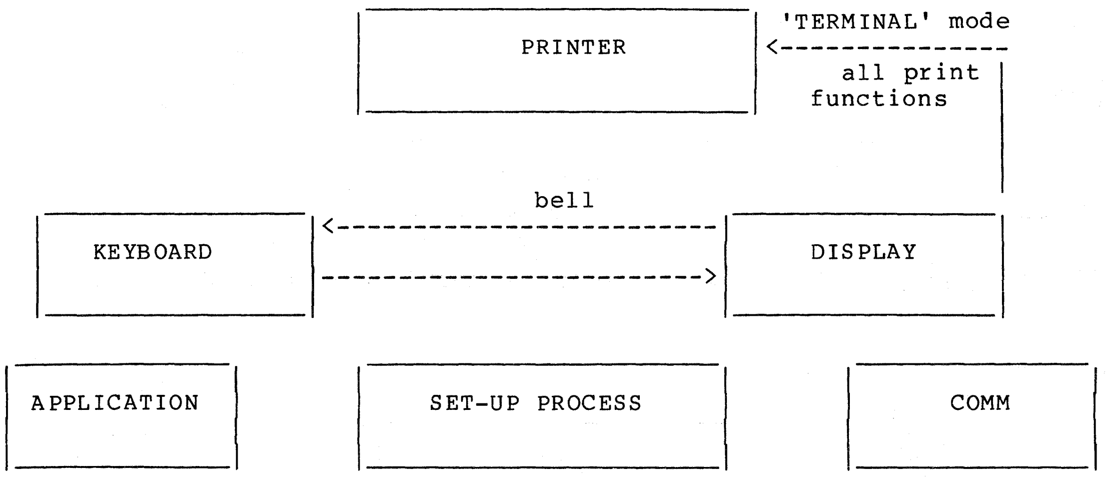


Figure 10. 'Terminal' or 'Console' Mode 'Off-line'

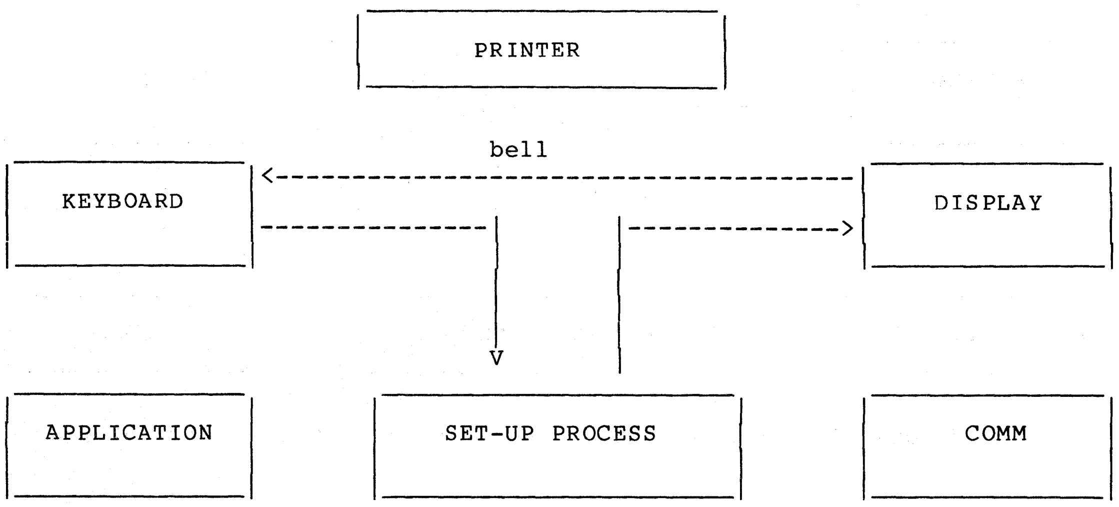


Figure 11. 'Terminal' or 'Console' Mode 'Set-Up'

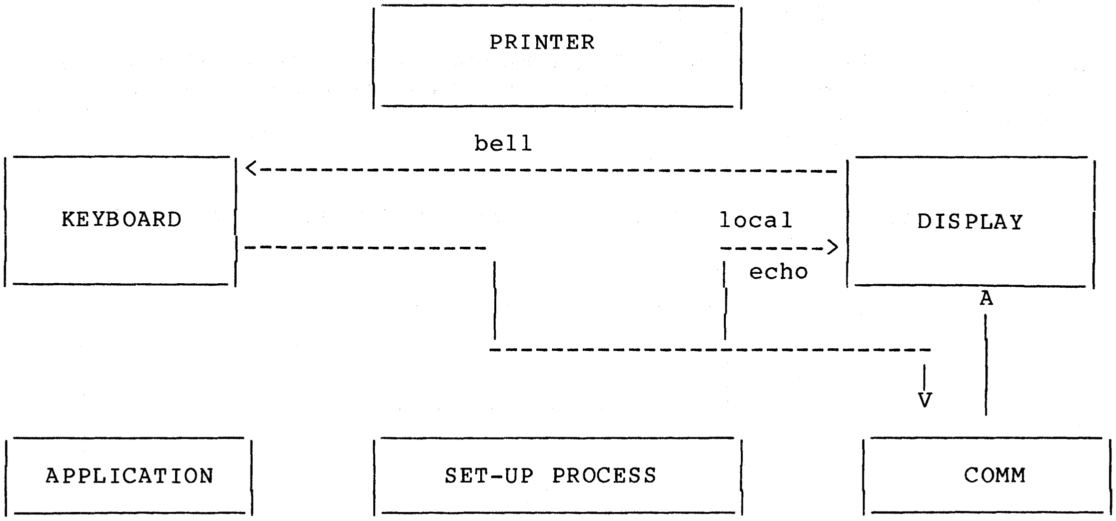


Figure 12. 'Terminal' Mode 'On Line' No Printer Functions

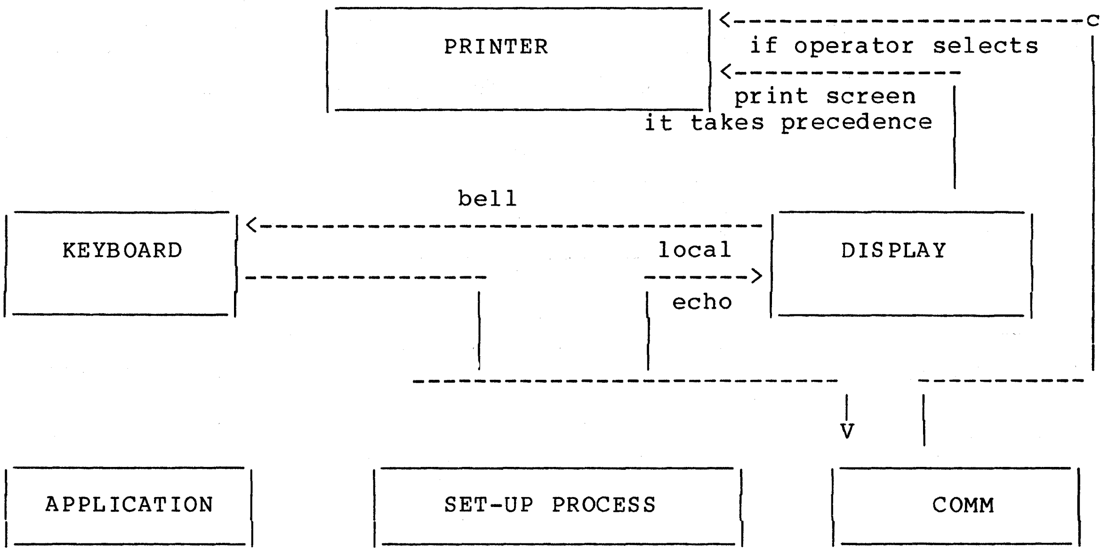


Figure 13. 'Terminal' Mode 'On Line' In 'Printer Controller'

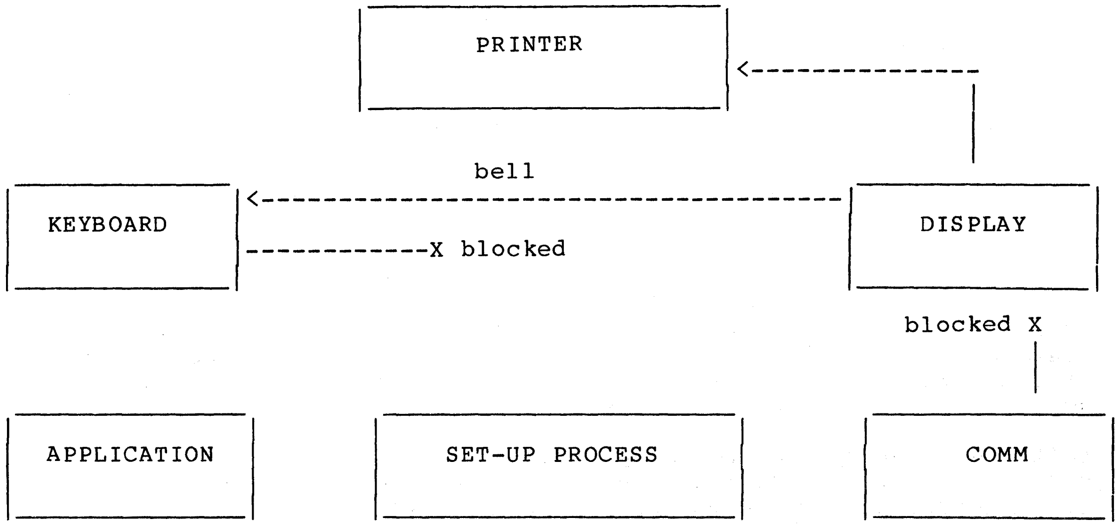


Figure 14. 'Terminal' Mode 'On Line' Printing From Screen

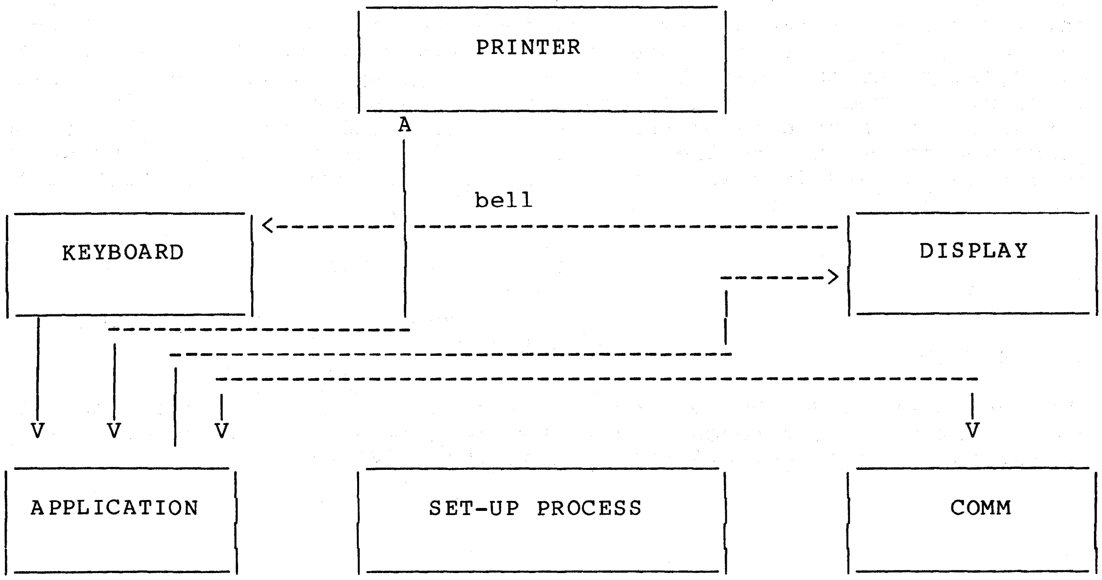


Figure 15. 'Console' Mode

7 EXTERNAL INTERACTIONS

7.1 OPERATING SYSTEM

VT1Ø2 emulation allows access to its subroutines by the operating system. It does not require operating system services and runs without the use of disks.

7.2 STACK OVERHEAD

Hardware interrupts and the attendant interrupt handlers impose a stack overhead on any program running at the time of the interrupt. The interrupt handlers swap to their own stack as soon as possible to minimize this overhead. However, it still requires three words due to the interrupt process itself (CS, IP, FLAGS) and the firmware pushes three 3 more words before it swaps to its own stack.

Software interrupts do not swap stacks, and the user must provide sufficient stack space (plus hardware interrupt overhead). The amount of stack required is 25 words (plus hardware, total = 31).

7.3 STACK USE BY APPLICATIONS

The firmware hardware interrupt handlers swap stacks, if required, so they always run on their own stack. Some of the handlers re-enable interrupts after this swap, but while still within the interrupt handler, so other interrupts are not held up. At the conclusion of the original firmware interrupt handler the stacks are swapped back to the original. There is a caution for 'application' interrupt handlers that expect to use the stack or registers to pass data between their interrupt handler and the main-line program. If the application interrupt occurs while still within the firmware interrupt handler then current stack and registers will not be preserved for main-line use. Application interrupt handlers must never use registers to pass data and must guarantee their own private stack if they are going to pass data on a stack.

8 VT1Ø2

8.1 SET-UP

The non-volatile Set-Up parameters are different from those in the VT1Ø2 due to hardware differences and system requirements. The non-volatile Set-Up parameters that are the same as the VT1Ø2 are:

Minor Field	Major Field	Default
on line/local mode	header	on line
8Ø/132 column mode	parameter	8Ø
tab stop bit map	TABs	every 8 positions
scroll - SMOOTH/JUMP	parameter	smooth
auto repeat - ON/OFF	parameter	on
screen background - LIGHT/DARK	parameter	dark
cursor - UNDERLINE/BLOCK	parameter	block
margin bell - ON/OFF	parameter	off
keyclick - ON/OFF	parameter	on
ANSI/VT52	parameter	ANSI
* auto xon/xoff - ON/OFF	parameter	on
US/UK char set	parameter	US
auto wrap - ON/OFF	parameter	off
line feed/new line	parameter	LF
* local echo - ON/OFF	parameter	off
* print termination char - none/FF	parameter	FF
* print extent - ALL SCREEN/SCROLLING REGION	parameter	all
comm stop bits - 1/2	parameter	1
* rcv parity check - ON/OFF	parameter	on
* break enabled - ON/OFF	parameter	on
* disconnect char enable - ON/OFF	parameter	off
* disconnect delay - 6Ø MS/2 SEC	parameter	2 sec
* auto answerback enabled - ON/OFF	parameter	off
5Ø/6Ø hz	parameter	6Ø
comm data/parity bits	modem	7S
comm xmit rate	modem	96ØØ
comm rcv rate	modem	96ØØ
* comm modem line discipline	modem	FDXA
* disconnect char	modem	none
answerback message - UP TO 2Ø CHAR (PLUS 2 DELIMITERS)	answerback	none
printer data/parity bits	printer	8N
printer xmit/rcv rate	printer	48ØØ

Insertion/replacement mode is always saved as replacement mode.

VT1Ø2 non-volatile parameters not included in the emulator are:

screen brightness - NO SOFTWARE CONTROL
WPS keyboard - NO KEYBOARD VARIATION FOR THIS
HDX protocol related parameters

New non-volatile parameters to be added in emulator are:

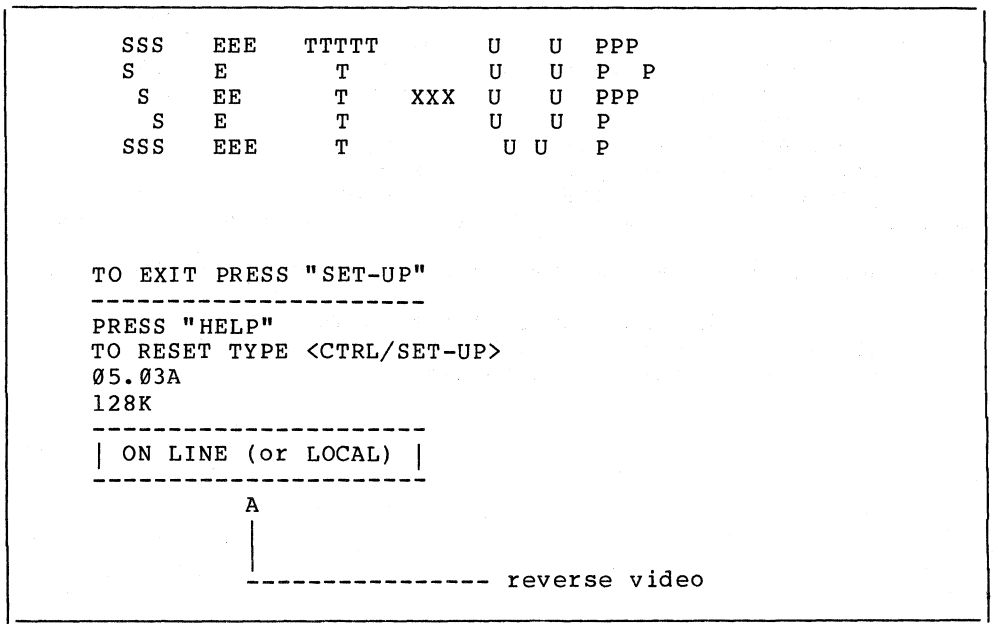
Minor Field	Major Field	Default
bell volume	misc	7(max)
keyclick volume	misc	7(max)
memory size installed	fixed header	none
automatic screen blanking after elapsed time - ON/OFF	parameter	on
scroll rate for smooth scroll - 3,6,12 LINES/SEC AT 6ØHZ (slower at 5ØHZ)	misc	6
keyboard key assignments (for foreign language keyboard support) CORRESPONDENCE/DATA PROCESSING	parameter	correspondence
lock mode - CAPS/SHIFT	parameter	caps
keyboard selected - YES/NO	parameter	no
character code mode - DEC-8/7-BIT	parameter	DEC-8
auto-boot from drive - ?,A,B,C,D,W	auto-boot	? (none)

Legend

* only apply in terminal mode

8.1.1 Set-Up Display and Operation

The Set-Up displays consist of two major screen areas. One is fixed and the other variable as a function of what part of Set-Up is used. The fixed part of the display, called the header, consists of the words shown below.



```
SSS   EEE   TTTTT        U    U   PPP
S     E       T          U    U   P  P
 S    EE      T    XXX   U    U   PPP
  S   E       T          U    U   P
SSS   EEE     T           U U     P



TO EXIT PRESS "SET-UP"
----------------------
PRESS "HELP"
TO RESET TYPE <CTRL/SET-UP>
Ø5.Ø3A
128K
----------------------
| ON LINE (or LOCAL) |
----------------------
          A
          |
          |
          ---------------- reverse video
```

Beneath the header is the variable area which is one of eight different displays called "major fields." On entering the Set-Up mode, the tab settings major field is automatically displayed. There are two types of major fields. One type is accessed by stepping through them sequentially by means of the Next Screen and Previous Screen keys. The other type is entered directly at any time (almost) by means of the Help key (help field) and shifted A (answerback field). Within some of the major fields are variable numbers of minor fields, one for each parameter selectable within that major field. Minor fields are stepped through by means of the left and right arrow keys.

Except for 'HELP' and 'ANSWERBACK' fields, the following keys always have the same effect:

SET-UP	exit set-up mode
CTRL-SET-UP	reset system
HELP	enter/exit help field
NEXT SCREEN	select next major field
PREV SCREEN	select preceeding major field
UPPER/LOWER CASE L	toggle line/local state
UP-ARROW	select next higher value
DOWN-ARROW	select next lower value
SHIFT-S	save current set-ups in NVM
SHIFT-D	set current set-ups to defaults (does not save)
SHIFT-R	recall saved set-ups from NVM
SHIFT-A	enter answerback field

8.1.1.1 HELP Field - The help field is displayed by pressing the Help key. It is entered at any time except when already in the answerback field. It presents a short list of how to access the major sequential fields, minor fields, and how to change values. The only key honored while in "Help" is the Help key which causes an exit to the field displayed before entry to "Help."

8.1.1.2 ANSWERBACK Field - The answerback field is displayed at any time by pressing and holding the SHIFT key and then pressing the A key. The major field heading ANSWERBACK is displayed and beneath it "A =". The first character typed (and displayed) is a delimiter. Up to 20 characters can be entered following the delimiter. The string automatically terminates after the 20 characters or before that when a second delimiter entry is made. Any character can be entered in the answerback string, including NULL, DELETE, or CTRL characters.

Normally non-printing control characters show as their related ASCII characters in reverse video. For example, ETX (control C) shows up as a reverse video C. Answerback is exited by typing the delimiter character a second time or by reaching the 20 character entry limit. String entry errors can only be fixed by exiting answerback and re-entering to make a new string. The string is eliminated by making the first two characters the same (2 delimiters with nothing between).

8.1.1.3 TAB SETTINGS Field - The tab setting field is displayed on entry to Set-Up. It consists of a row of numbers running repetitively from 1 through 0 with alternate groups of 10 shown in reverse video. The line above this row of numbers contains a variable number of the letter "T."

The location of the T symbol signifies that a tab stop is set at that point. There are 8Ø or 132 possible tab stops depending on the screen width setting in effect. A cursor shows the current active position. This cursor moves by means of the left and right arrow keys, Return (to start of line), Tab (to next "T" location), and the space bar(acts the same as right arrow). The "T" key (or up or down arrow keys) reverses the tab setting at the active position. In addition Ctrl Tab clears all tab stops and SHIFT TAB sets the default tab stops at every eighth position (beginning with the ninth position a T is displayed).

8.1.1.4 **PARAMETER MAJOR Field** - The parameter major field displays all the Set-Up parameters that have only two possible values (usually on/off). The parameters are shown as a row of 1's and Ø's with alternate groups of 4 shown in reverse video. The current active parameter is shown by a cursor. This cursor moves left and right by means of the left and right arrow keys, Return (to start of line),Tab, and space bar(acts the same as right arrow). As the cursor moves to a new position a field is displayed below the line that details the name of the parameter and what the Ø and 1 values mean. The Ø and 1 values are changed by means of the up and down arrow keys. Some of these parameters have an immediate affect if changed. Others do not take affect until exiting from Set-Up mode. Changing to either 8Ø or 132 columns destroys the user's text display (normally it is preserved during Set-Up and restored on exit).

8.1.1.5 **MODEM Field** - The modem field displays the parameters for the modem port (communications port). Some of these are only applicable in "terminal" mode. Each parameter has a range of possible values, and the field is shown as a list on the display as shown below. The active parameter is shown by a reverse video field.

```
 ------------------------------------------------------------------
|                                                                  |
|                                                                  |
|                                                                  |
|     MODEM                                                        |
|     -----                                                        |
|       8N       DATA/PARITY                                       |
|     --------                                                     |
|  |-->| 96ØØ |  XMIT BAUD                                         |
|  |  --------                                                     |
|  |    96ØØ    RCV BAUD                                           |
|  |                                                               |
|  |    FDXA    PROTOCOL                                           |
|  |                                                               |
|  ---------------- reverse video                                  |
|                                                                  |
|__________________________________________________________________|
```

The value of the active parameter changes by means of the up and down arrow keys which cause the parameter to step through its allowable values. The active parameter selection changes by means of the left and right arrow keys. Which moves the active field up and down the list of parameters.

8.1.1.6 PRINTER Field - The printer field displays the parameters for the printer port. The active parameter selection and value changes are the same as for the modem.

8.1.1.7 MISCELLANEOUS Field - The miscellaneous field is used to select the bell and key click volumes and the smooth scroll rate. The active parameter selection and value changes are the same as for the modem.

8.1.1.8 Auto-boot Device Selection - The device will be stored in NVM as 1 of 6 different values, nothing selected (shown as '?'), A, B, C, D (for the 4 possible floppy drives), or W (for the winchester disk). The default is 'nothing selected' The major field heading will be 'AUTO-BOOT '. The minor field choices will be '?,A,B,C,D,W' which are selected by the up/down arrow keys.

Example: AUTO-BOOT ? = unit (when no device has been chosen)
8.1.1.9 Storing the Set-ups - The current contents of all the Set-Up fields, including any answerback message, are transfered to non-volatile storage by pressing the SHIFT and "S" keys simultaneously (not while in HELP or ANSWERBACK).

Note

> Insertion/replacement mode is always set to replacement mode before saving is done.

8.1.1.1Ø Using Default Set-ups - All the Set-Up fields are set to their respective default conditions by pressing the SHIFT and "D" keys simultaneously (not while in HELP or ANSWERBACK). Defaults are listed in section 5.1.

Note

> This does not enter the defaults into non-volatile storage (must use "store"), but it causes the user's text to be erased from the screen and causes a line disconnect in "terminal" mode.

8.1.1.11 Recalling Stored Parameters - The parameters stored in non-volatile storage are recalled for use by pressing the SHIFT and "R" keys simultaneously. This also causes the user's text to be erased from the screen and causes a line disconnect in "terminal" mode. Stored parameters are also automatically recalled at power-up, system reset, and reception of the RIS escape sequence (ESC c). RIS will also do a channel reset of the 72Ø1 printer and communications ports destroying the mode and interrupt structure in use at the time.

8.2 RESET TECHNIQUE

Within Set-Up mode a "reset" key combination (CONTROL SET-UP) causes a jump to the start of the self-test at a point which allows it to be distinguished from a power-up.

9 INTERFACE LAYER

Functions provided by the firmware for use by operating systems and sophisticated applications.

9.1 SOFTWARE INTERRUPT TYPE 4Ø (DECIMAL)

Function code is passed in DI. It is organized as 16 even number values for ease in using as table offsets to dispatch to service routines.

FUNCTION CODES

HEX	
Ø	CONSOLE OUT
2	LEVEL 2 CONSOLE IN
4	LEVEL 2 CONSOLE IN STATUS
6	LEVEL 1 (16-BIT) CONSOLE IN
8	DISABLE CURSOR
A	ENABLE CURSOR
C	INITIALIZE INTERRUPT VECTORS
E	RETURN CLOCK RATE
1Ø	SET LEDs ON KEYBOARD
12	CLEAR LEDs ON KEYBOARD
14	SEND DATA TO SCREEN
16	INIT 72Ø1 TO NVM PARAMETERS
18	RAW KEYBOARD DATA
1A	RETURN ROM VERSION NUMBER
1C	CHANGE INTERRUPT VECTOR MAP
1E	RING KEYBOARD BELL

Note

> Only registers CS: , SS: , and DS: are preserved. All other general purpose and segment registers are not preserved.

9.1.1 Console Out

ENTRY DI = Ø
AL = character in ASCII (includes 8-bit multi-national characters)

9.1.2 Level 2 Console In

```
ENTRY DI = 2

EXIT  AL = Level 2 keyboard character if available
      CL = returned status

          Ø = no character available
         FF = character is in AL
```

9.1.3 Level 2 Console In Status

```
ENTRY DI = 4

 EXIT CL = returned status
     Ø = no character available
    FF = character is available
```

Note

Cannot be used to detect staus of level 1 character available.

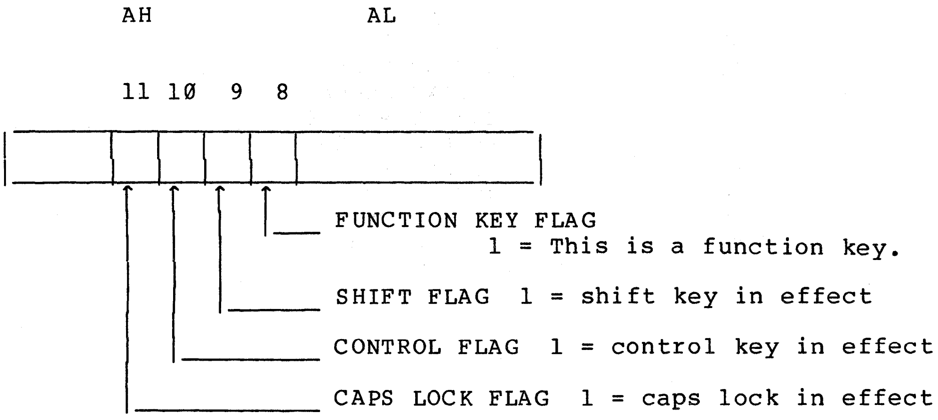
9.1.4 Level 1 Console In

```
ENTRY DI = 6

 EXIT AX = 16-bit level 1 character
      CL = returned status

             Ø = no character available
             1 = no character available due to Level 2 sequence not
             completed (previous key caused a string of level 2
             characters to be generated. Level 2 buffer has not been
             emptied of this string yet.)
    FF = character is in AX
```

RULES FOR LEVEL 1 CHARACTERS


AH
AL
11
1Ø
9
8
FUNCTION KEY FLAG
1 = This is a function key.
SHIFT FLAG 1 = shift key in effect
CONTROL FLAG 1 = control key in effect
CAPS LOCK FLAG 1 = caps lock in effect

For non-function keys, the AL = character (including 8-bit multi-national characters). The effect of shift/control/caps lock is already taken into account.

For function keys, the AH = flag data and AL = function key code (ASCII) as follows:

KEY CODE HEX	KEY
Ø	HELP
1	DO
2	not used
3	PRINT SCREEN
5	F4
7	F6
9	F7
B	F8
D	F9
F	F1Ø
11	F14
13	F17
15	F18
17	F19
19	F2Ø
1B	FIND
1D	INSERT HERE
1F	REMOVE
21	SELECT
23	PREV SCREEN
25	NEXT SCREEN
27	UP-ARROW
29	DOWN-ARROW
2B	RIGHT-ARROW
2D	LEFT-ARROW
2F	KEYPAD Ø
32	KEYPAD 1
35	KEYPAD 2
38	KEYPAD 3
3B	KEYPAD 4
3E	KEYPAD 5
41	KEYPAD 6
44	KEYPAD 7
47	KEYPAD 8
4A	KEYPAD 9
4D	KEYPAD DASH
5Ø	KEYPAD COMMA
53	KEYPAD PERIOD
56	KEYPAD ENTER
59	KEYPAD PF1
5C	KEYPAD PF2
5F	KEYPAD PF3
62	KEYPAD PF4
65	BREAK

9.1.5 Disable Cursor

ENTRY DI = 8

EXIT There is no cursor affect on attributes at cursor position. The cursor does not show on the screen.

Note

> The disable and enable cursor functions are only to be used immediately preceding and following a function 14 that may attempt to set the attributes at the current cursor position. They can be used to make the cursor 'invisible' while it is moved around by other escape sequences or control characters. These functions must be used in pairs, first disable then enable.

9.1.6 Enable Cursor

ENTRY DI = A

EXIT Cursor does affect attributes at cursor position. The cursor shows on the screen.

9.1.7 Initialize Interrupt Vectors

ENTRY DI = C

EXIT The following interrupt types are modified for use by the firmware:

Type 2	**NMI for RAM option parity error**
32.	vertical frequency refresh
34.	graphics controller option
35.	DMA controller of extended comms option
37.	72Ø1 of extended comms option
38.	keyboard 8251
44.	time tick

In addition, the extended comms option and graphics option are reset to the disabled state.

The extended comms option is reset by writing anything to 8Ø88 port 27 hex.

The graphics option is reset by toggling bit Ø of 8Ø88 port 5Ø hex from high to low to high.

9.1.8 Return Clock Rate

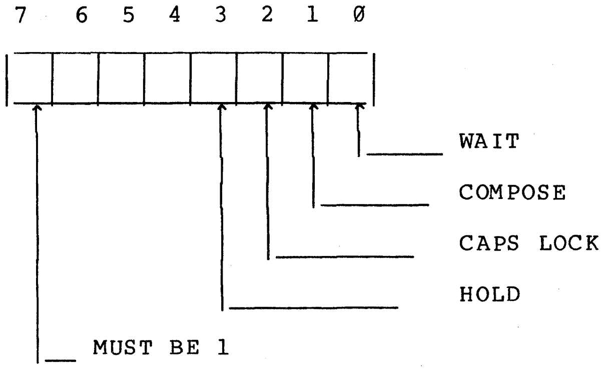
```
ENTRY DI = E

EXIT AL = clock rate
  bit 0 = 0 60 hz
        = 1 50 hz
```

9.1.9 Set Keyboard LEDs

Set and clear LEDs are only for the indicators. They do not cause any action that may be implied by the label of the LED affected. Firmware normally maintains all LEDs in the proper state.

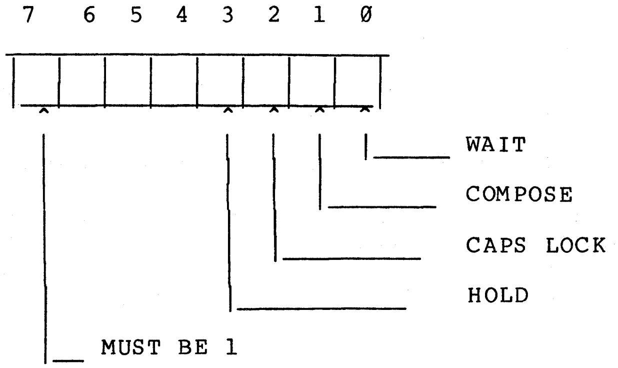
```
ENTRY DI = 10
      AL = bit pattern of LEDs to turn on
       bit set to 1 = LED on
```



```
EXIT    LEDs as requested
```

9.1.10 Clear Keyboard LEDs

```
ENTRY   DI = 12
        AL = bit pattern of LEDs to turn off
                      bit set to 1 = LED off
```



```
EXIT LEDs as requested
```

9.1.11 Send Data To Screen

```
ENTRY DI = 14

        AX = TRANSFER TYPE
          Ø = CHARACTERS AND ATTRIBUTES
          1 = ATTRIBUTES ONLY
          2 = CHARACTERS ONLY
          3 - FFFF = UNDEFINED

       BX = START LOCATION IN DISPLAY
       BL = LINE NUMBER (1-24)
       BH = COLUMN NUMBER (1-132)
```

Note

Maximum column number is a function of screen width (8Ø,132) and line width (single,double)

CX = NUMBER OF CHARACTERS/ATTRIB TO TRANSFER, IN BYTES

Note

User is responsible for limiting size of transfer so end-of-line is not exceeded.

DX = OFFEST TO START OF ATTRIBUTES RELATIVE TO USER'S DS:

SI = OFFSET TO START OF CHARACTERS RELATIVE TO USER'S DS:

BP = CHARACTER/ATTRIB SEGMENT COPY OF USER'S DS: USED FOR OFFSETS PASSED IN DX AND SI.

Note

Characters and attributes must be relative to same value of DS; this will not modify line attributes, only character attributes.

9.1.12 Init 72Ø1 To NVM Parameters

NVM refers to current contents of the shadow RAM which are displayed in SET-UP. They are not necessarily the same as the currently saved SET-UP parameters.

```
ENTRY DI = 16
      DL = Ø
    FUNCTION
```

1. Does a channel reset on both channels A and B
2. Sets baud rates for modem and printer ports according to NVM
3. Loads 1Ø (hex) into write register 2A, and Ø into write register 2B
4. Loads write registers 4A,B with X16 clock, parity/stop bits according to NVM for the port

5. Loads write registers 3A,B with number of receive data bits according to NVM for the port and enables receive
6. Loads write registers 5A,B with number of transmit data bits according to NVM for the port and enable transmit

Note

When data/parity is 7M or 7S, the 7201 is actually set for 8 data bits, no parity.

9.1.13 Raw Keyboard Data

ENTRY DI = 18

This function is provided for diagnostics to test the keyboard and is only intended for that purpose. It is documented here for completeness.

Note

The SHIFT, CAPS LOCK, and CONTROL keys can only be read in conjunction with another key by looking at the flag bits in AH. The SET-UP key will not be detectable by a program, but an operator will see entry into set-up mode on the display. The HOLD SCREEN key will not be detectable by a program, but an operator will see the 'HOLD SCREEN' LED on the keyboard light up.

```
EXIT CL = 0 no key available
        = 1 key available
```

AL = key location matrix code as defined in LK201 keyboard specification

AH = flag bits as for function 6, level 1 console

9.1.14 Return ROM Version Number

This function is provided for those programs that are hardware/ROM version dependant. It returns an ASCII text string of the form MM.mm<NULL>L<NULL>, where MM is the major version variation and mm is the minor version variation, L is the language variation and <NULL> is the ASCII null character, 00. This is the same format as the 'hardware part' of the extended console function that returns software and hardware versions. It may be used by the BIOS to provide the hardware version part of that function. By creating an 'empty' buffer and requesting this function, software can tell whether or not this new function (and its companions) is supported when it does or does not 'fill in' the buffer. Internally the version number is kept in ROM as two separate strings that are combined to produce the final result. ROM 0 contains an ASCII text string of the form MM.mm and ROM 1 contains a string of BCD values that are added to the ROM 0 string on a per-character basis to provide the final result. By choosing the proper characters and values the overall version number can be set to any desired result.

EXAMPLE:

```
    'Ø 4 . - 7'      ROM Ø ASCII string
     Ø 1 Ø 3 -1      ROM 1 BCD string
     ----------
    'Ø 5 . Ø 6'      resultant version number
```

The language variation comes from the keyboard selection stored in NVM. The language ID character is the same as used in the 1ØØA ROMs. It is obtained by using the keyboard ID, from NVM, code as an index into a list of characters such that the result is as shown in the following table. If a keyboard has not been assigned, the default ID character is lower case x. Therefore the version number for the base level 1 ROM set, which does not have keyboard selection implemented as yet, is Ø4.Ø1x.

Keyboard	**ID character**
AMERICAN	A
CANADIAN/FRENCH	C
UK	E
FINNISH	F
SWEDISH	M
NORWEGIAN	N
DANISH	D
SPANISH	S
GERMAN	G
SWISS/FRENCH	K
SWISS/GERMAN	L
DUTCH	H
BELGIAN/FLEMISH	B
FRENCH	P
ITALIAN	I

Function Code	**Description**
1A return	ROM (hardware) version number
ENTRY	DI = 1A DX = address offset of 8-byte buffer for returned version BP = address segment of buffer (i.e. buffer is at BP:DX)
EXIT	Buffer filled in with version number in form of ASCII text string as follows:

DB	tens digit of ROM/hardware version number major part
DB	units digit of ROM/hardware version number major part
DB	period character as separator between parts
DB	tens digit of ROM/hardware version number minor part
DB	units digit of ROM/hardware version number minor part
DB	NULL separator
DB	character that identifies language variation
DB	NULL terminator

9.1.15 Change Interrupt Vector Map

The purpose of this function is to re-map the interrupt vectors to another block of addresses primarily to resolve a conflict with MS-DOS on vector usage. The format is such that it is expandable for the future if any future hardware allows for selectable vector space. In this implementation only 1 other vector block is supported. Also in this implementation, only 16 consecutive vectors are affected. The format supports future expansion.

Function Code Description

```
     1C      Remap interrupt vectors

     ENTRY DI = 1C
           AL = vector number where transfer will begin from if AL = Ø,
                then begin from default vector number
           AH = vector number where transfer will go to if AH = Ø, go to
                default vector number
        if AX = Ø, then only initialize default vector space
```

For current Rainbow 1ØØ, the default vector number is 2Ø (hex), 32 (decimal)

```
          CX = number of vectors to be moved
       If CX = Ø on entry, exit with no changes made
```

For current Rainbow 1ØØ, the default number of vectors is 16 (decimal)
For the PC1ØØ-B board, where this will first be implemented, the only supported capabilities will be:

```
     Move 16 vectors from default (2Ø HEX) to AØ(HEX)
          AL = Ø (or 2Ø HEX)
          AH = AØ (HEX)
          CX = 16 (decimal)

     Move 16 vectors from AØ(HEX) to default (2Ø HEX)
          AL = AØ (HEX)
          AH = Ø (or 2Ø HEX)
          CX = 16 (decimal)
```

Note

> This will not relocate the time tick vector at 1ØØ (decimal), since it is unique to CP/M-86/8Ø BIOS and the firmware knows nothing about it, but it will relocate the time tick at 44 (decimal).

```
     EXIT CX = Ø to show successful remapping (this will serve as a quick
            check of support that can be used by software)
```

Requested block of interrupt vectors copied to new location. Old block of vectors remains unchanged. Firmware interrupt related routines now use new vector locations for hardware and software interrupts.

9.1.16 Sound The Keyboard Bell

The ringing of the keyboard bell required the user to use the normal console out data path. This function is to provide the bell function within the framework of the high performance data path.

```
Function Code     Description

     1E     Ring the keyboard bell

     ENTRY DI = 1E
```

9.1.17 Get/Set The DEC-8/7-bit Character Code Usage Parameter In NVM

The NVM parameter that determines usage of DEC-8 or national replacement 7-bit character codes is able to be read or set by an external user.

```
Function Code     Description

     2ØH    set/get DEC-8/7-bit character code state

     ENTRY DI = 2ØH

           AH = 1, for get function
                Ø, for set function

            for set function

          AL = Ø, for DEC-8
               1, for 7-bit national replacement character codes

     EXIT
          for get function

          AL = Ø, for DEC-8
               1, for 7-bit national replacement character codes
```

1Ø IMAGE OF Z8Ø RAM SPACE TO BE LOADED

Any routines that must be loaded into the Z8Ø space for it to run initially must be put there by the 8Ø88. The ROM must contain this code because it cannot be obtained from the floppy disk until after the floppy handler is loaded into the Z8Ø and the interface to access the loader is also in place. This initial code must know how to take care of the 'flipped' Z8Ø RAM addresses and relocate routines in the proper locations in RAM.

11 BOOT LOADER TO READ TRACK Ø, SECTOR 1 OF FLOPPY

The purpose of the boot loader is to get into memory a minimal routine which initiates the loading of the overall operating system. The boot loader accesses any available drive as selected by the operator from the opening menu. If an error occurs, an error message is displayed and the opening menu is re-displayed.

The boot loader loads 512 bytes from the specified drive, track Ø, sector 1 into the shared RAM beginning at address 1ØØØ (hex) and jumps to it. If the loaded routine returns an error, the opening menu is re-displayed. The selected drive is available by examining the Z8Ø drive select port (4Ø hex) which did the selection.

The loaded data must be Z8Ø instructions.

11.1 BOOT PROCESS

The boot process consists of the following steps:

1. Operator selects the drive to boot from by means of the menu.
2. The 8Ø88 passes control to the Z8Ø routine. The routine attempts to read track Ø, sector 1 of the selected drive into address 1ØØØ (hex).
3. While the Z8Ø is attempting to read, the 8Ø88 counts time and monitors a semaphore location at Ø:FFF (initially set to Ø).
4. If the semaphore does not change from Ø within approximately 1Ø seconds, there is some sort of major problem. The Z8Ø is stopped and the Z8Ø response failure message is displayed along with the opening menu. The operator may then make a selection from the menu.
5. If the Z8Ø routine detects a 'drive not ready' condition, it returns a value of 6 in the semaphore location. The 8Ø88 displays the message 'Failure, drive not ready, consult your user guide' and redisplays the opening menu. A 'drive not ready' is caused when either a drive door is opened, no disk is in the drive, or when there is no drive in the system.
6. If the Z8Ø detects an error reading track Ø, sector 1 of the selected drive due to a seek or CRC error, it retries up to two more times. If all three attempts fail, the Z8Ø returns the value 2 in the semaphore location. The 8Ø88 displays the message 'Failure, boot loader, consult your user guide' and redisplays the opening menu.
7. If the Z8Ø successfully reads track Ø, sector 1 of the selected drive, it checks the contents of address 1ØØØ (hex). If this is not the Z8Ø code for disable interrupts (F3), the Z8Ø returns the value 4 in the semaphore location. The 8Ø88 displays the message 'Failure, non-system disk, consult your user guide' and redisplays the opening menu.

Note

> This requires the Z8Ø code resident in track Ø, sector 1 to begin with a 'DI' instruction.

11.1.1 If the Z8Ø successfully reads track Ø, sector 1 of the selected drive into address 1ØØØ (hex), and the first byte is the 'DI' instruction, control transfers to this secondary boot by jumping to address 1ØØØ (hex).

Note

The secondary boot is Z8Ø code.

11.1.2 The secondary boot is responsible for loading the remainder of the system.

11.1.3 If the loading process fails, control returns to the 8Ø88 by placing the value 8 in the semaphore location. The 8Ø88 displays the message "Failure, system loader, consult your user guide', halts the Z8Ø, reloads the boot reading routine, and redisplays the opening menu.

11.1.4 When the loading process completes successfully, control passes to the 8Ø88 by placing the value A (hex) in the semaphore location. The 8Ø88 passes control to the loaded system by doing an indirect, intersegment, far jump via Ø:FFB.

Note

The contents of four bytes, starting at Ø:FFB, must be pre-loaded with the code segment and an offset of the 8Ø88 (system) start address:

FFB, FFC contain IP

FFD, FFE contain CS

Note

The selected drive is determined by reading the Z8Ø port 4Ø (hex), masking to read bits Ø and 1, and the drive selected is:

Bit Ø	Bit 1	Drive
Ø	Ø	A
1	Ø	B
Ø	1	C
1	1	D

11.1.5 When the menu choice is made, the screen is blanked by means of a hardware gate. This leaves the menu image still in display RAM, it just does not show. The loaded program should send escape strings to erase the screen and home the cursor, then unblank the display by writing an 83 (hex) to 8Ø88 port ØA (hex). This port should not be written to at any other time as it contains other bits which could cause major problems if they do not agree with an internally maintained copy of the port. The firmware properly maintains this port at all other times for NVM and Z8Ø control.

11.2 SUPPORT BOOT OF THE WINCHESTER DISK

Booting from the Winchester disk will be supported in the Start-up menu. The selection item, for booting from the winchester disk, will always be displayed. If the winchester disk is not present, a system message will be displayed in response to attempting the boot. The boot process will only consist of reading in the Wini boot block and passing control to the loaded code. The loaded code is responsible for selecting the partition to use, etc. If the Wini option is not present the message 'Drive not ready' will be displayed. The boot loading process will be similar to that used for the floppy disk drives. Instead of being Z8Ø code it will be 8Ø88 code and the first byte loaded at location 1ØØØH must be 9ØH (8Ø88 NOP).

Since the 8Ø88 is doing the remainder of the boot process (instead of the Z8Ø), there is no time-out error detection required. If the 8Ø88 loaded code goes off into limbo, the only recovery will be to reset the system. If it goes out in such a way as to prevent keyboard interrupts, the only recovery will be to cycle the power. The actual boot process consists of loading track Ø, sector 1 from the Wini into address 1ØØØH. If this process fails the error message 'MESSAGE 11, system load incomplete' is displayed. If the first byte of the loaded code is not 9ØH, the error message 'MESSAGE 23, non-system diskette' is displayed. If the previous two steps are completed successfully, the firmware then does a far call to address 1ØØ1H. If the loaded code or any code that it loads determines that there is some problem that requires terminating the process, it may execute a far return (using the original stack), in which case the firmware will regain control and the message 'MESSAGE 9, system load incomplete' is displayed and the opening menu is redisplayed allowing for another attempt or other choices.

12 MENU SELECTION PROCESS

After initialization and self-test (or reset), the operator is presented with a choice of things to do.

1. VT1Ø2 terminal mode - system looks like a VT1Ø2 to a host connected to the communications port.

2. Boot operating system - read in and start the operating system, drive is selectable

3. Run more extensive self-tests.

12.1 SUPPORT AUTO-BOOT ON POWER-UP OF A SELECTED DRIVE

The operator will be able specify a specific disk drive to boot from automatically on power-up. A Set-Up field will be defined for selection of the appropriate boot drive. Automatic booting will occur at power-up and on system reset if it is selected for in Set-Up. If the boot procedure cannot be successfully completed the Start-up menu will be displayed. The operator can select drives A, B, C, D, or the Hard Disk. Only the selected drive will accessed for the attempted boot. Multiple drives will not be searched.

13 SYSTEM PARAMETER INFORMATION

This is a word of data which maintains bit flags which define the system state. It is used mainly by the firmware, but is defined here for use by special routines (e.g. 3277 emulator) that need to know about these parameters in order to avoid problems.

Location Mnemonic - Syspar

Address - EFØØ:FFE
Bit Assignments

Bit Ø - Emulator Mode flag
Ø = console mode
1 = terminal mode

Bit 1 - On/Off Line flag
Ø = On Line
1 = Off Line, Local

Bit 2 - Set-Up Mode flag
Ø = normal
1 = in Set-Up

Bit 3 - Hold Screen Mode flag
Ø = normal
1 = Hold Screen in effect

Bit 4 - Scroll In Process flag
Ø = normal
1 = smooth scroll in process

Bits 5-6 reserved

Bit 7 - Print Screen Key flag
Ø = not pressed
1 = key pressed

Bit 8 - Bundle Card Option present flag
Ø = option present
1 = option not present

Bit 9 - Floppy Controller Board Present flag
Ø = floppy present
1 = floppy not present

Bit 1Ø - Graphics Option Present flag
Ø = graphics option present
1 = graphics option not present

Bit 11 - Memory Option Present flag
Ø = PC1ØØ-A Style memory option present
1 = PC1ØØ-A Style memory option not present
PC1ØØ-B Style memory option may or may not be present (determined by firmware)

Bits 12 - 15 reserved

14 DIAGNOSTIC AND MANUFACTURING SUPPORT

14.1 FIXED ENTRY POINT FOR MANUFACTURING ROM DIAGNOSTICS

A fixed entry point for the external manufacturing ROM diagnostics has been provided by means of the existing indirection jump table structure. This guarantees that even though the future location of the ultimate re-entry to the diagnostic code may be different, by using a fixed location for a JMP to the final location, the overall result is a fixed location for the external routine to access. The address of this JMP instruction that will be maintained for all versions of the 1ØØB ROMs is F4ØØ:ØØØ6. A far jump to this address will result in a near jump to label JSTSEG2, which is the desired re-entry to the diagnostics.

14.1.1 Head Load Timing Test

The head load timing diagnostic test is modified for a low limit value of 23Ø to 475 ms. This is based on measurements of the RX5Ø and a plus/minus 1Ø% tolerance.

14.1.2 Memory Diagnostic For Option Configurations

The Power-up memory diagnostics test the lower 128KB of memory. The self-test memory diagnostics test the full memory configuration. At power-up, the diagnostics size memory. Memory is sized in 64KB increments. If the sized memory is identical to the memory size currently saved in NVM, the sizing terminates normally with no message.

If the sized memory is different from that currently saved in NVM, the NVM contents will be changed to reflect the new memory size. A message will be placed on the Start-up menu screen in the upper left hand corner, informing the user of the new memory size. The user is not be allowed to explicitly change the memory size, contained in NVM, by means of Set-up. Set-up displays the total memory size as part of its fixed header information. The default memory size is 128KB. The main Set-Up screen reflects the system memory size, as determined by start-up sizing code.

14.1.3 Memory Test and Initialize Routines

The memory test is modified to account for 128KB of RAM in the base system and all the potential memory option sizes. It will recognize the old RAM memory options (64 and 192 K) by the 'option present' hardware signal. The new RAM option is recognized by the presence of actual memory with the absence of the hardware 'option present' signal. The RAM initialization to set the parity is modified to account for the maximum memory configuration. If the memory size found does not agree with the size stored in NVM, the new size will be stored in NVM and a warning message will be displayed as part of the opening menu. The message will consist of the following string with the xxx part filled in with actual total memory size.

New memory size = xxxK

14.1.4 Set-Up Algorithms For New Memory Sizes

The main Set-Up screen reflects the amount of memory in the system. The field will be informational only. The user may no longer set the memory field. The total memory size is displayed as xxxK on the line between the version number and the line/local status.

14.1.5 Memory Parity Option Test

PC100-B memory parity starts at address 128K. The memory tests start parity testing at the 128K address.

14.1.6 Extended Initialization Of Option Memory

Option memory must be written to, to initialize the parity flag for each byte. The diagnostics are modified to initialize the extended option memory.

14.1.7 Staggered Power-Up Of Winchester and RX50

The power-up of the Wini and RX50 will be staggered so the power supply surge will be within limits. The Wini will be powered up along with all the rest of the system. The RX50 motor-on will not occur until up approximately 15 seconds later.

14.1.8 Error Message Storage In ROM

The error messages have been modified to reduce the amount of text required. Each error message will consist of three parts: a fixed introductory part (See user's guide - message), a variable message number to define the specific problem, (nn,), and an optional text string to provide a general indication if appropriate. The optional text string defines the 'field replaceable unit' where the problem is occurring or be a reminder as in 'interrupts off'. This approach provides the knowledgeable user with a reminder and the new user with a reference message number which he can look up for detailed information on the problem. The following error message assignments have been made. The error numbers are the same as those used in the foreign language translations of the 100A to maintain a sense of continuity.

Original English Text	New English Text
VIDEO	MESSAGE 1,main board
UNSOLICITED INTERRUPTS	MESSAGE 2,main board
INDEX PULSE	MESSAGE 3,drive A (or B as appropriate)
MOTOR SPEED	MESSAGE 4,drive A (or B as appropriate)
SEEK	MESSAGE 5,drive A (or B as appropriate)
READ SECTOR	MESSAGE 6,drive A (or B as appropriate)
RESTORE	MESSAGE 7,drive A (or B as appropriate)
STEP	MESSAGE 8,drive A (or B as appropriate)
SYSTEM LOAD	MESSAGE 9,system load incomplete
VIDEO VFR	MESSAGE 10,main board
BOOT LOAD	MESSAGE 11,system load incomplete
NOT READY(during self-test)	MESSAGE 12,drive not ready

Original English Text	New English Text
KEYBOARD	MESSAGE 13,keyboard
NVM DATA	MESSAGE 14,main board
INTERRUPTS OFF	MESSAGE 16,interrupts off
VIDEO RAM	MESSAGE 17,main board
Z8ØCRC	MESSAGE 18,main board
RAM Ø-64K	MESSAGE 19,main board
UNSOLICITED INTERRUPTS Z8Ø	MESSAGE 2Ø,main board
NOT READY (during boot)	MESSAGE 21,drive not ready
REMOVE CARD OR DISKETTE	MESSAGE 22,remove card or diskette
NON-SYSTEM DISKETTE	MESSAGE 23,non-system diskette
	MESSAGE 24,new memory size = xxxK
SET UP DEFAULTS STORED	MESSAGE 25,set up defaults stored
RAM ARBITRATION	MESSAGE 26,main board
RAM OPTION	MESSAGE 27,memory board
RX5Ø CONTROLLER BOARD	MESSAGE 28,RX5Ø controller board
Z8Ø RESPONSE	MESSAGE 29,main board
ROM CRC, ROM #Ø	MESSAGE 3Ø,main board
ROM CRC, ROM #1	MESSAGE 31,main board
CONTENTION	MESSAGE 33,contention
MAIN BOARD	
PRINTER PORT	MESSAGE 4Ø,main board
KEYBOARD PORT	MESSAGE 5Ø,main board
COMM PORT	MESSAGE 6Ø,main board

15 POWER SUPPLY

The output connector on the power supply is a 13-pin in-line connector with the following pinout:

1 ACOK This signal indicates the presence or absence of valid ac power entering the power supply. When valid ac power is present, this signal will be high (open circuit) and when the ac power is lower than the required minimum input voltage, this signal will be low (short circuit to logic ground).

Low State (ac power invalid): The voltage level of this signal is Ø.45 volts maximum when sinking 2.Ø milliamperes.

High State (ac power valid): The voltage on this signal is pulled-up by external circuitry. When pulled up to 1Ø.Ø volts, the leakage current to logic ground shall be 25 microamperes maximum at the maximum external pull-up voltage of 1Ø.Ø volts.

Transition Times: The rise time (1Ø% to 9Ø%) and fall time (9Ø% to 1Ø%) shall be 1.Ø microsecond maximum.

2 VBIAS When the ac input power is within its valid range, this source has an open circuit voltage of 12.Ø volts + 1Ø% and a source impedance of 47Ø ohms + 1Ø%. This signal is used for manufacturing to automate test monitoring via connecting the LED write signal through a jumper to this pin.

Power-Up: During power-up of the power supply, this source has an open circuit voltage of 8.Ø volts minimum at the time when the dc output voltages start to increase from zero volts.

Power Down: During power-down of the power supply, the voltage on this source decreases toward zero. Due to output loading variations, there is no definable relationship between the decay of this output and the decay of the dc output voltages.

Note

This pin, on the corresponding motherboard connector, is used for manufacturing diagnostics purposes. A jumper is installed to connect the motherboard circuitry to this pin, for the purpose of enabling this feature. Otherwise this pin appears as an open circuit on the motherboard. Should the jumper be installed and the power supply connected and operated, circuit damage to the motherboard could result.

3 Key This pin must be missing on the connectors.

4 -12V OUT Maximum Current -12 VDC, Ø.Ø amperes minimum to Ø.35 amperes maximum

Output `Voltage Variations:

Total Tolerance	+ 7%
Initial Tolerance	+ 3%
Line Regulation	+ 1.5%
Load Regulation	+ 4.Ø%
5.1 Volt Load Interaction	+ 3.Ø%
Temperature Stability	+ Ø.Ø5%/C degrees
Long Term Stability	+ 1%/1ØØØ hrs
Ripple and Noise	12Ø millivolts, peak to peak
Short Circuit Current	3.Ø amperes, maximum
Overvoltage Protection	
Range,Minimum Trip Point	-13.Ø Volts
Absolute Maximum Output	-15.Ø Volts

5,6 +12.2V OUT Maximum Current

+12 VDC, Ø.9 ampere minimum to 6.7 amperes maximum, steady state
9.5 amperes maximum, transient(*)

Output Voltage Variations

Total Tolerance	+ 6%
Initial Tolerance	+ 2%
Line Regulation	+ 1%
Load Regulation	+ 3%
5.1 Volt Load Interaction	+ 3%
Temperature Stability	+ Ø.Ø5 %/C degrees
Long Term Stability	+ 1%/1ØØØ hrs
Ripple and Noise	75 millivolts, peak to peak

Overcurrent Trip Point Minimum: 9.5 amperes (*)
Maximum: 11.5 amperes
Short Circuit Current 4.Ø amperes (maximum)
Overvoltage Protection Range: not applicable

7,8,9 +5.1 V OUT Maximum Current: +5 VDC 2.5 amperes minimum to 11.5 amperes maximum

Output Voltage Variations:

Total Tolerance	+ 6%
Initial Tolerance	+ 2%
Line Regulation	+ 1%
Load Regulation	+ 3%
12.1 Volt Load Interaction	+ 3%
Temperature Stability	+ Ø.Ø5%/C degrees
Long Term Stability	+ 1%/1ØØØ h
Ripple and Noise	5Ø millivolts, peak to peak
Overcurrent Trip Point,	minimum: 12.Ø amperes maximum: 14.Ø amperes

Short Circuit Current:	8.Ø amperes, maximum
Overvoltage Protection Range,	
Minimum trip point:	5.8Ø volts
Absolute maximum output: Voltage	7.Ø volts

1Ø,11,
12,13 DC Power Return Signal Ground

(*) - The +12.2 Volt output shall be capable of sourcing 9.5 amperes for a transient of up to 3ØØ mS. Continuous current draw at this level will damage the supply.

16 CONNECTOR OUTPUTS

16.1 VIDEO INTERFACE CONNECTOR

This connector is a 15-pin D-type female connector supplying interface signals and power to the PC1ØØ-B monitor and keyboard with the following pinout:

Pin	Name	Description
1	Red Shield	Ground connector for red gun shield
2	Green Shield	Ground connector for green gun shield
3	Blue Shield	Ground connector for blue gun shield
4	Mono Shield	Ground connector for monochrome video gun shield
5,6	GND	+12V returns
7,8	+12V	+12V DC to monitor and keyboard
9	Blue	RS17Ø "like" composite red gun output
1Ø	Green	RS17Ø "like" composite green gun output
11	Red	RS17Ø "like" composite blue gun output
12	Mono video	RS17Ø "like" composite monochrome video output
13	Not used	
14	KBD RCV data	RS423 serial data from keyboard
15	KBD TX data	RS423 serial data to keyboard

Note

In a system configuration with a graphics option and color monitor only, the green gun of the color monitor would normally be connected to the monochrome video output, rather than the green video output.

16.2 COMMUNICATIONS INTERFACE CONNECTOR

Pin	Signal Description	Mnemonic	Direction
1	Protective Ground	PROT GND	
2	Transmit Data	XMIT DATA	
3	Receive Data	REC DATA	
4	Request To Send	RTS	
5	Clear To Send	CTS	
6	Data Set Ready	DSR	
7	Signal Ground	GND	
8	Receive Line Signal Det.	RLSD	
9	Not Used	N/U	
1Ø	Not Used	N/U	
11	Not Used	N/U	
12	Speed Indicator/Secondary Receive Line Signal Det.	SI/SRLSD	
13	Not used		
14	Not used		
15	Send Clock	SEND CLK	
16	Not Used	N/U	
17	Receive Clock	REC CLK	
18	Not Used	N/U	
19	Secondary Request To Send	SRTS	
2Ø	Data Terminal Ready	DTR	
21	Not Used	N/U	
22	Ring Indicator	RI	
23	Speed Select	SPDSEL	
24	Not Used	N/U	
25	Not Used	N/U	

16.2.1 Communications Signal Descriptions

Note

The following terminology is used interchangeably to describe the communications signals:

Negative Voltage = 1 = Mark = OFF
Positive Voltage = Ø = Space = ON

1. Protective Ground - This contact is connected to logic and chassis ground via a wire jumper, W17. This jumper may be cut out if local conditions require.

2. Signal Ground - This circuit establishes the common ground reference potential for all interface circuits except protective ground.

3. Transmitted Data (output) - Signals on the line represent the serially encoded characters that are transmitted from the communications port. This circuit is held in the marking state during intervals between characters and at all times when no data is being transmitted.

4. Receive Data (input) - Signals on this circuit represent the serially-encoded characters to be received.

5. Request to Send (output) - Assertion of this signal indicates that the channel is ready for transmission.

6. Clear to Send (input) - When this signal is asserted, it indicates that the modem is ready for transmission.

7. Data Set Ready (input) - The on condition of DSR indicates that the modem is in data mode, and that the control signals asserted by the modem are valid.

8. Receive Line Signal Detector (input) - Also called Carrier Detect. The modem asserts this signal ON when the received signal is of sufficient quality and magnitude.

9. Data Terminal Ready (output) - This signal is turned ON whenever the channel is ready for transmission.

1Ø. Ring Indicator (input) - The ON condition indicates that a ringing signal is being received from the comm line.

11. Speed Indicator (input) - This signal allows some modems to control channel bit rates.

12. Secondary Receive Line Signal Detect (input) - This circuit is used in half duplex coded control with reverse channel.

Note

> Speed Indicator and Secondary Receive Line Signal Detect are two different uses of the same physical line.

13. Speed Select (output) - This signal allows the 8Ø88 to control the modulation method of the modem to coincide with its selected bit rate.

14. Secondary Request to Send (output) - This signal is used for HDX restraint mode and Asymetric FDX Secondary Request to Send.

15. Secondary Clear to Send (input) - In FDX, this signal is the same as Clear to Send. In Asymetric FDX, it provides the functionality for the secondary channel.

16. Secondary Transmitted Data (output) - In FDX, this signal is the same as Transmitted Data, but when operating in Asymetric FDX, it provides functionality for the secondary channel.

17. Send Clock (input) - This is the external transmit clock that is supplied by the modem substituted for the communication transmit clock when the synchronous select bit is set.

18. Receive Clock (input) - This is the external receive clock that is supplied by the modem substituted for the communication receive clock when the synchronous select bit is set.

Note

The Communications Interface connector is configured as Data Terminal Equipment (DTE).

16.3 PRINTER PORT INTERFACE CONNECTOR

Pin	Signal Description	Mnemonic	Direction
1	Protective Ground	PROT GND	------
2	Receive Data	RXD	Output
3	Transmit Data	TXD	Input
5	Clear to Send	CTS	Output *
6	Data Set Ready	DSR	Output *
7	Signal Ground	GND	------
2Ø	Data Terminal Ready	DTR	Input

Notes

*This output is always asserted high.

The Protective Ground contact is connected to logic and chassis ground via a wire jumper, W16. This jumper may be cut out if local conditions require.

The Printer Interface connector is configured as Data Communications Equipment (DCE).

16.4 FLOPPY INTERFACE CONNECTOR

Pin	Signal Description	Mnemonic
1	Ground	GND
2	Track greater than 43	TG43
3	Ground	GND
4	Not Used	N/U
5	Ground	GND
6	Select 3	SEL3 L
7	Ground	GND
8	Index	INDEX L
9	Ground	GND
1Ø	Select Ø	SELØ L
11	Ground	GND
12	Select 1	SEL1 L
13	Ground	GND
14	Select 2	SEL2 L
15	Ground	GND
16	Motor On	MOTOR ON L
17	Ground	GND
18	Direction	DIR L
19	Ground	GND
2Ø	Step	STEP L
21	Ground	GND
22	Write Data	WRT DATA L
23	Ground	GND
24	Write Gate	WG L
25	Ground	GND
26	Track ØØ	TKØØ L
27	Ground	GND
28	Write Protect	WRT PRT L
29	Ground	GND
3Ø	Read Data	RD DATA L
31	Ground	GND
32	Side Select	Side OH
33	Ground	GND
34	Drive Ready	READY L

16.4.1 Floppy Signal Descriptions

1. Select Unit 3 (output, Pin 6) - When asserted, this signal indicates that the current disk in position 3 is selected.

2. Index Pulse (input, Pin 8) - This input informs the controller when the index hole is encountered on the diskette. Minimum pulse width is 2Ø microseconds.

3. Select Unit 2 (output, Pin 14) - When asserted, this signal indicates that the current disk in position 2 is selected.

4. Select Unit 1 (output, Pin 12) - When asserted, this signal indicates that the current disk in position 1 is selected.

5. Select Unit Ø (output, Pin 1Ø) - When asserted, this signal indicates that the current disk in position Ø is selected.

6. Motor On (output, Pin 16) - When asserted, this signal turns on the drive's spindle motor.

7. Stepping Direction Control (output, Pin 18) - This direction signal is an active low when stepping the head toward the spindle and active high when stepping the head away from the spindle.

8. Step Pulse (output, Pin 2Ø) - This is a 2-microsecond pulse to move the head one track. The direction of the step is determined by the direction output.

9. Write Data (output, Pin 22) - This is a 5ØØ ns pulse generated for each flux transition.

1Ø. Write Gate (output, Pin 24) - This output is made valid before writing is to be performed to the diskette.

11. Track ØØ (input, Pin 26) - When asserted, this signal informs the controller that the R/W head is positioned over track Ø.

12. Write Protect (input, Pin 28) - This input is sampled whenever a write command is received. When asserted, the command terminates and sets the write protect status bit in the FDC status register.

13. Read Data (input, Pin 3Ø) - This is the raw data signal from the drive. This signal should be a negative pulse from a minimum of 75Ø ns to a maximum of 125Ø ns for each flux transition.

14. Side Select (output, Pin 32) - When high, the outer surface is selected. At present, only single surface drives are available, so this pin would always be high.

15. Ready (input, Pin 34) - This bit indicates drive readiness and is sampled for a logic high before a read or write operation. This signal means that a disk is in place and the door is closed and drive selected. The motor does not have to be on.

17 OPTIONS

The following assemblies shall be also be offered as options to the base system:

1. Expansion RX5Ø consisting of:

   a. RX5Ø disk drive
   b. Data cable for second disk drive unit. This cable will be longer than the standard drive cable.

2. Color graphics, including a controller module, and optional color monitor and cable.

3. Extended communications, giving Bit/Byte/Async communications, as well as a high speed networking capability.

4. Winchester Disk Storage Sub-system (integral to system box)

18 CABLES

The following cables are included with the system:

1. Monitor cable - transports monochromatic RS-17Ø "like" video signals to the monitor and supplies the keyboard interface. This cable carries all power and ground to the monitor and keyboard. The cable runs external to the system.

2. RX5Ø cable - is be a 34-pin ribbon cable to supply interface and ground from the system board to a single RX5Ø disk drive.

3. Power harness - carries all DC power to the system board and up to two disk drive units.

The following cables are optionally offered with the system. They are non-standard cables.

1. RX5Ø add-on cable - allows an upgrade to a second RX5Ø disk drive.

2. Video color cable - allows connection to a color monitor. This cable supplies a keyboard connection to the system. Used with color graphics option.

19 ENVIRONMENTAL

The PC1ØØ-B meets the requirements of the DEC STD 1Ø2, Class A. The PC1ØØ-B and all peripherals as a part of the system meet the requirements for FCC Class B emitted radiation and conducted.

2Ø RELIABILITY

The PC1ØØ-B demonstrates mean time between failures of no less than 28ØØ hours. This correlates to roughly one year of operation.

21 PHYSICAL PACKAGING

The outer measurements of the BA25 are 17.5 inches in width, 13.625 inches in depth, and 6.Ø inches height. On the front of the unit is the main power switch, and a bezel with pop-out plugs for the RX5Ø disk drive(s). The back of the unit supplies access for ac power and primary circuit selection switch and circuit breaker, and access to standard and option connectors. The packaging includes a fan.

The system board is housed in the lower portion of the BA25 and is encased in sheet metal to minimize RFI problems. Enough space is available to support another plane of circuit at in the very least 7/8-inch above the system module.

22 VIDEO CHARACTER SET

See Appendix A.

23 APPLICABLE STANDARDS AND REGULATIONS

The PC1ØØ-B complies with the following standards:

EL-ØØ119-ØØ	DEC STD 119 - DIGITAL PRODUCT SAFETY
EL-ØØ1Ø2-ØØ	DEC STD 1Ø2 - ENVIRONMENTAL STANDARD FOR COMPUTERS AND PERIPHERALS. The PC1ØØ-B will be a Class A product.
EL-ØØ122-ØØ	DEC STD 122 - AC POWER LINE STANDARD
EL-ØØ1Ø3-ØØ	DEC STD 1Ø3 - ELECTROMAGNETIC COMPATIBILITY (EMC) HARDWARE DESIGN REQUIREMENTS. The PC1ØØ-B will meet FCC Class B Level.
EL-ØØØ52-Ø1	DEC STD Ø52-1 OPERATIONAL REQUIREMENTS FOR ASYNCHRONOUS, FULL DUPLEX, SERIAL TERMINALS AND SYSTEM INTERFACES OPERATING AS DTE'S CONNECTED TO EIA RS-232 OR CCITT V.28 POINT-TO-POINT MODEMS.
EL-ØØ11Ø-ØØ	DEC STD 11Ø ESCAPE SEQUENCE STANDARD
EL-ØØ111-ØØ	DEC STD 111 TERMINAL SYNCHRONIZATION STANDARD
EL-ØØ1Ø7-ØØ	DEC STD 1Ø7 DIGITAL STANDARD FOR TERMINAL KEYBOARDS
EL-ØØ138-ØØ	DEC STD 138 REGISTRY OF CONTROL FUNCTIONS (proposed)

In addition, the following non-DEC standards have been used in the design of the PC1ØØ:

ANSI X3.16	Character Structure and Character Parity Sense
ANSI X3.4-1977	USA Standard Code for Information Interchange (ASCII)
ANSI X3.41-1974	Code Extension Techniques for Use With ASCII
ANSI X3.64-1977	Additional Controls for Use With ASCII

23 APPLICABLE STANDARDS AND REGULATIONS (continued)

UL-478	Electronic Data-Processing Units and Systems
CSA C22.2, No. 54	Canadian Electronic Code, Part II, Safety Standards for Electrical Equipment
VDE Ø871	Limits of Radio Interference from Radio Frequency Apparatus and Installations
VDE Ø875	Regulations for Radio Frequency Suppression
IEC 485	Safety of Data Processing Equipment
FCC Part 15, Subpart J	Rules and Regulations - Radio
EIA RS17Ø	Electrical Performance Standards - Monochrome Television Studio Facilities
CCITT Recommendation	List of Definitions for Interchange Circuit V.24 Between Data Terminal Equipment and Data Circuit Terminating Equipment
CCITT Recommendation	Electrical Characteristic for Unbalanced V.28 Double-Current Interchange Circuit

Rainbow™ 100+/100B
System Specification
QV069–GZ

READER'S COMMENTS

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

□ First-time computer user
□ Experienced computer user
□ Application package user
□ Programmer
□ Other (please specify)

Name

Date

Organization

Street

City

State

Zip Code
or Country

------Do Not Tear - Fold Here and Tape--

SOFTWARE PUBLICATIONS
200 FOREST STREET MRO1–2/L12
MARLBOROUGH, MA 01752

----- Do Not Tear - Fold Here and Tape--

Cut Along Dotted Line

Rainbow™ 100+/100B

Terminal Emulation Manual

digital equipment corporation

First Printing, June 1984

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

CP/M and CP/M–86 are registered trademarks of Digital Research Inc.
CP/M–80 is a trademark of Digital Research Inc.

Z80 is a registered trademark of Zilog, Inc.

8088 is a registered trademark of Intel Corporation.

The following are trademarks of Digital Equipment Corporation:

digital™

DEC
DECmate
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL

MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
RSX

UNIBUS
VAX
VMS
VT
Work Processor

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

Printed in U.S.A.

CONTENTS

PREFACE

CHAPTER 1 TRANSMITTED CHARACTERS

INTRODUCTION 1
STANDARD KEYS 1
Cursor Control Keys 2
Control Character Keys 3
FUNCTION KEYS 5
Break 5
NUMERIC KEYPAD KEYS 7

CHAPTER 2 RECEIVED CHARACTER PROCESSING

GENERAL 9
RECEIVED CHARACTERS 9
CONSOLE MODE TABLES 9
DISPLAY CHARACTERS 12
CONTROL CHARACTERS 12
ESCAPE AND CONTROL SEQUENCES 13
Error Recovery 14
ANSI-Compatible Sequences 17
Set-Up Feature and Mode Selection 17
ANSI/VT52 Compatibility 21
Scrolling 21
Scrolling Region 22
Origin 22
Cursor Positioning 23
Columns Per Line 25
Auto Wrap 26
Screen Background 26
Line Feed/New Line 27
Keyboard Action 27
Auto Repeat 28
Local Echo (Keyboard Send-Receive) 28
Cursor Key Character Selection 28
Keypad Character Selection 29
Character Sets and Selection 31
Character Attributes 36
Tab Stops 37
Line Attributes 37
Erasing 38
Computer Editing 39
Inserting and Replacing Characters 39
Printing in Terminal Mode 40
Printer Extent in Terminal Mode 41
Print Termination Character in Terminal Mode . 42
Reports 42
Adjustments 44
VT52-Compatible Sequences 44
Modes 44
ANSI/VT52 Compatibility 44
Cursor Positioning 45
Keypad Character Selection 46
Character Sets and Selection 48
Erasing 49
Printing in Terminal Mode 50
Reports 51

APPENDIX A PROGRAMMING SUMMARY

GENERAL 52

APPENDIX B CONTROL FUNCTIONS (SEQUENCE FORMATS)

GENERAL 61
Control Functions 61
Escape and Control Sequences 65
Escape Sequence Introducer 65
Intermediate Characters 65
Final Character 65
Control Sequence Format 66
Control Sequence Introducer 66
Parameter Characters 66
Intermediate Characters 66
Final Character 67

APPENDIX C RAINBOW 100 COMPUTER AND VT100 TERMINAL FAMILY DIFFERENCES

DEC'S MULTINATIONAL 8-BIT CHARACTER 68
MULTINATIONAL 8-BIT CHARACTER CODES 68
8-BIT CHARACTER CODES 69
Cl CONTROL CODES 69
KEYBOARD AND 8-BIT KEY CODES 69
COMPOSE CHARACTER SEQUENCES 69
TWO-KEY COMPOSE SEQUENCE 69
THREE-KEY COMPOSE SEQUENCE 70
KEYBOARD CONTROL CODE GENERATION 70
SET-UP PURGING KEYBOARD BUFFER 70
WAIT INDICATOR 70
KEYBOARD PRINT SCREEN KEY IN TERMINAL MODE 70
KEYBOARD HOLD SCREEN KEY 71
KEYBOARD CURSOR KEY MODES 71
PRINTER CHARACTER SETS IN TERMINAL MODE 71
PRINTING BLOB CHARACTERS IN TERMINAL MODE 71
PRINTER PORT DEFAULTS 71
PRINT CURSOR LINE OPERATION IN TERMINAL MODE . . . 72
PRINTER PORT STATUS REQUEST IN TERMINAL MODE . . . 72
TERMINAL ID 72
INSERT AND DELETE LINE ESCAPE SEQUENCES 72
ALTERNATE ROM CHARACTER SETS 72
ALTERNATE ROM AND LED ESCAPE SEQUENCES 72
G2 AND G3 CHARACTER SETS 73
ABORTING ESCAPE SEQUENCES BY INTERMEDIATE CHARACTERS 73
INSERT AND REPLACE MODES 73
SELFTEST ESCAPE SEQUENCES 73
RESET TO INITIAL STATE 73
VT52 MODE AND ORIGIN MODE 73
AUTOWRAP MODE 73
TAB AND AUTO WRAP 74
XON/XOFF PROTOCOL AND BUFFER SIZE IN TERMINAL MODE 74
FULL DUPLEX COMMUNICATION PROTOCOL IN TERMINAL MODE 74
HALF DUPLEX COMMUNICATION SUPPORT IN TERMINAL MODE 74

APPENDIX D INTERNATIONAL LANGUAGE KEYBOARDS

APPENDIX E COMPOSE SEQUENCES

APPENDIX F 7-BIT/DEC 8-BIT TRANSLATIONS

FIGURES

1 Standard Key Codes 1
2 Editing and Cursor Keys 2
3 Function Keys 5
4 Standard Key Codes 52
5 LK201-AE British Keyboard 75

6 LK201-AA American (English) Keyboard 75
7 LK201-AC Canadian (French) Keyboard 76
8 LK201-AD Danish Keyboard 76
9 LK201-AF Finnish Keyboard 76
10 LK201-AG Austrian/German Keyboard 77
11 LK201-AH Dutch Keyboard 77
12 LK201-AI Italian Keyboard 77
13 LK201-AK Swiss (French) Keyboard 78
14 LK201-AL Swiss (German) Keyboard 78
15 LK201-AM Swedish Keyboard 78
16 LK201-AN Norwegian Keyboard 79
17 LK201-AP Belgian/French Keyboard 79
18 LK201-AT Flemish Keyboard 79
19 LK201-AS Spanish Keyboard 80
20 Mapping Keyboard to National Replacement Characters 83
21 French Canadian Character Set (7-bit) 84
22 Finish Character Set (7-bit) 85
23 French Character Set (7-bit) 86
24 German Character Set (7-bit) 87
25 Italian Character Set (7-bit) 88
26 Norwegian/Danish Character Set (7-bit) 89
27 Spanish Character Set (7-bit) 90
28 Swedish Character Set (7-bit) 91
29 Swiss Character Set (7-bit) 92
30 United Kingdom Character Set (7-bit) 93
31 7-bit/DEC8-bit Translations 94
32 Dutch Character Set (7-bit) 95

TABLES

1 Rainbow 100 Editing and Cursor Keys 2
2 Cursor Control Key Codes 3
3 Control Codes Generated 4
4 Rainbow 100 Key Changes 5
5 Rainbow 100 Function Keys 6
6 Keypad Codes 7
7 7-bit US/UK ASCII Characters 10
8 8-bit Control and Displayable Characters 11
9 Control Characters Recognized by Rainbow 100 Computer 12
10 Escape and Control Sequences 15
11 Set-Up Features and Modes 18
12 ANSI-Specified Modes 19
13 ANSI-Compatible Private Modes 19
14 Permanently Selected Modes 20
15 Line Feed/New Line Feature 27
16 ANSI Cursor Control Key Codes 29
17 ANSI Keypad Codes 30
18 7-bit US/UK ASCII Characters 33
19 8-bit Control and Displayable Characters 34
20 Special Characters and Line Drawing Character Set 35
21 VT52 Keypad Codes 47
22 Special Characters and Line Drawing Set and VT52 Graphics Mode Comparison 49
23 US/UK ASCII Characters 62
24 Control and Displayable Characters 63
25 Special Characters and Line Drawing Set 64
26 Implicit Compose Sequences 81
27 Dead Diacritical Keys 82

PREFACE

INTENDED READER

This guide assumes you are an application programmer.

The information in this guide describes escape sequences and codes used by the Rainbow 100B terminal emulation.

GUIDE ORGANIZATION

Chapter 1 shows the characters transmitted by each terminal key.

Chapter 2 describes how the terminal processes received characters. It also describes the use of control functions. Control functions control the display, processing, and transmission of characters received by the terminal. The application programmer uses the chapter when creating applications software for the terminal.

Appendix A summarizes the character codes and control functions used to program the terminal.

Appendix B describes the ANSI code extension techniques used to create escape and control sequences.

Appendix C describes the differences between the Rainbow 100 computer and a VT102 terminal.

Appendix D contains the international language keyboards

CHAPTER 1

TRANSMITTED CHARACTERS

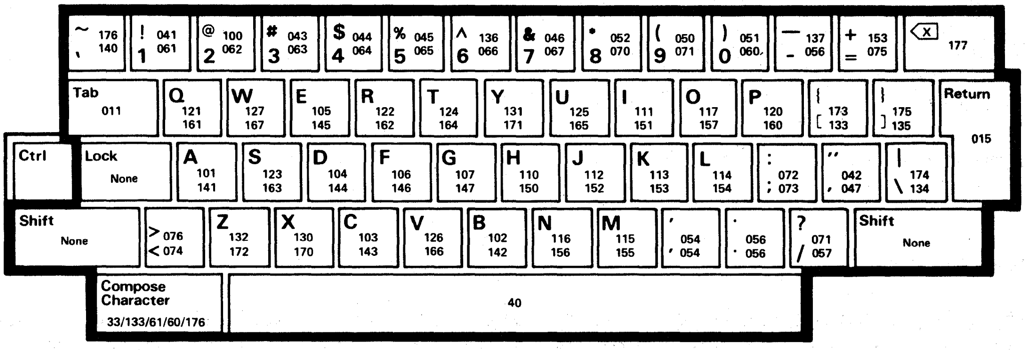
INTRODUCTION

This chapter describes the characters generated by the Rainbow 100 keyboard. The keys are divided into four groups: standard keys, editing and cursor keys, function keys, and numeric keypad keys.

A distinction is also made between console mode and terminal mode on the Rainbow 100 computer.

STANDARD KEYS

The keyboard generates American Standard Code for Information Interchange (ASCII) characters. The standard keys (Figure 1) generate lowercase ASCII characters when neither **Shift** nor **Lock** is down. These keys generate uppercase ASCII characters when either **Shift** or **Lock** is down. The Lock key can act as either a 'Caps Lock' or 'Shift Lock', selectable from SET-UP.



MR-9592

Figure 1: Standard Key Codes

Editing and Cursor Keys

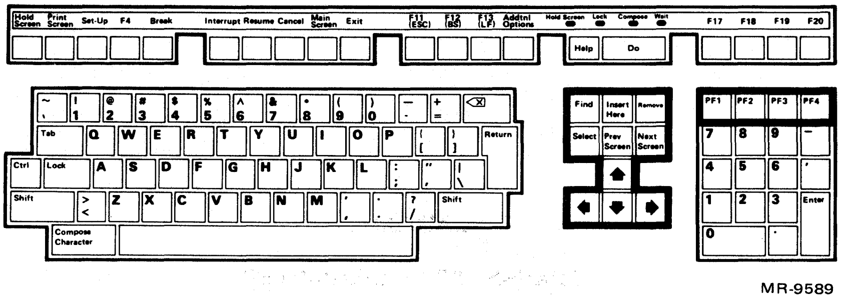


Figure 2: Editing and Cursor Keys

Table 1: Rainbow 100 Editing and Cursor Keys

Key	Characters Generated
Find	ESC [1 ~
Insert Here	ESC [2 ~
Remove	ESC [3 ~
Select	ESC [4 ~
Prev Screen	ESC [5 ~
Next Screen	ESC [6 ~
Up Arrow	ESC [A
Down Arrow	ESC [B
Right Arrow	ESC [C
Left Arrow	ESC [D

Cursor Control Keys

In ANSI mode the cursor keys generate either application or cursor control sequences. Cursor key mode selects the type of sequence.

The cursor keys generate ANSI cursor commands. The computer selects both cursor key mode and keypad mode. See Cursor Key Character Selection in Chapter 2 for more information.

In VT52 mode, the cursor keys only generate VT52 cursor control sequences. Table 2 lists the ANSI and VT52 compatible cursor key characters.

Table 2: Cursor Control Key Codes

	ANSI Mode		**VT52 Mode**
	Cursor Key	**Cursor Key**	
Cursor Keys	**Mode Reset**	**Mode Set**	
	ESC [A 033 133 101	ESC O A 033 117 101	ESC A 033 101
	ESC [B 033 133 102	ESC O B 033 117 102	ESC B 033 102
	ESC [C 033 133 103	ESC O C 033 117 103	ESC C 033 103
	ESC [D 033 133 104	ESC O D 033 117 104	ESC D 033 104

Control Character Keys

Table 3 lists the control characters generated by the keyboard. You can generate control characters in two ways.

- Hold down **Ctrl** and press any key in Table 3 under the Key Pressed column.

- Press any key in Table 3 under the Dedicated Key column. These dedicated keys generate control characters without the use of **Ctrl**.

Different computer systems may use each control character differently.

NOTE

> The Rainbow 100 computer generates some control characters differently than previous DIGITAL terminals. Table 4 lists the changes.

Table 3: Control Codes Generated

Control		**Transmitted**	**Key Pressed**	**Dedicated**
Character	**Mnemonic**	**Code (Octal)**		**Key**
Null	NUL	000	Space Bar	-
Start of heading	SOH	001	**A**	-
Start of text	STX	002	**B**	-
End of text	ETX	003	**C**	-
End of transmission	EOT	004	**D**	-
Enquire	ENQ	005	**E**	-
Acknowledge	ACK	006	**F**	-
Bell	BEL	007	**G**	-
Back space	BS	010	**H**	**Back Space**
Horizontal tabulation	HT	011	**I**	**Tab**
Line Feed	LF	012	**J**	**Line Feed**
Vertical tabulation	VT	013	**K**	-
Form feed	FF	014	**L**	-
Carriage return	CR	015	**M**	**Return***
Shift out	SO	016	**N**	-
Shift in	SI	017	**O**	-
Data link escape	DLE	020	**P**	-
Device control 1	DC1 (XON)	021	**Q**	-
Device control 2	DC2	022	**R**	-
Device control 3	DC3 (XOFF)	023	**S**	-
Device control 4	DC4	024	**T**	-
Negative acknowledge	NAK	025	**U**	-
Synchronous idle	SYN	026	**V**	-
End of transmission block	ETB	027	**W**	-
Cancel previous word or character	CAN	030	**X**	-
End of medium	EM	031	**Y**	-
Substitute	SUB	032	**Z**	-
Escape	ESC	033	**[**	**Escape**
File separator	FS	034	**/**	-
Group separator	GS	035	**]**	-
Record separator	RS	036	**~**	-
Unit separator	US	037	**?**	-
Delete	DEL	177		**Delete**
Null	NUL	000	2 unshifted	
Escape	ESC	033	3 unshifted	
File Sep	FS	034	4 unshifted	
Group Sep	GS	035	5 unshifted	
Record Sep	RS	036	6 unshifted	
Unit Sep	US	037	7 unshifted	
Delete	DEL	177	8 unshifted	

*In numeric keypad mode (application keypad mode off), you can change the **Enter** character code with the line feed/new line feature. When off, this feature causes **Enter** to generate a single control character (CR, octal 015). When on, this feature causes Enter to generate two characters (CR, octal 015 and LF, octal 012).

Table 4: Rainbow 100 Key Changes

Control Code	VT102	Previous Terminals	Rainbow 100 Computer	
NUL (octal 000)	Ctrl Space Bar	Ctrl @	Ctrl Space Bar	Ctrl 2
RS (octal 036)	Ctrl ~	Ctrl ^	Ctrl ~	Ctrl 6
US (octal 037)	Ctrl ?	Ctrl -	Ctrl ?	Ctrl 7
ESC	Ctrl [		Ctrl [	Ctrl 3
FS	Ctrl /		Ctrl /	Ctrl 4
GS	Ctrl]		Ctrl]	Ctrl 5
DEL				

FUNCTION KEYS

The function keys (Figure 3) generate characters used by the computer software or communication system. The following paragraphs describe the function keys.

Break

In terminal mode this key generates a break defined by the computer system when the break enable feature is on. This feature does not affect other key sequences using **Break.**

Hold down **Shift** and press **Break** to generate a long break disconnect. A long break disconnect usually disconnects the terminal from the communication line.

Hold down **Ctrl** and press **Break** to transmit the answerback message. In console mode this key is not functional.

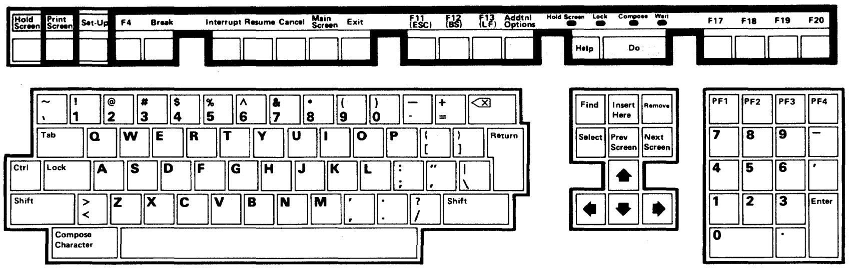


Figure 3: Function Keys

Table 5: Rainbow 100 Function Keys

Key	**Characters Generated**
Print Screen	ESC [12 ~
F4	ESC [14 ~
Interrupt	ESC [17 ~
Resume	ESC [18 ~
Cancel	ESC [19 ~
Main Screen	ESC [20 ~
Exit	ESC [21 ~
(ESC)	ESC
(BS)	BS
(LF)	LF
Addtnl Options	ESC [26 ~
Help	ESC [28 ~
Do	ESC [29 ~
F17	ESC [31 ~
F18	ESC [32 ~
F19	ESC [33 ~
F20	ESC [34 ~

NUMERIC KEYPAD KEYS

These keys generate characters selected by the ANSI/VT52 feature and alternate (application) keypad mode. The computer selects application keypad mode. See Keypad Character Selection in Chapter 2 for more information.

In numeric keypad mode, the numeric keypad generates the numeric, comma, period, and minus sign characters used by the main keyboard. In application keypad mode, the numeric keypad generates escape sequences. Table 6 lists the characters generated by the numeric keypad.

Table 6: Keypad Codes

	ANSI Mode		**VT52 Mode**	
Key	**Numeric Keypad Mode**	**Application Keypad Mode**	**Numeric Keypad Mode**	**Application Keypad Mode**
0	0 060	ESC O p 033 117 160	0 060	ESC ? p 033 077 160
1	1 061	ESC O q 033 117 161	1 060	ESC ? q 033 077 161
2	2 062	ESC O r 033 117 162	2 062	ESC ? r 033 077 162
3	3 063	ESC O s 033 117 163	3 063	ESC ? s 033 077 163
4	4 064	ESC O t 033 117 164	4 064	ESC ? t 033 077 164
5	5 065	ESC O u 033 117 165	5 065	ESC ? u 033 077 165
6	6 066	ESC O v 033 117 166	6 066	ESC ? v 033 077 166
7	7 067	ESC O w 033 117 167	7 067	ESC ? w 033 077 167
8	8 070	ESC O x 033 117 170	8 070	ESC ? x 033 077 170
9	9 071	ESC O y 033 117 171	9 071	ESC ? y 033 077 171

Table 6 (Cont.): Keypad Codes

	ANSI Mode		**VT52 Mode**	
Key	**Numeric Keypad Mode**	**Application Keypad Mode**	**Numeric Keypad Mode**	**Application Keypad Mode**
-	-(minus) 055	**ESC O m** 033 117 155	-(minus) 055	**ESC ? m** 033 077 155*
,	,(comma) 054	**ESC O l** 033 117 154	,(comma) 054	**ESC ? l** 033 077 054*
.	.(period) 056	**ESC O n** 033 117 156	.(period) 056	**ESC ? n** 033 077 156
Enter+	CR or **CR LF** 015 015 012	**ESC O M** 033 117 115	CR or **CR LF** 015 015 012	**ESC ? M** 033 077 115
PF1	**ESC O P** 033 117 120	**ESC O P** 033 117 120	**ESC P** 033 120	**ESC P** 033 120
PF2	**ESC O Q** 033 117 121	**ESC O Q** 033 117 121	**ESC Q** 033 121	**ESC Q** 033 121
PF3	**ESC O R** 033 117 122	**ESC O R** 033 117 122	**ESC R** 033 122	**ESC R** 033 122
PF4	**ESC O S** 033 117 123	**ESC O S** 033 117 123	**ESC S** 033 123	**ESC S** 033 123*

*These sequences are not generated by the VT52 terminal.

+In numeric keypad mode (application keypad mode off), you can change the **Enter** character code with the line feed/new line feature. When off, this feature causes **Enter** to generate a single control character (CR, octal 015). When on, this feature causes **Enter** to generate two characters (CR, octal 015 and LF, octal 012).

CHAPTER 2

RECEIVED CHARACTER PROCESSING

GENERAL

This chapter describes how the Rainbow 100 computer processes received characters. There are two types of received characters, display characters and control functions. The chapter covers all display characters and control functions used by the Rainbow 100 computer.

RECEIVED CHARACTERS

The Rainbow 100 computer processes characters according to American National Standards Institute (ANSI) standards X3.64-1979, X3.4-1977, and X3.41-1974. ANSI standard X3.4 defines the American Standard Code for Information Interchange (ASCII). Table 7 shows each ASCII character with its binary, octal, decimal, and hexadecimal values. ASCII corresponds to the International Standards Organization (ISO) Standard 646 and International Telegraph and Telephone Consultive Committee (CCITT) Alphabet 5.

The Rainbow 100 computer processes a received character based on character types defined by ANSI. Position in the ASCII table tells you whether a character is a control function or display character. The ASCII table is 8 columns wide and 16 rows long. The control functions are in columns 0 and 1. The display characters are in columns 2 through 7. In addition to the characters shown in Table 7, the Rainbow 100 computer displays the 8-bit characters shown in Table 8 and executes the control functions (C1) in columns 8 and 9.

CONSOLE MODE TABLES

In terminal mode, however, it depends on the communication port's configurations. In order to correctly process the 8-bit characters in Table 8, you must have the communication port parameter set to 8 data bits. If it is set to 7 data bits the high order bit is set to zero and the character is processed as though in Table 7.

Table 7: 7-bit US/UK ASCII Characters

	COLUMN	0		1		2		3		4		5		6		7	
	BITS b8 b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	* #/£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

NOTE: DEPENDS ON THE CHARACTER SET SELECTED; U.S.=# U.K.= £

|←C0 CODES→|← GL CODES (ASCII GRAPHICS) →|

KEY

CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MR-9593

Table 8: 8-bit Control and Displayable Characters

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80		220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81		221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82		222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83		223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84		224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85		225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
	206 134 86		226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
	207 135 87		227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
	211 137 89		231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
	214 140 8C		234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D		235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E		236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F		237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←C1 CODES→|← GR CODES (DEC SUPPLEMENTAL GRAPHICS) →|

KEY

CHARACTER	Æ	306	OCTAL
		198	DECIMAL
		C6	HEX

MR-9594

DISPLAY CHARACTERS

Display characters are received characters displayed on the screen. The actual character displayed depends on the character set selected. You select the character set by using control functions. See Character Sets and Selection in this chapter for more information.

CONTROL CHARACTERS

These single-character control functions start, modify, or stop terminal operations; the control functions are not displayed. Table 9 defines the control characters recognized by the terminal. All other control characters are ignored.

Each control character in this chapter has a mnemonic, listed in Table 9. The mnemonic is an abbreviation of the control character name.

Table 9: Control Characters Recognized by Rainbow 100 Computer

Character	Mnemonic	Octal Code	Function
Null	NUL	000	Ignored when received (not stored in input buffer) and used as a fill character
Enquire	ENQ	005	Transmits answerback message
Bell	BEL	007	Generates bell tone.
Backspace	BS	010	Moves cursor to the left one character position; if cursor is at left margin, no action occurs.
Horizontal tab	HT	011	Moves cursor to next tab stop, or to right margin if there are no more tab stops.
Line Feed	LF	012	Causes a line feed or a new operation. (See Line Feed/New Line). Also causes printing in terminal mode only if auto print operation is selected.
Vertical tab	VT	013	Processed as LF.
Form feed	FF	014	Processed as LF.
Carriage return	CR	015	Moves cursor to left margin on current line.
Shift out	SO	016	Selects G1 character set designated by a select character set sequence.
Shift in	SI	017	Selects G0 character set designated by a select character set sequence.

Table 9 (Cont.): Control Characters Recognized by Rainbow 100 Computer

Character	**Mnemonic**	**Octal Code**	**Function**
Device control 1	DC1	021	Processed as XON. DC1 causes the Rainbow 100 computer to resume (if previously stopped by XOFF) transmitting characters only in terminal mode.
Device control 3	DC3	023	Processed as XOFF. DC3 causes the Rainbow 100 computer to stop transmitting all characters except XOFF and XON.
Cancel	CAN	030	If received during an escape or control sequence, cancels the sequence and displays substitution character (cursor).
Substitute	SUB	032	Processed as CAN.
Escape	ESC	033	Processed as an escape sequence introducer.
Index	IND	204	Processes a LF.
Next line	NEL	205	Processes a CR LF sequence.
Horizontal	HTS	210	Sets a horizontal tab at the current cursor location.
Reverse index	RI	215	Equals a reverse line feed.
Single shift 2	SS2	216	Selects G2 character set for the next character only.
Single shift 3	SS3	217	Selects G3 character set for the next character only.
Control sequence introducer	CSI	233	Equals an ESC [.

ESCAPE AND CONTROL SEQUENCES

Escape and control sequences provide additional control functions not provided by the single-character controls of the character set. These multiple-character sequences are not displayed; instead, they control Rainbow 100 computer operation. Escape and control sequences are defined by ANSI X3.41-1977 and X3.64-1979. See Appendix B for more information about sequences and sequence formats.

The ANSI-compatible control functions in this user guide have a mnemonic assigned by ANSI. If the control function is an ANSI private control function (defined by DIGITAL), the mnemonic begins with DEC. The escape and control sequences shown here use ASCII characters. You must type the characters in the sequences exactly as shown (upper or lowercase). The text provides the octal equivalent of each character in the sequence as a second reference. See Table 7 for decimal and hexadecimal representations.

The following section groups sequences by software compatibility (ANSI or VT52) and function (Table 10). Appendix A summarizes all control functions.

Error Recovery

Current standards do not specify the action performed when the terminal receives a control function with an error. Errors are incorrect parameters; invalid control functions. The terminal usually recovers from these errors by performing as much of the function as possible. The specific error recovery procedures are as follows:

- Unrecognized control functions are usually ignored.
- Unsupported control functions (valid control functions not listed in this user guide) are usually ignored, but may produce unexpected results.
- If a 7-bit control character from Table 7 is received within a sequence, the terminal performs the function of the control character, followed by the function of the sequence.
- If cancel (CAN, octal 030) or substitute (SUB octal 032) is received during a sequence, the current sequence is aborted. The terminal displays the substitute character, followed by characters in the sequence received after CAN or SUB.
- If an 8-bit control character from Table 8 is received, the current escape sequence is aborted, and the function of the character is performed.
- If an 8-bit displayable character is received from Table 8, the 8th bit is stripped off and the escape sequence continues as if the equivalent 7-bit remainder had been received.

Table 10: Escape and Control Sequences

Ansi-Compatible Sequences

- Set-Up Feature and Mode Selection
    - Set mode (SM) and reset mode (RM)
- ANSI/VT52 Compatibility
    - ANSI/VT52 mode (DECANM)
- Scrolling
    - Scroll mode (DECSCLM)
- Scrolling Region
    - Set top and bottom margins (DECSTBM)
- Origin
    - Origin mode (DECOM)
- Cursor Positioning
    - Cursor up (CUU)
    - Cursor down (CUD)
    - Cursor forward (CUF)
    - Cursor backward (CUB)
    - Cursor position (CUP)
    - Horizontal and vertical position (HVP)
    - Index (IND)
    - Reverse index (RI)
    - Next line (NEL)
    - Save cursor (DECSC)
    - Restore cursor (DECRC)
- Columns Per Line
    - Column mode (DECCOLM)
- Auto Wrap
    - Auto wrap mode (DECAWM)
- Screen Background
    - Screen mode (DECSCNM)
- Line Feed/New Line
    - Line Feed/New Line mode (LNM)
- Keyboard Action
    - Keyboard action mode (KAM)
- Auto Repeat
    - Auto repeat mode (DECARM)
- *Local Echo
    - Send-receive mode (SRM)
- Cursor Key Character Selection
    - Cursor key mode (DECCKM)
- Keypad Character Selection
    - Numeric keypad (DECKPNM)
    - Application keypad (DECKPAM)
- Character Sets and Selection
    - Select character set (SCS)
    - Single shift 2 (SS2)
    - Single shift 3 (SS3)
- Character Attributes
    - Select graphic rendition (SGR)
- Tab Stops
    - Horizontal tab sets (HTS)
    - Tabulation clear (TBC)
- Line Attributes
    - Double-height line (DECDHL)
    - Single-width line (DECSWL)
    - Double-width line (DECDWL)

Table 10 (Cont.): Escape and Control Sequences

Ansi-Compatible Sequences

- Erasing
  - Delete character (DCH)
  - Insert line (IL)
  - Delete line (DL)
- Inserting and Replacing Characters
  - Insertion-replacement mode (IRM)
- *Printing
  - Media copy (MC)
- *Printer Extent
  - Printer extent mode (DECPEX)
- *Print Termination Character
  - Printer form feed mode (DECPFF)
- Reports
  - Device status report (DSR)
  - Cursor position report (CPR)
  - Device attributes (DA)
  - Identify terminal (DECID)
- Reset
  - Reset to initial state (RIS)
- Adjustments
  - Screen alignment display (DECALN)
- Modes
- ANSI/VT52 Compatibility
  - ANSI mode (DECANM)
- Cursor Positioning
  - Cursor up
  - Cursor down
  - Cursor right
  - Cursor left
  - Cursor to home
  - Direct cursor address
  - Reverse line feed
- Keypad Character Selection
  - Application keypad mode
  - Numeric keypad mode
- Character Sets and Selection
  - Enter graphics mode
  - Exit graphics mode
- Erasing
  - Erase to end of line
  - Erase to end of screen
- *Printing
  - Auto print
  - Print controller
  - Print cursor line
  - Print screen
- Reports
  - Identify

*Only in terminal mode.

ANSI-Compatible Sequences

ANSI-compatible sequences meet ANSI standards X3.64-1979 and X3.41-1974. This section describes the ANSI control functions used by the terminal. You can select ANSI compatibility from the keyboard in Set-Up or have the computer use a sequence. (See VT52-Compatible Sequences in this chapter).

Set-Up Feature and Mode Selection - Set-Up features change how the Rainbow 100 computer operates. You can select these features from the keyboard or through escape sequences.

Some Set-Up features are modes. A mode affects Rainbow 100 computer operation. The Rainbow 100 computer uses the selected mode until you or an escape sequence changes the selection. Table 11 lists Set-Up features and modes. Modes are changed by using set mode (SM) and reset mode (RM) sequences. Set and reset the terminal modes by using the following sequences.

NOTE

> Ps represents a variable parameter selected from a list of parameters. A series of asterisks (***) represent the parameter in the octal sequence. The parameter is transmitted using decimal ASCII characters. When you set several modes with a single SM or RM sequence, a semicolon (;, octal 073) separates parameters.

Set Mode (SM)

```
ESC  [  Ps   ;...  ;  Ps   h
033 133 *** 073   073 *** 150
```

Sets one or more modes specified by selective parameters (Ps) in the parameter string.

Reset Mode (RM)

```
ESC  [  Ps   ;... ;  Ps   l
033 133 *** 073  073 *** 154
```

Resets one or more modes specified by selective parameters (Ps) in the parameter string.

Table 11: Set-Up Features and Modes

Set-Up Feature or Mode	Change by Escape Sequences	Change from Keyboard in Set-Up
On/off line**	No	Yes
Columns per line	Yes (DECCOLM)	Yes
Tab stops	Yes (HTS/TBC)*	Yes
Scroll rate	No	Yes
Auto repeat	Yes (DECARM)	Yes
Screen background	Yes (DECSCNM)	Yes
Cursor	No	Yes
Margin bell volume	No	Yes
Keyclick volume	No	Yes
ANSI/VT52	Yes (DECANM)	Yes
Auto XON/XOFF**	No	Yes
US/UK character set	Yes (SCS)*	Yes
Auto Wrap	Yes (DECAWM)	Yes
Line Feed/New Line	Yes (LNM)	Yes
Local echo**	Yes (SRM)	Yes
Print termination character**	Yes (DECPFF)	Yes
Printer extent**	Yes (DECPEX)	Yes
One or two stop bits	No	Yes
Receive parity	No	Yes
Break enable**	No	Yes
Disconnect character enable**	No	Yes
Disconnect delay**	No	Yes
Auto answerback enable**	No	Yes
Power	No	Yes
Modem data/parity bits	No	Yes
Transmit speed	No	Yes
Receive speed	No	Yes
Modem control**	No	Yes
Printer data/parity bits	No	Yes
Transmit/receive speed	No	Yes
Application keypad mode/ numeric keypad mode	Yes (DECKPAM/DECKPNM)*	No
Cursor key mode	Yes (DECCKM)	No
Origin mode	Yes (DECOM)	No
Insertion-replacement mode	Yes (IRM)	
Country/KBD selected	No	Yes
8-bit/7-bit NRC	No	Yes
Lock mode Caps/Shift	No	Yes

*These features are not changed using the set mode (SM) and reset mode (RM) sequences.

**Happens only in terminal mode.

Table 12 lists the ANSI-specified modes and their selective parameters (Ps). Table 13 lists the ANSI-compatible private modes and their selective parameters. When you change ANSI-compatible private modes, the first character in the parameter string is a question mark (?, octal 077). All parameters in the sequence are interpreted as ANSI compatible private parameters. This chapter explains each mode in detail and provides the sequences to set and reset each mode.

The following example shows the use of the question mark (used with ANSI private parameters) and semicolon (used with multiple parameters). The sequence sets both column and scroll modes.

```
ESC  [   ?   3   ;   4   h
033 133 077 063 073 064 150
```

Table 14 describes modes specified in ANSI X3.64-1979 that are permanently set, permanently reset, or not applicable. See the ANSI standard for more information about these modes.

Table 12: ANSI-Specified Modes

Name	**Mnemonic**	**Parameter (Ps)**
Error (ignored)	-	0
Keyboard action	KAM	2
Insertion-replacement	IRM	4
Line Feed/New Line	LNM	20

Table 13: ANSI-Compatible Private Modes

Name	**Mnemonic**	**Parameter (Ps)**
Error (ignored)	-	0
Cursor key	DECCKM	1
ANSI/VT52	DECANM	2
Column	DECCOLM	3
Scroll	DECSCLM	4
Screen	DECSCNM	5
Origin	DECOM	6
Auto wrap	DECAWN	7
Auto repeat	DECARM	8
Printer form feed*	DECPFF	18
Printer extent*	DECPEX	19

*Happens only in terminal mode.

NOTE

> The application keypad and numeric keypad modes are selected using dedicated sequences, not set and reset mode sequences. See Keypad Character Selection in this chapter for more information.

Table 14: Permanently Selected Modes

Name	Mnemonic	Selection	Function
Control representation	CRM	Reset	Rainbow 100 computer performs control functions without displaying a character to represent control function received.
Editing boundary	EBM	Reset	Characters moved outside the margins are lost; terminal does not perform erasing and cursor positioning functions outside the margins. This does not affect horizontal and vertical position (HVP) and cursor position (CUP) sequences.
Erasure	ERM	Set	All characters displayed can be erased.
Format effector action	FEAM	Reset	Terminal immediately performs control functions that affect the screen display.
Format effector transfer	FETM	N/A	-
Guarded area transfer	GATM	N/A	-
Horizontal editing	HEM	N/A	-
Multiple area transfer	MATM	N/A	-
Positioning unit	PUM	Reset	Terminal specifies horizontal and vertical positioning parameters in control functions in units of character position.
Selected area transfer	SATM	N/A	-
Status reporting transfer	SRTM	Reset	Terminal transmits status reports by using device status report (DSR) sequences.
Tabulation stop	TTM	N/A	-
Vertical editing	VEM	N/A	-

ANSI/VT52 Compatibility - The Rainbow 100 computer is compatible with both ANSI and private DIGITAL standards. Therefore, you can use new software that meets both ANSI standards and existing software designed for previous terminals (such as the VT52).

ANSI-compatible sequences meet ANSI standards X3.64-1979 and X3.41-1974. You select ANSI compatibility by using the ANSI/VT52 mode (DECANM) sequence in VT52 mode. See VT52-Compatible Sequences in this chapter for details on selecting ANSI sequence compatibility. In ANSI mode, the following sequence selects (VT52 mode).

Features and modes selected in ANSI mode are also used in VT52 mode. However, these features and modes usually cannot change in VT52 mode.

VT52 Mode (DECANM)

```
ESC  [   ?   2   1
033 133 077 062 154
```

In ANSI mode, reset selects VT52 compatibility. In VT52 mode, the Rainbow 100 computer responds like a VT52 to private DIGITAL sequences.

Scrolling - Scrolling is the upward or downward movement of existing lines on the screen. This makes room for more display lines at either the top or bottom of the scrolling region. There are two methods of scrolling, jump scroll and smooth scroll. Select the type of scrolling by using the following sequences.

NOTE

> Op In full-duplex communication, the auto XON/XOFF Set-Up feature prevents the loss of received characters when using smooth scroll. If auto XON/XOFF is not used, fill characters are needed.

Scroll Mode (DECSCLM)

```
ESC [ ?  4 h
033 133 077 064 150
```

Set selects smooth scroll. Smooth scroll rate selected in Set-Up.

```
ESC  [   ?   4   1
033 133 077 064 154
```

Reset selects jump scroll. Jump scroll lets the terminal add lines to the screen as fast as possible.

Scrolling Region - This inclusive region is the area of the screen defined by top and bottom margins. The margins determine which screen lines move during scrolling. Characters added outside the scrolling region do not cause the screen to scroll. The minimum size of the scrolling region is two lines. Therefore, the line number of the top margin must be less than the number of the bottom margin. The origin mode selects line numbers relative to the whole screen or the scrolling region.

After the margins are selected, the cursor moves to the home position. The origin mode feature also affects the home position. Select the top and bottom margins of the scrolling region by using the following sequence.

NOTES: When you power up or use the system reset command, the scrolling region becomes the full screen.

Pt and Pb represent variable numeric parameters. The parameters are decimal numbers transmitted to the terminal as ASCII characters. Asterisks (***) represent one or more variable numeric parameters in the octal sequence.

Set Top and Bottom Margins (DECSTBM)

```
ESC [   Pt  ;   Pb  r
033 133 *** 073 *** 162
```

Selects top and bottom margins, defining the scrolling region. Pt is line number of first line in the scrolling region. Pb is line number of bottom line. If Pt and Pb are not selected, the complete screen is used (no margins).

Origin - This mode determines if the cursor can move outside the scrolling region (the area between the top and bottom margins). You can move the cursor outside the margins with the cursor position (CUP) and horizontal and vertical position (HVP) sequences.

Lines on the screen are numbered according to the location of the home position. Home position is always line 1, column 1. The cursor moves to the new home position whenever origin mode is selected. Select origin mode by using the following sequences.

NOTE

When you power up or use the system reset command, origin mode resets.

Origin Mode (DECOM)

```
ESC [   ?   6   h
033 133 077 066 150
```

Set selects home position in scrolling region. Line numbers start at top margin of scrolling region. The cursor cannot move out of scrolling region.

```
ESC [   ?   6   l
033 133 077 066 154
```

Reset selects home position in upper-left corner of screen. Line numbers are independent of the scrolling region (absolute). Use CUP and HVP sequences to move cursor out of scrolling region.

Cursor Positioning - The cursor indicates the active screen position where the next character will appear. The cursor moves:

- One column to the right when a character appears
- One line down after a line feed (LF, octal 012), form feed (FF, octal 014) or vertical tab (VT, octal 013) (Line feed/new line may also move the cursor to the left margin). If at the bottom margin, this causes an upward scroll.
- One line up after a reverse index, if at the top margin, it causes a downward scroll.
- To the left margin after a carriage return (CR, octal 015)
- One column to the left after a backspace (BS, octal 010)
- To the next tab stop (or right margin if no tabs are set) after a horizontal tab character (HT, octal 011)
- To the home position when the top and bottom margins of the scrolling region (DECSTBM) or origin mode (DECOM) selection changes.

You can also move the cursor by using the following sequences.

NOTE

> Pn represents a variable numeric parameter. The parameter is a decimal number transmitted to the terminal by using ASCII characters. If you select no parameter or 0, the terminal assumes the parameter equals 1. Asterisks (***) represent one or more characters in the octal sequence.

Cursor Up (CUU)

```
ESC [   Pn  A
033 133 *** 101
```

Moves cursor up Pn lines in same column. Cursor stops at top margin.

Cursor Down (CUD)

```
ESC [   Pn  B
033 133 *** 102
```

Moves cursor down Pn lines in same column. Cursor stops at bottom margin.

Cursor Forward (CUF)

```
ESC  [   Pn   C
033 133 *** 103
```

Moves cursor right Pn columns. Cursor stops at right margin.

Cursor Backward (CUB)

```
ESC  [   Pn   D
033 133 *** 104
```

Moves cursor left Pn columns. Cursor stops at left margin.

Cursor Position (CUP)

```
ESC  [   Pl   ;   Pc   H
033 133 *** 073 *** 110
```

Moves cursor to line Pl, column Pc. If Pl or Pc are not selected or selected as 0, the cursor moves to first line or column, respectively. Origin mode (DECOM) selects line numbering and ability to move cursor into margins.

NOTE

> Pl and Pc represent variable numeric parameters. The parameter is a decimal number that represents one or more characters transmitted to the terminal as ASCII characters. Asterisks (***) represent the variable parameter in the octal sequence.
>
> CUP operates the same as the horizontal and vertical position (HVP) sequence.

Cursor Position (Home) (CUP)

```
ESC  [   H
033 133 110
```

Moves cursor to home position, selected by origin mode (DECOM).

Horizontal and Vertical Position (HVP)

```
ESC  [   Pl   ;   Pc   f
033 133 *** 073 *** 146
```

Moves cursor to line Pl, column Pc. If Pl or Pc are not selected or selected as 0, the cursor moves to first line or column, respectively. Origin mode (DECOM) selects line numbering and ability to move the cursor into margins.

NOTE

> HVP operates the same as the cursor position (CUP) sequence.

Horizontal and Vertical Position (Home) (HVP)

```
ESC  [   f
033 133 146
```

Cursor moves to home position selected by origin mode (DECOM).

Index

```
ESC  D  or   IND
033 104      204
```

Moves cursor down one line in same column. If cursor is at bottom margin, screen performs a scroll-up.

Reverse Index (RI)

```
ESC  M  or RI
033 115    215
```

Moves cursor up one line in same column. If cursor is at top margin, screen performs a scroll-down.

Next Line (NEL)

```
ESC  E  or NEL
033 105    205
```

Moves cursor to first position on next line. If cursor is at bottom margin, screen performs a scroll-up.

Save Cursor (DECSC)

```
ESC  7
033 067
```

Saves cursor position, character attribute (graphic rendition), character set, and origin mode selection. (See restore cursor.)

Restore Cursor (DECRC)

```
ESC  8
033 070
```

Restores previously saved cursor position, character attribute (graphic rendition), character set, and origin mode selection. If none were saved, the cursor moves to home position.

Columns Per Line - This mode selects the number of columns in a display line, 80 or 132. With either selection, the screen can display 24 lines. Select the number of columns per line by using the following sequences.

NOTE

> When you change the number of columns per line, the screen is erased. This also sets the scrolling region for full screen (24 lines).

Column Mode (DECCOLM)

```
ESC  [   ?   3   h
033 133 077 063 150
```

Set selects 132 columns per line.

```
ESC  [   ?   3   l
033 133 077 063 154
```

Reset selects 80 columns per line.

Auto Wrap - This mode selects where a received character will appear when the cursor is at the right margin. Select auto wrap by using the following sequences.

NOTE

> Regardless of the auto wrap Set-Up feature selection, the tab character never moves the cursor to the next line.

Auto Wrap Mode (DECAWM)

```
ESC  [   ?   7   h
033 133 077 067 150
```

Set selects auto wrap. Any display characters received when cursor is at right margin appear on next line. The display scrolls up if cursor is at end of scrolling region.

```
ESC  [   ?   7   l
033 133 077 067 154
```

Reset turns auto wrap off. Display characters received when cursor is at right margin replace previously displayed character.

Screen Background - This mode selects either light (reverse) or dark display background on the screen. Select screen mode by using the following sequences.

Screen Mode (DECSCNM)

```
ESC  [   ?   5   h
033 133 077 065 150
```

Set selects reverse screen, a white screen background with black characters.

```
ESC  [   ?   5   l
033 133 077 065 154
```

Reset selects normal screen, a black screen background with white characters.

Line Feed/New Line - This mode selects the control character(s) transmitted by **Return.** Line feed/new line also selects the action taken by the terminal when receiving line feed, form feed, and vertical tab. Table 15 provides a summary of the feature. Select line feed/new line mode by using the following sequences.

Line Feed/New Line Mode (LNM)

```
    ESC  [   2   0   h
    033 133 062 060 150
```

Set causes a received line feed, form feed, or vertical tab to move cursor to first column of next line. **Return** transmits both a carriage return and line feed. This selection is also called new line option.

```
    ESC  [   2   0   l
    033 133 062 060 154
```

Reset causes a received line feed, form feed, or vertical tab to move cursor to next line in current column. **Return** transmits a carriage return.

Table 15: Line Feed/New Line Feature

Feature Selection	**Key Pressed-Character Sent**	**Character Received-Function**
Off	**Return**-CR	CR-Cursor moves to left margin.
Off	**Line Feed**-LF	LF, FF, VT-Cursor moves to next line but stays in same column.
On	**Return**-CR LF	CR-cursor moves to left margin.
On	**Line Feed**-LF	LF, FF, VT-Cursor moves to left margin of next line.

Keyboard Action - Keyboard action lets the computer turn the keyboard on or off. This mode always resets when you enter Set-Up. Select keyboard action mode by using the following sequences.

Keyboard Action Mode (KAM)

```
    ESC  [   2   h
    033 133 062 150
```

Set turns off keyboard and turns on the Wait indicator.

```
    ESC  [   2   l
    033 133 062 154
```

Reset turns on keyboard and turns off the Wait indicator.

Auto Repeat - This mode selects automatic key repeating. A key pressed for more than one-half second automatically repeats the transmission of the character. Key repeating does not affect **Set-Up, Hold Screen,** and **Ctrl.** Select auto repeat mode by using the following sequences.

Auto Repeat Mode (DECARM)

```
ESC  [   ?   8   h
033 133 077 070 150
```

Set selects auto repeat. A key pressed for more than one-half second automatically repeats.

```
ESC  [   ?   8   l
033 133 077 070 154
```

Reset turns off auto repeat. Keys do not automatically repeat.

Local Echo (Keyboard Send-Receive) - This mode selects local echo, only in terminal mode, which causes every character transmitted by the Rainbow 100 computer to automatically appear on the screen. Therefore, the host computer does not have to transmit (echo) the character back to the Rainbow 100 computer for display. When local echo is off, the Rainbow 100 computer only transmits characters to the host computer. The host computer must echo the characters back to the Rainbow 100 computer for display. Select send-receive mode by using the following sequences.

Send-Receive Mode (SRM)

```
ESC  [   1   2   h
033 133 061 062 150
```

Set turns off local echo. The Rainbow 100 computer transmits characters to the host computer, which must echo characters for display on screen.

```
ESC  [   1   2   l
033 133 061 062 154
```

Reset selects local echo. Characters transmitted to the host computer automatically appear on the screen.

Cursor Key Character Selection - Cursor key mode selects the set of characters transmitted by the cursor keys. See Table 16 for the codes transmitted by the cursor keys. Select cursor key mode by using the following sequences.

NOTE

> If you power up or use a system reset command, cursor key mode resets. This mode also resets during a communication line connection in all communication except full-duplex no modem control (FDX A).

Cursor Key Mode (DECCKM)

```
ESC  [   ?   1   h
033 133 077 061 150
```

Set selects cursor keys to generate (application) functions.

```
ESC  [   ?   1   l
033 133 077 061 154
```

Reset selects cursor keys to generate cursor control sequences.

Table 16: ANSI Cursor Control Key Codes

Cursor Key	**Cursor Key Mode Reset Sends Cursor Control Sequence**	**Cursor Key Mode Set Generates Application Functions**
	ESC [A 033 133 101	ESC O A 033 117 101
	ESC [B 033 133 102	ESC O B 033 117 102
	ESC [C 033 133 103	ESC O C 033 117 103
	ESC [D 033 133 104	ESC O D 033 117 104

Keypad Character Selection - The numeric keypad generates either numeric characters or control functions. Selecting application or numeric keypad mode determines the type of characters. The program function (PF) keys generate the same characters regardless of the keypad character selection. See Table 17 for the characters generated by the keypad. Select the keypad mode by using the following sequences.

NOTE

> When you power up or use a system reset command, the terminal selects numeric keypad mode. This mode is also selected during communication line connections, except full-duplex no modem control (FDX A).

Application Keypad Mode (DECKPAM)

```
ESC  =
033 075
```

Selects application keypad mode. Keypad generates control functions.

Numeric Keypad Mode (DECKPNM)

```
     ESC  >
     033 076
```

Selects numeric keypad mode. Keypad generates characters that match the numeric, comma, period, and minus sign keys on main keyboard.

Table 17: ANSI Keypad Codes

Key	Numeric Keypad Mode	Application Keypad Mode
0	0 060	ESC O p 033 117 160
1	1 061	ESC O q 033 117 161
2	2 062	ESC O r 033 117 162
3	3 063	ESC O s 033 117 163
4	4 064	ESC O t 033 117 164
5	5 065	ESC O u 033 117 165
6	6 066	ESC O v 033 117 166
7	7 067	ESC O w 033 117 167
8	8 070	ESC O x 033 117 170
9	9 071	ESC O y 033 117 171
-(minus)	-(minus) 055	ESC O m 033 117 155
,(comma)	,(comma) 054	ESC O l 033 117 154
.(period)	.(period) 056	ESC O n 033 117 156

Table 17 (Cont.): ANSI Keypad Codes

Key	Numeric Keypad Mode	Application Keypad Mode
Enter*	**CR or CR LF** 015 015 012	**ESC O M** 033 117 115
PF1	**ESC O P** 033 117 120	**ESC O P** 033 117 120
PF2	**ESC O Q** 033 117 121	**ESC O Q** 033 117 121
PF3	**ESC O R** 033 117 122	**ESC O R** 033 117 122
PF4	**ESC O S** 033 117 122	**ESC O S** 033 117 123

NOTE

In ANSI mode, if the codes are echoed back to the terminal or if the terminals is off-line, the last character of the sequence appears on the screen; for example, PF4 appears as an "S".

*In numeric keypad mode, **Enter** generates the same characters as **Return.** You can change the **Return** key character code with the line feed/new line feature. When off, this feature causes the key to generate a single control character (CR, octal 015). When on, this feature causes the key to generate two characters (CR, octal 015 and LF, octal 012).

Character Sets and Selection - The Rainbow 100 computer can display the characters found in Tables 18 through 20.

The Rainbow 100 computer can select only one character set at a time. Therefore, the Rainbow 100 computer uses the following three character sets, with some characters appearing in more than one set. The GR displayable characters found in Table 8 are always available.

United States
United Kingdom
Special characters and line drawing (VT100 compatible)

Tables 18 through 20 show the character sets. The United States and United Kingdom character sets meet the standard of the "ISO international register of character sets to be used with escape sequences." The space (SP) and control characters are the same in all sets.

The terminal uses two active character sets at any one time. The computer designates these sets as G0 and G1, using the select character set (SCS) sequence. Then a single control character can switch between sets. Shift in (SI, octal 017) invokes the G0 character set; shift out (SO, octal 016) invokes the G1 character set.

The designated character sets are active until the terminal receives another SCS sequence. You can use the SCS sequence as often as needed to designate G0 and G1. Designate G0 by using the following sequences.

NOTE

> The terminal uses the character set selected in Set-Up after all communication line connections, except full-duplex no modem control (FDX A).

Select Character Set (SCS)

```
ESC (   A
033 050 101
```

Designates the UK character set as G0.

```
ESC (   B
033 050 102
```

Designates the US character set as G0.

```
ESC (   0
033 050 060
```

Designates the special characters and line drawing character set as G0.

Designate G1 by using the following sequences.

Select Character Set (SCS)

```
ESC )   A
033 051 101
```

Designates the UK character set as G1.

```
ESC )   B
033 051 102
```

Designates the US character set as G1.

```
ESC )   0
033 051 060
```

Designates the special characters and line drawing character set as G1.

The terminal also has G2 and G3 character sets. However, these are always the default (selected in Set-Up) character sets. You select G2 and G3 for only one character at a time. The terminal returns to the previous character set after displaying a single character. Select G2 and G3 for one character by using the following sequences.

Table 18: 7-bit US/UK ASCII Characters

	COLUMN	0		1		2		3		4		5		6		7	
	b8 BITS b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	\`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	* #/£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

NOTE: DEPENDS ON THE CHARACTER SET SELECTED; U.S.=# U.K.= £

|←CO CODES→|← GL CODES (ASCII GRAPHICS) →|

KEY

CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MR-9593

Table 19: 8-bit Control and Displayable Characters

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80		220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81		221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82		222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83		223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84		224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85		225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
	206 134 86		226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
	207 135 87		227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
	211 137 89		231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
	214 140 8C		234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D		235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E		236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F		237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←C1 CODES→|← GR CODES (DEC SUPPLEMENTAL GRAPHICS) →|

KEY

CHARACTER	Æ	306	OCTAL
		198	DECIMAL
		C6	HEX

MR-9594

Table 20: Special Characters and Line Drawing Character Set

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	
BITS		COLUMN 0		1		2		3		4		5		6		7	
B4 B3 B2 B1	ROW																
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	♦	140 96 60	⎺ SCAN 3	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	▒	141 97 61	─ SCAN 5	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	␉	142 98 62	─ SCAN 7	162 114 72
0 0 1 1	3	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	␌	143 99 63	⎽ SCAN 9	163 115 73
0 1 0 0	4	EOT	4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	␍	144 100 64	├	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	␊	145 101 65	┤	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	°	146 102 66	┴	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	␤	150 104 68	│	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	␋	151 105 69	≤	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	(BLANK)	137 95 5F	⎺ SCAN 1	157 111 6F		177 127 7F

KEY

ASCII CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MR-9587

Single Shift 2 (SS2)

```
ESC  N  or SS2
033 116    216
```

Selects G2 (default) character set for one character. You select G2 in Set-Up.

Single Shift 3 (SS3)

```
ESC  O  or SS3
033 117    217
```

Selects G3 (default) character set for one character. You select G3 in Set-Up.

Character Attributes - The terminal can display the following character attributes that change the character display without changing the character.

- Underline
- Reverse video (character background opposite of the screen background feature)
- Blink
- Bold (increased intensity)
- Any combination of these attributes (applied in the order of reception)

You can select one or more character attributes at one time. Selecting an attribute does not turn off other attributes already selected. After you select an attribute, all characters received by the terminal appear with that attribute. If you move the characters by scrolling, the attribute moves with the characters. Select the character attributes by using the following sequences.

Select Graphic Rendition (SGR)

```
ESC  [   m  or ESC  [   0   m
033 133 155    033 133 060 155
```

Turns off character attributes.

```
ESC  [   1   m
033 133 061 155
```

Selects bold (increased intensity).

```
ESC  [   4   m
033 133 064 155
```

Selects underline.

```
ESC  [   5   m
033 133 064 155
```

Selects blink.

```
ESC  [   7   m
033 133 065 155
```

Selects reverse video.

Tab Stops - You select tab stop positions on the horizontal lines of the screen. The cursor advances (tabs) to the next tab stop when the terminal receives a horizontal tab (HT, octal 011). If no tab stops are set, horizontal tab moves the cursor to the right margin. Set and clear the tab stops by using the following sequences.

Horizontal Tabulation Set (HTS)

```
ESC  H  or HTS
033 110    210
```

Sets a horizontal tab stop at cursor position.

Tabulation Clear (TBC)

```
ESC  [   g  or ESC  [   0   g
033 133 147    033 133 060 147
```

Clears a horizontal tab stop at cursor position.

```
ESC  [   3   g
033 133 063 147
```

Clears all horizontal tab stops.

Line Attributes - These are display features that affect a complete display line. The cursor selects the line affected by the attribute. The cursor stays in the same character position when the attribute changes. However, if the attribute would move the cursor past the right margin, the cursor stops at the right margin. When you move lines on the screen by scrolling, the attribute moves with the line. Select line attributes by using the following sequences.

NOTE

> If you erase an entire line by using the erase in display (ED) sequence, the line attribute changes to single-height and single-width.

Double-Height Line (DECDHL)

```
Top Half:     Bottom Half:
ESC  #   3    ESC  #   4
033 043 063   033 043 064
```

Makes the line with the cursor the top or bottom half of a double-height, double-width line. Sequences work in pairs on adjacent lines. The same character must be used on both lines to form full characters. If the line was single-width, single-height, all characters to the right of center are lost.

Single-Width Line (DECSWL)

```
ESC  #   5
033 043 065
```

Makes the line with the cursor single-width, single-height. This is line attribute for all new lines on screen.

Double-Width Line (DECDWL)

```
ESC  #   6
033 043 066
```

Makes the line with the cursor double-width, single-height. If the line was single-width, single-height, all characters to the right of center screen are lost.

Erasing - Erasing removes characters from the screen without affecting other characters on the screen. Erased characters are lost. The cursor position does not change when erasing characters or lines.

If you erase a line by using the erase in display (ED) sequence, the line attribute becomes single-height, single-width. If you erase a line by using the erase in line (EL) sequence, the line attribute is not affected.

Erasing a character also erases any character attribute of the character. Erase characters by using the following sequences.

Erase in Line (EL)

```
ESC  [   K  or ESC  [   0   K
033 133 113    033 133 060 113
```

Erases from cursor to end of line, including cursor position.

```
ESC  [   1   K
033 133 061 113
```

Erases from beginning of line to cursor, including cursor position.

```
ESC  [   2   K
033 133 062 113
```

Erases complete line.

Erase in Display (ED)

```
ESC  [   J  or ESC  [   0   J
033 133 112    033 133 060 112
```

Erases from cursor to end of screen, including cursor position.

```
ESC  [   1   J
033 133 061 112
```

Erases from beginning of screen to cursor, including cursor position.

```
ESC  [   2   J
033 133 062 112
```

Erases complete display. All lines are erased and changed to single-width. Cursor does not move.

Computer Editing - Editing allows the computer to insert or delete characters and lines of characters at the cursor position. The cursor position does not change when inserting or deleting lines. Delete characters or insert and delete lines by using the following sequences.

NOTE

> Insertion-replacement mode (RM) selects how characters are added to the screen. See Inserting and Replacing Characters in this chapter for more information.

Delete Character (DCH)

```
ESC  [  Pn   P
033 133 *** 120
```

Deletes Pn characters, starting with character at cursor position. When a character is deleted, all characters to the right of cursor move left. This creates a space character at right margin. This character has all attributes off.

Insert Line (IL)

```
ESC  [  Pn   L
033 133 *** 114
```

Inserts Pn lines at line with cursor. Lines displayed below cursor move down. Lines moved past the bottom margin are lost. This sequence is ignored when cursor is outside scrolling region.

Delete Line (DL)

```
ESC  [  Pn   M
033 133 *** 115
```

Deletes Pn lines starting at line with cursor. As lines are deleted, lines displayed below cursor move up. Lines added to bottom of screen have spaces with same character attributes as last line moved up. This sequence is ignored when cursor is outside scrolling region.

Inserting and Replacing Characters - The terminal displays received characters at the cursor position. This mode determines how the terminal adds characters to the screen. Insert mode displays the character and moves previously displayed characters to the right. Replace mode adds characters by replacing the character at the cursor position. Select insertion-replacement mode by using the following sequences.

NOTE

> This mode resets after a communication line connection in all communication except full-duplex no modem control (FDX A). It also resets any time NVM is saved.

Insertion-Replacement Mode (IRM)

```
ESC  [   4   h
033 133 064 150
```

Set selects insert mode and turns INSERT on. New display characters move old display characters to the right. Characters moved past the right margin are lost.

```
ESC  [   4   l
033 133 064 154
```

Reset selects replace mode and turns INSERT off. New display characters replace old display characters at cursor position. The old character is erased.

Printing in Terminal Mode - The Rainbow 100 computer has a serial printer interface for local printing. The host computer can select all print operations by using escape sequences. You can only select two of the print operations from the keyboard, auto print and Print Screen.

When you print characters from the screen, Rainbow 100 computer terminal and printer tab stops are ignored. Print characters are spaced with the space (SP, octal 040) character. The terminal transmits a carriage return (CR, octal 015) and line feed (LF, octal 012) after the last printable character of a line - but not a space character.

A line of double-height characters prints as two identical lines of single-width characters. Double-width characters print as single-width characters on a single line.

Before selecting a print operation, check the printer status by using the printer status report (DSR) in ANSI mode. Do not select a print operation if the serial printer is not ready to print. Select print operations by using the following sequences.

Media Copy (Auto Print ON) (MC)

```
ESC  [   ?   5   i
033 133 077 065 151
```

Turns on auto print. A display line prints after you move cursor off the line, using a line feed, form feed, or vertical tab (also transmitted to printer).

The line also prints during an auto wrap. Auto wrap lines end with a CR,LF.

Media Copy (Auto Print Off) (MC)

```
ESC  [   ?   4   i
033 133 077 064 151
```

Turns off auto print.

NOTE

> Printer controller has a higher priority than auto print. Therefore, you can select printer controller and print characters during auto print.

Media Copy (Printer Controller On) (MC)

```
ESC  [   5   i
033 133 065 151
```

Turns on printer controller. The terminal transmits received characters to printer without displaying them. The terminal does not insert or delete spaces, provide line delimiters, or select the correct printer character set.

Media Copy (Printer Controller Off) (MC)

```
ESC  [   4   i
033 133 064 151
```

Turns off printer controller. Always move printhead to left margin before turning off printer controller.

Media Copy (Print Cursor Line) (MC)

```
ESC  [   ?   1   i
033 133 077 161 151
```

Prints display line with cursor. Cursor position does not change. Print cursor line ends when line prints.

Media Copy (Print Screen) (MC)

```
ESC  [   i  or ESC  [   0   i
033 133 151    033 133 060 151
```

Prints the screen. Printer extent (DECEXT) selects full screen or scrolling region to print. Select scrolling region by using set top and bottom margins (DECSTBM) sequence. Print Screen ends when screen prints.

Printer Extent in Terminal Mode - This mode selects the full screen or the scrolling region to print during a Print Screen. Select printer extent mode by using the following sequences.

Printer Extent Mode (DECPEX)

```
ESC  [   ?   1   9   h
033 133 077 061 071 150
```

Set selects the full screen to print during a Print Screen.

```
ESC  [   ?   1   9   l
033 133 077 061 071 154
```

Reset selects the scrolling region to print during a Print Screen.

Print Termination Character in Terminal Mode - This mode determines if the terminal should transmit a print termination character after a Print Screen. The form feed (octal, 014) control character serves as the print termination character. Select printer form feed mode by using the following sequence.

```
ESC  [   ?   1   8   h
033 133 077 061 070 150
```

Set selects form feed as print termination character. The terminal transmits this character to printer after each Print Screen.

```
ESC  [   ?   1   8   l
033 133 077 061 070 154
```

Reset selects no termination character.

Reports - The Rainbow 100 computer transmits reports in response to escape sequence requests. Reports determine terminal emulation type and status, and cursor position. The report requests and responses are as follows.

NOTE

> The terminal does not respond to the DSR, DA, or DECID sequences during printer controller operation.

Device Status Report (DSR)

```
ESC  [   5   n
033 133 065 156
```

Computer requests a status report (using a DSR sequence).

```
ESC  [   0   n
033 133 060 156
```

Terminal response: Ready, no malfunctions detected.

These next four codes apply to terminal mode only.

```
ESC  [   ?   1   5   n
033 133 077 061 065 156
```

Computer requests a printer status report. Terminal checks status of printer. This report should be requested before any print operation.

```
ESC  [   ?   1   3   n
033 133 077 061 063 156
```

Printer not connected to terminal. Data terminal ready (DTR) signal of the printer has not been on since terminal turned on.

```
ESC  [   ?   1   1   n
033 133 077 061 061 156
```

Printer not ready to print. Printer DTR was on, but is now off.

```
ESC  [   ?   1   0   n
033 133 077 061 060 156
```

Printer ready to print. Printer DTR is on.

Cursor Position Report (CPR)

```
ESC  [   6   n
033 133 066 156
```

Requests a cursor position report.

```
ESC  [  Pl   ;  Pc   R
033 133 *** 073 *** 122
```

Terminal reports cursor position in response to DSR sequence request from computer. Pl indicates line and Pc indicates column. No parameters, or parameters of 0, indicate cursor is at home position. Origin mode (DECOM) determines whether line numbering is relative to the top of the screen or the top of the scrolling region.

Device Attributes (DA)

```
ESC  [   c  or ESC  [   0   c
033 133 143    033 133 060 143
```

A request for Rainbow 100 computer identification.

Identify Terminal (DECID)

```
ESC  Z
033 132
```

A request for Rainbow 100 computer identification. Rainbow 100 computer uses device attributes (DA) to respond. Future DIGITAL terminals may not support this sequence. Therefore, new software should use device attributes.

Device Attributes (DA)

```
ESC  [   ?   6   c
033 133 077 066 143
```

Rainbow 100 response: "I am a VT102."

Reset to Initial State(RIS)

```
ESC  c
033 143
```

Resets the terminal to its initial state.

CAUTION

> It is recommended that this not be used due to unpredictable results.

Adjustments - The terminal has a screen alignment pattern that lets Field Service personnel adjust the screen. Display the screen alignment pattern by using the following sequence.

Screen Alignment Display (DECALN)

```
ESC  #   8
033 043 070
```

Fills screen with uppercase E's for screen focus and alignment. This command is used by DIGITAL Manufacturing and Field Service personnel.

VT52-Compatible Sequences

VT52-compatible sequences meet private DIGITAL standards. Therefore, the terminal can use existing software designed for previous terminals (such as the VT52). You can select VT52 compatibility from the keyboard in Set-Up or the computer can use a sequence. (See ANSI-Compatible Sequences in this chapter).

NOTE

> In VT52 mode, 8-bit control characters and displayable characters are processed just as in ANSI mode.

Modes - In VT2 mode, you cannot select most terminal features by using sequences. You can, however, select the following three modes by using sequences: ANSI mode, application keypad mode on, and application keypad mode off (numeric keypad mode on).

ANSI/VT52 Compatibility - The terminal is compatible with both ANSI and private DIGITAL standards. Therefore, the terminal can use new software that meets ANSI standards and existing software designed for previous terminals (such as the VT52). ANSI-compatible sequences meet standards X3.64-1979 and X3.41-1974. You use ANSI mode to select most terminal features; the terminal uses the same features when it switches to VT52 mode. You cannot, however, change most of these features in VT52 mode. Select ANSI compatibility by using the following sequence.

ANSI Mode (DECANM)

```
ESC  <
033 074
```

The terminal interprets all sequences according to ANSI standards X3.64-1979 and X3.41-1974. The VT52 escape sequences described in this chapter are not recognized.

Cursor Positioning - The cursor indicates the active screen position where the next character will appear. You must select the margins for VT52 mode in ANSI mode. If you do not select margins, the terminal uses the complete screen. The cursor moves:

- One column to the right when a character appears
- One line down after a line feed, form feed, or vertical tab (Line feed/new line may also move the cursor to left margin.)
- To the left margin after a carriage return
- One column to the left after a backspace
- To the next tab stop (or right margin if no tabs are set) after a horizontal tab character.

You can also move the cursor by using the following sequences.

Cursor Up

```
ESC  A
033 101
```

Moves cursor up one line in same column. Cursor stops at top margin.

Cursor Down

```
ESC  B
033 102
```

Moves cursor down one line in same column. Cursor stops at bottom margin.

Cursor Right

```
ESC  C
033 103
```

Moves cursor one column to right. Cursor stops at right margin.

Cursor Left

```
ESC  D
033 104
```

Moves cursor one column to left. Cursor stops at left margin.

Cursor to Home

```
ESC  H
033 110
```

Moves cursor to home position.

Direct Cursor Address

```
ESC  Y  line column
033 131 ***  ***
```

Moves cursor to specified line and column. Line and column numbers are ASCII characters whose codes are their octal value plus octal 037. For example, line 1 column 8 parameters are octal 040 (first line) and octal 047 (eighth column).

Reverse Line Feed

```
ESC  I  or RI
033 111    215
```

Moves cursor up one line in same column. If cursor is at top margin, screen performs scroll-down.

Keypad Character Selection - The numeric keypad generates either numeric characters or control functions. Select application keypad mode to generate control functions. Exit application keypad mode (select numeric keypad mode) to generate numeric characters. See Table 21 for the characters generated by the keypad. Enter and exit application keypad mode by using the following sequences.

NOTE

> When you power up or use a system reset command, the terminal exits application keypad mode (selects numeric keypad mode). This mode is also selected during communication line connections, except full-duplex no modem control (FDX A).

Enter Application Keypad Mode

```
ESC  =
033 075
```

Keypad generates sequences used by the application program.

Exit Application Keypad Mode (Numeric Keypad Mode)

```
ESC  >
033 076
```

Keypad generates characters that match the numeric, comma, period, and minus sign keys on main keyboard.

Table 21: VT52 Keypad Codes

Key	Application Keypad Mode Off (Numeric Keypad Mode)	Application Keypad Mode On
0	0 060	ESC ? p 033 077 160
1	1 061	ESC ? q 033 077 161
2	2 062	ESC ? r 033 077 162
3	3 063	ESC ? s 033 077 163
4	4 064	ESC ? t 033 077 164
5	5 065	ESC ? u 033 077 165
6	6 066	ESC ? v 033 077 166
7	7 067	ESC ? w 033 077 167
8	8 070	ESC ? x 033 077 170
9	9 071	ESC ? y 033 077 171
-(minus)	-(minus) 055	ESC ? m 033 077 155*
,(comma)	,(comma) 054	ESC ? l 033 077 154*
.(period)	.(period) 056	ESC ? n 033 077 156

Table 21 (Cont.): VT52 Keypad Codes

Key	Application keypad Mode Off (Numeric Keypad Mode)	Application Keypad Mode On
Enter+	**CR or CR LF** 015 015 012	**ESC ? M** 033 077 115
PF1	**ESC P** 033 120	**ESC P** 033 120
PF2	**ESC Q** 033 121	**ESC Q** 033 121
PF3	ESC R 033 122	ESC R 033 122
PF4	**ESC S** 033 123	**ESC S** 033 123*

*These sequences are not generated by the VT52.

+In numeric keypad mode, (application keypad mode off), **Enter** generates the same characters as **Return.** You can change the **Return** key character code with the line feed/new line feature. When off, this feature causes the key to generate a single control character (CR, octal 015). When on, this feature causes the key to generate two characters (CR, octal 015 and LF, octal 012).

Character Sets and Selection - In VT52 mode, the terminal uses either the US/UK character set selected in Set-Up or the special characters and line drawing character set. Tables 18 and 19 show the United Kingdom and United States character sets. Table 20 shows the special characters and line drawing character set. Table 22 compares the special characters and line drawing character set to VT52 graphics mode (character set). Select the character sets by using the following sequences.

NOTE

> The character set selected in Set-Up is used after all communication line connections, except full-duplex no modem control (FDX A).

Enter Graphics Mode

```
ESC  F
033  106
```

Selects the special characters and line drawing character set.

Exit Graphics Mode

```
ESC  G
033  107
```

Selects the character set selected in Set-Up.

Table 22: Special Characters and Line Drawing Set and VT52 Graphics Mode Comparison

Octal Code	US or UK Set	Special Characters and Line Drawing Set	VT52 Graphics Mode (Not Available in Rainbow 100 Computer
137	-	Blank	Blank
140	/	Diamond	Reserved
141	a	Checkerboard (error indicator)	Solid rectangle
142	b	Horizontal tab	1/
143	c	Form feed	3/
144	d	Carriage return	5/
145	e	Line feed	7/
146	f	Degree symbol	Degrees
147	g	Plus/minus	Plus or minus
150	h	New line	Right arrow
151	i	Vertical tab	Ellipsis (dots)
152	j	Lower-right corner	Divide by
153	k	Upper-right corner	Down arrow
154	l	Upper-left corner	Bar at scan 0
155	m	Lower-left corner	Bar at scan 1
156	n	Crossing lines	Bar at scan 2
157	o	Horizontal line - scan 1	Bar at scan 3
160	p	Horizontal line - scan 3	Bar at scan 4
161	q	Horizontal line - scan 5	Bar at scan 5
162	r	Horizontal line - scan 7	Bar at scan 6
163	s	Horizontal line - scan 9	Bar at scan 7
164	t	Left "T"	Subscript 0
165	u	Right "T"	Subscript 1
166	v	Bottom "T"	Subscript 2
167	w	Top "T"	Subscript 3
170	x	Vertical bar	Subscript 4
171	y	Less than or equal to	Subscript 5
172	z	Greater than or equal to	Subscript 6
173	{	Pi	Subscript 7
174	|	Not equal to	Subscript 8
175	}	UK pound sign	Subscript 9
176	~	Centered dot	Paragraph

Erasing - Erasing removes characters from the screen. Erased characters are lost. Erase characters by using the following sequences.

Erase to End of Line

```
ESC  K
033 113
```

Erases all characters from cursor to end of current line, including cursor position. Cursor does not move.

Erase to end of Screen

```
    ESC  J
    033 112
```

Erases all characters from cursor to end of screen, including cursor to end of position. Cursor does not move.

Printing in Terminal Mode - The Rainbow 100 computer has a serial printer interface for local printing. The host computer can select all print operations by using sequences. You can only select two print operations from the keyboard, auto print and Print Screen.

When you print characters from the screen, terminal and printer tab stops are ignored. Characters printed are spaced with the space (SP, octal 040) character. The Rainbow 100 computer transmits a carriage return and line feed - but not a space character - after the last printable character of a line.

A line of double-height characters print as two identical lines of single-width characters. Double-width characters print as single-width characters on a single line.

Before selecting a print operation, check the printer status by using the printer status report (DSR) in ANSI mode. Do not select a print operation if the serial printer is not ready to print. Select print operations by using the following sequences.

Auto Print

```
    ESC  ^
    033 136
```

Turns on auto print. A display line prints after you move cursor off the line, using a line feed, form feed, or vertical tab (also transmitted to printer).

The line also prints during an auto wrap. Auto wrap lines end with CR, LF.

```
    ESC  _
    033 137
```

Turns off auto print.

NOTE

> Printer controller has a higher priority than auto print. Therefore, you can select printer controller and print characters during auto print.

Print Controller

ESC W
033 127

Turns on print controller. The terminal transmits received characters to printer without displaying them. The terminal does not insert or delete spaces, provide line delimiters, or select printer character set.

ESC X
033 130

Turns off printer controller. Always move printhead to left margin before turning off printer controller.

Print Cursor Line

ESC V
033 126

Prints display line with cursor. Cursor position does not change. Print cursor line ends when the line prints.

Print Screen

ESC]
033 135

Prints the screen. Printer extent (DECPEX) determines whether full screen or scrolling region prints. Select scrolling region by using DECSTBM sequence. Print Screen ends when screen prints.

Reports - The Rainbow 100 computer transmits reports in response to escape sequence report requests. The Rainbow 100 computer generates only one report in VT52 mode. The report requests and responses are as follows.

Identify

ESC Z
033 132

This escape sequence requests the Rainbow 100 computer to identify itself.

ESC / Z
033 057 132

Rainbow 100 computer responds "I am a VT52." (Same as VT52.)

APPENDIX A

PROGRAMMING SUMMARY

GENERAL

This appendix provides a summary of Rainbow 100 computer escape and control sequences.

Figure 4 shows the codes generated by the standard keys. Figure 5 shows the control codes generated by the function keys; shaded keys do not need **Ctrl** down to generate the control character.

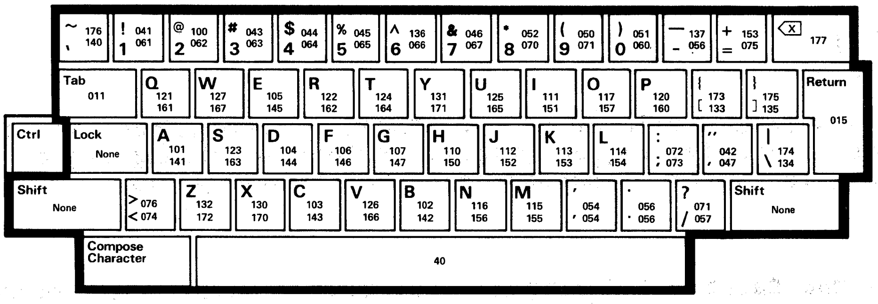


Figure 4: Standard Key Codes

Programming Sequences

The rest of this appendix repeats the information in summary form.

Control Characters Received

Name	Character Mnemonic	Octal Code	Function
Null	NUL	000	Ignored when received (not stored in input buffer) and used as a fill character.
End of transmission	EOT	004	Can be selected as a disconnect character. When used as a turnaround character, the disconnect character is DLE-EOT.

Name	Character Mnemonic	Octal Code	Function
Enquire	ENQ	005	Transmits answerback message.
Bell	BEL	007	Generates bell tone.
Backspace	BS	010	Moves cursor to the left one character position; if cursor is at left margin, no action occurs.
Horizontal	HT	011	Moves cursor to next tab stop, or to right margin if there are no more tab stops.
Line feed	LF	012	Causes a line feed or a new line operation (See line feed/new line mode.) Causes printing if in terminal mode and if auto print operation selected.
Vertical tab	VT	013	Processed as LF.
Form feed	FF	014	Processed as LF.
Carriage return	CR	015	Moves cursor to left margin on current line.
Shift out	SO	016	Selects G1 character set designated by a select character set sequence.
Shift in	SI	017	Selects G0 character set designated by a select character set sequence.
Device control 1	DC1	021	Processed as XON. DC1 causes terminal to continue transmitting characters. (Terminal mode only).
Device control 3	DC3	023	Processed as XOFF. DC3 causes terminal to stop transmitting all characters except XOFF and XON. (Terminal mode only).
Cancel	CAN	030	If received during an escape or control sequence, cancels the sequence and displays substitution character( ).
Substitute	SUB	032	Processed as CAN.
Escape	ESC	033	Processed as a sequence introducer.
Index	IND	204	Processes a line feed.
Next line	NEL	205	Processes as a CR LF sequence.
Horizontal tab set	HTS	210	Sets a horizontal tab at cursor location.
Reverse index	RI	215	Equals a reverse line feed.

Name	Character Mnemonic	Octal Code	Function
Single shift 2	SS2	216	Selects G2 character set for the next character only.
Single shift 3	SS3	217	Selects G3 character set for the next character only.
Control sequence introducer	CSI	233	Equals an ESC [.

ANSI Compatible Sequences

Set Mode

Name	Mnemonic	Mode	Sequence
Keyboard action	KAM	Locked	ESC [2 h
Insertion-replacement	IRM	Insert	ESC [4 h
Send-receive	SRM	Off	ESC [1 2 h **
Line feed/new line	LMN	New line	ESC [2 0 h
Cursor key	DECCKM	Application	ESC [? 1 h
ANSI/VT52	DECANM	ANSI	N/A
Column	DECCOLM	132 column	ESC [? 3 h
Scrolling	DECSCLM	Smooth	ESC [? 4 h
Screen	DECSCNM	Reverse	ESC [? 5 h
Origin	DECOM	Relative	ESC [? 6 h
Auto Wrap	DECAWM	On	ESC [? 7 h
Auto repeat	DECARM	On	ESC [? 8 h
Print form feed	DECPFF	On	ESC [? 1 8 h**
Print extent	EDCPEX	Full screen	ESC [? 1 9 h**

Reset Mode

Name	Mnemonic	Mode	Sequence*
Keyboard Action	KAM	Unlocked	ESC [2 l
Insertion-replacement	IRM	Replace	ESC [4 l
Send-receive	SRM	On	ESC [1 2 l**
Line feed/new line	LMN	Line feed	ESC [2 0 l
Cursor Key	DECCKM	Cursor	ESC [? 1 l
ANSI/VT52	DECANM	VT52	ESC [? 2 l
Column	DECCOLM	80 column	ESC [? 3 l
Scrolling	DECSCLM	Jump	ESC [? 4 l
Screen	DECSCNM	Normal	ESC [5 ? l
Origin	DECOM	Absolute	ESC [? 6 l
Auto wrap	DECAWM	Off	ESC [? 7 l
Auto repeat	DECARM	Off	ESC [? 8 l
Print form feed	DECPFF	Off	ESC [? 1 8 l**
Print extent	DECPEX	Scrolling region	ESC [? 1 9 l**

*The last character of the sequence is lowercase L(154 octal)
**Terminal mode only.

Cursor Key Codes Generated

	ANSI Characters Generated	
Cursor Key	**Reset**	**Set**
(Arrow)	**(Cursor)**	**(Application)**
Up	ESC [A	ESC O A
Down	ESC [B	ESC O B
Right	ESC [C	ESC O C
Left	ESC [D	ESC O D

Keypad Character Selection

Name	Mnemonic	Sequence
Alternate	DECKPAM	ESC =
Numeric	DECKPNM	ESC >

Keypad Codes Generated

Key	VT52 Numeric Keypad Mode	VT52 Alternate Keypad Mode	ANSI Numeric Keypad Mode	ANSI Alternate Keypad Mode
0	0	ESC ? p	0	ESC O p
1	1	ESC ? q	1	ESC O q
2	2	ESC ? r	2	ESC O r
3	3	ESC ? s	3	ESC O s
4	4	ESC ? t	4	ESC O t
5	5	ESC ? u	5	ESC O u
6	6	ESC ? v	6	ESC O v
7	7	ESC ? w	7	ESC O w
8	8	ESC ? x	8	ESC O x
9	9	ESC ? y	9	ESC O y
-(minus)	-(minus)	ESC ? m	-(minus)	ESC O m
,(comma)	,(comma)	ESC ? l*	,(comma)	ESC O l*
.(period)	.(period)	ESC ? N	.(period)	ESC O n
ENTER	Same as RETURN	ESC ? M	Same as RETURN	ESC O M
PF1	ESC P	ESC P	ESC O P	ESC O P
PF2	ESC Q	ESC Q	ESC O Q	ESC O Q
PF3	ESC R	ESC R	ESC O R	ESC O R
PF4	ESC S	ESC S	ESC O S	ESC O S

*The last character of the sequence is lowercase L (154 octal)

Select Character Sets SCS

Character Set	G0 Designator	G1 Designator
United Kingdom (UK)	ESC (A	ESC) A
United States (USASCII)	ESC (B	ESC) B
Special characters and line drawing set	ESC (0	ESC) 0

Name	Mnemonic	Sequence
Single shift 2	SS2	ESC N
Single shift 3	SS3	ESC O

Character Attributes

Name	Mnemonic	Sequence
Select graphic rendition (no attributes)	SGR	ESC [m
Select graphic rendition (no attributes)	SGR	EC [0 m
Select graphic rendition (select attribute bold)	SGR	ESC [1 m
Select graphic rendition (select attribute underline)	SGR	ESC [4 m
Select graphic rendition (select attribute blink)	SGR	ESC [5 m
Select graphic rendition (select attribute, reverse video)	SGR	ESC [7 m

Scrolling Region

Name	Mnemonic	Sequence
Cursor up	CUU	ESC [Pn A
Cursor down	CUD	ESC [Pn B
Cursor forward (right)	CUF	ESC [Pn C
Cursor backward (left)	CUB	ESc [Pn D
Cursor position	CUP	ESC [Pl; Pc H
Cursor position (home)	CUP	ESC [H
Horizontal and vertical position	HVP	ESC [Pl; Pc f
Horizontal and vertical position (home)	HVP	ESC [f
Index	IND	ESC D
Reverse index	RI	ESC M
Next line	NEL	ESC E
Save cursor (and attributes)	DECSC	ESC 7
Restore cursor (and attributes)	DECRC	ESC 8

Tab Stops

Name	Mnemonic	Sequence
Horizontal tab set (at current column)	HTS	ESC H
Tabulation clear (at current column)	TBC	ESC [g
Tabulation clear (at current column)	TBC	ESC [0 g
Tabulation clear (all tabs)	TBC	ESC [3 g

Line Attributes

Name	Mnemonic	Sequence
Double-height top half	DECDHL	ESC # 3
Double-height bottom half	ECDHL	ESC # 4
Single-width single-height	DECSWL	ESC # 5
Double-width single-height	DECDWL	ESC # 6

Erasing

Name	Mnemonic	Sequence
Erase in line (cursor to end of line)	EL	ESC [K
Erase in line (cursor to end of line)	EL	ESC [0 K
Erase in line (beginning of line to cursor)	EL	ESC [1 K
Erase in line (entire line containing cursor)	EL	ESC [2 K
Erase in display (cursor to end of screen)	ED	ESC [J
Erase in display (cursor to end of screen)	ED	ESC [0 J
Erase in display (beginning of screen to cursor)	ED	ESC [1 J
Erase in display (entire screen)	ED	ESC [2 J

Editing Functions

Name	Mnemonic	Sequence
Delete character	DCH	ESC [Pn P
Insert line	IL	ESC [Pn L
Delete line	DL	ESC [Pn M

Print Commands for Terminal Mode

Name	Mnemonic	Sequence
Media copy (enter auto print)	MC	ESC [? 5 i
Media copy (exit auto print)	MC	ESC [? 4 i
Media copy (enter printer controller)	MC	ESC [5 i
Media copy (exit printer controller)	MC	ESC [4 i
Media copy (Print Screen)	MC	ESC [i
Media copy (Print Screen)	MC	ESC [0 i
Media copy (print cursor line)	MC	ESC [? 1 i

Reports

Name	Mnemonic	Sequence
Device status report (request status of VT102)	DSR	ESC [5 n
Response:		
Terminal OK	DSR	ESC [0 n
Device status report (request status of printer)	DSR	ESc [? 1 5 n
Response:		
Printer ready	DSR	ESC [? 1 0 n*
Printer not ready	DSR	ESC [? 1 1 n*
No printer	DSR	ESC [? 1 3 n*
Device status report (report cursor position)	DSR	ESC [6 n
Cursor position report	CPR	ESC [Pl; Pc R
Device attributes (what are you)	DA	ESC [c
Device attributes (what are you)	DA	ESC [0 c
Identify terminal (what are you)	DECID	ESC Z

NOTE

ESC Z is not recommended.

Device attributes response: VT102	DA	ESC [? 6 c

NOTE

ESC c is not recommended.

*Terminal mode only.

Reset

Name	Mnemonic	Sequence
Reset to initial state	RIS	ESC c

CAUTION

Do not use, unpredictable results.

Tests and Adjustments

Name	Mnemonic	Sequence
Screen alignment display (fill screen with "Es")	DECALN	ESC # 8

VT52 Compatible Mode

Modes	Sequence
Enter ANSI mode	ESC <

Keypad Character Selection

Name	Sequence
Enter alternate keypad mode	ESC =
Exit alternate keypad mode (Numeric keypad mode)	ESC >

NOTE

VT52 alternate keypad and numeric keypad mode different than ANSI.

Character Sets

Name	Sequence
Special graphics character set	ESC F*
Select US/UK character set (as determined by US/UK character Set-Up feature)	ESC G

*Same as special character and line drawing set in ANSI mode.

Cursor Position

Name	Sequence
Cursor up*	ESC A
Cursor down*	ESC B
Cursor right*	ESC C
Cursor left*	ESC D
Cursor to home	ESC H
Direct cursor address	ESC Y pl Pc**
Reverse line feed	ESC I***

*Same when sent from the terminal.

**Line and column numbers for direct cursor address are single character codes whose values are the desired number plus 37 octal. Line and column number start at one.

***The last character of the sequence is an uppercase I (111 octal).

Erasing

Name	Sequence
Erase to end of line	ESC K
Erase to end of screen	ESC J

Print Commands for Terminal Mode

Name	Sequence
Enter auto print mode	ESC ^
Exit auto print mode	ESC -
Enter printer controller mode	ESC W
Exit printer controller mode	ESC X
Print Screen	ESC]
Print cursor line	ESC V

Reports

Name	Sequence
Identify (what are you)	ESC Z
Response: VT102 (same as VT52)	ESC / Z

APPENDIX B

CONTROL FUNCTIONS (SEQUENCE FORMATS)

GENERAL

This appendix summarizes the ANSI code extension techniques defined in standards X3.41-1974 and X3.64-1979. Those specifications cover many special cases and details not included here.

Control Functions

The ANSI standards define types of characters used for specific purposes. You can determine a character's type by its position in the ASCII table (Table 22). There are two general categories of characters:

- display (columns 2 through 7; 10-15)
- control (columns 0 and 1; 8 and 9)

This table and the ANSI system can work for either a 7-bit or 8-bit character environment. The Rainbow 100 computer uses both 7-bit and 8-bit characters.

NOTE

> The ASCII 7-bit table corresponds to International Standards Organization (ISO) standard 646 and International Telegraph and Telephone Consultive Committee (CCITT) alphabet 5.

All control characters and groups of characters (sequences) not intended for display on the screen are control functions. Not all control functions perform an action in every ANSI device, but each device can recognize all control functions and discard any that do not apply to it. Therefore, each device performs a subset of the ANSI functions.

Because different devices use different subsets, compliance with ANSI does not mean compatibility between devices. Compliance only means that a particular function, if defined in the ANSI standard, is invoked by the same control function in all devices. If an ANSI device does not perform an action that has a control function defined in the ANSI standard, it cannot use that control function for any other purpose.

Table 23: US/UK ASCII Characters

	COLUMN	0		1		2		3		4		5		6		7	
ROW	b8 BITS b7 b6 b5 b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	• #/£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

NOTE: DEPENDS ON THE CHARACTER SET SELECTED; U.S.=# U.K.= £

|←— C0 CODES —→|←—————— GL CODES (ASCII GRAPHICS) ——————→|

KEY

CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MR-9593

Table 24: Control and Displayable Characters

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80		220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81		221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82		222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83		223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84		224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85		225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
	206 134 86		226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
	207 135 87		227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
	211 137 89		231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
	214 140 8C		234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D		235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E		236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F		237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←— C1 CODES —→|←— GR CODES (DEC SUPPLEMENTAL GRAPHICS) —→|

KEY

CHARACTER		
Æ	306	OCTAL
	198	DECIMAL
	C6	HEX

MR-9594

Table 25: Special Characters and Line Drawing Set

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	
BITS		COLUMN **0**		**1**		**2**		**3**		**4**		**5**		**6**		**7**	
B4 B3 B2 B1	ROW																
0 0 0 0	0	**NUL**	0 0 0		20 16 10	**SP**	40 32 20	**0**	60 48 30	**@**	100 64 40	**P**	120 80 50	◆	140 96 60	⎺ SCAN 3	160 112 70
0 0 0 1	1		1 1 1	**DC1** (XON)	21 17 11	**!**	41 33 21	**1**	61 49 31	**A**	101 65 41	**Q**	121 81 51	▒	141 97 61	⎻ SCAN 5	161 113 71
0 0 1 0	2		2 2 2		22 18 12	**"**	42 34 22	**2**	62 50 32	**B**	102 66 42	**R**	122 82 52	␉	142 98 62	─ SCAN 7	162 114 72
0 0 1 1	3	**ETX**	3 3 3	**DC3** (XOFF)	23 19 13	**#**	43 35 23	**3**	63 51 33	**C**	103 67 43	**S**	123 83 53	␌	143 99 63	⎼ SCAN 9	163 115 73
0 1 0 0	4	**EOT**	4 4 4		24 20 14	**$**	44 36 24	**4**	64 52 34	**D**	104 68 44	**T**	124 84 54	␍	144 100 64	├	164 116 74
0 1 0 1	5	**ENQ**	5 5 5		25 21 15	**%**	45 37 25	**5**	65 53 35	**E**	105 69 45	**U**	125 85 55	␊	145 101 65	┤	165 117 75
0 1 1 0	6		6 6 6		26 22 16	**&**	46 38 26	**6**	66 54 36	**F**	106 70 46	**V**	126 86 56	°	146 102 66	┴	166 118 76
0 1 1 1	7	**BEL**	7 7 7		27 23 17	**'**	47 39 27	**7**	67 55 37	**G**	107 71 47	**W**	127 87 57	±	147 103 67	┬	167 119 77
1 0 0 0	8	**BS**	10 8 8	**CAN**	30 24 18	**(**	50 40 28	**8**	70 56 38	**H**	110 72 48	**X**	130 88 58	␤	150 104 68	│	170 120 78
1 0 0 1	9	**HT**	11 9 9		31 25 19	**)**	51 41 29	**9**	71 57 39	**I**	111 73 49	**Y**	131 89 59	␋	151 105 69	≤	171 121 79
1 0 1 0	10	**LF**	12 10 A	**SUB**	32 26 1A	*****	52 42 2A	**:**	72 58 3A	**J**	112 74 4A	**Z**	132 90 5A	┘	152 106 6A	≥	172 122 7A
1 0 1 1	11	**VT**	13 11 B	**ESC**	33 27 1B	**+**	53 43 2B	**;**	73 59 3B	**K**	113 75 4B	**[**	133 91 5B	┐	153 107 6B	π	173 123 7B
1 1 0 0	12	**FF**	14 12 C		34 28 1C	**,**	54 44 2C	**<**	74 60 3C	**L**	114 76 4C	****	134 92 5C	┌	154 108 6C	≠	174 124 7C
1 1 0 1	13	**CR**	15 13 D		35 29 1D	**-**	55 45 2D	**=**	75 61 3D	**M**	115 77 4D	**]**	135 93 5D	└	155 109 6D	£	175 125 7D
1 1 1 0	14	**SO**	16 14 E		36 30 1E	**.**	56 46 2E	**>**	76 62 3E	**N**	116 78 4E	**^**	136 94 5E	┼	156 110 6E	·	176 126 7E
1 1 1 1	15	**SI**	17 15 F		37 31 1F	**/**	57 47 2F	**?**	77 63 3F	**O**	117 79 4F	(BLANK)	137 95 5F	⎽ SCAN 1	157 111 6F		177 127 7F

KEY

ASCII CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MR-9587

Escape and Control Sequences

Escape and control sequences provide more controls in addition to the control characters in the ASCII 7-bit table. These multiple-character control sequences are not displayed but control the displaying, processing, and transmission of characters. At the end of a sequence or during an error condition, the terminal continues to display received characters.

Escape Sequences

The format for an escape sequence is as follows:

ESC	**1....1**	**F**
033	040-057	060-176
Escape sequence introducer	Intermediate characters (0 or more characters)	Final character (1 character)

Escape Sequence Introducer - This is the ESC character (octal 033) defined by ANSI X3.4-1977. After receiving ESC, the terminal stores (but does not display) all control function characters received in the proper range.

Intermediate Characters - These are characters received after ESC in the octal range of 040 - 057 (column 2 of the ASCII table). The terminal stores intermediate characters as part of the control function.

Final Character - This is a character received after ESC in the octal range of 060 - 176 (columns 3 - 7 of the ASCII table). The final character indicates the end of the control function. The intermediate and final characters together define the function of the sequence. The terminal then performs the specified function and continues to display received characters. ANSI standard control functions have a final character in the octal range of 100 - 176 (columns 4 - 7 of the ASCII table). Private sequences have a final character in the octal range of 060 - 077 (column 3 of the ASCII table).

Example

Action: Designate ASCII character set as G0.

Sequence

ESC (B
033 050 102

Escape sequence Intermediate Final character
introducer character

Control Sequence Format

The format of a control sequence is as follows:

CSI	**P.....P**	**1.....1**	**F**
033 133	060-077	040-057	100-176
Control sequence introducer	Parameter characters (0 or more characters)	Immediate characters (0 or more (characters)	Final character (1 character)

Control Sequence Introducer - The CSI is the ESC (octal 033) and [(octal 133) characters defined by ANSI X3.41-1977. These characters provide 8-bit control functions by using 7-bit characters. After receiving CSI characters, the Rainbow 100 computer stores (but does not display) all control function characters received in the proper range.

During an escape sequence, if the Rainbow 100 computer receives an 8-bit control character, (octal 200-237), the escape sequence continues after. The 8-bit control character's function is executed if it is one of the supported functions.

An 8-bit display character, received during an escape sequence, has the 8th bit stripped off and the escape sequence continues with the resultant 7-bit remainder.

Parameter Characters - These are characters received after the CSI character, in the octal range of 060 -077 (column 3 of the ASCII table). The parameter characters modify the action or interpretation of the control function. The terminal interprets parameter characters as private when the < = > ? characters (octal 074 - 077) begin the parameter string. The : character (octal 072) is reserved. This means an ANSI-specified control sequence can have a parameter function with a private interpretation.

The Rainbow 100 computer uses two types of parameter characters, numeric and selective. A numeric parameter represents a decimal number, designated by Pn. The decimal characters have a range of 0 -9 (octal 060 - 071). A selective parameter comes from a list of specified parameters, designated by Ps.

If a control sequence includes more than one parameter, the parameters are separated by a delimiter, the ; character (octal 073).

Intermediate Characters - These are characters received after the CSI character, in the octal range of 040 -057 (column 2 of the ASCII table). The terminal stores these characters as part of the control function.

NOTE

The terminal does not use intermediate characters in control functions.

Final Character - This is a character received after the CSI character, in the octal range of 100 - 176 (columns 4 - 7 of the ASCII table). The final character indicates the end of the control function. The intermediate and final characters together define the function of the sequence. The terminal then performs the specified function and continues to display received characters. ANSI standard control functions have a final character in the octal range of 100 - 157 (columns 4 - 6 of the ASCII table). Private sequences have a final character in the octal range of 160 - 176 (column 7 of the ASCII table).

Example

Action: Clear all horizontal tabs.

Sequence

```
ESC  [   3   g
033 133 063 147
Control       Parameter       Final
sequence      character       character
introducer
```

Sequence Examples

These examples show the use of multiple functions selected in one sequence, private parameters and private sequences.

```
ESC  [   ?   4   h       Set smooth scroll mode
033 133 077 064 150      (? = ANSI private parameter)

ESC  [   2   ;   1   y   Invoke self-test
033 133 062 073 061 171  (y = ANSI private sequence)
```

APPENDIX C

RAINBOW 100 COMPUTER AND VT100 TERMINAL FAMILY DIFFERENCES

The following is a list of the differences between the Rainbow 100 and members of the VT100 family of terminals. Also included are certain "points-of-interest" that should be considered by programmers.

DEC'S MULTINATIONAL 8-BIT CHARACTER

The Rainbow 100 computer implements the printing graphics found in DEC's Multinational Character set, and the 8-bit character codes for the printing characters. It is a subset of the Multinational Character set. It is not the full character set. In particular, it does not implement all the control sequences specified for the Multinational Character set.

When a keyboard other than the US LN201-AA is selected the Rainbow 100 computer can also implement the 7-bit National Replacement character set for that country. The active character set is selected via SET-UP.

MULTINATIONAL 8-BIT CHARACTER CODES

DIGITAL has extended standard ASCII coding to introduce the DEC Multinational Character set. This set gives a uniform coding for all characters used in most European languages. It also adds characters such as ¢, ©, 1/4, and 1/2. The full set is shown in Appendix F. To represent these extra characters, DIGITAL uses 8 bits (standard ASCII uses 7 bits). DIGITAL's Multinational Character set is an extension of, and fully compatible with, the ASCII codes generally used in North America.

European countries (and French Canada) also define modifications to ASCII codes that replace certain ASCII characters with some of the extra characters in use in their language. These codes are known as National Replacement Character (NRC) codes. They use 7 bits to code each character. These code sets can differ from the ASCII codes in up to twelve positions. They are listed in Appendix F.

The Rainbow computer allows you to choose either DIGITAL's 8-bit Multinational Character set, or the 7-bit National Replacement Character set associated with your keyboard.

8-BIT CHARACTER CODES

The Rainbow 100 computer accepts and acts on 8-bit character codes, the VT102 terminal always strips the 8th bit. If 8-bit codes are received in VT52 mode, they will be handled the same as in ANSI mode.

C1 CONTROL CODES

Rainbow 100 computer executes (8-bit) control codes for index, next line, horizontal tab set, reverse index, single shift 2, single shift 3, control sequence introducer. Reception of any C1 control code will cause the appropriate action and not abort an escape sequence in process (CSI restarts an escape sequence).

KEYBOARD AND 8-BIT KEY CODES

The Rainbow 100B keyboard is called the LK201 keyboard. It is available in 15 different international varieties (See Appendix D). Each keyboard can generate every character in the Multinational character set via the compose mechanism.

COMPOSE CHARACTER SEQUENCES

Compose character sequences allow you to create and display accented characters, ligatures, numerical fractions, and other special characters that may not be on your keyboard. There are two forms of compose sequences that you can use to create and display a character.

- Two-key compose sequence
- Three-key compose sequence

TWO-KEY COMPOSE SEQUENCE

A two-key compose sequence uses two keys to create a different character. For example, on the British keyboard, typing the acute accent key, then typing **E** results in a new character: an E with an acute accent.

Note that you do not use the **Compose Character** key to create two-key compose sequences. The only keys for which you can use the two-key compose sequence are:

- Diaeresis/umlaut mark ..
- Acute accent '
- Grave accent
- Circumflex ^
- Tilde ~

You must type the character key before the letter key in two-key compose sequences.

NOTE

> The two-key compose sequence cannot be used on the North American (English) keyboard.

THREE-KEY COMPOSE SEQUENCE

You can use the three-key compose sequence on any keyboard. To start a compose sequence, you press the **Compose Character** key; then, you press the next two characters of the compose sequence. (See Figure E-1 in Appendix E for the compose sequences and resulting characters.) For example, if you press the **Compose Character** key, the acute accent key, and the letter **E** in sequence, the screen displays an E with an acute accent.

The order in which you type the characters is important. Certain sequences have an obvious order - the AE ligature and 1/4 fraction, for example. In these examples, the order may not be reversed. When you compose the E with an acute accent, you may type either the **E** or the acute accent after you press the **Compose Character** key.

Use the ⌫ key if you want to cancel a compose sequence.

KEYBOARD CONTROL CODE GENERATION

The Rainbow 100 can be used with one of 15 different national LK201 keyboards. Since not all of the keyboards have all the characters used by the VT102 to generate control codes all the keyboards can generate the NULL, ESC, FS, GS, RS, US and DEL codes by using the Control key in conjunction with keys 2-8 as described in table 3. The numeric keys are always used in the unshifted position, even on those keyboards where the numeral is in the shifted position.

SET-UP PURGING KEYBOARD BUFFER

When the Set-Up key is pressed to enter Set-Up mode in the Rainbow 100, the key-holding buffer is cleared which causes any unserviced keys to be lost and Set-Up is immediately honored.

WAIT INDICATOR

When the keyboard buffer fills up, the Rainbow 100 computer ignores further entries and sounds the bell. It lights the Wait LED as the VT102 terminal does. The bell and Wait LED are used to notify the user that the key was not accepted.

KEYBOARD PRINT SCREEN KEY IN TERMINAL MODE

Terminal mode print functions are implemented via the Print Screen key on the Rainbow 100 computer. VT102 terminal uses the keypad Enter key. Rainbow 100B Print Screen is equivalent to VT102's <Shift/Enter> and Rainbow 100B <Ctrl/Print Screen> is equivalent to VT102's <Ctrl/Enter>.

KEYBOARD HOLD SCREEN KEY

The Hold Screen key on the Rainbow 100 computer does not work the same as the NO SCROLL key on a VT102 terminal. On a VT102 terminal it sends an XOFF/XON as it toggles back and forth, the <Ctrl/S> and <Ctrl/Q> typed from the keyboard can be used to get the same effect. Setting Hold Screen for the Rainbow 100 computer does not necessarily cause an XOFF to be sent. It sets an internal flag that causes the "receive character" process to loop until the flag is cleared. This effectively "hangs" any console output (normal or direct) in console mode. In terminal mode this "hang" causes the comm receive buffer to fill up until it reaches the high water mark at which point it will send an XOFF, if enabled by Set-Up. After the Hold Screen is removed, characters are removed from the receive buffer until the low water mark is reached which causes XON to be sent, if enabled.

As a result of this method of implementation, Rainbow 100 computer honors Hold Screen even in "local", VT102 terminal does not.

In Rainbow 100 terminal mode, after using Hold Screen on incoming data, the last char for display is being 'held'. Entering setup, switching to local, and exiting from setup does not clear the 'hold' state or the char. When 'hold' is finally removed, the char originally being 'held' is displayed before any locally generated characters.

In the Rainbow 100 computer, if the Print Screen key is depressed while the Hold Screen is asserted, the print is deferred until after the 'hold' is removed and char being 'held' is processed. The VT102 terminal prints a screen even if the NO SCROLL key as been depressed.

KEYBOARD CURSOR KEY MODES

Cursor key mode and keypad mode for the Rainbow 100 computer are independent. In the VT102 terminal, the cursor keys only send application codes if both cursor and keypad modes are set to 'application'.

PRINTER CHARACTER SETS IN TERMINAL MODE

The Rainbow 100 computer assumes the printer is capable of properly receiving 8-bit DEC Multinational characters.

PRINTING BLOB CHARACTERS IN TERMINAL MODE

When printing from the screen in terminal mode and encountering a 'blob' character, the VT102 terminal sends ASCII 'SUB' to the printer. The Rainbow 100 computer sends the VT100 line-drawing graphics character 'blob' bracketed by the appropriate character set selection escape sequence if required.

PRINTER PORT DEFAULTS

Factory Set-Up defaults are not the same as the VT102 terminal for the printer port.

PRINT CURSOR LINE OPERATION IN TERMINAL MODE

At the completion of a 'print cursor line' operation, Rainbow 100 computer sends the escape string to restore the printers G0 char set in between the terminating carriage return and line feed. VT102 terminal sends it after the line feed.

PRINTER PORT STATUS REQUEST IN TERMINAL MODE

The following anomaly occurs when a printer cable is attached to a Rainbow after it is powered up, but the printer end of the cable is not attached to anything.

```
A printer status request is made to Rainbow :  ESC [ ?  n
Rainbow responds:       Printer not ready   :  ESC [ 11 n
                        or No printer       :  ESC [ 13 n
VT102 responds:         No printer          :  ESC [ 13 n
```

TERMINAL ID

The Rainbow 100 computer identifies itself as a VT102 terminal.

INSERT AND DELETE LINE ESCAPE SEQUENCES

```
Insert Line:   CSI  Pn  L
Default line:  CSI  Pn  M
```

If the cursor is on the last line of the scrolling regions, and the line is double-width, then after execution of either of the above controls, the active line attributes are:

A0 On Rainbow 100: single-width

B0 On the VT102 terminal: whatever the active line attributes were before execution of the control sequence.

ALTERNATE ROM CHARACTER SETS

Rainbow 100 computer does not implement the alternate ROM character sets found in the VT102 terminal.

ALTERNATE ROM AND LED ESCAPE SEQUENCES

The Rainbow 100 computer parses but ignores the escape sequences to set G0 and G1 to the alternate ROM and alternate ROM special graphics (ESC (1 , ESC (2 , ESC) 1 , ESC) 2). It will parse but ignore the escape sequence for LED control (ESC [Pn q). Rainbow has no alternate ROMs and the LEDs are not available for software control.

G2 AND G3 CHARACTER SETS

G2 and G3 are permanently designated as the NVM default character set. They are either US or UK variations of Rainbow's subset of the multinational character set. They can be invoked for single characters by the single-shift-2 or single-shift-3 escape sequences or the C1 control codes.

ABORTING ESCAPE SEQUENCES BY INTERMEDIATE CHARACTERS

Rainbow 100 computer aborts escape sequence parsing when it finds an intermediate char causing all following characters to be displayed. VT102 terminal aborts the sequence but continues parsing until it finds a final char so the intervening part of the escape sequence does not display.

INSERT AND REPLACE MODES

Rainbow 100 computer always sets insertion/replacement mode to replacement before saving into NVM. These modes are not user selectable. They are only selectable by software.

SELFTEST ESCAPE SEQUENCES

The Rainbow 100 computer parses but ignores the escape sequences to run self tests (ESC [2 , Pn y). Also the device status report request (ESC [5 n) will always cause the ready, no malfunctions reply (ESC L 0 n).

RESET TO INITIAL STATE

In Rainbow 100 computer, ESC c (reset to initial state) does not reset keypad and cursor keys to their normal modes. RIS is a dangerous sequence to issue from workstation software. It is not recommended to be used. Its function will change in future versions of Rainbow.

VT52 MODE AND ORIGIN MODE

The Rainbow 100 computer in VT52 mode honors the origin mode setting, VT102 terminal in VT52 modes does not.

AUTOWRAP MODE

The Rainbow 100 computer maintains the wrap-pending flag unconditionally and tests it conditionally. VT102 terminal maintains the flag conditionally and tests it conditionally. This implementation affects where the next character goes when the auto-wrap mode is CHANGED while the cursor is in the 'line-filled' position. The VT102 terminal places the cursor in a different place than the Rainbow. Software is recommended to not use auto wrap mode for controlling the placement of the text on the screen.

TAB AND AUTO WRAP

In Rainbow 100 computer the Tab character always clears the wrap-pending flag. As a result, if Tab is the 81st char in an 80 char line, char 82 will not wrap but char 83 will. In a VT102 terminal, char 82 will wrap.

XON/XOFF PROTOCOL AND BUFFER SIZE IN TERMINAL MODE

In Rainbow 100 terminal mode, the second XOFF is sent at 'buffer-full'. In a VT102 terminal, the second XOFF is sent 12 char before 'buffer-full'. Also the Rainbow 100 buffer is 255 char in size, a VT102 terminal is 128.

FULL DUPLEX COMMUNICATION PROTOCOL IN TERMINAL MODE

The Rainbow 100 computer always precedes the dropping of DTR with a EOT character. The VT102 terminal does not always do this. The Rainbow 100 computer does not disconnect if it is placed in Local mode. The VT102 terminal disconnects if placed in local mode.

HALF DUPLEX COMMUNICATION SUPPORT IN TERMINAL MODE

The Rainbow 100 terminal emulation does not support the half duplex communication protocols of the VT102 terminal.

APPENDIX D

INTERNATIONAL LANGUAGE KEYBOARDS

The figures in this appendix illustrate the different national language keyboards that are or will be available. One of the main differences is the label strip that you add to the top of the keyboard. The label strip comes in the country kit.

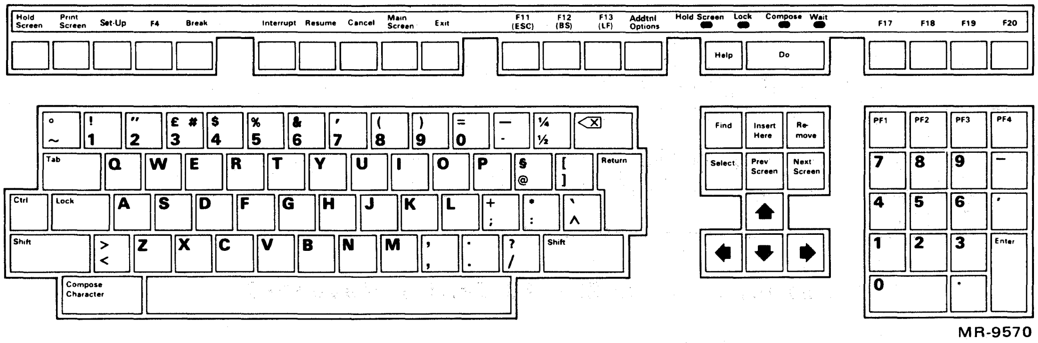


Figure 5: LK201-AE British Keyboard

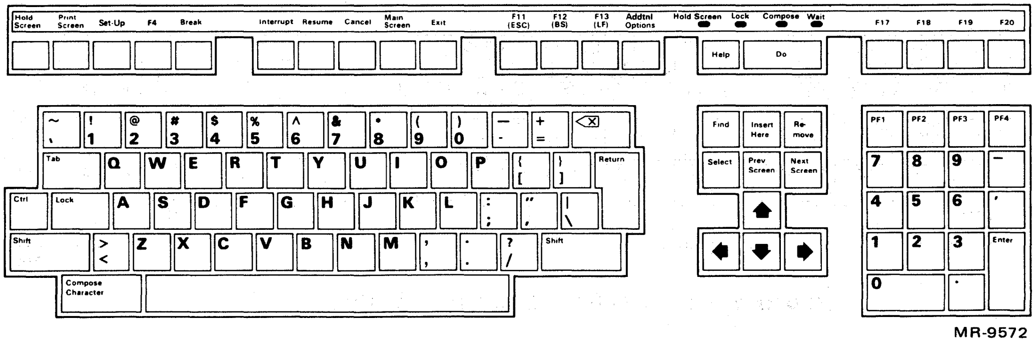


Figure 6: LK201-AA American (English) Keyboard

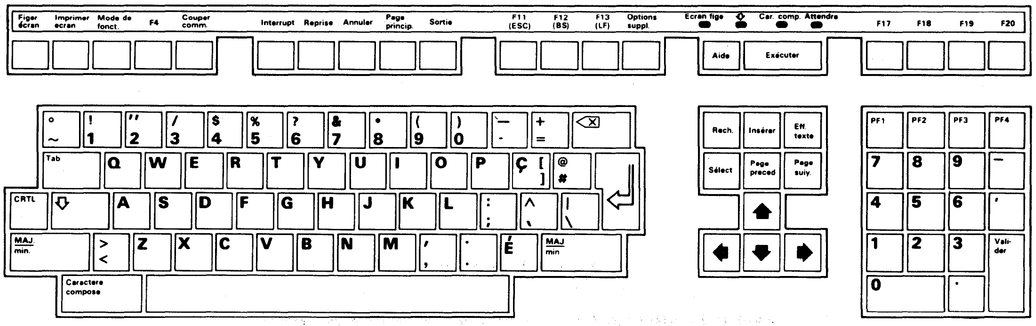


Figure 7: LK201-AC Canadian (French) Keyboard

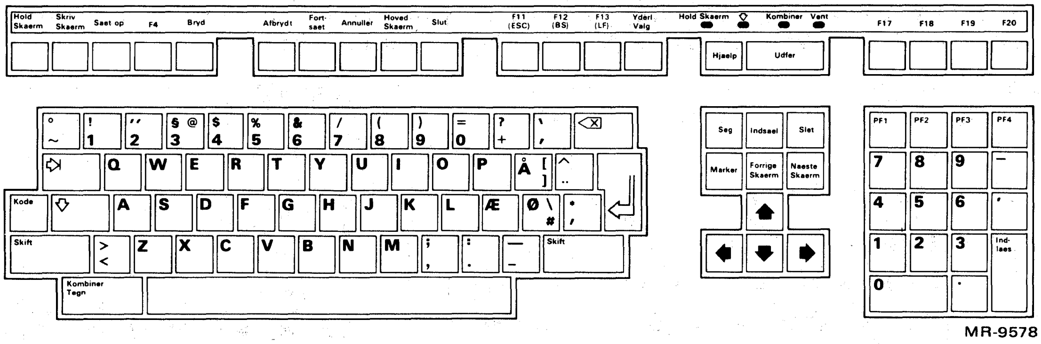


Figure 8: LK201-AD Danish Keyboard

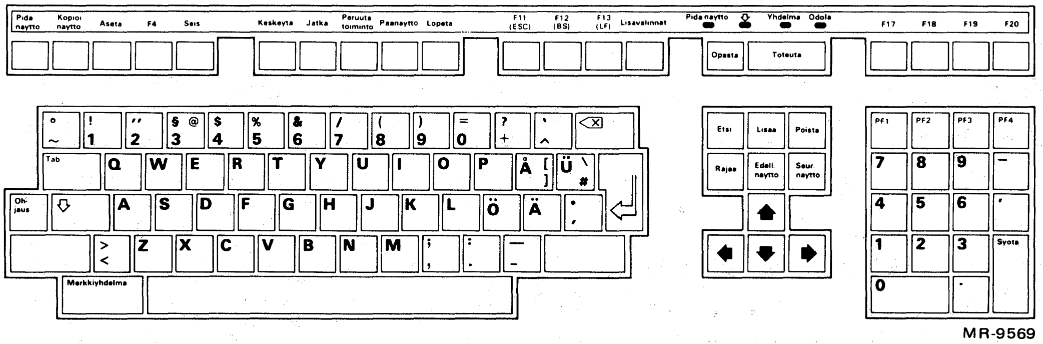


Figure 9: LK201-AF Finnish Keyboard

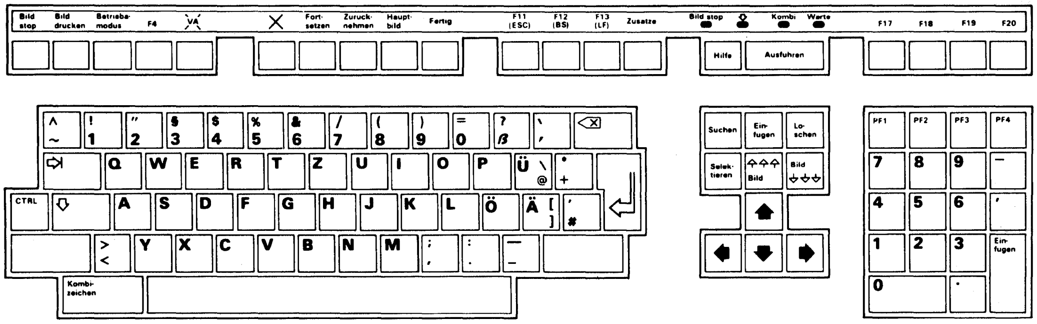


Figure 10: LK201-AG Austrian/German Keyboard

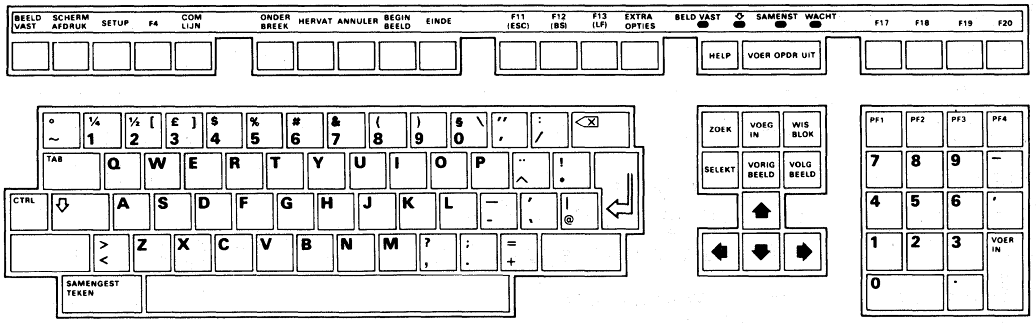


Figure 11: LK201-AH Dutch Keyboard

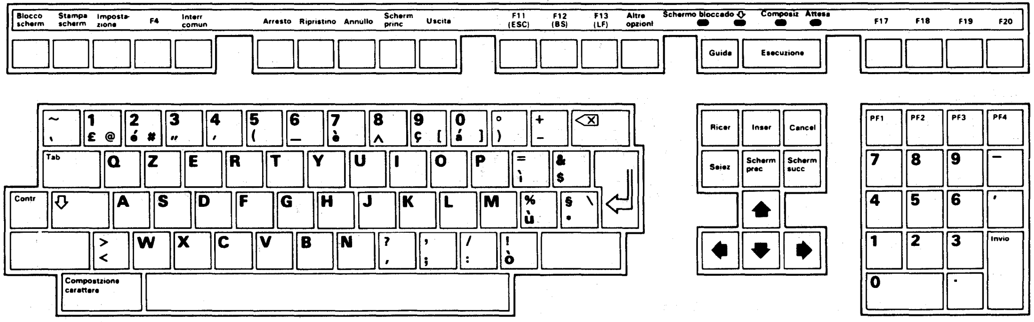


Figure 12: LK201-AI Italian Keyboard

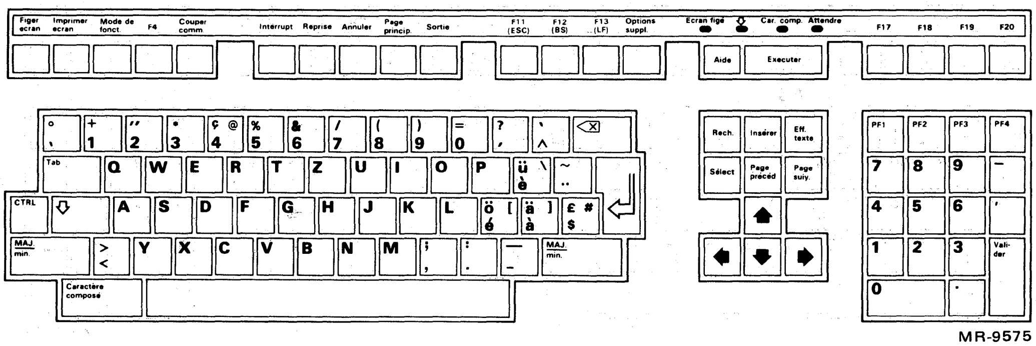


Figure 13: LK201-AK Swiss (French) Keyboard

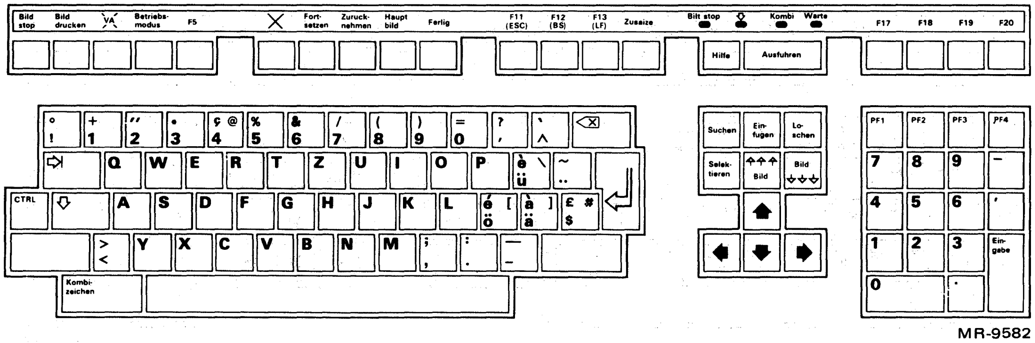


Figure 14: LK201-AL Swiss (German) Keyboard

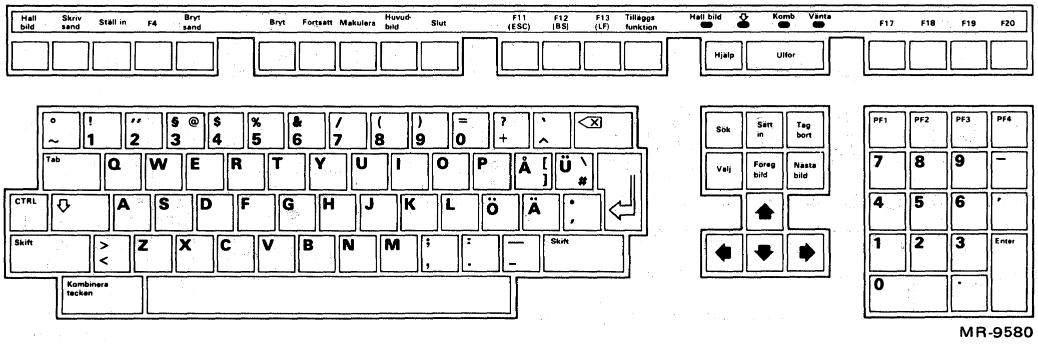


Figure 15: LK201-AM Swedish Keyboard

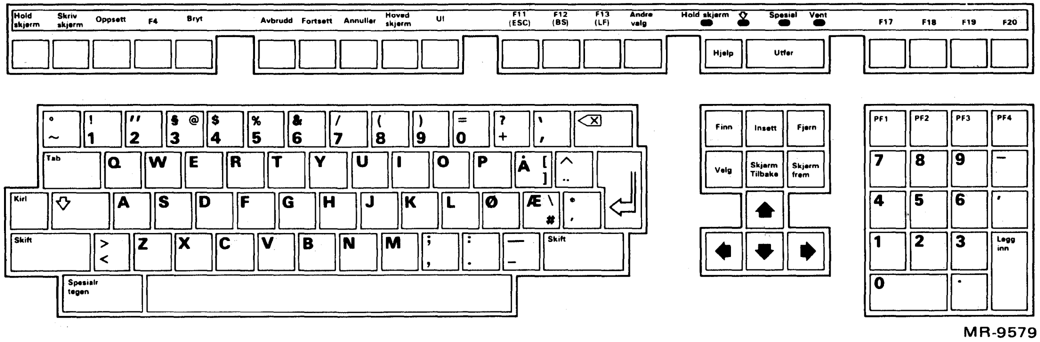


Figure 16: LK201-AN Norwegian Keyboard

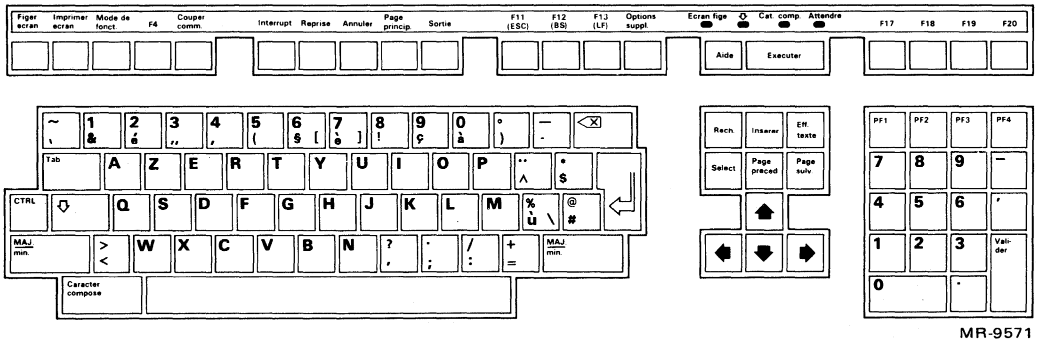


Figure 17: LK201-AP Belgian/French Keyboard

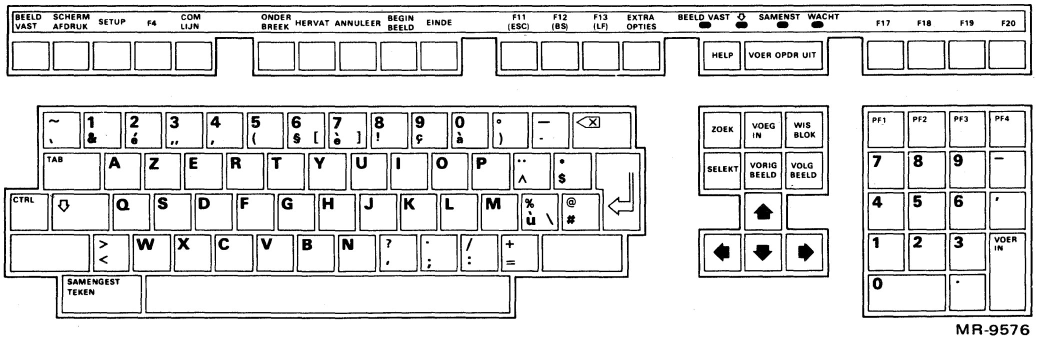


Figure 18: LK201-AT Flemish Keyboard

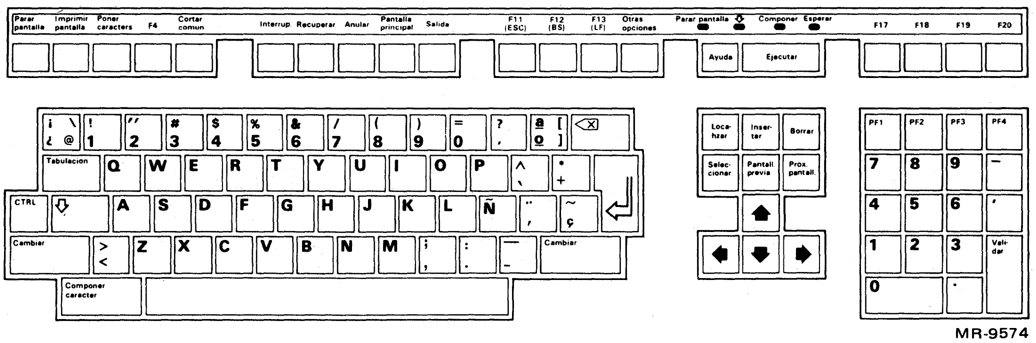


Figure 19: LK201-AS Spanish Keyboard

APPENDIX E

COMPOSE SEQUENCES

Table 26 shows the compose sequences you must type to generate some special characters that you may need.

Ä	A	"	@	a	a	Í	I	'	Ò	O	`	Ú	U	'	[	(	(
ä	a	"	Ç	C	,	í	i	'	ò	o	`	ú	u	'	{	(	-
Á	A	'	ç	c	,	Î	I	^	Œ	O	E	Û	U	^	]	)	)
á	a	'	¢	c	/	î	i	^	œ	o	e	û	u	^	}	)	-
Â	A	^	©	C	O	Ì	I	`	Õ	O	~	Ù	U	`	«	<	<
â	a	^	Ë	E	"	ì	i	`	õ	o	~	ù	u	`	»	>	>
À	A	`	ë	e	"	£	L	-	Ø	O	/	Ÿ	Y	"	¿	?	?
à	a	`	É	E	'	Ñ	N	~	ø	o	/	ÿ	y	"	¡	!	!
Æ	A	E	é	e	'	ñ	n	~	¤	O	X	¥	Y	-	•	^	.
æ	a	e	Ê	E	^	Ö	O	"	º	o	_	°	^	0	|	^	/
Ã	A	~	ê	e	^	ö	o	"	¶	P	!	¹	^	1	#	+	+
ã	a	~	È	E	`	Ó	O	'	§	S	O	²	^	2	±	+	-
Å	A	*	è	e	`	ó	o	'	ß	s	s	³	^	3	\	/	/
å	a	*	Ï	I	"	Ô	O	^	Ü	U	"	½	1	2	µ	/	u
ª	a	_	ï	i	"	ô	o	^	ü	u	"	¼	1	4			

MR-11083

Table 26: Implicit Compose Sequences

Table 27 shows the dead diacritical keys for each language keyboard.

KEYBOARD	DEAD DIACRITICAL KEYS				
K201-AB Belgian/Flemish	~	\	^	"	
K201-AC Canada (French)	~	\	^		
K201-AD Danish	~	\	^	'	"
K201-AE British	~	\	^		
K201-AF Finnish	~	\	^		
K201-AG Austrian/German	~	\	^	'	
K201-AH Netherlands	~	\	^	'	"
K201-AI Italian	~	\	^		
K201-AK Swiss (French)	~	\	^	"	
K201-AL Swiss (German)	~	\	^	"	
K201-AM Swedish	~	\	^		
K201-AN Norwegian	~	\	^	'	"
K201-AP Belgian/French	~	\	^	"	
K201-AS Spanish	~	\	^	'	"

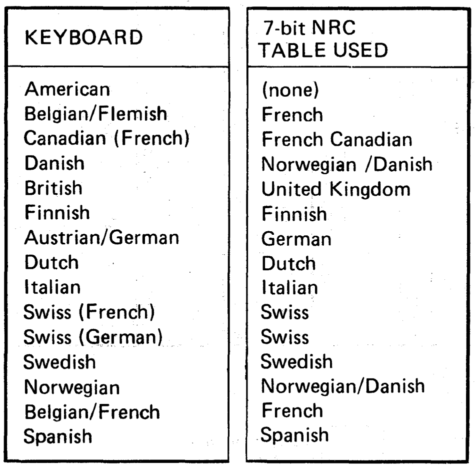
LJ-0140

Table 27: Dead Diacritical Keys

APPENDIX F

7-BIT/DEC 8-BIT TRANSLATIONS

The following figures show the 7-bit and 8-bit translations and character sets for each language keyboard.



KEYBOARD	7-bit NRC TABLE USED
American	(none)
Belgian/Flemish	French
Canadian (French)	French Canadian
Danish	Norwegian /Danish
British	United Kingdom
Finnish	Finnish
Austrian/German	German
Dutch	Dutch
Italian	Italian
Swiss (French)	Swiss
Swiss (German)	Swiss
Swedish	Swedish
Norwegian	Norwegian/Danish
Belgian/French	French
Spanish	Spanish

MR-11081

Figure 20: Mapping Keyboard to National Replacement Characters

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	
BITS		COLUMN															
B4 B3 B2 B1	ROW	0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	ô	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	â	133 91 5B	k	153 107 6B	é	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ù	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	ê	135 93 5D	m	155 109 6D	è	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	î	136 94 5E	n	156 110 6E	û	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

□ Highlights differences from ASCII

MR-11137

Figure 21: French Canadian Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS		COLUMN 0		1		2		3		4		5		6		7	
B4 B3 B2 B1	ROW																
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	é	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5		5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7		7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Ä	133 91 5B	k	153 107 6B	ä	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ö	134 92 5C	l	154 108 6C	ö	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Å	135 93 5D	m	155 109 6D	å	175 125 7D
1 1 1 0	14	SO	16 14		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	Ü	136 94 5E	n	156 110 6E	ü	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

□ Highlights differences from ASCII

MR-11143

Figure 22: Finish Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS		COLUMN															
B4 B3 B2 B1	ROW	0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	°	133 91 5B	k	153 107 6B	é	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ù	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	–	55 45 2D	=	75 61 3D	M	115 77 4D	§	135 93 5D	m	155 109 6D	è	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	¨	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

□ Highlights differences from ASCII

MR-11138

Figure 23: French Character Set (7-bit)

NOTE

Empty positons are reserved for future use.

B7 B6 B5 BITS		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Ä	133 91 5B	k	153 107 6B	ä	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ö	134 92 5C	l	154 108 6C	ö	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Ü	135 93 5D	m	155 109 6D	ü	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	ß	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

□ Highlights differences from ASCII

MR-11135

Figure 24: German Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	ù	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	o	133 91 5B	k	153 107 6B	à	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ò	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	é	135 93 5D	m	155 109 6D	è	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	ì	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER				
	ESC	33 27 1B	OCTAL DECIMAL HEX	□ Highlights differences from ASCII

MR-11136

Figure 25: Italian Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

B7 B6 B5 BITS		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	Ä	100 64 40	P	120 80 50	ä	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Æ	133 91 5B	k	153 107 6B	æ	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ø	134 92 5C	l	154 108 6C	ø	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Å	135 93 5D	m	155 109 6D	å	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	Ü	136 94 5E	n	156 110 6E	ü	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

□ Highlights differences from ASCII

MR-11133

Figure 26: Norwegian/Danish Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	¡	133 91 5B	k	153 107 6B	°	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ñ	134 92 5C	l	154 108 6C	ñ	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	¿	135 93 5D	m	155 109 6D	ç	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX	□	Highlights differences from ASCII

MR-11134

Figure 27: Spanish Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS B4 B3 B2 B1	ROW	COLUMN **0**		**1**		**2**		**3**		**4**		**5**		**6**		**7**	
0 0 0 0	0	**NUL**	0 0 0		20 16 10	**SP**	40 32 20	**0**	60 48 30	**É**	100 64 40	**P**	120 80 50	**é**	140 96 60	**p**	160 112 70
0 0 0 1	1		1 1 1	**DC1** (XON)	21 17 11	**!**	41 33 21	**1**	61 49 31	**A**	101 65 41	**Q**	121 81 51	**a**	141 97 61	**q**	161 113 71
0 0 1 0	2		2 2 2		22 18 12	**''**	42 34 22	**2**	62 50 32	**B**	102 66 42	**R**	122 82 52	**b**	142 98 62	**r**	162 114 72
0 0 1 1	3		3 3 3	**DC3** (XOFF)	23 19 13	**#**	43 35 23	**3**	63 51 33	**C**	103 67 43	**S**	123 83 53	**c**	143 99 63	**s**	163 115 73
0 1 0 0	4		4 4 4		24 20 14	**$**	44 36 24	**4**	64 52 34	**D**	104 68 44	**T**	124 84 54	**d**	144 100 64	**t**	164 116 74
0 1 0 1	5	**ENQ**	5 5 5		25 21 15	**%**	45 37 25	**5**	65 53 35	**E**	105 69 45	**U**	125 85 55	**e**	145 101 65	**u**	165 117 75
0 1 1 0	6		6 6 6		26 22 16	**&**	46 38 26	**6**	66 54 36	**F**	106 70 46	**V**	126 86 56	**f**	146 102 66	**v**	166 118 76
0 1 1 1	7	**BEL**	7 7 7		27 23 17	**’**	47 39 27	**7**	67 55 37	**G**	107 71 47	**W**	127 87 57	**g**	147 103 67	**w**	167 119 77
1 0 0 0	8	**BS**	10 8 8	**CAN**	30 24 18	**(**	50 40 28	**8**	70 56 38	**H**	110 72 48	**X**	130 88 58	**h**	150 104 68	**x**	170 120 78
1 0 0 1	9	**HT**	11 9 9		31 25 19	**)**	51 41 29	**9**	71 57 39	**I**	111 73 49	**Y**	131 89 59	**i**	151 105 69	**y**	171 121 79
1 0 1 0	10	**LF**	12 10 A	**SUB**	32 26 1A	*****	52 42 2A	**:**	72 58 3A	**J**	112 74 4A	**Z**	132 90 5A	**j**	152 106 6A	**z**	172 122 7A
1 0 1 1	11	**VT**	13 11 B	**ESC**	33 27 1B	**+**	53 43 2B	**;**	73 59 3B	**K**	113 75 4B	**Ä**	133 91 5B	**k**	153 107 6B	**ä**	173 123 7B
1 1 0 0	12	**FF**	14 12 C		34 28 1C	**,**	54 44 2C	**<**	74 60 3C	**L**	114 76 4C	**Ö**	134 92 5C	**l**	154 108 6C	**ö**	174 124 7C
1 1 0 1	13	**CR**	15 13 D		35 29 1D	**-**	55 45 2D	**=**	75 61 3D	**M**	115 77 4D	**Å**	135 93 5D	**m**	155 109 6D	**å**	175 125 7D
1 1 1 0	14	**SO**	16 14 E		36 30 1E	**.**	56 46 2E	**>**	76 62 3E	**N**	116 78 4E	**Ü**	136 94 5E	**n**	156 110 6E	**ü**	176 126 7E
1 1 1 1	15	**SI**	17 15 F		37 31 1F	**/**	57 47 2F	**?**	77 63 3F	**O**	117 79 4F	**—**	137 95 5F	**o**	157 111 6F	**DEL**	177 127 7F

KEY

ASCII CHARACTER				
ESC	33 27 1B	OCTAL DECIMAL HEX	□	Highlights differences from ASCII

MR-11141

Figure 28: Swedish Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	ô	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	ù	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5		5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7		7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 22 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	é	133 91 5B	k	153 107 6B	ä	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ö	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	ê	135 93 5D	m	155 109 6D	ü	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	î	136 94 5E	n	156 110 6E	û	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	è	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX	□	Highlights differences from ASCII

MR-10954

Figure 29: Swiss Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

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	
BITS		COLUMN 0		1		2		3		4		5		6		7	
B4 B3 B2 B1	ROW																
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

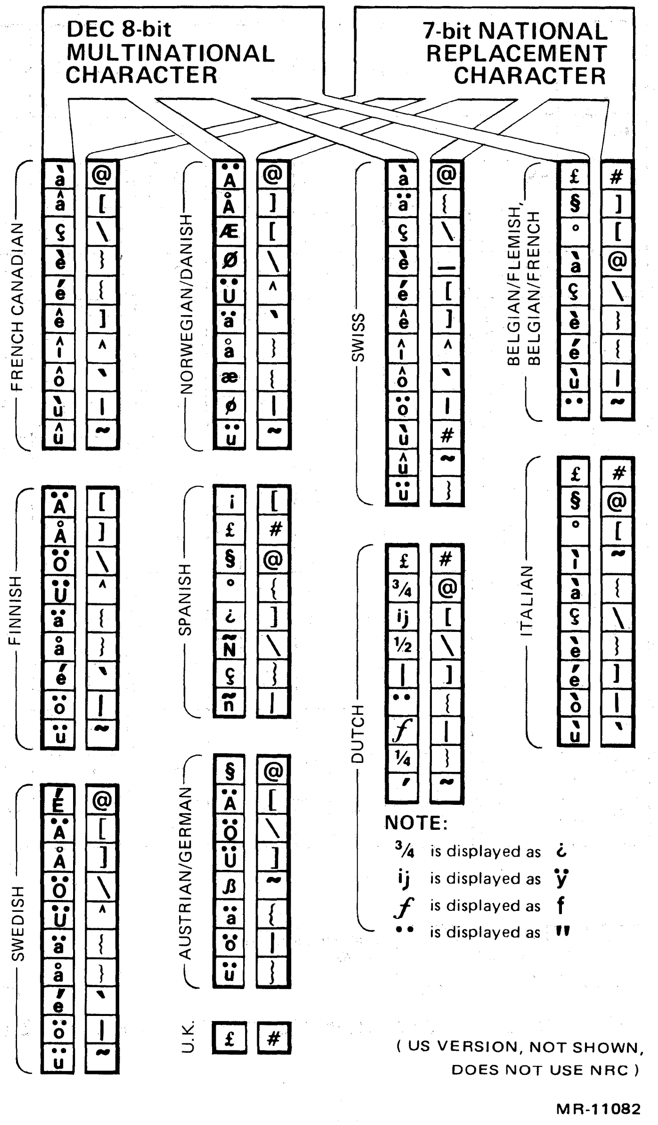
□ Highlights differences from ASCII

MR-11142

Figure 30: United Kingdom Character Set (7-bit)

NOTE

Empty positions are reserved for future use.



FRENCH CANADIAN

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
à	@
â	[
ç	\
è	}
é	{
ê	]
î	^
ô	`
ù	|
û	~

NORWEGIAN/DANISH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
Ä	@
Å	]
Æ	[
Ø	\
Ü	^
ä	`
å	}
æ	{
ø	|
ü	~

SWISS

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
à	@
ä	{
ç	\
è	_
é	[
ê	]
î	^
ô	`
ö	|
ù	#
û	~
ü	}

BELGIAN/FLEMISH, BELGIAN/FRENCH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
£	#
§	]
°	[
à	@
ç	\
è	}
é	{
ù	|
¨	~

FINNISH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
Ä	[
Å	]
Ö	\
Ü	^
ä	{
å	}
é	`
ö	|
ü	~

SPANISH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
¡	[
£	#
§	@
°	{
¿	]
Ñ	\
ç	}
ñ	|

DUTCH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
£	#
¾	@
ij	[
½	\
|	]
¨	{
ƒ	|
¼	}
´	~

ITALIAN

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
£	#
§	@
°	[
ì	~
à	{
ç	\
è	}
é	]
ò	|
ù	`

SWEDISH

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
É	@
Ä	[
Å	]
Ö	\
Ü	^
ä	{
å	}
é	`
ö	|
ü	~

AUSTRIAN/GERMAN

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
§	@
Ä	[
Ö	\
Ü	]
ß	~
ä	{
ö	|
ü	}

U.K.

DEC 8-bit MULTINATIONAL CHARACTER	7-bit NATIONAL REPLACEMENT CHARACTER
£	#

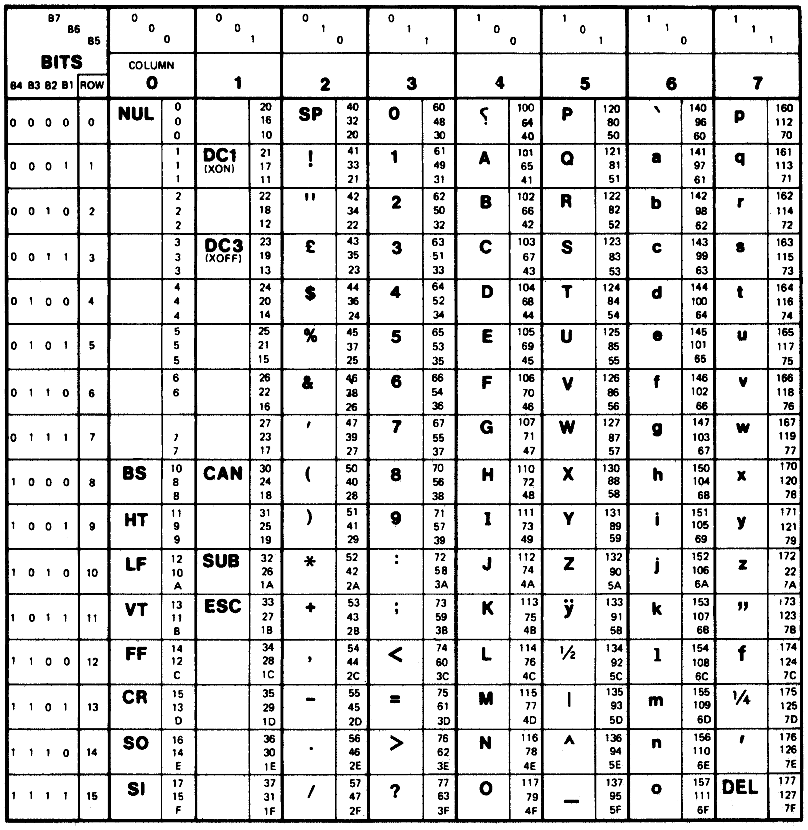
NOTE:

¾ is displayed as ¿
ij is displayed as ÿ
ƒ is displayed as f
¨ is displayed as "

(US VERSION, NOT SHOWN, DOES NOT USE NRC)

MR-11082

Figure 31: 7-bit/DEC8-bit Translations



B7 B6 B5 BITS		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
0 0 0 0	0	NUL	0 0 0		20 16 10	SP	40 32 20	0	60 48 30	⸮	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1		1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2		2 2 2		22 18 12	''	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3		3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4		4 4 4		24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5		5 5 5		25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6		6 6		26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7		7 7		27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9		31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 22 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	ÿ	133 91 5B	k	153 107 6B	”	173 123 7B
1 1 0 0	12	FF	14 12 C		34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	½	134 92 5C	l	154 108 6C	f	174 124 7C
1 1 0 1	13	CR	15 13 D		35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	|	135 93 5D	m	155 109 6D	¼	175 125 7D
1 1 1 0	14	SO	16 14 E		36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	'	176 126 7E
1 1 1 1	15	SI	17 15 F		37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

ASCII CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

□ Highlights differences from ASCII

MR-10955

Figure 32: Dutch Character Set (7-bit)

NOTE

Empty positions are reserved for future use.

Rainbow™ 100+/100B
Terminal Emulation
Manual
QV069–GZ

READER'S COMMENTS

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____

____

____

____

____

Did you find errors in this manual? If so, specify the error and the page number.

____

____

____

____

____

Please indicate the type of reader that you most nearly represent.

□ First-time computer user
□ Experienced computer user
□ Application package user
□ Programmer
□ Other (please specify)____________________

Name____________________________________

Date____________________________________

Organization______________________________

Street___________________________________

City____________________________________

State___________________________________

Zip Code
or Country______________________________

------Do Not Tear - Fold Here and Tape--

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MRO1–2/L12
MARLBOROUGH, MA 01752

---- Do Not Tear - Fold Here and Tape--

Cut Along Dotted Line