RC855 Work Station User´s Guide

RCSL No 42-i 2185

RC855 Work Station
User's Guide

A/S REGNECENTRALEN af 1979
Information Department

January 1982
RCSL No 42-i2185

Author: Jeffrey C. Snider

Keywords: RC855, Work Station, CP/M Operating System, Installation, Operation.

Abstract: This manual describes how to install and operate the CP/M operating system on the RC855 Work Station.

(88 printed pages)

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

APPENDICES:

TABLE OF CONTENTS (continued) PAGE

1. INTRODUCTION 1.

The interface between the RC855 Work Station and its user is the CP/M operating system. CP/M provides the user with a flexible and manageable means of control over the resources of the work station. Specifically:

- CP/M creates an orderly and consistent input/output environment for the various units of the work station to operate in. These units include the console, disk drives, and printer.

- CP/M allows the user to find out what files (programs and data) are on a disk, how large the files are, and how much space is left on the disk. CP/M also handles the writing and reading of information to and from the disk.

- CP/M provides for the loading and execution of user programs. These programs include utility programs for various "housekeeping" tasks as well as applications packages, for example, a word-processing system, mailing-list system, or accounting system.

The present User's Guide gives a brief description of the CP/M operating system available for the RC855 Work Station. Those wishing a complete description of CP/M should consult the Digital Research manuals listed in appendix A (ref. [1]). The User's Guide also contains reference material for the RC855 implementation of CP/M, which, besides the RC855 BIOS, includes several additional utility programs. CP/M and the RC855 implementation of CP/M are described in more detail below.

Chapter 2 explains how to get started using CP/M, and chapter 3 how CP/M "works" in general. After having read these two chapters, the user will be able to run programs on the RC855 Work Station, as described in chapter 4.

1.1 CP/M

CP/M (Control Program/Microprocessors) is a software system designed to record and retrieve programs and data on disks. Like other software systems, it is a collection of interrelated programs which perform specific tasks within the system. CP/M operates with 8080 and Z80 microprocessors and is largely independent of the design of the computer and disk system. It has therefore been adopted for use with most computers employing the 8080 and Z80 families of microprocessors. CP/M is today the industry standard in operating systems for small computers, and a multitude of high-level languages and applications software has been designed to run under its control.

CP/M is divided into four main parts:

CCP

The Console Command Processor, or CCP, is the interface between the user's console and CP/M. The CCP executes its own, built-in commands to

- list the filenames in a directory,
- delete a file,
- rename a file,
- display the contents of a text file, and
- save a copy of the memory contents in a file.

The CCP also executes transient commands to control the operation of various programs. The built-in and transient commands are described in chapter 4.

BDOS

The Basic Disk Operating System, or BDOS, provides disk and file management capabilities and dynamic file allocation. In addition, the BDOS executes primitive operations to

- select a disk drive for further operations,
- create a file entry and prepare for opening,
- open a file for further operations and read in the file control block,
- close a file and write out the file control block,
- search for a disk file by name,

- delete a filename and free the allocated disk space,
- rename a file,
- read a record from a file, and
- write a file record to a disk.

TPA

The Transient Program Area, or TPA, holds programs loaded from disk under control of the CCP. The TPA is found between random-access memory address 0100H ("H" denotes the hexadecimal radix) and the starting address of the CCP. A program executing in the TPA can overlay the memory areas occupied by other parts of CP/M in order to use them as its data area. The complete CP/M system is reloaded from disk, if the transient program branches to the bootstrap loader at the end of execution.

BIOS

The Basic I/O System, or BIOS, defines the particular hardware environment in which CP/M will operate and performs logical device mapping. The BIOS includes a buffer manager and the primitive operations necessary to interface standard peripherals, such as CRT terminals, printers, and disk drives. An example of patching a customized BIOS into CP/M is given in appendix C.

For a full discussion of these matters, see ref. [1], chapters 5 and 6.

1.2 RC855 CP/M 1.2

The CP/M system supplied for the RC855 Work Station is a 56K CP/M version 2.2 with 50 1/4 K bytes available for transient programs. The system supports one or two 8-inch diskette drives, each with a capacity of 1124K bytes. The diskette formats are described in appendix E.

The CP/M disk drives A and B (ref. [1], 1.2, p. 3) are supported as the RC855 disk drives 1 and 2, respectively. The RC855 printer port is supported as the CP/M output list device, LST (ref. [1], 1.6.1, p. 11). The RC855 line port is supported as the CP/M reader/punch devices, RDR and PUN (ref. [1], 1.6.1, p. 11). Further information on peripheral support is given in appendix F.

The system does not support the IOBYTE function nor the modification of logical-physical device assignments by the STAT command (ref. [1], 1.6.1, pp. 10-15).

The system does not include the MOVCPM program (ref. [1], 1.6.9, pp. 27-29).

The RC855 implementation of CP/M, release 1.3, includes all patches from Digital Research up to and including patch No 9 of February 11, 1982.

2. INSTALLING CP/M 2.

The CP/M package for the RC855 Work Station contains:

a. This manual.

b. The Digital Research manual:
CP/M Operating System manual

c. An 8-inch diskette containing your CP/M system.

d. Software License Agreement and Registration Card.

Your copy of the CP/M system for the RC855 is provided with a serial number and is licensed for your use only on a single RC855 Work Station. Therefore, before opening the diskette package, please read the Software License Agreement carefully and fill in and return the Registration Card.

Do not write on the original distribution diskette. It is your master copy and last resort in the event of errors. Start by making a backup copy of this diskette as described in section 2.3. It may be wise to keep an additional backup copy in a safe place.

2.1 Notational Conventions in This Manual 2.1

The following notational conventions apply in the examples presented in this manual:

- A comment is preceded by a semicolon (;).
- Keyboard input is underscored.
- A "cr" indicates that the "RETURN" key is pressed.

Note that the "RETURN" key is marked ↵ on the RC855 keyboard.

2.2 How to Bring Up the System 2.2

The following procedure will load the CP/M system from disk into the RC855 random-access memory. This system bootstrap operation is also known as a system boot, cold boot, or cold start.

- Turn on the console.
- Turn on the disk drive(s).
- Insert the System Diskette in drive A (drive 1).
- Wait for the following sign-on message to be displayed:

```
RC855 56k CP/M 2.2 rel. 1.3
A>
```

CP/M will now accept a command (ref. [1], 1.2, p. 3).

Note: Remove all disks before turning off the drives and console.

2.3 How to Copy the System Diskette 2.3

This section shows how to copy the System Diskette. The utility programs FORMAT and BACKUP are used. Both programs are fully described in chapter 4.

2.3.1 Two-Drive System 2.3.1

Start by formatting a new diskette for the copy. Insert the System Diskette in drive A (drive 1) and the new diskette in drive B (drive 2) and proceed as follows:

```
A>FORMAT cr
RC855 FORMAT UTILITY VERS 1.2 82.03.03
FORMAT: 1=SS/SD 2=DS/DD TYPE (1,2) ? 2
SELECT DRIVE (A/B) ? B
INSERT DISK AND TYPE RETURN cr
FORMAT DISKETTE IN DRIVE B (Y/N) ? Y
NUMBER OF FORMATTED TRACKS 077
```

```
FORMAT COMPLETED
TYPE T TO TERMINATE
TYPE C TO CONTINUE T
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
;continue with BACKUP
A>BACKUP cr
RC855 BACKUP VERS 1.0 81.12.17
SOURCE DRIVE (A OR B) ? A
DESTINATION DRIVE (A OR B) ? B
INSERT SOURCE DISK AND TYPE RETURN cr
INSERT DESTINATION DISK AND TYPE RETURN cr
BACKUP COMPLETE
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
```

2.3.2 One-Drive System 2.3.2

Start by formatting a new diskette for the copy. Insert the System Diskette in the drive and proceed as follows:

```
A>FORMAT cr
RC855 FORMAT UTILITY VERS 1.2 82.03.03
FORMAT: 1=SS/SD 2=DS/DD TYPE (1,2) ? 2
SELECT DRIVE (A/B) ? A
;change to new diskette
INSERT DISK AND TYPE RETURN cr
FORMAT DISKETTE IN DRIVE A (Y/N) ? Y
NUMBER OF FORMATTED TRACKS 077
FORMAT COMPLETED
TYPE T TO TERMINATE
TYPE C TO CONTINUE T
;change back to System Diskette
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
;continue with BACKUP
A>BACKUP cr
RC855 BACKUP VERS 1.0 81.12.17
```

```
SOURCE DRIVE (A OR B) ? A
DESTINATION DRIVE (A OR B) ? A
;source disk = System Diskette
;destination disk = new diskette
INSERT SOURCE DISK AND TYPE RETURN cr
INSERT DESTINATION DISK AND TYPE RETURN cr
;continue to alternate between the source disk and the destina-
;tion disk until the following message is displayed
BACKUP COMPLETE
;final change back to System Diskette
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
```

3. OPERATING CP/M 3.

This chapter contains general information on the operation of the CP/M system. The various commands which can be given to CP/M are described in chapter 4. Error recovery procedures are described in chapter 5.

3.1 Restarting the System 3.1

The CP/M system can be restarted in two different ways. In either case an executing program will be aborted, control will return to the CCP, the system prompt (usually "A>") will reappear, and CP/M will accept a new command.

System Boot (Cold Boot)

A system boot (cold boot or cold start) occurs whenever the system is brought up, as described in section 2.2. A system boot will also occur when the RESET button at the back of the console is pressed (hard reset) or when the CTRL and CLEAR keys are pressed simultaneously (soft reset). Note that a soft reset is possible only so long as the BIOS is active.

System Reboot (Warm Boot)

A system reboot (warm boot or warm start) will occur when the CTRL and C keys are pressed simultaneously. Note that a system reboot is required when a new disk is introduced with the intent to write on it (disk reset).

3.2 Command Lines 3.2

A command is generally given to CP/M by typing a command line immediately following the system prompt ("A>"). The typical command line contains, from left to right:

- the name of the command, e.g. DIR or STAT,
- a mandatory blank, and
- an argument, e.g. a file reference.

The command line is terminated by pressing the "RETURN" key, which causes the command to be executed. "RETURN" also generates a carriage return and line feed, moving the cursor to the beginning of the next line.

3.3 Line Editing and Output Control 3.3

The editing of command lines and the control of console output are described in ref. [1], 1.5, p. 9. Note the following with regard to the RC855 keyboard:

- The key marked ← deletes the last character typed.
- The key marked → deletes the entire line typed.
- The key marked PRINT copies all subsequent console output to the list device.

Note also that an up arrow (↑) may be used to denote the CTRL key in various CTRL-key combinations, e.g. "↑C" for "CTRL-C" (system reboot).

3.4 File References 3.4

Nearly all of the CP/M commands reference a file or a group of files.

A file reference identifies a particular file or group of files on a particular disk. The reference consists accordingly of a filename and possibly a preceding disk drive name. The filename comprises a primary name, which is the proper name of the file, and a secondary name, or extension, which indicates the type of the file. The general form of a filename is

```
pppppppp.sss
```

where pppppppp is the primary name of eight characters or less, and sss is the secondary name, or extension, of three characters or less. Note that a period (.) is mandatory between the two names.

Extensions

Specific extensions are required by CP/M for several types of files. (Note that word-processing and other systems may require different extensions). Specific extensions are not required by CP/M for data or text files, but are often a convenience for the user. Here are some standard CP/M extensions:

COM	command file of a transient command (directly executable program)
ASM	source file of an assembly language program
PRN	listing file of an assembly language program
HEX	program file of an assembly language program in "hex" format machine code
BAK	backup file created by ED before modification of the original file
SYS	system file (see, for example, subsection 4.3.5)

Unambiguous and Ambiguous Filenames

The filename form pppppppp.sss identifies a single file, and is therefore called an unambiguous filename. A reference containing an unambiguous filename is indicated by "ufn" in the command descriptions in chapter 4.

An ambiguous file reference may be satisfied by several different files. In an ambiguous filename, an asterisk (*) can be used to match all eight characters in the primary name and/or all three characters in the secondary name. Also, a question mark (?) can be interspersed throughout the primary and secondary names to match any single character in the "?" position. The "*" and "?" are sometimes called wild-card characters. A reference containing an ambiguous filename is indicated by "afn" in the command descriptions in chapter 4.

Drive Name

A filename can generally be preceded by a disk drive name ("A" or "B") in order to log in the drive containing the disk with the relevant file(s) before the file operation takes place. The filename and the drive name must be separated by a colon (:).

Examples

```
A>DIR FILE4.TXT cr      ;lists, if found, the file (unambiguously)
                        ;named FILE4.TXT

A>DIR *.COM cr          ;lists any and all files with the COM ex-
                        ;tension

A>DIR HAZ.* cr          ;lists any and all files with the primary
                        ;name HAZ

A>DIR *.* cr            ;lists all of the files on the disk in
                        ;drive A

A>DIR FILE?.TXT cr      ;lists any and all files whose primary
                        ;names match FILE? (e.g. FILE1, FILE2,
                        ;and FILE3) with the TXT extension

A>DIR B:*.TXT cr        ;lists any and all files with the TXT ex-
                        ;tension on the disk in drive B
```

For a more detailed description of file references, see ref. [1], 1.2.2, pp. 3-5. The DIR command is described in chapter 4.

3.5 Switching Drives 3.5

CP/M indicates the currently logged drive, and by implication the disk which it contains, by prompting with the disk drive name ("A" or "B") followed by a right angle bracket (>). On a two-drive system, the user can switch the currently logged drive by typing the disk drive name followed by a colon (:) when the CCP is awaiting console input (see further ref. [1], 1.3 p. 5).

Example

```
A>B: cr                  ;logs in drive B
B>
```

3.6 US ASCII Character Set 3.6

The RC855 keyboard has a number of variants, corresponding to different national ASCII character sets. Since the CP/M system assumes the use of the US ASCII character set, the question of "which key to press" may arise. A case in point is the square brackets ([]) required by CP/M for PIP parameters, which are not represented in all ASCII character sets. The characters on each RC855 keyboard which differ from the US ASCII characters are listed in appendix D.

4. CP/M COMMANDS 4.

This chapter describes the commands which can be given to the CP/M operating system from the user's console. The commands are arranged in three groups according to their principal applications: file handling, program handling, and device handling.

CP/M commands are implemented on two levels: built-in commands and transient commands.

Built-In Commands

The built-in commands can always be executed, as they are part of CP/M itself. Since they do not exist as files, they are not listed in the directory.

Transient Commands

A transient command can be executed only if it exists as a command file (COM extension) on the disk in the currently logged drive. The execution of a transient command causes the command file (containing a directly executable program) to be loaded into the TPA. The transient programs which can be executed in this manner include:

- the standard CP/M editor, assembler, debugger, and utilities,
- the RC855 CP/M utility programs, and
- high-level languages and applications software of the user's own choosing.

Note that the primary name of the command file is used as the name of the transient command in the command line.

Examples

```
A>ERA (argument) cr          ;built-in command ERA

A>PIP (argument) cr          ;standard transient command PIP
```

```
A>TRANSFER (argument) cr     ;RC855 transient command TRANSFER

A>COBOL (argument) cr        ;transient command to load the CIS
                             ;COBOL compiler

A>WS cr                      ;transient command to load the
                             ;WordStar word-processing system
```

4.1 File Handling Commands 4.1

This group of commands comprises DIR, ERA, REN, TYPE, STAT, PIP, TRANSFER, ED, CAT and FILEX.

4.1.1 DIR 4.1.1

```
Built-in command,
the forms of which are:      DIR ufn cr
                             DIR afn cr
```

The DIR (directory) command lists on the console one or more filenames in the directory of the disk in the currently logged drive.

Note that console output can be temporarily halted by pressing the CTRL and S keys simultaneously.

Examples

```
A>DIR FILE4.TXT cr       ;lists, if found, the file (unambiguous-
                         ;ly) named FILE4.TXT

A>DIR *.COM cr           ;lists any and all files with the COM ex-
                         ;tension

A>DIR HAZ.* cr           ;lists any and all files with the primary
                         ;name HAZ
```

```
A>DIR *.* cr            ;lists all of the files on the disk in
                        ;drive A

A>DIR cr                ;short form of the above command

A>DIR B: cr             ;equivalent to the command DIR B:*.* cr

A>DIR B:*.TXT cr        ;lists any and all files with the TXT ex-
                        ;tension on the disk in drive B
```

See further ref. [1], 1.4.2 pp. 6-7.

4.1.2 ERA 4.1.2

Built-in command,
the forms of which are: ERA ufn cr
ERA afn cr

The ERA (erase) command deletes one or more filenames in the directory of the disk in the currently logged drive, and frees the allocated storage space on the disk.

Examples

```
A>ERA FILE4.TXT cr      ;deletes, if found, the file (unambigu-
                        ;ously) named FILE4.TXT

A>ERA *.ASM cr          ;deletes any and all files with the ASM
                        ;extension

A>ERA HAZ.* cr          ;deletes any and all files with the pri-
                        ;mary name HAZ

A>ERA FILE?.TXT cr      ;deletes any and all files whose primary
                        ;names match FILE? (e.g. FILE1, FILE2,
                        ;and FILE3) with the TXT extension
```

```
A>ERA *.* cr              ;deletes all of the files on the disk in
ALL FILES (Y/N) ? Y       ;drive A following the Y (yes) confirma-
                          ;tion

A>ERA B:*.PRN cr          ;deletes any and all files with the PRN
                          ;extension on the disk in drive B
```

See further ref. [1], 1.4.1, p. 6.

4.1.3 REN 4.1.3

Built-in command,
the form of which is: REN ufn1=ufn2 cr

The REN (rename) command renames an unambiguously named file in the directory of the disk in the currently logged drive. The old filename is indicated by "ufn2", and the new filename by "ufn1".

Examples

```
A>REN HAZMAY31=HAZMAY30 cr      ;the file HAZMAY30 is renamed
                                ;HAZMAY31

A>REN FILE7.TXT=FILE7.BAK cr    ;the file FILE7.BAK is renamed
                                ;FILE7.TXT

A>REN B:DATA4=DATA1 cr          ;the file DATA1 on the disk in
                                ;drive B is renamed DATA4
```

See further ref. [1], 1.4.3, p. 7.

4.1.4 TYPE 4.1.4

Built-in command,
the form of which is: TYPE ufn cr

The TYPE command displays on the console the contents of an unambiguously named ASCII text file (e.g. a source file or PRN file) on the disk in the currently logged drive.

Note that console output can be temporarily halted by pressing the CTRL and S keys simultaneously.

Note also that console output can be copied to the list device by pressing the PRINT key.

Examples

```
A>TYPE HAZMAY31 cr           ;displays the contents of the file
                             ;HAZMAY31

A>TYPE FILE7.TXT cr          ;displays the contents of the file
                             ;FILE7.TXT

A>TYPE B:PROG1.PRN cr        ;displays the contents of the file
                             ;PROG1.PRN on the disk in drive B
```

See further ref. [1], 1.4.5, p. 8.

4.1.5 STAT 4.1.5

Standard transient command,
the forms of which are:

```
STAT cr
STAT argument cr
```

The STAT command can display on the console a variety of information, particularly on the status of files and disks, and perform certain file and device handling functions (see examples below). The status information for files includes the file size and the access mode, and for disks, the number of unused bytes and the access mode.

Note that files with the $SYS (system) attribute are not listed in the directory, but their status, enclosed in parentheses, will be displayed by STAT.

Examples

```
A>STAT FILE4.TXT cr               ;displays the status of the file
                                  ;(unambiguously) named FILE4.TXT

A>STAT FILE4.TXT $S cr            ;equivalent to the above command,
                                  ;but provides additional informa-
                                  ;tion on the file size

A>STAT B:*.TXT cr                 ;displays the status of any and all
                                  ;files with the TXT extension on
                                  ;the disk in drive B

A>STAT cr                         ;displays the status of the disk in
                                  ;drive A

A>STAT B:DSK: cr                  ;displays the drive characteristics
                                  ;of the disk in drive B

A>STAT SAMPLE.TXT $R/O cr         ;sets the read-only attribute on
                                  ;the file SAMPLE.TXT (until reset
                                  ;by a $R/W)

A>STAT SAMPLE.BAK $SYS cr         ;sets the system attribute on the
                                  ;file SAMPLE.BAK (until reset by a
                                  ;$DIR)

A>STAT B:=R/O cr                  ;sets the read-only attribute on
                                  ;the disk in drive B (until the
                                  ;next warm or cold boot)

A>STAT VAL: cr                    ;displays a list of the possible
                                  ;STAT commands and a list of the
                                  ;possible device assignments

A>STAT DEV: cr                    ;displays the actual device assign-
                                  ;ments
```

```
A>STAT USR: cr                ;displays the number of the current
                              ;user area (see subsection 4.3.8)
```

Note that, in the RC855 implementation of CP/M, the STAT command cannot be used to change the actual device assignments.

See further ref. [1], 1.6.1, pp. 10-15.

4.1.6 PIP 4.1.6

Standard transient command,
the forms of which are:

```
PIP cr
PIP destination=source cr
```

The PIP command (either form) loads and executes the CP/M Peripheral Interchange Program. The primary use of PIP is to copy, or transfer, files between disks or between a disk and a peripheral device. PIP, however, is not just a simple "copy" program, but a general transfer program provided with numerous processing options.

PIP can be executed as a single-line command, where the destination is the file or logical device that receives the copy, and the source is the file or logical device from which the copy is taken. Processing options are indicated by various PIP parameters, enclosed in square brackets ([]), following the source.

The second command form, PIP cr, causes the PIP program to read command lines directly from the console, prompted by an asterisk (*), until an empty command line (a single "cr") is typed. A sequence of copy operations can be executed in this way.

To transfer files between disks on a one-drive system, use the TRANSFER program (see subsection 4.1.7). To copy an entire diskette, irrespective of the number of disk drives, use the BACKUP program (see subsection 4.3.2).

Note the following general rule: Two files with the same name cannot exist on the same disk.

Examples

```
A>PIP DOC.TXT=DOC.BAK cr          ;copies the file (unambiguously)
                                  ;named DOC.BAK on the disk in
                                  ;drive A and names the copy DOC.TXT

A>PIP B:Z.ASM=A: cr               ;copies the file (unambiguously)
                                  ;named Z.ASM from the disk in drive
                                  ;A to the disk in drive B

A>PIP B:COPY1.BAK=FILE1.TXT cr    ;copies the file (unambiguously)
                                  ;named FILE1.TXT from the disk in
                                  ;drive A to the disk in drive B and
                                  ;names the copy COPY1.BAK

A>PIP B:=DOC?.* cr                ;copies any and all files whose
                                  ;primary names match DOC? (e.g.
                                  ;DOC1, DOC2, and DOC3) and with any
                                  ;extension from the disk in drive A
                                  ;to the disk in drive B

A>PIP B:=*.COM [V] cr             ;copies any and all files with the
                                  ;COM extension from the disk in
                                  ;drive A to the disk in drive B and
                                  ;verifies the copy

A>PIP LST:=B:FILE2 cr             ;copies the file (unambiguously)
                                  ;named FILE2 on the disk in drive B
                                  ;to the list device

A>PIP cr                          ;a sequence of copy operations
*FILE2=TEST2 cr
*LST:=FILE2 cr
*B:=PROG.ASM cr
*cr
A>
```

Note that using PIP to list a file on the printer is faster than using the TYPE command and the PRINT key.

Note also that, because PIP copies a file on adjacent sectors of the disk, the copied file may possibly be accessed faster (by ED and similar programs) than the original file.

See further ref. [1], 1.6.4, pp. 17-23.

4.1.7 TRANSFER 4.1.7

RC855 transient command,
the form of which is: TRANSFER cr

The TRANSFER program is used to transfer files between disks on systems with one disk drive. TRANSFER transfers a file in portions of 32K bytes: first it reads up to 32K bytes from the source file into main memory, then it asks the user to swap disks, and finally it writes the memory contents to the destination file. This sequence is repeated until the transfer is complete.

The format of both the source and the destination disk must be specified as either single-sided, single-density (SS) or double-sided, double-density (DD). It is possible to transfer files from the single-sided, single-density format to the double-sided, double-density format of the CP/M system.

The source and destination file may have the same name, as they normally reside on different disks. Note that a source file larger than 32K bytes will be destroyed, if it is written to the same disk (possibly because the user forgot to swap disks) under the same name.

TRANSFER prompts the console for commands, with interaction as shown below. A system reboot (warm boot) can be performed whenever the program is requesting input.

Example

The following example shows how to transfer a file named FILETXT.COM from a double-sided, double-density diskette to a single-sided, single-density diskette. The source file contains less than 32K bytes.

```
A>TRANSFER cr
TRANSFER UTILITY VERS 2.0 82.01.05
SOURCE DISK TYPE (SS:=1, DD:=2): 2
DESTINATION DISK TYPE (SS:=1, DD:=2): 1
SOURCE FILENAME: FILETXT.COM
DESTINATION FILENAME: FILETXT.COM
INSERT SOURCE DISK AND TYPE RETURN cr
INSERT DESTINATION DISK AND TYPE RETURN cr
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
TRANSFER COMPLETE
A>
```

4.1.8 ED 4.1.8

Standard transient command,
the form of which is: ED ufn cr

The ED command loads and executes the CP/M system context editor, which allows the user to create a file on disk, modify the contents of an ASCII text file, or prepare an assembly language source file. The unambiguous filename, ufn, specifies the file on which ED is to operate. A secondary name (extension) must be supplied.

The ED program creates a buffer in which the user can modify text. First, ED deletes any existing backup file whose primary name matches that of the source file, i.e. X.BAK is deleted when X.TXT is the source. Then, ED prompts the console with an asterisk (*) for editing commands. The user can now append text from the source file to the buffer, insert or delete text in the buffer, and perform other editing functions. The user can also write text to a temporary file, X.$$$, as he edits. When ED is terminated, it writes the buffer contents and any remaining text in the source file, X.TXT, to X.$$$, which now constitutes a revised version of X.TXT. ED then changes the name X.TXT to X.BAK, and X.$$$ to X.TXT.

Examples

```
A>ED SAMPLE.TXT cr              ;creates a file (unambiguously)
NEW FILE                        ;named SAMPLE.TXT on the disk in
*                               ;drive A

A>ED B:FILE3.TXT A: cr          ;specifies the source file as
*                               ;FILE3.TXT on the disk in drive B
                                ;and the revised source file as a
                                ;file on the disk in drive A
```

See further ref. [1], 1.6.5, pp. 23-24 and particularly ref. [1], chapter 2 pp. 33-45.

4.1.9 CAT

RC855 transient command,
the forms of which are:

```
CAT cr
CAT options cr
CAT ufn cr
CAT ufn options cr
```

The CAT command lists on the console one or more filenames, in alphabetical order.

The available options are:

1) $SYS - if this option is used the listing will include files with the SYS attribute. They will be listed in parentheses.

2) $R/O - if this option is used, all listed files with Read-Only status will be followed by an asterisk (*).

If the filename is omitted, for instance as in the call

```
   >CAT   $SYS cr
```

the option must be preceded by two blanks.

Examples

```
A>CAT *.COM cr          ;lists all files with extension
                        ;COM, except files with the SYS attribute.

A>CAT *.COM  $SYS cr    ;lists all files with the extension
                        ;COM. Files with the SYS attribute will
                        ;be listed in parentheses.

A>CAT PROG.* $R/O cr    ;lists all files with the primary name
                        ;PROG except files with the SYS attribute.
                        ;Files with the R/O attribute will be
                        ;marked with an asterisk (*).

A>CAT  $SYS $R/O cr     ;lists all files on the default drive.
                        ;Files with the SYS attribute will be
                        ;listed in parentheses. Files with the R/O
                        ;attribute will be marked with an asterisk
                        ;(*).

A>CAT B: cr             ;lists all files on disk B except
                        ;those with the SYS attribute.
```

4.1.10 FILEX

RC855 transient command,
the forms of which are:

```
                    FILEX cr
                    FILEX destination=source cr
```

The FILEX command (either form) loads and executes a program that transfers a file between two computers via the line port (see appendix F.2).

FILEX may only be used in conjunction with two computers that have been connected with an appropriate cable (see app. H). It is furthermore required that the line ports of both computers use the same linespeed (baud rate) and character format (use the CONFI command, 4.3.7, to make sure that this is the case).

The FILEX command may be entered on either computer, so the user must first decide which computer he wishes to operate from. This computer will be referred to as the local station. The other computer will be called the remote station.

Before operating from the local station, it is necessary to enter the command line

```
A>FILEX REMOTE cr
```

on the computer chosen as the remote station. This will put it into the remote mode.

The FILEX command can then be entered at the local station in one of two ways:

FILEX can be entered as a single-line command using the format: FILEX destination=source cr . A destination or source must include a drive name (A or B) followed by a colon and an unambiguous file-name. To make it possible to distinguish between the local drives A and B and the remote drives A and B, the remote drive names must be preceded by the letter R. (Thus RA: and RB: refer to the remote drives A and B respectively.)

The other command form, FILEX cr, causes the FILEX program to read command lines directly from the console, prompted by an asterisk (*). A command line must have the format: destination=source cr. One file can be transferred per command line. FILEX will continue to prompt the user until an empty command line (a single "cr") is typed.

When the local station has finished executing FILEX, the other computer will exit from the remote mode. If this is undesirable, the user should type NOEND between source and "cr" in the filex command line, e.g. A>FILEX destination=source NOEND cr.

Examples

```
A>FILEX REMOTE cr                  ;The remote station is ini-
                                   ;tialized.
A>FILEX A:PAP.COM=RB:PIP.COM cr    ;The file PIP.COM is copied
                                   ;from the remote station's
                                   ;B-drive to the local sta-
                                   ;tion's A-drive under the
                                   ;name PAP.COM.

A>FILEX RB:=A:OLD.TXT NOEND cr     ;The file OLD.TXT is copied
                                   ;from the local station's
                                   ;A-drive to the remote sta-
                                   ;tion's B-drive under the name
                                   ;OLD.TXT. The NOEND option
                                   ;prevents the remote station
                                   ;from exiting from the remote
                                   ;mode.
```

4.2 Program Handling Commands 4.2

This group of commands comprises SUBMIT, XSUB, ASM, LOAD, DUMP, DDT, and SAVE.

4.2.1 SUBMIT 4.2.1

Standard transient command,
the form of which is: SUBMIT ufn p1 p2 .. pn cr

The SUBMIT command allows CP/M commands (including SUBMIT itself) to be batched together for automatic processing.

The unambiguous filename, ufn, specifies the primary name of a file on the disk in the currently logged drive. The secondary name of this file is assumed to be SUB and is therefore not specified. A file with the SUB extension contains CP/M prototype commands, with possible parameter substitution.

SUB files are created using ED (see subsection 4.1.8) with interspersed formal parameters of the form

```
   $1 $2 .. $n
```

corresponding to the number of actual parameters to be included when the SUB file is submitted for execution.

When the SUBMIT program is executed, it substitutes the actual parameters p1 p2 .. pn in the command line for the formal parameters $1 $2 .. $n in the prototype commands of the SUB file.

SUBMIT creates a temporary file of substituted commands named $$$.SUB on the disk in the currently logged drive. When the system reboots on the termination of SUBMIT, the CCP reads $$$.SUB as a source of input, rather than the console.

SUBMIT allows the user to submit a file of programs for execution. This facility is extended by XSUB, which provides input for such programs (see subsection 4.2.2).

Example

```
DIR $1.*                      ;prototype commands in the file
PIP $2:=$1.BAK                ;WEE.SUB
ERA $1.BAK

A>SUBMIT WEE PROG B cr        ;WEE.SUB submitted with actual
                              ;parameters
```

```
DIR PROG.*                      ;substituted commands in the file
PIP B:=PROG.BAK                 ;$$$.SUB, which are executed
ERA PROG.BAK                    ;in sequence by the CCP
```

See further ref. [1], 1.6.7, pp. 25-27.

4.2.2 XSUB 4.2.2

Standard transient command,
the form of which is: XSUB

The XSUB program extends the SUBMIT facility (see subsection 4.2.1) to provide input to programs executed in the SUB file.

When XSUB is included as the first command line of a SUB file, it relocates to the memory area directly below the CCP in order to process the remaining command lines of the SUB file, and thereby provides buffered console input to the programs executed within the submit operation. The programs that read such input, and can therefore receive their input directly from the SUB file, include PIP, ED, and DDT. If a program within the SUB file performs a system reboot (warm boot), the message '(xsub active)' is displayed to indicate the presence of XSUB. XSUB remains active until the SUB file is exhausted or a system boot (cold boot) is performed.

Example

```
XSUB                            ;prototype commands in the file
DDT                             ;NOVUS.SUB
I$1.HEX
R
GO
SAVE 1 $2.COM

A>SUBMIT NOVUS HIC ILLE cr      ;NOVUS.SUB submitted with actual
                                ;parameters
```

```
                              ;substituted commands in the file
                              ;$$$.SUB, which are executed
                              ;in sequence by the CCP:

XSUB                          ;XSUB is loaded
DDT                           ;DDT is loaded
IHIC.HEX                      ;input to DDT from XSUB
R                             ;input to DDT from XSUB
G0                            ;input to DDT from XSUB
SAVE 1 ILLE.COM               ;input to the CCP from XSUB
```

See further ref. [1], 1.6.7, p. 26.

4.2.3 ASM 4.2.3

Standard transient command,
the form of which is: ASM ufn cr

The ASM command loads and executes the CP/M 8080 assembler. The unambiguous filename, ufn, specifies the primary name of a source file containing assembly language statements. The secondary name of this file is assumed to be ASM and is therefore not specified.

The assembler generates a file of the type PRN, which contains a listing for the user with diagnostics, and a file of the type HEX, which contains "hex" format machine code (see LOAD, below).

Example

```
A>ASM B:ALPHA cr              ;specifies that the source file is
                              ;ALPHA.ASM on the disk in drive B,
                              ;and that ALPHA.PRN and ALPHA.HEX
                              ;shall be placed on the same disk
```

See further ref. [1], 1.6.2, pp. 15-16, and particularly ref. [1], chapter 3 pp. 47-68.

4.2.4 LOAD 4.2.4

Standard transient command,
the form of which is: LOAD ufn cr

The LOAD command causes generation of a memory-image file, of the type COM, which can subsequently be executed as a transient program. The unambiguous filename, ufn, specifies the primary name of an object file containing "hex" format machine code (see ASM, above). The secondary name of this file is assumed to be HEX and is therefore not specified.

Example

```
A>LOAD B:ALPHA cr                   ;specifies that the file ALPHA.HEX
                                    ;shall be read from the disk in
                                    ;drive B, and the file ALPHA.COM
                                    ;placed on the disk in drive A

A>ALPHA cr                          ;loads the file ALPHA.COM into the
                                    ;TPA and executes the code
```

See further ref. [1], 1.6.3, p. 16.

4.2.5 DUMP 4.2.5

Standard transient command,
the form of which is: DUMP ufn cr

The DUMP command causes the file specified by the unambiguous filename, ufn, to be displayed at the console. The contents of the file are listed in hexadecimal (base 16) form, sixteen bytes at a time, with the absolute byte address of each line at the left in hexadecimal.

Note that an "H" is often used in descriptions to denote the 16 subscript (e.g. 10H stands for 10_{16}).

Note also that console output can be copied to the list device by pressing the PRINT key.

Examples

```
A>DUMP ANON.COM cr             ;displays the contents of the file
                               ;ANON.COM

A>DUMP B:SCRATCH.HEX cr        ;displays the contents of the file
                               ;SCRATCH.HEX on the disk in drive
                               ;B
```

See further ref. [1], 1.6.8, p. 27.

4.2.6 DDT 4.2.6

Standard transient command,
the form of which is: DDT ufn cr

The DDT command loads and executes the CP/M debugger, which can be used to load, alter, and test programs or to load a file in order to save a memory image of it (see SAVE, below). The unambiguous filename, ufn, specifies the file, of the type COM, HEX, or SYS, on which DDT is to operate. The secondary name must be specified.

The DDT program temporarily replaces the CCP in memory, and loads the specified file into the TPA. DDT has its own commands for inserting values, displaying memory locations, saving comments, setting breakpoints, and other debugging functions.

DDT also displays, in hexadecimal, the NEXT address (following the last address in the loaded file) and the PC (program counter) address.

Example

```
A>DDT CBIOS.SYS cr                ;loads DDT, which in turn loads
DDT VERS 2.2                      ;the file CBIOS.SYS from the disk
NEXT  PC                          ;in drive A, then displays the
2000  0100                        ;NEXT and PC values, and finally
                                  ;prompts the console with a dash
                                  ;(-)
```

See further ref. [1], chapter 4 pp. 69-87.

4.2.7 SAVE 4.2.7

Built-in command,
the form of which is: SAVE n ufn cr

The SAVE command creates a memory-image file, specified by the unambiguous filename, ufn, of whatever is currently in the TPA (from memory address 0100H). The decimal number n specifies the number of TPA pages to be placed on the disk.

A page is a block of 256 bytes. The number of pages, n, can be calculated as follows:

1) Bring the relevant file into the TPA by means of DDT (see DDT, above) and note the NEXT value.

2) If the NEXT value ends in two zeroes, subtract 1 hexadecimal, and convert the first two digits to decimal. For example:

   NEXT is 1200.
   1200 - 1 = 11FF
   11 hexadecimal is 17 decimal (i.e. 17 pages).

3) If the NEXT value does not end in two zeroes, convert the first two digits to decimal. For example:

   NEXT is 1205.
   12 hexadecimal is 18 decimal (i.e. 18 pages).

Examples

```
A>SAVE 31 CBIOS.SYS cr          ;saves memory from 0100H through
                                ;1FFFH (NEXT was 2000) and places
                                ;it in the file CBIOS.SYS on the
                                ;disk in drive A

A>SAVE 10 B:PROG.COM cr         ;saves memory from 0100H through
                                ;0AFFH (NEXT was 0B00) and places
                                ;it in the file PROG.COM on the
                                ;disk in drive B
```

See further ref. [1], 1.4.4, p. 8.

4.3 Device Handling Commands 4.3

This group of commands comprises FORMAT, BACKUP, ASSIGN, GETBIOS, WRTBIOS, SYSGEN, CONFI, USER, COPYSYS and VERIFY.

4.3.1 FORMAT 4.3.1

RC855 transient command,
the form of which is: FORMAT cr

The FORMAT program is used to format diskettes on a one-drive or a two-drive system. FORMAT will erase any and all data on the diskette. Several diskettes can be formatted in succession.

The diskettes must be soft-sectored 8-inch diskettes, either single-sided, single-density or double-sided, double-density. Two formats are available:

1 = SS/SD (single-sided, single-density, 128 bytes per sector)
2 = DS/DD (double-sided, double-density, 512 bytes per sector)

CP/M System Diskettes have format 2. The diskette formats are further described in appendix E.

FORMAT prompts the console for commands, with interaction as shown below. A system reboot (warm boot) can be performed whenever the program is requesting input.

Example

```
A>FORMAT cr
RC855 FORMAT UTILITY VERS 1.2 82.03.03
FORMAT: 1=SS/SD 2=DS/DD TYPE (1,2) ? 2
SELECT DRIVE (A/B) ? B
INSERT DISK AND TYPE RETURN cr
FORMAT DISKETTE IN DRIVE B (Y/N) ? Y
NUMBER OF FORMATTED TRACKS 077
FORMAT COMPLETED
TYPE T TO TERMINATE
TYPE C TO CONTINUE T
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
```

4.3.2 BACKUP 4.3.2

RC855 transient command,
the form of which is: BACKUP cr

The BACKUP program is used to copy an entire 8-inch diskette on a one-drive or a two-drive system. The diskettes may be single-sided, single-density (128 bytes per sector) or double-sided, double-density (512 bytes per sector). The diskette formats are further described in appendix E.

Factory-fresh diskettes are usually preformatted, but it is prudent to format them before use by means of the FORMAT program (see subsection 4.3.1), as this will improve their reliability.

BACKUP prompts the console for commands, with interaction as shown below. A system reboot (warm boot) can be performed whenever the program is requesting input.

Example

```
A>BACKUP cr
RC855 BACKUP VERS 1.0 81.12.17
SOURCE DRIVE (A OR B) ? A
DESTINATION DRIVE (A OR B) ? B
INSERT SOURCE DISK AND TYPE RETURN cr
INSERT DESTINATION DISK AND TYPE RETURN cr
BACKUP COMPLETE
INSERT CP/M SYSTEM DISK AND TYPE RETURN cr
A>
```

Recovery from read or write errors that may occur during copy operations is described in chapter 5.

4.3.3 ASSIGN 4.3.3

RC855 transient command,
the form of which is: ASSIGN d:=ff cr

The ASSIGN command assigns a format to the 8-inch diskette drive specified by the drive name, d, which may be "A" or "B". The specified format, ff, may be "SS" (single-sided, single-density) or "DD" (double-sided, double-density). The diskette formats are described in appendix E.

Example

```
A>ASSIGN B:=SS cr
INSERT DISK AND TYPE RETURN cr
A>
;the system is now ready for use with a single-sided, single-
;density diskette in drive B
```

4.3.4 GETBIOS

RC855 transient command,
the form of which is: GETBIOS cr

The GETBIOS command loads and executes the GETBIOS program, which copies the BIOS part of CP/M from its permanent memory location to the TPA. The BIOS can then be copied from the TPA to a file (CBIOS.SYS) by means of the SAVE command (see subsection 4.2.7). The file containing the BIOS can subsequently be read by WRTBIOS as a source file (see WRTBIOS, below).

Note that the SYSGEN program is used to get the BDOS and CCP parts of CP/M from a System Diskette and write them to another (see subsection 4.3.6).

The use of GETBIOS, WRTBIOS, and SYSGEN to create a new System Diskette is explained in appendix C.

Example

```
A>GETBIOS cr
RC855 GETBIOS VERS 1.0 81.12.15
- FUNCTION COMPLETE -
BIOS READY FOR: "SAVE 31 CBIOS.SYS"
A>SAVE 31 CBIOS.SYS cr
A>
```

4.3.5 WRTBIOS

RC855 transient command,
the form of which is: WRTBIOS ufn cr

The WRTBIOS command loads and executes the WRTBIOS program, which is used either to test a customized version of the BIOS or to write a customized version of the BIOS to the reserved tracks of a new or existing System Diskette. The unambiguous filename, ufn, specifies a file of the type SYS, created by GETBIOS and SAVE, which contains a customized version of the BIOS (see GETBIOS, above).

A System Diskette, containing the BIOS, BDOS, and CCP, must be present in drive A while the WRTBIOS program is in use.

Note that the SYSGEN program is used to get the BDOS and CCP parts of CP/M from a System Diskette and write them to another (see subsection 4.3.6).

The use of GETBIOS, WRTBIOS, and SYSGEN to create a new System Diskette is explained in appendix C, which also gives an example of how the BIOS is customized, or "patched", by means of DDT (see subsection 4.2.6).

Examples

```
A>WRTBIOS CBIOS.SYS cr
RC855 WRTBIOS VERS 1.1 82.10.13
READ - NO. OF RECORDS:  003E                           ;62 decimal
CBIOS READY FOR TEST/WRITE - SELECT FUNCTION: T TEST SELECTED
     TEST...... TYPE T
     WRITE..... TYPE W
     FINISH.... TYPE F
PATCHED VERSION READY FOR TEST
RC855 56k CP/M 2.2 rel. 1.3
A>
```

```
A>WRTBIOS CBIOS.SYS cr
RC855 WRTBIOS VERS 1.1 82.10.13
READ - NO. OF RECORDS:  003E
CBIOS READY FOR TEST/WRITE - SELECT FUNCTION: W WRITE SELECTED
     TEST...... TYPE T
     WRITE..... TYPE W
     FINISH.... TYPE F
SELECT DESTINATION DRIVE (A OR B): B
INSERT DISKETTE - DRIVE B - TYPE RETURN cr
PATCHED BIOS WRITTEN ON B - INSERT SYSTEM DISK AND TYPE RETURN cr
RC855 WRTBIOS VERS 1.1 82.10.13
```

```
READ - NO. OF RECORDS: 003E
CBIOS READY FOR TEST/WRITE - SELECT FUNCTION: F FINISH SELECTED
    TEST...... TYPE T
    WRITE..... TYPE W
    FINISH.... TYPE F
WRTBIOS TERMINATED
A>
```

4.3.6 SYSGEN 4.3.6

RC855 transient command,
the forms of which are: SYSGEN cr
SYSGEN ufn cr

The SYSGEN command is used, in the RC855 implementation of CP/M, to get the BDOS and CCP parts of CP/M, either from the reserved tracks of a System Diskette or from a file, and to write the BDOS and CCP to the reserved tracks of a new or existing System Diskette.

The SYSGEN program first reads the BDOS and CCP into the TPA.

The BDOS and CCP may now be copied from the TPA to a file (CPM56.COM) by means of the SAVE command (see subsection 4.2.7). The file containing the BDOS and CCP can subsequently be read by SYSGEN as a source file.

The SYSGEN program then writes the BDOS and CCP from the TPA to the diskette in the specified drive.

Note that the utility programs GETBIOS and WRTBIOS are used to get and write the BIOS part of CP/M (see subsections 4.3.4 and 4.3.5).

The use of GETBIOS, WRTBIOS, and SYSGEN to create a new System Diskette is explained in appendix C.

Example

```
A>SYSGEN cr
SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP) A
SOURCE ON A, THEN TYPE RETURN cr                     ;read operation
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) B
DESTINATION ON B, THEN TYPE RETURN cr               ;write operation
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) cr
A>
```

See further ref. [1], 1.6.6, pp. 24-25.

4.3.7 CONFI 4.3.7

RC855 transient command,
the form of which is: CONFI cr

The CONFI program enables the user to change the values of system configuration parameters stored in the nonvolatile memory of the RC855 Work Station. These parameters are described in detail in appendix G.

Example

```
A>CONFI cr
RC855              WORK STATION CONFIGURATOR          MENU C03


                         KBL :      0     alfalock
                         CP  :      2     cursor
                         PLS :      3     printer line speed
                         CF  :     79
                         LM  :      0
                         LS  :      3
                         LCF :     79
                         LE  :      0
                         SA  :      1
                         DPC :      0
                         NFR :      0
                         NUM :      0
                         MPL :     80
                         DDN :      0
                         PDN :      0
                         HCP :      0
                         PM  :      0
                         CU  :      0
                         CS  :      0
                         PDL :      0
                         BRM :      0
                         PTM :      0
```

Operation

1) Move the cursor (using, for example, the keys marked with arrows) to the displayed parameter value.

2) Enter the new value.

3) When all changes have been made in this manner, press the PRINT key to obtain a hard copy of the display screen image.

4) Press the PA1 or SEND key to terminate the CONFI program and return to the CP/M system.

Note that a "***" displayed anywhere on the screen indicates an illegal parameter value. Should this occur, check appendix G for legal values, re-enter the new value, and press PA1 or SEND once more.

4.3.8 USER 4.3.8

Built-in command,
the form of which is: USER n cr

The USER facility allows the maintenance of up to 15 logical user areas within the same directory. The USER command moves the user to the area specified by the decimal number n.

All of the files on the System Diskette supplied for the RC855 Work Station are in user area 0.

On a system boot (cold boot) the user is automatically logged in to user area 0, which will remain the current user area, if he does not employ the USER command.

See further ref. [1], 1.4.6, p. 8.

4.3.9 COPYSYS 4.3.9

RC855 transient command,
the form of which is: COPYSYS cr

The COPYSYS program is used to create a new system diskette. COPYSYS may be used on a one or two drive system to copy the operating system (CCP, BDOS and BIOS) and whatever files the user specifies. The diskette must be double-sided, double-density (512 bytes per sector).

The files to be copied must be specified in a text-file named SYSTEM.SYS. A text editor (e.g. ED) may be used to alter the text file to suit the needs of the individual user. When altering the file, the user should observe the following rules:

- Only unambiguous filenames are allowed.
- The text file may only contain one filename per line.
- No more than 45 filenames are allowed.

COPYSYS prompts the console for commands, with interaction as shown below. A system reboot (warm boot) can be performed by typing CTRL C whenever the program is requesting input.

Example

```
A> COPYSYS cr
RC855 COPYSYS vers. 1.0 83.01.07
Select source drive (A/B) A
Select destination drive (A/B) B
Insert SOURCE DISK and type RETURN cr
Insert DESTINATION DISK and type RETURN cr
COPYSYS completed - Insert CP/M disk and type RETURN cr
A>
```

Note: If one of the files to be copied already exists on the destination disk, the file will be overwritten unless it has R/O status (see STAT subsection 4.1.5). In the latter case a warning is printed on the console and the file is not copied.

If there is no text-file with the name SYSTEM.SYS on the system diskette, COPYSYS will copy the operating system (CCP, BDOS and BIOS) and notify the user that it cannot find SYSTEM.SYS.

4.3.10 VERIFY 4.3.10

RC855 transient command,
the forms of which are: VERIFY cr
VERIFY d: cr

where "d" may be the drive name "A" or "B".

The VERIFY command is used to check a disk for bad sectors. If any bad sectors are found, VERIFY will indicate the total number of blocks in which they appear and then display the following prompt:

```
   CREATE DUMMY FILE (Y/N):
```

DO NOT TYPE Y as typing Y may cause whatever files you have on the disk to be ruined. Instead copy any files worth saving onto another disk, reformat the bad disk (see 4.3.1) and try VERIFY on it once again.

If VERIFY again finds bad sectors, answer "CREATE DUMMY FILE (Y/N):" by typing Y. VERIFY will then create a dummy file called BLOCKS.BAD making it possible to use the disk inspite of the bad sectors.

The user should take care not to erase BLOCKS.BAD (i.e. never use ERA *.*). If erased, it must be created again.

Example

```
A> VERIFY cr
   RC855 DISKETTE VERIFICATION VERS. 1.0 82.10.12
   SELECT DRIVE: A
   INSERT FORMATTED DISKETTE IN DRIVE A AND TYPE RETURN cr
   CHECK READING CP/M BLOCK No 561
   NO BAD SECTORS FOUND
   INSERT CP/M SYSTEM DISKETTE IN DRIVE A AND TYPE RETURN cr
A>
```

```
A> VERIFY B: cr
   RC855 DISKETTE VERIFICATION VERS. 1.0 82.10.12
   INSERT FORMATTED DISKETTE IN DRIVE B AND TYPE RETURN cr
   CHECK READING CP/M BLOCK No 561 (BAD SECTOR ENCOUNTERED)
   7 BLOCKS FOUND WITH BAD SECTORS
   CREATE DUMMY FILE (Y/N): Y  ; Note that pressing Y may delete
                               ; existing file
   CREATED DUMMY FILE 'BLOCKS.BAD'
```

```
    14 Kbytes LOST
    INSERT CP/M SYSTEM DISKETTE IN DRIVE A AND TYPE RETURN cr
A>
```

5. ERROR RECOVERY 5.

Four error conditions are common to the system. All of them are reported through the general error message

```
BDOS ERR ON d:message
```

where "d" is the disk drive name, and "message" is one of the following:

```
BAD SECTOR
SELECT
READ ONLY
FILE R/O
```

For descriptions of PIP, ED, and other program errors, see ref. [1], appendix I, pp. 235-244.

5.1 BAD SECTOR Error 5.1

This error will occur if the disk drive controller cannot read or write data on the diskette, which will happen

- if the diskette is worn,
- if the controller is malfunctioning,
- if the diskette was missing from the drive when the user attempted to access it, or
- if data in a file has been damaged through diskette mishandling or by a damaged and erroneous program.

This error may also occur if the user attempts to read files that were placed on the diskette by a different controller than the one which he is now using.

Recovery

Reboot the system (↑C). This will abort the program or file processing and return control to CP/M.

The user may also choose to ignore the error and continue program execution or file processing by pressing the RETURN key ("cr"). It is unwise to do this, however, without adequate backup copies, for if a directory write operation is involved, the integrity of the diskette may be destroyed.

A Note on COPYSYS and BACKUP

The COPYSYS program will report this error, stop copying, and return control to CP/M.

The BACKUP program will report this error through the message BAD SECTOR ON SOURCE DISK or BAD SECTOR ON DESTINATION DISK. If the user chooses to continue copying, the transferred data will probably be incorrect, and he must then determine by other means whether the copy can be used. It may be possible to correct an error involving the destination diskette by reformatting it and then rerunning BACKUP.

5.2 SELECT Error 5.2

This error will occur if the user selects a non-existent disk drive, indicated by "d".

Recovery

Press the RETURN key ("cr") to reboot the system.

5.3 READ ONLY Error 5.3

This error will occur if the user attempts to write on a diskette that has been assigned the read-only attribute through use of the STAT command (or by a program using the BDOS function).

This error will also occur if the user inserts a new diskette and attempts to write on it without having performed a system reboot (↑C).

Recovery

Press the RETURN key ("cr") and the system will reboot. Note that this will also assign the read-write attribute to the diskette.

5.4 FILE R/O Error 5.4

This error will occur if the user attempts to write in a file that has been assigned the read-only attribute through use of the STAT command (or by a program using the BDOS function).

Recovery

Press the RETURN key ("cr") and the system will reboot. This will abort the operation involving the file. The read-only attribute of the file can be changed to read-write by means of the STAT command.

See further ref. [1], 1.6.1, p. 14, and ref. [1], 1.7, pp. 29-30.

A. REFERENCES

[1] CP/M Operating System Manual, Digital Research, 1982

[2] RCSL No 42-i2042:
RC855 Installation Manual, 1982

[3] RCSL No 42-i1686:
RC855 Work Station Operating Guide, 1982

B. DISPLAY HANDLING B.

The display handling routines supported by the CONOUT procedure of the RC855 BIOS are described in this appendix (see further ref. [1], chapters 5 and 6).

B.1 X-Y Addressing B.1

The control character 6 moves the cursor to the character position (x-y address) defined by the following two characters. Three characters, then, must be sent to the console via CONOUT:

a 6 character, followed by
the vertical position + 32 (20H), followed by
the horizontal position + 32 (20H).

It is the responsibility of the programmer to ensure that the value of the x-y address does not exceed the limits

$0+32 \leq$ vertical position $\leq 24+32$
$0+32 \leq$ horizontal position $\leq 79+32$

The (vertical,horizontal) address of the upper left corner of the display screen is (0,0), and that of the lower right corner is (24,79).

Note that CONOUT, after having received the 6 character, will invariably handle the next two characters as the x-y address.

B.2 Various Functions B.2

Various functions can be performed by sending the following control characters to the console via CONOUT:

1 Inserts a line, at the line position indicated by the cursor, and scrolls the remainder of the screen image down.

2 Deletes a line, at the line position indicated by the cursor, and scrolls the remainder of the screen image up.

5 Moves the cursor one position left (back-space).

8 Performs the same function as the 5 character.

9 Moves the cursor four positions forward.

10 Moves the cursor one position down (line feed).

12 Resets the console display. The screen is erased, the display buffer (including currently assigned attributes) is cleared, and the cursor is moved to its home position (0,0).

13 Moves the cursor to position 0 on the current line (carriage return).

24 Moves the cursor one position right (forward-space).

26 Moves the cursor one position up.

29 Moves the cursor to position (0,0), i.e. the upper left corner of the display screen, which is its "home" position.

30 Erases the current line from the cursor position to the end of the line, i.e. from (x,y) to (x,79).

31 Erases the screen image from the cursor position to the end of the screen, i.e. from (x,y) to (24,79).

B.3 Attributes B.3

A set of attributes is available for each character position on the console display. An attribute is assigned as follows:

1) Address the relevant character position (see section B.1).

2) Send a "set attribute" character to the console via CONOUT. A "set attribute" character is defined as 128 plus the value of the relevant attribute byte (see table below). The attribute will apply until a "reset attribute" character, defined as 128, is sent.

Any of the following attributes can be assigned:

Attribute		Value of Attribute Byte		
		Dec	Hex	Bin
underscore	129	1	01	00000001
intensified display	132	4	04	00000100
nondisplay	136	8	08	00001000
inverse video	144	16	10	00010000
fast flash	160	32	20	00100000
slow flash	224	96	60	01100000

Note that "fast flash" and "slow flash" must be combined with another attribute byte.

Note also that, to avoid confusion with the "printer on/off" character (decimal 16), "inverse video" should be combined with another byte value. We suggest using 16 + 64, that is to say:

binary 00010000 ("inverse video")
plus
binary 01000000 (frequency bit in "slow flash").

C. SYSTEM DISKETTE GENERATION C.

A new System Diskette is normally generated as described in section 2.3. This appendix describes how to generate a new System Diskette that contains a customized, or "patched", version of the BIOS (see further ref. [1], chapter 6, pp. 127-152). The BDOS and CCP can also be patched in an analogous manner.

1) Patch the BIOS (see the example below).

2) Format a diskette for the new System Diskette using FORMAT (see subsection 4.3.1).

3) Copy the existing System Diskette using BACKUP (see subsection 4.3.2) or COPYSYS (see subsection 4.3.9). Then delete the unwanted files using ERA (see subsection 4.1.2).

   Alternatively, transfer the wanted files to the new System Diskette using PIP (see subsection 4.1.6) or TRANSFER (see subsection 4.1.7).

4) Write the BDOS and CCP (possibly using the source file CPM56.COM) to the new System Diskette using SYSGEN (see subsection 4.3.6). Note that this step is superfluous if either BACKUP or COPYSYS was used in step 3.

5) Write the BIOS (using the source file CBIOS.SYS) to the new System Diskette using WRTBIOS (see subsection 4.3.5).

Example

The following example shows how to patch a keyboard conversion table in the BIOS.

One can alter the standard ASCII values generated by the input conversion routine using GETBIOS, SAVE, DDT, WRTBIOS, and the patch addresses for the key positions shown in appendix D.

In the example, we shall patch the PA1 key (key position 3) to generate 97 (61H) in lower case and 65 (41H) in upper case, instead of 32 (20H) in both cases.

```
;get a version of the BIOS
A>GETBIOS cr                              ;see subsection 4.3.4
RC855 GETBIOS VERS 1.0 81.12.15
- FUNCTION COMPLETE -
BIOS READY FOR:  "SAVE 31 CBIOS.SYS"
A>
;save the BIOS
A>SAVE 31 CBIOS.SYS cr                    ;see subsection 4.2.7
A>
;alter the values generated by the PA1 key (key position 3)
;see appendix D:
;patch address for lower-case conversion: 1E08H
;patch address for upper-case conversion: 1E95H
A>DDT CBIOS.SYS cr                        ;see subsection 4.2.6
DDT VER 2.2
NEXT  PC
2000  0100
                 adresse (keyboard)
-S1E08 cr
                 Hex værdi for tegn
-1E08 20 61 cr                            ;lower-case value altered
-1E09 20 (.) cr
                 Slut på rettelse
-S1E95 cr
-1E95 20 41 cr                            ;upper-case value altered
-1E96 20 . cr
-G0 cr
A>
;save the patched BIOS
A>SAVE 31 CBIOS.SYS cr
A>
;continued on next page
```

```
;test the patched BIOS
A>WRTBIOS CBIOS.SYS cr
RC855 WRTBIOS VERS 1.1-82.10.13
READ - NO. OF RECORDS:  003E
CBIOS READY FOR TEST/WRITE - SELECT FUNCTION: T TEST SELECTED
    TEST...... TYPE T
    WRITE..... TYPE W
    FINISH.... TYPE F
PATCHED VERSION READY FOR TEST
RC855 56k CP/M 2.2 rel. 1.3
A>
;press the PA1 and check the result
```

Note on the Use of Conversion ROM's

The conversion tables used by the BIOS are stored in a read-only memory (ROM). The BIOS object code contains the call of a procedure, ROMCONV, which copies the conversion tables from the ROM to the BIOS whenever a system boot (cold boot) occurs from the original System Diskette.

The conversion ROM is not read, however, on a system boot from a System Diskette that contains a version of the BIOS written by the WRTBIOS program. This is because the GETBIOS program, when it copies the BIOS from its permanent memory location to the TPA, removes the call of the ROMCONV procedure in order to prevent the overwriting of subsequently patched conversion tables.

D. RC855 KEYBOARD D.

D.1 National ASCII Character Sets D.1

The RC855 keyboard has a number of variants, corresponding to different national ASCII character sets. The following table indicates, for each RC855 keyboard, the characters which differ from the US ASCII characters assumed by the CP/M system. The keyboards are numbered as follows:

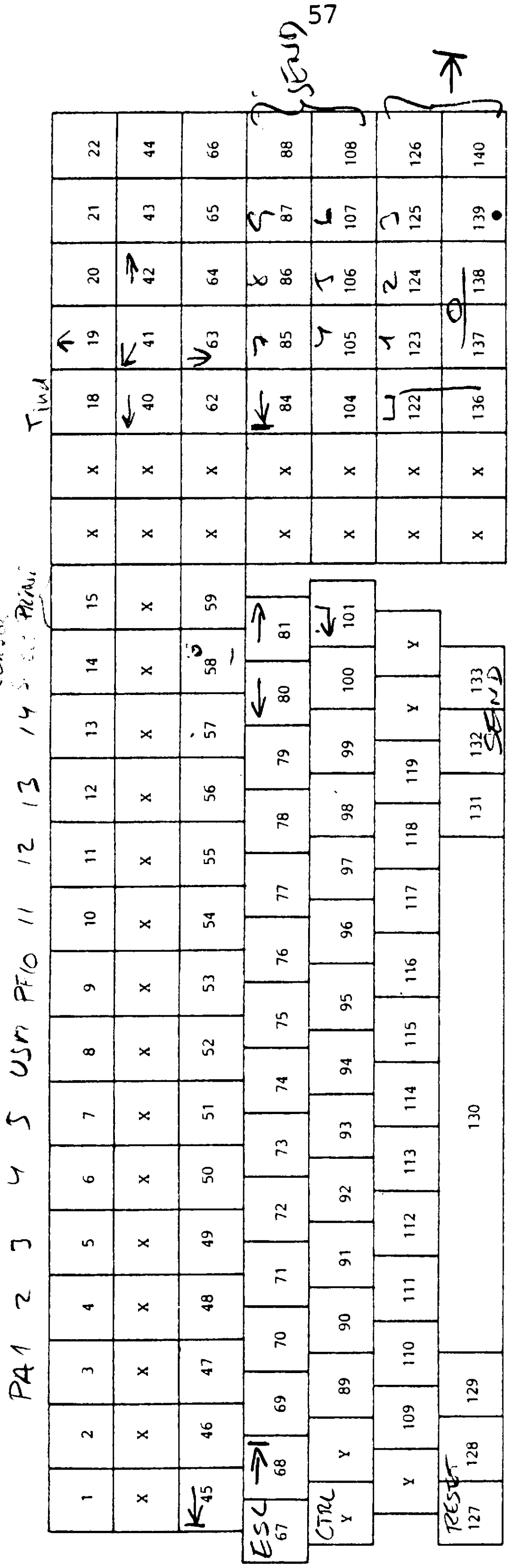
1 = US English	4 = Swedish
2 = UK English	5 = Danish standard
3 = German	6 = Danish public sector

STANDARD ASCII VALUES				RC855 KEYBOARDS					
Bin	Dec	Hex	CP/M	1	2	3	4	5	6
0100011	35	23	#		£				§
1000000	64	40	@			§	É	ü	ö
1011011	91	5B	[			Ä	Ä	Æ	Æ
1011100	92	5C	\			Ö	Ö	Ø	Ø
1011101	93	5D	]			Ü	Å	Å	Å
1011110	94	5E	^	↑	↑	↑	Ü	↑	Ü
1100000	96	60	`				é	ä	↑
1111011	123	7B	{			ä	ä	æ	æ
1111100	124	7C	¦			ö	ö	ø	ø
1111101	125	7D	}			ü	å	å	å
1111110	126	7E	~				ü	ö	ü

D.2 Key Positions D.2

The following figure shows the number of each key position on the RC855 keyboard. An "X" indicates that the key position is not used. A "Y" indicates that the use of the key position is permanently defined.

RC 855


PA1 2 3 4 5 USM PF10 11 12 13 14 CURSOR SELECT PRINT
Find
SEND
ESC
CTRL
RESET
SEND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 X X 18 19 20 21 22
X X X X X X X X X X X X X X X X X 40 41 42 43 44
45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 X X 62 63 64 65 66
67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 X X 84 85 86 87 88
Y Y 89 90 91 92 93 94 95 96 97 98 99 100 101 X X 104 105 106 107 108
Y 109 110 111 112 113 114 115 116 117 118 119 Y Y X X 122 123 124 125 126
127 128 129 130 131 132 133 X X 136 137 138 139 140
7 8 9
4 5 6
1 2 3
0

20: Slet linie
21: Slet felt
22: marker
43: Slet data
44: flyt
62: Slet tegn
64: Linie ind
65: FM
66: DUP

D.3 Patch Addresses D.3

The following table gives, for each numbered key position on the RC855 keyboard, the patch addresses for lower-case (LC) conversion and upper-case (UC) conversion.

KEY	LC	UC
1	1E06	1E93
2	1E07	1E94
3	1E08	1E95
4	1E09	1E96
5	1E0A	1E97
6	1E0B	1E98
7	1E0C	1E99
8	1E0D	1E9A
9	1E0E	1E9B
10	1E0F	1E9C
11	1E10	1E9D
12	1E11	1E9E
13	1E12	1E9F
14	1E13	1EA0
15	1E14	1EA1
18	1E17	1EA4
19	1E18	1EA5
20	1E19	1EA6
21	1E1A	1EA7
22	1E1B	1EA8
40	1E2D	1EBA
41	1E2E	1EBB
42	1E2F	1EBC
43	1E30	1EBD
44	1E31	1EBE
45	1E32	1EBF
46	1E33	1EC0
47	1E34	1EC1
48	1E35	1EC2
49	1E36	1EC3
50	1E37	1EC4
51	1E38	1EC5
52	1E39	1EC6
53	1E3A	1EC7
54	1E3B	1EC8
55	1E3C	1EC9
56	1E3D	1ECA
57	1E3E	1ECB
58	1E3F	1ECC
59	1E40	1ECD
62	1E43	1ED0
63	1E44	1ED1
64	1E45	1ED2
65	1E46	1ED3
66	1E47	1ED4
67	1E48	1ED5
68	1E49	1ED6
69	1E4A	1ED7
70	1E4B	1ED8
71	1E4C	1ED9
72	1E4D	1EDA
73	1E4E	1EDB
74	1E4F	1EDC
75	1E50	1EDD
76	1E51	1EDE
77	1E52	1EDF
78	1E53	1EE0
79	1E54	1EE1
80	1E55	1EE2
81	1E56	1EE3
84	1E59	1EE6
85	1E5A	1EE7
86	1E5B	1EE8
87	1E5C	1EE9
88	1E5D	1EEA
89	1E5E	1EEB
90	1E5F	1EEC
91	1E60	1EED

KEY	LC	UC	KEY	LC	UC
92	1E61	1EEE	116	1E79	1F06
93	1E62	1EEF	117	1E7A	1F07
94	1E63	1EF0	118	1E7B	1F08
95	1E64	1EF1	119	1E7C	1F09
96	1E65	1EF2			
97	1E66	1EF3	122	1E7F	1F0C
98	1E67	1EF4	123	1E80	1F0D
99	1E68	1EF5	124	1E81	1F0E
100	1E69	1EF6	125	1E82	1F0F
101	1E6A	1EF7	126	1E83	1F10
			127	1E84	1F11
104	1E6D	1EFA	128	1E85	1F12
105	1E6E	1EFB	129	1E86	1F13
106	1E6F	1EFC	130	1E87	1F14
107	1E70	1EFD	131	1E88	1F15
108	1E71	1EFE	132	1E89	1F16
109	1E72	1EFF	133	1E8A	1F17
110	1E73	1F00			
111	1E74	1F01	136	1E8D	1F1A
112	1E75	1F02	137	1E8E	1F1B
113	1E76	1F03	138	1E8F	1F1C
114	1E77	1F04	139	1E90	1F1D
115	1E78	1F05	140	1E91	1F1E

E. DISKETTE FORMATS E.

E.1 System Diskette E.1

- o 8-inch diskette
- o double-sided
- o double-density
- o 512 bytes per sector
- o 15 sectors per track
- o 77 tracks
- o recommended type: "3M" 743-0-512

CP/M characteristics:

- o 1124K bytes drive capacity in blocks of 2K bytes
- o 128 directory entries
- o 2 reserved tracks
- o logical sector mapping with 4 to 1 interleaved sectors and zero track-to-track skew

E.2 Files-Only Diskette E.2

- o 8-inch standard file-exchange diskette
- o single-sided
- o single-density
- o 128 bytes per sector
- o 26 sectors per track
- o 77 tracks
- o recommended type: "3M" 740/2-0

CP/M characteristics:

- o 243K bytes drive capacity in blocks of 1K bytes
- o 64 directory entries
- o 2 reserved tracks
- o logical sector mapping with 6 to 1 interleaved sectors and zero track-to-track skew

F. PERIPHERAL SUPPORT

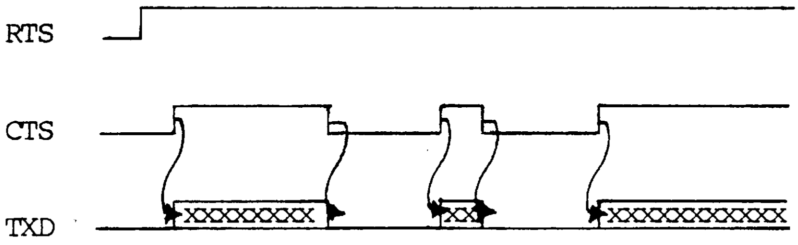
F.1 Printer

F.1.1 Interface

The RC855 printer port is supported as the CP/M ouput list device, LST (see ref. [1], 1.6.1, p. 11).

The configuration parameters for the printer line speed and character format, which are described in appendix G, are taken automatically from the nonvolatile memory whenever a system boot (cold boot) occurs.

The printer port can be used to attach most printers with a serial V.24 interface and busy control. The busy handshake mechanism uses the V.24 signals RTS and CTS as illustrated below:



(Note: Whenever the RC855 attempts transmission via the printer port, RTS ("ready to send") is switched on and sent to the printer. When the printer is ready to start receiving, CTS ("clear to send") is switched on and sent to the RC855. The RC855 can transmit as long as CTS is on. Transmission takes place on line TXD.)

The ready status of the output list device can be accessed by means of the BIOS procedure LISTST (see ref. [1], 6.6, pp. 137-143).

F.1.2 Patch Addresses

Fore each character sent to the list device, the table below supplies the patch address in CBIOS.SYS for the printer conversion table (see 4.3.4 and 4.3.5).

CHARACTER		ADDRESS
Dec.	Hex.	Hex.
0	00	1C05
1	01	1C06
2	02	1C07
3	03	1C08
4	04	1C09
5	05	1C0A
6	06	1C0B
7	07	1C0C
8	08	1C0D
9	09	1C0E
10	0A	1C0F
11	0B	1C10
12	0C	1C11
13	0D	1C12
14	0E	1C13
15	0F	1C14
16	10	1C15
17	11	1C16
18	12	1C17
19	13	1C18
20	14	1C19
21	15	1C1A
22	16	1C1B
23	17	1C1C
24	18	1C1D
25	19	1C1E
26	1A	1C1F
27	1B	1C20
28	1C	1C21
29	1D	1C22
30	1E	1C23
31	1F	1C24
32	20	1C25
33	21	1C26
34	22	1C27
35	23	1C28
36	24	1C29
37	25	1C2A
38	26	1C2B
39	27	1C2C
40	28	1C2D
41	29	1C2E
42	2A	1C2F
43	2B	1C30
44	2C	1C31
45	2D	1C32
46	2E	1C33
47	2F	1C34
48	30	1C35
49	31	1C36
50	32	1C37
51	33	1C38
52	34	1C39
53	35	1C3A
54	36	1C3B
55	37	1C3C
56	38	1C3D
57	39	1C3E
58	3A	1C3F
59	3B	1C40
60	3C	1C41
61	3D	1C42
62	3E	1C43
63	3F	1C44
64	40	1C45
65	41	1C46
66	42	1C47
67	43	1C48
68	44	1C49
69	45	1C4A
70	46	1C4B
71	47	1C4C

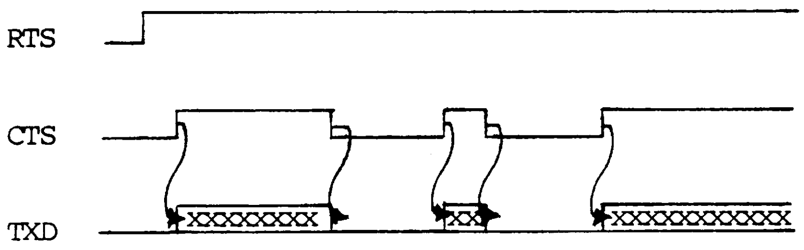
CHARACTER		ADDRESS
Dec.	Hex.	Hex.
72	48	1C4D
73	49	1C4E
74	4A	1C4F
75	4B	1C50
76	4C	1C51
77	4D	1C52
78	4E	1C53
79	4F	1C54
80	50	1C55
81	51	1C56
82	52	1C57
83	53	1C58
84	54	1C59
85	55	1C5A
86	56	1C5B
87	57	1C5C
88	58	1C5D
89	59	1C5E
90	5A	1C5F
91	5B	1C60
92	5C	1C61
93	5D	1C62
94	5E	1C63
95	5F	1C64
96	60	1C65
97	61	1C66
98	62	1C67
99	63	1C68
100	64	1C69
101	65	1C6A
102	66	1C6B
103	67	1C6C
104	68	1C6D
105	69	1C6E
106	6A	1C6F
107	6B	1C70
108	6C	1C71
109	6D	1C72
110	6E	1C73
111	6F	1C74
112	70	1C75
113	71	1C76
114	72	1C77
115	73	1C78
116	74	1C79
117	75	1C7A
118	76	1C7B
119	77	1C7C
120	78	1C7D
121	79	1C7E
122	7A	1C7F
123	7B	1C80
124	7C	1C81
125	7D	1C82
126	7E	1C83
127	7F	1C84

F.2 Line Port

The RC855 line port is supported as the CP/M reader/punch devices, RDR and PUN (see ref. [1], 1.6.1, p. 11).

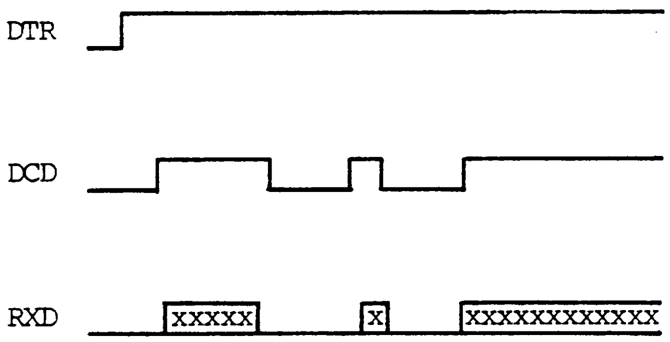
The configuration parameters for the line speed and character format, which are described in appendix G, are retrieved automatically from the nonvolatile memory whenever a system boot (cold boot) occurs.

It is possible to transmit data from the RC855 using the line port. The busy handshake mechanism uses the V.24 signals, RTS and CTS, as illustrated below:



(Note: Following a cold boot, RTS ("ready to send") is switched on and sent by the RC855 to the receiving device. When the receiving device is ready, CTS ("clear to send") is switched on and sent to the RC855. The RC855 can transmit as long as CTS is on. Transmission takes place on line TXD.)

The RC855 can also receive data via the line port. The V.24 signals DTR and DCD are exchanged as illustrated below:



(Note: Whenever a cold boot takes place, DTR ("data terminal ready") is switched on and sent by the RC855 to the transmitting device. The RC855 receiver is not enabled, however, until DCD ("data carrier detect") has been sent to the RC855 by the transmitting device. RXD is the line on which the RC855 will receive data from the transmitting device.)

The ready status of the reader can be accessed by means of the BIOS procedure READST (entry at BIOS+77 (4D hex)). The status is returned in register A (255 (FF hex) if a character is available).

G. SYSTEM CONFIGURATION PARAMETERS G.

This appendix describes the system configuration parameters, the values of which can be changed by means of the utility program CONFI (see subsection 4.3.7).

KBL Keyboard Lock

The KBL parameter determines whether the central keyboard is to be placed in the alpha-lock mode following a system boot (cold boot). In this mode, the SHIFT key function is applied automatically to all alphameric keys marked with (capital) alphabetic characters.

Value range: 0-1.

0: No alpha-lock (indicator on LOCK key off).

1: Alpha-lock (indicator on LOCK key on).

CP Cursor Presentation

The CP parameter defines the form of the cursor.

Value range: 0-3.

0: Underscore.

1: Flashing underscore.

2: Block.

3: Flashing block.

PLS Printer Line Speed

The PLS parameter defines the bit rate of transmission to a printer attached via a serial V.24 interface (see appendix F).

Value range: 0-6.

0:	110 bps.
1:	300 -
2:	600 -
3:	1200 -
4:	2400 -
5:	4800 -
6:	9600 -

CF Character Format

The CF parameter defines the format of characters transmitted to a printer attached via a serial V.24 interface (see appendix F).

Value range: 0-255.

The parameter value is the decimal value of a byte coded, from the least to the most significant bit position, as follows:

0:	when 0:	no parity
	when 1:	parity
1:	when 0:	odd parity
	when 1:	even parity
3-2:	when 01:	1 stop bit
	when 10:	1 1/2 stop bits
	when 11:	2 stop bits
5-4:	dummy	
7-6:	when 00:	5 bits/character
	when 01:	7 bits/character
	when 10:	6 bits/character
	when 11:	8 bits/character

The recommended parameter value for printers supplied by RC is 79 (binary: 01001111), i.e. even parity and 2 stop bits.

LM Line Mode

The LM parameter determines whether the line port is to operate in asynchronous or synchronous mode.

Value range: 0-1.

0: Asynchronous operation.

1: Synchronous operation.

LS Line Speed

The LS parameter defines the bit rate for the line port serial V.24 interface. The value range is as for PLS above.

LCF Line Character Format

The LCF parameter defines the format of characters received or transmitted via the line port serial V.24 interface. The value range is as for CF above.

SA Secondary Address

The SA parameter defines a unique "secondary address" for the CIRCUIT (which can interconnect as many as eight RC855 terminals in a cluster).

Value range: 0-7.

The remaining parameters displayed by the CONFI program are application dependent.

H. MORE ABOUT FILEX

Appendix H provides technical information concerning the FILEX utility.

H.1 describes various requirements for normal use.

H.2 describes FILEX transactions and the FILEX transmission protocol. It is intended for advanced users wanting to modify FILEX or implement a FILEX type file transfer program.

H.1 REQUIREMENTS FOR NORMAL USE

The two computers on which FILEX is to run must be connected by means of the appropriate cable.

To connect two RC700's, two RC855's, or one of each, one of the following cables should be used:

1) CBL912	(5 meters)
2) CBL913	(12 m)
3) CBL914	(25 m)

To connect an RC700 and the RC791 Line Selector, one of the following cables should be used:

1) CBL892	(5 m)
2) CBL893	(12 m)
3) CBL894	(25 m)

Furthermore, the user should use CONFI (4.3.7) to ensure:

1) that the two computers have the same baud rate for the line port,
2) that the line mode is set to asynchronous (LM=0),
3) that the line character format is set to 7 bits per character (LCF=79).

H.2 How FILEX Works

FILEX type file transfer takes place as follows: The local station sends a number of transactions to the remote station. Each time the remote station receives a transaction, it carries out the appropriate file operation and sends an answer back to the local station. The transactions sent depend upon whether the file is to be transferred to or from the local station (see the FILEX program listing for details).

The entire set of transactions is described in H.2.1 below.

H.2.2 explains the transmission protocol for transactions.

H.2.1 FILEX Transactions

(N.B. The effect of the file operations below is as described in ref. [1], section 5.2.)

OPEN

Request	Field	Answer
1	opcode	1
0	unused	0
0	result	result
file name (16 bytes)	name	

(N.B. For all answers, the possible result values are:

0 =ok

1 =does not exist

2 =full

3 =end of file .)

MAKE

Request	Field	Answer
2	opcode	2
0	unused	0
0	result	result
file name		
(16 bytes)	name	

READ

Request	Field	Answer
3	opcode	3
0	unused	0
0	result	result
		area
	area	(128 bytes)

WRITE

Request	Field	Answer
4	opcode	4
0	unused	0
0	result	result
area		
(16 bytes)	area	

CLOSE

Request	Field	Answer
5	opcode	5
0	unused	0
0	result	result

END

Request	Field	Answer
6	opcode	6
0	unused	0
0	result	result

H.2.2 TRANSMISSION PROTOCOL

The transactions described in H.2.1 are sent by means of the blocked transmission protocol described below:

A block consists of the following elements:

1) Start character:
ASCII VALUE 35.

2) Block size:
The size defines the number of characters (N) in the string to be sent, not the number of characters necessary to send the string (2*N+8, explained below).

The block size is a 16-bit integer (0..65535) split into four 4-bit digits. Each digit is interpreted as an integer to which 64 has been added, such that the resulting value lies between 64 and 79. These values are transmitted as characters, the most significant part first, the least significant last.

3) Data section:
Each character in the string to be sent is split into two 4-bit digits, to which 64 is added, as above. These two integers are transmitted as ASCII values, the most significant part first.

4) Checksum:
An 8-bit number which is transmitted as two ASCII values as explained above. The checksum is calculated such that the following condition is satisfied:

((the sum of the values of the characters in the original string) + checksum) modulo 256 = 0.

5) Stop character:
ASCII value 13.

If the number of characters in the string to be transmitted is N, then the actual number of characters transmitted is:

1 (start character)
+
4 (block size)
+
2*N (data section)
+
2 (check sum)
+
1 (stop character)

= 2*N+8 characters.

I. INDEX I.

RETURN LETTER

Title: RC855 Work Station
User's Guide

RCSL No.: 42-i2185

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

____

____

____

____

Do you find errors in this manual? If so, specify by page.

____

____

____

____

How can this manual be improved?

____

____

____

____

Other comments?

____

____

____

____

____

Name: ______________________ Title: ______________________

Company: ____

Address: ____

Date: __________

Thank you

. . . . Fold here . . . .

. . . . Do not tear - Fold here and staple . . . .

Affix
postage
here

A/S REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark