CP/M for the RC702 Microcomputer System
User's Guide

A/S REGNECENTRALEN af 1979
Information Department

January 1983
RCSL No 42-i2190

Authors: Jeffrey C. Snider, Vibeke Nielsen

Keywords: RC702, Microcomputer System, CP/M Operating System, Installation, Operation.

Abstract: This manual describes how to install and operate the CP/M-system on the RC702 Microcomputer System.

(90 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

FOREWORD

Second edition: RCSL No 42-i2190.
This publication is a revision of the publication CP/M for the RC702 Microcomputer System, User's Guide (RCSL No 42-i2131).

The present User's Guide gives a brief description of the CP/M operating system available for the RC700 Microcomputer. Those wishing a complete description of CP/M should consult the Digital Research manuals listed in appendix A (ref. [1] through [7]). The User's Guide also contains reference material for the RC700 implementation of CP/M, which, besides the RC700 BIOS, includes several additional utility programs.

The major difference between first and second edition of this manual is the description of the capacity of the 5 1/4" disks.

Furthermore two new utilities are included (FILEX, for file routing between two CP/M machines and AUTOEXEC for automatically execution of a program after boot) and the CAT utility has been extended.

Jeffrey C. Snider, Vibeke Nielsen
A/S REGNECENTRALEN af 1979, January 1983

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

APPENDICES:

TABLE OF CONTENTS (continued) PAGE

TABLE OF CONTENTS (continued) PAGE

1. INTRODUCTION

The interface between the RC700 Microcomputer 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 microcomputer system. Specifically:

- CP/M creates an orderly and consistent input/output environment for the various units of the microcomputer system 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.

CP/M and the RC700 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 RC700 Microcomputer, 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 RC700 CP/M

The CP/M system supplied for the RC700 Microcomputer system is a 56 K CP/M version 2.2 with 50 3/4 K bytes available for transient programs. The system supports either one or two 8-inch diskette drives, each with a capacity of 900 K bytes, or one or two 5 1/4 inch diskette drives, each with a capacity of 270 K bytes. In addition, the system now supports a hard disk with a capacity of 8,142 - 8,820 K bytes. The diskette formats are described in appendix E, and the hard disk is described in appendix H.

In the rest of the manual disk capacities are only stated with one decimal.

The CP/M disk drives A and B (ref. [1], 1.2, p. 3) are supported as the RC700 disk drives 1 and 2, respectively. The CP/M disk drives C - G are supported on the hard disk. The RC700 printer port is supported as the CP/M output list device, LST (ref. [1], 1.6.1, p. 11). The RC700 Terminal Port is supported as the CP/M reader and punch device (RDR: and PUN:). The RC700 Parallel Input/Output Port is currently not supported. 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).

2. INSTALLING CP/M 2.

The CP/M package for the RC700 Microcomputer contains:

a. This manual.

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

c. An 8-inch or 5 1/4-inch diskette containing your CP/M system.

d. Software License Agreement and Registration Card.

Your copy of the CP/M system for the RC700 is provided with a serial number and is licensed for your use only on a single RC700 Microcomputer. 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 RC700 keyboard.

2.2 How to Bring Up the System 2.2

The following procedure will load the CP/M system from disk into the RC700 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) and close the lid.
- Push the "RESET" button.
- Wait for the following sign-on message to be displayed:

```
RC700 56K CP/M vers.2.2 rel.2.1
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
RC700 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>
```

Please note that if the diskette drives were the 5 1/4 inch type, the FORMAT utility would not prompt for the diskette format, but instead supply a default value of DS/DD. In addition, the number of tracks formatted would be 36 and not 77 as above.

```
;continue with BACKUP
A>BACKUP cr
RC700 BACKUP VERS 1.0 82.10.12
SOURCE DRIVE (A/B/C) ? A
DESTINATION DRIVE (A/B/C) ? B
INSERT SOURCE DISKETTE AND PRESS RETURN cr
INSERT DESTINATION DISKETTE AND PRESS RETURN cr
BACKUP COMPLETE
INSERT CP/M SYSTEM DISKETTE IN DRIVE A AND PRESS RETURN cr
A>
```

2.3.2 One-Drive System

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

```
A>FORMAT cr
RC700 FORMAT UTILITY VERS 1.2 82.03.03
FORMAT: 1=SS/SD 2=DS/DD TYPE (1,2) ? 2     ;this prompt appears
                                           ;only when 8-inch disk-
                                           ;ettes are used.

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         ;with 5 1/4 inch diskette,
                                       ;36 tracks.

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
RC700 BACKUP VERS 2.1 82.10.12
SOURCE DRIVE (A/B/C) ? A
DESTINATION DRIVE (A/B/C) ? A
;source disk = System Diskette
;destination disk = new diskette
INSERT SOURCE DISKETTE AND PRESS RETURN cr
INSERT DESTINATION DISKETTE AND PRESS 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 DISKETTE IN DRIVE A AND PRESS RETURN cr
A>
```

3. OPERATING CP/M

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

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 front of the console is pressed (hard reset).

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

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.

The CP/M System disk may be modified to automatically execute a command line after each cold/warm boot. See subsection 4.3.12.

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 RC700 keyboard:

- The key marked <- deletes the last character typed.
- The key marked -> deletes the entire line typed.

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
.3 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

Unambiguous and Ambiguous Filenames

The filename form pppppppp.sss identifies a single file, and is
3.4 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 (e.g. "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

3.5 Switching Drives

CP/M indicates the currently logged drive, and by implication the disk which it contains, by prompting with the disk drive name ("A", "B", "C", etc.) followed by a right angle bracket (>). On a multidrive 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>
```

3.5

4. CP/M COMMANDS

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 RC700 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 cr                ;RC700 transient command TRANSFER

A>COBOL 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 (for an alphabetical listing, use CAT - see 4.1.9)

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

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 CTRL Key and P simultaneously.

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

4.1.5 STAT

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)

.1.5  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 current device
                                ;assignments
```

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

Note that, in the RC700 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

```
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

RC700 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). When 5 1/4 inch diskettes are used the format is set by default to double-sided, double-density. 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 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>
```

.1.7

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.

4.1.9 CAT 4.1.9

RC700 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 and states the total number of unused directory entries and free disk space.

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     u;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 C: cr              ;lists all files on disk C.
```

4.1.10 FILEX

RC700 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 terminal 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. I). It is furthermore required that the terminal ports of both computers use the same linespeed (baud rate) and character format (use the CONFI command, 4.3.9, 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, B etc.) followed by a colon and an unambiguous file-name. To make it possible to distinguish between the local drives A, B etc. and the remote drives A, B etc. 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

4.2 Program Handling Commands

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

4.2.1

4.2.1 SUBMIT

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 $$$.SUB on the disk in drive A. When the system reboots on the termination of SUBMIT, the CCP reads $$$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

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
G0
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

4.2.3 ASM

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

4.2.4 LOAD

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 CTRL key and P simultaneously.

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

4.2.6 DDT

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, etc. 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 PROG.COM cr           ;loads DDT, which in turn loads
DDT VERS 2.2                ;the file PROG.COM from the disk
NEXT  PC                    ;in drive A, then displays the
0300  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

4.2.7 SAVE

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 PROG.COM cr           ;saves memory from 0100H through
                                ;1FFFH (NEXT was 2000) and places
                                ;it in the file PROG.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, HDINST, BACKUP, ASSIGN, VERIFY, STORE, RESTORE, SYSGEN, CONFI, SELECT, USER and AUTOEXEC.

4.3.1 FORMAT 4.3.1

RC700 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.

Soft-sectored 8 or 5 1/4-inch diskettes must be used: either single-sided, single-density (8-inch diskettes only), or double-sided, double density (8 or 5 1/4-inch diskettes). With 8-inch diskettes, 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
RC700 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
```

4.3.2 HDINST 4.3.2

RC700 transient command,
the form of which is: HDINST cr

.1 HDINST is used to configure (or reconfigure) the RC763 hard disk. Consequently HDINST can only be used in connection with RC700 systems which include the hard disk.

HDINST makes 4 different configurations of the hard disk available to the user. In each configuration, the available space on the hard disk is assigned to 2 to 5 logical CP/M disk drives. These drives have been given the names C, D, E, F and G respectively. Once installed, each of the logical disk drives may be accessed and used in essentially the same way as the physical disk drives A and/or B. Possible restrictions are mentioned in connection with the individual CP/M command descriptions in this manual.

The four configurations the user may choose from each time he calls HDINST are described in appendix H. It is advisable to consult the appendix before using HDINST for the first time.

In all 4 configurations the first logical disk is called C even if the system only has one flexible disk drive (i.e. A but not B). Furthermore C will always have the same capacity as the system's flexible disk drive (A and/or B). Thus if the system has a mini disk drive A, then C's capacity will be 0.27 megabytes. On the other hand, if the system has a maxi disk drive, C's capacity will be 0.9 megabytes.

Please note that it is possible to perform a cold boot from the hard disk, provided that the RC700 system in question includes a hard disk autoload prom (MF016, micro feature No 016) and that the CP/M system has been copied to tracks 0 and 1 of the hard disk. To copy the system to tracks 0 and 1 of the hard disk the user need only answer "yes" (i.e. press the key marked Y) in response to the following question from HDINST:

```
COPY CP/M SYSTEM TO HARD DISK? (Y/N)
```

WARNING: Once a configuration has been selected, any and all data on the disk will be erased.

The sample dialogue below illustrates step-by-step how HDINST may be used.

Example (with comments):

```
A>HDINST cr
RC700 - HARD DISK INSTALLATION - VERS.1.1 82.09.28
NEW CONFIGURATION ON HARD DISK? (Y/N) Y ;Press Y to configure.
WARNING! A NEW CONFIGURATION WILL CAUSE OLD DATA ON THE HARD DISK
          TO BE ERASED - OK? (Y/N) Y ;Press Y to configure.

Type of configuration      Disk sizes
1: DISK C,D                0.3 MB  7.9 MB
2:      C,D,E              0.3 MB  3.9 MB  3.9 MB
3:      C,D,E,F            0.3 MB  1.9 MB  1.9 MB  3.9 MB
4:      C,D,E,F,G          0.3 MB  1.9 MB  1.9 MB  1.9 MB  1.9 MB

SELECT TYPE (1/2/3/4): 1   ;press number (from menu above) of con-
                           ;figuration wanted.

DRIVE A: 0.3 MB MINIDISKETTE       ;the new
DRIVE C: 0.3 MB HARD DISK UNIT     ;configur-
DRIVE D: 7.9 MB HARD DISK UNIT     ;ation
```

```
NEW CONFIGURATION OK? (Y/N) Y  ;press Y if configuration is OK.
WRITING CP/M FORMAT (E5's), TRACK: 192
FUNCTION COMPLETED
COPY CP/M SYSTEM TO HARD DISK? (Y/N) Y  ;if the system includes
                                        ;the hard disk autoload
                                        ;PROM, pressing Y makes
                                        ;it possible to perform
                                        ;a subsequent cold boot
                                        ;from the hard disk.
SELECT SOURCE DRIVE (A/B) A
INSERT CP/M SYSTEM DISKETTE IN A AND PRESS <RETURN> cr
CP/M SYSTEM COPIED
PRESS <RESET> TO INITIATE NEW SYSTEM CONFIGURATION ;reset loads
                                                   ;newly config-
                                                   ;ured system.
```

4.3.3 BACKUP 4.3.3

RC700 transient command, the forms of which are:

```
BACKUP cr
BACKUP FAST cr
```

The BACKUP command is used to copy an entire 8-inch or 5 1/4-inch diskette on a one-drive or a two-drive system. 8-inch diskettes may be single-sided, single-density (128 bytes per sector) or double-sided, double-density (512 bytes per sector). 5 1/4-inch diskettes must be double-sided, double-density. The diskette formats are further described in appendix E.

Factory-fresh 8-inch 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. 5 1/4-inch diskettes must always be formatted, as they are not preformatted.

The BACKUP command will normally copy the entire source disk and verify the copy. If the option "FAST" is used, however, it will copy without verifying.

Please note that if an RC700 system includes a logical disk C in its current configuration, C may also be used by BACKUP as a source and/or destination disk.

BACKUP prompts the console for commands, with interaction as shown below. A system reboot (warm boot) can be performed (by pressing the "CTRL"-key and C simultaneously) whenever the program is requesting input.

Example

```
A>BACKUP cr
RC700 BACKUP VERS 2.1 82.08.10
SOURCE DRIVE (A/B/C) ? A
DESTINATION DRIVE (A/B/C) ? B
INSERT SOURCE DISKETTE AND PRESS RETURN cr
INSERT DESTINATION DISKETTE AND PRESS RETURN cr
CHECK READING COPY
BACKUP COMPLETE
INSERT SYSTEM DISKETTE IN DRIVE A AND PRESS RETURN
A>
```

In the event of read or write errors that may occur during copy operations BACKUP will prompt the console as follows:

```
BAD SECTOR ON SOURCE DISK          ;after a read error
```

or

```
BAD SECTOR ON DESTINATION DISK     ;after a write error
```

Recovery from these errors is described in chapter 5.

4.3.3

4.3.4 ASSIGN

RC700 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
```

4.3.5 VERIFY

4.3.5

RC700 transient command,
the forms of which are: VERIFY cr
VERIFY d: cr

where "d" may be the drive name for disk drive 1 or 2 (i.e. "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 there are bad sectors and then display the following prompt:

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

Before typing Y - in which case VERIFY will create a read-only dummy file called BLOCKS.BAD - it is advisable to copy possible files from the bad disk onto another disk, reformat the bad disk
3.3 and try VERIFY on it once again. Please note that creating the dummy file may damage the files already on the disk.

If VERIFY again finds bad sectors, answer "CREATE DUMMY FILE (Y/N):" by typing Y. The effect of creating the dummy file is to make it possible to use the bad disk without having trouble with the bad sectors. An alternative to creating the dummy file is, of course, discarding the disk.

If you decide to create the dummy file, be careful not to erase it (i.e. never use ERA *.*). If you did erase it you would have to create the dummy file again (by executing VERIFY as explained above) or else discard the disk.

You can interrupt VERIFY at any time by typing CTRL-C.

Example

```
A>VERIFY cr          ;verification with no bad sector on disk
  RC700 DISKETTE VERIFICATION VERS.1.0  82.09.14
  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>VERIFY B: cr       ;verification with bad sectors on disk
  RC700 DISKETTE VERIFICATION VERS.1.0  82.09.14
  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     ;WARNING: pressing Y may damage
                                 ;some of the files on the disk-
                                 ;ette.
                                 ;press Y only if important files
                                 ;have been copied to another
                                 ;diskette.
  CREATED DUMMY FILE 'BLOCKS.BAD'
  14 Kbytes LOST
  INSERT CP/M SYSTEM DISKETTE IN DRIVE A AND TYPE <RETURN> cr
A>
```

4.3.6 STORE

RC700 transient command,
the forms of which are: STORE iden d: filerefs cr
STORE iden d: cr

The effect of STORE is as follows: The files named as "filerefs" - either directly or in an input file, after a prompt from STORE - will be copied from the logical disk specified as "d" onto one or more diskettes (depending on the length and number of the files). The name the user specifies as the "iden" will serve as a label which can be used with the RESTORE command (see 4.3.6) if and when the user wishes to re-transfer the files in question to the hard disk.

"Iden" may be a name of the user's own choosing, but must be no longer than 8 characters. As "d" the user may specify any hard disk unit (i.e. any logical disk in the current configuration, e.g. "C", "D", etc.) - please note that a colon must follow directly after the name of the hard disk unit. "Filerefs" may be the names of one or more files on the logical disk specified as "d". Alternatively, the user may omit a file reference (as in the second form of the command), in which case STORE will prompt for the name of an input file containing 1 or more file references.

In the event that the user chooses to use the form of STORE in which the "filerefs" are omitted, he must create an input file with the ED facility before calling STORE. This input file must have the following form:

```
first-file-ref
second-file-ref
.
.
.
last-file-ref
```

Please note that a maximum of 100 file references are allowed.

Example

```
A>STORE JCS G: WP.COM cr
RC700 STORE VERS.1.1  83.01.03
DESTINATION DRIVE (A/B)? A
TYPE DATE AND CLOCK (YYMMDD.HHMM), PRESS <RETURN> 821021.1630 cr
INSERT FORMATTED DISKETTE LABELED:
JCS      SERIAL NO.: 01 in drive A, press <RETURN> cr
SAVING-
  WP     .COM
STORE FINISHED
INSERT CP/M SYSTEM DISK IN DRIVE A, PRESS <RETURN> cr
A>
```

4.3.7 RESTORE

4.3.7

RC700 transient command,
the forms of which are: RESTORE iden d: ufn cr
RESTORE iden d: ufn CHECK cr

The effect of RESTORE is to copy a file which has been saved on a diskette by means of the STORE command. The file is copied back onto the hard disk unit (logical disk drive) identified by "d".

"Iden" must be the same name that was used in the STORE operation which originally transferred the file from the hard disk to the diskette. As "d" the user may specify the name of any hard disk unit in the current configuration (e.g. "C", "D", etc.).

For "ufn" the user may specify either:

1) a single unambiguous file name

or

2) *.*

If "*.*" is used, ALL files on the diskette identified by the label "iden" will be copied onto the hard disk unit specified as "d" in the command line.

If the CHECK option is used, the file or files in question will be checked but not copied.

Example

```
A>RESTORE JCS G: WP.COM cr
RC700 RESTORE VERS: 1.0 82.10.06
SOURCE DRIVE (A/B)? A
INSERT JCS SERIAL NO. 1 IN DRIVE A, PRESS <RETURN> cr
DISKETTE IDENTIFICATION ...JCS       -SERIAL NO. 1
CREATED...................821021.1630
CATALOG OF DISK A - ORIGIN, FILE NAME:
  G: WP       .COM
RESTORE  G:WP       .COM  (Y/N)? Y
INSERT SYSTEM DISKETTE IN DRIVE A, PRESS <RETURN> cr
A>
```

4.3.8

4.3.8 SYSGEN

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

The SYSGEN command is used, in the RC700 implementation of CP/M, to get the BIOS, BDOS and CCP parts of CP/M, either from the reserved tracks of a System Diskette or from a file, and to write the BIOS, BDOS and CCP to the reserved tracks of a new or existing System Diskette.

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

The BIOS, 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 BIOS, BDOS and CCP can subsequently be read by SYSGEN as a source file.

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

The use of 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.

4.3.9

4.3.9 CONFI

RC700 transient command,
the form of which is: CONFI cr

The CONFI command enables the user to change the values of configuration parameters in the RC700 BIOS drivers.

When the user calls CONFI by entering

```
A>CONFI cr
```

the program responds by displaying the following text:

```
RC700 CP/M CONFIGURATION UTILITY vers 2.1 13.02.83
SELECT SYSTEM DISK FOR CONFIGURATION (A/C)?
```

Once the user has made the proper entry - "A" or "C" - the program responds by displaying a menu containing the various reconfiguration possibilities:

```
1. PRINTER PORT
2. TERMINAL PORT
3. CONVERSION
4. CURSOR
5. MINI MOTOR STOP TIMER
6. SAVE CONFIGURATION DESCRIPTION
```

The user may then select one of the 6 possibilities by typing the corresponding number. When one of the six parameters has been selected, another menu will appear with a further choice of options, or with a list of possible values for the parameter in question.

For instance, the user might choose the parameter "CONVERSION" above by typing the number 3. The following menu would then appear:

```
  1. DANISH
  2. SWEDISH
  3. GERMAN
  4. UK-ASCII
* 5. US-ASCII
  6. FRENCH
  7. LIBRARY
```

The asterisk to the left of "US-ASCII" in the menu above indicates that this is the current value of the parameter "CONVERSION" (current value is determined by the value currently to be found in the configuration description on disk, not the current run-time value - the two values may differ if CONFI has been used after the previous cold boot).

To change the run-time configuration description the user can simply choose a new value for one or more configuration parameters. For instance: if he wishes to change the current value of the conversion parameter from US-ASCII to DANISH (see menu above), he can simply press the number (1) corresponding to DANISH (simply typing RETURN, on the other hand, is equivalent to choosing the current value, US-ASCII, as the new value).

Once the new run-time value has been chosen, the reconfiguration session can be continued (if the user wishes to change the values of other parameters as well) or terminated.

It is possible to terminate in two ways - in both cases one must first return to the original menu containing the SAVE-CONFIGURATION-DESCRIPTION option (if this menu does not appear on the screen immediately after the last new parameter value has been entered, then press RETURN). Once the menu with the SAVE-CONFIGURATION-DEscription option is displayed, one can

1) press RETURN in which case CONFI will terminate without saving the new configuration description (a subsequent cold boot will restore the old configuration).

or

2) press the key marked 6 (corresponding to the SAVE-CONFIGURATION option) in which case the new configuration will be saved on the system disk (a subsequent cold boot will load the new configuration description).

NOTE: The CP/M system diskette must be in drive A during execution of CONFI or, if C is to be reconfigured, the system must already be installed on the hard disk (the example in 4.3.2 shows how to copy the system to the hard disk).

4.3.10 SELECT

RC700 transient command, the form of which is: SELECT port function where "port" may be either P for printer port or T for terminal port and "function" may be A for line A, B for line B, or R for release line.

The SELECT command enables the user to access devices connected via an RC791 Line Selector as shown in the examples below.

Example 1:

```
A>SELECT T B cr                ;this call gives access to line B
                               ;via the terminal port.
```

If the selection was successful, the program will display the following message:

```
LINE READY
```

If the line was occupied by another user or the device connected to the line was offline, the following message will be displayed:

```
LINE BUSY OR DEVICE OFFLINE
```

Example 2:

```
A>SELECT T R cr                ;this call releases the line after
                               ;it has been used.
```

The SELECT program will then respond:

```
LINE RELEASED
```

4.3.11 USER 4.3.11

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 RC700 Microcomputer 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.

4.3.12

4.3.12 AUTOEXEC

RC700 transient command,
the form of which is: AUTOEXEC cr

The AUTOEXEC command modifies the CP/M System on a disk in drive A to automatically execute a command line whenever a cold boot (system boot) is performed from that disk.
The command line may optionally be executed after each warm boot (system reboot) too.

Example

The following example shows how to modify a CP/M system so that the CAT command will be executed after each cold boot.

```
A>AUTOEXEC cr
RC700 Autoexec vers. 1.0 10.01.83

Enter command line to auto-execute, then press <return>
CAT cr

When do you wish to auto-execute the command line?
    1.- after each cold- and warm boot.
    2.- after a cold boot only

Select option: 2
Insert CP/M system disk to modify in drive A, then press <return> cr
Insert CP/M system disk in drive A, then press <return> cr
A>
```

4.4 Extended BIOS Functions 4.4

This section deals with a number of functions maintained in the RC700 systems software and accessible to the user. The user should note that the functions described below, unlike those described in previous sections, cannot be accessed by means of the usual CP/M program execution commands. The entries described below are thus not CP/M command entries but call addresses which can be used at the level described in ref. [1], chapter 6, pp. 127-152.

4.4.1 Real-Time Clock 4.4.1

The RC700 systems software maintains a 32-bit real-time clock which is incremented in steps of 20 ms. The real-time clock can be set and read by calling the routine located at address BIOS+53_{16} (NB BIOS is the address of a jump to WARMBOOT. For 56 k CP/M systems, BIOS = $DA03_{16}$. Thus it is possible to access the clock-routine with the assembly command "CALL 0DA56H").

SET CLOCK:

Register	Call value	Return value
A	0	irrelevant
DE	2 least significant bytes of clock	irrelevant
HL	2 most significant bytes of clock	irrelevant

GET CLOCK:

register	Call value	Return value
A	1	irrelevant
DE	irrelevant	2 least significant bytes of clock
HL	irrelevant	2 most significant bytes of clock

4.4.2 EXIT Routine

An EXIT routine is defined using the entry BIOS+50_{16}.

Register	Call value
HL	Address of EXIT routine
DE	Count

After count * 20 ms. the routine will be called.

As the routine will be part of an interrupt service routine, the following restrictions should be obeyed:

1) The routine is entered with interrupts disabled and the routine must not enable interrupts.

2) The amount or processing in the routine should be as small as possible.

4.4.3

4.4.3 Line Selector Control

Printers can be shared among several RC700 Microcomputers using the RC791 Line Selector. The line selector is controlled using the entry BIOS+$4D_{16}$.

Register	Call value	Return value
A	PORT	RESULT or irrelevant
B	FUNCTION	irrelevant
C	irrelevant	irrelevant

FUNCTION:
0 = Release line (always legal, RESULT irrelevant)
1 = Select line A
2 = Select line B

PORT:
0 = Terminal port
1 = Printer port

RESULT:
255 = Selection OK
  0 = Line busy

The selection of a line is automatically preceded by a release line (in case a line was already selected).

After a line has been used, it should be released so as to allow other users to access the line.

4.4.4 Reader Status 4.4.4

To test the reader status (RDR:) there is an extra BIOS function at location BIOS+$4A_{16}$. The BIOS function is similar to CP/M function 11 (Get Console Status), where FF_{16} is returned when a character is available, otherwise 00 is returned.

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].

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 BACKUP

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

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] RCSL No 42-i1610:
CP/M Operating System Manual, Digital Research, 1982

B.

B. DISPLAY HANDLING

The display handling routines supported by the CONOUT procedure of the RC700 BIOS are described in this appendix (see further ref. [6] and [7]).

B.1

B.1 X-Y Addressing

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

(Note: By changing the configuration, it is possible to reverse the horizontal/vertical order above - see appendix G, G.4.2.)

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

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

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

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

B.2

B.2 Various Functions

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).

20 Causes all subsequent characters to be marked as background characters.

21 Causes all subsequent characters to be marked as foreground characters.

22 Causes all characters marked as foreground characters to be deleted without affecting the background characters.

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 (79,y).

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

B.3 Attributes

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
blinking	2	02	00000010
semigraphic	4	04	00000100
inverse video	16	10	00010000
underscore	32	20	00100000

The attributes may be combined in any order. Sending a 128+4+16 = 148-character to a given position, for instance, will assign to it the attribute inverted-semigraphic.

Note that an attribute occupies one position on the screen.

B.4 Semigraphic Character Set

The semigraphic attribute defines a new character set. The character values of this set are shown in fig. 1.

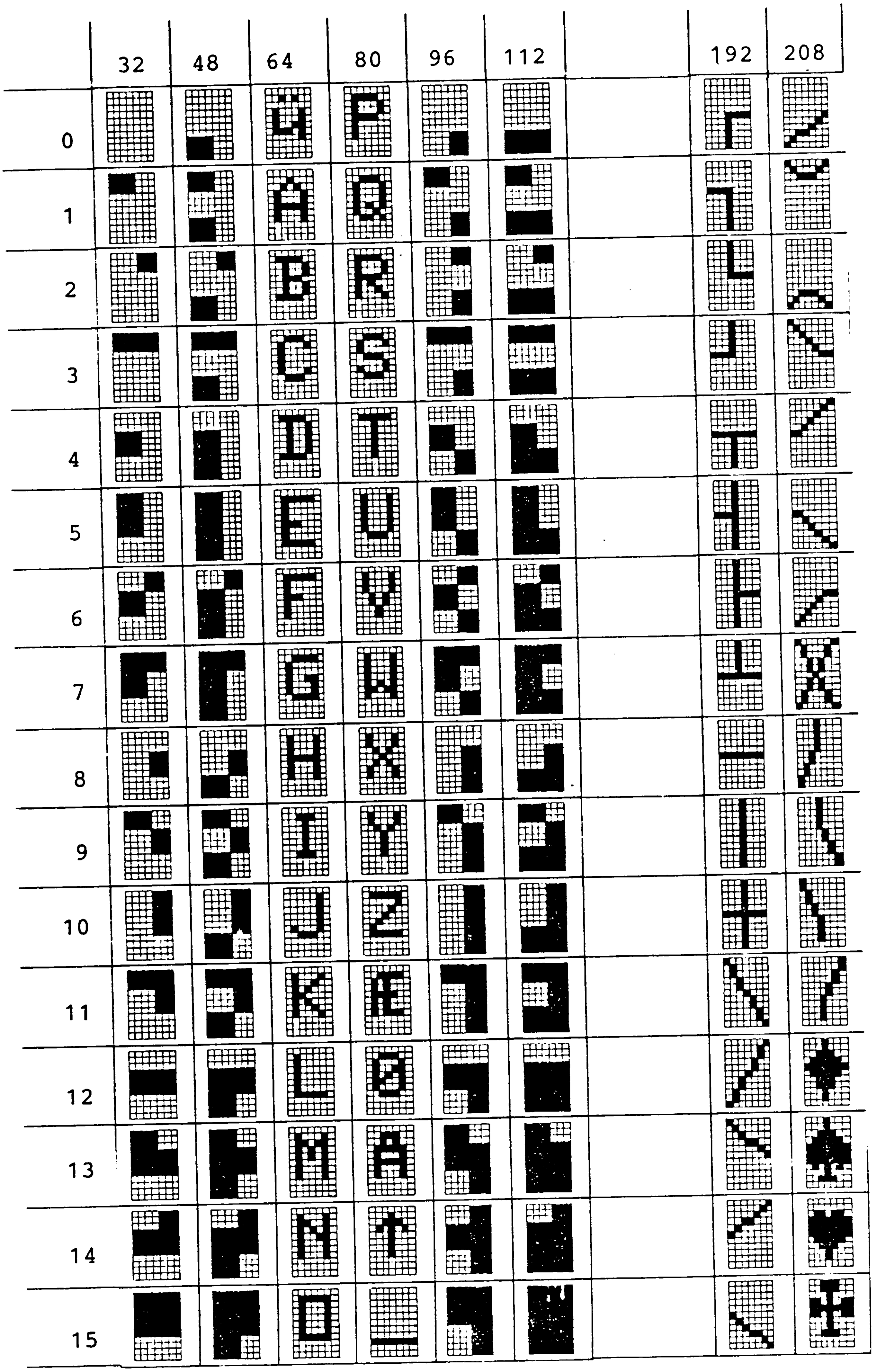


Figure 1: Semigraphic character set.

C. SYSTEM DISKETTE GENERATION

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, BDOS or CCP (see further ref. [7]).

1) Patch the BIOS (see the example below), BDOS and/or CCP as desired.

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) and then delete the unwanted files using ERA (see subsection 4.1.2).

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

4) Write BIOS, BDOS AND CCP (CPM56.COM) to the new System Diskette using SYSGEN (see subsection 4.3.8).

Please note that SYSGEN can operate with or without a previously created file containing BIOS, BDOS and CCP. Furthermore, note that SYSGEN can only operate on files created by SYSGEN itself.

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 SYSGEN, SAVE, DDT, and the patch addresses for the key positions shown in appendix D.

Please note that it will be assumed that an 8-inch drive is being used and that the keyboard is an RC722 (the layout of which is described in appendix D, fig. 3). ; get a version of BIOS, BDOS and CCP

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


   A>
  ;save the BIOS, BDOS and CCP. For mini diskette systems save 68
  ;pages of 256 bytes, and for maxi diskette systems, 107 pages.

   A> SAVE 107 CPM56.COM cr
   A>
  ;change PA2 to CONTROL-R, PA3 to CONTROL-U and PA4 to
   CONTROL-S.
  ;
  ;find the patch address by following the description in
  ;appendix D.
  ;
```

;patchaddress for PA2: = $4680_{16} + 84_{16} = 4704_{16}$
;patchaddress for PA3: = $4680_{16} + 8B_{16} = 470B_{16}$
;patchaddress for PA4: = $4680_{16} + 8E_{16} = 470E_{16}$

```
  ;patch the values into the patch address using DDT

    A> DDT CPM56.COM cr
    DDT VERS 2.2
    NEXT PC
    6C00 0100
    -S4704 cr
    -4704 04 12 cr    ;patching PA2
    -4705 05 . cr
    -S470B cr
    -470B 0B 15 cr    ;patching PA3
  -470C 0C . cr
  -S470E cr
  -470E 0E 13 cr     ; Patching PA4
  -470F 8F . cr
  -G0 cr

  A>
  ; Save the new patched BIOS, BDOS and CCP with SAVE
  A>SAVE 107 CPM56.COM cr
  A>
; Write the new BIOS, BDOS and CCP into the diskette with
; SYSGEN
  A>SYSGEN CPM56.COM cr
  SYSGEN VERS 2.0
  DESTINATION DRIVE NAME (OR RETURN TO REBOOT) A
  DESTINATION ON A, THEN TYPE RETURN cr
  FUNCTION COMPLETE
  DESTINATION DRIVE NAME (OR RETURN TO REBOOT) cr

  A>
; Cold boot the system and press PA2, PA3 and PA4 to check the
; result.
```

D. KEYBOARD

The RC700 microcomputer keyboard exists in two layouts, where RC721 keyboards with production number before KBU723 serial No 51, and RC722 keyboards before KBU722 serial No 384, follow the layout shown in fig. 2, latter productions of RC721/RC722 keyboards follow fig. 3.

The patch address for each individual key is: baseaddress + key value.

The baseaddress is, for mini diskette systems, 2E80H.
For maxi diskette systems it is 4680H.

Thus the patch address for the shifted PA5 key is:

2E80H + 1EH = 2E9EH for mini diskette systems.

4680H + 1EH = 469EH for maxi diskette systems.

RC721/722 Keyboard layouts.

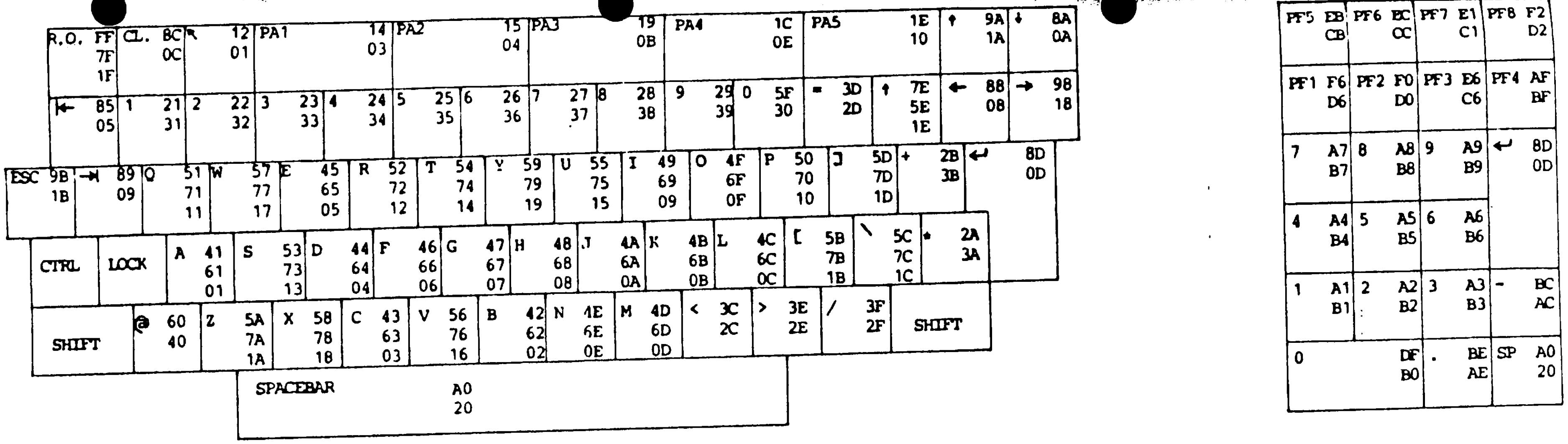


Figure 2.

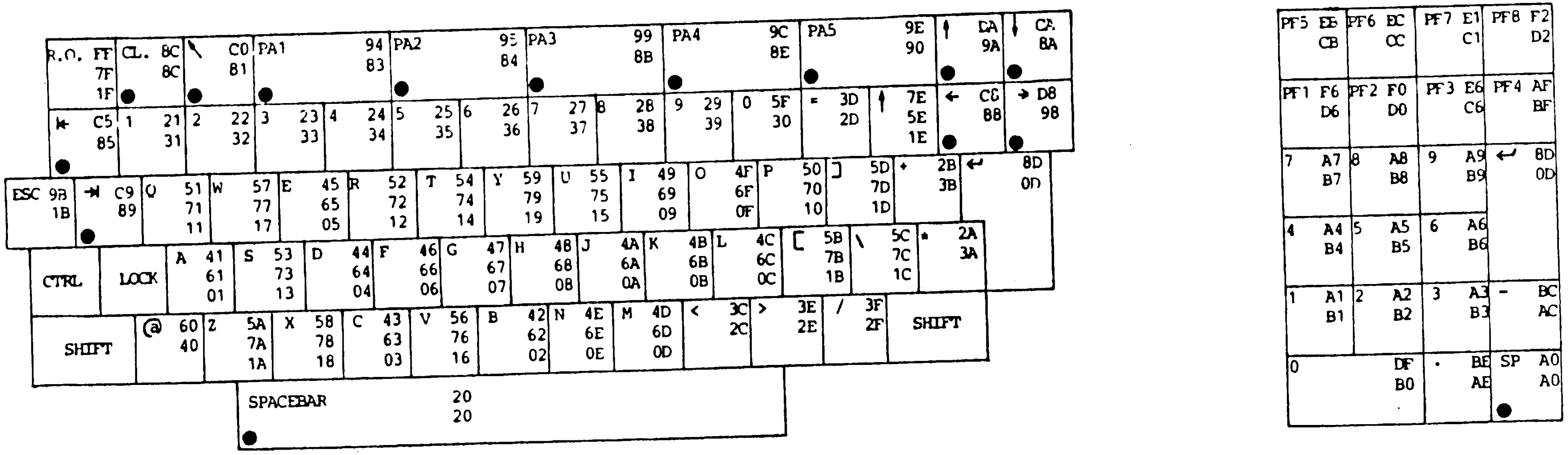
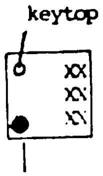


Figure 3.



XX - value delivered by keyboard (hexadecimal) when "SHIFT" is used

XX - value delivered by keyboard (hexadecimal) when "SHIFT" is not used

XX - value delivered (if different) by keyboard (hexadecimal) when "CTRL" is used

● indicating change.

E. DISKETTE FORMATS

E.

E.1 System Diskette

E.1

The 5 1/4 -inch diskette is a dual density, double sided, 512 bytes per sector, 9 sectors per track floppy disk with 36 cylinders.
Recommended type: Verbatim MD550-01-18188.

CP/M characteristics are:

- 270 KB drive capacity in blocks of 2 KB.
- 128 directory entries.
- 2 reserved cylinders.
- Logical sector mapping with 2 to 1 interleaved sectors and zero track to track skew.

The 8-inch diskette is a double-density, double-sided, 512 bytes per sector, 15 sectors per track floppy disk with 77 cylinders.
Recommended type: 3M 743-0-512

The CP/M characteristics are:

- 900 KB drive capacity in blocks of 2 KB
- 128 directory entries
- 2 reserved cylinders
- logical sector mapping with 4 to 1 interleaved sectors and zero track to track skew.

E.2 Data Diskettes

E.2

The RC700 CP/M system supports the "standard exchange" 8-inch diskette as a data diskette only. This diskette is a single-density, single-sided, 128 bytes per sector, 26 sectors per track diskette with 77 cylinders.
Recommended type: 3M-740/2-0

The CP/M characteristics are:

- 241 KB drive capacity in blocks of 1 KB

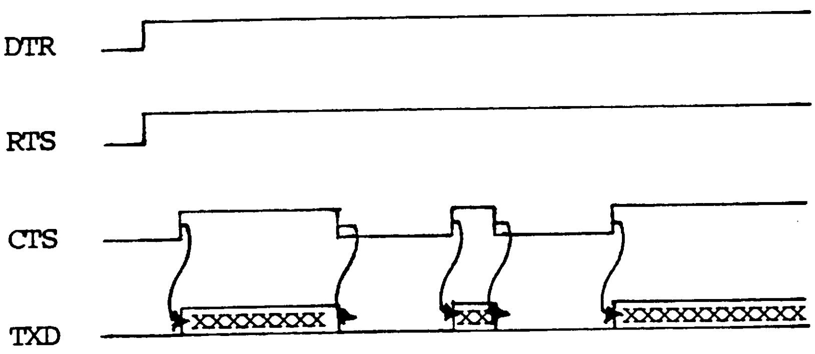
- 64 directory entries
- 2 reserved tracks
- logical sector mapping with 6 to 1 interleaved sectors and zero track to track skew.

F. PERIPHERAL SUPPORT

F.1 Printer Port

This port can be used for attachment of most printers with a serial interface and busy control.

The busy handshake mechanism uses the V.24 signals RTS and CTS as follows:



F.2 Terminal Port

The transmitter part of this port functions exactly as the printer port. The transmitter could be used for a punch device. The receiver part can be used to attach e.g. a reader. The receiver uses the V.24 signal DCD to enable receiving.

G. CONFIGURATION G.

This appendix describes the RC700 configuration parameters handled by the CONFI utility program, and the default values are marked with an asterisk.

G.1 Printer Port G.1

G.1.1 Number of Stop Bits per Character G.1.1

Range: 1-3
* 1: 1 bit
  2: 1.5 bit
  3: 2 bits

G.1.2 Parity G.1.2

Range: 1-3
* 1: even
  2: no
  3: odd

G.1.3 Baud Rate G.1.3

Range: 1-11
  1: 50 bps
  2: 75 -
  3: 110 -
  4: 150 -
  5: 300 -
  6: 600 -
* 7: 1200 -
  8: 2400 -
  9: 4800 -
  10: 9600 -
  11: 19200 -

G.1.4 Number of Bits per Character G.1.4

Range: 1-4

1: 1 to 5 bits

2: 6 bits

* 3: 7 bits

4: 8 bits

G.2 Terminal Port G.2

G.2.1 Number of Stop Bits per Character G.2.1

Range: 1-3

* 1: 1 bit

2: 1.5 bit

3: 2 bits

G.2.2 Parity G.2.2

Range: 1-3

* 1: even

2: no

3: odd

G.2.3 Baud Rate G.2.3

Range: 1-11

1: 50 bps

2: 75 -

3: 110 -

4: 150 -

5: 300 -

6: 600 -

* 7: 1200 -

8: 2400 -

9: 4800 -

10: 9600 -

11: 19200 -

G.2.4 Number of Bits per Character to Transmit G.2.4

Range: 1-4

1: 1 to 5 bits

2: 6 bits

* 3: 7 bits

4: 8 bits

G.2.5 Number of Bits per Character to Receive G.2.5

Range: 1-4

1: 1 to 5 bits

2: 6 bits

* 3: 7 bits

4: 8 bits

G.3 Conversion Tables G.3

Range: 1-7

1: Danish

2: Swedish

3: German

4: UK ASCII

* 5: US ASCII

6: French

7: Library

G.4 Cursor Presentation G.4

G.4.1 Cursor Format G.4.1

Range: 1-4

* 1: Blinking reverse video

2: Blinking underline

3: Reverse video

4: Underline

G.4.2 Cursor Addressing

G.4.2

Range: 1-2

```
* 1: H, V          ;horizontal, vertical
  2: V, H          ;vertical, horizontal
```

G.5 Mini Motor Stop Timer

G.5

Range: 5-1200 seconds

Default: 5 seconds.

H. HARD DISK CONFIGURATIONS

By means of the command HDINST (see 4.3.2), it is possible to configure the hard disk, RC763, in 4 different ways. The tables below indicate various specifications of the logical disks making up the possible configurations.

H.1 Disk Configurations and Disk Capacity

For each configuration below, it is shown what logical disks make up the configuration and the exact capacity of each.

	DISK	DISK CAPACITY IN Mb.
config. 1:	C	0.270 (with a mini)/0.900 (with a maxi)
	D	7.920
config. 2:	C	0.270/0.900
	D	3.936
	E	3.936
config. 3:	C	0.270/0.900
	D	1.968
	E	1.968
	F	3.936
config. 4:	C	0.270/0.900
	D	1.968
	E	1.968
	F	1.968
	G	1.968

H.2 Specifications of Logical Disks by Disk Size

DISK CAPACITY IN Mb	BLOCKSIZE	NUMBER OF DIRECTORY ENTRIES
0.270	2k	128
0.900	2k	128
1.968	4k	512
3.936	8k	512
7.920	16k	512

I. MORE ABOUT FILEX

Appendix I provides technical information concerning the FILEX utility.

I.1 describes various requirements for normal use.

I.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.

I.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 (4.3.9) to ensure:

1) that the two computers have the same baud rate for the terminal port,
2) that the line character format is set to 7 bits per character.

I.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 I.2.1 below.

I.2.2 explains the transmission protocol for transactions.

I.2.1 FILEX Transactions I.2.1

(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

I.2.2 TRANSMISSION PROTOCOL I.2.2

The transactions described in I.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.