PHILIPS

P 2093

8088 COPOWER BOARD

REFERENCE MANUAL

5103 993 30522

PREFACE

With the aid of the 8088 Copower Board and its associated software, your P2000C is converted to a mixed 8- and 16-bit microcomputer, which is capable of using the MS-DOS operating system, running on an 8088 processor, and the CP/M-80 operating system, running on the original Z80 processor.

This manual has been structured so that you will be led from the start, through the preliminaries - the contents of the package, configuration of the system and its operation - on to the more technical aspects of MS-DOS and CP/M, and their utility programs. The manual finishes with detailed hardware information and service requirements.

If you are a newcomer to microcomputers, you will find the first three chapters of most interest. The more technical features of the operating systems will mainly concern the more experienced user.

The following documents also contain useful information:

Microsoft MS-DOS User Guide
P2000C System Reference and Service Manual
P2519 CP/M Reference Manual

ACKNOWLEDGEMENTS

IBM is a trademark of International Business Machines.

Z80 is a trademark of Zilog, Inc.

MS is a registered trademark of Microsoft Corporation.

CP/M is a registered trademark of Digital Research Inc.

WordStar is a trademark of MicroPro International Corporation.

Contents

TABLE OF CONTENTS

Contents

Contents

Contents

Introduction

1 INTRODUCTION

1.1 Contents of the Package

If the 8088 Copower board is not already installed in your computer, you will find the following in your package:

- 8088 Copower PCB in one of three sizes: 128K, 256K or 512K RAM.
- Terminal ROM BIOS chip (which gives a 25-line display on your screen).
- CP/M-80 diskette (640K), containing system files and utility programs, but no CP/M system; i.e. it is not a bootable disk.
- MS-DOS diskette (800K), containing system files and standard utilities.
- Microsoft MS-DOS User Guide
- P2093 8088 Copower Board Reference Manual

1.2 Purpose of the Copower Board

With the addition of the 8088 Copower board and its associated software, the P2000C microcomputer is converted to a mixed 8- and 16-bit computer. The enhanced machine is capable of running the MS-DOS operating system on the 8088 processor or the CP/M-80 operating system on the original Z80 processor.

The Copower board contains memory that can be used as main memory by MS-DOS or, when the computer is running under CP/M, as a RAM disk (instead of the normal RAM disk extension card).

Within the limitations of the P2000C's architecture, the MS-DOS implementation has been made as compatible as possible with the [illegible].

1.3 Introduction to MS-DOS on the P2000C

For the P2000C, version 2.11 of MS-DOS is available. This operating system will, within limits, support IBM-compatible applications.

In this implementation, there are two modes of operation: the P2000C mode, for 7-bit MS-DOS applications, and the PC mode, for 8-bit PC DOS applications. Where necessary, sections of the manual have been split to reflect properly the different features of the two modes.

There are three categories of software to be considered:

- that which is "well behaved" MS-DOS software: using only official MS-DOS functions and requiring no access to the underlying machine. This should run on all machines supporting MS-DOS as long as enough memory and disk storage is provided. This will run on the Copower board in either the P2000C mode or the PC mode. The software must be available on a format of disk supported by the Copower board; i.e. Philips P2000C MS-DOS format or the various IBM PC formats (but not IBM PC AT formats).

- that which is "poorly behaved": using the MS-DOS functions and the authorized features of the IBM PC ROM BIOS. This should run on the Copower board in the PC mode. The software must be available on a suitably formatted disk.

- that which is "badly behaved": using unauthorized features of the IBM ROM BIOS or going directly to the hardware of the machine. This includes those programs that directly access video memory. These programs will not run on the P2000C Copower board.

Note: ROM = read only memory; i.e. memory which is permanently in the machine.

Note: BIOS = Basic Input Output System, a piece of software provided by the manufacturer to control the hardware. This is used to insulate the operating system from the fine detail of the underlying hardware. "Misbehaved" software will access this level directly to achieve a higher performance than can be obtained when using the operating system functions.

The P2000C MS-DOS can read disks which have a standard 160K, 180K, 320K or 360K IBM PC format, and write to them, with the restriction that they may not be readable on a standard IBM disk drive. This is because the higher capacity P2000C disk drives have a narrower head width and so the data written to the disk by the P2000C may be too narrow to be read correctly on the IBM PC. The ability to read correctly will be determined to a large extent by the correctness of the mechanical adjustments in the disk drives. (The drives can only be realigned by a suitably skilled service technician.)

1.4 Description of the Board

The 8088 Copower board is an extension card which uses the internal extension slot in the P2000C. The board contains an 8088 processor, together with the expansion capability of an 8087 mathematics processor, and memory available in 128K, 256K and 512K options.

The 8087 mathematics processor is not necessary for the operation of MS-DOS or most application software. Indeed, software that was not written specifically to use the 8087 instructions will not use the 8087 even if it is installed.

However, applications that have been written to take advantage of the 8087 will perform mathematical functions and calculations faster than they would without the coprocessor being present. The 8087's extra instructions (which are not available in the 8088's instruction set) allow faster execution of multiply, divide and other mathematical functions.

The 8088 board can hold varying amounts of RAM memory depending on the number and type of RAM memory chips installed. It is capable of supporting 64k-bit or 256k-bit memory chips. When the board is full with 64k-bit chips, it will provide 128K of memory. When it contains 256k-bit chips, it can provide either 256K or 512K of memory.

2 INSTALLATION AND CONFIGURATION

2.1 Hardware Installation

If the 8088 Copower board and terminal software are not already installed in your computer, refer to Chapter 10.

The essential items for installation are a P2000C with 640K disk drives, the 8088 Copower board, and the terminal software (which should be version 1.3 or greater).

If you are in doubt about the presence of the terminal software, you can check by holding the ESC key down and pressing the RESET button. If the software is present, you will see on your screen a message showing the terminal software version number. (The system will then go into a terminal test routine, so you should press the RESET button to carry on.)

You can check to see that the Copower board is installed by running the utility program MSBOOT. If the board is not present, you will see the message 'Hardware error, NO ANSWER from extension card'. If your disk is not yet installed (refer to sections 6 and 7 of this chapter for the installion procedure), you will see the message 'File not found: MSKEY.TAB'.

2.2 Preparing Disks

Before you go any further, put a write-protect tab on each of your two delivered disks.

2.2.1 Preparing the Copower Disks

The first thing to do is to make back-up copies of your original two disks supplied with the Copower board (the CP/M disk and the MS-DOS disk), using [illegible] as described in Chapter 2.4.2.

2.2.2 Preparing a CP/M Work Copy

Make a copy of your normal CP/M system disk which should contain the following files:

```
cpm61.com
cpm62.com
cpm63.com
config.com
config.msg
config.dat
pip.com
stat.com
submit.com
util.com
util.msg
```

The disk should have at least [illegible] of free space. You can use [illegible] to put the bootstrap information on the disk.

2.3 Disk Contents

2.3.1 CP/M Copower Disk

TEST SOFTWARE

test88.com - a Copower board test program

RAM DISK SUPPORT

cbios61.com - allow use of Copower board
cbios62.com as RAM DISK from CP/M
cbios63.com
instcpm.sub - submit file to install

MS-DOS SUPPORT

msconf.com - a modified version of CONFIG
msconf.dat - data files for MSCONF

msboot.com - a secondary bootstrap program to load MS-DOS into the Copower board and transfer control to MS-DOS

ibmkey.tab - special key translate table for PC mode

phikey.tab - special key translate table for P2000C mode

instibm.sub - submit file to install in PC mode

instphi.sub - submit file to install in P2000C mode

MSUTIL AND ASSOCIATED FILES

msutil.com - special format and copy program, and hard disk installation
msutil.msg - message file for MSUTIL
msutilt.mac - source code for disk tables
msutilm.mac - source code for messages
- these files can be altered to add new formats of diskette, or translate messages

MSUTILM and MSUTILT are written in Z80 mnemonics. They are loaded and linked, with the Microsoft utilities M80 and L80, in this way:

```
M80 =MSUTILM
L80 /P:100,MSUTILM,MSUTIL.MSG/N/E
```

2.3.2 MS-DOS Copower Disk

STANDARD MS-DOS FILES

msdos.sys (hidden file)
io.sys (hidden file)
command.com
edlin.com
debug.com
link.exe
chkdsk.com
sys.com
diskcopy.com
recover.com
print.com
more.com
find.exe
exe2bin.exe
fc.exe

P2000C-SPECIFIC FILES

format.com - modified for P2000C format

config.sys - adjusted to suit MS-DOS on P2000C

ansi.sys - an ANSI terminal output driver (no keyboard reassignment)

clr.com - clear screen in P2000C mode

reboot.com - restart MS-DOS without return to CP/M

mssort.exe - a sort program with straight collating sequence, can be renamed sort.exe

pcsort.exe - a sort program with the IBM PC collating sequence, can be renamed sort.exe

instibm.bat - a command file to set config.sys for PC mode

instibm.sys - a PC mode copy of config.sys

instphi.bat - a command file to set config.sys for P2000C mode

instphi.sys - a P2000C mode copy of config.sys

2.4 Disk Formatting and Copying

2.4.1 Formatting

FLOPPY DISKS

Before you can use floppy disks under MS-DOS, they must be formatted and have a directory installed, and have some other system information present. (Chapter 9.4 contains more detail.) For new disks, this can be done under the CP/M operating system using the MSUTIL program, or, under MS-DOS, using the FORMAT program.

You can format disks to an IBM PC format by using MSUTIL. (Note that these disks cannot be used as system disks.) To get a higher storage capacity with the P2000C format of 800K, use either MSUTIL or FORMAT.

In answer to the CP/M prompt A>, type MSUTIL. You will then see the MSUTIL main menu. Select option 1, FORMAT FLOPPY DISK. You will see the following message:

```
Enter sequentially the DRIVE NUMBER(S)
containing the disks to be formatted,
terminate with <CR>
exit with <ESC> or <0>
Formatting continues cyclically until an
empty drive is found

Drive(s) to format (1 to 4) =
```

After you have entered the drive number (or numbers), you will see the next prompt:

```
- Select DESTINATION disk type
  1 - P2000C DOUBLE SIDED
  2 - P2000C 160K CP/M
  3 - P2000M CP/M
  4 - P2500 300K CP/M
  5 - P2500 600K CP/M
  6 - P3500 CP/M
  7 - P5020 320K CP/M
  8 - IBM PC
  9 - RAINBOW 400K CP/M
  A - KAYPRO 200K CP/M
```

Now, you have 10 choices. Normally, you will choose the first option, P2000C DOUBLE SIDED. You will then be asked:

```
- Which SYSTEM?
  1 - CP/M
  2 - MS-DOS
```

Choose option 1, CP/M, if you are formatting a CP/M disk, and option 2 if you are formatting an MS-DOS disk.

You will see a warning message before the format program continues. Take care that the correct disk is in the correct drive. The format program will act on any disk, whether that disk is empty or not.

HARD DISKS

A hard disk should be formatted under CP/M in the normal way, i.e. using the UTIL program. Select option 2 from the UTIL main menu, FORMAT HARD DISK.

Note that under MS-DOS it is possible to support larger capacity hard disks than is possible under CP/M-80. So, the hard disk format option of the normal UTIL program can format larger capacities than would normally be used on the P2000C with CP/M.

2.4.2 Copying Floppy Disks

CP/M OR MS-DOS (VARIOUS FORMATS)

With MSUTIL, it is possible to make a copy of a complete disk by choosing the physical copy option; i.e. a copy from track to track. You can also copy a file to a file, if you wish.

If you are running under CP/M, you should use the MSUTIL program to copy disks. This program works even if you do not have a Copower board, and is capable of formatting and copying all the types and formats of disk listed in the selection menus.

If you are running under MS-DOS, you can use the DISKCOPY command. Now, you can copy only MS-DOS disks.

UNDER CP/M

From the main menu of the MSUTIL program select option 2, COPY FLOPPY DISK. Proceed as follows:

- Enter SOURCE drive (1 to 4) : 1
- Enter DESTINATION drive (1 to 4) : 2
- Do you wish to verify while copying (y/n) ? y

(Again, take care that you have the correct disk in the correct drive. It is a good idea to put a write-protect tab on the disk that you want to copy.)

- Select SOURCE disk type:
  1 - P2000C DOUBLE SIDED
  2 - P2000C 160K CP/M
  3 - P2000M CP/M
  4 - P2500 300K CP/M
  5 - P2500 600K CP/M
  6 - P3500 CP/M
  7 - P5020 320K CP/M
  8 - IBM PC
  9 - RAINBOW 400K CP/M
  A - KAYPRO 200K CP/M

Now, you have 10 choices. Normally, you will choose the first option, P2000C DOUBLE SIDED. You will then be asked:

- Which SYSTEM?
  1 - CP/M
  2 - MS-DOS

Choose option 1, CP/M, if you are copying a CP/M disk, and option 2 if you are copying an MS-DOS

disk. Now, you will be asked to choose the type of disk that will be the destination. You will see the same 10 options as for the source. Again, you will normally choose option 1, P2000C DOUBLE SIDED.

You will be asked:

- Which SYSTEM?
  1 - CP/M
  2 - MS-DOS

Choose 1 or 2, as applicable. In answer to the next prompt, 'Make PHYSICAL (track to track) copy (y/n)?', answer 'y'. You will finally be warned that your destination disk will be deleted (have you got the right disk in your destination drive?), and told to press any key to copy.

UNDER MS-DOS

To copy a complete P2000C 800K-format disk under MS-DOS, use the DISKCOPY command. To copy files, use the COPY command. Files can be copied from and to either P2000C 800K- or IBM PC-format disks.

You will find all the details in the MS-DOS User Guide.

2.5 Installing as a RAM Disk under CP/M

Although the Copower board provides the 16-bit processing capability for MS-DOS, it has another function. It acts as a RAM disk for the CP/M-80 operating system.

To use the board as a RAM disk, the CP/M-80 system must be configured using new versions of the supplied CBIOS files. These files also support the normal RAM disk extension card.

It should be remembered that the amount of memory on the Copower board may differ from that on the normal RAM disk extension card, and so the supported disk size may be different.

The data content of the Copower board, when used as a RAM disk, is preserved during a reset and restart of the CP/M system.

To install the new CBIOS files on the work copy of your CP/M system disk (as described in Chapter 2.2.2), you should boot up with the work copy in drive A and the back-up copy of your Copower CP/M disk, containing the support files, in drive B, and then use the submit file INSTCPM.SUB to transfer the files from drive B to drive A. The submit file will invoke the CONFIG program to ensure that you reconfigure using the new CBIOS files.

The command to do this is:

SUBMIT B:INSTCPM

Using CONFIG, don't forget to define a disk volume for the RAM disk, and also to write the new system information onto a disk that you intend to use later as a normal CP/M system disk.

2.6 Installing MS-DOS (PC Mode)

To run programs written for the IBM PC, you have to use this configuration.

2.6.1 What to do under CP/M

With the CP/M work copy (as described in Chapter 2.2.2) in drive A, the back-up copy of the Copower CP/M disk in drive B, and the system showing the prompt A>, it is only necessary to issue the following command to make the CP/M work disk suitable to load and start MS-DOS:

```
SUBMIT B:INSTIBM
```

The submit file will invoke MSCONF, so you should then select the keyboard and printer that you require and complete the configuration so that the new information is written back onto the disk. After a RESET the CP/M system should show on the screen the names of the printer, keyboard and video that you have selected. This can be used as a check that you have really configured as you expected.

The initial configuration will only support two floppy disk drives. But, if you have a hard disk, add it to the disk configuration.

2.6.2 What to do under MS-DOS

Boot CP/M from your work copy and type 'MSBOOT'. Then insert the back-up copy of the MS-DOS Copower disk and type any key. MS-DOS will be booted. Type the date and the time, followed by a carriage return.

Give the command 'INSTIBM', which will copy the contents of the text file INSTIBM.SYS into CONFIG.SYS and copy PCSORT.EXE into SORT.EXE. Then, the REBOOT program will be run to ensure that MSDOS uses the new CONFIG.SYS file.

It is sensible to familiarize yourself with the content of the CONFIG.SYS file to check that the ANSI driver is installed. This can be done with the command:

```
TYPE CONFIG.SYS
```

which will list the file, one line of which should say:

```
DEVICE=ANSI.SYS
```

2.7 Installing MS-DOS (P2000C Mode)

To run (non-IBM) MS-DOS programs, it is advisable to use this installation.

2.7.1 What to do under CP/M

With the CP/M work copy in drive A, the back-up copy of the Copower CP/M disk in drive B, and the system showing the prompt A>, it is only necessary to issue the following command to make the CP/M work disk suitable to load and start MS-DOS:

```
SUBMIT B:INSTPHI
```

The submit file will invoke CONFIG, so you should then select the keyboard and printer that you require and complete the configuration so that the new information is written back onto the disk. After a RESET the CP/M system should show on the screen the names of the printer, keyboard and video that you have selected. This can be used as a check that you have really configured as you expected.

The initial configuration will only support two floppy disk drives.

2.7.2 What to do under MS-DOS

It is necessary to load the back-up copy of the Copower MS-DOS disk when prompted by the MSBOOT program, and then to use CONFIG.SYS without the line:

```
DEVICE=ANSI.SYS
```

Do this by giving the command 'INSTPHI', which will copy the contents of the text file INSTPHI.SYS into CONFIG.SYS and copy MSORT.EXE into SORT.EXE. You should run REBOOT to ensure that MS-DOS will use the new CONFIG.SYS file.

It is sensible to familiarize yourself with the content of the CONFIG.SYS file so that you can check that 'DEVICE = ANSI.SYS' is not present in the file. Use the command:

```
TYPE CONFIG.SYS
```

which will list the file.

2.8 Sharing a Hard Disk

A hard disk can be added to the system. This disk can be divided into two sections: one for CP/M and one for MS-DOS. There must always be a CP/M section defined, and it must be at the beginning of the disk so that the initial CP/M-80 system can be booted from the hard disk. The CP/M section can be up to 5MB.

The CP/M system disk, from which the system was booted before the MSBOOT program was executed, must be configured so that it contains only one logical disk assigned to the hard disk. The logical disk should be selected as the lower section (either as 2MB low or 5MB low).

After the CP/M hard disk configuration has been defined in the configuration program (either MSCONF or CONFIG, as appropriate), you must carry out a further configuration from within the MSUTIL program to define the starting point of the MS-DOS volume. It will be defined to start at some point within the CP/M logical disk and extend physically upwards to the end of the disk.

It is possible that the CP/M-80 area will finish before the logical end of the hard disk volume. The section of the hard disk which is defined as common to both CP/M and MS-DOS will be made into a file and hidden, so that the CP/M system will not access that section, so allowing the MSBOOT program to determine where the MS-DOS section begins. In this way, MS-DOS can find and load its system files.

18

Using MSUTIL, it is not possible to define an MS-DOS section which will accidentally overwrite CP/M data that exists within the logical volume of the current configuration. But you should remember that MS-DOS can use any hard disk space (up to 32MB) which is above the configured CP/M logical volume.

2.9 Installing on Hard Disk

2.9.1 CP/M Files

The files from the Copower CP/M disk can be installed onto the hard disk if the normal CP/M system is started from the hard disk as drive A, and the Copower disk is placed in the floppy drive B. It is then only necessary to run the appropriate install submit file.

i.e. SUBMIT B:INSTCPM
or SUBMIT B:INSTPHI
or SUBMIT B:INSTIBM.

You may find it more convenient to perform the commands contained in the submit files manually, particularly if you wish to install onto the hard disk when drive B is not configured for a floppy disk.

Before installing MS-DOS on the hard disk, you must use MSUTIL to define the portion of the hard disk that is to be used by MS-DOS.

2.9.2 MS-DOS Files

To install the MS-DOS files on the hard disk, you should start MS-DOS from a floppy disk using MSBOOT. The hard disk will be recognized as drive C if it has already been partitioned by MSUTIL. The hard disk must be switched on before the MSBOOT program is used.

The hidden MS-DOS system files are then transferred to the hard disk with the MS-DOS command 'SYS'. Make sure that the current disk is A; i.e. showing 'A>' as the prompt. Then enter the command SYS C: which will copy the system files to the hard disk.

Next, the program COMMAND.COM must be copied to the hard disk with the following command:

COPY COMMAND.COM C:

The other files should then be copied with the command:

COPY *.* C:

Note: The SYS command must be used before any files are copied to the hard disk because the hidden system files MUST be at the beginning. If this is not done, MS-DOS will not be able to find the files and will give an error message.

2.10 APPLICATION SOFTWARE

Many application packages use copy protection schemes to ensure that a user is unable to copy the disks, and so reduce the incidence of "software piracy". Some of the protection schemes may make direct access to features of one manufacturer's hardware, and therefore that software may not operate on the P2000C. (Access directly to the disk controller from the 8088 software is sometimes used, which under the P2000C will not work because the controller can only be accessed from the Z80 processor.)

A popular copy protection scheme uses specially formatted disks, and the application checks that the format is of this special type. A user cannot format a disk to this format. So, a copy would lose its special format and this absence would be detected by the application as it ran.

The two different modes of operating under MS-DOS on the P2000C have to be considered when installing application software.

P2000C MODE

Some software has message files that are coded in 7-bit ISO characters with national variations; for example, an "A umlaut" in German software would be coded with the same value as a square bracket under ASCII. This software would have to be used under the P2000C mode.

The software may need to be configured so that it recognizes the control codes generated by the function keys - the cursor keys, for example. These codes would be similar to those recognized by "Wordstar"; that is, control characters such as CTRL-X.

The software may need to be configured so that it generates the correct screen control characters and escape sequences. In this mode they will be the normal P2000C codes as described in the P2000C System Reference Manual. The software will not work correctly in this mode if it generates ANSI escape sequences.

PC MODE

Some software has message files that are coded in the 8-bit IBM PC character set with national characters; for example, an "A umlaut" in German software, represented by an entry in the top half of the code set above the ASCII characters. This software would have to be used under the PC mode.

The software may need to be configured so that it recognizes the control codes generated by function keys - the cursor keys, for example. These codes will be similar to those generated by the IBM PC ROM BIOS.

The software may need to be configured so that it generates the ANSI screen control escape sequences or BIOS calls. If the software generates ANSI escape sequences then the MS-DOS

system must have ANSI.SYS selected. (This is done by having the line

DEVICE=ANSI.SYS

in the file "CONFIG.SYS".)

If the software generates BIOS calls to perform screen handling then it may not be necessary to have ANSI.SYS installed.

The ANSI.SYS code on the P2000C has a second function. This is to filter control characters and cause them to be displayed as they would on an IBM PC, instead of allowing them through to control the P2000C terminal. This second function means that although ANSI escape sequences are not required, it may be necessary to use ANSI.SYS.

Software that does not generate control sequences or ROM BIOS calls, but instead uses the video memory directly, will not operate.

Using MS-DOS

3 USING MS-DOS

Within the restrictions of the architecture of the P2000C, the MS-DOS implementation has been made as compatible as possible with that of the IBM PC. The Microsoft MS-DOS User Guide contains a full description of the operating system and how it should be used.

3.1 Disk Drive Naming Conventions

MS-DOS can access two internal 96 tpi double sided floppy disks (as supplied on the P2000C model with 640K disk drives) and an attached hard disk. The system cannot be configured for external floppy disk drives or for internal 48 tpi single sided floppy disk drives.

There are two predefined configurations. The process of configuration is automatic, depending on whether the system is booted from a floppy disk or from a hard disk.

Boot Device	Disk Identifiers		
	A	B	C
floppy disk	floppy 1	floppy 2	hard disk
hard disk	hard disk	floppy 1	floppy 2

3.2 Booting, Rebooting, Reconfiguring

Before we talk about the bootstrap procedures, there are two points to remember. Firstly, the system boots by default from floppy disk; and secondly, if you have a hard disk, it must be switched on before you run MSBOOT.

3.2.1 Booting from Floppy Disk

To boot MS-DOS from a floppy disk, do the following:

- put the MS-DOS work copy in drive 1 and leave the door open
- put the CP/M work copy in drive 2
- press the RESET button
- in response to the CP/M prompt A>, type msboot. You will see the message:

Insert MS-DOS disk in drive 1, type return to boot (or exit by CTRL-C)

- You already have the MS-DOS disk in drive 1, so close the door and press the carriage return key

Now, the MS-DOS system will be loaded and you will see a request to enter the date. From this point, you can refer to the MS-DOS User Guide for details.

3.2.2 Booting from Hard Disk

After installing the CP/M and MS-DOS files, as described in Chapter 2.9, it is only necessary to boot CP/M from the hard disk and run the MSBOOT program, which will start MS-DOS. The MSBOOT program can be started automatically with the CP/M autostart feature, so that it is only necessary to reset the P2000C - CP/M and MS-DOS will be restarted automatically. (To set the autostart feature, use CONFIG when in P2000C mode and MSCONF in PC mode.)

You must give either the INSTIBM or INSTPHI command to make sure that the MS-DOS, which is booted from the hard disk, is configured correctly. This need only be done on the first occasion you boot.

3.2.3 Rebooting MS-DOS

To reboot the system, you can either press the RESET button and restart in the normal way or, from MS-DOS, you can run the REBOOT program, which will restart MS-DOS without the need to go back first to CP/M.

3.2.4 Reconfiguring the System for Keyboard, Printer and Screen

The important point about reconfiguration is that the correct program should be used depending on the mode you are in. If you are in P2000C mode, you should run CONFIG. If you are

in PC mode, you should run MSCONF.

If you are reconfiguring MS-DOS, it may be necessary to run INSTIBM or INSTPHI, so that CONFIG.SYS may be set correctly for the appropriate mode.

3.3 Backup/Restore

For data files that do not exceed the storage capacity of one floppy disk (approximately 800K), you can use the normal MS-DOS COPY command to transfer data from a hard disk to a floppy disk, or vice versa.

If this is not adequate, you can get a special backup/restore package from your Philips P2000C dealer. This package is specifically designed to copy and restore large files.

3.4 The Keyboard

3.4.1 P2000C Mode

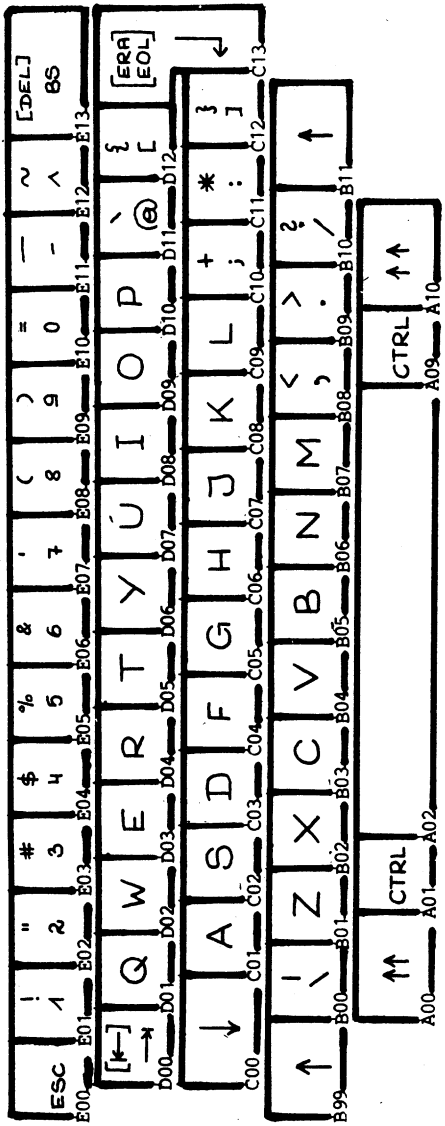
In this mode, keys are used in exactly the same way as under the P2000C CP/M system. No special function keys are defined other than the simple 'Wordstar keys' defined by the standard keyboard configuration of the CP/M-80 configuration program. These keys have normal control functions; for example, CTRL-C.

The MS-DOS special editing functions are not available when you are running in P2000C mode. These keys generate 7-bit ISO codes with national variations. If you want to configure some additional 8-bit codes on the supershift level, you must use the keyboard table editor in the CONFIG program. Some useful keys are listed in Chapter 6.9.

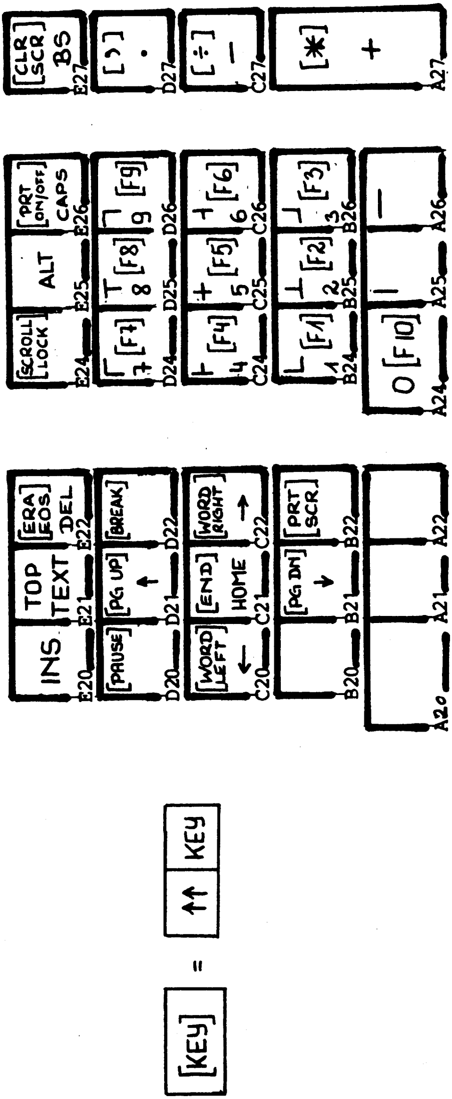
3.4.2 IBM PC Mode

In this mode, keys generate the 8-bit codes which are defined for the IBM PC. Function keys are generated in a similar way as on the IBM PC, and will be recognized as such by MS-DOS. These keys will perform the standard MS-DOS special editing functions as described in the MS-DOS User Guide.

Keyboard national versions, equivalent to those of the IBM PC, are made by reconfiguring the underlying CP/M-80 system using MSCONF.


ESC
[←]
[DEL]
BS
[ERA]
[EOL]
CTRL
CTRL

Using MS-DOS


CLR SCR
BS
E27
D27
C27
A27
PRT ON/OFF
CAPS
ALT
SCROLL LOCK
E26
E25
E24
F9
F8
F7
9
8
7
D26
D25
D24
F6
F5
F4
6
5
4
C26
C25
C24
F3
F2
F1
3
2
1
B26
B25
B24
F10
0
A26
A25
A24
ERA EOS
DEL
TOP TEXT
INS
E22
E21
E20
BREAK
PG UP
PAUSE
D22
D21
D20
WORD RIGHT
END
HOME
WORD LEFT
C22
C21
C20
PRT SCR
PG DN
B22
B21
B20
A22
A21
A20
KEY
=
KEY

FUNCTION KEYS

These are available in PC mode by entering one of the numbers on the numeric pad together with the supershift key [^^]. For example, F10 is [^^] with [0] and F1 is [^^] with [1].

Special functions are also available by entering the appropriate key, usually together with supershift, as you can see in the diagram. In the list below, we indicate those keys that should be pressed in conjunction with the supershift key.

```
     B21 Cursor down
[^^] B21 Page down
[^^] B22 Print screen
[^^] C13 Erase to end of line
     C20 Cursor left
[^^] C20 Word left
[^^] C21 End
     C21 Home
     C22 Cursor right
[^^] C22 Word right
[^^] D20 Pause
     D21 Cursor up
[^^] D21 Page up
[^^] D22 Break
```

E20 Insert (with or without supershift)
E21 Top of text and home (with or without supershift)
E22 Delete
[^^] E22 Erase to end of screen
[^^] E24 Scroll lock
[^^] E26 Printer on/off
E26 Capslock
[^^] E27 Clear screen

ALT KEYS

The ALT keys are available for the alphanumeric and numeric keys.

[^^] A...Z is equivalent to ALT A...Z

Note: When running with a national version, the letter may be marked on a different key position to that of the IBM PC US version. Use the letter as marked on the keyboard, not the key with the same position as that of the IBM version.

[^^] 1...0,-,^ is eqivalent to ALT 1...0,-,=

Note: The [-] and [^] symbols are marked on the P2000C UK keyboard. On other national versions, use the two keys that are in these positions. These keys are the two between the [0] and [backspace] keys on the top line of the keyboard.

[ALT]
[n]...[n] (up to 3 digits from the numeric pad)
[non-numeric, e.g. CR]

is equivalent to: hold ALT
enter number from numeric pad
release ALT

and will generate one character with ASCII value nnn and position code 0.

CTRL KEYS

The use of the CTRL key together with a number will not generate the same code as the IBM PC because of restrictions of the underlying P2000C terminal software and keyboard handling. The generated codes will conflict with CTRL and [letter].

The P2000C restrictions give the following:

[ESC] = [CTRL] ['['] = [CTRL][;]
[CR] = [CTRL] [m] = [CTRL] [-]
[BS] = [CTRL] [h]
[TAB] = [CTRL] [i]

ENTERING CHARACTERS NOT MARKED ON THE KEYBOARD

It is possible to enter any ASCII value from 0 to 255 with the [ALT] key, followed by up to 3 digits and ended with a non-numeric character.

You can also enter some characters at the supershift level, or shift/supershift level, that are not available on the normal national version keyboard. On any national version keyboard, those keys which are not alphabetic and are on the main area (excluding the top row) will at the supershift, or shift/supershift level, generate the symbols defined in the same position on the UK keyboard at the unshifted or normal shift level.

For example, the key (P2000C position B00) will generate on the UK keyboard a backslash or vertical bar at the normal levels. National versions will not, but will at the supershift or shift/supershift levels. These keys are at the following positions:

B00, B08, B09, B10
C10, C11, C12
D11, D12

unless that position generates a letter [A...Z] on the national version keyboard.

3.5 CHARACTER SETS

3.5.1 P2000C Mode

The character set for the MS-DOS P2000C mode is the 7-bit ISO code with national variations as described in the P2000C CP/M Reference Manual.

3.5.2 PC Mode

The character set for the MS-DOS PC mode is the 8-bit IBM PC character set, but with the restriction that not all characters can be represented on the screen due to the restrictions imposed by the inbuilt character generator of the P2000C. The nearest equivalent character will be used to represent the required character; for example, the Peseta character will be represented by a capital "P" on the screen but will print according to the level of support provided by the printer. Most printers do not fully support the IBM PC character set.

The user can redefine the displayed or printed characters by editing the applicable tables in the "MSCONF.DAT" file, if desired, by using the table editing facility of "MSCONF.COM". (It uses the same interface as the CP/M CONFIG program.)

Characters with an internal representation in the control character range (values from 0 to 32) cannot be displayed unless they pass through the ANSI driver, which will switch into mosaic graphic mode and display the characters after adding an offset of 32.

Programs that go directly to the BIOS for output of characters must perform this conversion through 'mosaic mode' themselves. (See the P2000C System Reference Manual for more information on mosaic graphics.)

3.6 Clear Screen Command

3.6.1 CLS (in PC Mode)

The inbuilt command CLS of COMMAND.COM is used in PC mode to clear the screen. This is documented fully in the associated Microsoft documentation. In the P2000C MS-DOS operating system this command is configured to generate the ANSI escape sequence for clear screen.

Use of CLR in PC mode will not clear the screen but will display the characters "2J" on the screen.

3.6.2 CLR (in P2000C Mode)

The program CLR.COM is used instead of CLS to clear the screen when operating in P2000C mode. This is because the operating system command program "COMMAND.COM" can only contain one version of the CLS command, which is configured to operate in the PC mode.

Use of CLS in P2000C mode will not clear the screen but will display a character on the screen which looks like the cursor.

3.7 System Parameters (CONFIG.SYS)

The text file CONFIG.SYS contains a number of commands which are used during the MS-DOS startup sequence to define some aspects of how the system will be configured. For further details of this file, and the available options, consult the MS-DOS User Guide.

The file is supplied with the following:

device=ansi.sys - this causes the system to load the ANSI screen output driver on startup.

buffers=10 - this defines the number of file buffers that are to be used.

files=10 - this defines the maximum number of files that can be used at any one time. This setting is adequate for most usage but can be increased as needed.

country=49 - this is set for Germany. (For the UK, it is set to 44, for France to 33, etc.) Its main use is to force the system to adopt the European date format.

Using MS-DOS

If you are not sure how to use the system editor EDLIN to modify CONFIG.SYS, the two files INSTIBM.SYS and INSTPHI.SYS (two versions of CONFIG.SYS) are provided to make it easy for you to select an appropriate CONFIG.SYS.

4 MS-DOS ON THE P2000C

4.1 Implementation of MS-DOS

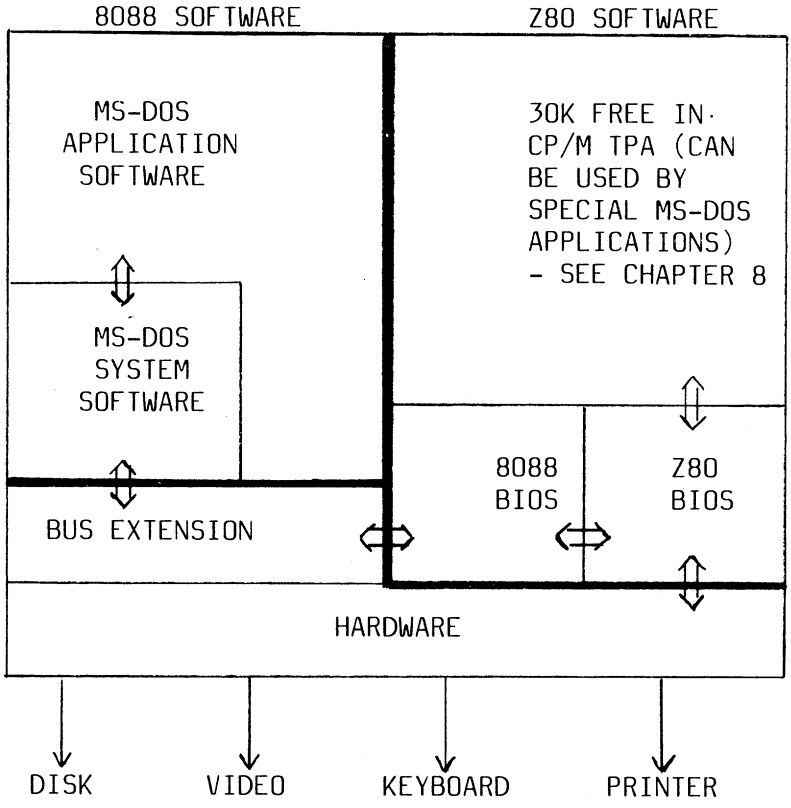
MS-DOS version 2.11 has been implemented on the P2000C.

The MS-DOS implementation on the P2000C is made so that input/output is performed via the 8088 BIOS (part of which resides in the 8088 memory, and part of which is in the Z80 memory). The 8088 BIOS will then communicate via the normal P2000C ROM BIOS to the attached peripherals.

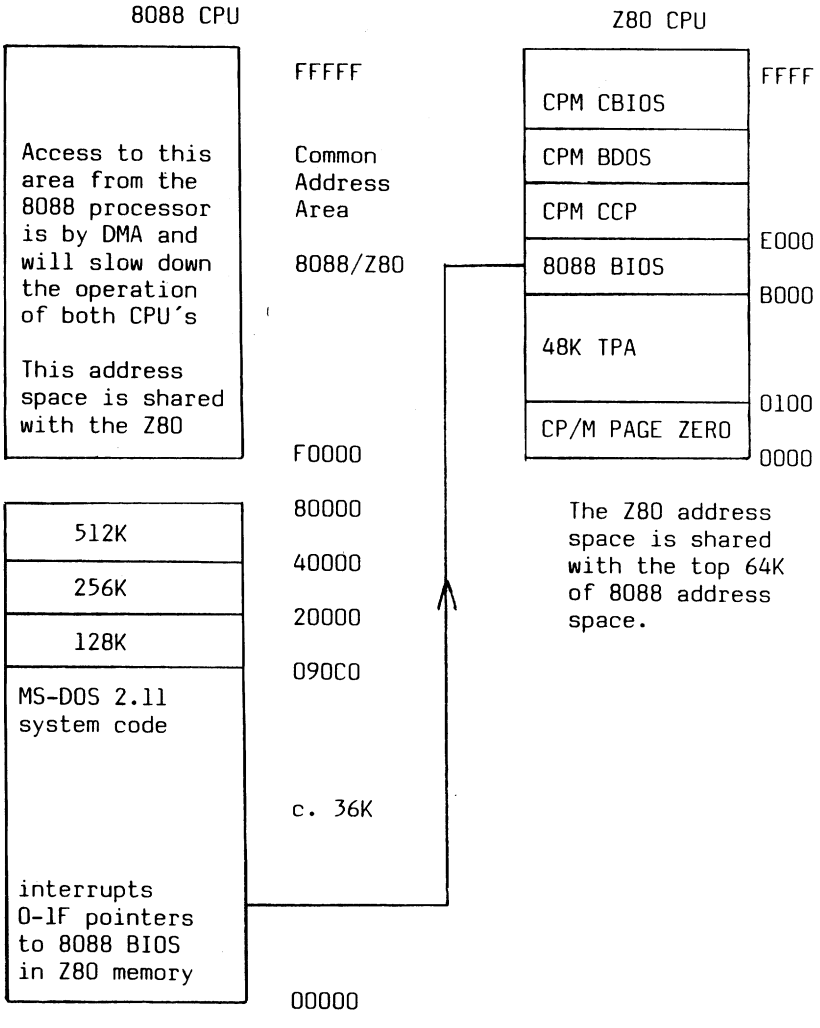
2

MS-DOS on the P2000C

4.2 System Structure

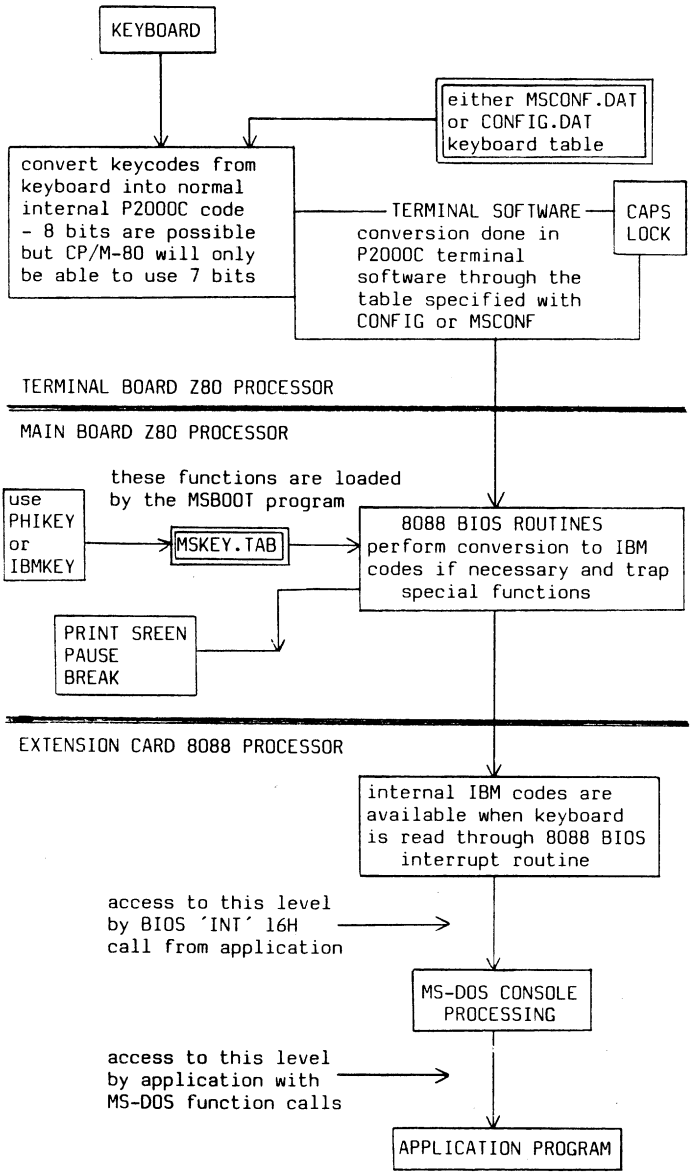


4.3 Memory Map



MS-DOS on the P2000C

4.4 Data Flow and Data Mapping



5 DISK ACCESS

The following formats are supported:

P2000C 800k floppy

80 track/sector, double sided, 10 sectors/track, 512 bytes/sector

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	02H (byte)
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	00C0H (word)
Number of sectors in log. image:	0640H (word)
Media descriptor:	FBH (byte)
Number of sectors per FAT:	0004H (word)
Sectors per track:	000AH (word)
Number of heads	0002H (word)
Number of hidden sectors:	0000H (word)

Disk structure:

Logical address	Contents
0000-01FF	Booting track and BPB.
0200-09FF	First file allocation table (FAT)
0A00-11FF	Second file allocation table (FAT)
1200-29FF	Directory
2A00-C7FFF	Data

Disk Access

P2000C hard disk format with variable length

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	??H (byte)
(to be patched by MSUTIL)	
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	00C0H (word)
Number of sectors in log. image:	????H (word)
(to be patched by MSUTIL)	
Media descriptor:	FAH (byte)
Number of sectors per FAT:	????H (word)
(to be patched by MSUTIL)	
Sectors per track (virtual):	0010H (word)
Number of heads (virtual):	0100H (word)
Number of hidden sectors:	0000H (word)

Disk Access

IBM 160k floppy

40 track/sector, single sided, 8 sectors/track, 512 bytes/sector

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	01H (byte)
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	0040H (word)
Number of sectors in log. image:	0140H (word)
Media descriptor:	FEH (byte)
Number of sectors per FAT:	0001H (word)
Sectors per track:	0008H (word)
Number of heads	0001H (word)
Number of hidden sectors:	0000H (word)

Disk Access

IBM 180k floppy

40 track/sector, single sided, 9 sectors/track
512 bytes/sector

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	01H (byte)
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	0040H (word)
Number of sectors in log. image:	0168H (word)
Media descriptor:	FCH (byte)
Number of sectors per FAT:	0002H (word)
Sectors per track:	0009H (word)
Number of heads	0001H (word)
Number of hidden sectors:	0000H (word)

Disk Access

IBM 320k floppy

40 track/sector, single sided, 8 sectors/track
512 bytes/sector

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	02H (byte)
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	0070H (word)
Number of sectors in log. image:	0280H (word)
Media descriptor:	FFH (byte)
Number of sectors per FAT:	0002H (word)
Sectors per track:	0008H (word)
Number of heads	0002H (word)
Number of hidden sectors:	0000H (word)

Disk Access

IBM 360k floppy

40 track/sector, single sided, 9 sectors/track
512 bytes/sector

Bios parameter block (BPB):

Bytes per sector:	0200H (word)
Sectors per allocation unit:	02H (byte)
Reserved sectors:	0001H (word)
Number of FATs:	02H (byte)
Number of root dir. entries:	0070H (word)
Number of sectors in log. image:	02D0H (word)
Media descriptor:	FDH (byte)
Number of sectors per FAT:	0002H (word)
Sectors per track:	0009H (word)
Number of heads	0002H (word)
Number of hidden sectors:	0000H (word)

6 MS-DOS BIOS (IBM BIOS EMULATION)

The interrupt vectors at the bottom of the 8088 memory will transfer control via a small portion of code in the 8088 to the BIOS which has been loaded in the Z80 memory space and which emulates where possible the IBM ROM BIOS.

The following software interrupts are supported:

- Print screen (INT 05H)
- Execute Z80 subroutine (INT 07H) P2000C onl
- Invalid operation (INT 0EH) P2000C onl
- Video output (INT 10H)
- Equipment check (INT 11H)
- Memory size (INT 12H)
- Disk (INT 13H)
- Communications (INT 14H)
- Keyboard (INT 16H)
- Printer (INT 17H)
- Bootstrap (INT 19H)
- Time of day (INT 1AH)

The following hardware interrupts are supported:

- Keyboard break (INT 1BH)
- Timer tick (INT 1CH)

2

MS-DOS BIOS (IBM BIOS Emulation)

6.1 INT 05H - Print screen

The screen contents will be printed. When a printer timeout error occurs, the beeper will sound.

6.2 INT 07H - Execute Z80 subroutine

On entry:
  BX = subroutine address in Z80 range
  AL = input value to the subroutine
       in register 'A'.
On exit:
  AL = value of register 'A' at exit
       from the subroutine.

To define the free space available for Z80 subroutines, a pointer to the first location occupied by the MS-DOS BIOS is set up at F000:6 and 7. The 8088 processor can use the range from F000:100 to the first occupied location-1.

6.3 INT OEH - Trap Illegal Call

This interrupt is trapped by the 8088 BIOS, which will display on the screen the message:

'ILLEGAL INTERRUPT'

This is used to trap programs that wander into memory which has not been loaded by the program. Unused memory that is filled with 'INT OEH' instructions will cause a jump to this entry when illegally called by an application program.

6.4 INT 10H - Video output

AH=0 Set mode

On entry:
AL=2 for 80*25 Character mode
AL=5 for 512*252 Graphics mode

AH=2 Set cursor position

On entry:
DH=row
DL=column (0,0 is upper left)
(paging is not supported)

AH=3 Read cursor position

On exit:
DH = row
DL = column
CH and CL are cleared (no cursor mode support)

AH=6 Scroll up screen

On entry:
AL = number of lines, if AL=0 blank whole windo
CH = row of upper line of scroll window
DH = row of lower line of scroll window

Input lines at the bottom of the window are blanked.

The column positions CL and DL are considered only if CH = DH (window in one line). In this case the field CL to DL is cleared.

AH=7 Scroll down screen

On entry:
AL = number of lines, if AL=0 blank whole windo
CH = row of upper line of scroll window
DH = row of lower line of scroll window

Input lines at the top of the window are blanked.

The column positions CL and DL are considered only if CH = DH (window in one line). In this case the field CL to DL is cleared.

AH=8 Read attribute/character
at current cursor position

On exit:
AL = character read
AH = attribute of character read
bit-7 is set for blink
bit-0 is set for underline
bit-3 is set for high intensity
bits 6, 5 and 4 are set for invert

AH=9 Write attribute/character
at current cursor position

On entry:
AL = character to write
CX = count of characters to write
BL = attribute of character
bit-7 is set for blink
bits 6-5-4 have the value 111 for invert
bit-3 is set for high intensity
bits 2-1-0 have the value 001 for underlin

AH=12 (0CH) Write dot

On entry:
DX = row number
CX = column number
AL = 0 for clear dot, non-zero for set dot

AH=13 (0DH) Read dot

On entry:
DX = row number
CX = column number

On exit:
AL = 0 if dot cleared, non-zero if set

AH=14 (0EH) Write teletype

On entry:
AL=character to type

All the characters are sent to the terminal board, allowing all the terminal control sequences of the P2000C to be used.

Note : After a terminal reset (18H) the terminal will be in the 24 line mode. To go back to the 25 line mode issue ESC (01BH) followed by 19H.

AH=15 (0FH) Read current video state

On exit:

for 80*25 char mode:
AL = 2 similar to the mode byte used on entry
to 'set mode' (AH=0)
AH = 80 (50H) as screen width

for graphics (512*252) mode:
AL = 5 similar to the mode byte used on entry
to 'set mode' (AH=0)
AH = 64 (40H) as screen width

6.5 INT 11H - Equipment check

On exit:
AX = attached IO
as indicated by the following bits

```
bit-15     = 0
bit-14     = 1 (one printer)
bit-13..10 = 0 not used
bit-09     = 1 (one V24)
bit-08     = 0
bit-07-06  = number of floppies - 1
bit-05-04  = 1,1 (80*25 black & white video)
bit-03-02  = 1,1
bit-01     = 0
bit-00     = 1 if IPL from floppy
```

6.6 INT 12H - Memory size

On exit:
AX = available RAM size in kbytes.

6.7 INT 13H - Diskette input/output

AH=0 Reset diskette system

This command has no effect (the disk will be reset internally as required by the BIOS).

6.7.1 Floppy Disk Access

A special byte at FF00:FFCB (in the common memory) is used to tell the Z80 CPU the track density of the disk in question: the 8088 (system) program must set it to 0 for 96 tpi (P2000C disk) and 1 for 48 tpi (IBM disk).

AH=1 Read the status of the disk system

On exit:

AL = status byte

80H	= timeout
40H	= seek error
20H	= NEC controller error
10H	= CRC error
04H	= sector not found
03H	= write protect error
02H	= address mark not found
01H	= bad command received

AH=2 Read the desired sectors into memory

On entry:
DL = Drive number (0-3 allowed, not checked)
DH = Head number (0-1 allowed, not checked)
CH = Track number (0-79 allowed, not checked)
CL = Starting sector number (1-0A allowed, not checke
AL = Number of sectors to read (max 0AH, not checked)
(ES:BX) address of buffer

On exit:
AH = Status of operation (See Chapter 6.7.1, AH=1)
CY = 0 if no error (AH=0)
CY = 1 if error found
AL = number of sectors read (valid if no error)

AH=3 Write the desired sectors from memory
(For parameters see Chapter 6.7.1, AH=2)

AH=4 Verify the desired sectors
(For parameters see Chapter 6.7.1, AH=2)
The specified sectors are checked for disk error.

AH=5 Format the desired track

The parameters are listed above in the section 'AH=2'. Here, only the drive, head and track information is used. The selected track is formatted with 10 sectors/track, 512 bytes/sector and 0F6H as the format pattern.

Note: To use a disk formatted by this command it is necessary to write the empty file allocation

tables and directory onto the disk. To create a system disk it is necessary to write the bootstrap sector onto the disk, and to use the MS-DOS SYS command to copy the required system files onto the new disk.

6.7.2 Hard Disk Access

Because of the different hard disk structure (basically a CP/M device with 256 bytes/sector format) it is NOT intended to have an IBM-like calling structure. The aim is to simplify the interrupt interface to the MS-DOS I/O.

Because the MS-DOS range on the hard disk is located somewhere in the CP/M range an appropriate offset is added to the called value. This offset is determined at booting time.

The operating system has a limit to the number of records that it can handle. This is because of the system's fixed internal representation. To achieve a larger addressable capacity, each record requested by the operating system will cause two records to be read from the hard disk. So, we arrive at a logical sector size of 512 bytes and a physical sector size of 256.

MS-DOS BIOS (IBM BIOS Emulation)

The following commands are implemented:

AH=2 Read the desired sectors into memory

On entry:
DL = Drive number (80H fix for hard disk)
DH = Virtual head number (0-7FH)
CH = Virtual track number (0 - FFH)
CL = Virtual sector number (1 - 1FH)
AL = Number of sectors to read (1 - 1FH)
(ES:BX) address of buffer

On exit:
AH = Status of operation
(0 if no error, 0FFH if error)
CY = 0 if no error (AH=0)
CY = 1 if error found

AH=3 Write the desired sectors from memory

For parameters, see the read command
(Chapter 6.7.2, AH=2)

AH=4 Verify the desired sectors

For parameters, see the read command
(Chapter 6.7.2, AH=2)

The specified sectors are checked for disk error.

AH=8 Return drive parameters

It is a dummy call to avoid the 'ILLEGAL CALL's when an IBM PC disk is booted.

On exit:
AH=1 (Bad command)

6.8 INT 14H - Communication

AH=0 Initialize the communication port

On entry:
AL=Initialization parameter byte

Bits:	7	6	5	4	3	2	1	0
	-Baud rate-			Parity	Stopbit		Word-length	
	000 - 110			X0-no	0-1		10-7bits	
	001 - 150			01-odd	1-2		11-8bits	
	010 - 300			11-even				
	011 - 600							
	100 - 1200							
	101 - 2400							
	110 - 4800							
	111 - 9600							

This above baud rate overwrites the one defined in the CP/M configuration program.

On exit:

Conditions are set as in the call to command status (see Chapter 6.8, AH=3)

AH=1 Send the character in AL

On exit:

bit-7 of AH is set if the routine was unable to send the character within 20 seconds. The other bits are set as in the status request command. (See Chapter 6.8, AH=3)

AH=2 Receive one character into AL

On exit:

bit-7 of AH is set if the routine was unable to receive the character within 20 seconds. The other bits are set as in the status request command. (See Chapter 6.8, AH=3.)

AH=3 Returns the command port status in (AX)

On exit:

AH register:

Bit-7	= timeout
Bit-6	= transmitter ready
Bit-5	= same as bit-6
Bit-4	= reserved for break detect
Bit-3	= framing error
Bit-2	= parity error
Bit-1	= receiver overrun
Bit-0	= receiver ready

AL register:

Bits 7-6	= 0
Bit-5	= reserved for data carrier detect (DCD
Bit-4	= reserved for clear to send (CTS)
Bits 3-0	= 0

MS-DOS BIOS (IBM BIOS Emulation)

6.9 INT 16H - Keyboard

AH=0 Read the next character from the keyboard buffer

A character is read from the terminal board and converted through the special keyboard table read from the MSKEY.TAB file. The table has three bytes output for each character:

ASCII code
Position code
Shift code

The keyboard codes received from the terminal board are converted through this table.

On exit:
AL = ASCII code
AH = position code

There are some special keys:
80H: Print screen
(the screen contents are printed)
9AH: Wait
(waits until another key is pressed)
9CH: Break
(produces a 01BH interrupt on the 8088 side and gives 0 for ASCII and position code
F3H: ALT
(the first 3 numeric keys are considered as decimal values to let the user enter any ASCII value on the keyboard. The position code is 0; the shift code will not change)

AH=1 Read keyboard status

On exit:
  Z flag = 1 no character available
  Z flag = 0 character available
  If Z flag = 0, the character remains in the
                 buffer
  AL = ASCII code
  AH = position code

AH=2 Read shift status

On exit:
  AL = character shift status as read from the
        translation table 'MSKEY.TAB'

6.10 INT 17H - Printer

AH=0 Print character

On entry:
AL = character

On exit:
AH = 10H if no error
= 01H if timeout

AH=2 Read the printer status byte

On exit:
AH = 10H if ready
= 90H if busy

6.11 INT 19H - Bootstrap

If the system has been booted from a floppy disk then the message is displayed:

'Insert MS-DOS disk in drive-1,
type return to boot'

The system then continues to reboot as it would during the normal startup procedure.

6.12 INT 1AH - Read/set timer

AH=0 Read the current clock setting

On exit:
CX = high portion of count
DX = low portion of count
AL = 0 if timer has not passed 24 hours since the last read
= the number of passed 'midnights' since the last read

AH = 1 Set the current clock

On entry:
CX = high portion of count
DX = low portion of count

The count frequency is 1193180/65536 (about 18.2) Hz.

Each time the internal timer is incremented, a 'timer-tick interrupt' (INT 1C) is generated on the 8088 side.

7 RUNNING APPLICATION SOFTWARE

The term 'IBM compatibility' has to be explained further. Compatibility means ensuring that a lot of internal features are similar between the 8088 Copower implementation of MS-DOS and the IBM PC range of computers with PC DOS.

This chapter explains how compatibility affects the choice of software to run on the machine. We shall explain the matter so that you can decide whether it would be better to use the IBM-compatible key codes and screen control or the normal P2000C control codes and escape sequences.

Application programs designed specifically for the IBM PC and PC DOS will usually perform screen control via the IBM ROM BIOS (such actions as clear screen and scrolling), and be supplied with only an installation option to allow selection of items such as the display board and monitor. It is assumed that the normal IBM PC keyboard will be available.

To run this kind of software, it is essential to use the IBM PC-like configuration which is possible on the 8088 Copower MS-DOS implementation, so that the software can get the correct characters from the keyboard and display them on the screen. The P2000C character set cannot display all the IBM characters, but it can be configured so that most are similar. As for the others, a representation of the most suitable character has been configured.

Running Application Software

Many application programs which are available in an IBM PC (with PC DOS) version expect 100% compatible hardware to be present, and will also expect direct access to the video memory. This is not possible with the Copower board.

Application programs written in BASIC on the IBM PC will usually perform screen control via the clear screen (CLS) or set cursor position (LOCATE) functions. These functions are not available under normal Microsoft BASIC, and so the programs would have to be converted to perform screen control by control codes and escape sequences. Screen control can be performed in one of two ways on the Copower board - through an installable ANSI terminal driver or via the normal P2000C escape sequences and control codes - but not both at the same time.

IBM PC-compatible machines are usually supplied with an ANSI installable device driver which will accept ANSI escape sequences and convert them into the appropriate screen control codes for the machine. An ANSI device driver (ANSI.SYS) is supplied with the 8088 Copower MS-DOS system.

The IBM PC only recognizes the control codes of Backspace, Carriage Return, Line Feed and Bell. All other characters in the ASCII range 0 to 31, when sent to the screen, would normally be displayed as one of the characters in the character table. On the P2000C, many more characters are recognized as control characters, and would normally be intercepted by the

terminal instead of being displayed as a character.

To overcome this problem, the ANSI device driver has been modified to intercept these control characters and perform an internal conversion, so that the 'mosaic character' representation can be used and let the characters be displayed. Therefore, the normal P2000C control characters will not be available when the ANSI driver is installed (unless the application program uses the BIOS call, software interrupt 10H - AH=0E).

Many application programs are available in both a general MS-DOS form and in a specific PC DOS form. If you have a choice, it is normally better to use the general MS-DOS version.

This version will usually come with an install program that allows the selection of screen control codes and escape sequences, so that it can be configured either for the standard ANSI sequences or for the P2000C sequences and control codes. If the package is configured for the latter, the ANSI device driver must NOT be installed. If it is installed for the former, the IBM PC-like keyboard option must be configured on the P2000C.

When a general MS-DOS version of a program is installed to give ANSI sequences (forcing the user to select an IBM PC-like keyboard and screen, so that the ANSI driver can be used), it may not be able to recognize the codes generated by the function keys (cursor movement, for instance) because, being general purpose, the

program must allow operation on a normal terminal with only the standard 7-bit ASCII characters. It will usually be better to ccnfigure general MS-DOS packages to run under the normal P2000C key codes and screen codes.

The internal codes used in PC mode are incompatible with those used in the P2000C for characters which are outside the normal US ASCII character set. This means that files containing character data created in PC mode will not appear correct when used in P2000C mode if any of the national language characters, or characters with internal 'IBM PC' code greater than 127, are used - and vice versa. An example of this is German text containing characters with umlauts. On the IBM PC, these would be represented by a code greater than 127, and so would appear false when viewed on a system running with 7-bit ISO codes.

The difference in character sets means that you should use either PC mode or P2000C mode, but you should not mix operations between them.

The character set defined under CP/M-80 for the P2000C is the 7-bit ISO code, allowing the normal US ASCII characters, with national characters if applicable.

It is possible to edit the configuration tables under CP/M-80 to make alterations, adding 8-bit definitions if required.

8 USING Z80 IN THE BACKGROUND

8.1 Normal Processing

It is possible to run both the 8088 and the Z80 at the same time. Z80 code has to be loaded into the TPA (Transient Program Area) from the MS-DOS environment and called via the 8088 BIOS interrupt vector.

- There is no protection to avoid a conflict of access from both the 8088 and the Z80 to the peripherals at the same time. It is possible for one part of the code to attempt an access in conflict with the other. It is the responsibility of the programmer to make sure that such a conflict is avoided.

The ability to run background Z80 code while the 8088 is active in the foreground is useful in the implementation of some programs for the 8088, using the remainder of the 30K TPA as a buffer; for example, the running of complex communications protocols. Code like this can run totally in the background, and not need disk access that might conflict with any foreground activity. No access to the screen or keyboard (peripherals used by the 8088) would be necessary.

The 8088 BIOS code running in the Z80 processor contains code to trap a jump to 38H in the Z80 memory. When this occurs the message 'Program hangs at 38H. Please Reset' appears on the screen. This feature will often catch a Z80

program if it fails, because the program tends to execute an accidental jump to 38H (0FFH) eventually.

The Z80 program code can be loaded into the CP/M-80 TPA from an MS-DOS program, and executed from the MS-DOS evironment on request by the interrupt 'INT 07' instruction. The 8088 registers will define the Z80 address at which execution is to begin, and will allow a byte as an input parameter to be passed to the Z80 code. It will probably be necessary to transfer the code from a CP/M-80 diskette onto an MS-DOS diskette, using the MSUTIL program, so that the program code can be read as data and transferred via the MS-DOS system into the CP/M-80 TPA.

While the MS-DOS operating system is running in the foreground, the 8088 BIOS must be active and in control of the Z80 processor, so that I/O requests from MS-DOS can be processed.

While the Z80 processor is running, the 8088 processor will be halted. To avoid the possibility of the MS-DOS system 'freezing', the Z80 subroutines should return control after a reasonable time. It is possible to operate more flexibly if extensive use is made of interrupt handlers in the Z80 code, so that the code is 'driven' from the hardware rather than from the MS-DOS side.

8.2 Communications Support

For simple asynchronous communications, the interrupt 14H has been implemented to provide the status, character receive and character send functions. The interface to this interrupt call is the same as that on the IBM PC. However, this may not be adequate to support many of the communications packages available for the IBM PC. But it is possible to implement more advanced features, as we shall describe below.

An MS-DOS applications program cannot directly access the communications ports through the 8088 I/O space. It is possible, however, for the application program to call a routine which exists in the Z80 memory and have that routine executed by the Z80 processor.

This Z80 routine can access the communications ports directly, so it is possible to implement communication drivers in Z80 code. The communication with and transfer of control to the Z80 routine is via an interrupt vector through the 8088 BIOS.

The overhead of the call to the Z80 routine may mean that for high baud rates (where no protocol for transfer control, either software or electrical, is implemented), you must set up an interrupt-driven received character queue in the Z80 memory space, with start and end queue pointers at known locations. The MS-DOS application can then read the characters efficiently, directly from the 8088 memory space which will overlap the Z80 space.

Using Z80 in the Background

The Z80 routine can be placed in memory from the MS-DOS application program. The Z80 code can be written to the memory starting at address F000:09H up to but not including the position pointed to by the word in F000:06H-F000:07H. To write above this limit is not allowed because it would overwrite the 8088 BIOS routines.