CPEmulator

Z80 and 8080 CP/M Emulator

User Manual

For PCDOS

LICENSE AGREEMENT
READ CAREFULLY THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THE DISKETTE PACKAGE. OPENING THE DISKETTE PACKAGE INDICATES YOUR ACCEP TANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THESE TERMS AND CONDITIONS, PROMPTLY RETURN THE PACKAGE UNOPENED; AND YOUR MONEY WILL BE REFUNDED.

"This software is a proprietary product of GTEK, Inc. It is protected by copyright and trade secret laws. It is licensed (not sold) for use on a single micro-computer system, and is licensed only on the condition that you agree to this LICENSE AGREEMENT." GTEK, INC. provides this program and licenses its use worldwide. You assume responsibility for the use of this software to acheive your intended results, and for the installation, use and results obtained from the software.

LICENSE
The Licensee may:

a. use the program on a single machine;

b. copy the program into any machine readable or printed form for backup or modification purposes in support of your use of the program on the single machine;

c. modify the program and/or merge it into another program for your use on the single machine (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement.); and,

d. transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM, OR ANY COPY, MODIFICATION OR MERGED PORTION, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE. IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

TERM
The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

LIMITED WARRANTY

THIS PRODUCT IS NOT A CONSUMER PRODUCT WITHIN THE MEANING OF THE UNIFORM COMMERCIAL CODE AND APPLICABLE STATE LAW. THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND
PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFEC-
TIVE, YOU (NOT GTEK, INC.) ASSUME THE ENTIRE COST OF ALL NECESSARY
SERVICING, REPAIR OR CORRECTION. SOME STATES DO NOT ALLOW THE EXCLUSION
OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.
THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE
OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

GTEK, Inc. does not warrant that the functions contained in the program
will meet your requirements or that the operation of the program will be
uninterupted or error free. However, GTEK, Inc. warrants the disk-
ette(s) on which the program is furnished, to be free from defects in
materials and workmanship under normal use for a period of ninety (90)
days from date of delivery to you as evidenced by a copy of your
receipt.

Licensee herein acknowledges that the software licensed hereunder is of
the class which inherently cannot be tested against all contingencies by
Licensor. Licensee acknowledges Licensee's obligation to test all pro-
grams produced by the licensed software to determine suitability and
correctness prior to use.

LIMITATIONS OF REMEDIES
GTEK, Inc.'s entire liability and your exclusive remedy shall be:

1. the replacement of any diskette(s) not meeting GTEK's "Limited War-
ranty" and which is returned to GTEK, Inc. with a copy of your receipt,
or

2. if GTEK, Inc. or the dealer is unable to deliver a replacemant
diskette(s) which is free of defects in materials or workmanship, you
may terminate this Agreement by returning the program and your money
will be refunded.

IN NO EVENT WILL GTEK, INC. BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING
ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PROGRAM EVEN IF
GTEK, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR
ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABLITY FOR
INCINDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLU-
SION MAY NOT APPLY TO YOU.

GENERAL
You may not substitute, assign or transfer the license or the program
except as expressly provided in this Agreement. Any attempt otherwise
to sublicense, assign or transfer any of the rights, duties or obliga-
tions hereunder is void.

This Agreement will be governed by the laws of the State of Mississippi.

Should you have any questions concerning this Agreement, you may contact
GTEK, Inc. by writing to GTEK, Inc. Sales and Service, P.O.Box 289,
Waveland, MS, 39576.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND
AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT
IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN US
WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND
ANY OTHER COMMUNICATIONS BETWEEN US RELATING TO THE SUBJECT MATTER OF
THIS AGREEMENT.

Model CPEmulator, Including Speed Kit
Z8Ø CPU Emulator Manual
8Ø8Ø CPU Emulator Manual
Document #CPESKV2.1
25 March 1986

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

1. INTRODUCTION

CPEmulator is a program which enables programs written for the CPM-8Ø Version 2.2 operating system to be run on the IBM Personal Computer. While emulating the full Z8Ø microcomputer instruction set and providing the path from CPM-8Ø to PC-DOS, CPEmulator remains transparent to the user. CPM-80 programs are stored under the PC-DOS operating system the same way in which PC-DOS programs are stored; that is as .COM programs.

Additionally, features are provided for executing inline 8Ø86 code within Z8Ø programs, thus making the emulated Z8Ø more powerful than a Z8Ø CPU.

Speed kit CPEmulator users must disreguard the references to Z8Ø instructions, since the Speed Kit doesn't emulate Z8Ø instructions, only 8Ø8Ø.

The COPYCPM program allows the transfer of CPM programs and data files to and from the IBM PC environment. This program contains proprietary software copyright by Selfware, Inc., copyrighted 1983, 1984, 1985, all rights reserved.

2. SYSTEM REQUIREMENTS

CPEmulator works on an IBM PC-AT, PC-XT, PC, PCjr, or equivalent microcomputer with the following minimum requirements:

   a) 128K of ram.

   b) PC-DOS version 2.0, 2.1, 3.0, 3.1 or equivalent.

   c) One floppy disk drive.

3. INSTALLATION

To install CPEmulator do the following:

a) First, make a copy of your master disk. Put the master disk away for safe keeping. The disk may be copied by using the PC-DOS diskcopy utility. DISKCOPY A: B:

b) Next copy the programs to a disk which you intend to use by executing COPY *.* B:, where B: is your target disk.

Please take note of the following programs which were transferred:
      BIND.COM
      UNBIND.COM
      COPYCPM.EXE
The other programs are sample Z80 programs and are not required for using the emulator.

Speed Kit users see Chapter 7 for additional installation instructions.

4. OPERATION

4.1 HOW TO GET GOING QUICKLY

Operating CPEmulator is very simple and is as follows:

a) Use the COPYCPM program to copy your CPM programs onto your PC-DOS disk and vice versa. Operation of COPYCPM is menu driven and self explanatory with the following possible exception: When copying files with a [.COM] extension from a CPM diskette to a PC-DOS diskette, you will find that the filename extension has been changed to [.B80]. This is to protect you from executing them on your PC before CPEmulator is attached to the program. When transferring from PC-DOS to CPM, a [.B80] extension will be converted to a [.COM] extension.

If you transfer a native PC-DOS file with a [.COM] extension to your CPM diskette, the filename extension will be changed to [.B86]. Transferring a [.B86] from a CPM diskette to a PC-DOS diskette results in the filename extension being changed back to [.COM].

You can override this feature by specifying the destination filename when the COPYCPM utility asks for one, although this is not recommended. See Section 6- Utilities.

b) Attach CPEmulator to your files with the [.B80] filename extensions using the BIND program. The command syntax is:

    BIND FILENAME

5

The result is a program which is executable on your PC, called FILENAME.COM. You will notice that FILENAME.B8Ø no longer exists. You may now execute your program. (When you do the bind operation you must have enough free disk space to hold the program you are converting plus CPEmulator which is approximatly 8K in length. Notice that after the binding process that your program is about 8K longer. The 8080 CPEmulator is about 3.2K longer.)

That is all there is to it. The UNBIND program is for removing CPEmulator from a program, if desired. The command syntax is the same as the BIND program and the resultant program has the [.B8Ø] extension. The program may now be exported to a native CPM-8Ø computer for execution.

4.2 COMPATIBILITY WITH CPM OPERATING SYSTEM

The following items should be taken into consideration to determine how a CPM program will act under CPEmulator.

a) The Transient Program Area (TPA) when operating under CPEmulator is at least 54K bytes.

b) CPM resident commands are not present. You are operating under PC-DOS, so use PC-DOS equivalents; DIR, RENAME, ERASE, TYPE.

   SAVE can be accomplished under DEBUG and DDT. Start with DEBUG DDT.COM. Then G<cr>. Now you are operating DDT. Use I and R commands to load files if desired, make your modifications and exit with GØ<cr> or

DOS. You can save memory with the W command under DEBUG. (See PC-DOS manual for more instructions on DEBUG use.) Note: the ES segment register is pointing to the Z80 environment when DDT terminates. Set CS=ES before doing the W command.

Transient commands such as PIP will work, but are better served by the PC-DOS command COPY.

Commands such as STAT.COM make use of the allocation vector and other functions dealing with the physical properties of a CPM diskette. Since we are operating under PC-DOS, this yields unpredictable results. Use the PC-DOS command CHKDSK instead.

c) Submit files do not run under PC-DOS. Make them batch files!

d) I/O byte. The Intel I/O byte does nothing, as is the case on many CPM computers. I/O may be redirected, however, using PC-DOS I/O routing features. See Terminal Emulation.

e) Users. PC-DOS is a single user operating system and all CPM files all belong to user number Ø.

f) Case in filenames. PC-DOS treats all filenames as upper case. CPM could, although unlikely, have two unique files with the same name, one in upper case, the other lower case.

g) Read Only and System files. There are no
   R/O or SYS attributes which CPEmulator can
   use under PC-DOS. All files are Read Write
   and show up in the directory. PC-DOS .SYS
   files are not accessible using standard
   file I/O functions.

h) When entering responses to console queries,
   if the program is using the Console Buffer
   function in order to get data, and the data
   is input to the end of the buffer, PC-DOS
   will beep and require a carriage return
   whereas CPM would return as soon as the end
   of the buffer was reached.

   Line editing is that of PC-DOS, not CPM.
   See PC-DOS manual for line editing
   functions. Be advised that the DEL key on a
   PC keyboard does not produce a delete. It
   produces one of the PC's extended keyboard
   codes. A DEL is produced by depressing
   CONTROL BACKSPACE.

   Control C. You may abort a program with
   control C just like CPM, however CPM only
   recognizes this at the beginning of a line,
   whereas PC-DOS senses it anywhere in the
   line.

4.3 HOW CPEmulator REMAINS TRANSPARENT.

The following guidelines describe the way in
which .COM files which have CPEmulator
attached to them are handled when they are
manipulated under a CPM program. (There is
nothing special about them when they are
manipulated with native PC-DOS commands.)

a) When a CPM .COM program creates a .COM file or renames a file to a .COM file, CPEmulator automatically prefaces the file with itself. Thus .COM files created under CPEmulator are ready to run. If a .COM file is renamed to a file with a NON .COM extension, CPEmulator is automatically removed from that file.

b) When an existing .COM file is opened by a CPM program, the CPEmulator attached to that program is skipped over. Thus everything is where it is supposed to be, i.e. the first sector contains the first sector of the CPM program, not the emulator. What this means to you is that when programs are loaded under a .COM program, they start at 1ØØh, not some address offset by the length of the emulator. It also means you can chain to other CPM .COM programs. If you attempt to open or rename a native 8Ø86 .COM program, the function will return unsuccessful. You can NOT open native 8Ø86 .COM files from within a CPM program!

REMEMBER. These rules do not apply if you copy or rename with the native PC-DOS copy or rename command.

5. TECHNICAL REFERENCE

5.1 TERMINAL EMULATOR

A PC's video driver, REGARDLESS of whether ANSI.SYS is installed, will respond to the following control codes.

CARRIAGE RETURN	0DH
LINE FEED	0AH
BACKSPACE	08H
BELL	07H

In addition to the above, CPEmulator has resident drivers to support the following Televideo / Lear Seigler control codes. Use of these codes does NOT require the installation of ANSI.SYS. (These codes will be directed at your PC monitor, regardless of any console re-routing you may have done under PC-DOS since they access PC-DOS' bios function INT 10H directly.)

```
CONTROL L - move cursor right.
CONTROL V - move cursor down.
CONTROL K - move cursor up.
CONTROL Z - clear screen and home cursor.
ESC *     - clear screen and home cursor
ESC T     - clear to end of line.
ESC=RC    - move cursor to row, column.
              (20h20h=row 0, column 0)
```

CPEmulator DISCARDS all escape sequences other than those mentioned above EXCEPT ESCAPE [, which is the escape prefix for ANSI control.

CPEmulators' terminal emulation feature may be disabled with the following procedure.

10

```
DEBUG BIND.COM<cr>     ;load bind.com
E11D<cr>               ;substitute at 11Dh
CD 21 C3<cr>           ;new coding.
W<cr>                  ;write bind.com
Q<cr>                  ;quit
```

This permanently disables CPEmulators terminal emulation. All characters will be sent to the console with no filtering.

If you have the device driver ANSI.SYS in your CONFIG.SYS file, then your programs can do cursor positioning and other smart (dumb) terminal functions using ANSI escape sequences. (See PC-DOS manual for more on ANSI.SYS) This will also allow you to program the PC's function keys or reassign other keys.

CPEmulator will pass ANSI escape sequences REGARDLESS of whether or not its own terminal control emulator has been disabled.

5.2 BDOS FUNCTION CALLS

CPEmulator translates requested CP/M BDOS function calls as follows:

Ø: System Reset
> Returns control to PC-DOS. Control is returned to the drive which was assigned as the default drive when the program was invoked.

1: Console Input
> Direct conversion to PC-DOS. If console input returns with ØØh for the character, then the next character received is an extended keyboard character.

2: Console Output
> Direct conversion with the exception that output characters are filtered by the terminal emulator. Parity is removed from the character being output.

3: Reader Input
> Direct conversion to PC-DOS auxin, function call 3.

4: Punch Output
> Direct conversion to PC-DOS auxout function call #4.

5: List Output
> Direct conversion to PC-DOS. Parity is removed. You may patch this function to not strip parity with the following:

```
DEBUG BIND.COM<cr>    ;load bind.com
E113<cr>              ;substitute at 113h
FF<cr>                ;disable stripping
W<cr>                 ;write bind.com
Q<cr>                 ;quit
```

(use 7F<cr> to enable stripping)

6:    Direct Console I/O
      Direct Conversion. Characters output are
      filtered by terminal emulator. If DCIO
      input function returns with z flag set
      and accumulator=0, then the next
      character received is an extended
      keyboard character.

7:/8: Get I/O Byte
      Gets and Sets the I/O byte at hex
      location 3. This doesn't perform any
      useful function, however.

9:    Print String Function
      Direct conversion to PC-DOS. Characters
      output are filtered by the terminal emu-
      lator.

1Ø: Read Console Buffer
      Direct conversion to PC-DOS. See section
      4.2.h for discussion on differences be-
      tween CPM and PC-DOS.

11:   Get Console Status
      Direct conversion to PC-DOS.

12:   Return Version Number
      Returns Version 2.2

```

13: Reset Disk System
Direct conversion to PC-DOS. Default DMA
area is restored to 8Øh.

14: Select Disk
Direct translation to PC-DOS. It should
be noted that the drive would not be
marked read only as it would under CPM if
the diskette were changed, since MS or PC
DOS allows you to change the disk anyway.

15: Open File
This is fully supported within the
guidelines set forth under section 4.3.b.

16: Close File
Direct conversion to PC-DOS.

17:/18: Search First / Search Next
These functions are fully supported
as follows. They return Øffh if no match
is found or ØØh in the accumulator if a
match is found. The directory image in
the DMA area is constructed as a CPM
directory would be, with the following
exceptions;

a) For .COM files, the record count
returned includes the space used by
CPEmulator. The actual size of the CPM
program is about 8K or so less.

b) The file will always be the first
entry in the dma. Search functions never
return with A = 1 or 2 or 3. The
remaining 6Øh bytes in the DMA are filled
with ØE5h. The extent number is that of
the last extent for the file so that
programs which use (extent number * 16k)
+ (128 * record number) to calculate

file sizes will work. The block number fields are all filled with block number $\emptyset 1$.

c) Some versions of PC-DOS use relative bytes 16-31 in the file control block used for the search function. CPM does NOT, and as a result, CPEmulator must use a separate file control block for doing search functions. The consequence of this is that only one file control block is internally available for doing search functions and that an application program cannot do alternate search next operations on two different FCB's.

19:   Delete File
Direct translation to PC-DOS.

2$\emptyset$:   Read Sequential
Direct translation to PC-DOS. Recall that for .COM files, the emulator is skipped, and that relative record $\emptyset$ is the first record of the CPM .COM program.

21:   Write Sequential
Direct translation to PC-DOS. The same rules apply for .COM files as for Read Sequential function #2$\emptyset$ above.

22:   Make File
Direct translation to PC-DOS except for .COM files. In the case of a .COM file, CPEmulator is prefixed to the file, and the file is closed and re-opened. Thus, even if nothing is written to the file, the file will be the length of CPEmulator.

15

23: Rename File
Direct translation to PC-DOS, except in the case of a .COM file being renamed to some other extension or vice-versa. In the case of .COM to OTHER THAN .COM, CPEmulator is removed. For OTHER THAN .COM to .COM, CPEmulator is added from the executing program. In either of these two cases a file copy is occurring which is transparent to the user. After the copy is complete, the original file is deleted. The result of this is that there must be adequate disk space available for the file copy to take place. If the disk becomes full, the rename function returns false and the source file is not deleted.

24: Return Login Vector
Returns HL=Øffffh and ACC=Øffh. PC-DOS has no equivalent function call because drives do not have to be logged in.

25: Return Current Disk
Direct translation to PC-DOS function call.

26: Set DMA Address
Direct translation to PC-DOS.

27: Return Allocation Vector
Returns address of a fake allocation vector which has a minimum of 152 blocks of free disk space.

28: Write Protect Disk
Not meaningful to PC-DOS. This function does nothing except return to the caller.

29: Get Read / Only Vector
Not meaningful to PC-DOS, however returns
all disks read/write.

30: Set File Attributes
Not meaningful to PC-DOS, does nothing!

31: Get Address of Disk Parameters
This function is emulated in the
following respect: It returns the address
of a Disk Parameter Block which has the
following characteristics. The block size
field is always 2k. The DSM field is set
at the time that this function is called,
and reflects the size of the currently
selected drive. The address of the
returned Disk Parameter Block is the same
for all drives, but the DSM field of the
DPB is subject to change, depending on
the drive size when the function is
called.

32: Get / Set User Code
Always returns a value of zero since PC-
DOS does not support multiple users.

33:/34: Read / Write Random
Direct translation to PC-DOS functions.
Rules regarding skipping of CPEmulator
with respect to .COM files are in effect.
(See function #20 above)

35: Compute File Size
Direct conversion to PC-DOS function. In
the case of .COM files, the size of the
emulator is not included in the result.
PC-DOS creates no holes in files, so this
always returns the true size of the file.
If the size of a .COM file including
CPEmulator is desired, this information
can be extracted with function call 17.


36: Set Random Record
Direct translation to PC-DOS function.


37: Reset Drive
Not supported, does nothing except
return.


38: Not Used by CPM 2.2
Illegal, results in termination of user
program.

39: Not Used by CPM 2.2
Illegal, results in termination of user
program.

4Ø: Write Random With Zero Fill
Invokes write random function call #34.

## 5.3  BIOS FUNCTION CALLS

CPEmulator translates requested CP/M BIOS function calls as follows:

BOOT:  Same as bdos system reset function $\emptyset$. Remember that control is returned to the invoking drive, not necessarily drive A.

WBOOT: Same as boot above.

CONST: Emulated using bdos function #6 in direct console input mode.

CONIN: Emulated using bdos function #6 in direct console input mode.

CONOUT:  Uses bdos function #6 in output mode. This causes characters output to be filtered by the terminal emulator. Parity is also removed.

LIST:  Uses bdos list function #5.

PUNCH: Uses bdos punch function #4.

READER: Uses bdos reader function #3.

LISTST: Always returns list device ready, or $\emptyset$ffh.

All BIOS disk I/O functions are illegal.
The following bios disk i/o functions will result in termination of the executing program. PC-DOS disks are organized in a completely different manner than CPM disks and any CPM program which would use these functions would more than likely destroy the PC-DOS disk. These disk i/o functions are as follows:

HOME:
SELDSK:
SETTRK:
SETSEC:
SETDMA:
READ:
WRITE:
SECTRAN:

## 5.4 EXTRA FEATURES
## 5.4.1 MACRO INSTRUCTIONS

CPEmulator provides for user defined Macro Instructions as well as supporting the Z80 instruction set. This makes CPEmulator more powerful than the Z80 in many respects.

Macro Instructions are actually composed of 8086 machine code which is executed in line with the Z80 machine code. The form of the macro instruction is as follows: ED ED 62 nn xx xx xx xx C3, where ED ED 62 is the opcode for executing the next nn bytes of 8086 code. The 8086 code is represented by the xx's. The C3 is the last of the nn bytes and terminates the 8086 code so that Z80 execution may begin at the next byte.

When execution of the 8086 code begins, the following conditions are true:

        AH=FLAGS, AL=ACCUMULATOR
        CX=BC, DX=DE, BP=HL
        ES:SI=SP
        ES:DI=PC upon return to Z80
        ES:BX pointer to the following;
            AF', BC', DE', HL', IX, IY, I, R
            (note: if your using Speed Kit, you
            only have 8080 instructions, so
            do not expect to find any alternate
            register set at es:bx)

NOTE THAT THE Z80 MEMORY IS RELATIVE TO ES, NOT DS OR CS, SO BE CAREFUL IF YOU DO JUMPS OR CALLS. SHORT AND CONDITIONAL JUMPS ARE RELATIVE AND REQUIRE NO SPECIAL ACTION.

DI AND SI AND ALL SEGMENT REGISTERS MUST BE PRESERVED BY THE USER.!!

Enough confusion.  Let's look at some examples
which  will simplify the picture.  Suppose you
want  to get the system time and date.  PC-DOS
function calls 2Ah and 2Ch do just that.

The  8Ø86 code to do the get time function  is
as follows.

```
        MOV AH,2C
        INT 21h
```

Looking at these instructions under debug,  we
find the code is Øb4h,Ø2ch,Øcdh,Ø21h.

So, our macro is as follows.

```
GETTIME MACRO
        ;returns b=hours,  c=minutes,  d=seconds,
        ;e=hundreths of second
        DB    ØEDH,ØEDH,62H      ;PREFIX
        DB    Ø5                 ;FIVE BYTES TO FOLLOW
        DB    ØB4H,Ø2CH          ;MOV AH,2CH
        DB    ØCDH,Ø21H          ;INT 21H
        DB    ØC3H               ;END OF 8Ø86 CODE
        ENDM
```

Here's one for date.

```
GETDATE MACRO
        ;returns year in bc
        ;month in d, day in e
        DB    ØEDH,EDH,62H,5,ØB4H,2AH,ØCDH,21H,ØC3H
        ENDM
```

How about a multiply instruction for your Z80.

```
MULTAC MACRO
       ;DE=A*C
       DB    ØEDH,ØEDH,62H       ;PREFIX
       DB    Ø8H                 ;EIGHT BYTES OF CODE
       DB    5ØH                 ;PUSH AX (save acc)
       DB    ØF6,ØE1H            ;MUL CL
       DB    Ø89H,ØC2H           ;MOV DX,AX
                                 (result to DX)
       DB    58H                 ;POP AX
       DB    9FH                 ;LAHF (set Z80 flags)
       DB    ØC3H                ;end of 8Ø86 code
       ENDM
```

5.4.2 MORE ABOUT TIME AND DATE

In addition to being able to get time and date
using macro instructions, the time and date at
which the running program was executed can be
found at locations 41H - 48H as follows:

LOCATION	CONTAINS
41H	DAY OF WEEK
43H,42H	YEAR
45H	MONTH
44H	DAY OF MONTH
47H	HOUR
46H	MINUTE
49H	SECONDS
48H	HUNDRETHS OF SECONDS

Again, this is not a real time clock or the
current time and date, but is the time that
the program was invoked. A typical application
would be for a BASIC program to PEEK the date
from these memory locations instead of
prompting the user for the information when
the program is invoked.

5.5 INSTRUCTION SET NOTES

Besides being able to add your own macro in-
structions, CPEmulator another difference from
a Z80 that a user should be aware of and that
is input and output instructions.

Since the 8088 or 8086 is capable of doing I/O
to 65536 ports, the page, or high 8 bits of
the I/O address is stored at memory location
40h in the Z80 segment. It is initialized to 0
at execution time, so without modification,
I/O is to port 0-255. For example, if you
wanted to address the first RS232 port on a
PC, you would need to store a 3 into location
40h since the serial board I/O ports are in
page 3 of the I/O map. A Z80 puts out the
contents of the B register on the high address
lines during I/O. Again, CPEmulator uses the
contents of 40h.

6. UTILITY PROGRAMS

MS-DOS or PC-DOS:

COPYCPM

Copycpm is a program to allow you to copy CP/M programs from your CP/M format disk to a MS or PC DOS disk. The program is self-explanatory, just execute COPYCPM from within the subdirectory or from the disk that you want to copy the CP/M programs to.

We have supplied some sample CPM programs which you may find useful. They are all written in 8080 code.

CP/M:

SDIR.COM

Provides sorted directory with file sizes. Also calculates total storage occupied by the files which it displays.

MX.COM

Setup utility for Epson printers, including IBM dot matrix printer, Version 2.1. Works with GTEK's Programmable Printer switch.

CLEANUP.COM
A querying erase program. Every computer should have one.

DUMPAS.COM

A file dump utility similar to CPM dump utility, except this one shows binary and ascii.

Model CPEmulator Speed Kit
8080 CPU Emulator Manual

7. CPEmulator Speed Kit Manual
7.1 INTRODUCTION

Speed Kit is designed to augment CPEmulator in speed sensitive applications. In a PC with a 4.77 mhz clock, it allows for emulation of the 8080 instruction set at speeds approximately equivalent to a 4mhz 8080 cpu. The TPA (transient program area) is 60+ K with the Speed Kit.

Additionally, due to the 70108 (70116) cpu's advanced architecture, execution of native 8088 (8086) instructions are performed about ten percent faster on the average. This means ALL your programs will run faster, not just CPM programs.

Since hardware is required, i.e. the processor which replaces the 8088 or 8086, Speed Kit may be useful as a deterrent to application software piracy.

7.2 INSTALLATION

How to install CPEmulator Speed Kit:

a) Turn off and unplug your PC.

b) Open the PC according to your installation
   manual.

c) Take note of the orientation of the 8Ø88
   or 8Ø86 cpu on the PC mother board. The
   notched end is the end with pin 1.

d) Remove the 8Ø88 or 8Ø86 CPU.

e) Install replacement CPU supplied with
   CPEmulator Speed Kit. Be sure to orient the
   notched end the same way as the 8Ø88 (8Ø86)
   you just removed. Check the list below to
   make sure you have installed the correct
   part number for your computer.

```
    PN          REPLACES       APPLICATION
  7Ø1Ø8D-5     8Ø88,  5mhz     IBM PC and eq.
  7Ø1Ø8D-8     8Ø88,  8mhz
  7Ø116D-5     8Ø86,  5mhz
  7Ø116D-8     8Ø86,  8mhz
```

f) Put PC back together.

g) Plug in PC and turn it on. Check for proper
   operation.

h) Copy the program BINDSK.COM from your
   master disk to another disk. Put the master
   disk away for safe keeping. The file may be
   copied by using the PC-DOS copy command as
   follows: COPY A:BINDSK.COM B:

7.3   OPERATION

a) Use   the COPYCPM utility supplied with
   CPEmulator (Not Speed Kit) to copy your CPM
   programs to your PCDOS diskette. You may
   also transfer your programs via RS-232 or
   other file copy utility programs, provided
   you rename the CPM .COM programs to have a
   .B8Ø extension.
      example: mbasic.com ---> mbasic.b8Ø

b) Attach CPEmulator to your CPM files with
   the [.B8Ø] filename extensions using the
   BINDSK program. The command syntax is:

      BINDSK FILENAME

   The result is a program which is executable
   on your PC, called FILENAME.COM. You will
   notice that FILENAME.B8Ø no longer exists.
   You may now execute your program. (When you
   do the bind operation you must have enough
   free disk space to hold the program you are
   converting plus CPEmulator which is ap-
   proximatly 3.2K in length. Notice that
   after the binding process that your program
   is about 3.2K longer.)

   Use the UNBINDSK.COM program supplied with
   the Speed Kit to convert back to .B8Ø for
   exporting CPM programs.

7.4 COMPATIBILITY WITH Z8Ø CPEmulator.

Operation is the same as that for the Z8Ø CPEmulator with the following differences:

a) Only the 8Ø8Ø instruction set is supported, not the Z8Ø set. This is usually not a problem because most commercially available programs written for CPM used only 8Ø8Ø op-codes.

b) Operation of Speed Kit requires replacing the PC's cpu. This removes the portability advantage achieved with the Z8Ø CPEmulator. (This may be an advantage for those who have problems with software piracy.)

c) Since the replacement processor is thus far available for 8Ø88's and 8Ø86's only, some machines which use the 8Ø188, 8Ø186, or 8Ø286 cpu cannot use speed kit. This is not really a large disadvantage though since these machines are typically much faster and may not require any speed enhancement.

d) Input and Output Instructions. The the 8Ø8Ø can only address the first 256 I/O locations in the 8Ø88's I/O map. Therefore, if you wish to address I/O other than this, you can use 8Ø86 I/O instructions via CPEmulators Macro facility. The Speed kit cannot change the I/O page by changing the byte at 4Øh as the Z8Ø version can.

e) The BX register, upon entry to a macro instruction, points at nothing since an 8Ø8Ø has no alternate register set.

7.5 PROCESSOR COMPATIBILITY

The 7Ø1Ø8 (7Ø116) is compatible with the 8Ø88 (8Ø86) cpu in all respects (known to us at the time of this printing). Enhancements over the 8Ø88 (8Ø86) cpu includes;

a) CMOS technology, low power consumption.
b) Dual Data Bus Method speeds add, subtract, logical and compare instructions by 3Ø%
c) Effective Address generator in hardware, takes 2 clocks maximum for EA calculation in all addressing modes.
d) Temporary Shifters increase speed of multiplication and division instructions by about 4ØØ%
e) Loop Counter used by repeat instruction in block transfer or multiple shift instruction doubles speed over microcode method.
f) Enhanced and unique instructions.
g) Program Counter and Prefetch Pointer.

Programs which use the cpu execution time for timing will obviously have some reaction to the decrease in execution time provided by this cpu.