CCGFCU.WS4

CP/M-86 Compatibility Guide For CP/M-80 Users   (= CCGFCU...)

Copyright (c) 1980 Digital Research
Pacific Grove, California

(Revision of 10/21/80) (= 21 October 1980)

(Retyped by Emmanuel ROCHE. Posted in comp.os.cpm in 2003)

CP/M-86 is a single-user operating system for the Intel 8086 microprocessor. Since the Intel 8088 CPU is functionally equivalent to the 8086, CP/M-86 is suitable for use with both processors. The design philosophy of CP/M-86 follows that of CP/M for the 8080, 8085, and Z-80 microprocessors, with additional facilities to account for the increased address space of the 8086 family. The new features also allow application programs to easily upgrade to the MP/M-86 and CP/NET-86 environment, where multi-programming and computer networking is supported. This document assumes basic familiarity with the 8-bit CP/M software product, and specifically describes the differences and extensions found in CP/M-86.

1. CP/M-86 Operational Differences

CP/M-86 maintains file compatibility with previous CP/M versions, and provides a familiar environment for the operator and programmer. Utility programs, such as ED, PIP, STAT, and SYSGEN operate in the same manner as their 8-bit counterparts, while ASM and DDT provide the basic tools for assembly language development using the 8086 microprocessor.

Under CP/M-86, multiple programs are loaded in "stack order" into memory for execution. Programs themselves can cause additional programs to be loaded for subsequent execution. Thus, for example, the background DESPOOLing utility can first be loaded, followed by execution of DDT. DDT may, in turn, load a test program for a debugging session, and transfer control to the test program between breakpoints. In general, CP/M-86 keeps account of the order in which programs are loaded and, upon encountering the program abort key (Control-C), discontinues execution of the most recent program activated at the console command level. Thus, in the above case, a Control-C at the command level first discards DDT and the test program, while the second Control-C aborts DESPOOL. At this point, subsequent Control-C characters are ignored, since there are no executing programs activated at the console level.

2. Memory Organization

The 8086 memory addresses range from 00000H to 0FFFFFH, where each memory location holds an 8-bit value, resulting in a one megabyte address space. The following terms are used throughout this document when 8086 memory organization is discussed:

Nibble	4-bit half-byte
Byte	8-bit value
Word	16-bit value
Double Word	32-bit value
Paragraph	16 contiguous bytes
Segment	Up to 64K contiguous bytes
Offset	16-bit displacement in a segment
Group	one or more segments

Word values are standard Intel format, with low order byte stored first in memory. Double words consist of two word values in standard format, and often represent an offset followed by a paragraph address. All paragraph addresses are on even 16-byte boundaries, and thus a paragraph address has an assumed low-order nibble value of 0, in order that the address can be stored in a word location. The paragraph address 0000H, for example, represents the actual memory location 00000H, while the paragraph address 0001H becomes the memory location 00010H.

CP/M-86 itself consists of the Console Command Processor (CCP), the Basic Disk Operating System (BDOS), and the user-configurable Basic I/O System (BIOS). All three portions are resident, and are not available as data space after a program is loaded. CP/M-86 memory consists of a sequence of (possibly) non-contiguous areas, which are addressed by a memory management scheme within CP/M-86. Up to eight non-contiguous memory areas can be defined within the BIOS, in order to provide basic allocation information. When non-contiguous memory areas are mapped in this manner, the base memory addresses must be on 16-byte paragraph boundaries. Normally, however, 8086 memory consists of contiguous RAM beginning at location zero, and thus only one memory area is defined within the BIOS.

The Boot Loader resides on the first two system tracks. For future expansion, however, the CP/M-86 system itself is read from the CPM.SYS file stored on the system disk. Thus, the Boot Loader contains a simple version of the CP/M-86 BDOS, which allows the CPM.SYS file to be opened and read into memory, similar to the operation of MP/M. The actual load address is determined when the Boot Loader is configured. CP/M-86 is not normally operated in the area from location 0 through 3FFH, since this area is often used for interrupt vector information. For standardization, it is suggested that CP/M-86 be loaded beginning at location 400H. In any case, the BIOS memory map specifically excludes the interrupt vector area and the memory area in which CP/M-86 resides, so that this area is not available for allocation to user or system programs.

3. Memory Models

CP/M-86 loads programs into the Transient Program Area (TPA), using three different memory models corresponding to three popular memory segmentation techniques. The three models are:

8080 Model
Small Model
Compact Model

The 8080 Model supports programs which are directly translated from the 8-bit CP/M environment, where code and data areas are intermixed. The Small Model is similar to that defined by Intel, where code and data spaces are separated into two segments of up to 64K bytes each. The Small Model is suitable for use by translated 8-bit programs where code and data are easily separated. The Compact Model allows execution of programs which perform their own segment management, allowing programs with code and data segments which reach the addressing capacity of the host memory configuration. The three models differ primarily in the manner in which the segment registers are initialized upon program loading.

In all three models, the DS register addresses a "base page" area, similar to 8-bit CP/M where various default values are stored. The base page extends from offset 0000H through offset 00FFH from the DS register, and contains the following pre-defined areas:

```
        +-----+-----+-----+
DS + 0000H: | LC0 | LC1 | LC2 |
            +-----+-----+-----+
DS + 0003H: | BC0 | BC1 | M80 |
            +-----+-----+-----+
DS + 0006H: | LD0 | LD1 | LD2 |
            +-----+-----+-----+
DS + 0009H: | BD0 | BD1 | xxx |
            +-----+-----+-----+
DS + 000CH: | LE0 | LE1 | LE2 |
            +-----+-----+-----+
DS + 000FH: | BE0 | BE1 | xxx |
            +-----+-----+-----+
DS + 0012H: | LS0 | LS1 | LS2 |
            +-----+-----+-----+
DS + 0015H: | BS0 | BS1 | xxx |
            +-----+-----+-----+
DS + 0018H: | LX0 | LX1 | LX2 |
            +-----+-----+-----+
DS + 001BH: | BX0 | BX1 | xxx |
            +-----+-----+-----+
DS + 001EH: | LX0 | LX1 | LX2 |
            +-----+-----+-----+
DS + 0021H: | BX0 | BX1 | xxx |
            +-----+-----+-----+
DS + 0024H: | LX0 | LX1 | LX2 |
            +-----+-----+-----+
DS + 0027H: | BX0 | BX1 | xxx |
            +-----+-----+-----+
DS + 002AH: | LX0 | LX1 | LX2 |
            +-----+-----+-----+
DS + 002DH: | BX0 | BX1 | xxx |
            +-----+-----+-----+
DS + 0030H:        Not
  ...           Currently
DS + 005BH:        Used
            +-----------------+
DS + 005CH: |   Default FCB   |
            +-----------------+
DS + 0080H: |  Default Buffer |
            +-----------------+
DS + 0100H: | Begin User Data |
            +-----------------+
```

where each byte is indexed by 0, 1, and 2, corresponding to the standard Intel storage convention of low, middle, and high-order (most significant) byte, and "xxx" marks unused bytes. LC is the last code group location (24-bits), while BC is the base paragraph address of the code group (16-bits). In the 8080 Model, the low order bytes of the LC (LC0 and LC1) never exceed 0FFFFH, and the high order byte (LC2) is always zero. It should be noted that bytes LD0 and LD1 appear in the same relative positions of the base page in both 8-bit CP/M-80 and 16-bit CP/M-86, thus easing the program translation task.

LD and BD provide the last position and paragraph base of the data group. Note that the last position is one byte less than the group length. The M80 byte is equal to 1 when the 8080 Memory Model is in use. LE and BE provide the length and paragraph base of the (optional) extra group, while LS and BS give the (optional) stack group length and base. The bytes marked LX and BX correspond to a set of four (optional) independent groups, which may be required for programs which execute using the Compact Model. The initial values for these descriptors are derived from the header record in the memory image file, described below.

4. Memory Image File Format

Similar to 8-bit CP/M, CP/M-86 loads and executes memory image files, with the file type "CMD" corresponding to "command" files. The command file results from the GENCMD program, which accepts either Intel L-modules, or Intel 8086 "hex" files produced using Intel translators or the Digital Research ASM-86 assembler. A CMD file begins with a header record created by the GENCMD program. The header record is a total of 128 bytes, and begins with a sequence of one or more group descriptors of nine bytes each, with zero fill to the end of the record. The format of each descriptor is shown below.

```
  8-bit   16-bit     16-bit     16-bit     16-bit
+------+----------+----------+----------+----------+
|G-Form| G-Length |  A-Base  |  G-Min   |  G-Max   |
+------+----------+----------+----------+----------+
```

where G-Form describes the group form, or has the value zero if no more descriptors follow. If G-Form is non-zero, then the 8-bit value is decomposed into two fields:

```
               G-Form:
           4-bit    4-bit
        +---------+--------+
        | x x x x | G-Type |
        +---------+--------+
```

G-Type	Group Type
1	Program Code Group
2	Independent Data Group
3	Independent Extra Group
4	Independent Stack Group
5	Independent Auxiliary Group #1
6	Independent Auxiliary Group #2
7	Independent Auxiliary Group #3
8	Independent Auxiliary Group #4
9-14	Unused, but Reserved
15	Escape Code for Additional Types

All remaining values in the group descriptor are given in increments of 16-bytes. That is, each value is the high order 16 bits of the 20-bit base or length, with an assumed low order nibble of zero. Thus, allocation requests are always in "paragraph" increments. G-Length gives the actual size of the group. Given a G-length of 0080H, for example, the size of the group is 00800H = 2048 bytes. A-Base defines the base paragraph address for a non-relocatable group. The group is assumed relocatable if A-Base is 0000H. G-Min and G-Max define the minimum and maximum size of the memory area to allocate to the group. Normally, the value of G-Length, G-Min, and G-Max are identical for a group containing object code, but may differ when the group designates a data area. A program which performs I/O processing, for example, may contain a data area used for buffers where the size of the allocated area may vary, depending upon available storage.

The particular memory model used by a transient program is determined implicitly by the group descriptors. The 8080 Model occurs when only a code group is included, since no independent data group is named. The Small Model is implied by the occurrence of a code group and a data group, but no additional group descriptors. Otherwise, the Compact Model is implied.

5. Program Initialization

Following program load, the CCP transfers control through an 8086 Far Call. The Stack Segment register (SS) is set to the base paragraph address of CP/M-86, while the Stack Pointer (SP) references the current base of the CCP stack. Similar to 8-bit CP/M, the transient program must save these registers before switching to a local stack if return to the CCP is anticipated. In this case, the transient program must reinstate the SS and SP registers, and execute an 8086 Far Return. Alternatively, the program may return control to CP/M-86 through a call to the BDOS, using function code 0, as described below. The initial values of the remaining segment registers upon program load are determined by the memory model.

When the 8080 Model is used, the Code Segment register (CS), Data Segment register (DS), and Extra Segment register (ES) are set to the base of the transient program area. The Instruction Pointer (IP) is initialized to 0100H. The reserved locations and initial values for maximum memory size, default FCB, and default buffer in the base page are in the same relative position as 8-bit CP/M. In particular, the base page is a part of both the code and data space, with the size of memory provided at address DS+0006H.

In the case of the Small Model, the CS register is set to the base of the code area, while the DS and ES registers are initialized to the base paragraph address of the data area. Again, the base page addressed by DS and ES correspond closely to 8-bit CP/M, in order to simplify translation to CP/M-86. Note that the only essential difference between the 8080 Model and the Small Model is that code and data must be separate, which is often the case with well-structured 8-bit programs. The Small Model has the advantage that the code and data space is not limited to a total of 64K bytes.

Finally, in the case of the Compact Model, the CS, DS, and ES registers are initialized to the base paragraph addresses of the code, data, and extra groups, respectively. It is the responsibility of the transient program to manage the various segment registers from the base page values filled-in when the program is loaded by the CCP if any group exceeds 64K bytes, or if auxiliary groups are included.

6. BDOS Function Calls

Programmatic interface to CP/M-86 corresponds to 8-bit CP/M with only a few exceptions, thus allowing simple translation of existing 8-bit programs to the 8086 environment.

The BDOS is entered using 8086 interrupt 224, which overlaps that of Intel's RMX-86 monitor. Register values upon entry to the BDOS are given below:

Function Code	CX
Byte Parameter	DL
Word Parameter	DX

Addresses passed to the BIOS are offset from the DS register which is positioned at the base of the CP/M-86 data area. Note that the total code and data space required by the CCP, BDOS, and BIOS is considerably less than 64K, and thus CS and DS are placed at the first address of the region occupied by CP/M.

Register content is not preserved through BDOS calls. Values resulting from BDOS calls are returned in the following registers:

Byte Value	AL
Word Value	AX and BX
Double Word Offset	BX
Segment Address	ES

A number of additional function codes are provided in CP/M-86. Note also that function code (0) has a single parameter value under CP/M-86.

(0) System Reset (Warm Start)
Restores CP/M-86 to the reset state, with a standard CCP prompt. The parameter value in the DL register has two possible values: if DL = 00H, then the currently active program is terminated (equivalent to Control-C). Otherwise, DL must be 01H, in which case the program remains in memory, and the memory allocation state remains unchanged. (If DL is neither 00H or 01H, it is currently treated as if it were 00H.)

(50) Direct BIOS Call
Transfers control through the BDOS to the BIOS, with the DX addressing a five-byte memory area containing the BIOS call parameters:

```
  8-bit    16-bit       16-bit
+------+------------+------------+
| Func | Value (CX) | Value (DX) |
+------+------------+------------+
```

where Func is the BIOS function number, starting at zero, and Value (CX) and Value (DX) are the 16-bit values which would normally be passed directly in the CX and DX registers with the BIOS call. The CX and DX values are loaded into the 8086 registers before the BIOS call is initiated.

(51) Set DMA Segment Base
Sets the base register for subsequent DMA transfers. The word parameter gives the paragraph address of the referenced segment. Note that, upon initial program loading, the default segment base is set to DS, and the DMA offset is set to 0080H, which provides access to the base page.

(52) Get DMA Base
Returns the double word value corresponding to the current DMA segment base and offset.

Several of the codes given below reference a Memory Control Block (MCB), defined in the transient program, which takes the form:

```
           16-bit     16-bit    8-bit
        +----------+---------+-------+
   MCB: |  M-Base  | M-Length | M-Ext |
        +----------+---------+-------+
```

where M-Base and M-Length are either input or output values expressed in 16-byte paragraph units, and M-Ext is a returned byte value, as defined with each function below. Note that an error condition is normally flagged with a 0FFH returned value, in order to match the file error conventions of CP/M.

(53) Get Max Mem
Allocate the largest contiguous area of memory. Upon entry, DX addresses a MCB which will be filled-in by the BDOS. If successful, M-Base is set to the base paragraph address of the allocated area, and M-Length is the paragraph length of the allocation. AL has the value 0FFH upon return if the requested memory is not available, and 00H if the request was successful. M-Ext is set to 1 if there is additional memory for allocation, and 0 if no additional memory is available.

(54) Get Abs Max
Allocate the largest contiguous memory area at the absolute address given by M-Base. Returned values are the same as those of function (53).

(55) Get Mem
Allocate a memory area according to the MCB addressed by DX. In this case, the allocation request size is obtained from M-Length on input. The resulting values of the MCB fields are identical to function (53).

(56) Get Abs Mem
Allocate memory at the absolute address given by M-Base, for the length given by M-Length. The resulting values are identical to function (53).

(57) Free Mem
Release the memory area of length M-Length at location M-Base given in the MCB addressed by DX.

(58) Free All
Release all memory in the CP/M-86 environment (normally used only by the CCP upon initialization).

(59) Program Load
Load the program in the file described by the FCB addressed by DX. AX has the value 0000H if the program load was unsuccessful. Otherwise, AX and BX both contain the paragraph address of the base page belonging to the loaded program. Note that, upon program load at the CCP level, the default paragraph address is initialized to the base page of the loaded program, and the default DMA address is initialized to offset 0080H. Note, however, that this is a function of the CCP, and thus a function

59 does not initialize these registers. In this case, it is the responsibility of the program which executes function 59 to first execute function 51 to set the DMA base, and then initialize the DS register before passing control to the loaded program.

In addition to these functions, two specific differences between 8-bit CP/M and CP/M-86 exist. First, the IOBYTE function is only accessible through the BDOS, as described in the 8-bit CP/M documentation. Second, Direct BIOS calls are provided only through the BDOS.

7. BIOS Interface

The interface to the CP/M-86 BIOS is identical to 8-bit CP/M, with the addition of four jump vector elements. The BIOS jump vector consists of a sequence of 3-byte 8086 Near Jumps to the individual subroutines. The additional BIOS entry points are listed below, and immediately follow the SECTRAN (Sector Translate) entry point defined in 8-bit CP/M version 2:

```
        JMP     SETDMAB         ; Set DMA Base segment address
        JMP     GETSEGT         ; Return address of SEGment Table
        JMP     GETIOB          ; Get I/O Byte
        JMP     SETIOB          ; Set I/O Byte
```

Entry points receive their first parameter in CX, and (optional) second parameter in DX. Values are returned as described in the BDOS interface, above.

SETDMAB (Set DMA Base) sets the base paragraph address for subsequent DMA operations.

The GETSEGT returns the offset to the BIOS-resident Memory Region Table (MRT). The Memory Region Table has the form:

```
               8-bit
             +-------+
        MRT: | R-Cnt |
             +---------------+---------------+
          0: |    R-Base     |   R-Length    |
             +---------------+---------------+
          1: |    R-Base     |   R-Length    |
             +---------------+---------------+
                            ...
             +---------------+---------------+
          n: |    R-Base     |   R-Length    |
             +---------------+---------------+
                  16-bit          16-bit
```

where R-Cnt is the number of memory region descriptors (equal to n+1 in the above diagram), while R-Base and R-Length give the paragraph base and length of each physically contiguous area of memory, not including the interrupt vector addresses (0-3FFH), or the area of memory where CP/M-86 resides. If all memory is physically contiguous, R-Cnt = 1 and n = 0. In this case, the single region descriptor normally addresses the first paragraph beyond the last BIOS address, with an R-Length which allows access to the highest paragraph address in the region.

The GETIOB and SETIOB entry points return and change the IOBYTE, as defined in 8-bit CP/M. The IOBYTE value itself is maintained in the BIOS.

EOF